CS4035D Computer Security
QUESTION 3
BOTNET SCENARIO

YACHA VENKATA RAKESH
B180427CS
S6 BTECH CSE A

The whole assignment basically deals with implementing DDos attack using a master and more than one bot attacking a victim

1. 4 VM's are used. Master - Kali Linux, Slaves - Ubuntu, And Victim - Windows XP.
2. Assigned IP addresses are Master : 172.16.28.4
   Slave1 : 172.16.28.6
   Slave2 : 172.16.28.8
   Victim : 172.16.28.10

Firstly using Metasploit using msfvenom made a malicious file which opens connection to the attacker's system whenever executed. So when this is being installed to the bot machine by bot unknowingly thinking this as a useful application attacker get shell access. So now there are two codes made tcp_server.py and tcp_client.py which is basically to connect the client with the server. This is done using multithreading and hence it supports concurrency and hence multiple users can access this at once. So in the attacker's machine tcp_server.py will be remotely running and as the attacker now has shell access to bot's machine he could run tcp_client.py code now in the server code the attacker can send one of the four commands.

Basic Command format is SEND TCP-SYN/UDP/HTTP IP_ADDR COUNT
STOP

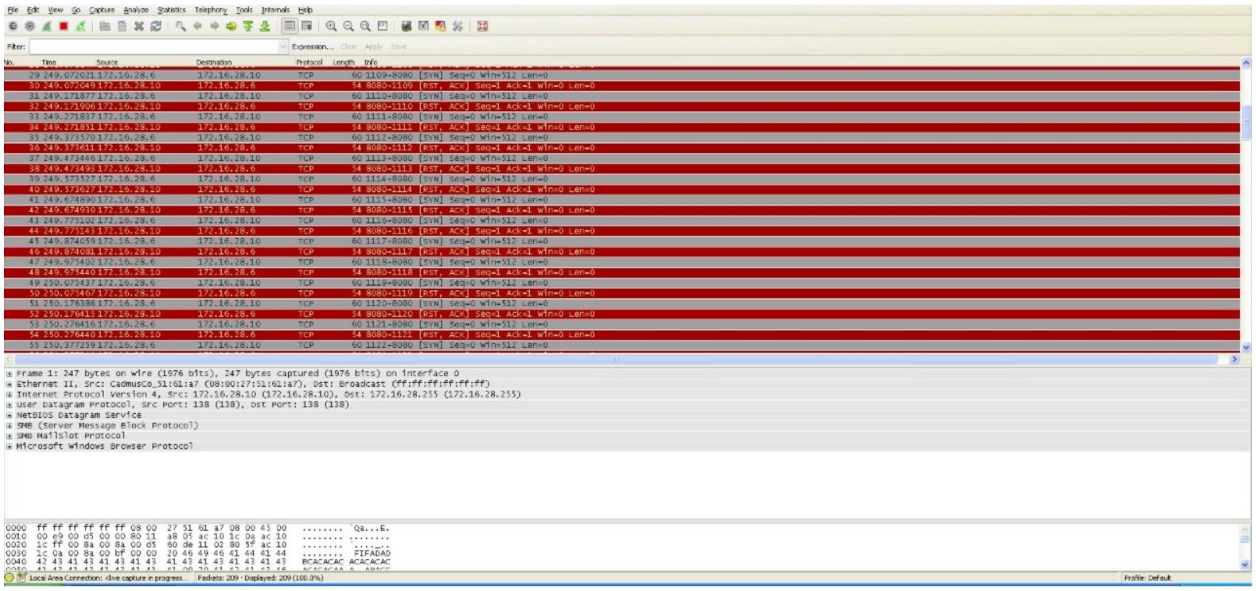
1. SEND TCP-SYN 172.16.28.10 15 // Here 15 is just count of number of TCP-SYN packets to be sent and 172.16.28.10 is the IP address of the Victim Machine
2. SEND UDP 172.16.28.10 20
3. SEND HTTP 172.16.28.10 30
4. STOP.

When the attacker sends this STOP command as it uses multithreading one thread will be dedicated to listening and when it listens to STOP command it immediately stops sending TCP packets. So all this information can be seen through the wireshark by installing in either victim or slave systems that Slave is the sender and the victim is the receiver. So we will be flooding TCP or UDP or HTTP packets from both the slaves onto the victim. When this limit reaches beyond the control of the Victim DDos is successful.

Additional Files used:

1. MSFvenom
   Malicious code containing payloads which is basically to connect to the attacker's server whenever the file is executed.
2. TCP Server
   a. Create Socket, Bind with IP address and Port number and start listening for connections (basically bots here)
   b. Whenever a new connection requested it starts a new thread to accept them and have a count and unique identification of each of the slaves.
   c. Now after enough number of bots are made the attacker can input a command.
   d. As attacker has complete access to the bots using their unique identifications this input command will be broadcasted to stored list of active bots.
   e. On receiving the command successfully bot will send a confirm message so that master can know if the attack is successfully going on.
   f. On receiving stop command as of now program will be stopped but it can also be changed such that using someother keyword as delimiter we can continue program but just terminate the ongoing attack.
3. TCP Client
   a. Create socket, and connect to the IP address of the Master using the port and IP mentioned in the code.
   b. It will go through an infinite loop waiting for master command.
   c. It will categorize the commands into four types based on the command received from the master.
   d. I am using python inbuilt library scapy to make tcp or udp or http packets and then send them directly based on the count in the command.
   e. Alternatively we can use other built in command os.command(" ") to execute the command
   f. And whenever it gets STOP message from the master it immediately stop sending any ongoing packets and disconnect from the master and close the socket

Note that only for the first time the bot has to run the file later once the attacker gets shell access then he can move the file to the startup directory. It varies machine to machine like os to os. In windows we just need to move to startup folder and in linux we need to move it to /bin folder and should open cronetab and edit the file by adding @reboot and file path followed by & (to tell to run this in background). So by this connection becomes persistent and attacker will have access to the system even after multiple reboots.


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source         Destination     Protocol Length Info
   29 249.072021 172.16.28.6    172.16.28.10    TCP      60 1109→8080 [SYN] Seq=0 Win=512 Len=0
   30 249.072049 172.16.28.10   172.16.28.6     TCP      54 8080→1109 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   31 249.171877 172.16.28.6    172.16.28.10    TCP      60 1110→8080 [SYN] Seq=0 Win=512 Len=0
   32 249.171906 172.16.28.10   172.16.28.6     TCP      54 8080→1110 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   33 249.271837 172.16.28.6    172.16.28.10    TCP      60 1111→8080 [SYN] Seq=0 Win=512 Len=0
   34 249.271851 172.16.28.10   172.16.28.6     TCP      54 8080→1111 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   35 249.373570 172.16.28.6    172.16.28.10    TCP      60 1112→8080 [SYN] Seq=0 Win=512 Len=0
   36 249.373611 172.16.28.10   172.16.28.6     TCP      54 8080→1112 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   37 249.473446 172.16.28.6    172.16.28.10    TCP      60 1113→8080 [SYN] Seq=0 Win=512 Len=0
   38 249.473493 172.16.28.10   172.16.28.6     TCP      54 8080→1113 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   39 249.573527 172.16.28.6    172.16.28.10    TCP      60 1114→8080 [SYN] Seq=0 Win=512 Len=0
   40 249.573627 172.16.28.10   172.16.28.6     TCP      54 8080→1114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   41 249.674890 172.16.28.6    172.16.28.10    TCP      60 1115→8080 [SYN] Seq=0 Win=512 Len=0
   42 249.674930 172.16.28.10   172.16.28.6     TCP      54 8080→1115 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   43 249.775102 172.16.28.6    172.16.28.10    TCP      60 1116→8080 [SYN] Seq=0 Win=512 Len=0
   44 249.775143 172.16.28.10   172.16.28.6     TCP      54 8080→1116 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   45 249.874059 172.16.28.6    172.16.28.10    TCP      60 1117→8080 [SYN] Seq=0 Win=512 Len=0
   46 249.874081 172.16.28.10   172.16.28.6     TCP      54 8080→1117 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   47 249.975402 172.16.28.6    172.16.28.10    TCP      60 1118→8080 [SYN] Seq=0 Win=512 Len=0
   48 249.975440 172.16.28.10   172.16.28.6     TCP      54 8080→1118 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   49 250.075437 172.16.28.6    172.16.28.10    TCP      60 1119→8080 [SYN] Seq=0 Win=512 Len=0
   50 250.075467 172.16.28.10   172.16.28.6     TCP      54 8080→1119 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   51 250.176386 172.16.28.6    172.16.28.10    TCP      60 1120→8080 [SYN] Seq=0 Win=512 Len=0
   52 250.176415 172.16.28.10   172.16.28.6     TCP      54 8080→1120 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   53 250.276416 172.16.28.6    172.16.28.10    TCP      60 1121→8080 [SYN] Seq=0 Win=512 Len=0
   54 250.276440 172.16.28.10   172.16.28.6     TCP      54 8080→1121 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   55 250.377259 172.16.28.6    172.16.28.10    TCP      60 1122→8080 [SYN] Seq=0 Win=512 Len=0
```

```
⊞ Frame 1: 247 bytes on wire (1976 bits), 247 bytes captured (1976 bits) on interface 0
⊞ Ethernet II, Src: CadmusCo_51:61:a7 (08:00:27:51:61:a7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 172.16.28.10 (172.16.28.10), Dst: 172.16.28.255 (172.16.28.255)
⊞ User Datagram Protocol, Src Port: 138 (138), Dst Port: 138 (138)
⊞ NetBIOS Datagram Service
⊞ SMB (Server Message Block Protocol)
⊞ SMB MailSlot Protocol
⊞ Microsoft Windows Browser Protocol
```

```
0000  ff ff ff ff ff ff 08 00  27 51 61 a7 08 00 45 00   ........ 'Qa...E.
0010  00 e9 00 d5 00 00 80 11  a8 05 ac 10 1c 0a ac 10   ........ ........
0020  1c ff 00 8a 00 8a 00 d5  60 de 11 02 80 5f ac 10   ........ `...._..
0030  1c 0a 00 8a 00 bf 00 00  20 46 49 46 41 44 41 44   ........  FIFADAD
0040  42 43 41 43 41 43 41 43  41 43 41 43 41 43 41 43   BCACACAC ACACACAC
```

```
Local Area Connection: <live capture in progress...   Packets: 209 · Displayed: 209 (100.0%)   Profile: Default
```

Like these we can capture those traffic in the wireshark in the victim’s machine for TCP-SYN, UDP and HTTP.
Clear screenshots are attached in the screenshots folder number properly. Haven’t included here as there is a limit for report to 2-3 pages.

Thank you sir