I thought I would share this with everyone here how I was able to get this working. I have been trying for sometime to configure my Linksys e4200v1 router with OpenVPN for sometime without success. It seems there is a huge disconnect with the wiki's out there and how the newer builds are configured. However, between reading about bugs in certain builds and improper configs etc I thought I would share this method to get you going.   
  
I don't know much about the technical details around how this all works but I was able to piece together what was required from this important post: <http://www.dd-wrt.com/phpBB2/viewtopic.php?p=641031> as well as the OpenVPN manual/wiki.   
  
A huge thank you to **GTI\_nut** and **maddes.b** for their posts which really helped me to get this all going.   
  
Here are a few things about my setup:   
  
**Router:** Linksys e4200v1   
**Build:** king kong r20500 mega <http://www.desipro.de/ddwrt/K26/r20500/>   
**OpenVPN version used to generate certs:** 2.2.2 h<ttp://swupdate.openvpn.org/community/releases/openvpn-2.2.2-install.exe>   
  
**What is working:**   
  
Ipod is able to connect to VPN   
Desktop is able to connect to VPN   
All VPN clients can access LAN PC's   
Internet on LAN PC's works perfectly when clients are connected   
  
Now before I go any further I thought I would also add that there is no need for any addition firewall rules for the router except for a few easy startup Commands which I will outline.   
  
Here we go:   
  
Basic Network Topology:   
Server: 192.168.1.0   
LAN: 192.168.1.0   
VPN: 10.10.10.0   
  
Of course feel free to use whatever IP Addressing you like for your VPN network.   
  
========================================   
DD-WRT v24-sp2 (02/03/13) kingkong (SVN revision 20500M) mega   
  
**Services** => VPN tab:   
**OpenVpn Server:** enabled   
**Start Type:** WAN up   
**Config as:** Server   
**Server Mode:** Routed (TUN)   
**Network:** 10.10.10.0   
**Netmask:** 255.255.255.0   
**Port:** 443 (you can use the default 1194)   
**Tunnel Protocol:** UDP (you can also use TCP just make sure to tweak your client config)   
**Encryption Cipher:** Blowfish CBC (I have also played around with this you can use what you want I tried AES-256-CBC and it's fine)   
**Hash Algorithm:** SHA1   
**Advanced Options:** Enabled   
**TLS Cipher:** None   
**LZO COmpression:** Yes   
**Redirect default Gateway:** Disable   
**Allow Client to Client:** Enable   
**Allow duplicate cn:** Disable (you can enable if you need this)   
**Tunnel MTU setting:** 1500   
**Tunnel UDP Fragment:** empty   
**Tunnel UDP MSS-Fix:** Disable   
  
Certs that I filled in:   
**Public Server Cert:** server cert   
**CA Cert:** CA Cert   
**Private Server Key:** server key   
**DH PEM:** dh1024.pem   
**TLS Auth Key:** Leave empty   
**Certificate Revoke List:** Leave empty   
  
  
If you need to know how to generate certs and keys, please look here: <http://www.howtogeek.com/64433/how-to-install-and-configure-openvpn-on-your-dd-wrt-router/> and look under the section Creating the Certificates and Keys   
  
**Additional Config:**   
  
client-client   
push "route 192.168.1.0 255.255.255.0"   
route 192.168.1.100 255.255.255.0 10.10.10.2   
  
Okay that takes care of the VPN server config and if you have a different LAN address change the IP addresses in the Additional config section to apply to your network. Now to complete the startup commands that I spoke of earlier   
  
Administration => Commands   
Startup:   
mkdir -p /tmp/openvpn/ccd   
echo "iroute 192.168.1.100 255.255.255.0" > /tmp/openvpn/ccd/**ipod**   
echo "iroute 192.168.1.100 255.255.255.0" > /tmp/openvpn/ccd/**laptop1**   
  
Okay so these lines that start with **echo** are important the bits that I have bolded need to be the same **common name** that you entered when generating your client keys. If you don't know what I am referring to then please see section where I mention how to create your certs and keys. Also the IP addresses above are the range that the DHCP addresses start from for my network that the router assigns. So if you have something different please change those because I believe that these are the commands that allow the VPN network to speak with your LAN.   
  
  
-------------------------------------   
  
Now the client config:   
  
client   
dev tun   
proto udp (if using TCP use "**proto tcp**" instead)   
remote your.ipaddress.com 443 (remember port needs to match what you configured so say 1194 if you used the default)   
ca ca.crt   
cert **ipod.crt**   
key **ipod.key**   
ns-cert-type server   
comp-lzo   
verb 3   
  
Now the bolded bits are the names of the client key I generated. So obviously these will change based on what you name your client keys. This config I typed into notepad and then saved it as **client.ovpn**   
  
To import the files into IOS use ITunes and under apps and click on the OpenVPN icon and then add these files:   
  
**client.ovpn**   
**ca.crt**   
**ipod.crt** (this will change based on what you named your keys)   
**ipod.key** (this will change based on what you named your keys)   
  
-----------------------   
Now for the laptop running Windows 7 I installed OpenVPN 2.2.2   
  
remote your.ipaddress.com 443 (I put this line first, I read it somewhere that its good to do that)   
client   
dev tun   
proto udp   
ca ca.crt   
cert **laptop1.crt** (change to your cert name)   
key **laptop1.key** (change to your corresponding key name)   
ns-cert-type server   
comp-lzo   
verb 3[/color]   
  
This config I typed into notepad and then saved it as **client.ovpn**   
  
In windows 7 64 bit navigate to this folder:   
  
**C:\Program Files(x86)\OpenVPN\config**   
  
**client.ovpn   
ca.crt   
laptop1.crt   
laptop1.key**   
  
Also run the OpenVPN GUI as "**Administrator**" and then click connect.   
  
-------------------   
  
That's pretty much it. I will try and answer any questions that you may have. I just wanted to share this because it's been really tough to find any tutorials that use the TUN connection and relates to DD-WRT specifically. I have found a few based on TAP but that does not allow for any connections from IOS devices and all of the tutorials I could find were based on older builds which required firewall rules.