CSC535: UDP Checksum

The goal of this project was to create a python program that accurately calculates a UDP checksum. The project submission, "Udp_checksum.py", is a program that accepts a string input of a UDP datagram's hexadecimal values (only for datagrams built over IPv4) and calculates the UDP checksum. Following is a description of how to use the program:

The program udp_checksum.py is hardcoded with the hex stream provided in the project instructions as the input datagram. If simply viewing the functionality, the program can be ran from the terminal with "python udp_checksum.py". The program will print out the appropriate fields from the datagram used for the calculation, followed by the calculated UDP checksum value.

0. Extract the hex stream

a. A hex stream can be extracted from a captured packet through the program WireShark. The captured packet must be a UDP packet built over IPv4. Figure 1 displays how you can copy the UDP datagram as a hex stream by right clicking the data window on a WireShark packet view. The data must be copied with the highlighted settings selected.

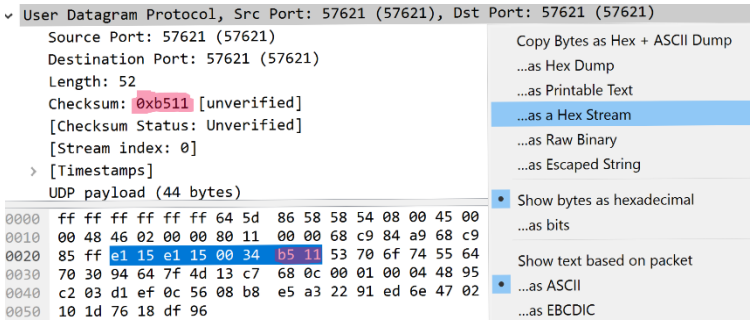

Figure 1: Wireshark output for UDP datagram

b. This will copy a single string of hexadecimal values (unspaced and without 0x-prefix).

1. Input the hex stream: 3 ways

a. The hex stream can be hardcoded into the program. The main() function is located near the bottom of the file. Within the main(), the first line defines a variable "hex_stream" equal to a default hex stream. Replace the String value within the quotation marks with the hex stream you extracted in step 1. Then, the program can be ran from the terminal with "python Udp_checksum.py".

b. The hex stream can be input through a .txt file. Store the hex stream extracted in step 1 into a .txt file within the same folder where the Udp_checksum.py file is located. Then, run the program from the terminal and input the hex stream's .txt file with the command "python udp_checksum.py (hex stream file name here).txt". An example file, "UDPpacket.txt" has been provided in the submission for testing.

c. The hex stream can be input directly from the terminal. Similar to method b, run the program from the terminal and input the hex stream with the command "python udp_checksum.py (hex stream value here)".

```
PS C:\Users\brain\OneDrive\Documentos\SCHOOL\CSC\535\P2> py Project2.py UDPpacket.txt
~~~ Datagram ~~~
~~~ IP Psuedo-Header ~~~
Protocol ID: 0x11
Source Port: 0x68c9 0x84a9
Destination Port: 0x68c9 0x85ff
~~~ UDP Header ~~~
Source Port: 0xe115
DestinationPort: 0xe115
UDP Length: 0x34
~~~ UDP Payload ~~~
['0x5370', '0x6f74', '0x5564', '0x7030', '0x9464', '0x7f4d', '0x13c7', '0x680c', '0x1'
01d', '0x7618', '0xdf96']
>>> UDP Checksum: 0xb511
```
Figure 2: Example output of Project2.py

2. Output:

a. The program will print out the appropriate fields from the datagram used for the calculation, followed by the calculated UDP checksum value. Figure 2 displays an example of the output derived by inputting the hex stream captured in figure 1 through the provided “UDPpacket.txt” file. The checksum value calculated is equal to the checksum value highlighted in Figure 1.

3. Implementation Summary

The program accepts a hex stream (as derived above) and parses through the string to store the appropriate values in a series of DatagramSection objects. These values are given to a ChecksumCalculator type object to return the checksum calculation.