Static Vulnerability Analysis of Docker Images

Static Vulnerability Analysis of Docker Images

Vipin Jain[1,2], Dr. Baldev Singh[2], Medha Khenwar[1], Milind Sharma[1]
Swami Keshvanand Institute of Technology, Management & Gramothan, (INDIA)
Vivekananda Global University, Jaipur-303012 (INDIA)

ervipin.skit@gmail.com

Abstract: Many organizations are renovating their businesses by grasping DevOps, microservices, and container technologies. Docker is emerged as a new technology, proving an efficient means to develop and deploy applications. Docker containers are created by images to run an application with all its dependencies so that it could run isolated from other processes. Security is always being a foremost concern as our industries are already persistent to improve the reliability and efficiency of new software applications. Security of local Docker containers from the attacks of malicious containers, perceived threats present in Docker images need to be detected and the risks identified when instances of Docker containers run on the host machine. This paper reviews Docker's existing security mechanisms, vulnerabilities, threats and the related tools required for static security analysis.

Keywords: Docker Security; Vulnerability; Container.

1. Introduction

Docker containers are already in demand because it allows the user to run multiple applications on the same host operating system. It also administers the bundle of operating system isolation and security features, but challenge arises when these Docker images are found to be insecure which may cause threats and other major security issues. They should be analyzed statically in order to overcome or render these container images more stable. The above strategy for the container is to be worked before the runtime of the container so that the attack of threats can be recovered without being detrimental to the software [1].

From security aspect, the Docker technology is relatively simple for container images. We need to give priority attention to two layers that can analyze containers in two different ways: Statically or Dynamically. We also usually do not need to concern more about APIs, as the overlay of networks or composite software-defined storage configurations, because these are not a major part of the analyzing portion, as the analysis is done after the container image is built i.e., before the execution of container.

2. Preliminaries

Many scholarly articles are available that are based on the security of Docker containers, but only a few pertaining directly to the static security analysis of Docker containers. Most of the security industries acknowledged that Docker container needs to be the subject of security and has the chances of dissipating. These security concerns are not to be neglected as it may be a big concern to optimize with the emergent of these attacks. This section gives a brief description of the vital terms that are going to be used throughout the paper.

2.1 Docker

The **"**Docker" was in the first instance, founded in March 2013, with the disclosure of a new approach called containerization that led to the virtualization at the OS level. Docker helps to provide a lightweight and quick environment. Docker is sometimes presented as superficial Virtual Machine (VM), but that is not the case. It is totally different from VM. The differences are shown in table 1.

Table 1: Differences between VMs and Docker containers

Virtual Machines	Docker container
It takes a couple of minutes to boot.	Boots in a few seconds.
VMs run using Hypervisor.	Docker makes the use of the Docker engine.
VM tools are straightforward to use and easy to work with.	Docker has a complicated usage mechanism consisting of both third party and Docker managed tools.

Docker gives the aptness to work with the infrastructure in a similar way that applications are handled, adding the image building instructions in "Dockerfile" and sustain a version control of the images. Furthermore, it provides the ability to increase or decrease the resources and scalability to run on different platforms. Containers are not a new abstraction but Docker technology makes its implementation much easier to use. Moreover, while in operation with images it is critical to understand that these images may contain vulnerabilities.

2.2 *Containers*

Docker containers are created by images to run an application with all its dependencies so that it could run isolated from other processes. Containers are hosted in a physical or virtual server on top of their operating system (OS). Containerization is a technology that integrates the application, system libraries, and related dependencies. Eventually, containerization uses host OS which serves relevant libraries and resources. Each container shares the host OS. This makes containers lightweight allowing them to be a few megabytes in size and take a few seconds to start. Containers do not have much management overhead, as they share a common operating system. That system may be patched and fixed if an issue arises [2]. Containers got its name from the shipping industry where different products are placed into compact shipping containers designed to minimize cost and time [3].

2.3 Security

The main concern with the container is security. There are major security issues surrounding containers, such as the security of the containers that are being run on, the content of images by unknown users. It is important to know that the content of container images must be correctly configured. Managing security in containerization is different from other traditional applications. The security challenges occur when an application goes to production. Its chances to be under threat may increase dramatically leading to multiplied user base which implies that it could be open to access from outside the organization [4]. Two types of security checking can be done- static security analysis and dynamic security analysis. Using both the security analysis techniques, it can be checked or confirmed if any of the container images have any bugs. According to the survey conducted in January 2015, 53% of the enterprises revealed that their considerable concern was security. This is because containers rely on the base image and images might contain vulnerabilities which can expand to all containers [5].

2.3.1 Docker Security Scanning

Security scanning has limited features as it can only contrast the software in an image to the familiar vulnerabilities and disclosure database for vulnerabilities [6]. Docker brings forth container images for the limited number of applications/operating systems. These images are of the finest quality for general

use but there may be some images that are already in use and are not provided officially by Docker which makes it the major security issue [7]. In the approach of Docker security scanning, there are many scanning techniques already available.

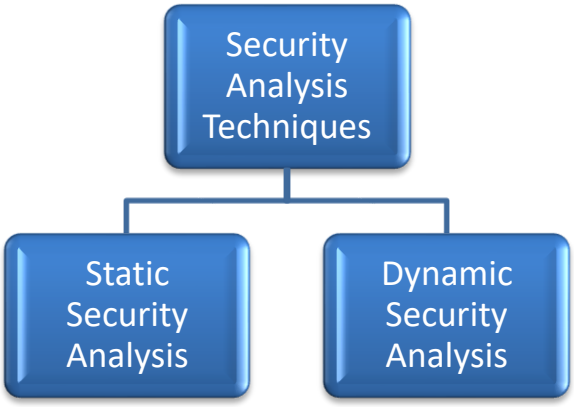

Figure 1: Types of Security Analysis Techniques

2.4 Image Security

Docker has more performance advantages than traditional virtual techniques [8]. There are few areas to consider when reviewing Docker image security like whether its packages are installed in the image or not. When it comes to security, the best choice for a base image is an empty container or distroless (set of images made by Google) or UBI provided by reputed vendors like redhat, IBM, that were created with the intent to be secure [9].

Docker images are- distributed, stored, and managed publically or they can be privately used to build containers. Official images are available on Docker hub used to build public or private processes. Docker images security is an unavoidable concern while creating an infrastructure [10].

- In Docker containers, there is a pre-definition of what is exactly running inside like the path of the data directories, daemon configurations, mount points, etc. A strong focus on security is a must, in this demanding world where the need is not only to make their system fast and reliable but also to make it secure. In 2018, the organizations that use container technology were facing the same issues over security and the percentage of those industries was 60% which is already a huge number that needs to be noticed and cannot be ignored.
- Not only those industries were having the issue of security but with this finding, hundreds of more organizations gave several statistics: They said around 47% of containers are having vulnerabilities and 46% of their developers accept that they had no idea if their containers had vulnerabilities or not.

Some threats and attacks that may cause vulnerabilities in container images and may target the container to fetch their confidential data and make the Docker container less secure are discussed in table 2 below.

Table 2: Docker Container Images Threats with their particular Solution

Threat	Attacks	Explanation	Solution
Cautious Images	Attacks on the containers and host operating system.	Cautious images may cause serious threats and harm the container.	To avoid these attacks, need to use trusted images or registries.
Vulnerable Images	Images may have inbuilt threats.	Some of the images have inbuilt threats that led to the corrupt all the data and damages the whole project.	These vulnerable images are required to follow the analysis process before executing in the container field.
Attack on images	Images that came from unknown sources.	These images may be used as threats and attacks on the container which may target confidential data.	Images need to be verified first either they are secured enough to use or not during build a container.

3. Methodology

In the last few years, various analysis techniques have been used for the security of images with the intention of attaining optimized solutions to get rid of these security issues. In this section, we evaluate these analysis techniques. The methodology adopted in this paper is the result of our work combined with the outcomes from the previous analysis techniques. To our knowledge, the most recent analysis of Docker images was performed by Socchi in 2019. They share the information about security measures introduced by Docker Inc., gives the information of verified and certified images that can improve the security of Docker hub. Besides, they discovered the distribution of all the vulnerabilities across repository types. They implement their software to analyze the image's security. Their conclusion says that the security measures do not improve overall Docker hub security [11].

The container would be able to configure the confidential data of other containers that target the integrity of the application and other information. Furthermore, these containers can also contain similar attacks to another semi-honest container which led another container to be targeted [15]. Our goal is to protect from those threats that may target the other containers and create errors in different ways. An appropriate way to protect those images from different threats and attacks would be to statically analyze the images so that no threats can be detected after the execution process.

For performing the static analysis of a particular number of images, we used an open-source Docker image scanner tool. In this section, we will compare all the analysis that has been already performed during these previous years to find out how the following analysis can be improved in most possible ways. Our findings can be summarized by the table that contains information about all the tools that analyze image security.

According to the surveys that are conducted during previous years, in order to identify all the possible solutions to decrease the vulnerability of images, it must be recognized that the scanning of images is one of the main possible solutions to save our containers from security issues.

4. Discussion

This section discusses the analysis of performance of the software that is already in use to scan Docker hub images or registries. Results are classified as shown in the table with their working environment. There are 10 open source software or tools that are already working to analyze all the trusted or untrusted images to check their vulnerabilities and enhance the security of Docker container images. They are shown in Table 3.

Table 3: Tools to Check Vulnerabilities in Images

Tools	Working	Result
Docker Bench for Security	A Tool to examine Docker container against security specification. It bases its tests on the industry-standard CIS benchmark.	The result conveys the information, pass logs, and warnings.
Clair	Built by CoreOS, performs static analysis of container vulnerability.	By using the Clair API, developers can ask the database for all the issues related to a specific image.
Cilium	Cilium is all about securing network connectivity. A Developer can apply the cilium security policies without making any changes in the application code or container configuration.	Cilium support is great to use, find extensive guides and documentation.
Anchore	A tool for inspecting container security using CVE data and user-defined policies.	It provides a list of vulnerabilities, threat levels, CVE identifiers, and other information.
OpenSCAPWork bench	An environment for developing and maintaining security policies for various platforms.	It allows multiple organizations to efficiently develop security content by avoiding redundancy.
Dagda	A tool for scanning for threats, attacks, vulnerabilities, and malware in Docker containers.	Developers can run it remotely or continuous, it shows the number of threats and other details.
Notary	Notary is a framework enhancing container security with a server of cryptographic responsibility.	It verifies the cryptographic integrity of Docker container images.
Grafes	An API to help and analyze internal security policies.	It helps speed remediation attempts.
Sysdig Falco	It serves behavioral activity monitoring with deep container visibility.	Sysdig provides further container troubleshooting materials.
Banyanops Collector	A framework for static analysis of Docker container images or registries.	It offers deep data analysis.

4.1 Container Image Authenticity

There are many Docker images and repositories on the internet and Docker hub which are doing all kind of useful work, but it comes out with complicated issues while pulling these images without any authenticity mechanism. The questions that relate to the authenticity of images are like:

- Where this image comes from?
- Does this image come from the trusted image creator?
- Does the cryptographic proof say that an author is a person?

- Does the image that has been pulled by you is secured enough to be used?

In any case, Docker allows pulling and running any image or registries by default. Even if the custom images are used during the process, it needs to make sure that nobody inside the organization is able to make changes in an image [4].

Container image authenticity directly implies the security issues of Docker images as the containers can be built by using the same Docker image from the Docker hub or repositories. While discussing the authenticity of Docker images, the first and easiest way to check the vulnerability of images can be done by scanning [12]. The various scanning tools have already been mentioned in table 3. This scanning process scans the vulnerabilities for Docker local images that run on the engine, those local Docker files and images provide users visibility into the security posture. Common vulnerabilities and exposures (CVE) database contain the list of all the vulnerability found during the scanning process [13].

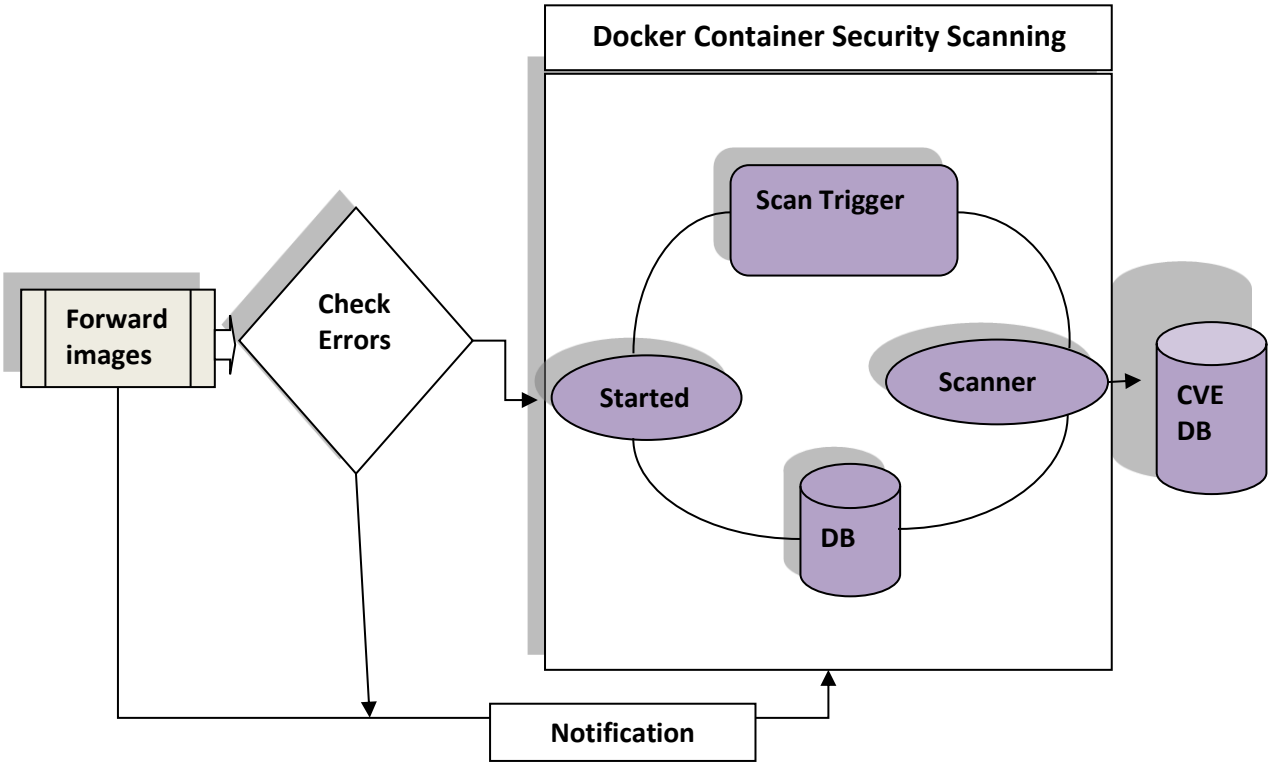

Figure 2: Docker Image Security Scanning

4.2 Vulnerability of Images

In this part, our focus is on the causes that make images more vulnerable as well as how these issues are solved with every other analyzing tool [14]. Below are the root causes behind the vulnerability of Docker images and repositories:

- Insecure generation of images.
- Un-trusted production of image cryptographic configuration.
- Possibility of errors in the image distribution, verification, storage process, and decompression.
- Vulnerabilities inside the images.
- Threats may be directly linked to the Docker or libcontainer.

It is important not to trust any of the images or repositories found on the internet. Docker already sponsors a team dedicated just to review and publish the images only in official repositories. In Docker hub 23% of images are tagged as the latest images and these images are already in the list of most downloaded images from the Docker hub, these images also contain a high amount of vulnerabilities

[16][17][18].

5. Conclusion

In this paper, a review on static security analysis of Docker images is presented. The comparative analysis is being done for security of Docker images using various tools. The security measures are also highlighted that are running with different tools and software created by various platforms to make the containers more secure and reliable. Moreover, creating a secure container images should be chosen prudently so that threats would not become a major concern for containers.

6. References

[1] A. Avram, “Docker: Automated and consistent software deploy-ments,” InfoQ. Retrieved, pp. 08–09, 2013.
[2] Abbott, Brendan Michael, "A Security Evaluation Methodology for Container Images" (2017), All Theses and Dissertations 6287, https://scholarsarchive.byu.edu/etd/6287
[3] Shetty, Jyoti. (2017). A State-of-Art Review of Docker Container Security Issues and Solutions. American International Journal of Research in Science, Technology, Engineering & Mathematics.
[4] D. Inc., “What is a container?” https://www.Docker.com/resources/ what-container, access date: 31. Jan 2020
[5] “Docker security,” https://docs.Docker.com/engine/security/ security/, access date: 15. Apr 2020.
[6] Y. Zhang, S. Wang, “Research on Docker Security”, Network Security Technology & Application, Aug., pp. 32-33, 2017.
[7] Sharma, Mahrishi, Hiran, Doshi, (2020), “ Reverse Engineering for potential Malware detection: Android APK Smali to Java″ Journal of Information Assurance & Security, Vol. 15 Issue 1, p26-34.
[8] “Cve-2019-9636 detail,” https://nvd.nist.gov/vuln/detail/ CVE-2019-9636, 2019, access date: 27. Mar 2020
[9] “Cve-2019-9948 detail,” https://nvd.nist.gov/vuln/detail/ CVE-2019-9948, 2019, access date: 27. Mar 2020.
[10] “Cve-2017-1000158 detail,” https://nvd.nist.gov/vuln/detail/ CVE-2017-1000158, 2019, access date: 27. Mar 2020.
[11] Adrian Mouat, ” Docker Security- Using Docker containers safely”, O’Reilly Media, August 2015
[12] E. Socchi and J. Luu, “A deep dive into Dockerhubâˇ A´Zs security landscape - a story of inheritance?” https://www.duo.uio.no/bitstream/handle/10852/696/ A-Deep-Dive-into-Docker-Hubs-Security-Landscape.pdf? sequence=1&isAllowed=y, 2019, access date: 5. Mar 2019.
[13] D. Huang, H. Cui, S. Wen and C. Huang, "Security Analysis and Threats Detection Techniques on Docker Container," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 1214-1220, doi: 10.1109/ICCC47050.2019.9064441.
[14] C. Diekmann, J. Naab, A. Korsten and G. Carle, "Agile Network Access Control in the Container Age," in IEEE Transactions on Network and Service Management, vol. 16, no. 1, pp. 41-55, March 2019. doi: 10.1109/TNSM.2018.2889009
[15] S. Kwon and J. Lee, "DIVDS: Docker Image Vulnerability Diagnostic System," in *IEEE Access*, vol. 8, pp. 42666-42673, 2020. doi: 10.1109/ACCESS.2020.2976874
[16] StackRox with observations and analysis from CyberEdge Group “Container and Kubernetes Security: An Evaluation Guide”.
[17] Z. Lu, J. Xu, Y. Wu, T. Wang and T. Huang, "An Empirical Case Study on the Temporary File Smell in Docker files," in IEEE Access, vol. 7, pp. 63650-63659, 2019. doi: 10.1109/ACCESS.2019.2905424
[18] Z. Chen, Y. Zhang, and Z. Chen, “A categorization framework for common computer vulnerabilities and exposures,” https://academic. oup.com/comjnl/article/53/5/551/415583, 2009, access date: 28. Jan 2020.K. Wist and M. Helsem, “An Extensive Analysis of the Current

Vulnerability Landscape in Docker Hub Images," Master's thesis, Norwegian University of Science and Technology (NTNU), 2020.