Hes·so VALAIS WALLIS

School of Engineering

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Degree Programme

Systems Engineering

Major Infotronics

BACHELOR'S THESIS

DIPLOMA 2024

Rémi Heredero

OT Security

PEN-testing and security about embedded devices

Professor
Prof. Medard Rieder, medard.reider@hevs.ch

Expert
Rico Steiner, rico.steiner@hooc.ch

Submission date of the report
30 August 2024

Valais excellence CERTIFIED

HES-SO Valais

SYND	ETE	TEVI
X	X	X

Données du travail de bachelor
Aufgabenstellung der Bachelorarbeit

FO 1.2.02.07.FB
che/13.03.2024

Filière / Studiengang **SYND**	Année académique / *Studienjahr* **2023-24**	No TB / *Nr. BA* **IT/2024/78**
Mandant / *Auftraggeber* ☒ HES—SO Valais-Wallis ☐ Industrie ☐ Etablissement partenaire *Partnerinstitution*	Etudiant·e / *Student/in* **Heredero Rémi** Professeur·e / *Dozent/in* **Rieder Medard**	Lieu d'exécution / *Ausführungsort* ☒ HES—SO Valais-Wallis ☐ Industrie ☐ Etablissement partenaire *Partnerinstitution*
Travail confidentiel / *vertrauliche Arbeit* ☐ oui / ja ☒ non / nein	Expert·e / *Experte/Expertin* *(nom, prénom, E-mail/Name, Vorname, E-Mail)* **Steiner Rico** - rico.steiner@hooc.ch HOOC AG, Visp	

Titre / *Titel* **OT Security labs development**

Description

The goal of this Bachelor thesis is to prepare several security related experiments in an Operational Technology (OT) environment. The environment is simulated, either using a platform named "Winter Resort Simulator Pro" or using Minecraft. In the latter case, the simulated environment still must be defined and developed. The scenarios that will be developed include wired communication by MODBUS, wireless communication using a 862 MHz radio, process control using sensors and PLC's.

Tasks

- Create an environment and select tools permitting to intercept, modify and reinject MODBUS packets, assuming the intruder has access to the communication infrastructure. Propose solutions to monitor the communication infrastructure and therefor detect and prevent such activity.
- Create an environment and select tools permitting to scan a wireless network. Intercept packets and reinject them into the network. Propose solution to harden wireless protocols against such actions.
- Create an environment and select tools to intercept sensor data to compromise the function of the controller due to invalid data. Propose approaches permitting hardening the controller software against such action.
- Prepare a setup and tools permitting to break into a PLC. Propose solutions hardening PLCs against such action.

Deliverables

- A report
- A working demonstrator
- All source code and tools
- A presentation

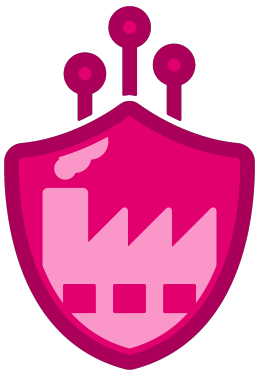
Signature ou visa / *Unterschrift oder Visum*	Délais / *Termine*
Responsable de l'orientation / *Leiter/in der Vertiefungsrichtung:* ..... [1] Etudiant·e / Student/in:	Attribution du thème / *Ausgabe des Auftrags:* **27.05.2024** Présentation intermédiaire / *Zwischenpräsentation*: **Semaine/Woche 27** (01-05.07.2024) Remise du rapport final / *Abgabe des Schlussberichts:* **30.08.2024, 12:00** Expositions / *Ausstellungen der Bachelorarbeiten:* **23.08.2024** – HEI **26.08.2024** – Monthey **29.08.2024** – Visp Défense orale / *Mündliche Verfechtung:* **Semaine/Woche 38** (16-20.09.2024)

[1] *Par sa signature, l'étudiant·e s'engage à respecter strictement la directive DI.1.2.02.07 « Travail de bachelor ».*
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 „Bachelorarbeit" zu halten.

Hes·so VALAIS WALLIS

School of Engineering

OT Security



Graduate Rémi Heredero

Objective

Produce attack scenarios that include an unsecure situation, an attack, and a solution to secure against such attack. These scenarios must be usable as the basis for a laboratory experience for students or industrial partner formation.

Methods | Experiences | Results

This thesis is separated into 2 scenarios. The first one, the Man in the Middle attack, consists of intercepting the communication between two devices and modifying the packets. The second scenario is a replay attack on a wireless transmission medium performed with a flipper zero.

The first scenario (Man in the Middle) is implemented on a modbus/TCP communication between a controller and a house security system. An attacker (a Kali Linux laptop) redirects communication to him to modify the sensor value sent to the controller. He can also send fake data. This Thesis also shows what happens if you secure your communication with certificates (modbus/TLS) but do not check the certificates. The attacker can intercept the communication in the same way. This thesis shows that it is important to always check certificates to guarantee the interlocutor.

The second scenario (Replay) is implemented on a 433MHz basic wireless communication. A flipper zero can record the transmission and replay it to trigger the same effect. Securing it is simple with implementation of a rolling code or signature. If the message is unique, an attacker cannot replay it.

Bachelor's Thesis
| 2024 |

Degree programme
Systems Engineering

Field of application
Infotronics

Supervision professor
Prof. Medard Rieder
medard.reider@hevs.ch

Information about this report

Contact Information

Author:	Rémi Heredero
	Bachelor Student
	HEI-Vs
Email:	remi.heredero@students.hevs.ch

Declaration of honor

I, undersigned, Rémi Heredero, hereby declare that the work submitted is the result of a personal work. I certify that I have not resorted to plagiarism or other forms of fraud. All sources of information used and the author quotes were clearly mentioned.

Place, date: Sion, 28.08.2024

Signature: 

Contents

Acknowledgements 1
Abstract 2
1 Introduction 3
2 Impact on Sustainability 4
3 Analysis 5
3.1 Attacks 6
3.1.1 Sniffing Attack 6
3.1.2 Spoofing 6
3.1.3 Denial of Service 7
3.1.4 Replay 8
3.1.5 Man in the Middle - Connected 8
3.1.6 Man in the Middle - Full interception 9
3.2 Communication media 9
3.2.1 Modbus 9
3.2.2 wM-Bus 10
3.3 Simulation environments 11
3.3.1 Factory I/O 11
3.3.2 Home I/O 12
3.3.3 Minecraft 12
3.4 Conclusion 13
4 Man in the Middle Scenario 14
4.1 Simulation Environment 15
4.2 Requirements 16
4.2.1 Tools 17
4.2.2 Closer look on Modbus 17
4.3 Attack on Modbus/TCP 18
4.3.1 Closer look on TCP 18
4.3.2 Modify packet on the fly 18
4.3.3 Summary 19
4.4 Implement Modbus/TLS 20
4.4.1 Closer look on TLS 20
4.5 Attack on Modbus/TLS 20
4.6 Conclusion 21
5 Replay Scenario 23
5.1 Simulation Environnement 24
5.2 Requirements 24
5.2.1 Closer look on Wireless M-Bus 24
5.3 Attack on Wireless M-Bus 24
5.3.1 Flipper Zero 24
5.3.2 Modulation FSK vs GFSK 24
5.4 Attack on basic 433 MHz transceiver 24
5.5 Security in wireless brodcast isolted devices 24

5.5.1 Closer look on rolling code 24
5.5.2 Closer look on signature 24
6 Conclusion 25
6.1 Project summary 25
6.2 Comparison with the initial objectives 25
6.3 Encountered difficulties 25
6.4 Future perspectives 25
7 Glossary 26

Figures

Figure 1: Sniffing attack 6
Figure 2: DoS attack 7
Figure 3: Replay attack 8
Figure 4: MitM on a connected network 8
Figure 5: MitM intercept everything 9
Figure 6: Factory I/O palletizer scene 11
Figure 7: Home I/O scene 12
Figure 8: Minecraft Electrical Age scene 12
Figure 9: Home I/O scene 15
Figure 10: MitM scenario implementation 15
Figure 11: ARP poisoning 16
Figure 12: Home IO details 17
Figure 13: TCP exchange 18
Figure 14: Modbus/TCP attack scenario 19
Figure 15: TLS handshake 20

Listings

Listing 1: Start an ARP poisoning with Ettercap 16
Listing 2: Put packets on queue 1 with Iptables 18
Listing 3: Redirect traffic to another port with Iptables 21

Acknowledgements

TODO

Bien que facultatifs, les remerciements sont l'occasion d'exprimer votre gratitude envers les personnes, les institutions ou les organisations qui vous ont soutenu tout au long de votre parcours universitaire. Bien qu'ils n'aient pas d'impact sur l'évaluation, les remerciements contribuent au ton général et à l'appréciation de votre thèse.

Abstract

TODO

Le résumé est une synthèse concise de l'ensemble de votre thèse, résumant les éléments clés sur une seule page, tels que

- Informations générales
- Objectif
- Approche et méthode
- Conclusions

Keywords: *OT*, *security*, *man in the middle*, *modbus*, *replay*, *attack*, *flipper*

1 Introduction

TODO

L'introduction sert à présenter le sujet de votre mémoire de bachelor et à éveiller la curiosité du lecteur par une vue d'ensemble. Nous expliquons ici pourquoi elle est importante et comment elle est structurée. Vous pouvez considérer l'introduction comme une accroche pour votre mémoire de licence. Vous éveillez l'intérêt et donnez un avant-goût en présentant votre motivation, votre méthode et l'état de la recherche dans votre introduction.

Dès l'introduction, convainquez vos examinateurs que votre mémoire de licence sera passionnant. Si votre professeur commence à lire votre mémoire avec impatience et intérêt, les chances d'obtenir de bonnes notes sont plus élevées. Accordez une attention particulière aux éléments suivants dans votre introduction :

- Introduire le sujet - Qu'est-ce qui caractérise le sujet ?
- Introduire l'objectif - Que voulez-vous atteindre avec votre thèse ?
- Susciter la curiosité du lecteur - Qu'est-ce qui motive le lecteur à poursuivre sa lecture ?
- Décrire la pertinence - Pourquoi ce mémoire de licence est-il scientifiquement pertinent ?

L'introduction doit contenir les éléments suivants :

- **Présentation initiale du sujet** - Vous introduisez le sujet par un "appât" passionnant. Vous fournissez des informations initiales sur le sujet et l'objet de la recherche et expliquez l'état actuel de la recherche.
- **Pertinence de la motivation du sujet** - Vous justifiez la pertinence de votre sujet (d'un point de vue scientifique) et le placez dans le contexte de votre domaine. En outre, il est souvent demandé que vous divulguiez votre motivation personnelle.
- **Description du problème et délimitation thématique** - Au moyen d'une question de recherche spécifique (ou d'une hypothèse), vous présentez votre intérêt de recherche explicite. Si nécessaire, expliquez les termes techniques.
- **Objectifs** - Votre introduction doit clairement indiquer l'objectif de votre travail et le résultat que vous espérez obtenir à l'issue de la rédaction de votre mémoire de licence.
- **Méthode** - Vous expliquez l'approche et justifiez le choix de la méthode.
- **Structure du mémoire de bachelor** - Enfin, vous donnez au lecteur un aperçu général de votre mémoire de bachelor en expliquant la structure, en montrant le fil rouge et en expliquant comment la question de recherche est résolue.

2 Impact on Sustainability

This section explores the impact of this thesis on sustainability, with a specific focus on the United Nations' Sustainable Development Goals (SDGs) . By examining the intersection of OT security and sustainability, this section demonstrates how securing industrial and home automation systems contributes to achieving global sustainability targets.

The SDGs provide a blueprint for achieving a better and more sustainable future. This thesis aligns particularly with the following goals:

Goal 9: Industry, Innovation, and Infrastructure
Industry 4.0 relies heavily on interconnected OT systems. By addressing vulnerabilities and enhancing the security of these systems, this thesis promotes the development of robust and resilient infrastructure. Secure industrial processes foster innovation and sustainable industrialization. This thesis contributes to building infrastructure that supports economic development and human well-being, with a focus on sustainable industrialization and fostering innovation.

Goal 11: Sustainable Cities and Communities
Home automation systems are integral to the development of smart cities. This thesis examines security measures for these systems, ensuring they are protected against cyber threats. Secure home automation contributes to the safety, efficiency, and sustainability of urban environments. By protecting the systems that manage energy use, water distribution, and waste management, this research supports the development of cities and human settlements that are inclusive, safe, resilient, and sustainable.

Goal 12: Responsible Consumption and Production
Efficient resource management is a key aspect of responsible consumption and production. This thesis enhances the security of systems that monitor and control resource usage, such as smart meters and automated manufacturing processes. By preventing tampering and ensuring accurate data collection, this research helps optimize resource consumption and reduce waste. This aligns with the goal of ensuring sustainable consumption and production patterns.

Conclusion
This bachelor thesis on OT security helps to advance on sustainability goals. By enhancing the security and reliability of industrial and home automation systems, this research supports the UN's Sustainable Development Goals, promoting a more sustainable and secure future. Integrating security into these systems is crucial for sustainable development, highlighting the need for interdisciplinary approaches in addressing global challenges.

3 Analysis

This section discusses various attacks, communication media and simulation environments that could be used in the laboratory. It aids in selecting the appropriate attack on the right medium and simulation environment, essential for the future laboratory. The requirements of this thesis include the use of Modbus and an attack with the Flipper Zero device.

Contents

3.1 Attacks 6
3.1.1 Sniffing Attack 6
3.1.2 Spoofing 6
3.1.3 Denial of Service 7
3.1.4 Replay 8
3.1.5 Man in the Middle - Connected 8
3.1.6 Man in the Middle - Full interception 9
3.2 Communication media 9
3.2.1 Modbus 9
3.2.2 wM-Bus 10
3.3 Simulation environments 11
3.3.1 Factory I/O 11
3.3.2 Home I/O 12
3.3.3 Minecraft 12
3.4 Conclusion 13

3.1 Attacks

Numerous attacks can occur in the context of OT security and can be classified into different categories. This thesis covers some attack as following, though many others exist.

3.1.1 Sniffing Attack

This attack consists in listening to the communication between two devices [1]. It can be performed on every communication medium with varying levels of difficulty. Wireless communication is particularly vulnerable because anyone can intercept the signals. For example, in Figure 1, Alice sends a message to Bob over the air without encryption, allowing Eve to listen to and read the message.

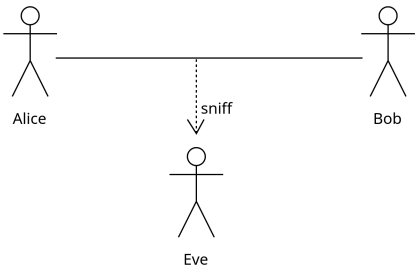


Figure 1: Sniffing attack

A sniffing attack can be performed to get secret information or understand a chemical recipe, for example. It can also be used as part of other more complex schemes.

Security Measures

To protect against sniffing attacks, the communication must be encrypted. The encryption must be strong enough to resist the attacker. A simple symmetrical encryption is enough. Particular attention must be paid to the key exchange. Protocols such as Diffie-Hellman with symmetrical encryption are recommanded to ensure protection against sniffing attacks.

3.1.2 Spoofing

In network security, spoofing involves impersonating another device. This attack is often used in combination to other attacks, such as creating a fake Wi-Fi hotspot. In the context of Operational Technology (OT) security, spoofing is less relevant and will not be discussed further in this thesis.

3.1.3 Denial of Service

An attack by DoS aims to render a service unavailable by overloading a device or network with messages [2]. In OT environments, a capable computer can execute a DoS attack effectively. To reach this goal, a device sends a large number of messages to surcharge a device or a network. In Figure 2 we can see that Mallory wants to overload Bob with messages. Bob is hence unable to answer to all messages and becomes unavailable. In Information Technology (IT) world, it is more often a Distributed Denial of Service (DDoS) attack because usually servers are more powerful than a computer, and thus can handle more concurrent messages. The attacker distributes the messages on multiple devices to make the offense more difficult to block. In OT world, it is useless to perform a DDoS to make a device unavailable since a DoS with a capable computer is generally enough. Another perspective is that OT world is typically in closed loop networks and thus not accessible from the outside world.

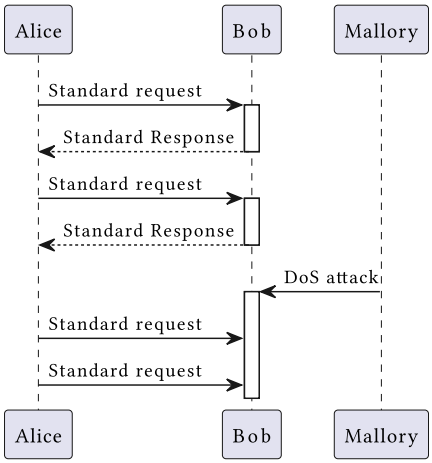


Figure 2: DoS attack

There are two primary types of DoS attacks in OT systems:

- **DoS on the communication medium**: This type of attack focuses on disrupting the data flow between devices by flooding the network with excessive traffic. This leads to network congestion and delays or blocks legitimate communication.
- **DoS on the controller**: This attack targets the processing capabilities of the controllers, such as Programmable Logic Controllers (PLCs), by sending numerous commands. This overcharge of the PLC can lead to system failure rendering it unresponsive.

> Did you know that when the Apollo 11 mission landed on the moon, the navigation system was overloaded because of a faulty sensor [3], forcing Neil Armstrong to take manual control of the landing?
> This is an example of an unintentional DoS.

Security Measures

Several ways exist to protect against DoS. At this stage, the best is to avoid doing an action at the reception of a message. Unfortunately, it is not always possible. Another way is to limit the number of messages that can be received in a particular timespan. To achieve that, it is often necessary to use another device to filter the messages. This device can be a firewall for example. However, in the case of a DDoS, it is almost impossible to block the attack. The only way is to have a very powerful filtering device. Still, even with such a device, the attack can be successful.

3.1.4 Replay

A replay attack involves resending a previously intercepted message as if it were from the original sender [4]. As we can see in Figure 3 Mallory sniffs the message between Alice and Bob. Mallory can then send the message to Bob as if Alice had sent it. This is particularly relevant in OT environment with wireless communication, such as the typical example: replaying a command to open a garage door. The attacker sniffs the command and replays it to open the door.

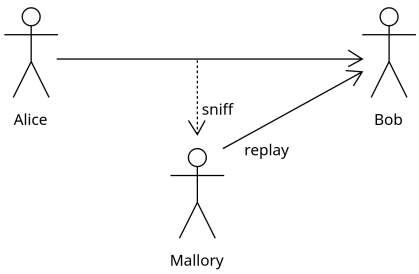


Figure 3: Replay attack

Security Measures

Two main ways exist to protect against replay attacks. It depends on whether the communication is broadcast or bidirectional.

When both devices communicate together in a bidirectional transmission, it is possible to add a timestamp to the message and sign the hash.

When communication is broadcast, the sender device is often not directly connected to other devices. In this case, it is not possible to have a timestamp. Using rolling codes is a good way to secure against replay attacks. The rolling code is a value that changes at each message. Both devices use a pseudo-random number generator. The receiver device can check whether it is a valid subsequent code.

3.1.5 Man in the Middle - Connected

A MitM attack occurs when a third party can actively intercept, modify or send packets on a network [5]. Usually, this involves connecting a new device to a star or bus network topology. Once connected to the network, Mallory (Figure 4) can perform a sniffing attack or send a message. The aim is often to intercept an message from Alice, modify it and send it to Bob.

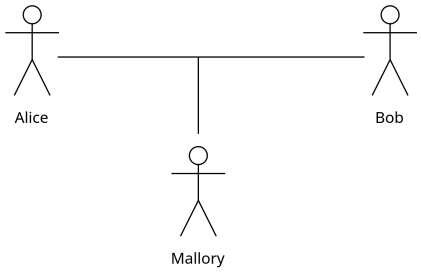


Figure 4: MitM on a connected network

Security Measures

Encryption with a symmetrical key can be used to protect the message from being intercepted, modified or impersonated. The key can be exchanged with the standard Diffie-Hellman algorithm.

3.1.6 Man in the Middle - Full interception

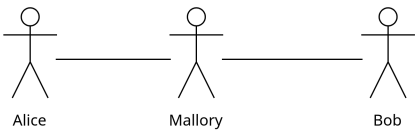


Figure 5: MitM intercept everything

When Mallory is on the gateway or between Alice and Bob, as in Figure 5, Mallory can intercept all messages and neither Alice nor Bob can be sure that they send and receive messages to the right person. This is the most dangerous attack because Mallory can impersonate Alice and Bob and send a message to the other person. Even with the security measures seen in Section 3.1.5, Mallory was able to impersonate Alice and Bob and create its own key with each other. This is why a MitM attack is so dangerous.

Security Measures

When an attacker can intercept all messages exchanged since the beginning of the communication, it is impossible to be completely sure that the messages are from the right person. We need to trust someone before. Certificates are made for this exact purpose. They are signed by a trusted third party and can be used to verify the identity of the person or device with which we are communicating. The most common type certificate is X.509 certificate.

3.2 Communication media

Different communication media are vulnerable to these attacks, highlighting the critical distinction between IT and OT security. In OT security, communication is a highly sensitive aspect, and historically, security measures were minimal or non-existent [6].

3.2.1 Modbus

Modbus is a communication protocol developed by Modicon in 1979 [7]. It involves a Modbus Master requesting data from a Modbus Slave. The client (master) sends a request to read from or write data to a server (slave). Modbus was originally designed for serial communication (called Modbus RTU). It has since been adapted for use over TCP/IP (called Modbus TCP).

Modbus RTU

Modbus with RTU is a serial, compacted, binary representation of the data. It is transported on the physical layer RS232 or RS485. Modbus RTU includes a Cyclic Redundancy Check (CRC)–16 bits checksum for error detection.

A frame is composed of:
- Address: 1 byte
- Function code: 1 byte
- Data: 0-252 bytes
- CRC: 2 bytes

Each byte of the frame is sent as 11 bits:
- 1 start bit
- 8 data bits
- 1 parity bit
- 1 stop bit

Modbus TCP

Modbus over TCP is a modern adaptation of Modbus. It is also a binary protocol, but transported over TCP. This adaptation eliminates the need for CRC due to inherent error detection in TCP. The rest of the frame is composed like Modbus RTU and the default port for Modbus TCP is 502.

3.2.2 wM-Bus

wM-Bus [8, part. 4] is a wireless version of the Meter-Bus (M-Bus) [8] protocol, used primarily in Europe for metering applications. It adheres to the ISO layer model [9] but implements only specific layers :

- Layer 1: Physical layer ([8, part. 2] for wired and [8, part. 4] for wireless)
- Layer 2: Data link layer ([8, part. 2] for wired and [8, part. 4] for wireless)
- Layer 7: Application layer ([8, part. 3])

The wireless specification has several modes of operation, to work on several frequency bands. The most common are:

Mode S

This mode [8, part. 4, p. 16] works on 868 MHz with 2-Level Frequency-Shift Keying (2FSK) modulation on a single channel. Meters send data many times a day to a stationary collector. The collector can be used in a power-saving function and is awakened up by the long heading of the frame. This mode has a one-directional (S1) or bidirectional (S2) sub-mode.

Mode T

This mode [8, part. 4, p. 19] works on 868 MHz with 2FSK modulation. The one-directional sub-mode T1 has a single-channel, but the bidirectional sub-mode T2 can use 2 channels. Meters are frequently sending data to a collector. This collector can be mobile.

Mode R2

This mode [8, part. 4, p. 24] works on 868 MHz with 2FSK modulation. This mode is bidirectional and has 10 channels. This mode can use frequency hopping for a duty cycle higher than with other modes.

Mode C

This mode [8, part. 4, p. 27] works on 868 MHz with 2FSK modulation. The one-directional sub-mode C1 has a single channel and the bidirectional sub-mode C2 has 2 channels with 2 different bandwidths. It can be used for stationary or mobile collectors.

Mode N

This mode [8, part. 4, p. 30] works on 169 MHz with 4-Level Gaussian Frequency-Shift Keying (4GFSK) modulation. It can be one- or bidirectional and has 13 channels. This mode is used for long-range communication with a stationary collector.

Mode F

This mode [8, part. 4, p. 35] work on 433 MHz with 2FSK modulation. This mode has only a bidirectional sub-mode. It is used for long-range communication with a stationary or mobile collector.

3.3 Simulation environments

As this thesis is part of the preparation of a new laboratory, the simulation environment must extend beyond abstract communication. The objective is to have a real physical controller interfaced with a simulated process. This simulated process remains necessary because a fully physical setup is prohibitively expensive and far less flexible than a simulated one.

3.3.1 Factory I/O

Factory I/O, developed by Real Games, is a realistic simulation software designed to emulate a factory environment. It allows for the creation of custom scenes, providing flexibility for various industrial scenarios. However, the software comes with an expensive licence, which must be considered when selecting the simulation environment. Factory I/O could also be beneficial for the Power & Control specialization. This software can interface with modbus over TCP, but an additional third-party software is required to implement a security layer. While Factory I/O is only available on Windows, it operates effectively on Linux using Wine.

Figure 6: Factory I/O palletizer scene

Scenario idea

In this scenario, a PLC could control the palletizer scene (). A wireless sensor could indicate the presence of a truck to be loaded. The wireless replay attack could target this sensor. The DoS attack could be executed on the same sensor. The MitM attack could be conducted on the Modbus/TCP communication between the PLC and the palletizer, with the objective of gaining control over the clamp.

3.3.2 Home I/O

Home I/O is also developed by Real Games and simulates a House () equipped with extensive home automation features. This software is available at a lower organizational licence cost and could be of interest to ETE students. Home I/O offers a REST API for interfacing with all sensors and actuators. Similar to Factory I/O, it runs well on Linux using Wine.

Figure 7: Home I/O scene

Scenario idea

In this scenario, a PLC manages the alarm and access systems, including the main door and garage. The garage door can be opened using a wireless remote, which could be the target of a wireless replay attack. An external presence detector on the main door could be used for the DoS attack. MitM attacks could be executed on the Modbus/TCP communication between the PLC and the alarm system, aiming to deactivate the alarm system.

3.3.3 Minecraft

Another suitable approach would be to use the Electrical Age world in Minecraft () which was previously featured in the Telecommunication course. The goal of this lab was to control a factory and energy system to maximize production. Continuing this laboratory could provide valuable opportunities for students to explore and secure communication systems. In Minecraft, security can be implemented using the Open Computers mod or by creating an extension mod for Modbus over Transport Layer Security (TLS), which might be simpler than using Lua with Open Computers.

Figure 8: Minecraft Electrical Age scene

Scenario idea

The scenario involves reusing the PLC from the previous lab, which controls various operations. A physical Human-Machine Interface (HMI) could be constructed to manage the factory and coal production, similar to the HTML interface used in the previous lab. A wind wireless sensor could be added to the setup. Wireless replay and DoS attacks could target this sensor. MitM attacks could be executed on the Modbus/TCP communication between the PLC and the factory, with the goal of gaining control over the factory's operations.

3.4 Conclusion

This section presented various attacks that can be performed on OT systems. The communication media discussed during the preliminary phase of this work were also outlined. Additionally, potential simulation environments for laboratory use were evaluated.

Based on this information, M.Rieder and M. Clausen have decided on the simulation environment. The chosen platform is Home I/O, as it could benefit ETE students.

The planned attacks include a replay attack on a wireless control or sensor, a DoS attack on an external sensor with valid data (overloading the controller rather than the communication medium) and a MitM attack on the Modbus/TCP communication. The MitM attack will be conducted in two phases. The first phase will involve an unencrypted Modbus/TCP communication while the second phase will involve encrypted Modbus/TCP communication with a symmetrical key exchanged by Diffie-Hellman.

Wired communication will be carried out using Modbus, a widely used protocol in OT systems, which aligns with the brief for this thesis.

For the replay attack on a wireless control or sensor, the idea is to use the Flipper Zero to record and replay a message. This attack will be performed only on the physical layer. The Flipper Zero can only execute such an attack on basic wireless protocols, as it cannot perform a replay attack on frequency hopping protocols. The wireless protocol must employ On-Off Keying (OOK), Amplitude-Shift Keying (ASK) (with 270 or 650 kHz Bandwidth) or 2FSK modulation. Consequently, protocols like Zigbee or DigiMesh are not suitable for this attack and were not explored in depth.

wM-Bus is a single canal 2-Level Gaussian Frequency-Shift Keying (2-GFSK) protocol and is well-suited for replay attacks using the Flipper Zero. Given that typical wM-Bus T-mode application include electricity or water meters, incorporating such meters into the simulation would be relevant with a replay attack targeting these devices.
If the wM-Bus implementation proves too challenging, a contingency plan involves using a simple OOK modulation at 433 MHz.

4 Man in the Middle Scenario

The Man-in-the-Middle (MitM) scenario focuses on intercepting, modifying, and forwarding packets to gain control over a Modbus/TCP installation. This protocol, commonly used in industrial settings, was selected for this thesis because it is widely adopted and fulfils the requirement to demonstrate an attack on it. The MitM attack was chosen due to its prevalence and potential for significant impact. It is a comprehensive attack that encompasses several other techniques, such as sniffing and spoofing through ARP poisoning. This scenario assumes that the attacker has already gained access to the network, enabling them to intercept and manipulate the data packets.

Contents

4.1 Simulation Environment 15
4.2 Requirements 16
4.2.1 Tools 17
4.2.2 Closer look on Modbus 17
4.3 Attack on Modbus/TCP 18
4.3.1 Closer look on TCP 18
4.3.2 Modify packet on the fly 18
4.3.3 Summary 19
4.4 Implement Modbus/TLS 20
4.4.1 Closer look on TLS 20
4.5 Attack on Modbus/TLS 20
4.6 Conclusion 21

4.1 Simulation Environment

Home I/O is a smart home simulation platform that allows interaction with a wide range of sensors and actuators, as detailed in Section 3.3.2. For the Man-in-the-Middle attack scenario, the entrance hall within this simulation is particularly relevant. As shown on Figure 9 the entrance hall has two doors with their sensor, one motion sensor and the house alarm.

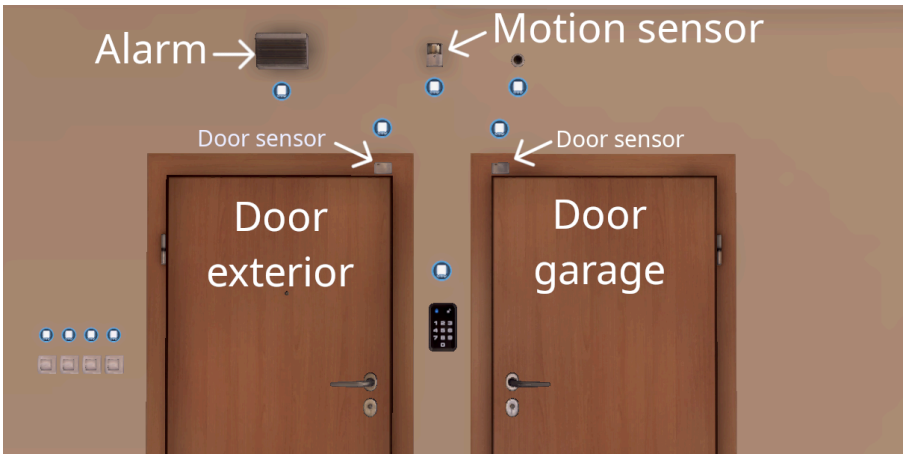


Figure 9: Home I/O scene

Since Home I/O does not natively support Modbus communication, a custom Go software was developed by Michael Clausen to convert Modbus requests into the REST requests required by the simulation. This software simulates a Modbus server for each room. In this thesis, the architecture consists of one computer running the simulation and another acting as the house controller, as shown in Figure 10. The controller, functioning as a Modbus client, is implemented using another Go program that could be deployed on a Wago CC100 PLC in a real-world scenario. For the purposes of this thesis, the controller is intentionally kept simple.

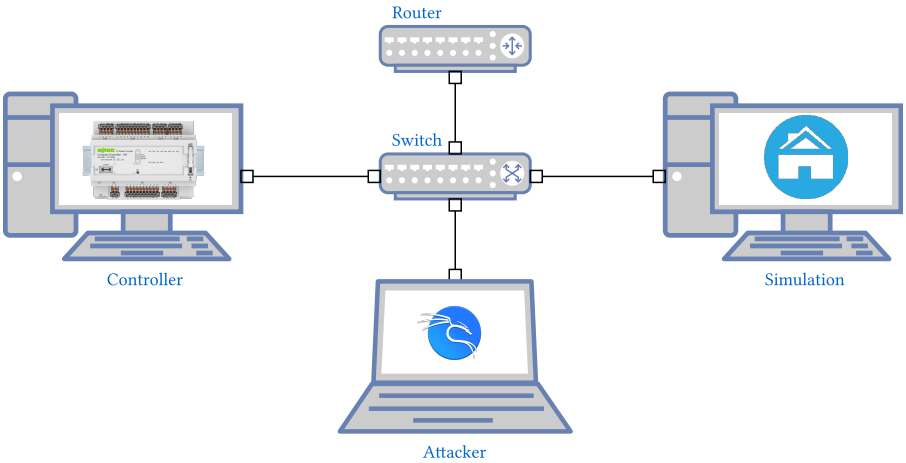


Figure 10: MitM scenario implementation

In this demonstration, the controller sends requests to check the status of both door sensors and the motion sensor. If either door is open or motion is detected, the controller

sends a request to the alarm to activate it. If no activity is detected, the controller sends a request to deactivate the alarm.

The controller, the simulation, and an attacker are all connected to the same network via a switch. This thesis was made under a Kali Linux (Icon 1) laptop, although it is performed on any computer with the appropriate tools.

4.2 Requirements

This scenario centres on a MitM attack, where the attacker must position themselves between the communication parties, ensuring that all packets pass through their system. Since Modbus/TCP is an IP-based protocol, the attacker can utilize a powerful tool called Ettercap (Icon 2). Ettercap simplifies the execution of an ARP poisoning attack.

An Address Resolution Protocol (ARP) poisoning attack involves sending fraudulentARP messages across the network, tricking the target into associating the IP address of another party with the attackers Media Access Control (MAC) address, as illustrated in Figure 11. As a result, the target's packets are sent to the attacker, who can them manipulate these packets before forwarding them (or not) to the intended recipient. This attack occurs at layer 3 of the OSI model [10].

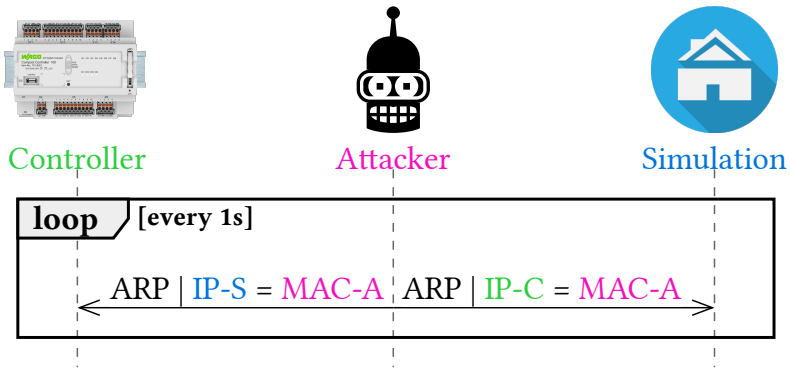


Figure 11: ARP poisoning

To use Ettercap, the attacker need to know the IP address of the controller and the Home I/O simulation. This information can be obtained by sniffing the network using Ettercap itself or other tools such as **`Wireshark`**, **`nmap`**, or **`hping3`**. Once the IP addresses are identified, the attacker can initiate the ARP poisoning attack with the following command:

```
ettercap -T -i eth0 -M arp /IP_CONTROLLER// /IP_HOMEIO//
```

Listing 1: Start an ARP poisoning with Ettercap

This command, shown in Listing 1 start Ettercap in text mode (**`-T`**) on the **`eth0`** network interface (**`-i eth0`**) and perform an ARP poisoning attack (**`-M arp`**). The IP addresses of the controller and the Home I/O simulation are specified by **`/IP_CONTROLLER//`** and **`/IP_HOMEIO//`**. An graphical interface is also available but have to be installed as an extra package.

To execute this attack, **`iptables`** (Icon 3) will be used to redirect the packet to a python (Icon 4) script that modifies them in real-time. This script employs the scapy (Icon 5) li-

brary, a powerful tool for crafting or decoding packets from a wide range of protocols. The second part of this attack scenario (see on Section 4.4) will use a go (Icon 6) script.

4.2.1 Tools

Here are all the tools that are used for this scenario :


Icon 1
Kali Linux


Icon 2
Ettercap


Icon 3
Iptables


Icon 4
Python


Icon 5
Scapy


Icon 6
Golang

4.2.2 Closer look on Modbus

As discussed on Section 3.2.1, Modbus is a straightforward protocol that was initially developed for serial communication via RTU but is now commonly implemented over TCP/IP. This protocol operates in request-response mode, as illustrated in Figure 12. In this setup, the controller sends a request to the Home I/O to retrieve a value or alter an output. The Home I/O then replies with the requested value or an acknowledgment message, after which the controller can issue another request or close the connection.

This request-response mechanism means that the attacker must capture traffic in both directions to modify packets in real-time effectively. The attacker needs to know the nature of the request to determine if the response needs alteration. Without this knowledge, it would be impossible to decide if a response needs to be modified.

In this thesis simulation, the controller is tasked with checking the status of two door sensors and a motion detector, all of which are connected to the same Modbus slave, the **Entrance Hall** (UnitID=5). The alarm system is represented as a coil on the same slave. A summary of these registers can be found in Table 1.

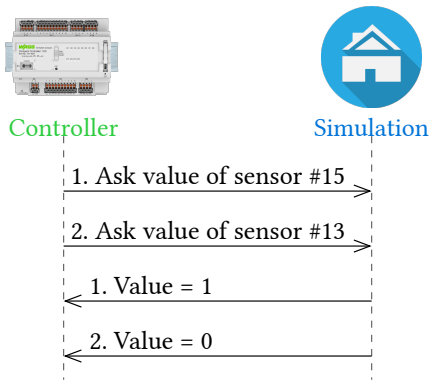


Figure 12: Home IO details

Sensor / Actuator	Unit ID	Fonction	Address
Door exterior	5	Discrete Inputs	13
Door garage	5	Discrete Inputs	14
Motion detector	5	Discrete Inputs	15
Alarm	5	Coils	5

Table 1: Modbus Registers

4.3 Attack on Modbus/TCP

To modify a packet during a Modbus/TCP attack, the first step is to establish a MitM position. This is achieved using ARP poisoning as described in Section 4.2. Once the attacker intercepts all the packets, they need to redirect them to a Python (Icon 4) script for real-time modification. This redirection can be accomplished by configuring iptables (Icon 3) to add rules to the attacker's firewall. The idea is to place all the packets into a queue, enabling the Python script to retrieve and analyse them sequentially. Listing 2 demonstrates how to use iptables (Icon 3) to enqueue all packets destined for the **`192.168.0.0/16`** subnet into queue **`1`**.

```
iptables -I OUTPUT -d 192.168.0.0/16 -j NFQUEUE --queue-num 1
```

Listing 2: Put packets on queue 1 with Iptables

4.3.1 Closer look on TCP

Transmission Control Protocol (TCP) operates at layer 4 of the OSI model [10] and is responsible for establishing a reliable connection between two devices. In this thesis, the primary concern is TCP's mechanism for ensuring packet integrity via checksums. The checksum ensures that the packet has not been corrupted during transmission; if the checksum does not match, the packet is discarded. This aspect is crucial for the attacker, who must modify the packet on the fly. If the modified packet has an incorrect checksum, it will be discarded, causing the attack to fail. An acknowledgment (ACK) is sent when the packet is received correctly, as illustrated in Figure 13. Within the TCP segment, the Modbus packet is encapsulated as shown in Table 2.

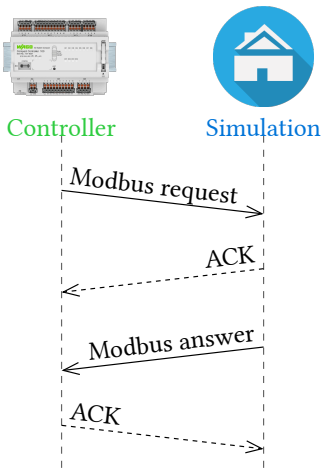


Figure 13: TCP exchange

0	1	2	3	4	5	6	7	8	9	10	...
Transaction ID		Protocol identifier		Length		Unit ID	Func. code	Reference number		Modbus data	

Table 2: Modbus TCP packet structure

4.3.2 Modify packet on the fly

The packet modifications are carried out using a Python (Icon 4) script, leveraging the scapy (Icon 5) library. Scapy is particularly useful for on-the-fly modification. With Scapy it is straightforward to dissect the different layers of a packet. To extract the IP layer, one can use the command **`scapy_packet = IP(pck.get_payload())`**. To check if the packet is TCP and retrieve its payload, the command **`payload = bytes(scapy_packet.payload.payload)`** can be used.

This binary payload contains the Modbus message, which can be inspected and altered as needed. If the destination port is 1502, the packet is from the controller heading to the

server. In this case, the attacker should check if the request concerns a door sensor or the motion sensor and then save the transaction ID of this request.

If the source port is 1502, the packet is from the server and is destined for the controller. The attacker should then verify whether the response corresponds to a previously saved transaction ID and, if so, modify the response as necessary.

4.3.3 Summary

The entire process is summarized in Figure 14.

(1) Initially, under normal conditions, the controller sends a request to the server, the server responds, and the controller triggers the alarm if necessary.

(2) However, when the attacker initiates an ARP poisoning attack, all packets are routed through the attacker.

(3) This allows the attacker to perform a MitM attack and modify the packets in real-time. In this scenario, the controller sends a request to the server, the attacker intercepts and alters the response, and the controller proceeds as if the modified response were legitimate.

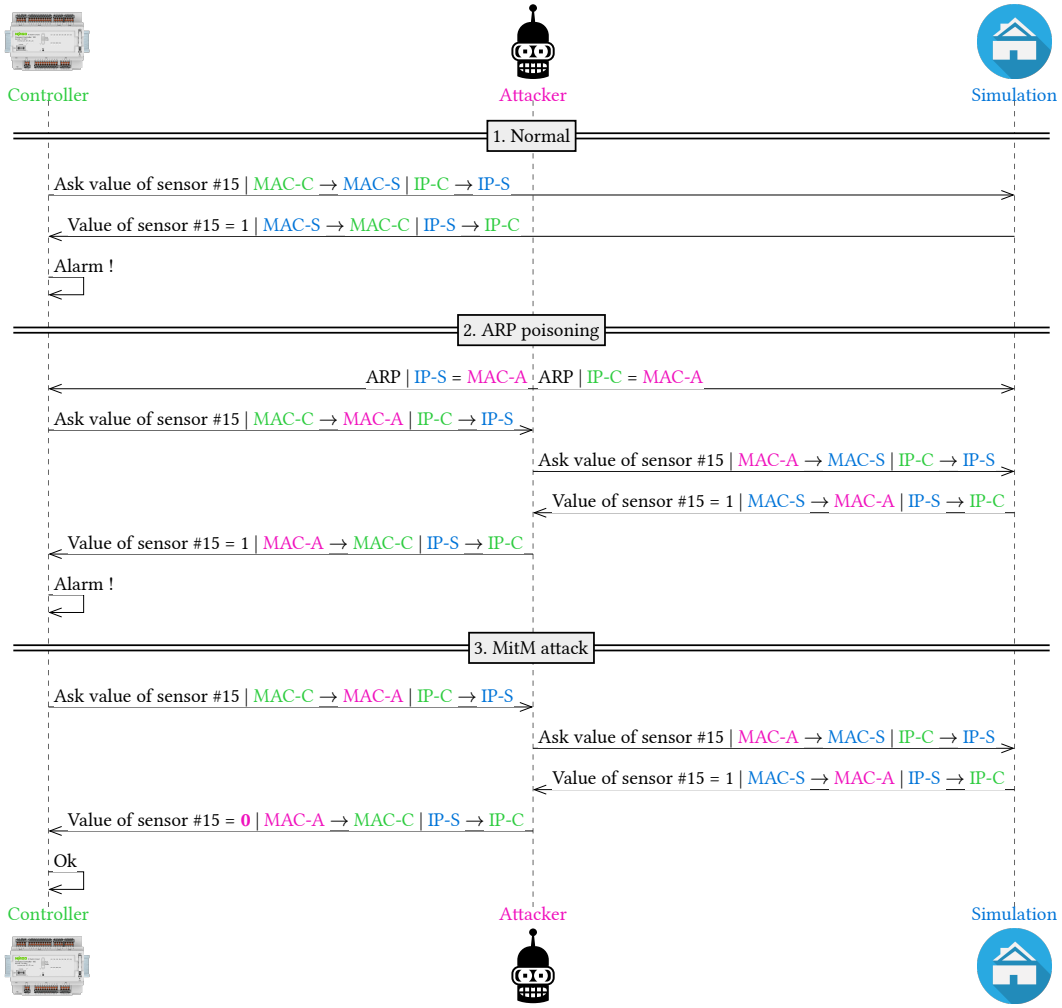


Figure 14: Modbus/TCP attack scenario

4.4 Implement Modbus/TLS

While clear communication can work, it falls short when it comes to security. To safeguard Modbus/TCP communication, it is essential to add an encryption layer. Modbus over TLS is a secure version of Modbus/TCP, encrypting the communication between the controller and the server. With Modbus/TLS in place, the attacker is unable to intercept and modify the packets in real-time, as they are protected by encryption.

4.4.1 Closer look on TLS

Transport Layer Security (TLS) is a cryptographic protocol that operates over TCP, primarily associated with layers 5 and 6 of the OSI model [10]. While it technically spans both these layers, establishing an encrypted session (layer 5) and handling the encryption itself (layer 6). TLSis commonly considered a layer 6 protocol.

TLS provides secure communication between two parties, typically a client and a server. The process begins with a handshake, as illustrated in Figure 15, where the client and server exchange a series of messages to establish a secure connection. This handshake involves the exchange of random numbers, a certificate, and a seed. The Diffie-Hellman key exchange is used to generate a master key, a symmetric key that will be used to encrypt all communications within that session.

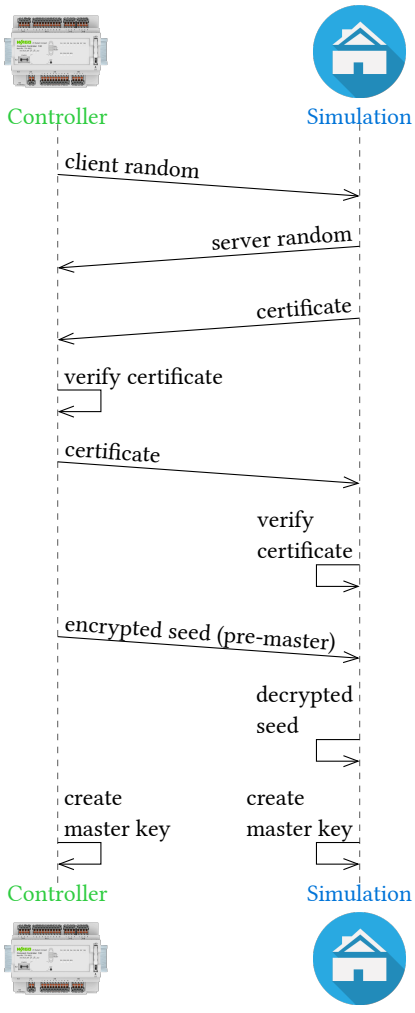


Figure 15: TLS handshake

During the handshake, the client, and server also exchange X.509 certificates to verify each other's identity. An X.509 certificate contains information about the certificate's owner, such as their name, expiration date, and address for TLS communication. The certificate also includes the owner's public key, which is used to authenticate their identity. X.509 certificates rely on a chain of trust provided by Certificate Authority (CA). A CA is a trusted third party that verifies the identity of the certificate's owner and signs the certificate to validate it. The client and server can verify the validity of each other's certificates by checking the CA's signature. If the signature is valid, the certificate is considered trustworthy, forming the basis of the TLS handshake.

4.5 Attack on Modbus/TLS

This thesis demonstrated how easily Modbus/TCP packets can be intercepted and modified. However, when Modbus/TLS is employed, the packets are encrypted, making it

seemingly impossible to perform a MitM attack to take control of the system. This is true, but only if every step of the TLS implementation has been executed correctly.

The thesis also reveals that control can still be compromised if certificates are not properly verified. In test or debugging environments, it's common for certificates to be unsigned by a CA, rendering them invalid. As a result, certificate verification is often bypassed in such environments. However, it is crucial to ensure that certificates are signed and verified in a production environment. In the OT world, it may be advisable for a company to maintain its own internal CA to sign all certificates. This makes it easier to install the CA on all company devices and ensure proper certificate verification.

To perform a MitM attack on a Modbus/TLS installation that does not check certificates, the attacker must first establish a MitM position. This can be done using the same ARP poisoning attack described in Section 4.2. During this thesis, considerable time was spent trying to modify random and certificates on the fly. This approach proved difficult because the TLS handshake (Figure 15) must be fully implemented, and there are no tools specifically designed to modify only the TLS layer in real-time. While many tools exist for performing MitM attacks in the IT world, such as **`Burp suite`**, **`mitmproxy`**, or **`bettercap`**, they are typically focused on the Hypertext Transfer Protocol Secure (HTTPS) protocol.

A more effective approach would be to handle the entire connection rather than attempting to modify packets on the fly. The attacker can redirect the TLS traffic from both targets to their own server, where the traffic can be decrypted. To redirect the traffic, **`iptables`** (Icon 3) can be used, as demonstrated in Listing 3.

```
iptables -t nat -A PREROUTING -p tcp --dport 5802 -j REDIRECT --to-port 5803
```

Listing 3: Redirect traffic to another port with Iptables

This command redirects all TCP traffic destined for port **`5802`** (used for TLS communication between the controller and Home I/O) to port **`5803`** (the port where the attacker's server is running).

The attacker can then set up their own Modbus/TLS server to decrypt the traffic and forward it to the Home I/O simulation. If desired, the attacker can modify the response before forwarding it back to the controller. It is easy to see a trace of such an attack because the attacker has to use a dummy certificate to create the symmetric encryption key. This certificate can be seen with tool like **`Wireshark`**

4.6 Conclusion

In summary, this chapter has outlined the simulation environment, detailed the tools required for this scenario, and demonstrated how to implement a MitM attack on both Modbus/TCP and Modbus/TLS communications.

The thesis focused on a simple system composed of two door sensors and a motion sensor controlling an alarm. The attacker's primary goal was to gain control over this system to open the doors without triggering the alarm. The findings illustrate how easily this can be achieved when no security measures are in place. The attacker only needs access to the local network, which could potentially be obtained by compromising an external sensor.

Additionally, this thesis has shown how implementing Modbus/TLS can secure the system and highlighted the critical importance of verifying certificates to ensure robust protection.

5 Replay Scenario

TODO
- introd
- pourquoi
- sans fils
- comment
- flipper

The replay scenario involves intercepting and resending a message on a wireless connection to trigger the same effect as the original message, like for example a garage door opening remote.

Contents

5.1 Simulation Environnement 24
5.2 Requirements 24
5.2.1 Closer look on Wireless M-Bus 24
5.3 Attack on Wireless M-Bus 24
5.3.1 Flipper Zero 24
5.3.2 Modulation FSK vs GFSK 24
5.4 Attack on basic 433 MHz transceiver 24
5.5 Security in wireless brodcast isolted devices 24
5.5.1 Closer look on rolling code 24
5.5.2 Closer look on signature 24

5.1 Simulation Environnement

TODO

5.2 Requirements

TODO

5.2.1 Closer look on Wireless M-Bus

TODO

5.3 Attack on Wireless M-Bus

TODO

5.3.1 Flipper Zero

TODO

5.3.2 Modulation FSK vs GFSK

TODO

5.4 Attack on basic 433 MHz transceiver

TODO

5.5 Security in wireless brodcast isolted devices

TODO

5.5.1 Closer look on rolling code

TODO

5.5.2 Closer look on signature

TODO

6 Conclusion

TODO
- résumé
- résultats
- usage pour un labo
- difficultés

6.1 Project summary

TODO

6.2 Comparison with the initial objectives

TODO

6.3 Encountered difficulties

TODO

6.4 Future perspectives

TODO

7 Glossary

***ARP* – Address Resolution Protocol**: The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. 6, 14, 16, 18, 19, 21

Flipper Zero: Flipper Zero is a portable multi-tool for pentesters and geeks in a toy-like body. 5, 13

***HMI* – Human-Machine Interface**: A human-machine interface (HMI) is a user interface or dashboard that connects a person to a machine, system, or device. 13

***IT* – Information Technology**: Information Technology (IT) is the use of computers to store, retrieve, transmit, and manipulate data or information. 7, 9, 21

***MAC* – Media Access Control**: A media access control address (MAC address) is a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment. 16

***OT* – Operational Technology**: Operational Technology (OT) is hardware and software that detects or causes a change through the direct monitoring and/or control of physical devices, processes, and events in the enterprise. 6, 7, 8, 9, 13, 21

***PLC* – Programmable Logic Controller**: A programmable logic controller (PLC) is an industrial digital computer that has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, robotic devices, or any activity that requires high-reliability control and ease of programming. 7, 11, 12, 13

***SDG* – Sustainable Development Goal** 4

7.1 Attacks

***DDoS* – Distributed Denial of Service**: A distributed DoS is basically the same as a DoS attack, but the attack comes from multiple sources. 7

***DoS* – Denial of Service**: A denial-of-service (DoS) attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. 6, 7, 11, 12, 13, 26

***MitM* – Man in the Middle**: A Man in the Middle (MitM) attack is a form of eavesdropping in which communication between two users is monitored and modified by an unauthorized party. 6, 8, 9, 11, 12, 13, 14, 15, 16, 18, 19, 21

7.2 Communications

***2FSK* – 2-Level Frequency-Shift Keying**: 2-Level Frequency-Shift Keying (2-FSK) is a form of Frequency-Shift Keying (FSK) modulation that uses two levels of frequency to encode digital data. 10, 13

***2-GFSK* – 2-Level Gaussian Frequency-Shift Keying**: 2-Level Gaussian Frequency-Shift Keying (2-GFSK) is a form of FSK modulation that uses two levels of Gaussian filtering to encode digital data. 13

***4GFSK* – 4-Level Gaussian Frequency-Shift Keying**: 4-Level Gaussian Frequency-Shift Keying (4-GFSK) is a form of FSK modulation that uses four levels of Gaussian filtering to encode digital data. 10

***ASK* – Amplitude-Shift Keying**: Amplitude-shift keying (ASK) is a form of modulation in which the amplitude of a carrier wave is varied proportionally to that of a modulating signal. 13, 27

***CRC* – Cyclic Redundancy Check** 9

***FSK* – Frequency-Shift Keying**: Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier wave. 26, 27

***HTTPS* – Hypertext Transfer Protocol Secure**: Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). HTTP is an application protocol for distributed, collaborative, hypermedia information systems. HTTPS is used for secure communication over a computer network and is widely used on the Internet. 21

***IP* – Internet Protocol** 9, 16, 17, 18

***M-Bus* – Meter-Bus** 10

***OOK* – On-Off Keying**: On-Off Keying (OOK) denotes the simplest form of ASK modulation that represents digital data as the presence or absence of a carrier wave. 13

***RTU* – Remote Terminal Unit** 9, 10, 17

***TCP* – Transmission Control Protocol** 9, 10, 11, 12, 13, 14, 16, 17, 18, 20, 21

***wM-Bus* – Wireless M-Bus** 5, 5, 10, 13

7.3 Cryptography

***CA* – Certificate Authority**: A certificate authority (CA) is an entity that issues digital certificates. 20, 21

Diffie-Hellman: Diffie-Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel. 8, 13, 20

***TLS* – Transport Layer Security**: Transport Layer Security (TLS) is a cryptographic protocol that provides communications security over a computer network. 12, 20, 21, 22

X.509: X.509 is a standard that defines the format of public key certificates. 9, 20

Bibliography

[1] "Sniffing Attack." Jun. 04, 2023. Accessed: Jun. 24, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sniffing_attack&oldid=1158437808

[2] "Denial-of-Service Attack." May 27, 2024. Accessed: May 28, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Denial-of-service_attack&oldid=1225934024#DDoS_extortion

[3] "Lunar - Missions - Apollo 11 Mission." Accessed: Jun. 24, 2024. [Online]. Available: https://www.lpi.usra.edu/lunar/missions/apollo/apollo_11/

[4] "Replay Attack." Jan. 04, 2024. Accessed: May 28, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Replay_attack&oldid=1193515559

[5] "Man-in-the-Middle Attack." Jun. 20, 2024. Accessed: Jun. 25, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Man-in-the-middle_attack&oldid=1230046662

[6] Fahmida Y. Rashid, "The Old Ways Aren't Working: Let's Rethink OT Security," Nov. 2021, [Online]. Available: https://www.darkreading.com/cyber-risk/the-old-ways-aren-t-working-let-s-rethink-ot-security

[7] "Modbus." May 23, 2024. Accessed: May 27, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Modbus&oldid=1225269451

[8] "EN 13757 - Communication Systems for Meters." Swiss Association for Standardization (SNV).

[9] "ISO/IEC 7498-1:1994 - Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model." Swiss Association for Standardization (SNV).

[10] "ISO/IEC 7498-1:1994." Accessed: May 29, 2024. [Online]. Available: https://connect.snv.ch/en/isoiec-7498-1-1994