SEC 150 Final Project: Securing Communications with VPN

Titus Barik (tbarik@staff.waynecc.edu)

Version 1.2

1 Project Description

In this assignment, you will use the Tiny Core Linux distribution to create a virtual machine that hosts several services, including an OpenVPN static-key server, an insecure (by default) telnet daemon, and an iptables-based firewall to force clients to connect through the VPN.

Upon completion of this project, you will implement an OpenVPN server and configure it to use a static key. You will then learn how to secure unencrypted protocols such as telnet or FTP by wrapping them in a VPN cryptographic layer. Finally, you will implement firewall rules to tunnel all traffic through the VPN port.

Because this is a project, rather than an assignment, you are expected to discover and learn about much of the system on your own. Therefore, several assumptions may have been made and not every step is explicitly written out. You should therefore actively use the Discussion Board to seek clarification when needed.

2 Pre-Requisites

Before starting this assignment, you will need VMWare Player or VMWare Workstation. You will also need to download the Tiny Core ISO file (`tinycore_3.5.iso`). In addition, you will need OpenVPN Portable[1] to test that your VPN system is functioning correctly.

It is also highly convenient to download a Windows TFTP Server[2] in order to transfer files back and forth from VMWare. Tiny Core comes with a `tftp` client application that can connect to your physical machine for file transfer.

3 Instructions

1. Create a virtual machine for your Tiny Core system. Create a virtual IDE (not SCSI) hard disk with approximately 100 MB of disk space (0.1 GB). If you've forgotten how

[1] `http://sourceforge.net/projects/ovpnp/`

[2] `http://tftpd32.jounin.net/tftpd32_download.html`

to do this, you can refer back to the Midterm Project notes. Or, you can simply re-use your virtual machine image from before to save some time.

2. Install the following packages using the Apps tool: `openvpn`, `inetutils-servers`, and `iptables`. You may also need to install the OpenSSL package. If you are not comfortable using the `vi` text editor, you will also want to install `nano` at this point.

3. Set the password of the `tc` user to `sec150`. Recall that the command to do this is `sudo passwd tc`.

4. Start the `telnetd` daemon using executing the command `sudo /usr/sbin/telnetd`. Verify that telnet is enabled by using your local machine to telnet. Note that you are able to telnet using an insecure communication channel! The telnet protocol performs all network operations in plain-text, including the transmission of your login and password. **This is bad, and we need to fix this**.

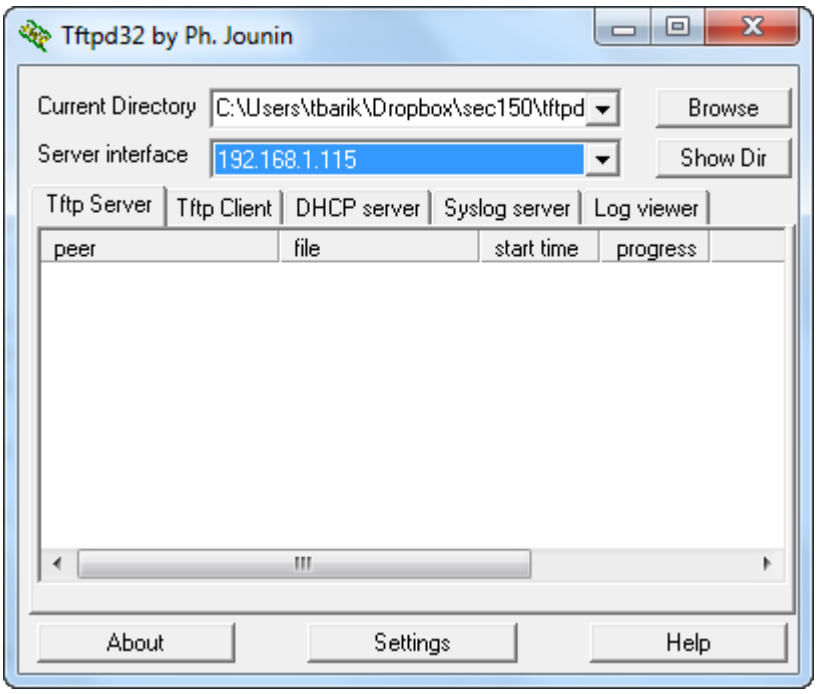
5. Create a directory called `/mnt/hda1/tce/final`. This directory will contain **all** of your files for the project.

6. Follow the instructions in the OpenVPN Static Key Mini-HOWTO[3]. Call the server file `server.ovpn`. Call the client file `client.ovpn`. Use the IP addresses `10.8.0.1` and `10.8.0.2` as given in the HOWTO. Be sure to change the `remote` line to point to the IP address of your Tiny Core VMWare image (`ifconfig`).

7. Don't forget to generate the `static.key` file. In PGP, we used asymmetric cryptograpy. While OpenVPN supports this technique, here we use symmetric cryptography by only having a single key. Unlike PGP, this single key is used by both the server and client for encryption and decryption.

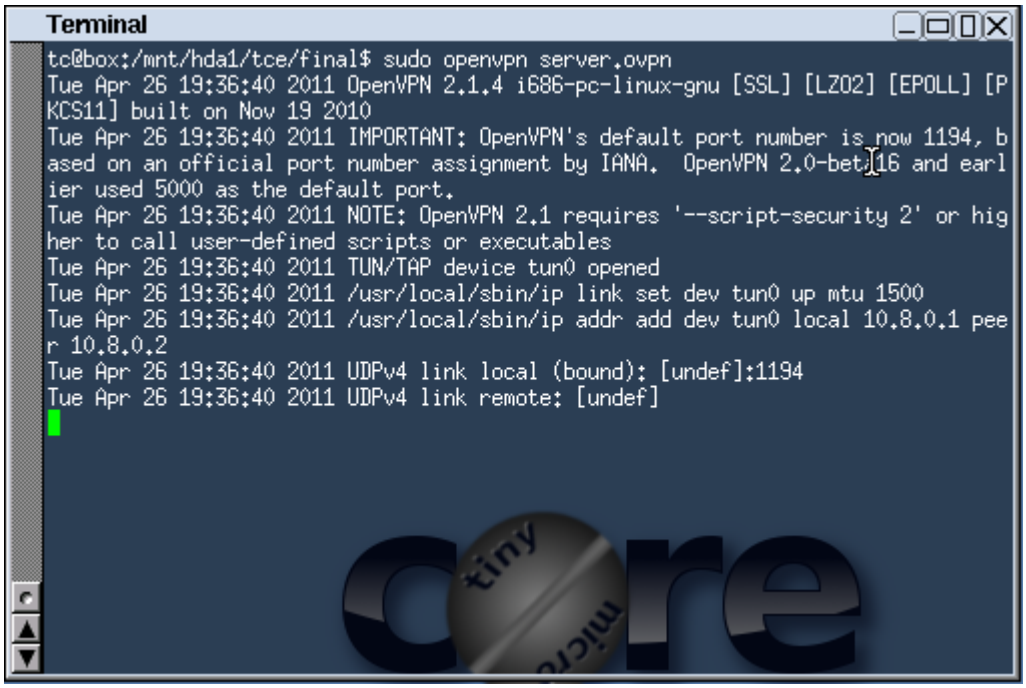
8. Use TFTP to transfer the `static.key` file and the `client.ovpn` file to your local machine. This is a portable application, so you can download just the zip file, not the full installer. The TFTP server will look something like the following:

[3] `http://openvpn.net/index.php/open-source/documentation/miscellaneous/78-static-key-mini-howto.html`

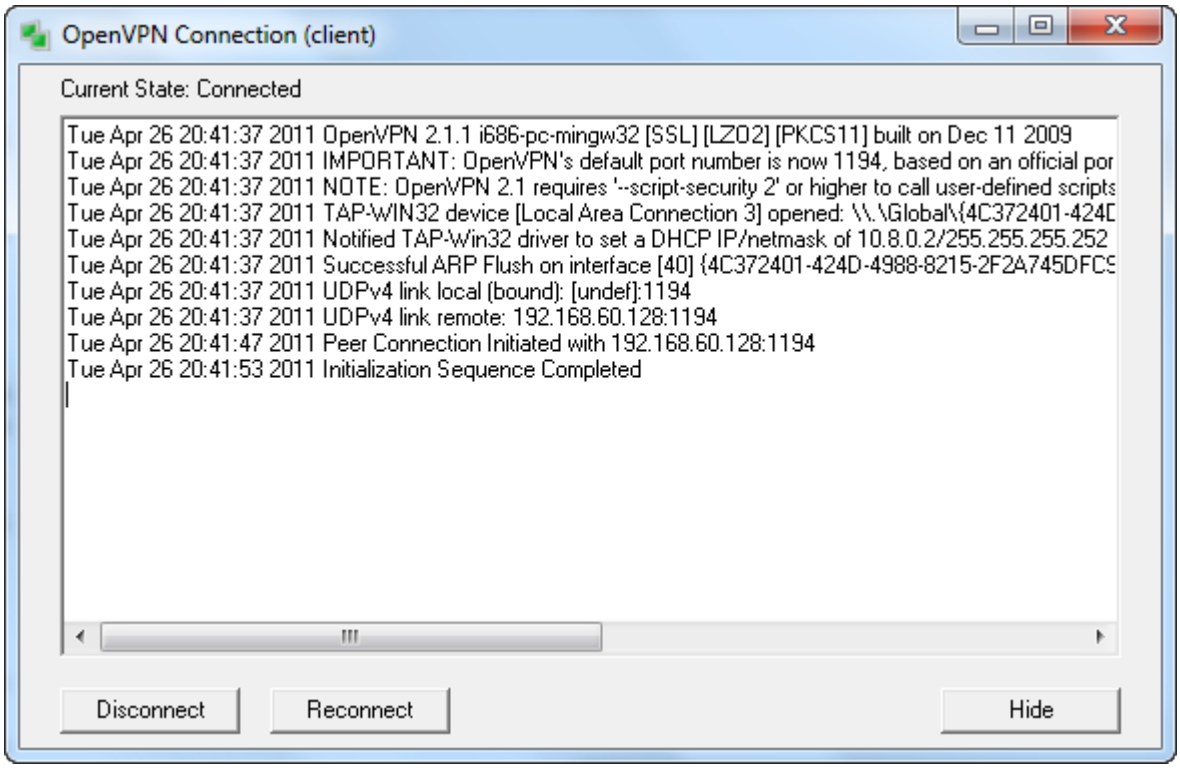


9. Next install OpenVPN Portable on your physical machine. Place the `static.key` and `client.ovpn` under `data/config`.

10. You can start the OpenVPN Server using the command `sudo openvpn --config server.ovpn`. The output will look something like the following:



11. Now, start the OpenVPN client. If successfully connected, the client will appear as follows:



12. Verify that you can `telnet` to the server using the VPN. `telnet 10.8.0.1`. If you don't have telnet, you can use PuTTY to accomplish the same task.

13. Observe that you can still telnet using the insecure channel (the non 10.8.0.1 address), which we must patch using iptables.

14. To do so, copy the file `/usr/local/sbin/basic-firewall` to `/mnt/hda1/tce/final`. Rename the file so that it is called `custom-firewall`. Edit the file and add the following statements after all the other iptables rules but before the `if` statement:

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
iptables -A INPUT -i tun+ -j ACCEPT
```

A full discussion of these rules is beyond the scope of this particular class (take the firewall class if you are interested). But briefly, this rule allows traffic to pass through UDP port 1194, which is the OpenVPN server port. It then allows all traffic to tunnel through the VPN tunnel driver.

15. Enable the firewall. Use `sudo ./custom-firewall`. If you've done everything correctly, the firewall should be successfully enabled:

16. Verify that telnet is **only** available through the VPN. You may have to restart the OpenVPN client to do this. Enabling a firewall while connections are currently active can sometimes cause strange results.

17. Congratulations! You have not only created a system that requires all users to VPN before accessing services, you have completed the final project!

4 Milestone

Submit a screen capture that shows what you've completed so far.

5 Submission

Submit all of your files as a single zip archive using Dropbox (unless you have personally made other arrangements with me). This archive should contain your entire VMWare image directory. It should also contain the files `client.ovpn` and `static.key`.

Under Moodle, post the URL of your Dropbox location, as well as a file called `md5.txt` containing the MD5 signature of your zip file. This is to prevent tampering after the deadline.

Students are encouraged to help each other; please post issues to the Forums.

6 Grading

Your assignment will be graded by executing the instructions in order. You can reproduce these same steps to ensure that your system functions correctly before submission. Here's how I will grade your submission:

1. I download your zip file using the URL you give me, and check that the attached MD5 signature you submitted using Moodle matches the signature of the zip file that I download through Dropbox.

2. Your VMWare image, if suspended, is first powered **off**. This is to make sure that your files persist across reboots.

3. I will then set a password of `sec150` (the password doesn't have to persist across power cycles).

4. The command `sudo /usr/sbin/telnetd` is run. I verify that I can telnet using my physical machine.

5. The command `sudo openvpn server.ovpn` is executed under your `final` directory.

6. I then copy the `client.ovpn` and `static.key` files, which you provided in the zip file itself, to the OpenVPN Portable directory. I modify the `remote` line in your client file for whatever IP address my VMWare image has.

7. I connect the server using VPN. I verify that telnet is functional through both the secure and insecure IP addresses.

8. I then enable the firewall using `sudo ./custom-firewall` to verify that I can only telnet to your system using the VPN IP address, and not the insecure IP address.

9. Then, I give you a perfect score if I am able to make it to this step. Otherwise, your score will be based on whatever step I get stuck at. **Make sure that I don't get stuck at the first step.**