Shreyas S, 2021, 9:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Overview on the Security of Containerization

Shreyas S, Rohith S Yadav, S.G Raghavendra Prasad, B.K Srinivas
Department of Information Science and Engineering,
R.V College of Engineering, Bengaluru, India

Abstract- In the course of the most recent couple of years, the utilization of virtualization advances has expanded drastically. This makes the interest for effective and secure virtualization arrangements become more self-evident. Compartment based virtualization and hypervisor-based virtualization are two principle sorts of virtualization innovations that have arisen to the market. Of these two classes, compartment based virtualization can give a more lightweight and productive virtual climate, yet not without security concerns. In this paper, we dissect the security level of Docker, a notable agent of compartment based approaches. The examination thinks about two regions: (1) the inner security of Docker, and (2) how Docker interfaces with the security highlights of the Linux part, like SE Linux and App Armor, to solidify the host framework. Moreover, the paper likewise talks about and recognizes what should be possible when utilizing Docker to build its degree of safety.

Keywords:- Docker, host framework , etc.

I. INTRODUCTION

The last decade has seen a blast of improvement in the space of virtualization advances, which permit the dividing of a PC framework into numerous segregated virtual conditions. The advancements offer generous advantages that have been driving their advancement quickly. Quite possibly the most regular explanations behind embracing virtualization advances is worker virtualization in server farms.

With worker virtualization, a director can make at least one virtual framework occasion on a solitary worker. These virtual frameworks work as genuine actual workers and can be leased on a membership premise. Amazon EC2, Rackspace, and Netmagic are a few famous cases of such server farm specialist organizations. Another normal use is for work area virtualization, where one PC can run a few OS occurrences. Work area virtualization offers help for applications that can run uniquely on a explicit OS.

The development in the utilization of virtualization advancements advances the interest for a virtualization arrangement which can give thick, versatile, and secure client conditions. A enormous number of virtualization arrangements have arisen to the market. They can be characterized into two significant classes: holder based virtualization and hypervisor-based virtualization. Of these two classes, holder based virtualization can give a more lightweight and proficient virtual climate. It permits multiple times more virtual conditions to run on an actual worker contrasted with hypervisor-based virtualization. Be that as it may, compartment based virtualization additionally accompanies security concerns.

A notable agent of compartment based virtualization approach. We think about two regions: the interior security of Docker, and how Docker communicates with the security highlights of the Linux portion, like SELinux and AppArmor, to solidify the host framework.

The investigation analysed the inward security of Docker dependent fair and square of disconnection Docker can give to its virtual surroundings. The collaboration among Docker and the security highlights of the bit was assessed dependent on how the highlights are upheld by Docker. Supposedly, Docker is a moderately new innovation, and this is one of the principal examinations of this sort that emphasis on its security perspectives.

International Journal of Science, Engineering and Technology

An Open Access Journal

II. APPROACHES IN VIRTUALIZATION

A large portion of the virtualization advancements can be characterized into two significant methodologies:

- Compartment based virtualization and
- Hypervisor- based virtualization

The previous gives virtualization at the working framework level, while the last gives virtualization at the equipment level. Every one of the methodologies enjoys its own benefits and impediments, which are depicted in this segment.

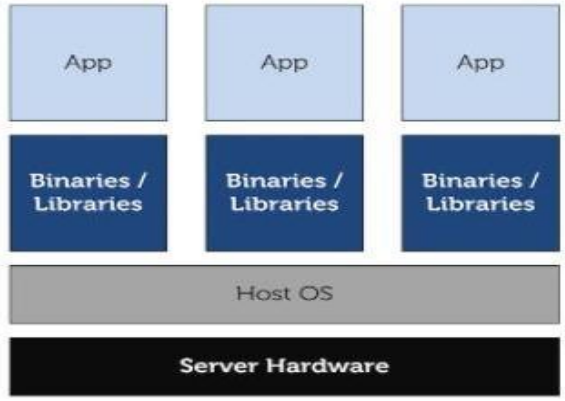

Fig 1. Container-based Virtualization Architecture.

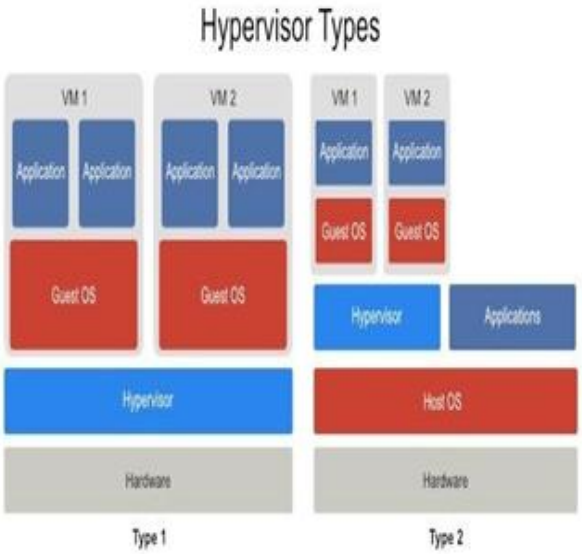

Fig 2. Hypervisor based Virtualization Type 1 and Type 2.

1. Type 1 Hypervisor:

In this case, the Hypervisor is a standalone machine with its own operating system installed on its own hardware, which produces a virtualization layer on which virtual machines with multiple operating systems such as Windows, OS X, or Linux can be formed. This is demonstrated using the VM Ware ESXI server. The Linux kernel is the primary virtual machine in the ESXi Hypervisor model, and it is started by the service console. The vmkernel runs on the bare computer during normal operation, while the Linux-based service console runs as the first virtual machine.

2. Type 2 Hypervisor:

The type 2 hypervisor, often known as embedded hypervisor, is the most often used hypervisor. The hypervisor has been popular among individuals and small businesses building software or servers since early 2017. The virtual machine is a piece of software that runs on top of a host machine and has its own operating system. So, there's a Host Machine with its own operating system, on which a VM Software with its own virtual operating system is loaded, and on top of that, a new set of virtual machines are installed.

III . DOCKER OVERVIEW

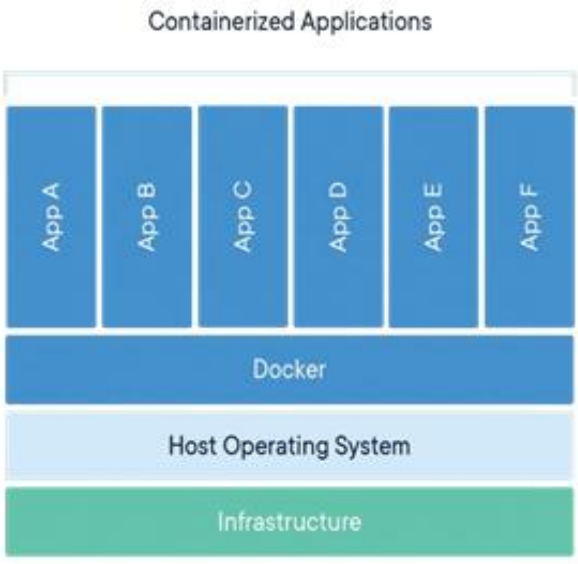

Fig 3. Architecture of Docker.

Docker is an open source container technology for "building, shipping and also for running distributed applications". It has been used in a number of well-known applications, including Spotify, Ebay and Yelp.

Docker, a relatively new candidate, is currently one of the most successful technologies considering container technologies have been around for more than a decade. It has new capabilities that previous technology lacked. It provides APIs for quickly and safely creating container management and controls

International Journal of Science, Engineering and Technology

An Open Access Journal

the container. Developers can package apps into lightweight Docker containers that can run on practically any platform without needing to be modified. Furthermore, Docker has the ability to deploy more virtual environments than VirtualBox.

On the same hardware, other technologies can be used. Last but not least, Docker works nicely with third-party applications. It makes the process of administration and deployment of Containers created with Docker.

IV. CONTAINER ORCHESTRATION OVERVIEW

The automated arrangement, coordination, and management of computer systems, middleware, and services is known as orchestration. While Docker established an open standard for bundling and dispersing containerized apps, another challenge arose. What would be the best way to build and plan these containers? How do the many stakeholders in your application communicate with one another?

What methods are there for scaling container occurrences? Before long, there were solutions for organising containers. Kubernetes, Mesos, and Docker Swarm are a few of the more well-known options for influencing a number of machines to function as if they were one large system, which is critical in a large-scale situation.

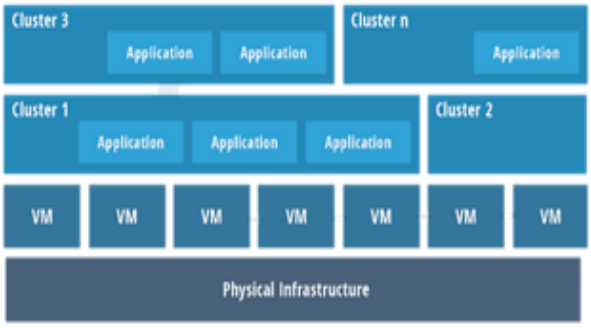

Fig 4. Kubernetes Architecture.

V. PROBLEM STATEMENT

As Docker is more prevalent in the industry than a Hypervisor or a Virtual Machine, more secure measures are required. According to a poll conducted by "RightScale," 49 percent of users have used docker- based cloud architecture in 2018, up from 35 percent in 2017. With such a huge growth in users, it's important to figure out what the security risks are and measures required to nullify the security issues of docker and Kubernetes.

VI. SECURITY THREATS AND ITS SOLUTION

1. IPC Isolation:

IPC isolation is achieved by Docker via IPC namespaces, which allow for the establishment of independent IPC namespaces.An IPC namespace's processes are unable to read or write. IPC resources should be written in other IPC namespaces. Each container is given an IPC namespace, which prevents the processes in a container from interfering with the processes in another container.

2. Isolation of the network:

To prevent network-based attacks like Man-in-the-Middle and ARP spoofing, network isolation is critical.Containers must be set up in such a way that they can work together.they are unable to listen in on network traffic or influence it neither the host nor the other containers.Docker builds a separate networking network for each container by utilising network namespaces.

3. Resource Limiting:

DoS attack is common in the system which is Multitenant, where a group of processes will accumulate all resources and leads to disrupting the normal working of the process. Hence Cgroups are the main component which docker uses in order to tackle this issue by controlling the amount of resources.

The Center for Internet Security (CIS) issued a number of recommendations for hardening Kubernetes or Docker containers. For example, enabling built-in Linux security mechanisms such as SELinux and Seccomp profiles is one of the recommended practises. SELinux is a kernel-level capability for controlling file and network access. whereas Seccomp profiles limit the number of system calls a programme can make. These capabilities, when combined, provide a fine- grained level of control over the workloads that operate on the node. (According to The New Stack, 2018). Real considerations of hub security, on the whole, include:

International Journal of Science, Engineering and Technology

An Open Access Journal

- Anchoring hub communications with TLS customer authentication to ensure that all fundamental Apus passageways are secured end-to-end.
- Enabling part-level security measures such as SELinux or Seccomp. These capabilities aid in limiting the attack surface on the hub, allowing for more control over the overall security of the system.
- Restricting direct access to Kubernetes hubs, such as Secure Shell (SSH) access: Forcing all hub access through Kubernetes ensures adequate access control and logging. This reduces the risk of unapproved access to assets.
- Use industry best practises to construct and solidify the Linux hubs that run compartments, such as the CIS Docker Benchmark.

4. Kubernetes Security Isuues and solution:
The primary problem with container orchestrator is that it relies on a single container to run the entire cluster. As a result, if one container is compromised, the entire cluster is destroyed.

The default behaviour of many Kubernetes clusters (where a token providing access to the Kubernetes API mounts into each container) can cause security issues mainly when token have the administrator right. So attacker with control over master node will have control over all the worker node, hence RBAC needs to be configured Securely.

5. Workload Configuration:
The configuration for deploying your apps in Kubernetes is often done in code, whether using Kubernetes YAML, Helm Charts, or templating tools. This code has an impact on the Kubernetes security controls, which control how a workload operates and what can and cannot happen in the event of a breach.

6. Kubernetes Networking:
When it comes to Kubernetes, network security is crucial. Pod communications, ingress, egress, service discovery, and, if necessary, service meshes (such as Istio) should all be considered. Every service and machine in the network is at danger if a cluster is infiltrated.

As a result, it's critical to make sure your services and communication are limited to what's required. This, paired with the use of cryptography to keep your computers and services private, can help contain the threat and prevent a large- scale network breach.

7. Security of Kubernetes infrastructure:
Particularly the master nodes, databases, and certificates—is critical since it is a distributed programme that runs across many servers (using physical or virtual networking and storage). If a hostile actor is successful in breaching your infrastructure, they will have complete access to your cluster and applications.

VII. CONLCUSION

Compared to the hypervisor - based virtualization, container-based virtualization can deliver higher density virtual environments and greater performance. However, it is believed that the latter is more secure than the former.

In this work, we investigated Docker, one of the most widely used container-based virtualization systems, to see how secure its containers are. Even with the default settings, Docker containers are fairly safe. Docker containers' security can be improved by running them as "non-privileged" and enabling additional hardening options in the Linux kernel, such as AppArmor or SELinux.

Following this report, further research could compare Docker container security to that of alternative containerization platforms or virtual machines. Such research could lead to a more in-depth static examination of Docker or a larger understanding of container security in general.

REFERENCES

[1] Container Security Considerations in a K8's Deployment. https://thenewstack.io/container-security-considerations-kubernetes-deployment/
[2] Lanzi, A., Balzarotti, D., & Kirda, E. Hypervisor-based malware protection with AccessMiner. Computers & Security. doi:10.1016/j.cose.2015. 03.007.
[3] Omernik, J., Bertucci, Ali, Z., & Constantin, L. (2018, December 04).Vulnerability Uncovered In Kubernetes. Retrieved from https://securitybo ulevard.com/2018/12/vulnerabil ity-uncovered-in-kubernetes/

International Journal of Science, Engineering and Technology

An Open Access Journal

[4] D. J. Walsh. Are docker containers really secured?http://opensource.com/business/14/7/doc ker-security-selinux.

[5] D. J. Walsh. Bringing new security features to docker.https://opensource.com/business/14/9/security-for-docker..

[6] Containers & docker: How secure are they?https://blog.docker.com/2013/08/containers- docker-how-secure-are-they.

[7] McCune, R. (2018, December 05). A hacker's guide to Kubernetes security. Retrieved December 8, 2018, from https://techbeon.com/hackers-guide- kubernetessecurity.