## There's more...

Port forwarding can be made more useful using non-interactive mode or reverse port forwarding. Let's see these.

### Non-interactive port forward

If you want to just set the port forwarding instead of having a shell to be kept open for the port forwarding to be effective, use the following form of `ssh`:

```
ssh -fL 8000:www.kernel.org:80 user@localhost -N
```

The `-f` instructs `ssh` to fork to background just before executing the command, and `-l` for the login name for the remote host machine. `-N` tells `ssh` that there is no command to run; we only want to forward ports.

### Reverse port forwarding

Reverse port forwarding is one of the most powerful features of SSH. This is most useful in situations where you have a machine which isn't publicly accessible from the Internet, but you want others to be able to access a service on this machine. In this case, if you have SSH access to a remote machine which is publicly accessible on the Internet, you can set up a reverse port forward on that remote machine to the local machine which is running the service.

Reverse port forwarding is very similar to port forwarding:

```
ssh -R 8000:localhost:80 user@REMOTE_MACHINE
```

This will forward port 8000 on the remote machine to port 80 on the local machine. As always, don't forget to replace `REMOTE_MACHINE` with the hostname of the IP address of the remote machine.

Using this method, if you browse to `http://localhost` on the remote machine, you will actually connect to a web server running on port 8000 of the local machine.

# Mounting a remote drive at a local mount point

Having a local mount point to access the remote host filesystem is really helpful while carrying out both read and write data transfer operations. SSH is the common transfer protocol available in a network and hence, we can make use of it with `sshfs` which enables you to mount a remote filesystem to a local mount point. Let's see how to do it.

### Getting ready

`sshfs` doesn't come by default with GNU/Linux distributions. Install `sshfs` by using a package manager. `sshfs` is an extension to the FUSE filesystem package that allows supported OSs to mount a wide variety of data as if it were a local filesystem.

For more information on FUSE, visit its website at `http://fuse.sourceforge.net/`.

### How to do it...

In order to mount a filesystem location at a remote host to a local mount point, use:

```
# sshfs -o allow_other user@remotehost:/home/path /mnt/mountpoint
Password:
```

Issue the password when prompted, and data at `/home/path` on the remote host can be accessed via a local mount point `/mnt/mountpoint`.

In order to unmount after completing the work, use:

```
# umount /mnt/mountpoint
```

### See also

- The *Running commands on remote host with SSH* recipe, explains the `ssh` command

## Network traffic and port analysis

Network ports are essential parameters of network-based applications. Applications open ports on the host and communicate to a remote host through opened ports at the remote host. Having awareness of opened and closed ports is essential for security context. Malwares and root kits may be running on the system with custom ports and custom services that allow attackers to capture unauthorized access to data and resources. By getting the list of opened ports and services running on the ports, we can analyze and defend the system from being controlled by root kits and helps to remove them efficiently. The list of opened ports is not only helpful for malware detection, but is also useful for collecting information about opened ports on the system which enables us to debug network-based applications. It helps to analyze whether certain port connections and port listening functionalities are working fine. This recipe discusses various utilities for port analysis.