Resumo executivo

Este relatório apresenta os resultados obtidos com o teste de invasão de Juice Shop do tipo Grey box para a disciplina de teste de intrusão a pedido do professor Petrônio. As recomendações deste relatório estão estruturadas para permitir a remediação dos riscos de segurança identificados. Este documento visa a formalização das ações tomadas no referido teste.

Os critérios de avaliação de riscos utilizados durante as atividades consideram as descobertas feitas, bem como os critérios de segurança padrão aceitos na área de segurança.

Este documento recomenda fortemente a revisão dos riscos para o negócio assim como as recomendações de alto nível para uma melhor compreensão dos riscos e questões de segurança descobertas.

### **Visão geral dos resultados**

| **Vulnerabilidade** | **Achados** |
| --- | --- |
| SQL Injection (SQLi) | 1. Falta de validação de entradas de usuários. |
| 2. Uso de concatenação de strings em consultas SQL. |
| 3. Ausência de prepared statements ou queries parametrizadas. |
| 4. Falta de controle de privilégios no banco de dados. |
| Broken Authentication | 1. Falta de autenticação multifatorial (MFA). |
| 2. Gerenciamento inadequado de senhas. |
| 3. Sessões mal controladas. |
| 4. Políticas de segurança mal configuradas. |
| Forgot Password | 1. Falta de verificação de identidade no processo de recuperação. |
| 2. Links ou códigos de recuperação fracos. |
| 3. Exposição de informações sensíveis (ex: dicas de senha). |
| DOM-based XSS (Cross-Site Scripting) | 1. Manipulação inadequada de dados de entrada no lado do cliente. |
| 2. Falta de validação e escape adequado dos dados de entrada. |
| 3. Vulnerabilidade devido ao uso inseguro de JavaScript no navegador. |
| Access to Confidential Document | 1. Falta de controle de acesso a documentos sensíveis. |
| 2. Armazenamento inadequado de documentos confidenciais. |
| 3. Ausência de monitoramento de acesso a dados confidenciais. |
| Forged Feedback (Zero Stars) | 1. Falha em verificar a autenticidade de avaliações e feedbacks. |
| 2. Possibilidade de manipulação de avaliações de produtos ou serviços. |
| 3. Exposição de dados críticos devido a avaliações fraudulentas. |

A tabela abaixo informa a quantidade de vulnerabilidades encontradas e agrupadas de acordo com a respectiva severidade. A severidade, por sua vez, foi determinada de acordo com os critérios definidos pelo CVSS3.1.

| Severidade | Alta | Média | Baixa |
| --- | --- | --- | --- |
| Quantidade | 3 | 2 | 1 |

# 

# Resumo de riscos para o negócio

Vulnerabilidades de severidade alta se configuram em ameaças ao negócio da seguinte forma:

### **Broken Authentication**

A falha de segurança de Broken Authentication se relaciona diretamente com os mecanismos de autenticação do aplicativo. No Juice Shop, como em outros aplicativos, a segurança na autenticação de usuários é fundamental para proteger as informações pessoais e sensíveis, como dados bancários e históricos de pedidos.

A aplicação Juice Shop apresenta falhas nos seus controles de autenticação, como senhas fracas, ausência de autenticação multifatorial ou falhas na manutenção de sessões de usuários, os atacantes podem contornar essas barreiras e acessar as contas de clientes, modificar pedidos ou até roubar informações sensíveis como dados de pagamento. Isso representa não apenas uma falha de segurança, mas também uma ameaça para a reputação e a confiança dos clientes no aplicativo, que é um ponto crítico em qualquer plataforma de comércio eletrônico.

Além disso, uma falha de Broken Authentication pode permitir que um atacante assuma a identidade de um usuário legítimo, realizando transações fraudulentas, acessando informações confidenciais ou até mesmo manipulando os dados de pedidos, comprometendo a integridade do sistema.

A exploração dessa vulnerabilidade no Juice Shop exemplifica os riscos reais que o Broken Authentication pode trazer, como o roubo de identidade, o acesso não autorizado a informações confidenciais, o prejuízo financeiro e a perda de confiança dos usuários, além de potenciais sanções legais. É uma falha crítica que demonstra a importância de se implementar uma autenticação robusta e eficaz para proteger os usuários e o sistema.

### **SQL Injection (SQLi)**

A falha de SQL Injection (SQLi) é uma vulnerabilidade crítica que pode ser explorada por atacantes para manipular o banco de dados do aplicativo. A aplicação Juice Shop, inclui áreas onde inputs de usuários não são adequadamente validados, o que pode ser explorado para injetar comandos SQL maliciosos.

Se um atacante conseguir realizar um SQL Injection, ele pode obter acesso a informações sensíveis armazenadas no banco de dados, como dados pessoais dos usuários, históricos de pedidos e até credenciais de login. Além disso, um ataque de SQLi pode permitir a modificação ou exclusão de dados importantes, comprometendo a integridade do sistema de pedidos, gerando falhas no processamento de pagamentos ou até corrompendo registros vitais para a operação do Juice Shop.

As consequências de uma exploração bem-sucedida dessa falha podem ser devastadoras para o negócio. O roubo de dados de clientes, como informações de pagamento e endereços, poderia resultar em prejuízos financeiros e perda de confiança dos consumidores. O controle total sobre o banco de dados também pode permitir que o atacante realize ações mais graves, como a alteração de preços, modificação de produtos, e até mesmo o comprometimento da segurança de toda a plataforma.

Além disso, um ataque de SQLi pode afetar a reputação da aplicação Juice Shop, causando danos irreparáveis à imagem da empresa, além de possíveis sanções legais em caso de violação de normas de proteção de dados. Portanto, a proteção contra SQL Injection é essencial para manter a segurança e a confiabilidade de qualquer aplicação que lide com dados sensíveis, como o Juice Shop.

### **Forgot Password**

A vulnerabilidade relacionada ao processo de recuperação de senha, como o "Forgot Password", pode ser explorada por atacantes para obter acesso não autorizado às contas dos usuários e, assim, comprometer a segurança do sistema.

Na aplicação Juice Shop, um processo de recuperação de senha mal implementado pode permitir que um atacante contorne as medidas de autenticação e se aproprie de contas de clientes ou até mesmo de administradores. Se o sistema de recuperação de senha não exigir verificações rigorosas de identidade, como autenticação multifatorial (MFA) ou validação adequada de dados, um invasor pode redefinir a senha de qualquer conta apenas acessando o e-mail ou adivinhando as respostas a perguntas de segurança fracas ou óbvias.

Esse tipo de vulnerabilidade representa sérios riscos para a segurança do Juice Shop, pois os atacantes poderiam acessar informações confidenciais dos usuários, como dados bancários, endereços de entrega e históricos de pedidos. Além disso, um atacante poderia modificar dados importantes do sistema, como preços de produtos, ou até realizar transações fraudulentas.

As consequências para o Juice Shop incluem o roubo de identidade, fraude financeira e comprometimento de dados sensíveis. Isso não só causaria danos financeiros, mas também impactaria negativamente a confiança dos clientes, prejudicando a reputação da plataforma e afetando diretamente a relação com os usuários. A exploração dessa vulnerabilidade pode gerar sérios danos à integridade e à credibilidade do Juice Shop, comprometendo sua operação e potencialmente gerando implicações legais devido à exposição de dados pessoais e violação de normas de segurança de dados.

Portanto, garantir um processo de recuperação de senha seguro e robusto, com a implementação de verificações rigorosas de identidade, é crucial para proteger a plataforma e a privacidade dos usuários no Juice Shop.

### **DOM XSS**

A vulnerabilidade **DOM-based XSS** (Cross-Site Scripting) representa um risco significativo para a segurança da aplicação e dos dados dos usuários. No Juice Shop, esse tipo de ataque pode ser explorado através da manipulação de dados fornecidos pelos usuários, como parâmetros de URL ou fragmentos de caminho, que são usados de maneira insegura para alterar o **DOM (Document Object Model)** da página diretamente no navegador.

Se o Juice Shop não validar corretamente esses dados antes de usá-los no código JavaScript, um atacante pode injetar scripts maliciosos diretamente no navegador da vítima. Isso pode levar ao roubo de cookies, informações de sessão e até mesmo credenciais de login dos usuários, que poderiam ser usadas para acessar contas de forma fraudulenta. Além disso, o script malicioso pode redirecionar os usuários para sites fraudulentos, comprometendo ainda mais a segurança dos clientes e do sistema.

O DOM-based XSS é particularmente perigoso para o Juice Shop porque, ao ocorrer no navegador do usuário, o ataque é mais difícil de detectar e prevenir utilizando métodos tradicionais de segurança baseados no servidor. Isso significa que, enquanto o usuário interage com a plataforma de maneira normal, ele pode ser vítima de um ataque sem perceber, comprometendo a confiança da aplicação Juice Shop.

As consequências desse tipo de vulnerabilidade incluem o roubo de informações sensíveis, como dados bancários, informações pessoais e credenciais de acesso. Além disso, um atacante poderia realizar ações maliciosas em nome do usuário, como alterar dados de pedidos ou até fazer compras fraudulentas. Esses danos podem afetar gravemente a reputação do Juice Shop, causar perdas financeiras e prejudicar a relação com os usuários, uma vez que a plataforma deixa de ser vista como confiável.

Portanto, é crucial que o Juice Shop implemente medidas de segurança robustas para evitar o DOM-based XSS, como a validação e o saneamento adequado dos dados fornecidos pelos usuários antes de serem manipulados via JavaScript, garantindo assim a segurança tanto do sistema quanto das informações dos usuários.

### **Access a Confidential Document**

As boas práticas de controle de acesso e proteção de dados confidenciais são essenciais para qualquer aplicação real, como a do Juice Shop, que manipula dados sensíveis de seus usuários, como informações pessoais e detalhes de pagamento.

No Juice Shop, uma vulnerabilidade relacionada ao acesso não autorizado a documentos confidenciais pode comprometer informações sensíveis de clientes, como dados bancários, endereços e históricos de compras. Portanto, é fundamental implementar medidas rigorosas de controle de acesso para garantir que somente usuários autorizados possam visualizar ou alterar esses dados. O princípio do "menor privilégio" também deve ser aplicado, permitindo que cada usuário tenha acesso apenas às informações necessárias para realizar suas funções, minimizando o impacto caso uma conta seja comprometida.

Além disso, a autenticação multifatorial (MFA) é uma medida importante para proteger as contas dos usuários no Juice Shop. Ao exigir múltiplas formas de autenticação, como senha e código enviado para o celular, o risco de acessos indevidos é significativamente reduzido, aumentando a segurança da plataforma contra ataques, como phishing ou força bruta.

A criptografia de dados também é essencial. No Juice Shop, se informações sensíveis não forem adequadamente criptografadas, um atacante que tenha acesso ao banco de dados poderia ler, modificar ou divulgar esses dados. Mesmo que o atacante consiga acessar os documentos ou registros confidenciais, a criptografia garante que os dados permaneçam ilegíveis sem a chave de decriptação correta.

A realização de auditorias regulares e o monitoramento de acessos também são cruciais para garantir a integridade da plataforma. Com registros detalhados, seria possível identificar rapidamente atividades suspeitas e realizar ações corretivas antes que um problema maior ocorra. Isso é especialmente importante no Juice Shop, já que ele é projetado para testar a exploração de falhas de segurança, e um monitoramento eficaz ajudaria a detectar atividades anômalas e mitigar riscos.

Por fim, o Juice Shop deve garantir que os dados confidenciais não sejam expostos em locais acessíveis ao público, como páginas ou endpoints não protegidos, ou em sistemas de backup inseguros. A configuração inadequada dos sistemas pode resultar em vazamentos de informações, afetando a confiança dos usuários na plataforma e prejudicando a reputação da empresa. O armazenamento seguro desses documentos e a educação dos usuários e administradores sobre as boas práticas de segurança são medidas fundamentais para proteger dados sensíveis e garantir a operação segura da plataforma.

Portanto, a implementação dessas boas práticas de segurança, incluindo controles de acesso, autenticação forte, criptografia, auditorias e armazenamento seguro, são essenciais para proteger a integridade e a confidencialidade dos dados no Juice Shop e, garantir a segurança e confiança dos usuários.

Vulnerabilidades de severidade média se configuram em ameaças ao negócio da seguinte forma:

### **Forged Feedback (Zero Stars)**

A vulnerabilidade de Forged Feedback (Zero Stars) pode ter implicações significativas para a plataforma do Juice Shop, assim como em qualquer loja online, os feedbacks e avaliações dos produtos são um aspecto importante da interação com os clientes. No entanto, se o sistema de avaliações não for devidamente protegido, ele pode ser manipulado por usuários mal-intencionados para publicar avaliações falsas ou forjar a pontuação dos produtos.

Essa falha ocorre quando o Juice Shop não tem um controle rigoroso sobre a origem das avaliações ou falha em validar adequadamente a identidade dos avaliadores. Por exemplo, se o sistema permitir que qualquer pessoa (mesmo sem realizar uma compra ou interação genuína com o produto) deixe um comentário ou avaliar um item, isso cria uma brecha para que um atacante manipule a pontuação de um produto ou serviço, atribuindo notas extremamente baixas ou falsas (como zero estrelas) sem justificativa.Podendo resultar em feedbacks fraudulentos que distorcem a percepção do cliente sobre a qualidade de um produto ou serviço.

Se um atacante for capaz de manipular as avaliações de produtos no Juice Shop, isso pode prejudicar a reputação do produto e da plataforma como um todo, causando um impacto direto na confiança dos usuários. Avaliações falsificadas podem levar os clientes a tomar decisões erradas, como evitar comprar produtos que na realidade são bons, apenas por causa de uma falsa avaliação negativa. Esse tipo de manipulação pode criar uma falsa impressão da qualidade dos produtos no Juice Shop, comprometendo a integridade do sistema de avaliações e diminuindo sua credibilidade.

Além disso, os riscos para o Juice Shop incluem danos à sua reputação, perda de confiança dos clientes e uma possível diminuição nas vendas ou no uso da plataforma. Em um cenário mais amplo, a falta de confiança nos sistemas de avaliação pode fazer com que os usuários duvidem de todas as avaliações de produtos, o que tornaria o processo de feedback ineficaz e prejudicaria a própria experiência de compra na plataforma. Portanto, é essencial que o Juice Shop implemente mecanismos de segurança adequados para garantir que as avaliações sejam genuínas e provenientes de usuários legítimos, como verificações de identidade ou sistemas de autenticação robustos.

A proteção contra Forged Feedback é importante não apenas para manter a integridade das avaliações e da reputação do Juice Shop, mas também para garantir que os usuários possam confiar nas recomendações e fazer escolhas de compra informadas e seguras.

Vulnerabilidades de severidade baixas se configuram em ameaças ao negócio da seguinte forma:

# Recomendações de alto nível

Levando em consideração todas as descobertas feitas, este relatório recomenda fortemente:

Priorizando a segurança desde o desenvolvimento, contratando profissionais especializados em segurança cibernética e garantindo que a equipe de TI esteja devidamente treinada para integrar a segurança ao processo de criação e manutenção de sistemas. Essa preparação inclui o uso de tecnologias seguras, como Prepared Statements e Stored Procedures, que evitam a manipulação indevida de consultas ao banco de dados. Além disso, é importante reforçar o processo de autenticação dos usuários, implementando autenticação multifatorial (MFA) e processos seguros de recuperação de senha. A recuperação de senha, por exemplo, deve ser configurada com links temporários e prazos curtos, evitando que dados sensíveis sejam expostos a ataques.

No que se refere à proteção do banco de dados, é crucial adotar medidas que limitem o acesso, garantindo que as contas de banco de dados possuam apenas os privilégios necessários para a execução das operações essenciais. Esse controle é uma forma eficaz de minimizar o impacto em caso de tentativa de ataque. No caso de manipulação de feedbacks, deve-se validar e verificar a autenticidade das avaliações, permitindo que apenas usuários verificados, que realmente utilizaram os produtos ou serviços, possam deixar suas opiniões. Além disso, é importante monitorar comportamentos suspeitos que possam indicar tentativas de manipulação, como avaliações excessivamente negativas ou criadas em massa.

A proteção contra o XSS exige que todas as entradas dos usuários sejam devidamente validadas e sanitizadas, impedindo que códigos maliciosos sejam injetados no código do site. A implementação de Políticas de segurança também pode ser uma ferramenta útil para restringir quais fontes de conteúdo podem ser executadas em um site, dificultando a execução de scripts maliciosos nos navegadores dos usuários.

Além disso, é fundamental realizar auditorias de segurança periódicas e testes de penetração, de forma a identificar vulnerabilidades antes que elas possam ser exploradas. O monitoramento constante do sistema também é uma prática essencial para detectar atividades suspeitas rapidamente e agir de forma preventiva. Para garantir que o sistema esteja sempre protegido, é imprescindível manter o software, as bibliotecas e as plataformas atualizados com os últimos patches de segurança, além de investir em ferramentas de defesa, como firewalls de aplicação web (WAF), que ajudam a bloquear que possíveis ataques cheguem aos servidores.

Outro aspecto importante é a educação constante dos funcionários sobre práticas seguras de desenvolvimento e segurança cibernética. Oferecer treinamentos regulares e orientações sobre a criação de senhas seguras e a proteção contra fraudes é uma medida simples, mas eficaz, para reduzir o risco de ataques. A transparência também desempenha um papel crucial: em caso de um ataque, é essencial comunicar de maneira clara aos clientes o ocorrido e oferecer suporte imediato, demonstrando comprometimento com a segurança e com a resolução do problema.

Ao adotar práticas seguras de desenvolvimento, investir em tecnologias de segurança, realizar auditorias regulares e manter a transparência com os clientes, pode garantir a integridade dos dados da empresa, proteger a confiança do cliente e minimizar os impactos de possíveis ataques.

# Ferramentas utilizadas

As seguintes ferramentas foram utilizadas durante os testes realizados:

* Burp suite;
* Docker;
* Calculator CVSS;
* MITRE CWE;

# Metodologia

Foram aplicadas metodologias e padrões conhecidos e bem aceitos na área de segurança cibernética, sendo possível destacar:

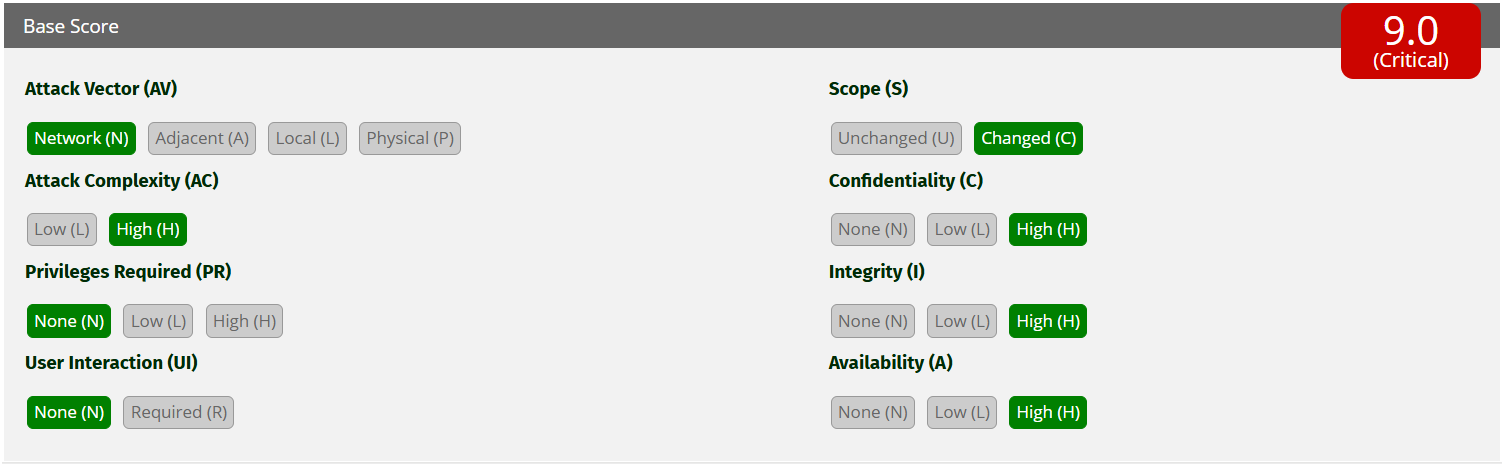
* OWASP Testing Guide
* OWASP Top 10 Application Security Risks - 2021
* SANS: Conducting a Penetration Test on an Organization
* SANS 25 Security Threats

# Detalhes técnicos das descobertas

Broken Authentication

Severidade (CVSS3.1): 3 (Alta)

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



CWE:

CWE-287: Improper Authentication;

CWE-522: Insufficiently Protected Credentials

Localização:

* <http://localhost:3000/#/login>

Descrição:

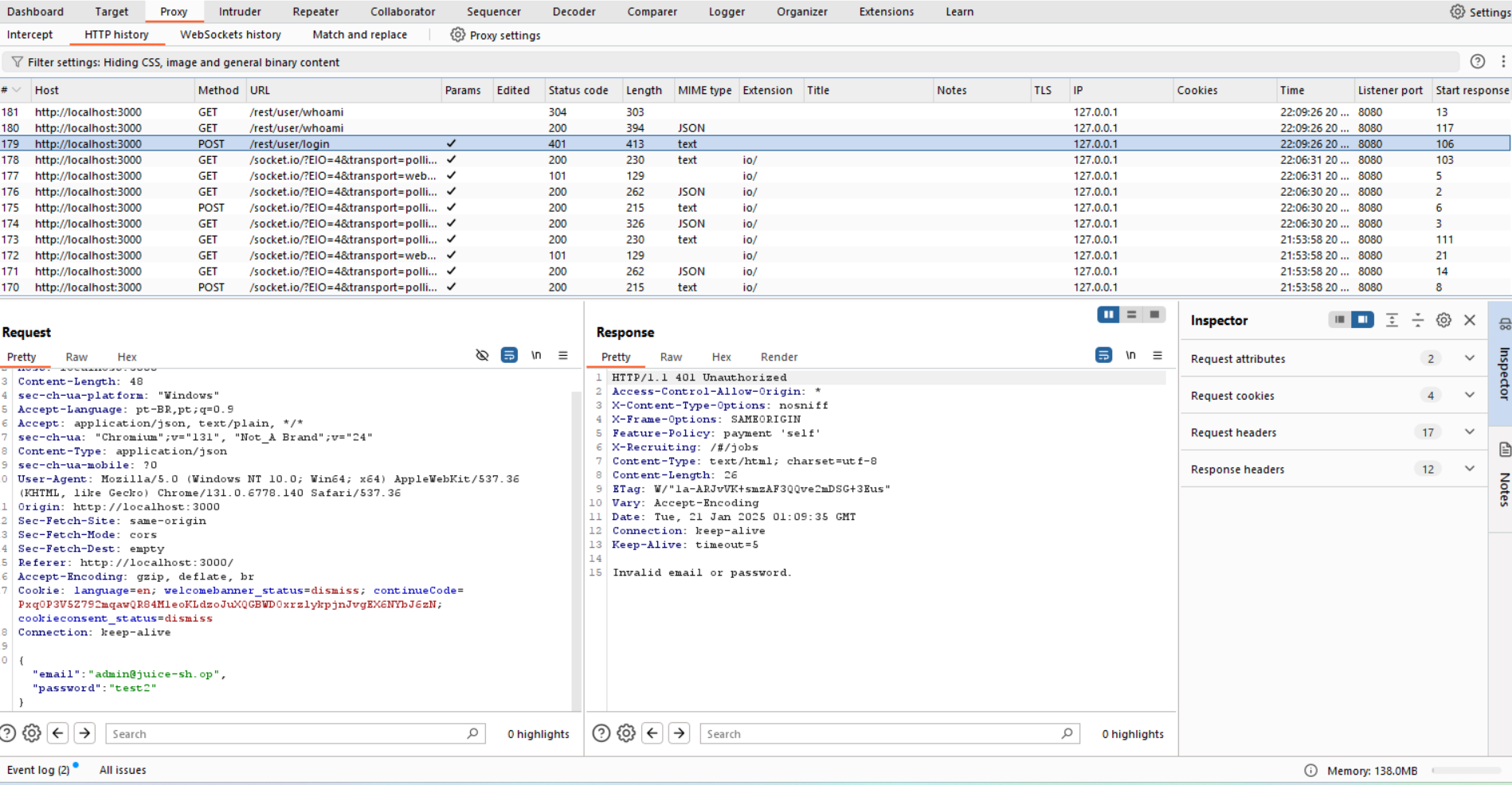
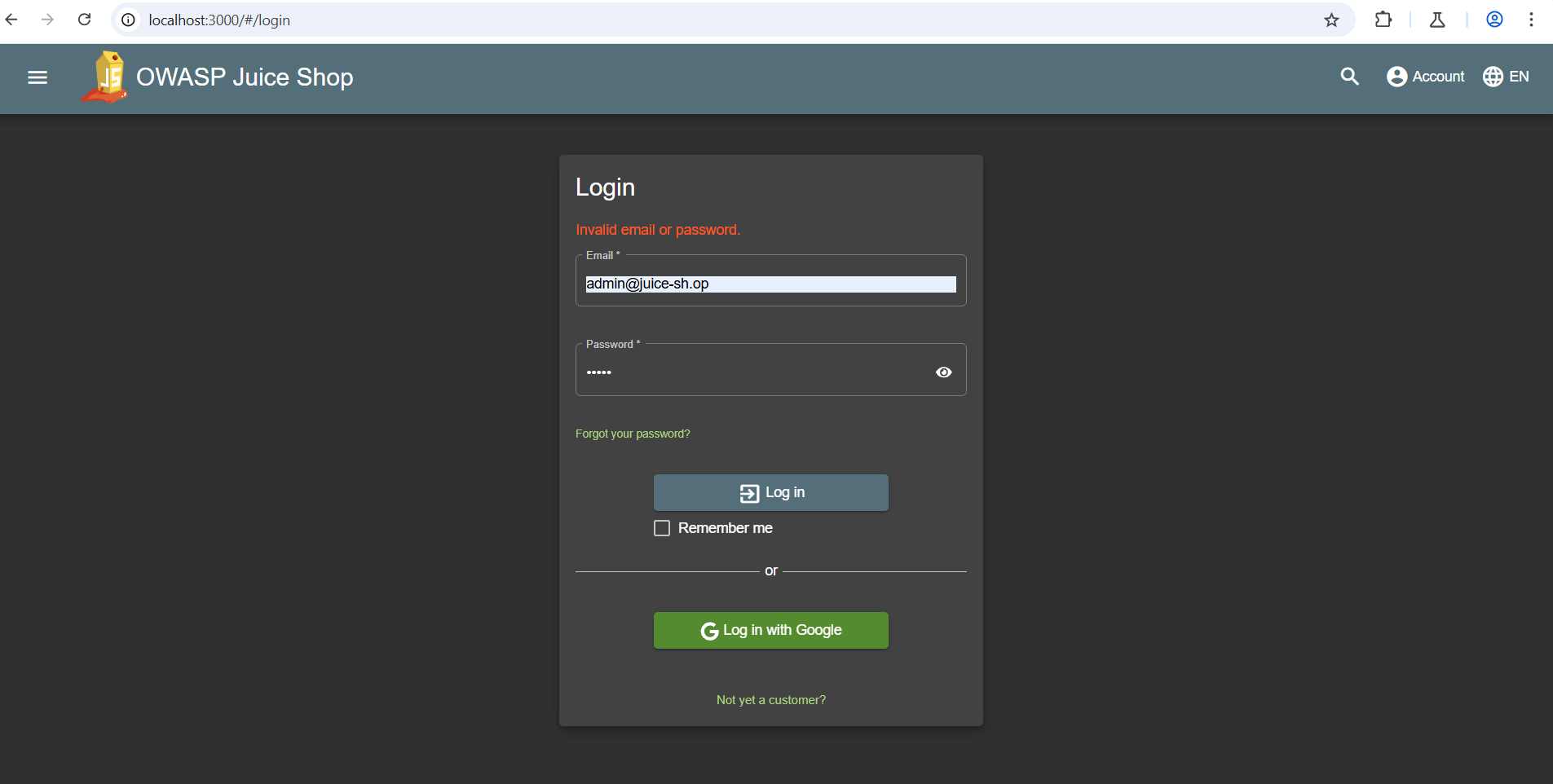
"Broken Authentication" (autenticação quebrada) é uma das vulnerabilidades mais comuns em aplicações web e está listada como parte do **OWASP Top 10**. Ela envolve falhas na forma como os sistemas tratam o processo de autenticação e gestão de sessões de usuários, permitindo que atacantes possam contornar o controle de acesso de maneira maliciosa.

Para realizar esse tipo de ataque, foi executado um processo de coleta de dados. O atacante, portanto, teve que acessar os comentários da página do Juice Shop para recolher os e-mails dos usuários. Em seguida, esses dados foram encaminhados para o Repeat, onde foi feita uma análise do usuário inserido para o teste. Durante o processo, foi observada uma resposta de não autorizado. Logo após, a lista de usuários coletados foi enviada para o Intruder, onde foram testadas algumas senhas fracas e previsíveis.

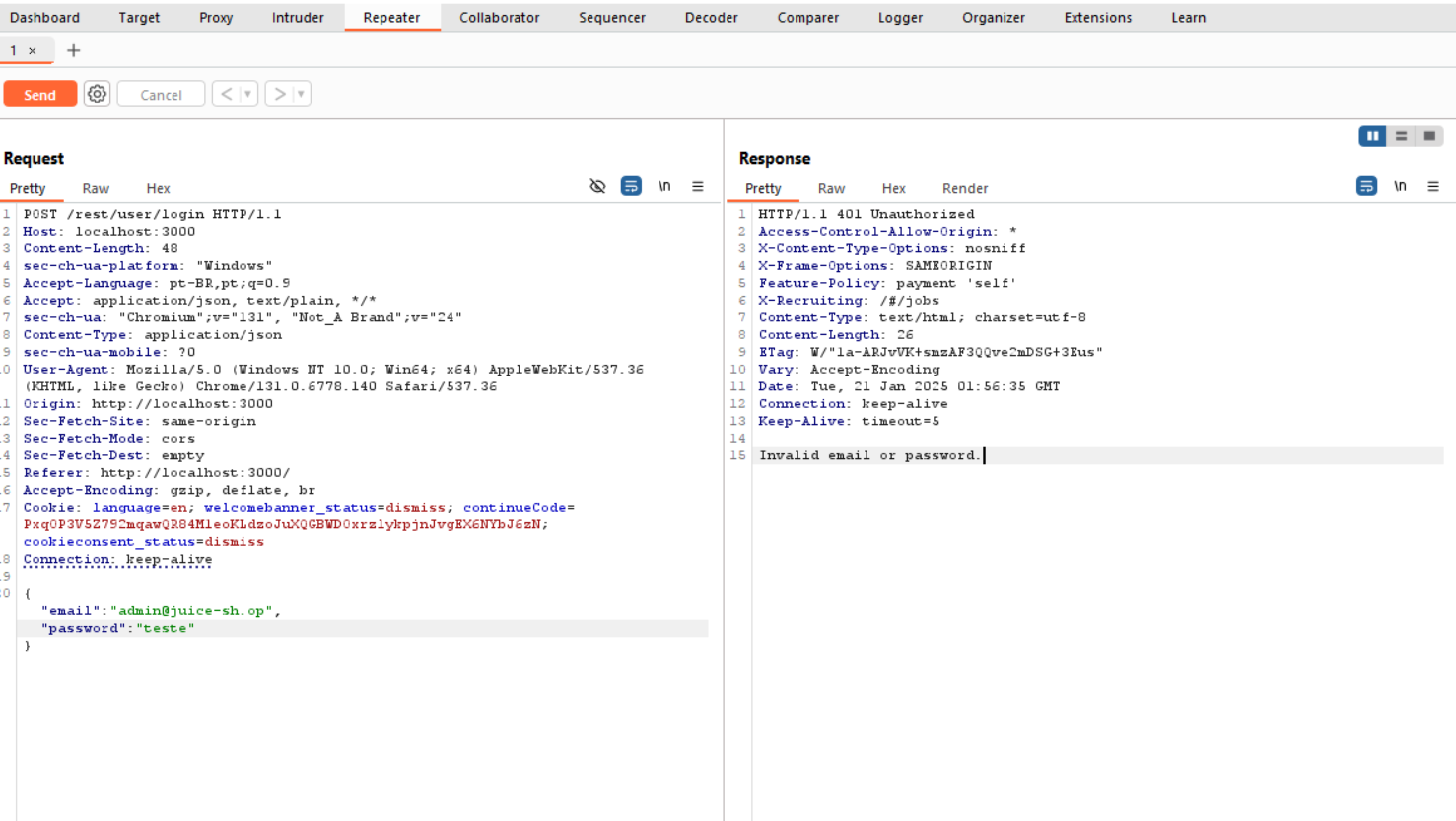
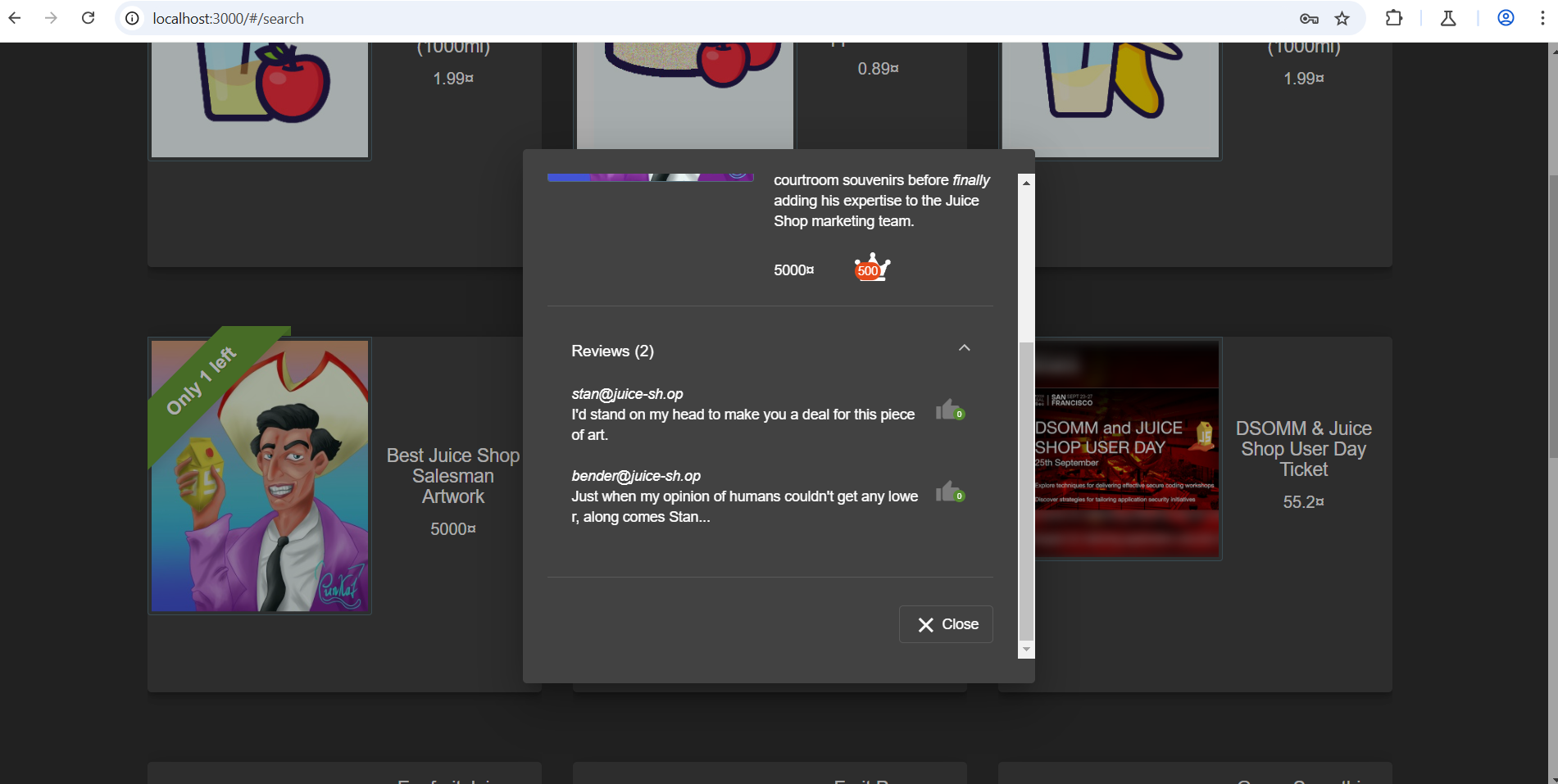
As consequências desse tipo de ataque para o usuário podem incluir roubo de identidade, acesso não autorizado a contas, perda de dados, danos à reputação, prejuízos financeiros e impactos emocionais. Para a organização, as repercussões podem ser igualmente graves, como a violação de dados e exposição de informações confidenciais, danos à reputação, perdas financeiras, interrupção de serviços devido a ataques e possíveis implicações legais.

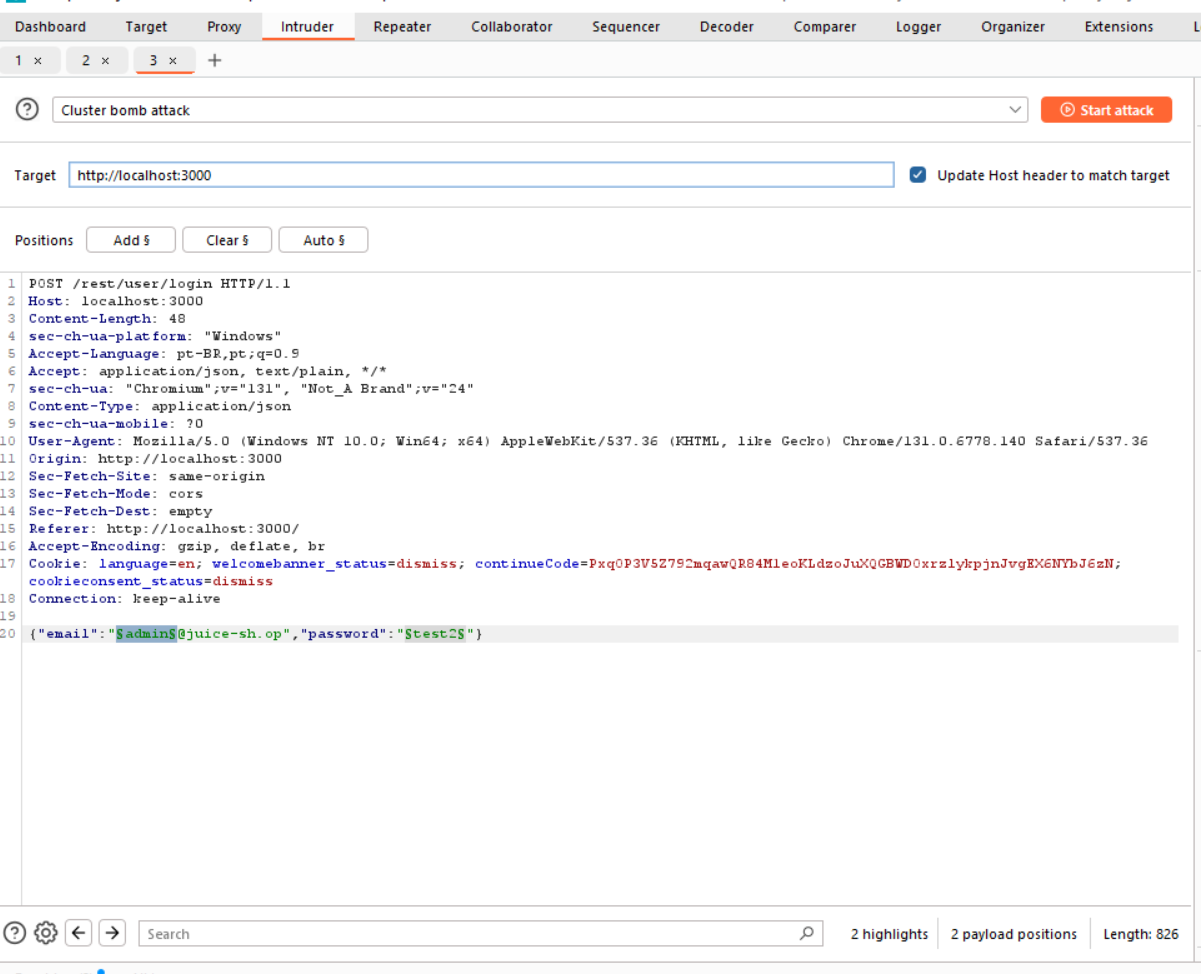
Evidências:

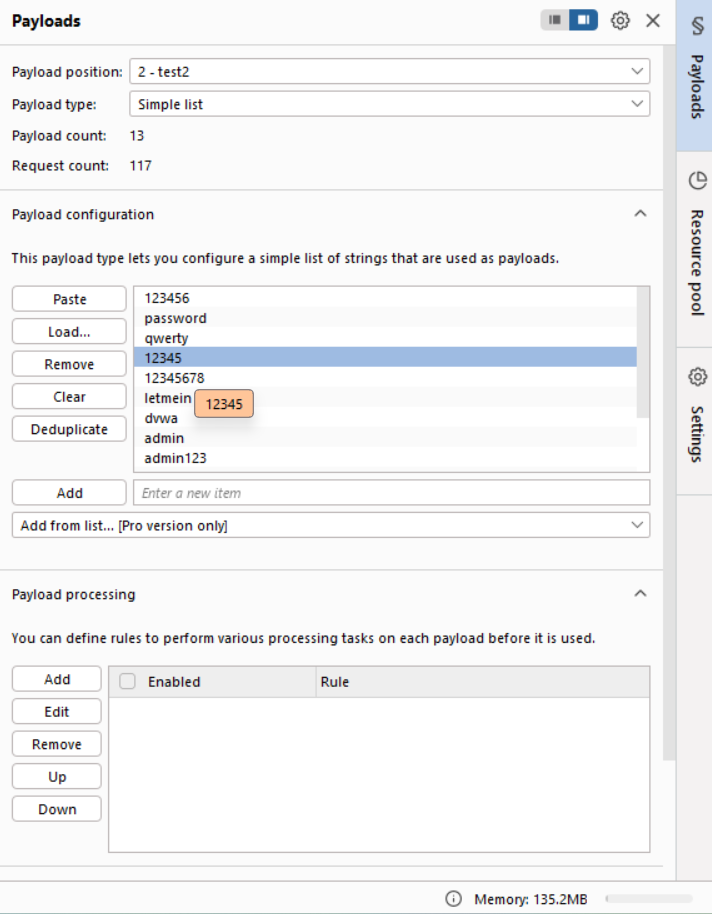
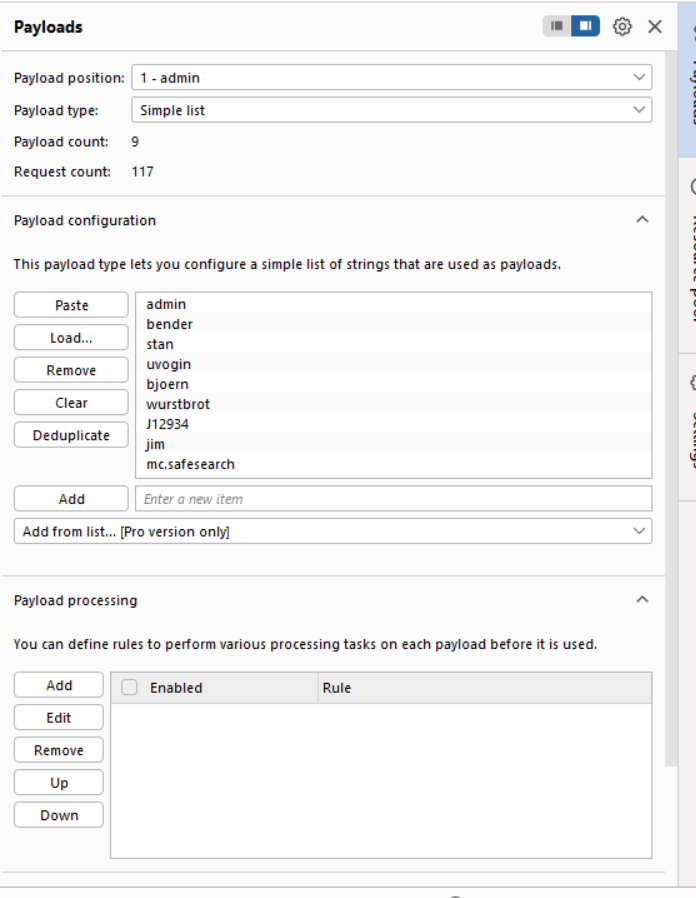
Na tela de login foi inserido o email do administrador com qualquer caracter no campo de senha

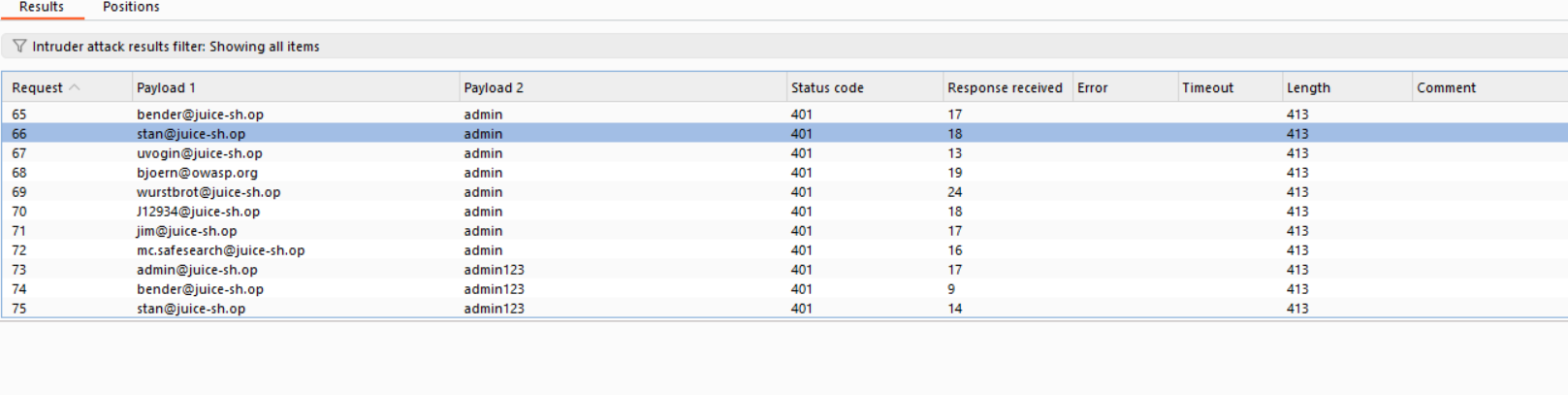


Foi realizado uma coleta de todos os usuários através dos comentários nos posts, e através do burp suite foi realizado um cluster bomb attack com os dados coletados e possíveis senhas

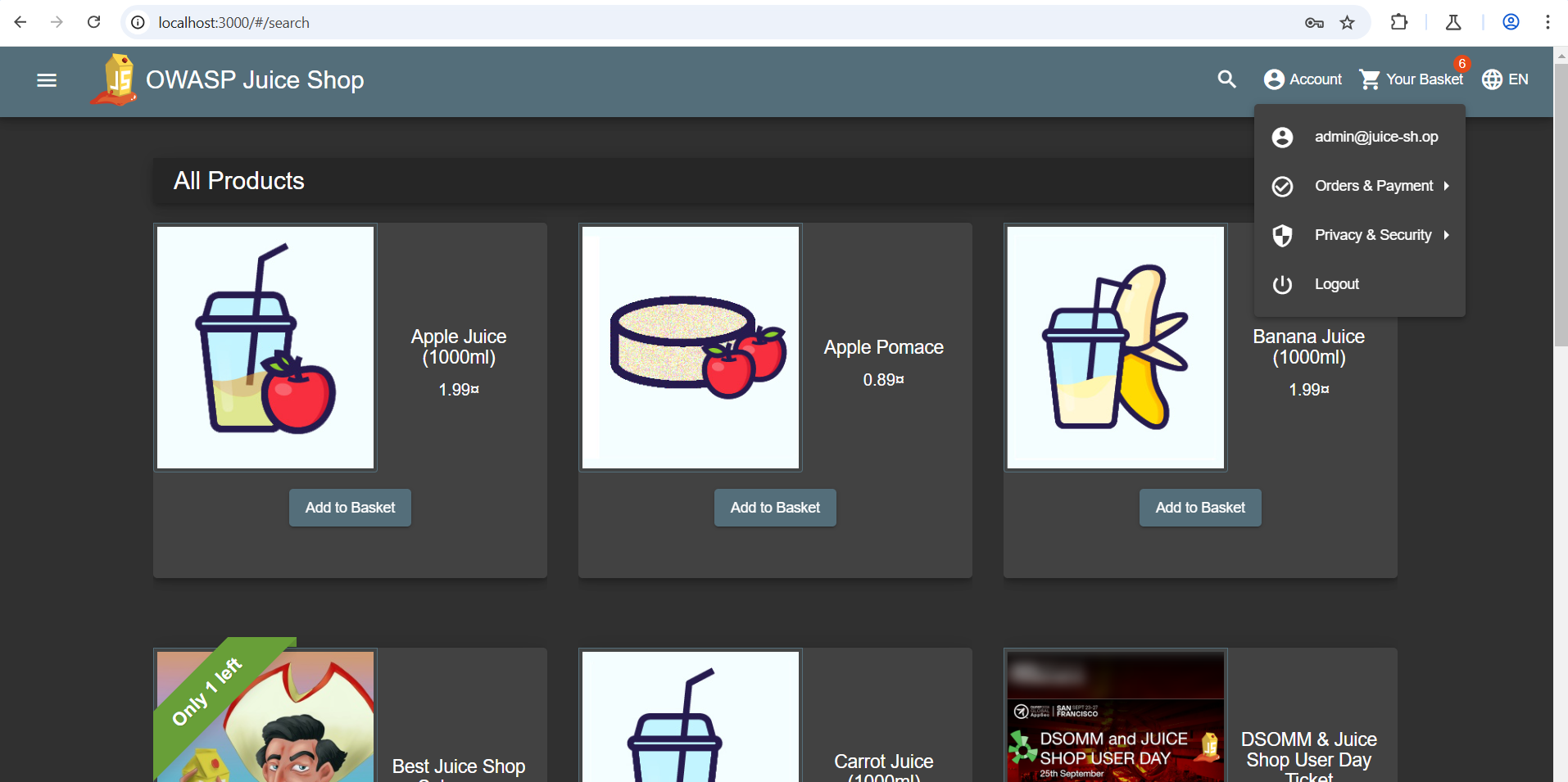
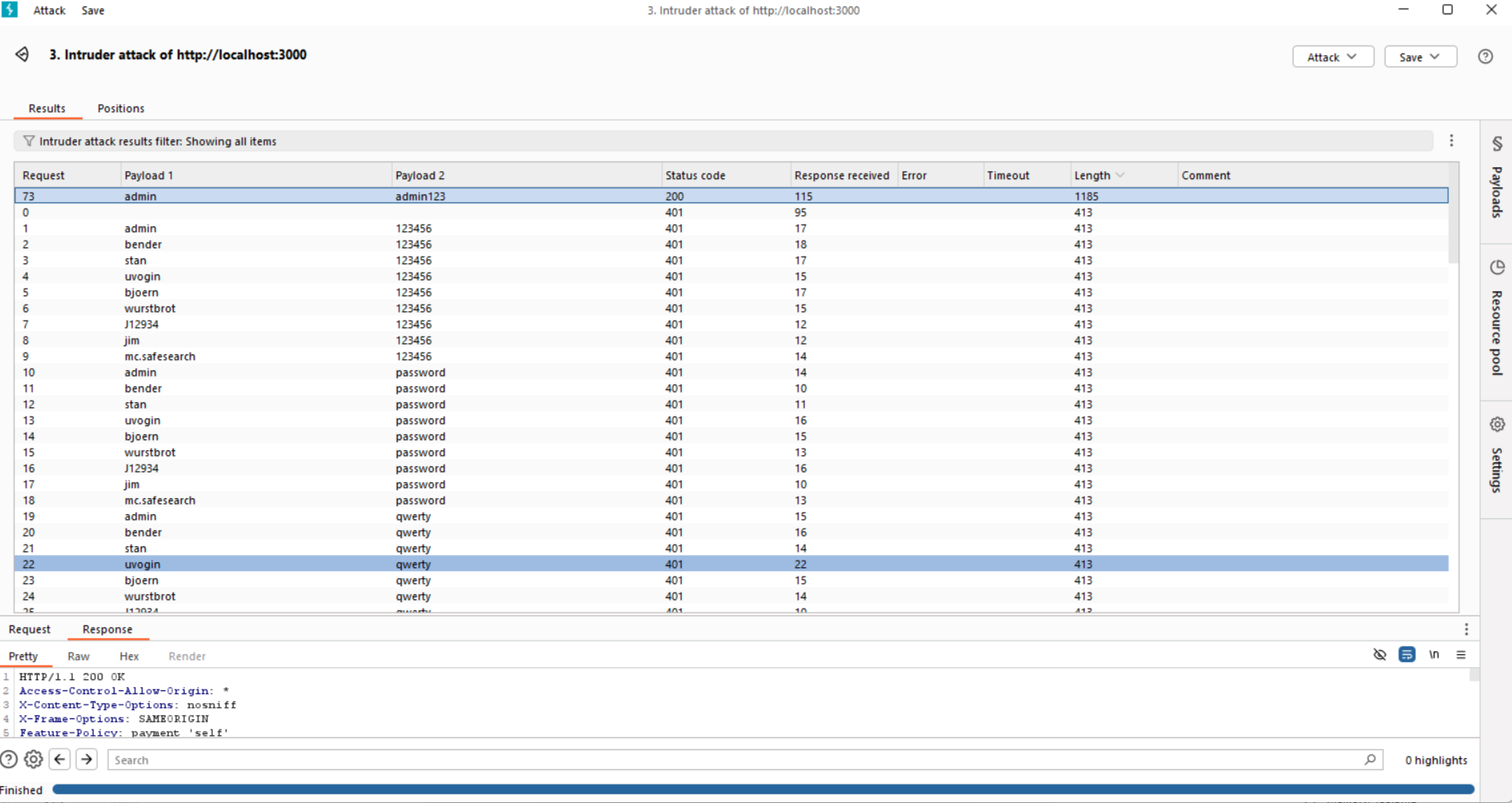








Ao analisar o attack, nota-se um o tamanho de um dos usuários, mostrando que a senha do mesmo é fraca e de fácil acesso;



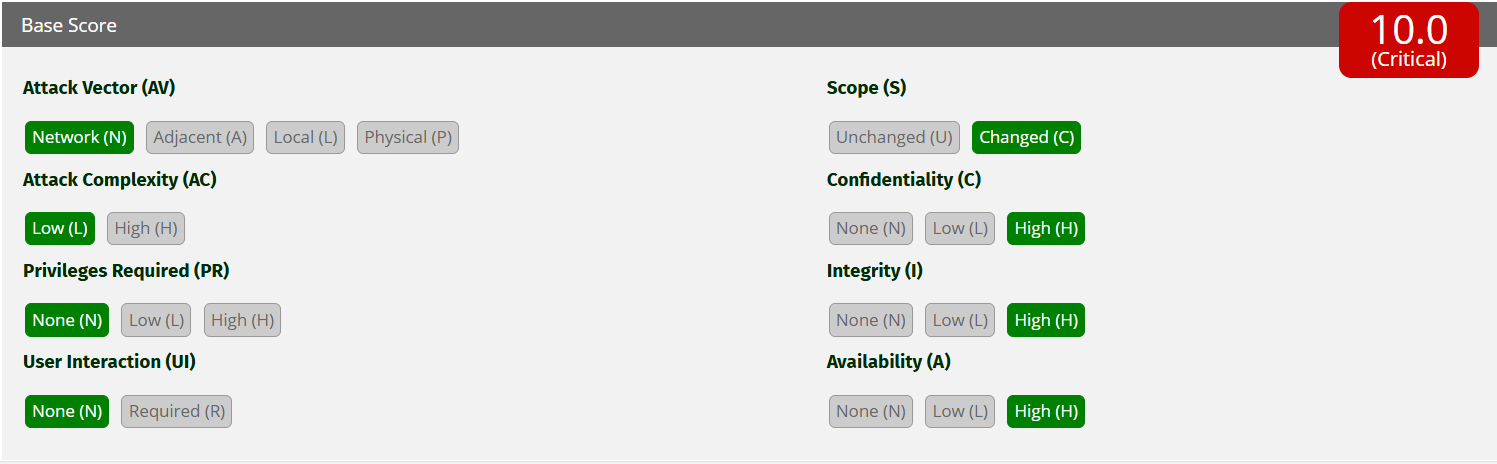
Recomendações:

* Usar autenticação forte: exigir aos usuários senhas complexas e seguras, adotar o uso de autenticação de múltiplo fator (MFA), criar limitação de uso de login, etc.;
* Usar protocolos de Autenticação seguros: implementar OAuth 2.0, utilizar tokens jwt e se a tentar a criptografia utilizada, etc.;
* Gerenciamento seguro de sessões: gerar IDs de sessão aleatórios e seguros, configurar expiração de sessão, revogar tokens e sessões de forma apropriada, etc.;
* Criptografar senhas de forma segura: armazenar as senhas com hashing e nunca em plaintext;
* Proteção contra ataque de força bruta: bloquear usuários que tiveram várias tentativas de login falhadas, pode ser temporariamente ou crie um atraso nas tentativas de login, utilize CAPTCHA nas telas de login, etc.;
* Prevenir session fixation: sempre que o usuário se autenticar, gere um novo Id de sessão para evitar ataques, etc.;
* Proteção contra o roubo de credenciais: sempre utilize HTTPS, evite tokens e senhas em URLs, etc.;
* Auditoria e monitoração de Acessos: utilize logs de autenticação, crie alertas de logins anormais.

SQL injection

Severidade (CVSS3.1): 3 (Alta)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H



CWE:

CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection');

Localização:

* <http://localhost:3000/#/login>

Descrição:

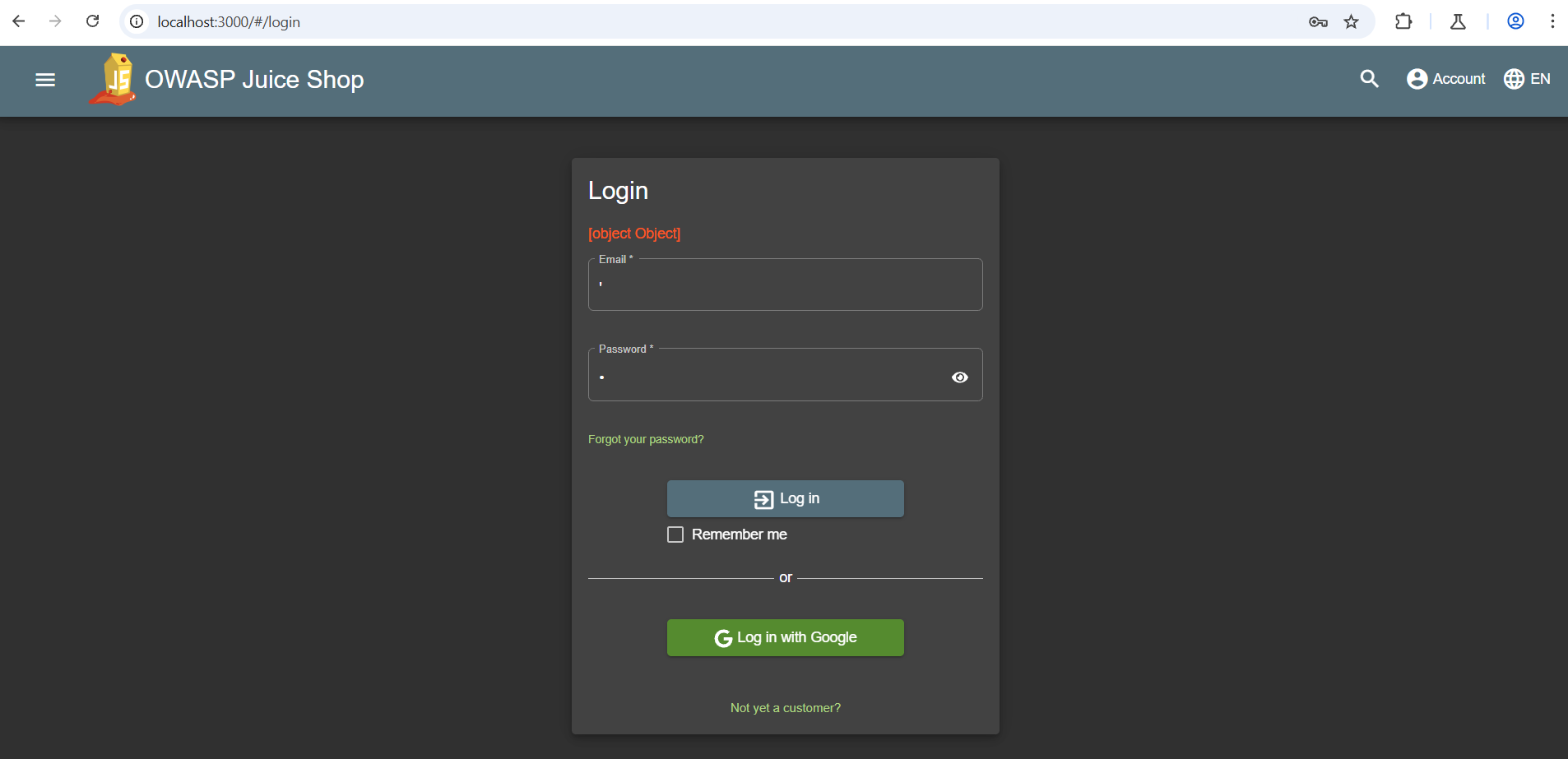
O SQL Injection (SQLi) é uma técnica de injeção de código que explora uma vulnerabilidade de segurança que ocorre na camada de banco de dados de um aplicativo.

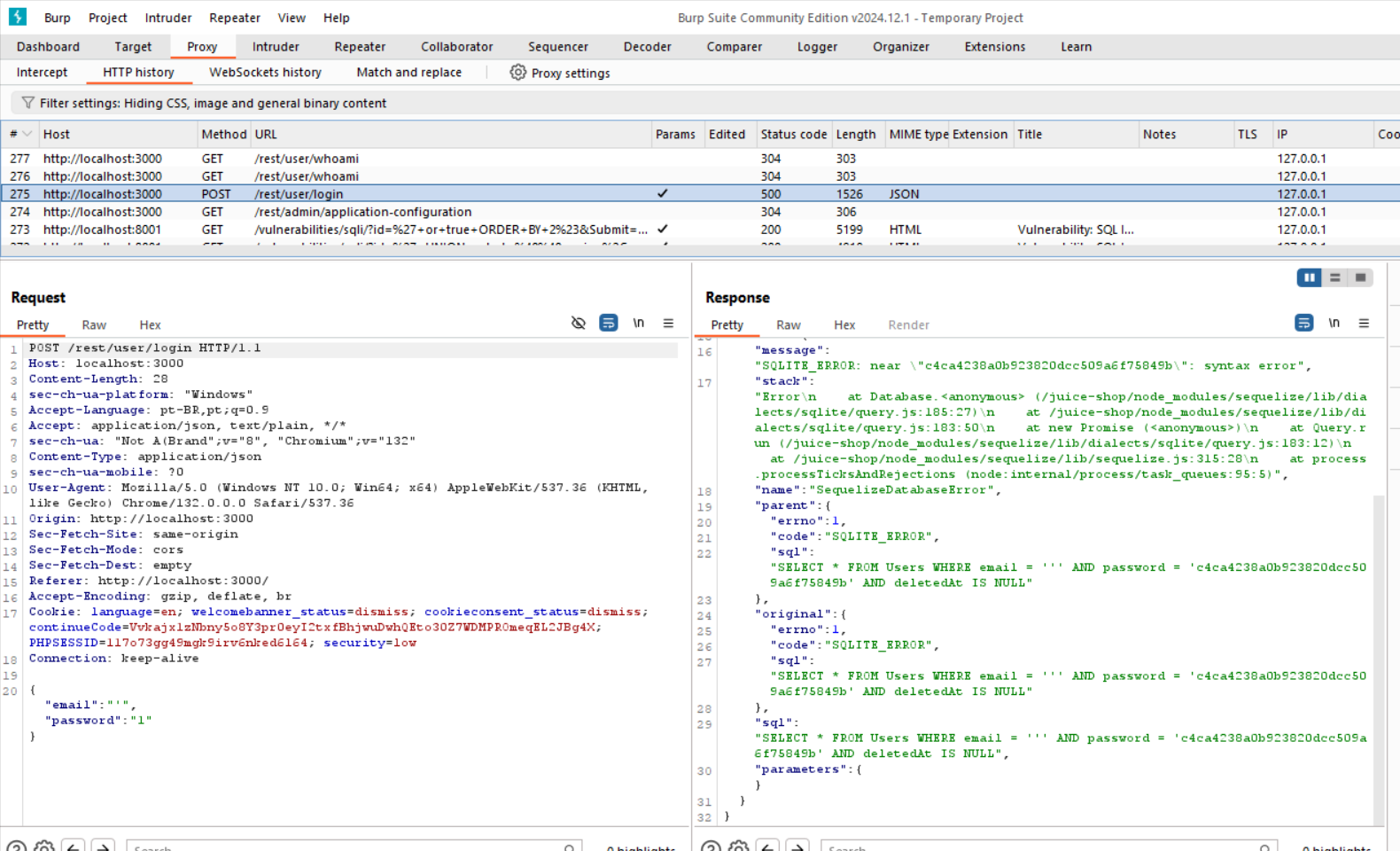
Foi realizado um ataque breve no Juice Shop utilizando o método *repeat*, enviando o seguinte comando SQL no campo de e-mail: ' OR 1 = 1 --. Ao receber um response com o código de status 200 (OK), o mesmo teste foi repetido na tela de login do site, conforme evidências anexadas.

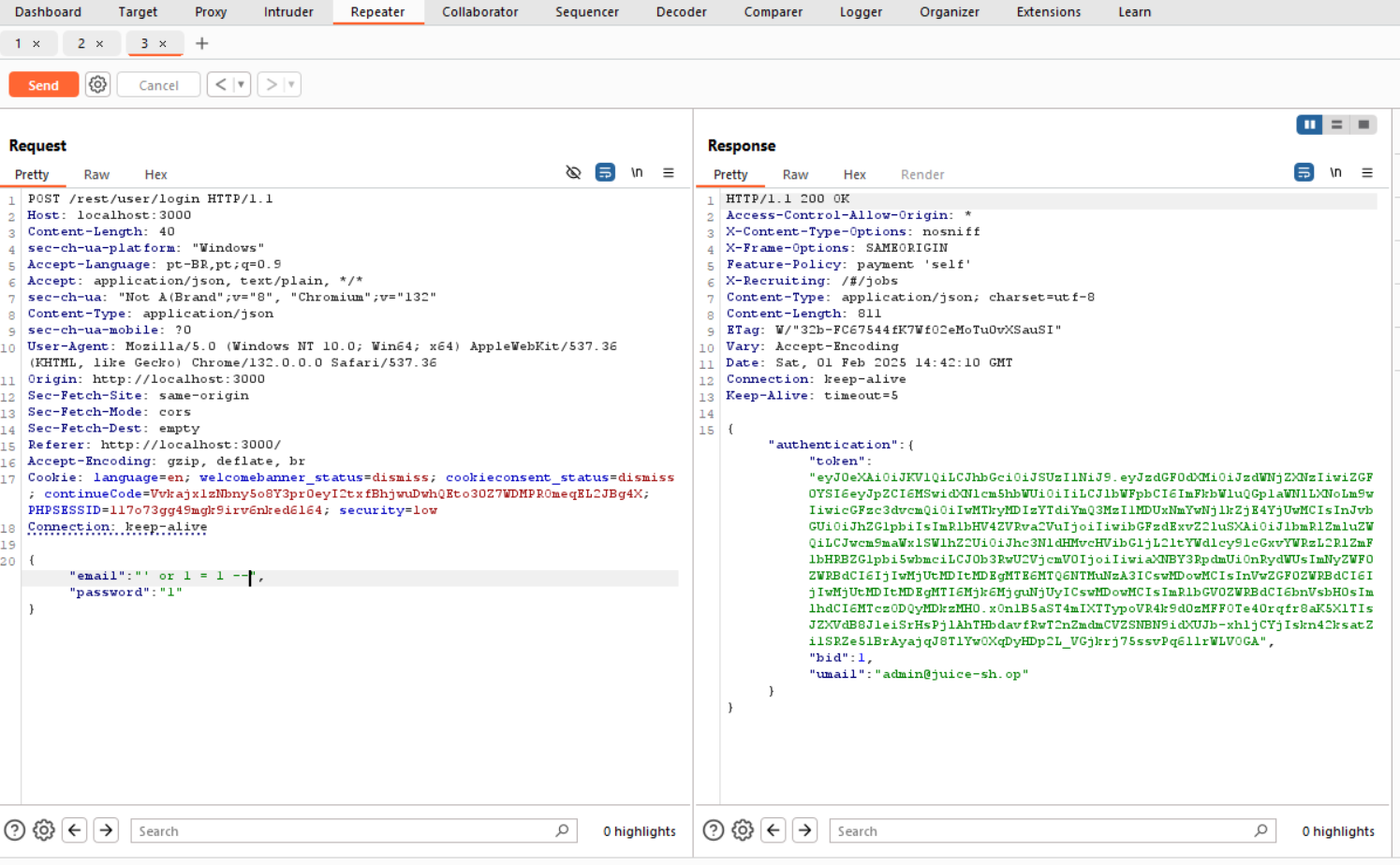
Obs.: Um ataque de SQL Injection pode ter diversas consequências, afetando áreas críticas como confidencialidade, autenticação, controle de acesso e integridade dos dados. Isso ocorre porque o atacante consegue enviar comandos diretamente ao servidor, o que pode permitir desde o acesso de administrador até a obtenção de informações sensíveis, como a versão do banco de dados e outros dados confidenciais.

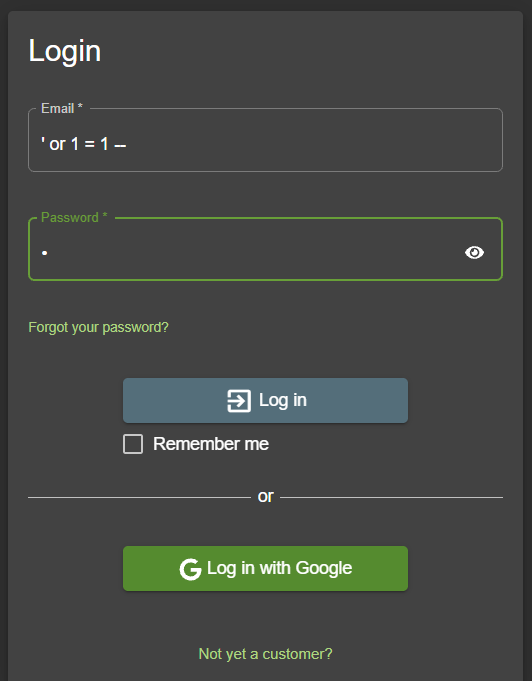
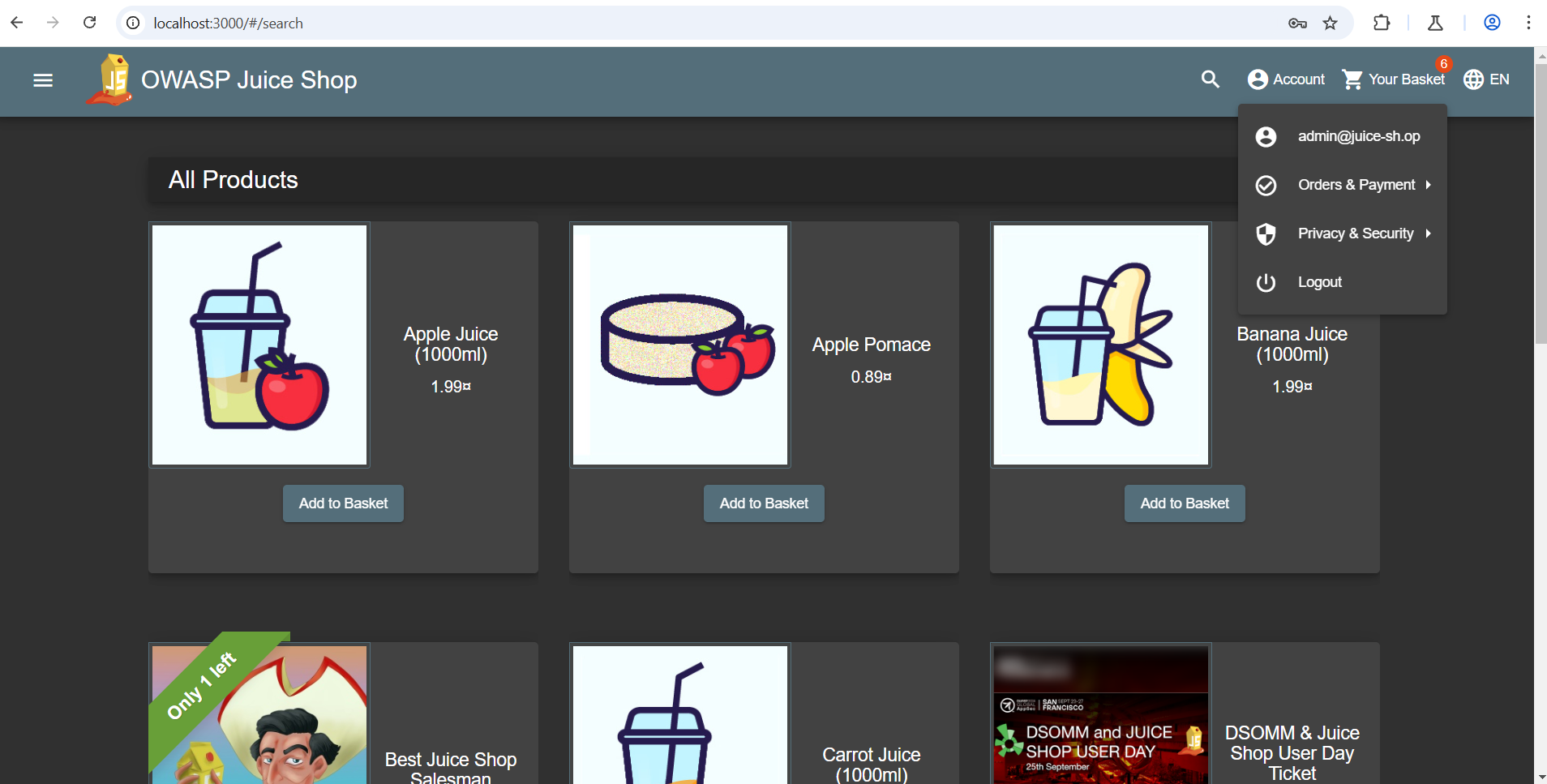
Evidências:

Inserido aspas simples no campo de email e um caracter qualquer para ser capturado pela ferramenta Burp Suite



Observa-se que no response temos algumas informações de SELECT que podem ser usadas:

Inserido o comando ' OR 1 = 1 – para validar a eficácia do ataque

Comando SQL injetado na tela de login através da URL <http://localhost:3000/#/login>  

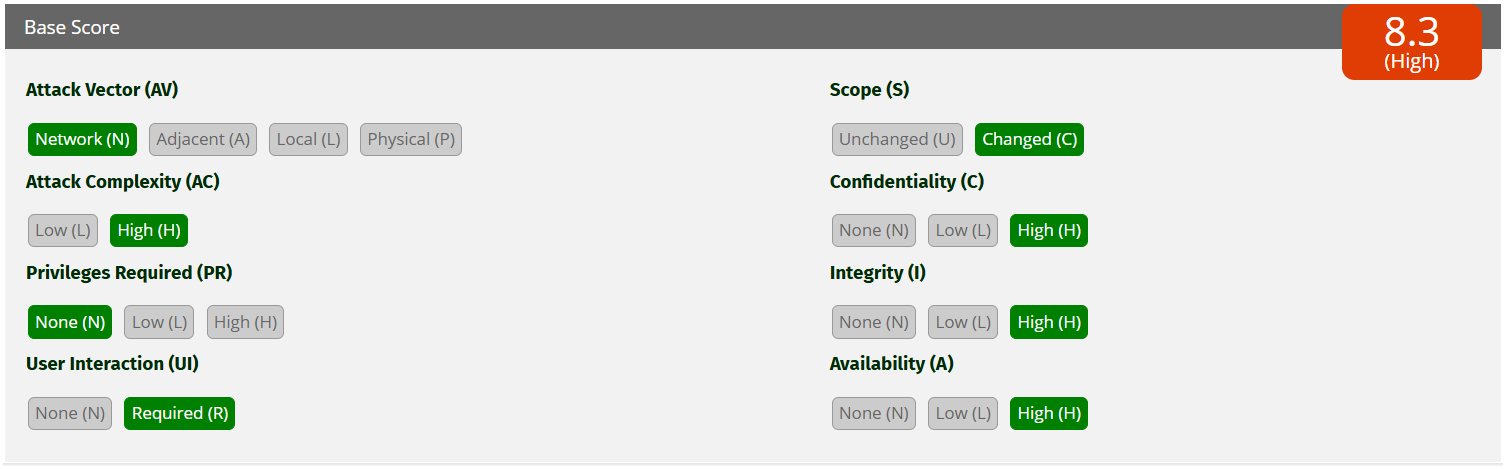
Recomendações:

* Utilizar bibliotecas e frameworks;
* Validação e higienização de entradas: uso de whitelists, validação dos campos fornecidos, converter caracteres como aspas, apóstrofos e outros símbolos que podem ser interpretados como parte de consulta sql;
* Utilização de Prepared Statements ou Queries Parametrizadas: evitar concatenação direta de strings, tratar os dados fornecidos pelo usuário como parâmetros, impedindo que sejam interpretados como comandos SQL.
* Princípio do menor privilégio: limitar os privilégios de acesso, utilizar diferentes usuários para diferentes tarefas, etc.
* Tratamento adequado de erros: Não exibir informações detalhadas sobre o banco de dados, registrar os erros em logs, etc;
* Utilização de ferramentas de proteção: Utilize Firewall de Aplicações Web (WAF), ferramentas de análise de código;
* Boas práticas de desenvolvimento: Realize teste de segurança, mantenha o sistema operacional atualizado, assim como servidores web, banco de dados e libs, etc.

**Forgot Password**

**Severidade (CVSS3.1):** 3 (Alta)

CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:H/A:H



**CWE:**

CWE-287: Improper Authentication;

CWE-522: Insufficiently Protected Credentials

CWE-640: Weak Password Recovery Mechanism for Forgotten Password

**Localização:**

* <http://localhost:3000/#/forgot-password>
* /rest/user/security-question?email=[bjoern@owasp.org](mailto:bjoern@owasp.org)
* <http://localhost:3000/#/photo-wall>
* http://localhost:3000/#/search

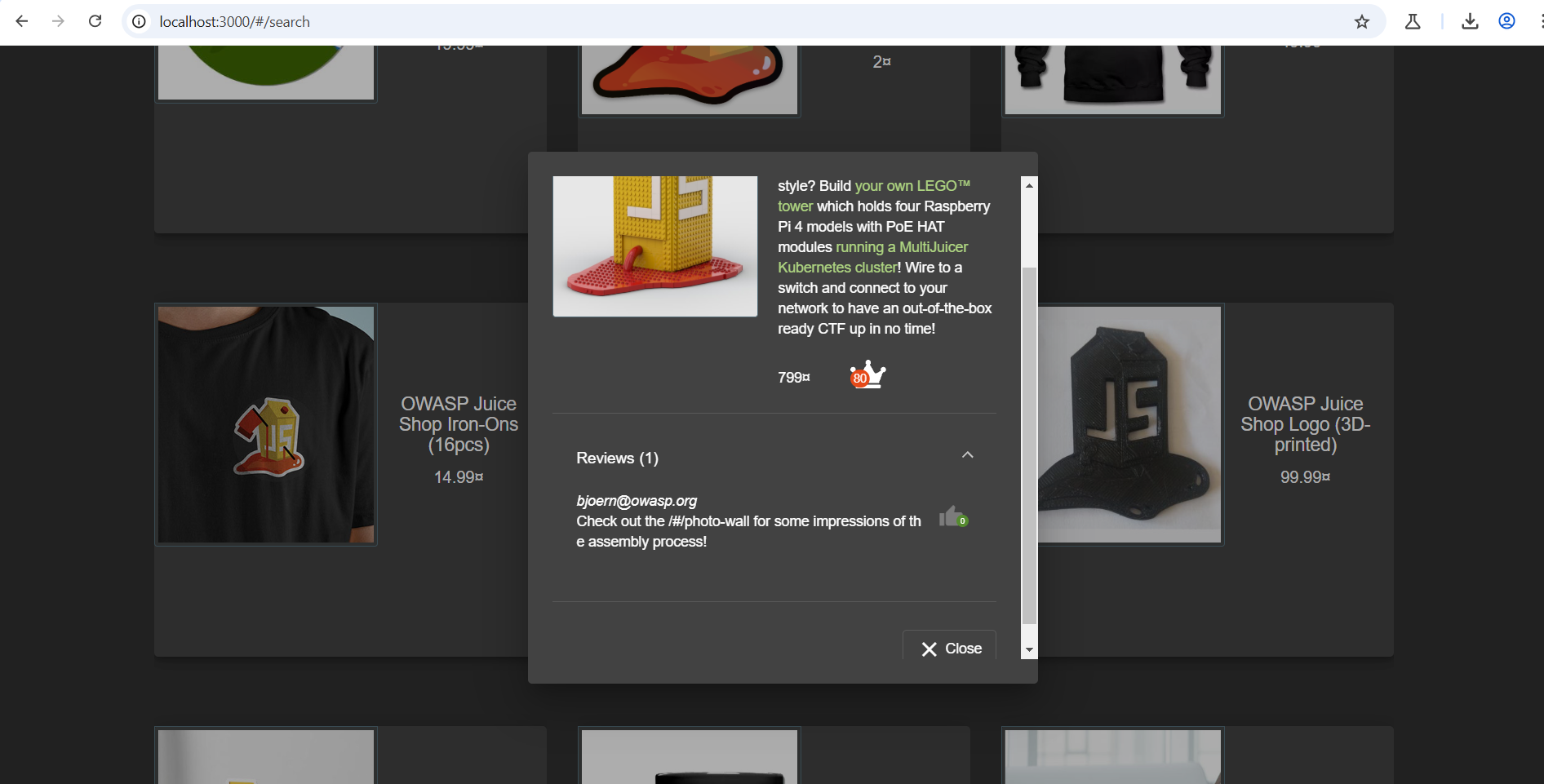
**Descrição:**

A funcionalidade "Esqueceu a Senha" é um recurso essencial para a conveniência do usuário, mas também pode ser um ponto de entrada para ataques se não for implementada com segurança.

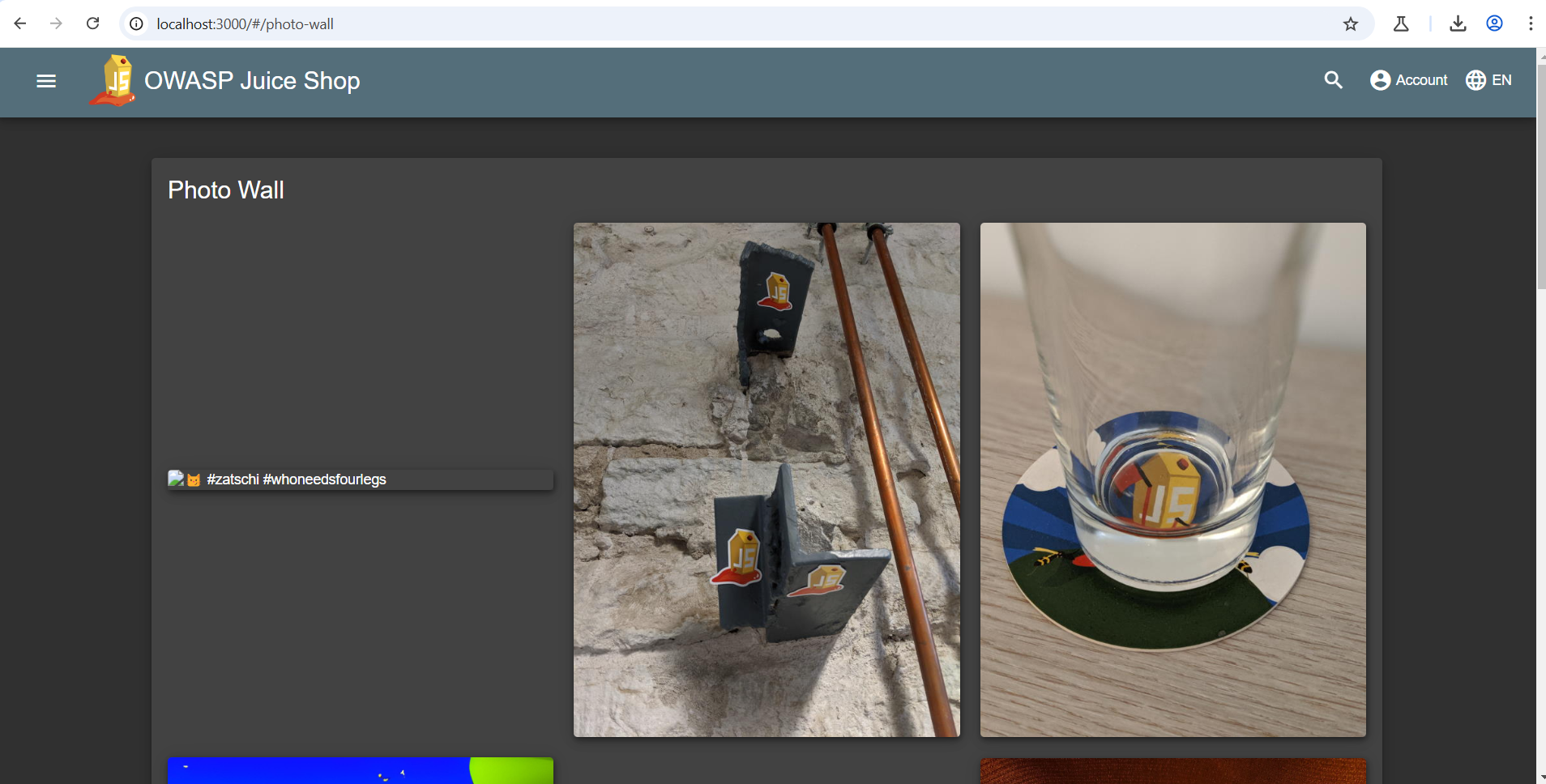
O ataque foi realizado por meio de engenharia social, com a necessidade de investigar detalhes da vida da vítima para identificar possíveis respostas às perguntas de recuperação de senha. A partir de comentários encontrados na URL (<http://localhost:3000/#/photo-wall>), foi possível descobrir que o apelido do gato da vítima é "Zatschi". No entanto, essa informação não foi suficiente para realizar a alteração da senha. Foi então necessário acessar o Twitter da vítima, onde ela menciona que o nome real do gato é "Zaya". Com essas informações, foi possível responder à pergunta de segurança e realizar o reset da senha com sucesso.

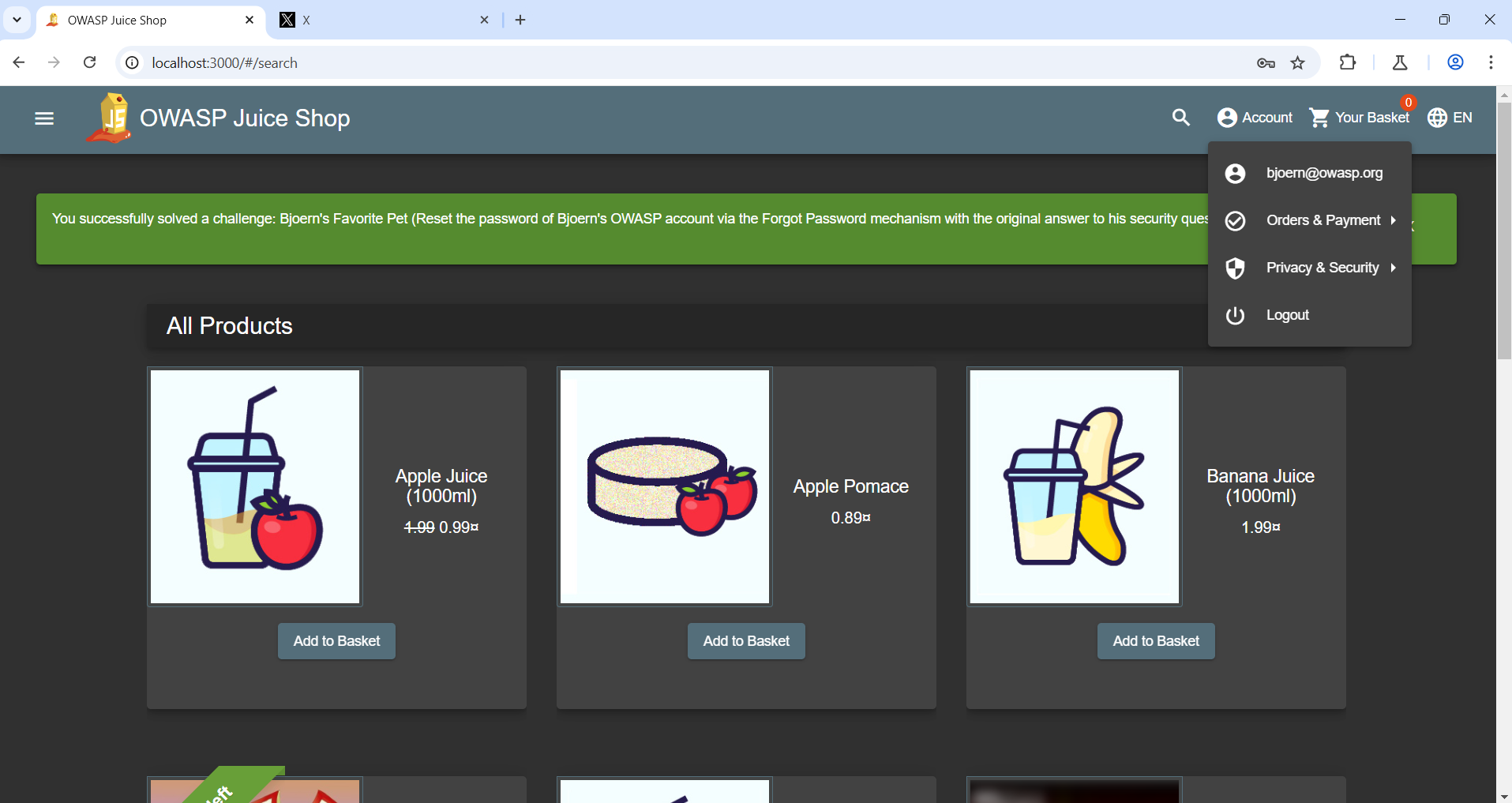
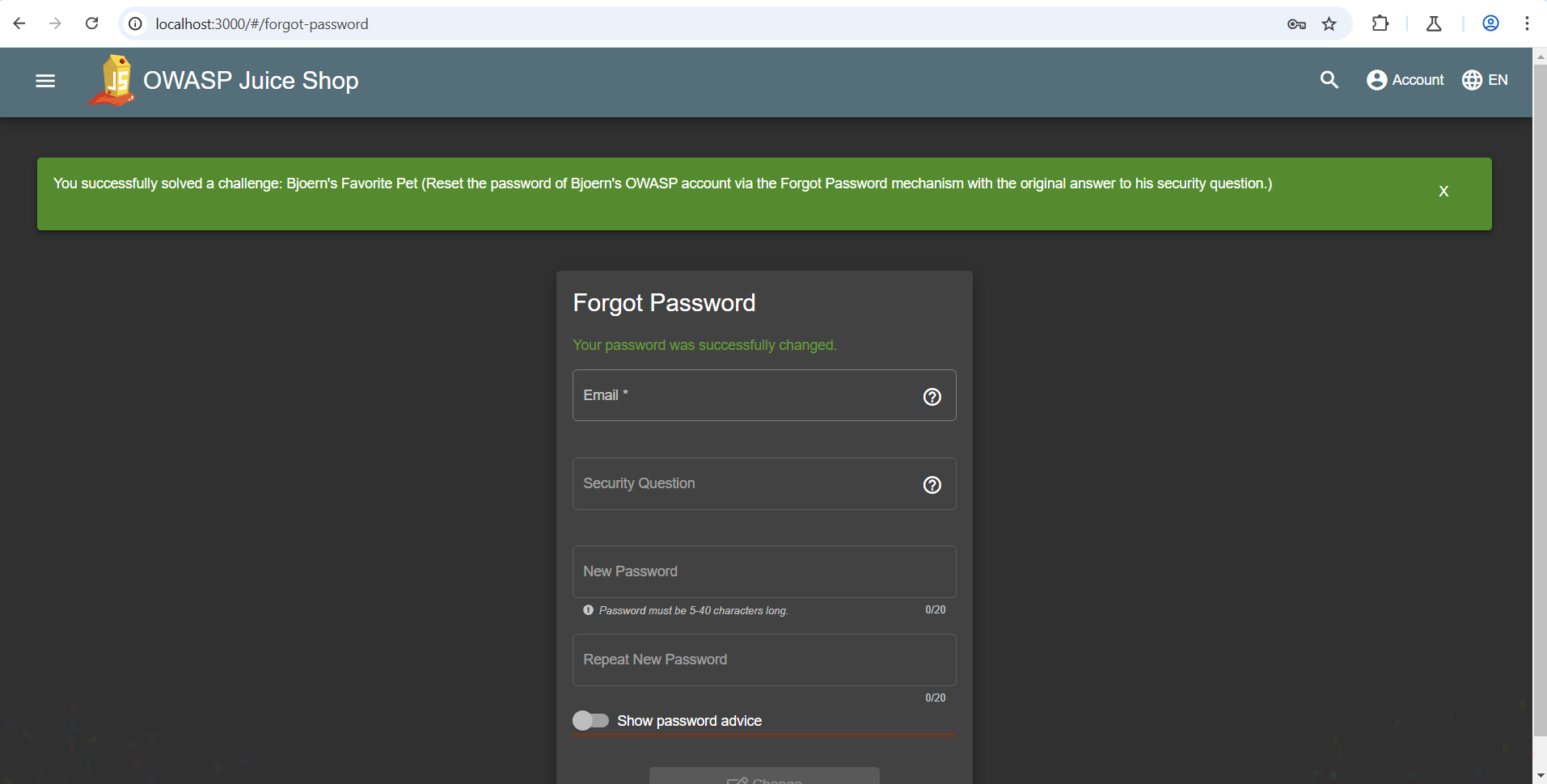
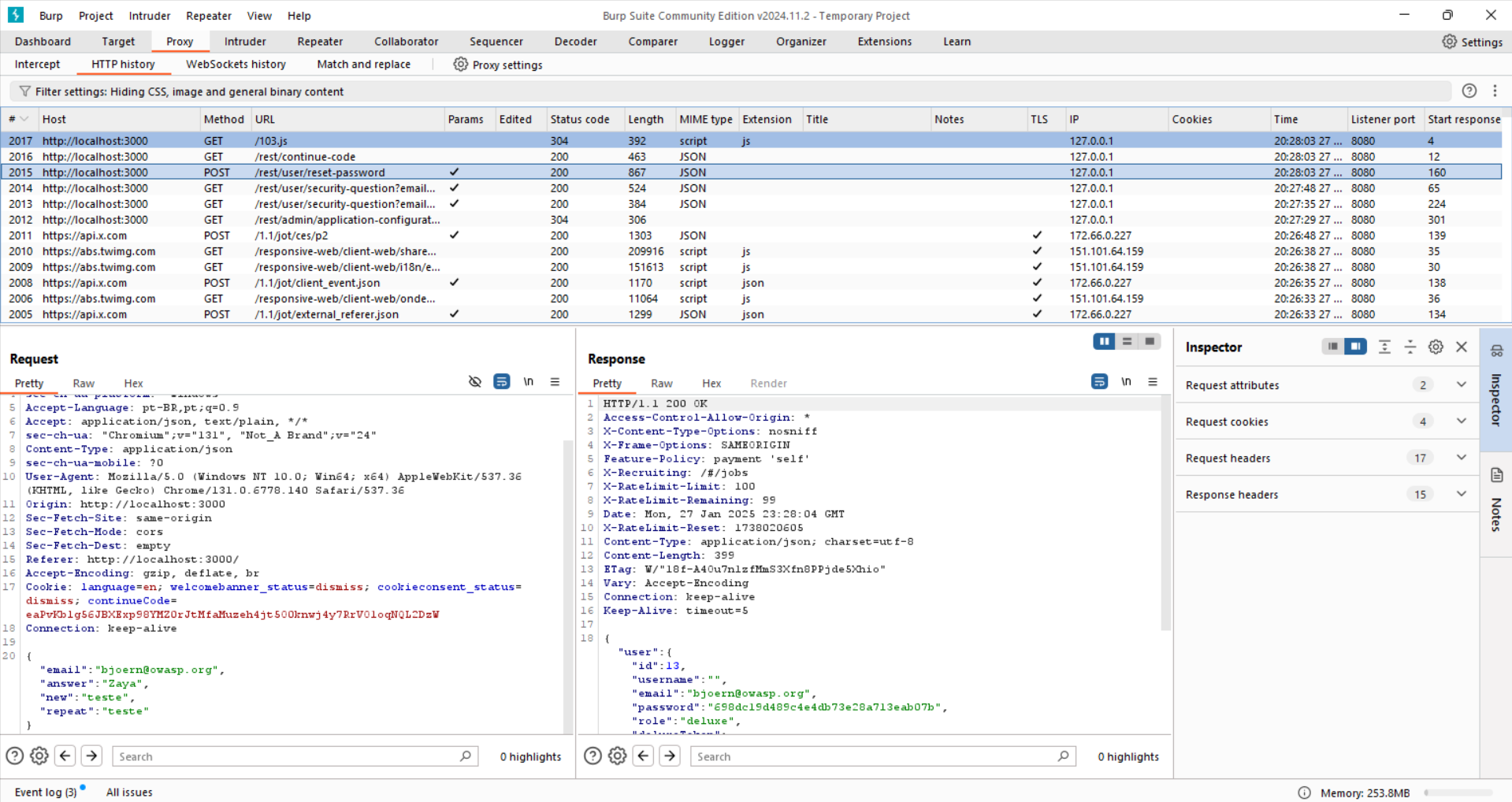
As consequências desse tipo de ataque para o usuário podem incluir roubo de identidade, acesso não autorizado a contas, perda de dados, danos à reputação, prejuízos financeiros e impactos emocionais. Para a organização, as repercussões podem ser igualmente graves, como a violação de dados e exposição de informações confidenciais, danos à reputação, perdas financeiras, interrupção de serviços devido a ataques e possíveis implicações legais.

Evidências:

Foi realizada uma análise dos comentários de todos os usuários, e um deles se destacou. Em um comentário, o usuário solicita o acesso a uma página específica (<http://localhost:3000/#/photo-wall>).

Ao acessar a página (<http://localhost:3000/#/photo-wall>), foi observada uma foto quebrada, mas com o nome ou apelido de um gato. A partir disso, foi iniciada uma análise de engenharia social para determinar se o nome real do gato era "Zatschi". Após investigar o perfil no Twitter da vítima, foi identificado que o nome do gato era, na verdade, "Zaya". Com essa informação, a requisição foi enviada para o "repeat" no Burp Suite, e a partir daí, foi realizada a troca de senha do usuário "bjoern".





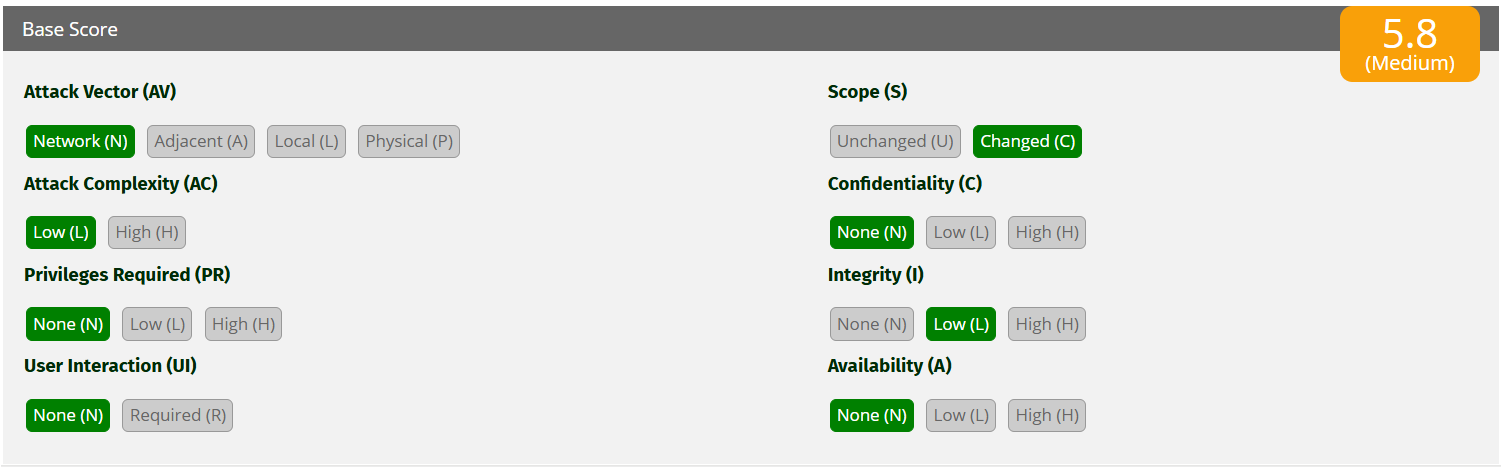
**Recomendações:**

* Crie senhas fortes e exclusivas;
* use um gerenciador de senhas;
* Habilite a autenticação de dois fatores (2FA);
* Desconfie de e-mails e mensagens suspeitas;
* Mantenha seus softwares atualizados como sistemas operacionais, navegadores e aplicativos;
* Evite redes Wi-Fi públicas;
* Monitore suas contas regularmente;
* Evite senhas de segurança muito óbvias como nomes de familiares, datas de aniversários, nomes de pets, etc.

**Broken Access Control (Forged Feedback)**

**Severidade (CVSS3.1):**  2 (Média)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:N



**CWE:**

CWE-284: Improper Access Control

CWE-862: Missing Authorization

CWE-20: Improper Input Validation

CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

**Localização:**

* <http://localhost:3000/#/contact>
* <http://localhost:3000/api/Feedbacks/>
* <http://localhost:3000/#/about>

**Descrição:**

A manipulação de estrelas em feedbacks pode estar associada a uma vulnerabilidade de controle inadequado sobre os dados inseridos pelo usuário, que pode resultar em comportamento inesperado, como manipulação indevida de classificações, como estrelas, em sistemas de avaliação.

Na página<http://localhost:3000/#/contact>, foi inserido um feedback com um CAPTCHA incorreto, com o objetivo de identificar a página que realmente envia o feedback. Após isso, a requisição foi capturada pelo 'Repeat' no Burp Suite e manipulada com os seguintes dados:

{

"captchaId": 2,

"captcha": "13",

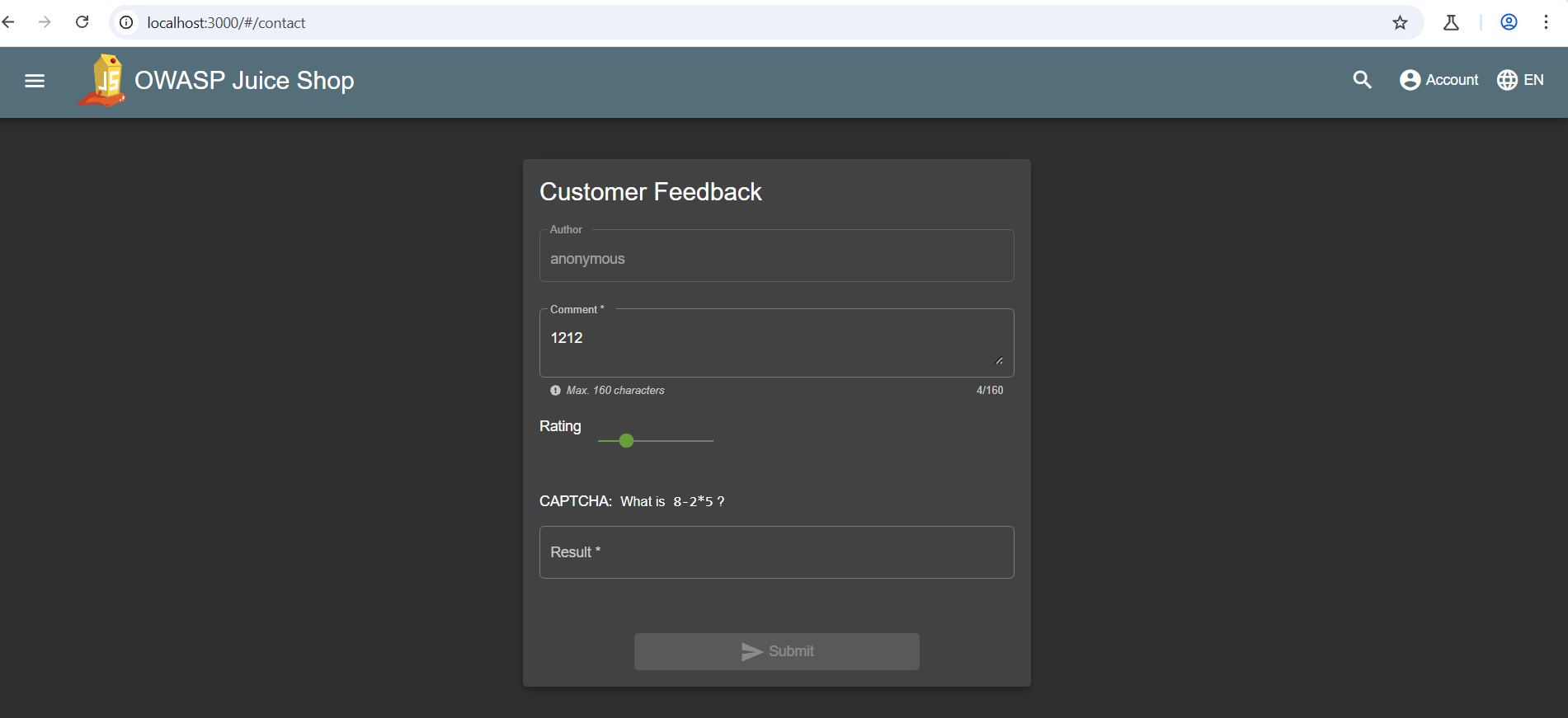
"comment": "teste de ady (\*\*yellen@juice-sh.op)",

"rating": 0

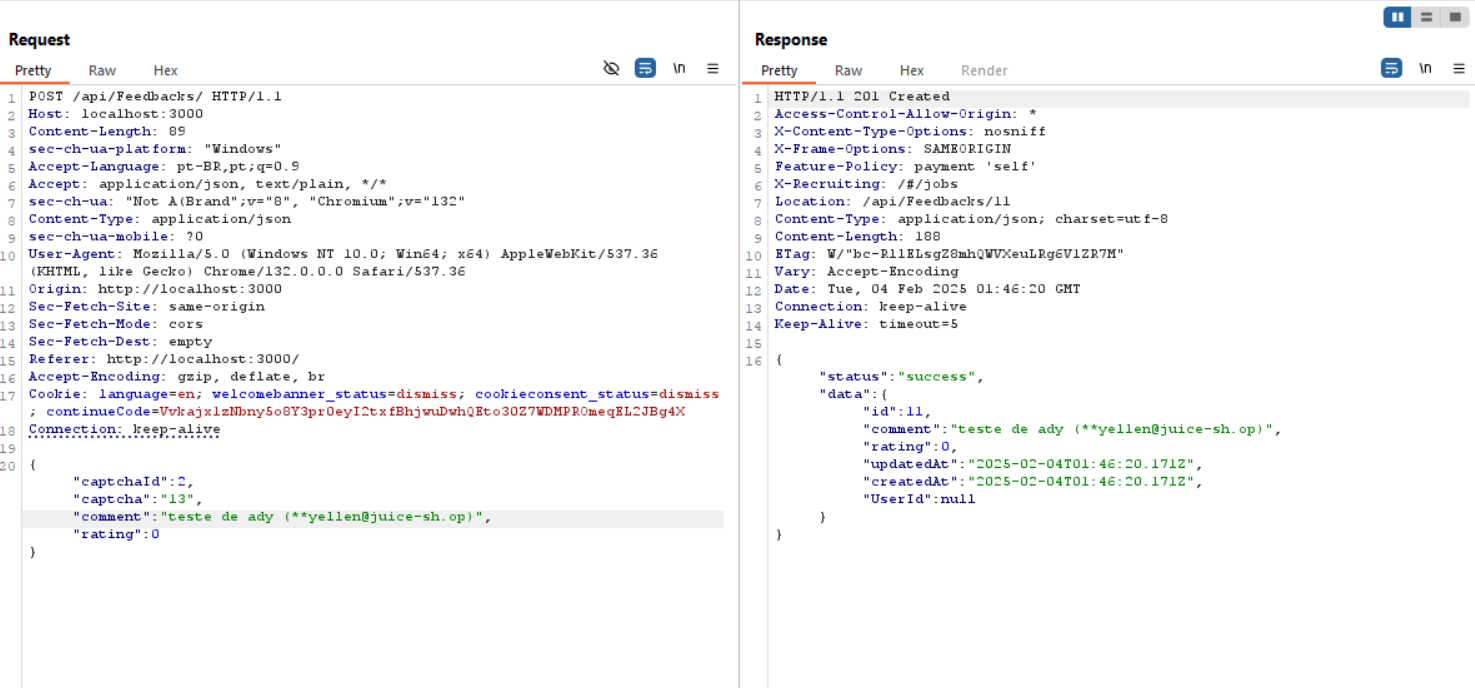
}

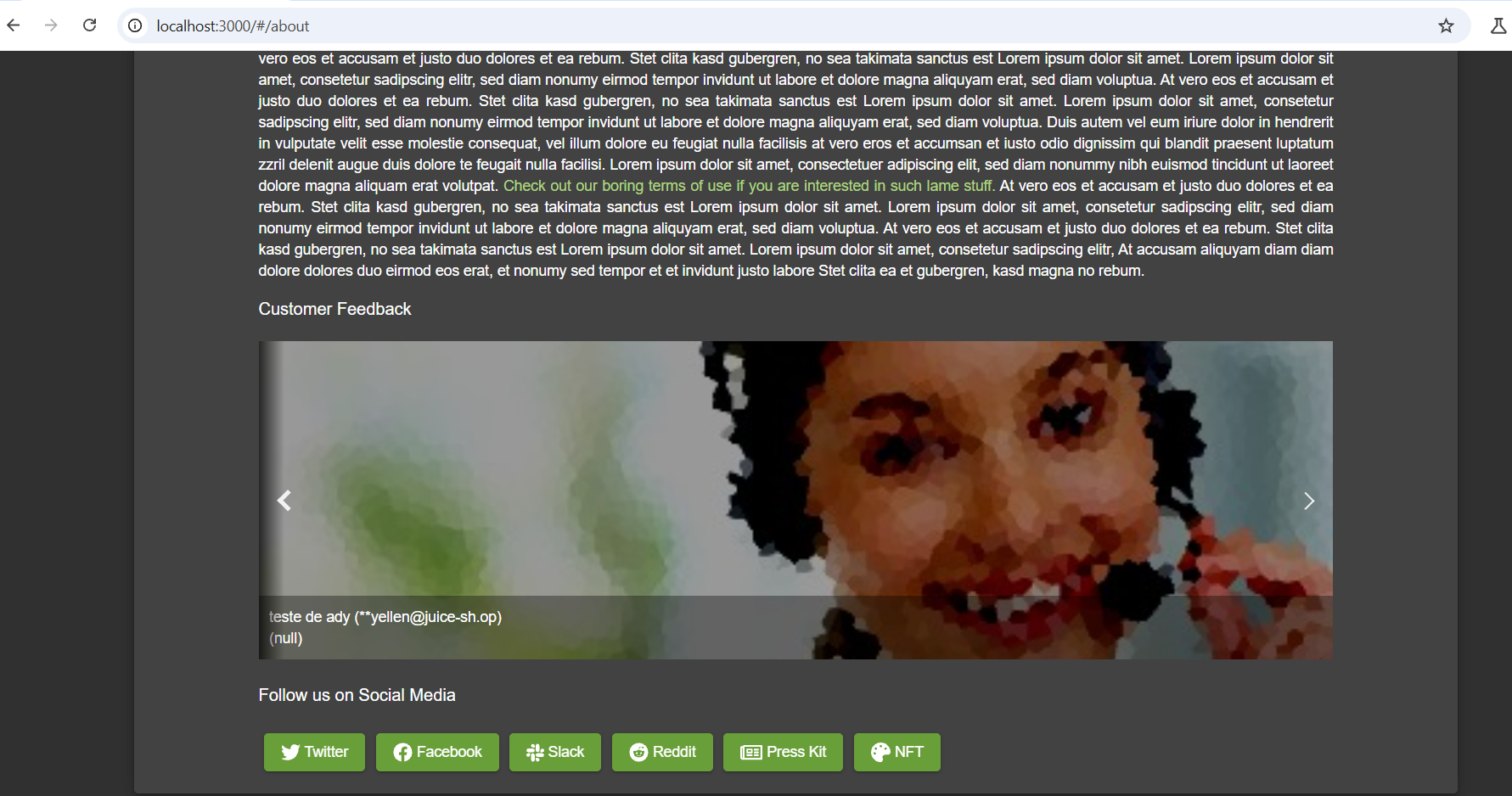
As consequências da manipulação de feedbacks, como a alteração de estrelas em avaliações, pode prejudicar gravemente a reputação e a confiança da organização, além de resultar em perda de clientes e receita devido à insatisfação e frustração geradas por avaliações falsas. Os impactos também podem incluir consequências legais e financeiras, como multas e processos, corrupção de dados e análises, e custos elevados com o monitoramento necessário para mitigar o problema.

**Evidências:**



No Burp Suite, foi possível observar que a requisição foi manipulada com uma avaliação de 0 estrelas e um comentário. Ao ser enviada, mesmo com o CAPTCHA igual ao da requisição inicial, obteve-se o código de resposta 200 OK. Vale destacar que há duas vulnerabilidades identificadas: a falha no CAPTCHA e a possibilidade de inserir 0 estrelas nas avaliações, como demonstrado no teste realizado no Burp Suite.





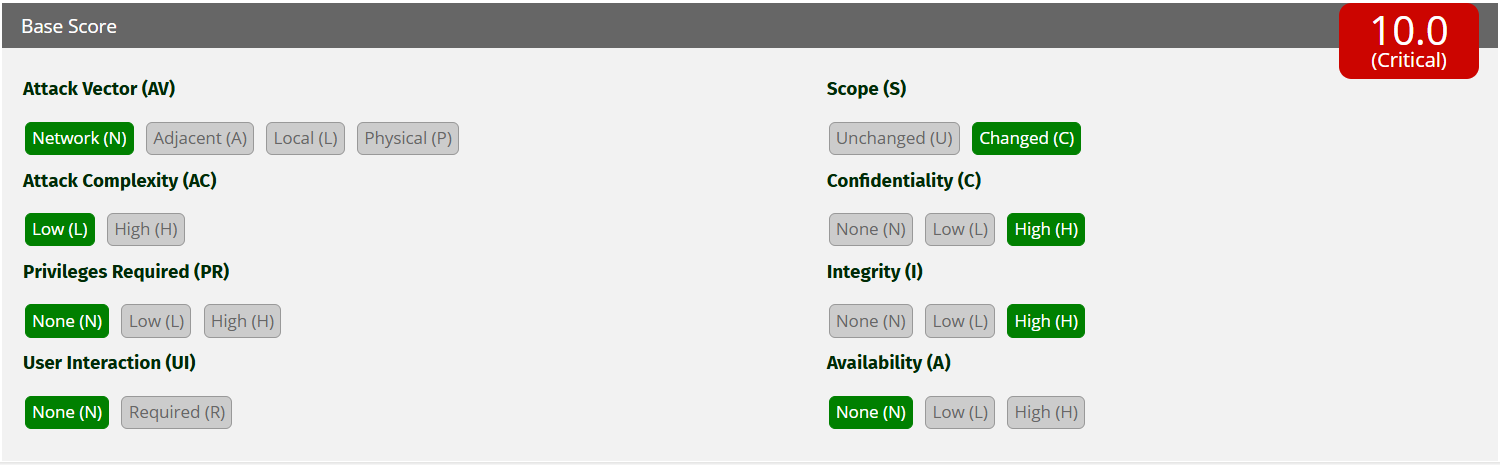
**Recomendações:**

* **Validação e sanitização de entradas** para impedir manipulações de dados maliciosos.
* **Autenticação e controle de acesso** rigorosos para garantir que apenas usuários autorizados possam enviar e modificar feedbacks.
* **Monitoramento de atividades suspeitas** para detectar manipulações de forma rápida.
* **Verificação de identidade** dos usuários para evitar manipulação por bots ou contas falsas.
* **Proteção de banco de dados** e uso de práticas seguras para garantir a integridade dos dados

**DOM-based XSS (Cross-Site Scripting)**

**Severidade (CVSS3.1):** 3 (Alta)l

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N**



**CWE:**

CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

**Localização:**

* <http://localhost:3000/#/search>
* <http://localhost:3000/#/search?q=%3Ciframe%20src%3D%22javascript:alert(%60xss%60)%22%3E>

**Descrição:**

O DOM-based XSS ocorre quando o código JavaScript em uma página web processa dados fornecidos pelo usuário (como entradas de formulário, parâmetros de URL, etc.) sem sanitização adequada. Esses dados podem ser manipulados pelo atacante para injetar scripts maliciosos que, uma vez executados no navegador da vítima, podem causar a execução de comandos indesejados.

Na URL [**http://localhost:3000/#/search**](http://localhost:3000/#/search), foi inserido um script JavaScript malicioso no campo de pesquisa (**search**), com o seguinte código:

<iframe src="javascript:alert('xss')">.

Quando a página é carregada, o script malicioso é executado no navegador da vítima. Isso pode resultar no roubo de informações confidenciais, modificação da interface do usuário ou a execução de outras ações maliciosas, comprometendo a segurança da aplicação e a privacidade do usuário.

As consequências de um DOM XSS são:

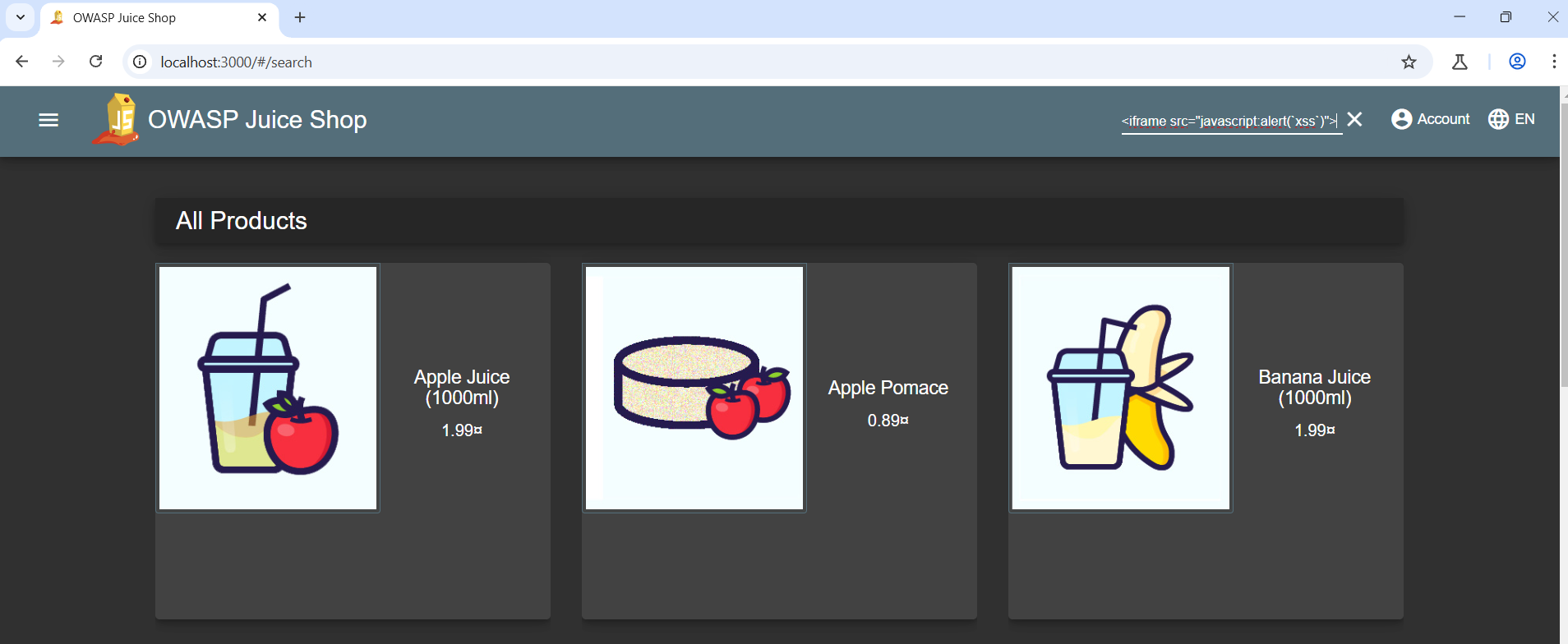
Roubo de informações confidenciais (cookies, credenciais, dados pessoais)

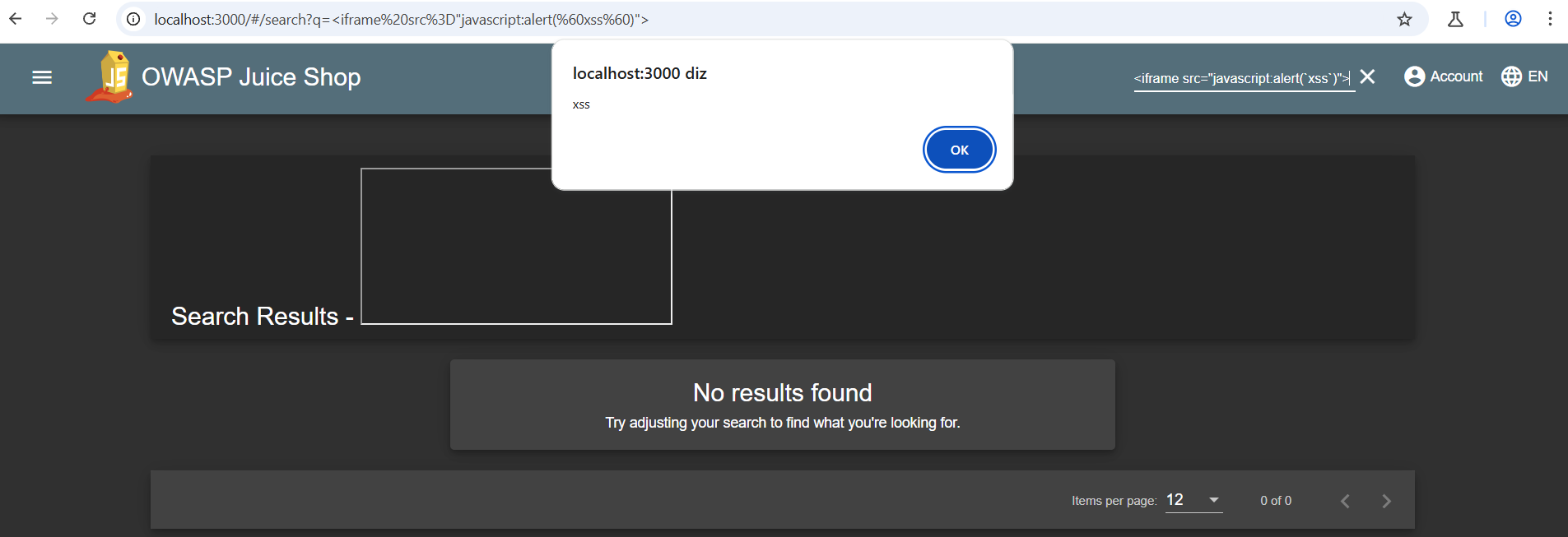
* Execução de ações não autorizadas em nome da vítima;
* Phishing e manipulação de conteúdo;
* Desfiguração do site;
* Distribuição de malware;
* Perda de reputação e confiança;
* Consequências legais e financeiras (multas, processos);
* Comprometimento de sistemas de autenticação e escalada de privilégios;

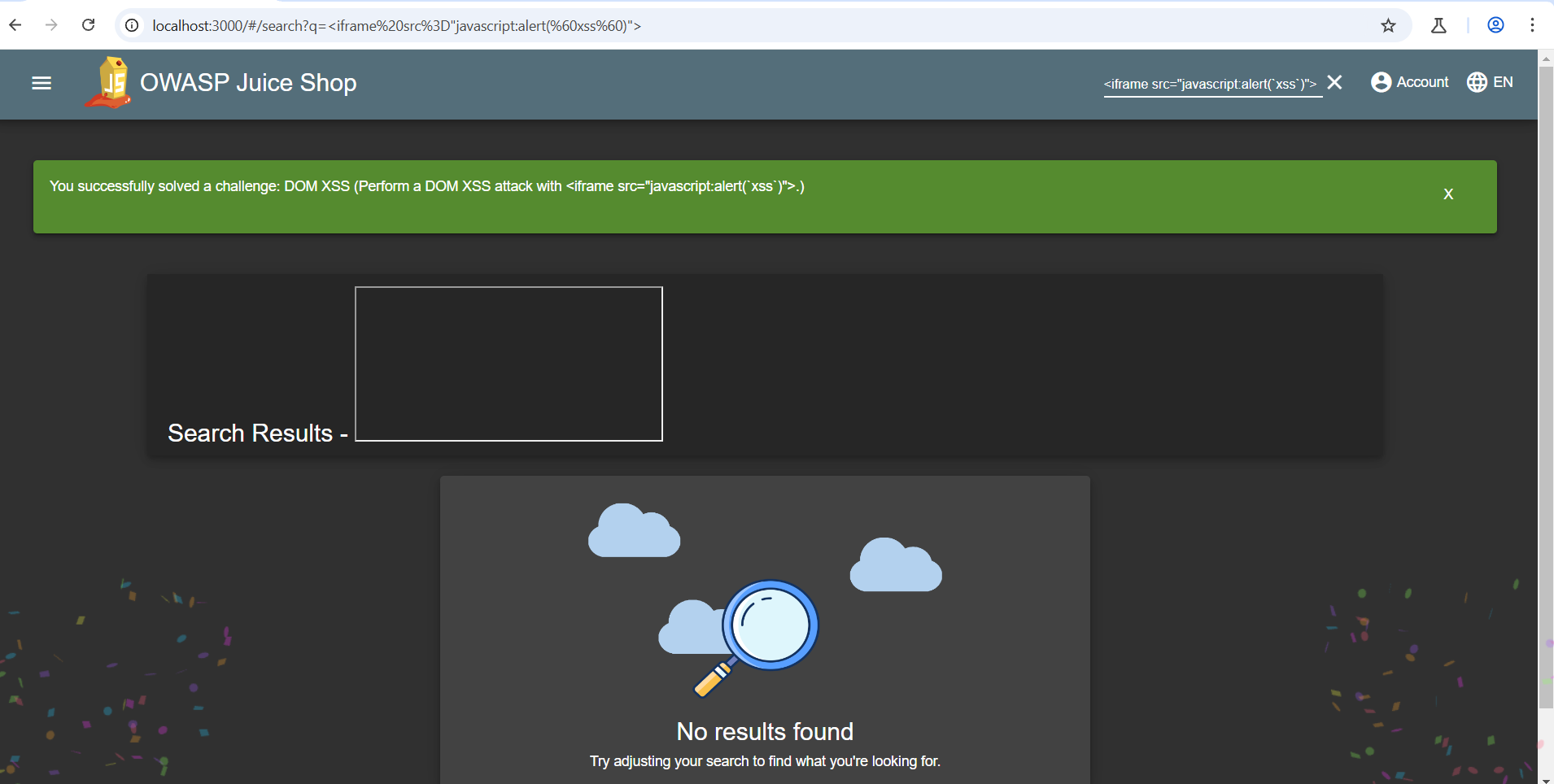
**Evidências:**

Foi inserido um script JavaScript malicioso no campo de pesquisa (**search**), com o seguinte código:

<iframe src="javascript:alert('xss')">.







**Recomendações:**

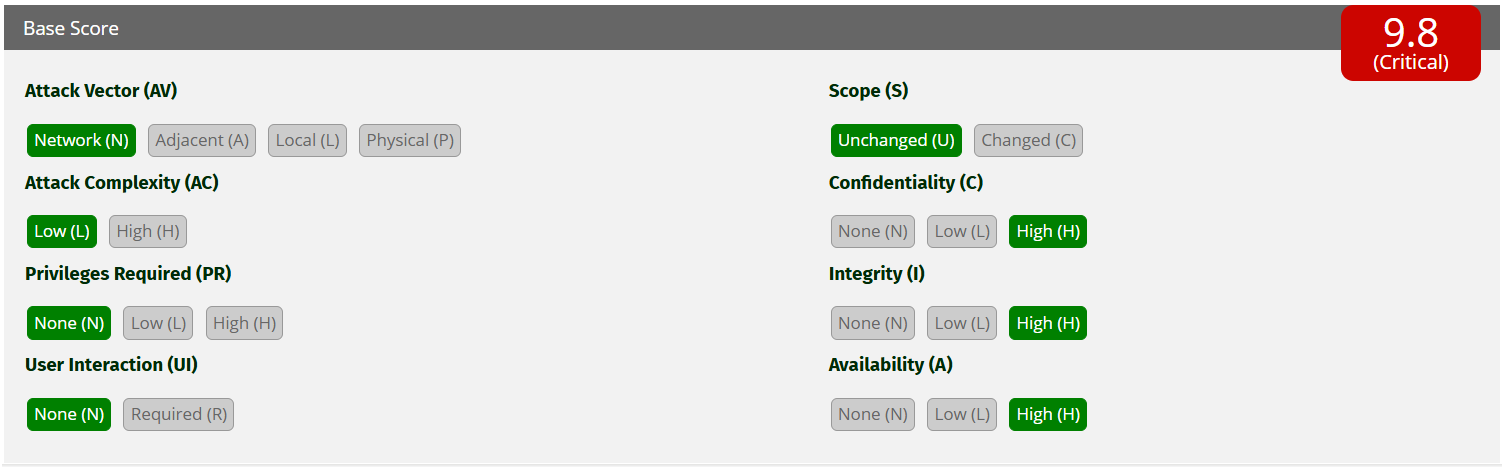
* **Sanitização da entrada:** Sempre validar e sanitizar entradas fornecidas pelos usuários, especialmente quando manipuladas e inseridas no DOM.
* **Uso de textContent ou setAttribute:** Evitar o uso de métodos como innerHTML, que interpretam HTML e permitem a execução de código JavaScript. Preferir textContent ou setAttribute, que tratam dados como texto puro.
* Escapar caracteres especiais em entradas de dados, como <, >, &, etc.
* Utilizar políticas de segurança de conteúdo para restringir a execução de scripts em páginas da web.

Em resumo, a vulnerabilidade DOM XSS ocorre quando o código JavaScript não manipula corretamente dados dinâmicos, o que possibilita a execução de scripts maliciosos no navegador da vítima. Para mitigar essa vulnerabilidade, é fundamental aplicar boas práticas de segurança no manuseio e exibição de dados fornecidos pelo usuário.

**Access a Confidential Document**

**Severidade (CVSS3.1):** 3 (Alta)

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H**



**CWE:**

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

CWE-538: Insertion of Sensitive Information into Externally-Accessible File or Directory

**Localização:**

* <http://localhost:3000/robots.txt>
* <http://localhost:3000/ftp>
* <http://localhost:3000/ftp/quarantine>

**Descrição:**

Ocorre quando um atacante consegue acessar documentos ou informações confidenciais sem autorização. Esse tipo de ataque pode ocorrer devido a falhas de segurança, como controle inadequado de permissões de acesso, falta de autenticação forte ou vulnerabilidades em sistemas de armazenamento de dados. Ao explorar essas falhas, o atacante pode obter informações sensíveis, como dados financeiros, pessoais ou comerciais, o que pode resultar em danos significativos à reputação da empresa e à privacidade dos indivíduos afetados.

Foi inserido na url (<http://localhost:3000/robots.txt>) um comando de um arquivo **robots.txt** que fica hospedado na raiz do site que apresenta instruções aos robôs das ferramentas de busca, como por exemplo o Google, Bing e Yahoo. Portanto, por meio dele foi possível indicar quais diretórios que não deveriam ser rastreados e arquivos que não deveriam ser indexados.

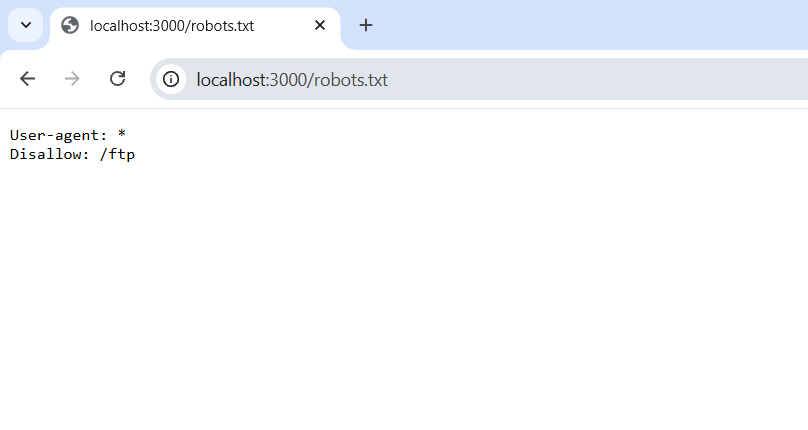
Os robôs das ferramentas de busca são programados para vasculhar toda a web em busca de novas páginas ou atualizações.

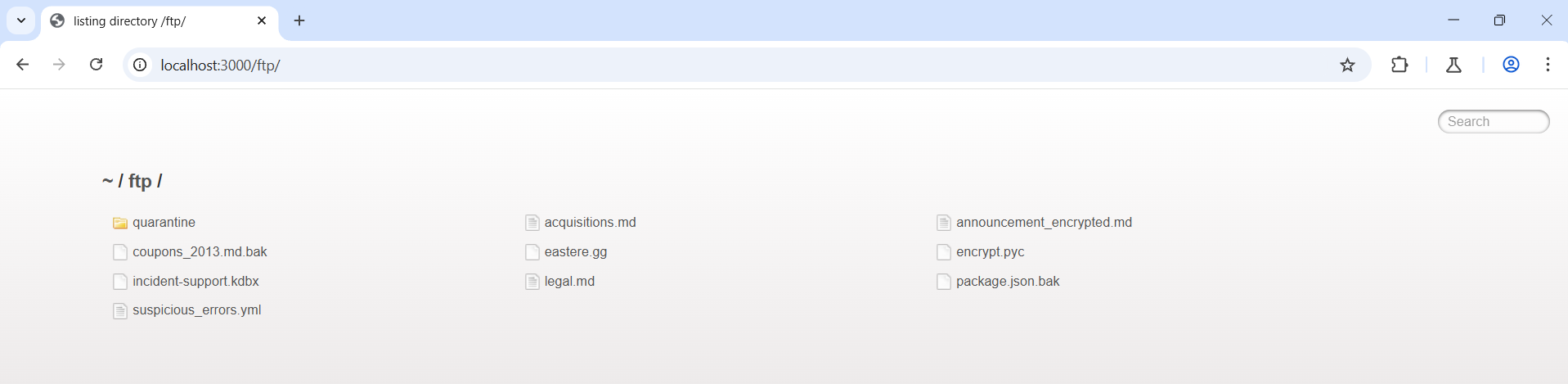
As consequências de um Access a confidential document são:

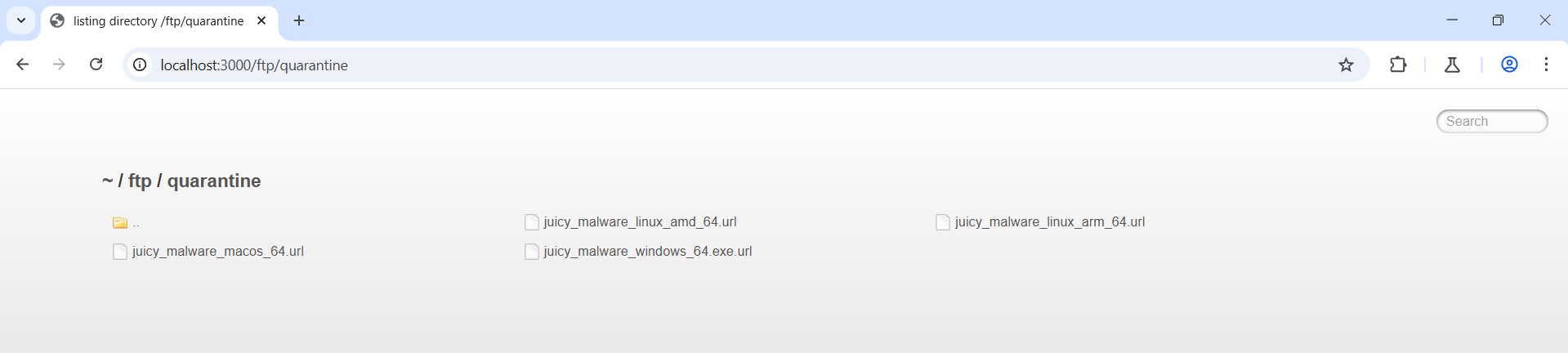
* Vulnerabilidade a Outros Ataques como SQLi, XSS, etc.;
* Danos à reputação da empresa pois a exposição de dados confidenciais pode prejudicar na confiança, confiabilidade e integridade;
* Roubo e exposição a dados sensíveis, etc.

**Evidências:**

Foi inserido o comando "/robots.txt" na URL<http://localhost:3000>, realizando uma varredura no site e expondo diretórios sensíveis, como "/ftp" e "/quarantine".







**Recomendações:**

* Controle Rigoroso de Acesso;
* Utilize autenticação multifatorial (MFA);
* Utilize criptografia de dados em repouso e trânsito, para impedir que, mesmo em caso de acesso não autorizado, não haja vazamento de dados;
* Evite expor arquivos de diretórios que possam ser facilmente acessados por robôs;
* Estabeleça políticas de segurança de dados;
* Utilize ferramentas de firewalls, antivírus, etc.