CI/CD Pipeline

Owner: Amar
Reviewer:
Contributors:
Date Generated: Mon Jul 08 2024

OWASP Threat Dragon

Executive Summary

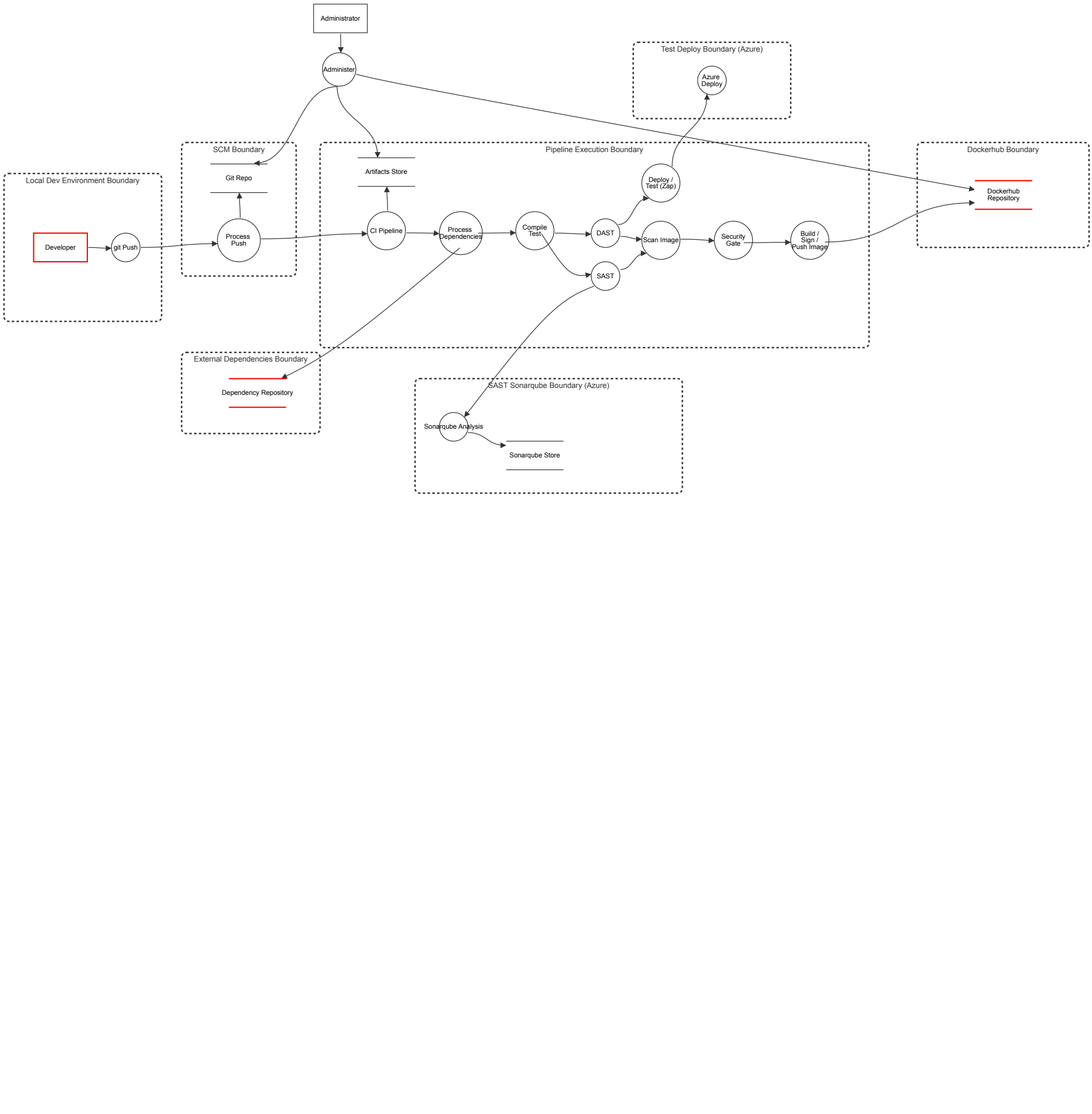
High level system description

DevSecOps Lab threat model

Summary

Total Threats	6
Total Mitigated	1
Not Mitigated	5
Open / High Priority	0
Open / Medium Priority	5
Open / Low Priority	0
Open / Unknown Priority	0

CI Pipeline


Administrator
Administer
Test Deploy Boundary (Azure)
Azure Deploy
SCM Boundary
Git Repo
Artifacts Store
Pipeline Execution Boundary
Deploy / Test (Zap)
Dockerhub Boundary
Dockerhub Repository
Local Dev Environment Boundary
Developer
git Push
Process Push
CI Pipeline
Process Dependencies
Compile Test
DAST
Scan Image
Security Gate
Build / Sign / Push Image
SAST
External Dependencies Boundary
Dependency Repository
SAST Sonarqube Boundary (Azure)
Sonarqube Analysis
Sonarqube Store

CI Pipeline

Developer (Actor)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Developer commits secret	Spoofing	Medium	Mitigated		A developer commits a secret to the source code repository, intentionally or not. Anyone with read access can obtain the secret and use it to exploit user data	Use a Vault to handle secrets. Follow the the principle of least privilege Follow the principle of need to know, so secrets for production are not known by developers
2	Developer Laptop Stolen/Compromised	Spoofing	Medium	Open		Developer laptop stolen, hard drive is not encrypted. Attacker gains commit access to repo and ssh access to systems the dev has access to	Encrypt all hard drives Follow Least Privilege principle Verify all devices are attached to the correct domain Run endpoint protection on all devices (Crowdstrike)
3	Developer adds malicious package	Spoofing	Medium	Open		Developer adds malicious package to the project, creating a backdoor into the codebase	Ensure PR is needed for all merge requests, requiring multiple approvals Create policy that ensures packages must only come from approved sources Build tooling into CI pipeline which identifies non sanctioned packages and repositories Enable service operation threat detection such as GuardDuty
7	Rogue Employee is used to find unpatched internal systems.	Spoofing	Medium	Open		A rogue employee is used to detect unpatched internal systems.	Add vulnerability scans to detect unpatched vulnerabilities Perform background checks on all employees Segregate network and follow Least Privilege principle Harden version control system

Dependency Repository (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
5	Compromised Open Source Dependency	Tampering	Medium	Open		An open source dependency gets compromised.	Enable automated dependency analysis to check for compromised dependencies Fix and stabilise dependency versions in a policy enabled BOM Enable service operation threat detection such as GuardDuty Create policy that ensures packages must only come from approved sources Build tooling into CI pipeline which identifies non sanctioned packages and repositories

Dockerhub Repository (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
6	Developer Access to Container Registry	Tampering	Medium	Open		Developer access to container registry. Compromised images can be inserted into the deployment process.	Enable full logging of push/pull operations for the registry Seperate container registries for development and production targets Restrict production container registry access to the CI/CD pipeline Implement image signing and verification