

## Using an SSH Tunnel to Reach a Protected Server – Part 1 – Local Port Forwarding



## EXAMINE A TOOL THAT IS READILY AVAILABLE TO ANYONE

How does one reach a remote server that is in a protected zone? Maybe there is a firewall protecting the server, but you simply must access that precious server. Naturally, exposing the whole server to the internet is simply too risky. In comes the SSH Tunnel to save the day.

There are many guides out there that use plain SSH commands on \*nix systems, but let's help our Windows users and examine a tool that is readily available to anyone. We will use Putty to achieve our goals.

## There are three types of SSH Tunnels:

1 Local

2 Remote

3 Dynamic

This post will focus on Local.

## **Local Port Forwarding**

In this procedure, we will demonstrate two different protocols being tunneled to remote servers:

- (1) RDP to manage a protected server
- (2) HTTP to reach a protected website

This diagram illustrates the main components that are involved. To minimize the surface area that is subject to potential attacks, ONLY A SINGLE PORT is allowed through the firewall. By default, SSH uses port 22, but since it's well-known and subject to casual scans, it would add a level of complexity if you used another port. For simplicity sake, we'll use the default ports in this procedure.



#### Here's what we need to achieve:



The line is a good representation of the SSH tunnel – a port is opened on your local host, and whatever you send to that port goes to the SSH Server, which forwards it to the remote host, on whichever port you specify.

#### STEP 1 – Specify SSH server details in Putty

In your Putty configuration, configure your **Host Name** and **Port** of the SSH server. Enter a name in **Saved Sessions**, and **click Save**.

🕵 PuTTY Configuration	n		? ×		
Category:					
<mark>⊫</mark> ⊷ Session	^	Basic options for your PuTTY session			
		Specify the destination you want to conne	ct to		
En Ieminal		Host Name (or IP address)	Port		
Bell		sshcomputer.dnsname.or.ip	22		
Features ⊡ Window		Connection type: Raw O Telnet O Rlogin	I O Serial		
···· Appearance ···· Behaviour ···· Translation		Load, save or delete a stored session Saved Sessions			
Selection		SSH Tunnel Server	]		
Colours ⊡. Connection Data		Default Settings SSH Tunnel Server	Load Save		
···· Proxy ···· Telnet ···· Rlogin			Delete		
⊡ ·· SSH ··· Kex ··· Cipher □·· Auth ✓	•	Close window on exit: Always Never Only on clean exit			
< >					
About H	Help	Open	Cancel		

#### **STEP 2** – Configure a tunnel for RDP

In the **Tunnels** section (Connection – SSH – Tunnels), configure a specific Source Port to be used as the local end of your tunnel. We have elected to use 50001 because it's in the Ephemeral port range — i.e. a port that is not officially reserved for any other services. For more information on ephemeral ports, <u>click here</u>. In the **Destination** field, enter the IP address of the remote server, followed by a colon, then the destination port. This is an address that the SSH Server **must** be able to reach. Since we are behind the firewall, it is from its own local subnet.

Select **Local** to create a Local SSH Tunnel. Select **Add** to save your tunnel settings. Finally, return to the root (Sessions) to save your session again.

# Here is an example of managing the 192.168.1.11 server using port 3389.

Once the configuration setup is complete and the Putty session is up and running, open an RDP connection and enter your local IP (**127.0.0.1**), along with the Local port that you have specified in Putty — 50001 in this case.

🕵 PuTTY Configuration ?  $\times$ Category: Behaviour Options controlling SSH port forwarding ^ Translation Port forwarding Selection Local ports accept connections from other hosts Colours Remote ports do the same (SSH-2 only) - Connection Forwarded ports: Data Remove Proxy 50001 192.168.1.11:33 Telnet Rlogin . ⊡- SSH Add new forwarded port Kex 50001 Cipher Source port Add 🚊 Auth 192.168.1.11:3389 Destination ..... GSSAPI TTY Local Remote O Dynamic X11 Auto O IPv4 O IPv6 Tunnels Bugs More bugs Serial < 3 About Help Open Cancel



As demonstrated above, by entering **127.0.0.1:50001** in mstsc.exe, the connection is redirected to 192.168.1.11:3389



### **STEP 3** – Configure a tunnel for HTTP

We will make use of the fact that you may configure multiple tunnels simultaneously. This time we'll configure **Local port 50002 to redirect to 80** of the destination server (192.168.1.12 in this case). Just close your tunnel and load the session again in Putty.

🕵 PuTTY Configuration			? ×
Category:			
Behaviour Translation Selection Connection Data Proxy Telnet Rlogin SSH Kex Cipher Auth TTY X11 Tunnels Bugs More bugs Serial	Opt Port forward Local po Remote Forwarded p L50001 L50002 Add new for Source port Destination Local Auto	ions controlling SSH ing its accept connection ports do the same (S ports: 192.168.1.11:3389 192.168.1.12:80 warded port: 50002 192.168.1.12 Q Remote Q IPv4	port forwarding ons from other hosts ISH-2 only) Remove Add 2:80 O Dynamic O IPv6
About H	p	Ope	n Cancel

Select Add, and then save your session.

To launch it, open your browser and set Localhost: 50002 as the server name. It will reach the remote host transparently.

Devolutions Server - Lo: × +					×
$\leftarrow$ $\rightarrow$ $\circlearrowright$ http://localhost:50002/server	□ ☆	<u>-</u>	1	٩	
Devolutions					
<b>E</b> Devolutions					
Server					

Remember that both tunnels open as soon as your Putty session is running, enabling you to use them simultaneously.

In part 2, we'll have a look at **Dynamic SSH Tunnels.**