SECURITY USING PROGRAMMABLE LOGIC CONTROLLER

[1]Reshma Sulthana S S, [2]Shabana S S
[1,2] Lecturer
Department of Electrical and Electronics Engineering

Government Polytechnic, Bellary, Ballari, India

Abstract: With the convergence of computer technology and industrial networks, attackers are not limited to attacking only individual users' computers, turning to attack industrial control systems that can cause major infrastructure problems. Programmable Logic Controllers (PLC) are the core components of industrial control systems. Its safety has a profound impact on the safety of the entire industrial system. This paper firstly classifies the security research of PLC according to the structure and function, and expounds the existing security defects of PLC from the aspects of firmware security, operation security and program security. Then it summarizes and analyzes four types of security protection measures: the integrity of verification firmware, protocol security encryption, code formal verification, and program security defence detection. Finally, according to the overall safety of the industrial system and the actual development of the current PLC, we discuss the development trend of safety research.

Index Terms **- Distributed Control Systems, Firmware, PLC code injection.**

I. INTRODUCTION

Industrial control systems (ICS) are usually highly interconnected and interdependent systems that are widely used in key national infrastructure industries such as natural gas, electric power, and nuclear facilities. Therefore, the security of the ICS is the primary prerequisite for ensuring the normal operation of the infrastructure. Unlike traditional computer attacks, which only cause data leakage, network denial of service, and computer damage, attacks against critical infrastructure control devices can even destroy physical equipment and cause irreparable damage to enterprises and even countries. Since the "Stuxnet" virus outbreak, there have been dozens of attacks on industrial networks. Because PLC is the core component of ICS, it has also been found from events and literature that attacks are all around PLC. For example, in 2010, Iran's nuclear facilities suffered from the "Stuxnet" virus[1]. The attack made the logic of the PLC change, and caused huge losses to the Iranian nuclear program. At the end of 2015, the Ukrainian national grid suffered a "BlackEnergy" malicious virus attack[2], and the Supervisory Control And Data Acquisition (SCADA) system was hit so that a large amount of key storage data was destroyed. In November 2017, Schneider Electric's Triconex Safety Instrumented System (SIS) was attacked by malware "TRITON", which crashed the SIS system by attacking control components such as PLC, and attacked the Distributed Control Systems (DCS) to expand the impact of the attack. It caused many energy plants in the Middle East to stop production. It is known from the frequent attacks on ICS in recent years that since the replacement of the early relay control device by PLC, the PLC security problem is worthy of attention because the PLC that loses part of the security function to ensure the practicability has become increasingly unable to resist attacks from the network. Therefore, this paper will classify the structure and functionality of PLC, and discuss the research focusing on the security aspects of firmware, operation and program. Finally, we discuss the two aspects of attack and defense and accord the prospect of future PLC security research.

2. PLC OVERVIEW

PLC is a network physics system specifically designed to control industrial systems, and its hardware structure is similar to that of a microcomputer. It is a kind of programmable memory for storing programs internally, performing logic operations, sequence control, timing, counting and arithmetic. One of the main uses of PLC is to control physical equipment in industrial sites. Figure 1 shows the structure of the PLC.

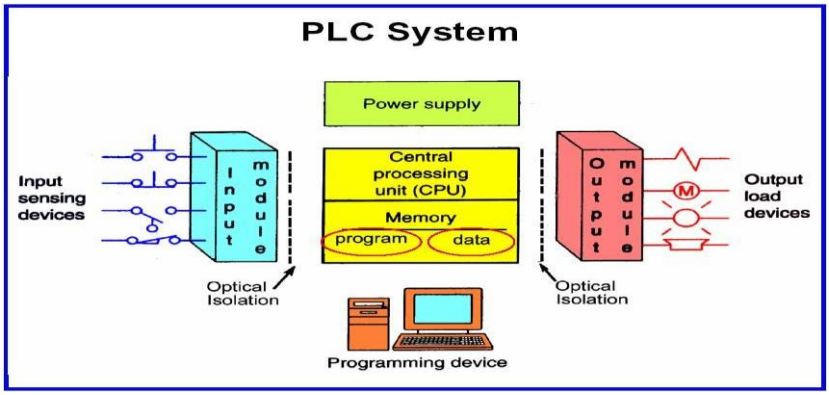

Figure1. Structure of PLC.

The internal PLC mainly consists of a power supply, a central processing unit, a memory, an input/output interface, a communication interface, and an expansion interface. Based on the internal structure of PLC and the functionality of link interaction, security issues can be studied in three ways:

2.1 PLC firmware

Firmware is the core of PLC, which determines the functional direction and performance of the device. It mainly includes the hypervisor and the instruction interpreter. It is written and solidified by the manufacturer in the memory, and the user cannot access and modify the system program. The function of the hypervisor is to manage the entire PLC so that the internal circuits can work in an orderly manner. The function of the instruction interpreter is to translate a user-written program into a program that the CPU can recognize and execute.

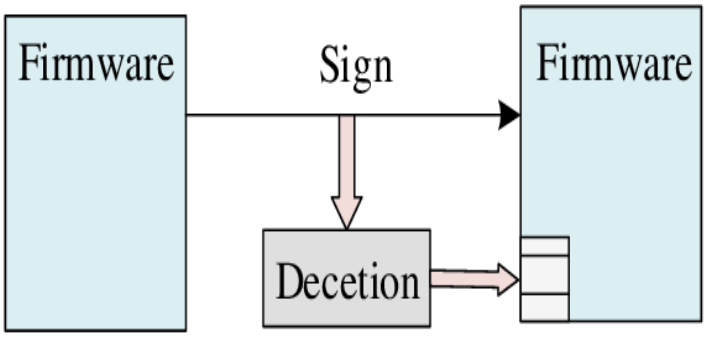

Figure2. Integrity of PLC Firmware.

2.2. The operational of PLC

The operational of PLC refers to the task of completing program delivery in a firmware environment. It usually includes the input and output of the status signal exchanged with the peripheral device through the I/O interface; using a proprietary communication protocol to realize the communication with the monitor, the host computer, or other devices; or other operations.

2.3 Program control flow of PLC

The program control flow of PLC mainly refers to the execution process of the running process, usually serving the system and hardware, and plays a vital role in the logic, communication, interaction and connection of the entire PLC. Figure 3 shows the flowchart of Program control flow of a PLC.

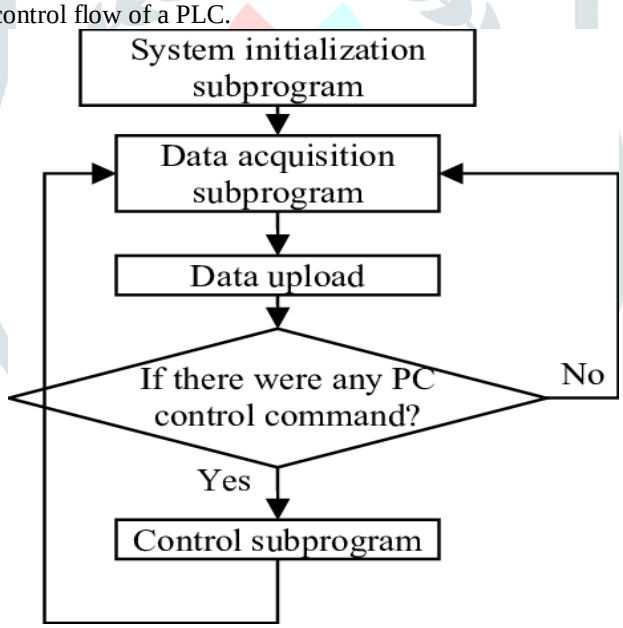

Figure3. Program control flow of PLC.

3. PLC SECURITY DEFECTS CLASSIFICATION

At the BlackHat European Conference in November 2016, according to the structure of the PLC, Ali Abbasi[4] proposed three attack methods for PLC, namely Firmware Modification Attacks (FMA), Configuration Manipulation Attacks (CMA), and Control-flow Attacks (CFA). Like traditional computers, PLC has the security problems of traditional protocol communication and configuration.

Because PLC is also a series of embedded devices, there are also problems such as security defects, memory corruption, and data signal storage. Therefore, the safety of PLC is becoming more and more serious. This section systematically studies the security flaws of PLC from three aspects: PLC's firmware, operation and program.

3.1 PLC firmware security defects

PLC firmware is vulnerable to Firmware Modification Attacks (FMA), which is caused by an attacker replacing a legitimate functional firmware with malicious firmware. For devices with reprogrammable firmware, the attacker has the opportunity to upload malicious firmware to the device because updating firmware requires appropriate access to the device. The firmware layer is the core of the bridging operation layer and user program, and is often regarded as the operating system of the embedded device. In a broader sense, the firmware also includes lower-level functions such as initialize and loads the

operating system. In some embedded devices, the firmware is installed at the factory and the device cannot be reprogrammed by the user. However, PLC typically has a firmware update feature that enables vendors to fix bugs and upgrade firmware without requiring physical changes to the hardware. The attacker exploited the PLC firmware update feature to develop a firmware replacement attack and firmware tampering attack.

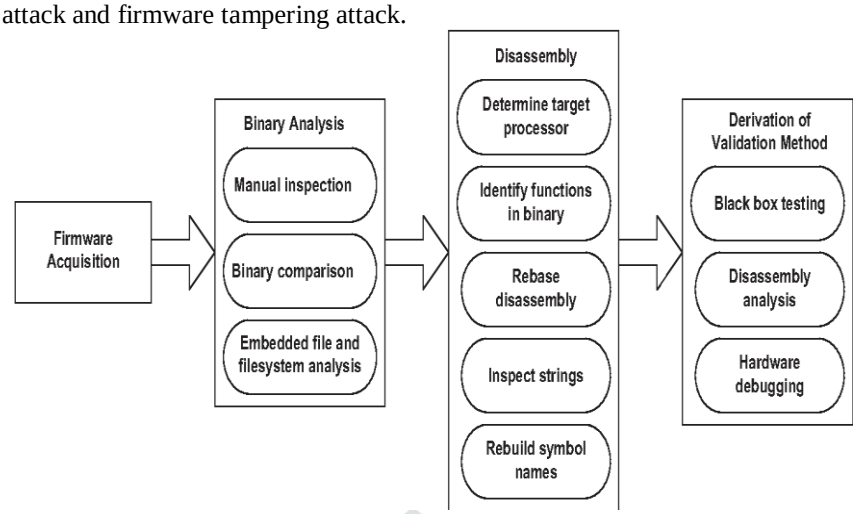

Figure4. Firmware Modification Attacks on PLC

3.2. PLC operation security defects During the operation, the PLC communicates with the PC control terminal through the network communication protocol, and accepts the instruction execution action of the host computer.
For PLC, code is the fundamental element of controlling PLC logic. As shown by figure 5, if the code is modified or bypassed, the established logic of PLC can be changed to achieve the attacker's desired purpose.

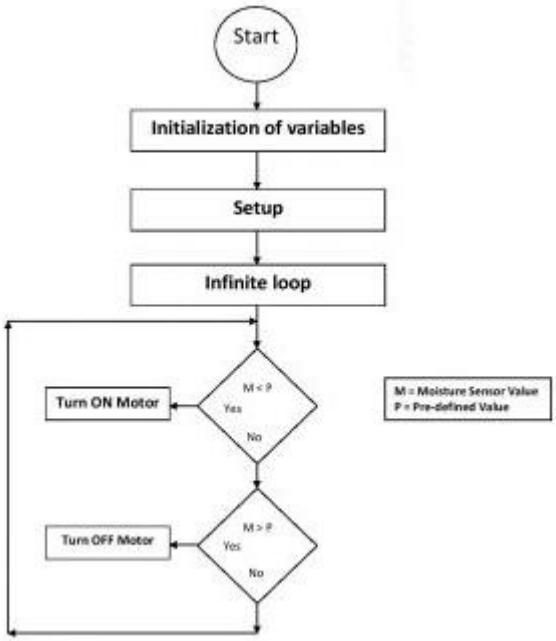

Figure 5. A flow chart of PLC code injection

Usually the attacker exploits the logic of the original code. For example, an object that is defined but not used can be hacked into an attacker's target object. The conditional competition vulnerability in some code can cause unpredictable competition errors, and even the loop in code may become unstoppable. The PLC code is usually programmed by engineers to implement functions, but often ignores the logic adaptation between codes. Therefore, it often suffers from serious problems such as MITM attacks, denial of service attacks, and malicious tampering of controller processes. The operation configuration defects of PLC are complex and concealed. Because of the endless attacks on the operation configuration, the operation configuration directly affects the normal operation of the controlled device. In the ICS, if only one shutdown occurs so that the execution process is changed, it may have serious consequences.

3.3. PLC program security defects
At present, the PLC program operation is not safeguarded by strong security protection measures like traditional PCs' programs, so it is vulnerable to Control-flow Attacks (CFA). In the case of a normal operation of the PLC controller, the attacker hijacking the control flow of the program, to make the running logic of the program violate the original design goal of the program. It is usually mainly through stack overflow vulnerabilities, release and reuse exploits, bypassing security mechanisms, allowing attackers to execute arbitrary code. Due to the similarity between embedded devices and real PCs, many studies have shown the possibility of controlling flow attacks in embedded devices. Beresford[20] found multiple vulnerabilities in Siemens PLCs that could allow an attacker to perform a remote code execution attack. Wightman[21] proof Schneider Electric PLC is vulnerable to buffer overflow attacks. Heffner[22] discovered multiple memory corruption

vulnerabilities router. Although there are a variety of techniques currently to detect or prevent control flow attacks, such attacks are still one of the most dangerous attacks. With the development of Internet technology, program flow hijacking attacks of traditional computers have spread in the industrial network, and no means have been developed that can effectively evade attacks without affecting the implementation of functions. In 2012, Vasilis Pappas[23] proposed the kBouncer technology to implement efficient and completely.

4. PLC SECURITY PROTECTION MEASURES

Based on the above analysis of the security defects of the PLC firmware layer, operation layer, and program security layer, this section will systematically explore four defense methods.

4.1 Verify the integrity of PLC firmware

Due to the similarity between PLC controllers and traditional computers, most firmware studies often follow the research methods of traditional computer programs, such as Drew Davidson[28], who proposed to use the symbolic execution method to check for vulnerabilities in the firmware program, and tested 99 open source MSP430 firmware programs and found 21 memory-related vulnerabilities. Jonas Zaddach[29] find a way to detect firmware. The article runs firmware on an emulator and interacts with physical I/O devices to dynamically detect vulnerabilities in the firmware. And in 2018 Marius Muench et al.[30] compared the firmware detection framework proposed by Jonas Zaddach and analyzed the shortcomings and challenges in each method. Another firmware study looks at the firmware itself, how to verify that the firmware itself has been replaced or modified. Mcminn et al.[31] presented a verification tool for PLC firmware in a SCADA system. The tool captures data during the upload and download phase of the firmware and is validated by known legitimate firmware, without any modifications to the SCADA system. In addition, it can analyze firmware using playback capture data without a PLC. Garcia[32] proposed an analysis technique that performs static differential analysis of suspected changed PLC firmware with good firmware, using a variety of test methods to compare firmware versions, models, and code differences, such as deleting, adding, or modifying existing in the original features. In the detection of the PLC firmware, it is difficult to achieve both the purpose of ensuring the security and ensuring that the performance is not reduced. The firmware detection method is usually adopted to ensure that the firmware is not replaced by detecting the integrity of the firmware, as shown in figure 6. Adelstein et al.[33] introduced a human-based signature-based detection method, which is tested its execution flow and integrity by the detector when it is running.

4.2 The security encryption of the PLC communication protocol

At present, most PLC communication protocols do not have mechanisms such as encryption and authorization authentication. Therefore, it is very convenient for an attacker to analyze the packets and construct malformed data to change the communication authentication, thereby achieving the malicious purpose of the attack. Achieved the authentication function is through the interaction of the handshake packet, so that some of the traffic packets intercepted by the attacker cannot be performed without authentication, as shown in figure 7. And the MAC address of the host computer is fixed. If the IP address and the MAC address are bound to the computer, it is difficult for the attacker to conduct a MITM attack from the third-party machine, as shown in figure 8. Nelson[34] proposed to bind the MAC address to ensure security.

4.3 The formal verification of PLC code

Usually, the defects of the PLC code are extremely difficult to find. The existing methods mainly rely on the security personnel to test the auditing method to avoid the existing problems. But gradually began to use the PLC code formal verification method, which can find a large number of logical defects in the code. The main purpose of code formal verification is to detect PLC code defects and avoid them from being invaded by malicious code. However, because PLC has many programming languages and is not a high-level language, the standards are different and the semantics are complex. It is difficult to analyze and correspondingly model. Saman Zonouz et al.[37] presented a study for PLC code analysis that used safety engineering to detect and characterize PLC infections for physical damage to power plants. It also draws on control theory, the engineering and mathematics field that deals with dynamic system behaviour, and reverse safety-critical code to identify complex and highly dynamic safety attributes for mixed code analysis methods. However, due to the high cost, it cannot be widely used in code analysis. Malchow[38] proposed the PLC Guard framework technology, which intercepts the flow between the engineering Random number Random number ID Checked Key Verify Content Client Server Firmware Logic Acquire signal Physical I/O Code PC、other PLC Program Update Register Control Control Ip、MAC conbined Information encryption AMIMA 2019 IOP Conf. Series: Materials Science and Engineering 569 (2019) 042031 IOP Publishing doi:10.1088/1757-899X/569/4/042031 9 workstation and the PLC. And Malchow used various levels of graphical abstraction and generalization for formal comparison, which helped the operation and maintenance personnel to correctly handle the accepted code commands, greatly reducing the analysis cost. Although there is no unified and effective framework for formal verification of PLC code, it is still an important and effective way for manual code auditing. But due to the complexity of its work and the difficulty of modelling, it is extremely difficult to extend the application, so the research prospects are still very broad.

4.4. The security defence detection of PLC program

Ali Abbasi et al.[39] presented a control flow integrity check tool to effectively detect control flow hijacking attacks while ensuring the real-time and availability of the PLC. As figure 9 shows, detect the assembler returns the address and the jump address, etc., and an alarm is issued when the control flow changes, which greatly ensures that the PLC program stream is not tampered with. In the realtime operating system, the priority of the detection task is lower than the priority of the control task, that is, the jump address is detected only when the CPU is idle, so that the real-time control effect of the PLC is ensured to the greatest extent. Saman Zonouz[40] proposed a method of using PLC code symbols to detect malicious code. Figure 9. The

principle of the detection assembler code The program flow control problem of PLC is still a problem that plagues security personnel. It is a research direction to learn the protection method of traditional computer programs, randomize the program address, prohibit execution of jumps, etc. However, the PLC program still has some different traditional PC programs. Therefore, control flow integrity detection is an effective means, but the technical means of accurately detecting and reducing overhead has always been the research direction of researchers.

5. CONCLUSIONS AND DISCUSSIONS

In summary, there are a large number of research scholars on the safety protection of PLC, but the security protection of PLC is a whole system engineering, and it is not possible to conduct one-sided research from a certain aspect. A simple study from a certain aspect cannot completely protect the vulnerability of the PLC, and will increase the cost. Therefore, in order to protect the safety of PLC, the key research directions of the future research on PLC security research are as follows:

(1) In terms of defence, develop a unique protection framework for PLC holistic research, comprehensively consider cost and functionality, and protect the integrity of PLC integrity from being destroyed.

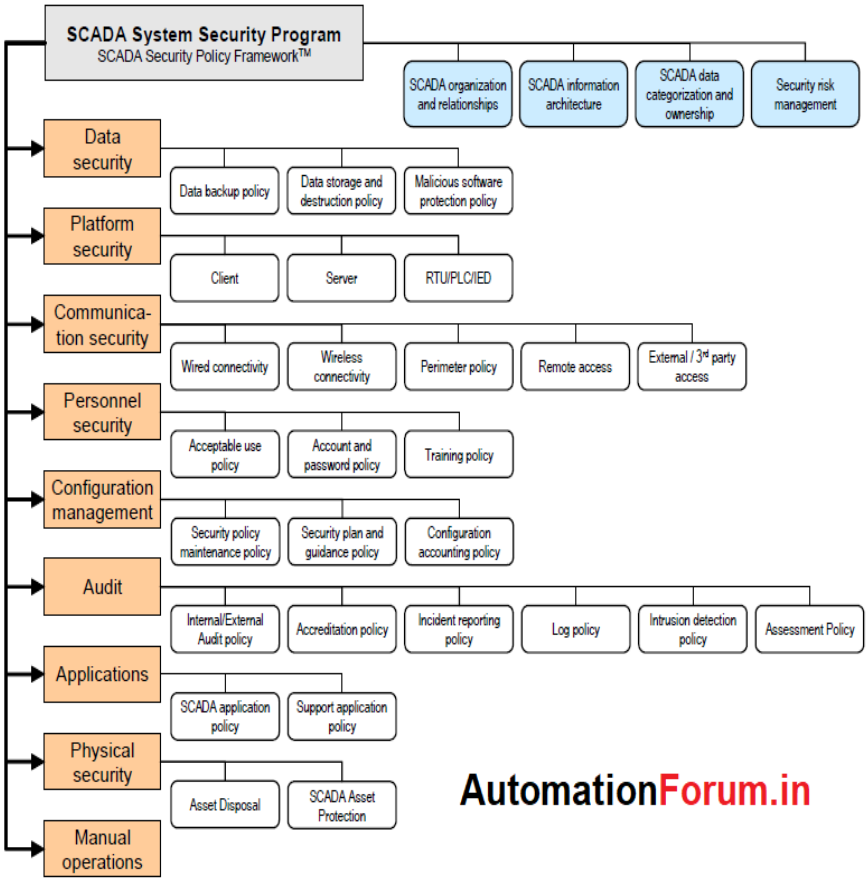

Figure 6. PLC security framework

Figure 6 shows a PLC security framework. A protection framework for PLC holistic research First of all, this framework has information encryption to prevent sensitive data from being stolen. Communication transmission encryption is carried out to ensure the authenticity of the data. Secondly, the PLC firmware program is signed online and verified by the detection device to ensure the integrity of the firmware and avoid the tampering of the firmware program. Finally, there should be a test record for the execution flow of the program to prevent tampering of the program control process. Need to consider the structure of PLC, study how to protect the system, and ensure the functional integrity of PLC in real time.

(2) In terms of attacks, the entire system industrial process of ICS needs to be considered to further evaluate the security of the PLC. Nowadays, simply studying the security problem from the structure of PLC itself cannot systematically analyze the existing problems. Any kind of ICS security incident not only attacks the vulnerability from PLC itself, but studies its attack link to form a complete problem. The killing chain completes the attack, achieving the most destructive and influential. Study How to use the Human Machine Interaction (HMI) to attack PLC, to deceive engineering workstations to attack PLC, etc. From the ICS whole system to study its security is the focus of future research.

REFERENCES

[1] James, P., Farwell, Rohozinski, R. (2011) Stuxnet and the Future of Cyber War. Survival, 53 (1): 23 - 40.

[2] Assante, M.J. (2016) Confirmation of a Coordinated Attack on the Ukrainian Power Grid. SANS Institute, Bethesda, USA, Jan. http://ics.sans.org/

[3] Erickson, K. (2010) Programmable logic controllers: Hardware, software architecture. https://www.isa.org/standardspublications/isapublications/intechmagazine/2010/december/automation-basicsprogrammable-logiccontrollers-hardware-softwarearchitecture.

[4] Abbasi, A., Hashemi M. (2016) Ghost in the PLC Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack.

[5] Peck, D., & Peterson, D. (2009). Leveraging ethernet card vulnerabilities in field devices. In: SCADA security scientific symposium. pp. 1-19.

[6] Basnight, Z., Butts, J., et al. (2013) Firmware modification attacks on programmable logic controllers. International Journal of Critical Infrastructure Protection. 6(2):76-84.
[7] Schuett, C., Butts, J., Dunlap, S. (2014) An evaluation of modification attacks on programmable Firmware Logic Physical I/O Code PC、other PLC Program Register Control Control Ip、MAC conbined Information encryption Communication encryption Random number signed Detection Record、 Detection、 Warning AMIMA 2019 IOP Conf. Series: Materials Science and Engineering 569 (2019) 042031 IOP Publishing doi:10.1088/1757-899X/569/4/042031 11 logic controllers. International Journal of Critical Infrastructure Protection. 7(1):61-68.
[8] Garcia, L., Brasser, F., Cintuglu, M.H., et al. (2017) Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit. In: NDSS.
[9] Beresford, D. (2011) Exploiting siemens simatic s7 PLCs. Black Hat USA, 16(2): 723-733. [10] Morris, T.H., Gao, W. (2013) Industrial Control System Cyber Attacks. In: International Symposium on ICS & Scada Cyber Security Research. BCS. 2013:22-29.
[11] Wardak, H., Zhioua, S., Almulhem, A. (2017) PLC access control: a security analysis. In: Industrial Control Systems Security. IEEE. 2017:1-6.
[12] Ponomarev, S. (2015) Intrusion Detection System of industrial control networks using network telemetry. Dissertations & Theses-Gradworks.
[13] Abbasi, A. (2016) Ghost in the PLC: stealth on-the-fly manipulation of programmable logic controllers' I/O. CTIT Technical Report Series, (TR-CTIT-16-02).
[14] Valentine, S.E. (2013) PLC code vulnerabilities through SCADA systems. In: University of South Carolina.
[15] McLaughlin, S.E. (2011) On Dynamic Malware Payloads Aimed at Programmable Logic Controllers. In: HotSec.
[16] McLaughlin, S., McDaniel, P., (2012) SABOT: Specification-based payload generation for programmable logic controllers. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, ser. CCS'12. New York, NY, USA: ACM, 2012, pp. 439– 449.
[17] Klick, J., Lau, S., Marzin, D., Malchow, J., Roth, V. (2015) Internet-facing PLCs - A New Back Orifice. In: Black Hat.
[18] Spenneberg, R., Brüggemann, M., Schwartke, H. (2016) PLC-Blaster: A Worm Living Solely in the PLC. In: Black Hat.
[19] Govil, N., Agrawal, A., Tippenhauer, N.O. (2017) On Ladder Logic Bombs in Industrial Control Systems.
[20] Beresford, D. (2011) Exploiting Siemens Simatic S7 PLCs. In: Black Hat. USA.
[21] Wightman, R. (2012) Project basecamp at s4. SCADA Security Scientific Sympo- sium. [Online]. Available: https://www.digitalbond.com/tools/basecamp/ schneider-modicon-quantum/ [22] Rapid7. (2014) Linksys wrt120n tmunblock stack buffer overflow. [Online]. Available: http://www.rapid7.com/db/modules/auxiliary/admin/http/linksys tmunblock admin reset bof
[23] Pappas, V. (2012). kBouncer: Efficient and transparent ROP mitigation. Apr, 1, 1-2.
[24] Fratrić, I. (2012). ROPGuard: Runtime prevention of return-oriented programming attacks. Technical report.
[25] Cheng, Y., Zhou, Z., Yu, M., Ding X., and Deng, R. H. (2014) ROPecker: A generic and practical approach for defending against ROP attacks. In: Proc. 21st Annual Network & Distributed System Security Sym. (NDSS).
[26] Schuster, F., Tendyck, T., Pewny, J., Maaß, A., Steegmanns, M., Contag, M., and Holz T. (2014) Evaluating the effectiveness of current anti-ROP defenses. In: Research in Attacks, Intrusions and Defenses, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Springer. pp. 88–108.
[27] Davi, L., Lehmann, D., Sadeghi, A.-R., and Monrose, F. (2014) Stitching the gadgets: On the ineffectiveness of coarse-grained control-flow integrity protection. In: USENIX Security Symposium.
[28] Davidson, D., Moench, B., Ristenpart, T., & Jha, S. (2013). FIE on Firmware: Finding Vulnerabilities in Embedded Systems Using Symbolic Execution. In: the 22nd USENIX Security Symposium (USENIX Security 13). pp. 463-478.
[29] Zaddach, J., Bruno, L., Francillon, A., & Balzarotti, D. (2014). AVATAR: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares. In: NDSS. pp. 1-16.
[30]https://forum-automation-uploads.sfo3.cdn.digitaloceanspaces.com/original/2X/4/47c9fca91e85616efa2b0958506f9.png.