Firewall Evasion Lab: Bypassing Firewalls using VPN

Table of Contents:

Lab Setup:

The Firewall Evasion Lab requires two Ubuntu 16.04 Virtual Machines (VMs)
VM1: VPN Client (IP: 10.0.2.15)
VM2: VPN Server (IP: 10.0.2.6)

Overview:

Organizations, Internet Service Providers (ISPs), and countries often block their internal users from accessing certain external sites. This is called **egress filtering**. For example, to prevent work-time distraction, many companies set up their egress firewalls to block social network sites, so their employee cannot access those sites from inside their network. The The most commonly used technology to bypass egress firewalls is **Virtual Private Network (VPN)**.
The learning objective of this lab is for students to see how VPN works in action and how VPN can help bypass egress firewalls. We will implement a very simple VPN in this lab, and use it to bypass firewalls.

Lab Tasks:

Task 1: VM Setup

As shown in the lab setup with two VMs with Ubuntu 16.04 Operating System. They both need to have the “NAT Network” Adapter enabled on them. VM1 is inside the firewall and VM2 is outside the firewall. The objective is to help the machine inside the firewall to reach out to the external sites blocked by the firewall. Provide a screenshot of the basic set-up.

Task 2: Set up Firewall

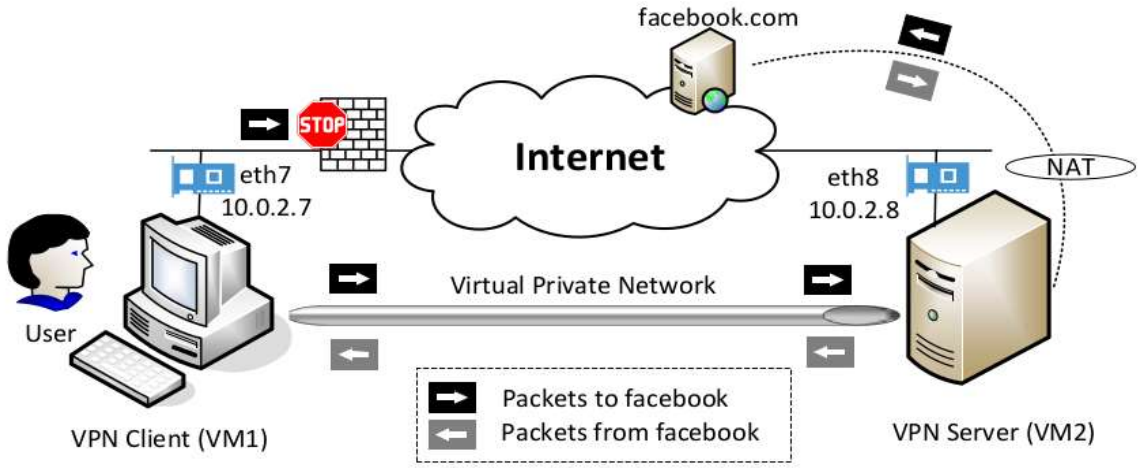
In this task, you will set up a firewall on VM1 to block the access of a target website. The following steps need to be followed.

1. Identify a website which you need to block on VM1 which is inside the firewall.
2. Make sure that the website has IP addresses in a fixed range or is fixed so that the blocking doesn’t get affected due to dynamic IP addresses allocated to a single website.
3. Execute following commands with appropriate IP address and interface to block access to the considered website.

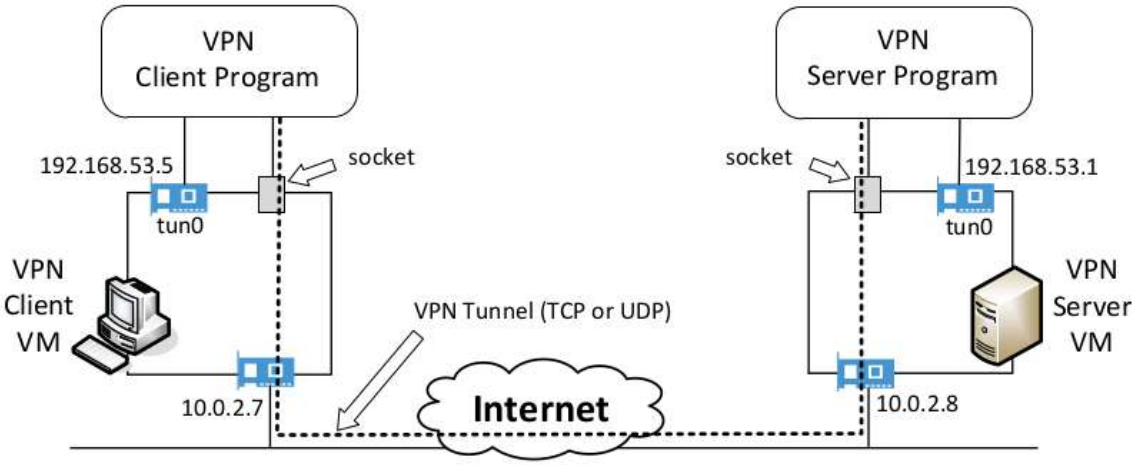
   $ sudo ufw enable
   $ sudo ufw deny out on eth12 to 128.230.210.0/24
   $ sudo ufw status
4. Provide the screenshot that ping to the website or access to the website is been blocked by the firewall.

Task 3: Bypassing Firewall using VPN

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.



The VPN client and server programs connect to the hosting system via a TUN interface, through which they do two things: (1) get IP packets from the hosting system, so the packets can be sent through the tunnel, (2) get IP packets from the tunnel, and then forward it to the hosting system, which will forward the packet to its final destination. Write down the vpnclient.c and vpnserver.c based on its implementation.



The following procedure describes how to create a VPN tunnel using the vpnclient and vpnserver programs.

Step 1: Run VPN Server: We first run the VPN server program vpnserver on the Server VM. After the program runs, a virtual TUN network interface will appear in the system (we can see it using the "ifconfig -a" command; the name of the interface will be tun0). This new interface is not yet configured, so we need to configure it by giving it an IP address. We use 192.168.53.1 for this interface, but you can use other IP addresses.

Run the following commands:

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

Please provide the screenshot and explanation on what these commands do.

The VPN Server needs to forward packets to other destinations, so it needs to function as a gateway. We need to enable the IP forwarding for a computer to behave like a gateway.

$ sudo sysctl net.ipv4.ip_forward=1

Step 2: Run VPN Client:
We now run the VPN client program on the Client VM. We assign IP address 192.168.53.5 to the tun0 interface

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

Please provide the screenshot for the above commands.

Step 3: Set Up Routing on Client and Server VMs:
After the above two steps, the tunnel will be established. Before we can use the tunnel, we need to set up routing paths on both client and server machines to direct the intended traffic through the tunnel.
One of the examples for the command:

$ sudo route add -net 10.20.30.0/24 eth0

Please analyze and provide screenshots of which command need to be implemented on the VPN client and the server.

Step 4: Set Up NAT on Server VM:
When the final destination sends packets back to users, the packet will be sent to the VPN Server first (think about why and write down your answer in the report). To reach the Internet, these packets will go through another NAT, which is provided by VirtualBox, but since the source IP is the Server VM, this second NAT will have no problem relaying back the returned packets from the Internet to the Server VM.
The following commands can enable the NAT on the Server VM (in your case, the name of the NAT Network adapter may not be called eth8; you just need to find its real name on your VM):

Clean all iptables rules:
$ sudo iptables -F
$ sudo iptables -t nat -F
Add a rule on postrouting position to the NatNetwork adapter (eth8) connected to the VPN server.
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o eth8

Task 4: Demonstration

If you have done the steps above correctly, you should be able to bypass the firewall. You should show that you can reach the blocked website from Client VM via the VPN. Provide screenshots regarding the same. The best way to show that is to capture the network traffic using Wireshark and describe the path of your packets using the captured traffic.

Submission:

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanations for the observations that are interesting or surprising. Please also list the important code snippets followed by an explanation. Simply attaching code without any explanation will not receive credits.