Cross-Site Scripting (XSS)

Definition from "Cross-site Scripting (XSS)" on OWASP (retrieved Oct. 31, 2018):

> "Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user."

XSS attacks fall within three (3) classes, although attacks can include more than one class:

- Reflected
- Stored
- DOM-based

Example of a Reflected attack

1. xss_reflected_sample_form.html
2. xss_reflected_sample_expected_results.html
3. xss_reflected_sample_attack_results.html

Example of a Stored Attack-- store a script in a database field, using a form typically

1. xss_stored_profile_form.html
2. xss_stored_search_form.html
3. xss_stored_search_expected_results.html
4. xss_stored_search_attack_results.html

Example of a DOM-based Attack

Example of potential impact of XSS attacks: Samy worm

- In 2005, Samy Kamkar updated his personal profile web page on social networking site MySpace.com.
- Kamkar included JavaScript exploit code on the page.
- When a MySpace user would view Samy's profile, the script would add Samy as a friend and include in the user's profile the statement, "but most of all, samy is my hero".
- Additionally, the script would copy itself to the user's profile web page so that visitors to that user's profile page would be similarly infected.
- The Samy Worm infection grew exponentially to over 1,000,000 user profiles within 24 hours.
- MySpace shut down its website in order to remediate the damage and remove the vulnerability.

- Same Origin Policy (SOP) was designed to restrict the content that a webpage could access, but XSS skirts around SOP.
- "Same-origin policy" from Wikipedia (retrieved Oct. 31, 2018):

  > "Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only if both web pages have the same origin. An origin is defined as a combination of URI scheme, hostname, and port number."

- Otherwise, filters can be used to sanitize data so that it cannot be used to create XSS attacks, but filters are only as effective as their design and deployment. See CERT: "Understanding Malicious Content Mitigation for Web Developers".

Question: What do these types of attack all have in common?

Answer: They all rely on the fact that data and command structure are intermingled; that is, data and command structure are placed together within the packet and separation is achieved solely by text delimiters.

- Although the HTTP packet itself does not intermingle like the HTML payload does, HTTP headers may also be disrupted because separation is achieved only by text delimiters.
- Therefore, a secure distributed system must take into account not only the protocol for transfer but also the specification of the payloads.

Solution

- Jeremy Crockett and Matthew Feingold, research students
- "All potentially untrustable external data sources, which are the root of XSS attacks, are...tokenized." Tokens are ultimately replaced with Document Object Model (DOM) strings, which are incapable of triggering any type of attack.
- Because the resulting inert strings are still present, the user will be alerted to the presence of the attempted attack and will be able to forward valuable information about the attack to a system administrator for the website.
- Subsequent students in our lab, Harsh Patel and Yu (Jack) Chai, devised a simplified version of the Crockett-Feingold method to use with Visual Basic on the .NET platform.
- In the Patel-Chai simplified approach, user input is converted to a DOM string using the Xelement of the VB XML library. Any occurrences of angle brackets ("<" and ">") are removed from the string, and any occurrence of "&" is replaced with an actual '&' character. These replacements are needed in order to prevent the DOM string from interfering with SQL queries.