Laboratorio de Evasión de Firewall: Bypasseando Firewalls usando VPN

1 Descripción

Organizaiones, Internet Service Providers (ISPs) y paises a menudo bloquean el acceso a determinados sitios externos a sus usuarios internos. Esto es llamado filtrado de salida o egress filtering. Por ejemplo, para prevenir distracción en horarios laborales, muchas companias configuran la salida de sus firewalls para bloquear sitios de redes sociales, por lo que sus empleados no podrán acceder a estos dentro de la red interna. Por razones políticas, muchos países configurar filtrados de salida en sus ISPs para bloquear a su gente el acceso a determinados sitios foráneos. Desafortunadamente, estos firewalls pueden ser fácilmente bypasseados y existen servicios/productos que ayudan a los usuarios bypassear estos firewalls, estas soluciones están a la alcance de todos. La tecnología más usada para bypassear estos filtrados de salida son las Virtual Private Network (VPN). Esta tecnología es ampliamente usada por usuarios que poseen smartphones y que son afectados por este tipo de bloqueo; existen muchas aplicaciones VPN (para Android, iOs y otras plataformas) que ayudan a los usuarios a evadir estas reglas de filtrado que se aplican en los firewalls.

El objetivo de este laboratorio es que los estudiantes vean como funciona una VPN y como una VPN puede ayudar a bypassear los filtrados de salida de un firewall. En este Laboratorio, implementaremos una VPN muy simple que usaremos para bypassear firewalls. Una VPN típica depende de dos piezas: IP tunneling y el cifrado. El tunneling es esencial ya que será quien nos ayude a byppasear firewalls; el cifrado es para proteger el contenido del tráficos que viaja a través del tunel de la VPN. Por un tema de simplicidad, en este laboratorio solamente nos centramos en el tunneling, por lo que el tráfico dentro de nuestro tunel no estará cifrado. Tenemos otro laboratorio sobre VPN en donde cubrimos tunneling y cifrado. Si los lectores están interesados, pueden trabajar sobre este laboratorio para aprender como construir una VPN completa. En este Laboratorio solamente nos centramos en como usar un tunel VPN para bypassear firewalls.

Este laboratorio cubre los siguientes tópicos:

- Firewall
- VPN

Lectura y Videos. Para una cobertura más detallada sobre firewalls, técnicas de evasión y VPN puede consultar:

- Capítulos 17 y 19 del libro de SEED, *Computer & Internet Security: A Hands-on Approach*, 2nd Edition, by Wenliang Du. Para más detalles `https://www.handsonsecurity.net`.
- Secciones 8 y 9 del curso de SEED en Udemy, *Internet Security: A Hands-on Approach*, by Wenliang Du. Para más detalles `https://www.handsonsecurity.net/video.html`.

Entorno de Laboratorio. Este laboratorio ha sido testeado en nuestra imagen pre-compilada de una VM con Ubuntu 20.04, que puede ser descargada del sitio oficial de SEED .

2 Tareas del Laboratorio

2.1 Tarea 1: Setup de la Máquina Virtual

Necesitamos dos máquinas, una dentro del firewall y la otra fuera del firewall. El objetivo es ayudar a la máquina que se encuentra dentro del firewall que pueda visitar sitios bloqueados por el firewall. Las máquinas que usaremos serán, VM1 y VM2, estas máquinas se suponen conectadas a internet a través de routers. Este setup puede requiera más de dos Máquinas Virtuales. Para hacer todo más sencillo, usaremos una LAN para emular la conexión a Internet. Básicamente, conectaremos VM1 y VM2 a la LAN usando un adaptor de `NAT Network`. La Figura describe el setup de nuestro laboratorio.

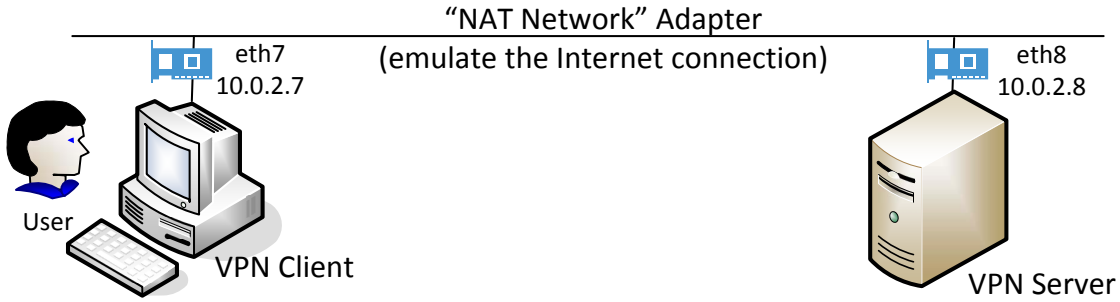

Figure 1: Setup del Laboratorio

2.2 Tarea 2: Setup del Firewall

En esta tarea, ud. hara el setup del firewall en la VM1 para bloquear el acceso a un sitio web. Debe asegurarse que la dirección IP de este sitio web sea fija o este dentro de un rango fijo; de otra forma puede terminar bloqueando por completo el sitio web. Por favor vea el laboratorio de Firewall para detalles en como bloquear sitios web.

En el mundo real, un firewall debería de correr en una máquina separada, no en VM1. Para minimizar el números de VMs usadas en el laboratorio, pondremos el firewall en VM1. Configurar un firewall en VM1 requiere de privilegios de superusuario como así también lo requiere la configuración del tunel VPN. Uno podría decir si contamos con privilegios de superusuario, porque no podemos desactivar el firewall en VM1 y listo. Este es un buen argumento, pero tenga en cuenta, que estamos poniendo el firewall en VM1 porque no queremos crear otra VM en el entorno de laboratorio. Además, aunque cuente con privilegios de superusuario en VM1, no le es permitido usar ese privilegio para reconfigurar el firewall. Ud. deberá de usar una VPN para bypassearlo.

A diferencia de situar el firewall en una máquina externa, poner el firewall en VM1 tiene un pequeño inconveniente con el cual necesitamos convivir. Cuando hacemos el setup del firewall para bloquear paquetes, necesitamos asegurarnos de no bloquear los paquetes que vienen de la interfaz virtual usada por la VPN o nuestra VPN no será capaz de obtener los paquetes. Además, no podemos setear la regla del firewall antes del routing o setearla en la interfaz virtual. Necesitamos setear la regla en la interfaz de red real de VM1, para no afectar a los paquetes que van hacia la interfaz virtual. El siguiente comando bloquea todo el tráfico hacia la red `93.184.216.0/24` (`example.com`).

```
$ sudo iptables -A OUTPUT -o enp0s3 -d 93.184.216.0/24 -j DROP
```

Por favor identifique el sitio web que quiere bloquear, haga el setup del firewall y demuestre que su firewall está funcionando y la dirección IP que esta bloqueando no se puede alcanzar. Provea capturas de pantalla en su informe del laboratorio.

2.3 Tarea 3: Bypasseando el Firewall usando VPN

La idea de usar una VPN para bypassear un firewall es ilustrada en la Figura 2. Establecemos un tunel VPN entre VM1 (Cliente VPN) y VM2 (Servidor VPN). Cuando un usuario trata de acceder al sitio bloqueado, el tráfico no viajará directo a través del adaptador de red, porque será bloqueado. En vez de esto, los paquetes que se dirigen al sitio bloqueado desde VM1 serán enrutados hacia el tunel VPN y llegarán a VM2. Una vez que estos llegan ahí, VM2 los enrutará hacia su destino final. Cuando la respuestas de los paquetes lleguen de regreso, regresarán hacia VM2, que redireccionará los paquetes hacia el tunel VPN y eventualmente los paquetes volverán a VM1. Así es como una VPN ayuda a bypassear firewalls.

The idea of using VPN to bypass firewall is depicted in Figure 2. We establish a VPN tunnel between VM1 (VPN Client VM) and VM2 (VPN Server VM). When a user on VM1 tries to access a blocked site, the traffic will not directly go through its network adapter, because it will be blocked. Instead, the packets to the blocked site from VM1 will be routed to the VPN tunnel and arrive at VM2. Once they arrive there, VM2 will route them to the final destination. When the reply packets come back, it will come back to VM2, which will then redirect the packets to the VPN tunnel, and eventually get the packet back to VM1. That is how the VPN helps VM1 to bypass firewalls.

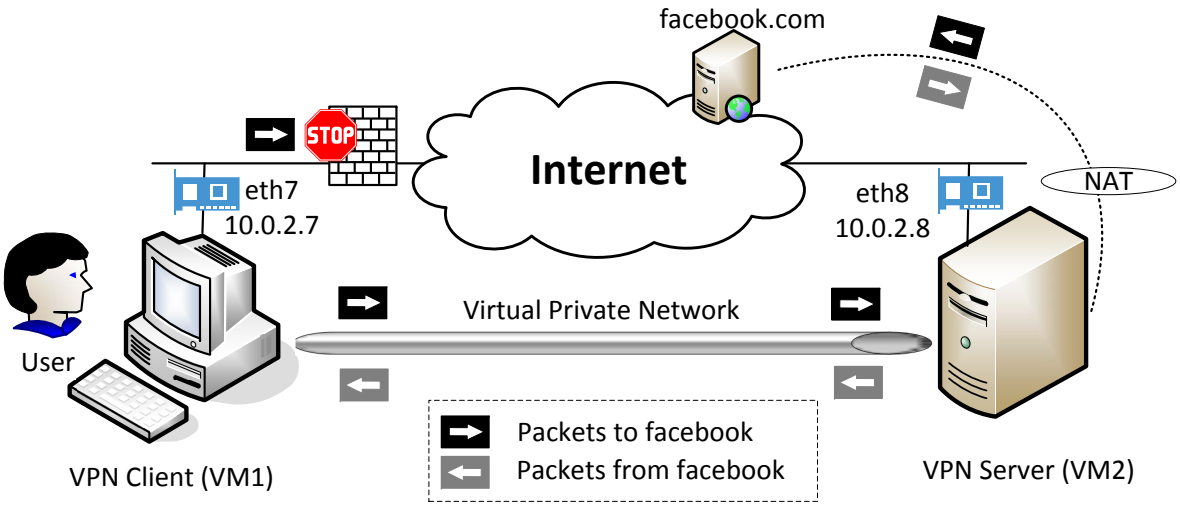

Figure 2: Bypasseando Firewall usando una VPN

Hemos creado un programa VPN de ejemplo, incluídos un programa cliente (`vpnclient`) y un programa servidor (`vpnserver`), ambos pueden ser descargados del sitio oficial del laboratorio. Estos programas VPN son muy simples y solamente establecen un tunel VPN entre el cliente y el servidor; no cifra el tráfico del tunel. El programa es explicado en detalle en el libro de SEED (Capítulo VPN).

Los programas `vpnclient` y `vpnserver` son los dos puntos finales del tunel VPN. Ellos se comunican entre sí por medio TCP o UDP a través de sockets, esto es ilustrado en la Figura 3. En nuestro código de ejemplo, elegimos usar UDP por un tema de simplicidad. La línea punteada entre el cliente y el servidor demarca el camino para el tunel VPN. Los programas del cliente y servidor VPN se conectan al sistema de hosting a través de la interfaz TUN, a través de la cual ellos hacen dos cosas: (1) obtienes los paquetes IP del

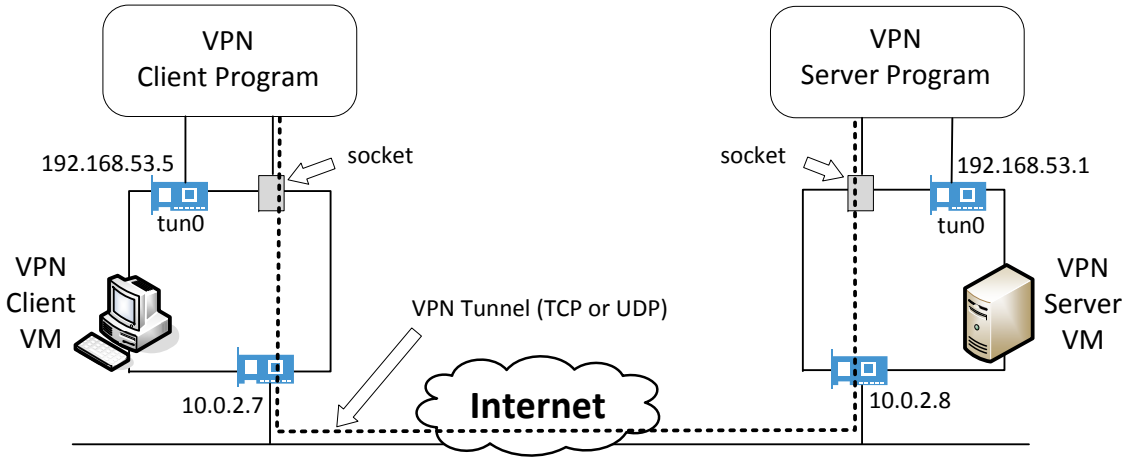

Figure 3: Cliente y Servidor VPN

sistema de hosting, por lo que los paquetes pueden ser enviados a través del tunel, (2) obtiene los paquetes IP desde el tunel y los forwardea hacia el sistema de hosting, el cual forwardeará los paquetes hacia su destino final. El siguiente procedimiento describe como crear un tunel VPN usando los programas `vpnclient` y `vpnserver`.

Paso 1: Iniciar el Servidor VPN. Primer iniciamos el servidor VPN `vpnserver` en la máquina virtual del servidor. Después de que el programa se ejecute, aparecerá una interfaz virtual de red TUN en el sistema (puede verla usando el comando "`ifconfig -a`"; en la mayoría de los sistema el nombre de esta interfaz será `tun0` pero puede ser que sea diferente, sea como sea será `tunX` donde `X` es un número). Esta nueva interfaz no está configurada todavía, por lo que necesitamos hacerlo, asignándole una dirección IP. Usaremos `192.168.53.1` para esta interfaz, pero puede usar otra IP.

Ejecute los siguientes comandos. El primer comando iniciará el programa servidor, el segundo comando asignará la dirección IP a la interfaz `tun0` y la activará. Debería de notar que el primer comando bloqueará y se quedará a la espera por conexiones, por lo que debemos de correr el segundo comando en una nueva ventana.

```
$ sudo ./vpnserver

Run the following command in another window:
$ sudo ifconfig tun0 192.168.53.1/24 up
```

Al menpos que se configura de una forma especial, una máquina actuará como host y no como gateway. El servidor VPN necesita forwardear los paquetes hacia otros destinos, por lo que necesita funcionar como un gateway. Necesitamos activar el IP forwarding para que la máquina se comporte como un gateway. IP Forwarding puede ser activado usando el siguiente comando:

```
$ sudo sysctl net.ipv4.ip_forward=1
```

Paso 2: Iniciar el Cliente VPN. Procederemos a ejecutar el cliente VPN en la máquina virtual cliente. Correremos el siguiente comando en la máquina. El primer comando se encargará de conectar al cliente con el servidor VPN que se encuentra corriendo en `10.0.2.8`. Este comando será bloqueante por lo que

necesitamos abrir otra ventana para configurar la interfaz `tun0` creada por el cliente VPN. A la interfaz `tun0` le asignaremos la IP `192.168.53.5` (Puede seleccionar otra IP si desea).

```
On VPN Client VM:
$ sudo ./vpnclient 10.0.2.8

Run the following command in a different window
$ sudo ifconfig tun0 192.168.53.5/24 up
```

Paso 3: Configurando el Routing en las Máquinas Cliente y Servidor. Después de realizar los pasos anteriores, el tunel será establecido. Antes de poder usar el tunel, necesitamos configurar los caminos de enrutamiento en las máquinas del cliente y del servidor para así poder direccionar el tráfico que nos interesa a través del tunel. Podemos usar el comando `route` para agregar entradas. El siguiente ejemplo muestra como enrutar los paquetes de `10.20.30.0/24` hacia la interfaz `eth0`.

```
$ sudo route add -net 10.20.30.0/24 eth0
```

Para bypassear los firewwalls en la máquina virtual del cliente, necesita configurar las entradas de enrutamiento que corresponden, de esta forma el tráfico que se dirige hacia el sitio bloqueado será enrutado hacia la VPN. Necesita pensar que entradas de enrutamiento agregar para bypassear el firewall.

Paso 4: Setup del NAT en la Máquina Virtual del Servidor. Cuando el destino final envía los paquetes de regreso a los usuarios, el paquete será enviado hacia el servidor VPN primero (piense el porque y escriba su respuesta en el informe del laboratorio). El retorno del paquete llegará al adaptador NAT del servidor VPN en primera instancia,porque las direcciones de IPs de origen de todo los paquetes salientes de la máquina virtual del servidor se cambian a la dirección IP externa del NAT (que es básicamente la IP de la máquina host de nuestro setup). Usualmente, el NAT reemplazará la IP de destino con la IP del paquete original (en nuestro caso `192.168.53.5`) y se la devolverá a quien sea el dueño de esa dirección IP. Desafortunadamente, aquí tenemos un problema.

Antes que la NAT envíe el paquete, necesita conocer la dirección MAC de la máquina que es la dueña de la dirección IP `192.168.53.5`, por lo que enviará una petición ARP. Nuestra red privada es virtual y esta dirección IP pertenece a la interfaz `tun0` del cliente VPN. Además `192.168.53.5` no recibirá la petición ARP (incluso si esto pasará, no serviría de nada). El NAT rechazará el paquete, porque el receptor no existe.

El receptor actual debería de ser la máquina virtual del servidor VPN, aunque no sea dueña de la IP `192.168.53.5`. Si podemos configurar el NAT como un gateway, podemos pedirle a el NAT que enrute los paquetes para `192.168.53.5` hacia el servidor VPN, el cual eventualmente entregará los paquetes a través del tunel al cliente VPN. Sin embargo, no hemos encontrado la forma de como configurar el NAT como un gateway en VirtualBox, hemos trabajado en dos soluciones. Una idea es "engañar" a la NAT haciendole creer que la dirección MAC de `192.168.53.5` la máquina del servidor VPN, por lo que el paquete será entregado al servidor VPN por el NAT. Podemos lograr esto usando ARP cache poisoning en la NAT, básicamente haciéndole saber a la NAT de antemano sobre la dirección MAC de `192.168.53.5`.

Una mejor solución para esta limitación del NAT es crear otra NAT en la máquina virtual del servidor, por lo que todos los paquetes que salen del servidor tendrán la dirección IP de la máquina virtual como dirección IP de origen. Para salir a la Internet, estos paquetes viajarán a través de la otra NAT, que es provista por VirtualBox, pero dado que la IP de origen es la máquina virtual del servidor, la segunda NAT no tendrá problemas para retransmitir los paquetes devueltos desde la Internet hacia la máquina virtual del servidor. Usando esta solución, no tenemos la necesidad de usar ARP cache poisoning para "engañar" a la

NAT. Los siguientes comandos pueden ser usados para activar la NAT en la máquina virtual del servidor (en su caso, el nombre del adaptador `NAT Network` puede que no se llame `enp0s3`; sólo necesita encontrar el verdadero nombre en su máquina virtual):

```
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3
```

Demostración. Si ud. ha realizado los pasos anteriores de manera correcta, debería de poder bypassear el firewall. Debería de mostrar que puede acceder al sitio bloqueado desde la máquina virtual del cliente a través de la VPN. Su solución no solamente debería de funcionar para tráfico web sino para todo tipo de tráfico. Por ejemplo, si la máquina bloqueada hace `telnet` al servidor bloqueado, debería de poder establecer una conexión `telnet` desde la máquina cliente.

En su informe del laboratorio, debe de proporcionar evidencia que muestre que su tráfico pasó por el tunel VPN y no por "puiertas laterales". La mejor forma de mostrarlo es capturar el tráfico de red usando Wireshark y describir el camino de sus paquetes usando el tráfico capturado. Sin esta evidencia, no tendremos idea si su éxito se debió a un firewall mal configurado (es decir el sitio web que se supone bloqueado no lo está) o por una VPN mal configurada.

3 Informe del Laboratorio

Debe enviar un informe de laboratorio detallado, con capturas de pantalla, para describir lo que ha hecho y lo que ha observado. También debe proporcionar una explicación a las observaciones que sean interesantes o sorprendentes. Enumere también los fragmentos de código más importantes seguidos de una explicación. No recibirán créditos aquellos fragmentos de códigos que no sean explicados.

Agradecimientos

Este documento ha sido traducido al Español por Facundo Fontana