English • Deutsch • Español • Français • Italiano • ??? • Polski • Português • ??????? • Svenska • **???(????)?** • ???(??)? •

Contents

??

SSH tunneling allows you to forward traffic from one location to another using encryption between them. It is great for accessing your home network from remote locations such as your workplace or public WIFI hotspots. You can also use it to securely browse the internet by forwarding your traffic from the remote location to your home and then out to the internet unencrypted from your home. This can allow you to bypass firewall restrictions at the remote location.

??

- You will need a firmware version that supports SSH for your home router.
- You will need a SSH client running at the remote location.

??DD-WRT ? SSH ???

1. Go to the **Services** tab and the **Services** sub-tab on the Web Interface.
2. Enable **SSHd** in the **Secure Shell** section.
3. **SSH TCP Forwarding** can be left disabled.
4. **Port** can be left set to 22.
5. Either enable **Password Authorization** (less secure but easier to set up) or see the main SSH Wiki page for instructions how to set up **Authorized Keys**.
6. Press the **Apply Settings** button.
7. Go to the **Administration** tab and the **Management** sub-tab on the Web Interface
8. Enable **SSH Management** in the **Remote Access** section.

9. Set the **Remote Port** to the TCP port number that you want to use to connect to your router from the internet. Port 443 is a good choice because it is typically left open to allow HTTPS usage, so your client will be able to connect even through very restrictive firewalls.
10. Press the **Apply Settings** button.

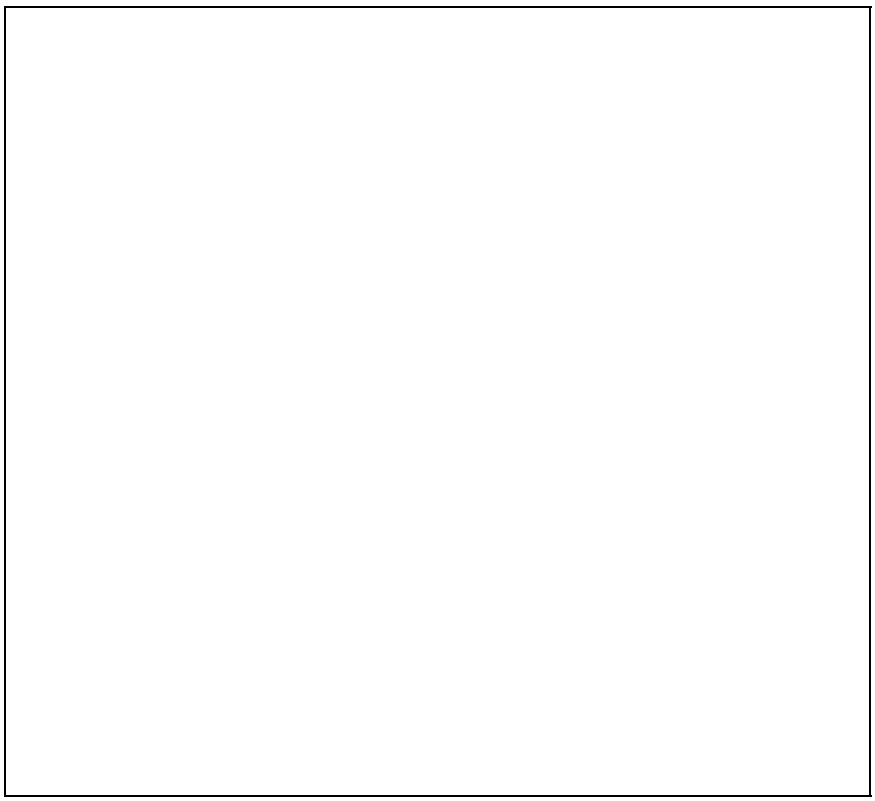
If you haven't already got one, you'll likely want to configure a Dynamic DNS service, especially if your ISP provides you a dynamic IP address. This will allow you to access your router over the internet using an easy to remember domain name instead of the WAN IP address.

?? SSH ???

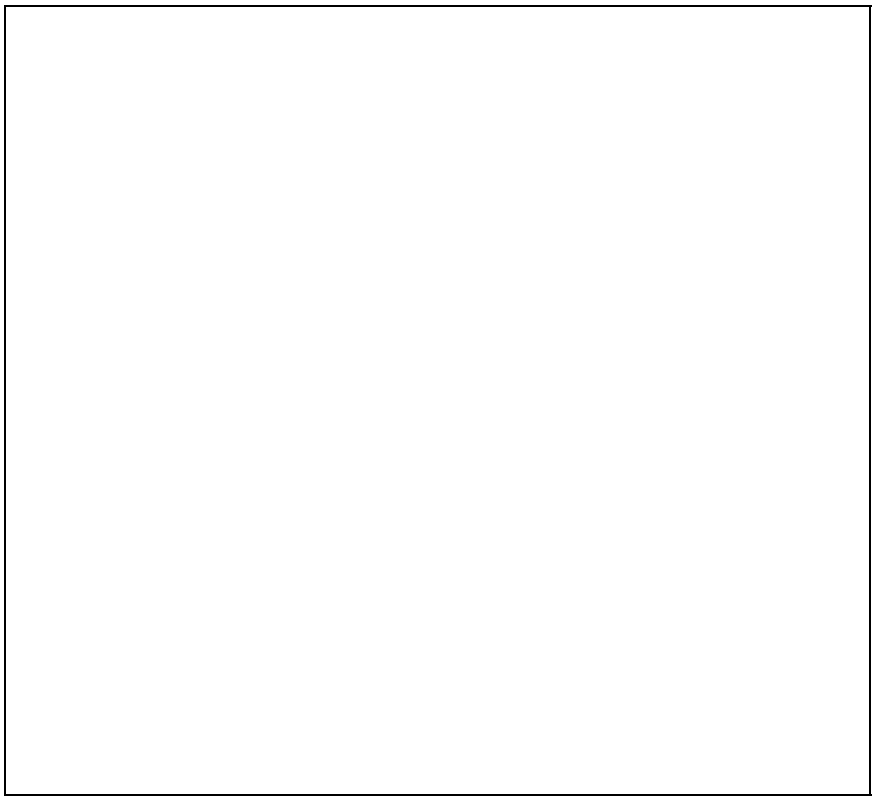
PuTTY ???

PuTTY is a graphical SSH client for Windows OS that is available for download [here].

1. Install and execute PuTTY on the client PC.
2. Set the **Host Name (or IP Address)** to either your home router's dynamic DNS domain name or its public Internet address. (in the picture below it's a private LAN address 10.0.0.1)
3. Set the **Port** to the **Remote Port** port that you set in Step 9 of the Server Configuration.
4. Set the **Connection type** to **SSH**.



1. Go to the **Connection -> Data** section.
2. Set **Auto-login username** to **root** so that you don't have to type the username each time you connect.
3. Go to the **Connection -> SSH -> Tunnels** section.
4. Type **8080** into the **Source port**.
5. Click on the **Dynamic** radio button to make it a dynamic tunnel that will act as a SOCKS proxy server.
6. Click on the **Add** button to add it to the list of forwarded ports. It will appear as **D8080** in the list.



1. If you set the server up to use an authentication key, then go to the **Connection -> SSH -> Auth** section and enter the location for your key file.
2. Go back to the **Session** section.
3. Enter a name you want to call the set of settings into the **Saved Sessions** field and press the **Save** button.
4. You can now double click the saved session name in the list to load it and open the connection to your SSH server.
5. Each time you connect you will need to enter your router password unless you're using an authentication key.

??????

Linux, Mac OS, and many other OS's come with command line SSH programs already installed. All you need to do is open a shell and issue this command with your SSH server's address filled in.

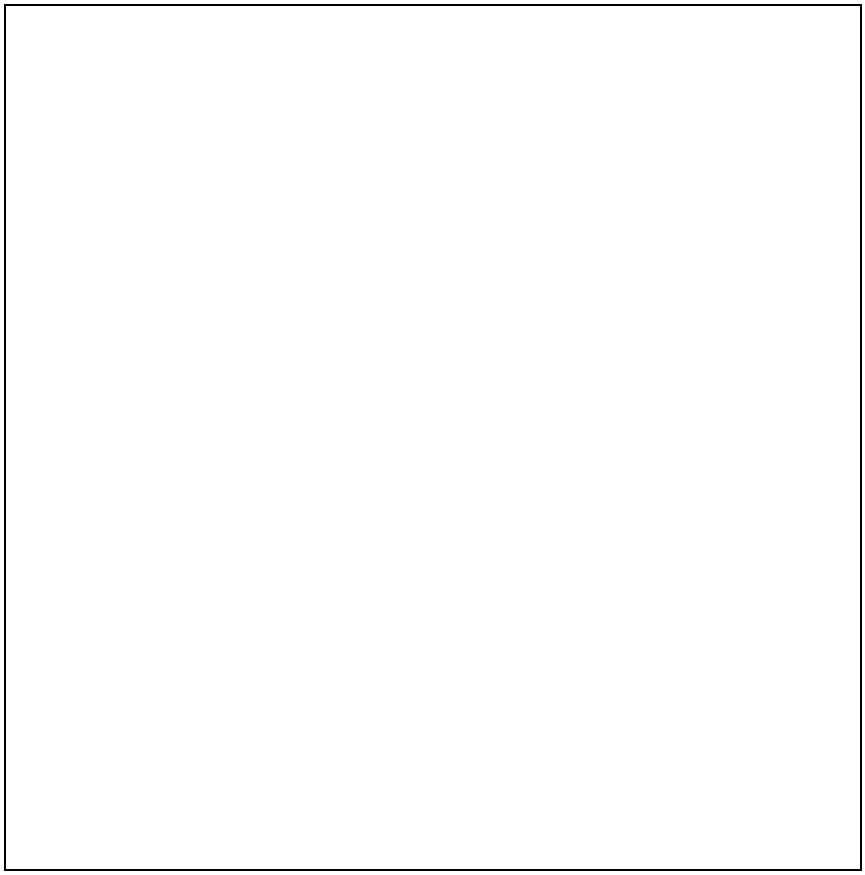
```
ssh root@[SSH server's IP or domain name] -p 443 -D 8080
```

If you use an authentication key, then consult your OS's documentation for details on how to use them with the ssh command or where to place the key file to have it automatically used. Otherwise you will be prompted to enter your password each time you connect. An added advantage of using a key file with the command line ssh program is that you can run it hidden as a background process instead of having to leave the shell open.

????????

Now that you have SSH running, all you need to do is configure your browser or other program to connect to the SOCKS proxy running on the client machine's port 8080.

1. Go to your program's connection settings.
2. Set the SOCKS Proxy settings to use address **localhost** port **8080**



With your browser configured to proxy over the SSH tunnel, visit a site that will tell you what your IP address is, such as [whatsmyip.org], and check that your IP address using the proxy matches the public Internet address of your home router.

You will have to have the SSH connection open whenever you want to utilize it for proxying. When you're done using the tunnel, change your program's settings to not use it anymore or else they will not have connectivity while the tunnel is down.

??