EE487: Applications of Cyber Engineering Name:
SX-12: Network Surveillance

Discussion: At this point we're ready to put everything we've learned about how networks work together and see some actual end-to-end communications between computers. We now hopefully understand how an application on one end of a conversation (called the client) initiates a connection to a computer on the other end (called the server) which responds (if everything goes as planned) and provides information. The conversation between these computers proceeds until one of the parties decides to terminate the connection, or something else goes wrong and something in that complicated chain of protocol stacks causes the conversation to crash.

We've also learned about how packets get encapsulated for transmission across the physical network medium. Here's the question to ponder: have we ever discussed how those individual packets might look if we could intercept them along the way? Think of a scenario like what citizens of North Korea probably experience. If they write a letter and address it to someone in another country, they put it in an envelope. Do you think the letter makes it out of the country without the state police opening it up and reading it? Is the information inside the letter somehow protected by anything other than the envelope? Do you think that IP packets are any different? Can the authorities learn anything just from the address on the envelope alone?

We've also learned that there are actually quite a few protocols being used all at the same time. The application has protocols (HTTP in the case of a web server, or MIME and SMTP in the case of email) as does the session layer (sockets), the transport layer (TCP, UDP), the network layer (IP), the datalink layer (Ethernet) and so on. Protocols are literally everywhere in a network.

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!

Reading: This lab follows very closely with the Erickson text from page 213 to page 217 (section 0x427). Sections 0x430 through 0x441 will give you a deeper understanding of the material.

Hacking Culture Note: A common Hacker insult is to call someone a "404." From the reading can you figure out what it means?

EE487: Applications of Cyber Engineering Name:
SX-12: Network Surveillance

Section 1: Tiny Web Server

1. The C code starting on page 214 is actually a web server program.

    a. What are the socket functions that a server must perform?

    b. What state is a server in just after running and before any other computer connects to it?

    c. Somewhere in the code is a section designed for that waiting state. What function (C code) does it start with?

    d. What is the server designed to do when a client connects to it?

2. The ports less than 1024 are traditionally known as "secure" ports. This is a misnomer in hindsight because they actually offer a hacker a great opportunity.

    a. Why are they called "secure?"

    b. Assume you have root privileges. Can you bind any program to port 80?

    c. What does it mean to bind a program to a port?

3. Compile and run the *tinyweb.c* program just as the author does at the bottom of page 216.

    a. Why is it necessary to perform the "**`chown`**" and "**`chmod`**" commands? What do they do to our program?

    b. What else needs to be true of our new program's attributes for it to run (hint: think **`chmod`**)?

4. Now connect to your own computer's web server from Firefox in Ubuntu. Yes, you can actually connect within your own computer. The loopback address is specifically for this purpose. When you use it, Firefox doesn't know the difference, but all of the IP traffic is redirected back to your own computer rather than going out on the Ethernet.

    a. What is the actual loopback IP address?

    b. Add a "/image.jpg" onto the end of the loopback address URL and see what happens.

    c. Add a "/favicon.ico" onto the end and see what happens. Why does it give a "URL not found" error?

5. Now let's connect to your lab partner's computer instead of doing so through your own computer. Instead of the loopback address, use your lab partner's IP address. Do you get the same results?

Finally, let's introduce you to the program "**`wireshark`**". So far you've just been looking at the end results of the program and the html code that the web server sends. Remember that all of that data had to go over the Ethernet (Wi-Fi in our case, but it's the same idea) to go from one computer to the other. The application data is in HTML format, encapsulated in TCP segments, encapsulated in IP packets, and then finally encapsulated in Ethernet frames. Let's see if we can see the data at that level. **Note**: Sections 0x430 – 0x433 are an excellent review if you're struggling with the concept of layering. Section 0x440 is highly advisable to understand what we're about to introduce.

6. One lab partner run the tiny web server as before. The other partner should run the commands below, and then connect to the web server using Firefox.

```
sudo ifconfig eth1 promisc
sudo tcpdump –l –X 'ip host XXX.XXX.XXX.XXX'
```

    Note: replace the X's above with your own IP address.

    a. What does the "promisc" option do?

b. What does an Ethernet interface normally do with regard to frames with MAC addresses not its own?

c. Note what you see in the terminal. It's a lot of gobblety-gook, but you probably can make out some intelligible stuff in the stream.

7. Now open the program "Wireshark" on your Ubuntu VM (just type the command "**`sudo wireshark –i eth1 –k –l &`**" at the CLI). Cool, we just opened a GUI based program from the CLI. Wireshark makes it a lot easier for human beings to see what's going on. **Note**: it won't work correctly if you forget the "**`sudo`**". Now run the tinyweb server again and connect to it (each partner run the server and then connect to the partner's server with Firefox). Exit from Firefox, then stop the capture and observe the Wireshark output.

   a. Sort the output by the protocol field. Do you recognize any of the protocols listed?

   b. Look at the ARP protocol messages. What is going on there?

   c. Are you seeing only the traffic that is going to and from your own computer? Why?

   d. It looks like some of the MAC addresses are strange. Rather than have all numbers, some of them have names on the front. What's going on there?

8. Isolate the conversation that occurs between you and your partner's computer using the IP addresses. Do this by finding a single packet that is going between the desired IP addresses, right click on it and select the option to "Conversation Filter: IP"

    a. Can you find the TCP three-way handshake at the beginning of the connection? Write down how you can tell.

    b. Can you find the end of the connection? How can you tell? **Note:** if you failed to close Firefox before ending your capture, you won't find this.

9. The possibilities with Wireshark are virtually endless. Have some fun going to a website of your choice and capturing the packets. Then try to isolate the conversation. Better yet, try to intercept your neighbor's conversation with the website he is visiting. We'll be using Wireshark several times throughout the semester.

EE487: Applications of Cyber Engineering Name:
SX-12: Network Surveillance

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.