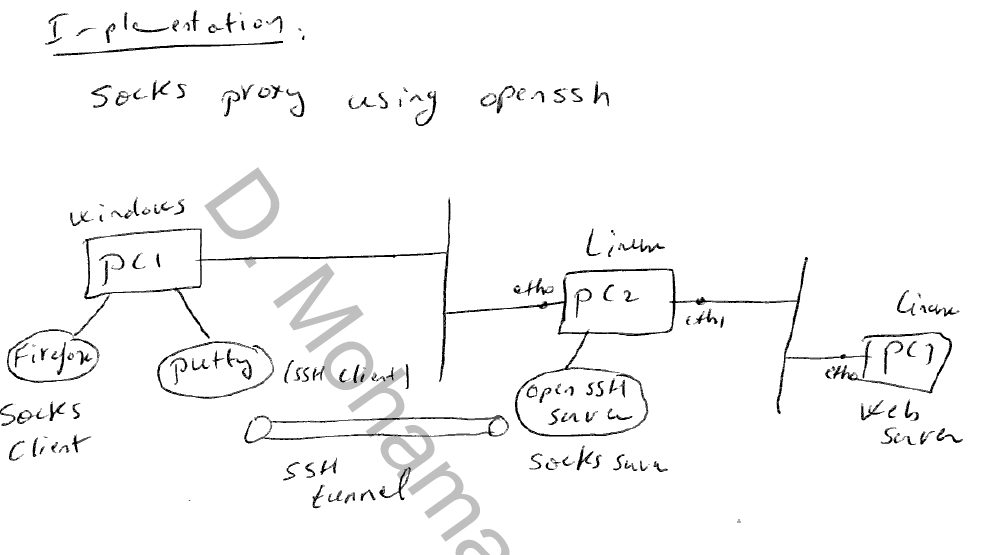
Socks Proxy Using Openssh

Description



1. **Proxy Firefox through a SSH tunnel**

Afast, privately secured tunnel to transfer web pages

Have you ever wanted to visit sites during the day from a location that denied access to those sites? Perhaps the company has denied access due to bandwidth considerations or you might have decided that the site you want to go to might not always be work safe depending on the story or pictures? What you need is the ability to create a secure and encrypted ssh connection to tunnel your browser traffic through.

Using a ssh tunnel to retrieve the data from websites is significantly faster than trying to use X forwarding to open a remote copy of Firefox on the remote machine. If a remote browser is used the connection will be saturated by the graphical front end of the remote browser window. Use the tunnel for the web site's data and leave the rendering of the browser to the local machine. This is the most efficient solution.

If you have access to a remote machine by way of ssh you can set up Firefox, or any other SOCKS v5 enabled application, to tunnel its connection through ssh. This way, if you were at work and wanted to browse your favorite sites like MySpace, Facebook or Maxim that are blocked at the company firewall you could.

1. PuTTY and SSH

If you haven’t remotely administered a Unix/Linux-like server before, you probably haven’t heard of Secure Shell (SSH). Secure Shell is simply a network protocol that allows for encrypted communication between two computers — usually yours, and a remote server. “Shell” refers to the command-line interface (CLI) that is present on almost every kind of computer, including Mac OS X and Windows. SSH is typically used to securely access a remote computer’s CLI, but it can also be used to copy files — or it can be used as a tunnel between your computer and another computer on the internet.

PuTTY is an SSH client. You can use it to access a remote CLI, or you can use it to set up a tunnel — and that’s what we’re going to do now.

1. Tunneling

When you type a URL or click a link, a request travels from your computer, through the local router and modem, over your ISP’s network, across the internet, and into the remote web server. Your request can be filtered at any stage, but generally it’s at the local router (the school/corporate firewall) or at the ISP (traffic shaping, federal censorship).

Tunneling bypasses the local router, modem, and your ISP’s network, and connects you directly to the internet. If you’re in China, for example, SSH tunneling all of your traffic through a computer in America will bypass any national-level filtering and censorship. The actual act of forcing your web traffic through another computer (and another port) is called [SOCKS proxying](http://en.wikipedia.org/wiki/SOCKS), incidentally — and you can SOCKS proxy without SSH, but it’s less secure and more likely to be filtered by your local ISP.

1. PuTTY and SSH

If you haven’t remotely administered a Unix/Linux-like server before, you probably haven’t heard of Secure Shell (SSH). Secure Shell is simply a network protocol that allows for encrypted communication between two computers — usually yours, and a remote server. “Shell” refers to the command-line interface (CLI) that is present on almost every kind of computer, including Mac OS X and Windows. SSH is typically used to securely access a remote computer’s CLI, but it can also be used to copy files — or it can be used as a tunnel between your computer and another computer on the internet.

PuTTY is an SSH client. You can use it to access a remote CLI, or you can use it to set up a tunnel — and that’s what we’re going to do now.

1. Tunneling

When you type a URL or click a link, a request travels from your computer, through the local router and modem, over your ISP’s network, across the internet, and into the remote web server. Your request can be filtered at any stage, but generally it’s at the local router (the school/corporate firewall) or at the ISP (traffic shaping, federal censorship).

Tunneling bypasses the local router, modem, and your ISP’s network, and connects you directly to the internet. If you’re in China, for example, SSH tunneling all of your traffic through a computer in America will bypass any national-level filtering and censorship. The actual act of forcing your web traffic through another computer (and another port) is called [SOCKS proxying](http://en.wikipedia.org/wiki/SOCKS), incidentally — and you can SOCKS proxy without SSH, but it’s less secure and more likely to be filtered by your local ISP.

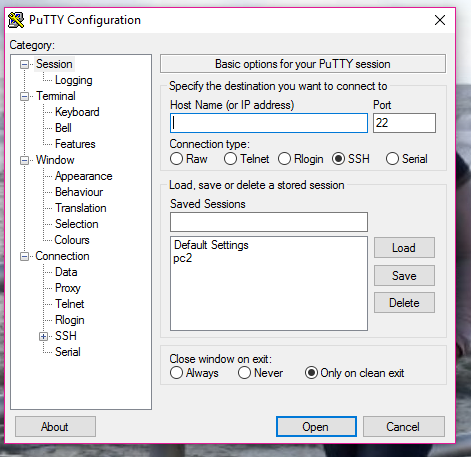
1. Setting up a tunnel

First, [download PuTTY](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) (putty.exe).

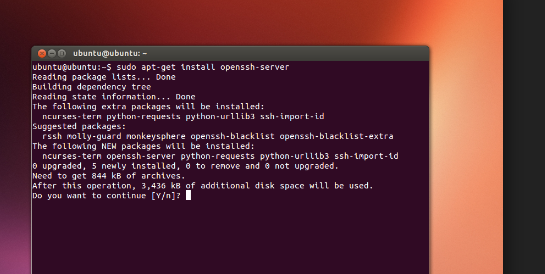
**PuTTY** is a popular SSH and **Telnet client** that helps you establish secure connections over the Internet and doesn't even require installation.

It's especially aimed for programmers and network administrators, which means that newcomers won't find it easy to use. The program features a simple, straightforward interface with no included documentation.

Despite its apparent simplicity, PuTTY is highly configurable and includes many options to tweak connections, sessions, SSH security features and even the window's appearance.

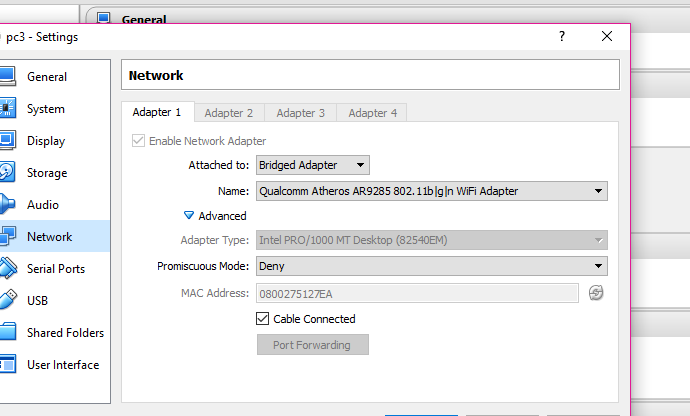


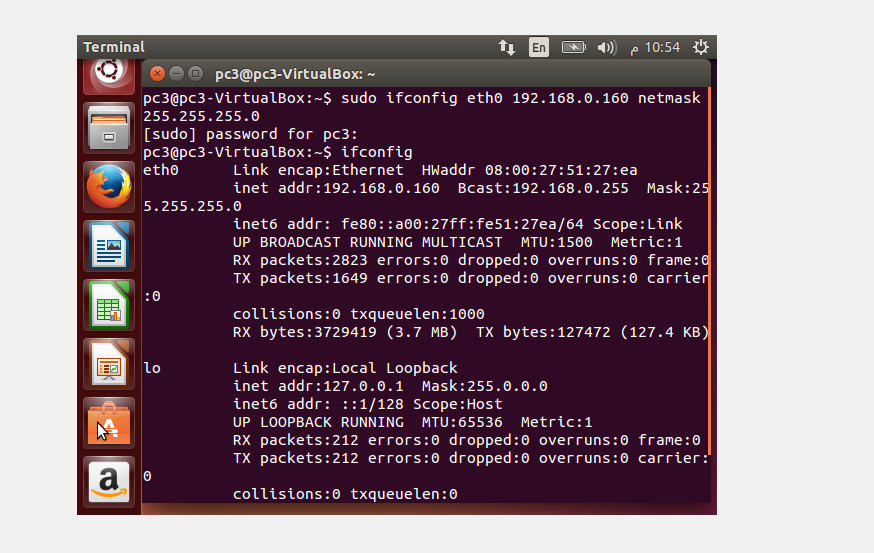
And on PC2 SSH server:



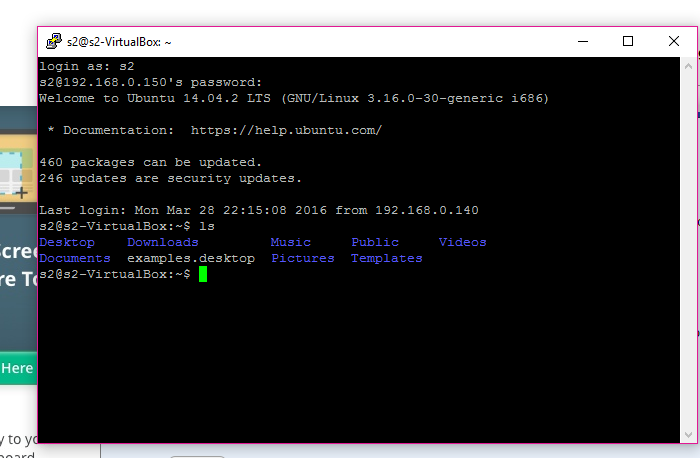
Here you can load a saved session here is pc2 for example, however before doing this you have to configure your ip’s address so you can reach the other machine so all the machines are on the same LAN.

On Linux:

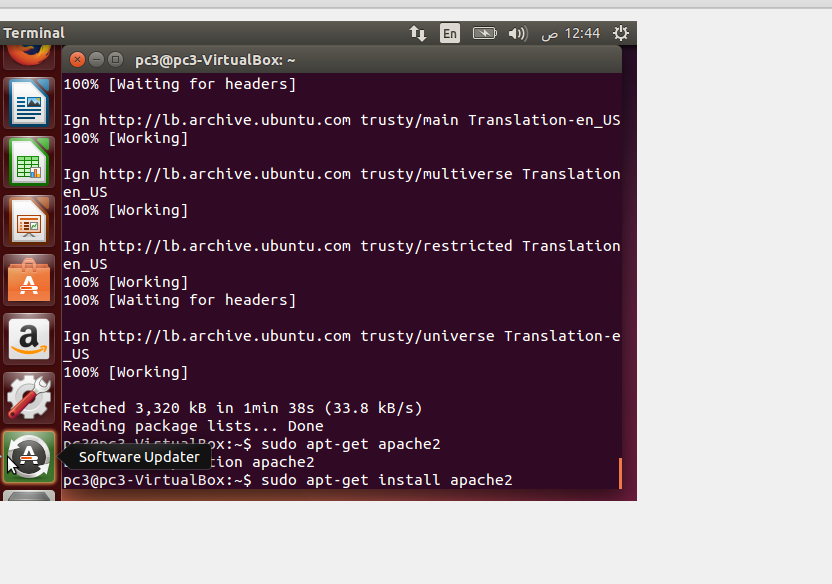




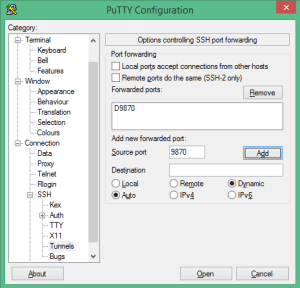
Now you can connect to the other machine and reach it’s command line and write any command you want:



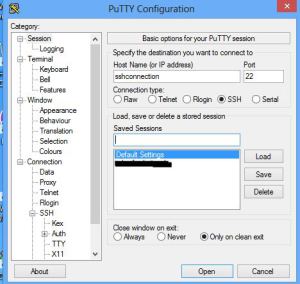
Now you need a webserver on PC3:



From Putty, go down to Connection > SSH > Tunnels. From this area, add in a source port (I’ve chosen 9870 but this can be any unused port), and choose ‘Dynamic’. Then click ‘Add’ and you’ll get an entry under “Forwarded Ports”:

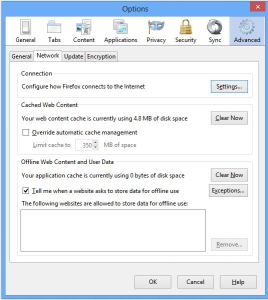
[](http://i2.wp.com/www.adamfowlerit.com/wp-content/uploads/2013/01/putty.png)

Go back to Session and connect to SSH. You can save your session to load later rather than retyping the details:

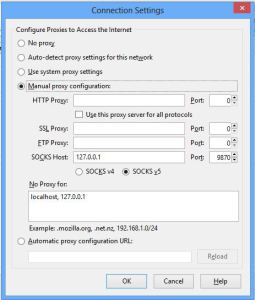
[](http://www.adamfowlerit.com/2013/01/05/using-firefox-with-a-putty-ssh-tunnel-as-a-socks-proxy/putty2/)  
Click “Open” and log onto your SSH session. You only need to log on with your credentials and get to a bash prompt.

Open Firefox, and go to the website [http://whatismyip.com](http://whatismyip.com/) – take note of your external IP address, because this should change once you’re tunneling through SSH.

In Firefox, press the ‘alt’ key to bring up the top menu and go into Tools > Options. Under the “Advanced” area, go to the “Network” tab and in the “Connection” area click the “Settings” button.

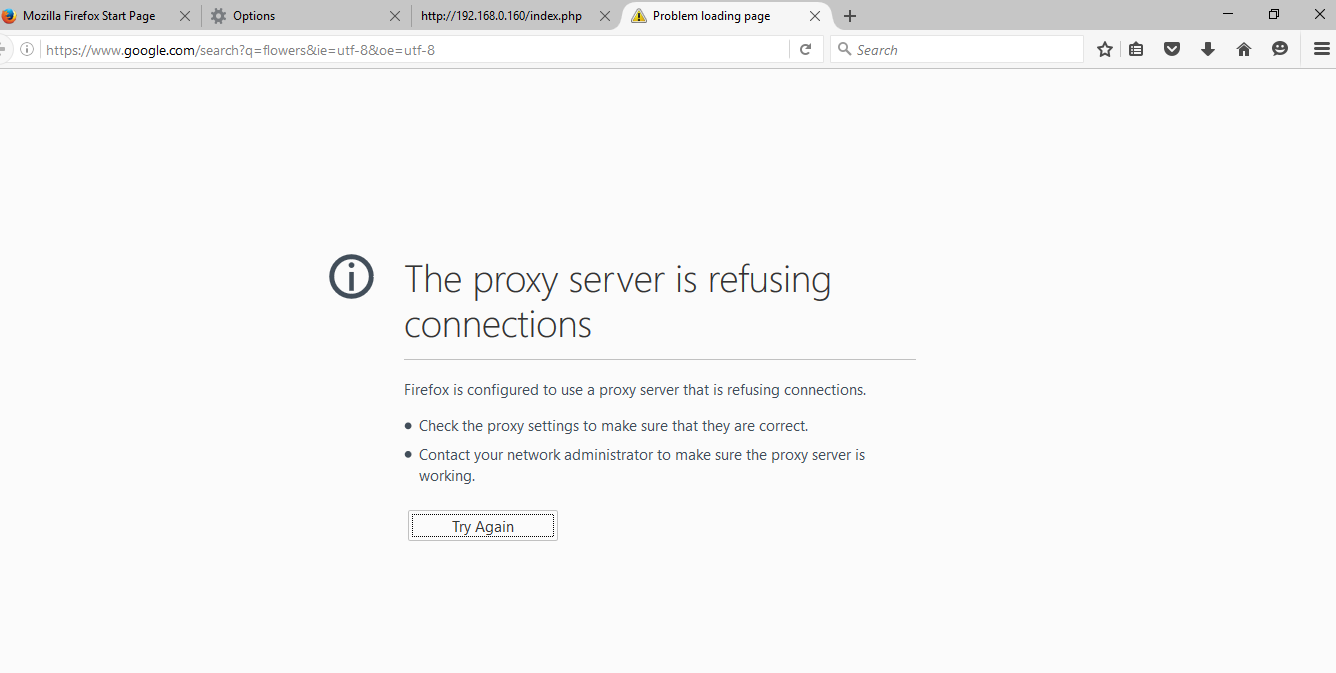
[](http://www.adamfowlerit.com/2013/01/05/using-firefox-with-a-putty-ssh-tunnel-as-a-socks-proxy/firefox1/)

Choose the “Manual proxy configuration” radio button, and under SOCKS Host enter “127.0.0.1” and the port you entered into Putty (in my case it’s 9870). Make sure SOCKS v5 is selected.

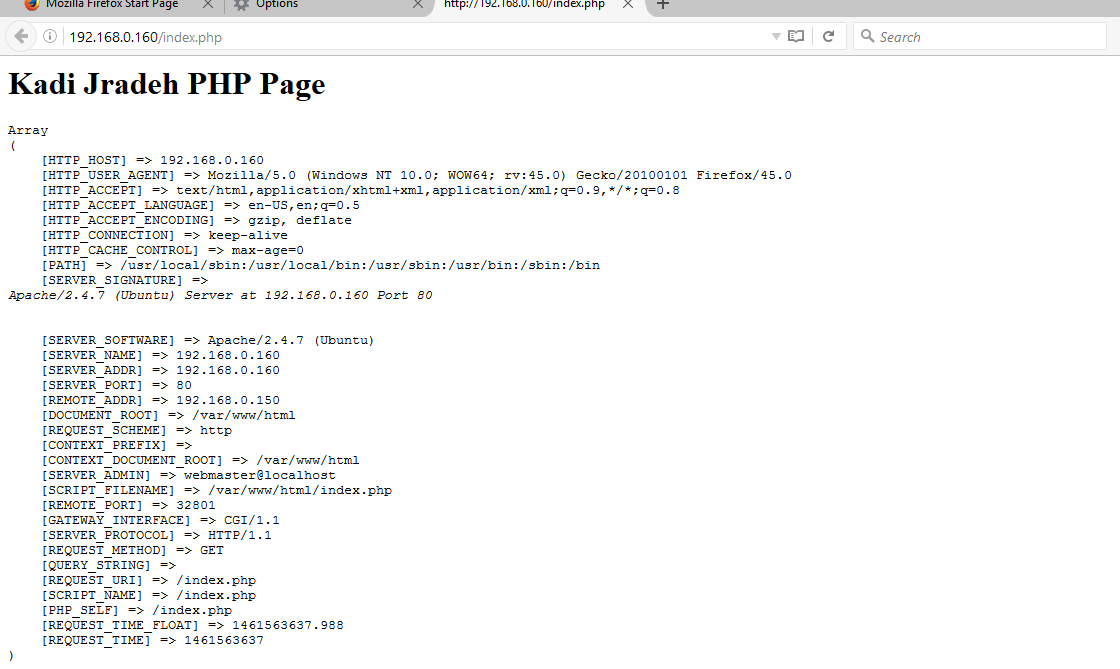
[](http://www.adamfowlerit.com/2013/01/05/using-firefox-with-a-putty-ssh-tunnel-as-a-socks-proxy/firefox2/)

Press OK twice to get back to the main view of the Firefox Browser. Try again to get to [http://whatismyip.com](http://whatismyip.com/) – if all is working, you’ll see a different IP address which is from the server you’re SSH’d to. If something isn’t configured correctly, you’ll more likely see a browser error like “The proxy server is refusing connections”.

If there is no ssh connection you will be unable to access the internet:



Whereas if all is right:



And by that all is done.

Future development includes stopping traffic from other browsers.

Done by Khadija Jradeh & Zahraa Al-Ashkaar under the supervision of Dr. Mahmud Shaito.