CAPÍTULO 2
PRINCIPALES PROTOCOLOS DE AUTENTICACIÓN

Hoy en día millones de usuarios necesitan conectar sus computadoras desde su casa a las computadoras de un proveedor para acceder a Internet. También hay muchas personas que necesitan conectarse a una computadora desde casa, pero no quieren hacerlo a través de Internet. La mayoría de estos usuarios disponen de una línea telefónica dedicada o de marcación. La línea telefónica proporciona el enlace físico, pero para controlar y gestionar la transferencia de datos se necesita un protocolo de enlace punto a punto (Figura 2-1).

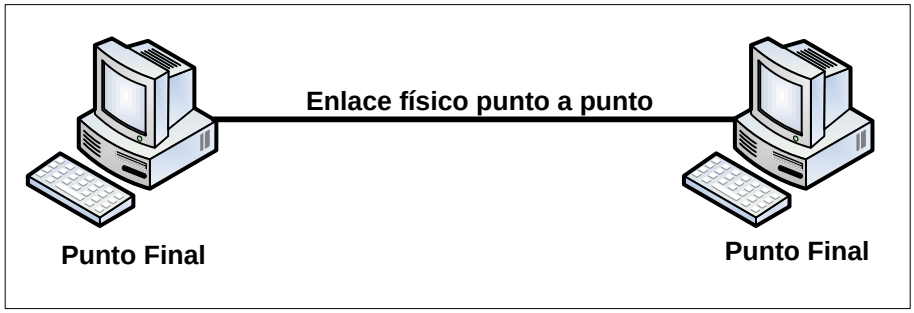

Figura 2-1. Protocolo de enlace punto a punto.

El primer protocolo diseñado para este propósito fue el Protocolo de Internet de línea serie (SLIP, Serial Line Internet Protocol). Sin embargo, SLIP tiene algunas deficiencias: no soporta protocolos diferentes al Protocolo Internet (IP), no permite que la dirección IP sea asignada dinámicamente y sobre todo no soporta la autenticación del usuario. El Protocolo punto a punto (PPP, Point-to-Point Protocol) es un protocolo diseñado para dar respuesta a estas deficiencias [8].

2.1 PROTOCOLO PPP

Las diferentes fases de una conexión PPP se pueden describir utilizando un diagrama de transición de estados como el que se muestra en la Figura 2-2.

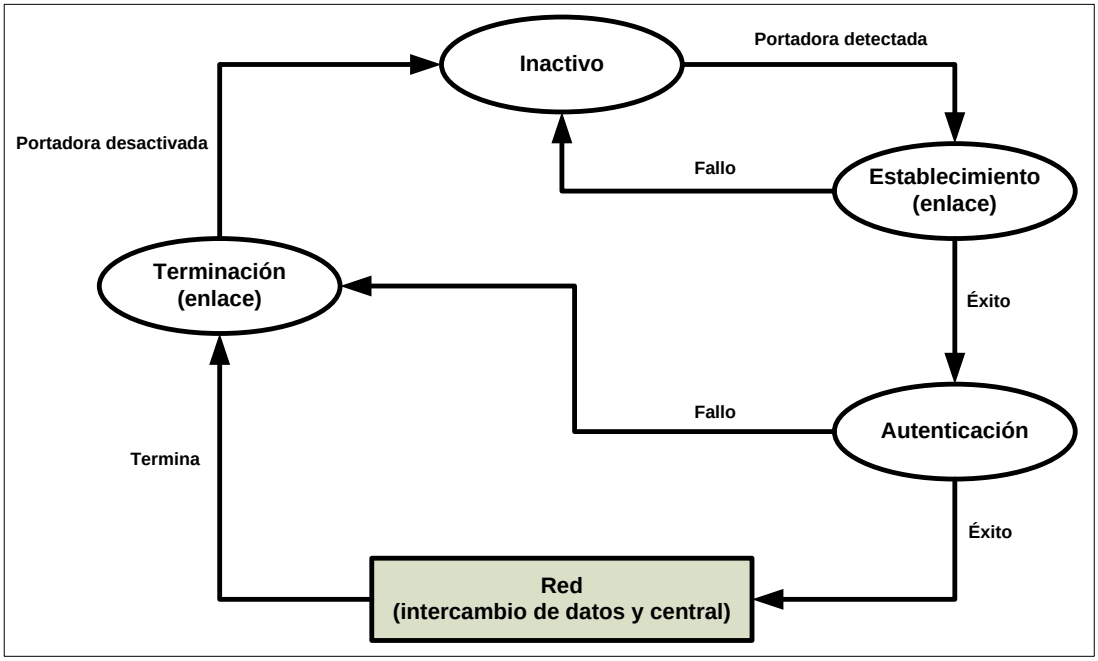

Figura 2-2. Protocolo PPP.

- Estado inactivo. El estado inactivo significa que el enlace no está siendo utilizado. No hay ninguna portadora activa y la línea está tranquila.
- Estado de establecimiento. Cuando uno de los puntos finales comienza la comunicación, la conexión realiza una transición hacia el estado de establecimiento. En este estado se negocian las opciones entre las dos partes. Si la negociación tiene éxito, el sistema se encamina hacia el estado de autenticación (si se necesita autenticación) o directamente al estado de red. Los paquetes *LCP* se utilizan para este propósito. Se pueden intercambiar varios paquetes durante este estado.
- Estado de autenticación. Este estado es opcional (aunque en una red de datos no debería ser omitida para ser siempre segura). Los dos extremos de la comunicación pueden decidir, durante el establecimiento de la conexión, no entrar en este estado. Sin embargo, si lo deciden pueden proceder con una fase de autenticación, enviándose paquetes de autenticación. Si la autenticación tiene éxito, la conexión se dirige al estado de red, en caso contrario pasa al estado de terminación.
- Estado de red. El estado de red constituye el corazón de los estados de transición. Cuando una conexión alcanza este estado, se puede comenzar el intercambio de paquetes de datos y control de usuario. La

conexión permanece en este estado hasta que uno de los extremos finales desea finalizar la conexión.

- Estado de terminación. Cuando una conexión alcanza el estado de terminación, se intercambian varios paquetes; entre los dos extremos para liberar y cerrar el enlace. [8]

Para este caso en particular, el diagrama de transición de estados servirá para enfocar la autenticación. Cabe aclarar que para tener un mayor panorama del protocolo PPP se dará a continuación una breve explicación del mismo.

2.1.1 Niveles del Protocolo PPP

La Figura 2-3 muestra los niveles del protocolo PPP.

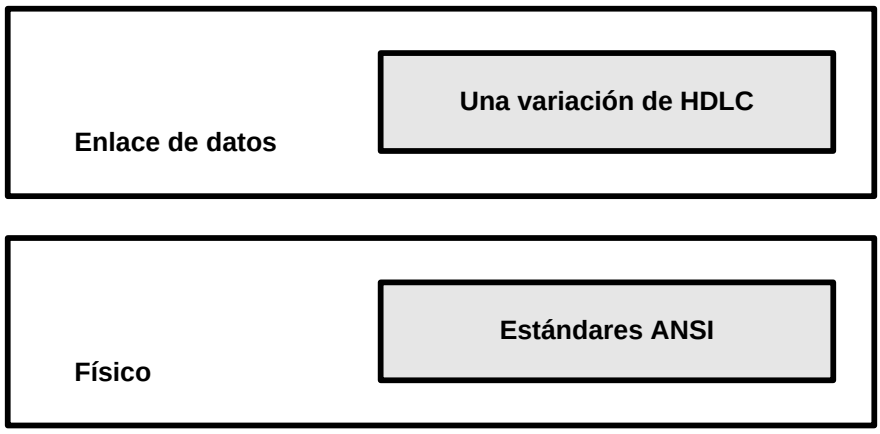

Figura 2-3. Niveles del protocolo PPP.

Este protocolo sólo dispone del nivel físico y de enlace de datos. Esto significa que un protocolo que quiera usar los servicios del protocolo PPP deberían tener los otros niveles (red, transporte y otros).

Nivel físico.

No se ha definido ningún protocolo específico para nivel físico en el protocolo PPP. En su lugar, se ha dejado que el implementador utilice cualquiera disponible. El protocolo PPP soporta cualquiera de los protocolos reconocidos por ANSI.

Nivel de enlace de datos.

En el nivel de enlace de datos, el protocolo PPP emplea una versión del protocolo HDLC. La Figura 2-4 muestra el formato de una trama del protocolo PPP.

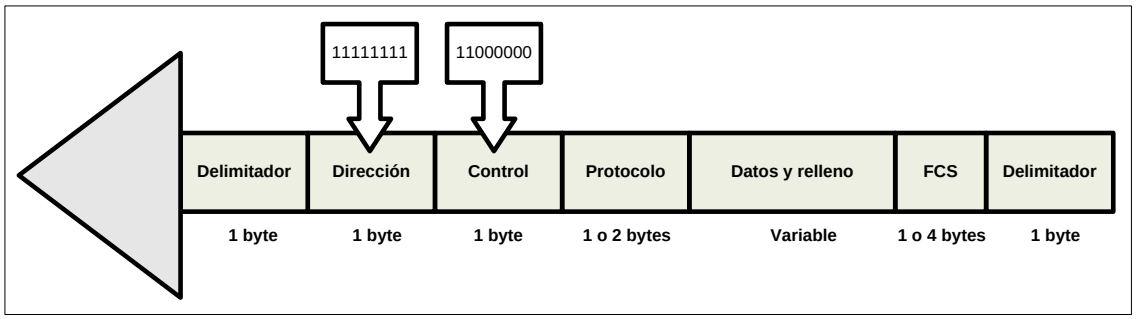

Figura 2-4. Formato de una trama del protocolo PPP.

A continuación se realiza una descripción de los campos de la trama:

- Campo delimitador. El campo delimitador, como en el protocolo HDLC, identifica los límites de una trama del protocolo PPP. Su valor es 01111110.

- Campo de dirección. Debido a que el protocolo PPP se utiliza para una conexión punto a punto, utiliza la dirección de difusión de HDLC, 11111111, para evitar una dirección de enlace de datos en el protocolo.

- Campo de control. El campo de control utiliza el formato de la trama U del protocolo *HDLC*. El valor es 11000000 para mostrar que la trama no contiene ningún número de secuencia y que no hay control de errores ni de flujo.

- Campo de protocolo. El campo de protocolo define qué está transportando el campo de datos: datos de usuario u otra información.

- Campo de datos. Este campo transporta datos de usuario u otra información.

- FCS. El campo de secuencias de comprobación de trama, como en HDLC, es simplemente una suma de comprobación de dos bytes o cuatro bytes [8].

La autenticación juega un papel muy importante en el protocolo PPP y en general en cualquier red de datos, debido a que está diseñado para su empleo en enlaces de

marcación donde la verificación de la identidad de los usuarios es necesaria. La autenticación significa validar la identidad de un usuario que necesita acceder a un conjunto de recursos. El protocolo PPP ha creado dos protocolos de autenticación:

- El protocolo de autenticación de palabra clave (PAP, Password Authentication Protocol).
- El protocolo de autenticación por desafío (CHAP, Challenge Handshake Authentication Protocol).

2.2 PAP

El Protocolo de Autenticación de Palabra Clave (PAP) es un procedimiento de autenticación sencillo que consta de dos etapas:

- El emisor que desea acceder al sistema envía una identificación de autenticación (normalmente el nombre de usuario) y una palabra clave.
- Un sistema comprueba la validez de la identificación y la palabra clave y acepta o deniega la conexión. [8]

Para aquellos sistemas que necesitan más seguridad, PAP no es suficiente: una tercera parte con acceso al enlace puede fácilmente copiar la palabra clave y acceder a los recursos del sistema. La Figura 2-5 muestra la idea del protocolo PAP.

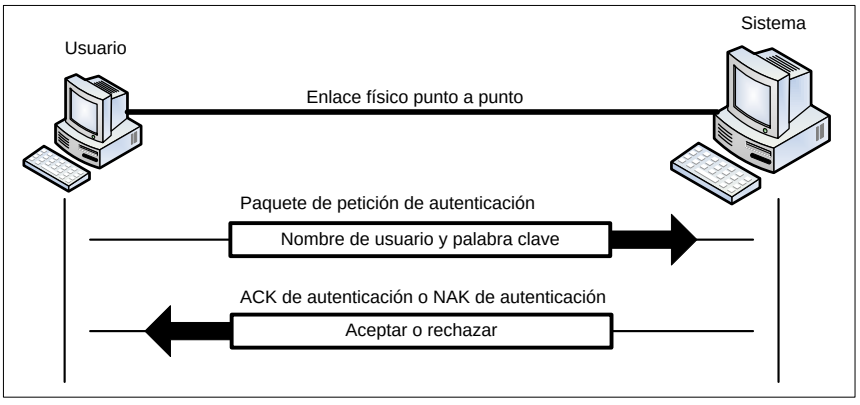

Figura 2-5. Protocolo PAP.

Paquetes de protocolo PAP.

Los paquetes del protocolo PAP se encapsulan en una trama del protocolo PPP. Lo que distingue a un paquete del protocolo PAP de otros paquetes es el valor del campo de protocolo, $C023_{16}$. Hay tres paquetes en el protocolo PAP: petición de autenticación, ACK de autenticación y NAK de autenticación. El primer paquete lo

utiliza el usuario que envía el nombre de usuario y la palabra clave. El segundo lo utiliza el sistema para permitir el acceso. El tercero lo utiliza el sistema para denegar el acceso. En la Figura 2-6 se muestra el formato de los tres paquetes.

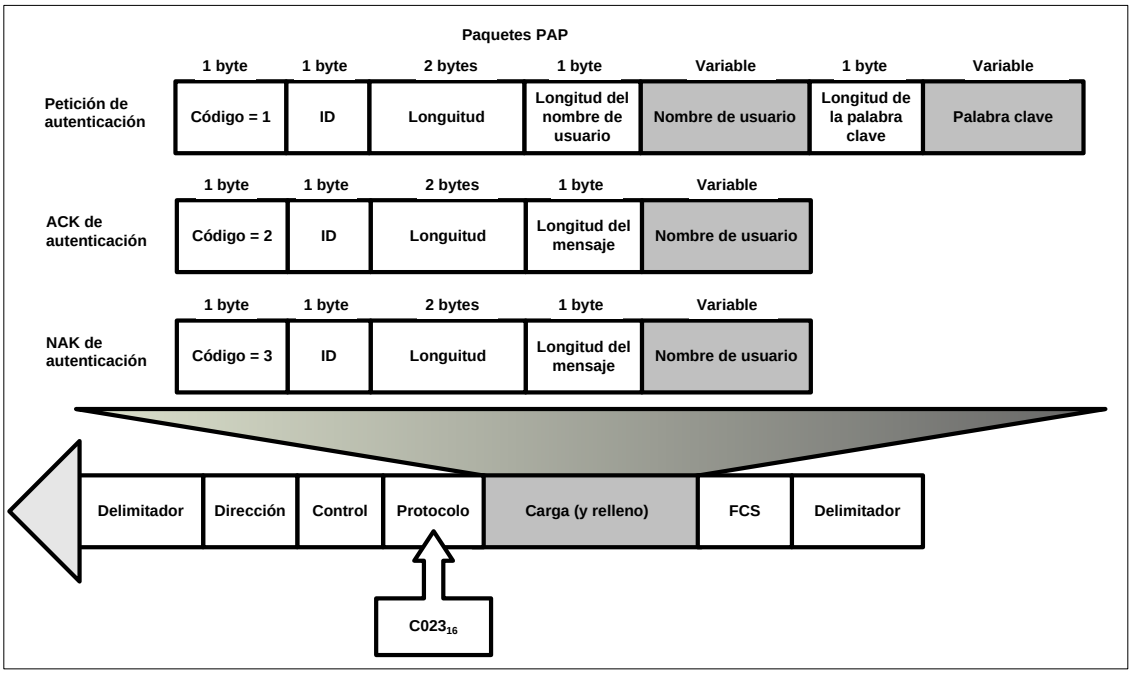

Figura 2-6. Formato de paquetes en el protocolo PAP.

2.3 CHAP

El Protocolo de Autenticación por Desafío (CHAP) es un protocolo de autenticación por desafío de tres fases que ofrece más seguridad que el protocolo PAP. En este método la palabra clave siempre se almacena de forma secreta y nunca se envía por la línea.

- El sistema envía al usuario un paquete de desafío que contiene un valor de desafío, normalmente unos cuantos bytes.

- El usuario aplica una función predefinida que torna el valor del desafío y su propia palabra clave y crea un resultado. El usuario envía el resultado en el paquete de respuesta al sistema.

- El sistema realiza el mismo proceso. Aplica la misma función a la palabra clave del usuario (conocida por el sistema) y el valor del desafío para crear un resultado. Si el resultado creado es el mismo que el

resultado enviado en el paquete de respuesta, se concede el acceso. En caso contrario se deniega [8].

El protocolo CHAP es más seguro que el protocolo PAP, especialmente si el sistema cambia continuamente el valor del desafío. Incluso aunque un intruso capture el valor del reto y el resultado, la palabra clave permanece secreta. La Figura 2-7 muestra la idea de este protocolo.

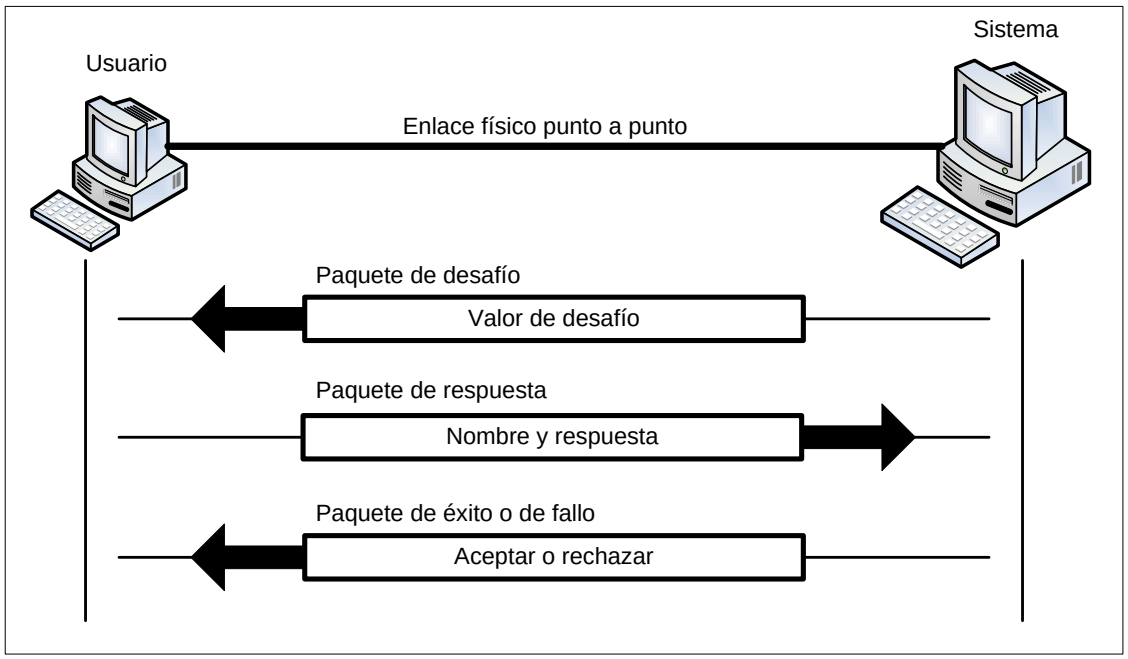

Figura 2-7. Protocolo CHAP.

Paquetes del protocolo CHAP.

Los paquetes de este protocolo se encapsulan en la trama del protocolo PPP. Lo que distingue a un paquete del protocolo CHAP de otros paquetes es el valor del campo de control, $C223_{16}$.

Hay cuatro paquetes en este protocolo: desafío, respuesta, éxito y fallo.

- El primer paquete lo utiliza el sistema para enviar el valor del desafío.
- El segundo lo utiliza el usuario para devolver el resultado del cálculo.
- El tercero lo utiliza el sistema para permitir el acceso al sistema.
- El cuarto lo utiliza el sistema para denegar el acceso al sistema.

En la Figura 2-8 se muestra el formato de estos cuatro paquetes.

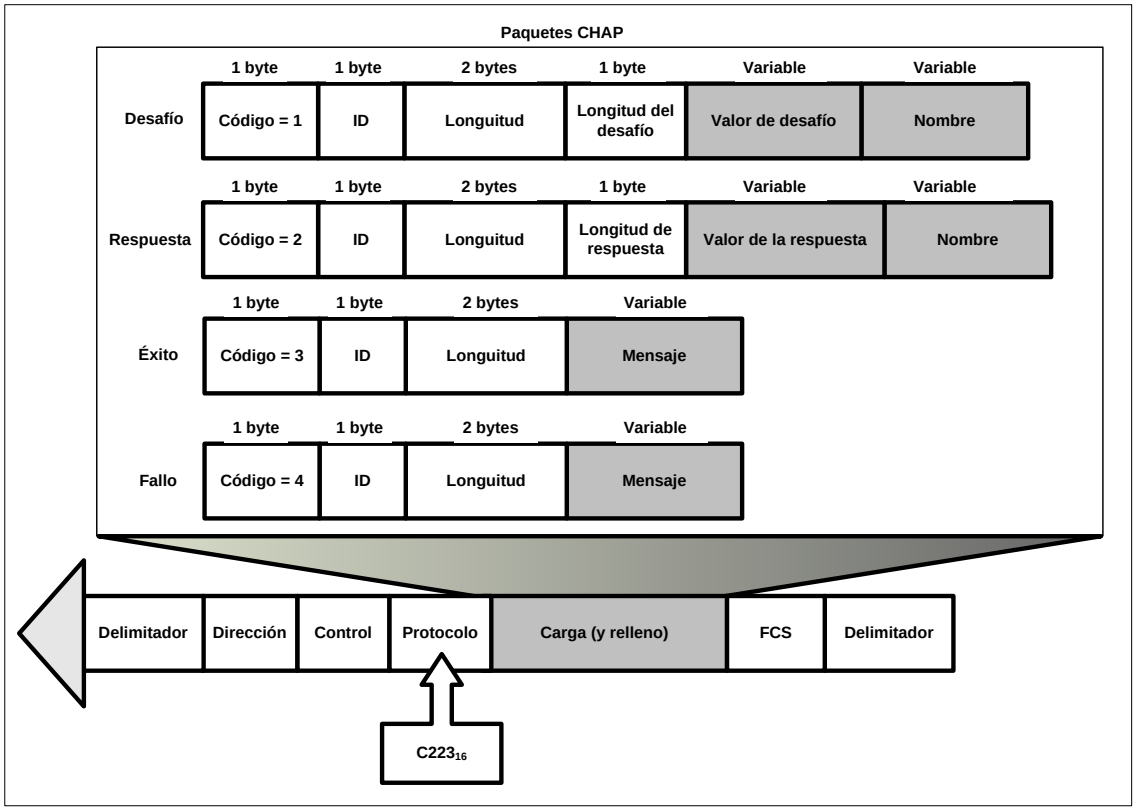

Figura 2-8. Paquetes del protocolo CHAP.

En la mayoría de las redes existen mecanismos de autenticación basados en contraseñas. Esto es, cuando algún servicio necesita autenticación, solicita al cliente un usuario y una contraseña, datos que el cliente envía a través de la red. El problema surge cuando esa transmisión de información viaja por la red en texto plano, pues cualquiera que tenga un *sniffer* y se encuentre en el segmento de red apropiado, por donde viajan dichos datos, será capaz de capturar el usuario y su contraseña.

2.3.1 MS-CHAP

Futuras versiones de CHAP aparecieron desarrolladas de la mano de Microsoft como MS-CHAPv1 y MS-CHAPv2 que mejoran algunos atributos de mensaje además de evitar el almacenamiento de contraseñas en texto plano tanto en el lado del servidor, como en el lado del cliente. Además permitía negociar los cambios de contraseña ante la caducidad de la misma. MS-CHAPv1 utiliza, a diferencia de CHAP, el hash de NT o de

LM para el cálculo del *challenge response* y no la contraseña en texto claro. MS-CHAPv2 elimina el uso del hash de LanManager (LM) por problemas de seguridad y añade autenticación mutua contra el equipo NAS. Sin embargo MS-CHAP tiene hoy en día muchas vulnerabilidades y problemas de seguridad, ya que la entropía de su sistema de encriptación DES es insuficiente (56 bits) y el sistema de cambio de contraseña puede comprometer más aún su seguridad.

2.4 EAP

Extensible Authentication Protocol o EAP es la extensión de autenticación que ha permitido que RADIUS siga implementándose, y es porque cuando ya los demás métodos de autenticación estaban en seria duda de seguridad, se precisaba de un nuevo método que pudiera extender la autenticación hacia un futuro a medio plazo.

Un error habitual es considerar a EAP como un protocolo de autenticación, ya que en realidad no lo es. EAP es un protocolo encargado del transporte, encapsulado y seguridad de la autenticación, y en su interior se encuentran los métodos de autenticación que se desea utilizar. Por ello cuando se habla de autenticación EAP siempre se incluye un sufijo como MD5, MSCHAP, etc. quedando el método de autenticación como EAP-MD5 o EAP-MSCHAP. Existen más de cuarenta métodos de autenticación sobre EAP, que lo hace muy versátil para cualquier tipo de implementación a cualquier escala. La verdadera potencia de EAP es que puede trabajar de forma independiente como protocolo de transporte sobre la capa dos de OSI (capa de enlace), prescindiendo de la dependencia hacia otros protocolos como IP o PPP. Al ser EAP un protocolo de transporte como PPP dispone de sus propios sistemas de control de entrega, retransmisión y de integridad de paquete.

Lo interesante del modelo de EAP es que es un protocolo de autenticación de tipo "pass-through", lo que significa que el NAS o autenticador sólo tiene que iniciar el proceso de autenticación mediante un paquete EAP-Request y a partir de ese momento reencamina todo el proceso de autenticación hacia un servidor de autenticación como RADIUS. Haciéndolo de esta manera, el NAS no tiene por qué realizar el papel de

autenticador, sino que lo deriva hacia el servidor de autenticación que es el que soportará el tipo de autenticación solicitada.

Existen innumerables métodos de autenticación que funcionan sobre EAP, entre los más comunes CHAP (MD5), PAP, TLS, TTLS, PEAP, SIM, AKA, o incluso Kerberos. Algunos de estos tipos de EAP, como EAN-SIM, se utilizan en la telefonía móvil porque implementan soporte para nuevas tecnologías de movilidad como el *roaming* o el protocolo MobileIP. Se pueden agrupar tres tipos principales de métodos de autenticación sobre EAP:

- Métodos basados en claves compartidas. Los métodos basados en claves compartidas han existido siempre y parece seguirán existiendo otros muchos años. El problema de estos consiste en la forma de distribución, transporte o almacenamiento de las credenciales. Dando por hecho que cada usuario debe tener bien guardada su clave en sitio seguro. Algunos métodos basados en claves compartidas son PAP, CHAP, EAP-MD5, EAP-MSCHAPv2, EAP-FAST, EAP-SIM, EAP-AKA...
- Métodos basados en certificados u otros sistemas de claves no compartidas. Estos son los métodos más adecuados para una buena implantación de seguridad, pero también son los más duros de implantar. Los sistemas basados en la generación de una clave inmediata como los *token* son más fiables que los anteriores, pero los certificados PKI o las tarjetas criptográficas ofrecen soluciones más complejas de implementar pero mucho más cómodos y adecuados, una vez funcionales. Algunos de estos métodos son EAP-TLS, EAP-TTLS, EAP-PEAP…
- Métodos basados en características físicas. En la actualidad están apareciendo nuevas implementaciones de seguridad basadas en EAP que utilizan características biométricas como medio de identidad.

También se pueden clasificar los métodos de autenticación sobre EAP en otros dos tipos, basándonos en su sistema de seguridad:

- Métodos no tunelados. Los primeros tipos de autenticación sobre EAP, como EAP-MDS, EAP-MSCHAPv2, EAP-SIM, etc., no son tunelados. El tráfico EAP

completo no es cifrado por el cliente, autenticador y servidor de autenticación. Sólo la información de contraseñas de usuario y algunos otros paquetes delicados se cifran en el interior de los paquetes que circulan por la red. Si se interceptan los paquetes que se generan en el proceso de autenticación, se pueden capturar los hashes[9] para poder obtener las credenciales de los usuarios. Existen otros tipos de EAP no tunelados como EAP-OTP y EAP-GTC que utilizan sistemas como Tokens generadores de claves instantáneas de un sólo uso.

- Métodos tunelados. El sistema de tunelamiento de EAP es principalmente EAP-TLS y sus sucesores que utilizan un sistema criptográfico simétrico/asimétrico para la encriptación completa del tráfico durante el proceso de autenticación, autorización y contabilidad. Este cifrado asimétrico se sustenta de certificados *X.509* que son intercambiados entre el servidor y el cliente y utiliza un tunelamiento similar a SSL. De esta manera se incrementa la seguridad del canal de forma bastante robusta contra la interceptación de tráfico o los ataques de *MiTM* Man in The Middle (hombre en medio). Algunos de estos métodos son EAP-PEAP y EAP-TTLS.

Cada uno de los tipos de EAP dispone de un identificador de tipo de EAP para establecer el método en las conversaciones EAP. El *RFC* base que define EAP es el RFC 3748.

2.4.1 EAP-MD5

EAP-MD5 es la primera versión, la más simple y, por lo tanto, la más insegura de EAP. El método utilizado desarrollado por RSA es análogo a CHAP. Como su nombre indica, utiliza el algoritmo MD5 para obtener un hash de la contraseña de usuario. Es un método de autenticación de una sola dirección (el servidor autentica al cliente pero no viceversa). Se considera el más inseguro de los tipos de EAP, ya que no incorpora ningún sistema de encriptación de los paquetes, que circulan en texto plano. No incorpora la característica de generar claves de sesión para el cifrado de protocolos como WEP o WPA como lo hagan TLS, TTLS o PEAP. Sólo se debe utilizar en canales de autenticación difíciles de interceptar como 802.1X para redes cableadas y con mucha precaución.

2.4.2 EAP-TLS

Los métodos de EAP basados en TLS (Transport Layer Security) se apoyan en PKI (Infraestructura de clave pública) para el uso de SSL y certificados X.509. Cabe aclarar TLS y SSL funcionan en capas diferentes del modelo OSI, TLS trabaja en la capa dos (enlace) y SSL sobre la capa cinco, pero su modelo basado en PKI es muy similar. Tras el intercambio y la comprobación de los certificados y confianzas, se establece un tunelamiento TLS (outer-tunnel) para el envío al cliente de una clave de cifrado que se utilizará en las consecutivas comunicaciones. Este primer intercambio de credenciales para el establecimiento del túnel TLS se conoce como outer-tunnel y produce un flujo de paquetes entre el servidor y el cliente. Apoyándose en la seguridad proporcionada por este sistema de certificados, se puede utilizar con bastante tranquilidad dentro de este túnel (inner-tunnel) cualquier otro sistema de autenticación más inseguro como CHAP, PAP u otros similares. Este segundo intercambio produce un segundo flujo de paquetes entre el servidor y el cliente. Se puede entender que se producen dos procesos de autenticación independientes entre el servidor de autenticación y el cliente, y así lo gestionan algunos servidores como FreeRADIUS.

Al enviar el cliente al servidor la solicitud de acceso con su nombre de usuario (que en algunos tipos como TTLS puede, y debe, ser anónimo), el servidor responde enviando su certificado de servidor para que el cliente lo verifique. Tras esa comprobación de la confianza sobre el servidor, el cliente realiza el envío de su certificado al servidor.

Pasado ese primer proceso de verificaciones, se establece un canal seguro mediante TLS (SSL) para que, si procede, se intercambien credenciales u otros métodos de autenticación y para finalmente acabar entregando al cliente y al equipo NAS una clave única a fin de que pueda establecer una sesión segura de comunicaciones.

EAP-TLS es un método de autenticación muy seguro, pero que requiere de una infraestructura medianamente compleja para su puesta en funcionamiento, por ese motivo seguramente su difusión está resultando un poco lento. Este protocolo es un protocolo de autenticación mutua, lo que significa, que tanto el cliente debe autenticarse

contra el servidor como el servidor contra el cliente. Esto hace que se necesite de dos certificados X.509, uno para el servidor de autenticación y otro para el cliente. De esa manera se evitan los ataques del tipo MiTM que pueden provocar que un cliente entregue sus credenciales a un falso servidor.

EAP-TLS necesita que se almacenen los certificados de cliente en el equipo donde reside el cliente. Esto puede también conllevar problemas de seguridad, ya que la parte principal de ese certificado, que es la clave privada, podría ser robada del equipo en cuestión si no se almacena cifrada, y esto suele ser así en algunas implementaciones. Para evitar esto, se pueden utilizar *smartcards* o tarjetas criptográficas que protegen mediante un procesador de cifrado y un pin, la clave privada del cliente. La única posibilidad de robar la identidad del usuario es robar su tarjeta y conocer su PIN pero aun así el sistema PKI se apoya en la revocación de certificados y al denunciar el usuario la pérdida el certificado de su tarjeta puede ser inmediatamente revocado y la tarjeta quedará inservible para esta red. Otra opción sería que los propios programas cliente cifraran los certificados al guardarlos y/o solicitarán un PIN al utilizarlos.

Un fallo de seguridad intrínseco al EAP-TLS es la forma en la que se intercambian los datos de identidad (User-Name) en texto plano, previamente al intercambio de certificados, de tal manera que interceptando este tráfico se pueden recopilar los nombres de usuario de aquellos que se estén autenticando. Si bien los nombres de usuario no bastan para realizar un ataque contra la red, es un dato que ayudará bastante a la enumeración del sistema.

EAP-TLS no permitía la reconexión rápida mediante recuperación del túnel TLS, aunque en el último RFC 5216 de marzo de 2008 ya se comienza a implementar. EAP-TLS es un estándar abierto (no propietario) y el RFC que lo define es el RFC 5216 que deja obsoleto al RFC 2716 de IETF.

2.4.3 EAP-TTLS

EAP-TTLS es una extensión de EAP-TLS, desarrollada por Funk y Certicom para simplificar la implantación de EAP-TLS. Su identificador de tipo EAP es el 21. Este

método no se basa en la autenticación mutua previa mediante dos certificados, ya que sólo el servidor debe disponer de un certificado X.509. Esto dificulta igualmente que se produzca un ataque MiTM, puesto que el cliente estará igualmente seguro de la identidad del servidor contra el que se autentica. No obstante, en TTLS el cliente puede utilizar opcionalmente un certificado X.509 si lo prefiere.

El sistema TTLS implementa un sistema de creación de dos túneles de seguridad respectivamente. El primer túnel TLS se crea para el intercambio de credenciales y el segundo para el traspaso de la clave de cifrado de sesión, con la que equipos NAS como AP cifran el tráfico con la estación que se conecta. Todo el tráfico circula encriptado, incluso los mensajes de EAP Success y Failure (Autenticación exitosa o fallida).

La secuencia comienza una vez que el cliente comprueba el certificado del servidor, con lo que se inicia un túnel o canal cifrado para el traspaso de las credenciales del cliente al servidor de forma segura. Utilizando este canal seguro se puede, ya dentro del túnel, utilizar otros medios de autenticación más primitivos como PAP, CHAP, MS-CHAP y esto no supondrá un riesgo. Este es un sistema de autenticación mixta, porque se basa en la autenticación mutua pero utiliza claves compartidas.

Esta autenticación inicial basada en un sólo certificado de servidor simplifica de forma importante la implementación de esta seguridad al no tener que generar certificados para cada cliente nuevo que desee conectarse a la red y por tanto no nos obliga a disponer de una Infraestructura de Clave pública o PKI activa.

Otra gran ventaja de TTLS es que utiliza un sistema de atributos similar al nativo de RADIUS llamado AVP (Attribute Value Pair o par de atributo/valor), con una notación parecida a la de RADIUS. El intercambio de atributos y valores se realiza en el canal cifrado TLS. Su principal valor es el de poder extender el protocolo mediante nuevas implementaciones de atributos, a diferencia de EAP-PEAP que no utiliza el sistema AVP sino un intercambio de mensajes EAP.

En cuanto al fallo de seguridad de EAP-TLS (captura de nombres de usuario), también queda solucionado al enviarse un nombre de usuario anónimo al inicio de la autenticación, diferente al nombre de usuario real que se utiliza en el traspaso de

credenciales CHAP, PAP... para el acceso. Por ello su seguridad es muy robusta. El tráfico de una sesión de autenticación EAP es importante, si se debe autenticar de nuevo a un cliente cada poco tiempo se genera demasiado tráfico. Por eso, EAP-TTLS permite la reconexión rápida mediante el parámetro "TLS session resume" que continúa la última sesión tunelada TLS evitando gran cantidad de tráfico. EAP-TLS no dispone inicialmente de la función fast reconnect. La función fast reconnect usando el "TLS resume" puede causar problemas en infraestructuras con varios servidores RADIUS, ya que la clave TLS cacheada para esa conexión no estaría disponible si la validación se realizara contra otro servidor de la cadena, por disponer de roaming o control dinámico del tráfico. Se discute también sobre si el uso de este parámetro puede permitir ataques MiTM.

2.4.4 EAP-PEAP

EAP-TTLS y EAP-PEAP prácticamente son protocolos de autenticación iguales. EAP-PEAP (Protected Extensible Protocol) fue desarrollado por Cisco, Microsoft y RSA y por eso se encuentra en los productos de estos fabricantes de forma más o menos nativa.

EAP-PEAP se basa en un sólo certificado de servidor como TTLS y soporta como métodos de autenticación MS-CHAPv2 y GTC (Generic Token Card). Si se emplea el método MS-CHAPv2 se le conoce como PEAPv0 que es prácticamente el único incluido en los sistemas operativos como Windows y si utilizamos GTC se le conoce como PEAPv1 que no tiene soporte nativo en ningún SO. Es por esto, y por intereses comerciales, que Microsoft conoce PEAPv0 como PEAP simplemente y tras la salida al mercado del estándar EAP-TTLS no tiene pensado dar soporte a la v1.

Lo que para algunos administradores puede suponer una ventaja de PEAP es que al haber sido en parte desarrollado por Microsoft posee soporte nativo para su sistema operativo a partir de Windows XP. Este soporte nativo, que forma parte de Windows Server 2003, incluye a PEAP en sus políticas de grupo, facilitando la divulgación de certificados y de confianzas de forma automatizada para toda la red basada en AD. Lo que puede ser una ventaja para la implementación de PEAP en sistemas Microsoft y Cisco puede ser una desventaja para otros sistemas por la falta de soporte que tiene

PEAP en ellos. Hay una muy larga discusión sobre si es mejor utilizar TTLS o PEAP. Ambos son dos productos con un nivel de seguridad muy adecuado, con el tiempo y los hackers, se verá cuál es más seguro o apropiado.

EAP-PEAP dispone también de la función fase reconnect para el restablecimiento de sesiones TLS.

2.5 KERBEROS

Kerberos se desarrolló originalmente para sistemas basados en Unix y se define en el RFC 1510. Es una infraestructura de autenticación utilizada para garantizar la identidad de los usuarios y sistemas en una red. Kerberos tiene como primer objetivo el asegurar las contraseñas para que nunca sean enviadas por la red sin ser previamente encriptadas. Kerberos es el protocolo que más a menudo es asociado con el marco AAA. La versión actual de Kerberos es la 5.0 y actualmente hay clientes de Kerberos para casi cualquier sistema operativo.

Para entender el funcionamiento de Kerberos, lo primero es familiarizarse con su terminología. Lo términos a emplear son los siguientes:

- Caché credencial o archivo de ticket: Fichero que contiene las claves para encriptar las comunicaciones entre el usuario y varios servicios de red.
- Centro de distribución de claves (KDC): Servicios que emite ticket Kerberos, que habitualmente se ejecutan en el mismo host que un Ticket Graming Server.
- Clave: Datos usados para encriptar o desencriptar otros datos. Los datos encriptados no pueden desencriptarse sin una clave correcta.
- Cliente: Usuario, host o aplicación que puede obtener un ticket desde Kerberos
- Dominio: Red que usa Kerberos compuesta de uno o varios servidores (también conocidos como KDC) y un número potencial de clientes. También conocido como Reino o *Realm*.
- Keytab: Fichero que incluye una lista desencriptada de los principal y sus claves.

- Principal: Usuario o servicio que puede autenticar mediante el uso de Kerberos. Un nombre de principal está en el formato siguiente: root[/instance]@REALM

Para un usuario típico, el *root* es igual a su ID de *login*. El *instance* es opcional. Si el principal tiene un instance, se separa del root con (“/”). Una cadena vacía (“) es un instance válido (que difiere del instance por defecto NULL), pero usarlo puede ser confuso. Todos los principal de un dominio tienen su propia clave, que se deriva de su contraseña (para usuarios) o aleatoriamente (para servicios).

- Servicio: Programa al que se accede en la red.
- Texto cifrado: Datos encriptados.
- Texto sin retocar: Datos no encriptados.
- Ticket: Grupo temporal de credenciales electrónicas que verifican la identidad de un cliente para un servicio particular.
- Ticket Granting Service (TGS): Emite ticket para un servicio deseado que usa el usuario para ganar acceso al servicio. El TGS se ejecuta en el mismo host que KDC.
- Ticket Granting Ticket (TGT): Ticket especial que permite al cliente obtener tickets adicionales sin aplicarlos desde KDC.

Kerberos se basa en una combinación de clave de cifrado y protocolos criptográficos para garantizar la autenticación de los usuarios. El proceso se indica en la Figura 2-9, es bastante simple, un administrador de la red se ha creado un servidor de autenticación, conocido como Ticket Granting Server (TGS). Uno o varios reinos (por lo general, los dominios) se crean en el TCG. Un usuario solicita el acceso a un determinado ámbito debe obtener un boleto de la TGS, mediante la autenticación en el servidor.

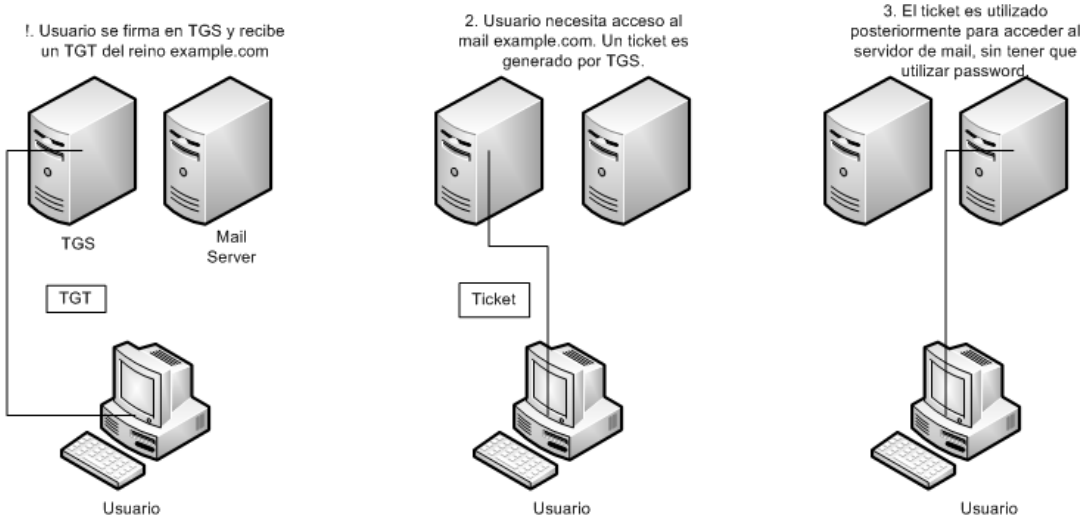

Figura 2-9. Proceso de autenticación Kerberos.

Cuando un usuario se autentica en contra de la TGS un ticket es emitido. Ticket Granting Ticket (TGT), se utiliza en cualquier momento que las necesidades de los usuarios a el acceso a un servicio o un dispositivo en el ámbito que requiere autenticación. El usuario presenta el TGT a la TGS, que emite un ticket para ese dispositivo o servicio.

El usuario sólo tiene que autenticarse en el TGS una vez durante una sesión. El resto del tiempo, el TGS utiliza la información en el TGT para conceder el acceso. Kerberos crea una clave basada en la contraseña del usuario para cifrar el TGT usando el paquete de cifrado de datos estándar (DES). Las versiones modernas utilizan cifrado 3DES. El usuario descifra el paquete y utiliza la entrada para acceder al servicio o el dispositivo.

Kerberos versión 4.0 se han encontrado varios fallos de seguridad, especialmente en el ámbito de la autenticación de contraseña. Es especialmente susceptible a ataques de diccionario, ya que sólo se utiliza una contraseña de base, una función de hash como manera de generar la codificación. Kerberos 5.0 evita este problema utilizando la contraseña y el campo para generar la encriptación. Esto hace que sea mucho más difícil para un atacante lanzar un ataque de contraseña.

A continuación se detalla el modo del funcionamiento del sistema Kerberos, cabe aclarar que el principal problema para Kerberos consiste en cómo usar contraseñas para autenticarse sin enviarlas a la red. En una red “kerberizada” la base de datos de

Kerberos contiene sus claves (para los usuarios sus claves derivan de sus contraseñas). La base de datos Kerberos también contiene claves para todos los servicios de la red.

Cuando un usuario, en una red que utiliza el sistema Kerberos, se registra en su estación de trabajo, su principal se envía al Key Distribution Center (KDC) como una demanda para un Ticket Granting Ticket (TGT). Esta demanda puede ser enviada por el programa login (para que sea transparente al usuario) o puede ser enviada por el programa kinit después de que el usuario se registre.

El KDC verifica el principal en su base de datos. Si lo encuentra, el KDC crea un TGT, lo encripta usando las claves del usuario y lo devuelve al usuario.

El programa login o kinit desencripta el TGP utilizando las claves del usuario. El TGT, que caduca después de un cierto periodo de tiempo, es almacenado en su caché de credenciales. Sólo se puede usar durante un cierto periodo de tiempo, que suele ser ocho horas (a diferencia de una contraseña comprometida, que puede usarse hasta que se cambie). El usuario no tiene que introducir su contraseña otra vez hasta que el TGT caduca o se desconecta y vuelve a conectarse.

Cuando el usuario necesita acceder a un servicio de red, el cliente usa el TGT para pedir un ticket para utilizar el servicio Ticket Granting Service (TGS), que se ejecuta en el KDC. El TGS emite un ticket por el servicio deseado que se usa para autenticar el usuario.

Kerberos depende de ciertos servicios de red para trabajar correctamente. Primero, Kerberos necesita una sincronización de reloj entre las computadoras y su red. Si no se ha configurado un programa de sincronización de reloj para la red, será necesaria su instalación. Ya que ciertos aspectos de Kerberos se apoyan en el DNS (Domain Name System), las entradas DNS y los host en la red deben estar configurados correctamente [10].

Algunos servidores de autenticación basados en Kerberos son:

- Windows Server 2008
- KERBEROS MIT

- KERBEROS Heimdal

2.6 RADIUS

El protocolo de RADIUS fue desarrollado originalmente para su uso con el acceso telefónico a redes. Aunque todavía es principalmente utilizada para autenticar las cuentas de dial-up, se ha convertido en una herramienta popular para la autenticación de otros dispositivos de red. Este crecimiento tiene sentido, ya que muchos administradores no les gusta la idea de mantener un servidor AAA para routers y switches, y otro para marcar a los usuarios.

RADIUS opera en el puerto 1812, el transporte sobre UDP, y se especifica en el RFC 2865. El protocolo original RADIUS incluyó el apoyo para el Punto-to-Point Protocol (PPP) y el inicio de sesión de Unix, los proveedores han incorporado soporte para otros tipos de accesos a sus versiones de RADIUS.

La autenticación RADIUS se maneja mediante el intercambio de claves secretas enviadas a través de paquetes de texto plano, sin embargo, las contraseñas son encriptadas utilizando MD5. Dado que se envían los paquetes de RADIUS mediante UDP, existen varios mecanismos de seguridad a fin de ayudar a garantizar que los datos llegan a su destino. Un cliente RADIUS puede ser configurado para reenviar las transmisiones a intervalos predefinidos, hasta que se recibe una respuesta, o puede ser ajustado a prueba de fallos a un segundo o tercer servidor RADIUS en el caso de un fracaso. La Figura 2-10 describe el proceso para el éxito de la autenticación RADIUS. El router tiene software, conocido como un cliente de RADIUS, que interactúa con el servidor RADIUS cuando intenta autenticar a los usuarios. El servidor RADIUS podrá remitir la solicitud a otro servidor RADIUS, o hacer una consulta sobre un Lightweight Directory Access Protocol (LDAP) para autenticar el servidor de información. En los casos en que el servidor RADIUS autentica contra otro, el servidor RADIUS actúa como el cliente y envía la solicitud de autenticación en un formato codificado.

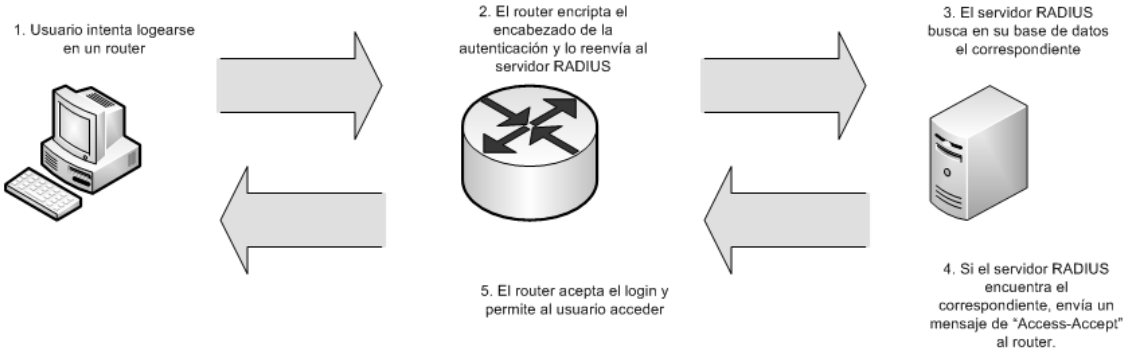

Figura 2-10. Proceso de autenticación RADIUS.

Si es compatible, el servidor RADIUS puede expedir una nueva solicitud desafío-respuesta de autenticación de usuario. Si ese es el caso, el servidor RADIUS emitirá un "desafío-acceso" a petición del cliente, que a su vez lo transmite al usuario. El usuario debe responder con una respuesta adecuada al desafío.

Si el servidor RADIUS no es capaz de localizar al usuario, o las contraseñas no coinciden, el servidor RADIUS cuestiona acceso-rechazar, junto con un mensaje de error, para el cliente, que lo transmite al usuario final. Después del mensaje de acceso-rechazar enviado, es descartado.

Porque las transacciones de RADIUS se realizan a través de UDP, no hay una confirmación entre el cliente y el servidor que el éxito de una solicitud se ha hecho hasta que el servidor responde al cliente. Con el fin de resolver este problema un paquete de acceso-petición acompaña a cada solicitud a un servidor RADIUS. El acceso-petición contiene el nombre de usuario y la contraseña del usuario, así como la identificación del cliente y el puerto cliente al que el usuario está intentando acceder. El cliente mantiene esta información y contando hacia atrás y comienza. Si la respuesta no se recibe desde el servidor RADIUS en un determinado periodo de tiempo, el cliente o bien reenviar la solicitud, o prueba un servidor RADIUS secundario en función de cómo ha configurado el cliente el administrador de RADIUS.

La capacidad para tratar de autentificar los inicios de sesión varias veces contra el mismo servidor o en contra de varios servidores RADIUS ayuda a hacer un protocolo robusto que es muy resistente a los problemas de red. De hecho, RADIUS tiene que ser

resistentes, ya que su uso primario es en el acceso telefónico a redes. Earthlink, MSN, y AT&T, todos utilizan RADIUS para la autenticación de acceso telefónico a sus redes. Ellos tienen millones de usuarios de marcación en sus redes al mismo tiempo, si RADIUS no es robusto, estos proveedores experimentaran frecuentes cortes.

RADIUS actúa sobre la capa 7 del modelo OSI, ya que es precisamente en la capa de Aplicación donde se definen y engloban los protocolos que utilizan las aplicaciones para el intercambio de datos. El usuario no suele manejar directamente estos protocolos de intercambio de datos, sino a través de alguna aplicación que a su vez maneja este lenguaje. En esta capa se incluyen una creciente cantidad de protocolos que se dedican a muchas y muy diferentes funciones, entre ellos algunos protocolos de autenticación como RADIUS y Kerberos.

Algunos servidores de autenticación basados en Radius son:

- FreeRADIUS
- GNU RADIUS
- OpenRADIUS
- Cistron RADIUS
- BSDRadius
- TekRADIUS
- WinRADIUS
- Windows Server 2008

2.7 DIAMETER

Tras la creación del grupo de trabajo en la IETF en 1995 dedicado a crear el RFC correspondiente de RADIUS, se pensó en crear un nuevo código limpio y mejorado de RADIUS que se llamaría RADIUS v. 2. Pero la IETF no permitió esta maniobra, debido a que RADIUS todavía no había sido ratificado en una RFC funcional y corregida, y no se debía crear otro estándar hasta que el primero hubiera sido publicado. Por ello, el nombre que recibió este nuevo estándar no pudo ser RADIUS v2 y se optó por Diameter (dos veces el radio o como definieron sus creadores “twicce as good as RADIUS”). Diameter fue diseñado en 1996 por Pat Calhoun de la compañía Black Storm Networks.

El RFC que regula Diameter pasó a ser el RFC-3588 (“Diameter Base Protocol”), y posteriormente se han ido creando diferentes RFC que regulan su aplicación en MobileIP, EAP, etc.

Diameter es un protocolo de segunda generación, cien por cien basado en AAA. Una de las premisas más importantes en su diseño fue que tenía que ser compatible con RADIUS (“legacy compatible”) para que pudiera sumir todas las instalaciones en forma de migración. Algunas de la mejoras que incorpora son: la sustitución de UDP por TCP y SCTP mejorando el control de errores en la transmisión, el uso de tunelación mediante IPSEC o TLS, y su cambio de modelo hacia peer to peer en vez de cliente-servidor, con lo que un servidor puede realizar consultas hacia un cliente, permitiendo sesiones dinámicas.

Diameter firma los mensajes mediante un código de tiempo, que impide duplicidades en la recepción de respuestas simultaneas, además de usar cifrado basado en certificados y firma digital. Diameter se apoya en un módulo criptográfico llamado CMS (Criptographic Message Syntax) integrado en su plataforma, que se encarga del cifrado de todos los mensajes. Diameter da soporte al nuevo estándar de gestión de NAS llamado NASREQ. Diameter permite definir cadena de Proxy para los envíos de mensajes.

2.8 TACACS+

TACACS+ es un protocolo similar a RADIUS que fue desarrollado por Cisco Systems. TACACS+ se inspira en dos protocolos depreciados, TACACS y TACACS ampliada (XTACACS), TACACS+ es incompatible tanto con TACACS y XTACACS. A causa de graves fallas de seguridad en el TACACS XTACACS y diseños, se recomienda que no se utilicen en favor del modelo de TACACS+.

Mientras que TACACS+ fue desarrollado por Cisco, el pliego de condiciones del protocolo TACACS+ se ha puesto a disposición del público. Otros proveedores de redes, incluyendo Extreme Networks y Foundry Networks, han incorporado TACACS+ en sus productos.

TACACS+, mientras que realiza la misma función como radio, sus orígenes son

diferentes. TACACS + fue desarrollado originalmente como un protocolo para el control de la AAA para los dispositivos de red, por lo que la arquitectura es diferente a la de RADIUS, que fue desarrollado originalmente para el acceso telefónico a redes.

TACACS+ opera a través de TCP, en lugar de UDP, y utiliza el puerto 49 por defecto, aunque TACACS+ se puede configurar para usar cualquier puerto de un administrador de red deseos. También a diferencia de RADIUS, TACACS+ encripta todos los paquetes de datos, no sólo la contraseña.

El protocolo TACACS+ es similar a RADIUS en la forma en que autentifica a los usuarios. Un usuario inicia sesión en un router o switch de interfaz que tenga TACACS+ habilitado. El dispositivo de la red obtiene el nombre de usuario y contraseña del servidor TACACS+ que está configurado para la interfaz y se lo pasa al usuario intentar autenticarse. El usuario introduce el nombre de usuario y contraseña, que se cifra y se pasó de la red al dispositivo de servidor de TACACS+.

El servidor TACACS+ enviará una de las cuatro respuestas a la red de dispositivo: ACEPATAR, RECHARZAR, ERROR, o CONTINUAR. ACEPTAR una respuesta se indica que la autenticación se ha realizado correctamente, y puede comenzar el período de sesiones. Adicional si se necesita información de autenticación, el usuario se le solicita que en este momento.

Un mensaje RECHAZAR indica que la autenticación fallo. El usuario tendrá que volver a introducir la contraseña, o la sesión se desconectará. Este comportamiento varía en función de la TACACS+ demonio.

Si es un ERROR, entonces hay un problema con el servidor de TACACS+, el dispositivo de red de la consulta, o un problema con la red. Si el dispositivo de red recibe el mensaje de error que se trate, ya sea nuevo o que intentará un suplente TACACS + servidor, dependiendo de cómo el administrador de la red se ha configurado.

CONTINUAR la respuesta se enviará cuando la autenticación es satisfactoria, pero se necesita información adicional.

TACACS+ permite múltiples tipos de autenticación. Autenticación de contraseña es el usado más comúnmente, la forma más básica de autenticación. Sin embargo, un administrador de red no se limita a la contraseña de autenticación, de hecho, cualquier forma de autenticación que cuenta con el apoyo de la TACACS + software puede ser utilizado. Además, las múltiples formas de autenticación puede ser necesaria, siempre y cuando el elegido TACACS + software apoya. Por ejemplo, si la contraseña de autenticación no es suficiente, un administrador puede configurar TACACS + para exigir un nombre de usuario / contraseña y una clave RSA para obtener acceso. El usuario se autentique primero por el envío el nombre de usuario y contraseña. En caso de que tuviera éxito, el servidor TACACS + ACEPTAR enviar un mensaje, seguido por solicitud de un nuevo desafío, para la clave RSA. Cuando los dos niveles de autenticación que se hayan completado, el usuario se puede acceder al router.

Algunos servidores de autenticación basados en TACACS+ son:

- ClearBox TACACS+ RADIUS Server

2.9 LDAP

LDAP (Lightweight Directory Access Protocol) es un protocolo de tipo cliente-servidor, encargado de almacenar y mantener todo tipo de información concerniente a una organización; desde nombres de usuario, contraseñas, datos de usuarios, credenciales de equipos, certificados, permisos y directivas de acceso a recursos o aplicaciones, cuentas de correo, etc. Es lo que se conoce como un servicio de directorio, que en la mayor parte de las ocasiones se almacena en formato de base de datos. En grandes organizaciones, su labor, además de almacenar información para todos los procesos de autenticación y autorización, es la de comportarse como un verdadero servidor de directorio, donde se localizan datos como teléfonos, direcciones, pertenencia a departamento, datos de contacto VoIP, correo electrónico, etc.: como unas páginas blancas de la organización. Su implementación se realiza en la capa de aplicación del sistema OSI.

Este protocolo es la adaptación y puesta en práctica del estándar X.500, que funciona sobre el protocolo TCP/IP y actualmente está en la versión 3. No hay que confundir ni relacionar a LDAP con una base de datos relacional; LDAP es un

protocolo que regula el acceso a los datos y su formato de almacenamiento. Su diseño está especialmente optimizado para la lectura, a fin de poderse ejecutar miles de consulta por minuto. La universalidad y estandarización de este protocolo ha hecho que, a lo largo de todos los años que lleva funcionando, muchos programas lo utilicen para acceder o almacenar directamente la información que necesitan. Esta información que se almacena en el directorio es fácilmente replicable entre servidores locales o remotos, para el mantenimiento de estructuras redundantes.

La información almacenada en LDAP utiliza un formato similar a RADIUS y a otras muchas implementaciones: AVP (Par atributo-valor), por ejemplo, o=unam.mx para establecer el nombre de organización (o) como unam.mx. Se utiliza el nombre de unidad organizativa para separar departamentos de la empresa como ou=RRHH para recursos humanos. Todos los objetos se almacenan como contenedores en los que se incluyen sus propiedades y componentes.

Para definir a cualquiera de los registros de información que se crean en LDAP, se utiliza el formato DN (Distinguished Name) o nombre distinguido, similar al utilizado para los certificados X.509. La estructura de almacenamiento de la información asemeja a un árbol, que se va derivando desde el tronco en diferentes ramas y subrayas. El nivel más alto de la información es el DNbase, que actualmente suele almacenar el nombre de la organización en forma de nombre de dominio, aunque en algunos casos se utiliza el nombre legal.

```
o=unam.mx
      dc=unam,dc=mx
            ou=sistemas
                  ou=seguridad
                  ou=sysadmin
                  ou=dba
            ou=rrhh
```

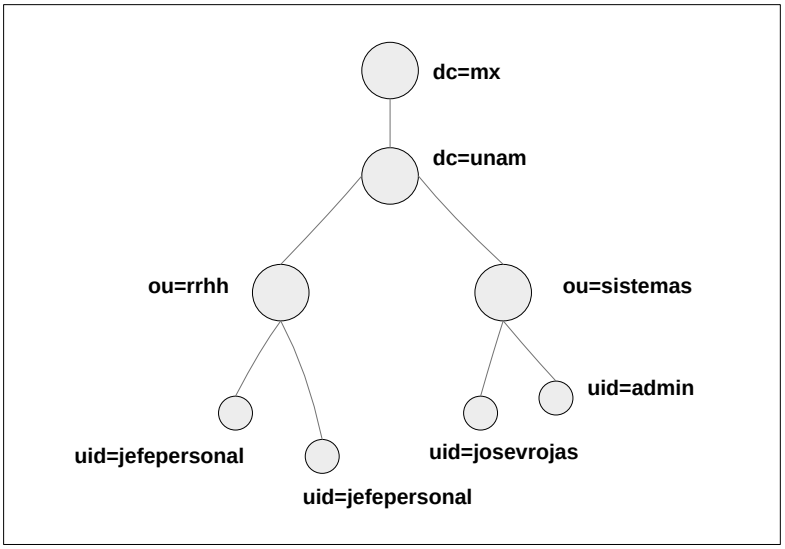

Figura 2-11. Ejemplo para protocolo LDAP.

En el ejemplo anterior y mostrado en la Figura 2-11, la DNbase es el nombre único de la organización (unam.mx), al igual que su nombre de dominio (DC) que es unam.mx. Esta estructura dispone de la unidades organizativas (OU) o departamentos sistemas (que se desglosa en los departamentos seguridad, sysadmin y dba) y recursos humanos. Cada una de esas OU es como un contenedor capaz de almacenar la información en su interior. A la hora de almacenar información en el directorio se utiliza un DN (único en el directorio) compuesto de un nombre relativo (RDN) y su localización en el directorio. La parte relativa se obtiene al extraer del DN el nombre único que define al objeto sin su localización en el árbol. Este nombre relativo se almacena normalmente en forma de nombre común o CN.

```
cn=ServidorRadius,ou=seguridad,ou=sistemas,dc=unam,dc=mx
```

Para almacenar los datos de un empleado de una organización (o de cualquier persona) se puede utilizar el formato anterior o el identificador de usuario UID.

```
cn=Jose Valdes Rojas,ou=seguridad,ou=sistemas,dc=unam,dc=mx
uid=Josevrojas,ou=seguridad,ou=sistemas,dc=unam,dc=mx
```

Además de estos nombres de campo o atributos, existen otros muchos para relacionar los valores que deseamos almacenar con su significado. Existen atributos estándar (ya propuestos por el protocolo) y otros que podemos crear a nuestra voluntad,

para satisfacer las necesidades de almacenamiento de datos. Veamos cómo quedaría una entrada completa de un usuario que hemos creado en el directorio:

```
dn: uid=josevrojas, ou=seguridad, ou=sistemas, dc=unam, dc=mx
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: unamPerson
uid: josevrojas
givenname: Jose
sn: Valdes Rojas
cn: Jose Valdes Rojas
telephonenumber: 555-555-355
roomnumber: 123
o: UNAM
mailRoutingAddress: josevrojas@unam.mx
mailhost: mail.unam.mx
userpassword: {crypt} 87a68979FvT34
uidnumber: 9991
gidnumber: 3443
homedirectory: /home/josevrojas
loginshell: /usr/local/bin/bash
```

Este objeto anterior muestra un ejemplo de registro de usuario en formato LDAP, con todos los datos que nos interesa almacenar sobre él.

La seguridad para el acceso a la información almacenada en LDAP es mantenida por las listas de control de acceso o ACL que determinan los niveles de seguridad en el acceso a los datos, para los diferentes usuarios.

Algunos servidores de autenticación basados en LDAP son:

- OpenLDAP
- Fedora Directory Server
- Windows Server 2008

La necesidad de tener una seguridad informática eficiente en una red de datos radica en los ataques que recibe, esto puede ser de manera interna o externa, dónde los

ataques realizados de manera interna son considerados más peligrosos y más difíciles de prevenir que los ataques externos. Un atacante que se conecte en una red interna se beneficia entre varias cosas del ancho de banda para el acceso a la red de datos. Una ejemplo para prevenir un ataque de este tipo es implementar una función de autenticación en la Capa 2 del modelo OSI usando el protocolo *802.1X*. Un switch habilitado con 802.1X y utilizando un servidor basado dentro de los protocolos de autenticación como Freeradius es todo lo que se necesita para implementar la autenticación en la Capa 2. Considerando que la autenticación de la Capa 2 opera en el nivel local de la red física, esto evita que los intrusos utilicen la red física sin autenticarse.

El protocolo estándar 802.1X maneja la autenticación y el servidor bajo algún protocolo de autenticación como Freeradius, el cual proporciona los servicios AAA (Autenticación, Autorización y Contabilidad). Por lo tanto el servidor de autenticación Freeradius accede a lo que se le conoce como el directorio del servidor, en este caso podría ser dado de alta por un directorio bajo OpenLDAP para obtener información de las cuentas.

Esta solución es un modelo básico para estructurar el servicio de autenticación en una red de datos, esto con el propósito de tener la autenticación entre usuarios y proporcione un alto nivel de seguridad.

Una vez que se tiene un panorama de los servidores de autenticación que existen actualmente comerciales y de código abierto, se realizará la implementación de algunos de ellos para los protocolos Kerberos, Radius y LDAP, lo cual se detalla en el Capítulo 3.