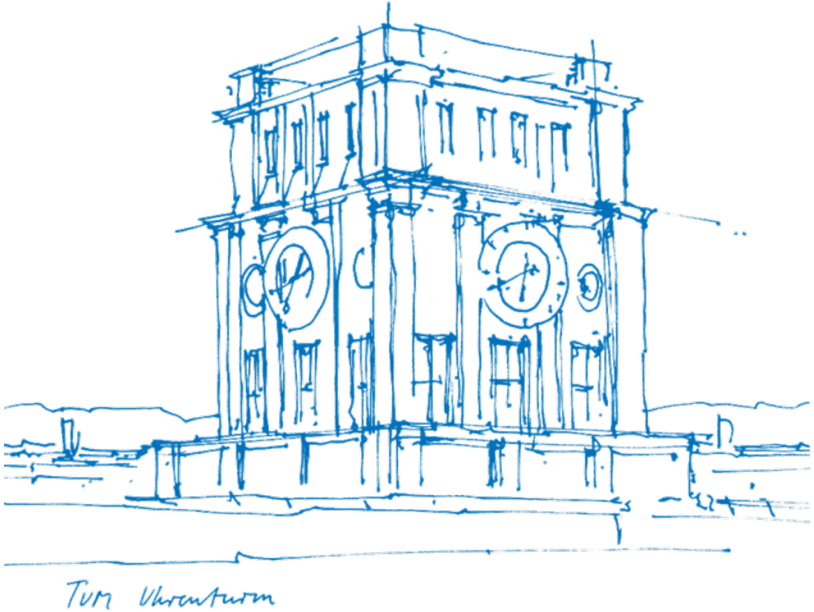
Development of a Testbed to Demonstrate Attacks on Emulated PLC Networks

Victor Embacher
Department of Informatics, Technical University of Munich (TUM)
Fraunhofer Institute for Applied and Integrated Security AISEC

06.05.2021



Outline

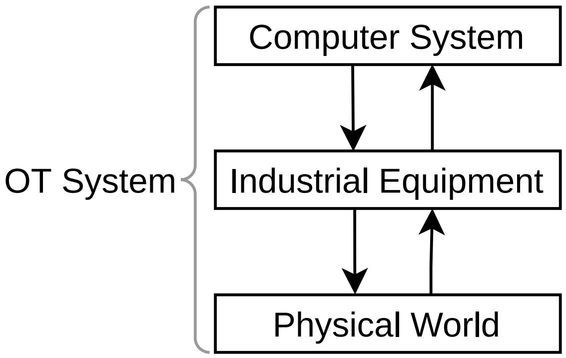
Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Introduction - Defining IT and OT

Information technology (IT): corporate systems that primarily process information

Operational Technology (OT):

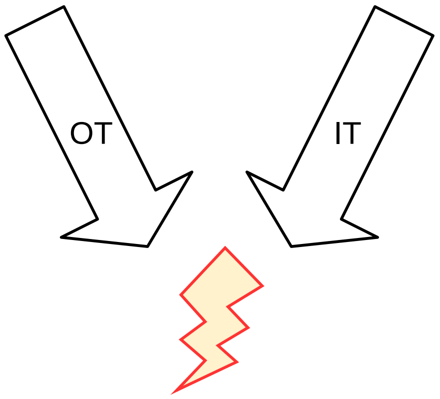
- directly interact with the physical world ⇒ **cyber-physical systems**
- control industrial components
- **industrial control systems (ICS)** are a common type of OT network. (more details later)



Introduction - Convergence of IT and OT

- IT and OT have become more connected to each other
  ⇒ **control networks previously were isolated**
  ⇒ but **now** are **connected to corporate networks or the Internet**
- IT hardware and software is now more frequent in OT systems
  ⇒ systems are now vulnerable to "general-purpose" malware
- previously **unaddressed problems become more apparent**:
  - insecure protocols
  - weak authentication
  - lack of attention towards network security

⇒ **OT networks receive more attention from attackers** . . .
. . . but from security researchers as well.



Growing Pains

Introduction - Motivation and Objectives

- ICS's (and other OTs) are essential to the functioning of society ⇒ **critical infrastructure**
- Pre-existing testing environments commonly do not address programable logic controllers (PLCs) in-depth

⇒ **Observation:** we require a **safe and realistic testbed** that . . .

- focuses on PLCs
- **can educate both IT and OT personnel** on known issues or attacks
- **enables security research**

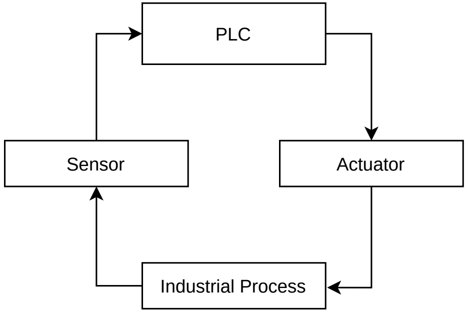
Background - Industrial Control Systems

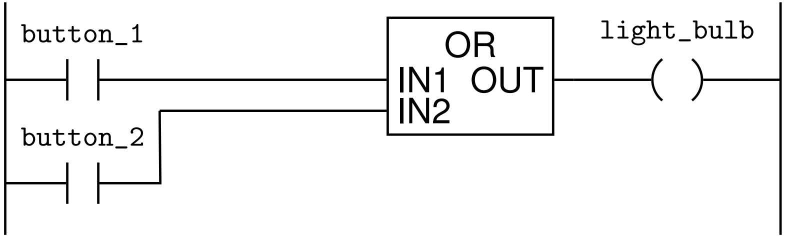
- ICS is a generic term for a wide array of **cyber-physical systems**.
- used to automate and control processes in industry and critical infrastructure.
- different ways to structure ICS networks:
  - **Distributed Control Systems (DCS)**: decentralized, local and a flat hierarchy
    $\Rightarrow$ e.g. **power plant**
  - **Supervisory Control and Data Acquisition (SCADA) Systems:** centralized, multiple regions, tall hierarchy
    $\Rightarrow$ e.g. **power grid**
  - **single PLC** controlling an industrial process

Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Background - Programable Logic Controllers

- devices used to **control processes**, used in **both SCADA systems and DCS's**
- designed for **rough environments** (low/high temperatures, vibration, etc.)
- **Run a loop:**
  1. read inputs (usually sensors, e.g. a thermometer)
  2. run PLC program
  3. write instructions to outputs that affect the physical world (actuators, e.g. servo)
- **PLC programs can be written in five languages** (standardized in IEC 61131-3)
  ⇒ we are only using Ladder Diagrams (LD)

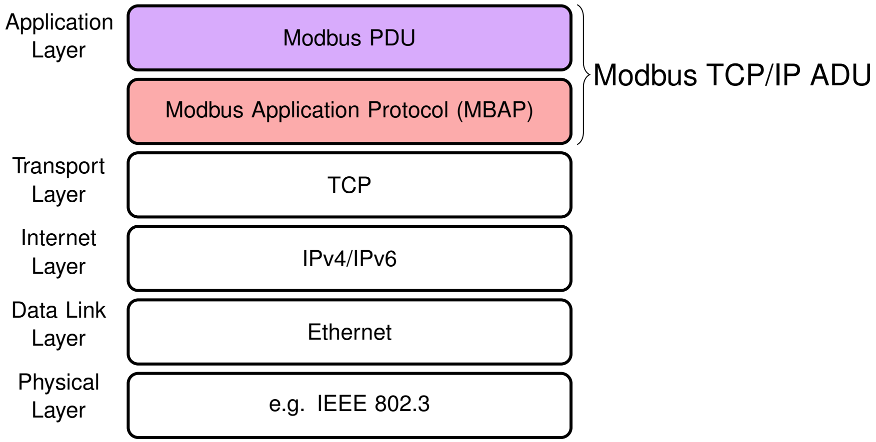




Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Background - Modbus Protocol

- commonly used in ICS networks, especially with PLCs to communicate with other ICS components.
- open protocol, standardized by the Modbus Organization.
- **originally** used asynchronous **serial** connections, **evolved to support TCP/IP**
- client-server protocol with request-response pairs.



Modbus TCP/IP in the five-layer Internet model

Background - ICS and PLC Attacks

ICS attacks:

- ICS's are also vulnerable to many attacks known from IT
- many attacks are made possible by bad practices
  - protocols lacking encryption, authentication and integrity protection
  - passwords are insecure or hardcoded
  - legacy OS's

PLC attacks:

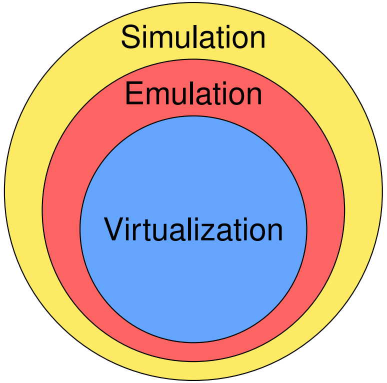
- modification of communicated sensor and actuator values
- upload a malicious PLC program

$\Rightarrow$ **only a small selection!**

Background - Simulation, Emulation and Virtualization

- **Simulation:** replication of a real-world system over time
- **Emulation:** replace hardware components with software
- **Virtualization:** decouple relationship between physical hardware and abstractions of it

Relationship of the three techniques:



Related Work (1)

I found the following relevant testbed surveys . . .

- *Holm et al.* (2015)
  - analyzes requirements posed to ICS testbeds and how they are adhered to ⇒ affected conceptualization
  - note that many testbeds do not address the requirement “fidelity”
  - **most important ICS testbed related survey**
- *Qassim et al.* (2017) and *Geng et al.* (2019)
  - both analyze the advantages and disadvantages of different realization techniques
  - assess hybrid techniques and virtualization as best ⇒ should be utilized whenever possible

Related Work (2)

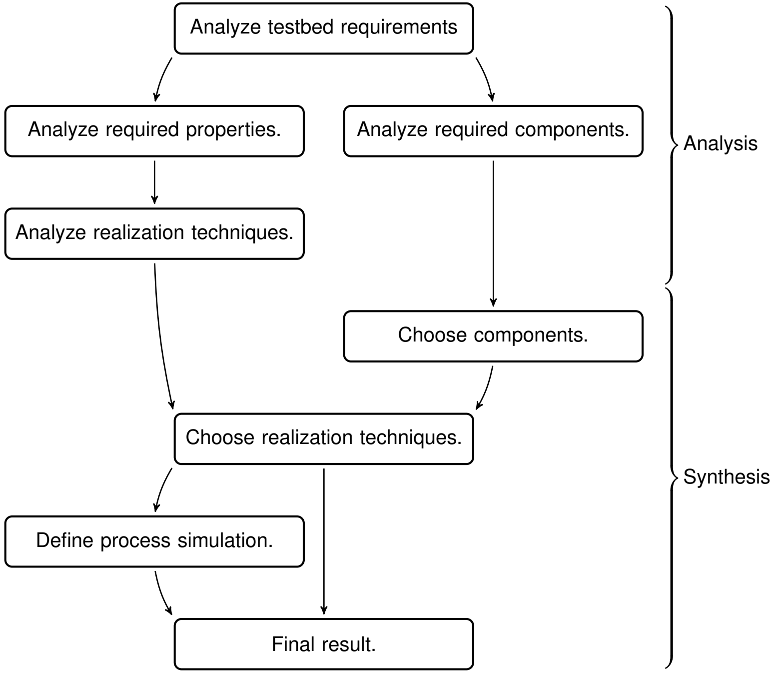
. . . and the following more concrete papers.

- *Formby et al.* (2018) developed an ICS testbed
  - does not include all components we include
  - partially uses proprietary software
- *Alves et al.* (2018)
  - **general approach** to SCADA testbeds **instead** of a **specific testbed** that can be used by readers
  - discuss a number of case studies using their testbed approach

⇒ **this thesis introduces**:

- **an exhaustive, high fidelity testbed** with a largely automated setup,
- **that only uses free and open-source components**,
- and is **shown to be able to demonstrate attacks and find new vulnerabilities**.

Conceptualization - Steps



Conceptualization - Required Properties

The ICS testbed survey from **Holm et al. (2015)** poses four properties to such an environment:

1. **Fidelity**
   - accuracy of mirroring real systems
   - addressed either by replicating a real system or based on standards
2. **Repeatability**
   - results need to be verified $\Rightarrow$ repetition is one way
   - **tests should have consistent results**
3. **Measurement Accuracy**
   - measuring should not affect the outcome of tests
   - **Comparison:** observer effect from physics
4. **Safe Execution of Tests**
   - no harm to living beings, environment, or equipment
   - testbed should not be damaged

Conceptualization - Additional Properties and Goal Conflicts

Furthermore, we have to address the following properties:

- **real-time properties**: e.g. low networking delays
- **affordability/low cost**: reduce financial barriers, make it affordable for everyone
- **open-source** components
- **reduced manual set-up**, use automation

However, some goals partially **conflict**:

- highest fidelity vs. open-source
- highest fidelity vs. low cost

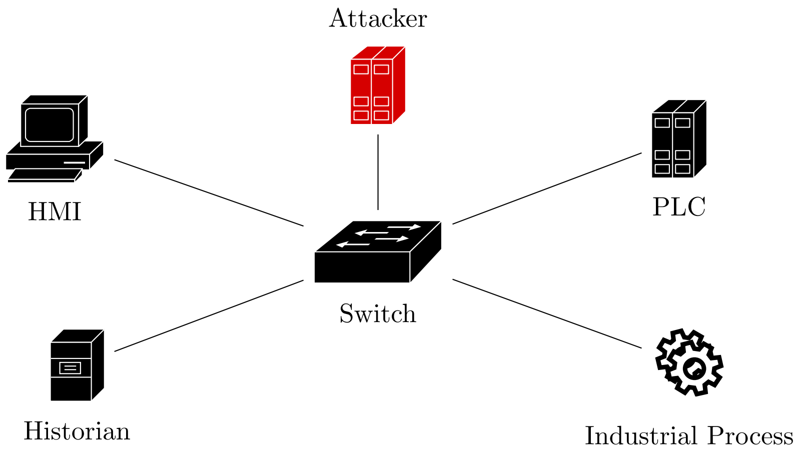
some **compliment** others:

- open-source $\Rightarrow$ low cost
- safe execution $\Rightarrow$ repeatability

Conceptualization - Required and Included Components

The ICS testbed survey from **Holm et al. (2015)** names a number of **required components**, we include:

- **control center:** HMI, data historian
- **field devices:** PLC
- **physical/industrial process:** an industrial process controlled by a single PLC
- **communication architecture:** switched ethernet network
- **used protocols:** (later slides)



Conceptualization - Realization Techniques: Pros and Cons

- **Physical replication** of the system
  - not an option for our entirely virtual testbed
  - High fidelity and accuracy, repeatability, but can be unsafe and expensive!
- **Simulation**
  - safe, scales well, affordable, but low fidelity, accuracy
  - **not the best option to research cybersecurity threats** $\Rightarrow$ does not use real hard or software
- **Emulation:**
  - **good to replace hardware components with software** (avoid cost and damage)
  - high fidelity, safety, repeatability, accuracy
  - however accuracy depends on the quality of the emulation
- **Virtualization:**
  - high fidelity, repeatability, accuracy, safety, lower cost, scales well
  - usually the best approach, but **not always possible**
- **Hybrid approach:**
  - choose best approach for each component $\Rightarrow$ best overall properties
  - however, can be expensive if physical hardware is used

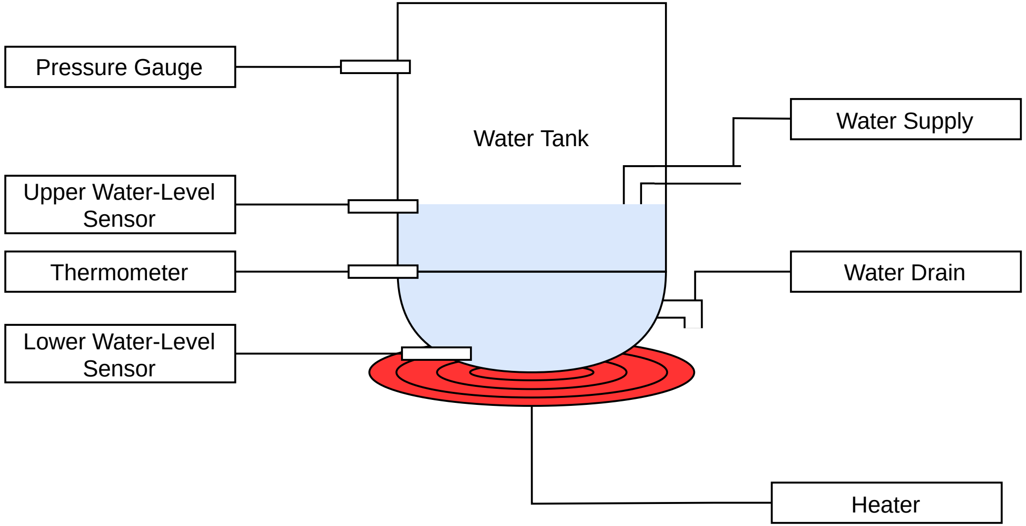
Conceptualization - Choosing the Correct Approach

- **HMI & data historian ⇒ real software + virtualization**
  - easy to virtualize
  - usually software running on x86 architecture + commodity OS (Windows, Linux, etc.)
- **PLC ⇒ emulation via software PLC**
  - simulation would be too inaccurate, limits ability to research cybersecurity of PLCs
  - we can use real PLC programs and protocols
- **industrial process ⇒ software simulation**
- **communication architecture ⇒ network virtualization**
  - use real network stacks and protocols ⇒ **examine real network based attacks**
  - **no major drawbacks**

⇒ **we use an hybrid approach**

Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

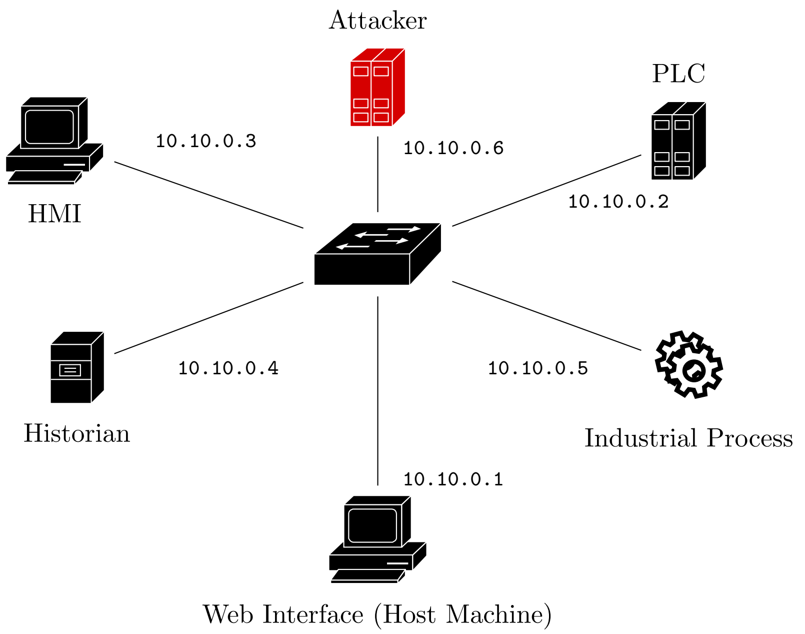
Conceptualization - Process Simulation Model



⇒ **behavior will become clear in the demo!**

Implementation - Virtualizing the Hosts and Network

- use **libvirt** as our virtualization management framework
  $\Rightarrow$ **provides virtualized networking and VMs**
- **hosts** run Ubuntu cloud images + cloud-init
- we automate the installation of dependencies and configuration as much as possible with libvirt and cloud-init
- communication between hosts uses the Modbus protocol



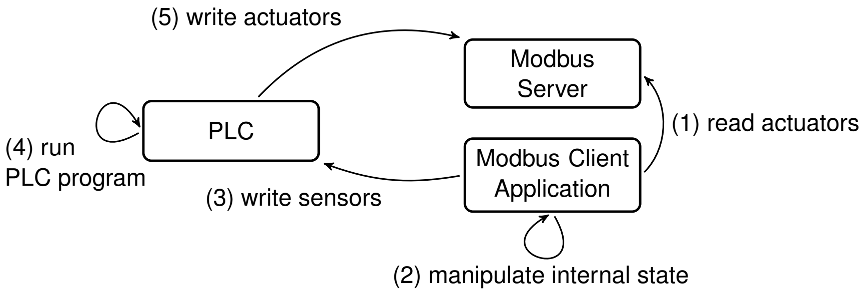
Implementation - Testbed ICS Components

- **PLC ⇒ OpenPLC**
  - only open-source PLC
  - supports all IEC 61131-3 programming languages
- **HMI ⇒ ScadaBR**
  - gathers data from the PLC and industrial process
  - can be used to build graphical views (control and monitor)
- **data historian ⇒ TICK stack**
  - based on **T**elegraf, **I**nfluxDB, **C**hronograf and **K**apcitor
  - a lot more powerful than just using the HMI for some limited record keeping
- **industrial process** ⇒ next slide!

Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Implementation - Simulator

- implements the simulation model we defined in the conceptualization part
- written in Python using the PyModbus library for Modbus communication
- consists of a Modbus server and a application using a Modbus client



Tech Demo

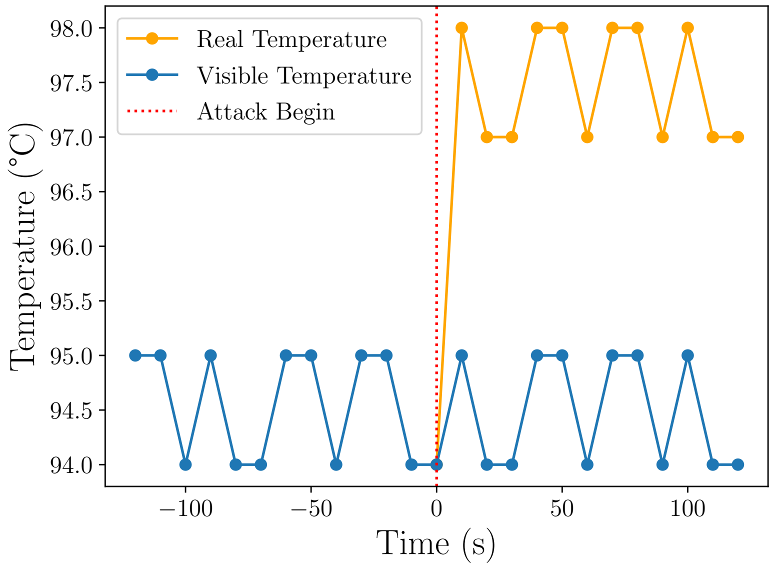
Attack Demonstration - Adversary

- **Before discussing attacks how could an adversary have gotten access?**
  - propagate from a corporate network to a field network ⇒ inadequate firewalls or DMZs
  - exploitation of a remote access point
- **What information does an attacker require to launch these attacks?**
  - all necessary information publicly avoidable (e.g Modbus specification)
  - no need for complicated reverse engineering tasks, that might require purchasing hardware

  ⇒ **required know-how is reasonable!**
- **Immediately before the attack:** attacker gains a MitM position via ARP cache poisoning

Attack Demonstration - Manipulating the Thermometer (1)

- **Background:** Modbus traffic is not secured at all
- **Idea:** alter sensor values communicated via Modbus
  - under report temperature
  - PLC receives false information and behaves accordingly
  - temperature reaches unsafe levels and system becomes damaged

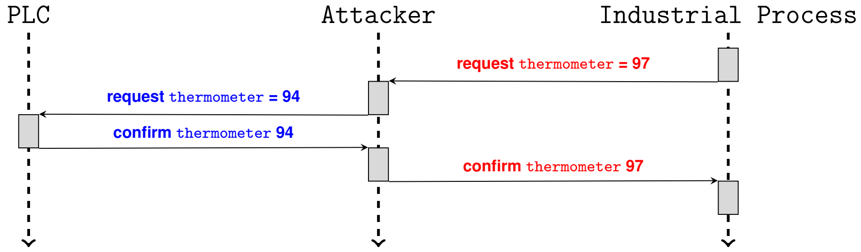


Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Attack Demonstration - Manipulating the Thermometer (2)

How does the attack work?

- we match specific write requests and their responses
- use a NetfilterQueue + ScaPy based application to modify packets in real-time
- both PLC and industrial process are oblivious to the manipulation
- **correct sensor values do not appear in any of the logs ⇒ stealthy**
- **attack principle can be applied to all other Modbus connections in our setup**

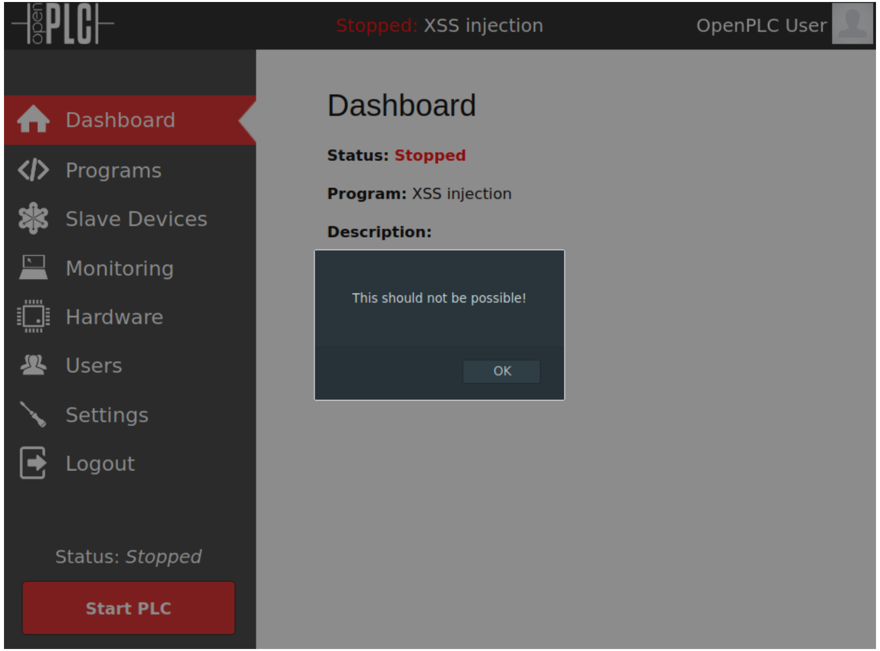


MitM modification

Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Attack Demonstration - Other Possible Attacks

- **variation of the previous attack**
  - redirect requests to a malicious server, instead of manipulating the traffic directly
  - choose what sensor or actuator values are passed on
- **sniff password to the OpenPLC web interface and upload a malicious PLC program**
- **most trivial attack:** write falsified values to any of the Modbus servers **(no authentication)**
- Exploit an **injection vulnerability** I discovered in the **OpenPLC web interface**
  - lack of input sanitation in multiple POST forms
  - **requires access to web interface** credentials (can be sniffed)
  - **consequences:** able to launch persistent XSS attacks



Attack Demonstration - Discussion

We observe that:

- the testbed is suitable to implement and demonstrate cyber-attacks
- ARP cache poisoning is an important factor in MitM attacks
- **lack of** attention towards security goals (e.g. integrity) in ICS software
- **attacks could be avoided**
  - lack of HTTPS support (OpenPLC, ScadaBR)
  - lack of TLS support in Modbus implementations $\Rightarrow$ TLS variant exists, but seems to lack common support
- confirm that PLCs are desireable attack targets

Conclusion - Status

- we succeeded in building a PLC testbed that can be used for attack demonstration and cybersecurity research

- **Strengths:**
  - good at examining network/communication based attacks $\Rightarrow$ virtualized networking
  - address common requirements (properties, components)
  - affordable/free
- **Weaknesses:**
  - strictly open-source components $\Rightarrow$ less used in industry, reduces fidelity, codebase-specific attacks
  - cannot examine more physically based attacks (e.g. those targeting I/O pins)

- **Realized Goals:** complete testbed suited for attack demonstration and research
- **Open Goals:** focus on real-time requirements, more realistic simulation

Conclusion - Future Work

- examine real-time behavior of our system and how it is affected by attacks
- improve testbed in collaboration with OT personnel ⇒ **cross-sectional teams can be very important**
- examine the applicability of the TICK stack for intrusion/anomaly detection
- explore the potential of lightweight virtualization techniques such a containerization (e.g. Docker) ⇒ might decrease fidelity

Chair for IT Security
Department of Informatics, Technical University of Munich (TUM), Fraunhofer Institute for Applied and Integrated Security AISEC
Technical University of Munich

Thank you for your attention!