7.4.5 Client DOM XSS

Come riconoscerla

Un utente malintenzionato può utilizzare il social engineering per indurre un utente a inviare l'input modificato in modo malevolo verso il sito Web, ad esempio inducendolo a cliccare su un URL con un'ancora (hash) modificata, facendo sì che il browser riscriva le pagine Web. L'aggressore può quindi dirottare la vittima verso un server fake (fasullo), che gli consentirebbe di rubare la password dell'utente, farsi inserire i dati della carta di credito, fornire informazioni false o eseguire del malware. Ovviamente la vittima rimane ignara di ciò che accade.
L'attacco è possibile perché la pagina Web dell'applicazione incorpora nella pagina dati provenienti dall'input dell'utente (incluso l'URL della pagina), facendo sì che il browser li visualizzi come parte della pagina Web. Se l'input include frammenti HTML o JavaScript, anche questi vengono visualizzati (ed eseguiti). La vulnerabilità è il risultato dell'incorporamento di input dell'utente arbitrario senza prima codificarlo in un formato che impedirebbe al browser di trattarlo come HTML anziché come testo normale.

Come difendersi

- I parametri devono essere limitati a un set di caratteri consentito e l'input non convalidato deve essere eliminato. Oltre ai caratteri, occorre controllare il tipo di dati, la loro dimensione, l'intervallo di validità, il formato e l'eventuale corrispondenza all'interno dei valori previsti (white list). Sconsigliata invece la black list, ossia un elenco di valori non consentiti: l'elenco sarebbe sempre troppo limitato, rispetto ai casi che potrebbero verificarsi.
- Effettuare un encoding (codifica) su tutti i dati dinamici prima di includerli nella pagina web. Considerare per tale scopo la libreria ESAPI4JS di OWASP.

Esempio:
codice vulnerabile:

```
document.write("Il sito si trova qui: " + document.location);
```

codice sicuro:

```
document.write("Il sito si trova qui: " +
        ESAPI4JS.encodeForURL(document.location));
```

Per maggiori informazioni vedere: http://cwe.mitre.org/data/definitions/79.html

7.5 Python

Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti, adatto, tra l'altro, per sviluppare applicazioni distribuite, scripting, applicazioni web, applicazioni di computazione numerica e di system testing.
Fu sviluppato da Guido van Rossum nel peridodo 1985-1990 come Open Source, sotto licenza GNU General Public License (GPL).
Dato il grande successo e la diffusione del linguaggio, sono sosrti numerosi framework e librerie che ne aumentanto le potenzialità, sia in termini di caratteristiche, che di prestazioni.
Di seguito, un elenco delle principali vulnerabilità alle quali i programmi Python possono essere soggetti e le contromisure da adottare per mitigarle.

7.5.1 Cross-site scripting (XSS)

Come riconoscerla

Il Cross Site Scripting consiste nella possibilità di inoculare uno script e di mandarlo in esecuzione sul front-end dell'applicazione. Tramite tecniche sviluppate da malintenzionati per ottenere informazioni personali, possono, ad esempio, essere simulate pagine quasi identiche ad altri siti molto frequentati per ottenere informazioni riservate. La prassi del "social engineering" consente di ingannare gli utenti per indurli a visitare pagine fraudolente. Gli attacchi XSS di tipo reflected si verificano ogni qualvolta uno script viene