CONF42 2022

MINIMUM VIABLE SECURITY

FOR PYTHON APPS

David Melamed
Co-Founder & CTO, Jit

JAN 27, 2022

Jit

WHO IS THIS GUY?

- Co-Founder & CTO at **Jit**
- Passionate about technology and security
- PhD in BioInformatics (France)
- Full-stack Engineer in the CTO Office at CloudLock (acquired by Cisco)
- Cloud Security CTO Office at Cisco
- Has been involved in various communities (PyCon IL, AWS User Group...)

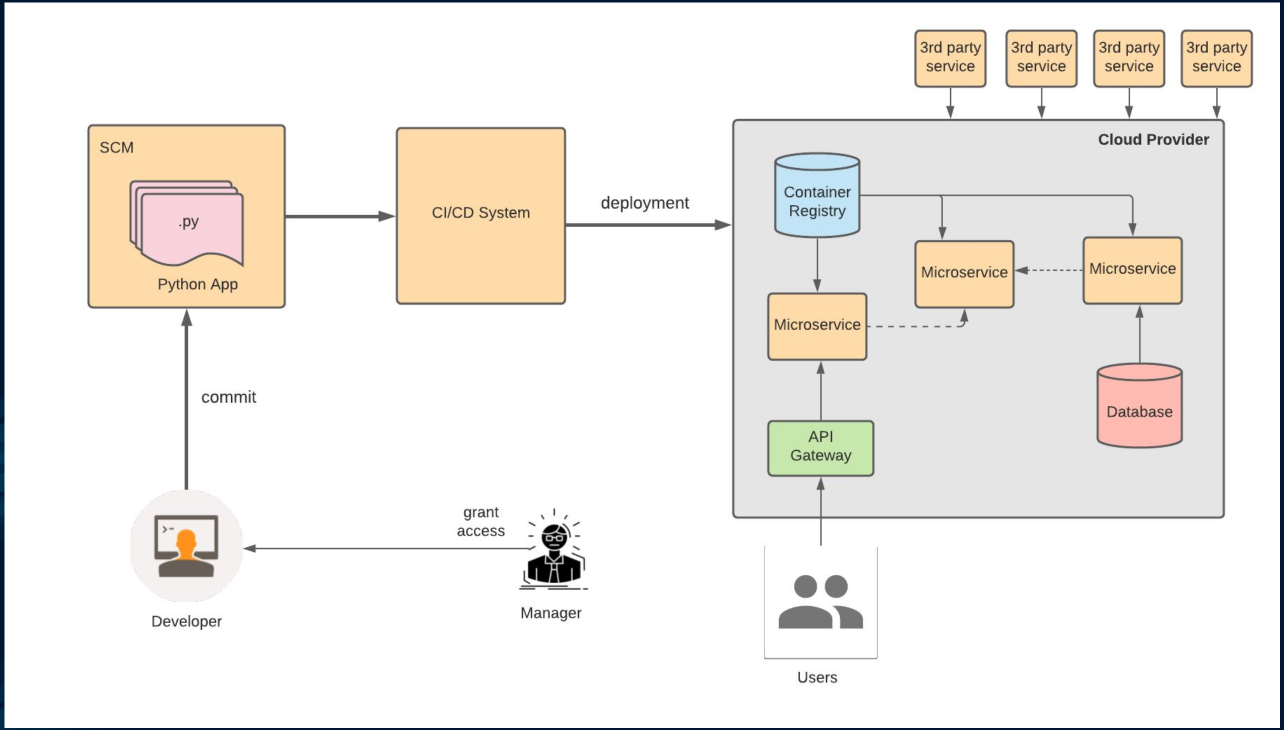
Jit

SECURITY: START ON DAY 0

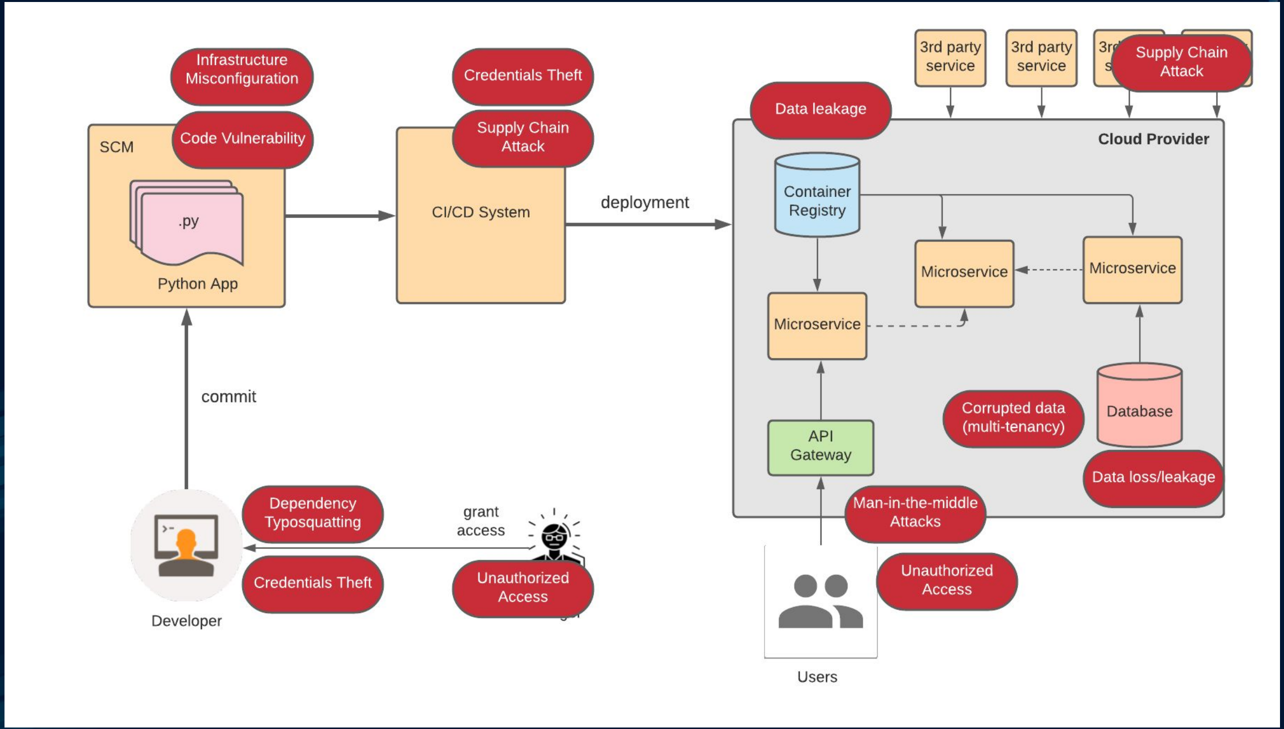
- Never too early to start
- Manage security debt from Day 0
- Makes security a Continuous concern
- **Start minimal and iterate**

> **“Writing a secure web application starts at the architecture phase. A vulnerability discovered in this phase can cost as much as 60 times less than a vulnerability found in production code.”**
>
> ●— Andrew Hoffman (Salesforce)

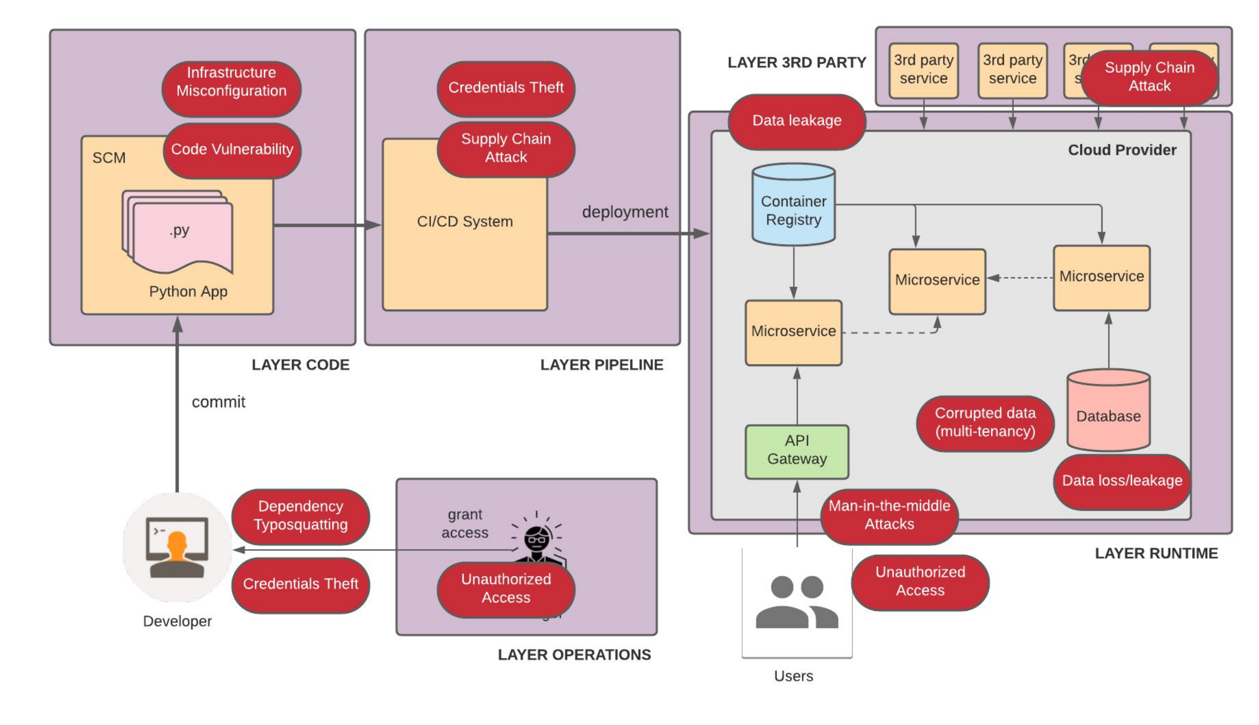
TYPICAL CLOUD APP - ARCHITECTURE



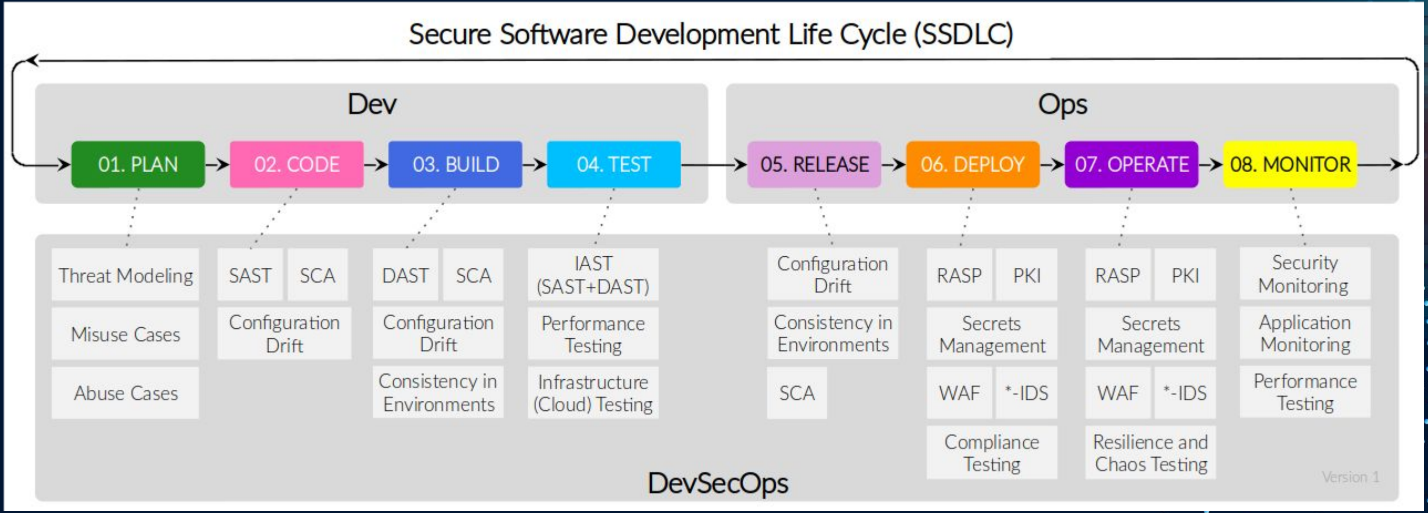
TYPICAL CLOUD APP - RISKS



TYPICAL CLOUD APP - LAYERS



SECURE SLDC



Secure Software Development Life Cycle (SSDLC)

Dev

- 01. PLAN: Threat Modeling; Misuse Cases; Abuse Cases
- 02. CODE: SAST; SCA; Configuration Drift
- 03. BUILD: DAST; SCA; Configuration Drift; Consistency in Environments
- 04. TEST: IAST (SAST+DAST); Performance Testing; Infrastructure (Cloud) Testing

Ops

- 05. RELEASE: Configuration Drift; Consistency in Environments; SCA
- 06. DEPLOY: RASP; PKI; Secrets Management; WAF; *-IDS; Compliance Testing
- 07. OPERATE: RASP; PKI; Secrets Management; WAF; *-IDS; Resilience and Chaos Testing
- 08. MONITOR: Security Monitoring; Application Monitoring; Performance Testing

DevSecOps

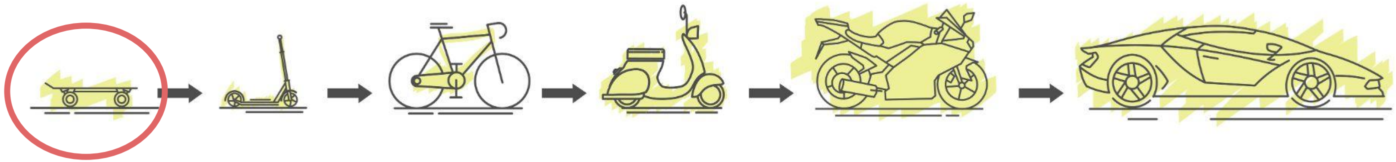
Version 1

Source: https://holisticsecurity.io/2020/02/10/security-along-the-sdlc-for-cloud-native-apps/

FROM MVP TO MVS

MV~~P~~S

Minimal **V**iable **S**ecurity



MINIMAL VIABLE SECURITY

3rd-Party apps security

- **MFA on all 3rd party services**

Runtime Application Security

- **API security**
- Yearly pentesting

Security Operations

- Employee offboarding process
- Incident response plan
- Generate a privacy/security policy

Code Security

- **Static code scanning**
- **Dependency check**
- **Hard-coded secrets**

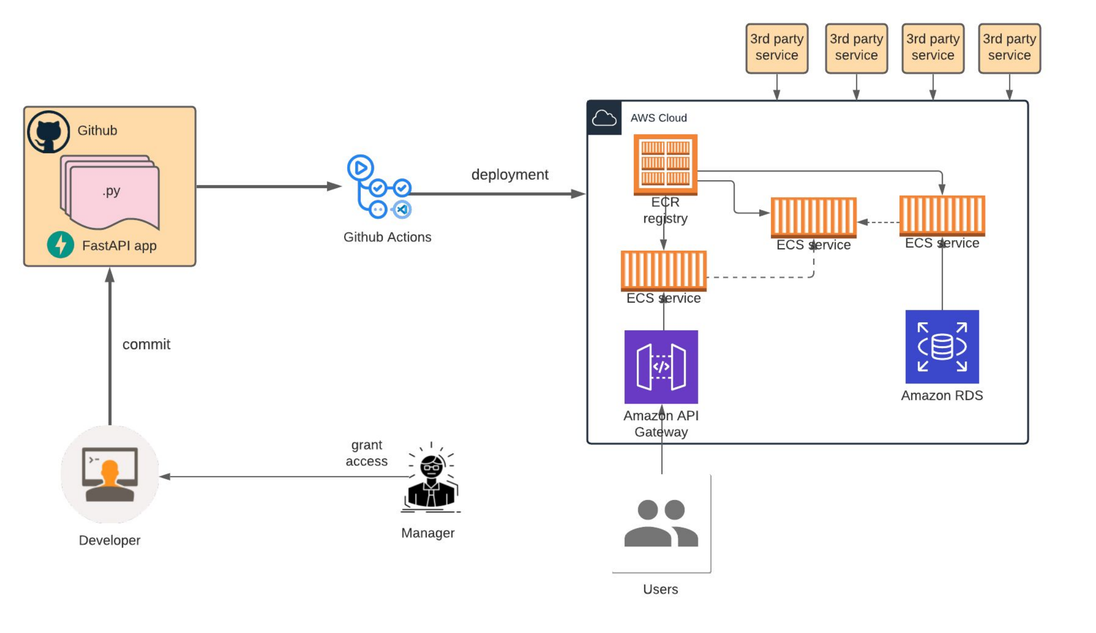
CI/CD Security

- Source control and CI/CD tools security
- Account hardening
- Container image scanning

Infrastructure security

- Cloud misconfiguration detection
- Secure remote access
- Cloud account hardening

DEMO: SECURING A PYTHON APP (1)



Sample Python cloud application

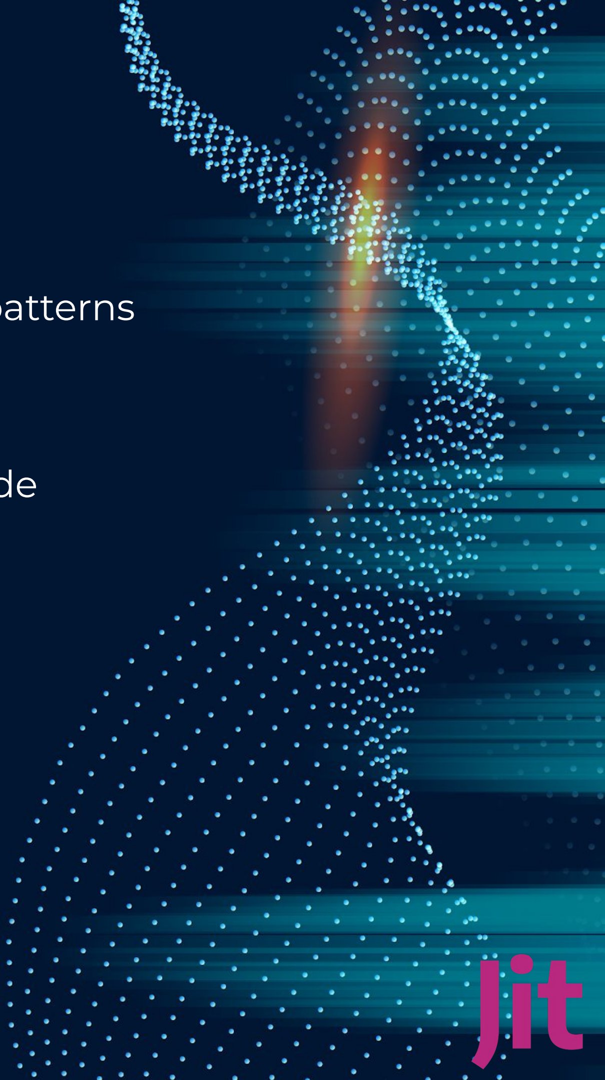
DEMO: SECURING A PYTHON APP (2)

SAST	SAST (Secrets)	IAC	SCA	DAST	MFA
Bandit	Gitleaks	KICS	Safety	OWASP ZAP	Custom

CODE VULNERABILITIES

- Code source static analysis and detection of existing patterns
- For this demo, we will use: Bandit
  - Security open-source linter for Python source code
  - Includes 35 rules for detecting vulnerabilities

SOURCE CODE



Jit

SOURCE CODE

SECRET DETECTION

- Part of SAST analysis, looks for hard-coded secrets based on regexes and high entropy

- For this demo, we will use: Gitleaks
  - Supports multiple types of secrets: API keys, AWS credentials, SSH keys…
  - Supports detecting secrets in git history

INFRASTRUCTURE

INFRASTRUCTURE AS CODE

- When the infrastructure is expressed as code, it is possible to detect misconfigurations early by scanning the code

- A popular tool : KICS
  - Supports many infrastructure types: CloudFormation, Terraform, Ansible, Kubernetes, Docker, Ansible, ARM…
  - Includes 2,000+ built-in queries

SOURCE CODE

DEPENDENCY VULNERABILITY

- Publicly disclosed vulnerabilities in project dependencies (CPE / CVE)
- For this demo, we will use: Safety
  - Detects publicly disclosed vulnerabilities contained within a project's dependencies
  - Open Source (monthly update) or commercial

Malware in PyPI Code Shows Supply Chain Risks

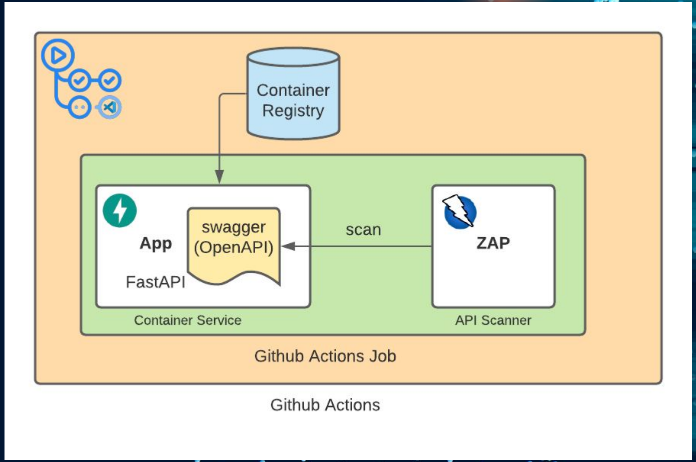
A code backdoor in a package on the Python Package Index demonstrates the importance of verifying code brought in from code repositories.

Twelve malicious Python libraries found and removed from PyPI

One package contained a clipboard hijacker that replaced victims' Bitcoin addresses in an attempt to hijack funds from users.

RUNTIME MONITORING

- Some vulnerabilities can only be detected at runtime, e.g. cross site scripting (XSS) or SQL injection (SQLi)
- For this demo, we will use: ZED Attack Proxy
  - Free web app scanner by OWASP
  - Includes 17 built-in rules
  - Uses OpenAPI to crawl endpoints



RUNTIME

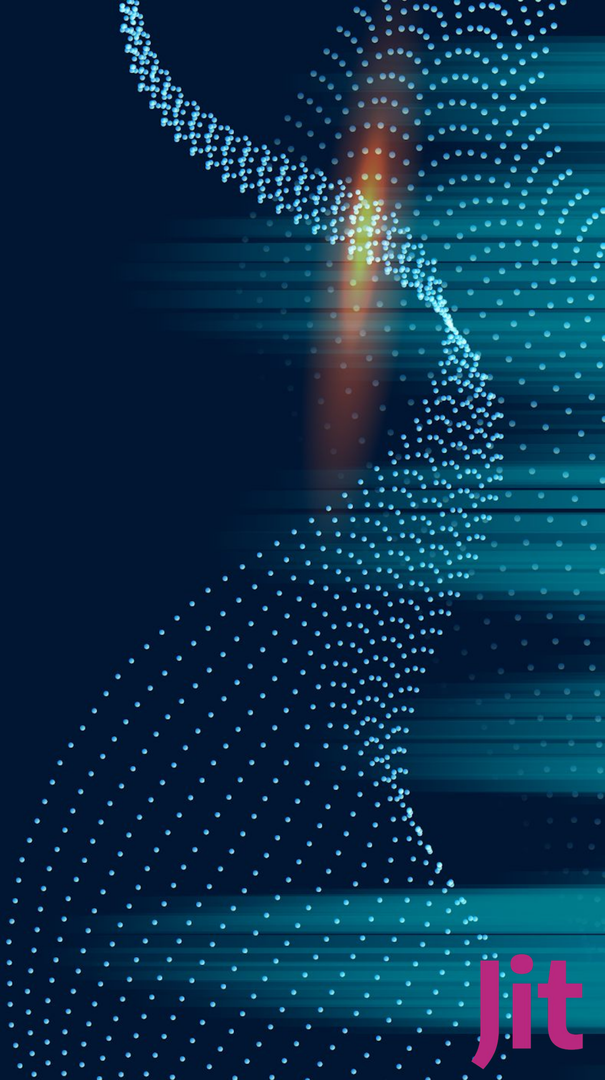
PIPELINE

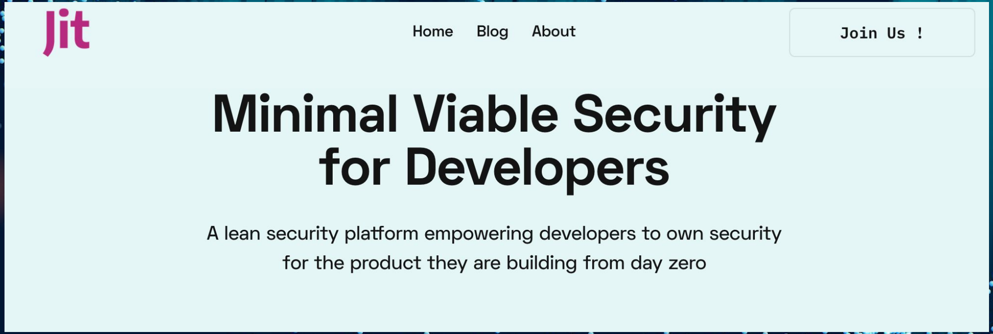
SCM SECURITY

- With rising supply chain attacks, it is critical to ensure that the SCM service and the pipeline are properly secured
- The minimum is to ensure that MFA is enabled everywhere
- For this demo, we will write a custom control
    - List Github users that don't have MFA enabled
    - Fail the control if the list is not empty
    - Will leverage a token with **admin:read** score stored as Github secret

Source: https://mentorphile.com/2018/09/14/demo-or-die/

https://github.com/dvdmelamed/conf42-2022-talk





Minimal Viable Security for Developers

A lean security platform empowering developers to own security for the product they are building from day zero

Want to help developers with security?

Continuous Security Platform for Developers

WE ARE HIRING!

Drop me an email david@jit.io

Website: www.jit.io

Beta Program: jit-me-in@jit.io