Securing Industrial Control Systems with a Simulation-based Verification System

Dong Jin
Department of Computer Science
Illinois Institute of Technology
dong.jin@iit.edu

Ying Ni
Department of Computer Science
Illinois Institute of Technology
yni6@hawk.iit.edu

ABSTRACT

Today's quality of life is highly dependent on the successful operation of many large-scale industrial control systems. To enhance their protection against cyber-attacks and operational errors, we develop a simulation-based verification framework with cross-layer verification techniques that allow comprehensive analysis of the entire ICS-specific stack, including application, protocol, and network layers.

1. INTRODUCTION

Protection of industrial control systems (ICSes) is a critical component of protecting against a potential "cyber Pearl Harbor" [2] — an attack that devastates the critical infrastructure and paralyzes the nation. ICSes represent a wide variety of networked information technology systems connected to the physical world to monitor and control physical processes. ICSes perform vital functions in national critical infrastructures, such as electricity, oil, gas, and water distribution; transportation systems; and even weapons systems. The disruption of these control systems could have a significant impact on public safety and health, and lead to large financial losses. Modern ICSes are increasingly adopting Internet technology to boost control efficiency, which also increases the risk of attacks and failures inherited from the commodity network infrastructure. In practice, a common way to implement security policy in ICSes is direct deployment of commercial off-the-shelf products, such as firewalls and antivirus software, which only provide fine-grained protection at single devices. Without a way to check system-wide requirements, serious security vulnerabilities can and do exist in real implementations of critical systems. To ensure that the entire system functions correctly, we need to verify not only the network layer, such as reachability among end-hosts as being verified by tools like VeriFlow [3], but also application behaviors. That would seem to be impossible in traditional networks, since many applications are outside the network operator's control. How can the operator know what applications are running and what defines "correct operation" for them? Fortunately, most ICSes have a small set of applications whose run is restricted and controlled by the operator. That unique property of ICSes allows us to investigate modeling and verification techniques to check applications with the key idea of *cross-layer verification* for detection of malicious activities and system errors. We address two key research issues: (1) the network should be giving applications an environment with required performance properties (Section 2.1), and (2) the applications should be behaving correctly on the network (Section 2.2). To answer those questions, we are developing a simulation-based framework to perform system-wide properties verification in large-scale ICSes.

2. VERIFYING ICS APPLICATIONS WITH CROSS-LAYER VERIFICATION

Figure 1 overviews the system design. The core component is the verification framework, which uses a model-checking approach to verify models against the constraints, such as security and performance requirements. Violations indicate cyber-attacks or misconfigurations. The verification framework takes inputs of network states, such as forwarding tables and topologies, from the network model, and inputs of correct application behaviors from the state-machine-based application models. The network and application models are simulated, and the verification framework is emulated (with a modified version of VeriFlow [3]) in a parallel discrete event network simulation/emulation testbed, S3FNet [4]. Our ongoing work of verifying network performance properties and the approach to extend the verification framework from the network layer to the application layer is described in the remainder of the section.

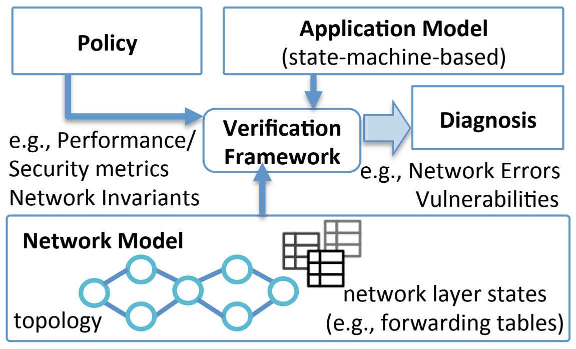

Figure 1: Simulation-based Verification System

2.1 Verifying Network Performance Metrics

Meeting specification-based performance requirements is crucial for ICS to function correctly. For example, a generic object-oriented substation event message must be delivered within 4 ms, according to IEC 61850. Typical network model-checking tools formalize packet header and location changes as state transitions. Continuous states are needed to model metrics like latency, which is not practical for large-scale systems because of the exponentially increasing state space. One approach is to leverage VeriFlow's forwarding graph model, and model the network forwarding behaviors as *weighted forwarding graphs*, e.g., we can assign delays to the links (graph edges) as weights, and compute the sum of delays when traversing the graph to check the end-to-end delay requirements). Verification of such invariants requires more storage space for link attributes and more operations during traversals than the reachability test in VeriFlow. Both storage and time increases is bounded by a constant factor times the size of the network.

We implemented the weighted forwarding graph in VeriFlow, and performed a case study in our testbed. We simulated a network consisting of 172 routers following a Rocketfuel [5] topology (AS 1755), and simulated the BGP activities by replaying traces collected from the Route Views Project [1]. We initialized the network with a BGP trace containing 90,000 updates. We then fed 1 million updates, and measured the end-to-end delay verification cost of the updates. The experiments were performed on a Dell PowerEdge R720 server with two 8-core processors (2.00GHz) and 64 GB RAM, and installed with 64-bit Linux OS. The results are shown in Figure 2. Our system is able to verify the end-to-end requirement on 86% updates within 10 ms, with mean verification time of 6.08 ms. However, the system exhibit long tail properties because a small fraction of updates result in the generation of large number of forwarding graphs.

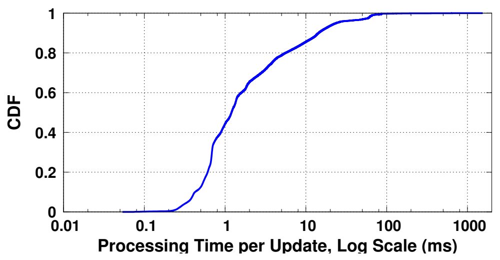

Figure 2: CDF of Update Verification Time

There are many questions we want to explore. Can we leverage knowledge from upper layers to verify performance metrics in ICSes, (e.g., the transport layer will naturally provide flow-based delay and throughput)? Can we leverage the network traffic and application domain knowledge to verify performance metrics of the physical infrastructure of an ICS, such as transmission system effectiveness, power quality and distribution reliability?

2.2 Verifying Application Semantics

Verification of applications can efficiently capture attacks that are not seen by lower layers, or some attacks may be detected much earlier in the application layer, because of the richer semantic information available. Therefore, we are exploring

(i) How to express application layer behavior. Given the fact that an ICS runs a small set of *managed applications*, it is effective to take a specification-based modeling approach for characterizing the correct behaviors. We conduct semantics analysis based on the application protocol specifications, and create models based on correct packet payloads and communication patterns. (1) Modeling packet payloads is based on the analysis of packet content to define what an application should do and reveal what an application intends to do. For example, the DNP3 protocol uses an 8-bit integer to represent the function code, in which 37 out of 256 combinations are predefined, and only a subset of the 37 function codes are supported in a real SCADA. A specification-based policy is therefore generated; any DNP3 requests with a function code out of the range may indicate that a reconnaissance scan from an adversary, exploitation of an unknown backdoor, or a denial-of-service attempt. Generation of such rules will require extensive vulnerability assessments of the particular protocols; (2) Modeling packet sequences is based on the analysis of the communication patterns among network components. For instance, in the DNP3 protocol, an "OPERATE" packet, is almost always issued right after a "SELECT" packet to control remote field devices chosen by the "SELECT" packet. The unmatched requests and responses are signs of denial-of-service attacks or replay attacks. We investigate protocol-specific analyzers that maintain flow-based states from the observed traffic, and useful info can be extracted from correlated packets of the same flow. It was recently proposed that state estimation and contingency analysis in power systems can be performed based on measurements from a specific set of substations [6]. We will leverage that algorithm to reduce the state space and further analyze the strongly correlated states to verify appropriate communication patterns.

(ii) How to check the model against what is actually happening. Verification of application and network models against a set of constraints (e.g., security requirements and network invariants) can expose malicious activities and system faults. To achieve that goal, we are developing a formal cross-layer verification framework to mathematically prove to validate different types of critical operations. The framework will be built hierarchically in several layers including a model of the network, a set of pre-defined operations and policies, protocol specifications, and a verification theorem. The layers will be implemented in ACL2, which is a software tool combining a programming language, a logic, and a theorem prover based on Common Lisp. ACL2 can automate most of the proof effort using techniques such as rewriting and mathematical induction, but we will investigate means to guide the proof by adding lemmas that the mechanical prover cannot deduct by itself.

3. REFERENCES

[1] University of Oregon Route Views Project.

[2] E. Bumiller and T. Shanker. Panetta Warns of Dire Threat of Cyberattack on U.S., New York Times, October 2012.

[3] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. B. Godfrey. Veriflow: Verifying network-wide invariants in real time. In *Proc of NSDI*, 2013.

[4] D. M. Nicol, D. Jin, and Y. Zheng. S3F: The Scalable Simulation Framework revisited. In *Proc of WSC*, 2011.

[5] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. Aug. 2002.

[6] T. Yang, H. Sun, and A. Bose. Transition to a two-level linear state estimator – part 1: architecture. *IEEE Transactions on Power Systems*, 26(1):46–53, 2011.