



Project Report on
**Implementation of Devsecops using Jenkins, Maven, Docker, OWASP,
Trivy and Kubernetes for web application deployment with vulnerability
scanning.**

Submitted by

Vaibhav Chavan	(240344223038)
Sushant Ghatolkar	(240344223036)
Anjali Bais	(240344223002)
Umesh Panchal	(240344223023)

Under the guidance of
Mr. Sandeep Walvekar

**In partial fulfillment of the award of Post Graduate Diploma in
IT Infrastructure, Systems and Security
(PG-DITISS)**



**Sunbeam Institute of Information Technology,
Pune (Maharashtra)
PG-DITISS -2024**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Pune

Date:

Vaibhav Chavan
(240344223038)

Sushant Ghatolkar
(240344223036)

Anjali Bais
(240344223002)

Umesh Panchal
(240344223023)

CERTIFICATE

This is to certify that the project report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker, OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**”, submitted by **Vaibhav Chavan** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar

Guide

Mr. Vishal Salunkhe

Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**”, submitted by **Sushant Ghatolkar** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar

Guide

Mr. Vishal Salunkhe

Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**”, submitted by **Anjali Bais** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar

Guide

Mr. Vishal Salunkhe

Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**”, submitted by **Umesh Panchal** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar

Guide

Mr. Vishal Salunkhe

Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

APPROVAL CERTIFICATE

This Project II report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**” by **Vaibhav Chavan (240344223038)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: _____

(Signature)

(Name)

APPROVAL CERTIFICATE

This Project II report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker and Kubernetes for web application deployment with vulnerability scanning**” by **Sushant Ghatolkar (240344223036)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: _____

(Signature)

(Name)

APPROVAL CERTIFICATE

This Project II report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**” by **Anjali Bais (240344223002)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: _____

(Signature)

(Name)

APPROVAL CERTIFICATE

This Project II report entitled “**Implementation of Devsecops using Jenkins, Maven, Docker OWASP, Trivy and Docker Hub for web application deployment with vulnerability scanning**” by **Umesh Panchal (240344223023)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITIIS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: _____

(Signature)

(Name)

CONTENTS

TITLE	PAGE NO
Declaration	
Certificate	
Approval Certificate	
Abstract	
1.INTRODUCION	1
1.1 Applications	1
1.2 Organization and Project Plan	3
2. LITERATURE SURVEY	4
Paper 1	4
Paper 2	4
Paper 3	5
3. SYSTEM DEVELOPMENT AND DESIGN	6
3.1 Proposed System	6
3.2 Flow Chart	6
3.3 Technology used	7
3.3.1 AWS EC2	7
3.3.2 Git	7
3.3.3 Docker	7
3.3.4 Jenkins	7
3.3.5 Trivy	8
3.3.6 Sonarqube	8
3.3.7 Docker Hub account	8
3.3.8 OWASP	9
4. PROJECT OUTPUT	10
5. CONCLUSION	13

5.1 Conclusion	13
5.2 Future Scope	13
REFERENCES	14

ABSTRACT

In today's rapidly evolving software development landscape, security integration into the DevOps pipeline, known as DevSecOps, is critical for ensuring the delivery of secure and reliable applications. This paper explores the implementation of a DevSecOps pipeline for web application deployment using Jenkins, Maven, Docker, OWASP, Trivy and Kubernetes with a strong focus on automating vulnerability scanning.

The proposed framework utilizes Jenkins as the central automation server, orchestrating the entire CI/CD pipeline. Maven is employed for build management, ensuring that the application is consistently compiled and tested. Docker is leveraged to containerize the application, enabling consistent deployment across various environments, while Kubernetes handles the orchestration and scaling of these containers in a production environment.

A significant emphasis is placed on security within the pipeline. Integrated vulnerability scanning tools are used at different stages to identify and mitigate security risks early in the development cycle. This includes scanning code dependencies during the build phase with Maven, container images with Docker, and Kubernetes configurations before deployment. The implementation demonstrates how security can be seamlessly embedded into the CI/CD pipeline, providing continuous feedback and ensuring that only secure and compliant code is deployed.

This approach offers a robust, automated solution for modern web application deployment, ensuring that security is a continuous, integrated process rather than an afterthought. The results underscore the importance of embedding security in every stage of the development pipeline to deliver resilient applications in a dynamic and threat-prone environment.

1. INTRODUCTION

In the current digital age, the demand for rapid software development and deployment has driven the adoption of DevOps practices, which bridge the gap between development and operations. However, as the frequency and sophistication of cyber threats continue to increase, there is a growing need to integrate security into every phase of the software development lifecycle. This integration, known as DevSecOps, shifts the traditional security considerations left, embedding them into the development pipeline from the outset.

This paper presents a comprehensive approach to implementing DevSecOps for web application deployment, focusing on the use of Jenkins, Maven, Docker, and Kubernetes. Jenkins, an open-source automation server, is utilized to orchestrate the continuous integration and continuous deployment (CI/CD) pipeline. Maven, a powerful build automation tool, is employed to manage project dependencies and automate the build process. Docker, a leading containerization platform, ensures that the application and its dependencies are packaged in a consistent environment, facilitating smooth deployments. Kubernetes, the de facto standard for container orchestration, manages the deployment, scaling, and operation of containerized applications in production environments.

A key aspect of this implementation is the incorporation of automated vulnerability scanning tools throughout the pipeline. By integrating security checks at each stage—ranging from code analysis during the build phase with Maven to container image scanning with Docker and configuration checks in Kubernetes—this approach aims to detect and remediate security vulnerabilities early in the development cycle. This continuous monitoring and automated feedback loop ensure that only secure code is promoted through the pipeline, reducing the risk of security breaches in the deployed application.

The implementation outlined in this paper highlights the practical steps and tools required to achieve a secure, efficient, and scalable DevSecOps pipeline. By integrating security into the core of the CI/CD process, organizations can enhance their ability to deliver secure web applications at the speed demanded by today's fast-paced development environment.

1.1 Applications

Continuous Integration and Continuous Deployment (CI/CD) Pipeline Automation:

Jenkins serves as the backbone of the CI/CD pipeline, automating the entire process from code commit to deployment. Developers can push code changes to a version control system, triggering Jenkins to initiate a series of automated tasks. Maven is used within Jenkins to handle project builds, managing dependencies and ensuring that the application is consistently compiled and tested. This automation reduces manual intervention, speeds up the development process, and ensures that security checks are consistently applied at each stage.

Automated Security Testing and Vulnerability Scanning: Security is embedded within the pipeline through automated vulnerability scanning tools. During the build phase, Maven can be configured to run static code analysis tools that check for potential vulnerabilities in the codebase. Docker is then used to containerize the application, and each Docker image is scanned for vulnerabilities before being pushed to a registry. This ensures that no insecure images are deployed to production. Kubernetes, with its extensive configuration options, can be integrated with security tools that continuously monitor the running containers for potential threats, ensuring that the production environment remains secure.

Feedback and Continuous Improvement: The integration of security tools within the CI/CD pipeline provides continuous feedback to developers. If a vulnerability is detected at any stage, the pipeline can be configured to halt further progress, alerting developers to the issue. This immediate feedback loop allows for quick remediation of vulnerabilities, ensuring that only secure code is deployed to production. Over time, this process fosters a culture of security awareness among development teams, leading to continuous improvement in the overall security posture of the organization.

1.2 Project Plan

Table: Activities Details

Sr. No.	ACTIVITY	WEEK			
		1	2	3	4
1	Project group formation				
2	Project work to be started in respective labs				
3	First review with PPT presentation				
4	Design Use-Case view as per project				
5	Design Block diagram as per project				
6	Second review with PPT presentation				
7	Selection				
8	Final review with PPT presentation				
9	Implementation coding as per project				
10	Testing, Troubleshooting with different techniques				
11	Created Soft copy of project and then final hard copy				

2. LITERATURE SURVEY

Paper 1: - A Qualitative Study of DevOps Usage in Practice

Author: Floris Erich, C. Amrit & M. Daneva

Description: Organizations are introducing agile and lean software development techniques in operations to increase the pace of their software development process and to improve the quality of their software. They use the term DevOps, a portmanteau of development and operations, as an umbrella term to describe their efforts. In this paper we describe the ways in which organizations implement DevOps and the outcomes they experience. We first summarize the results of a Systematic Literature Review that we performed to discover what researchers have written about DevOps. We then describe the results of an exploratory interview-based study involving six organizations of various sizes that are active in various industries. As part of our findings, we observed that all organizations were positive about their experiences and only minor problems were encountered while adopting DevOps.

Paper 2: - Devops, A New Approach To Cloud Development & Testing

Author: Dhaya Sindhu Battina

Description: The main purpose of this paper is to explore DevOps and its applications in Cloud development and testing. There's no denying it: DevOps and cloud go hand in hand. This trend will only continue since the bulk of cloud development projects now use DevOps. The advantages of utilizing DevOps with cloud applications are increasingly becoming evident. Competing well in the market necessitates a company's ability to supply services and applications at a rapid rate. To be effective, management procedures and tools need a model that is both swift and dependable. Because of this, we must automate the DevOps processes utilizing cloud and noncloud DevOps automation technologies while designing cloud-native apps. The purpose of this article is to discuss how to migrate DevOps to the cloud and improve software development and operational agility. Likewise, this project will examine ways to expand such DevOps processes and automation to public and/or private clouds. If one is interested in learning more about how the emerging field of DevOps is changing the IT industry, read this paper. Understanding how DevOps and the Cloud work together to aid organizations in transforming themselves is the ultimate objective.

2. SYSTEM DEVELOPMENT AND DESIGN

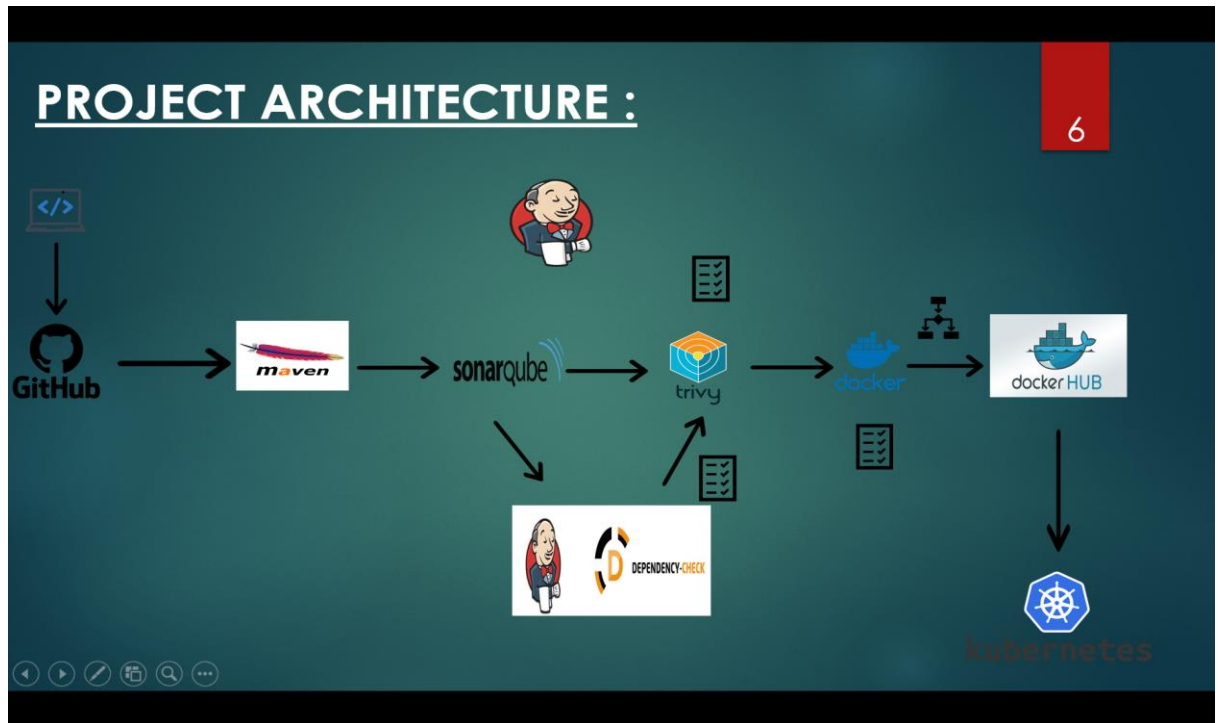
3.1 Proposed System

We propose a system where we are setting up two Amazon EC2 instances to host your web application. Storing our application code in a version-controlled repository (Git-Hub).

Set up a CI/CD pipeline using Jenkins. On code changes, trigger an automated build and deployment process using Jenkins. Jenkins will initiate the pipeline by fetching the code from the remote repository. At Maven Maven will run the fetched code. If, for some reason, the build process fails, then it will stop the pipeline and inform the user. In case the build succeeds, the pipeline goes to the next step if it exists, otherwise, it completes the process. At Maven Maven will run the fetched code. In case the build succeeds, the pipeline goes to the next step if it exists, otherwise, it completes the process. SonarQube Scanner will analyse the code and send the results to the SonarQube server.

If the code fails to meet the Quality Gate standards, the pipeline will fail, and Jenkins will notify the user. If the Quality Gate is passed, the pipeline will proceed. OWASP Dependency-Check Ensures that all dependencies used in the project are free from known vulnerabilities before the build process. During building, Docker pulls the application image. If Docker build fails, Jenkins breaks the pipe and informs the user. The scans with Trivy mean the vulnerabilities of a Docker image and OS vulnerabilities, and in application dependencies. The Docker image is pushed to Docker Hub. If the push operation fails, Jenkins stops the pipeline and notifies the user.

3.2 Flow chart



3.3 Technology used

3.3.1 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service provided by Amazon it allows you to rent virtual servers in the cloud, known as instances, to run your applications and workloads. EC2 provides a scalable and flexible infrastructure that enables you to quickly deploy and manage virtual servers without the need to invest in physical hardware.

3.3.2 Git

Git is a distributed version control system (VCS) designed to manage source code history and facilitate collaborative software development

3.3.3 Docker

Docker is an open-source platform that allows you to automate the deployment, scaling, and management of applications using containerization. Containers are lightweight, portable, and isolated environments that package an application and its dependencies together.

3.3.4 Jenkins

Jenkins is an open-source automation server that facilitates the continuous integration and continuous delivery (CI/CD) of software projects. It helps automate various tasks related to building, testing, and deploying applications, making the development and release process more efficient and reliable.

3.3.5 Docker Hub

Docker Hub is a cloud-based registry service provided by Docker that allows users to store and share Docker container images. It serves as a central repository for Docker images, making it easy for developers to distribute and deploy applications in containerized environments

3.3.6 Maven

Maven is a powerful project management and comprehension tool that provides developers with a complete build lifecycle framework. Using an XML file to describe the project structure, dependencies, build order, and required plugins, Maven automates the build process and manages project dependencies from a central repository. This simplifies the work of developers by handling project building, reporting, and documentation from a central piece of information.

3.3.7 Sonarqube

SonarQube is an open-source platform designed for continuous inspection of code quality. It performs automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities in over 20 programming languages. SonarQube integrates with CI/CD workflows to provide developers with immediate feedback on code quality issues and improvement suggestions, facilitating better codebase maintainability and reducing the risk of vulnerabilities.

3.3.8 Trivy

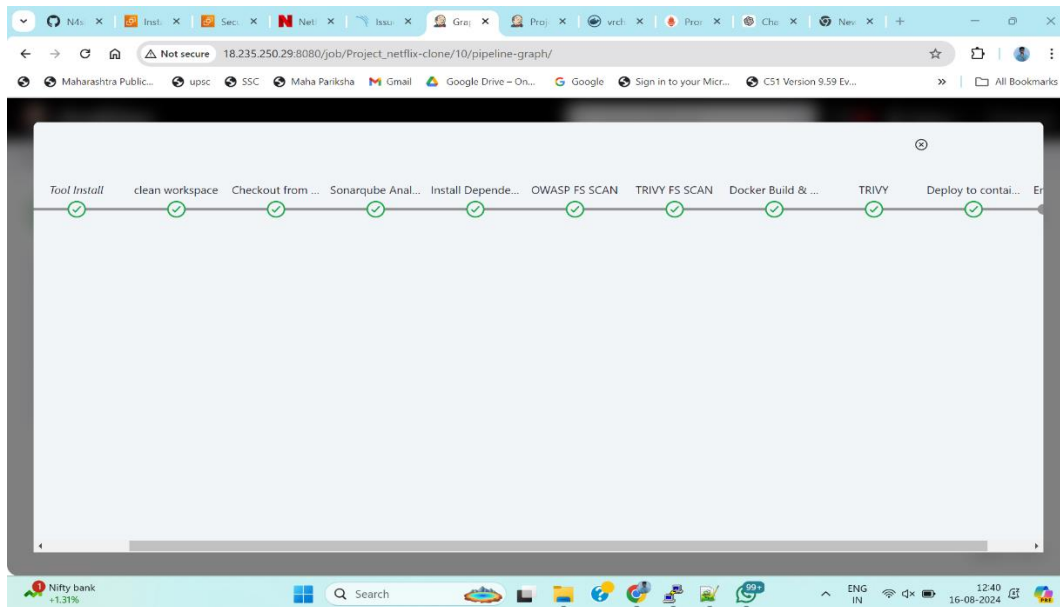
Trivy is a comprehensive, open-source vulnerability scanner that specializes in detecting security vulnerabilities within container images, file systems, and Git repositories. Trivy scans for vulnerabilities in OS packages (Alpine, RHEL, CentOS, etc.) and application dependencies (Bundler, Composer, npm, yarn, etc.), providing detailed reports to help developers identify and fix security issues efficiently.

3.3.9 OWASP

OWASP Dependency-Check is a tool developed by the Open Web Application Security Project (OWASP) that is used to identify vulnerabilities in third-party libraries and dependencies that your application uses. It is particularly useful in modern software development where applications often rely on open-source components, libraries, and frameworks.

4. Project Output

4.5 Jenkins



4.5.1 Trivy

```
ubantu@ip-172-31-19-150: ~/DevSecOps-Projects
shom      Scan SBOM for vulnerabilities and licenses
vm         [EXPERIMENTAL] Scan a virtual machine image

Management Commands
module     Manage modules
pipeline   Manage pipeline
vex        [EXPERIMENTAL] VEX utilities

Utility Commands
clean      Remove cached files
completion Generate the auto-completion script for the specified shell
convert    Convert Trivy JSON report into a different format
help       Help about any command
server     Server mode
version    Print the version

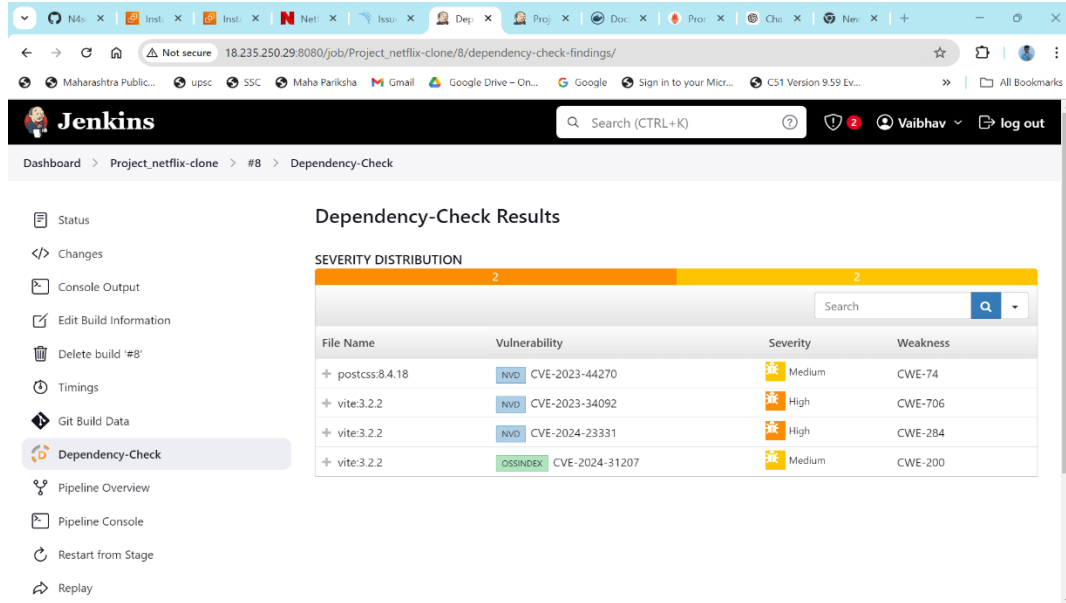
Flags:
--cache-dir string      cache directory (default "/root/.cache/trivy")
--config string         config path (default "trivy.yaml")
-d, --debug             debug mode
-f, --format string     version format (json)
--generate-default-config write the default config to trivy-default.yaml
-h, --help              help for trivy
--insecure              allow insecure server connections
-q, --quiet             suppress progress bar and log output
--timeout duration      timeout (default 5m0s)
-v, --version           show version

Use "trivy [command] --help" for more information about a command.

2024-08-16T08:14:38Z   FATAL   Fatal error   unknown command "vrchavan029/netflix" for "trivy"
ubantu@ip-172-31-19-150:~/DevSecOps-Projects$ sudo trivy image vrchavan029/netflix
2024-08-16T08:15:09Z   INFO    [db] Need to update DB... repository="ghcr.io/aquasecurity/trivy-db:2"
2024-08-16T08:15:09Z   INFO    [vuln] Vulnerability scanning is enabled
2024-08-16T08:15:12Z   INFO    [secret] Secret scanning is enabled
2024-08-16T08:15:12Z   INFO    [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-08-16T08:15:12Z   INFO    [secret] Please see also https://aquasecurity.github.io/trivy/v0.54/docs/acanner/secret#recommendation for faster secret data
2024-08-16T08:15:12Z   INFO    Detected OS   family="alpine" version="3.20.2"
2024-08-16T08:15:12Z   INFO    [alpine] Detecting vulnerabilities... os_version="3.20" repository="3.20" pkg_num=66
2024-08-16T08:15:12Z   INFO    Number of language specific files num=0

vrchavan029/netflix (alpine 3.20.2)
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
ubantu@ip-172-31-19-150:~/DevSecOps-Projects$
```

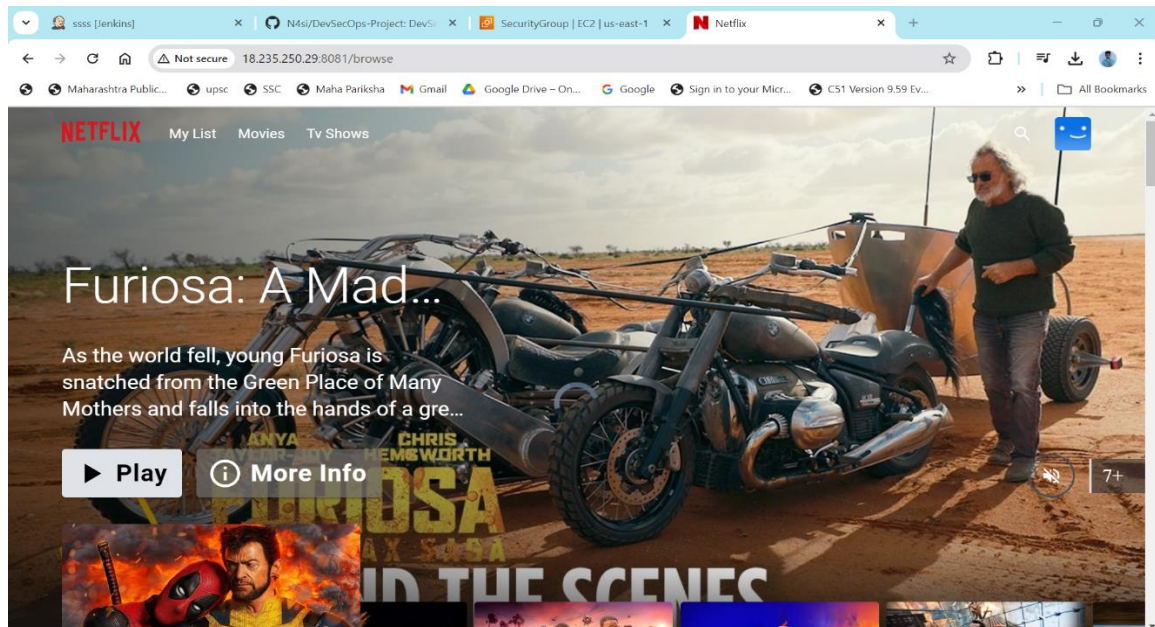
4.5.2 OWASP Dependency Check



The screenshot shows the Jenkins web interface for a job named 'Project_netflix-clone'. The 'Dependency-Check' section is active, displaying 'Dependency-Check Results'. A 'SEVERITY DISTRIBUTION' bar chart shows 2 High severity findings (orange) and 2 Medium severity findings (yellow). Below the chart is a table of findings:

File Name	Vulnerability	Severity	Weakness
+ postcss:8.4.18	NVD CVE-2023-44270	Medium	CWE-74
+ vite:3.2.2	NVD CVE-2023-34092	High	CWE-706
+ vite:3.2.2	NVD CVE-2024-23331	High	CWE-284
+ vite:3.2.2	OSINDEX CVE-2024-31207	Medium	CWE-200

4.5.3 WebPage



The screenshot shows the Netflix web interface for the movie 'Furiosa: A Mad Max Saga'. The main image features Chris Hemsworth standing next to a motorcycle in a desert setting. The title 'Furiosa: A Mad...' is displayed prominently. Below the title, a synopsis reads: 'As the world fell, young Furiosa is snatched from the Green Place of Many Mothers and falls into the hands of a gre...'. The 'Play' button is visible. At the bottom, there are thumbnails for other content, including a Deadpool movie.

4.5.4 Docker Hub

The screenshot shows a web browser window with multiple tabs. The active tab is 'vrch' and the address bar shows 'hub.docker.com/repository/docker/vrchavan029/netflix/general'. The page content is as follows:

vrchavan029/netflix

Updated 5 minutes ago

This repository does not have a description **INCOMPLETE**

This repository does not have a category **INCOMPLETE**

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	5 minutes ago	5 minutes ago

[See all](#)

Docker commands Public View

To push a new tag to this repository:

```
docker push vrchavan029/netflix:tagname
```

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

Repository overview **INCOMPLETE**

An overview describes what your image does and how to run it. It displays in [the public view of your repository](#) once you have pushed some content.

5. CONCLUSION

5.1 Conclusion

- Project Implementation: DevSecOps Framework for Web Application. Successfully automated and secured deployment of a Java-based web application.
- Utilized tools like GitHub, Jenkins, Maven, Docker, SonarQube, OWASP, Trivy and monitoring.
- Established robust, secure, and efficient lifecycle management system. Enhanced operational performance and exemplified DevSecOps principles in software development.

5.2 Future Scope



Kubernetes is an open-source container orchestration platform designed to automate deploying, scaling, and operating application containers. It groups containers that make up an application into logical units for easy management and discovery.



Grafana is an open source tool which is used for monitoring and visualization of data from different sources in real time. It is mainly used for developing real-time and active application dashboards that would help to monitor the health and performance of the applications, infrastructure as well as the services being offered.



Amazon Simple Storage Service (AWS S3) is a scalable, durable, and highly available object storage service offered by Amazon Web Services (AWS). S3 is designed to store and retrieve any amount of data from anywhere on the web, making it a fundamental building block for cloud-based applications.

REFERENCES

Paper 1: - A Qualitative Study of DevOps Usage in Practice

Author: Floris Erich, C. Amrit & M. Daneva

Paper 2: - Devops, A New Approach To Cloud Development & Testing

Author: Dhaya Sindhu Battina

Paper 3: - Review paper on Snort and reviewing its applications in different fields

Author: Harpreet Sandhu, Manpreet Kaur.