TCIPG

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

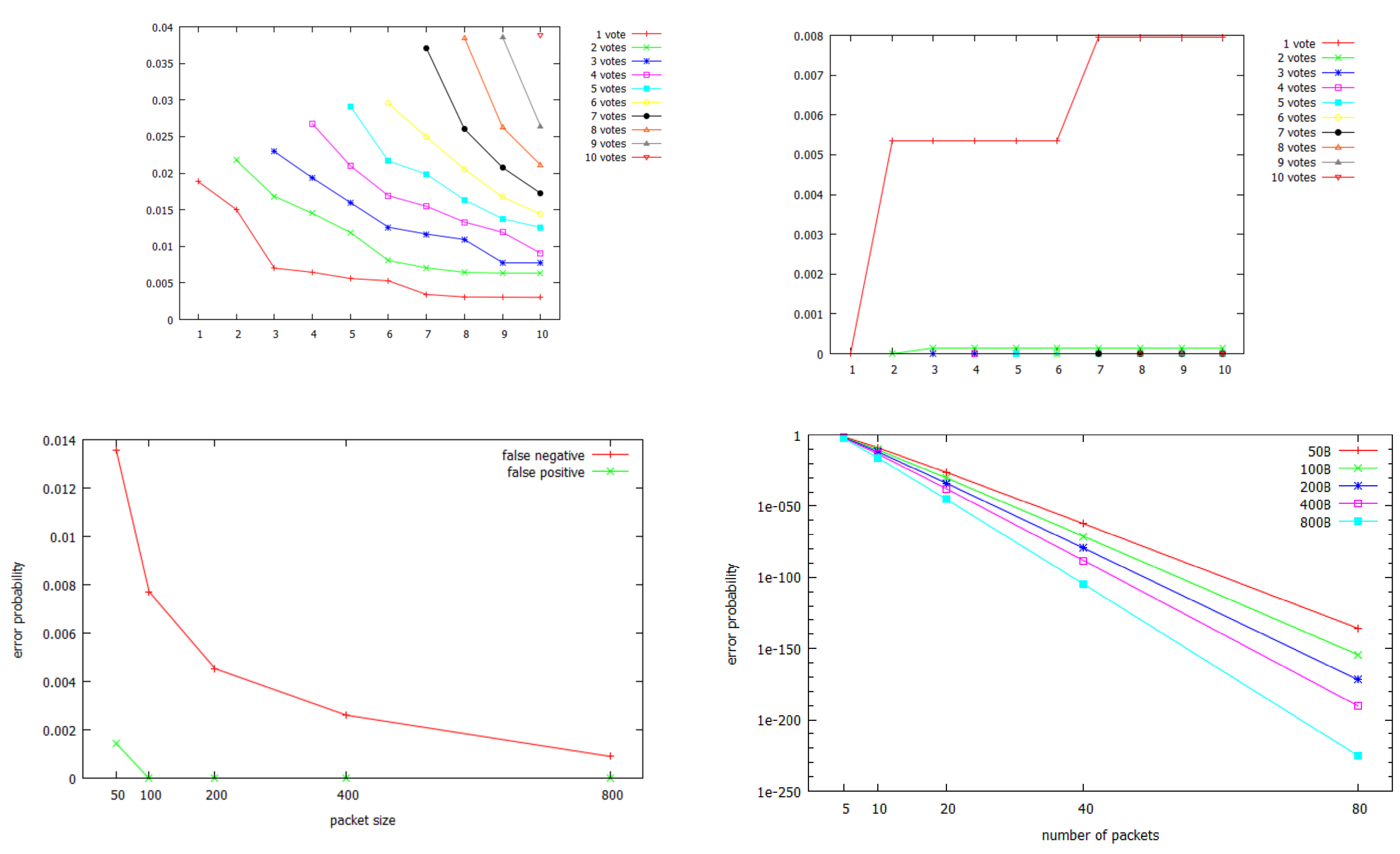
David M. Nicol and Vignesh Babu

GOALS

- Modify and integrate a previously proposed set of policies to screen malicious application-level traffic in ANSI C12.22 protocol payloads.
- Develop policies to detect the presence of x86 executables in application-level traffic.
- Evaluate the effectiveness of the policy engine.
- Evaluate the performance impact of building the policy engine in AMI applications.

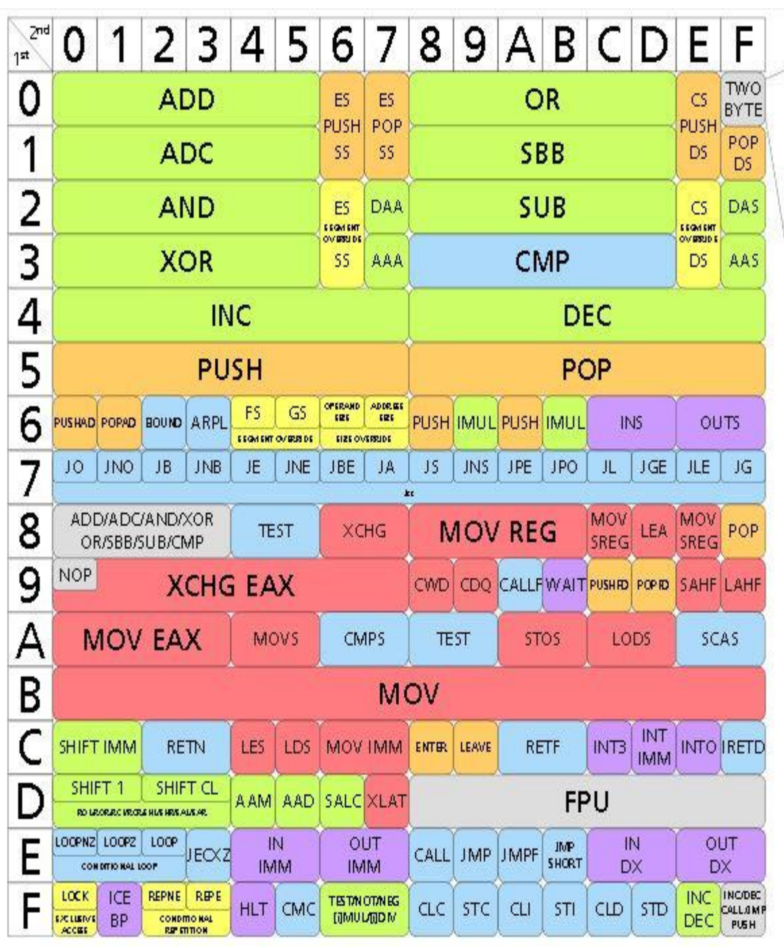
PREVIOUS WORK

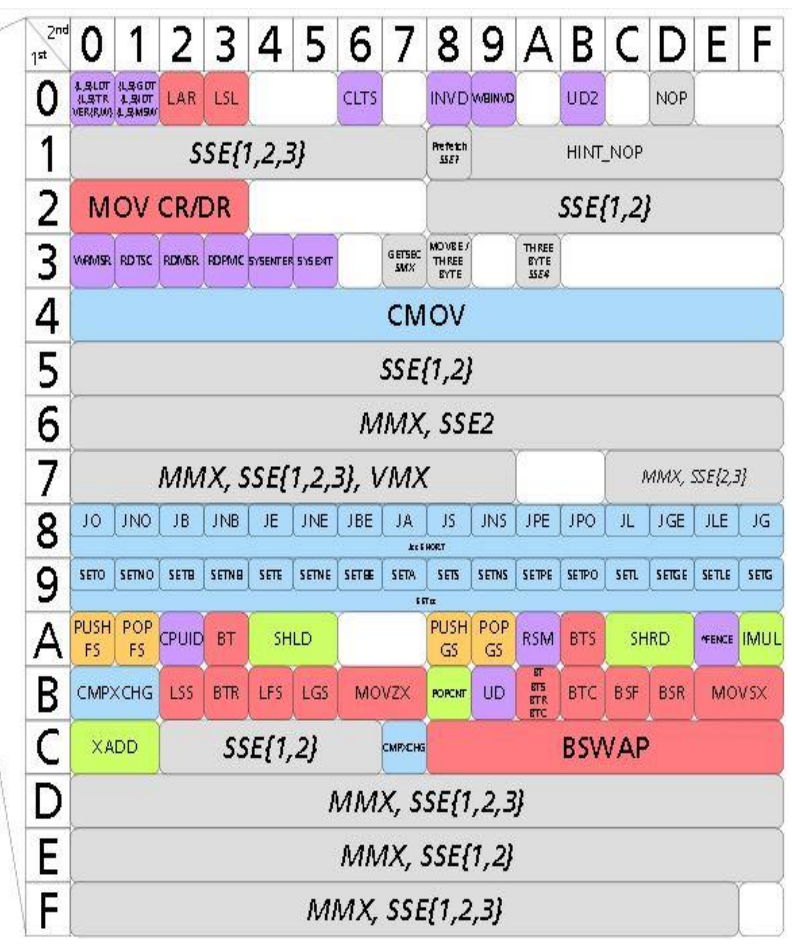
- Integrated a prototype policy engine with an open-source implementation of DLMS/COSEM.
- Design of policy rules tuned to DLMS/COSEM protocol and detection of ARM executables.
- The gathered experimental results show that the prototype policy engine is effective, with low error rates.
- Only 0.265% performance overhead was observed.



FUNDAMENTAL QUESTIONS/CHALLENGES

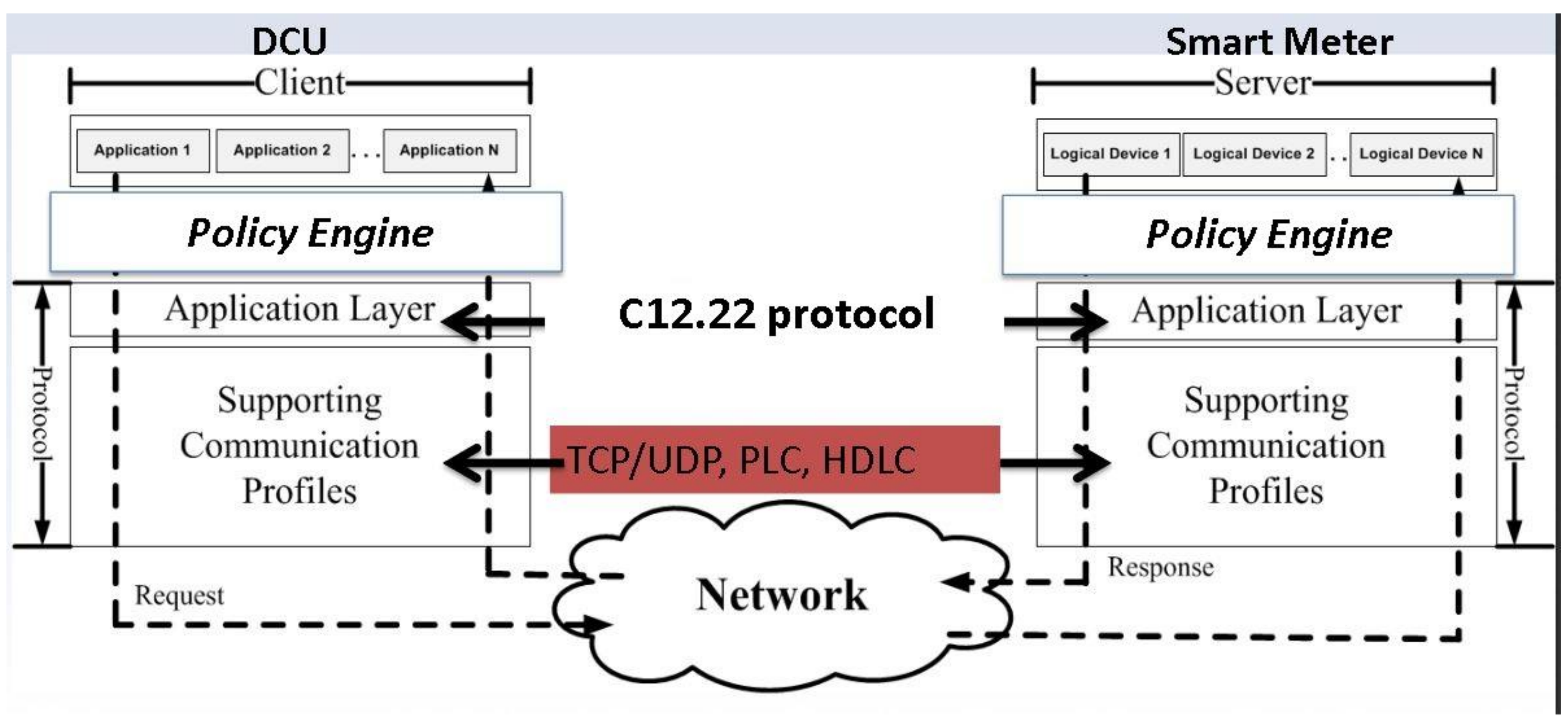
- C12.22 protocol provides several services that can be misused to inject malware into the metering infrastructure.
- The attacker might encrypt, compress, or permute bytes to avoid detection.
- Detection of x86 executables is complex because of their complex structure and variable-length instructions.
- Metering data don’t usually exhibit identifiable patterns.
- How do we design policies that successfully screen x86 executables that could be obfuscated or encrypted and minimize error rates?
- How do we minimize the overhead of this deep packet inspection process?





RESEARCH PLAN

- Developed a general framework for executing policy rules on the C12.22 protocol payloads.
- Work on developing effective policy rules for detecting x86 executables and implement them within the framework.
- Investigate the feasibility of pattern-matching approaches and machine-learning-based methods to perform classification of binaries and metering data.
- Evaluate effectiveness in terms of false positive and false negative errors.
- Conduct performance analysis that evaluates the impact the policy engine has on the throughput and latency of protocol messages.
- Bring the framework into a suitable state for open-source release.



BROADER IMPACT

- Provides an open-source framework for malicious traffic detection.
- Provides a general method for developing effective rules for policy engine.
- Provides experiment designs to evaluate such a host-based malware detection system.
- Reduces the resource requirements for deploying the policy engine.

INTERACTION WITH OTHER PROJECTS

- This host-based malware detection technology can be combined with other hardware-based or software-based intrusion detection systems (TCIPG) to detect and stop cyber attacks in AMI systems.

REFERENCES

- Park, Younghee, et al. “Prevention of malware propagation in AMI.” *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2013.
- Line, Maria B., Inger Anne Tondel, and Martin Gilje Jaatun. “Cyber security challenges in Smart Grids.” *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*. IEEE, 2011.
- Images: courtesy of Google images and adaptation from Younghee Park’s slides on “Design of policy engine for prevention of malware propagation in AMI.”

FUNDING SUPPORT PROVIDED BY DOE-OE AND DHS S&T