Forensic Science International: Digital Investigation 40 (2022) 301354

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

A study on command block collection and restoration techniques through detection of project file manipulation on engineering workstation of industrial control system

Jiho Shin [a], Hyunpyo Choi [b], Jung Taek Seo [b, *]

[a] Police Science Institute, Korean National Police University, Asan-si, Chungcheongnam-do, 31539, South Korea
[b] Department of Computer Engineering, Gachon University, Seongnam-si, Gyeonggi-do, 13120, South Korea

ARTICLE INFO

Article history:
Received 16 March 2021
Received in revised form
23 August 2021
Accepted 9 February 2022
Available online 17 February 2022

Keywords:
Industrial control system
Digital forensic
Engineering workstation
Siemens TIA Portal
Project file

ABSTRACT

Major control systems such as ICS/SCADA are currently being used in the key infrastructures of our society, notably in the energy, transportation, and healthcare sectors. The PLC, which is located in the lowest layer of the control system, is directly connected to diverse field devices including sensors and actuators. In the event of a cyber-attack on the PLC, the potential ripple effects, such as a collapse of the national infrastructure, could be very large, making a high level of cyber safety essential. Therefore, it is necessary to apply digital forensic technology to the PLC to respond to cyber-attacks against the control system. However, conventional methods of directly investigating the PLC, such as network and memory forensics, have practical difficulties as they could compromise the availability of the control system. As such, this paper proposes a method of detecting and restoring logic data when the logic data are changed as a result of a cyber-attack, by using the digital forensic technology developed for the Engineering Workstation (EWS), which has been used to develop, manage and set the control system program logic.

1. Introduction

Our society has achieved rapid industrial development by applying the industrial control system (ICS) to core infrastructures such as automated processes, power generation facilities, energy supply facilities, transportation control, and smart cities. Although it is difficult to define the ICS with a universally recognized meaning due to its broad scope, it is generally understood to be both an actual physical system and an embedded system that controls and monitors a physical system (Humayed et al., 2017). Since the ICS basically relies on traditional IT systems, it is also defined as an integrated IT system that links physical systems and IT systems (Gollmann, 2013). The cyber physical system (CPS) and supervisory control and data acquisition (SCADA) are often discussed together as being similar to ICS or as detailed concepts. This paper considers them as a control system to monitor and operate major national infrastructures or physical systems such as energy, transportation, and smart factories, effectively without making any distinction between them.

The exposure of the ICS to cyber threats could inflict serious disasters on our society, as the following examples demonstrate. In 2010, a cyber-attack using Stuxnet, known to be the first malware to specifically target control systems, infected the programmable logic controllers (PLC) controlling the field devices of a nuclear power plant and destroyed more than 1,000 centrifuges (Langner, 2011a). In 2012, a malware attack against Saudi Aramco interrupted the power supply, causing a problem with oil production (Bronk and Tikk-Ringas, 2013). In 2014, an email phishing attack against KHNP in Korea resulted in a leakage of important internal data (Lee and Lee et.al., 2020). In 2015, a malware attack using Black Energy3 against transformers in Ukraine caused a massive blackout (Khan et al., 2016). Moreover, ransomware called LogicLocker, which specifically targets SCADA, has appeared, indicating that cyber-attacks against control systems are becoming increasingly intelligent and sophisticated (Formby et al., 2017). Although security awareness and responses to attacks have steadily increased due to a wave of security attacks and threats against control systems,

Please cite this article as: Jiho Shin et al., A Study on Command Block Collection and Restoration Techniques through Detection of Project File Manipulation on Engineering Workstation of Industrial Control System, Forensic Science International: Digital Investigation (2021), http:// doi.org/xxxxxx.
* Corresponding author.
E-mail addresses: suchme@police.go.kr (J. Shin), glucose6126@gmail.com (H. Choi), seojt@gachon.ac.kr (J.T. Seo).

https://doi.org/10.1016/j.fsidi.2022.301354

the threat has not diminished as hitherto unknown vulnerabilities are continuously detected.

The PLC, as the controller at the bottom of the ICS configuration hierarchy, is directly connected to diverse field devices. Since most cyber-attacks targeting the ICS aim to hijack control of the physical system and eventually cause malfunctions of the field devices, it is necessary to detect cyber threats and defend the PLC, and to conduct digital forensic investigations into intrusion incidents. Security activities that use digital forensic techniques can be classified into two types, i.e. ones that detect potential vulnerabilities of the system in advance, and the others that identify an attacker's behavior or intrusion path afterward.

However, there are limitations to forensic investigations of PLCs operating in the field. Since PLCs are directly connected to field devices, it is crucial to maintain their availability continuously. In other words, while data must be collected while the controller continuously is operating with the power turned on, the resulting loss can be enormous and a burdening factor if the collection and investigation of data require stopping the controller or inadvertently causes a problem with its operation. Therefore, there must be other ways of collecting and investigating data that do not involve a direct investigation of the PLC (Shin and Seo, 2019).

This paper reviews ways to collect data for a forensic investigation of the PLC and discusses the possibility of using an engineering workstation (EWS) that controls and manages PLCs. An EWS is a control computer that develops the logic of the program running in the PLC and sets the operational environment. The management program artifacts, network information, and the DLL information loaded in the EWS can be used to acquire the valid data required for the forensic investigation. The results of this study could be utilized in an environment in which is not feasible to investigate the PLC directly, or they could be used to investigate an environment that does not have a PLC.

2. Related works

Previous forensic research on PLCs utilized methods of investigating a PLC directly, with the main focus on detecting vulnerabilities by reverse-engineering the program logic related to the operation of field devices in the PLC physical memory (RAM) or the network packets transmitted and received by the PLC, or by monitoring normal operation. Tina Wu and Jason R.C. Nurse (WuJason and Nurse, 2015) proposed a method of acquiring PLC memory data using a PLC debugging tool and obtaining the program code from it, since the program code for attack exists in the PLC's memory at the time of a cyber-attack, and it is an important trace of the attacker's intention. Ken Yau and Kam-Pui Chow (Yau and Chow, 2015) developed Control Program Logic Change Detection (CPLCD), a collection and detection tool designed to detect whether the ladder logic running in a PLC is changed by external attacks, and verified it through a detection test. However, their tools were limited since exploring the PLC memory in real-time was a burden and thus could not guarantee the PLC's operational stability. Kam-Pui Chow and Siu-Ming Yiu (Yauet al., 2018) proposed a method of collecting data from network traffic for a PLC forensic investigation and storing it in a specific file format for a digital forensic investigation.

While existing studies related to PLC forensics have mostly focused on direct PLC investigations, such methods are limited in that they are not applicable in a situation where the PLC is not directly accessible, and such forensic investigations can significantly deteriorate the ICS's availability. Therefore, it should be possible to detect and respond to attacks on PLCs effectively by analyzing valid data collected from the EWS if is it is not feasible to directly collect data from the PLC. This paper intends to detect changes in the logic data of the project files used by the EWS for controlling field devices such as PLC. The proposed method consists in collecting the logic data in a separate storage area when a project file is altered by a cyber-attack and then restoring it to its pre-attack state.

3. ICS components and security threats

3.1. ICS components (Kentet al., 2006; Aquino-Santos, 2010)

Historian refers to a system that accumulates all collectible types of data, such as events, timestamps, and alarms generated by an ICS or SCADA system, using a database. The accumulated data are accessible from an operating layer such as the Human Machine Interface (HMI). It is a centralized database that provides statistical and trend insights to clients in the form of queries. The term HMI refers to software and hardware that enable interaction between operators and controllers through the operators' monitoring of the control process state and modification of the control settings. Operators can monitor the process status or record information, adjust the parameters, send commands to preset variables, and control the algorithms. The Engineering Workstation (EWS) is usually a high-end very reliable computing platform designed for configuration, maintenance, and diagnostics of the control system applications and other control system equipment, and is usually made up of redundant hard disk drives, high-speed network interface, reliable CPUs, performance graphics hardware, and applications that provide configuration and monitoring tools to perform control system application development, compilation and distribution of system modifications (https://us-cert.cisa.gov/). The Remote Terminal Unit (RTU) is a special-purpose data collection and control device designed to support Distributed Control Systems (DCS) and SCADA remote base stations. It is a field device equipped with a network function that includes wired/wireless radio interfaces for communication with a supervisory controller, and it is sometimes implemented as a Programmable Logic Controller (PLC) to act as a remote terminal device.

Although a PLC is very similar to an RTU, its control functions are more sophisticated, and it can control processes and perform simple and complex logic operations locally. PLCs can collect information from sensors and actuators and exchange data through supervisory control. A sensor in a control system refers to a device that measures a physical quantity or state and converts it into a value that an operator can check. In general, a sensor is used as a device that responds to an input quantity with an electric or optical signal. An actuator is a device that moves or controls a system, and also performs tasks related to current, pressure and air pressure.

3.2. Security threats targeting the ICS (Edenet al., 2015)

Previously, the ICS was regarded as being relatively safe from cyber-attacks. It uses an air-gap strategy to isolate a protected system from the external Internet network (see Fig. 1) at the early implementation phase. However, as the ICS has been integrated with new information and communication technologies such as Ethernet and wireless communication, its safety from external cyber-attacks through the network can no longer be guaranteed. While the air-gap strategy isolated the ICS operation from outside, the adoption of the Ethernet network environment has exposed it to more security vulnerabilities. It succumbs easily to social engineering attacks using an authorized system user's USB storage unit or laptop computer for infection as well as to bypass attacks using OSINT (Shin and Seo, 2019). As a result, it has become a relatively easy attack target compared to conventional information systems for which heavy investments have been made to maintain their

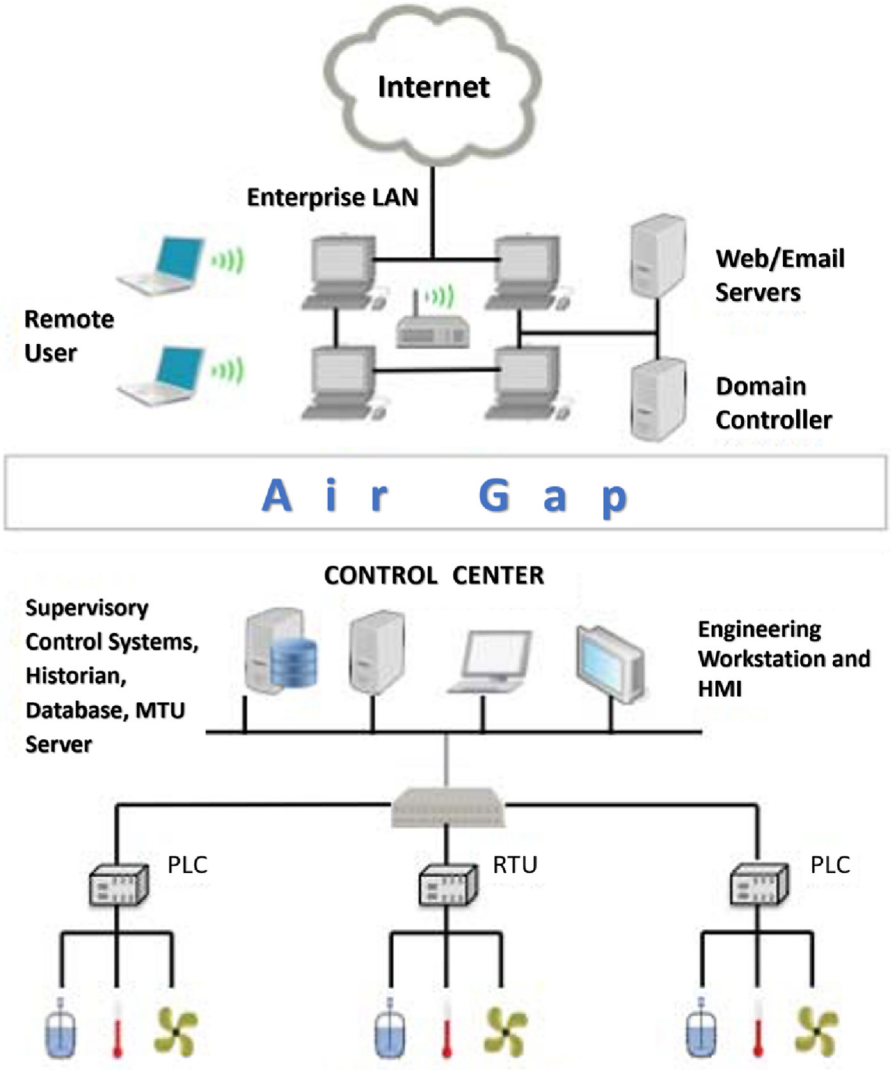

Fig. 1. Conceptual Illustration of Air gap on SCADA (https://us-cert.cisa.gov/).

security. This was proved by the Stuxnet attack against an Iranian nuclear power plant in 2010 (Langner, 2011b), while ICS security loopholes in many facilities worldwide have been disclosed online.

The PLC that directly controls field devices in an ICS facility loads a separate program code called "ladder logic" from the EWS and operates the field devices according to a set rule. However, there are many security vulnerabilities in this system structure. For instance, the PLC may become a target of a direct cyber-attack using an unauthorized network communication session. A typical example of this kind of attack is the code injection attack, which attacks the PLC memory to inject a modified code. Other types of attacks use the EWS directly connected to the PLC. For example, the Stuxnet attack against an Iranian nuclear facility in 2010 used malware to attack the EWS first, then altered the ladder logic and transmitted it to the connected PLC to cause a centrifuge malfunction by manipulating the number of rotations. These examples show that attacks against the ICS do not distinguish between field devices and the operating layer, making it necessary to detect anomalies in advance in order to maintain availability. Since such attacks use the operating layer, i.e. the EWS, to attack field devices that are difficult to penetrate, special cyber safety measures are necessary.

4. Siemens TIA (totally integrated automation) Portal Step 7

4.1. Siemens TIA Portal Step 7

Siemens, Europe's largest engineering company, provides intelligent infrastructures for buildings and distributed energy systems, automation and digital solutions for process and manufacturing industries, and smart mobility solutions for railway and road transportation (https://new.siemens.com). Siemens' TIA Portal Step 7 is an integrated development environment tool for configuring and developing Siemens' PLC logic programs and monitoring the status of PLCs (see Fig. 2). TIA Portal Step 7 allows remote connection to a PLC or HMI hardware through the Ethernet network and provides various functions, such as the transmission of program logic specifically developed according to the given requirements, change of setting parameters, PLC restart, and process monitoring.

4.2. PEData file structure (Chan, Chow)

TIA Portal Step 7's project files are divided into data files and index files. The project files are generated when a new project is

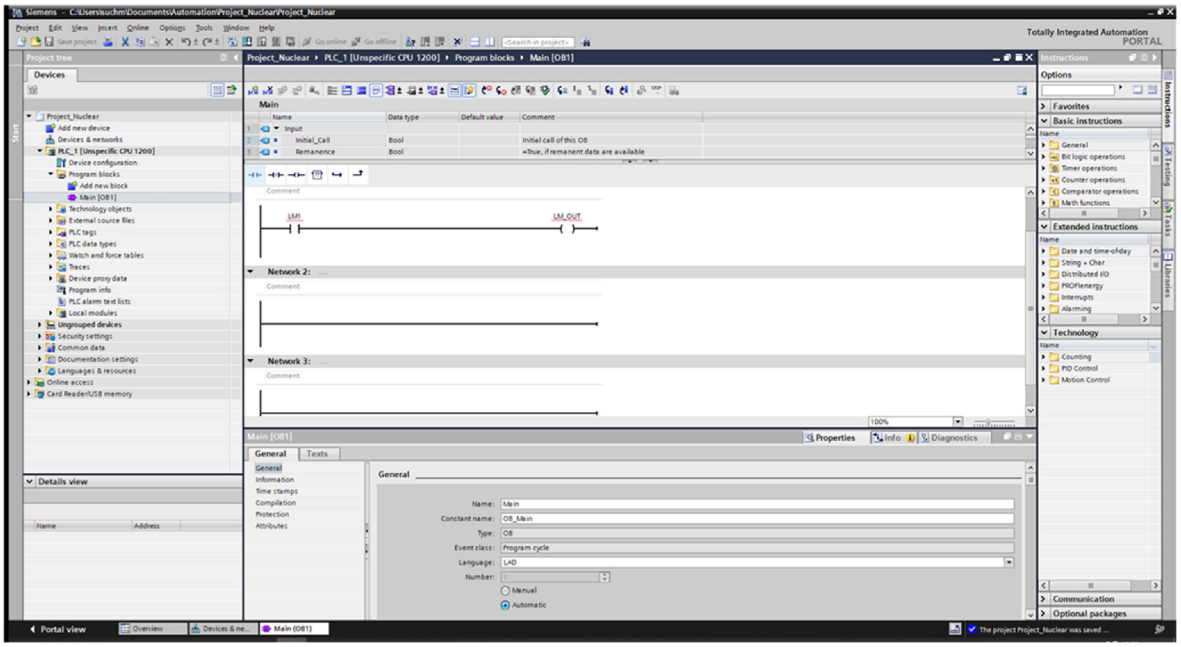

Fig. 2. Siemens TIA Portal Step 7 software.

PEData.plf

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0007CC50	54	02	00	00	6F	02	00	00	91	02	00	00	D5	02	00	00
0007CC60	49	03	00	00	12	00	00	00	08	00	00	00	0A	50	4C	55
0007CC70	53	42	4C	4F	43	4B	B5	00	00	00	48	00	00	00	7D	06
0007CC80	66	ED	F7	95	D8	48	30	00	00	00	00	00	9E	1C	EA	34
0007CC90	FC	95	D8	48	33	00	00	00	31	00	00	00	00	00	38	00
0007CCA0	00	00	32	00	00	00	01	01	01	05	4D	61	69	6E	10	4F
0007CCB0	42	2E	50	72	6F	67	72	61	6D	43	79	63	6C	65	6D	6C
0007CCC0	00	00	00	5C	00	00	00	FF	FF	FF	FF	02	00	00	00	09
0007CCD0	04	FF	FF	1C	00	00	00	3C	00	00	00	1C	00	00	00	22
0007CCE0	4D	61	69	6E	20	50	72	6F	67	72	61	6D	20	53	77	65
0007CCF0	65	70	20	28	43	79	63	6C	65	29	22	1C	00	00	00	22

Fig. 3. PLUSBLOCK signature and Main Organization Block Cycle on PEData.plf file.

created in the TIA portal and stored in “C:\Users\{account name}\Documents\Automation\{project name}\System\“. PEData.plf is a data file that contains the actual program code blocks, such as the ladder logic. PEData.idx contains the index data for the project data and is stored in the same path as the plf file. The project files contain all information about a PLC's program code blocks and configuration. In particular, since the PEData.plf file contains the PLC information as a general text, anyone can directly check the internal data. Fig. 3 shows the project tasks stored in a project file. It also shows that a new area that includes signatures, such as PLUSBLOCK and Commit, is added to the end of the project file whenever a project file is edited and stored. In other words, it means the “PLUSBLOCK” provides the information about what changes have been applied to the PLC program (CHAN, 2018). Forensic investigators can identify what has been changed by analyzing tags and ladder logic areas in PLUSBLOCK.

Since each update of new data is denoted by the end of file (EOF) in the project file, it is relatively easy to identify the modified area.

4.3. Detection of project data file updates

As described above, a user's project tasks are sequentially added at the end of the PEData.plf file (Chan, Chow). Anyone who has access to the file can detect the project file update events. As shown in Fig. 4, the addition of a problem block's ladder logic increased the file's size by 200,425 bytes. Checks of the binary code before and after the update showed that 200,425 bytes of data were added next to the EOF. However, it is difficult to determine whether the modification event in the PEData.plf was a project task update undertaken by an authorized user through the TIA portal, or whether the EWS was infected by malware and altered maliciously. Therefore, it is necessary to monitor all changes to a project file, collect and store the project file safely, attribute information before the modification, and then restore the data to its normal state as and when necessary. All this requires an automated tool capable of detecting changes to project files, collecting data, and restoring data to its original state (see Fig. 5).

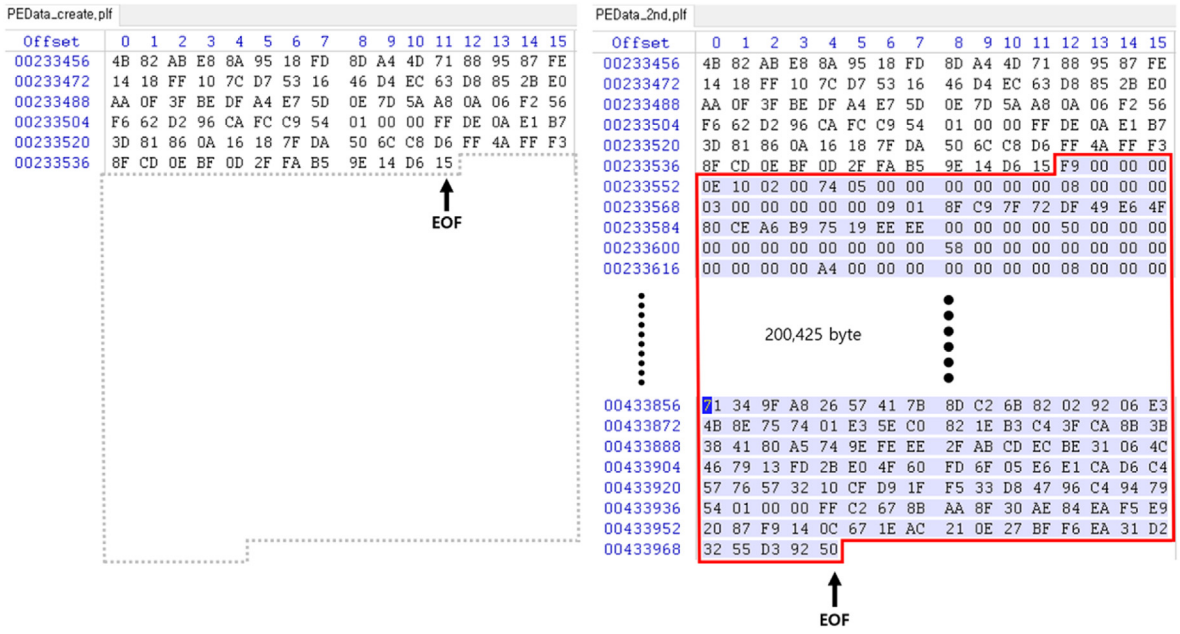

Fig. 4. Changes inside the file before and after PEData.plf modification (Modified Part is added to EOF).

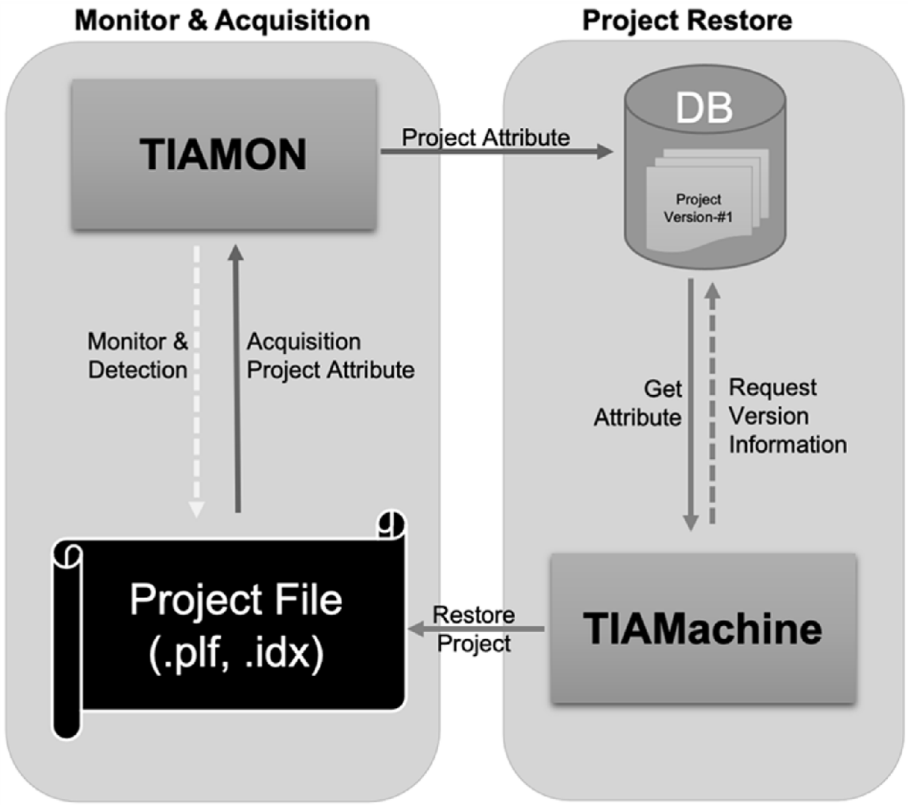

Fig. 5. Concept diagram for project file change detection, acquisition, restoration process.

5. Development of detection and recovery tool

5.1. Detection

This study proposes a tool for collecting, retaining and recovering files by monitoring changes to project files. Developed with Python programming language, the tool consists of three parts: 1) TIAMon, which detects project file updates and collects and retains the related data; 2) a database for storing the attribute information and data of the obtained project file; and 3) TIAMachine, which conducts the recovery of the project file. The following figure shows its overall configuration.

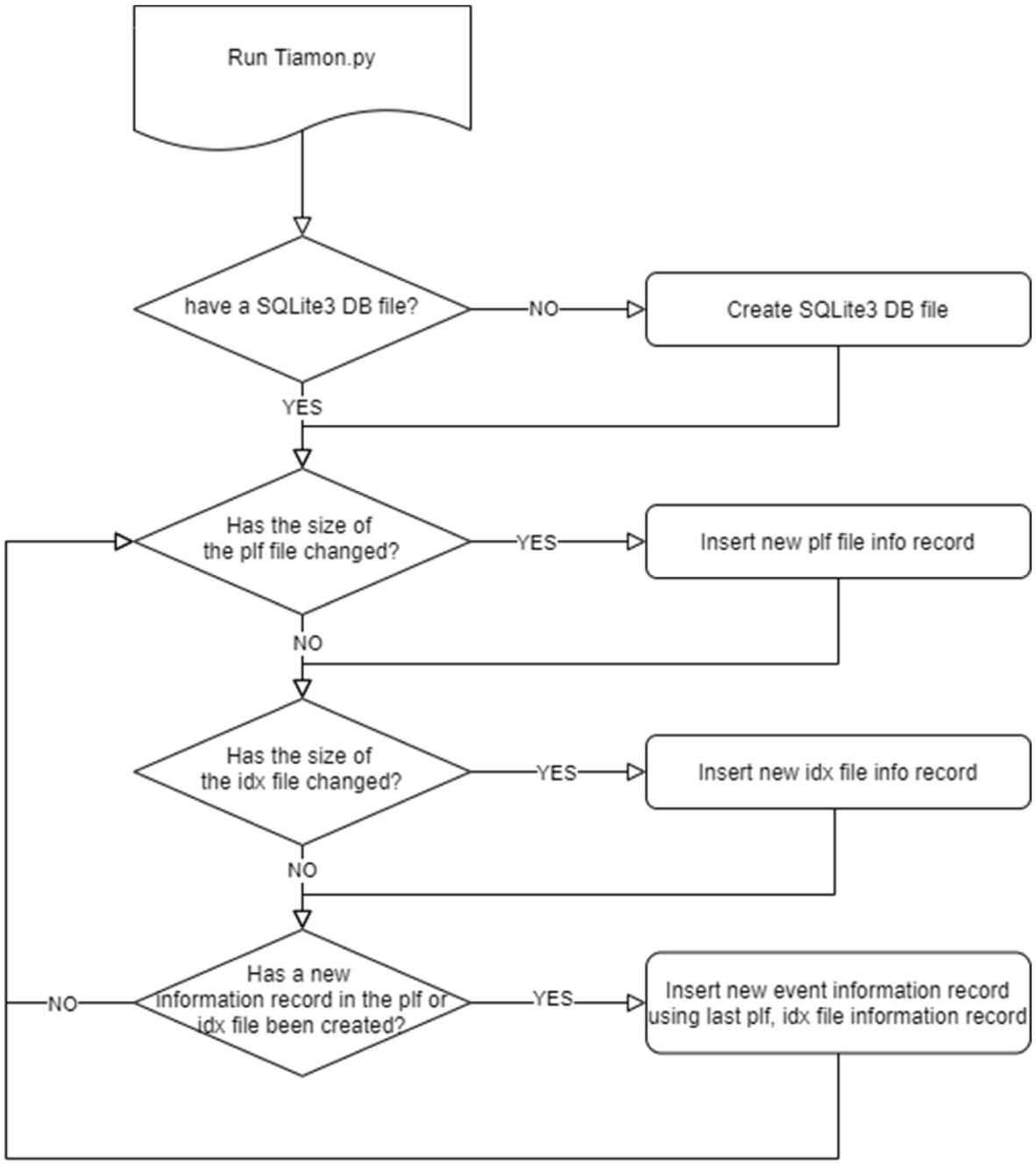

Fig. 6. Flow Chart of TIAMon after the program launching.

First, the detection tool (TIAMon.py) detects any changes to a project file. It also detects all update events of modification, whether a normal user task by TIA Portal Step 7 or a malicious alteration attack (see Fig. 6). Moreover, it was developed to detect even cases in which an attacker directly accesses project files (.plf and.idx) and modifies the binary code without going through the TIA portal. The detection tool is designed to monitor project files continuously to detect changes.

5.2. Project file collection and storage

When a project file is updated, TIAMon acquires the attribute information of the latest version of the project file pair (PEData.plf and PEData.idx) before the update. It then calculates the hash value of the acquired data to maintain the integrity of data and linked storage in preparation for any possible corruption that could occur while keeping the collected data. The collected data – including event time, hash value, and version - are stored in a pre-built database. A new database is generated if there is no existing one, and the collected data are stored in it. Table 1 below shows the database schema for storing the attribute information of the collected project files.

5.3. Recovery

TIAMachine is responsible for project file recovery. When TIAMachine runs, it determines which version of the project file needs to be restored using the project information and event time information stored in the database, and then proceeds with the file

Table 1
TIAMon.db database schema.

TABLE	COLUMNS	TYPE	DESCRIPTION	CONSTRAINT	CONDITION
MON_Projects	pid	Integer	Project ID	PRIMARY KEY	AUTO_INCREMENT
	projname	Text	Project Name	NOT NULL	
MON_Events	eid	Integer	Event ID	PRIMARY KEY	AUTO_INCREMENT
	pid	Integer	Project ID	NOT NULL	
	time	Text	Event Occur Time	NOT NULL	
	plf_fid	Integer	.plf File ID	NOT NULL	
	idx_fid	Integer	.idx File ID	NOT NULL	
MON_Files	fid	Integer	Project File ID	PRIMARY KEY	AUTO_INCREMENT
	pid	Integer	Project ID	NOT NULL	
	time	Text	File Acquisition Time	NOT NULL	
	path	Text	File Path	NOT NULL	
	size	Integer	File Size(byte)	NOT NULL	
	MTime	Text	Modification Time	NOT NULL	
	ATime	Text	Last Accessed Time	NOT NULL	
	CTime	Text	Create Time	NOT NULL	
	md5	Text	MD5 Hash Value		
	sha1	Text	SHA1 Hash Value		

recovery procedure (see Fig. 7). The project file version information can be requested in the order of project → event. TIAMachine uses the requested project file's version information to query the file size and two hash values from the MON-Files table in the database. It can use this information to carve the recovery area from the original project file and calculate the hash value. It then compares them with the hash value stored in the database to verify the project file's integrity and restore it (see Fig. 8) (see Fig. 9).

6. Experiment results

6.1. Setting the experiment environment

The test was conducted to verify the aforementioned method of detecting project file updates and collecting and recovering data based on a scenario involving an attack on the project file. TIAMon and TIAMachine (presented in Section 4) were used for detection and recovery. Since Siemens TIA Portal's project data storage and management functions have remained the same from past versions to the current one, Ver. 15, the highest version for which the developer's technical support is available, was selected as the test subject. Table 2 summarizes the test environment, including the operating system version, the file system of the tested EWS computer, and the database attributes used by the collection and recovery tools.

6.2. Scenario

The scenario involving an attack on a TIA portal project file was configured using the part on performing an attack that targets EWS

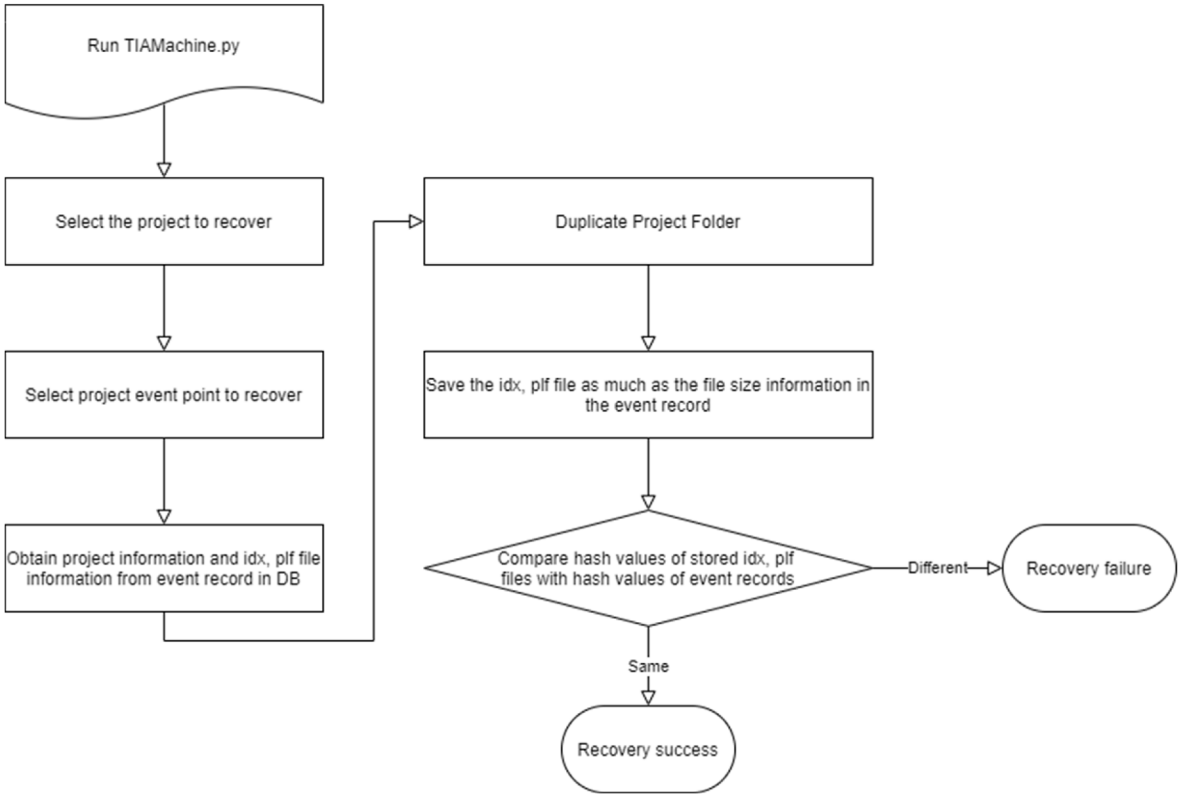

Fig. 7. Flow chart of TIAMachine after program launching.

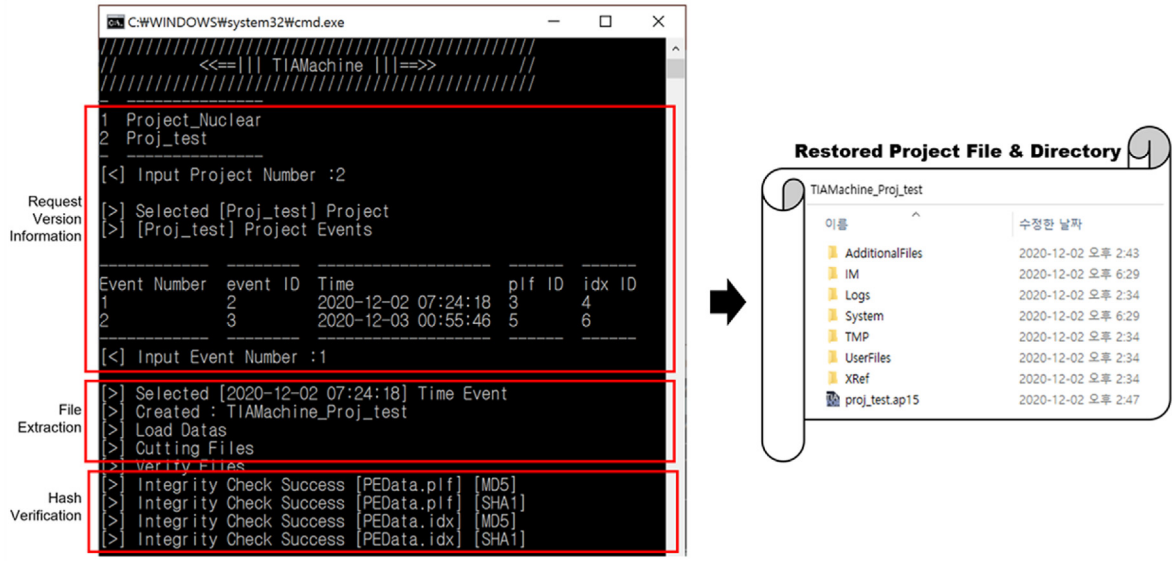

Fig. 8. Project file recovery process and results using TIAMachine.

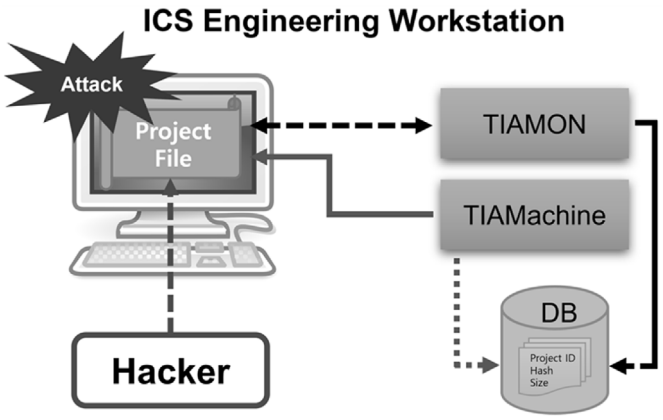

Fig. 9. Diagram of experiment scenario.

Table 2
Properties of experiment environment.

Category	Property	Value
EWS	Operating System	Microsoft Windows 10
	Platform	NT INTEL X86
	File System	NTFS
	TIA Portal	version 15
	Step 7	version 15
	WinCC	version 15
	Connected PLC	SIMATIC S7 1200
	Project Name	“tiamon_exam”
TIAMon	Development Language	Python v2.7
TIAMachine	Database	SQLite v3.7

of Stuxnet. Stuxnet, which was used to compromise the centrifuges of an Iranian uranium concentration program in 2010, attacked the SIMATIC Step 7 system of the EWS, the development environment, in an attempt to alter the PLC code block (Langner, 2011b). Moreover, it used an authorized engineer's USB, a portable storage medium, to infect the SIMATIC Step 7 system and overcome the air-gap of the Siemens SIMATIC system (Park, 2010). It infected the EWS through a user's worm-infected USB and altered the dynamic link library (DLL) related to the control logic program to manipulate the code block which set the centrifuge rotations, thereby causing the field device to malfunction. The infection activity was initiated by creating a copy of a Stuxnet infection program with the same name as any DLL file in the project directory (Park, 2010). When the Step7 software loaded the project, a malicious DLL file that was a clone of the Stuxnet infected program was loaded due to a vulnerability that could allow insecure library loading to execute remote code (MS Security Advisory #2262269). The attack and test scenario can be summarized as follows.

1 The malware infects the EWS and attempts to alter the project file to modify the PLC control logic.
2 The monitoring tool detects the alteration of the project file, collects the project file data before the alteration, and saves it in a separate database.
3 The project file is restored using the recovery tool, and the success of the recovery attempt is checked by comparing the project file hash values before and after the alteration attack.

6.3. Experiment results

To conduct this test, the project file was altered in two ways, and the same test was conducted to detect, collect, and recover the control logic using the TIA portal and to alter the project file by directly accessing it. In other words, an additional test was conducted to detect a direct modification of the project file's binary through file writing.

First, it was observed that the proposed tool was able to detect a file update when a part of the control logic was modified through the TIA portal. The test also confirmed that the project file attribute information and the hash value were stored in the database. It also confirmed that the hash values before and after the modification matched (see Fig. 10), and the TIA portal operated normally using the restored file.

Next, the tool also detected an alteration when the project file PEData.plf was directly modified in a random area with a hex editor. The recovery test also generated the same result as that obtained following modification using the TIA portal (see Fig. 11).

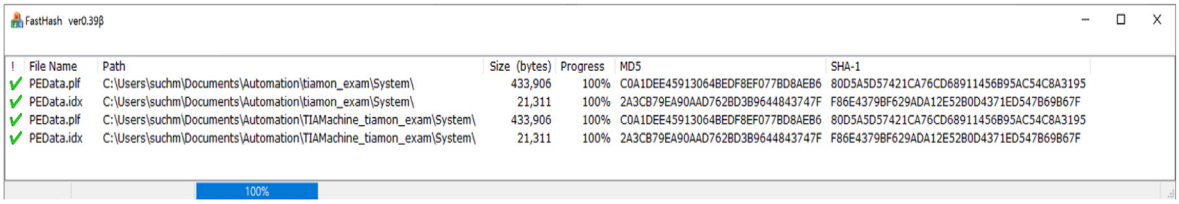

Fig. 10. Hash value comparison of modification-restoration project files using TIA Portal.

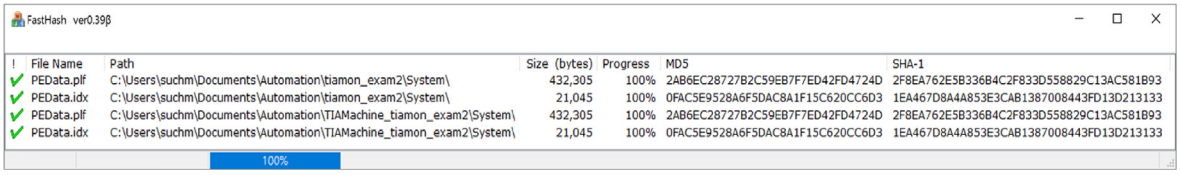

Fig. 11. Comparison of hash values of modification-restoration project files using binary direct modification.

7. Conclusion

The exposure of an ICS to cyber-threats could inflict serious disasters on our society. However, since the ICS has a limitation in that its availability must be the highest priority, it may not be feasible to conduct a forensic investigation of a PLC while it is being operated in the field. Therefore, it is necessary to continue researching methods of data collection and investigation without affecting the availability of devices, such as PLCs, which are directly connected to field devices. As such, this paper proposes a method of detecting and collecting data for the PLC forensic investigation of the ICS components. It studied the method of detecting and restoring the alteration attack of TIA Portal Step 7's project file used to control a PLC from an EWS and developed monitor and control logic. This study developed a tool for monitoring alterations and updates of a project file, collecting various kinds of data attribute information of the project file and storing them in a form that guarantees their integrity, and restoring the project file. In addition, a test environment was built to verify the developed tools, and it confirmed the detection of changes to the project file and its restoration in the following two cases: i) alteration of the control block by an authorized user, and ii) direct access to and alteration of the project file. Although we proposed a technique to respond project file manipulation attacks, the research result does not include performance and security tests for it. If the system impact, performance, and security test results for it are successful, it is thought that it will be very useful in practice. Future studies will be focused on an evaluation of the developed tool's performance using official data, such as its efficiency of computer resources and recovery speed, and its security.

Declaration of competing interest

To the best of our knowledge, the named authors have no conflict of interest, financial or otherwise.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2020R1A2C1012187, 60%), Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government(MSIT) (No. 2021-0-01806, Development of security by design and security management technology in smart factory, 40%).

References

Aquino-Santos, R., 2010. Emerging technologies in wireless ad-hoc networks: applications and future development: applications and future development. IGI Global 43.
Bronk, Christopher, Tikk-Ringas, Eneken, 2013. The cyber attack on Saudi Aramco. Survival 55 (2), 81–96.
Chan, Ching-bon, 2018. Forensic and Security Analysis of Programmable Logic Controller. HKU Theses Online (HKUTO).
Chan, R., Chow, K.-P. Forensic analysis of a Siemens programmable logic controller, IFIP Advances in Information and Communication Technology, pp. 117–130.
Eden, Peter, et al., 2015. A forensic taxonomy of SCADA systems and approach to incident response. In: Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research. BCS Learning & Development Ltd., pp. 42–51
Formby, David, Durbha, Srikar, Beyah, Raheem, 2017. Out of Control: Ransomware for Industrial Control Systems. RSA Conference.
Gollmann, Dieter, 2013. Security for cyber-physical systems. In: Mathematical and Engineering Methods in Computer Science. Springer, pp. 12–14.
https://new.siemens.com.
https://us-cert.cisa.gov/ics/Control_System_Engineering_Workstation-Definition.html.
Humayed, Abdulmalik, et al., 2017. Cyber-physical systems security—a survey. IEEE Internet Things J. 4 (6), 1802–1831.
Kent, Karen, et al., 2006. Guide to Integrating Forensic Techniques into Incident Response, Special Publication 800-86. National Institute of Standards and Technology.
Khan, Rafiullah, et al., 2016. Threat Analysis of Blackenergy Malware for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid. 4th International Symposium for ICS & SCADA Cyber Security Research.
Langner, Ralph, 2011a. Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. 9 (3), 49–51.
Langner, R., 2011b. Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. Mag. 9 (3), 49–51.
Lee, Jae-il, Lee, Yong-joon, et al., 2020. A profiling case study to phishing mail attack group. J. Internet Comput. Serv. 21 (2), 91–97.
Park, Hyung-Geun, 2010. Detailed Analysis Report of Stuxnet. IBM Security, IBM Korea.
MS Security Advisory #2269637. Insecure Library Loading Could Allow Remote Code Execution. http://www.microsoft.com/technet/security/advisory/2269637.mspx.
Shin, Jiho, Seo, Jung Taek, 2019. Research trends of SCADA digital forensics and future research proposal. J. Korea Inst. Inform. Secur. Cryptol. 29 (6), 1351–1364.
Wu, Tina, Jason, R., Nurse, C., 2015. Exploring the use of PLC debugging tools for digital forensic investigations on SCADA systems. J. Digital Foren. Secur. Law 10 (4), 79–96.
Yau, Ken, Chow, Kam-Pui, 2015. PLC forensics based on control program logic change detection. J. Digital Foren. Secur. Law 10 (4), 59–68.
Yau, Ken, et al., 2018. A forensic logging system for Siemens programmable logic controllers. IFIP Int. Confer. Digital Foren. 331–349 (Chapter 18).