Sviluppo di software sanitario sicuro

CSP009

PRESENTAZIONE DA PARTE DI: STYLIANOS KARAGIANNIS (PDMFC, PORTOGALLO)

Incomplete String Escaping or Encoding

Problema di codice sul sistema PACS

Escaping o codifica della stringa incompleta

- Posizione: Trovato in jquery.mobile.simpledialog2.js alla riga 137.
- Problema: L'escape incompleto delle stringhe, che riguarda in particolare solo la prima occorrenza di ", lascia potenzialmente le altre istanze vulnerabili agli attacchi di iniezione.
- Strumento utilizzato: CodeQL ha rilevato questo problema.
- Raccomandazione: Assicurare l'escape o la codifica completa delle stringhe per tutte le occorrenze di caratteri speciali, per prevenire attacchi di tipo SQL injection o cross-site scripting (XSS).
- Esempio: Dimostra un escape incompleto in cui solo la prima occorrenza di " viene trattata, lasciando vulnerabili le occorrenze successive.
- Riferimenti: CWE-20, CWE-80, CWE-116 evidenziano le debolezze associate a un escape incompleto delle stringhe e l'importanza di una sanitizzazione completa per la sicurezza.

Problema di codice sul sistema PACS

Costruzione di HTML non sicuro da input di libreria

- Posizione: Identificato in jquery.blockui.js alla riga 253.
- Problema: La costruzione di HTML si basa su input potenzialmente non sicuri, che possono portare a vulnerabilità cross-site scripting (XSS).
- Strumento utilizzato: CodeQL ha segnalato questo problema.
- Raccomandazione: Documentare che la funzione deve essere utilizzata solo con input affidabili per evitare l'iniezione involontaria di frammenti HTML non sicuri.
- Esempio: Mostra la costruzione di HTML dipendente dall'input, potenzialmente esponendo l'applicazione ad attacchi XSS se vengono utilizzati input non sicuri.
- Riferimenti: CWE-79 e CWE-116 evidenziano i rischi associati alla costruzione di HTML non sicuro e l'importanza della convalida degli input per mitigare le vulnerabilità XSS.

Codice del sistema informativo sanitario

Attacchi cross-site scripting

- Posizione: Rilevato in jqModal.js alla riga 48.
- Problema: Costruzione di HTML dinamico senza un'adeguata sanificazione dell'input dell'utente (c.ajaxText), che può portare ad attacchi cross-site scripting (XSS).
- Strumento utilizzato: CodeQL ha identificato questa vulnerabilità.
- Raccomandazione: Documentate tutti gli input suscettibili di attacchi XSS e implementate solide misure di sanitizzazione degli input per ridurre i rischi.
- Esempio: Dimostra un approccio più sicuro utilizzando il metodo .find() di jQuery per la manipolazione dell'HTML.
- Riferimenti: OWASP e CWE forniscono indicazioni sulla prevenzione degli XSS e sull'importanza delle pratiche di sviluppo sicuro dei plugin jQuery.

Grazie

Presentatore: Stylianos Karagiannis (PDMFC, Portogallo)

Si prega di inviare tutte le domande a:
stylianos.karagiannis@pdmfc.com