User Manual

Table of Contents

Introduction

Summary

The SCADA Home Automation System is a physical and interactive education model designed to emulate today's smart home systems. The purpose of the project is to introduce and generate interest in cybersecurity among students and young professionals through the use of an interactive model. By creating an interactive model, students and young professionals will be able to see the effects of cyber-attacks manifest in a physical manner. To accomplish the project objectives, a solenoid, stepper motor, and sensors will be integrated using Arduino microcontrollers running OpenPLC. In addition, a Human Machine Interface (HMI) will be used to control and monitor the SCADA Home Automation System. The audience will have the opportunity to interact with the system through physical interactions with the model and by performing cyber-attacks on the system.

Team Members

Jon Beason - Team Lead

Chad Bryan - CPE - Physical System Integration

Ben Curths - CPE - Physical System Integration

Simone Gbouomou - CBSY - Cybersecurity Development

Ben Calvert - CBSY - Cybersecurity Development

Ben Mcanulty - CBSY - Cybersecurity Development

Setting Up

Hardware Setup

Model Setup:

Plug the power cord into the pass-through port on the side of the model and then into a stable power outlet. Connect the ethernet cable to the ethernet pass-through port and to the attack laptop.

Software Setup

HMI Setup:

The Pi that is running ScadaBR will boot up when supplied power. The first thing you need to do is double click on the ScadaBR Chromium shortcut. Type in the url **localhost:8080/ScadaBR** if not already directed to the login screen.

This will open up the browser app for ScadaBR. If prompted for a login, the credentials are:
Username: admin
Password: admin

The next thing we need to do is click on the **“Graphical Views”** icon. There is a dropdown box that we need to change to **“Home_Tab”**. Now that the Home view is up, we need to fullscreen it by clicking the fullscreen button above the graphical view area. To make it completely fullscreen, go to the browser options (the vertical ellipsis) and click on the fullscreen icon beside the **“Zoom”** options.

Attack VM Setup:

Download the “Kali VM.zip” file, and unzip it. Open VirtualBox and click on the top left **File->Import Appliance**. Select the .ova file when prompted for a file location and then hit **Next** and **Finish** after that.

*Once the machine has imported, it is important to go to **Settings->Network** and change the name of the network that the VM is attached to to whatever your ethernet driver is.*

Now we can launch the VM and are good to move on.
Username: kali
Password: kali

Attacks/Mitigations

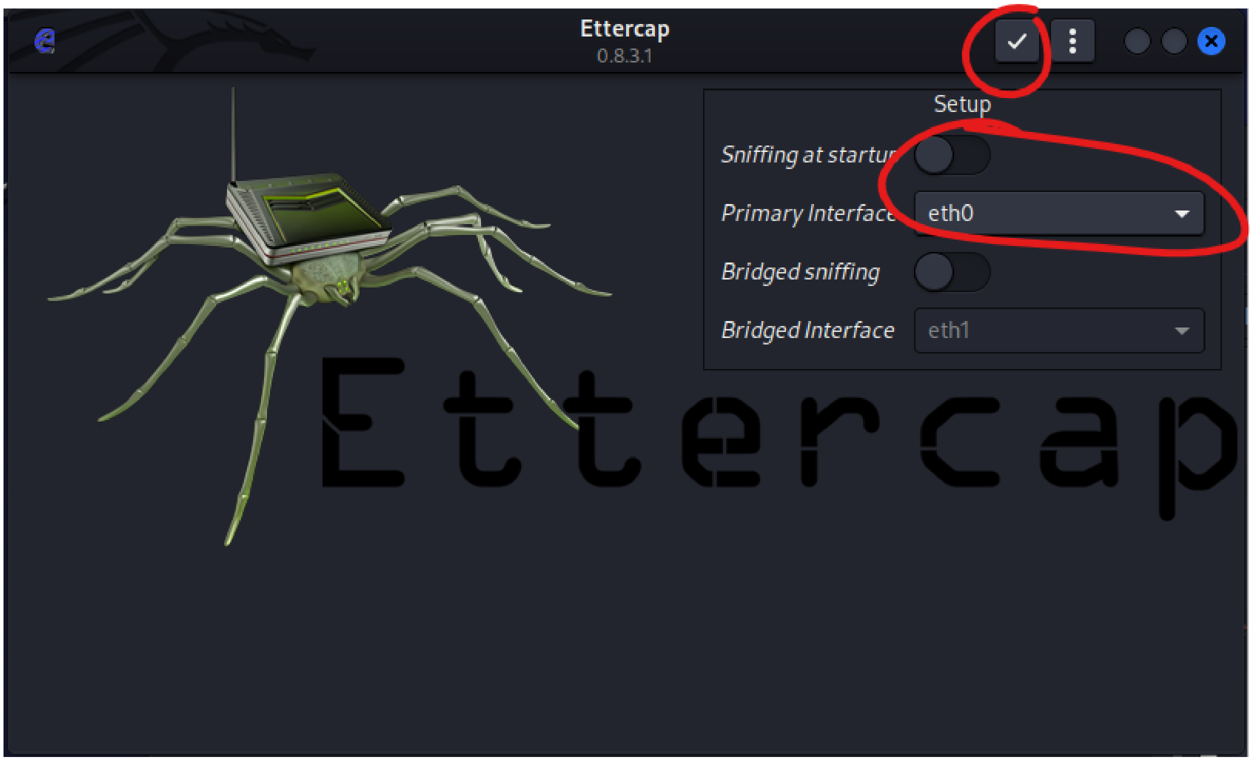
Launching Attacks

ARP Poisoning Attack:

There are two ways to launch the attack. There is a script that will do the commands for you. To do this option run ***./ArpPoisoning.sh*** in the terminal. This is less interactive and teachable for the audience, so we recommend using ettercap manually.

For a "manual" attack start by launching ettercap with the command ***sudo ettercap -G*** in the terminal. This will pull up the ettercap graphical user interface.

The next step is to uncheck the **"Sniffing at startup"** toggle and making sure the **"Primary Interface"** is set to ***eth0***. Click the checkmark at the top right to accept these settings and proceed.



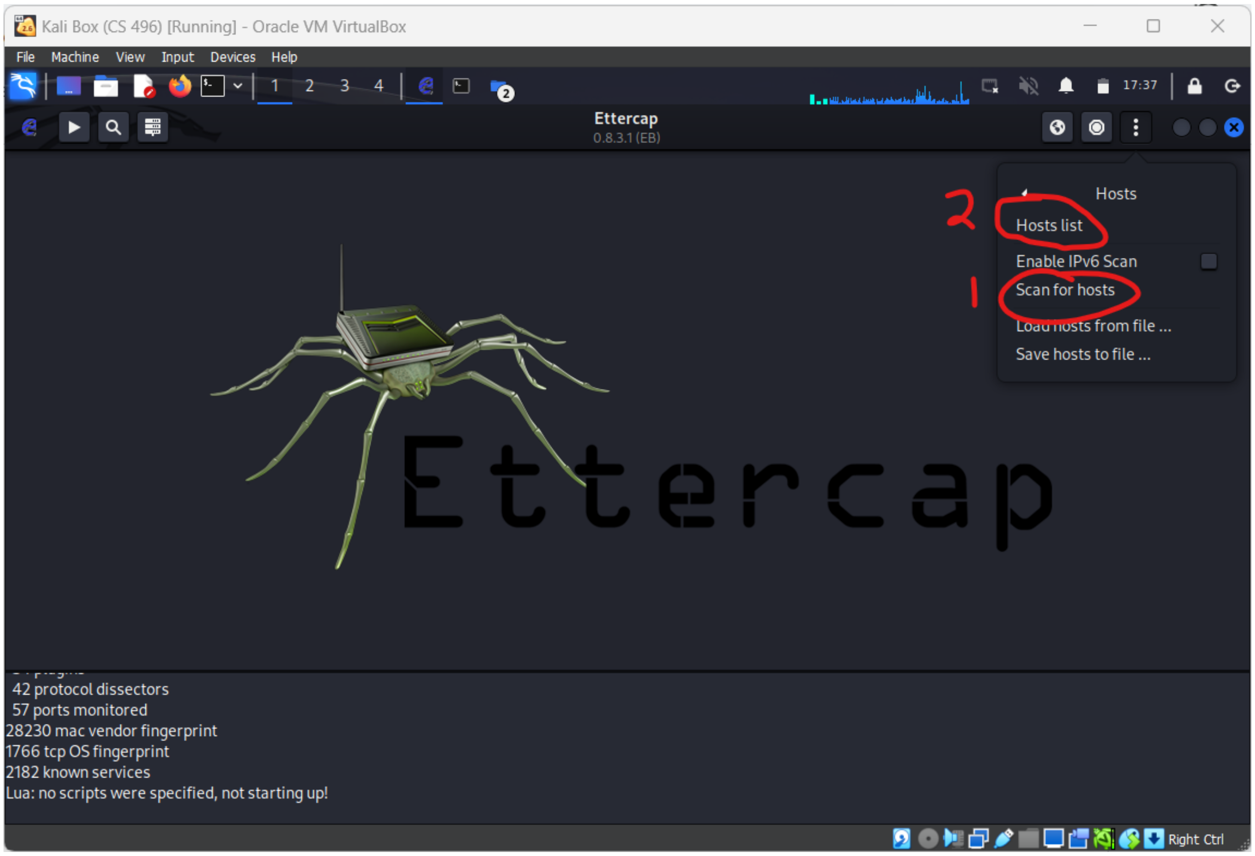
Sniffing at startup ○

Primary Interface: eth0

Bridged sniffing ○

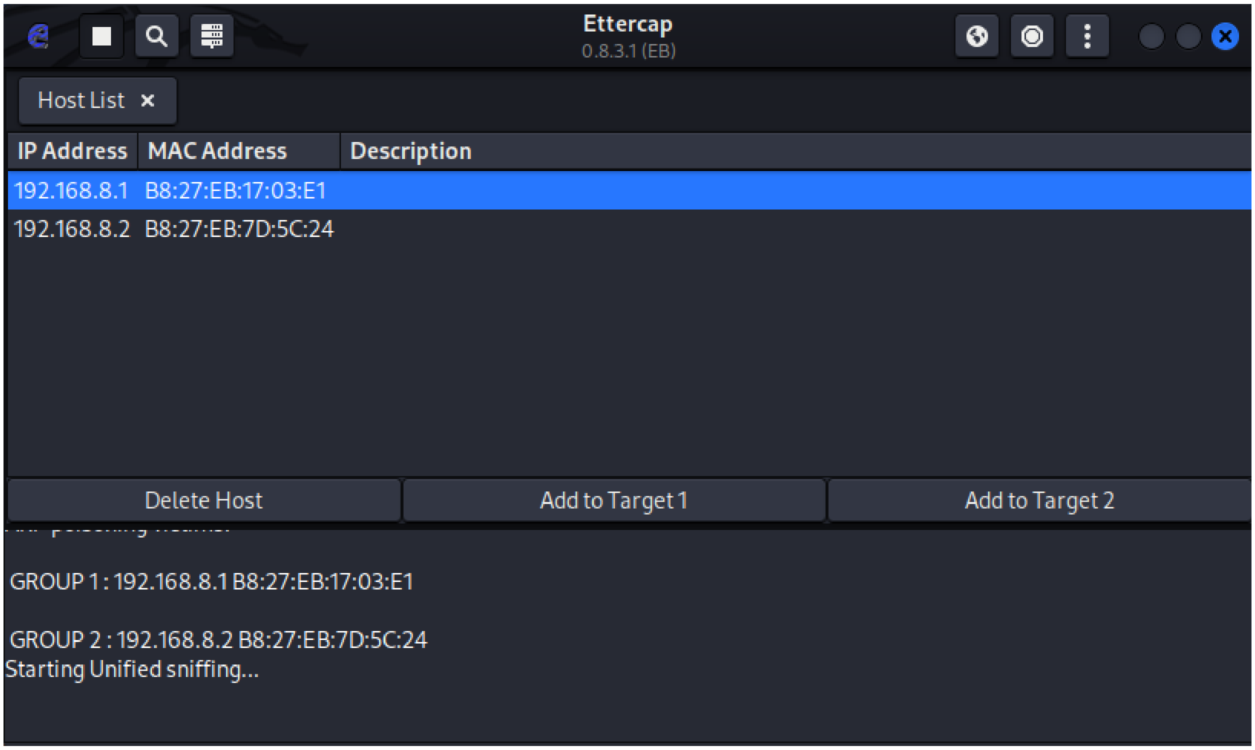
Bridged Interface: eth1

Next, we want to hit the vertical ellipsis on the top right, click on **"Hosts"**, and then click the **"Scan for hosts"** option, as shown in the screenshot below. This will scan the network for potential hosts. Then, click **"Hosts list"** to pull up the list of hosts that we just scanned for.



```
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
```

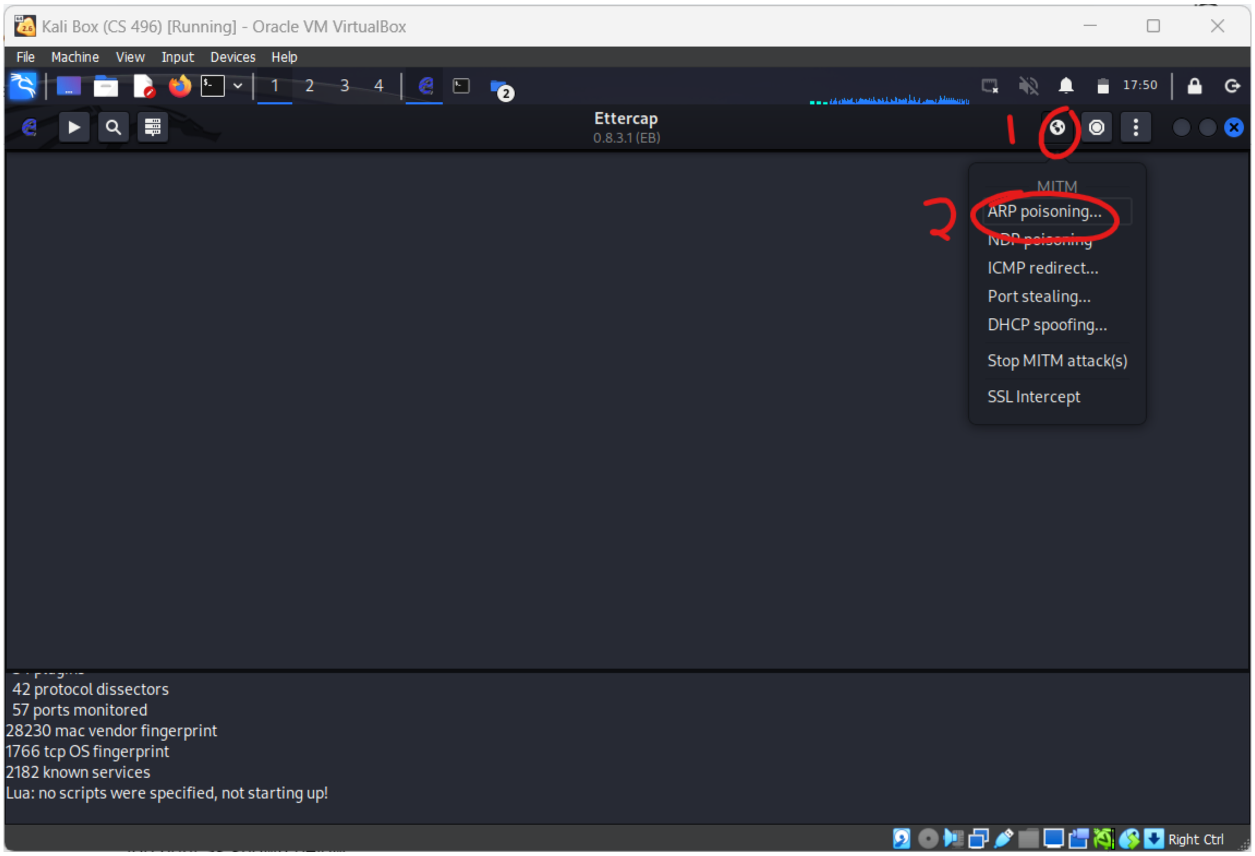
Now that we have the hosts list open, we need to click on **192.168.8.1** and set this as **“Target 1”**. We then need to set **192.168.8.2** as **“Target 2”**.



IP Address	MAC Address	Description
192.168.8.1	B8:27:EB:17:03:E1	
192.168.8.2	B8:27:EB:7D:5C:24	

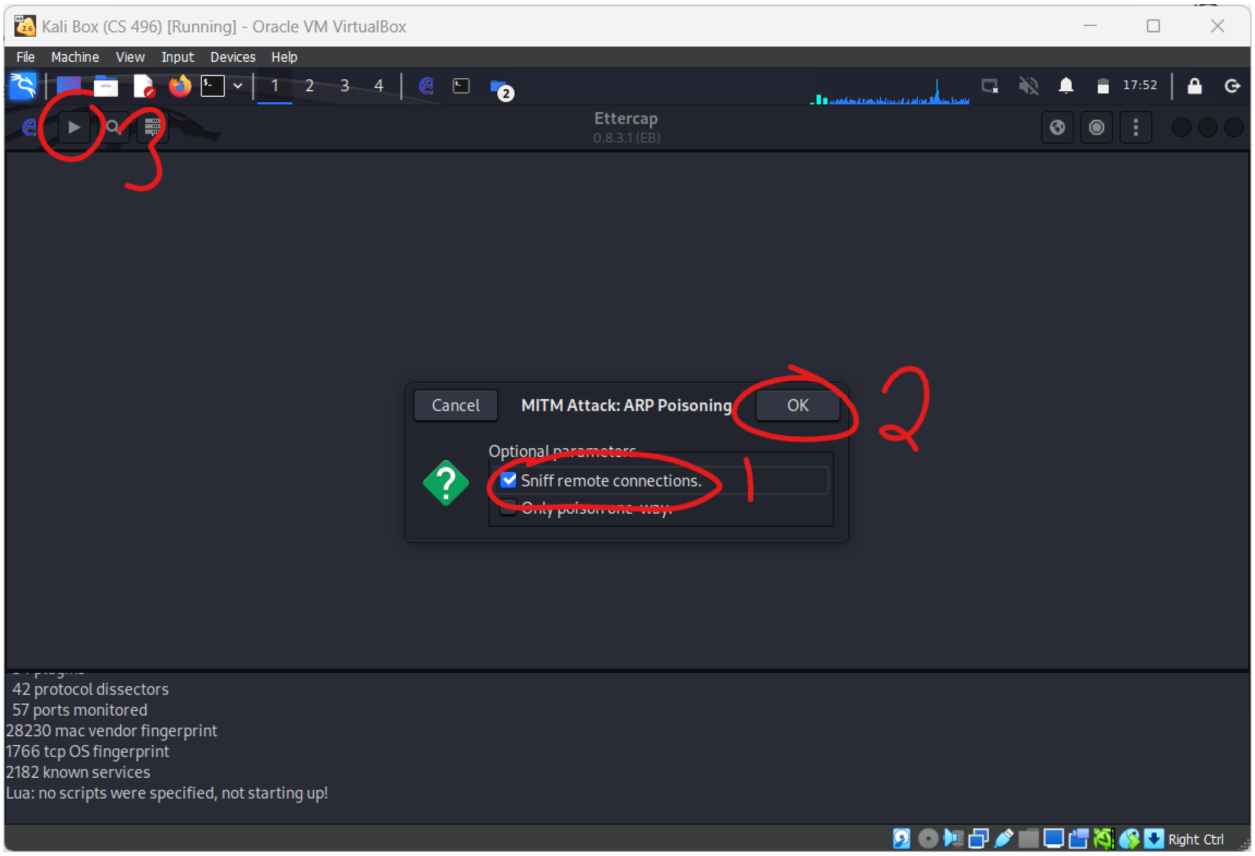
```
ARP poisoning victims:

 GROUP 1 : 192.168.8.1 B8:27:EB:17:03:E1

 GROUP 2 : 192.168.8.2 B8:27:EB:7D:5C:24
Starting Unified sniffing...
```

Now we want to set up our ARP Poisoning by clicking on the “**Globe” icon** on the top right. A drop-down menu will appear and we want to click on **“Arp Poisoning…”** as shown below.



```
 42 protocol dissectors
 57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
```

Here, we just want to make sure that **“Sniff remote connections”** is checked. Then, hit “OK” to close this tab. Now we can start sniffing by clicking on the **“Play” icon** on the top left corner.


Kali Box (CS 496) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
17:52
Ettercap
0.8.3.1 (EB)
Right Ctrl
1
2
3

Cancel **MITM Attack: ARP Poisoning** OK

Optional parameters

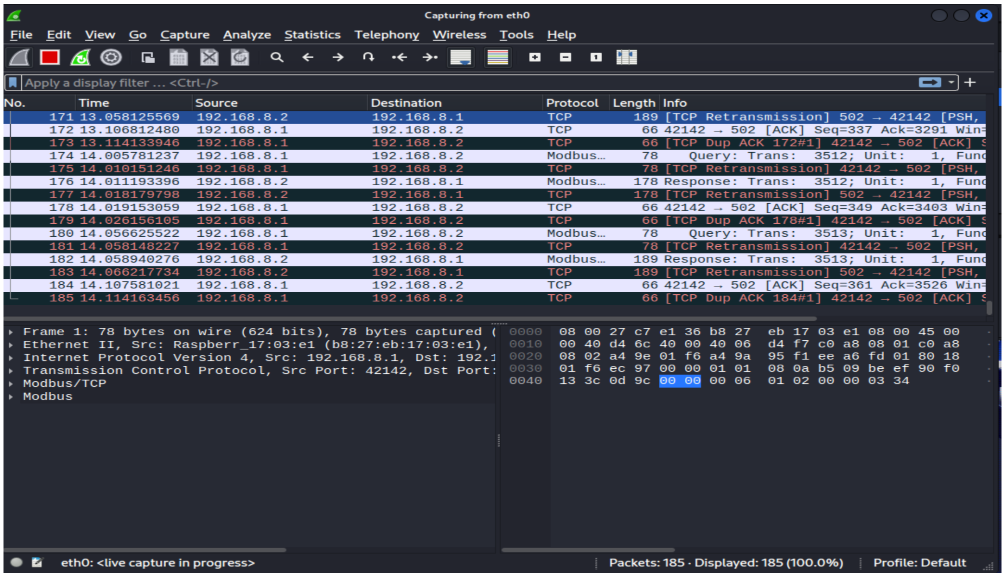
☑ Sniff remote connections.

□ Only poison one-way.

```
 42 protocol dissectors
 57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
```

Verifying ARP Poisoning:

The last step of the ARP Poisoning attack is to make sure that it worked, which will require opening up wireshark and checking if we can see the modbus packets being transferred to and from the PLC pi, as shown below.


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time          Source         Destination    Protocol  Length Info
  171 13.058125569  192.168.8.2    192.168.8.1    TCP       189 [TCP Retransmission] 502 → 42142 [PSH,
  172 13.106812480  192.168.8.1    192.168.8.2    TCP        66 42142 → 502 [ACK] Seq=337 Ack=3291 Win=
  173 13.114133946  192.168.8.1    192.168.8.2    TCP        66 [TCP Dup ACK 172#1] 42142 → 502 [ACK] S
  174 14.005781237  192.168.8.1    192.168.8.2    Modbus…    78    Query: Trans:  3512; Unit:   1, Func
  175 14.010151246  192.168.8.1    192.168.8.2    TCP        78 [TCP Retransmission] 42142 → 502 [PSH,
  176 14.011193396  192.168.8.2    192.168.8.1    Modbus…   178 Response: Trans:  3512; Unit:   1, Func
  177 14.018179798  192.168.8.2    192.168.8.1    TCP       178 [TCP Retransmission] 502 → 42142 [PSH,
  178 14.019153059  192.168.8.1    192.168.8.2    TCP        66 42142 → 502 [ACK] Seq=349 Ack=3403 Win=
  179 14.026156105  192.168.8.1    192.168.8.2    TCP        66 [TCP Dup ACK 178#1] 42142 → 502 [ACK] S
  180 14.056625522  192.168.8.1    192.168.8.2    Modbus…    78    Query: Trans:  3513; Unit:   1, Func
  181 14.058148227  192.168.8.1    192.168.8.2    TCP        78 [TCP Retransmission] 42142 → 502 [PSH,
  182 14.058940276  192.168.8.2    192.168.8.1    Modbus…   189 Response: Trans:  3513; Unit:   1, Func
  183 14.066217734  192.168.8.2    192.168.8.1    TCP       189 [TCP Retransmission] 502 → 42142 [PSH,
  184 14.107581021  192.168.8.1    192.168.8.2    TCP        66 42142 → 502 [ACK] Seq=361 Ack=3526 Win=
  185 14.114163456  192.168.8.1    192.168.8.2    TCP        66 [TCP Dup ACK 184#1] 42142 → 502 [ACK] S
```

```
▸ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (      0000  08 00 27 c7 e1 36 b8 27  eb 17 03 e1 08 00 45 00
▸ Ethernet II, Src: Raspberr_17:03:e1 (b8:27:eb:17:03:e1),      0010  00 40 d4 6c 40 00 40 06  d4 f7 c0 a8 08 01 c0 a8
▸ Internet Protocol Version 4, Src: 192.168.8.1, Dst: 192.1     0020  08 02 a4 9e 01 f6 a4 9a  95 f1 ee a6 fd 01 80 18
▸ Transmission Control Protocol, Src Port: 42142, Dst Port:     0030  01 f6 ec 97 00 00 01 01  08 0a b5 09 be ef 90 f0
▸ Modbus/TCP                                                    0040  13 3c 0d 9c 00 00 00 06  01 02 00 00 03 34
▸ Modbus
```

```
eth0: <live capture in progress>          Packets: 185 · Displayed: 185 (100.0%)    Profile: Default
```

Injection Attack:

Note: To be able to launch the injection attack, we need to have the ARP tables poisoned, so make sure the ARP Poisoning attack has been launched.

To launch the Injection attack, we just need to navigate to the "**Mitigations**" folder on the desktop of the vm by opening a terminal and typing ***cd Desktop/Mitigations***. Now that we are in the folder, we can launch the attacks. **The attacks must be done in sequence of floors 1, 2, 3, and then there is a cleanup script that must be run after all the attacks, as it resets everything that was changed for a next demonstration.**

To run the scripts, run the command ***python3 Injection_Floor_1.py*** for the floor 1 attack. This will open the garage door, unlock the front door, and allow the users to press the green button to move to the next floor, while disabling the alarm from going off. Next, run ***python3 Injection_Floor_2.py*** which will disable the magnetic sensor on the side panel and disable the motion sensor, allowing the user to press the floor 2 green button. Then, run ***python3 Injection_Floor_3.py*** which will disable the IR sensor on that floor and allow the user to press the final green button. Upon pressing all the green buttons without triggering the alarm, the attacks are considered successful.

To reset everything to restore the system to full functionality and allow the mitigations to work, we need to run ***python3 Injection_Cleanup.py***.

Launching Mitigations

Both Mitigations can be launched the same folder ***Desktop/Mitigations/***. The mitigations should be run in the order of injection mitigation first, then arp mitigation.

The Injection mitigation can be run by being in the correct filepath (see beginning of Injection Attack section if not sure how to get there) and then running the command ***./InjectionMitigation***. If prompted for a password for either of the machines, they match the username of the machine requesting (openplc@raspberrypi is openplc, scadabr@raspberrypi is scadabr).

The Arp poisoning mitigation is run in the same way as the injection mitigation. Simply run the command ***./ArpMitigation.sh*** and if prompted for a password, supply it accordingly.

To undo the mitigations, navigate to the same Mitigations folder on the Desktop and run the commands ***./InjectionUnMitigate.sh*** and ***./ArpUnMitigate.sh***.

Troubleshooting

The Raspberry Pi's should auto-boot and auto-login. If there are things that are not working, refer to other documentation in the project folder. To access the Pi's remotely,

SSH into the Raspberry Pi 1 (Over Eth: 192.168.8.1 /// Over Student 5 WIFI: 10.4.152.2).
Username: scadabr
Password: scadabr

SSH into the Raspberry Pi 2 (Over Eth: 192.168.8.2 /// Over Student 5 WIFI: 10.4.176.212).
Username: openplc
Password: openplc