FICHA TRABALHO 3

MODALIDADE:	Aprendizagem +	Não aplicável	
CURSO:	Técnico de Gestão da Informação e Ciência de Dados		
UFCD:	Segurança no desenvolvimento do Software	CÓDIGO UFCD:	10795
FORMADOR/A:	Bruno Silva	DATA:	

OBJETIVOS

- Introdução a ferramenta OWASP ZAP e aprender como configurar e atacar aplicações com casos práticos

Segurança de sistemas de informação

Tal como qualquer linguagem de programação, não estamos livres de potenciais erros ou falhas, e como tal, são frequentemente detetadas vulnerabilidades para que seja feita as correções nas versões seguintes, e assim contribuir para que as linguagens sejam cada vez mais fiáveis e robustas.

Contudo, nenhuma linguagem de programação consegue resistir a algumas práticas de programação menos corretas, que podem levar, muitas vezes, ao desenvolvimento de scripts/aplicações pouco seguras, contribuindo assim para alguma má reputação da linguagem de programação em causa.



Por isso, não há segurança nem robustez de linguagem de programação que resistam, quando o programador não emprega boas práticas de programação e desenvolvimento de aplicações ou quando introduz erros ou falhas na aplicação, que podem ser exploradas por atacantes com algum tempo e dedicação.

Resumindo, quando ***ignoramos algumas das medidas preventivas***, estas por sua vez não só colocam o servidor da aplicação, mas também o utilizador final em risco, pois a simples injeção de código malicioso poderá deixar a máquina suficientemente vulnerável para desencadear um ataque com grandes consequências.



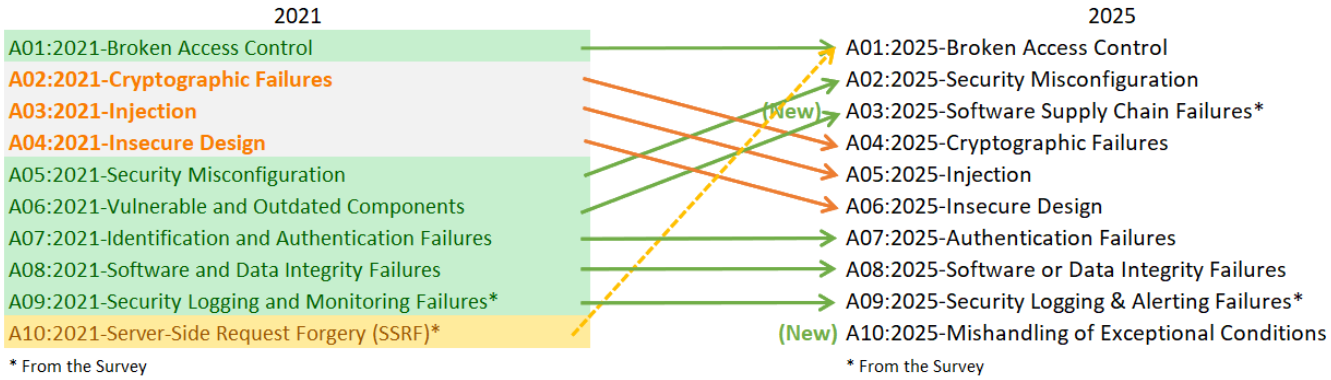
Para perceber as fraquezas desta linguagem, vamos analisar sobre os aspetos de segurança, mais especificamente, as top 10 vulnerabilidades Web inventariadas pelo ***Open Web ApplicationSecurity Project*** (OWASP) e as possíveis formas de proteção que poderão ser aplicadas.

Vamos ver a versão mais recente ***que é a versão 2025: https://owasp.org/www-project-top-ten/*** ou ***https://owasp.org/Top10/2025/0x00_2025-Introduction/***

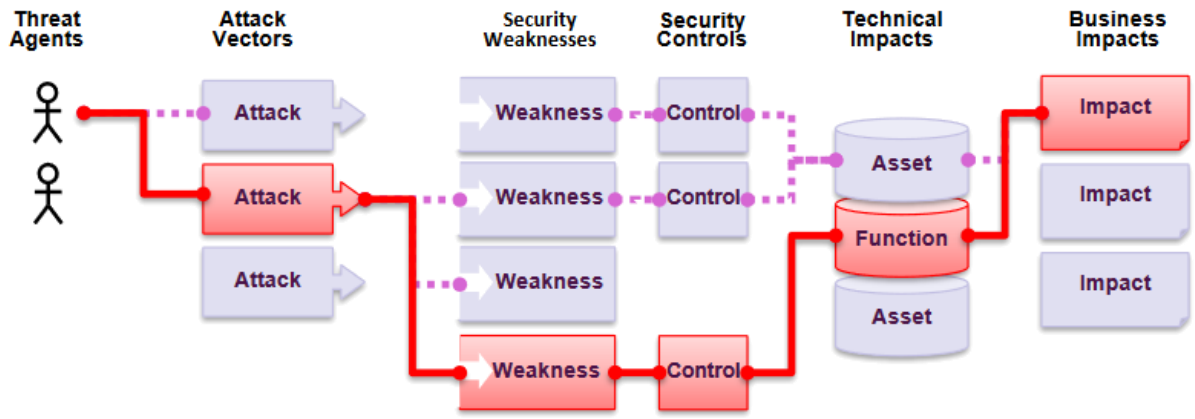
No entanto, vamos comparar e ver as diferenças que houve da versão de 2021 para 2025!

Lista das TOP 10 vulnerabilidades da Web

Comparação da versão top-10 de 2021 para 2025 (mais atual):



Metodologia de ataques



Os invasores podem usar vários caminhos de ataque na nossa aplicação, com o objetivo de atingir a informação essencial. Cada um desses caminhos representa um risco que pode, ou não, ser sério o suficiente para merecer atenção.

Às vezes, esses caminhos são triviais de encontrar e explorar e, às vezes, são extremamente difíceis. Da mesma forma, o dano causado pode não ter consequências. Para determinar o risco para a organização, temos de avaliar:

- ***a probabilidade associada a cada agente de ameaça;***
- ***vetor de ataque;***
- ***vulnerabilidade de segurança;***

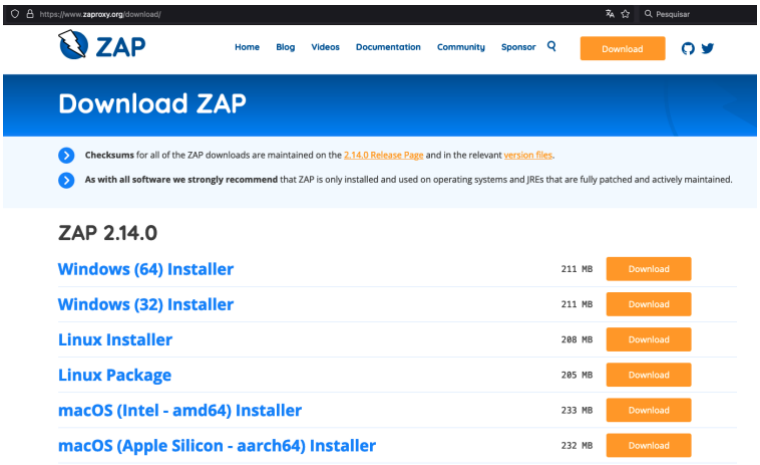
Juntos, esses fatores determinam o risco geral do impacto na segurança.

O ***OWASP Top 10*** foca na identificação dos ***riscos de segurança mais graves***. Para cada um dos riscos, são fornecidas informações sobre a probabilidade e o impacto técnico usando o seguinte esquema:

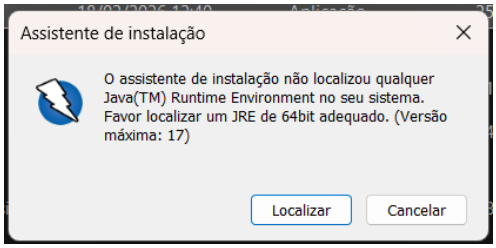
Agentes de ataque	Explorabilidade	Prevalência de fraqueza	Deteção da ameaça	Impactos Técnicos	Impacto no negócio
Aplicação especifica	*Fácil: 3*	*Generalizado: 3*	*Fácil: 3*	*Severo: 3*	*Negócio especifico*
	Médio: 2	*Comum: 2*	*Médio: 2*	*Moderado: 2*	
	Difícil: 1	*Incomum: 1*	*Difícil: 1*	*Menor: 1*	

Instalação da aplicação OWASP ZAP

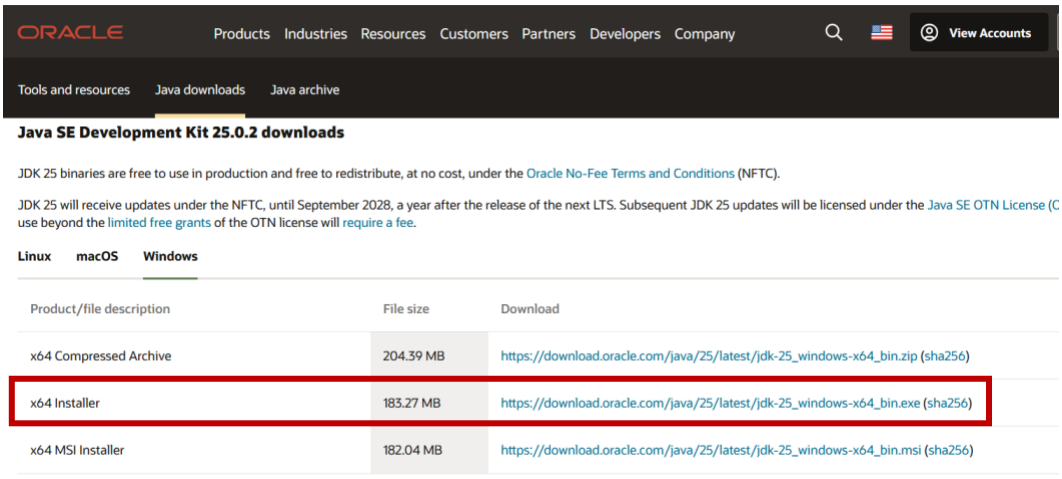
1. Para descarregar a ferramenta ZAP, devemos ir ao seguinte endereço: https://www.zaproxy.org/download/ e retirar o software para o sistema operativo Windows 32 bits ou 64 bits (dependendo do processador da vossa máquina):



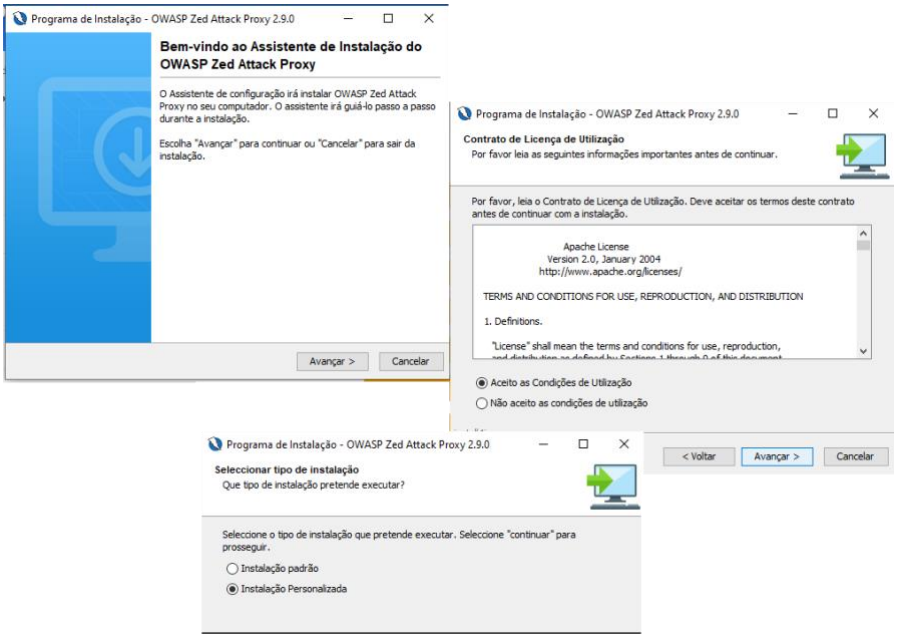
Nota Importante: o programa utiliza uma biblioteca que pertence a linguagem de programação Java (JDK). Como o programa foi desenvolvido nessa linguagem de programação e se o mesmo não detetar a biblioteca com esses dados, pode aparecer a seguinte informação:

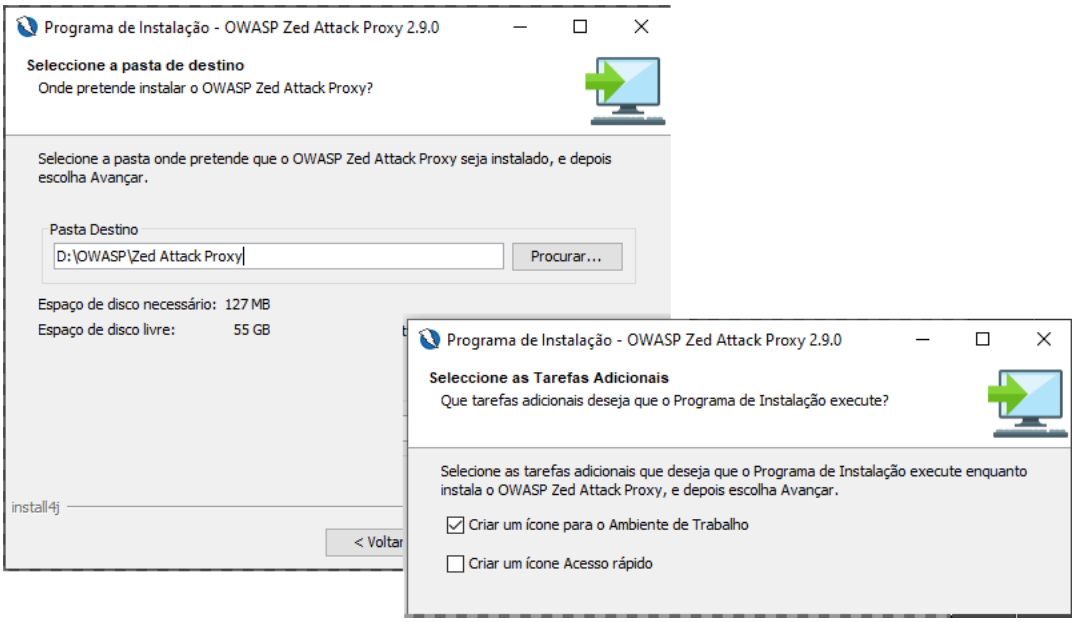


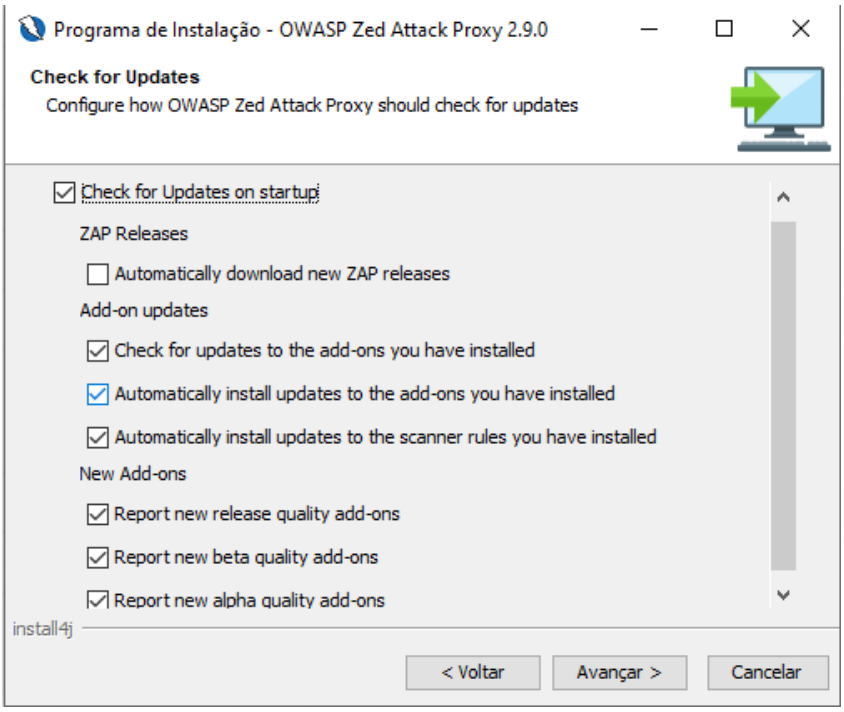
Para retirar a biblioteca do JDK/JRE deve clicar na seguinte hiperligação: https://www.oracle.com/java/technologies/downloads/



De seguida, vamos fazer a instalação da aplicação

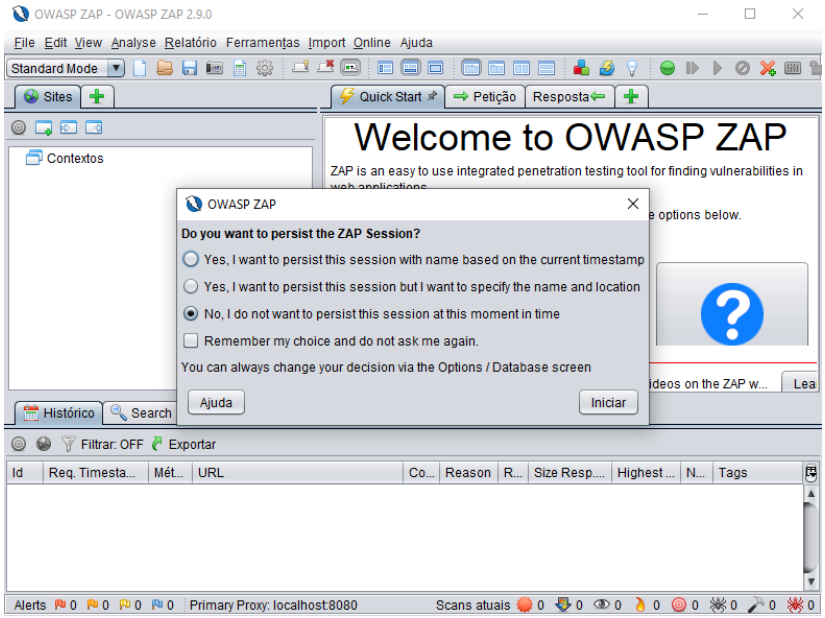






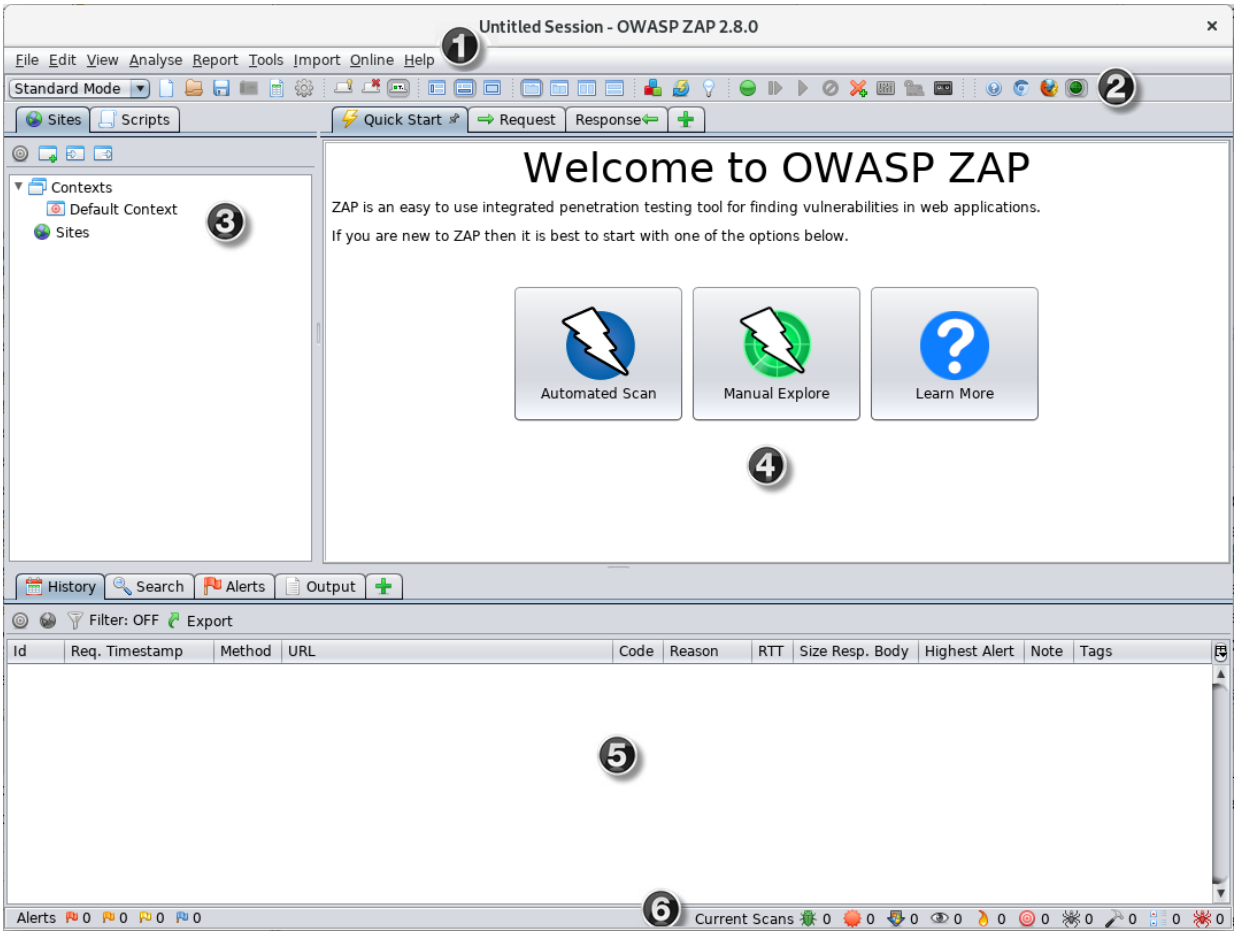
2. Ao iniciar o ZAP pela primeira vez, será perguntado se deseja persistir com a sessão ZAP. Por padrão, as sessões ZAP são sempre registadas numa base de dados com um nome e local padrão. Se não quiserem ter sessão ativa, os dados e pesquisas realizados no programa serão excluídos quando sair do programa.

   Por enquanto, selecione ***Não, não quero persistir esta sessão neste momento*** e clique em Iniciar.



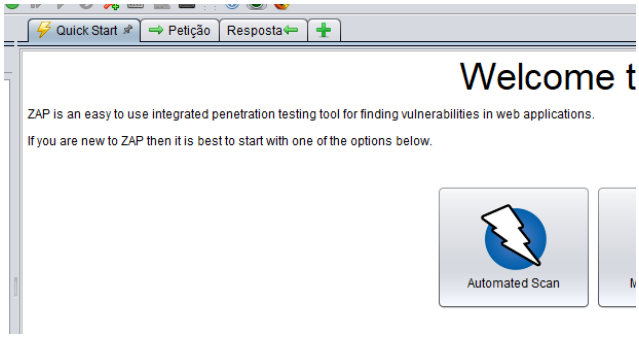
3. O ambiente de trabalho e constituído ***pelas seguintes seções***:

   1. ***Barra de menus*** - Fornece acesso a muitas das ferramentas automatizadas e manuais;
   2. ***Barra de ferramentas*** - Inclui botões que fornecem acesso fácil aos recursos mais usados;
   3. ***Janela em árvore*** - exibe a árvore de sites e a árvore de scripts;
   4. ***Ambiente de trabalho*** - mostra solicitações, respostas, scripts e permite que a edição dos mesmos;
   5. ***Janela de informações*** - mostra detalhes das ferramentas automatizadas e manuais;
   6. ***Rodapé*** - mostra um resumo dos alertas encontrados e o status das principais ferramentas automatizadas;

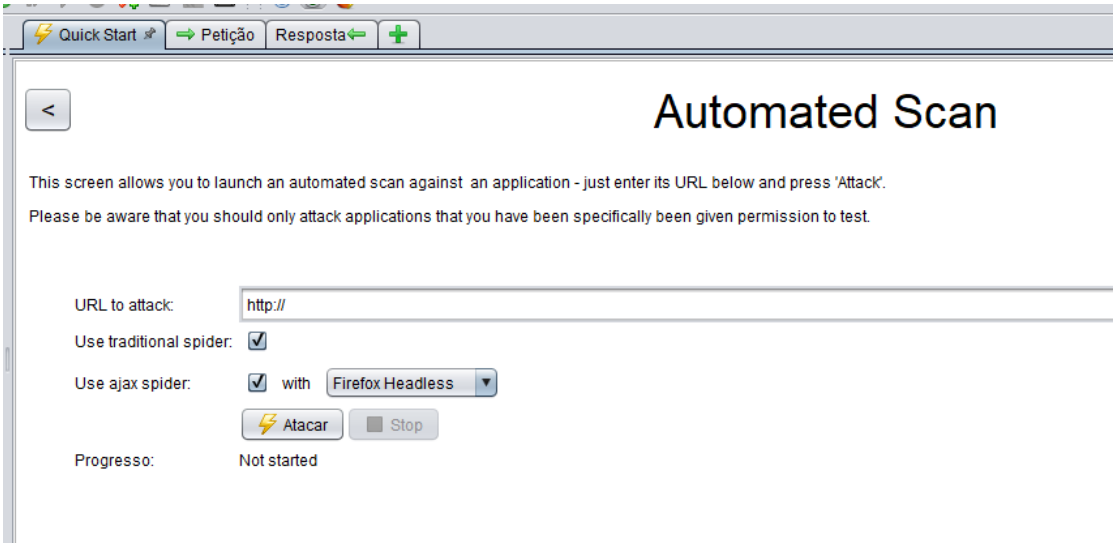


- Neste momento, já estamos em condições para um conjunto de testes de auditoria, baseados em testes de intrusão, para verificar a existência de vulnerabilidades de segurança em aplicações.
  - ***Nota Importante***: Pode haver vulnerabilidades fora dos top 10 ameaças da OWASP.

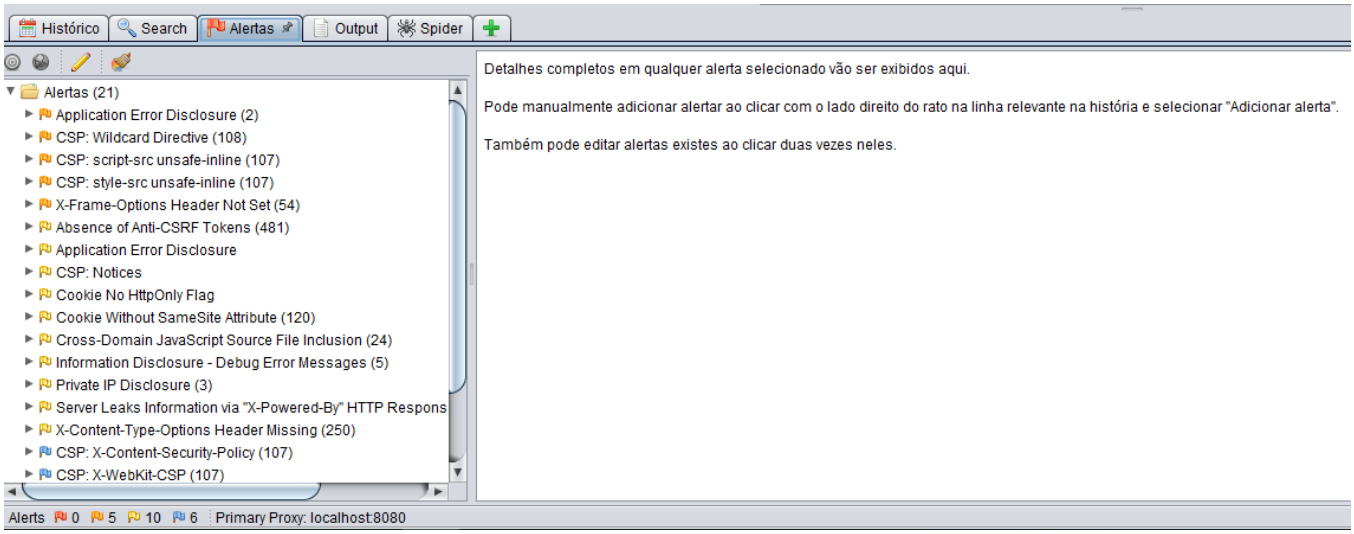
- A maneira mais fácil de começar a usar o ZAP é por meio da guia Início rápido. Para tal devemos clicar no seguinte ícone, para abrir a janela de execução automatizada e clicar na opção "***Automated Scan***":



- Na janela de execução automatizada, vamos ter as seguintes opções:
  - ***Url to attack***: localização da aplicação onde queremos executar os testes/ataques;
  - ***Use Traditional spider***: ataca a aplicação com técnicas de intrusão para encontrar vulnerabilidades de segurança e regista alertas das solicitações e respostas enviadas para cada página se houver algo errado;
  - ***Use AJAX Spider***: igual ao anterior ***Use Traditional spider,*** mas esta opção é mais virada para testes de scripts (também podemos simular qual o navegador que vai utilizar para atacar);



- A medida que vai testando a aplicação, são exibidos os avisos (como podemos ver no próximo diapositivo):



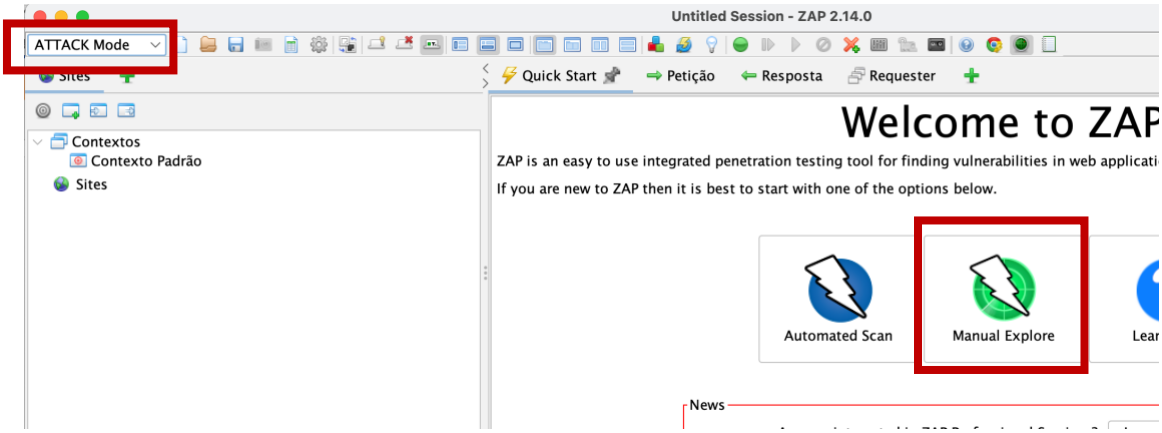
- Reparem que os avisos são representados com as seguintes bandeiras:

High	***Vulnerabilidade muito grave / Grande impacto***
Medium	*Vulnerabilidade média / Médio impacto*
Low	*Vulnerabilidade baixa / Pouco impacto*
Informational	*Vulnerabilidade muito baixa / Muito Pouco impacto*
False Positive	*Falsos positivos / Embora que não represente perigo, devemos jogar pelo seguro e verificar a causa.*

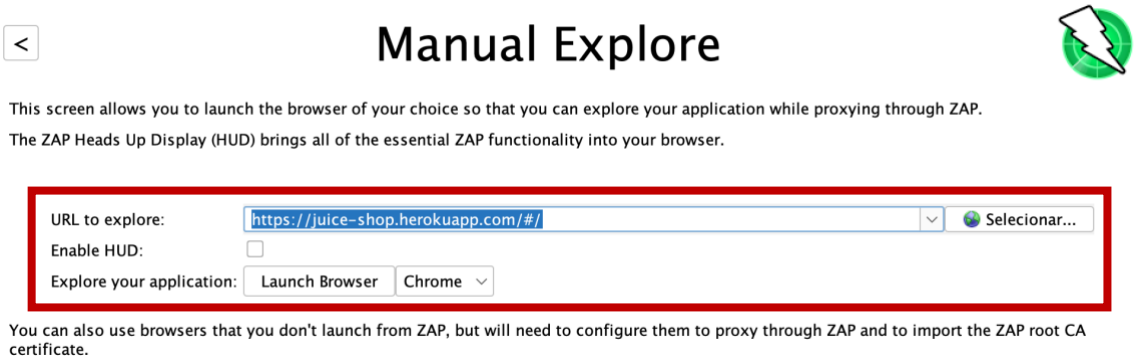
Casos de Estudo OWASP ZAP

Exemplo 1 - SQL Injection

Colocar o em modo de ataque:



Colocar o seguinte endereço: https://juice-shop.herokuapp.com/#/ (em alternativa: https://preview.owasp-juice.shop/#/) e não ter ativado a opção “Ativar UHD”:



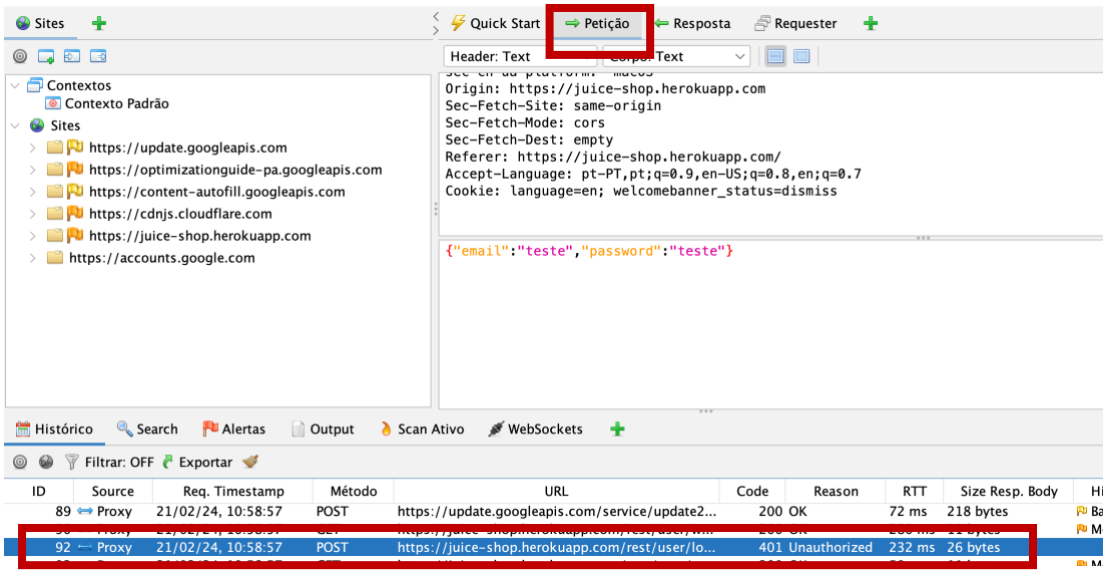
De seguida, o programa vai abrir uma sessão do navegador em modo de testes e vamos fazer login no website (devem esperar algum tempo, para este carregar o conteúdo da página.

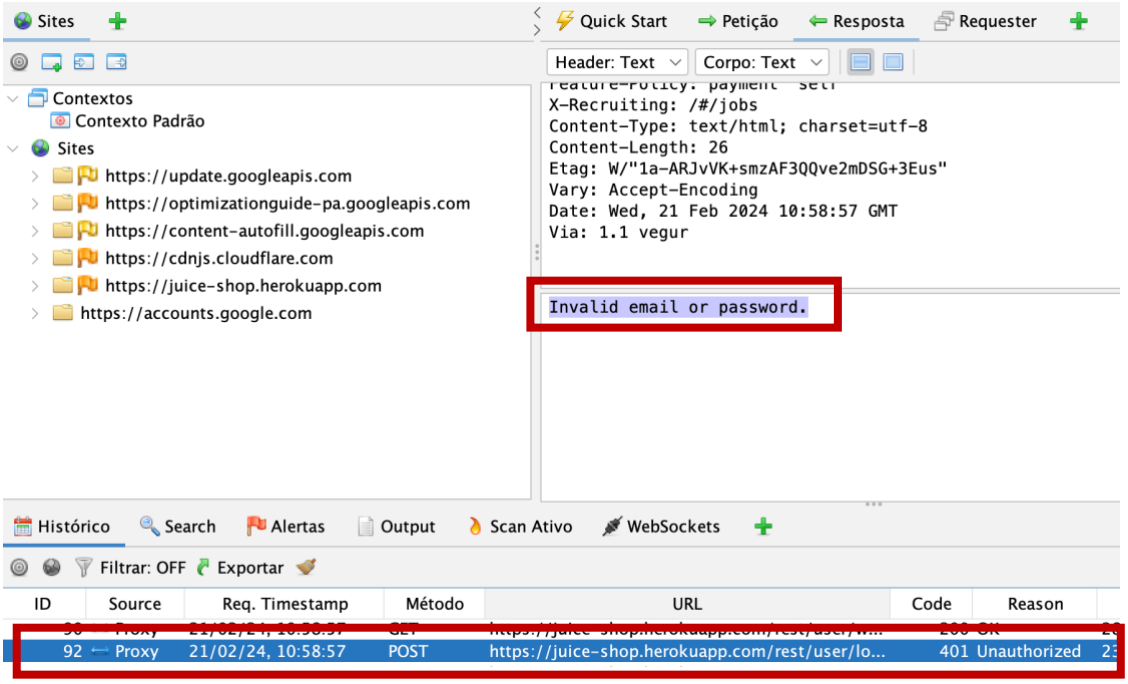
Clique na opção para fazer login e coloque os dados:



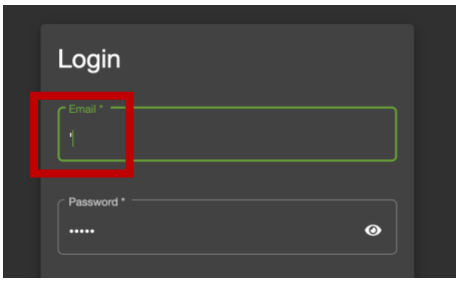
- Utilizador: teste
- Password: teste

De seguida volte ao ZAP e analise o histórico dos pedidos do navegador (no pedido e na resposta):

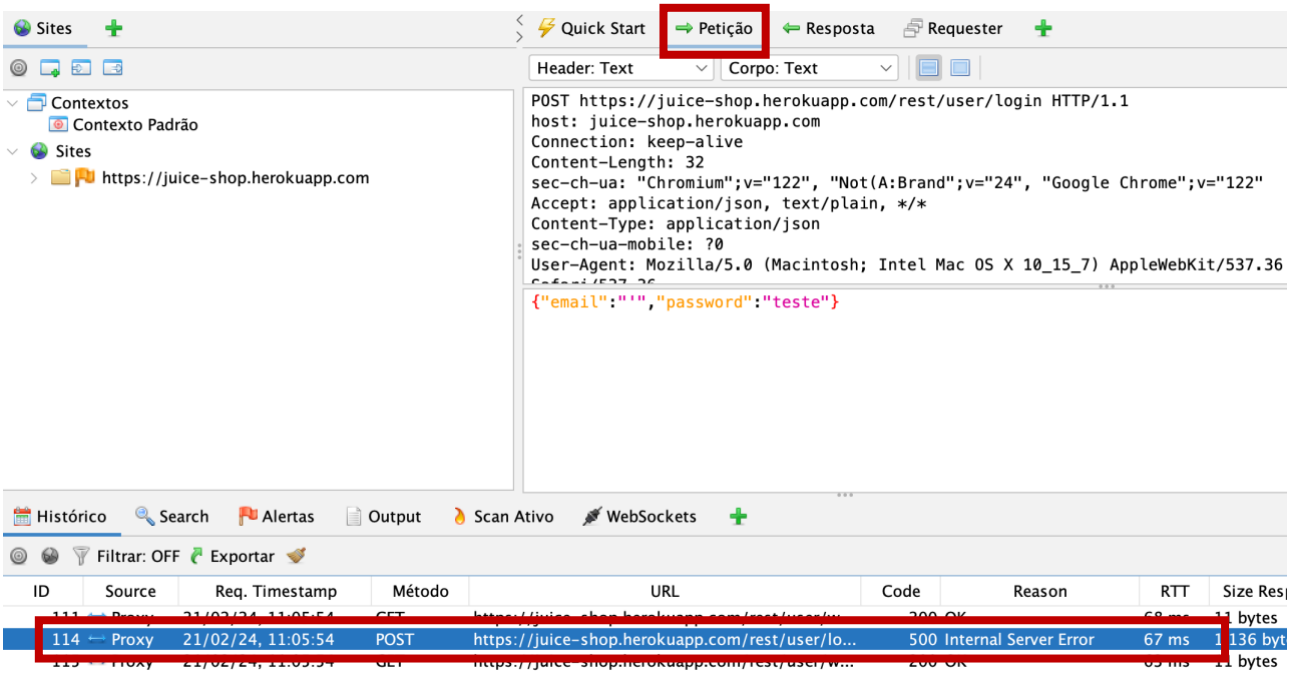


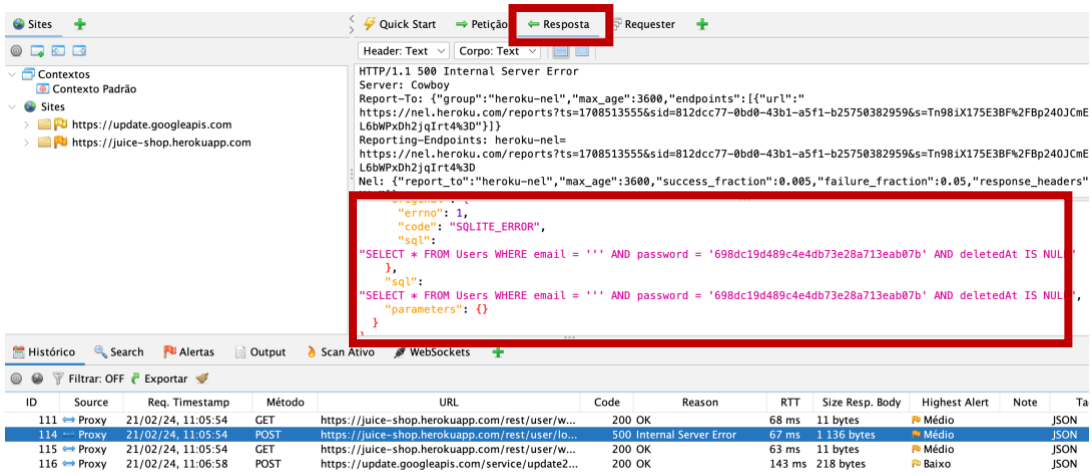


Este apenas deu a mensagem que não existe dados, mas foi apenas uma mensagem muito simples. Agora, vamos tentar enganar o nosso sistema e vamos colocar um caracter especial ‘ (aspa simples) para atrapalhar o processamento do campo do ***nome do utilizador*** (password pode ficar igual):



Voltem novamente ao histórico da navegação e vejam se este devolveu mensagem ou se devolveu todo o processo de autenticação (com as instruções nas bases de dados), do qual ressoltou um erro de processamento interno no servidor (naveguem no separador Petição para ver os dados que foram enviados e depois no separador resposta para ver os dados do erro):

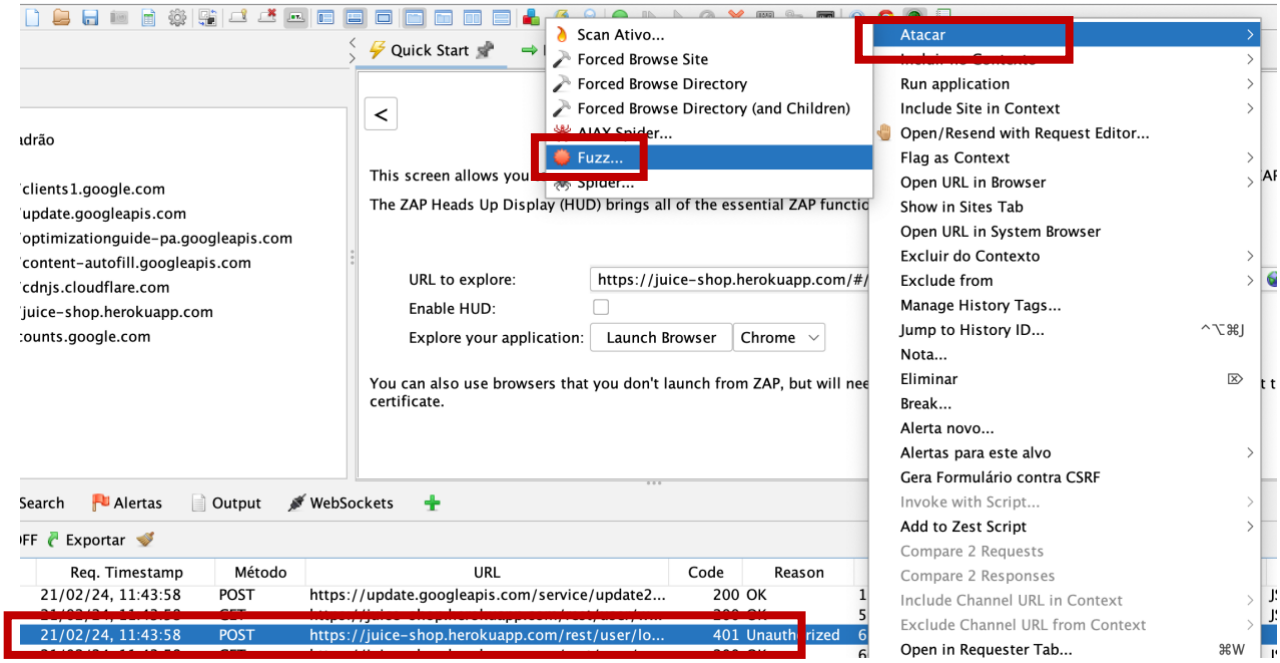




Isto é muito mau (para não dizer desastre)!!!! Reparem que conseguirem ver as instruções de consultas na base de dados e até sabem qual a tabela e os campos onde está a ser requerida a autenticação do utilizador.

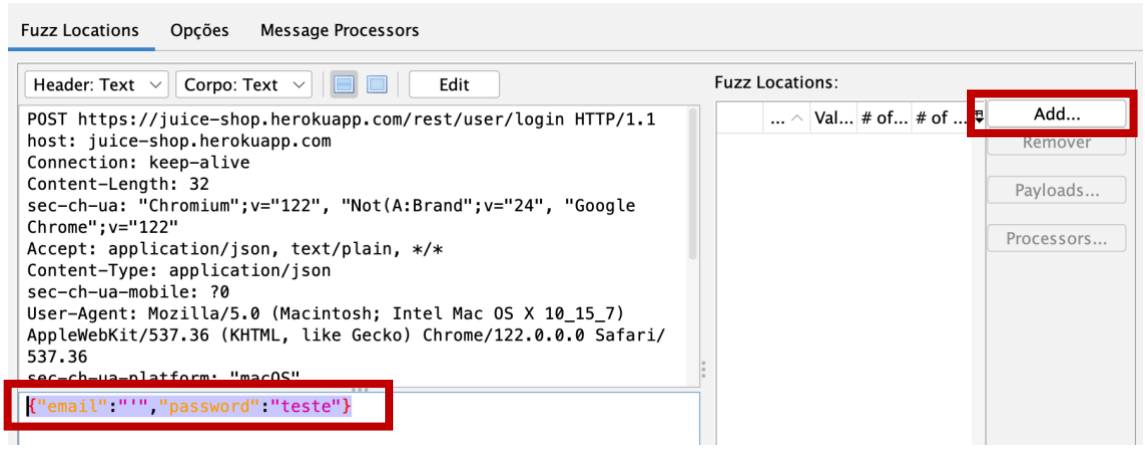
Agora que já explorarmos um pouco esta funcionalidade, vamos agora fazer um ataque mais personalizado. O objetivo será enviar dados modificados pela aplicação ZAP com o objetivo de conseguir saber informações, conhecido como SQL Injection.

Para tal, vamos clicar em cima da requisição do histórico da primeira vez que fizemos a autenticaçãoe vamos selecionar com o lado direito do rato as opções Attack → Fuzz:

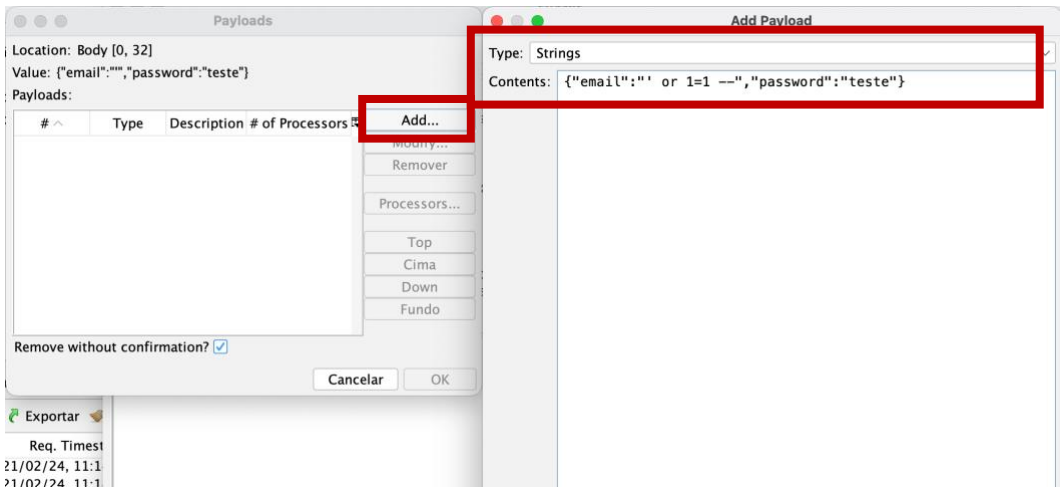


Este vai abrir a janela com detalhes do pedido da requisição e vamos alterar as informações. Para tal:

- Clique na caixa de texto com os dados que enviamos antigamente;
- ***Selecione*** e ***copie*** todo o texto da instrução (para conseguir passar a informação);
- ***Vamos alterar o pedido de requisição*** com uma instrução de SQL Injetion clicando no ***botão Add***;



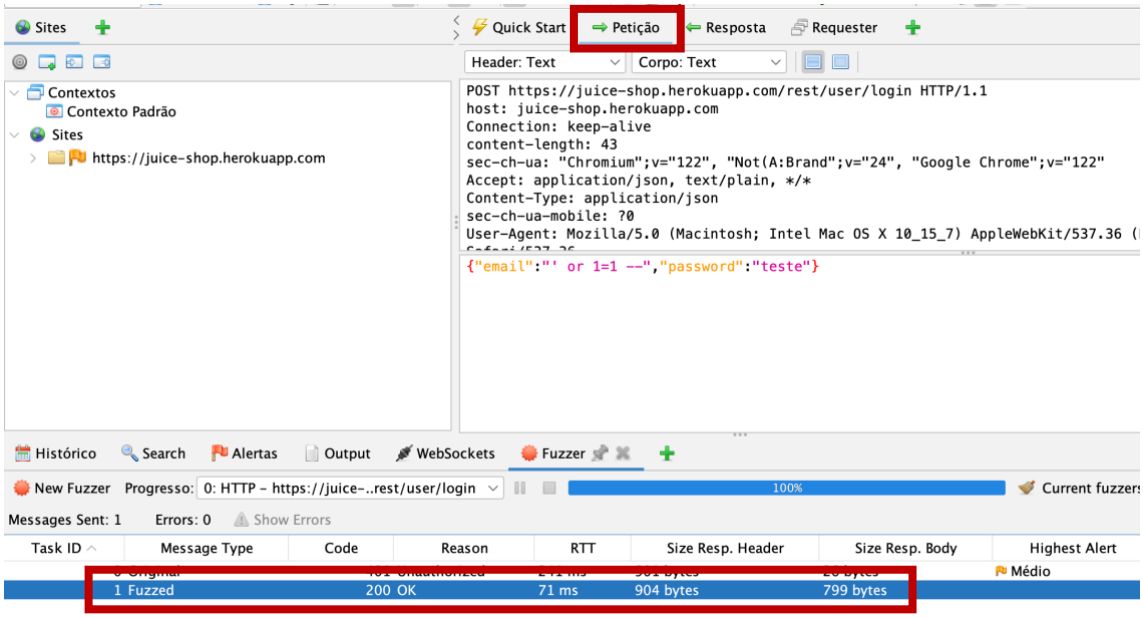
- Na nova janela irá ver as instruções que foram selecionadas e vamos indicar o seguinte payload: {"email":"' **or 1=1 --**","password":"teste"} e clique no botão para adicionar:



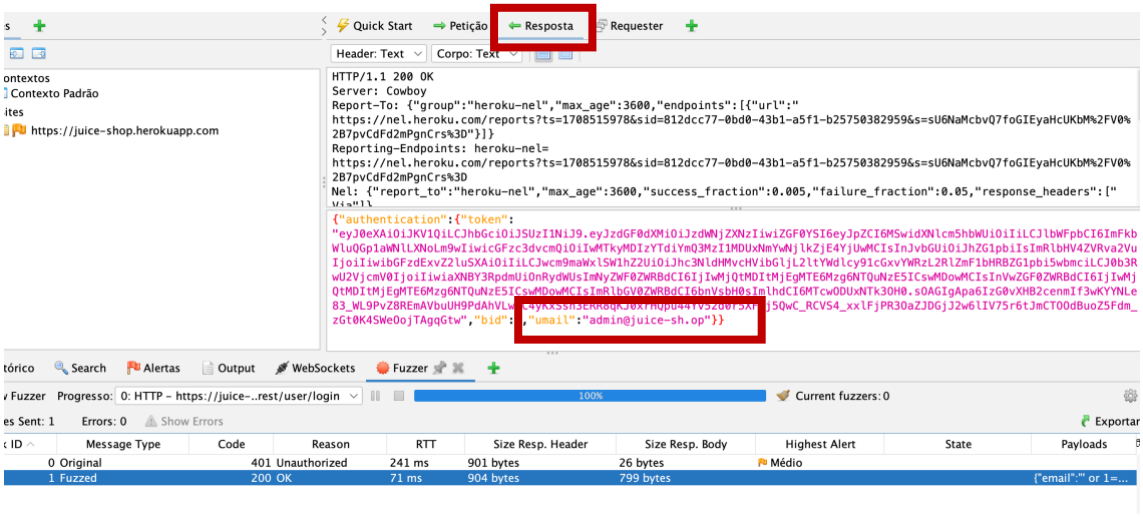
- Confirma nova operação e quando voltar a janela principal, clicar no botão “Start Fuzzer”:



- Depois cliquem no botão Add para confirmar a introdução das novas alterações e a outra janela por detrás. De seguida, veja as novas requisições que foram efetuadas, mas agora, vamos ao separador “Fuzzer” e veja os separadores do pedido e da resposta:



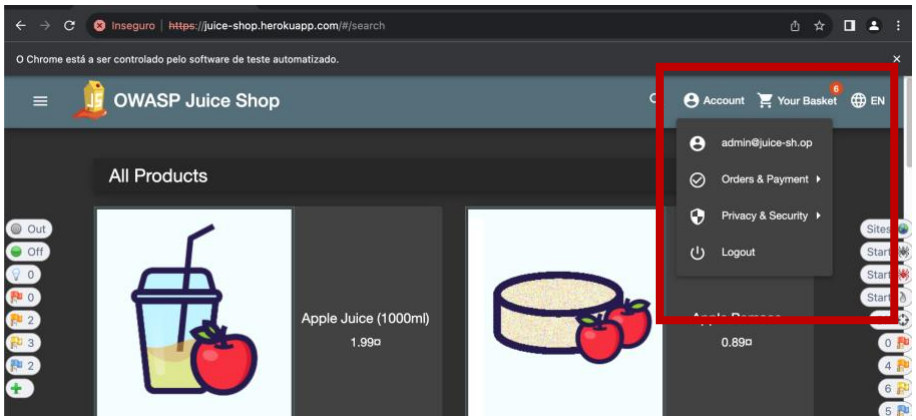
- Reparem que na resposta apanhamos as informações da tabela users (do qual podemos ver a conta do administrador):



Feito isto, vimos como funciona o ataque por detrás das requisições, agora vamos aplicar no website em tempo real. Como tal, coloquem as seguintes informações:

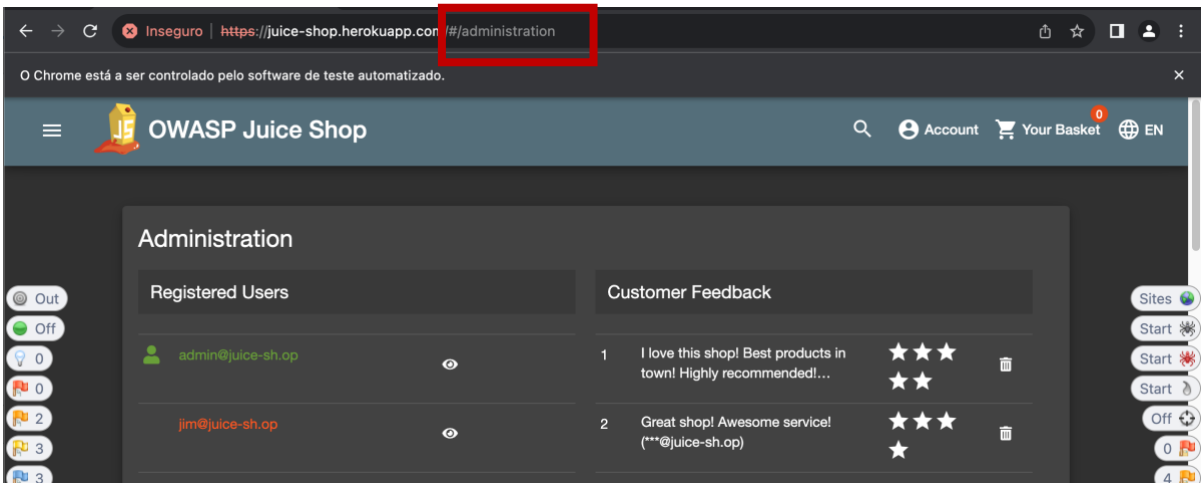
- ***Utilizador***: **' or 1=1 --**
- ***Password*****:** qualquer coisa

E.... entramos com o administrador:



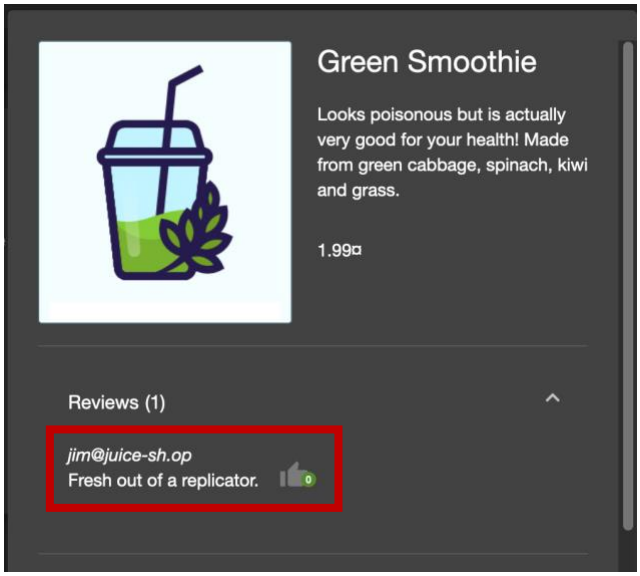
Exemplo 2 - Broken Access Control - página de administração oculta

Vamos tentar entrar em páginas protegidas pelo administrador (que não estão no menu da imagem anterior). Para tal, escreva o seguinte endereço: https://juice-shop.herokuapp.com/#/administration



Exemplo 3 - SQL Injection (teste com outras contas de utilizador)

Para entrar na conta de outro utilizador, vamos fazer outra manobra. Para tal, vamos encontrar o email do jim nos produtos da loja virtual:



Se colocar o email e uma password este não vai funcionar.

Isso não funcionou! Precisamos primeiro criar um ataque SQL!

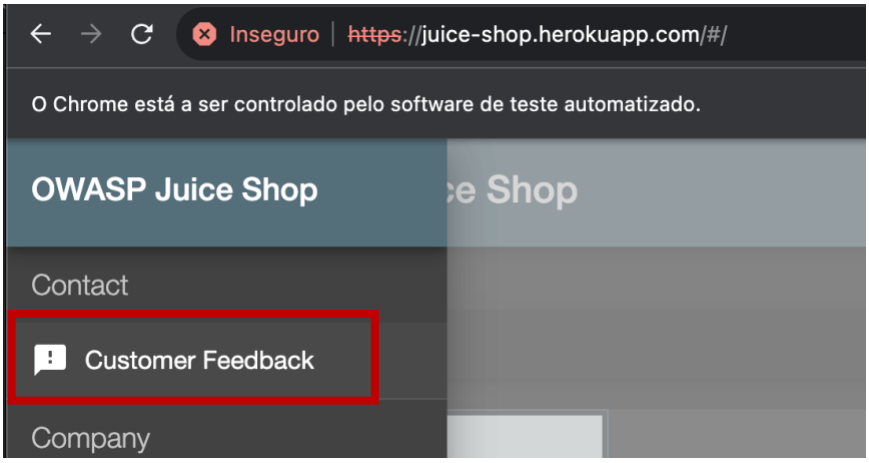


De seguida, vamos colocar o email do Jim no login da plataforma e de seguida, vamos colocar a instrução '-- (aspa simples com dois traços):

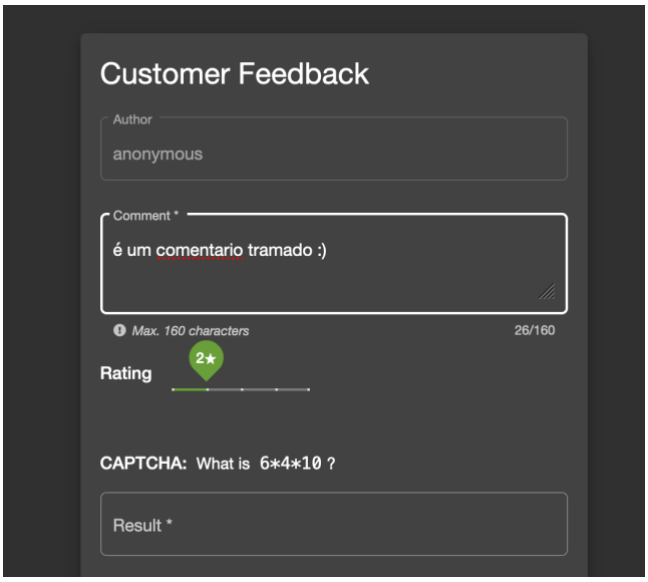
Teste e verifique o resultado!

Exemplo 4 - Insecure design - Improper Input Validation

Agora queremos ir ao suporte do cliente e dar um feedback de 0 estrelas na aplicação.

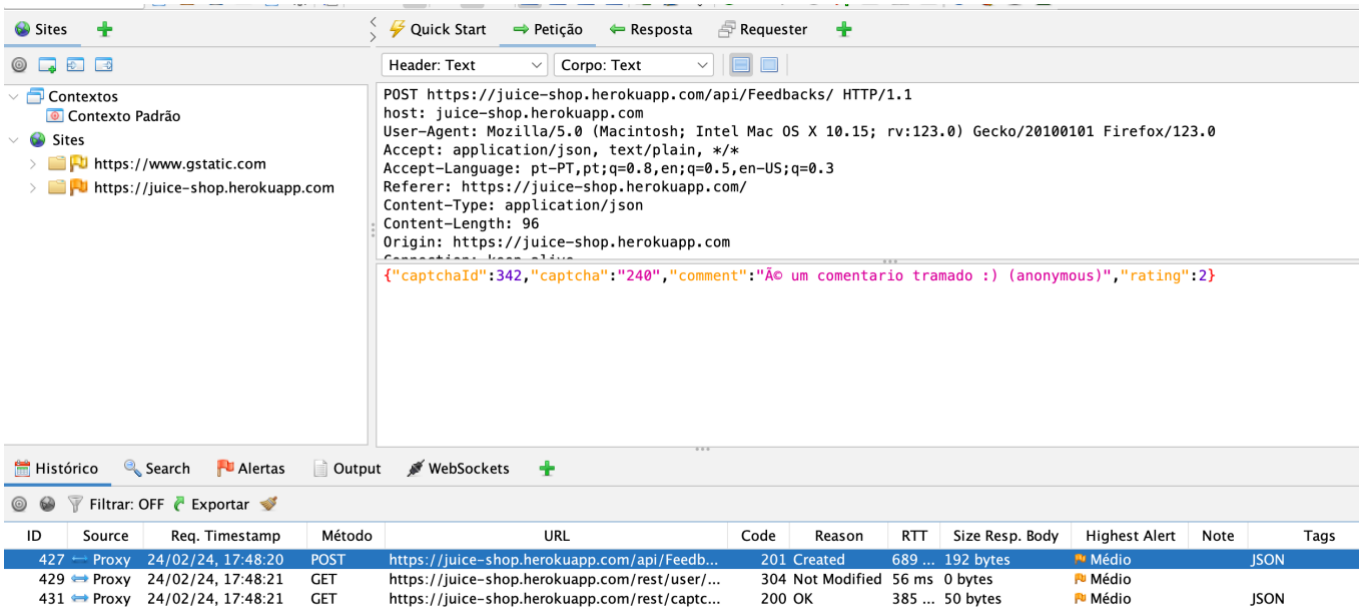


Mas como tal, este pede sempre para colocar valores de 1 a 5.

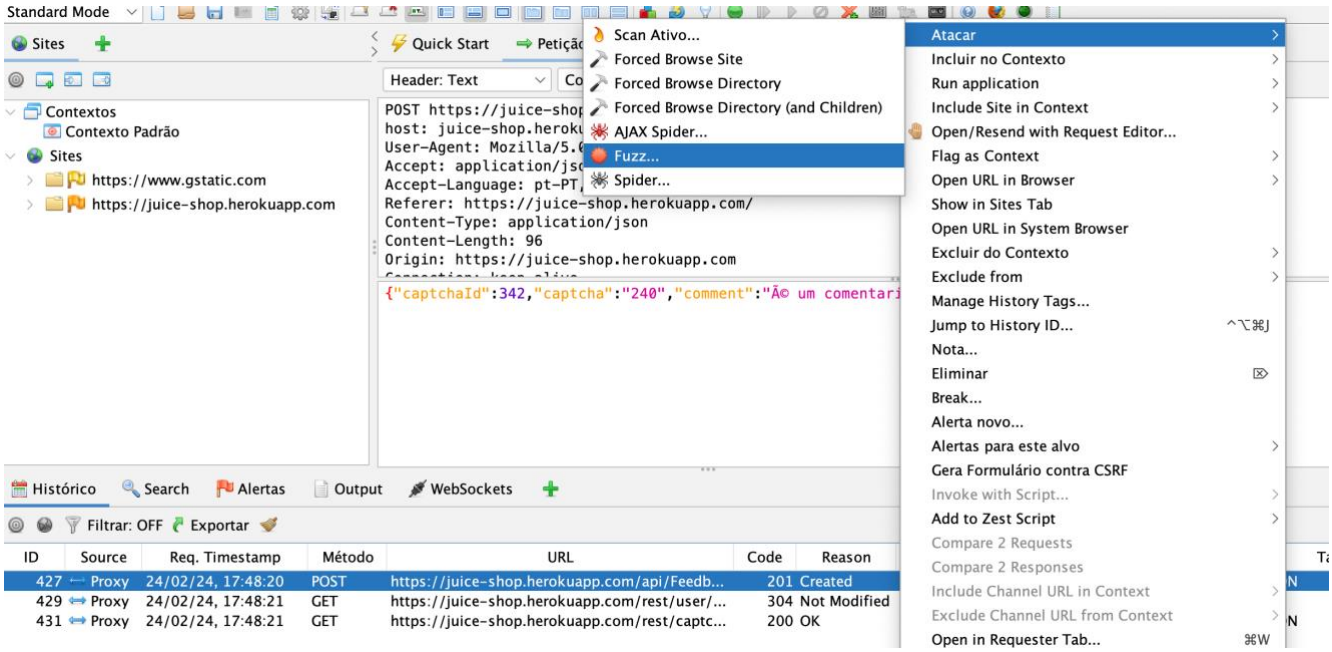


Para tal, vamos ter de intercetar a ligação quando envia os dados para a aplicação e de seguida, mudar os parâmetros de envio de informação.

Preencha uma vez os dados do formulário e envie o feedback. De seguida, vamos ao nosso ZAP e verifique o pedido com dados que forma enviados:



Como tal, vamos fazer a alteração dos valores e verificar se a plataforma vai ser ou não enganada. Vamos clicar o lado direito do rato em cima da ligação e escolher as opções:

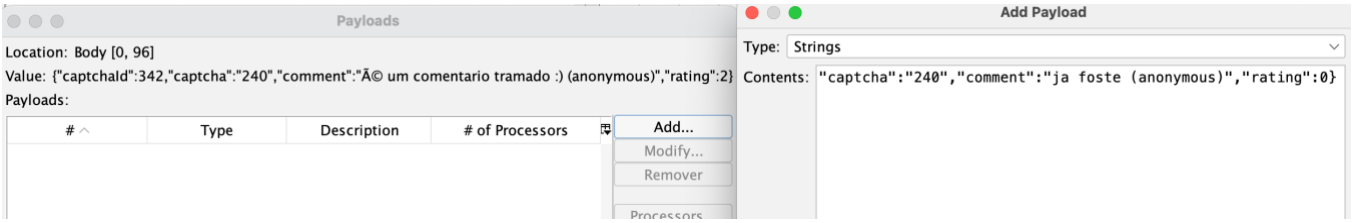


Depois vamos fazer os mesmos passos que o exercício que modificou os dados para descobrir os dados do administrador.

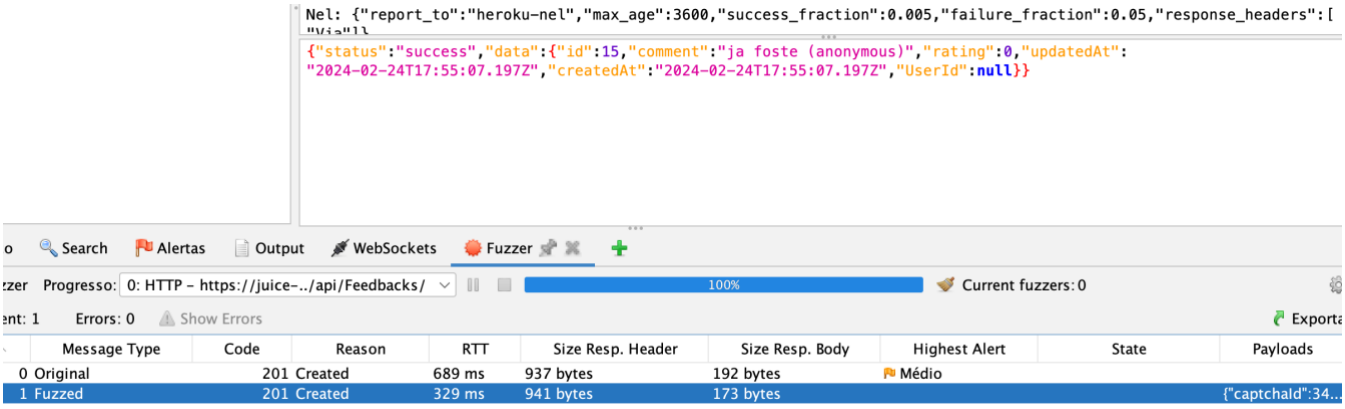
1 – Copie a informação do pedido e deixe selecionado o texto de envio da informação;

2 – Clique no botão Add;

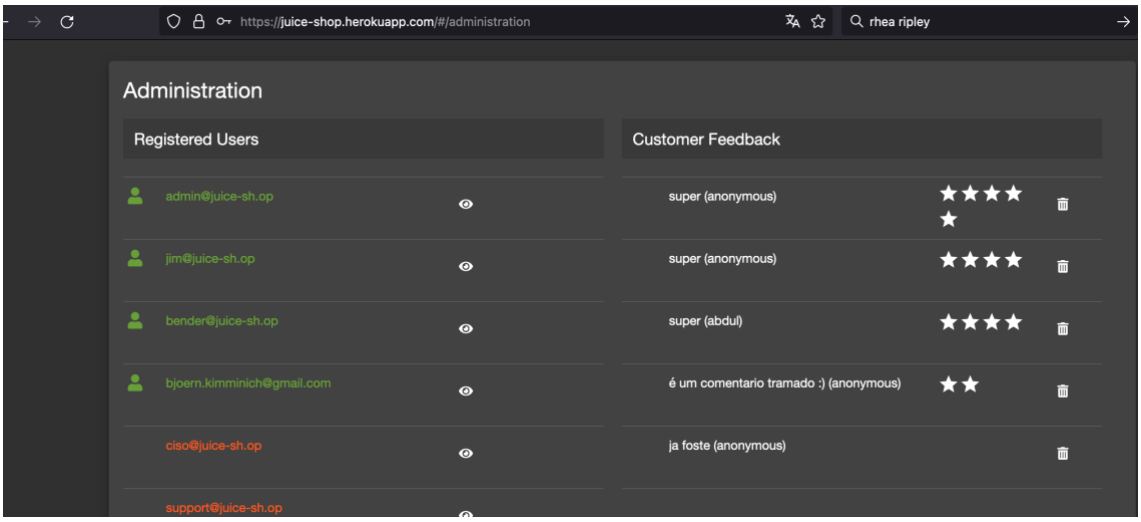
3 – Coloque os dados e altere a mensagem para já foste e o rating para o valor 0;



4 – Submeta o novo pedido e veja a resposta:



5 – Entre na página como administrador e aceda a página secreta da administração e confirme o resultado:



Exemplo 5 – Ataques XSS

Entre na conta do Jim e vá a edição do perfil. De seguida, vamos ao campo do Username (nome de Utilizador) e alterar a informação para a seguinte:

<script>alert("já foste!")</script>

Como tal, este código não vai funcionar pois já existe validação para elementos que comecem o código script e as bases de dados podem recusar atualizar a informação. Como tal, vamos enganar o sistema e vamos dar uma outra forma de introdução do ataque XSS mas com a introdução de uma nova maneira:

<<script>sscript>alert("já foste!")</script>

Carregue no botão para definir o novo nome de utilizador e verifique o resultado.

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

```
<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe>
```

Gerar Relatórios

Depois de atacar ou fazer as operações de testes numa aplicação, poderá gerar o relatório de todos os pontos que este gerou.

