Cibersegurança

Grupo 11

Nota: Encontram-se fotografias de todos os exercícios realizados no anexo na última parte do relatório.

Exercício 1

Exercício i)

Na biblioteca marsdb, existe uma vulnerabilidade critíca de "Command Injection", esta vulnerabilidade enquadra-se nos riscos de segurança OWASP como A1 – Injection, pois encontramo-nos numa situação que é possível ser injetado comandos que causem danos.

A outra vulnerabilidade critíca é "Authorization Bypass", que implica que é possível passar ao lado de autorizações necessárias para aceder a recursos de alguma forma. Esta vulnerabilidade de acordo com a nossa análise enquadra-se em A5 – Broken Access Control, pois permite a passar por controls de segurança, neste caso de autenticação.

Exercício ii)

Foram identificadas duas vulnerabilidades de Cross-Site-Scripting (XSS), e ambas se encontram presentes no package sanitize-html.

Exercício 2

Neste exercício, andámos a procurar como eram feitos os carregamentos de ficheiros de servidor, como imagens, ficheiros pré carregados, etc. Apenas tivemos sorte quando chegámos à página de about us, e tentámos ver os termos de utilização, que eram um download. Encontrámos esse link, que era https://g11-cs-isel.herokuapp.com/ftp/legal.md, e alterámos para o link https://g11-cs-isel.herokuapp.com/ftp/aquisitions.md, que funcionou bem e fez download do ficheiro.

As primeiras três linhas são :

Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.

Exercício 3

Este exercício foi bastante simples, pois apenas implicou abrir a developer tool do Firefox, e pesquisar pela palavra chave "scoreboard", após isso foi apresentado um url que continha essa palavra. Ao testarmos esse url, pareceu ser exatamente o scoreboard. O url é https://g11-cs-isel.herokuapp.com/api/Challenges/?name=Score%20Board

Exercício 4

Na página de resultados o texto do critério de pesquisa é inserido exclusivamente no span com id "searchValue", que é o sitío que aparece à frente de "Search Results". A caixa de texto de pesquisa

utilizada para pesquisar “lemon” continua com essa mesma string. A partir daqui conseguimos inferir, que o valor da searchBox apenas é utilizado para construir o url, e não atualizada com o valor do mesmo, enquanto que o span “searchValue” é sempre atualizado com o valor mais recente do url no caso de uma pesquisa.

Exercício 5

Neste exercício começámos por tentar injetar o texto fornecido no único sítio onde conseguíamos inserir texto até ao momento, que era na barra de pesquisa. Para tal começámos por testar aí, pois para efetuar um ataque de XSS temos sempre de injetar código no site de alguma forma. Testando com o texto apresentado no enunciado, após pesquisar apareceu o alerta a dizer “xss” que era o esperado, de acordo com o texto apresentado.

Exercício 6

Para este exercício era proposto que encontrássemos a password da conta de administrador do serviço, para fazer isto, fizemos uma primeira tentativa com o user admin@juice-sh.op, e com a password “admin000”, pois ia de encontro ao explicado no enunciado. Após isso fomos ver o pedido no ZAP, e encontrámos, e escolhemos a opção Attack/Fuzzing.

Para efetuar este ataque é necessário escolher que parte será fuzzed, e foi a password, e qual será a regra de fuzzing. No nosso caso optámos por utilizar REGEX para isto, tendo em conta que nos pareceu o mais simples de implementar, a regra utilizada foi “(admin[0-9]{3})”, que basicamente vai verificar todas as passes com admin e 3 digítos à frente, totalizando 1000 pedidos. Após isto esperámos até terminarem todos os pedidos, e procurámos se algum tinha retornado um código HTTP diferente de um da série dos códigos 4XX. Um deles, nomeadamente o com a password admin123, retornou 200 – OK, pelo que percebemos que era a password.

Por fim fomos ao site e testámos esse e-mail com essa password e conseguimos efetuar login.

Exercício 7

Nesta alínea, o objetivo era tentar colocar um comentário de Feedback com o user de outra pessoa, de forma a levar crer outros utilizadores que essa outra pessoa estava a fazer o comentário.

Para chegar a este objectivo, fizemos um comentário como anonymous, e analizámos com o ZAP, e percebemos que o campo de utilizador era apenas parte da string do comentário. Por isso tentámos substituir essa string para “admin” e executar o pedido, de forma a falsificar a identidade de admin.

Quando fizemos este pedido forjado, o sistema não verificou a string contra o userID que continuava a ser null, e foi colocado na roleta de comentários como sendo do admin. Assim sendo concluímos com uma mensagem do admin a dizer que não gostava do seu trabalho.

As provas para isto estão dentro no anexo do exercício 7.

Exercício 8

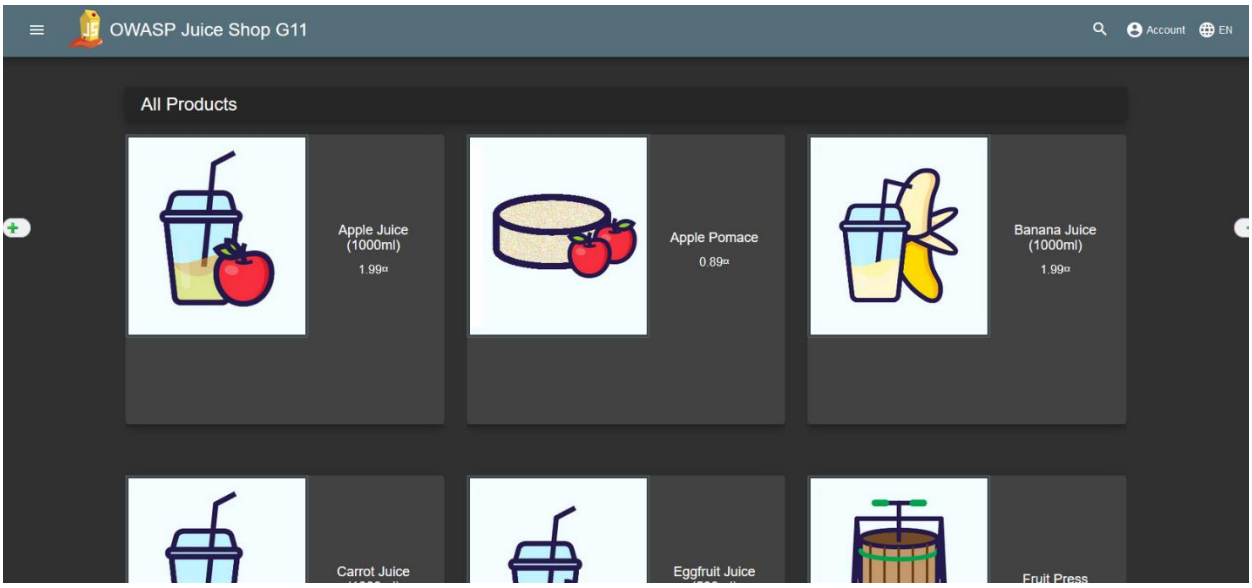
Para este exercício não conseguimos executar código através de um comentário com XSS, quando era suposto conseguir, pelo que a demonstração não foi possível. No entanto a forma que este ataque se executa, seria inserindo por exemplo um comentário num produto, com XSS que executava automaticamente e ia buscar a cookie ao browser de quem visse este comentário.

Após isto, seria necessário ter um servidor local à escuta num porto que apenas serviria para receber estas mensagens. Esse servidor local seria exclusivamente acedido pelo código XSS e enviaria os tokens de qualquer utilizador que visse este comentário.

A parte de engenharia social implicaria enviar o URL exclusivo deste produto e comentário para alguém, sob forma de um e-mail por exemplo, de forma a que esta pessoa clicasse, e sem saber enviasse o seu cookie de acesso (token) para o servidor privado do atacante.

Anexo

Exercício 0



Exercício 1

```
~ $ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
```

- Resultado Audit

=== npm audit security report ===

Run npm install express-jwt@6.0.0 to resolve 6 vulnerabilities

SEMVER WARNING: Recommended action is a potentially breaking change

Critical Verification Bypass

Package jsonwebtoken

Dependency of express-jwt

Path express-jwt > jsonwebtoken

More info https://npmjs.com/advisories/17

Moderate Regular Expression Denial of Service

Package moment

Dependency of express-jwt

Path express-jwt > jsonwebtoken > moment

More info https://npmjs.com/advisories/55

High Forgeable Public/Private Tokens

Package jws

Dependency of express-jwt

Path express-jwt > jsonwebtoken > jws

More info https://npmjs.com/advisories/88

Low Regular Expression Denial of Service

Package moment

Dependency of express-jwt

Path express-jwt > jsonwebtoken > moment

More info https://npmjs.com/advisories/532

Moderate Out-of-bounds Read

Package base64url

Dependency of express-jwt

Path express-jwt > jsonwebtoken > jws > base64url

More info https://npmjs.com/advisories/658

Moderate Out-of-bounds Read

Package base64url

Dependency of express-jwt

Path express-jwt > jsonwebtoken > jws > jwa > base64url

More info https://npmjs.com/advisories/658

Run npm install jsonwebtoken@8.5.1 to resolve 4 vulnerabilities

SEMVER WARNING: Recommended action is a potentially breaking change

Critical Verification Bypass

Package jsonwebtoken

Dependency of jsonwebtoken

Path jsonwebtoken

More info https://npmjs.com/advisories/17

High Forgeable Public/Private Tokens

Package jws

Dependency of jsonwebtoken

Path jsonwebtoken > jws

More info https://npmjs.com/advisories/88

Moderate Out-of-bounds Read

Package base64url

Dependency of jsonwebtoken

Path jsonwebtoken > jws > base64url

More info https://npmjs.com/advisories/658

Moderate Out-of-bounds Read

Package base64url

Dependency of jsonwebtoken

Path jsonwebtoken > jws > jwa > base64url

More info https://npmjs.com/advisories/658

Run npm install sanitize-html@2.1.2 to resolve 6 vulnerabilities

SEMVER WARNING: Recommended action is a potentially breaking change

Moderate Cross-Site Scripting

Package sanitize-html

Dependency of sanitize-html

Path sanitize-html

More info https://npmjs.com/advisories/135

Moderate Cross-Site Scripting

Package sanitize-html

Dependency of sanitize-html

Path sanitize-html

More info https://npmjs.com/advisories/154

Low Prototype Pollution

Package lodash

Dependency of sanitize-html

Path sanitize-html > lodash

More info https://npmjs.com/advisories/577

High Prototype Pollution

Package lodash

Dependency of sanitize-html

Path sanitize-html > lodash

More info https://npmjs.com/advisories/782

High Prototype Pollution

Package lodash

Dependency of sanitize-html

Path sanitize-html > lodash

More info https://npmjs.com/advisories/1065

Low Prototype Pollution

Package lodash

Dependency of sanitize-html

Path sanitize-html > lodash

More info https://npmjs.com/advisories/1523

Manual Review

Some vulnerabilities require your attention to resolve

Visit https://go.npm.me/audit-guide for additional guidance

Critical Command Injection

Package marsdb

Patched in No patch available

Dependency of marsdb

Path marsdb

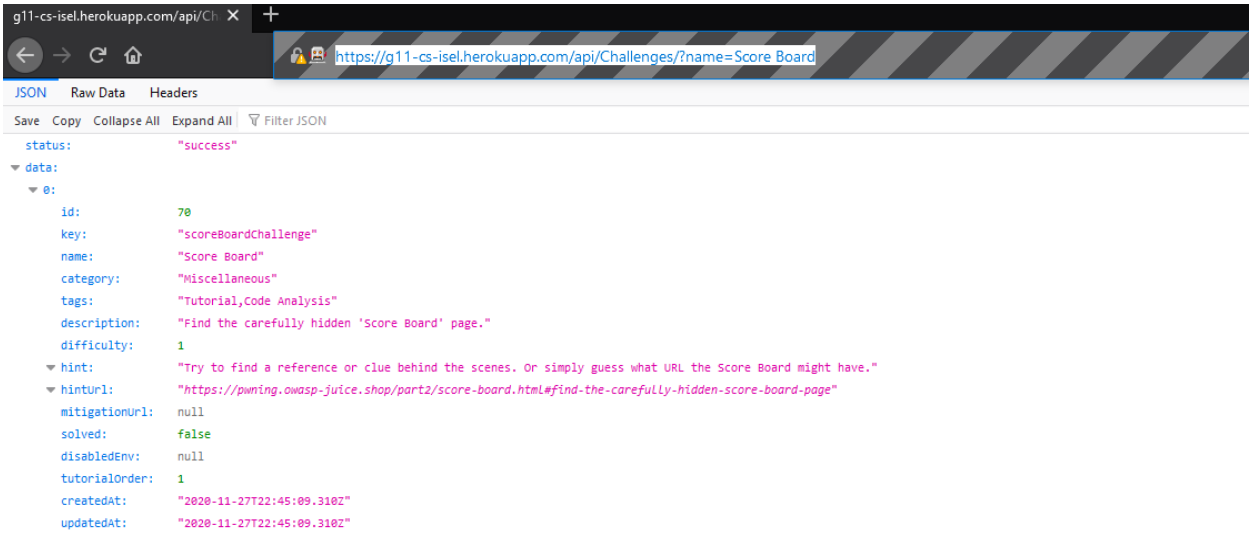
More info https://npmjs.com/advisories/1122

found 17 vulnerabilities (3 low, 7 moderate, 4 high, 3 critical) in 1032 scanned packages

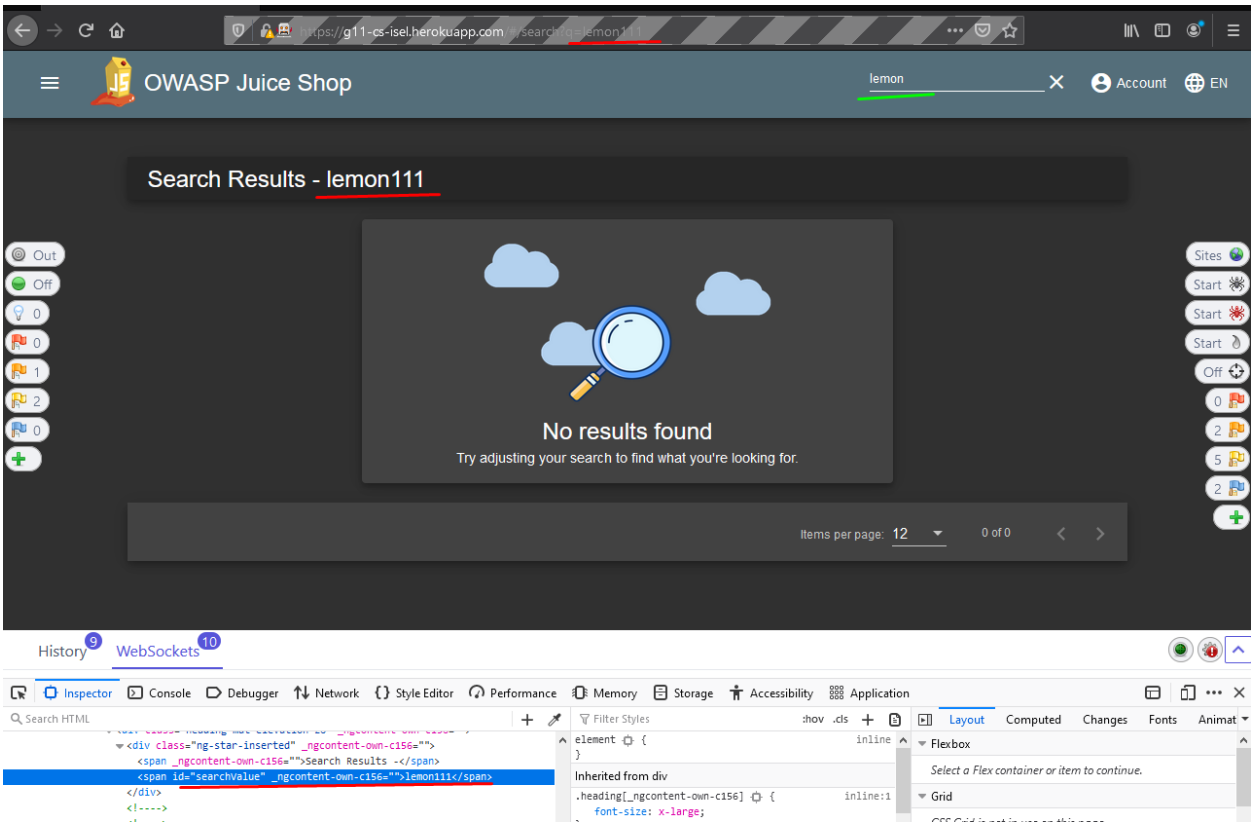
16 vulnerabilities require semver-major dependency updates.

1 vulnerability requires manual review. See the full report for details.

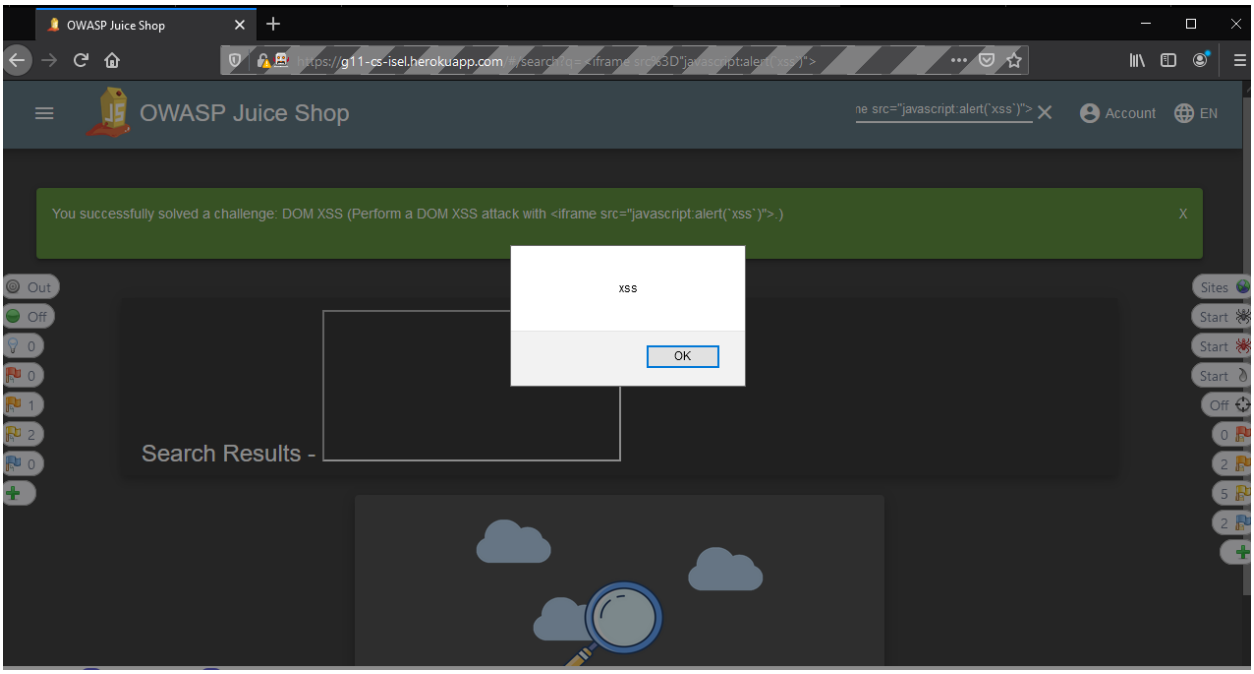
Exercício 3



Exercício 4

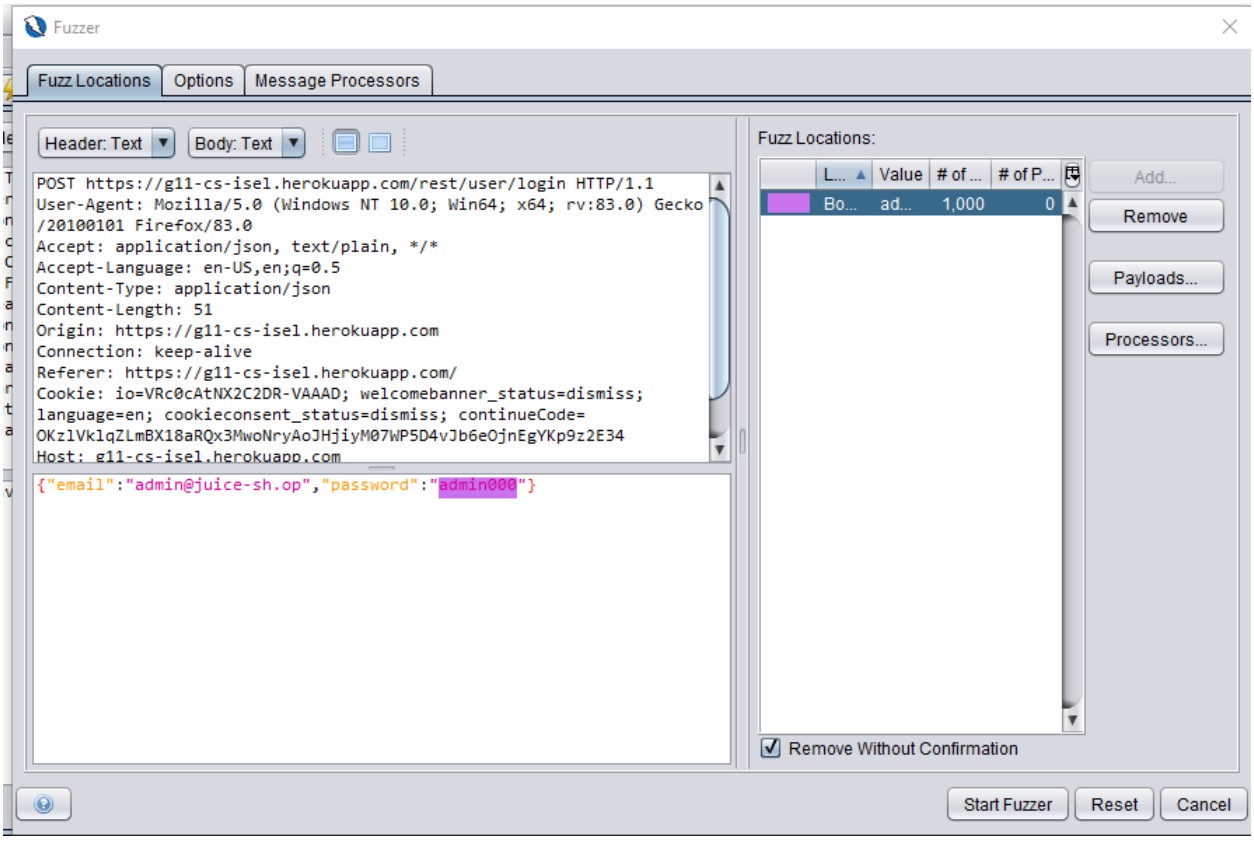


Exercício 5

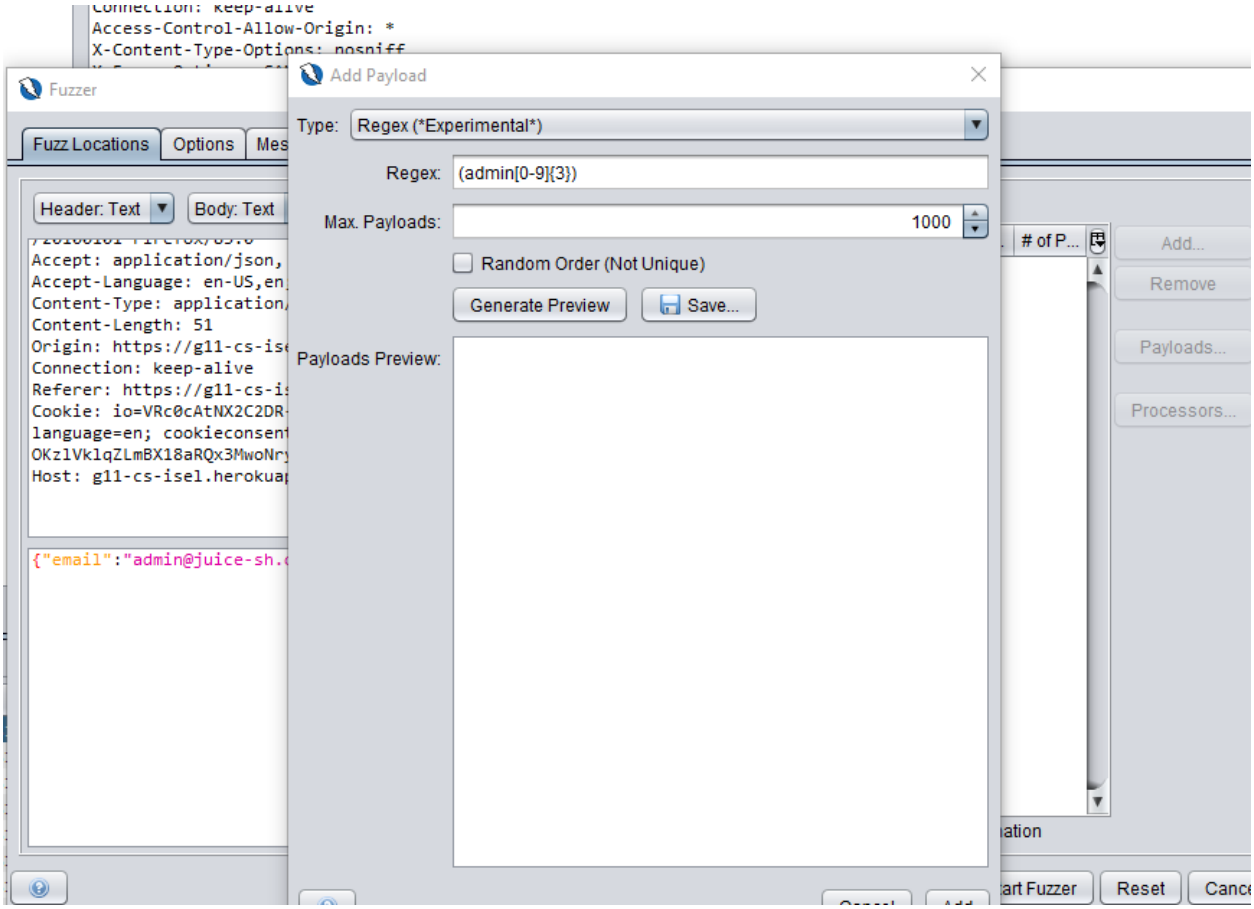


Exercício 6

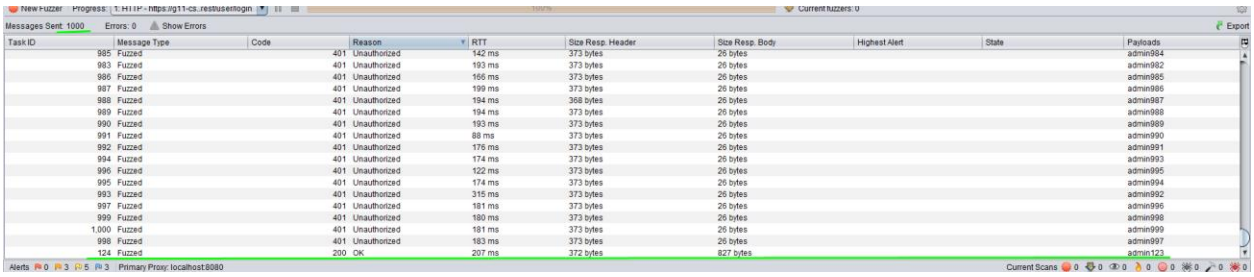
Pedido de teste



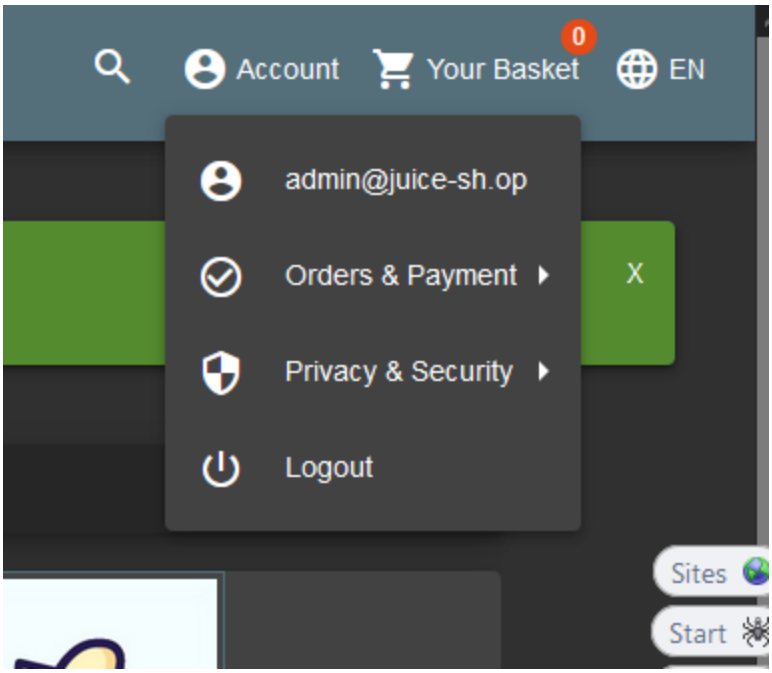
Regra de fuzzing em Regex



Attaque de Fuzzing



Prova do ataque



Exercício 7

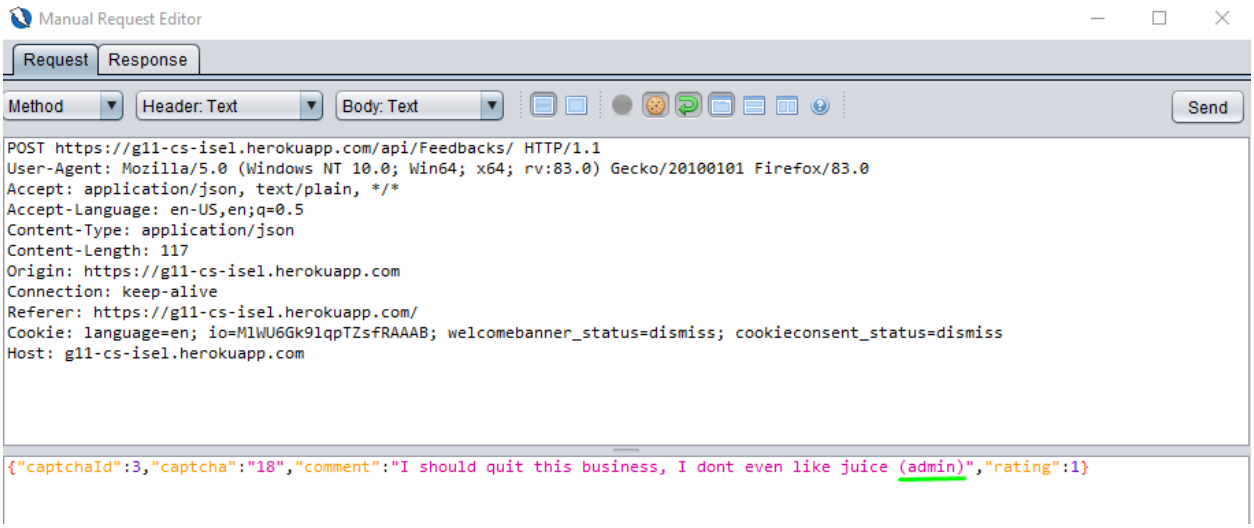
Pedido de teste com user anónimo

```
POST https://g11-cs-isel.herokuapp.com/api/Feedbacks/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Content-Length: 117
Origin: https://g11-cs-isel.herokuapp.com
Connection: keep-alive
Referer: https://g11-cs-isel.herokuapp.com/
Cookie: language=en; io=M1WU6Gk9lqpTZsfRAAAB; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Host: g11-cs-isel.herokuapp.com
```

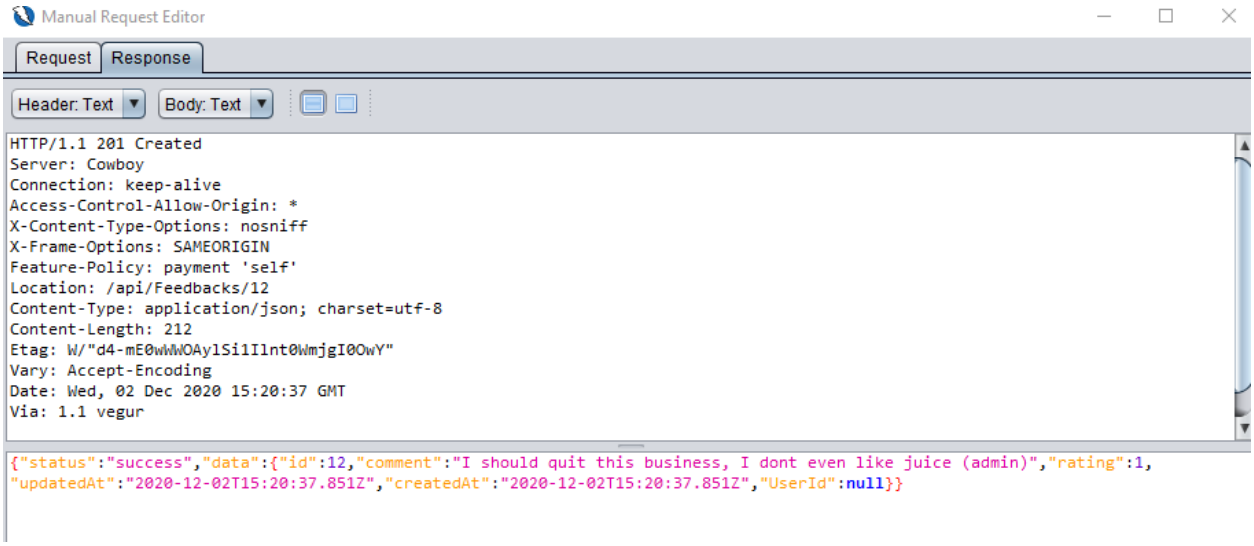
```
{"captchaId":3,"captcha":"18","comment":"The site looks quite good and not vulnerable at all (anonymous)","rating":4}
```

	Code	Reason	RTT	Size Resp. Body	Highest Alert
sel.herokuapp.com/rest/captcha/	200	OK	649 ms	47 bytes	Medium
sel.herokuapp.com/rest/user/whoami	304	Not Modified	72 ms	0 bytes	Medium
sel.herokuapp.com/api/Feedbacks/	201	Created	372 ms	216 bytes	Medium

Pedido falsificado com admin como user



Resposta ao pedido falsificado



Prova do pedido falsificado