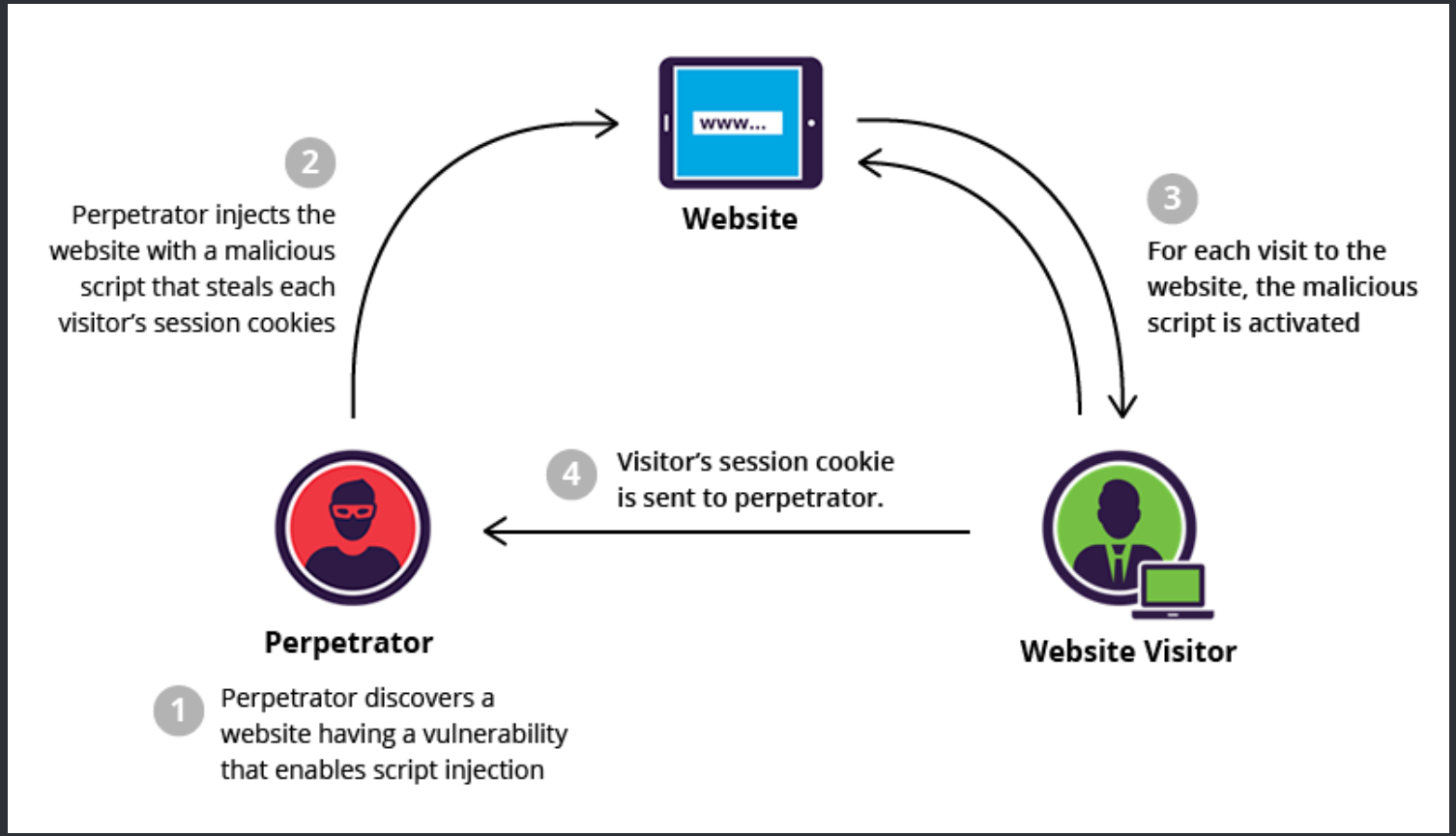
KEYLOGGING

STORED CROSS SITE SCRIPTING (XSS)

HOW STORED CROSS SITE SCRIPTING WORKS

- Basics of Stored XSS
  - Hacker identifies vulnerable website that asks for data and stores it in a database
    - Malicious code is injected into webpage
  - User visits the website containing the Stored XSS
    - Code is executed by the user's web browser
    - (Specifically, running a JavaScript browser based keylogger in this example)


www...
Website
2
Perpetrator injects the website with a malicious script that steals each visitor's session cookies
3
For each visit to the website, the malicious script is activated
4
Visitor's session cookie is sent to perpetrator.
Perpetrator
Website Visitor
1
Perpetrator discovers a website having a vulnerability that enables script injection

XSS KEYLOGGER ATTACKS

HOW?

This type of keylogger attack is launched using XSS; using a JavaScript file on a PHP server to create a log of keystrokes.

WHERE?

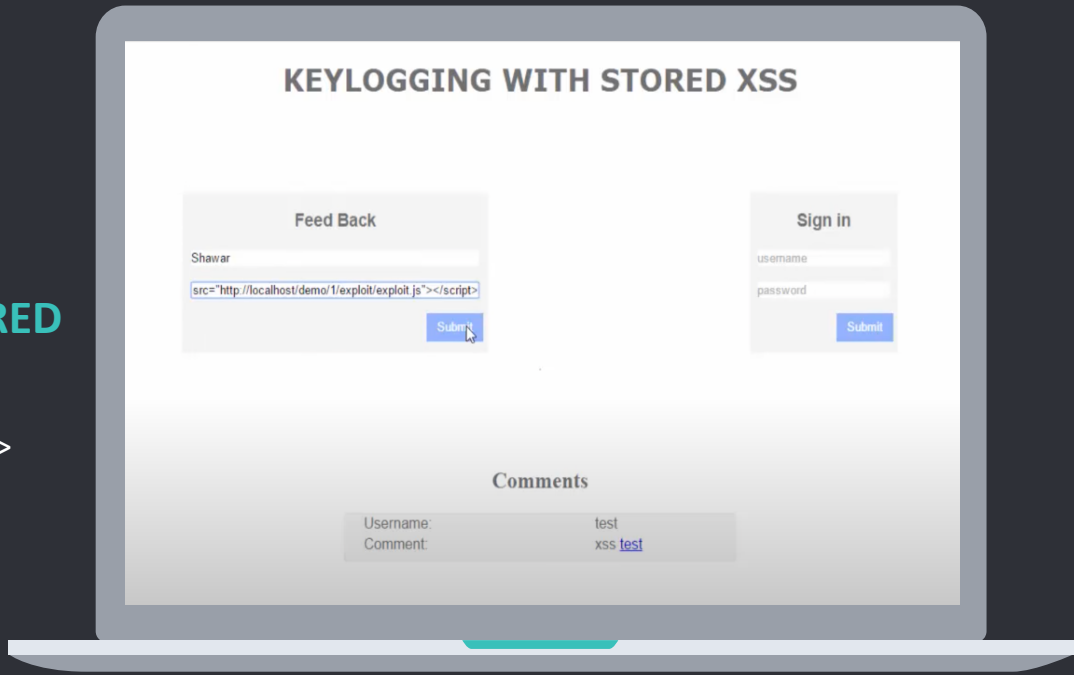
This type of keylogger attack is executed when the victim enters an infected website.

WHAT FAILED?

Keylogger attacks can be hardware or software-based. Hardware-based keyloggers are tied to a specific computer while software-based ones are not.

HOW THE PAYLOAD IS DELIVERED

- “><img src=x onerror=prompt(1)>
- <script src=”location of .js”></script>



HOW TO PREVENT THESE ATTACKS

Validate and Sanitize Input

This XSS attack is injected because the data entry field allows for input containing HTML tags. First, the input can be validated to contain only certain characters such as letters, numbers, dashes, etc. Second, the PHP FILTER_SANITIZE_STRING filter can be used to remove all HTML tags from a string.

Escape Output

Use the PHP FILTER_SANITIZE_SPECIAL_CHARS filter to escape "<>& as well as characters with ASCII values under 32 if they are present. This ensure that any HTML tags are escaped and converted to their “entity equivalents” which will prevent the malicious HTML from rendering when it is outputted to the webpage.

FILTER_SANITIZE_STRING

FILTER_SANITIZE_FULL_SPECIAL_CHARS

HOW TO PREVENT THESE ATTACKS

HttpOnly Flag

This is an additional flag in the Set-Cookie HTTP response header, if the browser supports it, that makes it so that the user's cookies cannot be accessed through client-side script. Even if a XSS breach exists, the user's cookie is not revealed to a third-party, causing the attack to fail.

Make Raw Data Unreadable

Ensure the only authorized users have read access to raw data and do not store data in a web accessible folder.

WORKS CITED

"How to Prevent Cross Site Scripting Attacks." *Wordfence*, 4 Jan. 2017, www.wordfence.com/learn/how-to-prevent-cross-site-scripting-attacks/.

"HttpOnly." *OWASP*, owasp.org/www-community/HttpOnly.

"Keylogging with Cross-Site Scripting Vulnerability." *Web Security 24x7*, websecurity247.blogspot.com/2016/07/keylogging-with-cross-site-scripting.html.

"What Is Stored Cross Site Scripting or Stored XSS?" *Learn Ethical Hacking and Penetration Testing Online*, www.hackingloops.com/what-is-stored-cross-site-scripting-or-stored-xss/.

"What Is XSS: Stored Cross Site Scripting." *Imperva*, 29 Dec. 2019, www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/#:~:text=To%20successfully%20execute%20a%20stored,%2C%20via%20a%20comment%20field).&text=Every%20time%20the%20infected%20page,transmitted%20to%20the%20victim's%20browser.