International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-2 Issue-10 June-2013

Content Sniffing Attack Detection in Client and Server Side: A Survey

Bhupendra Singh Thakur[1], Sapna Chaudhary[2]

Abstract

In today's environment we cannot think about internet. It has the interface of client and server. After analysing several research studies, we conclude that the communication between client and server may suffer from several security concerns like Denial of Service (DoS) attack, Content Sniffing Attack and Replay attack. In this paper we mainly concentrate on content sniffing attack. We survey several traditional techniques on content sniffing attack and major the advantage and disadvantages. We also focus on finding the better security provision which can be applied during data communication through client and server. Our main aim of this paper is to find the outcomes which can better detect the content sniffing attack in client and server side.

Keywords

Content Sniffing Attack, MIME, DoS, Security Provision.

1. Introduction

Content sniffing and Cross-site scripting (XSS) vulnerabilities are the major security threats today when we are in the server-client environment or using any web browser. There are several other attacks which are discuss[1] and [2].XSS vulnerabilities allow an attacker to inject malicious content into web pages from trusted web servers. The malicious code with the source script run on the same ground as other web pages and shows the fake presence of trusted authority. Since the malicious content runs with the same privilege as the trusted content from the web servers, the malicious content can steal the victim users' private data or take unauthorized actions on the users' behalf. Content sniffing attack is an attempt to deduce the content of a file format of the data or alteration in the byte stream. It is also called media type sniffing or MIME sniffing. Traditional approaches include static analysis [3], and dynamic monitoring [4, 5], and browser-based defenses [6, 7] are also considered.

Bhupendra Singh Thakur, M.Tech Scholar, Computer Science, Shri Ram Group of Institutes, Jabalpur.

Sapna Chaudhary, Assistant Professor, Computer Science, Shri Ram Group of Institutes, Jabalpur.

If we think of the aggregation of static and runtime approaches provide more accuracy in detecting [8] at the cost of the deployment of customized frameworks. Browser-based approaches require end user interventions and rewriting of entire implementations [8].

So in the above direction, we survey several web based attacks which can possible through communication, we also discuss about the security concern which can be applied in future for better security in web communication.

2. Web based Attacks

There are several web based attacks, so we only concentrate on those which are related to my paper. In the above direction we discuss the following web based attack:

Phishing: In this type of attack the victim is led to believe that he or she is on a website which is true or real, when in fact it is just a copy of the real one but not true. It means it is the fake presence of showing the look as same as the trusted web domain. These types of attacks mainly target the official email and high profile identity.

Web browser exploits: In this type of exploits the web attacker design such website, which is helpful in the attack. This technique allows them to gain access without the victim's knowledge.

Third party add-ons: The majority of websites require the use of third party add-ons such as flash player, reader, songs and video plugin and Acrobat Reader. Both of these widely used products have become a favorite target for web attacker. As more administrators and

Download Executable: The attacker use to play on people fears that their machine has been infected with malware; users are encouraged to download antivirus software. This is nothing but malware that infects the machine and demands payment if the user wants to uninstall the software.

Hybrid attack: While the web offers much greater scope for attackers, email still remains a powerful tool. Combined with the web, the threats not only multiply but the risk that the user becomes a willing prey is very high. One common trick is to use current news events to spread malware spam. Emails purporting to offer exclusive news, videos or files are popular online traps to open dangerous attachments or be redirected to infected or fake websites.

Content Sniffing: According to [9][10] 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 an attacker 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.

3. Related Work

In 2010, Zubair M. Fadlullah et al. [11] to combat against attacks on encrypted protocols; they propose an anomaly-based detection system by using strategically distributed monitoring stubs (MSs). They have categorized various attacks against cryptographic protocols. The MSs, by sniffing the encrypted traffic, extract features for detecting these attacks and construct normal usage behavior profiles. Upon detecting suspicious activities due to the deviations from these normal profiles, the MSs notify the victim servers, which may then take necessary actions. In addition to detecting attacks, the MSs can also trace back the originating network of the attack. They call their unique approach DTRAB since it focuses on both Detection and TRAceBack in the MS level. The effectiveness of their proposed detection and traceback methods are verified through extensive simulations and Internet datasets.

In 2011, Misganaw Tadesse Gebre et al. [12] 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. [13] 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, Syed Imran Ahmed Qadri et al. [9] 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 approach in a java based environment that can be integrated in web applications written in various languages.

In 2012, Namrata Shukla et al. [14] present an efficient approach for fraud detection. In our approach they first maintain a log file for data which contain the content separated by space, position and also the frequency. Then they encrypt the data by substitution method and send to the receiver end. They also send the log file to the receiver end before proceed to the encryption which is also in the form of secret message. So the receiver can match the data according to the content, position and frequency, if there is any mismatch occurs, they can detect the fraud and does not accept the file.

In 2013, Animesh Dubey et al. [10] propose an efficient partition technique for web based files (jsp, html, php), text (word, text files) and PDF files. They are working in the direction of attack time detection. For this motivation they are considering mainly two

factors first in the direction of minimizing the time, second in the direction of file support. For minimizing the time we use partitioning method. They also apply partitioning method on PDF files. There result comparison with the traditional technique shows the effectiveness of their approach.

4. Problem Domain

After discussing several research works we can come with some problem area in the traditional approaches which are following:

1) Missing of automated identification of file upload procedures in web applications and auto response.
2) Reduction of time overhead for analyzing images.
3) There is no work related to flash like images.
4) Encryption techniques can be improved with some message digest algorithm as suggest in [15]. This can improve the data security and prevent it from brute force attack. Because brute force attack is difficult when the keys are large.
5) In [10][16] author suggests that traditional server side approaches that detect injected JavaScript code (e.g., [17, 18, 19]) 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 [20]. So there is need of browser based implementation to interrupt the above.
6) Zip files can consider for reduction of time by splitting technique.
7) PS files can consider for reduction of time by splitting technique.

5. Analysis

After analysis of several research paper. We come with some result analysis by the authors working in the same field. In [13] authors working in the content sniffing area taking different file format, but suggesting some time reduction technique for reduce the attack detection time as the future suggestion. The result by [13] is shown in table 1. They also suggest that in future we can work on flash files.

Table 1: Result [13]

After Attack		
Fname	**Size(KB)**	**Time Difference (MS)**
DOCX1	78.74	167
GIF1	7.65	98
PDF1	212.92	251
TEXT1	0.02	42
ZIP1	222.99	237
PS1	155.74	214

In [9] authors working in the content sniffing area taking different file format and using splitting technique for reducing the time. The result by [9] is shown in table 1. But fail to cover several file format like .pdf,.ps,.zip,.gif and also fail in the automatic supervision. They also suggest that in future we work in those files along with flash type file format.

Table 2: Result [9]

After attack				
fname	**Size (KB)**	**Attack time**	**Server time**	**Time Difference (MS)**
file1.html	2822	10:40:1:332	10:40:1:480	148
file2.html	4104	10:44:22:461	10:44:22:650	189
file3.html	10945	10:46:28:431	10:46:28:610	179
file4.html	12826	11:1:23:570	11:1:23:713	143
file5.html	14023	11:2:45:231	11:2:45:370	139

In [10] authors working in the content sniffing area taking different file format and using splitting technique for reducing the time. They cover .pdf files also. The result by [10] is shown in table 1. But fail to cover several file format like .ps,.zip,.gif and also fail in the automatic supervision.

Table 3: Result [10]

After attack				
Fname	**Size (KB)**	**Attack time**	**Server time**	**Time Difference (MS)**
ab.html	78827	3:4:6:426	3:4:6:567	141
54.pdf	190143	3:7:48:142	3:7:48:289	147
Ab1.txt	13111	8:8:43:140	8:8:43:202	62

6. Conclusion and Future Direction

Web-based attacks due to program security vulnerabilities are huge concerns for users. In this paper we survey several attacks including content

sniffing attacks and discuss their advantages and disadvantages. We also discuss their research work and come with some future suggestions.

The future insights in this area are automatic file rendering. There are several file format which are not covered like .ps,.zip,.gif. There is also a scope in the direction of flash type files.

References

[1] Gaurav S. Kc, Angelos D. Keromytis, and Vassilis Prevelakis. Countering code-injection attacks with instruction-set randomization. In CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, pages 272–280, New York, NY, USA, 2003.

[2] E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic. Noxes: A Client-Side Solution for Mitigating Cross Site Scripting Attacks. In Proceedings of the ACM Symposium on Applied Computing (SAC), Dijon, France, April 2006.

[3] G. Wassermann and Z. Su, "Static Detection of Crosssite Scripting Vulnerabilities", Proceedings of the 30th ICSE,Leipzig, Germany, May 2008, pp. 171-180.

[4] Tramontana, "Identifying Cross Site Scripting Vulnerabilities in Web Applications", Proceedings of the Sixth International Workshop on Web Site Evolution (WSE 2004), Chicago, September 2004, pp. 71-80.

[5] D. Balzarotti, M. Cova, V. Felmetsger, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna, "Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications", Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, May 2008, pp. 387-401.

[6] E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic, "Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks", Proceedings of 21st ACM Symposium on Applied Computing, Dijon, France, April 2006, pp. 330-337.

[7] E. Ofuonye and J. Miller, "Resolving JavaScript Vulnerabilities in the Browser Runtime", Proceedings of the 19th International Symposium on Software Reliability Engineering, Washington DC, November 2008, pp. 57-66.

[8] Hossain Shahriar and Mohammad Zulkernine," MUTEC: Mutation-based Testing of Cross Site Scripting", IEEE 2009.

[9] Syed Imran Ahmed Qadri, 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.

[10] Animesh Dubey, Ravindra Gupta, Gajendra Singh Chandel," An Efficient Partition Technique to reduce the Attack Detection Time with Web based Text and PDF files", International Journal of Advanced Computer Research (IJACR),Volume-3, Number-1, Issue-9, March-2013.

[11] Zubair M. Fadlullah, Tarik Taleb,Athanasios V. Vasilakos, Mohsen Guizani and Nei Kato, "DTRAB: Combating Against Attacks on Encrypted Protocols Through Traffic-Feature Analysis", IEEE/ACM Transactions On Networking, Vol. 18, No. 4, August 2010.

[12] Misganaw Tadesse Gebre, Kyung-Suk Lhee and ManPyo Hong, "A Robust Defense against Content Sniffing XSS Attacks", IEEE 2010.

[13] Anton Barua, Hossain Shahriar, and Mohammad Zulkernine, "Server Side Detection of Content Sniffing Attacks", 2011 22nd IEEE International Symposium on Software Reliability Engineering.

[14] Namrata Shukla,Shweta Pandey," Document Fraud Detection with the help of Data Mining and Secure Substitution Method with Frequency Analysis", International Journal of Advanced Computer Research (IJACR),Volume 2, Number 2, June 2012.

[15] 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.

[16] Hossain Shahriar and Mohammad Zulkernine, "Injecting Comments to Detect JavaScript Code Injection Attacks", 2011 35th IEEE Annual Computer Software and Applications Conference Workshops.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

I completed BE Degree from Guru Ramdas Khalsa Institute of Science & Technology Jabalpur. Currently pursuing M.Tech in CSE Branch from Shri Ram Group of Institutes Jabalpur, M.P, India.