

## Chapter 9

# Enhancing the Browser–Side Context–Aware Sanitization of Suspicious HTML5 Code for Halting the DOM–Based XSS Vulnerabilities in Cloud

**B. B. Gupta**

*National Institute of Technology Kurukshetra, India*

**Shashank Gupta**

*National Institute of Technology Kurukshetra, India*

**Pooja Chaudhary**

*National Institute of Technology Kurukshetra, India*

### ABSTRACT

*This article presents a cloud-based framework that thwarts the DOM-based XSS vulnerabilities caused due to the injection of advanced HTML5 attack vectors in the HTML5 web applications. Initially, the framework collects the key modules of web application, extracts the suspicious HTML5 strings from the latent injection points and performs the clustering on such strings based on their level of similarity. Further, it detects the injection of malicious HTML5 code in the script nodes of DOM tree by detecting the variation in the HTML5 code embedded in the HTTP response generated. Any variation observed will simply indicate the injection of suspicious script code. The prototype of our framework was developed in Java and installed in the virtual machines of cloud environment on the Google Chrome extension. The experimental evaluation of our framework was performed on the platform of real world HTML5 web applications deployed in the cloud platform.*

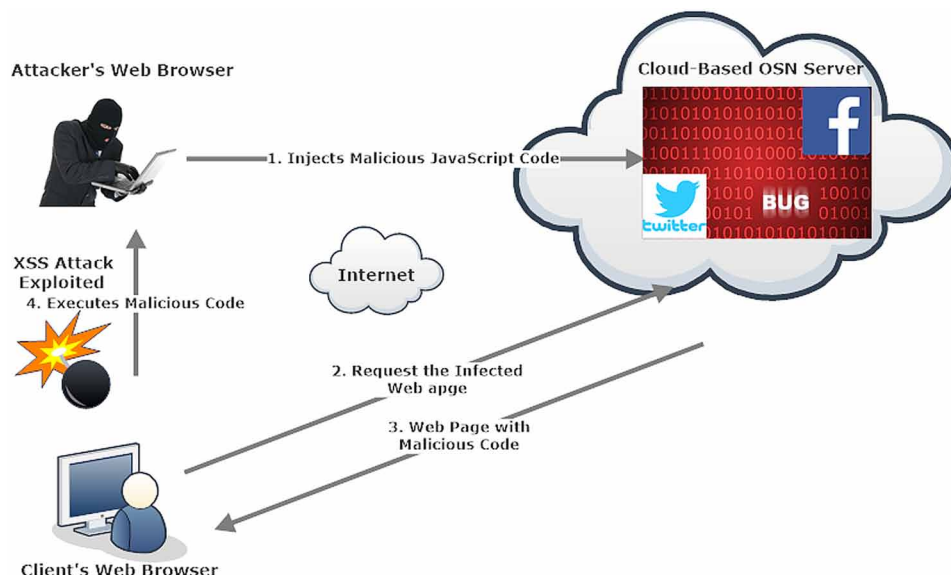
DOI: 10.4018/978-1-5225-3422-8.ch009

## INTRODUCTION

The tremendous explosion in cloud computing produced numerous security issues related to data security of cloud users (Dinh et al., 2013; Gupta et al., 2016c). The propagation of XSS worms are considered to be topmost threat originated in HTML5 web applications deployed in the framework of cloud infrastructure. In the contemporary era of cloud computing, cloud security has turned out to be a serious issue, as numerous on-demand resources are being offered by utilizing the virtualization technologies of cloud services (Modi et al., 2013). Instead of referring the outdated Internet settings for constructing an expensive setup, numerous commercial IT organizations are accessing the services of Online Social Networking (OSN) sites (such as Twitter, Facebook, LinkedIn, etc.) on the cloud platforms. In the modern era of Web 2.0 technologies and HTML5-based web applications, OSN is considered to be the most popular method for information sharing has drawn most of public attention. However, it is clearly known that the cloud settings are installed on the backbone of Internet. Therefore, numerous HTML5 web application vulnerabilities in the conventional Internet infrastructures also exist in the backgrounds of cloud-based environments.

The most prominent attack found on HTML5 web applications is the Cross Site Scripting (XSS) attack [Gupta et al. (2016a), Gupta et al. (2016b), Gupta et al. (2015a), Gupta et al. (2015b), Gupta et al. (2014)]. Figure 1 highlights the injection of XSS worm on the OSN web server deployed in the virtual machines of cloud platforms. XSS worms have turned out to be a plague for the cloud-based HTML5 web applications. Such worms steal the sensitive credentials of the active users by injecting the malicious HTML5 script code in the form of some posts on such web applications [Gupta et al. (2015c), Duchene et al. (2014), Shahriar et al. (2011), Doupe et al. (2013), Chandra et al. (2011), Xiao et al. (2014)]. Input sanitization is considered to be the most effective mechanism for alleviating and mitigating the effect of XSS worms from the cloud-based HTML5 web applications on the virtual machines of cloud platforms.

Figure 1. Exploitation of XSS attack on cloud platform



30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/enhancing-the-browser-side-context-aware-sanitization-of-suspicious-html5-code-for-halting-the-dom-based-xss-vulnerabilities-in-cloud/188209](http://www.igi-global.com/chapter/enhancing-the-browser-side-context-aware-sanitization-of-suspicious-html5-code-for-halting-the-dom-based-xss-vulnerabilities-in-cloud/188209)

## Related Content

---

### Adapting Agile Practices to Mobile Apps Development

Alberto Heredia, Javier Garcia-Guzman, Roberto Esteban-Santiago and Antonio de Amescua (2014). *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement* (pp. 63-82).

[www.irma-international.org/chapter/adapting-agile-practices-to-mobile-apps-development/100271](http://www.irma-international.org/chapter/adapting-agile-practices-to-mobile-apps-development/100271)

### Aspect-Oriented Self-Configuring P2P Networking in Mobile Environments: A Formal Specification and Verification

Phan Cong-Vinh (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications* (pp. 557-576).

[www.irma-international.org/chapter/aspect-oriented-self-configuring-p2p/66487](http://www.irma-international.org/chapter/aspect-oriented-self-configuring-p2p/66487)

### Model-Based Testing of Embedded Systems Exemplified for the Automotive Domain

Justyna Zander and Ina Schieferdecker (2010). *Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation* (pp. 377-413).

[www.irma-international.org/chapter/model-based-testing-embedded-systems/36350](http://www.irma-international.org/chapter/model-based-testing-embedded-systems/36350)

### Flexible Provenance Tracing

Liwei Wang, Henning Koehler, Ke Deng, Xiaofang Zhou and Shazia Sadiq (2011). *International Journal of Systems and Service-Oriented Engineering* (pp. 1-20).

[www.irma-international.org/article/flexible-provenance-tracing/55120](http://www.irma-international.org/article/flexible-provenance-tracing/55120)

### Establishing Ethos on Proprietary and Open Source Software Websites

Kevin Brock (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 915-935).

[www.irma-international.org/chapter/establishing-ethos-proprietary-open-source/77739](http://www.irma-international.org/chapter/establishing-ethos-proprietary-open-source/77739)