questione viene caricata. Quest'attacco è stato sfruttato ampiamente nel recente passato, soprattutto laddove veniva consentito agli utenti di inserire recensioni, commenti e altri contributi.
Volendo schematizzare, possiamo categorizzare gli attacchi XSS nelle seguenti tipologie:

- Reflected XSS, in cui la stringa dannosa proviene dalla richiesta dell'utente.
- Stored XSS, in cui la stringa dannosa proviene dal database del sito Web.
- DOM based XSS, in cui la vulnerabilità è nel codice lato client anziché nel codice lato server.

Poiché Javascript è molto potente e flessibile, un sito sotto attacco mette in pericolo le proprie informazioni e quelle degli utenti che vi si collegano.
I danni del Cross Site Scripting possono essere esemplificati schematicamente come segue:

Furto di cookie

L'aggressore può accedere ai cookie associati al sito Web bersaglio, inviarli al proprio server e utilizzarli per estrarre informazioni riservate come gli ID di sessione.

Keylogging

L'aggressore può registrare un listener che registri tutti gli di eventi provenienti dalla tastiera e quindi inviare tutte le sequenze di tasti dell'utente al proprio server. In tal modo può entrare in possesso di informazioni potenzialmente sensibili come password e numeri di carta di credito.

Phishing

Utilizzando la manipolazione DOM , l'autore dell'attacco può far comparire sulla pagina un modulo di accesso falso e indurre l'utente a inviare informazioni riservate al proprio server.

Come difendersi

Le seguenti contromisure sono efficaci per evitare che gli attacchi XSS riescano nel loro intento:

- Encoding (codifica), che opera l'escaping all'input dell'utente in modo che il browser lo interpreti solo come testo, non come codice. Si tratta di filtrare i caratteri specifici dei tag HTML e della codifica Javascript, sostituendoli con del testo.
- Validation (convalida), che controlla nel merito l'input dell'utente, valutando che risponda a determinati criteri attesi.
- CSP. Oltre a questi rimedi, è necessario attivare lo standard Content Security Policy (CSP) in modo che solo le risorse scaricate da fonti attendibili possano essere utilizzate. Per risorsa s'intende qui uno script, un foglio di stile, un'immagine o altri tipi di file trattati nella pagina. Ciò significa che anche se un utente malintenzionato riesce a iniettare contenuti dannosi nel sito Web, CSP può impedirne l'esecuzione.
- Impostare il flag HttpOnly a true, per evitare tentativi di furto tramite la lettura, tramite script, dei cookie di sessione.

7.4.2 Client DOM Code Injection

Come riconoscerla

Un attaccante può eseguire codice arbitrario sulla macchina dell'application server. A seconda dei permessi di cui dispone l'applicazione, potrebbe: accedere al database, leggere o modificare dati sensibili; leggere, creare, modificare o cancellare file; aprire una connessione al server dell'attaccante; modificare il contenuto delle pagine; decifrare dati utilizzando le chiavi dell'applicazione; arrestare o avviare i servizi del sistema operativo; organizzare un reindirizzamento verso siti fake (fasulli) per operazioni di phishing; prendere il completo controllo del server.
Accade perché l'applicazione esegue alcune azioni eseguendo codice incluso nei dati in input non opportunamente validati e verificati. In questo caso, il codice non attendibile viene letto dal browser ed eseguito sul lato client.

Come difendersi

- Come prima cosa, l'applicazione non dovrebbe eseguire alcun codice non attendibile da qualsiasi fonte esterna possa provenire, inclusi l'input dell'utente, dei file caricati (upload) o un database.