CIT 371 lab 21: DNS and TCP/IP

This lab can be done using SSH/PuTTY or the Web Console. See the Student VM Access document for more information on accessing your VMs.

Read chapter 12 before doing this lab. su to root for this entire lab except as noted in 2.

1. When do you need to use the DNS? If you need to convert an IP alias into an IP address. Type **nslookup www.nku.edu** and **nslookup www.uc.edu** and make a note both IP addresses. ping both computers using the hostnames (the aliases). ping requires the destination’s IP address so your computer must use the domain name system (DNS) to translate the hostname to address. What happens if you can’t access a DNS server? Type **mv /etc/resolv.conf /etc/resolv.conf2**. resolv.conf stores the IP addresses of your local DNS name server(s). Without that file, your computer cannot start the DNS request process. Re-ping the two computers. *What happens?*Now ping them using the IP addresses you obtained earlier. You can still access the Internet if you know IP addresses. Using DNS is built into our operating systems and often very fast, but usage requires communicating with other computers. A shortcut to bypass DNS is to hardcode IP alias to IP address translations in the **hosts** file (/etc/hosts in Linux). This file already contains a couple of entries that define your loopback address (named localhost.localdomain and variations). **Edit** this file in **vi** and add the following line where *IPaddress* is the address of www.uc.edu obtained above. Make this the first entry in the file. **Save** the file and **exit** vi.

***IPaddress* www.uc.edu**

Re-ping both computers using their aliases. *Why can you ping UC but not NKU?* We can place any alias in our hosts file. Edit /etc/hosts and change **www.uc.edu** to **uc** (leave the IP address as is in the line). Save and exit vi. **ping www.uc.edu** again and then try **ping** **uc**. *What happens in both cases? Explain the role of the /etc/hosts file.* Rename /etc/resolv.conf2 back to /etc/resolv.conf and edit your /etc/hosts file to remove the uc entry.

**When they re-ping the two they are unknown. We can ping UC but not NKU because the UC address is unknown to the computer since we had to declare it. The ping with UC works but not with** [**www.uc.edu**](http://www.uc.edu)**. This is because we have saved the IP address to UC now instead of www.uc.edu. The /etc/hosts is made to hold different IP address files that are to be known by a different name.**

1. Let’s look at some Linux programs to explore TCP/IP. TCP/IP uses different types of data packets for Internet communication (primarily TCP and UDP). UDP packets are smaller and take less time and bandwidth to transmit because they do not contain data that makes delivery reliable. We use UDP for streaming, DNS (usually) and DHCP while the more reliable TCP is used for important data (email, web pages, etc). Aside from IP addresses, Internet communication uses source and destination *ports*, numbered 0 to 65535. Most types of applications have *well-defined port numbers*, such as 80 for HTTP requests and 22 for ssh.
   1. Open a second terminal window as yourself (not root). **ss** displays active domain sockets, including those being used by TCP/IP communication. As yourself, type **ssh zappaf@10.2.3.17**, when asked about the key fingerprint, answer **yes**, log in as zappaf (password is **gail**). This opens an connection to another VM. Type **who** to see who is logged in. Your entry will be listed as zappaf with your VM’s IP address in parentheses (other people might be logged in as zappaf). In your first terminal window, as root, type **ss**. This shows all domain sockets in use. Type **ss –t** to show only sockets pertaining TCP communication – there should only be one entry, your ssh command. *What Port is listed under Netid?* This is the type of datagram that ssh uses. *What is the state of the communication?* Also listed are Recv-Q and Send-Q, which should be 0, the local address/port and the peer (destination) address/port. These are both indicated as IPaddress:port where the source (local) port is a number while the destination (peer) port is labeled as ssh. *What local port is being used?* Usually your operating system selects the next available port for your source port. Let’s test this. **Exit** out of ssh in your second terminal window and repeat the ssh command to log back in. In the first terminal window, redo **ss -t**. *What port is used now?*Type **ss -4**. This shows all IPv4 communications, which will include TCP and UDP messages. *How many entries are there? Are they TCP, UDP or a combination? What software is responsible (this will be indicated as the peer port).*NOTE: if nothing is listed, just answer that way.
      1. **41724 is listed under Netid. The state of communication is ESTAB. The local port being used is 41724. The port used now is ssh. There are 4 entries. They are all TCP. The software responsible is port number 41726.**
   2. Type **ss –l** (lower case “L”) to display all ports currently (or previously) listened too. *What programs are currently listening (these are indicated as the port after local address)?* As they are not currently connected, there is no peer address. *What packets are they listening for, TCP, UDP or a combination?*
      1. **The current program listed is 45499, 49279, 48438, 45640. They are all listening TCP.**
   3. netcat (or nc), allows us to send network messages from the command line, opening a connection between two computers where we specify the destination port number to use. Type **nc –l 301** (lower case “L”) in the first terminal window (where you are root, 301 is the port number, an unassigned port that I picked arbitrarily). This command tells Linux to listen over port 301. Since nc is running, you won’t get your prompt back. In the other window, type **nc *ipaddress* 301** where *ipaddress* is your computer’s IP address. Again, you do not get your prompt back. Both windows have placed you into buffers where, whatever you type, will be sent to the other window. In either window, type anything followed by <enter>. *Was the message echoed in the other window?* In the second window (where you are not root), type **control+c**. *What happens in both windows?* **Open** a third window. Repeat both nc commands in your first two windows and in the third window, type **ss –4**. You will see the two ports being used. In one window (the one where you are root), the port is 301 as you have specified. The OS picks a random source port to use to receive communication back. *What port is being used?* **control+c** out of nc and close both your second and third windows.
      1. **Yes, the message we typed is in the other window. Both windows close running operations and control c is the command. The port being used is 58866.**
   4. A command similar to ss is netstat, which is being deprecated in favor of ss, but you can still use it. Type **netstat**. You will see for every port that is or has been used what its status is. In the last column, you will see the software that is communicating over that port. Type **netstat | grep CONNECTED | wc –l** (lower case “L”). This will tell you how many ports have the status CONNECTED. *How many?* Most connections are indicated as STREAM, but a few are DGRAM (datagram). Issue a command to list just those that are DGRAM. *What command did you enter?* Most of these have large port numbers but a few have ports under 1024. *Which processes specifically are using these lower numbered ports (there are only a few, maybe 4)?*
      1. **There are 0 ports connected. The command I entered was Netstat | grep DGRAM | wc -l . The lowered number ports is 4.**
   5. Type **ifstat** to view network statistics for your interfaces. This displays information about errors and dropped packets. *Do you have any dropped packets? How many errors have been recorded for ens33?* Type **nstat**; this instruction provides statistics on total network usage (number of bytes and packets received and sent of various types). *How many ICMP input and output errors have occurred? How many UDP datagrams have been sent or received (input or received is “in” and output or sent is “out”)?*
      1. **No, there are no dropped packets. There were 84147 errors recorded from ens33. There were 24 ICMP input and output errors. There were 3 UDP datagrams that have been sent or received.**
   6. mtr tests your network and reports back problems. It uses ping and traceroute and counts the number of packets lost. It uses a low TTL for the packets. *What does TTL stand for?* 
      1. **TTL stands for Time to Live.**

Shut down your VM, exit out of VMware and submit your lab report.