Finding 2 — Reflected XSS (Search field)

Location / Endpoint: Search input on main page

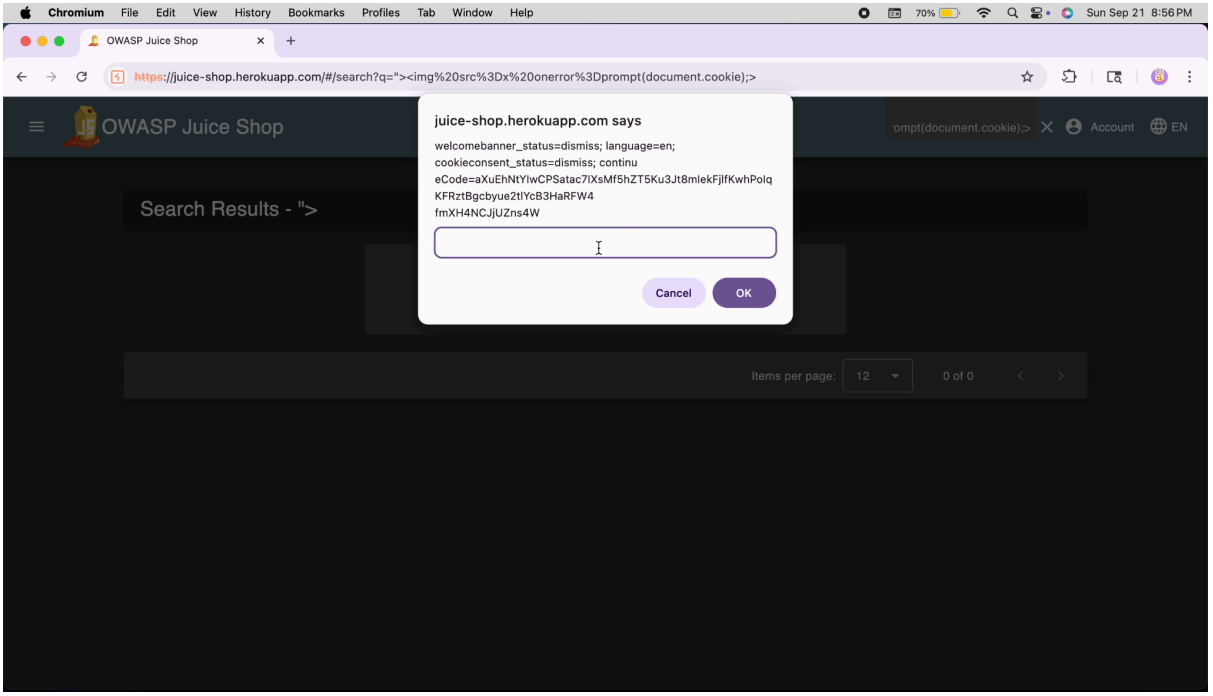
- **Payload used:**

"><img src=x onerror=prompt(document.cookie);>

Steps to reproduce:

1. Open Juice Shop → locate search box.
2. Paste payload and submit.
3. A JavaScript prompt appears showing `document.cookie` (proof that cookie string was accessible to injected JS).

Proof / PoC:



- **Impact (Severity: High):**
  Because the payload returned the cookie value in the prompt, an attacker can exfiltrate session cookies or other sensitive client-side data. This can lead to session hijacking, account takeover, persistent phishing (if combined with social engineering), or execution of any JS in the victim's session context. Where cookies contain auth tokens, this is effectively account compromise.
- **Recommendation / Mitigation:**
  - Properly escape/encode output before rendering in HTML contexts (HTML entity encoding).
  - Enforce strict Content Security Policy (CSP) to restrict inline scripts and untrusted sources.

- Sanitize input on the server and use safe templating frameworks that auto-escape.
- Set `HttpOnly` on session cookies to reduce the impact of client-side XSS when possible (but note `HttpOnly`alone does not fully mitigate reflected XSS effects on other secrets).