**Graphical user interface, application

Description automatically generated**

**NCL CTI Report Analyzer Visualization Intern Assignment [06/05/2022]**

**Project Description**

This assignment is a sub project (visualization) of NCL “Towards Automated and Large-scale Cyber Attack Reconstruction with APT Reports” (CTI Report Analyzer) Project. The CTI Report Analyzer is aimed to provide an automated platform for researchers and analysts to expedite their understanding and significantly reduce their turnaround time in addressing cyberthreats.

The CTI Report Analyzer Visualization will create App and Web based UI for NCL customer use and monitor the process to convert the rich details found in CTI reports to reconstruct a dynamic environment. It is a group programming project which aims to let National Cybersecurity R&D Laboratory (NCL) interns can pick up the knowledge about CTI report, API events, python UI development and webpage design. Then create a Application and Web which NCL user can use it to control and monitor their CTI report analysis process. (As shown below)

Diagram

Description automatically generatedProject type: Program Visualization, group project

Project workload: 3 day/ week, total 12 week.

1. **Assignment Introduction**

**1.1 Assignment background**

National Cybersecurity R&D Lab (NCL) was established in 2015 and funded under the National Cybersecurity R&D (NCR) Programme. NCL is providing support to the Singapore Cybersecurity R&D Community in terms of their R&D, research experimentation and testing requirements. One of NCL business service is providing the provide an automated platform for researchers and analysts to expedite their understanding and significantly reduce their turnaround time in addressing cyberthreats.

Cyber Threat Intelligence (CTI) reports are valuable sources that researchers and analysts seek to have a deeper understanding of the current APT activities and the cyberthreat landscape. These reports are used to obtain insights of vulnerabilities and their associated attack techniques.

The CTI report analyzer UI project is aimed to provide two kinds of user interface which also NCL customer can directly control and monitor their CTI report analysis progress. The program workflow is shown below:

Diagram

Description automatically generated

The CTI report Analyzer UI contents 7 main modules

1. CTI Report Analyzer UI App Module: Main program running on user’s local comptuer to init other module with individual threading.
2. Data Manger Module: Process all the input data checking, converting and pre-processing for the CTI report user uploaded.
3. Application UI module: The main UI module user to upload the CTI report, Config the report analysis parameters, monitor report analysis process and check the result.
4. Communication manager module: The communication module to handle the data updating and data transfer (such as report upload).
5. Web page UI module: the Web page with the same function as the Application UI module.
6. Web host module: The Web Host program to handle the user control on the UI and provide same function as the data manager.
7. Control Hub Adapter: The main control hub use to collect data from the CTI report analyzer and the data base.

In this assignment the Intern are expected to implement the “Application UI”, “Webpage UI” module, “Web host” modules.

**1.2 Related knowledge and reference doc**

The related knowledge needs to learn for the project:

* CTI report
* Python user interface programming (wxpython)
* HTML, CSS, javascript
* Python web host programming (flask)

Needed document:

* NDSS2022Poster\_paper\_37\_final.pdf

1. **Assignment Main Task**

**2.1 Task 1: VM image testing.**

This task is aiming to test all 36 public VM images in NCL OpenStack [Beta] platform and create a VM image manual. For each VM image, below contents are needed to list down:

1. VM type: Linux/Windows. (32bit/64bit)
2. VM OS + version+ UI state: (ubuntu 18.04, CentOS)
3. VM image CPU min limit config.
4. VM image RAM min limit config
5. VM image Hard disk min limit config.
6. VM remote access config. (Such as SSH/RDP/VNC enabled?)
7. VM login detail. (username/password)
8. VM network interface config detail. (Such as network interface number, Ipv4 config: DHCP/Fix static)
9. VM image lib/tool included: (openssl, python3.x, docker, g++ ...)

By checking the list for all the public VM, the customer can find the correct image with the login detail based on their requirement instead of sending message to NCL Biz-Team to ask these basic questions such what the VM log in password.

Expected workload: 1.5 day/week, total 10 weeks.

**2.2 Task 2: VM image creation.**

After the intern has got familiar about the testing, they can try to create some VM images, such as some lite Linux OS for IOT:

1. raspberry pi OS: raspbian 32bit/64bit
2. raspberry pi OS: noobs 32bit
3. raspberry pi OS: Ubuntu 16/18/Lite
4. raspberry pi OS: CentOS 6/7
5. BeagleBone Black OS: Debian-C
6. BeagleBone Black OS:  fedora

The intern will test the customized VM image upload and usage part in OpenStack[Beta] and create a report about the detail steps for how to build these images. The created VM images can be used in the future OT/IOT security testbed.

Expected workload: 0.5 day/week, total 6 weeks.

**2.3 Assignment Final Goal**

After finished the assignment, the Intern student need to provide below document and program:

1. Provide an Intern assignment proposal/timeline plan.
2. Provide a VM image manual/dictionary to listed down all the VM detail information.
3. Provide a Linus shell script to list down all the all the information about the OS/VM.
4. Provide a Windows cmd script file to list down all the information about the Windows VM.
5. Provide more than 3 embedded system VM image.
6. Provide a manual about building a VM image.
7. Provide a general report and presentation about the task implement.
8. [Optional] Provide a knowledge sharing doc to share with other intern about the learning experience.

The Intern need to finish and submit all these files for project evaluation:

1. Improved assignment introduction doc: OpenStackVM\_Testing\_Intern.docx
2. Project progress tracking doc: TimeLine.md
3. Intern project final report: OpenStackVM\_Testing\_final\_report.doc
4. Intern project final presentation: OpenStackVM\_Testing\_final\_report.pptx
5. Linux system information scan shell script: Linux\_sys\_checker.sh
6. Windows system information scan shell script: win\_sys\_checker.bat
7. OpenStack VM introduction manual: OpenStack\_Beta\_VM\_manual.doc
8. OpenStack VM introduction creation steps manual: OpenStack\_Image\_creation\_manual.doc
9. Project problem and solution tracking document: Problem and Solution.docx
10. **Assignment Timeline/Milestone**

Below is the project timeline draft and we will do adjustment and change in the future. We may do a very short discussion every week and every month to track the project progress.

|  |  |
| --- | --- |
| **Week Index** | **Task/Milestone** |
| Week 1 | * Improve the project design document. * Create a project implement plan timeline document. (TimeLine.md) * List down all the knowledge need to pick up. |
| Week 2 | * Read and try the NCL document/OpenStack manual. * Pick up the related knowledge |
| Week 3 | * Start testing and continues knowledge learning if needed. |
| Week 4-7 | * Finish the task 1 the main testing part. * Implement the test report.(OpenStackVM\_Testing\_final\_report.doc) |
| Week 8-9 | * Finish the task 2 VM image creation. * Implement the VM creation manual. (OpenStack\_Image\_creation\_manual.doc) |
| Week 10-12 | * Finish all the documents. * Short presentation to the team. |

1. **Reference**

The Intern can list down all the links/document he used for the project here:

OpenStack Official web: <https://www.openstack.org/>