**CTF: JavaScript Vulnerability Detection Using "JS Vulnerability Detector" Extension**

**Introduction**

"JS Vulnerability Detector" is a Chrome extension designed to detect potential vulnerabilities in JavaScript code on webpages. This Capture the Flag (CTF) scenario introduces beginners to the basics of using the "JS Vulnerability Detector" extension to analyze JavaScript code and identify vulnerabilities on a given webpage.

**Tool and Requirements**

* JS Vulnerability Detector Chrome Extension: An extension for Google Chrome that detects potential vulnerabilities in JavaScript code.
* JS Vulnerability Detector Chrome Extension:

<https://chromewebstore.google.com/detail/js-vulnerability-detector/bmcojnncgfmglejiinbdnahmkmbgifhk?hl=en>

**Scenario**

In this scenario, you will use the "JS Vulnerability Detector" extension to analyze the JavaScript code on a specific webpage. Your tasks include identifying potential vulnerabilities, understanding their nature, and capturing flags based on the analysis.

**Let’s Begin**

To start, install the "JS Vulnerability Detector" extension from the Chrome Web Store. Use this extension to analyze the JavaScript code on the provided webpage and solve the challenges to capture the flags.

**Questions**

**Flag 1: Identify Cross-Site Scripting (XSS) Vulnerability**

**Task: Use the "JS Vulnerability Detector" extension to identify a potential Cross-Site Scripting (XSS) vulnerability in the JavaScript code on the provided webpage.**

URL: <http://testphp.vulnweb.com/>

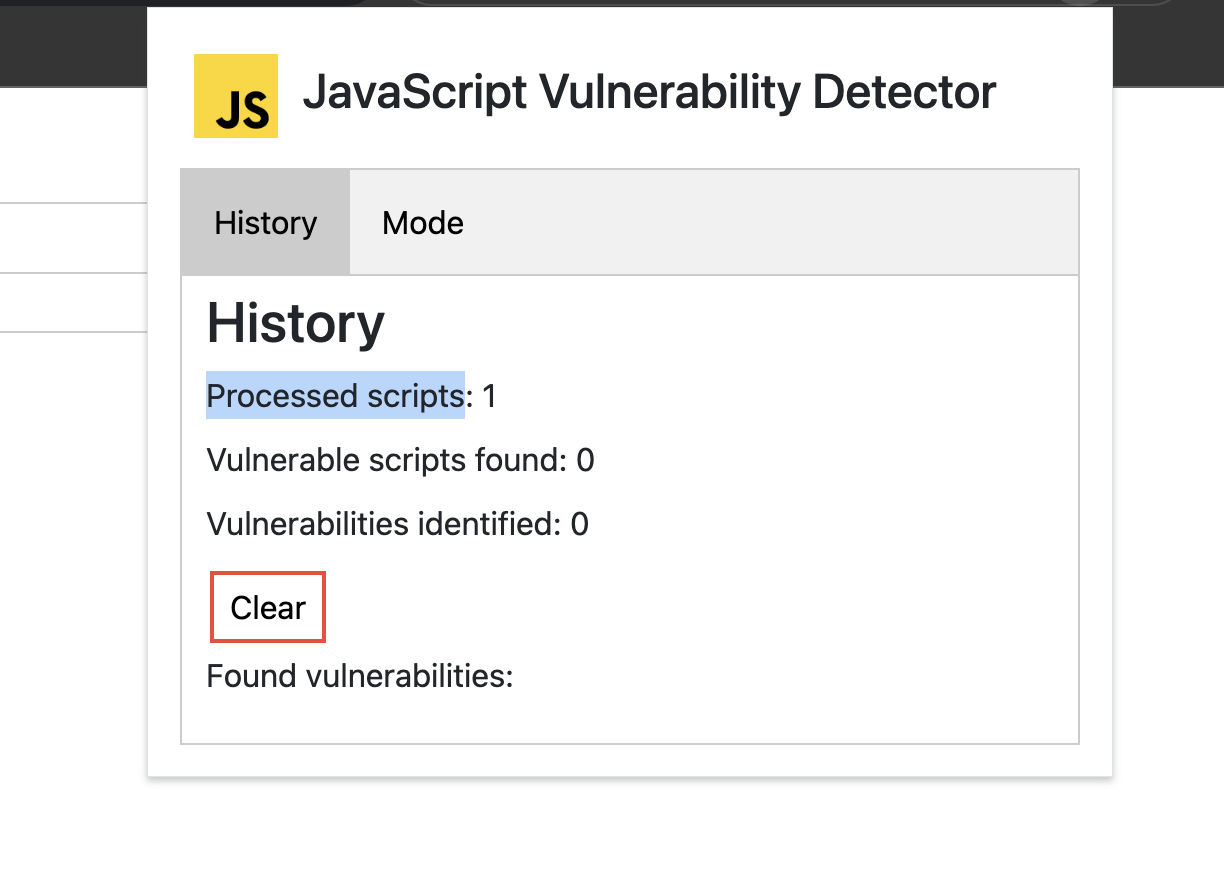
Question: What is the potential XSS vulnerability found in the JavaScript code?

Hint: Use the extension icon to detect XSS vulnerabilities.

Answer: Processed scripts

Flag Captured.

**Hint**

Use the documentation and help provided by the "JS Vulnerability Detector" extension to understand how to detect potential vulnerabilities in JavaScript code. Look for common vulnerability patterns such as XSS, injection, CSRF, and unvalidated redirects to help identify.