

Measuring HTTP server performance

● Graded

Group

AVANISH R SAMALA

AKSHAJ KAMMARI

[✎ View or edit group](#)

Total Points

44.4444 / 100 pts

Autograder Score

44.4444 / 100.0

Failed Tests

Test case 3: pcap file 3 (5.55555555555556/8.33333333333334)
Test case 4: pcap file 3 (1.38888888888889/8.33333333333334)
Test case 5: pcap file 4 (4.16666666666667/8.33333333333334)
Test case 6: pcap file 4 (4.16666666666667/8.33333333333334)
Test case 7: pcap file 5 (4.16666666666667/8.33333333333334)
Test case 8: pcap file 5 (1.38888888888889/8.33333333333334)
Test case 9: pcap file 5 (0/8.33333333333334)
Test case 10: pcap file 6 (4.16666666666667/8.33333333333334)
Test case 11: pcap file 6 (1.38888888888889/8.33333333333334)
Test case 12: pcap file 6 (1.38888888888889/8.33333333333334)

Passed Tests

Test case 1: pcap file 1 (8.33333333333334/8.33333333333334)
Test case 2: pcap file 2 (8.33333333333334/8.33333333333334)

Autograder Results

Test case 1: pcap file 1 (8.33333333333334/8.33333333333334)

Test case 2: pcap file 2 (8.33333333333334/8.33333333333334)

Test case 3: pcap file 3 (5.55555555555556/8.33333333333334)

Test case 4: pcap file 3 (1.38888888888889/8.33333333333334)

Test case 5: pcap file 4 (4.16666666666667/8.33333333333334)

Test case 6: pcap file 4 (4.16666666666667/8.333333333333334)

Test case 7: pcap file 5 (4.16666666666667/8.333333333333334)

Test case 8: pcap file 5 (1.38888888888889/8.333333333333334)

Test case 9: pcap file 5 (0/8.333333333333334)

Test case 10: pcap file 6 (4.16666666666667/8.333333333333334)

Test case 11: pcap file 6 (1.38888888888889/8.333333333333334)

Test case 12: pcap file 6 (1.38888888888889/8.333333333333334)

Submitted Files

```
1  from scapy.all import *
2  import sys
3  import time
4  import math
5
6  load_layer("http")
7
8  if len(sys.argv) != 4:
9      print("Usage: measure-webserver.py <input-file> <server-ip> <server-port>")
10     sys.exit(1)
11
12  pcap = sys.argv[1]
13  servip = sys.argv[2]
14  servport = sys.argv[3]
15
16  sessions = rdpcap(pcap).sessions()
17
18  http = {}
19  latencies = []
20
21  for session in sessions:
22      for packet in sessions[session]:
23          if packet.haslayer(TCP):
24              if HTTPRequest in packet:
25                  reqtime = packet.time
26                  http[(packet[IP].src, packet[TCP].sport, packet[IP].dst, packet[TCP].dport)] = reqtime
27
28              elif HTTPResponse in packet:
29                  reqkey = (packet[IP].dst, packet[TCP].dport, packet[IP].src, packet[TCP].sport)
30
31                  if reqkey in http:
32                      latencies.append(packet.time - http[reqkey])
33
34  def calculate_percentile(data, percentile):
35      if len(data):
36          sortdata = sorted(data)
37          return sortdata[int(round(percentile / 100 * (len(data) - 1)))]
38
39      else:
40          return None
41
42  if latencies:
43      print("AVERAGE LATENCY: {:.5f}".format(sum(latencies) / len(latencies)))
44      print("PERCENTILES: {:.5f}, {:.5f}, {:.5f}, {:.5f}, {:.5f}".format(*[calculate_percentile(latencies, p) for p in
45      [25, 50, 75, 95, 99]]))
```

```
46 else:
47     print("No HTTP request-response pairs found to calculate latency.")
48
```