

DevSecOps

Um dos objetivos do DevOps é entregar atualizações de software rápidas e de alta qualidade. E para que o software seja de alta qualidade, ele precisa atender a requisitos rigorosos de cibersegurança. É aí que entra a segurança do DevOps, ou DevSecOps. Neste texto, você vai conhecer o DevSecOps, suas verificações de segurança e sua dependência da automação.

O que é DevSecOps?

DevSecOps é uma extensão do DevOps que automatiza as verificações de segurança em todo o ciclo de vida de desenvolvimento de software (SDLC – *Software Development Life Cycle*) para evitar vulnerabilidades no produto final. Uma **vulnerabilidade** é uma fraqueza em potencial, como a ausência de criptografia de dados, que alguém pode explorar em um sistema.

Tradicionalmente, os desenvolvedores escreviam a maior parte do código de produção sem pensar em segurança e somente no fim do SDLC uma equipe de segurança testava o software. Essa abordagem funciona quando as atualizações ocorrem apenas algumas vezes por ano, mas as equipes de DevOps produzem atualizações em poucas semanas ou em intervalos menores.

Com o DevSecOps, os desenvolvedores incorporam segurança em cada etapa do SDLC. As equipes consideram e planejam possíveis ameaças à segurança desde o início e testam, verificam, auditam e revisam o código durante o desenvolvimento.

Componentes

Os principais componentes do DevSecOps incluem responsabilidade compartilhada, velocidade e qualidade, verificações de segurança e automação.

Responsabilidade compartilhada

No DevSecOps, todos, as equipes de desenvolvimento, operações e segurança, compartilham a responsabilidade pela segurança.

- As equipes devem entender as estratégias básicas de segurança e mitigação de aplicativos.
- Elas devem seguir as atualizações do [Open Web Application Security Project \(OWASP\) Top 10](#), uma lista padrão do setor de riscos críticos de segurança para aplicativos da Web.

- Os desenvolvedores devem concordar e seguir práticas de codificação seguras.

Velocidade e qualidade

Problemas de segurança em um aplicativo exigem tempo e dinheiro para serem corrigidos, especialmente aqueles encontrados no fim do SDLC, e podem atrasar consideravelmente uma versão.

Com o DevSecOps, as equipes consideram a segurança desde o estágio de planejamento, e identificam e resolvem problemas com antecedência e rapidez. Dessa forma, elas podem continuar entregando atualizações pequenas, constantes e de alta qualidade.

Verificações de segurança

Com o DevSecOps, o software passa por inúmeras verificações de segurança em todo o SDLC. Vamos discutir algumas verificações padrão.

- A **modelagem de ameaças** é um processo no qual as equipes identificam e categorizam as ameaças de segurança para levar em conta no desenvolvimento e suporte de software. A modelagem de ameaças geralmente ocorre durante o estágio de design ou planejamento do desenvolvimento antes que os desenvolvedores escrevam o código.
- As **verificações de vulnerabilidades** identificam vulnerabilidades no aplicativo e nas bibliotecas (coleções de código reutilizável) de que o aplicativo depende. As equipes podem automatizar o patching para lidar com vulnerabilidades o mais rápido possível. Duas verificações comuns de vulnerabilidades são testes estáticos de segurança de aplicativos e testes dinâmicos de segurança de aplicativos.
- As ferramentas de **teste de segurança de aplicativos estáticos (SAST – Static Application Security Testing)** verificam vulnerabilidades dentro do código e de suas bibliotecas. "Estático" significa que o aplicativo não está sendo executado, está em repouso.
- As ferramentas de **teste dinâmico de segurança de aplicativos (DAST – Dynamic Application Security Testing)** detectam vulnerabilidades observáveis fora do código enquanto o aplicativo é executado. Para isso, essas ferramentas simulam técnicas de hackers reais, como injeção de SQL, para descobrir pontos fracos que os criminosos cibernéticos podem explorar.

- **Detecção de segredos** analisa a pesquisa de segredos que os programadores acidentalmente deixam em arquivos de código ou de configuração. Os segredos são credenciais sensíveis, como senhas e chaves de criptografia, e as organizações devem protegê-los contra vazamentos. Se os segredos chegarem à base de código da aplicação, os cibercriminosos poderão encontrá-los.
- **Testes de unidade** são testes que avaliam um único componente ou unidade de um aplicativo para verificar se o componente é executado corretamente. Esses testes são executados quando os desenvolvedores enviam código recém-escrito para integração na base de código principal do software. No DevSecOps, os desenvolvedores projetam testes de unidade adicionais, chamados **testes de unidade de segurança**, que verificam se há problemas de segurança.
- As ferramentas de **monitoramento de segurança** monitoram aplicativos em tempo real em busca de problemas de segurança, como ataques cibernéticos, e enviam alertas imediatos quando tal atividade ocorre. Dessa forma, os funcionários podem responder rapidamente para minimizar os danos e corrigir o aplicativo, se necessário.

Automação

A automação é essencial para DevOps e o DevSecOps não é diferente.

As ferramentas de integração contínua e entrega contínua (IC/EC) podem automatizar as verificações de segurança em quase todas as etapas do SDLC, liberando todos para se concentrarem em outras tarefas.

- As ferramentas de automação verificam se o código passa nos testes da unidade de segurança e se as dependências de software estão nos patches mais recentes.
- As ferramentas SAST detectam vulnerabilidades em novos códigos antes de os desenvolvedores os incorporarem na base de código.
- As ferramentas DAST avaliam atualizações em um ambiente de pré-produção.
- As ferramentas podem automatizar a configuração de sistemas e serviços, garantindo a conformidade com a segurança e reduzindo o erro humano.