Exploiting web applications

via XSS

●●●

NepSec Sydney

Chapter 0x05 | 6th Meetup

@imhaxormad
@MrMeterpreter

First things first - The Top 10 (as referred by OWASP)

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Today’s focus - XSS (Cross Site Scripting)

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise begin and trusted websites. - OWASP

Cross-site Scripting (XSS) is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. - Acunetix

Cross site scripting (XSS) is a common attack vector that injects malicious code into a vulnerable web application. XSS differs from other web attack vectors (e.g., SQL injections), in that it does not directly target the application itself. Instead, the users of the web application

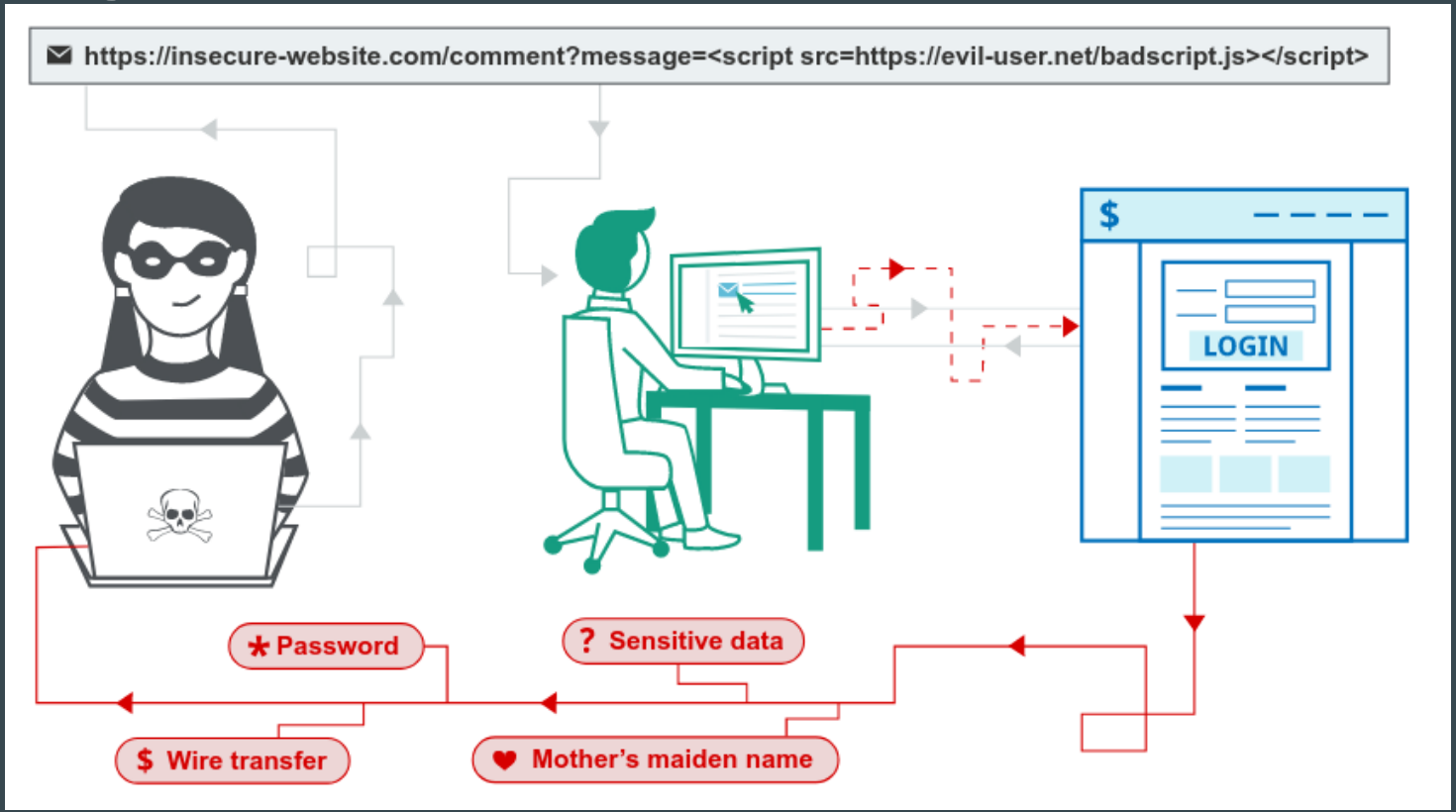
How Cross-site Scripting Works

1. To run malicious JavaScript code in a victim’s browser, an attacker must first find a way to inject malicious code (payload) into a web page that the victim visits.

2. After that, the victim must visit the web page with the malicious code. If the attack is directed at particular victims, the attacker can use social engineering and/or phishing to send a malicious URL to the victim.

As outlined by : https://www.acunetix.com/websitesecurity/cross-site-scripting/

Working of XSS



Source: https://portswigger.net/web-security/cross-site-scripting

Types of XSS

Persistent XSS - where the malicious string originates from the website's database.

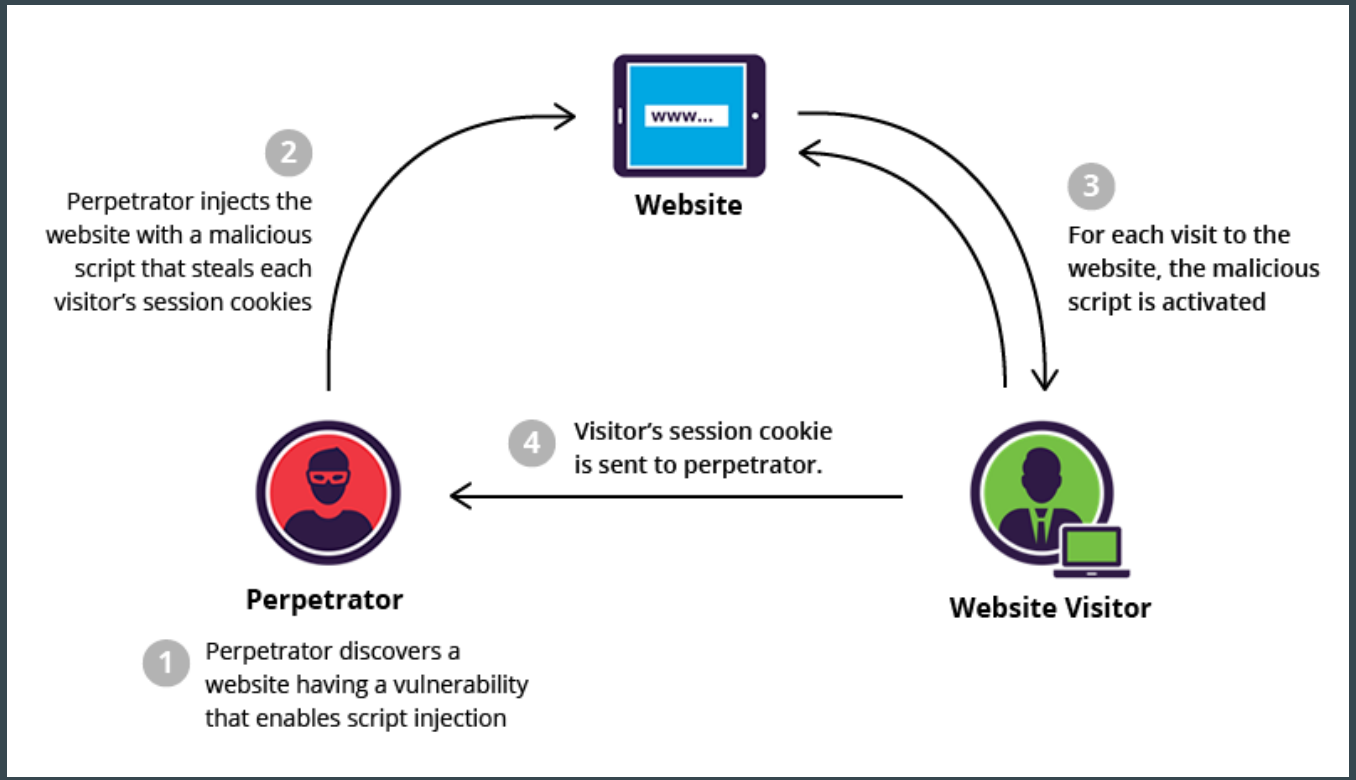
(as the name suggests - it’s persistent - and remains inside the webpage.)

Reflected XSS, where the malicious string originates from the victim's request.

(as the name suggests - it’s reflected there in the webpage.)

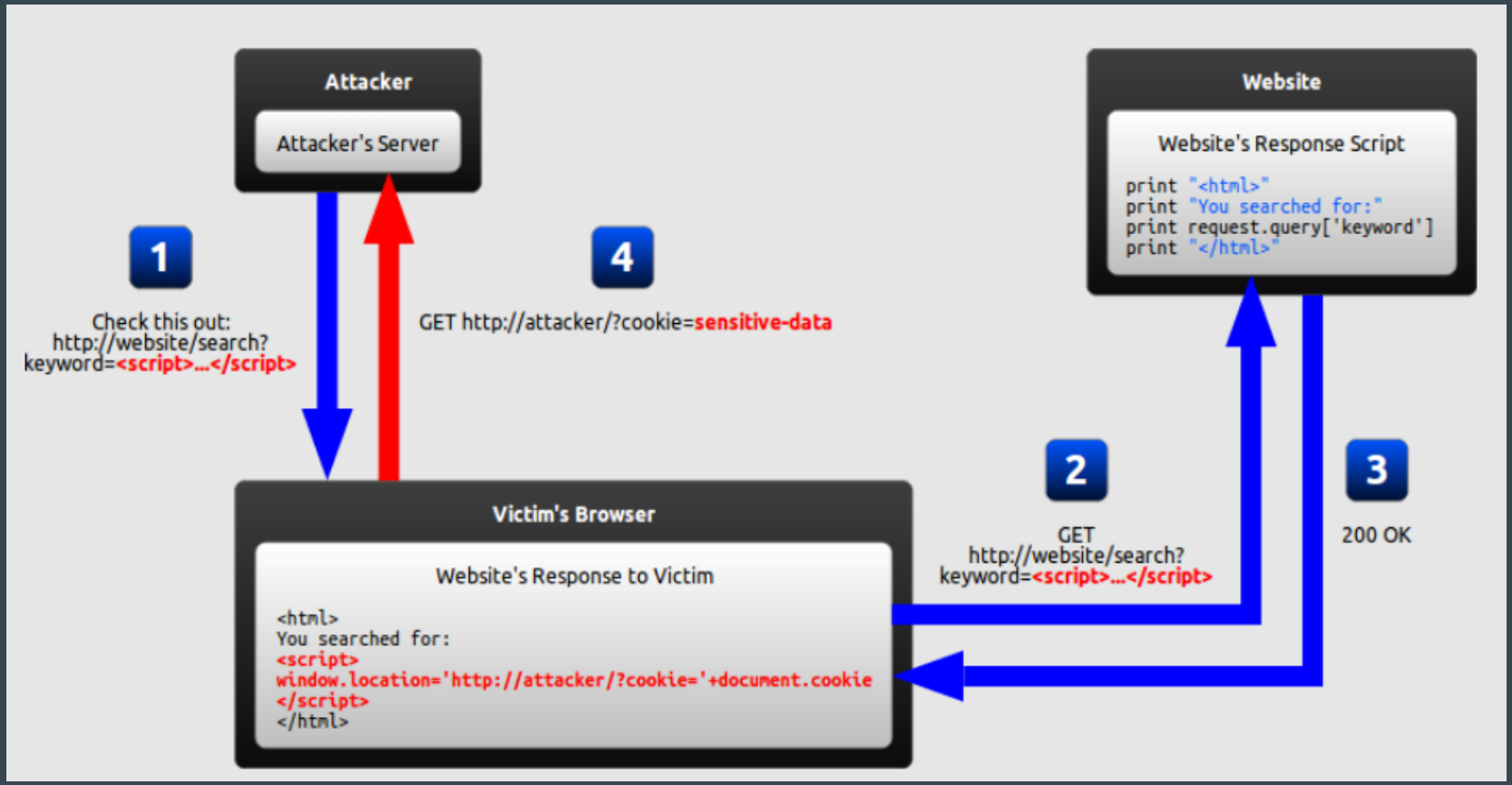
DOM-based XSS, where the vulnerability is in the client-side code rather than the server-side code.

Persistent XSS aka Stored XSS

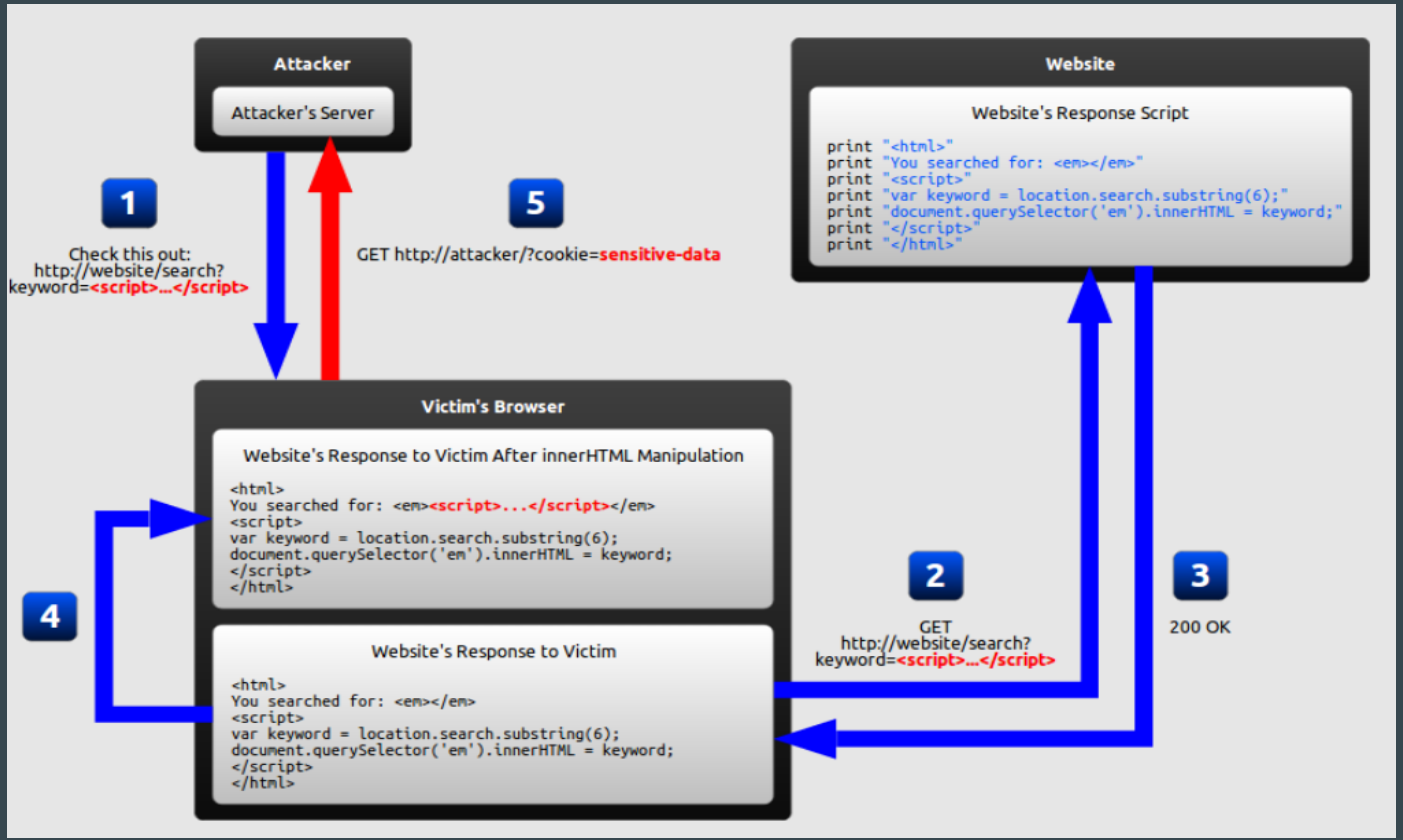


Source: https://www.imperva.com/learn/wp-content/uploads/sites/13/2019/01/sorted-XSS.png

Reflected XSS



DOM based XSS aka Type-0 XSS



So what exactly can XSS be used for?

Impersonate or masquerade as the victim user.

Carry out any action that the user is able to perform.

Read any data that the user is able to access.

Capture the user's login credentials.

Perform virtual defacement of the web site.

Inject trojan functionality into the web site.

Cross-site Scripting Attack Vectors

<script> tag → <script> alert("XSS"); </script>

JavaScript events → <body onload=alert("XSS")>

<body> tag → <body background="javascript:alert("XSS")">

<iframe> tag → <iframe src="http://attacker.com/xss.html">

<input> tag → <input type="image" src="javascript:alert('XSS');">

 tag → <div style="background-image:
url(javascript:alert('XSS'))">

etc etc....

Building/Bypassing PayLoad

- “>Test1234<’; (to check what and how things being filtered and where it is reflecting)
- Double and triple URL Encoding
- Base64 encoding
- Open redirection → redirect_url=//javascript:alert(1); else redirect_url=data:text/html;base64,PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4= (Data URI)
- Polyglot (mixture of 5-8 xss payloads and tags)

Lets get some hands on! + Demo!

Knowledgeable materials

1. OWASP Top 10 (https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf)
2. Web application hackers handbook (
   https://repo.zenk-security.com/Magazine%20E-book/The%20web%20application%20hackers%20handbook%20finding%20and%20exploiting%20security%20flaws%20-ed2%202011.pdf
)
3. The Tangled Web - Guide to securing modern web applications (
   https://repo.zenk-security.com/Techniques%20d.attaques%20%20.%20%20Failles/The%20Tagled%20Web%20A%20Guide%20to%20Securing%20Modern%20Web%20Applications.pdf
)
4. https://excess-xss.com/
5. https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
6. https://portswigger.net/web-security/cross-site-scripting
7. https://pentest-tools.com/blog/xss-attacks-practical-scenarios/

If infosec interests you then give our podcast a listen as well. Cheers!

“Nepal Got Hacked”

Link: https://soundcloud.com/nepal-got-hacked - or just google it, available in other platforms too!

We are also seeking for sponsors, So if you know someone who would lets chat!