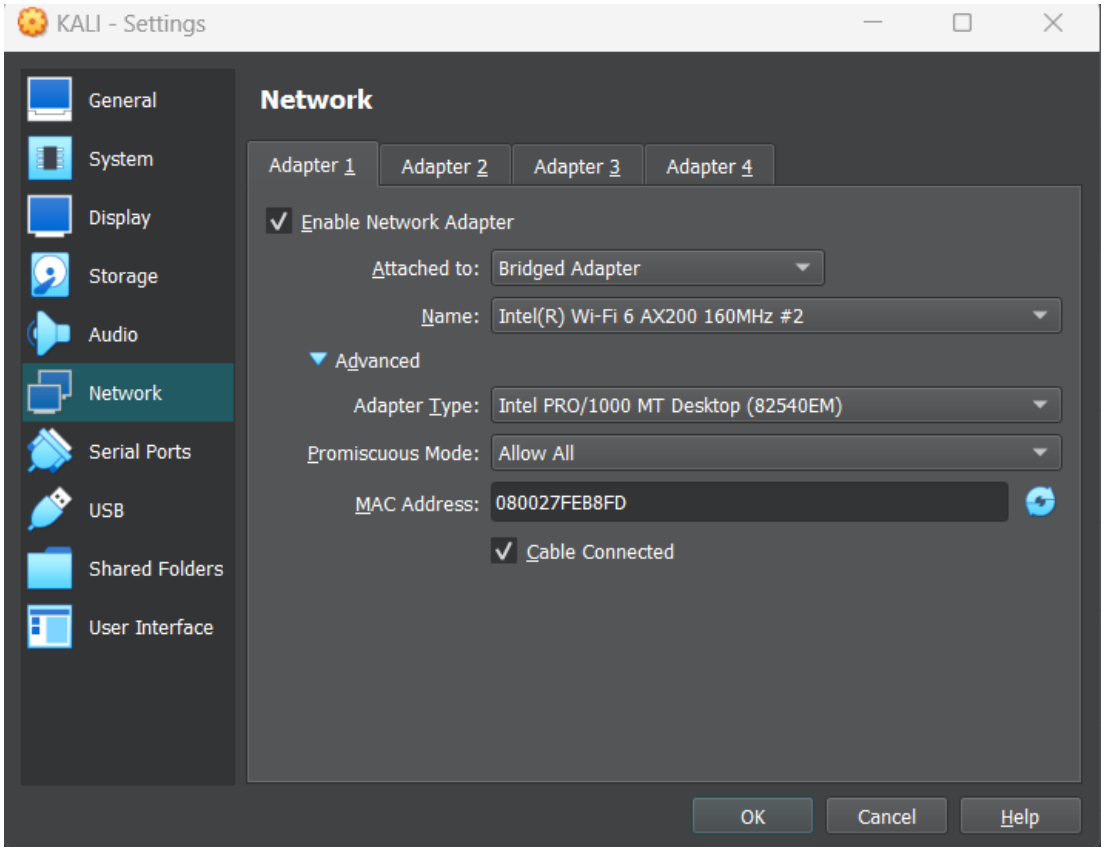
Matthew Townsend

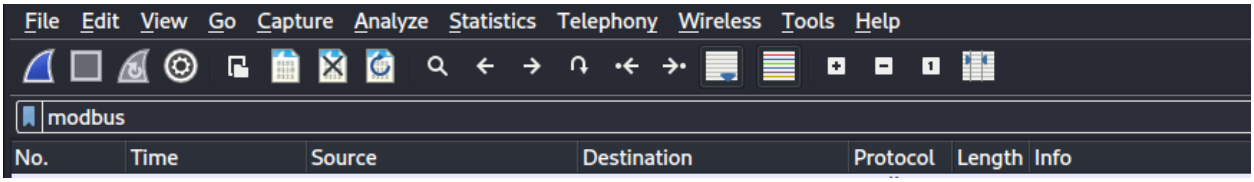
4/11/23

KALI payload delivery

In this lab we displayed that we could read the packets using Wireshark and then send a fake packet using data from legitimate packets to control our circuit. To begin this lab requires the environment from previous labs, Scadabr, openplc, a breadboard circuit and a pi 400. Next will be downloading and configuring a kali VM in Virtual Box. Next is to set both VMs (ScadaBR and kali) network settings to promiscuous mode to Allow All.

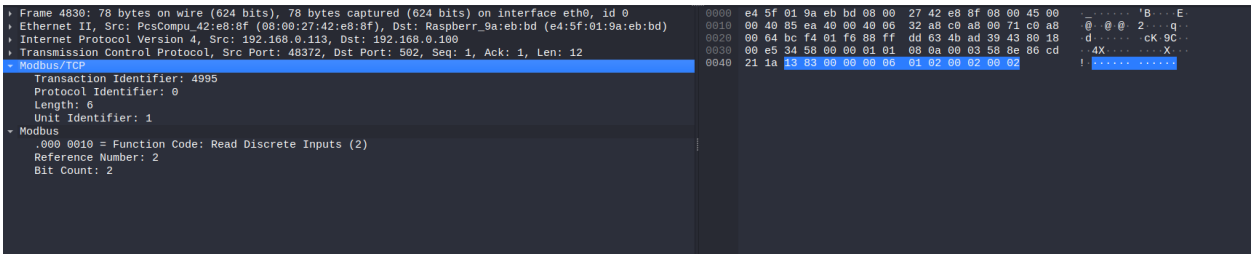


Setting promiscuous mode will allow the VMs to detect each other's traffic which is needed for the rest of the lab. Next begin running the environment from either the first circuit or second with the HMI working and updating with the circuit. Next we will open Wireshark on the Kali VM and sort traffic using Modbus

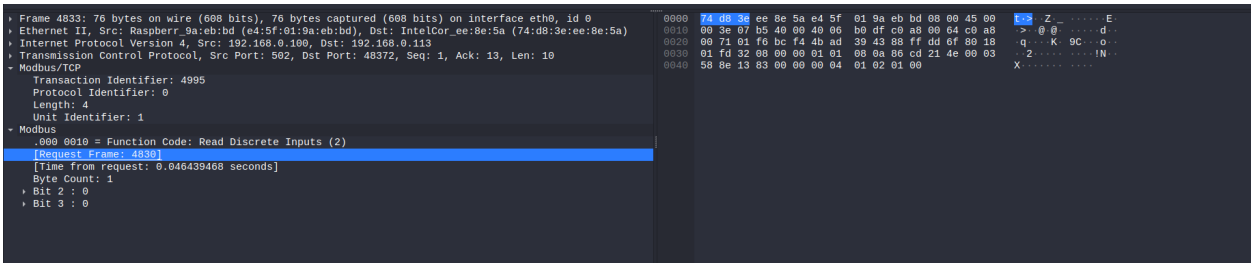


Entering Modbus will sort the results and only returning the packets that are using the Modbus protocol. This will allow us to view only the traffic we need coming ScadaBr and the response from openplc. These messages will look like this

Request:



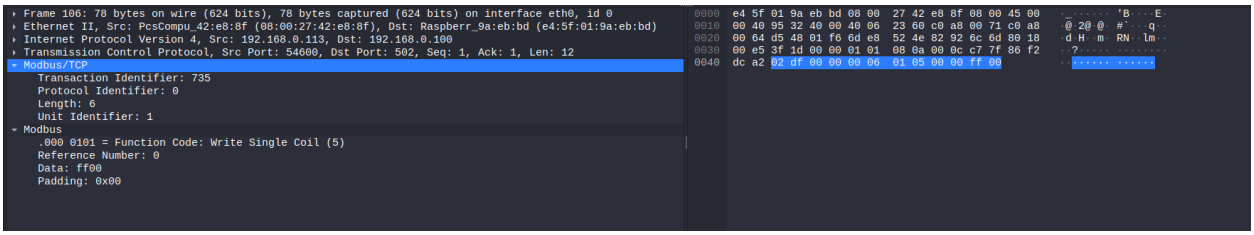
Response:



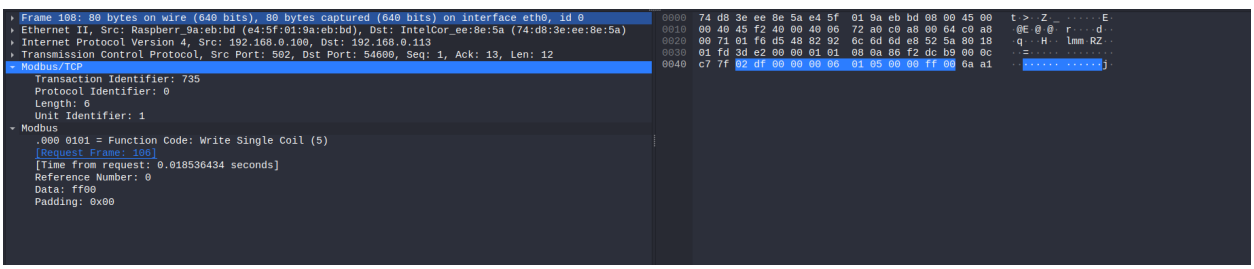
The messages come in pairs and there is different requests, read coils, discreate inputs, and write single coil. For this lab we need the write single coil messages, they look like this

|----------| data needed

1



2



These packets contain the data we need to exploit the HMI. The write command in this instance is the manual changing of values from 0 to 1 turning the light on and then a second set of packets turning it off. Here the data wee need to take to replicate this is highlighted on the right side of the first image.

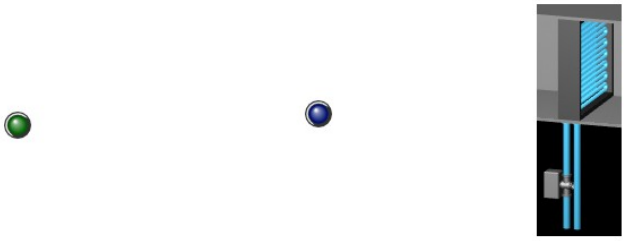
Before the Command

The Command

```
┌──(matt-kali㉿kali)-[~]
└─$ echo -n -e "\x02\xdf\x00\x00\x00\x06\x01\x05\x00\x00\xff\x00" | nc 192.16
8.0.100 502 -w 1 # writing 'on' to slave 1,
◆◆
```

The command is split into two parts the payload and the delivery method. The payload is the string of hexadecimal taken from the write ‘on’ packet and it is being sent using Netcat. The delivery is the nc command that delivers the previous echo statement to the address 192.168.0.100 on port 502. The HMI uses port 502 as seen in the packets.

After Command



The off command

```
┌──(matt-kali㉿kali)-[~]
└─$ echo -n -e "\x02\xf4\x00\x00\x00\x06\x01\x05\x00\x00\x00\x00" | nc 192.16
8.0.100 502 -w 1 # writing 'off' to slave 1,
◆
```

The “Off Command” is the same command as before just with the hex from the write packet that changed the lights value from 1 to 0. The write ’off’ packet is nearly identical except for the Modbus portion.

After command

We remotely activated a circuit with only the collection of a few select packets. This shows the dangers of having unsecured devices and networks in infrastructure environments, there is nothing to prevent or detect this action out of the box.