University of Kent

Kent Academic Repository

Wu, Tina and Nurse, Jason R. C. (2015) *Exploring The Use Of PLC Debugging Tools For Digital Forensic Investigations On SCADA Systems.* Journal of Digital Forensics, Security and Law, 10 (4). pp. 79-96. ISSN 1558-7215.

Downloaded from

https://kar.kent.ac.uk/67499/ The University of Kent's Academic Repository KAR

The version of record is available from

This document version

Publisher pdf

DOI for this version

Licence for this version

CC BY-NC-ND (Attribution-NonCommercial-NoDerivatives)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

PLC FORENSICS BASED ON CONTROL PROGRAM LOGIC CHANGE DETECTION WORKS

Ken Yau and Kam-Pui Chow
University of Hong Kong, Hong Kong, China
kenyaufriends@yahoo.com.hk, chow@cs.hku.hk

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) system is an industrial control automated system. It is built with multiple Programmable Logic Controllers (PLCs). PLC is a special form of microprocessor-based controller with proprietary operating system. Due to the unique architecture of PLC, traditional digital forensic tools are difficult to be applied. In this paper, we propose a program called Control Program Logic Change Detector (CPLCD), which works with a set of Detection Rules (DRs) to detect and record undesired incidents on interfering normal operations of PLC. In order to prove the feasibility of our solution, we set up two experiments for detecting two common PLC attacks. Moreover, we illustrate how CPLCD and network analyzer Wireshark could work together for performing digital forensic investigation on PLC.

Keywords: PLC Forensics, SCADA Security, Ladder Logic Programming

1. INTRODUCTION

Digital forensics plays an important role for incident investigations on digital devices, for example, personal computer, smart-phone, digital camera, and flash drive. Standard guidelines and procedures are provided to implement the digital forensic processes: identification, collection, analysis and reporting [11]. According to the collected evidence, investigators can re-construct the incident and present to a court if crime is involved. In addition, the evidence can be used to trace what causes the incidents in order to avoid the same incident happening again in the future [2].

A programmable logic controller (PLC) is a special form of microprocessor-based controller. It uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes [1]. Lighting Control system is one example of PLC applications. It is used to turn lights on automatically when the area becomes occupied and off when it becomes unoccupied.

A simple automation control system can be monitored and controlled by a single PLC. However, a complex and larger automation control system called Supervisory Control and Data Acquisition (SCADA) system, needs to be built with multiple PLCs. SCADA system is an automation system widely used to monitor and control industrial processes such as electric power generation, public transportation, chemical plants, water management and so on. If any undesired incidents occur on the SCADA systems, substantial risk to the health and safety of human lives, serious damage to the environment, as well as serious financial

© 2015 ADFSL

issues such as production losses, negative impact to a nation's economy, and compromise of proprietary information [2].

SCADA systems are always built with proprietary technologies and communication protocols. For example, PLC manufacturers provide its proprietary operating systems to their products. Due to the uniqueness of SCADA system, there are several challenges when performing digital forensic processes on these systems. Those challenges will be discussed in Section 4. In this paper, we propose a solution to detect and record abnormal operations of PLC based on the control program logic change in PLC. The abnormal operations are stored in the format of a log file which could help SCADA forensic investigation.

2. SECURITY ISSUES OF PLC

An automation control system can be setup by connecting a PLC with Input and Output devices. Input devices might be switches, temperature sensors, flow sensors, etc. Output devices might be motors, solenoid valves, etc. Besides hardware installation, the PLC has to be programmed in order to monitor the inputs and control the outputs based on a set of control rules. Each PLC manufacturer has its own software for programming their PLCs. For example, PLCs of Siemens Simatic S7 series are programmed, configured, and managed using software STEP 7.

Nowadays, many PLCs have evolved to utilize common networking standard such as Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11) for communication among the connected devices [3]. Therefore, cyber-attacks on SCADA systems become one of the important security issues after the systems have been exposed to the Internet.

3. ATTACKS ON PLCS

There are various kinds of attacks on PLCs. One of the famous attacks is STUXNET, a malware to infect Simatic programming device (i.e., PC running Step 7 on Windows). The malware was used to reprogram the PLCs by inserting its own blocks of code, and replacing or infecting existing blocks [6]. According to the paper "Exploiting Siemens Simatic S7 PLCs" [3], Dillon Beresford mentioned that the network protocols designed for communication among field devices in control systems were intended to be open and reliable, but not secure in past. International Standards Organization Transport Service Access Point (ISO-TSAP RFC 1006) is one of the not secure network protocols.

Dillon Beresford demonstrated several attacks on Siemens Simatic S7 PLCs during the presentation at Black Hat 2011 Conference. The attacks were 1) TCP Replay over ISO-TSAP Attack; 2) S7 Authentication Bypass; 3) CPU Stop and Start Attack; 4) Memory Read and Write Logic Attack; 5) Decrypting Siemens Simatic firmware; 6) Getting a Shell on the PLC. The exploits demonstrated by Beresford were using PROFINET and communicating across TCP/IP port 102 (ISO-TSAP).

PROFINET is a standard for Industrial Ethernet based on Industrial Ethernet and support the following three protocols [9]:

1. TCP/IP with reaction times in the range of 100 ms
2. RT (real-time) protocol with 10 ms cycle times
3. IRT (Isochronous Real-Time) with cycles times of less than 1 ms

Data is transmitted in plain text over TCP/IP port 102 (ISO-TSAP). Therefore, if attackers record the network traffic, they could easily extract data such as user names,

 © 2015 ADFSL

passwords, commands, negotiated sessions, logic, etc. Any of these variables could lead a PLC to be compromised.

4. DIGITAL FORENSICS CHALLENGES ON SCADA SYSTEMS

Cyber forensics can be challenging when being applied to non-traditional environments like SCADA systems. The systems are not comprised of current information technologies and not designed with technologies to provide adequate data storage or audit capabilities. In addition, further complexity is introduced if the environment is designed using proprietary solutions and protocols, thus limiting the ease of which modern forensic methods can be utilized [11].

Due to the complexity of SCADA systems, evidence is difficult or impossible to be extracted and collected by using traditional digital forensics for investigation. One of the challenges is that the operations of SCADA systems must be kept running, therefore, investigators cannot shut down the system to perform data acquisition [2]. In such situation, live forensics is the possible way to perform data acquisition. However, performing live forensics on SCADA systems might affect its normal operations. The second challenge is that many SCADA systems use proprietary and legacy software, hardware and communication protocols. Therefore, traditional digital forensic tools might not be able to apply to the systems. Furthermore, SCADA forensics is lack of event logs for investigation

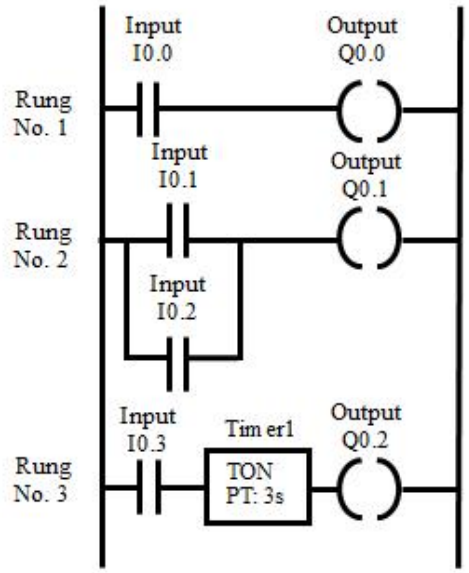

Figure 1. LAD Diagram of a Control Program

5. LADDER LOGIC PROGRAMMING

PLC is designed to be operated by engineers. The engineers are good at electrical circuit design but might have limited knowledge of computer programming. Meanwhile, different kinds of automation control systems need different kinds of PLC control programs. Therefore, Ladder Logic (LAD) is the most common programming Language for PLC because the programming language looks like a simple wiring diagram for an electrical circuit [1].

A LAD diagram in Figure 1 consists of a set of ladder rungs and each rung has a set of input instructions and output instructions [1]. The ladder diagram has two vertical rails. The left vertical rail supplies power to all the horizontal rungs of the ladder. Each rung on the ladder defines one operation in the control process.

A LAD diagram is read from top to bottom and left to right. Each rung must start with an input or inputs and must end with at least one

© 2015 ADFSL

output. The input is used for a control action, such as closing the contact of a switch. The output is used for a device connected to the output of a PLC such as a motor or a valve. In Figure 1, Rung No. 1 can be interpreted as if Input I0.0 (Switch) is ON, then the Output Q0.0 (Motor) is ON. Likewise if I0.0 is OFF, then Q0.0 is OFF. Rung No. 2 is that if either Input I0.1 or I0.2 is ON, then the Output Q0.1 is ON, otherwise Q0.1 is OFF. Rung No. 3 is that if Input I0.3 is ON, the Output Q0.2 will be ON after 3 seconds delay. The time delay is controlled by the Timer1 (TON: Timer on Delay).

6. PROPOSED SOLUTION FOR PLC FORENSICS BASED ON CONTROL PROGRAM LOGIC CHANGE DETECTION

In this paper, we propose a solution to detect two most common attacks on PLC. The first attack is Control Program Attack which reprograms the PLCs, like STUXNET. The second attack is Memory Read and Write Logic Attack which alters values of memory variable of a control program on a running PLC.

To detect these attacks, we use a program called Control Program Logic Change Detector (CPLCD) with a set of Detection Rules (DRs). CPLCD is a program developed by using Libnodave. Libnodave is a free library for data exchange between a PC and a Siemens PLC over TCP/IP port 102 (ISO-TSAP) [4]. CPLCD is working on Microsoft Windows environment. It has to work with a set of defined Detection Rules (DRs) for detecting these two PLC attacks. DR is in the form of Boolean expression derived from the LAD control program. Different designs of control programs are converted to different sets of DRs.

To define a DR, we have to transform each rung of a Ladder Logic (LAD) diagram into a Boolean Expression according to the way of connection among Inputs, Outputs, memory variables, etc. For example, if Input A connects Input B in series with Output C, then we can formulate a DR as A AND B = C. If Input A connects Input B in parallel, then we can formulate a DR as A OR B = C. The three DRs shown in Table 1 are derived from the three rungs of LAD diagram in Figure 1.

If the PLC operations do not follow the instructions of the control program, we assume the PLC might suffer from attacks or PLC failure. Followings are the procedures for detecting the two PLC attacks using CPLCD and DRs. Assume the PLC control program is same as Figure 1.

Table 1
Detection Rules

Rung No.	Detection Rule
1	I0.0 = Q0.0
2	I0.1 OR I0.2 = Q0.1
3	(I0.3 AND Timer1=3) = Q0.2

1. Transform each ladder rung of the LAD Diagram into DRs (see Table 1).
2. Connect CPLCD to PLC via TCP/IP port 102.
3. Run CPLCD to read the memory variables (e.g. I0.0, M0.0, Q0.0, etc.) of control program from the PLC and assign values (TRUE/FALSE/Numbers) of the variables to the DRs. When any one of the values violates the DRs, CPLCD raises alert and logs the event and timestamp in a file for forensic investigation.

CPLCD is able to perform real-time PLC attack detection on a running PLC and

 © 2015 ADFSL

capture the attack details in a log file for forensic investigation.

In order to prove the feasibility of our solution, we set up two experiments for detecting the two attacks. Experiment 1 is for detecting the Control Program Attack and Experiment 2 is for detecting Memory Read and Write Logic Attack. As Siemens is one of the popular PLC manufacturers, we select Siemens Simatic S7-1200 for our experiments.

6.1 Experiment 1: Setup and procedures for detecting Control Program Attack

To set up this experiment, we used one PLC, one PC and a router. They were connected as diagram shown in Figure 2. The PC was installed with CPLCD for detecting the attack.

The procedures for detecting the attack were as follows:

Step 1: Designed two control programs. One was the original program in Figure 4 and the other one was altered by an attacker in Figure 5 to create abnormal instructions. The attacker altered the second rung of the original control program from (I0.1 AND M0.1) = Q0.1 to (I0.1 OR M0.1) = Q0.1

Step 2: Defined two Detection Rules (DRs) based on the original control program as follows:

DR No. 1: (I0.0 OR M0.0) = Q0.0

DR No. 2: (I0.1 AND M0.1) = Q0.1

Step 3: Loaded the altered control program with initial memory values, M0.1=False and Input I0.1=True into the Siemens Simatic S7-1200 PLC by Siemens programming software STEP 7 and started running the PLC.

Step 4: Started the CPLCD program with the defined DRs to monitor memory variables of the control program in the PLC and check any DR was violated by the values of the variables.

Step 5: An alert was raised, checked error log file of CPLCD as shown in Figure 3.

6.2 Experiment 1: Result

According to the error log captured by CPLCD in Figure 3, an alert was raised and logged on 27/6/2015, 11:02:43 PM because the values of I0.1, M0.1 and Q0.1 violated DR No. 2. Under normal operation of DR No. 2, Q0.1=True only if both I0.1 and M0.1=True at the same time, otherwise, Q0.1=False. However, CPLCD detected that I0.1=True, M0.1=False and Q0.1=False which violated DR No. 2.

6.3 Experiment 2: Setup and procedures for detecting Memory Read and Write Logic Attack

This experiment setup was same as Experiment 1 but one more PC was added as shown in Figure 7. The PC installed with Snap7 acting as an Attacker to perform Memory Read and Write Logic Attack. We used Snap7 to alter the PLC's memory data. Snap7 is not a PLC attacking tool, it is an open source, 32/64 bit, multi-platform Ethernet communication suite for interfacing natively with Siemens S7 PLCs [7].

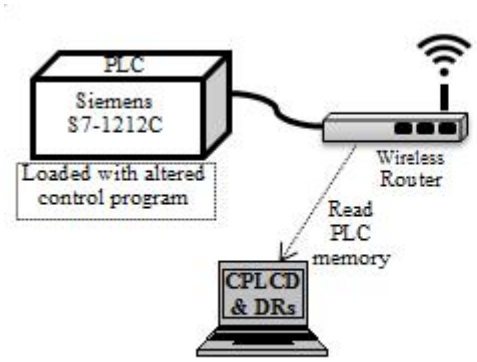

*Figure 2.*Setup of Experiment 1

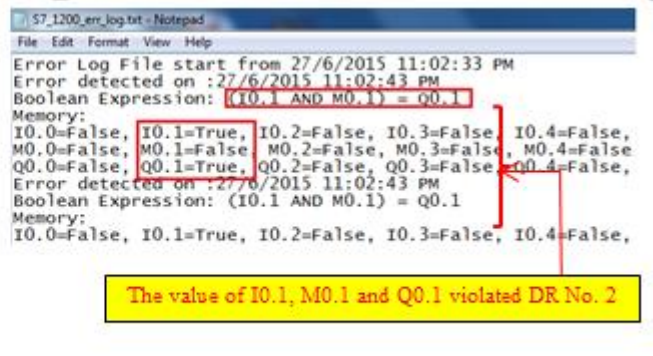

Figure 3. Result of Experiment 1

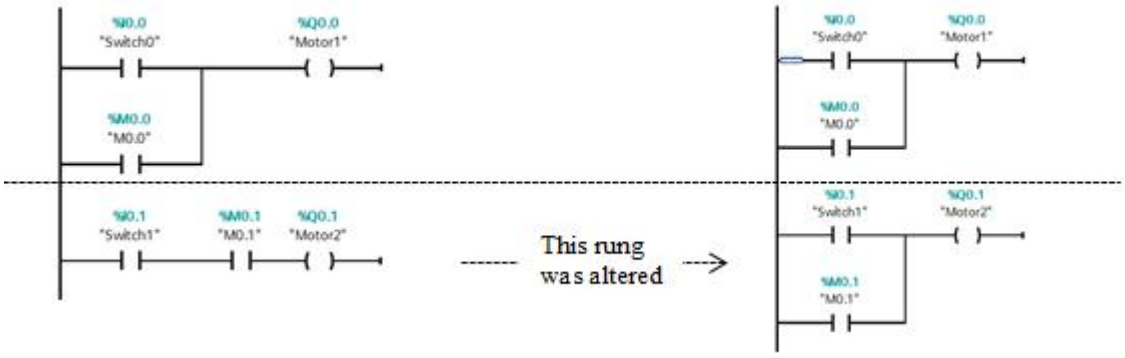

Figure 4. Original Control Program

Figure 5. Altered Control Program

The procedures for detecting the attack were as follows:

Step 1: Designed a control program (Figure 6) and load the control program into the PLC. Set initial value of Input I0.0, I0.1, I0.2, I0.3 and I0.4 to False so that Output Q0.0 and Q0.1 were False, and then started running the PLC.

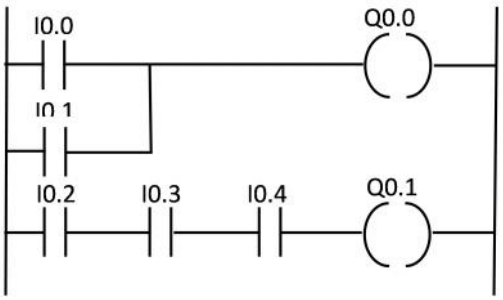

Figure 6. Control Program

Step 2: Defined two Detection Rules (DRs) according to the control program as follows:

DR No. 1: (I0.0 OR I0.1) = Q0.0

DR No. 2: (I0.2 AND I0.3 AND I0.4) = Q0.1

Step 3: Started the CPLCD program with the defined DRs to monitor memory variables of the PLC control program and check any DR was violated by the values of the variables.

Step 4: Used Snap7 to alter the PLC's memory Output Q0.0 and Q0.1 to True from False.

Step 5: An alert was raised, checked error log file of CPLCD as shown in Figure 8.

 © 2015 ADFSL

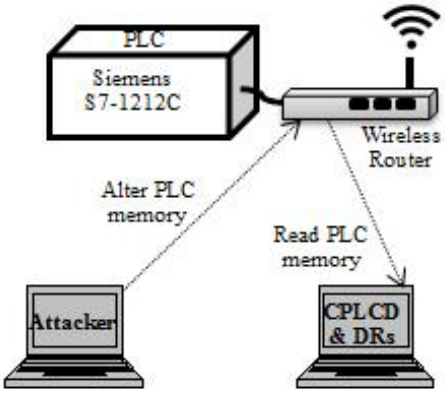

Figure 7. Setup of Experiment 2

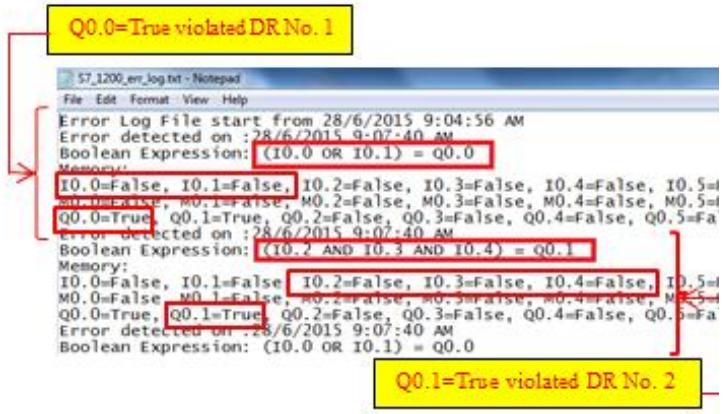

Figure 8. Result of Experiment 2

6.4 Experiment 2: Result

According to the error log captured by CPLCD in Figure 8, DR No. 1 and DR No. 2 were violated on 28/6/2015 9:07:40 AM because of the action in Step 4. DR No. 1 was violated by the value Q0.0=True and DR No. 2 was violated by the value Q0.1=True.

7. DISCUSSION

Based on the defined Detection Rules (DRs), Control Program Logic Change Detector (CPLCD) was able to detect the Control Program Attack and Memory Read and Write Logic Attack. However, it is difficult or impossible to define the entire DRs from a complicated LAD diagram which has a lot of rungs and to monitor all the memory variables. To address this issue, we propose to select important rungs for monitoring instead of all the rungs. Different control systems have different important rules. In general, a rung used to stop control system under dangerous condition should be important.

CPLCD can detect Memory Read and Write Logic Attack, however, it cannot provide sufficient information for forensic investigation. CPLCD does not capture information about how and who changes the PLC memory which induced the abnormal PLC operations. In order to supplement more information for forensic investigation, we recommend CPLCD to work with network packet analyzer Wireshark. Wireshark supports PROFINET to record and analyze the Ethernet message frames. It can be used to dissect the ISO on TCP-packets for communication to Siemens S7 PLCs after adding Wireshark dissector plugin for S7 communication.

S7 Protocol is Function oriented or Command oriented, i.e. each transmission contains a command or a reply. It is the backbone of the Siemens communications, its Ethernet implementation relies on ISO-on-TCP (RFC1006) which is Block oriented by design [8]. Each block is named S7 PDU (Protocol Data Unit). Each command (S7 Telegram) consists of a header, a set of parameters (Params), a parameters data (Pardata) and a data block (Data) as shown in Figure 8. The

first two elements are always present, the other two are optional. If a S7 Telegram consists of Header="Write", Params="DB, 10", Pardata="4" and Data="data", it can be interpreted as "Write 'data' into Data Block 10 starting from the offset 4" [8]. S7 Protocol, ISO TCP and TCP/IP follow the well-known encapsulation rule, shown in Figure 8 [8]. According to the S7 Protocols specifications and the well-known encapsulation rule, forensic investigator is able to construct useful information from the TCP/IP data packets for investigation.

Following is a case to illustrate how CPLCD and Wireshark work together for forensic investigation on cyber-attack to a PLC. A PLC controls a motor (Q0.0) by two switches (I0.0 and I0.1). The motor is ON only when the two switches are ON, otherwise it is OFF. Therefore, the DR is "I0.0 AND I0.1 = Q0.0". Assuming an attacker tries to turn on the motor by altering Output Q0.0 from FALSE to TRUE when both input switches are OFF (FALSE).

Before Attack:

The switches are OFF and the Motor is OFF.

The DR is "FALSE AND FALSE = FALSE".

After Attack:

The switches are OFF but the Motor is ON.

The DR "FALSE AND FALSE = TRUE" is violated.

Since the DR is violated after the attack, CPLCD raises an alert and logs the variable names/values and timestamp on a file. A forensic investigator can use the timestamp to filter out the relevant packets from the file captured by Wireshark. Based on the analysis of those network packets, the investigator can trace what and how the S7 commands were executed to attack the PLC. Besides, we might be able to reveal the IP address of the attacker in this way.

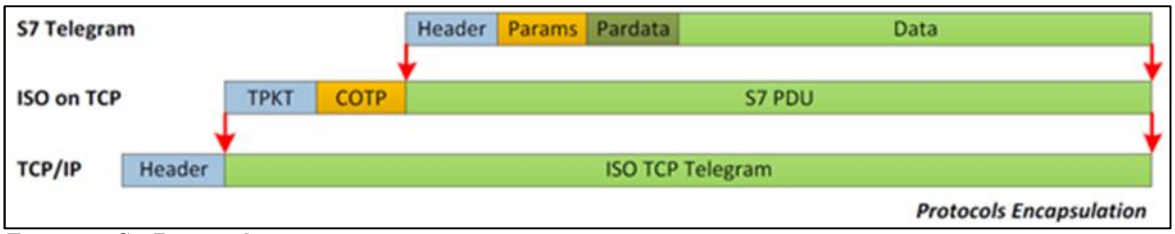

Figure 8. S7 Protocols
Source: Snap7 Reference manual [8]

8. CONCLUSION AND FUTURE WORK

PLC is one of the important components in SCADA systems but lack of security control and hard to perform digital forensics on it. It is facing many cyber-attacks after exposing SCADA systems to Internet dramatically in recent years. Siemens is aware of the security issues and provides warning on Simatic S7-1200 Manual "If an attacker can physically access your networks, the attacker can possibly read and write data" [12].

Digital Forensics is an essential part of cyber defense and becomes relevant when there is a security breach [10]. However, there are insufficient forensic tools and procedures to perform digital forensics on PLC. To help

 © 2015 ADFSL

overcome challenges on PLC protection and forensic investigation, in this paper, we introduced how Control Program Logic Change Detector (CPLCD) and Detection Rules (DRs) can be used to detect Control Program Attack and Memory Read and Write Logic Attack. In addition, we illustrated how CPLCD work with Wireshark dissector for S7 communication to perform forensic investigation on S7 PLCs.

In future, we will apply CPLCD to various types and brands of PLCs for testing of performance, accuracy and feasibility. Furthermore, we will expand our testing to a simulated control system application such as elevator, traffic light, robotic arm, etc.

© 2015 ADFSL

REFERENCES

W. Bolton, Programmable Logic Controllers (4th Edition)

Irfan Ahmed, Sebastian Obermeier and Martin Naedele, Golen G. Richard III: SCADA System: Challenges for Forensics Investigations, IEEE Computer, Vol. 45 No. 12, December 2012, pp 44–51, USA.

Dillon Beresford, Exploiting Siemens Simatic S7 PLCs, Black Hat USA+2011, July 8, 2011

Alex Sentcha, LibNoDave – exchange data with Siemens PLC, https://alexsentcha.wordpress.com/ Last accessed on 31 May 2015

R.M. van der Knijff, Control systems/SCADA forensics, what's the difference?, Digital Investigation 11 (2014)

Nicolas Falliere, Liam O Murchu, and Eric Chien: W32.Stuxnet Dossier, Version 1.4, Symantec Corporation, February 2011

Davide Nardella, Snap7 http://snap7.sourceforge.net/ Last accessed on 13, June 2015

Davide Nardella, Snap7 Reference manual Rev.5, January 1, 2015

PROFINET, Wikipedia http://en.wikipedia.org/wiki/PROFINET_IO, Last accessed on 18 June 2015

K. Mandia, C. Prosise and M. Pepe, "Incident Response and Computer Forensics", McGraw-Hill/Osborne, Emeryville, California, 2003

Fabro, M: Recommended Practice: Creating Cyber Forensic Plan for Control Systems, Department of Homeland Security (2008), Idaho National Laboratory (INL), August 2008, USA

SIEMENS SIMATIC S7-1200 Easy Book Manual 01/2015

 © 2015 ADFSL