Toward Automatic Detection of Cloud Server Security Vulnerabilities

Olufogorehan Tunde-Onadele
oatundeo@ncsu.edu
North Carolina State University
Raleigh, North Carolina

Yuhang Lin
ylin34@ncsu.edu
North Carolina State University
Raleigh, North Carolina

Xiaohui Gu
xgu@ncsu.edu
North Carolina State University
Raleigh, North Carolina

Jingzhu He
hejzh1@shanghaitech.edu.cn
ShanghaiTech University
Shanghai, China

ABSTRACT

Cloud systems have been widely adopted in many real world production applications. Thus, security vulnerabilities in those cloud systems can cause serious widespread impact. Although previous intrusion detection systems can detect security attacks, understanding the underlying software defects that cause those security vulnerabilities is little studied. In this work, we conduct a systematic study over 109 software security vulnerabilities in 13 popular cloud server systems. To understand the underlying vulnerabilities, we answer the following questions: 1) what are the root causes of those security vulnerabilities? 2) what threat impact do those vulnerable code have? 3) how do developers patch those vulnerable code? Our results show that the root causes of the studied security vulnerabilities comprise five common categories: 1) *improper execution restrictions*, 2) *improper permissions checks*, 3) *improper resource path-name checks*, 4) *improper sensitive data handling*, and 5) *improper synchronization handling*. We further extract principal code patterns from those common root causes.

CCS CONCEPTS

• **Security and privacy → Virtualization and security**.

KEYWORDS

Cloud Security, Vulnerability Detection, Bug Study

1 INTRODUCTION

Cloud server systems provide a convenient platform for deploying web applications. Cloud users and organizations focus on building production software while relying on cloud providers to manage the hardware that hosts their applications. Cloud users can also deliver their services to many customers using virtual machine and container technologies that scale to meet varying levels of demand. Hence, cloud server systems have become popular for production applications. However, cloud environments are vulnerable to security attacks, which can have extensive impact on the end users. For instance, the credit reporting agency, Equifax, suffered a data breach in 2017 that affected over 147 million customers and cost the firm about $650 million in claim settlements [1, 2]. Production applications consist of many components that need to be secure. For example, applications have subsystems that process sensitive user information, such as credentials, that need to be confidential. In addition, applications have powerful execution functionality to evaluate user requests, which must be protected from running malicious commands. Security vulnerabilities in any component exposes the whole application to security attacks.

Cloud security has become increasingly important for many real world critical applications. In response to security risks, previous work proposes various intrusion detection systems to meet the resource constraints and dynamic workload challenges in cloud environments [3–5]. These approaches inspect system telemetry data such as network metrics or system logs to identify abnormal attack behavior. However, previous work does not provide information about the underlying software code problems that are needed by the developer for fixing the security vulnerabilities. To mitigate those security vulnerabilities, developers have to manually analyze massive code base to figure out the root causes. In this work, we use *security bugs* to refer to software issues with underlying code defects that allow unapproved access to privileged resources in cloud server systems. Our work aims at making the first step to understand the common root cause patterns among many real world security bugs, which can provide foundations for developing automatic security bug detection and patching schemes.

In this work, we investigate 109 recent security bugs mainly selected from over 300 CVEs in the past five years in 13 popular cloud server systems. We categorize the vulnerabilities by answering the following questions: 1) what are the root causes of the security bugs? 2) what threat impact do the security bugs have? 3) how do developers patch the security bugs?

Specifically, we make the following contributions:

- We identify five common root cause patterns by systematically analyzing 109 security bugs: 1) *improper execution restrictions* due to inadequate restrictions of functions that execute commands, 2) *improper permissions checks* due to insufficient checks for parameter to privileged functions, 3) *improper resource path-name checks* due to incomplete checks for requested resource paths, 4) *improper sensitive data handling* due to improper protection of sensitive data revealed in program output, and 5) *improper synchronization handling* due to errors in functions that handle concurrent requests.
- Our study shows that the leading root causes of the security bugs are improper execution restrictions (37%), improper permissions checks (25%), and improper resource path-name

checks (24%). The remaining bugs are due to improper sensitive data handling (7%) and improper synchronization handling (7%).

REFERENCES

[1] 2020. Equifax Data Breach Settlement. https://www.ftc.gov/enforcement/cases-proceedings/refunds/equifax-data-breach-settlement

[2] Stacy Cowley. 2019. Equifax to pay at least $650 million in largest-ever data breach settlement. https://www.nytimes.com/2019/07/22/business/equifax-settlement.html

[3] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1285–1298.

[4] Yuhang Lin, Olufogorehan Tunde-Onadele, and Xiaohui Gu. 2020. CDL: Classified Distributed Learning for Detecting Security Attacks in Containerized Applications. In *Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 179–188.

[5] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. 2013. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 199–208.