Threat Detective: Cyber Physical Protection

Team Members: Issac Navarro(UCSD), Sebastian Magana (UCSB), Greg Norton(Naval Post Graduate), Alessandro Guaresti(CSUCI) /
EXWC Project Lead: Jorge Lacoste NAVFAC EXWC Public Works Department
Graduate Student Mentor: Esther Showalter UCSB

Abstract:

The Navy faces cyber threats from adversaries who grow more capable and persistent every day. The stakes are high: if an attacker compromises a critical control system, the consequences could be millions of dollars in damage, injury, or even loss of life. In order to develop defenses against these attacks we developed a rapid testing network security platform. The platform has three components: simulation, data collection, and threat detection. After selecting a system to test, simulation is accomplished by constructing a threat detection model using Node-Red software. Next, realistic network traffic can be collected and logged in a database using the ELK stack. Our demonstration showcases a model communicating by Modbus protocol, which is a legacy networking protocol that is still widely used despite glaring security vulnerabilities. We collect and analyze network traffic generated from simulated attacks. Our example uses the TCP SYN flood denial of service attack. The last step is threat detection, which we plan to accomplish via machine learning using the Amazon SageMaker service. By training the machine learning agent using network traffic generated from the model, we can establish a baseline for normal network activity, and raise alarms when anomalies are detected.

Project Goals:

The main goal of this project was to create a small scale virtual environment to conduct modeling and simulation of Industrial Control Systems (ICS) and to leverage this environment as a framework to advance our understanding of how machine learning can be used to baseline control system networks and detect potential malicious network traffic.

Impact:

The impact for the Navy is that our work can be leveraged to advance the understanding of how to conduct modeling and simulation and how to deploy machine learning using small datasets.

Industrial Control System Overview:

ICSs are used to automate repetitive tasks; they are widely used in industry and in government facilities. They are used in a variety of environments including vehicle assembly lines, water treatment plants, the electrical grid, and onboard naval vessels where they may be responsible for engine control.

Modbus Overview:

Modbus is the oldest and most popular communications protocol for supervisory control and data acquisition (SCADA), it has been the de facto standard since 1979. Modbus relies on a master/slave relationship between the hosts. Only one device (the master) can initiate requests (queries). The other devices (the slaves) must respond with the requested information or take the requested action. Modbus can be used for peer-to-peer connections. Modbus supports both the point-to-point or multidrop network topologies. Modbus was built without any security considerations.

Acknowledgements:

Thank you to our mentors for their support, and Naval Facilities Engineering and Expeditionary Warfare Center for providing funding for this project.

Methods:

Initially, we generated data using the following hardware components:

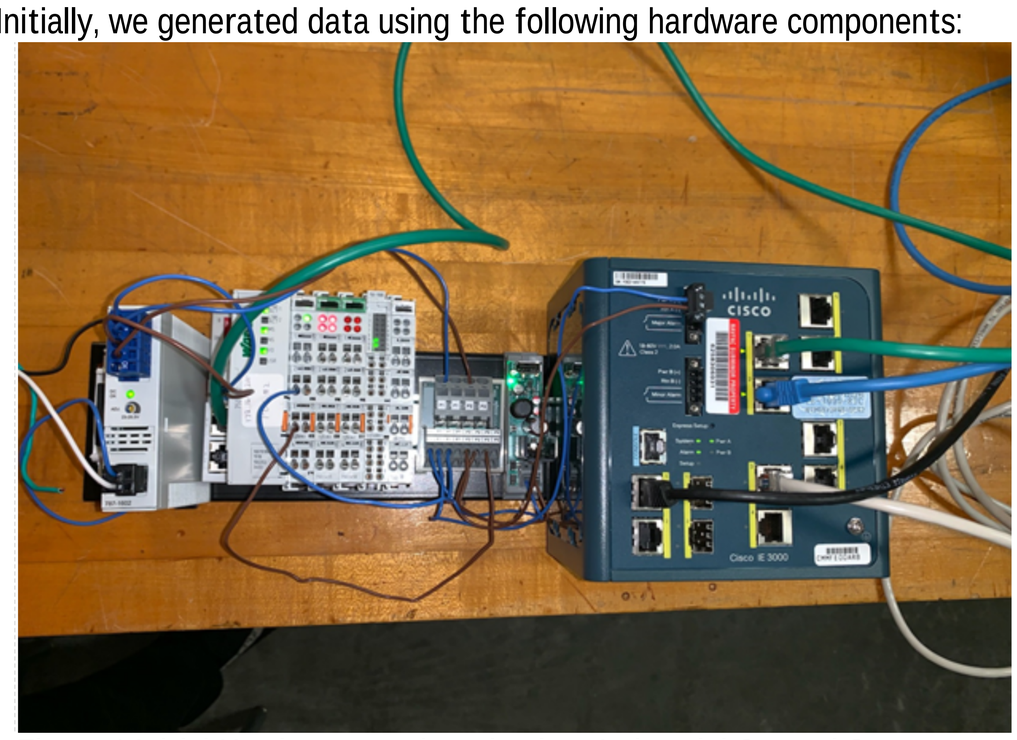

Figure 1 Initial Hardware used to simulate Industrial Control System

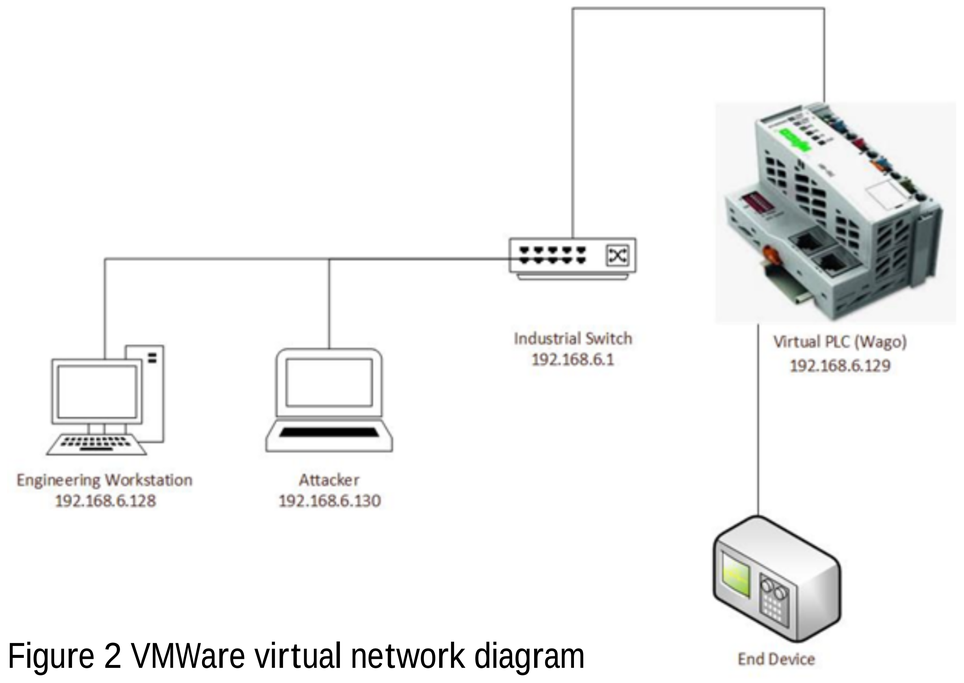

Figure 2 VMWare virtual network diagram

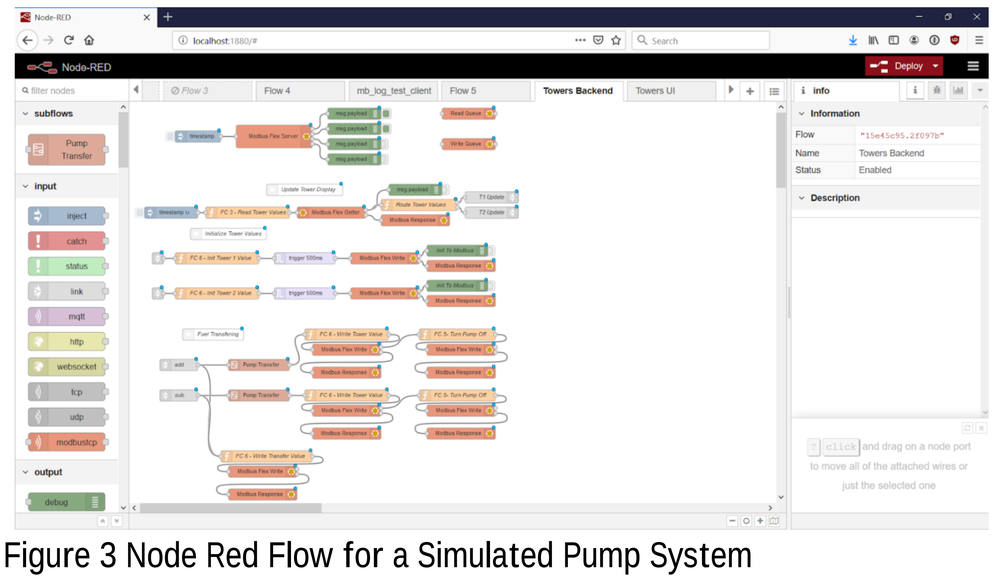

Figure 3 Node Red Flow for a Simulated Pump System

Results and Future Work:

Running our demonstration Node-RED flows on the virtual network yielded the expected logs. We hope to use this data to train a machine learning agent to recognize when the network is under attack. By establishing baseline network behavior, the machine learning agent will raise an alarm when an anomaly is detected, and can trigger the appropriate action such as alerting the network administrator or setting a firewall rule that will take the attacker off the network immediately. Amazon SageMaker is being considered as a method of performing machine learning on the data generated.

```
File  Edit  Search  Options  Help
INFO [msg.payload] 1565328099749 192.168.6.128 192.168.6.129 00310000000601050001ff00 64 no_attack
INFO [msg.payload] 1565328099749 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328099750 192.168.6.128 192.168.6.129 00320000000601010001 0001 64 no_attack
INFO [msg.payload] 1565328099750 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328102820 192.168.6.128 192.168.6.129 00330000000601050001ff00 64 no_attack
INFO [msg.payload] 1565328102821 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328102822 192.168.6.128 192.168.6.129 00340000000601010001 0001 64 no_attack
INFO [msg.payload] 1565328102822 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328105893 192.168.6.128 192.168.6.129 00350000000601050001ff00 64 no_attack
INFO [msg.payload] 1565328105894 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328105895 192.168.6.128 192.168.6.129 00360000000601010001 0001 64 no_attack
INFO [msg.payload] 1565328105895 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328108966 192.168.6.128 192.168.6.129 00370000000601050001ff00 64 no_attack
INFO [msg.payload] 1565328108966 192.168.6.128 192.168.6.129 null 52 no_attack
INFO [msg.payload] 1565328108967 192.168.6.128 192.168.6.129 00380000000601010001 0001 64 no_attack
INFO [msg.payload] 1565328108967 192.168.6.128 192.168.6.129 null 52 no_attack
```

Figure 4 Logs generated under Scenario A: Normal Network Behavior