ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

Survey and Analysis of Client Side Detection of Content Sniffing Attack

Animesh Dubey[1], Ravindra Gupta[2], Gajendra Singh[3]

M.Tech Scholar, CSE, SSSIST, Sehore, Bhopal, India[1]

Assistant Professor (CSE/IT), SSSIST, Sehore, Bhopal, India [2]

HOD (CSE/IT), SSSIST, Sehore, Bhopal, India [3]

Abstract: From the last few years, the attacks based on web portals have caused significant harm to users. Many of these attacks occur through the exploitations of common security vulnerabilities in web-based programs. Given that, mitigation of these attacks is extremely crucial to reduce some of the harmful consequences. Web-based applications contain vulnerabilities that can be exploited by attackers at client-side (browser) without the victim's (browser user's) knowledge. Our work is intended to some exploitation due to the presence of security vulnerabilities in web applications while performing seemingly benign functionalities at the client-side. In this paper we survey the aspects of content sniffing attack mainly on client side and analyses how the control should be monitor from the server side after attack.

Keywords: Content sniffing, Security Measures, Web Based Attack, HTML Rendering

1. INTRODUCTION

If we analyze our daily routine, then we surprise to observe that we are relying on Internet. It is our need. For example email, e-shopping, trading, game etc. In the meantime we share our crucial and confidential data by HTML browser. Most of the data we share is text files, doc file, PDF file and Images. So we are very much concern on the security issue when we exchange the data from source to destination. We can understand the phenomena better in terms of client and server side data exchange. Sending data usually reside on a server-side and are accessed from its client-side [16]. There are some approaches which is either applied on client side as well as the server side but overall the approaches are not well enough to protect with the vulnerabilities. As a result users are fear and sometimes he/she may be suffering from those vulnerabilities [16].

If we understand the above scenario then we better understand and realize the unwanted security threats.
personal information for example phishing websites [1] instead of providing legitimate functionalities. Thus, the mitigation of web-based security vulnerability exploitations is extremely important to reduce some of the consequences. Phishing is a rapidly growing problem, with 9,255 unique phishing sites reported in June of 2006 alone [2]. It is unknown precisely how much phishing costs each year since impacted industries are reluctant to release figures; estimates range from $1 billion [3] to 2.8 billion [4] per year. For this reason we study a number of common program security problems and vulnerabilities [5][6]. Our study focuses that the number of web-based attacks has increased in recent years [7][8], existing research has addressed a subset of security vulnerabilities in web applications for example SQL Injection. Some security encryption technique like RSA is also suggested in [9]. After observation from several research by different authors, we analyze there are several numbers of vulnerabilities are still in the communication process when we want to access data from the web. So in this paper we want to survey the aspects of content sniffing attacks. What are the major precautions considered by different authors with their pros and cons are discussed.

The remaining of this paper is organized as follows. In Section 2 we discuss about content sniffing attack. The related work in section 3.In section 4 we discuss about problem domain. In section 5 we discuss the analysis. The conclusions and future directions are given in Section 6. Finally references are given.

2. CONTENT SNIFFING ATTACK

Content sniffing is a way of attempting or deducing the file format or change the content. It is also called media type sniffing (MIME Sniffing). The files are uploaded by attackers that contain malicious contents or which is intentional payloads. These files seem benign when we consider their content types or Multipurpose Internet Mail Extension (MIME) information. For example, a GIF file having a MIME image/gif might contain JavaScript code

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

(<script>...</script>). An attack occurs when a victim's browser renders a non-HTML file as an HTML file. A successful attack might result in severe consequences such as stealing of session information and passing information to third party websites. The attack can be the beginning points of other known exploitations such as phishing [10].

Table 1: Examples of File and MIME Types

File Type	MIME Type
HTML	Text/html
Textual Data	Text/plain data
Java Script	Application/java script
PDF	Application/PDF

In this type of attack attacker create an environment in which the user of the browser having no knowledge of attack. It feels like an authenticate data. The victim of the above, then downloads the file for use and rendering. This will be possible by the content header and content body type. Content-Type header indicates the MIME of the content. A MIME is a two-part identifier for file formats on the Internet [11]. For example, the MIME of a GIF file is image/gif. Table 1 shows some examples of file types and corresponding MIMEs. Ideally, a server should send the MIME of a file in the Content-Type header. Based on the header, a browser determines how to present the file.

3. RELATED WORK

In 2010, Hossain Shahriar et al. [12] discuss about Cross Site Request Forgery (CSRF) which allows an attacker to perform unauthorized activities without the knowledge of a user. An attack request takes advantage of the fact that a browser appends valid session information for each request. As a result, a browser is the first place to look for attack symptoms and take appropriate actions. According to the author Current browser-based detection methods are based on cross-origin policies that allow white listed third party websites to perform requests to a trusted website. To alleviate these limitations, they present a CSRF attack detection mechanism for the client side. Their approach relies on the matching of parameters and values present in a suspected request with a form's input fields and values that are being displayed on a webpage (visibility). To overcome an attacker's attempt to circumvent form visibility checking, they also compare the response content type of a suspected request with the expected content type.

In 2011, Misganaw Tadesse Gebre et al. [13] proposed a server-side ingress filter that aims to protect vulnerable browsers which may treat non-HTML files as HTML files. Their filter examines user uploaded files against a set of potentially dangerous HTML elements (a set of regular expressions). The results of their experiment show that the proposed automata-based scheme is highly efficient and more accurate than existing signature-based approach.

In 2011, Anton Barua et al. [14] developing a server side content sniffing attack detection mechanism based on content analysis using HTML and JavaScript parsers and simulation of browser behavior via mock download tests. They have implemented our approach in a tool that can be integrated in web applications written in various languages. In addition, they have developed a benchmark suite for the evaluation purpose that contains both benign and malicious files. They have evaluated our approach on three real world PHP programs suffering from content sniffing vulnerabilities. The evaluation results indicate that their approach can secure programs against content sniffing attacks by successfully preventing the uploading of malicious files.

In 2012, Usman Shaukat Qurashi et al. [15] discusses about AJAX (asynchronous JavaScript and XML) based attack. According to the authors an AJAX enabled web application is composed of multiple interconnected components for handling HTTP requests, HTML code, server side script and client's side script. These components work on different layers. Each component adds new vulnerabilities in the web application. The prolifiration AJAX based web applications increases the number of attacks on the Internet. These attacks include but not limited to CSR forgery attacks, Content-sniffing attacks, XSS attacks, Click jacking attacks, Mal-advertising attacks and Man-in-the-middle attacks against SSL etc. Current security practices and models are focus on securing the HTM. They focus on addressing security issues observed in AJAX and Rich Internet Applications (RIA) and compiling best practices and methods to improve the security of AJAX based web applications.

In 2012, Syed Imran Ahmed Qadri et al. [16] provide a security framework for server and client side. In this they provide some prevention methods which will apply for the server side and alert replication is also on client side. Content sniffing attacks occur if browsers render non-HTML files embedded with malicious HTML contents or JavaScript code as HTML files. This mitigation effects such as the stealing of sensitive information through the execution of malicious JavaScript code. In this framework client access the data which is encrypted from the server side. From the server data is encrypted using private key cryptography and file is send after splitting so that we reduce the execution time. They also add a tag bit concept which is included for the means of checking the alteration; if alteration performed tag bit is changed. Tag bit is generated by a message digest algorithm. We have implemented our

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

approach in a java based environment that can be integrated in web applications written in various languages.

4. PROBLEM DOMAIN

In [17] author presents a concept of enhancing the security in wireless communication. Communication has a major impact on today's business. It is desired to communicate data with high security. These days wireless communication has become an essential form of communication in all aspects of daily life. The main reason for this popularity among other things like the speed of communication and low cost is the convenience of managing and handling data transfer. However this communication is diminished by the insecurity of communication and unidentified intrusion into the network. They deal with a communication protocol that can be used in any wireless network for enhancing the security and preventing any unwanted intruders in penetrating the network. The same thing is needed for the authentication also.

In [18] author discusses how the vernam cipher method can be extended to a new symmetric key cryptographic method called Bit Level generalized modified vernam cipher method with feedback. Authors have used bit level modified vernam cipher method. using random key generator. The authors have introduced a special bit manipulation method so the encryption algorithm will work even for all characters with ASCII Code 0 or all characters with ASCII Code 255. This type of encryption technique is missing when we share the data among the client and server.

In [19] author study and discuss that large pool of data and information is available over the internet for sharing. The social websites, personnel websites, banks database, NGO's data base, telecommunication company's data etc. all are flooded with abundance of information. The rise is not single directional. The exposure to volumes of good and bad people has increased the risk of security of these software and information systems and hence the responsibility of developers has increased multiple times than their early companions. The time we are going through
requires the software security and policies to be considered as early in development process so as to make them part of software architecture. So the security improvement must in the architecture or in the server side.

5. DISCUSSION

The standard encryption algorithm will fail to encrypt a file where all characters are ASCII '0' or all characters with ASCII '255' but the present method will be able to encrypt a file where all characters are ASCII '0' or all characters are ASCII '255'[17]. So this type of approach can be done with the prevention of content sniffing.

From [16] we analyses that there is the possibility to detect an automatic system which automatically enable the bit and the associated clients if the client is the part of the network. Auto uploading feature can be put into consideration so that we save the time[13]. We can consider the file type as Text, HTML web browser and PDF files also.

In 2009 Adam Barth et al. [20] propose and implement a principled content-sniffing algorithm that provides security while maintaining compatibility. Their principles have been adopted, in part, by Internet Explorer 8 and, in full, by Google Chrome and the HTML 5 working group. They also suggest continuing working with browser vendors to converge their content sniffers towards a secure, standardized algorithm. In [15] author also suggest that AJAX developers need to ensure that the code of a web applications has no security hole but also need to implement techniques to sanitize and validate the data exchanged with a server or another client application.

In [21] author suggests that traditional server side approaches that detect injected JavaScript code (e.g., [22, 23, 24]) suffer from a number of limitations. First, detection of injected JavaScript code is performed at browsers where the server side gathers information on legitimate JavaScript code and transfers it to the client side. As a result, browser implementations need to be modified to interpret the information sent by the server side and execute JavaScript code accordingly [25]. So there is need of browser based implementation to interrupt the above.

6. CONCLUSION

Web-based attacks due to program security vulnerabilities are huge concerns for users. While performing seemingly benign functionalities at the browser-level, users might become victims without their knowledge. These might lead to unwanted malicious effects such as the execution of JavaScript code that accesses and transfers credential information to unwanted websites and the filling of forms that result in stealing login credentials. In this paper we survey several aspects of content sniffing and analyses the pros and cons. The future insights in this area are automatic file rendering. Include more file types including PDF, and Word.

REFERENCES

[1] D. Geer, "Security Technologies Go Phishing," Computer Archive, Volume 38, Issue 6, June 2005, pp. 18-21.
[2]Anti-Phishing Working Group, Phishing Activity Trends Report. 2006.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

[3] Keizer, G., Phishing Costs Nearly $1 Billion, TechWeb Technology News.
[4] McMillan, R., Gartner: Consumers to lose $2.8 billion to phishers in 2006, NetworkWorld, 2006.
[5]H. Shahriar and M. Zulkernine, "Mitigating Program Security Vulnerabilities: Challenges and Approaches," ACM Computing Surveys, Vol. 44, Issue 3, September 2012.
[6] H. Shahriar and M. Zulkernine, "Taxonomy and Classification of Automatic Monitoring of Program Security Vulnerability Exploitations," Journal of Systems and Software, Elsevier Science, Vol. 84, Issue 2, February 2011, p. 250-269.
[7] Z. Mao, N. Li, and I. Molloy, "Defeating Cross-Site Request Forgery Attacks with Browser - Enforced Authenticity Protection," Proc. of Financial Cryptography and Data Security, Barbados, Feb 2009, p. 238-255.
[8] Phishing Activity Trends Report, 2010, Accessed from www.antiphishing.org/reports/apwg_report_Q1_2010.pdf.
[9] Ashutosh Kumar Dubey,Animesh Kumar Dubey, Mayank Namdev, Shiv Shakti Shrivastava,"Cloud-User Security Based on RSA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment", CONSEG 2012.
[10] OWASP Top 10 Application Security Risks, Accessed from http://www.owasp.org/index.php/Top_10_2010-Main, November 2011.
[11] Top 25 Dangerous Software Errors, http://cwe.mitre.org/top25/index.html, November 2011.
[12] Hossain Shahriar and Mohammad Zulkernine, "Client-Side Detection of Cross-Site Request Forgery Attacks", 2010 IEEE 21st International Symposium on Software Reliability Engineering.
[13] Misganaw Tadesse Gebre, Kyung-Suk Lhee and ManPyo Hong, "A Robust Defense Against Content-Sniffing XSS Attacks",IEEE 2010.
[14] Anton Barua, Hossain Shahriar, and Mohammad Zulkernine , "Server Side Detection of Content Sniffing Attacks", 2011 22nd IEEE International Symposium on Software Reliability Engineering.
[15] Usman Shaukat Qurashi , Zahid Anwar, "AJAX Based Attacks:Exploiting Web 2.0",IEEE 2012.
[16] Syed Imran Ahmed Qadri, Prof. Kiran Pandey, "Tag Based Client Side Detection of Content Sniffing Attacks with File Encryption and File Splitter Technique", International Journal of Advanced Computer Research (IJACR), Volume-2, Number-3, Issue-5, September-2012.
[17] Ranbir Sinha, Nishant Behar, Devendra Singh, "Secure Handshake in Wi-Fi Connection (A Secure and Enhanced Communication Protocol)", International Journal of Advanced Computer Research (ISSN (IJACR) ,Volume 2, Number 1,March 2012.
[18] Prabal Banerjee, Asoke Nath, "Bit Level Generalized Modified Vernam Cipher Method with Feedback", International Journal of Advanced Computer Research (IJACR), Volume-2, Number-4, Issue-6, December-2012.
[19] Rakesh Kumar, Dr.Hardeep Singh, "Analysis of Information Systems Security Issues and Security Techniques", International Journal of Advanced Computer Research (IJACR), Volume-2, Number-4, Issue-6, December-2012.
[20] Adam Barth,Juan Caballero and Dawn Song, "Secure Content Sniffing for Web Browsers, or How to Stop Papers from Reviewing Themselves",IEEE 2009.
[21] Hossain Shahriar and Mohammad Zulkernine, "Injecting Comments to Detect JavaScript Code Injection Attacks", 2011 35th IEEE Annual Computer Software and Applications Conference Workshops.
[22] P. Wurzinger, C. Platzer, C. Ludl, E. Krida, and C. Kruegel, "SWAP: Mitigating XSS Attacks using a Reverse Proxy,"Proc. of the SESS, Vancouver, May 2009, pp. 33-39.
[23] A. Futoransky, E. Gutesman, and A. Waissbein, "A Dynamic Technique for Enhancing the Security and Privacy of Web Applications," Proc. of Black Hat USA, Las Vegas, 2007.
[24] P. Bisht and V. Venkatakrishnan, "XSS-GUARD: Precise Dynamic Prevention of Cross-Site Scripting Attacks," Proc. of the 5th DIMVA, Paris, July 2008, pp.23-43.
[25] M. Gundy and H. Chen, "Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart Cross-site Scripting Attacks," Proc. of NDSS, San Diego, Feb 2009.

BIOGRAPHY

Animesh Dubey was born in Madhya Pradesh on 01 January 1987. He received the B.E. degree in Computer Science from Shree Institute of Science Technology, Bhopal, India in 2009 and pursuing M.Tech degree from SSSIST, Sehore, Bhopal, India in Computer Science Engineering.