Name: Virali Shailesh gada

Intern id:303

**DevOps Threat Matrix (Microsoft)**

# Tactic: Initial Access

## Technique: SCM authentication

* Exploit leaked GitHub/GitLab tokens found in public repos.
* Reuse stolen access tokens to access SCM repositories.

## Technique: CI/CD service authentication

* Authenticate using previously compromised CI/CD API keys.
* Leverage exposed GitHub Actions secrets from old builds.

## Technique: Configured webhooks

* Abuse misconfigured webhooks to trigger attacker-controlled actions.
* Inject malicious payloads via exposed webhook endpoints.

# Tactic: Execution

## Technique: Poisoned pipeline execution

* Modify pipeline scripts to run reverse shells.
* Include rogue steps in YAML to execute malware.

## Technique: Dependencies tampering

* Submit pull request with malicious open-source dependency.
* Publish malicious package to public repo used by target.

## Technique: DevOps resources compromise

* Use stolen API keys to execute actions in cloud CI services.
* Invoke unauthorized build jobs via DevOps consoles.

# Tactic: Persistence

## Technique: Change code/pipeline configuration in repository

* Edit .gitlab-ci.yml to reintroduce attacker shell.
* Configure recurring backdoor step in Jenkinsfile.

## Technique: Inject in artifacts

* Embed malicious code in build artifacts like jars or containers.
* Add backdoor scripts in zipped build bundles.

## Technique: Modify images in registry

* Push altered Docker image with trojan to internal registry.
* Replace latest tag with image containing exploit.

# Tactic: Privilege Escalation

## Technique: Secrets stored in private repositories

* Scan internal repos for hardcoded secrets.
* Extract cloud credentials from .env or config files.

## Technique: Commit from pipeline to protected branches

* Exploit weak permissions to merge malicious code.
* Auto-push backdoored commits to release branch.

## Technique: Certificates and identities from metadata services

* Query cloud metadata endpoints inside CI containers.
* Harvest IAM tokens and temporary credentials.

# Tactic: Credential Access

## Technique: User credentials

* Dump shell history to collect login info.
* Harvest credentials from version control history.

## Technique: Service credentials

* Extract API keys from logs or environment variables.
* Access credentials stored in config files.

## Technique: Create service credentials

* Use access to generate new API keys.
* Add new CI tokens using admin privileges.

# Tactic: Lateral Movement

## Technique: Compromise build artifacts

* Alter artifacts that get deployed to production.
* Inject commands in scripts run during deployment.

## Technique: Registry injection

* Push malicious image to trusted internal container registry.
* Replace existing image versions with trojaned ones.

## Technique: Spread from pipeline into deployment resources

* Trigger deployment job to target staging or prod systems.
* Deploy poisoned containers to cloud workloads.

# Tactic: Defense Evasion

## Technique: Service logs manipulation

* Modify or delete CI logs to erase traces.
* Suppress output of suspicious commands.

## Technique: Compilation manipulation

* Alter build scripts to bypass checks.
* Hide backdoors in compiled binaries.

## Technique: Reconfigure branch protections

* Disable protected branch settings to allow malicious commits.
* Reduce required code reviewers to none.

# Tactic: Impact

## Technique: DDoS using pipeline compute resources

* Flood external target using scripts in pipeline jobs.
* Run botnet scripts during CI builds.

## Technique: Crypto mining over pipeline compute resources

* Mine cryptocurrency during CI builds using free runners.
* Inject Monero miner into CI script.

## Technique: Local DoS to CI/CD pipelines

* Introduce infinite loop in CI config.
* Break runner agent with malformed job.

# Tactic: Exfiltration

## Technique: Clone for private repositories

* Mirror internal Git repos to attacker host.
* Clone private repositories via SSH using stolen keys.

## Technique: Access to pipelines logs

* Download logs that include environment variables or secrets.
* Scrape logs for sensitive data using script.

## Technique: Exfiltrate data from production resources

* Send database credentials to external server.
* Leak application configs over HTTPS.