SECURITY CONCERNS ON THE ADOPTION OF SOFTWARE CONTAINERS

Sergiu EFTIMIE[1]
Violeta OPRIS[2]

[1]Inf. Ph.D. Student, Military Technical Academy - Electronic, Information and Communication Systems for Defense and Security Doctoral School
[2]Inf.Ph.D. Student, Military Technical Academy - Electronic, Information and Communication Systems for Defense and Security Doctoral School

Abstract: *In this paper we provide an overview of the present security concerns on the adoption of software containers by enterprise organizations. The proven benefits of containers such as application scalability and faster time to market can be overshadowed by security issues. Although the container design is considered secure, the detection and mitigation of vulnerabilities should be part of a strong security strategy in the development of an application.*

Keywords: *Containers, Security, Enterprise adoption, Application deployment*

Introduction

The cloud revolution and the common goal of the service providers to increase elasticity and density in data centers has led to the development of containerization, currently the most dense and flexible virtualization technology to support cloud environments. Organizations are starting to adopt containerization because of its proven benefits such as application scalability and faster time to market. Containers also are faster and more agile compared to virtual machines. Despite this ongoing adoption, security properties of containers remain an unexplored field. In this paper, we will analyze the security properties of containers along four main areas: Resource isolation, secure administration and management, support for common security controls and secure operations management**.**

Resource isolation

Containerization adds new security challenges to the enterprise landscape. Software containers combine a lightweight application isolation with a deployment method based on images thus keeping together applications and their runtime components. By packaging apps along with the libraries and the other binaries on which they depend, software containers add autonomy to applications. The conflicts between apps and the components of the underlying host operating system are thus avoided. Container platforms such as the ones provided Docker and Red Hat use services provided by the Linux kernel in order to obtain container isolation. Compromised or malicious containers attacking other containers that are running on the same system represent a major threat.

"The security and isolation of the containers is correctly perceived as the most critical point for container security"[1].

Given the security concerns around containerization, providers are moving their strategy towards a customer reassurance about container security. The present focus is on the use of encryption to secure the code to protect users from backdoors included in shared application images.

This method, used by Docker (Docker Content Trust) has received many critics because it covers only one aspect of container security, without taking into consideration the fact that exploitable versions of open source code could exist within software stacks and applications.

New vulnerabilities that impact older versions of open source software are being constantly discovered. Container providers need an open source technology that has the ability to provide users with an informed image regarding the risks.

The security risk posed by a software container depends on the location of where the container is deployed and on the classification of the data accessed by it.

For example, a publicly available attack will subject containers to a range of threats, from SQL injection to denial-of–service attacks. An internal network would limit the exposure to such attack thus limiting the risk.

The creation of a robust process for determining what software is deployed along with an application, the location of the software, and the possible security vulnerabilities is critical for the container industry.

Enterprises are adopting software containers because of their benefits: application scalability, reduced number of deployment errors, a faster time to market and a simplified management but containers have also reached a point where security concerns could inhibit further adoption.

DOI: 10.21279/1454-864X-16-I2-061

Looking at the past, the adoption of virtualization technologies was done before the establishment of strong security requirements. Industry analysts have different opinions regarding the similarity of the situation with the present container adoption.

The presence of software vulnerabilitiesis inevitable. A Gartner analysis [2]on the properties of containers managed by Docker has showed that as enterprises continue to evaluate containerization for continuous deployment, security concerns are still appearing in the landscape.

The detection and remediation of vulnerabilities should be seen as an imperative part of a strong strategy towards application security. Enterprises can overcome these concerns by taking advantage of the different automated tools available to regain control over the elements of their infrastructure.

Docker uses a feature called name spacing in order to improve security. Namespaces represent a way to isolate areas like process space and network. This feature enables applications to run in containers without having root permission. Having user namespaces allows the Docker service to run as root user and to handle containers separately. Each container has user privileges and does not need root privileges to run.

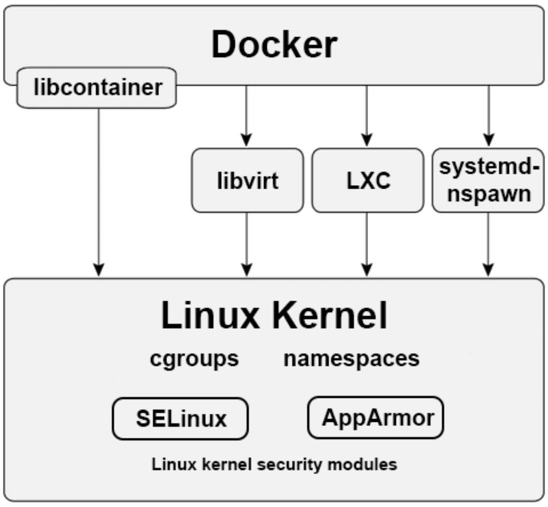

Fig. 1. Docker and the Linux kernel

As Linux and Microsoft have started to provide extensive support for virtualization in their kernels, virtualization security has steadily migrated into the host operating system. In [Fig. 1] we can observe the relationship between the Linux container (LXC) and the users pace file systems. This adds security to containers at the host OS level.

Container systems have a smaller attack surface than traditional virtualization systems. Containers consolidate shared resources and in consequence there will be fewer processes to manage and fewer versions of resources to attack. A smaller threat surface leads to an improved security posture.

Secure Administration and Management

Red Hat's container strategy [3] addresses directly security and certification concerns and provides container management capabilities that are robust regardless of the container deployment platform.

Docker lacks some features on the management side. According to a Gartner research paper[2]"*They disappoint when it comes to secure administration and management, and in support for common controls for confidentiality, integrity and availability"*

The analysis recommends that enterprises that don't need a virtual private system emulation should standardize on the nsenter API for interacting with running containers, and protect it by limiting the input set. There is also a recommendation that organizations should select a framework for managing resources and deploying containers at scale, and should consider using secure wrappers (SSL/TLS) for the Docker/Swarm API for an increased integrity and confidentiality.

There is also a lack of endpoint protection platform and encryption tools for Docker containers.

Organizations have to mitigate risks by using application white listing, SELinux or a strategic DevOps automation tool to secure containers.

Support for Common Security Controls

The use of containerization has increased the need for encryption and container security. Large volumes of sensitive data reside in dynamic containers and this leads to a critical need to implement strong security controls.

Data-at-rest in containers is susceptible to many of the common security threats found in traditional data centers, such as insider threats or cyber-attacks. Containers are also easy to replicate, which can lead to an uncontrolled expansion. In a similar way, the sensitive data produced by the applications that use containers also needs a tight control.

Automation and orchestration solutions will have an increased importance to the effective management of large-scale cloud computing [4].

From a workload mobility point of view, containers have a potential to change the architecture of cloud computing, by removing decisions taken by orchestration tasks. Instructions can be carried inside the container format, and can trigger a highly automated system that could build a logical server from them.

DOI: 10.21279/1454-864X-16-I2-061

External data loss prevention and protection capabilities are needed in order to keep container environments (both applications and data) secure.
These capabilities include automation, platform and location independence, transparency and storage choices.
Like any data source, applications that are built on containers or their data can be lost, corrupted, or modified. The roll back ability depends on the data protection plan put in place for containers.

Secure Operations Management

Docker has an effective solution for resource isolation and has advanced initiatives in secure operations management and configuration governance with hypervisors and the Linux operating system[5]. Docker Trusted Registry is a tool that allows the storage and management of container images in virtual private clouds or on premise to meet security and regulatory compliance requirements.

Docker has developed Docker Data Center, a tool that enterprises canuse to deploy a CaaS (Container as a Service)infrastructure in a private cloud or on-premises. A CaaS will provide a secure environment where developers can deploy applications in a self-service manner.
The Docker Datacenter provides tools requiredby organizations to manage the application lifecycle of applications built on containers. The Docker Universal Control Plane is a tool that provides a virtual private cloud or an on-premises container management solution.
Docker Trusted Registry allows the storage and management of container images in virtual private clouds or on premise in order to support security and regulatory compliance requirements. This tool can be installed through a web administration console, and can integrate important software development workflows like continuous integration and continuous delivery.

CONCLUSIONS

Enterprise interest in containers is growing. Organizations have moved over the years from viewing open source as a curious novelty to the appreciation of its business advantages. In the present days there is a question whether containers will reach an acceptable level of security for enterprise use. Container providers are in a race to create best practices and tools to enable organizations to develop the right skill sets in order to leverage the big promises of software containers.
Although the container design is considered secure, the detection and mitigation of vulnerabilities should be part of a strong security strategy in the development of an application.
According to Docker [5] there are four major areas to consider when reviewing container security:

1. The intrinsic security of the kernel and its support for namespaces and control groups
2. The attack surface of the Docker daemon
3. Loopholes in the configuration profile of the container
4. Hardening security features of the kernel and their interaction with containers.

The assurance that a container does not present vulnerabilities at the time of the initial build and deployment is necessary, but it is not sufficient. By excluding the presence of exploitable open source code in the software stacks and application portfolios we only ensure that the container images are the exact copies of the ones provided by developers. Therefore, a critical element of container security is the ability to see and evaluate the code inside containers.
The deployment location of a container will affect the associated level of risk. A container that is deployed on an internal framework will have a smaller level of risk than an internet-facing application.
The adoption of containers resembles the early adoption of virtualization technologies that was done by enterprises before the establishment of some strong security mechanisms. Industry analysts differ in opinions regarding this similarity. Some analysts predict that until security standards are identified and established, enterprises will not rush the adoption of containers.
Fundamentally, container security can be seen as equivalent to the hypervisor security. Although container providers such as Docker and Red Hat are not as mature as VMware for example. As container security matures we can envision how the reduced threat surface offered by containers may lead to fewer vulnerabilities than those exposed by virtualized environments

BIBLIOGRAPHY

[1] Enrico Bacis, Simone Mutti, Steven Capelli, Stefano Paraboschi – *Docker Policy Modules: Mandatory Access Control for Docker Containers,* IEEE CNS 2015 Poster Session, 2015
[2] Joerg Fritsch -*Security Properties of Containers Managed by Docker*, Gartner Research, 2015
[3] http://www.red-hat.com, accessed April 2016
[4]Andrea Tosatto, Pietro Ruiu, Antonio Attanasio - *Container-based orchestration in cloud: state of the art and challenges,* 9thInternational Conference on Complex, Intelligent, and Software Intensive Systems, 2015
[5] https://docs.docker.com/engine/security, accessed April 2016

DOI: 10.21279/1454-864X-16-I2-061