evernow

Desenvolvimento Seguro de Ponta a Ponta

As melhores práticas para o desenvolvimento seguro.

Criador Responsável: Ney Piacentini

Confira o Conteúdo:

evernow

As melhores práticas para o desenvolvimento seguro

Desenvolvimento Seguro de Ponta a Ponta

Bem-vindo ao Whitepaper sobre Desenvolvimento Seguro de Ponta a Ponta!

Explore conosco as melhores práticas do DevSecOps e do ciclo de vida seguro do desenvolvimento de software. Descubra como antecipamos a segurança, integramos testes abrangentes e garantimos a proteção contínua dos aplicativos, tudo isso sem comprometer a agilidade no desenvolvimento.

Vamos juntos navegar pelas estratégias essenciais para construir um futuro digital mais inovador e seguro.
Boa leitura!

Introdução

Na era da transformação digital, empresas de todos os setores estão incorporando práticas de desenvolvimento de software para criar aplicativos inovadores e alcançar resultados excepcionais. A necessidade crescente de adicionar novos recursos e melhorias funcionais é crucial para atender às demandas em constante evolução e expandir a base de clientes.

A agilidade no lançamento de aplicativos tornou-se uma estratégia essencial, pois as organizações precisam disponibilizá-los rapidamente para manter uma vantagem competitiva. À medida que a importância dos aplicativos continua a aumentar, a necessidade de uma segurança sólida e abrangente também se intensifica.

No entanto, esses mesmos aplicativos tornaram-se alvos prioritários para ataques cibernéticos, mostrando a importância de que estes não apresentem vulnerabilidades de segurança e estejam em conformidade com as regulamentações. Segundo um relatório da Adjast, o Brasil foi o segundo país que mais cresceu no mercado de aplicativos no mundo nos últimos anos.

Este artigo explora a integração da segurança no desenvolvimento de aplicativos, com foco na abordagem DevSecOps e na adoção de testes de segurança ao longo de todo o ciclo de vida, sem impactar as entregas ou o desempenho dos aplicativos.

Principais Considerações

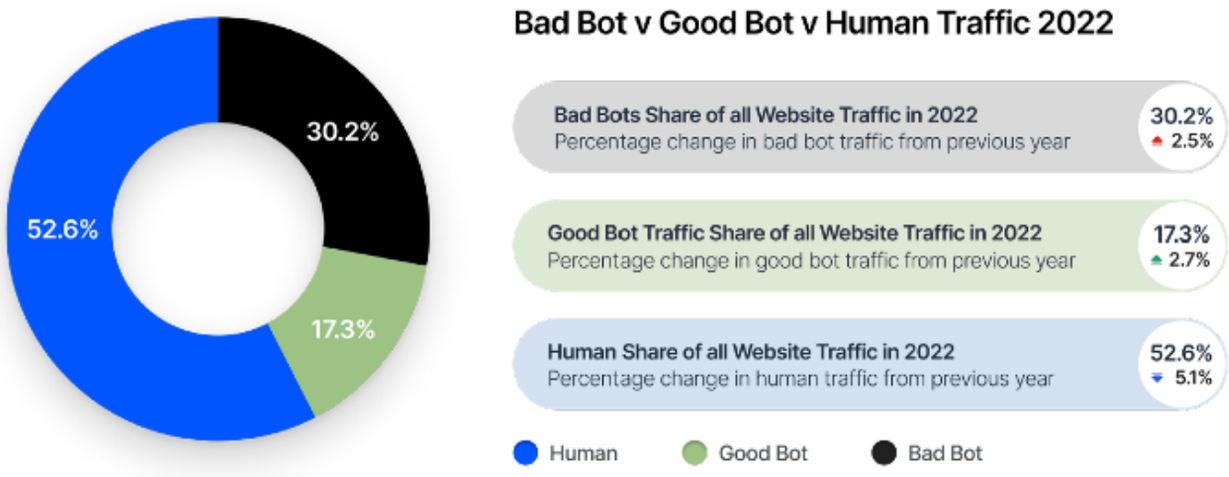
- As soluções de perímetro, como firewalls, IDS/IPS, VPNs e antivírus não são suficientes para proteger aplicativos na internet.
- 47,4% de todo tráfego de internet é automatizado. Destes, 30,2% são bots maliciosos que exploram a lógica de negócios e outras vulnerabilidades.
- A análise de ameaças de aplicações por Desenvolvedores e PO's nas fases iniciais traz eficiência e proatividade do ponto de vista técnico e de negócio.
- A utilização de testes estáticos e dinâmicos proporciona a identificação precoce de falhas e redução de custos na correção.
- Aplicativos codificados com segurança ainda podem ser vulneráveis, pois não estão protegidos contra engenharia reversa e adulteração.
- Proteja os dados de suas aplicações com criptografia, tokenização e mascaramento mantendo o formato (FPE) para evitar prejuízos decorrentes de vazamento de dados.
- Os testes de aplicativos devem ser integrados com os portões de segurança do DevSecOps para publicação de aplicativos comprovadamente seguros.
- Integre os resultados de toda sua stack em um Dashboard unificado para manter as equipes na mesma página.
- Práticas seguras de desenvolvimento devem ser adotadas de forma simples e fácil pare não interferir no trabalho dos desenvolvedores.

Segurança de Perímetro Não é o Suficiente.

Os aplicativos são como portas que se abrem para o mundo, permitindo a sua utilização em qualquer local. Não só por humanos, mas também por agentes automatizados.

De acordo com Imperva Bad Bot 2023, de todo o tráfego na internet em 2022, 47,4% foi tráfego automatizado, comumente referido como bots. Destes, 30,2% eram bots maliciosos e 17,3% benéficos.

De todos os ataques registrados pela Imperva no último ano, 27% foram causados por bots maliciosos que exploram vulnerabilidades em lógica de negócios, enquanto 26% foram outros tipos de ameaças automatizadas.



Vulnerabilidades de segurança são como buracos que permitem acessos indevidos a aplicativos e sistemas e resultam em indisponibilidade de serviços e roubo de dados.

Essas vulnerabilidades podem ser causadas por falhas no software ou por ações de hackers. É importante identificar e corrigir essas vulnerabilidades para manter os sistemas seguros e protegidos contra ameaças cibernéticas.

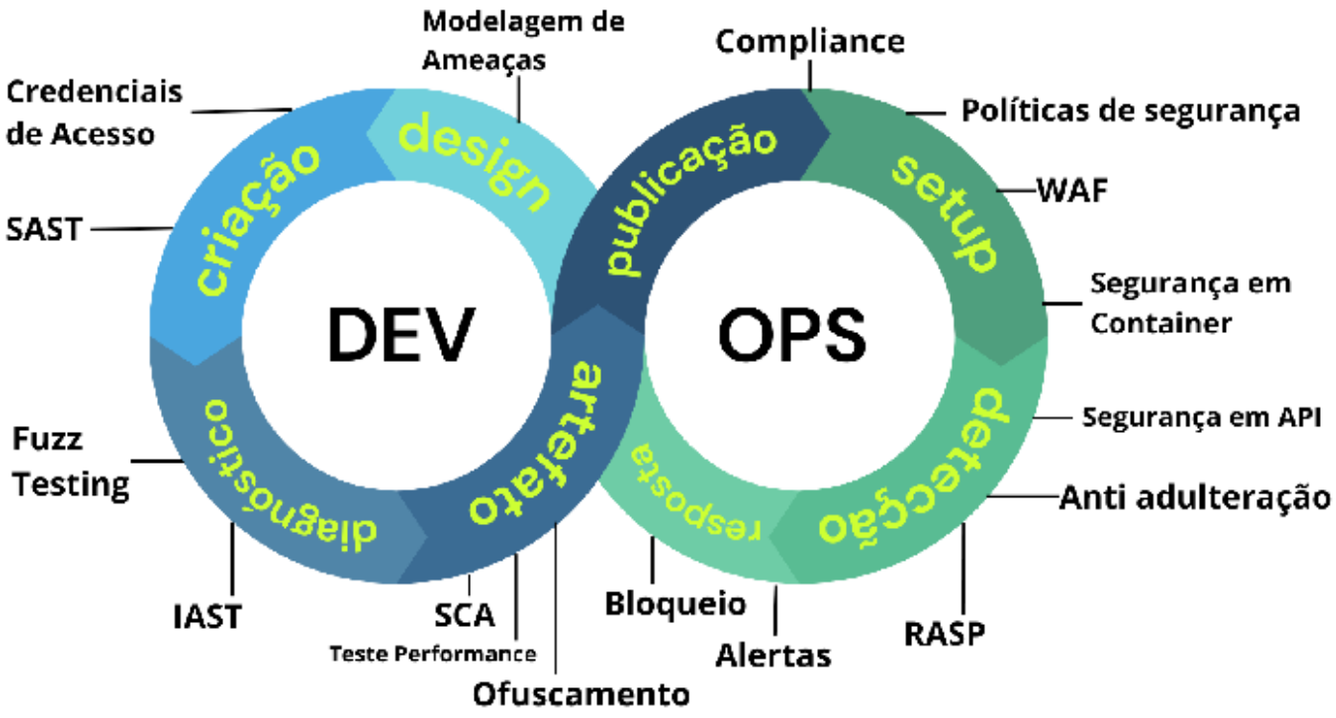
Para evitar essas ameaças, as organizações necessitam integrar a segurança durante o processo de desenvolvimento sem prejudicar as funcionalidades e o desempenho do aplicativo ou criar complicações para os desenvolvedores.

DevSecOps e SSDLC

As empresas estão incorporando tecnologias para integrar a segurança de maneira contínua e automatizada nos processos de desenvolvimento. O modelo DevSecOps, é uma evolução do DevOps e visa unificar a segurança buscando a entrega eficiente e segura de software de alta qualidade.

Essa abordagem fundamenta-se nos aspectos culturais do DevOps, integrando o trabalho das equipes de segurança ao Ciclo de Vida do Desenvolvimento de Software (SDLC), adicionando a dimensão "Segura" na sigla (SSLDC).

A Figura abaixo ilustra que a segurança não deve ser implementada apenas no início ou no final, mas em cada ponto do pipeline de entrega. Como parte do DevSecOps, portões garantem que os requisitos de segurança sejam atendidos em cada estágio do SDLC, com a possibilidade de barrar novas vulnerabilidades de alta criticidade.



O DevSecOps está intimamente relacionado ao conceito de "Shift Left" - a prática de integrar a segurança no processo de desenvolvimento e tornar correções prévias de uma parte das responsabilidades do desenvolvedor.

Em um cenário ideal, os desenvolvedores compreendem profundamente e praticam a codificação segura, mas essa não é uma realidade. Muitos desenvolvedores não possuem conhecimento ou experiência suficiente nestas práticas, o que pode resultar na introdução acidental de falhas de segurança no código do aplicativo. A pesquisa de 2023 sobre Adoção, Técnicas e Ferramentas da IDC constatou que a falta de conhecimento de segurança do desenvolvedor é o principal desafio organizacional relacionado à adoção do DevSecOps.

DevSecOps e SSDLC

À medida que as organizações buscam aprimorar o conhecimento dos desenvolvedores em codificação segura, os testes em aplicativos ajudam a preencher esta lacuna.

No contexto do Ciclo de vida do Desenvolvimento Seguro (SSLDC), diversas técnicas e boas práticas podem ser incluídas rapidamente em qualquer pipeline, visando minimizar os riscos. São elas:

1. Análise de Ameaças
2. Análise em Código Fonte e Componentes (SAST/SCA)
3. Análise Dinâmica em Aplicações (DAST)
4. Ofuscação e Anti adulteração (Hardening)
5. Proteção de Dados em Trânsito e Repouso
6. Autoproteção de aplicativos em tempo de execução (RASP)

Na jornada de cibersegurança, cada passo importa!

Vamos avançar juntos, construindo gradualmente a proteção que sua empresa precisa!

evernow

1. Análise de Ameaças

A análise de ameaças é uma prática de segurança que visa avaliar os protocolos, processos e procedimentos utilizados com o objetivo de identificar possíveis ameaças e vulnerabilidades nas fases iniciais de desenvolvimento.

As vulnerabilidades são pontos críticos para possíveis ataques cibernéticos, exigindo sua identificação para avaliar os riscos. Métricas tradicionais, como CVSS e EPSS, possuem limitações, sendo necessário complementá-las com dados sobre ativos vulneráveis, informações sobre ameaças atuais e tendências.

Pense como exemplo, uma empresa que lida com dados de saúde desenvolve uma função para que os pacientes acessem os resultados dos exames.

Na análise de riscos, identificam-se ativos críticos, como dados pessoais e sistemas de informação de saúde. As possíveis ameaças destas pesquisas, incluem acesso não autorizado, interceptação de comunicação e falhas de autenticação. O impacto ao negócio pode envolver o comprometimento da privacidade, perda de confiança dos pacientes e possíveis consequências legais. Mitigações sugeridas são controles de acesso, criptografia, monitoramento de segurança e testes regulares, além de investimento em educação e um plano de resposta a incidentes para fortalecer a segurança.

A plataforma de categorização de risco para os desenvolvedores é uma solução para facilitar a identificação e priorização de ameaças. Antes dos testes de segurança, as ameaças são categorizadas com base em probabilidade e impacto, sensibilidade dos dados trafegados e exposição, permitindo que as equipes se concentrem nas partes mais críticas. Isso proporciona uma visão clara das ameaças, permitindo ações proativas para mitigá-las e redução real no risco de violações.

A plataforma direciona testes automatizados para partes principais do sistema, maximizando sua eficácia.

Escolha da Evernow: SD Elements

O SD Elements da Security Compass oferece uma solução completa para o gerenciamento de ameaças, integrando práticas de segurança diretamente no ciclo de vida do desenvolvimento de software.
A plataforma automatiza e simplifica a identificação e mitigação de vulnerabilidades. Isso resulta em maior eficiência, redução de erros e custos associados, proporcionando uma abordagem mais proativa e eficaz para todo o processo de desenvolvimento de software.

2. Análise em Código Fonte e Componentes (SAST/SCA)

As análises em código fonte e componentes são técnicas de teste de segurança que ajudam a identificar vulnerabilidades na fase inicial do desenvolvimento de software. Existem várias ferramentas disponíveis, como SAST e SCA, que ajudam a identificar vulnerabilidades em aplicativos.

De acordo com a Veracode, empresas que implementam análise estática de segurança no desenvolvimento têm 48% menos vulnerabilidades em seus aplicativos.

O SAST (Static Application Security Testing) analisa o código-fonte de um aplicativo para identificar vulnerabilidades. Concentrando-se no código escrito pelos desenvolvedores, o SAST busca identificar potenciais vulnerabilidades que podem ser exploradas.

O SCA (Software Composition Analysis) é um teste de segurança com foco nos componentes de terceiros utilizados em um aplicativo. Tem como objetivo encontrar possíveis vulnerabilidades nestas bibliotecas. Um exemplo notório foi a vulnerabilidade do Log4j, identificada pelo SCA, que permitia acesso remoto via console ao servidor de aplicação. Além disso, o SCA auxilia na prevenção do uso não autorizado de dependências no código-fonte, mitigando riscos e evitando potenciais multas de licenciamento.

Os testes estáticos identificam falhas de segurança nos estágios iniciais de desenvolvimento e são adicionados ao processo de integração contínua com rapidez e facilidade. Além de prevenir problemas de segurança, economizam recursos, promovem padronização e facilitam a conformidade com diretrizes de desenvolvimento.

Escolha da Evernow: Fortify Source Code Analyzer e DeBricked

Fortify é líder em Análise Estática de Segurança de Software (SAST), identificando vulnerabilidades no código-fonte para fortalecer a segurança durante o desenvolvimento.

Debricked destaca-se em Análise de Componentes de Software (SCA), gerenciando riscos de bibliotecas de terceiros.

Além de serem soluções compatíveis e integradas, ambas são essenciais para uma visão abrangente de segurança assegurando prevenção eficaz de ameaças e proteção do software em desenvolvimento.

3. Análise Dinâmica em Aplicações (DAST)

O DAST (Dynamic Application Security Testing) se concentra no comportamento do aplicativo em vez de seu código-fonte. O objetivo do DAST é identificar vulnerabilidades que possam ser exploradas de fora para dentro da aplicação em um ambiente de produção.

Abaixo as diferenças entre SAST e DAST no SSLDC:

SAST

- Abordagem voltada para desenvolvedores - testadores têm acesso ao framework subjacente, design e implementação.
- Requer código-fonte ou binário, não exige execução do programa.
- Realizado no início do Ciclo de Vida de Desenvolvimento de Software (SDLC).
- Avalia a aplicação.
- Suporta testes em ambientes de design sequencial, sistemas em tempo real, aplicativos móveis e software em dispositivos embarcados.

DAST

- Abordagem voltada para hackers-testadores não têm conhecimento interno.
- Exige a execução do programa, não é necessário acesso ao código ou binário.
- Realizado no final do SDLC.
- Avalia o ambiente e problemas em tempo de execução.
- Suporta testes em aplicativos web, serviços, servidores, bancos de dados e caches.

Escolha da Evernow: Fortify WebInspect

Fortify WebInspect oferece segurança integrada no mesmo pacote e é utilizado para realizar testes dinâmicos de segurança de aplicativos durante o processo de garantia de qualidade, reduzindo vulnerabilidades em novas e existentes aplicações.

A automação aumentada atende à crescente demanda dos clientes por segurança em aplicativos web e móveis, fornecendo relatórios detalhados de varreduras.

4. Ofuscação e Anti adulteração (Hardening)

Mesmo que um desenvolvedor entregue um aplicativo codificado de forma segura, isso não significa que não existam vulnerabilidades exploráveis, isso porque a codificação segura não garante proteção contra engenharia reversa e adulteração.

Com acesso aos binários executáveis, qualquer um com conhecimento apropriado e ferramentas de engenharia reversa pode inspecionar o código fonte e promover alterações!

Além disso, as organizações não podem controlar todos os ambientes nos quais seus aplicativos serão implantados e usados. Por exemplo, os usuários podem usar aplicativos em dispositivos que foram desbloqueados ou que tiveram acesso root.

Atualmente, o tipo mais comum de endurecimento do aplicativo é a ofuscação de código, que transforma o código simples gerado pelo compilador em código que compartilha o mínimo possível das características do código, perservando a integridade e o comportamento do aplicativo.

A ofuscação de código remove o contexto que humanos e descompiladores usam para entender a funcionalidade do aplicativo. As transformações podem incluir coisas como renomear símbolos, criptografia de strings, alteração de fluxo de controle e substituição de instruções.

É possível automatizar esse processo, levando a aplicativos mais seguros sem que o desenvolvedor precise de conhecimentos específicos de segurança.

No entanto, nem todas as técnicas de ofuscação são iguais. Algumas são mais fracas do que outras e mais suscetíveis à desofuscação e engenharia reversa.

Portanto, é importante empregar soluções que estejam atualizando e desenvolvendo regularmente novas técnicas de ofuscação.

Escolhas da Evernow: JScrambler

A utilização do JScrambler oferece uma abordagem robusta para o fortalecimento de aplicações, proporcionando proteção avançada contra ameaças cibernéticas.

O JScrambler oferece ofuscação e transformação de código JavaScript, além de proteção de código em aplicações Android e Java, resultando em uma segurança extensiva e multicamada.

5. Proteção de Dados em Trânsito e em Repouso

A proteção e criptografia de dados em trânsito e em repouso são práticas essenciais para garantir a segurança dos dados em um processo de desenvolvimento seguro. A criptografia de dados ajuda a impedir que usuários não autorizados leiam dados em um cluster e em sistemas de armazenamento físico de dados associados. Isso inclui dados salvos em mídias persistentes, conhecidos como dados em repouso, e dados que podem ser interceptados enquanto viajam pela rede, conhecidos como dados em trânsito.

A criptografia de dados é uma das principais medidas de segurança que podem ser tomadas para proteger informações confidenciais e sensíveis. Além disso, a proteção de dados é um requisito para a conformidade com muitas regulamentações de privacidade de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil e GDPR na Europa, entre outras. É importante que as organizações incorporem essas práticas em seus processos de desenvolvimento seguro para garantir a segurança dos dados e a conformidade com as regulamentações aplicáveis.

Escolha da Evernow: Voltage SecureData

O uso do Voltage SecureData oferece vantagens significativas na proteção de dados em trânsito e em repouso.

Com recursos avançados de criptografia, tokenização e mascaramento utilizando o FPE, a solução garante a segurança abrangente dos dados, reduzindo os riscos de violações de dados e proporcionando conformidade com regulamentações.

Sua abordagem inovadora oferece uma camada adicional de proteção, fortalecendo a segurança de dados em ambientes dinâmicos e estáticos.

Criptografia Preservando o Formato (Format Preserve Encryption / FPE)

Abordagens tradicionais de criptografia, como AES 256 em modos CBC, GCM, CTR, etc... têm um impacto significativo nas estruturas de dados, esquemas e aplicações. O FPE é um método de tokenização que utiliza o modo FF1, padrão do NIST, do algoritmo Advanced Encryption Standard (AES), que criptografa dados sensíveis preservando seu formato original sem comprometer a força da criptografia.

Dados estruturados, como CPF, cartão de crédito, conta, data de nascimento, campos de salário ou endereços de e-mail, podem ser protegidos, mantendo seu número de caracteres e formato, sem exigir alterações em bancos de dados, impactando funcionalidades e o desempenho da aplicação. Tipicamente, sistemas inteiros podem ser protegidos rapidamente em apenas alguns dias a um custo significativamente reduzido.

evernow

6. RASP e a importância da camada de proteção automatizada

O Runtime Application Self-Protection (RASP) é uma tecnologia que incorpora funcionalidades de segurança dentro de aplicativos de software para prevenir ataques maliciosos enquanto o aplicativo está em execução.

Enquanto as medidas de segurança tradicionais de rede e infraestrutura, como um firewall de aplicativos da web (WAF) ou um sistema de prevenção de intrusões (IPS), são usadas para monitorar o tráfego de rede e as sessões do usuário para identificar atividades suspeitas, essas ferramentas não monitoram o tráfego e os dados dentro do aplicativo, deixando a organização vulnerável a ataques de aplicativos.

Diferentemente das soluções de segurança tradicionais, que oferecem proteção no nível da rede ou do endpoint, o RASP move a segurança para dentro do aplicativo, permitindo que a organização colete dados em tempo real e os avalie no contexto desse aplicativo, utilizando informações contextuais para monitorar anomalias e interromper ameaças automaticamente em tempo real.

Como a ferramenta é específica para cada aplicativo e seu uso real, o RASP oferece um nível de precisão e proatividade incomparável por ferramentas e soluções legadas, além de, por intermédio de aprendizado de máquina de última geração, é capaz de bloquear ataques e notificar os responsáveis para correção em sua origem.

Escolha da Evernow: Imperva RASP

A utilização do Imperva para Runtime Application Self-Protection (RASP) oferece vantagens significativas na segurança de aplicações em tempo de execução. Com detecção em tempo real e resposta automatizada a ameaças, o Imperva RASP proporciona uma defesa proativa contra ataques, identificando e bloqueando vulnerabilidades enquanto a aplicação está em execução. Sua capacidade de adaptação dinâmica às mudanças nas ameaças cibernéticas e a integração perfeita com o ambiente de desenvolvimento tornam o Imperva uma escolha robusta para a proteção eficaz de aplicações.

Resumo

Para fortalecer a segurança em um ambiente digital em constante evolução, profissionais de TI devem adotar uma abordagem abrangente e integrada.

Isso inclui a identificação proativa de ameaças, análises minuciosas no código fonte e nos componentes (SAST/SCA), avaliação dinâmica em tempo de execução (DAST), implementação de medidas de ofuscação e anti adulteração (Hardening), garantia da segurança de dados em trânsito e repouso, além da autoproteção de aplicativos durante a execução (RASP).

A natureza ágil e dinâmica das aplicações modernas exige a integração da segurança em todo o ciclo de vida do processo de desenvolvimento (DevSecOps).

Negligenciar essas práticas expõem as organizações a riscos significativos, potencialmente comprometendo aplicativos e dados críticos. Portanto, é essencial adotar ferramentase práticas que automatizam e simplificam a inserção dasegurançanociclodevidadoDevSecOps.

Além disso, a utilização de uma suíte de mecanismos de proteção em camadas, indo além do simples código seguro, é fundamental para garantir uma segurança robusta contra ameaças emergentes.

- A adoção de tecnologias e práticas DevSecOps é crucial para mitigar riscos desnecessários e proteger aplicativos e dados.
- Os processos de desenvolvimento seguro devem ser práticos e não intrusivos, facilitando a aceitação pelos desenvolvedores.
- A utilização de uma suíte abrangente de mecanismos de proteção em camadas, incluindo ofuscação forte, verificações de integridade, monitoramento em tempo real e criptografia, é essencial para uma segurança robusta, indo além da dependência apenas do código seguro.

Conclusão

A complexidade e constante evolução da segurança cibernética representam desafios para os desenvolvedores, que muitas vezes têm dificuldade em compreender as nuances das diversas medidas e práticas em segurança. A Evernow, reconhecendo essa dinâmica, auxilia sua equipe a garantir a proteção do aplicativo, dados e infraestrutura, mesmo para desenvolvedores menos familiarizados com as melhores práticas de segurança.

Ao escolher a Evernow, sua organização se beneficia de uma sólida experiência no mercado. A nossa parceria não só auxilia na identificação e correção de vulnerabilidades, mas também oferece soluções personalizadas de ponta a ponta, adaptadas às necessidades específicas da sua organização.

Contando com profissionais atualizados nas últimas tendências e tecnologias e atuando nos maiores clientes de diversos segmentos, a Evernow se posiciona como um parceiro capaz de fornecer soluções mais eficientes do que uma equipe interna de segurança de aplicativos.

evernow

Sobre a Evernow

A Evernow é uma empresa especializada em cibersegurança, abrangendo desenvolvimento de software, proteção de dados, infraestrutura e serviços em nuvem.

Nossas operações têm foco no Brasil, mas nossos serviços se estendem globalmente, compartilhando nosso conhecimento com clientes líderes em diversos setores de atuação.

Parceira para regulamentação de segurança LGPD, GDPR, HIPAA, PCI, SOC, NIST e outros. Contamos também com parcerias em serviços de nuvem como a Microsoft, AWS e Google e prática de desenvolvimento seguro em programação, IoT, inteligência artificial e aprendizado de máquina.

Contatos:

(11) 3042-1384

contato@evernow.com.br

R. Sansão Alves dos Santos, 433 - Cidade Monções, São Paulo - SP, 04571-090