Volume 6, Issue 6, June 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

XSS Attacks: Analysis, Prevention & Detection

[1]Monika Rohilla*, [2]Rakesh Kumar, [3]Girdhar Gopal
[1] Scholar, [2] Professor, [3]Assistant Professor
[1, 2, 3] Dept. of Comp. Sci. & App., Kurukshetra University, Kurukshetra, Haryana, India

Abstract- Internet is ubiquitous in every person's life, there is exponential growth in web application services like health insurance, shopping, banking, mailing, news etc. Most of the web applications have security vulnerabilities like XSS (Cross Site Scripting) attacks, phishing attacks which are exploited by the attackers to hack the credential and personal data from the web application for malicious purpose. In this paper XSSattacks have been discussed with their classifications. Selection of victim web application which is vulnerable for XSS attack and some vulnerability scanners are also discussed. Some of the XSS worms are discussed in detail with real life case studies and guidelines to prevent them are also discussed in this paper.

Keywords1- Cross Site Scripting, Vulnerability, Web Applications, XSS Worms.

I. INTRODUCTION

Web applications are a standard platform for data and services provided over the World Wide Web via internet. Web started gain popularity with HTML 2.0, provides specification of text/HTML internet media Type (RFC 159) and MIME content type (RFC 1521). HTML was designed to specify the logical organization of a document. It provides static pages to web applications. Static web sites have the problem of content update and scalability. At the same time, Java Script was introduced that transform the static web pages into partially dynamic pages that resolve the problem of static web sites. With the development of server side scripting language like PHP (Hypertext preprocessor) that provides interactive nature to web applications. This interactive nature of web pages helps in exchange of information between client and server.Today World Wide Web has become a multifaceted platform of web applications. The advanced technologies like ASP, JSP,AJAXetc. have some security issues. These security challenges lead to web application vulnerabilities. Vulnerability is a weakness that can be exploited by an attacker to steal the system and data information. However, now more and more attacks are targeting technical as well as technological flaws in design of web applications. According to PCI (Payment Card Industry) security standard council, 38% of web applicationscontain 98%severe web security vulnerabilities and 55% contains medium web security vulnerabilities (Acunetix, 2015).

This paper is organized in 7 sections. Section II defines the Cross site scripting attack, its types and XSS with phishing. Section III discusses the literature review for detection and prevention of XSS attacks. Section IV defines selection of victim vulnerable web site. Section V lists some black box vulnerability scanners. Section VIdiscusses some real life XSS worms. In section VII Case studies of vulnerable applications are discussed with some guideline. Section VIII presents the conclusion of paper.

II. CROSS SITE SCRIPTING ATTACKS

Cross site scripting (OWASP, Top_10_2013) is one of the most serious problems of web applications that can be exploited by an attacker very easily. An attacker can inject the malicious java script code in web application through any external resource. The malicious java script code is executed by the web browser as a legitimate code. Browser cannot distinguish between legitimate code and injected malicious code. Generally Cross Site Scripting attacks can be classified into three categories.

- Reflected XSS (Non Persistent)
- Stored XSS (Persistent)
- DOM (Document Object Model)

A. Reflected XSS

Reflected XSS is the most common type of XSS in which a page containing a malicious code that is reflected by the browser as a search result. This attack targets the website vulnerability that deals with dynamic property of web application. Every input has potential to be an attack vector. Using social engineering attacker deceives a user to visit a manipulated URL with embedded malicious code. When a user clicks on the malicious link, the embedded code in the URL is to be executed by the web browser.

B. Stored XSS

A malicious script is injected in web application and is permanently stored on the server. When a user requests some information from server, injected script is reflected by a server as an error message or search result. These types of attacks are more serious than other types because an attacker injects a script once and affect a large number of users. Attacker tricks the user to click on the link that contains malicious code. If the malicious script is stored in the server

database then a user becomes a victim by viewing the page without clicking on any link. Here some examples of real world persistent attacks. The Indian security researcher ShubhamUpadhyaya found a new permanent XSS on 13 November 2012 that affected the products listings on ebay.com (KF).

C. DOM (Document Object Model)

DOM based attack exploit the vulnerability of insecure DOM object which can be controlled by client. This type of vulnerability is different from stored and reflected XSS attack because it does not inject malicious script code in page. It is used in HTML and XML. It is a program that modifies the content, structure and style of document.

Cross Site Scripting with Phishing: Cross site scripting vulnerability can be exploited by phishing attack also. Phishing is a method to tricking the user to take out the sensitive information like password, credit card number, bank account information and personal data using social engineering techniques(Milletary). The phishing attack misdirects the victim to fake site. But if the Phishing siteis the real site, this kind of Phishing attack is more dangerous, since users trust the real site. Such attacks can be achieved, when a site is XSS vulnerable. The steps below will demonstrate sample of this kind of attack.

a. Find a vulnerable web site.
b. Embed malicious content into a fraudulent email. Attacker can encode URL to unclear the true destination.
c. Send the spoofed email to victims. When a user clicks on the spoofed email then a login page turned to the user. User login to fake page and fill their credential information that is sent by the attacker. The user is not aware of this and logs in with his personal information, which will be sent to the attacker. After login, the user will be redirected to the real site.

III. LITERATURE REVIEW

Various approaches have been implemented for detecting the different types of XSS attacks. There is no standard technique to mitigate and prevent all types of attacks. A variety of techniques like static analysis, dynamic analysis, proxy firewall, and sanitization can be used for the prevention of client side and server side attacks. Some of these techniques are as follows.

A. Static Analysis

A static string analyzer (Minamide, 2005) checks the string output of a program with context free grammar. This technique checks the presence of "<script>" tag in the whole document. The web application vulnerability is detected by a static analysis tool pixy (Jovanovic, Kruegel, & Kirda, 2006). The authors address the problem of vulnerable web application by means of static source code analysis. The flow sensitivity data flow analysis was used to discover the vulnerable point in program. Non persistent attack is detected by matching the incoming data and outgoing java script using a simple metric like matching the incoming data to HTML java script code. deDacota(deDcota: Toward Preventing Server - Side XSS via Automatic Code and Data Seperation, 2013)statically rewrites the existing application to separate the code and data. The static analysis is used to find out all inline java script code in web pages. There is a second order problem, dynamic inline script. deDacota provides the partial solution to this problem. A prototype of deDacota is implemented to analyze and rewrite ASP.Net web applications. The performance test is applied to check the functionality of web application and they found, there is no difference in page loading time of original and rewritten .

1. Static as well as Dynamic analysis

Static and dynamic analysis is used to identify the faulty sanitization procedure that can be bypassed by an attacker. A tool saner (Balzarotti, et al., 2008) was used for implementation. Experiment was done on various real world applications. They identified the novel vulnerability that stem from incorrect and incomplete sanitization. A static analysis tool (Jovanovic, Kruegel, & Kirda, 2006) was used for detecting the web application vulnerability. Flow sensitivity, inter procedural and context sensitivity data flow analysis was used to discover vulnerable point in program. Static analysis provide false positive. If the number of false positive is large it means site is vulnerable to XSS attack, so they perform dynamic analysis. Dynamic analysis was performed by checking the code with various input value which have different way of encoding, then try to understand which type of input be a cause of security violation.

Using machine learning algorithm (A & P, 2014) (Naïve Bayes, Support vector Machine and J48 Decision Tree) they classify a normal page and malicious page based on the feature extracted from URL and java script code. After experiment, the authors find that J48 provides better performance with respect to FPR (False Positive Rate). But time required for file creation was greater than NB (Naïve Bayes) classifier but less than SVM (Support Vector Machine). To prevent the cross site scripting in server side, White box vulnerability scanner and black box (Acunetix, 2008) (2008) web application testing tool is purposed by various researchers. To identify the technical flaws, scanner is the best.

2. Proxy firewall

A proxy firewall AppShield (Scott & Sharp, 2002) is used to mitigate the XSS attack by learning from the traffic to a specific application. AppShield is a plug and play application that provides limited protection from attacks because it has lack of security policies. A major drawback of this solution is that it protects web application at deployment phase rather than development phase. NOXES (Kirda, Jovanovic, Kruegel, & Vigna, 2009) was the first client side solution to mitigate the cross site scripting attack. Noxes focuses on ensuring the confidentiality of sensitive data by analyzing the flow of data through the browser. Duraisamy (Duraisamy, Kannan, & Selvamani, 2010)described web proxy to protect the information leakage from the user environment. Shalini(Shalini & Usha, 2011)proposed a model that provides a client side solution which does not degrade the performance of application. It provides efficient security from the XSS attacks with optimized web browsing.

3. Input and output sanitization

Server side sanitizer prevents XSS attack by checking existing sanitization correctness. Correct sanitization is a challenging task in web application. It is difficult to guarantee that all part of web application are covered (Balzarotti, et al., 2008)(Akhawe, Saxsena, & Weinberge, 2011). Firstly, find all the paths through which attack can be done. A single input might appear in different context in the output of application(Saxena, Molnar, & Livshits, 2011). HTML input filter (Duraisamy, Sathiyamoorthy, & Chandrasekar, 2013) is used against the security of web application in browser side. A server side solution is proposed against XSS attacks, this solution is not depends on web application provider. The authors reduce the amount of information leakage in browser side. This solution allows easy integration of filter (Java Script Filter) in java based application.

In Browser enforced embedded policy(Jim, Swamy, & Hicks, 2007) modified the web application and embedded some policies. Policy contains a hook function that will run before execution of any script. Modifications to browser and web application are not difiicult to perform but some browser can not support hook function. BEEP did not provide any guidance to trust on third party. According to our web application it suffers from scalability problem. Web request and server response (Vigna, 2005) are used to detect the XSS attack. The web request parameter passed to the HTML parser. They modified the HTML, JAVA script tags, method, method calls and expression with tokens. The features are extracted based on syntax tree and compared to the white list. If malicious tags and malicious scripts are present, alert message of XSS is generated. There was no requirement of modification in browser or server engine. It has weak validation from client side and server side. Johns proposed a prototype for detection of reflected XSS attack (Johns, Engelmann, & Posegga, 2008). The authors define two tasks that require special attention: Script extraction and script parsing. The identification of all the script is a non-trivial task (Hansen, 2007) . A novel approach Dynamic Hash Generation(Gupta, Sharma , Gupta, & Gupta, 2012) makes the cookies useless for the attacker. This approach is easy to implement on web server without any changes required on web browser. The purposed technique does not affect the performance of client side web browser and there is no single point of failure. It is a server side solution; it affects the performance of whole system. Major disadvantage of this solution is that it does not intercept the HTTP and SSL connection

IV. SELECTION OF VICTIM VULNERABLE WEBSITE

Most of the web applications contains search engine. When a user search anything then result is displayed on the screen.Search result is generated dynamically according to the user input without sanitization.Here is an example of web site that is vulnerable to attack.

A. Real life Example of XSS Vulnerable web site

A Web site is vulnerable to XSS attack if injected input is shown by the user without validation. Here is an example of web site: http://www.itworld.com/.

XSS Vulnerable link:http://www.itworld.com/find/results/%22-alert%28%22XSS%20-%20Sasori%22%29-%22

Fig. 1 Seacrh for XSS vulnerability

When search phrase (/%22-alert%28%22XSS%20-%20Sasori%22%29-%22) is inserted to the search box as shown in figure 1. User input is not properly sanitized by the server and displayed back to the user. This script may contain some XSS attack.

B. Demonstration for extracting the cookie information

In figure 2 a feedback form is demonstrated. In which any user can provide their name and comments but a malicious user can send script. In this example attacker injects a script as shown in figure2 that take the cookies informtion and send it to the print_cookie.php page, which is in attacker controll.

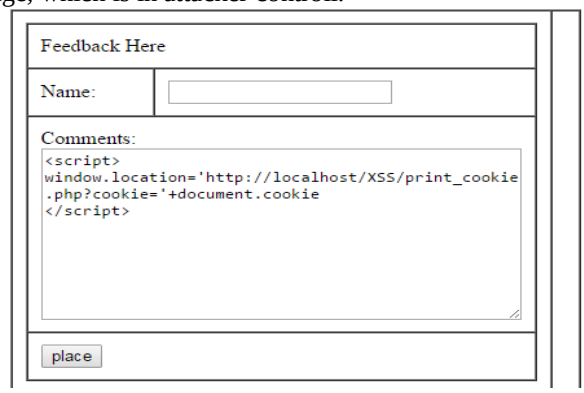

Fig. 2 Cross Site Scripting in Comment using window.location

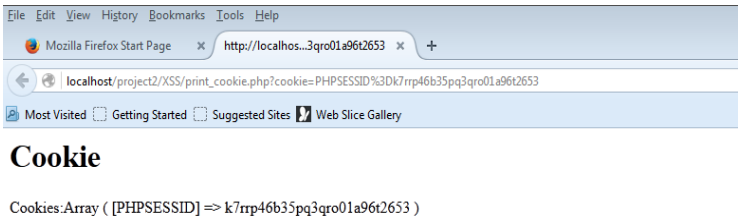

Fig.3Stealing of Cookie Information

When attacker clicks on the place button then cookies information get stored on print_cookie.php page as shown in figure 3.From print_cookie.php attacker collect these cookies and can misuse these cookies information.

V. XSS BLACK BOX VULNERABILITY SCANNERS

Vulnerability scanner (OWASP, Category:Vulnerability Scanning Tools) is a tool that exploit the weakness of the web application without analyzing the source code of web application. Table 1 lists some black box scanners with their owner and licence information. Some scanners are free of cost. Generally their cost lies between the 10 to 1000 dollars.

Table 1 list of XSS black box vulnerability Scanners

Name	Owner	License
Acunetix WVS	Acunetix	Commercial/Free(LimitedCapability)
AppScan	IBM	Commercial
NetSparker	MavitunaSecurity	Commercial
Web Inspect	HP	Commercial
WebApp 360	Trip Wire	Commercial

VI. REAL WORLD XSS WORMS

In recent survey various worms are found on real world applications like (Facebook, twitter, ebay etc.). Mainly these worms are persistent in nature. Table 2 shows the list of XSS worms with their discovery date.

Table 2 several real world XSSworm (Gupta & Gupta, 2015)

XSS Worm	Discovered In
MyspaceSamy Worm	October,2005
Xanga	December, 2005
JavaScript Yamanner Worm	June, 2006
SpaceFlash Worm	July, 2006
My Year Book	July, 2006
Gaia	January, 2007
U-Dominion	January, 2007
Orkut 'BomSabado' Worm	December, 2007
Hi5	December, 2007
Justin.tv	June, 2008
Twitter	April, 2009
Renren	2009
Apache Tomcat	February, 2010
Boonana Java Worm	2010
Facebook XSS Worm	March, 2011

Some of the popular worms are discussed below:

A. Samy Worm (2005)

Samy Worm was first self-propagated XSS worm onto (Samy APPLIED HACKING)MySpace. It was persistent in nature. The worm carried the payload that will print the string "Samy is my hero". On victim profile and causes the victim to send the request to Kamkar unknowingly. When a user view the profile then payload is planted to their page. Within 20 hours 1 million user had effected with payload.

B. Space Flash Worm (18 July, 2006)

Space Flash worm is executed when a user visit user's About me page on myspace.com. This worm redirected the user to URL [http:] editprofile.myspace.com/index.swf?fuseaction=blog.view[REMOVED]

This URL contains a file with .swfextension to retrieve the cookie. It extracts the java script code snippet from the blog entry URL and executes it. It remove the usersAbout me page and add a different src.

<embed src="hhttp://i105.photobucket.com/album/m225/[REMOVED]by[REMOVED]

This worm affects the Window 2000, Window 95, Window 98, Window me, Window NT, Window server 2003, Window XP.

C. Yamanner Worm

The Yamanner worm is a computer worm that is written in java script that exploits the vulnerability in Yahoo mail services. The worm spread through Yahoo mail. This worm affect each and every one who open the email then send the address book to remote server. This worm connects to the http://av3.net/index.htm to send email addresses that are collected from Yahoo mail folder. This mail comes from the address and new graphics site as subjects. Note is placed like: forward message attached this is test.

Technical Details

This worm is applied to onload event handling. When email is opened then automatically worm is executed. Intellishied provide the malicious code alert for the worm. Alert Bit Defender also provides alert for jsBlackworm an alias of Yamanner. Some popular web applications have same vulnerability. Yahoo provides an alert to this worm.

Guidelines:The Company advised user to change their firewall software and block the address "av3Zyahoo.com" from which mail is received.Administrator turns off the java script in web browser.

A. Twitter Worm 2011

Mikeyy worm spread through twitter by trick the people to click on a link to rival microblogging service StalkDaily.com. When you click on that link this will begin to send the message to all your followers and encourage another users to click on it.Mostly United States users are affected by clicking on well over 18000 times.

Guidelines**:** A security firm Fsecure crafts a worm and user received a message that telling them how to remove mikeyy and encourage the people to click on the link URL bit.ly. The bit.ly redirects the users to twitter profile name "reberbreber" while the user also infected. Since this variant of worm used bit.ly to redirects the file. From these efforts Fsecure was able to track the worm.

B. Facebook XSS worm

XSS vulnerability is exploited on Facebook by self-propagating spam worm on social network. The vulnerability is existed because there is insufficient java script validation. To exploit the attack, attacker creates a page that contains a specially designed crafted iframe element that forces the user of facebook to visit it to post the rouge message on their walls. The attacker send spammed/malicious message to trick the user to visit the malicious site. The attack

Guidelines

- Facebook security team use a security tool that block the user from going to infected URL.
- No scriptextension for Firefox browser is able to detect the XSS worm attack.

VII. CASE STUDIES

Several case studies are based on several XSS and XSS with phishing attacks. Someone apparently identified these XSS vulnerabilities. In these attacks a huge amount of data and system information are breached. Some of real life case studies are discussed below.

A. The Anthem Data Breach (2014)

The Anthem data breach was advanced persistent attack with phishing**.** The attacker gained unauthorized access to Anthem's IT system and obtained personal information of current and former members. The scope of breach was 80 million records exposed. The Anthem breach may have started in April 2014 (anthem-breach-may-have-started-in-april-2014/, 2015). In Anthem data breach it seemed there was a lack of encryption on anthem data. According to Wall Street journals, forensic investigation and security experts found that various server and attack tools were used in this breach. The attacker created various fake domains to trick Anthem employees with phishing emails and tempt to fake sites. Anthem attacker created a fake domain name "we11point[dot]com" (based on the "wellpoint.com" the former name of Anthem). When a mapping is performed between internet address and domain name, this reveals other sub domain that are associated with "we11point[dot]com" site.

Guidelines:For preventing such type of attacksdeny all incoming connections and only allow services they explicitly want to offer to the outside world. Reviewing the account statement and should monitoring the free credit reports.

B. Facebook

Facebookinvolve third party application which are served externally but hosted with Facebook URL. Hidden from id can be used to find the session information. Attacker embeds a form into an innocent page. When a user logged in and views that page, then page would automatically submit. It is important to protect the hidden ID form and session from third party application. Using form_id, elements of the page can be easily accessed. The following coding can be used to find out the secret form-id and change the way of displaying the user name in Facebook profile (Felt).

```
varattr = document.getElementById("post_form_id").attributes;
var hidden = attr.getNamedItem("value").value;
document.styleSheets.insertRule('.profile_name h2 {
color: #aa1c75; text-transform: uppercase; letter-spacing:6px;}',0);
```

Guidelines:Strictly validate before inserting untrusted data in HTML style. Escape the URL before inserting into Varattr = document.getElementById("post_form_id").style.filter should be used to save form_id from attackers.

C. eBay XSS Attacks

On 13 November 2012 The Indian Security Researchers ShubhamUpadhyay find a persistent XSS attack that affect the product listing on ebay.com For exploiting this vulnerability you need a seller's account (KF). When a customer login to the seller account on eBay and create a list of item for sale then script will be executed in HTML code
</style></script>< script>alert("XXSed by Cb3r_Shubh4m") </script>
The malicious javascript code can be executed with IFrame on www.ebay.com.When a user clicks on print button then a temporary link open.
http://www.ebay.com/item/ws/ebayISAPI.dll?
gGZ3pfope JXsxzOi4IMd7G4X42Q%3D&4print=all&category=172602
It also execute in cgi.ebay.com domain when user logged in seller account.
In 2014 eBay announce the eBay Breach. This breach affects 145 million customer who were urged (persistently to persuade customer) to change their password.
Guideline: This malicious script was executed with iframe. XFrame header option should be used. Three options should set for Xframe. They are defined as:

- Deny: deny prevent any domain from framing the content.
- Sameorigin: It will allow the frame content from same origin.
- Allow from URI: Define third party url that you want to allow in your web application.

Onbeforeunload() is event handler that prevent the user from navigation to malicious link. It provides an alert message to the user and asks do you want to navigate? User can cancel the navigation and prevents itself from attacks. Disadvantage of this solution is that it requires user involvement. If users do not aware of these attacks then he cannot understand what should do?

D. Apple Pay Fraud (4 June 2015)

InApple pay fraud system Wander's security researchers find a social engineering method through which hacker inject a malicious hacker controlled page. According to Wander's security researchers all the users of Apple Pay on device with Wi Fi service enable (WYK). When a user goes to a different Wi-Fi network, mobile get connected automatically to malicious network. In attacker controlled network attacker can inject the malicious software through which attacker exploit the vulnerability. Page contains the same information as Apple Pay enrolment process and encourages the users to enter their credit card details. Users can not differentiate between fake card entry page and real one. Hacker exploits the users trust.
Guideline:For the prevention of such type of attack payment industry should provide consumer guidance to distinguish between legitimate page request and fake request page.
Application should positively identify the user when any credit card details for taxi service or original wallet are accepted.
Data and system breaches in web sites and organizations:
According to 2015 survey done by the white hat security team (Website Security Statistics Report, 2015), there were 118 responses. 24% of the survey respondent experienced a data or system breach. 17% of the respondent experienced finance/ insurancedata and system breach and 20% have experienced information data and system breach.

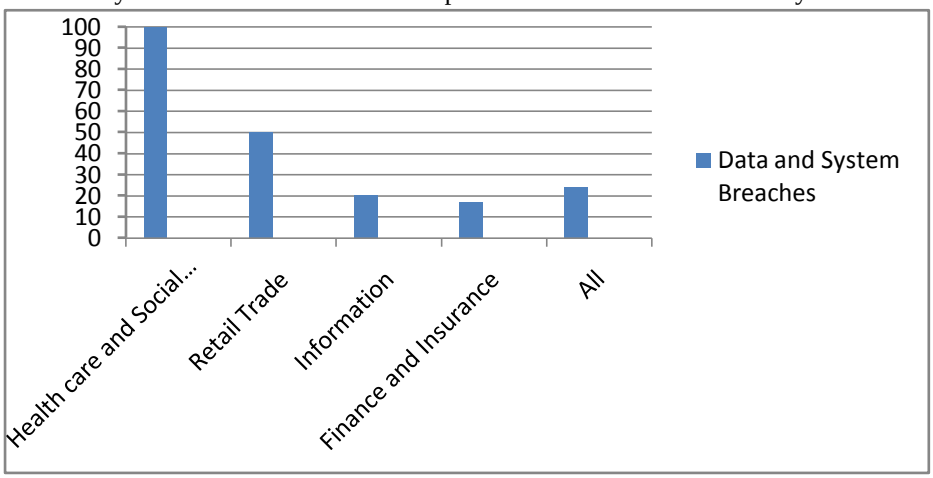

Fig.4 Data and System Breaches using Application Layer Vulnerability

Threat From Cross site scripting Worm
Some of the common threats from XSS attacks are listed below:
Cookie Theft and account hijacking: One of the most severe impacts of cross site scripting attack is theft of cookies and account hijacking. An attacker can stole the credential information stored in cookies thus it is very easy for attacker to steal the user identity. It means, normal user personal data such as credit card number bank account number may be misused by attackers. If a user has administrator level privileges and these are stolen by XSS then attackers are able to access the database system and have full control over the web application.

*Misinformation:*A harmful threat form XSS is the credential misinformation. Attacker can inject malicious code in application which spy on user's surf behavior. Thus attackers gain the behavior of the users and history of sites visited. This information may lose the privacy of the users. A malicious code can modify the content of the page if it is executed by the web browser. Thus attackers can modify the stock price of companies, important messages, price of products which leads to problem of integrity. Using phishing attack an infected script can modify the login page and user submits their credential information like credit card details password etc. to the illegitimate login page.

*Denial of Service:*Web applications accessibility plays an important role in web the enterprise. Web application should be accessible all the time. The spread of XSS code in web application make a user's browser crash by throwing the infinite alert boxes. User cannot reach to the web application any more. The DDoS attack can be enabled by exploiting the persistent XSS vulnerability that allowed the attacker to inject JavaScript code into the <img> tag associated with the image. As a result, every time the image was loaded the malicious code also loaded inside the page. When a user views the video, java script is activated and adding a hidden iframe with DDOS tool that send get request to various sites in every second. If script has a large size video and more viewers thus more powerful DDOS attack.

Browser Exploitation: malicious script can redirect client browsers to an attacker's site,so that the attacker is able to take advantage of specific security hole in web browsers toControl user's computer by executing arbitrary commands, such as to install Trojan horsePrograms on the client or upload local data containing sensitive information.

VIII. CONCLUSION

Nowadays web applications are used all around the world. Cross site scripting is considered as a serious vulnerability in Web applications. Most of the web applications are vulnerable nowadays. In this paper a demonstration of cookie information hack is shown. Also a detailed survey is conducted on Cross Site Scripting attack with various real life worms to show how easy it is to exploit the vulnerabilities of a web based application. These worms affect the banking, social networking, health care, retail services etc. A serious effect of these worms on web applications like stealing cookie details, credit card number, password and data breaches. Case studies of Anthem Breach, Facebook, eBay and Apple pay are discussed with their guidelines to prevent these attacks. These Guidelines provide a way how to secure a web applications from Cross site scripting attacks.

REFERENCES

[1] Gupta, S., Sharma , L., Gupta, M., & Gupta, S. (2012). Prevention of Cross-Site Scripting Vulnerabilities using Dynamic Hash Generation Technique on the Server Side. International Journal of Advanced Computer Research, 2(5), Start Page- 49. (2008). (Acunetix) Retrieved from http://www.acunetix.com

[2] anthem-breach-may-have-started-in-april-2014/. (2015, 02). Retrieved from http://krebsonsecurity.com. (2015). Website Security Statistics Report. White Hat Security.

[3] A, V. B., & P, J. K. (2014). Prediction of Cross Site Scripting Attack Using Machine Learning Algorithm. ICONIAAC. Amritapuri India: ACM.

[4] Acunetix. (2015). Acunetix Web Application Vulnerability Report 2015.

[5] Akhawe, B., Saxsena, F., & Weinberge, S. (2011). A Systematic Analysis od XSS Sanitization in Web Application Framework. ESORICS'11 Proceeding of the 16th European Conference on Research in Computer Security. ACM.

[6] Balzarotti, D., Cova, M., Felmetsger, V., Jovanovic, N., Kirda, E., Kruegel, C., et al. (2008). Paper Review: Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications. Security and Privacy, 2008 IEEE symposium (pp. 378-401). Oakland CA: IEEE.

[7] Doupe, A., Cui, W., & Jakubowski, M. H. (2013). deDcota: Toward Preventing Server - Side XSS via Automatic Code and Data Seperation. CCS'11. Berlin Germany: ACM.

[8] Duraisamy, A., Sathiyamoorthy, M., & Chandrasekar, S. (2013, March). A Server Side Solution for Protecting of Web Application from Cross Siye Scripting Attack. International Journal Of Innovative Technology and Exploring Engineering (IJITEE), 2(4).

[9] Duraisamy, Kannan, & Selvamani. (2010). Protection of Web Application from Cross Site Scripting Attack in Browser Side. IJCSIS, 229-236.

[10] Felt, A. (n.d.). Defacing Facebook: A Security Case Study. Retrieved from felt@virginia.edu, www.cs.virginia.edu/felt/fbook

[11] Gupta, S., & Gupta, B. (2015). Cross-Site Scripting (XSS) attacks and defense mechanisms: State- of- the-art. Springer.

[12] Hansen. (2007, 05 05). Cross- Site Scripting Cheat Sheet. Retrieved from http://hacker.org/XSS.html

[13] Jim, T., Swamy, N., & Hicks, M. (2007). Defeating Script Injection Attacks with Browser - Enforced Embedded Policies. Interntional World Wide Conference Committee (IW3C2). Canada: ACM.

[14] Johns, M., Engelmann, B., & Posegga, J. (2008). XSSDS: Server-Side Detection of Cross-Site Scripting Attacks. Computer Security Applications Conference, 2008. ACSAC 2008. Annual (pp. 335-344). Anaheim, CA: IEEE.

[15] Jovanovic, N., Kruegel, C., & Kirda, E. (2006). Pixy: a static analysis tool for detecting Web application vulnerabilities. 2006 IEEE Symposium on Security and Privacy (S&P'06) (pp. 6pp.-263). Berkeley/Oakland, CA: IEEE.

[16] KF. (n.d.). Another Ebay permanent XSS. Retrieved from http://www.xssed.com/news/131/Another_Ebay_permanent_XSS/
[17] Kirda, E., Jovanovic, N., Kruegel, C., & Vigna, G. (2009). Client - Side Cross Site Scripting protection.
[18] Milletary, J. (n.d.). Technical Trends in Phishing Attacks. US-CERT.
[19] Minamide, Y. (2005). Static Approximation of Dynamically Generated Web Pages. WWW '05 Proceedings of the 14th international conference on World Wide (pp. 432-441). New York, NY, USA: ACM.
[20] OWASP. (n.d.). Category:Vulnerability Scanning Tools. Retrieved from https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools
[21] OWASP. (n.d.). Top_10_2013-Top_10. Retrieved from https://www.owasp.org/index.php/
[22] Samy APPLIED HACKING. (n.d.). Retrieved from http://samy.pl/popular/
[23] Saxena, P., Molnar, D., & Livshits, B. (2011). ScriptGard: Automatic Context-Sensitive Sanitization for Large-Scale Legacy Web Applications. CCS'11. Chicago, Illinois, USA: ACM.
[24] Scott, D., & Sharp, R. (2002). Abstracting application-level web security. WWW '02 Proceedings of the 11th international conference on World Wide Web (pp. 396-407). ACM New York, NY, USA: ACM.
[25] Shalini, & Usha. (2011). Prevention of Cross Site Scriptig Attack (XSS) on Web Application In The Client Side. IJCSI International Journal Of Computer Science Issue.
[26] Vigna, H. (2005). Detecting malicious JavaScript code in Mozilla. 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05) (pp. 85-94). IEEE.
[27] Wueest, C. (2011). New XSS Facebook Worm Allows Automatic Wall Posts. Retrieved from http://www.symantec.com/connect/blogs/new-xss-facebook-worm-allows-automatic-wall-posts
[28] WYK, K. V. (n.d.). Apple Pay's weakest link. Retrieved from http://www.computerworld.com/article/2926733/mobile-payments/apple-pay-s-weakest-link.html

