CI/CD Pipeline

Owner: Mahesh Bisl
Reviewer:
Contributors:
Date Generated: Tue Jun 25 2024

OWASP Threat Dragon

Executive Summary

High level system description

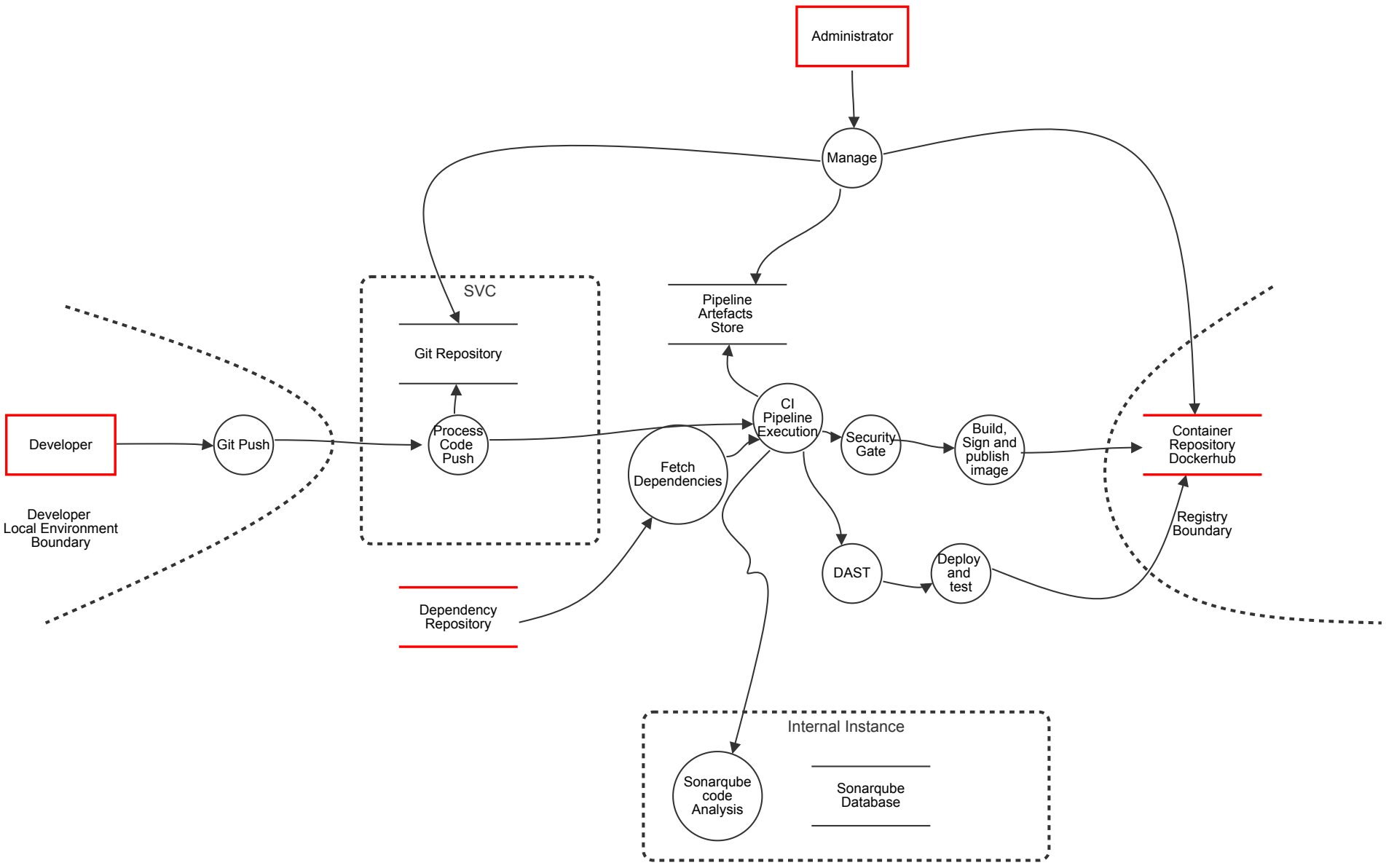
CI/CD pipeline for DevSecOps labs

Summary

Total Threats	7
Total Mitigated	0
Not Mitigated	7
Open / High Priority	0
Open / Medium Priority	7
Open / Low Priority	0
Open / Unknown Priority	0

CI Pipeline

CI Pipeline for DevSecOps lab



CI Pipeline

Developer (Actor)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Developer Commits secret to Git	Spoofing	Medium	Open		A developer commits secrets to version control either accidentally or on purpose. An angry employee will read access to the repository finds the secret and proceeds to use them to steal end-user data	Vault Adhere to principle of "Need to know" and ensure that production secrets are not shared with individuals who don't have a need to know Adhere to the principle of "Least privilege" and ensure secrets only have the permission to the pipeline.
2	Developer laptop stolen	Spoofing	Medium	Open		A developers laptop is lost or stolen. The hard drive is not encrypted so the attacker is able to use a tool such as Konboot to bypass password authentication and have commit access to all the repositories and ssh access to all the systems the developer has access to.	Mitigations Ensure all company laptops are encrypted Ensure developers repository access follows the principle of least privilege and only has access to what they need to Ensure devices are using enterprise management software like Jamf, etc. Ensure all devices are attached to the domain Ensure all devices are running end point protection like Crowdstrike
3	A developer commits AWS secrets	Spoofing	Medium	Open		A developer commits AWS secrets to version control either accidentally or on purpose. An angry employee with read access to the repository finds the secrets and proceeds to use them to steal end-user data.	Use something like git-secrets in repositories to help detect, flag, and block merge requests which may contain secrets Ensure the company is using secure secret management techniques such as tools like Hashicorp Vault Ensure secrets are stored encrypted at rest Adhere to the principal of “Need to Know” and ensure that produciton secrets are not shared with individuals who don’t have a need to know Adhere to the principal of “Least Privilege” and ensure secrets only have the the permissions to fulfill their use case Ensure logging is enabled so that secret usage can be tracked to the system and/or user who acquired and used them Enable and use threat detection such as GuardDuty to detect anomolies in patterns of service operation Ensure version control system is only available on the corporate VPN
4	developer creates a benign looking and named, but yet malicious package	Spoofing	Medium	Open		A developer creates a benign looking and named, but yet malicious package and hosts it on a public GitHub repo. While working on a feature, the developer adds their malicious package into the project, backdooring the codebase.	Ensure all merge requests require review and approval from other engineers. Create administrative policy that mandates packages must only come from company approved sources Build and integrate tooling into CI pipelines as a technical control that reviews package files and identifies non-company sanctioned repositories; enforcing company policy Enable and use threat detection such as GuardDuty to detect anomolies in patterns of service operation

Git Push (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

(Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Data Flow (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Process
Code
Push (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Git Repository (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

CI
Pipeline
Execution (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Pipeline
Artefacts
Store (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Security Gate (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Build, Sign and publish image (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Dependency Repository (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
6	An popular opensource dependency used by the application gets compromised by an attacker.	Tampering	Medium	Open		An popular opensource dependency used by the application gets compromised by an attacker. The attacker implants bitcoin mining software into the dependency so that consuming applications will mine bitcoin for the attacker	Ensure automated dependency analysis is being performed on application dependencies to detect dependencies known to have been compromised Ensure lock files are being used so that builds repeatedly use the same version of a dependency until it is explicitly upgraded Enable and use threat detection such as GuardDuty to detect anomolies in patterns of service operation Create administrative policy that mandates packages must only come from company approved sources Build and integrate tooling into CI pipelines as a technical control that reviews package files and identifies non-company sanctioned repositories; enforcing company policy Subscribe the engineering team to security disclosure distro lists so that they may be alerted when supply chain attacks are discovered

Fetch Dependencies (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Container Repository Dockerhub (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
7	Container registry permissions are not well segmented	Information disclosure	Medium	Open		Developers are given access to the container registry so that they may publish POC / testing images or pull production images for debugging purposes. Container registry permissions are not well segmented and allow a developer to push images to the same registry that production images are pulled from. A developer proceeds to publish a backdoored image directly to the repository and tags as latest, knowing that it will get picked up as part of the next deploy.	Ensure logging is configured for the container registry so that all image pushes and pulls are captured Do not use the same container registry for developer testing / POC that is used for production image storage Restrict publish access to the production container registry to only the CI/CD system Enable and use threat detection such as GuardDuty to detect anomolies in patterns of service operation Implement image signing that will only allow the deployment of images that have been properly signed Review Google's Binary Autnorization for Borg and consider adopting some of their controls

DAST (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Deploy and test (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Manage (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Administrator (Actor)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
5	A nation-state attacker proceeds to get one of their operatives employed with the company	Spoofing	Medium	Open		A nation-state attacker proceeds to get one of their operatives employed with the company. After performing reconnaissance on the corporate network, they determine that the version control system has been operating with unpatched vulnerabilities. The attacker proceeds to compromise the version control server and implant malware which backdoors the company code releases	Ensure corporate patch management policy exists that outlines SLA's for patching of systems Automate vulnerability scanning of systems and software to identify unpatched vulnerabilities Ensure the company performs background checks on all new employees Ensure the principal of least privilege is followed and employees who don't need access to systems are not provided access to systems Ensure the network is segregated so employees only have network access to the systems that are applicable to their job function Enable and use threat detection such as GuardDuty to detect anomolies in patterns of service operation Ensure version control system is only available on the corporate VPN Ensure security groups / firewalls are configured to only allow necessary network access Ensure company hardening standards exist which outline procedures for hardening systems Ensure the version control system is hardened in alignment with corporate standards including things such as: Removing all un-necessary services Disabling root login Disabling password login Using a "golden image" Using minimal containers if running a containerized environment Etc. Implement a SIEM and forward events, syslog, etc.

Sonarqube code Analysis (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Sonarqube Database (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations