UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

ELO 322: REDES DE COMPUTADORES I

"TUNNELING PROTOCOL"

Proyecto	
Grupo	Byron Popper 2803050-9 Adrián Vasquez 2921010-1 Yen-kung Yu 2921063-2
Fecha	23/08/2013
Revisado por	**Nota**

1. Resumen:

La técnica de tunneling consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel dentro de una red de computadoras (internet),

2. Introducción:

El establecimiento del túnel se implementa incluyendo una PDU determinada dentro de otra PDU con el objetivo de transmitirla desde un extremo al otro del túnel sin que sea necesaria una interpretación intermedia de la PDU encapsulada. En esencia , es la transferencia de un paquete de información ,dentro de otro paquete que hace de "envoltorio". El protocolo del paquete que hace de envoltorio , solo es entendido por el emisor y por el receptor, en concreto, por el gateway que lo envía y por el gateway que lo recibe. De esta manera se encaminan los paquetes de datos sobre nodos intermedios que son incapaces de ver en claro el contenido de dichos paquetes. El túnel queda definido por los puntos extremos y el protocolo de comunicación empleado, que entre otros, podría ser SSH. ¿Para qué se utiliza? La técnica de tunelizar se suele utilizar para transportar un protocolo determinado a través de una red que en condiciones normales, no lo aceptaría. Otro uso de la tunelización de protocolos es la creación de diversos tipos de redes privadas virtuales. La técnica de tunelizar puede ser usada también para evitar o circunvalar un cortafuegos. Para ello, se encapsula el protocolo bloqueado en el cortafuegos dentro de otro permitido, habitualmente HTTP. Para hacer más fácil el entendimiento de ésta técnica ,imaginemos que un Tunneling es como un paquete enviado por una mensajería (ej.chilexpress). El que envía el paquete (protocolo pasajero) en una caja (protocolo de encapsulamiento) que se carga en la furgoneta de chilexpress (protocolo del carrier) y viaja por la autopista (Internet). El camión (protocolo carrier) viaja por la autopista (Internet) a casa del destinatario (salida del tunnel) y entrega la caja (protocolo de encapsulamiento). El destiantario abre la caja (protocolo de encapsulamiento) y saca el paquete (protocolo pasajero).

3. Protocolos Orientados a Datagramas

El Tunneling posee múltiples protocolos únicamente destinados a trabajar con datagramas que son: L2TP (Layer 2 Tunneling Protocol), MPLS (Multiprotocol Label Switching), GRE (Generic Routing Encapsulation), PPTP (Point-to-Point Tunneling Protocol), PPPoE (point-to-point protocol over Ethernet), PPPoA (point-to-point protocol over ATM), IPSec (Internet Protocol security), IEEE 802.1Q (Ethernet VLANs), DLSw (SNA over IP), XOT (X.25 datagrams over TCP), 6to4 (IPv6 over IPv4 as protocol 41) y Teredo. De los cuales uno de los más importantes es el **IPSec** que es el que pasaremos a desarrollar a continuación:

3.1. Internet Protocol Security (IPSec)

Su traducción es Seguridad IP y es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y cifrando cada paquete IP en un flujo de datos. Los protocolos de IPsec actúan en la capa de red. Otros protocolos de seguridad para Internet de uso extendido, como SSL, TLS y SSH operan de la capa de transporte hacia arriba. Esto hace que IPsec sea más flexible, ya que puede ser utilizado para proteger protocolos de la capa transporte, incluyendo TCP y UDP. Como el Protocolo de Internet no provee intrínsecamente de ninguna capacidad de seguridad, IPsec se introdujo para proporcionar servicios de seguridad tales como:

1. Cifrar el tráfico (de forma que no pueda ser leído por nadie más que las partes a las que está dirigido)
2. Validación de integridad (asegurar que el tráfico no ha sido modificado a lo largo de su trayecto)
3. Autenticar a los extremos (asegurar que el tráfico proviene de un extremo de confianza)
4. Anti-repetición (proteger contra la repetición de la sesión segura).

Así pues para lograr sus objetivos IPsec cuenta con dos protocolos importantes que han sido desarrollados para proporcionar seguridad a nivel de paquete:

- **Authentication Header (AH):** Es un encabezado para los paquetes IPSec diseñado para garantizar integridad, sin conexión y autenticación de los datos de origen de los datagramas IP. Una cabecera AH mide 32 bits.
- **Encapsulating Security Payload (ESP):** Parecido a AH, proporciona confidencialidad, autenticación y/o protección de integridad. Pero posee la diferencia de que soporta configuraciones de sólo cifrado, sólo autenticación o ambas, la última opción es la más utilizada ya que otorga mayor seguridad.

El protocolo IPSec posee una poderosa aplicación llamada **VPN** que detallaremos a continuación:

Virtual Private Network (VPN)

Su traducción es Red Privada Virtual y es una tecnología de red que permite la extensión de una red interna de una Organización o Empresa sobre una red pública o no controlada (red insegura), como por ejemplo Internet, otorgando al usuario los mismos privilegios y nivel de acceso a la información que tendría si estuviese dentro de la Organización (físicamente). Para llevar a cabo la implementación de esto se requiere de algún protocolo tunneling de alta seguridad, así como el IPSec, que nos brinda encriptación, autentificación, confidencialidad e integridad de los datos transmitidos a través de la red pública. Vale la pena mencionar otra forma de utilizar las redes VPN, que son las VPN internas: Consiste en establecer redes privadas virtuales dentro de una misma red local. El objetivo es aislar partes de la red y sus servicios entre sí, aumentando la seguridad.

4. Protocolos Orientados a Flujos

El Tunneling posee dos protocolos únicamente destinados a trabajar con flujos de datos que son: TLS (Transport Layer Security) y SSH (Secure Shell). De los cuales por su popularidad e por nuestro acercamiento se desarrollará el protocolo **SSH**:

4.1. SSH (Secure Shell):

SSH™ (o Secure SHell) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. El protocolo fue desarrollado 1995 por el finlandés Tatu Ylönen y a diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo. Además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

4.1.1. Características:

- **Encriptación de Datos:** SSH utiliza un método seguro de encriptación de 128 bits al enviar y recibir datos, lo cual hace que sea muy difícil descifrar y leer el contenido transmitido bajo una red no segura.
- **Operabilidad:** El uso de SSH es muy simple, sólo se realiza una conexión mediante una aplicación cliente hacia el servidor, una vez autentificado, el cliente toma completamente el control del servidor remoto.
- **Canalización:** Después de realizada la autenticación entre el cliente y el servidor remoto, se abren canales múltiples para cada sesión sea por consola de comandos ó interfaz gráfica X mediante el método llamado Multiplexación.
- **Reenvío por X11:** Mediante éste protocolo es posible el tráfico seguro de sesiones remotas bajo entornos gráficos X.
- **Redirección de Puertos:** Permite el envío de conexiones TCP / IP no seguras, mediante un canal seguro y cifrado. Cuando esto es realizado, el servidor SSH establece un túnel encriptado hacia el cliente SSH.
- **Copia y Transferencia de Archivos:** Con SCP y SFTP podemos realizar copiados y transferencias de archivos entre 2 computadoras y todo por medio de una canal seguro en una red insegura.

4.1.2. Funcionamiento:

El funcionamiento de éste protocolo se resume en los siguientes pasos:

1. El cliente inicia una conexión TCP sobre el puerto 22 del servicio
2. El cliente y el servidor se ponen de acuerdo en la versión de protocolo a utilizar, así como el algoritmo de cifrado utilizado para el intercambio de información.
3. El servidor, que tiene en su poder dos claves (una privada y otra pública), manda su clave pública al cliente.
4. Cuando el cliente recibe la clave enviada por el servidor, la compara con la que tiene almacenada para verificar su autenticidad. El protocolo SSH exige que el cliente la confirme la primera vez.
5. Con la clave pública del servidor en su poder, el cliente genera una clave de sesión aleatoria, creando un mensaje que contiene esa clave y el algoritmo seleccionado para la encriptación de la información. Toda esa información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada.
6. Si todo es correcto, el cliente queda autenticado, iniciando la sesión para comunicarse con el servidor.

4.1.3. SSH Tunneling:

El protocolo SSH (secure shell) se utiliza con frecuencia para tunelizar tráfico confidencial sobre Internet de una manera segura. Por ejemplo, un servidor de ficheros puede compartir archivos usando el protocolo SMB (Server Message Block), cuyos datos no viajan cifrados. Esto permitiría que una tercera parte, que tuviera acceso a la conexión (algo posible si las comunicaciones se realizan en Internet) pudiera examinar a conciencia el contenido de cada fichero trasmitido. Para poder montar el sistema de archivo de forma segura, se establece una conexión mediante un túnel SSH que encamina todo el tráfico SMB al servidor de archivos dentro de una conexión cifrada SSH. Aunque el protocolo SMB sigue siendo inseguro, al viajar dentro de una conexión cifrada se impide el acceso al mismo

5. Aplicación:

5.1. Uso de openSSH

La UTFSM está suscrita para acceder a material digital para sus alumnos y académicos. Sin embargo, solo se puede ingresar a estas bibiliotecas virtuales conectándose desde el campus.

- **Problema:** Estos sitios web, identifican la IP de la universidad lo cual indica que tiene permisos para acceder a sus contenidos de forma libre. Sin embargo, cuando se desea acceder desde fuera de la institución, se bloquea el material.

- **Solución:** Se utiliza Open-SSH, con el cual se conectará al servidor Aragorn del departamento de Electrónica, y con ello se podrá navegar por internet pasando la conección por la universidad y la IP será apta para acceder a las bibliotecas virtuales.
- **Funcionamiento:** Se conecta al servidor Aragorn, con el comando:

  ```
  ssh -D [puerto] user@aragorn.utfsm.cl
  ```

  Luego se configura el proxy del navegador, para este caso se utiliza Firefox:

  - Abrir el programa y dirigirse a Editar → Preferencias → Avanzado → Red → Configuraciones.

Fig. 1: Conección por SSH al servidor Aragorn.

- Seleccionar la configuración **manual de proxy**.

- En la casilla **Servidor SOCKS**, escribir **localhost** y el **puerto** que se haya utilizado para conectarse al servidor Aragorn.

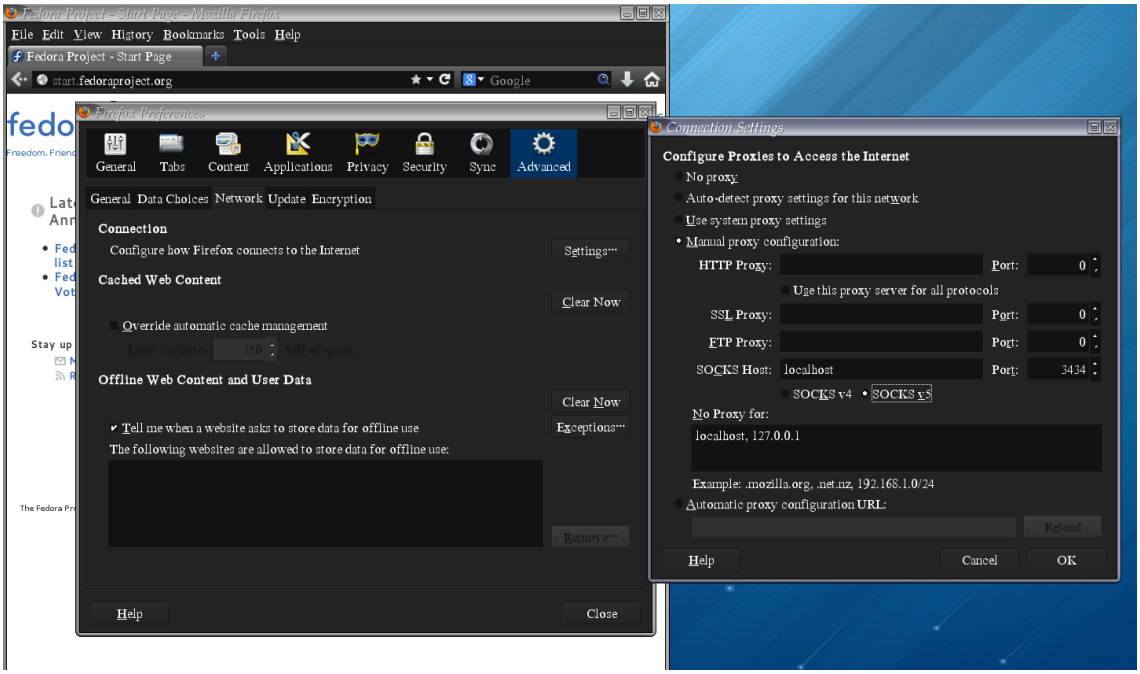

Fig. 2: Configuración de Firefox

- **Resultado:** Se puede verficar la correcta conección al servidor desde el navegador con la IP, para esto se ingresa a la pag.: **http://redes.dcsc.utfsm.cl/cualesmiip.html**

5.2. Uso de HTTP Tunneling:

Hay situaciones en que se está situado en una red controlada por un Proxy, lo cual limita las conecciones a ciertas servidores exteriores. Estos casos pueden ser en una red empresarial o en alguna institución.

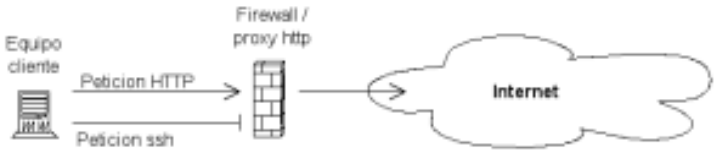

Fig. 3: HTTP Tunneling v/s SSH directo

- **Problema:** Acceder a servidores externos públicos a través de SSH, sin embargo el proxy bloquea este tipo de conecciones.
- **Solución:** Se implementará HTTP Tunneling, lo cual permite camuflar o encapsular la conección SSH, haciendola pasar por una conección ”normal”HTTP y evadir el proxy.
- **Funcionamiento:** Existen varias alternativas de programas (Windows y linux), ya que es open-source.

  Instalación, para el caso en linux:

```
wget http://www.nocrew.org/software/httptunnel/httptunnel-3.0.5.tar.gz
tar xvzf httptunnel-3.0.5.tar.gz
./configure
make
```

  Se ejecuta en el **Servidor**:

```
hts -w -F localhost:[PORT_SSH] [PORT_HTTP]
```

  Se ejecuta en el **Cliente**:

```
htc -P [IP_PROXY] -F [PORT_LOCAL] [IP_SERVER]:[PORT_SERVER]
```

  Luego, se realiza normalmente una conección SSH al servidor deseado.

Fig. 4: Esquema funcionamiento de SSH con HTTP Tunneling.

6. Conclusión:

El tunneling es una herramienta de uso amplio en el informática, que se usa tanto para crear seguridad como para violar tal seguridad y como tal queda mucho camino para seguir investigando y creando protocolos tunneling.

Ya que es posible evadir firewall, es importante tener un criterio formado para no realizar malas prácticas para aprovecharse de estas utilidades que ofrece el tunneling.

7. Referencias:

- $http://profesores.elo.utfsm.cl/\ agv/elo322/1s10/.../Enalece.web.remoto.a.traves.de.SSh$
- $http://debianitas.net/doc/minicomos/httptunel/html/httptunnel.html$
- $http://es.wikipedia.org/wiki/Tunneling$