Cross-Site Scripting (XSS) Prevention in PHP

1. Cross-site scripting vulnerabilities occur when web applications allow users to add custom code into a url path or onto a website that will be seen by other users. This vulnerability can be exploited to run malicious JavaScript code on a victim's browser. For example, an attacker could send an email to a victim that appears to be from a trusted bank, with a link to that bank's website. This link could have some malicious JavaScript code tagged onto the end of the url. If the bank's site is not properly protected against cross-site scripting, then that malicious code will be run in the victim's web browser when he clicks on the link.

2. Another example: It can also be performed by submitting scripts by client instead of normal text in html input forms, then admin or moderators may review these texts and execute these scripts unwillingly resulting in leakage of important information such as login session cookie, so attacker may get his hands on the admin account using this attack, unless filters and encoders are applied to the inputs in the backend side.

Prevention:

Mitigation strategies for cross-site scripting include escaping untrusted HTTP requests as well as validating and/or sanitizing user-generated content. Using encoding or escaping functions or modern web development frameworks which provides some built-in cross-site scripting protection.

Make sure you filter or escape all user inputs for XSS attacks, to do that, you can use:

1- htmlentities() and htmlspecialchars()
Example:

```
<?php
      $input = htmlentities(htmlspecialchars($_GET[“user_input”]));
?>
```

2- Or The best User Input Filter is htmlpurifier library, you can read and download it from http://htmlpurifier.org/ to apply it in your project.

3- Or if you are using a PHP framework, the framework itself should have its own XSS input filtering you can use.

4- Don't use '.innerHTML' property in javascript, it can be injected, use '.innerText' instead.

5- Don't use eval() function in javascript, needing to use eval() usually indicates a problem in your code design.

All Above Explained ideas will be Scanned in the Toolkit.

To learn more, go here:
https://www.youtube.com/watch?v=IuzU4y-UjLw
https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS)