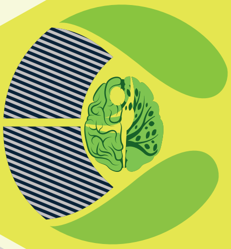
4



Container Security (Docker & Kubernetes)

– **Pratul Goyal**
Assistant Professor, Graphic Era Hill University
https://orcid.org/0000-0003-1040-8109 pratul.goyal111@gmail.com

– **Spardha Bisht**
Research Scholar Graphic Era Hill University
https://orcid.org/0000-0003-3818-7002 spardhabisht9@gmail.com

Docker is a containerization tool that packages all the software and their dependencies in a docker container to guarantee that other software does not impact the applications that are deployed. This can be any type of nature (dev, stage or qa). Docker will pull images as per application need and configure according to application needs. It can also be used to deploy the application to an environment that might have a conflict with the present settings.

Keywords

- Docker
- Kubernetes
- Containerization

ARTICLE HISTORY

Paper Nomenclature:
Theme Based Paper (TBP)
Paper Code: CYBNMV2N9SEP2020TBP1
Submission Online: 01-Sep-2020
Manuscript Acknowledged: 03-Sep-2020
Originality Check: 05-Sep-2020
Originality Test Ratio: 09% (Drillbit)
Peer Reviewers Comment: 12-Sep-2020
Blind Reviewers Remarks: 16-Sep-2020
Author Revert: 18-Sep-2020
Camera-Ready-Copy: 23-Sep-2020
Editorial Board Citation: 28-Sep-2020
Published Online First: 30-Sep-2020

Introduction

Suppose we already have an application running PHP 5.2 on a server and want to deploy a new application that requires PHP 7.4 on that same server. This will cause some version conflict on that server and might cause some features in the existing application to fail.

Here we are using the above Docker to containerize the application with the old application. Here the concept of Docker containers arises.

The above applications can be put into docker containers or if changes are getting occurred then these containers do not change the container configuration. Each container has a different operating system on the same physical machine. Docker helps to create, run, and deploy the application using containers.

Kubernetes:

Essentially, a container is a lightweight, virtualized, portable, software-defined environment in which software can run in isolation from other software running on the same physical host machine. Software that runs inside the container has only one purpose. In simple terms, a container is a packaged application with OS and libraries already installed. So, we can run it on any platform. It can behave like an immutable component for infrastructure. So, it becomes very cheap to destroy it and run it whenever and wherever we need.

Master Node -

- It is liable for the managing of the Kubernetes cluster.
- It is the major entry aspect of all administrative tasks.
- One that is managing the worker nodes, where the real services are running.

It is made of the following components:

API Server: It is managing the workloads and organization units. It is also the entry points of all rest commands which are controlling the cluster. Various libraries can easily communicate with it.

Etcd storage: It is a simple, lightweight, key-value distributed store that can be spread across several nodes. It uses configuration data which can be used in the cluster by each of the nodes.

Scheduler: It configures all pods and services on nodes. It is also liable for resource utilization on every host so that workloads are properly managed.

Controller-manager: The state of the cluster and repetitive activities are governed by this. This is called a controller for replication. It guarantees that replicas that are present for services are equal to the services that are deployed.

Nodes: They perform work in Kubernetes. A node can be a virtual machine or a physical machine, depending on the cluster. Every single node has the services necessary to run pods and is managed by the master components.

Kubernetes terminologies

Pods– Pods are a collection of one or more containers. It acts as a Kubernetes' core unit of management. Pods set the logical boundary for containers sharing the same context and resources.

Labels– Labels are random marks that can be placed as a member of a category on the above work units to mark them. For management purposes and activity targeting, these can then be picked.

Services– Services is a device that functions with other containers as a fundamental load balancer and ambassador. To offer an image of a single body, a service community integrates logical arrays of pods executing the same purpose.

Replication Controller– A more complex version of the pod is known as a replicated pod. These are handled as a type of work unit known as a replication controller. Replication controllers do that a specific number of pod replicas are running at any one time.

Continuous Container Security	
Build	**Ship**
Code Analysis	Image Signing
Hardening	User Access Controls
Image Scanning	
RUN	
Preparation	**Production**
Host & Kernel Security	Network Inspection & Visualization
SELinux & AppArmor	Layer 7-Based Application Isolation
Secure Docker Daemon	Threat Detection
Access Controls	Privilege Escalation Detection
Encryption	Container Quarantine
Auditing e.g. Docker Bench	Run-Time Vulnerability Scanning
Orchestration Security & Networking	Process Monitory, Packet Capture & Event Logging

Security Threats			
External Attacks	**Vulnerable Deployments**	**Compromised or Lost Data**	**Insiders & Third-Party Vendors**
Attackers that gains access to your cluster, deployed resources, apps, or personal information	Known Vulnerabilities are exploited to gain access to the cloud environment & run malicious software.	Incorrect storage of sensitive data and missing data encryption	Missing network isolation and segmentation that leads to misuse of legitimate permissions

Let us discuss about security loopholes in Kubernetes and Docker:

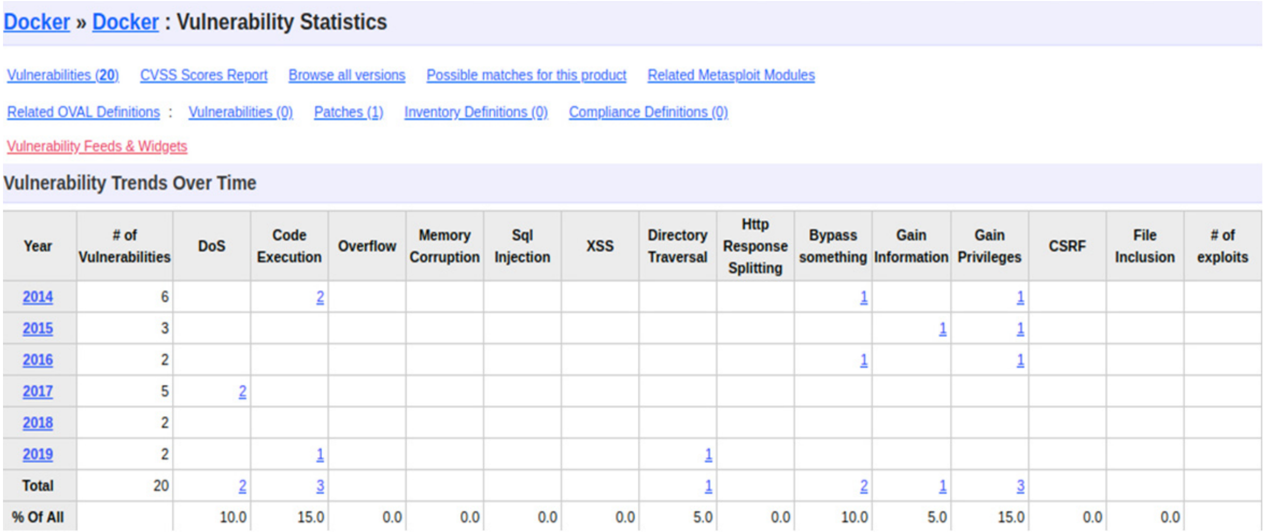
1. One of the greatest attacks happened few days back where world renowned automotive company has been hit by cryptojacking and data infiltration attack.

   In this attack, Kubernetes cluster was compromised because it was not password protected and they are directly accessible. This attack we call as "**Exploit Chain**".

For exploitation of this vulnerability:

- Always run deep port scan services and reverse IP-lookup also.
- Always look for deserialization vulnerabilities or deserializable variables in source code.
- Always look for Jenkins servers with port 8083, 8080.

2. Let us look up at the graph **CVE databases for Docker and Kubernetes**:



Docker » Docker : Vulnerability Statistics

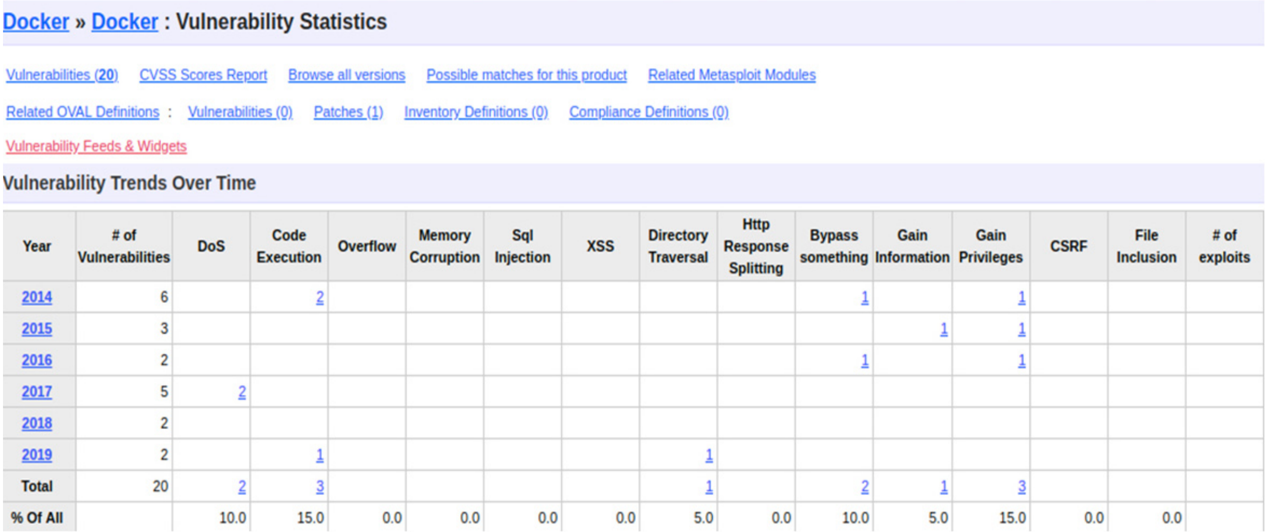
Vulnerabilities (20)   CVSS Scores Report   Browse all versions   Possible matches for this product   Related Metasploit Modules

Related OVAL Definitions :   Vulnerabilities (0)   Patches (1)   Inventory Definitions (0)   Compliance Definitions (0)

Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2014	6		2							1		1			
2015	3										1	1			
2016	2									1		1			
2017	5	2													
2018	2														
2019	2		1					1							
Total	20	2	3					1		2	1	3			
% Of All		10.0	15.0	0.0	0.0	0.0	0.0	5.0	0.0	10.0	5.0	15.0	0.0	0.0	



Docker » Docker : Vulnerability Statistics

Vulnerabilities (20)   CVSS Scores Report   Browse all versions   Possible matches for this product   Related Metasploit Modules

Related OVAL Definitions :   Vulnerabilities (0)   Patches (1)   Inventory Definitions (0)   Compliance Definitions (0)

Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2014	6		2							1		1			
2015	3										1	1			
2016	2									1		1			
2017	5	2													
2018	2														
2019	2		1					1							
Total	20	2	3					1		2	1	3			
% Of All		10.0	15.0	0.0	0.0	0.0	0.0	5.0	0.0	10.0	5.0	15.0	0.0	0.0	

Pratul Goyal, is an Assistant Professor with Graphic Era Hill University and IIM Postgraduate, He got extensive experience in a vivid domain like data science and cyber security. He is also a consultant and has worked as a corporate trainer for Simplilearn. He also has his blogging website datasciencejourney.com and YouTube Channel.

.

pratul.goyal111@gmail.com

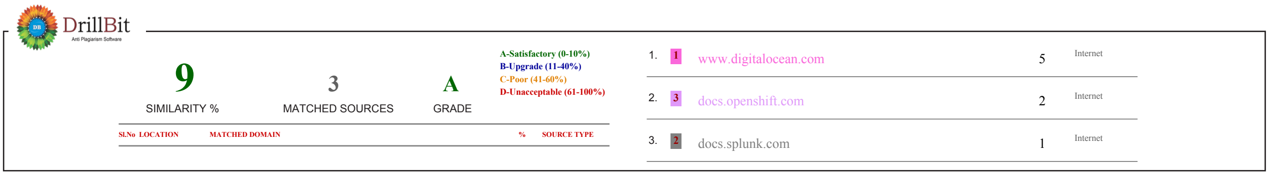
Spardha is working as an academic counsellor with Graphic Era University. She completed her post-graduation and pursuing PhD. She got a keen interest in public relation and organizational management.

.

spardhabisht9@gmail.com

Annexure I

Submission Date	Submission Id	Word Count	Character Count
05-Sep-2020	168048	883	5041



DrillBit
Anti Plagiarism Software

9	3	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

Sl.No	LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1.	1	www.digitalocean.com	5	Internet
2.	3	docs.openshift.com	2	Internet
3.	2	docs.splunk.com	1	Internet

Note: The Cybernomics had used the DrillBit plagiarism [https://www.drillbitplagiarism.com/] tool to check the originality.

Reviewers Comment

Reviewer's Comment 1: Article highlights one of the most needed and important aspects of organization or multi system working. It explains the technology which can be used to increase the work efficiency on the same server without any conflicts which is very beneficial for any organization.

Reviewer's Comment 2: The Article is very nicely written and sequencing of topics is very wonderful. This article is not only showing the benefits and good aspects of a technology but also its disadvantages and security loopholes which is going to be one of the most eye-catching parts of the article.

Reviewer's Comment 3: The article is comprehensive in nature. It talks about various aspects including benefits, disadvantages and security loopholes too. The authors have shown Docker and Kubernetes statistics with different numbers of vulnerabilities.

Editorial Excerpt

The article has 9% plagiarism which is an acceptable percentage for publication. The comments related to this manuscript are noticeably related to “Container Security” both subject-wise and research-wise. In this paper the authors have discussed Docker and Kubernetes, Docker generally puts application needs and configure accordingly. On the other hand, Kubernetes is a lightweight, virtualized, portable, software-defined environment in which software runs in isolation. The authors have shown Docker and Kubernetes statistics with different numbers of vulnerabilities.After comprehensive review and suggestions by the editorial board the paper has been categorized under the “**Theme Based Paper**” category.

Acknowledgement

The author is highly indebted to Scholastic Seed Inc.a publisher of Cybernmics Magazine and the editorial team including Resident Associate Editors (Ms.Sonakshi Jaiswal, Ms. Jyoti & Ms.Shailza), for making the write-up in the shape of an article.

Disclaimer

All the views expressed in this paper are my own, of which some of the content is taken from open source websites for knowledge purpose. The content drawn from different sources have been mentioned above in the references section.

Citation

Pratul Goyal and Spardha
“Container Security
(Docker & Kubernetes)”
Volume-2, Issue-9, September 2020.
(www.cybernomics.in)

Frequency: Monthly, Published: 2020
Conflict of Interest: Author of a Paper
had no conflict neither financially
nor academically.

Scholastic Seed Inc.
www.scholasticseed.in
https://orcid.org/0000-0002-7952-096X