International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

XSS Worm Propagation and Detection in Online Social Network

Kolanoori Pravallika[1], B. Srinivas Reddy[2]

[1]Student, M. Tech CSE Department, Institute of Aeronautical Engineering, Hyderabad-500043, Telangana, India

[2]Associate Professor, H&BS Department, Institute of Aeronautical Engineering, Hyderabad-5000043, Telangana, India

Abstract: ***Cross-site scripting (XSS) vulnerabilities make it possible for worm to spread quickly to a broad range of users on popular web sites. Today, the detection of XSS worm has been largely UN explored. This paper proposes the first purely client-side solution to detect XSS worms. Our sight is that an XSS worm must spread from one user to another by reconstructing and propagating its payload. Our approach prevents the propagation of XSS worms by monitoring out going request that sends self-replicating pay loads. We intercept all HTTP request on the client side and compare them with currently embedded scripts. We have implemented a cross-platform Firefox extension that is able to detect all existing self-replicating XSS worms that propagate on the client side. Our test results show that it incurs low performance overhead and reports no false positive when tested on popular web-sites.***

Keywords: Security, Social Networks worms, Propagation dynamics, Modeling, Malware.

1. Introduction

Web application has drawn the attention of attackers due to their ubiquity and the fact that regulate access to sensitive user information. To provide users with a better browsing experience a number of interactive Web applications take advantages of the JavaScript language. The support for JavaScript, however Provides fertile ground for XSS attacks .According to a recent report from WASP, XSS worm vulnerabilities are the most prevalent vulnerabilities in Web applications. They allow attackers to easily by pass the same origin policy (SOP) to steal victims' private information or act on behalf of the victims.

XSS vulnerabilities exist because of the in appropriate validated user input. Mitigating all possible XSS attacks is in feasible due to the size and complexity of modern web application and the various ways that browsers invoke their JavaScript engines. Generally speaking there are two types of XSS vulnerabilities. Non-president XSS vulnerabilities also known as reflected XSS vulnerabilities, exist when user provided data are dynamically included in pages immediately generated by web servers; persistent XSS vulnerabilities also referred to as stored XSS vulnerabilities, exist when insufficiently validated user inputs all presidently stored on the server side and later displayed dynamically generated Web pages for others to read president XSS vulnerabilities allows more power full attacks then non-persistent XSS vulnerabilities as attackers do not need trick users into clicking specially crafted links. The emergency of XSS worms. Can raise the influence level of persistence XSS attacks in community driven web applications.XSS worms are special cases of XSS attacks in that they replicate themselves to propagate. Just like traditional worms to do. Different from traditional XSS attacks, XSS worm can collect sensitive information from a great number of users with in a shorter period of time because of their self propagating nature.

The threats that come from XSS worm are on the rise as attackers are switching their attention to major web sites, especially social network sites, to attack a board user based. Connecting among different users with in web applications provided channels for worm propagation. In community driven web applications.XSS worms tend to spread rapidly- Sometimes exponentially. For example the first well known XSS worm, the samy worm affected more than one million My space users in less than 20 hours in October 2005.my space which had over 32 million users at that time, was forced to shut down to stop the worm from further propagation. in April 2009,During the outbreak of the stalk daily XSS worm which hit twitter.com. Users became infected when they simply viewed the infected profiles of other user. We should a list the XSS worms in table1 common play grounds of XSS worm includes Social networking sites, forums, blogs and web-based email services.

At present as much research has been done to detect traditional worms or XSS vulnerabilities, little research has been done to detect XSS worm. This is because XSS worm usually contain site specific code which evades in put filters XSS worm can infect stealthily infected users by sending asynchronous HTTP request on behalf of users using the Asynchronous java script and XML(AJAX) technology. Spectator is the first java script worm detecting solution. It works by monitoring worm propagation traffic between browsers and specific web application. However it can only detect a JavaScript worm that have propagated for enough and is unable to stop an XSS worm in its initial stage. In addition Spectator requires server cooperation and is not easily deployable

In this paper we present the first purely client-side solutions to detect the propagation of self replicating XSS worm. Clients are protected against XSS worms on all web applications. We detect worm payload propagation on the client side by performing a string based similarity calculation. We compare out going request with scripts that are embedded in the currently loaded web page. Our approach is similar in spirit to traditional worm detection techniques that are based on pay load propagation

Licensed Under Creative Commons Attribution CC BY

monitoring we have developed the first effective client-side solution to detect XSS worm.

This paper makes the following contributes:

- We propose the first client-side solution to detect XSS worms. Our approach is able to detect self-replicating XSS worms in a timely manner, at the very early stage of their propagation.
- We have developed a cross-platform Firefox extension that is able to detect all existing XSS worms that propagate on the client side.
- We evaluated our extension on the top 100 most visited Web sites in the United States. Our results demonstrate that our extension produces no false positives and incurs low performance overhead, making it practical for everyday use.

2. Our Approach

2.1 Back ground of XSS worm Propagation

The Document Object Model (DOM) is a cross play form and language independent interface for valid HTML and well formed XML documents. It closely resembles the tree like logical structure of the document it models. Every node in a DOM tree represents an object in the corresponding document with the DOM programs and scripts can dynamically access and update the content, structure and style of the document.

In practice, building a flew less web application is a extremely difficult task due to the challenges of sufficiently sanitizing user-supplied data. Site-specific XSS vulnerabilities become a growing concern as a attackers discover that they can comprise more users by exploiting a single vulnerability with in a popular web site then by compromising numerous small web-site. Exploiting persistent vulnerabilities.XSS worms are normally stored on the servers of vulnerable web applications. The typical infection process of an XSS worm is as follows:

1. A user, Alice is lured into viewing a malicious web page that is dynamically generated by a comp remised web application. The web page for example can be profile page of Alice’s trusted friend.
2. The XSS worm payload, which is embedded in the dynamically generated web page, is interpreted by a JavaScript engine on the client side. During the interpretation process an XSS worm usually replicates its payload and injects the replicated payload into an outgoing HTTP request.
3. The crafted malicious HTTP request is sent to the Web application server on Alice’s behalf. By exploiting the server’s trust in Alice, the XSS worm also compromises Alice’s account. Later on when Alice’s friends visit her profile, their account will also be infected.

2.2 High-level Overview

Our goal is to detect the self-replicating characteristics of XSS worms with no modification to existing Web applications or browser architecture. To this end, we compare outgoing HTTP requests with scripts in the currently loaded DOM tree. Figure 1 shows the architecture of our client-side XSS worm detection mechanism. Our solution captures the essential self-propagating characteristics of XSS worms and protects users from infection. We choose to detect XSS worms on the client side because the propagation process of an XSS worm is normally triggered during the script interpretation process on the client side. Moreover, a client-side solution is easily deployable.

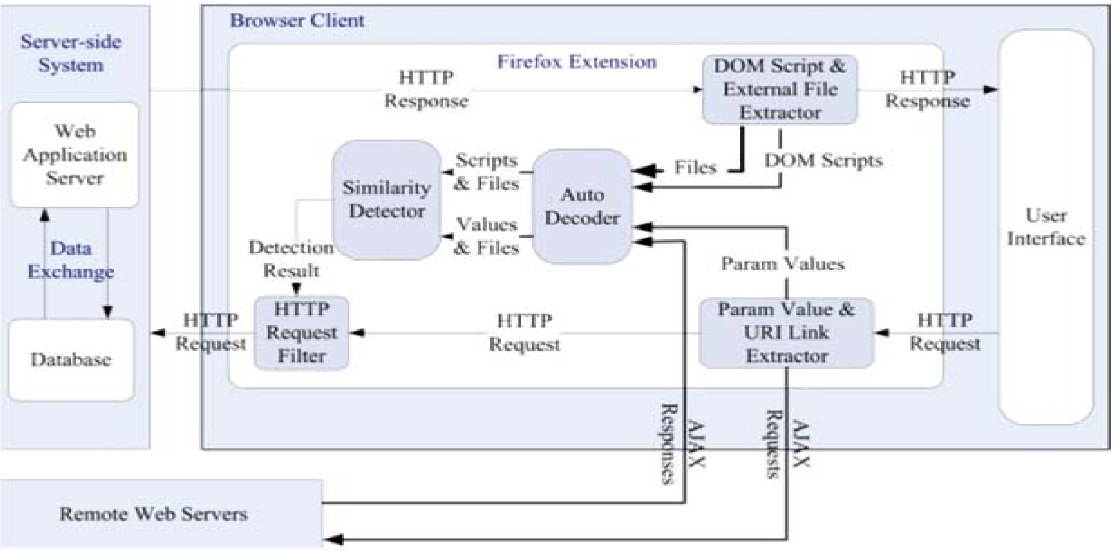

Figure 1: Client Architecture of XSS worm detection

3. The Spread of Social Network Worm

3.1 Social Network Worm Primer

Social Network Worm leverage social platforms to send worm mails or messages containing malicious links. Once the recipients are lured to read the infectious email attachments are visit the malicious web links, they will get infected and start spreading such worms copies are links to their friends found in the contact list, such as email address list different from scanning worms, the spread of social network worms depends on the topology of social networks and need much human involvement. In the early stage of the propagation, users have no knowledge of the newly emerging worm. After scientists detected it and encourage users to update their antivirus software, or when users become more skeptical of the email or links which seem out-of-character, the infected computers will be cleaned and the momentum of the spreading will slow down until the worms become extinct in networks.

3.2 Social Network Topology Characters

The social network topologies have the charecters:1)they can be thought of as a "semi-directed network "a graph in which some edges are directed and others are undirected. In this paper we can reciprocity rate to depict the fraction of edges which point back to the source.2)The in degree of nodes tends to match the out degree be equal to in degree and both follow the power-law distribution.3)Users how have large group of friends in social network tent to appear in the contact lists of many other Thus, any edge from one node to another node is chosen by the large-degree-preferential principle.4)The weight of each edge denotes the probability of an unwary user reading malicious messages from their friends. This probability is determined by human factor. The topologies are generated by using a 2k series method which was proposed and verified. In the case of XSS malware, user behaviors can be characterized by the tendency of visiting friends' profiles versus strangers' profiles, i.e., by the visiting-friends probability .We assume that a user's profile is always accessible to all of his/her friends. However, a person's profile may not be available to all strangers. As our proposed analytical model and simulation results will show, visiting friends more often than stranger helps to contain a malware within a community, slowing down its propagation.

4. Future Work

Install a security solution with proactive technologies on the computer. This way, you will be protected against malicious codes that spread through these networks, even if no previous attack has been launched.

- Keep the computer up-to-date: users must be aware of and resolve all the vulnerabilities that affect the programs installed on the computer.
- Don't share confidential information: If you access forums and chats to exchange information, talk, etc. remember not to provide confidential information (email addresses, credentials, etc.).
- Only provide the information necessary in the profiles: When creating user profiles, only provide the information necessary. If it requests private data like the email address, select the option to prevent other users from seeing the information, to ensure no users other than yourself and the administrator can access your data.
- Report crimes: If you observe inappropriate or criminal behavior (attempts to contact children, inadequate photos, modified profiles, etc.) you must inform the social network administrators.

5. Conclusion

We study the problem and propose a solution for containing worm propagation on dynamic social networks. Our method takes into account the network community structure and prevents worms from spreading out by distributing patches to most influential users in every community. In order to keep this structure up-to-date as the network evolves, we propose two adaptive algorithms for quickly and efficiently update the network community structure without re-computing. Experimental results on real social network Face book dataset confirm the strength and efficiency of our method when applied to practical dynamic social networks. As a centralized scheme, our method may not scale well with a real world network which could be quite large and improving this limitation would be our research direction in the future.

References

[1] V. D. Blonde, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Jul 2008.

[2] Bose, X. Hu, K.G. Shin, and T. Park. Behavioral detection of malware on mobile handsets. In MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services, pages 225–238, New York, NY, USA, 2008. ACM.

[3] Bose and K. G. Shin. Proactive security for mobile messaging networks. In WiSe '06: Proceedings of the 5th ACM workshop on Wireless security, pages 95–104, New York, NY, USA, 2006. ACM.

[4] Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. Aug 2004.

[5] R. Dantu, J. W. Cangussu, and S. Patwardhan. Fast worm containment using feedback control. IEEE Trans. Dependable Secur. Comput, 4(2):119–136, 2007.

[6] W. Enck, P. Tray nor, P. McDaniel, and T. La Porter. Exploiting open functionality in sms-capable cellular networks. In CCS '05: Proceedings of the 12th ACM conference on Computer and communications security, pages 393–404, New York, NY, USA, 2005. ACM.

[7] http://www.pcworld.com/article/155017/ face book virus turns your computer into a zombie.html.

[8] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 99(12):7821–7826, June 2002.

[9] http://news.cnet.com/koobface-virus-hits face book/.

Licensed Under Creative Commons Attribution CC BY