Constructing Token Graph for Adaptive Fuzzing for Cyber-Physical Systems

Incheol Shin, Lan Dan
Department of Computer Engineering, Pukyong National University
(48513) 45, Yongso-Ro, Nam-Gu, Busan, Korea
{icshin, 202256827}@pknu.ac.kr

ABSTRACT

The proliferation of IoT(Internet of Things) technologies integrate the computational devices into physical objects in the networks, called Cyber-Physical Systems (CPS). Since the priority order of security triad on CPS differs from the Internet, the security assessment would play vital role to support their proper operations against various attacks and vulnerabilities. In this work, we study the way to construct token graph to diagnose the communicational relations among the components of CPS and mutate efficient testing messages in order to design adaptive fuzzer for CPS. The token graph would be helpful to reason the feedback loops on the systems and identify the safety-vulnerabilities under security-vulnerabilities.

KEYWORDS

Software Vulnerabilities, Fuzzing, Cyber-Physical System Security

1 INTRODUCTION

Just as the computer networks have been converting the way people exchange with information, CPS(Cyber-Physical Systems) are revolutionizing the way people control things in the world. The systems integrate sensing, computing, networking with the physical processes to optimize controlling physical objects and infrastructure. The computing systems in the CPS consistently monitor and control the physical processes, with feedback loops where the processes affect computations and vice versa. As the matter of fact, the exploitation of the CPS in the various range of infrastructures emphasizes their potential role in ensuring economic development [1, 2].

As CPS proliferate on daily operations of modern societies, they would become more appealing targets for malicious attacks [3, 4]. The widespread employment of the systems results in a significant expansion in their attack surface, which leads failures or crashes on the different component the CPS [5]. However, the priority order of security triad on CPS is AIC(Availability, Integrity, Confidentiality) because their nature to deliver not only the information but also the control message. In other words, the security triad for the Internet is security-critical with the priority order of CIA(Confidentiality, Integrity, Availability), which differs from the CPS, safety-critical systems.

In the past decade, it is no longer rare to see the safety incidents on CPS have caused more serious impacts on the systems as well as humen lives. More specifically, the improper operations or failure of the systems affected by security attacks would result in interdependent and complicated safety-issues, rather than separately, in sequence, limited in cyber spaces. Consequently, the risk assessment on the CPS to enhance their openness has been considered one of the essential countermeasures against the attacks. The traditional means of fuzzing looking into expanding code coverage by iteratively mutating inputs can be computationally ineffective. The existing fuzzing methods may not be able to capture the relationship within the network flow or meaning of packets, as the typical loading profile is highly volatile and the aggregation effect vanishes. The fuzzers for electrical control systems must be designed to facilitate their security evaluation in adaptively generating input data so as to explore deeper into the control systems.

The broader definition of successful fuzzing for critical infrastructure assets describes more aggressive mutation to craft hostile inputs so as to discover bugs that result in any kind of disruption on the target systems [6]. However, the widespread utilization of proprietary protocols for control systems, i.e., SCADA, Smart Grid and Microgrid, makes it infeasible to employ existing fuzzing tools, which operate in the cases that the protocol semantics are known, testing target could be instrumented, or at least large network traces are available [7]. Although the study of fuzzing networks without the protocol specification have been investigated in handful of applications, such as GPF (General Purpose Fuzzer), CFG9000 fuzzer, PROTOS [8–10], extracting information about the network protocols from the captured network traffic to produce the fuzzed inputs by applying various token-specific transformations may result in prohibitive computational costs[11,12]. In this work, we devise the construction algorithm for token graph from the network traffic to analyze the relation in the message by the basic idea of LZFuzzing techniques.

The rest of this paper is organized as follows. Section 2 provides a preliminaries and the problem definition. In the section 3, we discuss the construction algorithm for the token graph from the network traffic. Section 4 concludes the idea of token graph construction algorithm.

2 Fuzzing for CPS

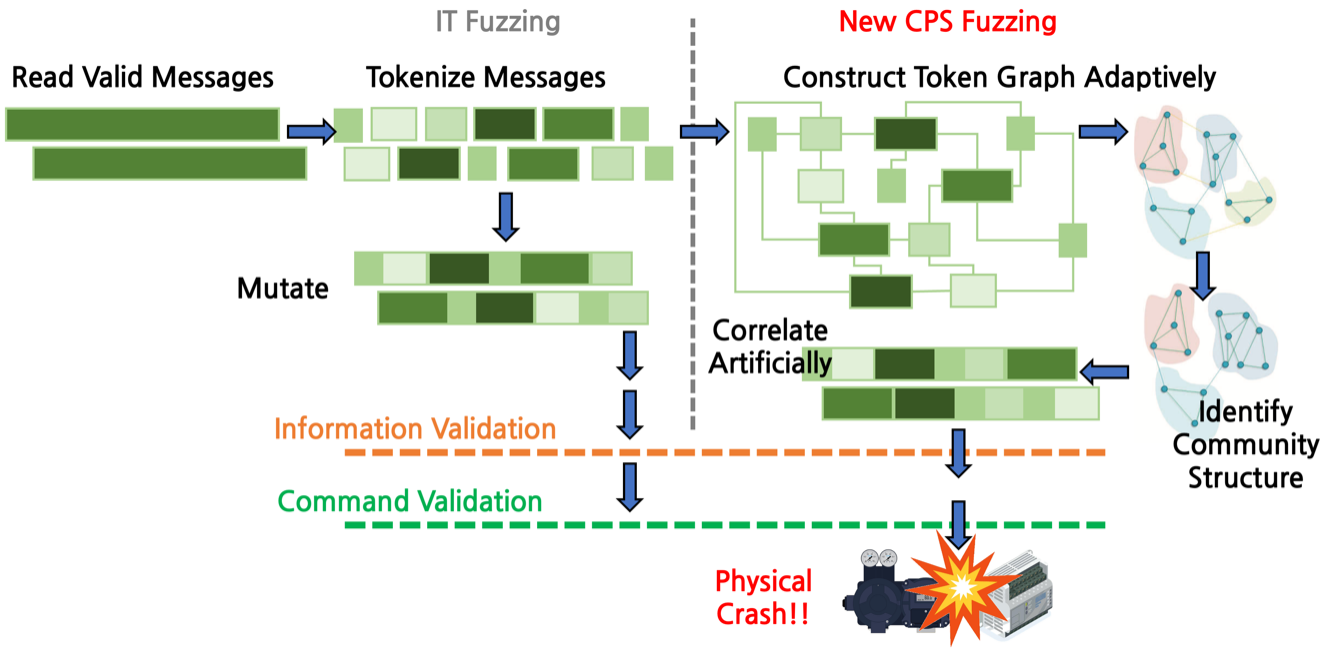

Figure 1: Overview of fuzzing technique for CPS

The crux of our research is to devise an inline algorithm for adaptive mutational fuzzing by employing the dependence relations of the evolutionary network traffic. This primary intuition of the approach leaves us with three challenges: what to estimate the packet structure on the proprietary protocols, how to reveal the interplay in the variation of network packets, and how to discretize the mutational process in terms of running time. Throughout this paper, our mutational fuzzing consists of the following three techniques with appropriate modification. First of all, constructing token graphs from the payload in the captured network traffic would help to discover the structural sequences that they are correlated. The reverse-engineering the poorly documented protocols adapts the tokenizing techniques from Lempel-Ziv(LZ) compression algorithm[12, 13]. However, the pieces of strings from the tokenization have been merely mutated to generate test cases by flipping inputs of a seed, which circumvents the identification of the dependence between packet fields to fuzz. Secondly, detecting community structure corresponding to groups of tokens with the same function may demonstrate not only the functional relationship between tokens but also the interplay between the functionalities. In order to cope with the dynamics and evolution of the token graph as the network traffic flows large, this is further motivated by a previous research we conducted: the module identification in evolving networks [11] minimizes the runtime by providing a compact graph representation while it dynamically updates community structures in each network snapshot. Lastly, since the value of the information field in the cyber physical system network packets may not affect the crash but pass them through the sanity checks, the adaptive fuzzer in Micro grid should maximize the probability to pick a set of desired valid data with invalid control commands so as to prevent it from creating trivial inputs that fail error handling code or syntax checks. In this paper, we introduce a prioritized dominating tree to determine the mutational sequences on the possible input parameter combinations in the sets.

3 Token Graph for CPS Fuzzing

This section presents a systematic way of building a set of mutational inputs to reveal bugs harboring vulnerabilities in Microgrid systems by considering the evolutionary dependence relations in the network traffic. Our fuzzing technique may not only predict the structural format of the protocol but also trace the interdependent changes of the structure to improve the adaptiveness into the target networks.

3.1 Constructing Token Graph

While we base our fuzzing model for the inference on network packet fields from the tokenization in the LZ compression algorithm, the complexity of the sequential dependency in the components of packets is distinctively embedded in the relation-based approach through the graph concept. The existing LZFuzzer[7] works by building up a dictionary of tokens, the longest unique subsequences of bytes, from the network streams of packets and merely mutates them through the GFP engine. Instead, we exhibit radically different estimation phase to discover the interdependency between the fields in the Microgrid network packets. We represent the term interdependency in the same vein to the meaning of correlation between the fields since cyber-physical systems like Microgrid may exchange control commands as well as informative data within a single packet. Inherently, only the legitimate commands trigger the transition of the device state, and the legitimacy would be validated by authentication of the data in

the packet. That is, the fuzzing strategy for Microgrid required to be expanded to attempt to learn more about the semantic dependence in the packets so that it would benefit to explore the deeper application logics.

3.2 Constructing Token Graph Algorithm

```
Constructing LZCT Graph Procedure(b)

Input: Byte stream b to mutate for fuzzed inputs and
LZCT Graph G
Output: Lempel-Ziv compression token based graph
G=(V, E)

G = {empty}
len = strlen(b)
start = 0
end = 0

while end < len do
  b_sub = b[start : end]
  if b_sub = v where v in V then
    end = end + 1
    p = v
  end
  else
    V = V merge {b_sub}
    start = end
    if G not empty then
      /* Decrease weights in the edges between p and
NEIGHBOR(p) by 1  */
      forall v in NEIGHBOR(p) do
        e = e(v, p, w+1)
      end
      w = 0 /* Initial weight 0 */
      /* Connect p and b_sub with initial weight. */
      E = E merge e(p, b_sub, w)
    end
  end
end
return G
```

The runtime complexity for this algorithm would be dependent on the number of the bytes in the network traffic until the inspection on the byte stream finalized. Inside the iterative process of while statement, a vertex in the token graph would need to check the neighbors to see if the weights of the edges associated to the vertex. Consequently, the total time complexity is the maximum degrees multiplied by the number of bytes in the traffic.

4 Conclusion

Through this work, we address the limitations to a set of existing fuzz-testing techniques for Microgrid control systems and introduce the details of our adaptive LZFuzzing mutator. Previous LZfuzzing technique presents the motivation to improve the fuzzing efficiency by exploring the identification of community structure in the evolving networks, token graph specifies an adaptive LZFuzzing approach for tuning mutational operations so as to create test inputs that would be able to bypass the sanity checks and find the deeply nested bugs or vulnerabilities in the Microgrid systems.

REFERENCES

[1] L. Shi, Q. Dai, Y. Ni, *Cyber–Physical Interactions in Power Systems: View of Models, Methods, and Applications,* Electr. Power Syst. Res. 163 (2018) 396–412.

[2] G. Wen, W. Yu, X. Yu, J. Lü, *Complex Cyber-Physical Networks: from Cybersecurity to Security Control*, Journal of Systems Science and Complexity 30 (1) (2017) 46–67.

[3] J.R. Lindsay, *Stuxnet and the Limits of Cyber Warfare*, Security Studies 22 (3) (2013) 365–404.

[4] J.P. Farwell, R. Rohozinski, *Stuxnet and the Future of Cyber War*, Survival (Lond) 53 (1) (2011) 23–40.

[5] E.E. Miciolino, R. Setola, G. Bernieri, S. Panzieri, F. Pascucci, M.M. Polycarpou, *Fault Diagnosis and Network Anomaly Detection in Water Infrastructures*, IEEE Design & Tesçt 34 (4) (2017) 44–51.

[6] J. Butts, S. Shenoi. Haider S, Li G, Wang K. A Dual Control Strategy for Power Sharing Improvement in Islanded Mode of AC Microgrid. Protect Control Mod Power System., 2018, 3, 1–8.

[7] Rebecca Shapiro, Sergey Bratus, Edmond Rogers, Sean Smith. Identifying Vulnerabilities in SCADA Systems via Fuzz-Testing. Critical Infrastructure Protection V - 5th IFIP WG 11.10 International Conference on Critical Infrastructure Protection, ICCIP 2011. , 2011, 57–72.

[8] Rebecca Shapiro, Sergey Bratus, Edmond Rogers, Sean Smith. Do-It-Yourself SCADA Vulnerability Testing with LZFuzz. , 2013

[9] Rebecca Shapiro, Sergey Bratus, Edmond Rogers, Sean Smith. *Identifying Vulnerabilities in SCADA Systems via Fuzz-Testing.* , 2017

[10] Sergey Bratus, Anna Shubina. *LZfuzz: A Fast Compression-Based Fuzzer for Poorly Documented Protocol*s. , 2008

[11] Sergey Bratus, Anna Shubina. *A General Approach for Modules Identification in Evolving Networks*. Thang N. Dinh, Incheol Shin, Nhi K. Thai, My T. Thai. Open Systems and Information Dynamics, 2010, 40, 83–100.

[12] J. Ziv and A. Lempel. Compression of Individual Sequences via Variable Length Coding. IEEE Transaction on Information Theory, 1978, 24(5), 530–536.

[13] J. Ziv and A. Lempel. *A Universal Algorithm for Sequential Data Compression*. IEEE Transaction on Information Theory, 1977, 23(3), 337–343.