SCADA SECURITY ACCESS CONTROL ATTACKS: NETWORK DETECTION

BY

HAROON ABDUL-ALEEM WARDAK

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SECURITY AND INFORMATION ASSURANCE

DECEMBER 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **HAROON ABDUL-ALEEM WARDAK** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN SECURITY AND INFORMATION ASSURANCE**.

Thesis Committee

[illegible]
Dr. Sami Zhioua (Adviser)

[illegible]
Dr. Ahmad Almulhem (Member)

[illegible]
Dr. Moataz Ahmed (Member)

[illegible]
Dr. Khalid Al-Jasser
Department Chairman

[illegible]
Dr. Salam A. Zummo
Dean of Graduate Studies

عمادة الدراسات العليا
جامعة الملك فهد للبترول والمعادن
DEANSHIP OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
1963

13/6/17
Date

©Haroon Wardak
2016

To my parents. To my family. To my wife.

ACKNOWLEDGMENTS

All praise be to Allah, the Creator and Sustainer of the worlds.

I would like to convey my great appreciation to my advisor Dr. Sami Zhioua for his guidance, support, encouragement and patience all the time I spent in my thesis work. I would like to thank Dr. Ahmad Almulhem and Dr. Moataz Ahmad for their valuable comments and guidance.

I would also like to express my gratitude to all of my friends in KFUPM, especially Afghan friends. I would like to thank Mr. Asem Ghaleb for many hours of fruitful discussions.

I owe an eternal debt of gratitude to my family for all the unconditional love, support and encouragement they provide me. I pray that they live long and happy lives. I would like to thank my brother Hamed for taking all my responsibilities at home beside of his own.

Last but not least I would like to acknowledge the support of King Fahd University for Petroleum and Minerals for giving me this opportunity to pursue my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SCADA	:	Supervisory Control and Data Acquisition
ICS	:	Industrial Control Systems
PLC	:	Programmable Logic Controller
RTU	**:**	Remote Terminal Unit
HMI	:	Human Machine Interface
MTU	:	Master Terminal Unit
Profibus	:	Process Field Bus
IDS	:	Intrusion Detection system
MiTM	:	Man-In-The-Middle
DoS	:	Denial-of-Service
IT	:	Information Technology
TCP / IP	:	Transmission Control Protocol / Internet Protocol
LAN	:	Local Area Network

THESIS ABSTRACT

NAME:	Haroon Abdul-Aleem Wardak
TITLE OF STUDY:	SCADA Security Access Control Attacks: Network Detection
MAJOR FIELD:	Security and Information Assurance
DATE OF DEGREE:	December 2016

Supervisory Control and Data Acquisition (SCADA) systems are used in many critical infrastructures to provide 24/7 system monitoring and control different type of operations. Thus, they control nations vital assets such as water desalination, power plants and nuclear power generation. With the aim of enhancing efficiency and reducing costs, SCADA systems are connected with corporate networks or Internet. However, such connectivity leads to significant increase in security attacks against SCADA systems because of the lack of proper and dedicated security solutions for such systems. Therefore, it is important to protect SCADA systems and install countermeasures against cyber attacks.

A Programmable Logic Controller (PLC) is a very common industrial control system device used to control output devices based on data received (and processed) *from input devices. Given the central role that PLCs play in deployed industrial control systems, it has been a preferred target of ICS attackers. A quick search in the ICS-CERT repository reveals that out of a total of 589 advisories, more than 80 target PLCs. Stuxnet attack, considered the most famous reported incident on ICS, targeted mainly PLCs. Most of the PLC reported incidents are rooted in the fact that the PLC being accessed in an unauthorized way. This research aims at studying SCADA network attacks and develop detection and mitigation techniques. We investigated the PLC access control problem. We discussed several access control models but we focused mainly on the commonly adopted password-based access control. We showed how such password-based mechanism can be compromised in a realistic scenario as well as the list the attacks that can be derived as a consequence. In addition, we created set of rules to detect any attempt to attack the PLC.*

ملخص الرسالة

الاسم الكامل: هارون عبدالعليم وردك

عنوان الرسالة: الهجمات الأمنية ضد آلية حماية الوصول في أنظمة سكادا: استكشاف الشبكة

التخصص: أمن وضمان المعلومات

تاريخ الدرجة العلمية: ديسمبر 2016

تستخدم أنظمة التحكم الإشرافية وتجميع البيانات (سكادا) في العديد من البنى التحتية الحيوية لتوفير مراقبة النظام والسيطرة على أنواع مختلفة من العمليات على مدار الساعة وبالتالي، فإنها تنظم وتتحكم على منشآت الدولة الحيوية مثل تحلية المياه ومحطات توليد الكهرباء وتوليد الطاقة النووية. وبهدف تعزيز الكفاءة وخفض التكاليف، تُوصّل أنظمة سكادا بشبكة المنشأة والإنترنت. ولكن هذا الربط يؤدي إلى زيادة كبيرة في الهجمات الأمنية ضد أنظمة سكادا بسبب عدم وجود حلول أمنية مناسبة ومخصصة لمثل هذه الأنظمة. ولذلك، من المهم جداً حمايتها واستخدام التدابير المضادة ضد الهجمات السيبرانية.

جهاز التحكم (Programable Logic Controller) هو جهاز تحكم صناعي يستخدم للتحكم في المشغلات الميكانيكية استناداً إلى البيانات المستلمة والمعالجة من أجهزة الاستشعار. ونظراً للدور المحوري الذي تلعبه (PLC) في أنظمة سكادا المنتشرة، فقد أصبح الهدف المفضل للمهاجمين. كما تبين من خلال بحث سريع في قاعدة بيانات (ICS-CERT) أنه من أصل 589 تنبيهاً، أكثر من 80 تستهدف (PLC) بشكل خاص.

هجوم (Stuxnet)، الذي يعتبر الحادث الأشهر في تقارير الهجمات على سكادا، استهدف بشكل رئيسي (PLC). وكشفت هذه التقارير أن الهجوم على (PLC) يتم بطريقة اختراق آلية التحكم في الدخول إلى (PLC) بطريقة غير مصرح بها. لذلك، تهدف هذه الأطروحة لدراسة الهجمات على شبكة سكادا وتطوير تقنيات كشف الهجمات والتخفيف من آثارها. كما تهدف إلى التحقيق ومناقشة آليات التحكم في الدخول إلى (PLC). وركز البحث بشكل رئيسي على التحكم في الوصول المعتمد على كلمة المرور. كما أظهر البحث كيف يمكن اختراق هذه الآلية في تجربة واقعية وكذلك قائمة الهجمات التي يمكن أن تُنفذ نتيجة لهذا الاختراق. وبالإضافة إلى ذلك، تم إنشاء مجموعة من القوانين للكشف عن أي محاولة لمهاجمة (PLC).

CHAPTER 1

INTRODUCTION

Industrial control systems (ICS) is a general term denoting several computer-based control systems that involve Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS) and Process Control System (PCS). SCADA are process control systems which are commonly deployed to continuously monitor and control industrial facilities to guarantee proper functioning. SCADA is mainly composed of Human Machine Interface (HMI) that runs a SCADA software to present the actual status of ongoing operations and processes of industry. Data is being sent to HMI by Intelligent Electronic Devices (IEDs) such as Programmable Logic Controllers (PLCs) or Remote Terminal Unites (RTUs) through a communication network. IEDs are collecting data from sensors, send it to Master Terminal Unit (MTU) which is a central server processing the inputs and send command back to IEDs which in turn send it to an actuator to adjust the status of an equipment device (e.g. valve). Figure 1.1 illustrated typical SCADA network.

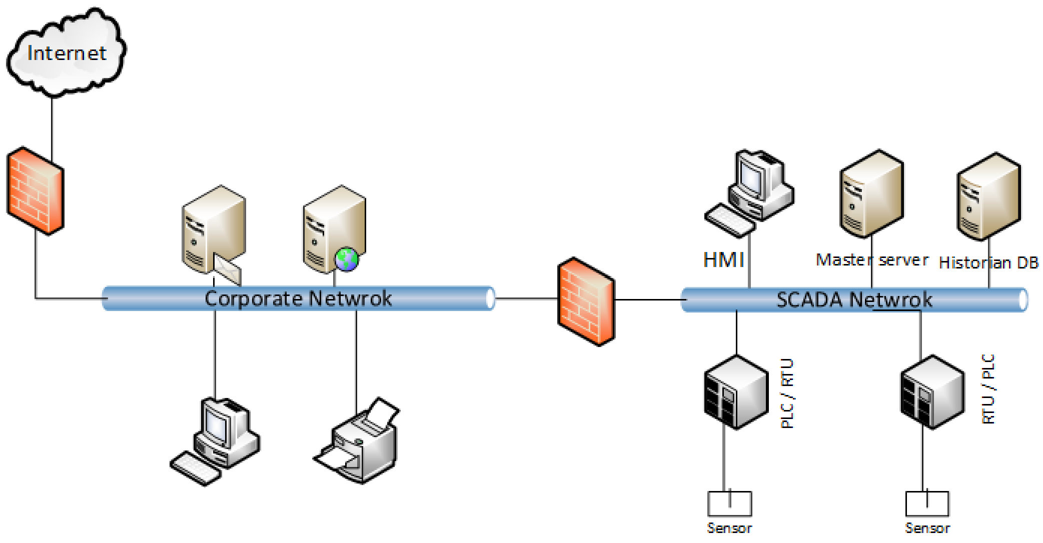

Figure 1.1: Typical SCADA Network Topology

These systems are used in many critical infrastructures to provide 24/7 system monitoring and control different type of operations. They are commonly used to control organization's and nation's vital assets such as water refineries, power plants, nuclear power generators, etc. A failure in these systems may lead to catastrophic consequences not only in terms of financial losses but also in terms of human lives.

1.1 Motivation and Scope

Historically, the priority was given to system reliability and availability while security was considered a minor issue. ICS systems security was relying on the use of proprietary devices and protocols without publicly available documentation (security by obscurity). In addition, ICS networks were typically isolated from other networks (corporate LANs, Internet, etc.). In recent years, most ICS systems became connected (directly or indirectly) to the Internet. This can be explained by the need for sharing real-time information with the business operations which is now a necessity in order to improve efficiency, minimize costs, and maximize profits. As a consequence, ICS devices are increasingly required to support mainstream protocols in order to communicate with a broader range of networks and devices. This, however, exposes ICS devices to various types of exploitation which is particularly relevant for Programmable Logic Controllers.

Over the past five years, there was a period of sharp growth in the number of vulnerabilities, reflecting that industrial systems are gaining the attention of not only the researchers and system owners but also potential attackers. the index has increased from 19 vulnerabilities in 2010 to 189 vulnerabilities in 2015. And this number keeps growing, more and more information about vulnerabilities in these systems is becoming public. For 26 of the vulnerabilities published in 2015, exploits are available [1].

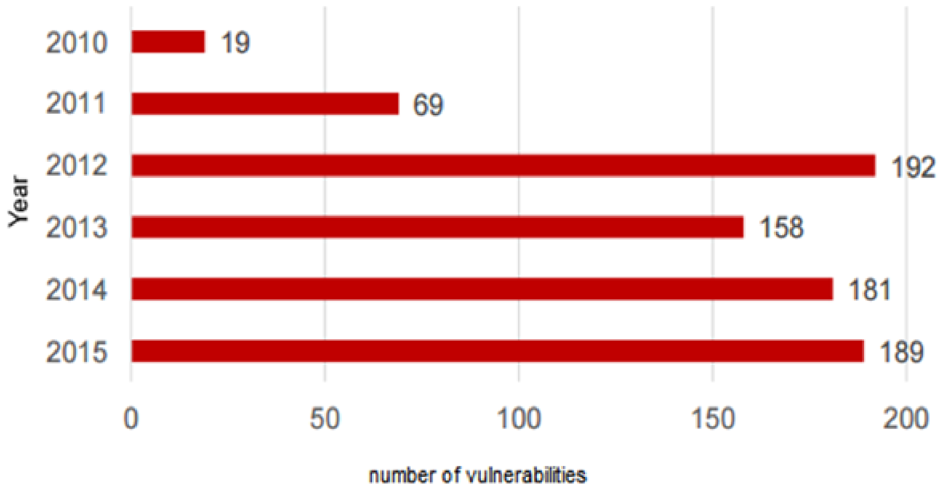

Figure 1.2: Number of Vulnerabilities over The Past Five Years

A Programmable Logic Controller (PLC) is an important component in an ICS system. It is a control device used to automate industrial processes via collecting input data from field devices such as sensors, processing it, then send commands to actuators devices such as motors. Being a pivotal device in ICS systems, PLCs are preferred target for cyber security attacks. ICS-CERT, the repository for ICS specific incidents, includes a large number of PLC related issues. A quick search performed in November 2016 reveals that out of a total of 589 advisories, 89 target directly PLCs and out of a total of 114 alerts, 17 involve PLCs. Another manifestation of the exposure of PLCs to cyber security attacks is the Stuxnet malware [2] which is designed to attack primarily PLCs of the Iranian nuclear facility.

PLC security issues range from simple DoS to sophisticated remote code execution vulnerabilities. Most of PLC attacks are possible because attackers could have access and compromise the PLC device. In addition, SCADA protocols suffer from insufficient authentication or integrity checking mechanism, which expose SCADA systems to cyber attacks when connected to external networks. That is because security was not the priority when these protocols were first designed. Recently, many researches have been conducted in the mean of detection the attempts of the attacks and develop techniques to prevent them from damaging the systems.

The communication patterns in SCADA network is assumed to be well-behaved, predictable flow patterns. ICS systems have fixed topology with limited number of network devices, protocols and application running. Most of SCADA communications are generated in a polling manner. The server demands a client for a data and a client starts a communication, i.e. HMI requests a PLC for sensors measurements or sends commands to the actuators. By this assumption, it is feasible to model normal operations traffic or signatures to discover if there is a malicious action that violate the models.

Intrusion Detection Systems (IDS) mainly uses one of two detection techniques: Anomaly-based IDS and Signature-based IDS. The latter technique monitors the network packets and compares them with a database of signatures of malicious threats. This type of IDS has high accuracy rates in detecting known attacks. On the other hand, anomaly-based detection is very useful for new attacks as it depends on the behavior of normal operations or resources accessed by operations. In this case, network patterns should be well configured to avoid false negative alarm for harmful use or false positive alarm for legitimate use. Usually, the best practice of IDS is using a hybrid IDS which is based on signatures as well as network anomaly behavior.

In this work, we discuss the different access control models for PLCs, but we focus on the most commonly deployed access control mechanism, namely, password-based access control. Using recent PLC devices (2016) with updated firmware, we show how passwords are stored in PLC memory, how passwords can be intercepted in the network, how they can be cracked, etc. As a consequence of these vulnerabilities, we could carry out advanced attacks on ICS system setup, such as replay, PLC memory corruption, etc. In addition, we developed signature-based IDS to identify normal from abnormal operations in SCADA network.

1.2 Thesis Contribution

In what follows is the outline of the contribution of this research:

- We carry out a detailed study of the PLC access control problem.
- We propose a set of traffic analysis attacks on the communication between the ES and the PLC.
- We show how PLC access control authentication can be broken.
- We design and implement a set of PLC commands detection techniques based on network signatures.

1.3 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 lists and briefly describes related work. Chapter 3 is a summary of the most common PLC vulnerabilities as reported in the ICS-CERT repository. The different access control models and a security analysis of the PLC password access control model are described in Chapter 4. In the light of the security analysis, Chapter 5 shows the possible attacks that can be launched as a consequence. Chapter 6 presents detection techniques of different attacks we conducted in this research. Finally, in Chapter 7 a brief summary of the work and future works are discussed.

CHAPTER 2

ICS SECURITY RELATED WORK

This chapter highlights recent works in the area of ICS security particularly, attacks on PLC. In addition, it reviews research about SCADA network detection and prevention techniques.

2.1 PLC Security

Although many cybersecurity incidents targeted ICS systems in the past 20 years, it was only after the Stuxnet [2] malware hit Iranian nuclear facilities that ICS security turned into a high priority topic. Stuxnet incident was a wake up call for those still not realizing the severe consequences of ICS security breaches [3]. Adopting a holistic approach to ICS security analysis, Johnson showed that ICS systems are designed in levels and there are gaps in the existing security measurements used, in particular, in SCADA systems [4]. He proposed two attack vectors to exploit PLC vulnerabilities, namely, bypass logic attack which is based on compromising the PLC internal RAM and brute force output attack that results in arbitrary commands sent to the PLC. The described attacks did not assume a flaw in the PLC. Instead, a PLC can be attacked through its normal operation given a network access. Morris and Gao showed that attacks resulting in consequences similar to Stuxnet can be achieved with relatively simple scenarios [5]. They described 17 attacks on SCADA devices (RTUs, PLCs, etc.) exploiting vulnerabilities in Modbus protocol. Very close to our work, in a BlackHat talk, Beresford demonstrated a number of vulnerabilities in Siemens Simatic PCS7 software including replay attacks, authentication bypass, fingerprinting and remote exploitation using Metasploit framework [6]. This paper deviates from Beresford's demonstrations since our attacks are more interactive and use the recent and more secure versions of the PCS7 software as well as the more up to date firmware of Siemens PLC S7-400. As a generalization of Beresford's attacks, Milinkovic and Lazic reviewed a set of commercial Operating Systems running on PLCs of major vendors, highlighting serious vulnerabilities with some experiments of few attacks conducted on ControlLogix PLC [7].

As mentioned above, a major PLC attack vector consists in exploiting weaknesses in the communication protocols or their implementations, such as MODBUS, MODBUS/TCP, etc. Byres et al. investigated commonly used ICS protocols with the aim of finding defects that caused the devices to respond in inappropriate manners or to inject malicious packets in the normal traffic [8]. Closer to our work, Sandaruwan et al. showed how to attack Siemens S7 PLCs by exploiting flaws in the ISO-TSAP (Transport Service Access Point) protocol used for data exchange between controllers and PLCs [9].

Another family of PLC related attacks is the PLC Firmware modification attack. PLC firmware is the operating system of the PLC embedded device. PLC firmware needs to be installed and updated frequently. If the firmware update is not thoroughly verified and validated, attackers may craft malicious firmware and use it to compromise PLCs. Basnight et al. studied the firmware attack development process and demonstrated a successful attempt of uploading a modified firmware to an Allen-Bradley ControlLogix L61 PLC [10]. Costin et al. pushed the idea to a larger scale by carrying out a large-scale analysis of about 32,000 firmware images [11].

A significant body of work in the literature focuses on security solutions for ICS systems which yield several countermeasures to reinforce the security of such systems. These can be classified into communication protocols improvement [12, 13], and firewalls, filtering methods, DMZs [4, 7, 9]. However, unlike typical IT systems, it is impractical and cost-effective to embrace several layers of mitigations due to performance and availability considerations.

2.2 Intrusion Detection System

As mentioned above, SCADA systems have exposed to wide variety of attacks since Stuxnet, which may cause catastrophic consequences for the nation's economy and even worse to human lives. These attacks encouraged researchers to look for algorithms of defending in order to stop or mitigate these damages. Authors in [14] highlighted the challenges involved in securing control systems. However, the solutions currently in use or proposed in literature either are developed for certain SCADA system protocol, which are not applicable to others, or have some limitations in detecting certain attacks.

The basic idea of signature-based intrusion detection systems is trying to recognize predefined malicious patterns (virus) in network traffic flow and signal an alarm if it has taken place. This approach has key advantage: its false alarm rate is very low, if any. Thomas Morris et al. [15] derived a set of 50 signature-based IDS rules from vulnerability analysis of Modbus/TCP and Modbus over serial line systems. The rules are generic that can be used for multiple IDS platforms. In addition, they used Quickdraw Snort preprocessor [16] and other Snort rules to detect malicious activities on Modbus communication networks. Oman and Phillips [17] produced comprehensive signatures for unauthorized access to SCADA devices in electrical power grid which supporting several protocols, such as DNP3, Modbus and RTU/ASCII protocols. The IDS maintains SCADA devices details in XML profile such as IP address, Telnet port, etc. It uses Perl programming for parsing the XML profiles and created Snort rules for legal commands on the RTU. For evaluating the performance of the system, they developed a SCADA/sensor testbed.

However, the previous technique can be bypassed with a simple modification in the attack scheme and has limitation in detecting new attacks. These facts have motivated the research towards anomaly-based IDS for SCADA networks. This detection technique is feasible due to the thoughts that SCADA is well-behaved networks, means that it is predictable environment. Garitano et al. [18] provided a survey of the research effort focused on the development of anomaly detection for SCADA systems. Cardenas et al. [14] proposed algorithm for attack detection using mathematical framework. The idea of this approach is to understand the behavior of physical processes then analyze the network to detect any modification in control commands or sensor data. Authors in [19] designed model-based IDS for Modbus TCP/IP SCADA networks. Basically, it checks Modbus TCP requests and responses to detect any violation to the standards of the protocol, violation in the communication between components and detects service availability. The main drawbacks of this system that it detects commands with malicious code or response validity; it cannot detect attacks with valid commands. Also, it is specific only to Modbus TCP/IP networks.

Ramachandruni and Poornachandra [20] took advantage of honeypots to mimic the services of control systems (HMI, Modbus). It used to attract attackers and monitor network attack vectors. The goal of this research is to analyze attackers activities and build models of attacking methodologies. The authors recommend to utilize honeypot system which can help in identifying more attack vectors, report and share it to prevent future attacks on SCADA systems. Alternative technique in anomaly-based IDS is neural network algorithm. The performance of this method depend essentially on the training data set and the selected input features of the network traffic. Although research efforts in neural network are still at an early stage but such works [21] [22] have shown the effectiveness of this method as the most appropriate tool for anomaly detection.

Anomaly-based IDS has clear drawback; it may issue high false positive or false negative alerts if the model was not well configured. Thus, hybrid intrusion detection system get benefits of both, signature-based and anomaly-based technique [23] to improve IDS performance and enhance alarm accuracy. The cool advantage of this type is that it is capable in detecting known attacks as well as zero-day attacks. The works in [24] and [25] proposed ruled-based intrusion detection system for SCADA networks using IEC 60870-5-104 protocol. This IDS using signature-based rules to detect some abnormal events on this protocol. Also, it uses model-based techniques contains protocol fields analysis and communications analysis to identify unexpected behavior of the protocol. Valdes and Cheung [26] adopted a multi-layered model-based IDS which can be deployed in both, the network and host level. The IDS is combining model-based and complemented by signature-based approach and using Snort for detection. It was evaluated using Modbus and DNP3 over TCP/IP protocol.

2.3 Summary

This chapter highlighted a variety of related works in SCADA security area. Part of them investigated in the attacks against PLC, while others conducted their researches in finding a solutions and mitigation techniques to detect and prevent such attacks. Our work deviates from existing works by targeting initial communication between PLC and engineering station (e.g. PCS7) for configuration or re-programming the PLC. In addition, we contribute to the knowledge in the area of IDS signature-based to detect attacks executing against PLC using normal command operations.

CHAPTER 3

PLC VULNERABILITIES

A PLC is a particular type of industrial embedded devices that is programmed to manage, control and continuously monitor the state of physical components (motors, valves, sensors, etc.) based on system inputs and custom requirements. A PLC typically has three main components, namely, an embedded operating system, control system software, and analog and digital inputs/outputs. Hence, a PLC can be considered as a special digital computer executing specific instructions that collect data from input devices (e.g. sensors), sending commands to output devices (e.g. valves), and transmitting data to a central operations center.

PLCs are commonly found in supervisory control and data acquisition (SCADA) systems as field devices. Because they contain a programmable memory, PLCs allows a customizable control of physical components through a user-programmable interface. There are many languages used to program a PLC in a variety of ways. Ladder Logic (LAD) is the most commonly used, Statement List (STL), Function

Block Diagram (FBD), etc. It can also be programmed by high level languages such as C.

Nowadays, PLCs are used in just about every critical process in many different industries. If an attacker successfully compromises a PLC, he has the ability to shut down the PLC, overwrite PLC logic or values, perform code injection, to name a few. While some of these attacks may result in information disclosure, others can give a false system state, harm the PLC device physically, and in an extreme case, lead to human injury and loss.

The ICS-CERT repository, dedicated to ICS related security incidents, includes several reports involving PLC vulnerabilities and alerts. Most of the reports are relatively recent (2010 and later). The increase in ICS and PLC incidents coincides with the increasing interconnection of ICS and corporate networks which became a necessity to improve efficiency, minimize costs, and maximize profits. This, however, exposes ICS systems, and PLCs in particular, to various types of exploitation.

Most of PLC vulnerabilities can be grouped into three categories, namely, network vulnerabilities, firmware vulnerabilities, and access control vulnerabilities.

3.1 Network Vulnerabilities

The network architecture and protocols of an ICS system have a great impact on its vulnerability to external attacks. Previously, to guarantee ICS security, PLCs were deployed inside an 'air-gapped' network, isolated physically from other networks and facilities, plus, using proprietary protocols. However, recent incidents proved to be ineffective way of protection. Moreover, with the optimal productivity needs of modern critical infrastructures, PLCs are increasingly required to be interconnected with corporate LANs, Intranets, and Internet. As a result, PLCs are expected to support mainstream network protocols. Such standard protocols (e.g. TCP, IP, ARP, etc.) facilitate interconnection with multiple devices of different vendors, but bring their own vulnerabilities (e.g. Spoofing, Replay, MITM, etc.). Consequently, any ICS protocol based on the standard protocols will inherit their vulnerabilities.

Another source of PLC vulnerabilities is the set of additional features that typically come with PLCs, such as web servers, FTP servers, e-mail sending capabilities, etc. Each one of these features come with its own implementation vulnerabilities (e.g. server software flaws, web application flaws, etc.). Exploits for these flaws exist publicly and can be leveraged to compromise PLC devices in a variety of ways, such as updating PLC firmware, disrupting PLC services, etc.

However, the most common type of network vulnerabilities is related to

ICS specific network protocols such as Modbus, profinet, DNP3, etc. In particular, Modbus protocol, the de facto standard for connecting industrial electronic devices, has number of security weaknesses such as lack of authentication, lack of integrity checking of data sent over the protocol. Other protocols send credential data in clear-text format. These vulnerabilities may grant adversary an unauthorized access or unauthorized manipulations of values that affect SCADA operations. Secure DNP3 is a suitable alternative to be used in data transmission, however, not all software vendors provide support for this protocol. Table 3.1 lists a sample set of PLC network vulnerabilities as reported in ICS-CERT repository.

Table 3.1: Examples of PLC network vulnerabilities as reported in ICS-CERT advisories

Advisory	**Affected product**	**Vulnerability**	**Exploit**
ICSA-11-223-01A	Siemens SIMATIC PLCs	Use of Open Communication Protocol	Execute unauthorized commands
ICSA-15-246-02	Shneider Modicon PLC Web Server	Remote file inclusion	Remote file execution
ICSA-12-283-01	Siemens S7-1200 Web Application	Cross-site Scripting	Run malicious javascript on Engineering station browser
ICSA-15-274-01	Omron PLCs	Clear text transmission of sensitive information	Password sniffing
ICS-ALERT-15-224-02	Schneider Electric Modicon M340 PLC Station	Local file inclusion	Directory traversal/file manipulation

3.2 Firmware Vulnerabilities

Firmware is the operating system of controller devices, in particular, PLCs. It consists of data and code bundled together with several features such as OS kernel and file system. Firmware is typically written by device vendors. As any software, a firmware is prone to flaws and security vulnerabilities. Vulnerabilities include buffer overflow, improper input validation, flawed protocol implementation, etc. In addition, today, most of controller devices run by standard operating systems to leverage its features. However, using standard OS eliminates the concept of ”security by obscurity” and controllers will inherit vulnerabilities of used OS. Therefore, PLC vendors offer updates or patches regularly to fix functional bugs or security flaws which could be done remotely for devices scattered in dispersed locations.

There are several issues that need to be considered when patching or uploading a new firmware to a PLC. For example, patching a PLC should take into consideration the nature of ICS systems which require the availability of devices 24/7. More importantly, firmware and patches must be certified by vendors to make sure that they will not break system functionalities. Unfortunately, a large number of PLC vendors use weak firmware update validation mechanisms allowing unauthenticated firmware updates [11]. Uploading custom or malicious firmware could bring the system down, give access to running system and allow an adversary to have full control of the device. Consequently, arbitrary code can be run, malicious services can be installed, backdoors can be planted, denial of Service (DoS) attack can be launched, privilege can be escalated, and so on. Table 3.2 lists a sample set of PLC firmware vulnerabilities as reported in ICS-CERT repository.

Table 3.2: Examples of PLC firmware vulnerabilities as reported in ICS-CERT advisories

Advisory	**Affected product**	**Vulnerability**	**Exploit**
ICSA-16-026-02	Rockwell MicroLogix 1100 PLC	Stack-based buffer overflow	Remote execution of arbitrary code
ICSA-13-116-01	Galil RIO-47100 PLC	Improper input validation (allowing repeated requests to be sent in a single session)	Denial of Service
ICSA-14-086-01	Shneider Modbus Serial Driver	Stack-based buffer overflow	Arbitrary code execution with user privilege
ICSA-12-271-02	Optimalog Optima PLC	Improper handling of incomplete packets	Denial of Service
ICSA-16-152-01	Moxa UC 7408-LX-Plus Device	Non-recoverable firmware overwrite	Permanently harming the device

3.3 Access Control Vulnerabilities

A PLC is a sensitive component of ICS systems and hence only authorized entities should be allowed to access it and any such access should be appropriately authenticated. The first layer should be physical access control preventing physical compromise of the PLC. At the network layer, ICS networks must be protected by employing DMZs, firewalls, and intrusion detection and prevention systems to prevent unauthorized intrusion.

The most common PLC access control vulnerabilities include poor authentication mechanism, lack of integrity methods, flawed password protection, and flawed communication protocols. For example, PLC vendors use hidden or hard coded usernames and passwords to fully control the device. Attackers setup a database of default usernames and passwords and can brute-force such devices. Once unauthorized access is performed, an adversary can retrieve sensitive data, modify values, manipulate memory, gain privilege, change PLC logic, etc. All these issues are discussed in details in the following Chapter. Table 3.3 lists a sample set of PLC firmware vulnerabilities as reported in ICS-CERT repository.

Table 3.3: Examples of PLC Access Control vulnerabilities as reported in ICS-CERT advisories

Advisory	**Affected product**	**Vulnerability**	**Exploit**
ICSA-16-224-01	Rockwell MicroLogix 1400 PLC	Use of insecure SNMP functions for firmware update	Unauthorized change to device configuration
ICSA-13-011-03	Rockwell ControlLogix PLC	Improper access control	Unauthorized commands (reset, stop, etc.)
ICSA-15-274-01	Omron CJ2M and CJ2H PLCs	Password stored in recoverable format	Password recovery (local exploitation)
ICS-ALERT-12-020-05A	Koyo ECOM100 Ethernet Module for PLCs	Weak authentication (8-byte passcode)	Easily crackable passcode

CHAPTER 4

PLC ACCESS CONTROL

This chapter highlights PLC access control from three different views. It starts with physical deployment of PLC and restrictions that must be followed to keep PLC in a safe place. The second section discusses techniques of protection for a PLC from external attacks through the network. It followed by password based access control of PLC and requirements need to be employed in developing password mechanism. The last section introduces a security analysis of password-based access control, that we conducted on Siemens PLC, namely, Siemens S7-400

4.1 Physical Access Control

Proper deployment and access control of PLC as well as other ICS controllers mitigate significantly security breaches either from internal or external adversaries. Access control vulnerabilities can be significantly reduced by implementing recommendations in established standards such as the ANSI/ISA-99 [27]. It is a complete security life-cycle program that define procedures for developing and deploying policy and technical solutions to implement secure ICS systems. ISA99 is based on two main concepts, namely, zones and conduits, whose goal is to separate various subsystems and components. Devices that share common security requirements have to be in the same logical or physical group and the communication between them take place through conduits. This way, network traffic confidentiality and integrity is protected, DoS attacks are prevented and malware traffic is filtered. Thus, if any infection takes place, it affects only limited components in a particular zone. In addition, control system administration must restrict physical and logical access to ICS devices to only those authorized individuals expected to be in direct contact with system equipment.

4.2 Network Access Control

ICS network access control is typically implemented in layers. The first layer is network logical segmentation achieved typically with security technologies such as firewalls and VPNs. All controller devices, in particular PLCs, must be located behind firewalls and not connected directly to corporate or other networks. Most importantly, critical devices should not be exposed directly to Internet. Remote access to all ICS devices should be through secure tunnels such as VPNs. Indeed, many vendors provide special appliances for securing ICS networks. For example, Siemens provides a special type of switch, namely, Scalance S, with firewall and VPN features to secure the communication from/to PLCs. Notice that even with full deployment, these technologies may not block all breaches due to weak or inadequate configurations and filtering rules.

It is important to note that security countermeasures of SCADA systems are different from mainstream ones used in traditional IT systems in the means of operational priorities and risk. In SCADA system, actions and responses must be in real-time manner. In contrast to IT system, it cannot tolerate delays in receiving or sending data which may lead to sever consequences. In addition, anti-virus and encryption techniques are very time and resources consuming. Moreover, SCADA systems must provide 24/7 monitoring and control processes, thus patching or replacement of applications or controllers is hard to implement even if they are ruing with known vulnerabilities. There are a lot more challenges and differences between The two systems. SCADA Engineers, physical and Cyber security professionals and system architects need to join together and create well-established network architecture to limit or prevent unauthorized access to the SCADA sensitive assets.

4.3 Password Access Control

Password based access control is by far the most commonly used type of access control and most important as well. The common mistake between PLC vendors is that they are using built-in password protection (e.g. "Admin, Admin", "Admin, 1234", etc) to prevent unauthorized access and tampering attacks. One of the reasons for using default password is that in some cases they need to respond quickly to an alarm or critical event. If they used sophisticated password, it might be difficult to remember or type. Attackers are aware of this vulnerability, so they are usually prepare a set of default pair of user-name and password to decrease the time of password cracking. For effective password access control, important requirements need to be satisfied. In particular, password protection:

- must be enabled whenever possible
- must be properly configured
- must use strong encoding scheme
- must not need high processing operations
- must not use hardcoded credentials
- must be frequently and periodically changed.

In addition, it is highly recommended to delete default accounts or change default passwords. Unfortunately, not all vendors comply with and enforce these principles, therefore several password related incidents are reported (Table 3.3).

4.4 Security Analysis of PLC Password Access Control

As its name indicates, a PLC is programmable, that is, by loading a new program, the PLC can be reconfigured to function in a different way. To do so, major

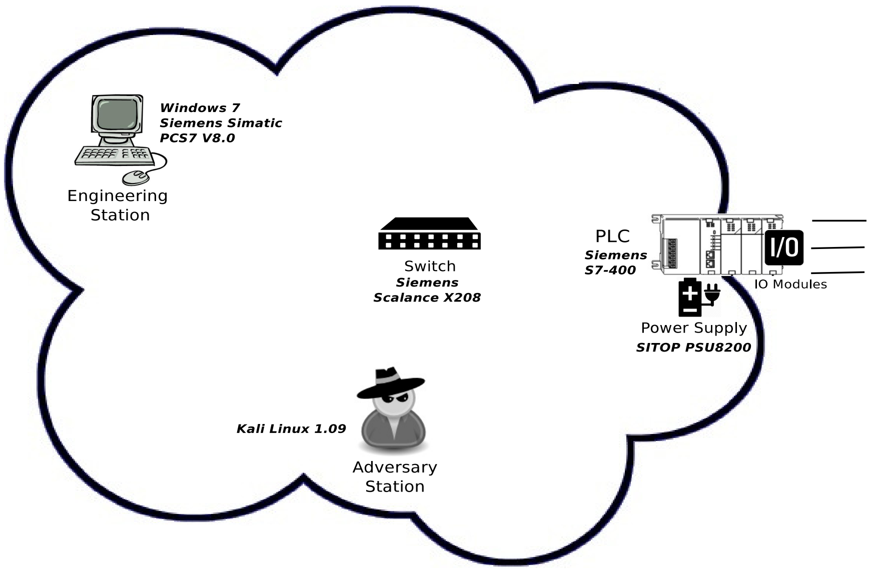

Figure 4.1: PLC Lab Setup

manufacturers of PLC (e.g. Siemens, Modicon Schneider Electric, Allen-Bradley, etc.) provide integrated software platform to program, debug or configure a PLC as well as other controllers connected to PLC. Typically, a new program is loaded by connecting the PLC with an engineering station (equipped with a control system software) via direct wire or through a LAN after it has been written and tested at engineering station. In addition to reprogramming the PLC, the engineering station can send control commands to the PLC such as *start, stop, check status*, etc. Therefore, an attacker who can interfere with the PLC access control mechanism can cause a lot of damage to the whole ICS system.

To carry out a realistic security analysis of PLC access control, we selected a commonly used PLC model, namely, Siemens S7-400, with its software namely, Simatic PCS7, and setup a lab including common ICS configuration (Figure 4.1).

Simatic PCS7 is control system platform used to configure and program controller devices of Siemens such as PLC. It is equipped with a range of engineering tools and wide variety of functions designed in graphical user interface and uses COTP (Connection Oriented Transport Protocol) in the communications between PLCs and Engineering stations. COTP is an obsolete, unsecured protocol built on the top of TCP but it is still using by Siemens PCS7 software.

Based on S7-400 documentation, several test cases have been performed which revealed three access control levels for the PLC, namely, no protection, write-protection and read/write-protection. The first level of access control, which is the default level, does not provide any form of access control. Using this level, any entity (device, station, etc.) can access the PLC processes and data without restriction. Access is possible provided that the remote entity "speaks" a PLC supported communication protocol (e.g. COTP, Modbus, Profinet). The second level, write-protection, provides as its name indicates a write protection on PLC data and processes. That is, any attempt to modify data or processes on the PLC (e.g. Load new program, clear data) is password authenticated. The third level, which is the most restrictive, is read/write-protection. Using that level, any interaction, that is, read from or write to the PLC is password authenticated. Through our analysis of password mechanism implemented in Simatic PCS7, we found that it doesn't fulfill the requirements mentioned in Chapter 4.3

4.4.1 Password Policy

The configuration software, namely, SIMATIC PCS7 accepts any 8 ASCII characters password to protect PLC form unauthorized access and manipulation of programs and configurations. If the password is less than 8 characters long, PCS7 pads it with white spaces. To set a PLC password, a user has to change the protection level and set the password in the PCS7 hardware configuration tool before loading the changes to the PLC.

In addition to being loaded to the PLC memory, the password is stored locally in the engineering station's local files. In a normal scenario any command sent to the PLC (e.g. start, stop, clear memory) should be authorized by providing the password. However, since the password is stored locally in the engineering station, PCS7 software will ask for the password only one time after the new configuration is loaded to the PLC. In subsequent interactions, PCS7 will include automatically the password in the packet requests sent to the PLC. This behaviour can be changed any time through hardware configuration tool. Bear in mind, that if PLC was protected, it doesn't execute any command unless password is supplied.

SIMATIC PCS7 using password to control the access to projects stored locally in engineering station. Since, this is not in our scope, we ignored it through our investigation.

4.4.2 PLC Memory Structure

As mentioned above, setting a password consists in changing the protection level, selecting a password and then loading the new configuration to the PLC memory. The latter is organized into labeled blocks. Each block holds a specific type of information (Figure 4.2). Therefore, information loaded to the PLC is divided into blocks as well. Mainly, there are two categories of blocks according to what it holds; blocks that are hold user program and blocks that hold configuration data.

Most of PLC blocks are used to organized the PLC program into independent sections corresponding to individual tasks. Function Block (FB) is a block that holds user-defined functions with memory to store associated data. Functions (FC) is used to keep frequently used routines in the PLC operations. Data Block (DB) stores user data. Organization Block (OB) is an interface between operating system and user program, used to determine the CPU behavior, for example, define error handling. System Function Block (SFB) and System Functions (SFC) hold low level functions (libraries) that can be called by user programs such as handling the clock and run-time meters.

The password is communicated and stored in the System Data Block (SDB). SDB itself is divided into sub-blocks with different roles. The sub-blocks numbered from 0000 to 0999 and from 2000 to 2002 hold data that is updated in each

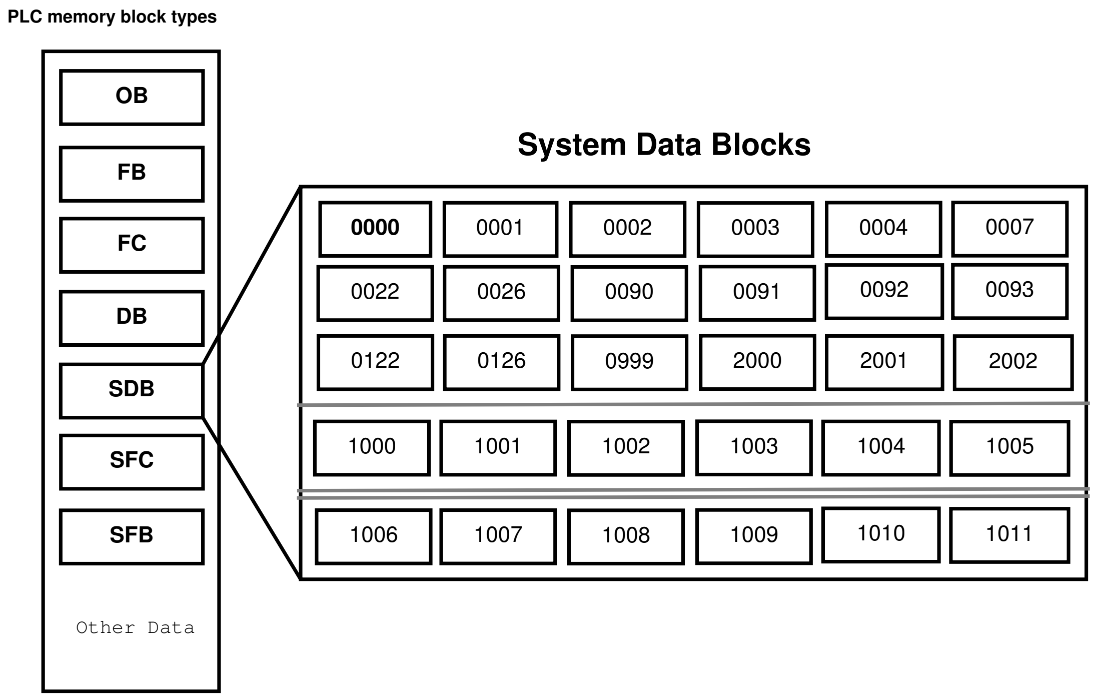

Figure 4.2: S7-400 PLC memory structure

download process. The rest of the sub-blocks are divided into two sets: sub-blocks from 1000 to 1005 which contains some sort of configuration data used in the first downloading time. In the second time, the configuration is stored in the sub-blocks from 1006 to 1011 . Thus, in the next time, the first sub-blocks used to hold the configuration data, and so on.

Loading a new configuration to the PLC yields to delete data of all sub-blocks of the SDB block, then load new configuration, except the 0000 sub-block which contains the password. This sub-block is overwritten with a new password. If an adversary aims at updating the password, he needs to clear the 0000 block first with a dedicated command and then set a new password with another command, which is not the case in normal PCS7 process.

4.4.3 PLC Password Sniffing

Typically, the common strategy for retrieving sensitive information (e.g. password) from data that is flowing across the network is "Packet Sniffing" (Figure 4.3).

Methodology

Packet sniffing where packets are intercepted and captured at the Ethernet frame level. After capturing, the data can be filtered and examined using manual and automated techniques to review granular-level details within network traffic. It helps also in identifying traffic patterns and vulnerabilities of a system to penetrate into it or commit other attacks such as replay attack. The capturing process can be managed either by installing a script in the targeted machine [28] or monitoring the communication between nodes using dedicated program (e.g. Wireshark). Next, the recorded packets are filtered based on specific criteria, e.g. protocol or IP address.

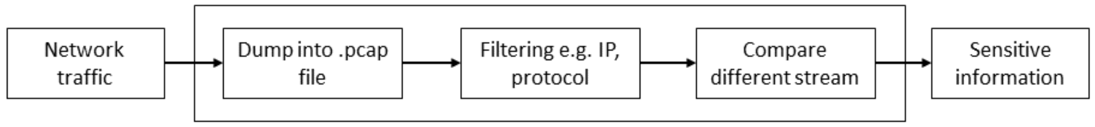

Figure 4.3: Sensitive Information Extraction Steps

Most of standard protocols' documentations are publicly available which makes it less difficult to find the location of targeted information. However, for proprietary protocols, the filtered data can be analyzed by comparing different streams with each other to locate the similarity and differentiation between them. For example, if password is the target, then dump number of streams that have the same configurations except the password, compare them using different comparison tools and try to figure out password location. Note that before you can effectively identify the information, you need some knowledge about that fields structure, such as, the policy, length, etc.

Implementation

In order to evaluate the security of the password-based access control, a first step is to sniff the network packets containing the password. Typical network sniffing software is used to capture packets exchanged between the engineering station (PCS7) and the PLC during a password setting process (e.g. Wireshark, tcpdump). Since password setting is achieved through load configuration command sent to the PLC, the process is repeated several times with different passwords to collect a good number of samples. The captured traffic is first filtered to extract complete TCP streams of the packets which were sent to PLC only. Packets received from PLC were ignored. The streams are then compared using byte comparison tools (e.g. Burp Suite Comparator). These tools help finding similarities and differences between TCP streams. This allowed to identify the specific packets containing the password and the exact bytes shift for the password location inside the packets. It turned out that the 8 characters password is encoded in each packet. Hence configuration software in the engineering station uses an encoding scheme to encode the password before uploading it to the PLC.

It is important to note that when the PLC is configured with no-protection level, sniffed packets during load configuration have the same size as with the other levels of protection (read protection and read/write protection). Hence, the packet is padded with random bits (....*.6ES7) which is part of PLC module number.

4.4.4 Reverse Engineering Password Encoding Scheme

After locating the 8 bytes inside the network packets containing the password, the next step is to decode the bytes to retrieve the plain-text version of the password.

Methodology

The best practice to break a cipher text starts with well-developed strategy. A random selection of a tool is a time wasting. There are number of methods that can be used to crack the password. The Dictionary attack, Hybrid Dictionary attack, Rainbow table or Brute force attack. The difficulty of cracking depends on the available information about the password; it falls into four categories, from hard to easy:

- Cipher-text only where the attacker is presumed access only a set of cipher-text.
- Known plain-text where the attacker has access to both the plain-text, and its corresponding encrypted version, cipher-text to reveal the secret key.

- Chosen plain-text which assumes that the attacker can get the cipher-texts for random plain-texts. He needs to get information that cut down the security of the encryption mechanism.

- Chosen cipher-text where the crypt-analyst can gather information by obtaining the decryption of chosen cipher-texts.

In some cases, some encoding algorithm would be reasonable and easy to recognize, e.g. Base64, which we excluded in this experiment. So, in our case , we have the plain-text and cipher-text too and a kind of hybrid dictionary table consisting of plain-text and its corresponding cipher-text. Before using a specific decoding algorithm, we did try the possibility of using full-fledged cryptographic, hashing functions, typical encoding schemes such as URL encoding, etc.

Implementation

Full-fledged cryptographic (DES, AES, RC4, etc.) as well as hashing (MD5, SHA512, etc.) functions are excluded in the investigation because of four reasons:

1. There is no key exchange stage involved before password communication, This holds for cryptographic functions.

2. If cryptographic and hashing functions were used, the encoded password bytes would be completely shuffled compared to the plain text version, which is not the case here (the cipher text is encoded byte by byte).

3. Cryptographic and hashing functions are too processing intensive for PLCs. PLC is controlling very critical processes, that means any delay in delivering data may cause troubles. Moreover, PLC main function is to facilitate industry operations. So, it is not practical to use such functions.

4. Size of cipher text in AES, MD5, SHA512, etc. is fixed and it is more than the size of encoded password found in the traffic.

The reverse-engineering started by trying typical encoding schemes, namely, URL encoding, ASCII Hex, variants of Xor (single-byte, multiple-byte, rolling, etc.) using Burp Suite tool in Kali Linux. However, none of these typical schemes retrieved the plain text version of the password, pre-set in our samples.

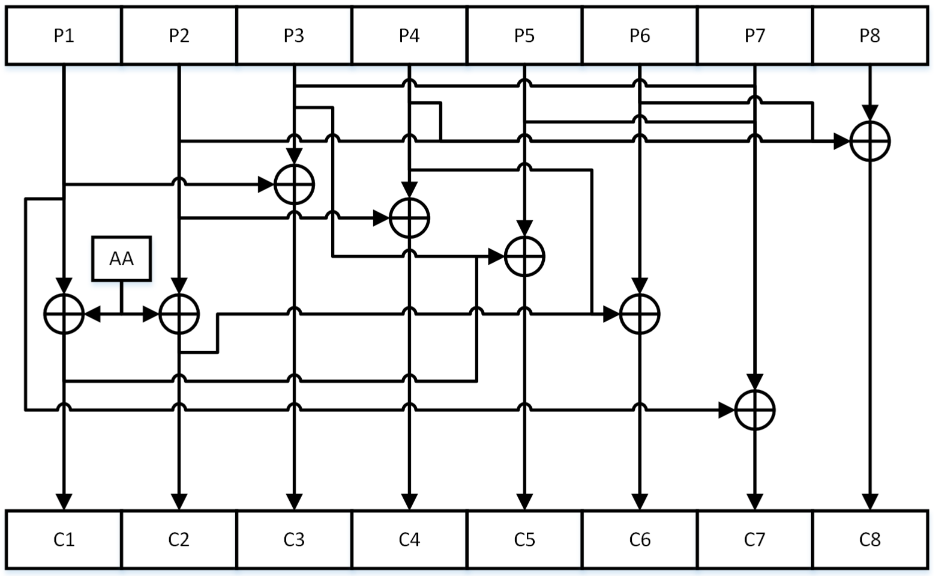

Figure 4.4: PLC Password Encoding

Xor is a very common encoding scheme that is suitable for resource limited hardware devices. As mentioned above, the password encoding is not using typical Xor (single-byte, multiple-byte, etc.). Taking into consideration the fact that the encoding is done byte-by-byte and the requirement of a lightweight encoding algorithm, we focused on trying customized Xor transformations. To this end, a representative list of (plain-text password, encoded-text password) pairs have been sampled from the network (Table 4.1). Then, using automated scripts to brute-force each byte, we could successfully reverse-engineer the Xor based encoding scheme. A graphical representation of the nested Xor based encoding scheme is shown in Figure 4.4.

Table 4.1: Sample of Passwords and Corresponding Encoded Passwords

Password (ASCII)	**Encoded Password (Hex)**
a a a a a a a a	cb cb 00 00 cb cb 00 00
b b b b b b b b	c8 c8 00 00 c8 c8 00 00
a a b b c c d d	cb cb 03 03 ca ca 04 04
b b a a d d c c	c8 c8 03 03 cd cd 04 04

First of all, the first character in the password (P_1) is xored with "0xAA", so with second character. The third encoded character (C_3) is produced by xoring (P_1) with (P_3) and fourth character in encoded password (C_4) is produced by xoring (P_2) and (P_4). The fifth encoded character (C_5) is the result of xoring (P_1), (P_3), (P_5) and "0xAA". (C_6) is the result of xoring (P_2), (P_4), (P_6) and "0xAA". The last two encoded character are produced as follows: (P_1) xored with (P_3), (P_5) and (P_7) to produced (C_7), and (P_2) xored with (P_4), (P_6) and (P_8) to produce (C_8).

It is important to note that the PLC is using two variants of the encoding scheme: one used to load a configuration to the PLC and the other is used during the authentication process. Both variants differ by the static byte constant used: “0x55” and “0xAA”.

CHAPTER 5

PLC ACCESS CONTROL ATTACKS

This chapter presents a security analysis of the communication between PLC, namely, Siemens S7-400, and engineering stations equipped with SIMATIC PCS7. We conducted bunch of attacks, namely, replay normal commands attack, Man in the middle (MiTM) attack , clear PLC memory, update CPU password, etc. In the following sections, we explain how to launch each attack and what tools are needed.

5.1 Replay Attack

As a consequence of compromising the password based PLC access control, mentioned in chapter 4.4, several concrete attacks can be carried out on the PLC, ranging from simple replay to unauthorized password update attacks. A replay attack on the PLC consists of recording a sequence of packets related to a certain legitimate command and then replaying it later without authorization. The attack consists of 3 steps: starting a given command (stop, start, load configuration, clear memory block, etc.), capturing the packets, and replaying the captured packets at a later time, see figure 5.1. The captured packets of a command need to be processed before replay it by discarding all packets received from the PLC then store it in a pcap file. The target PLC may or may not be password protected.

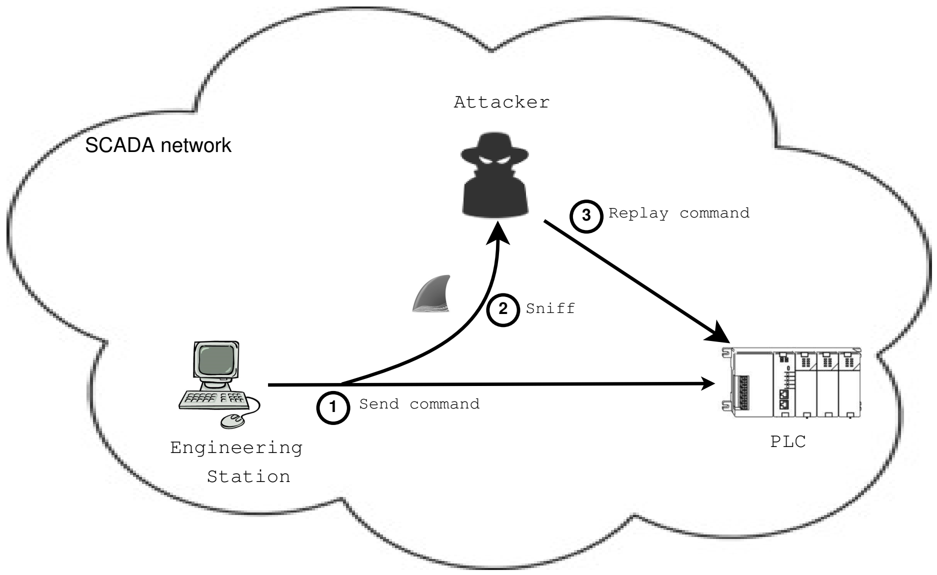

Figure 5.1: Attacks general approach

There are mainly three challenges that need to be addressed in order to replay packets successfully. First, packet headers (IP address and TCP port) need to be rewritten and checksums need to be recomputed. The modified IP address and TCP port correspond to attacker machine. Second, sequence and acknowledgement numbers need to be changed, otherwise the PLC will tag them as duplicates or out of order and will discard them. Third, packets need to be sent in the right order and timing, in particular, some packets should be sent after receiving some packets in the other direction. These problems have been recently observed by Maynard et al. [29].

There are several useful tools used in replay attack, for example tcpreplay. This tool can be used to change IP address, Tcp port, recompute checksum, etc. However, the attack was failed because of TCP/IP kernel at the PLC discarded most of the packets due to two reasons: First, the sequence and acknowledgement numbers were not changed by the tool before replayed it. So, PLC considered it as a duplicate or out or order. Second, the tool didn't wait for a reply from the PLC and it sent the packets one after another. To overcome these obstacles and to guarantee that the replayed packets are accepted by the TCP/IP kernel at the PLC, we resorted to write a customized python script using scapy [30]. Scapy is a powerful packet manipulation program written in python and hence can be easily used in python scripts. It features a variety of packet manipulation capabilities including: sniffing and replaying packets in the network, network scanning, tracerouting, etc. However, the most useful scapy features for our replay attack are the ability to rewrite the sequence and acknowledgement numbers and to match requests and replies.

Updating the sequence and acknowledgement numbers consists of recalculating these numbers and rewriting them with scapy. Manipulating packet headers using scapy is straightforward since any packet field is simply accessible by the dot operator (e.g. *ip.src*, *tcp.flags*, *rcv[TCP].seq*). Initially, random sequence and acknowledgement numbers are chosen. Then, at each packet sending, the numbers are incremented and added to the next packet. Scapy offers variety of Send functions, which is used to send a packet to the other end of the communication. Some packets do not require an acknowledgement, so *sendp()* can be used. It takes as an input the packet and the network interface. On the other hand, packets which require a response, Scapy offers several functions. This function *srp1()* is the most suitable one for our attack, as it sends and receives packets in layer 2 and returns only the first answer from which we can extract the sequence number.

Algorithm 1 shows the core of the python script using the scapy features. The *REPLAY* subroutine takes as input the pcap file, the network interface, the attacker's IP address and port number. In addition, arbitrary values are chosen to initialize the ACK and RSTACK numbers. The *for* loop inside the subroutine goes through the packets one by one. For each packet, IP and TCP checksums are removed (lines 7) so that the network interface card recalculates newer values, the source IP and port numbers are updated (lines 8 and 9), the sequence numbers are incremented (lines 11), the packet is replayed using either *scapy sendp* function (for SYN and RST packets) or the *scapy srp1* function (lines 13 and 18).

Algorithm 1 Replay a sequence of captured packets using Scapy

```
 1: function REPLAY(pcapfile, eth_interface, srcIP, srcPort)
 2:     recvSeqNum ← 0
 3:     SYN ← True
 4:     for packet in rdpcap(pcapfile) do
 5:         ip ← packet[IP]
 6:         tcp ← packet[TCP]
 7:         del ip.chksum                          ▷ Clearing the checksums
 8:         ip.src ← srcIP      ▷ Specify the attacker machine IP and PORT
 9:         ip.sport ← srcPort
10:         if tcp.flags == ACK or tcp.flags == RSTACK then
11:             tcp.ack ← recvSeqNum+1
12:             if SYN or tcp.flags == RSTACK then
13:                 sendp(packet, iface=eth_interface)
14:                 SYN ← False
15:                 continue
16:             end if
17:         end if
18:         rcv ← srp1(packet, iface=eth_interface)
19:         recvSeqNum ← rcv[TCP].seq
20:     end for
21: end function
```

First of all, the above python program has been tested using two attack scenarios. In the first scenario, the replay attack was launched from the same host (IP address) used in the capture process, that is, the engineering station with the configuration software. In the second scenario, the replay attack was launched from a different host on the same network, that is, the attacker machine with Kali. In each scenario, two types of commands are tried, namely, start and stop which require password authentication. The replay attack was successful in both scenarios for both types of commands. Hence, an unknown attacker machine (without appropriate configuration software) on the same network, can turn the PLC ON or OFF by simply replaying a start or stop command without knowing the PLC password. This clearly might cause significant damage to a SCADA system. We extended the attack and use this program to replay other commands. The attacks are detailed in the following sections

5.2 Man-In-The-Middle (MiTM) Attack

Basically, some of PLC networking facilities are based on TCP/IP model. In this model, Address Resolution Protocol (ARP) is used to link network address (e.g. IPv4 address) to physical address (MAC). When a sender host needs another host's MAC address residing in the same LAN, it broadcasts an ARP request to all hosts in the network. ARP request contains the IP address of a host which the sender needs its MAC address. So, in a typical scenario, only the owner of that IP address responses to that request with an ARP reply which contains its MAC address. Since ARP is a stateless protocol, network host accepts any ARP reply in despite of whether it requested it or not. In addition, it updated and overwrote the ARP entries whenever it receives new APR reply packet even if the existence entries have not yet expired. Thus, it is most likely prone to MiTM attack implementing what it is known as ARP Poisoning.

ARP Poisoning attack is a technique by which an attacker floods a falsified ARP messages over a LAN. As a result, an attacker associates his MAC address with the IP address of a legitimate host in the network. Thus an attacker implant himself between two hosts and all the traffic tunneled through

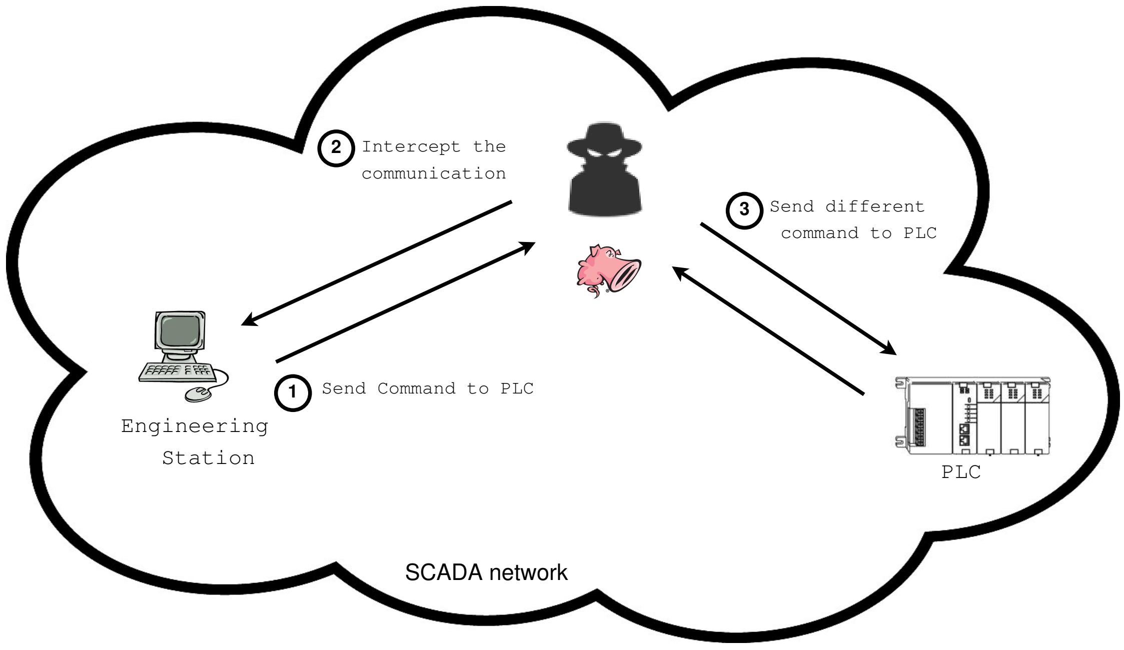
2 Intercept the communication
3 Send different command to PLC
Engineering Station
1 Send Command to PLC
PLC
SCADA network

Figure 5.2: Man-in-The-Middle Attack

his machine. There are two types of MiTM attacks; passive and active attack. Passive attack simply watching the communicated data between two hosts and hence breaking the confidentiality and stealing sensitive data. A more dangerous type is the active attack, where attacker can be able to intercept, modify and even stop the traffic causing Denial of Service.

In our attack, we implemented ARP Poising attack between engineering station equipped with PCS7 and the PLC using Ettercap tool [31]. The attack was successfully and all the traffic between PC7 and PLC was passing through attacker machine using Kali. By utilizing this attack with the previous one, replay, we can conduct several attacks and tamper with the packet's contents and commands sent by PCS 7 to the PLC. The following sections illustrate the attacks in details.

5.3 Unauthorized Password Setting and Updating

As detailed in Chapter 4, packets between the engineering station and the PLC are sent in clear format including the encoded passwords. Based on a representative set of samples, we could locate the password inside packets and reverse-engineer the password encoding scheme. This allowed us to retrieve the plain-text password from the network traffic between the engineering station and the PLC.

In a legitimate scenario, the PLC password is set and updated from the configuration software, in particular, from Hardware configuration component, in the engineering station. Typically, in case of password update, the old password should be supplied first. Due to the PLC access control vulnerability, an attacker can set and update the password by replaying malicious packets directly to the PLC.

When a password is written on the PLC, the SDB (System Data Block) is overwritten. The load process first checks the SDB to see if it's clean or has a configuration already. If there is a configuration, the process checks if a password is set or not. Hence, there are two main cases: setting a configuration with a password for the first time, and updating an old configuration that has already a password.

Case 1: PLC Memory is Cleared

For the first case, setting a password for the first time requires to record a password setting packets sequence used in an old session and then replaying them. Since the goal is mainly to set the password, only packets in charge of overwriting block [0000] in the SDB, which contain the password, are kept (More details in Chapter 4.4.2). All other packets are not needed in the unauthorized password setting attack. Hence, to set a new password without going through the configuration software, the password should be encoded using the cracked algorithm (Chapter 4.4.4) and then injected in the appropriate packet at the appropriate bytes shift.

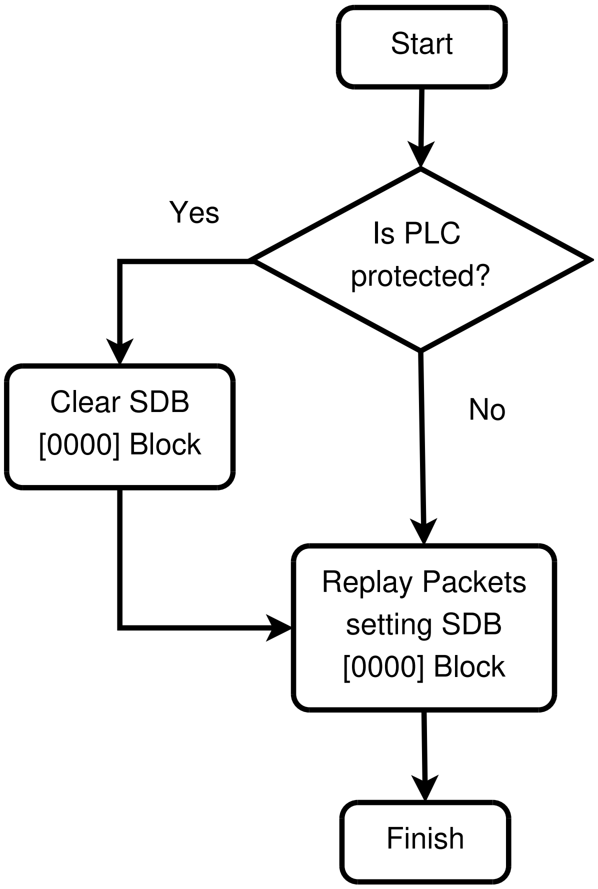

Figure 5.3: Unauthorized Password Setting Attack

Case 2: PLC memory Is Protected

For the second case, the goal of the attack is to set a password while the PLC is already protected by an existing password. Using the same procedure as the first case as-is did not work. After investigation it turned out that the sub-block [0000] of the SDB holding the password cannot be overwritten by replaying packets. As a result, the PLC keeps sending a FIN packet whenever an attempt is made to overwrite the SDB sub-block [0000]. To overcome this problem, we resorted to a two-stage procedure where initially we clear the content of [0000] sub-block and then we replayed packets to overwrite only that block with a new password. Since there is no legal command to just clean [0000] sub-block, we looked for a sequence of packets to delete a different block and we modified them to delete [0000] sub-block. With this two-stage procedure, the password is successfully updated by a different workstation without the configuration software and without knowing the old password.

5.4 Clear PLC Memory

The first stage of the unauthorized password updating attack consists in clearing the [0000] sub-block of the SDB without a need for the password. This step can be generalized to clear other blocks. More importantly, in an extreme use case, all PLC memory blocks can be cleared. With this vulnerability, an attacker can launch a DoS attack by clearing all PLC memory data and turning the PLC into unresponsive device.

5.5 Unauthorized Change of PLC User Program

User program is set of functions and data required to control specific automation task. The program comprises number of blocks correspond to individual tasks in controlling a machine. There are several types of blocks used in SIMATIC PCS 7 software; Logic blocks and Data blocks. Logic blocks contains sections of the program and variables structured in Organization Block (OB), System function blocks (SFB), system functions (SFC), Function blocks (FB) and Functions (FC). These blocks programmed by different programming languages such as Ladder Logic (LAD), Function Block Diagram (FBD), or Statement List (STL). Data blocks contains values of a machine or plant to control.

In this attack, OB and FB are programmed with set of codes and declared some variables. These blocks then downloaded to the PLC and sniffed the network packets of this command. The aim was to change a value of "preset speed" variable in FB and replay the command directly to CPU. So, this process was repeated several times to have enough samples for analysing and extracting the location of that value in the packets stream. At the end, the location was successfully specified and we prepared two pcap files for the attack, one is containing a password using in case a PLC is protected and the other one without a password. The attack was successful in both cases and we could easily replayed the packet and changed the value of the "preset speed" variable.

5.6 Summary and Mitigation

In typical IT networks, the basic protection measure against such network attacks (replay, password stealing, etc.) is to use encrypted communications. This assumes that both ends of the communication can encrypt and decrypt messages. While the engineering station and the configuration software can support encrypted communications, most of PLCs do not have cryptographic capabilities. The alternative to have cryptographic capabilities implemented within the PLC in addition to standard control features, is to place the PLC behind a network security device (e.g. firewall, secure router, etc.). An example of such devices is the Scalance S security module for PLCs which comes with VPN and IPSec features. If configured correctly, all the communications from/to the PLC will go through the security module. All the communications between the security module and any other device should be encrypted. However, the communication between the PLC and the security module is in clear. Therefore, the PLC should be connected physically to the security device only. All the other devices in the ICS network (engineering station, input/output devices, RTUs, etc.) should communicate with the PLC through the security module.

Although several security advisories in the ICS-CERT recommend using such security modules as mitigating solutions, it is not still widely adopted because of budget and practical (the need to reconfigure the ICS system, the wiring, etc.) considerations. The 6 years old Stuxnet attack is a manifestation of this fact. Indeed, PLCs got compromised because they were directly connected to infected engineering stations.

Another mitigating approach would be to use a network intrusion detection system in the LAN and detect malicious attacks (replay, PLC memory block clear command, password updated, etc.). Since communication is sent in clear between the engineering station and the PLC, generating network signatures for the described attacks is not a difficult task. However, this will allow only attack detection.

CHAPTER 6

PLC ATTACKS NETWORK DETECTION

6.1 Preliminary

The general purpose of signature-based detection is to identify and alert the occurrence of specific event, odd traffic, unusual header characteristics, an attempt of attack, to name a few. it ranges from very simple signature where it looks for a header value, to very complex one that a state of a connection is tracked or deep protocol analysis is performed. Simple signature may generate more false positives, on the other hand, complex or very specific signature may generate more false negatives as the characteristics of an event may change over a time. The key advantages of signature-based systems are the low rate of false positives, the effectiveness and accuracy in detecting the known security events. It maintains a database of signatures or attributes of known events (e.g. attack) and compare them against the monitored packets on the network.

The design of a signature depends on how it is implemented in real world not only on protocol standards as many implementations violate protocol standards. Basically, the signature should represent the unique characteristics of an event which can be identified by collecting large number of samples and analyze them by checking for specific features such as: Packet header values, Unique set of characters in packet payload either by using regular expression or specific words, Number of packets exchanged, Size of packets and so on. Finally, evaluate the signatures in terms of detection rate and processing performance.

6.2 Introduction

The communication traffic between PLC and engineering station is not encrypted, as discussed in Chapter 5. Thus, some clear-text words that indicate some useful information about the PLC and the sent commands can be easily noticed in the network traffic. For example, in each command executed by PCS7, the software first identify the model of the PLC it negotiates with. The model is located in the PLC memory and sent during the communication traffic in a normal text format. In case the PLC was protected, all commands must be authorized to be executed. The password is provided and can be noticed in the traffic although it is encoded, (see Chapter 4.4). We conducted deep analysis of these traffic for several operations using wide variety of tools contained in Kali Linux OS.

In this chapter, we introduce the first layer of defense against attacks targeting PLC. The implemented security measurement is Intrusion Detection System (IDS). It inspects the traffic between PLC and Engineering station equipped with PCS7. The IDS looks for several commands sent to PLC, such as start, stop, etc. In addition, it can be used to detect some attacks launched from internal or external network targeting PLC (Figure 6.1).

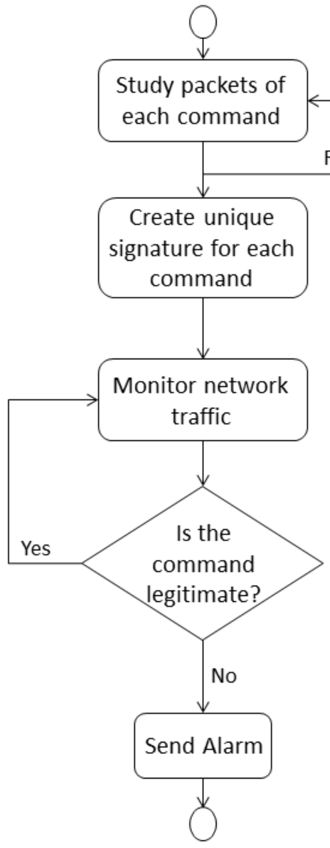
Study packets of each command
Repeat
Create unique signature for each command
Monitor network traffic
Is the command legitimate?
Yes
No
Send Alarm

Figure 6.1: General Detection Steps

The goal is to develop signatures for each command that can differentiated between them while monitoring the network traffic and notify the system administrator whenever an action or instance of an attack takes place. Signature-based approach is quite effective for detecting known attacks, as it usually captures the distinguished properties of the attacks it is designed to detect. Then, the signatures are evaluated using Snort IDS placed into PLC network along with Engineering station and attacker machine hosting Kali OS connected altogether by a switch (Figure 6.2).

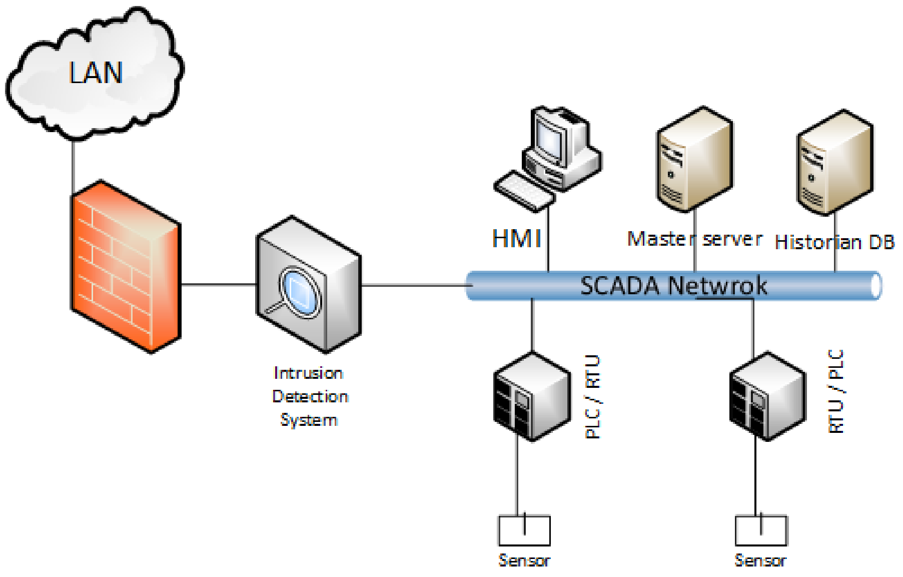

Figure 6.2: Intrusion Detection System in SCADA network

6.3 Detecting Replayed Packets

Replay attack is the main attack that is implemented in our experiments. A tool was developed and used to launch various attacks against PLC. There are many ways to implement such attacks. Normally, the prevention and detection techniques will differ. In this work, it was a straight replay of captured packets correspond to a command and replayed at a later time (Chapter 5).

The detection process is implemented as follows: PLC must only receive commands from a specific IP address, which is for Engineering station. So, any command sent to PLC from other machine is considered as an attack and the system will raise an alarm to SCADA control center. In addition, packet's content-based signature is created to recognize the commands sent to PLC. It is known that replay attack takes longer execution time than normal operations. Thus, this has been investigated and we found differences between them as expected. All these properties together constitute a detection formula for each command and can distinguish normal traffic from a malicious one. Some commands are not usually being used in normal situations. Detecting such commands outside a "Scheduled time" is considered a malicious action.

The implemented attacks using replay technique are as follows: Replay PLC start command, Replay PLC stop command, Replay password updating command, Replay PLC memory clear command and Replay user program blocks.

6.4 Start/Stop PLC Replay Attack Detection

First detection criteria is that these commands must be only issued from dedicated engineering station machine with fixed IP address. Stuxnet attack exploited this point where engineering station was installed in a laptop and that laptop was compromised by a malicious software installed from a USB. Second, we figured out a unique packet's content for both, start/stop PLC commands. Start operation can be done in many ways in PCS7, namely, warm start, cold start and normal start. All these variations have a common packet content that can be used to construct a signature along with the IP address of the engineering station. In addition, start/stop PLC are sensitive operations, i.g. PLC can run for long time controlling sensors and actuators and only be stopped in rare cases. Thus, detecting such commands in unscheduled time is not a normal operation and must be notified. The following snippet illustrated start/stop Snort detection rules.

```
############################################
# Rules for Start/Stop PLC Detection
############################################
alert tcp $Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"28 00 00 00 00 00 00 fd 00 00 09 50 5f 50 52 4f 47 52 41 4d—";
 rawbytes; msg: "Normal Start command sent to PLC from Engineering Station";
 sid: 1000001;)

alert tcp !$Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"28 00 00 00 00 00 00 fd 00 00 09 50 5f 50 52 4f 47 52 41 4d—";—
 rawbytes; msg: "malicious user send Start command to PLC";
 sid: 1000002;)

alert tcp $Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"00 29 00 00 00 00 00 09 50 5F 50 52 4F 47 52 41 4D—";—
 rawbytes; msg: "Normal Stop command sent to PLC from Engineering Station";
 sid: 1000003;)

alert tcp !$Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"00 29 00 00 00 00 00 09 50 5F 50 52 4F 47 52 41 4D—";—
 rawbytes; msg: "malicious user send Stop command to PLC";
 sid: 1000004;)
```

To trigger Snort rules, the packet should match the rule header as well as rule options. Normally, rule header contains the protocol (e.g. TCP), source IP address, source port number, destination IP address and destination port number. In our detection rules, `PLC` is a variable that correspond to PLC IP address and `PLC_PORT` corresponds to typical PLC port 102 that is used in all communications with PCS7 while `Eng_Stn` corresponds to IP address of engineering station that equipped with SIMATIC PCS7. The above rules are aimed to detect any occurrence of Start/Stop commands either from legitimate user or attacker machine. Typically, Snort alerts are sent to SCADA control center and it will react according to the severity of the received alarm.

6.5 Detecting MiTM Attacks

This attack is implemented mainly by spoofing an IP address of an entity in a network and hijack the communication channel. In our scenario, attacker represent himself as the other end of communication, e.g. for a PLC, the attacker is the engineering station. On the other side, the attacker is the PLC. All the traffic goes through attacker machine. However, MiTM by itself is not an easy attack, if the type of ARP table entry is static, ARP spoofing is not possible. In our detection phase, Snort has capability to detect ARP poisoning attacks by modifying the Snort configuration file snort.conf, specifically, remove the "#" from this line: `preprocessor arpspoof` and define the IP and MAC addresses of the devices in the network, namely, PLC, engineering station and IDS. This is illustrated in the snippet below. In addition, the commands execution time will differ accordingly. The request first sent to attacker machine, he might alter something, then send it to the intended destination. Time execution measurement will be as follows: Whenever Snort detect SYN request, it starts a timer and ends it when detecting RST flag. During this connection stream, if any command is detects, IDS will compare it with a previously defined time execution of such command. If there is any difference, that means an attacker sent that command and IDS alerts system control center.

```
############################################
# Detecting MiTM Attack
############################################
preprocessor arpspoof
preprocessor arpspoof_detect_host: $Eng_Stn_IP $Eng_Stn_Mac
preprocessor arpspoof_detect_host: $PLC_IP $PLC_Mac
preprocessor arpspoof_detect_host: $IDS_IP $IDS_Mac
```

6.6 Detecting PLC Memory Clear Command

This operation is very crucial to SCADA system as it clears everything in the memory, configuration information, control programs, etc. An engineer may request this command in very rare cases, e.g. during test phases, upgrading the system, to name a few. Thus, observing this command in the traffic raises doubts for possible attack that can convert PLC to unresponsive device and cause Denial of Service (DoS).

The detection system will check the sender IP address to verify that this command is originated by a legitimate user machine. In addition, it will check the time of sending this command. The system can verify whether this command is "clear memory" or others by looking for a specific contents in the packets. In our implementation, we developed Snort rules to detect the occurrence of request to clear memory. The following snippet pertain to inspect the traffic and alert for authorized and unauthorized attempts to clear PLC memory.

```
############################################
# Rules for Clear Memory Detection
############################################
alert tcp $Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"00 01 12 08 12 41 0c 00 00 00 00 00—”; rawbytes;—
 msg: "Normal user send clear memory command to PLC"; sid: 1000005)

alert tcp any any -> $PLC_IP $PLC_PORT
 (content:"00 01 12 08 12 41 0c 00 00 00 00 00—”; rawbytes;—
 msg: "malicious user send clear memory command to PLC"; sid: 1000006)
```

6.7 Detecting Unauthorized Password Update Command

The password is updated through Hardware configuration tool, it does not have specific command but updated during configuration download process. As mentioned in Chapter 4.4.2, the password is located in block [0000] of type SDB. In our attacks, there are two main cases: setting PLC password for the first time where PLC memory is empty of data, and updating an old configuration that has already a password (Chapter 5.3). Although in both cases we used the same command, however, the attacks behaviour and the detection rules are both different.

Since the memory is empty in first attack case, no deletion or overwritten is required for the password block. Thus, the packets which are in charge of setting the password are simply replayed. The IDS has two signature for this operation. First it looks for the request of engineering station to download this sub-block. Second it looks for PCS7 Program Invocation (PI) service _INSE which is a request to activate the block. Once these contents observed in the traffic, it means that a block [0000] is downloaded. These signatures are the same for both, attacks and normal request. The difference will be in the source IP address since the IP address of engineering station is known to the IDS.

The second attack case is using a malformed command which is delete sub-block [0000] first then update the password. In normal download process, this block is never deleted, it is overwritten. However, in the attack scenario, that didn't work. So, we modified a delete command for a random block, e.g. [2001] by changing the number of block to [0000]. The IDS will inspect the traffic for this command, which is only used in our attack. One last case is failed password attempts. A customized signature is created to detect failed authentications. The following rules illustrate all signatures for detecting password updating process.

6.8 Evaluation and Summary

In this work, we conducted deep security analysis of the communication between PLC and engineering station. We created signatures for several commands for detection purposes. A key advantages of this detection method is:

- Signatures are easy to develop and efficient if you know what network behavior you are trying to identify.
- The amount of power needed to perform these checks is minimal for a confined rule set. This is crucial for SCADA systems in which data transmission

```
############################################
# Rules for Password Update Detection
############################################
alert $Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"5f 30 42 30 30 30 30 30 50 0d 31—"; rawbytes;—
 msg: "Request sent to PLC to download sub-block [0000] from Engineering
 station"; sid: 1000007)

alert !$Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"5f 30 42 30 30 30 30 30 50 0d 31—"; rawbytes;—
 msg: "Request sent to PLC to download sub-block [0000] from malicious
 user"; sid: 1000008)

alert $Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"30 42 30 30 30 30 30 50 05 5f 49 4e 53 45—"; rawbytes;—
 msg: "Request sent to PLC to activate sub-block [0000] from Engineering
 station"; sid: 1000009)

alert !$Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"30 42 30 30 30 30 30 50 05 5f 49 4e 53 45—"; rawbytes;—
 msg: "Request sent to PLC to activate sub-block [0000] from malicious
 user"; sid: 1000010)

alert !$Eng_Stn_IP any -> $PLC_IP $PLC_PORT
 (content:"01 00 30 42 30 30 30 30 30 41 05 5f 44 45 4c 45—"; rawbytes;—
 msg: "Attacker deleted sub-block[0000]"; sid: 1000011)
```

delay is not affordable.

- No false-positive or false-negative is generated since it looks for a specific well-selected packet content which is transmitted in plainly manner.

Table 6.1 illustrates the developed detection signatures in each work. However, the weakness of this approach is that it can not detect zero-day attack since its signature is not yet developed. Also, if Siemens updated the commands, these signatures might not work. This is the nature of this approach, it has to be updated as long as new update is released.

Table 6.1: Comparing our work with others

Attacks Detection	**Our Work**	**Oman & Phillips**	**Morris et al.**
Replay Attack	✓		
CPU Start / Stop	✓		
MiTM	✓		
Password update	✓		
clear memory	✓	✓	
Unauthorized access	✓	✓	
Protocol deviation			✓

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The vulnerabilities, attacks and detection methods reported in this thesis have been tested and validated on a very common PLC, namely S7-400, with an updated firmware. However, based on the advisories and alerts in ICS-CERT repository, several other PLCs are vulnerable to similar attacks. The attacks are possible because of the common practice of connecting the PLC directly to the engineering station through a typical LAN. With that setting, any station or device in the same LAN can carry out all the attacks described above. Securing a site with an IDS alone is not effective, integrating an IDS into a network that has well thought out security capabilities can greatly enhance the security of a network.

As a general security measure, controlling network access to vital SCADA components is strongly recommended with proper mechanisms in order to run the system in a protected IT environment. Organizations have to perform impact analysis, risk assessment and take defensive measures to mitigate the consequences of exploitation of vulnerabilities. In detail, They have to:

- Ensure that ICS components or systems are not accessible from the Internet and minimize the exposure to the network.
- Isolate filed networks and devices from business network using e.g. Firewall.
- Use secure mechanism for remote access, e.g. VPNs. It is important to note that VPN itself might has vulnerabilities and must use the most current version.

7.2 Future Work

As mentioned in Chapter 3, PLC is suffering from several vulnerabilities. In this work we focused on communication flaws exist in Siemens PLC and PCS7 software. This work can be extended to analyze the communication between PLC and HMI, PLC and sensors, etc. The best detection solution for our focus area is signature-based IDS what we implemented. However, this may not be the best for other solution as it has a lack in detection zero-day attacks.

Another area is firmware analysis which goes through analyzing the strength and capability of validation algorithm implemented in PLC to detect any modification in the firmware. Firmware is the OS of the PLC, if an attacker compromised it, he can change the behaviour of PLC, implant backdoors, etc.

REFERENCES

[1] O. Andreeva, S. Gordeychik, G. Gritsai, O. Kochetova, E. Potseluevskaya, S. I. Sidorov, and A. A. Timorin, "Industrial control systems vulnerabilities statistics," 2016.

[2] N. Falliere, L. O. Murchu, and E. Chien, "W32.stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[3] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pp. 4490–4494, 2011.

[4] R. E. Johnson, "Survey of scada security challenges and potential attack vectors," *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pp. 1–5, 2010.

[5] T. H. Morris and W. Gao, "Industrial control system cyber attacks," *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013*, pp. 22–29, 2013.

[6] D. Beresford, "Exploiting siemens simatic s7 plcs," *Black Hat USA*, 2011.

[7] S. A. Milinković and L. R. Lazić, "Industrial plc security issues," *Telecommunications Forum (TELFOR)*, pp. 1536–1539, 2012.

[8] E. J. Byres, D. Hoffman, and N. Kube, "On shaky ground–a study of security vulnerabilities in control protocols," *Proc. 5th American Nuclear Society Int. Mtg. on Nuclear Plant Instrumentation, Controls, and HMI Technology*, 2006.

[9] G. Sandaruwan, P. Ranaweera, and V. A. Oleshchuk, "Plc security and critical infrastructure protection," *2013 IEEE 8th International Conference on Industrial and Information Systems*, pp. 81–85, 2013.

[10] Z. Basnight, J. Butts, J. Lopez, and T. Dube, "Firmware modification attacks on programmable logic controllers," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 76–84, 2013.

[11] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A large-scale analysis of the security of embedded firmwares," *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 95–110, 2014.

[12] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, "Dnpsec: Distributed network protocol version 3 (dnp3) security framework," in *Advances in Computer, Information, and Systems Sciences, and Engineering.* Springer, 2007, pp. 227–234.

[13] J. Heo, C. S. Hong, S. H. Ju, Y. H. Lim, B. S. Lee, and D. H. Hyun, "A security mechanism for automation control in plc-based networks," *2007 IEEE*

International Symposium on Power Line Communications and Its Applications, pp. 466–470, 2007.

[14] A. Cardenas, S. Amin, and S. Sastry, “Research challenges for the security of control systems,” *HOTSEC’08 Proceedings of the 3rd conference on Hot topics in security*, 2008.

[15] T. H. Morris, B. A. Jones, R. B. Vaughn, and Y. S. Dandass, “Deterministic intrusion detection rules for modbus protocols.”

[16] D. Peterson, “Quickdraw: Generating security log events for legacy scada and control system devices,” *Conference for Homeland Security*, 2009.

[17] P. Oman and M. Phillips, “Intrusion detection and event monitoring in scada networks,” *Critical Infrastructure Protection*, 2007.

[18] I. Garitano, R. Uribeetxeberria, and U. Zurutuza, “A review of scada anomaly detection systems,” *Soft Computing Models in Industrial and Environmental Applications*, 2011.

[19] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, “Using model-based intrusion detection for scada networks,” *Proceedings of The SCADA Security Scientific Symposium*, 2007.

[20] R. Ramachandruni and P. Poornachandran, “Detecting the network attack vectors on scada systems,” *Advances in Computing, Communications and Informatics*, 2015.

[21] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," *Proceedings of International Joint Conference on Neural Networks*, 2009.

[22] W. Gao, T. Morris, B. Reaves, and D. Richey, "On scada control system command and response injection and intrusion detection," *eCrime Researchers Summit*, 2010.

[23] W. Gao and T. Morris, "On cyber attacks and signature based intrusion detection for modbus based industrial control systems," *Journal of Digital Forensics, Security and Law*, 2014.

[24] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, and W. H.F., "Rule-based intrusion detection system for scada networks," *Renewable Power Generation Conference*, 2013.

[25] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and W. H., "Intrusion detection system for iec 60870-5-104 based scada networks," *Power and Energy Society General Meeting*, 2013.

[26] A. Valdes and S. Cheung, "Intrusion monitoring in process control systems," *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.

[27] E. Byres, "Revealing network threats, fears: How to use ansi/isa-99 standards to improve control system security," 2011.

[28] A. Amro, S. Almuhammadi, and S. Zhioua, “Netinfominer: High-level information extraction from network traffic,” *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pp. 143–150, 2017.

[29] P. Maynard, K. McLaughlin, and B. Haberler, “Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks,” in *Proceedings of the 2Nd International Symposium on ICS & SCADA Cyber Security Research 2014*, ser. ICS-CSR 2014. UK: BCS, 2014, pp. 30–42.

[30] P. Biondi, “Scapy,” *http://www. secdev. org/projects/scapy*, 2011.

[31] ALor and NaGA, “Ettercap,” http://ettercap.sourceforge.net.

Vitae

- Name: Haroon Abdul-Aleem Wardak
- Nationality: Afghanistani
- Born in September 13$^{\text{th}}$, 1988, Makkah
- Email: *hawardak@gmail.com*
- Permanent Address: Jeddah, KSA
- Bachelor of Science (BSc) degree in Information Technology (Software Engineering), King Abdul-Aziz University, Jeddah, Saudi Arabia, First Honor, August 2011.
- **Research Interest** Cyber Security, ICS Security, Digital Forensics.
- **Job Experience:**
  - Quality Assurance in Futures Business Development company (Oct 2011-July 2012)
  - Developer in Al-shatiby institution (Nov 2012 Sep 2013)
- **Publication**
  - Haroon Wardak, Sami Zhioua, and Ahmad Almulhem. "PLC access control: a security analysis." Industrial Control Systems Security (WCICSS), 2016 World Congress on. IEEE, 2016.