Evan Smith

Internet Security

Lab 7

Task 1- VM Setup:

We set up VM V to be connected to the internal network only:

Text

Description automatically generated

Note that we can ping the gateway (192.168.60.1) but not the other host, VM U (10.0.2.4)

We set up the gateway to have both a NAT connection as well as an Internal Network:

Text, letter

Description automatically generated

The gateway is able to ping both hosts, U and V.

We setup Host U to only have a NAT connection:

Text

Description automatically generated

It is able to ping the NAT connection on the gateway, but not either of the Internal Network connections.

**Observations:** We have successfully setup the network layout as described in the lab. There are two hosts, each of which cannot communicate with the other at all. In between is a gateway VM which can communicate with both Hosts independently.

**Task 2- Creating a VPN Tunnel using TUN/TAP**

Step 1- Run VPN server:

We first update the vpnclient.c file to reference the IP of the gateway VM:

Graphical user interface, text, application

Description automatically generated

We then build and run vpnserver.c on the gateway:

A screenshot of a computer

Description automatically generated with medium confidence

We can now set up the tun0 connection:

Text, letter

Description automatically generated

Step 2- Run VPN Client:

We setup and run the vpnclient.c program on Host U:

Text

Description automatically generated with medium confidence

We see that the connection is opened to the VPN server, and we confirm this by looking at the gateway:

Text

Description automatically generated

Step 3- Set Up Routing on Client and Server VMs:

We set up the routing on Host U, sending traffic to 192.168.60.0/24 through the tun0 VPN connection

Text, letter

Description automatically generated

We also verify that the server already has the necessary routing:

Text

Description automatically generated

Step 4- Set Up Routing on Host V:

A picture containing text

Description automatically generated

We now route responses back through the VPN gateway properly.

Step 5- Test the VPN Tunnel:

We are now able to ping Host V from Host U:

Text

Description automatically generated

And can Telnet as well:

Text, letter

Description automatically generated

Looking at traffic on the VPN server, we see VPN traffic moving through the tun0 connection, with all other traffic going elsewhere:

A picture containing timeline

Description automatically generated

Step 7- Tunnel-Breaking Test:

With the telnet active through the VPN, we can disconnect the VPN on both the client and server. When sending keypresses through the telnet, we can see them cached on Wireshark, and when the VPN is brought back online, the telnet session is resumed, including the cached keypresses.

**Observations:** We were able to set up a functional VPN that routes traffic through a middleman server (the gateway) and connect Hosts U and V. The traffic that passes through the specific tunnel connection is only that which uses the specific routing addresses we setup. Furthermore, the connection can cache information in the case that the VPN tunnel is momentarily lost and will send the information as soon as the connection is available again.

**Comments:** I found the toggling of the VPN on and off quite challenging, as the routing kept changing on me and I was finding it hard to keep track of what had to be done. Otherwise, I was impressed by the simplicity of the c programs that run the VPN service overall.