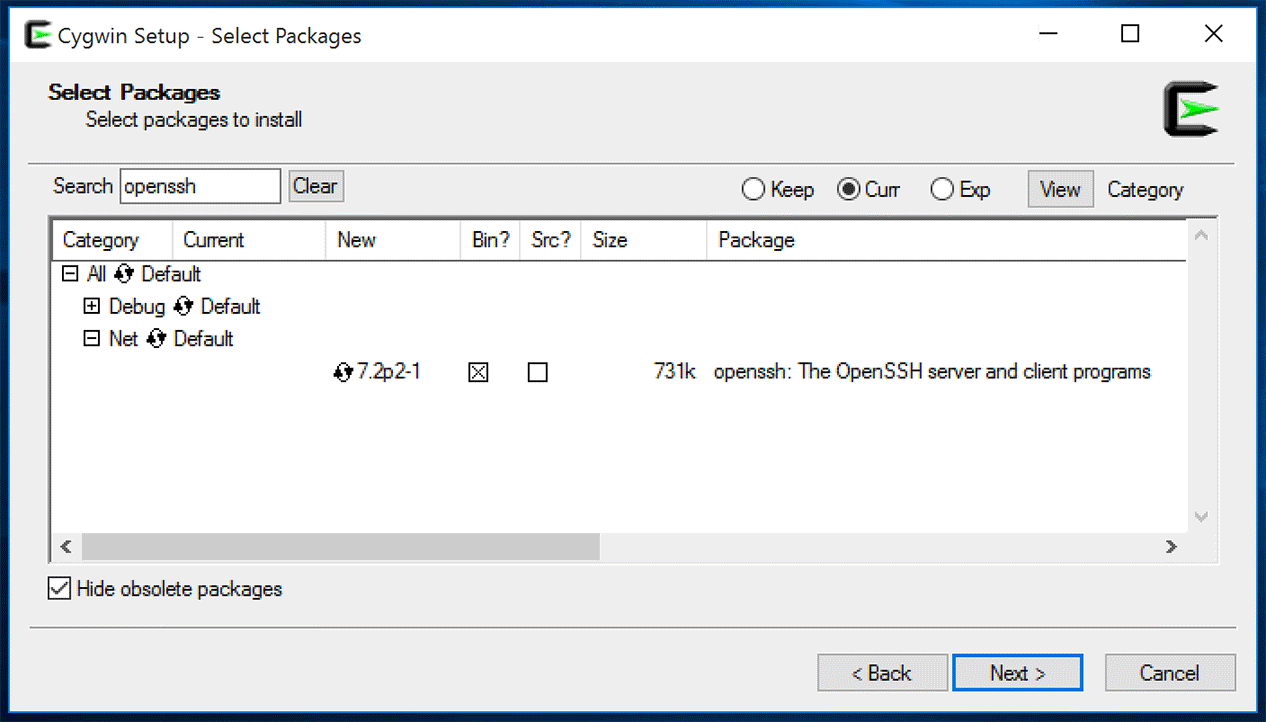
**Create Certificates:**

Easy RSA: <https://github.com/OpenVPN/easy-rsa>

Based on instructions: <https://secure.sparklabs.com/support/kb/article/creating-certificates-and-keys-for-your-openvpn-server/#windows>

# Windows

We will assume that you are running Windows 10. First you will need to install a copy of [Cygwin](http://cygwin.com/). We will use Cygwin to run the scripts provided by Easy-RSA. Download and run the setup executable to install it. When prompted to **Select Packages**, install **OpenSSH** by typing openssh in the search bar and clicking on the word **Skip** on the line corresponding to **openssh: The OpenSSH server and client programs** in the **Net** drop down. This will change the word from **Skip** to the version number. Now continue with the rest of the Cygwin installation. We will assume that Cygwin has been installed to the directory C:\cygwin64\. Note that directories are separated in Cygwin with forward slashes (/).

[](https://secure.sparklabs.com/support/kb/article/creating-certificates-and-keys-for-your-openvpn-server/)

You will need to download Easy-RSA so that we can create the required certificates. In a browser, go to <https://github.com/OpenVPN/easy-rsa/releases> and download the latest .zip version for your PC (EasyRSA-\*.zip). We will assume that this file has been downloaded to the C:\Users\your-account-name\Downloadsdirectory. Once the download is complete, right click on this file and select **Extract all**. Extract the contents of the archive to C:\Users\your-account-name\Downloads\ Now that we have extracted the contents of this file, we need to open the command window. Type cmd into the **Search the web and Windows** box in the taskbar and press ENTER. This will open the Command Prompt app.

## Certificate Information

The certificates used by your OpenVPN server carry information about you and/or your organization. Follow the instructions below to configure those details.

1. ~~Create a directory to store the keys we will generate for the server:~~

~~mkdir C:\Users\ziegl\Documents\Viscosity\server\keys~~

1. ~~Create a directory for the client credentials:~~

~~mkdir C:\Users\ziegl\Documents\Viscosity\client\keys~~

1. Copy the **vars** file to configure the certificate authority parameters:

copy C:\Users\ziegl\Downloads\EasyRSA-3.0.1\vars.example C:\Users\ziegl\Downloads\EasyRSA-3.0.1\vars

1. Open this file for editing in Sublime Text:

C:\Users\your-account-name\Downloads\EasyRSA-3.0.1\vars

The certificates created by the authority carry information about the person or organization using them to help identify the certificates and keys. Scroll down until you see the section:

#set\_var EASYRSA\_REQ\_COUNTRY "US"

#set\_var EASYRSA\_REQ\_PROVINCE "California"

#set\_var EASYRSA\_REQ\_CITY "San Francisco"

#set\_var EASYRSA\_REQ\_ORG "Copyleft Certificate Co"

#set\_var EASYRSA\_REQ\_EMAIL "me@example.net"

#set\_var EASYRSA\_REQ\_OU "My Organizational Unit"

1. Uncomment all 6 of these lines by deleting the '#' symbol from the start of each line
2. Modify these details to suit you and/or your organization

Scroll down to the next section until you see:

#set\_var EASYRSA\_KEY\_SIZE 2048

1. Uncomment this line by deleting the '#' symbol from the start of it
2. We will use a 2048 bit key, so leave this number unchanged
3. Save and exit Wordpad

## Confirm EasyRSA is ready

## Generating the Server Credentials

Easy-RSA provides a number of scripts that we can use to create our Certificate Authority and our required certificates and keys.

1. Open a Cygwin prompt:

C:\cygwin64\bin\mintty.exe -

1. In the Cygwin prompt, navigate to the easy-rsa directory:

cd C:/Users/ziegl/Downloads/EasyRSA-3.0.1/

1. Remove any Windows-specific characters:

sed $'s/\r//' -i ./vars

1. Initialize the PKI (Public Key Infrastructure), by typing:

./easyrsa init-pki

this should create a directory **/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki** that will store our certificates and keys.

1. Create the certificate authority by typing:

./easyrsa build-ca nopass

1. You will be prompted to provide a Common Name. Enter the name **server** and press ENTER
2. ~~Copy the certificate authority certificate to your server keys directory:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/ca.crt C:/Users/your-account-name/Documents/Viscosity/server/keys/ca.crt~~

You now have a certificate authority set up and we can move on to creating certificates for your OpenVPN server.

1. Create the server .crt and .key files by typing:

./easyrsa build-server-full server nopass

1. ~~Copy the server certificate:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/issued/server.crt C:/Users/your-account-name/Documents/Viscosity/server/keys/server.crt~~

1. ~~Copy the server key:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/private/server.key C:/Users/your-account-name/Documents/Viscosity/server/keys/server.key~~

1. Now generate the encryption (Diffie-Hellman) parameters. Be patient, this process can take up to a few minutes:

./easyrsa gen-dh

1. ~~Copy the Diffie-Hellman file:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/dh.pem C:/Users/your-account-name/Documents/Viscosity/server/keys/dh2048.pem~~

## Generating the Client Credentials

To connect to your OpenVPN server, you need to create client credentials. We will name our client certificate **client1**. Feel free to use whatever name you prefer.

1. Create client credentials for every device that will connect to the OpenVPN server (‘client1’ is the example name below, name them appropriately per device):

./easyrsa build-client-full client1 nopass

1. ~~Copy the client certificate:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/issued/client1.crt C:/Users/your-account-name/Documents/Viscosity/client/keys/client1.crt~~

1. ~~Copy the client key:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/private/client1.key C:/Users/your-account-name/Documents/Viscosity/client/keys/client1.key~~

1. ~~Copy the CA certificate:~~

~~cp C:/Users/your-account-name/Downloads/EasyRSA-3.0.1/pki/ca.crt C:/Users/your-account-name/Documents/Viscosity/client/keys/ca.crt~~

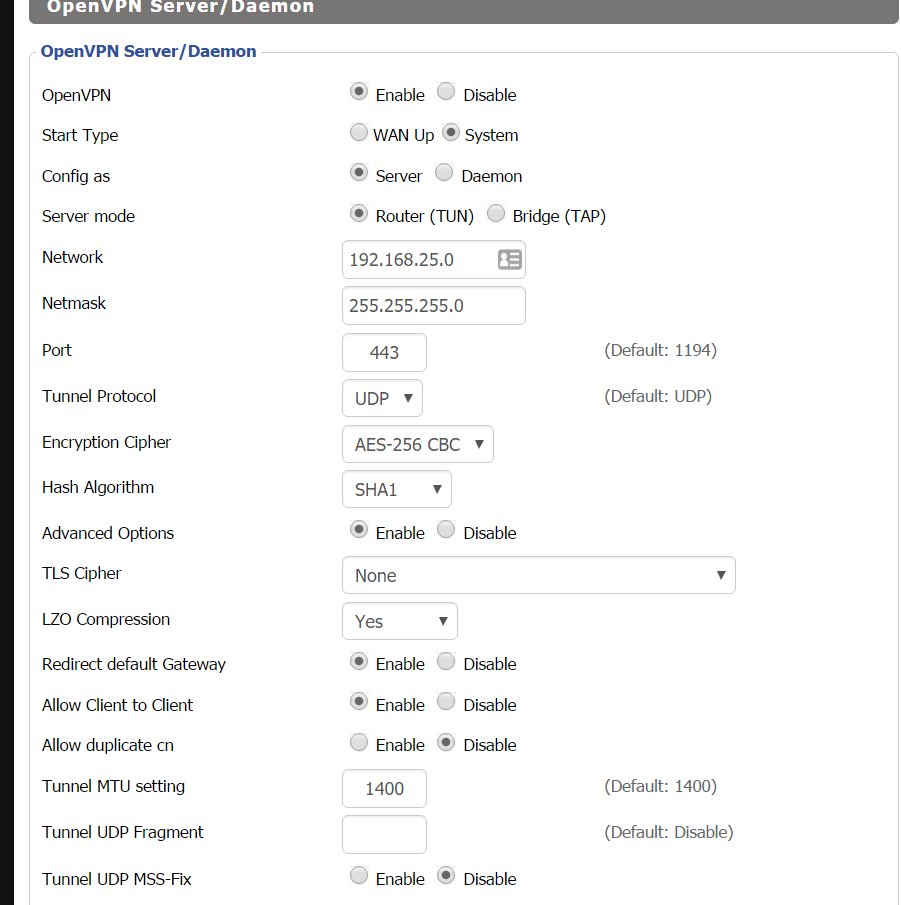
These client credentials are ready to be loaded to your OpenVPN server. Your server credentials need to be transferred to the server itself.

1. ***Copy*** the certificates and keys (see locations in strikethrough lines above) to one secure folder for safe keeping (e.g. veracrypt drive)
2. ***Securely move*** the ‘pki’ folder! (this is all your crypto, don’t leave it lying around! Recommend copying the whole EasyRSA folder to a veracrypt drive, this way if you have to add more clients it’s all where it needs to be).

**Setup VPN in DD-WRT:**

Good source of general setup repeated below: <https://www.fperkins.com/uncategorized/openvpn-setup-on-dd-wrt-router-with-mac-and-ios-10.php>

1. Login to your router.
2. Go to Services, VPN, enable OpenVPN Server.  Use these options:



1. CA Cert, Public Server Cert, Private Server Key, DH PEM:  These are the four SERVER certs you created in the previous step.  You need to copy and paste the block of text that ALSO includes —BEGIN CERT… until —END CERT.  Don’t leave those out (they are not comments)

Some tips:

server.crt has a bunch of other stuff in it.  Only copy the cert stuff (again, including the —- lines)

ca.cert = CA Cert (config section in DD-WRT)

dh2048.pem = DH PEM config section

server.crt = Public Server Cert config section

server.key = Private Server Key config section

Ok, now here is the big one.  Add this to the Additional Config section:

push "route 192.168.1.0 255.255.255.0"

or

server 192.168.25.0 255.255.255.0

route add 192.168.25.0 mask 255.255.255.0 gw 192.168.1.1

push "route 192.168.1.0 255.255.255.0"

1. Then in DD-WRT Admin GUI go to Status, OpenVPN and see if it’s running. If you don’t see a lot of information, it failed to start. (See troubleshooting on the website).
2. Once you have it running, now you get to play with iptables! In order for the .25 OpenVPN segment to see your other computers on the network, you have to setup some rules. What does that mean? Basically in order for traffic in the .25 segment to play with your internal computers, you need to explain how they can travel there. Go to the DD-WRT Administration Tab > Commands, and enter the below commands:

iptables -t nat -A POSTROUTING -s 192.168.25.0/24 -j MASQUERADE

iptables -I INPUT 1 -p tcp -–dport 1194 -j ACCEPT

iptables -I FORWARD 1 –-source 192.168.25.0/24 -j ACCEPT

iptables -I FORWARD -i br0 -o tun0 -j ACCEPT

iptables -I FORWARD -i tun0 -o br0 -j ACCEPT

Websites lines:

iptables -I INPUT 1 -p udp –dport 1194 -j ACCEPT

iptables -I FORWARD 1 –source 192.168.66.0/24 -j ACCEPT

iptables -A FORWARD -i tun+ -j ACCEPT

iptables -A FORWARD -o tun+ -j ACCEPT

* 1. The first line ensures that your firewall does not block access to your router. We want to make sure UDP port 1194 is open. If it’s not open then the VPN will never work! To test if the port is open, you’ll need to connect from somewhere remote (after you run these commands obviously).
  2. The next three lines all allow the forwarding of those requests from your VPN into your local network.

1. Once they are all set, Click “Save Firewall”

**Client Setup:**

1. Create .OVPN file in Sublime Text:

client

dev tun

proto udp

remote yourDDNSwebsite.com 1194

nobind

persist-key

persist-tun

verb 4

float

ca ca.crt

cert jarvis.crt

key jarvis.key

comp-lzo yes

tun-mtu 1400

auth SHA1

cipher AES-256-CBC

1. download a VPN client for your iPhone so go to the app store and download OpenVPN.
2. Now this is the pain in the butt part, you have to copy your client certs (NOT SERVER!) onto your iPhone.  Of course you can’t magically drag files onto your phone so you have to use our friend iTunes.  To do this, connect your phone to iTunes (via cable I guess).  Click on your device, click on Apps, scroll DOWN to the bottom until you see a list of apps that allow you to copy files.  (This is not the section above where you can rearrange your app icons).  Click on OpenVPN app and drag the 4 clients files into the right hand side Documents section:
   1. ca.crt
   2. client1.crt
   3. client1.key
   4. client1.opvn
3. Open the OpenVPN app on your iPhone and you should see a configuration for the public ip address of your OpenVPN server.  If you screw up, you can delete it by touching on the ip address, upper right hand corner touch Delete, touch red circle on the left with the line through it (very bad UI experience).  Before you connect to your VPN, ensure to turn off Wifi on your iPhone otherwise it’s not a very good test (I don’t think it will work anyway!)
4. To test connected to your OpenVPN, slide the button to the right (again very bad UI design).  You should see it say Waiting for server, Connecting, Connected!  Down below in VPN IPv4 section you should see the IP address that was assigned.  For me it’s 192.169.25.X  If you are not connected, then touch the “Connecting” section of the app to read the log.

Note: Initially when I set it up, there was a time difference between where I setup the cert and GMT.  I had to wait 5 hours for it to be resolved.  There is a way to fix it, but since I was already going to sleep the problem worked itself out the next day ;-).  To manually check this look at your clien1.crt file.  There is a Validity section which helps you understand when the cert STARTS working and when it ends (in 10 years).

1. If all that worked, now you can see if you can access anything in your home network.  You can try to use VNC Lite to connect to a home computer or download the Ping app from the app store and ensure you can see your local network.  I assume there is something you want to VPN in for so test it.  You can also test to ensure you are connected by pinging the IP address of your phone. For example, from a computer on my local network,  ping the IP address that your phone was assigned. e.g. ping 192.168.25.2

DD-WRT Wiki: <https://www.dd-wrt.com/wiki/index.php/VPN_(the_easy_way)_v24%2B>

Other Site that had great info: <https://www.dd-wrt.com/phpBB2/viewtopic.php?t=304754&sid=a9a46abfa46f81dacd0bb6515fbd6bc9>

**Tips for Setup:**

<https://ipcamtalk.com/threads/vpn-primer-for-noobs.14601/>

* Not disabling uPNP and shutting down old port forwards after having VPN Setup.
* Really the only choice is OpenVPN vs L2TP/IPSec, little else is trustworthy as those two; for most people OpenVPN is easier to setup and run.. OpenVPN requires clients to be installed on all your devices, whereas L2TP/IPSec clients are built in natively on every modern device (Windows/OSX/iOS/Android/Linux).. typically its best use what you have available already.. If you configure your OpenVPN server to listen on port 443, the same port as HTTPS websites, then you can expect it to work on even the most restrictive remote networks.
* Give each device its own unique login and generate a one time password for it and save it to the device.. this way if a device gets lost or stolen you can simply delete that user account, or if you upgrade/replace the device you just generate a new password and render everything else unable to login without having to change the credentials on all your devices anytime you upgrade/loose an item.
* Use AES 256:
  + aes-128-cbc, 3200/45.24 = 70.73 Mbps
  + aes-256-cbc, 3200/48.57 = 65.88 Mbps
* The --tls-auth option uses a static pre-shared key (PSK) that must be generated in advance and shared among all peers. This features adds "extra protection" to the TLS channel by requiring that incoming packets have a valid signature generated using the PSK key. If this key is ever changed, it must be changed on all peers at the same time (there is no support for rollover.)
  + The primary benefit is that an unauthenticated client cannot cause the same CPU/crypto load against a server as the junk traffic can be dropped much sooner. This can aid in mitigating denial-of-service attempts.
  + This feature by itself does not improve the TLS auth in any way, although it offers a 2nd line of defense if a future flaw is discovered in a particular TLS cipher-suite or implementation (such as CVE-2014-0160, Heartbleed, where the tls-auth key provided protection against attackers who did not have a copy). However, it offers no protection at all in the event of a complete cryptographic break that can allow decryption of a cipher-suite's traffic.
  + Generate a PSK with:
    - openvpn --genkey --secret ta.key
    - And reference it in the configs as such. The 0/1 value is arbitrary and must be the *opposite* between peers (or omitted entirely.)
    - # server-example
    - --tls-auth ta.key 0
    - # client-example
    - --tls-auth ta.key 1
* Blue Iris’ iOS app: you need to set both the LAN and WAN address to the same internal (local/LAN) IP:PORT. (Because you’ll have to turn on the VPN first, and once you do it’ll be as if you’re on the local network).