Detection/Interdiction of Malware Carried by Application-Layer AMI Protocols

Overview and Problem Statement

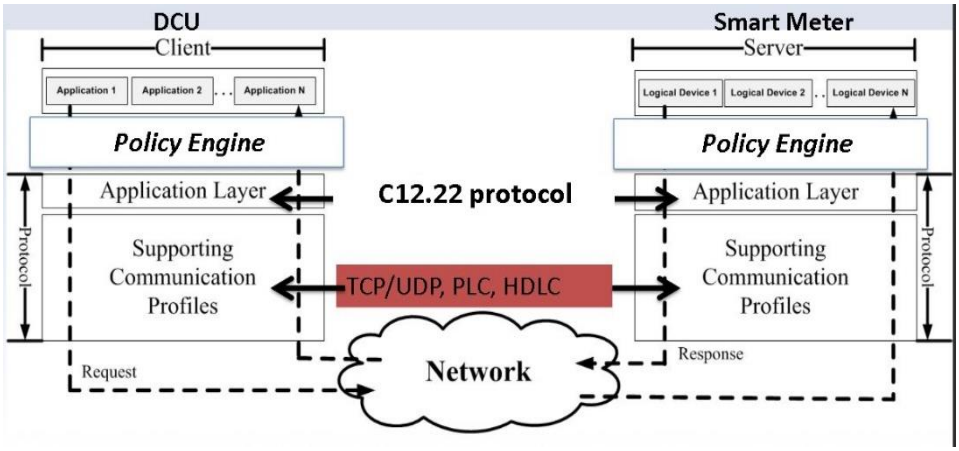
Malware could potentially hide in the payloads of standard application-layer protocols like C12.22 and DLMS/COSEM, which are extensively deployed in the AMI infrastructure. Previous work on this application-level malware detection problem by David M. Nicol and Huaiyu Zhu resulted in the proposal of a policy engine that analyzes both ingress and egress traffic from the application layer using a predefined set of policies or rules that check the packet semantics, analyze entropy levels, and look for ARM executable signatures in DLMS/COSEM protocol payloads. We extend this approach to formulate a set of rules for analyzing C12.22 protocol payloads with an additional requirement of detecting x86 binary executables hiding in packets.

Research Objectives

- Formulate semantic and communication rules to detect anomalies in C12.22 protocol payloads.
- Build a general framework for executing policy rules on C12.22 packets and provide capabilities for extensions.
- Design signature-based policy rules and investigate machine-learning-based approaches to detect x86 binaries that could be in an obfuscated, encrypted, or compressed state inside the packet.
- Evaluate the classification error rate.
- Evaluate the performance overhead.
- Integrate the policy engine with an open-source C12.22 library and demonstrate its functionality.
- **Smart Grid Application Area:** AMI, smart grid meter devices, data concentration unit devices.

Technical Description and Solution Approach

- Analyze the entropy of incoming and outgoing network traffic to detect the existence of encrypted content or packed content that might be part of malware.
- Examine non-encrypted portions of packets for specific signatures that could signify a decryption routine.
- Detect binary executables in the payload. It is harder to detect x86 binaries than ARM binaries, because of the former's complex structure and variable-length instructions.
- Use x86 instruction opcodes to construct signatures and byte sequence characteristics such as opcode frequencies, order of occurrence, and histograms and feature vectors for training classifiers.



Results and Benefits

- We have integrated the existing policy engine to work with C12.22 protocol libraries, and we have formulated and implemented the semantic rule checks required to successfully parse metering data.
- We are currently investigating and evaluating approaches to detect x86 binaries in packets.
- **Technology Readiness Level:** In progress and currently in the initial stages.

Researchers

- David M. Nicol, dmnicol@illinois.edu
- Vignesh Babu, babu3@illinois.edu

funded by the U.S. Department of Energy and the U.S. Department of Homeland Security