PLC Memory Attack Detection and Response in a Clean Water Supply System

Andres Robles-Durazno[a,*], Naghmeh Moradpoor[a], James McWhinnie[b], Gordon Russell[a] and Inaki Maneru-Marin[b]

[a]*School of Computing, Edinburgh Napier University, Edinburgh.*

[b]*School of Engineering & Built Environment, Edinburgh Napier University, Edinburgh.*

Abstract— ***Industrial Control Systems (ICS) are frequently used in manufacturing and critical infrastructures like water treatment, chemical plants, and transportation schemes. Citizens tend to take modern-day conveniences such as trains, planes or tap water for granted without considering the critical systems involved for their operations. Interrupting these industries could lead to disastrous consequences, leading to financial losses or even costing human lives. For that reason, researchers have been actively investigating the threats targeting ICS. In this paper, the authors propose a mechanism of attack detection and mitigation for attacks focusing on the input memory of Programming Logic Controllers (PLCs). To help investigate this concept, a testbed that models a clean water supply system was built using components and technologies currently used in the industry. The mechanism supporting attack detection and response for the input memory is implemented within the PLC itself as part of its programming. The mechanism of response involves three different techniques: optimised datablocks, switching between control strategies and obtaining the sensor readings directly from its analogue channel. The results demonstrate the feasibility of the proposed approach along with the effectiveness of each response mechanism.***

Keywords — Industrial Control System, Clean Water Supply System, SCADA, Testbed, Attack Detection, Attack Response, Programming Logic Controller, Memory Attacks.

1. Introduction

Industrial Control Systems (ICS) have been used across many businesses including critical infrastructures such as water supply, telecommunications and transportations. These industries are essential for the functioning of a society and economy. For instance, in December 2015 an attack targeting a power plant in the Ukraine caused a massive power outage lasting hours and affecting approximate 230,000 people. An email with an infected attachment was the origin of the attack, highlighting critical infrastructure vulnerabilities. Several attacks hitting US nuclear facilities and UK energy sectors have been also reported in 2017 and 2018 showing the attackers' growing interest in these industries[1].

ICS consist of sensors for measurements, actuators for affecting change (e.g. valves, motors and heaters) and logic resolvers or controllers in addition to related hardware and software. These control systems are designed to achieve high levels of control and efficiencies that will enhance the final product or service[2].

Over the years, controllers have evolved from mechanical through electrical/electronic to microprocessor-based systems. However, the control techniques used for the majority of process control systems have not changed significantly, given that control algorithms mimicking early physical controllers [3]. The most widely used closed-loop control method for Single Input Single Output (SISO) systems is the Proportional Integral Derivative (PID) controller. The PID controller can also be used as the basis of more advanced Multi Input Multi Output (MIMO) controllers, for example ratio, cascade, and feedforward control systems [4].

A Programmable Logic Controller (PLC) is a computer used for industrial automation and process control, and it can be used to automate a specific process, machine function, or even an entire production line. The PLC is a commonly used component in the Supervisory Control and Data Acquisition (SCADA) system responsible for

* Corresponding author.
E-mail addresses: a.roblesdurazno@napier.ac.uk (A. Robles-Durazno), n.moradpoor@napier.ac.uk (N. Moradpoor), j.mcwhinnie@napier.ac.uk (J. McWhinnie), g.russell@napier.ac.uk (G. Russell), inakimaneru@gmail.com (I. Maneru-Marin)

monitoring, collecting and processing input and output data from field devices such as motors, valves, sensors, and pumps[5].

In the PLC market, the main end-user segment includes industries such as automotive, chemical and petrol-chemical, mining and metallurgy, paper, packaging and printing, food and beverages, water and waste-water treatment as well as oil, gas and nuclear power plants[6]. Many manufacturers have registered the PLC as their trademarks. This includes worldwide leading automation vendors such as Siemens, ABB, Emerson, Schneider (Modicon), Rockwell (Allen- Bradley), Mitsubishi, Fortive (Danaher), Yokogawa, GE, Honeywell and Omron[7]. The major PLC sellers in the global market are Schneider, Rockwell, Siemens, Mitsubishi, and Omron[6]. Based on online resources[8], a total of 80% of all PLCs sold have been by sold the world's top seven vendors. Siemens has the greatest share of the market, with a contribution of as much as 30.7%, followed by Rockwell, Mitsubishi, and Schneider with a contribution of 21.6%, 13.9% and 8.9% respectively. Moeller with the market share contribution of 2.3% and is the last company on the list. The PLC market by geography is categorised into North America, which occupies the largest market share in 2016, then Asia Pacific, Europe, Latin America, the Middle East and Africa. Additionally, the US and Canada are the largest revenue contributors in the PLC global market [6]. Fig. 1 reveals the top eleven leading automation vendors worldwide in 2016 based on revenue in U.S. dollars[9].

Although there are many types of PLC, it is important to choose one for this research which reflected the capabilities of the market as a whole. Therefore, after comprehensive research, we chose the latest Siemens S7-1500 [10] PLC product, also known as SIMATIC S7-1500, which features include a fast backplane bus, PROFINET interface, and short reaction times. The S7-1500 PLC has a command processing time of less than 1 ns and is probably the fastest controller worldwide.

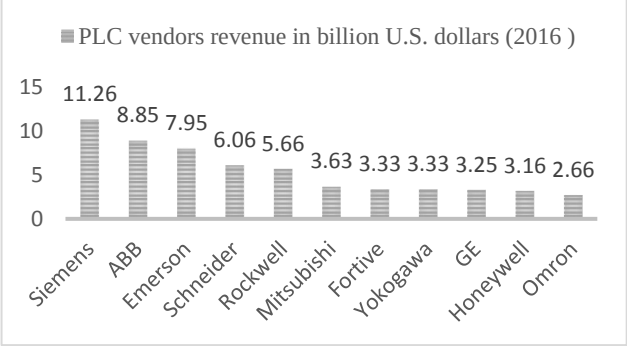

Fig. 1. PLC vendors revenue in billion U.S. dollars (2016)

In a given PLC, four sections exist which are common to all PLCs. This includes: 1) the power supply section, which can include battery backup and provides DC power to the PLC and the I/O modules, 2) the user program section, which carries the PLC software program that defines the controller actions, 3) the CPU module, which holds the processor(s) and the memory chip(s), and 4) the I/O section, which includes the input and output interface modules that read and control the peripheral devices[5].

Given that in this paper the focus is on attacks, targeted on the PLC memory, which is a part of the CPU module as explained above, it is important to understand the general PLC memory specifications before considering on PLC memory area descriptions for the Siemens S7 family. PLC memory stores digital information electronically, which is retrievable and regularly scanned by the processor. PLC memory has two modes of "write" and "read" and contains several types of data, including data tables, image registers, and the software program.

In the S7 family of Siemens PLC (e.g. S7-1500), the CPU provides a variety of dedicated memory areas. This includes process image inputs (I), process image outputs (Q), bit memory (M), datablock (DB), and local or temporary memory (L). Each different memory location has a unique address which is used by the user program to access the information in the memory location (i.e. to read from or write to)[11].

The process image inputs or "I" memory is where the CPU copies the state of the physical inputs to at the beginning of each scan cycle. The process image inputs or "Q" memory is where the CPU copies the state from

to the physical outputs at the end of each scan cycle. The bit memory or "M" memory is where the user program reads the data from and writes the data into. The "M" memory can be accessed by any code blocks. Datablock or "DB" memory, which can be specified to be either read/write or read-only, is used for storing various types of data. This includes the intermediate status of an operation, other control information parameters, timers or counters. The local/temporary memory or "L" memory is allocated by the CPU to a code block when needed for its execution, and when the execution finishes the CPU reallocates it to other code blocks.

1.1 Research Questions

The experiments described in this paper aim to tackle a cyber-security issue which can be found in control systems regarding PLC vulnerability from memory attacks. The following research questions are identified, which are aimed to be responded through the conducted experiments:

RQ1: How do the PLC memory attacks affect its control process operation?

RQ2: Is it possible to minimise the impact of cyber-attacks on the control systems using control methods?

RQ3: What countermeasures could be taken into consideration to continue the control system's current operation when a cyber-attack is detected?

1.2 Contribution

In this paper, a novel technique is presented for attack detection and response for the input memory of the PLC. This technique is coded inside the PLC and it does not require an additional module and/or equipment. Results are provided which were obtained from a physical testbed using modern control equipment.

1.3 Organization of the Paper

This paper is organized as follows. In Section 2, we review the related work in the field. In Section 3, we describe the testbed design and implementation followed by our proposed attack detection and response. In Section 4, we present the testbed scenarios. In Section 5, we describe the technique for attack detection and response. In Section 6, we discuss the results captured from our conducted experiments. In section 7, we discuss the findings and finally in Section 8, we present the conclusion for the paper along with future work follows by the acknowledgement and the references.

2. Related Work

Several researchers have studied Cyber Attacks on water systems (e.g. water treatment systems and clean water supply systems) in the past. In this section, a number of existing works related to attack vectors as well as attack detection and response in water treatment systems are discussed, with a particular focus on those targeting the sensors/actuators on Secure Water Treatment (SWaT) system [12]. The SWaT is an operational but scaled down water treatment testbed which includes a six-stage filtering process and is capable of producing five gallons of twice filtered water per minute. Almost all of the published work in this area uses the SWaT testbed. The reason for this literature focus is because we have also used a similar testbed named Festo MPS PA Compact Workstation Rig [13] in addition to our attacks, which are based on injecting wrong sensor/actuator values targeting the PLC memory vulnerabilities. The electronic literature search was conducted on Google Scholar and IEEE Xplore using query phrases such as: "water treatment systems", "attack model", "attack detection" and "attack response" with a 2010 to 2018 publication year filter applied.

In [14], the authors studied the behaviour and response of a Cyber Physical System (CPS) by implementing jamming attacks on the SWaT system using Software Defined Radio (SDR) exposing vulnerabilities associated with the design of the system. In their experiments, the attacker's aim was to block Level 0 and Level 1 network communication channels. While the former, which is also referred to as the field-bus network [15], is a communication path between each PLC and a set of sensors and actuators, the latter provides a communication route between the PLCs in a SWaT system. Addressing their results, jamming Level 0 caused the compromised sensors to no longer send data without the operator being alerted via the SCADA/HMI screen. However, jamming Level 1 resulted in disconnecting the unit's wireless link from the SCADA, HMI and the SWaT Server, thus making the operator notice that there is a problem in communication due to the significant impact on the event.

In [16], the authors proposed sensor noise fingerprinting to detect attacks on physical components of a CPS with a focus on ultrasonic level sensors on the SWaT testbed. To calculate the noise fingerprint for a sensor, first they collected its data from the healthy runs which contained no attack. The noise is then extracted from the collected data and averaged to obtain the sensor's fingerprint. Once the noise fingerprinting for all the sensors are collected, fresh data will be obtained by running the SWaT plant. At the end the noise vector of the fresh data is extracted and correlated with the respective sensor's noise fingerprints to detect anomalies. For the experiments, they implemented two attacks: sensor swap attack, where the attacker swaps level sensors between two tanks of the SWaT systems, and sensor replace attack, where the attacker brings his own sensors and replaces them with the existing ones. Addressing the results, their proposed sensor noise fingerprinting was successful in detecting anomalies on the testbed.

In [17], the researchers proposed an approach in which the behaviour of the first three stages of the SWaT plant along with its sensors and actuators is captured in approximate, discrete models, and their interaction is analysed to discover potential attacks that involve a number of compromised sensors and actuators. For this, they first extracted a model of the system from the code and provided the attack specifications. Using these two elements, they then employed an Alloy analyser to automatically generate an attack scenario describing how the system can be compromised and ended up in an unsafe state. The attack planner is then used to simulate the impact of the generated attack on the system. They then performed the validation sequence on the SWaT testbed to confirm whether the attack is feasible or invalid. This process continues until the analyser fails to detect any further attacks on the system. Their results showed that their proposed model-based approach is successful in automatically discovering and exploring attacks on the water treatment system.

In [18], the authors proposed a Design to Invariants (D2I) approach to derive state-based invariants programmed into a PLC to detect cyber-attacks on ICS with a focus on a fully operational 6-stage SWaT testbed. They first used an extended hybrid automata to model the system's process dynamics from which the invariants are derived. SWaT components that have discrete time and continuous time behaviour such as actuators and those whose physical states are being measured by sensors are included in the creation of invariants. Each invariant is programmed and then inserted into the associated PLC as a guard for the control code. The invariants are active during the 6-stage SWaT operation to check the system state validity with regard to the system design and to further detect anomalies. For the evaluation, they considered two type of attacks: Single Stage Single Point (SSSP) and Single Stage Multi Point (SSMP), and while the former includes a single sensor located at a single stage, the latter comprises multiple sensors /actuators but at a single stage. Addressing their results, the D2I approach was successful in detecting all SSME attacks, however for the SSSP attacks it was not effective at detecting those attacks launched while the PLC is being reset after power failure. Additionally, given that the invariant violation does not necessarily imply an anomaly as it may also occur due to the component failure, it is rather unclear how they distinguish sensor failure from anomalies.

In [19], authors presented an anomaly-based IDS for attack detection in critical infrastructures. Their proposed IDS operates at the industrial control process level and performs detections in a real-time. Their implemented IDS works in two phases. In the phase one, the IDS learns the normal behaviour of the control process. In the phase two, which is also known as detection phase, their proposed IDS raises an alarm every time an abnormal behaviour is found in the system. The core of the IDS is based on two algorithms: the latest version of Negative Selection Algorithm and the Artificial Immune System. The authors validated their proposed approach using different network traffic datasets including the dataset provided by the SWaT testbed. The results showed that their proposed IDS achieved an accuracy of 85% considering nominal attack and attacks with no labels. Although this approach operates at the industrial control process level, it still needs to analyse the network traffic which adds extra overhead to the process.

In [12], authors introduced three basic attack models for the SWaT testbed and conducted some initial experiments to assess the security vulnerabilities of the system. This includes system reconnaissance using open source tools such as Wireshark and Zenmap to determine the industrial protocol vulnerabilities (e.g. ENIP) as well as services running on local devices such as PLC and HMI, in addition to ARP spoofing attacks using Ettercap which resulted in re-directing local traffic through the hacker's device. They were also successful in acting as a Man-In-The-Middle (MITM) between two parties (i.e. two PLCs) to capture sensor data and actuator commands and re-write them on-the-fly by using Ettercap rules, in addition to manipulating remote firmware and logic updates from the SCADA to each individual PLC. Moreover, they discussed compromises through wireless

networks (e.g. by impersonating the legitimate Access Point to trick the PLCs) and through direct physical access (e.g. by re-wiring networking cables and inserting passive taps). They also discussed the system's response to the attack. However, their work is rather basic, general and unclear.

In [20], the authors presented a network security analysis of the communication between the PLC and the Engineering Station, where the Engineering Station is in charge of PLC set up and configuration. For this, they implemented three common computer network attacks: Reply, MITM, and Command Modification, to compromise the communication between the PLC and the Engineering Station. For the experiments, they used Siemens S7 - 400 with Simatic PCS7 8.1 software along with open source tools and python scripts. Addressing their captured results, they have shown that the programming and configuration traffic between the Engineering Station and the PLC can be replayed, sniffed, and/or modified after successfully executing Reply, MITM, and Command Modification attacks. They provided some general defence theories with no implementations including the use of encryption with external hardware cipher models to defeat Reply and Command Modification attacks along with static entries in the ARP tables to counter MITM attacks. Additionally, they suggested measuring the PLC response time as a MITM defence mechanism, given that it is slightly less during a benign communication compared with the malicious scenario.

In terms of response to the attacks for ICS, the closest work to ours is presented in [21] where the authors proposed an Anomaly Detection Module (ADM) which sends estimated values to the controller when an attack is detected. Their results showed that the ADM module has a considerable amount of success when an attack is detected. However, the response is not effective when a false and/or a positive alarm is raised. In addition, although using the controlled environment allows experimenting with a wide range of control systems, it is rather unclear whether the work proposed in [21] is applicable to real scenarios e.g. a real PLC in the industry.

In this paper, we implemented and fully discuss memory attacks on a real PLC of a Festo MPS PA Compact Workstation Rig, which is a working model of a clean water supply system, targeting the system's sensor/actuator vulnerabilities which is novel and also differs from the existing work such as work represented in [20]. Additionally, in this paper, we proposed the PLC's inbound detection and response to the attacks which is lacking in the existing work. Our proposed technique differs from [21] which is the only paper we found relevant to our proposed response technique. Given that our technique is implemented inside a PLC, unlike [21] we did not rely on an external module/equipment that can be tampered by attackers. Additionally, our proposed technique is implemented on a real and modern PLC currently used in the industry. Furthermore, the work presented in this paper is different from the existing work on the memory attacks in general in terms of the application where our focus is on PLCs in CPS/ICS/SCADA systems rather than computer systems in general.

3. Design and Implementation

In our previous work [22], we physically modelled a continuous clean water supply system using the default configuration of the Festo MPS PA Compact Workstation Rig. However, to make this experiment more realistic, we changed the configuration of the Festo Rig. Fig. 2 shows our final Festo rig equipment along with the control diagram representation and tags. The major changes are discussed as follows.

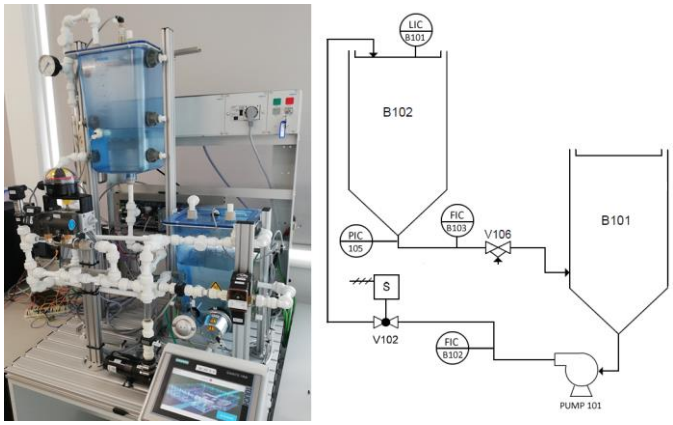

Fig. 2 Festo rig (on the left), Festo rig control diagram (on the right)

With the original Festo rig configurations, the water going through the pipeline is supplied using a solenoid valve. This valve can only be either open or close. Therefore, for a more realistic approach and also to model a better water demand curve, the solenoid valve, which is tagged with V102 on Fig. 2, was swapped with the proportional valve, which is tagged with V106.

In our previous work [23], we proposed a water demand model based on the real model of power consumption in the UK [24]. The same model is used in this paper. We keep this water demand model simplistic, so it could be reproduced in the future. According to the experiment presented in this paper, the main water is usually gravity fed to a surrounding area from a water tank located at a high elevation to sustain a suitable delivery pressure. This water tank is tagged with B102 in Fig. 4. We tested the control implementation after switching the proportional and solenoid valves, as a result, the water pressure going from the reservoir tank (B102) to the lower tank (B101). The lower tank simulates a town in our scenario. We increased the height of the tank B102 approximately 25 centimetres to obtain better water pressure. Fig. 2 shows the change of height in tank B102. When the water is flowing from tank B101 to the reservoir tank B102 the solenoid valve is opened, thereby the water is able to go through. Nevertheless, one of the issues we encountered was that when the pump is not operating the water goes back to the tank B101 through the pump, which alters the behaviour of the control system. To solve this issue, we simulate the solenoid valve as a non-return valve; as a result, we avoid the water returning when the pump is not operating.

The default configuration of the Festo Rig includes only one flowmeter, which is placed right after the pump and tagged as FIC B102 in Fig. 2. This allows implementing a control system which is self-regulation. The water from the main supply is pumped via a piping system. The flow rate is detected by means of an optoelectronic vane sensor. To expand the possible control techniques implementations, another flowmeter is added to the Festo Rig and it is placed on the outlet of the reservoir tank B102. This sensor is tagged as FIC B103 in Fig. 2. Adding a new sensor allows implementing a feedforward control using the values provided by the flowmeters. Fig. 2 shows the placement of this sensor tagged as FIC B103. The Festo Rig includes a pressure control function, which involves one pressure tank and one pressure sensor. This sensor measures the pressure in the pipes when the pump delivers water from the tank B101 to the tank B102. Right after the tank B102, we have added another pressure sensor, which is tagged with PI 105 in Fig.2, because it allows the implementation of pressure control which is capable of measuring the weight of water inside the reservoir tank B102.

3.1 Testbed Architecture

Fig. 3 shows the architecture of the testbed used in this paper which includes the following components:

--Festo MPS PA Compact Workstation Rig. It includes four sensors, one actuator, one solenoid valve and one proportional valve.
--SCADA system implemented in a Windows OS computer.
--Human Machine Interface (HMI).
-- PLC (SIMATIC S7-1500).
--Attacker Machine equipped with Kali OS.

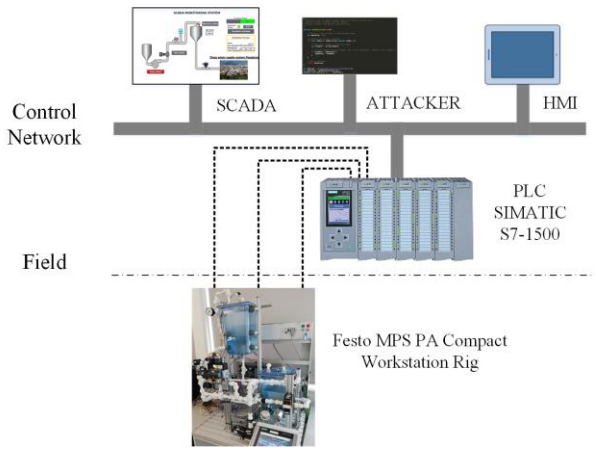

Fig. 3. Testbed Architecture

3.2 PLC Code

The control techniques implemented in the Siemens PLC to control the operation of the model of our clean water supply system (Festo Rig) are described as follows. Table I summarizes the control techniques implemented and the sensors involved in each technique. The column Tag can be mapped to Fig. 2 for a better understanding of the implementation.

TABLE I
CONTROL TECHNIQUES IMPLEMENTED IN THE PLC

Control Technique	Sensor(s)	Tag
PID	Ultrasonic Sensor	LIC/B101
	Pressure Out	PIC/105
Cascade	Flowmeter In - Ultrasonic Sensor	FIC/B102 - LIC/B101
	Flowmeter In - Pressure Out	FIC/B102 - PIC/105
FeedForward	Flowmeter In - Flowmeter Out	FIC/B102 - FIC/B103

3.2.1 PID Implementation

The PID controller is a control technique which is based on early mechanical and electronic controllers and consists of three basic control actions: proportional, integral and derivative which act on the error to determine the controller output (OP). The effect of these parameters can be modified to match or tune the controller to the dynamics of the process to be controlled[25]. There are several forms of the PID algorithm implemented on today's controllers, but they achieve similar levels of control. Fig. 4 shows the representation of the parallel or separated form of PID controller. The controller output (OP) is determined from the error (E) which is obtained by subtracting the process variable (PV) from the Setpoint (SP) [4].

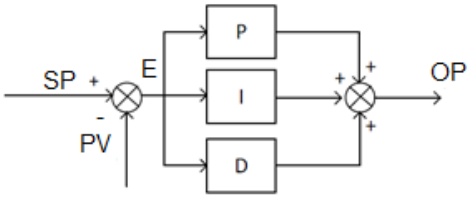

Fig. 4. Parallel PID Controller Structure.

Based on the process requiring control, there are four possible combinations for PID controller: Proportional Only (P), Proportional Integral (PI), Proportional Derivative (PD), Proportional Integral Derivative (PID).

In this paper, the PID water level control was implemented on the clean water supply system testbed. The water level of the tank is measured using an ultrasonic transducer to provide the Process Variable (PV). The Output (OP) of the controller was used to regulate the speed of the delivery pump to maintain the required tank water level Set Point (SP). The controller was implemented on a Siemens PLC S7-1500 and tuned using the Ziegler Nichols methodology[26].

The implementation of the PID Control for this control process takes the latest version of the PID block available on Siemens TIA Portal V14. The PID control is allocated inside a cyclic interrupt block which is active every 100ms. The Setpoint value is obtained from the HMI interface. This value is stored in an optimized datablock and then forwarded to the PID Control. The process variable is previously calculated using the values obtained from the ultrasonic sensor or pressure sensor. These values are obtained from the analogue input memory (I) of the PLC. The output value of the PID control represents the required speed at the pump on the

scale of 0 to 100 percent. However, it is required to convert the PID output in a value understandable for the pump controller. To achieve this, we created a function block that converts this representative speed in an integer value between 0 and 27648 for the D/A process. Finally, this value is written in the digital analogue output memory of the PLC.

3.2.2 Cascade Controller Implementation

Cascade control is an advanced control strategy used to improve the control performance over a single loop controller. The cascade architecture consists of two controllers, requiring two measured process variables and one final output. The outer loop controller's output is suitably ranged to become the inner loops set point. In this paper, the clean water supply system can be also controlled by a cascade control, as shown in Fig 5. The cascade control technique is beneficial when the inner loop is at least three times more dynamic than the outer loop, as it is in our scenario. We started designing and implementing the PID controller, for the model of a clean water supply system, in the inner loop. The parameters used are the flow_in as the process variable, and its setpoint is given by the output from the outer loop controller. For the outer loop, the process variable is the reservoir tank level via the ultrasonic sensor or the pressure_out. The primary controller is in the range of 0 to 100. The secondary controller expects a setpoint in the range of 0 to 4. This is because the maximum flow of this control system is 4.1 litres/min when the pump is working at 100% of its capacity. As a result, the output from the primary controller is scaled down. The output from the secondary controller drives the pump and maintains the water level in the reservoir tank.

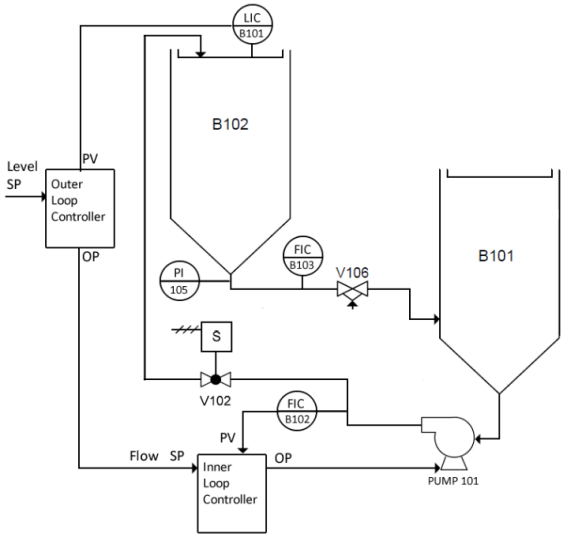

Fig. 5. Cascade Water Level Control.

3.2.3 Feedforward Controller Implementation

Feedforward control systems measure the disturbance and modify the controller output before the process variable has time to respond. For this to be successful, the designer requires to understand how the disturbance will affect the process variable. In this work, we have also applied this control strategy. In this case, the disturbance will be the change in the outlet flow from the reservoir tank. If we are controlling the tank water level using Cascade control, we can feed this forward to the inner loop SP as shown in Fig. 6.

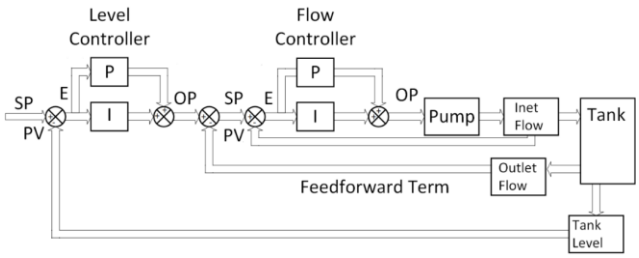

Fig. 6. Cascade level control with feedforward.

4. Testbed Scenarios

In this section, we define the normal operation along with the attack scenarios in our implemented testbed.

4.1 Normal Operation

In this paper, an uninterrupted clean water supply system was modelled in the Festo MPS PA Compact Workstation Rig. In our testbed, we assumed that the water has already passed a treatment process and it is ready to be distributed for example along a town. The tank B101 contains the water that supplies the reservoir tank B102 through the variable speed pump 101. The water demand from customers was modelled and implemented using the proportional valve of the Festo Rig. In a normal operation, the water level in the reservoir tank B102 needs to be maintained at a certain setpoint introduced by the operator. To achieve this, we implemented three different control techniques such as PID, Cascade and Feedforward. Table I describes each technique and sensors used during its operation. For example, the Feedforward control techniques used flowmeter in and flowmeter out sensors the PID control mechanism used the pressure out and ultrasonic sensors.

4.2 Attack Scenario

Industrial control networks were isolated from the traditional computer networks or business networks by placing their components in an "air-gapped" environment. This means they were not reachable from external devices[27]. With the development of technology and the introduction of industry 4.0, most of the companies seek to enable the connectivity between the physical processes and the Internet because it allows obtaining benefits such as: visibility, efficiency, real time and rapid decisions and better customer experience. However, connecting the traditional ICS to the Internet exposes the previously isolated environments to all sort of cyber threats [28]. In this paper, we assume that the attacker has access to the control network and can communicate with the PLC SIMATIC S7-1500 either as an insider or external hacker.

4.3 Attack Model

We present a model of attacks to the PLC memory, which can be used to understand the possible attack vectors in an intuitive and concise way. Let us assume H denotes the attacker while C denotes the control process in operation, which in this paper involves a critical infrastructure represented by the model of a clean water supply system.

Furthermore, we denote the possible origin of the attacks to the PLC memory as T. Assuming our testbed scenario the attacks could be originated from the HMI interface denoted as H, the SCADA system denoted as S, any computer connected illegally to the network denoted as NC, and a physical attack denoted as P. Thus, we define T as follows:

$$T \triangleq \{\mathrm{H} \cup \mathrm{S} \cup \mathrm{NC} \cup \mathrm{P}\}$$

According to our model, every attack originates from an attacker h where $h \in \mathrm{H}$ by a means T towards a target C. We can model this relationship as follows:

$$h \mapsto \mathrm{t} \rightsquigarrow \mathrm{c}$$

where $h \in H$, $t \subseteq T$ and $c \in C$. The notation $\mapsto$ maps the attacker to the possible points of attack execution and the notation $\rightsquigarrow$ leads to the victim.

As it is described before, the attacks disturb the PLC memory. However, the PLC has different spaces of memory that could be affected. Thereby, we denoted A as the attack attempted, a_1 attack to the input memory, a_2 attack to the output memory and a_3 attack to the working memory. Thus, we define A as follows:

$$A \subseteq \{a_1 \cup a_2 \cup a_3\}$$

Each attack has a probability of being successful. We denoted the probability of the attack as P_a. The probability of the attack defines its severity. The higher the probability of an attack the higher the damage into the system. However, in our model a successful attack might affect the system in two different ways. We assume that the severity of the attack denoted as R can affect the control operation in two ways. It could have a severe impact on

the control operation denoted as r_1 or it could affect the performance denoted as r_2. The severity of the attack is defined as follows.

$$R: a \rightarrow \{r_1, r_2\}$$

where a ∈ A. The high probability of an attack to succeed is denoted as ∂ and a low probability is denoted as ℟. Hence, the severity of an attack a ∈ A can be represented as follows:

$$R(a) = r_1 \qquad \text{If } ℟ < P_a > \partial$$

$$R(a) = r_2 \qquad \text{If } ℟ < P_a > \partial$$

r_1 defines an attack that results in stopping the control operation, for instance damage in an actuator like the pump or a tank overflow. On the other hand, r_2 represents an attack that increases or decreases the water level without affecting the entire operation.

The attack is represented as follows:

$$a \mapsto t \rightsquigarrow c, r$$

where a ∈ A, t ⊆ T, c ∈ C and r ⊆ R.

4.4 Attack

An attack attempted a ∈ A by the intruder h ∈ H to the control process c ∈ C might affect the performance $r_1 \in R$ or the operation $r_2 \in R$. The attacker might execute one attack (single point) at the time or multiple attacks (multiple point). In this paper, the attacker attempts single and multiple point attacks to the PLC memory. The attack is considered successful when the $P_a > \partial$.

4.5 ICS Protocol

The SIMATIC S7-1500 Advanced Controllers use an industrial Ethernet standard for automation called PROFINET to communicate with other devices connected to the same Local Area Network. The PROFINET is not a Siemens proprietary protocol, instead, this standard was designed to allow controlling equipment in industrial environments with tight time constrains such as 1ms or less. The PLC SIMATIC S7-1500 allows integrating with the different environments because it provides a wide range of communication capabilities through its interfaces. The SIMATIC S7-1500 also supports TCP/IP, UDP, ISO-on-TCP, Modbus TCP and more. We take advantages of the wide variety of protocols supported by Siemens to perform the attacks to the input memory of the PLC. The attacks discussed in this paper are not exploiting vulnerabilities in the protocols discussed before, instead we take advantage that the TCP/IP implementation relies on the block-oriented ISO transport service allowing us to craft our own valid packets and send them to the input memory of the PLC.

4.6 Packet Crafting

The large variety of protocols supported by the PLC's allow them to integrate diverse control networks, however, from a security point of view, it is also one of the biggest challenges when it is required to secure such systems. One of the major issues with control protocols is lack of traffic encryption during network communication. From an attacker's point of view, it only requires dissecting the TCP/IP packet, and then understanding the parameters and values sent during the communication among the control devices. To perform the attacks, we craft ISO 8073/X.224 COTP [15] packets targeting the input memory spaces of the PLC. Fig. 7 shows the structure of the crafted packet. The Siemens PDU is wrapped in the TPKT and ISO-COTP protocols. This allows the packet to be sent over TCP/IP. Inside the Siemens PDU the parameter header contains length of the information, message and message type. The integrity part manages connection parameters whereas data contains the values written in the input memory.

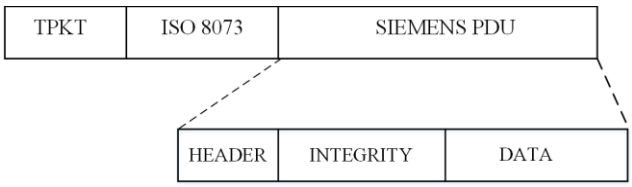

Fig. 7. ISO-COTP Packet Structure

4.7 Attack Methodology

In this section, we lay out the methodology for attacking the PLC Input Memory assuming that the attacker is already connected to the control network. The attack is divided in the steps described as follows:

4.7.1 Reconnaissance

The first step is to perform an active reconnaissance on the network. This is achieved by scanning the devices connected to the control network. By default, the PLC SIMATIC S7-1500 uses the port 102 for TCP/IP communications. Thereby, the attacker's aim is to look up for devices with such a port open. Fig. 8 shows the use of the nmap tool to execute the network scan and the response obtained from the devices connected to the network. For this scenario, we scanned the devices connected to the network 192.168.0.0/24 with the port 102/TCP open. We obtained a positive answer indicating that the port 102/TCP is open from the device 192.168.0.1. It also reveals that this is a Siemens device. The next step is identifying the device connected to the network. To perform this, we crafted an ISO 8073/X2.224 COTP packet, using the tool Scapy[29], requesting information from the PLC CPU. Fig. 9 shows the packet sent over the network and the response received from the PLC. As it can be seen, the response received shows the PLC model, brand, and CPU model.

```
user@pc:~# nmap -sS 192.168.0.0/24 -p 102

Nmap scan report for 192.168.0.1
Host is up (0.00051s latency).

PORT    STATE SERVICE
102/tcp open  iso-tsap
MAC Address: 28:63:36:92:BD:CA (Siemens AG -
Industrial Automation - EWA)
```

Fig. 8. Network Reconnaissance.

4.7.2 Attack to the PLC Memory.

The Siemens PLCs use a fixed space of memory for their inputs and outputs. This space of memory is updated in every PLC scan. It is possible to access these spaces of memory over the network to allow data loggers to collect information of the control process; however, it is also possible for attackers to craft packets and overwrite these areas of memory. Fig. 9 shows the crafted packet sent to the PLC over the network. This packet writes in the input space of memory addressed to the ultrasonic sensor with a value of 150. The important information is highlighted inside the dotted area in Fig. 9. Outside the dotted area is the general information of the packet sent, such as IP addresses and MAC addresses.

Fig. 9. Packet Request on the left and Response on the right.

The attacker keeps overwriting the spaces of memory at his will when this packet is coded inside a loop. For this scenario, we executed the attacks against the spaces of memory addressed to the ultrasonic sensor, pressure

sensor and the flowmeters. In the same way, it is possible to write in the spaces of memory addressed to the PLC Outputs aiming to attack devices such as the pump. Fig. 10 shows a crafted packet that writes in the space of memory addressed to the pump. This attack allows interrupting the pump's operation which might lead to disrupting the control process operation.

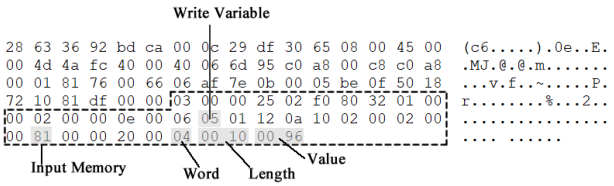

Fig. 10. Crafted Packet to the PLC input memory.

5. Attack detection and response

In this paper, we introduce a novel mechanism of detection and response to cyber-attacks in the PLC memory. Most of the research focuses on detecting the attacks at the TCP/IP level in the control network [16], [17]; however, we in this paper propose to detect such memory changes inside the PLC. Furthermore, we implement in the PLC a mechanism of response to attacks when changes into the PLC memory are detected.

5.1 Attack Detection

The aim of the attacker is to overwrite the spaces of the memory of the PLC addressed to the Inputs. The PLC updates its memory each cycle, which is usually measured in one millisecond. Thus, the attacker has to be fast enough to keep the wrong value in the input memory the majority of the time. According to the data obtained from the testbed, the attacker is able to overwrite the PLC memory addressed to the Inputs with 67% of wrong values during one second, which represents 670 values out of 1000. Fig. 11 shows a flow chart of the algorithm implemented in the PLC for detecting the changes in its memory. Table II presents the description of the variables represented in Fig. 11. The flow chart is described as follows.

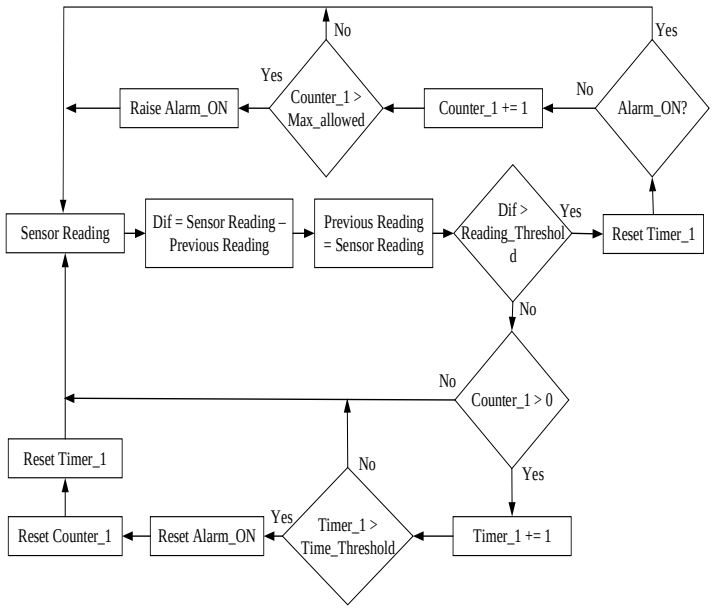

Fig. 11. Detection to attacks to the PLC Input Memory.

TABLE II
VARIABLES DESCRIPTION

Variable	Description
Dif	Contains the difference between the current and previous sensor reading.
Timer_1	How long the alarm has been triggered
Counter_1	Number of times the variable Dif has been greater than the established reading threshold
Alarm_ON	This variable is set to ON when an attack on the control process has been detected.
Reading_Threshold	The maximum difference allowed between the current and previous reading.
Time_Threshold	When the attack stops, how much time has to pass to turn off the alarm.
Max_Allowed	The number of wrong readings before turning on the alarm.

This algorithm takes the readings from the inputs addressed to each one of the sensors connected to the PLC. The variable Dif stores the subtraction of the previous and current sensor reading. For instance, for the transducer, the maximum sudden change expected in the water level is given by subtracting the value of the maximum flow when the pump is working at 100% of its capacity minus the flow with the lowest water demand.

$$Reading_{Threshold} = MaxFlowIn - MinFlowOut$$

When the variable Dif is greater than the expected value, the variable Timer_1 is reset and we verify whether the alarm has been turned on. If not, we increase the Counter_1 variable, which keeps a record of the number of times the difference between the previous and current sensor reading has been greater than expected value. When the Counter_1 variable is greater than the maximum allowed value, it turns on an alarm indicating that the space of memory addressed to that sensor is under attack. It should be noted that during the experimentation phase we realised that the external factors such as humidity, affected some of the readings obtained from the sensors. In addition, the water turbulence affected the readings of the ultrasonic sensor from time to time. For this reason, we take this into consideration when we calculate each threshold value. The main purpose is to reduce the number of generated false/positive alarms. We also take into consideration the scenario when the intruder stops the attack. When the Dif variable is less than the Reading_Threshold variable, we compare whether the variable counter is greater than zero. If so, we start increasing the Timer_1 variable. If the variable is greater than the Time_Threshold variable, it means that we can say the attack stops. Finally, we reset the alarm: Counter_1 and Timer_1 variable.

5.2 Attack Response

In related work, most of the research focuses on detecting Cyber Attacks on ICS [12], [18], only a few approaches provide a mechanism of response to intrusions [21]. One of the reasons might be that critical infrastructures are composed of complex and expensive equipment. In most of the cases replicating such systems for testing purposes is not feasible.

For instance, in an attack scenario, the operation of the process under control is disrupted when the attacker overwrites the space of memory in the PLC which is addressed to the ultrasonic sensor. Although, the supervisor console alerts about the attack, the process is affected. Fig. 12 shows the water level in the reservoir tank when the attack is executed. The x-axis shows the time elapsed and the y-axis shows the readings from the ultrasonic sensor. The dotted line represents the value written in the PLC memory. The control process understands that the water level in the reservoir tank is below three litres, for that reason the pump starts working at its maximum speed. The continuous line shows the real water level in the reservoir tank.

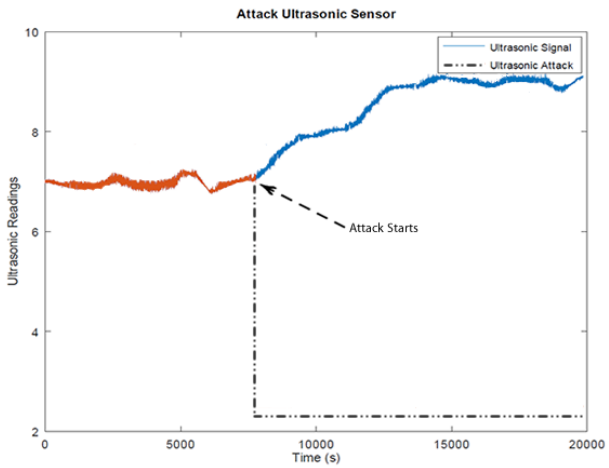

Fig. 12. Attack to the ultrasonic sensor.

In this paper, we propose different mechanisms of response to attacks to the input memory of the PLC, intending to minimize the attacks impact. We introduce the use of optimized datablocks to minimize the attacks to the input memory in addition to different control techniques for attack response.

5.2.1 Optimised Datablocks

The attack on the PLC's memory overwrites the correct readings from the sensor involved in the control process with the values injected by the attacker. The first mechanism of response implemented in this paper is to copy the values obtained at the beginning of the PLC scan in an optimised datablock and then use those values during the entire PLC scan. The advantage of using optimized datablocks is the allocation of this information in the PLC

memory is randomized, thereby, the attacker does not know its exact location. This memory optimization function is a feature available in Siemens PLC's.

5.2.2 Auto-Controller Selection

The second mechanism of response is to switch between the control techniques based on the sensors under attack. For instance, if the control process is operating with a cascade control technique using the flow_in and the ultrasonic sensor, when the attacker targets the space of memory addressed to the ultrasonic sensor and overwrites it with the invalid values, the control system detects the attack and isolates the information originated from that space of memory. The next action is to replace the ultrasonic sensor with the pressure_in sensor and continuing with the system operation. The attacker may understand how this mechanism of response operates and start attacking the ultrasonic and flow_in sensors. The immediate action of the PLC is to switch the control technique to PID using the pressure_in sensor. It is possible to switch to different control techniques such as PID, Cascade and FeedForward when an attack compromises the related sensors involved in the technique. The last mechanism of response when an attack on the PLC memory is detected and there are no other mechanisms available because the entire sensors have been compromised is to set the pump into a fixed speed. Finally, we configure the PLC CPU into stop mode to avoid further damage from the attacker.

5.2.3 Data from the Analogue Channel

The third mechanism of response involves copying the values of the analogue sensors directly from the analogue channel into an optimized datablock. This mechanism is similar to the first detection technique previously explained however in this case, the space of memory assigned to analogue input channel in the PLC has the property of being read-only, for instance direct from the A/D process. When the attack is detected the internal code of the PLC discard the values obtained directly from the input memory and starts selecting the values obtained from the analogue channel. The advantage of this mechanism of response relies on the fact that we can differentiate between an attack and a sensor failure because we compare the values obtained from the PLC memory and the signal converter. When those values are significantly different, we can conclude that an attacker has overwritten the PLC memory. Alternatively, when both values are identical, then we can conclude that it is a sensor failure, which is considered a false/positive alarm.

6. Results

We ran a set of attacks to the input memory of the Siemens PLC (SIMATIC S7-1500) to test our proposed detection techniques. The following describes the results obtained.

6.1 Optimised Datablocks

The first proposed mechanism of response, which is optimised datablocks, shows that although the control process is slightly affected, it is possible to minimize the impact of the attack. Fig. 13 shows the monitoring of the ultrasonic sensor in normal operation and when the input memory of the ultrasonic sensor is under attack. Unlike the ultrasonic sensor signal as shown in Fig. 12, this signal increases during the execution of the attack.

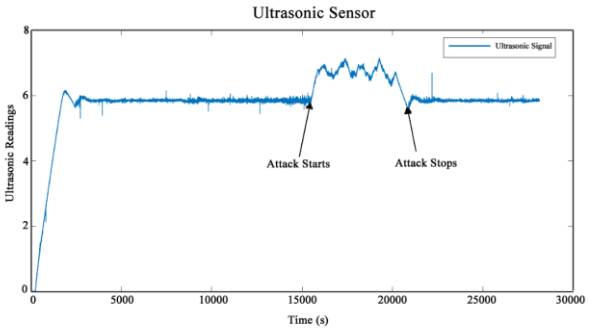

Fig. 13. First mechanism of response to memory attacks.

6.2 Auto-Controller Selection

The second mechanism of response to attacks to the input memory of the PLC automatically selects the controller strategy depending on the availability of the sensors. In Fig. 14, the process under control starts with the cascade controller using the flow_in sensor for the inner controller and the ultrasonic sensor for the outer controller. The intruder starts attacking the input memory addressed to the flow_in sensor. When the PLC detects the attack, it switches automatically to the PID controller using the ultrasonic sensor. In the scenario where the attacker wants to perform further damage and also attacks the space of memory addressed to the ultrasonic sensor, the PLC responds switching to the pressure sensor. In this scenario, the flow_in sensor and ultrasonic sensor are discarded and not used when the attack is present. The supervisory console alerts about the attacks.

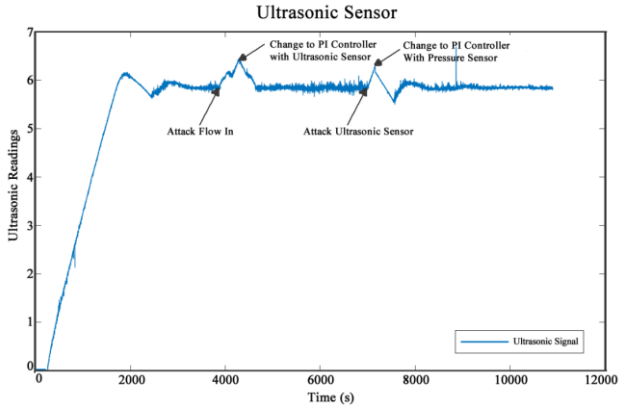

Fig. 14. Second mechanism of response to memory attacks.

6.3 Data from Analogue Channel

The third mechanism of response, which is reading data from analogue channel, shows that copying the sensors values directly from the analogue channel reduces the impact of the attack to zero, although it should be noted that we add a small delay to the control process because making a copy from the sensors readings increases the control operation time. However, because in this testbed the time constraint is not as important as in a manufacturing process, we can allow some time delay in this process, thus, this response to the attacks to the input memory is feasible. Fig. 15 shows the signal from the ultrasonic sensor and the points where the intruder executes the attack and the response from the PLC. The attack does not affect the operation of the system and it maintains the water level in the desired setpoint.

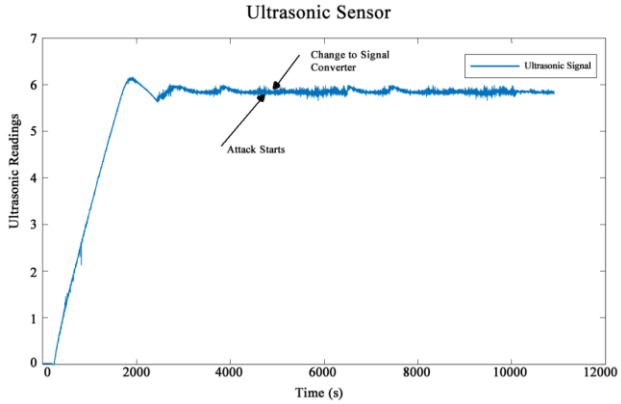

Fig. 15. Third mechanism of response to memory attacks.

7. Discussion

In this section, the research questions stated at the beginning of this paper are addressed as follows.

RQ1: How do cyber-attacks to the memory of the PLC affect the control process operation?

The PLC is able to receive and transform electrical signals, from the sensors involved in the control process, in numerical values through the A/D converter. These numerical values are stored in spaces of memory addressed

to the inputs in the PLC. The control techniques such as PID, Cascade and Feedforward perform operations with those values, and drive actuators connected to the PLC outputs. When an attacker has an access and overwrites the spaces of memory addressed to the input memory of the PLC, the implemented control techniques perform operations with tampered values, as a result, the devices driven by the PLC are affected. For instance, in Fig. 14 presented in section 3, the intruder overwrites the input memory addressed to the ultrasonic sensor with small values pretending that the water level in the tank is low, as a result the pump speeds up its operation increasing the water level in the tank.

RQ2: Is it possible to minimize the impact of cyber-attacks to control systems using control methods?

We demonstrate that it is possible to minimize the impact of attacks to the testbed implemented for this research by embedding in the PLC a mechanism of attack detection and response. When an intruder overwrites the PLC input memory, the PLC detects the attack and writes the values obtained from the sensors in an optimized datablock. These values are used through the entire PLC cycle reducing the impact of the attack. This technique is feasible because the attacker is not as fast as the PLC cycle. Thus, in some scans, the PLC will copy correct values and in some the PLC will be affected by the attacker. For instance, in Fig. 15 presented in section 4, the intruder performs an attack to the input memory addressed to the ultrasonic sensor, when the attack is detected the PLC copies the values from the input memory and uses the same value during the entire cycle. It can be seen that the attack is still present and affect the level of the water tank, however, the system operation continues. It should be noted that an alarm raises when the attack is detected giving time to the operator to apply a manual action that stops the attack. In addition, this technique is feasible because we use the memory optimization feature available on Siemens PLC's which permits the allocation of information in the PLC memory in an address defined internally by the PLC.

RQ3: What countermeasures could be taken into consideration to continue with the operation of a control system when a cyber-attack is detected?

In this research, we analysed and implemented an algorithm that detects and respond to attacks to the PLC memory. To achieve this, we implemented different control techniques involving the sensors available. Thus, when an intruder executes an attack from a single-point to one sensor, the algorithm of detection and response isolates the sensor compromised and analyses the possible control techniques combinations available on Table I.

8. Conclusions

In this paper, we analyse the impact of network attacks to the area of memory addressed to the PLC inputs. The attacks performed in this research show that it is possible to disrupt the control system operation bringing the system to an unstable state. The attacks are performed to the input memory of the PLC; however, it should be noted that it is also possible to execute the same attacks to the PLC output. For instance, the attacker could drive the pump at different speeds by overwriting the space of memory addressed to it.

The same mechanism of defence which is implemented in this paper to detect the attacks to the inputs could be used to detect the attacks to the output; however, until now we have not found a mechanism of response that mitigate those attacks. Current research does not analyse the potential damage of performing these types of attacks. The main reason could be that most of the research is based in theoretical analysis only and the cost of implementing testbeds for research purposes is significantly high.

Most of the current research for attack detection on industrial control systems focused on detecting anomalies in the control network traffic and then alerting of possible intrusions. Unlike other approaches, our mechanism of detection and response to attacks to the PLC memory is implemented in the PLC itself, meaning that external equipment is not required for detecting the cyber-attacks leading to reduce the response time and overall cost. The results obtained from the mechanisms of response to attacks, shows that obtaining the sensor readings directly from the analogue channel allows us to minimize the impact of the attacks to the input memory, however, it should be considered that performing this action add a small delay in the control system operation. It can be argued that the testbed implemented here is not affected for small delays, however, in a control process where the time of response is critical this mechanism of response might not be adequate.

This mechanism of detection and response relies in the fact that Siemens controllers have a feature called memory optimization available from the Simatic S7-1200 onwards. This feature does not have a specifically defined structure. The data elements receive only one symbolic name in the declaration and no fixed address in the block which makes difficult for an attacker to access that information.

We would therefore encourage designers to use function blocks as much as possible in their design to minimize the susceptibility to attacks to the input memory. In addition, the hardware design should also consider redundant sensor architecture aiming to switch the control strategies in case an attack is detected. We want to inspire to cyber-security and control practitioners to collaborate and analyse this challenging topic from both points of view.

8.1 Future work

In future work, we plan to apply machine learning for attack detection to the PLC memory. We want to compare in a fairly manner the approach taken in this paper with our future work using machine learning under the same conditions.

8.2 Acknowledgment

This research is supported by the School of Computing and the School of Engineering & the Built Environment of the Edinburgh Napier University. Thanks to my supervisory team for the support during this research

References

[1] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine Blackout: Implications for False Data Injection Attacks," IEEE Trans. Power Syst., vol. 32, no. 4, pp. 3317–3318, 2017.
[2] I. Nagrath and M. Gopal, "Introduction," in Control Systems Engineering, Tunbridge Wells: New Age International (P) Ltd., 2008, pp. 1–20.
[3] X. M. Zhang, Q. L. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," IEEE Trans. Ind. Informatics, vol. 12, no. 5, pp. 1740–1752, 2016.
[4] K. Kamel and E. Kamel, "Introduction to PLC Control Systems and Automation," in Programmable Logic Controllers: Industrial Control, McGraw-Hill Education, 2014, pp. 1–31.
[5] W. Bolton, "Programmable Logic Controllers," in Programmable Logic Controllers, Sixth Edit., J. Simpson, Ed. 2015, pp. 1–22.
[6] Arizton, "PLC Market - Global Outlook and Forecast 2017-2022." [Online]. Available: https://www.arizton.com/market-reports/plc-market-analysis. [Accessed: 23-Jul-2018].
[7] Frank, "PLC Manufacturer Rankings." [Online]. Available: http://automationprimer.com/2013/10/06/plc-manufacturer-rankings. [Accessed: 23-Jul-2018].
[8] E. E. Community, "The top most used PLC Systems around the world." [Online]. Available: http://engineering.electrical-equipment.org/electrical-distribution/the-top-most-used-plc-systems-around-the-world.html. [Accessed: 23-Jul-2018].
[9] Statista, "Leading automation vendors worldwide in 2016, based on revenue (in billion U.S. dollars)." [Online]. Available: https://www.statista.com/statistics/257058/ranking-of-the-leading-automation-vendors-worldwide. [Accessed: 23-Jul-2018].
[10] Siemens, "Our fastest controller for automation." [Online]. Available: https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc/simatic-s7-1500.html. [Accessed: 30-Jul-2018].
[11] Siemens, "Getting started with S7-1200." [Online]. Available: https://support.industry.siemens.com/cs/attachments/39644875/s71200_getting_started_en-US_en-US.pdf?download=true. [Accessed: 30-Jul-2018].
[12] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," 2016 Int. Work. Cyber-physical Syst. Smart Water Networks, CySWater 2016, no. Figure 1, pp. 31–36, 2016.
[13] FESTO, "MPS PA Compact Workstation with level, flow rate, pressure and temperature controlled systems." [Online]. Available: https://www.festo-didactic.co.uk/gb-en/learning-systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and-temperature-controlled-systems.htm?fbid=Z2IuZW4uNTUwLjE3LjE4Ljg4Mi40Mzc2. [Accessed: 07-Jul-2018].

[14] S. Adepu, J. Prakash, and A. Mathur, “WaterJam: An Experimental Case Study of Jamming Attacks on a Water Treatment System,” Proc. - 2017 IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2017, pp. 341–347, 2017.
[15] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, “Guide to Industrial Control Systems (ICS) Security,” 2015.
[16] C. M. Ahmed and A. P. Mathur, “Hardware Identification via Sensor Fingerprinting in a Cyber Physical System,” Proc. - 2017 IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2017, pp. 517–524, 2017.
[17] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, “Model-based security analysis of a water treatment system,” Proc. 2nd Int. Work. Softw. Eng. Smart Cyber-Physical Syst. - SEsCPS ’16, pp. 22–28, 2016.
[18] S. Adepu and A. Mathur, “From Design to Invariants: Detecting Attacks on Cyber Physical Systems,” Proc. - 2017 IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2017, pp. 533–540, 2017.
[19] X. Clotet, J. Moyano, and G. León, “A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of Critical Infrastructures,” Int. J. Crit. Infrastruct. Prot., 2018.
[20] A. Ghaleb, S. Zhioua, and A. Almulhem, “On PLC network security,” Int. J. Crit. Infrastruct. Prot., vol. 22, pp. 62–69, 2018.
[21] A. A. Cárdenas, S. Amin, and Z. Lin, “Attacks Against Process Control Systems : Risk Assessment , Detection , and Response Categories and Subject Descriptors,” Security, pp. 355–366, 2011.
[22] A. Robles-durazno, N. Moradpoor, J. Mcwhinnie, and G. Russell, “A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system,” in In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2018), 2018.
[23] A. Robles-durazno, N. Moradpoor, J. Mcwhinnie, G. Russell, and I. Maneru-Marin, “Implementation and Detection of Novel Attacks to the PLC Memory of a Clean Water Supply System,” in In CITT 2018.
[24] NORDPOOL, “Consumption.” [Online]. Available: https://www.nordpoolgroup.com/Market-data1/Power-system-data/Consumption1/Consumption/ALL/Hourly1/?view=table. [Accessed: 30-Apr-2018].
[25] K. H. Ang, G. Chong, and Y. Li, “PID control system analysis, design, and technology,” IEEE Trans. Control Syst. Technol., vol. 13, no. 4, pp. 559–576, Jul. 2005.
[26] D. Valério and J. S. da Costa, “Tuning of fractional PID controllers with Ziegler-Nichols-type rules,” Signal Processing, vol. 86, no. 10, pp. 2771–2784, 2006.
[27] E. Byres, “The myths and facts behind cyber security risks for industrial control systems,” Proc. VDE Kongress, pp. 1–6, 2004.
[28] M. Rüßmann et al., “Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries,” Bus. Inf. Syst. Eng., vol. 6, no. 4, pp. 239–242, 2015.
[29] Y. Lopes, D. C. Muchaluat-Saade, N. C. Fernandes, and M. Z. Fortes, “Geese: A traffic generator for performance and security evaluation of IEC 61850 networks,” IEEE Int. Symp. Ind. Electron., vol. 2015–Septe, pp. 687–692, 2015.