Short tutorial on:

SSH Tunneling and SSH Port forwarding

By

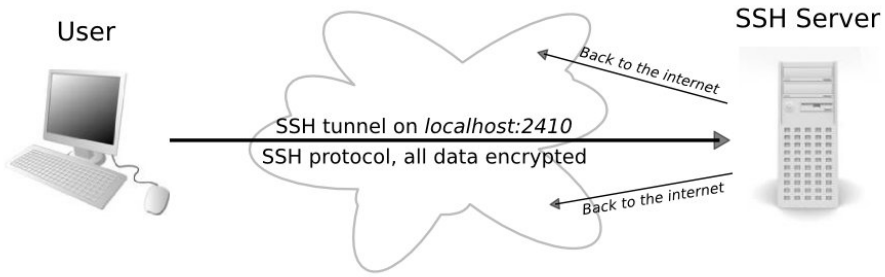
n3tpr0b3 [at] gmail [.] com

Security plays an important role in our every-day internet life. As the number of internet users increases so do the methods for protecting the privacy (read, LONG DEAD) of the people increase. SSH tunneling is a method for protecting your web traffic through the SSH protocol and provide you the privacy you need above all other methods (IMHO). You'll need to know what SSH is, what a shell is and other stuff in order to understand the content of this tutorial. Firs, I want to apologize for my English, I might make some mistake here and there cos' English is not my native language, but I'll try to keep it correct as I can possibly do.

I'll provide a code example on how to accomplish the statements using the OpenSSH-client installed on a GNU/Linux box.

Section 1, SSH Tunneling

SSH tunneling can be accomplished by having only one box to which you can SSH to. This is the block diagram on how the traffic goes on:



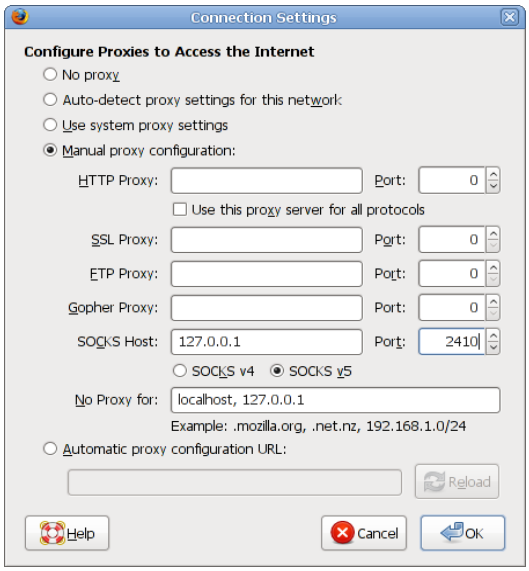
We can see that what we actually do is *tunnel* our traffic (it could be http traffic, ftp or any) through the ssh server back to the internet. The main catch here is that the traffic is encrypted when we talk to the SSH server using the SSH encryption algorithms. From the SSH server, our data gets transferred back to the internet as normal data and we continue to surf normally.

Code example :

```
ssh -D 2410 user1@sshbox1
```

Lets explain the above code. First we have 'ssh' which is the name for our SSH client. Next we have a command line option '-D' with an argument '2410'. This creates a dynamic port forwarding on the ssh protocol. This port binds to our localhost IP address that is 127.0.0.1. When we successfully login to sshbox1, 127.0.0.1:2410 will act as a SOCK4/5 proxy and will forward all traffic made to that port to sshbox1. At the end we have 'user1@sshbox1' which are our login details for the SSH server, user1 – the user and sshbox1 – the SSH server hostname.

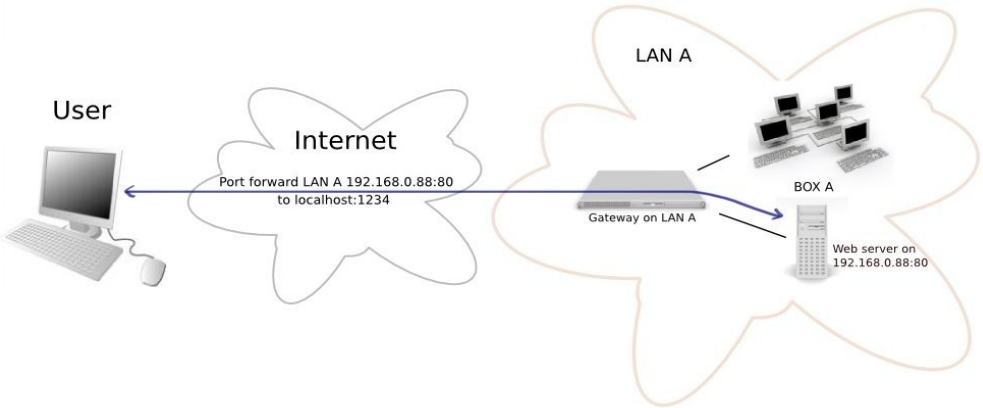
Next is actually using this tunnel. I'm only going to show how we can tunnel our http traffic through this tunnel. We'll use FireFox (cos' firefox is leet =)). Go into the FF options and under Advanced > Network set 127.0.0.1 with a port 2410 as a SOCK5 proxy. Here is a picture :



Now we can surf the web and know that our traffic is encrypted and more secure. The speed of the web browsing may be reduced depending on the speed that the SSH box has to which you are connecting to.

Section 2, SSH Port forwarding

Lets imagine the following scenario: We have hacked or gain SSH access to a gateway that connects lanA to the internet. Next, on lanA there is a box with an LAN IP of 192.168.0.88. On that box, we'll name it boxA, there is a local web server that we want to browse. Because its on the LAN, boxA and its web server listening on port 80, can not be seen from the outdoor, from the internet. We could install a CLI based web client on the gateway and surf the boxA web pages or we could forward port 80 from that box (boxA) to some port that we can connect to on our local PC through the gateway. Here is a diagram on what we are about do to :



Lets explain. We are forwarding '192.168.0.88:80' to localhost:1234 using the SSH access on the gateway. Now, when we access http://localhost:1234 we will actually access 192.168.0.88:80 on boxA, which is in lanA. This method also comes in handy when we want to circle around firewalls and port blocking systems.

Code example :

```
ssh -L 1234:192.168.0.88:80 user@gatewayONlanA
```

Here we have the '-L' command line options followed by the argument '1234:192.168.0.88:80' and then comes our login info on the gateway. The '-L' options tells SSH that we are going to be port forwarding, '1234:192.168.0.88:80' is actually, *localbindip : localport : remoteip : remoteipport* and we tell ssh to bind port 1234 to localhost (if there is no argument for the localbindip it is considered that we want to bind the port to localhost – 127.0.0.1) that we are going to forward from 192.168.0.88:80 through the gateway – through the ssh server.

Well... I hope you learned something. Any comments about this tutorial are welcomed. Contact me through my mail.

Greetz to http://zeroscience.org