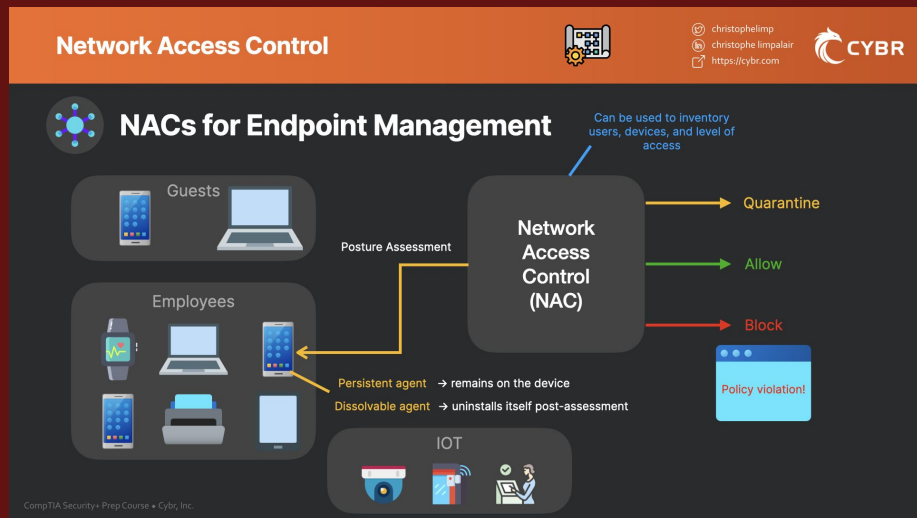


OpenVPN

Trey Atwood

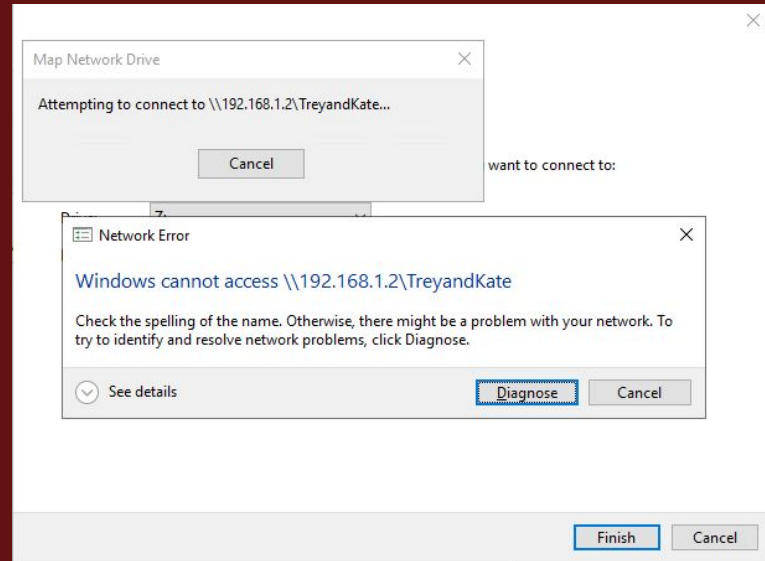
Network Access Control (Internal and External)

Network Access Control or NAC, is all about controlling who can get onto our network, and what they can do once they're in. External NAC that protects the network side from the internet into our closed network environment. For this activity we setup OpenVPN as our only entry point or External NAC into our network. PFSense only exposes the VPN to the internet, and the users must authenticate with their username and password that we setup in the local users in PFSense. As for the Internal NAC, this is everything that controls what a connected user to reach inside our network. In our OpenVPN settings in this lab we have a separate VPN subnet for the users connected. VPN clients get IPs in the 192.168.0.0/24 range, while our LAN is 192.168.1.0. This separation allows us to add rules that only allow users from the range 192.168.0.0/24 to reach our file server 192.168.1.2 over SMB.



Network Policy (Development and Enforcement)

Network policy is the set of rules that says who can do what within the network, under what conditions, and how it's going to be monitored. The development side is where you are "writing the rules" here you define the goals of what you want the network policy to do. For this exercise we want to protect access to our file sharing with a VPN, so that is our goal. As for our lab we want to create a certain user who gets access to the file sharing which is another step in the development process. As for the enforcement, this is where you make the rules work. In the lab we tunneled network traffic through "192.168.0.0/24" and the local network "192.168.1.0/24" to enforce users to be separate from the LAN and they must cross through the firewall. On the WAN, only OpenVPN ports 1194 are allowed in, and in the WAN rules we disable SMB from the internet by disabling port 445 which forces users to use the VPN.



To start setting up OpenVPN, we first need to decide where our users are going to be authenticated. For this we want our users to be authenticated from our own PFSense, so for this we choose "Local User Access".

Next we'll need to set up a Certificate Authority. This is necessary because all of the users who are using the VPN locally are going to need a certificate, and in order to have a certificate given to the users we need to properly setup a working Certificate Authority.

For the name for the CA I named it "Local User Access CA (OpenVPN)"

The screenshot shows the PFSense Community Edition interface. The top navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main heading is "Wizard / OpenVPN Remote Access Server Setup /". Below this, a "Step:" label is followed by the title "OpenVPN Remote Access Server Setup". A message states: "This wizard will provide guidance through an OpenVPN Remote Access Server Setup . The wizard may be stopped at any time by clicking the logo image at the top of the screen." The next section is "Select an Authentication Backend Type". It features a "Type of Server" label and a dropdown menu currently set to "Local User Access". A red arrow points to the dropdown. Below the dropdown is a note: "NOTE: If unsure, leave this set to 'Local User Access.'". At the bottom of this section is a blue button labeled ">> Next".

The screenshot shows the "Create a New Certificate Authority (CA) Certificate" form. It has a dark header with the title. Below the header, the label "Descriptive name" is followed by a text input field containing "Local User Access CA (OpenVPN)". A red arrow points to the input field. Below the input field is a note: "A name for administrative reference, to identify this certificate."

Next we will configure a Host Name for our VPN. For this I used the name of our class CIS 3880. Normally for the host name you would use the domain name of the company would be using the VPN.

Next, we will need to setup the server information. Since we are going to be using the VPN to access file sharing I'll name the server "Remote File Server Access", keeping it specific as to what it's going to be used for. Next we will configure the IPv4 tunnel network, which we will set as "192.168.0.0/24", and set the IPv4 Local Network to "192.168.1.0/24". Our local network is the network that we are attaching the VPN to.

Create a New Server Certificate

Descriptive name

vpn.cis3880.com

A name for administrative reference, to identify this certificate.

General OpenVPN Server Information

Description

Remote File Server Access

A name for this OpenVPN instance, for distinguish the purpose of the service this VPN on clients.

Tunnel Settings

IPv4 Tunnel Network

192.168.0.0/24

This is the virtual network used for private communications between clients (eg. 10.0.8.0/24). The first network address will be assigned to the server and subsequent addresses will be assigned to connecting clients.

Redirect IPv4 Gateway

☐ Force all client generated traffic through the tunnel.

IPv4 Local Network

192.168.1.0/24

This is the network that will be accessible from the remote endpoint. It is not adding a route to the local network through this tunnel on the server.

Finishing out the server setup, we will need to configure the DNS default domain, and the IP for our DNS server. For this I just used the name of our class once again "[CIS3880.com](https://cis3880.com)" as the domain name, and set the DNS server to "192.168.1.1" which is the IP address for PFSense. If someone connects to the VPN they are going to use PFSense as the DNS for the VPN.

After we finish configuring all of these steps we are complete with the setup of the VPN!

Next we need to set up a new User.

Advanced Client Settings

DNS Default Domain

Provide a default domain name to clients.

DNS Server 1

DNS server IP to provide to connecting clients.

Finished!

OpenVPN Remote Access Server Setup Wizard

Configuration Complete!

The configuration is now complete.

Adding users for the VPN depends on the chosen authentication method under [System > User Manager](#). For remote authentication servers, add

To easily export client configurations, browse to [System > Packages](#)

For the new user I used my own name "Trey Atwood" and set a password for my new user account. After setting up the username and password check the box that creates a new user certificate for the user.

When creating the new certificate for the user, I named it with my first name followed by the domain name that we are using for the VPN. And make sure you select the CA that we just created in the last steps as the CA for the user.

User Properties	
Defined by	USER
Disabled	<input type="checkbox"/> This user cannot login
Username	<input type="text" value="Trey Atwood"/>
Password	<input type="password" value="*****"/> <input type="password" value="*****"/>
	Enter a new password. Type the new password again for confirmation.

Certificate



Click to create a user certificate

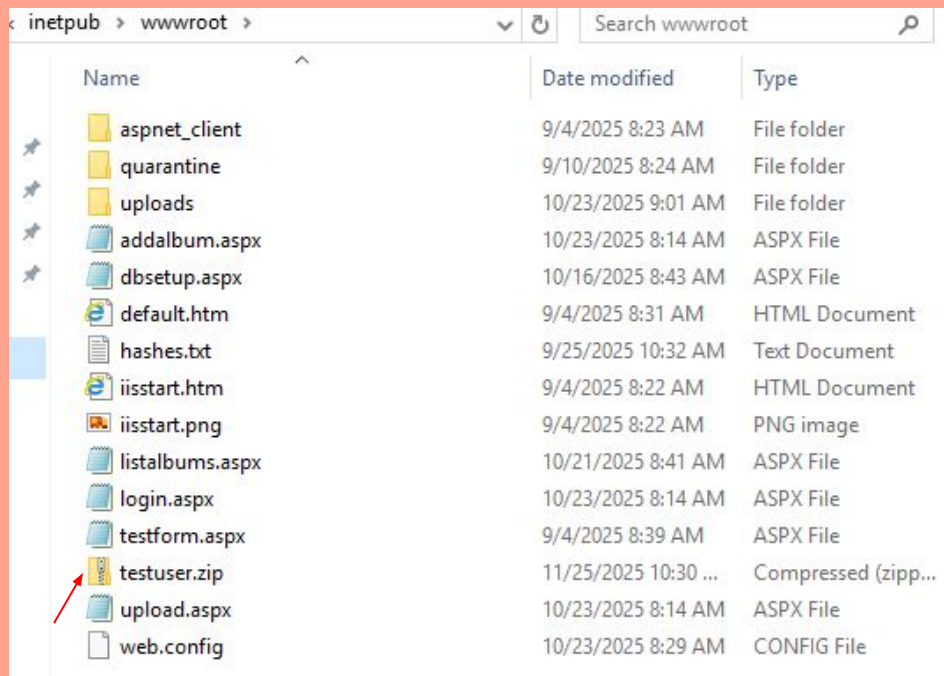
Create Certificate for User	
Descriptive name	<input type="text" value="trey.CIS3880.com"/>
Certificate authority	<input type="text" value="Local User Access CA (OpenVPN)"/>

To export our client to our user through PFSense, we will navigate to our OpenVPN client list and download the current windows installed x64.

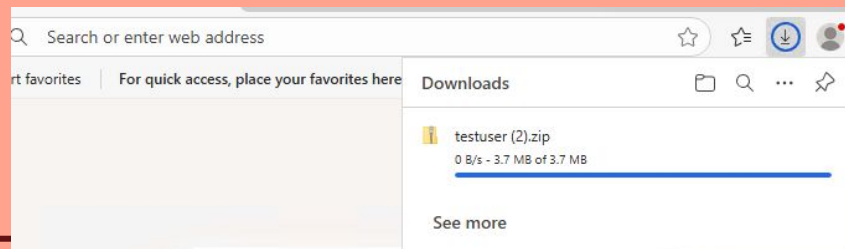
Once we download this we will need to move it over to our Azure machine so we can download and set up OpenVPN there. To do this I'll all the downloaded client file to my web server to make it easy to download on my Azure machine.

User	Certificate Name	Export
TreyAtwood	trey.CIS3880.com	<div>- Inline Configurations: Most Clients Android OpenVPN Connect (iOS/Android)</div> <div>- Bundled Configurations: Archive Config File Only</div> <div>- Current Windows Installers (2.6.7-lx001): 64-bit 32-bit</div> <div>- Previous Windows Installers (2.5.9-lx601): 64-bit 32-bit</div> <div>- Legacy Windows Installers (2.4.12-lx601): 10/2016/2019 7/8/8.1/2012r2</div> <div>- Viscosity (Mac OS X and Windows): Viscosity Bundle Viscosity Inline Config</div>

Here in my wwwroot folder I added the file test.user.zip which has the file we need to configure OpenVPN on the Azure machine.

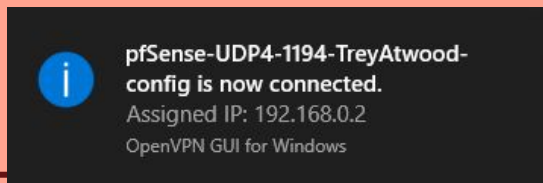
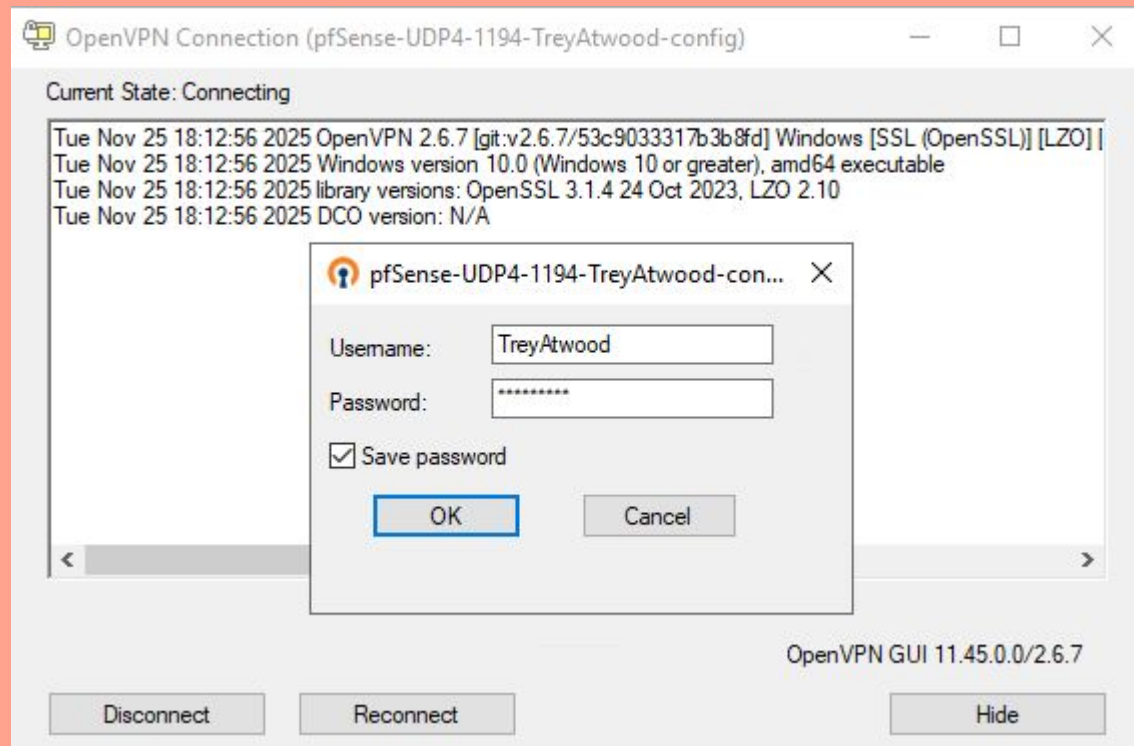


Name	Date modified	Type
aspnet_client	9/4/2025 8:23 AM	File folder
quarantine	9/10/2025 8:24 AM	File folder
uploads	10/23/2025 9:01 AM	File folder
addalbum.aspx	10/23/2025 8:14 AM	ASPX File
dbsetup.aspx	10/16/2025 8:43 AM	ASPX File
default.htm	9/4/2025 8:31 AM	HTML Document
hashes.txt	9/25/2025 10:32 AM	Text Document
iisstart.htm	9/4/2025 8:22 AM	HTML Document
iisstart.png	9/4/2025 8:22 AM	PNG image
listalbums.aspx	10/21/2025 8:41 AM	ASPX File
login.aspx	10/23/2025 8:14 AM	ASPX File
testform.aspx	9/4/2025 8:39 AM	ASPX File
testuser.zip	11/25/2025 10:30 ...	Compressed (zipp...
upload.aspx	10/23/2025 8:14 AM	ASPX File
web.config	10/23/2025 8:29 AM	CONFIG File



Now we need to install the VPN onto our Azure Machine to test to make sure that it works for our file sharing. Follow the setup wizards that come with the installation for OpenVPN and login with the user credentials.

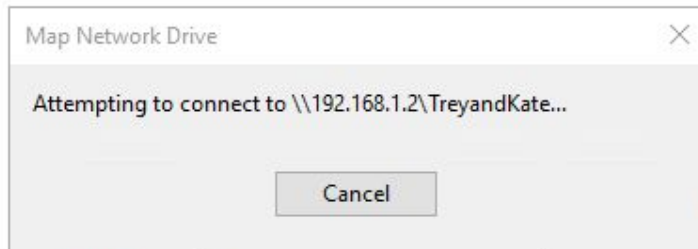
Here we login with the username "TreyAtwood" and the password I created.



To make sure that connection works, I'll connect to the shared folder that we created in a previous exercise.

This PC > TreyandKate (\\192.168.1.2) (Z:) Search TreyandKate (\\192.168.1.2)

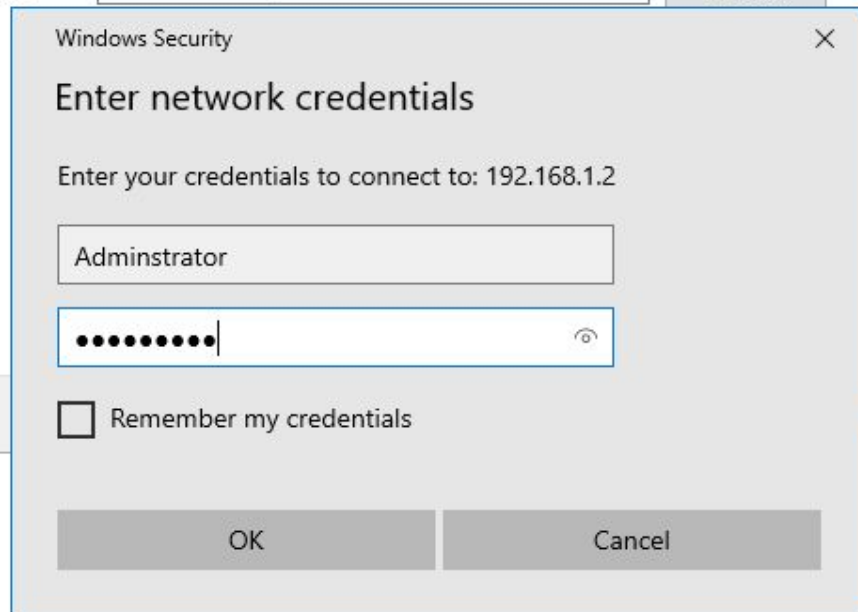
Name	Date modified	Type	Size
AdminCreatedThis.txt	9/16/2025 5:27 PM	Text Document	1 KB
KateCreatedThis.txt	9/16/2025 5:27 PM	Text Document	1 KB
TreyCreatedThis.txt	9/16/2025 5:27 PM	Text Document	1 KB



want to connect to:

Drive: Z: ▼

Folder: \\192.168.1.2\\TreyandKate ▼ Browse...



To make sure that we can't access the file share without the connection of the VPN I'll disable the VPN and try to connect to the file share.

As you can see here windows isn't able to access the file share without an active connection to the VPN that we just set up!

