

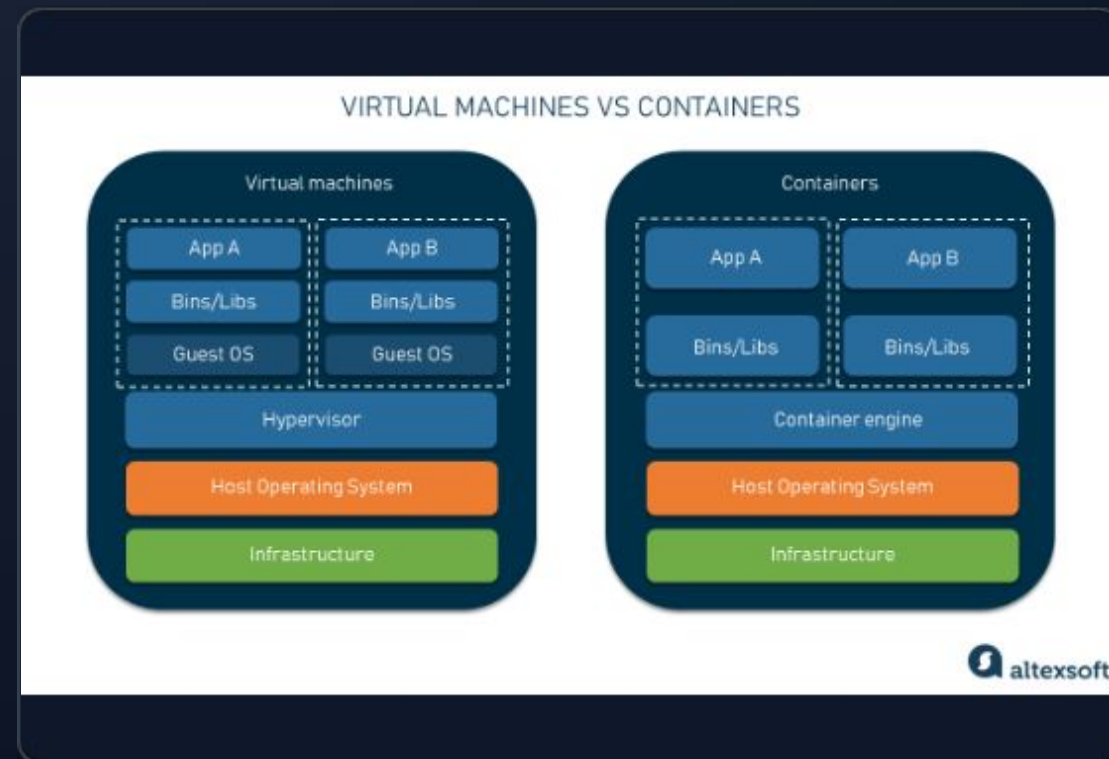
Instalação de Servidor Web Vulnerável e Testes de Penetração

Trabalho T2.2 - Segurança em Computação (2025/2)

Alunos: José Vitor Zorzal e Gustavo Breda Sarti

Visão Geral do Ambiente Configurado

- > **Alvo dos Testes:** OWASP Juice Shop.
- > **Natureza da Aplicação:** Uma aplicação web moderna (Node.js/Angular) desenvolvida deliberadamente com falhas de segurança para fins educacionais e de treinamento.
- > **Objetivo do Trabalho:** Realizar testes práticos de invasão para identificar e explorar falhas clássicas descritas no OWASP Top 10.
- > **Metodologia:** Instalação local garantindo isolamento de rede e reprodutibilidade completa dos ataques registrados.



Arquitetura e Componentes Usados



Hospedagem

Sistema Operacional Windows 11 com **WSL2** (Subsistema Windows para Linux) habilitado para performance nativa em contêineres.



Containerização

Docker Desktop executando a imagem oficial vulnerável `bkimminich/juice-shop`, mapeada e isolada na porta local 3000.



Proxy de Pentest

Burp Suite Community Edition configurado como proxy reverso em `127.0.0.1:8080`, utilizando o navegador Chromium embutido para captura.

Principais Vulnerabilidades Encontradas

1. SQL Injection (Auth Bypass)

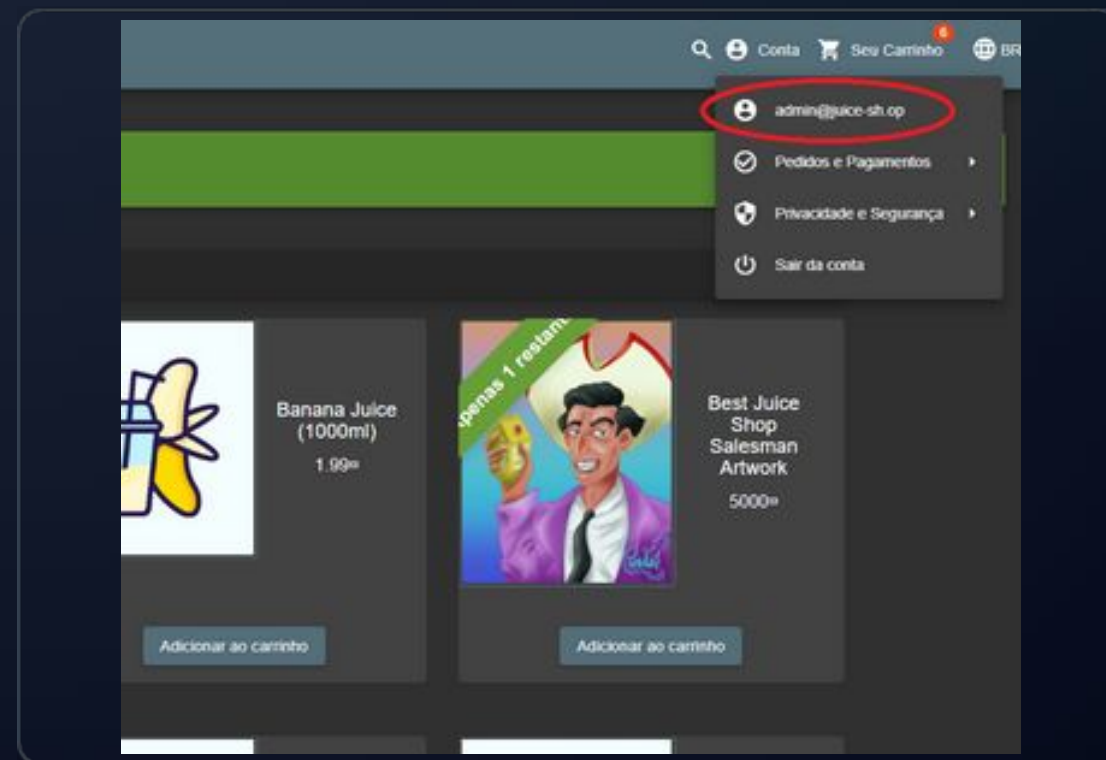
Falha crítica no mecanismo de autenticação da tela de login. A ausência de sanitização nas entradas de dados (e-mail) permite injetar comandos diretamente no banco de dados da aplicação, resultando em **acesso não autorizado a contas privilegiadas** (Administrador) sem o conhecimento prévio da senha.

2. DOM Cross-Site Scripting (XSS)

Vulnerabilidade na manipulação do Document Object Model pela interface de busca. A aplicação reflete dados maliciosos fornecidos via URL diretamente na renderização do HTML, permitindo a **injeção e execução imediata de scripts arbitrários** no lado do cliente (client-side).

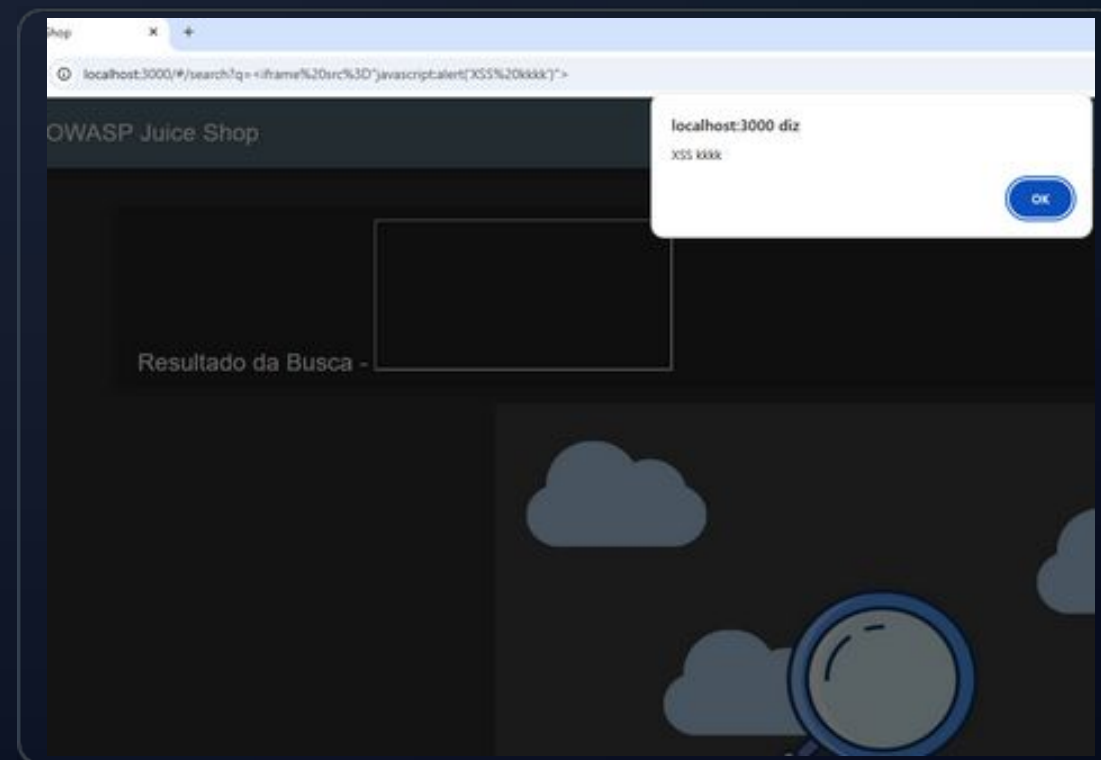
Demonstração: SQL Injection

- > **Vetor de Ataque:** Rota de autenticação /#/login.
- > **Ação Executada:** Interceptação do método POST com o Burp Suite. O tráfego foi pausado para manipulação do corpo JSON.
- > **Payload Injetado:** ' or 1=1-- inserido no campo de e-mail da requisição original.
- > **Efeito Visual:** Bypass instantâneo do login, retornando o Token de autenticação JWT da conta Admin na resposta HTTP (Status 200 OK).



Demonstração: DOM XSS

- > **Vetor de Ataque:** Barra de pesquisa da aplicação (/#/search?q=).
- > **Ação Executada:** Inserção do payload diretamente na interface do usuário (ou via URL no navegador).
- > **Payload Injetado:**
`<iframe src="javascript:alert('XSS kkkk')">`
- > **Efeito Visual:** O navegador interpreta a tag de iframe e executa a função alert nativa do JavaScript, comprovando a falha sem depender de requisições de backend.



Análise Técnica dos Resultados



Dinâmica do SQL Injection O payload manipulou a consulta ao banco de dados com êxito. A aspa simples (') fechou a instrução nativa esperada pelo backend, a cláusula `or 1=1` forçou uma validação logicamente verdadeira (bypass), e os traços duplos (--) comentaram o restante da query que efetuaría a verificação criptográfica da senha.



Dinâmica do DOM XSS O framework de frontend da aplicação (Angular) processou o parâmetro de busca contido na URL e o inseriu no DOM sem executar rotinas de *Output Encoding*. Consequentemente, o navegador do cliente interpretou a tag injetada como HTML legítimo e válido, disparando a execução do script malicioso de imediato.

Conclusões e Mitigações



Mitigação de SQLi

A prevenção exige o abandono da concatenação direta de strings em queries, adotando a implementação estrita de **Prepared Statements** (Consultas Parametrizadas) ou frameworks ORM para tratar dados apenas como texto.



Mitigação de XSS

A proteção requer **Output Encoding** robusto antes de renderizar qualquer entrada no DOM, transformando caracteres especiais em entidades seguras, aliado a uma política estrita de *Content Security Policy (CSP)*.



O Papel do Docker

A tecnologia de contêineres mostrou-se essencial como uma solução segura, rápida e padronizada para isolar ambientes de testes de invasão, protegendo a máquina hospedeira contra efeitos colaterais.