ComS 252 Homework 11: Build a Router

Extra credit (points divided by group size)

Due December 5, 2023

1 Objectives

For this assignment, you will build a NAT/Router on a Linux (virtual) machine with two interfaces. The following HOWTOs may be helpful for this assignment, as supplements to the lecture material.

- DHCP: http://tldp.org/HOWTO/DHCP/x369.html
- DNS: http://tldp.org/HOWTO/DNS-HOWTO.html

2 Build the virtual machines

1. Download the ISO file to initialize the virtual machines for homework.
2. In VirtualBox, create **two** new virtual machines for this assignment. The default disk size (a few GB) should be sufficient. You will need to run both machines at the same time, so adjust the memory settings accordingly.
3. For each VM, set the ISO file as the optical disk, and boot up the VM.
4. Select "Build Hw11c virtual machine" or "Build Hw11s virtual machine"
5. You are encouraged to take a snapshot of each VM after installation completes.
6. At first boot, each VM initializes itself by fetching and running a script from the server. The script will, among other things, create a user account with your ISU username. All user accounts will initially have passwords that are the account name, followed by "`pw`".
7. When each VM shuts down after initialization, you are again encouraged to take a snapshot.

3 Set up network interfaces

3.1 Using VirtualBox (probably required)

You will set up a network between the two VMs, using the topology shown in Figure 1. This Internal Network should use subnet `172.27.11.0/24`. If this causes a conflict on your system (e.g., your server VM is using this subnet already for Adapter 1), you may choose a different private subnetwork but keep the same host numbers. In other words, you may change the first 3 numbers of the IP addresses if necessary.

Note that the client VM, `Hw11c`, will have access to the Internet only through the server VM, `HW11s`, and only after configuring the server as a home router using NAT.

1. Under the settings for the *server* VM, under "Network", you should already have Adapter 1 enabled and attached to a "NAT" network. Now, enable Adapter 2, attach it to an "Internal Network".
2. On the server VM, use `nmcli` to give Adapter 2 a private static IP address on the Internal Network, something with a host number below 20 (e.g., `172.27.11.1` or `172.27.11.11`). This will be referred to as the *server address*. Reboot the server VM and make sure both adapters obtain IP addresses, and Adapter 2 obtains the desired static IP address.

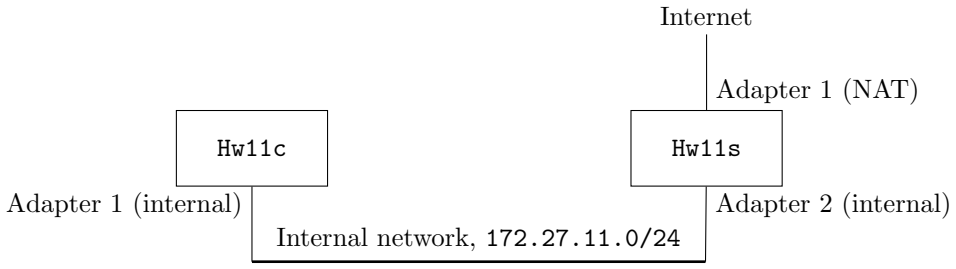

Figure 1: Network topology

3. Under the settings for the *client* VM, under "Network", you should already have Adapter 1 enabled and attached to a "NAT" network. Change this instead to an "Internal Network". The name of the internal network should be the same as the server VM. Once this is done, **no more configuration is needed on the client**.

3.2 Using UTM (probably impossible)

It is highly non-trivial, if not impossible, to set up network interfaces as required in Figure 1 using UTM. To do so would require the following.

1. Build a second network interface on `Hw11s`. This, I think, is possible.
2. Make sure `Hw11s` and `Hw11c` can communicate on some virtual network.
3. Be sure that this virtual network does NOT have DHCP running. To date, I have **not** figured out how to do this in UTM. Unfortunately this is required; otherwise you will be setting up a network with two competing DHCP servers on it, which is a recipe for disaster.
4. On the server VM, use `nmcli` to give Adapter 2 a private static IP address on the Internal Network, something with a host number below 20 (e.g., `172.27.11.1` or `172.27.11.11`). This will be referred to as the *server address*. Reboot the server VM and make sure both adapters obtain IP addresses, and Adapter 2 obtains the desired static IP address.

4 Set up the server for DHCP

You will configure the server VM as a DHCP server for the internal network, so that it gives out addresses on the Internal network.

1. Configure the DHCP server to start at boot time.
2. Configure the DHCP server to give out IP addresses in the range `172.27.11.100` to `172.27.11.199` for a generic client.
3. The client VM should *always* obtain an IP address of `172.27.11.42`, based on the MAC address of the client's network adapter.
4. All clients should receive the *server address* as the router (gateway) IP address.
5. All clients should receive, *for now*, DNS server IP addresses of `8.8.8.8` and `8.8.4.4`. You will change this later, after you set up the server VM as a DNS server.

Once the server is set up, you can test it by starting the client VM and checking the client's obtained IP address, and by running `resolvectl` on the client. When it works, you should be able to `ping` the server IP address from the client machine. You will not be able to `ping` any other addresses from the client, yet.

On the server, verify that you can ping the client's IP address, an Internet IP address (such as `8.8.8.8`), and an Internet FQDN (such as `google.com`).

Note: Complex file structure (for example, nested braces) in your `dhcpd.conf` configuration file will confuse the Turnin script. To ensure that your work can be automatically graded, you should have most options as "global" (not within any braces) and use braces only when you must. The script will warn you if your configuration file is too complex.

5 Set up the server as a NAT gateway router

To configure the server as a router, do the following.

- Make sure `firewalld` is running.
- Configure the firewall so that Adapter 2 belongs to zone `internal`, Adapter 1 belongs to zone `external`.
- Turn on masquerading for the `external` zone. This should turn on packet forwarding.
- Set the firewall to allow *all* traffic to proceed from zone `internal` to zone `external` as follows.

  1. Create a firewall policy named `username-out`, where `username` is replaced with the primary username on the server VM.
  2. Set the ingress zone of the policy to `internal`
  3. Set the egress zone of the policy to `external`.
  4. Set the target of the policy to `ACCEPT`.
  5. Check that the policy is set up correctly using `firewall-cmd --list-all --policy=username-out`.

  See https://firewalld.org/2020/09/policy-objects-introduction, and/or the man page for `firewalld.policies`, for more information.

- Set the firewall to allow *all* traffic to proceed from zone `external` to zone `internal`. Use steps similar to those listed above, but this time use a policy named `username-in`.

Make sure that all these changes persist across reboots of the server VM (i.e., are `--permanent`).

The easiest way to test if this works is to (re)start the server and then try to ping `8.8.8.8`, and then `google.com`, from the client machine. If `ping 8.8.8.8` shows "Packet Filtered" for the packets, then likely your `username-out` policy is not set up correctly. If `ping 8.8.8.8` works correctly but `ping google.com` does not, then likely your `username-in` policy is not set up correctly. If both work, then your router is functioning properly. You should also try to view a webpage on the client machine using `lynx`.

6 Set up a forwarding DNS server

6.1 Changing DNS servers used by the client

Re-configure DHCP on the server VM to use the server's IP address as the (only) domain name server. To test, reboot the client and run `resolvectl`, making sure the server VM is listed as the only DNS server.

6.2 Configuring DNS

- Make sure service `named` is running. The utilities `named-checkconf` and `named-checkzone` are useful to check for errors in `named` configuration files.
- Edit the configuration file `/etc/named.conf` to set up a recursive (forwarding) nameserver (see section 4 of the DNS–HOWTO). Use `8.8.8.8` and `8.8.4.4` as the "forwarders".
- Edit the configuration file `/etc/named.conf` so that the nameserver responds to queries from any host on the `172.27.11.0/24` subnet; add this subnet to the `listen-on port` and `allow-query` lines in `/etc/named.conf`.

- Edit the configuration file `/etc/named.conf` to disable `dnssec-validation`.
- Allow service `dns` through the firewall, for zone `internal`.

Make sure these changes persist across reboots of the server VM. To test, reboot the server and client. Then, on the client, ping `www.google.com` (or any other FQDN) on the client. If this works, then your recursive DNS server is working.

7 Set up an authoritative DNS server

Define your own domain, `username.cs252`, where the IP addresses are in the `172.27.11.0/24` subnet and `username` is replaced with the primary username on the VM. Machines in your domain will be named `<hostname>.username.cs252`. Your server should know about the following machines:

- `hw11s.username.cs252`, with appropriate IP address.
- `hw11c.username.cs252`, with appropriate IP address.
- `printer.username.cs252`, with IP address `172.27.11.200`.

Configure your DNS server as the master server for your domain. Be sure to include inverse queries, so you can convert from an IP address to a hostname. If you can ping `hw11s.username.cs252` and `hw11c.username.cs252` on the client, then your authoritative DNS server is working. Use `dig` to test the inverse queries on the client. You can use ping `printer.username.cs252` to make sure the IP address is correct, but of course no packets will get to this non-existant machine.

8 Finishing touches

Configure the server as follows.

- Edit the DHCP server configuration so that the line

  ```
  search username.cs252.
  ```

  automatically appears in the client's `/etc/resolv.conf` file when the client obtains its IP address. When this works, you will be able to drop the domain name and simply use `ping hw11s`, `ping hw11c`, and `ping printer` on the client.
- When the server VM boots up, the firewall should allow *only* the following services through: for zone `external`, service `ssh`; for zone `internal`, services `ssh` and `dns`. No other services or ports should be allowed.

9 Submitting your work

There is no need to submit anything on the client VM. On the server VM, from your user account, run "`sudo Turnin`" to submit your work.

As usual, submission requires Internet access (and VPN access, from off campus), as this will collect and upload your work to the homework server. Feedback on your submission is collected in a text file, that you can view later using "`cat submit.log`" or "`less submit.log`".

To shutdown the VMs cleanly, run "`poweroff`".