



Lab 9: OT/SCADA Penetration Testing

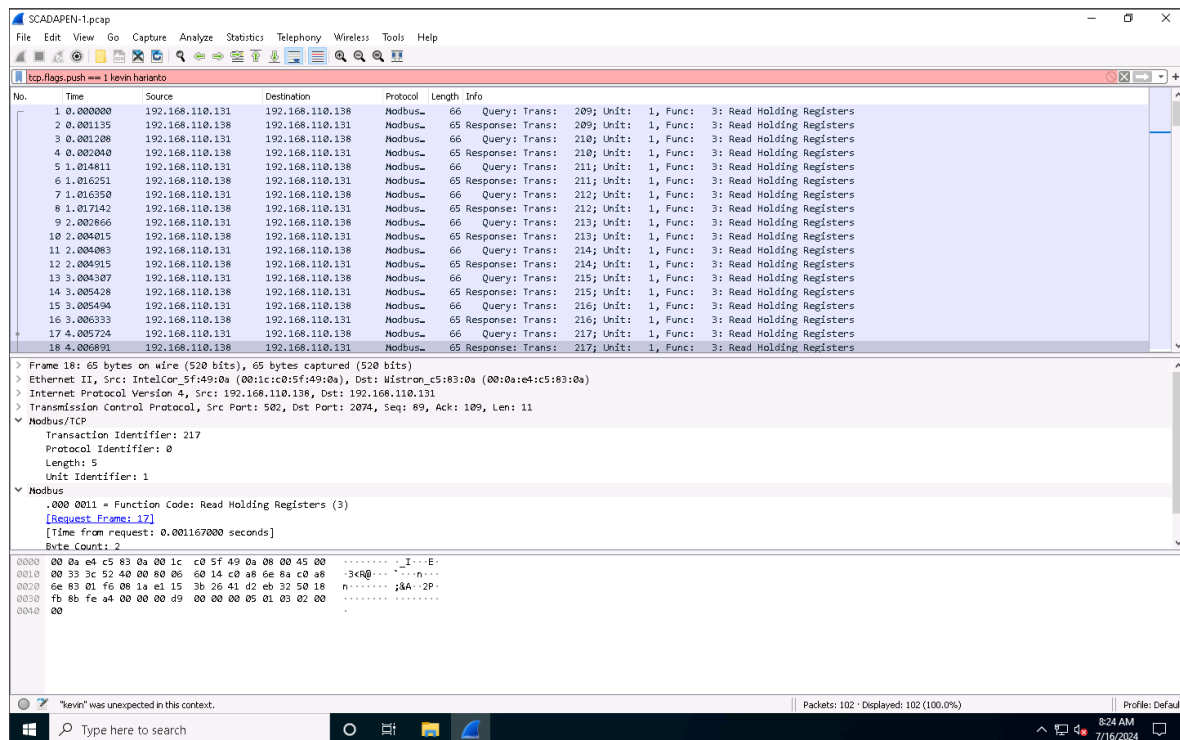
INFO40587: ETHICAL HACKING

Kevin Harianto | 991602128 | July 16, 2024

Exercise 1: ModBus Protocol Analysis - I

1.1 OUTPUT SCREENSHOTS

Exercise 1, Step 18: Now, select packet number **10** and observe the **Modbus** node in the middle section. Since the image is a response, the data include the information in the register. There are two types of places where information can be stored: coils and registers. Each of these datastore types has two different types of registers: a read/write and a read only. Each of these datastore types, in turn, is a reference to a memory address.



Exercise 1, Step 32: The process of applying this filter reduces the amount of packets to perform the process against. Observe at **lower right corner** of the window. The packets have been reduced from 35,430 to 11,490, which is an improvement. This way, you can make these conversations more digestible for analysis.

SCADAPEN-2.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src==172.27.224.250 and ip.dst==172.27.224.251 kevin.harianto

No.	Time	Source	Destination	Protocol	Length	Info
9	0.516082	172.27.224.250	172.27.224.251	TCP	60	502 → 52515 [ACK] Seq=1 Ack=2 Win=8712 Len=0
10	0.516916	172.27.224.250	172.27.224.251	TCP	60	502 → 52515 [FIN, ACK] Seq=1 Ack=2 Win=8712 Len=0
12	0.517094	172.27.224.250	172.27.224.251	TCP	60	502 → 52515 [RST, ACK] Seq=2 Ack=2 Win=8712 Len=0
13	0.520089	172.27.224.250	172.27.224.251	TCP	60	502 → 52515 [RST] Seq=2 Win=0 Len=0
31	1.966258	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
45	1.978056	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [ACK] Seq=1 Ack=6 Win=8712 Len=0
47	1.987125	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [ACK] Seq=1 Ack=13 Win=8707 Len=0
48	1.987974	172.27.224.250	172.27.224.251	Modbus	66	Response: Trans: 1; Unit: 1, Func: 6: Write Single Register
64	3.416135	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [ACK] Seq=13 Ack=14 Win=8712 Len=0
65	3.417004	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [FIN, ACK] Seq=13 Ack=14 Win=8712 Len=0
67	3.417178	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [RST, ACK] Seq=14 Ack=14 Win=8712 Len=0
68	3.420162	172.27.224.250	172.27.224.251	TCP	60	502 → 52516 [RST] Seq=14 Win=0 Len=0
88	4.866282	172.27.224.250	172.27.224.251	TCP	60	502 → 52517 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
102	4.877850	172.27.224.250	172.27.224.251	TCP	60	502 → 52517 [ACK] Seq=1 Ack=7 Win=8712 Len=0
103	4.887158	172.27.224.250	172.27.224.251	TCP	60	502 → 52517 [ACK] Seq=1 Ack=13 Win=8707 Len=0
104	4.887991	172.27.224.250	172.27.224.251	Modbus	66	Response: Trans: 1; Unit: 1, Func: 6: Write Single Register
122	6.316177	172.27.224.250	172.27.224.251	TCP	60	502 → 52517 [ACK] Seq=13 Ack=14 Win=8712 Len=0
123	6.317015	172.27.224.250	172.27.224.251	TCP	60	502 → 52517 [FIN, ACK] Seq=13 Ack=14 Win=8712 Len=0

Frame 9: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

Ethernet II, Src: Telemech_09:51:1b (00:80:14:09:51:1b), Dst: ASUSTekC_64:40:79 (48:5b:39:64:40:79)

Internet Protocol Version 4, Src: 172.27.224.250, Dst: 172.27.224.251

Transmission Control Protocol, Src Port: 502, Dst Port: 52515, Seq: 1, Ack: 2, Len: 0

Source Port: 502

Destination Port: 52515

Stream index: 1

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

Sequence number (raw): 1954923156

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 2 (relative ack number)

Acknowledgment number (raw): 534429383

0101 = Header Length: 20 bytes (5)

0000 48 5b 39 64 40 79 00 00 f4 09 51 1b 00 00 45 00 H[000y...Q::E

0010 00 28 fe 48 00 00 40 06 22 5a ac 1b e0 fa ac 1b (-H000Z.....

0020 e0 fb 01 f6 c0 23 74 05 e7 6a 68 c9 39 c4 f8 af 50 18 ...h9...P

0030 22 08 20 92 00 00 00 01 00 00 00 06 01 06 00 06 ..

0040 00 1e

"Kevin" was unexpected in this context.

Packets: 35430 · Displayed: 4980 (14.1%)

Profile: Default

8:29 AM 7/16/2024

Exercise 1, Step 36:

You have a request that is writing data **001e** to the register. You will not see the response, because, when you captured the specific packets, you only explored one direction. You can either show this by naming the file or capturing both sides of the conversation. When you return to the original file, you can review the conversation again. This time, select the filter for both directions and review the “**write**” statement to check if it was successful.

con1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr eq 172.27.224.250 and ip.addr eq 172.27.224.251 kevin.harianto

No.	Time	Source	Destination	Protocol	Length	Info
32	10.172823	172.27.224.250	172.27.224.251	Modbus	66	Response: Trans: 1; Unit: 1, Func: 6: Write Single Register
33	11.590179	172.27.224.250	172.27.224.251	TCP	60	502 → 52519 [ACK] Seq=13 Ack=14 Win=8712 Len=0
34	11.591432	172.27.224.250	172.27.224.251	TCP	60	502 → 52519 [FIN, ACK] Seq=13 Ack=14 Win=8712 Len=0
35	11.591613	172.27.224.250	172.27.224.251	TCP	60	502 → 52519 [RST, ACK] Seq=14 Ack=14 Win=8712 Len=0
36	11.600180	172.27.224.250	172.27.224.251	TCP	60	502 → 52519 [RST] Seq=14 Win=0 Len=0
37	13.040322	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
38	13.051840	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [ACK] Seq=1 Ack=7 Win=8712 Len=0
39	13.061466	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [ACK] Seq=1 Ack=13 Win=8707 Len=0
40	13.062789	172.27.224.250	172.27.224.251	Modbus	66	Response: Trans: 1; Unit: 1, Func: 6: Write Single Register
41	14.490243	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [ACK] Seq=13 Ack=14 Win=8712 Len=0
42	14.491855	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [FIN, ACK] Seq=13 Ack=14 Win=8712 Len=0
43	14.491253	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [RST, ACK] Seq=14 Ack=14 Win=8712 Len=0
44	14.500348	172.27.224.250	172.27.224.251	TCP	60	502 → 52520 [RST] Seq=14 Win=0 Len=0
45	15.940374	172.27.224.250	172.27.224.251	TCP	60	502 → 52521 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
46	15.951916	172.27.224.250	172.27.224.251	TCP	60	502 → 52521 [ACK] Seq=1 Ack=7 Win=8712 Len=0
47	15.961286	172.27.224.250	172.27.224.251	TCP	60	502 → 52521 [ACK] Seq=1 Ack=13 Win=8707 Len=0
48	15.962049	172.27.224.250	172.27.224.251	Modbus	66	Response: Trans: 1; Unit: 1, Func: 6: Write Single Register
49	17.390260	172.27.224.250	172.27.224.251	TCP	60	502 → 52521 [ACK] Seq=13 Ack=14 Win=8712 Len=0

Frame 48: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

Ethernet II, Src: Telemech_09:51:1b (00:80:14:09:51:1b), Dst: ASUSTekC_64:40:79 (48:5b:39:64:40:79)

Internet Protocol Version 4, Src: 172.27.224.250, Dst: 172.27.224.251

Transmission Control Protocol, Src Port: 502, Dst Port: 52521, Seq: 1, Ack: 13, Len: 12

Modbus/TCP

Modbus

0000 48 5b 39 64 40 79 00 00 f4 09 51 1b 00 00 45 00 H[000y...Q::E

0010 00 34 fe aa 40 00 00 21 ec ac 1b e0 fa ac 1b 4 00 1:.....

0020 e0 fb 01 f6 fd 2c e7 6a 68 c9 39 c4 f8 af 50 18 ...h9...P

0030 22 08 20 92 00 00 00 01 00 00 00 06 01 06 00 06 ..

0040 00 1e

"Kevin" was unexpected in this context.

Packets: 4980 · Displayed: 4980 (100.0%)

Profile: Default

8:32 AM 7/16/2024

1.2 Questions

Question 11.1.1

Use the sample SCADAPEN-1.pcap file available on the Desktop of the Scada Master machine to review the network traffic of communication among devices on a network that uses the ModBus protocol. Enter the length of the Response packet.

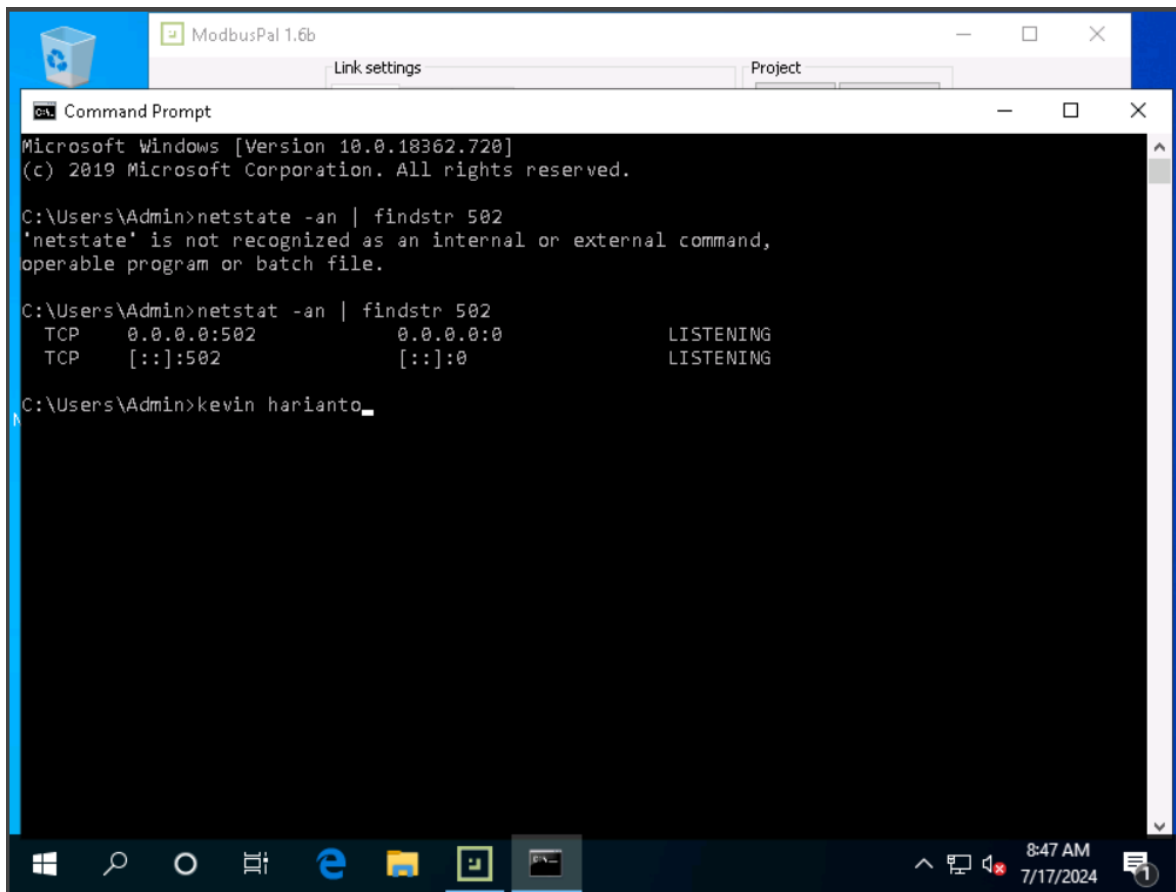
Score

✓ Correct

Exercise 2: ModBus Protocol Analysis - II

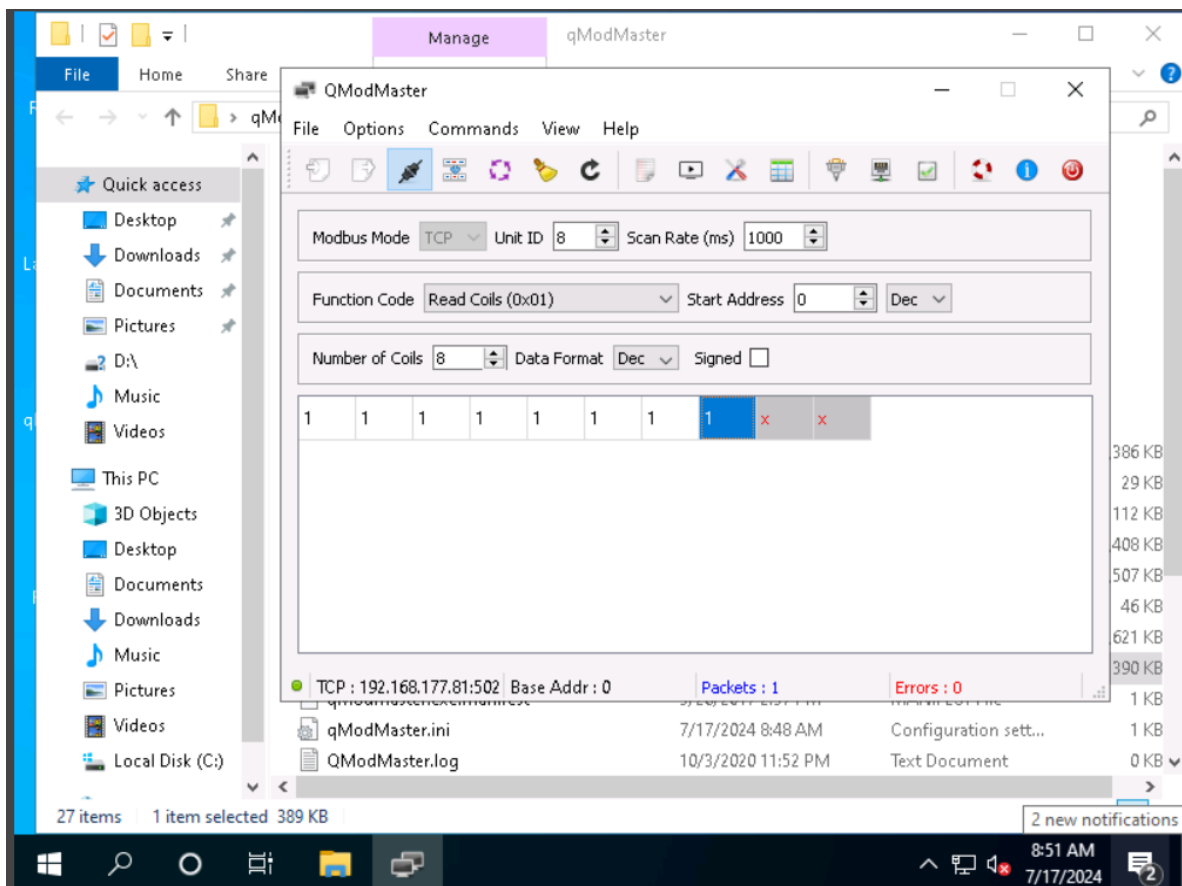
2.1 OUTPUT SCREENSHOTS

Exercise 2, Step 37: To verify the slave has started, open Command prompt and type **netstat -an | findstr 502** and **Enter**; the port should be listening, as shown in the screenshot. **Close** the Command Prompt window.



Exercise 2, Step 44:

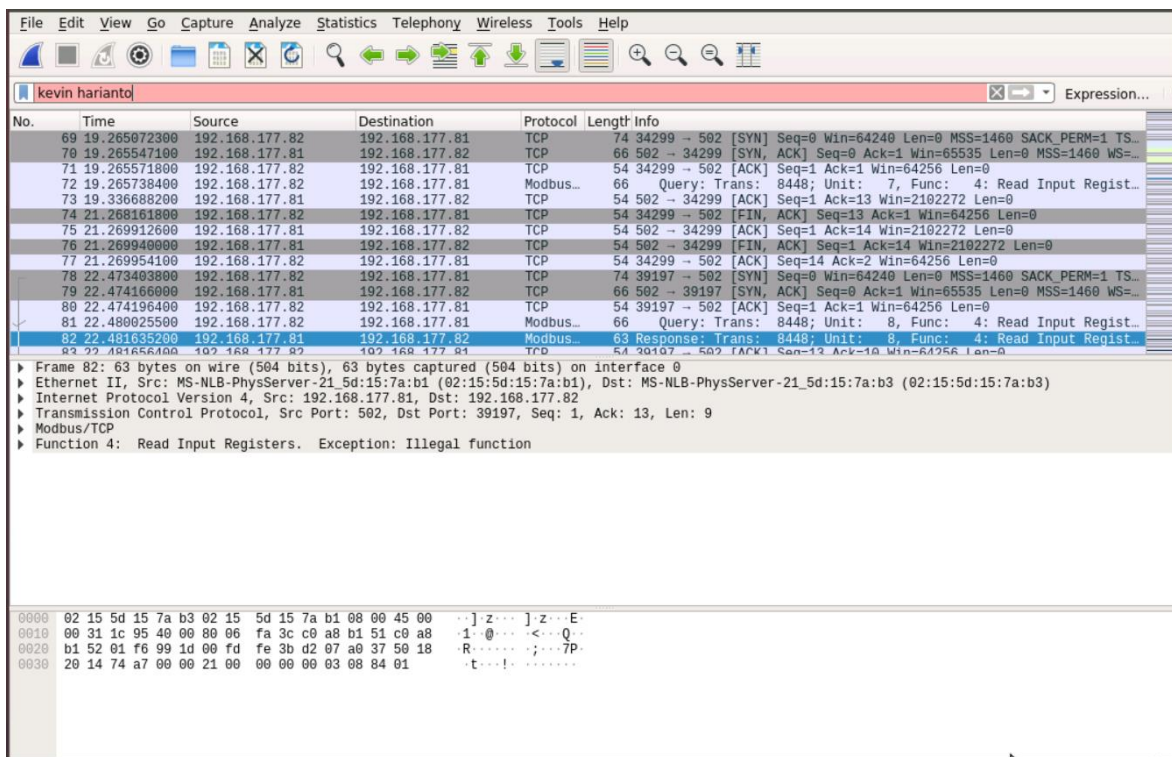
Switch to [SCADA Master](#). Click **Connect** icon to **Disconnect** the connection and then click **Connect** icon to **reconnect**. Click **Read / Write** icon next to the **Connect** icon. The master should be updated with the new values, as shown in the screenshot.



Exercise 2, Step 47: Once you have verified the machine, use the Nmap scripting engine script against it. Type **sudo nmap --script modbus-discover.nse 192.168.177.81 -p 502 -Pn**

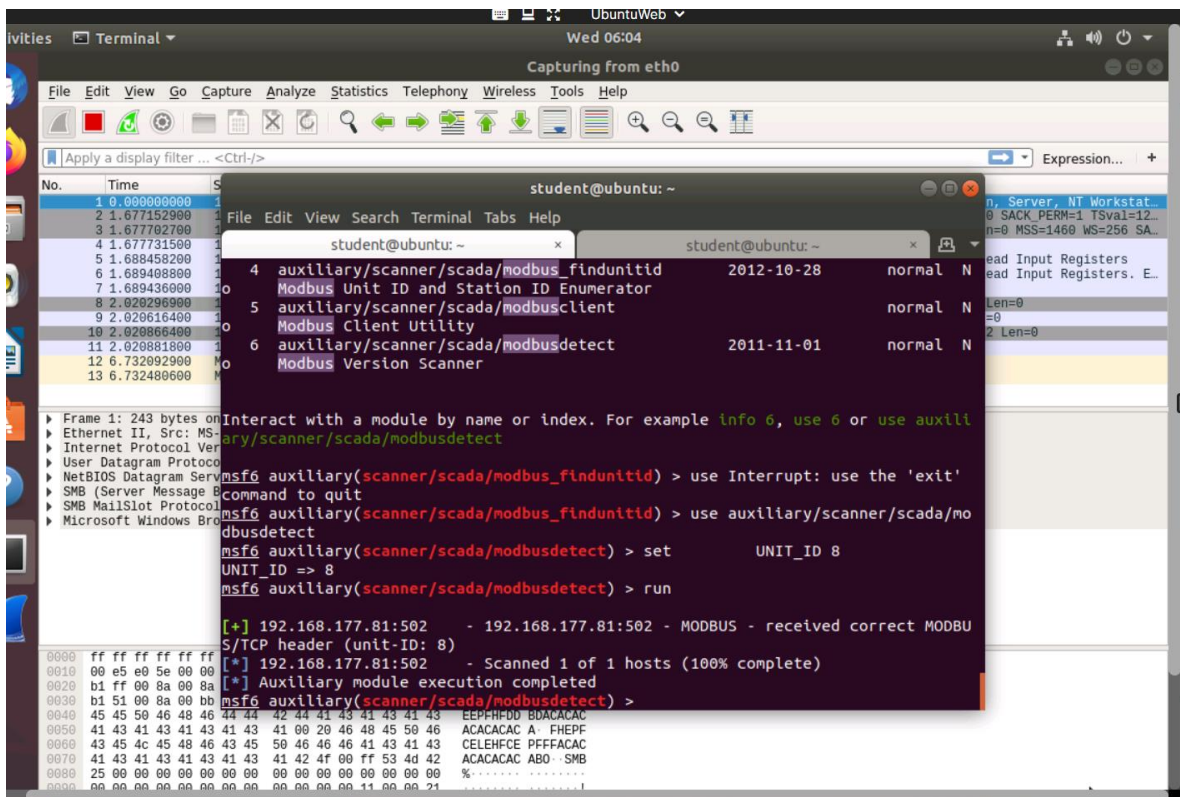
```
student@ubuntu:~$ echo "## Screenshot by Kevin Harianto 991602128 ['date +%F %T
']" ##
## Screenshot by Kevin Harianto 991602128 ['date +%F %T'] ##
student@ubuntu:~$
```

Exercise 2, Step 59: With the **UNIT-ID** identified, return to **Wireshark**; capture and review the data at the packet level. Look for when the ID is **correctly discovered**. **Stop** the capture to review.



Exercise 2, Step 60: **Restart** your capture in Wireshark.

In **msfconsole** press **Ctrl+C** to stop, and then return to the **modbusdetect** module by typing **use auxiliary/scanner/scada/modbusdetect** and press **Enter**, and type **set UNIT_ID 8** and press **Enter**. Type **run** and press **Enter** to perform the exploit.



2.2 QUESTIONS

Question 11.2.1

Use the Internet resources <https://whois.com>, <https://www.exploit-db.com>, and <https://archive.org> to extract information about SCADA networks. Flag submission is not required for this task; enter "No flag" as the answer.

No flag

Score

✓ Correct

Congratulations, you passed!

Your score: 2 / 2

Close Window