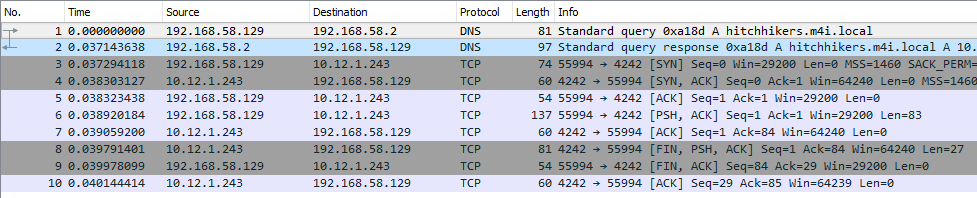
Capt Alamri, 2Lt Hayden, 2Lt Mireles

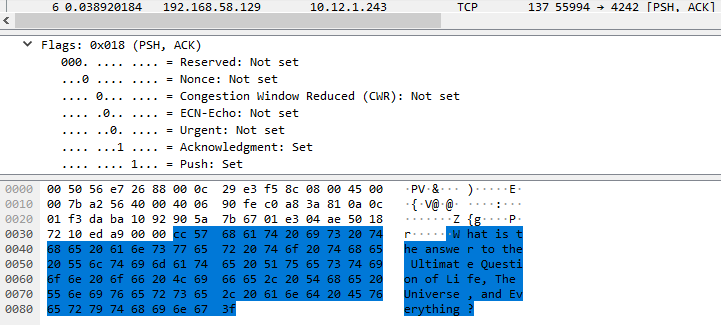
Cyber Attack Project Writeups

**Task 1: Hitchhiker’s Guide to the Galaxy**

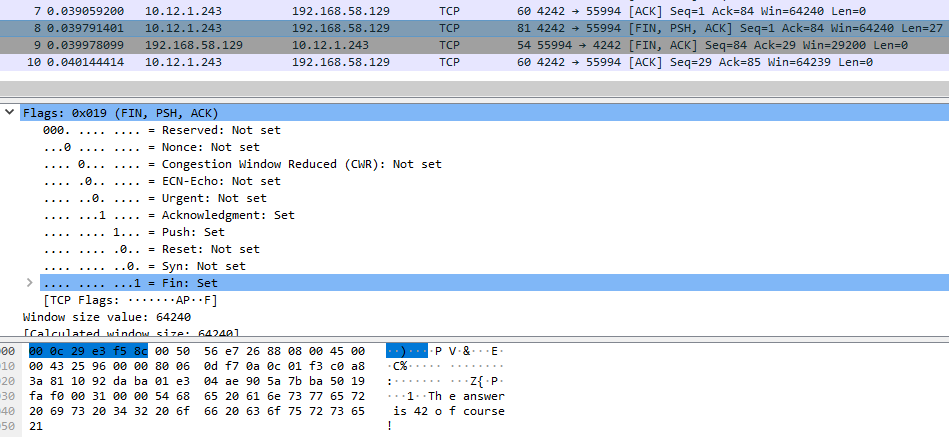
How did the server respond? Why? Explain the ports used by both computers as well as the flags set for all packets. Provide a Wireshark screenshot showing just the packets flowing between your computer and the server.

10 Packets captured in Wireshark, the source and destination ports are boxed in red:

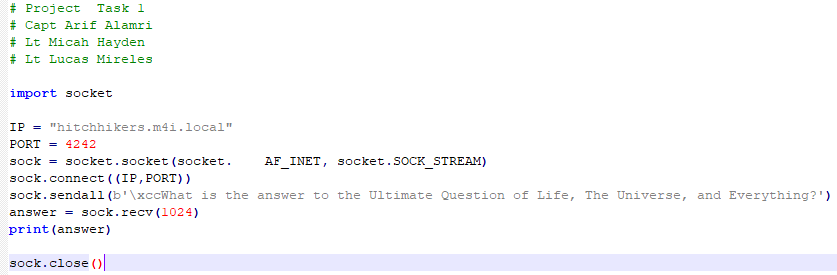
* The server uses port 4242 and our computer uses port 55994.
* Packets 3, 4, and 5 are used to complete the 3 way handshake.
  + Packet 3 – SYN flag set, from our computer
  + Packet 4 – response from server, SYN and ACK flags set
  + Packet 5 – local computer sending ACK packet with no data
* Packet 6 has the ACK and PUSH flags set, telling the server to immediately send a response. The data contains our question to the server, preceded by 0xcc.



* Packet 7 – from server to our client, is an ACK packet with no data, followed by its response to the question we sent in Packet 8.
* Packet 8 – from server to our client, has the FIN flag set, indicating the termination of the TCP session, the PUSH flag set – indicating it wants an immediate response, and the ACK flag set indicating successful receipt of our previous packet.
  + Its data contains the response to our question: “The answer is 42 of course!”



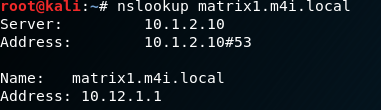
* Packet 9, from our computer to the server, has the FIN and ACK flags set, indicating the acknowledgement of the previous packet, as well as allowing the TCP connection to close.
* Packet 10, from the server, has the ACK flag set, indicating it received the previous packet successfully.

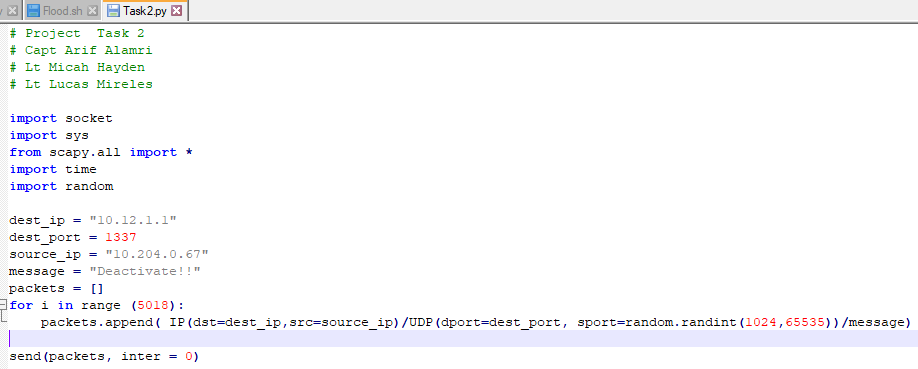
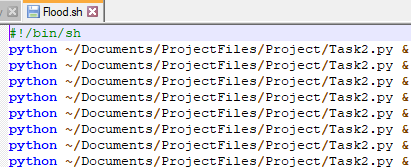
**Python code used:**

**Task 2: *Matrix Sentinel* Flood**

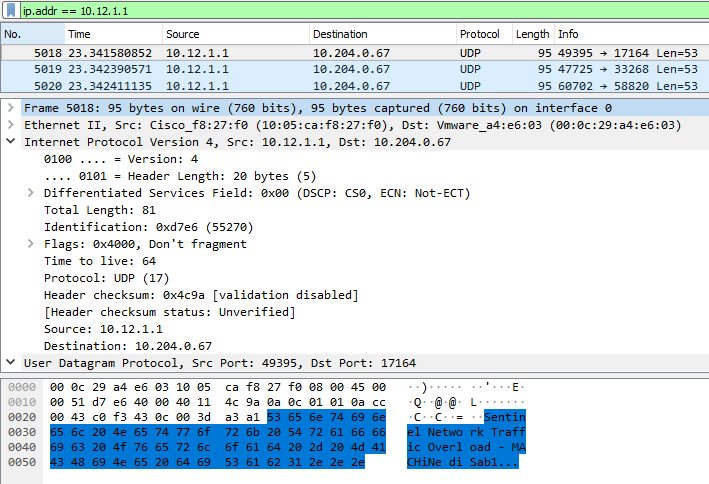
**Steps taken:**

* NS Lookup of destination IP: produces IP address of **10.12.1.1**



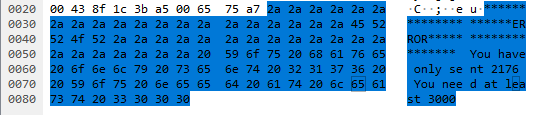
* Ran ifconfig on “partner’s” machine to determine what to set spoofed source IP address to
* Created Task2.py to generate and send spoofed packets using scapy.
  + We utilized a packets array to store spoofed packets to increase sending speed.
  + Each spoofed UDP packet contained the above destination IP, destination port, spoofed source IP, and a random source port outlined in red.
* Created Flood.sh to execute the flood of UDP packets by running Task2.py several times concurrently to reach desired speed.

Wireshark capture of responses from partner’s computer after the UDP flood:

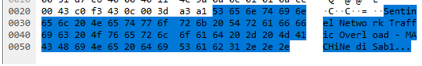


* UDP packets arriving from IP 10.12.1.1, the matrix1.m4i.local machine to partner’s machine at 10.204.0.67.
* Each source/destination port number differs each response:

Screenshot of Sentinel’s response during UDP flood:



Screenshot of Sentinel’s response after UDP flood was successful:



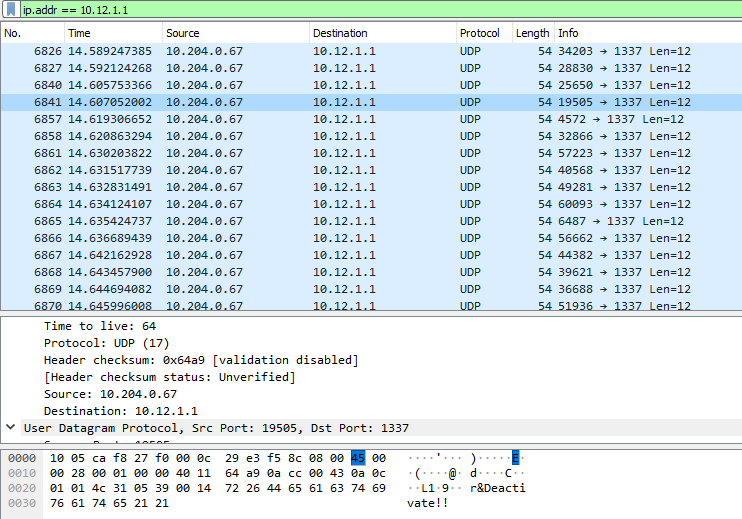
What is the msssage from the sentinel when your packet is successful? (i.e. the response does not contain “ERROR”).

* As shown in the previous Wireshark capture:
  + “Sentinel Network Traffic Overload – MACHiNe diSabl…”

What type of attack might use this spoofed IP feature coupled with an un-throttled flood of messages?

* A Denial of Service (DoS) attack would use this to send a stream of UDP packets and hopefully keep the source of the attack hidden.

**Screenshot from attacking computer Wireshark capture:**



* Source IP is 10.204.0.67, which is spoofed
* Destination IP is 10.12.1.1, address of matrix1.m4i.local
* Destination port is 1337 on all flood packets
* Source port is a random number between 1024 and 65535
* Message payload is “Deactivate!!”