Analyze Network Traffic with TCPDump

Description

I simulated that someone is trying to open SSH sessions into my workstation and decided to set up a surveillance script to catch any TCP traffic coming through as SSH.

I also did another script to catch any TCP traffic coming from a specific IP addresse.

Languages and Utilities Used

- TCPDump

Environments Used

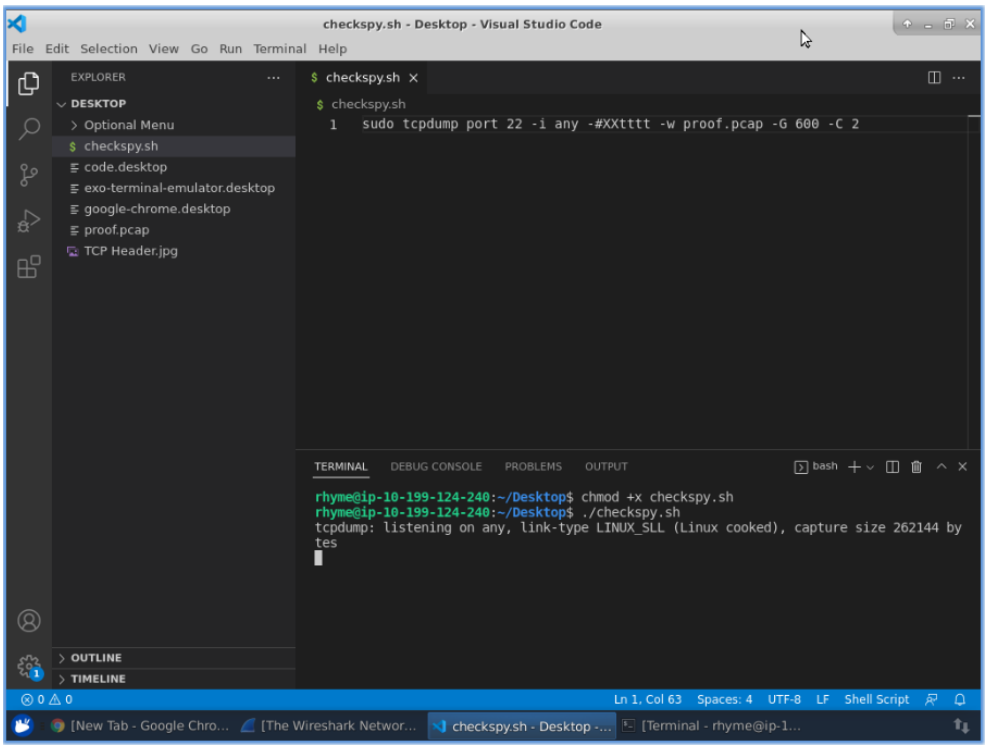
- Linux

Program walk-through:

Create a shell script file to capture SSH traffic and give it permissions

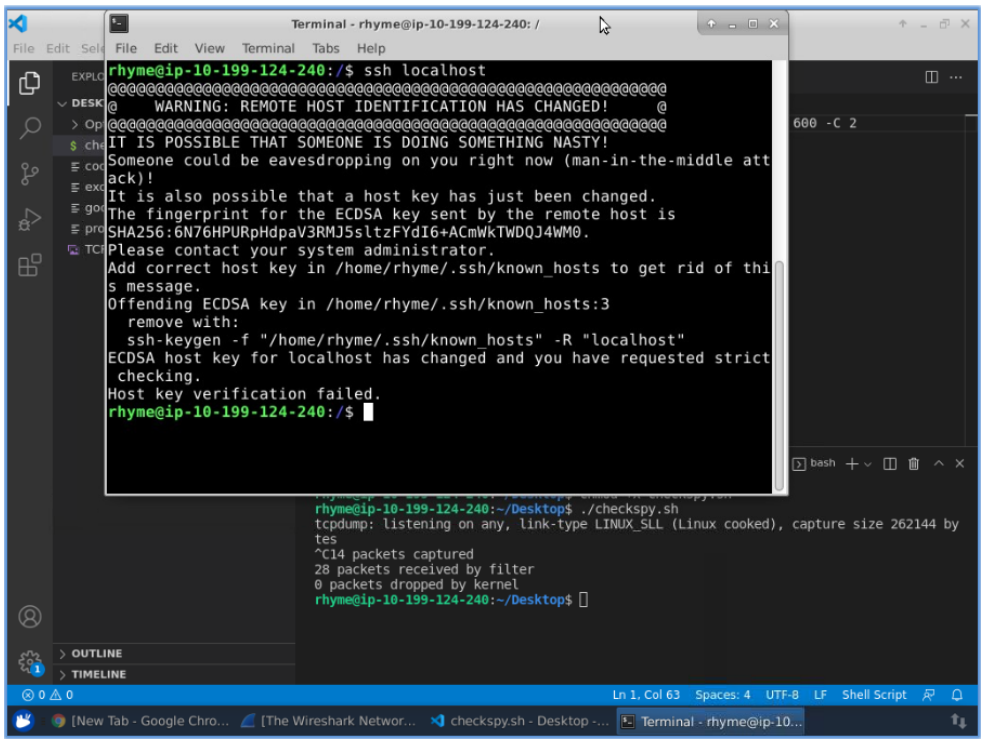
First I wanted to create a shell script file to capture SSH traffic with TCPDump through port 22 with the interface option *any,* since I´m testing this locally. I also wanted to print the timestamp of each captured packet in human-readable format and to write the captured packets to a file named proof.pcap with no more than 2 megabytes and 10 minutes of capture. To do that I used the command: *sudo tcpdump port 22 -i any -#XXtttt -w proof.pcap -G 600 -C 2*

After that, I gave it executable permissions in the terminal with the command: *chmod +x checkspy.sh*



Test the script

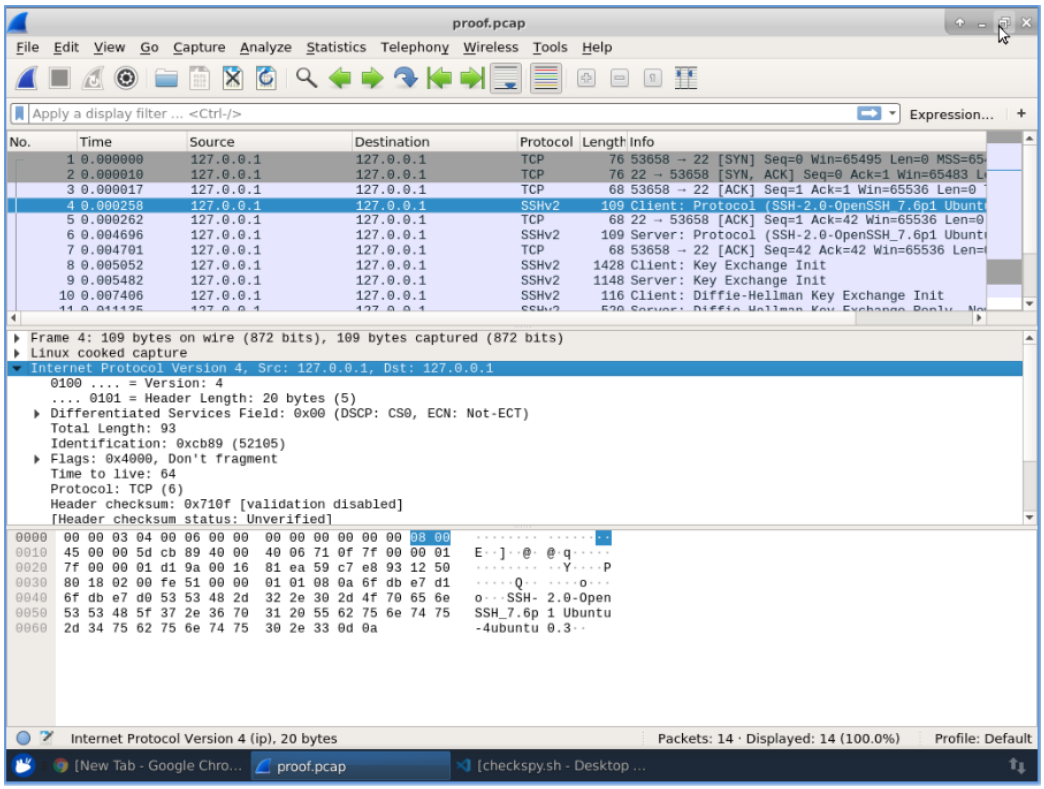
After launching the script to test it, I opened a terminal window and ran SSH localhost (it will not be successful, but it will generate SSH traffic)



As expected SSH localhost failed but generated some packets as shown in the terminal

Analyze the dump file with Wireshark

The SSH traffic is easily noticeable on Wireshark. Though the source IP is from our host machine, it would appear vastly different in real-world scenarios.

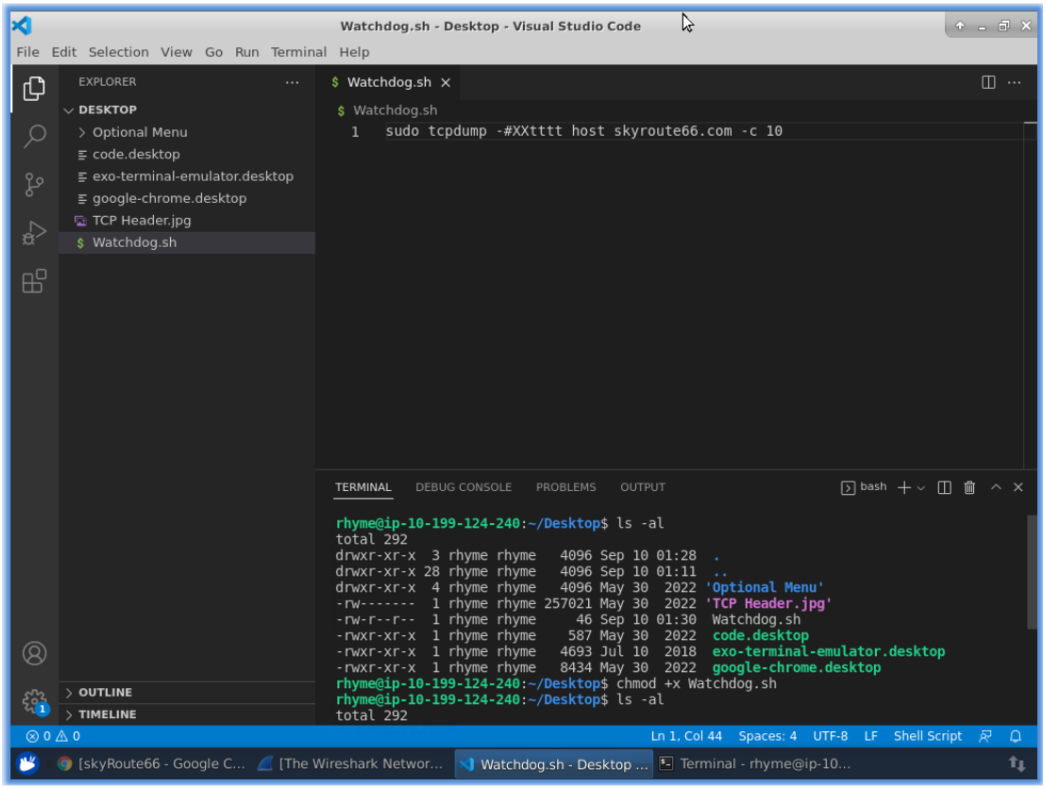


If there were numerous packets involved, it would be easier to locate them quickly by using display filters, in this case the Wireshark filter named *ssh*.

Create a shell script file to capture traffic coming from a specific IP addresse and give it permissions

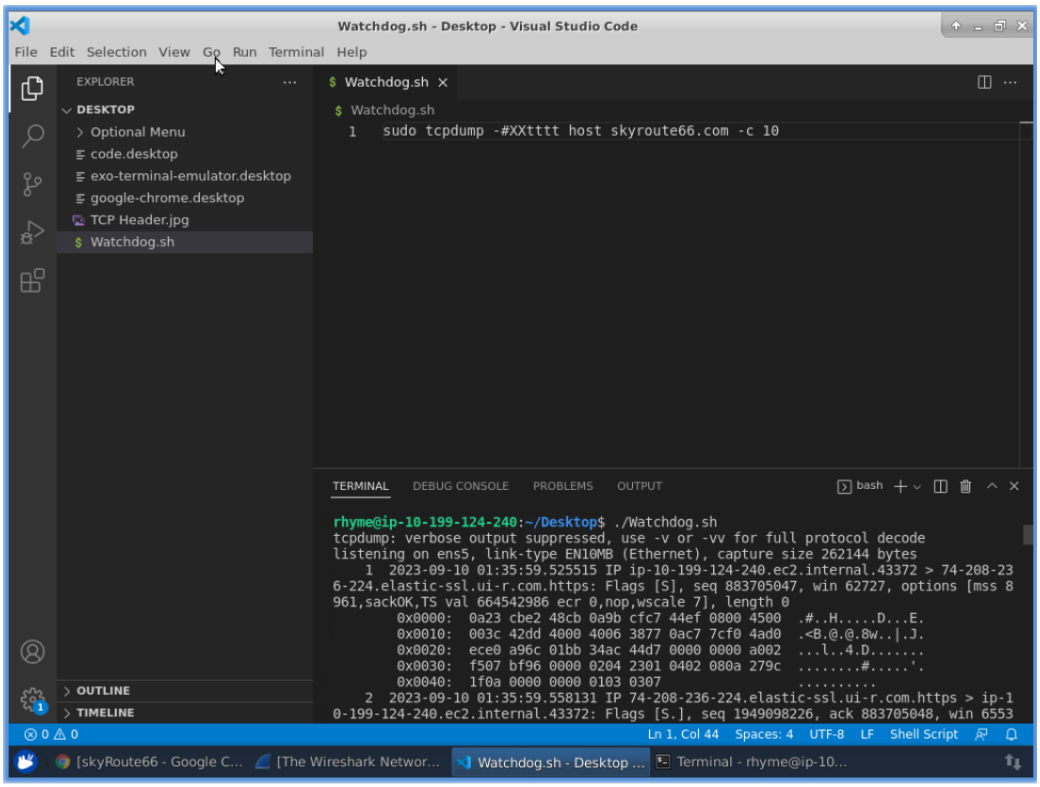
So I wanted to create a shell script file to capture 10 packets coming from *skyroute66.com* with TCPDump in which the timestamp of each captured packet is shown in human-readable format, to do that I used the command: *sudo tcpdump -#XXtttt host skyroute66.com -c 10*

After that, I gave it executable permissions in the terminal with the command: *chmod +x Watchdog.sh*

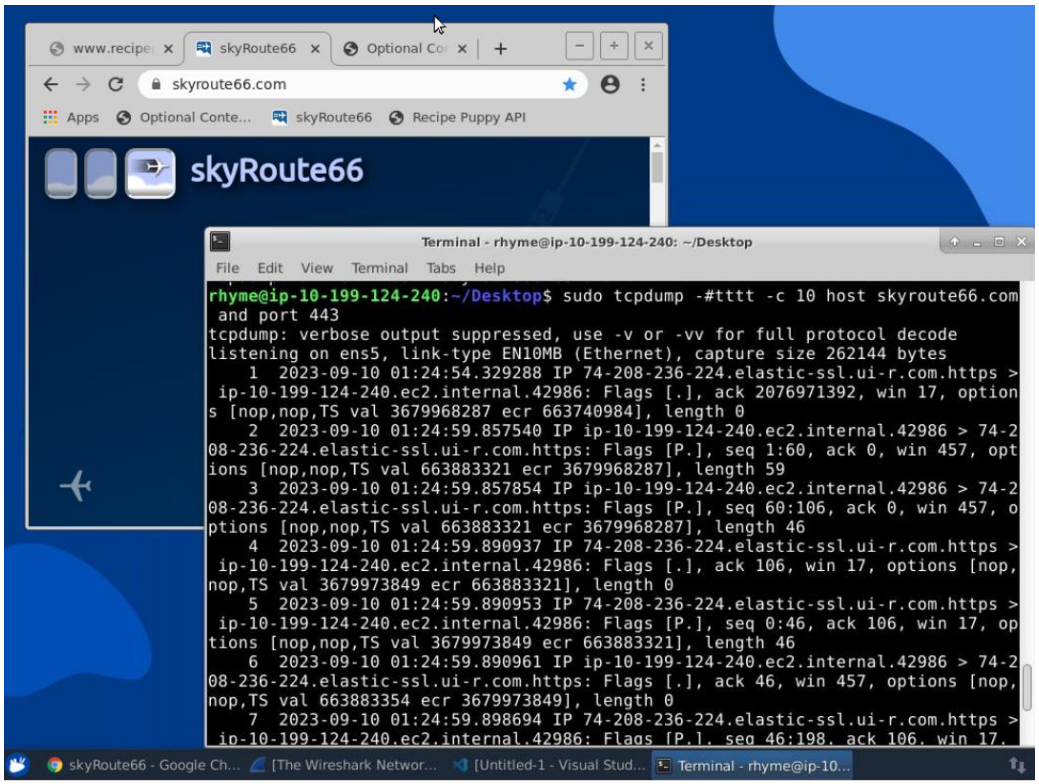


Test the script

To test the script I started it in the terminal and opened the website *skyroute66.com*



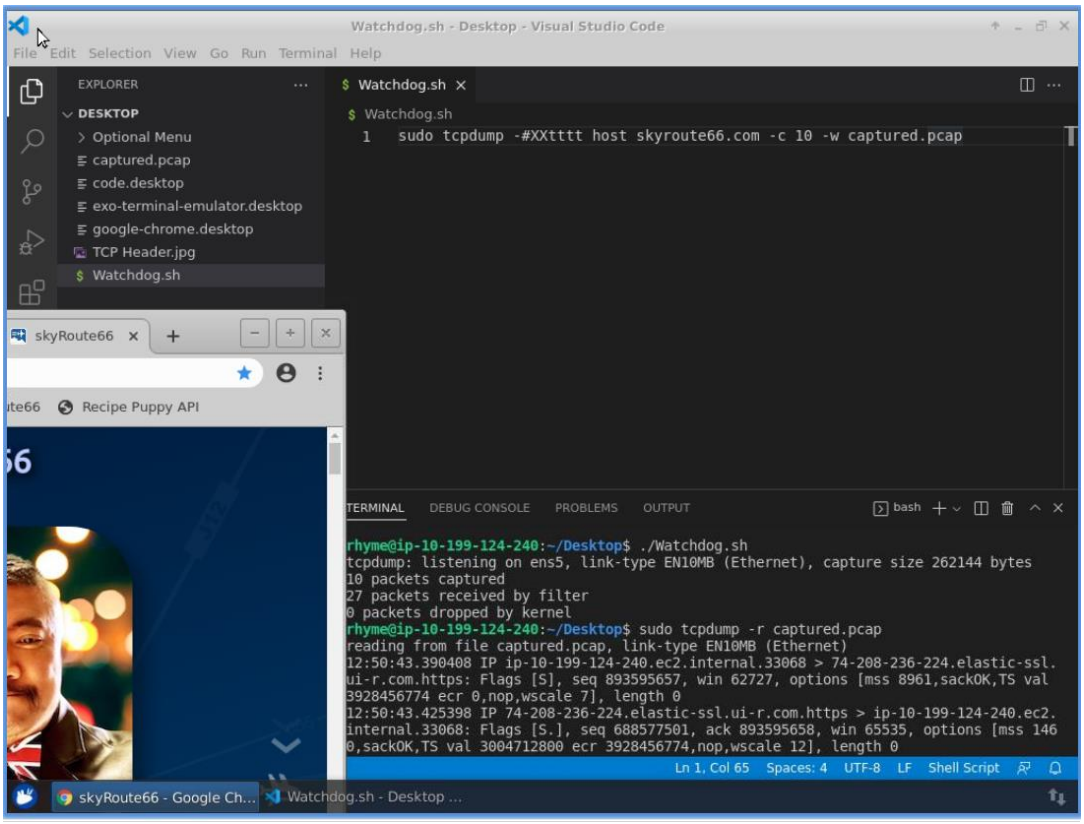
I could also use the following command to capture the same traffic with the same filters but only the traffic coming through port 443 (HTTPS traffic): *sudo tcpdump -#tttt -c 10 host skyroute66.com and port 443*



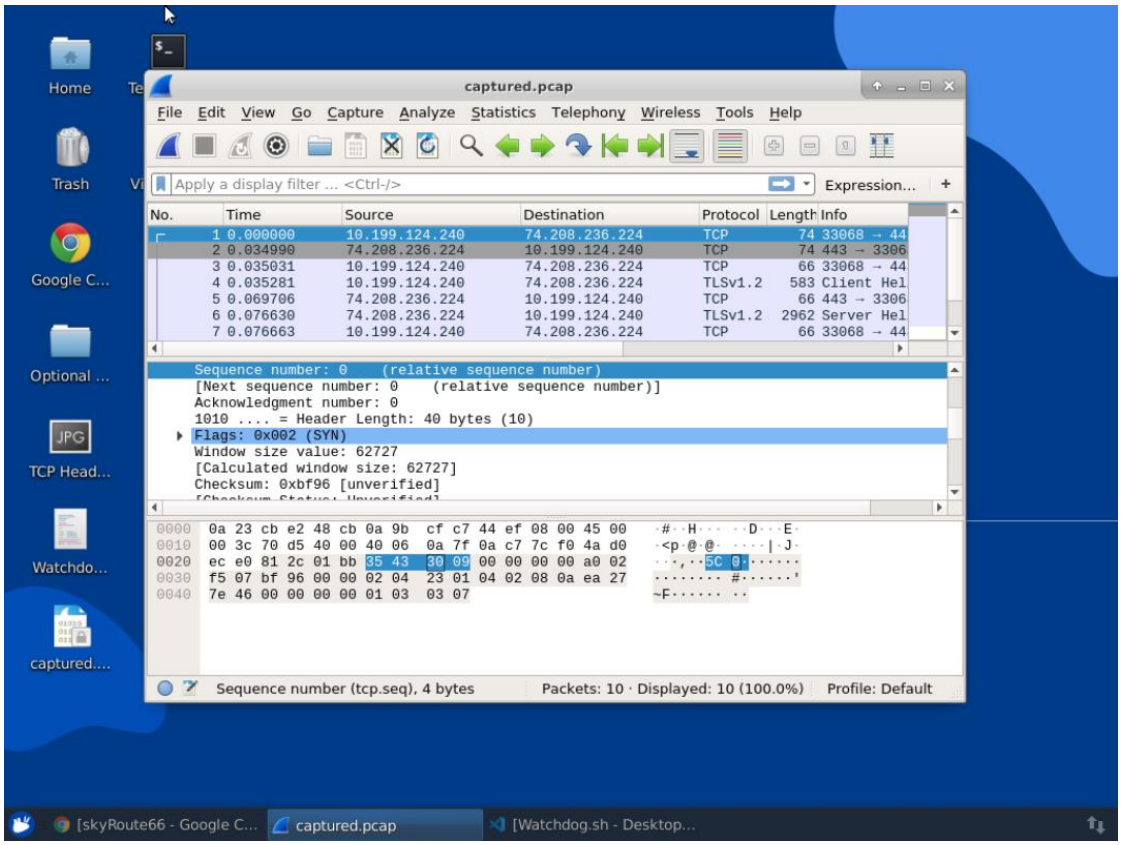
Add the option to create a dump file to the script and test

For the script to create a file with the captured packages I need to add the -w option, making the script look like this: *sudo tcpdump -#XXtttt host skyroute66.com -c 10 -w captured.pcap*

Then I ran the script, opened the website and read the file in the terminal with the command: *sudo tcpdump -r captured.pcap*



For a better analysis of the packets, I could analyze the dump file with Wireshark



Advanced Filtering: Zooming into Specific Narratives

Simulating that I just wanted to capture a specific part of a protocol, I used the following command to capture only HTTP GET requests: *sudo tcpdump -#XXtttt 'tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420'*