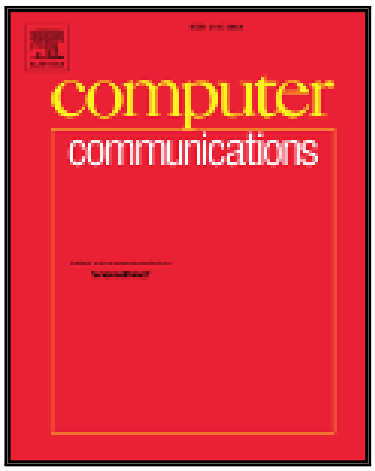

Docker ecosystem – Vulnerability Analysis

A. Martin, S. Raponi, T. Combe, R. Di Pietro

PII: S0140-3664(17)30095-6
DOI: 10.1016/j.comcom.2018.03.011
Reference: COMCOM 5670

To appear in: *Computer Communications*

Received date: 21 January 2017
Revised date: 23 January 2018
Accepted date: 13 March 2018

Please cite this article as: A. Martin, S. Raponi, T. Combe, R. Di Pietro, Docker ecosystem – Vulnerability Analysis, *Computer Communications* (2018), doi: 10.1016/j.comcom.2018.03.011

Docker ecosystem – Vulnerability Analysis

A. Martin[a], S. Raponi[b], T. Combe[a], R. Di Pietro[b,]

[a]*Nokia Bell Labs, 1, route de Villejust, 91620 Nozay, France*
[b]*HBKU-CSE, Education City, Doha, Qatar*

Abstract

Cloud based infrastructures have typically leveraged virtualization. However, the need for always shorter development cycles, continuous delivery and cost savings in infrastructures, led to the rise of containers. Indeed, containers provide faster deployment than virtual machines and near-native performance. In this paper, we study the security implications of the use of containers in typical use-cases, through a vulnerability-oriented analysis of the Docker ecosystem. Indeed, among all container solutions, Docker is currently leading the market. More than a container solution, it is a complete packaging and software delivery tool. In this paper we provide several contributions: we first provide a thorough survey on related work in the area, organizing them in security-driven categories, and later we perform an analysis of the containers security ecosystem. In particular, using a top-down approach, we identify in the different components of the Docker environment several vulnerabilities—present by design or introduced by some original use-cases. Moreover, we detail real world scenarios where these vulnerabilities could be exploited, propose possible fixes, and, finally discuss the adoption of Docker by PaaS providers.

Keywords:
Security, Containers, Docker, Virtual Machines, DevOps, Orchestration.

1. Introduction

Virtualization-rooted cloud computing is a mature market. There are both commercial and Open Source driven solutions. For the former ones, one may mention Amazon's Elastic Compute Cloud (EC2) [1], Google Compute Engine [2] [3], VMware's vCloud Air, Microsoft's Azure, while for the latter ones examples include OpenStack combined with virtualization technologies such as KVM or Xen.

Recent developments have set the focus on two main directions. First, the acceleration of the development cycle (agile methods and *devops*) and the increase in complexity of the application stack (mostly web services and their frameworks) trigger the need for a fast, easy-to-use way of pushing code into production. Further, market pressure leads to the densification of applications on servers. This means running more applications per physical machine, which can only be achieved by reducing the infrastructure overhead.

In this context, new lightweight approaches such as containers or unikernels [4] become increasingly popular, being more flexible and more resource-efficient. Containers achieve their goal of efficiency by reducing the software overhead imposed by virtual machines (VM) [5] [6] [7], thanks to a tighter integration of guest applications into the host operating system (OS). However, this tighter integration also increases the attack surface, raising security concerns.

The existing work on container security [8] [9] [10] [11] focuses mainly on the relationship between the host and the container. This focus is completely justified by the fact that, while virtualization exposes well-defined resources to the guest system (virtual hardware resources), containers expose (with restrictions) the host's resources (e.g., IPC / file-system) to the applications. However, the latter feature represents a threat to both confidentiality and availability of applications running on the same host.

Containers are now part of a complex ecosystem - from container to various repositories and orchestrators - with a high level of automation. In particular, container solutions embed automated deployment chains [12] meant to speed up code deployment processes. These deployment chains are often composed of third parties elements, running on different platforms from different providers, raising concerns about code integrity. This can introduce multiple vulnerabilities that an adversary can exploit to penetrate the system. To the best of our knowledge, while deployment chains are fundamental for the adoption of containers, the security of their ecosystem has not been fully investigated yet.

The vulnerabilities we consider are classified, relatively to a hosting production system, from the most remote ones to the most local ones, using Docker as a case study. We actually focus on Docker's ecosystem for three rea-

Email addresses: antonymartin.pro@gmail.com, antony.martin@nokia.com (A. Martin), sraponi@hbku.edu.qa (S. Raponi), theo-nokia@sutell.fr (T. Combe), rdipietro@hbku.edu.qa (R. Di Pietro)