Security Spotlight articles discuss recent noteworthy threats that users may encounter and should be aware of while surfing the Web.

XSS ATTACK HITS *YOUTUBE*

The cross-site scripting (XSS) attack on YouTube, *which allowed random code to be embedded in comments, reminds us that XSS vulnerabilities and the risks they bring still exist. In fact, these threats may never entirely disappear, as sites that inadequately implement user input verification processes will always remain potential cybercriminal targets.*

What Is XSS?

XSS is a security vulnerability that enables attackers to inject unwanted executable client-side scripts (usually JavaScript, sometimes VBScript) into a trusted Web page.

XSS attacks enable cybercriminals to inject code into otherwise benign pages, modify affected sites, inject forms into the said pages, steal victims' user names and passwords, and enable other phishing techniques.

XSS vulnerabilities are caused by flaws in Web applications that cause them to fail to properly validate user input. This can possibly cause legitimate sites to serve injected arbitrary script code to their victims via XSS exploits.

The Dangers That XSS Poses

The threats that XSS poses arise when a malicious code inserted into sections of a legitimate site, such as comment boxes and Web forms, redirects users to malicious sites. XSS vulnerabilities can cause a variety of problems for unsuspecting Web surfers. These can range from mere annoying pop-up windows to complete credential compromise and information and data theft. Code injection via XSS allows cybercriminals to steal information such as login and banking credentials. They can also use cookie stealers to instigate user account theft, as this grants them administrative privileges on hacked sites.

XSS is usually the first step in a larger attack. Some XSS attacks even incorporate disclosure of users' session cookies, allowing perpetrators to gain control over victims' browsing sessions and to take over their accounts and to hijack their HTTP sessions. XSS vulnerabilities can also redirect users to fake site pages that cybercriminals craft in order to trick their victims into revealing their credit card or banking details. The rationale behind committing XSS attacks vary. Regardless of purpose, however—website defacement, phishing, or spam—an XSS attack always leads to undesirable effects, as these give cybercriminals the ability to take over targeted sites and to alter their functions.

TREND MICRO

Security Spotlight articles discuss recent noteworthy threats that users may encounter and should be aware of while surfing the Web.

Recent XSS Attacks

Earlier this month, we learned about an XSS vulnerability found on Google-owned video-sharing site *YouTube.* The perpetrators of the attack added comments containing HTML code to a number of videos on the site, which led to code execution on users' browsers. Several reports indicate that videos of the popular teen singer Justin Bieber were targeted by these attacks, which put site visitors' cookies at risk. Some visitors saw "tasteless messages pop up about the teen star and were also redirected to external sites with adult content," according to *Network World.*

Google immediately reacted by temporarily deactivating the pages' Comments feature and by fixing the vulnerability. However, comment posting still remains disabled for some of the targeted videos. Luckily, the now-patched XSS vulnerability on *YouTube* didn't affect other *Google* accounts and the attack did not involve malware infections.

Figure 1. YouTube *XSS attack infection diagram*

In January, another notable XSS attack occurred. Cybercriminals reportedly took advantage of XSS vulnerabilities on the site *www.eu2010.es,* which kept track of the events that transpired during the 2010 Spanish Presidency of the Council of the European Union. Unidentified attackers defaced the website by replacing Spanish Prime Minister Jose Luis Rodriguez Zapatero's picture with Mr. Bean's, accompanied by the message, "Hi there," on the home page.

TREND MICRO

Though *BBC News* reported that the site itself was not attacked, the short-lived compromise was still a clear demonstration of how XSS exploits can result in damaging changes to a legitimate Web page. Though it appears that there was no malicious intent involved in the *YouTube* attack, the same vulnerability found may be used as an entry point for more dangerous attacks and should thus not be underestimated.

Looking at the Bigger Picture

XSS attacks are gaining prevalence as the trends in website design move toward providing more user interactivity. The problem lies in lack of site administrator awareness of the dangers that XSS vulnerabilities pose and the damage attacks using them cause.

Website operators can always take palliative measures by deploying a mechanism to prevent users from embedding active content such as JavaScript in comments and profiles. However, the cost of fixing these problems is bound to increase, depending on the severity of the exploits used. Users will eventually lose confidence and, in effect, interest in the trusted sites.

Countering XSS Attacks

The administrators of sites that accept content from users have the obligation to filter out or to sanitize any content to minimize with the purpose of eliminating the possibility of XSS vulnerability exploitation. They should always be suspicious of inputs until these have been checked and verified as nonmalicious.

At the most fundamental level, the burden of preventing XSS attacks falls upon site developers. The administrators of sites that accept content from users have the obligation to filter out or to sanitize any content with the purpose of eliminating the possibility of XSS vulnerability exploitation. They should always be suspicious of inputs until these have been checked and verified as nonmalicious.

The use of scripts as the first step toward instigating system infection as well as executing malicious routines has been observed in recent Web attacks and is bound to continue in the future. These attacks present a serious threat to users since they require minimal user interaction. All it takes is a single visit to a tainted site for the malicious routines to commence.

To prevent Web pages from being exploited on both the client and server sides, the Open Web Application Security Project (OWASP) Community provides some tips and simple rules on XSS attacks on this XSS prevention cheat sheet.

In the meantime, users must exercise caution when surfing the Web. They should be wary of messages from friends inviting them to click a link or message, especially if the said link or message has been sent multiple times by different people.

In addition, users should make sure that their systems are protected by a security solution that effectively blocks malicious URLs and that prevents malware and malicious scripts from running.

TREND MICRO

References:

- BBC. (January 4, 2010). *BBC News.* "Mr. Bean Replaces Spanish PM on EU Presidency Site." http://news.bbc.co.uk/2/hi/8440554.stm (Retrieved July 2010).
- Bojan Zdrnja. (July 4, 2010). *Internet Storm Center.* "Stored XSS Vulnerability on *YouTube* Actively Abused?" http://isc.sans.edu/diary.html?storyid=9130 (Retrieved July 2010).
- Homeland Security. (June 18, 2010). *CWE.* "CWE-79: Improper Neutralization of Input During Web Page Generation (Cross-Site Scripting)." http://cwe.mitre.org/data/definitions/79.html (Retrieved July 2010).
- Joseph Pacamarra. (May 31, 2008). *TrendLabs Malware Blog.* "XSS Methods Also Seen Being Used in Mass Compromises." http://blog.trendmicro.com/xss-methods-also-seen-being-used-in-mass-compromises/ (Retrieved July 2010).
- Juan Carlos Perez. (July 4, 2010). *Network World.* "Google Acknowledges *YouTube* Hack." http://www.networkworld.com/news/2010/070410-google-acknowledges-youtube.html (Retrieved July 2010).
- OWASP. (June 8, 2010). "XSS (Cross-Site Scripting) Prevention Cheat Sheet." http://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet (Retrieved July 2010).
- Rik Ferguson. (January 5, 2010). *CounterMeasures.* "Mr. Bean Comes Out of Retirement, Takes over Spain." http://countermeasures.trendmicro.eu/mr-bean-comes-out-of-retirement-takes-over-spain/ (Retrieved July 2010).
- Steve Christey (editor). (February 15, 2010). *CWE.* "2010 CWE/SANS Top 25 Most Dangerous Software Errors." http://cwe.mitre.org/top25/#CWE-79 (Retrieved July 2010).
- Trend Micro Incorporated. (December 2009). *TrendWatch.* "The Future of Threats and Threat Technologies: How the Landscape Is Changing." http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/trend_micro_2010_future_threat_report_final.pdf (Retrieved July 2010).
- Wikimedia Foundation Inc. (July 28, 2010). *Wikipedia.* "Mr. Bean." http://en.wikipedia.org/wiki/Mr._Bean (Retrieved July 2010).
- Wikimedia Foundation Inc. (July 29, 2010). *Wikipedia.* "Cross-Site Scripting." http://en.wikipedia.org/wiki/Cross-site_scripting (Retrieved July 2010).

TREND MICRO