Linux Fu: VPN For Free With SSH

Originalartikel

Backup

<html> <p>If you see a lot of banner ads on certain websites, you know that without a Virtual Private Network (VPN), hackers will quickly ravage your computer and burn down your house. Well, that seems to be what they imply. In reality, though, there are two main reasons you might want a VPN connection. You can pay for a service, of course, but if you have ssh access to a computer somewhere on the public Internet, you can set up your own VPN service for no additional cost.</p><p>The basic idea is that you connect to a remote computer on another network and it makes it look like all your network traffic is local to that network. The first case for this is to sidestep or enhance security. For example, you might want to print to a network printer without exposing that printer to the public Internet. While you are at the coffee shop you can VPN to your network and print just like you were a meter away from the printer at your desk. Your traffic on the shop&#8217;s WiFi will also be encrypted.</p><p>The second reason is to hide your location from snooping. For example, if you like watching the BBC videos but you live in Ecuador, you might want to VPN to a network in the UK so the videos are not blocked. If your local authorities monitor and censor your Internet, you might also want your traffic coming from somewhere else.</p><p>Using SSH for VPN will work for both cases, although if you are mostly interested in the first case, you are probably going to be happier using a dedicated router or a small computer like a Raspberry Pi dedicated to the task. However, if you are leasing a server somewhere, that option isn&#8217;t going to work for you.</p><h2>Prerequisites</h2><p><a href=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg“ target=„_blank“><img data-attachment-id=„435640“ data-permalink=„https://hackaday.com/2020/11/23/linux-fu-vpn-for-free-with-ssh/map-12/“ data-orig-file=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg“ data-orig-size=„800,369“ data-comments-opened=„1“ data-image-meta=„{&quot;aperture&quot;:&quot;0&quot;,&quot;credit&quot;:&quot;&quot;,&quot;camera&quot;:&quot;&quot;,&quot;caption&quot;:&quot;&quot;,&quot;created_timestamp&quot;:&quot;0&quot;,&quot;copyright&quot;:&quot;&quot;,&quot;focal_length&quot;:&quot;0&quot;,&quot;iso&quot;:&quot;0&quot;,&quot;shutter_speed&quot;:&quot;0&quot;,&quot;title&quot;:&quot;&quot;,&quot;orientation&quot;:&quot;0&quot;}“ data-image-title=„map“ data-image-description=„“ data-image-caption=„“ data-medium-file=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg?w=400“ data-large-file=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg?w=800“ class=„alignright wp-image-435640 size-medium“ src=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg?w=400“ alt=„“ width=„400“ height=„185“ srcset=„https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg 800w, https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg?resize=250,115 250w, https://hackaday.com/wp-content/uploads/2020/10/map-1.jpg?resize=400,185 400w“ referrerpolicy=„no-referrer“ /></a>You really only need root access to both machines and SSH server on the remote machine along with the SSH client. There is some configuration required on both sides. I use KDE so I used NetworkManager to set things up, although that isn&#8217;t necessary. It just makes things easier.</p><p>The server needs a few special items set up, but those items may already be present. In

```
/etc/ssh/sshd_config
```

you will want

```
PermitTunnel=yes
```

and you may need to set

```
AllowTCPForwarding
```

to yes, as well.&#160; The firewall may need some tweaks, too. The setup instructions for the <a href=„https://github.com/danfruehauf/NetworkManager-ssh“ target=„_blank“>NetworkManager plug-in</a> will be useful even if you don&#8217;t want to use it.</p><h2>Client Side</h2><p>If you are using NetworkManager, you&#8217;ll need the plug-in. For Neon and other Debian-type distributions, you can find the

```
network-manager-ssh
```

package and that&#8217;s all you need. If you don&#8217;t want to use it, you can use this line from the plug-in author&#8217;s <a href=„https://bashinglinux.wordpress.com/2013/03/23/networkmanager-ssh/“ target=„_blank“>blog</a>:</p><pre class=„brush: plain; title: ; notranslate“ title=„“>ssh -f -v -o Tunnel=point-to-point -o ServerAliveInterval=10 -o TCPKeepAlive=yes -w 100:100 root@YOUR_SSH_SERVER \'/sbin/ifconfig tun100 172.16.40.1 netmask 255.255.255.252 pointopoint 172.16.40.2' &amp;&amp; \/sbin/ifconfig tun100 172.16.40.2 netmask 255.255.255.252 pointopoint 172.16.40.1</pre><p>You will need to be root on both ends because you are creating a tunnel device. This leads to a few problems, even if you use the plug-in. Obviously, you aren&#8217;t going to want SSH bugging you for passwords and host key verifications, but if you establish the VPN manually, you could deal with that.</p><h2>Problems</h2><p>However, most modern systems don&#8217;t allow root login with a password, or even at all. So you&#8217;ll need to fix that first. In addition, when the NetworkManager runs SSH, it will be looking for host keys and such as root, not as your user. If it can&#8217;t find things, it will just die. So you&#8217;ll need to make sure that root can log in with no intervention.</p><p>To allow root logins to the server, you need to edit

```
/etc/ssh/sshd_config
```

and change

```
PermitRootLogin
```

to yes. I suggest you do this only long enough to do the next few steps. You&#8217;ll need to restart the

```
sshd
```

server which means something like:</p><pre>systemctl restart sshd</pre><p>or</p><pre>/etc/init.d/ssh restart</pre><p>Then, logged in as your normal user on your local machine, use

```
ssh-copy-id
```

to install your certificate to the host computer. As soon as that works, you should go back and change

```
/etc/ssh/sshd_config
```

to use &#8220;

```
PermitRootLogin prohibit-password
```

.&#8221; That way you can log in as root with a certificate, but not with a password.</p><p>If you&#8217;ve logged on from your root account once, SSH probably asked you if you want to accept the server key. If not, that&#8217;s going to be a problem. If you can, log in and answer yes so it quits asking. However, if you can&#8217;t, we can also turn off

```
StrictHostKeyChecking
```

.</p><p>In theory, you can pass extra ssh options to the NetworkManager plugin, but for some reason that doesn&#8217;t work on the version from the repositories. If you are starting manually, of course, you can add what you want. However, it is also possible to set root&#8217;s SSH configuration in

```
/root/.ssh/config
```

or the global configuration at

```
/etc/ssh/ssh_config
```

.</p><p>If you do change the global, consider using

```
/etc/ssh/ssh_config.d
```

if your system supports it. That lets you put snippets in for a particular host that won&#8217;t get written over on system upgrades. For example, you might make a file in that directory named

```
hackaday.conf
```

:</p><pre class=„brush: plain; title: ; notranslate“ title=„“>Host *.hackaday.com hackaday.comStrictHostKeyChecking noTunnel yes</pre><p>Again, if you object to the host key checking, then just log in from your root account once and manually accept the remote key. Or, if you are brave, manually edit

```
/root/.ssh/known_hosts
```

.</p><h2>Prosper</h2><p>That should do it. If you are using the NetworkManager plug in, just make a new connection. From there, pick the VPN connections section and select SSH.</p><p><a href=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png“ target=„_blank“><img data-attachment-id=„435220“ data-permalink=„https://hackaday.com/2020/11/23/linux-fu-vpn-for-free-with-ssh/vpn0/“ data-orig-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png“ data-orig-size=„905,730“ data-comments-opened=„1“ data-image-meta=„{&quot;aperture&quot;:&quot;0&quot;,&quot;credit&quot;:&quot;&quot;,&quot;camera&quot;:&quot;&quot;,&quot;caption&quot;:&quot;&quot;,&quot;created_timestamp&quot;:&quot;0&quot;,&quot;copyright&quot;:&quot;&quot;,&quot;focal_length&quot;:&quot;0&quot;,&quot;iso&quot;:&quot;0

&quot;,&quot;shutter_speed&quot;:&quot;0&quot;,&quot;title&quot;:&quot;&quot;,&quot;orientation&quot;:&quot;0&quot;}“ data-image-title=„vpn0“ data-image-description=„“ data-image-caption=„“ data-medium-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png?w=400“ data-large-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png?w=775“ class=„aligncenter wp-image-435220“ src=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png“ alt=„“ width=„492“ height=„397“ srcset=„https://hackaday.com/wp-content/uploads/2020/10/vpn0.png 905w, https://hackaday.com/wp-content/uploads/2020/10/vpn0.png?resize=250,202 250w, https://hackaday.com/wp-content/uploads/2020/10/vpn0.png?resize=400,323 400w, https://hackaday.com/wp-content/uploads/2020/10/vpn0.png?resize=775,625 775w“ referrerpolicy=„no-referrer“ /></a></p><p>You&#8217;ll have to put in a few parameters, including the certificate you want to use to log in to the remote machine:</p><p><a href=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png“ target=„_blank“><img data-attachment-id=„435221“ data-permalink=„https://hackaday.com/2020/11/23/linux-fu-vpn-for-free-with-ssh/vpn1/“ data-orig-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png“ data-orig-size=„472,861“ data-comments-opened=„1“ data-image-meta=„{&quot;aperture&quot;:&quot;0&quot;,&quot;credit&quot;:&quot;&quot;,&quot;camera&quot;:&quot;&quot;,&quot;caption&quot;:&quot;&quot;,&quot;created_timestamp&quot;:&quot;0&quot;,&quot;copyright&quot;:&quot;&quot;,&quot;focal_length&quot;:&quot;0&quot;,&quot;iso&quot;:&quot;0&quot;,&quot;shutter_speed&quot;:&quot;0&quot;,&quot;title&quot;:&quot;&quot;,&quot;orientation&quot;:&quot;0&quot;}“ data-image-title=„vpn1“ data-image-description=„“ data-image-caption=„“ data-medium-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png?w=219“ data-large-file=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png?w=343“ class=„aligncenter wp-image-435221“ src=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png“ alt=„“ width=„319“ height=„583“ srcset=„https://hackaday.com/wp-content/uploads/2020/10/vpn1.png 472w, https://hackaday.com/wp-content/uploads/2020/10/vpn1.png?resize=137,250 137w, https://hackaday.com/wp-content/uploads/2020/10/vpn1.png?resize=219,400 219w“ referrerpolicy=„no-referrer“ /></a></p><p>Once you save the connection, you can activate it like you would any other network interface. If you want to see if it works, ask a website for <a href=„https://www.myip.com“ target=„_blank“>your IP address</a>. Then activate the VPN and do it again. If you have trouble getting the VPN to connect, you can look in the system log to find out what errors SSH is throwing.</p><h2>Of Course&#8230;</h2><p>There are other VPN solutions. However, since it is almost a sure bet that your remote computer has an SSH server on it, this is very simple to set up with very little planning.</p><p>You can do a lot with SSH if you <a href=„https://hackaday.com/2019/12/17/linux-fu-stupid-ssh-tricks/“>know the tricks</a>. We especially like using it to <a href=„https://hackaday.com/2020/09/21/linux-fu-simple-ssh-file-sharing/“>mount files</a>.</p> </html>