7-2020

ObjSim: Efficient testing of cyber-physical systems

Jun SUN

Zijiang YANG

ObjSim: Efficient Testing of Cyber-Physical Systems

Jun Sun
junsun@smu.edu.sg
Singapore Management University
Singapore

Zijiang Yang
yang@guardstrike.com
GuardStrike Inc.
China

ABSTRACT

Cyber-physical systems (CPSs) play a critical role in automating public infrastructure and thus attract wide range of attacks. Assessing the effectiveness of defense mechanisms is challenging as realistic sets of attacks to test them against are not always available. In this short paper, we briefly describe smart fuzzing, an automated, machine learning guided technique for systematically producing test suites of CPS network attacks. Our approach uses predictive machine learning models and meta-heuristic search algorithms to guide the fuzzing of actuators so as to drive the CPS into different unsafe physical states. The approach has been proven effective on two real-world CPS testbeds.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**.

KEYWORDS

cyber-physical system, network, fuzzing, machine learning, testing

ACM Reference Format:
Jun Sun and Zijiang Yang. 2020. ObjSim: Efficient Testing of Cyber-Physical Systems. In *Proceedings of the 4th ACM SIGSOFT International Workshop on Testing, Analysis, and Verification of Cyber-Physical Systems and Internet of Things (TAV-CPS/IoT '20), July 19, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3402842.3407158

1 INTRODUCTION

Cyber-physical systems (CPSs) are characterized by computational elements and physical processes that are deeply intertwined, each potentially involving different spatial and temporal scales, modalities, and interactions. We define CPSs as systems in which algorithmic control and physical processes are tightly integrated. Concretely, we assume that they consist of computational elements such as programmable logic controllers (PLCs), distributed over a network, and interacting with their processes via sensors and actuators. The operation of a CPS is controlled by its PLCs, which receive readings from sensors that observe the physical state, and then compute appropriate commands to send along the network to the relevant actuators. In our work we assume that the sensors read continuous data and that the states of the actuators are discrete.

CPSs are commonly used to automate aspects of critical civil infrastructure, such as water treatment or the management of electricity demand [7]. Given the potential to cause massive disruption, such systems have become prime targets for cyber attackers, with a number of successful cases reported in recent years [4, 6]. However, CPSs are very difficult to reason about: while individual control components (e.g. PLC programs) may be simple in isolation, reasoning about the behaviour of the whole system can only be done with consideration of how its physical processes evolve and interact. This often requires considerable domain-specific expertise beyond the knowledge of a typical computer scientist, which is one of our principal motivations for achieving full automation.

2 OUR APPROACH

Fuzzing, which plays a key role in our solution, is in general an automated testing technique that attempts to identify potential crashes or assertion violations by generating diverse and unexpected inputs for a given system [8]. Most well-known tools perform fuzzing on programs, but in the context of CPSs, we consider fuzzing at the network level. Furthermore, the goal of our fuzzing differs in that we are trying to drive physical sensors out of their safe ranges, using an underlying method that is ML-guided.

Our technique uses predictive machine learning models and meta-heuristic search to intelligently fuzz actuator commands, and systematically drive the system into different categories of unsafe physical states. Smart fuzzing consists of two broad steps. First, we learn a model of the CPS by training ML algorithms on physical data logs that characterize its normal behaviour. The learnt model can be used to predict how the current physical state will evolve with respect to different actuator configurations. Second, we fuzz the actuators over the network to find attack sequences that drive the system into a targeted unsafe state. This fuzzing is guided by the learnt model: potential manipulations of the actuators are searched for, and then the model predicts which of them would drive the CPS closest to the unsafe state.

Our design for smart fuzzing was driven by four key requirements. First, that it should be general, in the sense that it can be implemented for different CPSs and a variety of sensors and actuators. Second, that the approach should be comprehensive, in that the suites of attacks it constructs should systematically cover different categories of sensed physical properties, rather than just a select few. Third, that it should be efficient, with each attack achieving its goal quickly, posing additional challenge for countermeasures. Finally, that it should be practically useful, in that it is straightforward to implement for real CPSs without any formal specification or specific technical expertise, and that the 'test suites'

TAV-CPS/IoT '20, July 19, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8032-4/20/07…$15.00
https://doi.org/10.1145/3402842.3407158

of attacks are of comparable quality to expert-crafted benchmarks, thus a reasonable basis for assessing attack defense mechanisms.

Our approach for automatically finding network attacks on CPSs consists of two broad steps in turn: learning and fuzzing. In the first step, we learn a model of the CPS that can predict the effects of actuator configurations on the physical state. The model takes as input the current readings of all sensors and a proposed configuration of the actuators, returning as output a prediction of the sensor readings that would result from adopting that configuration for a fixed time interval. The idea is that this model can later be used to analyze different potential actuator configurations, and help inform which of them is likely to drive the system closer to a targeted unsafe state. To learn this model, we extract a time series of sensor and actuator data from the system logs and train a suitable machine learning algorithm.

The second step of our approach searches for commands to fuzz the actuators with that will drive the CPS into an unsafe physical state. To find the right commands, our approach applies a search algorithm over the space of actuator configurations, returning the configuration that is predicted by the model (of the first step) to drive the CPS the closest to an unsafe state. We explore different search algorithms for this task, including random, but also metaheuristic (e.g. genetic algorithms) given that the state space of actuators can grow quite large (e.g. 2^{26} possible configurations in SWaT). We use fitness functions to evaluate predicted sensor states with respect to the attack goal.

3 EVALUATION

To evaluate our approach against these requirements, we implemented it for two CPS testbeds. First, the Secure Water Treatment (SWaT) testbed [1], a fully operational water treatment plant consisting of 42 sensors and actuators, able to produce five gallons of drinking water per minute. Second, the Water Distribution (WADI) testbed [3], a scaled-down version of a typical water distribution network for a city, built for investigating attacks on consumer water supplies with respect to patterns of peak and off-peak demand. The designs of these testbeds were based on real-world industrial purification plants and distribution networks, and thus reflect many of their complexities. We found that smart fuzzing could automatically identify suites of attacks that drove these CPSs into 27 different unsafe states involving water flow, pressure, tank levels, and consumer supply. Furthermore, it covered six unsafe states beyond those in an established expert-crafted benchmark [5]. Finally, we evaluated the utility of smart fuzzing for testing attack defense mechanisms by launching it with SWaT's invariant-based monitoring system enabled [2]. Our approach was able to identify two attacks that evaded detection by its physical invariant checks, highlighting a potential weakness that could be exploited by attackers with the capabilities to bypass its other conditions.

Our evaluation addresses five research questions based on our original design requirements for smart fuzzing (Section I): RQ1 (Efficiency): How quickly is smart fuzzing able to find a targeted attack? RQ2 (Comprehensiveness): How many unsafe states can the attacks of smart fuzzing cover? RQ3 (Setup): Which combinations of model and search algorithm are most effective? RQ4 (Comparisons): How do the attacks compare against those of other approaches or those in benchmarks? RQ5 (Utility): Are the discovered attacks useful for testing CPS attack detection mechanisms? RQs 1-2 consider whether smart fuzzing achieves its principal goal of finding network attacks. We assess this from two different angles: first, in terms of how quickly it is able to drive the CPS into a particular unsafe state; and second, in terms of how many different unsafe states the attacks can cover. RQ 3 considers how different setups of smart fuzzing (i.e. different models or search algorithms) impact its ability to find attacks. RQ 4 compares the effectiveness of smart fuzzing against other approaches: first, the baseline of randomly mutating actuator states without reference to a model of the system; and second, an established, manually constructed benchmark of attacks [5]. Finally, RQ 5 investigates whether the attacks found by smart fuzzing are useful for testing existing cyber-security defense mechanisms. Our empirical study shows promising results on all five research questions.

We remark on some threats to the validity of our evaluation. First, our approach was implemented for CPS testbeds: while they are real, fully operational plants based on the designs of industrial ones, they are still smaller, and our results may therefore not scale-up (this is difficult to test due to the confidentiality surrounding plants in cities). Second, the initial states of the testbeds were not controlled, other than to be within their normal ranges, meaning that our performance results may vary slightly. Finally, for testing CPS attack detection mechanisms, we only studied an invariant based solution, meaning that our conclusions may not hold for other types of defenses.

4 CONCLUSION

Active fuzzing, a black-box approach for automatically building test suites of packet-level CPS network attacks, overcomes the enormous search spaces and resource costs of such systems. Key to achieving this was our use of online active learning, which reduced the amount of training data needed by sampling examples that were estimated to maximally improve the model.

REFERENCES

[1] [n.d.]. iTrust Labs_SWaT - iTrust. https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/. (Accessed on 09/26/2019).

[2] Sridhar Adepu and Aditya Mathur. 2018. Distributed attack detection in a water treatment plant: Method and case study. *IEEE Transactions on Dependable and Secure Computing* (2018).

[3] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[4] ICS-CERT Alert. 2016. Cyber-attack against ukrainian critical infrastructure. *Cybersecurity Infrastruct. Secur. Agency, Washington, DC, USA, Tech. Rep. ICS Alert (IR-ALERT-H-16-056-01)* (2016).

[5] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2016. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security*. Springer, 88–99.

[6] John Leyden. 2016. Water treatment plant hacked, chemical mix changed for tap supplies. *The Register* (2016).

[7] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. 2010. Cyber-physical systems: the next computing revolution. In *Design automation conference*. IEEE, 731–736.

[8] Ari Takanen, Jared D Demott, Charles Miller, and Atte Kettunen. 2018. *Fuzzing for software security testing and quality assurance*. Artech House.