OPatch: Beyond Patch Tuesday for Containerized Applications

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

Jingzhu He
jhe16@ncsu.edu
North Carolina State University
Raleigh, North Carolina

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

ABSTRACT

Containers have become increasingly popular in distributed computing environments. However, recent studies have shown that containerized applications are susceptible to various security attacks. Traditional pre-scheduled software update approaches not only become ineffective under dynamic container environments but also impose high overhead to containers. In this paper, we present OPatch, a new on-demand targeted patching framework for applications running inside container-based environments. OPatch provides dynamic vulnerability exploit identification using unsupervised neural network techniques and triggers targeted vulnerability patching over victim containers upon attack detection. We evaluated our schemes over 31 real world security vulnerability exploits in 23 commonly used server applications. Results show that OPatch can accurately detect and classify 81% vulnerability exploits and reduce memory overhead by up to 84% and disk overhead by up to 40%.

CCS CONCEPTS

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

KEYWORDS

Container Security, Anomaly Detection, Security Patching

1 INTRODUCTION

Containers have become increasingly popular in distributed computing environments by providing an efficient and lightweight deployment method for various applications. However, recent studies [2][4] have shown that containers are prone to various security attacks, which has become one of the top concerns for users to fully adopt container technology [1].

Containerized applications pose a set of new security challenges to distributed computing environments such as computing clouds and data centers. First, container image repositories are prone to vulnerabilities. Indeed, previous study [4] reveals an alarming degree of vulnerability exposure and spread in the official Docker Hub container repository. It is complex to maintain a public or private container repository which often consists of a large number of container images and many inheritance layers. If a container is created from a base image, any vulnerability detected in the base image needs to be patched in the containers that are built on top of the base image. Second, containers are often allocated with limited resources because a large number of containers often share the resources of a single physical host. Security patching might cause significant resource increase (e.g., memory bloating) in a patched container, which makes the container unable to run after patching.

Existing security patching schemes in distributed computing environments often follow a scheduled whole upgrade approach, that is, updating all applications on a certain day (e.g., every Tuesday). The approach works well in stable distributed computing environments consisting of long running hosts or virtual machines. However, containers are often short-lived, which makes periodical patching schemes ineffective if the vulnerable containers miss the pre-scheduled patching day. Moreover, general software upgrade often significantly increases the memory footprint of the patched containers. As a result, those containers quickly become too heavy to fit in constrained resource allocations.

In this research work, we present OPatch, an on-demand targeted patching framework to achieve effective and low-cost security attack protection for containerized applications. Our framework consists of three key components: 1) an online *anomaly detection* module which can catch vulnerability exploit behavior using low-cost, non-intrusive system call tracing and unsupervised machine learning algorithm: autoencoder neural network [3]; 2) a *signature extraction* module which extracts the most frequently appeared system calls during the exploit period to create a signature for the vulnerability exploit and use it to map to a culprit vulnerability with Common Vulnerabilities and Exposure (CVE) identifiers based on profiled results; and 3) a *targeted patch execution* module which is responsible for triggering the proper software library update to fix the vulnerability. We adopt a hybrid approach to achieve targeted patching by combining package manager tools such as Advanced Package Tool (APT) for Linux distributions and manual package installations.

Specifically, this work makes the following contributions.

- We propose a new on-demand targeted patching framework to achieve practical and effective security protection for applications running inside the container environments.
- We present a light-weight vulnerability exploit detection and signature extraction scheme using out-of-box system call tracing and unsupervised machine learning method autoencoder neural networks.
- We have implemented a prototype of OPatch and evaluated it over 31 real world security vulnerability exploits in 23 commonly used server applications.

Our experimental results show that OPatch's dynamic exploit detection scheme can successfully detect and classify 81% vulnerability exploits with 16.38 seconds lead time on average. In comparison, other commonly used anomaly detection schemes such as k-nearest neighbor (k-NN) and k-means clustering algorithm can only detect 6% and 68% exploits, respectively. k-means also produce 7% false alarms while OPatch only incurs 0.7% false alarms. Among those successfully detected exploits that attack different applications, OPatch can achieve 100% classification accuracy by extracting unique signature for each vulnerability. For attacks targeting different vulnerabilities within the same application, OPatch accurately classify different vulnerabilities for 22 out of 23 tested applications with only one exception (i.e., Ghostscript). We further found that for those Ghostscript vulnerabilities that share common signature can be in fact corrected by the same code patch. To quantify the benefit of OPatch, we compare the memory and disk footprint change before and after patching between OPatch and the existing version-based software upgrade approach. Our results show that our patching scheme can reduce the memory footprint increase (caused by the applied patches) by up to 84% and disk size increase by up to 40%.

REFERENCES

[1] Anthony Bettini. 2015. Vulnerability Exploitation in Docker Container Environments. https://www.blackhat.com/docs/eu-15/materials/eu-15-Bettini-Vulnerability-Exploitation-In-Docker-Container-Environments-wp.pdf.

[2] Docker Image Vulnerability Research. 2017. https://www.federacy.com/docker_image_vulnerabilities.

[3] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[4] Rui Shu, Xiaohui Gu, and William Enck. 2017. A Study of Security Vulnerabilities on Docker Hub. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy.* ACM, 269–280.