Setting up a Network

Mattias Holm

2009-01-23

Abstract

In this lab assignment, you are going to learn how to setup a working network, complete with DNS, SMTP, IMAP, HTTP and XMPP servers. Having an understanding for these systems will deepen your understanding of how the Internet works.

1 Assignment

The lab consist of a number of tasks which individually are described in the following sections. You will either complete all the assignments perfectly, as stated in the descriptions or you will fail on this lab.

For the lab you will work in groups of two students.

The purpose of the assignments is to configure a server and a client to connect to the Internet. The network we are going to build is going to look like Figure 1. The operating system used is a Ubuntu Linux based server system (see `http://www.ubuntu.com/` for more info).

Your server has an IP addresses assigned. You have been assigned a segment of addresses to use for the client / server network. The IP you use is assigned as detailed in Figure 1. The network address is thus 132.229.136.$\{(N-1)*8\}$/29, where N is the group number. The subnet mask for these networks is 255.255.255.248. The server also need an IP address in the same segment that the gateway is in. The gateway is located in the network 132.229.136.240/28. This mean that the two IPs that need to be set for the two different network cards must have different netmasks.

You will be able to log in to the machine initially with the username *netlab* and password *netlab*. In order to prevent that others use your machine, you should change the password for this user as soon as possible. This account have root-access to the machine, getting a root shell can be done by typing `sudo -i` in the terminal. We have reserved a special account for our administrative access, so if you would forget your password, it is still possible for you to have it reset.

I would like to point out that it is very important to be prepared for the labs, you only have 4 weeks in order to finish the assignments here, so be prepared.

Roughly, in order to be on good track you should have finished the routing part during the first week, DNS should be working after two weeks, then e-mail and after the 4th week everything should work. If you manage to finish early, all the better. But, if you do not manage to complete the tasks within the time as described in this paragraph, it is very important that you notify the lab assistants (Mattias Holm or Harmen van der Spek) as soon as possible.

2 Routing

BEFORE PROCEEDING WITH THIS EXERCISE, READ THE FOLLOWING: *The Ubuntu operating system that is used by the server machines in this exercise come with a daemon named network-manager. The daemon is there for helping users of client machines automatically configuring the network system, however, it will interfere with the tasks that you are about to do. This software package should be removed before proceeding. You can remove software with the synaptic or apt programs. After this you need to add the networking devices in /etc/network/interfaces. For all client machines except machine A, the devices are named eth1 and eth2, where eth2 is the secondary network card that you should use to connect your client machine, the machines are not assigned to any specific group, but will be when you start to use to them.*

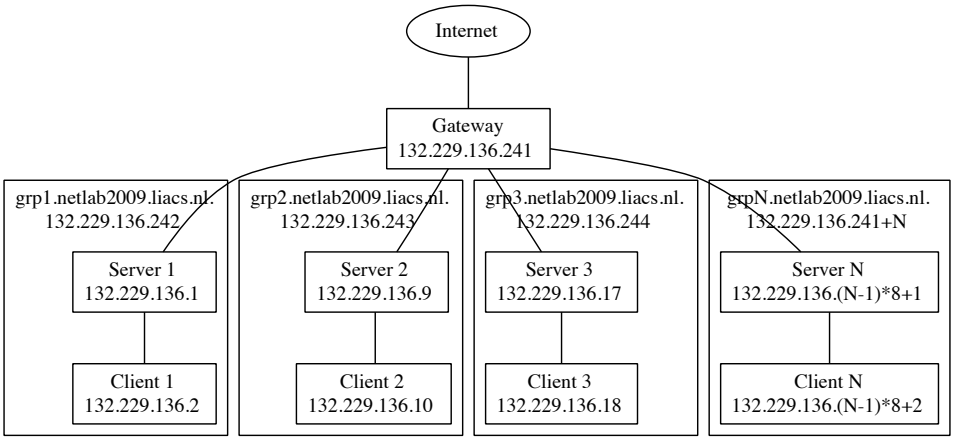

Figure 1: Network Topology

Your first task is to setup your server machine as a router. The server is supposed to communicate with the central gateway and have secondary routing paths to some of the other groups' servers in order to support fault tolerance in the network. For large scale routing systems, software such as OpenBGPD or OSPFD are typically used. However, this is overkill if we want to understand how routing works. Therefore, you will have the task to setup the Linux kernels built in routing system.

Routing can be done in the linux kernel, but if more sophisticated routing is needed one need to use a routing daemon.

The following command can be used in order to show the kernel routing tables:

```
netstat -nr
```

The following command can be used in order to add static routes to the kernel:

```
route add
```

Note that those routes will disappear when the networking subsystem of Linux is restarted. But the route add command does serve as a nice starting point for the lab. When you are happy with the routing tables you should add them as permanent routes that are configured during start-up.

Check the man pages for `netstat` and `route` for more information. Note that you will not have internet access before configuring the routing and DNS tables properly.

Your first assignment is thus to configure the routing tables of Linux on the server to correspond to the network in Figure 1. This means that your tables should route packets going to the other groups networks without going past *132.229.136.241*.

When everything is working properly, `traceroute` should work between the different groups by tracing a path with two hops. Similar to the output detailed on the following lines:

```
traceroute to 132.229.136.17 (132.229.136.17), 64 hops max, 40 byte packets
 1  132.229.136.243 (132.229.136.243)  0.658 ms  0.250 ms  0.300 ms
 2  132.229.136.17 (132.229.136.17) 0.658 ms  0.250 ms  0.300 ms
```

Note that it is not OK, if the traceroute pass through the main gateway.

3 Domain Name System

Your second assignment is to configure the DNS server `bind` to give out DNS names for your subdomain. Your subdomain is *grpN.netlab2009.liacs.nl*, you are expected to create a number of hosts on this

subdomain. These include www, smtp, imap and xmpp that should all point out the same machine (i.e. the servers internal IP, do not point these at the external IP).

You should configure the DNS-forwarders properly in your named options file. In this lab, you should add all the other groups' DNS-server IPs so that a request for *grpN.netlab2009.liacs.nl* will be handled by the proper DNS-server. You should do that and also add *132.229.44.11* as an additional forwarding server (that server will forward requests outside the netlab2009-domain). In principle, the parent server should contain pointers downward to all the groups' servers, but in order to avoid having to modify the LIACS domain to much, this approach has been taken.

For more information you can see the bind documentation at `http://www.bind.org/` and the HOWTO located at `http://langfeldt.net/DNS-HOWTO/BIND-9/`

3.1 Important Considerations

The DNS system typically caches requests to DNS entries, these are cached in multiple directions. I.e. both sub-domains and super-domains get their translation records cached. When you configure your domain server, you need to make sure that the TTL limit is set to a reasonably low value. The default values in bind that comes with Ubuntu Linux is about 1 week. If you leave this at the default values, the *netlab2009.liacs.nl* DNS-server will cache your entries and when you make mistakes in the DNS configuration file, you will have to come back next week as the DNS entries have been cached with a TTL value of one week. You should thus, before you turn on the DNS server on your server machine, make sure that the TTL is lowered to something more manageable like around 10 seconds or so. Note that you should change all of the $TTL, *Refresh*, *Retry*, *Expire* and *Negative Cache TTL* variables.

4 Connecting the Client

In many networks, a server will be distributing IP-numbers to clients that connect to the network through DHCP. You should install the DHCP server on the server machine and configure it to use the *ethN* device for listening more information on the DHCP3 server can be found at `https://help.ubuntu.com/8.10/serverguide/C/dhcp.html`. When the server is properly set up, the client should when it is started up obtain an IP within your assigned IP range and be able to connect to the main network.

5 E-Mail

The next assignment is to get e-mail up running. You should make sure that the SMTP server *postfix* is up running and that clients on your subnet can send e-mail through it. You should also install the IMAP server *cyrus* and configure it so that you can fetch e-mail from the client machine.

5.1 Security Considerations

An SMTP server must not forward e-mail from unknown hosts. Open relay servers are usually found and exploited by spammers. Therefore, in order to pass the lab-assignment your SMTP server must not forward e-mail from anyone else but your own subnet. This will be tested.

6 Web Server

You should get the Apache webserver up running and replace the default page with something more personal. The servers should be accessed at `http://www.groupN.netlab2009.liacs.nl` from the client machines.

7 XMPP Server

The final task here is to setup an XMPP server. XMPP is the IETF standard for instant messaging. Several providers offers IM through XMPP. XMPP can unlike services such as MSN, Yahoo, ICQ and AIM work in federated mode. This means in plaintext that it works roughly as e-mail works (i.e. that

each domain is responsible for its own server). In order to federate an entry in the DNS server must be made that points out the XMPP server for your domain (note that these are not normal *IN A* records).

At present there is only one major service for XMPP based chat (Google Talk), but several smaller service providers exist as well. Look at `http://www.jabber.org/` and `http://www.ejabberd.im/node/661` for more information.

When this is done, you should be able to connect an XMPP client such as PSI or Pidgin to your server and add and chat with other users that are located on the other groups XMPP servers.

8 Links

SASL information: `http://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer`
Cyrus IMAP server: `https://help.ubuntu.com/community/Cyrus`
Postfix on Ubuntu: `https://help.ubuntu.com/8.10/serverguide/C/postfix.html`
Postfix server: `http://www.postfix.org/docs.html`

Appendices

A General Info

This section contain information only, you are not required to add load balancing or IPv6 support in this lab.

A.0.1 Domain Names in DNS

The DNS system works in a hierarchy, the root-servers are known as "." and these servers keep track of all the *top-level domains* on the Internet such as *.eu*, *.nl*, *.se* and *.com*. When you configure *bind* you can work with both full domain names and local sub-domain names. In order to differ between these two kinds, you always add the final "." to the fully qualified names. For example:

```
@       IN      MX      10      smtp.netlab2009.liacs.nl
```

refers to *smtp.netlab2009.liacs.nl.netlab2009.liacs.nl* and

```
@       IN      MX      10      smtp.netlab2009.liacs.nl.
```

refers to *smtp.netlab2009.liacs.nl*.

A.0.2 Load Balancing with DNS

DNS supports aliased names that can map to many real hosts. This is done with the CNAME record type:

```
www0 IN A 192.168.0.1
www1 IN A 192.168.0.2
www IN CNAME www0.mydomain.com.
    IN CNAME www1.mydomain.com.
```

Most DNS-servers then serve the IPs assigned to the CNAME record with round robin method.

A.0.3 Important Considerations for the Future

Since IPv4 is in the process of running out of addresses, this means that DNS servers need to be updated with IPv6 addresses. At present very few ISPs (and in in principle none in the Netherlands) support IPv6 addressing.

This is however expected to change in time. When configuring a DNS server to supply IPv6 addresses you use the *AAAA*-records instead of *A*-records that are used for IPv4 address records. In this lab you

do not need to configure IPv6 support, but you should be aware of that this will be necessary in the near future.

Another item of interest for the future is the introduction of DNSSEC. DNS is a critical system for the Internet, but it is inherently insecure. DNSSEC will bring in signed domain entries, signed hierarchically. At the moment this is not done for the root servers, but several TLDs have started with DNSSEC locally, these include the *.org* domain and the Swedish and Bulgarian TLDs.