

## Creating arbitrary sockets

For defined operations such as file transfer, remote shells, and so on, we have prebuilt tools such as `ftp` and `ssh` respectively. However, in some cases you will find the need to do a customized network operation. An example for this might be writing a script which will do something when a remote client connects to your machine. In this recipe, we will create simple network sockets and use them for communication.

### Getting ready

To perform these things, we will need to create network sockets which enable us to do data transfer over a TCP/IP network. The easiest way to do this is by using the command `netcat` (or `nc`). We need two sockets: one listens for connections and the other connects to this one.

### How to do it...

1. Set up the listening socket using the following:

```
nc -l 1234
```

This will create a listening socket on port 1234 on the local machine.

2. Connect to the socket using the following:

```
nc HOST 1234
```

If you are running this on the same machine that the listening socket is, replace `HOST` with `localhost`, otherwise replace it with the IP address or hostname of the machine.

3. To actually send messages, type something and press *Enter* on the terminal where you performed step 2. The message will appear on the terminal where you performed step 1.

### There's more...

Network sockets can be used for more than just text communication; let's see how.

### Quickly copying files over the network

We can exploit `netcat` and shell redirection to easily copy files over the network:

1. On the receiver machine, run the following command:

```
nc -l 1234 > destination_filename
```

2. On the sender machine, run the following command:

```
nc HOST 1234 < source_filename
```

## Sharing an Internet connection

In the modern computing world, we are using different kinds of devices in our day-to-day life. We use desktop computers, netbooks, laptop computers, tablets, smartphones and so on. Additionally, we feel the need to keep all of these connected to the Internet for which people usually keep a wireless router which distributes an Internet connection across these devices. But what if you don't have a router (or your router just blew up) and you need to share the Internet? No problem! Linux, iptables and some scripting are at your rescue.

### Getting ready

This recipe uses `iptables` for setting up **Network Address Translation (NAT)** which lets a networking device share a connection with other devices. You will need the name of your wireless interface for which just use the `iwconfig` command.

### How to do it...

1. Connect to the Internet. In this recipe, we are assuming that the primary wired network connection, `eth0` is connected to Internet. Change it according to your setup.
2. Using your distro's network management tool, create a new ad hoc wireless connection with the following settings:
  - IP address: 10.99.66.55
  - Subnet mask: 255.255.0.0 (16)

3. Use the following shell script to share the internet connection:

```
#!/bin/bash
#filename: netsharing.sh
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -i $1 -o $2 -s 10.99.0.0/16 -m conntrack
--ctstate NEW -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT
iptables -A POSTROUTING -t nat -j MASQUERADE
```

4. Run the script as follows:

```
./netsharing.sh eth0 wlan0
```

Where `eth0` is the interface which is connected to the Internet and `wlan0` is the wireless interface which is supposed to share the Internet to other devices.