ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

A Novel Approach of MIME Sniffing using AES

Ankita Singh, Amit Saxena, Dr.Manish Manoria
TRUBA Institute of Engineering and Information Technology (TIEIT), Bhopal (M.P)

Abstract*— *In today's scenario communication is rely on web, users can access these information from web with the use of browsers, as the usage of web increases the security of data is required. If browser renders malicious html contents or JavaScript code block, the content sniffing attack may occur. In this paper we provide a framework with AES algorithm to secure the content sniffing for the web browsers with text, image and PDF files. In this work the data files having encryption then partition in multiple parts for reducing the duration of file transmission and transferring with parity bit checking to identify the attack.

***Index Terms*— Cross-Site Scripting, Web Application Security, Content Sniffing, MIME, AES.**

I. INTRODUCTION

Internet growth rapidly increases; it has consolidated itself as a very powerful platform that has changed the way for business, and the way for communication with the help of web applications. But many web applications are vulnerable to different types of attack like cross site scripting, which can be exploited by injecting the JavaScript code. It is very difficult to identify the attack in the client as well as server side. To identify the multiple vulnerabilities, code injection attacks and the prevention methods for the client and server side is the proposed work area, which helps to users to work safe over internet.

Cross-Site Scripting (XSS) is a new common vulnerability which can let hackers inject the code into the output application of web page which will be sent to a visitor's web browser and then, the code which was injected will execute automatically or steal the sensitive information from the visits input. XSS is a widespread security issue in many modern Web applications. One way to detect these vulnerabilities is to use fully automated tools such as Web Vulnerability Scanners. But the detection rate of certain types of XSS vulnerabilities is rather disappointing. With XSS, a section of attack is Content Sniffing Attack.

In current web environment every browser has a content sniffing algorithm that monitors the contents of HTTP replications and infrequently overrides the MIME type. Content sniffing attack, a strew ineradicable encircling hellish payload is erroneously handled by a victim's browser, usher in the period of HTML content and conduct of JavaScript pandect. The maleficent payload transmission analysis is uploaded by assailers to legitimate websites [4,5]. These line show affectionate in a second their sense types or Multipurpose Internet Mail Extension (MIME) information are considered [3]

We discuss some web application attacks which can be possible over browser also discuss security concern can be applied in future for security on web application environment.

The contents are divided in different sections. In section 2 we mention different types of attacks. Related work is discussed in section 3. Proposed work is discussed in section 4. Result analysis in section 5. Conclusion and future direction in Section 6, and then references are mention.

II. ATTACKS

We discuss about some attacks, associated with this work.

ClickJacking[11] - The purpose of this attack is to open the target website in an invisible frame and get the user to click somewhere in the frame when they don't even know they're clicking in that website," says Ari Elias-Bachrach, application security consultant and trainer for security consultancy Defensium. "In this way, you can trick the user into making a mouse click that does something [malicious] on the website.

]Phishing [11] - Phishing attacks are designed to trick users into thinking they are a link from an organization or person they know, making people feel safe enough to click or divulge information they otherwise wouldn't. Many corporate security training programs have helped users spot the most obvious first-generation phishing attempts, which were designed to steal credentials such as banking passwords. But attackers are getting more crafty.

Web browser exploits [12] - In this manufacturer of exploits the scold assailant stumbling-block such websites, which is helpful in the attack. This close allows them to effect access without victim's knowledge.

Malvertising [11] - Online advertising that contains embedded malware or links to malicious websites, otherwise known as malvertising, is among the most common high-volume, Web-based attacks assaulting online machines today, according to researchers with OpenDNS.

Third party add-ons [12] - The duration of websites petition the consideration of third gather add-ons such as shred entrant, school books, songs and video plugin and Acrobat Reader. Both of these publicly old retail undertaking mature a favourite target for web attacker

Content Sniffing [10] - It also known as Mime Sniffing is the practice of inspecting the content of a byte stream to attempt to deduce the file format of the data within it. Many HTTP servers supply a Content-Type that does not match the actual contents of the response. If an attacker

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

manipulates the content in a way to be accepted by the web app and rendered as HTML by the browser, it is possible to inject malicious code.

- Browser second-guesses Content-Type header
- Looks at response content, URI and also tag that initiated the request
- An attacker can trick older browsers into guessing the wrong Content-Type

III. RELATED WORK

All In 2010, Hossain Shahriar and Mohammad Zulkernine et al. [6] discuss about Cross Site Request Forgery (CSRF) which sanctions an assailant to perform unauthorized activities without the cognizance of a utilizer. An assailment request capitalizes on the fact that a browser appends valid session information for every request. In output, a browser is the only probe location for attack symptoms and take congruous actions. According to the author Current browser-predicated detection methods are predicated on cross-inchoation policies that sanction white listed third party websites to perform requests to a trusted website. To alleviate these constraints, 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 exhibited on a webpage (overtness). To surmount an attacker's endeavor to circumvent form overtness checking, they withal compare the replication content type of a suspected request with the expected content type.

In 2010, Zubair M. Fadlullah et al. [9] proposed 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 our unique approach DTRAB since it focuses on both Detection and TRAceBack in the MS level. The effectiveness of the proposed detection and traceback methods are verified through extensive simulations and Internet datasets.

In 2013, Animesh Dubey, Ravindra Gupta, Gajendra Singh Chandel [8] proposed an efficient partition technique for web based files. They were working for the detection of attack time. For this they take opted two approaches, first is in the track of minimizing the time and second in the way of web based files support. To reduce the time they uses the partition method. The comparative study with the traditional approach show the effectiveness of their approach.

In 2013, Saket Gupta[15], proposed partition technique with DES algorithm, by which author can provide the security and notify to the client for vulnerability. Also implemented Data Encryption Standard (DES) algorithm as an encryption technique and split files using the partition algorithm as file splitter technique in order to reduce the time overhead.

IV. PROBLEM DOMAIN

After reviewing various research works we identify some problem area, which are as follows:

1) Lack of identification for different types of files and response messaging to server. [13].

2) There is no work related to any compressed file type [13].

3) PDF and some compressed files reflected using file splitter algorithm.

4) Encryption techniques could be more reliable with speed.

V. PROPOSED WORK

The research motivation of our work is to secure the communication between client and server with attached documents. This research paper describes the content sniffing attack, if browser executes files other than HTML having malicious code. However, existing server side content sniffing attack detection approaches suffer from a number of limitations. In [14], the researchers implemented a tool that integrated in web applications. They have evaluated their approach with program tormented by content sniffing vulnerabilities. Their future work includes distinguishing ways in which to scale back the overhead for analysing giant files and automatic rendering.

Our proposed system supports, Text, HTML, Word document, PDF and Images. Our system is web based application have two group of users, admin and clients. If clients want to make connection with server to take documents, client must have to register, after accept the request by admin, client can access the documents. After authorization, data preparation process will start. As DES with file partition technique [15] uses 64-bit block cipher is lesser then the AES encryption technique, which encrypts 128-bit block with 128, 192 or 256 bit in key length.

AES Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical. Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption. The overall structure of AES[16] is shown in figure 1

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

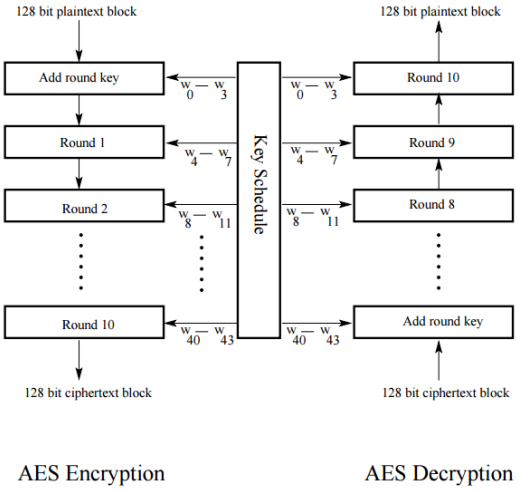


Fig 1 : AES Encryption/Decryption Process

The number of rounds is 10 as shown in figure 1, when the encryption key is 128 bit long. Before any round-based processing for encryption can begin, the input state array is XOR with the first four words of the key schedule. The same thing happens during decryption — except that now we XOR the cipher text state array with the last four words of the key schedule.

For encryption, each round consists of the following four steps: 1) Substitute bytes, 2) Shift rows, 3) Mix columns, and 4) Add round key. The last step consists of XOR the output of the previous three steps with four words from the key schedule.

For decryption, each round consists of the following four steps: 1) Inverse shift rows, 2) Inverse substitute bytes, 3) Add round key, and 4) Inverse mix columns.

After securing document with AES, document splitting process occurs and if any unauthorised client access these patterns, the hidden parity bit turns 1 otherwise 0 and then send document to client and store in the log data repository. The overall process is depicted in figure 3. For response automation algorithm is shown below:

Algorithm 1: For Response Automation Mapping

1) Inputs: The set of Document Request (DR1, DR2…….DRn) from the full set of request by the client user.
2) Output: Map Document Request (DR1, DR2 ……..DRn).
3) do
   Check the Log Request set.
   Design a log of Document Request (LDr1,LDr2……LDrn) to search the peak request.
   For each log of Document request (LDR= LDr1,LDr2……LDrn) do
     goto Algorithm 2;
     goto splitting;
   End;
4) Information will be send with 0/1 append with reconfiguring document request loads.
5) Details are added including the attack time in the log file mentioning the client name
6) Finish.

Algorithm 2: AES algorithm for encryption and decryption [16]

Step 1. This step is SubBytes consists of using a 16 × 16 lookup table to find a replacement byte for a given byte in the input state array. The entries in the lookup table are created by using the notions of multiplicative inverses in GF (28) and bit scrambling to destroy the bit-level correlations inside each byte.

Step 2. This step is Shift Rows for shifting the rows of the state array during the forward process. The corresponding process during decryption is denoted InvShiftRows for Inverse ShiftRow Transformation. The goal of this transformation is to scramble the byte order inside each 128-bit block.

Step 3. This step is called Mix Columns for mixing up of the bytes in each column separately during the forward process. The corresponding transformation during decryption is denoted InvMixColumns and stands for inverse mix column transformation. The goal is to further scramble up the 128-bit input block.

Step 4. This step is called AddRoundKey for adding the round key to the output of the previous step during the forward process. The corresponding step during decryption is denoted InvAddRoundKey for inverse add round key transformation.

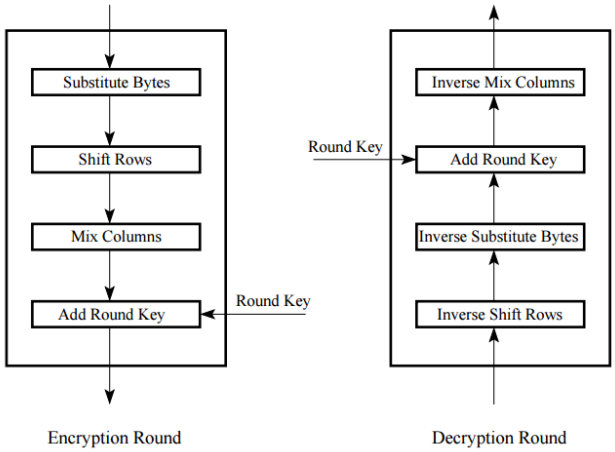
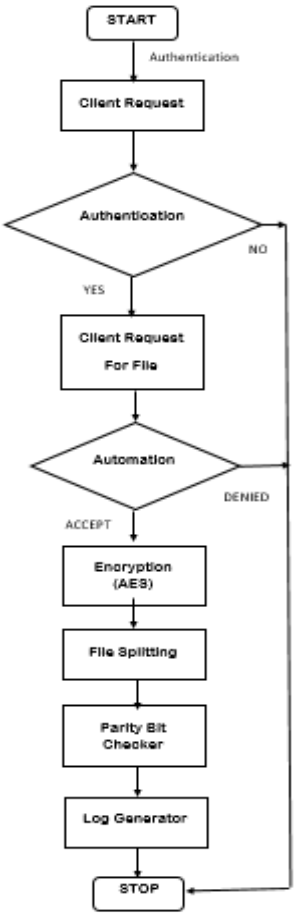


Fig 2 : One round of Encryption and decryption of AES.

Algorithm 3: Splitting Algorithm [8]

Step 1: Initialization
int counter=0, length=0
Step 2: File f=new File(f1);
Step 3: long size=f.length()/1024;
Step 4: if(size<=100) len=(int)f.length()/2;
Step 5: else if(size<=250) len=(int)f.length()/3;
Step 6: else if(size<=500) len=(int)f.length()/4;
Step 7: else len=(int)f.length()/6;

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015



VI. RESULT ANALYSIS

A comparative analysis in figure 4 shows the effectiveness of our approach which is compared by [17]. It shows the attack detection time is best compared to the standard technique. The analysis is finished on the idea on automation method and for manual method.

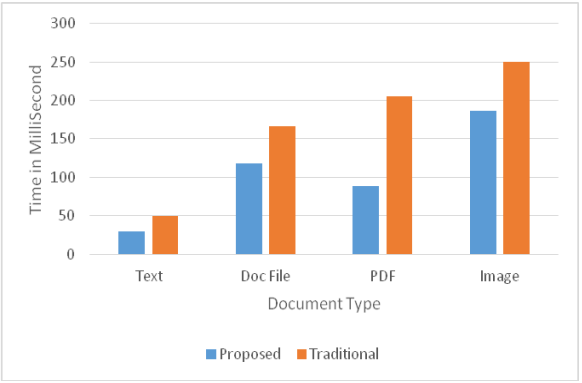


Fig 4: Comparative Analysis

VII. CONCLUSION & FUTURE SCOPE

Web-based attacks attributable to program security vulnerabilities are immense issues for users. Whereas playacting apparently benign functionalities at the browser-level, users would possibly become victims while not their data. These would possibly cause unwanted malicious effects like the execution of JavaScript code that accesses and transfers credentials info to unwanted websites and therefore the filling of forms that end in stealing login credentials. During this paper, we have a tendency to address the mitigation of a number of these exploitations by developing automatic attack detection approaches at each server and client-sides.

In future work on MIME Sniffing, we will implement automated identification of file upload procedure as well as automated type conversion to MIME type and incorporate multiple file type detection.

REFERENCES

[1] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B.: A survey of information- centric networking. IEEE Communications Magazine 50(7), 26–36 (2012).

[2] 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, Volume-2 Number-3 Issue-5 September-2012.

[3] Multipurpose Internet Mail Extensions (MIME), http://www.ietf.org/rfc/rfc2046.txt?number=2046.

[4] Barth, A., Caballero, J., Song, D.: Secure Content Sniffing for Web Browsers, or How to Stop Papers from Reviewing Themselves. In: Proceedings of IEEE Security & Privacy, pp. 360–371 (2009).

[5] Gebre, M.T., Lhee, K.-S., Hong, M.: A Robust Defense against Content sniffing XSS Attacks. In: Proceedings of 6th International Conference on Digital Content, Multimedia Technology and its Applications, pp. 315–320 (2010).

[6] Hossain Shahriar and Mohammad Zulkernine, "Client-Side Detection of Cross-Site Request Forgery Attacks", 2010 IEEE 21st International Symposium on Software Reliability Engineering.

[7] 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.

[8] 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.

[9] 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.

[10] http://dunnesec.com/2014/05/26/content-sniffing/.

[11] http://www.darkreading.com/risk/10-web-based-attacks-targeting-your-end-users/d/d-id/1140224?.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

[12] Bhupendra Singh Thakur, Sapna Chaudhary, "Content Sniffing Attack Detection in Client and Server Side: A Survey", International Journal of Advanced Computer Research (IJACR), Volume- 3 Number-2 Issue-10 June-2013.

[13] Brad Wardman, Tommy Stallings, Gary Warner, Anthony Skjellum," High-Performance Content-Based Phishing Attack Detection", " eCrime Researchers Summit (eCrime), 2011 , vol., no., pp.1,9, 7-9 Nov. 2011.

[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] Saket Gupta, Secure and Automated Communication in Client and Server Environment, International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-3 Number-4 Issue-13 December-2013.

[16] Avi Kak, AES: The Advanced Encryption Standard, Avinash Kak, Purdue University, https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf.

[17] Barua, A., Shahriar, H., Zulkernine, M.: Server Side Detection of Content Sniffing Attacks. In: 2011 22nd IEEE International Symposium on Software Reliability Engineering (2011).

[18] Multipurpose Internet Mail Extensions (MIME),http://www.ietf.org/rfc/rfc2046.txt?number=2046 Paros - Web application security assessment, http://www.parosproxy.org/index.shtml (accessed).

[19] Open Source Vulnerability Database, http://osvdb.org.

[20] Dubey, A.K., Dubey, A.K., Namdev, M., Shrivastava, S.S.: Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment. In:2012 CSI Sixth International Conference on Software Engineering (CONSEG) (2012).

[21] Dubey, A.K., Dubey, A.K., Agarwal, V., Khandagre, Y.: Knowledge discovery with a subset- superset approach for Mining Heterogeneous Data with dynamic support. In: 2012 CSI Sixth International Conference on Software Engineering (CONSEG) (2012).

AUTHOR BIOGRAPHY

Author 1-4th sem Mtech scholar in CSE from TRUBA Institute of Engineering and Information Technology (TIEIT), Bhopal (M.P)

Author 2-Guide and HOD, CSE in TRUBA Institute of Engineering and Information Technology (TIEIT), Bhopal (M.P)

Author 3-Director of TRUBA Institute of Engineering and Information Technology (TIEIT), Bhopal (M.P)