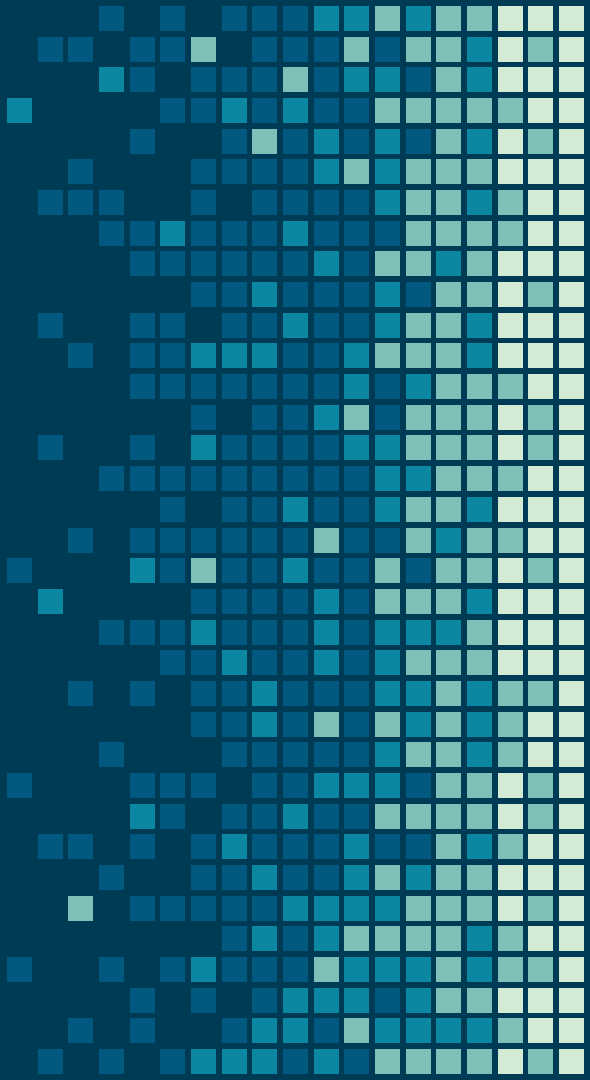
Implementing OpenPLCs into a Cyber Defense Competition

Team 16

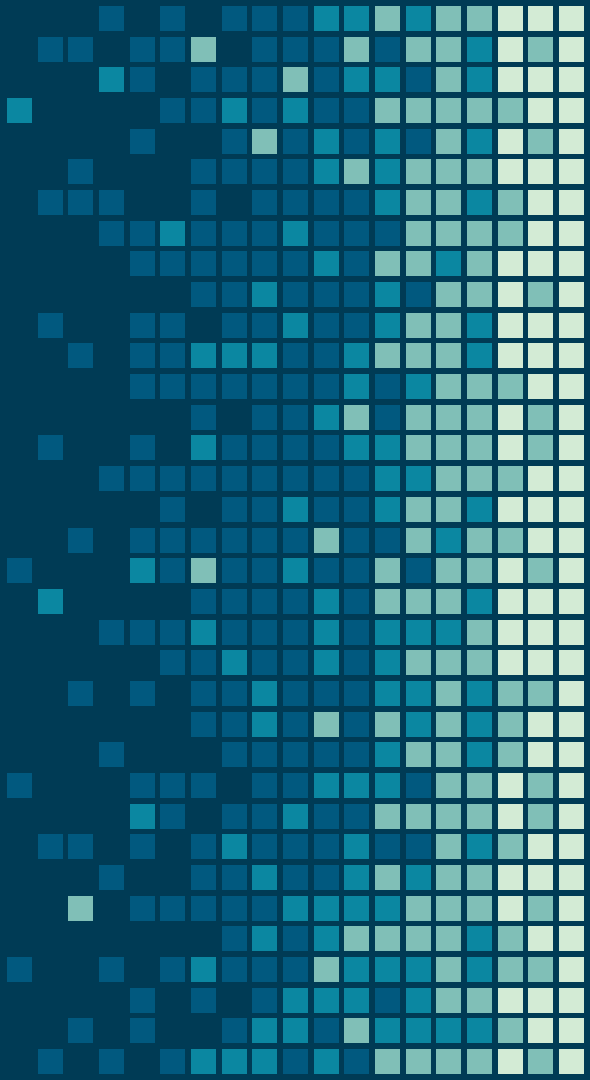
Dr. Jacobson and Dr. Rursch



Team Roles and Responsibilities

- Matthew McGill: App Developer
- Josh Przybyszewski: Lead App Developer
- Nick Springer: Security Engineer
- Brennen Ferguson: Hardware Engineer
- Liam Briggs: Hardware Engineer
- Val Chapman: Test Engineer
- Joseph Young: Security Engineer

Project Plan



Problem Statement

- Simulate cyber-physical infrastructure (OpenPLC)
- Designed for Cyber Defense Competitions (CDC)
- Provide participants with experience securing systems resembling those in real-world infrastructure, such as a factory floor

Project Definitions

- PLC: Programmable Logic Controller
- OpenPLC: open-source PLC platform
- CDC: Cyber Defense Competition
  - Blue Team
  - Red Team

Functional Requirements

- Rely on OpenPLC
- Vulnerabilities in competition scenario
- Quantifiable success in competition

Non-Functional Requirements

- Introduce PLC’s to contestants
- Realistic and relevant scenario
- Easily redistributed and adjusted
- Simple and useful UI

Technical Considerations

- **Portability:** easily deploy to other competitions
- **Scalability:** allow a variable number of competition teams (average of 20-40)
- **Expandability:** allow a variety of scenarios with different simulated equipment

Market Survey

- Little evidence of similar work outside ISU
- Rarely incorporated cyber-physical elements
  - Mock Cities: Defend power and water sys.
- Expand upon existing infrastructure

Resource & Cost Estimate

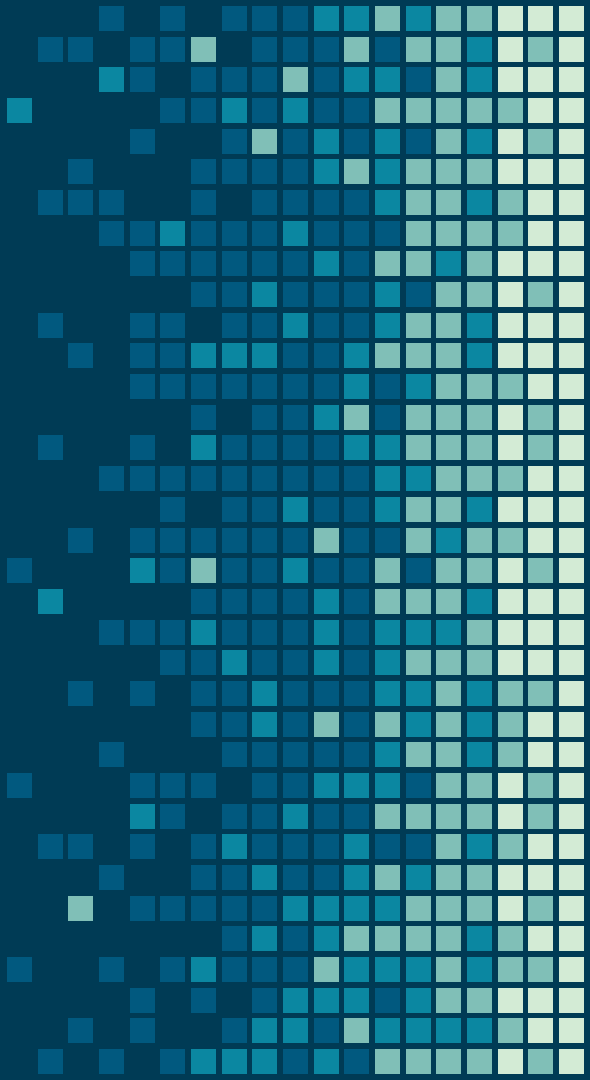
- Resources
  - ISEAGE Platform - developed at Iowa State
  - Factory I/O
  - OpenPLC

Resource & Cost Estimate

- Cost
  - Factory I/O Licensing
    - Utilize demo software
  - Potential for corporate sponsorships

System Design

+ Technology Platforms



OpenPLC

- Project Requirements
- Flexible PLC Solution
- PLCopen Editor

OpenPLC Use Cases

I/O used for

- Lights
- Motors
- Sensors

Implemented on

- Raspberry Pi
- Arduino
- Virtualized

FACTORY I/O

- Simulation
- Visual
- Easy Interface with OpenPLC (TCP/IP Modbus)

Server Setup

- VCenter
- Ubuntu, Linux Mint, Windows Server 2016
- Web Servers
- Factory I/O
- Vulnerabilities

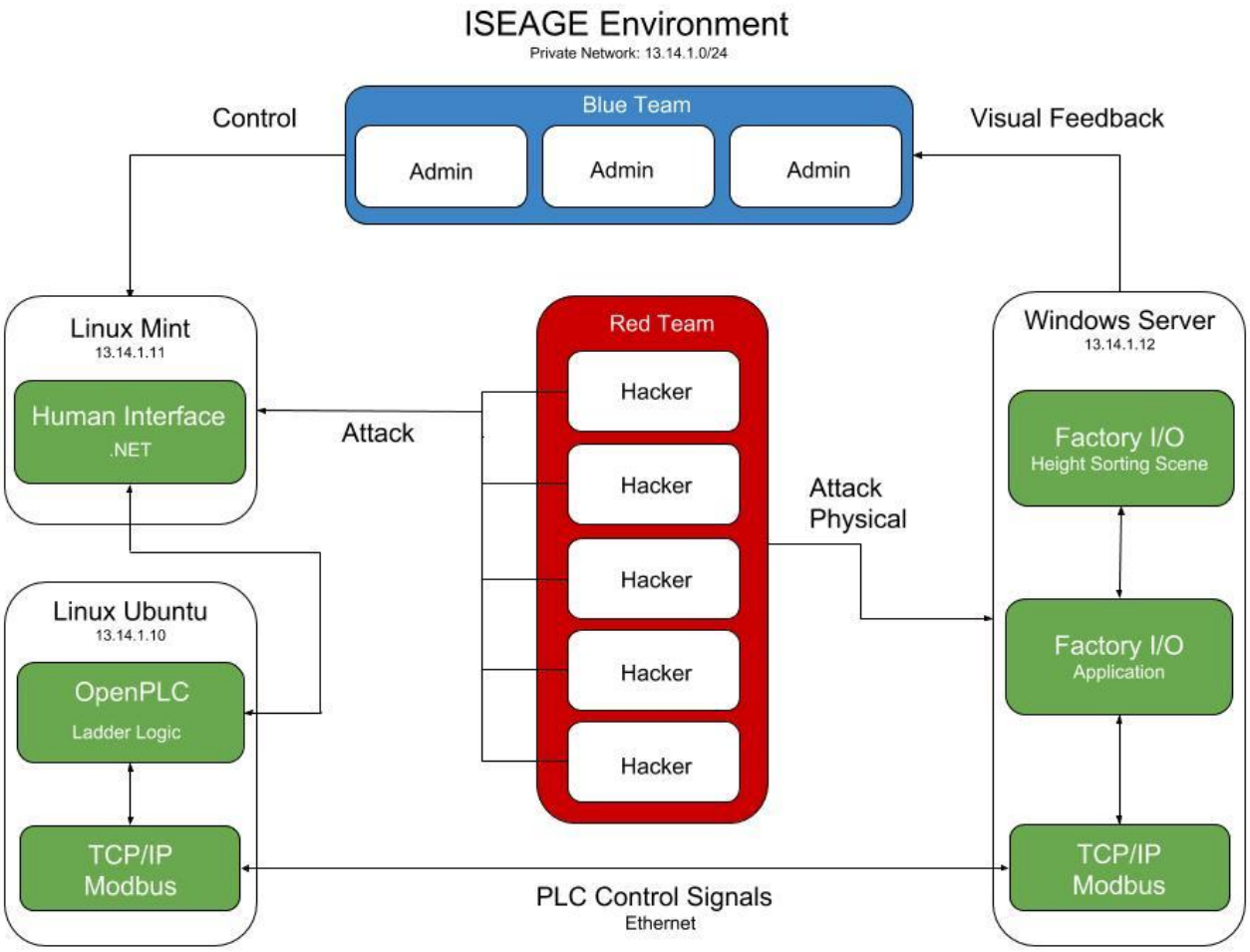
Windows Server 2016

OpenPLC/Node JS

- Easy to set up
- Upload ladder logic
  - Default function
- Leave mostly untouched

Angular Dart Application

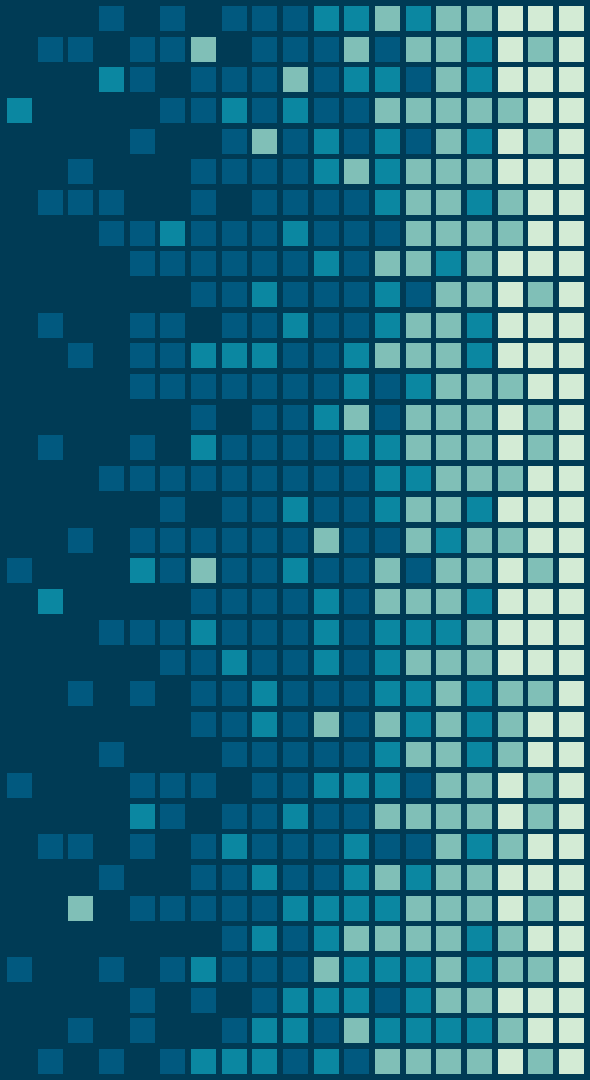
- Remote management of the factory
  - View sensor values, send signals
- Back-end: .NET Web-API project
  - Factory I/O SDK
- Front-end: AngularDart
  - API requests


ISEAGE Environment
Private Network: 13.14.1.0/24
Control
Blue Team
Admin
Admin
Admin
Visual Feedback
Linux Mint
13.14.1.11
Human Interface
.NET
Attack
Red Team
Hacker
Hacker
Hacker
Hacker
Hacker
Attack
Physical
Windows Server
13.14.1.12
Factory I/O
Height Sorting Scene
Factory I/O
Application
Linux Ubuntu
13.14.1.10
OpenPLC
Ladder Logic
TCP/IP
Modbus
TCP/IP
Modbus
PLC Control Signals
Ethernet

Testing

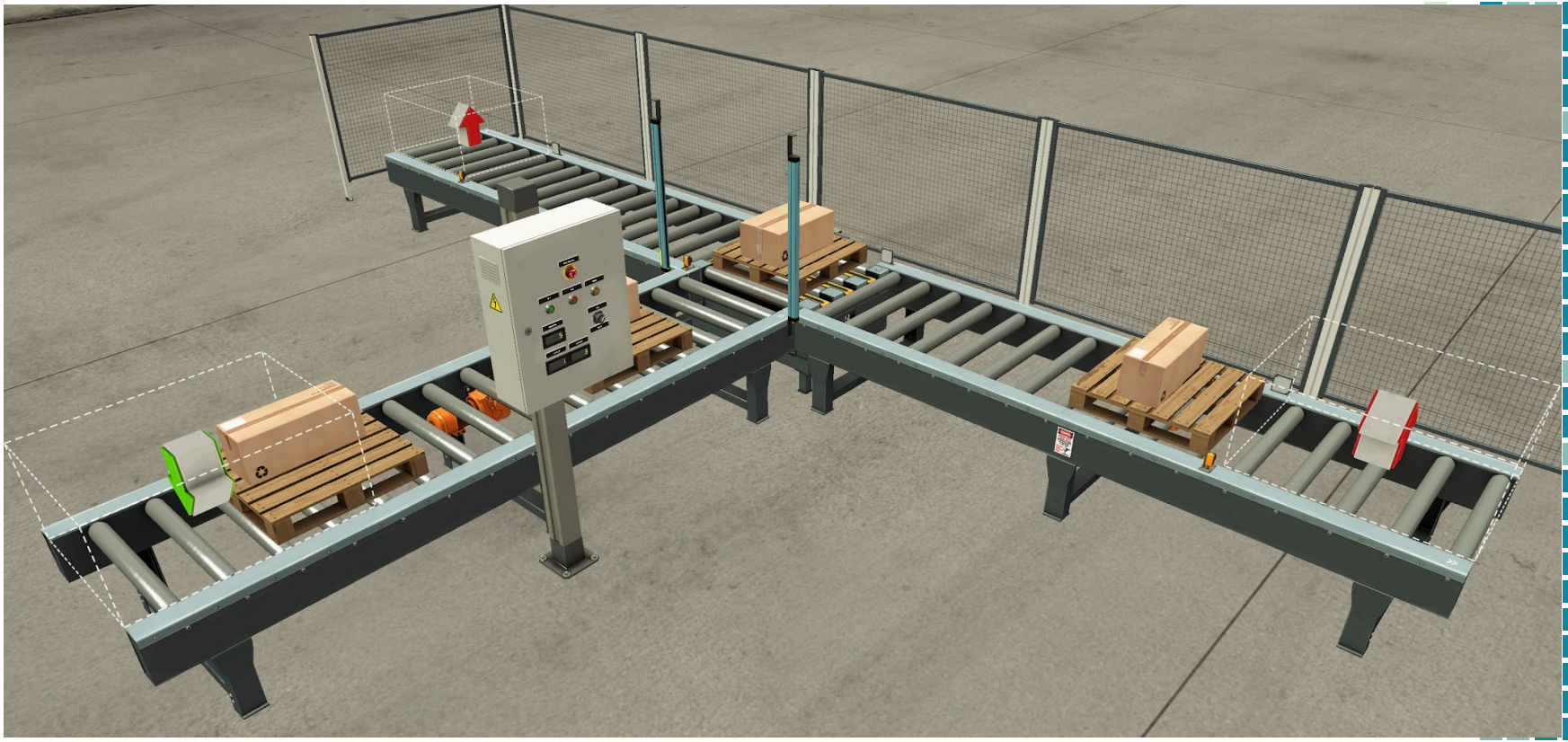
- Can OpenPLC be implemented into as a Cyber Phyical CDC?
- Can the system be scaled to the size of the CDC?
- Can Factory I/O and OpenPLC be implemented every CDC?

CDC Scenario



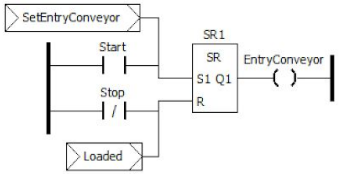
The Nile Shipping and Bookstore Co.

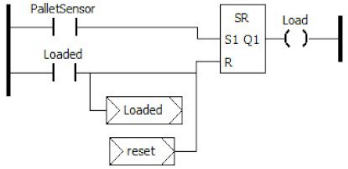
- Sorting boxes by height
  - Increases shipping efficiencies
- Blue team goal: run a smooth factory
- Red team goal: disrupt logic

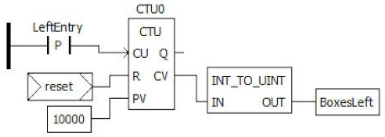


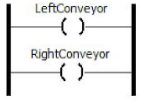
Description: Class Filter: All

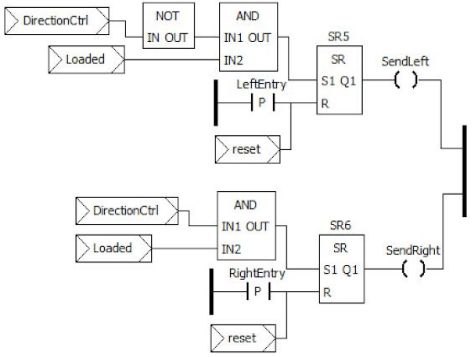
#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	Start	Local	BOOL	%QX0.0			
2	Stop	Local	BOOL	%QX0.1			
3	PalletSensor	Local	BOOL	%QX0.2			
4	HighSensor	Local	BOOL	%QX0.3			
5	Loaded	Local	BOOL	%QX0.4			
6	LeftEntry	Local	BOOL	%QX0.5			

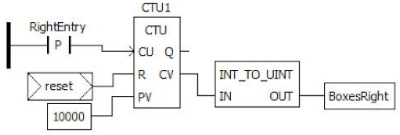


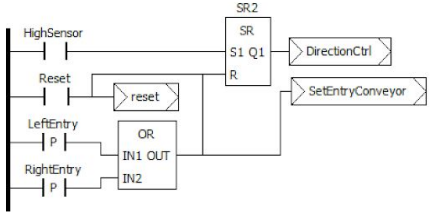


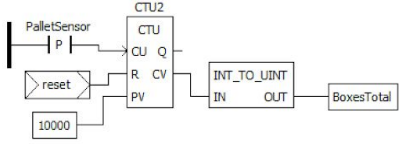












Next Steps

- Develop Scenario to be used in the CDC
- Run a mock CDC with students
- Automate system setup

Questions?

You can find us at:

https://sdmay18-16.sd.ece.iastate.edu