JavaScript Infection Model

By Aditya K Sood and Richard J. Enbody

Join the Discussion Connect

Advancements in Web 2.0 technologies have enhanced Internet functionality but at the same time have created numerous threats to the World Wide Web. This paper talks about the negative nature of JavaScript, which is exploited heavily by malware writers to spread infections throughout the online world.

Abstract

Advancements in Web 2.0 technologies have enhanced Internet functionality but at the same time have created numerous threats to the World Wide Web. The biggest issue the online world is grappling with is web malware, which is an outcome of intensive exploitation of web vulnerabilities. This paper talks about the negative nature of JavaScript, which is exploited heavily by malware writers to spread infections throughout the online world.

Exploitation Shift

There is always an element of discrepancy present between current and upcoming technologies. With the advent of powerful operating system protection mechanisms, the attack surface has shifted to web exploitation vectors because memory exploitation is becoming tougher for the attackers. Technologies such as Microsoft Data Execution Protection (DEP),[1] Address Space Layout Randomization (ASLR),[2] and GS cookies[3] have circumvented the attack and exploitation of system-level vulnerabilities. The use of string functions is completely isolated from systems as they are considered as a base for buffer overflow attacks. Exploitation has shifted from system vulnerabilities to web vulnerabilities.

The attack landscape of the Web has a panorama of exploitations that are proliferating day by day. With the rise of blogs, wikis, atom feeds, RSS, and others, the insecurity level is increasing[4] in spite of versatile functions. These new technologies have made the Web flexible and robust by allowing the inclusion of content from third-party sites and sending content to other domains. In reality, data from the third parties cannot be verified against presence of potential malware. As a result malware can accompany the data back into the parent website without restriction and continue spreading across the Web. Security considerations have to be undertaken in the best possible manner to combat web exploitation.

New technologies

With the advent of new technologies, the sphere of attack surface vulnerability has widened. The Web is getting exposed to identity theft, exploitation, scams, phishing, redirection vulnerabilities, cross site scripting (XSS), and cross site request forgery (CSRF).[5] CSRF, for example, is a type of attack in which HTTP requests are sent in a stealth manner without the knowledge of user. This type of attack allows the attacker to execute commands and requests on user's behalf. The inherent vulnerabilities in web applications are exploited by various application injections such as PHP, ASP, LDAP, SQL, and DOM (Document Object Model).[6] The injections are widely used to manipulate the content, steal information, and spread malware. One step ahead is HTTP Protocol manipulation comprising of attack type Response Splitting,[7] which bypasses browser protection mechanisms by splitting the HTTP response from the server thereby fooling the browser to interpret two responses instead of one.

1 Data Execution Prevention, http://support.microsoft.com/kb/875352.

2 Address Space Layout Randomization, http://blogs.technet.com/b/security/archive/2006/05/26/430538.aspx.

3 GS, http://blogs.technet.com/b/srd/archive/2009/03/20/enhanced-gs-in-visual-studio-2010.aspx.

4 RSS Attacks, http://www.techspot.com/news/20098-increased-rss-malware-attacks-predicted.html.

5 Cross Site Request Forgery, https://www.isecpartners.com/files/CSRF_Paper.pdf.

6 DOM XSS, http://www.webappsec.org/projects/articles/071105.shtml.

7 HTTP Response Splitting, http://www.securiteam.com/securityreviews/5WP0E2KFGK.html.