International Journal of Innovation Sciences and Research
Vol.4, No, 3, pp. 090-095 March- 2015

Available online at
http://www.ijisr.com

RESEARCH ARTICLE

WEB DATA SECURITY IN E-COMMERCE AGAINST VARIOUS VULNERABILITIES USING AES ALGORITHM

***Rupica Puri and Sheetal Kalra**

GNDU, RC Jalndhar, Punjab, India

Accepted 09th February, 2015; Published Online 31st March, 2015

ABSTRACT

E-commerce Security is a part of the Information Security framework and is specifically applied to the components that affect E-commerce that include Computer Security, Data security and other wider areas of the Information Security framework. E-commerce security has its own particular touch and is one of the highest visible security components that affect the end user through their daily payment interaction with business. E-commerce security is the protection of E-commerce assets from unauthorized access, use, alteration, or destruction. Dimensions of E-commerce security-Integrity, Non-repudiation, Authenticity, Confidentiality, Privacy, Availability has been discussed. E-Commerce offers the banking industry great opportunity, but also creates a set of new risks and vulnerability such as security threats. Information security, therefore, is an essential management and technical requirement for any efficient and effective Payment transaction activities over the internet. Still, its definition is a complex Endeavour due to the constant technological and business change and requires a coordinated match of algorithm and technical solutions. In this paper, the proposed methodology for E-commerce security has been discussed, various reasons of vulnerabilities arising in the Security of E-commerce, Different security issues in E-commerce and proposed methodology using AES algorithm for prevention against SQL injections, Price manipulation, weak authentication and authorization, cross site scripting.

Key Words: E-commerce, security issues, threats, cross site scripting, SQL injection.

INTRODUCTION

Internet technology and the variety of resulting applications have revolutionized the way customers do business and interact with sellers of commercials products and services. In retail industries, websites for business-to-customer electronic commerce provide more accessible, easier, faster, and cheaper methods for individual consumers to conduct their retail transactions. As a result, online shopping has continued to gain popularity as a transaction medium. E-commerce security is a part of the information security framework and is specifically applied to the components that affect E-commerce that include computer security, data security and other wider sphere of the information security framework. E-commerce plays essential role in today's business, and that will continue to grow in future. With the development of information technology and communication technology and the popularization of the Internet, E-Commerce is sweeping through all walks in world with an irreversible trend. E-commerce security has its own particular variations and is one of the highest visible security components that the end user through their daily payment interactions with business. Privacy and security are major concern for electronics technologies. With the development of information technology and communication technology and the popularization of the internet, E-commerce holds many advantages for the commercial world, for example efficiency and convenient and also unfortunately have some disadvantages as security issues are emerging and have become the bottleneck of E-commerce development. E-commerce will not be successful without protecting the consumers' right especially in the area of information security. The downside to this is that while online, all Internet-based electronic commerce is vulnerable to misuse either by unauthorized users penetrating the system or by authorized users abusing their privileges.

Privacy concern reveals a lack of trust in the variety of contexts, including commerce, electronic health records, electronic recruitment technology and social networking and this directly influence users. In the development of the E-commerce security has always been the core and the main issue (Sangjae Lee and Hyunchul Ahn, 2011). Security is one of the principal and continuing concerns that restrict customers and organizations engaging with E-commerce. Web E-commerce applications that handle payments have more compliance issues, are at increased risk from targeted than other websites and there are greater consequences if there is data loss or alterations (Niranjanamurthy and DR. Dharmendra Chahar, 2013). Online shopping through shopping websites having certain steps to buy a product which are safe and secure .The E-commerce industry is slowly addressing security issues on their internal networks. Trojan horse programs launched against client systems pose the greatest threat to E-commerce because they can bypass most of the authentication and

****Corresponding author: Rupica Puri***
GNDU, RC Jalndhar, Punjab, India

authorization mechanisms used in an E-commerce transaction. The literature suggests that information systems practitioners and researches generally agree that security is a multidimensional construct that is derived from several dimensions (E.g. confidentiality, integrity, availability, non-repudiations).

Various security Concerns/Dimensions/Features

Security is the challenge and the problem for successful E-commerce implementation (Mohanad Halaweh and Christine Fidler, 2008). E-commerce security is the protection of E-commerce assets from unauthorized access, use, alteration and destruction. While security features do not guarantee a secure system, they are necessary to build a secure system. Due to the practicality of E-commerce and open of the internet, security issues are emerging and have become the bottleneck of the E-commerce development (Sengupta *et al*., 2005). Security features categories:

Confidentiality

Confidentiality refers to the degree to which improper disclosures of information are anticipated and prevented. Systems with superior confidentiality are better able to anticipate and prevent improper disclosure of information, such as leakage of information to an unauthorized party. A system's inability to anticipate and prevent improper disclosure of information may well indicate system insecurity. Common security measures to maintain confidentiality include encryption and authentication such as password-based and token-based authentication.

Integrity

Integrity refers to the degree to which improper modifications to information are anticipated and prevented. Systems with superior integrity are better able to anticipate and prevent improper modification of information, such as faulty alteration, deletion, or addition. While some erroneous modifications of information are accidental, others may be made intentionally by unauthorized parties. Common security measures to maintain integrity include digital signatures and anti-virus programs that prevent a virus from destroying data.

Availability

Availability refers to the degree to which information is available to authorized subjects when required. Systems with superior availability are better able to consistently provide relevant information to authorized parties. Common security measures to maintain availability include back-up systems and countermeasures for distributed-denial of- service attacks.

Privacy

Privacy is the ability to ensure that information is accessed and changed only by the authorized parties. This can be achieved by Encryption. Sensitive data such as credit cards details, sales figures etc are common security measure used to ensure privacy.

Authorization

Authorization allows a person or computer system to determine if one has the authority to request or approve an action or information that is allows you to manipulate your resources in specific ways. Authorization is tied with Authentication. If a system can securely verify that a request for information or a service has come from known individual, the system can then check against its internal rules to see if that person has sufficient authority for the request to proceed.

Non-repudiation

Non-repudiation in a buyer–seller exchange refers to the ability to which the systems is capable of ensuring that information sent by the customer is received by the person the seller claims to be. The goal is to ensure that the seller cannot later deny a completed transaction. Systems with superior non-repudiation are better able to provide verifiable proof of identity. Digital signature is a common security measure used to ensure non-repudiation.

Related Work

Debabrata Kar and Panigrahi (2013) proposed a lightweight approach to prevent SQL Injection attacks by a novel query transformation scheme and hashing in which SQL queries are converted into their structural form and then applying MD5 hashing to generate unique hash keys for each legal query collected during normal use. This approach minimizes the size of the legitimate query repository and facilitates fast and efficient searching at run-time using a primary index. This approach does not require major changes to application code and has negligible effect on performance even at higher load conditions due to low processing overhead.

Experimental results show that this approach can effectively prevent all types of SQL injection attempts except second order SQL injection which can be researched in future. Srinivas Avireddy (Srinivas Avireddy *et al*., 2012) proposed a solution to the problem of SQL injection by preventing it using an encryption algorithm based on randomization. It has better performance and provides increased security in comparison to the existing solutions. Also the time to crack the database takes more time when techniques such as dictionary and brute force attack are deployed.

Liban and Hilles (2014) enhanced SQL-injection vulnerability scanning tool for automatic creation of SQL-injection attacks (MYSQL Injector) using time-based attack with Inference Binary Search Algorithm. It considers four types of blind SQL injection attacks, true/false, true error, time-based and order by attacks.

This tool will mechanize the process of the blind SQL injection attacks to check the blind SQL injection vulnerability in the PHP-based websites that use MySQL databases. They tested 44 susceptible websites and 30 non susceptible websites to make sure the accuracy of the tool. The result shows 93% accuracy for detecting the vulnerability while MySQL injector performs 84%. Yan and Chiu (2007) introduced Notified Credit Card Payment System (NCCPS) that provide a unified Web service

for Merchants to request for credit card payment, instead of using different protocols developed by different banks. They show how to use the alert mechanism and Web service technologies to integrate a security credit card online payment system. The NCCPS further integrates with the customer service call center with the same platform. They explain the significance of the alert mechanism and how various technologies help. Monika and Raman (2014) presents the initial client-side resolution called Noxes, to moderate XSS attacks. It works as a web proxy and utilizes both automatically and manually generated rules to block cross-site scripting attacks. it is the first client-side solution that provides XSS protection without relying on the web application providers. It supports mitigation mode of an XSS that appreciably decreases the count of relationship attentive prompts.

Vibhakti Mate, Milind Tote, Abdulla Shaik (2014) proposes client-side solution to the cross site scripting. We dynamically track the flow of sensitive values (e.g., user cookies) on the client side by modifying the web browser. Whenever such a sensitive value is about to be transferred to a third party (i.e., the adversary), the user is given the possibility to stop the connection. With this combination of dynamic and static techniques, protection is provided against XSS attacks in a reliable and efficient way. They tested the enhanced browser on more than one million web pages by means of a crawler that is capable of interpreting JavaScript code.

The results demonstrate that only a small number of false positives is generated. Jyoti Chhikara Ritu Dahiya Neha Garg Monika Rani (2013) paper gives brief information about phishing, its attacks, steps that users can take to safeguard their confidential information. This paper also shows a survey conducted by Netcraft on phishing after responses from over 630,790,500 web sites and concluded that Taobao draws the second highest number if phishing attacks next to Facebook.

E-commerce and various vulnerabilities

There are number of reasons why security vulnerabilities arise in E-commerce. The reasons are not exclusive to these systems but their impact becomes much greater because of the wide exposure that an online website has and because of the financial nature of the transactions.

SQL Injection

SQL injection is a phenomenon with which malware author insert SQL characters in field of user input. Through which queries are executed at the back-end of database. If the e-commerce website is vulnerable to such attacks, they have the power to attack even the restricted areas of website. Typically, attackers will first determine if a site is vulnerable to such an attack by sending in the single-quote (') character. The results from an SQL injection attack on a vulnerable site may range from a detailed error message, which discloses the back-end technology being used, or allowing the attacker to access restricted areas of the site because he manipulated the query to an always-true Boolean value, or it may even allow the execution of operating system commands. Depending upon the knowledge of the attacker, they may steal sensitive data viz. credit card numbers, transaction details, etc. The tendencies of getting such an attack via log in page are common.

Price Manipulation

This is vulnerability where the total payable price of the goods purchased is stored over a hidden HTML field, which is dynamically generated by web page. With use of some tools, the modification of payable amount is changed. For example, An attacker can use a web application proxy such as ACHILLES to simply modify the amount that is payable, when this information flows from the user's browser to the web server.

Cross-Site Scripting

The Cross-site Scripting (XSS) attack is primarily targeted against the end user and due to two factors, Firstly, The lack of input and output validation being done by the web application. Secondly, the trust placed by the end-user in a URL that carries the vulnerable web site's name. It is because of lack of proper input/output validation by the web application that such circumstances are faced by the websites driving commerce. The forms which are present on such website which are basically for feedback or suggestion about the products, here the malware author can induce his own content and make a whole new script running on the victim's system. This way they steal sensitive information and session ID's. It leads to stealing credential of users again.

Weak Authentication and Authorization

The Authentication mechanisms are simple criteria to break into the target system by malware authors. Some of system which does not limit the failed login often gets to face the circumstances of stealing away the credentials of users or even sometimes leading to fake online purchase being some other person. Similarly, if the web site uses HTTP Basic Authentication or does not pass session IDs over SSL (Secure Sockets Layer), an attacker can sniff the traffic to discover user's authentication and/or authorization credentials. This way there is huge risk to authentic user's credentials.

MATERIALS AND METHODS

For providing the data security, a new AES (Advanced Encryption Standard) Algorithm to secure User data and Web data is proposed. AES algorithm is a symmetric key algorithm, which means the same key is used for encryption and decryption of the data. AES carries out its operation on a 4*4 column major order matrix of bytes. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the Cipher text. AES is currently available in three key sizes: 128,192 and 256 bites. The design and strength of all key lengths of the AES algorithm are sufficient to protect classified information up to the secret level. As far as security is concerned, AES is considered to be the more secure than its predecessor DES (Data encryption standards). In my proposed method I am using AES 256 encryption using java and will use java cryptography extension for Encryption, Decryption, Key generation etc.

AES ALGORITHM for text and image

```
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

public class Aes_encrypt_string {

    private static final String ALGO = "AES";
    private static final byte[] keyValue =new byte[] { 'T', 'h', 'e', 'B','e', 's', 't','S', 'e', 'c', 'r','e', 't', 'K', 'e', 'y' };


public static String encrypt(String Data) throws Exception {
        Key key = generateKey();
        Cipher c = Cipher.getInstance(ALGO);
        c.init(Cipher.ENCRYPT_MODE, key);
        byte[] encVal = c.doFinal(Data.getBytes());
        String encryptedValue = new BASE64Encoder().encode(encVal);
        return encryptedValue;
    }

    public static String decrypt(String encryptedData) throws Exception {
        Key key = generateKey();
        Cipher c = Cipher.getInstance(ALGO);
        c.init(Cipher.DECRYPT_MODE, key);
        byte[] decordedValue = new BASE64Decoder().decodeBuffer(encryptedD
        byte[] decValue = c.doFinal(decordedValue);
        String decryptedValue = new String(decValue);
        return decryptedValue;
    }
    private static Key generateKey() throws Exception {
        Key key = new SecretKeySpec(keyValue, ALGO);
        return key;
}
   public static void main(String[] args) throws Exception {
        String st="hello";
        String str=encrypt(st);
        System.out.println("enc-->"+str);
        str=decrypt(str);
        System.out.println("dec==>"+str);
    }
}
```

Practical Implementation

In this methodology, during the user registration process whenever the user request is received on to the Servlets it goes through the AES process. In this proposed method symmetric cipher is used to encrypt and decrypt data. The Encryption and Decryption keys are trivially related to each other that is they may be identical or there is a simple transformation to get one key from the other. Key encryption is called single key or private key encryption. For the text, after the necessary imports, create an instance of the JCE classes *Cipher* and *KeyGenerator*. The *Cipher* class provides the functionally of a cryptographic cipher used for encryption and decryption. The *KeyGenerator* class can be used to generate secret keys for symmetric algorithms. The cipher object is created by using the static factory method *getInstance (String transformation)* from the *Cipher* class. The parameter is the name of the requested transformation, where a transformation is of the form "algorithm" that is ALGO. Next an instance of the *KeyGenerator* class is created which generate secret keys for the AES algorithm. The secret key is generated with the *generatekey ()* method and is later used for encryption and decryption. The cipher objects is initialized since the *getInstance ()* method returns uninitiated objects. The initialization is done with *init (int opmode, Key key) method*. The first parameter determines the operation mode of the cipher. As to encrypt data, ENCRYPT_MODE is used. The second parameter is the secret key which is used for encryption. Now encryption is performed using the *doFinal ()* method on byte array of the input string and will encode the data to obtain the final encrypted data. Now the same cipher is used for encryption, but initiate it for decryption with the previously generated secret key. That is, Initialize the cipher to use decryption mode and pass it the key which will be used for decryption.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

public class Encryption_Decryption {

    static KeyGenerator keyGenerator = null;
    static SecretKey secretKey = null;
    static Cipher cipher = null;

    public Encryption_Decryption() {
        try {
            byte[] keyBytes = new byte[]{0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x0
                0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f};
            secretKey = new SecretKeySpec(keyBytes, "AES");
            cipher = Cipher.getInstance("AES");
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void decrypt(String srcPath, String destPath) {
        File encryptedFile = new File(srcPath);
        File decryptedFile = new File(destPath);
        InputStream inStream = null;
        OutputStream outStream = null;
        try {
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            inStream = new FileInputStream(encryptedFile);
            outStream = new FileOutputStream(decryptedFile);
            byte[] buffer = new byte[1024];
            int len;
            while ((len = inStream.read(buffer)) > 0) {
                outStream.write(cipher.update(buffer, 0, len));
                outStream.flush();
            }
            outStream.write(cipher.doFinal());
            inStream.close();
            outStream.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public String encrypt(String srcPath, String destPath) {
        File rawFile = new File(srcPath);
        File encryptedFile = new File(destPath);
        InputStream inStream = null;
        OutputStream outStream = null;
        try {
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            inStream = new FileInputStream(rawFile);
            outStream = new FileOutputStream(encryptedFile);
            byte[] buffer = new byte[1024];
            int len;
            while ((len = inStream.read(buffer)) > 0) {
                outStream.write(cipher.update(buffer, 0, len));
                outStream.flush();
            }
            outStream.write(cipher.doFinal());
            inStream.close();
            outStream.close();
            return "File Encryption Done";
        } catch (Exception ex) {
            return "File Encryption Fails"+ex;
        }
    }

    public static void main(String[] args) {
```

Now decryption is performed on the byte array of the cipher text. The cipher text is now converted into string and final result will be displayed. *Javax. crypto. Cipher; and javax. crypto. spec. Secret Key Spec*; are the classes used for the cipher encrypt or decryt mode and key generation. By using these classes the encrypted data is reverted back. In general for encryption , the key generate data bytes retrival, these data bytes concatenate with key generated and produce cipher text, which is stored in database. Reverse is the case with decryption. For the image, Import the necessary packages, Get the file or image and create a file for putting the encrypted image. The *encypt()* and *decrypt()* methods are used for encryption and decryption of the image from the file. The *KeyGenerator* class is created which generate secret keys. The secret key is generated with the *generatekey ()* method and is later used for encryption and decryption. The generated key for encryption is generated in bytes. The file or image is first converted into bytes and then those bytes are concatenated with the cipher along with the key and the resultant byte array is generated. The resultant byte array is comprises of image means that the resultant byte array will return us a encrypted image. For decryption of an image read the encypted image from the file and decrypt it using the secret key. The encypted file will be stored in scrpath and decrypted file will be stored in destpath.

Conclusion

E-commerce is widely considered as the buying and selling of products over the internet, but any transaction that is completed exclusively through electronic measures can be considered E-commerce. Many of the indicators promise a bright future for E-commerce. In essence, E-commerce has become a reality, and its prospects and capabilities do not stop at an end. In E-commerce, security, trust and privacy are very significant to achieve E-commerce success. In this paper, E-commerce, various issues and threats of ecommerce are presented. Solutions to different threats and vulnerabilities are also discussed and a method using AES algorithm has been proposed for encryption and decryption of an image and text.

REFERENCES

Sangjae Lee and Hyunchul Ahn, 2011. “The hybrid model of neural networks and genetic algorithms for the design of controls for internet-based systems for business-to-consumer electronic commerce”, Expert Systems with Applications Vol.38, 4326–4338.22

Sengupta, A., C. Mazumdar and M. S. Barik, 2005. “e-Commerce security – A life cycle approach”, Vol. 30, Parts 2 & 3, April/June, pp. 119–140.44

Niranjanamurthy, M. and DR. Dharmendra Chahar, 2013. “The study of E-Commerce Security Issues and Solutions” International Journal of Advanced Research in Computer and Communication Engineering,Vol. 2, Issue 7.

Mohanad Halaweh and Christine Fidler, 2008. "Security Perception in Ecommerce: Conflict between Customer and Organizational Perspectives". The International Multiconference on Computer Science and Information Technology, pp. 443 – 449, ISBN 978-83-60810-14-9-IEEE.35

Debabrata Kar and Panigrahi, S., 2013. ”Prevention of SQL Injection attack using query transformation and hashing”, IEEE 3rd International Advance Computing Conference (IACC),DOI: 10.1109/IAdCC.2013.6514419, Page(s): 1317 – 1323.

Srinivas Avireddy_, Varalakshmi Perumal_, Narayan Gowraj_, Ram Srivatsa Kannan_, Prashanth Thinakaran_, Sundaravadanam Ganapathi_, Jashwant Raj Gunasekarany and Sruthi Prabhu, 2012. An Application Specific Randomized Encryption Algorithm to prevent SQL injection, Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, DOI: 10.1109/ TrustCom.2012.232, Page(s):1327 – 1333.

Liban, A. and Hilles, S.M.S., 2014. “Enhancing Mysql Injector vulnerability checker tool (Mysql Injector) using inference binary search algorithm for blind timing-based attack, ”IEEE 5th Control and System Graduate Research Colloquium (ICSGRC), Page(s): 47 – 52.

Yan, W.N.Y. and Chiu, D.K.W. 2007. ”Enhancing E-Commerce Processes with Alerts and Web Services: A Case Study on Online Credit Card Payment Notification“, International Conference on Machine Learning and Cybernetics, Volume: 7DOI: 10.1109/ICMLC.2007. 4370814, Page(s): 3831 – 3837.

Monika, A. and Raman, D. 2014. Justified Cross-Site Scripting AttacksPrevention from Client-Side, International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397 Vol. 6 No.07.

Vibhakti Mate, Milind Tote and Abdulla Shaik, 2014. “Building A Secure & Anti-Theft Web Application By Detecting And Preventing Owasp Critical Attacks- A Review”, International Conference on Industrial Automation And Computing (ICIAC).

Jyoti Chhikara Ritu Dahiya Neha Garg Monika Rani, 2013. “Phishing & Anti-Phishing Techniques: Case Study”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5.

****