Prerna Arote (SR No 15230)

Question 1 – Network Packet Analysis : Solution

Part a) Trace 1 [HTTP Traffic]

1. www.tripadvisor.com,
   www.baidu.com,
   www.bing.com,
   www.amazon.com
2. "Adventures in stochastic processes"
   "Madison map"
   "Chicago metro"

Part b) Trace 2 [FTP Traffic]

1. Username: shiningmon
   Password: public
2. FTP connections use two ports to send files a data and a control port. So, two different port numbers need to be opened on the server and client. Active FTP opens both ports by having the client connect to port 21 on the server from an arbitrary port for control. The server then responds by connecting to the arbitrary port on the client plus one for data. Usually any service trying to connect to a port from outside network to client will be blocked by firewall unless the connection has already been established from the client to that service.
   Passive FTP connections solve this problem with active FTP connections being firewalled by having the server respond with an open port number for the client to connect to for data. This way all client connections are established from the client connecting to open ports on the server and not the other way around.
3. There is no active connection found. That is none of the FTP connections are active.
4. All FTP connections are passive. Packet range for passive connection is given below:
   i)
   - 30: PASV command sent from client
   - 34: client opens connection from port 58758 to port 55513
   - 41: Server issue FIN to close the data connection

   ii)
   - 75: PASV command sent from client
   - 78: Client opens connection from 58760 to port 50800
   - 94: Server issues FIN to close the data connection

   iii)
   - 110: PASV command sent from client
   - 114: Client opens connection from port 58761 to port 58774
   - 121: Server issues FIN to close the data connection

   iv)
   - 146: PASV command sent from client
   - 154: Client opens connection from port 58763 to port 50042
   - 167: server issues FIN to close the data connection

v)
- 180: PASV command sent from client
- 184: Client opens connection from port 58764 to port 58890
- 193: Server issues FIN to close the data connection

vi)
- 217: PASV command sent from the client
- 220: Client opens the connection from port 58765 to port 58898
- 229: Server issues FIN to close the data connection

vii)
- 243: PASV command sent from the client
- 247: Client opens the connection from port 58766 to port 40607
- 258: Server issues FIN to close the data connection

5. Files downloaded are:
   "dragon.zip"
   "ARP.java"
   "L2switch.java"
   "phase1.html"

Part c) Trace 3 [Traceroute]

1. Source IP: 192.168.0.100
2. Destination IP: 74.125.225.46
3. 192.168.0.1
   10.131.180.1
   96.34.20.20
   96.34.17.95
   96.34.16.112
   96.34.16.77
   96.34.2.4
   96.34.0.7
   96.34.0.9
   96.34.3.9
   96.34.152.30
   209.85.254.120
   209.85.250.28

Part d) Trace 4 [POP]

1. Username: "cs155@dummymail.com"
   password: "whitehat"

2. Number of emails in User mailbox are 5

3. From: "joe <cs155@dummymail.com>"
   To: "cs155@dummymail.com"
   Subject: "foobar"

Date: “Friday, 23 Apr 2010 08:20:52 -0700”

From: “joe <cs155@dummymail.com>”
To: “cs155@dummymail.com”
Subject: “can you see this subject?”
Date: “Friday, 23 Apr 2010 08:23:25 -0700”

From: “hariny <harinym@standford.edu>”
To: “cs155@dummymail.com”
Subject: “test message”
Date: “Friday, 23 Apr 2010 10:25:00 -0700”

From: “hariny <harinym@standford.edu>”
To: “cs155@dummymail.com”
Subject: “geology rocks!”
Date: “Friday, 23 Apr 2010 08:22:28 -0700”

From: “hariny <harinym@standford.edu>”
To: “cs155@dummymail.com”
Subject: “wassup”
Date: “Friday, 23 Apr 2010 08:21:50 -0700”

Question 2 – Network based Denial-of-service: Solution

Part a)

- The setup mentioned have done offline packet logging. With offline packet logging a large SYN DoS attack could fill the storage capacity of the IDS machine’s disk and probably it will crash it.
- The IP header struct is not ipv6 compatible and could either bypass the IDS system or possibly exploit it to crash the logger
- There is no verification of the IP or TCP headers which could exploited. Such as line 76 and line 80 the IP and TCP header lengths are not checked to be valid which could be used to spoof the check on line 78 or the payload extraction on line 80 bypass logging. An arbitrary offset in memory could be logged as the payload.
- The way the report buffer is created on line 24 and written to by the payload with memcpy on line 45 appears to be open to an integer overflow attack which with a buffer overflow could lead to arbitrary code execution crashing the IDS, or even maliciously logging packets for adversary.

Part b)

- Making the offline packet logging online would fix the issue of filling the machine’s disk, or deleting old logs could possibly fix it.
- Use a dual stack header or implementation to handle both IPv4 and IPv6 packets.

- Verify all inputs that are used to control the program including packet headers for IP and TCP

Question 3) Solution – Please, find it in main.py