



## What is Reverse SSH Port Forwarding



---

**REVERSE SSH IS A TECHNIQUE  
THROUGH WHICH YOU CAN ACCESS  
SYSTEMS THAT ARE BEHIND A  
FIREWALL FROM THE OUTSIDE WORLD**

---

Reverse SSH Port Forwarding specifies that the given port on the remote server host is to be forwarded to the given host and port on the local side. To try to put this as simple as can be, Reverse SSH is a technique through which you can access systems that are behind a firewall from the outside world.

So instead of your machine doing a simple SSH, the server does an SSH and through the port forwarding makes sure that you can SSH back to the server machine.

## Why should you use Reverse SSH Port Forwarding

SSH is a very useful and powerful tool when comes time to access a remote machine or server securely. But now the problem occurs when you try to connect to a remote server that is behind a firewall and that firewall rejects any incoming connection or data transfer request that has no prior outgoing request.

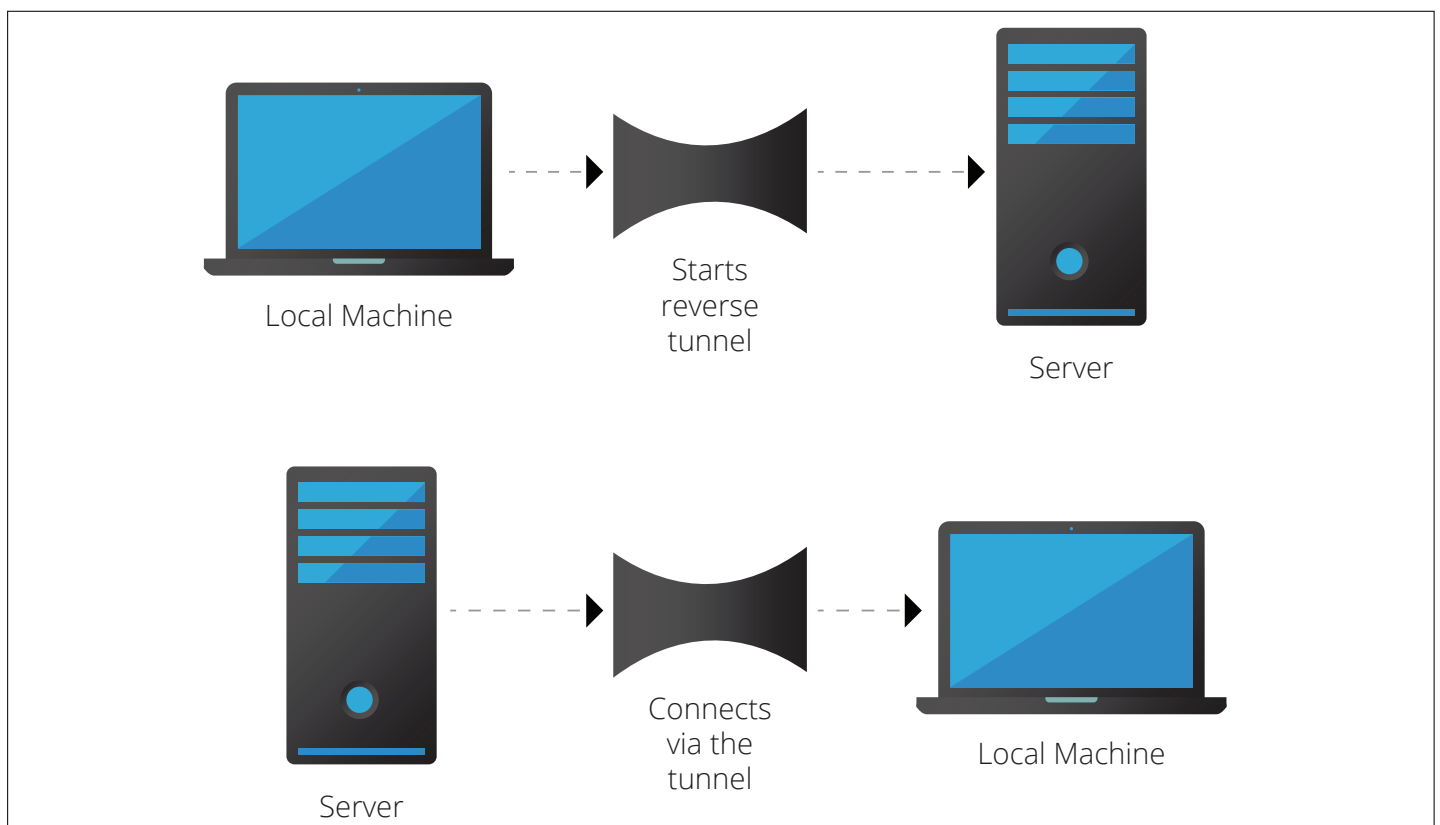
So let's say that you're working from home and need to access your office computer that is behind some very restrictive corporate firewall, well you wouldn't be able to connect because of the firewall policies. Usually this would be more than welcome since no outsiders should be allowed to access internal parts of a secure network. A secure VPN access would certainly solve your problem but what if you don't have access to that VPN? You know that the same firewall wouldn't have any issues with the connection coming straight from the server machine! Well this is when Reverse SSH Port Forwarding comes to the rescue! With reverse port forwarding you can

forward a port on the remote machine to the local machine while still initiating the tunnel from the local machine.

This works by assigning a socket to listen to the port on the remote side, and whenever a connection is made to this port, the connection is forwarded over the secure channel, and a connection is made to the host port from the local machine. Simply put, you want to connect your local machine to a server, so that you can use the tunnel to connect from the server to your local machine.

## How to Reverse SSH Port Forwarding

In order to SSH into a machine behind a firewall you will need to use Reverse SSH Port Forwarding. The machine in question needs to open an SSH connection to the outside world and include a -R tunnel whose entry point is the remote side (from server in our example) to connect to your machine, allocate a port there and make certain that any connection request on that port is then forwarded to the SSH port of the remote side (server).



**From the remote server side run the following command on the server:**

```
ssh -R 2210:localhost:22  
username@yourMachine.com
```

This command will initiate an ssh connection with reverse port forwarding option which will then open listening port 2210: who is going to be forwarded back to localhost's port :22 and all this will happen on the remote computer `username@yourMachine.com`.

The `-R` option tells the tunnel to answer on the remote side, which is the SSH server and the `-L` option tells the tunnel to answer on the local side of the tunnel, which is the host running your client.

You could also add some options to your command:

```
ssh -f -N -T -R 2210:localhost:22  
username@yourMachine.com
```

- **-f** : tells the SSH to background itself after it authenticates, saving you time by not having to run something on the remote server for the tunnel to remain alive.
- **-N** : if all you need is to create a tunnel without running any remote commands then include this option to save resources.
- **-T**: useful to disable pseudo-tty allocation, which is fitting if you are not trying to create an interactive shell.

**Now you need to do an SSH connection request from your machine to your own machine at port 2210 :**

```
ssh -p 2210 username@localhost
```

It may seem like you're doing an SSH on localhost but instead your request would be forwarded to the remote host. This command will establish a connection to the firewall host through the tunnel.

Hopefully this will help you to clear the fog surrounding Reverse SSH Port Forwarding. Have fun!

As always, please let us know your thoughts by using the comment feature of the blog. You can also visit our forums to get help and submit feature requests, you can find them [here](#).