

- Postel, J.: *Domain Name System Structure and Delegation*. RFC 1591. Marzo, 1994.
- Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V.: *RTP: A Transport Protocol for Real-Time Applications*. Audio-Video Transport Working Group. RFC 3550. Julio, 2003.
- Schulzrinne, H.; Rao, A.; Lanphier, R.: *Real Time Streaming Protocol (RTSP)*. RFC 2326. Abril, 1998.
- Sollins, K.; Masinter, L.: *Functional Requirements for Uniform Resource Names*. RFC 1737. Diciembre, 1994.
- Stevens, W.R.: *TCP/IP Illustrated, Vol. 1. The Protocols*. Ed. Addison Wesley, 2000.
- VanBokkelen, J: *Telnet Terminal-Type Option*. RFC 1091. Febrero, 1989.
- Ylonen, T.; Lonwick, C.: *The Secure Shell (SSH) Protocol Architecture*.RFC 4251. Enero 2006.



GESTIÓN Y SEGURIDAD EN REDES

- 12.1. Introducción
- 12.2. Gestión de redes
- 12.3. Seguridad en redes de computadores

12.1. Introducción

Uno de los aspectos más importantes de una red de computadores es el concerniente a su gestión. El concepto de gestión de red es muy amplio puesto que hace referencia a la supervisión y mantenimiento de todas aquellas facetas de un sistema en red que afectan o pueden afectar a las prestaciones de este en cuanto al número y características de los servicios y recursos ofertados, eficiencia en el uso de los mismos, accesos permitidos, etc. Si bien esto puede resultar excesivamente vasto y complejo, es una tarea perfectamente definida en sus objetivos y primordial para el adecuado funcionamiento de todo sistema en red.

Función cuya responsabilidad principal recae en el administrador del sistema, la gestión de red, según OSI, cubre los siguientes aspectos:

- *Control de la planificación*. El proceso de gestión de una red no se inicia cuando esta ya se encuentra en funcionamiento, sino que parte desde su propia concepción. Resulta fundamental así un adecuado estudio de las necesidades a cubrir y del dimensionado planteado tanto en infraestructura como en servicios; debiéndose tomar ya desde sus orígenes decisiones de importancia que marcarán la vida futura de la red a desplegar.
- *Gestión de la configuración*. Un sistema de red es algo «vivo», dinámico, y así debe ser si, como es previsible, las necesidades, los usuarios, los servicios, etc., varían a lo largo del tiempo. Se entiende, pues, que el proceso de planificación mencionado anteriormente no es algo puntual que se realiza al inicio de la vida del sistema, sino que se extiende a lo largo del tiempo y

permite la actualización permanente de aquel en cualquiera de sus facetas. En fin, es importante controlar la incorporación de cambios en la configuración que conlleven la consiguiente modificación en la estructura física y/o lógica del sistema.

- **Control de la eficiencia.** Dada una cierta configuración, digamos estable, del sistema, uno de los aspectos más importantes de su gestión hace referencia a la supervisión de la eficiencia alcanzada. En este sentido, aunque obvio, hay que decir que el interés del administrador o gestor del sistema se centrará en la optimización de las prestaciones de los elementos que componen la red y, por ende, del sistema global. Para ello se hace imprescindible una monitorización continua y pormenorizada de todos los recursos existentes en el entorno.
- **Gestión de fallos.** Si en el punto anterior se ha mencionado la necesidad de optimizar las prestaciones del sistema, no digamos ya la conveniencia de llevar a cabo un proceso de supervisión a través del que poder determinar la ocurrencia de fallos y problemas que impidan el normal funcionamiento del entorno. La gestión de fallos es un aspecto básico por cuanto que la aparición de estos implicará generalmente la deficiencia de servicio/s, si no su inhabilitación completa, de cara a los usuarios. En este sentido, resulta necesario llevar a cabo una monitorización que permita, en la medida de lo posible, prevenir la ocurrencia de problemas y, en caso de que estos se produzcan, resolverlos con la mayor prontitud posible de acuerdo con una adecuada categorización de severidad de los mismos y consideración de elementos de respaldo si se estimase oportuno. Este último aspecto debería ser tenido en cuenta en las fases de planificación y de configuración del sistema.
- **Gestión de la seguridad.** Un aspecto de gran y creciente relevancia en los últimos años es el de la seguridad. Todo sistema conectado en red es, por definición, inseguro. Esta inseguridad puede manifestarse en dos aspectos diferenciados: desde el punto de vista de las transacciones de información involucradas en las comunicaciones y desde la perspectiva del acceso a los recursos y/o servicios disponibles en el sistema. La utilización de herramientas diseñadas al efecto, al tiempo que el respeto de una serie de normas de funcionamiento, permitirán al administrador la prevención, detección y corrección de situaciones no deseadas tales como comunicaciones inseguras, accesos no autorizados, etc.
- **Control de la tarificación.** Una última cuestión importante en todo proceso de gestión de un sistema en red es la referente al control del uso que de los recursos hacen los usuarios a fin de, si procede, permitir su adecuada tarificación desde un punto de vista económico. Diversas son las técnicas desarrolladas al efecto, debiéndose tener en cuenta a la hora de elegir una u otra cuestiones tales como carga extra introducida en el sistema, complejidad de implementación y compatibilidad con la tecnología de red considerada, uso como técnica de control de tráfico y congestión, etc. Así, por ejemplo, la técnica de tarificación de todos conocida como *tarifa plana*, consistente en una cuota económica fija independiente de cualquier consideración respecto del uso de los recursos realizado, tiene como principales ventajas la sencillez de implementación y el no consumo de recursos extra del sistema, pero, por el contrario, presenta el inconveniente de que no resulta persuasiva de cara a un potencial «abuso» de los recursos del sistema por parte de los usuarios.

No es objetivo de este tema plantear un estudio en profundidad de todos y cada uno de los aspectos de gestión anteriormente comentados; eso se pospone para cursos más avanzados. En nuestro caso, dos serán básicamente las cuestiones desarrolladas a lo largo del presente capítulo: modelos de gestión y seguridad en redes. Respecto de la primera de ellas se presentarán no tanto herramientas específicas de gestión existentes en el mercado¹, como el conjunto de estándares desarrollado al efecto y en

¹ Son numerosas las herramientas de gestión de red actualmente disponibles, tanto comerciales (CiscoWorks, HP OpenView, Tivoli NetView de IBM, etc.) como de código abierto (OpenNMS, Nagios, Cacti, etc.).

el que se fundamentan los productos comerciales. En este sentido, el Apartado 12.2 presenta un estudio de los modelos de gestión SNMP y CMIS/CMIP en los que se sustenta, respectivamente, la gestión de redes TCP/IP y OSI.

Con la penetración social experimentada por Internet en los últimos años, el tema de la seguridad en redes se ha convertido en preocupación y responsabilidad principales de los administradores de sistemas en red. De la provisión de ella depende la confianza de los usuarios en las redes y servicios y, en suma, el éxito social de las tecnologías de la información y las comunicaciones. Dada la importancia y actualidad del tema, en el Apartado 12.3 del capítulo se estudian las distintas facetas de la seguridad y se presentan diversas técnicas desarrolladas para la consecución de esta. Como ámbitos específicos de aplicación de todo ello se discutirá la disposición de protocolos de comunicación seguros y la existencia de tecnologías de control de accesos. Al final del apartado, y como conclusión del capítulo, se estudiará la seguridad en sistemas finales de usuario y se darán unos breves apuntes sobre aspectos legales y éticos de la seguridad.

12.2. Gestión de redes

La gestión de red forma parte de lo que se conoce como *Operations, Administration and Management* (OAM), consistente en la supervisión y mantenimiento de todas aquellas facetas de un sistema de red que afectan o pueden afectar a las prestaciones de este en cuanto al número y características de los servicios y recursos ofertados, eficiencia en el uso de los mismos, accesos permitidos, etc. En suma, como se ha establecido en el Apartado 12.1, el conjunto de actividades orientadas a racionalizar la operación del entorno en cuestión.

12.2.1. Fundamentos de gestión

Para conseguir el fin perseguido con la gestión de un sistema en red, cuatro son los elementos funcionales involucrados:

- **Gestor o estación de gestión:** entidad encargada de la gestión a partir de la obtención de información relacionada. Por clarificar, un gestor se refiere a una aplicación de control que se ejecuta típicamente en una máquina final de usuario y que solicita información de estado.
- **Agente:** entidad que interacciona con el gestor para proporcionar a este información de gestión. Una aplicación agente puede ser instalada en cualquier dispositivo susceptible de gestión y control (*hosts*, *routers*, impresoras, etc.) a fin de controlar su estado.
- **Protocolo de comunicación:** procedimiento que permite la interacción gestor-agente.
- **Objeto:** variable que contiene información de gestión y que es intercambiada entre el gestor y el agente; por ejemplo, el número de paquetes ICMP de «solicitud de eco» recibidos en un *host*, el número de trabajos pendientes en una impresora, etc.

Tomando como base constitutiva estos elementos, un sistema de gestión queda definido a través de los siguientes modelos:

- **Modelo organizativo:** conjunto de componentes y relaciones entre ellos (véase Figura 12.1).
- **Modelo de información:** sintaxis y semántica, o SMI («Structure of Management Information»), de los objetos para su referencia. El conjunto de estos constituye la base de datos de información, o MIB («Management Information Base»).
- **Modelo de comunicaciones:** conjunto formado por el protocolo de transporte que define la interacción gestor-agente y por el protocolo de aplicación que establece el grupo de comandos y respuestas (esto es, tipos de paquete) aceptados entre ambas partes.

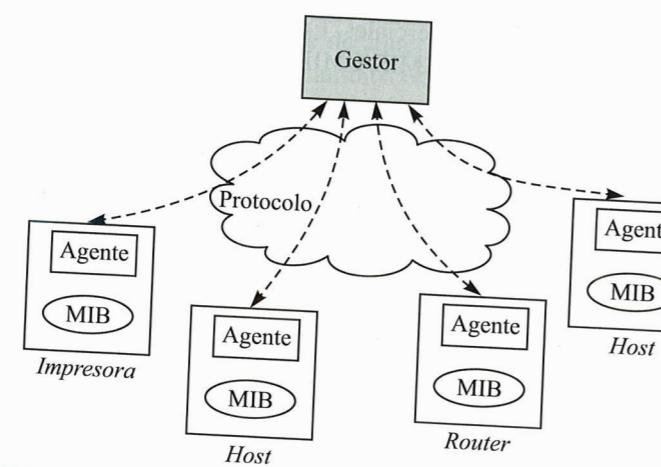


Figura 12.1. Modelo organizativo de la gestión de redes.

El desarrollo concreto de todo lo anterior da lugar a la existencia de distintos modelos o arquitecturas de gestión de red, siendo los más conocidos SNMP («Simple Network Management Protocol»), para redes TCP/IP, y CMIS/CMIP («Common Management Information Service/Common Management Information Protocol»), para redes OSI. Otros también conocidos aunque no tratados aquí son TMN («Telecommunications Management Network»), desarrollado por la ITU, y los basados en web como WBEM («Web-Based Enterprise Management») y JMX («Java Management eXtensions»).

12.2. Gestión de redes TCP/IP

La gestión de redes TCP/IP se basa en el modelo SNMP («Simple Network Management Protocol»). Fundamentado en los cuatro elementos funcionales ya comentados con anterioridad, la característica diferencial de este modelo de gestión es el empleo del protocolo que le da nombre: SNMP, el cual conforma el conjunto de reglas y mensajes intercambiados entre el gestor y los agentes para llevar a cabo determinadas acciones sobre los objetos MIB. Entre los mensajes SNMP podemos destacar dos: Get, que permite obtener el valor de un objeto, y Set, que permite establecer su valor. Seguidamente se discute en mayor detalle este protocolo.

Protocolo SNMP

El protocolo simple de gestión de red, SNMP, se sitúa en la capa de aplicación TCP/IP, implementándose el servicio sobre el puerto 161; usualmente sobre UDP, aunque también se permite sobre TCP. En la Figura 12.2 se indica la situación de SNMP en la pila de protocolos TCP/IP y el esquema seguido en la comunicación gestor-agente de gestión.

Los mensajes SNMP intercambiados entre el agente y el gestor son los siguientes (ver RFC 1157 para SNMPv1):

- **GetRequest.** Mensaje enviado por el gestor hacia el agente a fin de solicitar el valor de una variable en la MIB.
- **GetNextRequest.** Mensaje enviado por el gestor hacia el agente para solicitar el valor de la siguiente variable. Este mensaje es útil, por ejemplo, para la transferencia de información estructurada en forma de árbol.

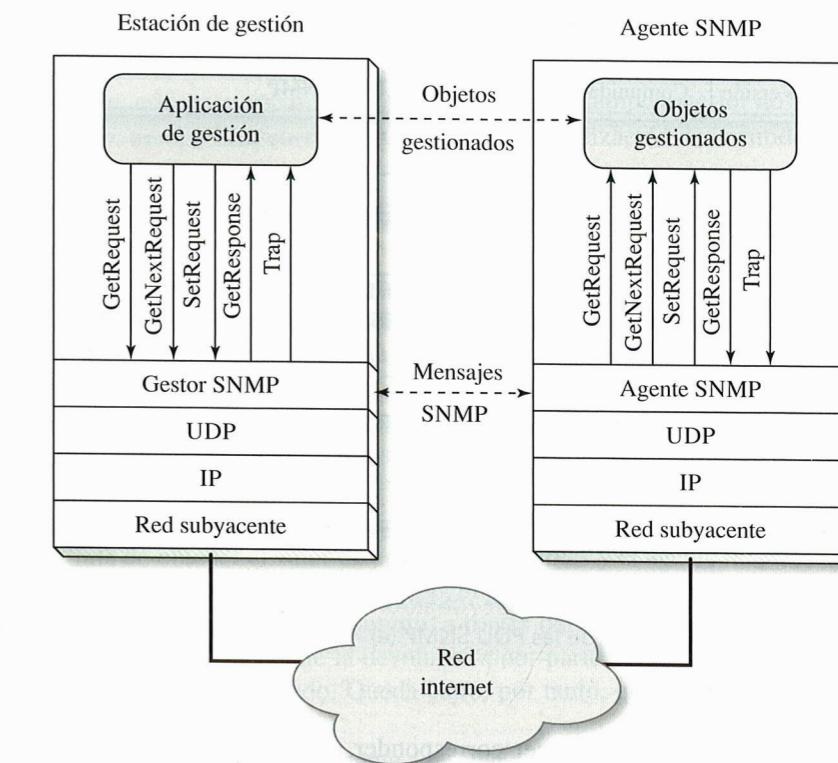


Figura 12.2. Situación de SNMP en la pila TCP/IP.

- **SetRequest.** Mensaje enviado por el gestor hacia el agente a fin de solicitar la modificación del valor de una variable.
- **GetResponse.** Mensaje enviado por el agente en respuesta a cualquiera de los tres anteriores.
- **Trap.** Cuando de un gestor depende un gran número de agentes, resulta inabordable desde un punto de vista práctico la consulta periódica de todos los objetos gestionados. Para solucionar este problema, SNMP considera el mensaje Trap, a través del cual, generado por los agentes, se lleva a cabo la comunicación asíncrona (sin necesidad de consulta previa) hacia el gestor ante cualquier eventualidad en alguno de los objetos gestionados.

El mensaje Trap involucra una comunicación adicional sobre el puerto 162.

En la Figura 12.3 se muestra el formato de los distintos mensajes SNMP comentados. Por lo que se refiere al formato general, y a los comandos Get y Set en particular, los campos involucrados son los siguientes:

- **Versión:** versión de SNMP.
- **Comunidad:** nombre que reciben las parejas agente-entidades de aplicación, entendiendo por «entidades de aplicación» las residentes en la estación de gestión así como los elementos de red que se comunican entre sí a través de SNMP.
- **Tipo:** campo que indica el tipo de PDU o mensaje SNMP de que se trata: GetRequest, GetResponse, etc.

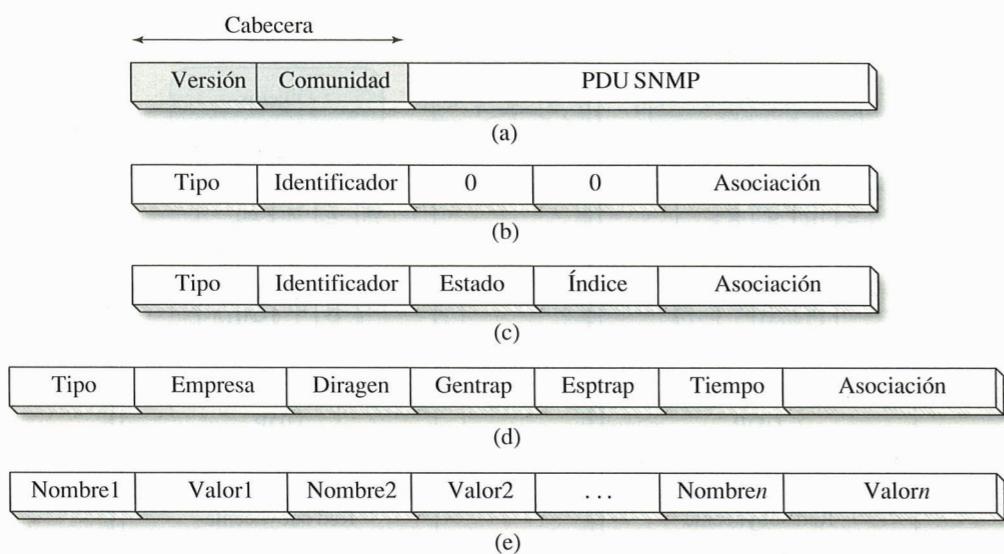


Figura 12.3. Mensaje SNMP (a) y PDU GetRequest/GetNextRequest/SetRequest (b), GetResponse (c) y Trap (d). Formato del campo asociación de las PDU SNMP (e).

- *Identificador*: campo para hacer corresponder solicitudes con respuestas. También puede usarse como método para detectar mensajes duplicados.
- *Asociación*: relación de nombres de variables y valores asociados (Figura 12.3(e)). A través de este campo se permite la ejecución de varias operaciones del mismo tipo en un solo mensaje.
- *Estado*: campo que indica la detección de un error en el procesamiento de la solicitud.
- *Índice*: en caso de error, este campo señala la variable dentro de una lista que lo ocasionó.

Por su parte, la PDU Trap consta de los siguientes campos:

- *Tipo*: como antes.
- *Empresa*: tipo de objeto que provocó el mensaje (ver MIB más adelante).
- *Diragen*: dirección del agente que generó el mensaje.
- *Gentrapp*: tipo genérico de mensaje Trap, el cual puede tomar los siguientes valores:
 - 0 (*coldStart*) → La entidad de protocolo emisora va a reinicarse, pudiendo verse alterada la configuración del agente o la implementación de la entidad de protocolo.
 - 1 (*warmStart*) → La entidad de protocolo emisora va a reinicarse, no provocando alteración ni en la configuración del agente ni en la implementación de la entidad de protocolo.
 - 2 (*linkDown*) → La entidad de protocolo emisora ha detectado un error en uno de los enlaces presentes en la configuración del agente.
 - 3 (*linkUp*) → La entidad de protocolo emisora ha detectado la activación de uno de los enlaces presentes en la configuración del agente.
 - 4 (*authenticationFailure*) → Fallo en la autenticación de un mensaje de protocolo. El servicio de autenticación solo sirve para verificar que el nombre de comunidad autoriza la recepción de mensajes procedentes de la entidad SNMP origen.
 - 5 (*egpNeighborLoss*) → Ha desaparecido una relación de vecindad con un vecino EGP (ver Apartado 9.3).

- 6 (*enterpriseSpecific*) → La entidad de protocolo reconoce la ocurrencia de un evento de empresa específico.

- *Esptrap*: código específico presente incluso si el campo anterior no toma el valor 6.
- *Tiempo*: tiempo transcurrido entre la última inicialización de la entidad de protocolo y la generación del mensaje Trap.
- *Asociación*: como en el resto de PDU SNMP, relación de nombres de variables y valores asociados.

SNMPv1 solo permite el intercambio de una cierta cantidad de información en cada transacción. Por ello, SNMP puede implicar una cierta sobrecarga en la red. Para evitar este problema y mejorar la eficiencia de la transferencia, la versión 2 de SNMP, SNMPv2 (RFC 1901-1908), introduce un comando adicional a los anteriormente comentados: GetBulk. Enviado desde el gestor hacia el agente, este mensaje permite la transferencia de información organizada en tablas; SNMPv1, en cambio, solo permite este tipo de transferencia fila a fila tras el respectivo intercambio de mensajes GetRequest/GetResponse.

Otra característica de SNMPv2 es el carácter «no atómico» de los mensajes Get. En SNMPv1, cuando se trata de obtener el valor de una o más variables y el agente no puede devolver al menos uno de ellos, el comando entero falla; en tal caso, el gestor deberá recurrir a la solicitud de nuevos comandos Get con menos variables. En cambio, a través del comando Get de SNMPv2 se permite obtener resultados parciales mediante la devolución, por parte del agente, de aquellos valores que pueda, e ignorando el resto del comando. Queda claro, por tanto, que se reduce el número de transacciones en la red frente al uso de SNMPv1.

Una última característica importante a reseñar acerca de SNMPv2 es la posibilidad de llevar a cabo una gestión de red descentralizada con objeto de abordar con mayores garantías el control de una red de gran tamaño y/o elevada carga de tráfico. En un esquema de gestión descentralizado se permite la existencia de varias estaciones de gestión de alto nivel, conocidas como *servidores de gestión*. Cada servidor gestionará una porción del número total de agentes, pudiendo un servidor dado delegar responsabilidades de gestión a gestores intermedios. Estos gestores intermedios realizarán las siguientes funciones:

- Monitorización y control de los agentes bajo su responsabilidad.
- Envío de información y generación de respuestas a mensajes de control procedentes de un servidor de gestión de nivel superior.

Para la comunicación gestor-gestor, SNMPv2 introduce dos características adicionales: un mensaje Inform, que permite el envío no solicitado de información entre gestores, y una MIB gestor-gestor.

Un paso más allá de las versiones SNMP 1 y 2 comentadas lo constituye la versión 3. SNMPv3 incluye la funcionalidad de SNMPv2 al tiempo que incorpora características de seguridad. SNMPv3 comprende los siguientes tres módulos:

- El módulo de procesamiento de mensajes y de control gestiona la creación y análisis de mensajes SNMP.
- El módulo de procesamiento local lleva a cabo el control de acceso a variables y el procesamiento de datos y de mensajes Trap.
- El módulo de seguridad proporciona autenticación y control de acceso, además de cifrado de la información transferida.

Para más detalles acerca de SNMPv3 se recomienda consultar la revisión que de este protocolo puede encontrarse en los RFC 2571-2576 y actualizaciones posteriores.

MIB y SMI

Un aspecto fundamental en la gestión de redes es el relativo, como ya se ha dicho, a las bases de datos (MIB) que definen el conjunto de objetos y variables susceptibles de ser gestionados. La independencia de diseño establecida entre estas y el protocolo SNMP permite, de forma fácil y flexible, la incorporación, borrado y modificación de objetos.

Cuando hablamos de MIB nos referimos de forma implícita a lo que se conoce como MIB-II (RFC 1212 y 1213, este último actualizado por los RFC 2011, 2012 y 2013), versión actualizada de las originales MIB definidas para SNMPv1. Los objetos MIB-II se agrupan en las siguientes diez categorías²:

- *Sistema* (1): clase correspondiente a objetos relacionados con el sistema operativo de *hosts* y nodos de encaminamiento que componen la red. Un ejemplo de variable de este tipo es *sysUpTime*, a través de la cual se indica el tiempo transcurrido desde que el sistema inició su funcionamiento.
- *Interfaces* (2): objetos y variables relacionados con las interfaces de red. Algunos ejemplos de variables de esta categoría son *ifMtu*, indicativa de la MTU asociada a una interfaz, e *ifNumber*, que indica el número de interfaces de red existentes en un dispositivo gestionado.
- *Traducción de direcciones* (3): grupo de objetos relativos a, por ejemplo, traducciones ARP.
- *IP* (4): conjunto de variables relativas al protocolo IP. Ejemplo de ellas son *ipRoutingTable*, correspondiente a una tabla de encaminamiento, *ipDefaultTTL*, variable que indica el valor usado en el campo TTL de los paquetes IP, o *ipFragOKs*, relativa al número de datagramas fragmentados correctamente.
- *ICMP* (5): variables relativas al protocolo ICMP, como por ejemplo *icmpInEchos*, a través de la que se controla el número de mensajes de eco ICMP recibidos.
- *TCP* (6): conjunto de objetos relativos al protocolo TCP. Un ejemplo de variable correspondiente a esta categoría es *tcpMaxConn*, la cual indica el número máximo de conexiones TCP permitidas.
- *UDP* (7): variables relativas al protocolo UDP. Un ejemplo es *udpInDatagrams*, número de datagramas UDP recibidos.
- *EGP* (8): variables relativas al protocolo EGP, como *egpAs*, que indica el sistema autónomo asociado a la entidad EGP.
- *Transmisión* (10): grupo relativo a distintos tipos de medios de transmisión.
- *SNMP* (11): variables que proporcionan información acerca de SNMP. Un ejemplo es la variable *snmpOutTraps*, que indica el número de mensajes Trap enviados por la entidad de protocolo SNMP.

Adicionalmente a las MIB, la especificación SMI establece un conjunto de reglas para definir e identificar variables MIB. Estas reglas siguen la norma ASN.1 («Abstract Syntax Notation One») de OSI, la cual es un lenguaje formal que permite definir nombres y tipos de variables mediante dos características principales: una notación que puede ser comprendida por el usuario humano y una compacta codificada empleada en los protocolos de comunicaciones.

Los nombres de las variables MIB se toman del *espacio de nombres de identificación de objetos* administrado por ISO e ITU, parte del cual se muestra en la Figura 12.4. Es el conocido como MIT («Management Information Tree»), cuya principal característica es su naturaleza absoluta, global, lo que significa que los nombres de las variables son únicos. Así, la variable *ipRouteMask* se referencia únicamente en el espacio como *iso.identified-organization.dod.internet.mgmt.mib-2.ip.ipRouteMask* o, atendiendo a su valor numérico codificado, como 1.3.6.1.2.1.4.11.

² En el caso de las MIB originales el número de grupos considerado era ocho.

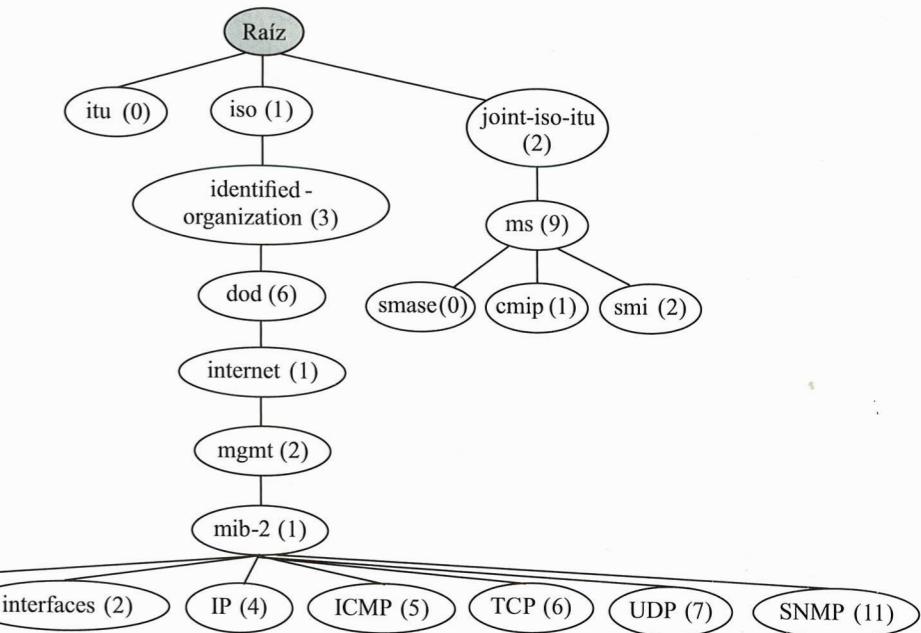


Figura 12.4. Parte del espacio de nombres ISO-ITU: MIT.

12.2.3. Gestión en redes OSI

Haciendo válida su aplicación a todo sistema de gestión independientemente de implementaciones concretas, los elementos, modelos y estructura funcional ya presentados al comienzo del Apartado 12.2 constituyen también la base de un sistema de gestión de redes OSI.

El modelo que sustenta la gestión en OSI se denomina CMIS/CMIP, siglas de «Common Management Information Service/Common Management Information Protocol», refiriéndose CMIS al conjunto de funciones que permiten la gestión de los elementos de red y CMIP al protocolo de comunicaciones entre los agentes y el gestor (véase Figura 12.5). Algunos de los documentos más relevantes relacionados con la gestión OSI son:

- ISO/IEC («International Electrotechnical Commission») 7498-4, relativo a la gestión en el modelo de referencia OSI (recomendación X.700 de ITU-T).
- ISO/IEC 9595 (X.710), donde se define CMIS.
- ISO/IEC 9596-1 (X.711), el cual define CMIP.
- ISO/IEC 10165 (X.72?), en donde se describe SMI.

Al margen de la existencia del modelo de gestión CMIS/CMIP como modelo independiente para redes OSI, cabe mencionar también la disposición de CMOT («CMIS over TCP/IP», RFC 1189), consistente en el uso de CMIS/CMIP sobre la familia de protocolos TCP/IP.

Arquitectura CMIS/CMIP

En la Figura 12.6 se muestra la arquitectura CMIS/CMIP, donde se explicita la intervención de los siguientes elementos:

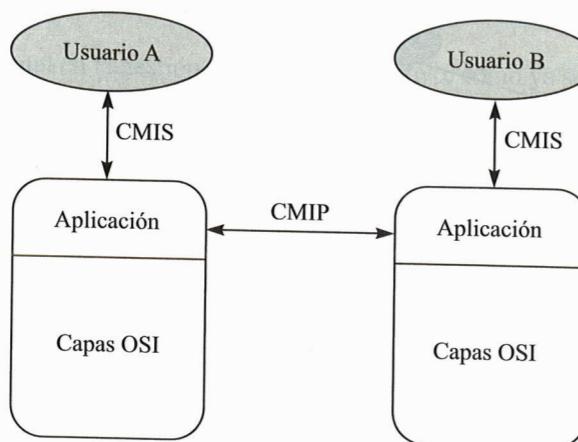


Figura 12.5. Modelo de gestión CMIS/CMIP.

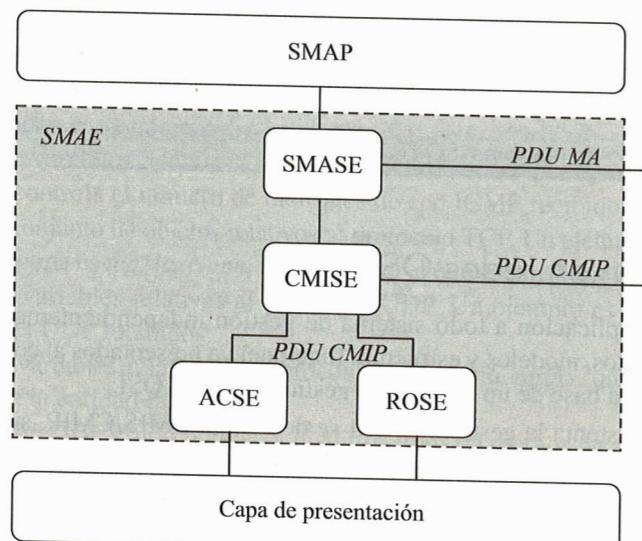


Figura 12.6. Arquitectura CMIS/CMIP.

- **SMAP** («Systems Management Application Process»), proceso de aplicación que proporciona la interfaz con la información MIB.
- **SMAE** («Systems Management Application Entity»), entidad que soporta la comunicación de los SMAP y usa CMIP para intercambiar información entre agentes. La SMAE está compuesta a su vez por:
 - **SMASE** («Systems Management Application Service Element»), elemento que interacciona directamente con la SMAP.
 - **CMISE** («Common Management Information Service Entity»), entidad que proporciona los servicios básicos de gestión para reportar eventos, manipular datos de gestión y generar requerimientos.

- **ROSE** («Remote Operations Service Element»), elemento que lleva a cabo la invocación de una operación en un sistema remoto de la siguiente manera: se solicita una operación remota, el agente la ejecuta y devuelve el resultado.

Los servicios proporcionados por ROSE son: RO-INVOKE, para llevar a cabo una petición; RO-RESULT, para devolver el resultado obtenido; RO-ERROR, para indicar error en la ejecución; y RO-REJECT, para rechazar una petición.

- **ACSE** («Association Control Service Element»), elemento que establece y libera asociaciones entre entidades de aplicación.

Los servicios de ACSE son: A-ASSOCIATE, para iniciar una asociación; A-RELEASE, para liberar una asociación; A-ABORT y A-P-ABORT, para una liberación anormal con posible pérdida de información.

Modelo de comunicaciones

Por lo que respecta a los modelos organizativo y de información, poco hay que reseñar diferente a lo ya apuntado en los apartados anteriores para la gestión de redes en general. Si acaso, hacer mención explícita a la rama conjunta ISO-ITU específica para CMIS/CMIP en el MIT (véase Figura 12.4). Frente a ellos, es el modelo de comunicaciones, sustentado en el uso del protocolo CMIP, lo que realmente diferencia este modelo de gestión de otros como SNMP.

La PDU CMIP presenta el formato indicado en la Figura 12.7(a). De entre sus campos constituyentes es de destacar el de *valor*, indicativo del tipo de PDU CMIP de que se trata; esto es, de la acción a realizar sobre el objeto cuya identidad se especifica en el campo *ID*. Los diferentes tipos de PDU contempladas son los indicados en la Figura 12.7(b), los cuales pueden ser categorizados dentro de dos servicios distintos proporcionados a través de CMIS:

- Servicio de notificación (M-EVENT-REPORT), utilizado para reportar eventos acerca de un objeto.
- Servicios de operación de gestión (M-GET, M-SET, etc.), referentes a acciones sobre la información de gestión.

Para concluir este punto, y de modo similar a como se indicó para SNMP en la Figura 12.2, en la Figura 12.8 se muestra la situación de CMIS/CMIP en la pila OSI, donde se indican los posibles mensajes intercambiados entre las entidades.

ID	Valor	Clase de objeto	Instancia de objeto	Información
----	-------	-----------------	---------------------	-------------

(a)

Servicio	Valor	Descripción
M-EVENT-REPORT	0/1	Envío de notificaciones
M-GET	3	Obtención de valores y atributos
M-SET	4/5	Modificación de atributos
M-ACTION	6/7	Inicio de acción
M-CREATE	8	Creación de objeto
M-DELETE	9	Borrado de objeto
M-CANCEL-GET	10	Cancelación de comando M-GET previo

(b)

Figura 12.7. Formato de PDU CMIP (a) y tipos (b).

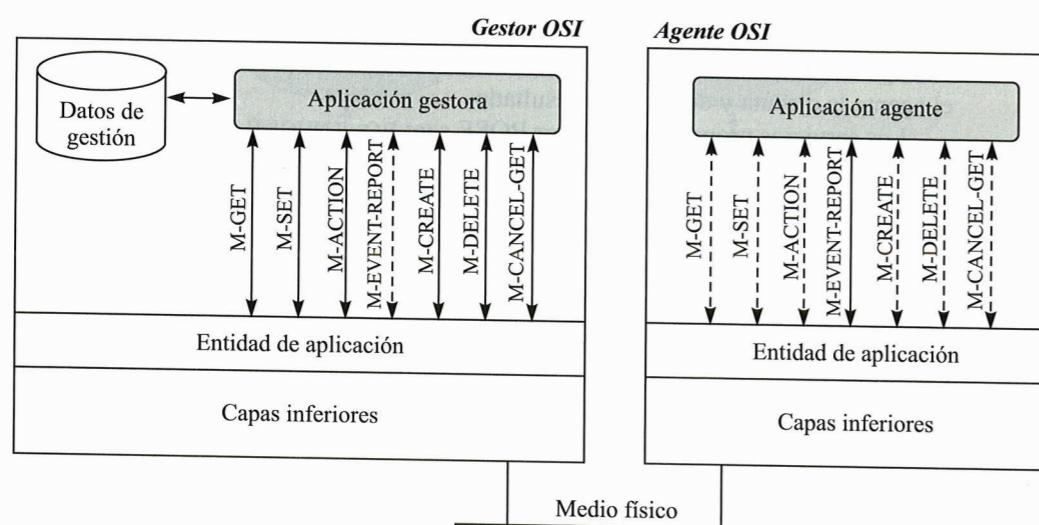


Figura 12.8. Situación de CMIS/CMIP en la pila OSI.

12.3. Seguridad en redes de computadores

El campo de la seguridad en redes y comunicaciones ha adquirido en los últimos años una relevancia primordial, motivado principalmente por la dependencia actual de nuestra sociedad de las TIC. Y ello en aspectos tan diversos como la economía, la cultura o el ocio. Para el individuo, el acceso a los servicios de Internet tales como la web o el correo electrónico se ha convertido en una actividad cotidiana, desde la simple búsqueda de información hasta la realización de compras y transacciones bancarias. El uso de la tecnología resulta igualmente relevante para las empresas y organizaciones de todo tipo, con servicios que van desde el uso del correo electrónico para las comunicaciones internas y externas hasta el soporte de transacciones *online* básicas para su actividad diaria, sea esta comercial o no.

A modo de ejemplo de la relevancia actual de la seguridad en sistemas y comunicaciones, sírvase indicar la evolución del número de incidentes de seguridad en este tipo de entornos reportados por el CERT («Computer Emergency Response Team»; <http://www.cert.org>) entre 1999 y 2008 (Figura 12.9). Estos incidentes se traducen en numerosas ocasiones en actividades de fraude económico que suponen pérdidas de varias decenas de miles de millones de euros anuales para las empresas y particulares en países como EE.UU. y Reino Unido, por no mencionar los daños en equipos y sistemas. Por lo que respecta a España, en la Figura 12.10 se muestra el porcentaje de usuarios de Internet que han sufrido algún intento de fraude en 2010, según un estudio llevado a cabo por INTECO (<http://www.inteco.es>).

En suma, la preocupación por el tema, hay que decir que justificada, es tal que cualquier empresa u organización que desee prevenir y minimizar en la medida de lo posible potenciales ataques, debe invertir gran cantidad de recursos humanos y económicos en una gestión y supervisión adecuadas de sus sistemas y comunicaciones.

12.3.1. Fundamentos de seguridad

Antes de abordar en profundidad los numerosos aspectos involucrados en la seguridad de redes y sistemas, en lo que sigue se proporcionan fundamentos y terminología básica al respecto.

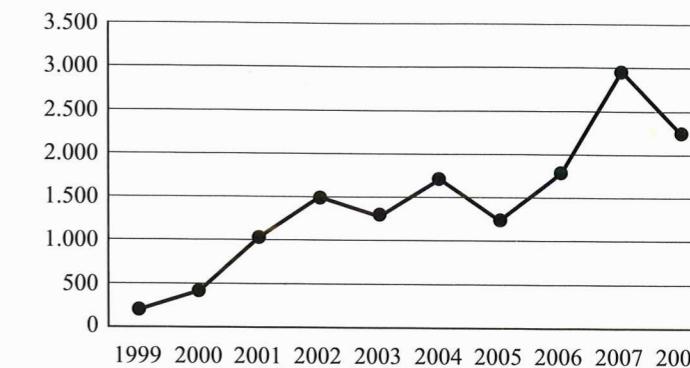


Figura 12.9. Número de incidentes de seguridad entre 1999 y 2008, según el CERT.

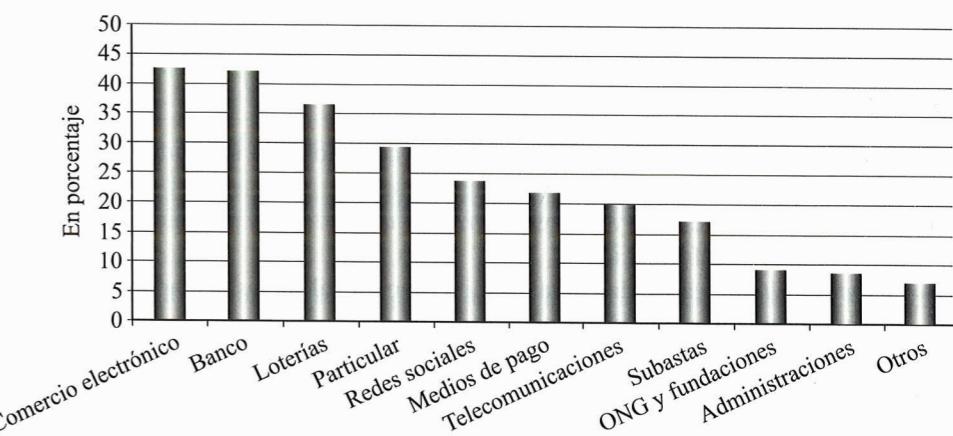


Figura 12.10. Porcentaje de usuarios que han sufrido algún intento de fraude, por ámbitos (fuente: INTECO, 2010).

Conceptos básicos

En el contexto anteriormente establecido, la seguridad de la información puede definirse como:

La protección de los activos frente a amenazas, entendiendo por amenaza la capacidad potencial para hacer un mal uso de los activos protegidos. Todas las categorías de amenazas deben ser consideradas, aunque en el dominio de la seguridad se le presta más atención a aquellas relacionadas con actividades de carácter malicioso, humanas o de cualquier otro tipo.

La terminología aquí utilizada y alguna otra relacionada es la siguiente (véase recomendación X.800 de la ITU-T y RFC 4949 del IETF):

- **Activo:** elemento que forma parte de un sistema, incluyendo hardware, software y personas.
- **Amenaza:** circunstancia que puede ocasionar pérdida o daño. Las amenazas pueden ser accidentales (p.e., fuego, fallo de energía, error del operador), medioambientales (p.e., inundaciones, tormentas eléctricas) o deliberadas (p.e., software malicioso, escuchas secretas).
- **Vulnerabilidad:** posibilidad de que una amenaza se haga realidad, a través de la explotación de un sistema o de alguno/s de sus elementos.

- **Riesgo:** amenazas y vulnerabilidades relacionadas con un activo particular. Un riesgo puede tratarse reduciendo la vulnerabilidad de nuestros activos ante amenazas importantes.
- **Impacto:** resultado de la ocurrencia de una violación de la seguridad. Representa el efecto de un fallo en la preservación de alguno de los aspectos de la seguridad y puede medirse en términos de revelación, denegación, destrucción o modificación de datos y elementos del sistema.
- **Consecuencia:** resultado adverso de un impacto, es decir, daño derivado de un incidente ocurrido en un sistema (p.e., pérdidas financieras).

Otro concepto importante en este marco es el de *ataque*, entendido este como la instanciación de una amenaza, esto es, una amenaza hecha realidad. Son diversas las clasificaciones propuestas para categorizar los ataques de acuerdo a distintos criterios, tales como la relación de los atacantes con la organización/sistema atacado o su nivel de conocimientos. Es habitual así hablar de atacantes (y ataques) *pasivos* frente a *activos*, en función del procedimiento ejecutado para llevar a cabo el acceso a la información. En el primer caso, el atacante o intruso se limita a escuchar o monitorizar la información, no alterando esta. Por su parte, un atacante es activo si interviene de forma efectiva en el sistema objetivo para conseguir el acceso a la información.

Los así denominados servicios de seguridad, relacionados con los diferentes aspectos de la misma, son los siguientes:

- **Confidencialidad:** asociado al concepto más cercano de la seguridad, este se refiere a la prevención de accesos no autorizados a la información.
- **Integridad de los datos (y de los programas del sistema):** prevención de la modificación no autorizada de la información.
- **Responsabilidad:** capacidad para asumir como «legales» las actuaciones llevadas a cabo en el sistema. Entre ellas cabe destacar:
 - **Autenticación:** garantía de que las partes implicadas en una comunicación/actuación son quienes dicen ser.
 - **No repudio:** garantía de que las partes han participado activamente en una actuación/comunicación dada.
 - **Control de accesos:** garantía de identidad para el acceso autorizado a la información y recursos de un sistema.
- **Disponibilidad:** necesidad de que los datos y sistemas estén accesibles y usables (por usuarios autorizados) siempre que se requiera.

Los servicios de confidencialidad, integridad y autenticación constituyen el conocido como *modelo CIA*, si bien este se ha completado, como se ha evidenciado antes, con la inclusión de otros aspectos claves en el robustecimiento de la seguridad en redes y sistemas.

En relación a estos servicios, es posible establecer cuatro tipos generales de ataques (Figura 12.11):

- **Interrupción:** el objetivo del ataque es la fuente de información o canal de comunicación, de manera que el activo del sistema queda inutilizado (Figura 12.11(b)). En otras palabras, este ataque es contra la disponibilidad del sistema pues impide el uso normal de los recursos.
- **Intercepción:** en este caso el atacante consigue acceso no autorizado a la información (Figura 12.11(c)). Se trata, pues, de un ataque contra la confidencialidad.
- **Modificación:** el atacante o intruso no solo consigue el acceso a la información, sino que además la modifica (Figura 12.11(d)). En este caso se trata de un ataque contra la integridad.
- **Fabricación:** una parte no autorizada inserta información en la comunicación, provocando así un ataque contra la autenticación (Figura 12.11(e)). Cuando la información insertada ha sido

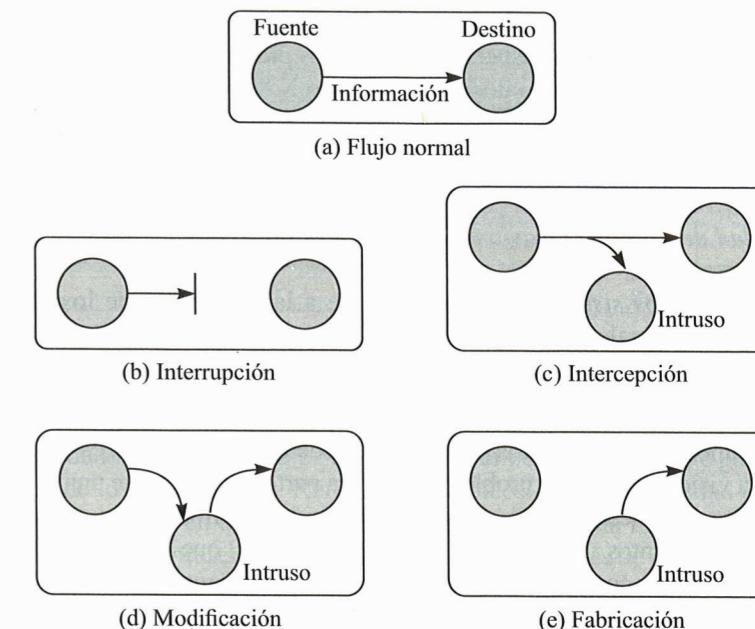


Figura 12.11. Tipos de ataques según el proceso general realizado.

previamente capturada, se habla de ataques de repetición. Además, se usa el término «suplantación» cuando el atacante trata de hacerse pasar por un usuario legítimo.

Entendiendo por *defensa* de un sistema un conjunto de mecanismos, procedimientos y herramientas que permiten proteger aquel frente a ataques, en la Figura 12.12 se indican las líneas de defensa típicas consideradas:

- **Prevención:** conjunto de acciones adoptadas por el equipo administrador de un sistema para evitar la aparición de ataques.
- **Detección:** dada la imposibilidad de garantizar que las medidas preventivas eviten totalmente la ocurrencia de ataques, se hace precisa la implantación de mecanismos que, en base a la monitorización del entorno, puedan determinar el potencial desarrollo de ataques.
- **Respuesta:** detectada una acción atacante, el sistema debe reaccionar de manera que se activen procedimientos que den solución a aquellas. Así, es aconsejable que dichos procedimientos actúen sobre la propia configuración del entorno para reforzar su seguridad.

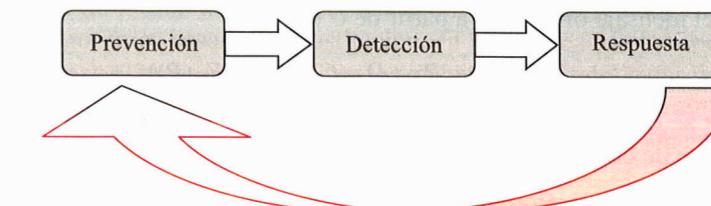


Figura 12.12. Líneas de defensa en un sistema.

En los apartados siguientes se presentan distintos esquemas cuyo fin es proporcionar seguridad en un sistema en uno o más de los aspectos comentados previamente. Son los conocidos como mecanismos de seguridad y se definen en documentos como X.800; entre ellos cabe destacar el cifrado, la firma digital, la autenticación, la integridad de los datos y el control de acceso.

Conviene señalar en este punto que dentro del campo «seguridad en redes» pueden establecerse al menos tres aspectos bien diferenciados:

- Seguridad de las comunicaciones*, relativo a la protección de las transmisiones propiamente dichas.
- Seguridad de los sistemas*, correspondiente a la protección de los elementos/dispositivos (intermedios y finales) que intervienen y hacen posibles las comunicaciones.
- Seguridad de la información*, es decir, protección de los datos en sí mismos, transmitidos o almacenados.

Si bien cada uno de estos aspectos podría tratarse de forma separada, es evidente que ello proporcionaría solo una visión parcial del problema. Por otra parte, es obvio que una red de computadores no es solo un conjunto de procedimientos para la transmisión de información, sino que aquella no puede entenderse sin los elementos y sistemas (hardware y software) que la conforman. En suma, la seguridad en redes debe abordar conjuntamente los distintos aspectos antes mencionados. Así se evidenciará en lo que sigue, dedicándose el resto del presente Apartado 12.3.1 a dar respuesta a la cuestión c), el Apartado 12.3.2 a la a) y los Apartados 12.3.3 y 12.3.4 a la b). Tras todo ello, en el Apartado 12.3.5 se hará referencia a aspectos legales y éticos de la seguridad, aspectos estos en los que conviene formar también mínimamente a los profesionales de la seguridad.

Cifrado de la información: clave secreta y clave pública

Ya en tiempos de Julio César, en su guerra contra las Galias, este utilizó sistemas de cifrado de información para el envío de mensajes «privados». Desde entonces, estos sistemas han evolucionado enormemente gracias, de forma especial, a la actividad militar desarrollada en el campo. Los esquemas de cifrado pretenden preservar la privacidad, esto es, la confidencialidad, para lo cual se realizan los siguientes procesos en una hipotética comunicación emisor-receptor en la que el objetivo es el intercambio de un mensaje *P*, al que suele referirse como *texto llano* o *plano*, del inglés «plaintext» (Figura 12.13):

- El emisor transforma el mensaje original *P* en uno *C* a través del empleo de un algoritmo de cifrado *E()*, en el que se hace uso de un número especial al que se conoce como *clave de cifrado* («key» en inglés) y se referencia con la letra *K*. El mensaje resultante *C* se llama *texto cifrado* («ciphertext») y, de acuerdo con lo anterior, se obtiene como:

$$C = E_K(P)$$

C será el mensaje finalmente enviado hacia el receptor.

- Recibido *C* en el otro extremo, el proceso de descifrado llevado a cabo, *D()*, será tal que se recupera el mensaje original *P* a partir de *C* según:

$$P = D_K(C) = D_K(E_K(P))$$

De lo anterior se deben extraer las siguientes conclusiones respecto de las características deseables de un algoritmo de cifrado/descifrado:

- Las funciones *E()* y *D()* deben ser de bajo coste computacional y fácilmente implementables.
- Es evidente que debe cumplirse que $E_K(P) \neq E_K(P')$ y $D_K(C) \neq D_K(C')$, $\forall P \neq P'$ y $C \neq C'$, de modo que el proceso de cifrado/descifrado resultante sea único.

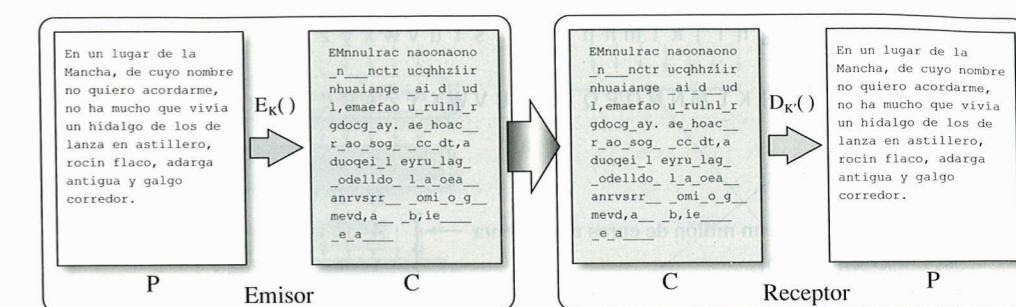


Figura 12.13. Proceso general de cifrado y descifrado de mensajes.

- El algoritmo de cifrado/descifrado utilizado debe ser lo suficientemente robusto como para que, dado *C*, sea imposible obtener *P* sin conocer *D_K()*. En este punto, cabe mencionar que uno de los principios de Kerckhoffs establece que la efectividad de un sistema de cifrado no debe recaer en que su diseño sea secreto, sino en la robustez de la/s clave/s utilizada/s. Teniendo ello presente, se consideran cuatro esquemas de *criptoanálisis* o rotura de cifrado:
 - Texto cifrado: en este caso se persigue la rotura del proceso de cifrado a partir de la única disposición de un mensaje *C*.
 - Texto plano conocido: frente al caso anterior, para la rotura del esquema de cifrado utilizado ahora se cuenta con un mensaje *C* y su *P* asociado. Es obvio que cualquier sistema de cifrado es más vulnerable a este tipo de ataque que al anterior.
 - Texto llano seleccionado: un paso más en la robustez de los sistemas de cifrado consiste en su invulnerabilidad ante ataques en los que el atacante dispone no solo de un *C* y su *P* asociado, sino que es capaz incluso de cifrar por sí mismo algunos mensajes arbitrarios.
 - Diferencial: este esquema de criptoanálisis se fundamenta en la observación de pares de texto cifrado cuyos textos planos correspondientes tienen ciertas diferencias entre sí. Se estudia la evolución de estas diferencias mientras los textos llanos se cifran con la misma clave, empleándose las diferencias en los textos cifrados resultantes para asignar probabilidades a las distintas claves posibles. A medida que se van analizando más y más pares de texto cifrado, una clave surgirá como la más probable; esa será la clave correcta.

Existen dos tipos básicos de técnicas de cifrado: sustitución y transposición, o permutación. El primer tipo de esquemas se caracteriza por el hecho de que el proceso de cifrado llevado a cabo consiste en la simple sustitución de un carácter o conjunto de estos por otro/s. Un ejemplo sencillo de este tipo de algoritmos lo constituye el denominado cifrado del César, llamado así por haber sido utilizado por este emperador romano y consistente en una simple rotación del alfabeto en la que se hace corresponder, por ejemplo, el carácter A con D, B con E, C con F, D con G, y así sucesivamente (véase Figura 12.14(a)). El criptoanálisis de este tipo de técnicas es relativamente sencillo, bastando en la mayoría de los casos con llevar a cabo un análisis estadístico lingüístico-gramatical del mensaje cifrado.

Frente a los algoritmos basados en sustitución, en las técnicas de transposición o permutación los caracteres que forman el mensaje cifrado son los mismos que los del mensaje llano, pero colocados en una posición distinta de la original. En la Figura 12.14(b) se muestra un ejemplo en el que la secuencia de caracteres que forma *P* se ordena para su transmisión alfabéticamente en columnas según una matriz creada a partir de una clave ASCII.

Las técnicas de cifrado actuales consisten en una combinación más o menos compleja de módulos de sustitución y de permutación (Figura 12.14(c)), pudiendo clasificarse en dos grandes grupos: de clave secreta y de clave pública. A continuación se describen ambos tipos de esquemas.

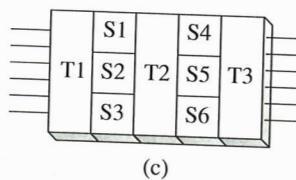
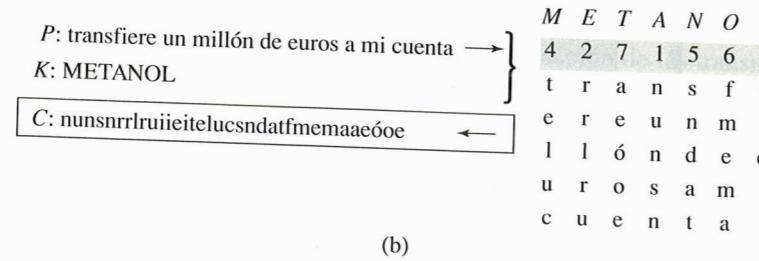


Figura 12.14. Ejemplo de esquemas de cifrado de sustitución (a) y de transposición (b). Técnica basada en la combinación de varios módulos o cajas de sustitución (S) y de transposición (T) —(c)—.

Los algoritmos de **cifrado de clave secreta** se caracterizan por el hecho de que la clave utilizada en el proceso es solo conocida, compartida, por los extremos de la comunicación. Generalmente, este tipo de técnicas se denominan *simétricas* por cuanto que la clave de cifrado y descifrado son la misma, diferenciándose ambos procesos por el algoritmo en sí utilizado en cada extremo. Un ejemplo de este caso es la técnica DES («Data Encryption Standard»). Desarrollada por IBM en 1975, en la Figura 12.15(a) se esquematiza el proceso de cifrado seguido por este algoritmo sobre bloques de 64 bits del mensaje plano:

1. Se realiza una transposición inicial del bloque de bits de entrada P .
 2. A continuación, y de forma sucesiva, tienen lugar 16 iteraciones en las cuales entra en juego la clave privada de cifrado K , de 56 bits de longitud. Para cada iteración i se procede como sigue (Figura 12.15(b)):
 - a) Los 32 bits de la derecha pasan a ser los de la izquierda para la iteración siguiente; es decir: $R_{i-1} \rightarrow L_i$.
 - b) Por su parte, los 32 bits de la derecha se obtienen según $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, donde \oplus representa la función XOR, K_i la clave de cifrado adaptada para la iteración i y $f()$ una función que opera dicha clave y los 32 bits de la derecha del bloque a cifrar correspondientes a la iteración i .
- Respecto de este punto hemos de comentar la función $f()$ y la forma en que se obtiene K_i a partir de K . Por lo que se refiere a la clave, inicialmente se lleva a cabo una transposición $K' = T(K)$, generándose después para cada iteración una clave K'_i distinta según el siguiente proceso:
- Dividida la clave K' en dos partes de 28 bits cada una, cada bloque se rota de forma independiente a la izquierda de acuerdo con el número de iteración: $rL_i(K'_{1-28})$ y $rL_i(K'_{29-56})$.
 - Al bloque así obtenido se le realiza una transposición de 56 bits: $K'_i = T[rL_i(K'_{1-28}) rL_i(K'_{29-56})]$.

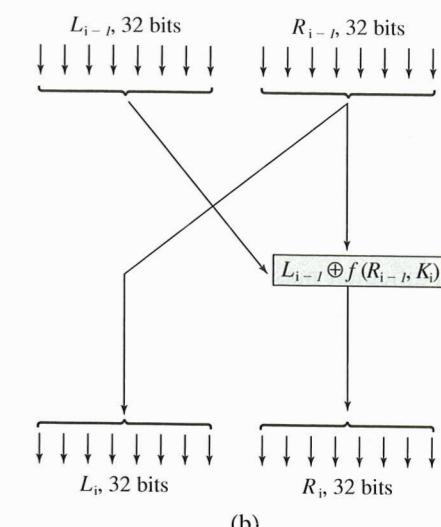
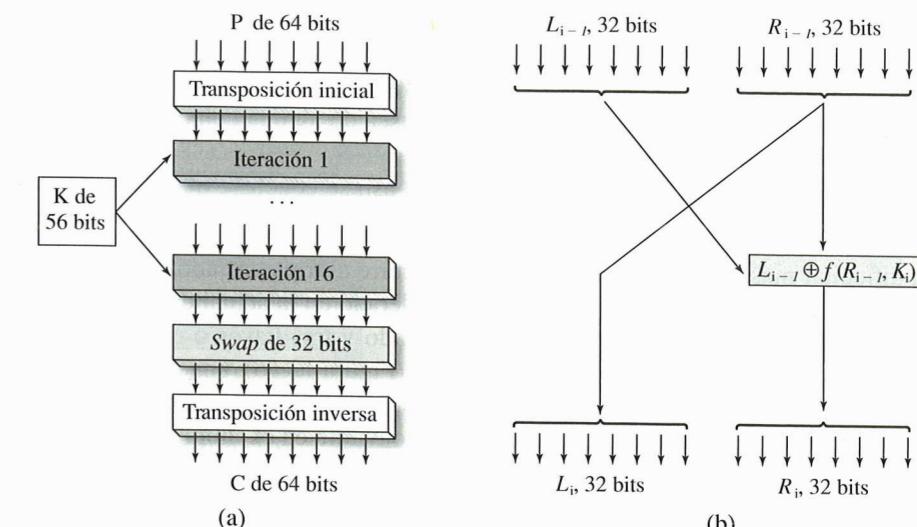


Figura 12.15. Esquema de cifrado DES: proceso general (a) y detalle de una de las 16 iteraciones en él desarrolladas (b).

- Finalmente, se extrae de los 56 bits un subconjunto de 48 bits diferente para cada iteración. Estos 48 bits son, además, permutados: $K_i = T(K'_i, 48$ bits).

Por su parte, los procesos involucrados en la función $f()$ son los siguientes:

- Sobre los bits R_{i-1} se hace una transposición y, sobre esta, una duplicación con objeto de conseguir 48 bits: $E = dup[T(R_{i-1})]$.
- Este bloque de 48 bits se opera con K_i mediante una función XOR: $E' = E \oplus K_i$.
- Los 48 bits resultantes se introducen, en grupos de 6, en 8 módulos de sustitución: $S_1(E'_{1-6}), S_2(E'_{7-12}), \dots, S_8(E'_{43-48})$, siendo la salida de cada módulo S de 4 bits de longitud.
- Finalmente, sobre los 32 bits resultantes (8 bloques de 4 bits) se hace una transposición.
- 3. Tras las 16 iteraciones comentadas, se realiza un intercambio (*swap*) de los 32 bits de orden más alto por los 32 de orden inferior.
- 4. El proceso de cifrado concluye con la transposición inversa a la efectuada en el paso 1. El bloque de 64 bits de salida obtenido corresponderá al texto cifrado que venimos denotando como C .

Según se deduce de lo explicado, es claro que la dificultad de romper DES será 2^{56} . Es decir, dado que la longitud de la clave es 56 bits, habría que probar un máximo de 2^{56} claves distintas para poder decodificar satisfactoriamente cualquier mensaje. Dada la alta potencia de cómputo con que cuentan los sistemas actuales, la tarea de descifrar DES resulta relativamente fácil. Además, la posibilidad de distribuir rangos de claves entre varios ordenadores en lo que se conoce como «grid computing» (al estilo, por ejemplo, del análisis llevado a cabo en Internet en el proyecto SETI de búsqueda de vida extraterrestre inteligente) hace de DES un esquema vulnerable. Una forma sencilla de aumentar la complejidad de DES consiste en efectuar un cifrado doble, $C = E_{K2}(E_{K1}(P))$, de modo que, al utilizarse dos claves, la complejidad del proceso crece hasta 2^{112} . Desgraciadamente, Merkle y Hellman desarrollaron en 1981 la técnica de criptoanálisis denominada *encuentro a la mitad*, en la que se siguen los pasos especificados a continuación:

1. Se parte del conocimiento de una secuencia de parejas (P_i, C_i).
2. Se calcula $R_i = E_{K_i}(P_1)$ para los 2^{56} posibles valores de K_i .
3. Se calcula $S_j = D_{K_j}(C_1)$ para los 2^{56} posibles valores de K_j .
4. Se comparan ambas listas de resultados en busca de alguna coincidencia. Tal caso se corresponde, en potencia, con $K_i = K_1$ y $K_j = K_2$.
5. Se comprueba si $E_{K_j}(E_{K_i}(P_n)) = C_n, \forall n$. Si es así, se concluye que $K_i = K_1$ y $K_j = K_2$; si no, se vuelve al paso 3 en busca de nuevas parejas K_i, K_j .

Para dificultar más el criptoanálisis de DES se recurre al triple cifrado, lo que equivale a la consideración de una clave de 168 bits de longitud. Considerado en la actualidad como «suficientemente» seguro, este proceso no implica tres módulos de cifrado y tres llaves o claves, sino dos bloques de cifrado y uno de descifrado, con un total de dos claves (Figura 12.16): una para los módulos de cifrado, bloques primero y tercero, y otra para el de descifrado, bloque intermedio del proceso. De esta forma, sin más que hacer $K_1 = K_2$ el *triple DES* (3DES) se convierte en el simple.

El principal problema que plantea 3DES, sin embargo, es su lentitud de operación, además de trabajar solo sobre bloques de 64 bits. Principalmente motivado por ello, un algoritmo de cifrado simétrico más avanzado y utilizado actualmente es AES («Advanced Encryption Standard»), el cual fue aprobado por el NIST («National Institute of Standards and Technology») en el FIPS PUB 197 (<http://www.nist.gov>) en 2001 y que trabaja sobre bloques de datos de 128 bits y acepta claves de longitud 128, 192 y 256 bits.

De forma análoga que para DES, en la Figura 12.17 se muestra el proceso general de cifrado y descifrado seguido por AES para 128 bits, que consta de 10 rondas. Este es sucesivamente como sigue:

1. En primer lugar, es de significar que el mensaje a cifrar/descifrar se estructura en forma de matriz, donde cada fila corresponde a un bloque de 128 bits.
2. La clave se expande en 44 palabras de 32 bits: $w[0,43]$, de manera que cuatro distintas de ellas (128 bits) son la clave de entrada para cada una de las rondas.
3. En cada una de las rondas se consideran hasta cuatro etapas adicionales, una de permutación y tres de sustitución:
 - a) Substituir bytes: haciendo uso de una tabla, se lleva a cabo una sustitución del bloque byte a byte.
 - b) Desplazar filas: se realiza una permutación sencilla de los 128 bits correspondientes a la fila.
 - c) Mezclar columnas: sustitución de cada byte en la columna en función del resto de bytes en la misma.
 - d) Sumar clave: operación XOR entre el bloque resultante con la porción correspondiente de la clave para la ronda.

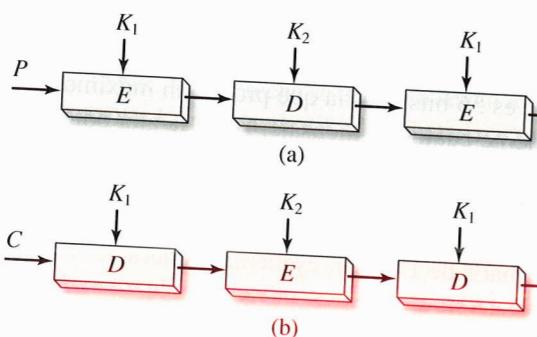


Figura 12.16. Triple DES (3DES): cifrado (a) y descifrado (b).

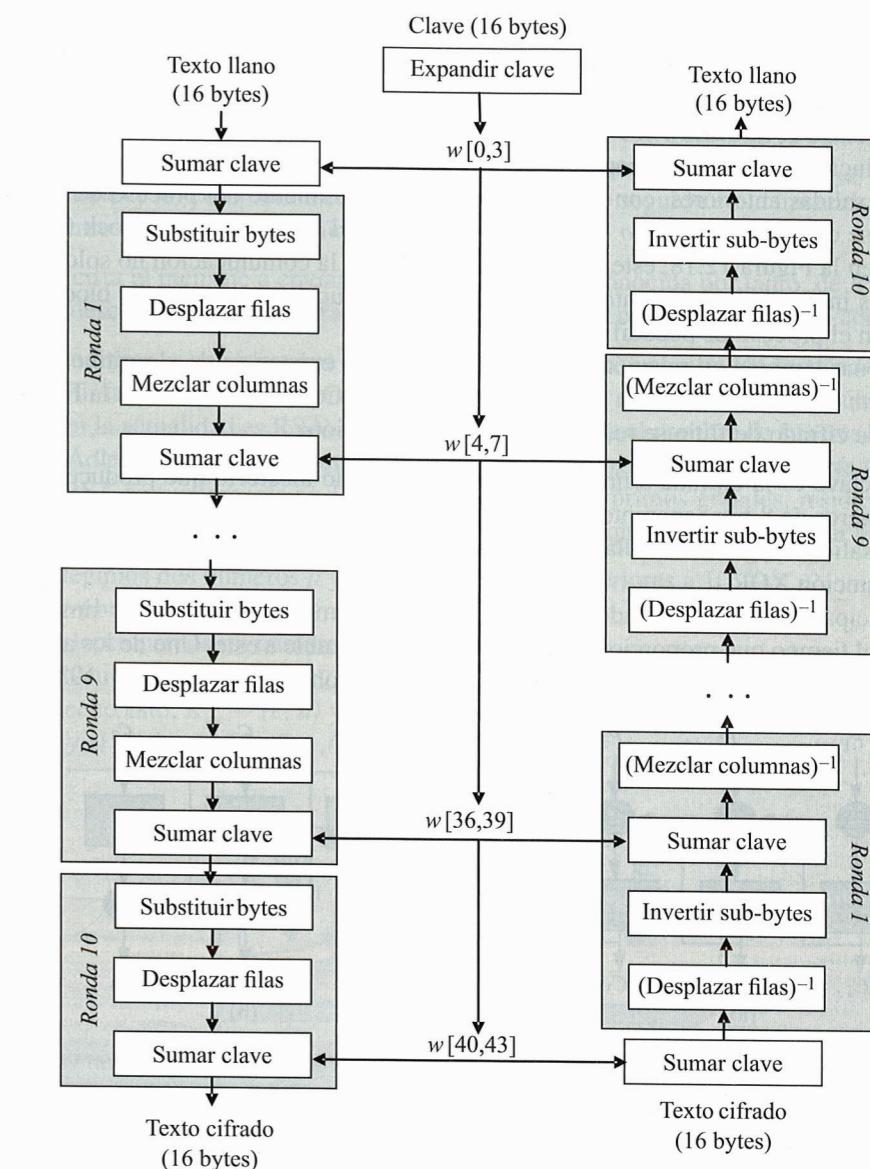


Figura 12.17. Proceso general de cifrado y descifrado AES.

4. Tanto el cifrado como el descifrado comienzan por una etapa de suma de clave, seguida de 9 rondas de cuatro etapas y una décima de tres etapas.
5. Solo se hace uso de la clave en la etapa de suma de clave, siendo las cuatro etapas específicamente reversibles.
6. Como se observa en la figura mencionada, el proceso de descifrado no es exactamente el mismo que el de cifrado.
7. La última ronda del proceso (tanto en cifrado como descifrado) consta de solo tres etapas, lo cual es exigible para hacer reversible el cifrado.

Tanto DES como AES se corresponden básicamente con un esquema de sustitución monoalfabética, en el que, dado un bloque de X bits de texto llano de entrada, se genera un bloque cifrado también de longitud X . Además, la salida será siempre la misma para la misma entrada (suponiendo, claro está, que la clave K no varía).

Una solución que permite proporcionar una salida dependiente no solo de la entrada actual, sino también de salidas anteriores, con el consiguiente robustecimiento del proceso de cifrado, es la operación que se conoce como *modo encadenado*, o CBC (del inglés «Cipher Block Chaining»). Según se observa en la Figura 12.18, este esquema proporciona a la comunicación no solo confidencialidad, sino además integridad, por cuanto que la potencial sustitución de uno o más bloques cifrados sería detectada en el proceso de decodificación posterior.

Mencionar también en relación al cifrado simétrico la existencia de algoritmos *de flujo*, frente a los *de bloque* comentados hasta la presente (DES y AES). Como se muestra en la Figura 12.19, en un algoritmo de cifrado de flujo se realiza la siguiente operación:

- La clave es la entrada a un generador de bits seudo-aleatorio que produce una secuencia de números de 8 bits (aparentemente) aleatorios.
- La salida del generador, llamada *keystream*, se opera byte a byte con el texto plano mediante la función XOR.

La principal ventaja del cifrado de flujo es que es de implementación más simple que el cifrado de bloque, al tiempo que proporciona una seguridad comparable a este. Uno de los algoritmos de flujo más conocidos y usados en la actualidad es RC4 («Rivest Cipher 4»), diseñado en 1987 por Ron Rivest.

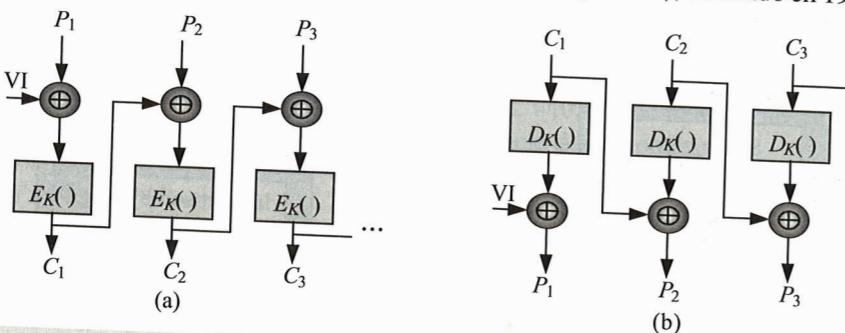


Figura 12.18. Modo de operación CBC o encadenado: cifrado (a) y descifrado (b). (VI es un vector de inicialización del proceso).

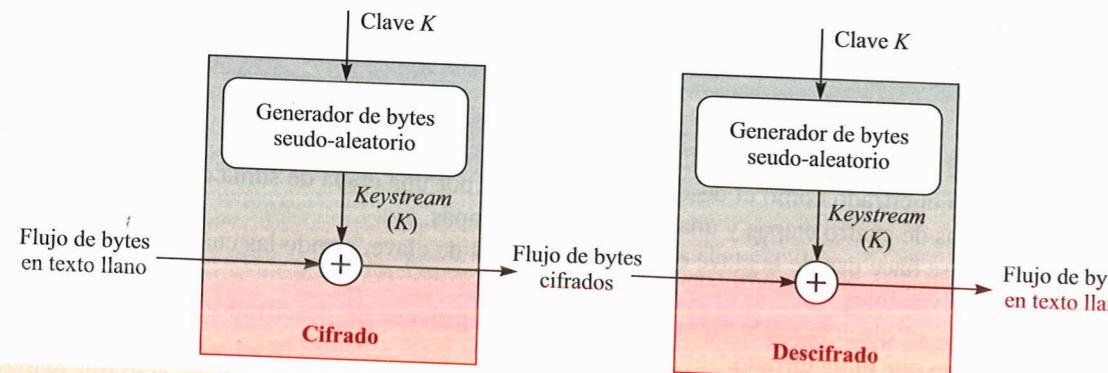


Figura 12.19. Diagrama de cifrado de flujo.

El principal inconveniente que presentan los algoritmos de cifrado de clave secreta radica en el hecho de que esta debe ser, en principio, distinta para cada pareja emisor-receptor, debiendo mantenerse oculta a terceros. En 1976, Diffie y Hellman, de la Universidad de Stanford, propusieron las técnicas de *cifrado de clave pública*. Estas, frente a las de clave secreta, se caracterizan por la consideración de dos claves por usuario: una pública conocida por todo el mundo, K_{pu} , y una privada conocida solo por él y no compartida con nadie, K_{pr} . El proceso seguido en la transmisión-recepción de un mensaje entre A y B sería el siguiente (Figura 12.20(a)):

1. A cifra el mensaje a enviar con la clave pública, conocida por tanto, de B : $C = E_{K_{puB}}(P)$.
2. B descifra el mensaje cifrado recibido con su clave privada, que solo él conoce: $P = D_{K_{prB}}(C)$.

De esta forma se garantiza que solo B pueda recibir mensajes cifrados destinados a él, sin necesidad de compartir una clave secreta con cada posible emisor. El algoritmo de clave pública más conocido y utilizado en la actualidad es RSA, llamado así en referencia al nombre de sus tres diseñadores: Rivest, Shamir y Adleman, investigadores del Instituto de Tecnología de Massachusetts (MIT). Desarrollado en 1978, RSA se fundamenta en la teoría de los números primos grandes, residiendo la dificultad de su criptoanálisis en la elevada complejidad computacional que implica factorizar este tipo de números:

1. Elegimos dos números p y q primos grandes (superiores a 10^{100}).
2. Se obtienen $n = p \times q$ y $z = (p - 1) \times (q - 1)$.
3. Seleccionamos un número d primo respecto de z .
4. Calculamos e tal que $e \times d = 1 \bmod z$.
5. Hecho esto, $K_{pu} = (e, n)$ y $K_{pr} = (d, n)$, de modo que el proceso de cifrado será $C = P^e \bmod n$ y el de descifrado $P = C^d \bmod n$.

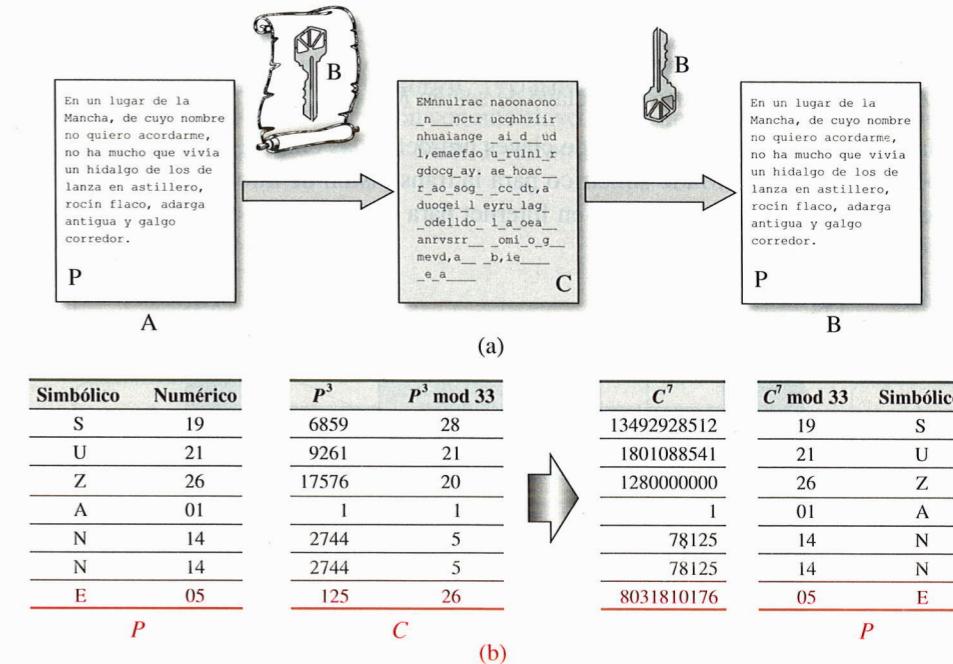


Figura 12.20. Esquema conceptual de las técnicas de cifrado de clave pública (a) y ejemplo de transmisión siguiendo el esquema RSA con $K_{pu} = (3,33)$ y $K_{pr} = (7,33)$ (b).

- Un ejemplo que aclara conceptualmente los pasos anteriores es el siguiente (Figura 12.20(b)):
1. Supongamos que elegimos los números primos $p = 3$ y $q = 11$.
 2. Entonces, $n = 33$ y $z = 20$.
 3. Elegimos $d = 7$ dado que 7 y 20 no tienen factores comunes.
 4. $7 \times e = 1 \bmod 20$, por lo que $e = 3$.
 5. Por lo tanto: $K_{pu} = (3,33)$ y $K_{pr} = (7,33)$, llevándose a cabo los procesos de cifrado y descifrado como se muestra en la Figura 12.20(b).

Según se deduce de lo expuesto, las técnicas de cifrado de clave pública son intrínsecamente *asimétricas*, puesto que las claves de cifrado y descifrado son diferentes.

Es importante señalar que los algoritmos de cifrado de clave pública resultan más costosos computacionalmente que los de clave secreta, por lo que su implementación se evita en determinadas aplicaciones, entre otras, en las que los dispositivos de trabajo presentan limitaciones de cómputo, almacenamiento y/o batería. Sin embargo, frente a los esquemas de clave pública, como ya es sabido, el principal problema que plantea un esquema de cifrado de clave secreta es la necesidad de compartir varias claves origen-destino. La técnica de *intercambio de claves de Diffie-Hellman* evita esta necesidad mediante el establecimiento dinámico de la clave de sesión. Como se muestra en la Figura 12.21, el proceso seguido para ello entre un origen A y un destino B es simple:

1. A, origen de la comunicación, selecciona dos números primos grandes n y g con ciertas características; por ejemplo, que $(n - 1)/2$ sea también primo. Ambos números son públicos, por lo que A puede comunicárselos abiertamente a B.
2. Además de n y g , A escoge un número grande, x , que mantendrá en secreto. La información al respecto enviada a B es el número $g^x \bmod n$.
3. Recibidos estos números en B, este elegirá un número grande y y hará $(g^x \bmod n)^y = g^{xy} \bmod n$. Además, enviará hacia A el número $g^y \bmod n$.
4. Análogamente a como procedió B, A opera el número recibido en la forma $(g^y \bmod n)^x = g^{xy} \bmod n$.
5. Al final, A y B compartirán la clave de sesión $K_{AB} = K_{BA} = g^{xy} \bmod n$.

Dada la no necesidad de disponer de claves pre-definidas entre el emisor y el receptor, al tiempo que se permite el uso de cifrado simétrico para la transmisión de información posterior, el algoritmo de Diffie-Hellman es de uso frecuente en Internet para el establecimiento de claves de sesión (y posterior uso de cifrado simétrico).

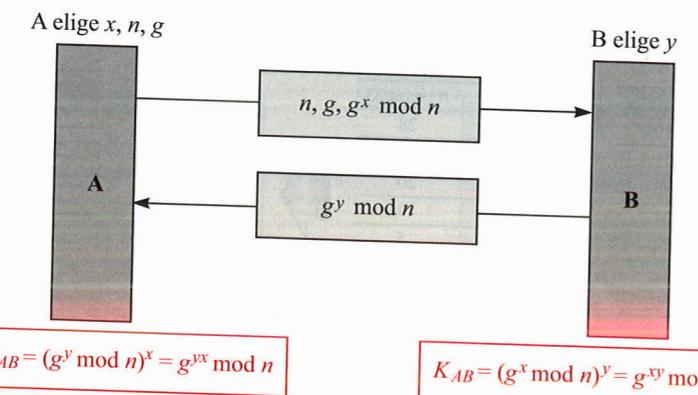


Figura 12.21. Procedimiento de establecimiento de clave de sesión de Diffie-Hellman.

Autenticación de mensajes e integridad

Uno de los servicios de seguridad más ampliamente utilizados es el de integridad, esto es, la garantía de que una cierta información no ha sido modificada por terceros. En caso de serlo, será detectado por el receptor. Unido a este servicio de seguridad en la literatura se encuentra el de autenticación de mensajes, el cual se refiere a la constatación de que el mensaje ha sido generado por el origen adecuado.

La forma más habitual de proporcionar ambos servicios de seguridad mencionados es mediante los conocidos como MAC («Message Authentication Code»), los cuales se suelen expresar como una función del mensaje M y la clave entre el emisor y el receptor K_{AB} : $MAC_M = F(K_{AB}, M)$. El procedimiento seguido en la obtención de un código MAC es el mostrado en la Figura 12.22:

- a) Dado un mensaje M y una clave K , un algoritmo MAC permite derivar un código identificativo de M .
- b) Este código se adjunta al mensaje original, de manera que:
 - Un hipotético receptor de dicha información calcula el código MAC del mensaje recibido.
 - Si dicho código coincide con el adjunto a M , se determinará la integridad del mensaje. En otro caso, habremos de concluir que este no es auténtico.

La función $F()$ puede referirse a algoritmos de cifrado bien conocidos (y ya presentados con anterioridad) tales como AES, RC4, RSA, etc. En este caso, los códigos MAC se denominan CMAC («Cipher-based MAC»). Sin embargo, suele ser más habitual el empleo de *funciones compendio, resumen o hash*, cuyas características principales son las siguientes:

- Son de cálculo sencillo.
- Proporcionan un mensaje de salida de longitud fija, independientemente de la longitud del mensaje de entrada.
- Dados dos mensajes de entrada distintos, los resúmenes correspondientes son también distintos.
- Desde el punto de vista de la seguridad, resulta imposible para un atacante obtener un mensaje a partir de su compendio. Esto es, son funciones de un solo sentido («one-way functions»).

Así, en el RFC 2104 se describe HMAC («Hash-based MAC»), un esquema MAC basado en el empleo de funciones *hash* en conjunción con una clave. Entre las funciones o algoritmos *hash* más utilizados hemos de destacar dos: MD5 y SHA-1. A continuación se describen ambos.

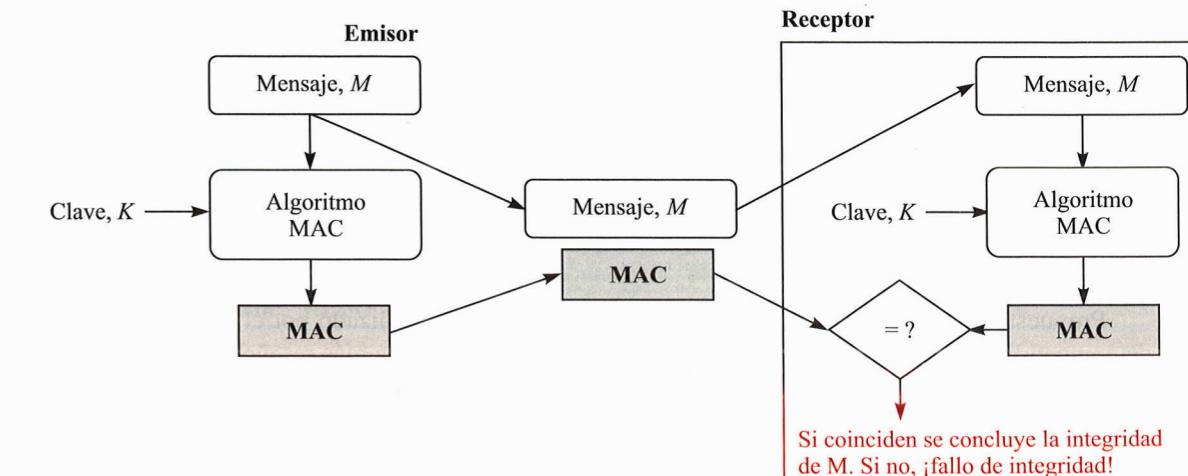


Figura 12.22. Códigos de autenticación del mensaje.

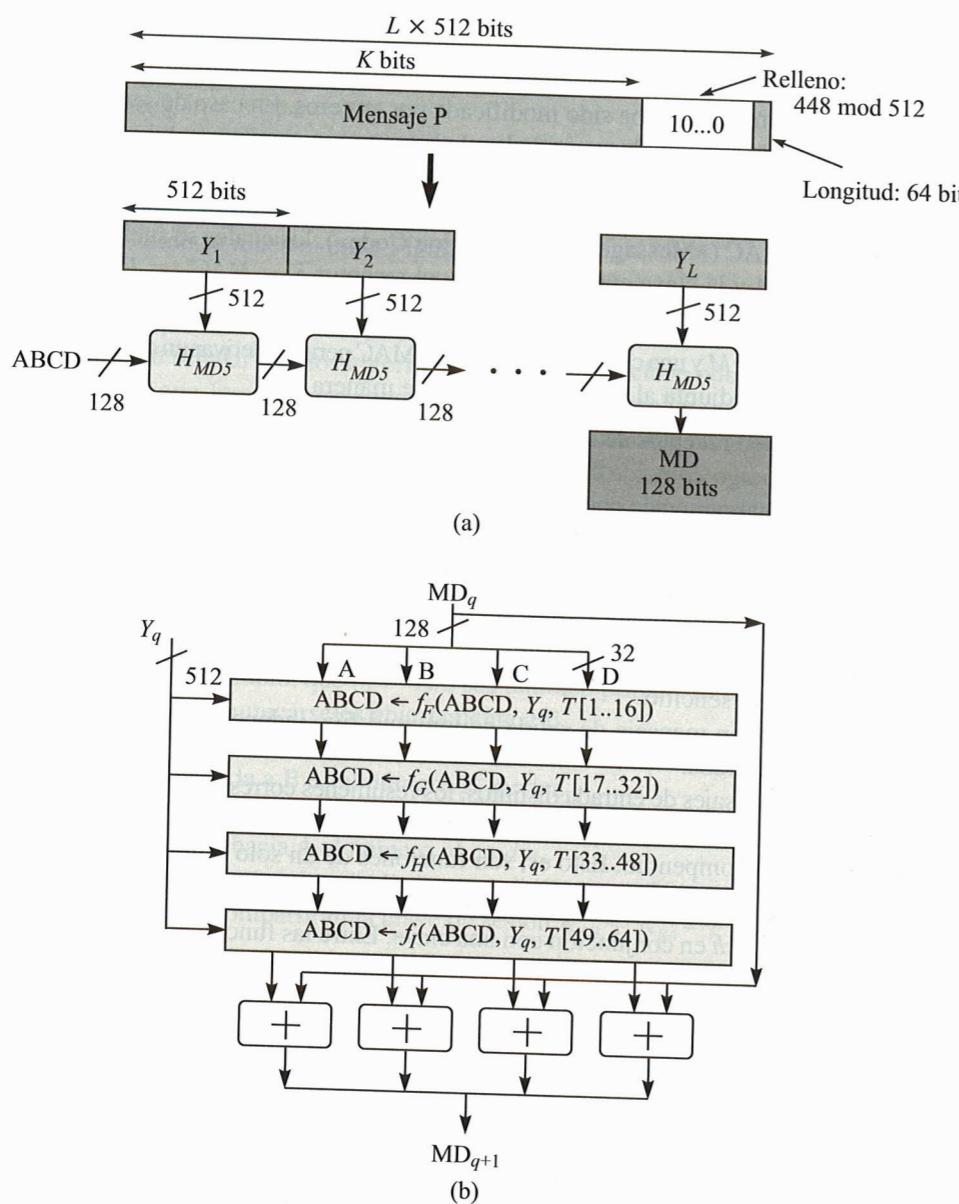


Figura 12.23. Procesamiento general MD5 de un mensaje P (a) y detalle del proceso para un bloque q (b).

Propuesto por Rivest en 1992 y descrito en el RFC 1321 (actualizado por el 6151), el algoritmo de compendio *Message Digest 5* (MD5) presenta como característica más reseñable el hecho de que, procesado el mensaje de entrada en bloques sucesivos de 512 bits, el resumen obtenido tiene una longitud fija de 128 bits. Como se indica en la Figura 12.23(a), las operaciones realizadas sobre un mensaje P a procesar mediante MD5 son las siguientes:

1. P se rellena por la derecha con una secuencia de bits de la forma 10...0 y de longitud máxima igual a 448 bits.

2. Junto al relleno anterior, se añade un campo de 64 bits de longitud donde se especifica el tamaño del mensaje, incluido el relleno y el propio campo de longitud. Tras ambos rellenos, la longitud total del mensaje final a procesar, P' , debe ser múltiplo de 512 bits.
 3. P' se divide en bloques de longitud fija de 512 bits: Y_1, \dots, Y_L .
 4. Se inicializa el valor de cuatro registros A, B, C y D, cada uno de longitud 32 bits, a los valores hexadecimales:
- A = 01234567
B = 89ABCDEF
C = FEDCBA98
D = 76543210
5. El proceso MD5 se lleva a cabo secuencialmente, bloque a bloque, a lo largo del mensaje P' . Dicho proceso opera sobre dos entradas: el bloque Y_q actual, de 512 bits, y los registros A, B, C y D, de una longitud total igual a 128 bits. El resultado del proceso se almacena en dichos registros y sirve como entrada al procesamiento del siguiente bloque, Y_{q+1} . Las operaciones realizadas sobre cada bloque son las esquematizadas en la Figura 12.23(b) y sus características más reseñables son las siguientes:

- a) Se lleva a cabo un procesamiento en cuatro rondas sucesivas, en las que se utilizan, respectivamente, las funciones f_F, f_G, f_H y f_I , definidas como:

$$\begin{aligned}f_F(X, Y, Z) &= (X \& Y) \mid (!X \& Z) \\f_G(X, Y, Z) &= (X \& Z) \mid (Y \& !Z) \\f_H(X, Y, Z) &= X \oplus Y \oplus Z \\f_I(X, Y, Z) &= Y \oplus (X \mid !Z)\end{aligned}$$

siendo $\&$, \mid , $!$ y \oplus los operadores binarios AND, OR, NOT y XOR, respectivamente.

- b) Los argumentos de las funciones mencionadas son:

$$X = ABCD$$

$$Y = Y_q$$

Z = $T[i \dots i + 15]$, con $i = 1, 17, 33, 49$ y $T[i] = \text{int}(2^{32} \times \text{abs}(\text{sen } i))$, estando « i » especificado en radianes.

- c) Cada función realiza 16 operaciones en bloques de 32 bits ($32 \times 16 = 512$) del tipo (para más detalles consultar RFC):

$$A = B + ((A + f(B, C, D) + Y_{q1\dots16} + T[1\dots64]) \ll shift)$$

- d) Completadas las cuatro rondas, se realiza una operación suma módulo 2^{32} .

- e) La salida constituye la entrada para el procesamiento del siguiente bloque. Si $q = L$, es decir, si este era el último bloque de P' , el resultado corresponderá al compendio del mensaje: $MD5(P) = MD_{L+1}$.

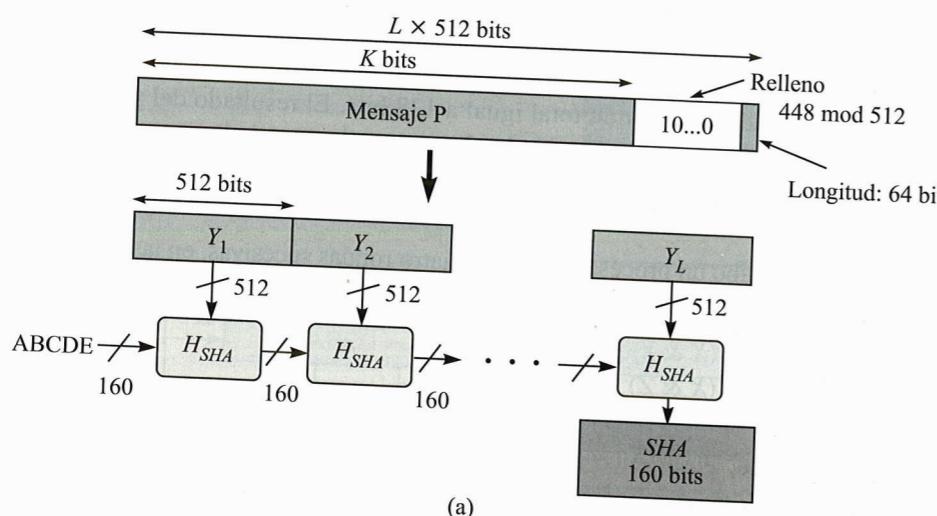
Una función resumen alternativa a MD5, y más robusta que este, es SHA-1 («Secure Hash Algorithm 1»), propuesto por el NIST en el FIPS 180-1 y especificado también en el RFC 3174. La forma en que se procesa el mensaje del que se desea obtener el compendio es similar a la descrita para MD5, con dos salvedades principales (Figura 12.24(a)):

1. La longitud del resumen obtenido es 160 bits en lugar de los 128 resultantes mediante MD5.

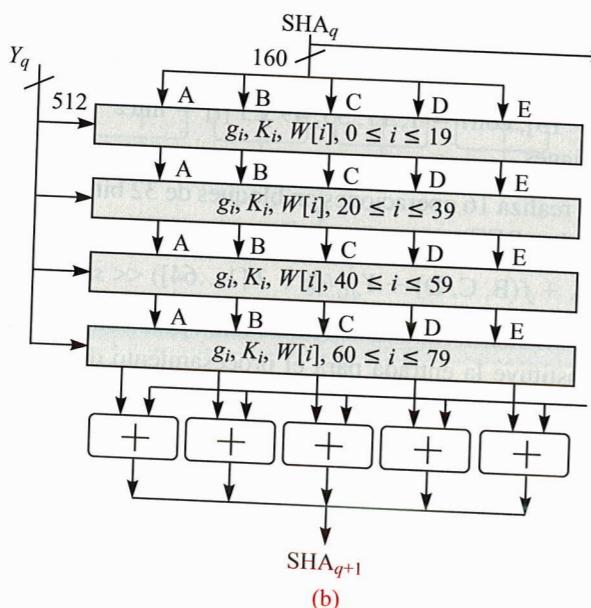
2. El proceso SHA toma dos entradas: el bloque Y_q de 512 bits en cuestión y cinco registros, A, B, C, D y E, de 32 bits cada uno, inicializados a los valores hexadecimales.

A = 67452301
 B = EFCDAB89
 C = 98BADCFE
 D = 10325476
 E = C3D2E1F0

en lugar de los cuatro usados en MD5.



(a)



(b)

Figura 12.24. Procesamiento general SHA-1 de un mensaje P (a) y detalle del proceso para un bloque q (b).

Las operaciones realizadas por el módulo SHA-1 sobre cada bloque Y_q de 512 bits son las siguientes (Figura 12.24(b)):

- Se divide el bloque en 16 palabras de 32 bits: $w[0], \dots, w[15]$.
- Se extienden estas 16 palabras a un total de 80 en base a combinaciones de las originales: $W[0], \dots, W[79]$.
- Tomadas las 80 palabras en cuatro grupos de 20 ($0 \leq i \leq 19, 20 \leq i \leq 39, 40 \leq i \leq 59, 60 \leq i \leq 79$), iteramos un total de 80 rondas de manera que, en cada una de ellas:

$$\text{ABCDE}_i \leftarrow f(g_i(\text{ABCDE}_{i-1}), K_i, W[i])$$

donde ABCDE_{i-1} toma los valores iniciales ya indicados en el punto 2 anterior, $g_i(\text{ABCDE})$ se refiere a operaciones AND, OR, NOT y XOR y, además:

$$\begin{aligned} K_i &= 5A827999, \text{ si } 0 \leq i \leq 19 \\ &6ED9EBA1, \text{ si } 20 \leq i \leq 39 \\ &8F1BBCDC, \text{ si } 40 \leq i \leq 59 \\ &CA62C1D6, \text{ si } 60 \leq i \leq 79 \end{aligned}$$

- Completadas las cuatro rondas, se realiza una operación suma módulo 2^{32} .
- La salida será la entrada para el procesamiento del siguiente bloque. Si $q = L$, lo que corresponde al último bloque, el resultado será el compendio del mensaje: $\text{SHA}(P) = \text{SHA}_{L+1}$.

Para más detalles acerca de SHA-1 consultese el sitio web <http://csrc.nist.gov>. En él podemos encontrar descritos también los algoritmos SHA-256, SHA-384 y SHA-512, los cuales, procesado el mensaje de entrada en bloques de 512 bits, en el caso de SHA-256, y de 1.024 bits, para SHA-384 y SHA-512, proporcionan un resumen de 256, 384 y 512 bits de longitud, respectivamente.

Autenticación del usuario y no repudio: firma digital

El empleo de técnicas de cifrado para proteger la información no tiene sentido si la comunicación se realiza con un usuario no autorizado. Es por ello necesario arbitrar técnicas que permitan validar la identidad de un interlocutor dado. Es lo que se conoce como *autenticación* («authentication» en inglés), proceso que puede llevarse a cabo a través de las técnicas de cifrado previamente estudiadas.

Al margen de la posible autenticación MAC vista con anterioridad, un esquema de autenticación simple basado en las técnicas de cifrado de clave secreta es el denominado de *reto-respuesta* («challenge-response»). En él, como se muestra en la Figura 12.25, dada una pareja emisor-receptor, A-B, y la clave entre ellos compartida, K_{AB} , los pasos seguidos en la identificación de ambos extremos son:

- A contacta con B indicándole su deseo de comunicación.
- B responde a esta solicitud con el envío de un reto, R_B , consistente en un número aleatorio grande que A debe devolver cifrado.
- Para ello A utilizará la clave compartida con B, K_{AB} .
- Por su parte, A hará lo propio para identificar a B; es decir, enviará a este un reto, R_A , que deberá devolver cifrado con K_{AB} .
- Si ambos extremos determinan que el cifrado realizado por el otro es correcto, se da por válida la autenticación y por iniciada la comunicación.

Un ataque simple que puede plantearse en el esquema de autenticación de reto-respuesta es el conocido como *ataque de repetición* («reply» en inglés). Como se deduce de su nombre, a través de este se devolvería al otro extremo el mismo reto que este ha enviado con anterioridad para que sea él



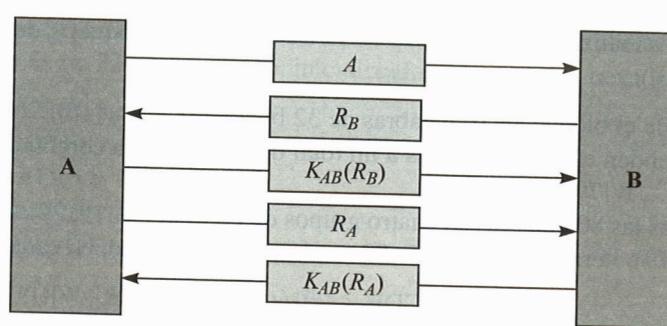


Figura 12.25. Autenticación mediante cifrado de clave secreta: esquema de reto-respuesta.

mismo quien proporcione la solución. Es decir, el ataque de repetición consistiría en hacer $R_A = R_B$. Por supuesto, este tipo de ataque es fácil de resolver; basta con seguir uno o más de los siguientes criterios: (a) el emisor (A en el caso comentado) debe identificarse antes de que lo tenga que hacer el receptor (B en el caso anterior); (b) utilización de dos claves de comprobación distintas, K_{AB} y K_{BA} ; (c) uso de dos conjuntos diferentes de retos, por ejemplo, números pares en un sentido e impares en el otro.

De forma parecida a como se procede con las técnicas de cifrado de clave secreta, los esquemas de clave pública también pueden utilizarse para implementar un servicio de autenticación. Pensemos en un usuario A que desea enviar un mensaje a otro B. La privacidad del mensaje queda garantizada, según vimos, sin más que A cierre la información con la clave pública de B; solo este, a través de su clave privada, podrá acceder al mensaje original.

Como se muestra en la Figura 12.26, para proporcionar autenticación además de privacidad, bastará con que A, previamente al cifrado con la clave pública de B que garantiza la confidencialidad en el canal, cierre el mensaje haciendo uso de su propia clave privada. De esta forma, B, tras recibir el mensaje cifrado y descifrarlo con su clave privada, comprobará que el emisor es A al descifrar en una segunda etapa con la clave pública de este.

Es evidente, por tanto, que la autenticación mutua puede llevarse a cabo, por ejemplo, sin más que proceder al cifrado, mediante la clave privada propia, de un reto especificado por el otro extremo, el cual, al recibir este mensaje, se limitará a descifrarlo haciendo uso de la clave pública del emisor.

En las transacciones comerciales electrónicas interesa poder demostrar de forma fehaciente que un usuario dado ha participado en una cierta comunicación. Esto permitiría resolver, por ejemplo, posibles demandas legales. Pensemos una hipotética situación en la que un usuario, tras ordenar a través de Internet un reintegro de su banco, niega haber sido él el ordenante de la operación y que, en consecuencia, demandará a la entidad si esta no repone en su cuenta la cantidad extraída. Para evitar esta y otras

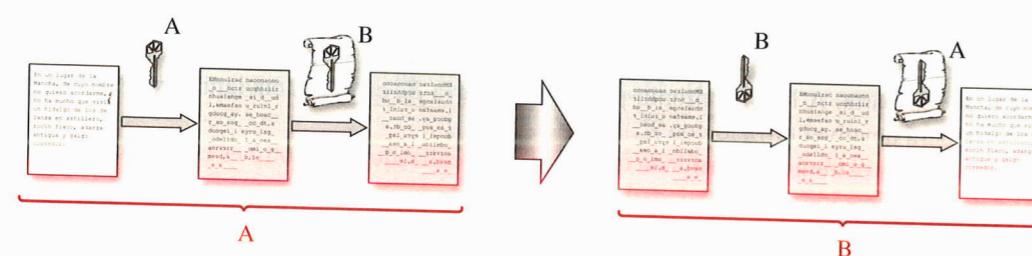


Figura 12.26. Autenticación (y confidencialidad) mediante cifrado de clave pública.

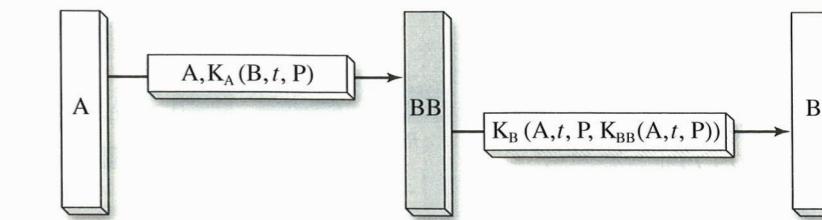


Figura 12.27. Firma digital mediante cifrado de clave secreta con Big Brother.

situaciones similares, se precisa un esquema que permita «firmar» un mensaje enviado de modo que, no pudiendo haber sido generado por nadie más, el emisor no pueda repudiar el mensaje transferido.

Por analogía con las firmas manuscritas, este tipo de mecanismos de no repudio se conocen como *firma digital*. Varios son los esquemas de firma digital ideados y propuestos en la bibliografía. Al igual que sucede con el proceso de autenticación, el no repudio puede proporcionarse a través del empleo de las técnicas de cifrado, tanto de clave secreta como de clave pública. Por lo que respecta a las primeras, se suele recurrir a una entidad central, referida en la literatura como *Big Brother* (BB), de acuerdo al siguiente procedimiento (Figura 12.27):

1. Supuesto que A desea contactar con B, el primero envía su identidad a BB. Además, cifrado con la clave que con él comparte, K_A , le comunica la identidad de B y el mensaje a transmitir a este, P . Adicionalmente puede incluirse una marca de tiempo a fin de identificar únicamente el mensaje y así prevenir ataques de repetición (véase más adelante en *gestión de claves*).
2. Recibida esta solicitud de envío, BB transmitirá a B, cifrado todo con la clave secreta compartida entre ambos, K_B , la identidad de A, el mensaje P , un sello de tiempo y la firma digital de BB. La firma digital consiste en los tres primeros elementos antes mencionados cifrados con una clave privada particular de BB, K_{BB} , de modo que solo él puede descifrar la información.

Aceptada globalmente la fiabilidad de BB, la firma digital es la prueba que puede presentar B en una hipotética demanda judicial por parte de A. Así, demostrado el mensaje transferido, la fecha y la identidad del emisor, el caso quedará legalmente zanjado.

Por su parte, la utilización de las técnicas de cifrado de clave pública para la generación de firmas digitales sigue el mismo proceso que el descrito para proporcionar autenticación (Figura 12.26). El simple descifrado del mensaje recibido con la clave pública del emisor basta para demostrar que aquel solamente pudo ser generado por este. En la Figura 12.28 se muestra el conjunto de procesos seguidos en una comunicación entre un emisor A y un receptor B en la que se intercambian un mensaje P mediante un esquema de firma digital:

— Los pasos seguidos por el emisor para el envío del mensaje son:

1. Obtención del resumen del mensaje: $R = \text{hash}(P)$.
2. Cifrado mediante RSA del resumen con la clave privada de A: $FD = RSA_{KprA}(R)$; es lo que constituye la firma digital en sí.
3. El mensaje cifrado finalmente transmitido hacia B, C, consistirá en la firma digital, FD , más el mensaje, P , todo cifrado con la clave pública de B para garantizar la confidencialidad del mensaje. O sea, $C = RSA_{KpuB}(FD + P)$.

— Recibido C en el receptor, este procederá como sigue:

1. Obtención de la firma digital, FD , además del mensaje original, P , sin más que descifrar C , utilizando la técnica RSA, con la clave privada de B: $FD + P = RSA_{KprB}(C)$.

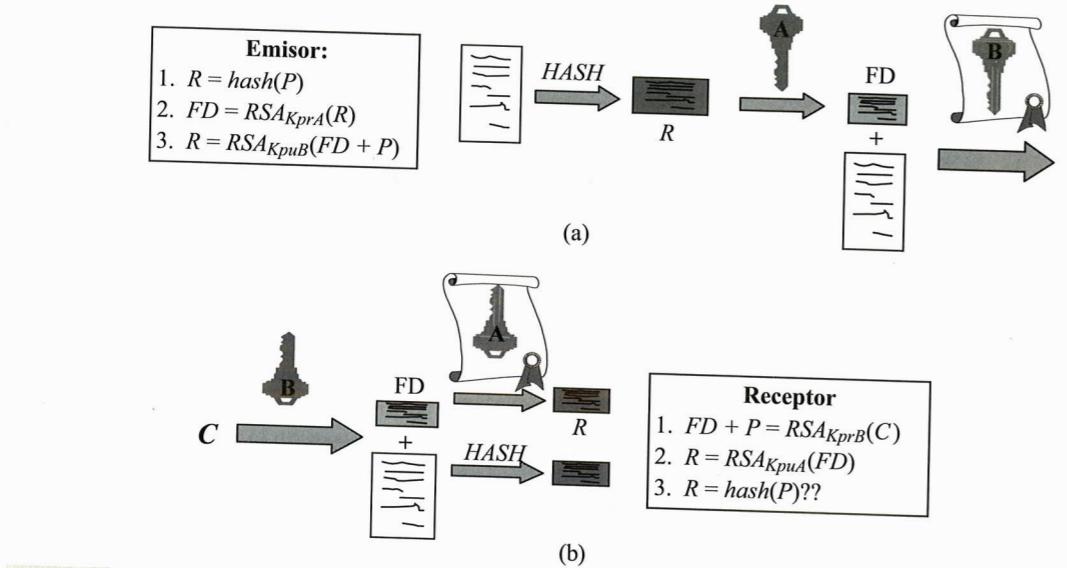


Figura 12.28. Procesos de emisión (a) y recepción (b) en los que se utiliza firma digital basada en funciones compendio.

2. Obtención del resumen de P tras descifrar, mediante RSA, la firma con la clave pública de A: $R = RSA_{KpuA}(FD)$.
3. Dado que P es conocido tras el paso 1, se calcula a este el compendio y se compara con el recibido. Si, como es esperable, ambos coinciden, la transmisión se da por válida.

Tal como se ha descrito el proceso, a través de la utilización de esquemas de firma digital basados en funciones hash se cubre no solo el no repudio, sino también los siguientes otros aspectos relativos a la seguridad:

- Confidencialidad, a través de la utilización de la técnica de cifrado RSA con la clave pública del receptor.
- Integridad, por cuanto que el resumen o compendio es único para cada mensaje.
- Autenticación, al utilizar la clave privada del emisor para cifrar, mediante RSA, el mensaje transmitido.

Gestión de claves

Un aspecto crucial en todos los sistemas de cifrado, y por ende en los procedimientos para la provisión de confidencialidad, integridad y autenticación basados en aquellos, es el referido al establecimiento y gestión de las claves involucradas. En esta línea, ya ha sido comentado previamente el algoritmo de Diffie-Hellman para el establecimiento dinámico de claves de sesión. Aunque, como ya se apuntó, este procedimiento resulta de alto interés, no permite per se el no repudio. Para ello existen otras técnicas que se comentan a continuación.

La primera se refiere al uso de un *centro de distribución de claves* (KDC, «Key Distribution Center») para las comunicaciones. El KDC es una entidad tercera, reconocida globalmente y distinta de las partes comunicantes, con la que cada usuario mantiene una clave privada y a través de la cual se inicia toda comunicación. En la Figura 12.29 se muestra el esquema de la *rana de boca ancha* basado

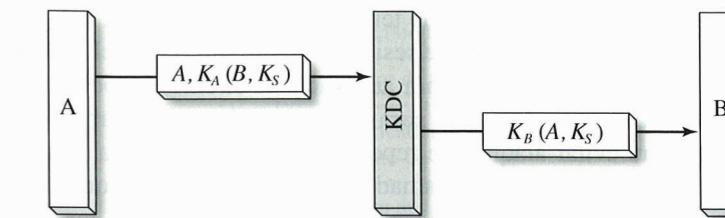


Figura 12.29. Protocolo de autenticación de rana de boca ancha en el que se hace uso de un KDC.

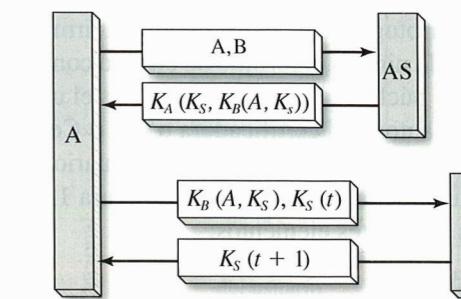


Figura 12.30. Protocolo de autenticación Kerberos 5.

en el uso de un KDC. En él, si un origen A desea establecer una comunicación con un receptor B, seguirá los siguientes pasos:

1. A se identifica al KDC a través del cifrado de los datos transmitidos mediante la clave secreta con él compartida, K_A . Los datos enviados corresponden a la identidad del receptor deseado, B, y la clave de sesión a utilizar con él a lo largo de la comunicación.
2. Verificada la identidad de A a través del descifrado de la información recibida mediante K_A , el KDC envía a B, cifrados con la clave secreta con él compartida, K_B , la identidad del origen que solicita la comunicación y la clave de sesión a utilizar.
3. Tras los pasos anteriores, A y B están seguros de la identidad de la otra parte y comparten una clave secreta de sesión, la cual podrá desarrollarse a partir de este momento.

A pesar de todos los esquemas previos, un ataque de repetición permitiría, aun sin conocer exactamente su contenido, el envío de un mensaje cifrado observado con anterioridad. Esta duplicación de mensajes previos puede solucionarse sin más que introducir en ellos (como ya se ha señalado anteriormente) un número que lo identifique únicamente en el tiempo. Dicho número puede consistir, por ejemplo, en un sello de tiempo o un valor monótonamente creciente; es lo que se conoce en el argot como *nonce*. Algunos esquemas que utilizan *nonces* son el de Needham-Schroeder, el de Otway-Rees y Kerberos. Este último, ampliamente utilizado por su fiabilidad, fue desarrollado en el MIT y las especificaciones en su versión 5 pueden encontrarse en el RFC 4120 y actualizaciones posteriores. Los pasos básicos seguidos en un proceso de autenticación mediante Kerberos son los siguientes (Figura 12.30):

1. Un cliente A envía a un servidor de autenticación (AS) una solicitud requiriendo «credenciales» acerca de un servidor B dado con el que desea establecer una comunicación.
2. AS responde al cliente con las credenciales solicitadas, cifradas con la clave secreta compartida entre AS y el cliente, K_A . Dichas credenciales consisten en: (a) un ticket para el servidor

- B solicitado y (b) una clave de cifrado temporal para la sesión. El ticket contiene la identidad del cliente y una copia de la clave de sesión, todo cifrado con la clave del servidor B, K_B .
3. La clave de sesión, K_S , se utiliza para autenticar el cliente y, opcionalmente, el servidor B. También puede utilizarse, claro está, para cifrar posteriores intercambios de información entre ambas partes. Para evitar ataques de repetición, adicionalmente a la información comentada incluida en el ticket se añade otra, llamada *autenticador*, entre la que se encuentra un sello de tiempo, a fin de probar que el mensaje es original.

La ITU-T estableció en su recomendación X.509, la cual forma parte de las series X.500 de servicio de directorio, un mecanismo para la autenticación de los usuarios. X.509 es un estándar importante por cuanto que se usa en una gran variedad de contextos (p.e., en S/MIME, IPsec y SSL/TLS).

X.509 se basa en el uso de criptografía de clave pública y firma digital mediante funciones *hash*. Aunque en el estándar no se señala ningún algoritmo de cifrado concreto ni función *hash* específica, sí se recomienda el uso de RSA. El núcleo del esquema X.509 es el uso de certificados de clave pública asociados a los usuarios por una autoridad certificadora o CA («Certification Authority»).

Cada certificado X.509 contiene la clave pública de un usuario, y es firmado con la clave privada de una CA para garantizar la autenticidad del usuario. La Figura 12.31 muestra el formato de un certificado X.509, que incluye los siguientes elementos:

- *Versión*: a valor 1, 2 o 3.
- *Número de serie*: valor entero y único en la CA que identifica el certificado.
- *Identificador del algoritmo de firma*: algoritmo usado para firmar el certificado.
- *Nombre de la CA*: nombre X.500 de la CA que ha creado y firmado el certificado.
- *Período de validez*: fecha de inicio y de fin de la validez de uso del certificado.
- *Nombre del usuario*.
- *Información de clave pública del usuario*: clave pública del usuario, más un identificador del algoritmo con el que se usará aquella.
- *Identificador único de la CA*: cadena de bits opcional para identificar únicamente a la CA, en el caso de que el nombre X.500 haya sido reutilizado para otras entidades.
- *Identificador único del usuario*: cadena de bits opcional para identificar únicamente al usuario, en el caso de que el nombre X.500 haya sido reutilizado para otras entidades
- *Extensiones*: extensiones incorporadas en la versión 3 y usadas, entre otras cosas, para especificar nombres alternativos del usuario y de la CA o atributos de directorio.
- *Firma digital de la CA*: código *hash* de los otros campos, cifrado con la clave privada de la CA. Este campo incluye el identificador del algoritmo de firma.

Establecidos los certificados digitales por parte de una o varias CA³, aparece a continuación la pregunta de cómo gestionar estos de modo que se posibilite su adopción y despliegue transparente y generalizado. Surge así el concepto de PKI («Public-Key Infrastructure»), el cual, según se recoge en el RFC 4949, se refiere al *conjunto de hardware, software, personas, políticas y procedimientos necesarios para crear, gestionar, almacenar, distribuir y revocar certificados digitales basados en criptografía asimétrica*.

En relación a la PKI y a los certificados X.509, es de mencionar la disposición por parte del IETF del modelo PKIX («Public-Key Infrastructure X.509»), a través del cual se pretende el despliegue en Internet de una arquitectura de certificados (véase, entre otros, el RFC 2527).

³ Por mencionar algunas CA conocidas, sírvase citar VeriSign a nivel internacional (<http://www.verisign.com>), además de varias a nivel nacional como ACE (<http://www.ace.es>), correspondiente a la Asociación de Certificación Electrónica, Ceres (<http://www.cert.fim.es/ceres.htm>), autoridad pública de Certificación Española desarrollada para la relación usuario-administración por la Fábrica Nacional de Moneda y Timbre, y Camerfirma (<http://www.camerfirma.com>), puesta en marcha por las Cámaras de Comercio.

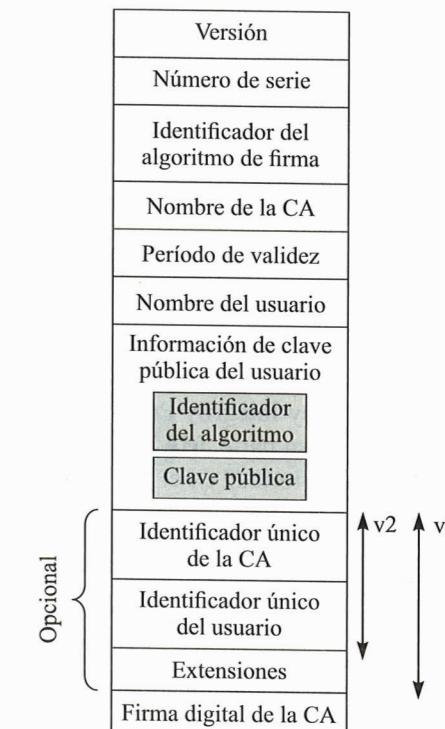


Figura 12.31. Certificado X.509.

12.3.2. Protocolos de comunicación seguros

Hasta la fecha se ha propuesto una variedad de protocolos que, sustentados en los mecanismos de seguridad reportados en los apartados anteriores, persiguen una comunicación segura entre los participantes. Seguidamente se describen algunos de los esquemas de comunicaciones seguros más conocidos y utilizados en la actualidad, organizados según la capa en la que operan, desde la más baja hasta la de aplicación, pasando por las de red y de transporte.

Aunque evidente, es de resaltar en este punto que la inclusión de seguridad en una capa dada frente a hacerlo en otras presenta como hecho diferencial la actuación directa sobre las cabeceras implicadas en la capa en cuestión. Así, incluir seguridad en la capa de aplicación no evita el posible acceso de terceros a información tal como las direcciones IP o los puertos de origen y destino.

Seguridad inalámbrica

Los sistemas de comunicación inalámbricos constituyen hoy en día uno de los principales (si no el principal) paradigmas de transmisión de información. Sin embargo, su naturaleza abierta hace a este tipo de sistemas altamente vulnerable a acciones maliciosas tales como escuchas no deseadas o comunicaciones no autorizadas con suplantación de identidad de las partes. Es por ello que se hace necesaria la adopción de mecanismos de seguridad que garanticen, cuando menos, la confidencialidad e integridad de la información transmitida así como la autenticación de las partes.

Con estos requerimientos en mente, en 1999 se introdujo en los sistemas WiFi el esquema WEP («Wired Equivalent Privacy») para proporcionar confidencialidad, integridad y autenticación en las

comunicaciones. La primera se consigue mediante el uso del esquema de cifrado de flujo RC4. Para la integridad se hace uso de un CRC de 32 bits, según lo ya estudiado en el Capítulo 4. En cuanto a la autenticación de las partes, esta puede ser *abierta* (esto es, no existe) o de tipo PSK («Pre-Shared Key»), consistente en un procedimiento de reto-respuesta a través de la disposición de una clave secreta compartida entre cada pareja origen-destino.

WEP hace uso de una clave de 40 bits más un vector de inicialización, VI, de 24 bits; esto es, la longitud total de la clave es 64 bits. Dada la reducida longitud de VI, es fácil estimular la red hasta conseguir la repetición de valores de VI. Dado, por otra parte, que VI se transmite en texto plano, el sistema WEP resulta poco seguro.

Con objeto de solventar las limitaciones de WEP, en 2003 se introdujo el sistema WPA («Wi-Fi Protected Access»). Este hace uso de TKIP para confidencialidad, MIC para integridad y PSK + 802.1X para autenticación. TKIP («Temporal Key Integrity Protocol») combina VI con la clave, en lugar de concatenar ambas como hace WEP. Por su parte, MIC («Message Integrity Code») hace referencia a los códigos MAC ya discutidos con anterioridad en el presente capítulo. Como en WEP, una posibilidad para llevar a cabo la autenticación es PSK.

Por su parte, el sistema 802.1X se basa en EAP («Extensible Authentication Protocol», RFC 3748), en el cual intervienen las entidades *solicitante*, *autenticador* y *servidor de autenticación*. Como se muestra en la Figura 12.32, el solicitante envía su autenticación al autenticador (1) antes de poder acceder libremente a la red. El autenticador, a su vez, remite la solicitud al servidor, el cual, si es correcta, concede el acceso al solicitante a través del autenticador (2). Finalmente, el solicitante podrá llevar a cabo la transmisión/recepción de información a y desde la red (3).

El solicitante aquí se refiere a una estación o usuario WiFi que desea acceder a la red, el autenticador es típicamente un punto de acceso (AP) y el servidor de autenticación un dispositivo posiblemente independiente dispuesto al efecto. Por otro lado, mencionar que el proceso de autenticación propiamente dicho se fundamenta en el empleo de un mecanismo de reto-respuesta implementado típicamente a través del protocolo RADIUS («Remote Authentication Dial In User Service», RFC 2058), o el más actual DIAMETER (RFC 3588).

En 2004 apareció WPA2, también denominado IEEE 802.11i. La diferencia principal con WPA radica en el empleo de AES/CCMP⁴ como mecanismo de cifrado, lo que proporciona una mayor robustez en las comunicaciones.

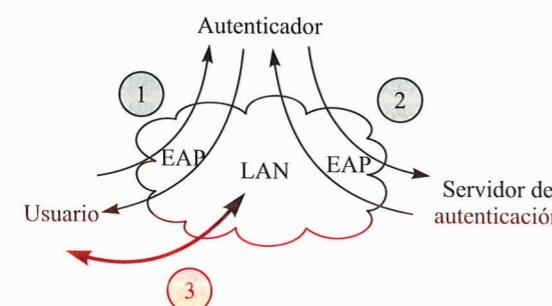


Figura 12.32. Autenticación 802.1X.

⁴ CCMP («Counter mode with Cipher block chaining MAC Protocol») es un modo de cifrado encadenado donde se considera una suerte de vector de inicialización cuyo valor (contador) varía entre bloques consecutivos.

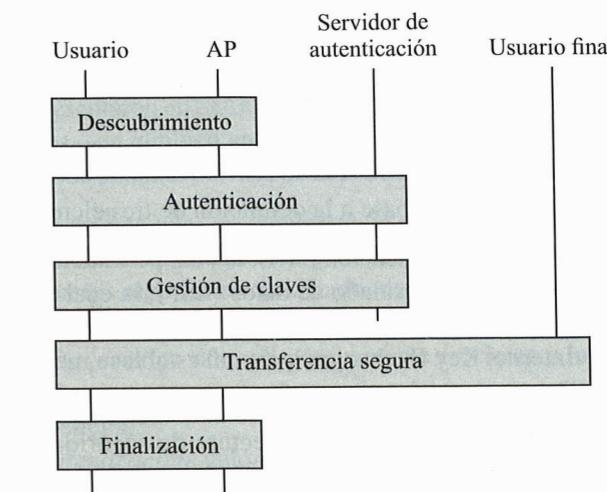


Figura 12.33. Fases de operación en una comunicación WiFi.

Tomando como base todo lo anterior, en la Figura 12.33 se muestra un diagrama donde se especifican sucintamente las fases de operación y entidades involucradas en una comunicación inalámbrica típica:

1. *Descubrimiento*. En esta primera fase, realizada entre el usuario WiFi y el punto de acceso, se llevará a cabo el reconocimiento de las partes, la definición de las capacidades de seguridad disponibles (p.e., mecanismos de cifrado) y el establecimiento de una asociación haciendo uso de estas capacidades.
2. *Autenticación*. Como se ha descrito con anterioridad, la autenticación se basa en el sistema 802.1X, debiéndose reseñar que tras el éxito de esta fase el usuario aún no tiene permiso para acceder a la red. Esto solo será posible al finalizar la siguiente etapa.
3. *Gestión de claves*. En esta fase se establecen las claves a utilizar en las comunicaciones entre el usuario y la red. Son varias las claves generadas y distribuidas en este punto: entre pares, de grupo, maestra, etc. Solo si se alcanza con éxito el final de esta etapa el usuario estará en disposición de llevar a cabo el acceso a la red.
4. *Transferencia segura*. Haciendo uso de las claves derivadas en el proceso anterior se proporcionará confidencialidad e integridad en las comunicaciones desarrolladas mediante los algoritmos anteriormente citados: TKIP, MIC, AES/CCMP.
5. *Finalización de la conexión*. Concluido el intercambio de información, la conexión segura finaliza y se restaura a su estado inicial.

IPsec

El IAB identificó a través del RFC 1636 una serie de áreas clave para proporcionar seguridad en Internet. Una de ellas se refería a la securización de las comunicaciones extremo-a-extremo en base al uso de esquemas de cifrado y de autenticación. En este contexto, el RFC 4301 introdujo IPsec («Internet Protocol security»), un conjunto de extensiones para IP orientadas a proveer autenticación, confidencialidad y gestión de claves. En esta línea, IPsec se ha convertido (junto con TSL) en uno de los protocolos de comunicación seguros más ampliamente adoptados en la actualidad. Concretamente, los servicios especificados en el RFC 4301 son:

- Control de accesos.
- Integridad de los datos.
- Autenticación del origen de los datos.
- Rechazo de paquetes repetidos.
- Confidencialidad.

Estos servicios se proporcionan en base a la definición de tres elementos principales:

- Cabecera de extensión de autenticación (AH), la cual permite la autenticación de los datos.
- Cabecera de extensión de encapsulado de datos (ESP), la cual permite autenticación y confidencialidad.
- Protocolo IKE («Internet Key Exchange»), el cual establece mecanismos de gestión de claves para su uso con IPsec.

Tomando como base estos elementos, la arquitectura de seguridad IPsec es la mostrada en la Figura 12.34. En ella se observan las siguientes entidades y sus relaciones:

- **SAD** («Security Association Database»): conjunto de asociaciones de seguridad, entendidas estas como una conexión lógica entre un emisor y un receptor que provee de servicios de seguridad a la información intercambiada entre ellos.

Una asociación de seguridad queda definida únicamente por tres parámetros:

- SPI («Security Parameters Index»), secuencia de bits asignada a una asociación y con solo significado local.
- Dirección IP de destino, correspondiente a la dirección IP del otro extremo de la asociación.
- Identificador de protocolo de seguridad, para señalar si se trata de una asociación AH o ESP.

- **SPD** («Security Policy Database»): conjunto de entradas que definen un tráfico IP (direcciones IP y puertos locales y remotos) y punteros a una asociación para ese tráfico, esto es, los mecanismos de seguridad a aplicar sobre dicho tráfico.

- **IKE**: protocolo para la gestión de las claves implicadas en las comunicaciones. Especificado en su versión 2 en el RFC 4306, la interacción habitual entre las partes es como sigue:

- En primer lugar se intercambian información acerca de algoritmos criptográficos y otros parámetros de seguridad a usar. El resultado de este intercambio es el establecimiento de una asociación de seguridad IKE inicial que permite el cifrado y la integridad de los mensajes IKE posteriores.

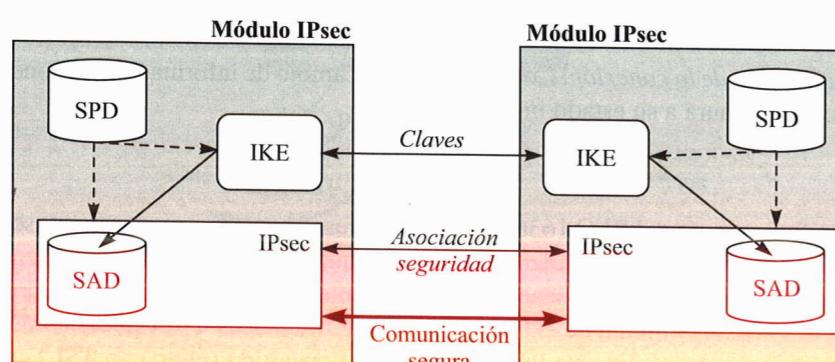


Figura 12.34. Arquitectura de seguridad IPsec.

- En una segunda etapa, las partes se autentican entre sí y establecen una asociación de seguridad IP para proteger las comunicaciones ordinarias (no IKE) entre ellas.

AH y ESP se presentaron en el Apartado 9.1.2, dentro de las cabeceras de extensión de IPv6 (las cuales, como ya dijimos, son también aplicables a IPv4). Solo por recordar los campos más relevantes de ambas cabeceras desde el punto de vista de la provisión de seguridad, haremos mención a los siguientes:

- **SPI**: campo de bits que, en combinación con la dirección IP de destino, identifica únicamente la asociación de seguridad para este paquete. Es decir, el conjunto de elecciones relacionadas con los algoritmos de cifrado, de resumen o hash y de autenticación a utilizar entre ambos extremos de la comunicación.
- **Número de secuencia**: valor monótonamente creciente a lo largo de la transmisión utilizado a efectos de detección de envíos duplicados. A través de esta técnica simple se trata de evitar ataques de repetición.
- **Datos de autenticación**: valor de comprobación de integridad para el paquete, el cual está derivado de la aplicación de algoritmos HMAC.

Tanto AH como ESP soportan dos modos de funcionamiento: *transporte* y *túnel*. En la Figura 12.35 se muestra la operación de ambos. En el primero, solo el campo de datos del paquete IP es protegido por la cabecera correspondiente. En cambio, en el modo túnel, la protección se extiende al paquete entero. Para ello, el paquete completo es considerado como el campo de datos de un nuevo paquete con una nueva cabecera más externa. Así, en el modo túnel, el paquete original se encapsula y transmite entre dos puntos de la red IP.

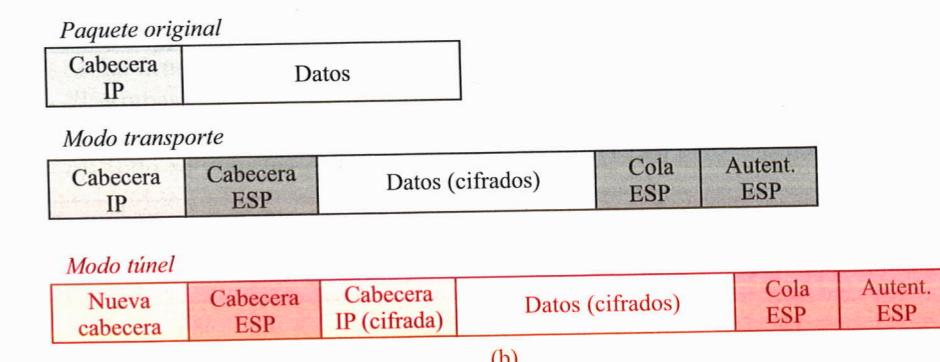
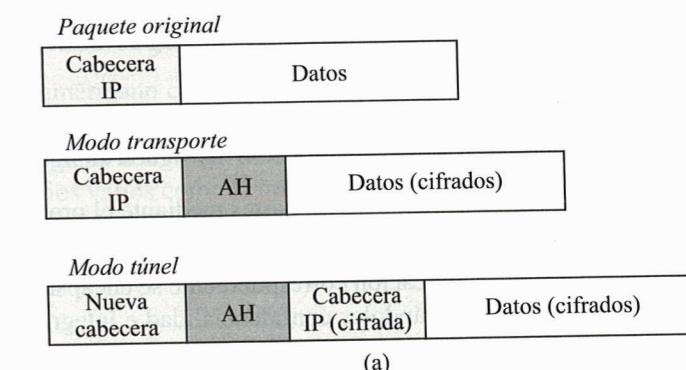


Figura 12.35. Modos de transporte y túnel para AH (a) y ESP (b).

SSL/TLS

SSL («Secure Socket Layer») fue definido originalmente por Netscape. La versión 3 de este protocolo, SSLv3, ha sido discutida por la IETF para generar el estándar TLS («Transport Layer Security»; RFC 2246, 4346 y 5246, para las versiones 1.0, 1.1 y 1.2), el cual, en suma, puede ser considerado como SSLv3.1 y casi compatible con SSLv3.

Los servicios de seguridad proporcionados por SSL/TLS son: confidencialidad, integridad y autenticación. Para ello, la arquitectura SSL es la mostrada en la Figura 12.36, donde se observan los siguientes componentes:

- Protocolo de registro:* proporciona confidencialidad e integridad mediante esquemas de cifrado de clave secreta (p.e., DES, AES) y MAC (p.e., SHA-1), respectivamente, a partir de una clave compartida establecida a través del protocolo de negociación.
- Protocolo de especificaciones de cifrado:* permite establecer el conjunto de esquemas de cifrado a usar en una conexión.
- Protocolo de alertas:* usado para indicar situaciones de error en la comunicación entre las partes (p.e., parámetro ilegal, certificado no válido).
- Protocolo de negociación:* parte más compleja de SSL, el protocolo de negociación permite la autenticación mutua de las partes (cliente y servidor) y la negociación de los algoritmos de cifrado y MAC, así como las claves criptográficas para proteger los datos enviados con SSL. Cuatro son las etapas involucradas en la operación del protocolo de negociación:
 1. Establecimiento de las capacidades de seguridad.
 2. Autenticación del servidor e intercambio de clave.
 3. Autenticación del cliente e intercambio de clave.
 4. Finalización.

En suma, las comunicaciones sustentadas sobre SSL/TSL siguen el siguiente proceso básico:

- a) Primero se autentican convenientemente las partes mediante el protocolo de negociación.
- b) También con este protocolo se establecen los algoritmos y claves a usar en las comunicaciones.
- c) Tras todo ello, los datos de la aplicación correspondiente se encapsulan en el paquete SSL del protocolo de registro, proporcionándoles confidencialidad e integridad mediante los algoritmos y claves establecidos en el paso b).

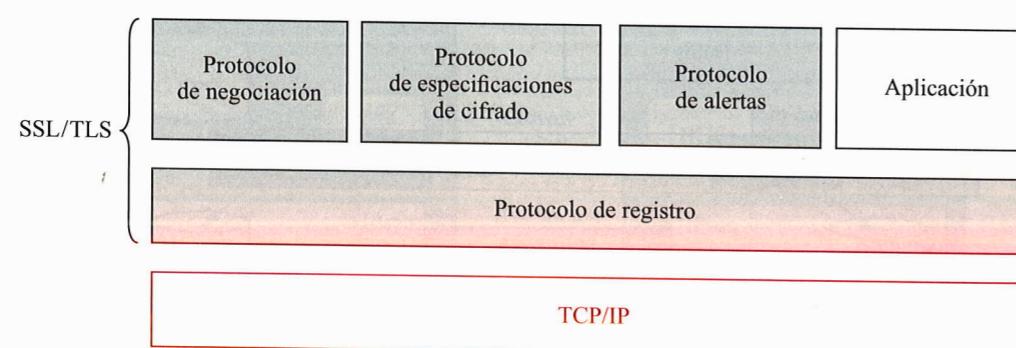


Figura 12.36. Arquitectura SSL/TLS.

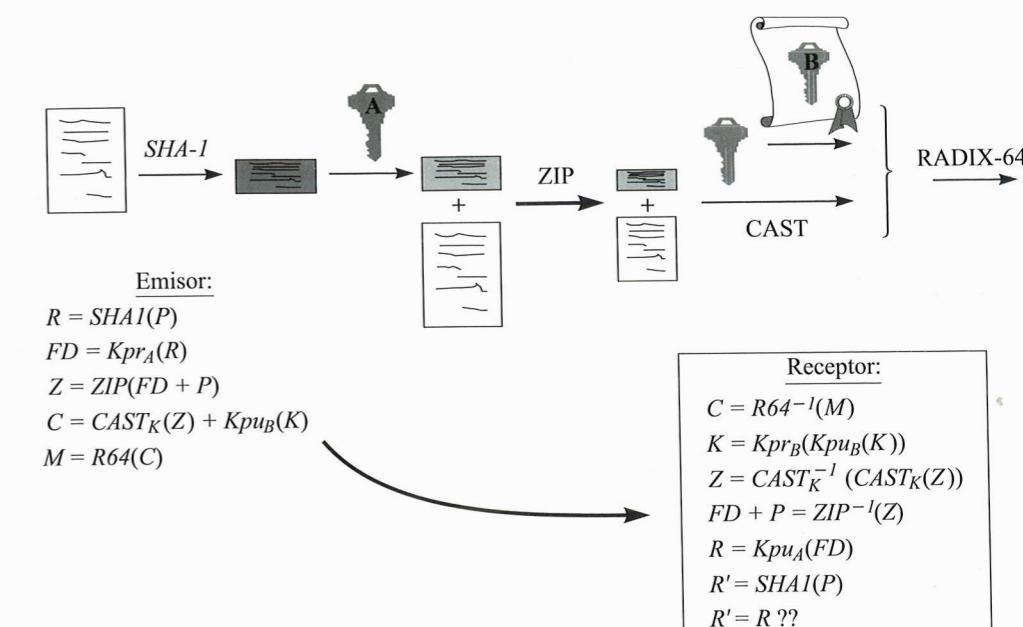


Figura 12.37. Procesamiento PGP.

Correo electrónico seguro

En 1991 Phil Zimmermann creó PGP («Pretty Good Privacy»), un paquete de propósito general que proporciona autenticación, integridad y confidencialidad combinando cifrado simétrico y asimétrico. Aunque se suele referir al servicio de correo electrónico, PGP puede ser usado para proveer de seguridad a otros servicios varios como la transferencia de ficheros. Hoy PGP es un estándar de la IETF llamado OpenPGP y está definido en el RFC 4880.

En la Figura 12.37 se muestra el proceso seguido en la transmisión de un mensaje, P , mediante PGP:

— El emisor:

- a) Obtiene el resumen del mensaje mediante alguno de los algoritmos permitidos, como SHA-1.
- b) Cifra RSA⁵ dicho resumen con su clave privada, generando así la firma digital asociada al mensaje.
- c) La firma, junto con el mensaje, se comprimen utilizando ZIP.
- d) Ambas cosas se cifran mediante algoritmos como CAST-128⁶ con una clave secreta de sesión generada al efecto, la cual se cifra RSA usando la clave pública del receptor.
- e) Todo se codifica según el esquema RADIX-64 (similar a BASE64 salvo por el hecho de que se añade una suma de verificación CRC de 24 bits) y se envía hacia el extremo receptor.

⁵ Aunque no afecta al funcionamiento conceptual aquí presentado, hemos de señalar que, por lo general, PGP, como otros muchos protocolos, utiliza el esquema de intercambio de claves Diffie-Hellman para, subsiguientemente, llevar a cabo un cifrado simétrico en lugar de uno RSA.

⁶ En lugar de CAST-128, otros algoritmos de cifrado permitidos son IDEA («International Data Encryption Algorithm»), 3DES o AES.

— El receptor:

- a) Decodifica RADIX-64 el mensaje recibido.
- b) Descifra la clave de sesión mediante su clave privada y RSA.
- c) Descifra, de acuerdo al algoritmo de cifrado simétrico, el mensaje y su firma comprimidos.
- d) Descomprime ZIP ambas cosas.
- e) Descifra el resumen del mensaje a partir de la firma haciendo uso de la clave pública del emisor.
- f) Calcula el compendio del mensaje recuperado y lo compara con el obtenido en el paso anterior. Si ambos coinciden, se da por válida la transmisión.

Otro protocolo de correo seguro es S/MIME («Secure/Multipurpose Internet Mail Extensions»), el cual puede consultarse, entre otros, en los RFC 2632 y 2633 (actualizados a través de 5750 y 5751). En términos de funcionalidad, S/MIME es parecido a PGP, permitiendo el cifrado de los datos y/o la integridad de los mismos. El primero de los servicios se denomina *recubrimiento de datos* y el segundo *firmado de datos*. El firmado de los datos se consigue cifrando el resumen del contenido con la clave privada del emisor y, tras ello, codificando BASE64 esta firma además del contenido. Si solo se codificase BASE64 la firma digital pero no el contenido, el servicio se denomina *firmado claro de datos*.

Los algoritmos criptográficos permitidos contemplados en S/MIME son DSS («Digital Signature Standard»), RSA y Diffie-Hellman. Por su parte, para la firma digital se usan SHA-1 y MD5.

Aunque solo sea citarlo sin entrar en mayores detalles en este punto, es de señalar también aquí la existencia de DKIM («DomainKeys Identified Mail»), especificación de la IETF a través del RFC 6376. En esencia, DKIM funciona como sigue:

1. Un correo es firmado con la clave privada del dominio administrativo donde se genera el mensaje.
2. En recepción, el agente de entrega de correo puede acceder a la clave pública correspondiente vía un DNS y verificar la firma, autenticando así que el mensaje procede del dominio administrativo pretendido. En caso contrario, el correo será descartado.

En resumen, DKIM está diseñado para proporcionar autenticación de los mensajes de correo electrónico de forma transparente al usuario final.

12.3.3. Control de accesos

No menos importante que la provisión de seguridad a la información en tránsito es evitar el acceso de terceros no autorizados a recursos estáticos, entendiendo por tales cuentas de usuarios, ficheros y otros recursos hardware y software disponibles. En este aspecto de la seguridad vamos a centrar nuestra atención de forma sucinta a lo largo del presente apartado del tema.

El control de accesos a un sistema en red puede desglosarse en las siguientes tareas:

- Control de accesos básico. Antes de plantearse medidas más complejas, un buen control de accesos a un sistema debe partir de algunas consideraciones de base:
 - Dado que una de las principales fuentes de problemas radica en la explotación de vulnerabilidades del software de que se dispone, la primera premisa de trabajo debe ser la instalación de las últimas versiones disponibles de dicho software, así como la actualización continua del mismo a través de los sucesivos parches desarrollados al efecto.

Este principio de actuación no debe entenderse como exclusivo de un control de accesos, sino que transciende este para ser considerado como una política genérica de seguridad y, en consecuencia, debe aplicarse a cualquier aspecto de la administración y mantenimiento de un sistema.

- Del mismo modo, resulta prioritario el establecimiento de una buena política de permisos que impida el acceso a recursos propios de otros usuarios. En este sentido, cabe hacer mención al hecho de que la mayor parte de los excesos cometidos en un sistema están provocados por los propios usuarios del mismo.

- El acceso a un sistema se fundamenta, habitualmente, en la consideración de palabras de paso o *passwords* asociadas a los usuarios. Una buena filosofía por parte del administrador del sistema, aunque generalmente impopular entre los usuarios, pasa por cambiar periódicamente dichas claves de acceso. No solo eso, sino que además es conveniente proteger y supervisar con cierta frecuencia los ficheros asociados a dichas claves a fin de evitar la existencia de entradas no permitidas y/o no controladas por el administrador.
- Otra premisa básica consiste en el control de la «seguridad física»; es decir, la gestión de las conexiones y topología del sistema.
- Evitar el uso de software de origen desconocido, en especial cuando se trate de servicios de red (ver más adelante). Dentro de esta categoría podemos mencionar programas que propagan virus, troyanos, etc. (véase Apartado 12.3.4 más adelante).

— Control de accesos a servicios en red. Aunque, como se ha dicho, gran parte de los excesos cometidos contra un sistema tienen su origen en el propio sistema, la disponibilidad de servidores tales como los de correo electrónico, de páginas web, etc., pueden constituir agujeros de seguridad importantes para accesos desde el exterior. Es por ello conveniente tener claras, entre otras, las siguientes premisas:

- El sistema más seguro es aquel aislado del mundo. Aunque esto no hay que tomarlo al pie de la letra, sí es importante tener claro que solo hay que habilitar aquellos servicios estrictamente necesarios, prohibiendo sin dudar el resto.
- La activación de los servicios seleccionados no puede llevarse a cabo alejadamente; antes bien, debe hacerse de forma racionalizada, estableciendo los niveles de privilegios mínimos precisos, de manera que se reduzcan al máximo los riesgos.
- Interesa disponer de mecanismos de defensa que supongan barreras difíciles de superar por parte de potenciales atacantes. Entre estos mecanismos podemos destacar los dispositivos conocidos como *cortafuegos*, los cuales, dada su gran popularidad actual, comentaremos específicamente más adelante.
- Por encima de los mecanismos anteriores conviene disponer de procedimientos que permitan un control más exhaustivo sobre los servicios solicitados. Un ejemplo lo constituyen los *recubrimientos TCP* («TCP wrappers» en inglés), originarios de los sistemas Unix.

— Monitorización y rastreo. Adicionalmente a las medidas de seguridad mencionadas anteriormente, todo buen administrador debe llevar a cabo constantes tareas de monitorización y supervisión del sistema a fin de localizar, y evitar si es posible, potenciales accesos no permitidos. Para ello se cuenta con herramientas tan diversas como los denominados escáneres y rastreadores («sniffers») a través de los que se sondan posibles vulnerabilidades existentes en el sistema y se monitoriza el tráfico en la red, respectivamente. Estas herramientas, propias del sistema o adquiridas externamente al mismo, permiten la generación de ficheros de traza en los que se almacena la actividad del sistema a lo largo del tiempo. La supervisión llevada a cabo por el administrador se fundamenta básicamente en el análisis de dichos ficheros de traza.

Seguidamente se presentan de forma breve tres tipos de tecnologías relacionadas con el control de accesos a sistemas: esquemas biométricos, cortafuegos y redes privadas virtuales.

Tabla 12.1. Técnicas de acceso biométricas más usadas, ordenadas de mayor a menor fiabilidad.

Técnica de reconocimiento	Descripción
Vascular	Reconocimiento del patrón de venas de la mano, de los dedos, ...
Iris	Reconocimiento del patrón reticular del iris del ojo
Retina	Reconocimiento del patrón de venas de la retina
Dactilar	Reconocimiento del patrón de las líneas de la huella dactilar
Geometría mano	Reconocimiento de la forma de la mano
Cara	Reconocimiento facial, el cual puede ser 2D o 3D
Voz	Reconocimiento del patrón vocal del usuario
Escritura/firma	Reconocimiento del patrón de escritura

Accesos basados en técnicas biométricas

Frente al uso generalizado de los tradicionales *passwords* (o PINs) para posibilitar el acceso a recursos e instalaciones, en los últimos años se está extendiendo el empleo de las conocidas como técnicas biométricas. La biometría, del griego *bios*, vida, y *metron*, medida, se refiere al estudio de rasgos físicos humanos intrínsecos. Con este tipo de tecnologías se persiguen dos fines principales:

1. Evitar la habitual necesidad de disponer de varios *passwords* diferentes para acceder a distintos recursos.
2. Al tratarse de características personales, la seguridad del sistema se supone superior a la mera utilización de una clave alfanumérica, fácilmente reproducible.

Aunque no es objetivo en este punto entrar a discutir en profundidad los distintos tipos de técnicas de acceso biométricas existentes, sí se considera oportuno hacer mención al menos a su existencia. En la Tabla 12.1 se indican las más relevantes a fecha de hoy, priorizadas en orden de fiabilidad de mayor a menor.

Todas las técnicas referidas presentan el mismo esquema de funcionamiento general. Este, esquematizado en la Figura 12.38, consta de las siguientes fases:

- *Entrenamiento*: en una primera fase es precisa la derivación de un patrón o modelo asociado a las características biométricas de cada uno de los usuarios registrados en el sistema. Para ello será necesario disponer de una base de datos adecuada al efecto.

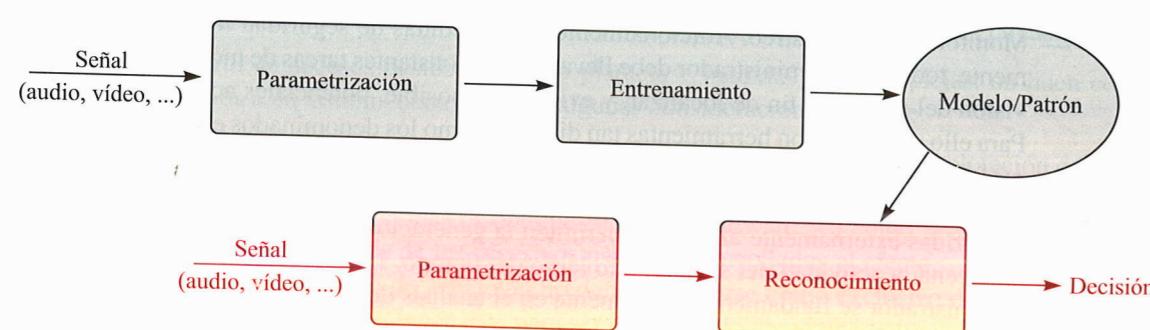


Figura 12.38. Operación típica realizada en un control de accesos basado en técnicas biométricas.

— *Reconocimiento*: solo en base a la existencia del modelo o patrón antes mencionado será posible la decisión de si un determinado usuario forma parte del grupo de personas autorizadas a acceder al recurso o instalación correspondiente.

— *Pre-procesamiento y parametrización*: tanto el entrenamiento como el reconocimiento posterior precisan de una fase previa donde, por una parte, se «normalice» la señal analizada (solucionándose así, por ejemplo, posibles problemas de tamaño, rotaciones, etc.) y, tras ello, se obtengan las características principales que definen la señal de entrada.

Cortafuegos

Un mecanismo de seguridad de uso aceptado universalmente en el entorno de redes de computadores lo constituyen los dispositivos conocidos como *cortafuegos* («firewall» en inglés). Como se esquematiza en la Figura 12.39, la idea consiste en disponer un equipo pasarela que haga de barrera entre nuestro sistema y el exterior, de manera que: (a) todo el tráfico entre ambos entornos lo atraviese, (b) facilitándose así su gestión y control.

Existen dos tipos básicos de cortafuegos: de filtrado y *proxy* o de aplicación. En los primeros, también llamados IP, se efectúa un filtrado de los paquetes que se reciben en el dispositivo, permitiendo o denegando el acceso correspondiente en base a la información contenida en la cabecera (p.e., direcciones y puertos). En este sentido, un cortafuegos de filtrado dispone de tres filtros:

- a) *De entrada*: filtro a través del que se trata de controlar el tráfico que tiene como destino el propio cortafuegos, independientemente de si el origen se encuentra en la red corporativa o en el exterior. Desde este punto de vista, podemos decir que el filtrado de entrada actúa sobre los servicios proporcionados por el cortafuegos.
- b) *De salida*: frente al anterior, este filtro trata de controlar el tráfico que tiene como origen el propio cortafuegos, independientemente de si el destino se encuentra en la red corporativa o en el exterior. Así pues, el filtrado de salida se refiere a los servicios solicitados como cliente por el cortafuegos.
- c) *De retransmisión*: relativo a la función de pasarela realizada por el cortafuegos, este tercer filtro actúa sobre el tráfico procedente de una de las partes y con destino la otra, permitiendo, denegando o restringiendo dicha comunicación. Podríamos así, por ejemplo, denegar el acceso FTP desde el exterior, pero no el servicio web o el de correo electrónico.

Los procesos de filtrado realizados pueden ser selectivos en base a diversos criterios tales como puerto, dirección IP (origen y/o destino) y/o protocolo. Además del filtrado, los cortafuegos IP realizan dos funciones de interés:

- *Contabilidad* («accounting» en inglés): el cortafuegos no solo acepta o prohíbe ciertos paquetes, sino que permite la supervisión de todo el tráfico que lo atraviesa. Esto, por supuesto, resulta de enorme importancia desde el punto de vista de la administración del sistema.



Figura 12.39. Esquema conceptual de un cortafuegos.

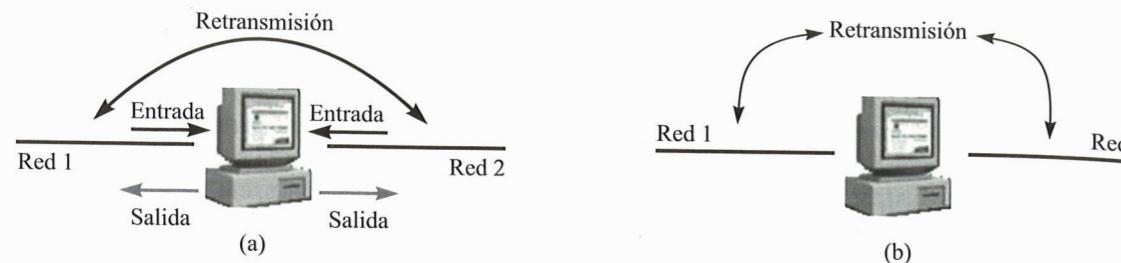


Figura 12.40. Esquematización de cortafuegos IP o de filtrado (a) y proxy (b).

- Enmascaramiento («masquerading» en inglés): a través de esta función, el cortafuego reenvía (si así se autoriza) los paquetes procedentes de la red interna tras sustituir la dirección IP origen de los mismos por la suya externa⁷. Esta función es útil, por ejemplo, para permitir la conectividad exterior de la red interna si en esta se hiciese uso de direcciones IP privadas no válidas en el exterior, o, aun en el caso de que sí lo fuesen, para ocultarlas por motivos de seguridad (ver Apartado 8.8 del texto).

Aunque la función de enmascaramiento resulta de un alto interés, hemos de decir que su utilización puede resultar incompatible con el desarrollo efectivo de ciertos servicios.

Como se muestra en la Figura 12.40(a), el filtro de retransmisión en un cortafuego IP es tal que, aceptado un servicio dado, se establece una comunicación directa entre los *hosts* origen y destino situados en las redes interconectadas. Frente a este esquema, la filosofía seguida en un cortafuego *proxy* es bien distinta. En este caso —ver Figura 12.40(b)—, la comunicación origen-destino no es directa, sino indirecta; es decir, la transmisión no es una: origen-destino, como sucede en un cortafuego IP, sino dos: origen-*proxy* más *proxy*-destino. Esto se traduce en que la instalación del cortafuego *proxy* implica la puesta en marcha de dos versiones por servicio: la estándar correspondiente al servicio a desarrollar, si así lo permite el cortafuego, para la comunicación *proxy*-exterior, y una modificada, para la comunicación red interna-*proxy*. Todo ello, por supuesto, de forma transparente al usuario. Aunque hay que decir que la instalación de un cortafuego de tipo *proxy* resulta más compleja que la de uno de filtrado, el nivel de seguridad proporcionado es superior en el primer caso.

Estos tipos de cortafuegos pueden ser sin estado o con estado. En el primer caso las decisiones se realizan sobre paquetes individuales, mientras que en un cortafuego con estado se almacenan y tienen en cuenta las conexiones/sesiones existentes.

Por hacer mención a algún software de cortafuegos mencionaremos las herramientas de Linux *iptables*, para la configuración de cortafuegos de filtrado, y el protocolo SOCKS⁸, como ejemplo de *proxy* que garantiza el acceso a través de la autenticación del cliente.

La disposición de los dispositivos cortafuegos puede ser diversa, si bien la más habitual es la indicada en la Figura 12.41. En ella se observa la existencia de un cortafuego externo, justo después del *router* de acceso a Internet, y uno interno, que protege la red corporativa de la organización/empresa. Entre ambos existe la conocida como DMZ («DeMilitarized Zone»), compuesta por un conjunto de servicios públicos accesibles externamente (p.e., servidores HTTP, de correo electrónico, etc.) y que, evidentemente, deben ser protegidos.

⁷ Como pasarela que es, un cortafuego debe disponer de, al menos, una dirección IP correspondiente a cada uno de los entornos que interconecta. De esta forma, si el cortafuego constituyese la puerta de enlace de una intranet a Internet, una de sus direcciones sería pública y la otra privada, no válida para transmisiones globales.

⁸ En el RFC 1928 se puede encontrar la versión 5 de SOCKS.

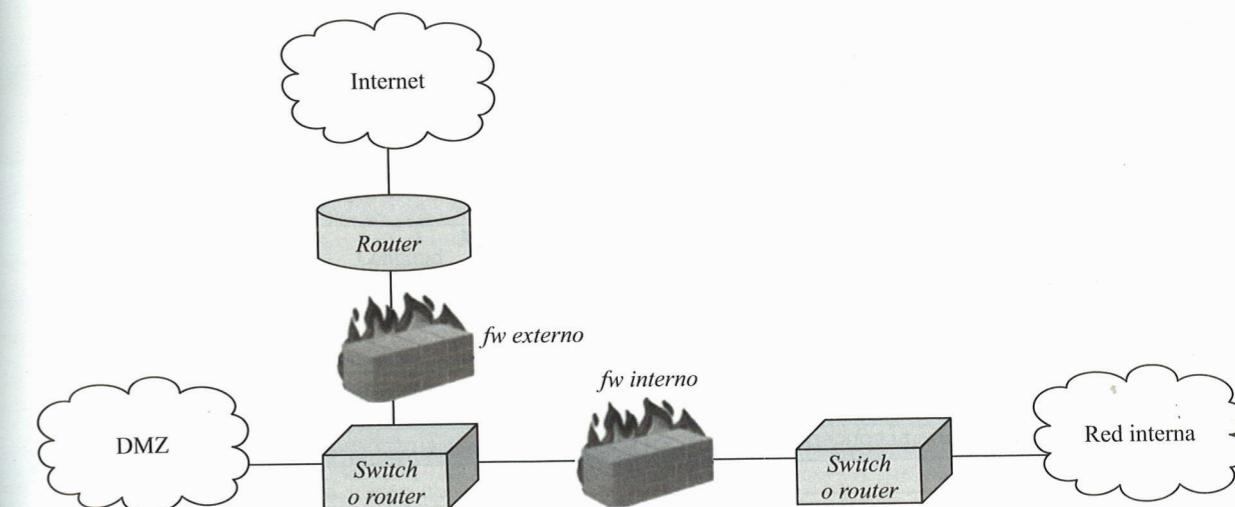


Figura 12.41. Disposición típica de cortafuegos en un entorno de red corporativo.

Redes privadas virtuales

Una disposición habitual también de los cortafuegos es la referida a las denominadas *redes privadas virtuales* o VPN («Virtual Private Network»). Las VPN hay que entenderlas principalmente como una respuesta flexible, escalable y de bajo coste para la interconexión de entornos corporativos remotos y de estos con empleados deslocalizados. En este sentido, las VPN hacen uso de una infraestructura de red compartida (por ejemplo, la pública proporcionada por Internet) para dar respuesta a las distintas necesidades de interconexión corporativa privada por parte de empresas (entre sucursales y/o entre estas y sus empleados). Como se muestra en la Figura 12.42, las compañías o empresas que utilizan una VPN establecen una conexión con su proveedor de servicio a través de lo que se conoce como *punto de presencia* o PoP (ver Apartado 5.4) y dejan que sea este el encargado de enviar los datos hacia el destino correspondiente vía la infraestructura de red compartida (Internet, por ejemplo).

Dada la consideración de una infraestructura de red compartida (generalmente pública), la tecnología VPN debe proporcionar seguridad en las comunicaciones. En concreto, cuatro son los aspectos críticos de seguridad que deben ser garantizados en este tipo de entornos: autenticación, confidencialidad, integridad de los datos y control de acceso. Para dar respuesta a estas distintas cuestiones se recurre a los esquemas y técnicas estudiados a lo largo de los Apartados 12.3.1 a 12.3.3 del tema.

Frente a las redes privadas implementadas en el pasado, basadas en el empleo de líneas físicas dedicadas, la tecnología VPN actual se caracteriza por el hecho de que las líneas utilizadas son lógicas (de ahí el adjetivo *virtual* utilizado para estas redes), las cuales se establecen de forma dinámica según las necesidades requeridas en cada momento. Las transmisiones realizadas sobre dichas líneas se fundamentan en el concepto ya visto con anterioridad de *túnel*, encapsulado de paquetes gracias al cual se posibilita la transmisión de datos correspondientes a una red a través de las conexiones de una red diferente. De este modo, un túnel permite el encaminamiento de un paquete de datos con unas características especiales sobre nodos intermedios que no implementan y, en consecuencia, no son capaces de operar con, las prestaciones requeridas. Con ello se posibilitaría, por ejemplo, la transmisión *unicast* de paquetes *multicast* sobre nodos que no soportan este último tipo de envío.

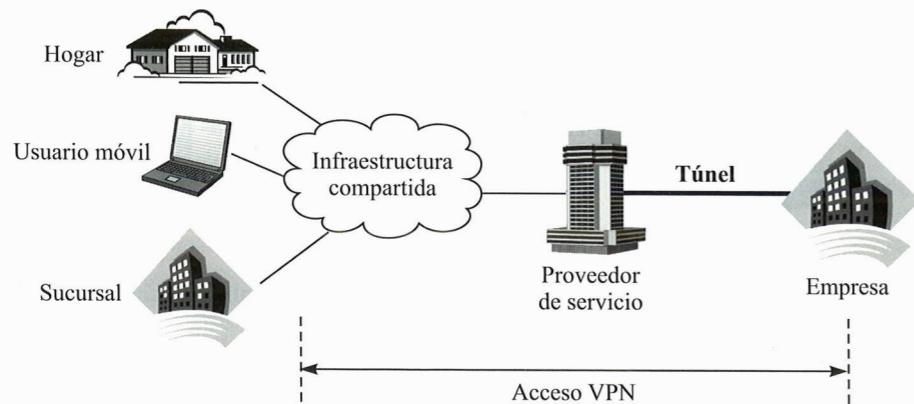


Figura 12.42. Esquema general conceptual de una red privada virtual.

Un túnel queda definido por dos puntos extremos y un protocolo de comunicación utilizado entre ellos. Los extremos pueden ser de dos tipos: computador personal o LAN con un dispositivo de acceso (de tipo cortafuegos por ejemplo), utilizándose solo dos de las posibles combinaciones de ellos: cliente-LAN y LAN-LAN, apareciendo involucrado en ambos casos un dispositivo de acceso.

Por otro lado, son diversos los protocolos de encapsulado propuestos para la creación de VPN, algunos, los más habituales, operando en la capa de enlace y otros en la de red. Por lo que respecta a los segundos, mencionar el ya discutido IPsec, además de otros como IP-IP (RFC 2003), encapsulado mínimo (RFC 2004) y GRE («Generic Routing Encapsulation», RFC 1701 y 1702). En cuanto a protocolos de encapsulado a nivel 2, enlace, los más conocidos son: PPTP («Point-to-Point Tunneling Protocol», RFC 2637), L2F («Layer Two Forwarding», RFC 2341) y L2TP («Layer Two Tunneling Protocol», RFC 2661).

PPTP funciona básicamente de la siguiente manera:

- El objetivo es el encapsulado de tramas PPP sobre un túnel GRE —ver Figura 12.43(a)—.
- El control del túnel se realiza en base a mensajes PPTP específicos sobre TCP (puerto 1723) cuyo formato de cabecera es el mostrado en la Figura 12.43(b).

Los mensajes PPTP de control del túnel son de cuatro tipos: de gestión de control de la conexión, de gestión de llamada, de reporte de error y de control de sesión PPP.

Por su parte, L2F es muy similar a PPTP pero no depende de IP, pudiendo trabajar con otras tecnologías como ATM o FR.

L2TP fue publicado en 1999, si bien su versión 3 data de 2005. En esta se proporcionan características de seguridad, se mejora el encapsulado y se posibilita el transporte de datos distintos a PPP sobre IP (p.e., ATM, Ethernet, FR). Las características de L2TP son:

- Los paquetes L2TP se envían sobre UDP (puerto 1701).
- Dado que L2TP no proporciona habitualmente una adecuada confidencialidad ni autenticación, se suele usar en combinación con IPsec en lo que se conoce como L2TP/IPsec.
- El túnel L2TP se establece entre dos puntos conocidos como LAC («L2TP Access Concentrator») y LNS («L2TP Network Server»); el primero es el iniciador del túnel y el segundo actúa como servidor a la espera de nuevos túneles.
- Establecido un túnel L2TP, las capas superiores podrán hacer uso de él en lo que se conoce como sesiones o llamadas L2TP.

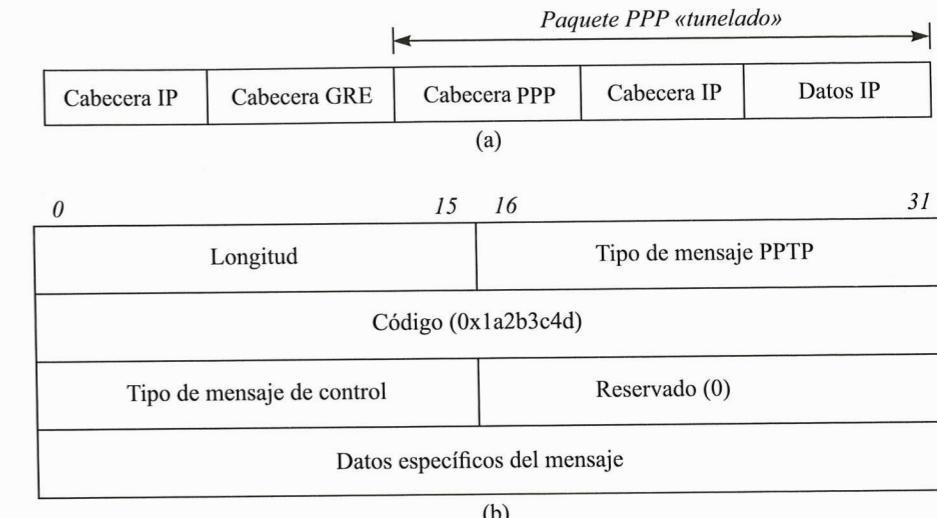


Figura 12.43. Envío de datos con PPTP (a) y cabecera de paquetes PPTP (b).

0	15	16	31
Indicadores y versión	Longitud		
ID túnel	ID sesión		
<i>Ns</i>	<i>Nr</i>		
Tamaño offset	Relleno offset		
Datos específicos del mensaje			

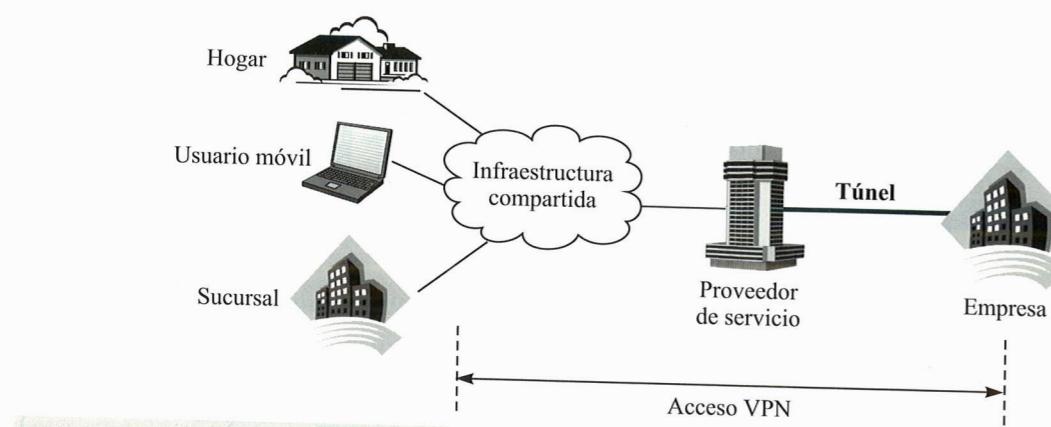
Figura 12.44. Formato de paquete L2TP (en cursiva, los campos opcionales).

- Los paquetes intercambiados sobre un túnel L2TP pueden ser de control o de datos, lo cual se especifica a través del primer campo del paquete cuyo formato general es el mostrado en la Figura 12.44. Los paquetes de control se refieren a acciones de tipo *establecimiento de túnel* o *establecimiento de sesión* y los de datos al encapsulado de tramas PPP sobre L2TP.

Más información de interés relacionada con la tecnología VPN puede encontrarse en el sitio web <http://www.vpnc.org>.

12.3.4. Seguridad en sistemas

Toda la seguridad que podamos proporcionar a nivel de protección de la información y de las comunicaciones no sirve de nada si los sistemas que intervienen (*hosts*, *routers*) están comprometidos. Esto es, si la operación de estos está controlada por un usuario malicioso, la confiabilidad del sistema entero estará en entredicho.



Un túnel queda definido por dos puntos extremos y un protocolo de comunicación utilizado entre ellos. Los extremos pueden ser de dos tipos: computador personal o LAN con un dispositivo de acceso (de tipo cortafuegos por ejemplo), utilizándose solo dos de las posibles combinaciones de ellos: cliente-LAN y LAN-LAN, apareciendo involucrado en ambos casos un dispositivo de acceso.

Por otro lado, son diversos los protocolos de encapsulado propuestos para la creación de VPN, algunos; los más habituales, operando en la capa de enlace y otros en la de red. Por lo que respecta a los segundos, mencionar el ya discutido IPsec, además de otros como IP-IP (RFC 2003), encapsulado mínimo (RFC 2004) y GRE («Generic Routing Encapsulation», RFC 1701 y 1702). En cuanto a protocolos de encapsulado a nivel 2, enlace, los más conocidos son: PPTP («Point-to-Point Tunneling Protocol», RFC 2637), L2F («Layer Two Forwarding», RFC 2341) y L2TP («Layer Two Tunneling Protocol», RFC 2661).

PPTP funciona básicamente de la siguiente manera:

- El objetivo es el encapsulado de tramas PPP sobre un túnel GRE —ver Figura 12.43(a)—.
- El control del túnel se realiza en base a mensajes PPTP específicos sobre TCP (puerto 1723) cuyo formato de cabecera es el mostrado en la Figura 12.43(b).

Los mensajes PPTP de control del túnel son de cuatro tipos: de gestión de control de la conexión, de gestión de llamada, de reporte de error y de control de sesión PPP.

Por su parte, L2F es muy similar a PPTP pero no depende de IP, pudiendo trabajar con otras tecnologías como ATM o FR.

L2TP fue publicado en 1999, si bien su versión 3 data de 2005. En esta se proporcionan características de seguridad, se mejora el encapsulado y se posibilita el transporte de datos distintos a PPP sobre IP (p.e., ATM, Ethernet, FR). Las características de L2TP son:

- Los paquetes L2TP se envían sobre UDP (puerto 1701).
- Dado que L2TP no proporciona habitualmente una adecuada confidencialidad ni autenticación, se suele usar en combinación con IPsec en lo que se conoce como L2TP/IPsec.
- El túnel L2TP se establece entre dos puntos conocidos como LAC («L2TP Access Concentrator») y LNS («L2TP Network Server»); el primero es el iniciador del túnel y el segundo actúa como servidor a la espera de nuevos túneles.
- Establecido un túnel L2TP, las capas superiores podrán hacer uso de él en lo que se conoce como sesiones o llamadas L2TP.

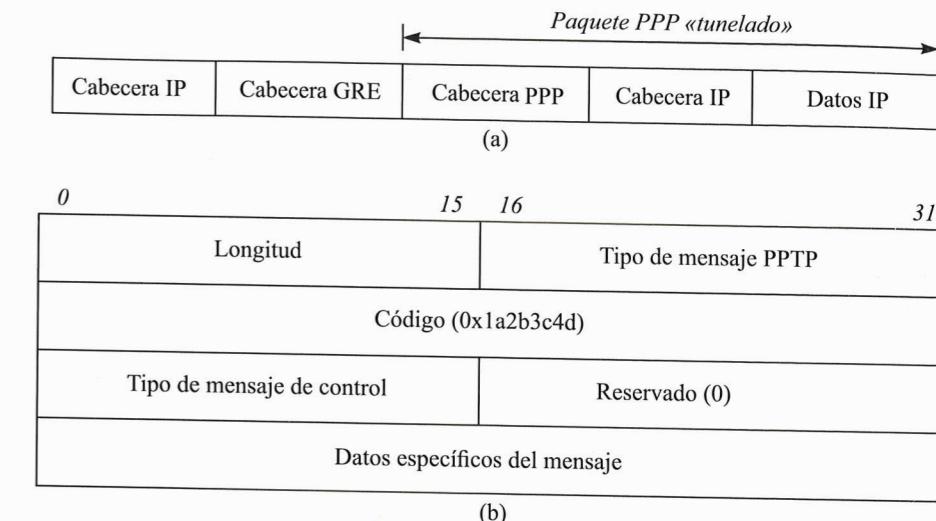


Figura 12.43. Envío de datos con PPTP (a) y cabecera de paquetes PPTP (b).

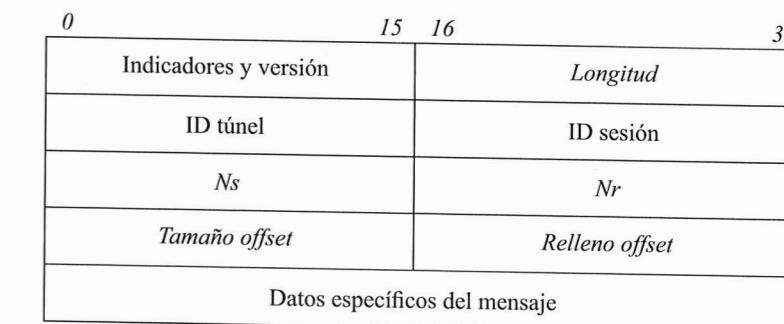


Figura 12.44. Formato de paquete L2TP (en cursiva, los campos opcionales).

— Los paquetes intercambiados sobre un túnel L2TP pueden ser de control o de datos, lo cual se especifica a través del primer campo del paquete cuyo formato general es el mostrado en la Figura 12.44. Los paquetes de control se refieren a acciones de tipo *establecimiento de túnel* o *establecimiento de sesión* y los de datos al encapsulado de tramas PPP sobre L2TP.

Más información de interés relacionada con la tecnología VPN puede encontrarse en el sitio web <http://www.vpnc.org>.

12.3.4. Seguridad en sistemas

Toda la seguridad que podamos proporcionar a nivel de protección de la información y de las comunicaciones no sirve de nada si los sistemas que intervienen (*hosts*, *routers*) están comprometidos. Esto es, si la operación de estos está controlada por un usuario malicioso, la confiabilidad del sistema entero estará en entredicho.

Seguidamente se discuten distintos riesgos y procedimientos para conseguir el control, total o parcial, de un sistema informático en red. Solo siendo conscientes de ellos podremos adoptar los mecanismos necesarios para prevenirlos, detectarlos y, llegado el caso, resolverlos.

Malware

Las formas en que un sistema dado puede verse comprometido son variadas, si bien la gran mayoría de ellas se refiere a la actuación de software malicioso o *malware* (del inglés «malicious software»). Se entiende, así, por *malware* un cierto software malicioso incluido o insertado intencionadamente (y sin permiso) en un sistema para causar daño. Existe multitud de tipos de *malware*, unos independientes y otros parásitos (dependientes del sistema en el que operan), unos estáticos y otros replicantes (i.e., con capacidad de auto-generación).

Entre los tipos de *malware* existentes podemos mencionar en un lugar preeminente los virus. El término virus, por analogía con su variante humana, hace referencia a todo software que infecta a otro, modificándolo para llevar a cabo la ejecución de una tarea perniciosa y distinta a aquella para la que ha sido originalmente diseñado.

Las partes o etapas de un virus son: *vector de infección*, referente a la forma en que se expande; *disparo*, relativo a la acción que desencadena la infección (p.e., una hora de una fecha dada); y *carga*, correspondiente a la acción final puesta en marcha por el virus.

En cuanto a las vías de infección de los virus, las más frecuentes son:

- Dispositivos USB/DVD/CD infectados.
- Copias de programas, que incluyen el *malware* insertado.
- Sitios web fraudulentos.
- Webs legítimas pero infectadas.
- Propagación a través de redes sociales.
- Redes P2P, de nuevo con el *malware* insertado.
- Adjuntos en correos electrónicos no solicitados (*spam*).
- Fallos de seguridad del propio SO del sistema.

Las metodologías que sustentan las tecnologías antivirus son las siguientes:

- *Primera generación*: escaneo simple a la búsqueda de patrones conocidos.
- *Segunda generación*: escaneo heurístico en busca de comportamientos indeseados.
- *Tercera generación*: identificación de actividades fuera de lo normal.
- *Cuarta generación*: combinación de las anteriores.
- *Avanzadas*: entre otras, una posibilidad consistiría en ejecutar el software analizado en un entorno totalmente controlado, de manera que no tengan impacto los posibles efectos maliciosos. Es la metodología que se conoce como *generic decryption*.

Bajo la denominación genérica de virus podemos encontrar multitud de tipos de *malware* con características diferenciadoras. Así, un gusano es un virus con tres características especiales:

- a) Capacidad de *auto-propagación*, es decir, que puede desplegarse y expandirse por sí mismo. En la Figura 12.45 se muestra la evolución típica de la propagación de un gusano, donde se observan tres fases: dos lentas, al inicio y fin, y una rápida, intermedia.
- b) Capacidad de *auto-replicación*, esto es, no se precisan ficheros terceros para su expansión.
- c) Para conseguir ambos hechos, los gusanos suelen explotar vulnerabilidades existentes en los sistemas.

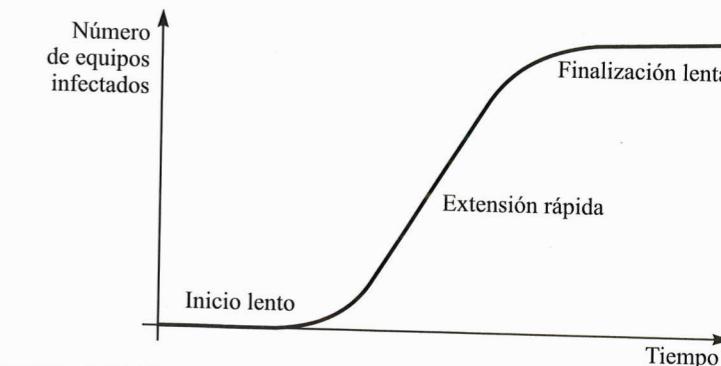


Figura 12.45. Evolución típica de la propagación de un gusano.

En la Tabla 12.2 se indican sucintamente algunos ejemplos de virus y gusanos bien conocidos por su impacto hasta la fecha. Otros virus y/o gusanos más actuales son *Flashback*, para equipos Mac; *Koobface*, que se propaga por Facebook; *DarkAngle* y *KuluoZ*, troyanos para el robo de información varía como *passwords*; y *Ainslot*, gusano con puerta trasera.

Los gusanos actuales son cada vez más complejos, presentando como características principales las siguientes:

- Multi-plataforma, lo que significa que pueden afectar a distintos tipos de sistemas simultáneamente: Windows, Linux, IOS, etc.
- Polimórficos, de manera que pueden mutarse a sí mismos para cambiar su comportamiento y dificultar su detección.
- Metamórficos, lo que implica que el *malware* se puede reescribir completamente y, en suma, convertirse en otro distinto.
- Multi-exploit, aprovechándose no solo de una, sino de varias vulnerabilidades, lo que los hace más devastadores y difíciles de detener.
- Zero-day, lo que significa que el *malware* trata de explotar una vulnerabilidad aún desconocida por la comunidad a fin de magnificar su efecto y distribución y retrasar su detección y solución.

Las aproximaciones actualmente propuestas para la detección de gusanos son:

- Basadas en firmas, donde se trata de encontrar patrones o comportamientos identificativos de ataques conocidos en el software analizado.
- Filtrado de contenidos, similar a la anterior pero centrada en el contenido del gusano en lugar de en su firma.
- Aleatorización de destinos, esto es, se analizan los destinos con los que se desea comunicar a la búsqueda de una distribución aleatoria (o no) en los mismos que permita concluir la existencia de un escáner.
- Limitación del tráfico, donde se restringe el número de máquinas nuevas con las que un *host* puede mantener una comunicación en una ventana de tiempo dada.
- Bloqueo del tráfico, de manera que este se inhibe cuando se supera un cierto umbral en la tasa de salida o en el número de intentos de conexión.

Hemos de mencionar que esta técnica y la anterior son simultáneamente de detección y de respuesta.

Tabla 12.2. Ejemplos de virus y gusanos.

Nombre	Año de aparición	Descripción
CIH (Chernobyl)	1998	El código fuente del virus CIH es capaz de sobrescribir en determinadas circunstancias la BIOS y dejar la máquina inoperante.
Melissa	1999	Este virus se distribuyó por primera vez en la discusión del grupo de noticias Usenet:alt.sex. Estaba dentro de un archivo llamado «List.doc», que decía contener una lista de contraseñas con las que se permitía el acceso a 80 sitios web pornográficos. La forma original del virus fue enviada por e-mail a muchas personas.
ILOVEYOU	2000	Virus tipo gusano que se propaga a través de correo electrónico e IRC. Su apariencia en forma de correo es un mensaje con el asunto «ILOVEYOU» y el fichero adjunto LOVE-LETTER-FOR-YOU.TXT.vbs. Cuando se abre el archivo infectado el gusano infecta nuestra máquina y se intenta auto-enviar a las entradas en la agenda de OutLook.
Code Red	2001	Este gusano explota una vulnerabilidad en el indexado de la distribución de software IIS. Se extiende aprovechando una vulnerabilidad de buffer overflow en el archivo IDQ.DLL. En concreto, lo que hace es usar una cadena larga de caracteres repetidos «N» hasta conseguir que se desborde el buffer, permitiendo al gusano ejecutar código propio e infectar a la máquina atacada. Una versión de este, Code Red II, no lleva a cabo ataque pero sí dispone de una puerta trasera para permitir ataques. Code Red II intenta infectar máquinas de la misma subred de la infectada.
Nimda	2001	Gusano que recupera la lista de direcciones encontrada en las libretas de direcciones de Microsoft Outlook y Eudora, así como también direcciones contenidas en archivos HTML que se encuentran en el disco duro del equipo infectado. A continuación, el virus envía un correo electrónico a todos estos destinatarios sin texto y con un asunto escogido en forma aleatoria (y a menudo, muy extenso). Agrega al mensaje un adjunto llamado Readme.exe o Readme.eml (archivo que contiene un ejecutable).
Klez	2002	Este virus explota una vulnerabilidad de Internet Explorer por la cual es capaz de auto-ejecutarse con solo visualizar el correo electrónico en el que llega como adjunto. El virus es capaz de impedir el arranque del sistema y de inutilizar ciertos programas.
Win32/Simile	2002	Se trata de un virus de construcción sumamente compleja. Infecta los archivos ejecutables de todos los directorios de discos duros locales y compartidos en red que estén visibles en el momento que el virus es ejecutado. No posee carga destructiva, salvo el hecho de la infección a archivos ejecutables. Estos archivos infectados pueden mostrar ciertos mensajes en determinadas fechas.
Blaster/Lovsan	2003	Virus con una capacidad de propagación muy elevada en base a una vulnerabilidad de buffer overflow en los sistemas Windows. Sus efectos consisten en lanzar ataques de denegación de servicio contra la web de Microsoft «Windows Update» y provocar inestabilidad en el sistema infectado.
Sobig worm	2003	Gusano de envío masivo de correo (spam) cuya propagación se realiza a todas las direcciones electrónicas encontradas dentro de los ficheros de extensiones: .txt, .eml, .html, .htm, .dbx y .wab. El correo en el que se propaga el gusano parece como si fuese enviado por «big@boss.com». También realiza copias de sí mismo en máquinas remotas a través de recursos compartidos en red.
Mydoom	2004	Se propaga masivamente a través del correo electrónico y la red P2P KaZaa. Este virus utiliza asuntos, textos y nombres de adjuntos variables en los correos en los que se envía, por lo que no es posible identificarlo o filtrarlo fácilmente. Tiene capacidades de puerta trasera que podrían permitir a un usuario remoto controlar el ordenador infectado.
Sasser	2004	Gusano que se propaga aprovechando una vulnerabilidad en el proceso LSASS («Local Security Authority SubSystem»). Está programado para ejecutar procesos que analizan direcciones IP aleatorias que buscan sistemas con LSASS vulnerables en el puerto 455/TCP. El virus instala un servidor FTP en el puerto 5554 para que otros equipos infectados puedan descargarlo. Después, cuando encuentra un equipo vulnerable, el gusano abre un shell remoto en el equipo (TCP 9996) y hace que descargue una copia del gusano.

Otro tipo de *malware* son los troyanos, altamente conocidos y temidos por los usuarios de equipos informáticos. Los troyanos se refieren a programas independientes que no infectan en sí mismos, sino que tienen como misión una variedad de acciones contra el sistema: robo de información (como los *spyware*, a la búsqueda de información personal como códigos de tarjetas; o los *keyloggers*, esto es, detección de *passwords* e información varía a través de la captura de la entrada desde teclado).

Los troyanos constan de dos partes: un cliente, relativo al usuario malicioso que envía las órdenes a ejecutar, y un servidor, que las ejecuta en la máquina infectada y devuelve el resultado al cliente.

Algunos otros términos relativos a formas de *malware* son:

- *Puerta trasera (back door)*: mecanismo que evita los procedimientos de seguridad desplegados en un sistema y permite accesos no autorizados.
- *Rootkit*: conjunto de herramientas de *hacker*⁹ usadas tras haber entrado en un sistema y haber ganado privilegios de administrador o *root* del mismo.
- *Bot / Zombie*: programa en una máquina infectada, controlado desde otra y que se usa para llevar a cabo ataques contra terceros. El conjunto de *bots* controlados desde un mismo *botmaster* es lo que se conoce como *botnet*, o red de *bots*.

Ataques DoS

Un tipo de ataques de alto impacto en la actualidad y referidos a un aspecto de la seguridad no cubierto hasta este punto son los de *denegación de servicio* o DoS («Denial of Service»). Estos provocan que un servicio o recurso del sistema atacado sea inaccesible a los usuarios legítimos, esto es, se trata de un ataque contra la disponibilidad. Pensemos por ejemplo en un atacante que lleva a cabo la petición continua de solicitudes de servicio a un servidor. Es evidente que el efecto consecuente será que las peticiones cursadas por usuarios legítimos resulten retrasadas, si no directamente desestimadas.

La forma más evidente para denegar un cierto servicio es el envío masivo de peticiones. Sin embargo, este comportamiento indeseable tiene el riesgo de que resulta fácilmente detectable. Por ello, frente a los denominados DoS de fuerza bruta (esto es, envío indiscriminado de solicitudes de servicio; Figura 12.46) surgen los de *baja tasa* (o *low-rate*) consistentes en el envío selectivo de peticiones (solo) en aquellos momentos en que resultan efectivas desde el punto de vista del objetivo perseguido. Como es evidente, son también de mencionar los ataques DoS debido a vulnerabilidades del software. Por ejemplo, el así conocido como *ping de la muerte (ping of death, o PoD)* consistía en el envío de un mensaje ICMP de eco malformado, con una longitud superior a la esperada, lo que provocaba en las primeras implementaciones de la pila TCP/IP un problema de desbordamiento de *buffer o buffer overflow* (véase más adelante).

No obstante lo anterior, dadas las altas prestaciones de los sistemas servidor actuales, además de que estos se suelen configurar en forma de granja¹⁰, no resulta fácil para un usuario malicioso conseguir denegar un cierto servicio (menos aún si el ataque es de baja tasa). Surge así la variante DoS conocida como DDoS («Distributed DoS»), en la cual se ataca la disponibilidad del servicio a través de la existencia de varios atacantes simultáneos. La distribución del ataque entre los distintos atacantes hace aquél difícilmente detectable.

Los procedimientos de ataque más usuales en DoS son:

- Consumo de recursos computacionales, tales como ancho de banda, espacio de disco o tiempo de procesador. Es el caso del ataque *SYN flooding*, donde se genera una elevada cantidad de segmentos *SYN* de TCP para saturar los recursos de la entidad de transporte correspondiente.

⁹ Un *hacker* es un usuario que busca y explota vulnerabilidades de los sistemas.

¹⁰ Una granja de servidores no es más que la disposición conjunta de varias máquinas para la distribución de tareas, lo que deriva en la consecución efectiva de una «máquina global» de más altas prestaciones que las individuales.

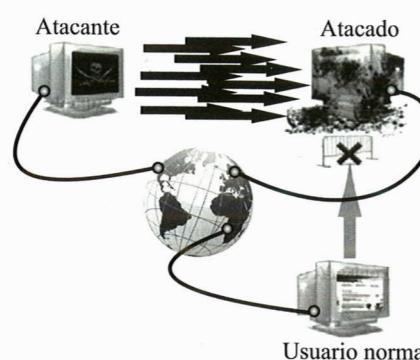


Figura 12.46. Operación típica de un ataque DoS, donde el elevado número de solicitudes del atacante evita la normal provisión del servicio a usuarios legítimos.

- Alteración de información de configuración, como la relativa a las tablas de encaminamiento.
- Alteración de información de estado, como la interrupción de sesiones TCP (*TCP reset*).
- Interrupción de componentes físicos de red.
- Obstrucción de medios de comunicación entre usuarios de un servicio y la víctima, de manera que no puedan comunicarse adecuadamente.

Hemos de concluir la exposición de los ataques DoS reseñando la escasez de mecanismos de detección (y consecuentemente de respuesta) propuestos hasta la fecha en la literatura especializada.

Intrusiónes

Otro concepto relevante relacionado con la seguridad es el de *intrusión*. Se entiende por intrusión un ataque llevado a la práctica con éxito y consistente en un evento en el que se explota una vulnerabilidad, resultando en una brecha de seguridad en el sistema; esto es, una violación de la política de seguridad del mismo. Desde este punto de vista, un ataque consiste en la formulación y ejecución específica de un plan contra el esquema de seguridad de un sistema. Si este evento se lleva a cabo con éxito desde el punto de vista del atacante y, por tanto, ocasiona una brecha de seguridad en el sistema, entendemos que se ha producido una intrusión.

Un proceso intrusivo consiste típicamente en la ejecución de una serie de etapas bien identificadas (Figura 12.47). En esta discusión nos centraremos en intrusos externos, puesto que las amenazas provenientes del interior del perímetro del sistema pueden eludir directamente las fases preliminares del proceso.

1. Fase 1: Exploración del objetivo

En esta primera fase el atacante intentará recopilar tanta información del objetivo deseado como sea posible. Esta exploración está altamente dirigida a la búsqueda de vulnerabilidades, resultando en la obtención de la más variada y vital información.

Las tareas a realizar durante esta fase se suelen agrupar en tres etapas, denominadas *intelligence gathering*:

- a) *Elaboración del perfil del objetivo*. En este punto se hace acopio de información que delimita las características del sistema objetivo desde el punto de vista de la seguridad: nombres de usuario, números de teléfono, rangos de direcciones IP, servidores DNS y de correo, etc. Así, por ejemplo, la provisión de un servicio HTTP o de correo electrónico por parte

de una organización puede revelar detalles importantes sobre el sistema, como nombres de máquinas y el sistema operativo que ejecutan.

- b) *Escaneo*. Se trata de la detección de qué sistemas se encuentran activos desde el exterior y qué servicios ofrecen, utilizando para ello técnicas como uso de mensajes *ping* y escaneo de puertos. La información recopilada en esta etapa se refiere principalmente a los servicios TCP/UDP existentes, la arquitectura del sistema, la topología de la red y el tipo de sistema operativo.
- c) *Enumeración*. Extracción de nombres de cuentas de usuario o de recursos existentes en los sistemas. La obtención de esta información suele implicar una acción más agresiva que las anteriores, en el sentido de que su puesta en práctica puede ocasionar toda una intrusión en sí misma.

2. Fase 2: Identificación de vulnerabilidades

Toda la información y conocimientos adquiridos en la fase previa son utilizados para la determinación de vulnerabilidades en los elementos del sistema. Estas suelen ser debidas a diseños defectuosos, implementaciones poco rigurosas o gestión deficiente. Así, las aplicaciones servidoras tipo HTTP, DNS y correo electrónico, además del sistema operativo, suelen ser las más explotadas. En la bibliografía especializada puede consultarse la magnitud del problema de vulnerabilidades en sistemas en red, siendo responsabilidad del equipo de administración la realización de supervisiones periódicas, actualizaciones de software, etc. que traten de solucionar aquellas.

3. Fase 3: Penetración

Identificadas las vulnerabilidades susceptibles de ser explotadas, la fase siguiente es la realización efectiva del ataque o intrusión. Para ello, el primer paso es la ejecución de algún tipo de proceso en el sistema destino para la obtención de privilegios. Como es evidente, las técnicas a utilizar para llevar a cabo la penetración son tan variadas como las propias vulnerabilidades a explotar. Solo por mencionar algunas, citaremos los ataques *XSS* y *SQL Injection*, ambos relacionados con el acceso a servicios web.

4. Fase 4: Escalado

La penetración en un sistema suele conducir al acceso a un conjunto limitado de recursos. Por tanto, el siguiente paso natural en un ataque es el escalado de los privilegios conseguidos, a fin de ampliar el control sobre el sistema. Dicho escalado suele implicar la repetición de los pasos anteriores, pero esta vez desde dentro del sistema. Así, en este punto podemos decir que confluyen los ataques externos y los internos.

5. Fase 5: Modificaciones del acceso

Conseguidos el acceso y los privilegios necesarios para tener un control total del sistema, el atacante trata de modificar este a fin de posibilitar posteriores entradas sin necesidad de

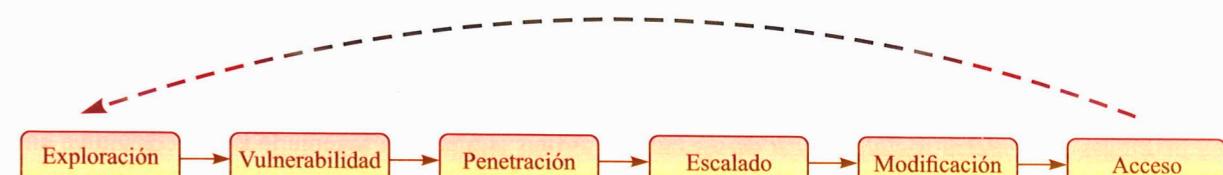


Figura 12.47. Fases de una intrusión.

repetir todas las etapas ya comentadas. Para ello, el intruso llevará a cabo algunas acciones tales como agregar usuarios con altos privilegios y robar contraseñas de otros usuarios o servicios (mediante *sniffers*, *keyloggers*), además de instalar herramientas tipo *rootkit* que permitan ocultar su actividad y troyanos para la generación de puertas traseras (*back doors*).

6. Fase 6: Acceso a la información

Llegados a este punto, el atacante no solo ha conseguido control total sobre el sistema, sino que ha garantizado el acceso oculto al mismo. El robo, modificación o destrucción de la información en el sistema pueden llegar a constituir pérdidas irreparables de todo tipo a la organización atacada.

Llevado a cabo con éxito un ataque, es frecuente que el atacante desee conservar el acceso al sistema y utilizarlo como trampolín para posteriores ataques a otros sistemas. Es evidente que el seguimiento de un ataque a través de otros sistemas resulta un problema de difícil solución, lo que supone una garantía para el atacante ante la potencialidad de ser descubierto. En esta línea, en ocasiones se dice que una fase adicional de toda intrusión es la de *repetición*, entendiendo por esta la ejecución de todos los pasos ya mencionados desde el sistema penetrado con objeto de tratar de acceder a nuevos sistemas externos.

Es interesante mencionar en este punto el proyecto *Metasploit* (<http://www.metasploit.com>) como herramienta de alto interés para llevar a cabo tests de penetración en sistemas, a fin de detectar vulnerabilidades y, en su caso, darles solución.

Concluiremos diciendo que, como es natural, no toda intrusión debe llevar a cabo todas y cada una de las etapas comentadas, o ejecutarlas en el orden estricto indicado.

Para el despliegue de una intrusión, el atacante hace uso de diversas herramientas y técnicas conducentes a la consecución de los resultados deseados. Algunas de los más frecuentes son las siguientes, de uso variado en las distintas fases antes descritas:

- *Escaneo*: el escaneo forma parte de las primeras etapas de cualquier ataque, y su objetivo es la identificación de elementos activos y alcanzables en el sistema, los servicios que ofrece y otra información útil para etapas posteriores
- *DNS*: mediante un simple proceso de consulta y análisis de la información proporcionada por uno o varios DNS se puede obtener información muy valiosa sobre el mapeado de nombres de máquina en direcciones IP
- *Desbordamiento de buffer (buffer overflow)*: mencionados ya con anterioridad en varias ocasiones, los desbordamientos de *buffer* constituyen una amplia y explotada categoría de ataques, todos ellos inspirados en debilidades en el software. El núcleo de este tipo de ataques lo constituye el paso de un valor inesperado (por ejemplo, un nombre de fichero excesivamente largo) como parámetro a una aplicación.
- *Puertas abiertas y abuso de relaciones de confianza*: en el contexto de la seguridad en las TIC es bien conocida la ley que vincula de manera inversamente proporcional la funcionalidad, usabilidad y facilidad de manejo de un sistema con los mecanismos de seguridad presentes en el mismo. Con objeto de alcanzar un equilibrio entre estos factores, los responsables de la configuración y mantenimiento de los sistemas tienden a simplificar los mecanismos de control de acceso y autenticación entre máquinas que se consideran mutuamente confiables.
- *Ingeniería social*: a pesar de ser una de las más antiguas, la ingeniería social constituye una herramienta de ataque ampliamente utilizada y efectiva en la mayoría de los casos. Esta puede definirse como *saltarse los mecanismos de seguridad engañando a alguien con la capacidad de hacerlo por ti*.

- *Ataques contra aplicaciones*: las aplicaciones, especialmente las que actúan como servidores, constituyen un punto de acceso al sistema y a sus recursos. Los ataques contra ellas pueden perseguir diversos objetivos.
- *Malware*: los problemas asociados a lo que ha venido en denominarse código malicioso son bien conocidos, como ya se ha discutido con anterioridad.
- *Operaciones con paquetes* (repetición, modificación, escucha y captura, construcción e inyección): la manipulación del tráfico de red constituye, en todas sus vertientes, una de las mayores amenazas para un sistema.

Frente a las técnicas de ataque, también es de reseñar la existencia de diversos mecanismos de defensa que persiguen reducir, si no eliminar completamente, la ocurrencia de intrusiones. Estas pueden encuadrarse en seis categorías que se aplican en función de la distancia al elemento defendido, y son:

1. *Revocación*. Luchar y eliminar cualquier amenaza potencial antes de que esta tenga la oportunidad de desplegar un ataque es la forma más contundente de evitarla. No obstante, esta aproximación es la mayoría de las veces peligrosa, cuando no ilegal.
2. *Prevención*. Prohibir, o al menos limitar severamente, la verosimilitud de una intrusión exitosa reduce considerablemente la exposición al riesgo. Es posible, por ejemplo, aislar al sistema de las conexiones con redes externas o utilizar mecanismos de restricción, como los cortafuegos, para controlar y filtrar las actividades deseadas.
3. *Disuasión*. Comprende todas las medidas de persuasión frente al atacante para que este abandone la intrusión. La forma más evidente de llevarla a la práctica es incrementar la percepción de las consecuencias negativas que podrían afectarlo si es capturado.
4. *Detección*. La detección comprende todas las técnicas y mecanismos de identificación de los intentos de intrusión, de manera que puedan ser puestas en marcha las respuestas adecuadas (ver *contramedidas* más adelante). La mayor parte de las veces estas últimas toman la forma de notificaciones a la autoridad pertinente. Los principales problemas se presentan bajo dos aspectos: las falsas alarmas ante una actividad que no tiene carácter intrusivo (falsos positivos), y la ausencia de alarmas en el caso de que la intrusión se esté llevando a cabo (falsos negativos).
5. *Desviación*. Las medidas de desviación pretenden atraer al atacante hacia un cebo haciéndole creer que ha conseguido el acceso deseado, cuando en realidad ha sido derivado hacia un área en la que no puede realizar ningún daño. Este tipo de sistemas son conocidos comúnmente como «tarro de miel» (*honeypot* o *honeynet*).
6. *Contramedidas*. Se entiende por tales un conjunto de técnicas y mecanismos que, de manera activa y autónoma, se oponen a un ataque mientras este se encuentra en progreso. En general, estos mecanismos podrían funcionar sin la necesidad de un sistema previo de detección del ataque, aunque la discriminación proporcionada por aquel incrementará considerablemente la efectividad de los mismos.

El despliegue en un sistema dado de un conjunto de mecanismos de protección que engloben todas y cada una de las facetas (o *murallas*) anteriores es lo que se conoce como *defensa en profundidad*. Ello se representa en la Figura 12.48.

A pesar de la necesidad y eficacia de todas estas barreras o medidas, en base a la disposición de herramientas específicamente diseñadas para ello, estas pueden ser traspasadas por actividades maliciosas. Por ello, es usual contar con personal técnico especializado responsable de la seguridad del sistema (SSO, «Site Security Officer»). Dicho personal se encarga de analizar los registros de actividad del sistema (trazas o *logs*) a fin de detectar actividades que comprometen la seguridad del mismo y responder ante ellas tomando las decisiones apropiadas.

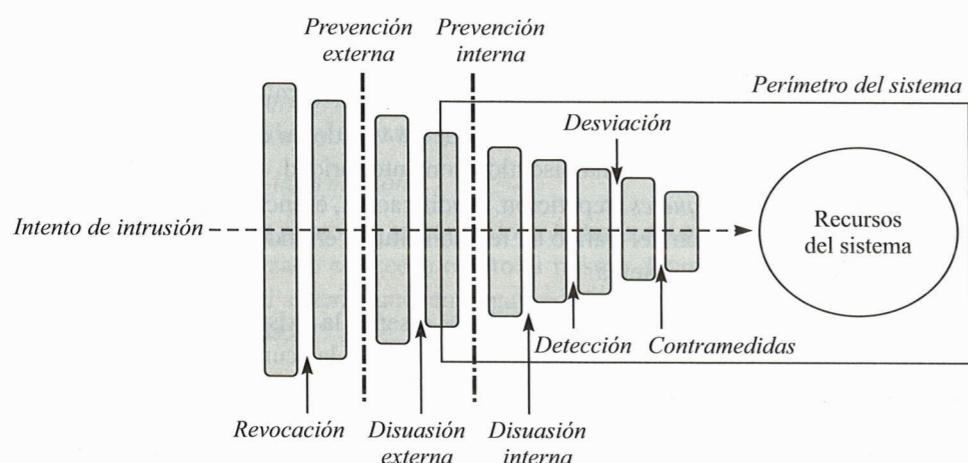


Figura 12.48. Defensa en profundidad ante intrusiones.

Tanto por el alto volumen habitual de información manejado, como por el bajo tiempo de respuesta deseado en la toma de decisiones, se hace necesario automatizar en la medida de lo posible las capacidades de detección del SSO. Si bien esto entraña una alta dificultad, a lo que contribuye activamente el grado de complejidad de los sistemas y redes actuales, resulta incuestionable la necesidad de investigar técnicas y métodos que permitan mejorar la detección de violaciones en las políticas de seguridad de los sistemas.

En este contexto y con este ánimo surgieron los denominados *sistemas de detección de intrusos* o IDS (del inglés «Intrusion Detection System»), cuyo uso se ha extendido en los últimos años y entre algunos de los cuales cabe destacar *Snort* (<http://www.snort.org>) y *Bro* (<http://www.bro.org>).

12.3.5. Aspectos legales y éticos de la seguridad

Aunque requeriría un estudio mucho más extenso del aquí realizado, no podemos concluir la introducción a la seguridad en redes y sistemas proporcionada en el presente capítulo sin referenciar aunque sea brevemente dos aspectos de alta relevancia en este campo: la protección de contenidos en el marco de la propiedad intelectual y la ciberdelincuencia.

Propiedad intelectual

Según define la organización mundial de la propiedad intelectual (WIPO, «World Intellectual Property Organization»; <http://www.wipo.int>), esta se refiere a *toda creación de la mente humana en relación a invenciones, obras literarias y artísticas, así como símbolos, nombres e imágenes utilizadas en comercio*. La legislación permite proteger la propiedad intelectual mediante las patentes (invenciones), los derechos de autor (creador) y las marcas (signos para diferenciar entre productos o servicios).

Las normas que regulan la propiedad intelectual en nuestro ámbito inmediato son:

- Ley 23/2006, que modifica el real decreto legislativo RDL 1/1996.
- Decreto 2001/29/CE, sobre la armonización de ciertos aspectos de los derechos de autor y relacionados en la sociedad de la información de la Unión Europea, conocida comúnmente como Directiva de la Unión Europea sobre derechos de autor o EUCD (del inglés «European Union Copyright Directive»).

- Ley 5/1998, que incorpora al derecho español la Directiva 96/9/CE del Parlamento Europeo sobre la protección jurídica de las bases de datos.
- Real decreto legislativo RDL 1/1996, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual.

En este contexto surgen las tecnologías DRM («Digital Rights Management»), a través de las cuales se pretende el control de acceso a recursos digitales. Ello puede llevarse a cabo a través de dos aproximaciones principales:

1. Control sobre el consumo, donde se trata de hacer un seguimiento del uso de los productos/servicios a través de procedimientos tales como autenticación *online* de los usuarios o alteración del software para la monitorización de su utilización.
2. Control sobre el recurso (ERM, «Enterprise digital Rights Management»), en base al cifrado de este o a su «marcado» con información bien del autor (*watermarking*) bien del consumidor (*fingerprinting*). Estas técnicas, que conforman los esquemas denominados de *esteganografía*, permiten el seguimiento de la potencial distribución fraudulenta de los recursos protegidos.

En el marco de la protección de contenidos es de mencionar la Ley Orgánica 15/1999 de protección de datos (LOPD; <http://www.agpd.es>), cuyo objeto es garantizar y proteger, en lo que concierne a los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente su honor e intimidad personal y familiar.

Ciberdelincuencia

Si bien la expansión generalizada de Internet y las TIC proporciona indudables beneficios a la sociedad en múltiples aspectos, no debemos perder de vista los por otra parte numerosos riesgos que implica su uso. En este sentido, son cada vez más frecuentes los episodios reportados en todo el mundo acerca de la nueva lacra en que se está convirtiendo año a año la *ciberdelincuencia* o *cibercrimen*. De acuerdo con el convenio de ciberdelincuencia del Consejo de Europa, se define esta como todos aquellos delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos. Asimismo, se incluyen en esta categoría los delitos relacionados con el contenido, la propiedad intelectual y derechos afines. También tienen cabida aquí los delitos relacionados con el racismo y la xenofobia.

Los tipos penales reconocidos en España en relación al cibercrimen son:

- Amenazas.
- Exhibicionismo y provocación sexual.
- Prostitución y corrupción de menores.
- Descubrimiento y revelación de secretos.
- Calumnias.
- Injurias.
- Estafas.
- Daños.
- Propiedad intelectual.
- Propiedad industrial.
- Mercado y consumidores.
- Falsedad documental.
- Racismo y xenofobia.

Finalizar sin más estos breves apuntes en relación a la ciberdelincuencia señalando diversos esfuerzos realizados a nivel judicial en la persecución de esta nueva tipología de criminalidad. Son así de citar el Grupo de Delitos Telemáticos de la Guardia Civil (<http://www.gdt.guardiacivil.es>) y la Brigada

de Investigación Tecnológica de la Policía Nacional (http://www.policia.es/org_central/judicial/udef/bit_alertas.html) a nivel español, y el *European Cybercrime Centre* (EC3; <https://www.europol.europa.eu/ec3>) a nivel europeo.

RESUMEN

Una vez estudiada la pila de protocolos TCP/IP en los capítulos precedentes, este tema ha comenzado tratando un aspecto de vital importancia en todo sistema en red: el de la gestión y administración del mismo. Tras comentar los diversos aspectos involucrados en el desarrollado de las labores propias de una gestión de redes adecuada, a continuación se ha introducido el modelo de gestión seguido en TCP/IP. Acerca de este se ha presentado su estructura a partir de los elementos que la forman: una o más estaciones de gestión, los agentes instalados en los dispositivos gestionados, las bases de datos o MIB a las que se refieren los objetos gestionables y el protocolo de gestión SNMP, a través del cual se realiza la comunicación estación de gestión-agente. Adicionalmente a estos elementos se ha comentado el espacio de nombres de identificación de objetos sobre el que, siguiendo la sintaxis ASN.1, se refieren los distintos objetos de las MIB. También se ha discutido el modelo de gestión seguido en las redes OSI: CMIS/CMIP. Con la misma arquitectura funcional que SNMP, se ha hecho especial énfasis en el modelo de comunicaciones específico derivado del empleo del protocolo CMIP.

Seguidamente al análisis de la gestión de redes se ha abordado uno de los aspectos más importantes de esta, el de la seguridad. Planteados los diversos aspectos relacionados con ella, a lo largo del Apartado 12.3.1 se han presentado técnicas de cifrado, tanto de clave secreta como pública, con objeto de proporcionar confidencialidad en las comunicaciones; esquemas de integridad, para la protección de los contenidos; procedimientos de autenticación, que permiten la identificación de los extremos participantes en una transmisión; y algoritmos de firma digital, que posibilitan el no repudio además de proporcionar privacidad e integridad.

Tomando como base todo lo anterior, seguidamente se han presentando diferentes protocolos que, actuando en distintas capas (enlace, red, transporte o aplicación), persiguen la provisión de seguridad en las comunicaciones entre las partes implicadas. Se han discutido así protocolos como IPsec y SSL/TLS, los cuales constituyen la base de la mayor parte de los servicios de usuario seguros disponibles actualmente.

Una cuestión adicional de relevancia tratada acerca del tema de seguridad en redes de computadores ha sido la referente al control de accesos en un sistema en red, en concreto a través de la consideración de los dispositivos conocidos como cortafuegos. Teniendo como objetivo central el control del tráfico entre una red interna y una externa, se han comentado brevemente los dos tipos que de estos dispositivos existen: de filtrado o IP, y proxy o de aplicación. En relación a esta cuestión también se ha presentado de forma breve una tecnología de gran uso actual: las redes privadas virtuales, núcleo de la cual forma parte el concepto de túnel.

Con fines principalmente de completitud, en el Apartado 12.3.4 se ha abordado la seguridad en sistemas finales. Se ha evidenciado así que de poco sirve robustecer las comunicaciones si los participantes están comprometidos en base a la operación de software malicioso de muy distinta naturaleza en sus equipos: virus, gusanos, troyanos, etc. También en relación a la seguridad en sistemas finales se ha hecho mención a los ataques de denegación de servicio y a la potencial ocurrencia de intrusiones en sistemas.

Por último, y aunque de forma muy breve, también se han discutido algunas cuestiones de actualidad como son la propiedad intelectual, la protección de contenidos y los esfuerzos legislativos realizados en el ámbito de la seguridad. Todo ello constituye una realidad ineludible para los profesionales de la seguridad.

En suma, todo lo expuesto a lo largo del presente capítulo constata la dificultad de mantener adecuadamente operativo un sistema en red, especialmente si tenemos en consideración los numerosos riesgos y vulnerabilidades existentes al respecto. Este campo resulta fundamental si deseamos la confianza de los usuarios en las TIC y con ello el despliegue e implantación generalizada de sistemas y servicios.

EJERCICIOS

- Compare desde un punto de vista funcional los protocolos SNMP y CMIP. ¿Podría concluirse alguna ventaja y desventaja clara de cada uno de ellos frente al otro?
- Acuda a los RFC correspondientes a SNMPv3 a través de <http://www.rfc-editor.org>, y especifique los servicios de seguridad proporcionados y los mecanismos asociados a los mismos.
- Haga lo mismo para CMIP a partir de los documentos ISO/IEC correspondientes (<http://www.iso.org/iso/home/standards.htm>).
- Proponga un esquema de sustitución y otro de permutación y lleve a cabo en cada caso el cifrado del mensaje: «*Nos vemos mañana a las 9 en la estación*».
- Repita el cifrado del mensaje anterior en base a la combinación de los esquemas de sustitución y permutación propuestos.
- Considere un algoritmo de cifrado simétrico de bloques de 32 bits mediante una clave K de 64 bits, donde el esquema de cifrado es

$$C = (P \oplus K_0) \otimes K_1$$

donde C es el texto cifrado, P el texto llano, K_0 los 32 bits más significativos de la clave K , K_1 los 32 bits menos significativos de K , \oplus la función XOR y \otimes la función suma módulo 2^{64} .

A partir de lo anterior, indique la expresión del proceso de descifrado, esto es, la ecuación de P como función de C , K_0 y K_1 .

- Suponga que se produce un error en la transmisión de un bloque cifrado mediante CBC. ¿Qué ocurre en los bloques de texto descifrados en el receptor?
- Realice el cifrado y descifrado RSA en los siguientes casos:
 - $p = 7, q = 11, e = 17, M = 9$.
 - $p = 17, q = 31, e = 7, M = 5$.
- En un esquema de cifrado público RSA interceptamos el mensaje $C = 10$ enviado a un usuario cuya clave pública es $e = 5, n = 37$. ¿Cuál es el mensaje original M ?
- Vaya al RFC correspondiente a HMAC y describa el proceso seguido. ¿Qué ventajas presenta este esquema MAC basado en hash frente, por ejemplo, al uso de un esquema de cifrado simétrico tradicional como DES o AES?
- Contacte con una autoridad de certificación e instale un certificado digital en su ordenador. Explique el proceso e indique el contenido de dicho certificado.
- Considere la siguiente técnica de autenticación de un solo sentido:

$$\begin{aligned} A &\rightarrow B: ID_A \\ B &\rightarrow A: R_1 \\ A &\rightarrow B: R_2 \end{aligned}$$

- Explique el esquema en cuestión y un tipo de ataque al que es susceptible.
- Complete el procedimiento para evitar el ataque mencionado.
- Acuda al RFC correspondiente y detalle el protocolo de autenticación DIAMETER.
- Cuando se usa IPsec para proporcionar tanto autenticación como confidencialidad a través del empleo conjunto de AH y ESP, primero se ejecuta ESP y después AH. ¿Por qué no hacerlo al contrario; primero AH y después ESP?
- Considere los siguientes ataques web y describa cómo pueden evitarse mediante SSL/TLS:

- a) *Ataque de repetición*, donde mensajes de negociación SSL/TLS previos son replicados.
- b) *Ataque de hombre en medio* («man-in-the-middle», MitM), en el que un atacante se interpone durante la fase de intercambio de clave, actuando como el cliente de cara al servidor y como servidor para el cliente.
- c) *SYN flooding*, donde un atacante envía paquetes SYN TCP para requerir una conexión pero no responde al mensaje final de establecimiento de conexión.

16. Comente el significado de cada una de las siguientes entradas en la tabla de un cortafuegos:

Dirección origen	Dirección destino	Puerto origen	Puerto destino	Acción
any	192.169.2.0	any	>1024	deny
192.169.2.3	any	any	80	allow
192.169.2.0	any	any	25	allow

17. Consulte el software *OpenVPN* y describa la funcionalidad que ofrece en relación a la definición de VPN.
18. Considere el siguiente fragmento de código:
- ```
...código legítimo...
if data is Viernes 14;
 inutiliza_el_ordenador ();
...código legítimo...
```
- ¿De qué tipo de software malicioso se trata?
19. Consulte Internet y consiga al menos 2 herramientas software para detectar algunos de los tipos de software malicioso comentados en el capítulo.

## BIBLIOGRAFÍA

- Aboba, B.; Blunk, L.; Vollbrecht, J.; Carlson, J.; Levkowetz: *Extensible Authentication Protocol (EAP)*. RFC 3748. Junio, 2004.
- Anderson, J.P.: *Computer Security Threat Monitoring and Surveillance*. Technical Report, James P. Anderson Company, Fort Washington, Pennsylvania, USA, 1980.
- Blumenthal, U.; Wijnen, B.: *User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*. RFC 2574. Abril, 1999.
- Braden, R.; Clark, D.; Crocker, S.; Huitema, C.: *Report of IAB Workshop on Security in the Internet Architecture*. RFC 1636. Junio, 1994.
- Calhoun, P.; Loughney, J.; Guttman, E.; Zorn, G.; Arkko, J.: *Diameter Base Protocol*. RFC 3588. Septiembre, 2003.
- Callas, J.; Donnerhacke, L.; Finney, H.; Shaw, D.; Thayer, R.: *OpenPGP Message Format*. RFC 4880. Noviembre, 2007.
- Case, J.; Harrington, D.; Presuhn, R.; Wijnen, B.: *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*. RFC 2572. Abril, 1999.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1907. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1906. Enero, 1996.

- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework*. RFC 1908. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1904. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Introduction to Community-based SNMPv2*. RFC 1901. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1905. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1902. Enero, 1996.
- Case, J.; McCloghrie, K.; Rose, M.; Waldbusser, S.: *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC 1903. Enero, 1996.
- Case, J.D.; Fedor, M.; Schoffstall, M.L.; Davin, C.: *Simple Network Management Protocol (SNMP)*. RFC 1157. Mayo, 1990.
- CC: *Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and General Model*. Versión 3.1, Revisión 1. 2006.
- Chokhani, S.; Ford, W.: *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*. RFC 2527. Marzo, 1999.
- Comer, D.E.: *Internetworking with TCP/IP. Volume I: Principles, Protocols and Architecture*. 3.<sup>a</sup> edición. Prentice Hall, 1995.
- Crocker, D.; Hansen, T.; Kucherawy, M.: *DomainKeys Identified Mail (DKIM) Signatures*. RFC 6376. Septiembre, 2011.
- Dierk, T.; Rescorla, E.: *The Transport Layer Security (TLS) Protocol Version 1.1*. RFC 4346. Abril, 2006.
- Dierks, T.; Allen, C.: *The TLS Protocol Version 1.0*. RFC 2246. Enero, 1999.
- Eastlake, D.; Jones, P.: *US Secure Hash Algorithm 1 (SHA1)*. RFC 3174. Septiembre, 2001.
- Frye, R.; Levi, D.; Routhier, S.; Wijnen, B.: *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*. RFC 2576. Marzo, 2000.
- Furnell, S.; Katsikas, S.; López, J.; Patel, A. (editors): *Securing Information and Communications Systems*. Artech House, 2008.
- Hamzeh, K.; Pall, G.; Verthein, W.; Taarud, J.; Little, W.; Zorn, G.: *Point-to-Point Tunneling Protocol (PPTP)*. RFC 2637. Julio, 1999.
- Hanks, S.; Li, T.; Farinacci, D.; Traina, P.: *Generic Routing Encapsulation (GRE)*. RFC 1701. Octubre, 1994.
- Hanks, S.; Li, T.; Farinacci, D.; Traina, P.: *Generic Routing Encapsulation over IPv4 Networks*. RFC 1702. Octubre, 1994.
- Howard, J.D.: *An Analysis of Security Incidents on the Internet, 1989-1995*. PhD Dissertation, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.
- ISO/IEC 10165: *Information Technology -- Open Systems Interconnection -- Structure of Management Information*.
- ISO/IEC 7894-4: *Information Processing Systems -- Open Systems Interconnection -- Basic Reference Model -- Part 4: Management framework*.
- ISO/IEC 9595: *Information Technology -- Open Systems Interconnection -- Common Management Information Service*.
- ISO/IEC 9596-1: *Information Technology —Open Systems Interconnection— Common Management Information Protocol – Part 1: Specification*.
- ITU-T: *Arquitectura de Seguridad de la Interconexión de Sistemas Abiertos para Aplicaciones del CCITT*. Recomendación X.800. 1991.