Testowanie podatności wybranych aplikacji webowych na ataki typu Cross-site scripting (XSS)

Przygotowali Paweł Gasz i Krzysztof Duda

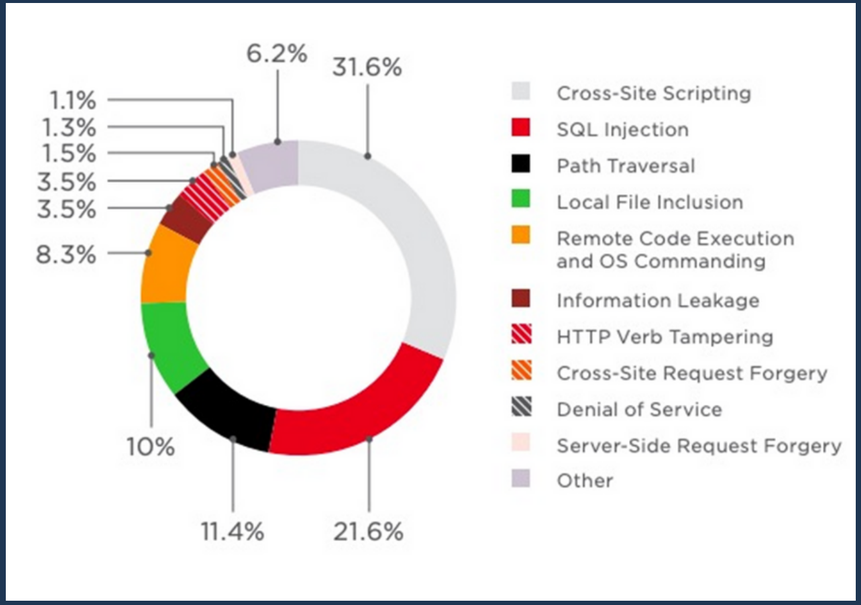
CZYM JEST XSS

Cross-Site Scripting (XSS) - to jedna z najczęściej występujących podatności aplikacji webowych.
Jest to atak na serwis WWW, który polega na możliwości osadzenia w treści strony własnego kodu JavaScript, co w konsekwencji może doprowadzić do wykonania niepożądanych akcji przez użytkowników odwiedzających tę stronę.

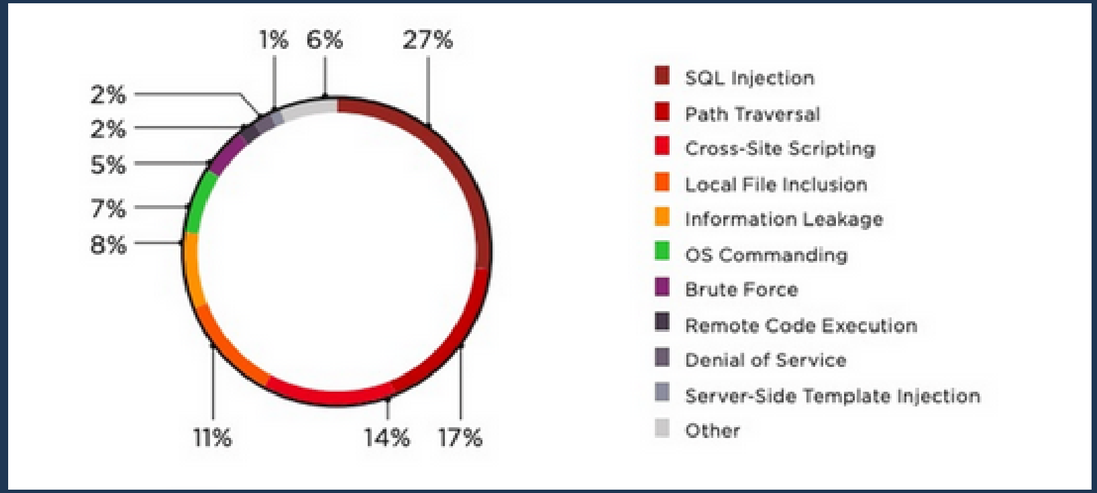
```
<script>alert('Cross-Site Scripting!');</script>
```

Popularność XSS

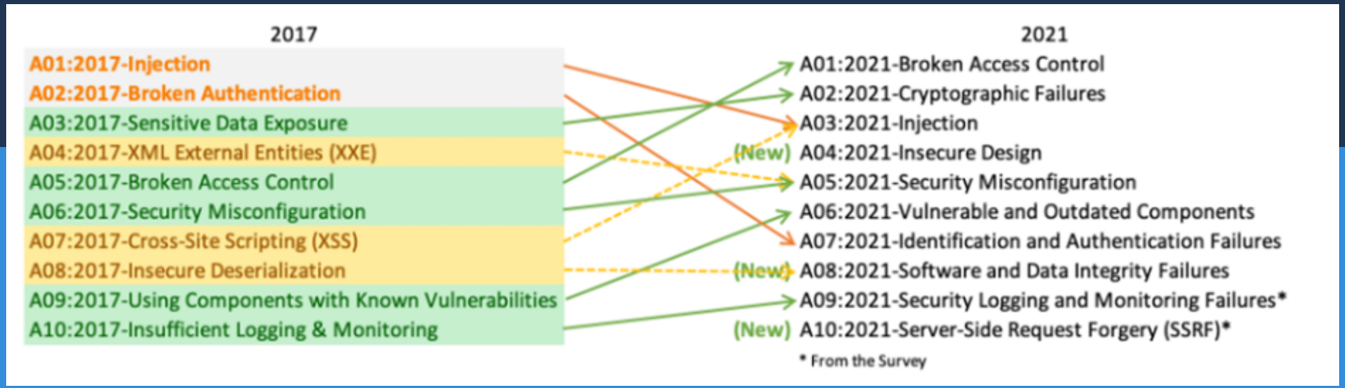
2017



2020



2021



Na czym polega XSS?

Wyszukiwarka książek

Podaj tytuł

Szukaj

Wyszukiwarka książek

Podaj tytuł

Szukaj

Szukana pozycja to:Tytuł

Wyszukiwarka książek

Podaj tytuł

Szukaj

Szukana pozycja to:**Tytuł**

<u>Tytuł

Skąd wiemy, że to XSS ?

<script>alert(1)</script>

Przy testowaniu XSS najczęściej korzystamy z funkcji alert(). Wynika to z tego, że :

- **jest łatwa w użyciu,**
- **wspierana przez wszystkie przeglądarki,**
- **jest funkcją synchroniczną,**
- **jest funkcją wywołującą widoczne efekty,**

DLACZEGO WSZĘDZIE TEN ARERT() ???

Dzięki swoim cechom umożliwia nam określenie, czasu oraz miejsca w kodzie, gdzie udało nam się wstrzyknąć nasz infekujący fragment, przy czym dodatkowo pomaga nam to, że alert wstrzymuje wykonywanie dalszego kodu JS, aż to potwierdzenia.

<script>alert(' alert(1) ;-) ');</script>

Rodzaje XSS

- Reflected https://example.com/search?name=<script>alert(1)</script>
- Stored
- DOM Based

DOM-based XSS

```
Select your language:

<select><script>

document.write("<OPTION value=1>"+document.location.href.substring(document.location.href.indexOf("default=")+8)+"</OPTION>");

document.write("<OPTION value=2>English</OPTION>");

</script></select>
```

http://www.some.site/page.html?default=Polish

http://www.some.site/page.html?default=<script>alert(document.cookie)</script>

Skutki XSS

-wykradnięcie danych
-wykonanie dowolnej akcji
-przejecie sesji
-wykonanie innych ataków :

- skanowanie portów
- keyloggery
- ataki phishingowe

Konteksty XSS

xyz onerror=alert(1)//

<img src=/images/[id].png>

<img src=/images/xyz onerror=alert(1)//.png>

<img src onerror=alert(1)>

Fragment kodu

[XSS]

Metoda ataku

<img src onerror=alert(1)>

Obrona

Zamiana znaków specjalnych na encje HTML.

Encje...

-znak “ zamieniamy na "
-znak ‘ zamieniamy na '
-znak <> zamieniamy na < i >
-znak & zamieniamy na &

Przykład zastosowania:

<u>Tytuł

<u>Tytuł

Fragment kodu

```
<div class="[XSS]"></div>
```

Metoda ataku

```
<div class="" onmouseover=alert(1) "></div>
<div class=""><script>alert(1)</script> "></div>
```

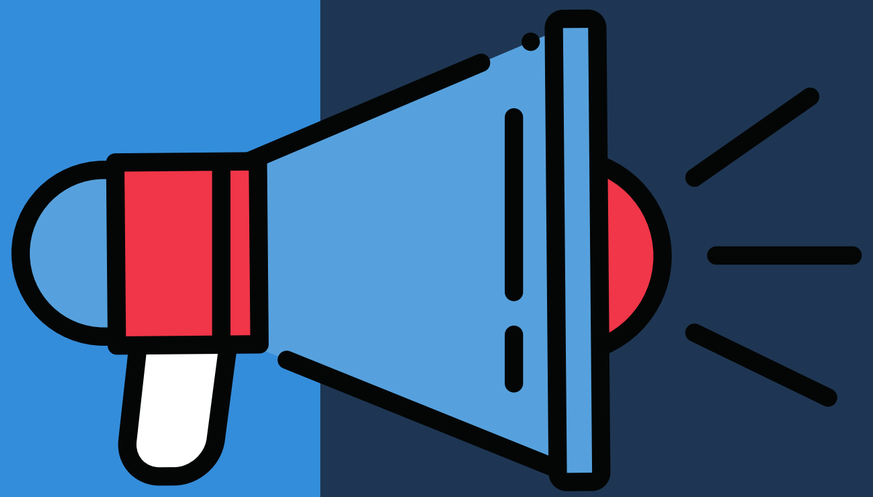
Obrona

Zamiana znaków specjalnych na encje HTML.

Fragment kodu

Metoda ataku

Obrona Umieszczenie nazw atrybutów w cudzysłowach



Fragment kodu

Metoda ataku

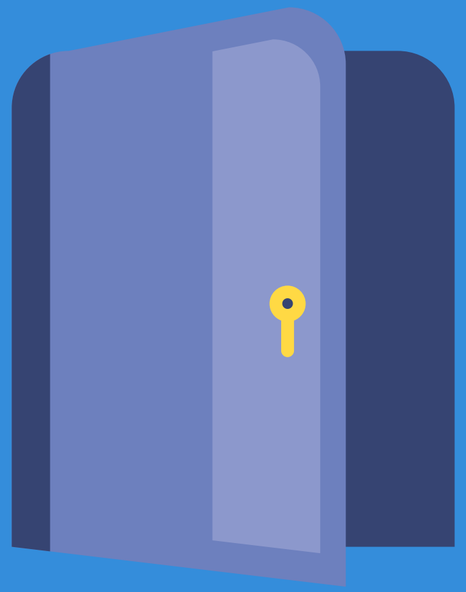
Obrona Odrzucanie innych protokołwó niz HTTP/HTTPS

Konteksty DOM XSS

-funkcje typu eval

-funkcje przyjmujące kod HTML

-funkcje przyjmujące adres URL



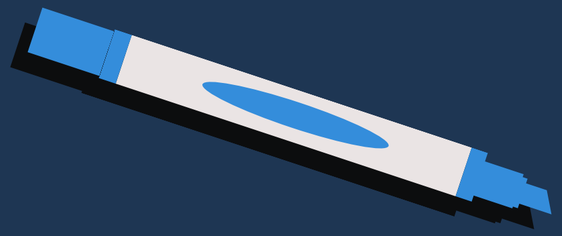
FUNKCJE TYPU EVAL

eval() - funkcja, która interpretuje string tak, jakby był to wyrażenie w języku, i zwraca wynik; natomiast w niektórych językach wykonuje wiele wierszy kodu tak, jakby zostały dołączone zamiast wiersza zawierającego eval

w JS eval() jest to funkcja, która wykonuje kod zapisany jako string

```
alert("zwykły alert", eval(console.log(eval('2 + 2'))));
```

ten kod wyświetla alert z napisem "zwykły alert", a w konsoli wypisywana jest wartość 4



Przykład

Fragment kodu

eval("console.log('Hello " + user + " !')"); , gdzie
np. zmienna user jest podana przez urzytkownika

Metoda ataku

zmieniamy wartość user na wartość zawierającą nasz kod
user = " '); alert(1)// ";
w konsekwencji wykona się
console.log('Hello '); alert(1)// !

Obrona

Zaleca się aby nie używać funkcji tego typu

FUNKCJE PRZYJMUJĄCE KOD HTML

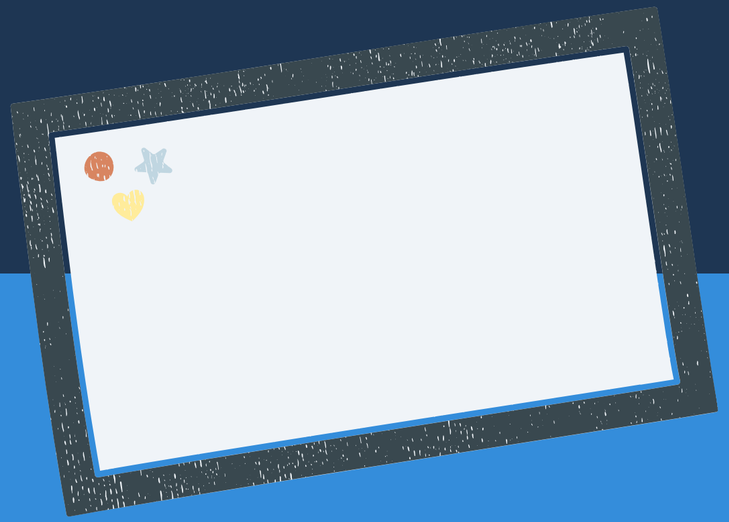
funkcje przyjmujące kod HTML - do takiej funkcji można przesłać kod HTML jako parametr

innerHTML, outerHTML, insertAdjacentHTML, document.write

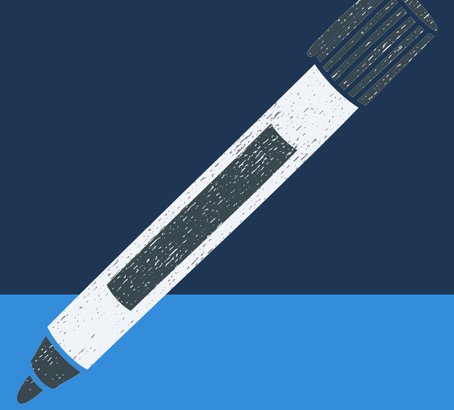
Używanie z danymi od użytkownika przy braku lub szczątkowej walidacji, może powodować, że dowolny kod HTML może zostać umieszczony na stronie

atakujący może w ten sposób zamieścić na stronie tag <script>

Jest to na tyle niebezpieczne, że w taki sposób można zamieścić kod w źródle strony, który może byś o wiele bardziej szkodliwy niż alert(1). Za pomocą atrybutu src można załączać pliki JS.



Przykład



Fragment kodu

```
window.addEventListener('hashchange', (e) => {
    let id = unescape(location.hash.slice(1));
    document.getElementByld('imageContainer').innerHTML =
    '<img src="//example.com/image'+ id + '.png">';
});
```

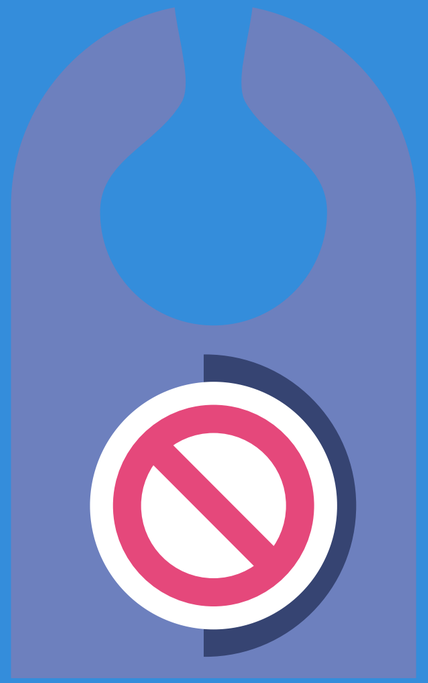
Metoda ataku

przekazanie do adresu URL odpowiedniej wartości

http://naszastrona.pl#"/onerror=alert(1)//

Obrona

Walidacja danych od użytkownika

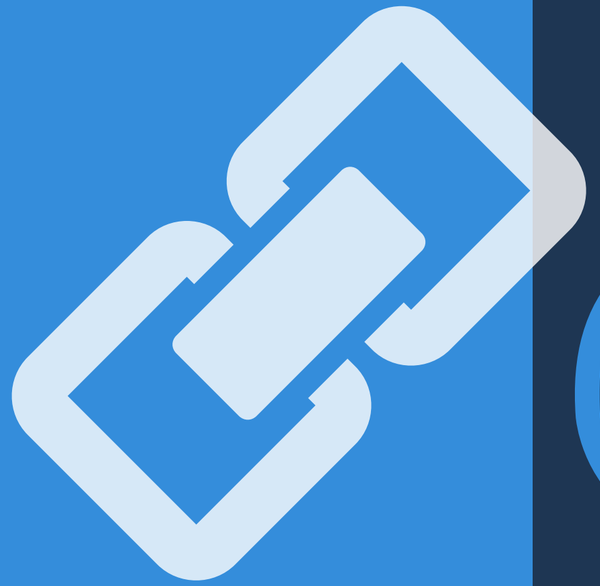


funkcje przyjmujące adres URL - możemy przekazać jako argument takiej funkcji spreparowany przez nas url

Można wykorzystać:
- obiekt location i jego metody
- właściwość atrybutu href
- właściwości atrybutu src

W tym przypadku groźne jest przypisanie URL-a z protokołem javascript:, który pozwala na wykonanie własnego kodu JS.

np. 'javascript:alert(1)'



FUNKCJE PRZYJMUJĄCE ADRES URL

usypia czujność

Przykład

Fragment kodu

eval("console.log('Hello " + user + " !')"); , gdzie np. zmienna user jest podana przez urzytkownika

Metoda ataku

zmieniamy wartość user na wartość zawierającą nasz kod

user = " '); alert(1)// ";

w konsekwencji wykona się

console.log('Hello '); alert(1)// !

Obrona

Zaleca się aby nie używać funkcji tego typu

Ochrona przed DOM XSS

-Nie używanie funkcji typu eval

- Nie uzywać funkcji i właściwości przypisujących bezpośrednio kod HTML do elemntów drzewa DOM

-uważać przy przekierowaniu użytkonika pod adres URL

Przykłady z życia

-Google 2005r
-CBS News i BBC 2006r
-Włoski bank Banca Fideuram 2008r
-Sammy worm 2005r
-Facebook 2011r
-CIA 2011r
-Fortnite 2019r
-British Airways 2018r
-eBay 2016r

Zadanie 1

1. Atak reflected XSS

Aby wykonać to zadanie, należy znaleźć na stronie miejsce, gdzie jest możliwe wykonanie ataku reflected XSS. Należy wstrzyknąć kod JavaScript i wyświetlić ciasteczka użytkownika. W odpowiedzi należy podać użyty kod oraz screen potwierdzający autentyczność rozwiązania.

2. Atak stored XSS

Aby wykonać to zadanie, należy znaleźć miejsce, w którym można dodać jakiś wpis widoczny dla wszystkich użytkowników. Następnie należy w nim umieścić kod wyświetlający ciasteczka danego użytkownika. Aby sprawdzić, czy kod działa poprawnie, należy zalogować się na innego użytkownika i sprawdzić, czy po wejściu na odpowiednią stronę na ekranie zostaną wyświetlone ciasteczka. W odpowiedzi należy podać użyty kod oraz screen potwierdzający autentyczność rozwiązania.

Zadanie 2

1. Atak DOM XSS

Zadanie to polega na wykorzystaniu luki w zabezpieczeniach strony, która wykorzystuje niebezpieczną funkcję document.write do zapisywania danych na stronie. Funkcja ta wykorzystuje dane z location.search, którymi można sterować za pomocą adresu URL witryny. Co można wykorzystać w ataku.

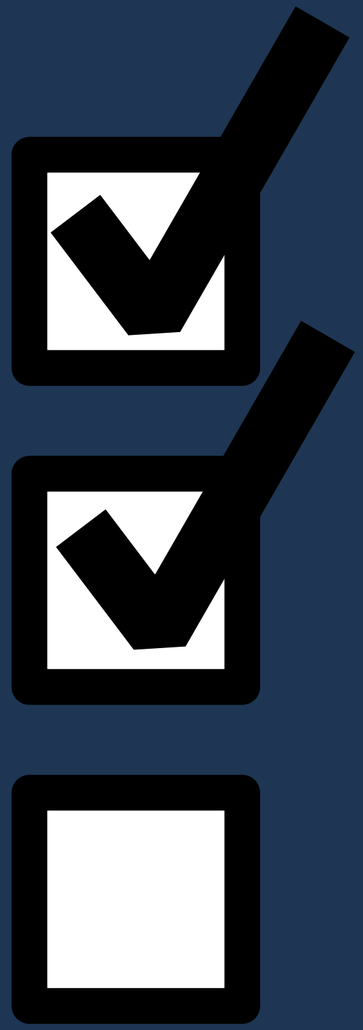
2. Atak reflected DOM

W tym przykładzie serwer przetwarza dane z żądania i powtarza je w odpowiedzi. Tworzy się w ten sposób niebezpieczna luka, która można wykorzystać. Aby wykonać to zadanie, należy wstrzyknąć kod, który wywoła funkcję alert().

Zadanie 3

Zadanie polega na przeprowadzeniu ataku stored XSS, który umożliwi nam zalogowanie się na cudze konto bez loginu i hasła. W tym zadaniu będziemy używać dziurawej aplikacji DVWA. Należy w niej dodać odpowiedni wpis w zakładce XSS (Stored), który wyśle ciasteczka na serwer, który będzie nasłuchiwał. Serwer należy uruchomić na WSL-u, wpisując następującą komendę ”nc -lvp [port]” gdzie [port] to numer portu, na którym będziemy nasłuchiwali. Należy sprawdzić, na jakim poziomie zabezpieczeń możliwy jest atak. W odpowiedzi należy podać użyty kod oraz screen potwierdzający autentyczność rozwiązania.

Zadanie 4



Przeprowadź atak stored XSS na aplikacji DVWA. Dodaj wpis który, umożliwi przesyłanie danych wpisanych, przez innego użytkownika, w formularzu(Name, Message) na nasz nasłuchujący serwer stworzony tak jak w zadaniu poprzednim.W odpowiedzi należy podać użyty kod oraz screen potwierdzający autentyczność rozwiązania.