Vulnerability: Stored Cross-Site Scripting (XSS)

CVSS v3.1 Severity Rating

Metric	Value
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	Required
Scope	Unchanged
Confidentiality Impact	Low
Integrity Impact	Low
Availability Impact	None

CVSS Base Score: `6.1 (Medium)`
CVSS Calculator – v3.1

Fix Recommendations

1. Output Encoding

Escape HTML characters in user input using functions like:

```
php
CopyEdit
htmlentities($input, ENT_QUOTES, 'UTF-8');
```

2. Input Validation & Sanitization

Block or clean dangerous characters like <, >, ", ', and script tags at input level.

3. Implement Content Security Policy (CSP)

Add headers to prevent unauthorized script execution:

```
http
CopyEdit
Content-Security-Policy: default-src 'self';
```

4. Use Secure Frameworks

Choose frameworks that auto-sanitize input/output like:

- Django (Python)
- Laravel (PHP)
- Express.js with Helmet middleware (Node.js)

5. Set HTTP-Only & Secure Cookie Flags

Prevent JavaScript access to cookies using:

```
http
CopyEdit
Set-Cookie: sessionID=xyz; HttpOnly; Secure;
```

Impact Summary

A stored XSS vulnerability allows an attacker to inject malicious JavaScript into the application. The script is **stored on the server** and executed in the browser of any user who visits the infected page.

Potential threats:

- Session hijacking
- Credential theft
- Defacement
- Phishing via redirection