

DevSecOps Pipeline - Integration of security tools in CI/CD

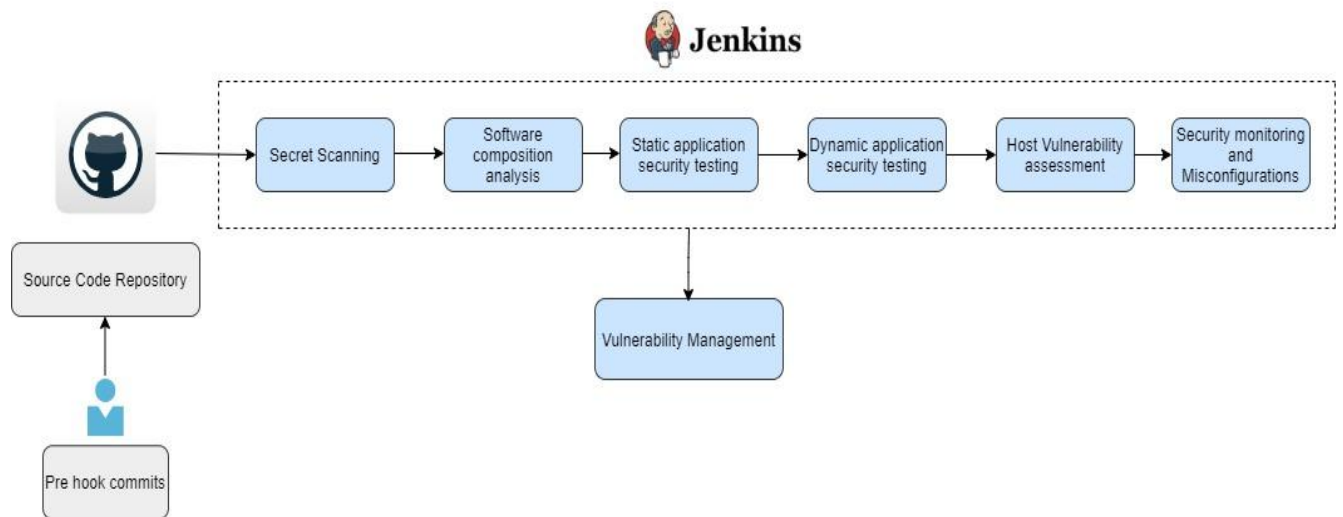
For implementing devsecops pipeline we are using a Java based vulnerable WebGoat application. For setting up the infrastructure on cloud (AWS), we will require 4 ec2 instances (t2.medium)

- Jenkins Server
- SAST, DAST tools
- Application Server
- Vulnerability Management server

and for cloud security monitoring :

- AWS Config service
- AWS Security Hub service
- AWS GurdDuty
- IAM access analyzer
- AWS Macie

Pipeline Architecture



Lab Environment

- 1) Install Pre-commit or Pre-push hooks on developer workstation.

Talisman is a tool that installs a hook to a repository to ensure that potential secrets or sensitive information do not leave the developer's workstation.

Install talisman as a pre-commit hook

```
Selinux webGoat_java:$ sudo ./install-talisman.sh pre-commit
Downloading and verifying binary...

Talisman successfully installed to '.git/hooks/pre-commit'.
Selinux webGoat_java:$
```

It validates the outgoing changeset for things that look suspicious - such as potential SSH keys, authorization tokens, private keys

```
Selinux webGoat_java:$ git branch
* master
Selinux webGoat_java:$ sudo git add config.json
Selinux webGoat_java:$ sudo git commit -m "Added configuration file"

Talisman Report:
+-----+-----+-----+
| FILE | ERRORS | SEVERITY |
+-----+-----+-----+
| config.json | Expected file to not to contain  
| base64 encoded texts such as:  
| keys:["AKIAVV4VIKBHJ4QEZCBZ","BV30EZEJRsgJlf1u... | medium |
+-----+-----+-----+
| config.json | Potential secret pattern : Key :  
| "b5baa024-a121-49c8-9c80-3b17679e0555" | medium |
+-----+-----+-----+
| config.json | Potential secret  
| pattern : "AWS  
| keys:["AKIAVV4VIKBHJ4QEZCBZ" | medium |
+-----+-----+-----+
| config.json | Potential secret pattern :  
| Sql-password: zaxa9qwmd9 | medium |
+-----+-----+-----+

If you are absolutely sure that you want to ignore the above files from talisman detectors, consider pasting the following format in .t
alismanrc file in the project root

fileignoreconfig:
- filename: config.json
  checksum: 227e58083bee077745fe301134301c91e8e3537222f0db5a7c9b54273af7c094

Selinux webGoat_java:$
```

2) For setting up the Jenkins server Launch an ec2-instance (t2.medium)

Pre-requisites

- JDK 8 or 11, Maven
- Install Jenkins on that instance
- Once Jenkins is installed, install suggested plugins

After installing suggested plugins install following plugins

1. Blue Ocean
2. Maven integration
3. SSH agent
4. SonarScanner
5. OWASP Dependency-Check

Configure maven, sonarqube and owasp dependency check tools in global tool configurations of Jenkins (Jenkins -> Manage Jenkins -> Global tool configuration)

- For SonarQube

SonarQube Scanner

SonarQube Scanner installations

[Add SonarQube Scanner](#)

☰ SonarQube Scanner

Name

sonar

☒ Install automatically

☰ Install from Maven Central

Version

SonarQube Scanner 4.6.2.2472 ▾

- For Maven

Maven

Maven installations

[Add Maven](#)

☰ Maven

Name

Maven

MAVEN_HOME

/usr/share/maven

☐ Install automatically

- For OWASP Dependency Check

Dependency-Check

Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name

OWASP-DC

☒ Install automatically

Install from dl.bintray.com

Version

dependency-check 6.1.6

Add Installer

Delete Installer

Launch one more ec2-instance(t2.medium) and install sonarqube on that server. After installing sonarqube server configure sonarqube server URL and API key in jenkins (Jenkins -> Manage jenkins ->Configure system)

SonarQube servers

☐ Environment variables Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

sonar

Server URL

http://15.206.170.150:9000/

Default is http://localhost:9000

Server authentication token

Sonarqube API token

Add

SonarQube authentication token. Mandatory when anonymous access is disabled.

Install trufflehog tool on jenkins for scanning secret

- `pip install truffleHog3`

Create a pipeline and configure a project URL, Jenkins file path and branch name. After creating a pipeline for every commit pipeline will trigger and will identify security vulnerabilities in each stage. The scan results will be uploaded to the vulnerability management tool.

✓ DemoProject 37 >

PipelineChangesTestsArtifacts🔄⚙️📄Logout✕

Branch: —

🕒 6m 7s

Changes by suyash550kulkarni

Commit: —

🕒 6 days ago

Started by an SCM change

Start

Initialize

Check secrets

Software composition ana...

Static analysis

Generate build

Deploy to server

Dynamic analysis

Host vulnerability assessment

Incidents report

End

< >

Incidents report - <1s

🔄 Restart Incidents report

📄

⬇️

✓ > Maven — Use a tool from a predefined Tool Installation

<1s

✓ > Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.

<1s

✓ > echo "In-Progress" — Shell Script

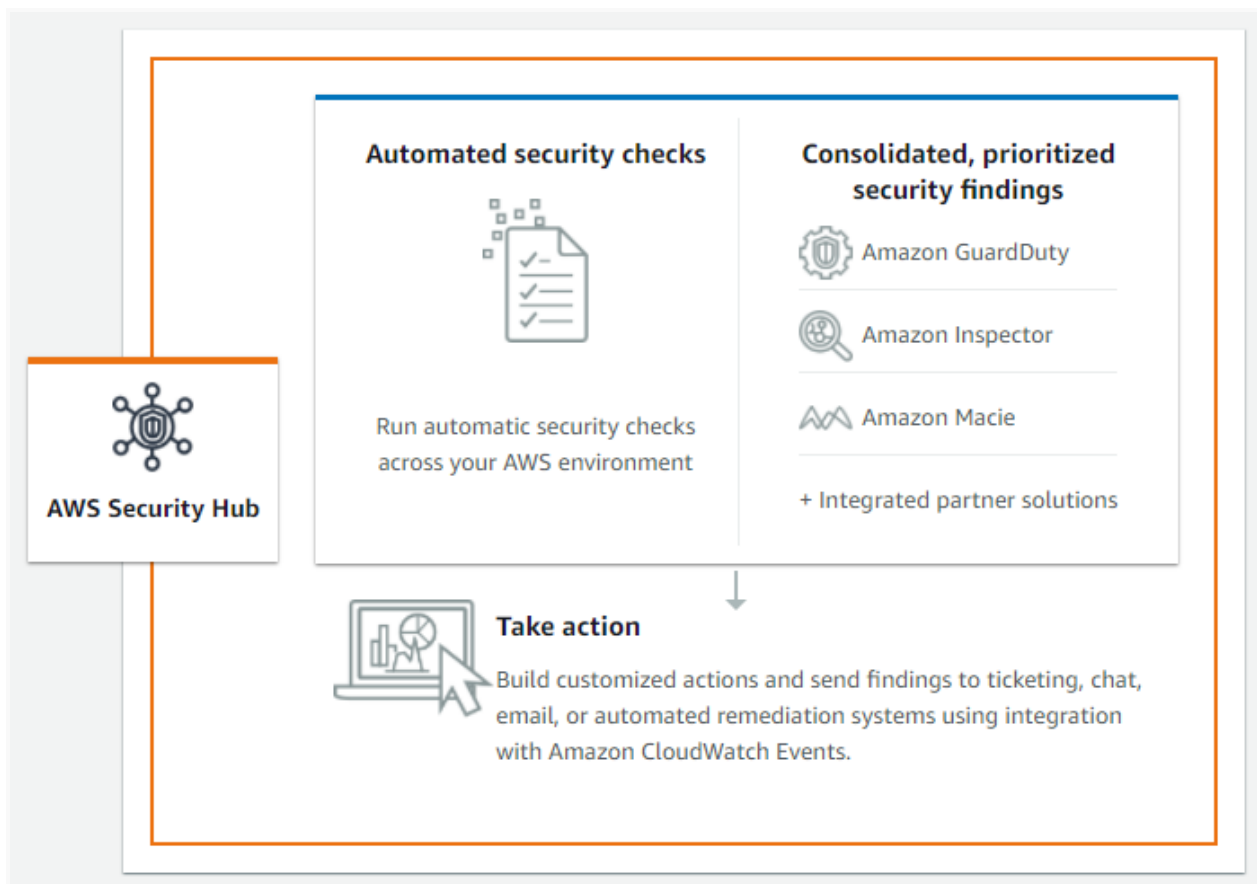
<1s

3) Security Monitoring and Misconfigurations

Prerequisites

- Enable AWS config and global services
- Enable GuardDuty, Macie
- Create IAM access analyzer

AWS Security Hub gives you a comprehensive view of your security alerts and security posture across your AWS accounts. We can consolidate different accounts in AWS Security Hub



Findings for infrastructure security misconfigurations are based on the following standards:

- CIS AWS Foundations Benchmark v1.2.0
- AWS Foundational Security Best Practices v1.0.0
- PCI DSS v3.2.1

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

suyash.kulkarni @ 3906-1817-3518

Mumbai

Support

Security Hub

Summary

Security standards

Insights

Findings

Integrations

Settings

What's new 10

Security Hub

Findings

Severity label is CRITICAL

Workflow status is NEW

Workflow status is NOTIFIED

Record state is ACTIVE

Add filters

< 1 >

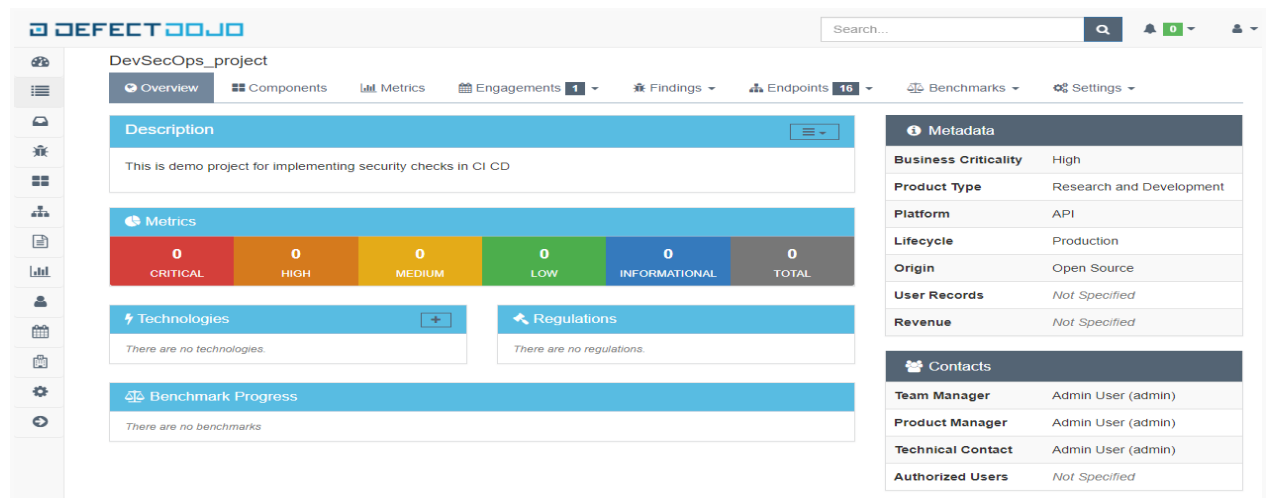
	Severity	Workflow status	Record State	Company	Product	Title	Resource	Status
<input type="checkbox"/>	CRITICAL	NEW	ACTIVE	AWS	Security Hub	S3.2 S3 buckets should prohibit public read access	S3 Bucket roshan.logs	Failed
<input type="checkbox"/>	CRITICAL	NEW	ACTIVE	AWS	Security Hub	S3.2 S3 buckets should prohibit public read access	S3 Bucket roshandel	Failed

4) Vulnerability management server

Launch an ec2 instance and install defectdojo as a vulnerability management tool on the server.

```
git clone
https://github.com/DefectDojo/django-DefectDojo
cd django-DefectDojo
# building
docker-compose build
# running
docker-compose up
```

Login to the application and create a product, engagements



To import the result of different security findings we are using defect dojo API's (<https://defectdojo.readthedocs.io/en/latest/api-v2-docs.html>)

For example, Import ZAP scan report to defectdojo

- ```
curl -X POST
"http://<IP_address>:8080/api/v2/import-scan/" -H
"accept: application/json" -H "Content-Type:
multipart/form-data" -H "X-CSRFToken:
vUsaTeXI2m1I94DBRizQBR2dpS68XO4HD70CQx1q5bPxLiGGpylhcI
Wbiw8uVSfR" -F "scan_date=2021-06-17" -F
"minimum_severity=Info" -F "active=true" -F
"verified=true" -F "scan_type=ZAP Scan" -F
```



```
"file=@zap_report" -F "engagement=1" -F
"close_old_findings=false" -F "push_to_jira=false"
```

After Importing the results , you can check the findings under engagements

The screenshot displays the DefectDojo interface for a CI/CD test engagement. The main content area shows a list of scans with their respective findings. The table has columns for Title / Type, Date, Lead, Total Findings, Active (Verified), Mitigated, Duplicates, and Notes. The scans listed are: AWS Security Hub Scan, Dependency Check Scan, Trufflehog3 Scan, and ZAP Scan. The right sidebar provides details for the CI/CD test, including Status (In Progress), Dates (7th June - 2nd July), Length (25 days), Service Account (Admin User), Tracker (Not Specified), Repo (Not Specified), Test Strategy (Not Specified), Updated (9 minutes ago), and Created (2 weeks, 1 day ago). Below this, the CI/CD Engagement Details section shows Build ID (Not Specified), Commit Hash (Not Specified), and Branch/Tag (Not Specified).

| Title / Type           | Date                          | Lead | Total Findings | Active (Verified) | Mitigated | Duplicates | Notes |
|------------------------|-------------------------------|------|----------------|-------------------|-----------|------------|-------|
| !AWS Security Hub Scan | June 22, 2021 - June 22, 2021 |      | 81             | 81 (81)           | 0         | 0          |       |
| !Dependency Check Scan | June 18, 2021 - June 18, 2021 |      | 287            | 287 (287)         | 0         | 0          |       |
| !Trufflehog3 Scan      | June 18, 2021 - June 18, 2021 |      | 49             | 49 (49)           | 0         | 0          |       |
| !ZAP Scan              | June 18, 2021 - June 18, 2021 |      | 7              | 7 (7)             | 0         | 0          |       |