Jurnal Informatika, Vol. 12, No. 2, November 2014, 53-56
ISSN 1411-0105
DOI: 10.9744/informatika.12.2.53-56

SETTING UP AN INDUSTRIAL CONTROL SYSTEMS LABORATORY

Haryanto Natalius Liuwan[1*], Casper Schellekens[2], Daniel Trivellato[3]
[1] Petra Christian University, Jl. Siwalankerto 121 – 131 Surabaya 60236, Phone: +62 31 – 2983455
[2] Fontys Hogescholen ICT, Rachelsmolen 1, 5612MA Eindhoven, Netherlands, Phone: +31 8850 73223
[3] Security Matters B.V., Twinning Centre, 5612 AR Eindhoven, Netherlands, Phone: +31 642483416
E-mail: hary.232@gmail.com, c.schelleckens@fontys.nl, d.trivellato@secmatters.com
*Corresponding autor

Abstract: With the evolution of Industrial Control Systems, many solutions from vendors are offered for industries. But sadly, most of those solutions are close-sourced, delivering lack of support for third parties who aim to develop Industrial Control Systems further. A start-up company named SecurityMatters needs an industrial instrument to simulate industrial environment to have a better idea how a particular protocol works. The application made in this project was developed using Java programming language to have compatibilities across platforms. An Object-Oriented-Programming and Model-View-Controller pattern are used as well to ensure maintainability. This application can be used to demonstrate capabilities of Modbus protocol and test industrial devices for vulnerabilities.

Keywords: Industrial Control Systems, Modbus, Java.

INTRODUCTION

Industrial Control Systems along with its components are the core of industrial production processes today. They enable automation of production processes with a simple and effective system rather than manual and 24/7 analog-monitored systems used in old times. Figure 1 shows an example of Supervisory Control and Data Acquisition network where multiple Industrial Control Systems exchanges information to carry out the industrial process.

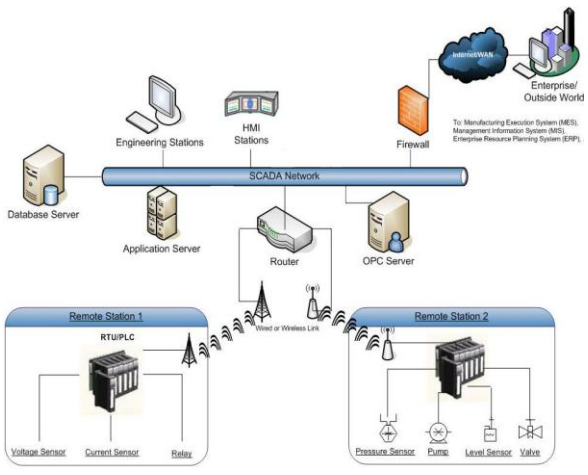

Figure 1. Example of Supervisory Control and Data Acquisition network

The most common Industrial Control Systems/ Supervisory Control and Data Acquisition systems used to monitor and manage operations of automation instruments is the HMI. It represents industrial devices in an interface, and is used to control and monitor industrial devices anywhere and everywhere. With the usage of TCP/IP networks in ICS/ SCADA systems, cyber-attacks in the past years have been increasingly targeting such systems. For example, the StuxNet malware attack in 2010 successfully disrupted the process of Iran's nuclear power plants. These circumstances led to a need to find better way to protect industrial networks and its equipment. Figure 2 shows how StuxNet works.

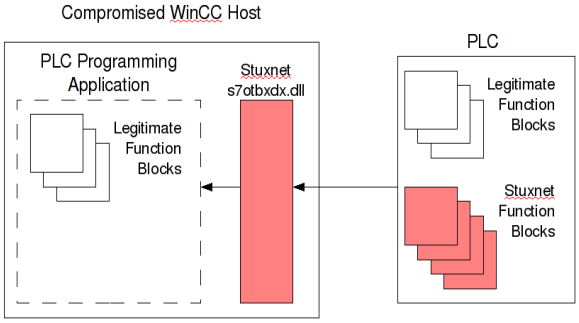

Figure 2. How StuxNet works

LITERATURE

Industrial Control Systems/Supervisory Control and Data Acquisition Systems

Industrial Control Systems (ICS) or Supervisory Control and Data Acquisition Systems (SCADA) are the systems which can be used to control industrial environment. Figure 3 shows an example of old ICS for monitoring purpose.

Figure 3. Example of old ICS

As the technology evolves, computer has been used to modernize ICS and SCADA systems. Figure 4 shows an example of modern Human-Machine Interface (HMI) for a water treatment plant. In this way, companies can save lots of money and effort since the data can be accessed anywhere and everywhere.

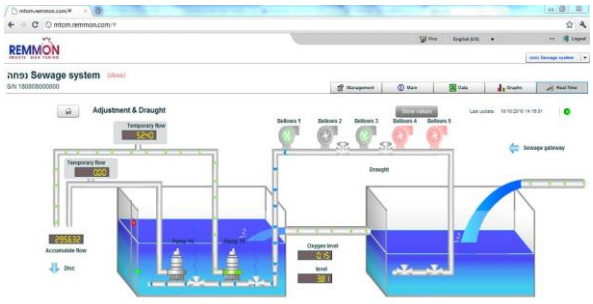

Figure 4. Example of Modern HMI

One of the core of modern ICS/SCADA Systems is a component called PLC (Programmable Logic Controller). It is essentially a small computer put in industrial environments to read values from sensors and takes decisions on a subsystem. It also sends information to HMI. Figure 5 shows an example of ABB PLC.

Figure 5. ABB PLC

MODBUS Protocol

One of the protocols used for ICS/SCADA System communication is called Modbus. It was developed by Modicon/ Schneider Electric. This protocol was developed as a simple and free protocol at first, and modern features such as Ethernet were added lately. Figure 6 shows simple implementation of Modbus protocol on industrial environment.

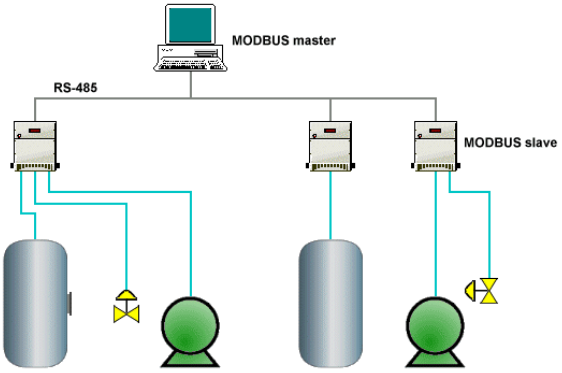

Figure 6. Modbus Protocol Implementation

Device Vulnerabilities

The PLC device itself could be attacked by sending wrong commands or bursting commands above device's capacity. Every device could have different response for each type of attacks, depending how it designed. This vulnerability should be verified for devices which are available in the market, to ensure that devices used by critical institutions are safe.

SYSTEM DESIGN AND ANALYSIS

Programming Language and Platform

In software development phase, Java programming language was chosen, and Object Oriented Programming and Model-View-Controller pattern were used to maintain compatibility and maintainability of the program itself. Figure 7 shows Model-View-Controller concept diagram.

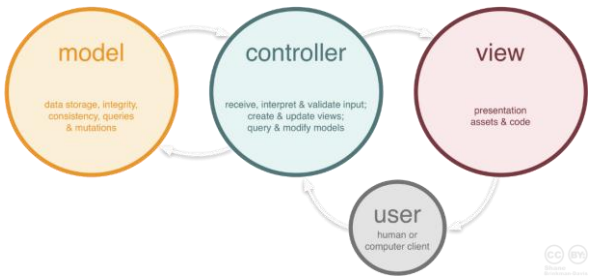

Figure 7. MVC Diagram

In Graphical User Interface side, JavaFX platform were chosen to ensure compatibility and most up-to-date platform. One of the benefits of using JavaFX is it has an integrated designer called SceneBuilder. By using this software, the Graphical User Interface can be designed using a previewer which makes designing experience better compared with other platforms. Figure 8 shows JavaFX SceneBuilder in usage for this project.

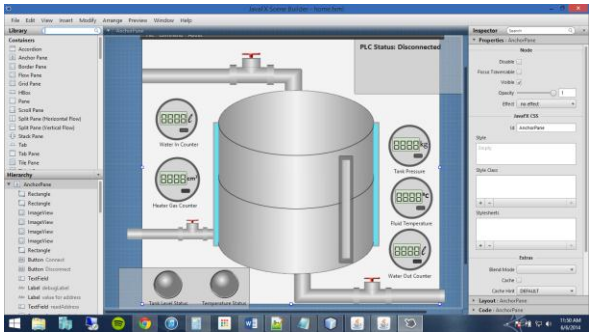

Figure 8. JavaFX SceneBuilder

During the software development, it was decided to create an Application Programming Interface to ensure effective programming and possible future expansion using codes written in this project.

Library Selection Process

A library is used in this project to simplify development process. There are plenty of library choices that could be found in internet such as Libmodbus and Modbus4j. But at the end, Jamod were chosen because it is written in Java language, and Modbus4j is too complicated for this case.

Before library selection started, a protocol behavior learning were started to know how Modbus protocol usually behaves. This task were done by having server and client simulator running between a Wireshark tool to wiretap the communication messages. Figure 9 shows Wireshark tool along with wiretapped messages.

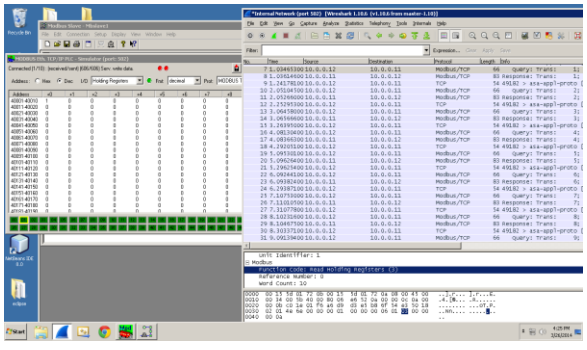

Figure 9. Wireshark tool

Development Process

During the software development, all Modbus protocol traffics are simulated using software called PLC simulator to simplify the development process. It is basically a PLC simulator based on a VBScript it load. While it might not represent real PLCs in showing vulnerabilities, it is sufficient for testing Modbus official commands. Figure 10 shows PLC Simulator software.

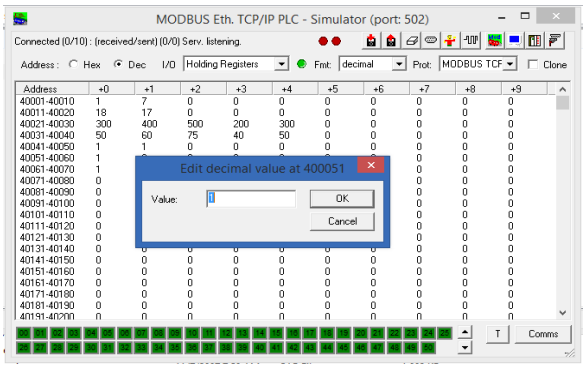

Figure 10. PLC Simulator Software

RESULT

The result of this project is a HMI application, which is based on Java programming language. This application has an ability to demonstrate industrial process, and also has some advanced functions that would be useful for security tester. Figure 11 shows application running normally for demonstration purpose. In this program, the case used is about industrial water boiler that can be controlled manually or automatic using defined thresholds. The application also has a feature to automatic reconnect whenever it is disconnected from PLC using last known good configuration.

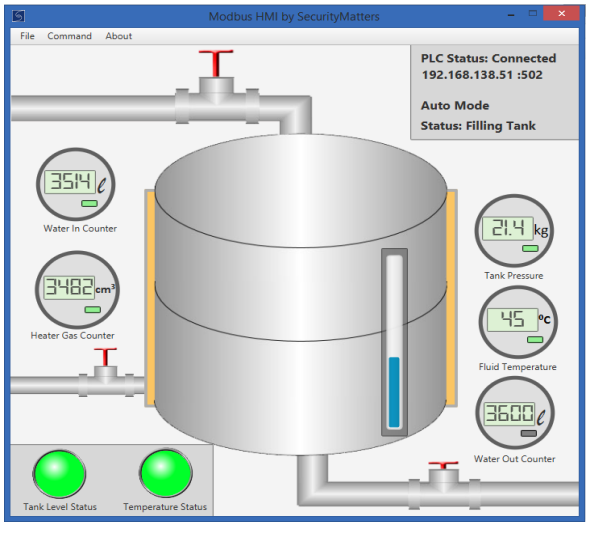

Figure 11. Demonstration Function

Along with demonstration function shown in figure 11, this application also has a functionality to send a Modbus command according to Modbus official specification and stack. One of the examples is shown in figure 12, where this application has a functionality to read a value from PLC. Using this function, user can see raw values that PLC runs in case they want to debug the program on the PLC.

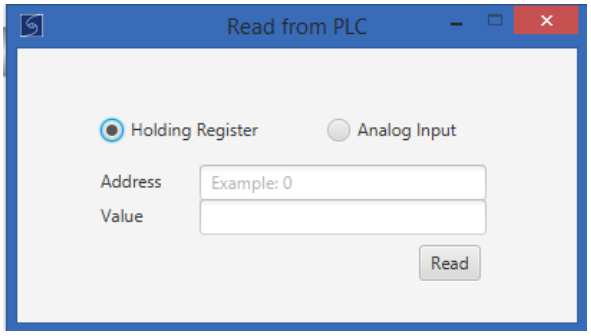

Figure 12. Testing Program

Figure 13 shows advanced security testing feature that this application has. This function sends raw hexadecimal values straight to PLC. By having this functionality, testers can understand how a PLC reacts to particular attack. For example, testers can try to replicate signature from known worms to send it directly to PLC to understand how a PLC reacts.

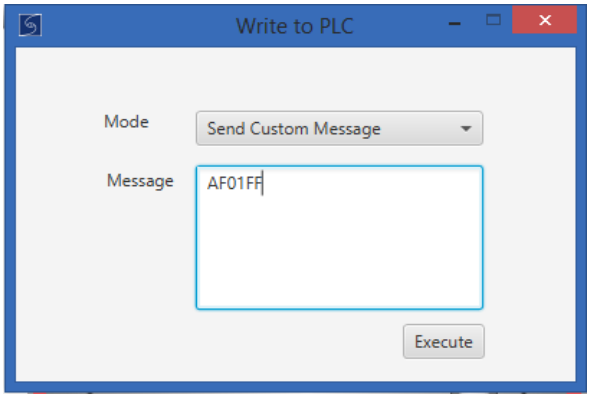

Figure 13. Security Testing

Along with all that features, this software are going to be published in open-source ICS community to give an idea to community about example of working Modbus HMI application, which does not exist yet.

CONCLUSION

Conclusions that could be taken from this project are as follows:

- This application can be used for demonstation purpose
- This application can be used for security testing purpose
- This application can be used with real PLC that uses Modbus protocol. Figure 14 shows test conducted to test compatibility with the software.
- This application provides open source community with working HMI application which does not exist yet.

Figure 14. Testing Process

REFRENCES

[1] http://blog.industrialdefender.com/blog/stuxnet-whitepaper-updated
[2] Clarke, G., Reynders, D. 2004. Practical Modern SCADA Protocols.IDC Technology
[3] http://www.modbus.org
[4] http://www05.abb.com/global/scot/scot349.nsf/veritydisplay/b48192f9da2e1947c12579c 7005ea32a/$file/3BSE063717_A_en_Compact_800_5.1_Flexible_process_control_products.pdf
[5] http://www.remmon.com/en/products/web_based_hmi/mtom.php
[6] http://vietphatgroup.com/index.php?mod= news &cpid=207
[7] http://jamod.sourceforge.net/
[8] http://yalantis.com/blog/lightweight-ios-view-controllers-separate-data-sources-guided-mvc/
[9] http://www.technologyuk.net/telecommuni-cations/ industrial_networks/modbus.shtml
[10] http://www.enisa.europa.eu/media/press-releases/industrial-control-systems-security-recommendations-for-europe-member-states