Tunnel HTTP thru SSH Using Squid

Created October 3, 2005
by Bruce A. Westbrook

Revisions:
December 12, 2006 – BAW – Updated wording for public distribution

Introduction

This document describes the process of tunneling web traffic out from an unsecured network to ensure that your traffic is not being sniffed. This is ideal for when your out at a conference (particularly security conferences like BlackHat or SANS or DefCon where people are going to be screwing with the networks, especially the wireless, looking for interesting traffic) and you want to keep your traffic as confidential as possible. It also works for other aspects, such as circumventing a secured network in an office environment to avoid the firewall, web content filtering, port filtering, etc. that may be taking place. By tunneling your connections to your own server on the 'net running SSH we will be able to encrypt and therefore hide your traffic.

To accomplish this, I will detail how to perform HTTP tunneling via SSH from your office, conference, wireless web café, etc. to your home and then out to the 'net. You can tunnel other traffic in a similar manner. There are essentially five steps: setup SSH and on your home box, setup Squid on your home box, poke a hole through your home firewall to your SSH box, setup Putty on your remote Windows box, and finally setup your browser.

As long as you have any TCP port out of the network you're on you can setup your remote SSH box to listen on that port. But for the sake of these instructions will assume that wherever you are you have uninhibited access outbound on TCP port 443.

Requirements

The requirements for accomplishing this are pretty simple:

- Home linux box (these instructions are based on Red Hat Fedora)
- Ability to either place your home linux box directly on the Internet (properly protected with iptables, of course) or forward a port through whatever firewall you're using
- A Putty SSH client

Setup SSH

√		Description

	Install SSH	Install the SSH daemon on your home linux box. Note that the specific steps outlined here are for **Red Hat Fedora** distributions. Adopt the steps as needed for your distribution.
		
		`yum install openssh`
		`yum install openssh-clients`
		`yum install openssh-server`

	Configure SSH	Configure SSH to use a port that you can connect to from your office. This may mean using a port other then port 22 for SSH – such as port 80 or 443 if your office is filtering egress traffic that strictly. For these procedures, we'll set it to port 443 (of course, this means you can't also be running a secure web server on port 443 on this box):
		
		`vi /etc/ssh/sshd_config`
		
		Unremark the **`Port`** line and change **22** to **443**
		`Port 443`
		
		Now you need to ensure that iptables is allowing connections to your SSH port. One way to do this is via the GUI:
		
		1. **Application > System Settings > Security Level**
		2. In the **Other Ports** dialog box, type **443:tcp**
		3. Click **OK**
		
		Or from the command line, you would execute something like this:
		
		`iptables –D INPUT –j DROP`
		`iptables –A INPUT –p tcp --dport 443 –j ACCEPT`
		`iptables –A INPUT –j DROP`
		`service iptables save`
		`service iptables restart`

Setup Squid

√		Description
	Install Squid	Install Squid proxy on your home linux box. `yum install squid`

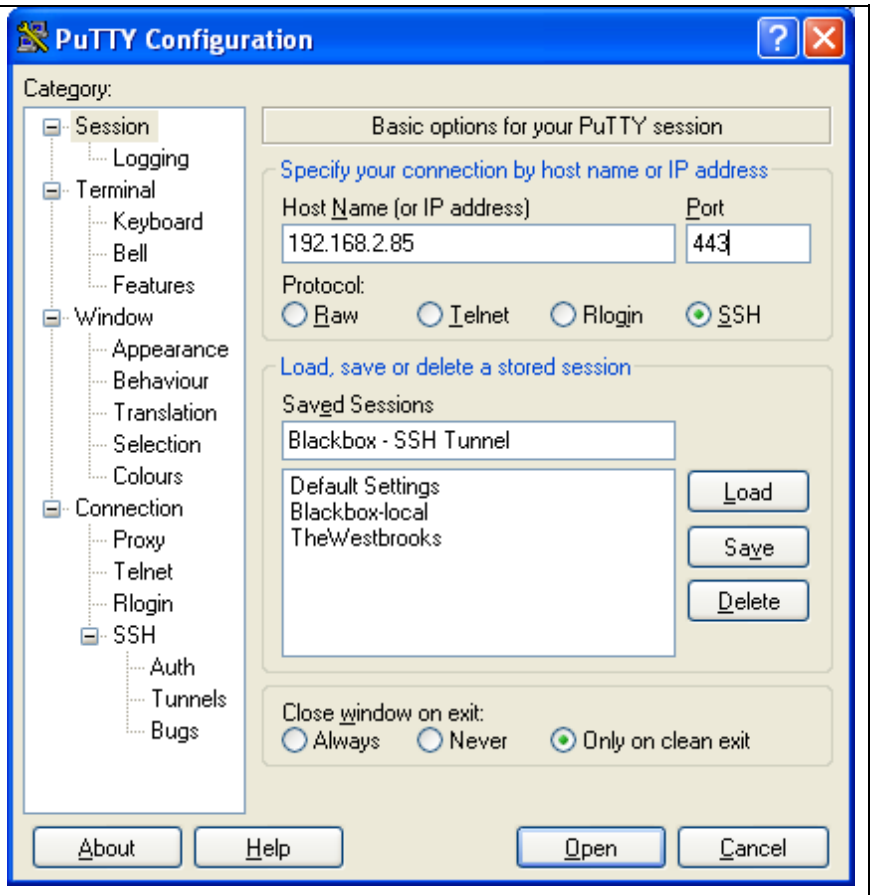
	Configure Squid	To configure squid to listen on the network we need to add three simple lines to the configuration.
		
		1. `vi /etc/squid/squid.conf`
		2. Locate the line `http_access allow our_networks`
		3. Add the following two lines immediately after the above line, substituting your own network schema as appropriate or using 0.0.0.0 to allow everyone:
		`acl localnet src 0.0.0.0/0.0.0.0`
		`http_access allow localnet`
		4. Save and exit the file
		5. Set squid to start automagically at boot
		a. `chkconfig squid on`
		b. `chkconfig --list squid`
		6. Start Squid
		a. `service squid start`

Home Firewall

√		Description
	Open Necessary Port	You now need to open the port on your home firewall to allow access to your SSH/Squid box via the port you configured for SSH. So if you are listening on port 443 for SSH, forward port 443 on your home firewall to your linux box.

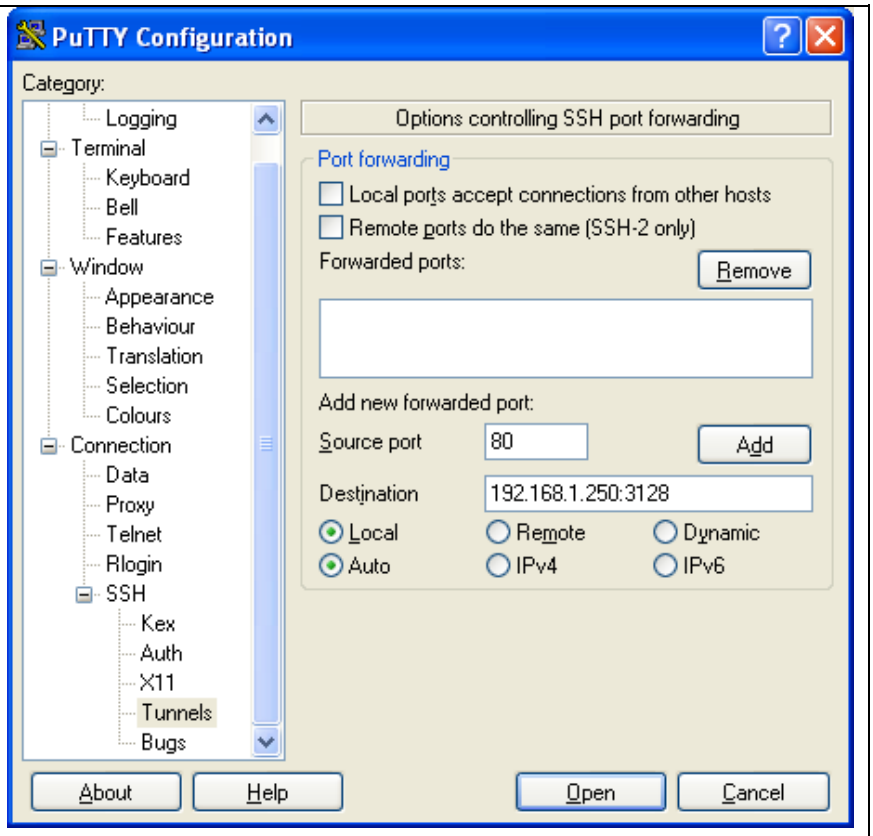
Setup Putty

√		Description
	Install Putty	Download Putty from http://www.chiark.greenend.org.uk/~sgtatham/putty/ Putty is simply an executable – there is not install. To make it easier to user from the CLI, either put the putty.exe in your Windows directory or add a path to wherever you drop putty.exe.
	Configure Putty	To configure Putty to use your SSH tunnel, perform the following: 1. Under **`Session`**, enter your home computer's ***public*** hostname or IP address (note that the example screenshot shown below has a private IP address) 2. For **`Protocol`**, select **`SSH`** 3. For **`Port`**, enter **`443`** 4. For **`Saved Sessions`**, enter a name for your connection (e.g. **`SSH Tunnel`**)

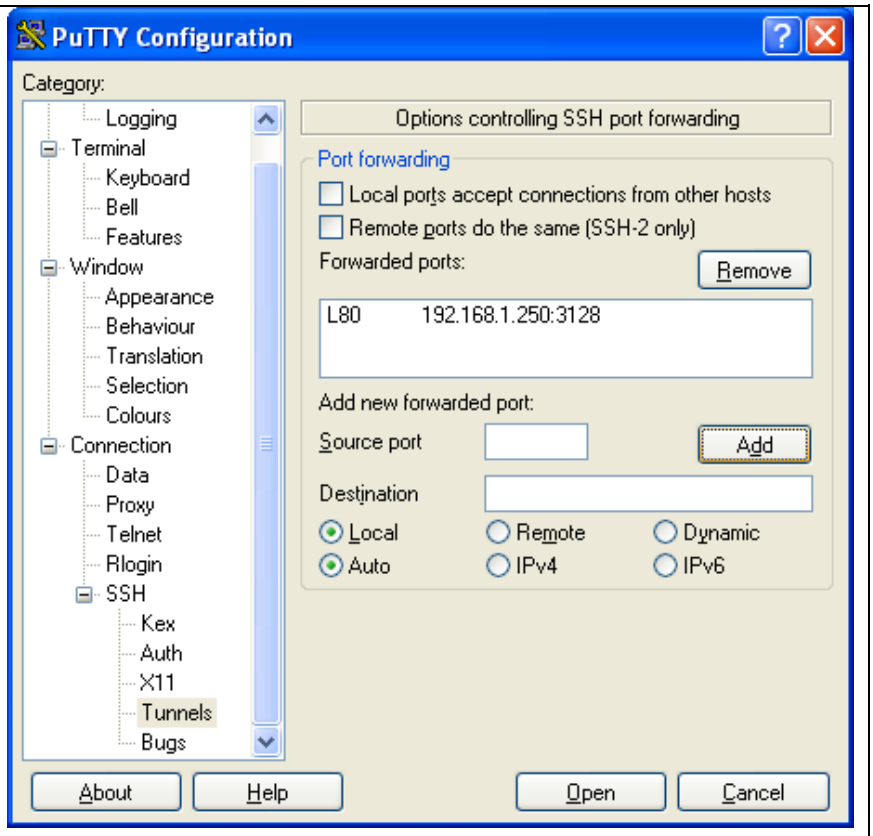


Now we need to configure putty to connect ports 80 and 443 (web browsing) on the local host to the proxy port of 3128 (Squid) on the remote host (your home PC).

5. Navigate to **`Connection`** > **`SSH`** > **`Tunnels`**
6. For **`Source Port`**, enter **`80`**
7. For **`Destination`**, enter the ***private*** IP address of your home linux/squid machine then **`:3128`** (e.g. 192.168.1.250:3128).
8. Select **`Local`**



9. Click **Add**
10. Your **`Forwarded port screen`** should now look like this:

		 11. Navigate back to the **Session** screen and click **Save** 12. Finally, click **Open** to establish your SSH tunnel and login

Setup Browser

√		Description
	Configure your Browser	To use your secure tunnel with a browser, you simply point it to your localhost using the port(s) you setup in the SSH Tunnels portion of Putty. As an example, here's the Internet Explorer settings: 1. Go to **Tools** > **Internet Options** > **Connections** > **LAN Settings > Advanced**. 2. For **Address**, enter **localhost** 3. For **Port**, enter **80** While for Firefox 1.x, the settings are: 1. Go to **Tools > Options > Connection Settings** 2. Select **Manual proxy configuration** 3. For **HTTP Proxy**, enter **localhost** 4. For **Port**, enter **80** You're done! Browse to your heart's delight!