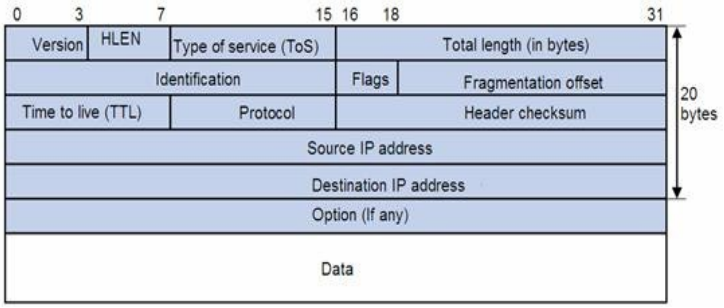
CSCD 433/533
Raw Socket Lab – Lab 6
Winter 2017

Raw sockets allow IPv4 protocols to receive or send raw packets with headers not including link level headers.

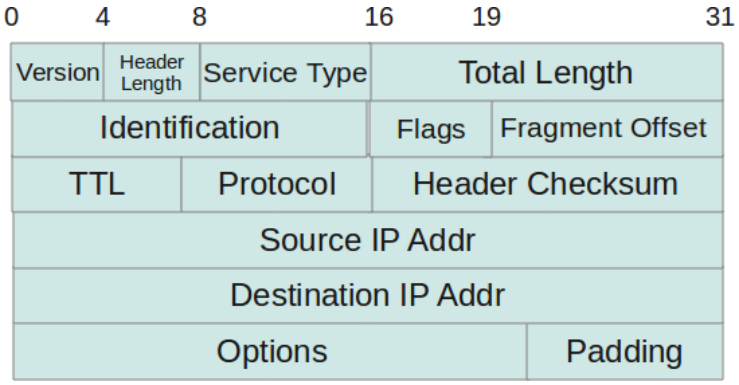
In the normal case, the operating system takes care of the header for the packet to be sent over the network. However, in the case that the application needs to use its own header for security purposes like a packet sniffer or pen-tester application there is a need to use Raw sockets.

We can describe the layout of a raw TCP packet.

Packet = IP Header + TCP Header + Data



TCP Header
No Options = 20 bytes



IP header
No Options = 20 bytes

If you were processing packet data, you would need to skip over the headers in order to get to your data. As shown in class, there are ways to figure out the header lengths and skip them.

Task 1: TCP Protocol
Download the rawsockets.zip file. Save it in a directory and unzip it.

You will need to first compile the .c files with gcc.
The provided C files should work fine with Linux and MacOSX.

Compile the provided C file named TCP.c . Then run it with root privileges as follows:

gcc -o TCP TCP.c

$ sudo ./TCP

When the TCP server runs is displays
"`Waiting for incoming connections...`"

Open wireshark and start capturing on the "lo" interface. lo = Localhost

Open a new terminal and type in, "**telnet localhost 8888**" as the client side.

Observe the packets in Wireshark. Stop the Capture.

Answer the following questions:

1- What is the message you got from the server?
2- Examine the source code from TCP.c. Describe briefly what it is doing.
3- Is this program using Raw Sockets?

Task 2: UDP Protocol

Compile the provided C files named UDPs.c and UDPc.c, capture output in wireshark and run it with root privileges. See following.

gcc -o UDPs UDPs.c
gcc -o UDPc UDPc.c

Begin a new capture in Wireshark on the lo interface.
Run the server and then the client each in their own windows.

$ sudo ./UDPs
$sudo ./UDPc

Stop the capture.
Observe the traffic in Wireshark.

Anwer the Questions

1- How many messages were sent from the server?
2- How many responses did you receive?
3- What message was sent and what was the reply?
4- What source port and destination ports are used?
5- What happens if first you run the client without the server?
6- What is the purpose of using SOCK_RAW as opposed to SOCK_DGRAM? In other words, why would you use SOCK_RAW?

Task 3: Hping3 flooder

In this task you will explore the program **hping3**.
This program is based on raw sockets and has various options for flooding, testing hosts.

You can ask it to send a lot of packets that look like they come from random IP addresses or check out open ports on the local or other hosts.

Because, we could not run it in the lab, first install it to your computer.

$ sudo apt-get install hping3 - if you have Linux. Its that easy.

This should install it and then, you need to be root to run it. Use sudo.

First, run Wireshark and start to sniff the lo interface.

Run the hping3 command to see if some ports are open.

$ sudo hping3 -S -p 80 -c 4 127.0.0.1
-S sets the Syn flag for a TCP packet
-p 80 sends packets to port 80
-c 4 sends at most 4 packets

Stop the capture.

Examine the packets in Wireshark.

Questions
1. How does this traffic differ from “normal” TCP traffic?
2. What is the response of your localhost machine back? What tcp flags seem to be set? Can you explain the response?
4. What can an attacker learn from executing this hping3 command?

Redo the command and now check the port 25. Use -p 25 instead of -p 80.
Only, Answer Question 2 again above.

Now, redo the command but this time send packets to Google.
First, start a Wireshark capture on the eth0 interface or your wireless interface if at home.

$sudo hping3 -S -p 80 -c 4 www.google.com

Answer the questions above again.

Craft the hping3 command to target the IP address of your router.
Not to worry, you can stop hping3 without flooding the router.
In fact, you can use the -c option to limit the number of packets.

Start a Wireshark capture on the eth0 interface if in the lab. Or, your wlan0 if at home.
Use a tcp filter to make it easier to see traffic.

$ sudo hping3 -S -p 25 -c 15 127.0.0.1

Stop Wireshark.

5. Look at your traffic again in Wireshark. What does it look like?

6. As the sender, what responses from this
   scan are you getting back?

One more test with hping3 will use random ip addresses.
If at home, do it against lcoalhost.

Start a new Wireshark capture. Use the wlan0 interface if at home.
Run the following hping3 command.

$ sudo hping3 -S -p 25 -c 15 - - rand-source router-ip

Stop the capture and view the packets sent.

7. Are there strange looking IP addresses displayed? List the first 3.
   Why would an attacker use random IP addresses? What is the attacker seeing on his machine?

Answer the questions. Send them in an email. CSCD433-Lab 6

References

https://www.darkmoreops.com/2014/08/21/dos-using-hping3-spoofed-ip-kali-linux/
https://linux.die.net/man/8/hping3
http://0daysecurity.com/articles/hping3_examples.html