User vs User, who wins? DACYS – TP 6

Nuno Peralta

Nsp@isep.ipp.pt

Como é que um utilizador ataca outro utilizador?

Como funciona o HTML

http://site.com/?input=teste

```
<html>
<body>
<user input>
</body>
</html>
```

http://site.com/?input=teste

```
<html>
<body>
teste
</body>
</html>
```

http://site.com/?input=<script>alert(1);</script>

```
<html>
<body>
<user input>
</body>
</html>
```

http://site.com/?input=teste

```
<html>
<body>
<script>alert(1);</script>
</body>
</html>
```

O que é XSS (Cross-Site Scripting)

With cross-site scripting, an attacker injects their own code onto a legitimate website; the code then gets executed when the site is loaded onto the victim's browser.

XSS primarily exploits vulnerabilities existing in programming languages such as Flash, ActiveX, VBScript and JavaScript. JavaScript is most common due to its close integration with most browsers. Because XSS can exploit these popular platforms, XSS attacks are both dangerous and common.

https://www.verizon.com/business/resources/articles/s/how-to-mitigate-cross-site-scripting/

Tipos de XSS (Cross-Site Scripting)

Reflected

Dom

Stored

"Samy didn't want to be everyone's hero. He didn't even want new friends.

But thanks to a few clever lines of code, in less than a day, he became the "hero," and a "friend," to more than a million people on what was, at the time, the most popular online social network, MySpace."

https://www.vice.com/en/article/wnjwb4/the-myspace-worm-that-changed-the-internet-forever

Como é possível?

Que tipo de XSS foi utilizado?

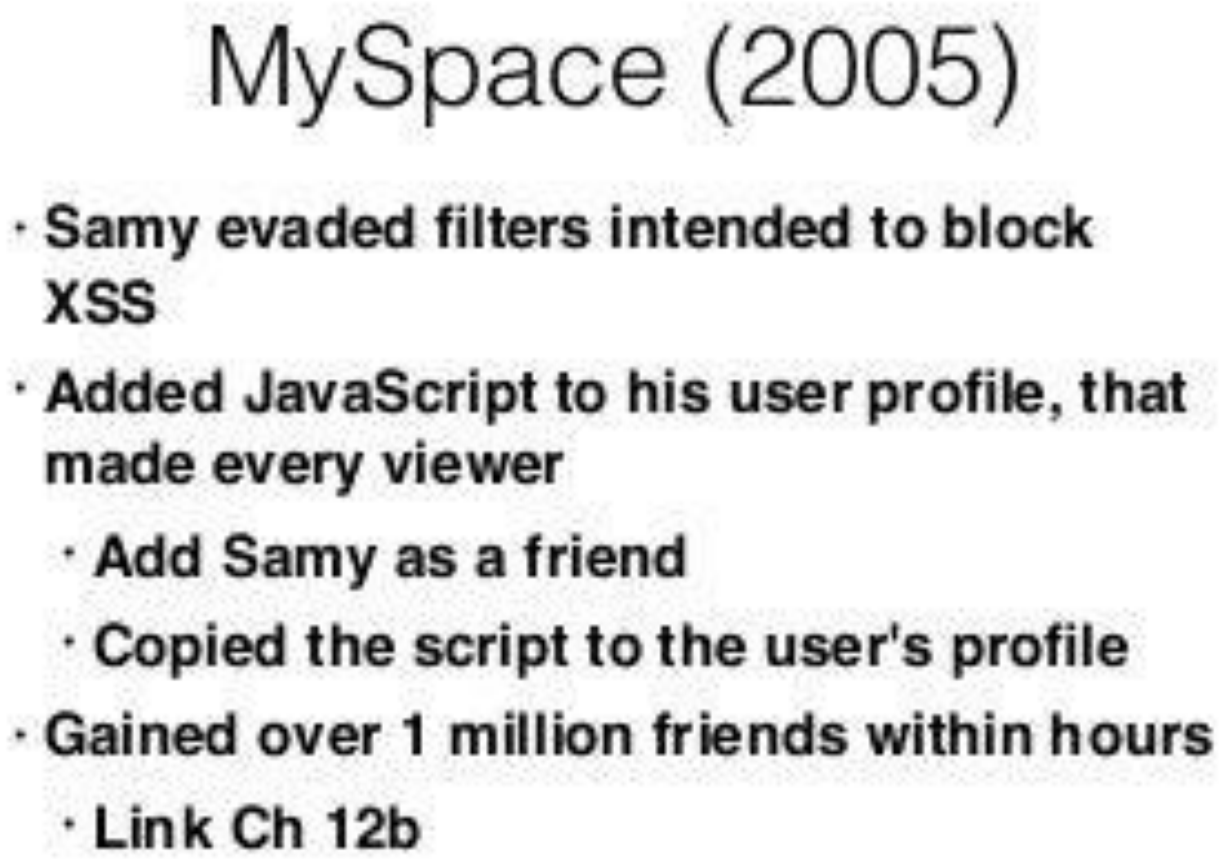
A história

It was around midnight on October 4, 2005, in Los Angeles, when Samy Kamkar, then a 19-year-old hacker, released what has come to be known as the "Samy worm," perhaps the fastest-spreading computer virus of all time, a virus that changed the world of web security forever.

Kamkar, who had dropped out of high school at 16 and founded a software startup called Fonality at 17, says he just wanted to impress his techie friends, nothing more. Everything started around a week earlier, Kamkar tells me. At the time, MySpace gave users a lot of freedom to customize their profiles, allowing the use of HTML code, which resulted in colorful, and often painful to look at profiles.

"Once I was able to do that, I realized I was able to actually do anything on the page."

Payload


MySpace (2005)

- Samy evaded filters intended to block XSS
- Added JavaScript to his user profile, that made every viewer
  - Add Samy as a friend
  - Copied the script to the user's profile
- Gained over 1 million friends within hours
  - Link Ch 12b

British Airways

"In 2018, British Airways was attacked by Magecart, a high-profile hacker group famous for credit card skimming attacks. The group exploited an XSS vulnerability in a JavaScript library called Feedify, which was used on the British Airway website. "

British Airways

"Attackers modified the script to send customer data to a malicious server, which used a domain name similar to British Airways. The fake server had an SSL certificate, so users believed they were purchasing from a secure server. They succeeded in performing credit card skimming on 380,000 booking transactions before the breach was discovered."

Como mitigar?

- Whenever possible, prohibit HTML code in inputs. Preventing users from posting HTML code into form inputs is a straightforward and effective measure.

- Validate inputs. If you're going to accept form inputs, validating the data to ensure it meets specific criteria will be helpful

- Secure your cookies. Setting rules for your web applications defining how cookies are handled can prevent XSS and even block JavaScript from accessing cookies.

- Sanitize data. Similar to validation, sanitizing occurs after data has been posted but before it is executed. Look for online tools like HTMLSanitizer to sanitize HTML code online for XSS vulnerabilities.

- Use a web application firewall (WAF). Rules can be created on a WAF to specifically address XSS by blocking abnormal server requests. A robust WAF should be a key component of your organization's security strategy, as it can also prevent SQL injection attacks, distributed denial-of-service attacks and other common threats.

Dúvidas?