Report XSS reflected

Cross-Site Scripting (XSS) è una vulnerabilità delle applicazioni web in cui un attaccante inserisce codice JavaScript malevolo all'interno di un'applicazione web e questo codice viene poi eseguito nei browser degli utenti finali quando visitano la pagina infetta. XSS reflected è uno dei tipi più comuni di attacchi XSS ed è chiamato "riflesso" perche il payload dannoso è "riflesso" dal server web all'utente attraverso un'interazione diretta, come un clic su un link o una richiesta http.

Vulnerabilità nell'applicazione

L'attacco XSS reflected inizia con una vulnerabilità nell'applicazione web. Questa vulnerabilità si verifica quando l'applicazione non sanifica correttamente l'input fornito dagli utenti prima di restituirli in una pagina web.

Creazione del payload

L’attaccante sfrutta questa vulnerabilità inserendo un payload di scripting malevolo nell’applicazione. Un payload XSS può essere costituito da un codice JavaScript o da altri marcatori HTML.

Un esempio di payload potrebbe essere:

‘<script>alert(‘XSS Attack !’);</script>’

Inserimento dei payload nell’applicazione

L’attaccante può inserire il payload nell’applicazione attraverso vari mezzi, come un modulo input, una richiesta URL, un campo di ricerca o qualsiasi altro punto in cui l’applicazione accetta dati dall’utente.

Esecuzione del payload

Una volta che il payload è stato inserito nell'applicazione un utente legittimo interagisce con l'applicazione, il payload viene restituito dal server web insieme alla pagina web generata. Quando il browser dell'utente riceve questa pagina, eseguirà automaticamente il codice JavaScript incluso nel payload.

Impatto

L'attacco XSS può avere vari impatti negativi, tra cui il furto di cookie di sessione, il reindirizzamento a pagina malevole, la visualizzazione di contenuti dannosi, la manipolazione dei dati dell'utente o altri comportamenti indesiderati.

Per prevenire gli attacchi XSS bisogna seguire alcuni passaggi :

1) Validare e sanificare rigorosamente tutti i dati di ingresso
2) Utilizzare librerie di sicurezza o framework che offrono protezione automatica contro XSS
3) Implementare l'http Only e il flag Secure sui cookie per prevenire i furti di sessione
4) Impostare correttamente gli header http, come Content Security Policy (CSP) per mitigare gli attacchi XSS
5) Educare gli sviluppatori e gli utenti sulla sicurezza delle applicazioni web