CSI OT 3D Platform Cyber Attack Demonstration Control Website

Design Manual

VERSION: CORPLAB-2021-T5F-A1

30/03/2021

Prepared by

Liu Yuancheng
Senior Security Development Engineer
yuancheng.liu@trustwave.com

Wong Jun Wen
Asst R&D Manager
junwen.wong@trustwave.com

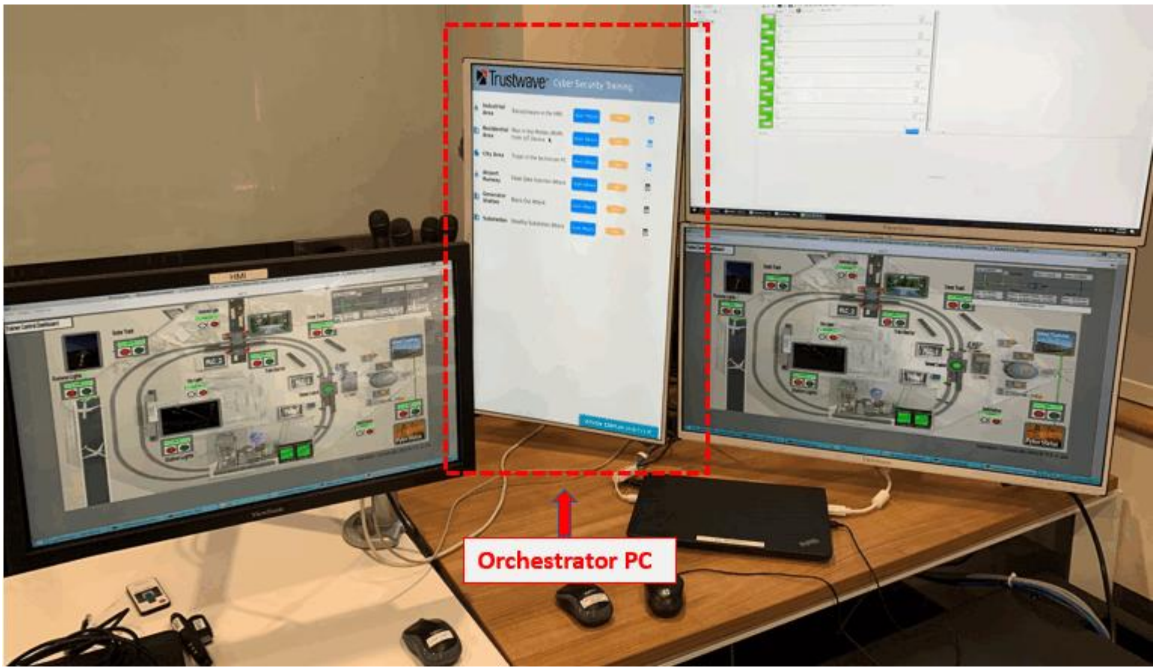
Dr. Shantanu Chakrabarty
NUS Research Fellow
shantanu1088@gmail.com

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

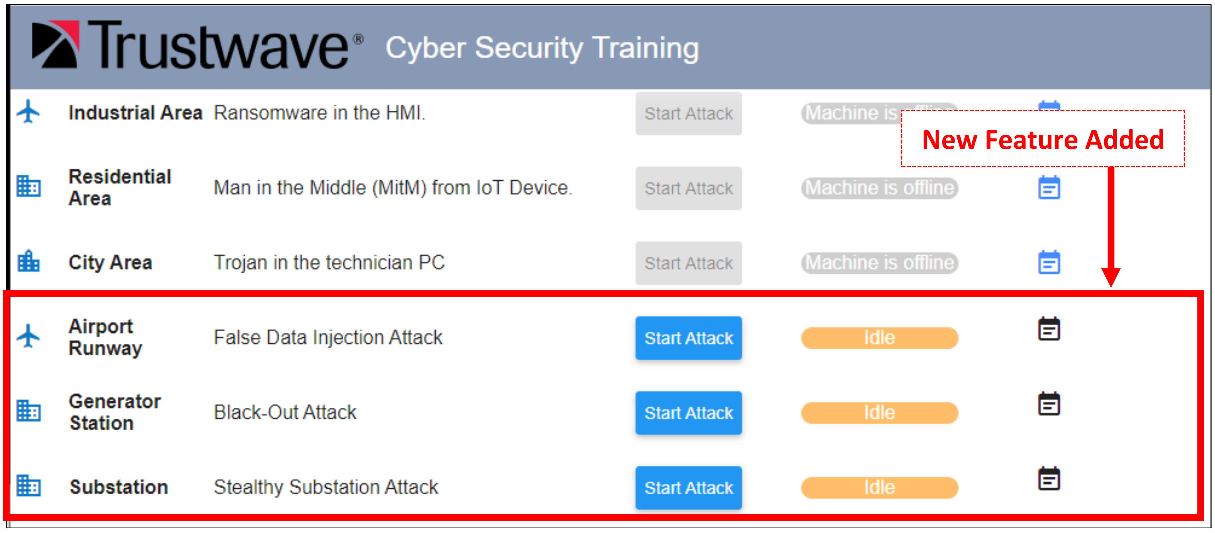
1. Project Introduction

The “CSI OT 3D Platform Cyber Attack Demonstration Control Website” is a web interface dashboard to let the user to start and stop different cyber-attack demos on OT 3D platform and check the attack feedback during the presentation. This manual will introduce the web host program detail workflow, program structure and the usage of the webpage which shown on the Orchestrator PC (figure1.0).



<Figure_1.0 Orchestrator PC View >

The presenter will use the “Orchestrator PC” in the system to access the webpage, Orchestrator PC login part refer to doc < CSI OT 3D Platform Cyber Attack Demonstration User Manual.pdf >. Open a web browser on Orchestrator PC and type in URL: http://localhost:8080/.(Figure_1.1)



Trustwave® Cyber Security Training

✈	**Industrial Area**	Ransomware in the HMI.	Start Attack	Machine is	📅
🏢	**Residential Area**	Man in the Middle (MitM) from IoT Device.	Start Attack	Machine is offline	📅
🏢	**City Area**	Trojan in the technician PC	Start Attack	Machine is offline	📅
✈	**Airport Runway**	False Data Injection Attack	Start Attack	Idle	📅
🏢	**Generator Station**	Black-Out Attack	Start Attack	Idle	📅
🏢	**Substation**	Stealthy Substation Attack	Start Attack	Idle	📅

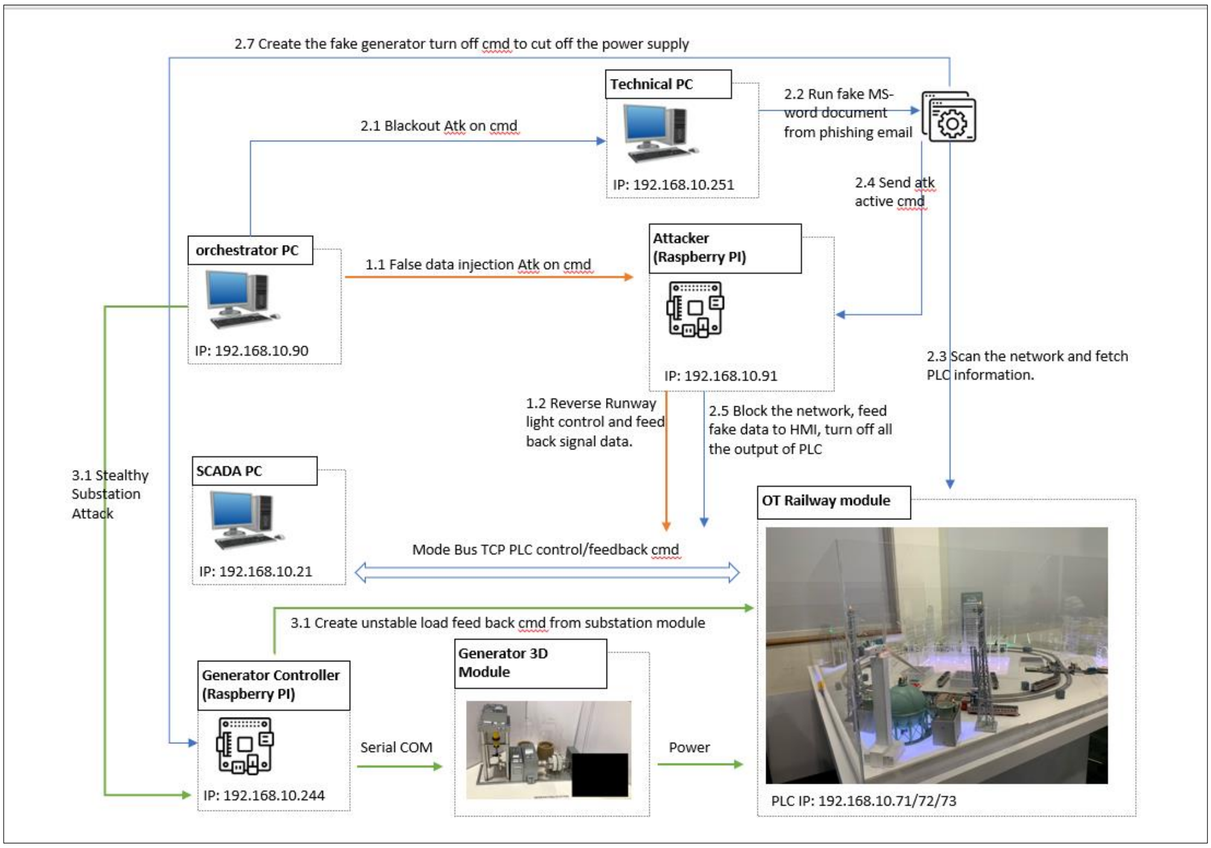
New Feature Added

<Figure_1.1 Cyber Security Training Webpage>

2. Web Host Program Workflow

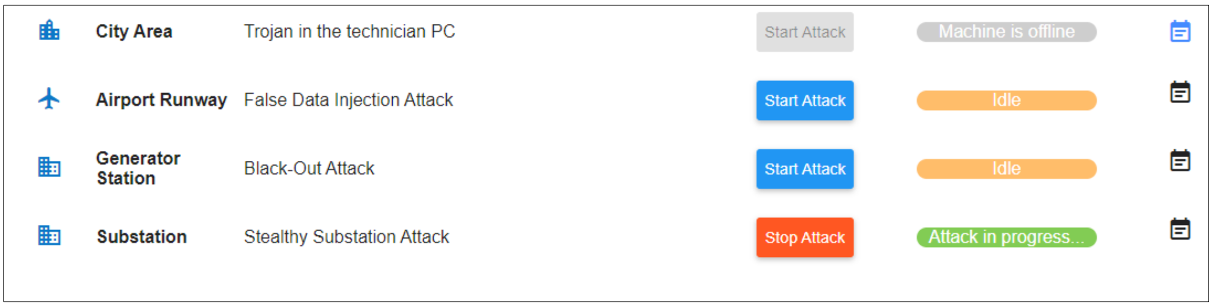
2.1 System Control Flow

In our system, the cyber-attack action is implemented the by an attack device (Raspberry PI), the web host will send the start and stop command to the attack device through UDP [port 5005/5006]. The Control Flow Diagram of the System is shown below (Figure_2.1):


2.7 Create the fake generator turn off cmd to cut off the power supply
Technical PC
2.1 Blackout Atk on cmd
2.2 Run fake MS-word document from phishing email
IP: 192.168.10.251
2.4 Send atk active cmd
orchestrator PC
1.1 False data injection Atk on cmd
Attacker (Raspberry PI)
IP: 192.168.10.90
IP: 192.168.10.91
2.3 Scan the network and fetch PLC information.
1.2 Reverse Runway light control and feed back signal data.
2.5 Block the network, feed fake data to HMI, turn off all the output of PLC
3.1 Stealthy Substation Attack
SCADA PC
OT Railway module
IP: 192.168.10.21
Mode Bus TCP PLC control/feedback cmd
3.1 Create unstable load feed back cmd from substation module
Generator 3D Module
Generator Controller (Raspberry PI)
Serial COM
Power
IP: 192.168.10.244
PLC IP: 192.168.10.71/72/73

<Figure_2.0 System Control Flow Diagram >

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 through. When the user pressed the "Start Attack" button, the web host will send the attack activation command to the related agent program running on technical PC or the attack device. After you press the 'Start Attack' button, the text on the button will change to 'Stop Attack' which means the attack has been activated. The current system state will also show in the progress bar behind the attack control button, as shown in the below image Figure_2.1, the program shows the stealthy substation attack is under progressing.



City Area	Trojan in the technician PC	Start Attack	Machine is offline
Airport Runway	False Data Injection Attack	Start Attack	Idle
Generator Station	Black-Out Attack	Start Attack	Idle
Substation	Stealthy Substation Attack	Stop Attack	Attack in progress...

<Figure_2.1 Start and stop the attack>

2.2 Attack Control Command

All the action trigger commands are sent through UDP under format **<ctrlTag>;<parameter>** with UTF-8 byte encode. Attack control tag is "A" and the parameter [1, 2,3] specify starting different attack, 0 for stopping attack.

2.2.1 False Data Injection Attack

UDP Channel: Orchestrator PC (192.168.10.90) => Attack Raspberry PI (192.168.10.91) [Port 5005]

UDP Sender program: Orchestrator PC [/home/orchestrator/Attack/server/server/App.js]

UDP Receiver program: Attack Raspberry PI [/Home/remoteAtk/attackServ.py]

UDP message: Start Attack: '**A;2**' Stop Attack: '**A;0.**'

2.2.2 Black Out Attack [From **App.js** to **actionServer.py**, port:5006]:

Step 1 Open the word doc:

UDP Channel: Orchestrator PC (192.168.10.90) => Technical PC (192.168.10.251) [port 5006]

UDP Sender program: Orchestrator PC [/home/orchestrator/Attack/server/server/App.js]

UDP Receiver program: Technical PC [C:\Admin\Documents\remoteAtk\actionServer.py]

UDP message: Start Attack: '**A;2**' Stop Attack: '**A;0**'

Step2 Start the attack:

UDP Channel: Technical PC (192.168.10.251) => Attack Raspberry PI (192.168.10.91) [Port 5005]

UDP Sender program: Orchestrator PC [C:\Admin\Documents\remoteAtk\attackHost.py]

UDP Receiver program: Attack Raspberry PI [/Home/remoteAtk/attackServ.py]

UDP message: Start Attack: '**A;2**' Stop Attack: '**A;0**'

2.2.3 Stealthy Substation Attack:

UDP Channel: Orchestrator PC (192.168.10.90) => Generator control Raspberry PI (192.168.10.244) [port:5005]

UDP Sender program: Orchestrator PC [/home/orchestrator/Attack/server/server/App.js]

UDP Receiver program: Attack Raspberry PI [/Home/GenMgr/src /attackServ.py]

UDP message: Start Attack: '**A;3**' Stop Attack: '**A;0.**'

2.3 Webpage HTTP Command

The attack control from webpage to web host program is using HTTP get request: WebPage's JavaScript(server/public/js/index.js) => WebHost (server/App.js)

2.3.1 False Data Injection Attack

Attack start HTTP GET request: "/actions/falseInj_Attack/startAttack".

Attack stop HTTP GET request: “/actions/falseInj_Attack/stopAttack”.

2.3.2 Black Out Attack

Attack start HTTP GET request: “/actions/ blackOut_Attack/startAttack”.

Attack stop HTTP GET request: “/actions/ blackOut_Attack/stopAttack”.

2.3.3 Stealthy Substation Attack:

Attack start HTTP GET request: “/actions/ sub_Attack/startAttack”.

Attack stop HTTP GET request: “/actions/ sub_Attack/stopAttack”.

3. Website Host Program Setup

3.1 Program execution environment configuration

3.1.1 Development environment:

NodeJs(v12.18.4)/JavaScript HTML5

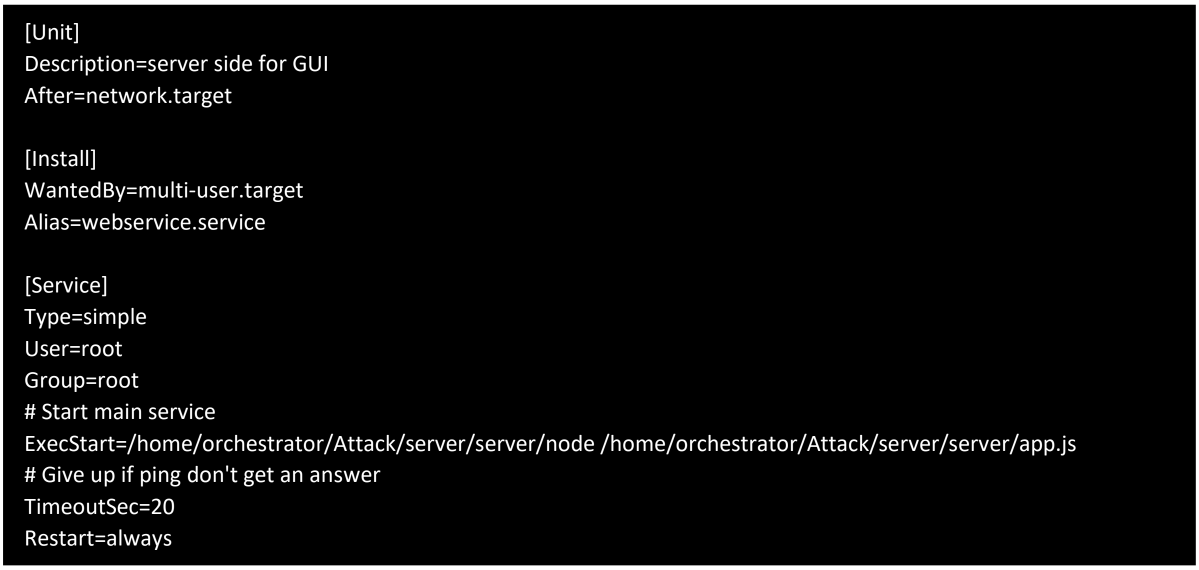
3.1.2 Additional lib/software need:
NodeJs(v12.18.4) (Download and Install NodeJs : https://www.guru99.com/download-install-node-js.html)

3.1.3 Hardware needed:
Orchestrator PC with Centos 7

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

3.2 First time setup the website host program on a computer:

3.2.1 Copy the server folder to the computer and setup the auto run service during the system boots up: 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
```

<Figure_3.2 httpserver.service file detail>

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

```
# systemctl daemon-reload
```

3.2.3 Enable this service to start after reboot automatically:

```
# systemctl enable httpserver.service
```

3.2.4 Start the service:

```
# systemctl start httpserver.service
```

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

```
# systemctl reboot
```

After finishing the above step, server will run automatically when the computer was powered on.

4. Program File Structure

4.1 Program File List

All the source code are under the server folder:

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.

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

5. Reference

N.A

End (last edited 10/05/2021)