On IoT Misbehavior Detection in Cyber Physical Systems

Ilsun You, Kangbin Yim, Vishal Sharma, Gaurav Choudhary
Soonchunhyang University
Dept. of Information Security Engineering
ilsunu@gmail.com,
yim@sch.ac.kr,vishal_sharma2012@hotmail.com,
gauravchoudhary7777@gmail.com

Ing-Ray Chen
Department of Computer Science
Virginia Tech
irchen@vt.edu

Jin-Hee Cho
Department of Computer Science
Virginia Tech
jicho@vt.edu

***Abstract*— This article discusses a lightweight behavior rule specification-based monitoring solution for identifying misbehavior of an embedded IoT device. These unusual activities are exhibited because of attacks exploiting the vulnerability exposed through automatic model checking and formal verification. It is conclusive in the presented research that rule specification-based misbehavior detection technique outperforms contemporary anomaly-based misbehavior detection techniques for an unmanned aerial vehicle (UAV) cyber-physical system.**

Keywords—IoT, intrusion detection, cyber physical systems, zero-day attacks.

1. INTRODUCTION

Misbehavior detection techniques for Internet of Things (IoT) embedded cyber-physical systems (CPS) in general can be classified into three types: signature-based, anomaly-based and specification-based techniques [5]. Our behavior rule specification-based misbehavior detection technique proposed in this work falls under specification-based detection. The signature-based detection is disposed to counteract the possibilities of zero-day attacks. Specification-based techniques are considered rather than anomaly-based techniques for misbehavior detection to avoid the high cost associated with profiling and learning anomaly patterns for resource-constrained IoT devices and to prevent high false positives. It is argued that contemporary anomaly-based misbehavior detection methods for IoT-embedded CPSs [6] [7] based on profiling and machine learning through correlation and statistical analysis of a large amount of data or logs for classifying misbehavior will not work for IoT-embedded CPSs because of high memory, run time, communication, and computational overhead, considering the fact that many embedded IoT devices are severely constrained in resources.

2. SPECIFICATION-BASED MISBEHAVIOR DETECTION FOR EMBEDDED IOT IN CPS

2.1Modeling and Verification of Behavior Rules

The design concept of "operational profile" [3] is used during the testing and debugging phase of an embedded IoT device when the IoT software is built to identify the complete set of behavior rules. A mission assignment in an embedded IoT device's operational profile explicitly defines a set of security requirements for the mission to be successful, from which a set of threats as well as a set of behavior rules to cope with the threats may be automatically derived. The verification that the behavior rules generated are correct and cover all the threats (or satisfy the security requirements) is done through automatic model verification of the behavior rules. The basic idea is to prove that the behavior rules can guarantee all security requirements are not violated, so any violation of the security requirements implies violations of the behavior rules.

2.2 Automatic Transformation of a Behavior Rule Set to a State Machine for Misbehavior Detection

The behavior-rule-to-state-machine transformation process is automatic and it involves the identification of a "bad behavior indicator" based on violation of a defined rule. More specifically, a conjunctive normal form predicate [6] is created to define each bad behavior indicator such that if the predicate is evaluated true then it means that the corresponding behavior rule is violated; otherwise, the behavior rule is satisfied. Each bad behavior rule indicator is therefore a state component with a true (1) or false (0) value in the underlying state machine. Following this, there are 2^n states, out of which only one is a safe state (when all n bad behavior indicators are false taking the value of 0) and all other 2^n - 1 states are unsafe states. It is worth mentioning that we defend against zero-day attacks that try to avoid pre-established behavior rules by identifying the complete set of "bad behavior indicators" (mechanically derived from the corresponding set of behavior rules) that can possibly fail a mission assigned for execution. A malicious embedded IoT device can avoid being detected only if it never enters an unsafe state.

2.3 Lightweight Runtime Collection of Compliance Data

Unlike anomaly detection which frequently requires heavy resources to profile/learn anomaly patterns, the behavior rule specification-based data collection process is lightweight. By using the transformed state machine, only periodical monitoring is required to check if a trustee IoT device is in safe or unsafe states. A monitor device is aware of the transformed state machine of the trustee device at bootstrap time. The monitor device evaluates the states of the trustee IoT device in safe and unsafe states during its event triggered transitions. A monitor device can save energy by monitoring and recording the state of the trustee device in discrete time space so that $c=\sum_i o_i/N$ where c is the compliance degree of a device, N is the number of times it probes the state of the trustee device in a monitoring time interval T and $o_i = 1$ if the

trustee IoT device is in a safe state in the i^{th} probe; 0 otherwise. With this lightweight data collection process, a monitor device manages a trustee device's cooperation history $c_1, c_2, \ldots, c_n$over n monitoring time intervals.

2.4. Lightweight Statistical Analysis

The compliance degree of a trustee IoT device is modeled by a random variable X with $G(\cdot)$=*Beta*(α,β) distribution [6]. X=0 means the behavior is totally unacceptable and X=1 means the behavior is completely acceptable, such that $G(a)$, 0≤a≤1, is given by:

$$G(a) = \int_0^a \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\,dx \quad (1)$$

The expected value of X is given by:

$$E[x] = \int_0^1 x \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\,dx = \frac{\alpha}{\alpha+\beta} \quad (2)$$

The α and β are maximum likelihood-based estimated parameters observable by using the compliance degree history $c_1, c_2, \ldots, c_n$ of a device collected during runtime. In this case, the run time complexity is O($nlogn$). For an extremely resource-constrained monitor IoT device such as a sensor, the compliance degree can be modeled by X with a simple one-parameter *Beta*(β) distribution, i.e. α = 1. For this, the density becomes $\beta(1-X)^{\beta-1}$for $0 \le X \le 1$ and 0 otherwise, where β is given by a simple analytical expression:

$$\beta = n/\sum_{i=1}^{n} \log(1/1-c_i). \quad (3)$$

In this case, the run time complexity is only O(n). The misbehavior detection prediction accuracy can be measured by false negative and false positive probabilities, denoted by P_{fn} and P_{fp}, respectively. A simple threshold-based criterion is adopted, which means if a "bad" IoT device's compliance degree denoted by X_b following the earlier defined G (.)=*Beta*(α, β) is higher than the system defined compliance threshold C_T, then there is a false negative, i.e.,

$$P_{fn} = \Pr(X_b > C_T) = 1 - G(C_T). \quad (4)$$

In case of a "good" IoT device's compliance degree denoted by X_g with a G(.)=*Beta* (α, β) distribution is less than or equal to C_T, then there is a false positive, i.e.

$$P_{fp} = \Pr(X_g \le C_T) = G(C_T). \quad (5)$$

3. Intrusion Detection of UAV CPS

The defined IoT IDS technique is applied to a UAV (a drone) embedded in a UAV-CPS [1]. We first conduct automatic model verification of the behavior rules generated by verifying that the behavior rules generated are correct and cover all the threats (or satisfy the security requirements). Currently we are exploring ACL2 [2], a theorem prover, to define security requirements as well as behavior rules as ACL functions. The verification is done by defining a theorem (also an ACL function) that is evaluated to be true, thus proving that under certain assumptions (to cover properties that may not be monitored by the behavior rules) if this UAV does not violate the behavior rules, it does not violate the security requirements either. We transform the behavior rules identified into a state machine for lightweight misbehavior detection. A monitor UAV is assigned to monitor a trustee UAV. With the compliance degree history $c_1, c_2, \ldots, c_n$collected, we calculate P_{fn} and P_{fp}. The minimum compliance threshold C_T is adjusted to control P_{fn} and P_{fp}.

The IoT IDS technique is compared with ACCM developed by Tsang and Kwong [4] for industrial CPSs. Figure 1 compares the ROC graphs (true positive rate or $1 - P_{fn}$ versus P_{fp}). A mis-monitoring probability $p_{err} = 1\%$ is considered due to environment noises. Figure 1 shows that the IoT IDS technique outperforms ACCM; especially in P_{fp}.The AUROC (area under ROC) of our IoT IDS technique is dominantly greater than that of ACCM.

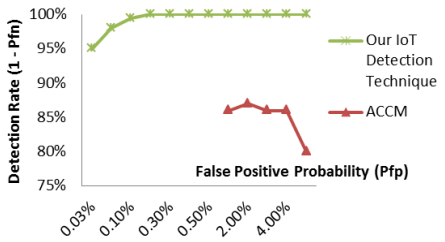

Figure 1: IDS Performance Comparison in terms of AUROC.

4. Conclusion

The behavior rule specification-based misbehavior detection technique can be positioned as the only feasible solution in terms of low memory, run time, communication, and computation overhead, and high misbehavior detection prediction accuracy to ensure protection of resource-constrained embedded IoT devices against zero-day attacks. This work is a start toward verifying our position.

Acknowledgment

This work is partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00664, Rule Specification-based Misbehavior Detection for IoT-Embedded Cyber Physical Systems). This work is also supported in part by the U.S. AFOSR under grant number FA2386-17-1-4076. This article is the poster version of our partially published as well as ongoing research.

References

[1] D. He, S. Chan, and M. Guizani, "Drone-assisted Public Safety Networks: The Security Aspect," *IEEE Communications Magazine,* 2017, pp. 2-8.

[2] M. Kaufmann and J.S. Moore, A Computational Logic for Applicative Common Lisp, http://www.cs.utexas.edu/users/moore/acl2/, 2017.

[3] J. Musa, "Operational profiles in software reliability engineering," *IEEE Software*, pp. 14–32, Mar. 1993.

[4] C.-H. Tsang and S. Kwong, "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction," *Inter. Conference on Industrial Technology*, pp. 51–56, Dec 2005.

[5] B.B. Zarpelao, R.S. Miani, C.T. Kawakani, and S.C. de Alvarenga, "A Survey of Intrusion Detection in Internet of Things," *Journal of Network and Computer Applications,* vol. 84, 2017, pp. 25-37.

[6] R. Mitchell and I.R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593-604, May 2014.

[7] I. You, K. Yim, V. Sharma, G. Choudhary, I.R. Chen, and J.H. Cho, "Misbehavior Detection of Embedded IoT Devices in Medical Cyber Physical Systems," *ACM MedSPT*, Sept 2018.