Cross Site Scripting(XSS)

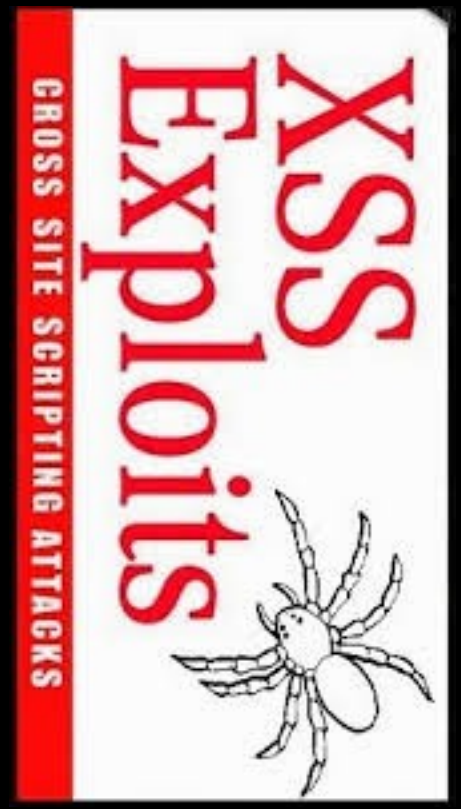
What is XSS?

How it works?

How is it done?

Detect if you are vulnerable?

Prevention



What is XSS?

- Type of injection in which malicious scripts are injected into the site
- this occurs when data is included to dynamic content without validation
- Not only server is vulnerable but, the user of the web application are vulnerable
- Attacker could retrieve cookies, session tokens, other sensitive information from the user’s browser, false advertising, changing uses contents, phishing attacks
- Types:
  - Reflected XSS / Non-Persistent
  - Stored XSS / Persistent
  - DOM Bases XSS

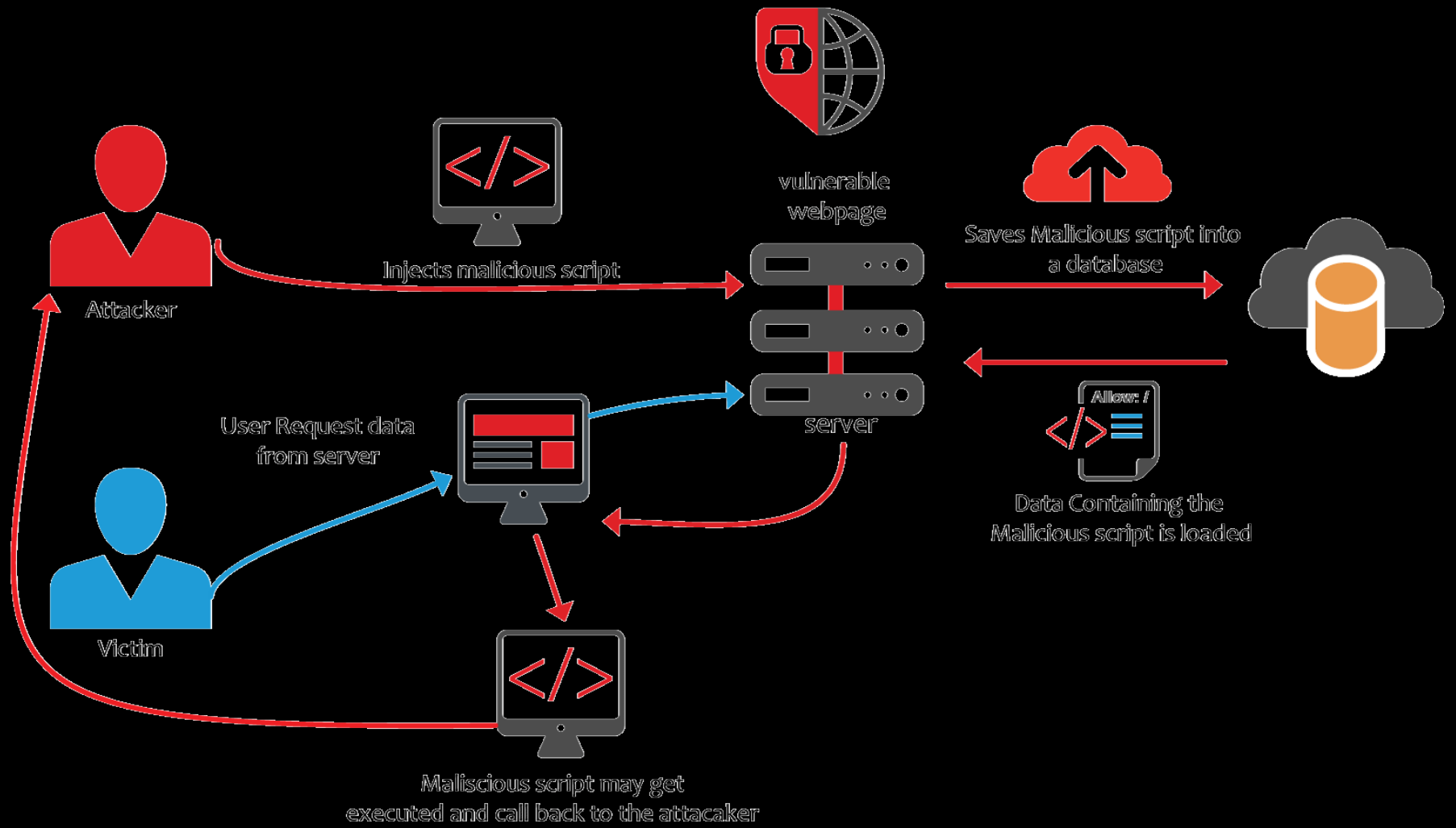
Where untrusted data is used

Data Persistence	XSS	Server	Client
	Stored	Stored Server XSS	Stored Client XSS
	Reflected	Reflected Server XSS	Reflected Client XSS

- DOM Based XSS is a subset of Client XSS (where the data source is from the DOM only)
- Stored vs. Reflected only affects the likelihood of successful attack, not the nature of vulnerability or the most effective defense

Src: https://www.owasp.org/images/c/c6/Server-XSS_vs_Client-XSS_Chart.jpg

How does it work?



How its done?

- Reflected XSS
  - Injected script is directly reflected off the webserver

We will be using DVWA to test the vulnerabilities...

Demo...

How its done? Contd...

- Stored XSS
  - The malicious script is stored in the database permanently on the target servers, like in database

Demo...

Different XSS attacks

- <script>alert(‘XSS’);</script>
- <scRipt>alert(‘XSS’);</script>
- “><script>alert(‘XSS’);</script>
- Vuln Links
- <IMG onmouseover="alert('xxs')“>

- http://ha.ckers.org/xsscalc.html
- %22%3E%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%22%48%69%22%29%3B%3C%2F%73%63%72%69%70%74%3E

Detect if you are vulnerable

- Developer should review detailed manual of the input and output handles in the code, where HTTP request are likely to occur.
- See if form validations are done in the client side or server side
- Use vulnerability scanners like Acunetix, VEGA in Kali Linux, etc.

Prevention

- Never insert untrusted data except in allowed locations
- Never accept JS code from untrusted source
- HTML/CSS/URL escape before inserting untrusted data into HTML elements
- Medium level security measure may include, manually validating the content from the inputs, to remove tags from input
- Output encoding before inserting into HTML elements
  - Htmlspecialchars(), htmlentities()

Src: https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sh
RULE_.230_-_Never_Insert_Untrusted_Data_Except_in_Allowed_Locations