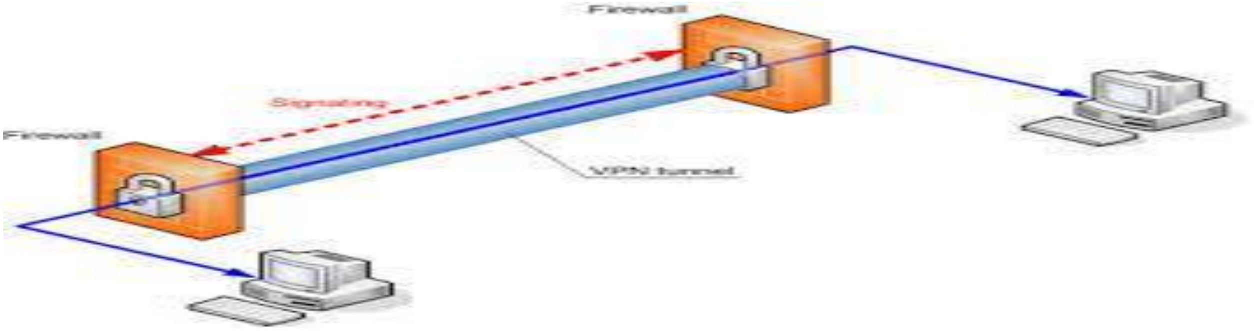
VPN Lab: OpenVPN

This lab is designed to give some experience with a virtual private network (VPN). We will use OpenVPN with symmetric encryption to create a secure connection between 2 systems. The VPN will increase security in 2 ways: restrict access and encrypt. A new virtual IP tunnel is established between 2 systems that has a restricted pair of IP addresses.



.

Overview READ_ME_FIRST!.txt

This lab needs to be performed with administrative rights, so you will need to use your Win10 VM. Also, we will be performing this lab with a partner in order create an end-to-end connection for our VPN. Detailed steps are provided in the next section. One system will be designated as the server, and the other as the client.

We will install OpenVPN, and then generate a symmetric secret key. Both of you should generate a secret key so you have the experience, but you must pick one and use just that one for the connection between you. The configuration is reasonably simple using a text editor. When doing so, make sure the correct file extensions are used for configuration files.

The most complicated aspect is that Windows uses a backslash i.e. \ to separate directories, and the OpenVPN configuration file requires the backslash to be escaped with a \. Spaces also need to be escaped with a \ so the path to our configuration file will look something like:

C:\\Program\ Files\\OpenVPN\\config\\username.ovpn

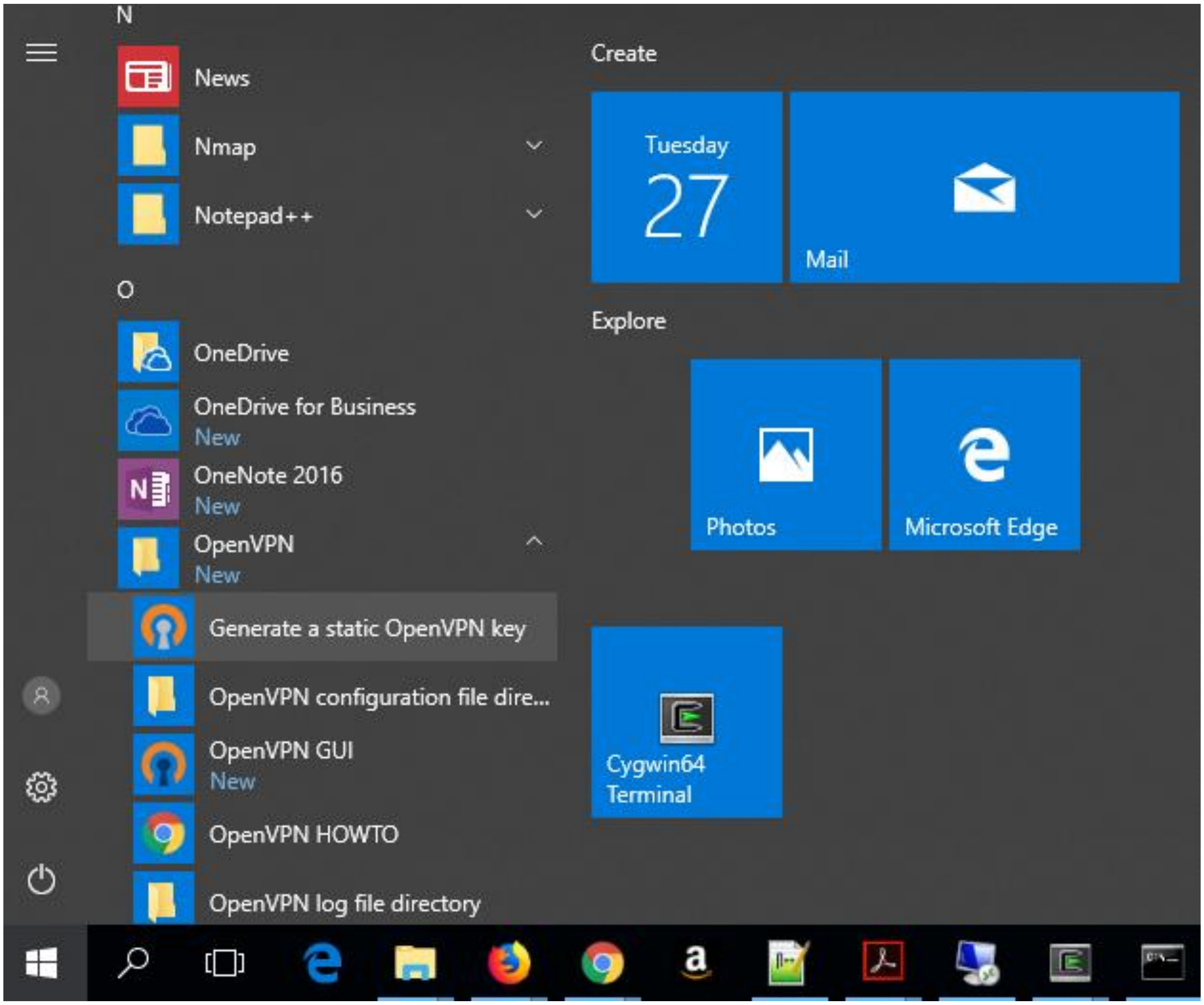
In the configuration, you will define a tunnel, configure the interface with the local VPN IP and the remote VPN IP and provide a path to the configuration file. The client needs to add an additional line to the configuration file designating the LAN IP address of the server so OpenVPN knows who to connect to.

After the VPN connection is successfully created, we will test it first through pinging each other's VPN IPs, and then connecting to our partner's web server through the VPN IPs.

Finally, we will use Wireshark to collect a web transfer on the VPN, and analyze the stream to see what VPN traffic looks like from outside the tunnel.

Detail Steps:

1. On your Win10 VM, install openvpn v 2.4.10 & make sure to include option for software for TAP devices.
2. You need one secret key for this connection. Have the Server generate a key, and then copy it over to the Client. Run Generate a static OpenVPN key as administrator (right-click):



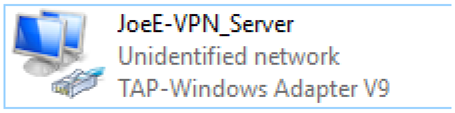
3. Be sure the same key is copied to C:\Program Files\OpenVPN\config\key.txt on both machines. This is the key.txt I made on my system:

Key.txt:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
32db5415b11ff64a33d50f848a427840
a299fcfd2e82858982c42fba00db60c3
68ce631c8d95295cdd62897dad2afacb
fa98166f56aeb2d3c201d44712ec69b3
f85024d45cb43040a5a5d47e754e4953
6d3f1dc47acdc381d4ec93653831b0c9
6b5b228a179ac4e776268af2122bf245
```

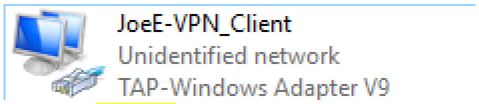
```
ad0ff3b1c1dab0b0f24ddfddbb410585
468a937af1b796dd3558930fb0432e78
28c905a6695ca12df4a865da7344ff13
811860dc7bf39c45f6469ff3320f5442
ce993a0261c0e14f8ce4e3c7df029f56
599488e55d257289b3985b9adcf2dd81
04b00977e1a104c7fe41542241c54859
268c0f73d32e697c3555c8cc3af063be
c2830d6484b780a4c7dc26ff19e73778
-----END OpenVPN Static key V1-----
```

4. Before we get crazy with network connections, let’s rename them to prevent some confusion later.
    a. Server: rename your new TAP-Windows Adapter to “<first name + last initial>-VPN_Server”, similar to below:



JoeE-VPN_Server
Unidentified network
TAP-Windows Adapter V9

    b. Client: rename your new TAP-Windows Adapter similarly as below:



JoeE-VPN_Client
Unidentified network
TAP-Windows Adapter V9

    c. Snips of <u>both</u> Server & Client renamed TAP-Windows Adapters.
5. On our systems, we will create a VPN on the private IP space of 10.0.0.0/8. Use two addresses (10.1.X.1 and 10.1.X.2), where you randomly select X to be between 0 and 255. One system will be designated as the “server” (10.1.X.1), and the other as the “client" (10.1.X.2).
    a. Server: Create a configuration file as follows for the server as C:\Program Files\OpenVPN\config\server.ovpn:

```
dev tun
remote 192.168.1.Z
ifconfig 10.1.99.1 10.1.99.2
secret C:\\Program\ Files\\OpenVPN\\config\\key.txt
--float
#This is my VPN Server config! – <Server person’s name here>
```

Note the remote address is the real physical ip address of the client machine

b. Client: Get the LAN IP of the server (needed for the remote address) and create a configuration file as follows for the client as C:\Program Files\OpenVPN\config\client.ovpn:

```
dev tun
remote 192.168.1.Z
ifconfig 10.1.99.2 10.1.99.1
secret C:\\Program\ Files\\OpenVPN\\config\\key.txt
--float
#This is my VPN client config! – <Client person's name here>
```

Note the remote address is the real physical ip address of the server machine

a. Snips of both Server & Client VPN config files.

6. Note that for the client the addresses in the ifconfig declarations for the server and client are reversed.
7. Also note, the 192.168.1.Z is the server's real non-VPN IP address, so modify it accordingly.
8. On both systems, find the OpenVPN icon in the Notification Area of the Windows 10 Taskbar (typically all the way to the right), right-click on it and select connect. If there is no OpenVPN icon, then run OpenVPN GUI from the OpenVPN menu.
9. If all goes well you should see a box showing that you have connected and what your assigned IP for the VPN is. It should look like this with a different IP. Also your OpenVPN icon should turn green.
10. In a command prompt window, run ipconfig and find the interface that has your new VPN IP address. Note the subnet mask and default gateway for that connection.
    a. Snips of both VPN's Ethernet adapter names, IPv4 addresses, subnet mask & gateway from the CLI.
11. Ping each other's VPN IP address to verify connectivity via the VPN.
    a. Snips of both successfully pinging each other's VPN IPs via CLI.
12. Start your IIS Web Servers and verify you can connect to your partner's web server using their VPN IP Address.

    a. Snips of both successfully browsing each other's websites via VPN IPs.

13. Start Wireshark, and using your LAN adapter (192.168.1.X) start a capture. Reload the web page you connected to in the above step.
14. Stop the Wireshark capture and set a filter for <ip.addr == 192.168.1.X>, where X is your LAN address.
15. Locate packets between your address and your partner's address. Click on one of these and use the Analyze->Follow UDP Stream.
    a. Snip of the UDP Stream.

Reflections

1. Describe in language someone from a non-technical program at Champlain College could understand, the reason and method used to create a VPN.
2. In one of steps above, you were asked to note the subnet mask and default gateway for your VPN. What was the subnet mask and can you explain why it makes sense?
3. In one of steps above, what was the nature of the data in the UDP stream?
4. Do some research on the Internet. How much more difficult would it be to setup asymmetric encryption? Explain what would that look like.