

**PROPUESTA PARA LA IMPLEMENTACION DE SEGURIDAD A NIVEL DE RED EN LA
RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA**



Claudia Patricia Arenas Guerrero

Julián Andrés Parra Chacón

Anexos

Director: Ing. Francisco Javier Terán

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2006**

ANEXO A

1. SEGURIDAD EN REDES INALÁMBRICAS DE AREA LOCAL

Las redes LAN inalámbricas de alta velocidad ofrecen las ventajas de conectividad de red sin las limitaciones que supone estar atado a una ubicación por cables. Existen numerosos escenarios en los que este hecho puede ser de interés. La Universidad del Cauca cuenta con varios proyectos de redes inalámbricas, entre ellos y el más importante, es una conexión inalámbrica entre la sede de la Facultad de Ingenierías y el municipio de Silvia en el marco del proyecto EHAS. Este proyecto busca brindar beneficios sociales en aspectos de salud a varias poblaciones del departamento del Cauca. Por este motivo y debido a que una infraestructura de red inalámbrica adicional podría ser implementada en cualquier momento en la red de datos de la Universidad del Cauca, las redes de área local inalámbricas se constituyen en un tema de estudio para descubrir sus fortalezas y debilidades en cuanto a seguridad se refiere.

1.1 TOPOLOGÍAS DE REDES LAN INALÁMBRICAS

Las redes LAN inalámbricas se construyen utilizando dos topologías básicas. Para estas topologías se utilizan distintos términos, como administradas y no administradas, alojadas y par a par, e infraestructura y "ad hoc". Estos términos están relacionados, esencialmente, con las mismas distinciones básicas de topología.

1.1.1 Modo Infraestructura

Una topología de infraestructura es aquella que extiende una red LAN con cable existente para incorporar dispositivos inalámbricos mediante una estación base, denominada *punto de acceso*. El punto de acceso une la red LAN inalámbrica y la red LAN con cable y sirve de controlador central de la red LAN inalámbrica. El punto de acceso coordina la transmisión y recepción de múltiples dispositivos inalámbricos dentro de una extensión específica; la extensión y el número de dispositivos dependen del estándar de conexión inalámbrica que se utilice y del producto. En la modalidad de infraestructura, puede haber varios puntos de acceso para dar cobertura a una zona grande o un único punto de acceso para una zona pequeña, ya sea un hogar o un edificio pequeño como se muestra en la figura 1.

El dispositivo inteligente, denominado "estación" en el ámbito de las redes LAN inalámbricas, primero debe identificar los puntos de acceso y las redes disponibles. Este proceso se lleva a cabo mediante el control de las tramas de señalización procedentes de los

puntos de acceso que se anuncian a sí mismos mediante el sondeo activo de una red específica con tramas de sondeo. La estación elige una red entre las que están disponibles e inicia un proceso de autenticación con el punto de acceso. Una vez que el punto de acceso y la estación se han verificado mutuamente, comienza el proceso de asociación. La asociación permite que el punto de acceso y la estación intercambien información y datos de capacidad. El punto de acceso puede utilizar esta información y compartirla con otros puntos de acceso de la red para diseminar la información de la ubicación actual de la estación en la red. La estación sólo puede transmitir o recibir tramas en la red después de que haya finalizado la asociación.

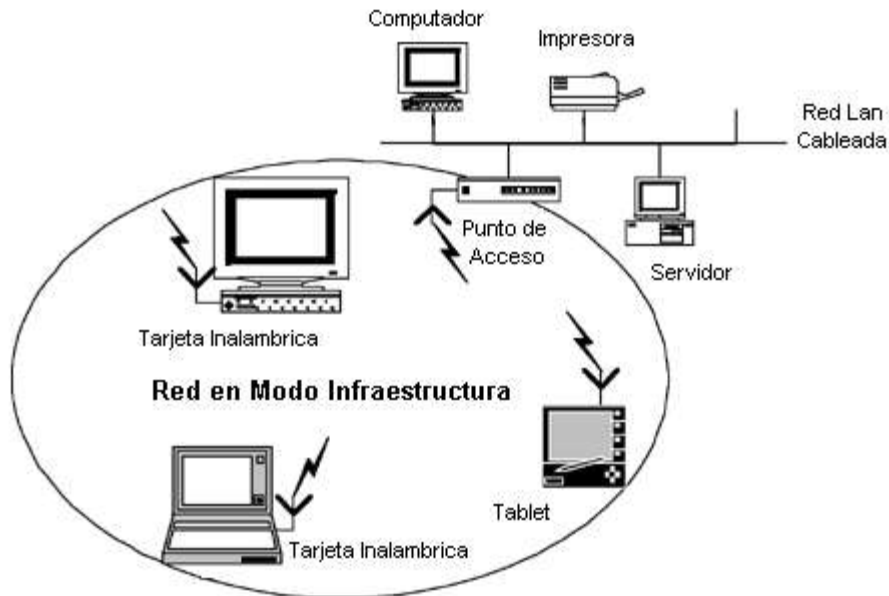


Figura 1. Red Inalámbrica en modo infraestructura

En la modalidad de infraestructura, todo el tráfico de red procedente de las estaciones inalámbricas pasa por un punto de acceso para poder llegar a su destino en la red LAN con cable o inalámbrica. El acceso a la red se administra mediante un protocolo que detecta las portadoras y evita las colisiones. Las estaciones se mantienen a la escucha de las transmisiones de datos durante un período de tiempo especificado antes de intentar transmitir (ésta es la parte del protocolo que detecta las portadoras). Antes de transmitir, la estación debe esperar durante un período de tiempo específico después de que la red está despejada. Esta demora, junto con la transmisión por parte de la estación receptora de una confirmación de recepción correcta, representa la parte del protocolo que evita las colisiones. Se puede observar que, en la modalidad de infraestructura, el emisor o el receptor es siempre el punto de acceso.

Dado que es posible que algunas estaciones no se escuchen mutuamente, aunque ambas estén dentro del alcance del punto de acceso, se toman medidas especiales para evitar las colisiones. Entre ellas, se incluye una clase de intercambio de reserva que puede tener lugar antes de transmitir un paquete mediante un intercambio de tramas "petición para emitir" y "listo para emitir", y un vector de asignación de red que se mantiene en cada estación de la red. Incluso aunque una estación no pueda oír la transmisión de la otra estación, oír la transmisión de "listo para emitir" desde el punto de acceso y puede evitar transmitir durante ese intervalo.

El proceso de movilidad de un punto de acceso a otro no está completamente definido en el estándar. Sin embargo, la señalización y el sondeo que se utilizan para buscar puntos de acceso y un proceso de reasociación que permite a la estación asociarse a un punto de acceso diferente, junto con protocolos específicos de otros fabricantes entre puntos de acceso, proporcionan una transición fluida. La sincronización entre las estaciones de la red se controla mediante las tramas de señalización periódicas enviadas por el punto de acceso. Estas tramas contienen el valor de reloj del punto de acceso en el momento de la transmisión, por lo que sirve para comprobar la evolución en la estación receptora. La sincronización es necesaria por varias razones relacionadas con los protocolos y esquemas de modulación de las conexiones inalámbricas.

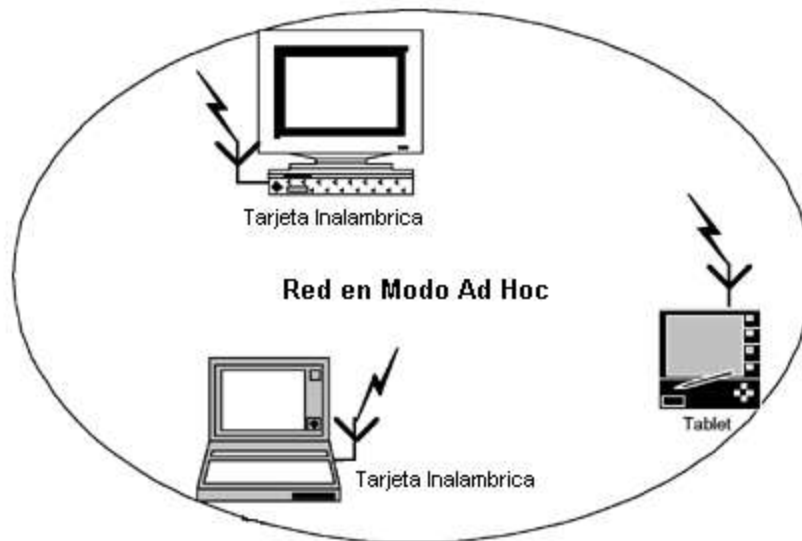


Figura 2. Topología de una red en modo Ad Hoc

1.1.2 Modo Ad Hoc

En una topología ad hoc, los propios dispositivos inalámbricos crean la red LAN y no existe ningún controlador central ni puntos de acceso. Cada dispositivo se comunica directamente con los demás dispositivos de la red, en lugar de pasar por un controlador central. Esta topología es práctica en lugares en los que pueden reunirse pequeños grupos de equipos que no necesitan acceso a otra red. Ejemplos de entornos en los que podrían utilizarse redes inalámbricas ad hoc serían un domicilio sin red con cable o una sala de conferencias donde los equipos se reúnen con regularidad para intercambiar ideas. La topología de la red se ilustra en la figura 2.

No tiene punto de acceso. En esta red sólo hay dispositivos inalámbricos presentes. Muchas de las operaciones que controlaba el punto de acceso, como la señalización y la sincronización, son controladas por una estación. La red *Ad Hoc* no disfruta todavía de algunos avances como retransmitir tramas entre dos estaciones que no se "ven" mutuamente.

1.2 ESTÁNDARES 802.11

Las redes de área local inalámbricas son conocidas como redes Wi-Fi. Wi-Fi (Wireless Fidelity) es un nombre comercial desarrollado por un grupo de comercio industrial llamado Wi-Fi Alliance (Inicialmente 3Com, Aironet, Harris, Lucent, Nokia y Symbol Technologies, hoy en día más de 150 miembros); el nombre oficial de esta alianza es WECA (Wireless Ethernet Compatibility Alliance) y son los primeros responsables del estándar más conocido, el 802.11b.

La familia de estándares 802.11 son una gran cantidad de complejos modos de operación para las redes inalámbricas de área local por ello se describirá muy brevemente cada una de ellos resaltando sus principales características:

- 802.11a: (5,15,2 Ghz, 5,25,3 Ghz, 5,7-5,8 GHz), 54 Mbps. OFDM: Multiplexación por división de frecuencias ortogonal. Rudimentario sistema de cifrado denominado WEP (Wired Equivalent Privacy).
- 802.11b: (2,4-2,485 GHz), 11 Mbps. Se mantiene WEP.
- 802.11c: Define características de puntos de acceso como Bridges.
- 802.11d: Múltiples dominios reguladores (restricciones de países al uso de determinadas frecuencias).
- 802.11e: Calidad de servicio (QoS).
- 802.11f: Protocolo de conexión entre puntos de acceso (AP), protocolo IAPP: Inter Access Point Protocol.
- 802.11g: (2,4-2,485 GHz), 36 o 54 Mbps. OFDM: Multiplexación por división de frecuencias ortogonal. Aprobado en 2003 para dar mayor velocidad con cierto grado de compatibilidad a equipamiento 802.11b.
- 802.11h: DFS: Dynamic Frequency Selection, habilita una cierta coexistencia con HiperLAN y regula también la potencia de difusión.
- 802.11i: Seguridad. Aprobada solo hasta Julio de 2004.
- 802.11j: Permitiría armonización entre IEEE (802.11), ETSI (HiperLAN2) y ARIB (HISWANa).
- 802.11m: Mantenimiento de redes wireless

1.3 SEGURIDAD EN LAS REDES INALÁMBRICAS

Debido a que la topología más usada dentro de las organizaciones es el modo infraestructura, incluyendo la Universidad del Cauca, este modo de operación es el que será estudiado. Por este motivo es necesario conocer algunos conceptos adicionales para comprender el funcionamiento de los mecanismos de seguridad usados en redes inalámbricas, como por ejemplo el ESSID y los Beacons Frames.

El ESSID (Extended Service Set Identifier) es un identificador que sirve para referenciar una red inalámbrica y consta de máximo 32 caracteres. Cada elemento de la red debe conocer el ESSID del punto de acceso al que pertenece para poder pertenecer a la red; si

no concuerdan, no podrá establecer ninguna comunicación entre el equipo inalámbrico y el punto de acceso y por ende no podrá comunicarse con cualquier otro dispositivo.

Los puntos de acceso envían constantemente anuncios de red con mensajes broadcast para que los clientes inalámbricos puedan detectar su presencia y así poder conectarse a la red inalámbrica. Estos anuncios son conocidos como *Beacon Frames*. Si se analizan las tramas de red de una red inalámbrica se observaría que normalmente el AP (Access Point) envía el ESSID de la red en los Beacon Frames; aunque esta característica viene por defecto, puede ser deshabilitada por software en la mayoría de puntos de acceso que se comercializan actualmente. Esto se hace ya que si un intruso conoce el ESSID puede fácilmente unirse a la red sin ser un usuario autorizado y realizar tanto funciones pasivas como funciones activas sobre la red.

Las medidas de seguridad más comúnmente tomadas por los administradores de redes inalámbricas son las siguientes:

- Crear ACL (Lista de Control de Acceso) basadas en las direcciones MAC de los clientes para que solo esas direcciones puedan comunicarse con el punto de acceso. Esta técnica puede ser fácilmente vulnerada debido a la facilidad de llevar a cabo una suplantación de la dirección MAC de un cliente.
- No emitir Beacon Frames, o emitirlos sin el ESSID.
- Utilizar WEP para cifrar los datos.

1.3.1 WEP (Privacidad Equivalente a la Cableada)

Las redes inalámbricas son de por sí más inseguras que las redes alámbricas ya que el medio físico por el que se transmiten los datos es el espectro electromagnético al cual cualquier persona puede tener acceso físico. Para proteger los datos que se envían a través de las WLAN el estándar 802.11 define el uso del protocolo WEP el cual intenta proveer la seguridad de una red cableada a una red inalámbrica cifrando los datos en las dos capas más bajas del modelo OSI (capa física y capa de enlace de datos). El protocolo WEP está basado en el algoritmo de cifrado RC4 y utiliza claves de 64 bits (40 bits reales), 128 bits (104 bits reales) y algunas marcas están introduciendo claves de 256 bits. El protocolo WEP es inseguro debido a su arquitectura y el hecho de aumentar el tamaño de las claves solo aumenta el tiempo para vulnerarlo. Para comprender por qué es tan fácil vulnerar este protocolo en tiempos relativamente cortos, es necesario comprender su funcionamiento.

1.3.1.1 Llaves Utilizadas por WEP

La llave de 40 ó 104 bits, se genera a partir de una clave (passphrase) estática de forma automática, aunque existe software que permite introducir esta llave manualmente. La clave o passphrase debe ser conocida por todos los clientes que quieran conectarse a la red inalámbrica que utiliza WEP; esto implica que muchas veces se utilice una clave fácil de recordar y que no se cambie de forma frecuente. A partir de la clave o passphrase se generan 4 llaves de 40 bits, sólo una de ellas se utilizará para la encriptación WEP. El proceso que se realiza para generar las llaves se muestra en la figura 3.

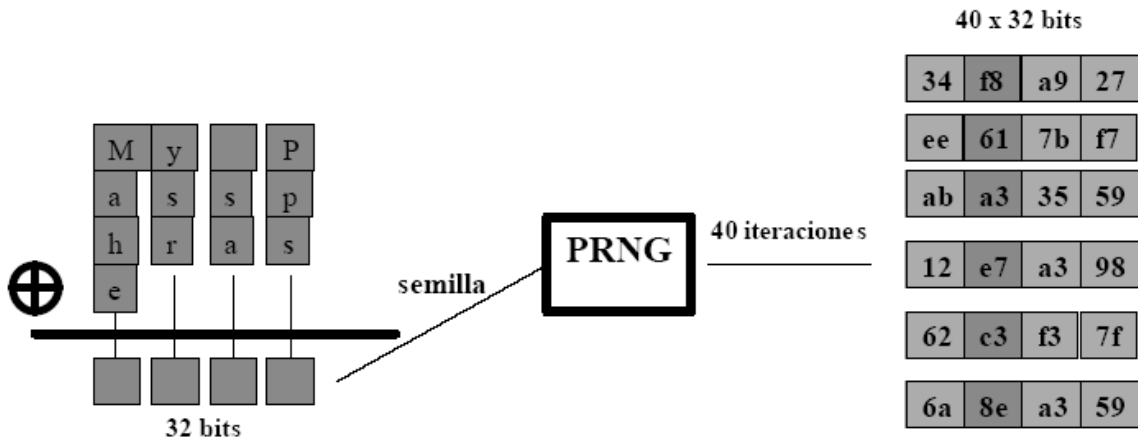


Figura 3. Generación de una clave para encriptación WEP

Se hace una operación XOR con la cadena ASCII (*My Passphrase*) que queda transformada en una semilla de 32 bits que utilizará el generador de números pseudoaleatorios (PRNG) para generar 40 cadenas de 32 bits cada una. Se toma un bit de cada una de las 40 cadenas generadas por el PRNG para construir una llave y se generan 4 llaves de 40 bits. De estas 4 llaves sólo se utilizará una para realizar la encriptación WEP.

1.3.1.2 Cifrado de Datos en WEP

Para generar una trama cifrada con WEP se sigue el siguiente proceso: se parte de la trama que se quiere enviar; esta trama sin cifrar está compuesta por una cabecera (Header) y contiene unos datos (Payload). El primer paso es calcular el CRC de 32 bits del payload de la trama que se quiere enviar. El CRC (Chequeo de Redundancia Cíclica), un método de identificación de errores basado en un algoritmo que genera un identificador único del payload en concreto, que nos servirá para verificar que el payload recibido es el mismo que el enviado, ya que el resultado del CRC será el mismo. Se añade este CRC a la trama como valor de chequeo de integridad (ICV - Integrity Check Value) como lo muestra la figura 4.

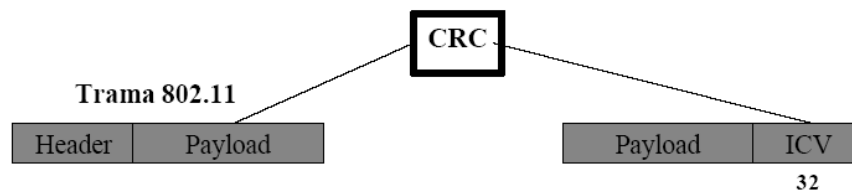


Figura 4. Cálculo del Chequeo de Redundancia Cíclica

Luego se selecciona una de las llaves de 40 bits de las 4 llaves posibles y se añade el vector de inicialización (IV) de 24 bits al principio de la llave seleccionada. Este proceso se observa en la figura 5.

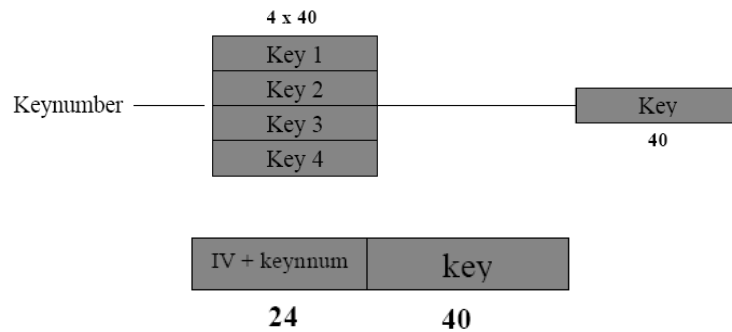


Figura 5. Selección de llave y adición del vector de inicialización

El IV es simplemente un contador que suele ir cambiando de valor a medida que se van generando tramas, aunque según el estándar 802.11 también puede ser siempre cero. Con el IV de 24 bits y la llave de 40 se consiguen los 64 bits de llave total que utilizaremos para cifrar la trama. En el caso de utilizar encriptación de 128 bits se tendrían 24 bits de IV y 104 de llave. En este punto, se aplica el algoritmo RC4 al conjunto IV+Key se obtiene el keystream o flujo de llave. Realizando una operación XOR con este keystream y el conjunto Payload+ICV se consigue el Payload+ICV cifrado, este proceso se detalla en la figura 6.

Se utiliza el IV y la llave para encriptar el Payload + ICV:

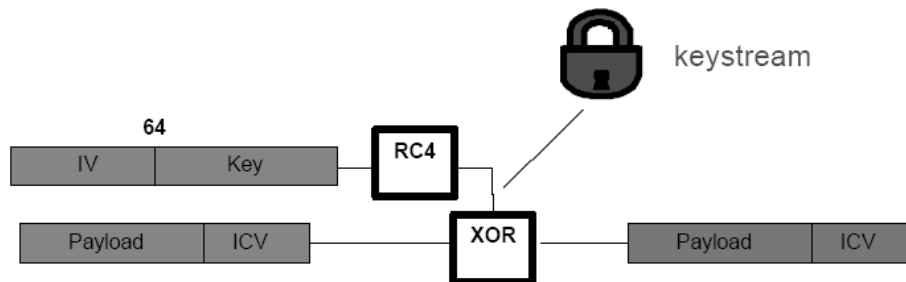


Figura 6. Cifrado del Payload de la trama 802.11

Después se adiciona la cabecera y el IV+Keynumbersin cifrar. Así queda la trama definitiva lista para ser enviada. El proceso de cifrado en conjunto se ve resumido en la figura 7.

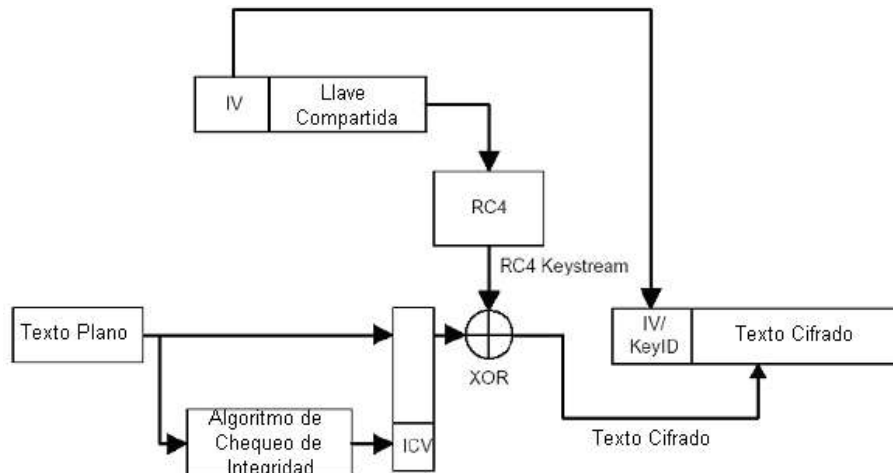


Figura 7. Diagrama de flujo del proceso de cifrado en WEP

1.3.1.3 Descifrado de Datos en WEP

Se utiliza el número de llave que aparece en claro en la trama cifrada junto con el IV para seleccionar la llave que se ha utilizado para cifrar la trama. Se añade el IV al principio de la llave seleccionada, consiguiendo así los 64 bits de llave. Aplicando RC4 a esta llave se obtiene el keystream válido para obtener la trama en texto plano realizando una XOR con el Payload+ICV cifrados y la llave completa. Una vez obtenido el plaintext, se vuelve a calcular el ICV del payload obtenido y se compara con el original.

1.3.2 Mecanismos de autenticación

1.3.2.1 Open System Authentication

Open system authentication es el protocolo de autenticación por defecto para 802.11b. Como su nombre lo indica, este método autentica a cualquier cliente que pide ser autenticado. Es un proceso de autenticación NULO, las tramas se mandan en texto plano aunque esté activado el cifrado WEP.

1.3.2.2 Shared Key Authentication

Este mecanismo utiliza una clave secreta compartida, que conocen cliente y AP. La estación que quiere autenticarse (cliente), envía una trama AUTHENTICATIONREQUEST indicando que quiere utilizar una "clave compartida". El destinatario (AP) contesta enviando una trama que contiene 128 octetos de texto (desafío) al cliente. El texto del desafío se genera utilizando el PRNG (generador de números pseudoaleatorios de WEP) con la clave compartida y un vector de inicialización (IV) aleatorio. Una vez el cliente recibe la trama, copia el contenido del texto de desafío en el payload de una nueva trama, que cifra con WEP utilizando la clave compartida (passphrase) y añade un nuevo IV (elegido por el cliente). Una vez construida esta nueva trama cifrada, el cliente la envía al AP, y éste descifra la trama recibida y comprueba que:

- El ICV (Integrity Check Value) sea válido (CRC de 32 bits).
- El texto de desafío concuerde con el enviado en el primer mensaje.

Si la comprobación es correcta, se produce la autenticación del cliente con el AP y se vuelve a repetir el proceso pero esta vez el primero que manda la trama con el AUTHENTICATION REQUEST es el AP. De esta manera se asegura una autenticación mutua.

1.3.3 Vulnerabilidades

- Puntos ocultos: Este es un problema específico de las redes inalámbricas, pues suele ser muy común que los propios empleados de la empresa por cuestiones de comodidad, instalen sus propios puntos de acceso. Este tipo de instalaciones, si no se controlan, dejan huecos de seguridad enormes en la red. El peor de estos casos es la situación en la cual un intruso lo deja oculto y luego ingresa a la red desde cualquier ubicación cercana a la misma. La gran ventaja que queda de este problema es que es muy fácil su identificación siempre y cuando se propongan medidas de auditorías periódicas específicas para las infraestructuras WiFi de la empresa, dentro del plan o política de seguridad.
- Falsificación de AP (Punto de Acceso): Es muy simple colocar una AP que difunda sus SSID, para permitir a cualquiera que se conecte, si sobre el mismo se emplean técnicas de "Phishing", se puede inducir a creer que se está conectando a una red en concreto.
- Deficiencias en WEP: Ya existen varias herramientas automáticas para descifrarlo.
- ICV independiente de la llave: Se trata de un control de integridad débil, cuya explotación permite inyectar paquetes en la red.
- Tamaño de IV demasiado corto: Como se mencionó, es el principal problema del protocolo WEP.
- Deficiencias en el método de autenticación: Si no se configura adecuadamente una red WiFi posee un débil método de autenticación, lo cual no permite el acceso, pero sí hacerse presente en la misma.
- Debilidades en el algoritmo key Scheduling de RC4: Este es el algoritmo de claves que emplea WEP, y con contraseñas débiles existen probabilidades de romperlo. Esto fue la sentencia definitiva para WEP.
- Debilidad en WPA: Nuevamente existe un tema de seguridad con el empleo de claves débiles (esto lo soluciona la versión dos de WPA).

1.3.4 Deficiencias en la encriptación WEP

1.3.4.1 Características lineales de CRC32

Esta vulnerabilidad fue demostrada teóricamente por Nikita Borisov, Ian Goldberg y David Wagner (Universidad de Berkeley). Como se ha observado anteriormente, el campo ICV (Integrity Check Value) de una trama encriptada con WEP contiene un valor utilizado para verificar la integridad del mensaje. Esto provee de un mecanismo de autenticación de mensajes a WEP, por lo tanto el receptor

aceptará el mensaje si el ICV es válido. El ICV se genera simplemente haciendo un CRC (Cyclic Redundancy Check) de 32 bits, del payload de la trama. Este mecanismo tiene dos graves problemas:

- Los CRCs son independientes de la llave utilizada y del IV
- Los CRCs son lineales.

1.3.4.2 MIC Independiente de la llave

Esta vulnerabilidad fue demostrada teóricamente por David Wagner (Universidad de Berkeley). Esta vulnerabilidad en WEP es conocida en inglés como "Lack of keyed MIC": Ausencia de mecanismo de chequeo de integridad del mensaje (MIC) dependiente de la llave. El MIC que utiliza WEP es un simple CRC-32 calculado a partir del payload, por lo tanto no depende de la llave ni del IV. Esta debilidad en la encriptación da lugar a que conocido el texto plano de un solo paquete cifrado con WEP sea posible inyectar paquetes a la red.

1.3.4.3 Tamaño de IV demasiado corto

Otra de las deficiencias del protocolo viene dada por la corta longitud del campo IV en las tramas 802.11b. El vector de inicialización (IV) tiene sólo 24 bits de longitud y aparece sin cifrar. Matemáticamente sólo hay 2^{24} (16.777.216) posibles valores de IV. Aunque esto pueda parecer mucho, 16 millones de paquetes pueden generarse en pocas horas en una red inalámbrica con tráfico intenso: Un punto de acceso que constantemente envíe paquetes de 1500 bytes (MTU) a 11 Mbps, acabará con todo el espacio de IV disponible después de $1500 \times 8 / (11 \times 10^6) \times 2^{24} \approx 1800$ segundos, o 5 horas. Este tiempo puede ser incluso más pequeño si la MTU es menor que 1500. La corta longitud del IV, hace que éste se repita frecuentemente y dé lugar a la deficiencia del protocolo, basada en la posibilidad de realizar ataques estadísticos para recuperar el texto plano gracias a la reutilización del IV.

1.3.4.4 Reutilización de IV

Esta vulnerabilidad fue demostrada teóricamente por David Wagner (Universidad de Berkeley). Se basa en que WEP no utiliza el algoritmo RC4 "con cuidado": el Vector de Inicialización se repite frecuentemente. Se pueden hacer ataques estadísticos contra textos cifrados con el mismo IV. Si un IV se repite, se pone en riesgo la confidencialidad.

El método de autenticación *Shared Key Authentication* descrito anteriormente se puede explotar fácilmente mediante un ataque pasivo: El atacante captura el segundo y el tercer *management messages* de una autenticación mutua (*Authentication Challenge* y *Authentication Response*). El segundo mensaje contiene el texto de desafío en claro, y el tercer mensaje contiene el desafío cifrado con la clave compartida. Como el atacante conoce el desafío aleatorio (texto plano, P), el desafío cifrado (texto cifrado, C), y el IV público, el atacante puede deducir el flujo pseudoaleatorio (keystream) producido usando WEP

1.3.5 Comprobando Deficiencias de WEP

En la actualidad las vulnerabilidades del protocolo WEP son muy conocidas, por ello se han desarrollado múltiples herramientas para el desciframiento de este protocolo. Una de las más conocidas es *airsnort*, este programa permite descifrar WEP por los métodos convencionales para atacar este protocolo, los cuales son: ataque por fuerza bruta, ataque de diccionario y ataque estadístico. Para llevar a cabo la práctica se utilizó un punto de acceso *D-Link DWL-2000*, configurado por interfase Web, un cliente del punto de acceso corriendo bajo *Windows XP* y un atacante corriendo sobre la distribución *Ubuntu*, la cual es una distribución basada en *Linux Debian*. Para que el punto de acceso brindara cifrado de datos vía WEP se accedió a la interfase Web del mismo y se configuró como lo muestra la figura 8.

The screenshot displays the 'Wireless Settings' page of a web interface. At the top, there is a navigation bar with tabs for 'Home', 'Advanced', 'Tools', 'Status', and 'Help'. Below the navigation bar, the 'Wireless Settings' section is visible. The configuration parameters are as follows:

- AP Name: DWL-2000AP+
- SSID: prueba1
- Channel: 3
- Authentication: Open System Shared Key WPA WPA-PSK
- WEP: Enabled Disabled
- WEP Encryption: 64Bit
- WEP Mode: ASCII
- Key1: segur
- Key2: [empty]
- Key3: [empty]
- Key4: [empty]

Figura 8. Configuración de WEP en punto de acceso

En la figura 8 se observa que el SSID de la red inalámbrica es *prueba1*, el canal de operación es el número 3, la autenticación se realiza por clave compartida, se habilita el cifrado por WEP, la llave de cifrado es de 64 bits y la clave compartida es la palabra *segur*.



Figura 9. Conexión a red inalámbrica de prueba

Posteriormente se configuró el cliente Windows. Para tal fin se hizo un barrido de las redes disponibles identificables con SSID. En este caso se escogió la red con SSID *prueba1* y se introdujo su llave compartida, la palabra *segur*, como lo muestra la figura 9. En las propiedades de la conexión inalámbrica se tiene la configuración mostrada en la figura 10.



Figura 10. Configuración de las propiedades de la conexión de prueba

La última parte de la práctica es configurar el atacante, lo cual inicia por la instalación de la herramienta *airsnort* por medio de la búsqueda en los repositorios de paquetes de *Ubuntu* con el manejador de paquetes *apt* con el siguiente comando:

```
#apt-get install aircrack-ng
```

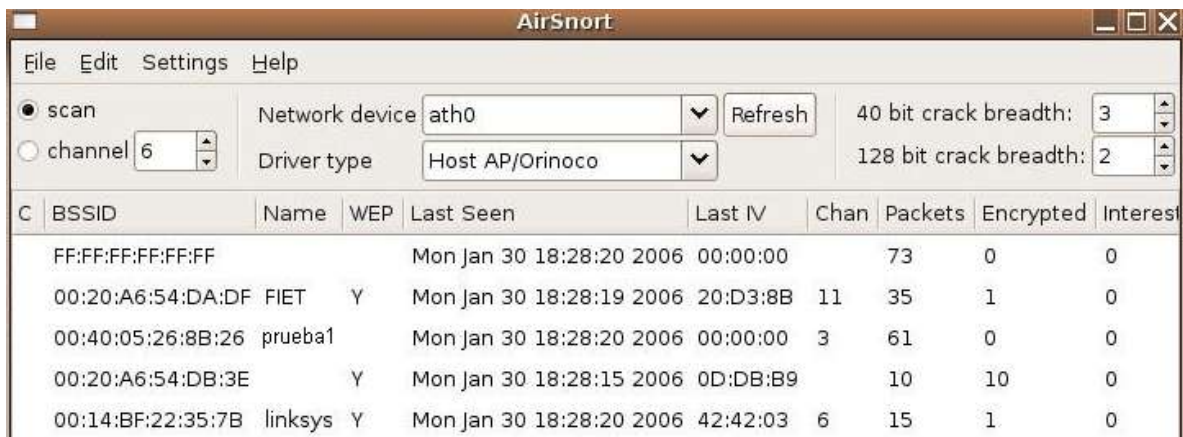
Aircrack-ng se ejecuta sobre una interfaz gráfica a la cual se accede por medio del comando del mismo nombre. El comando *aircrack-ng* desplegará la ventana mostrada en la figura 11.

En la figura 11 se puede notar que *aircrack-ng* busca en todos los canales de radiofrecuencia Wi-Fi, aunque puede hacerlo en uno específico y esto lo realiza por la interfaz inalámbrica *eth0*. *Aircrack-ng* puede guardar la captura realizada en el formato de la librería *libpcap* en este caso se guardó en un archivo llamado *captura.cap*, lo cual se realiza en el menú *File*. Luego se usó el programa *decrypt* para realizar un ataque de diccionario a los paquetes capturados. Previamente se debe haber descargado un archivo de diccionario el cual se puede encontrar en múltiples sitios en Internet, en este caso se llama *diccionario.txt*. El comando para realizar el ataque es el siguiente:

```
#root@ryst17:~# decrypt -f diccionario.txt -m 00:40:05:26:8B:26 -e captura.cap
```

La salida del comando anterior es:

```
Found key for 00:40:05:26:8B:26 - "segur"
```



C	BSSID	Name	WEP	Last Seen	Last IV	Chan	Packets	Encrypted	Interest
	FF:FF:FF:FF:FF:FF			Mon Jan 30 18:28:20 2006	00:00:00		73	0	0
	00:20:A6:54:DA:DF	FIET	Y	Mon Jan 30 18:28:19 2006	20:D3:8B	11	35	1	0
	00:40:05:26:8B:26	prueba1		Mon Jan 30 18:28:20 2006	00:00:00	3	61	0	0
	00:20:A6:54:DB:3E		Y	Mon Jan 30 18:28:15 2006	0D:DB:B9		10	10	0
	00:14:BF:22:35:7B	linksys	Y	Mon Jan 30 18:28:20 2006	42:42:03	6	15	1	0

Figura 11. Ejemplo de captura de paquetes por parte de aircrack-ng

Una vez se tiene la llave, se puede descifrar el tráfico capturado y colocarlo en otro archivo en formato *libpcap*. En este caso la figura 12 muestra el tráfico cifrado con WEP y la figura 13 muestra el tráfico descifrado con la clave compartida descifrada aplicando un filtro de visualización *ICMP*. Ya que el tráfico que estaba cifrado eran solicitudes y respuestas de eco ICMP entre el cliente Windows y el punto de acceso.

No. -	Time	Source	Destination	Protocol	Info
544	22.245390		D-Link_56:f8:f0	(R IEEE 802.11)	Acknowledgement
547	22.255119		D-Link_56:f8:f0	(R IEEE 802.11)	Acknowledgement
548	22.255887		D-Link_56:f8:f0	(R IEEE 802.11)	Acknowledgement

Figura 12. Tráfico cifrado por medio de WEP.

No. -	Time	Source	Destination	Protocol	Info
627	33.097347	172.16.2.15	172.16.41.113	ICMP	Echo (ping) request
857	37.679729	172.16.41.113	172.16.2.15	ICMP	Echo (ping) reply
1030	42.102521	172.16.2.15	172.16.41.113	ICMP	Echo (ping) request

Figura 13. Tráfico descifrado con la llave compartida descifrada.

1.3.6 Medidas de Seguridad en WiFi

- Emplear las mismas herramientas que los intrusos: realizar la misma actividad, pero para el “lado bueno”.
- Mejorar la seguridad física.
- Cancelar puertos que no se emplean.
- Limitar el número de direcciones MAC que pueden acceder. Esta actividad se realiza por medio de ACLs (Access Control List) en los AP.
- Satisfacer la demanda: si se están empleando AP no autorizados por parte de los empleados, es porque les resulta útil, por lo tanto, se pueden adoptar las medidas para que se implanten, pero de forma segura y controlada; de otra forma, seguirán apareciendo, pero de forma clandestina.
- Controlar el área de transmisión: todos los puntos de acceso inalámbrico permiten ajustar el poder de la señal.
- Implemente la autenticación de usuario: mejorar los puntos de acceso para usar las implementaciones de las normas WPA2 y 802.11i.
- Proteger la WLAN con la tecnología “VPNIPSec”: ésta es la forma más segura de prestar servicios de autenticación de usuario e integridad y confidencialidad de la información en una WLAN.
- Activar el mayor nivel de seguridad que soporta su hardware: incluso si se tiene un equipo de un modelo anterior que soporte únicamente WEP, asegúrese de activarlo. En lo posible, utilizar por lo menos WEP con un mínimo de encriptación de 128 bits.
- Instalar Firewalls personales y protección antivirus en todos los dispositivos móviles: la Alianza WiFi recomienda imponer su uso continuo.
- Adquirir equipamiento que responda a los estándares y certificado por “WiFi Alliance”.

ANEXO B

1. NOCIONES DE CRIPTOGRAFIA Y AUTENTICACIÓN

La criptografía involucra el estudio relacionado al diseño de sistemas para encriptar o cifrar información, y el criptoanálisis se basa en el proceso inverso, involucrando los sistemas para descifrar o descifrar códigos. Los algoritmos criptográficos proveen confidencialidad de datos al convertir un mensaje (texto plano) en texto no legible (cibertexto) y viceversa.

Los sistemas de criptografía se han clasificado como se muestra en la Figura 1. Los sistemas simétricos basan su cifrado y descifrado en una sola llave, mientras que los sistemas asimétricos o de llave pública, basan su seguridad en llaves diferentes, una privada para descifrar y una pública para cifrar.

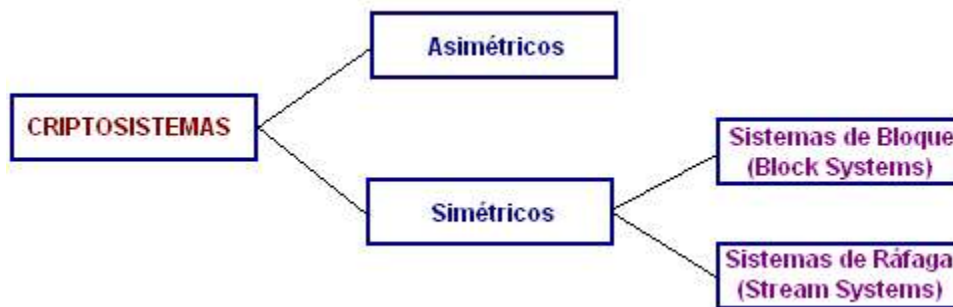


Figura 1. Clasificación de la Criptografía

Definiremos una clave (*Key*) como la llave que permite cifrar o descifrar la información recibida de forma correcta. Como se mencionó, los sistemas de criptografía actualmente se pueden clasificar en dos grandes grupos dependiendo de cómo distribuyan sus claves:

- *Sistema de clave privada o simétrica* (ver figura 2a). Este tipo de sistema se caracteriza por la existencia de una única clave que permite cifrar y descifrar los mensajes. Esto implica que tanto el emisor como el receptor comparten la misma clave. El principal inconveniente radica en cómo dar a conocer la clave privada únicamente al receptor. Este sistema fue el único utilizado hasta 1976.
- *Sistema de clave pública o asimétrica* (ver figura 2b). Este sistema desarrollado por Whitfield Diffie y Martin Hellman en 1976 se basa en que tanto el emisor como el receptor disponen de dos claves (una pública y otra privada). Estas dos claves

están relacionadas entre sí matemáticamente, pero la relación entre ellas no es trivial, con lo que el conocimiento de la clave pública e incluso de texto claro y texto cifrado no compromete la clave privada. Se dispone de una clave pública que se hará conocer a todos los posibles emisores (por e-mail o por autoridades de certificación) para que puedan cifrar los mensajes a enviar. La clave privada es guardada en secreto, puesto que es la que permitirá descifrar el mensaje cifrado con la clave pública. El mayor inconveniente de este sistema es autenticar la procedencia de los datos, ya que como cualquiera tiene acceso a la clave pública, puede cifrar información y enviarla con otro nombre. Un uso muy común del sistema de clave pública es en la firma de documentos, donde el emisor firma (cifra) un documento con su clave privada y cualquier puede verificarlo.

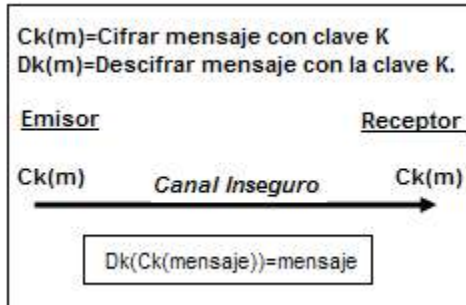


Figura 2a. Sistemas de clave privada o Simétricos.

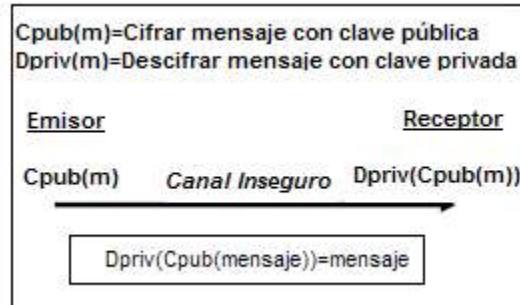


Figura 2b. Sistemas de clave Pública o Asimétricos

1.1 SISTEMAS DE CLAVE PRIVADA O SIMÉTRICA

En los sistemas de clave privada existen dos formas de cifrado: *Cifrado de Bloques* y *Cifrado de Flujo*.

1.1.1 Cifrado de bloques

Los sistemas basados en una única clave pueden cifrar la información en bloques (Block Ciphers) donde la información se divide en bloques de una misma longitud y son precisamente estos bloques los que son cifrados. Estos bloques se cifran independientemente unos de otros, utilizando un modo denominado ECB (Electronic Code Block). De esta forma, hasta que no se tenga la suficiente cantidad de información (un bloque) esta no puede ser cifrada y por tanto enviada (una solución que se adopta a veces es rellenar el espacio que falta hasta completar el tamaño del bloque con ceros o espacios en blanco, por ejemplo cuando el último bloque a transmitir no está completo y no deseamos enviar más información). No obstante, este tipo de cifrado hace cifrar el mismo texto de la misma salida cifrada. Para evitar esto existe en modo CBC (Cypher Block Chaining) donde el bloque ya cifrado se aprovecha para cifrar el siguiente. De esta forma la misma información no se cifrará igual dos veces. Algunos de los algoritmos de esta clase son:

1.1.1.1 Algoritmo DES:

DES (Data Encryption Standard) es un esquema de encriptación simétrico desarrollado en 1977 por el Departamento de Comercio y la Oficina Nacional de Estándares de EEUU en colaboración con la empresa IBM, que se creó con objeto de proporcionar al público en general un algoritmo de cifrado normalizado para redes de ordenadores. Estaba basado en la aplicación de todas las teorías criptográficas existentes hasta el momento, y fue sometido a las leyes de USA. Posteriormente se sacó una versión de DES implementada por hardware, que entró a formar parte de los estándares de la ISO con el nombre de DEA (Data Encryption Algorithm).

DES se basa en un sistema monoalfabético, con un algoritmo de cifrado consistente en la aplicación sucesiva de varias permutaciones y sustituciones. Inicialmente el texto en claro a cifrar se somete a una permutación, con bloque de entrada de 64 bits (o múltiplo de 64), para posteriormente ser sometido a la acción de dos funciones principales, una función de permutación con entrada de 8 bits y otra de sustitución con entrada de 5 bits, en un proceso que consta de 16 etapas de cifrado.

En general, DES utiliza una clave simétrica de 64 bits, de los cuales 56 son usados para la encriptación, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso, lo que le permite cifrar tanto en ECB como en CBC. Como la clave efectiva es de 56 bits, son posibles un total de 2 elevado a 56, es decir, unos 72.000 billones de claves, por lo que la ruptura del sistema por fuerza bruta o diccionario es sumamente improbable, aunque no imposible si se dispone de suerte y una gran potencia de cálculo. Existe una versión denominada DESX (DES eXtension), la cual es una extensión de la empresa RSA que eleva la clave DES a 120 bits.

Los principales inconvenientes que presenta DES son:

- Se considera un secreto nacional de EEUU, por lo que está protegido por leyes específicas, y no se puede comercializar ni en hardware ni en software fuera de ese país sin permiso específico del Departamento de Estado.
- La clave es corta, tanto que no asegura una fortaleza adecuada. Hasta ahora había resultado suficiente, y nunca había sido roto el sistema. Pero con la potencia de cálculo actual y venidera de los computadores y con el trabajo en equipo por Internet se cree que se puede violar el algoritmo, como ya ha ocurrido una vez, aunque en un plazo de tiempo que no resultó peligroso para la información cifrada.
- No permite longitud de clave variable, con lo que sus posibilidades de configuración son muy limitadas, además de permitirse con ello la creación de limitaciones legales.
- La seguridad del sistema se ve reducida considerablemente si se conoce un número suficiente de textos elegidos, ya que existe un sistema matemático, llamado Criptoanálisis Diferencial, que puede en ese caso romper el sistema en 2 elevado a 47 iteraciones.

Entre sus ventajas cabe citar:

- Es el sistema más extendido del mundo, el que más máquinas usan, el más barato y el más probado.
- Es muy rápido y fácil de implementar.
- Desde su aparición nunca ha sido roto con un sistema práctico.

Actualmente DES ya no es estándar y fue roto en Enero de 1999 con un poder de cómputo que efectuaba aproximadamente 250 millones de ensayos en un segundo.

1.1.1.2 Algoritmo Triple DES:

Como ya se vio, el sistema DES se considera en la actualidad poco práctico, debido a la corta longitud de su clave. Para solventar este problema y continuar utilizando DES se creó el sistema Triple DES (TDES), basado en tres iteraciones sucesivas del algoritmo DES, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con DES simple. Este hecho se basa en que DES tiene la característica matemática de no ser un grupo, lo que implica que si se encripta el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave. Para implementarlo, se toma una clave de 128 bits y se divide en 2 diferentes de 64 bits, aplicándose el siguiente proceso al documento en texto claro (Figura 3):

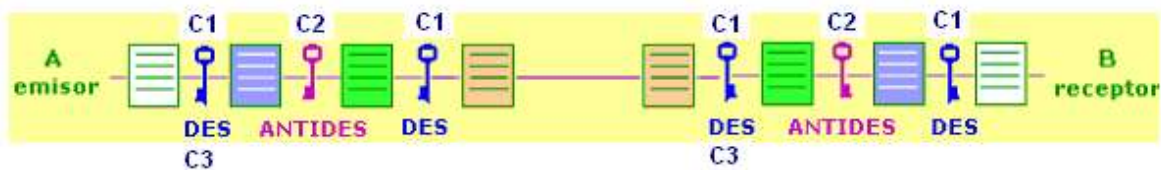


Figura 3. Algoritmo Triple DES

Se le aplica al documento a cifrar un primer cifrado mediante la primera clave, C1. Al resultado (denominado ANTIDES) se le aplica un segundo cifrado con la segunda clave, C2. Y al resultado se le vuelve a aplicar un tercer cifrado con la primera clave, C1. Si la clave de 128 bits está formada por dos claves iguales de 64 bits ($C1=C2$), entonces el sistema se comporta como un DES simple. Tras un proceso inicial de búsqueda de compatibilidad con DES, que ha durado 3 años, actualmente TDES usa 3 claves diferentes, lo que hace el sistema mucho más robusto, al conseguirse longitudes de clave de 192 bits (de los cuales son efectivos 168), mientras que el uso de DES simple no está aconsejado.

1.1.1.3 Algoritmo IDEA:

IDEA (International Data Encryption Algorithm), es un sistema criptográfico simétrico, creado en 1990 por Lai y Massey, que trabaja con bloques de texto de 64 bits, operando siempre con números de 16 bits usando operaciones como OR-Exclusiva, suma y multiplicación de enteros; utiliza una longitud de clave de 128 bits reales sin bits de paridad como en el DES.

El algoritmo de descifrado es muy parecido al de encriptación, por lo que resulta muy fácil y rápido de programar, y hasta ahora no ha sido roto nunca, aportando su longitud de clave una seguridad fuerte ante los ataques por fuerza bruta (prueba y ensayo o diccionarios). Este algoritmo es de libre difusión y no está sometido a ningún tipo de restricciones o permisos nacionales, por lo que se

ha difundido ampliamente, utilizándose en sistemas como UNIX y en programas de cifrado de correo como PGP (*Pretty Good Privacy*, o Seguridad bastante buena, uno de los primeros proyectos para cifrado de datos, hoy en día convertido en un estándar Internacional, descrito en el RFC2440).

1.1.1.4 Algoritmo RC5:

RC2 es un algoritmo propietario de la empresa RSA Data Security que tiene un tamaño de bloque de 64 bits, y permite utilizar los modos ECB (Electronic Code Block) y CBC (Cypher Block Chaining). Fue desarrollado como alternativa al DES y tiene una longitud de clave variable que va de 64 a 256 bits. El sistema criptográfico simétrico RC5 es el sucesor de RC4, frente al que presenta numerosas mejoras. RC4 consiste en hacer un XOR al mensaje con un arreglo que se supone aleatorio y que se desprende de la clave, mientras que RC5 usa otra operación, llamada dependencia de datos, que aplica shift a los datos para obtener así el mensaje cifrado. Ambos han sido creados por RSA Data Security Inc., la empresa creada por los autores del sistema RSA (Ronald Rivest, Adi Shamir y Leonard Adleman), que es actualmente una de las más importantes en el campo de los sistemas de cifrado y protección de datos. RC5 se caracteriza por permitir bloques de 32, 64 o 128 bits; su tamaño de clave varía de 0 a 2040 bits (255 bytes) y funciona como un generador de números aleatorios que se suman al texto mediante una operación de tipo OR-Exclusiva. Es además ampliamente configurable, permitiendo fijar diferentes longitudes de clave, número de iteraciones y tamaño de los bloques a cifrar, por lo que le permite adaptarse a cualquier aplicación. Por ejemplo, este algoritmo es el usado por Netscape para implementar su sistema de seguridad en comunicaciones SSL (Secure Socket Layer).

En cuanto a su seguridad, no es posible afirmar que es totalmente confiable, debido a que en 1996 una universidad francesa consiguió romper el sistema RC4 con clave de 40 bits, lo que hace sospechar que RC5 con longitudes de clave de 56 bits no es lo suficientemente seguro.

1.1.2 Cifrado de flujo

Esta forma consiste en cifrar la información en flujo (Stream Ciphers): de esta forma se escoge una unidad fundamental de información (por ejemplo un byte) y esta se va cifrando según es producida. No hay necesidad entonces de esperar hasta completar un bloque o rellenar el espacio sobrante en este.

1.1.2.1 Algoritmo RC4:

RC4 se caracteriza por utilizar la misma información de entrada que ha de cifrar para la generación de un número pseudoaleatorio que utilizará como clave, realizando un XOR entre la entrada y la clave. Esto significa que tanto el cifrado como el descifrado son operaciones idénticas. No se debe utilizar la misma clave más de una vez, ya que al utilizar un XOR como operación básica un atacante podría fácilmente descubrirla ($XOR(XOR(X)) = X$). La clave varía de 8 a 2048 bits. RC4 con MAC (Message Authentication

Code) es una extensión del RC4 que busca asegurar integridad en los datos mediante el uso de una función (MAC) que a partir del mensaje genera una secuencia de bits de tal forma que si es modificado (deliberadamente o no), el receptor puede saberlo.

1.2 SISTEMAS DE CLAVE PÚBLICA O ASIMÉTRICA

Los sistemas basados en clave pública se caracterizan por la presencia de un par de claves (una pública y otra privada) eliminando el mayor problema de los sistemas de clave privada, dar a conocer únicamente al receptor autorizado la clave usada en el sistema de cifrado/descifrado. No obstante introduce un nuevo problema, la autenticación del origen de los datos. Puesto que todo el mundo conoce la clave pública, se puede enviar un mensaje falseando la procedencia. En los sistemas de clave privada esto no pasaba, ya que la clave la compartían únicamente el emisor y el receptor de la información, asegurando la confidencialidad y la procedencia de la información. Las clave pública y privada están relacionadas matemáticamente, con lo que a partir de una es posible obtener la otra. No obstante esta relación no es directa, con lo que el hecho de conocer la clave pública (e incluso información y cómo esta es cifrada) no compromete la seguridad de este sistema. Para poder dar a conocer las claves públicas de los usuarios sin ningún riesgo, debemos asegurarnos que estas no pueden ser ni modificadas ni alteradas en ninguna forma. Con esta función se crearon las autoridades de certificación (Certification Authorities, CA), que son organismos encargados de distribuir las claves públicas y velar por ellas.

1.2.1 Algoritmo RSA

El algoritmo de clave pública RSA fue creado en 1978 por Rivest, Shamir y Adleman, y es el sistema criptográfico asimétrico más conocido y usado. Estos señores se basaron en el artículo de Diffie-Hellman sobre sistemas de llave pública, crearon su algoritmo y fundaron la empresa RSA Data Security Inc., que es actualmente una de las más prestigiosas en el entorno de la protección de datos.

El sistema RSA se basa en el hecho matemático de la dificultad de factorizar números muy grandes. Para factorizar un número el sistema más lógico consiste en empezar a dividir sucesivamente éste entre 2, entre 3, entre 4, ..., y así sucesivamente, buscando que el resultado de la división sea exacto, es decir, con residuo 0, con lo que ya se tiene un divisor del número.

Ahora bien, si el número considerado es un número primo (el que sólo es divisible por 1 y por él mismo), tendremos que para factorizarlo habría que empezar por 1, 2, 3, ... hasta llegar a él mismo, ya que por ser primo ninguno de los números anteriores es divisor suyo; si el número primo es lo suficientemente grande, el proceso de factorización es complicado y lleva mucho tiempo.

Basado en la exponenciación modular de exponente y módulo fijos, el sistema RSA crea sus claves de la siguiente forma:

- Se buscan dos números primos lo suficientemente grandes: p y q (de entre 100 y 300 dígitos).
- Se obtienen los números $n = p * q$ y $\phi = (p-1) * (q-1)$.
- Se busca un número e tal que no tenga múltiplos comunes con ϕ .
- Se calcula $d = e^{-1} \pmod{\phi}$, con $\text{mod} =$ residuo de la división de números enteros.

Ya con estos números obtenidos, n es la clave pública y d es la clave privada. Los números p , q y ϕ se destruyen. También se hace público el número e , necesario para alimentar el algoritmo. El cálculo de estas claves se realiza en secreto en la máquina en la que se va a guardar la clave privada, y una vez generada ésta conviene protegerla mediante un algoritmo criptográfico simétrico. En cuanto a las longitudes de claves, el sistema RSA permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits (se han roto claves de hasta 512 bits, aunque se necesitaron más de 5 meses y casi 300 ordenadores trabajando juntos para hacerlo).

RSA basa su seguridad en una función computacionalmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo ϕ no es factible a menos que se conozca la factorización de e , clave privada del sistema.

RSA es el más conocido y usado de los sistemas de clave pública, y también el más rápido de ellos. Presenta todas las ventajas de los sistemas asimétricos, incluyendo la firma digital, aunque resulta más útil a la hora de implementar la confidencialidad el uso de sistemas simétricos, por ser más rápidos. Se suele usar también en los sistemas mixtos para encriptar y enviar la clave simétrica que se usará posteriormente en la comunicación cifrada. A continuación se mencionan algunos aspectos importantes a la hora de utilizar RSA:

- La longitud de las claves

Existe una gran discusión, sobre este aspecto pero sin duda en la actualidad se acepta que es recomendable usar claves de longitud 768 (231 dígitos) para actividades personales, 1024 bits (308 dígitos) para corporaciones y 2048 (616 dígitos) para actividades de alto riesgo. La longitud de las claves tiene que ver con la seguridad del sistema si el número n pudiese ser factorizado entonces sin mucha dificultad puede calcularse d a partir de e , p , y q por lo tanto descifrar cualquier mensaje.

- La aleatoriedad de las claves

La generación de las claves RSA es muy importante, ya que muchos ataques son evitados si las claves son elegidas de forma aleatoria; esto incrementa la seguridad del sistema.

- Método de codificación

El método que actualmente es usado para aplicaciones en el esquema de cifrado es el OAEP (Relleno Óptimo de cifrado Asimétrico); este resiste los ataques conocidos actualmente.

- Elección de parámetros

La elección adecuada de los parámetros que se usan aumenta la seguridad del sistema así como su fácil y rápida implementación. Como elegir $a = e = 65537 = (01\ 00\ 01)_{16}$, para poder efectuar la operación exponente eficientemente. Además de elegir d , la clave privada, de longitud grande para evitar el ataque de Wiener. Los números primos p, q además de ser generados aleatoriamente deben tener la misma longitud y no estar cerca.

1.2.2 Algoritmo Diffie-Hellman

Este algoritmo de encriptación de Whitfield Diffie y Martin Hellman supuso una verdadera revolución en el campo de la criptografía en 1976, ya que fue el punto de partida para los sistemas asimétricos, basados en dos claves diferentes, la pública y la privada. Su importancia se debe sobre todo al hecho de ser el inicio de los sistemas asimétricos, ya que en la práctica sólo es válido para el intercambio de claves simétricas, y con esta funcionalidad es muy usado en los diferentes sistemas seguros implementados en Internet, como SSL (Secure Socket Layer) y VPN (Virtual Private Network). Este algoritmo proporciona autenticación y cifrado.

Matemáticamente se basa en las potencias de los números y en la función mod (módulo discreto). Uniendo estos dos conceptos se define la potencia discreta de un número como $Y = X^a \text{ mod } q$. Si bien el cálculo de potencias discretas es fácil, la obtención de su función inversa, el logaritmo discreto, no tiene una solución analítica para números grandes.

Para implementar el sistema se realizan los siguientes pasos:

- Se busca un número primo muy grande, q .
- Se obtiene el número β , raíz primitiva de q , es decir, que cumple que $\beta \text{ mod } q, \beta^2 \text{ mod } q, \dots, \beta^{q-1} \text{ mod } q$ son números diferentes.
- β y q son las claves públicas.

Para generar una clave simétrica compartida entre dos usuarios, A y B, ambos parten de un generador de números pseudoaleatorios, que suministra un número de este tipo diferente a cada uno, X_a y X_b . Estos son las claves privadas de A y B. Con estos números y las claves públicas β y q que ambos conocen, cada uno genera un número intermedio, Y_a e Y_b , mediante las fórmulas:

$$Y_a = \beta^{X_a} \text{ mod } q$$

$$Y_b = \beta^{X_b} \text{ mod } q$$

Estos números son intercambiados entre ambos, y luego cada uno opera con el que recibe del otro, obteniendo en el proceso el mismo número ambos:

$$K = Y_b^{X_a} \text{ mod } q$$

$$K = Y_a^{X_b} \text{ mod } q$$

Este número K es la clave simétrica que a partir de ese momento ambos comparten, y que pueden usar para establecer una comunicación cifrada mediante cualquiera de los sistemas asimétricos. Con este esquema, si se desea compartir una clave privada con otro usuario cualquiera, basta con acceder a su Y_u y enviarle la propia. Para facilitar este proceso se suelen publicar las Y_u de todos los usuarios interesados en un directorio de acceso común. Actualmente se está trabajando en sistemas basados en curvas elípticas como el ECC (Elliptic Curve Cryptosystem) de la empresa Certicom y la empresa RSA.

1.2.3 Sistemas CEE (de curvas elípticas)

CCE es otro tipo de criptografía de clave pública en el que se usan curvas elípticas definidas en un campo finito. La diferencia que existe entre este sistema y RSA es el problema matemático en el cual basan su seguridad. RSA razona de la siguiente manera: te doy el número 15 y te reto a encontrar los factores primos. El problema en el cual están basados los sistemas que usan curvas elípticas es el del logaritmo discreto elíptico, de la siguiente forma: se da el número 15 y el 3 y se reto a encontrar cuántas veces tienes que sumar el mismo 3 para obtener 15.

Los CCE son el mejor candidato para reemplazar a las aplicaciones que tienen implementado RSA, ya que estos definen también esquemas de firma digital, Intercambio de claves simétricas y otros.

1.3 INFRAESTRUCTURA DE CLAVE PÚBLICA – PKI

La estructura de claves Públicas (Public Key Infrastructure, PKI) se basa en la criptografía de clave pública, cuyos orígenes se remontan al artículo seminal de Diffie y Hellman en 1976, donde se explica la idea revolucionaria de utilizar, para las operaciones criptográficas, una pareja de claves, una pública, conocida por todos, y otra privada, sólo conocida por el usuario a quien le es asignada. Un mensaje puede ser cifrado por cualquier persona usando la clave pública, ya que es públicamente conocida, aunque sólo el poseedor de la clave privada podrá descifrarlo. Recíprocamente, un mensaje cifrado con la clave privada sólo puede ser cifrado por su poseedor, mientras que puede ser descifrado por cualquiera que conozca la clave pública.

Estas propiedades de que goza la criptografía de clave pública, cuyo uso más común se plasma en la *firma digital*, la convierten en candidata ideal para prestar servicios como la autenticación de usuarios (para asegurarse de la identidad de un usuario, bien como signatario de documentos o para garantizar el acceso a servicios distribuidos en red, ya que sólo él puede conocer su clave privada, evitando así la suplantación), el no repudio (para impedir que una vez firmado un documento el signatario se retracte o niegue haberlo redactado), la integridad de la información (para prevenir la modificación deliberada o accidental de los datos firmados, durante su transporte, almacenamiento o manipulación), la auditabilidad (para identificar y rastrear las operaciones), y el acuerdo de claves secretas para garantizar la confidencialidad de la información intercambiada, esté firmada o no.

Ahora bien, la mejor forma de asegurar que la clave pública de un usuario, que se ha encontrado en un directorio o una página Web, correspondierealmente a ese individuo y no ha sido falsificada por otro, consiste en recurrir a una tercera parte confiable, erigida en la figura de una *Autoridad de certificación (AC)*. La función básica de una AC reside en verificar la identidad de los solicitantes de certificados, crear los certificados y publicar listas de revocación cuando éstos son inutilizados. El certificado contiene de forma estructurada información acerca de la identidad de su titular, su clave pública y la AC que lo emitió. Actualmente, el estándar que se utiliza es el X.509v3.

Con el tiempo, una autoridad de certificación puede verse fácilmente desbordada si cubre un área geográfica muy extensa o muy poblada, por lo que a menudo delega en las llamadas *Autoridades de Registro (AR)* la labor de verificar la identidad de los solicitantes. Las AR pueden abrir multitud de oficinas regionales dispersas por un gran territorio, llegando hasta los usuarios en los

sitios más remotos, mientras que la AC se limitaría así a certificar a todos los usuarios aceptados por las AR dependientes de ella. Gracias a esta descentralización se agiliza el proceso de certificación y se aumenta la eficacia en la gestión de solicitudes.

En definitiva, una PKI incluirá una o varias autoridades de registro para certificar la identidad de los usuarios; una o varias autoridades de certificación que emitan los certificados de clave pública; un repositorio de certificados, accesible vía web u otro medio, donde se almacenen los certificados: las *Listas de Revocación de Certificados (CRL)*, donde se listan los certificados suspendidos o revocados y, por supuesto, los propios certificados.

La Infraestructura de Claves Públicas resulta ideal en una Intranet, en la que se comparten documentos, se accede a recursos de red, se intercambia correo certificado, entre otras cosas; PKI resulta mucho más ágil que los sistemas tradicionales de control basados en nombre y contraseña y listas de control de acceso. En el caso de una red externa o de Internet, PKI es de uso obligado; de hecho, es la única forma conocida actualmente de confirmar una comunicación segura con otro participante de la red del que no se tiene información verificable. Los protocolos SSL (Secure Socket Layer) y SET (Secure Electronic Transaction) se están convirtiendo en estándares de facto que atestiguan el éxito de las tecnologías de clave pública en escenarios de seguridad descentralizados como Internet.

1.3.1 Pre-Shared Keys

Este tipo de autenticación utiliza una única llave en texto plano la cual comparten todos los nodos que pertenecen a una red segura. Un nodo IPSec que recibe la conexión desde otro nodo IPSec, determina la validez de la conexión, si el nodo origen tiene la llave correcta.

Entre las ventajas de utilizar *Pre Shared Keys* se tiene la facilidad para configurar este tipo de autenticación y que esta disponible en la mayoría de las implementaciones IPSec. Una desventaja de este tipo de autenticación es la pobre gestión de llaves que utiliza.

1.3.2 Firmas digitales

Una firma digital es un bloque de caracteres que acompaña a un documento (o fichero), acreditando quiénes su autor (autenticación) y que no ha existido ninguna manipulación posterior de los datos (integridad). Para firmar un documento digital, el autor utiliza su propia *clave secreta*, a la que solo él tiene acceso, lo que impide que pueda después negar su autoría (no revocación); de esta forma, el autor queda vinculado al documento que firma. Cualquier persona puede verificar la validez de una firma si dispone de la *clave pública* del autor.

De esta forma, una firma digital es una secuencia de caracteres calculados a partir del documento original mediante unas funciones de resumen (Digest) o Hash (funciones que dada cualquier entrada producen una salida asociada a un rango determinado). Un simple ejemplo de una función Hash sería contar el número de letras del mensaje: si es par, se asocia un 0 y si es impar un 1; el principal

inconveniente de este sistema es que pueden existir colisiones (dos mensajes diferentes producen la misma salida) por lo que conviene que las funciones tengan un rango de salida lo suficientemente grande (128 bits o más) para poder considerarlas libres de colisión.

Los algoritmos más utilizados actualmente para el cálculo de la firma digital son el MD4, MD5 y SHA. Estas funciones producen una secuencia que acredita la autenticidad e integridad del documento ante terceras personas. La autenticación viene avalada por una autoridad de certificación (CA) en la cual se confía. El principal inconveniente junto con la falta de una normativa internacional común, se encuentra cuando el emisor y el receptor no comparten la misma autoridad de certificación. La solución adoptada actualmente es crear una jerarquía de autoridades de certificación, de esta forma, aunque dependan de dos entidades distintas siempre se podrá subir al nivel superior hasta encontrar una entidad común.

Para la interacción entre las diferentes autoridades de certificación (CA) y su reconocimiento mutuo, se utilizan las *Infraestructuras de clave pública*. Estas estructuras pueden negociar entre sí los certificados concedidos a sus usuarios y aceptar o denegar el de otras entidades. Este sistema de intercambio utiliza por lo general el estándar **X.509**.

1.3.2.1 Firmas Digitales RSA

Son firmas asincrónicas que se basan en el manejo de una pareja de llaves; cada llave puede cifrar información que solo la otra puede descifrar. La llave privada únicamente es conocida por su propietario; la llave pública se da a conocer abiertamente, pero sigue asociada al propietario.

Estas llaves se pueden usar de dos maneras: para garantizar la confidencialidad al mensaje y para probar la autenticidad del emisor del mensaje. En el primer caso, el emisor usa la llave pública del receptor para cifrar un mensaje, de manera que éste continúe siendo confidencial hasta que sea descifrado por el receptor con la llave privada. En el segundo caso, el emisor cifra un mensaje usando la llave privada, a la cuál solo él tiene acceso. De esta forma la llave pública del receptor asegura la confidencialidad y la llave privada del emisor verifica la identidad del mismo.

1.3.3 Certificados Digitales

Los certificados digitales permiten navegar por la red Internet, dando identidad al usuario y asegurando que pueda navegar con seguridad. De igual forma que la licencia de conducir o un pasaporte sirve para dar identidad a quien la porta en ciertos casos, el certificado digital da identidad a una clave pública y se comporta como una persona en el espacio cibernético.

El certificado digital nace con el fin de resolver el problema de administrar las claves públicas y que la identidad del dueño no pueda ser falsificada. La idea es que una tercera entidad intervenga en la administración de las claves públicas y asegure que las claves públicas tengan asociado un usuario claramente identificado.

Las tres partes más importantes de un certificado digital son:

- Una clave pública.
- La identidad del implicado: nombre y datos generales.
- La firma privada de una tercera entidad llamada *Autoridad Certificadora* que todos reconocen como tal y que valida la asociación de la clave pública en cuestión con el tipo que dice ser.

En la actualidad casi todas las aplicaciones de comercio electrónico y transacciones seguras requieren un certificado digital, y se ha propagado tanto su uso que se tiene ya un formato estándar de certificado digital, este es conocido como X.509 v. 3

Los certificados digitales X.509 no contienen únicamente el nombre de un usuario y la clave pública, sino también otra información acerca del usuario. Estos certificados son algo más que obstáculos en una jerarquía digital de confianza. Permiten a la CA proporcionar al destinatario de un certificado un medio de confianza de la clave pública del sujeto emisor del certificado y de otros datos acerca del mismo. Estos otros datos pueden ser, entre otras cosas, una dirección de correo electrónico, una autorización para firmar documentos de un determinado valor o la autorización para convertirse en una entidad emisora de certificados y firmar otros certificados.

Los certificados X.509 y muchos otros tienen un periodo de validez. Un certificado puede caducar y perder su validez. Una entidad emisora de certificados puede revocar un certificado por diversos motivos. Para controlar las revocaciones, la CA mantiene y distribuye una lista de certificados revocados denominada Lista de revocaciones de certificados (CRL). Los usuarios de la red tienen acceso a la lista para determinar la validez de un certificado.

ANEXO C

1. OTROS PROTOCOLOS DE SEGURIDAD

1.1 PROTOCOLOS A NIVEL DE RED

1.1.1 PPTP – Point to Point Tunneling Protocol

Point-to-Point-Tunneling Protocol (PPTP) es uno de los protocolos más populares a nivel de red y fue originalmente diseñado para permitir el transporte (de modo encapsulado) de protocolos diferentes al TCP/IP a través de Internet; además es una tecnología que soporta Redes Privadas Virtuales Multiprotocolo, habilitando a usuarios remotos para que accedan a redes corporativas de forma segura a través de estaciones de trabajo con sistema operativo Windows y otros sistemas con protocolos Punto a Punto (PPP) para acceder a un proveedor de servicio local y conectarse de forma segura a su red corporativa a través de Internet; esto es particularmente útil para personas que trabajan desde su casa o que viajan constantemente y deben acceder a una red corporativa remotamente a realizar diferentes actividades.

El Point-to-Point Tunneling Protocol de Microsoft está destinado a la creación de Redes Privadas Virtuales (VPN, del inglés Virtual Private Networks); son virtuales porque utilizan software para formar una conexión sobre una red pública (normalmente Internet), y son privadas porque codifican la información que transportan para impedir que alguien pueda leer la información mientras viaja a través de la red pública. Las VPN pueden incluir o soportar otros protocolos de red como IPX o NetBEUI dentro del protocolo TCP/IP; también pueden formar conexiones permanentes o de acceso telefónico entre diversos sitios. Las VPN se suelen utilizar en situaciones de acceso telefónico a redes en las que el usuario final establece de forma manual la red virtual para conectarse de forma temporal a una red remota. Un empleado que se encuentre fuera de la oficina, por ejemplo, puede conectarse a Internet a través de su proveedor de Internet y, a continuación, utilizar una VPN para establecer una conexión segura con la oficina de su empresa. El protocolo PPTP permite utilizar enlaces de Internet económicos para crear conexiones seguras entre ordenadores.

Como ya se ha visto, PPTP no es el único protocolo de red que puede utilizarse para crear redes privadas virtuales, aunque sin duda resulta fácil de adquirir y de utilizar; de hecho, a partir del sistema 95 se incluye gratuitamente. También se pueden obtener versiones para Macintosh en Network Telesystems¹, al igual que un cliente Linux. Así, se puede utilizar PPTP para crear redes privadas virtuales entre distintos sistemas operativos. Generalmente hay tres equipos involucrados en el uso del PPTP; hay un cliente PPTP, un servidor de acceso a la red y un servidor de PPTP. En el caso de una LAN, el servidor de acceso a la red no es necesario, porque ya está en la

¹ Network Telesystems, distribuidores de aplicaciones para Macintosh. (<http://www.nts.com>).

misma red. La comunicación segura creada usando el protocolo PPTP conlleva tres fases, cada una de las cuales requiere la finalización correcta de las anteriores. Estas son: PPP conexión y comunicación, PPTP control de conexión, PPTP data tunneling.

- *PPP conexión y comunicación:*

Primero el cliente necesita una conexión a Internet, conectando con un Servidor de Acceso a Red (NAS Network Access Server) vía un Proveedor de Servicios de Internet (ISP). Un cliente PPTP usa el PPP para establecer esta conexión. La conexión requerida por un cliente consiste en unas credenciales de acceso (usuario, password) y un protocolo de autenticación para que el servidor de PPTP pueda autenticar al cliente. Una vez conectado el cliente puede enviar y recibir paquetes sobre Internet.

- *PPTP control de conexión:*

Cuando el cliente tiene establecida la conexión PPP con el ISP, se realiza un segundo establecimiento de llamada, sobre la conexión PPP existente. Esto crea la conexión VPN (conexión de control) a un servidor PPTP de una LAN privada a una empresa y actúa como un túnel a través de la cual fluyen los paquetes de red. Un set de ocho mensajes de control establecerá, mantendrá y finalizará el túnel PPTP. Los mensajes son los siguientes:

- PPTP_START_SESSION_REQUEST: inicio de sesión.
- PPTP_START_SESSION_REPLY: Respuesta al requerimiento de inicio de sesión.
- PPTP_ECHO_REQUEST: Requerimiento de mantenimiento de sesión.
- PPTP_ECHO_REPLY: Respuesta al requerimiento de mantenimiento de sesión.
- PPTP_WAN_ERROR_NOTIFY: Reporte de error en la conexión PPP.
- PPTP_SET_LINK_INFO: Configuración de la conexión Cliente/Servidor.
- PPTP_STOP_SESSION_REQUEST: Fin de sesión.
- PPTP_STOP_SESSION_REPLY: Respuesta al requerimiento de fin de sesión.
- PPTP Data Tunneling.

Después de establecer el túnel PPTP, los datos son transmitidos entre el cliente y el servidor PPTP. Los datos son enviados en formato de datagramas IP que contienen paquetes PPP, a los que se refiere normalmente como paquetes PPP encapsulados. Los datagramas IP contienen paquetes IPX, NetBEUI, o TCP/IP y tiene el siguiente formato (Figura 1):



Figura 1. Datagrama PPP encapsulado

La cabecera IP de entrega proporciona la información necesaria para que el datagrama atraviese la red Internet. La cabecera GRE se usa para encapsular el paquete PPP dentro de un datagrama IP. La zona ensombrecida representa los datos encriptados. Después de que la conexión VPN está establecida, el usuario remoto (cliente) puede realizar cualquier operación como si fuera un usuario local.

- *La seguridad en PPTP*

Hay tres áreas en la seguridad PPTP que lo hace más atractivo. Son la autenticación, encriptación de datos y filtrado de paquetes PPTP. La autenticación de un cliente PPTP remoto se hace de la misma manera que la autenticación PPP usada por cualquier cliente RAS (Remote Access Service). Las cuentas de usuarios son configuradas para que solo los usuarios específicos tengan acceso a la red a través del dominio de confianza. El uso de passwords seguros es una de las mejores formas de utilización exitosa del PPTP.

Básicamente, PPTP lo que hace es encapsular los paquetes del protocolo punto a punto PPP (Point to Point Protocol) que a su vez ya vienen encriptados en un paso previo para poder enviarlos a través de la red. El proceso de encriptación es gestionado por PPP y luego es recibido por PPTP; este último utiliza una conexión TCP llamada *conexión de control* para crear el túnel y una versión modificada de la Encapsulación de Enrutamiento Genérico (GRE, Generic Routing Encapsulation) para enviar los datos en formato de datagramas IP, que serían paquetes PPP encapsulados, desde el cliente hasta el servidor y viceversa. El proceso de autenticación de PPTP utiliza los mismos métodos que usa PPP al momento de establecer una conexión, como por ejemplo PAP (Password Authentication Protocol) y CHAP (Challenge Handshake Authentication Protocol).

Por otro lado, los datos enviados por el túnel PPTP en los dos sentidos son encriptados. Los paquetes de red son encriptados en la fuente (cliente o servidor), viajan a través del túnel, y son desencriptados en el destino. Como todos los datos en una conexión PPTP fluyen dentro del túnel, los datos son invisibles al resto del mundo. La encriptación de datos dentro del túnel da un nivel adicional de seguridad. El método de encriptación que usa PPTP es el *Microsoft Point to Point Encryption*, MPPE, y solo es posible su utilización cuando se emplea CHAP (o MS-CHAP en los NT) como medio de autenticación. MPPE trabaja con claves de encriptación de 40 o 128 bits, la clave de 40 bits es la que cumple con todos los estándares, en cambio la de 128 bits está diseñada para su uso en Norte América. Cliente y servidor deben emplear la misma codificación, si un servidor requiere de más seguridad de la que soporta el cliente, entonces el servidor rechaza la conexión. La opción de Filtrado de Paquetes PPTP incrementa el rendimiento y fiabilidad de la seguridad de red si está activada en el servidor PPTP. Cuando es así, el servidor acepta y enruta solo los paquetes PPTP de los usuarios autorizados. Esto previene que el resto de paquetes entren a la red privada y al servidor de PPTP.

1.1.2 L2TP - Layer 2 Tunneling Protocol

El protocolo de túneles L2TP, ha nacido de la combinación de las características del protocolo PPTP y L2F (Layer 2 Forwarding). L2TP es un protocolo de red que facilita la creación de túneles para enviar tramas PPP. Encapsula las tramas PPP para que puedan ser enviadas sobre redes IP, X.25, Frame Relay o ATM. La carga útil de las tramas PPP, puede ser encriptada y/o comprimida. Se puede usar L2TP directamente sobre diferentes tipos de WAN, por ejemplo, Frame Relay, sin una capa de transporte. L2TP usa UDP y una serie de mensajes de L2TP para el mantenimiento de túneles sobre redes IP. L2TP permite múltiples túneles entre los dos puntos

finales. L2TP está compuesto de dos partes: el concentrador de acceso L2TP (LAC) y el servidor de red L2TP (LNS). El LAC se sitúa entre un LNS y un sistema remoto, manda paquetes a cada uno de los dos. El LNS es el par del LAC, y es un punto de terminación lógica de una sesión PPP a la cual se le está aplicando el túnel desde el sistema remoto por el LAC. El L2TP soporta dos modos de túneles, el modo Obligatorio y el Voluntario.

L2TP usa el Protocolo de Control de Red (Network Control Protocol - NCP) para asignar la IP y autenticar en PPP, llamado comúnmente, PAP o CHAP. La seguridad en L2TP requiere que estén disponibles los servicios de encriptación, integridad y autenticación para todo el tráfico L2TP. Este transporte seguro opera en todo el paquete L2TP y es funcionalmente independiente de PPP y del protocolo que este transporta.

- *Túnel Obligatorio L2TP*

1. El usuario remoto inicializa una conexión PPP a un ISP.
2. El ISP acepta la conexión y el enlace PPP se establece.
3. El ISP solicita la autenticación parcial para saber el nombre de usuario.
4. El ISP mantiene una lista de todos los usuarios admitidos, para servir el final del túnel LNS.
5. El LAC inicializa el túnel L2TP al LNS.
6. Si el LNS acepta la conexión, el LAC encapsula el PPP con el L2TP, y entonces lo enviará a través del túnel.
7. El LNS acepta estas tramas, y las procesa como si fueran tramas PPP.
8. El LNS utiliza la autenticación PPP para validar al usuario y entonces asigna una dirección IP.

- *Túnel Voluntario L2TP*

1. El usuario remoto tiene una conexión a un ISP ya establecida.
2. El cliente L2TP (LAC), inicializa el túnel L2TP al LNS.
3. Si el LNS acepta la conexión, LAC encapsula con PPP y L2TP, y lo manda a través del túnel.
4. El LNS acepta estas tramas, y las procesa como si fueran tramas normales de entrada.
5. El LNS entonces usa la autenticación PPP para validar al usuario y asignarle una IP.

1.2 PROTOCOLOS DE CAPAS SUPERIORES

Tal y como se ha comentado en los puntos anteriores, la seguridad en la versión 4 del protocolo IP no fue contemplada en su diseño original, con lo que al querer introducir ampliaciones en las especificaciones IP se encontraron muchos problemas, entre ellos la gran cantidad de software que debía modificarse para adoptar esta ampliación debido al gran tamaño que ya tenía Internet. Además se tardó mucho tiempo en finalizar las nuevas especificaciones, con lo que al desarrollarse el comercio electrónico (e-Commerce), las empresas de venta por Internet, puesto que no podían modificar ninguna definición de los protocolos (IP, TCP, UDP), fueron desarrollando e imponiendo los suyos en los niveles que podían modificar, los correspondientes a la capa de aplicación (ver Figura 2).

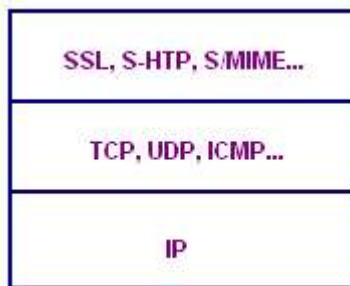


Figura 2 Ubicación de los protocolos en capas superiores

1.2.1 SSL (Secure Socket Layer)

El protocolo SSL fue diseñado originalmente por Netscape Development Corporation; la versión 3.0 fue diseñada con apoyo público y sugerencias de la industria con el siguiente objetivo: establecer una conexión segura (con criptografía) entre cliente y servidor y proveer privacidad y confidencialidad en la comunicación de dos aplicaciones. SSL está compuesto por dos capas. En la capa inferior, se encuentra el *protocolo SSL de registro*, que trabaja sobre algún protocolo de transporte (TCP y UDP por ejemplo); este protocolo se utiliza para encapsulamiento, encriptación, autenticación, servicios de secuencia y compresión. En la capa superior se encuentran 4 protocolos: el *protocolo SSL de inicio de comunicación entre dos entidades o handshake*, que negocia mecanismos de encriptación, autenticación, secuencia, compresión y establece los parámetros clave entre cliente y servidor. El *protocolo Change Cipher Spec*, que invoca cambios sincrónicos de mecanismos de seguridad y parámetros clave entre cliente y servidor. El *protocolo de datos de aplicación* para transportar los mensajes de aplicación entre los pares de cliente y servidor, y el *protocolo de Alerta*, que comunica mensajes de cierre y error de conexión.

SSL es un protocolo ampliamente utilizado que se basa en una arquitectura de tipo cliente/servidor y que permite una comunicación segura entre dos aplicaciones. Este protocolo permite la negociación de un algoritmo de cifrado y de las claves necesarias para asegurar un canal de seguridad (Channel Security) entre el cliente y el servidor; este canal tiene tres propiedades principalmente:

- Garantiza la *privacidad*. Después de negociar la clave privada todos los mensajes son cifrados.
- Garantiza la *autenticidad*. El servidor siempre se autentica mientras que los clientes pueden hacerlo o no.
- Garantiza la *fiabilidad*. Los mensajes incluyen una integridad proporcionada por el uso del sistema MAC.

SSL ha sido ampliamente utilizado, tanto en productos comerciales como de dominio público (Open SSL/mod SSL para apache por ejemplo) para el protocolo HTTP. Fue sometido el Internet Draft a la IETF y propuesto como estándar en 1996; la IETF redefinió su construcción y estableció TLS 1.0 como estándar, que corresponde a la versión 3.1 de SSL.

Para establecer una comunicación segura utilizando SSL se tienen que seguir una serie de pasos, como se describe a continuación:

- *Solicitud de SSL:*

Antes de que se establezca SSL, se debe hacer una solicitud. Típicamente esto implica un cliente haciendo una solicitud de un URL a un servidor que soporte SSL. SSL acepta solicitudes por un puerto diferente al utilizado normalmente para ese servicio. Una vez se ha hecho la solicitud, el cliente y el servidor empiezan a negociar la conexión SSL, es decir, hacen el *SSL Handshake*.

- *SSL Handshake:*

Durante el *handshake* se cumplen varios propósitos. Se hace autenticación del servidor y opcionalmente del cliente, se determina que algoritmos de criptografía serán utilizados y se genera una llave secreta para ser utilizada durante el intercambio de mensajes subsiguientes durante la comunicación SSL. Los pasos que se siguen son los siguientes:

1. **Client Hello:** El "saludo de cliente" tiene por objetivo informar al servidor que algoritmos de criptografía puede utilizar y solicita una verificación de la identidad del servidor. El cliente envía el conjunto de algoritmos de criptografía y compresión que soporta y un número aleatorio. El propósito del número aleatorio es para que en caso de que el servidor no posea un certificado para comprobar su identidad, aún se pueda establecer una comunicación segura utilizando un conjunto distinto de algoritmos. Dentro de los protocolos de criptografía hay un protocolo de intercambio de llave que define como cliente y servidor van a intercambiarla información, los algoritmos de llave secreta que definen que métodos pueden utilizar y un algoritmo de hash de una sola vía. Hasta ahora no se ha intercambiado información secreta, solo una lista de opciones.
2. **Server Hello:** El servidor responde enviando su identificador digital el cual incluye su llave pública, el conjunto de algoritmos criptográficos y de compresión y otro número aleatorio. La decisión de que algoritmos serán utilizados está basada en el más fuerte que tanto cliente como servidor soporten. En algunas situaciones el servidor también puede solicitar al cliente que se identifique solicitando un identificador digital.
3. **Aprobación del Cliente:** El cliente verifica la validez del identificador digital o certificado enviado por el servidor. Esto se lleva a cabo descriptando el certificado utilizando la llave pública del emisor y determinando si este proviene de una entidad certificadora de confianza. Después se hace una serie de verificaciones sobre el certificado, tales como fecha, URL del servidor, etc. Una vez se ha verificado la autenticidad de la identidad del servidor, el cliente genera una llave aleatoria y la encripta utilizando la llave pública del servidor y el algoritmo criptográfico y de compresión seleccionado anteriormente. Esta llave se le envía al servidor y en caso de que el *handshake* tenga éxito será utilizada en el envío de futuros mensajes durante la sesión.
4. **Verificación:** En este punto ambas partes conocen la llave secreta, el cliente por que la generó y el servidor por que le fue enviada utilizando su llave pública, siendo la única forma posible de descriptarla utilizando la llave privada del servidor. Se hace una última verificación para comprobar si la información transmitida hasta el momento no ha sido alterada. Ambas partes se envían una copia de las anteriores transacciones encriptada con la llave secreta. Si ambas partes confirman la validez de las transacciones, el *handshake* se completa, de otra forma se reinicia el proceso.

Ahora ambas partes están listas para intercambiar información de manera segura utilizando la llave secreta acordada y los algoritmos criptográficos y de compresión. El *handshake* se realiza solo una vez y se utiliza una llave secreta por sesión.

- *Intercambio de datos:*

Ahora que se ha establecido un canal de transmisión seguro SSL, es posible el intercambio de datos. Cuando el servidor o el cliente desea enviar un mensaje al otro, se genera un *digest* (resumen, utilizando un algoritmo de hash de una vía acordado durante el *handshake*), se encripta el mensaje y el *digest* se envían; cada mensaje es verificado utilizando el *digest*.

- *Terminación de una sesión SSL:*

Cuando el cliente deja una sesión SSL, generalmente la aplicación presenta un mensaje advirtiendo que la comunicación no es segura y confirma que el cliente efectivamente desea abandonar la sesión SSL.

De esta forma puede verse que este protocolo es bastante confiable y por lo tanto muy utilizado.

1.2.2 Protocolo Transport Layer Security - TLS

TLS es el estándar creado por la IETF como el protocolo de la capa de transporte. Surgió como respuesta a SSL de Netscape y PCT de Microsoft, ya que se consideró negativo para la industria el manejo de dos protocolos similares, por lo que se estableció TLS (RFC 2246). Está basado en SSL, de hecho se considera una actualización, versión 3.1 de SSL y presenta las siguientes modificaciones:

- Requiere soporte para el algoritmo DSA y DH, RSA es opcional.
- El algoritmo de generación de llaves está modificado; utiliza MD5 y SHA-1 con HMAC como función pseudo aleatoria, a diferencia del algoritmo de llaves MAC definido en SSL.
- Contiene un conjunto más completo de alertas.

TLS es la propuesta por el grupo de trabajo de la IETF, pero sin embargo, ha habido mayor desarrollo sobre SSL.

1.2.3 SSH (Secure SHell)

SSH es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de la red. Permite manejar por completo un equipo mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si se tiene un Servidor X arrancado. Además de la conexión a otras máquinas, SSH permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro de SSH.

SSH cumple la misma función que telnet o rlogin pero además, usando criptografía, logra brindar seguridad a los datos. SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión; a pesar de esto, es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre los destinos.

El protocolo SSH cuenta con dos versiones. La primera de ellas se mantiene por motivos de compatibilidad, pero se recomienda generalmente el uso de la segunda, por su mayor seguridad. OpenSSH es una implementación de cliente y servidor para estos protocolos; la versión disponible para Debian permite usar tanto SSH1 como SSH2. SSH es un protocolo para iniciar sesiones en máquinas remotas que ofrece autenticación, confidencialidad e integridad. Consta de tres componentes:

- Protocolo de transporte: Que normalmente opera sobre TCP/IP dando autenticidad, confidencialidad e integridad.
- Protocolo de autenticación de usuario: Que autentica al usuario ante el servidor.
- Protocolo de conexión: Que multiplexa un canal encriptado en diversos canales lógicos.

Este protocolo requiere que los servidores tengan "llaves", las cuales son usadas por los clientes cada vez que se conectan a un servidor para verificar que no fue suplantado; una llave es un número codificado y encriptado en un archivo. Para la encriptación de llaves, OpenSSH ofrece los algoritmos RSA y DSA.

A diferencia de telnet u otro servicio similar, SSH utiliza el puerto 22 para la comunicación y la forma de efectuar su trabajo es muy similar al efectuado por SSL. Para su uso se requiere que por parte del servidor exista un demonio que mantenga continuamente en el puerto 22 el servicio de comunicación segura, el sshd. El cliente debe ser un software que permita la hacer pedidos a este puerto 22 de forma cifrada.

La forma en que se entabla una comunicación en base a la misma para todos los protocolos seguros:

- El cliente envía una señal al servidor pidiéndole comunicación por el puerto 22.
- El servidor acepta la comunicación en el caso de poder mantenerla bajo encriptación mediante un algoritmo definido y le envía la llave pública al cliente para que pueda descifrar los mensajes.
- El cliente recibe la llave teniendo la posibilidad de guardarla la llave para futuras comunicaciones o destruirla después de la sesión actual.

Se recomienda que si se está en un computador propio, la clave sea guardada, en otro caso, manejarla cuidadosamente.

1.2.4 S-HTTP (Secure Hypertext Transfer Protocol)

S-HTTP (Secure HTTP) es una extensión del protocolo Hypertext Transfer Protocol –http, utilizado en el servicio WWW que proporciona seguridad en el intercambio de documentos multimedia. Proporciona servicios de confidencialidad, autenticidad, integridad y no repudio (poder demostrar a una tercera persona que la información recibida proviene realmente del emisor). Cada archivo S-HTTP es encriptado, contiene un certificado digital, o en ocasiones maneja las dos opciones; así mismo permite múltiples algoritmos

de cifrado (DES, IDEA y RC2) y de intercambio de claves (RSA, Kerberos, Out-band e Inband). Para algunos casos, en un muy buen complemento usar el Protocolo de seguridad SSL. La gran diferencia es que S-HTTP permite al cliente enviar un certificado para autenticar el origen de los datos, usando SSL, de forma que solo el usuario definido sea autenticado. Este protocolo es ampliamente utilizado en situaciones donde el servidor representa un banco y requiere autenticación del usuario, lo cual es más seguro que un *user id* y un *password*.

S-HTTP no utiliza un solo algoritmo de encriptación: soporta el sistema de criptografía de la infraestructura de llave pública; por otro lado, SSL trabaja en la capa de aplicación, por encima del protocolo TCP (Transmisión Control Protocol). Sin embargo, S-HTTP trabaja en la parte más alta de la capa de aplicación. Tanto SSL como S-HTTP pueden ser usados por un usuario cualquiera de un browser, pero solo uno de los dos puede ser usado con un documento dado. S-HTTP ha sido enviado al Internet Engineering Task Force (IETF) para ser considerado un estándar; el Request for Comments (RFC) Internet Draft 2660 describe al protocolo S-HTTP en detalle.

A continuación se presenta una descripción breve de las funciones adicionales: un cliente solicita un documento, le dice al servidor qué tipo de cifrado puede manejar y le dice también dónde puede encontrar su clave pública. Si el usuario con esa clave está autorizado a acceder al documento, el servidor responde cifrando el documento y enviándolo al cliente, que usará su clave secreta para descifrarlo y mostrárselo al usuario. Las negociaciones entre el cliente y el servidor tienen lugar intercambiando datos formateados. Estos datos incluyen una variedad de opciones de seguridad y algoritmos a utilizar. Las líneas usadas en las cabeceras incluyen:

- Dominios privados S-HTTP, que especifican la clase de algoritmos de cifrado así como la forma de encapsulamiento de los datos (PEM o PKCS7).
- Tipos de certificado S-HTTP, que especifica el formato de certificado aceptable, actualmente X.509.
- Algoritmos de intercambio de clave S-HTTP, que indica los algoritmos que se usarán para el intercambio de claves (RSA, fuera de banda, dentro de banda y Kerberos).
- Algoritmos de firmas S-HTTP, que especifican el algoritmo para la firma digital (RSA).
- Algoritmos de resumen de mensaje S-HTTP, que identifican el algoritmo para proporcionar la integridad de los datos usando funciones de hash (RSA-MD2, RSA-MD5).
- Algoritmos de contenido simétrico S-HTTP, que especifica el algoritmo simétrico de cifrado en bloque usado para cifrar los datos.
- Algoritmos de cabecera simétrica de S-HTTP, que proporciona una lista del cifrado de clave simétrica utilizada para cifrar las cabeceras.
- Mejoras de la intimidad de S-HTTP, que especifica las mejoras en la intimidad asociadas con los mensajes, como firmar, cifrar o autenticar. Uno de los métodos de cifrado disponible en S-HTTP es el popular PGP (Pretty Good Privacy), que es un paquete completo de seguridad para correo electrónico. Presta servicios de encriptación, autenticación, firmas digitales y compresión de datos. Todo el paquete se distribuye de forma gratuita, incluyendo el código fuente. Es posible conseguir PGP en Internet para varias plataformas incluidas Unix, Windows y MacOS. PGP utiliza algoritmos existentes de encriptación, en vez de crear unos

propios. Estos algoritmos son: RSA, IDEA y MD5. También soporta compresión de texto, utilizando el algoritmo ZIP. Para enviar un mensaje encriptado y firmado, ambas partes deben tener el software PGP e intercambiar sus llaves públicas.

1.2.5 S/MIME (Secure/Multipurpose INTERNET Mail Extensions)

S/MIME (Secure/Multipurpose Internet Mail Extensions) es una extensión del protocolo MIME (descrito en el RFC 1521), que añade las características de firma digital y cifrado. De esta forma, MIME se ha convertido en un formato oficial estándar propuesto para envío de correo electrónico a través de Internet. Los mensajes electrónicos constan de dos partes: una cabecera (header) donde se especifican todas las opciones importantes para la conexión, junto con el origen/destino (el formato del header se puede encontrar en detalle en el RFC 822); y el cuerpo del mensaje (body), que por lo general no tiene estructura a menos que el e-mail tenga formato MIME; éste formato permite a un e-mail incluir texto especial, gráficas, audio y mucho más de forma estandarizada. Sin embargo, MIME por sí solo no provee servicios de seguridad. El propósito de S-MIME es definir estos servicios, siguiendo la sintaxis dada en PKCS#7 para firmas digitales y cifrado.

Una sección del cuerpo del protocolo MIME porta un mensaje PKCS #7, el cual por sí mismo es el resultado de un proceso criptográfico en otra sección del cuerpo. La estandarización de S-MIME ha pasado a cargo de la IETF, y una serie de documentos que describen el protocolo han sido publicados por ellos.

1.2.6 POP3-S (Post Office Protocol)

POP (Protocolo de Oficina de Correos, Post Office Protocol) fue diseñado para la gestión, el acceso y la transferencia de mensajes de correo electrónico entre dos máquinas, habitualmente un servidor y una máquina de usuario; además simplemente permite listar mensajes, recibirlos y borrarlos. Existen muchos servidores POP disponibles para Linux; el original que viene con la mayoría de las distribuciones suele ser adecuado para la mayoría de los usuarios. Los problemas principales con POP son similares a los de muchos otros protocolos; los nombres de usuarios y sus contraseñas se transmiten en texto claro, haciendo de ello un buen objetivo para un sniffer de paquetes. POP se puede utilizar conjuntamente con SSL, sin embargo no todos los clientes de correo soportan POP seguro mediante SSL.

Los servidores POP3 permiten tener acceso a una sola bandeja de entrada a diferencia de los servidores IMAP (Internet Message Access Protocol), que proporcionan acceso a múltiples carpetas en los servidores. El puerto que utiliza es generalmente el 110.

1.2.7 IMAP4 (Internet Message Access Protocol)

Protocolo de Acceso a Mensajes de Internet. Similar al POP3, aunque ofrece prestaciones adicionales, como la búsqueda por palabras clave mientras los mensajes están en el servidor o la elección de los mensajes que se desean recuperar. Al igual que POP, IMAP utiliza SMTP (Simple Mail Transfer Protocol) para comunicaciones entre el cliente de correo y el servidor. Desde el punto de vista del

cliente, IMAP es ideal para usuarios que acceden a correspondencia desde diferentes equipos, porque pueden mantener todos los almacenes de mensajes en sincronía. Permite mantener con facilidad múltiples cuentas, permitir a múltiples personas acceso a una cuenta, dejar correo en el servidor, simplemente descargarlos encabezados, o los cuerpos sin attachments, etc. Los servidores POP e IMAP que traen la mayoría de las distribuciones (empaquetados en un único paquete llamado *imapd*) cubren la mayoría de las necesidades. Fundamentalmente, los dos protocolos les permiten a los clientes de correo acceder a los mensajes almacenados en un servidor de correo. Algunas de las características importantes que tiene IMAP y que carece POP3 incluyen:

- Soporte para los modos de operación *connected* y *disconnected*: Al utilizar POP3, los clientes se conectan al servidor de correo brevemente, solamente lo que les tome descargar los nuevos mensajes. Al utilizar IMAP4, los clientes permanecen conectados el tiempo que su interfaz permanezca activa y descargan los mensajes bajo demanda. El patrón de IMAP4 puede dar tiempos de respuesta más rápidos para usuarios que tienen una gran cantidad de mensajes.
- Soporte para la conexión de múltiples clientes simultáneos a un mismo destinatario: El protocolo POP3 asume que el cliente conectado es el único dueño de una cuenta de correo. En contraste, el protocolo IMAP4 permite accesos simultáneos a múltiples clientes y proporciona ciertos mecanismos a los clientes para que se detecten los cambios hechos a un mail box por otro cliente concurrentemente conectado.
- Soporte para acceso a partes MIME de los mensajes y obtención parcial: Casi todo el mail de Internet es transmitido en formato MIME. El protocolo IMAP4 le permite a los clientes obtener separadamente cualquier parte MIME individual así como, obtener porciones de las partes individuales o los mensajes completos.
- Soporte para que la información de estado del mensaje se mantenga en el servidor: A través de la utilización de *banderas* definidas en el protocolo IMAP4 de los clientes, se puede vigilar el estado del mensaje, por ejemplo, si el mensaje ha sido o no leído, respondido o eliminado. Estas *banderas* se almacenan en el servidor, de manera que varios clientes conectados al mismo correo en diferentes tiempos pueden detectar los cambios hechos por otros clientes.
- Soporte para acceder múltiples buzones de correo en el servidor: Los clientes de IMAP4 pueden crear, renombrar y/o eliminar correo (por lo general presentado como carpetas al usuario) del servidor, y mover mensajes entre cuentas de correo. El soporte para múltiples buzones de correo también le permite al servidor proporcionar acceso a los *fólder* públicos y compartidos.
- Soporte para búsquedas de parte del servidor: IMAP4 proporciona un mecanismo para que los clientes le pidan al servidor que busque mensajes de acuerdo a una cierta variedad de criterios. Este mecanismo evita que los clientes descarguen todos los mensajes de su buzón de correo con el fin de agilizar las búsquedas.

- Soporte para un mecanismo de extensión definido: Como reflejo de la experiencia en versiones anteriores de los protocolos de Internet, IMAP define un mecanismo explícito mediante el cual puede ser extendido.

Ya sea que se utilice POP3 o IMAP4 para obtener los mensajes, los clientes utilizan SMTP para enviar mensajes. Los clientes de correo electrónico son comúnmente denominados clientes *POP* ó *IMAP*, pero en ambos casos se utiliza SMTP. IMAP es utilizado frecuentemente en redes grandes; por ejemplo los sistemas de correo de un campus. IMAP les permite a los usuarios acceder a los nuevos mensajes instantáneamente en sus computadoras, ya que el correo está almacenado en la red. Con POP3 los usuarios tendrían que descargar el mail a sus computadoras o accederlo vía Web. Ambos métodos toman más tiempo de lo que le tomaría a IMAP, y se tiene que descargar el mail nuevo o refrescar la página para ver los nuevos mensajes. Se han propuesto muchas extensiones de IMAP4 que son de uso común, y el día de hoy es posible hablar de una versión segura, IMAP4S, donde se ofrece de manera contraria a otros protocolos de Internet, mecanismos nativos de cifrado ofreciendo al usuario autenticación y confidencialidad. La transmisión de contraseñas en texto plano también es soportada.

1.2.8 SMTP Seguro (Simple Mail Transfer Protocol)

SMTP (Protocolo Simple de Transferencia de Correo Electrónico) es un protocolo basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras y/o distintos dispositivos (PDA's, Celulares, etc). Para implementarlo se encuentran disponibles muchos paquetes, siendo el más viejo y el más utilizado el Sendmail, aunque tiene contendientes como Postfix y Gmail, que han sido implementados desde cero teniendo en cuenta la seguridad.

En 1982 se diseñó el primer sistema para intercambiar correos electrónicos para ARPANET, definido en dos Request for Comments: *RFC 821* y *RFC 822*. La primera de ellas define el protocolo y la segunda el formato del mensaje; con el tiempo se ha convertido en uno de los protocolos más usados en Internet. Para adaptarse a las nuevas necesidades surgidas del crecimiento y popularidad de Internet se han hecho varias ampliaciones a este protocolo, como poder enviar texto con formato o archivos adjuntos. SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. En el conjunto de protocolos TCP/IP, el SMTP va por encima de TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión. En primer lugar se ha de establecer una conexión entre el emisor (cliente) y el receptor (servidor). Esto puede hacerse automáticamente con un programa cliente de correo o mediante un cliente telnet. En el siguiente ejemplo se muestra una conexión típica. Se nombra con la letra C al cliente y con S al servidor.

```
S: 220 Servidor ESMTP
C: HELLO
S: 250 Hello, please to meet you
C: MAIL FROM: yo@midominio.com
S: 250 Ok
C: RCPT TO: destinatario@sudominio.com
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: Campo de asunto
```



```
C: From: yo@midominio.com
C: To: destinatario@sudominio.com
C:
C: Hola,
C: Esto es una prueba.
C: Adios.
C: .
S: 250 Ok: queued as 12345
C: quit
S: 221 Bye
```

En el ejemplo pueden verse los comandos básicos de SMTP:

- HELLO, para abrir una sesión con el servidor.
- MAIL FROM, para indicar quien envía el mensaje.
- RCPT TO, para indicar el destinatario del mensaje.
- DATA, para indicar el comienzo del mensaje, éste finalizará cuando haya una línea únicamente con un punto.
- QUIT, para cerrar la sesión.

Las respuestas que da el servidor pueden ser de varias clases:

- 2XX, para una respuesta afirmativa
- 3XX, para una respuesta temporal afirmativa
- 4XX, para una respuesta de error, pero se espera a que se repita la instrucción
- 5XX, para una respuesta de error.

Una vez que el servidor recibe el mensaje finalizado con un punto puede bien almacenarlo si es para un destinatario que pertenezca su dominio, o bien retransmitirlo a otro servidor para que finalmente llegue a un servidor del dominio del receptor. El mensaje está compuesto por dos partes:

- *Cabecera:* en ella se usan unas palabras clave para definir los campos del mensaje. Estos campos ayudan a los clientes de correo a organizarlos y mostrarlos. Los más típicos son subject (asunto), from (emisor) y to (receptor). Estos dos últimos campos no hay que confundirlos con los comandos MAIL FROM y RCPT TO, que pertenecen al protocolo, pero no al formato del mensaje.
- *Cuerpo del mensaje:* es el mensaje propiamente dicho. En el SMTP básico está compuesto únicamente por texto, y finalizado con una línea en la que el único carácter es un punto.

Una de las limitaciones del SMTP original es que no facilita métodos de autenticación a los emisores, así que se definió la extensión SMTP AUTH. A pesar de esto, el SPAM es aún el mayor problema y no se cree que las extensiones sean una forma práctica para prevenirlo. *Internet Mail 2000* es una de las propuestas para reemplazarlo.

1.2.9 SET (Secure Electronic Transaction)

Es un protocolo desarrollado por las empresas VISA y MASTERCARD para las transacciones electrónicas (e-Commerce). Soporta los protocolos DES y RSA para el intercambio de claves y el cifrado de datos, además proporciona servicios como Transmisiones confidenciales, Autenticación de los dos usuarios, Comprobación de la integridad en los pagos y las cantidades, Autenticación cruzada (del comerciante ante el usuario y del usuario al comerciante). SET utiliza la criptografía de clave pública, para garantizar la seguridad de las transacciones. Otro dispositivo de seguridad de SET consiste en el uso de firmas digitales, que certifican aún más la validez del mensaje. Para ello, SET emplea compendios (digesto resumen) de mensaje.

SET agrupa a las siguientes entidades en un solo sistema de pago:

- Tarjeta habiente: aquella persona poseedora de una tarjeta de crédito.
- Emisor: entidad financiera que emite la tarjeta.
- Comerciante: conocido en la literatura SET como el mercader, es la empresa que vende bienes o intercambia servicios por dinero.
- Adquirente: institución financiera que establece una cuenta con el Comerciante y procesa autorizaciones y pagos.
- Intermediario para pago: dispositivo operado por un adquirente o designado a un tercero para que procese los mensajes de pago, incluyendo instrucciones de pago de un tarjeta habiente.
- Marcas: Las instituciones financieras emiten tarjetas con marcas en ellas, para hacer publicidad a la marca y establecen ciertas reglas de uso y aceptación de sus tarjetas y proveen redes que las interconectan a las instituciones financieras.
- Terceros: los emisores y los adquirentes pueden asignar a terceros para el procesamiento de las transacciones.

Para poder hacer una transacción SET cada uno de los participantes debe estar registrado por una entidad certificadora, que como su nombre lo indica emite un certificado electrónico en el que hace constar la identidad de una entidad.

SET pretende masificar el uso de Internet como "el mayor centro comercial del mundo", pero para hacerlo SET tiene que lograr:

- Confidencialidad de la información.
- Integridad de los datos
- Autenticación de la cuenta del tarjeta habiente
- Autenticación del comerciante
- Interoperabilidad

A diferencia de una transacción o compra persona a persona, por teléfono o correo, donde la transacción la inicia el comerciante, en SET la transacción la inicia el tarjeta habiente. Una vez todos los participantes estén registrados ante una autoridad certificadora, pueden empezar a realizar transacciones seguras.

ANEXO D

1. IMPLEMENTACIÓN PRÁCTICA DE VLANs Y AUTENTICACIÓN

1.1 Implementación práctica de VLANs utilizando Switches Cisco

Como ya se mencionó, una VLAN es una red lógicamente segmentada por funciones, equipos de trabajo o aplicación, sin cambiar la localización física de los usuarios. Las VLANs tienen los mismos atributos que las LANs físicas, pero con la diferencia de que se pueden agrupar estaciones finales aún si no están físicamente localizadas en el mismo segmento LAN. Un puerto de un switch puede pertenecer a una VLAN y los paquetes, ya sean unicast, broadcast o multicast, serán reenviados sólo a las estaciones que pertenecen a esa VLAN. Cada VLAN se considera una red lógica, y los paquetes que están destinados a una estación que no hace parte de la VLAN deben ser reenviados a través de un Router o un switch que soporte enrutamiento, como se muestra en la figura 1. Como una VLAN es considerada una red lógica separada, esta contiene su propia base de datos, llamada Management Information Base (MIB).

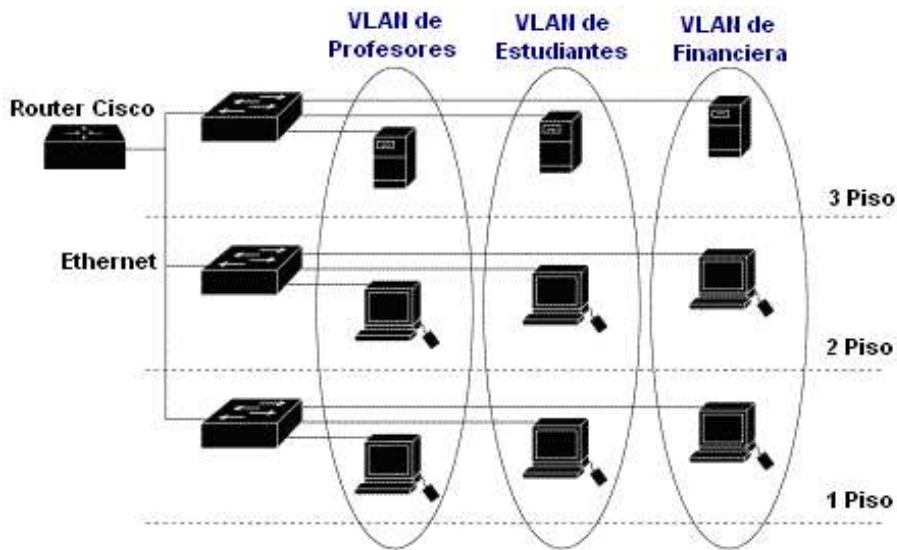


Figura 1. Ejemplo de separación en VLANs

Para el desarrollo de esta prueba se utilizó un switch Cisco Catalyst 2900 y un switch Cisco Catalyst 2950. Estos switches traen una configuración de fábrica, en la cual todos los puertos pertenecen a una única VLAN definida, la VLAN 1. Para comenzar, es necesario configurar dos nuevas VLANs, utilizando los siguientes comandos:

```
1. Switch# configure terminal  
2. Switch(config)# vlan 2  
3. Switch(config-vlan)# name Estudiantes  
4. Switch(config-vlan)# end
```

```
1. Switch# configure terminal  
2. Switch(config)# vlan 3  
3. Switch(config-vlan)# name Profesores  
4. Switch(config-vlan)# end
```

La interpretación de cada uno de los comandos es la siguiente:

1. Entrar al modo de configuración global.
2. Se introduce el ID (Identificador) de la VLAN para entrar al Modo de Configuración de esa VLAN. Se puede introducir un nuevo ID para crear una VLAN nueva, o se puede introducir un ID de una VLAN existente para modificarla.
3. Introduce un nombre para la VLAN; si no se especifica un nombre, el nombre por defecto corresponde al ID de la VLAN precedido de ceros, así: la VLAN0004, es el nombre por defecto para la VLAN4.
4. Retorna al modo privilegiado.

El comando `#show vlan`, permite ver la configuración de las VLANs existentes en el switch. El paso siguiente es asignar los puertos que se quiere, pertenezcan a una determinada VLAN; por ejemplo, para asignar el puerto 2 a la VLAN2, se utilizan los siguientes comandos:

```
1. Switch# configure terminal  
   Enter configuration commands, one per line. End with CNTL/Z.  
2. Switch(config)# interface gigabitethernet0/2  
3. Switch(config-if)# switchport mode access  
4. Switch(config-if)# switchport access vlan 2  
5. Switch(config-if)# end
```

La interpretación de cada uno de los comandos es la siguiente:

1. Entrar al modo de configuración global.
2. Entrar al modo de configuración de la interface que se quiere adherir a la VLAN.
3. Define el modo de calidad de miembro para el puerto.
4. Asigna el puerto a la VLAN2.
5. Retorna al modo privilegiado.

De esta forma se asignan los puertos que se desee a cada VLAN, y este proceso debe llevarse a cabo de igual forma en los switches. Una vez configurado el switch con diferentes VLANs y puertos asignados a cada una de ellas, es posible comprobar la segmentación del tráfico, ya que no es posible la comunicación entre dos puertos que pertenezcan a diferentes VLANs. Si se tiene la configuración que se muestra en la figura 2, no es posible realizar ping entre el Host 1 y el Host 2 ya que están conectados a puertos que pertenecen a diferentes VLANs: el puerto 2 pertenece a la VLAN2 y el puerto 8 pertenece a la VLAN3.

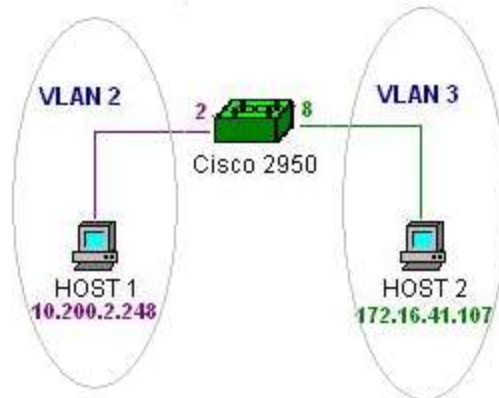


Figura 2. Dos host conectados a dos VLANs diferentes

Ahora es importante introducir el concepto de un Puerto Multi-VLAN; un puerto de un switch puede ser configurado para que pertenezca a varias VLANs y de esta forma permita que a través de él pase el tráfico de todas ellas cuando sea necesario; de esta forma es posible conectar varios switches y comunicar las VLANs con el mismo ID que estén configuradas en ellos, así: si se tienen 2 switches como en la figura 3, y en los dos se ha configurado la VLAN2, todos los host conectados a puertos que pertenezcan a la VLAN2 (Host 2 y Host 4) van a poder comunicarse, como si estuvieran físicamente en la misma red LAN; pero de igual forma habrá segmentación del tráfico, ya que host conectados a puertos que pertenecen a diferentes VLANs (Host 2 y Host 3) no van a poder comunicarse. En los switches Cisco este puerto es llamado *Puerto Trunk* y es un enlace punto a punto entre una o más interfaces Ethernet del switch y cualquier otro dispositivo, como un router u otro switch; gracias a los puertos Trunk es posible llevar el tráfico de múltiples VLANs sobre un solo enlace y de esta forma es posible extender las VLANs a través de toda la red; un puerto Trunk se configura como se muestra a continuación:

1. Switch# **configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
2. Switch(config)# **interface gigabitethernet0/5**
3. Switch(config-if)# **switchport mode trunk**
4. Switch(config-if)# **switchport trunk encapsulation dot1q**
5. Switch(config-if)# **switchport trunk allowed vlan all**
6. Switch(config-if)# **end**

La interpretación de cada uno de los comandos es la siguiente:

1. Entrar al modo de configuración global.
2. Entrar al modo de configuración de la interfaz que se quiere modificar.

3. Se configura la interfaz como un puerto Trunk de nivel 2 de forma permanente y hace la negociación para convertir el enlace en un enlace Trunk aú si la interfaz con la que se va a comunicarse es una interfaz Trunk.
4. Configura el puerto para soportar encapsulación ISL (Inter-Switch Link) o 802.1Q. Por defecto el switch viene configurado para negociar el tipo de encapsulación con el puerto que se desea comunicar. Cada extremo del enlace debe estar configurado con el mismo tipo de encapsulación.
5. Configura las VLANs cuyo tráfico será autorizado para ser transmitido por el puerto Trunk. En este caso se define que pueda transmitirse el tráfico de todas las VLANs.
6. Retorna al modo privilegiado.

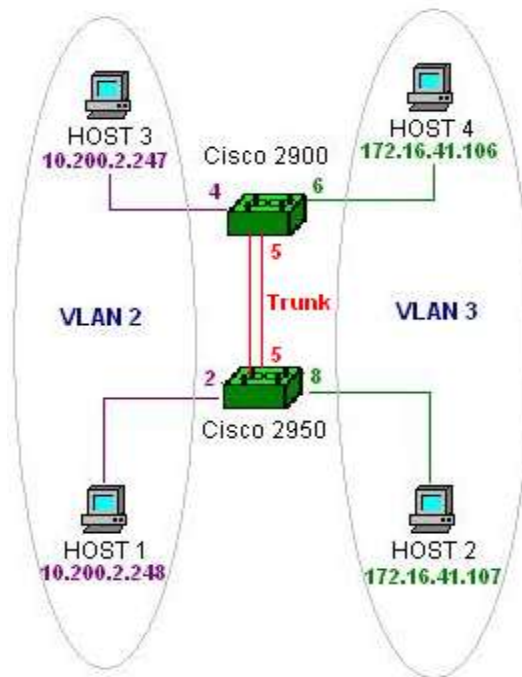


Figura 3. Dos Switches conectados por medio de un puerto Trunk

Teniendo esta configuración, no existe ningún problema si a uno de los Host se le cambia la dirección IP para llevar a cabo un ataque. Por ejemplo, el Host 1 podría fácilmente colocarse una dirección IP de la VLAN 3 y de esta forma ocultar su identidad. Además como ya se mencionó, teniendo esta configuración no es posible que haya comunicación entre VLANs, lo cual es importante debido al concepto de interconectividad que caracteriza a las redes; obligatoriamente habrá VLANs que necesitarán comunicarse con otras VLANs. En este caso, es necesaria la utilización de un Router, como se muestra en la figura 4:

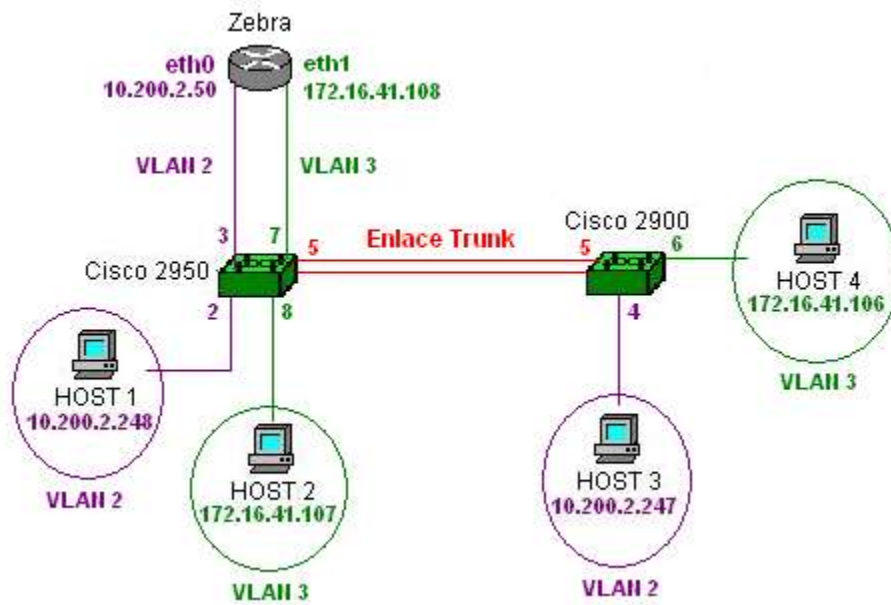


Figura 4. Comunicación entre VLANs utilizando un Router

En este caso, se tiene un Router por Software utilizando Zebra, al cual se le asignaron las direcciones 10.200.2.50 para la interfaz `eth0` y 172.16.41.108 para la interfaz `eth1`. De igual forma que para el ejemplo anterior, se tienen los dos switches en cada uno de los cuales se han configurado la VLAN 2 y la VLAN 3; así, es posible tener comunicación entre un host conectado a un puerto perteneciente a la VLAN 2 y un host conectado a un puerto perteneciente a la VLAN 3, ya que el Router se encarga de encaminar los paquetes hacia el destino deseado. La configuración de las direcciones IP del Host 2 y 3 se muestra a continuación:

Dirección IP: 172.16.41.107
Máscara de Subred: 255.255.0.0
Puerta de Enlace Predeterminada: 172.16.41.108

Dirección IP: 10.200.2.247
Máscara de Subred: 255.255.255.0
Puerta de Enlace Predeterminada: 10.200.2.50

Como se puede observar, los Hosts pertenecientes a la VLAN 2 tendrán como puerta de enlace la interfaz `eth0` del Router: 10.200.2.50, y los Hosts pertenecientes a la VLAN 3, tendrán como puerta de enlace la interfaz `eth1` del Router: 172.16.41.108. Así, es imposible cambiar la dirección IP a un Host de la VLAN 2 por una dirección de la VLAN 3 para llevar a cabo un ataque, ya que habría un error de configuración y el Router no encaminaría los paquetes.

Con estas configuraciones, es posible entender la importancia de crear VLANs, ya que además de permitir segmentar una red LAN física para proteger determinada información crítica, son una buena solución contra ataques por suplantación.

1.2 Implementación Práctica de Autenticación en Windows 2000 Server y XP utilizando un Servidor RADIUS

Para Implementar una práctica se utilizó un cliente con Microsoft Windows 2000 (Service Pack 4), un switch Cisco Catalyst 2950 y un servidor de autenticación soportado sobre Fedora Core 4 y FreeRadius.

Para habilitar la autenticación por 802.1X en el cliente Windows se debe ir a *Inicio/Programas/Herramientas Administrativas/Administración de equipos/* y en el panel izquierdo escoger la opción de *Servicios*. En el Panel derecho escoger el servicio de *Configuración Inalámbrica*, cambiar el tipo de inicio a automático e iniciar el servicio como lo muestra la figura 5. Este procedimiento dará la opción para habilitar 802.1X en las interfaces de red del equipo ya sean cableadas o inalámbricas. Para habilitar la autenticación mediante 802.1X se abre la ventana de *Propiedades* de la interfaz de red que se va a configurar; en la pestaña *Autenticación*, se habilita el *Control de Acceso* y se escoge el tipo de EAP que se utilizará, en este caso PEAP (EAP Protegido) como lo muestra la figura 6.

Como configuración adicional, se puede hacer click en el botón *Propiedades* para decirle a Windows que no utilice certificados y que no utilice el usuario y contraseña de inicio de sesión de Windows para validarse en el switch, como se muestra en la figura 7.

Para habilitar la autenticación 802.1X en el puerto deseado del switch Cisco Catalyst 2950, se realizó la siguiente configuración por medio de la interfaz de consola, utilizando los siguientes comandos:

```
1. Switch# configure terminal
2. Switch(config)# aaa new-model
3. Switch(config)# aaa authentication dot1x default group radius
4. Switch(config)# dot1x system-auth-control
5. Switch(config)# interface gigabitethernet0/3
6. Switch(config)# switchport mode access
7. Switch(config-if)# dot1x port-control auto
8. Switch(config-if)# end
9. Switch(config)# radius-server host 172.120.39.46 auth-port 1812 key cisco
```

La interpretación de cada uno de los comandos es la siguiente:

1. Entrar al modo de configuración global.
2. Habilitar los nuevos comandos de configuración de AAA (Autenticación, Autorización y Cuentas).
3. Crea una lista de métodos de autenticación para 802.1X. Usando la lista de servidores RADIUS para autenticar como método por defecto.
4. Habilita globalmente la autenticación 802.1X en el switch.
5. Entra al modo de configuración de interfaces, especificando que se va a configurar la interfaz gigabit Ethernet 3 del panel 0.

6. Se coloca la interfase en modo *access*.
7. Establece autenticación 802.1X en el puerto o interfase actual.
8. Retorna al modo privilegiado.
9. Configura los parámetros del servidor RADIUS, como dirección IP, puerto UDP y palabra clave que debe coincidir con la palabra clave en la configuración del servidor RADIUS.

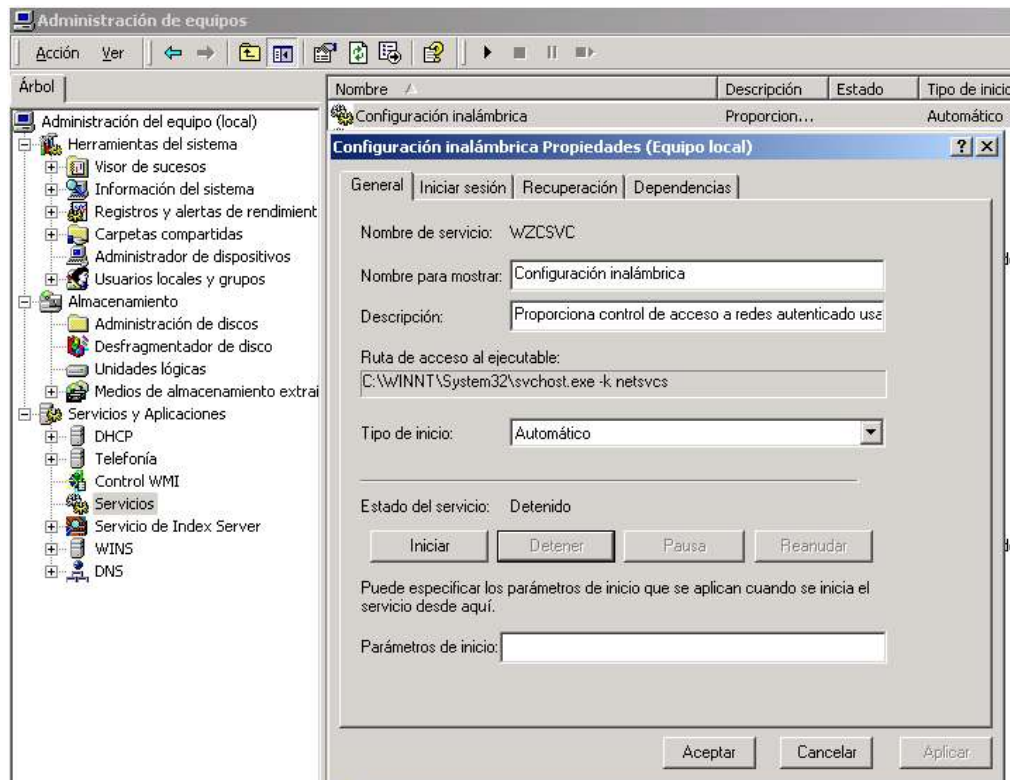


Figura 5. *Habilitando las opciones de Autenticación en Windows 2000 Server*

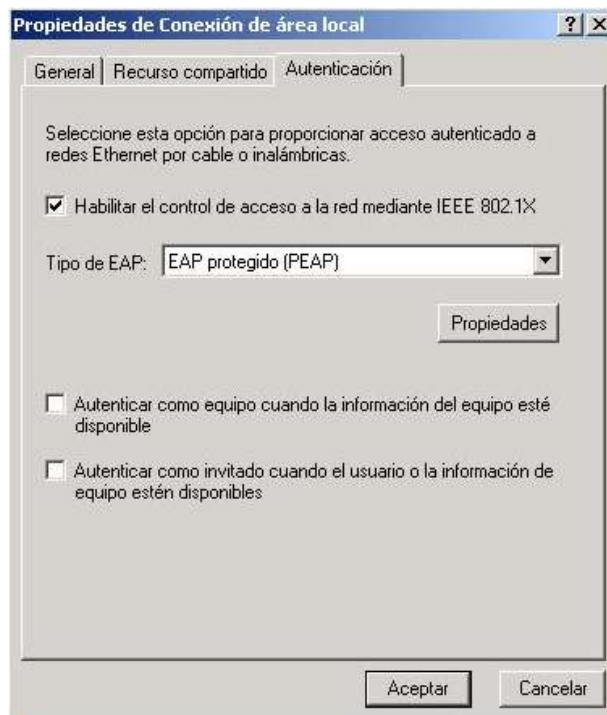


Figura 6. *Habilitando autenticación 802.1x en un equipo Windows 2000 Server*

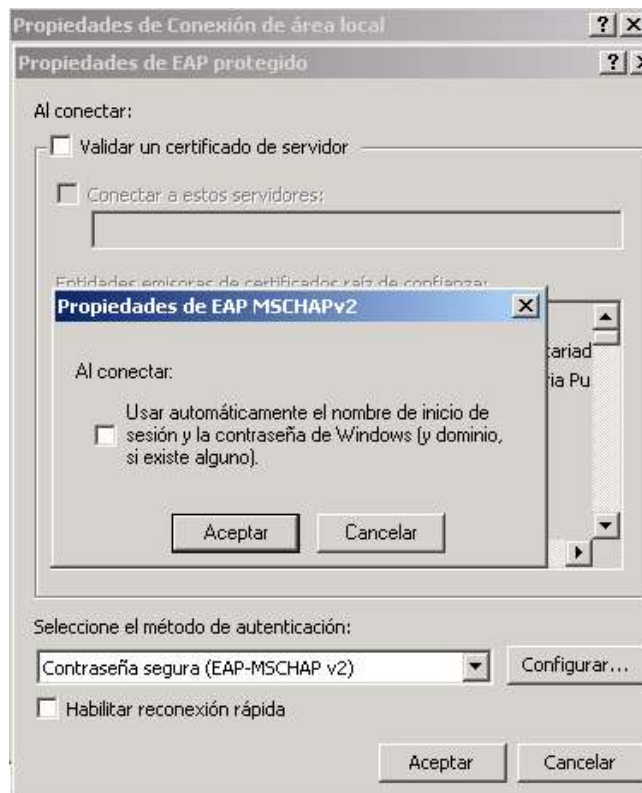


Figura 7. *Configuración de otras propiedades de 802.1x*

La última fase de la práctica es la instalación y configuración del servidor RADIUS. Para ello se utilizó el paquete *Freeradius*, disponible en www.freeradius.org. Una vez descargada la última versión del paquete se descomprime y se siguen los tres pasos de configuración, compilación e instalación de la mayoría de los paquetes para Linux:

```
#!/configure  
#make  
#make install
```

Una vez instalado satisfactoriamente *Freeradius* se debe dirigir al directorio `/usr/local/etc/radbd` para la correspondiente configuración.

El primer archivo a modificar es el *clients.conf*; este archivo contiene los clientes autorizados para hacer solicitudes al servidor RADIUS identificados por un nombre y una palabra clave. Teniendo en cuenta que el dispositivo que va a realizar dichas solicitudes es el switch el cual tiene por dirección IP 10.200.2.244 y la palabra clave que se le configuró para acceder al servidor RADIUS es *cisco*, al archivo *clients.conf* se le deben agregar las siguientes líneas.

```
client 10.200.2.244 {  
    secret = cisco  
    shortname = cisco  
}
```

El siguiente archivo que se debe modificar es el archivo *users*; este archivo contiene la información de configuración para los procesos de autenticación y seguridad de cada usuario. Para la práctica realizada se utilizó autenticación en el sistema, de manera que un usuario que se desee autenticar en el servidor RADIUS, debe ser un usuario válido del sistema operativo. Para este caso se creó el usuario *jparra*.

Para configurar autenticación en el sistema, se debe verificar que exista la siguiente línea en el archivo *users*:

```
DEFAULTAuth-Type= System
```

Además debe agregarse el usuario con su correspondiente contraseña al final del archivo, con el siguiente formato:

```
"jparra"User-Password=="jparra100"
```

En el archivo *eap.conf*, verificar el estado de la siguiente línea:

```
default_eap_type= peap
```

Esto le dice a RADIUS que el método de autenticación que usarán los clientes será PEAP, tal y como se configuró el cliente Windows para autenticarse sobre el switch. También, sobre el mismo archivo verificar que la configuración de PEAP, sea la siguiente:

```
peap {  
    default_eap_type = mschapv2  
}
```

Por último, el archivo *radiusd.conf* almacena la configuración general del servidor RADIUS, como puerto UDP, entre otros aspectos. Se debe verificar que incluya la configuración de los archivos anteriores de la siguiente manera:

```
$INCLUDE ${confdir}/clients.conf  
$INCLUDE ${confdir}/users  
$INCLUDE ${confdir}/eap.conf
```

Además comprobar que en las secciones de autorización y autenticación se encuentren los siguientes ítems respectivamente:

```
authorize {  
    preprocess  
    chap  
    mschap  
    suffix  
    eap  
    files  
}  
  
authenticate {  
    Auth-Type PAP {  
        pap  
    }  
    Auth-Type CHAP {  
        chap  
    }  
    Auth-Type MS-CHAP {  
        mschap  
    }  
    unix  
    eap  
}
```

En este punto solo resta iniciar el servicio de RADIUS; se puede llevar a cabo de dos maneras, como servicio (*/etc/init.d/freeradius start*), o en modo depuración (*radiusd -x*). El modo depuración muestra en pantalla las opciones con las que el servidor RADIUS inicia y queda esperando solicitudes de clientes. Cuando llega la solicitud de un cliente, en este caso el switch previa solicitud del cliente Windows, se observa una salida como en la figura 8.

En la figura se observa un requerimiento de acceso desde la dirección 10.200.2.144 para el usuario *jparra* y luego una respuesta de acceso aceptado, autorizando el acceso a la red al cliente Windows por el puerto del switch que está configurado con 802.1X.

```
rad_recv: Access-Request packet from host 10.200.2.244:1812, id=10, length=147
  NAS-IP-Address = 10.200.2.244
  NAS-Port-Type = Async
  User-Name = "jparra"
  Service-Type = Framed-User
  Framed-MTU = 1500
  Calling-Station-Id = "00-b0-d0-c2-cc-2d"
  State = 0x1884eec542d535d921c53fbc7b5f2009
  EAP-Message = 0x020900261900170301001b25cc2af4939bec28ba62a0dd854337c261da45292c1991646cd2e2
  Message-Authenticator = 0x70957ca6eee72ac857b2d968dd726f9c
Sending Access-Accept of id 10 to 10.200.2.244:1812
  Framed-IP-Address = 255.255.255.254
  Framed-MTU = 576
  Service-Type = Framed-User
  MS-MPPE-Recv-Key = 0x0cb7d68fe941cfb35b3ef2eba31e46f2a59096c962209e171d050fae82664b7d
  MS-MPPE-Send-Key = 0x69f2d867246a2f88ec39204b5f8917c20ea852cb9f55c3bfb98e4542c1f93ecd
  EAP-Message = 0x03090004
  Message-Authenticator = 0x00000000000000000000000000000000
  User-Name = "jparra"
```

Figura 8. Requerimiento de Acceso desde el Switch

En la red de datos de la Universidad del Cauca se utiliza el servicio RADIUS para autenticar a los usuarios que se conectan vía telefónica a la red universitaria. Este a su vez utiliza el servicio de directorio LDAP (*Lightweight Directory Access Protocol* - protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red) para acceder a información de nombre de usuario y contraseña de cada usuario. Por lo cual 802.1X + RADIUS + LDAP, se convierte en un esquema de seguridad muy robusto para implementar en la red de datos. Ya que un usuario no autorizado, no podría tener acceso a la red hasta que no se autentique satisfactoriamente. El usuario usaría el mismo nombre de usuario y contraseña que usa para acceder a otros servicios de la red como correo electrónico, servicio de shell, acceso remoto, entre otros.

Otro aspecto que hace atractivo este esquema es que, se permite la asignación dinámica de VLANs dependiendo del lugar donde se conecte el usuario, ya que cada usuario puede tener asignada una VLAN (por ejemplo VLAN de estudiantes, de profesores, administrativos, etc), la cual se configurará en el proceso de autorización, de manera que el puerto del switch se asigna a la VLAN que reciba de parte del servidor RADIUS según la categoría del usuario que se está autenticando en la red. Esto es muy útil, ya que da movilidad a los usuarios de la VLANs dado que no importa desde el lugar que se conecten, siempre van a estar en la VLAN que les corresponde.

ANEXO E

1. OTROS MECANISMOS DE SEGURIDAD EN REDES IP

1.1 VLANS

1.1.1 Clases de VLANs

Las VLAN pueden clasificarse de varias formas: Implícitas y Explícitas, Estáticas y Dinámicas. Las *implícitas* no necesitan cambios en la trama, pues de la misma forma que reciben información la procesan; ejemplo de ello son las VLAN basadas en puertos. En esta clase de VLAN el usuario no modifica ni manipula la trama, ya que solo posee una marca y por lo tanto el sistema se vuelve propietario.

Las VLAN *explícitas* si requieren modificaciones, adiciones y cambios (MAC) a la trama, por lo que se crearon los estándares 802.1p y 802.1q, en donde se colocan ciertas etiquetas o banderas en la trama para manipularla. Esta clase surge ante la necesidad de interoperar en un ambiente con diferentes marcas, pero basadas en estándares.

Un problema actual de las VLAN implícitas es que aun son propietarias y las explícitas son abiertas. Ambas clases de VLAN deberá utilizar los métodos de Networking, Inter-Domain Inter-VLAN para realizar sus funciones de forma más simple. Otro de los problemas de las VLAN es la Calidad de Servicios (QoS - Quality of service), porque ahora se busca que las redes puedan proporcionar QoS, para que dentro de las VLAN el usuario pueda indicar la prioridad de sus paquetes y de esta forma hacer uso eficiente del ancho de banda.

Las VLAN Estáticas son puertos en un switch que se asignan estáticamente a una VLAN. Estos puertos mantienen sus configuraciones de VLAN asignadas hasta que se cambien; aunque las VLAN estáticas requieren que el administrador haga los cambios, este tipo de red es segura, de fácil configuración y monitoreo. Las VLAN estáticas funcionan bien en las redes en las que el movimiento se encuentra controlado y administrado (Ver Figura 1).



Figura 1. Ejemplo de LAN Estática

Las VLAN Dinámicas son puertos del switch que pueden determinar automáticamente sus tareas VLAN. Las VLAN dinámicas se basan en direcciones MAC, direccionamiento lógico o tipo de protocolo de los paquetes de datos (Ver Figura 2).



Figura 2. Ejemplo de VLAN Dinámica

1.1.2 Las generaciones de VLANs

- Basadas en puertos y direcciones MAC.
- Internetworking; se apoyan en protocolo y dirección de paquetes.
- De aplicación y servicios: aquí se encuentran los grupos multicast y las VLAN definidas por el usuario.
- Servicios avanzados: ya se cumple con los tres criterios antes de realizar alguna asignación a la VLAN; se puede efectuar por medio de DHCP (Dynamic Host Configuration Protocol; Protocolo de configuración dinámica) o por AVLAN (Authenticate Virtual Local Area Networks; Redes virtuales autenticadas de área local).

1.1.2.1 *VLAN por Puerto*

Este tipo es el más sencillo ya que un grupo de puertos forma una VLAN (un puerto solo puede pertenecer a una VLAN); el problema se presenta cuando se quieren hacer MAC ya que la tarea es compleja. Aquí el puerto del switch pertenece a una VLAN, por tanto, si hay un servidor conectado a un puerto y este pertenece a una VLAN1, el servidor estará en la VLAN1.

1.1.2.2 *VLAN por MAC*

Se basa en MAC Address, por lo que se realiza un mapeo para que el usuario pertenezca a una determinada VLAN. Obviamente dependerá de la política de creación. Este tipo de VLAN ofrece mayores ventajas, pero es complejo porque hay que meterse con las direcciones MAC y si no se cuenta con un software que las administre, será muy laborioso configurar cada una de ellas.

1.1.2.3 *VLAN por Protocolo*

Lo que pertenezca a IP se enrutará a la VLAN de IP e IPX se dirigirá a la VLAN de IPX, es decir, se tendrá una VLAN por protocolo. Las ventajas que se obtienen con este tipo de VLAN radican en que dependiendo del protocolo que use cada usuario, este se conectará automáticamente a la VLAN correspondiente.

1.1.2.4 *VLAN por subredes de IP o IPX*

Aparte de la división que ejecuta la VLAN por protocolo, existe otra subdivisión dentro de este para que el usuario – aunque esté conectado a la VLAN del protocolo IP – sea asignado en otra VLAN (subred) que pertenecerá al grupo 10 o 20 dentro del protocolo.

1.1.2.5 *VLAN por direcciones IP multicast*

Generalmente son las direcciones de clase D las que ayudan a formular la VLAN. El mapeo o la asignación a la VLAN se basa o referencia en la dirección multicast.

1.1.2.6 *VLAN definidas por el usuario*

En esta política de VLAN se puede generar un patrón de bits, para cuando llegue la trama. Si los primeros cuatro bits son 1010 se irán a la VLAN de ingeniería, sin importar las características del usuario (protocolo, dirección MAC y puerto). Si el usuario manifiesta otro patrón de bits, entonces se trasladará a la VLAN que le corresponda; aquí el usuario define las VLAN.

1.1.2.7 VLAN Binding

Se conjugan tres parámetros o criterios para la asignación de VLAN: si el usuario es del puerto x, entonces se le asigna una VLAN correspondiente. También puede ser puerto, protocolo y dirección MAC, pero lo importante es cubrir los tres requisitos previamente establecidos, ya que cuando se cumplen estas tres condiciones se coloca al usuario en la VLAN asignada, pero si alguno de ellos no coincide, entonces se rechaza la entrada o se manda a otra VLAN.

1.1.2.8 VLAN por DHCP

Aquí ya no es necesario proporcionar una dirección IP, sino que cuando el usuario enciende la computadora automáticamente el DHCP pregunta al servidor para que tome la dirección IP y con base en esta acción asigna al usuario la VLAN correspondiente. Esta política de VLANs es de las últimas generaciones.

1.1.3 Aplicaciones y productos

- *Movilidad:* Como ya se ha visto, el punto fundamental de las redes virtuales es el permitir la movilidad física de los usuarios dentro de los grupos de trabajo.
- *Dominios lógicos:* Los grupos de trabajo pueden definirse a través de uno o varios segmentos físicos, o en otras palabras, los grupos de trabajo son independientes de sus conexiones físicas, ya que están constituidos como dominios lógicos.
- *Control y conservación del ancho de banda:* Las redes virtuales pueden restringir los broadcasts a los dominios lógicos donde han sido generados. Además, añadir usuarios a un determinado dominio o grupo de trabajo no reduce el ancho de banda disponible para el mismo, ni para otros.
- *Conectividad:* Los modelos con funciones de routing permiten interconectar diferentes conmutadores y expandir las redes virtuales a través de ellos, incluso aunque estén situados en lugares geográficos diversos.
- *Seguridad:* Los accesos desde y hacia los dominios lógicos, pueden ser restringidos, en función de las necesidades específicas de cada red, proporcionando un alto grado de seguridad. Una de las características más importantes.
- *Protección de la inversión:* Las capacidades VLAN están, por lo general, incluidas en el precio de los conmutadores que las ofrecen, y su uso no requiere cambios en la estructura de la red o cableado, sino más bien los evitan, facilitando las reconfiguraciones de la red sin costes adicionales.

El primer suministrador de conmutadores con soporte VLAN fue ALANTEC (familia de concentradores/conmutadores multimedia inteligentes PowerHub), pero actualmente son muchos los fabricantes que ofrecen equipos con soluciones VLAN: Bytex (concentrador inteligente 7700), Cabletron (ESX-MIM), Chipcom (OnLine), Lannet (MultiNet Hub), Synoptics (Lattis System 5000), UB (Hub Access/One) y 3Com (LinkBuilder).

1.2 SISTEMAS DE DETECCIÓN DE INTRUSOS BAJO GNU/LINUX

Bajo GNU/Linux corre un amplio abanico de soluciones para detectar intrusos, desde las más sencillas hasta las más complejas. Se puede decir que un simple *find*, ejecutado buscando cambios aparentes en los archivos que tienen de *setuid root* en el sistema y comparándolos con otros anteriores, puede ser un sistema de detección de intrusos donde aparecen casi todos sus componentes: recabamos datos del sistema, la regla y el filtrado es inherente al *find* y a la condición que se ha impuesto sobre el y el sistema de notificación el que quiera darse con un pequeño script. Esto es fruto de la flexibilidad de un sistema Unix. Sin embargo, se tratará a continuación una de las soluciones específicas diseñadas a tal efecto para servir como sistema de detección de intrusos general en su ámbito concreto: SNORT.

1.2.1 SNORT

SNORT es un sistema de detección de intrusos en tiempo real y basado en red muy potente. Este sistema sigue el planteamiento de colocar una máquina con una interfaz en modo promiscuo que monitorice el tráfico que circula por la red; de este modo SNORT busca patrones que hagan presagiar que se está desencadenando un ataque sobre la red que este monitoriza. Al igual que *logcheck* y siguiendo la arquitectura general, el sistema incorpora paquetes de reglas para realizar chequeos determinados sobre el tráfico de red, en este caso categorizados en diversos y numerosos grupos como *smtp.rules*, *ddos.rules*, etcétera.

Otra de las grandes virtudes de SNORT es que incorpora un sistema bastante sencillo para escribir las reglas, de modo que se puede adaptar a los requerimientos reescribiendo las reglas para los incidentes que se desean monitorizar. Con este lenguaje se permite introducir no solo el protocolo o el puerto al que va destinado el paquete, también se puede indicar el contenido de este, flags determinados de los protocolos, etcétera, de modo que hace el programa extremadamente flexible.

Snort implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos, aprovechar alguna vulnerabilidad, análisis de protocolos, etc., todo esto en tiempo real. Snort¹ está disponible bajo licencia GPL, gratuito, disponible para plataformas UNIX/Linux y plataformas Windows. Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos,

¹ Snort, disponible en <http://www.snort.org/>

así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad.

Este IDS implementa un lenguaje de creación de reglas flexibles, potente y sencillo. Durante su instalación provee de cientos de filtros o reglas para backdoor, ddos, finger, ftp, ataques Web, CGI, escaneos Nmap. Puede funcionar como sniffer (es posible ver en consola y en tiempo real qué ocurre en la red y con todo el tráfico), registro de paquetes (permite guardar en un archivo los logs para su posterior análisis offline) o como un IDS normal (en este caso NIDS). La colocación de Snort en la red puede realizarse según el tráfico que se quiere vigilar: paquetes que entran, paquetes salientes, dentro del Firewall, fuera del Firewall, y en realidad prácticamente donde se quiera (Figura 3).

Una característica muy importante e implementada desde hace pocas versiones es FlexResp; permite, dada una conexión que emita tráfico malicioso, darla de baja, hacerle un DROP mediante el envío de un paquete con el flag RST activa, con lo cual cumpliría funciones de Firewall, cortando las conexiones que cumplan ciertas reglas predefinidas. No sólo corta las conexiones ya que puede realizar otras muchas acciones. Se verá más adelante su funcionamiento y ejemplos.

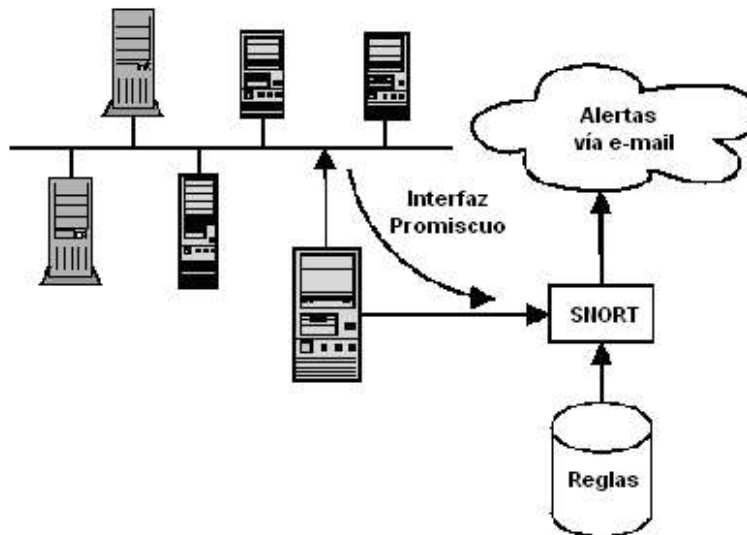


Figura 3. Esquema de funcionamiento de SNORT

Snort es en realidad un sniffer capaz de actuar como sistema de detección de intrusos en redes de tráfico moderado; su facilidad de configuración, su adaptabilidad, sus requerimientos mínimos (funciona en diferentes plataformas) y sobre todo su precio (se trata de un software completamente gratuito) lo convierten en una óptima elección en multitud de entornos, frente a otros sistemas como NFR (Network Flight Recorder) o ISS RealSecure que, aunque quizás sean más potentes, son también mucho más pesados e infinitamente más costosos. Para instalar un sistema de detección de intrusos basado en Snort, en primer lugar se necesita evidentemente este programa, que se puede descargar desde su página Web. Además, para compilarlo correctamente es necesario disponer de las

librerías libpcap, una interfaz para tratamiento de paquetes de red desde espacio de usuario, y es recomendable también (aunque no obligatorio) instalar Libnet, librería para la construcción y el manejo de paquetes de red. Con este software correctamente instalado en el sistema, la compilación de Snort es trivial. Teniendo en cuenta la clasificación de IDSs que se presentó anteriormente, se puede clasificar a Snort como un sistema basado en red (se monitoriza todo un dominio de colisión) y que funciona mediante detección de usos indebidos. Estos usos indebidos se reflejan en una base de datos formada por patrones de ataques; dicha base de datos se puede descargar también desde la propia página Web de Snort, donde además se pueden generar bases de patrones a medida de diferentes entornos (por ejemplo, ataques contra servidores Web, intentos de negaciones de servicio, exploits, etc.). El archivo que se utilice en el entorno será la base para el correcto funcionamiento del Sistema de Detección de Intrusos.

Una vez se ha compilado e instalado correctamente el programa llega el momento de ponerlo en funcionamiento; y es aquí donde se produce uno de los errores más graves en la detección de intrusos. Por lógica, uno tiende a pensar que el sensor proporcionaría mejores resultados cuantos más patrones de ataques contenga en su base de datos; nada más lejos de la realidad. En primer lugar, es muy probable que no todos los ataques que Snort es capaz de detectar sean susceptibles de producirse en el segmento de red monitorizado; si se sitúa el sensor en una zona desmilitarizada donde únicamente se ofrece servicio de Web, lo lógico es que las políticas implementadas en el cortafuegos ni siquiera dejen pasar tráfico hacia puertos que no sean los de los servidores Web pero, incluso en caso de que el potencial ataque se produjera entre máquinas del propio segmento, se debe evaluar con mucho cuidado si realmente vale la pena sobrecargar la base de datos con patrones que permitan detectar estos ataques. Evidentemente, si el sensor ha de analizar todo el tráfico, quizás mientras trata de decidir si un paquete entre dos máquinas protegidas se adapta a un patrón, se están dejando pasar tramas provenientes del exterior que realmente representan ataques: se debe tener presente que el sniffer no detendría el tráfico que no sea capaz de analizar para hacerlo más tarde, sino que simplemente lo dejaría pasar. Así, se deben introducir en la base de patrones de ataques los justos para detectar actividades sospechosas contra la red.

En segundo lugar, pero no menos importante, es necesario estudiar los patrones de tráfico que circulan por el segmento donde el sensor escucha para detectar falsos positivos y, o bien reconfigurar la base de datos, o bien eliminar los patrones que generan esas falsas alarmas. Aunque suene algo crudo, si un patrón genera un número considerable de falsos positivos, se debe plantear su eliminación: simplemente no se puede decidir si se trata de verdadero o de falsas alarmas. Esto es especialmente crítico si se lanzan respuestas automáticas contra las direcciones atacantes (por ejemplo, detener todo su tráfico en el Firewall); volviendo al ejemplo de la zona desmilitarizada con servidores Web, se puede llegar al extremo de detener a simples visitantes de una página, simplemente porque han generado falsos positivos; aunque en un entorno de alta seguridad quizás vale la pena detener muchas acciones no dañinas con tal de bloquear también algunos ataques (aunque constituiría una negación de servicio en toda regla contra los usuarios que hacen uso legítimo de un sistema), en un entorno normal de producción esto es impensable. Seguramente sería más provechoso detectar y detener estos ataques por otros mecanismos ajenos al sensor.

Las hazañas de los intrusos, "piratas informáticos", hackers, crackers o como se quieran denominar, constituyen un obvio problema de seguridad tanto para las redes como para los hosts. Los problemas que causan estos personajes no son algo nuevo: el primer trabajo sobre detección de intrusos data de 1980 y desde entonces este campo ha sido, es, y con toda probabilidad seguirá siendo, uno de los más activos en la investigación dentro del mundo de la seguridad informática.

De esta forma, los Sistemas de Detección de Intrusos son uno de los mejores complementos a la hora de pensar en una propuesta de seguridad robusta y que brinde alta confiabilidad y entre ellos Snort es uno de los IDS más utilizados y que proveen mayor número de aplicaciones para todas las necesidades en una red.

1.3 SEGURIDAD WEB (WEB SECURITY)

Hasta el momento, se ha presentado un servicio Web que ofrece un acceso abierto a un conjunto de información que explícitamente se hace pública. Sin embargo, en determinadas circunstancias, es interesante poder limitar el acceso a documentos reservados o útiles para un conjunto restringido de personas. Se pueden establecer dos tipos de restricciones:

- Limitación de acceso en función de direcciones IP o dominio. Sólo los usuarios de un dominio u organización tendrán acceso a la información.
- Limitación de acceso por nombres de usuario y claves de acceso. Sólo los usuarios que conozcan una clave de acceso válida pueden acceder a la información.

Otro aspecto que está cobrando especial importancia es la seguridad de la información que se intercambia en el Web. La explotación comercial de Internet exige disponer de sistemas de comunicación seguros, capaces de adaptarse a las necesidades de los nuevos servicios, como la compra electrónica o la banca a distancia. En estos servicios, se manejan dos conceptos fundamentales: la autenticación (garantizar que tanto el usuario de un cliente Web como un determinado servidor de información son quienes dicen ser) y la confidencialidad (hacer que la información intercambiada no pueda ser interceptada por terceros).

Con los sistemas de comunicación actualmente en uso, es técnicamente posible escoger un enlace de comunicación e interceptar el contenido de las comunicaciones TCP/IP que por él se transmiten. Cuando se envía información privada, por ejemplo un número de tarjeta de crédito en un formulario de compra, es vital garantizar que la información sea recibida exclusivamente por su destinatario, y que la identidad sea la esperada.

➤ *Seguridad en la transmisión*

La seguridad de este tipo se basa en el hecho de poder encriptar los mensajes que se envían por una red entre un servidor y un cliente y que solo ellos puedan descifrar los contenidos a partir de una clave común conocida solo por los dos.

Para llevar a cabo esta seguridad se crearon diversos protocolos basados en esta idea:

- SSH: Usado exclusivamente en reemplazo de telnet.
- SSL: Usado principalmente en comunicaciones de hipertexto pero con posibilidad de uso en otros protocolos.
- TLS: Es del mismo estilo del anterior.
- HTTPS: Usado exclusivamente para comunicaciones de hipertexto.

➤ *Control de acceso a la información*

Se utiliza para limitar el acceso a determinados documentos de un servidor Web, en función del origen y tipo de petición. La forma de hacerlo varía con el entorno en el que se publican las páginas (sistema operativo y servidor HTTP, principalmente); en general, todas las soluciones pasan por definir un fichero que contiene las diferentes limitaciones de acceso, en un formato característico del servidor HTTP. En algunos casos se utiliza un fichero global con las restricciones de acceso o bien un fichero por cada directorio al que se quiere limitar el acceso.

Cuando un cliente Web accede a un fichero protegido, el servidor devuelve un código de error asociado a la falta de permisos para realizar la operación (código 401). Si el acceso se realiza desde un dominio o dirección IP prohibida, no será posible acceder a la información desde ese sistema. Cuando la protección se basa en nombres y claves de acceso, el *browser* solicitará estos datos y los enviará al servidor para que sean verificados. Las claves de acceso se envían al servidor por diferentes sistemas, sin codificar (sencillo pero inseguro) o codificadas (DES o Kerberos, por ejemplo). Será el propio servidor HTTP el que informe sobre la manera en que se deben enviar estas claves de acceso.

➤ *Control de acceso en un servidor CERN*

La versión 3.0 del servidor desarrollado por el CERN (Centro Europeo para la Investigación Nuclear) permite limitar el acceso a documentos o grupos de documentos, en función de nombres de usuario o direcciones de origen. El control de acceso se puede realizar para todo el servidor, modificando los ficheros globales de configuración o para un directorio concreto. El control de acceso al contenido de un directorio se realiza creando un fichero de nombre **.www_acl**, en el mismo directorio que los ficheros cuyo acceso se quiere controlar. Este fichero está formado por líneas, cada una de ellas fijando una limitación de acceso diferente. Para cada especificación de ficheros, se indica los comandos HTTP permitidos y los usuarios o grupos de usuarios que pueden acceder. Cuando se añade un control de acceso, automáticamente se deshabilita el acceso para los usuarios o grupos no incluidos. Se utiliza el mecanismo de autenticación básica, en la cual las claves de acceso son transferidas por la red sin codificar.

Se pueden crear usuarios o grupos de usuarios con la aplicación **htadm**, a través de la cual se generan nuevos usuarios y se les asigna claves de acceso. Además, a través de la configuración global del servidor, es posible fijar permisos de acceso por defecto, o restringir el uso del servidor a determinadas direcciones (o rangos de direcciones) IP.

➤ *Control de acceso en un servidor NCSA*

El servidor HTTP de la NCSA (National Center for Supercomputing Applications) permite limitar el acceso en función de direcciones de origen o nombres de usuario (esto se aplica también a Apache, desarrollado a partir de éste). El procedimiento es similar al del servidor del CERN. Se debe crear un fichero de nombre **.htaccess** en cada directorio cuyos ficheros requieran protección; de esta forma el control de acceso afecta a todos los ficheros del directorio protegido. Se puede conceder o denegar el acceso en función de direcciones IP, en cuyo caso se utilizaría un fichero de control de acceso de la forma (all equivale a cualquier petición):

```
<Limit GET POST>
deny from all
allow from usuarios.unicauca.edu.co
</Limit>
```

Además, NCSA soporta los sistemas de autenticación básico (en el que las claves circulan de forma visible por la red) o MD5 (que añade una codificación a estas claves). Los ficheros de usuarios y claves se crean con la aplicación `htpasswd`, que permite editar un fichero de claves (similar al `passwd` de UNIX).

➤ *Control de acceso en un servidor Microsoft*

El servidor HTTP de Microsoft puede limitar el acceso a máquinas o grupos de máquinas, a través de la utilidad de configuración del servidor (el *Administrador de Servicios Internet*). Para ello, se agrega la máscara de red de aquellos sistemas a los que se concede (o niega) el acceso al servidor. Además, es posible controlar de forma individual el acceso a cualquier documento o directorio del servidor, sirviéndose de los permisos de acceso a ficheros y la base de usuarios del servidor. Para poder utilizar este tipo de control de acceso, es necesario que el sistema de ficheros en que residen los documentos Web tenga formato NTFS, el único que permite asignar permisos de acceso a ficheros.

1.3.1 Seguridad y privacidad

El intercambio seguro de información a través de una red abierta e insegura como Internet ha obligado a desarrollar numerosos sistemas de encriptación y autenticación de las transacciones, destinados a cubrir tres problemas fundamentales:

- Conocer la identidad real de los clientes y servidores que se comunican, de forma que ambos dispongan de algún sistema para verificar la identidad del otro. Este tipo de identificación tiene particular importancia en los negocios virtuales, ya que al enviar un número de tarjeta de crédito para realizar un pago se tiene que estar seguro de que el destinatario es quien dice ser.
- Garantizar que la transferencia de datos sólo pueda ser entendida por las aplicaciones que se comunican, utilizando métodos criptográficos para codificar todos los datos intercambiados, y evitar los accesos no autorizados en la red.
- Garantizar la integridad de los datos enviados, teniendo capacidad de detectar cualquier cambio, intencionado o no, en los mismos.

La mayoría de los intercambios seguros de información se realizan según un sistema denominado *Criptografía de Clave Pública*; cada extremo de la comunicación dispone de dos claves, una pública que cualquiera puede solicitar y conocer, y otra

privada, cuya seguridad es fundamental para el éxito de la codificación. Para enviar un mensaje seguro a una persona, se solicita su clave pública, con la que se codifica el mensaje. El sistema garantiza que el mensaje resultante sólo puede ser descodificado con la clave privada del destinatario. El siguiente paso es asegurar la correcta identidad del destinatario. Para ello, se han creado las *autoridades de certificación*, organizaciones o empresas que distribuyen *certificados*, unos documentos digitales que contienen la identidad y clave pública de una determinada organización. Cuando se establece una conexión segura con un servidor HTTP, es posible acudir a una de estas autoridades de certificación para verificar su identidad. Una de las autoridades de certificación más conocidas es *Verisign* (www.verisign.com), una empresa que proporciona diversos tipos de certificados, personales o para empresas, tras un proceso de verificación de la identidad del solicitante (previo pago de una tarifa). Los clientes Web tienen una pequeña base de datos con las claves públicas de diversas autoridades de certificación, a partir de las cuales se pueden verificar las claves públicas de otros servidores. De esta forma, si se codifica un mensaje con la clave pública de un determinado servicio Web, la información enviada sólo podrá ser interpretada por el destinatario del mensaje.

A partir de la criptografía de clave pública se ha desarrollado SSL (*Secure Sockets Layer*), un sistema de codificación de información propuesto por Netscape que codifica toda la información transferida al nivel de conexiones TCP, por lo que es compatible con todos los protocolos y servicios de Internet. SSL está incluido en los clientes Web de Netscape, Microsoft, IBM, etc. SSL combina criptografía de clave pública, para el intercambio de las claves de cifrado, y criptografía de clave privada, para el intercambio de información. Otro sistema bastante utilizado es S-HTTP, una versión de HTTP que incorpora criptografía de clave pública para la autenticación y el intercambio seguro de datos. Sin embargo, S-HTTP sólo puede utilizarse para intercambiar datos entre clientes y servidores Web, mientras que SSL actúa de forma transparente para cualquier aplicación de comunicaciones TCP/IP.

Los clientes Web muestran avisos de advertencia cuando la conexión pasa de modo normal a modo seguro y viceversa (el candado o llave abiertos de la esquina inferior izquierda de Netscape Navigator o Internet Explorer). A partir de este momento se puede asegurar la identidad del servidor remoto, y la total privacidad de los datos intercambiados. Los clientes Web modernos además disponen de una base de datos con los datos de varias autoridades de certificación, que se utilizarán para verificar las 'identidades digitales' de los servidores HTTP con los que se trate de establecer una conexión segura.

1.3.2 El sistema operativo

El sistema operativo está formado por el software que permite acceder y realizar las operaciones básicas en un ordenador personal o sistema informático en general. Los sistemas operativos más conocidos son: AIX (de IBM), GNU/Linux, HP-UX (de HP), MacOS (Macintosh), Solaris (de SUN Microsystems), las distintas variantes del UNIX de BSD (FreeBSD, OpenBSD), y Windows en sus distintas variantes (de la empresa Microsoft). En lo que a seguridad se refiere, un sistema operativo puede caracterizarse por:

- Consideración de la seguridad en el diseño: Hay sistemas operativos que han sido creados con la seguridad como objetivo fundamental de diseño. Estos serán de entrada más seguros que los demás. En otros sistemas operativos aunque no fuera el

objetivo fundamental sí ha podido ser un parámetro importante y por último en otros no se ha considerado más que posteriormente. Es de esperar que sean éstos últimos los que más problemas de seguridad tienen.

- Capacidades de comunicación y configuración de esta: Los sistemas operativos modernos ofrecen grandes capacidades de comunicación. Desde el punto de vista de la seguridad estas capacidades pueden convertirse en puntos de acceso a posibles atacantes y será necesario protegerlos. El sistema operativo deberá proveer de los mecanismos y/o herramientas necesarias para llevar a cabo esta tarea de forma suficientemente fiable. Esto incluye ofrecer la capacidad de cerrar toda vía de comunicación que no se use y limitar la que sí se emplea los casos y usuarios que realmente se deseen permitir.
- Capacidades de auditoría: Estas capacidades son las que van a permitir determinar qué elementos acceden a qué partes del sistema en sus distintos niveles (ficheros, dispositivos, elementos de comunicación), etc.
- Herramientas disponibles: Dado que en general las aplicaciones no son portables entre sistemas operativos de distintos fabricantes (exceptuando algunos casos entre las distintas variantes de UNIX) otro elemento a considerar en la seguridad de cada sistema operativo se caracteriza por la cantidad y calidad de herramientas de seguridad que tiene disponibles.

En algunos sistemas operativos se accede al sistema por medio de un usuario único que tiene permiso para realizar cualquier operación. Este es el caso de los sistemas operativos más antiguos como MS-DOS y algunos más recientes como la serie Windows 95/98/Me de Microsoft o MacOS (antes de MacOSX) de Macintosh. En estos sistemas no existe una diferenciación clara entre las tareas que realiza un administrador del sistema y las tareas que realizan los usuarios habituales; no disponiendo del concepto de multiusuario, un usuario común tiene acceso a todas las capacidades del sistema, pudiendo borrar, incluso, información vital para su funcionamiento. Un usuario malicioso (remoto o no) que obtenga acceso al sistema podrá realizar todo lo que desee por no existir dichas limitaciones.

Otros sistemas operativos, sin embargo, han estado siempre preparados para soportar sistemas multiusuario, permitiendo agruparlos y asignar distintos privilegios a cada uno de ellos o a sus grupos. Este es el caso de todos los sistemas UNIX y de los sistemas Windows NT/2000/XP. Esta característica es enormemente útil desde el punto de vista de seguridad. Por ejemplo en el caso de que un usuario se vea afectado por un virus, una intrusión, etc. el resto de los usuarios (si los hay) y, sobre todo el sistema, no tendrán por qué verse afectados a menos que vulnerabilidades en éstas puedan ser utilizadas por un atacante para elevar sus privilegios.

Cabe notar que los sistemas operativos libres (Linux y BSD) no soportan una asignación de grupos y usuarios tan versátil como Windows NT, 2000 o XP. Los grupos en UNIX son mucho menos versátiles (y más difíciles de administrar). Queda claro que en todo equipo donde la seguridad es un factor que se considera importante, debe optarse por un sistema operativo que soporte varios usuarios con distintos privilegios.

1.3.3 HTTP - Protocolo de Transferencia de Hipertexto

HTTP (HyperText Transfer Protocol) es un protocolo cliente/servidor usado en cada transacción de la Web (WWW) que articula los intercambios de información entre los clientes Web y los servidores HTTP. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuesta de esa Web, remitiendo la información que se verá en pantalla. También sirve para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares. HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las *cookies*, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio Web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio Web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio. La versión actual de HTTP es la 1.1, y su especificación está en el documento RFC2616.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores. HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

El estándar HTTP/1.0 recoge únicamente tres comandos, que representan las operaciones de recepción y envío de información y chequeo de estado:

- **GET:** Se utiliza para recoger cualquier tipo de información del servidor. Se utiliza siempre que se pulsa sobre un enlace o se teclea directamente una URL. Como resultado, el servidor HTTP envía el documento correspondiente a la URL seleccionada, o bien activa un módulo CGI, que generará a su vez la información de retorno.
- **HEAD:** Solicita información sobre un objeto (fichero): tamaño, tipo, fecha de modificación; es utilizado por los gestores de cachés de páginas o los servidores Proxy, para conocer cuándo es necesario actualizar la copia que se mantiene de un fichero.

- *POST*: Sirve para enviar información al servidor, por ejemplo los datos contenidos en un formulario. El servidor pasará esta información a un proceso encargado de su tratamiento (generalmente una aplicación CGI). La operación que se realiza con la información proporcionada depende de la URL utilizada. Se utiliza, sobre todo, en los formularios.

Un cliente Web selecciona automáticamente los comandos HTTP necesarios para recoger la información requerida por el usuario. Así, ante la activación de un enlace, siempre se ejecuta una operación GET para recoger el documento correspondiente. El envío del contenido de un formulario utiliza GET o POST, en función del atributo de `<FORM METHOD="...">`. Además, si el cliente Web tiene un caché de páginas recientemente visitadas, puede utilizar HEAD para comprobar la última fecha de modificación de un fichero, antes de traer una nueva copia del mismo.

Posteriormente se han definido algunos comandos adicionales, que sólo están disponibles en determinadas versiones de servidores HTTP, con motivos eminentemente experimentales. La última versión de HTTP, recoge estas y otras novedades, que se pueden utilizar, por ejemplo, para editar las páginas de un servidor Web trabajando en remoto.

➤ *Etapas de una transacción HTTP.*

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor, etc.
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.

- Se cierra la conexión TCP.

1.4 ROUTER SECURITY

Los enrutadores pueden jugar un rol muy importante en la seguridad de las redes; en esta parte se pretende describir algunos aspectos generales a tener en cuenta a la hora de proteger un router y gestionarsu seguridad.

1.4.1 Protegiendo al Router por sí mismo

- Seguridad Física:

Hay muchas formas de proveer seguridad física a un router. El cuarto donde se encuentran estos equipos debe estar libre de electrostática o interferencia magnética. Además debe controlarse la temperatura y la humedad del mismo. Es muy importante poder asegurar una alimentación ininterrumpida por medio de la instalación de una UPS y configurarlo para que utilice la mayor cantidad de memoria soportada, para evitar los ataques por denegación de servicio. Estos equipos deben ser ubicados en cuartos asegurados que sean accedidos solo por personas autorizadas.

- Sistema Operativo:

El sistema operativo de un router es un elemento crucial. Deben analizarse que características de la red se necesitan, y utilizar esta lista de características para seleccionar el sistema operativo. Sin embargo, la última versión de algún sistema operativo no siempre es la más acertada, y puede no estar aún en un nivel estable, aunque si es importante mantener actualizado el sistema operativo del router con la última versión estable, ya que estas por lo general han mejorado aspectos de seguridad y rendimiento.

- Fortaleza de la Configuración:

Un router es similar a muchos computadores y por lo tanto, tiene muchos servicios habilitados por defecto; muchos de estos servicios son innecesarios y pueden ser usados por un atacante para reunir información o para explotación. Todos los servicios que sean innecesarios deben ser deshabilitados en la configuración del router y mantener una auditoría continua de ésta para mantenerla actualizada.

1.4.2 Protegiendo una red con un Router

Los routers son dispositivos muy importantes a la hora de hablar de la operación y la seguridad de la red; su rendimiento les permite ser utilizados en varias formas, que se pueden resumir en 3 grupos:

1. *Routers Internos*: Un router interno reenvía tráfico entre dos o más redes locales dentro de una organización o empresa. Las redes conectadas por medio de un router interno a menudo comparten las mismas políticas de seguridad, y el nivel de confianza entre ellas es usualmente alto. Si una empresa tiene muchos routers internos, se empleará un Protocolo de Enrutamiento Interior para gestionar estos routers; la ventaja en la seguridad, es que éstos routers pueden imponer algunas restricciones en el tráfico que ellos reenvían entre las redes. Una configuración muy utilizada se muestra a continuación:

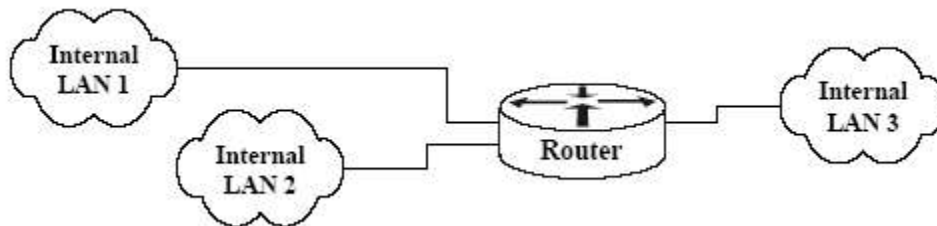


Figura 4. Router Interno

2. *Routers Externos o de Backbone*: Este tipo de router reenvía tráfico entre redes de diferentes empresas o Sistemas Autónomos. El tráfico entre las diferentes redes que componen Internet es redireccionado por routers externos. El nivel de confianza entre las redes conectadas por routers externos es usualmente muy bajo. Típicamente, los routers externos están diseñados y configurados para reenviar tráfico lo más rápido posible, sin imponer muchas restricciones sobre éste. La meta principal de seguridad de un router externo es asegurar que la gestión y operación del router están siendo manejadas solo por las partes autorizadas, además de proteger la información de enrutamiento que se usa para reenviar el tráfico. Estos routers utilizan Protocolos de Enrutamiento Exteriores. La configuración de routers externos es mucho más compleja y se requiere tener claro el nivel de seguridad que se desea aplicar, sin afectar el rendimiento de la red. A continuación se muestra un ejemplo de varios routers externos conectando varias redes:

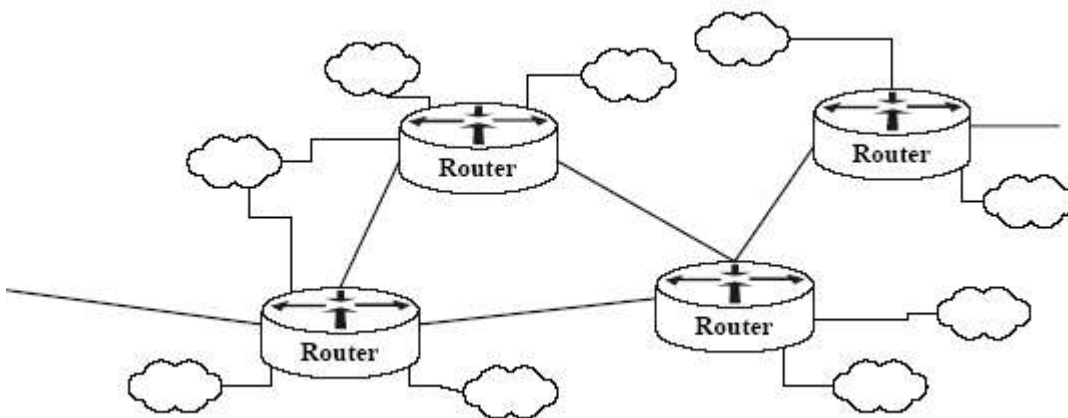


Figura 5. Routers Externos

3. *Routers de Borde*: Un router de borde reenvía tráfico entre una empresa u organización y una red externa. El aspecto clave en un router de borde es que éste forma parte de la frontera entre las redes internas de la empresa que manejan un muy alto nivel de confianza y las redes externas y poco confiables (Internet). Este puede ayudar a asegurar el perímetro de la red de una empresa reforzando las restricciones del tráfico que él controla. Un router de borde puede utilizar protocolos de enrutamiento, o puede depender totalmente de rutas estáticas predefinidas por el administrador.

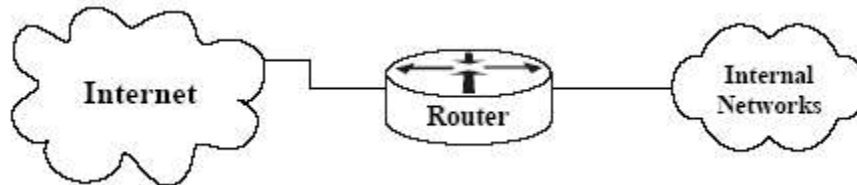


Figura 6. Router de Borde

Por lo general un router de borde no es el único componente de la frontera; muchas empresas emplean Firewalls para reforzar el cumplimiento de las Políticas de Seguridad. En la figura 6, el router de borde actúa como la primera línea de defensa y por esto a menudo se le conoce como Router de Prueba. Este contiene una ruta estática que envía hacia el Firewall todas las conexiones que se dirigen a la red protegida; el Firewall provee control de acceso adicional sobre estas conexiones y sobre el tráfico de la red. El Firewall puede también proporcionar autenticación de usuarios. Utilizar un router y un Firewall de esta forma, definitivamente ofrece un mayor nivel de seguridad que utilizarlos cada uno por separado.

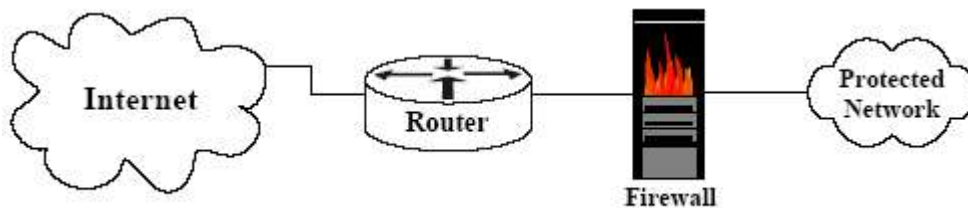


Figura 7. Configuración de Seguridad usando un Router y un Firewall

1.4.3 Filtros de Paquetes

Un servicio de filtrado de paquetes provee control de la información transferida entre redes basadas en direcciones y protocolos. Los Routers pueden aplicar filtros en diferentes formas. Algunos routers tienen filtros que se aplican a servicios de red en ambas direcciones, hacia el interior y hacia el exterior, mientras otros tienen filtros que se aplican en una sola dirección. La mayoría de los routers pueden filtrar uno o más de los siguientes parámetros: Dirección IP del origen, Puerto origen, Dirección IP del destino, Puerto

Destino y tipo de Protocolo. Algunos routers pueden aún filtrar algún bit o algún patrón de bits en el encabezado IP; sin embargo, los routers típicamente no tienen la capacidad de filtrar por contenido o por nombre de los servicios.

Los filtros de paquetes son especialmente importantes para los routers que actúan como gateways entre redes de confianza y poco confiables. En ese caso, el router puede aplicar la seguridad rechazando protocolos y restringiendo puertos de acuerdo a las políticas de la red confiable. Los filtros son también importantes por su capacidad de aplicar restricciones de direccionamiento. Cuando un router aplica la restricción para que los paquetes enviados por el Firewall o desde la red protegida tengan un rango de direcciones particular, se denomina *Egress Filtering*. De forma similar, cuando el router aplica la restricción para los paquetes que llegan desde Internet con una dirección de Origen fuera del rango válido para la red protegida, se denomina *Ingress Filtering*.

1.4.4 Filtrado de Direcciones

Los filtros de los Routers deben ser usados para proteger contra el IP Spoofing, especialmente en Routers de borde. En la mayoría de los casos las reglas de filtrado deben aplicarse en el filtrado entrante y saliente incluyendo bloques de direcciones reservadas. Los principios que deben ser aplicados en los Routers de borde se listan a continuación:

- Rechazar todo el tráfico desde las redes internas que lleven una dirección IP de origen la cual no pertenezca a las redes internas.
- Rechazar todo el tráfico de las redes externas que lleven una dirección IP de origen que pertenezca a las redes internas, ya que, si las direcciones han sido asignadas correctamente, el tráfico enviado desde las redes externas debe siempre llevar una dirección de origen diferente a las del rango asignado para las redes internas.
- Rechazar todo el tráfico con una dirección de fuente o destino perteneciente a un rango reservado, no enrutable o ilegal.

1.4.5 Mitigando los ataques de Denegación de Servicio

La pérdida de servicio o la severa degradación del rendimiento de la red pueden resultar de una variedad de causas. La Denegación de Servicio (DoS) se refiere a una serie de atentados que ocasionan estos problemas en una red. No hay una solución completa para evitar que estos ataques afecten la red; mientras los recursos de una red sean limitados y estén disponibles abiertamente, ellos serán vulnerables a ataques. Hay ciertas medidas que los administradores de la red pueden tomar para proteger las redes de los ataques de Denegación de Servicio y de sufrir sus efectos. Esas medidas requieren esfuerzo corporativo entre los administradores de hosts, de dispositivos de red y de los proveedores del servicio.

Para cualquier empresa, hay 3 estrategias primarias para combatir los ataques DoS:

- Prevenir el tráfico malicioso desde la entrada de una red común hasta la red de la empresa.
- Configurar y desarrollar medidas de protección local tanto en los routers de borde como en los routers interiores.
- Coordinar medidas de protección contra los ataques DoS distribuidos en conjunto con los proveedores de acceso y/o administradores del backbone de la red.

Hay varios mecanismos disponibles en los routers para frustrar ciertas clases de ataques DoS. Muchos de esos ataques requieren el uso inválido de direcciones de origen; hay muchas formas de filtrar esos paquetes con direcciones inapropiadas. Las listas de control de acceso son una facilidad de filtrado disponible en todos los routers. La mayoría de los routers Cisco soportan una facilidad llamada *Unicast Reverse-Path Forwarding Verification* que utiliza la tabla de rutas para detectar y disminuir paquetes diseccionados inapropiadamente. En el caso de ser posible, se deben utilizar los registros del log de eventos, identificando paquetes defectuosos y otras violaciones que puedan ayudar a identificar hosts comprometidos que necesiten ser removidos de la red. Obviamente esta detección dependerá de una revisión constante de los logs de los routers.

Como se puede ver, no hay un set de métodos preestablecidos para eliminar todas las clases de ataques, pero los routers sí son parte de la solución, junto con precauciones de diseño, contingencia y planeación por parte de los administradores de la red.

1.5 ACCESO REMOTO

Como su nombre indica, el Servicio de acceso remoto le permite conectarse a la red por medio de una conexión telefónica. Una vez conectado, puede hacer lo mismo que si estuviera trabajando en un equipo conectado físicamente a la red. Por medio de acceso remoto es posible realizar las mismas actividades que si se estuviera físicamente en el equipo, a menos que, por alguna razón, se necesite tener acceso físico al equipo.

Coincidiendo con la aparición del concepto del teletrabajo, y de la necesidad de interconectar tanto redes locales, por ejemplo de diversas delegaciones de una misma empresa, como puestos de trabajo autónomos móviles con la oficina o de buscar mecanismos de acceso a bases de datos y otras redes de información (Internet), se determina la aparición de un nuevo tipo de dispositivos de internetworking: los servidores de acceso remoto.

En un principio, las necesidades de interconexión entre dos redes locales, se resolvía mediante el uso de puentes (bridges) o encaminadores (routers), e incluso mediante pasarelas (gateways) en algunos casos. Sin embargo, estos dispositivos, son extremadamente caros y complejos en su configuración y mantenimiento. En cualquier caso, cuando se trata únicamente de interconectar dos redes locales, no son mala solución, pero hay que tener en cuenta que se requieren dos equipos iguales, o con protocolos totalmente compatibles en cada extremo, lo que implica una inversión doblemente elevada. Sería como emplear dos routers para interconectar a un usuario remoto con su oficina, por ejemplo un teletrabajador, o bien un vendedor, con su portátil que tiene que reportar diariamente a su oficina principal para enviar la información de pedidos de clientes. Evidentemente, no es el costo la única razón para no emplear routers en este tipo de conexiones, sino su tamaño y especialmente su complejidad. Por esto, hoy en día los encaminadores son capaces de aceptar conexiones directas de modems o adaptadores de terminal, sin necesidad de que al otro lado hubiera un equipo equivalente. Esto permite denominar los servidores de comunicaciones, o *servidores de acceso remoto*.

Básicamente, un servidor de comunicaciones o acceso remoto es un encaminador, con una serie de puertos serie que a su vez pueden tener diferentes tipos de interfaz (RS-232, V.35, RDSI, etc.), en función del tipo de conexiones que pueda aceptar. Por lo general, un servidor de acceso remoto se puede comportar como un encaminador entre dos redes, y es capaz de recibir llamadas de equipos remotos, que a su vez no son encaminadores. Para esto, ambos, el servidor de acceso remoto, y el equipo remoto, deben emplear un protocolo compatible. El más usado es el PPP (Point to Point Protocol, o Protocolo Punto a Punto), y en segundo plano el SLIP (Serial Line Interface Protocol, o Protocolo de Interconexión de Líneas Serie). Por supuesto, hay variaciones de ambos, fundamentalmente orientadas a lograr una compresión de los datos transmitidos.

Esto requiere, en el caso del equipo remoto, la instalación de un software de comunicaciones o conjunto de utilidades del sistema operativo que incorporen dicho protocolo. Así por ejemplo, Windows incorpora de base ambos protocolos. Por supuesto, detrás de dichos protocolos existirá otro u otros, como pueden ser TCP/IP, IPX, LAT, NetBEUI, etc., en función del sistema operativo o aplicaciones.

1.5.1 Aplicaciones

Básicamente se pueden dividir las aplicaciones de un servidor de comunicaciones en cinco grupos fundamentales:

- Interconexión entre redes LAN: sustituyendo por completo a las funciones de los encaminadores, permiten realizar la conexión entre dos redes locales remotas (típicamente una oficina principal y sus delegaciones), y siendo en este caso su principal misión el enrutamiento (routing) de los paquetes, de modo que dicha conexión sea transparente a usuarios, aplicaciones y hardware/software existente en ambas redes. Se pueden incluso dedicar varias líneas para interconectar dos redes, en función del tráfico existente en cada momento entre ambas (ancho de banda por demanda o "bandwidth on demand").
- Acceso de nodos remotos: cuando la conexión que se requiere es entre una red (oficina) y un solo usuario (vendedor, o teletrabajador), mediante un software en el equipo remoto que sea compatible con el protocolo empleado en el servidor de comunicaciones.
- Acceso a Internet, o redes similares: en realidad se trata de ejemplos aplicables al caso 1 o 2, antes mencionados.
- Acceso a Bases de datos y otras aplicaciones: un servidor de comunicaciones puede ser empleado para gestionar un conjunto de módems, para permitir a los usuarios de la red local a la que está conectado, el acceso a diversos servicios (bases de datos, y otros), sin necesidad de que cada usuario tenga su propio módem. Esto puede ser válido también para el envío de fax.
- Servicios de terminales e impresoras remotas: empleando así terminales e impresoras serie tanto para su uso por parte de usuarios locales como de nodos remotos.

- Los servidores de acceso remoto no solo incorporan funcionalidades de puentes y encaminadores, sino también de otros dispositivos como servidores de terminales e impresoras, lo que demuestra su alto nivel de sofisticación, que sin duda se verá incrementando aún más en un futuro muy cercano.

1.6 REGISTROS Y AUDITORIA

Una vez que se han establecido los mecanismos de seguridad en una red, es necesario monitorearlos para asegurarse de que están funcionando debidamente; de igual forma es importante observar algunos indicadores, comportamientos errados u otros problemas. Para analizar efectivamente la seguridad de una red y responder a los incidentes de seguridad, deben establecerse procedimientos para recolectar información de las actividades de la red y llevar un registro de éstos. Esto es a lo que se le llama Auditoría (Accounting or Auditing). Una auditoría permite rastrear sucesos que ocurren en una máquina, servidor o estación de trabajo. Para redes con estrictas políticas de seguridad, auditar la información debe incluir todos los intentos para conseguir autorización y autenticación para cualquier persona. Es especialmente importante registrar accesos “anónimos” o “invitados” a servidores públicos. Esta información debe registrar también todos los intentos de los usuarios para cambiar sus privilegios de acceso.

La información recolectada debe incluir nombre de usuario y de host para intentos de inicio o terminación de sesión (login y logout), y privilegios de acceso previos y nuevos en el caso de un cambio en los privilegios de acceso. Cada entrada en un archivo de registro o log debe tener asociado la hora de ocurrencia. El proceso de auditoría no debe reunir passwords. Recolectar la información de los passwords crea una potencial brecha de seguridad si estos registros de auditoría son accedidos indebidamente.

Otra extensión de la auditoría es el concepto de Evaluación de Seguridad; en ésta, la red es examinada desde adentro por profesionales, y calificada en las vulnerabilidades explotadas por los invasores de la red. Parte de algunas políticas de seguridad y procedimientos de auditoría deben ser evaluaciones periódicas de las vulnerabilidades en una red. El resultado debe ser un plan específico para corregir deficiencias, el cual puede ser un simple estado de reciclaje.

Las auditorías son esenciales para mantener la seguridad de los servidores y las redes, ya que permiten un seguimiento de las actividades realizadas por los usuarios. Básicamente se pueden realizar 3 tipos de auditorías:

- Auditoría de cuentas de usuario: rastrear los sucesos de seguridad y escribir apuntes en el registro de seguridad. Algunos de los eventos que se pueden auditar son:
 - Log on y log off en la red.
 - Acceso a ficheros, directorios o impresoras.
 - Ejercicio de los derechos de un usuario.
 - Seguimiento de procesos.
 - Arranque del sistema.

- Auditoría del sistema de archivos: rastrea sucesos del sistema de archivos. Los eventos que se pueden auditar son: lectura, escritura, ejecución, eliminación, cambio de permisos y toma de posesión.
- Auditoría de impresoras: uso de la impresora, cancelar trabajos de impresión, control total de la impresora, etc.

De la misma forma es importante mencionar un concepto relacionado: la Gestión de Red, que ha venido cobrando fuerza a medida que los avances tecnológicos aumentan y el tamaño y prestaciones de las redes son mayores; la Gestión consiste en monitorizar y controlar los recursos de una red con el fin de evitar que ésta llegue a funcionar incorrectamente degradando sus prestaciones. Entre los elementos principales que se deben tener en cuenta a la hora de hablar de gestión se encuentran:

- Gestor: estación donde se lleva a cabo la gestión.
- Agente: sistemas que van a ser gestionados; se trata de software que responde a solicitudes de información del gestor y que proporciona información no solicitada pero de vital importancia.
- MIB: base de información de gestión
- Objetos Variable que representa el aspecto de un agente.
- Protocolo.

Una forma muy utilizada de llevar a cabo auditoría de un sistema o una red es utilizando los archivos *log*, que proveen todos los principales Sistemas Operativos; en éstos es posible obtener información de lo que ha pasado en la red o en un equipo particular. Hoy en día los archivos log prestan muchas facilidades, como recolectar información a cerca de archivos transferidos sobre la red, intentos de usuarios por obtener privilegios de administradores, correos electrónicos y muchas más. Estos logs son un bloque importante a la hora de construir un sistema de seguridad: forman una especie de historia grabada, o *camino de auditoría*.

1.6.1 Gestión de Seguridad

La Gestión de Seguridad le permite al administrador de la red mantener y distribuir passwords y otra información de autenticación y autorización; también incluye procesos para generar, distribuir y almacenar claves de cifrado. Esta puede incluir herramientas y reportes para analizar la configuración de un grupo de routers y switches para que obedezcan una serie de estándares de seguridad. Un aspecto importante de la Gestión de Seguridad es un proceso para reunir, almacenar y examinar los logs de auditoría de seguridad. Como ya se mencionó, estos logs deben documentar inicios y términos de sesión y intentos de usuarios de cambiar sus niveles de autorización.

Reunir información de auditoría puede resultar en una rápida acumulación de datos. El almacenamiento requerido puede ser minimizado manteniendo la información por un corto periodo de tiempo y resumirla. Sin embargo, una desventaja de mantener menos información almacenada es que hace más difícil investigar los incidentes de seguridad ocurridos; la mayoría de las veces la mejor

solución es comprimir los datos. También es una buena idea encriptar los logs de auditoría; un hacker que logre acceder estos logs puede causar daños irreversibles para la red si logra obtener la información conveniente. El hacker puede además cambiar el archivo log sin ser detectado y también obtener información sensible de éste.

Existe una gran variedad de herramientas para mantener los logs de seguridad, entre ellas Event Viewer en las máquinas Windows 2000, Syslog en Windows, Unix y Dispositivos Cisco y C2 Auditing para reportes detallados de auditoría en sistemas Unix.

1.6.2 Syslog

Syslog es un estándar de facto para el envío de mensajes de registro en una red informática IP. Por *syslog* se conoce tanto al protocolo de red como a la aplicación o biblioteca que envía los mensajes de registro. Un mensaje de registro suele tener información sobre la seguridad del sistema, aunque puede contener cualquier información. Junto con cada mensaje se incluye la fecha y hora del envío.

➤ Usos

Es útil registrar, por ejemplo:

- Un intento de acceso con contraseña equivocada.
- Un acceso *correcto* al sistema.
- Anomalías: variaciones en el funcionamiento normal del sistema.
- Alertas cuando ocurre alguna condición especial.
- Información sobre las actividades del sistema operativo.
- Errores del hardware o el software.
- También es posible registrar el funcionamiento normal de los programas; por ejemplo, guardar cada acceso que se hace a un servidor web, aunque esto suele estar separado del resto de alertas.

➤ Protocolo

El protocolo *syslog* es muy sencillo: existe un equipo servidor ejecutando el servicio de syslog, conocido como *syslogd* (demonio de syslog). El cliente envía un pequeño mensaje de texto (de menos de 1024 bytes). Los mensajes de syslog se suelen enviar vía UDP, por el puerto 514, en formato de texto plano. Algunas implementaciones del servidor, como *syslog-ng*, permiten usar TCP en vez de UDP, y también ofrecen [Stunnel](#) para que los datos viajen cifrados mediante SSL/TLS. Aunque *syslog* tiene algunos problemas de seguridad, su sencillez ha hecho que muchos dispositivos lo implementen, tanto para enviar como para recibir. Eso hace posible integrar mensajes de varios tipos de sistemas en un solo repositorio central.

Syslog fue desarrollado por Eric Allman como parte del proyecto [Sendmail](#) (1980). Sin embargo, se comprobó que era muy útil, y otras aplicaciones empezaron también a usar *syslog*. Hoy en día, *syslog* está presente por defecto en casi todos los sistemas [Unix](#) y GNU/Linux, y también se encuentran diversas implementaciones de *syslog* para otros sistemas operativos, como Microsoft Windows.

Es ahora, después de tantos años, cuando *syslog* está en proceso de convertirse en estándar, para -entre otras cosas- poder mejorar la seguridad de sus implementaciones. IETF asignó un grupo de trabajo, y en 2001, se documentó su funcionamiento en el RFC 3164. La estandarización del contenido del mensaje de las diferentes capas de abstracción está planificada para este año.

➤ *Estructura del mensaje*

El mensaje enviado se compone de tres campos:

- Prioridad
- Cabecera
- Texto

Entre todos no han de sumar más de 1024 bytes, pero no hay longitud mínima.

La **prioridad** es un número de 8 bits que indica tanto el *recurso* (tipo de aparato que ha generado el mensaje) como la *severidad* (importancia del mensaje), números de 5 y 3 bits respectivamente. Los códigos de recurso y severidad los decide libremente la aplicación, pero se suele seguir una convención para que clientes y servidores se entiendan.

El segundo campo de un mensaje *syslog*, la **cabecera**, indica tanto el *tiempo* como el *nombre* del ordenador que emite el mensaje. Esto se escribe en codificación ASCII (7 bits), por tanto es texto legible. El primer campo, *tiempo*, se escribe en formato *Mmmdd hh:mm:ss* donde *Mmm* son las iniciales del nombre del mes en inglés, *dd*, es el día del mes, y el resto es la hora. No se indica el año. Justo después viene el *nombre* del equipo (*hostname*), o la dirección IP si no se conoce el nombre. No puede contener espacios, ya que este campo acaba cuando se encuentra el siguiente espacio.

Lo que queda de paquete *syslog* al llenar la *prioridad* y la *cabecera* es el propio **texto** del mensaje. Éste incluirá información sobre el proceso que ha generado el aviso, normalmente al principio (en los primeros 32 caracteres) y acabado por un carácter no alfanumérico (como un espacio, "." o "/"). Después, viene el contenido real del mensaje, sin ningún carácter especial para marcar el final.

1.6.3 Principales Protocolos de Gestión

1.6.3.1 SNMP (Protocolo Simple de Gestión de Redes)

Simple Network Management Protocol (SNMP), es el protocolo de gestión de red más importante y usado en la actualidad. Forma parte del conjunto de protocolos TCP/IP y está definido en la capa de aplicación del mismo. SNMP busca la sencillez y es por ello que en la capa de transporte está soportado por el protocolo UDP (caracterizado por su rapidez).

La información que proporciona un dispositivo y los comandos que se pueden efectuar sobre él se definen en un lenguaje llamado **SMI** (*Structure of Management Information*, un subconjunto de ASN.1). Al árbol completo de la información estándar definida en SMI se le denomina MIB (*Management Information Base*), aunque coloquialmente se usa este nombre para referirse a los archivos definidos en SMI. SNMP permite a las aplicaciones de administración (*managers*) hacer consultas e incluso actualizaciones de los objetos definidos en el MIB. Estas peticiones son atendidas por los agentes SNMP que se ejecutan en los dispositivos de red. Este proceso permite la administración y monitorización remota de los dispositivos. Existen programas *manager* que periódicamente obtienen datos por SNMP de algún parámetro en especial, esto permite hacer gráficas de uso del sistema. Los agentes también pueden notificar algún evento detectado en el sistema a los *managers* mediante mensajes llamados *traps*.

➤ *Funcionamiento del protocolo - Tipos de PDU*

Como mecanismo básico y directo para intercambiar información de gestión entre un gestor y un agente se utilizan 7 tipos distintos de PDU (*Protocol Data Unit*). A continuación se muestran agrupados estos 7 PDUs según la naturaleza de sus emisores/receptores.

- Emitidos por el gestor:
 - *get request*. Lista de nombres de objetos para los que se solicita un valor.
 - *get next request*. Obtiene el siguiente objeto en orden lexicográfico permitiendo descubrir la estructura MIB dinámicamente.
 - *get bulk request*. Disponible a partir de la versión 2 de SNMP, tiene como objetivo minimizar el número de intercambios entre dos entidades.
 - *set request*. Sirve para modificar los valores de uno o más objetos de una MIB.
- Gestora gestor:
 - *inform request*. Se trata de una petición de información de gestión.
- Transmitidas por agente a gestor:
 - *response*. (es posible que un gestor en su papel de agente la use si responde a un *inform request*.)
 - *trap*. Para cuando ocurre un evento inusual, se trata de una notificación asíncrona unidireccional (no requiere respuesta).

SNMP es soportado por la mayoría de dispositivos de Internet working incluyendo switches, routers, servidores y estaciones de trabajo. Ha ganado su popularidad gracias a su simplicidad y por su facilidad de implementación, instalación y uso. La interoperabilidad entre implementaciones de SNMP de diferentes fabricantes es muy fácil debido a que es muy simple.

SNMPv3 ha suplantado gradualmente las versiones 1 y 2 porque ofrece mejor seguridad, incluyendo autenticación para protegerse contra modificaciones de la información y un set de operaciones seguras para configuración remota de los dispositivos SNMP gestionados.

SNMP es especificado en 3 grupos de documentos:

- RFC 2579 define mecanismos para describir y nombrar parámetros que son gestionados con SNMP. Los mecanismos son llamados la Estructura de la Información de Gestión o SMI (Structure of Management Information).
- RFC 1905 define los protocolos de operación para SNMP.
- Bases de Información de Gestión (MIBs – Management Information Bases): definen los parámetros de gestión que son accesibles vía SNMP. Varios RFCs definen MIBs de diferentes tipos. Los fabricantes pueden definir sus propios MIBs.

1.6.3.1.1 *Management Information Bases (MIBs)*

Una MIB almacena información recogida por un agente local de gestión o un dispositivo gestionado. Cada objeto en una MIB tiene un único identificador. Las aplicaciones de Gestión de Red usan el identificador para recuperar un objeto específico. La MIB es estructurada como un árbol. Objetos similares están agrupados bajo la misma rama del árbol MIB.

MIB II define los siguientes grupos de objetos gestionados para redes TCP/IP:

- El grupo Sistema
- El grupo Interfaz
- El grupo Traducción de Direcciones
- El grupo IP
- El grupo ICMP
- El grupo TCP
- El grupo UDP
- El grupo EGP
- El grupo de Transmisión
- El grupo SNMP

1.6.3.2 *RMON (Monitoreo Remoto)*

El RMON MIB (Remote Monitoring MIB) fue desarrollado por la IETF a comienzos de los 90's para corregir los defectos del estándar MIB, el cual era deficiente en su capacidad de proveer estadísticas de los enlaces de datos y parámetros en el nivel físico. La IETF originalmente desarrolló el RMON MIB para proporcionar estadísticas de tráfico Ethernet y diagnósticos de fallas.

Los agentes RMON recogen estadísticas en errores por Chequeo de Redundancia Cíclica (CRC), Colisiones Ethernet, errores Token Ring, distribución del tamaño del paquete, número de paquetes entrantes y salientes, y la tasa de paquetes broadcast. El grupo de Alarma RMON permite al administrador de la red fijar límites para los parámetros de la red y configurar agentes para entregar automáticamente alertas a una NMS (Sistema de Gestión de Red - Network Management System). RMON soporta también captura de paquetes (con filtros si se quiere) y envío de paquetes capturados a un NMS para análisis de protocolos.

RMON entrega la información en 9 grupos de parámetros. Los grupos para las redes Ethernet se muestran en la Tabla 4.7:

Tabla 1. Grupos Ethernet de RMON

GRUPOS ETHERNET DE RMON	
Grupo	Descripción
Estadísticas	Trayectorias de paquetes, octetos, distribución del tamaño de los paquetes, broadcast, colisiones, paquetes perdidos, fragmentos, errores por alineación CRC, y paquetes sobredimensionados muy pequeños.
Historia	Almacena múltiples valores de muestreo desde el grupo de Estadísticas para la comparación del comportamiento actual de una variable seleccionada con su rendimiento sobre un periodo de tiempo.
Alarmas	Habilita entornos límite e intervalos de muestreo en una estadística para crear una condición de alarma. Los valores límite pueden ser valores absolutos, ascendentes o descendentes, o valores delta.
Hosts	Proveen una tabla para cada nodo activo que incluye una variedad de estadísticas de nodo incluyendo paquetes y octetos entrantes y salientes, paquetes multicast y broadcast entrantes y salientes, y contadores de errores.
Host Top N	Extiende la tabla host para ofrecer un estudio definido por el usuario de estadísticas de host ordenados. Host top N es calculado localmente por el agente, para así reducir el tráfico de la red y el procesamiento en el NMS.
Matriz	Despliega la cantidad de tráfico y el número de errores ocurridos entre pares de nodos en un segmento.
Filtros	Permite a un usuario definir filtros de paquetes específicos y mantenerlos como mecanismo Starastop para la actividad de captura de paquetes.
Captura de Paquetes	Paquetes que pasan los filtros son capturados y almacenados para su posterior análisis. Un NMS puede requerir el buffer de captura y analizar los paquetes.
Eventos	Permite al usuario crear entradas en un monitor de logs o generar trampas SNMP desde el agente al NMS. Los eventos pueden ser iniciados por un límite cruzado en un contador o por un contador de paquetes.

RMON provee información a los administradores de la red sobre el rendimiento del segmento de red en el cual reside el agente RMON. Los beneficios de RMON son obvios, pero el ámbito de RMON versión 1 (RMON1) es limitado porque se enfoca en los parámetros de las capas física y de enlace de datos. La IETF está trabajando en el estándar RMON2 que se mueve entre segmentos de información para obtener información del rendimiento de las aplicaciones de la red en comunicaciones extremo a extremo. RMON2 se describe en el RFC 2021. La tabla 4.8 muestra los grupos en RMON2:

Tabla 2. Grupos RMON2

Grupos RMON2	
Grupo	Descripción

Directorio del Protocolo	Provee una lista de protocolos soportados por el dispositivo.
Distribución del Protocolo	Contiene estadísticas de tráfico para cada protocolo soportado.
Mapeo de Direcciones	Contiene el mapeo de direcciones de la capa de Red a la capa Física (MAC).
Host del nivel de Red	Contiene estadísticas para el tráfico del nivel de Red hacia o desde cada host.
Matriz del nivel de Red	Contiene las estadísticas de tráfico del nivel de Red para conversaciones entre pares de hosts.
Host del nivel de Aplicación	Contiene estadísticas para el tráfico del nivel de Aplicación hacia o desde cada host.
Matriz del nivel de Aplicación	Contiene las estadísticas de tráfico del nivel de Aplicación para conversaciones entre pares de hosts.
Colección de la historia del usuario	Contiene muestras periódicas de las variables específicas del usuario.
Prueba de Configuración	Provee un camino estándar para configurar remotamente parámetros de prueba tales como un destino trampa y gestión fuera de banda.