PartC - Wireshark

Sravya Beesabathuni

March 5, 2018

1 Tcpdump commands for the three ports:

```
sudo tcpdump -G 5 -W 1 'host sbunetsyslabs.com and port 1080' -w ~/Desktop/http_1080.pcap
Password:
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
Maximum file limit reached: 1
2637 packets captured
2809 packets received by filter
0 packets dropped by kernel

sudo tcpdump -G 5 -W 1 'host sbunetsyslabs.com and port 1081' -w ~/Desktop/tcp_1081.pcap
Password:
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
Maximum file limit reached: 1
2496 packets captured
3076 packets received by filter
0 packets dropped by kernel

sudo tcpdump -G 5 -W 1 'host sbunetsyslabs.com and port 1082' -w ~/Desktop/tcp_1082.pcap
Password:
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
Maximum file limit reached: 1
2043 packets captured
2100 packets received by filter
0 packets dropped by kernel
```

2 High level view of the analysis_pcap_http code

- Firstly, the pcap files for each port are generated using above mentioned command lines.

- Function call implemented for HTTP analysis getHTTPRequestsAndResponses() . To be done only for port 1080.
- For each packet, we find out the dataoffset value which resides in the first 4 bits of 12th byte after the first 34 bytes (which consists of Etherenet and IP header).
- Then we find the payload length, by packetLength - 4*dataoffset .Our payload byte array starts from dataoffset with this length.
- Hence for each packet, we have the HTTP header information in payload byte array.
- For each flow, we loop through all the sent packets using sourceMap and find a match by matching with the received packets (receiveMap).
- If the acknowledgement number of the sent packet is equal to the sequence number of the received packet, then there is a match.
- We parse the payload byte array, convert it to String and split with "slash r slash n" to get the HTTP header information.

```
public static void parseHTTPHeader(byte[] payload) {
    final Charset UTF8_CHARSET = Charset.forName("UTF-8");
    String httpHeader = new String(payload, UTF8_CHARSET);
    String[] headerParts = httpHeader.split("\r\n");
    for (int i=0;i<headerParts.length-3;i++) {
      System.out.println(headerParts[i]);
    }
  }
```

- Loop through this, to get the header information
- We can observer that the received packets have HTTP/1.0 written in the header. Ex:

```
=====REQUEST=======Source : 51522 Destn : 1080 Seq : 3348514599 Ack : 851662159
====HEADER===
GET /img/raspberrypi.jpeg HTTP/1.1
Host: www.sbunetsyslabs.com:1080
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTM
Accept: image/webp,image/apng,image/*,*/*;q=0.8
=====RESPONSE=======Source : 1080 Destn : 51522 Seq : 851662159 Ack : 3348515032
====HEADER===
HTTP/1.0 200 OK
```

```
Date: Mon, 05 Mar 2018 07:43:47 GMT
Server: Apache/2.4.25 (Ubuntu)
Last-Modified: Wed, 15 Feb 2017 00:46:39 GMT
ETag: "247c-54887037b3e66"
Accept-Ranges: bytes
Content-Length: 9340
Connection: close
```

- Apart from this, also calculating the total number of bytes and the total time for the entire transaction to compare all the 3 pcap files and to identify the version of HTTP

3 analysis_pcap_http code

Attached along with output logs

4 Answers

- Identification of HTTP Versions:
  - **http_1080.pcap** : It uses HTTP/1.0 as can be observed in the header information of the received packet pasted above.

    ```
    Total number of packets in pcap file : 2471
    Total Number of TCP Flows initiated by the sender = 17
    Total time taken : 1521.0 ms
    Total number of bytes : 2171948.0
    ```

  - **tcp_1082.pcap** : It uses HTTP/2.0 as the number of flows(connections) is only 1 but the number of bytes transferred is considerably high.

    ```
    Total number of packets in pcap file : 2043
    Total Number of TCP Flows initiated by the sender = 1
    Total time taken : 353.0 ms
    Total number of bytes : 2257322.0
    ```

  - **tcp_1081.pcap** : Since tcp_1082.pcap uses HTTP/2.0 , tcp_1081.pcap uses HTTP/1.1 as the number of connections are more considered to the prev pcap file with nearly same amount of bytes getting transferred.

    ```
    Total number of packets in pcap file : 2496
    Total Number of TCP Flows initiated by the sender = 4
    Total time taken : 1362.0 ms
    Total number of bytes : 2319188.0
    ```

- Other Observations:
  - Fastest : HTTP/2.0
  - Slowest: HTTP/1.0
  - Most packets: HTTP/1.1
  - Least packets: HTTP/2.0
  - Most Bytes: HTTP/1.1
  - Least Bytes: HTTP/1.0
- HTTP/2.0 supports headers compression and has more intelligent packet streaming management and hence takes least amount of time to get most of the bytes with minimum number of connections.