Real-time Speech Emotion Recognition Using Support Vector Machine

Article · March 2014

CITATIONS
0

READS
8

1 author:

Anny Leema
VIT University
81 PUBLICATIONS **100** CITATIONS

SEE PROFILE

PREVENTION OF XSS ATTACKS THROUGH DATA SANITIZATION TECHNIQUES

K. Vijayalakshmi, R. Srimathi, Dr. A. Anny Leema,

Assistant Professor, Department of MCA,
Ethiraj College for Women, Chennai, TamilNadu.
lkviji@gmail.com

Research Scholar, Department of MCA,
Ethiraj College for Women, Chennai, TamilNadu.
1992srimathi@gmail.com

Assistant Professor (Sr. Grade), Department of MCA
B.S. Abdur Rahman University, Chennai, TamilNadu.
annyleema@bsauniv.ac.in

Abstract

Internet plays an important role in our day-to-day life. Data security in web application has become very crucial. XSS attack is considered to be one of the major attacks in web application. XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting (XSS) vulnerability may be used by attackers to bypass access controls such as the same origin policy. To prevent this XSS attack we can enable data sanitization techniques. Data Sanitization is the process of disguising sensitive information and overwriting it with realistic looking but false data of a similar type. Some of the techniques are masking data, shuffling record, substitution and encryption/decryption. Masking data replaces certain fields with a mask character. Substitution technique randomly replaces the content of data with information that looks similar but is completely unrelated to the real details. In shuffling, the data in a column is randomly moved between rows. Shuffling is very effective for large amount of data. Cryptography techniques like encryption and decryption can be applied for data sanitization. In encryption plain text is converted into cipher text vice versa is done for decryption.

Keywords : *XSS, OWASP, Masking, shuffling, substitution, encryption/decryption*

1. INTRODUCTION

Information Systems security management is undoubtedly a critical activity in a world where computing is ubiquitous and information systems are interconnected globally. Information security plays an important role in protecting the data and assets of an organization. Accessing web applications has become a daily routine for many people. We depend on these applications to accomplish transactions,

be it business, personal or otherwise. We interact dynamically with web applications when we access our emails, conduct banking transactions, visit social networking sites, etc. This dynamic nature of the web applications allows users to input information that will determine how website responds to the user.

1.1 Open Web Application Security Project (OWASP) is an open source web application security project which finds the causes of insecure software project. OWASP is a non-profitable organization and it provides powerful security awareness for web application. OWSAP (Open Web Security Application Project) is a community effort to define and share knowledge about web application security approaches .According to Open Web Application Security Project (OWASP), SQL injection (SQLI) and cross site scripting (XSS) are the two most common and critical web application vulnerabilities.

1.2 Cross-site scripting vulnerabilities (XSS) are a security problem that occurs in web applications. They were discovered in the 1990s in the early days of the World Wide Web. They are among the most common and most serious security problems affecting web applications. They are a type of injection problems that enable malicious scripts to be injected into trusted web sites[5]. This is a result of a failure to validate input from the web site users. In this XSS attack, what happens is either the web site fails to neutralize the user input or it does it incorrectly, thus opening an avenue for a host of attacks exploiting for vulnerabilities. XSS attacks are of three types namely reflected, stored and DOM-based. Reflected XSS is executed by the victim's browser and occurs when the victim provides input to the web site. Stored XSS attacks store the malicious script in databases, message forums, comments fields, etc. of the attacked server. The malicious script is executed by visiting users thereby passing their privileges to the attacker[7]. Both reflected and stored XSS are executed on the server side script. On the other hand, DOM-based XSS attacks are executed on the client side[9]. Attackers are able to collect sensitive information from the user's computer.

1.3 Data Sanitization is the process of disguise sensitive information and overwriting it with realistic looking but false data of a similar type. Heuristic methods have been proposed to choose the appropriate data for sanitization in order to hide the sensitive information[10]. Some of the techniques which we are going to use in our proposed system in data sanitization are masking data, shuffling record, substitution and encryption/decryption. Masking data replaces certain fields with a mask character. Substitution technique randomly replaces the content of data with information that looks similar but is completely unrelated to the real details. In shuffling, the data in a column is randomly moved between rows, it is very effective for large amount of data. Cryptography techniques like encryption and decryption can be applied for data sanitization. In encryption plain text is converted into cipher text vice versa is done for decryption.

2. LITERATURE SURVEY

Data Sanitization was discussed by (**Abigail Goldsteen et al., 2014**)[1] a prominent technique for dealing with this trade-off is on-the-fly screen masking of sensitive data in applications. This paper presents a unique hybrid approach to screen-masking by combining the advantages of the context available at the presentation layer with the flexibility and low overhead of masking at the network layer. Their solution enables the identification of sensitive information in the visual context of the screen and then it

automatically generates the masking rules to be enforced at runtime on the network traffic. This approach is more powerful and user friendly than the regular expression based mechanism typically employed the network-based solutions and also it supports the creation of highly expressive masking rules.

Data Sanitization was discussed by (**Vrushali S et al., 2013**) [2] SQL injection and cross site scripting attacks are the most common application layer attacks. Reverse Proxy is a technique used to sanitize the user inputs that may transform into a database attacks. In this technique a data redirector program redirects the user's input to the proxy server before it is sent to the application server. Reverse proxy makes communication between client and server more secure and efficient, user's confidential information is protected.

An article has been released by (**M.I.P. Salas et al., 2014 Elsevier**)[3] The proposed approach makes use of two Security Testing techniques, Penetration Testing and Fault Injection, in order to emulate XSS attack against Web Browser. This technology, combined with WSSecurity (WSS) and Security Tokens, it identify the sender and guarantee the access control to the SOAP messages exchanged. The results show that the use of WSInject, in comparison to soapUI, improves the detection of vulnerability to emulate XSS attack in web browser.

A security analysis was proposed by (**Zhiwei Li et al., 2013**)[4] They conduct a security analysis of five popular web-based password managers security. Unlike password managers, web-based password managers run in the web browser. This system identifies four key security concerns for web-based password managers and, for identify representative vulnerabilities through case studies. Their attacks are severe: in four out of the five password managers, an attacker can learn a user's credentials for arbitrary websites. They find vulnerabilities in diverse features like one time passwords and shared passwords. The root-causes of the vulnerabilities are also diverse: ranging from logic and authorization mistakes to misunderstandings about the web security model, in addition to the typical vulnerabilities like CSRF and XSS.

An article has been released by (**Ben Stock et al., 2013**)[8] to ease the burden of repeated password authentication on multiple web sites, Web browsers provide password managers, which offer to automatically complete password fields on Web pages, after the password has been stored once. The managers operate by simply inserting the clear-text password into the document DOM, and it is accessible by JavaScript. Successful Cross-site Scripting attack can be leveraged by the attacker to read and leak password data which has been provided by the password manager. In this paper, they assess the potential threat through a survey of the current password manager generation and characteristics of password fields in most popular Web sites.

An article has been released by (**Asif Muhammad et al., 2012**)[6] and they discussed an effective and convenient ID management system to handle the problem in Open ID. Open ID is one of the better solutions to manage this task on heterogeneous web applications due to its lightweight and simple protocol. However, it is quite vulnerable to session hijacking, resulting in identity theft of a particular user. In this paper, they present a modified approach, based on double authentication that minimizes

the risk of session hijacking in an Open ID environment. Open ID is one of a good solution for identity management on web application services.

3. PREVENTION USING DATA SANITIZATION TECHNIQUES

Data Sanitization is the process of disguise sensitive information and overwriting it with realistic looking but false data of a similar type. We use the following data sanitization techniques for our proposed system,

3.1 Masking Data
3.2Shuffling Record
3.3Substitution Method
3.4Encryption/Decryption

3.1 Masking Data

Masking data means replacing certain fields with a Mask character (such as an X or Y). This effectively disguises the data content while preserving the same formatting on front end screens and reports.
4346 6454 0020 5787
4493 9238 7315 5379
4297 8296 7496 8726

Figure 1 : Masking Data

Masking operation appears like this:
4346 XXXX XXXX5787
4493 XXXX XXXX5379
4297 XXXX XXXX8726
The masking characters effectively remove the sensitive information from the record while still preserving the look and feel. In figure 1, the data is masked to preserve the security. It would not be hard to regenerate the original credit card number from a masking operation such as: 4297 8296 7485 87XX since the numbers are generated with a specific and well known checksum algorithm. Also we must take care not to mask out potentially required information. A masking operation such as XXXX XXXXXXXX 5379 would strip details of card issuer from the credit card number.

The masking rules into two basic types: Content-based rules that take into consideration only the content of the text and can be defined either by regular expressions or text analytics tools; and Context-based rules are the visual structure of the screen.

3.2 Shuffling Record

The shuffling method is a very common form of data obfuscation, it is similar to the substitution method but it derives the substitution set from the same column of data. The data is randomly shuffled within the column. The shuffling method is also open be reversed if the shuffling algorithm can be deciphered.
Shuffling is a great technique to include in your overall masking approach as it has some real strength. If for instance, to maintain the end of year figures for any financial information in the test database. We can mask the first name of the suppliers and then shuffle the value of their accounts throughout the masked database. If even someone with intimate knowledge of the original database could derive a true data record back to its original values. Shuffling is effective when used on small amounts of record. Shuffle rules are best used on large tables and leave the look and feel of the data intact. They are fast, but great care must be taken to use a sophisticated algorithm to randomize the shuffling of the rows and columns.

3.3 Substitution Techniques

Substitution is one of the most effective methods of applying data masking and being able to preserve the authentic look and feel of the data records .Different types of substitution cipher. The cipher operates on single letters simple **substitution cipher**. The cipher that operates on larger groups of letters is **poly graphic cipher**. A **mono alphabetic cipher** uses fixed substitution over the entire message, **polyalphabetic cipher** uses a number of substitutions at different positions in the message, unit from the plaintext is mapped to the possibilities in the cipher text and vice versa.

3.3.1 Simple Substitution

Substitution of single letters separately is known as simple substitution. It can be demonstrated by writing out the alphabet in some order to represent the substitution is termed as substitution alphabet. The cipher alphabet may be shifted or reversed (creating the Caesar and Arbash ciphers, respectively) or scrambled is mixed alphabet or deranged alphabet. Mixed alphabets may be created by first writing out a keyword, removing repeated letters and then writing all the remaining letters in the alphabet(Figure 2).

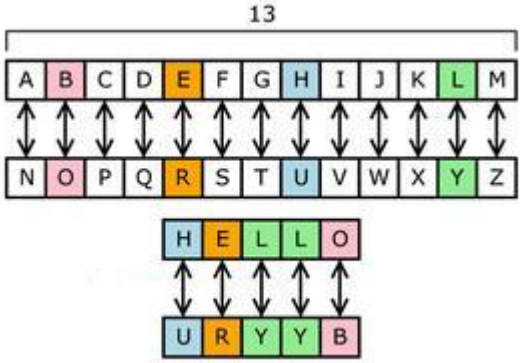

Figure 2 : Substitution technique

Then to sanitize telephone numbers and list of phone numbers must be available. Frequently, the ability to generate known invalid data i.e., credit card numbers that will pass the checksum tests but never work.

3.4 Encryption/Decryption

Encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not itself prevent interception, but it denies the message content to the interceptor.

3.4.1 Symmetric Key Encryption

In symmetric-key schemes, the encryption and decryption keys are the same. The communicating parties must have the same key before they can achieve secret communication. **Symmetric**-key algorithms are a class of algorithms for **cryptography** that use same cryptographic keys for both **encryption** of plaintext and decryption of cipher text. The keys may be identical are there may be a simple transformation to go between the two symmetric keys.

3.4.2 Public Key Encryption

In public-key encryption schemes encryption key is available for anyone to encrypt messages from the sender. The receiving party has access to the decryption private key that enables messages to be read. A publicly available public key encryption application called Pretty Good Privacy (PGP). The Encryption technique algorithmically scrambles the data. This technique offers the option of leaving the data in place and visible to those with the appropriate key while remaining effectively useless to anybody without the key.

4. PROPOSED SYSTEM

We proposed a system with Data Sanitization techniques, which includes masking data, shuffling record, substitution techniques, Encryption/decryption. These four data sanitization technique in figure 3, are used to prevent the XSS attack in web applications. We use our data sanitization mechanism to handle the request from the client and response from the server. If any XSS vulnerabilities or threats are encountered our data sanitization mechanism will detect and prevent from those vulnerabilities or threats.

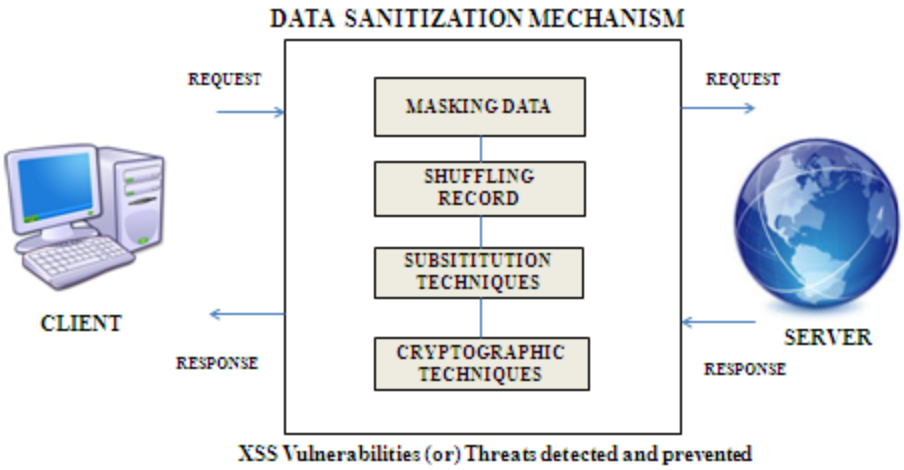

Figure 3 : Proposed System Architecture

Masking data is used to replace certain character in data, when client request a data the response of the server is masked to prevent the attack. Shuffling record is used to shuffle the record to prevent the attack. Shuffling is very effective for small amount of data. Cryptography techniques like encryption and decryption can be applied for data sanitization. The user's confidential information is protected while they perform any online transaction through internet through data sanitization techniques.

5. CONCLUSION

In this paper, various methods for detection and prevention of Cross site scripting attacks are discussed in brief. The goal of Data Sanitization is to handle security issues. We have proposed a Data Sanitization mechanism to prevent XSS attacks in web application. If any XSS vulnerabilities or threats are encountered our data sanitization mechanism will detect and prevent from those vulnerabilities or threats. Data Sanitization is the process of disguise sensitive information and overwriting it with realistic looking but false data of a similar type. With the help of Data Sanitization techniques, we are able to protect web application against XSS attacks.

REFERENCES

[1]. Abigail Goldsteen, KsenyaKveler, Tamar Domany, Igor Gokhman, Boris Rozenberg, Ariel Farkash, "Application-screen Masking: A Hybrid Approach" Information Privacy and Security, IBM Research (2014) pp.423-429

[2]. Vrushali S. Randhe, Archana B. Chougule, DebajyotiMukhopadhyay, "Reverse Proxy Framework Using Sanitization Technique for Intrusion Prevention in Database" CIIT International Conference (2013) pp.343-342

[3]. M.I.P. Salas, E. Martins "Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-Security" Journal of Elsevier (2014) pp.133-154

[4]. Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song, “The Emperor’s New Password Manager: Security Analysis of Web-based Password Managers” pp.465-479

[5]. Kolanoori Pravallika, B.Srinivas Reddy, “XSS Worm Propagation and Detection in Online Social Network” International Journal of Science and Research Impact Factor (2012) pp.458-460

[6]. AsifMuhamma, NitinTripathi, “Evaluation of OpenID Based Double-Factor Authentication for Preventing Session Hijacking in Web Applications” Journal of Computer (2012) pp.623-628

[7]. Divya, Mahesh, R.Ushasree, “Data Implication Attacks on Social Networks with Data Sanitization” International Journal of Current Engineering and Technology (2014) pp.417-422

[8]. Ben Stock, Martin Johns, “Protecting Users Against XSS-based Password Manager Abuse” Journal of ACM (2014).

[9]. Ben Stock, Sebastian Lekies, Tobias Mueller “Precise client-side protection against DOM-based Cross-Site Scripting” pp.478-486

[10]. Abdul RazzaqZahid Anwar, H. Farooq Ahmad, Khalid Latif FaisalMunir, “Ontology For Attack Detection: An Intelligent Approach To Web Application Security” Journal of Elsevier (2014) pp.124-146