EE487: Applications of Cyber Engineering Name(s):
SX-10: IP Routing & TCP Connections

**Discussion**: We have seen now that our network communications have a physical layer (in class we are using wi-fi which we will learn about in the next module) necessary to actually transmit bits of data, a data-link layer (Ethernet in our case) necessary to create point-to-point links between individual computers, and a network layer (IP in our case) necessary to enable addressing and routing of data packets across the internet. Ethernet is only capable of sending data as individual frames from one computer to another, but the frames are not globally routable. IP adds the capability of routing packets anywhere in the world because IP's address scheme is hierarchal in nature (like a fully formed US Mail address). The problem however, with all of these methods so far is that none of them are reliable. They check only certain types of errors, and when they find one, they simply discard the flawed data. They are also only aware of one packet or frame at a time. They have no concept of groups of packets – every single one is addressed and routed individually.

TCP solves this problem by setting up and maintaining a connection from one end point to another end point in a conversation. Thus, we say that it is the first "end-to-end" layer in the TCP/IP stack. What are the end points? Let's say you go to http://www.navy.mil. Your computer is one end point and the other is the navy.mil web server. There are going to be many, many links and routers in between you and the web server (in fact the packets can take many different routes and arrive out of sequence). TCP only functions at the two end points, whereas the data-link and IP are used all along the path.

Today we will explore more details on how IP routes packets and how TCP cleans up any messes that IP leaves behind. IP is an unreliable protocol, otherwise known as "best-effort." It will do its best to route packets, but if it has any trouble with one of them, it simply discards it capriciously, and then moves on to serving other packets. This prevents the Internet from becoming congested with problem packets. TCP has the tougher job of keeping track of all of those packets and recovering from the inevitable errors that occur in transmission. Let's say that 100 packets are sent and packet # 92 is lost by IP somewhere along the way. We have no way of knowing where, but TCP on the receiving end figures out when it takes muster that #92 is missing and it simply asks the sender to re-send packet #92. It will keep trying until it succeeds or after a time-out period it will give up and report an error to the application.

Setup: Open your personal Ubuntu Virtual Machine (VM) using VSphere Client and be sure you are at a CLI with the "midshipman" user prompt. You will have sudo NOPASSWD privileges on your machine, so be careful!

EE487: Applications of Cyber Engineering Name(s):
SX-10: IP Routing & TCP Connections

Section 1: IP routing

1. Last lab we left off with the command "`ip route show`" and we got a "`default`" route with a particular IP address. Do that again and write down what you see. The default IP address is the "gateway" router to the Internet. The top line is the route for the local LAN. There may or may not be other entries, but disregard them for now. The computer's IP layer uses the routing table to figure out where to send packets. First looks to see if the IP address of the destination is local, and if not it looks up the gateway's address and sends it there for further routing to the Internet (or another network outside our LAN).

   a. What is the IP address and network of your computer? Write it in two different (equivalent) ways: <u>CIDR notation</u> and as an <u>IP address and network</u> mask.

   b. What is the network address of the network that your computer is on? **Hint**: It should be the very first address listed in the routing table.

   c. The routing table also contains a "`dev`" (device). Why?

   d. Can you figure out why the first routing table entry says to send all packets destined for the local network to your computer? `Hint`: Does any other computer on the network use your routing table, or is it just for your computer?

   e. What happens if the IP address you want to send to is not in the local network?

2. Once IP figures out where to route a packet for the destination IP address, it now knows which data-link to use for service (from the routing table "`dev`" entry). So it hands off the packet to the data-link layer and asks it to send it. But there's a problem. We can't send data-link frames to IP addresses.

   a. What type of address is used on the data-link?

   b. How does the data-link figure out the address for the destination? Perform this command.

   c. Type in the necessary command (see the last SX if necessary) to find the correct destination for a frame destined for your neighbor's computer. **Hint**: you'll need to ask him what his IP address is, and then look in the output from step b.

   d. Ethernet has a value called an MTU. What is that value for your eth1 adapter, and what does it mean?

3. Now let's see what happens if we want to send something to the Internet. Let's say that you want to communicate with a web site of your choice. Type in "**`dig +noauth www.apple.com`**" or your website of choice. Also try another useful tool called "**`nslookup www.apple.com`**" DiG is just a bit more useful for administrators and **`nslookup`** gives quick simple answers. **Note:** if for some reason any of this step fails to work on your Ubuntu VM, use **`nslookup`** on your Windows host instead.

   a. See if you can figure out the IP address of the webserver. Using your web browser (Firefox), go to http://www.apple.com (or website of choice) and then try going to http://XXX.XXX.XXX.XXX where the X's are the IP address returned by **`dig/nslookup`**. If you got more than one IP address, try them both. Why might you get two or more answers?

   b. What do you think would happen if you type in a website URL (the http://www.apple.com) and the DNS system was somehow not functioning?

   c. There are some other domain names listed above the IP address returned by DiG. DNS is a hierarchal system. What does that mean?

   d. So now let's assume that we're trying to send an IP datagram to this website IP address. IP hands the datagram off to Ethernet for transmission. Where does Ethernet send it?

   e. What MAC address does it use? Look it up using the tools we've used so far (**Hint:** use the ARP list).

   f. What happens to the packet after the last step? (If all went well, the packet should have made it to the final layer)

EE487: Applications of Cyber Engineering Name(s):
SX-10: IP Routing & TCP Connections

Section 2: TCP connections and ports

4. The problem with what we've done so far is that we have no idea if the packets we send are actually arriving at the destination. IP does its best, but it is "fire and forget" and does not care about the packet after it sends it. Type in the command "**`netstat -t`**" (If using Windows, drop the "**`-t`**"). This will display the current state of TCP connections on your computer.

   a. Why does your computer need to keep track of TCP connections?

   b. Can you figure out which connection is for the website that you visited in the last step? Jot down that line.

   c. What is the source port?

   d. What is the destination port?

   e. What is the "state" of that connection?

   f. What three TCP packets must be sent before any connection is established?

   g. What TCP packet is sent to end a connection?

   h. When do you think that the computer adds and deletes entries from this table?

   i. Try opening another browser window and go to the very same website. Then look at the connection table again. What do you notice about the source and destination ports? Can you explain this?

Section 4: Making mischief

5. Recall from the first section of the course that we could rather easily gain root shell access to a computer using a buffer overflow attack. You happen to have sudo (root) access to your machine, so let's use that for the moment.

   a. What might happen if you were able to change the ARP table of your classmate's computer and replaced the entry for the default route (gateway to the Internet)?

   b. What could you replace that entry with to intercept the data that he sends?

   c. Could you do something similar with the routing table instead? What?

   d. Try to modify your own **ARP** table. **Hint**: Use the "`ip neighbor change`" command. Use the man page if necessary.
   Try to modify your own **routing** table. **Hint**: Use the "`ip route change`" command. Were you able to make these changes?

EE487: Applications of Cyber Engineering Name(s):
SX-10: IP Routing & TCP Connections

**Conclusion and Results:**
Your typed lab report will consist of two paragraphs, in the first paragraph:
• Briefly describe what you did in the lab in your own words.
• Discuss something new that you learned.

In the second paragraph, answer the questions:
• How could an adversary use this knowledge or these tools for malicious purposes?
• How could you use your new understanding to protect your systems and personnel from attack?

**Staple** the completed report to the back of your original lab and turn it in to your instructor at the beginning of the next class.