UNIVERSITATEA POLITEHNICA BUCUREȘTI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

|  |  |
| --- | --- |
| upb | cs |

PROIECT DE CERCETARE

Implementarea si verificarea comparativă a firewall-urilor pentru protocoale seriale industriale

Ioana - Laura Popescu

**Coordonator științific:**

Prof. dr. ing. Dumitru Cristian Trancă

BUCUREŞTI

2019

UNIVERSITY POLITEHNICA OF BUCHAREST

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

COMPUTER SCIENCE DEPARTMENT

|  |  |
| --- | --- |
| upb | cs |

RESEARCH PROJECT

Comparative implementation and verification of serial-protocol dedicated firewalls

Ioana - Laura Popescu

**Thesis advisor:**

Prof. dr. ing. Dumitru Cristian Trancă

BUCHAREST

# 2019

# CUPRINS

[CUPRINS 1](#_Toc536714851)

[1 Introduction 2](#_Toc536714852)

[1.1 Common Security Challenges 2](#_Toc536714853)

[1.2 Attacks 3](#_Toc536714854)

[1.2.1 Attacking field level devices 4](#_Toc536714855)

[1.2.2 Attacking control level devices 4](#_Toc536714856)

[2 Protocols of interest 5](#_Toc536714857)

[2.1 Modbus 5](#_Toc536714858)

[2.1.1 Modbus TCP 7](#_Toc536714859)

[2.1.2 Modbus over TCP 7](#_Toc536714860)

[2.2 Function codes 7](#_Toc536714861)

[2.3 IEC 61850 8](#_Toc536714862)

[3 Existing security solutions 9](#_Toc536714863)

[4 Bibliografie 13](#_Toc536714864)

# Introduction

## Common Security Challenges

In the last 2 to 3 decades, the industrial environment has changed significantly, by the fact that network control systems have become almost inevitable in modern Industrial IoT Systems. [1]

However, along with the growth of the network and standard protocols usage, the number of vulnerabilities has grown as well.

According to [1], before adopting „Internet Technology” approach, most of the IioT systems used to operate in isolation with the „outside world” so there was absolutely no concern regarding the vulnerabilities which were faced only by the enterprise information systems for decades.

Consequences of previously mentioned threats can be, of course, much more severe in Inustrial IoT than in IT enterprise.

For example:

* Most that could happen when an attack has occured in IT enterprise is loss of data or expose of confidentiality. In IIoT, undefined behaviour can cause loss of lives.
* By the nature of these systems, and the fact that they use protocols like ModBus enabled anyone who could interface ModBus to communicate, and therefore sending commands or read sensor data.

Furthermore, in [1], because of the fact that in the past, all the IIoT operations were happening in isolation, authentication and encryption have been overlooked.

Most Industrial Control Systems are basically made of sensors/actuators, along with control devices (PLC’s) and many of them are monitored by HMI (Human Machine Interfaces), which are, most probably, Internet enabled [1].

As a prerequisite, any industrial control system that involves Control Level Equipment can be considered as part of IIoT category.

Because of this, we can confirm that attacks can vary when targetting different parts of the industrial system. Having this said, the severity of theire respective outcomes will vary as well. We will discuss possible attacks and a small categorization below.

## Attacks

Authors of [2] mentioned that the vulnerable points in Industrial Automation Control Systems are usually between Field level devices (sensors/actuators) and control level devices as they rely on protocols with no security capabilities.

Attacks can be launched on any of the devices mentioned above regardless their placement in the system’s network. There are several outcomes categorized based on the effect obtained of the attack [10], for instance:

* Equipment failure:
  + Power outages
  + Blocking elevators, causing safety risk, loss of lives
  + Blocking pipelines, causing lost of production and/or poor quality
  + Erasing programs in controllers
  + Mechanically braking running motors and destroying the equipment
* Data gathering
  + Immediate effects are not visible

We will further discuss the device-oriented categoizing of attacks in general, which can be done by approaching of the following [10]:

* Already having physical access to devices, and therefore the attacker is placed inside the network.
* The attacker is not inside the network, and therefore he is required to obtain previously spoken of access by applying diverse exploiting methods. This means they should study access points and somehow force the system to break into it.

## 1.2.1 Attacking field level devices

Some sensors, like analog and non-configurable digital sensors, as well as analog actuators, by not having processors or control devices attached, those are not „hackable” by network, and one should have physical access to it in order to physically damage them.

Configurable digital sensors and digital actuators, on the other hand, are „hackable” by somehow gathering control on the microprocessor attached to said device and modify output range.

Of course, they can all be physically damaged if one has physical access inside the system directly to the devices themselves.

For further information, a more detailed table is exposed in [10].

## 1.2.2 Attacking control level devices

Authors of [3] have identified three major misconceptions, and also proven them wrong, the most relevant for this project being the following:

* „SCADA systems require specialized knowledge, making them difficult for network intruders to access and control”.

They have shown that most common access control rules provided in order to protect SCADA systems are minimal and rarely protected.

One of the vulnerabilities affecting these networks is also the lack of permanent monitoring.

Authors of [1] have also exposed a few case studies, in order to strentghen the point that these vulnertabilities discussed exist, and are real and actually happening in industry.

* 1-WA, USA, June 1999 – Gas pipeline ruptured and ignited fires and killed 3 people as well as spilling 0.25 million gallons. Cause was attemtping update on live SCADA.
* 2-Australia, April 2000- SCADA radio controlled sewage equipment in Queensland, done by ex-employee, who issued a series of contro commands in order to spill substances outside the environment.
* 3-Attack of 23 States in USA, August 2003 – signal dispatch system CSX Railroad caused by worm infection used to dispatch and cargo train traffic
* 4-US, 2013- power plant decoy system under the control of a hacker group. Further details about this in [4].
* 5-US, 2003 – Seven American States remained without electricity caused by malware.[5]
* 6-Trojan caused a pipelien to explode.[6]
* 7-Night Dragon – Several Chinnese attack attempts on SCADA networks [7]

# Protocols of interest

## Modbus

Modbus is a serial standard communications protocol commonly used in connecting industrial electronic devices [8].

It is an application layer positioned at level 7 of the OSI model, that provides client/server communication between devices connected via different types of networks. [9]

Modbus is a request/reply protocol and offers services specified by function codes. It is based on the client/server paradigm and it usually operates with one master and several slaves. These function codes are elements of request/reply PDUs.[9]

It’s implementation uses:

* TCP/IP over Ethernet
* Asynchronous serial transmission via different media

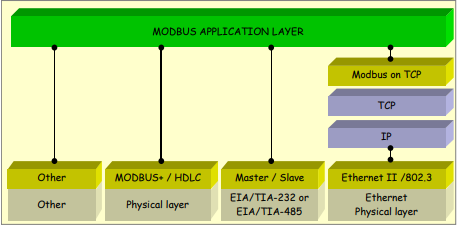


Figure 1 - Modbus communication stack 1 [9]

The Modbus protocol is independent of the underlying communication layers. The typical frame of the Modbus protocol is evidenced below. The figure clearly shows that Modbus protocol data unit (PDU) is independent of other communication layers and is encapsulated inside the ADU (application data unit).

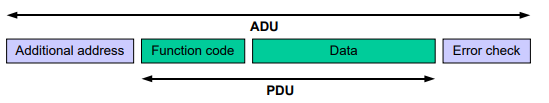


Figure 2 - Modbus frame [9]

* The function code is simply a byte that indicates to the server what kind of action it is required to perform. When in response form server and if error free, the server simply echoes back the function request it received. Otherwise indicates if it is an exception response, which happens when some error occurred.
* Data field is filled with additional information from the client that the server can use when performing the action determined by the function field (i.e. number of data bytes in the field, how many items should be handled etc.). It can also be an empty field. If coming from server and the error check field suggests that there were no errors occurred, then this field shall contain the data regarding the response.
* Error check – self explanatory

Modbus standard implementation has it’s data model based on a series of tables with different characteristics.[9]

* Discrete Input – Single bit, Read-Only, it can be provided by an I/O stream.
* Coils – Single bit, Read – Write, it can be modified by an application program.
* Input Registers – 16-bit word, Read-Only, provided by an I/O stream.
* Holding Registers – 16-bit word, Read-Write, can be modified by application program.

Each data is addressed from 0 to 65535 inside a PDU, and also the data model is composed of 4 blocks including several elements numbered consecutively.[9]

There are several versions of Modbus protocol, some of which will be detailed further.

### Modbus TCP

Is a version of Modbus which is encapsulated and transmitted through TCP, and while being encapsulated in TCP, it does not need a checksum field, since this is take care of from the lower levels of the OSI model. [10]

### Modbus over TCP

Simply the Modbus protocol encapsulated in the TCP frame and keeps calculating the checksum at application layer, although the lower layers have already implemented this feature. [10]

## Function codes

There are three categories of Modbus Functions codes listed below:

* Public function codes – publicly documented, unique, well defined function codes
* User-defined Function codes – can be user implemented, not necessarily unique and having 2 large ranges for user-defined function. If a user wishes to implement it’s own functionality and pudlish it, it must register with some RFC.
* Reserved Function codes – not available for public use.[9]

## IEC 61850

IEC 61850 is an international standard defining communication protocol, bsed on Ethernet, for inteligent electronic devices.

It’s abstract data defined within ca be mapped to a number of protocols, currently being mapped to MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Event), SMV (Sampled Measured Values), capable of running over TCP/IP networks.[11]

The major and typical part of 61850 is that of „abstracting” data items, by creating data objects independent of any underlying protocols.[12]

This allows the mapping between these data objects to any protocol with data service requirenments. The mapping between all this abstract data and services into an actual, real protocol is the final step.[12]

For instance if we add some kind of device as input to an actual IEC 61850 relay, this relay will automatically detect the device pluged into it and assign measurement units.[12]

However, IEC 61850 has several vulnerabilties as well although it has a significant number of security enhancements.[10]

Authors of [13] have analyzed several attacks on IEC 61850 and concluded that this standard still has „serious security problems”, because of the fact that all the security enhancements the protocol already has, these only help protect against only a small part of the attacks.[10]

# Existing security solutions

There are several papers and guidelines discussing the problem of cyber security in IioT and also providing solutions by adding firewalls.

For instance in [14], authors discuss the implementation of a hardware packet firewall designed to filter packets using TCP port number, IP source and destination address range, as well as a combination between port number of MAC address source and destination and IP address source and destination. Their design has been implemented in Verilog Hardware Descriptive Language and tested on Altera FPGA device, and the implementation is illustrated in Figure 3 below.

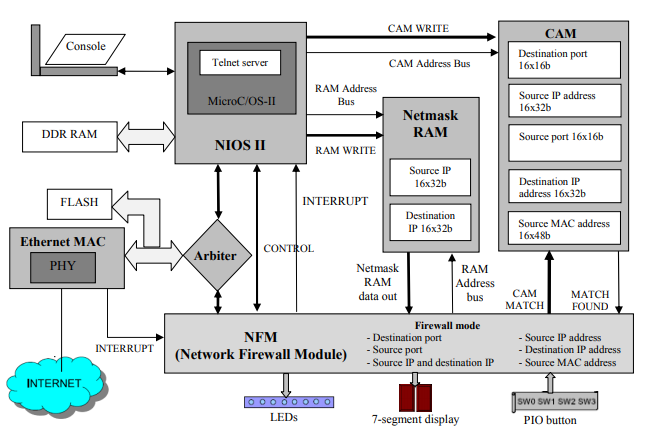


Figure 3 - Firewall Block Diagram [14]

As an overview of their main modules and implementation Flow:

* Nios II with 32-bit microprocessor unit which is used to run a software to initialize all other software according to the mode selected. When a new TCP packet arrives and targets the Nios II module, the Nios II module will interrupt the NFM, which in turn, shall analyze the CAM (Content Addressable Memory) in order to find a match against the Ethernet field of the packet and decide whether it is free to go or not. Also, The Telnet Server inside this module is used when changing the configuratin „on the fly”.

For testing purposes, the setup included 2 PC’s, the FPGA in cause, and a hub, and they’re results included no delay at all, as the data could have been processed at around 3-6.5 Giga bits per second.

Another interesting solution is the one presented in [15]. They have discussed applying „only first line of defense”, which is usually used to prevent useless traffic to reach certain controlling parts of the system, while, in the same time, allowing necessarry information pass through [15]. Their design is to split the network in 2 subnetworks, N1 and N2, where:

* N1 is the control part which requires protection
* N2 is the office part, from which, in certain situations, harm may try to pass to N1.

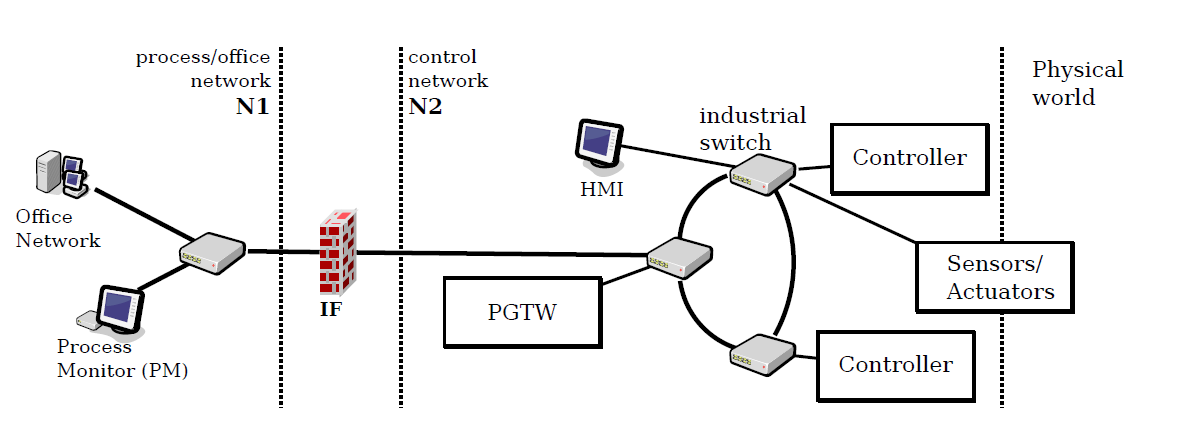


Figure 4 - Network schema [15]

The devices in the above design can be listed as follows:

* Controller – control unit that communicate with one another and toghether they build up the control core of the system
* Sensors/Actuators – self explanatory devices connected to physical system and to the controllers as well
* HMI – Human Machine Interfaces, for operators access to the system
* PGTW – Gateway between the 2 networks, and is constantly aware of the system’s status by periodic heartbeats from control units.
* IF – The Firewall

They’re main goals were such that between N1 and N2 can be exchanged only strictly authorised messages, comming from only authorised nodes inside N1, and making sure they don’t add any latency from processing the data.

In order to emulate a large number of systems based on Modbus/TCP communications, they have deployed their own processes (software) on an experimantal setup, which included:

* A linux PC with Ubuntu/Linux
* MacBook
* Some Rpi’s running linux (to also emulate arm processors)

The authors have exposed they’re results in Figure 5 by evaluating thresholds, which were determined by controlling the interfering load traffic comming from the PM (which was loaded from two separate traffic generators working in parrallel, which were hosted on two different nodes in the network).

They achieved this by periodically increasing the ammount of interfering messages, and tried to analyze the combined resulting effect of those interfering messages.

Their observatins suggested that „the firewall was able to deal with a significant amount of „office” traffic before being forced to delay messages too much and violate Process Monitor’s timing requirements.”

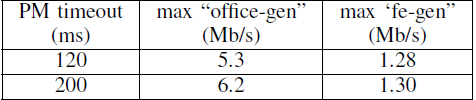


Fig. 5 - Treshold values for different PM configurations [15]

They have concluded that further investigation is needed to better understand the boundaries belonging to the critical region found by experiment.

# Bibliografie

|  |  |
| --- | --- |
| [1] | A. Mahboob and J.Zubairi, "Intrusion Avoidance for SCADA Security in Industrial Plants" 2010. [Online]. Available:  https://www.researchgate.net/publication/216485705\_Intrusion\_avoidance\_for\_Scada\_security\_in  \_industrial\_plants |
| [2] | L. Rosa et al., "Attacking SCADA systems: a practical perspective," *Integr. Netw. Serv. Manag. (1M), 2017 IFIP/IEEE Symp*., 2017. |
| [3] | "Understanding SCADA system Security". 2001. [Online]. Available: http://www.iwar.org.uk/cip/resources/utilities/SCADAWhitepaperfinal1.pdf. |
| [4] | S.Tom, "Chinese Hacking Team Caught Taking Over Decoy Plant," 2013. [Online]. Available: https://www.technologyreview.com/s/517786/chinesehacking-  team-caught-taking-over-decoy-water-plant/. |
| [5] | “SCADA Systems Face Diverse Software Attack Threats,” 2013. [Online].  Available: https://www.afcea.org/content/scada-systems-face-diverse-softwareattack-  threats. [Accessed: 30-Jan-2019]. |
| [6] | G. JAKABOCKZI and E. ADAMKO, "Vulnerabilties of Modbus RTU Protocol - A case Study", Ann. Oradea Univ. Fascicle Manag. Tchnol. Eng., 2015. |
|  |  |
| [7] | K. Gregg "Sloppy' Chinese hackers scored data theft coup with 'Night Dragon'" 2011. [Online] Available: https://www.computerworld.com/article/2513128/security0/-sloppy-chinese-hackers-scored-data-theft-coup-with-night-dragon-.html. [Accessed: 31-Jan-2019] |
| [8] | Drury, Bill (2009). Control Techniques Drives and Controls Handbook (PDF) (2nd ed.) Institution of Engineering Technology |
| [9] | Modbus Application Protocol Specification V1.1b [Online] Available: http://www.modbus.org/docs/Modbus\_Application\_Protocol\_V1\_1b.pdf |
| [10] | D.C Tranca, PhD Thesis, Optimizaions in Industrial Internet of Things [Online] Available:  https://drive.google.com/file/d/1TB9TDCKJkuFsnKEjocKFjU282Q2vJkgg/view |
| [11] | Technical Committee 57, [Online] Available: http://tc57.iec.ch/index-tc57.html |
| [12] | R. Mackiewicz, Sterling Heights, "Technical Overview and Benefits of IEC 61850 Standard for Substation Automation" [Online] Available:  https://library.e.abb.com/public/04519389e504d7ddc12576ff0070704d/ 3BUS095131\_en\_IEC61850\_Overview\_and\_Benefits\_Paper\_General.pdf |
| [13] | M.T.A Rashid, S. Yussof, Y. Yussoff, and R. Ismail, "A review of security attacks on IEC 61850 substation automation system network", in Conference Proceedings - 6th International Conference on Information Technology and Multimedia at UNITEN: Cultivating Creativity and Enabling Techynology Through the Internet of Things, ICIMU 2014, 2015. |
| [14] | R. Ajami, Anh Dinh, "Embedded Network Firewall on FPGA", [Online] Available: https://www.researchgate.net/publication/220841654\_Embedded\_Network\_Firewall\_on\_FPGA |
| [15] | M.Cheminod, L. Durante, A. Valenzano, C. Zunino, "Performance Impact of Commercial Industrial Firewalls on Networked Control Systems" [Online] Available: https://ieeexplore.ieee.org/abstract/document/# |
|  |  |
|  |  |
|  |  |
|  |  |