Port Forwarding configuration for your home router

Mon 14 January 2019
By hbsc & friends

Introduction

The whole premise of the homebrewserver.club is the simple — yet often overlooked — fact that your home internet subscription theoretically also allows you to host services. The internet is in its essence a bi-directional medium. Anyone with an internet connection can not only look up on-line content but also host it!

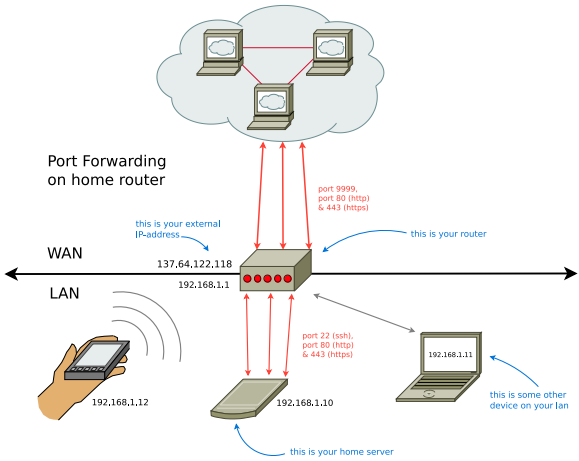
In times of ‘cloud providers’ and ‘virtual private servers’ it is an easy thing to forget, and internet service providers don’t make it easy on you either, but a homebrew server can be as simple as an old laptop connected directly to your home router. However, you do need to change some settings on the router to make that happen!

Requirements

To begin serving from home you need the following:

- Make sure you have physical access to your home router.
- Get to know the password of the admin user (this is usually provided in the box or written on the label on the underside of the router).
- Have an available power socket next to your router.
- Have a homeserver with a web server and open SSH server running on it.
- An ethernet cable to connect your server to the router.

Port forwarding theory



By default home routers have configured the firewall so that the devices behind your router are inaccessible to the internet. This is to prevent your private network from being public.

Machines behind your router (called your local area network or `LAN`) can make connections to the wider internet (known as `WAN`), but not the other way around.

However, when hosting a server at home, we do want that server to be reachable from the internet. In order to do that we need to open so-called *network ports*.

Ports are logical 'gates' that are open or closed to connections. These ports have numbers and are *standardized* (https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#Well-known_ports) for specific protocols or applications. For example, HTTP traffic from a website would default to port `80`. HTTPS defaults to `443` and SSH defaults to port `22`.

To make our server accessible over the internet we need to open the ports on the router and forward them to our server. This is called port-forwarding.

The exact method (and terminology) of port-forwarding differs from router to router. However, it always follows a similar scheme where you designate inbound traffic on a certain port to be forwarded to the your server's IP-address and port on the local area network.

For this you need to have access to the administrative panel of your router.

Find your router

To access the administrative panel of your router you need to find it's IP-address. You can do this by connecting to that router via Ethernet or Wi-Fi and then finding out what your own IP-address is.

On Debian based systems this is done like this in the terminal:

```
$ ifconfig
```

If you get a command not found warning try this:

```
$ ip address
```

This will return information on your network connection. Look for the line saying `inet`

```
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
  1500 qdisc mq state UP group default qlen 1000
  link/ether ac:ab:00:00:ac:ab brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.11/24 brd 192.168.1.255 scope global
  wlp3s0
     valid_lft forever preferred_lft forever
  inet6 fe80::eab1:fcff:acab:374e/64 scope link
     valid_lft forever preferred_lft forever
```

In this case the IP-address is `192.168.1.11` as a rule of thumb you can then change the last digit to either `1` or `254` to find the router.

Log in to your home router and get to know your LAN

Using a web browser navigate to the IP-address you found above to reveal the router's admin panel. It should provide you with a log in field where you can enter the router's admin details to get access to the control panel.

There you will see a lot of possible settings. Look at the options "LAN", "DHCP Leases" or "Network" to get an overview of all the devices.

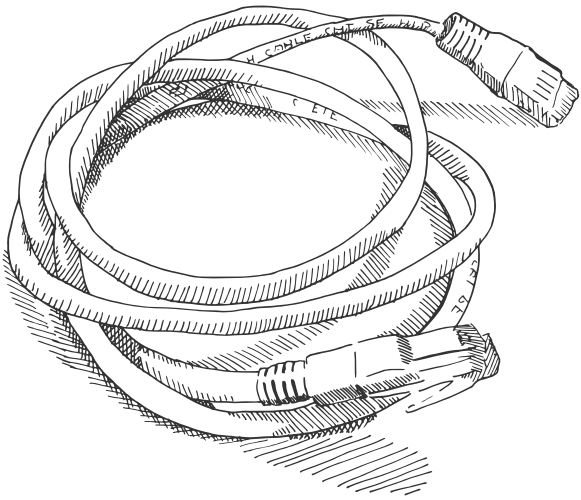
Connect your home server

Use an ethernet cable to connect your home server to your router. In case that it has ethernet ports in different colors/markings make sure you take something that says either `LAN` or `INET` .

Have a look at your router's interface again and look for the IP-address that your server was assigned. In this guide I'll assume it was 192.168.1.10.

Next try to find an option called "Static (DHCP) Lease" or "DHCP Binding" or something similar in your LAN view. Then make sure to assign your server a static DHCP

lease. This will make sure that the server is always reachable under the same IP-addres.



Forward the ports

Once you've set up a static lease to your home server you can start port forwarding. Depending on the make of the router it can be also be called *Port Sharing* or *Traffic Forwarding*. Again, depending on the make of the router, it can be found under the tabs 'security', 'access control' or 'internet'.[1] (#fn:1)

The basic process is to determine which external port to open and to which IP address on the LAN and which port to forward it to.

You might be asked a few things, including the name of the rule, the protocol (TCP, UDP or both), the external port and the internal port. Sometimes you are given the option to open a range of ports.

To open the ports for the web server, we're opening two separate ports, one for plain HTTP and one for secure HTTP.

Open the external port `80` for plain HTTP and redirect it to the local IP-address of the homeserver:

```
Name: "HTTP Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 80
  Port to device: 80
```

Open the external port `443` for HTTPS and redirect it to the local IP-address of the homeserver:

```
Name: "HTTPS Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 443
  Port to device: 443
```

Lastly we will open a port for `SSH`. The change here is that we open the external port `9999` and map that to `22` internally.

Setting SSH on a non-standard port is a low-level way to prevent automated scripts gaining access to your homeserver. TODO add link to basic security

```
Name: "SSH Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 9999
  Port to device: 22
```

Concluding

Now that you have opened the corresponding ports you should be able to type your external IP-address in your browser and should be automatically redirected to the website on your home server.

How to find out which ports to open?

While a majority of applications will work on `80` and `443` you might need to open different port for different applications. For example in the series describing *self-hosted chat over XMPP* (https://homebrewserver.club/configuring-a-modern-xmpp-server.html#set-up-firewall-and-dns) ports `5000`, `5222`, `5269` and `5281` are opened and forwarded.

Most installation guides for software will tell you whether you need to open ports. However it is also possible to see what applications are listening to what port using:

```
$ netstat -tulp .
```

Notes:

1. *https://portforward.com/* (https://portforward.com/router.htm) has a large list of routers and visual instructions on how to set up port forwarding on them. ↩ (#fnref:1)