Development of IoT Components for the CDC (Cyber Defense Competition)

Team: Jacob Foster, Megan Litterer, Jake Martin, Justin Tyler

Client: Daji Qiao/CDC staff

Advisors: Daji Qiao and Doug Jacobson

Background Information

- **IoT Device** - “Internet of Things” device, connects to the internet. People are unaware of the dangers which accompany the lack of cybersecurity in IoT devices

- **CDC** - Cyber Defense Competition
  - A student competition where the objective is to secure vulnerable devices
  - 2 Main Teams:
    - Blue - students, given vulnerable devices to fix
    - Red - hacks Blue Team’s devices

- **Problem -** Adding physical IoT devices to a CDC is expensive and time consuming. Each blue team requires their own IoT device which creates unnecessary overhead.

Project Vision

- We are creating a vulnerable, virtual IoT device to implement in the Spring ISU CDC

- The purpose of our project is to raise awareness of the dangers surrounding IoT devices
  - 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
  - Exploiting vulnerabilities will enable software developers to program with a security-focused mindset

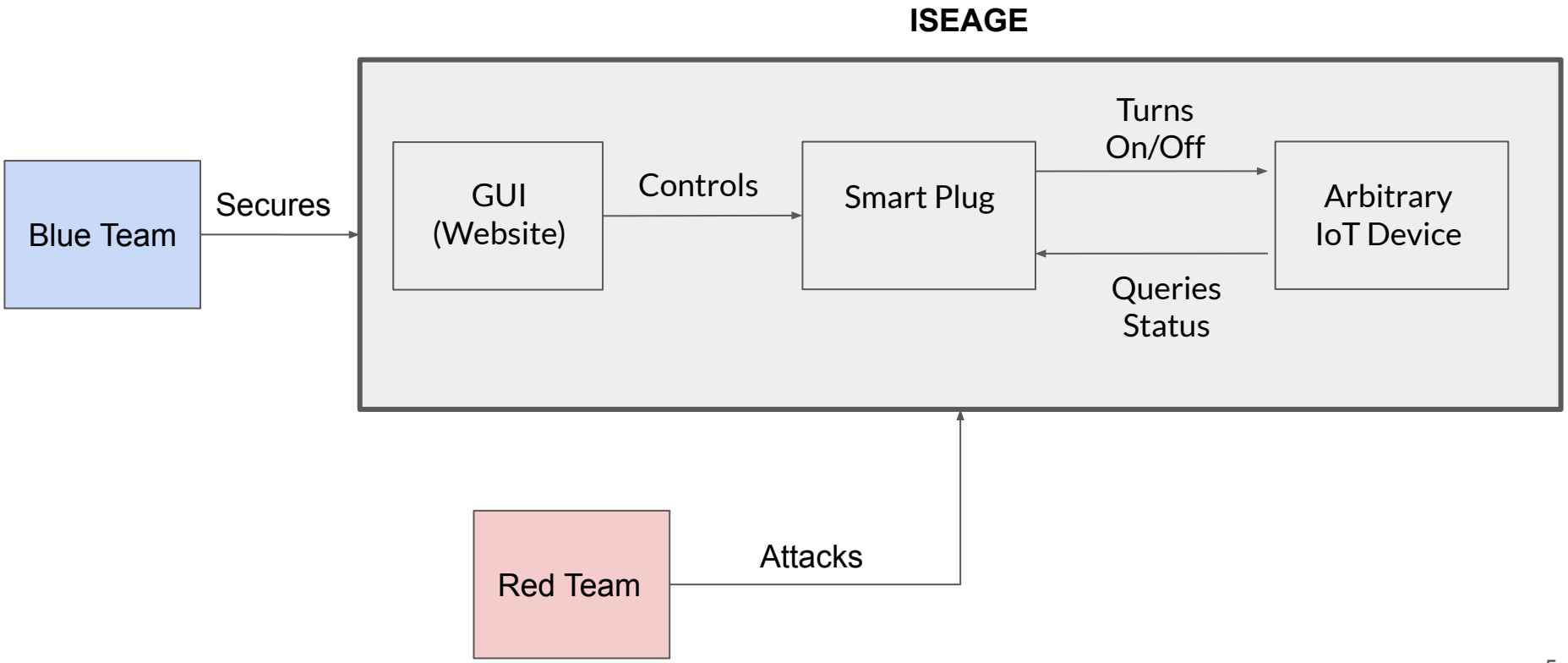
Functional Requirements

- **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

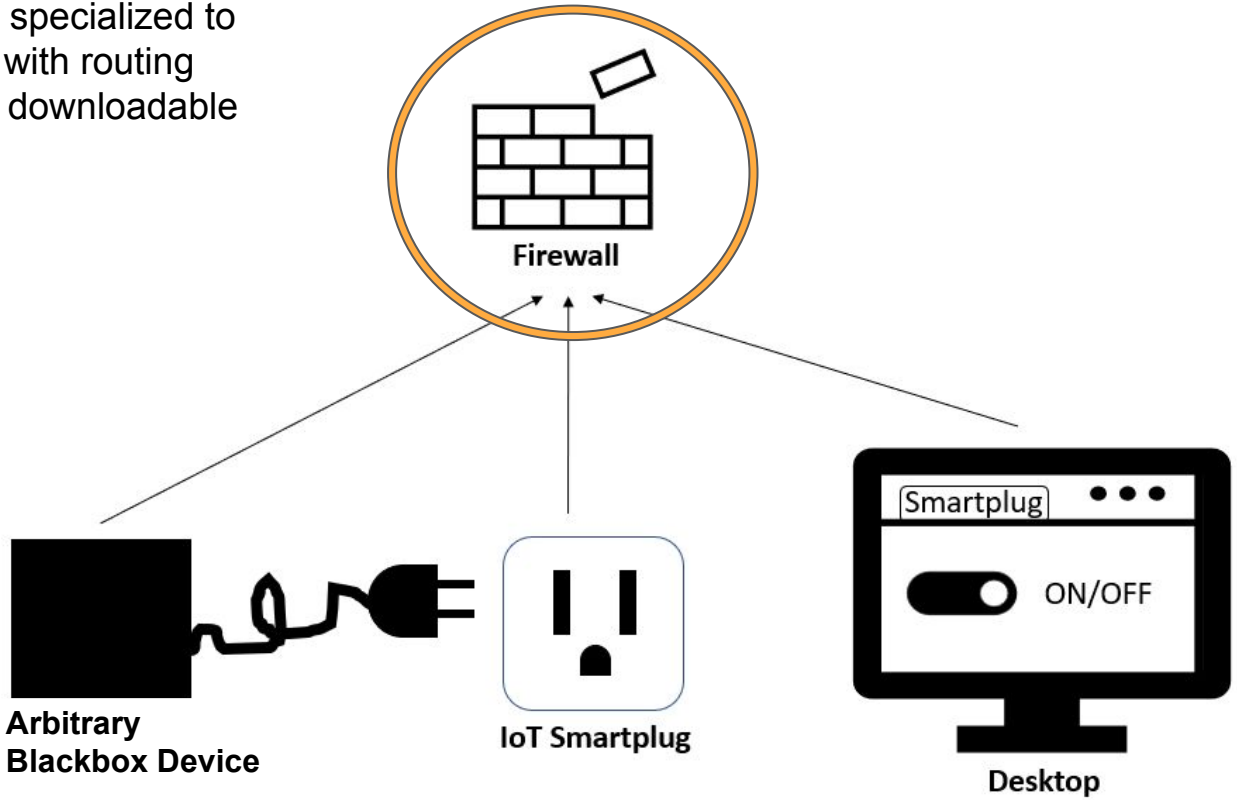
Conceptual Design Diagram



Firewall

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

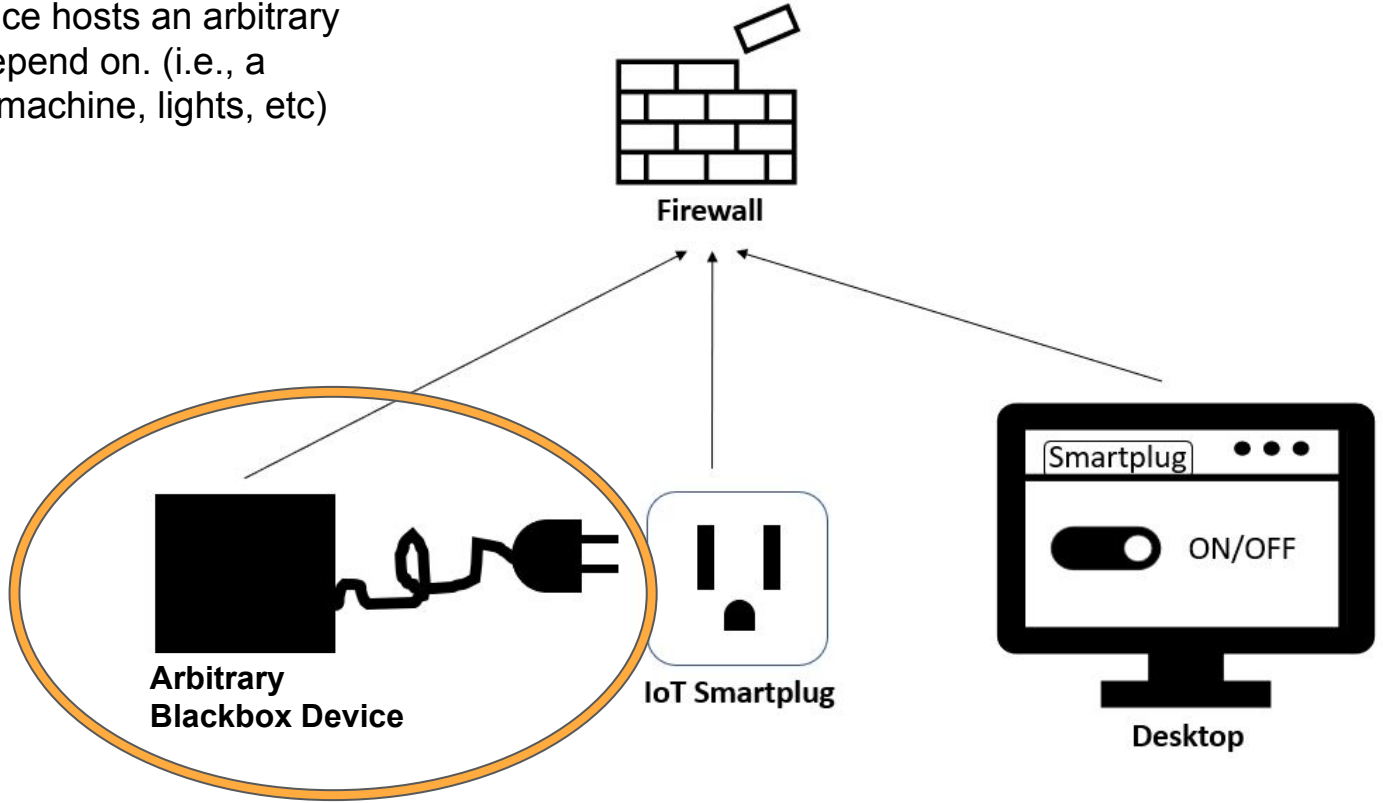
Dependencies:
pfSense 2.5.1 OS



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)

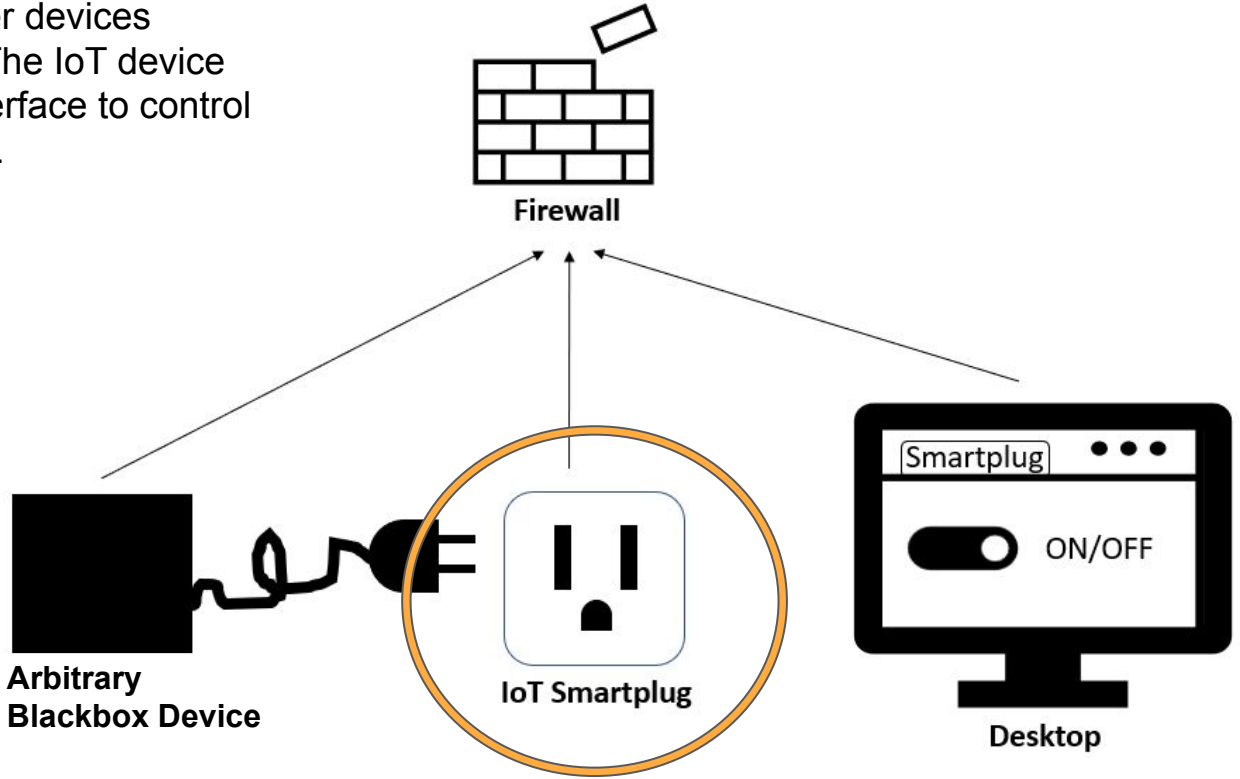
Dependencies:
Any linux-based OS



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.

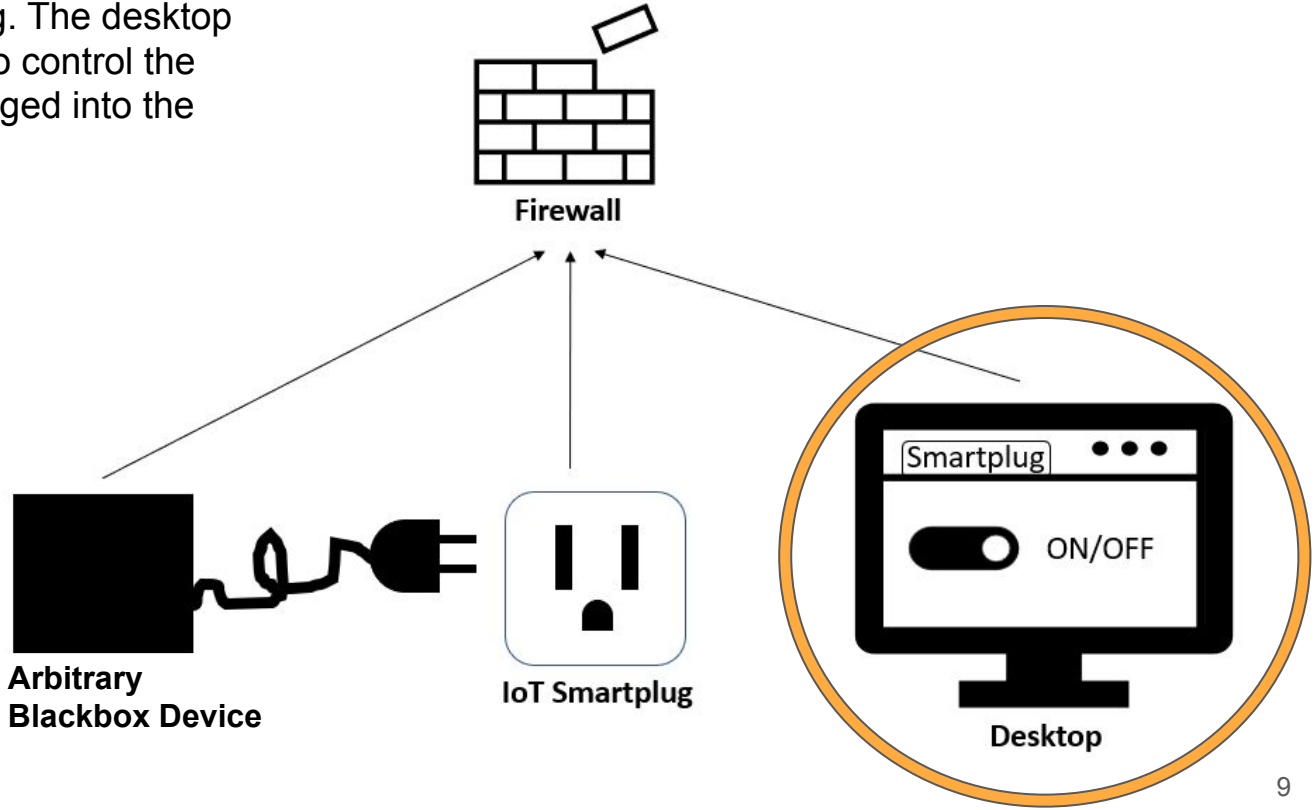
Dependencies:
Python Flask Web Application
Ubuntu Server 20.04 OS
SSH v8.2
Python3
Expect
Telnet
Bash

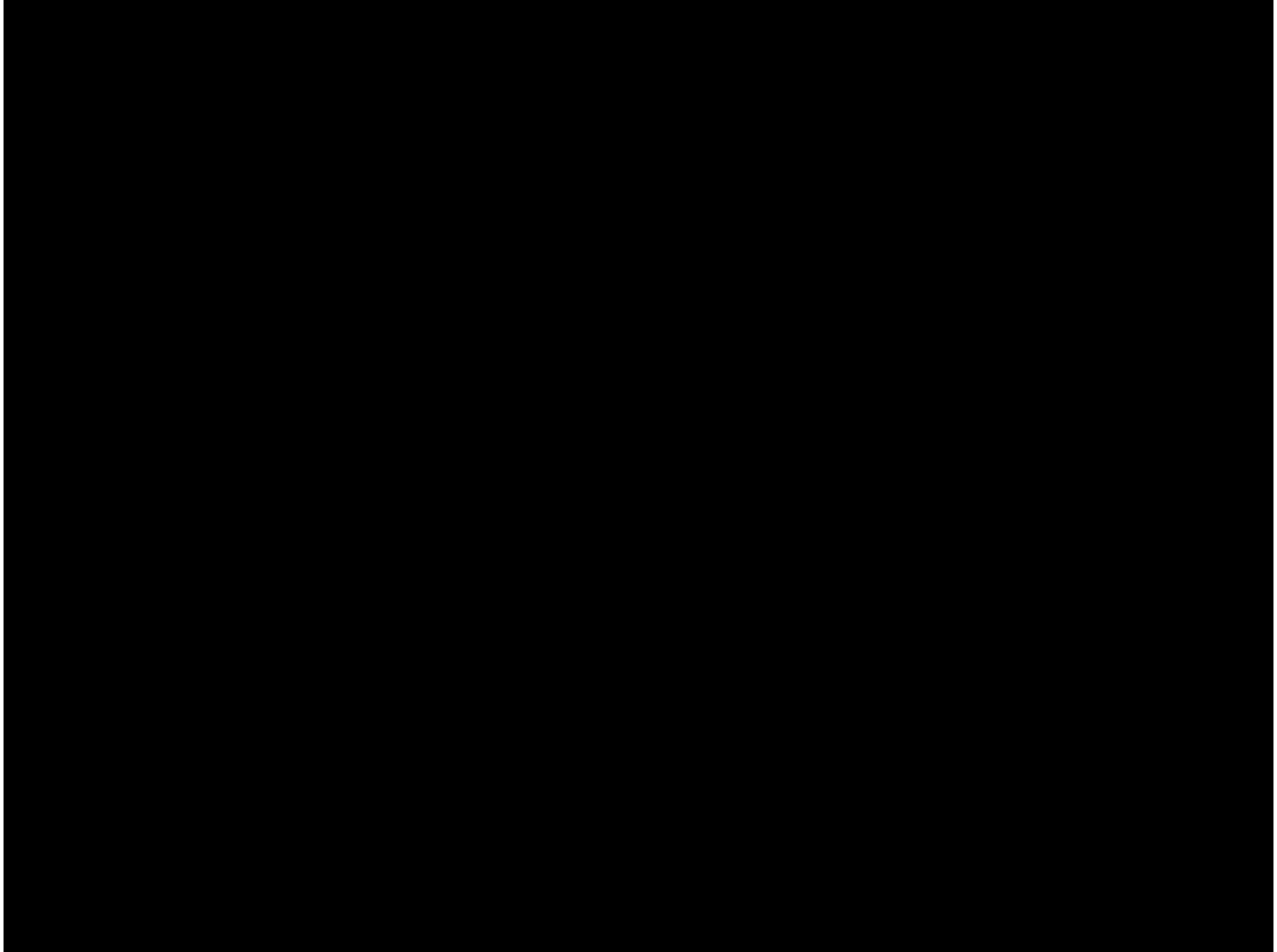


Desktop

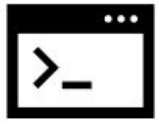
An arbitrary GUI computer on the same network as the IoT Smartplug. The desktop can access the IoT website to control the power supply to devices plugged into the smartplug.

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





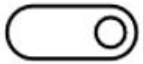
POST http://192.168.1.5:8000/enable



./enableScript

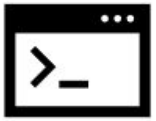
```
#!/usr/bin/expect

#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 e
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
```

ON



OFF

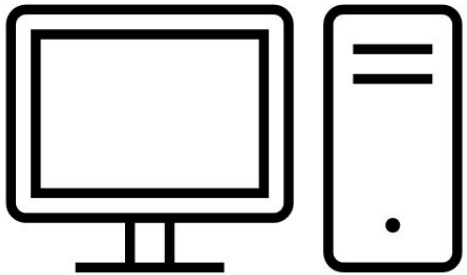
POST http://192.168.1.5:8000/disable



./disableScript

```
#!/usr/bin/expect

#Change variables as needed
set hostIP 192.168.1.206
set password "12345678"
set 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
```

Design Problem



VirtualBox

> apt update & apt update
> apt install ssh
> nano /etc/network/interfaces
> passwd root
> chmod 777 /etc/passwd
...
....
> apt install ftp
> echo “nc –l localhost:50073”
> mkdir /temper/

ISEAGE

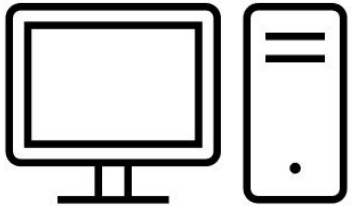
> apt update & apt update
> apt install ssh
> nano /etc/network/interfaces
> passwd root
> chmod 777 /etc/passwd
...
....
> apt install ftp
> echo “nc –l localhost:50073”
> mkdir /temper/

Design Solution

```
#!/bin/bash
apt update & apt update
apt install ssh
nano /etc/network/interfaces
passwd root
chmod 777 /etc/passwd
...
....
apt install ftp
echo “nc –l localhost:50073”
mkdir  /temper/
```

ISEAGE

> git clone https://sdteam14/vm.git
> ./setup-script.sh



VirtualBox

> git clone https://sdteam14/vm.git
> ./setup-script.sh

Project Plan

Due to the quantity and complexity of deliverables, we risk not finishing all of our assignments. This is mitigated through task prioritization. If time is insufficient, we shall ensure the most vital tasks are completed

1. Smartplug IoT Virtual Machine
2. CDC Implementation and Documentation
3. VirtualBox distributable CDC bundle
    - Three virtual machines
    - Red team emulation scripts
4. Previous team’s IoT projects

Project Plan - Schedule/Milestones

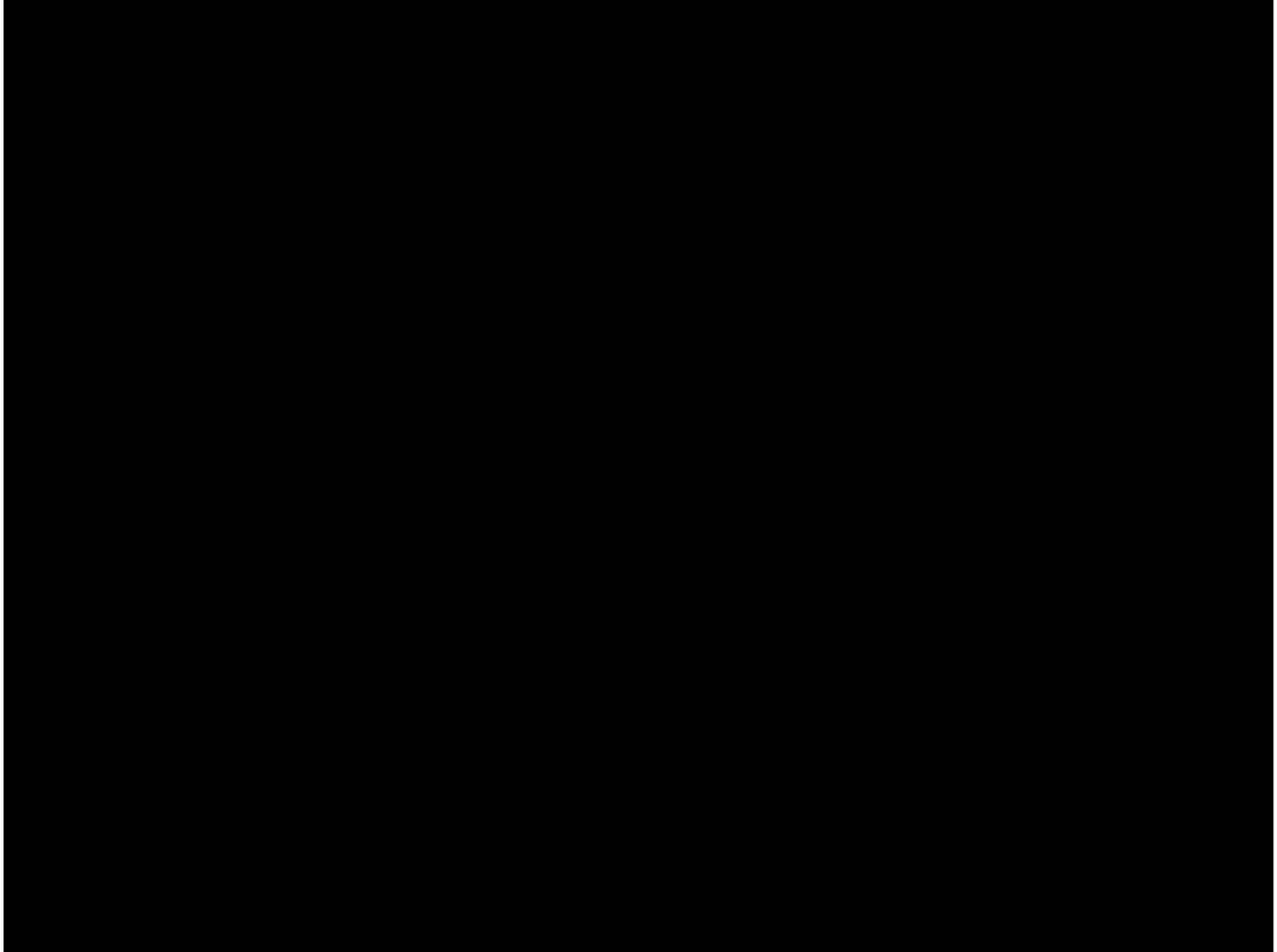
- **Virtual device (Complete by early 2nd semester)**
  - Script to set up IoT device
  - Website interface for IoT device
  - Documentation for CDC
  - Integrated into ISEAGE 2.0
- **VirtualBox distributable CDC (Complete by mid of 2nd semester)**
  - Use previously made script to initially set up VM
  - Create service for the IoT VM (i.e., online game)
  - Red team script to exploit vulnerabilities
- **Continue last year's IoT projects (Complete by end 2nd semester)**
  - Follow previous team's documentation to get the camera to function
  - Follow previous team's instructions to hopefully get device to work
  - If unable to get device to work, redo the device

Test Plan - Unit Tests

Unit tests are composed of scripted exploits for our vulnerabilities

- **Smartplug Device**
  - SSH bruteforce
  - Telnet bruteforce

- **Smartplug Website**
  - Broken access control
  - Broken authentication
  - Unencrypted/plaintext login credentials
  - Sensitive data exposure
  - Replay attacks



Design complexity

- **Develop IoT device for two different environments**
  - ISEAGE vs VirtualBox
- **Previous team's documentation expected in depth knowledge of firmware**
  - Made assumptions on software experience
  - Skipped set-up steps
  - Lacked troubleshooting for potential errors
- **Wide variety of expected tasks**
  - Finish previous team's virtual device
  - Run previous teams' physical device
  - Create our own virtual device

Conclusions

- **Current progress**
  - About halfway done implementing our virtual device and the red team script

- **Next semester**
  - Finish our virtual device and red team script
  - Begin working on physical devices

- **Individual contributions**
  - Few specific roles, more of a “everyone does everything”
  - Megan - handled team website
  - Jake - pioneered scripting
  - Jacob - started up the VirtualBox implementation
  - Justin - began troubleshooting previous team’s virtual device

Development of IoT Components for the CDC

Jacob Foster, Megan Litterer, Jake Martin, Justin Tyler