Implementing OpenPLCs into a Cyber Defense Competition

DESIGN DOCUMENT

Team 16

Dr. Doug Jacobson
Dr. Julie Rursch

Nick Springer - Security Engineer
Matthew McGill - Project Manager
Val Chapman - Software Testing Engineer
Josh Przybyszewski - Software Engineer
Joseph Young - Software Engineer
Liam Briggs - Hardware Engineer
Brennan Ferguson - Hardware Engineer

sdmay18-16@iastate.edu
http://sdmay18-16.sd.ece.iastate.edu

Revised: 10/13/2017

Table of Contents

List of figures/tables/symbols/definitions 2

1 Introduction (Same as project plan) 3

1.1 Acknowledgement 3

1.2 Problem and Project Statement 3

1.3 Operational Environment 3

1.4 Intended Users and uses 4

1.5 Assumptions and Limitations 4

1.6 Expected End Product and Deliverables 4

2. Specifications and Analysis 5

2.1 Proposed Design 5

2.2 Design Analysis 5

3. Testing and Implementation 6

3.1 Interface Specifications 6

3.2 Hardware and software 6

3.3 Process 6

3.4 Results 6

4 Closing Material 7

4.1 Conclusion 7

4.2 References 7

4.3 Appendices 7

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

We would like to acknowledge the assistance provided to us by Doug Jacobson and Julie Rursch throughout the duration of our project. We are extremely grateful for the time, help and equipment provided to our entire team.

1.2 Problem and Project Statement

At core of our project, is a desire to create a system to simulate a Cyber Physical hardware with the OpenPLC platform. The CDC has provided a great opportunity for students to learn concepts related to Operating Systems, security and even SCADA systems. With the OpenPLC platform, we would like to build a system that will allow the participants to also learn about Cyber Physical systems. With the increasing use of computing techniques in physical systems, it is important to teach security concepts related to these systems.

The project is an effort to create systems for the Cyber Defense Competition that will simulate Cyber Physical concepts. The project will utilize the the OpenPLC platform to interact with the physical systems that we will simulate.

In the end, we would like to have a project that can be integrated into the Cyber Defense competition. Specifically, we would like to have servers setup that can be copied and integrated into a CDC with relative ease. The servers that we setup will have a majority of the system setup, but will include security flaws and other issues the Blue Teams (see section 1.4 for defintion) will need to resolve.

1.3 Operational Environment

We will be virtualizing our system, so our project will exist on the same server that hosts the ISEAGE environment. Therefore, we will not need to worry about physical conditions because the server is already hosted in a secure and appropriate environment.

1.4 Intended Users and uses

Blue Team

Students participating in the Cyber Defense Competitions are responsible for completing set up and securing the systems provided to them via the competition scenario. Blue teams make up the largest demographic of active participants. Each blue team ranges in size from 5-8 individuals. From the perspective of the proposed project, each blue team will receive an individual network of servers revolving around the Factory I/O platform. Their primary job is to become familiar with interfacing with the PLCs within Factory I/O, as well as get them fully functioning and operational. Security will become a priority, as the Red Team will be on the prowl during the competition, looking to take down individual systems and disrupt the manufacturing process.

White Team

The White Team is responsible for running the entire competition the day of, generating the anomalies, maintaining systems, etc. This committee of students distribute the network of servers to each Blue team, overseeing official competition rules, and keep all of the working parts running together smoothly. As our project team gets closer to integrating the virtualized PLC environment in with an actual CDC, we will be working closely with the white team designated for that competition to sure everything is implemented properly.

Red Team

The Red Team is responsible for launching attacks against each of the blue teams throughout the day of the competition, striving to gain access to vulnerable systems, steal flags, and wreck havoc - attempting to make the lives of the blue teams miserable. The Read Team is comprised of security "experts" - including past students, industry professionals, security analysts, penetration testers, etc. By introducing the OpenPLC project into future CDCs, this will have a great impact on the way the Red Team interfaces with each of the Blue Teams. Instead of trying to exploit vulnerable web servers running older versions of outdated software, they will be targeting actual physical systems (through virtualized means). This adds additional layers of difficulty/complexity, as the end result will not be to simply shut down these physical systems or "gain root access", but figure out other ways to cause real-life, practical damage.

Company Sponsorship

Companies have the unique opportunity to become sponsors of current CDCs. Many of them contribute financially, as well as provide employees to compete with the Red Team. They also are on-site the day of the event, recruiting the next generation of cyber security talent, collecting résumés, scheduling interviews, meeting with students, etc. However, very few companies ever get the ability to help design competition scenarios. Their involvement with the competition itself is limited to day-of activities. With designing Factory I/O, we will now allow employers to design different scenarios right alongside us, modeling various cyber-physical systems that occur in their own manufacturing facilities. Employer interest will hopefully increase dramatically with the progressive move towards utilizing the OpenPLC project, and modeling real-life physical situations.

1.5 Assumptions and Limitations

Assumptions

There are several assumptions to be made regarding our proposed design, with these assumptions primarily revolving around the fact that our project will be tightly integrated with the Cyber Defense Competitions (CDCs) held at Iowa State. There will be a maximum of four teams interacting with our project at any one given time (white team, green team, blue team(s), and red team). Each team will have a significant number of users, so our project will be modularized to accommodate the fluctuation each semester. Factory I/O will be the platform our project is based upon. This industry leading software platform allows us to virtualize complex environments and emulate real-life situations. Security will be a major focus in this project, with a primary emphasis based on securing these virtualized systems and protecting them from outside attack (from the red team).

Limitations

The system must live inside of a containerized network environment called ISEAGE (all CDCs occur within ISEAGE). The final product must be built on top of the OpenPLC project. The OpenPLC project is the first fully functional standardized open source PLC. It is a standard industrial controller built upon open-source hardware and real time responses. This platform has been selected for its standardized nature, and for being open source for continued long-term development. While this was not a strict limitation, the ability for this system to exist in a virtualized form was strongly encouraged by our faculty members, for scalability and cost reasons. Since we can assume over 30 blue teams would need their own, individual system, this made perfect sense.

1.6 Expected End Product and Deliverables

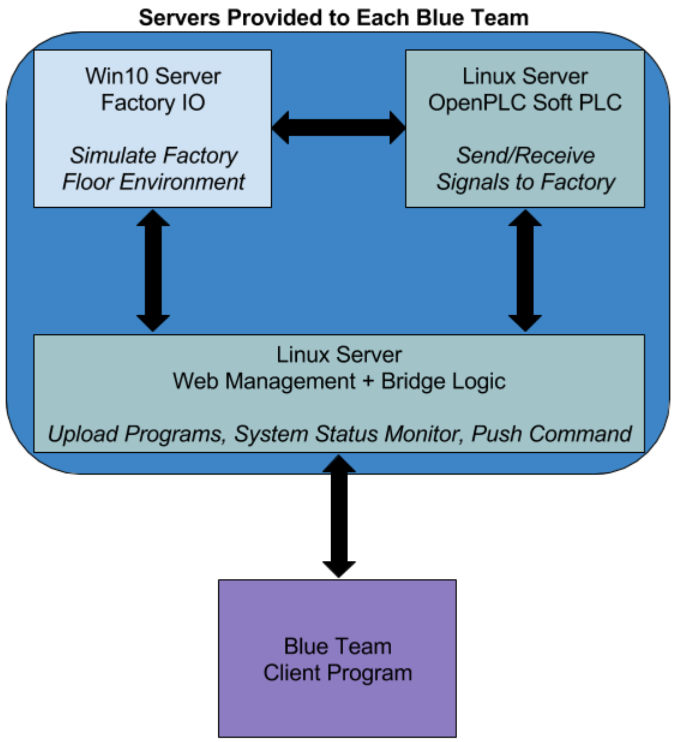
The end product for this project will consist of the following:

- Network of servers. The exact specifications of these servers and their intended functions can be found in section 2.1 of this document, where a breakdown of every service is completed.
- Formalized documentation for our findings on the OpenPLC project, exact step-by-step procedures for building particular systems in Factory I/O, detailed explanation of the intentional vulnerabilities/loopholes we've designed into each system, etc. The list goes on and on, but documentation will surely play a vital role in this project.
- Other ideas of PLC integrations with the CDCs, and what role the OpenPLC project will play in the process of adding these.
- Modularized network environment that is easily scalable and designed for use of many teams, with many inputs, with many external variables.

These products will be deliverable by next spring of 2018, around the April timeframe. Because the CDCs occur earlier in the semester, the first opportunity and CDC to take advantage of the integration of PLCs will be fall of 2018, in which our system will be up and running, fully operational and documented (for future senior design project teams to continue work where we left off).

2. Specifications and Analysis

2.1 Proposed Design



In designing our implementation of the OpenPLC project into the Cyber Defense Competition environment, we have made a shift from our previous hardware-centric focus, which simulated PLCs using Raspberry Pi 3 and Arduino units to manipulate real physical systems, to our current software driven model. This software-centric model utilizes one or more Linux servers to host a virtualized OpenPLC (soft PLC) and Web Management Server allowing the user to interact with the PLC by monitoring its outputs, sending it commands, or uploading new programs for execution.

Our design shall provide a base platform that includes the existing CDC environment with the addition of the above components (soft PLC + Web Management server) and is extendable for producing a variety of potential CDC scenarios with cyberphysical systems; this may be accomplished by changing the virtualized hardware used within the Factory I/O software, or by replacing this Factory I/O module with different simulated or physical hardware entirely.

In addition, our design shall provide a functional simulation of PLC hardware and the systems they control via common protocols used by real-world PLC's, such as Modbus; furthermore, they shall be fed instructions using traditional methods such as Ladder Diagrams.

In researching solutions for simulating the systems controlled/monitored by the PLC, we discovered Factory I/O, which is a software solution that models many pieces of machinery to simulate the environment of a factory floor. This will provide "eye candy" for Blue Team participants; not only does this simulated environment allow participants to visually check that their team's systems are functioning properly, but also serves as a fun and engaging prop for relating their work to real-world applications. Factory I/O's services shall be available via Windows 10 virtual machines. To provide steady rendering of simulated factory floor equipment during a CDC (normal use case), ISU's Hypervisor shall be upgraded to include several modern graphics processing units.

Virtualizing OpenPLC on Linux, rather than exclusively using the Raspberry Pi + Arduino format, has the advantage of allowing our team to accomplish our goals while simultaneously reducing the hardware requirements for each team; if we do not need to provide each team with a Raspberry Pi, our scenarios become scalable for varying competition sizes. For a stable installation, a current up-to-date distribution of Linux shall be used. However, vulnerabilities may be introduced for the Blue Teams to repair by intentionally providing them with outdated Linux distributions or software versions.

An additional Linux system may be used to serve the Web Management portal; when creating the minimal number of virtual machines is desirable, these services may be hosted alongside the soft PLC. Similar to the above, stable configurations should utilize up-to-date software for serving the website and providing intermediary logic between Blue Team clients and the OpenPLC instance. However, it may be useful to provide systems with outdated software to introduce difficulty into the CDC scenario for participants. The web client and any bridge logic should be modularly programmed to allow adaptation for future scenarios.

2.2 Design Analysis

In the early stages of this project, all the plausible ideas of the team were based on an implementation of a Raspberry Pi or Arduino computer. While most of the ideas met the basic project qualifications, they were not as complex or interesting as the team wanted. The initial idea was to use the OpenPLC software on a Raspberry Pi to power a model train and utilities on the track. After some research, the team decided against the train idea because of the limitations pricing and availability of model trains. Finding models that could be altered to be controlled by a PLC was not that difficult, as the hobbyists community is very active. Finding affordable versions of these proved to be very difficult. After abandoning this idea, we avoided requiring specific hardware that we wouldn't be able to create ourselves.

Keeping with a transport theme, the team discussed the idea of a traffic simulator. The PLC would control either self driving cars, stop lights, or other automated utilities. To represent the vehicles, programmable LED light strips would have been used. After much discussion, the team decided to move away from this idea as well. The end model would require a lot of back end code to work but the contestants in the CDC would have very little actual interaction with the model. It seemed as though the interaction for the users would be trivial and the project would be merely 'eye-candy.'

After discussion with our mentors with our concerns that the project could easily become purely cosmetic, the team decided to look for more physically dependant and interactive ideas. We came across some interesting elevator-like implementations based off weight and color of real world objects to take inspiration from. The team discussed the difficulties and freedom of building a custom mechanism like this and in the end decided against it because of the difficulty of scaling. Creating the individual elevators would be time consuming and unrewarding. If the competition had even one more team than elevators, they would not be able to be implemented. The team wanted the customizability of building but the scaling of something easily reproducible.

We found these qualities in a software deliverable. The team believes to have found the tools needed to accomplish them as well as create an interesting and effective representation delivery in Factory I/O. Using multiple systems to handle replicated interactions, all virtualized, creates enough interaction and complication to keep the blue team working for finding vulnerabilities as well as interesting and potentially challenging work within the software itself. The team has also discussed the many ways we could implement short cuts for red team to interact with the software for an even more challenging experience for blue team. We have had success in implementing different components within the software and using other computers on the network to interact with the system and software. As we continue to test the possible interactions between the OpenPLC system and FactoryI/O, we will discover the most optimal ways to take advantage of this software and find creative ways to implement the OpenPLC's and challenge the future blue team.

3 Testing and Implementation

3.1 Interface Specifications

Software Interfacing:

Our front end interface will have two uses. One aspect will be that the blue team will be able to use it to set up security, and have a UI to see the factory. The other side of user interface is to give us the development team a solid way to test that our Ladder Logic files are being uploaded to our SoftPLC correctly, and that signals are being sent to and from the PLC correctly. Using our front end UI, we will be able to test, the securities of our product and lack there of, the signals being sent to the PLC, and the Factory I/O Connection and display. These three portions of our project are some of the most vital portions of this project, and this interface will allow for connection and testing of all of them.

3.2 Hardware and software

Software Used In Testing:

Selenium (http://www.seleniumhq.org/):

Due to us having a front end that many users will be using and securing, we will want to have some automated tests, to make sure that the frontend is setup correctly. Selenium is an open source web interface testing software. We will be able to write scripts that check to see that any changes we make, it not changing some of the required components of our front end.

Cucumber (https://cucumber.io/):

We will use Cucumber to make automated test cases, that are in plain English to allow for other users to come along some time and use or fork our code, and make their own projects. We will use cucumber so when others are changing our UI, and web interfaces they will learn if they did break something, and will be able to narrow their errors down and hopefully fix them.

3.3 Process

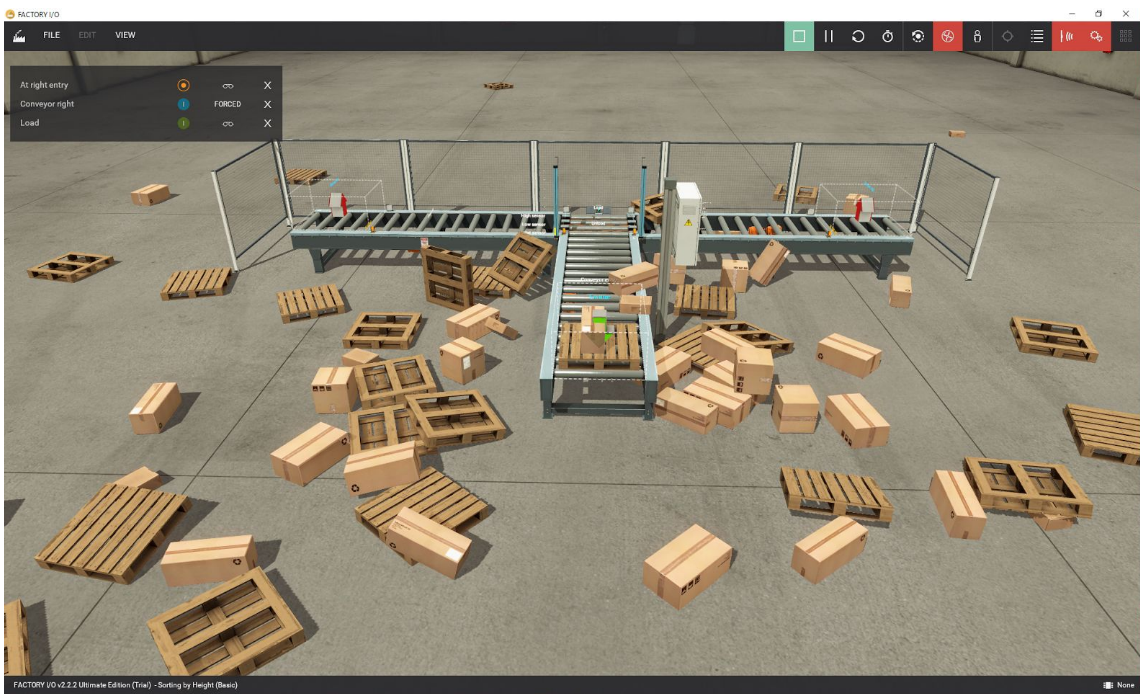
We have used two methods to find our current solution we are implementing into a CDC. The first method we used to find our current implementation is researching into possible solutions that contain OpenPLC. The second method we have used to make our progress is testing and the viability of Factory I/O into our OpenPLC server connection.

A large part of project has been devising potential solutions and seeing if they are practical. We had four possible solutions, a train, light boards, elevator, and Factory I/O all controlled by our OpenPLC. We found that the trains were too expensive to implement. The light board, we researched into the connection, and found that the complexity of programming the lights, and the control boards controlling them were too expensive, and very difficult to create multiple scenarios easily. The elevator was also rejected for the reason of difficulty to implement diverse scenarios. This testing and research led us to Factory I/O.

To test the Factory I/O we set up an OpenPLC server on a linux machine, and then set up Factory I/O on a windows machine. For testing Factory we created a few scenes that are conveyor belts, moving boxes, to show that Factory works as we would like. We then set up a connection between OpenPLC and Factory and made sure we can receive connection calls between both systems. We are currently setting up a Ladder Logic drive to control Factory I/O.

3.4 Results

Factory I/O can be used to display that a factory floor has been hacked, in this case, a scene of a failing factory:



This was the result of our first testing phase of checking that Factory I/O. This test showed that Factory I/O can use PLC's to control a system and that when programmed incorrectly have a visual display that something went wrong. This was one of the reasons we chose Factory I/O for feedback and this test proved it will work.

4 - Closing Material

4.1 Conclusion

Our team has accomplished a significant amount of work so far. We discovered many uses of PLCs in cyber-physical systems and identified the most scalable option. Through discussions with our advisors, we have determined the most desirable option for teaching security in CDCs: simulating a factory floor. We have successfully set up the software so that the OpenPLC can communicate with the virtualized factory in a local environment, and we have put in the necessary requests to get this environment in ISEAGE, the CDC production environment. We are on track to have a test environment setup by winter break, so that we have all of spring semester to create scenarios and fill out vulnerabilities.

Our goal is to have a real-world example of cyber-physical security vulnerabilities to give students experience in defending them. We hope to expose many vulnerabilities for Blue Team to defend and learn about, and to provide ample hacking opportunity for Red Team to exploit. Additionally, we will speak to professionals in the industry to hear first-hand accounts of cyber attacks.

Our course of action is primarily focused on delivering a factory I/O/OpenPLC server interaction in ISEAGE by winter break. Therefore, we are in conversation with the ISEAGE staff to secure a physical server for our use. Until then, we are actively testing factory I/O and OpenPLC on our local machines. Additionally, we regularly meet to brainstorm scenarios, troubleshoot each other's code, and flush out our factory program.

4.2 References

ISU CDC: https://cdc.iseage.org/

Factory I/O: https://factoryio.com/

OpenPLC: http://www.openplcproject.com/