ICS Network Traffic

1 Overview

This exercise uses the GrassMarlin network traffic tool to observe network activity that you generate while working with a simple PLC device. GrassMarlin is a software tool released by the NSA that provides a method for discovering and cataloging Supervisory Control & Data Acquisition (SCADA) and Industrial Control System (ICS) hosts on IP-based networks.

This exercise assumes that student has performed the `softplc2` lab. Preliminary steps taken by the student are similar to those from that lab.

2 Lab Environment

This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer plc-traffic
```

A link to this lab manual will be displayed, along with a link to the GrassMarlin user guide.

3 Lab topology

The lab includes four components as shown in Figure 1:

- **Slave Device** A simple slave device containing two buttons and a lamp. The device has an ethernet interface via which it is connected to a PLC using Modbus TCP/IP.
- **PLC** A programable logic controller implemented using the OpenPLC Soft-PLC implementation on a Linux based computer. The PLC is connected to the slave device via Modbus TCP/IP over ethernet. The PLC has a seperate ethernet connection to an HMI computer. For convenience of the lab, the PLC computer includes Wireshark and will be used to capture, review and replay network traffic between the PLC and the slave device.
- **HMI** A computer that interacts with the PLC using a web browser to load programs and monitor its operation. This computer also contains an OpenPLC Editor for constructing programs using "Ladder Logic" (LD).
- **NETMON** A device that has taps on both of the networks, and collects PCAPs from those networks into files within its `/taps` directory.

The PLC and the slave device communicate via Modbus TCP/IP, with the PLC acting as the master. The HMI component in this topology does not use Modbus, rather it simply interacts with the PLC via the PLC's web server.

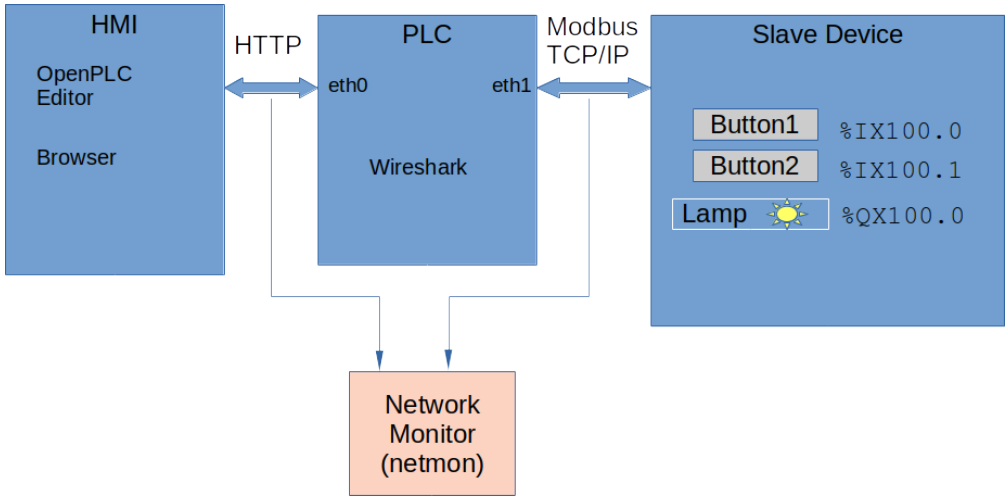

Figure 1: Soft PLC Lab Topology

4 Tasks

In this lab, you will upload a simple "hello world" program from the HMI to the PLC and use it to control the slave device. You will use GrassMarlin to view the resulting traffic.

4.1 Explore

Find the `Physical Board` window. This is your virtual slave device. Click the two buttons and confirm nothing happens. This board is connected to the PLC via Modbus TCP/IP, however the PLC is not yet running and is not yet programmed.

4.2 Upload and run PLC program

In this task, you will deploy an existing "hello world" program onto the PLC and demonstrate its use.

4.3 Start the PLC user interface & Explore

On the HMI terminal, use the firefox browser to access the PLC using port 8080:

```
firefox plc:8080 &
```

Login to the PLC through the browser with `openplc` as the username and password. You should now see the Open PLC web page that controls the PLC. Click the `Slave Devices` button on the left. And double click on the `myserver entry`. This is the slave device that your PLC program will control. Note we define only 3 bits to control the slave device. Two bits as *Descrete Inputs* for two buttons (only one of which is initially used), an one bit for the lamp as *Coils*. Make a note of the slave device IP address and the port number used to communicate using Modbus.

4.3.1 Upload the ST program to the PLC

Select `Programs` on the left pane and use the `Browse` button to locate the `hello.st` file. Then upload it to the PLC. Provide a `Name` in the resulting form and press the `Upload program` button. Note the program is compiled on the PLC and results are displayed in the browser. Then click the `Go to Dashboard` button.

The PLC now has your `hello world` program loaded, but the PLC is not yet running. Click the `Start PLC` button. Once the PLC is started, click the Monitoring button.

4.3.2 Test the PLC program

On the Physical Board, click button1. The light bulb should come on for a few seconds. Observe the indicators on the browser Monitoring page.

Stop the PLC and close the browser.

4.4 View traffic

You will now view the traffic that your activity has generated. The `netmon` component has taps into both of your networks, and collects traffic into pcap files stored in the `/taps` directory.

These steps are performed from the `netmon` computer.

- Start the program with the command: `grassmarlin`. If the resulting GUI is not readable (i.e., is a black or blank window), close it and restart the program.
- From the "File" menu, select "Import Files" and use the "Add Files" button and the file browser to add the two PCAP files found in `/taps`. Then use the "Import Selected" button on the "Import" dialog to import those files. It may take a minute to complete the import.
- When the import completes, close the "Import" dialog.

You should then see the "Logical Graph" of system components that the GrassMarlin tool discovered from the PCAP files. If the component icons overlap, use the mouse to seperate them. You should now be able to correlate the computers in the lab network with elements within the logical graph. Note that Logical Graph is organized by subnets, and not physical devices. For example, it lacks information necessary to associate the two IP addresses of the PLC as belonging to the same device. Consider why that is.

On the left hand panel, expand the node lists to locate summaries of packet traffic sent and received by each node. Right click on one having substantial traffic, and select "View Frames" to view the captured packets. If you find interesting traffic, you can right click on a frame and select "Open in Wireshark".

Return to the GrassMarlin window and right click on components and select "View Details" to learn what the tool has discovered about each component.

The tool uses textitfingerprinting to make guesses about the products and software associated with each node on the network. Use the `Tools / Fingerprint Manager` to view the fingerprint criteria. Refer to the GrassMarlin user guide for information about how fingerprint criteria are defined.

4.4.1 Remove poor fingerprint

View the details of the HMI device in the logical graph. You will notice that GrassMarlin has fingerprinted it as potentially being two products from a major US manufacturer of industrial equipment. Go to the Fingerprint Manager and disable only those fingerprint entries that appear in the HMI device `View Details`.

Note the tool is not kind enough to present information in alphabetical order, or any other useful arrangement. So you will have to perform a search for the two items of interest. Use `File/disable` to disable the two poorly expressed fingerprints. (Do not just use `Disable All`).

After you have disabled the fingerprints, close the fingerprint manager and use `File / New session` in the main window to create a new session (no need to save the old session). Then import the PCAP files again. View the details of the HMI device and confirm that the false fingerprint information is no longer associated with the device.

The use `File / Save as` and save the session in

```
/home/ubuntu/GrassMarlin/session1.gm3
```

Save the file at the above path, not in the default directory!

Minimize the GrassMarlin application.

4.5 Connect to internet

Note how all the network traffic was confined to the three active components on the network. However, a helpful sysadmin has connected a gateway component to the HMI network. [1] Go to the HMI windows and add this gateways using:

```
sudo route add default gw 172.25.0.101
```

Then start the Firefox browser (`firefox plc:8080`. Do not do anything in the browser, just start it. Then,

- Return to the GrassMarlin application and use `File / New Session` to open a new session.
- Import the two PCAP files again from `/taps/`.
- Close the import window and observe the revised logical graph. Look who you just invited onto your ICS network.
- Save this session in

  ```
  /home/ubuntu/GrassMarlin/session2.gm3
  ```

5 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under sponsorship from the DoD CySP program. This work is in the public domain, and cannot be copyrighted.

[1] Skip this section if your installation is isolated from the internet.