ISSN:0975-9646

Model To Prevent Websites From XSS Vulnerabilities

Tejinder Singh Mehta , Sanjay Jamwal

Department of Computer Science,
Baba Ghulam Shah Badshah University, Rajouri, J&K, India

***Abstract*— As social websites get more and more users across internet, Cross Site Scripting is becoming one of the major problems, which results in serious consequences, such as theft of some personal trusted data and information. This Paper describes the possibilities of securing web applications on client side as well as on server side. The attacks are the worst because they are easy to make but very tough to be traced. Cross-Site Scripting is one of the major attacks of many Web based Applications. Web browsers support the execution of Malicious JavaScript and Attacker access the system feature maliciously to violate the security such as confidentiality. Networking sites (i.e. social network) provide the attacker with flexibility to put there malicious code into the web applications. Detecting these malicious script codes is very tough for client side; the detection can be done by using detection tools both at client end and server end as well. Our approach is to minimize theft space for such unlawful contents by using QualysGuard (WAS) tool, so to minimize the vulnerabilities to cause any harm to web applications. This paper will help us to overcome from this problem and elaborates on the possibilities to reduce this problem of Cross-Site Scripting attack to some extent.**

***Keywords*— XSS Cross Site Scripting,DOM Document Object Model, HTML Hypertext Markup Language, HTTP Hypertext Transfer Protocol, IDS Intrusion Detection System, URL Uniform Resource Locator, WAS Web Application Scanning**

INTRODUCTION

Cross-Site Scripting (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. In particular, scanners face problems in detecting stored XSS properly.

XSS can be defined as a security exploit in which an attacker inserts malicious code into a web page returned by a web server trusted by a user. This code may reside on the web server or be explicitly inserted when the user browses to a site, it may contain JavaScript or just HTML, and it may use third party sites as sources or rely only upon the resources of the targeted server. XSS attacks typically involve JavaScript code from a malicious web server executing on a user's web browser.

XSS is one of the most common web application layer attacks that hackers use to reflect the malicious code to victim users. Also use to deface or hijack websites, enable malicious phishing attacks, and provide entry points for larger-scale attacks against business assets and user data. Statistic breakdown of web security vulnerabilities in the first half of 2009, to gives the reader a rough idea of what are the major security problems through which a websites and a web applications suffers.

After an application on a website is known to be vulnerable to cross-site scripting XSS, an attacker can formulate an attack. The technique most often used by attackers is to inject JavaScript, VBScript, ActiveX, HTML, or Flash for execution on a victim's system with the victim's privileges. Once an attack is activated, everything from account hijacking, changing of user settings, cookie theft and poisoning, or false advertising is possible.

To understand cross site scripting several theories and techniques by which an attacker put his malicious contents so that trusted user can be get into the hacked list. As the user trust the browser and the malicious script runs automatically. Injection of script into a field which user used to search valid information can be a valid vector, but if filters were used then what will happen? It is to bypass the filters, Because of the fact that XSS is constantly a major problem among all attacks with new methods of injection and exploiting the code.

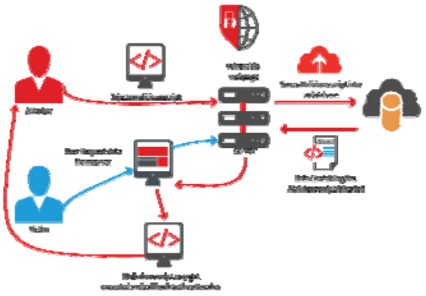


Fig1: XSS attack [51]

II. Statement of the problem

There are a large number of ways an attacker can put their code in browsers. Cross Site Scripting vulnerability that forces web application to prompt malicious code in user browser. The Server is much more authentic, so attacker needs most trusted site to perform such attack. The user is only victim not the server. If attacker gets the control in user's browser so he can perform various problems such as account hijacking, Cookie stealing etc.

Browsers can get deteriorated by such attack only if they visit web application containing JavaScript Malicious malware scripts. JavaScript malware injected to a simple

Html scripting tags and calling exploiting code. There are two primary types of XSS vulnerabilities

A. Reflected (Non Persistent):

Attacker needs to monitor the web application then design URL so user provide the data to server for processing of request and in the mean time clicking on the URL. Attacker hijacks the web site and prompts the error message.

B. Stored (Persistent):

In Stored attack the malware code is embedded in a web page or the vulnerable data stored in web server. So the malicious malware injects everyone into this attack. So this attack is much more powerful and harmful to affect web pages.

The malicious code runs at client side of the participant to compromise its information security blindly. The participated browsers are poor in capability detecting such scripts with assumes that the service providers protected them. Some of these capabilities (all the special characters (e.g., "<", ">", "&", etc.)) need to be identified and encoded if they are included into the output, or they need to be filtered by the web application included into the input. As consequence, the problem should be considered at the client side in default. The accuracy and performance of previous works which used to detect malicious JavaScript attacks that doesn’t satisfies the users need, moreover the generality of the tools is a problem to detect malicious JavaScript code from different websites.

III. RELATED WORKS

A. Works on Anomaly Instruction Detection System:

Anomaly Instruction Detection System is effective in preventing external attackers to detect web browser attacks a signature base method was proposed. They include mechanism to reduce the number of false alarms.

Denning proposed an intrusion detection model which is based on hypothesis that system security violations can be detected by audit record of system usage. This model represents profiles of subject with respect to object and audit records for detecting anomalous behavior.

There are various methods to identify the weakness of an anomaly intrusion detection system. These are techniques which explore threats and suggests for improvement for existing and future anomaly based intrusion detection. There are some techniques for analyzing the structure of parameters, Length and mimicry attacks by the IDS. A multi model proposed to detect web based attacks. This model analyzes client queries which refer to server side programs to create model for different features. This work is selected for three different areas of intrusion detection namely application level intrusion detection, learning based anomaly detection and detection of attacks against web server.

Giovanni et al [1] provided a mechanism for web based anomaly detector and SQL injection detector which increases the detection rate to reduces the error with the possibility for an attacks. Marco Cova et al [2] presented a way to detect the malicious JavaScript code. In this approach the system automatically identifies JavaScript Code and also supports their analysis to establish normal JavaScript code. During detection phases their behavior compares to establish profiles. A new technique came in a way to researcher for preventing intrusions to websites.

B. Policy Based Access Control Techniques or Mechanism:

Marian Ventuneac et al [3] developed a policy to secure framework for web applications. There are certain policies for authentication, Security management parameters and access controls.

David Scott and Richard Sharp ([4], [5]) describe a method for analysis of code. In their approach code analysis is to be done by working on policy based mechanism. After few years attack prevention related to their framework was given by Garlia Alfaro et al [6]. In their proposed work they described mechanism to be putted in kernel of the operating system for protecting the websites from data comprised by developer. Frank S Rietta [7] describes the threats related to XSS and SQL injection by Intrusion Detection System. The model analyzes the traffic from the database. Fredrick Valeur et al [8] describes an anomaly based system which learns the users system and access to database and also check websites to detect both SQL and XSS attacks. Even Unknown attacks can be detected. In fact they come up with the concept of (IDS) Anomaly detection system reverses the http proxy order. With this concept there is a split of contents between sensitive and non sensitive data. This assumption evaluates certain existing web based application, which proves to be a worthwhile approach. Manar H alaf et al [9] describes a frame work to control the access of dynamic web applications. In which they apply reverse engineering to access the security control model and also this framework to verify the control policies. Guillanme et al [10] describes few new methods as combined effect of both information flow and information flow control to check out the operating system level to verify the state of attacks.

C. Website Security analysis for static work:

From proposed work in past the researcher Cook[11], James[12], Vidar[13], Viktoria [14] for the protection of websites from Cross Site Scripting. Our analysis works with two modules static and dynamic methods. There are numerous literature proposed to secure websites from XSS attacks. In almost work static methods were proposed to secure from these attacks. Static analysis method [15] tells us about problems related websites by static code analysis. They analyze the sensitivity of data to find out vulnerabilities in a program. So with this quantity and quality of work is to be generated by applying two phase algorithm for fast results. Information flow [16] approach is to secure website form vulnerabilities like XSS. Static approach by Gary[17] and Zhendong [18] finds a review to input validations and also finds out unknown and known vulnerabilities. Static and dynamic method [19] describes how to reduce runtime for monitoring phases. The most effective analysis for scripting language is too analyzed through byte code to search out vulnerable sites.

Static analysis method by Lachmund [20] describes auto generating access control policies to secure web applications if we use such policies we can mitigate attacks to some extent because of these policies user input got recognition and it also generate access rights for user input. Primarily analysis of static techniques Andrea and Mariano ([21], [22]) describes method for security on integration work and also static check is processed to control the access flow but their work is only static not for dynamic analysis

D. *Work analysis for dynamic:*

Work for dynamic analysis proposed by many researcher Wei Xu,William and Halfond ([23], [24], [25]) describes to secure web application from XSS vulnerabilities. Website security mechanism Yao Wen [26] describes the poor coding which render websites vulnerable to XSS attacks. Dynamic tainting analyses by Doudalis [27] describes the method for illegal access and also check memory allocation at run time. It also provides technique to binary level so to handle third party source code.

E. *Vulnerability Analysis:*

The process that identifies the security issues in a computer network is called as vulnerability assessment. Vulnerability analysis consists of several steps:

- Defining or classifying client network or system resources.
- A blueprint to deal with the potential vector problems first.

If security issues were found due to vulnerability analysis, vulnerability publication information may be required. If the vulnerability is not rated as a high level threat, the dealer may be given a certain amount of time to fix the problem before the vulnerability is reveal publicly.

Website security is an important concern of various organizations as well as financial sectors. In order to provide services to user, Internet is cheapest and frequent way but to protect the highly sensitive data of server, firewall is not only the best solution. The literature survey is to analyze the development in web security for preventing XSS attacks. Our analysis is to form new method to prevent attacks in a web application.

Several approaches which rely to static analysis mechanism to detect vulnerabilities in web applications and it also checks data flow of input validation. Wassermann and Balzarotti ([18], [19]) describes a mechanism for correctness of functions using filter.

F. *Client Side:*

Client Side is a process in which relationship between client and server-side operates in a network system.

Kirda [28] and Vogt [29] describes the client side defense which provides us flow of sensitive data by analysis of web browser but without providing solution to malicious scripts and it protect us from XSS but without an effective defense. Whereas attacker which violates Same Origin Policy are left unprotected.

Comparison of Http parameters with output of web application by filtering the request of response describes the capabilities of both client sides by D Ross and G Maone ([30], [31]) and server side by M Johns and R Sekar([32], [33]) provides scalable solutions for preventing XSS vulnerabilities.

G. *Server Side:*

It is a process which involves sending information to another computer across the internet. The server then runs a piece of code containing information and returns the results, typically a webpage. To minimize the risk of scripts parsing is use as a software tool given by ([34], [35], [36]) and advanced filtering. Server Side protection was purposed by Pietraszek Thorsten and Johns ([37], [38], [39]) describes the detection system to identify successful XSS. In their work they crafted 500000 HTTP responses to check XSS exploits. There approach is having zero false negative and 80% of false positive results. Wurzinger [40] describes mechanism by securing web proxies for server side to find out the vulnerabilities like XSS. They designed a system to reverse the proxy which intercepts html response of request. `Prithvi [41] describes mechanism for input validation to protect from malicious code on server side and also describe the reasons for the failure of filtering. After that a new mechanism called QualysGuard WAS came to market for defense against vulnerabilities.

H. *Browser Manipulations:*

There must be a way so that browser can communicate with content security policies. Several approaches like DSI [42], Nonescape [43] describes the mechanism to preserve the integrity of a document.DSI also explain issues of dynamic protection to the integrity of a document. Browser needs processing power to remove overheads.

XSS attack, JavaScript Malware was considered as serious threats to browser security. Just a simple click and you could get in hacked list and even (SSL) secure socket layer breach of security like heartbleed bug problem shook the whole world that even HTTPS,SSL is not secure. Firewall, antivirus, anti phishing, 2Dl authentication or any other tool couldn’t save them from occurring of such attacks.

Joon and Ravi ([44]) describes the testing tactics and provide mechanism for preventing website from vulnerabilities like XSS, SQL Injection attacks. If any other vulnerability exists in server than it would be very much difficult to protect it on client side system. Further Christopher, Vigna, Robertson ([45], [46], [47]) describes that it would be difficult to protect from IDS. CERT (center of internet expertise) as per their views no client side solution can be completely safe and authentic. Few ideas of research (Jovanovic [48] and Kirda [49]) Wes Masin and Andy Podgurshi stated that, information flow work will increase false positive rate. Some validation mechanism given by Chung Hung [50] scanners posed to prevent XSS vulnerability. Detection techniques can accurately sensitized by false negative and false positive rate. False positive rates among total alerts of vulnerabilities and David Scott suggested policies for input validation and requires correct validating policies for entry point in a web site.

IV METHODOLOGY

A. *Detection Mechanism:*

A website has two important aspects software which runs on server and the html which runs on user's browser. The browser is for user interaction with websites. So browser takes mixture of HTML and JavaScript that is why attacker put malicious code in JavaScript. Cross Site Scripting doesn't merely dependent on JavaScript. Scanners were used for such problems by injecting html and JavaScript on web form and even on cookies and looks for subversion to its regular html contents.

Cross Site Scripting and SQL injection are not actual attack parameter of the link they can target the link by changing parameters. Scanner can also modify the link, but scanner here used for finding the flaws in website. Scanner may purposely find the flaws and weakness in website. Scanner doesn't locate every type of vulnerability but it will reduce the flaw of website to maximum. The critical challenge is to locate the bugs and it can be very time consuming. QualysGuard WAS is a web scanning process. It also defines common behavior to secured site or error pager in websites.

Crawling is a fundamental requirement for web applications scanner. Crawler tells the user how to use browser, so that scanner can better scan all breaches in the security. QualysGuard WAS opposed to test every redundant link. It tests all the operating system programs effectively.

When there is a check introduced to vulnerability a number is to be assigned with it. Like while checking if it founds a damaged link next time it will never check the same link again. The QualysGuard WAS is better because it adjusts it speed automatically and also monitors timeouts of server such as SMTP 480 message sometimes prompted to our system. If scanner produced such error then it cancels the crawl so that no unintentional user can enter in authentic area. Crawling include a part of site which supports JavaScript to create dynamic pages but have certain restrictions as well given below

- Form can be disabled while submission.
- Permission can be denied.
- It points to specific address and doesn't bifurcate the IP address path

Some more granular controls during scan process with QualysGuard WAS like

Blacklists: - The pattern was restricted by scanners.

Whitelists: - The pattern having some exceptions in blacklist entries

Some Websites access requires better authentication so as to access more functionality. QualysGuard WAS authentication can be categorized into two categories

- Server Based Authentication(SSL client authentication and HTTP(Basic)
- HTML Form(Like Login Page)

Through crawl QualysGuard WAS looks for login form controls to be entered in appropriate form. Certain test can tells us about the best practice for handling cookie, by using SSL or cookie attributes. QualysGuarWAS, may also care to configure the scraps for HTML while using credit card number. Then the scanner matches the same pattern to avoid false positives.

The QualysGuard WAS vulnerability management provides us platform in the network and network attached with the hosts is based on cloud based identification. The QualysGaurd WAS automate websites scanning and extends the platform by securing web applications. In cloud computing scenario QualysGaurd WAS acts as a software-as-a-service (SAAS). This means there is no need to purchase software web application software (WAS), but it acts as utility when we need it, we can use it, and through this we obtained centralized access for scanning the web application.

QualysGuard WAS automates the websites scanning process. You don't require any prior knowledge to learn about this website. The scanner automatically finds and determines the behavior to authenticate or to reflect error message in a web application. It also ensures how to keep your web application secure.

Fig 2: Modules Of Qualys [52]

B. *Slow moving web application:*

This part deals with how user uses the browser to interact with the websites. Scanner ability is to locate the breach areas of security. QualysGuard WAS functionality opposes to test every link of website. Web application scanning quickly tells us which website is vulnerable or not. QualysGuard WAS tool is the best tool to check browser with three options:

- Basic Scan: Checks the browser issues and reflect the issues i.e. shown in figure
- Intermediate Scan: Checks the browser issues and system settings to reflect the issues i.e. shown in figure
- Advanced Scan: Checks the browser issues and system settings to reflect the issues with fix the issue with multiple options i.e. shown in figure:

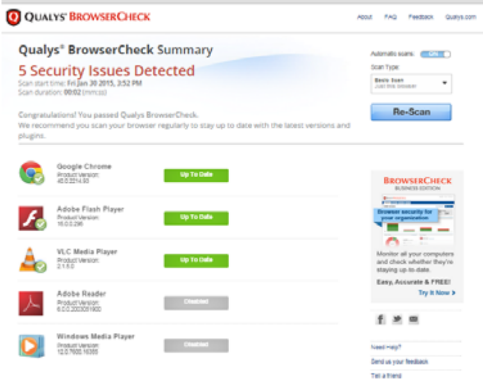
While scanning operating system it checks and assigns QID. If QID is not considered QualysGuard WAS doesn't check the vulnerability and also display all the links of QID on a web page. There are certain adjustments to be made during scanning of a web application are:

- Processing capacity
- Response Time

QualysGuard WAS can handle thousands of request. QualysGuard WAS automatically make adjustments of server side average response time with a threshold scan. QualysGuard WAS monitor timeouts like 404 messages is reflected to your screen time out of server. If too many time outs performed frequently then it cancels crawl requests. Such problems unintentionally cause denial of servers to web applications. QualysGuard Was behaves like normal visitor and sites also crawl which completely rely on dynamic pages. Form authentication can be disabled.

QualysGuard WAS tool is more granular control with scanning. Some websites requires authenticated access to provide functionality. QualysGuard WAS Scanning can be set in two different areas.

- Server based authentication.
- Html forms.



Qualys® BrowserCheck Summary

5 Security Issues Detected

Scan start time: Fri Jan 30 2015, 3:52 PM

Scan duration: 00:02 (mm:ss)

Congratulations! You passed Qualys BrowserCheck.

We recommend you scan your browser regularly to stay up to date with the latest versions and plugins.

Automatic scans: ON

Scan Type: Basic Scan

Google Chrome — Product Version: 40.0.2214.93 — Up To Date

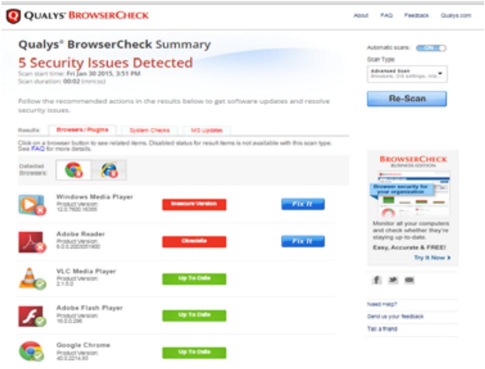
Adobe Flash Player — Product Version: 16.0.0.296 — Up To Date

VLC Media Player — Product Version: 2.1.5.0 — Up To Date

Adobe Reader — Product Version: 6.0.0.2003051900 — Disabled

Windows Media Player — Product Version: 12.0.7600.16385 — Disabled

Fig 3: Basic Browser Check [53]



Qualys® BrowserCheck Summary

5 Security Issues Detected

Scan start time: Fri Jan 30 2015, 3:51 PM

Scan duration: 00:02 (mm:ss)

Follow the recommended actions in the results below to get software updates and resolve security issues.

Automatic scans: ON

Scan Type: Advanced Scan

Results: Browsers / Plugins | System Checks | MS Updates

Click on a browser button to see related items. Disabled status for result items is not available with this scan type. See FAQ for more details.

Detected Browsers:

Windows Media Player — Product Version: 12.0.7600.16385 — Insecure Version — Fix It

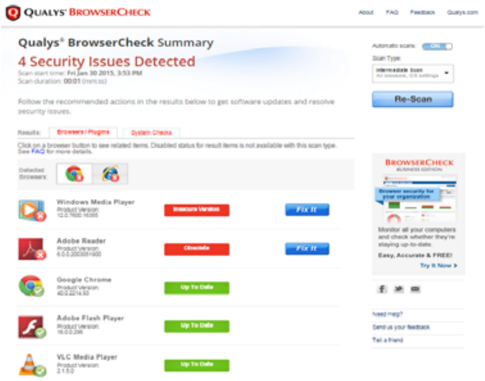
Adobe Reader — Product Version: 6.0.0.2003051900 — Obsolete — Fix It

VLC Media Player — Product Version: 2.1.5.0 — Up To Date

Adobe Flash Player — Product Version: 16.0.0.296 — Up To Date

Google Chrome — Product Version: 40.0.2214.93 — Up To Date

Fig 4: Advanced Scan results with OS [53]



Qualys® BrowserCheck Summary

4 Security Issues Detected

Scan start time: Fri Jan 30 2015, 3:53 PM

Scan duration: 00:01 (mm:ss)

Follow the recommended actions in the results below to get software updates and resolve security issues.

Automatic scans: ON

Scan Type: Intermediate Scan

Results: Browsers / Plugins | System Checks

Click on a browser button to see related items. Disabled status for result items is not available with this scan type. See FAQ for more details.

Detected Browsers:

Windows Media Player — Product Version: 12.0.7600.16385 — Insecure Version — Fix It

Adobe Reader — Product Version: 6.0.0.2003051900 — Obsolete — Fix It

Google Chrome — Product Version: 40.0.2214.93 — Up To Date

Adobe Flash Player — Product Version: 16.0.0.296 — Up To Date

VLC Media Player — Product Version: 2.1.5.0 — Up To Date

Fig 5: Intermediate Scan results [53]

Scanning websites may require advanced settings. Some authentication form prevents WAS tool from access. QualysGuard WAS doesn't always apply active test on web applications. But in case of SQL injection and XSS it performs active test.

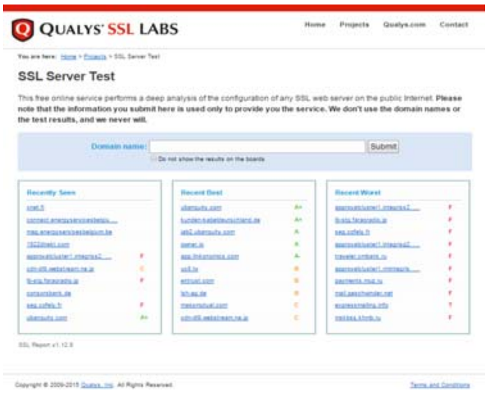
QualysGuard also checks out the traffic of web applications and also provide tools for handling cookies, use of SSL or cookie history. QualysGuard WAS uses vulnerability test in a significant way to reduce the false positive rates. The important concern is of user weather scan rate is safe or not. It is not possible to give 100% result but QualysGuard reduces the vulnerabilities to maximum extent.

V. EVALUATING THE RESULTS WITH SCANNING

Several potential vectors are to be checked for vulnerabilities analysis.

- File parameters of links, IP etc
- Form fields or Input fields
- HTTP headers of browser.

The tests like browser check as shown in above figures. Now Server test is to check and find out errors, coding mistakes and security holes to which compromises to security were made and it also report findings of developers.



You are here: Home > Projects > SSL Server Test

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

Domain name: Submit

□ Do not show the results on the boards

SSL Report v1.12.8

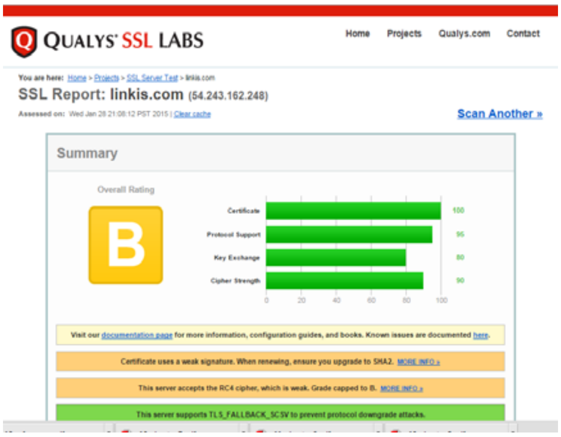
Copyright © 2009-2015 Qualys, Inc. All Rights Reserved. Terms and Conditions

Fig 6: SSL Test Server URL [54]

Information about the web page that display error message revels which IP or malicious code is supplied by an attacker.SSL problem to html form with grading system and other authentication form problems like certificate are not valid or not obtained. The test phase uses 90% of time to scan report and to reduce such scan time QID were used in QualysGuard WAS.

A. *Reducing false positive results:*

Scanner accuracy depends on its usefulness. If you cannot make trust to the results or tool spends too much of time for scanning\g then the tool is not of any use. False positive vulnerability doesn't exist but they were reported by the scanner. It happens because of scanner detection mechanism for detecting was insufficient. But due to custom profiling of data by use of QualysGuard WAS tool reduced the false positive because of certain comparisons across the group of response. Scanner tries to find payloads which were incorporated in database.



You are here: Home > Projects > SSL Server Test > linkis.com

SSL Report: linkis.com (54.243.162.248)

Assessed on: Wed Jan 28 21:08:12 PST 2015 | Clear cache

Scan Another »

Summary

Overall Rating

B

Certificate	100
Protocol Support	95
Key Exchange	80
Cipher Strength	90

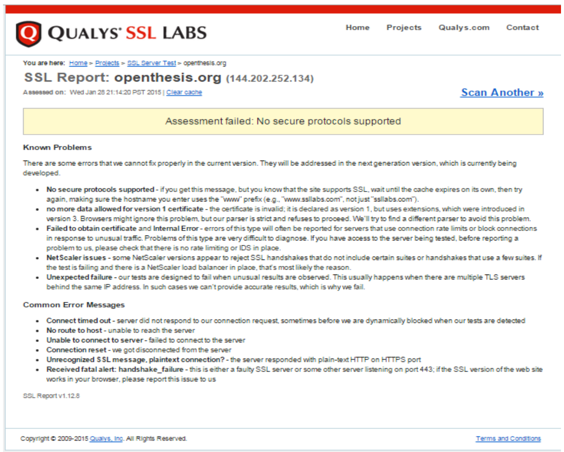
Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

Certificate uses a weak signature. When renewing, ensure you upgrade to SHA2. MORE INFO »

This server accepts the RC4 cipher, which is weak. Grade capped to B. MORE INFO »

This server supports TLS_FALLBACK_SCSV to prevent protocol downgrade attacks.

Fig 7: Scan Check with Grades [55]



You are here: Home > Projects > SSL Server Test > openthesis.org

SSL Report: openthesis.org (144.202.252.134)

Assessed on: Wed Jan 28 21:14:20 PST 2015 | Clear cache

Scan Another »

Assessment failed: No secure protocols supported

Known Problems

There are some errors that we cannot fix properly in the current version. They will be addressed in the next generation version, which is currently being developed.

- **No secure protocols supported** - if you get this message, but you know that the site supports SSL, wait until the cache expires on its own, then try again, making sure the hostname you enter uses the "www" prefix (e.g., "www.ssllabs.com", not just "ssllabs.com").
- **no more data allowed for version 1 certificate** - the certificate is invalid; it is declared as version 1, but uses extensions, which were introduced in version 3. Browsers might ignore this problem, but our parser is strict and refuses to proceed. We'll try to find a different parser to avoid this problem.
- **Failed to obtain certificate** and **Internal Error** - errors of this type will often be reported for servers that use connection rate limits or block connections in response to unusual traffic. Problems of this type are very difficult to diagnose. If you have access to the server being tested, before reporting a problem to us, please check that there is no rate limiting or IDS in place.
- **NetScaler issues** - some NetScaler versions appear to reject SSL handshakes that do not include certain suites or handshakes that use a few suites. If the test is failing and there is a NetScaler load balancer in place, that's most likely the reason.
- **Unexpected failure** - our tests are designed to fail when unusual results are observed. This usually happens when there are multiple TLS servers behind the same IP address. In such cases we can't provide accurate results, which is why we fail.

Common Error Messages

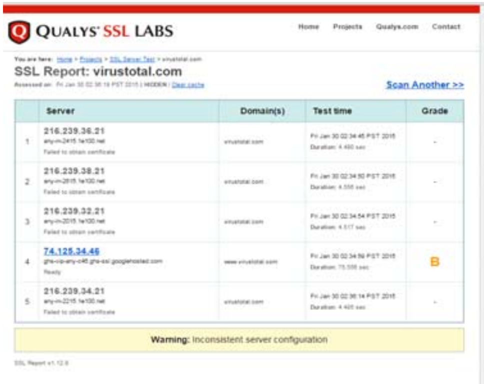
- **Connect timed out** - server did not respond to our connection request, sometimes before we are dynamically blocked when our tests are detected
- **No route to host** - unable to reach the server
- **Unable to connect to server** - failed to connect to the server
- **Connection reset** - we got disconnected from the server
- **Unrecognized SSL message, plaintext connection?** - the server responded with plain-text HTTP on HTTPS port
- **Received fatal alert: handshake_failure** - this is either a faulty SSL server or some other server listening on port 443; if the SSL version of the web site works in your browser, please report this issue to us

SSL Report v1.12.8

Copyright © 2009-2015 Qualys, Inc. All Rights Reserved.

Terms and Conditions

Fig 8: Certificate Issue [56]



You are here: Home > Projects > SSL Server Test > virustotal.com

SSL Report: virustotal.com

Assessed on: Fri Jan 30 02:36:19 PST 2015 | HIDDEN | Clear cache

Scan Another >>

	Server	Domain(s)	Test time	Grade
1	216.239.36.21 any-in-2415.1e100.net Failed to obtain certificate	virustotal.com	Fri Jan 30 02:34:45 PST 2015 Duration: 4.480 sec	-
2	216.239.38.21 any-in-2815.1e100.net Failed to obtain certificate	virustotal.com	Fri Jan 30 02:34:50 PST 2015 Duration: 4.355 sec	-
3	216.239.32.21 any-in-2015.1e100.net Failed to obtain certificate	virustotal.com	Fri Jan 30 02:34:54 PST 2015 Duration: 4.517 sec	-
4	74.125.34.46 ghs-vip-any-c46.ghs-ssl.googlehosted.com Ready	www.virustotal.com	Fri Jan 30 02:34:59 PST 2015 Duration: 75.558 sec	B
5	216.239.34.21 any-in-2215.1e100.net Failed to obtain certificate	virustotal.com	Fri Jan 30 02:36:14 PST 2015 Duration: 4.405 sec	-

Warning: Inconsistent server configuration

SSL Report v1.12.8

Fig 9: Grading of IP [57]

B. Addressing False Negative Results:

False negative is the actual vulnerabilities which is present in web application and scanner fails to detect it. They occur because it fails to detect or find the links or because of scanner was not robust to detect it. QualysGuard WAS continually monitoring the feedback from the user to create new test statics as intended.

At Last QualysGuard WAS tool automatically scan IP of different ranges and it is the most authentic tool to reduce the vulnerabilities.

C. Proposed Model:

First User initiate the request if no modification is preformed by attacker then request of user will be processed. But if modification of url is preformed by attacker then our tool QualysGuard WAS will detect this and scanning of Url is preformed and regarding grading will be given and more our this tool will detect which vulnerability is this.

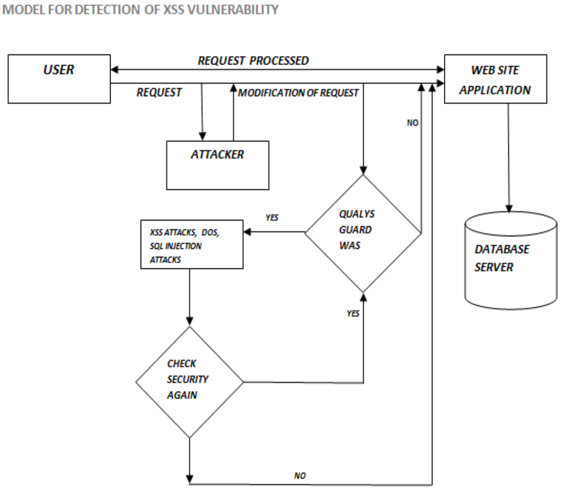


Fig11: Proposed Model

VI. Prevention Mechanism

Cross Site Scripting is a tedious problem and it's not going to be solved easily. The problems of two types

Browser is not much authentic by design. JavaScript is the standard language and attacker can perform both functions good as well as bad. Developers doesn't developed secured site. So filtering is used for blocking input sanitation. If data placed in an Email having malicious contents then only input sanitation can catch the invalid contents or the blocked content will be reflected back on page. The principal concept of input blocking would check which content is not matched to input or an error message is reflected back. Simply allow the content which is not malicious and block all the contents which is creating nuisance. HyperText Markup language encoded in both decimal and Hexa decimal. Simply example will work with IE 7.0

<IMG SRC=java&#xxA;script:alert('XSS')>

This code will not be authenticated because #XXA; is known string and will surely blocked by above filter. Filters had done good job but failed to sanitize the end vector to fire any modification to make XSS vector attack possible.

In input encoding all outcomes known in prior but do disadvantage as well regarding issues for making of large website design and there are some functions in html tags or inside of JavaScript allows attacker to modify the character set. Still many advantages of input coding like if you want to scale the performance of website like hit ratio you need to check every hit and then filter the request.

In output encoding outcomes are not known and it is completely different from other machine and often used because of life time reliability.

The security related problems to web application give rise to JavaScript popularity. Web worms such as samy spread widely and affecting trillion of victim worldwide. The injection of JavaScript into a document makes the input unlawful for the web browsers.

Out.println ("<p> hello"+name_victim" Welcome </p>") this code get vulnerable as this include input commands

<Script>
Steal Cookie ( )
</Script>

Stealing of cookie and document has no submission of unauthorized.

Input validation is a common type of defense of the XSS attack. In order to process untrusted input by filtering module it can be done. In filtering we can place constraints on input (Such as "Zip code" is exactly up to 5 characters). Filtering defense fails, when user input include rich html. Filtering has to make a particular sequence of character that may appear on a browser. Few browsers ignore "/" character other approaches to defend against XSS on server side, Dynamic tainting is to track the unauthentic information. Before giving an output to web application first we need to pass untrusted information to filters there are still some issues in filtering.

In this dissertation I studied large no. of recent and more realistic real world XSS problem failure of filtering tools and then we propose a new model for detecting XSS attacks on server side.

Malicious and special crafted inputs of the program leading into an html response type helps to occur XSS attacks. Web applications were written without having security module and developers barely pay attention while developing code. The idea is for discovering the shadow response. The purpose is to generate shadow response to intended set of constraints and submit authorized script as per response of http XSS Guard sets the real response of http and gives robust techniques for the real browser code. XSS Guard check that weather scripts is authorized or unauthorized. This is done as per the shadow response intended by script. XSS Guard removes the script and sends a response to client.

VII. Conclusions

This paper realized the problem of current web vulnerability. We called it as XSS vulnerabilities. Even vulnerability scanners were failed to detect all the vulnerabilities like XSS. One main reason of this problem which I observed during my research is developers barely pay attention while developing a web sites or applications. In my research work, I tried to find out security holes and limitations but vulnerabilities are such problems that cannot be removed completely. In my research work I used QualysGuard WAS to detect Cross Site Scripting malicious code which is the main only reason to attack security of client side as well as server side. Two main aspects for evaluating any tool are:

- Performance
- Accuracy

Performance depends on detection speed rate where as accuracy depends on quality detection time. QualysGuard WAS tool was tested against two scenarios of secured web applications and unsecure web applications. The detection rate of QualysGuard WAS tool is 98.24% and XSS exploits with 96.99% and even user needs were satisfied with XSS detection of Qualys Tool.

In addition performance also depends on execution time i.e. 139/sec, 189/sec and it varies with different module tests.

Few researches evaluate the injected malicious script with normal mode or advanced mode. Normal mode detection is about 33.6%. But advanced scanner with better accuracy was shown. But in QualysGuard WAS tool we had grading system which tells us which URL is safe or not to be run on system. But one thing during building a software process developers must be provided with training to find out security parameter must be considered before developing software. Security awareness must be provided to all stage of documentation.

The final conclusion is QualysGuard WAS tool can satisfy the client needs as well as server needs. But it won't cover all features unless user or client manually scans their URL of web sites before being used.

References

[1] Giovanni, Vigna, Fredrik, Valeur, Davide, Balzarotti, William, Robertson, Christopher, Kruegel, and Engin, Kirda. "Reducing Errors Anomaly Based Detection of Web Based Attacks through the Combined Analysis of Web Requests and SQL Queries", Journal of Computer Security,Vol.17, No.3 , pp. 305-329, 2009.

[2] Marco, Cova, Christopher, Kruegel and Giovanni, Vigna. "Detection and Analysis of Drive by Download Attacks and Malicious JavaScript Code", pp.281-290, 2010.

[3] Marian, Ventuneac, Tom, Coffey, Ioan, Salomie. "A Policy Based Security Frame work for Web Enabled Applications",Proceedings of the 1st International Symposium on Information and Communication Technologies,Trinity College, Dublin, pp. 487-492,2003.

[4] David, Scot, Richard, Sharp. "Developing Secure Web Applications.",IEEE Internet Computing, Vol.6 , No.6 , pp.38-45,2002.

[5] David, Scott, Richard, Sharp. "Abstracting Application Level Web Security" ,Proceedings of the 11th International Conference on World Wide Web, ACM, Hawaii, USA, pp.396-407, 2002.

[6] Garcia, AlfaroJ, Castillo,G.S Navarro Arribas G. Borrell,J."Mechanisms for Attack Protection on a Prevention Framework", Proceedings of the 39th Annual IEEE International Carnahan Conference on Security Technology, Spain, pp.137-140, 2005b.

[7] Frank S.Rietta."Application Layer intrusion Detection for SQL Injection", Proceedings of the 44th Annual Southeast Regional Conference (ACM-SE44), Florida, USA, pp.531-536, 2006.

[8] Fredrik, Valeur, Darren, Mutz and Giovanni, Vigna. "A Learning Based Approach to the Detection of SQL Attacks", Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), pp.123-140, 2005.

[9] Manar, H.Alalfi, James, R.Cordy, Thomas R.Dean. "A Verification Framework for Access Control in Dynamic Web Applications",Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering, USA, pp.109-113, 2009.

[10] Guillaume, H, Valerie Viet, T.T, Ludovic, M Benjamin, M." Policy Based intrusion Detection in Web Applications by Monitoring Java Information Flows", International Journal of Information and Computer Security, Vol.3, No.3, pp.265-279, 2009.

[11] Cook, W.R and Rai,S. "Safe Query Objects as Remotely Executable

Queries" ,Proceedings of the 27th International Conference on Software Engineering, pp.97-106, 2005.

[12] James, B.D.Joshi, Walid, G.Aref, Arif, G Eugene, H.S."Security Models for Web Based Applications", Communications of the ACM, Vol.44,No. 2, pp.38-44, 2001.

[13] Vidar, Kongsli."Towards Agile Security in Web Applications" ,proceedings of 21st ACMSIG PLAN Symposium on Object Oriented Programming Systems, Languages, and Applications, Portland, USA,pp.805-808, 2006.

[14] Viktoria, Felmetsger, Ludovico, Cavedon, Christopher, Kruegel, Giovanni, Vigna." Toward Automated Detection of Logic Vulnerabilities in Web Applications", Proceedings of the 19th USENIX Conference on Security, Berkeley, USA, pp.10-10, 2010.

[15] Nenad, Jovanovic, Christopher, Kruegel, Engin, Kirda."Precise Alias Analysis for Static Detection of Web Application Vulnerabilities", Proceedings of the 2006 Work shop on Programming Languages and Analysis for Security, Ontario, Canada, pp.27-36, 2006.

[16] Yao, Wen Huang, Fang, Yu, Christian, Hang, Chung Hung Tsai, DerT sai Lee, Sy Yen Kuo."Securing Web Application Code by Static Analysis and Runtime Protection" , Proceedings of the 13th International Conference on World Wide Web, New York, USA, pp.40-52, 2004.

[17] Gary, Wassermann, Zhendong, Su." Static Detection of Cross Site Scripting Vulnerabilities", Proceedings of the 30th International Conference on Software Engineering, Leipzig, Germany, pp.171-180, 2008.

[18] G.Wassermann and Z.Su, "Static detection of cross-site scripting vulnerabilities ," in 30th International Conference on Software Engineering, Leipzig, Germany, May 2008.

[19] D. Balzarotti, M. Cova,V.Felmetsger, N. Jovanovic, Kruegel, E.Kirda , and G.Vigna, "Saner Composing static and dynamic analysis to validates Sanitization in web applications," in IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2008.

[20] Sven, Lachmund." Auto generating Access Control Policies for Applications by Static Analysis with User Input Recognition", Proceedings of the 2010 ICSE Workshop on Software Engineering for SecureSystems, CapeTown, SouthAfrica, pp.8-14, 2010.

[21] Marian, Ventuneac, Tom, Coffey, Ioan, Salomie." A Policy Based Security Framework for Web Enabled Applications", Proceedings of the 1st International Symposium on Information and Communication Technologies, Trinity College, Dublin, pp. 487-492, 2003.

[22] Wei, Xu,Sandeep,Bakhtar, Eep, Bhatkar and Sekar,R. "A Unified Approach to Prevent Attacks Exploiting Range of Software Vulnerabilities",Technical Report SECLAB - 05-05. Department of Computer Science, Stony Brook University, 2005.

[23] William G J Halfond. And Alessandro, Orso. "Combining Static Analysis and Runtime Monitoring to Counter SQL Injection Attacks", Proceedings of the 3rd International Workshop on Dynamic Analysis (WODA2005), USA, pp.1-7, 2005a.

[24] William G.J Halfond. And Alessandro, Orso." Using Positive Tainting and Syntax Aware Evaluation to Counter SQL Injection Attacks", Proceedings of 14th ACMSIG SOFT International Symposium on Foundations of Software Engineering,pp. Portland, USA, pp.175-185, 2006.

[25] Yao, Wen Huang, Fang, Yu, Christian, Hang, Chung Hung Tsai, Der Tsai Lee, Sy Yen Kuo."Securing Web Application Code by Static Analysis and Runtime Protection",Proceedings of the 13th International Conference on World Wide Web, New York, USA, pp. 40-52, 2004.

[26] Doudalis, I Clause, J, Venkataramani,G,Prvulovic, M,Orso, A. "Effective and Efficient Memory Protection Using Dynamic Tainting" , IEEE Transactions on Computers, Vol.PP ,No.1, pp.1-15, 2007.

[27] Doudalis, I Clause, J, Venkataramani,G,Prvulovic, M,Orso, A. "Effective and Eifficient Memory Protection Using Dynamic Tainting" , IEEE Transactions on Computers, Vol.PP ,No.1, pp.1-15, 2007.

[28] E.Kirda, C.Kruegel, G.Vigna, and N.Jovanovic," Noxes: Aclient- side solution for mitigating cross-site scripting attacks," in 21st Annual ACM Symposium on Applied Computing, Dijon, France, Apr. 2006.

[29] P.Vogt, F.Nentwich, N. Jovanovic, E. Kirda, C.Kruegel, and G.Vigna," Cross-site scripting prevent I on with dynamic data tainting and static analysis," in 14th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, Feb.2007.

[30] D. Ross,"IE 8 XSS filter architec-ture/implementation," Aug.2008.[Online]. Available:http://blogs.technet.com/swi/archive/2008/08/19/ie-8-xss-filter-architecture - implementation.aspx

[31] G.Maone ,"No Script features: Anti-XSS protection." [Online]. Available: http:// noscript. Net /features#xss

[32] M.Johns, B.Engelmann, and J.Posegga," XSSDS: Server –side detection of cross-site scripting attacks," in 24th Annual Computer Security Applications Conference, Anaheim, CA, USA,Dec.2008.

[33] Sekar, " An efficient black-box technique for defeating web application attacks,"in 16th Annua lNetwork & Distributed System Security Symposium, San Diego, CA, USA, Feb. 2009

[34] E.Z.Yang, "HTMLPurifier."[Online]. Available: http ://htmlpurifier.org

[35] "PHP input filter," 2008. [Online]. Available:http: //www.phpclasses.org/browse/ package /2189.html

[36] The KSES project,"2008. [Online].Available: http://sourceforge.net /projects /kses

[37] Pietraszek, T. and Berghe, C.V. "Defending Against Injection Attack through Context Sensitive String Evaluation",In Recent Advances in Intrusion Detection, Vol.3858, Springer,pp.124-145,2006.

[38] Thorsten, Holz Simon and Marechal., Frederic Raynal ."NewT hreats and Attacks on the World Wide Web", IEEE Security and Privacy, Vol.4, No.2, pp.72-75, 2006.

[39] Martin, Johns, Bjorn, Engelmann, Joachim, Posegga. "XSSDS: Server Side Detection of Cross Site Scripting Attacks", Proceedings of the 24th Annual Computer Security Conference, ACSAC, IEEE Computer Society , USA, pp.335-344, 2008.

[40] Wurzinger, P.,Platzer, C.,Ludl, C Kirda, E Kruegel C." SWAP: Mitigating XSS Attacks Using a Reverse Proxy" ,Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems, Vancouver, Canada, pp.33-39, 2009.

[41] Prithvi Bisht. And Venkatakrishanan, V.N. "XSS Guard: Precise Dynamic Prevention of Cross Site Scripting Attacks", Proceedings of International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA'08) LNCS, France, Vol. 5137, pp.23-43,2008.

[42] P.Saxena, D.Song, and Y.Nadji, "Document structure in-tegrity: A robust basis for cross-site scripting defense," in 16th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, Feb. 2009.

[43] M.Van Gund y and H.Chen, "Noncespaces: Using random-ization to enforce information flow tracking and thwart cross-site scripting attacks," in 16th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, Feb.2009.

[44] Joon S Park, Ravi Sandhu, "Secure Cookies on the web ",IEEE internet computing . Volume 4. Pp. 36-44 July/August 2000.

[45] Christopher Krugel G Vigna willam Robertson, " A Multi Model Approach to Detection of Web Based Attacks", Computer Networks, Volume 48, Issue 5, pp 717-738, August 2005.

[46] D. Balzarotti, M. Cova,V.Felmetsger,N. Jovanovic,Kruegel, E.Kirda , and G.Vigna, "Saner :Composing static and dynamic analysis to validates Sanitization in web applications , " in IEEE Symposium on Security and Privacy,Oakland,CA,USA,May2008.

[47] Christopher Krugel G Vigna William Robertson, 2005 " A Multi Model Approach to Detection of Web Based Attacks", Computer Networks, Volume 48, Issue 5, pp 717-738, August 2005.

[48] N. Jovanovic ,C Kruegel and E Kirda, " Pixy: A Statics Analysis Tool for Detecting web Applications Security Vulnerabilities ", In Proceedings of 2006 IEEE Symposium on Security and Privacy (S&P'06),California, USA pp. 27-36 May 2006.

[49] D. Balzarotti, M. Cova, V.Felmetsger, N. Jovanovic,Kruegel, E.Kirda , and G.Vigna, "Saner Composing static and dynamic analysis to validates Sanitization in web applications,"in IEEE Symposium on Security and rivacy,Oakland,CA,USA,May2008.

[50] Yao Wen Chung Hung Tsung Po lin and Chung hang Tsai,2007, "Web Application Security Assessment By Fault Injection and Behavior Monitoring", In Proceedings of 12th international conference on World Wide Web, Budapest, Hungary. Pp 861-999.

WEBSITES

[51] [Aug,2013] acunetix website.[Online].Available: http://www.acunetix.com/blog/articles/blind-xss/
[52] [Sep 24, 2013] Qualys website.[Online].Available: https://www.qualys.com/qualysguard-subscription-plans/
[53] [Sep 24, 2013] Qualys website.[Online].Available: https://browsercheck.qualys.com/?euid=&ls=1
[54] [Sep 24, 2013] Qualys website.[Online].Available: https://www.ssllabs.com/ssltest/
[55] [Sep 24, 2013] Qualys website.[Online].Available: https://www.ssllabs.com/ssltest/analyze.html?d=linkis.com
[56] [Sep 24, 2013] Qualys website.[Online].Available: https://www.ssllabs.com/ssltest/analyze.html?d=openthesis. org
[57] [Sep 24, 2013] Qualys website.[Online].Available: https://www.ssllabs.com/ssltest/analyze.html?d=virustotal

.com&latest