

Melhores Práticas em Segurança de Aplicações (AppSec)

Integrando Segurança ao Ciclo de Vida de
Desenvolvimento de Software (SDLC)

Marcus Lelis - AFKConf



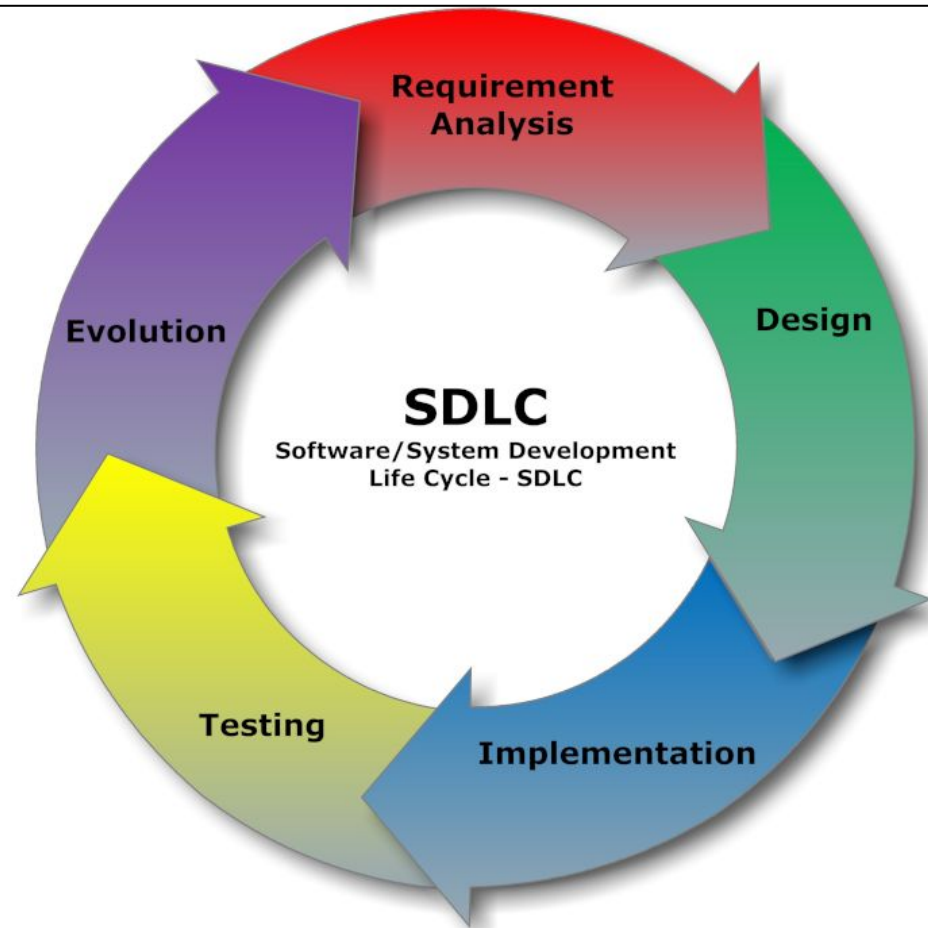
Introdução ao AppSec:

A segurança de Aplicações (AppSec) é um pilar fundamental para desenvolver software seguro. Através da integração de práticas de segurança no Ciclo de Vida de Desenvolvimento de Software (SDLC), podemos identificar e mitigar vulnerabilidades desde as fases iniciais, protegendo nossas aplicações contra ataques maliciosos.



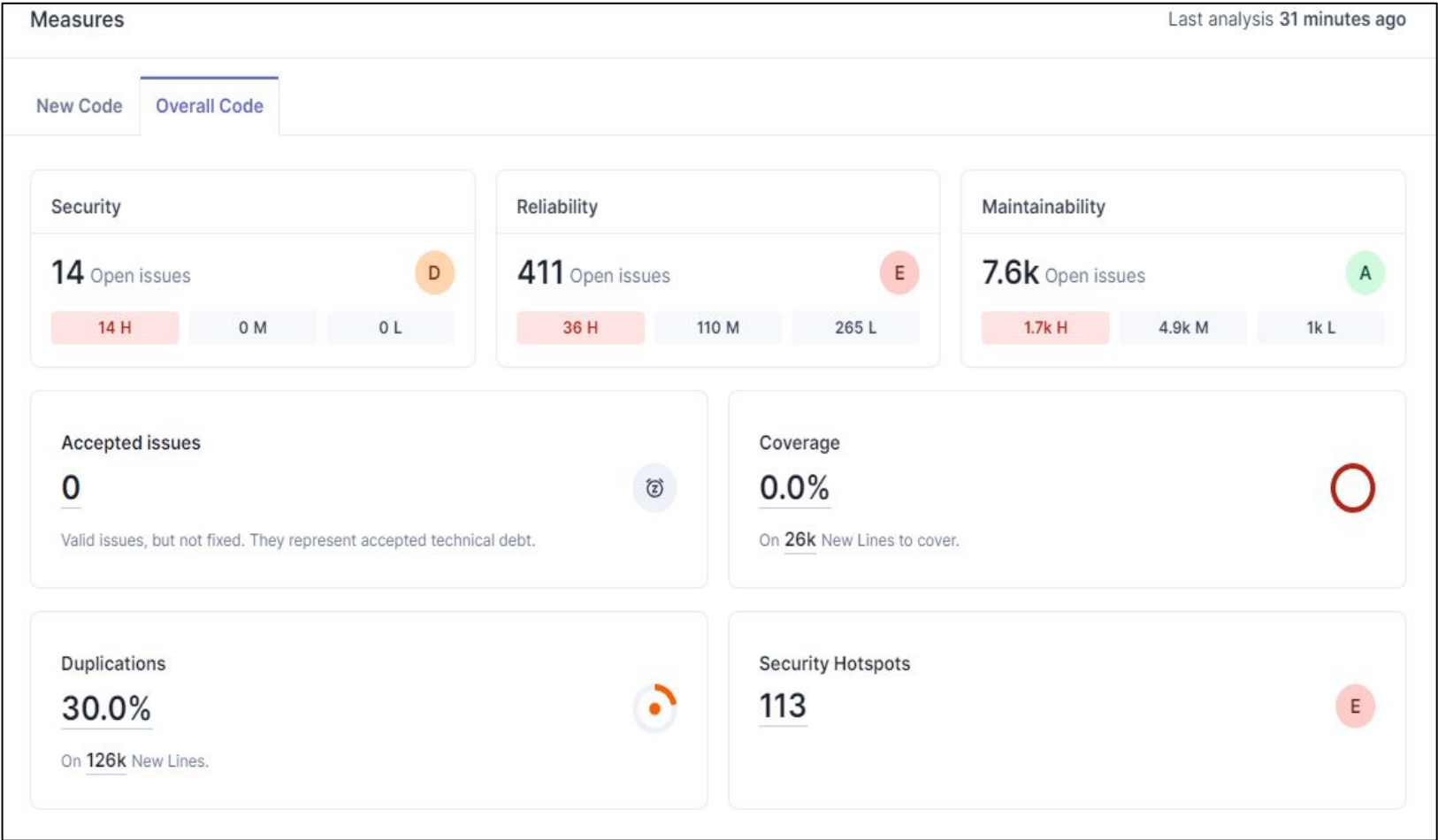
Ciclo de Vida de Desenvolvimento de Software (SDLC) e AppSec:

Integrar AppSec ao SDLC significa adotar medidas de segurança em cada etapa, desde o planejamento e design até a manutenção. Isso inclui análises de requisitos de segurança, revisões de código, testes de segurança e monitoramento contínuo para garantir a aplicação segura ao longo do tempo.



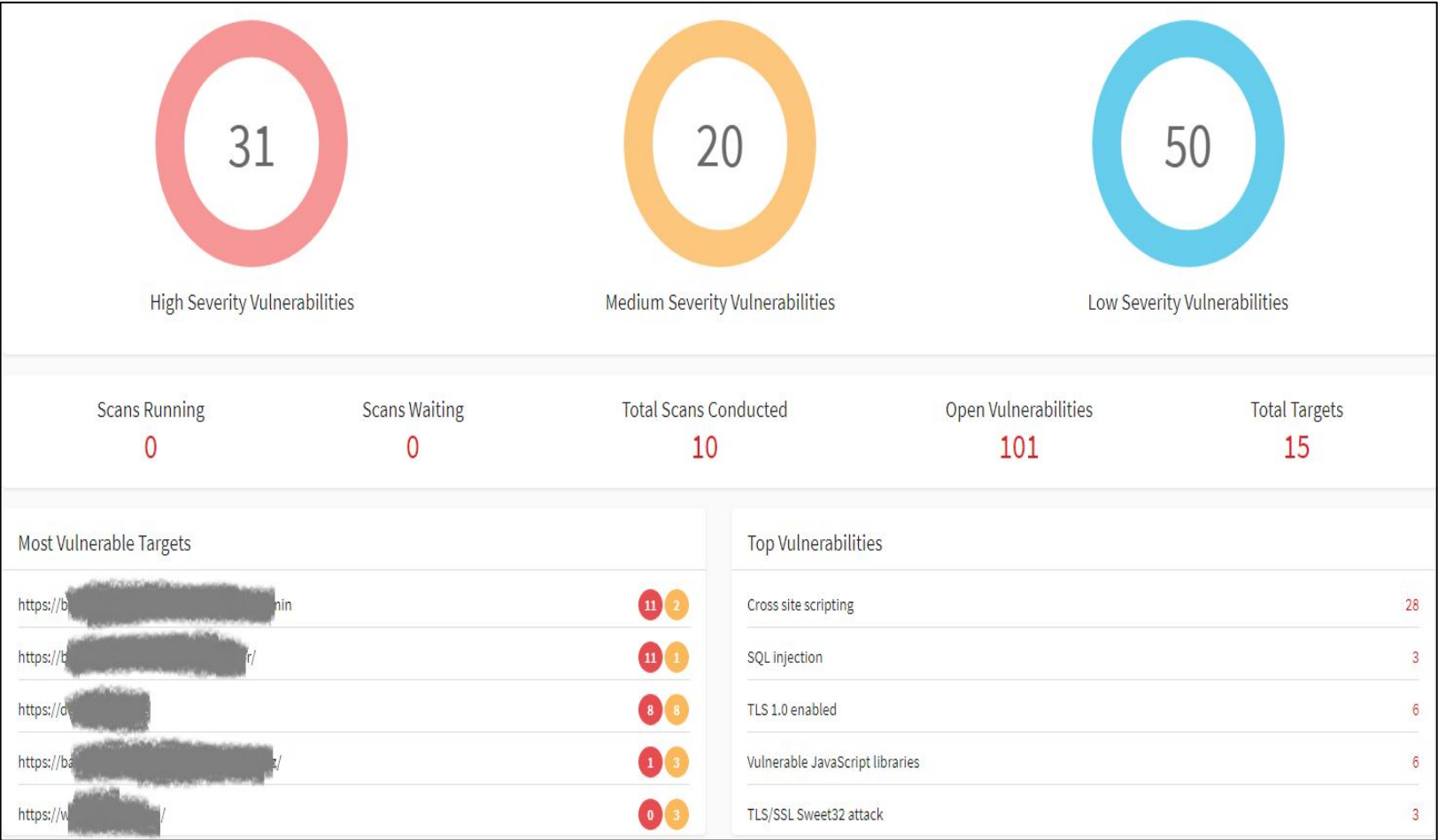
Teste Estático de Aplicativos (SAST):

O Teste Estático de Aplicativos (SAST) analisa o código-fonte em busca de vulnerabilidades de segurança antes da execução do programa. Incorporar ferramentas SAST no início do SDLC ajuda a identificar e corrigir falhas de segurança de maneira eficiente e preventiva.



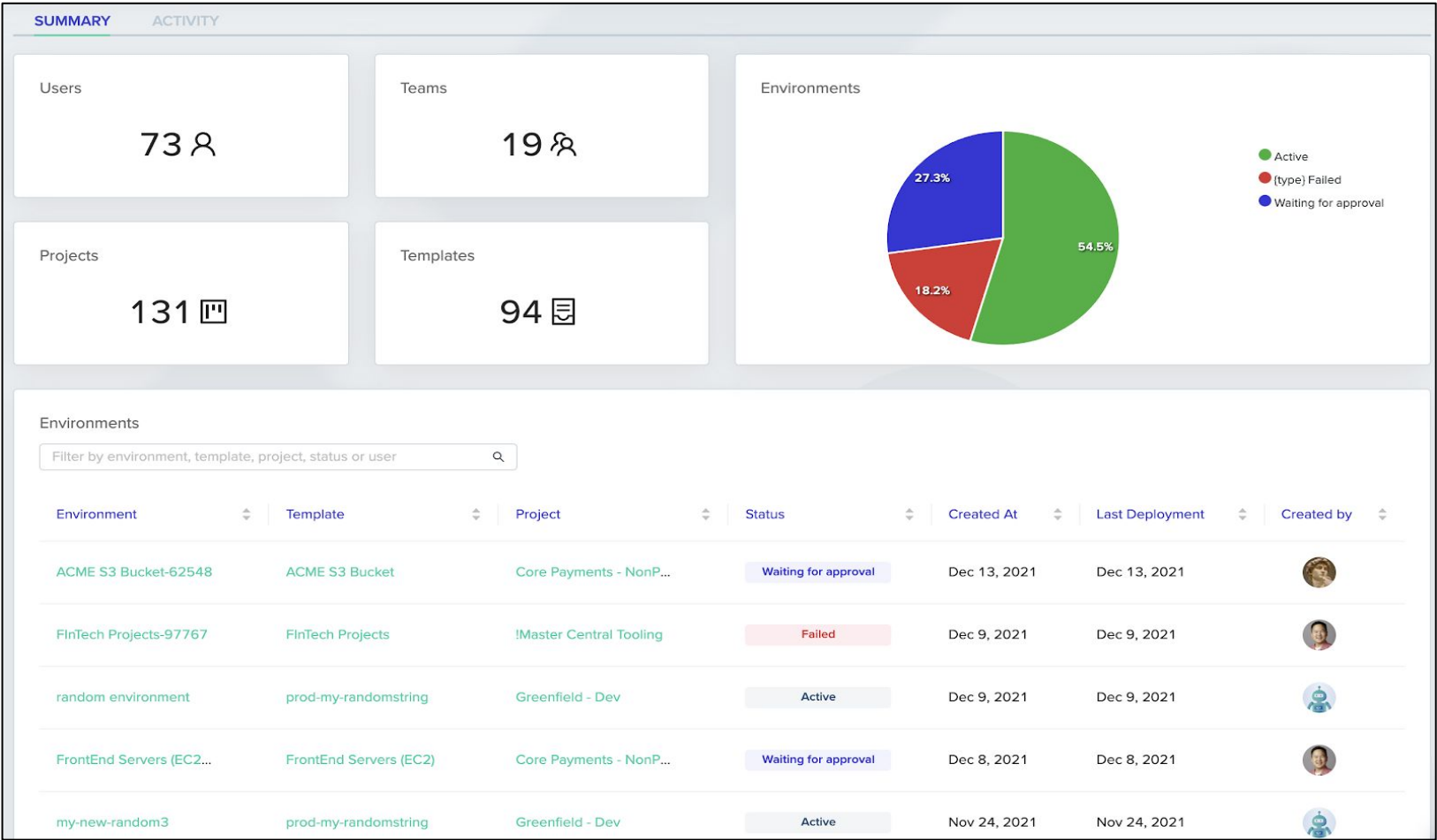
Teste Dinâmico de Aplicativos (DAST):

Complementar ao SAST, o Teste Dinâmico de Aplicativos (DAST) examina a aplicação em execução para detectar vulnerabilidades. Este teste simula ataques externos, fornecendo insights valiosos sobre a segurança da aplicação em condições reais.




Código como Infraestrutura (IaC) e Segurança:


O Código como Infraestrutura (IaC) transforma a gestão de infraestrutura em código automatizado, aumentando a eficiência. Incorporar práticas de segurança no IaC é crucial para evitar configurações inseguras e vulnerabilidades na infraestrutura de TI.



Análise de Composição de Software (SCA):

A análise de composição de Software (SCA) identifica vulnerabilidades em componentes de terceiros e bibliotecas usadas nas aplicações. Manter essas dependências atualizadas e monitoradas é essencial para a segurança da aplicação.

 zehavitprisma/cves-prisma-prod



Status

Errors

Category

Vulnerabilities

Severity

Select Severity

Tags

Select Tags

Code Status

main










View PR scan (4 Total)

8

lodash v3.1.0 (package.json)

Suppress

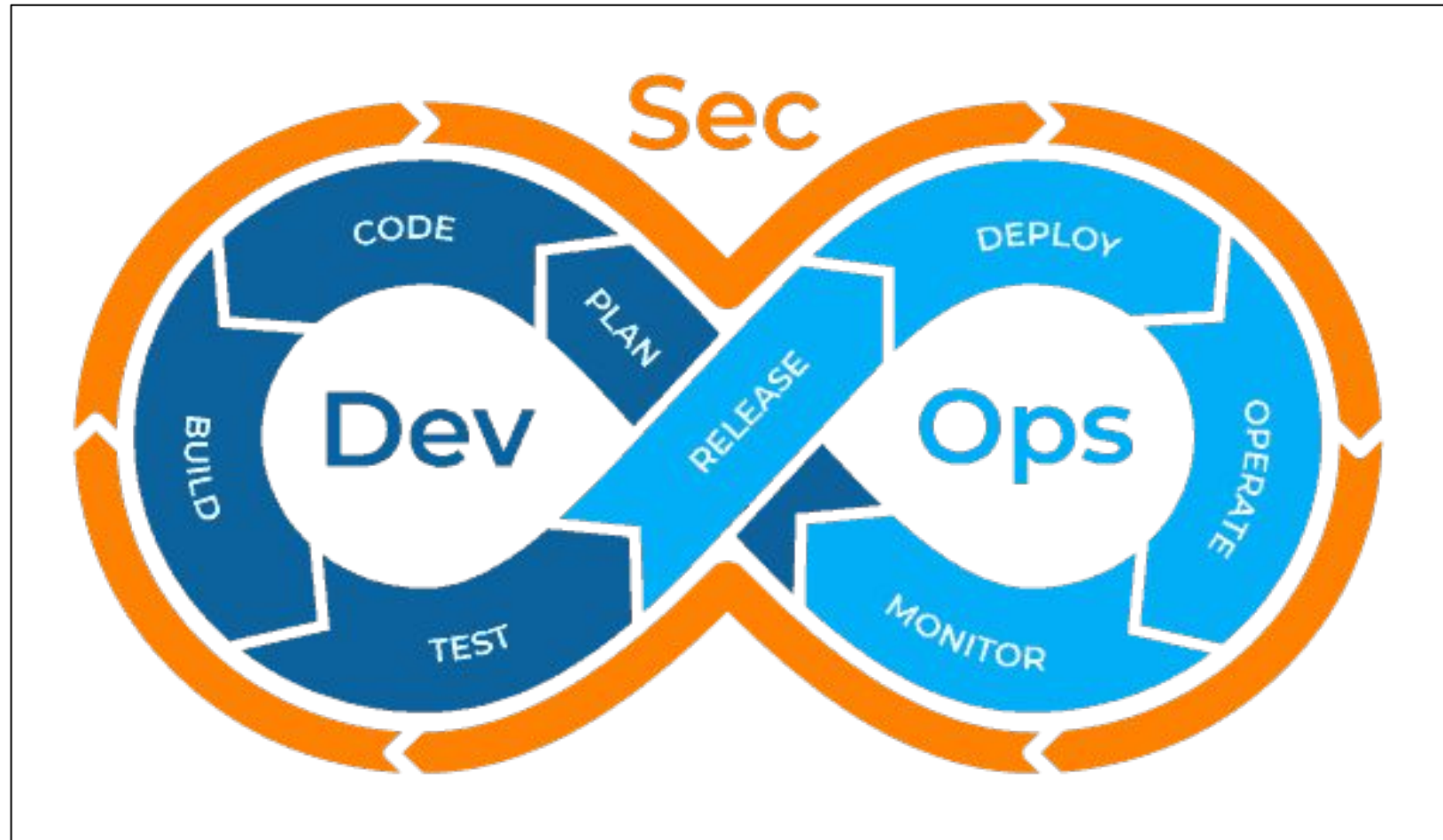
Fix

<input type="checkbox"/>	CVE ID	Bump to	CVSS	Risk factors	Published
<input type="checkbox"/>	 CVE-2019-1010266	4.17.11	 6		3 years ...
<input type="checkbox"/>	 CVE-2018-16487	4.17.11	 5		4 years ...
<input type="checkbox"/>	 CVE-2020-28500	4.17.21	 5		2 years ...

Show All

Integração Contínua e Entrega Contínua (CI/CD) com foco em Segurança:

A integração de práticas de segurança em pipelines CI/CD permite a detecção e correção automatizada de vulnerabilidades em cada commit ou pull request. Isso assegura que a segurança seja uma parte contínua do processo de desenvolvimento e implantação.



Cultura de Segurança e Treinamento:

Construir uma cultura de segurança dentro das equipes de desenvolvimento e promover treinamentos contínuos são fundamentais para a segurança das aplicações. Isso garante que as melhores práticas de segurança sejam conhecidas e aplicadas por todos os envolvidos.



Conclusão e Melhores Práticas:

Adotar uma abordagem integrada de segurança em todas as etapas do SDLC é vital para o desenvolvimento de software seguro. Ao aplicar as práticas recomendadas de AppSec, podemos proteger nossas aplicações contra ameaças emergentes e garantir a confiança dos usuários.



AFKConf

04/04/2024

FIM!