PLC Secure Software Development

CL-PLC | Virtual classroom | 2 days

Audience: PLC developers
Preparedness: General PLC development
Exercises: Hands-on

PLCs play a significant role in the world today. They are used in office buildings, factories, and even power plants to automate tasks previously done by relays. While a compromised PLC in an office building usually does not mean a serious threat to the employees, a PLC in a nuclear power plant that has been overtaken by an attacker can cause a blackout in a city or even human deaths.

Embedded PLCs are often used for decades. Therefore, their security is a crucial question. But even when a PLC is manufactured perfectly, human programming and implementation errors are still present.

This course gives an overview of PLC security based on the Top 20 PLC Secure Coding Practices List, explained with examples that can occur anytime in real life.

Outline:

- IT security and secure coding
- Security architecture of ICS / SCADA / DCS networks
- PLC Input Validation
- PLC Access Control and Integrity Check
- Human Machine Interface (PLC HMI)
- Secure Coding Principles
- Principles of security and secure coding
- Knowledge sources

Participants attending this course will:

Be able to plan a secure industrial architecture
Be ready to write code for PLCs with proper input validation
Have the competence to implement access control mechanisms
Be able to check and ensure the integrity of the code running on the PLC
Understand basic concepts of security, IT security and secure coding
Be able to display relevant information on the HMI
Write modularized code split into sub-routines to enhance maintainability
Get sources and further readings on secure coding practices

Related courses:

- CL-CPA - C and C++ secure coding (ARM) (Onsite / Virtual classroom, 3 days)
- CL-CCA - Comprehensive C and C++ secure coding (ARM) (Onsite / Virtual classroom, 4 days)
- CL-CMA - C and C++ security master course (ARM) (Onsite / Virtual classroom, 5 days)
- CL-ATP - Advanced TPM Security (Virtual classroom, 3 days)

scademy
secure coding academy

Detailed table of contents

Day 1

IT security and secure coding

- Nature of security
- What is risk?
- IT security vs. secure coding
- From vulnerabilities to botnets and cybercrime
  - Nature of security flaws
  - From an infected computer to targeted attacks

Security architecture of ICS / SCADA / DCS networks

- Firewalls and the DMZ
- Security of communication
  - Communication with the HMI
  - Connection to measuring elements
  - Access to external networks

PLC Input Validation

- 6. Validate timers and counters
  - HMI trust level control
- 7. Validate and alert for paired inputs
  - Handling paired inputs
- 8. Validate HMI input variables at the PLC level, not only at HMI
  - Crafted packets
- 9. Validate indirections
  - Indirection examples
- 10. Assign designated register blocks by function (read/write/validate)
  - Main and register memory
  - By-pass Logic Attack

- 11. Instrument for plausibility checks
  - Compare integrated and time-independent measurements
  - Example: Metered pump and tank level gauge
  - Compare different measurement sources
  - Example: Airplane climbing / descending
  - Tampering detection
- 12. Validate inputs based on physical plausibility
  - Deviation and inactivity

PLC Access Control and Integrity Check

- 2. Track operating modes
  - The Remote (REM) mode
- 3. Leave operational logic in the PLC wherever feasible
  - Data manipulation in the HMI

Day 2

PLC Access Control and Integrity Check

- 4. Use PLC flags as integrity checks
  - Typical attacks
- 5. Use cryptographic and / or checksum integrity checks for PLC code
  - Checksums
  - Hashes
- 13. Disable unneeded / unused communication ports and protocols
  - Data flow diagram
- 14. Restrict third-party data interfaces
  - Sniffing and Spoofing
- 15. Define a safe process state in case of a PLC restart
  - Basic attack vectors
- 20. Trap false negatives and false positives for critical alerts
  - TRITON/TRISYS/HatMan attacks
  - Example: bus-injection

Human Machine Interface (PLC HMI)

- 16. Summarize PLC cycle times and trend them on the HMI
  - Change in cycle times
- 17. Log PLC uptime and trend it on the HMI
  - Force crash / restart
- 18. Log PLC hard stops and trend them on the HMI
  - Check before restart
- 19. Monitor PLC memory usage and trend it on the HMI
  - Tampering detection

Secure Coding Principles

- 1. Modularize PLC Code
  - Change detection

Principles of security and secure coding

- Matt Bishop's principles of robust programming
- The security principles of Saltzer and Schroeder

Knowledge sources

- Secure coding sources – a starter kit
- Vulnerability databases
- PLC security resources
- Recommended books – PLC security