

CSI OT 3D Platform Cyber Attack Demonstration Control Website

Design Manual

VERSION: CORPLAB-2019-T3.1-P2

Prepared by

Liu Yuancheng

Senior Security Development Engineer

<yuancheng.liu@trustwave.com>

Wong Jun Wen

Asst R&D Manager

<junwen.wong@trustwave.com>

Dr. Shantanu Chakrabarty

NUS Research Fellow

<shantanu1088@gmail.com>

Confidentiality Notice: This document is confidential and contains proprietary information and intellectual property of Trustwave, a Singtel company, its parent company and its subsidiaries. Neither this document nor any of the information contained herein may be reproduced or disclosed under any circumstances. Please be aware that disclosure, copying, distribution or use of this document and the information contained therein is strictly prohibited.

**CSI OT 3D Platform Cyber Attack Demonstration Control Website Design and Usage Manual**

**Introduction**

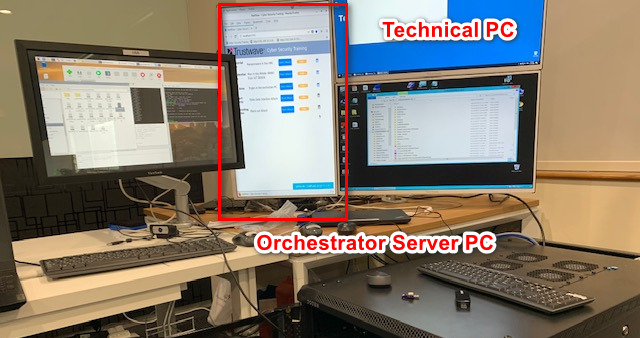
The “CSI OT 3D Platform Cyber Attack Demonstration Control Website” is a web interface dashboard to let the user to control different cyber-attack demos on OT 3D railway module platform and check the attack feedback during the presentation. This manual will introduce the web host program workflow, program structure and the usage of the webpage which shown on the Orchestrator PC.

Website View:



* Open a web browser on Orchestrator PC and type in link: <http://localhost:8080/> .

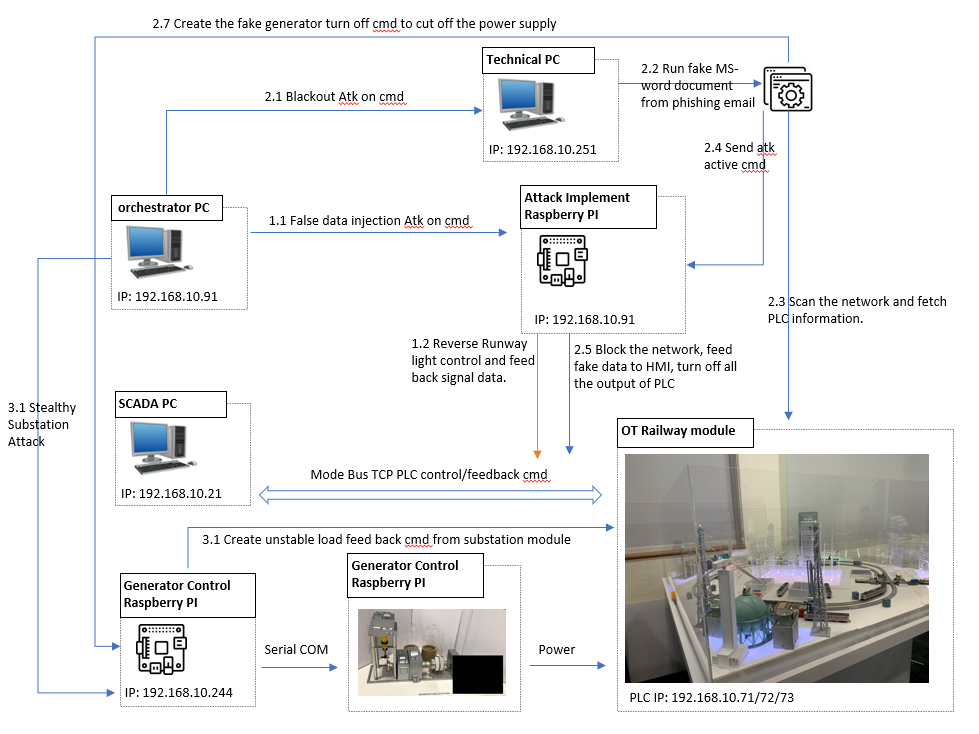
Orchestrator PC View:



* Orchestrator PC Login refer to doc < CSI OT 3D Platform Cyber Attack Demonstration User Manual.pdf >

**Web Host Program Workflow**

Control Flow Diagram of the System:



As shown in the above diagram, the Web host program will communicate with the Attack Raspberry PI (section 1.1) and the Generator Control Raspberry PI ( section 3.1) by UDP. When the user pressed the “Start” attack button, the web host will send the attack activation command to the related agent running on technical PC or the attack device. These are the action trigger cmd for different attack when use press the ‘Start Attack’ or ‘Stop Attack’ button (After you press the ‘Start Attack’ button, the text on the button will change to ‘Stop Attack’ which means the attack is under progress). The control cmd format is **<ctrlTag>;<parameter>**.



False Data Injection Attack:

* UDP Channel: Orchestrator PC (192.168.10.90) => Attack Raspberry PI (192.168.10.91) Port [5005]
* UDP message: Start Attack: '**A;2**' Stop Attack: ‘**A;0**’

Black Out Attack:

* UDP Channel: Orchestrator PC (192.168.10.90) => Technical PC (192.168.10.251) Port [5006]
* UDP message: Start Attack: '**A;2**' Stop Attack: ‘**A;0**’

Stealthy Substation Attack:

* UDP Channel: Orchestrator PC (192.168.10.90) => Generator Control Raspberry PI (192.168.10.244) Port [5005]
* UDP message: Start Attack: '**A;3**' Stop Attack: ‘**A;0**’

**Website Host Program Setup**

Development environment: NodeJs(v12.18.4)/JavaScript HTML5

Additional lib/software need:

* NodeJs(v12.18.4) (Download and Install NodeJs : https://www.guru99.com/download-install-node-js.html)

Hardware needed: Orchestrator PC with Centos 7

Program execution cmd: Cd to the "server" folder, run cmd **Node app.js**

First time setup the website host program on a computer:

Copy the server folder to the computer and setup the auto run service during the system boot up:

1. Create a new service unit file at **/etc/systemd/system/httpserver.service** with below content. The name of the service unit is user defined and can be any name of your choice.

[Unit]

Description=server side for GUI

After=network.target

[Install]

WantedBy=multi-user.target

Alias=webservice.service

[Service]

Type=simple

User=root

Group=root

# Start main service

ExecStart=/home/orchestrator/Attack/server/server/node /home/orchestrator/Attack/server/server/app.js

# Give up if ping don't get an answer

TimeoutSec=20

Restart=always

2.Reload the systemd process to consider newly created httpserver.service OR every time when httpserver.service gets modified:

**# systemctl daemon-reload**

3. Enable this service to start after reboot automatically:

# **systemctl enable httpserver.service**

4. Start the service:

# **systemctl start httpserver.service**

5. Reboot the host to verify whether the scripts are starting as expected during system boot.

# **systemctl reboot**

**Program File Structure**

All the source code are under the server folder:

You can clone the source code from: <https://github.com/LiuYuancheng/OT_Platform_Attack_Web>

| **Program File/Folder** | **Execution Env** | **Description** |
| --- | --- | --- |
| node\_modules [Folder] | Node.js(JavaScript) | All the additional module which needed to import in the app.js |
| Public [Folder] | HTML, CSS, JavaScript | The main web page interface. |
| app.js | Node.js(JavaScript) | Main server program. |
| httpserver.service | sh | Auto run setup when running on Linux Platform. |
| playAlert.sh | sh | Play the alert sound when the attack happens. |
| runServer.sh | sh | Run the app.js and stop the duplicated running if found. |

End (last edited 22/02/2021)