



SEGURANÇA DE SOFTWARE

Moises Brandalise - Aula 02

Professores

MOISES BRANDALISE

Professor Convidado

Atua como Especialista em Segurança da Informação em uma instituição financeira. Na carreira, atuou no ramo da indústria por 10 anos no papel de líder técnico em infraestrutura de tecnologia e 3 anos como Analista de desenvolvimento de Sistemas em fábrica de software. Em segurança da informação, atuou na indústria da mídia por 6 anos como Analista e no segmento financeiro, por 4 anos como Especialista, além de 2 anos como Especialista em Proteção de dados pessoais, totalizando cerca de 25 anos de mercado.

AVELINO ZORZO

Professor PUCRS

Associado da Sociedade Brasileira de Computação (SBC) e da IEEE. Possui graduação em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1986-1989), mestrado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1990-1994), doutorado em Ciência da Computação pela University of Newcastle Upon Tyne (1995-1999) e pós-doutorado na área de segurança no Cybercrime and Computer Security Centre da Newcastle University (2012-2013). Atualmente é professor titular da Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Coordenador de Programas Profissionais da área de Computação da CAPES/MEC, avaliador de condições de ensino do Ministério da Educação, consultor ad hoc do CNPq, CAPES e da FAPERGS.

Ementa da disciplina

Estudo sobre os métodos e utilização de criptografia para transmissão e armazenamento. Estudo sobre protocolo de comunicação em navegadores (HTTPS) ou aplicativos de conversa (LibSignal). Estudo sobre segurança no desenvolvimento de software. Estudo sobre os problemas mais frequentes indicados pela OWASP. Estudo sobre métodos de autenticação e autorização.

Segurança de Software

Por Moises Brandalise

EMENTA DA DISCIPLINA

MÉTODOS E UTILIZAÇÃO DE CRIPTOGRAFIA PARA TRANSMISSÃO E ARMAZENAMENTO.
PROTOCOLO DE COMUNICAÇÃO EM NAVEGADORES (HTTPS) OU APLICATIVOS DE CONVERSA (LIBSIGNAL).
SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE.
OS PROBLEMAS MAIS FREQUENTES INDICADOS PELA OWASP.
MÉTODOS DE AUTENTICAÇÃO E AUTORIZAÇÃO.

CONTEÚDO DAS AULAS

1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
2. MÉTODOS DE CRIPTOGRAFIA
3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP
6. AUTENTICAÇÃO E AUTORIZAÇÃO

ENCONTROS DA DISCIPLINA



1º PARTE

INTRODUÇÃO A
SEGURANÇA DE
SOFTWARE

1 HORA



2º PARTE

MÉTODOS DE
CRIPTOGRAFIA

1 HORA



3º PARTE

PROTOCOLOS DE
COMUNICAÇÃO
SEGURA

1 HORA

ENCONTROS DA DISCIPLINA



4º PARTE

SEGURANÇA NO
DESENVOLVIMENTO
DE SOFTWARE

1 HORA



5º PARTE

PROBLEMAS
COMUNS DE
SEGURNAÇA
INDICADOS PELA
OWASP

1 HORA



6º PARTE

AUTENTICAÇÃO E
AUTORIZAÇÃO

1 HORA

PROFESSOR(A) CONVIDADO(A)

MOISES BRANDALISE

Atuação:

- Segurança da Informação no ramo financeiro (6 anos);
- Analista de Segurança (5 anos);
- Analista de Sistemas e Programador (5 anos);
- Analista de Infraestrutura (5 anos)

Formação acadêmica:

- Especialista em Gestão Estratégica de TI (Pucrs);
- Especialista em Segurança e Gestão de Redes (Ufrgs);
- Ciência da Computação (Upf);

Certificações:

- CISSP, CISM, CDPSE, ISFS.

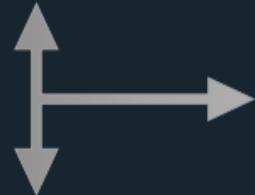
PROFESSOR(A) PUCRS

AVELINO F. ZORZO

- Doutor em Ciência da Computação pela University of Newcastle Upon Tyne;
- Pós-doutorado na área de segurança no Cybercrime and Computer Security Centre da Newcastle University.
- Professor titular da Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS);
- Coordenador da área de Computação, Membro do CTC-ES e do Conselho Superior na CAPES/MEC.

GLOSSÁRIO

TEORIA



EXEMPLO

S
PRÁTICO
S



- 1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
- 2. MÉTODOS DE CRIPTOGRAFIA
- 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
- 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
- 5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP
- 6. AUTENTICAÇÃO E AUTORIZAÇÃO

GLOSSÁRIO

- 
1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
 2. MÉTODOS DE CRIPTOGRAFIA
 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
 4. **SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE**
 - a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA
 - b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO
 - c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS
 5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP
 6. AUTENTICAÇÃO E AUTORIZAÇÃO

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA

É fundamental para garantir que as aplicações sejam robustas e confiáveis.....



- IDENTIFICAÇÃO DOS RISCOS DE SEGURANÇA E VULNERABILIDADES
- GERENCIAMENTO DE VULNERABILIDADES E RISCOS
- INTEGRAÇÃO DA SEGURANÇA NO PROCESSO
- TREINAMENTO E CONSCIENTIZAÇÃO DOS DESENVOLVEDORES



- Confiança
- Redução de Riscos
- Redução de Custos
- Qualidade do software

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE



- **Identificação dos riscos de segurança e vulnerabilidades**
- Gerenciamento de vulnerabilidades e riscos
- Integração da segurança no processo
- Treinamento e conscientização dos desenvolvedores

Realizar uma análise de risco...

Analisar as Ameaças:

Identificar as ameaças que podem afetar a aplicação....

Identificar Ativos:

Listar os ativos críticos da aplicação...

Avalie as Vulnerabilidades:

Verifique se existem vulnerabilidades em potencial em cada um dos ativos identificados...

Avalie o Impacto:

Avalie o impacto potencial de cada vulnerabilidade identificada...

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA



Durante a pandemia de COVID-19 houve um aumento significativo em ataques explorando vulnerabilidades de segurança em conexões de rede remotas.

Classifique os Riscos:

Classifique os riscos identificados em ordem de prioridade...

Plano de mitigação:

Desenvolva planos de ação para mitigar os riscos identificados....

Monitoramento:

Monitore continuamente a aplicação em busca de novas ameaças e vulnerabilidades...

Exemplo:

Considere uma aplicação bancária com recursos tradicionais de uma conta corrente.



- **Identificação dos riscos de segurança e vulnerabilidades**
 - Gerenciamento de vulnerabilidades e riscos
 - Integração da segurança no processo
 - Treinamento e conscientização dos desenvolvedores

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA

Exemplo: Considere uma aplicação bancária com recursos tradicionais de uma conta corrente.

Analise do Risco	Análise
1 - Ameaças	Comprometimento da credencial de usuários
2 - Ativos	Tabela de usuários
3 - Vulnerabilidades	Senhas fracas ou fáceis de adivinhar
4 - Impacto	Financeiro; Imagem
5 – Classificação do Riscos	Probabilidade: Média / Impacto: Alto
6 – Plano de mitigação	Incluir 2FA; Treinar usuários;
7 - Monitoramento	Análise regular dos logs do servidor

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA



- Identificação dos riscos de segurança e vulnerabilidades
- **Gerenciamento de vulnerabilidades e riscos**
- Integração da segurança no processo
- Treinamento e conscientização dos desenvolvedores

Envolve a **implementação** de medidas de segurança para **mitigar riscos**

Responsável

Definir uma equipe (ou pessoa) **responsável** pelo gerenciamento de **vulnerabilidades** e riscos....

Testes regulares:

Para **identificar** possíveis **vulnerabilidades** e riscos à segurança.

Acompanhar as mudanças

Viabilizar um controle para que as **mudanças** na aplicação sejam avaliadas...

Treinamento

Garantir que todos os **desenvolvedores** sejam **treinados** em **segurança** de aplicativos.



Em 2010 o Google descobriu que havia sido alvo de um ataque cibernético em grande escala, com origem na China. O ataque ficou conhecido como "Operação Aurora".





- Identificação dos riscos de segurança e vulnerabilidades
- **Gerenciamento de vulnerabilidades e riscos**
- Integração da segurança no processo
- Treinamento e conscientização dos desenvolvedores

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA

Exemplo: Considerando uma aplicação genérica.

Atividade	Análise 1
Planejamento	Proteger os dados confidenciais dos clientes.
Análise	Identificação dos dados sensíveis que precisam ser protegidos.
Design	Implementação de criptografia de dados sensíveis.
Desenvolvimento	Controle de permissões de usuários.
Testes	Testes de vulnerabilidade, simulação de ataques
Implantação	Plano de monitoramento e detecção
Manutenção	Atualizações regular de software (novas tecnologias)

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA



- Identificação dos riscos de segurança e vulnerabilidades
- Gerenciamento de vulnerabilidades e riscos
- **Integração da segurança no processo**
- Treinamento e conscientização dos desenvolvedores

Uma maneira de integrar a segurança no processo de desenvolvimento é utilizar a metodologia DevSecOps....

Quando:

Requisitos de segurança sejam considerados desde o **início do projeto...**

Desenvolvimento:

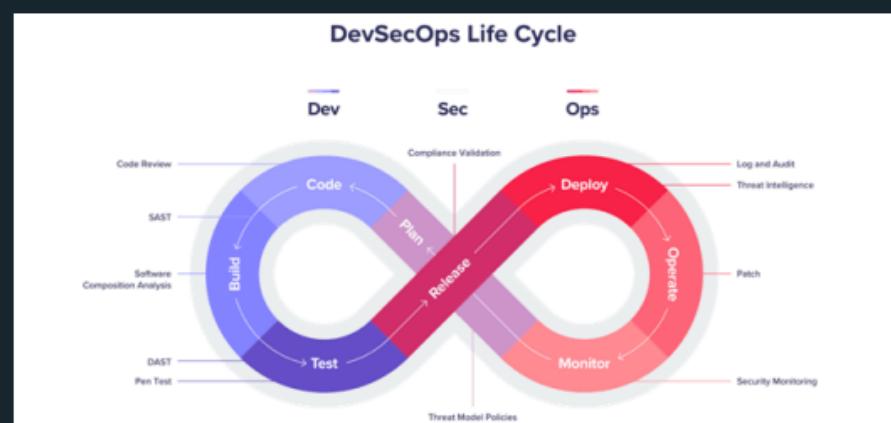
Utilizar **ferramentas** de análise de segurança de código...

Como:

Conjunto de diretrizes a serem seguidas....



A empresa de segurança Snyk revelou que 67% dos desenvolvedores estão preocupados com a segurança das aplicações, mas apenas 27% dizem ter tempo suficiente.



4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA



- Identificação dos riscos de segurança e vulnerabilidades
- Gerenciamento de vulnerabilidades e riscos
- Integração da segurança no processo
- **Treinamento e conscientização dos desenvolvedores**

Oferecer **treinamento** em segurança de software e **disponibilizar** recursos...

Treinamentos

Presenciais ou online sobre segurança de software....

Materiais

Vídeos, tutoriais e documentos sobre segurança de software...

Revisão conjunta de código:

Os desenvolvedores revisam o código uns dos outros...

Checklist

Usar como guia para implementar as melhores práticas de segurança.



Estudos têm mostrado que desenvolvedores com habilidades interpessoais fortes tendem a ter mais sucesso em suas carreiras.





- Identificação dos riscos de segurança e vulnerabilidades
- Gerenciamento de vulnerabilidades e riscos
- Integração da segurança no processo
- **Treinamento e conscientização dos desenvolvedores**

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA

Checklist de Segurança de Código:

- **Validação de Entrada**
 - ✓ Todas as entradas recebidas pela aplicação são validadas e sanitizadas corretamente?
 - ✓ A validação de entrada é feita no servidor e no cliente?
- **Gerenciamento de Sessão**
 - ✓ A sessão do usuário é gerenciada corretamente?
 - ✓ A sessão é encerrada quando o usuário faz logout ou após um período inativo?
 - ✓ A sessão usa cookies seguros e HttpOnly?
- **Proteção contra Injeção de Código**
 - ✓ A aplicação está protegida contra ataques de injeção de código, como SQL Injection e XSS?
 - ✓ As consultas de banco de dados estão usando Prepared Statements ou Stored Procedures?
 - ✓ As entradas de usuário são sanitizadas antes de serem exibidas na aplicação?
- **Gerenciamento de Senhas**
 - ✓ As senhas dos usuários são armazenadas de forma segura?
 - ✓ As senhas são armazenadas em hash e salteadas?
 - ✓ A senha é validada com requisitos mínimos de complexidade?



- Identificação dos riscos de segurança e vulnerabilidades
- Gerenciamento de vulnerabilidades e riscos
- Integração da segurança no processo
- **Treinamento e conscientização dos desenvolvedores**

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA

Checklist de Segurança de Código:

- **Configuração do Servidor**
 - ✓ O servidor está configurado corretamente para evitar vulnerabilidades conhecidas?
 - ✓ As atualizações do servidor são aplicadas regularmente?
- **Proteção de Dados**
 - ✓ Os dados da aplicação são protegidos corretamente?
 - ✓ Os dados são criptografados em trânsito e em repouso?
 - ✓ A aplicação segue as leis de proteção de dados relevantes?
- **Controle de Acesso**
 - ✓ A aplicação controla corretamente o acesso a recursos protegidos?
 - ✓ O acesso a recursos é limitado apenas aos usuários autorizados?
 - ✓ As permissões de usuário são verificadas corretamente?



- Identificação dos riscos de segurança e vulnerabilidades
- Gerenciamento de vulnerabilidades e riscos
- Integração da segurança no processo
- **Treinamento e conscientização dos desenvolvedores**

Oferecer treinamento em segurança de software e disponibilizar recursos...

Treinamentos

Presenciais ou online sobre segurança de software....

Materiais

Vídeos, tutoriais e documentos sobre segurança de software...

Revisão conjunta de código:

Os desenvolvedores revisam o código uns dos outros...

Checklist

Usar como guia para implementar as melhores práticas de segurança.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

a) OS PRINCIPAIS DESAFIOS DE SEGURANÇA



Estudos têm mostrado que desenvolvedores com habilidades interpessoais fortes tendem a ter mais sucesso em suas carreiras.

Recompensas

Identificação de vulnerabilidades....



b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO

Segurança do software desde a concepção até a sua retirada de uso.



- PLANEJAMENTO E DEFINIÇÃO DE REQUISITOS DE SEGURANÇA
- PROJETO E IMPLEMENTAÇÃO SEGUROS
- TESTES DE SEGURANÇA E AVALIAÇÃO DE VULNERABILIDADES
- IMPLANTAÇÃO E MANUTENÇÃO SEGURAS



- Proteção de informações confidenciais
- Melhor experiência do usuário



- **Planejamento e definição de requisitos de segurança**
- Projeto e implementação seguros
- Testes de segurança e avaliação de vulnerabilidades
- Implantação e manutenção seguras

Realizar uma análise de riscos...

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO



O modelo de ciclo de vida em cascata, que foi o primeiro modelo formal de desenvolvimento de software.





- Planejamento e definição de requisitos de segurança
- **Projeto e implementação seguros**
- Testes de segurança e avaliação de vulnerabilidades
- Implantação e manutenção seguras.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO

A **modelagem de ameaças** é uma prática importante que deve ser aplicada....



Analise do Risco	Análise
1 - Ameaças	Comprometimento da credencial de usuários
2 - Ativos	Tabela de Usuários
3 - Vulnerabilidades	Senhas fracas ou fáceis de adivinhar
4 - Impacto	Financeiro; Imagem
5 – Classificação do Riscos	Probabilidade: Media / Impacto: Alto
6 – Plano de mitigação	Incluir 2FA
7 - Monitoramento	Análise regular dos logs do servidor



- Planejamento e definição de requisitos de segurança
- **Projeto e implementação seguros**
- Testes de segurança e avaliação de vulnerabilidades
- Implantação e manutenção seguras.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO

- **Spoofing:** Falsificação de identidade
- **Tampering:** Violação ou Adulteração
- **Repudiation:** Negação de ter realizado uma ação
- **Information Disclosure:** Divulgação ou vazamento
- **Denial of Service:** Negação de Serviço
- **Elevation of Privilege:** Aumento indevido de privilégios

Vulnerabilidade: Senhas fracas ou fáceis de adivinhar	S	T	R	I	D	E
Tabela de Usuários	-	S	-	S	S	S
Banco de Dados	-	S	-	S	S	S
Servidor de Aplicação	S	S	S	S	S	S
....						

- **Injeção de Código Malicioso**
 - ✓ Validar extensão do arquivo
 - ✓ Biblioteca verificação de vírus
 - ✓ Executar com privilégios mínimos
- **Sobrecarga no servidor**
 - ✓ Limitar tamanho do arquivo.
 - ✓ Implementar taxa de upload.
 - ✓ Balanceamento de carga.
- **Dados Sensíveis**
 - ✓ Criptografar os dados ao enviar para o servidor.
 - ✓ Definir permissões adequadas no servidor.



- Planejamento e definição de requisitos de segurança
- Projeto e implementação seguros
- **Testes de segurança e avaliação de vulnerabilidades**
- Implantação e manutenção seguras.

Utilizado para **identificar** possíveis vulnerabilidades e ameaças à aplicação.

Teste de Invasão:

Tem como objetivo identificar possíveis vulnerabilidades e

Análise de Vulnerabilidades

É uma técnica que se concentra em analisar a aplicação

Exemplo:

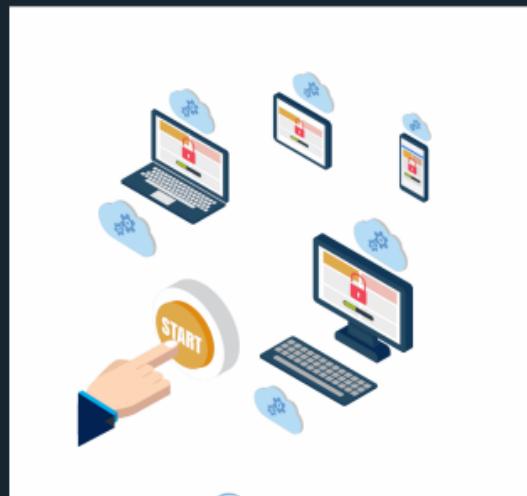
A empresa XYZ está desenvolvendo um software de processamento de pagamentos online.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO



Nos anos 2000, com o crescimento do desenvolvimento ágil de software, o teste de software passou a ser considerado uma parte essencial.





- Planejamento e definição de requisitos de segurança
- Projeto e implementação seguros
- Testes de segurança e avaliação de vulnerabilidades
- **Implantação e manutenção seguras.**

Garantir que a **aplicação** seja implantada e **mantida** de forma **segura**.....

Configuração de Servidor:

Importante configurar o servidor de forma segura, desabilitando serviços desnecessários...

Autenticação e Autorização:

Garantir que somente usuários autorizados possam acessar a aplicação e suas funcionalidades.

Monitoramento:

Busca de possíveis ameaças ou vulnerabilidades.

Exemplo:

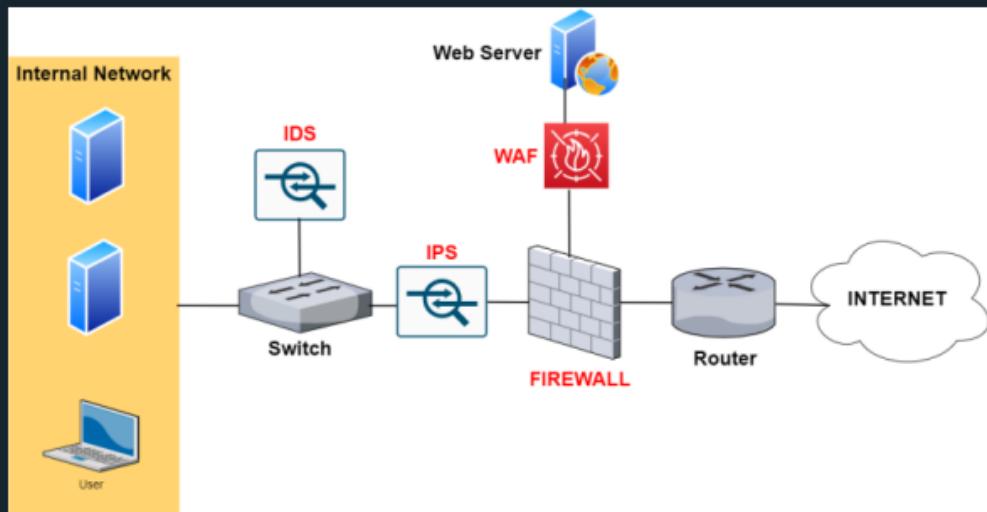
Suponha uma aplicação de e-commerce de uma média empresa e sua infraestrutura.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

b) O CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE SEGURO



A primeira fechadura moderna de segurança foi patenteada em 1778 pelo inventor britânico Robert Barron.



c) MÉTODOS E FERRAMENTAS DE SEGURANÇA
UTILIZADOS.

Métodos e Ferramentas são essenciais para garantir que um software seja desenvolvido e implantado com segurança.



- ANÁLISE DE CÓDIGO ESTÁTICA E DINÂMICA
- TESTE DE INVASÃO E AVALIAÇÃO DE VULNERABILIDADES
- GERENCIAMENTO DE CONFIGURAÇÃO SEGURA
- AUTENTICAÇÃO E AUTORIZAÇÃO SEGURAS
- CRIPTOGRAFIA E GERENCIAMENTO DE CHAVES
- MONITORAMENTO E DETECÇÃO DE INCIDENTES



- Proteção de dados
- Maior confiabilidade
- Redução de custos
- Proteção da reputação



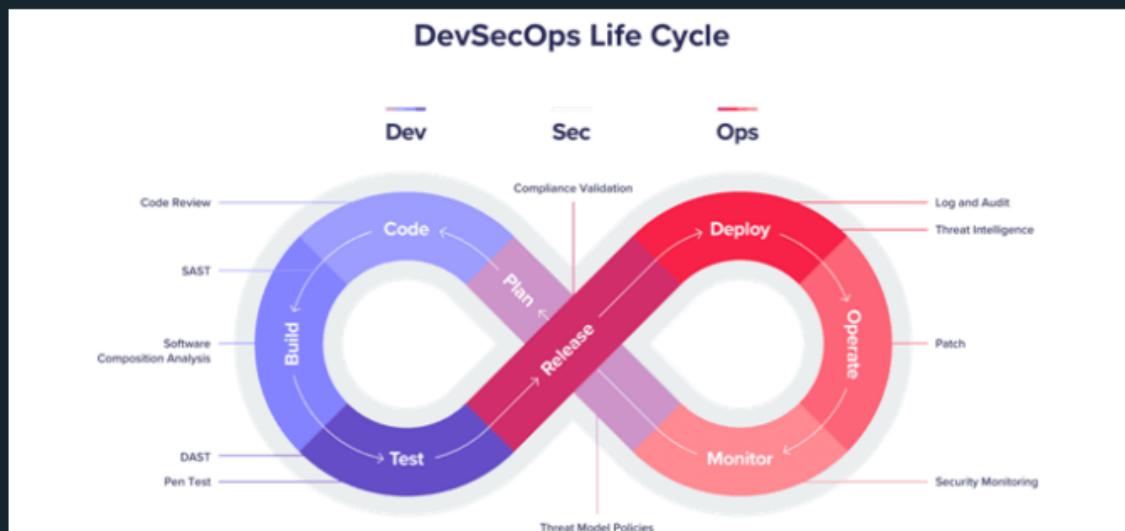
- **Análise de código estática e dinâmica**
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

É uma técnica utilizada para **identificar** possíveis **vulnerabilidades** no código da aplicação



4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.





- **Análise de código estática e dinâmica**
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Estática

SAST

- Análise de Código Fonte
 - SonarQube
 - Veracode
 - ESLint
 - PMD
 - FindBugs

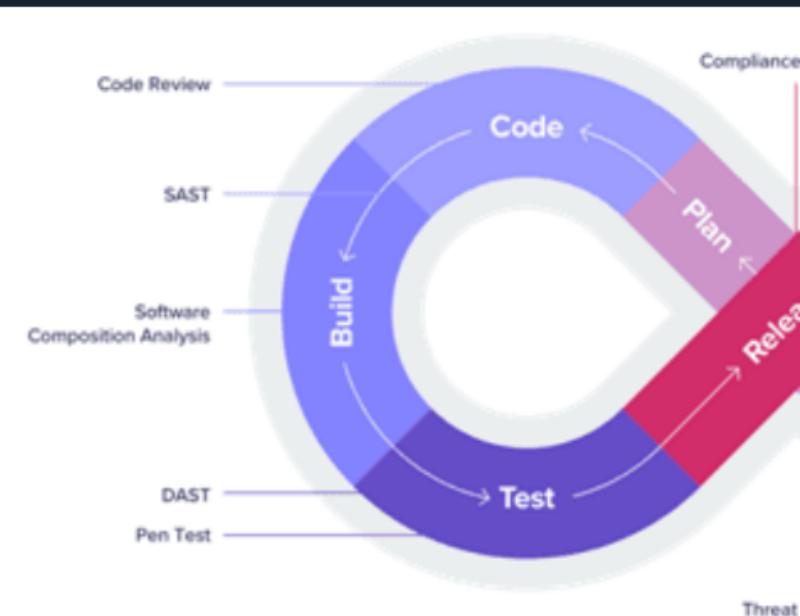
Dinâmica

DAST

- Execução da Aplicação
 - Burp Suite
 - OWASP ZAP
 - Acunetix
 - Qualys
 - IBM AppScan

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.





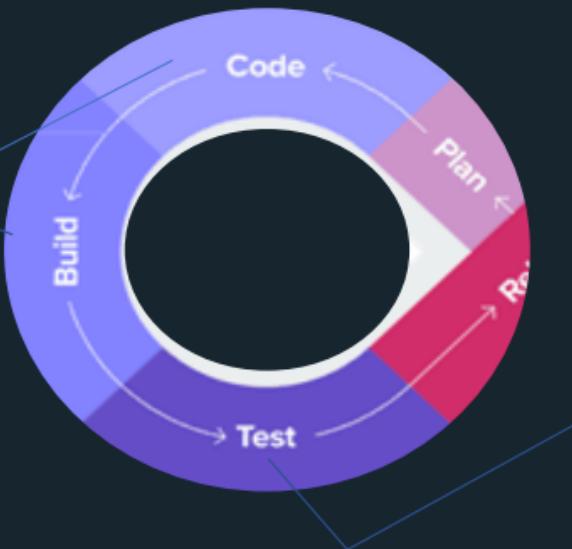
- **Análise de código estática e dinâmica**
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

SAST (estáticas)

- ✓ funções inseguras
- ✓ variáveis não inicializadas
- ✓ vulnerabilidades de buffer overflow
- ✓ Outras



DAST (dinâmicas)

- ✓ Injeção de SQL
- ✓ Ataques de buffer overflow
- ✓ Cross-site scripting (XSS)
- ✓ Cross-site request forgery (CSRF)
- ✓ Vulnerabilidades de autenticação e autorização
- ✓ Outras



- Análise de código estática e dinâmica
- **Teste de invasão e avaliação de vulnerabilidades**
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Identificar possíveis vulnerabilidades



QA testers

Black box - we do not know anything

- **Owasp ZAP**
 - ✓ Injeções de SQL
 - ✓ Cross-site scripting (XSS)
 - ✓ Scripts entre sites (CSRF)
 - ✓ Outros
 - ✓ Simular ataques (testar resposta)

- **Nesus**
 - ✓ Vulnerabilidades em redes
 - ✓ Sistemas operacionais
 - ✓ Aplicativos
 - ✓ Componentes de infraestrutura de TI.
 - ✓ Problemas de configuração de segurança.
 - ✓ Avaliar a eficácia das políticas de segurança.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



A expressão "caixa preta" foi usada pela primeira vez em 1953 pela empresa aeronáutica britânica BEA (British European Airways).

ros

o

- Relatório ou Tabela
- Vulnerabilidades
- Como Corrigir

Documentação



- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- **Gerenciamento de configuração segura**
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

É uma prática utilizada para garantir que a aplicação esteja configurada de forma segura...

Configuração:

Com o Ansible é possível realizar a instalação de patches de segurança em todos os servidores em execução.

Ferramentas:

Ferramentas comuns para gerenciamento de configuração segura incluem **Ansible**, Chef e Puppet.

Simulação

A equipe de desenvolvimento da empresa PUCRS está trabalhando em um novo aplicativo que será executado em um servidor web.

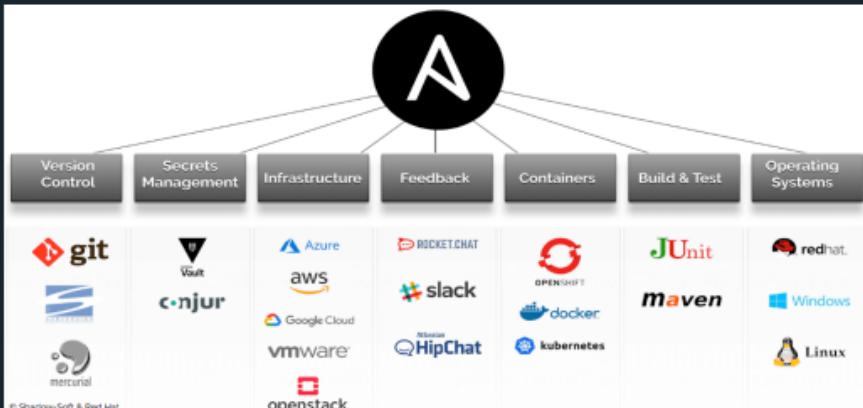
4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



A automação muitas vezes permite que os trabalhadores se concentrem em tarefas mais complexas e criativas.

Permite aos usuários gerenciar e configurar sistemas, aplicativos e serviços de forma eficiente e escalável.





- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- **Gerenciamento de configuração segura**
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Permite aos usuários **gerenciar** e configurar sistemas, aplicativos e serviços de forma eficiente e **escalável**.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

The image shows two screenshots of the Ansible Documentation website. The left screenshot displays the 'All modules' page, which lists various module categories such as All modules, Cloud modules, Clustering modules, Commands modules, Crypto modules, Database modules, Files modules, Identity modules, Inventory modules, and Messaging modules. The right screenshot shows the 'Cloud modules' index, listing providers including Alibaba, Amazon (highlighted with a red underline), Atomic, Azure (highlighted with a red underline), Centurylink, Cloudscale, Cloudstack, DigitalOcean, Dimensiondata, EC2, Google Compute Engine, IBM Cloud, Linode, Oracle VM, Rackspace, SoftLayer, and VMware. A large letter 'A' is overlaid on the left side of the image.

docs.ansible.com/ansible/2.9/modules/list_of_all_modules.html

Documentation

All modules

- a10_server – Manage A10 Networks
- a10_server_axapi3 – Manage A10 Networks
- a10_service_group – Manage A10 Networks
- a10_virtual_server – Manage A10 Networks
- aci_aaa_user – Manage AAA
- aci_aaa_user_certificate – Manage AAA
- aci_access_port_block_to_interface – (infra:HPoT5, infra:PortBk)
- aci_access_port_to_interface – (infra:RsAccBaseGrp, infra:PoE)

contain unfixed security vulnerabilities in the documentation. For Red Hat customers, we recommend using the Red Hat Security Advisories for the latest information.

docs.ansible.com/ansible/2.9/modules/list_of_cloud_modules.html

Documentation

Galaxy Developer Guide

REFERENCE & APPENDICES

Module Index

All modules

Cloud modules

- AliCloud
- Amazon
- Atomic
- Azure
- CenturyLink
- Cloudscale
- Cloudstack
- DigitalOcean
- DimensionData
- EC2
- Google Compute Engine
- IBM Cloud
- Linode
- Oracle VM
- Rackspace
- SoftLayer
- VMware



- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- **Gerenciamento de configuração segura**
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Como o Ansible pode ajudar na esteira de DevSecOps:



Configuração e implantação de infraestrutura segura: Configurando políticas de segurança e gerenciando certificados SSL.



Teste de segurança automatizado: Usando ferramentas de teste de segurança como o OpenVAS.



Gerenciamento de vulnerabilidades: Automatizando a aplicação de patches de segurança e a atualização de sistemas.

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



Monitoramento de segurança: Configurar e gerenciar ferramentas de monitoramento de segurança, como IDS/IPS e SIEM.



Autorização e autenticação: Configurar sistemas de autorização e autenticação, como LDAP, Active Directory.



Auditoria e conformidade: Verificação de configurações de segurança ou a implementação de políticas de conformidade.



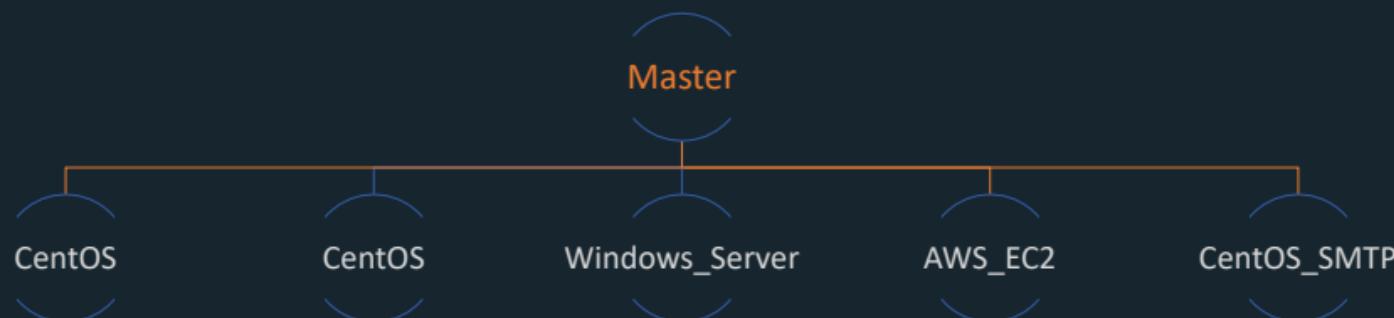
- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- **Gerenciamento de configuração segura**
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



Configuração e implantação de infraestrutura segura: Configurando políticas de segurança e gerenciando certificados SSL.



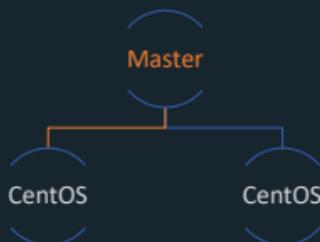


- Análise de código estática e dinâmica
 - Teste de invasão e avaliação de vulnerabilidades
 - **Gerenciamento de configuração segura**
 - Autenticação e autorização seguras
 - Criptografia e gerenciamento de chaves
 - Monitoramento e detecção de incidentes

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

Configuração e implantação de infraestrutura segura:
Configurando políticas de segurança e gerenciando certificados SSL.



```
[vagrant@ansible-master .ssh]$ ansible all -m ping
The authenticity of host '192.168.25.148 (192.168.25.148)' can't
be established.
EDCSA key fingerprint is SHA256:nQaSPzWUp7vAizIqa1NdV+hS6chMjcj
oN8aryGMiP0.
EDCSA key fingerprint is MD5:ff:83:cf:b0:51:8c:de:e9:49:b9:f7:b
4:1a:bf:54:56.
Are you sure you want to continue connecting (yes/no)? 192.168.
25.146 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
yes
192.168.25.148 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[vagrant@ansible-master .ssh]$
```



- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- **Autenticação e autorização seguras**
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Medidas importantes para **garantir** que apenas usuários **autorizados** tenham acesso à aplicação e aos **dados sensíveis**.

Protocolos:

Os mais comuns para autenticação e autorização seguras são OAuth, OpenID Connect e [JWT](#).

SSO (single Sign-on)

Permite aos usuários fazer **login** uma única vez....

Multi-fator

Camada adicional de segurança... SMS, Google Authenticator....

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



Microserviços permitem que as funções sejam distribuídas em sistemas menores e mais ágeis.

Permissões:

Definir **permissões** de acesso específicas para **cada grupo** de usuário.. ..

Monitorar:

Implementar sistemas de monitoramento para **detectar** atividades suspeitas....

Atualização:

Manter a infra estrutura de autenticação, como os servidores e as soluções, sempre atualizadas..



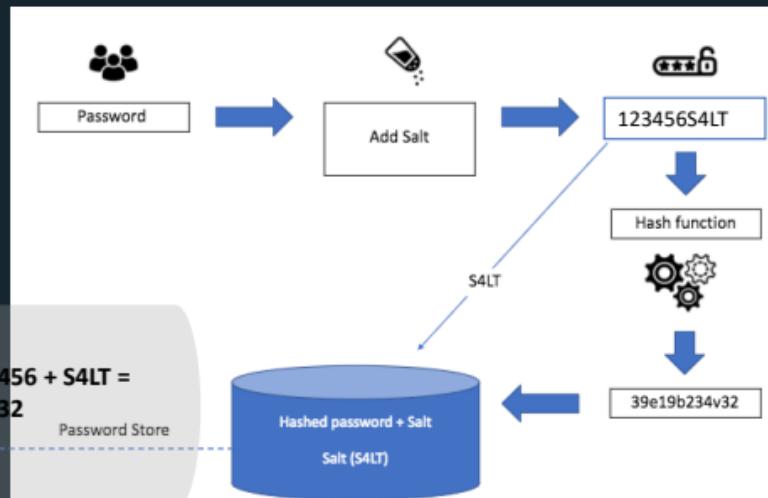
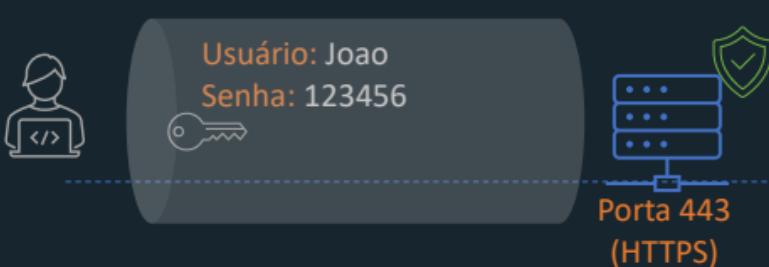
- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- **Autenticação e autorização seguras**
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

Simulação: Como o processo de autenticação pode ser implementado.

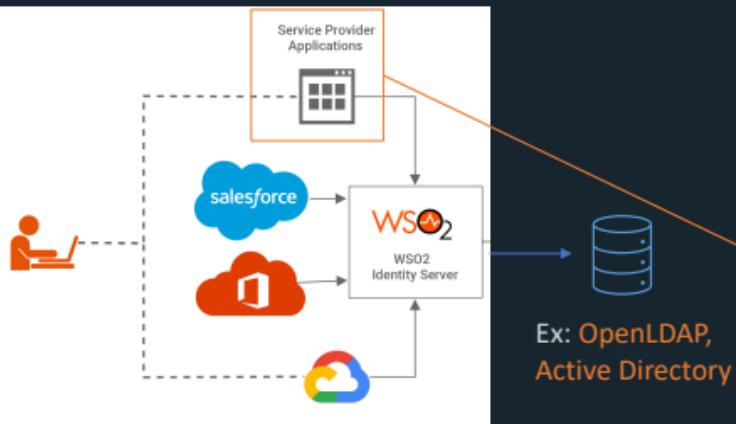
- Validações
 - ✓ Armazenamento Seguro?
 - ✓ Proteção contra ataque de força Bruta?
 - ✓ Login de Sessão?
 - ✓ Uso de HTTPS



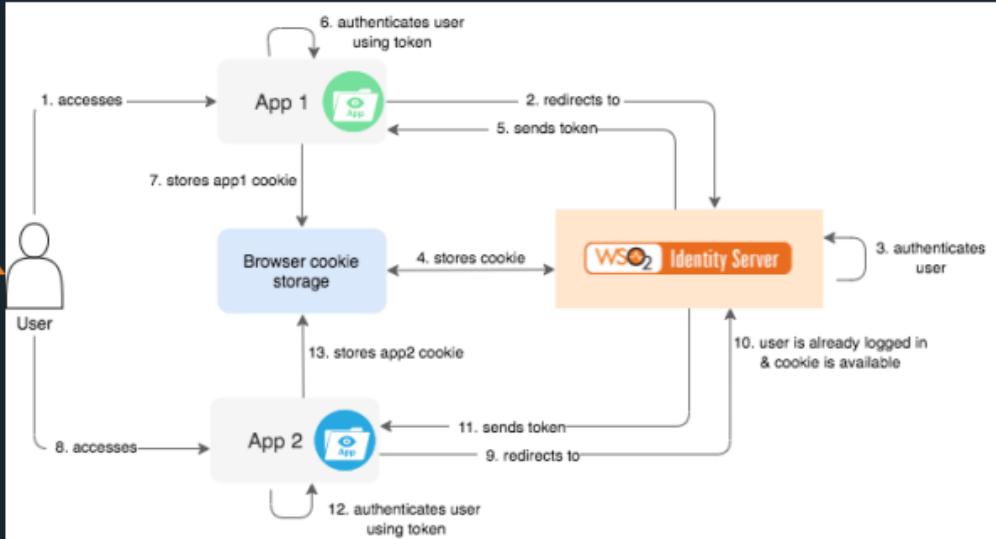
c) MÉTODOS E FERRAMENTAS DE SEGURANÇA
UTILIZADOS.

- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- **Autenticação e autorização seguras**
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Exemplo de Tecnologias – Identidades



Ex.: WSO2 IS **fornecer** acesso ao Office 365 usando sua própria base de dados de identidade.

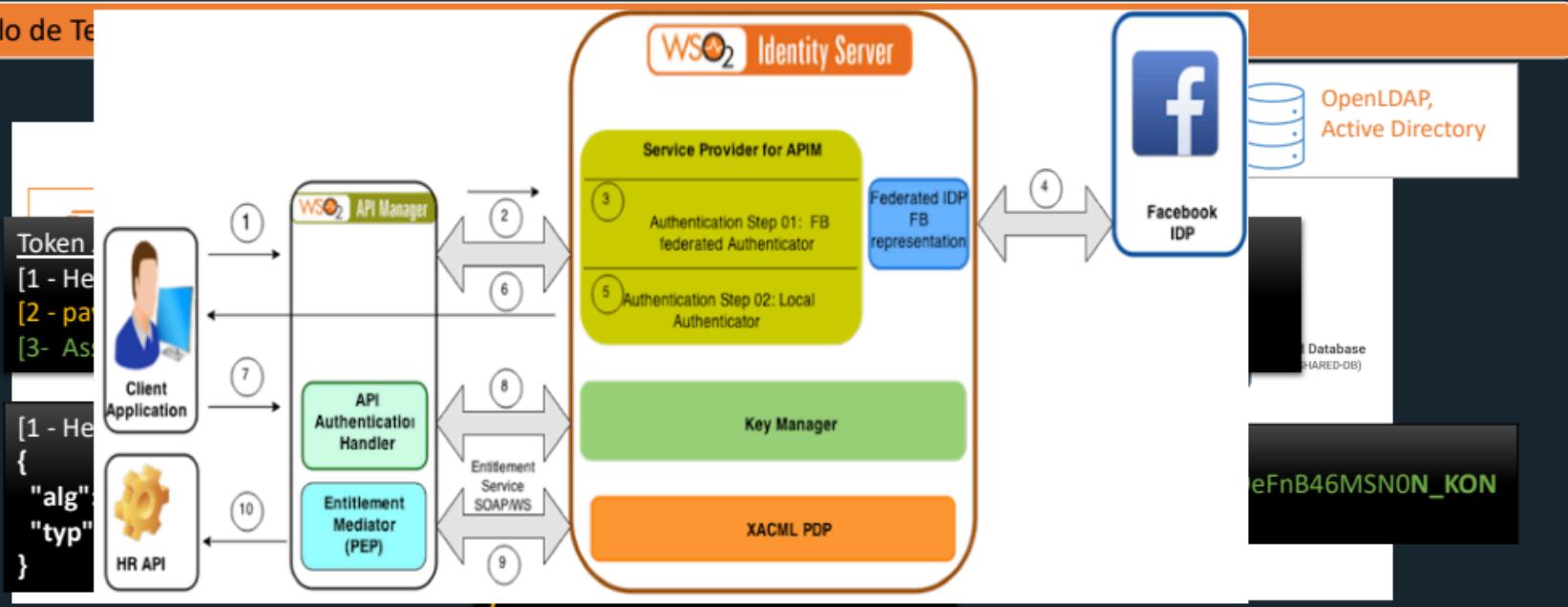


4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- **Autenticação e autorização seguras**
- Criptografia e gerenciamento de chaves
- Monitoramento e detecção de incidentes

Exemplo de Teste





- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- **Criptografia e gerenciamento de chaves**
- Monitoramento e detecção de incidentes

Garantir que as chaves de criptografia sejam armazenadas de forma segura e acessíveis apenas para usuários autorizados.

- **HashiCorp Vault**
- Amazon Web Services (AWS) Key Management Service (KMS)
- Microsoft Azure Key Vault
- Google Cloud Key Management Service (KMS)

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.



Durante o Renascimento, as joias eram armazenadas em cofres especiais, conhecidos como "cassettone".

Simulação

Os desenvolvedores de um aplicativo de pagamentos online devem implementar criptografia para proteger as informações financeiras dos usuários, como números de cartão de crédito.

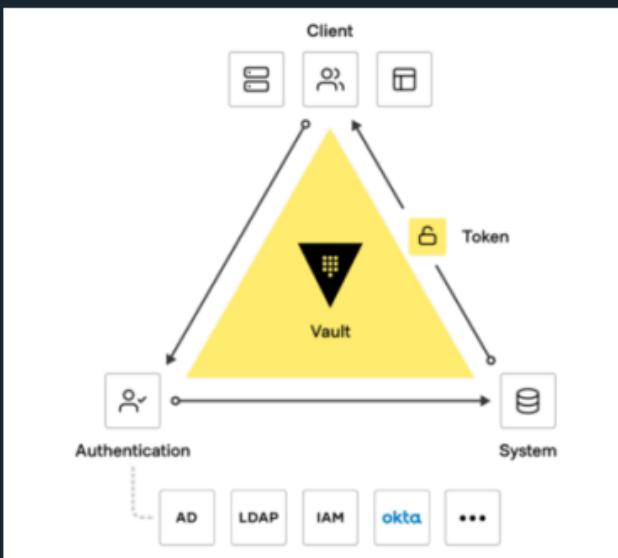


- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- **Criptografia e gerenciamento de chaves**
- Monitoramento e detecção de incidentes

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

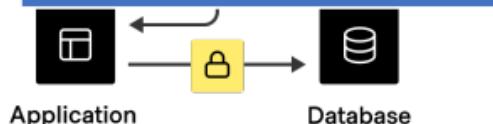
c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

Simulação: Implementar criptografia para proteger as informações de cartão de crédito.



Application Workflow

1. Aplicativo chama API do vault.
2. API do vault retorna chaves de criptografia.
3. Aplicativo criptografa informações financeiras.
4. Aplicativo armazena informações criptografadas.





- Análise de código estática e dinâmica
- Teste de invasão e avaliação de vulnerabilidades
- Gerenciamento de configuração segura
- Autenticação e autorização seguras
- Criptografia e gerenciamento de chaves
- **Monitoramento e detecção de incidentes**

4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE

c) MÉTODOS E FERRAMENTAS DE SEGURANÇA UTILIZADOS.

Identificar possíveis ataques e vulnerabilidades....

Planejamento:

Defina o que precisa ser monitorado.

- **IBM Qradar – IBM .Inc.**

- **Splunk - Splunk Inc**

Coleta de Dados:

Enviar dados para o SIEM.

Exemplo:

1. Um usuário acessa o APP de madrugada.
2. O IBM QRadar detecta um login não autorizado em um aplicativo.
3. O QRadar gera um alerta.

Notificação:

Quando uma violação de segurança for det...

Investigar e Responder:

Tomar medidas para mitigar os danos...

GLOSSÁRIO

- 
1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
 2. MÉTODOS DE CRIPTOGRAFIA
 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
 5. **PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP**
 - a) OWASP e TOP 10 (sub capítulo único)
 6. AUTENTICAÇÃO E AUTORIZAÇÃO

a) OWASP e TOP 10 (sub capítulo único)

Orientações e recursos para ajudar a melhorar a segurança do software e reduzir o risco de violações de dados em aplicativos da web.



- O QUE É OWASP
- OWASP TOP 10



- Conhecer as principais ameaças
- Participar de cursos on-line, Conferências e Workshops
- Comunidade Global



- O que é OWASP
- OWASP Top 10

... conhecida por produzir o "Top 10" de problemas de segurança em aplicativos da web...

Voluntariado:

... sem fins lucrativos composta por voluntários dedicados à segurança do software.

Foco em Segurança

Dedicada exclusivamente à segurança do software....

Global

Uma comunidade global

Aberta e Transparente

Fornece acesso aberto aos seus recursos...

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)



As listas são uma ferramenta popular para ajudar as pessoas a organizar suas tarefas e prioridades diárias.

Please support the OWASP mission to improve software security through Open Source initiatives and community education. [Donate Now!](#)

OWASP

PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org

Store Donate Join

Who is the OWASP® Foundation?

The Open Worldwide Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

- Tools and Resources
- Community and Networking
- Education & Training

For nearly two decades corporations, foundations, developers, and volunteers have supported the OWASP Foundation and its work. [Donate](#), [Join](#), or become a [Corporate Member](#) today.

A **OWASP** (Open Web Application Security Project)
<https://owasp.org/>



- O que é OWASP
- OWASP Top 10

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)

...fornecer uma visão geral das 10(dez) vulnerabilidades mais críticas em aplicativos da web...

<https://owasp.org/www-project-top-ten/>

Situações reais

As vulnerabilidades incluídas na lista são baseadas em dados reais de ataques e explorações de segurança

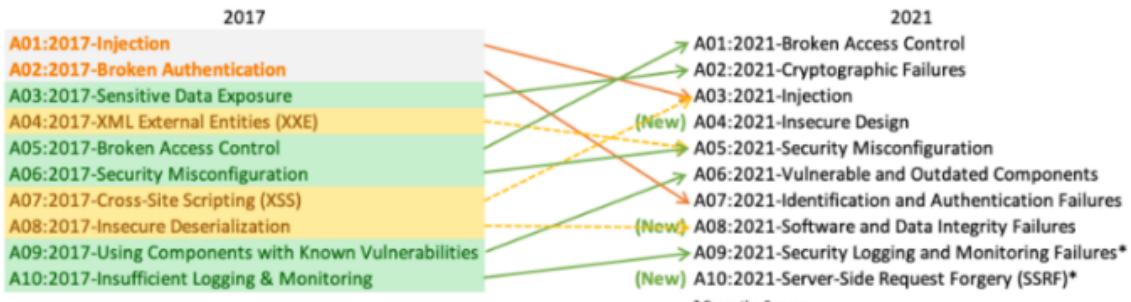
Versões

Desde a versão de 2017, a OWASP passou a adotar um modelo de atualização contínua da lista Top 10...



Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.

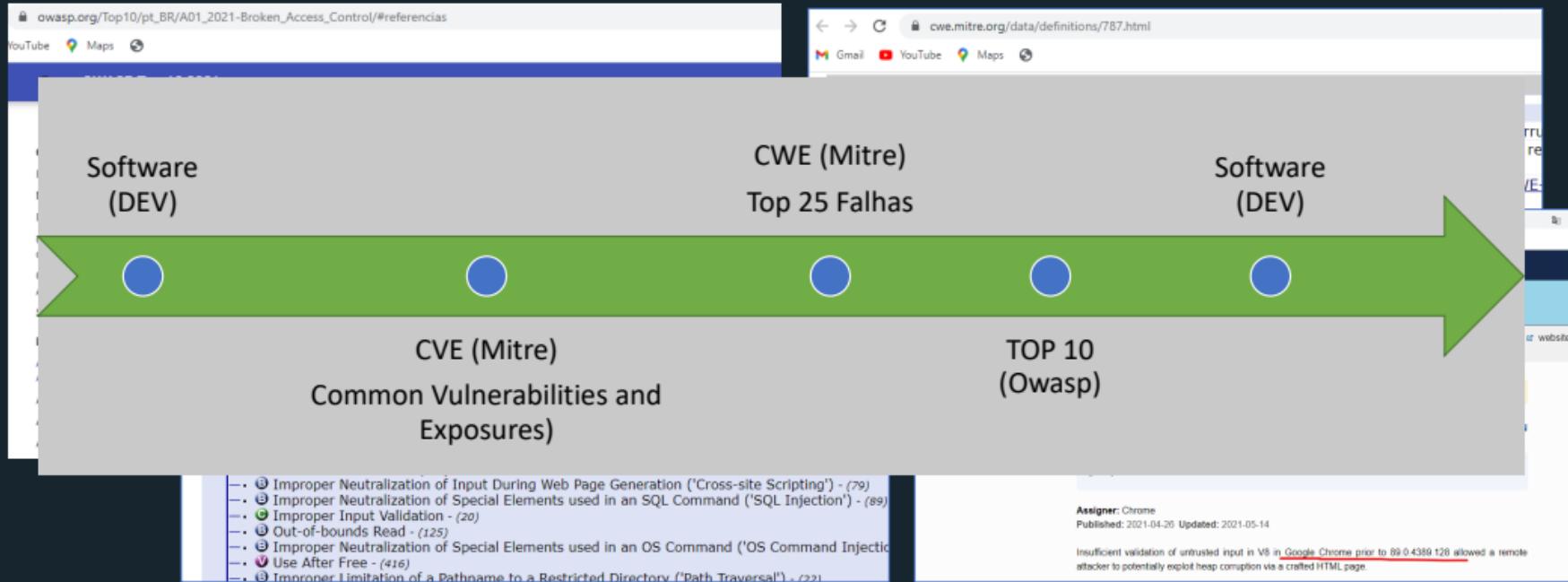




- O que é OWASP
- OWASP Top 10

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)





- O que é OWASP
- OWASP Top 10

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)

Seguir as **práticas** recomendadas da OWASP pode ajudar os desenvolvedores a construir aplicativos mais robustos...



Importante estar ciente da existência de ataques cibernéticos e entender que a segurança deve ser uma prioridade.



1. Quebra de Controle de Acesso
2. Falhas criptográficas
3. Injeção
4. Design inseguro
5. Configuração de segurança incorreta
6. Componentes vulneráveis e desatualizados
7. Falhas na identificação e autenticação
8. Falhas na integridade de software e dados
9. Falhas no registro e monitoramento de segurança
10. Falsificação de solicitação do lado do servidor (SSRF)



- O que é OWASP
- OWASP Top 10

1. Quebra de Controle de Acesso
2. Falhas criptográficas
3. Injeção
- Outros 7

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)



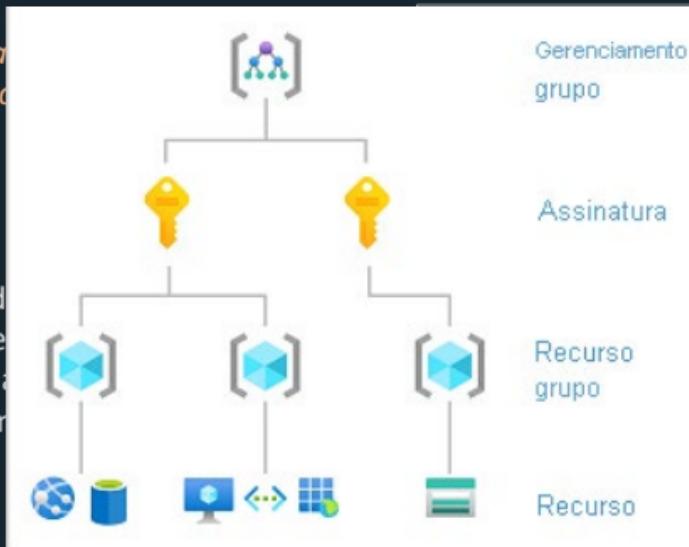
A01:2021-Broken Access Control (Quebra de controle de Acesso)



Falta de controles de acesso a informações confidenciais.



- Violação do princípio de não autorização.
- Ignorar **verificações de permissão** ao modificar a URL ou URLSearchParams.
- Acessando API sem controlar os métodos PUT e DELETE.



O:
ar acessar informações confidenciais
o autorizadas em um sistema.

autenticação forte.
cesso baseado em funções (RBAC) em
web.
es inativas e evitar sessões múltiplas.

- O que é OWASP
- [OWASP Top 10](#)

1. Quebra de Controle de Acesso
 2. Falhas criptográficas
 3. Injeção
- Outros 7

5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)



A02:2021-Cryptographic Failures (Falhas Criptográficas)



Ocorre quando um sistema implementa criptografia de forma inadequada...

Exemplo de exploração:

Uma falha **criptográfica** pode permitir que um invasor decifre informações que deveriam ser **protegidas**.



- Força bruta: tentativas repetidas de decifrar uma chave criptográfica, senhas....
- Interceptação: captura de tráfego de rede para interceptar.



- Utilizar bibliotecas criptográficas confiáveis.
- Usar criptografia forte para armazenar senhas e chaves de autenticação.

- O que é OWASP
- OWASP Top 10

1. Quebra de Controle de Acesso
 2. Falhas criptográficas
 3. Injeção
- Outros 7

5. PROBLEMAS COMUNS DE SEGURANÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)



A03:2021-Injection (Injeção)



Ocorre quando um atacante é capaz de inserir

```
$sql = "SELECT * FROM usuarios WHERE login = '$login' AND senha = '$senha'";  
$resultado = $pdo->query($sql)->fetchAll();
```

Exemplo de exploração:

Um invasor executa as seguintes SQL maliciosas em um banco



- Injeção obtém

```
$stmt = $pdo->prepare("SELECT * FROM usuarios WHERE login = :login AND senha = :senha");  
$stmt->execute([':login' => $login, ':senha' => $senha]);  
$resultado = $stmt->fetchAll();
```

Imagine

de comércio eletrônico. O usuário insere na barra de pesquisa a seguinte entrada: "sapato'; DROP TABLE produtos;--"



- O que é OWASP
- OWASP Top 10

1. Quebra de Controle de Acesso
 2. Falhas criptográficas
 3. Injeção
- Outros 7

5. PROBLEMAS COMUNS DE SEGURANÇA INDICADOS PELA OWASP

a) OWASP e TOP 10 (sub capítulo único)



Vulnerabilidade (A04:2021 até A10:2021)	Característica	Exemplo
4 - Design inseguro	Falha na arquitetura	Exposição de arquivos.
5 - Configuração de segurança incorreta	Configuração não segue melhores práticas	Uso de senhas padrão
6 - Componentes vulneráveis e desatualizados	Componentes desatualizados	Uso de versão OpenSSL desatualizada
7 - Falhas na identificação e autenticação	Validação inadequada de usuários ou sistemas	Falha de validação de token
8 - Falhas na integridade de software e dados	Modificar ou corromper dados ou software	Não validação de upload de arquivo.
9 - Falhas no registro e monitoramento	Monitoramento inadequado	Falha de login não monitorada.
10 - Falsificação de solicitação do lado do servidor	Ações maliciosas em nome do usuário	Comando malicioso executado usando sessão do usuário logado.

GLOSSÁRIO

- 
1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
 2. MÉTODOS DE CRIPTOGRAFIA
 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
 5. PROBLEMAS COMUNS DE SEGURNAÇA INDICADOS PELA OWASP
 6. **AUTENTICAÇÃO E AUTORIZAÇÃO**
 - a) O QUE É E POR QUE SÃO IMPORTANTES
 - b) MÉTODOS DE AUTENTICAÇÃO
 - c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

Permite a **verificação** da identidade dos usuários e o **controle** de acesso a **recursos** com base em permissões e **privilégios**.

a) O QUE É E POR QUE SÃO IMPORTANTES



- O que é AUTENTICAÇÃO
- O que é AUTORIZAÇÃO



- Proteção contra roubo de identidade.
- Impede modificações indevidas.
- Definir o que os usuários acessam.



- O que é AUTENTICAÇÃO
- O que é AUTORIZAÇÃO

a) O QUE É E POR QUE SÃO IMPORTANTES

Autenticação é o processo de **verificar** a identidade de um **usuário** ou sistema.

Senha:

Código secreto que é conhecido apenas por ele.



- Caracteres
- Frase
- Seleção sequencial
-



Smart Cards e outros dispositivos:

Mecanismos de autenticação baseados em posse.



- Smart Card
- Aplicativo de Autenticação
- Chave USB
- ...



Biometria:

Características biológicas únicas.



- Impressão digital
- Reconhecimento facial
- Reconhecimento de íris
- ...



- O que é AUTENTICAÇÃO
- O que é AUTORIZAÇÃO

Garantir que os recursos protegidos

Armazenamento
Garantir que as senhas armazenadas de

Sessões de login
Quando um usuário faz uma sessão de login.

Uso de HTTPS
Importante proteger a comunicação do usuário durante a navegação.

2020					
Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 secs	22 mins	1 hour	8 hours
9	Instantly	2 mins	19 hours	3 days	3 weeks
10	Instantly	58 mins	1 month	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15 bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	6tn years	100 tn years	7qd years

-Data sourced from [HowSecureIsMyPassword.net](https://www.howsecureismy password.net)

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	2 secs	7 secs	31 secs
8	Instantly	Instantly	2 mins	7 mins	39 mins
9	Instantly	10 secs	1 hour	7 hours	2 days
10	Instantly	4 mins	3 days	3 weeks	5 months
11	Instantly	2 hours	5 months	3 years	34 years
12	2 secs	2 days	24 years	200 years	3k years
13	19 secs	2 months	1k years	12k years	202k years
14	3 mins	4 years	64k years	750k years	16m years
15	32 mins	100 years	3m years	46m years	1bn years
16	5 hours	3k years	173m years	3bn years	92bn years
17	2 days	69k years	9bn years	179bn years	7tn years
18	3 weeks	2m years	467bn years	11tn years	438tn years



Learn about our methodology at hivesystems.io/password



- O que é AUTENTICAÇÃO
- O que é AUTORIZAÇÃO

a) O QUE É E POR QUE SÃO IMPORTANTES

Conceder ou negar **permissões** de acesso a **recursos**....

Controle de acesso:

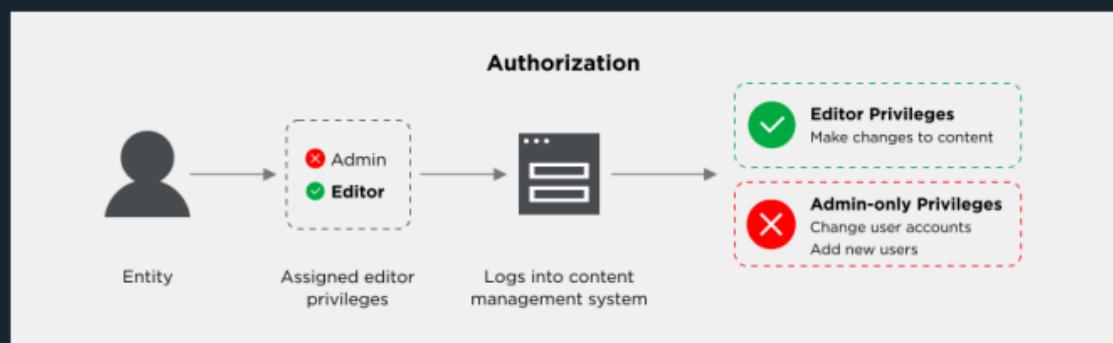
Permite que os administradores de sistemas controlem o acesso a recursos...

Granularidade:

Acesso em recursos específicos ou parte deles...

Flexibilidade:

Configurada de forma flexível para atender às necessidades específicas...





- O que é AUTENTICAÇÃO
- O que é AUTORIZAÇÃO

a) O QUE É E POR QUE SÃO IMPORTANTES

Processo de determinar as permissões de um recurso. Pode ser baseado em Papel/Função ou Regra.

FUNÇÃO ou **REGRA**

- **Função** ou **Regra** dentro do contexto do projeto ou do negócio.

• Gerente de vendas

• Pedidos em horário comercial

Permissões

- Que tipo de **permissões** são necessárias

...dados de vendas

... INSERT entre 9hs - 18hs.

FUNÇÃO



REGRA



Um método de **autenticação** adequado evita **ataques** de hackers e **fraudes**.

b) MÉTODOS DE AUTENTICAÇÃO



- VALIDAÇÃO DA IDENTIDADE
- MFA ADAPTATIVO
- PASSWORDLESS



- Confidencialidade
- Integridade
- Disponibilidade



- **VALIDAÇÃO DA IDENTIDADE**
- MFA ADAPTATIVO
- PASSWORDLESS

b) MÉTODOS DE AUTENTICAÇÃO

Maneira de **saber** se a pessoa que está tentando **acessar** determinadas informações ou **recursos** é realmente a pessoa autorizada.



SABER

- Senhas
- Perguntas e respostas secretas
- Frases secretas
- Códigos PIN



POSSUIR

- Cartões inteligentes
- Tokens de segurança
- Dispositivos biométricos
- Leitores de cartão
- Dispositivos USB de autenticação
- Chaves de segurança
- Telefones celulares
- Smartwatches



SER

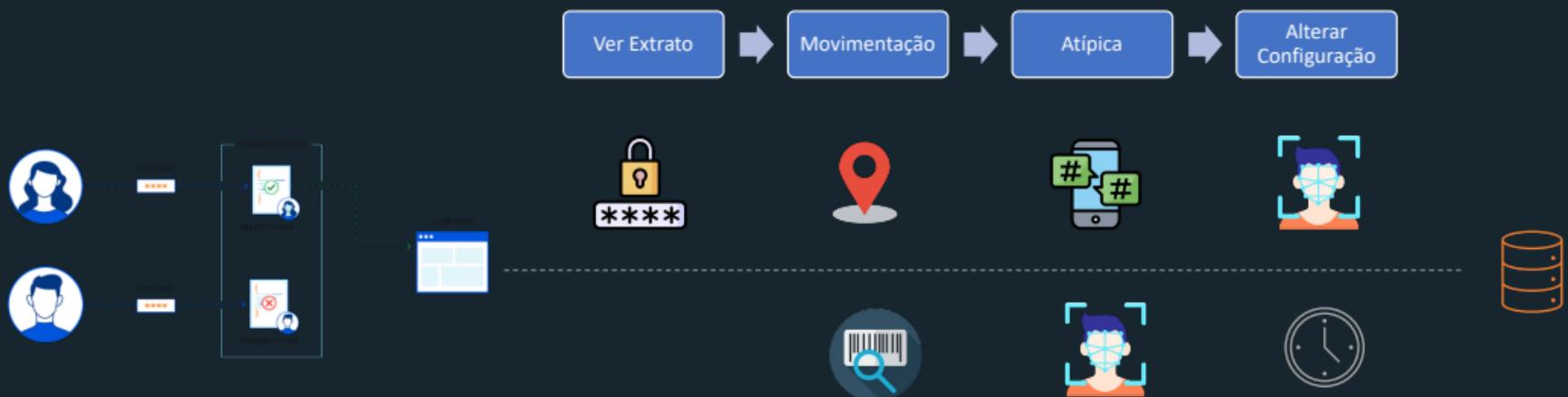
- Voz
- Assinatura
- Reconhecimento de íris
- Reconhecimento de veias
- Padrão de digitação
- Reconhecimento de rosto
- Reconhecimento de retina
- Reconhecimento de palma



- DISPOSITIVOS MAIS COMUNS
- **MFA ADAPTATIVO**
- PASSWORDLESS

b) MÉTODOS DE AUTENTICAÇÃO

Exemplo: Autenticação em uma plataforma financeira.





- DISPOSITIVOS MAIS COMUNS
- **MFA ADAPTATIVO**
- PASSWORDLESS

b) MÉTODOS DE AUTENTICAÇÃO

Exemplo: Autenticação em uma plataforma financeira.





- DISPOSITIVOS MAIS COMUNS
- MFA ADAPTATIVO
- PASSWORDLESS

b) MÉTODOS DE AUTENTICAÇÃO

Abordagem de autenticação que permite aos usuários acessar uma conta **sem** a necessidade de inserir uma senha.



Geralmente a autenticação e autorização envolve o uso de um sistema de **gerenciamento de usuários**.

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO



- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS



- Maior segurança
- Eficiência e Produtividade na gestão.
- Conformidade regulatória



- **SISTEMAS DE GERENCIAMENTO DE USUÁRIOS**
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

example (Groups)

LDAP search configuration

Name

Data source

LDAP

LDAP search

Test

Groups

Data repository

Filters

Limits

Notification

Summary

Search base: CN=Users,DC=cosupport1,DC=com
CN=Users

Search scope: Sub-tree

Search filter: objectclass=group

Name: %CN%

Examples... Defaults Advanced...

Enter the search query information in the fields above, then click on **Next**.

Permite criar e excluir usuários, gerenciar senhas...

LDAP:

Protocolo padrão da Internet para gerenciar diretórios e serviços de rede.

Operações:

Usadas para consultar e modificar entradas no diretório.

Modelo:

Descreve como as informações são organizadas no diretório.

○ E-mail



- **SISTEMAS DE GERENCIAMENTO DE USUÁRIOS**
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

Protocolo LDAP

Redes corporativas baseadas no Windows.



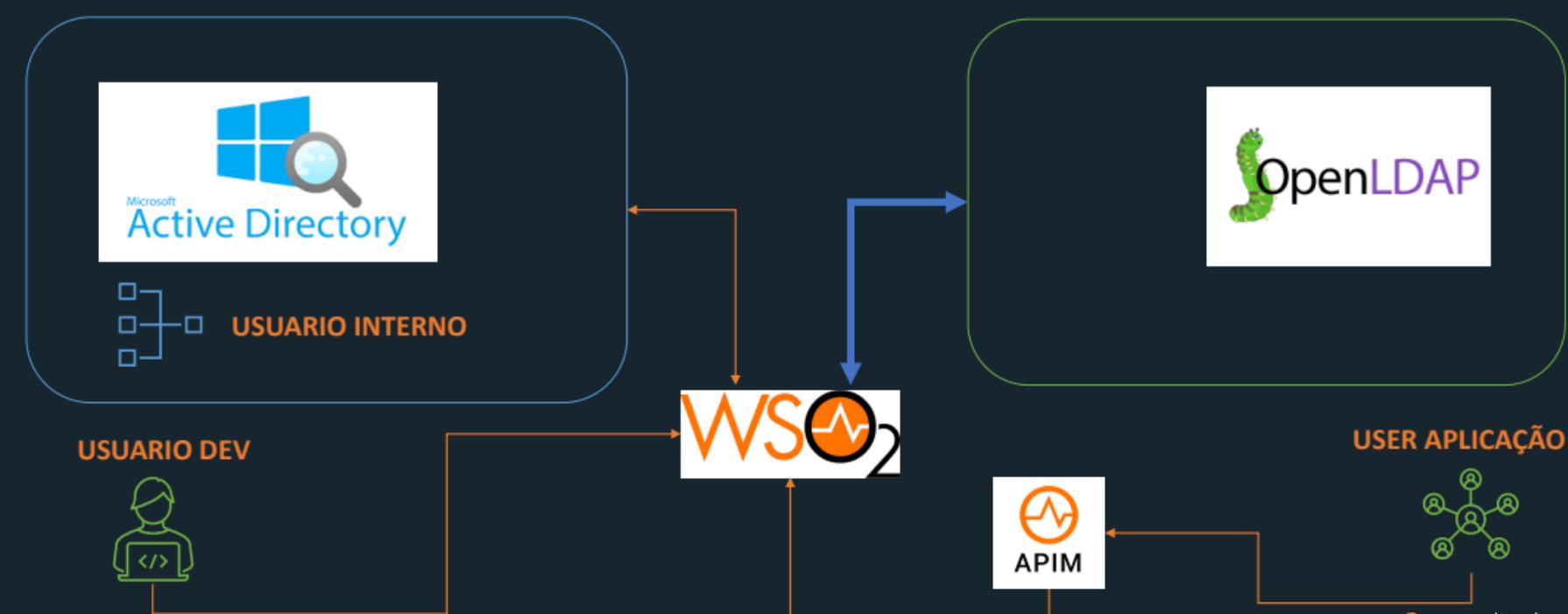
Software livre de gerenciamento de usuários





- **SISTEMAS DE GERENCIAMENTO DE USUÁRIOS**
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO





- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

Processo de determinar as **permissões** de um recurso. Pode ser baseado em Papel/Função ou Regra.

FUNÇÃO ou **REGRA**

- **Função** ou **Regra** dentro do contexto do projeto ou do negócio.

Permissões

- Que tipo de **permissões** são necessárias

• Gerente de vendas

...dados de vendas

FUNÇÃO

• Pedidos em horário comercial

... INSERT entre 9hs - 18hs.

REGRA





- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

- Gerente de vendas

...dados de vendas



FUNÇÃO RuBAC
(role base access control)

GERENTE DE VENDAS

document
<input checked="" type="checkbox"/> delete
<input checked="" type="checkbox"/> create
<input checked="" type="checkbox"/> read
<input checked="" type="checkbox"/> update

VENDEDOR

document
<input type="checkbox"/> delete
<input checked="" type="checkbox"/> create
<input checked="" type="checkbox"/> read
<input type="checkbox"/> update



SOFTWARE DE
VENDAS

- Pedidos em horário comercial

... INSERT entre 9hs - 18hs.



REGRA – RuBAC
(rule base access control)

- Inclusão de Pedidos: **9 as 18hs**
- Envio de e-mail externo: **Sim**
- Excluir documentos: **Não**



- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- **CONTROLE DE ACESSOS**

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO



GERENTE DE VENDAS

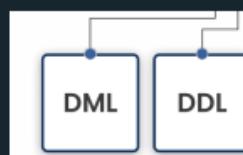
document
<input checked="" type="checkbox"/> delete
<input checked="" type="checkbox"/> create
<input checked="" type="checkbox"/> read
<input checked="" type="checkbox"/> update

VENDEDOR

document
<input type="checkbox"/> delete
<input checked="" type="checkbox"/> create
<input checked="" type="checkbox"/> read
<input type="checkbox"/> update

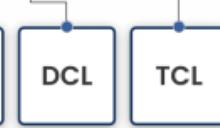


DEV SENIOR



- Select
- Insert
- Update
- Delete

DEV JR



- Create
- Alter
- Drop
- Truncate



- Grant
- Revoke



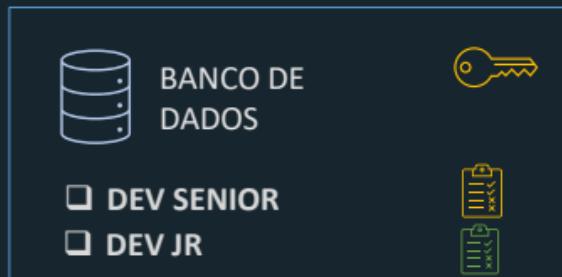
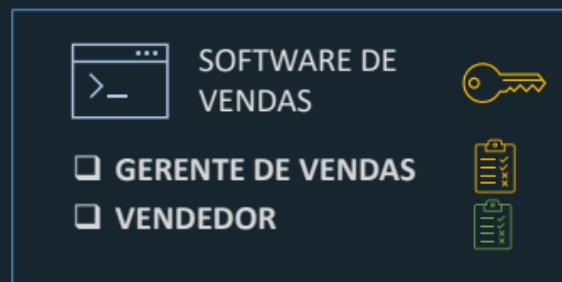
- Save Point
- Roll Back
- Commit

- ❑ Inclusão de Pedidos: **9** as 18hs
- ❑ Envio de e-mail externo: Sim
- ❑ Excluir documentos: **Não**



- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO

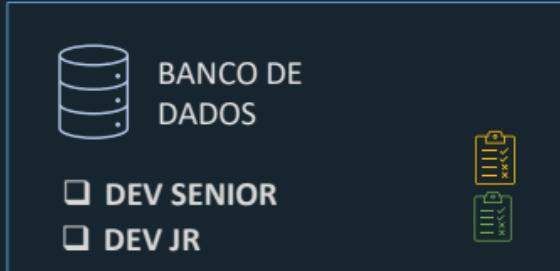
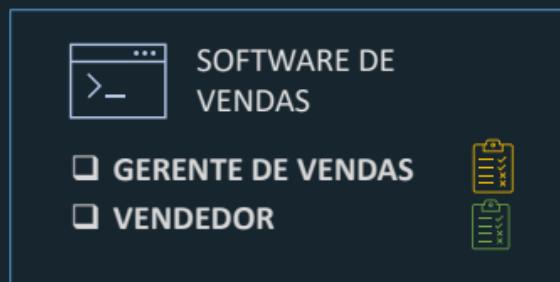


1. Direto no recurso
2. Controlador de Domínio
3. Single SignOn



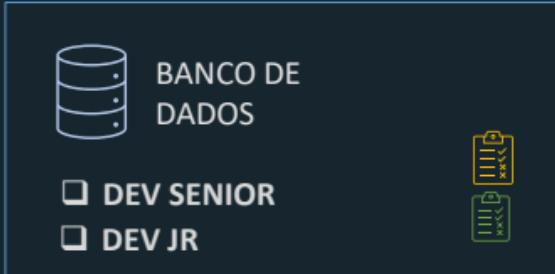
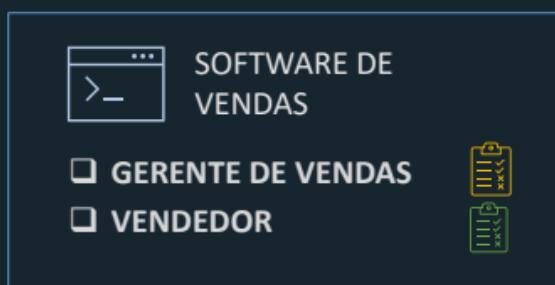
- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO



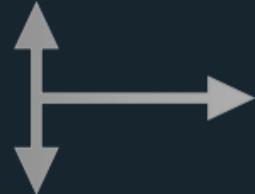
- SISTEMAS DE GERENCIAMENTO DE USUÁRIOS
- CONTROLE DE ACESSOS

c) COMO IMPLEMENTAR AUTENTICAÇÃO E AUTORIZAÇÃO



1. Direto no recurso
2. Controlador de Domínio
3. Single SignOn

TEORIA



EXEMPLO

S
PRÁTICO
S



- 1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
- 2. MÉTODOS DE CRIPTOGRAFIA
- 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
- 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
- 5. PROBLEMAS COMUNS DE SEGURANÇA INDICADOS PELA OWASP
- 6. AUTENTICAÇÃO E AUTORIZAÇÃO

PUCRS online  uol edtech