



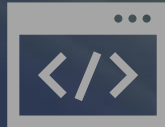
Tahir Ali

# Cross-Site Scripting (XSS)

# What is Cross-Site (XSS) Scripting?



Being able to run your own JavaScript on a web application



Cross-Site Scripting is only a vulnerability when you visit a website and you see an area in which the web page is asking you as the user to input something, such as your username or password.



These types of user inputs that allow the user to insert characters within them may be vulnerable to XSS which is why it is essential for developers to ensure that only certain characters are allowed.



Logical thinking is a must to developers, for example why would a user need to insert a special character such as an exclamation mark or a semi colon in the username section of a website?

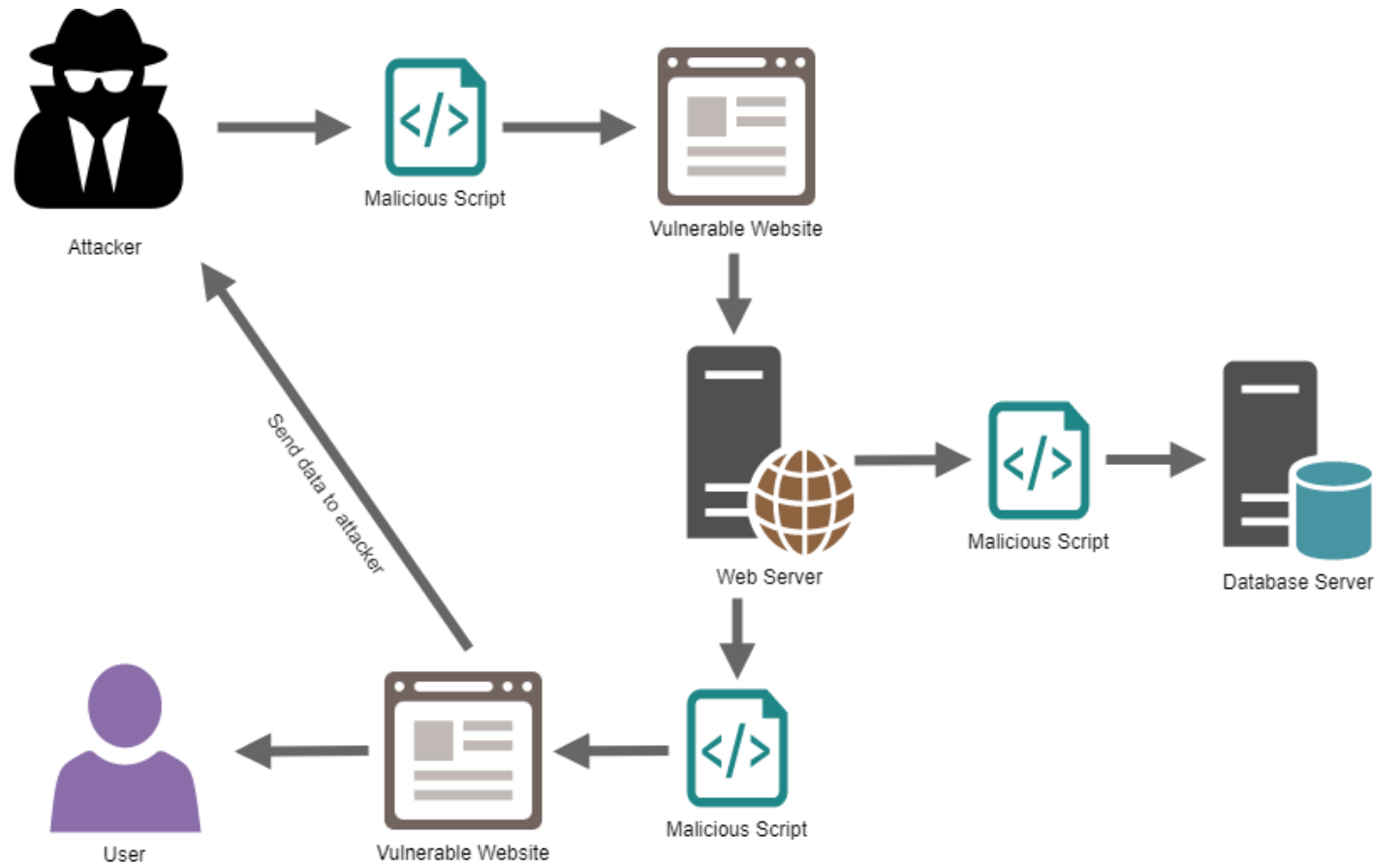


# What are session cookies?

- Our personal online ID's
- Users can remove their session cookie simply by deleting their browser history, as this is where the session cookie is stored
- Session cookies improve user experience

[2]

# Types of XSS



# DOM XSS

- Stands for Document Object Model-based Cross-Site Scripting
- A DOM-based XSS attack is only possible when the web application writes data onto the Document Object Model (DOM) without appropriate sanitisation.
- Attackers can manipulate what is written into the DOM to include malicious JavaScript code.

[4]

Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

[Home](#)

WE LIKE TO  
**BLOG** 

Search



Figure 1

...0399af5283b1199e00de00a2.web-security-academy.net says

1

OK

[Home](#)

0 search results for "'><svg onload=alert(1)>'

Search the blog...

Search

[< Back to Blog](#)


Figure 2

# Stored XSS


- This type of XSS can impact any user browsing a website.
- Stored XSS is different from Reflected XSS and DOM XSS because those require the user to be convinced to click on a URL or input a malicious payload into a textbox.
- Harmful scripts can be placed into a normal form field, like a textbox or in my example a comment section of a blog page.

[5]

know.

 Andy Tool | 03 May 2023

I want to pass this off as my own but friends pointed out I can't spell?

 B.B Gun | 20 May 2023

I've never read anything twice' and today will be no exception.

 Tahir | 24 May 2023

## Leave a comment

Comment:

`<script>alert(1)</script>`

Name:

Tahir

Email:

Tahirali@gmail.com

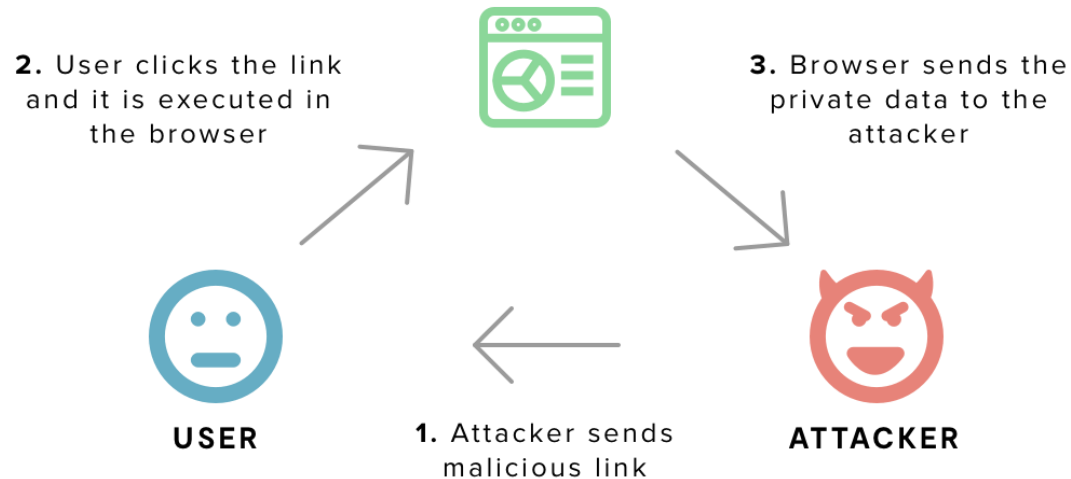
Website:

Post Comment

[< Back to Blog](#)

Figure 3

# Reflected XSS



- Reflected XSS refers to a vulnerability where the XSS payload is in the URL
- The impact of reflected XSS is usually less detrimental than stored XSS, where an attack can be delivered within the vulnerable application itself
- Reflected XSS has different types. The type depends on the location of the reflected data within the web apps response
- If the application executes some type of validation or other processing on the data before it is reflected, then the type of XSS payload needed will usually be affected

[6]



# Mitigating XSS Risks

There is no simple fix to prevent XSS attacks, the protection strategies vary and differ depending on the web application. Nonetheless, it is extremely important to consider the following strategies when thinking about ways to mitigate XSS attacks:

- **Prohibit HTML/JavaScript code in inputs**
- **Validate inputs**
- **Secure your cookies**
- **Use a web application firewall (WAF)**

A vulnerability assessment and/or a penetration test can be incredibly helpful to identify not only XSS but also any other vulnerabilities within your network.

**[7]**

# Real-Life Examples of Cross-Site Scripting Attacks



# British Airways

- British Airways was attacked in 2018 by Magecart, which is a high-profile hacker group famous for credit card skimming attacks. They exploited an XSS vulnerability in a JavaScript library called Feedify, which was used on the British Airways website.
- The Magecart hackers altered the script to send customer data to a malicious server, which used a domain name alike the British Airways one.
- Victims believed they were purchasing tickets from British Airways, and Magecart managed to perform credit card skimming on 380,000 booking transactions before the breach was discovered.



# Fortnite

- Fortnite, which is an extremely popular multiplayer game experienced an XSS vulnerability in 2019. An unsecure page went unnoticed by Fortnite developers which was a grave mistake as the page had an XSS vulnerability that allowed attackers to gain unauthorised access to the data of all Fortnite users.
- Attackers could have used XSS, in combination with an insecure single sign on (SSO) vulnerability which would have redirected users to a fake login page allowing them to steal virtual currency within the game, and record player conversations.
- The vulnerability was discovered and Fortnite was notified, however it is unknown if the vulnerability was exploited by attackers.

# eBay

- eBay had a severe XSS vulnerability in 2016. The website used a “url” parameter that redirected users to different pages on eBay. This is normally fine, but in this case the value of the parameter was not validated which enabled attackers to inject malicious code into a page.
- This was an extremely serious case, as the vulnerability allowed the attackers to gain full access to eBay seller accounts, sell products at a discount, and steal payment details. It was actively used by attackers
- eBay eventually remediated the vulnerability, but attacks continued until 2017.







The End

---

# Bibliography

[1] PortSwigger (n.d.). *What is cross-site scripting (XSS) and how to prevent it?* [online] Portswigger.net. Available at: <https://portswigger.net/web-security/cross-site-scripting>. [Accessed 25 May 2023].

[2] Secureprivacy.ai. (2023). Available at: <https://secureprivacy.ai/blog/session-cookies-vs-persistent-cookies> [Accessed 25 May 2023].

[3] www.javatpoint.com. (n.d.). *Session vs Cookies: What's the Difference? - javatpoint*. [online] Available at: <https://www.javatpoint.com/session-vs-cookies#:~:text=Why%20Use%20Session%3F%201%20Sessions%20are%20used%20to> [Accessed 25 May 2023].

[4] Nidecki, T.A. (2019). DOM XSS: An Explanation of DOM-based Cross-site Scripting. [online] Acunetix. Available at: <https://www.acunetix.com/blog/articles/dom-xss-explained/#:~:text=DOM%20XSS%20stands%20for%20Document%20Object%20Model-based%20Cross-site> [Accessed 25 May 2023].

[5] portswigger.net. (n.d.). *What is stored XSS (cross-site scripting)? Tutorial & Examples | Web Security Academy*. [online] Available at: <https://portswigger.net/web-security/cross-site-scripting/stored> [Accessed 25 May 2023].

[6] PortSwigger (2019). *What is reflected XSS (cross-site scripting)? Tutorial & Examples*. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/cross-site-scripting/reflected> [Accessed 25 May 2023].

[7] Stone, M. (n.d.). *What Is & How to Mitigate Cross-Site Scripting (XSS) Attacks*. [online] Verizon Enterprise. Available at: <https://www.verizon.com/business/resources/articles/s/how-to-mitigate-cross-site-scripting/>. [Accessed 25 May 2023].

[8] Dizdar, A. (2022). *XSS Attack: 3 Real Life Attacks and Code Examples*. [online] Bright Security. Available at: <https://brightsec.com/blog/xss-attack/#:~:text=Here%20are%20some%20ramifications%20of%20an%20XSS%20attack%3A> [Accessed 25 May 2023].