Development of IoT Components for CDC

DESIGN DOCUMENT

Team Number: 14
Client: Daji Qiao
Advisers: Daji Qiao
Team Email: sdmay22-14@iastate.edu
Team Website: https://sdmay22-14.sd.ece.iastate.edu

Team Members/Roles (TBD):
Jake Martin - Scripting Developer/Task Manager
Jacob Foster - IoT Engineer/Point of contact
Megan Litterer - IoT Website Developer/Team Website developer
Justin Tyler- Minutes Taker/Initial Tester

Executive Summary

Development Standards & Practices Used

The standards and practices utilized by this team are the following:

ISO 9660

IEEE 802.1Qca-2015 - IEEE Standard for Local and metropolitan area networks-- Bridges and Bridged Networks

Summary of Requirements

Functional Requirements

- IoT device integration into Iowa State's ISEAGE/CDC environment
    - Virtualization of at least two IoT devices for integration (into CDC main scenario)
    - At least one physical IoT device integrated (anomaly)
    - Physical and Virtual IoT devices shall be different from each other
- Easily reproducible:
    - IoT infrastructure that can be used in future CDCs
    - Vulnerabilities, Attacks, Defense Strategies
    - Machines that can be remade and deployed as needed
- A packaged, distributable bundle which holds resources necessary to configure IoT devices into a CDC
    - Formal documentation, ISOs, and packaged software to develop a CDC
    - Automated scripts for initial VM setup
    - Automated scripts to emulate the Red Team

Non-Functional Requirements

- Ease-of-use network interfacing
- Different levels of vulnerability difficulty
- High performance virtual machines with minimal virtual hardware requirements
- Clear setup and use instructions

Economic Requirements

- Keep budget reasonable while looking at vulnerabilities vs. cost
- Inexpensive equipment can be bought for more in depth recreation
- Expensive equipment is to be completely virtual

Applicable Courses from Iowa State University Curriculum

- CPRE 230 - Cyber Security Fundamentals
- CPRE 231 - Cyber Security Concepts and Tools
- CPRE 430 - Network Protocols and Security
- COMS 252 - Linux Operating System Essentials
- COMS 309 - Software Development Practices

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- HTML coding
- Types of hacks and how to prevent them
- Bash scripting
- Virtual machine development
- Flask web development
- Jquery and Ajax client-side development
- Split networking (public/private networks)
- DNS (Domain Name Service)

Table of Contents

1 Team 6

1.1 Team Members 6

1.2 Required Skill Sets for Your Project 6

1.3 Skill Sets covered by the Team 6

1.4 Project Management Style Adopted by the team 6

1.5 Initial Project Management Roles 6

2 Introduction 7

2.1 Problem Statement 7

2.2 Requirements & Constraints 7

2.3 Engineering Standards 8

2.4 Intended Users and Uses 8

3 Project Plan 9

3.1 Project Management/Tracking Procedures 9

3.2 Task Decomposition 9

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria 9

3.4 Project Timeline/Schedule 10

3.5 Risks And Risk Management/Mitigation 10

3.6 Personnel Effort Requirements 10

4 Design 11

4.1 Design Context 11

4.1.1 Broader Context 11

4.1.2 User Needs 11

4.1.3 Prior Work/Solutions 12

4.1.4 Technical Complexity 13

4.2 Design Exploration 13

4.2.1 Design Decisions 13

4.2.2 Ideation 13

4.2.3 Decision-Making and Trade-Off 14

4.3 Proposed Design 15

4.3.1 Design Visual and Description 15

4.3.2 Functionality 16

4.3.3 Areas of Concern and Development 17

4.4 Technology Considerations 17

4.5 Design Analysis 17

4.6 Design Plan 18

5 Testing 20

5.1 Unit Testing 20

5.2 System Testing 20

5.3 Regression Testing 20

5.4 Acceptance Testing 21

5.5 Security Testing 21

5.6 Results 21

6 Implementation 21

7 Professionalism 22

7.1 Areas of Responsibility 22

7.2 Project Specific Professional Responsibility Areas 22

7.3 Most Applicable Professional Responsibility Area 23

8 Closing Material 24

8.1 Discussion 24

8.2 Conclusion 24

8.3 References 24

8.4.1 Team Contract 25

List of Figures

1 Gantt Chart 10

2 Overview of ISERink environment 12

3 Virtual IoT decision matrix 14

4.1 Packaged CDC Bundle diagram 15

4.2 Diagram of toggling the smartplug interface 15

4.3 Smartplug login interface 16

4.4 Smartplug website user interface with list of running services 16

4.5 Diagram of firewall in our packaged CDC 18

4.6 Diagram of blackbox device in our packaged CDC 18

4.7 Diagram of IoT Smartplug in our packaged CDC 19

4.8 Diagram of Desktop in our packaged CDC 19

1 Team

1.1 Team Members

Jacob Foster Jake Martin Megan Litterer Justin Tyler

1.2 Required Skill Sets for Your Project

Knowledge of the CDC (Blue Team, White Team, Red Team, etc.) and ISEAGE environment

Knowledge of common programming languages, Linux OS commands, bash scripting, and creation and configuration of Virtual Machines

Comprehension of basic cybersecurity concepts and how they relate to IoT devices

Documentation etiquette and effective communication abilities

1.3 Skill Sets covered by the Team

Jacob: Cybersecurity Engineer: a thorough understanding of cybersecurity concepts, techniques, and tools. Experience with CDCs and ISEAGE.

Jake: Computer Engineer: understanding of networking and cybersecurity concepts, Java and C programming, bash scripting, and Linux virtual machine configuration

Justin: Electrical engineer: Organization, C programming, public speaking

Megan: Software Engineer: understanding of Javascript, Python, Java, and C programming, embedded systems, cyber security, diagram design, network

1.4 Project Management Style Adopted by the team

We are using Scrum as our project management style. Project managers using this style should clearly explain responsibilities and roles to team members, and foster ongoing communication between all members of the team.

This applies perfectly with our project as we are periodically communicating with CDC staff and our adviser to implement different parts of our project based on highest priority at the current time.

1.5 Initial Project Management Roles

Jake Martin - Scripting Developer/Task Manager

Jacob Foster - IoT Engineer/Point of contact

Megan Litterer - IoT Website Developer/Team Website developer

Justin Tyler- Minutes Taker/Initial Tester

2 Introduction

2.1 Problem Statement

Living in the 2st century, everything is connected to the Internet; whether it be a smart watch, speaker, camera, or even a fridge, it seems that all of our devices are linked to the global Internet in some capacity. With this growing number of IoT devices, also comes with a growing number of vulnerabilities for hackers to exploit.

The purpose of our project is to raise awareness of the dangers surrounding IoT devices. To do so, we will be selecting and creating a virtual and physical IoT (Internet of Things) device with built in cybersecurity vulnerabilities for the CDC (Cyber Defence Competition) as well as a distributable package for anyone to download. Implementation in the CDC will allow students to practice securing vulnerable IoT devices. The lessons learned from dealing with vulnerable devices will create experienced cybersecurity professionals, and will enable software developers to program with a security-focused mindset.

2.2 Requirements & Constraints

Functional Requirements

- IoT device integration into Iowa State's ISEAGE/CDC environment
  - Virtualization of at least two IoT devices for integration (into CDC main scenario)
  - At least one physical IoT device integrated (anomaly)
  - Physical and Virtual IoT devices shall be different from each other
- Easily reproducible:
  - IoT infrastructure that can be used in future CDCs
  - Vulnerabilities, Attacks, Defense Strategies
  - Machines that can be remade and deployed as needed
- A packaged, distributable bundle which holds resources necessary to configure IoT devices into a CDC
  - Formal documentation, ISOs, and packaged software to develop a CDC
  - Automated scripts for initial VM setup
  - Automated scripts to emulate the Red Team
- Physical and Virtual IoT devices shall be different from each other

Non-Functional Requirements

- Ease-of-use network interfacing
- Different levels of vulnerability difficulty
- High performance virtual machines with minimal virtual hardware requirements
- Clear setup and use instructions

Economic Requirements

- Keep budget reasonable while looking at vulnerabilities vs. cost
- Inexpensive equipment can be bought for more in depth recreation
- Expensive equipment is to be completely virtual

Ethical Requirements

- Vulnerability Disclosure
    - International Organization for Standardization (ISO) 29147:2014
        - Will follow this standard to create vulnerability reports and submit to vendors
    - MITRE CVE Database
        - Find relevant past vulnerability information in the MITRE CVE database
- These devices are constructed for the purposes of education, and not in any way to be malicious.

2.3 Engineering Standards

- ISO 9660
    - Our simulated IoT will be connected to a simulated router/firewall that will act as a bridged network.
- IEEE 802.1Qca-2015 - IEEE Standard for Bridges and Bridged Networks
    - Our simulated IoT will be connected to a simulated router/firewall that will act as a bridged network.

2.4 Intended Users and Uses

- People learning about Cyber Security or the concept of IoT
- Participants of the ISU CDC
- Universities and highschools besides ISU

3 Project Plan

3.1 Project Management/Tracking Procedures

We are using Scrum as our project management style. Project managers using this style should clearly explain responsibilities and roles to team members, and foster ongoing communication between all members of the team.

This applies perfectly with our project as we are periodically communicating with CDC staff and our adviser to implement different parts of our project based on highest priority at the current time.

We will use Gitlab to keep track of everything because GitLab allows us to keep track of process, version history, etc.

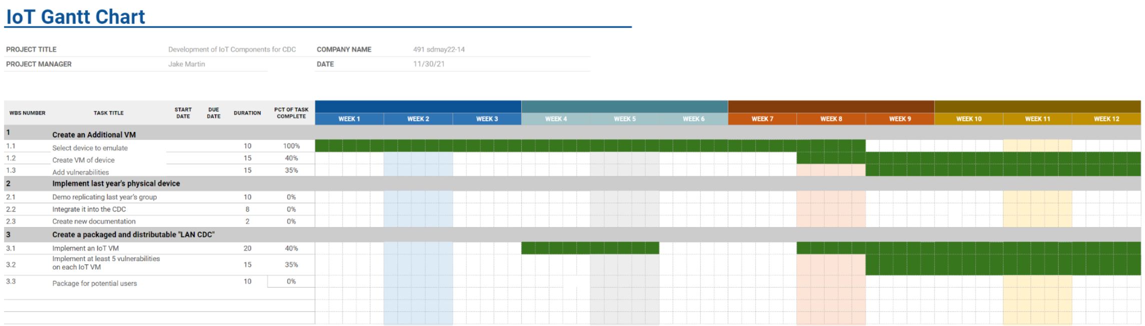
3.2 Task Decomposition

- Create an additional VM
    - Select device to emulate
    - Create VM of device
    - Add vulnerabilities
    - Package for potential users
- Work with last years Physical Device
    - Perform demo replicating last year’s group
    - Integrate it into the CDC
    - Create new documentation
- Create a packaged/downloadable IoT environment
    - A downloadable package in VirtualBox for anyone with VirtualBox installed to download

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Integrate previous team’s physical camera IoT device into CDC.
- Have a virtual LAN CDC implemented with 2 IoT devices.
    - Virtual IoT boots in less than 5 minutes.
    - Implement at least 5 vulnerabilities on each IoT VM.
- For each Vulnerability we implement, we shall also script an exploit to take advantage of the vulnerability. This will ensure the red team can be automated in case teams need to substitute automation for volunteer professional penetration testers.
- If a lightweight program is selected, have it be runnable on non-high end computers.

3.4 Project Timeline/Schedule



IoT Gantt Chart

PROJECT TITLE	Development of IoT Components for CDC	COMPANY NAME	491 sdmay22-14
PROJECT MANAGER	Jake Martin	DATE	11/30/21

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Create an Additional VM**				
1.1	Select device to emulate			10	100%
1.2	Create VM of device			15	40%
1.3	Add vulnerabilities			15	35%
2	**Implement last year's physical device**				
2.1	Demo replicating last year's group			10	0%
2.2	Integrate it into the CDC			8	0%
2.3	Create new documentation			2	0%
3	**Create a packaged and distributable "LAN CDC"**				
3.1	Implement an IoT VM			20	40%
3.2	Implement at least 5 vulnerabilities on each IoT VM			15	35%
3.3	Package for potential users			10	0%

Figure 1: Gantt chart

3.5 Risks And Risk Management/Mitigation

Agile project can associate risks and risk mitigation with each sprint. The risk of not getting last year's project to work is greater than 0.5. Therefore, in the event we are unable to make last year's project functional, we shall perform one of the following tasks:

- We shall develop a completely unrelated IoT device in replacement for their inoperable deliverable.
- We shall develop another device which emulates last year's functionality and vulnerabilities.
- We shall document our failed efforts in getting last year's materials to work for the next team to succeed us.

The risk of not integrating our physical IoT device into ISEAGE is greater than 0.5. Therefore, in the event we fail to integrating our physical IoT device into ISEAGE, we shall perform one of the following tasks:

- We shall thoroughly document our efforts and processes so that next year's team is supplied with enough resources to see their success
- We shall default to producing a Virtual VM that emulates a physical IoT device

3.6 Personnel Effort Requirements

TASK	PROJECTED EFFORT (in PersonHours)
Integrate Our New VM into the Spring CDC	75
Work with last year Physical IoT Device	50
Develop a Downloadable CDC	50

4 Design

4.1 Design Context

4.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	We must produce a virtual machine that possesses vulnerabilities that can be exploited for education; However, we must also ensure that we do not maliciously introduce vulnerabilities which can be exploited from unwanted threat actors with the intention to cause harm. We have a responsibility to ensure the products we create do not threaten the integrity of legitimate networks and devices.	Exposing a vulnerable IoT device from a legitimate network to the public-facing internet. Configuring a vulnerable IoT device that compromises the security of a functional network
Global, cultural, and social	Our device will help the CDC create a cultural shift towards internet security. By bringing devices' vulnerabilities to peoples' attention, they will be more conscious of how cybersecurity affects their daily lives.	Enable people to gain hands-on experience on IoT vulnerabilities to educate and spread awareness.
Environmental	We must be conscious of the hardware we acquire and the software we produce. We have an obligation to make our deliverables as small and functional as possible.	We will try to select a physical device that has its raw resources responsibly resourced. We must not unnecessarily create virtual machines that consume excessive and unnecessary resources.
Economic	The physical device must be as affordable as we can make it. While we only have to have the one, our clients will have to purchase a plethora. If the price is too high, this will quickly compound.	We shall not purchase an IoT device which costs more than $100.

4.1.2 User Needs

The Cyber Security Department and the Cyber Defence Competition needs a way to simulate the Internet of Things because buying the devices for each group would be incredibly expensive.

The students in the Cyber Security Department and the participants in the Cyber Defense Competition must use the devices we implement in ISERink to complete schoolwork and the competition.

4.1.3 Prior Work/Solutions

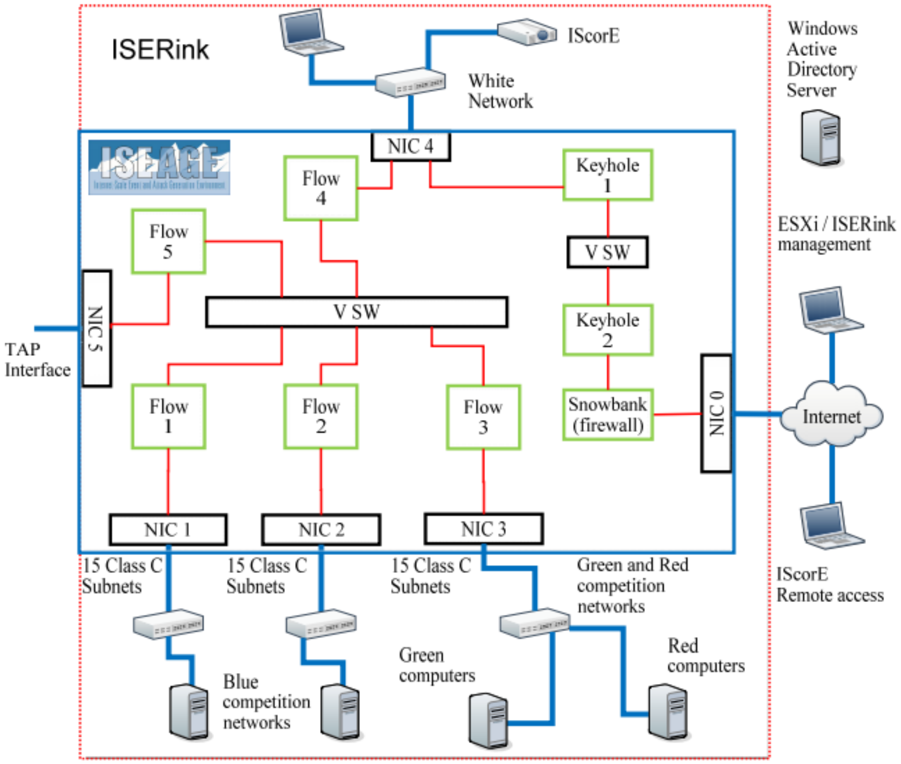


Figure 3 ISERink VM topology

Figure 2: Overview of ISERink environment

- Our project design stems from the already existing ISERink environment. ISERink is a virtual environment that allows multiple teams to safely defend and attack virtual devices/machines. The virtual machines are implemented with services that have vulnerabilities that need to be mitigated by the Blue Team, exploited by the Red Team, and usable by the White Team.
- Relevant background review for project
    - OWASP Top 10 is a good resource. Many IoT devices have web interfaces with common vulnerabilities listed and described in the OWASP Top 10.
    - We are following the work documented for ISERink as seen here. This link contains information about the network architecture of which we are to follow to successfully integrate our IoT components into the Spring CDC for Iowa State University.
    - There was a team last year with a nearly identical goal to ours. We both are tasked with creating a physical and virtual IoT device that the CDC can use. The end parts should be similar, but we want a much heavier emphasis on usability. We will modify the previous team's work to match our own allowing the CDC to have an easier time with its use. Previous team's getlab here.

4.1.4 Technical Complexity

Our project includes integrating physical and virtualized IoT devices into ISERink. This includes manually configuring these devices to use DHCP, export their commands and traffic to use Iowa State's ISELAB proxy, and configure other various exports to manually configure network settings.

Alongside our requirements to integrate our deliverables into ISEAGE, we must also study and replicate common security vulnerabilities. We must not only be proficient enough to comprehend security vulnerabilities (i.e., Buffer Overflow, XSS, CSRF), but we must also know enough of the underlying causes to emulate and fully recreate them within an already secure product.

4.2 Design Exploration

4.2.1 Design Decisions

1. Choose an IoT device to implement.
2. Decide vulnerabilities that can be both attacked and mitigated.
3. Implement a virtual machine from within ISERink
4. Create bash scripts for initial VM configuration and installation

4.2.2 Ideation

"Choose an IoT device to implement."

We will choose an IoT device from a decision matrix composed of the following traits. These traits were given to us as considerations to keep in mind when picking our IoT devices.

a. Flashiness: The final product should have some aesthetic/entertaining exploit that can be demonstrated

b. Cost: The estimated cost to acquire the IoT device

c. CVSS Score: The common vulnerability score system ranks the harm of a given exploit from Low, Medium, High, and Critical. We want a High CVSS score. This is weighted double as it is the most important aspect.

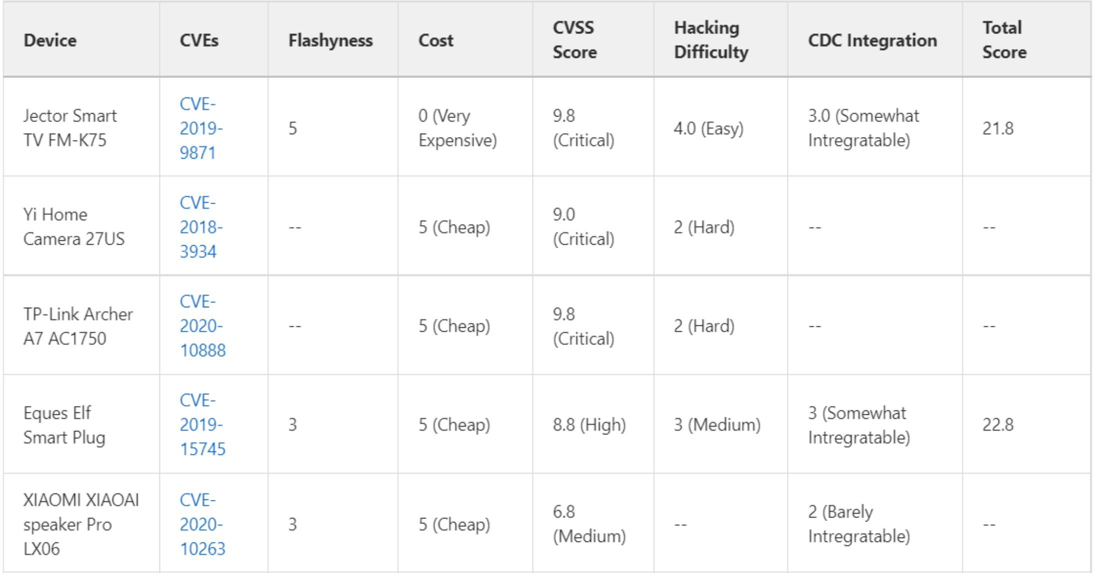
d. Hacking Difficulty: The difficulty/expertise needed to successfully exploit the device. (i.e., default passwords (Easy) vs buffer overflow (Hard))

e. CDC Integration: How well the device would integrate into the CDC (i.e., providing green team services, red team vulnerabilities, blue team defenses)

f. Total Score: A higher score a score is, the better the device

4.2.3 Decision-Making and Trade-Off

Demonstrate the process you used to identify the pros and cons or trade-offs between each of your ideated options. You may wish you include a weighted decision matrix or other relevant tool. Describe the option you chose and why you chose it.



Device	CVEs	Flashyness	Cost	CVSS Score	Hacking Difficulty	CDC Integration	Total Score
Jector Smart TV FM-K75	CVE-2019-9871	5	0 (Very Expensive)	9.8 (Critical)	4.0 (Easy)	3.0 (Somewhat Intregratable)	21.8
Yi Home Camera 27US	CVE-2018-3934	--	5 (Cheap)	9.0 (Critical)	2 (Hard)	--	--
TP-Link Archer A7 AC1750	CVE-2020-10888	--	5 (Cheap)	9.8 (Critical)	2 (Hard)	--	--
Eques Elf Smart Plug	CVE-2019-15745	3	5 (Cheap)	8.8 (High)	3 (Medium)	3 (Somewhat Intregratable)	22.8
XIAOMI XIAOAI speaker Pro LX06	CVE-2020-10263	3	5 (Cheap)	6.8 (Medium)	--	2 (Barely Intregratable)	--

Figure 3: Virtual IoT decision matrix

4.3 Proposed Design

We have created two scripts to emulate the functionality of an IoT Smartplug, as well as scripts that configure vulnerable services on a linux server. We have developed a vulnerable web application to serve as the interface between our users and the IoT Smartplug. We have tested our scripts via demonstrations and tools to exploit vulnerabilities (i.e Hydra.)

4.3.1 Design Visual and Description

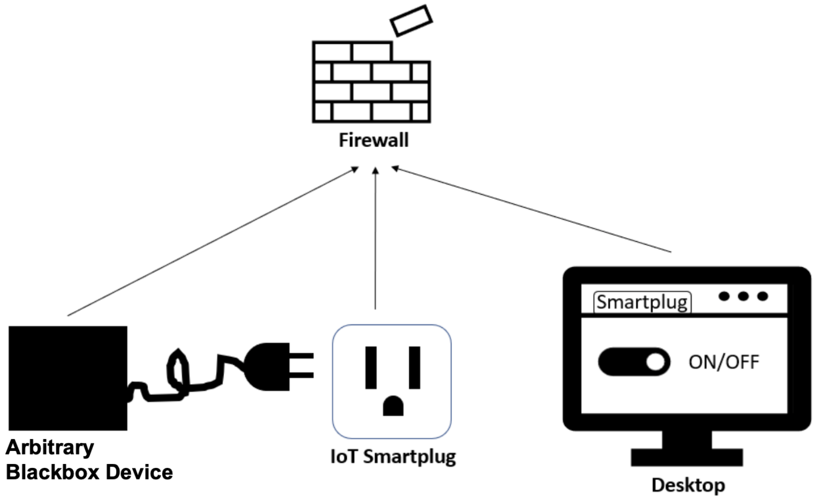
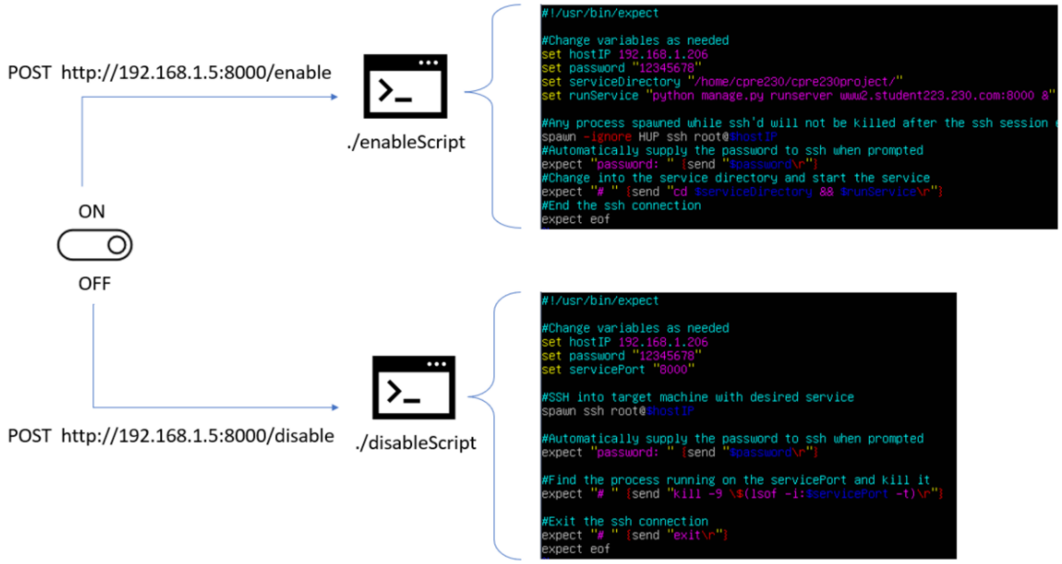


Figure 4.1: Packaged CDC Bundle diagram


POST http://192.168.1.5:8000/enable
./enableScript
ON
OFF
POST http://192.168.1.5:8000/disable
./disableScript

```
#!/usr/bin/expect

#Change variables as needed
set hostIP 192.168.1.206
set password "12345678"
set serviceDirectory "/home/cpre230/cpre230project/"
set runService "python manage.py runserver www2.student223.230.com:8000 &"

#Any process spawned while ssh'd will not be killed after the ssh session
spawn -ignore HUP ssh root@$hostIP
#Automatically supply the password to ssh when prompted
expect "password: " {send "$password\r"}
#Change into the service directory and start the service
expect "# " {send "cd $serviceDirectory && $runService\r"}
#End the ssh connection
expect eof
```

```
#!/usr/bin/expect

#Change variables as needed
set hostIP 192.168.1.206
set password "12345678"
set servicePort "8000"

#SSH into target machine with desired service
spawn ssh root@$hostIP

#Automatically supply the password to ssh when prompted
expect "password: " {send "$password\r"}

#Find the process running on the servicePort and kill it
expect "# " {send "kill -9 \$(lsof -i:$servicePort -t)\r"}

#Exit the ssh connection
expect "# " {send "exit\r"}
expect eof
```

Figure 4.2: Diagram of toggling the smartplug interface

4.3.2 Functionality

Our design is intended to emulate an IoT Smartplug device. This device has a web portal that allows a user to login to the website and remotely control the power supply to devices plugged into the smartplug. The figures below show the login page and the template for the functionality of the smartplug.

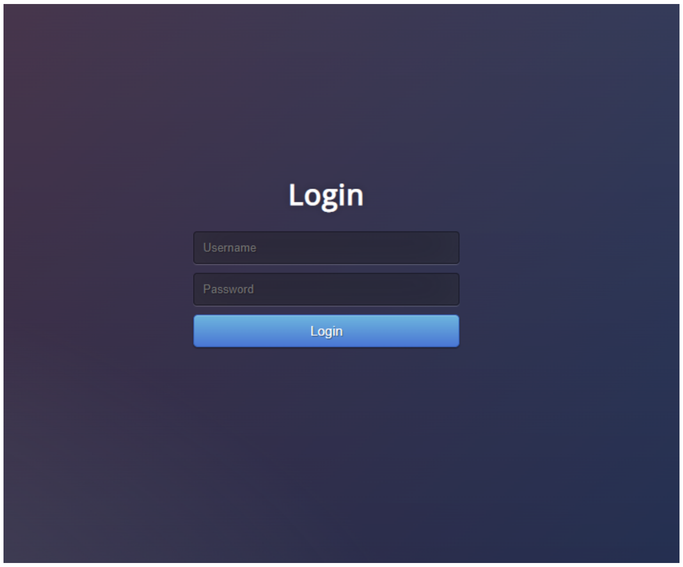
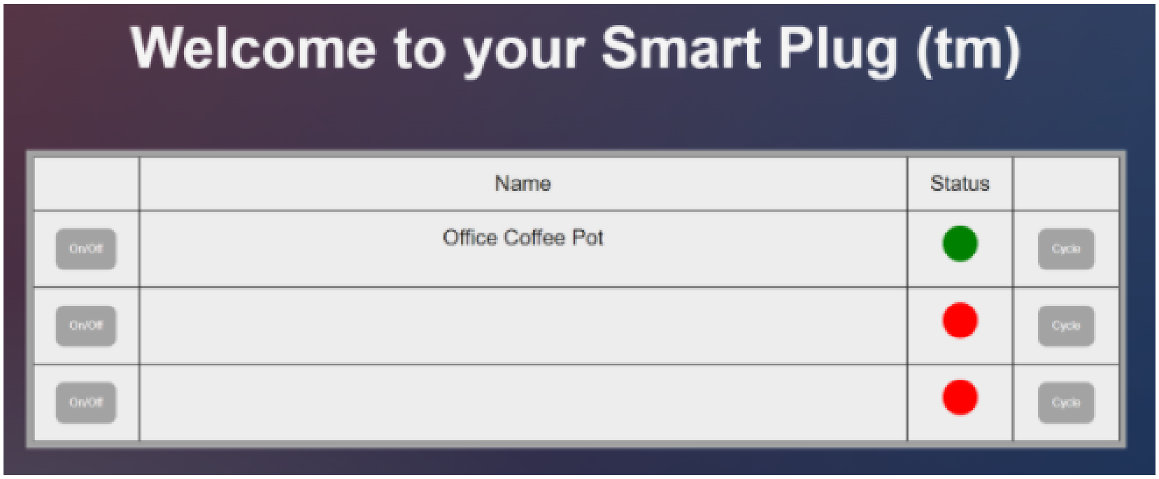


Figure 4.3: Smartplug login interface



Welcome to your Smart Plug (tm)

	Name	Status	
On/Off	Office Coffee Pot	🟢	Cycle
On/Off		🔴	Cycle
On/Off		🔴	Cycle

Figure 4.4: Smartplug website user interface with list of running services

The current design meetings the function and non-functional requirements very well. Because the design is scripted, it can be implemented in both Virtualbox and ISEAGE, which makes the product bundable and distributable. The website also enabled differing levels of exploit difficulty with more attack vectors.

4.3.3 Areas of Concern and Development

Our current design relies on remotely controlling an arbitrary service. Our implementation only emulates a Smartplug by turning off and on critical services on a remote server as opposed to turning off and on the physical device. This requires our team to have our smartplug have the ability to autonomously and remotely control the services on another machine, which introduces dependency conflicts.

We brought these concerns to the CDC staff and came to the conclusion that the dependencies can be enforced from within the CDC white team. Inside Virtualbox we have control over the services and the environment, which means there is no issue for dependencies other than the need to create an arbitrary service which is controlled by our smartplug device.

4.4 Technology Considerations

The technology we used requires certain software to be installed, which takes up more space on virtual machines. In addition, the operating systems we chose have a significant impact on performance and storage. The strength of our implementation is that it is minimal in size, but requires lower performance.

If we were to prioritize performance over storage constraints, we could use larger operating systems which are capable of running applications with better performance. This however would have tradeoffs between storage.

4.5 Design Analysis

Our design from 3.3 currently works. We found success in our design by running multiple versions of different implementations with the CDC staff to hear their opinion. Because we sought advice and feedback in the design phase, we were able to build services and products that are functional in their current environments.

Because we have two different environments (ISEAGE and VirtualBox), we may consider prioritizing different functions. Because we prioritize storage in Virtualbox does not mean we have to make the safe performance tradeoffs within ISEAGE. We may produce two separate environments which operate the same, but with different emphasis on storage and performance.

4.6 Design Plan

Firewall

A FreeBSD operating system specialized to operate as a network firewall with routing capabilities. Used to create a downloadable internal network.

Dependencies:
pfSense 2.5.1 OS

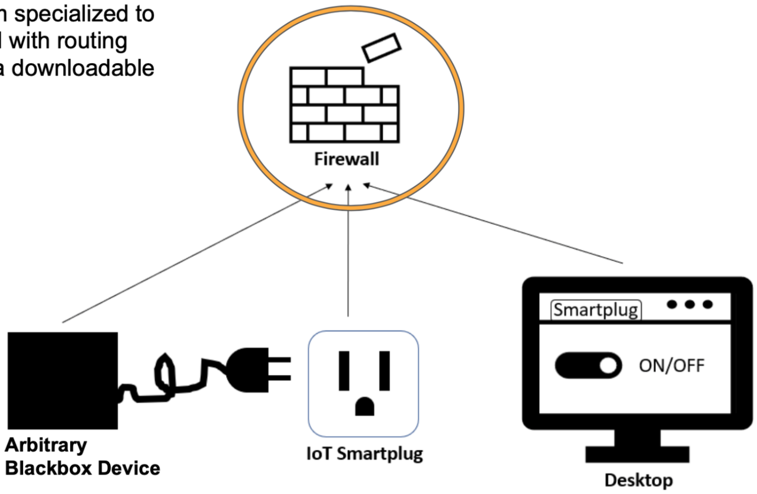


Figure 4.5: Diagram of firewall in our packaged CDC

Arbitrary Blackbox Device

A Linux-based device plugged into the IoT Smartplug. This device hosts an arbitrary service that users depend on. (i.e., a webserver, a coffee machine, lights, etc)

Dependencies:
Any linux-based OS

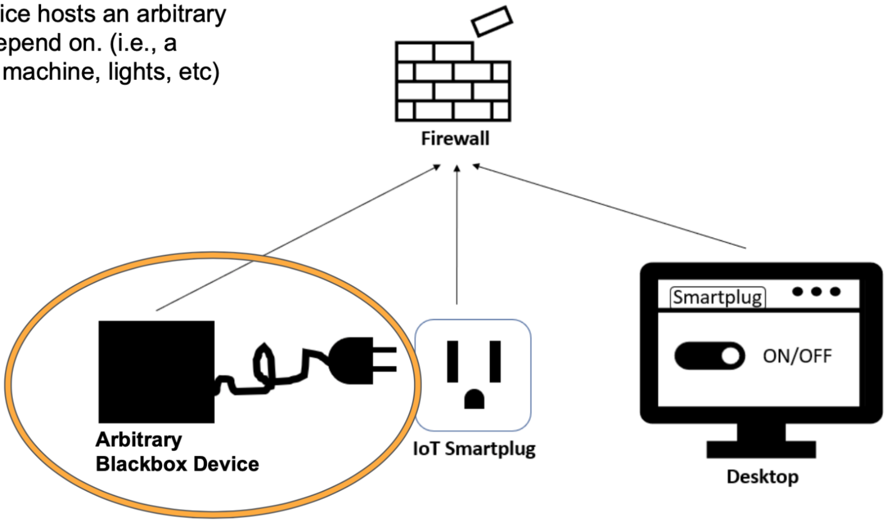
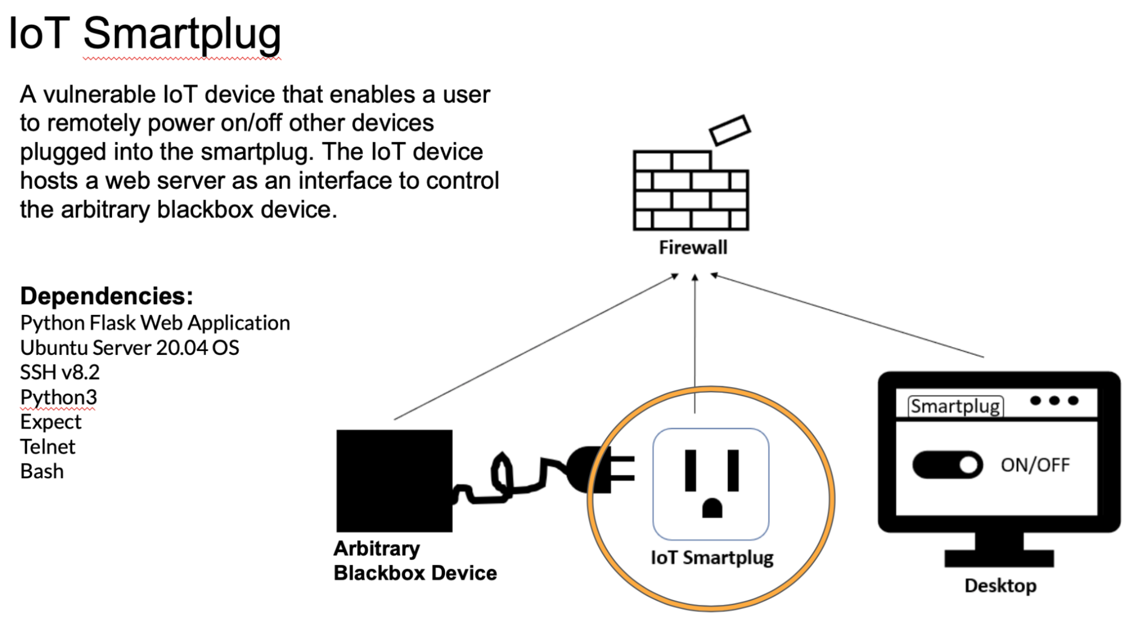


Figure 4.6: Diagram of blackbox device in our packaged CDC

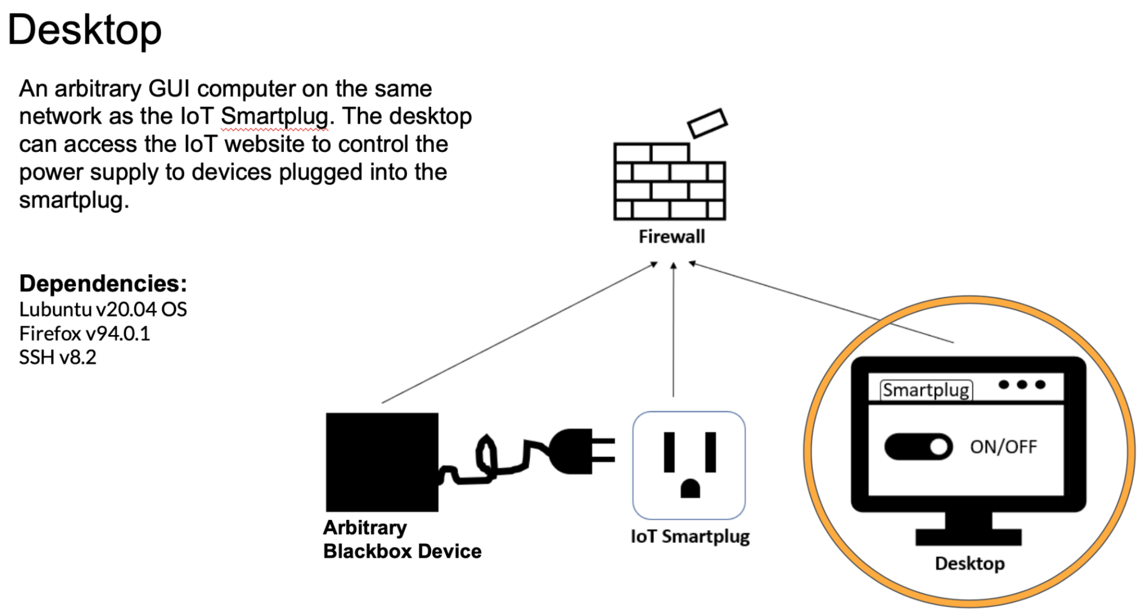


IoT Smartplug

A vulnerable IoT device that enables a user to remotely power on/off other devices plugged into the smartplug. The IoT device hosts a web server as an interface to control the arbitrary blackbox device.

Dependencies:
Python Flask Web Application
Ubuntu Server 20.04 OS
SSH v8.2
Python3
Expect
Telnet
Bash

Figure 4.7: Diagram of IoT Smartplug in our packaged CDC



Desktop

An arbitrary GUI computer on the same network as the IoT Smartplug. The desktop can access the IoT website to control the power supply to devices plugged into the smartplug.

Dependencies:
Lubuntu v20.04 OS
Firefox v94.0.1
SSH v8.2

Figure 4.8: Diagram of Desktop in our packaged CDC

5 Testing

5.1 Unit Testing

The “units” to be tested are the vulnerabilities we create. We must make a vulnerable IoT device that can be exploited and abused by an attacker. This means we must intentionally implement vulnerabilities to be exploited. It is paramount to ensure that the IoT device we build can successfully be exploited. We can use NMAP to ensure that our services are exposed from an attacker's perspective and vulnerable. (Ports 80, 22, 21, etc.) Since our project consists mainly of creating vulnerable IoT devices(virtual and physical), we will create tests to try and exploit these vulnerabilities. A bash script will be created to emulate a Red Team exploiting said vulnerabilities. The script will try to expose the following vulnerabilities:

- gaining sudo/root privileges through
  - SSH
  - Telnet
  - Default user credentials
- gaining access to unauthorized files
- packet sniffing for user credentials and passwords
  - automate process by using a packet analyzer(tcpdump) to search for plaintext user credentials and passwords

It is critical that we integrate our IoT machine into ISEAGE hosted by Iowa State, and then replicate that functionality into a VirtualBox virtual machine. We shall run scripts on each VM to ensure that each is susceptible to the exploits.

5.2 System Testing

We shall use NMAP and custom scripts that our team shall develop to verify our IoT device is exploitable. Our main mission is to develop exploitable virtual machines, and so we must verify we can identify and exploit these vulnerabilities. We can use other linux utilities such as tcpdump to capture the traffic generated by our hacking attempts to verify they were successful.

5.3 Regression Testing

We will use snapshots and testing branches to ensure any progress is not lost. We will create our designs and make changes as needed. If the changes break the functionality of our IoT devices or scripts and we cannot figure out how to fix the changes, we will resort back to the previous version.

5.4 Acceptance Testing

We will give our client graphics explaining how our devices work as well as live demonstrations. He will be able to see it for himself and give direct feedback on the requirements being met.

5.5 Security Testing (if applicable)

Our infrastructure shall be hosting vulnerable machines on people’s computers. It is imperative that we are able to host these vulnerable machines while not compromising the legitimate security of a user's computer. We must implement proper network security measures to secure our insecure devices.

5.6 Results

From the results, we should have virtual IoT devices with cybersecurity vulnerabilities that people can exploit while being contained within a secure environment. The vulnerabilities in the devices should also be mitigatable by the blue team of the CDC competition. They should be usable by the general public, specifically high schools and community colleges.

6 Implementation

Our team’s preliminary implementation plan starts with integrating our current virtual IoT device into ISEAGE 2.0’s environment. Once we have done this, we will create clear documentation on any manual configuring the CDC competitors will have to do in order to follow the main CDC scenario guidelines. We will then move on to developing the VirtualBox distributable CDC. This will consist of running our previously made setup script, creating services to be run on our virtual IoT device, and creating a Red Team script to exploit vulnerabilities.

7 Professionalism

7.1 Areas of Responsibility

How IEEE correlates to these Areas of Responsibilities:

1.) Work competence:
With a set of data, reasonable and honest estimates will be used when making claims/assumptions.

2.) Financial Responsibility:
It is the responsibility of engineers to prevent the risk of being approached and offered financial bribery. We must uphold strong moral codes, and lowering the risk of being bribed includes having well managed finances. In addition, many times engineers are commissioned or paid for services, and it is our responsibility to deliver products and services to our clients to fulfill their investments.

3.) Communication Honesty:
Be open about conflicts of interest.
Give accurate estimates of progress.

4.) Health, Safety, Well-Being:
Make decisions with the well being of the public as priority, and to disclose any information that endangers the safety of others.

5.) Property Ownership:
Avoid injuring other people's property.

6.) Sustainability:
Be open about factors that could harm the environment.

7.) Social Responsibility:
Help society at whole learn more about technology.

7.2 Project Specific Professional Responsibility Areas

1. Work competence:

This relates to our project as this area encompasses a lot of areas related to a professional work environment. These areas relate to upholding a certain level of professionalism, integrity, and honesty. This also relates to maintaining an efficient work schedule and meeting deadlines. Our team performance for this category: Medium. We have maintained a regular work schedule, but have had minor hiccups related to communication with our client. Since then, we have made progress in this area, and have a clear picture and path of our design.

2. Financial Responsibility:

This relates not to our project. We have no budget for our design, and our development process is to eliminate the need to buy physical IoT devices by virtualizing access to these IoT devices. Our team performance for this category: N/A.

3. Communication Honesty:

This relates to our project as we have weekly meetings with our client. Our team performance: Medium. We give weekly updates on our progress and what we plan on doing. We also get weekly feedback from our client about what he would like. We have experienced slight communication confusion in the past preventing a High score.

4. Health, Safety, Well-Being:

Nothing we are working on will realistically allow harm to come to someone. Our team score: N/A.

5. Property Ownership:

This applies to us as we work on the Client's physical IoT devices. There is some risk in this as we must install vulnerable software on his devices. We are being careful to make sure the devices may be hacked but not permanently disabled. Our team performance: High. We have this as a central focus as we work. We are enabling vulnerabilities, yet not allowing lateral compromise to legitimate services and functionality.

6. Sustainability:

Be open about factors that could harm the environment. Our team score: N/A. This does not relate to our project as we need not worry about consequences to the environment when we consume minimal resources to provide a functional project.

7. Social Responsibility:

This applies to our project as the primary purpose of our project is to educate the public on the importance of cybersecurity by providing a resource that is accessible to the general public. Our team score is: High. This is the entire purpose of our project. If we do not complete this goal our project is pointless.

7.3 Most Applicable Professional Responsibility Area

The area of professional responsibility that is most important to our project is social responsibility. Our project aims to improve the public understanding of IoT devices and the dangers surrounding them. In order to most effectively perform this responsibility we are focused on making our product user friendly with clear use instructions. With usability in mind while developing our project we have gained a focus on documentation that will be able to be utilized to better understand our process.

8 Closing Material

8.1 Discussion

The main results for our project will satisfy the clients functional and non-functional requirements. Our product will be accessible by both the ISU CDC for the Spring of 2022 in addition to high school students who wish to participate in a CDC without the hardware dependencies for ISEAGE. For our downloadable CDC, we shall include necessary documentation to enable custom configuration. We shall also include scripts to emulate a red team to test the defences put in place by the blue team.

8.2 Conclusion

In summary, this semester we have selected a smart plug as our virtual IoT device to be integrated into the CDC. We have also created a web interface to control our smart plug. Our team also successfully created a configuration/setup script to automatically install certain services and vuln

8.3 References

"IEEE 802.1qca-2015 - IEEE standard for Bridged Networks - Amendment 24," *IEEE SA - The IEEE Standards Association - Home*. [Online]. Available: https://standards.ieee.org/standard/802_1Qca-2015.html. [Accessed: 03-Dec-2021].

"ISO 9660 - ISO images for computer files," *ISO*, 24-Jun-2020. [Online]. Available: https://www.iso.org/iso-9660-images-for-computer-files.html. [Accessed: 03-Dec-2021].

"ISERink setup guide," *ISERink Setup Guide - ISERink Setup Docs 1.4 documentation*. [Online]. Available: https://docs.iseage.org/iserink/current/. [Accessed: 03-Dec-2021].

8.4.1 Team Contract

Team Name sdmay22-14

Team Members:
1) Jacob Foster
2) Jake Martin
3) Megan Litterer
4) Justin Tyler

a) Required Skill Sets for Your Project: (if feasible – tie them to the requirements)
Knowledge of the CDC (Blue Team, White Team, Red Team, etc.) and ISEAGE environment
Knowledge of common programming languages, Linux OS commands, bash scripting, and creation and configuration of Virtual Machines
Comprehension of basic cybersecurity concepts and how they relate to IoT devices
Documentation etiquette and effective communication abilities

b) Skill Sets Covered by the Team: (for each skill, state which team member(s) cover it)
Jacob: Cybersecurity Engineer: a thorough understanding of cybersecurity concepts, techniques, and tools. Experience with CDCs and ISEAGE.

Justin: Electrical engineer: Organization, C programming, public speaking

Jake: Computer Engineer: understanding of networking and cybersecurity concepts, Java and C programming, bash scripting, and Linux virtual machine configuration

Megan: Software Engineer: understanding of Javascript, Python, Java, and C programming, embedded systems, cyber security, diagram design, network

c) Project Management Style Adopted by the Team:
Scrum - Project managers using this style should clearly explain responsibilities and roles to team members, and foster ongoing communication between all members of the team.

d) Initial Project Management Roles: (enumerate which team member plays what role)
Jake Martin - Scripting Developer/Task Manager
Jacob Foster - IoT Engineer/Point of contact
Megan Litterer - IoT Website Developer/Team Website developer
Justin Tyler- Minutes Taker/Initial Tester

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
Wednesdays, 7:30pm Central Time, Face-toFace

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
Discord

3. Decision-making policy (e.g., consensus, majority vote):
Consensus, we will all come to a unanimous consensus

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
Justin will write down the minutes and share them through the shared team google doc.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
   Each member will be present at meetings (i.e., Team Meetings, Meeting with Client and TAs)

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
   Each team member shall accomplish their assigned tasks by their deadline. Team members shall give reasonable and accurate timelines, and hold each other accountable for meeting those deadlines.

3. Expected level of communication with other team members:
   Regular communication shall be expected. Namely, members should actively look at discord on Tuesdays for our meetings with the Client, and also during other meeting times.

4. Expected level of commitment to team decisions and tasks:
   We expect the members of this team to share a steadfast commitment to meeting the project requirements outlined by the team and our client. We expect that when decisions are made, the team stays committed to them.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
   Jake Martin - Team Leader - Maintains pace of the team's workflow
   Jacob Foster - Point of contact, Client Interaction
   Justin Tyler - Documentation
   Megan Litterer - Designer

2. Strategies for supporting and guiding the work of all team members:
   We shall develop weekly high-level tasks and then delegate each of those tasks to more specific tasks to individual members. Team members shall collectively agree on reasonable deadlines. Open and positive communication will be used if team members need to help others complete delegated tasks.

3. Strategies for recognizing the contributions of all team members:
   If it isn't documented, it cannot be recognized; The team must thoroughly document our work processes, and weekly achievements to keep us on track. We will use the delegated tasks to guide us in recognizing contributions of team members.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

   Jacob: Cybersecurity engineer, a thorough understanding of cybersecurity concepts, techniques, and tools. Experience with CDCs and ISEAGE.

   Justin: Electrical engineer: Organization, C programming, public speaking

   Jake: Computer Engineering: Network and Cybersecurity focus. Great leadership skills

   Megan: Software Engineer: Embedded systems, cyber security, diagram design, network (pending)

2. Strategies for encouraging and support contributions and ideas from all team members:
   Positive feedback, constructive criticism, in person meeting for support and questions. If team members need additional support, we as a team will try to be accommodating, flexible, and understanding.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
   Treat others the way we want to be treated; In the event a group or individual members have concerns about work processes, certain behaviors, etc, the team is responsible for creating an environment where these concerns can be freely expressed. Members are encouraged to express their concerns among the team, or escalate the situation to the teaching staff if necessary.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
   a. Study and understand how CDC and ISEAGE works
      i. Participate in Cyber Defence Competition
   b. Determine and resolve outstanding issues from previous teams
   c. Research and determine new vulnerable IoT device for implementation
   d. Design structure of new virtual IoT device
      i. Emulate a large number of virtual IoT devices, thus making it scalable

2. Strategies for planning and assigning individual and team work:
   a. Assign tasks and goals during weekly team meetings (put on trello)

3. Strategies for keeping on task:
   We shall develop weekly high-level tasks and then delegate each of those tasks to more specific tasks to individual members. Team members shall collectively agree on reasonable deadlines. Open and positive communication will be used if team members need to help others complete delegated tasks.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
   Any infractions shall be addressed at the team level.

2. What will your team do if the infractions continue?
   If these infractions continue, we shall escalate the problem to the teaching staff in an attempt to resolve the problem.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) __Megan Litterer_________________________________ DATE __9/15/21__________
2) __Justin Tyler____________________________________ DATE __9/15/21__________
3) __Jacob Foster___________________________________ DATE __9/15/21__________
4) __Jake Martin ___________________________________ DATE __9/15/21_________