S5L6,

BENVENUTI LUIGI

Traccia:

Nell'esercizio di oggi, viene richiesto di exploitare le vulnerabilità:

- XSS stored.

- SQL injection (blind). Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW.

Scopo dell'esercizio:

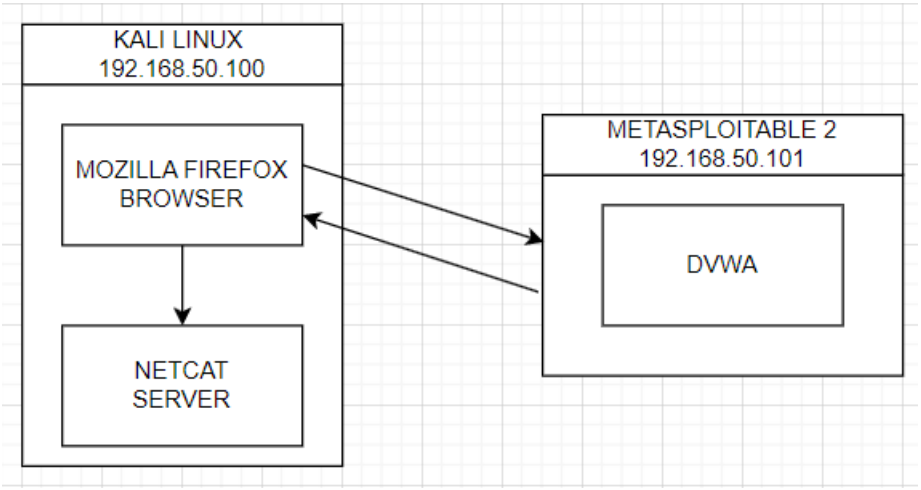
1 - Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

2 - Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi). Agli studenti verranno richieste le evidenze degli attacchi andati a buon fine.

Fase di exploiting - 1: XSS Stored vulnerability

Settiamo il nostro laboratorio virtuale in modo che la macchina Kali si comporti sia da host del browser attaccante sia da server in ascolto sul quale viene ricevuta la richiesta GET contenente il cookie di sessione da rubare; al contempo, la macchina Metasploitable 2 prende la parte della vittima.

Di seguito uno schema riepilogativo:



Dopo aver controllato che le 2 macchine siano effettivamente in connessione, apriamo il browser in Kali e digitiamo nella barra degli indirizzi l'URL http://192.168.50.1017dvwa/.

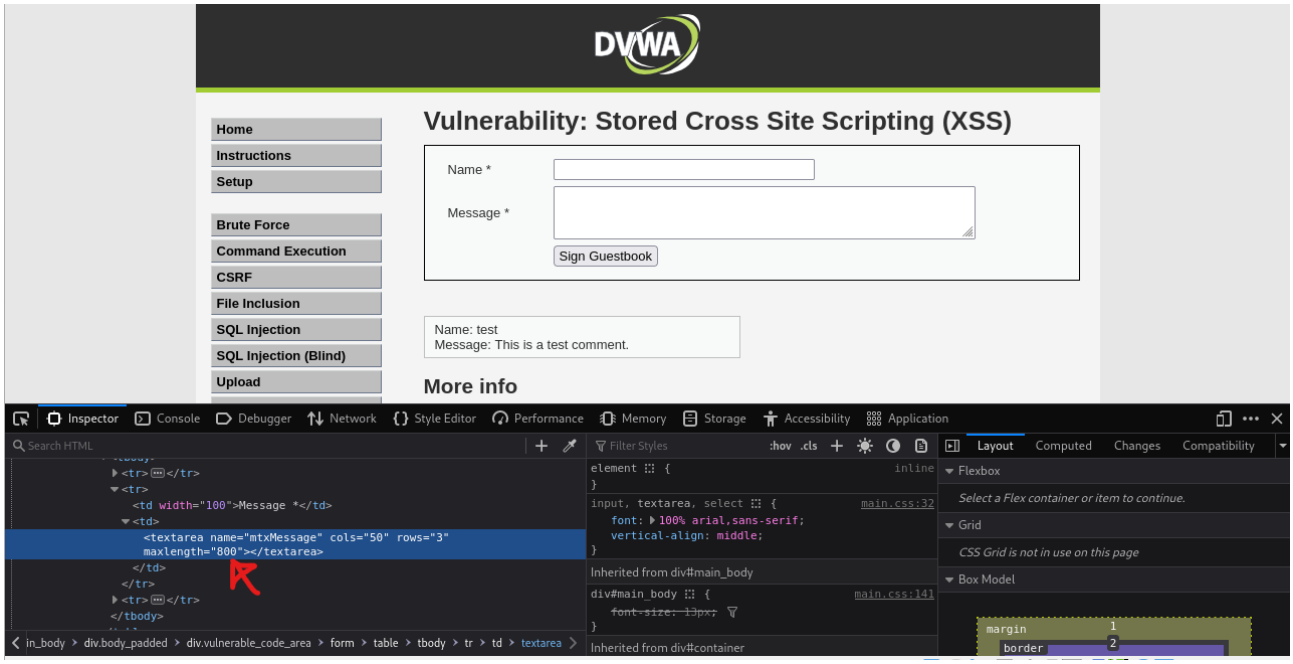
N.B. PER TUTTO L'ESERCIZIO IL SECURITY LEVEL DELLA MACCHINA DVWA DOVRA' ESSERE SETTATO SU LOW

Inseriamo le credenziali di accesso e procediamo nel path dedicato alla XSS stored.

Una vulnerabilità di tipo XSS Stored dà la possibilità ad un attaccante di salvare un payload malevolo all'interno di una pagina web: è piuttosto pericoloso, perché chiunque accederà a quell'URL è una vittima.

Notiamo subito come nei campi "Name" e "Message" sia presente un controllo che non permette di inserire più di un tot di caratteri.

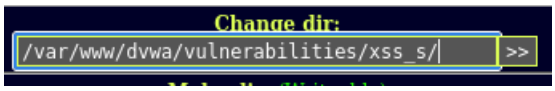
Si potrebbe aggirare questo controllo semplicemente modificando il codice HTML ed aumentando la quantità di caratteri disponibili, come in figura:



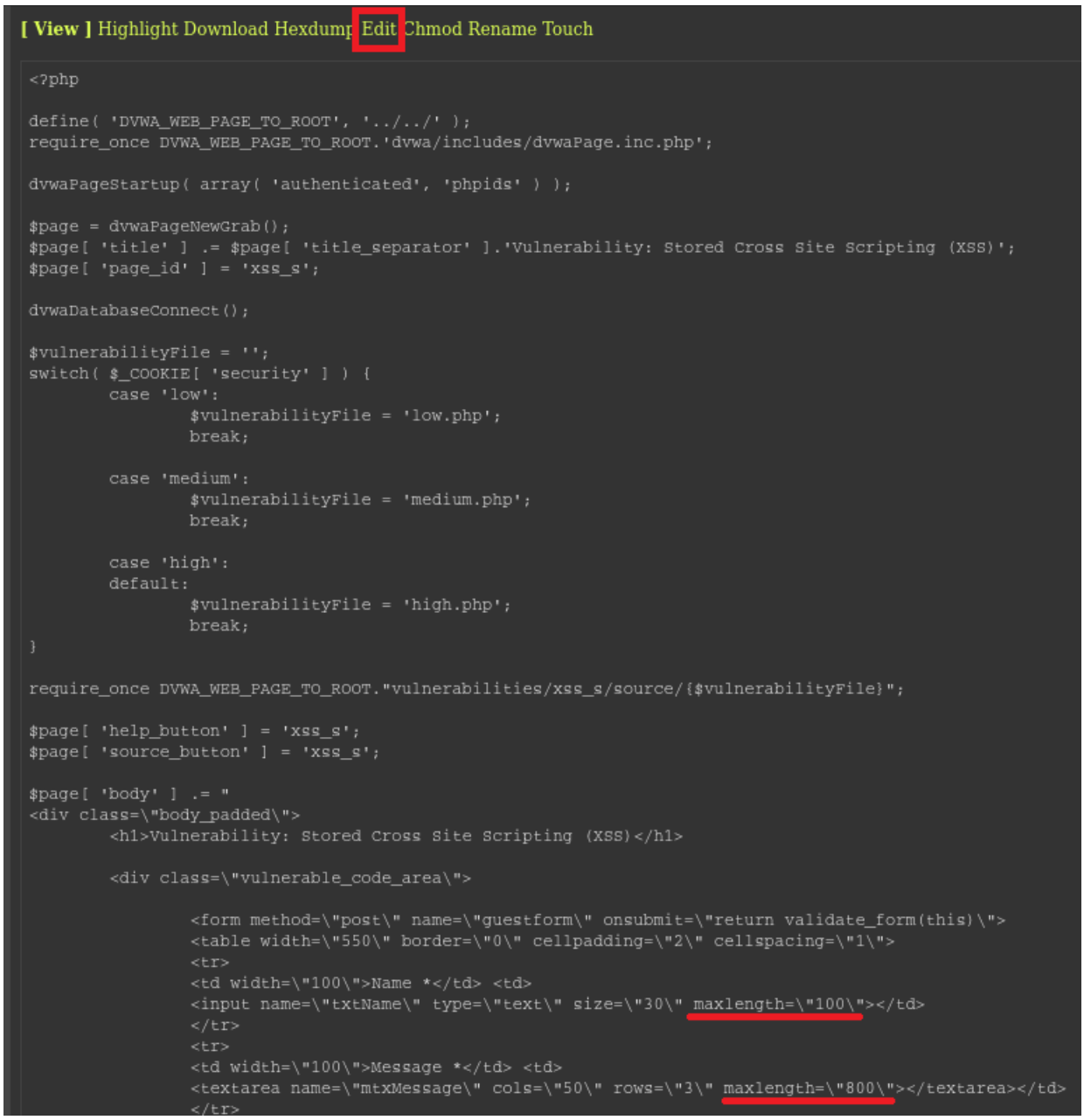
Tuttavia, questa sarebbe una modifica temporanea, fino al nuovo caricamento della pagina.

Andiamo dunque a vedere un possibile modo per renderla effettiva; andiamo nel path file upload della DVWA, e sfruttiamo la debolezza presente in questa pagina caricando una shell in PHP malevola in grado di accedere alla configurazione della pagina web.

Selezioniamo il path in cui risiede l'index della pagina XSS Stored:



E ne modifichiamo permanentemente il codice sorgente sfruttando il tool presente nella shell caricata:

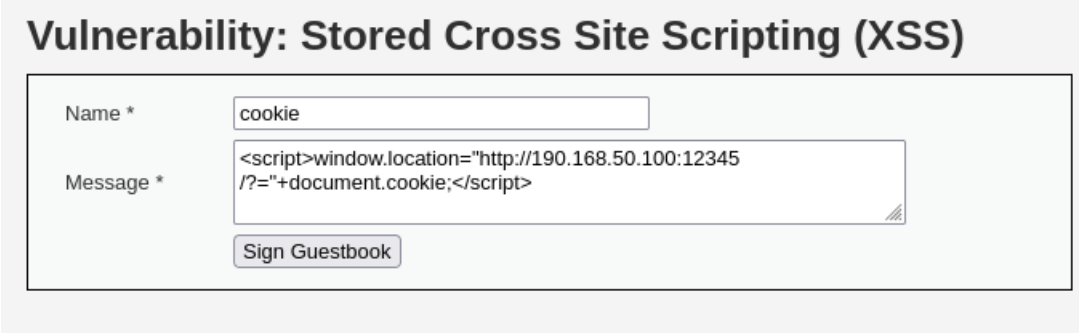


Attiviamo intanto su un terminale di kali un servizio Netcat sulla porta 12345 con il comando seguente:

nc -l 192.168.50.100 -p 12345

Ci servirà successivamente per catturare la richiesta GET contenente il cookie di sessione.

A questo punto possiamo tornare nella sezione XSS Stored della DVWA e inserire lo script malevolo all'interno del form: nel nostro caso, proveremo a rubare il cookie di sessione della macchina Metasploitable.



Premiamo "Sign Guestbook" per salvare sulla pagina questo script.

Dopo il salvataggio, la pagina verrà ricaricata e si avvierà l'esecuzione del codice malevolo appena inserito.

Troveremo la richiesta GET in chiaro sul server netcat in ascolto, come di seguito:

Fase di exploit – 2: SQL Injection Blind

Per sfruttare la seconda debolezza, ci spostiamo sul path SQL injection blind della DVWA.

Una vulnerabilità SQL Injection permette ad un attaccante di sfruttare una query dinamica non correttamente configurata per inserire a sua volta delle query malevole in grado di garantire l'accesso non autorizzato a dati sensibili di un database.

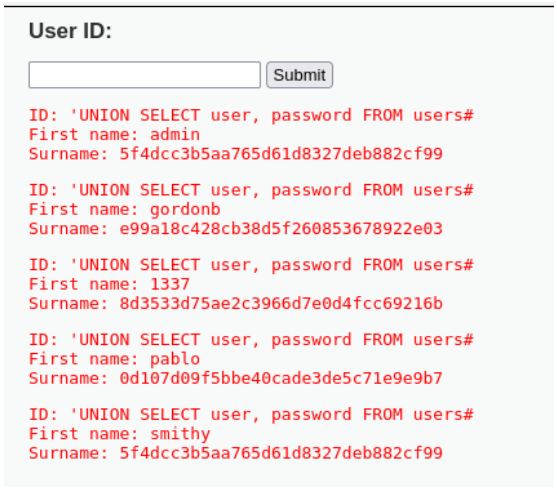
La differenza tra un SQLi semplice ed un SQLi Blind sta nel fatto che la seconda non ritorna eventuali errori.

Nel nostro caso, recupereremo le password contenute nel database "users" a cui fa riferimento la DVWA.

Per fare ciò, inseriamo nel form la seguente query:

'UNION SELECT user, password FROM users#

Di seguito l'output:



Possiamo notare come le password siano state criptate tramite un algoritmo di hashing MD5, ovvero un algoritmo di cifratura a senso unico grazie quale non si può in alcun modo risalire direttamente alla password originale.

Possiamo utilizzare lo strumento John the Ripper per effettuare un attacco a dizionario tramite una wordlist di parole conosciute e cercare di trovare delle corrispondenze con le password in formato hash.

Possiamo affermare che se una stringa, dopo essere stata sottoposta allo stesso algoritmo di hash, ritorna un valore uguale a quello delle nostre password, ci sia corrispondenza.

Il comando di John the Ripper che ci permette di fare questo è:

john –-wordlist=/usr/share/wordlists/rockyou.txt –format=raw-md5 ./hash.txt

Dove il primo parametro indica la wordlist da utilizzare, il secondo parametro indica il formato di hash in cui ci si aspetta di trovare corrispondenza ed il terzo parametro un file contente le nostre password in hash.

I risultati che il tool ci restituisce sono mostrati in figura: