DevSecOps

Guia orientativo

ANATEL

Agência Nacional de Telecomunicações

DEVSECOPS

Guia Orientativo

Brasília, outubro de 2023

Sumário

Lista de Abreviaturas, Acrônimos e Siglas

- CIS - *Center for Internet Security*
- DAST - *Dynamic Application Security Testing*
- DevOps - *Development and Operations*
- DevSecOps - *Development, Security, and Operations*
- IaC - *Infrastructure as Code*
- IAST - *Interactive Application Security Testing*
- IP – *Internet Protocol*
- MFA - Autenticação por Múltiplos Fatores
- NIST - *National Institute of Standards and Technology*
- OWASP - *Open Web Application Security Project*.
- PPSI - Programa de Privacidade e Segurança da Informação
- PMI - *Project Management Institute*
- RACI – *Responsible, Accountable, Consulted, Informed*
- RASP - *Runtime Application Self-Protection*
- SAST - *Static Application Security Testing*
- SBOM - *Software Bill of Materials*
- SCA - *Software Composition Analysis*
- SIEM - Sistemas de Gerenciamento de Eventos e Informações de Segurança
- TI – Tecnologia da Informação

Apresentação e Objetivo

O presente documento foi elaborado visando a disseminação de boas práticas em desenvolvimento e operação seguras de *software* no contexto de uma cultura de segurança cibernética para prestadoras de serviço de telecomunicações. O estudo aprofundado da temática DevSecOps foi proposto por profissionais de desenvolvimento de *software* das próprias prestadoras do setor. Destaca-se especialmente a liderança na elaboração de Jader Lucas de Souza Santos e contribuições de Nilson Luiz Tedeschi, colaboradores, respectivamente, dos Grupos Oi S/A e Claro S/A.

Dessa forma, o Guia foi elaborado de forma colaborativa no âmbito do Setor de Telecomunicações, utilizando-se do fórum do Grupo de Estudos do Exercício Guardião Cibernético 5.0 (EGC 5.0), bem como da estrutura do Grupo Técnico de Segurança Cibernética e Gestão de Riscos de Infraestrutura Crítica (GT-Ciber) da Anatel e construído com base nas deliberações do GT-Ciber. O GT-Ciber foi constituído pelo R-Ciber, possuindo uma série de atribuições relacionadas ao acompanhamento da Política de Segurança Cibernética e Gestão de Infraestrutura Crítica; à elaboração das definições complementares para implementação do R-Ciber; à conscientização, 4 capacitação, estudos e à interação com as Comissões Brasileiras de Comunicações (CBCs); dentre outras.

Conjugando a temática do Guia com as premissas estabelecidas no R-Ciber, salienta-se a norma de que "*pessoas naturais ou jurídicas envolvidas direta ou indiretamente na gestão ou no desenvolvimento das redes e serviços de telecomunicações devem atuar em Segurança Cibernética observando as seguintes diretrizes: [...] incentivar a adoção de conceitos de* security *by design e privacy by design no desenvolvimento e aquisição de produtos e serviços no setor de telecomunicações",* nos termos do art. 5, VIII, do R-Ciber, a qual dialoga diretamente com esse Guia.

Inicialmente, o documento foi desenhado para servir como referência orientativa às prestadoras de telecomunicações, mas sua aplicabilidade pode se estender para outras instituições que busquem orientações sobre a temática DevSecOps. A elaboração do texto final foi fruto de um trabalho conjunto entre profissionais das prestadoras de telecomunicações, sob a coordenação da Anatel, contendo referências em publicações tanto do *estado da arte* na literatura técnica quanto outros documentos técnicos de organismos de padronização, de governos e da indústria, com destaque para as publicações do *National Institute of Standards and Technology* (NIST) [1], da *Open Web Application Security Project (*OWASP) [2] e do Programa de Privacidade e Segurança da Informação (PPSI) da Secretaria de Governo Digital [3]. Algumas referências foram traduzidas de forma

livre pelos autores do documento, com propósitos não comerciais a fim de promover a educação e difusão de conhecimentos.

O objetivo principal desse Guia é a difusão das melhores práticas na adoção do DevSecOps na cultura de desenvolvimento seguro de *software* nas empresas e instituições interessadas no assunto.

Por fim, ressalva-se que a Anatel e as organizações que participaram do trabalho de desenvolvimento deste Guia:

a) não representam, tampouco se manifestam em nome do NIST, da OWASP, do PPSI, bem como de quaisquer outras instituições públicas e privados;

b) não são coautoras das publicações nacionais e internacionais abordadas; e

c) não assumem nenhuma responsabilidade administrativa, técnica ou jurídica pelo uso ou pela interpretação inadequados, fragmentados ou parciais do presente Guia.

Caso o leitor deseje se certificar de que o Guia atende integralmente os requisitos das publicações referenciadas, deverá consultar diretamente as fontes oficiais de informação ofertadas pelas referidas instituições.

Agradecimento especial ao NIST, OWASP, PPSI pelas valiosas contribuições para a promoção do desenvolvimento seguro.

Introdução

O Guia orientativo de DevSecOps é um documento elaborado pelas prestadoras de telecomunicações do Brasil, com supervisão e revisão da Anatel, para incentivar cultura de DevSecOps, bem como a adoção do conceito de *security by design* no desenvolvimento de *softwares* e aplicações no setor de telecomunicações, visando a construção de sistemas e aplicativos mais seguros, fornecendo diretrizes abrangentes para implementação e desenvolvimento da cultura de DevSecOps, tendo como premissas as recomendações fornecidas pelo NIST e OWASP.

Ressaltamos que cada um dos interessados na temática do DevSecOps é livre para adequar todas as proposições desse Guia à sua realidade. A abordagem proposta oferece uma sugestão de uso para a implantação do DevSecOps. Contudo, isso não exclui a necessidade de que o interessado compreenda sua própria postura de risco institucional. A intenção é cooperar para que cada organização possa concentrar seus esforços com base nos recursos disponíveis, implementando uma cultura mais centrada em segurança, sendo de inteira responsabilidade dos interessados a observância e o cumprimento das normas e da legislação brasileira vigentes. Este Guia será atualizado sempre que se fizer necessária a inclusão de ajustes para acompanhar o amadurecimento dos processos.

O documento Guia Orientativo de DevSecOps está organizado em capítulos e anexos conforme mostrados a seguir:

- No Capítulo 1, são introduzidos os conceitos fundamentais de DevSecOps.
- No Capítulo 2, são descritos os requisitos para implementação do DevSecOps.
- No Capítulo 3, é descrito o conceito de desenvolvimento seguro.
- No Capítulo 4, é apresenta a integração da segurança ao fluxo de desenvolvimento.
- No Capítulo 5, são descritos o monitoramento e a resposta a incidentes.
- No Capítulo 6, é descrita a governança e conformidade.
- No Capítulo 7, são tratados os conceitos de treinamento e de conscientização.
- Por fim no Capítulo 8, são apresentas as considerações finais, ressaltando os benefícios esperados com a aplicação do Guia pelos interessados.

Capítulo 1 - Fundamentos de DevSecOps

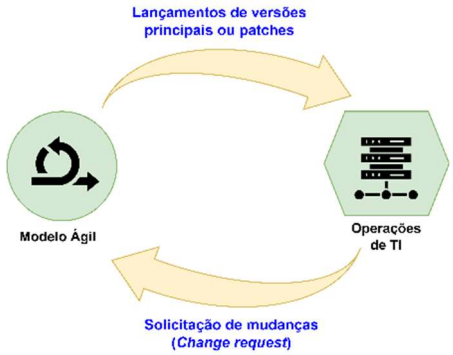
Neste Capítulo, são apresentados os fundamentos e conceitos básicos do DevOps e do DevSecOps.

1.1. Entendendo o conceito de DevOps

De acordo com as definições da organização internacional *Project Management Institute* – PMI, o DevOps pode ser definido como sendo a "*otimização das atividades relacionadas ao desenvolvimento de soluções de TI (Dev) e às operações de TI (Ops)*" [4].

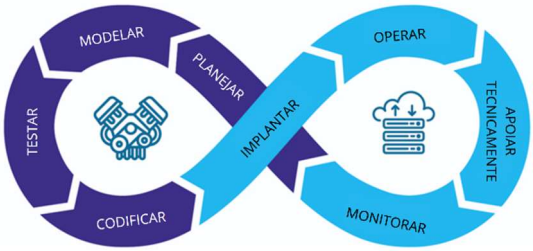
O modelo tradicional de implantação de sistemas, tem uma "lacuna DevOps", como representado na Figura 1, entre as equipes de entrega de soluções e as operações de TI. Essa lacuna resulta em implantações mais lentas e, portanto, custos mais elevados para implantar; um longo tempo médio entre implantações, frequentemente medido em meses; redução da competitividade no mercado; e uma capacidade reduzida de governar seus esforços de TI devido à falta de inteligência em tempo real.

Figura 1 – Modelo tradicional de lacuna DevOps



Ao remover as barreiras que inibem a colaboração eficaz, na prática, as empresas de *software* fecham a referida lacuna DevOps. Uma representação comum da estratégia chamada na literatura de "*loop* DevOps" é mostrada na Figura 2. As organizações adotam esse fluxo tendo como base uma mentalidade que promove maneiras de trabalhar de forma colaborativa, centradas no aprendizado, apoiadas por práticas ágeis, e frequentemente, investindo em automação de processos.

Figura 2 – Modelo *loop* DevOps

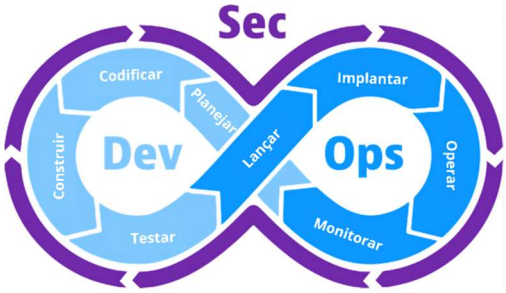


Fonte: PMI [4] – tradução livre

1.2. Visão geral do DevSecOps

O DevSecOps é um paradigma de desenvolvimento de *software* que enfatiza a cultura de colaboração, do Inglês *security as a culture* [5], entre as equipes de desenvolvedores “Dev”, Segurança “Sec” e equipes de operações de TI “Ops”, com o objetivo de integrar a segurança ao longo de todo o ciclo de vida do desenvolvimento de *software* [6]. Essa abordagem busca eliminar as lacunas existentes entre equipes de desenvolvimento, segurança e operações, promovendo uma colaboração mais estreita e uma cultura de responsabilidade compartilhada pela segurança, colocando a segurança como o centro de todo o processo. A ilustração desse processo otimizado é apresentada na Figura 3.

Figura 3 – Modelo DevSecOps



Fonte: [7] – tradução livre

Tradicionalmente, a segurança sempre foi considerada uma etapa posterior ao desenvolvimento de *software*, geralmente adicionada durante as fases finais antes da implantação e, frequentemente, com o tempo de testes reduzido. No entanto, essa abordagem apresenta-se ineficiente na proteção de sistemas e dados contra ameaças cada vez mais sofisticadas. O DevSecOps procura mudar essa cultura, integrando a segurança em cada etapa do ciclo de desenvolvimento de *software*, desde o planejamento até a etapa de pós-produção. O processo de trazer a segurança para os estágios iniciais do desenvolvimento de *software* recebeu o nome de "*Shifting Left*".

Com a segurança integrada ao processo de desenvolvimento, é possível identificar e corrigir vulnerabilidades de forma antecipada, antes da fase tradicional do *pentest* e, consequentemente, experimentar otimização no processo. Posteriormente, quando o *software* for lançado no mercado, as vulnerabilidades identificadas já terão sido preventivamente sanadas, evitando a exploração por criminosos cibernéticos e reduzindo significativamente o risco de ataques.

É importante lembrar que a implementação de DevSecOps tanto requer uma mudança de cultura na organização, quanto investimentos em capacitação dos times de desenvolvimento. Também se faz necessária a seleção das ferramentas corretas para integrar a segurança de forma contínua. A implementação de DevSecOps implicará em constante estímulo ao desenvolvimento dos times de operações, em uma cultura centrada na segurança desde a concepção (*security by design*) dos produtos.

Capítulo 2 - Preparação para a Implementação do DevSecOps

Neste Capítulo, são apresentados os passos para a preparação da implementação de DevSecOps.

2.1. Avaliação de segurança e conformidade inicial

A implementação de DevSecOps é um processo complexo que requer a integração de segurança na cultura e nos processos de desenvolvimento, operações e segurança. Para garantir que a implementação de DevSecOps seja bem-sucedida, é crucial realizar uma avaliação de segurança e conformidade para identificar as necessidades e requisitos específicos da empresa.

A OWASP desenvolveu um guia [7] de quatro níveis de maturidade e cinco níveis de controles, que ajuda as organizações a avaliarem e melhorarem sua maturidade em DevSecOps. A partir desse guia, as empresas podem definir suas metas e planejamentos para atingir níveis superiores de maturidade. Além do guia disponibilizado pela OWASP, as empresas devem analisar o estado atual dos processos, ferramentas e práticas de segurança, bem como verificar a conformidade com regulamentações e normas nacionais aplicáveis à sua área de atuação.

2.2. Papéis e responsabilidades no DevSecOps

A abordagem DevSecOps visa integrar a segurança no ciclo de vida do desenvolvimento e operações de *software*, tornando-a uma responsabilidade compartilhada entre desenvolvedores, profissionais de segurança e operações. Os papéis e responsabilidades em DevSecOps, podem ser listados como:

a) **Desenvolvedores:** Responsáveis pela criação e manutenção do código seguro. Eles devem considerar a segurança como uma responsabilidade compartilhada e não apenas como uma tarefa exclusiva dos profissionais de segurança;
b) **Profissionais de segurança:** Responsáveis pela análise de vulnerabilidades, riscos, conformidades, modelagem de ameaças e resposta a incidentes. Eles devem trabalhar em conjunto com os desenvolvedores e operações desde o início do projeto, planejando, modelando ameaças e participando desde as primeiras iterações do produto;
c) **Operações:** Responsáveis pela infraestrutura, monitoramento e manutenção do ambiente. Eles devem garantir que a infraestrutura esteja em conformidade com as normas de

segurança e que as vulnerabilidades sejam identificadas e corrigidas rapidamente. As equipes de operações também devem colaborar com as equipes de desenvolvimento e segurança para garantir que a segurança seja considerada em todas as etapas do processo;

d) **Liderança e equipe DevSecOps:** Responsáveis por engajar e a formar uma equipe multidisciplinar que inclua desenvolvedores, profissionais de segurança e operações. A liderança deve promover também uma cultura de colaboração e comunicação entre as equipes, incentivando a responsabilidade compartilhada pela segurança.

É recomendada a criação e utilização de uma matriz RACI para todos os envolvidos no processo DevSecOps. A matriz RACI 0 – *Responsible, Accountable, Consulted, Informed* (Responsável, Aprovador, Consultor, Autoridade a ser informada) é uma ferramenta de gestão que ajuda a definir e documentar as responsabilidades e atribuições de cada membro da equipe em cada etapa do projeto, promovendo clareza e melhor comunicação entre os participantes. Ao aplicar a matriz RACI no contexto do DevSecOps, a definição de papéis e responsabilidades durante todo o processo de desenvolvimento e implantação de soluções tecnológicas fica centralizada, garantindo que todos compreendam suas funções específicas e trabalhem de forma colaborativa para alcançar os objetivos de segurança.

Capítulo 3 - Desenvolvimento Seguro

O desenvolvimento seguro é um componente fundamental da cultura DevSecOps, pois garante que os aplicativos sejam projetados, desenvolvidos e implantados com segurança desde o início. Neste Capítulo, serão apresentados os conceitos de desenvolvimento seguro aplicados a DevSecOps. Para mais detalhes sobre desenvolvimento seguro e ciclo de vida do desenvolvimento de *software*, acesse: Framework de Desenvolvimento de Software Seguro 0.

3.1. Princípios de segurança e desenvolvimento seguro

a) **Defesa em profundidade**: Segundo consta em documentação do NIST 0, esse princípio se trata da estratégia de Segurança da Informação que integra pessoas, tecnologia e capacidades operacionais para estabelecer barreiras variáveis em várias camadas e missões da organização;

b) **Princípio do menor privilégio**: De acordo com a documentação do NIST 0, esse princípio define que uma arquitetura de segurança deve ser projetada de forma que cada entidade receba os recursos e autorizações mínimas do sistema necessárias para desempenhar sua função;

c) **Princípio de privacidade por design:** Segundo o guia de Requisitos Mínimos de Privacidade e Segurança da Informação para APIs 0, Privacidade desde o projeto inicial (*Privacy by Design*), é incorporar medidas protetivas de privacidade e dados pessoais, em todas as fases dos projetos em desenvolvimento, não sendo permitido desenvolver nenhum projeto, produto ou serviço sem que os princípios de proteção da privacidade estejam no centro desse desenvolvimento;

d) **Modelagem de ameaças:** De acordo com a documentação do OWASP 0, é uma abordagem usada para identificar, comunicar e entender ameaças e medidas de mitigação no contexto de proteger algo de valor, sendo uma representação estruturada de todas as informações que afetam a segurança de uma aplicação. Essencialmente, é uma visão da aplicação e de seu ambiente por meio da perspectiva da segurança. Mais detalhes disponíveis em: *Threat Modeling Process* 0*;*

e) **Gerenciamento de vulnerabilidades:** É realizado através processo de gerenciamento de vulnerabilidades para acompanhar e corrigir as vulnerabilidades identificadas durante o desenvolvimento e a operação do *software* pré-produção e pós-produção;

f) **Educação e conscientização:** Todos os membros da equipe de desenvolvimento devem ser treinados em boas práticas de segurança e atualizados regularmente sobre as últimas ameaças e técnicas de ataque. Isso ajuda a criar uma cultura de segurança e responsabilidade compartilhada entre as equipes.

3.2. Ambientes de Desenvolvimento

O Ambiente de Desenvolvimento é o ambiente em que os desenvolvedores criam e testam o *software*. Ele deve ser seguro e confiável para garantir a qualidade do código e a prevenção de vulnerabilidades. Alguns controles a serem implementados são:

a) Todo desenvolvimento de sistemas deve empregar as ferramentas e produtos definidos pela Diretoria de TI;
b) Qualquer ferramenta utilizada para o desenvolvimento de sistemas deve possuir licença regularizada com o fornecedor e autorização de uso do responsável pela custódia do produto;
c) Ferramentas de controle de versionamento devem ser utilizadas para o desenvolvimento e a manutenção de sistemas;
d) O controle de acesso deve ser devidamente configurado para garantir que os desenvolvedores tenham apenas as permissões necessárias;
e) Os ambientes utilizados para desenvolvimento, homologação e produção dos sistemas e aplicativos devem ser independentes através do uso segregado por servidores, domínios, diretórios ou instâncias de banco de dados.

3.3. Testes de Segurança

Os testes de segurança são um processo de avaliação do *software* para identificar vulnerabilidades e falhas de segurança. Eles podem ser realizados em qualquer fase do ciclo de vida do desenvolvimento. Algumas medidas que devem ser observadas são:

a) Todo sistema ou aplicação desenvolvido pela ou para a empresa deve ser submetido a testes de segurança definidos pela Diretoria de Segurança antes da entrada em produção;
b) Todo sistema, aplicação e/ou aplicativos, quando pronto para serem movidos para os ambientes produtivos, deverão ser submetidos a um processo formal de gestão das mudanças que contemple, no mínimo: documentação do sistema, testes funcionais, aprovação dos gestores envolvidos e plano de retorno (*rollback*);

c) Segundo os critérios de Realização de Testes de Segurança em Sistemas, Aplicativos e Aplicações *Web*, não poderão entrar em produção sistemas que tenham apresentado determinados níveis de exposição a vulnerabilidades.

3.4. Documentação e Boas práticas de codificação segura

Todo sistema deve possuir Manuais de Usuário, Sistema e Produção, gerados e atualizados, obrigatoriamente, pela área de TI responsável do projeto / melhoria que originou a sua necessidade de criação ou atualização.

A OWASP fornece uma lista abrangente de práticas de codificação segura, conhecida como OWASP *Secure Coding* 0. As recomendações para o desenvolvimento seguro apresentadas nesse documento são:

a) **Validação de Entrada**: Garanta que todas as entradas do usuário sejam validadas antes de serem processadas pelo sistema. Isso ajuda a prevenir ataques, como injeção de SQL e *cross-site scripting* (XSS);
b) **Codificação de Saída**: Utilize técnicas de codificação para garantir que os dados enviados ao cliente sejam interpretados corretamente e não causem problemas de segurança;
c) **Autenticação e Gerenciamento de Senhas**: Implemente autenticação multifatorial e métodos de autenticação sem senha para garantir a segurança das credenciais dos usuários;
d) **Gerenciamento de Sessão**: Monitore e gerencie sessões de usuário para evitar sequestro de sessão e outros ataques relacionados à sessão;
e) **Controle de Acesso**: Estabeleça políticas de controle de acesso para garantir que apenas usuários autorizados tenham acesso aos recursos do sistema;
f) **Práticas Criptográficas**: Utilize criptografia para proteger dados sensíveis e garantir a segurança das comunicações entre o cliente e o servidor;
g) **Tratamento de Erros e Registro de Logs**: Implemente um sistema de registro de logs para monitorar e analisar eventos de segurança e garantir que os erros sejam tratados adequadamente;
h) **Proteção de Dados**: Proteja os dados armazenados e em trânsito usando técnicas de criptografia e gerenciamento de chaves;
i) **Segurança de Comunicação**: Garanta a segurança das comunicações entre os componentes do sistema e entre o sistema e os usuários;
j) **Configuração do Sistema**: Mantenha as configurações do sistema atualizadas e seguras, seguindo as melhores práticas e diretrizes de segurança;

k) **Segurança de Banco de Dados**: Proteja os bancos de dados contra ataques e vazamentos de dados, utilizando técnicas como criptografia, controle de acesso e monitoramento;
l) **Gerenciamento de Arquivos:** Implemente políticas e práticas de gerenciamento de arquivos para garantir a segurança dos dados armazenados e prevenir vazamentos de informações;
m) **Gerenciamento de Memória**: Utilize técnicas de gerenciamento de memória seguras para evitar vulnerabilidades, como estouro de buffer e vazamento de memória;
n) **Práticas Gerais de Codificação**: Siga as diretrizes e melhores práticas de codificação segura, como as fornecidas pela OWASP.

Cada linguagem de programação ou *framework* possui seus próprios recursos, bibliotecas e controles de segurança. Portanto, além de seguir as recomendações gerais de desenvolvimento seguro, os desenvolvedores devem buscar formas de aplicar práticas de segurança específicas para a linguagem e o *framework* escolhidos, devendo o desenvolvedor estar atento às atualizações e novos recursos de segurança disponíveis, o que inclui acompanhar boletins de segurança, verificar atualizações de bibliotecas e manter-se atualizado sobre as melhores práticas de segurança específicas.

Ao utilizar bibliotecas de terceiros, é fundamental realizar uma análise de segurança antes de incorporá-las ao projeto. Sempre verificar o código-fonte, consultar boletins de segurança do fornecedor e verificar se a biblioteca possui vulnerabilidades conhecidas. Tais controles são imperativos antes de realizar implementação de códigos ao projeto desenvolvido.

Capítulo 4 - Integração de Segurança ao Fluxo de Desenvolvimento

Neste capítulo, serão apresentados os conceitos de integração da segurança nas etapas do ciclo de vida do desenvolvimento de *software*, buscando garantir que a segurança seja considerada em toda a fase do processo de Integração Continua (CI), Entrega Continua (CD) e produção.

4.1. Ferramentas de Segurança em uma esteira CI/CD

A CI e a CD são práticas de desenvolvimento de *software* que visam tornar a integração de código mais eficiente por meio de *builds* e testes automatizados, permitindo a entrega de aplicações com mais frequência aos clientes. As ferramentas de segurança são usadas para fazer a identificação de vulnerabilidades, análise de código e na garantia da conformidade com as melhores práticas de segurança. As principais ferramentas de segurança que podem ser incorporadas à esteira CI/CD são:

a) **SAST (*Static Application Security Testing*):** técnica que analisa o código-fonte e/ou o *bytecode* em busca de vulnerabilidades de segurança. Pode identificar problemas como vulnerabilidades conhecidas, vulnerabilidades de configuração, uso inseguro de APIs, possibilidade de injeção SQL e outras más práticas de codificação;

b) **DAST (*Dynamic Application Security Testing*):** Abordagem de teste de segurança que examina um aplicativo em execução em busca de vulnerabilidades. Testa o aplicativo de fora, simulando ataques e procurando por pontos fracos que possam ser explorados, por isso, esse teste deve ser executado após o *deploy* da aplicação;

c) **IaC (*Infrastructure as Code*) Scanning:** Prática de segurança que envolve a análise dos arquivos de configuração em modelos de IaC em busca de vulnerabilidades e problemas de conformidade;

d) **SCA (*Software Composition Analysis*):** Usado para identificar e rastrear as bibliotecas de terceiros e componentes de código aberto utilizados em um aplicativo detectando vulnerabilidades conhecidas nessas bibliotecas, garantindo que apenas as versões seguras sejam utilizadas;

e) **SBOM (*Software Bill of Materials*):** Inventário de todos os componentes e dependências de software envolvidos no desenvolvimento e entrega de uma aplicação;

f) **IAST (*Interactive Application Security Testing*)**: Monitora a aplicação em tempo real durante a execução, identificando vulnerabilidades e fornecendo informações detalhadas sobre como explorá-las;

g) **Proteção de repositórios Git:** Ajudam a garantir que informações sensíveis não sejam expostas acidentalmente no histórico de *commits* ou nos arquivos do repositório;

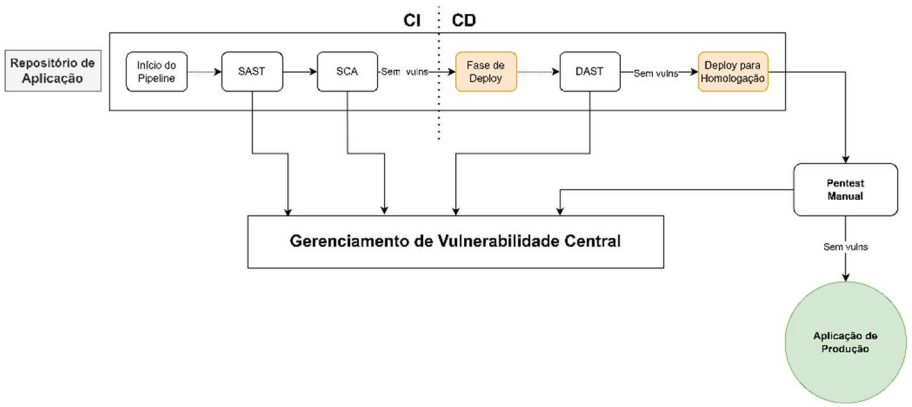
h) ***Container scan*:** A varredura de contêineres é o processo de analisar contêineres e seus componentes para identificar possíveis ameaças à segurança. As ferramentas em geral ajudam a identificar vulnerabilidades conhecidas nas bibliotecas e componentes utilizados no *software*, garantindo que apenas as versões seguras e atualizadas sejam utilizadas;

i) **RASP (*Runtime Application Self-Protection*):** Tecnologia de segurança que usa a instrumentação em tempo de execução para detectar e bloquear ataques a aplicativos de computador, aproveitando informações de dentro do *software* em execução. Ela protege o ambiente de execução de alterações e adulterações indesejadas, permitindo a identificação de vulnerabilidades e a prevenção de ataques em tempo real.

4.2. Esteira de Desenvolvimento Sugerida

Seguindo a recomendação da OWASP 0, para as empresas que desejam implementar uma esteira de desenvolvimento integrando as ferramentas de segurança ao processo de desenvolvimento de *software*, a abordagem mais assertiva é começar com algumas ferramentas e posteriormente evoluir a esteira cobrindo as demais necessidades da companhia, implementando em fases as demais ferramentas. A primeira fase: Segurança de Aplicações, encontra-se descrita na Figura 4. Já a segunda fase: Segurança de Aplicações e Infraestrutura, encontra-se descrita na Figura 5.

Figura 4 – Fase 1: Segurança de Aplicações

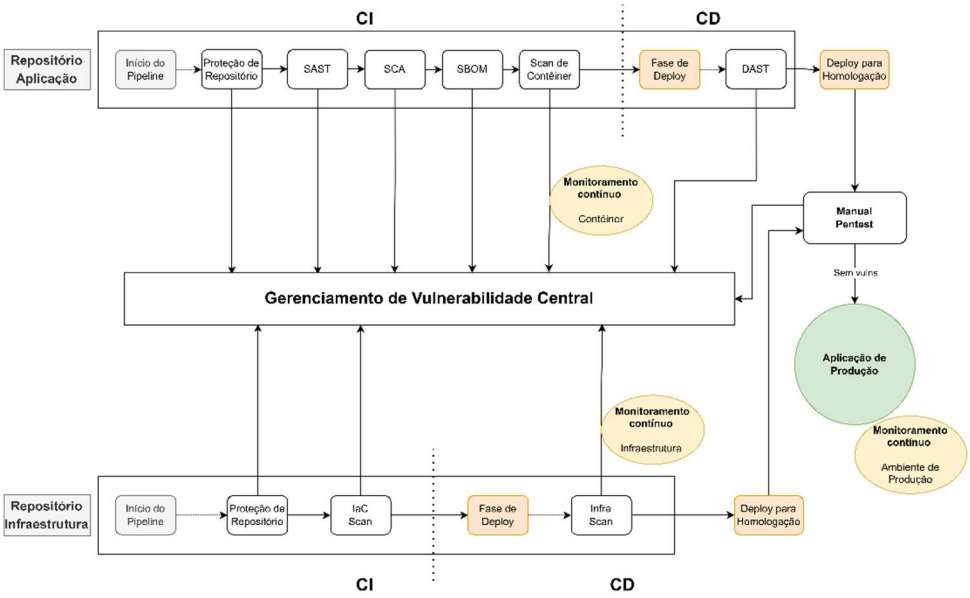


A abordagem de dividir a implantação das ferramentas de segurança em fases, traz diversas vantagens significativas para as equipes de desenvolvimento. Essas fases foram definidas com base

na complexidade e no impacto das ferramentas, permitindo uma implementação mais eficiente e efetiva da segurança em todo o ciclo de vida do desenvolvimento de *software*.

A primeira fase envolve um conjunto menor de ferramentas de segurança. Nela, as equipes podem iniciar o processo de integração de segurança. As primeiras ferramentas permitem que as equipes se familiarizem e se adaptem gradualmente às práticas de segurança, sem sobrecarregá-los com muitas ferramentas desde o início. Essa abordagem gradual facilita a aceitação e a adoção por parte dos desenvolvedores, tornando a transição mais suave. Outra vantagem é que a equipe tem a oportunidade de compreender completamente cada ferramenta, interpretar e ajustar os resultados obtidos e entender como as ferramentas se integram ao fluxo de trabalho existente, proporcionando um ambiente de aprendizado mais focado e iterativo.

Figura 5 – Segurança de Aplicações e Infraestrutura



Na segunda fase, com a adição de mais ferramentas de segurança, a equipe vai ampliar a cobertura e a detecção de vulnerabilidades e ameaças em sua aplicação e infraestrutura. Isso permitirá identificar e resolver uma gama mais ampla de problemas de segurança, garantindo que os aplicativos estejam protegidos contra diversas ameaças.

É importante ressaltar que cada empresa tem suas próprias necessidades e desafios de segurança. Sendo assim, é crucial que cada companhia avalie cuidadosamente quais ferramentas de segurança são mais importantes para seu negócio específico. A esteira proposta anteriormente serve como um modelo conceitual base, devendo ser estudada, adaptada e personalizada de acordo com as exigências e requisitos específicos de cada empresa.

4.3. Gerenciamento de vulnerabilidades e patches

O gerenciamento de vulnerabilidades é uma prática que envolve a identificação, avaliação, priorização e correção de falhas de segurança em dispositivos, redes e aplicações, tanto em fase de desenvolvimento, quanto produção, a fim de reduzir os riscos de ataques cibernéticos e violações. De forma complementar, a gestão de *patches* descreve os processos e as ferramentas criadas para detectar, distribuir e implantar atualizações de *software* (*patches*) com eficácia nos sistemas. Ambos são componentes complementares e essenciais para manter a segurança contínua dos sistemas e aplicativos em uma companhia, e consequentemente, da abordagem de DevSecOps, que visa a segurança no desenvolvimento ágil.

Algumas das práticas recomendadas para gerenciamento de vulnerabilidades e patches em DevSecOps, consistem em automatizar os de testes de segurança, visando identificar vulnerabilidades e aplicar correções de forma mais rápida e eficiente, reduzindo a janela de oportunidade para os agentes de ameaças explorarem as vulnerabilidades. Garantir que as configurações de segurança sejam aplicadas corretamente e de forma consistente em todos os ambientes, desde o desenvolvimento até a produção, utilizando frameworks de mercado, como o *CIS Benchmark* 0 e as configurações do fabricante que indicam as melhores práticas de segurança para aquele ativo.

Uma outra etapa consiste em promover a mentalidade de que todos são responsáveis pela segurança e garantir que os membros da equipe estejam cientes das melhores práticas de gerenciamento de vulnerabilidades e *patches*, é um aspecto importante, que deve ser levado em consideração por toda a organização, incluindo os times de operações e de negócios, que devem entender que o tratamento de vulnerabilidades deve ser priorizado junto com as demais atividades. Para mais detalhes sobre gerenciamento de vulnerabilidades, acesse: Guia de Gerenciamento de Vulnerabilidades Informação [3].

Capítulo 5 - Monitoramento e Resposta a Incidentes

O monitoramento contínuo de segurança é uma prática que envolve a coleta, análise e resposta a eventos de segurança em tempo real ou quase real, permitindo que as empresas identifiquem e respondam rapidamente a ameaças e vulnerabilidades, minimizando o risco e o impacto de incidentes de segurança. Neste capítulo, serão apresentadas a importância do monitoramento contínuo de segurança e a resposta a incidentes. Para mais informações, acesse os documentos 0 e 0.

5.1. Monitoramento contínuo de segurança

O monitoramento contínuo de segurança é uma parte crucial, pois permite que as equipes de segurança e operações identifiquem e corrijam problemas de segurança rapidamente, reduzindo o risco de exploração por atores mal-intencionados. Alguns dos componentes-chave do Monitoramento Contínuo de Segurança, são:

a) **Coleta de dados**: É o primeiro passo no monitoramento contínuo de segurança. Isso envolve a coleta de informações de várias fontes, como logs de aplicativos, logs de servidores, logs de rede e sistemas de detecção de intrusão;
b) **Análise de dados**: Identificação de eventos de segurança e padrões suspeitos. Isso pode ser feito usando ferramentas de análise de *logs*, sistemas de gerenciamento de eventos e informações de segurança (SIEM) e soluções de inteligência de ameaças;
c) **Resposta a incidentes**: Minimização do impacto e o risco. Isso pode incluir ações como bloquear endereços IP maliciosos, aplicar patches de segurança ou reconfigurar sistemas afetados;
d) **Aprendizado e melhoria contínua**: O monitoramento contínuo de segurança não é uma atividade única. As equipes de DevSecOps devem aprender com os incidentes de segurança e usar essas informações para melhorar continuamente suas práticas e processos de segurança.

Existem várias ferramentas e tecnologias disponíveis para ajudar as equipes de DevSecOps a implementar o monitoramento contínuo de segurança. Algumas das opções populares incluem os Sistemas de Gerenciamento de Eventos e Informações de Segurança (SIEM), as ferramentas de Análise de Log, as soluções de Detecção de Intrusão e as ferramentas de Inteligência de Ameaças**.** Ao implementar um sistema de monitoramento contínuo de segurança, as empresas podem obter uma visão mais clara do cenário de ameaças e responder proativamente a incidentes de segurança, reduzindo o tempo de resposta a ameaças.

5.2. Detecção e resposta a incidentes

A detecção e resposta a incidentes permite uma abordagem proativa para lidar com ameaças em potencial. Embora os esforços sejam feitos para construir sistemas seguros desde o início, é necessário reconhecer que existem atacantes habilidosos e persistentes, e que eles podem encontrar vulnerabilidades para explorá-las. Portanto, o monitoramento constante e a prontidão para responder a incidentes são essenciais para minimizar danos e garantir a resiliência do sistema.

Alguns dos componentes-chave da Detecção e Resposta a Incidentes para DevSecOps, são:

a) **Definição de papéis e responsabilidades**: As funções e responsabilidades da equipe de resposta a incidentes, devem estar bem documentadas e identificadas. Isso inclui a designação de um coordenador de resposta a incidentes, analistas de segurança, especialistas técnicos e outros membros relevantes da equipe;
b) **Preparação**: A preparação envolve a criação de um plano de resposta a incidentes que define os procedimentos a serem executados quando um incidente ocorrer, as ferramentas, tecnologias e recursos a serem utilizados, além de especificar os colaboradores que fazem parte do processo e quais são suas responsabilidades e ações;
c) **Detecção:** A detecção envolve o uso de ferramentas e tecnologias para identificar eventos de segurança e atividades suspeitas em tempo real. Isso pode incluir sistemas de gerenciamento de eventos e informações de segurança, soluções de detecção de intrusão e ferramentas de análise de logs;
d) **Análise:** Após a detecção de um incidente, é necessário analisar as informações coletadas para determinar a extensão do problema, seus impactos e as ações corretivas necessárias;
e) **Resposta**: A resposta envolve a execução das ações corretivas identificadas na etapa de análise e documentadas no plano de resposta a incidentes, como bloquear endereços IP maliciosos, aplicar *patches* de segurança ou reconfigurar sistemas afetados;
f) **Recuperação**: A recuperação envolve a restauração dos sistemas afetados ao seu estado normal e a implementação de medidas para prevenir incidentes futuros;
g) **Aprendizado e melhoria contínua:** As equipes de DevSecOps devem aprender com os incidentes de segurança e usar essas informações para melhorar continuamente suas práticas e processos de segurança;
h) **Registro de incidentes:** Documentar e manter um registro de todos os incidentes de segurança, incluindo detalhes do incidente, ações tomadas, lições aprendidas e recomendações para futuras melhorias. Isso ajudará na análise e no desenvolvimento contínuo do programa de segurança.

Além de estabelecer um plano de resposta a incidentes, é fundamental realizar testes de simulação periódicos para testar a eficiência e a eficácia do plano. As simulações de incidentes, também conhecidas como exercícios de resposta a incidentes, fornecem uma oportunidade valiosa para treinar a equipe, identificar lacunas e aprimorar os processos de resposta. Durante as simulações, as equipes podem testar diferentes cenários de incidentes, permitindo que cada membro pratique suas responsabilidades e avalie a coordenação e a comunicação entre os envolvidos. Isso ajuda a identificar possíveis pontos fracos e ajustar o plano de resposta a incidentes. A realização de testes de simulação não só ajuda no estabelecimento de planos mais robustos de respostas a incidentes, colaborando, também na análise cuidadosa dos incidentes e na garantia de respostas eficazes aos esses incidentes de segurança.

Capítulo 6 - Governança e Conformidade

Neste capítulo, serão apresentados alguns aspectos que devem ser observados no contexto de Governança e conformidade na implantação de DevSecOps.

6.1. Políticas de segurança da informação e gerenciamento de acesso

Sobre gerenciamento de acesso, é importante tomar medidas com relação ao acesso privilegiado, que se refere às políticas de segurança da informação são diretrizes e regras que estabelecem as práticas e os procedimentos necessários para proteger os ativos de informação da organização. Elas são essenciais por vários motivos, como definir os controles necessários para garantir a confidencialidade, integridade e disponibilidade dos ativos da empresa. Para mais detalhes sobre políticas de segurança da informação, acesse: Guia do Framework de Privacidade e Segurança da Informação [3].

O acesso privilegiado refere-se às permissões e privilégios concedidos a usuários ou contas que têm autoridade significativa sobre sistemas, redes e dados sensíveis. Esses usuários privilegiados, como administradores de sistemas e administradores de bancos de dados, possuem amplos poderes e acesso a recursos críticos, o que os torna um alvo para atacantes. O gerenciamento adequado do acesso privilegiado é essencial pois possibilita a redução de riscos e aplica o conceito de menor privilégio.

Ao implementar o gerenciamento de acesso privilegiado, as melhores práticas como, identificação e categorização de usuários privilegiados, políticas de acesso, autenticação por múltiplos fatores (MFA), monitoramento continuo, rotação de credenciais, revisões de auditoria periódicas, limitações de acesso baseados em tempo, e modelos de controle de acesso baseado em função (RBAC) e/ou baseado em atributo (ABAC), devem ser considerados. Para mais detalhes sobre gestão de acesso privilegiado, acesse o guia: Modelo de Política de Gestão de Controle de Acesso 0.

6.2. Auditorias de segurança e conformidade com regulamentações e normas

As auditorias de segurança envolvem a avaliação sistemática dos sistemas, processos e práticas de uma organização para identificar vulnerabilidades e garantir a eficácia das medidas de segurança implementadas. Essas avaliações ajudam a identificar áreas de melhoria e a garantir que a organização esteja em conformidade com as regulamentações e normas aplicáveis em cada país.

Para realizar auditorias de segurança eficazes e garantir a conformidade em um ambiente DevSecOps, é importante integrar essas atividades em todas as fases do ciclo de vida do desenvolvimento de *software*. As empresas interessadas na implementação do DevSecOps devem estabelecer processos para monitorar continuamente a conformidade com as regulamentações e normas aplicáveis e identificar áreas de melhoria. Além disso, é importante garantir que todos os membros da equipe estejam cientes das políticas de segurança da informação e compreendam suas responsabilidades no que diz respeito à conformidade.

Capítulo 7 - Treinamento e Conscientização

É fundamental que os desenvolvedores tenham um entendimento sólido das melhores práticas de segurança e estejam atualizados sobre as últimas ameaças e técnicas de ataque. Algumas estratégias para implementar a educação em segurança para o time de desenvolvimento incluem:

a) **Treinamento e workshops:** Tópicos relevantes de segurança devem ser abrangidos tais como OWASP Top 10 0, princípios de codificação segura, gerenciamento de identidade e acesso, e segurança em APIs;
b) **Recursos de aprendizagem:** Fornecer recursos de aprendizagem tais como documentação, guias de referência, tutoriais e vídeos, que os desenvolvedores possam acessar para aprimorar seu conhecimento em segurança;
c) **Programas de mentoria**: Nelas, desenvolvedores mais experientes podem orientar e compartilhar seus conhecimentos de segurança com os membros mais novos da equipe para promover uma cultura de aprendizado contínuo e colaboração;
d) **Participação em comunidades de segurança**: Participação em comunidades de segurança, fóruns e conferências relacionadas à segurança de *software*;
e) **Desenvolvimento seguro como parte da cultura organizacional:** Promoção de uma cultura organizacional que valorize o desenvolvimento seguro, com recompensas e ambiente propício para o aprendizado contínuo;
f) **Estratégia de *Security Champion***: Estratégia na qual um ou mais desenvolvedores são designados como defensores da segurança dentro da equipe de desenvolvimento. Eles são responsáveis por promover práticas de segurança e facilitar a comunicação e a colaboração entre a equipe de desenvolvimento e a equipe de segurança;
g) **Eventos de CTFs (*Capture-The-Flag*)**: Promoção e incentivo à participação dos desenvolvedores em eventos de CTFs, que são competições de segurança cibernética baseadas em desafios, com cenários reais de ataques.

O investimento em educação e treinamento para o time de desenvolvimento, fortalece a capacidade dos times em construir aplicativos seguros e resilientes. A conscientização e o conhecimento em segurança são elementos essenciais para alcançar a verdadeira integração de DevSecOps.

Capítulo 8 - Conclusões

Neste Capítulo, serão apresentadas as necessidades identificadas para a melhoria contínua e superação de obstáculos na adoção do DevSecOps. Por fim, são feitas as considerações finais.

8.1. Necessidades Identificadas: Melhoria Contínua e Superação de Obstáculos

O monitoramento de métricas e indicadores de desempenho avaliam o desempenho do processo de DevSecOps, identificando possíveis gargalos, atrasos ou ineficiências e, assim, otimizando o fluxo de trabalho e melhorando a produtividade da equipe. Monitorar as taxas de vulnerabilidades não corrigidas, seus tempos de Acordo de Nível de Serviço (SLA) e tempo de resposta a incidentes, possibilitará a identificação precoce de problemas de segurança, permitindo que a equipe de segurança atue prontamente para mitigar os riscos, ou melhore seus processos.

Outro aspecto importante é a medição da eficácia das práticas de desenvolvimento seguro implementadas na organização, incluindo o uso de ferramentas de análise de código, testes de segurança automatizados e revisões de código. Ajustes e melhorias no processo de desenvolvimento de *software* com base nas descobertas do monitoramento, devem ser constantes, como um ciclo, levando em consideração que o processo sempre poderá ser melhorado e aprimorado.

Um dos principais obstáculos a serem superados ao implementar o DevSecOps é a resistência à mudança por parte das equipes de desenvolvimento, segurança e operações. A integração das práticas de segurança em todas as etapas do ciclo de vida do desenvolvimento de *software* exige mudanças significativas na cultura organizacional e nos processos de trabalho no time de desenvolvimento. Para superar resistências, é importante promover a comunicação e a colaboração entre as equipes, enfatizando os benefícios do DevSecOps como a redução de riscos, a melhoria da qualidade do *software* e a agilidade no processo de desenvolvimento.

Outro obstáculo que pode ser observado é a falta de conhecimento e habilidades necessárias para implementar práticas do DevSecOps que podem incluir a falta de familiaridade com ferramentas e tecnologias específicas, bem como a falta de compreensão das melhores práticas de segurança. Para enfrentar esse desafio, é essencial investir em treinamento das equipes de desenvolvimento, segurança e operações, garantindo que todos os membros estejam dotados de conhecimentos e habilidades para implementar práticas de DevSecOps.

A complexidade das infraestruturas de TI e a rápida evolução das ameaças de segurança também podem representar desafios significativos para a implementação de DevSecOps. Para lidar

com esses desafios, é importante adotar uma abordagem proativa e adaptável à segurança, utilizando ferramentas e práticas que permitam identificar e responder rapidamente a vulnerabilidades e ameaças emergentes. Apesar de todos os desafios, é importante celebrar o sucesso e os avanços, reconhecendo e celebrando os marcos alcançados na jornada de DevSecOps. Cada líder deve comemorar o sucesso das suas equipes em identificar e corrigir vulnerabilidades, bem como a implementação de práticas de desenvolvimento seguro. Isso motivará a equipe e reforçará a importância da segurança em toda a organização. Para conhecer mais desafios de implementação e como superá-los, conforme conteúdo publicado pela Universidade de Carnegie Mellon 0.

8.2. Considerações Finais

No presente Guia orientativo, foram apresentados os conceitos de DevSecOps e as principais ferramentas para a disseminação de boas práticas de desenvolvimento e operação seguras de *software* no contexto de uma cultura de segurança cibernética, com base numa proposta oriunda dos profissionais de empresas no setor de telecomunicações. A aplicabilidade dessas informações comporta tanto as prestadoras de telecomunicações quanto empresas em geral. Para o desenvolvimento deste documento, foi utilizada a estrutura de trabalho do Grupo de Estudos do Exercício Guardião Cibernético 5.0 - EGC 5.0, realizado em Brasília/DF em outubro de 2023.

Conforme apresentado, a inserção da segurança nos processos e rotinas de DevOps implicará em maior envolvimento das pessoas que compõem as equipes de desenvolvimento de *software* e Operações de TI e, por consequência, gerará colaboração direta entre as equipes. O DevSecOps foi apresentado como um vetor no estabelecimento da segurança como uma cultura nas empresas, sendo relevante no desenvolvimento de sistemas para aplicações críticas. Certamente, este material 100% em língua portuguesa e lançado ao término do EGC 5.0 poderá proporcionar a toda comunidade de profissionais de *software* no país mais aprendizado e, por consequência, mais desenvolvimento do ecossistema de segurança cibernética das infraestruturas críticas no Brasil.

A implementação de práticas de DevSecOps pode apresentar desafios significativos, mas com a abordagem correta e com o compromisso com a melhoria contínua, os autores deste Guia concluem que será possível superar as barreiras e garantir que privacidade e segurança sejam prioridades em todo o ciclo de desenvolvimento de *software*, elevando o grau de maturidade das empresas e proporcionando melhorias no desenvolvimento dos seus produtos e aplicativos digitais.

Referências

[1] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Estrutura de privacidade do NIST.* Versão 1.0. Disponível em: https://www.nist.gov/privacy-framework/privacy-framework. Acesso em: 15 jul. 2023.

[2] OPEN WEB APPLICATION SECURITY PROJECT (OWASP). *OWASP Project*. Disponível em: https://owasp.org. Acesso em: 12 set. 2023.

[3] BRASIL. Ministério da Gestão e da Inovação em Serviços Públicos. Secretaria de Governo Digital. *Guia do Framework de Privacidade e Segurança da Informação*. Versão 1.0. Disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi-atual. Acesso em: 26 set. 2023.

[4] Disponível em: https://www.pmi.org/disciplined-agile/process/disciplined-devops/defining-devops. Acesso em: 11 set. 2023.

[5] SÁNCHEZ-GORDÓN, Mary; COLOMO-PALACIOS, Ricardo. Security as culture: a systematic literature review of DevSecOps. *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. 2020. p. 266-269.

[6] NAMAL RAJAPAKSE, Roshan; ZAHEDI, Mansooreh; BABAR, Muhammad Ali. *Collaborative Application Security Testing for DevSecOps: An Empirical Analysis of Challenges, Best Practices and Tool Support*. arXiv e-prints, p. arXiv: 2211.06953, 2022.

[7] Disponível em: https://marvel-b1-cdn.bc0a.com/f00000000236551/dt-cdn.net/images/devsecops-image-2000-6557ba1b00.png. Acesso em: 9 set. 2023.

[8] Disponível em: https://owasp.org/www-project-devsecops-maturity-model/. Acesso em: 9 set. 2023.

[9] Disponível em: https://lms.ev.org.br/mpls/Custom/Cds/COURSES/2495-FUND_COBIT/pag/1_3_2.html. Acesso em: 9 set. 2023.

[10] Disponível em: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf. Acesso em: 11 set. 2023.

[11] Disponível em: https://www.nist.gov/publications/measuring-and-improving-effectiveness-defense-depth-postures . Acesso em: 9 set. 2023.

[12] Disponível em: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf. Acesso em: 9 set. 2023.

[13] Disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi/guia_requisitos_minimos_apis.pdf. Acesso em: 9 set. 2023.

[14] Disponível em: https://owasp.org/www-community/Threat_Modeling . Acesso em: 9 set. 2023.

[15] Disponível em: https://owasp.org/www-community/Threat_Modeling_Process . Acesso em: 9 set. 2023.

[16] Disponível em: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide. Acesso em: 9 set. 2023.

[17] Disponível em: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/. Acesso em: 9 set. 2023.

[18] CENTER INTERNET SECURITY. *CIS Benchmarks*. Disponível em: https://www.cisecurity.org/cis-benchmarks. Acesso em: 15 jul. 2023.

[19] Disponível em: guia_gerenciamento_vulnerabilidades.pdf (www.gov.br). Acesso em: 9 set. 2023.

[20] Disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi/guia_resposta_incidentes.pdf . Acesso em: 9 set. 2023.

[21] Disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi/modelo_politica_controle_acesso.pdf . Acesso em: 9 set. 2023.

[22] Disponível em: https://owasp.org/www-project-top-ten/. Acesso em: 9 set. 2023.

[23] Disponível em: https://insights.sei.cmu.edu/blog/5-challenges-to-implementing-devsecops-and-how-to-overcome-them/. Acesso em: 11 set. 2023.