

# Cross-Site Scripting (XSS)

CWE-79  
OWASP Top Ten A03:2021 - Injection

# Cross-Site Scripting



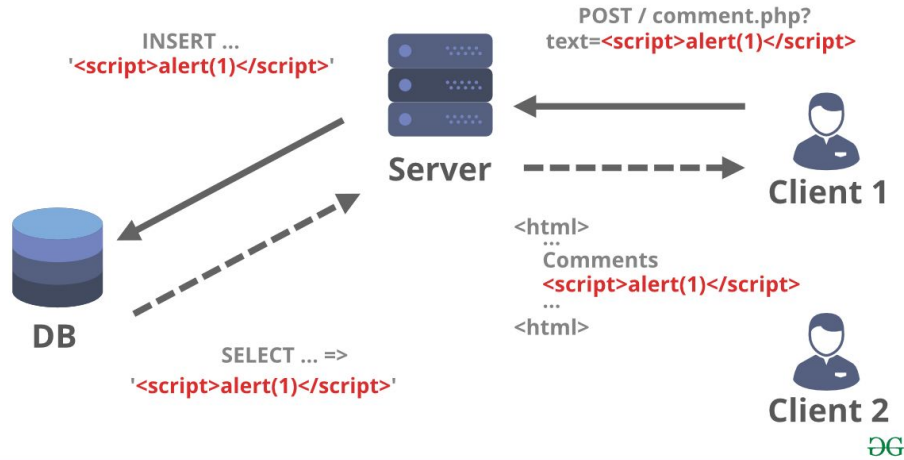
Źródło: <https://www.wpxplorer.com/wp-content/uploads/wordpress-cross-site-scripting-guide-prevention.png>

- Injection na 3 miejscu listy w 2021 roku
- Od lat jedna z najczęściej występujących luk aplikacji internetowych
- Pozwala wykonać niepożądany kod na przeglądarce użytkownika i przejąć interakcje między nim a stroną docelową
- Jeśli użytkownik jest uprzywilejowany można tego nadużyć

# Na czym polega XSS?

- Atakujący przygotowuje kod w języku skryptowym (np. JavaScript) i umieszcza go na stronie
- Użytkownik wchodzi w interakcję z elementem zawierającym wstrzyknięty kod przez np. kliknięcie w link
- Przeglądarka nie odróżnia niepożądanego kodu od kodu źródłowego zaufanej strony
- Kod jest wykonany w przeglądarce użytkownika

## Cross Site Scripting(XSS)



Źródło:  
<https://media.geeksforgeeks.org/wp-content/uploads/20190516152959/Cross-Site-ScriptingXSS.png>

## TweetDeck XSS - przykład ataku z 2014 r.

- Luka spowodowana niepoprawną obsługą zamiany niektórych znaków Unicode na obrazki
- Przygotowany kod w JavaScript włącznie ze znakiem serca w Unicode (❤️) jest wysyłany jako tweet
- Podczas przeglądania TweetDeck samo wyświetlenie tweeta powoduje wykonanie kodu w przeglądarce
- <- Kod po lewej stronie powoduje automatyczny retweet oraz wyświetlenie alertu w przeglądarce z komunikatem "XSS in Tweetdeck"

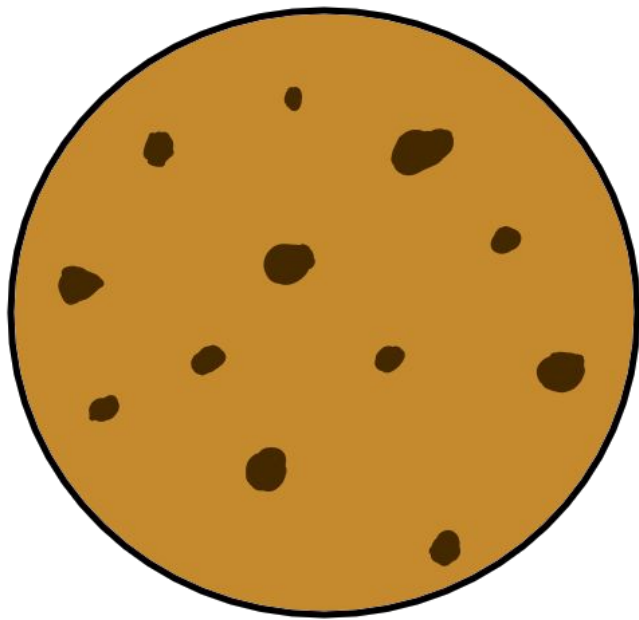


Źródło: @derGuruhn na Twitter.com

# Cross Site Request Forgery (CSRF)

CWE-352  
OWASP Top Ten A01:2021 - Broken Access Control

# Jak działa atak CSRF?



- Wykorzystuje istniejącą sesję użytkownika - podszywa się pod niego
- Atak csrf wykorzystuje istniejące funkcjonalności aplikacji
- Identyfikator sesji znajduje się w ciasteczku zapisanym w przeglądarce
- Ofiara otwiera stronę, która zawiera ukryte złośliwe zapytanie do serwera z istniejącą sesją:
  - GET - zawsze
  - POST - formularze
  - XHR(dowolna metoda) - CORS all origins
- Cookie SameSite=None
- Przeglądarka automatycznie wysyła ciasteczka do powiązanej domeny
- Zapytanie jest wykonywane na serwerze, bez wiedzy użytkownika, ponieważ ciasteczko uwierzytelniające zostało dostarczone



# Przykładowy atak

- Użytkownik wchodzi na stronę internetową
- Przeglądarka automatycznie ładuje zdjęcia
- Jedno ze zdjęć w atrybucie src ma link ze spreparowanym zapytaniem:
- `img src = goodomain/burnMoney/?amount=all`
- przeglądarka wysyła zapytanie które zostanie zrealizowane
- użytkownik staje się ofiarą złośliwego zapytania

uTorrent miał podatność CSRF w webowej konsoli aplikacji zainstalowanej u użytkownika pozwalającą na pobieranie dowolnych plików .torrent

# **Dziękujemy za uwagę!**

Autorzy:

Tymoteusz Urbanowicz - 20149

Jan Wieczorek - 21024



# Źródła

- [https://www.theregister.com/2014/06/12/tweetdeck\\_xss\\_vuln\\_uncovered\\_by\\_heart\\_hunting\\_teenager](https://www.theregister.com/2014/06/12/tweetdeck_xss_vuln_uncovered_by_heart_hunting_teenager)
- <https://portswigger.net/web-security/cross-site-scripting>
- [https://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery)
- <https://cwe.mitre.org/data/definitions/352.html>