Proyecto Redes de Computadores:

“Túnel y la encapsulación de un protocolo”

Integrantes:

Antonio Cayulao 201030018-8

Mario Hazard 201004502-1

Humberto Gonzalez 201004155-7

Fecha: lunes 30, 2012

Universidad Técnica Federico Santa María

Resumen

La explosión de IPv6, los costos asociados al mantenimiento de una red y el control remoto de una máquina, son parte de las materias que toca el proceso de tunelización, a lo largo de este trabajo buscamos dar el apropiado enfoque y presentación de los datos de modo tal, todo lector pueda entender sin mayores dificultades lo extenso del *tunneling.* Podemos agregar que elegimos este tema, la tunelización, ya que en el desarrollo del curso no hubo una gran profundización al respecto, tocándose brevemente para la implementación de IPv6, tal que al indagar un poco mas pudimos ver como se nos abría un mundo de posibilidades que mediante este proceso, eran alcanzables, dando este trabajo como resultado, una breve introducción al mundo de la tunelización, al tocar temas como la implementación de una VPN, 6 to 4, flujo cifrado de datos, para finalizar con una breve explicación sobre la vulneración de un firewall usando el protocolo y aplicación ssh.

¿Qué es un túnel?

En palabras simples es la utilización de protocolos de red para el encapsulamiento de datos en otro protocolo, transformándolo para así poder llevar las solicitudes a destino, pudiendo presenciar (revisar caso IEEE). Un determinado protocolo tiene formatos de acceso determinados, al efectuar un túnel generamos código adicional que reemplaza el antiguo encabezado, por uno temporal en el formato del protocolo que queremos traspasar y generando un subdestino, el antiguo encabezado junto a los datos solicitados, son considerados en conjunto como los datos a enviar, luego al llegar al subdestino, los datos se desencapsulan (destunelizan) continuando el paquete con su destino original y ruta, por el protocolo previo a realizar el túnel.

Los túneles se implementaron principalmente para solucionar los conflictos de compatibilidad en la transmisión de datos entre diferentes protocolos, donde destacan dos tipos, los orientados a datagramas como MPLS (*Multiprotocol Label Switching*), L2TP (*Layer 2 Tunneling Protocol*), IEEE 802.1Q (*Ethernet VLANs*), y 6to4 -que trata de la migración de IPv4 a IPv6-, y los orientados a flujos como TLS (*Transport Layer Security*) y SSH (*Secure Shell)*.

Protocolos orientados a datagramas

Primero describiremos que un datagrama es un fragmento de paquete que es enviado con la suficiente información para que la red pueda simplemente encaminar el fragmento hacia el equipo terminal de datos receptor, de manera independiente a los fragmentos restantes. Esto puede provocar una recomposición desordenada o incompleta del paquete en el ETD destino.

Multiprotocol Label Switching

MPLS (siglas de *Multiprotocol Label Switching*) es un mecanismo de transporte de datos estándar. Opera entre la capa de enlace de datos y la capa de red del modelo OSI. Fue diseñado para unificar el servicio de transporte de datos para las redes basadas en circuitos y las basadas en paquetes. Puede ser utilizado para transportar diferentes tipos de tráfico, incluyendo tráfico de voz y de paquetes IP.

MPLS el son circuitos virtuales en las redes IP, sobre las que introduce una serie de mejoras:

- Redes privadas virtuales.
  - es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada.
- Ingeniería de tráfico.
  - Se refiere al ámbito de las telecomunicaciones para planificar, diseñar, proyectar, dimensionar, desarrollar y supervisar redes de telecomunicaciones en condiciones óptimas de acuerdo a la demanda de servicios, márgenes de beneficios de la explotación, calidad de la prestación y entorno regulatorio y comercial.
- Mecanismos de protección frente a fallos.
  - A través del encabezado que no requiere revisar el interior del mensaje

La tecnología MPLS ofrece un servicio orientado a conexión:

- Mantiene un «estado» de la comunicación entre dos nodos.
- Mantiene circuitos virtuales

MPLS funciona anexando un encabezado a cada paquete. Dicho encabezado contiene una o más "etiquetas", y al conjunto de etiquetas se le llama pila o "stack". Cada etiqueta consiste en cuatro campos:

Donde:

- Label (20 bits): Es la identificación de la etiqueta.
- Exp (3 bits): Llamado también bits experimentales, también aparece como QoS en otros textos, afecta al encolado y descarte de paquetes.
- S (1 bit): Del inglés *stack*, sirve para el apilado jerárquico de etiquetas. Cuando S=0 indica que hay más etiquetas añadidas al paquete. Cuando S=1 estamos en el fondo de la jerarquía.
- TTL (8 bits): Time-to-Live, misma funcionalidad que en IP, se decrementa en cada enrutador y al llegar al valor de 0, el paquete es descartado. Generalmente sustituye el campo TTL de la cabecera IP.

MLPS en la creación de la red privada

En el contexto de las Redes Privadas Virtuales, los enrutadores que funcionan como ingreso o regreso a la red son frecuentemente llamados enrutadores a la Orilla del Proveedor (enrutadores PE), los dispositivos que sirven solo de tránsito son llamados similarmente enrutadores de Proveedor (enrutadores P).

Paso de un paquete por una red

Cuando un paquete no etiquetado entra a un enrutador de ingreso y necesita utilizar un túnel MPLS, el enrutador primero determinará la Clase Equivalente de Envío (FEC), luego inserta una o más etiquetas en el encabezado MPLS recién creado. Acto seguido el paquete salta al enrutador siguiente según lo indica el túnel.

Cuando un paquete etiquetado es recibido por un enrutador MPLS, la etiqueta que se encuentra en el tope de la pila será examinada. Basado en el contenido de la etiqueta el enrutador efectuará una operación apilar (PUSH), desapilar (POP) o intercambiar (SWAP).

- En una operación SWAP la etiqueta es cambiada por otra y el paquete es enviado en el camino asociado a esta nueva etiqueta.
- En una operación PUSH una nueva etiqueta es empujada encima de otra (si existe). Si en efecto había otra etiqueta antes de efectuar esta operación, la nueva etiqueta «encapsula» la anterior.
- En una operación POP la etiqueta es retirada del paquete lo cual puede revelar una etiqueta interior (si existe). A este proceso se lo llama «desencapsulado» y es usualmente efectuada por el enrutador de egreso con la excepción de PHP.

Durante estas operaciones el contenido del paquete por debajo de la etiqueta MPLS no es examinado, de hecho los enrutadores de tránsito usualmente no necesitan examinar ninguna información por debajo de la mencionada etiqueta. El paquete es enviado basándose en el contenido de su etiqueta, lo cual permite «ruteado independiente del protocolo».

En el enrutador de egreso donde la última etiqueta es retirada, sólo queda la «carga transportada», que puede ser un paquete IP o cualquier otro protocolo. Por tanto, el enrutador de egreso debe forzosamente tener información de ruteo para dicho paquete debido a que la información para el envío de la carga no se encuentra en la tabla de etiquetas MPLS.

En algunas aplicaciones es posible que el paquete presentado al LER ya contenga una etiqueta MPLS, en cuyo caso simplemente se anexará otra etiqueta encima. Un aspecto relacionado que resulta importante es PHP.

En ciertos casos, es posible que la última etiqueta sea retirada en el penúltimo salto (anterior al último enrutador que pertenece a la red MPLS); este procedimiento es llamado «remoción en el penúltimo salto» (PHP). Esto es útil, por ejemplo, cuando la red MPLS transporta mucho tráfico. En estas condiciones los penúltimos nodos auxiliarán al último en el procesamiento de la última etiqueta de manera que este no se vea excesivamente forzado al cumplir con sus tareas de procesamiento.

IPv6 sobre IPv4

Cuando un paquete IPv6 tiene que ser transportado a través de una red que sólo es IPv4, pueden utilizarse túneles para lograrlo. Un túnel trabaja encapsulando un paquete IPv6 dentro de un paquete IPv4 para que el mismo pueda viajar por estas redes. El paquete es a su vez des encapsulado al llegar al destino, que deberá ser un nodo IPv6 o doble pila.

Túneles IPv6 sobre IPv4

Los túneles proporcionan un mecanismo para utilizar la infraestructura montada bajo IPv4 mientras avance la transición hacia el pleno soporte de IPv6. Este mecanismo consiste en el encapsulamiento de datagramas IPv6 en paquetes IPv4. Los extremos finales del túnel siempre son los responsables de realizar la operación de encapsulado y/ó des encapsulado de paquetes.
Los túneles pueden ser utilizados de formas diferentes:

Router a router: Routers con doble pila (IPv6/IPv4) se conectan mediante una infraestructura IPv4 y transmiten tráfico IPv6. El túnel comprende un segmento que incluye la ruta completa, extremo a extremo, que siguen los paquetes IPv6.

Host a router. Hosts de doble pila se conectan a un router intermedio (también de doble pila), alcanzable mediante una infraestructura IPv4. El túnel comprende el primer segmento de la ruta seguida por los paquetes.

Protocolos orientados a flujo

Secure Shell

SSH (Secure SHell, en español: intérprete de comandos seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.

Además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), 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.

SSH trabaja de forma similar a como se hace con telnet. La diferencia principal es que 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; aunque es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre destinos.

Por ejemplo, para conectar con un servidor web de forma segura, utilizando SSH, haríamos que el cliente web, en vez de conectarse al servidor directamente, se conecte a un cliente SSH. El cliente SSH se conectaría con el servidor tunelizado, el cual a su vez se conectaría con el servidor web final. Lo atractivo de este sistema es que hemos añadido una capa de cifrado sin necesidad de alterar ni el cliente ni el servidor web.

Historia SSH

Al principio sólo existían los r-commands, que eran los basados en el programa rlogin, el cual funciona de una forma similar a telnet.

La primera versión del protocolo y el programa eran libres y los creó un finlandés llamado Tatu Ylönen, pero su licencia fue cambiando y terminó apareciendo la compañía SSH Communications Security, que lo ofrecía gratuitamente para uso doméstico y académico, pero exigía el pago a otras empresas. En el año 1997 (dos años después de que se creara la primera versión) se propuso como borrador en la IETF.

A principios de 1999 se empezó a escribir una versión que se convertiría en la implementación libre por excelencia, la de OpenBSD, llamada OpenSSH.

Versiones SSH

Existen 2 versiones de SSH, la versión 1 de SSH hace uso de muchos algoritmos de cifrado patentados (sin embargo, algunas de estas patentes han expirado) y es vulnerable a un hueco de seguridad que potencialmente permite a un intruso insertar datos en la corriente de

comunicación. La suite OpenSSH bajo Red Hat Enterprise Linux utiliza por defecto la versión 2 de SSH, la cual tiene un algoritmo de intercambio de llaves mejorado que no es vulnerable al hueco de seguridad en la versión 1. Sin embargo, la suite OpenSSH también soporta las conexiones de la versión 1.

Tunelizar para evitar un Cortafuegos

La técnica de tunelizar puede ser usada también para evitar o circunvalar un cortafuego. Para ello, se encapsula el protocolo bloqueado en el cortafuego dentro de otro permitido, habitualmente HTTP. Por ejemplo, en consola de comandos la orden seria lo siguiente, si lo que quisiéramos fuese entrar a la red de tareas de informática, la cual solo permite acceso desde dentro de la red del DI:

$ ssh -L 9999:tareas:22 ssh2.inf.utfsm.cl -l nombreusuario

Aquí ssh es nuestra aplicación y protocolo, la opción -L es lo más potente, pues lo que nos está diciendo es, crea en el puerto 9999 en nuestro ordenador de origen de la solicitud, tal que todo lo que llegue se almacene ahí, luego viene “tareas”, que es el nombre del servidor al cual estamos accediendo, como en este caso tiene nombre es que colocamos este para su acceso, de no tener, simplemente escribimos la IP del servidor, finalmente 22 nos señala el puerto que accederemos finalmente en nuestra maquina final, siendo nuestra maquina final tareas, entonces 22 es el puerto en esta máquina final en la cual nos conectaremos. Lo siguiente, ssh2.inf.utfsm.cl viene siendo el servidor que usamos para romper la seguridad, tiene que ser un servidor accesible tanto desde mi maquina como desde la maquina final, por lo tanto representa la maquina mediadora, que nos permitirá hacer todo el proceso de tuberización. El -l es una opción que nos permite acceder con nuestro nombreusuario a la maquina mediadora, es gracias a esta opción que nos podemos logear y empezar a trabajar remotamente en los servidores, simulando a nuestra posición como si estuviéramos físicamente escribiendo en la consola de la maquina remota.

Conclusión

En definitiva, hemos podido apreciar la gran cantidad de usos que tiene la tunelización, desde la creación de VPN’s hasta la vulneración de firewall, si bien hay variaciones entre las formas de implementación en general mantienen la estructura de encapsular datos en un protocolo, dentro de otro, hecho que queda especialmente reflejado cuando se trabaja con el protocolo ssh, tanto en la actualidad como a futuro, no existe un desarrollo en túneles propiamente, si no que son usados como herramienta para salir al paso de problemas, como en la actualidad se ha generado con la migración a IPv6 desde IPv4, principal tema exponente de túneles de entre todos, el cual presenta a la tunelización como su carta bajo la manga para solucionar este problema.

Otra parte del desarrollo lo lleva ssh bajo la dirección de OpenSSH, siendo este el principal protocolo de tunelización, podemos decir que al ser de código abierto y libre, siempre estará en constantes cambios, sobre todo la parte del staff que se encarga de hacer compatible la aplicación con todas las plataformas existentes.