What is the Future of Automated XSS Defense Tools?

Jim Manico

SANS AppSec March 8, 2011 (4:20pm - 5:20pm)

INFRARED SECURITY

Jim Manico

- Managing Partner, Infrared Security
- Web Developer, 15+ Years

- OWASP Connections Committee Chair
- OWASP ESAPI Project Manager
- OWASP Podcast Series Producer/Host

- Kauai/Hawaii Resident with wife Tracey

Eliminating XSS Today is Challenging

INFRARED SECURITY

1. All untrusted data must first be canonicalized
   Reduced to simplest form

2. All untrusted data must be validated
   Positive Regular Expression Rule
   Blacklist Validation

3. All untrusted data must be contextually sanitized/encoded
   Difficult because there are multiple contexts
   - HTML Body
   - HTML Attribute
   - URI Resource Locator
   - Style Tag
   - Event handler
   - Within Script tag

(1) Auto-Escaping Template Technologies

INFRARED SECURITY

- **XHP from Facebook**
  - Makes PHP understand XML document fragments similar to what E4X does for ECMAScript
- **Context-Sensitive Auto-Sanitization (CSAS) from Google**
  - Runs during the compilation stage of the Google Closure Templates to add proper sanitization and runtime checks to ensure the correct sanitization.
- **Java XML Templates (JXT) from OWASP**
  - Fast and secure XHTML-compliant context-aware auto-encoding template language that runs on a model similar to JSP.

(2) Javascript Sandboxing

INFRARED SECURITY

- **Capabilities JavaScript (CAJA) from Google**
  - Applies an advanced security concept, capabilities, to define a version of JavaScript that can be safer than the sandbox
- **JSReg by Gareth Heyes**
  - Javascript sandbox which converts code using regular expressions
  - The goal is to produce safe Javascript from a untrusted source
- **ECMAScript 5**
  - Object.seal(obj)
    Object.isSealed(obj)
  - Sealing an object prevents other code from deleting, or changing the descriptors of, any of the object's properties

(3) Browser Protections

- **Content Security Policy**
  - Website administrators specify which domains the browser should treat as valid sources of script.
  - The browser will only execute script in source files from the white-listed domains and will disregard inline scripts and event-handling HTML attributes.
  - Sites that never want to have JavaScript included in their pages can choose to globally disallow script.
- **Reflective Defense XSS in Chrome**
- **IE 8 Cross-Site Scripting Filter**

INFRARED SECURITY

THANK YOU!

jim@infraredsecurity.com

SANS AppSec March 8, 2011