

Intro to XSS



HyperionDev



Welcome



Lecturer: Liano Naidoo

Moderator: Zahir Junjeo

Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment - please engage accordingly.
- ❑ No question is daft or silly - **ask them!**
- ❑ There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- ❑ You can also submit questions here:
hyperiondev.com/sbc4-cs-questions
- ❑ For all non-academic questions, please submit a query:
www.hyperiondev.com/support
- ❑ Report a safeguarding incident:
hyperiondev.com/safeguardreporting
- ❑ We would love your feedback on lectures:
<https://hyperiondev.wufoo.com/forms/zsgv4m40ui4i0g/>

Previously:

- Learned about Ciphers
- Explored encryption
- Math involved with ciphers

Objectives

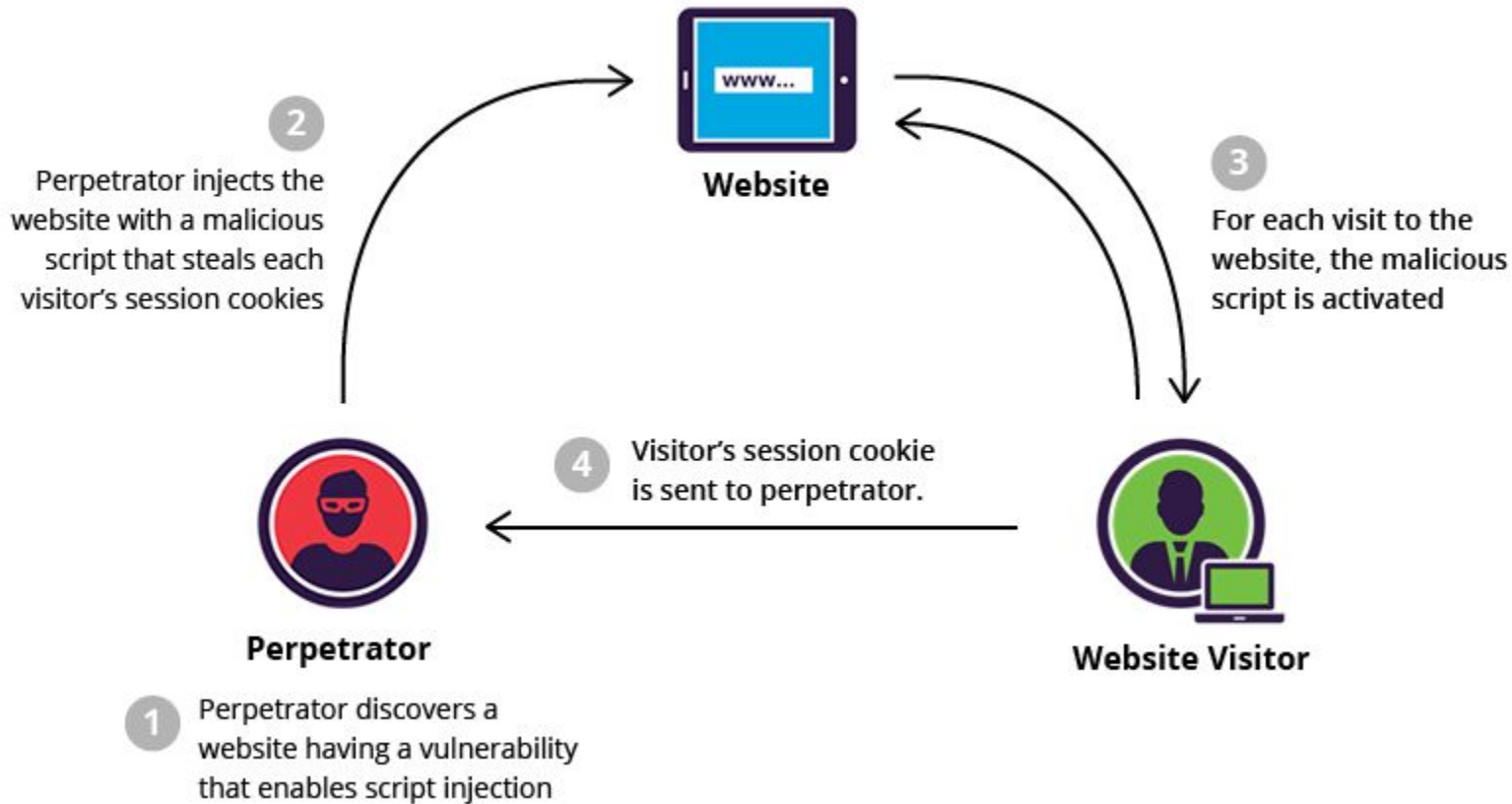
- What XSS is
- How it works
- Types of XSS attacks
- Real world Examples
- XSS Prevention

What is XSS?

- A type of web security flaw called cross-site scripting (XSS) enables an attacker to insert malicious code into a web page that is being viewed by other users. By doing this, the attacker may be able to obtain sensitive data from unwitting victims, such as login credentials, session cookies, or personal information. Hackers frequently use this method to compromise websites and web applications.

How does this work?

- XSS takes advantage of the trust that exists between a web application's users. By taking advantage of flaws in the input validation or output encoding of the web application, an attacker can insert malicious code into a web page. The browser of the victim then runs the malicious code, giving the attacker the opportunity to steal confidential data or carry out other malicious deeds



Types of XSS attacks

- There are three main types of XSS attacks: reflected, stored, and DOM-based.
- Reflected XSS attacks involve injecting malicious code into a web page that is immediately returned to the victim's browser.
- Stored XSS attacks involve injecting malicious code that is stored on the server and executed whenever the victim views the affected page.
- DOM-based XSS attacks involve injecting malicious code that is executed by the victim's browser as part of the Document Object Model (DOM) tree.

Real World Attacks

Over the years, there have been many prominent XSS attacks, including the infamous **MySpace worm**, which in 2005 affected over one million users. The **Samy worm** on MySpace in 2007, the **TweetDeck XSS attack** in 2014, and the **Equifax breach in 2017**—all of which were brought on by unpatched vulnerabilities in **Apache Struts** that permitted remote code execution via XSS—are additional noteworthy attacks.

XSS Prevention

- Avoiding HTML in inputs if at all possible Preventing users from entering HTML into form inputs is a very effective way to avoid persistent cross-site scripting attacks. Other options, like markdown and WYSIWYG editors, enable users to create rich content without using HTML.
- Validating inputs - Validation is the process of putting in place rules that stop users from entering data into forms that don't meet specific requirements. For instance, the validation rules for an input that requests the user's "Last Name" should restrict user input to alphanumeric characters only. The "script>" tags and other symbols frequently used in cross-site scripting can be rejected by validation rules.
- Data sanitization - Data sanitization is similar to data validation in that it takes place after the data has been posted to the web server but before it is shown to another user. A variety of online tools are available to clean HTML and remove any malicious code injections.

XSS Prevention

- Using special rules for their cookie handling, Web applications can reduce cookie theft caused by cross-site scripting attacks. This is another method for implementing cookie security measures. Cross-site scripting attackers cannot access cookies if they are tied to specific IP addresses. Additionally, rules can be made to completely prevent JavaScript from accessing cookies.
- Setting WAF rules - A WAF can also be configured to enforce rules which will prevent reflected cross-site scripting. These WAF rules employ strategies that will block strange requests to the server, including cross-site scripting attacks. The Cloudflare WAF offers turnkey installation and protects web applications from cross-site scripting, DDoS attacks, SQL injection, and other common threats.

Case 1: British Airways

In 2018, Magecart, a well-known hacker group well-known for credit card skimming attacks, attacked British Airways. The group took advantage of an XSS flaw in the **Feedify JavaScript library**, which was used on the British Airways website.

Attackers altered the script so that it would send user information to a malicious server that had a domain name that sounded similar to British Airways. Users thought they were making purchases from a secure server because the fake server had an SSL certificate. Before the breach was discovered, they were able to successfully skim 380,000 booking transactions for credit card information.

Case 2: Fortnite

Over 200 million players of the well-known multiplayer game were affected by an XSS vulnerability in 2019. The Fortnite team missed a retired, unprotected page. An XSS vulnerability on the page allowed attackers to access all Fornite users' data without authorization.

Users could have been redirected to a fake login page by attackers using XSS in conjunction with an unsafe single sign-on (SSO) vulnerability. This would give them the chance to intercept player conversations to gather intelligence for upcoming assaults and steal in-game virtual currency. It is unknown if the vulnerability was used by attackers in the interim after Checkpoint discovered the attack and alerted Fortnite.

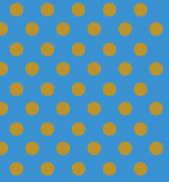
Case 3: eBay

Late in 2015 and early in 2016, eBay had a serious XSS flaw. The website employed a "url" parameter that sent visitors to various pages on the platform, but the parameter's value was not verified. This made it possible for attackers to insert harmful code into a page.


Due to a vulnerability, attackers were able to take full control of eBay seller accounts, offer products at a lower price, and steal payment information. Attackers actively used it to manipulate eBay listings for expensive goods like vehicles. Although the vulnerability was eventually fixed by eBay, recurrent attacks persisted until 2017.



Question:



What preventative measure/best practice could eBay have taken to avoid the vulnerability?



Our Responsibility

It is our responsibility as ethical practitioners of cyber security to defend web applications against such assaults. We can create a safer online environment for everyone if we have a better understanding of the different XSS attack types, their effects, and how to prevent them.

In terms of pertinent British values, defending web applications from XSS attacks is an important part of national security and adds to the UK's overall security. Additionally, we demonstrate the virtues of responsibility and respect for the law, as outlined in the UK's legal and ethical frameworks, by educating ourselves and others on cyber security.

Wrapping up

In conclusion, XSS is a serious flaw in web security that could have disastrous effects on both individuals and businesses.

Web developers and security experts should be aware of the dangers posed by XSS attacks and take proactive measures to stop and mitigate them.

We can contribute to building a safer and more secure web for everyone by adhering to best practices and staying up to date with the most recent security tools and techniques.

Realize the importance of keeping up with the most recent advancements in the field, to learn from their mistakes, and to continually advance their knowledge and skills.

Next up

Recap on Web Lecture



Hyperiondev

Thank you for joining us

1. Take regular breaks
2. Stay hydrated
3. Avoid prolonged screen time
4. Practice good posture
5. Get regular exercise



Questions and Answers

