UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

ANÁLISIS COMPARATIVO EN SERVIDORES DE AUTENTIFICACIÓN EN REDES DE DATOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTAN:
MARIO MORALES INCLÁN
MÁXIMO JESÚS SALAZAR QUIROZ
JOSÉ EDUARDO VALDÉS ROJAS

DIRECTOR DE TESIS:
M. C. ALEJANDRO VELÁZQUEZ MENA

CIUDAD UNIVERSITARIA
MÉXICO, DISTRITO FEDERAL, 2010

VNIVERSIDAD NACIONAL AVFNOMA DE MEXICO

FACULTAD DE INGENIERIA
DIRECCION

Designación de sinodales de Examen Profesional

A los señores profesores:

Presidente	M.I. AURELIO ADOLFO MILLAN NAJERA
Vocal	M.C. ALEJANDRO VELAZQUEZ MENA
Secretario	ING. LUIS ARENAS HERNANDEZ
1o. suplente	ING. FILIBERTO MANZO GONZALEZ
2o. suplente	M.C. MARCO ANTONIO VIGUERAS VILLASEÑOR

Me permito informar a ustedes que han sido designados sinodales del Examen Profesional de los señores:

No. CUENTA	NOMBRE	CARRERA
30022215-1	MORALES INCLAN MARIO	INGENIERO EN COMPUTACIÓN
30024659-5	SALAZAR QUIROZ MAXIMO JESUS	INGENIERO EN COMPUTACIÓN
30001983-0	VALDES ROJAS JOSE EDUARDO	INGENIERO EN COMPUTACIÓN

quienes han concluído el desarrollo del tema que les fue autorizado . Ruego a ustedes se sirvan revisar el trabajo adjunto y manifestar a esta Dirección, si es el caso, la aceptación del mismo.

Con el fin de asegurar el pronto cumplimiento de las disposiciones normativas correspondientes y de no afectar innecesariamente los tiempos de titulación, les ruego tomar en consideración que para lo anterior cuentan ustedes con un plazo máximo de **cinco días hábiles** contados a partir del momento en que ustedes **acusen recibo de esta notificación**. Si transcurrido este plazo el interesado no tuviera observaciones de su parte, se entendería que el trabajo ha sido aprobado, por lo que deberán **firmar el oficio de aceptación del trabajo escrito.**

Doy a ustedes las más cumplidas gracias por su atención y les reitero las seguridades de mi consideración más distinguida.

Atentamente,
"POR MI RAZA HABLARA EL ESPIRITU"

Cd. Universitaria, D.F., a 26 de Noviembre de 2010.

EL DIRECTOR
Mtro. José Gonzalo Guerrero Zepeda

JGGZ ' MFB ' ICH

EP-5

A nuestros padres.

A la Universidad Nacional Autónoma de México.

AGRADECIMIENTOS

A mis padres Mario Morales y Laura Inclán por apoyarme incondicionalmente y darme siempre ánimos para alcanzar uno de mis grandes anhelos en esta vida, demostrándome que con humildad, perseverancia y entusiasmo cualquier meta se puede lograr.

A mis hermanas Laura, Gaby, Perla y Vanessa por estar conmigo en todo momento y ser siempre un ejemplo de superación.

Gracias a mi gran Universidad por darme una excelente formación académica y gracias a la docencia de sus profesores y asesores por brindarme todos sus valiosos conocimientos.

Agradezco a mis compañeros de tesis Máximo y José Eduardo, que con el transcurso del tiempo se han convertido en mis grandes amigos, gracias por su apoyo, dedicación y tolerancia para poder alcanzar este mutuo logro.

Agradezco al M.C. Alejandro Velázquez Mena por su valioso apoyo, recomendaciones y enseñanzas para poder ver realizado este proyecto de tesis.

Mario Morales Inclán

AGRADECIMIENTOS

Primero y antes que nada, gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo este periodo de estudio.

A mis padres, María de los Angeles y Senorino, por su infinito amor, cariño y apoyo brindado a lo largo de toda esta etapa de mi vida, ellos siempre han sido mi principal soporte durante todo este proceso de educación y formación, y les estaré eternamente agradecido. Esta tesis es por ellos y para ellos.

Agradezco a mis compañeros de tesis y amigos, Mario y José Eduardo, porque a pesar de las circunstancias y contratiempos, nunca dejamos morir este proyecto y ahora, nuestro esfuerzo realizado en estos años, se ve reflejado con este trabajo de tesis.

A mis grandes amigos Joel y Julio César, por su incondicional apoyo y amistad otorgados a través de todos estos años.

Un agradecimiento especial al M.C. Alejandro Velázquez Mena, nuestro director de tesis, por la colaboración, paciencia y apoyo brindados desde siempre hasta la culminación de este proyecto.

Por último, pero no menos importante, gracias a la Universidad Nacional Autónoma de México por hacerme sentir orgulloso de formar parte de la máxima casa de estudios, y gracias a la Facultad de Ingeniería por todos los conocimientos y experiencias que me fueron transmitidos a través de sus profesores, asesores, tutores y los mismos compañeros de carrera.

Máximo Jesús Salazar Quiroz

AGRADECIMIENTOS

Agradezco a la Universidad Nacional Autónoma de México y a todo el sistema que la conforma, especialmente al Colegio de Ciencias y Humanidades Plantel Oriente así como a la Facultad de Ingeniería, por todo el apoyo y conocimiento otorgado a lo largo de mis años de estudio que me permitieron alcanzar este objetivo.

Agradezco a mis compañeros de tesis, Mario y Máximo, por el tiempo dedicado a este trabajo, así como por su compañía y amistad a lo largo de los años.

Agradezco a mis amigos Yssel, Belinda y Julio, cuya presencia y apoyo durante esta etapa me permitió concluir este trabajo.

A mis padres y mi hermana, por la educación y apoyo brindado durante toda mi vida, para seguir y conseguir nuevos objetivos.

Agradezco a nuestro asesor de tesis, el M.C. Alejandro Velázquez Mena por su tiempo y apoyo.

José Eduardo Valdés Rojas

INTRODUCCIÓN

La transmisión y flujo de información en redes de datos ha crecido aceleradamente durante el último par de décadas, principalmente por la facilidad con que es posible acceder a un equipo de cómputo e implementarlo en la mayoría de los rubros de la vida diaria, afectando principalmente las formas de comercio y la economía en general, con lo cual las decisiones comerciales deben ser tomadas cada vez con mayor rapidez lo que implica que aquellas personas que las toman, tengan acceso inmediato a información exacta y segura.

Sin embargo, no sólo es necesario que la planeación de una red incluya los aspectos de funcionalidad y rapidez, que son visibles para el usuario, sino que adicionalmente debe contemplar todos los aspectos necesarios que permitan a la red mantener la robustez necesaria a fin de mantenerse óptima, segura, funcional y en permanente crecimiento.

El flujo de información es de suma importancia y ha conseguido un aumento considerable; por lo que no sólo la información personal de los usuarios de la red es transmitida, ya sea de sus cuentas bancarias, contraseñas, datos personales o simples conversaciones, también se transmiten datos de investigaciones y mercantiles de una empresa, al igual que todos los documentos laborales de las mismas; es decir, en esta época cualquier información es valiosa.

Considerando que la transmisión de información en una red de datos es el intercambio de datos entre dos dispositivos a través de algún medio de transmisión, la *seguridad de la información* tiene gran importancia para la realización de este trabajo, ya que la transmisión de datos debe cumplir una efectividad en el sistema de comunicación.

En este sentido, las comunicaciones en redes de datos no siempre son seguras. Para evitar este defecto se deben aplicar servicios que garanticen la *seguridad informática* del sistema, tales como la confidencialidad, autentificación, integridad, no repudio, control de acceso y disponibilidad.

En un sistema de transmisión de información, la *confidencialidad* es la capacidad de asegurar que sólo las personas autorizadas tienen acceso a ella, la *autenticación* o *autentificación*[1] es considerada como la forma de verificar la identidad entre los participantes de una comunicación, la *integridad* garantiza que la información no ha sido alterada o destruida en el lapso de transferencia, el *no repudio* ofrece la prevención tanto a emisores como a receptores de comprobar que su mensaje fue transmitido, el *control de acceso* en un sistema ofrece la opción de poder controlar qué usuario está autorizado para usar un recurso del sistema y la *disponibilidad* hace referencia a poder acceder a la información deseada cuando lo requiera y cuantas veces sea necesario por el usuario previamente autorizado.

Cabe mencionar que cada uno de estos servicios puede ser muy sencillo o extremadamente complejo, según el número de usuarios y niveles que garanticen la seguridad.

El presente trabajo de tesis, se enfocara en la *autenticación* de equipos de una misma red, centralizando el análisis en los protocolos que se utilizan para el reconocimiento entre las diferentes entidades, y se basará en el supuesto de que en su mayoría los equipos de las redes se encuentran interconectados por cable. De esta forma, se establecerán las características de los protocolos de autenticación con lo cual cada administrador de red puede decidir cuál será el más conveniente para el tipo de red que se pretende implementar. Este trabajo está conformado de cuatro capítulos que se encuentran debidamente relacionados.

En el capítulo uno denominado CONCEPTOS BÁSICOS se proporciona una serie de definiciones sobre la seguridad informática, redes de datos y criptografía, los cuales son explicados en términos simples con el propósito que lectores no familiarizados con estos conceptos comprendan lo abordado en este capítulo y queden claros conceptos claves para seguir con la explicación y aprendizaje de los protocolos de seguridad abordados en el siguiente capítulo.

[1] En este trabajo se referirán indistintamente los términos *autenticación* y *autentificación* como el mismo concepto.

En el capítulo dos de nombre PRINCIPALES PROTOCOLOS DE AUTENTICACIÓN se da un panorama de los distintos protocolos de autenticación que actualmente existen, donde se mencionan sus antecedentes, características de seguridad informática y en algunos casos se mencionan los servidores de autenticación que existen en el mercado. Se describirán esencialmente por tipo comercial o de código abierto sobre plataformas Windows y Unix, lo que dará un panorama para elegir las implementaciones de autenticación descritas en el capítulo siguiente.

En el capítulo tres llamado IMPLEMENTACIÓN DE SERVIDORES DE AUTENTICACIÓN se describen los pasos de instalación para el sistema operativo, instalación y configuración del servidor de autenticación y las distintas pruebas de conexión para los clientes dados de alta en el servidor, proporcionando muestras de administración en una red de datos y pruebas reales de la autenticación de un cliente con el servidor. Estas implementaciones dan paso al capítulo siguiente, donde se describen sus correspondientes análisis.

En el capítulo cuatro ANÁLISIS DE IMPLEMENTACIONES se da un análisis puntual de las implementaciones realizadas, donde se dará detalle de acuerdo a las siguientes características; administración del servidor implementado, optimización de recursos del servidor de acuerdo a las necesidades del sistema operativo y del servicio de autenticación instalado, así como eficiencia del protocolo de seguridad que emplea cada servidor de autenticación implementado.

CAPÍTULO 1
CONCEPTOS BÁSICOS

1.1 SEGURIDAD Y REDES

Un sistema informático se compone de 5 elementos: hardware, software, datos, memoria y usuarios.

De estos componentes cualquiera puede convertirse en un objetivo para el delincuente informático. Con estas opciones para poder atacar algún sistema, se dificulta el análisis de riesgos y ofrece la ventaja de aplicar al delincuente la filosofía del punto más débil, lo que significa, atacar al sistema por su punto más vulnerable. Por lo tanto, de cara a la protección del sistema, será necesario considerar por igual a los elementos antes citados como vulnerables de un ataque.

1.1.1 Principios de la Seguridad Informática

La filosofía del punto más débil da lugar al primero de los tres Principios de la Seguridad Informática conocido como Principio del Acceso más Fácil. Los tres puntos de los Principios de la Seguridad Informática son:

- Principio del Acceso más fácil.
- Principio de la Caducidad de la Información.
- Principio de la Eficiencia.

- Principio del Acceso más fácil.

"El intruso al sistema utilizará cualquier artilugio o mecanismo que haga más fácil su acceso al sistema y posterior ataque."

Si se afirma que todo sistema informático presenta vulnerabilidades o debilidades en la seguridad de un sistema, aparecen tres preguntas básicas:

a) ¿De qué forma se manifiestan estas debilidades?
b) ¿Cómo se pueden clasificar las amenazas?
c) ¿Qué medidas de control se deben utilizar?

Dando solución a la primera cuestión; las debilidades de todo sistema informático se pueden agrupar en función de los problemas que ocasionan debido a la exposición, vulnerabilidades, ataques y amenazas.

La exposición se refiere a la posible pérdida o daño en el sistema debido a modificación, extravío de datos o acceso no autorizado al sistema. La vulnerabilidad es el punto débil del sistema que, si se traspasa, produce los efectos nocivos indicados. Un ataque es el hecho de la intromisión con daño manifiesto al sistema y, por último, las amenazas consisten en desastres naturales, errores humanos, fallos de hardware y software, sean fortuitos o voluntarios.

En cuanto a la segunda pregunta, dado que los objetivos principales de ataque son el hardware, el software y los datos, se clasifican las amenazas en cuatro tipos: amenazas de interrupción, intercepción, modificación y generación de la información en general.

Los objetivos amenazados (hardware, software, datos) pueden caracterizarse como un flujo (información, servicio, programas, datos, etc.). Si éste es el caso, una representación visual de cada una de estas amenazas se pueden observar en la Figura 1-1.

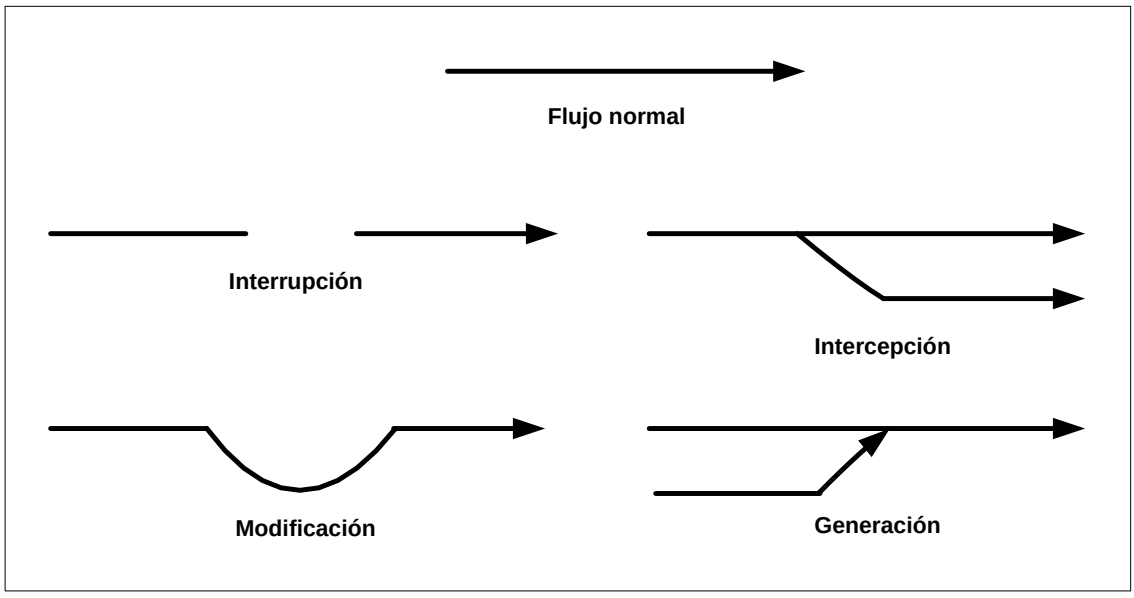

Figura 1-1. Representación gráfica de amenazas.

A continuación se detalla el alcance de cada una de éstas amenazas:

Interrupción:

Se produce cuando un punto del sistema se daña, pierde o deja de funcionar. La detección de este problema es inmediata, tanto por el sistema como por el usuario. Como ejemplos de interrupción son la destrucción maliciosa del hardware, borrado de programas y/o datos, fallos del sistema operativo, etc.

Intercepción:

Es el acceso a la información por parte de personas no autorizadas. Su detección resulta difícil dado que no deja huellas. Como ejemplos de intercepción son las copias ilícitas de programas y la escucha de una línea de datos.

Modificación:

Se produce una amenaza de modificación cuando alguien no autorizado accede al sistema y cambia el entorno para su beneficio. Dependiendo de las circunstancias puede resultar difícil de detectar, siendo ejemplos típicos los de modificación de una base de datos y los de hardware, aunque éste último es más sofisticado y menos frecuentes.

Generación:

Contempla la creación de nuevos objetivos dentro del sistema informático, tales como añadir transacciones específicas de red o registros a una base de datos. Su detección resulta difícil y en muchos casos se trata de un delito de falsificación.

- Principio de la Caducidad de la Información.

"Los datos deben protegerse sólo hasta que pierdan su valor."

En función de la caducidad de la información, se puede pensar en minutos, horas, días o años el tiempo en que se debe mantener la confidencialidad de los datos. Por ejemplo, no tendrán igual tratamiento los datos sobre un censo electoral que los de un desarrollo de un nuevo prototipo de software.

Cabe aclarar que este principio de la caducidad forzará a diseñar algoritmos criptográficos que cumplan con una determinada fortaleza al criptoanálisis (acción de

romper de forma ilegal un mensaje cifrado), en función del tiempo que se desee mantener en secreto la información.

Conociendo las debilidades y clasificando las amenazas, sólo resta decidir qué medidas de control se pueden implementar para proteger al sistema y a la información allí almacenada. Ello conlleva diversas acciones y procedimientos (planes de contingencia, controles de acceso, niveles de seguridad, etc.), así como el uso de dispositivos físicos específicos.

Para defenderse frente a estas amenazas se deben crear métodos de control que preserven el supuesto secreto asociado a la información, el acceso a esos datos solamente a las personas autorizadas y, por último, que tales datos estén disponibles a dicho usuario cuando éste lo desee. Estos tres aspectos darán lugar a los tres elementos básicos de la seguridad informática conocidos como confidencialidad, integridad y disponibilidad de la información.

En cuanto a los sistemas de control, éstos pueden ser mediante hardware, a través del uso de dispositivos que limiten físicamente el acceso a un programa, aplicación o datos; mediante software directo, los relacionados con el desarrollo de los sistemas operativos y programas que contemplan la protección de archivos, directorios, definición de niveles de usuarios, etc., y por último, el software de aplicación para el cifrado de la información.

- Principio de la eficiencia.

"Las medidas de control se implementan para ser utilizadas de forma efectiva. Deben ser eficientes, fáciles de usar y apropiadas al medio."

El decir que sean efectivas significa que cuando sean invocadas por un programa o por el usuario, funcionen perfectamente, lo que se puede asociar al hecho de estar en el lugar y momento oportunos. En cuanto a la eficiencia, se refiere a indicar que debe funcionar sin producir trastornos ni fallos al sistema informático, en términos de consumo de tiempo, ocupación de espacio de memoria o deficiente interfaz hombre/máquina. En resumen, que funcione y lo haga bien, optimizando el uso de recursos.

Todo esto lleva a la afirmación de que un buen sistema de seguridad es aquel que contempla controles eficaces y no obstante, pasa desapercibido por el sistema informático y por sus usuarios. [1]

1.1.2 Servicios de Seguridad

Un servicio de seguridad es aquel que mejora la seguridad de un sistema de información y el flujo de la información de una organización. Los servicios están dirigidos a evitar los ataques de seguridad y utilizan uno o más mecanismos de seguridad para proveer el servicio. [2]

Anteriormente se mencionaron tres elementos en los que se basa la seguridad informática, entendida ésta como seguridad lógica. Estos elementos son la confidencialidad, integridad y disponibilidad.

La confidencialidad de la información significa que los componentes del sistema son accesibles solamente por aquellos usuarios autorizados. La forma de acceder a estos datos puede ser mediante la lectura y observación de los mismos, su impresión, así como el simple conocimiento de su existencia. Si un documento es confidencial, se supone que existe un transmisor y uno o varios receptores autorizados y sólo ellos deberían tener acceso a dicho documento, es decir un secreto compartido.

Con respecto a la integridad, entendemos ésta como el hecho de que los componentes del sistema no sean modificados en la transición de la información. Esta modificación puede ser por medio de la escritura, cambios de datos, modificación de estatus, borrado y creación de nuevos objetos.

La disponibilidad indica que aquellos usuarios autorizados deben tener disponibles los componentes del sistema cuando así lo deseen y tantas veces como sea necesario.

Adicionalmente a estos tres servicios de seguridad se deben considerar los servicios de seguridad de no repudio, control de acceso y autenticación. Donde el

servicio de seguridad de la autenticación será el tema principal para la realización de esta tesis.

Se le conoce al servicio de seguridad de no repudio, como aquel que previene a los emisores o a los receptores de negar un mensaje transmitido. Cuando un mensaje es enviado, el receptor puede probar que el mensaje fue enviado por el presunto emisor. De manera similar, cuando un mensaje es recibido, el remitente puede probar que el mensaje fue recibido por el presunto receptor. Esto es, el no repudio ofrece protección a un usuario frente a otro usuario que niegue, posteriormente, haber realizado cierta comunicación o recepción de un mensaje enviado.

El control de acceso se refiere al mantener controlado el acceso a un medio de información ya sea a través de un dispositivo pasivo tal como una puerta cerrada o a través de un dispositivo activo como lo puede ser un monitor. Un monitor de control de acceso determina qué usuario está autorizado para usar un recurso de manera requerida. Antes de otorgar el acceso, el monitor puede validar la identidad del usuario.

La autenticación es el servicio de seguridad referente al "verificar" la identidad. En la vida diaria generalmente la autenticación se hace de manera informal y, en ocasiones, sin pensarlo. Todos inconscientemente autenticamos gente, compañías y ubicaciones todo el tiempo.

Por ejemplo, cuando se asiste a casa, autentica el hogar comparándolo con la memoria. Si se visita el hogar de un amigo, se verifica que está en la ubicación correcta comprobando la dirección dada por la calle y el número sobre la casa. Cuando se entra a una sucursal de un banco, lo autentica por su logotipo y colores.

La forma más popular de autenticación individual es una firma. Una firma se usa para autenticar al titular de la cuenta en el banco, para comprometer a una persona para alojarse en un hotel y para autenticar al titular de la tarjeta de crédito al realizar alguna compra. La firma se usa no solamente para autenticar la identidad, sino también para dar autorización.

El servicio de autenticación trata de asegurar que una comunicación sea auténtica. En el caso de un sólo mensaje como una señal de alarma o una advertencia, la función del servicio de autenticación asegura al receptor que el mensaje proviene de la fuente que éste espera que provenga.

En el caso de una interacción en curso como la conexión de una terminal a un anfitrión, dos aspectos son envueltos:

- Al momento en el que la conexión se inicia, el servicio verifica que las dos entidades sean auténticas (esto significa que cada entidad es en realidad la que se supone que debe ser).

- El servicio debe asegurar que la conexión no pueda ser interferida por un tercer individuo que pueda enmascararse como una de las dos entidades legítimas con el único propósito de realizar una transmisión o recepción no autorizada.

La autenticación es utilizada para proporcionar una prueba al sistema de que en realidad se es la entidad que se pretende ser. El sistema verifica la información que alguien provee contra la información que el sistema sabe sobre esa persona.

La autenticación es realizada principalmente a través de:

- Algo que se sabe: una contraseña o un número personal de identificación, es algo que se sabe. Cuando se le provee al sistema, éste lo verifica contra la copia que está almacenada en el sistema para determinar si la autenticación es exitosa o no.

- Algo que se tiene: una tarjeta o un pasaporte es un ejemplo de algo que se tiene, lo cual es utilizado por el sistema para verificar la identidad.

- Algo que se es: la voz, la retina, la imagen del rostro o una huella digital pueden identificar de quién se trata y pueden ser utilizadas en el proceso de autenticación. [2]

Por lo tanto, los servicios de seguridad que se mencionaron:

- Confidencialidad
- Integridad
- Disponibilidad
- No repudio

- Control de Acceso
- Autenticación

Si se cumplen completamente estos seis servicios se considera que los datos en una red de datos están protegidos y seguros.

1.1.3 Autenticación, Autorización y Auditoría (Contabilidad)

Una parte importante de la seguridad de la red es la autenticación, autorización y auditoría, conocidas colectivamente como la AAA (Authentication, Authorization and Accounting). AAA es un marco, en el que un administrador puede mantener el control de acceso sobre los dispositivos de red.

AAA cubre control de acceso sobre routers, switches, firewalls, servidores, etc. Cualquier dispositivo de red que no sea una estación de trabajo y que permite el acceso remoto, puede caer bajo las políticas AAA. AAA no es un protocolo en sí mismo, sino que es un conjunto de directrices promovidas por The Internet Engineering Task Force (IETF) que describe cómo deben comportarse los protocolos de acceso a optimizar sus beneficios de la seguridad.

Los protocolos más utilizados asociados a la AAA son Kerberos, Remote Authentication Dial-In User Service (RADIUS) y la terminal de acceso de controlador de sistema de control de acceso+ (TACACS+, Terminal Access Controller Access Control System+).

Al proporcionar un marco para el control de acceso, la AAA ofrece al administrador de red una forma de aplicar una política uniforme en todos los dispositivos de red. Este tipo de estándar de la política tiene dos ventajas: provee a un administrador de red la capacidad de centralizar toda la información contable, y crea un nivel de acceso que pueden aplicarse uniformemente a través de la red.

La capa AAA, cuando es necesario en las redes, permite la mezcla de los diferentes tipos de autenticación, no sólo dentro de la red, sino también en la misma interfaz de red. AAA, como con cualquier buen modelo de seguridad, proporciona a un

administrador de red una gran flexibilidad. Se ajusta en torno a una red existente, en lugar de obligar a la red a entrar en un rígido modelo de seguridad.

En la Figura 1-2 se muestra cómo un modelo de auditoría AAA encajaría en una red. Los servicios AAA en general se encuentran en las máquinas remotas, de modo que si un dispositivo de red está comprometido y, por consiguiente, la validez de sus propios registros es cuestionable, hay un registro independiente de los tiempos de acceso y, posiblemente, los cambios realizados en el dispositivo.

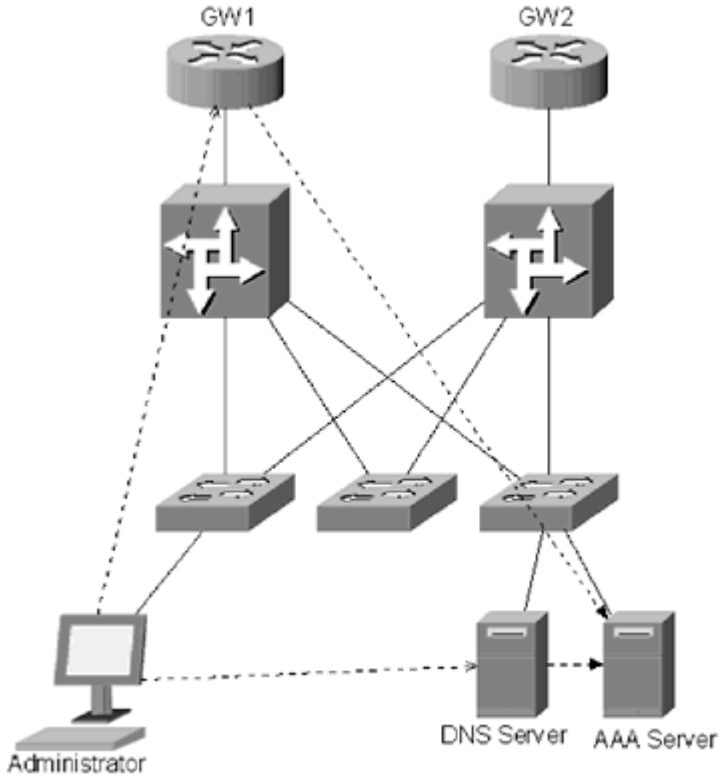

Figura 1-2. Una típica configuración de la red utilizando los servicios de la AAA.

La autenticación es el proceso en el que se identifica un usuario en un dispositivo. Esto incluye el nombre de usuario y la contraseña de proceso y el tipo de cifrado, si los hubiere, que se utiliza durante el proceso de autenticación. El objetivo de la autenticación es restringir el acceso a los dispositivos de red, por lo que tiene que producirse la autenticación antes de que un usuario tenga acceso a un dispositivo. La autenticación se define para cada interfaz. Múltiples formas de autenticación son compatibles con cada interfaz, sin embargo, una autenticación por defecto pueden ser asignada a todas las interfaces.

La autorización es el perfil de usuario. Es lo que determina el nivel de acceso, o lo servicios a los que un usuario tiene acceso. Autorización se puede definir en un par de maneras. Si la política de autorización para cada usuario va a ser coherente en toda la red, entonces la política de autorización se puede definir en el servidor AAA. Si la autorización de la política va a variar de un dispositivo a dispositivo, entonces las políticas de autorización se pueden definir sobre el dispositivo de red individual. Por ejemplo, un administrador de red puede querer definir distintas políticas para los routers y servidores, o un desarrollador web puede tener pleno acceso al servidor web, pero sólo un acceso limitado al servidor DNS. Las políticas de autorización no tienen que estar limitada para cada usuario. Se puede definir para cada grupo, con diferentes grupos con diferentes privilegios.

Controlar los registros, y qué privilegios se han registrado y cuando, no es suficiente. También tienen que ser capaces de controlar lo que hacen mientras está conectado, que es donde la auditoría es importante. La auditoría permite a un administrador de red controlar los tiempos de conexión de una cuenta, las órdenes emitidas mientras está conectado, los recursos utilizados, y los datos transferidos. Las características de la auditoría pueden añadir sobrecarga a la red, sin embargo, la información adicional puede ser de un valor incalculable cuando se trata de localizar, ya sea interno o externo un atacante como la contabilidad del servidor AAA tiene un registro completo de los movimientos realizados por un atacante.

1.1.4 Redes de datos

Una red de datos es un sistema de comunicación que permite a un número de sistemas y dispositivos comunicarse unos con otros [3]. La cual permite enviar y recibir mensajes entre cada uno de los usuarios, los mensajes pueden ser un mail, un documento, una imagen o cualquier forma de comunicación entre los mismos.

Es posible clasificar las redes por su escala [3]:

Local Area Network (LAN, Red de Área Local), son redes óptimas para un área geográfica moderada, como un campus de pocos kilómetros o algún edificio. Son utilizadas para conectar computadoras personales y estaciones de trabajo en oficinas,

sus restricciones se encuentran tanto en el número de usuarios que soportan como en el tiempo de transmisión que es conocido y limitado. Utiliza generalmente conexión mediante cable Ethernet o fibra óptica.

Metropolitan Area Network (MAN, Red de Área Metropolitana), son redes de tamaño medio que abarca una ciudad. Usualmente conectadas mediante cable coaxial o microondas.

Wide Area Network (WAN, Red de Área Amplia), son redes que se expanden en una gran área geográfica, generalmente un país o un continente. Estas contienen un conjunto de máquinas llamadas host, diseñadas para aplicaciones de usuarios. Los host se encuentran conectados en subredes, que se encargan de llevar los mensajes de un host a otro. En la mayoría de las redes de área amplia la subred está compuesta de líneas de transmisión y elementos de conmutación.

1.1.5 Diseño de redes: protocolos y capas

Un protocolo de red es un conjunto de reglas sobre el intercambio de comunicación en la red. Dos sistemas o usuarios que intercambian información deben tener un protocolo en común para tal efecto. Un protocolo determina el formato y la secuencia en la que los mensajes pasan de emisor a receptor, sin importar el medio o la forma con la que se haga la comunicación [3].

Para reducir la complejidad del diseño de redes, la mayor está organizada como una pila de capas o niveles independientes. El propósito de una capa de protocolos es proveer servicios a la capa superior. Un conjunto de capas y protocolos se conoce como arquitectura de red.

El modelo de referencia Open Systems Inteconnection (OSI, Interconexión de Sistemas Abiertos) es una propuesta desarrollada por la International Organization for Standardization (ISO, Organización Internacional de Estándares) para la estandarización de la comunicación entre sistemas, su estructura se utiliza para mostrar cada una de las

funciones de cada capa y tener un estándar a seguir en el desarrollo de aplicaciones de comunicación.

La seguridad, no se refiere o aplica a una sola capa del modelo, debido a que es posible realizar una implementación de seguridad en cada una de ellas, al revisar cada capa (Figura 1-3) es posible indagar que cada una tiene cientos de vulnerabilidades, por ejemplo, es visible que si una de las capas es vulnerada, las comunicaciones están en peligro sin que las otras capas sean conscientes del problema, por lo que es necesario llevar a cabo todas las posibles soluciones que se han desarrollado tanto de protocolos y aplicaciones, como de hardware y software para tratar de mantener segura la información viaja a través de la red.

Figura 1-3. El modelo de referencia OSI.

A continuación se dará una breve explicación de las capas en que se centrará el análisis [3]:

La capa física consiste en estándares que describen el orden de los bits, las tasas de transmisión de bits, tipos de conectores y otras especificaciones. La información en esta capa es transmitida en formato binario.

La capa de enlace de datos de manera general esta capa se encarga de transformar los bits puros recibidos del canal de comunicación y llevarlos a la capa de red sin errores. Generalmente esto se resuelve mediante la fragmentación de la información en tramas enviadas secuencialmente y recibiendo una confirmación de recepción. De la misma manera es posible controlar en esta capa la no saturación de los datos enviados.

La Institute of Electrical and Electronics Engineers (IEEE, Instituto de Ingenieros Eléctricos y Electrónicos) divide la capa de enlace de datos en dos subcapas: subcapa Logical Link Control (LLC, Control de Enlace Lógico) y la subcapa Medium Acces Control (MAC, Control de Acceso al Medio).

1.1.6 Seguridad en capas

Un sólo mecanismo no puede ser utilizado para proteger una red. Para proteger la infraestructura debe aplicarse la seguridad en capas, también conocida como defensa profunda [5]. La idea es crear varios sistemas, de tal forma que si existe un fallo en alguno de ellos no se convierte en una vulnerabilidad, pero es interceptado en la siguiente capa. Adicionalmente la vulnerabilidad puede ser limitada y controlada en la capa afectada debido a la seguridad aplicada a diferentes niveles. La seguridad en capas es el método preferido y más escalable para proteger una red.

1.1.7 Seguridad en redes

En los primeros días de las redes, el administrador de la red por lo general tenía un estricto control sobre la conexión remota de los sistemas. En la actualidad, la proliferación de las redes interconectadas y el fácil acceso remoto e intercambio de recursos, resulta casi imposible identificar y confiar en todos los puntos de acceso de un sistema.

La seguridad en redes es definida (por la United States National Security Agency, NSA, Agencia de Seguridad Nacional de los Estados Unidos) como la protección de las redes y sus servicios de la modificación, destrucción o divulgación no autorizada, asegurándose que la red trabaje correctamente sus funciones críticas y sin efectos secundarios perjudiciales [4].

Hay una serie de estrategias diferentes para lograr la seguridad en un entorno de red. La elección de cuáles y cuántas estrategias se utilizaran depende en gran medida del

tipo y alcance de la red, el nivel de confianza de los usuarios y el valor de la información transmitida.

La seguridad en redes por lo tanto es un sistema, no es un firewall, un detector de intrusos, una red privada virtual, no es la autorización, la autenticación y la auditoría. La seguridad son todas las soluciones que existen en el mercado para la protección de los servicios de red [4].

Con ambas referencias, es posible entender que un sistema de seguridad de redes es una colección de dispositivos y tecnologías conectadas a la red, aunadas a buenas prácticas que trabajan complementariamente para proporcionar seguridad a los activos informáticos.

1.1.7.1 Estándares de Seguridad en Redes

La IEEE ha desarrollado un conjunto de normas, conocidas como el estándar 802, principalmente para redes de área local [4]. Los primeros estándares 802, que van del 802.1 al 802.10, básicamente abordan las dos capas más bajas del modelo OSI. El 802.10 es un estándar para la interoperabilidad de la seguridad LAN, conocido como Standard for Interoperable LAN Security (SILS), que está orientado al intercambio de datos en redes.

El estándar X.400 desarrollado por la ISO y Consultative Committee for International Telegraphy and Telephony (CCITT, Comité Consultivo de Telegrafía y Telefonía Internacional), conocido actualmente como International Telecommunication Union (ITU-T, Unión Internacional de Telecomunicaciones), es orientando al modelo OSI para mensajes, que incluyen normas para la seguridad de la mensajería [4].

El estándar X.500 también elaborado por la ISO y CCITT, son las normas elaboradas para la asignación de nombres [4]. Permiten a los usuarios y programadores identificar un objeto (archivo, disco, etc.) sin saber la ubicación del objeto en una red o la ruta de acceso necesaria para alcanzarla. Incluye normas para garantizar la autenticación y nomenclatura segura. La mayoría de los servicios actuales de autenticación dependen de un sistema de parámetros definidos en el X.500.

El estándar X.500 es un sistema global, en la mayoría de los casos el directorio X.500 es generalmente accesible utilizando una herramienta llamada Lightweight Directory Access Protocol (LDAP, Protocolo Ligero de Acceso a Directorios).

Ejemplo de una red de datos [6].

A continuación se mostrara un ejemplo de una típica red corporativa para una empresa de 100 personas. Esta red es bastante insegura. Por supuesto, no hay un modelo de seguridad correcta. Las necesidades en materia de seguridad varían de una compañía a otra, pero es más fácil para los administradores de red, visualizar como corregir y detectar deficiencias para crear mejores métodos.

La infraestructura de red.

Para el ejemplo de la Figura 1-4 es simple: un router conectado a un firewall que tiene tres interfaces, una pública al router y dos privadas, una a la red de los empleados y otra a los servidores.

El conjunto de reglas utilizado por este firewall es muy simple. No es permitido el tráfico hacia la red de los empleados. Todo el tráfico es permitido hacia la red de los servidores. Las reglas a la red de los servidores son ligeras debido a que es necesario tener todos los puertos abiertos.

La compañía utiliza una infraestructura de red Transmmision Control Protocol/Internet Protocol (TCP/IP, Protocolo de Control de Transmisión/Protocolo de Internet), pero no se han realizado auditorías para saber que otros protocolos están funcionando en las máquinas. Utilizan el bloque de red 10.10.10.0 255.255.255.0 (clase C). Las direcciones IP han sido asignadas sin subredes.

Finalmente, todas las máquinas conectadas a un switch utilizan la Virtual Local Area Network (VLAN, Red de Área Local Virtual) por default, ya que los administradores no han asignado diferentes VLAN a los puertos.

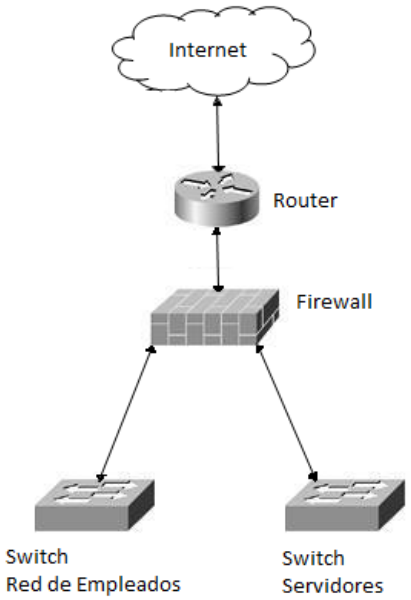

Figura 1-4. Infraestructura de red.

Los Servidores.

Consta de cinco servidores, todos excepto dos, realizan funciones únicas (Fig. 1-5). El servidor de archivos también hace las tareas de un servidor acceso remoto que permite a los empleado entrar a la red desde casa, mientras el controlador de dominio hace a la vez de servidor de monitoreo.

El servidor de archivos, Exchange y el controlador de dominio funcionan bajo Windows NT, con Service Pack 4 instalado. El servidor de dominio y web se encuentran en Red Hat Linux 6.2.

Las cuentas son creadas, según las necesidades, y no han sido auditadas.

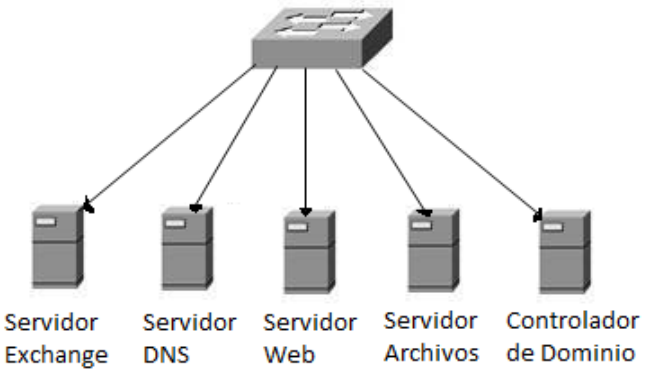

Figura 1-5. Servidores.

La red de los empleados.

Varios grupos de empleados, como recursos humanos o contabilidad, están conectados vía hub a switch de la red (Fig. 1-6). Estos empleados utilizan una combinación de versiones de Windows. De nueva cuenta no hay auditorías o políticas que limiten el tipo de equipos que se unen a la red.

A todos los equipos en red se les asigna una dirección IP por el controlador de dominio cuando inician sesión en la red.

La empresa utiliza red inalámbrica sólo en un par de salones de conferencia, permitiendo a todo equipo con tecnología inalámbrica conectarse a la red.

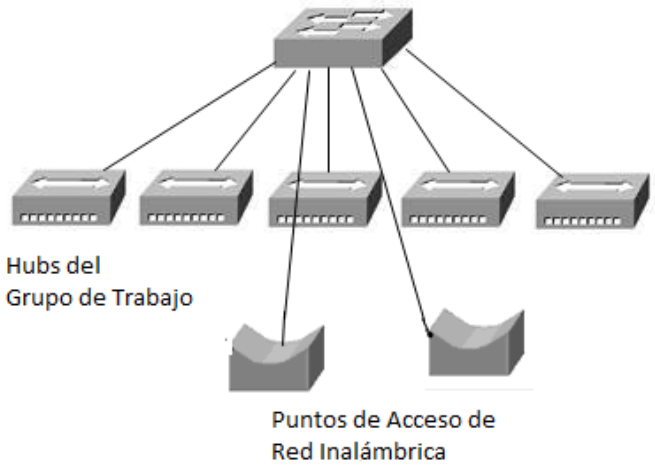

Figura 1-6. Red de empleados.

1.2 CRIPTOGRAFÍA

Criptografía (Etimología de ocultar + escritura) es el arte, técnica o ciencia que permite proteger a la información por medio de la aplicación de un cifrado. Esta sólo puedes ser descifrada por el remitente autorizado. La criptografía da lugar al criptólogo y al criptoanalista, el criptólogo es la persona que trabaja en nombre de un transmisor o receptor no autorizado y cuya función básica es la de crear algoritmos de cifrado y descifrado, mientras el criptoanalista es la persona que trabaja en nombre de un transmisor o receptor no autorizado y cuya función básica es la de romper códigos y textos cifrados para recuperar de forma ilegítima la información allí contenida, utiliza el criptoanálisis como herramienta para descriptar.

Se le conoce a la Criptología como la ciencia que estudia e investiga todo lo concerniente a la criptografía. Esto da lugar al cifrado y descifrado, donde el cifrado es la técnica por la cual, a través de un algoritmo, se modifica o altera la representación de un texto en claro convirtiéndolo en un criptograma de forma que su intercepción por extraños no entregue información alguna del mensaje original y el descifrado es la técnica por la cual a través de un algoritmo, generalmente el inverso del cifrado, un receptor legítimo puede recuperar la información contenida en el criptograma.

Alternativamente existe la Codificación, que es la técnica de cifrar por medio de códigos, no por algoritmos de cifrado. Y la Decodificación, que es la recuperación de la información codificada aplicando una relación directa entre código palabra.

En el proceso de la criptografía se obtiene el Criptograma, el cual es el documento obtenido al cifrar un texto en claro. La representación o alfabeto del criptograma puede ser igual o distinta a la del texto en claro. De modo contrario el texto en claro es el documento original o mensaje que se desea enviar a uno o más destinatarios o bien, almacenar en forma criptografiada.

Dentro de la Criptografía además se emplea el concepto de Clave privada, la cual es la clave secreta utilizada para cifrar un mensaje y cuyo secreto mantiene la inmunidad del sistema. Esto da lugar a criptosistemas de clave secreta. Por otro lado la Clave pública es la clave utilizada en criptosistemas, conjuntamente con una clave privada o secreta, de forma que se cifra con una de ellas y se descifra con la otra. La inmunidad de estos sistemas se basa en el hecho de que, incluso conociendo el algoritmo o transformación y la clave para cifrar, resulta extremadamente difícil romper o descriptar un criptograma sin conocer la segunda clave.

Estos tipos claves dan lugar a la Sustitución, que es la técnica criptográfica que consiste en sustituir un carácter del texto en clave por otro en el texto cifrado. Existen dos tipos de sustitución; la Sustitución monoalfabética y la Sustitución polialfabética. La Sustitución monoalfabética es el cifrado que sustituye cada carácter del texto en clave por otro carácter único en el criptograma, usando un único alfabeto. La Sustitución polialfabética es el cifrado que, mediante una clave, sustituye los caracteres

del texto en claro por otro carácter en el texto cifrado, utilizando para ello más de un alfabeto. Adicional a estos cifrados existe la Transposición, el Cifrador de Bloques y Cifrador de Flujo. La Transposición es la técnica de cifrado que consiste en reordenar un texto en claro (también se conoce como permutación), el Cifrador de Bloques es un sistema que decide previamente el mensaje en bloques de igual tamaño y que luego cifra con la misma clave y Cifrador de Flujo es un Sistema que cifra el mensaje carácter a carácter (o bit a bit) mediante el elemento i-ésimo del flujo de una clave.

1.2.1 Criptografía Simétrica

La criptografía simétrica es el método criptográfico que usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez que ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. En otras palabras, no debería ser de ninguna ayuda para un atacante conocer el algoritmo que se está usando. Sólo si el atacante obtuviera la clave, le serviría conocer el algoritmo. Los algoritmos de cifrado ampliamente utilizados tienen estas propiedades, por ejemplo el algoritmo GPG en sistemas *GNU*.

Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinar el tipo de clave. Esto quiere decir que el abanico de claves posibles, o sea, el espacio de posibilidades de claves, debe ser amplio. Richard Feynman fue famoso en Los Álamos por su habilidad para abrir cajas de seguridad; para alimentar la leyenda que había en torno a él, llevaba encima un juego de herramientas que incluían un estetoscopio. En realidad, utilizaba una gran variedad de trucos para reducir a un pequeño número la cantidad de combinaciones que debía probar, y a partir de ahí simplemente probaba hasta que adivinaba la combinación correcta. En otras palabras, reducía el tamaño de posibilidades de claves.

Actualmente, los ordenadores pueden descifrar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los criptosistemas modernos. El algoritmo de cifrado DES usa una clave de 56 bits, lo que significa que hay 2 elevado a 56 claves posibles (72, 057, 594, 037, 927, 936 claves). Esto representa un número muy alto de claves, pero un ordenador genérico puede comprobar el conjunto posible de claves en cuestión de días. Una máquina especializada puede hacerlo en horas. Algoritmos de cifrado de diseño más reciente como 3DES, Blowfish e IDEA usan claves de 128 bits, lo que significa que existen 2 elevado a 128 claves posibles. Esto equivale a muchísimas más claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, tardarían más tiempo en encontrar la clave que la edad del universo.

1.2.1.1 Criptosistemas de Clave Secreta

También conocidos como de clase única o criptosistemas simétricos, basan su fortaleza en el secreto de la clave k (Figura 1-7). Si se llega a descubrir esta clave secreta, resultaría fácil por lo menos en la teoría obtener las funciones de cifrado y descifrado. La clave k es secreta y compartida por los dos usuarios, el transmisor y el receptor. Se verá un poco más a detalle cómo se aseguran en estos sistemas la confidencialidad y la integridad.

Puesto que solamente el usuario receptor autorizado conocerá la clave con la que ha cifrado el mensaje el usuario transmisor, se asegura de esta forma la confidencialidad: otro usuario no autorizado y que por tanto desconoce la clave, no podrá interpretar el criptograma. Por su parte, dado que sólo el usuario transmisor auténtico está en conocimiento de la clave secreta, el receptor puede estar seguro de que no se trata de un impostor y, por lo tanto, se confirma la integridad de la información. En resumen, al ser la clave única, secreta y compartida por ambos usuarios, podemos afirmar que en los criptosistemas de clave secreta la confidencialidad y la integridad de la información se obtienen al mismo tiempo.

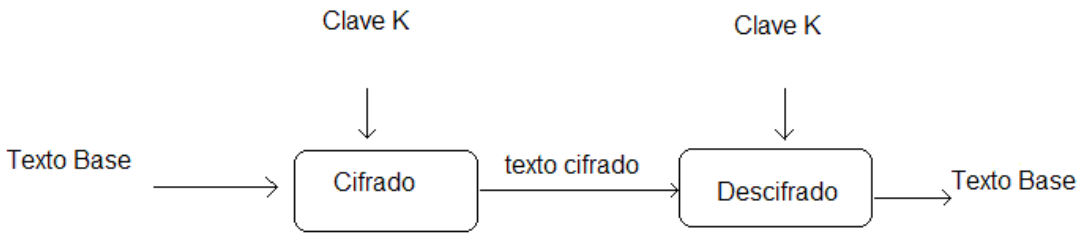

Figura 1-7. Criptosistema de clave secreta.

Aplicando las relaciones de transformación E_k y D_k, podemos representar el criptosistema como se indica en la Figura 1-8.

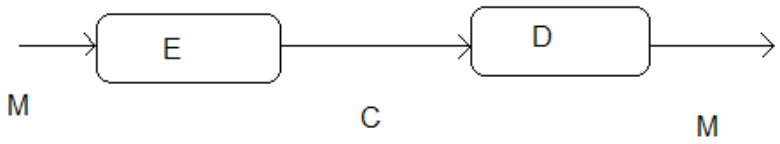

Figura 1-8.Transformaciones en un criptosistema de clave secreta.

Del esquema presentado en la Figura 1-8, se puede deducir las funciones de transformación que tienen lugar un criptosistema de clave secreta:

$$C=E(M)$$

$$M=D(C)$$

$$M = D(E(M))$$

Si la clave es K: $C = E(K, M))$

Luego $M = C(K, E(K, M))$

Representando a la clave de cifrado como K_E y a la clave de descifrado por K_D, ambas son inversas entre sí módulo n. La ecuación anterior sigue siendo válida y se transforma en:

$$M = D(K_D, E(K_E, M))$$

El primer conjunto de ecuaciones se corresponde con los criptosistemas clásicos sin clave como, por ejemplo el cifrado del César, en el que el criptograma se obtiene

simplemente aplicando un desplazamiento de k lugares a la derecha (k = 3 en este caso) en módulo n a cada uno de los caracteres del texto en claro.

Ejemplo 1-1: Usando la función indicada del cifrado del César, se pide:

a) Encontrar el alfabeto de cifrado y luego cifrar el siguiente Mensaje M = PELIGRO.

b) Repetir el punto anterior usando ahora como clave la cadena MURCIELAGO

Solución: Los alfabetos de cifrado y sus correspondientes criptogramas serán los que se indican a continuación:

a) A B C D E F G H I J K M L N Ñ O P Q R S T U V W X Y Z

Alfabeto cifrado: D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

M = PELIGRO => C = SHÑLJUR

b) A B C D E F G H I J K M L N Ñ O P Q R S T U V W X Y Z

Alfabeto cifrado: M U R C I E L A G O D F H J K N Ñ P Q S T V W X Y Z B

M = PELIGRO => C = ÑIFGLQN

Del ejemplo anterior se puede apreciar la diferencia entre utilizar solamente un algoritmo (en ese caso el del César) para cifrar y usar, además, una verdadera clave. Resulta evidente que en el caso de utilizar una clave existe una difusión mayor entre los elementos del alfabeto y, por consiguiente, dificulta en cierta forma el ataque al criptograma. Veremos a continuación cómo puede conseguirse la confidencialidad y la integridad en estos sistemas mediante un análisis genérico.

Para alcanzar la confidencialidad deberá ser imposible entonces para un intruso determinar D_k a partir del criptograma C, incluso en el caso extremo que conozca por algún medio el mensaje claro M. De esto se concluye que para mantener el secreto, esto es, que no se pueda determinar de forma ilegal M partir de C, la condición necesaria es que D_k se mantenga en secreto.

En otras palabras, para alcanzar el objetivo de la confidencialidad deberá cumplirse que:

a) El criptoanalista no podrá determinar sistemáticamente la operación de descifrado: esto es, no podrá descifrar C u otro texto cifrado bajo la transformación E_k.

b) El criptoanalista no podrá determinar sistemáticamente el texto en claro sin contar con la transformación de descifrado.

Estos dos requisitos, que se entremezclan, deberán mantenerse independientemente de la longitud y del número de mensajes interceptados. Esto es, no por tener un criptograma más largo o, por el contrario, contar con una gran cantidad de criptogramas, será más fácil el trabajo de un criptoanalista. Ahora bien, siempre hay que contar con el factor suerte de forma que, aunque sea muy difícil determinar D_k y requiera de una gran cantidad de cálculos, por un acierto genial el criptoanalista logre determinarla sin más. Se debe observar, no obstante, que si sólo se desea mantener el secreto de la información, bastará con mantener en secreto la función de descifrado, con lo cual podríamos hacer pública la función de cifrado E_k, salvo que su conocimiento por parte de un extraño permitiera inferir la función D_k.

Si se profundiza ahora sobre el objetivo de la integridad o autenticidad de la información, ésta se logra si resulta imposible para un impostor enviar un mensaje haciéndose pasar por el transmisor legítimo. Esto es lo mismo que decir que sea imposible determinar la función E_k a partir del criptograma C, incluso si se conoce el mensaje en claro M. Por lo tanto, para que sea imposible encontrar de forma sistemática un criptograma C' tal que D_k (C') sea un texto base válido en el conjunto de los mensajes M, la función de cifrado E_k deberá ser secreta (Figura 1-9).

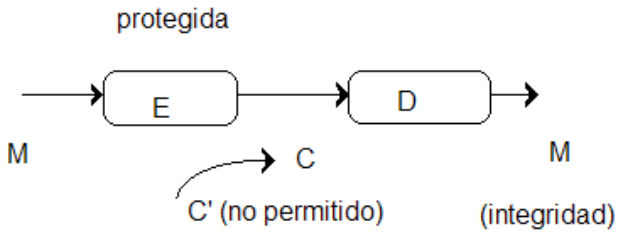

Figura 1-9. Integridad en un sistema de clave secreta.

En otras palabras, para alcanzar el objetivo de integridad deberá cumplirse que:

a) Debe ser computacionalmente imposible que un criptoanalista sistemáticamente determine la transformación de cifrado E_k a partir de C, aunque se conozca el texto en claro del mensaje M. Esto es, no podrá cifrar un texto en claro diferente M' y enviarlo como C' = E_k (M') al destinatario en vez de C.

b) Debe ser computacionalmente imposible que un criptoanalista encuentre de forma sistemática un texto cifrado C' tal que, al aplicarle la transformación D_k, obtenga un texto en claro válido en el espacio de mensajes M sin la transformación de cifrado.

De lo anterior se deduce que la función de cifrado E_k deberá estar protegida. No obstante, de forma similar al caso anterior de la confidencialidad, ahora podríamos hacer pública la función de descifrado si lo que nos interesa es únicamente preservar la autenticidad del mensaje. ¿Cómo se logra entonces que se cumplan ambos requisitos de seguridad, es decir, la confidencialidad y la integridad, en un sistema de clave secreta? La respuesta es obvia: protegiendo o haciendo secretas ambas funciones, la de cifrado y la de descifrado. De ahí que dado el secreto de la clave, en estos sistemas la confidencialidad y la integridad se obtienen de forma conjunta. No llegaremos a la misma conclusión en los sistemas de clave pública que se analizaran un poco más adelante en este mismo capítulo.

En resumen, la protección total en los sistemas de clave secreta se puede dar tal y como se muestra en la Figura 1-10.

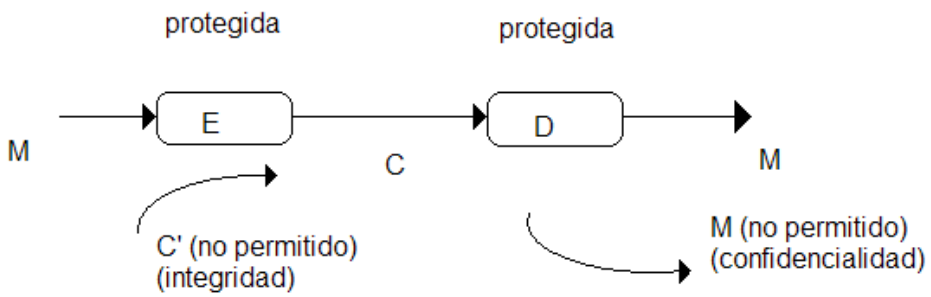

Figura 1-10. Integridad y confidencialidad en un sistema de clave secreta.

1.2.2 Criptografía Asimétrica

La criptografía asimétrica es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona a la que se ha enviado el mensaje. Una clave es pública y se puede entregar a cualquier persona, la otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Los sistemas de cifrado de clave pública o sistemas de cifrado asimétricos se inventaron con el fin de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario. Es más, esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí.

Como con los sistemas de cifrado simétricos buenos, con un buen sistema de cifrado de clave pública toda la seguridad descansa en la clave y no en el algoritmo. Por lo tanto, el tamaño de la clave es una medida de la seguridad del sistema, pero no se puede comparar el tamaño de la clave del cifrado simétrico con el del cifrado de clave pública para medir la seguridad. En un ataque de fuerza bruta sobre un cifrado simétrico con una clave del tamaño de 80 bits, el atacante debe probar hasta 280-1 claves para encontrar la clave correcta. En un ataque de fuerza bruta sobre un cifrado de clave pública con una clave del tamaño de 512 bits, el atacante debe factorizar un número compuesto codificado en 512 bits (hasta 155 dígitos decimales). La cantidad de trabajo para el atacante será diferente dependiendo del cifrado que esté atacando. Mientras 128 bits son suficientes para cifrados simétricos, dada la tecnología de factorización de hoy en día, se recomienda el uso de claves públicas de 1024 bits para la mayoría de los casos.

1.2.2.1 Criptosistemas de Clave Pública

Como se observó en el subtema anterior, al parecer los criptosistemas de clave secreta funcionan bien, pues sólo con mantener en secreto una clave, se asegura el secreto de la información y su autenticidad. Es más, muchos de ellos presentan una apreciable seguridad, siendo uno de los más conocidos y ampliamente utilizados en el mundo empresarial y de negocios el DES (Data Encryption Standard). Si esto es así, ¿Por qué plantearse entonces otro tipo de sistema conocido como de clave pública?

La respuesta a esta pregunta es muy sencilla aunque mucho más extensa que la que presentamos a continuación, no por ello menos válida.

Suponer que dos usuarios desean intercambiarse mensajes secretos y para ello deciden utilizar un sistema de cifrado de clave secreta; es esto es, eligen ambos una clave que sólo ellos conocen y mantienen en secreto. En este entorno, es evidente que ambas claves son iguales.

Suponer ahora que el grupo decide integrar a otro usuario, y deciden mantener las conversaciones secretas independientes de forma que, además de las dos claves anteriores, se crean dos claves más, para mantener comunicaciones independientes del tercer usuario con el primero y el segundo. Esto implica que el número de claves ha crecido de una a tres, mientras que el número de usuarios sólo ha aumentado en uno. Es fácil comprobar que si añadimos un cuarto participante en este grupo, el espacio de claves crecerá ahora hasta 6.

Por tanto no existe una relación lineal entre el número de usuarios del sistema y el número de claves secretas necesarias para conservar el secreto y la autenticidad de los mensajes que se intercambian.

Deduciendo que al aumentar el número de usuarios n, el número de claves secretas tiende a n^2. Por ese motivo, resulta impracticable para sistemas con muchos usuarios, básicamente por dos aspectos puntuales:

- Si no existe un control exhaustivo de las claves, pueden aparecer claves repetidas, lo que obviamente desvirtúa el concepto de secreto y autenticidad de todo sistema de cifrado.

- Por otra parte, al crecer de forma cuadrática el número de claves secretas, dificulta el trabajo del usuario que desea comunicarse con todos los demás, puesto que deberá conocer un gran número de claves diferentes, que al ser secretas tendría que tenerlas en mente y no en un archivo y menos en un listado.

Todo esto se complica aún más si se considera la tendencia actual de intercomunicación a través de redes mundiales. De ahí nace la necesidad de crear un criptosistema en el que el número de claves crezca en forma lineal con respecto al número de usuarios, sin por ello verse afectada la seguridad de la información que por él se transmite. Asimismo, al verse reducido el número de claves secretas, la gestión de las mismas se simplifica y el sistema es más económico.

La filosofía de un criptosistema de clave pública reside en que cada usuario dispone de dos claves, una de ellas de carácter privado (secreta) nombrado de forma genérica u_iv, y otra de carácter público que es u_ib. El subíndice i indica que dicha clave, privada o pública, pertenece al usuario i-ésimo. Las claves u_iv y u_ib serán inversos en el cuerpo o módulo en que trabaje el cifrador, en el sentido matemático.

La idea es que cada usuario cifre sus mensajes con una de las dos claves y los descifre con la otra, en función de que le interese bien conservar la confidencialidad, bien la integridad de su información, o bien ambas a la vez. El secreto del sistema está en que por mucho que los demás usuarios conozcan nuestra clave pública, les será computacionalmente imposible determinar nuestra clave privada. Ahora bien, no debemos perder de vista que el algoritmo de cifrado y descifrado es público; más aún, será muy sencillo.

A diferencia de los criptogramas de clave secreta, en los criptogramas de clave pública la confidencialidad y la integridad de la información se obtienen de forma separada. Por otra parte resulta obvio, que hemos solucionado el problema cuadrático del número de claves puesto que si un nuevo usuario j entra en el sistema el número de

claves simplemente crece en una unidad, que corresponde a su clave pública u_ib, ya que la clave privada u_iv es personal y no forma parte del sistema de claves.

A continuación y al igual que en los sistemas de clave secreta, realizaremos el análisis de las transformaciones de cifrado y descifrado que aseguran el secreto y la autenticidad en los sistemas de clave pública. Para una mayor sencillez en la explicación, supondremos una comunicación en que el usuario A envía un mensaje secreto a su amigo B por lo que las claves serán las siguientes:

- E_A es la operación con clave de cifrado de A y es pública.
- D_A es la operación con clave de descifrado de A y es privada.
- E_B es la operación con la clave de cifrado de B y es pública.
- D_B es la operación con la clave de descifrado de B y es privada.

Como se ha comentado, para lograr la confidencialidad, A envía el mensaje a B cifrándolo con la clave pública de B, es decir, E_B. Por su parte, B recibe el criptograma C y los descifra utilizando su clave privada D_B (Figura 1-11).

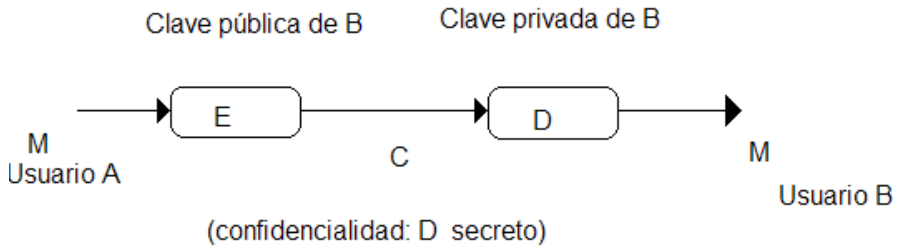

Figura 1-11. Confidencialidad en un sistema de clave pública.

$$C = E_B(M) \qquad \text{(operación de cifrado)}$$
$$M = D_B(C) = D_B(E_B(M)) \qquad \text{(operación de descifrado)}$$

Hay que observar que esta operación de cifrado sólo asegura el secreto; esto es que solamente el usuario a quien se dirige el mensaje podrá descifrarlo, pero no así la autenticidad. Cualquier otro usuario impostor A' podría hacerse pasar por el usuario A y enviar un mensaje M' al usuario B, en tanto que la clave pública de B la conoce todo el mundo.

Si el usuario A desea mantener la integridad de sus mensajes, esto es que nadie pueda hacerse pasar por él, entonces los cifrará con su clave privada D_A. El usuario B (o cualquier otro que pudiera leer el mensaje) lo podrá descifrar con la clave pública de A, es decir E_A (Figura 1-12).

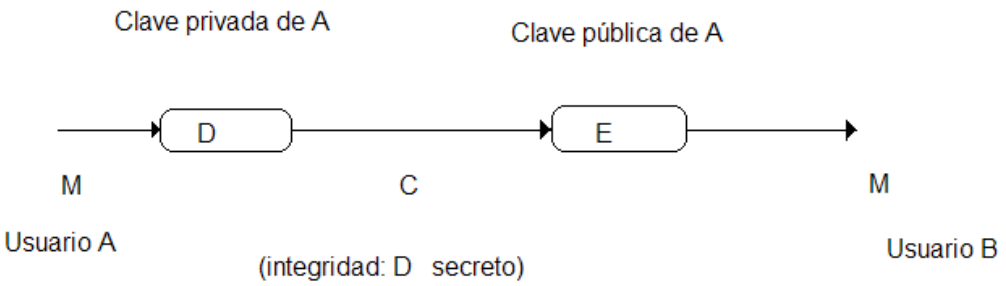

Figura 1-12. Integridad en un sistema de clave pública.

Las transformaciones aplicadas serán en este caso:

$C = D_A (M)$ (operación de cifrado)

$M = E_A(C) = E_A (D_A (M))$ (operación de cifrado)

Resulta claro que con esto se asegura la autenticidad de quien envía el mensaje, pero no así que dicho mensaje sea secreto. Como todo el mundo conoce la clave pública de A, dicho mensaje es también público. En este caso, y a diferencia de los criptogramas de clave secreta, estas dos características se obtienen por separado.

¿Cómo se puede entonces obtener para un mensaje cifrado con un sistema de clave pública la confidencialidad y la integridad? La respuesta: aplicando las dos operaciones anteriores tal y como se muestra en la Figura 1-13.

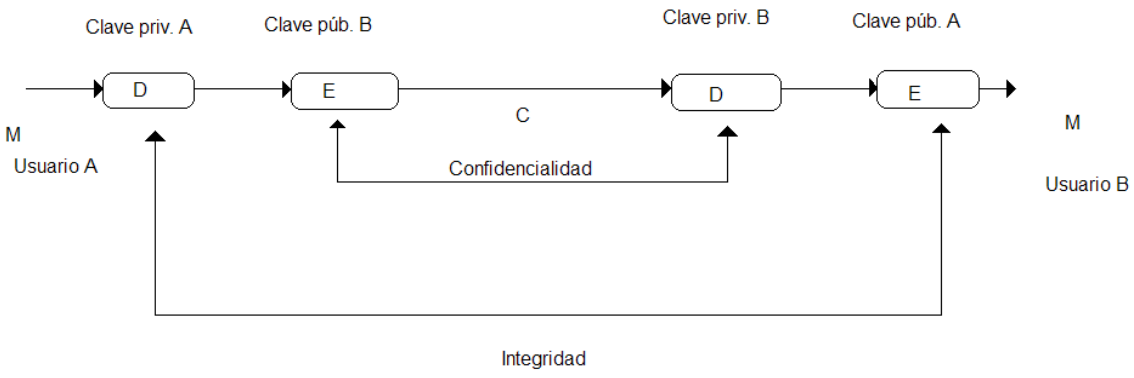

Figura 1-13. Confidencialidad e Integridad en un sistema de clave pública.

En este caso las transformaciones serán las siguientes:

$C = E_A (D_A (M))$ (operación de cifrado)

$M = E_A (D_B (C))$ (operación de descifrado)

De esta manera, se puede obtener un esquema criptográfico de clave pública que permite simultáneamente asegurar el secreto y la autenticidad del mensaje ampliamente conocido y que basa su fortaleza en la dificultad matemática que presenta factorizar números grandes.

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).

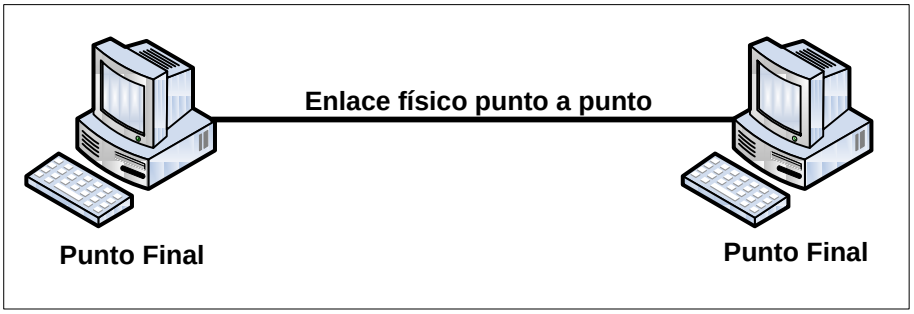

Figura 2-1. Protocolo de enlace punto a punto.

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.

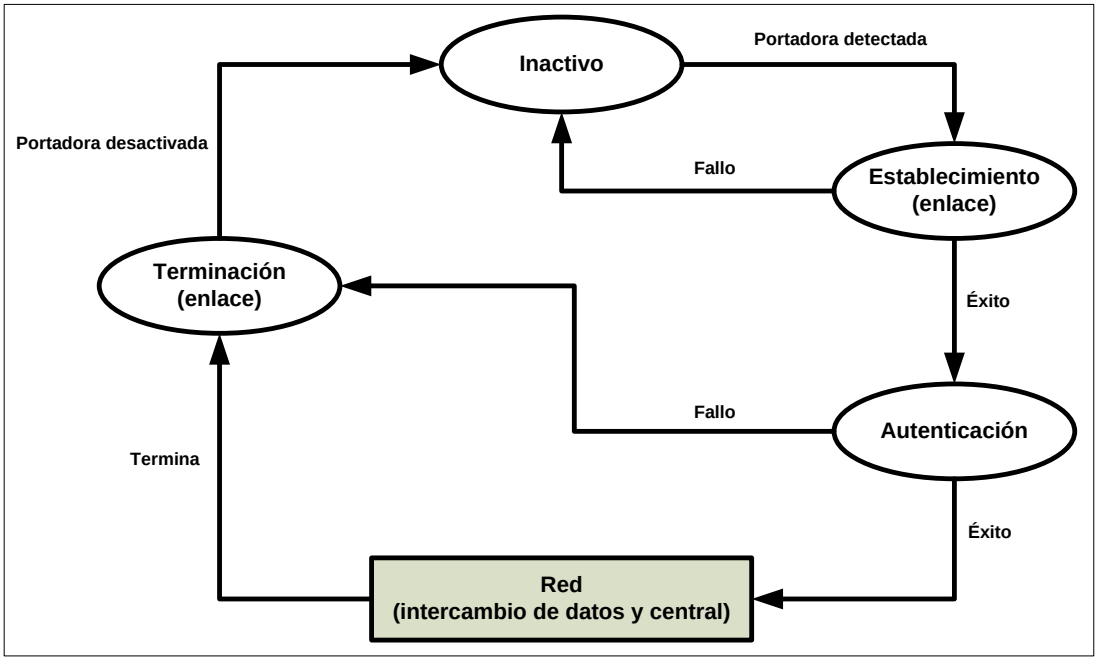

Figura 2-2. Protocolo PPP.

- 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.

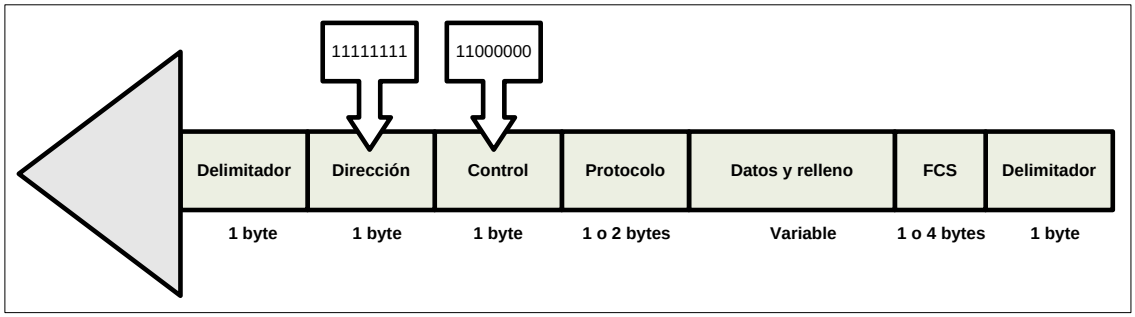

Figura 2-4. Formato de una trama del protocolo PPP.

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.

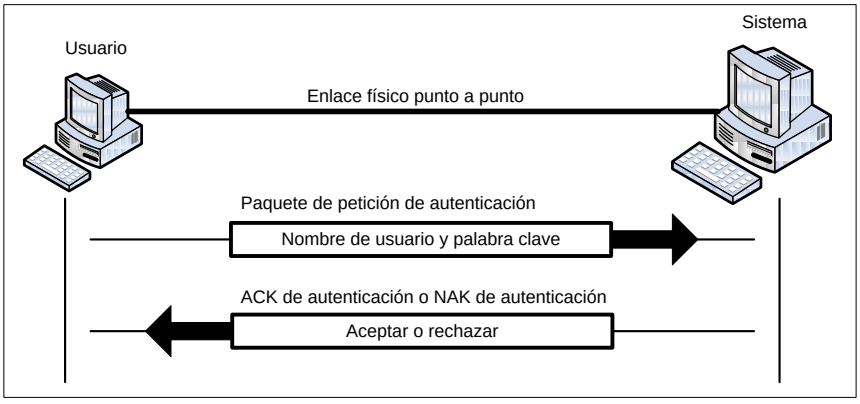

Figura 2-5. Protocolo PAP.

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.

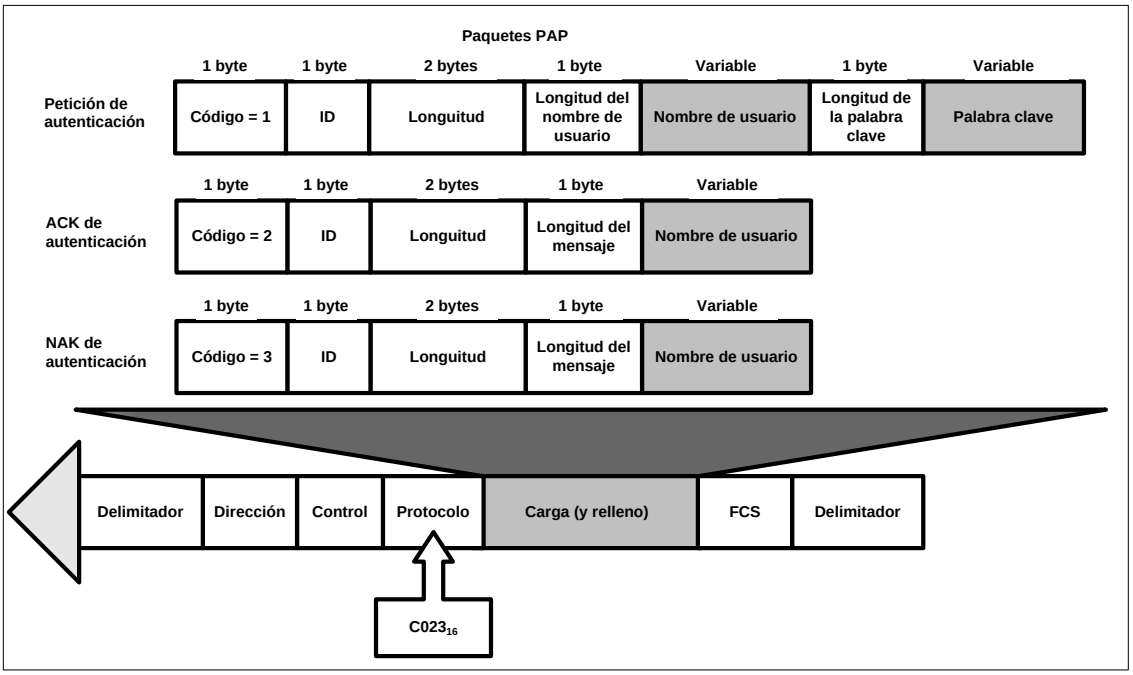

Figura 2-6. Formato de paquetes en el protocolo PAP.

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.

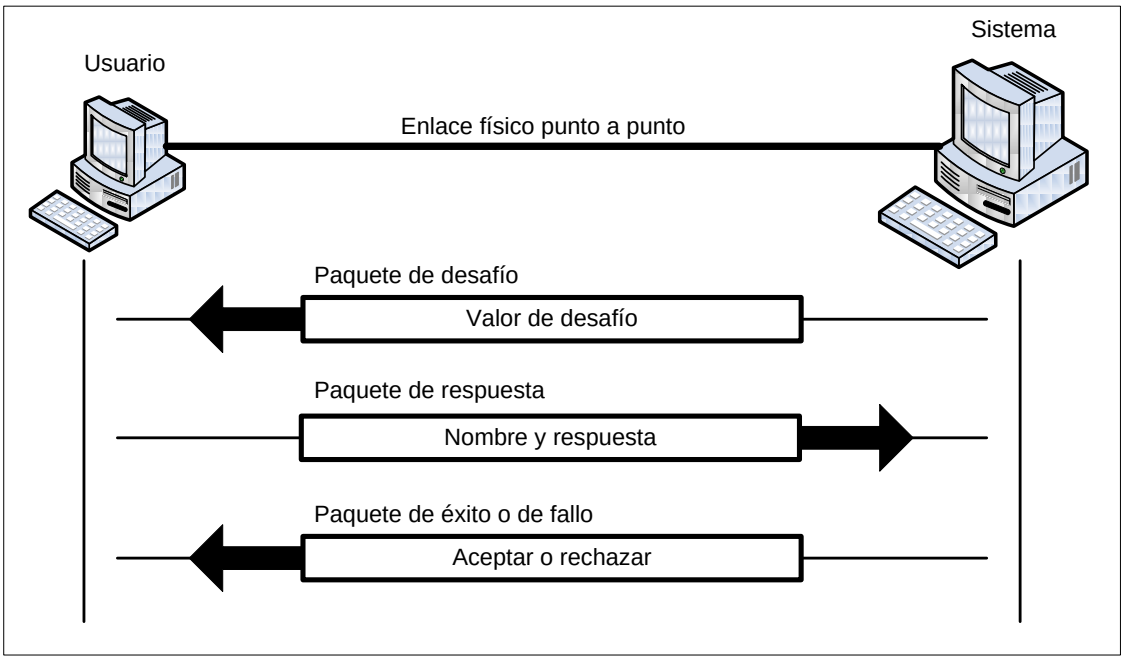

Figura 2-7. Protocolo CHAP.

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.

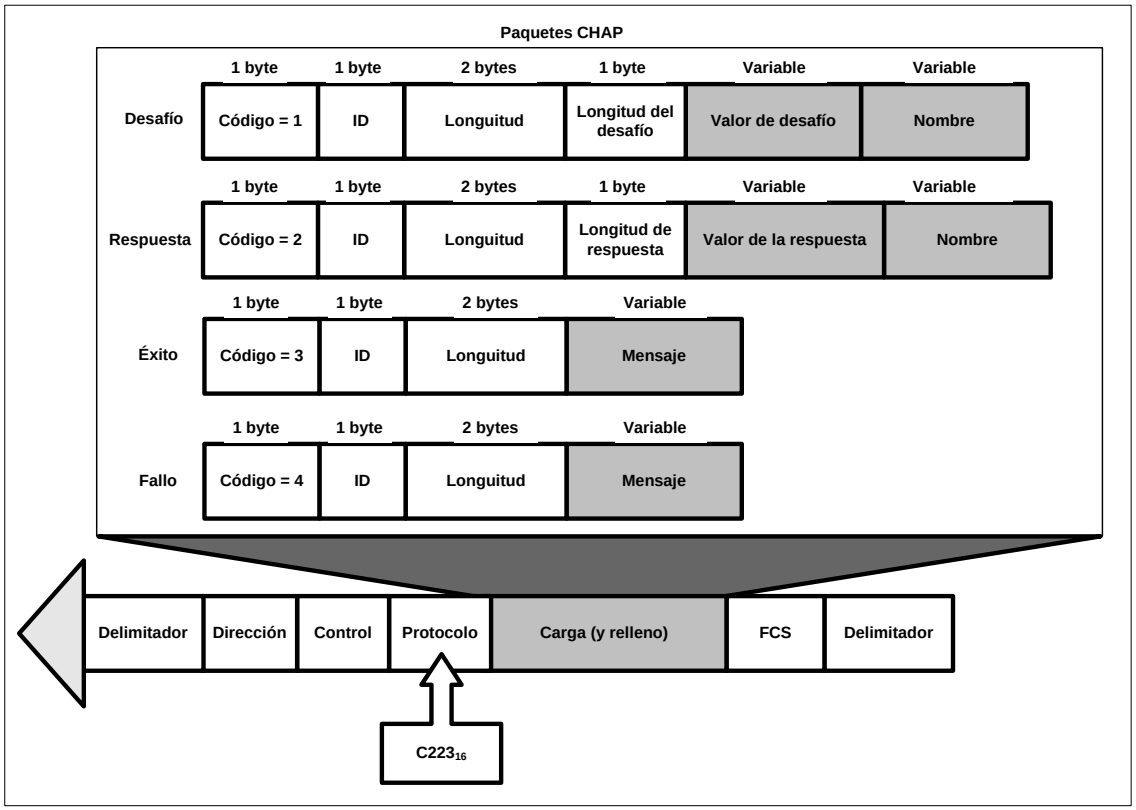

Figura 2-8. Paquetes del protocolo CHAP.

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.

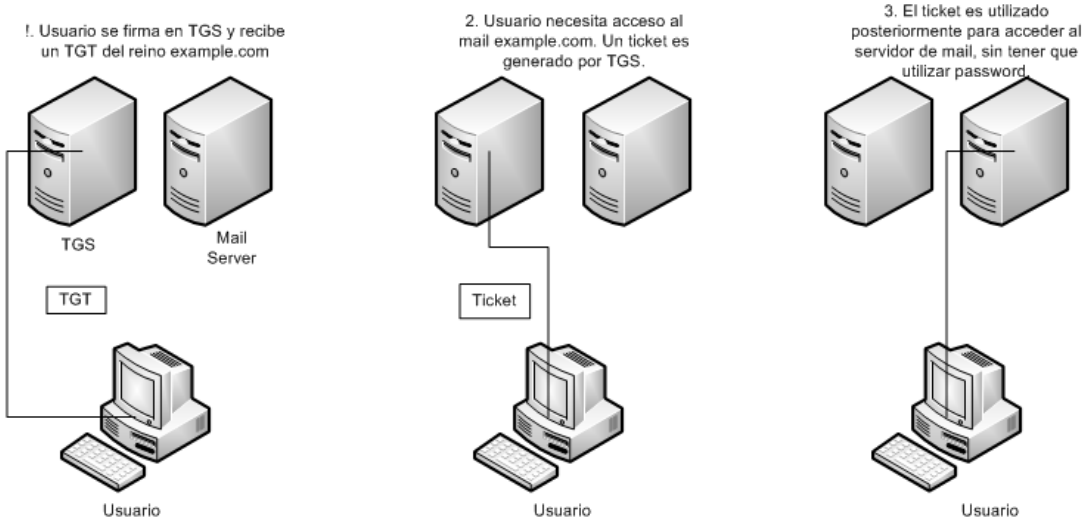

Figura 2-9. Proceso de autenticación Kerberos.

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.

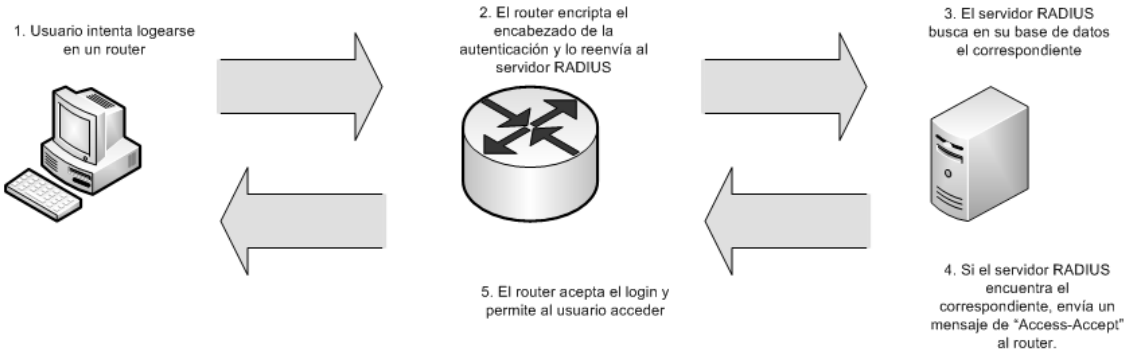

Figura 2-10. Proceso de autenticación RADIUS.

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
```

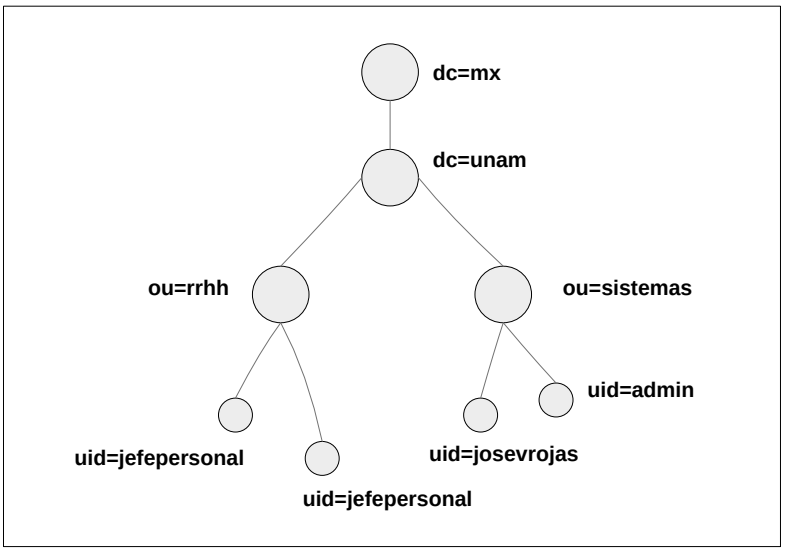

Figura 2-11. Ejemplo para protocolo LDAP.

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.

CAPÍTULO 3
IMPLEMENTACIÓN DE SERVIDORES DE AUTENTICACIÓN

3.1 KERBEROS

3.1.1 Implementación de Kerberos en Windows

Instalación y configuración de Microsoft Windows Server 2008

-Insertar DVD de instalación de Windows Server 2008, utilizando la versión Enterprise

-Proporcionar contraseña de la cuenta de administrador.

-Configurar fecha, hora y zona horaria. (GMT -06:00) Guadalajara, Ciudad de México, Monterrey. El horario es importante ya que el protocolo no funcionara si los relojes no se encuentran sincronizados.

-Configurar nombre de servidor. WIN-72NP5HN05CF

-Configurar red. Asignando una dirección IP v.4 fija. 192.168.1.88

-Instalar rol de Active Directory Domain Services.

En Administrador del Servidor, ubicar las funciones y agregar el "Servicio de dominio de Active Directory". Esto se puede observar en la Figura 3-1.

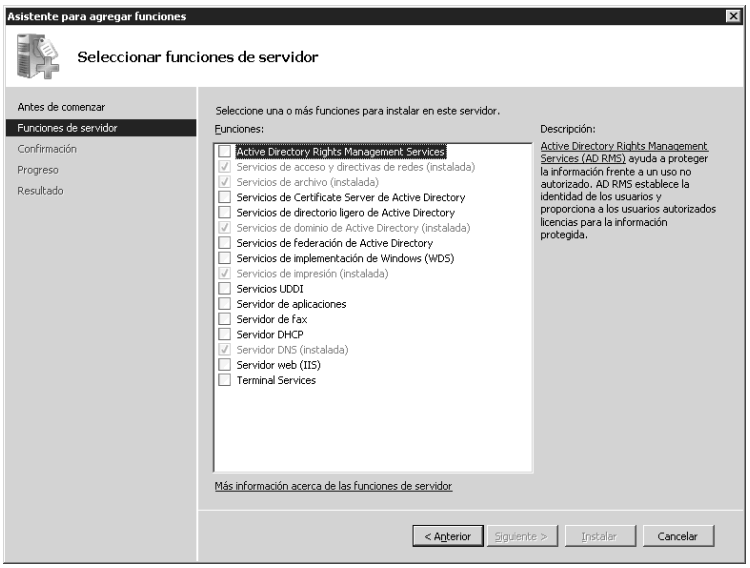

Figura 3-1. Administrador de funciones de Microsoft Windows Server 2008.

-Crear el controlador de dominio.

Para promocionar el rol antes instalado, es necesario, ejecutar el comando dcpromo.exe.

Continuar con las instrucciones y crear un nuevo dominio en un bosque nuevo.

Nombrar el bosque PROTOCOLOSFI.ORG, posteriormente se solicita instalación opcional de un servidor DNS, en este caso, como no se tiene otro servidor que realice tal función, se instala, por default al ser este el primer controlador de dominio de un bosque se instala el Catalogo Global.

-Indicar las ubicaciones de las bases de datos de los archivos log y de la página de Sysvol.

-Asignar la contraseña para la restauración de servicios del Directorio Activo.

-Finalizar y reiniciar el sistema.

-Crear usuarios de dominio y permisos.

Esta consola de administración se puede observar en la Figura 3-2.

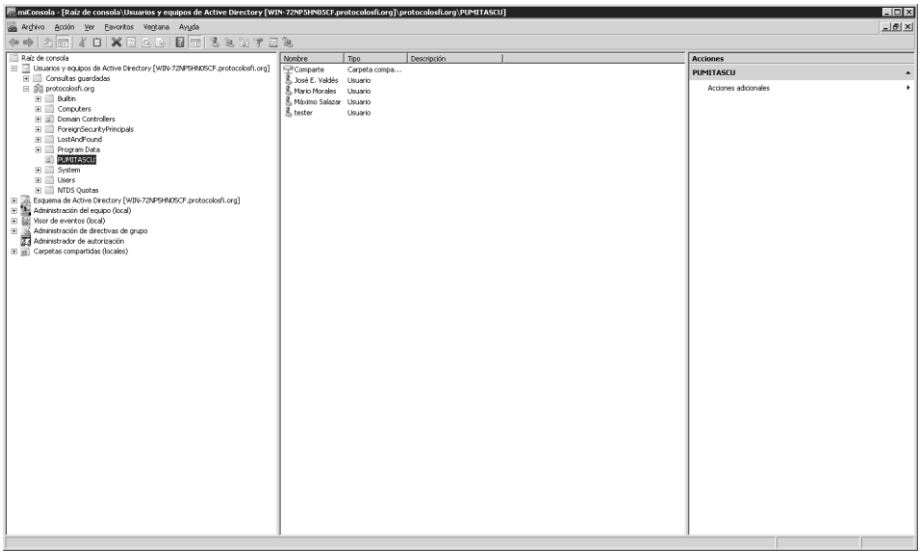

Figura 3-2. Consola de administración de servidor y usuarios.

a) Inicio de sesión de dominio en Windows XP

La configuración de usuario en un equipo con sistema operativo Windows XP, bajo el protocolo Kerberos en Microsoft Windows Server 2008, incluye agregar el equipo al dominio PROTOCOLOSFI.ORG, así como al usuario en caso de proporcionarle permisos de administrador. Esta opción se muestra en la Figura 3-3.

Figura 3-3. Acceso a Windows XP en el dominio PROTOCOLOSFI con la cuenta jvaldes.

Posteriormente, al estar con la sesión iniciada en el dominio, se puede hacer uso de cualquiera de los servicios compartidos por el servidor y con los permisos otorgados por el mismo, en este caso, una carpeta compartida y una impresora. Lo mencionado anteriormente se muestra en las Figuras 3-4 y 3-5.

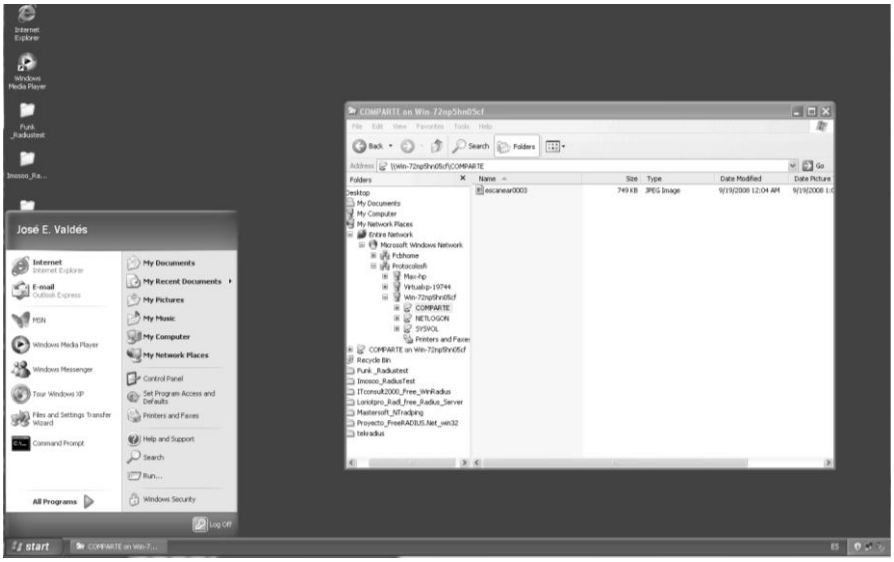

Figura 3-4. Carpeta compartida en el dominio PROTOCOLOSFI.

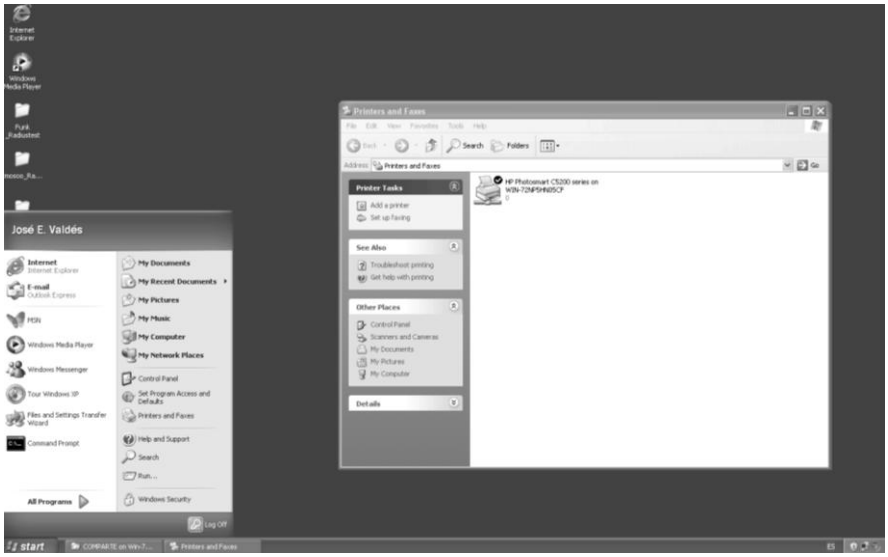

Figura 3-5. Impresora compartida por el dominio PROTOCOLOSFI.

b) Inicio de sesión de dominio en Windows 7

La configuración de usuario en un equipo con sistema operativo Windows 7, bajo el protocolo Kerberos en Microsoft Windows Server 2008, incluye agregar el equipo al dominio PROTOCOLOSFI.ORG, así como al usuario en caso de proporcionarle permisos de administrador. Esto se puede observar en la Figura 3-6.

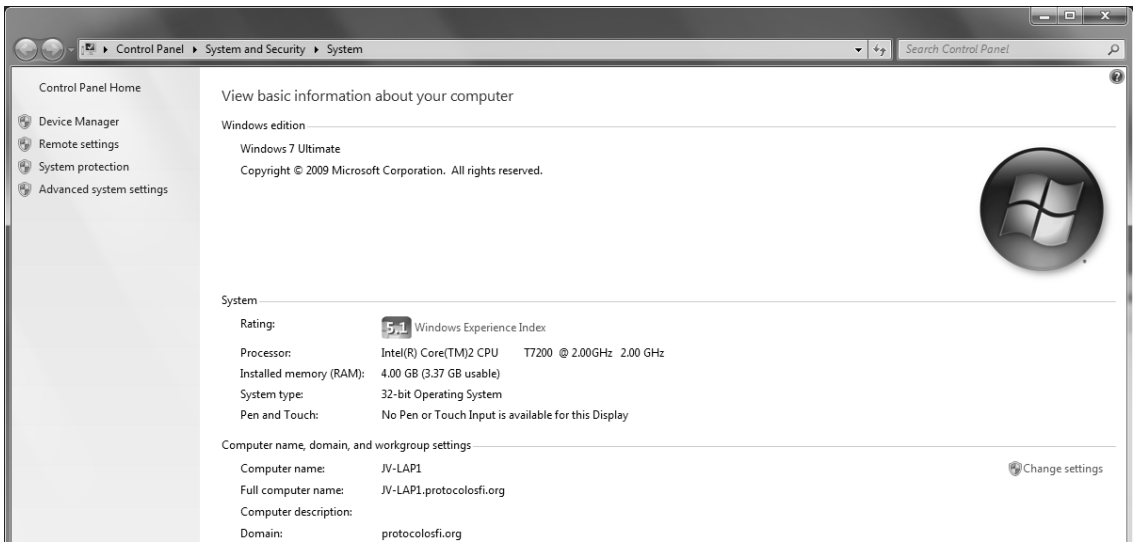

Figura 3-6. Equipo en dominio PROTOCOLOSFI.ORG.

Posteriormente, al estar con la sesión iniciada en el dominio, se puede hacer uso de cualquiera de los servicios compartidos por el servidor y con los permisos otorgados por el mismo, en este caso, una carpeta compartida y una impresora. Esto se puede observar en la Figura 3-7.

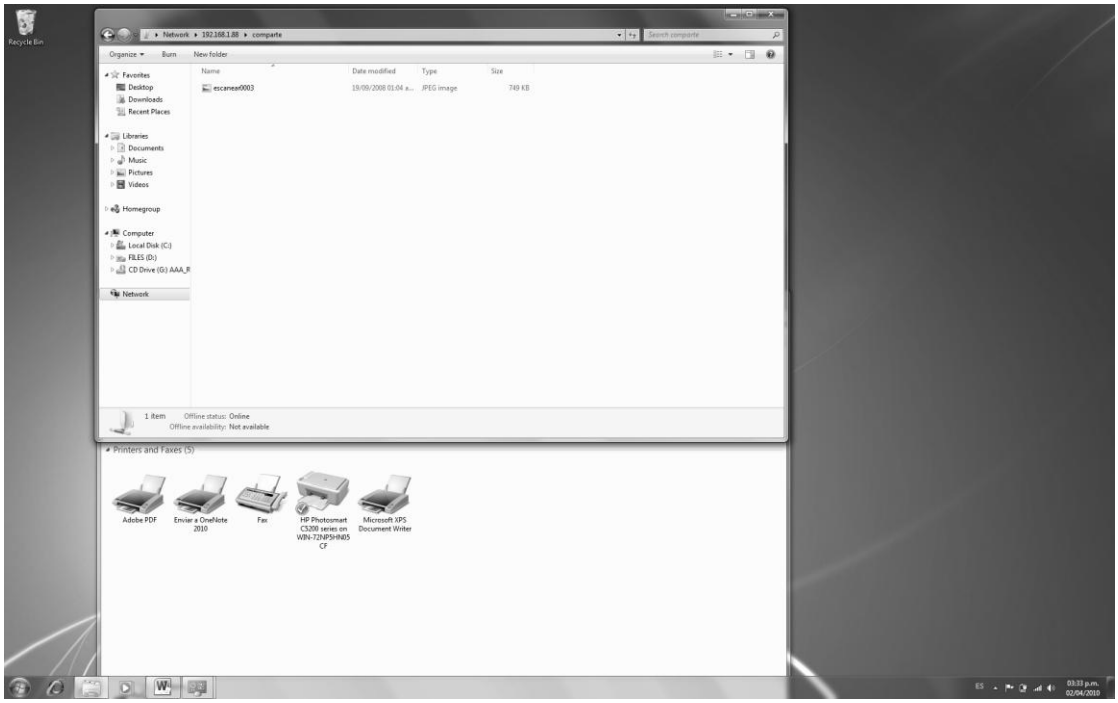

Figura 3-7. Carpeta e impresora compartida en PROTOCOLOSFI.

c) Inicio de sesión de dominio en Ubuntu Linux

Para agregar un equipo con sistema operativo Linux, se procede a instalar el programa Likewise, con lo cual es fácil establecer la relación entre el servidor de dominio y el equipo de cómputo, aunque no pertenezcan al mismo ambiente.

Es necesario descargar el paquete LikewiseOpen-4.1.0.1846-linux e instalarlo o simplemente agregarlo por medio de Synaptic.

Finalmente es necesario configurar el programa Likewise con el dominio y la sesión a utilizar. Esta consola de configuración se puede observar en la Figura 3-8.

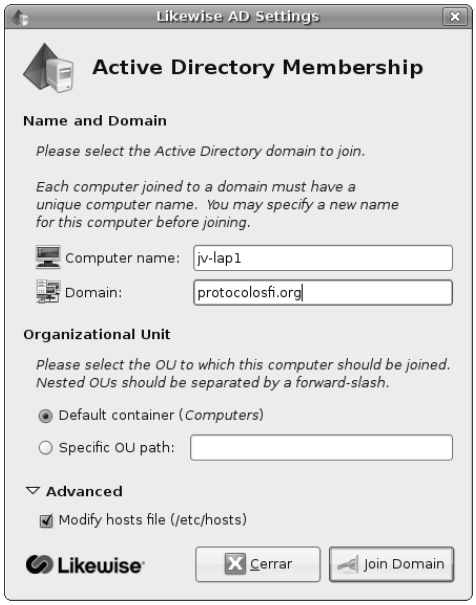

Figura 3-8. Configuración de Likewise en PROTOCOLOSFI.

Se ingresa al dominio dando clic en "Join Domain" y posteriormente se ingresa la contraseña y se verifica que el equipo se encuentra en el dominio correspondiente. Esto se muestra en la Figura 3-9.

Figura 3-9.Dominio PROTOCOLOSFI.ORG

Una vez ingresado en el dominio, en éste se encuentran los datos del perfil y se puede verificar que el usuario y el equipo pertenecen al dominio. Estos datos se muestran en la Figura 3-10.

Figura 3-10. Propiedades de usuario en Ubuntu, del dominio PROTOCOLOSFI.

Mediante la consola de Linux, y utilizando el comando kinit y el nombre de usuario, se prueba que el protocolo funciona y se ingresa correctamente recibiendo el ticket de kerberos, esto se muestra en la Figura 3-11.

```
root@jv-lap1:~# kinit jvaldes
jvaldes@PROTOCOLOSFI.ORG's Password:
root@jv-lap1:~# klist
Credentials cache: FILE:/tmp/krb5cc_0
        Principal: jvaldes@PROTOCOLOSFI.ORG

  Issued           Expires          Principal
Apr  2 23:34:18  Apr  3 09:34:05  krbtgt/PROTOCOLOSFI.ORG@PROTOCOLOSFI.ORG
```

Figura 3-11. Ticket de Kerberos en consola de Ubuntu.

3.1.2 Implementación de Kerberos en UNIX

La instalación del servicio de autenticación Kerberos se realizará en el Sistema Operativo Ubuntu. Para comenzar a levantar este servicio será necesario previamente instalar en un equipo de cómputo el sistema operativo mencionado. La instalación y configuración inicial de Ubuntu 8.04 se precisa con mayor detalle en el Anexo 1.

Instalado el Sistema Operativo Ubuntu en el equipo de cómputo, lo siguiente a realizar es instalar el servidor de autenticación Kerberos.

Si se está estableciendo un nuevo reino Kerberos, se tendrá que empezar por la elección de una implementación de Kerberos para el KDC. Hay muchos KDC disponibles de diferentes proveedores, tanto comerciales como de código abierto. Cada aplicación KDC es diferente, con ventajas y desventajas sobre los demás.

Para este caso la elección del KDC debe ser de código abierto y de distribución libre. Dentro de las distribuciones KDC que cumplen estas características son MIT y Heimdal. Se elegirá como el KDC para instalación y configuración la distribución proporcionada por MIT (Massachusetts Institute of Technology), esto porque el KDC de MIT es la primera y aún la implementación de referencia para Kerberos 4 y 5. Muchas grandes entidades, la mayoría universidades, usan como KDC la autenticación con MIT. Hay una gran base de apoyo para el MIT de Kerberos y se utiliza en muchos ambientes, que ayuda a ejercitar los errores del sistema. MIT Kerberos contiene soporte para los estándares de encriptación de Kerberos, en particular para simple y triple DES. Además, la versión 1.8 (la más reciente) del MIT Kerberos, incluye soporte para el tipo de encriptación RC4 utilizado por el servicio Active Directory de Microsoft Kerberos, así como el nuevo Advanced Encryption Standard (AES).

Kerberos requiere del buen funcionamiento de varios servicios externos. En particular los relojes de todos los equipos participantes deben estar sincronizados a pocos minutos. En primer lugar, el NTP (Network Time Protocol) debe estar instalado en cada servidor y el KDC en la red. El NTP sincroniza los relojes de las máquinas a una fuente central el cual puede ser un recurso de hora local. Si bien es posible configurar para sincronizar todas las máquinas a través de la red, debe haber un servidor de tiempo a disposición del público, donde para los administradores del sitio se recomienda realizar la creación de una fuente de tiempo centralizado en la red y crear otras máquinas en la red para sincronizar con el servidor. La máquina de servidor de tiempo se puede sincronizar a una fuente de reloj de precisión, como un servidor de tiempo público.

Los detalles para implementar el NTP en la red de datos no se contempla en esta explicación de la implementación de Kerberos, pero si se debe contemplar este punto en la implementación ya que relojes mal sincronizados es la raíz de muchos problemas que se generan en el uso de Kerberos a pesar que las implementaciones de Kerberos prevén normalmente unos más o menos cinco minutos de error al comparar los tiempos. Los sistemas Unix no suelen tener NTP instalado y configurado por lo que la configuración de forma manual es necesaria para mantener los relojes de estos sistemas en sincronización. [11]

Dado que la distribución MIT de Kerberos está disponible como código abierto, hay dos maneras de instalarlo: la construcción desde el código fuente o la obtención de una distribución binaria de su proveedor de Unix. Se va a cubrir la instalación por código fuente y dicha distribución del MIT está disponible en la página principal de MIT Kerberos, ubicado en http://web.mit.edu/network/kerberos-form.html.

Una vez descargado el archivo de la página de Internet mencionada, éste está comprimido del siguiente modo krb5-1.8-signed.tar. Para descomprimir este archivo se ejecuta lo siguiente:

```
home/mario# tar krb5-1.8-signed.tar
```

Tras ser descomprimido se obtienen los archivos krb5-1.8.tar.gz.asc y krb5-1.8.tar.gz, el primer archivo mencionado servirá para verificar la versión, esto se obtiene ejecutando lo siguiente:

```
/home/mario# gpg --verify krb5-1.8.tar.gz.asc
```

El resultado que arroja es el siguiente:

```
gpg: Firmado el mar 02 mar 2010 12:24:50 CST usando clave RSA ID
F376813D
```

En este caso como se cuenta con la versión más reciente, no es reemplazado el archivo por alguna otra versión. Lo siguiente a realizar es descomprimir el archivo krb5-1.8.tar.gz. Esto se obtiene ejecutando en *prompt* lo siguiente:

```
/home/mario# tar xzvf krb5-1.8.tar.gz
```

Esto crea un directorio krb5-1.8 en la ubicación donde se descomprimió el archivo, en esta ruta se colocan todos los archivos de la distribución de código fuente de Kerberos. Dentro del directorio krb5-1.8, hay un directorio de nombre /src y otro directorio /doc. Dado que se pretende construir la distribución, ingresar al directorio krb5-1.8/src.

Para compilar la distribución se ejecuta lo siguiente:

```
# ./configure && make
```

Una vez que la compilación está completa, instalar el software:

```
# make install
```

El paso de instalación establece la estructura de directorios bajo su prefijo (/usr/local por default) y en lugares binarios, incluye archivos, bibliotecas y en sus lugares apropiados en el directorio prefijo.

Tener en cuenta que el proceso de instalación no crea el directorio (/usr/local/var por default). Por lo que se tendrá que crear este directorio antes de ejecutar el make install, así como crear un directorio krb5kdc debajo de ella. Además, se tiene que establecer permisos en el directorio krb5kdc para garantizar que los usuarios no autorizados no pueden acceder a datos sensibles del KDC. Para crear esto se ejecuta en prompt lo siguiente:

```
# mkdir /usr/local/var
# mkdir /usr/local/var/krb5kdc
# chown root /usr/local/var/krb5kdc
# chmod 700 /usr/local/var/krb5kdc
```

3.1.2.1 Creación del reino

Ahora se va a realizar una nueva instalación de Kerberos para crear el reino. Este paso sólo se realizará en el KDC principal. Crear los archivos necesarios de base de datos y llenar la base de datos KDC son los principios necesarios. En primer lugar, se necesitan crear algunos archivos de configuración.

Estrictamente la única configuración de archivo que se necesita es crear el archivo krb5.conf. El archivo krb5.conf vive en /etc y contiene los parámetros que se utilizan para las librerías de Kerberos. El archivo krb5.conf tiene una apariencia similar a un archivo Windows del tipo in, con estrofas (o grupos) entre corchetes y los key-value separados por un signo igual. En este punto todo lo que se necesita para este archivo es lo siguiente:

```
[libdefaults]
    ticket_lifetime = 24000
    default_realm = PROTOCOLOSFI.ORG
    dns_lookup_realm = false
    dns_lookup_kdc = false

[realms]
  PROTOCOLOSFI.ORG = {
            kdc = kdc.protocolosfi.org
            admin_server = kdc.protocolosfi.org
            default_domain = protocolosfi.org

  }

[domain_realm]
    .protocolosfi.org = PROTOCOLOSFI.ORG
```

```
        protocolosfi.org = PROTOCOLOSFI.ORG

    [logging]
        default = FILE:/usr/local/var/krb5kdc/krb5lib.log
        kdc = FILE:/usr/local/var/krb5kdc/krb5kdc.log
        admin_server = FILE:/usr/local/var/krb5kdc/kadmin.log

    [kdc]
        profile = /usr/local/var/krb5kdc/kdc.conf

    [pam]
        debug = false
        ticket_lifetime = 36000
        renew_lifetime = 36000
        forwardable = true
        krb4_convert = false
```

Otro archivo de configuración que se va a crear es el archivo de configuración del KDC, este archivo tiene el nombre de kdc.conf.

Si no se ha modificado el prefijo o las opciones del localstatedir en el proceso de configuración, este archivo se encuentra en la ruta /usr/local/var. En general los archivo de la las bases de datos KDC viven bajo el directorio krb5kdc. La forma general de archivo kdc.conf es el siguiente:

```
    [kdcdefaults]
        kdc_ports = 88

    [realms]
     PROTOCOLOSFI.ORG = {
        kadmind_port = 749
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        admin_keytab = /usr/local/var/krb5kdc/kadm5.keytab
        acl_file = /usr/local/var/krb5kdc/kadm5.acl
        master_key_type = des3-hmac-sha1
        supported_enctypes = des3-hmac-sha1:normal des-cbc-crc:normal
des-cbc-cr
c:v4
        kdc_supported_enctypes = des3-hmac-sha1:normal des-cbc-
crc:normal des-cb
c-crc:v4
     }

    [logging]
        kdc = FILE:/usr/local/var/krb5kdc/kdc.log
        admin_server = FILE:/usr/local/var/krb5kdc/kadmin.log
```

Con los archivos de configuración realizados se está listo para inicializar la base de datos Kerberos. Para realizar este paso se va a utilizar el programa kdb5_util, incluido con la distribución de Kerberos. Este programa realiza diversas tareas administrativas en la base de datos Kerberos. Por ahora, se utilizará el parámetro create para crear una base de datos y con esto crear el esquema del nuevo reino.

Con la siguiente línea de comandos se crea un nuevo reino:

```
# /usr/local/sbin/kdb5_util create –s
```

La opción -s indica que se estará usando un archivo de contraseñas Kerberos que pide la clave maestra. A continuación se mostrará el cuadro de diálogo que se presenta durante la ejecución del programa kdb5_util cuando se ejecuta el comando create:

```
# /usr/local/sbin/kdb5_util create -s
Initializing database '/usr/local/var/krb5kdc/principal' for realm
'protocolosfi.org',
master key name 'K/M@protocolosfi.org'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

Una vez creado el reino, agregar el archivo kadm5.acl en la ruta /usr/local/var/krb5kdc/kadm5.acl . Este archivo contendrá lo siguiente:

```
*/admin@PROTOCOLOSFI.ORG       *
```

Finalmente indicar el KDC (krb5kdc) del siguiente modo:

```
# krb5kdc
# krb5kdc start
```

Iniciar el kadmin de manera local ejecutando lo siguiente:

```
# /usr/local/sbin/kadmin.local
Authenticating as principal minclan/admin@PROTOCOLOSFI.ORG with
password.
      kadmin.local:
```

Una vez dentro de kadmin.local, se pueden enumerar los *principals* que se tienen actualmente en el KDC:

```
kadmin.local:  listprincs
K/M@PROTOCOLOSFI.ORG
kadmin/history@PROTOCOLOSFI.ORG
krbtgt/PROTOCOLOSFI.ORG@PROTOCOLOSFI.ORG
```

El comando ank sirve para añadir algunos clientes al KDC:

```
kadmin.local:  ank minclan
WARNING:  no  policy  specified  for  minclan@PROTOCOLOSFI.ORG;
defaulting to no
policy
Enter password for principal "minclan@PROTOCOLOSFI .ORG":
Re-enter password for principal "minclan@PROTOCOLOSFI .ORG":
    Principal "minclan@PROTOCOLOSFI .ORG" created.


kadmin.local:  ank jvaldes
WARNING:  no  policy  specified  for  jvaldes@PROTOCOLOSFI.ORG;
defaulting to no
```

```
    policy
    Enter password for principal "jvaldes@PROTOCOLOSFI .ORG":
    Re-enter password for principal "jvaldes@PROTOCOLOSFI .ORG":
        Principal "jvaldes@PROTOCOLOSFI .ORG" created.


    kadmin.local:  ank mwindows
    WARNING:  no  policy  specified  for  mwindows@PROTOCOLOSFI.ORG;
defaulting to no
    policy
    Enter password for principal " mwindows@PROTOCOLOSFI .ORG":
    Re-enter password for principal " mwindows@PROTOCOLOSFI .ORG":
        Principal " mwindows @PROTOCOLOSFI .ORG" created.
```

Para observar los clientes recién dados de alta al KDC se ejecuta nuevamente el comando listprincs:

```
    kadmin.local:  listprincs
    K/M@PROTOCOLOSFI.ORG
    jvaldes@PROTOCOLOSFI.ORG
    mwindows @PROTOCOLOSFI.ORG
    kadmin/history@PROTOCOLOSFI.ORG
    krbtgt/PROTOCOLOSFI.ORG@PROTOCOLOSFI.ORG
    minclan@PROTOCOLOSFI.ORG
```

Una vez vistos en la lista que arroja el comando listprincs los clientes que se dieron de alta en la Base de Datos, se da por terminada la fase de configuración del KCD Kerberos.

En el equipo de cómputo que está actuando como el servidor Kerberos, en su archivo hosts ubicado en la ruta /etc, modificarlo por el editor vi y agregar las siguientes líneas:

```
   127.0.0.1 localhost
   192.168.1.101 kdc.protocolosfi.org
   192.168.1.2 jvaldes.protocolosfi.org
   192.168.1.3 minclan.protocolosfi.org
   192.168.1.4 mwindows.protocolosfi.org
```

Esto actuará como servidor DNS y el servidor ubicará los equipos de cómputo que tendrá como clientes.[12]

3.1.2.2 Configuración de clientes en Linux

Para llevar a cabo la configuración de un cliente bajo el sistema Linux realizar lo siguiente:

1) Copiar el archivo de configuración krb5.conf del servidor KDC que vive en la ruta /etc a los equipos de cómputo cliente en la ruta /etc.
2) En los equipo de cómputo que estén actuando como clientes, en su archivo

hosts ubicado en la ruta /etc agregar las siguientes líneas:

```
127.0.0.1 localhost
192.168.1.101 kdc.protocolosfi.org kdc
```

3.1.2.3 Configuración de clientes en Windows

Para llevar a cabo la configuración de un cliente bajo el sistema Windows realizar lo siguiente:

1) En el equipo cliente bajo el sistema operativo Windows se descargará el archivo .exe y el archivo .bin para Windows de la página de internet http://web.mit.edu/kerberos/www/.
2) Ejecutar el archivo .exe dando doble clic al icono del programa cliente Kerberos. El archivo .exe es similar al icono que se muestra en la Figura 3-12.

Figura 3-12 Icono ejecutable del programa cliente Kerberos.

3) Pasar los archivos dll del archivo .bin descargado a la ruta C:\WINDOWS\system32. Estos archivos se muestran en la Figura 3-13.

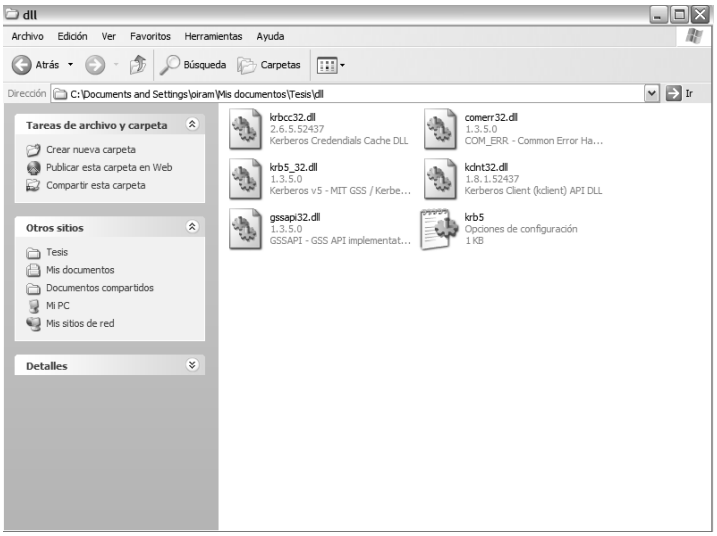

Figura 3-13. Archivos DLL para configuración de cliente Kerberos.

También pasar el archivo krb5.ini con la misma configuración que el archivo krb5.conf que se creó en la ruta /etc en el KDC a la ruta C:\WINDOWS\system32

4) Ir a la ruta C:\WINDOWS\system32\drivers\etc y editar el archivo hosts. En este archivo agregar la dirección IP del KDC. Este archivo de configuración se

muestra en la Figura 3-14.

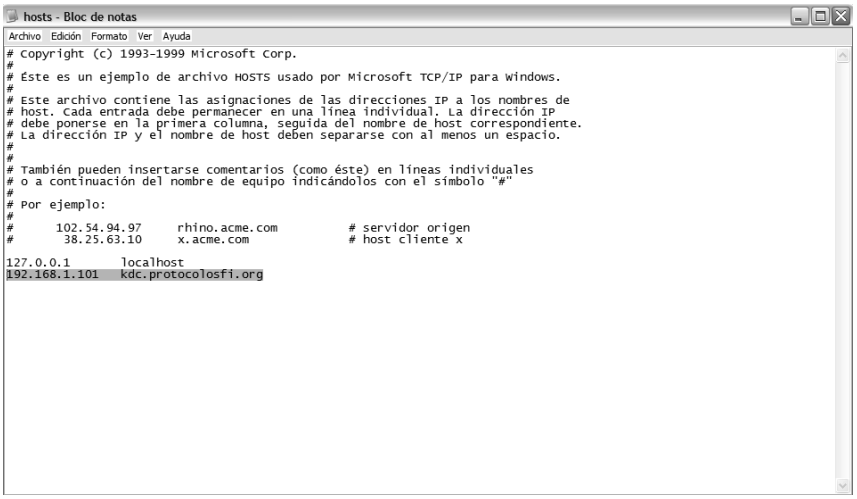

Figura 3-14. Archivos de configuración hosts.

5) Abrir el programa Leash para iniciar el programa de Kerberos para Windows. Éste se encuentra ubicado en la ruta que se muestra en la Figura 3-15:

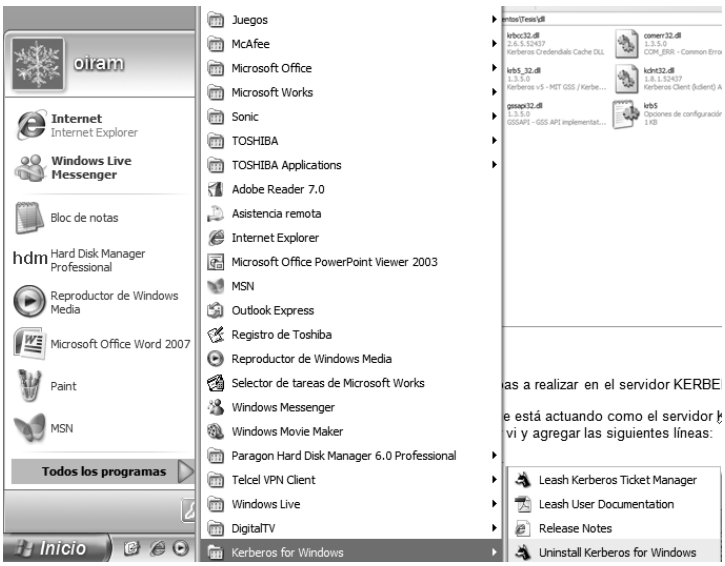

Figura 3-15. Ruta de programa Leash.

Una vez abierto el programa Leash, éste se visualiza de modo que muestra la Figura 3-16:

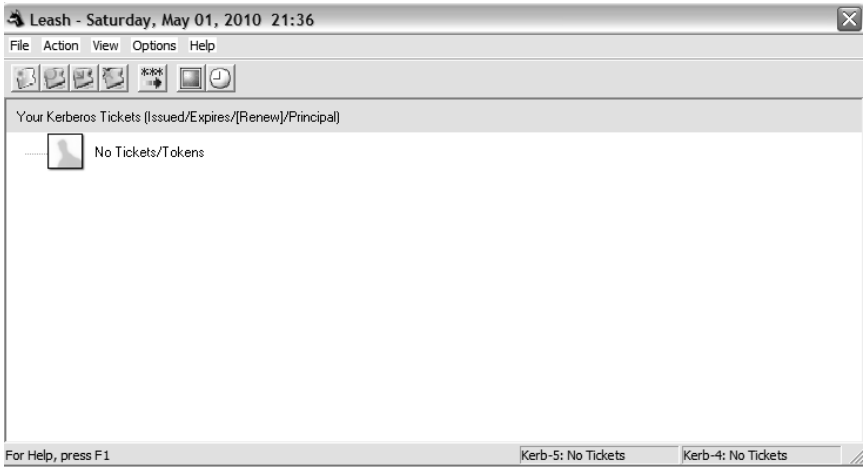

Figura 3-16. Ventana inicial de programa Leash.

6) Configurar las opciones del Leash para que reconozca el reino PROTOCOLOSFI.ORG, esto se hace yendo a la pestaña Options y posterior a la pestaña Kerberos v5 Properties. Esto se visualiza en la Figura 3-17.

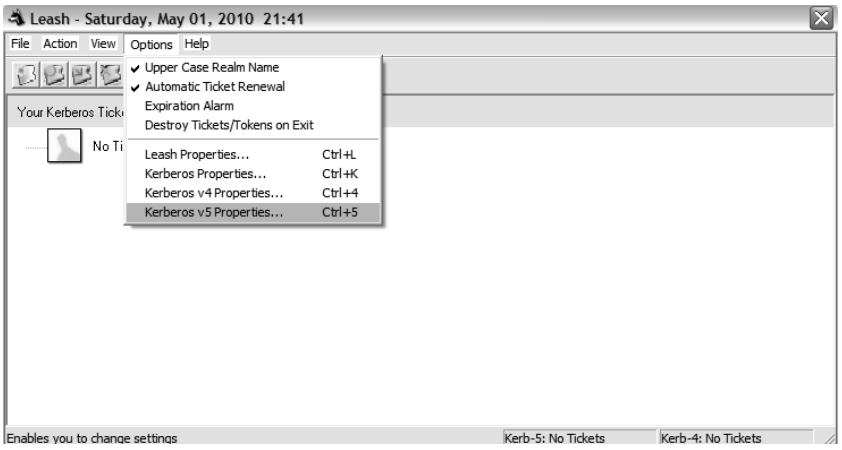

Figura 3-17. Ruta para configurar ruta del KDC.

7) Una vez abierta dicha ventana, cambiar la ruta dada por default por la ruta donde se encuentra el archivo de configuración krb5.ini que se copió en la ruta C:\WINDOWS\system32. Esto se visualiza en la Figura 3-18.

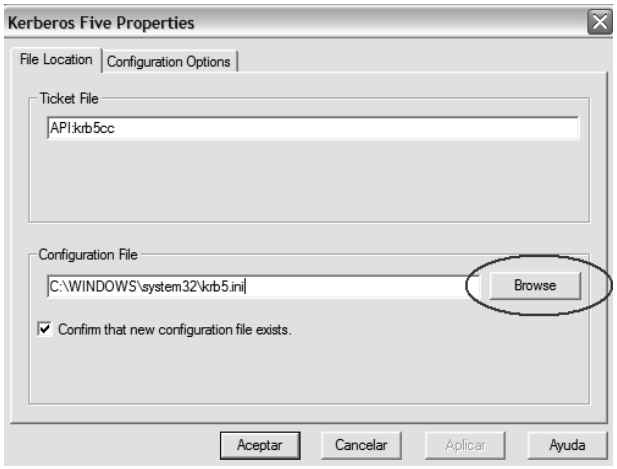

Figura 3-18. Ruta para configurar KDC de Kerberos.

3.1.2.4 Pruebas de conexión con servidor Kerberos

Para hacer algunas pruebas al servidor KERBEROS con clientes bajo sistemas operativos Linux y Windows realizar lo siguiente:

Los clientes que se intenten conectar al servidor Kerberos podrán realizarlo sin necesidad de estar dentro de un DNS. Tener en cuenta que hay que configurar sus respectivas IP en los equipos de cómputo que conforman la red del modo que muestra la Tabla 3-19.

IP address	Hostname	Tipo de Equipo
192.168.1.101	kdc.protocolosfi.org	Servidor
192.168.1.2	jvaldes.protocolosfi.org	Cliente Linux
192.168.1.3	minclan.protocolosfi.org	Cliente Linux
192.168.1.4	mwindows.protocolosfi.org	Cliente Windows

Tabla 3-19. Ruta para configurar KDC de Kerberos.

Una vez configurados los equipos de cómputo clientes con su respectiva IP, se podrán realizar las siguientes pruebas de conexión:

Prueba 1)

Autenticar un equipo de cómputo cliente bajo sistema operativo UNIX con el servidor Kerberos. En el equipo de cómputo que actuará como cliente ejecutar lo siguiente:

```
root@oiram-lap:/etc# kinit minclan
Password for minclan@PROTOCOLOSFI.ORG:
```

Si no se arroja en el prompt algún error, la autenticación pudo realizarse sin ningún problema. Para corroborar esto, ejecutar en el equipo de cómputo cliente lo siguiente:

```
root@oiram-lap:/etc# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: minclan@PROTOCOLOSFI.ORG

Valid starting     Expires            Service principal
04/02/10         21:27:42                 04/03/10         21:27:42
krbtgt/PROTOCOLOSFI.ORG@PROTOCOLOSFI.ORG
          renew until 04/02/10 21:27:42

Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

Con esto se observa que el equipo de cómputo cliente ha obtenido los Tickets necesarios para poder autenticarse con el servidor Kerberos.

Prueba 2)

Realizar prueba con un equipo de cómputo bajo sistemas operativos UNIX con un USUARIO NO DADO DE ALTA en el servidor Kerberos. Ejecutar lo siguiente en algún equipo de cómputo cliente:

```
root@oiram-lap:/usr/local/bin# kinit msalazar
kinit(v5): Client not found in Kerberos database while getting
initial
credentials
```

Con esto se observa que algún usuario que no esté dado de alta dentro en la base de datos del KDC, no podrá autenticarse con el servidor.

Prueba 3)

Realizar prueba con un equipo de cómputo bajo sistemas operativos UNIX con un USUARIO EXISTENTE PERO CON CONTRASEÑA DISTINTA A LA DADA DE ALTA en la base de datos del KDC. Ejecutar lo siguiente en algún equipo de cómputo cliente:

```
root@oiram-lap:/usr/local/bin# kinit minclan
Password for minclan@PROTOCOLOSFI.ORG:
kinit(v5): Password incorrect while getting initial credentials
```

Esta prueba demuestra que no puede autenticarse con el servidor Kerberos un

usuario si ingresa de manera errónea su respectivo *password*, a pesar que el usuario exista en la base de datos.

Prueba 4)

Realizar pruebas de usuario bajo el sistema operativo Windows XP utilizando el usuario mwindows. Una vez que se inició sesión en un equipo de cómputo que actuará como cliente bajo el sistema operativo Windows XP y que ya tiene configurado e instalado el programa MIT Kerberos para Windows, se puede observar que tiene instalado dicho programa si se observa el icono del programa cliente Kerberos Leash tal como lo muestra la Figura 3-20.

Figura 3-20. Icono de programa Leash instalado en equipo de cómputo.

Iniciar el programa Leash, una vez abierto ir a la pestaña Action y posteriormente abrir la pestaña Get Tickets(s) para poder iniciar sesión con el principal dado de alta en la base de datos del KDC en el reino PROTOCOLOFI.ORG. Esto lo muestra la Figura 3-21.

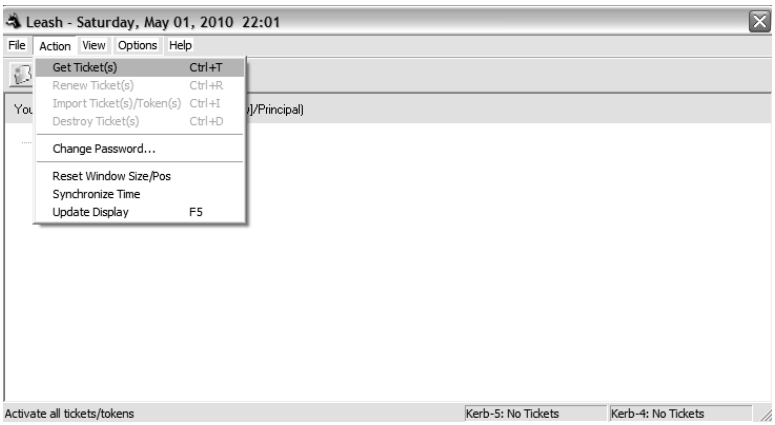

Figura 3-21. Ruta para abrir opción "Initialize ticket".

Esta pestaña abre la ventana que se muestra en la Figura 3-22.

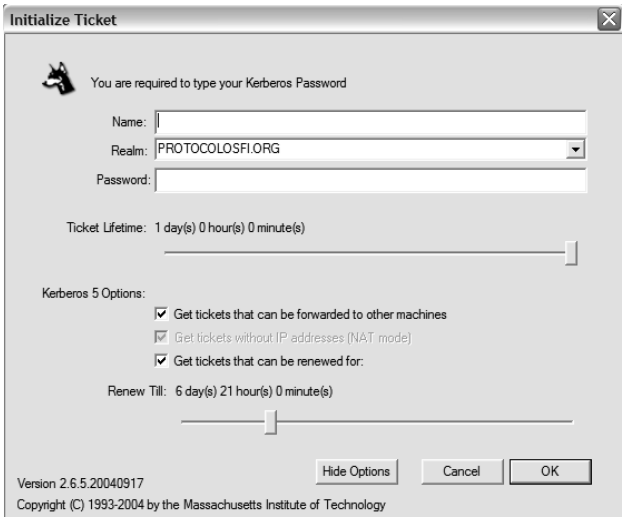

Figura 3-22. Ventana "Initialize ticket".

En esta ventana ingresar los datos del principal para poder iniciar el intercambio de tickets entre el KDC y el cliente, un ejemplo de lo mencionado anteriormente se muestra en la Figura 3-23.

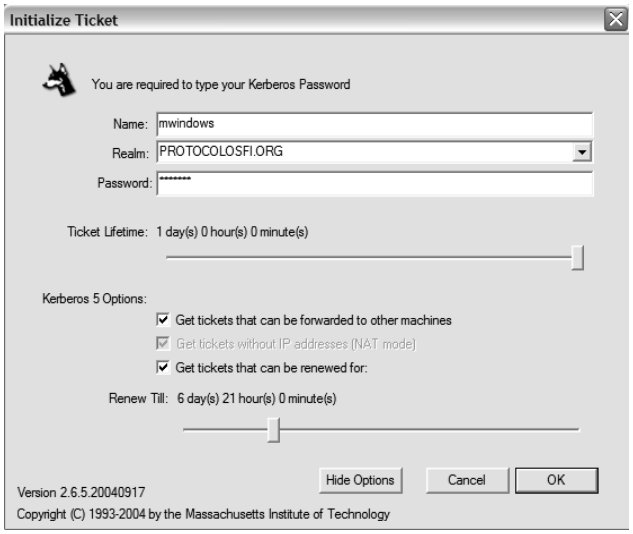

Figura 3-23. Configuración de datos del cliente en ventana "Initialize ticket".

Una vez ingresados de manera correcta los datos del principal, oprimir el botón OK. Observar que el icono del Leash ha cambiado al estatus de conectado tal como lo muestra la Figura 3-24:

Figura 3-24. Estatus de conexión afirmativo con reino Kerberos.

El programa del Leash muestra el intercambio de tickets entre el KDC y el principal mwindows. Esto se muestra en las Figuras 3-25 y 3-26:

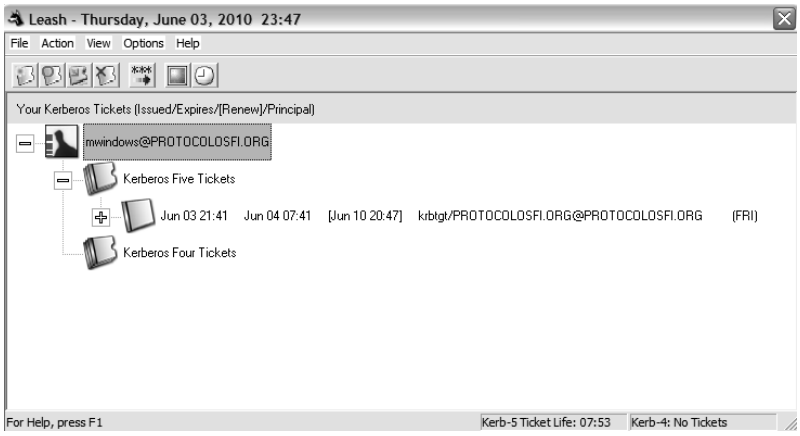

Figura 3-25. Intercambio de tickets con servidor Kerberos.

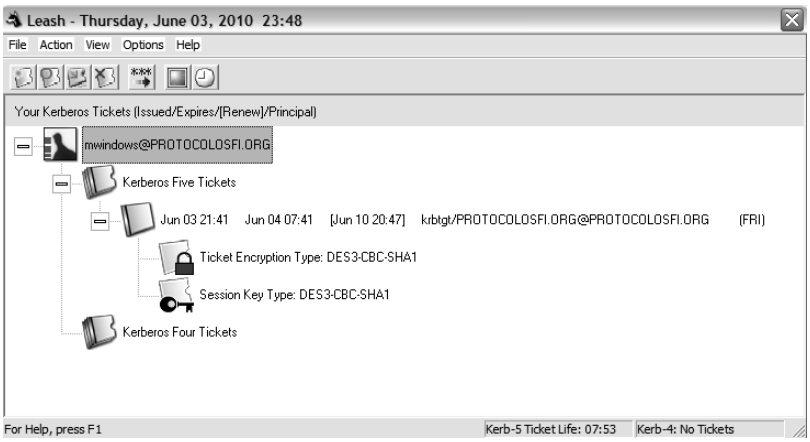

Figura 3-26. Intercambio detallado de tickets con servidor Kerberos.

Para observar el intercambio de tickets a nivel consola teclear desde el prompt de Windows lo que se muestra en la Figura 3-27:

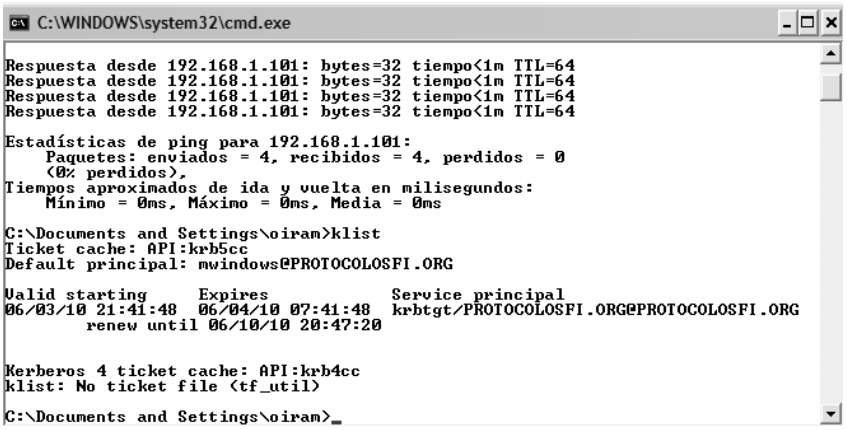

Figura 3-27. Intercambio de tickets desde el pompt de Windows

Cabe aclarar que si se ingresa de modo incorrecto la contraseña con la que se dio de alta el usuario en la base de datos del KDC, se obtiene la ventana de error que se muestra en la Figura 3-28:

Figura 3-28. Ventana de error de conexión con servidor Kerberos.

3.2 RADIUS

3.2.1 Implementación de RADIUS en Windows

Instalación y configuración de Microsoft Windows Server 2008

-Insertar DVD de instalación de Windows Server 2008, utilizando la versión Enterprise

-Proporcionar contraseña de la cuenta de administrador.

-Configurar fecha, hora y zona horaria. (GMT -06:00) Guadalajara, Ciudad de México, Monterrey. El horario es importante ya que el protocolo no funcionara si los relojes no se encuentran sincronizados.

-Configurar nombre de servidor. WIN-72NP5HN05CF

-Configurar red. Asignando una dirección IP v.4 fija. 192.168.1.88

-Instalar rol de Active Directory Domain Services.

En Administrador del Servidor, ubicar las funciones y agregar el "Servicio de dominio de Active Directory". Esto se muestra en la Figura 3-29.

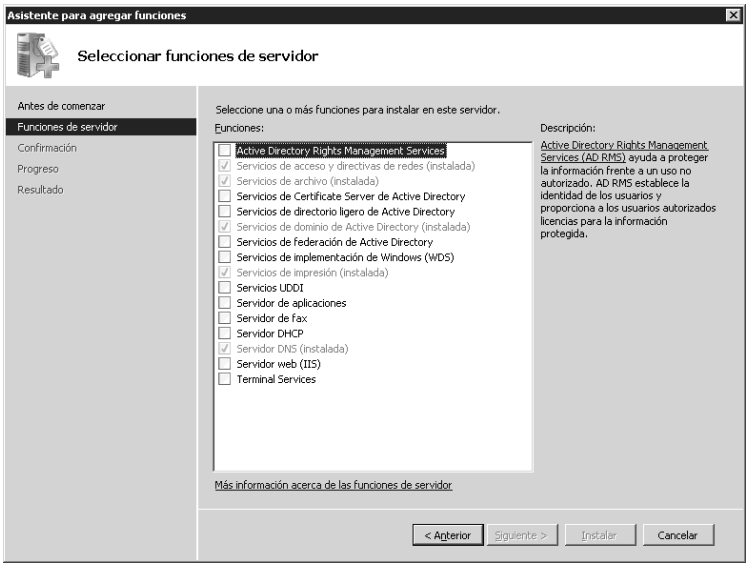

Figura 3-29. Administrador de funciones de Microsoft Windows Server 2008.

-Crear el controlador de dominio.

Para promocionar el rol antes instalado, es necesario, ejecutar el comando dcpromo.exe.

Continuar con las instrucciones y crear un nuevo dominio en un bosque nuevo.

Nombrar el bosque PROTOCOLOSFIR.ORG, posteriormente se solicita instalación opcional de un servidor DNS, en este caso, como no se tiene otro servidor que realice tal función, se instala, por default al ser éste el primer controlador de dominio de un bosque se instala el Catalogo Global.

-Indicar las ubicaciones de las bases de datos de los archivos log y de la página de Sysvol.

-Asignar la contraseña para la restauración de servicios del Directorio Activo.

-Finalizar y reiniciar el sistema.

-Crear usuarios de dominio y permisos.

Esta consola se muestra en la Figura 3-30.

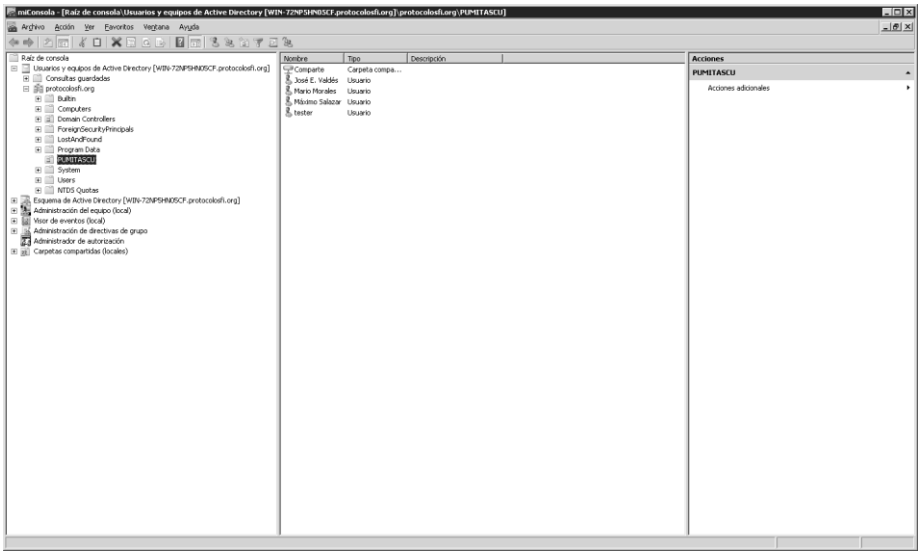

Figura 3-30. Consola de administración de servidor y usuarios.

-Instalar los Servicios de Certificate Server de Active Directory, para activar la Autoridad Certificadora para generar y firmar certificados para el dominio. Agregando la Autoridad Certificadora y la Autoridad Certificadora para el Registro Web.

-El tipo de Autoridad Certificadora será Enterprise y Root CA.

-Creamos una nueva llave privada y la configuración default sha1.

Lo mencionado anteriormente se muestra en la Figura 3-31.

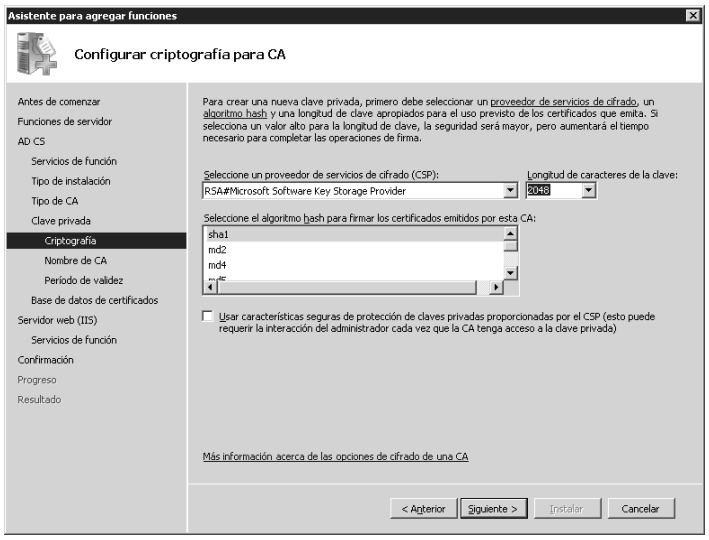

Figura 3-31. Selección de Criptografía para la Autoridad Certificadora.

-Instalar “Servicios de acceso y directivas de redes”, que se encuentra en las funciones del servidor. Seleccionar los servicios de “Servidor de directivas de redes (NPS)”, “Servicio de Enrutamiento y Acceso Remoto”, “Servicio de acceso remoto” y “Enrutamiento” e instalar.

-Ingresar a la consola de "Servidor de directivas de redes (NPS)", a través de comando nps.mmc. Seleccionar "Servidor RADIUS para conexiones cableadas o inalámbricas 802.1X". Esta opción se muestra en la Figura 3-32.

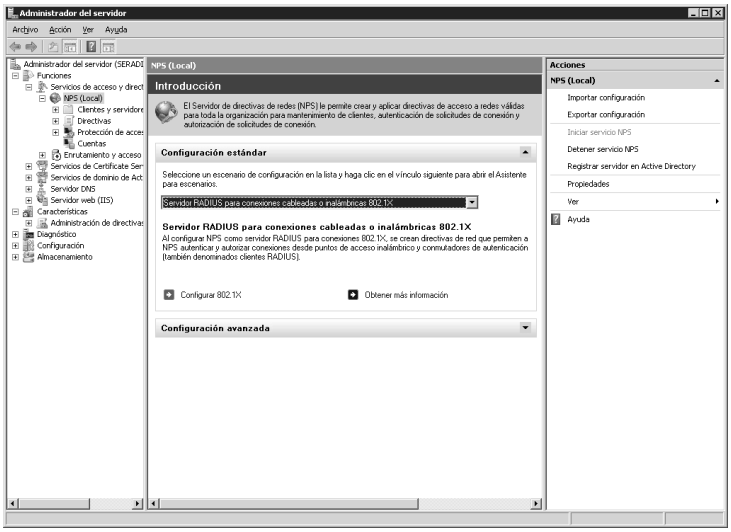

Figura 3-32. Consola NPS.

-Seleccionar "Configurar 802.1X" y elegir la conexión de tipo alámbrico.

-Al "Especificar Switches 802.1X" se agrega la configuración del cliente RADIUS. Utilizando como "Secreto compartido": INGENIERIA2010. Esta opción se muestre en la Figura 3-33.

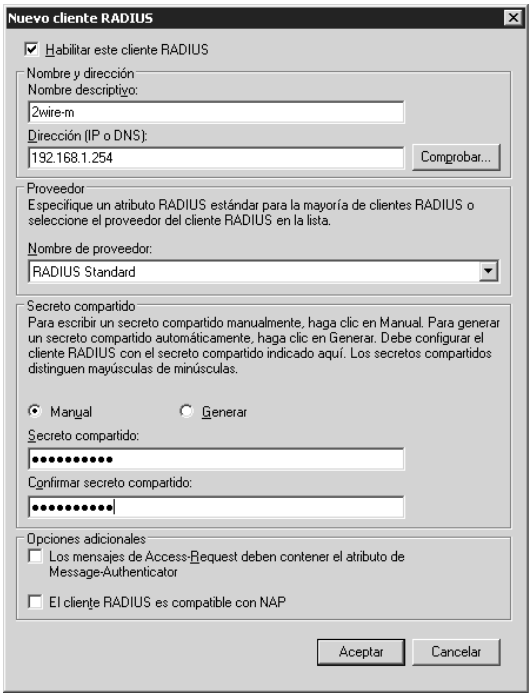

Figura 3-33. Configuración de nuevo cliente RADIUS.

-En la "Configuración del Método de Autenticación", seleccionar "Microsoft Protected EAP (PEAP)" y especificar los grupos de usuarios correspondientes, en este caso serán todos los usuarios del dominio PROTOCOLOSFIR.ORG.

Las pruebas de conexión con el directorio activo se encuentran a detalle en el Anexo 2.

3.2.2 Implementación de RADIUS en UNIX

Para la implementación del protocolo de autenticación RADIUS en el ambiente de UNIX, se instalará y configurará FreeRADIUS en un sistema Ubuntu Server 8.04 LTS.

3.2.2.1 Instalación y configuración de Ubuntu Server 8.04 LTS

La instalación y configuración inicial de Ubuntu Server 8.04 *LTS* se precisa con mayor detalle en el Anexo 3. Una vez instalado el sistema base, como primer paso, se debe cambiar o crear la contraseña del *root*, para lo cual se teclean las siguientes sentencias:

```
msalazar@radius1:~# sudo passwd root
[sudo] password for msalazar: ing110
Enter new UNIX password: rootsecret
Retype new UNIX password: rootsecret
passwd: password updated successfully
msalazar@radius1:~# su -
Password: rootsecret
root@radius1:/home/msalazar# _
```

Una vez probada la configuración de red y el acceso a Internet desde el servidor, lo siguiente a realizar es actualizar todos los paquetes instalados en el servidor (*update*), así como del propio sistema operativo (*upgrade*), esto se hace con los siguientes comandos:

```
root@radius1:~# apt-get update
root@radius1:~# apt-get upgrade
```

3.2.2.2 Instalación de FreeRADIUS

A continuación se listan todas las sentencias, paso por paso, empleadas para instalar FreeRADIUS una vez compilados los binarios y guardados en un CD:

```
root@radius1:~# apt-get install make
root@radius1:~# mount /cdrom
root@radius1:~# dpkg -i cdrom/utils/freeradius_2.0.4-0_i386.deb
root@radius1:~# apt-get -f install
root@radius1:~# dpkg -i cdrom/utils/freeradius-mysql_2.0.4-
0_i386.deb
root@radius1:~# dpkg -i cdrom/utils/freeradius-
dialupadmin_2.0.4-0_all.deb
root@radius1:~# apt-get -f install
```

3.2.2.3 Arranque de FreeRADIUS

Una vez instalado (aunque no configurado) FreeRADIUS, se puede arrancar el daemon (servicio) para ver si se ha instalado correctamente o tiene algún tipo de problema en el arranque. Para esto ejecutar los siguientes scripts para arrancar y parar el daemon *freeradius*:

```
root@radius1:~# /etc/init.d/freeradius start
root@radius1:~# /etc/init.d/freeradius stop
root@radius1:~# /etc/init.d/freeradius restart
```

Otra operación que se realiza muy frecuentemente en FreeRADIUS es arrancar *freeradius* en modo programa y no en modo daemon, además de solicitar que arranque en modo debug (depuración), para observar todas las fases del arranque para cada uno de los módulos de autenticación, autorización y auditoría. Para poder hacerlo, se descarga el daemon de la memoria mediante el script situado en */etc/init.d/* con el modificador *stop*, antes de poder ejecutarlo en modo programa. Los comandos para arrancar freeradius en modo debug trace son los siguientes:

```
root@radius1:~# /etc/init.d/freeradius stop
root@radius1:~# freeradius -X
```

3.2.2.4 Configuración básica de FreeRADIUS

Todos los archivos importantes de configuración de FreeRADIUS se encuentran, como es habitual, en la estructura de directorios de Linux, en el directorio */etc/*, concretamente en */etc/freeradius*.

Mediante el comando *ls* se puede ver cuáles son los archivos de configuración de FreeRADIUS, esto se observa en la Figura 3-34:

```
root@radius1:~# cd /etc/freeradius
root@radius1:/etc/freeradius# ls
acct_users                  dictionary           policy.conf          snmp.conf
attrs                       eap.conf             policy.txt           sql
attrs.access_reject         experimental.conf    preproxy_users       sql.conf
attrs.accounting_response   hints                proxy.conf           sqlippool.conf
attrs.pre-proxy             huntgroups           radiusd.conf         tamplates.conf
certs                       ldap.attrmap         sites-available      users
clients.conf                otp.conf             sites-enabled
```

Figura 3-34. Archivos de configuración de FreeRADIUS.

Los archivos para realizar las pruebas de autenticación contra el servidor, son los archivos *users* y *clients.conf*, los demás se dejaran con su configuración por default. Estos dos archivos serán configurados con la ayuda del editor *nano*.

En el archivo *users* se crean todos los usuarios deseados, con sus atributos relacionados. En este caso, se creará el usuario con el que serán realizadas las pruebas. Este archivo también permite configurar los parámetros o atributos por defecto para la autenticación mediante los campos *DEFAULT*. La configuración de este archivo se observa en la Figura 3-35:

```
mmorales Cleartext-Password := "saeta"
          Service-Type = Framed-User,
          Framed-Protocol = PPP,
          Framed-IP-Address = 192.168.1.101
          Framed-IP-Netmask = 255.255.255.0
          Framed-Routing = Broadcast-Listen,
          Framed-Filter-Id = "std.ppp",
          Framed-MTU = 1500,
          Framed-Compression = Van-Jacobsen-TCP-IP

DEFAULT Auth-Type := Local
         Fall-Through = Yes

DEFAULT Framed-Protocol == PPP,
         Framed-Protocol = PPP,
         Framed-Compression = Van-Jacobsen-TCP-IP

DEFAULT Hint == "CSLIP"
         Framed-Protocol = SLIP,
         Framed-Compression = Van-Jacobsen-TCP-IP

DEFAULT Hint == "SLIP"
         Framed-Protocol = SLIP
```

Figura 3-35. Archivo de configuración *users*.

Ahora, como siguiente paso se agrega el cliente con el que se realizarán las pruebas de autenticación sobre el sistema en el archivo *clients.conf*. El archivo de configuración de clientes o NAS *clients.conf* se observa en la Figura 3-36.

```
client localhost {
        secret = testing123
        shortname = localhost
        nastype = other
}

client 192.168.1.101 {
        secret = saeta
        shortname = mmorales
       nastype = other
}
```

Figura 3-36. Archivo de configuración *clients.conf*.

Con la configuración de estos dos archivos el sistema está listo para realizar pruebas de autenticación contra el servidor.[13]

3.2.2.5 Test de funcionamiento

Detener el servicio *freeradius* para posteriormente ser lanzado en modo programa con la opción de debug trace (depuración completa).

En la consola actual ejecutar lo mostrado en la Figura 3-37 y observar como *freeradius* se queda esperando solicitudes para procesarlas:

```
root@radius1:~# /etc/init.d/freeradius stop
root@radius1:~# freeradius -X
[...]
Listening on authentication address 127.0.0.1 port 1812
Listening on accounting address * port 1813
Ready to process requests.
```

Figura 3-37. FreeRADIUS a la espera de solicitudes.

Ahora abrir otra consola o terminal y realizar una prueba de autenticación con el servidor mediante el comando *radtest*. Esto se muestra en la Figura 3-38.

```
root@radius1:~# radtest mmorales saeta localhost 0 testing123
Sending-Access-Request of id 152 to 127.0.0.1 port 1812
User-Name = "mmorales"
User-Password = "saeta"
NAS-IP-Address = 192.168.1.100
NAS-Port = 0
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=152, length=20
```

Figura 3-38. Prueba de autenticación contra el sistema operativo.

Se ha recibido un paquete de tipo Access-Accept, con esto se demuestra que FreeRADIUS ha sido capaz de leer el fichero de usuarios del servidor. Una prueba de error de una autenticación rechazada se muestra en la Figura 3-39.

```
root@radius1:~# radtest mmorales saeto localhost 0 testing123
Sending-Access-Request of id 19 to 127.0.0.1 port 1812
User-Name = "mmorales"
User-Password = "saeto"
NAS-IP-Address = 192.168.1.100
NAS-Port = 0
rad_recv: Access-Reject packet from host 127.0.0.1 port 1812, id=19, length=20
```

Figura 3-39. Ejemplo de autenticación rechazada por error en el password.

En la consola previamente abierta, en la que se encuentra corriendo *freeradius* en modo debug trace, se puede apreciar la salida de la depuración. Esto se muestra en la Figura 3-40.

```
[...]
++[pap] return reject
auth: Failed to validate the user.
Login incorrect (rlm_pap: CRYPT password check failed): [mmorales/saeto] (from
client localhost port 0)
Found Post-Auth-Type Reject
```

Figura 3-40. Resultado en el modo debug tras una autenticación errónea.

3.3 LDAP

3.3.1 Implementación de LDAP en Windows

Instalación y configuración de Microsoft Windows Server 2008

-Insertar DVD de instalación de Windows Server 2008, utilizando la versión Enterprise

-Proporcionar contraseña de la cuenta de administrador.

-Configurar fecha, hora y zona horaria. (GMT -06:00) Guadalajara, Ciudad de México, Monterrey. El horario es importante ya que el protocolo no funcionara si los relojes no se encuentran sincronizados.

-Configurar nombre de servidor. WIN-72NP5HN05CF

-Configurar red. Asignando una dirección IP v.4 fija. 192.168.1.88

-Instalar rol de Active Directory Domain Services.

En Administrador del Servidor, ubicar las funciones y agregar el "Servicio de dominio de Active Directory". Esto se muestra en la Figura 3-41.

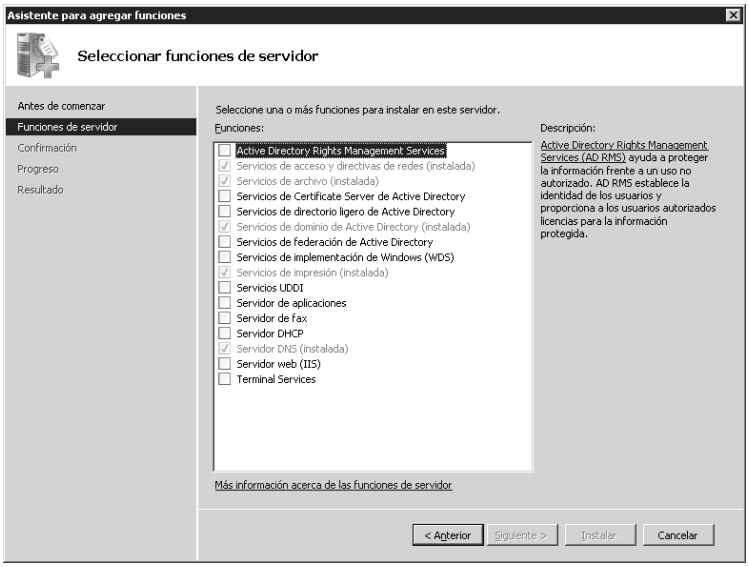

Figura 3-41. Administrador de funciones de Microsoft Windows Server 2008.

-Crear el controlador de dominio.

Para promocionar el rol antes instalado, es necesario, ejecutar el comando dcpromo.exe.

Continuar con las instrucciones y crear un nuevo dominio en un bosque nuevo.

Nombrar el bosque PROTOCOLOSFIL.ORG, posteriormente se solicita instalación opcional de un servidor DNS, en este caso, como no se tiene otro servidor que realice tal función, se instala, por default al ser éste el primer controlador de dominio de un bosque se instala el Catalogo Global.

-Indicar las ubicaciones de las bases de datos de los archivos log y de la página de Sysvol.

-Asignar la contraseña para la restauración de servicios del Directorio Activo.

-Finalizar y reiniciar el sistema.

-Crear usuarios de dominio y permisos.

La consola de administración se muestra en la Figura 3-42.

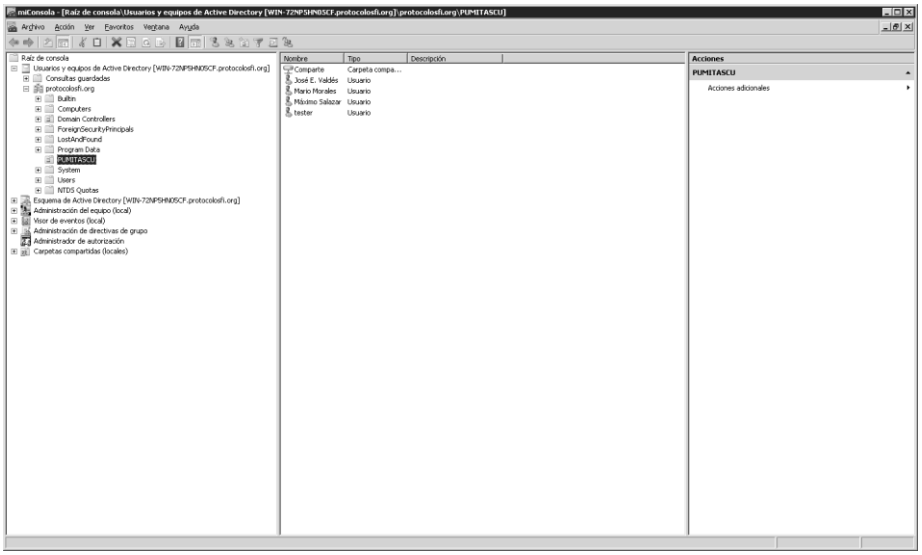

Figura 3-42. Consola de administración de servidor y usuarios.

Las pruebas de conexión con el directorio activo se encuentran a detalle en el Anexo 2.

3.3.2 Implementación de LDAP en UNIX

Para la instalación, configuración y pruebas del protocolo LDAP en el ambiente Linux se han elegido Fedora como sistema operativo y Fedora Directory Server como servidor LDAP.

3.3.2.1 Instalación de Fedora

A continuación se describen de manera breve los pasos necesarios para el proceso de instalación y primer arranque con el sistema operativo Fedora:

1. Descargar la imagen en formato ISO del DVD arrancable, desde la página oficial del proyecto Fedora (http://fedoraproject.org/es/get-fedora-options#formats).
2. Grabar el archivo imagen en un DVD e iniciar la computadora desde el DVD recién creado.
3. Se mostrará una pantalla de bienvenida de Fedora, seleccionar el idioma español y elegir la opción de instalar Fedora.
4. Elegir la configuración adecuada para nuestro teclado.

5. Sólo en el caso de que aplique esta opción, seleccionar y asignar los dispositivos de almacenamiento que estarán disponibles para el sistema Fedora.
6. Inicializar el disco duro donde se instalará Fedora.
7. Configurar el nombre del equipo, para este caso el nombre queda como *ldap.localhost.localdomain*.
8. Las siguientes pantallas son para la configuración del huso horario y para escribir la contraseña del superusuario o root.
9. A continuación siguen las pantallas para configurar la partición o particiones del disco duro donde se instalará Fedora, luego de dar en aceptar se escribirán los cambios al disco.
10. Enseguida se procede a la instalación del sistema base, este proceso puede durar entre 5 y 10 minutos aproximadamente.
11. Seleccionar los paquetes a instalar junto con Fedora, para este caso instalar el Servidor Web Apache y el paquete Java JRE, que son los que se necesitan para el correcto funcionamiento de FDS.
12. Finalmente, reiniciar la computadora la cual ha quedado lista para el primer arranque de Fedora.

3.3.2.2 Instalación y Configuración de Fedora Directory Server

Fedora Directory Server es un servidor LDAP (Lightweight Directory Access Protocol) para Linux desarrollado por Red Hat y la comunidad de Fedora, permite un completo sistema de identidades y una plataforma integral para múltiples servicios de servidor. Enfocado principalmente a instituciones y empresas corporativas, cuenta con múltiples características que lo hace el favorito para implementaciones del mundo real.

Se destaca su capacidad de replicación Multimaster (MMR), compatibilidad con Microsoft Active Directory, Soporte SNMP, Integridad Referencial, Grupos estáticos y dinámicos, Roles, Clases de Servicios, Vistas, Editor Gráfico de Esquema y todo un conjunto de herramientas para un mejor control operacional. En la actualidad está trabajando en una amplia variedad de empresas e instituciones a nivel mundial, principalmente por su alto rendimiento y fácil administración.

La suite Fedora Directory Server consta principalmente de 4 subsistemas:

- Fedora Directory Core
- Fedora Directory Administration
- Fedora Directoty Console
- Fedora Org Chart

El *JRE* es requerido para poder usar la consola de Administración de FDS, una forma de saber si está instalado en el sistema es ejecutando en la consola el comando `java -version` el cuál se encarga de mostrar la versión de java, si al ejecutarlo la salida contiene algo como gcj o GCJ (Gnome Compiler Java) hay que actualizar el JRE pues la instalación del FDS requiere las versiones openjdk o icedtea que se pueden instalar vía yum usando el comando `yum install java-1.7.0-icedtea` o `yum install java-1.6.0-openjdk`, o vía rpm descargando el paquete desde el sitio oficial http://icedtea.classpath.org/download/fedora/.

El Servidor HTTP Apache modelo worker, se encuentra instalado como un demonio del sistema. Para ver el estado del servicio ejecutar desde la consola `/etc/init.d/httpd status`, para correrlo ejecutar `/etc/init.d/httpd start` y para pararlo `/etc/init.d/httpd stop`.

Descargar el archivo fedora-ds-1.0.4-1.PLATFORM.ARCH.opt.rpm desde http://directory.fedoraproject.org/wiki/Download, donde PLATFORM es reemplazado por RHEL3, RHEL4, FC4, FC5, o FC6 y ARCH es i386 o x86_64; es decir, PLATFORM hace referencia a la plataforma sobre la que se va a instalar el FDS y ARCH hace referencia a la arquitectura del procesador. Para este caso de implementación descargar el archivo fedora-ds-1.0.4-1.FC6.i386.opt.rpm.

Para instalar desde la línea de comandos ejecutar:

```
rpm -Uvh fedora-ds-1.0.4-1.FC6.i386.opt.rpm
```

Si no hay problemas en la instalación, la consola nos mostrará:

```
Install finished. Please run /opt/fedora-ds/setup/setup to
complete installation and set up the servers.
```

Para ejecutar el asistente de instalación FDS, ir al directorio de instalación escribiendo en la consola:

```
cd /opt/fedora-ds
```

Antes de continuar, se debe asegurar que el hostname esté apropiadamente registrado en el servidor DNS o en el archivo /etc/hosts, para ello ejecutar el comando `ping ldap.localhost.localdomain` en una consola; si al ejecutar retorna 127.0.0.1 o unknown host esto significa que el hostname no esta registrado correctamente.

Después de haber comprobado el hostname, crear un usuario y un grupo llamados fds para correr el servicio, esto se logra con la ayuda del administrador gráfico para grupos y usuarios que incluye Fedora.

Ahora escribir en el prompt `./setup/setup` para instalar Fedora Directory Server, esto mostrará un asistente que desplegará una serie de preguntas para ir configurando el servicio, como sigue:

1. **LICENSE AGREEMENT AND LIMITED PRODUCT WARRANTY**
   FEDORA(TM) DIRECTORY SERVER
   This agreement governs the use of Fedora Directory...

En este apartado el asistente pide leer la licencia del FDS, al final de la misma se deberá indicar si está de acuerdo y querer continuar.

2. Your system has been scanned for potential problems, missing patches, etc. The following output is a report of the items found that need to be addressed before running this software in a production environment...

El asistente muestra advertencias de problemas que tiene el sistema para que sean resueltas antes de iniciar la instalación, en caso de que puedan ser pasadas por alto se debe escribir *yes* para continuar.

3. Choose a setup type:
   1. **Express**: Allows you to quickly set up the servers using the most common options and pre-defined defaults. Useful for quick evaluation of the products.
   2. **Typical**: Allows you to specify common defaults and options.
   3. **Custom**: Allows you to specify more advanced options. This is recommended for experienced server administrators only.

Se tienen 3 modos de instalación: 1-Express: El más útil para evaluar el producto. 2-Typical: Permite especificar los parámetros comunes y las opciones principales. 3-Custom: Permite especificar opciones más avanzadas.

Por defecto, seleccionar 2 y continuar.

4.
```
hostname to use (default: localhost.localdomain)...
Computer name [ldap.localhost.localdomain]:
System User [nobody]: fds
System Group [nobody]: fds
```

Dejar el hostname que se había configurado con anterioridad, y especificar el usuario y el grupo que se creará para la administración del servicio.

En caso de que aparezca el error:

```
./ns-config: error while loading shared libraries:
libtermcap.so.2: cannot open shared object file: No such file or
directory
ERROR Exiting . . .
Log file is /tmp/lognIfjhl
```

Resolverlo de la siguiente manera:

- ✓ Descargar los paquetes **libtermcap-2.0.8-47.i386** y **termcap-5.5-1.20060701.1.noarch.rpm**
- ✓ Instalarlos con el comando:

  ```
  rpm -Uvh termcap-5.5-1.20060701.1.noarch.rpm libtermcap-
  2.0.8-47.i386
  ```
- ✓ Después de instalar correctamente, la consola arrojará lo siguiente:

  ```
  Preparando...    ##################################    [100%]
  1:termcap    ##################################    [   50%]
  2:libtermcap    ##################################    [100%]
  ```

Ahora hay que volver al paso 1 de la instalación.

5.
```
Server information is stored in the configuration directory
server...
Do you want to register this software with an existing
configuration directory server? [no]: ↵
```

Como se trata de la primera instalación del Servicio de Directorio, escribir *no* para continuar.

```
6. If you already have a directory server you want to use to
   store your data...
   Do you want to use another directory to store your data? [No]
```

Al igual que en la pregunta anterior, escribir *no* para continuar, pues no se tiene otro Servicio de Directorio instalado.

```
7. The standard directory server network port number is 389...
   Directory server network port [389]: ↲
```

Dejar el puerto por defecto.

```
8. Each instance of a directory server requires a unique
   identifier...
   Directory server identifier:[ldap] ↲
```

El asistente reconoció sin problemas el Identificador del equipo, dar *enter* para continuar.

```
9. Please enter the administrator ID for the Fedora
   configuration...
   administrator ID [admin]:
   Password:
   Password (again):
```

El usuario por defecto *admin* será quien administre desde la consola del FDS, ingresar su password y continuar.

```
10. The suffix is the root of your directory tree. You may have
    more than one suffix.
    Suffix [dc=localhost, dc=localdomain]: ↲
```

El sufijo es usado para almacenar los datos del administrador, esta es la parte del FQDN ldap.localhost.localdomain en la forma dc=localhost, dc=localdomain; siendo dc el acrónimo de Domain Controller.

```
11. Certain directory server operations require an administrative
    user. This user is referred...
    Directory Manager DN [cn=Directory Manager]:
    Password:
    Password (again):
```

El usuario Directory Manager será usado para ciertas operaciones de administración, siendo su uso muy similar al usuario root bajo entornos Unix. Presionar enter e ingresar el password 2 veces, teniendo en cuenta que no debe ser inferior a 8 caracteres.

```
12. The information stored in the configuration directory server
    can be separated...
    Administration Domain [localhost.localdomain]: ↵
```

La información de configuración puede ser almacenada en diferentes Dominios Administrativos, en cuyo caso se ingresan los identificadores de los mismos, esto puede ser útil para algunas empresas con sedes diferentes.

Para este caso se está instalando el FDS para un sólo dominio de administración, por lo tanto presionar *enter* para continuar.

```
13. The Administration Server is separate from any of your web or
    application servers since it listens to a different port...
    Administration port [9830]: 26492
```

Cambiar el puerto de administración HTTP por defecto, el cual es usado por FDS para escuchar el servicio de Administración que monta sobre Apache.

```
14. The Administration Server program runs as a certain user on
    your system...
    Run Administration Server as [root]: ↵
```

Presionar *enter* para continuar, debido a que se usará el usuario root para escribir los archivos de configuración y arrancar el servicio.

```
15. The Administration Server runs on the Apache web server.
    Please provide the directory...
    Apache Directory [/usr/sbin/]:
```

Pulsar enter ya que el binario del demonio de Apache, httpd, se encuentra en la ubicación especificada.

Con estos pasos, la instalación ha sido finalizada satisfactoriamente. Para ejecutar la consola de Administración ejecutar lo siguiente:

```
# cd /opt/fedora-ds
# ./startconsole -u admin -a
http://ldap.localhost.localdomain:26492/
```

Si no hay problemas se obtendrá la ventana de login. Ingresar el password y dar click en OK, con esto se accede a la consola de administración, tal como se observa en la Figura 3-43.

Figura 3-43. Consola de Administración de Fedora Directory Server.

3.3.2.3 Pruebas de autenticación con Fedora Directory Server

Antes de proceder a las pruebas de autenticación de usuarios mediante FDS, es necesario iniciar el servidor Apache, así como los servicios Fedora Directory Core y Fedora Directory Administrator. Los comandos para iniciar estos servicios son los siguientes:

```
# /etc/init.d/httpd start
# /opt/fedora-ds/slapd-ldap/start-slapd
# /opt/fedora-ds/start-admin
```

Una vez inicializados estos servicios, lo siguiente a realizar es agregar un usuario para probar la autenticación mediante FDS. Desde la consola de administración se crea un nuevo registro con los siguientes datos:

Nombre: Mario Morales

Usuario: MMorales

Password: saeta

```
uid=MMorales,ou=People,dc=localhost,dc=localdomain
```

Las pruebas de autenticación con FDS fueron las mismas tanto para ambientes Windows como Linux y se realizaron desde sus respectivos exploradores de Internet, ingresando por HTTP a la URL del servidor LDAP a través del puerto 26492 configurado durante la instalación (http://ldap.localhost.localdomain:26492/).

En las Figuras 3-44 a 3-47 se pueden observar los resultados obtenidos en dichas pruebas.

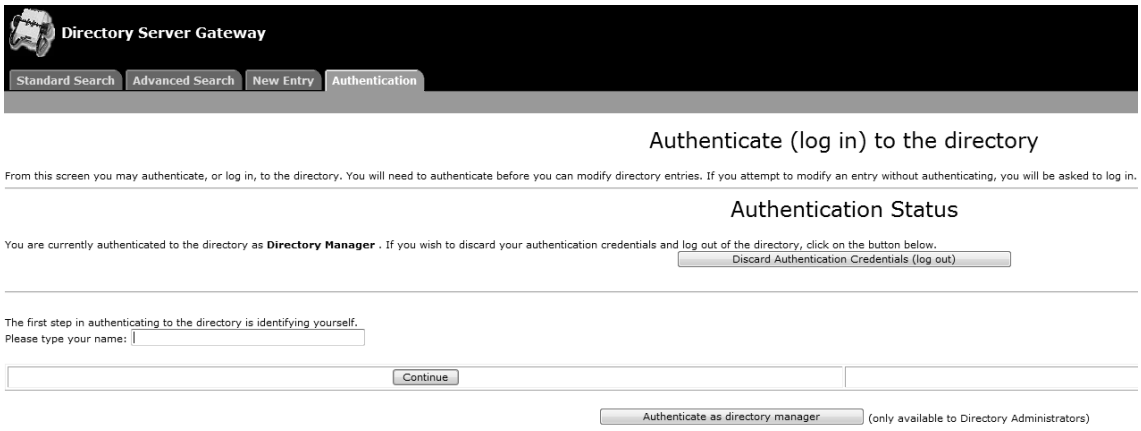

Figura 3-44. Pantalla de login para autenticarse e ingresar al directorio.

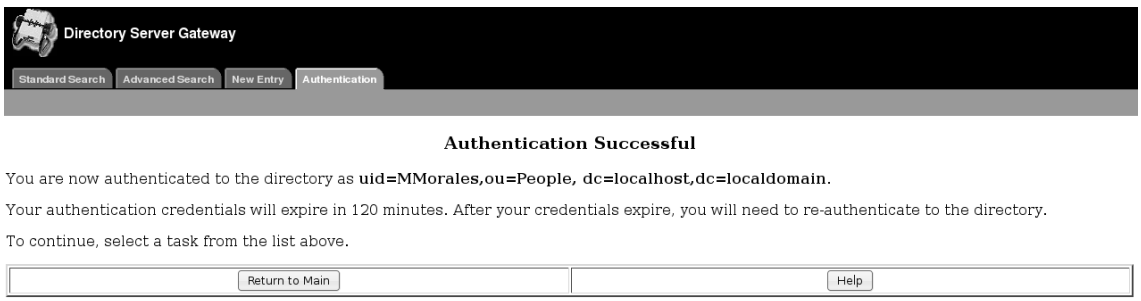

Figura 3-45. Pantalla de autenticación exitosa y obtención de credenciales.

Figura 3-46. Pantalla de autenticación errónea por introducir un password incorrecto.

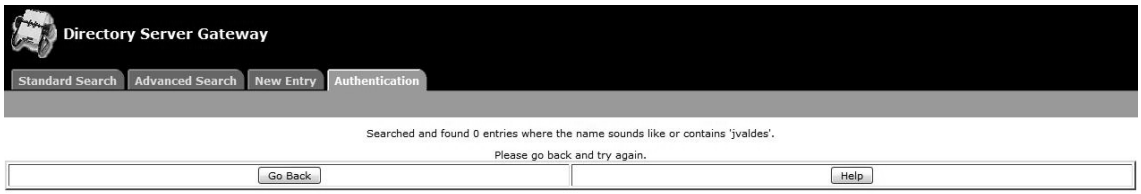

Figura 3-47. Pantalla de usuario no encontrado en el directorio.

Adicionalmente, en las siguientes figuras se observan los resultados obtenidos tras ejecutar el comando *ldapsearch* desde ambientes Unix o Linux, para los casos en que el servidor de autenticación FDS se encontraba apagado (Figura 3-48), y los resultados arrojados una vez que el servidor y los servicios de FDS ya habían sido inicializados (Figura 3-49):

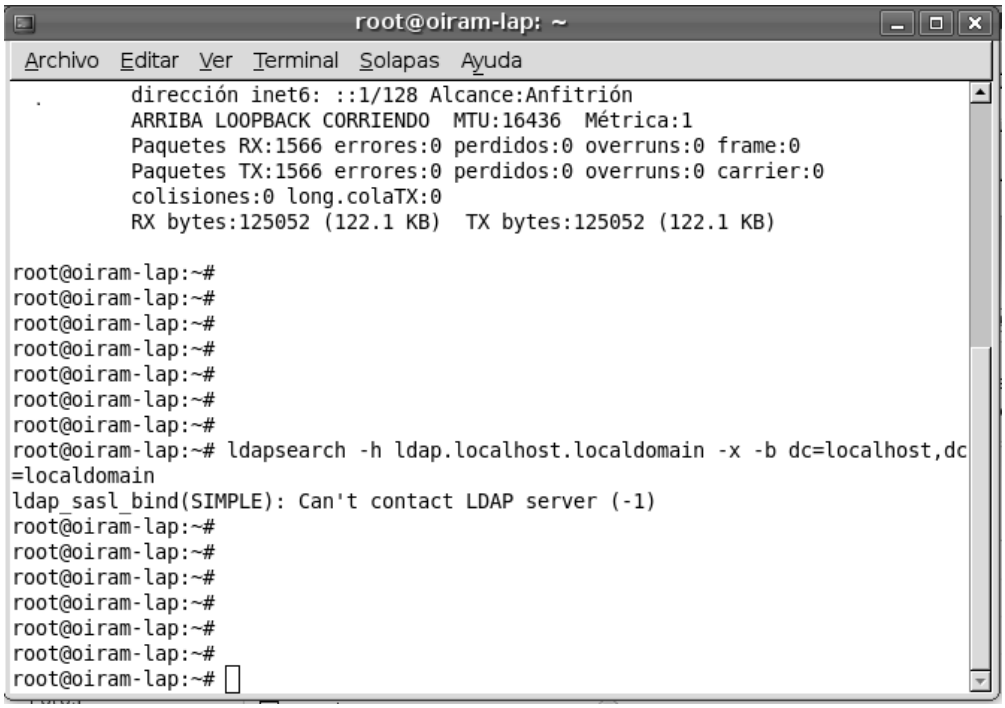

Figura 3-48. Servidor LDAP no encontrado mediante *ldapsearch*.

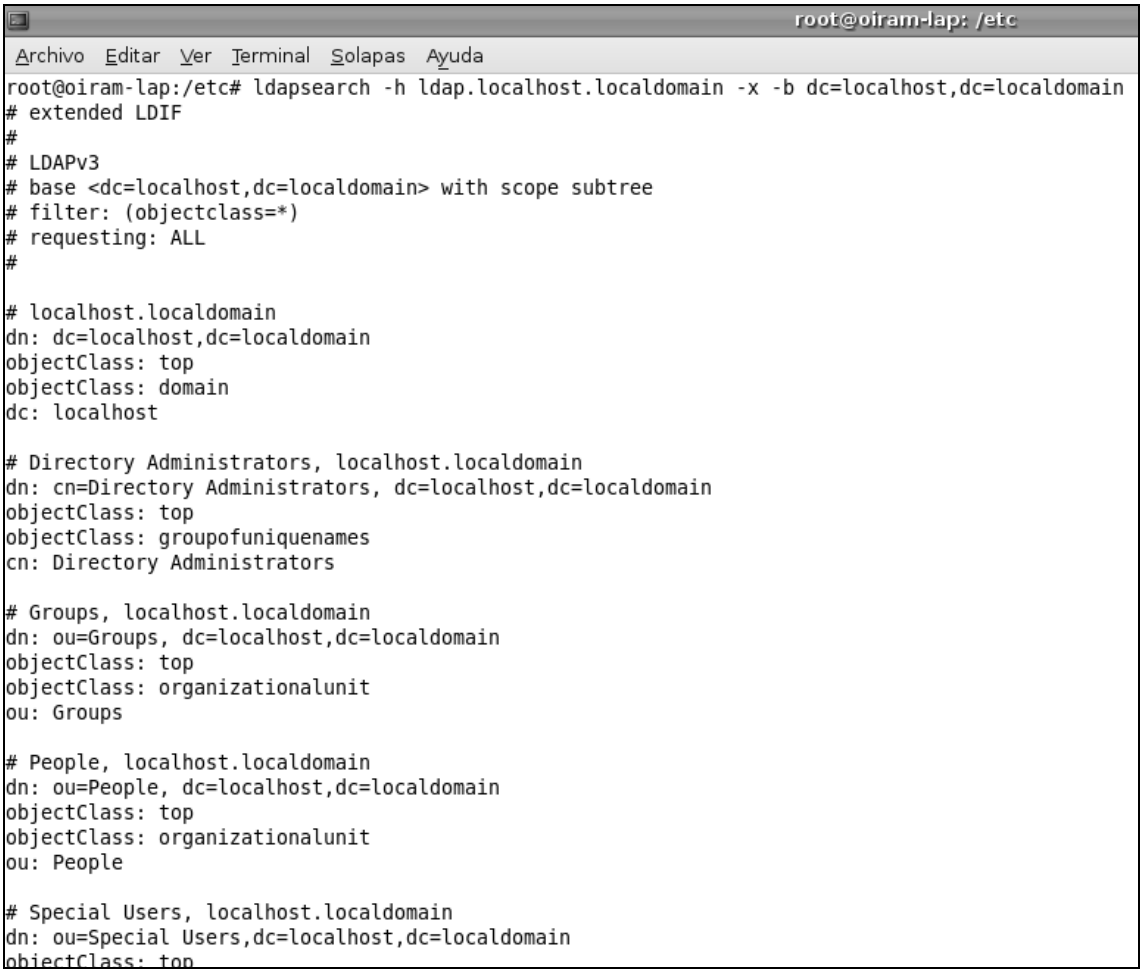

Figura 3-49. Servidor LDAP encontrado mediante *ldapsearch* y obtención de información del directorio.

CAPÍTULO 4
ANÁLISIS DE IMPLEMENTACIONES

En el anterior capítulo se realizaron implementaciones en una red de datos para los protocolos de autenticación Kerberos, Radius y LDAP bajo las plataformas Windows y Unix. Por tal motivo, en base a las implementaciones realizadas a continuación se realizará un análisis más detallado de acuerdo a las siguientes características:

- Administración que se tiene con en el servidor implementado.
- Optimizar recursos del servidor de acuerdo a las necesidades del sistema operativo y del servicio de autenticación instalado.
- Eficiencia del protocolo de seguridad que emplea cada servidor de autenticación implementado.

4.1 ADMINISTRACIÓN DE SERVIDOR

Analizando las distintas implementaciones realizadas de los protocolos de autenticación Kerberos, Radius y LDAP bajo las plataformas Windows y Unix, se puede observar sus características de acuerdo a la complejidad de instalación del sistema operativo, implementación y configuración del servicio de autenticación, facilidad de administración para el usuario y la cantidad de documentación disponible para ayudar en la administración del servidor.

4.1.1 Kerberos, Radius y LDAP para Windows

La implementación de los protocolos de autenticación Kerberos, Radius y LDAP bajo el sistema operativo Windows es similar para los tres casos, ya que la instalación del sistema operativo Windows Server 2008 es la misma para los tres protocolos, donde dicha instalación del sistema operativo resulta intuitiva y fácil de realizar.

Por otro lado, la implementación y configuración del servicio de autenticación que se desea implementar radica básicamente en saber qué servicio de seguridad se desea habilitar en la consola del Administrador de funciones de Microsoft Windows Server 2008. Para comprender esto existe una gran cantidad de manuales en la página de internet de Windows.

Haciendo referencia a la facilidad de administración para el usuario, se considera como una interfaz muy amigable, intuitiva y de gran alcance administrativo para generar usuarios y asignar perfiles por medio de la Consola de Administración de servidor y usuarios Active Directory. Además se cuenta con comandos a nivel consola para realizar cargas masivas de usuarios.

4.1.2 Kerberos, Radius y LDAP para UNIX

La implementación de los protocolos de autenticación Kerberos, Radius y LDAP en UNIX resulta muy distinta al modo del cómo se realizan las implementaciones en Windows. Por tal motivo el análisis de estos tres servicios en UNIX resulta detallado y sin poder generalizar características.

4.1.2.1 Kerberos para UNIX

La implementación de Kerberos en Unix no resulta muy compleja haciendo referencia al sistema operativo dónde se instala el KDC, ya que la instalación del sistema operativo Ubuntu 8.04 es sencilla, sólo se necesita tener conocimientos sobre el tipo de formateo Ext2 que se realiza a la partición del disco donde se instala el sistema operativo y la ubicación del /(root) como punto de montaje.

La parte compleja de esta instalación radica en la implementación y configuración del servicio de seguridad Kerberos, ya que para realizar esto es necesario tener conocimientos en el dominio del ambiente Ubuntu para configurar la tarjeta de red, conocimiento en el manejo de comandos UNIX para el uso adecuado de la consola y para el manejo del editor de texto *vi*. Además los manuales de usuario de la página del MIT para realizar la instalación de Kerberos son insuficientes ya que no contempla muchos puntos para una correcta configuración.

La parte administrativa del servidor es insuficiente para que el usuario pueda realizar una fácil administración, ya que no se cuenta con una interfaz gráfica para realizar esta operación, por lo que la carga de usuarios y asignación de perfiles sólo se puede realizar a nivel consola, aunque se tiene la opción de realizar carga masiva de usuarios por medio de comandos a nivel consola. Es posible instalar Apache y php-myAdmin para tener un ambiente gráfico de administración del sistema, pero es muy complicado y tardado optimizar este recurso al servidor, además que no hay

información al respecto en las página principal del MIT y en Internet tampoco se cuenta con mucha información para realizar esta configuración.

4.1.2.2 Radius para UNIX

La implementación de Radius en Unix es compleja en la instalación del sistema operativo donde trabaja el servicio de autenticación, ya que se tiene una gran desventaja en la instalación y ambiente del sistema operativo, porque Ubuntu Server 8.04 LTS trabaja totalmente a nivel consola.

Se tiene la ventaja que se cuenta con gran cantidad de información en Internet y libros especializados para realizar la implementación, por tal motivo no resulta complicado realizar su respectiva configuración puesto que se cuenta con bastante información para lograr este fin. A pesar de ello el administrador debe tener dominio en el ambiente Ubuntu Server para configurar la tarjeta de red, conocimiento en el manejo de comandos UNIX para el uso adecuado de la consola y para el manejo del editor de textos *nano*. También se puede configurar sin problemas Apache en el servidor para tener un ambiente gráfico para la administración del servidor y con ello poder conectarse a él de manera remota.

La administración del servidor también resulta fácil de realizar, ya que por la basta cantidad de información que se cuenta para realizar este fin, cualquier duda al respecto puede ser aclarada sin mayor complicación por los manuales de usuario que existen en Internet, además el ambiente gráfico proporcionado por Apache facilita la creación de usuarios y asignación de perfiles, además que se cuenta con la opción de realizar carga masiva de usuarios por medio de comandos a nivel consola.

4.1.2.3 LDAP para UNIX

La implementación de LDAP en UNIX es sencilla de realizar para instalar el sistema operativo Fedora y para configurar el servicio de autenticación. Para instalar el sistema operativo Fedora no son necesarios conocimientos de experto, ya que se cuenta con un ambiente gráfico para realizar la instalación y además se pueden seleccionar los paquetes para complementar la instalación y de gran ayuda para la configuración posterior del servicio de autenticación como el Servidor Web Apache y el paquete Java JRE.

La instalación y configuración del servicio de autenticación LDAP no es complicada, puesto que no es necesario editar archivos de configuración y con la instalación de los paquetes de Directory Server se completa gran cantidad de configuración con un test en la instalación. En la página principal de Fedora se cuenta con un manual detallado para instalar y configurar lo necesarios para una correcta implementación.

La administración del servidor es fácil de realizar, ya que la Consola de Administración de Fedora Directory Server tiene interfaz gráfica y es de gran ayuda para crear usuarios y asignar perfiles, además se cuenta con la opción de realizar carga masiva de usuarios por medio de comandos a nivel consola.

4.2 OPTIMIZACIÓN DE RECURSOS DEL SERVIDOR

Dentro de la instalación y adecuación de una red de datos, es determinante, que se cuente con el hardware suficiente o mayor para que la capacidad de la misma no se vea sobrepasada, considerando las necesidades de aplicación y la demanda a la cual será sujeto el servicio, con lo cual se tendrá una eficiencia considerablemente mayor y los problemas de aplicación serán menores.

4.2.1 KERBEROS

En las Tablas 4-1 y 4-2 se muestran los requisitos de hardware para servidores en Kerberos.

Windows Server 2008		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	1.4 GHz (x64 processor) or 1.3GHz (Dual Core)	AMD Athlon XP 3000+ 2.10 GHz
Memoria	512 MB RAM	1.25 GB RAM
Espacio en Disco Duro Requerido	32 GB o mayor	80 GB
Pantalla	Super VGA (800 × 600) o alta	VGA
Otros	DVD drive, Teclado, Mouse, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-1. Requisitos servidor Kerberos en Windows

Ubuntu 8.04 Desktop Edition		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	1 GHz x86 Procesador	Pentium 4 1.60 GHz
Memoria	512 MB RAM	512 MB RAM
Espacio en Disco Duro Requerido	5 GB o mayor	30 GB
Pantalla	Tarjeta gráfica y monitor que soporte 1024x768	NVIDIA
Otros	CD/DVD drive, Teclado, Mouse, Audio, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-2. Requisitos servidor Kerberos en Unix

4.2.2 RADIUS

En las Tablas 4-3 y 4-4 se muestran los requisitos de hardware para servidores en Radius.

Windows Server 2008		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	1.4 GHz (x64 processor) or 1.3GHz (Dual Core)	AMD Athlon XP 3000+ 2.10 GHz
Memoria	512 MB RAM	1.25 GB RAM
Espacio en Disco Duro Requerido	32 GB o mayor	80 GB
Pantalla	Super VGA (800 × 600) o alta	VGA
Otros	DVD drive, Teclado, Mouse, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-3. Requisitos servidor Radius en Windows

Ubuntu Server 8.04 LTS		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	Intel x86 and AMD64 Procesador	Intel Core 2 T7200 2.00 GHz
Memoria	128 MB RAM	3 GB RAM
Espacio en Disco Duro Requerido	1 GB o mayor	100 GB
Pantalla	Super VGA (800 × 600) o alta	NVIDIA
Otros	CD/DVD drive, Teclado, Mouse, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-4. Requisitos servidor Radius en Unix

4.2.3 LDAP

En las Tablas 4-5 y 4-6 se muestran los requisitos de hardware para servidores LDAP.

Windows Server 2008		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	1.4 GHz (x64 processor) or 1.3GHz (Dual Core)	AMD Athlon XP 3000+ 2.10 GHz
Memoria	512 MB RAM	1.25 GB RAM
Espacio en Disco Duro Requerido	32 GB o mayor	80 GB
Pantalla	Super VGA (800 × 600) o alta	VGA
Otros	DVD drive, Teclado, Mouse, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-5. Requisitos servidor LDAP en Windows

Fedora 9 Core		
Componente	***Requisitos (Mínimos)***	***Implementación***
Procesador	X86 o x86_64 Procesador	Intel Core 2 T7200 2.00 GHz
Memoria	256 MB RAM	3 GB RAM
Espacio en Disco Duro Requerido	4 GB o mayor	100 GB
Pantalla	Super VGA (800 × 600) o alta	NVIDIA
Otros	CD/DVD drive, Teclado, Mouse, Internet.	CD/DVD drive, Teclado, Mouse, Internet.

Tabla 4-6. Requisitos servidor LDAP en Unix

Windows server en modo gráfico supone un uso de mayor recursos que cualquiera de los otros sistemas operativos, aunque tiene la posibilidad de instalación en modo consola, que supone menos recursos pero la pérdida de su principal características que es el ambiente amigable.

Ubuntu Server tiene requisitos mínimos debido a que es totalmente modo de consola.

Ubuntu Desktop y Fedora 9 son muy similares en su ambiente gráfico y requisitos por lo cual sus características de implementación son parecidos y menores a los de Windows Server.

4.3 EFICIENCIA DEL PROTOCOLO

Todos los protocolos analizados en este trabajo tienen el mismo reto de seguridad: la autenticación. Al considerar una aplicación de seguridad, la autenticación es un componente clave de cualquier solución de seguridad. La autenticación mutua, donde el cliente y el servidor deben autenticarse entre sí, se utiliza para garantizar que sólo a los usuarios autorizados se les permite el acceso a la red. Como se ha visto, Kerberos, RADIUS y LDAP son las más populares y útiles soluciones de autenticación que hacen frente a este desafío de seguridad en las redes de datos.

4.3.1 Eficiencia del protocolo Kerberos

Kerberos está diseñado para que dos partes puedan intercambiar información privada a través de una red, que de otra manera sería insegura. Kerberos proporciona autenticación mutua entre un cliente y un servidor, así como entre los servidores, antes de que una conexión de red se pueda abrir. Utiliza una técnica que consiste en un secreto compartido, que funciona como una contraseña. Esto sucede mediante la asignación de una clave única, llamada ticket, la cual se asigna a cada usuario que se conecta a la red. El ticket se incrusta en los mensajes para identificar al remitente del mensaje.

4.3.2 Eficiencia del protocolo RADIUS

Los servidores RADIUS son servidores robustos y escalables que proporcionan las funciones de autenticación, autorización y contabilidad (AAA), así como políticas avanzadas y gestión de configuración personalizada para controlar el acceso de usuarios a las redes cableadas. RADIUS y LDAP se utilizan a menudo juntos en algunas aplicaciones.

4.3.3 Eficiencia del protocolo LDAP

El Lightweight Directory Access Protocol (LDAP) es un extensible, un estándar de protocolo de red independiente del proveedor, un sistema de autenticación, y un servicio de directorio que se basa en el modelo de servicios de directorio X.500. LDAP es un repositorio de información, así como un protocolo para consultar y manipular los datos en un directorio LDAP. LDAP es uno de los directorios de autenticación más ampliamente utilizados en las redes modernas. LDAP se basa en las normas contenidas

en el estándar X.500, pero es mucho más simple y compatible con TCP/IP, que es necesario para cualquier tipo de acceso a Internet. Muchos de los dispositivos actuales de seguridad en redes, tienen soporte nativo para clientes LDAP.

En resumen, los tres protocolos cumplen con la tarea de autenticación de manera eficiente, pero cada uno de forma diferente e independiente; sin embargo, son protocolos que no están pelados entre sí sino que por el contrario pueden combinarse para formar sistemas mucho más robustos y seguros. Pero si de optar por uno se trata, se puede decir que el más completo en cuanto a seguridad, menor número de vulnerabilidades, y sobretodo el manejo de interfaces de administración amigables y más facilidades en cuanto a configuraciones se trata, es el protocolo LDAP.

En la Tabla 4-7 se muestra un resumen de las implementaciones realizadas, así como algunas de sus características.

PROTOCOLO	SISTEMA OPERATIVO	COMPLEJIDAD DE INSTALACIÓN	INTERFAZ	ADMINISTRACIÓN DE USUARIOS	CONFIGURACIÓN DE CLIENTE	INFORMACIÓN DE LOGS	CIFRADO
Kerberos	Windows Server 2008 Enterprise	Fácil: La instalación del sistema operativo y los componentes para utilizar el protocolo es muy intuitiva ya que solo es necesario seguir los pasos y confirmar opciones.	Amigable: Los menus y los textos incluidos en el sistema operativo son de facil acceso y uso para el administrador.	Fácil: La consola de administración de usuarios es muy comoda y accesible para agregar a cualquier usuario, ademas de tener varias opciones de configuración según las necesidades del administrador.	**Windows**: Agregar equipo a Dominio, Agregar perfil de usuario con privilegios correspondientes; **Linux**:Instalar Likewise, Configurar cuenta.	Muestran información sobre errores en general que el cliente envia, pueden o no evitar la conexión o ser importantes.	AES 128 y 256
	Ubuntu 8.04	Dificil: La instalación del sistema operativo no es compleja, pero la configuración del servidor requiere de conocimientos básicos en el uso de comandos Unix, ya que toda la instalación del servidor Kerberos es a nivel consola.	Complicada: No exíste una interfaz para manipular y consultar opciones del servidor, por lo que es necesario que el administrador tenga conocimientos básico en Unix.	Dificil: La admnistración de usuarios se hace compleja al no tener una consola dónde se pueda observar la unidad organizacional. Sólo es posible agregar usuarios por medio de comandos y cargas masivas de los mismos en archivos .txt.	**Windows:** Instalar programa MIT Kerberos para iniciar sesión en clientes Windows. Copiar achivo krb5.ini a C:\WINDOWS\system32 y archivos dll de configuración. Agregar IP de servidor kerberos a archivo hosts en ruta C:\WINDOWS\system32\drivers\etc. **Linux:**Copiar achivo krb5.conf y agregar IP de servidor kerberos a archivo hosts en ruta /etc.	Logs con información de inicio, caídas y fin de servicios en el servidor.	DES y 3DES

TABLA 4-7. ANÁLISIS COMPARATIVO

PROTOCOLO	SISTEMA OPERATIVO	COMPLEJIDAD DE INSTALACIÓN	INTERFAZ	ADMINISTRACIÓN DE USUARIOS	CONFIGURACIÓN DE CLIENTE	INFORMACIÓN DE LOGS	CIFRADO
Radius	Windows Server 2008 Enterprise	Fácil: La instalación del sistema operativo y los componentes para utilizar el protocolo es muy intuitiva ya que solo es necesario seguir los pasos y confirmar opciones. Requiere mas configuraciones y referencias sobre el tratamiento que da Windows Server 2008 a RADIUS, ya que lo llama NPS y es necesario realizar configuraciones especificas.	Amigable: Los menus y los textos incluidos en el sistema operativo son de facil acceso y uso para el administrador.	Fácil: La consola de administración de usuarios es muy comoda y accesible para agregar a cualquier usuario, ademas de tener varias opciones de configuración según las necesidades del administrador.	**Windows:** Agregar equipo a Dominio, Agregar perfil de usuario con privilegios correspondientes;	Muestran información sobre errores en general que el cliente envia, pueden o no evitar la conexión o ser importantes.	MD5 (Permite protección adicional configurando Ipsec, ESP y 3DES)
	Ubuntu Server 8.04 LTS	Difícil: Tanto la instalación del sistema operativo como la configuración del servidor Radius tiene un grado de complejidad alto, que requiere de ciertos conocimientos sobre sistemas Unix, ya que todo el proceso de instalación se debe hacer desde la consola o línea de comandos.	Muy complicada: La interfaz no es para nada amigable, ya que todo se realiza desde la consola; sin embargo se puede optar por la opción de configurar un servidor web para poder administrar remotamente desde cualquier otro equipo de la red a través de cualquier explorador de internet.	Difícil: La administración de usuarios se realiza desde la consola por medio de comandos, lo que resulta complicado; sin embargo, es posible instalar y configurar herramientas gráficas mediante las cuales esta tarea se vuelve considerablemente sencilla.	**Linux:** Agregar los usuarios a los archivos *users* y *clients.conf* del servidor.	Logs con información sobre las conexiones tanto exitosas como fallidas con el servidor de autenticación, así como errores en general.	WPA y WPA2.

TABLA 4-7. ANÁLISIS COMPARATIVO

PROTOCOLO	SISTEMA OPERATIVO	COMPLEJIDAD DE INSTALACIÓN	INTERFAZ	ADMINISTRACIÓN DE USUARIOS	CONFIGURACIÓN DE CLIENTE	INFORMACIÓN DE LOGS	CIFRADO
LDAP	Windows Server 2008 Enterprise	Fácil: La instalación del sistema operativo y los componentes para utilizar el protocolo es muy intuitiva ya que solo es necesario seguir los pasos y confirmar opciones.	Amigable: Los menus y los textos incluidos en el sistema operativo son de facil acceso y uso para el administrador.	Fácil: La consola de administración de usuarios es muy comoda y accesible para agregar a cualquier usuario, ademas de tener varias opciones de configuración según las necesidades del administrador.	**Windows:** Agregar equipo a Dominio, Agregar perfil de usuario con privilegios correspondientes;	Muestran información sobre errores en general que el cliente envia, pueden o no evitar la conexión o ser importantes.	SSL (Secure Sockets Layer)
	Fedora Core 9	Fácil: La instalación del sistema operativo es muy sencilla ya que se cuenta con la ayuda de una interfaz gráfica, y lo único en lo que se puede demorarse un poco es en definir bien el tamaño de las particiones. Por otro lado, la instalación del servidor LDAP, en este caso Fedora Directory Server, se realiza desde la consola, sin embargo los pasos son sencillos y no se requiere de configuraciones complejas.	Muy amigable: El sistema dispone de varias herramientas gráficas para la gestión de usuarios y demás configuraciones del sistema.	Fácil: La consola de administración de usuarios es muy comoda y accesible para agregar a cualquier usuario, ademas de tener varias opciones de configuración según las necesidades del administrador, de hecho es muy similar a la de LDAP para sistemas Windows.	**Windows y Linux:** Agregar equipo al Dominio, Agregar usuario y asignarlo el perfil o perfiles correspondientes.	Los logs son muy completos, muestran errores en general, conexiones con el servidor, duración de las sesiones y estadísticas generales sobre los usuarios.	SSL (Secure Sockets Layer)

TABLA 4-7. ANÁLISIS COMPARATIVO

CONCLUSIONES

En todo sistema informático es de vital importancia proteger cada uno de sus componentes, como son: hardware, software, datos, memoria y usuarios. Cualquiera de estos es vulnerable y proclive a ataques. Para mejorar la seguridad de un sistema de información se puede hacer uso de los servicios de seguridad cuya meta es evitar los ataques a sus componentes, evitando así la interrupción, intercepción, modificación y generación de contenido diferente al que el usuario original pretende enviar. Si se cumplen estos servicios de seguridad se considera que los datos están protegidos. Los servicios son: confidencialidad, integridad, disponibilidad, no repudio, control de acceso, autenticación.

El control de acceso se refiere al dispositivo que controla el acceso a un medio de información, tal como un monitor por el cual se verifica si es posible el acceso o no. Este acceso requiere una verificación, en este caso, una autenticación por la cual se compruebe que la persona que requiere el acceso tiene la autorización para realizarlo. El servicio de autenticación asegura que la comunicación proviene de una fuente autentica y mantiene ese aspecto durante todo el proceso de comunicación. En general, el proceso utiliza algo que se tiene, algo que se sabe, algo que se es, y que poseen las fuentes que originan la comunicación.

Por otro lado, las vulnerabilidades de un servidor o servicio de autenticación son los agujeros o bugs de seguridad, documentados o no, que ayudan a los hackers a encontrar las puertas traseras de acceso a un sistema.

A medida que se van divulgando y se van haciendo conocidas, todas ellas se van solucionando por parte de los programadores que crean los programas; pero el principal problema consiste en que los administradores y los usuarios suelen ser mucho más lentos actualizando sus sistemas

Este trabajo aporta un estudio del servicio de la autenticación, donde se ofrece a un administrador de red la posibilidad de aplicar una política de seguridad uniforme en todos los dispositivos de red, basándose en la Autenticación, Autorización y Auditoría, conocida como AAA. Este tipo de política tiene dos ventajas: provee a un administrador

de red la capacidad de centralizar toda la información contable, y crea un nivel de acceso que pueden aplicarse uniformemente a través de la red. Los protocolos más utilizados asociados a la AAA son Kerberos, Remote Authentication Dial-In User Service (RADIUS) y Lightweight Directory Access Protocol (LDAP).

Esta política maneja conceptos claros respecto a lo que es la autenticación, definiéndole como el proceso en el que se identifica un usuario en un dispositivo. La autorización es lo que determina el nivel de acceso a los que un usuario tiene acceso. Ambos son registrados por la auditoría, que permite tener un control sobre cada uno de los usos que se le han dado a un dispositivo.

En este trabajo se han realizado implementaciones prácticas para observar el funcionamiento la autenticación en diferentes servidores implementados bajo plataformas en Windows y Unix, donde se observa que este servicio de seguridad proporciona una gran seguridad dentro del sistema informático, ya que al implementar este servicio en una red de datos se puede llevar una eficaz administración en la creación y asignación de perfiles para los usuarios que tienen acceso al sistema, controlando con esto la identidad y autorizar el acceso a los distintos recursos que conforman la red de datos.

Analizando las implementaciones realizadas de los servidores de autenticación que emplean los protocolos Kerberos, Radius y LDAP bajo las plataformas en Windows y Unix, se pueden comparar sus características y elegir la mejor opción de acuerdo a la necesidad del usuario en base a su complejidad de instalación, administración y optimización de recursos.

Para el rubro de “complejidad de instalación” se considera como el más adecuado al servidor de autenticación bajo la plataforma en Windows con los protocolos Kerberos, Radius y LDAP, esto porque la instalación de Windows Server soporta cualquiera de los tres protocolos mencionados, sólo basta con elegir el tipo de protocolo que se desea implementar y continuar aplicando las políticas deseadas. Esto para el administrador significa de gran utilidad ya que permite realizar una instalación con baja complejidad y sin necesidad de conocer conceptos de programación.

Se considera al servidor de autenticación Fedora Directory Server bajo el protocolo LDAP como la mejor opción para la administración de la red de datos, esto porque analizando su interfaz de usuarios resulta bastante amigable diseñar el árbol para la asignación de perfiles, así como realizar la autorización de los distintos recursos de la red de datos.

Los servidores de autenticación bajo la plataforma Unix se consideran como los más ideales para la optimización de recursos en el servidor donde se instale la aplicación, esto porque es menor el consumo de memoria RAM por la poca utilización de gráficos, además se necesita menor cantidad de disco duro para instalar un Sistema Operativo UNIX, que al instalar un servidor bajo plataforma Windows.

En el caso de los servidores de autenticación tratados en este trabajo de tesis, con los años de vida que tienen cada uno de ellos, se han encontrado gran cantidad de fallos de seguridad y vulnerabilidades en los métodos de encriptación utilizados y en sus módulos de funcionalidades. Por supuesto que si se mantienen dichos servidores actualizados y al día, con los últimos componentes, es muy difícil que esto suceda. Y esta es la primera y primordial tarea para la implementación de un servidor de autenticación: mantenerse informado y actualizado.

Finalmente, hay que recordar que la seguridad integral de un sistema la decide el componente de seguridad más débil que incorpore. Un método avanzado de autenticación debe ser el único permitido para el acceso; activar métodos avanzados de seguridad y dejar los más débiles también activados, reduce la seguridad total al más débil.

.

ANEXOS

ANEXO 1. INSTALACIÓN DE UBUNTU 8.04

A continuación se enumeran de manera general los pasos para instalar Ubuntu 8.04:

1. Insertar el CD con la versión de Ubuntu 8.04 en el CD-ROM del equipo de cómputo. Una vez que éste inicia se indica el idioma con el cuál trabajará la distribución a instalar, en este caso se selecciona español y se continúa con la instalación.
2. Seleccionar el tipo de teclado adecuado para el equipo de cómputo, en este caso se selecciona como tipo de teclado Latinoamérica.
3. A continuación se muestran las opciones para seleccionar la partición dónde se instalará el sistema operativo. Para este caso seleccionar la opción Personalizado, ya que se pretende que vivan en este equipo de cómputo dos o más Sistemas Operativos. Por lo tanto una vez seleccionada la opción Personalizado, se muestran las particiones existentes en el Disco Duro. Se debe seleccionar la partición libre y que fue creada previamente con el programa Hard Disk Manager desde el Sistema Operativo Windows XP. Esta partición libre es aproximadamente del tamaño de 5GB y será formateada al tipo Ext2. Cabe aclarar que en ésta partición se cargará el / (root) como punto de montaje.
4. Una vez configurada la partición dónde será instalado el Sistema Operativo Ubuntu, se muestra un resumen de las configuraciones mencionadas anteriormente. Si todo está debidamente configurado oprimir el botón Instalar.
5. Al terminar la instalación y reiniciarse el Sistema Operativo, se pedirá el nombre de usuario y su respectivo password. Una vez proporcionados estos datos se puede ingresar a la distribución de Ubuntu.
6. Como paso final se recomienda actualizar el Sistema Operativo, esto se hace desde el Gestor de Actualizaciones.

ANEXO 2. PRUEBAS DE CONEXIÓN CON CLIENTES PARA SERVIDORES WINDOWS

Tanto para Radius como para LDAP, Windows Server utilizamos active directory para instalar la red de datos, con lo cual el proceso de inicio de sesión es similar para ambos protocolos.

a) Inicio de sesión de dominio en Windows XP

La configuración de usuario en un equipo con sistema operativo Windows XP, bajo el protocolo Radius o LDAP en Microsoft Windows Server 2008, incluye agregar el equipo al dominio PROTOCOLOSFIR.ORG o PROTOCOLOSFIL.ORG, según corresponda, así como al usuario en caso de proporcionarle permisos de administrador.

Figura A2-1. Acceso a Windows XP en el dominio.

Posteriormente, al estar con la sesión iniciada en el dominio, podemos hacer uso de cualquiera de los servicios compartidos por el servidor y con los permisos otorgados por el mismo, en este caso, una carpeta compartida y una impresora.

b) Inicio de sesión de dominio en Windows 7

La configuración de usuario en un equipo con sistema operativo Windows 7, bajo el protocolo Radius o LDAP en Microsoft Windows Server 2008, incluye agregar el equipo al dominio PROTOCOLOSFIR.ORG o PROTOCOLOSFIL.ORG, según corresponda, así como al usuario en caso de proporcionarle permisos de administrador.

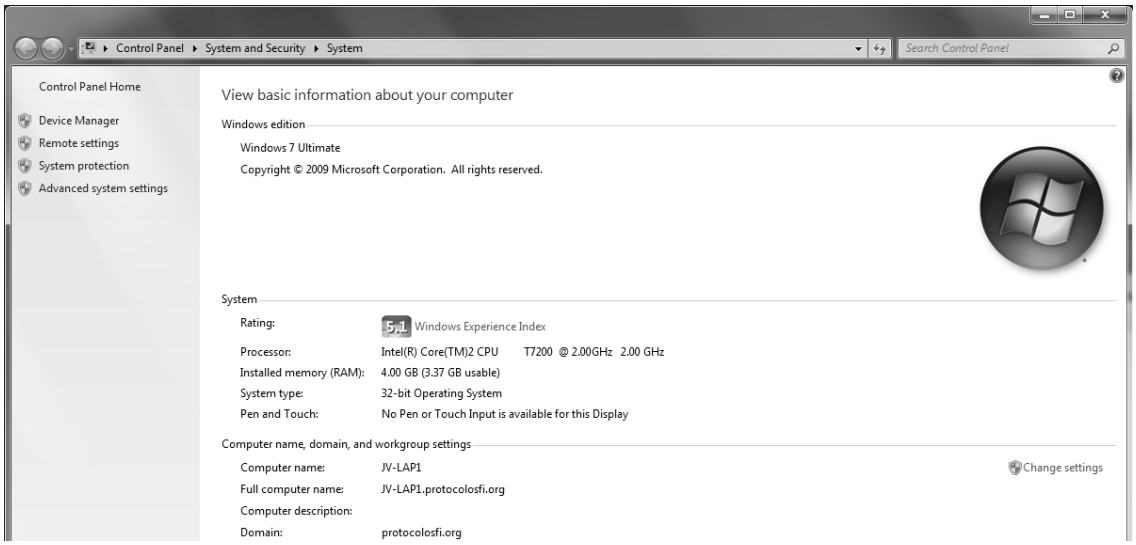

Figura A2-2. Equipo en dominio.

Posteriormente, al estar con la sesión iniciada en el dominio, podemos hacer uso de cualquiera de los servicios compartidos por el servidor y con los permisos otorgados por el mismo, en este caso, una carpeta compartida y una impresora.

c) Inicio de sesión de dominio en Ubuntu Linux

Para agregar un equipo con sistema operativo Linux, se procede a instalar el programa Likewise, con lo cual es fácil establecer la relación entre el servidor de dominio y el equipo de cómputo, aunque no pertenezcan al mismo ambiente.

Es necesario descargar el paquete LikewiseOpen-4.1.0.1846-linux e instalarlo o simplemente agregarlo por medio de Synaptic.

Finalmente es necesario configurar el programa Likewise con el dominio PROTOCOLOSFIR.ORG o PROTOCOLOSFIL.ORG, según corresponda, y la sesión a utilizar.

Figura A2-3. Configuración de Likewise.

Se ingresa al dominio dando clic en "Join Domain" y posteriormente se ingresa la contraseña y se verifica que el equipo se encuentra en el dominio correspondiente.

Figura A2-4. Dominio.

Una vez ingresado en el dominio se encuentra, los datos del perfil y se verifica que el usuario y el equipo pertenecen al dominio.

Figura A2-5. Propiedades de usuario en Ubuntu en dominio.

ANEXO 3. INSTALACIÓN DE UBUNTU SERVER 8.04 LTS

A continuación se enumeran de manera general los pasos para instalar Ubuntu Server 8.04 LTS:

1. Descargar y grabar la ISO de Ubuntu. La ISO se puede obtener desde la página oficial de descargas de Ubuntu (www.ubuntu.com/getubuntu/download). En ella se pueden ver las versiones disponibles, en este trabajo se ha optado por la versión 8.04, porque es una versión de tipo LTS (Long Term Support) con soporte hasta 2011.
2. Introducir el CD con la ISO grabada y bootear desde el CD/DVD autoarrancable de Ubuntu Server.
3. En el menú inicial cambiar el idioma de instalación a español y elegir la opción "Instalar en el disco duro". Por tratarse de un servidor, se debe asignar por norma general una dirección IP estática y no dinámica, por lo que es recomendable que antes del siguiente paso, se desconecte el cable de red, por si existiera algún servidor DHCP en la red o un router con el servidor DHCP funcional. De esta manera se evita que Ubuntu reciba una dirección IP de forma automática.
4. Al tener el cable de red desconectado o no haber un servidor DHCP funcional en la red, se dará un fallo de asignación y se debe continuar para llegar a la pantalla de configuración.
5. Ahora se procede a indicarle al instalador que se desea configurar manualmente el direccionamiento IP.
6. A continuación se especifica la dirección IP estática, que se va a utilizar desde este momento para el servidor RADIUS. Se ha decidido utilizar la 192.168.1.100, pero cada uno debe utilizar la que mejor se adapte a sus necesidades.
7. Se indica la máscara de subred que se va a utilizar. En este caso es una subred de clase C, por lo que se utiliza la 255.255.255.0.
8. Enseguida se configura la dirección IP del Gateway, que para este caso es la 192.168.1.254 (el router que nos da acceso a Internet para nuestra red). Esta misma dirección IP será la que se configure a continuación para el DNS.

9. Ahora se procede a asignar el nombre de host y el dominio que utilizará este servidor RADIUS para la red interna o pública. Se asigna el nombre *radius1*, en previsión de que posteriormente se pueda instalar algún otro servidor Proxy RADIUS o de redundancia. El dominio imaginario que se va a utilizar es "protocolosfi.org". Por lo tanto, se establece el nombre de la máquina como *radius1.protocolosfi.org*.

10. Tras la introducción de los datos de red, se continúa con el particionado y formateado del disco duro que va a alojar a Ubuntu Server. Dependiendo de la configuración de cada equipo, existen multitud de posibilidades de particionado a seleccionar, desde asignar el disco duro completo para Ubuntu (utiliza muy poco disco), hasta crear particionados dinámicos LVM o RAID. En este caso se han destinado 4GB del disco duro para instalar Ubuntu Server, por lo que se selecciona el método de partición marcado como "Manual".

11. En el espacio libre de 4GB se crea una partición primaria y se debe formatear como ext3 de Linux.

12. Una vez asignado el espacio y creada la partición, el instalador muestra el resumen de opciones de particionado que se han creado; cuando se tenga completamente claro que no existen equivocaciones en ninguna opción, se procede entonces con la aplicación de todos estos cambios presionando sobre la opción "Sí".

13. Enseguida vienen un par de pantallas para configurar la zona horaria y el reloj.

14. Posteriormente siguen tres pantallas en donde se configuran el nombre y apellidos del usuario principal del sistema, el nombre de inicio de sesión o alias para este usuario, y una contraseña. Luego viene una pantalla para verificar la contraseña asignada y se pide para ello que se vuelva a escribir.

15. En este momento, tras la configuración básica se produce la copia de archivos y la descarga y actualización de las dependencias necesarias. Es necesario que en este momento el equipo disponga de conexión a Internet para comprobar los repositorios de Ubuntu en búsqueda de actualizaciones.

16. Tras la instalación de los paquetes básicos, se procede a la instalación del Kernel de Linux y a su configuración de arranque.

17. Luego el instalador se va a conectar a Internet para la actualización de los repositorios de aptitude (apt), que es uno de los instaladores de paquetes binarios (programas) de Linux.

18. Después de que el instalador actualice y configure el apt, viene una pantalla que pregunta si se desea hacer uso de algún servidor proxy de la organización para el acceso a Internet. En este caso no se reencaminará el tráfico http por un proxy Server, por lo que se deja el campo vacío.

19. En la siguiente ventana se deben seleccionar los paquetes LAMP y OpenSSH para la conexión de sesiones remotas tuneladas.

20. Tras esta selección, comienza el copiado de los paquetes elegidos para su instalación.

21. En otro par de pantallas se debe establecer la contraseña del usuario *root* o superusuario de MySQL.

22. Tras un par de pantallas de instalación de los programas elegidos, se ha finalizado la instalación de Ubuntu Server Linux. Ya se tiene el servidor instalado y preparado para la configuración de todos los servicios que se van a utilizar.

23. Finalmente, se retira el disco de Ubuntu Linux y se reinicia el sistema para el primer arranque.

GLOSARIO

AUTENTICACIÓN – Acción y efecto de autenticar.

AUTENTIFICACIÓN – Acción y efecto de autentificar.

AUTENTICAR – Autorizar o legalizar algo.

AUTENTIFICAR – Autenticar (autorizar o legalizar algo).

BLOWFISH – En criptografía, Blowfish es un codificador de bloques simétricos, diseñado por Bruce Schneier en 1993 e incluido en un gran número de conjuntos de codificadores y productos de cifrado. Toma una clave de longitud variable, entre 32 y 448 bits. Mientras que ningún analizador de cifrados de Blowfish efectivo ha sido encontrado hoy en día, se ha dado más atención de la decodificación de bloques con bloques más grandes, como AES y Twofish.

CHALLENGE RESPONSE – El mecanismo de Challenge/Response (Desafío/Respuesta) tiene como objetivo principal realizar la validación de un usuario mediante su *Nombre de Usuario (username)* y *Password* evitando el traslado de esa información a través de la red. Está pensado sobre todo para redes de carácter público en las que se está expuesto a un ataque de sniffing o Man-in-the-middle.

DES – Data Encryption Standard, Estándar para el Cifrado de Datos. Algoritmo para el cifrado de datos, desarrollado por IBM, que utiliza bloques de datos de 64 bits y una clave de 56 bits.

GNU – Acrónimo de GNU is Not UNIX (o GNU No es UNIX). Sistema operativo libre diseñado por Richar Stallman, basado en programas que pueden ser descargados y modificados de forma gratuita por cualquiera.

GPG – GNU Privacy Guard. Es una herramienta para cifrado y firmas digitales, que viene a ser un reemplazo del PGP (Pretty Good Privacy) pero con la principal diferencia que es software libre licenciado bajo la GPL. GPG utiliza el estándar del IETF denominado OpenPGP.

HASH – Es un algoritmo que funciona en base a tomar una cadena y lo convierte en un código numérico. Un hash es un código, calculado en base en el contenido de un mensaje. Se utiliza en criptografía para la búsqueda rápida de datos y códigos de corrección de errores. El algoritmo está diseñado para que el rango de valores sea bastante extendido y las posibilidades de colisiones (dos cadenas que tienen el mismo valor de hash) sean mínimas. En criptografía, una contraseña puede ser enviada a un servidor y compara los valores hash almacenados allí. Esto evita que sean interceptadas contraseña en texto plano.

HDLC – High-level Data Link Control (Control de enlace de datos de alto nivel). Es un protocolo de enlace de datos orientados a bit diseñados para soportar la comunicación semidúplex y dúplex a través de enlaces punto a punto y multipunto.

Todos los protocolos orientados a bit están relacionados con el protocolo de control de enlace de datos de alto nivel (HDLC).

***INSTANCE* –** Forma de referirse a la ubicación o directorio de un proceso o archivo.

***JRE* –** Subconjunto de Java Development Kit (JDK), que contiene los ejecutables y los archivos del núcleo que constituyen la plataforma Java estándar. JRE comprende Java Virtual Machine (JVM), las clases del núcleo y los archivos de soporte.

LCP – Link Control Protocol (Protocolo de control de enlace). Es responsable del establecimiento, mantenimiento, configuración y terminación del enlace.

***LOGIN* –** Forma de referirse al modo de iniciar sesión en un sistema.

***LTS* –** Long Time Support, Soporte de Tiempo Largo. Expresa la idea que para las versiones de Ubuntu que tengan asociadas las siglas LTS, éstas tendrán un periodo de tiempo más extenso en cuanto al soporte que prestará Canonical, la empresa detrás de Ubuntu, ya sea en servicios o actualizaciones de seguridad. No todas las versiones de Ubuntu son LTS, esto es así para enfocar los esfuerzos en menos versiones y poder así ser más eficiente con los recursos a largo plazo. Las empresas son las más interesadas en las versiones LTS, el usuario común, cambia con mucha más rapidez entre versiones.

***MitM* –** En criptografía, un ataque *man-in-the-middle* (MitM o *intermediario*, en español) es un ataque en el que el enemigo adquiere la capacidad de leer, insertar y modificar a voluntad, los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado. El atacante debe ser capaz de observar e interceptar mensajes entre las dos víctimas. El ataque MitM es particularmente significativo en el protocolo original de intercambio de claves de Diffie-Hellman, cuando éste se emplea sin autenticación.

NAS – Un Network Access Server (servidor de acceso a la red) es el primer punto de entrada a una red de la mayoría de los usuarios de los servicios de red que se encuentran protegidos. Es el primer dispositivo de la red para prestar servicios a un usuario final, y actúa como una puerta de enlace para todos los servicios adicionales. Como tal, su importancia para los usuarios y los proveedores de servicios por igual es primordial. El NAS no contiene información acerca de qué clientes pueden conectarse o qué credenciales son válidas. Todos los NAS envían las credenciales suministradas por el cliente a un recurso que sabrá cómo procesar dichas credenciales.

***PASSWORD* –** Contraseña de un usuario para acceder a un sistema.

***PROMPT* –** Línea de comandos en un sistema operativo.

***RFC* –** Abreviatura de *Request For Comments* (Solicitud de Comentarios). Es el nombre que se da a una serie de normas que definen el protocolo TCP/IP, así como sus documentos relacionados.

***ROAMING* –** Tecnología que permite que el usuario de un teléfono móvil pueda utilizarlo en una red celular fuera de la cobertura de la red a la que pertenece,

permitiendo así hacer y recibir llamadas, por ejemplo, desde un país a otro. El término roaming significa callejeo o vagabundeo y sólo es posible si hay un acuerdo entre operadores de redes de telefonía móvil.

ROOT – Usuario principal con todos los privilegios de acceso sobre un sistema.

SMART CARD – Una tarjeta inteligente (*smart card*), o tarjeta con circuito integrado (TCI), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI. Las Tarjetas de memoria contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad. Las tarjetas microprocesadoras contienen memoria y microprocesadores.

SNIFFER – Programa que monitoriza los paquetes de datos que circulan por una red, en busca de información referente a cadenas prefijadas. Es un monitor de la red; es decir, un programa que mira todos los paquetes que pasan por la red.

TOKEN – Un token o también llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Ejemplos de tokens, podrían ser palabras clave (if, while, int,...), identificadores, números, signos, o un operador de varios caracteres (por ejemplo, :=). Son los elementos más básicos sobre los cuales se desarrolla toda traducción de un programa, surgen en la primera fase, llamada análisis léxico, sin embargo se siguen utilizando en las siguientes fases (análisis sintáctico y análisis semántico) antes de perderse en la fase de síntesis.

UNIX – Registrado oficialmente como UNIX®. Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy. Se utiliza principalmente como programa de control maestro en las estaciones de trabajo y en especial en los servidores.

X.509 – Estándar UIT-T para PKI (Public Key Infraestructure) infraestructura de claves publicas. X.509 especifica, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

802.1X – El protocolo IEEE 802.1X proporciona control de acceso en la Capa 2 de OSI (la Capa MAC). IEEE 802.1X soporta la autenticación de clientes mientras se establece la conexión a la red, antes de que al cliente se le asigne una dirección IP vía DHCP (Dynamic Host Configuration Protocol). Entre otras cosas, el estándar especifica como el protocolo de autenticación (EAP, Extensible Authentication Protocol) se encapsula en marcos Ethernet.

REFERENCIAS

[1] Ramió Aguirre Jorge, Aplicaciones Criptográficas, Primera Edición. Departamento de Publicaciones de la Escuela Universitaria de Informática de Madrid, 1998. Páginas 3-33, 297-304.

[2] Lopez Barrientos María Jaquelina, Quezada Reyes Cintia, Fundamentos de Seguridad Informática, Primera Edición. Facultad de Ingeniería UNAM, 2006. Páginas 118-125.

[3] Tenenbaum, Andrew S. Redes de Computadoras, Cuarta Edición. Pearson Education, 2003. Páginas 14-20, 37-41, 721-724.

[4] Convery, Sean. Network Security Architectures. Cisco Press, 2004.

[5] Lehtinen, Rick; Russell, Deborah; and Gangemi Sr., G.T. Computer Security Basics, Second Edition. O'Reilly, 2006.

[6] Liska, Allan, CISSP. The Practice of Network Security: Deployment Strategies for Production Environments. Prentice Hall PTR, 2002. Capítulo 6.

[7] Joshi, James…[et al.]. Network Security: Know It All. Morgan Kaufmann, 2008. Página 66.

[8] Forouzan A. Behrouz, Transmisión de Datos y Redes de Comunicaciones, Segunda Edición. Mc Graw Hill, 2002. Páginas 437-447.

[9] HASH - http://www.hitachi-id.com/concepts/hash.html
Última visita: 4 de Junio 2010

[10] Pérez Agudín Justo, Míguez Pérez Carlos, Matas García Abel Mariano, Picouto Ramos Fernando, Ramos Varón Antonio Ángel, La Biblia de Hacker, Edición 2006. Ediciones Anaya, 2006. Páginas 63-80, 433-434.+

[11] Kerberos MIT - http://web.mit.edu/kerberos/
Última visita: 4 de Junio 2010

[12] Instalación Kerberos http://www.marblestation.com/?p=735
Ultima visita: 10 de Julio 2010

[13] Fernández Hansen, Yago; Ramos Varón, Antonio; García-Morán, Jean P. AAA/RADIUS/802.1X: Sistemas basados en la autenticación en Windows y Linux/GNU, 1a. Edición, Alfaomega, 2009. Páginas 161-280, 383-485.

[14] Kerberos capítulo 4
http://web.mit.edu/kerberos/
Última visita: 4 de Junio 2010

[15] http://www.marblestation.com/?p=735
Última visita: 10 de Julio 2010

[16] Dominios y Grupos de Trabajo: Autenticación y Autorización
http://w2k8-server.spaces.live.com/Blog/cns!ACD63A60B4BAF014!173.entry
Última visita: 18 de Junio 2010

[17] Authentication Protocols – Cisco Systems
http://www.cisco.com/en/US/tech/tk59/tsd_technology_support_protocol_home.html
Última visita: 1 de Junio 2010

[18] Instalar Active Directory Windows Server 2008
http://serversandservers.wordpress.com/2008/10/29/20/
Última visita: 5 de Julio 2010

[19] chrisp – Kerberos for the Quick
http://chrisp.de/en/rsrc/kerberos.html
Última visita: 4 de Junio 2010

[20] AAA RADIUS authentication with Windows Server 2008
http://youritguy.wordpress.com/2009/10/02/aaa-radius-authentication-with-windows-server-2008/
Última visita: 19 de Junio 2010

[21] Windows Server 2008 – Instalar Active Directory Domain Services
http://www.aprendeinformaticaconmigo.com/windows-server-2008-instalar-active-directory-domain-services
Última visita: 5 de Junio 2010

[22] Windows Server 2008 – Crear un controlador de dominio
http://www.aprendeinformaticaconmigo.com/windows-server-2008-crear-un-controlador-de-dominio
Última visita: 14 de Junio 2010

[23] Windows Server 2008 – Añadir usuarios y equipos a los grupos.
http://www.aprendeinformaticaconmigo.com/windows-server-2008-anadir-usuarios-y-equipos-a-los-grupos
Última visita: 14 de Agosto 2010

[24] Fat of the LAN | Using Windows 2008 for RADIUS Authentication
http://www.fatofthelan.com/technical/using-windows-2008-for-radius-authentication/
Última visita: 14 de Agosto 2010

[25] Configuring Server 2008 for RADIUS Authentication
http://www.bunkerhollow.com/blogs/matt/archive/2008/06/04/configuring-server-2008-for-radius-authentication.aspx
Última visita: 14 de Agosto 2010

[26] Understanding the new Windows Server 2008 Network Policy Server
http://www.windowsnetworking.com/articles_tutorials/Understanding-new-Windows-Server-2008-Network-Policy-Server.html
Última visita: 14 de Agosto 2010

[27] Windows Server 2008: how to configure Network Policy Server or Radius Server
http://araihan.wordpress.com/2009/11/11/windows-server-2008-how-to-configure-network-policy-server-nps-or-radius-server/
Última visita: 28 de Agosto 2010

[28] Network Policy and Access Services
http://technet.microsoft.com/es-es/library/cc753220(WS.10).aspx
Última visita: 28 de Agosto 2010

[29] Microsoft TechNet Home Page
http://technet.microsoft.com/es-es/default.aspx
Última visita: 30 de Octubre 2010

[30] Diccionario de la lengua española
http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=cultura
Última visita: 23 de Octubre 2010

[31] Manual de instalación de FDS (Fedora Directory Server)
http://katherine057.blogspot.com/2008/08/instalacin-de-fds-fedora-directory.html
Última visita: 17 de Octubre 2010

[32] Manual Directory Server en Fedora 9
http://www.scribd.com/doc/8571957/Manual-Directory-Server-en-Fedora-9
Última visita: 17 de Octubre 2010

[33] The Kerberos Network Authentication Service RFC 4120
http://www.ietf.org/rfc/rfc4120.txt
Última visita: 17 de Octubre 2010

[34] Remote Authentication Dial In User Service (RADIUS) RFC 2865
http://www.ietf.org/rfc/rfc2865.txt