



# NJCCIC

NJ CYBERSECURITY AND COMMUNICATIONS INTEGRATION CELL

## Cross-Site Scripting: Many Websites Remain Vulnerable to Common Web Exploit

November 20, 2015

TLP: WHITE | *The NJCCIC assesses with moderate confidence that many websites remain at high risk of cross-site scripting (XSS), one of the most commonly exploited web application security vulnerabilities.* [XSS](#) is a code injection tactic—similar to [SQL injection](#)—in which a hacker inputs malicious code into a legitimate web application or website that is then executed in a user’s web browser, often to compromise user credentials or take control of the user’s session. [XSS](#) exploits weaknesses in common scripting languages present in internet browsers, such as JavaScript, HTML, and Flash. The InfoSec Institute, provider of information security training, classifies XSS as one of the [most dangerous](#) website vulnerabilities and security researchers have identified the flaw in many [high-traffic websites](#), often disclosed on a [website](#) dedicated to XSS vulnerabilities. Unsuspecting users can fall victim to XSS exploits in the course of normal activity on otherwise trusted websites; however, these tactics can also be initiated through phishing emails, when a user is deceived into clicking a link that executes the malicious script. Websites that request, transmit, and store user data are at the highest risk of XSS exploits, such as social media, retail, and government sites.

- According to cybersecurity firm High-Tech Bridge, XSS accounts for [80%](#) of website security flaws. In a recent study, another security firm, Tinfoil Security, tested the networks of 557 state universities and discovered that [25 percent](#) were vulnerable to XSS.
- The two main [types of XSS](#) attacks are reflected (non-persistent) and stored (persistent). In a reflected attack, malicious script is injected into a component of a website, such as a [search form](#), and reflected off the web server and executed within the victim’s browser. A stored attack occurs when malicious script is injected and stored on a targeted server, such as in a database or forum, and executed when a user visits the website. Whereas the reflected and stored types are server-side tactics, another XSS tactic known as [DOM-based](#) is a client-side attack.
- The information technology firm Cisco recently revealed an XSS vulnerability in the WeChat page of their SocialMiner product, a social media brand management tool. Cisco’s [advisory](#) stated the vulnerability could allow an unauthenticated, remote attacker to send a malicious script to an unsuspecting user, and that there is no security patch or workaround to mitigate the flaw.

The [consequences](#) of a successful XSS exploit include the compromise of user files or sensitive information, installation or spreading of malware, hijacking an account, redirecting victims to malicious sites, or modifying the presentation of content in the user’s browser. The manipulation of content can be problematic as an XSS vulnerability could allow an actor to modify proprietary documents such press releases, news publications, or other official communications, affecting the integrity of data and potentially impacting public trust and consumer confidence. Other [possible results](#) of XSS exploits include impersonating end-users, as well as hijacking users’ webcams and microphones.

### Mitigation

The responsibility of mitigating XSS exploits falls on website owners and developers to proactively scan and patch vulnerabilities, as well as maintain awareness of the latest threats and tactics. While [XSS](#) flaws can be easy to identify, they are often difficult to remove from a web application. The most effective identification method is to perform a comprehensive security review of the code and search for all locations where input from an HTTP request could make its way to the HTML output. [Users](#) can prevent most XSS exploits by disabling JavaScript and other plugins in their browsers, or enable browser settings to ask for user permission before running these scripts. It is also recommended to use browsers that offer script-blocking XSS protection and ensure all browsers, plugins, and software are kept up-to-date with the latest security patches. A comprehensive set of preventative XSS measures can be found on the [Open Web Application Security Project](#) (OWASP) site.

Traffic Light Protocol: **WHITE** information may be distributed without restriction.