An Automated Reverse Engineering Cyber Module for 5G/B5G/6G:

ML-Facilitated Pre-“ret” Discernment Module for Industrial Process Programmable Logic Controllers

Steve Chan
Decision Engineering Analysis Lab
schan@dengineering.org

DECISION engineering Analysis Laboratory

IARIA

Researcher Bio

Dr. Steve Chan is an International Academy, Research and Industry Association (IARIA) Fellow. He is an inventor with both international and U.S. patents and serves as a reviewer for 21 peer-reviewed journals/conference proceedings, such as IEEE Access in the area of cyber security for real-time control and monitoring for smart grids.

He has been active in the areas of Cyber as well as Power & Energy/Power Electronics Societies of various IEEE chapters, and he has been an invited speaker, such as at the IEEE Smart Grid Utility Cybersecurity Workshop.

He has authored/co-authored papers that were presented at the IEEE International Conference on Distributed Computing Systems (ICDCS) Workshop, IEEE International Conference on Condition Monitoring and Diagnosis (CMD), IEEE Sensors Applications Symposium (SAS), IEEE Computing and Communication Workshop and Conference (CCWC), IEEE Information Technology, Electronics & Mobile Communication Conference (IEMCON), IEEE Technically Sponsored Future of Information and Communication (FICC) Conference, IEEE International Conference on Information and Communications Technology (ICOIACT), IEEE Future Technologies Conference (FTC), IEEE International Conference on Digital Ecosystems and Technologies (DEST), and the IEEE International Conference on Collaborative Computing (CollaborateCom).

Research Domains

The Decision Engineering Analysis Laboratory has engaged in a variety of cyber-related research, such as in the areas of:

- Leveraging Sidecars for a More Probabilistic Cyber Convergence
- Systems Resilience: Reliable Cyber-protection (cyber defense, guaranteed reliability, cyber awareness, cyber-space, on-line cyber protection, traffic, etc.)
- Log Analysis
- Challenges to Cyber Services
- The Nexus of Cognitive Computing, Artificial Intelligence and Cyber Security – Anomaly Detection at Scale
- Leveraging Artificial Intelligence/Cognitive Computing to Meet the Increasing Cycles of Adaptation within the Cyber Domain
- Advances to Protect Critical Assets
- Cyber Attack Surfaces and the Interoperability of Architectural Application Domain Resiliency
- Advanced Approaches to Enhance Cyber Applications
- Cyber-Centered Major Challenges, Monitoring and Evaluating the Cyber-health of Industrial Systems
- Enhancing Cyber Infrastructural Resilience for Cyber Cities

The Challenge

Industrial Control System (ICS) components have been subject to heightened cyber risk as hardware/software supply chain vulnerabilities have been illuminated and cyberattacks have become increasingly sophisticated. At the center of this ICS cyber issue is the Programmable Logic Controller (PLC), as it is a main controller for physical processes (e.g., the control of an actuator). Many PLCs were designed for another era; they are resource-constrained, non-optimized, and beset with a variety of legacy facets (e.g., compiler, programming language, etc.). As the PLC is a principal controller for Industry 4.0, it has become a key target for cyber attackers.

A viable Monitoring/Detecting/Mitigating Module (MDMM) construct that leverages a priori scan cycle traffic and utilizes Machine Learning (ML)-facilitated PLC logic/code optimization can help mitigate against cyberattacks via an effective Automated Reverse Engineering (ARE) mechanism.

However, the ARE, if not properly architected, can also constitute a vulnerability, if it is somehow exploited by attackers.

Despite this presented dilemma and the distinct possibility of being utilized as an attack accelerant, the efficacy of ARE constitutes a key capability for forensic investigations.

The Posited Approach

To conduct PLC exploitation (e.g., malware) analysis, it is necessary to examine the PLC binary.

A ML-facilitated “Pre-‘ret’” Discernment Module (MLPRDM) shows promise. The MLPRDM focuses upon recognizing the set of legitimate instruction calls prior to the return or “ret” (or “Pre-‘ret’”) instruction contained within the subroutines of the PLC Program. Legitimate instruction calls proceed accordingly while MLPRDM-recognized illegitimate instruction calls may experience intervention (time-permitting and if practical).

The various mechanism are shown in the following Figures 1 through 4.

Figure 1

We present a Monitoring/Detecting/Mitigating Module (MDMM) Amalgam.

The MDMM utilizes an Automated Reverse Engineering (ARE) Cyber Module (ARECM), which is less prone to being utilized as an attack accelerant.

Central to the requisite "less prone" protective element is a Machine Learning (ML)-facilitated Discernment Module (MLDM), which strives to detect that an attack is occurring/has occurred.

Central to the MLDM is yet another module, which also utilizes ML facilitation so as to perform PLC logic/code optimization (a.k.a., ML-facilitated Logic/Code Optimization Module or MLLCOM).

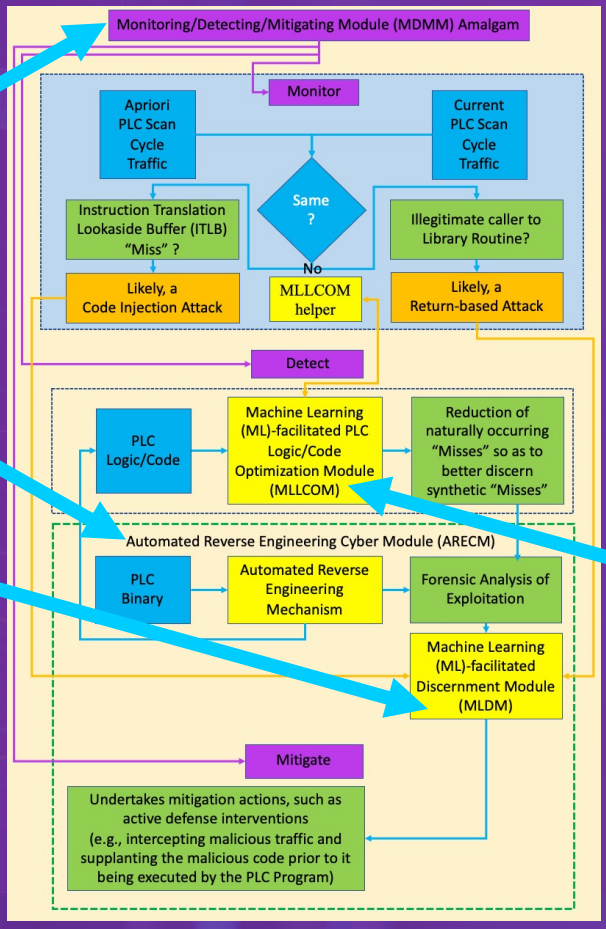
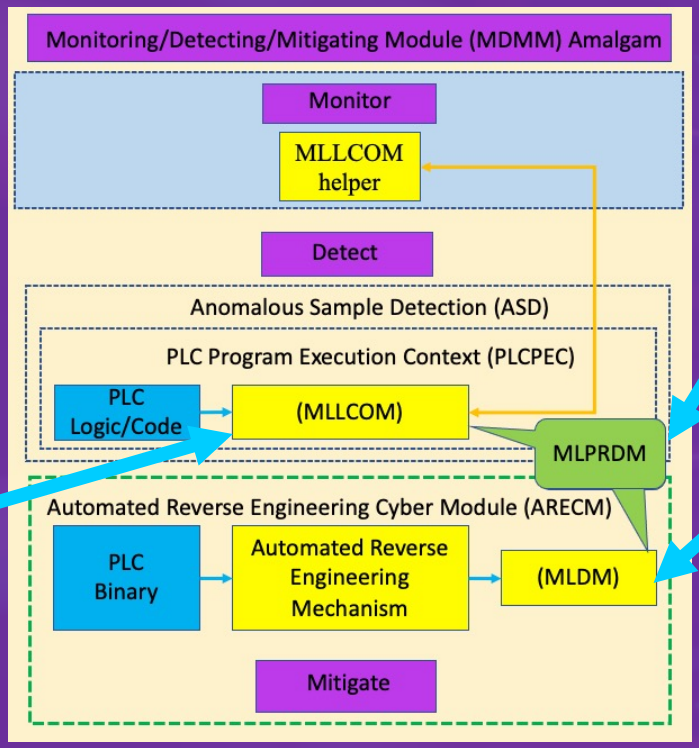


Figure 2



The ML-facilitated "Pre-'ret'" Discernment Module (MLPRDM) focuses upon recognizing the set of legitimate instruction calls prior to the return or "ret" (or "Pre-'ret'") instruction contained within the subroutines of the PLC Program. Legitimate instruction calls proceed accordingly while MLPRDM-recognized illegitimate instruction calls may experience intervention (time-permitting and if practical).

The MLPRDM gleans patterns from the work of the MLLCOM and is the key engine for the MLDM.

Figure 2 cont'd

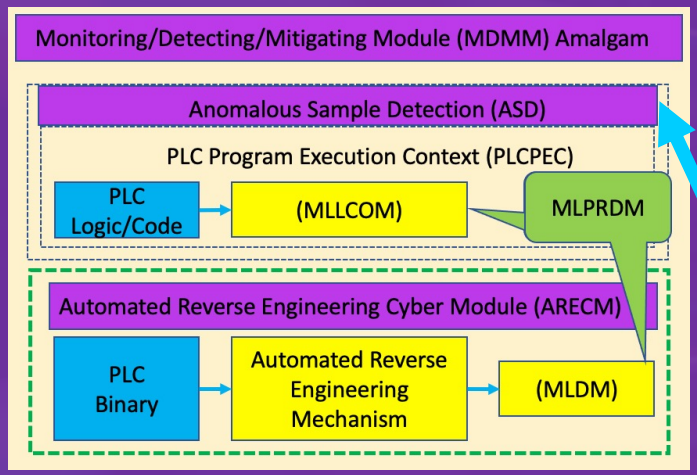
To clarify the value-added proposition of the MLPRDM, some background information is required. Broadly speaking, there are three methods for PLC exploitation: Firmware Modification Attacks (FMA), Control-flow Attacks (CFA), and Configuration Manipulation Attacks (CMA). PLC exploitation is classified by security defects: firmware security defects, program security defects, and operation security defects.*

Operation security defects can be further sub-divided into: (1) attack on protocol defects (e.g., since most communications protocols are not encrypted, packets can be captured and/or the data store of the registers can be read, and replay attacks [legitimate data is repeated and/or delayed], etc. can then be effectuated), (2) tampering attack at the Input/Output (I/O) interface (e.g., since modifying the I/O pin configuration does not necessarily issue an alarm, a tampering attack can be covertly effectuated), (3a) injection attack to affect the program flow control instructions or operational control flow (e.g., as operational control flow is dictated by PLC code blocks, intermediate code instrumentation and/or malicious code execution can exploit this facet), and (3b) return-oriented attack to affect the operational control flow (e.g., by leveraging exploits, wherein legitimate instructions are overwritten, malicious instructions can be indirectly executed.*

The latter sub-divisions (3a, 3b) are the focus of the MLPRDM.

* Please see Reference 14 within the paper.

Figure 3



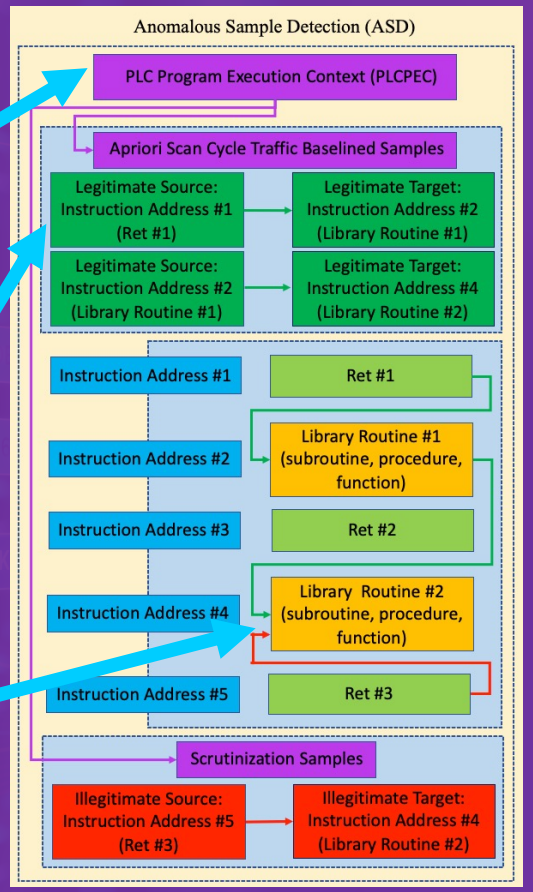
The notion of installing a general detection module aboard the PLC has, to date, met with limited appeal; due to the already limited computational resources onboard the PLC, installing an additional program (with its additional computational load requirements) aboard the already resource-constrained PLC, so as to examine the PLC Program, has met with various heuristical challenges. For example, for real-time operations, the task of detection likely is given a lower priority than that of a control task. This de-prioritization sets the stage for detection misses, so the potential efficacy is already in question.

The challenge then becomes one of designing a specialized detection module, which has both minimal impact on the computational load of the already resource-constrained PLC as well as a minimal footprint given the higher priority control tasks at hand. It turns out that this approach vector is somewhat feasible in the form of Anomalous Sample Detection (ASD).

Figure 4

ASD is underpinned by PLC Program Execution Context (PLCPEC). This PLCPEC is in the form of baselined instruction addresses and callers of library routines. The contextualizing performed, via the PLCPEC, better distinguishes between legitimate and illegitimate control flow behavior.

To operationalize the ASD approach, several facets were baselined a priori by pre-processing the PLC binary and recording the following: (1) legitimate callers of the library, (2) legitimate callers for each library routine, and (3) legitimate callers for various consecutive call patterns for the library routines. Furthermore, the instructions occurring prior to any "ret" instruction were recorded, and a relative weighting was assigned to each instruction depending upon their distance to their associated "ret."



Experimentation Findings

Due to the continuous nature of the execution scan cycle, dynamic analyses of the PLC binary is non-trivial. The MDMM architecture lends to overcoming this challenge, via the positioning of its various constituent modules. In particular, the ARECM is nicely operationalized within the ASD of the MDMM by way of the interplay between the MLDM and MLLCOM, via the MLPRDM.

With the enhanced context from the MLLCOM (given the optimization work) and the insights from the MLDM (e.g., comparison of the current scan cycle traffic with a priori scan cycle traffic) serving as accelerants for the ARECM, the MDMM architecture and underpinning MLPRDM provide enhanced discernment.

Conclusion

It has been shown that PLCs have been vulnerable to various cyber attacks, wherein the legitimate instruction in a control logic is replaced with data to cause the PLC to malfunction, and/or the PLC may have had legitimate instructions relaced with malicious instructions. In either case, the operational control flow has been compromised.

Two synergistic pathways for mitigation, among others, are plausible.

First, detection is ideal; to the degree that this can be achieved with enough time to take mitigation action, then it would be ideal to effectuate an active defense (e.g., ironically, a man-in-the-middle counter-attack) by intercepting the malicious traffic and supplanting the malicious code prior to that code being executed by the PLC Program.

Second, ARE is prudent; to the degree that it can occur in real-time for a robust diagnosis of the attack, such that intervention can be effectuated, then the intent for which discussed MLPRDM was designed would be operationalized.