Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.517 – 527

SURVEY ARTICLE

A Survey on Security Properties and Web Application Scanner

Mr. K.Naveen Durai[1], K.Priyadharsini[2]

[1]Department of Computer Science, Anna University (Chennai), India

[2]Department of Computer Science, Anna University (Chennai), India

[1] srmnaveen18@gmail.com
[2] ppriyakannan10@gmail.com

Abstract— Web application is one of the most powerful communication channel and service providers for information delivery over internet today. Open Web Application Security Project is the top 10 vulnerability list that resulted more number of attacks in the websites in the few years. The main objective is to find out the effectiveness in detecting the vulnerability in web application. The main motive of the scanner is to identify the vulnerability and produce a better result/report of each web application in effective manner. User can face with different types of challenges to give effective secure web applications. Root Cause Analysis is an in-depth process or technique for identifying the most basic factors underlying a variation in performance and find symptoms,security bugs (coding errors), and insecure-configuration. Here we mainly focus on working of web application and construction of secure web application will also discuss about the root cause of the problem and give better report.

Keywords—Web application, Web security scanners, Web security vulnerability, OWASP

1. INTRODUCTION

Web application is the most important, kinds of communication channels service providers and clients. On the internet vulnerability may compromise all the sensitive information and give report continuously which results in damage of cost. The main reason behind this is developers having limited programming skills and lack of security awareness. In this paper we mainly explained about broken authentication and session management. Web application vulnerability is present in the application firewall or intrusion detection system. Web page contain HTML, images, script code and become more user friendly but it also exploits security vulnerability. The Ten Most Critical Web Application Security Vulnerabilities are explained in OWASP.AJAX (Asynchronous java script and XML) it enhances web application and gives more interaction and response to the user. Web application stores the sensitive information such as financial and health, ethical and legal consequences and also to evaluate and identify the potential vulnerability for future research in this area. In web application consortium 49% of vulnerability has been reviewed of highly dangerous risk are identified as 13% are analysed completely. Web application scanner will reduce the vulnerabilities present in the web application. Web application scanner is an automated program that examines web application for vulnerability. Vulnerability present in the different stages of web application. In Web application malware threats are most discussed area of web application security. Vulnerability can be measured as "Weakness in information system". Web application mechanisms include a web browser that may interacts with one or more number of web servers via a series of HTTP requests and HTTP responses. Technology is increasing and security also increasing. Web application tools are nowadays more reliable, cost is also low and it is easy to use. Web scanners cannot find all the possible flaws in web application, we can reduce the vulnerability using scanners but we cannot fully eradicate. AI integrated scanners are used in real time scan application. Many numbers of URLs were to find & exploiting these vulnerability are able to retrieve the information from the database.

2. A SURVEY ON SECURITY AND VULNERABILITY OF WEB APPLICATION

Some of the vulnerabilities present in top 10 OWASP security web application. OWASP focussed on identify some vulnerability for the broad array of the organization [2]. In here more popular input validation attacks includes SQL injection and cross side scripting (XXS).

2.1 Working of web application

Web application enables the dynamic information and service delivery. Figure 1 shows the web application includes client side and server side components.

i) The client side components include static HTML pages with scripting languages are executed within the web browser.

ii) Server side request are processed by web server using dynamic HTML pages through execution part i.e. Java Servlet and the interpreter gives response to the client request. HTTP is a stateless protocol. Separate mechanism is used to maintain session state in web application.

iii) Session is a serious interaction between web application and user during the time of single visit to the website.

Web application is a gateway of database that holds a critical vulnerability application and asserts. Most web application store the information in databases or in file system bases.

2.2 Requirements for web application scanner

A web application scanner must have the following qualities:

- Identify the vulnerabilities in web application.
- Generate report/result for vulnerability.
- Low amount of errors must be identified.

2.3 SQL Injection

It is an injection attack by an attacker i.e. a code is injected to retrieve the information from the database. Attacker sends a user input through SQL queries. SQL injection results in data loss, denial of service and authentication bypass etc. Avoiding using the connection of database, avoid dynamic SQL queries and use some hash function to store the confidential information.

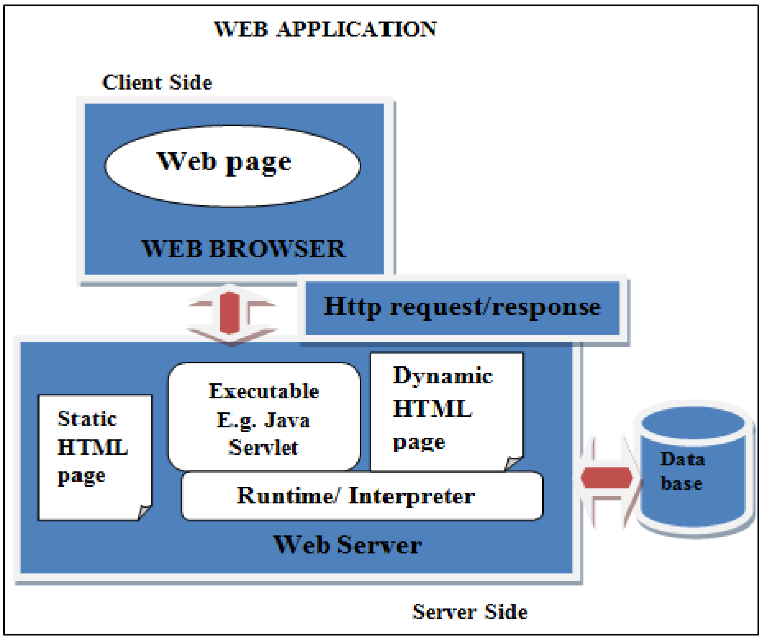

Fig 1: Diagram For Working Of Web Application.

2.4 Cross site scripting

This is also called as XSS, this flow occur during the processing of application. User gives the data from web pages without any proper validation. Effects of XSS may results in session hijacking, sensitive information and site trusting. Three types of XSS are based on how a malicious code has been injected Stored XSS, reflected XSS and DOM -based XSS.

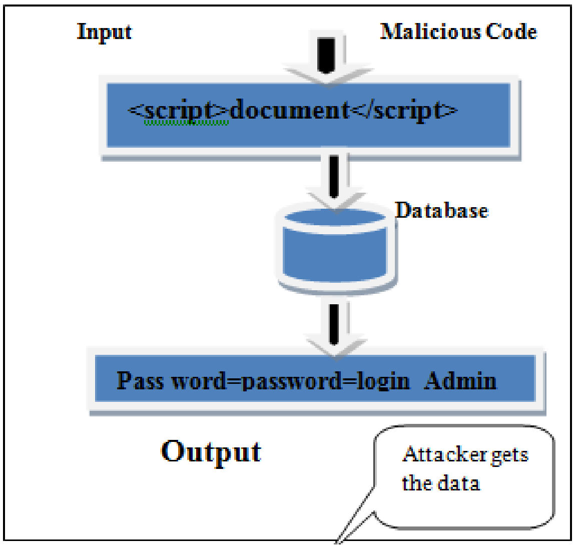

Fig: 2 XSS In Vulnerability

2.5 Broken authentication and session management

In broken authentication and session management, functions are not implemented correctly. Some of the key points are log out functionality, password management, time out, secret question and account update etc. Attacker uses this broken authentication and session management functionality. E.g. Session IDs, mail account and password etc. A loophole from authentication and session management attacker steals the user information such as user credentials and session IDs.

Strong functions are implemented for broken authentication and session management are functions for request time out, user account log in-out and allows secure http cookies. Highly sensitive information travels in different format. Avoid URLs and session IDs and follows some security standards for broken authentication and session management. Using some vulnerability we have composed large number of vulnerable components such as library files and application frameworks. Vulnerability exploits causes data loss. Web application–malware threats are most discussed area of web application security. In past 5years many attacks happens on the bank websites, private organizations and some other websites include Google’s blacklisting. Websites owner mainly concentrates on their profit and mainly focus for improving the experience and customer, popularity of the websites, social network websites and support all mobile application and used in android application. In the recent year of 2011,286 malware variants were identified.

3. A SURVEY ON WEB APPLICATION SECURITY

There are three aspects in web application development, which posses the challenges for developing a secure web application.

It has 3 levels of security properties.

- Input Validation
- State Integrity
- Logic Correctness

Understand the failure and find the root cause of corresponding vulnerability. Here the existing research works are classified into security by consortium, security by verification and security by protection. The recent reports level that over 80% of web sites on the internet have atleast one serious vulnerability. Here the logic implementation is a key to the functioning of web application.

Logic flaws are vulnerable to class of attack. It is mainly known as logic attack or state violation attack. Missing and failure state are checked and introduced to logic vulnerability into web application. Here first property is hiding technique; it follows the principle of security and obscurity. It follows the attacker to recover the hidden links and directly access the unauthorized information and affect data flow in web application. Here 49% of web application consortium has been reviewed and it contains vulnerabilities. Some of the web programming languages are PHP, JavaScript etc.

3.1 State maintenance

It is a basic building for state full web application, which requires a secure web application to preserve the integrity. State maintenance makes the assurance of state integrity of challenge issues for web application. Countermeasures for developing the secure web application has 2 dimensions

3.2 Security by construction

The main aim of this method is to build a secure web application, ensure that has no potential vulnerabilities exist in the web application. To solve the problem to identify the root and it must detect the vulnerabilities correctly.

3.3 Security by verification

This technique to verify the security property and identify the vulnerabilities present in the application. This technique is defined as vulnerability analysis. Dangerous risk level are identified more than 13% of websites are automatically completed. Web application maintains both a large number of persistent states in the database. First is the security property. Second dimensions give the outline of 3 classes.

Web application development features are logical implementation, state maintenance and programming languages.

3.4 Security by protection

The information flow must be specified in 3 tasks.

- Identification
- Tracking
- Handling

The main aim is to protect a vulnerability web application against exploits by constructing the runtime environment that supports secure execution.

3.5 Program analysis

In this analysis we have 3 techniques such as: static, dynamic, hybrid analysis. Static has several techniques dataflow string and pointer analysis and it gives the details about insecure information flows. Dynamic it tracks the dataflow during the runtime it may affect the stability and performance. Hybrid is the combination of both static and dynamic analysis. Scanner cannot identify the vulnerability clearly. Scanners are detecting small number of vulnerability for testing small application. Scanners are not constantly top-ranked across all vulnerable application. Here they are mainly discussing about black-box scanners whose prices ranges from dollars. In this paper they describe various types of scanner usage scenarios. In some test bed experiments user set our own default setting as automatic mode. The testing result/report of the scanner is day by day increasing. Here we get the 4 test vector statics of scanner (MCAfee, IBM, HP, Acunetic).

3.6 Session management

Session management function includes some session management flaws and secures authentication flaws. The testbed authentication includes some credentials send through unencrypted HTTP, and sensitive information over weak password, password recovery, remember me password. Cookie vulnerability includes insecure session and Http Only cookies, session fixation, sensitive content. Vulnerability detection rate reported in this paper is less than 50%, the scanner performance is better at detecting this vulnerability class SQLI, XSS and XCS.

3.7 Session Management Capabilities

These are the capabilities of session management

- Start a new session
- Session token refresh
- Session expired
- Reacquire session tokens

Session Management vulnerabilities were detected in 79% of applications tested in 2013, more than any other application vulnerability class. 96% of applications have vulnerabilities with a median of 14 per application. These flaws can lead top Hijacking of user or administrative accounts, undermine authorization and accountability controls and cause privacy violations. OWASP is a new kind of organization. Our freedom from commercial pressures allows us to provide unbiased, practical, cost-effective information about application security. The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security.

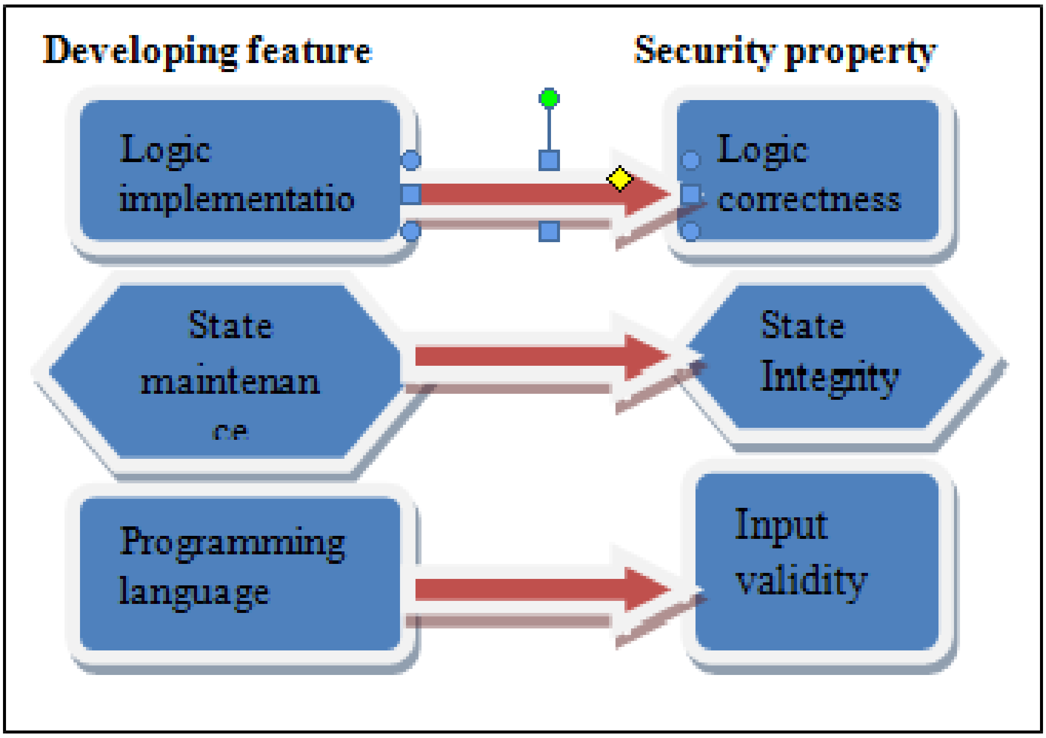

Fig 3: security properties and web application

4. SECURE WEB APPLICATION USING BROKEN AUTHENTICATION AND SESSION MANAGEMENT, CROSS SITE REQUEST FORGERY & SCRIPTING ATTACK AND SQL INJECTION

In this paper we mainly discuss about broken authentication and session management, Cross site Scripting Forgery & Scripting Attack and SQL Injection. In the first part broken authentication and session management, it evolves trial and error method, guessing a person username and password, PAN number, credit card number and so on. An attacker can check the password word by word and generate thousands of guesses. Attacker guesses the password, then it is ok means attacker can use that account. It is mainly used for managing user secure authentication and handling sessions. Session management protocol is a session layer for delay tolerance is developed at Ericsson research centre.

4.1 Cross site request Forgery

Here we discuss about hijacking and other attacks are complex. So we mount this attack. Someone inject their malicious JavaScript code into an application. A site has a comments section on it that allows people to post comments. Comments are stored in the database when posted and then displayed on the page for everyone to see. Session hijacking is the exploitation of a valid session. A CSRF attack forces a hacker browser to send a forged HTTP request, the hacker session cookie are included authentication information are vulnerable to web application. This allows the attacker to force the hacker to generate requests from the victim. Method of attack-Logout, Password Management, Timeout, Remember me, Secret question and account update. Session fixation attacks attempt to exploit the vulnerability of a system which allows one person to set another person's session identifier (SID). Session fixation is only web based attack.

4.2 SQL Injection

SQL Injection is an attack thread to confidentiality, integrity, and availability on web application. In some cases an attacker can use SQL Injection techniques to backdoor the Web application or execute operating system commands. SQL structures database queries language. SQL is a language implemented by numerous database systems, including MySQL, PostgreSQL, and SQLite. To build dynamic SQL queries and we have alter the semantic structure of the query.

TABLE I
THIS TABLE SHOWS THE DIFFERENT TECHNIQUES ANALYSED IN THIS PAPER

S. No	Methods	Usage
1	ARDILLA	Inputs through the execution, it has second-order Vulnerability.
2	FLAX	Black box fuzzy technique, to find the Client side input validation vulnerability
3	AMNESIA	Models the structure of legitimate SQL queries
4	CANDID	Uses dynamic Techniques to extract More accurate Structure
5	RoleCast	It is used for finding Access control check

4.3 Cross Site Scripting Attacks

Cross site Scripting attack is referred to as XSS. When an attacker uses an application to send malicious code, in the form of a script, to the end user. It is a Wide spread problem. **Stored-**Attacks injected code is permanently stored on the target web server (database, visitor log, comment field). Reflected attacks are delivered via another route such as an e-mail message. A user is then tricked into clicking on the malicious link or submitting data. The browser then executes the code from what is considered a trusted server.

All web servers, application servers, and web application environments are susceptible to cross site scripting. XSS Flaws can be difficult to identify and remove from a web application. The best way to find flaws is to perform a security review of the code and search for all places where input from an HTTP request could possibly make its way into the HTML output. Validation of all headers, cookies, queries strings, from fields, and hidden fields against a rigorous specification.

5. WEB APPLICATION SCANNERS: A REVIEW OF RELATED ARTICLES

In web application scanner we discuss about 3 articles mainly first article gives the detailed overview of security purpose of web scanners and it gives basic idea of what the scanner should do. Then Article 2 describes about automated scanning and manual testing. Article 3gives in-depth of penetration testing and gives the overview of web scanner. Security tools are very important but it is not secure up to 100% and it is used for current and future use, but it is not enough new vulnerabilities are grown day by day.

Source code scanners are used to identify the flaws and vulnerability and they are code line by line in the web application. It takes time to check each of the web applications. The main comparisons between the 3 articles are given below. Three articles give the common message that automated scanning is a good way to find the vulnerability but it is not a correct solution to secure web application. No standard version is followed for developing web scanners and there is also no rule for testing the scanners. For testing a false vulnerability, we have to compare one or more web application. By comparing each result/report we identify which is the false vulnerability and which is good.

TABLE II
COMPARISONS BETWEEN 3 ARTICLES

Article 1	-Describes about security vulnerability and how it is mainly used for test the special type of vulnerability.
Article 2	-Describes about different real-world scanners are tested, and to find the ability of test cases functionality for designing the web scanners. -It mainly focuses on which scanner is to find the vulnerability easily and gives the test report.
Article 3	-Concentrate on penetration testing and gives basic knowledge of what the web scanner do.

6. STATE OF THE ART: AUTOMATED BLACK BOX WEB APPLICATION VULNERABILITY TESTING

The study of current automated black box testing web application vulnerability scanner to identify the potential value in the research. There are three main researches are given below

i) What types of vulnerability are detected by the web scanners?
ii) How the scanners are effective in scanning for vulnerability?
iii) How representatives the scanner are test for vulnerability in the wild?

The scanner having low detection may be used in smaller number of vulnerabilities it detects individually more important to the customers. Here they discussed more vulnerability scanners and how they are effective in describing well-known vulnerability.

6.1 Scanner result on custom Testbed

Here we discusses general scenario about software architecture of black box vulnerability scanners. Vulnerability aim is to detect the test vector scanner. Scanner prises ranges from thousands to dollars. Session vulnerability includes session management flaws as authentication and cookies. Authentication vulnerability enable auto-complete in password fields. Figure 3 shows different scanner test vector scenarios.

6.2 scanner results on common web application

Vulnerability scanners are grouped and correlated with vulnerability population in the wild, now we have to find the vulnerability using scanners. Drupal, PHPBB2, and wordpress are the three web application with known vulnerability. Backup source code are accessible and path disclosure present.

6.3 Detection of malware

Detect all the malware threads in the testbed with an open source code. Mainly we designed for 2 potential loophole “traps” for false positive in testbed. First uses java script alert(). Second involves the right hand side of the assignment within the <script>. The alert() cannot find false positive and begin with the <script>. There are 90 total number of vulnerability in the testbed. Some of the scanner profiles used for testing the vulnerability are Acuneitx uses default and stored XSS, Cenzic uses best practices, PCI and session architectures, HP uses all checks, IBM uses complete scanning profiles, McAfee uses hack simulation and DoS.

7. CAPTCHA BASED WEB SECURITY: AN OVERVIEW

CAPTCHA (Completely Automatic Public Turing Test to Tell Computer and Human Apart). CAPTCHA helps to identify some artificial intelligent automated programs known as bots. Web-bots are threat to web services. Its methods are based on Artificial Intelligence (AI).

TABLE III
OCR-BASED CAPTCHA METHODS

Text-based CAPTCHA	Ability of people to read images of text more reliably than Optical Character Recognition (OCR)	Becoming more difficult for genuine users
Gimpy method	uses its word from a dictionary with 850 words	It can easily be broken
Pessimal Print Method	Prevent the operations of destructive computer software by artificial lowering the quality of the printed letters.	Mori-Malik algorithms and brute-force.
Baffletext Method	Words that are not provided in English dictionaries are produced.	Changed with different degrees is ease or difficulty.

CAPTCHA is also referred as Reverse Turing test. It has following 2 specifications:

- Judge is a machine instead of a human.
- Goal is virtually all human users will be recognize and pass these test.

If a problem cannot be solved by computer, CAPTCHAs is used. Text-based CAPTCHAs are not safe for computer techniques. OCR-base CAPTCHAs are text-based CAPTCHAs and the user shows distorted images, letters and digits and it has a drawback. CAPTCHAs are problem for mobile phones and palmtops, as the use of keyboard is difficult.

7.1 NON-OCR-BASED CAPTCHA

Non-OCR based CAPTCHAs are used to test the audio/video capability of humans.

- Implicit CAPTCHA
- Audio CAPTCHA
- Video CAPTCHA

7.2 CAPTCHA for web security

Critical vulnerabilities in various texts based CAPTCHA are identified from security. CAPTCHA implementations can employ Global Unique Identifier (GUID).

8. VULNERABILITY ASSESSMENT OF IPV6 WEBSITES TO SQL INJECTION AND OTHER APPLICATION LEVEL ATTACKS

The utilization of keyword search are mainly available in search engine such as Google, Yahoo are connected with web crawler and describe about black box penetration testing. Web crawler overview- Crawler is a computer system that browsers in World Wide Web. It is otherwise called as internet boat. Crawler validates the hyperlink and HTML code 4 key elements. In Selection policy the links has been previously visited web page. In reservation policy links are refreshed in order to detect changes. In politeness policy server is not overloaded with request. Parallelization scheme is the process that parallelization for efficient searching. Crawling is a set of hyper links of a website is given as an input for scanning purpose.

8.1 Web Crawler Working

a) Starts with a list of URL seeds.
b) Visited URL's are referred as hyperlink.
c) Add of URL to be visited – crawl frontier.
d) URL from the frontier is recursively visited according to the set of policies.
e) Visited URL is stored in the **live web.**
f) Stored URL is in live web is like a **snap shots**.

Crawler features are Distributed, Scalable, Performance and efficiency, Quality, Extendible.Test cases may be defined as the plan to test against vulnerability. Good scanner has two main things to be followed. First it should avoid reporting false positive, second need to test against many test cases. False positive are difficult to handle fewer scanner report for vulnerability and give access to the private folders and web servers. Dynamic analysis is to identify the security problem and interact with the functionality of the websites. Static mining module runs on depth mining of the website. Other functionality is to identify the e-mail information, broken links contain private information and it controls the related web pages by using breadth first search (BFS). Dynamic scanning module are used in many search engine, we find billions of web pages and URLs. This system can find all the related functionality of web pages and inspect vulnerability risk is present or not.

9. DESIGN AND EVALUATION OF A REAL-TIME URL SPAM FILTERING SERVICE

The extensive research of spam filtering is to protect from other web services. Here Monarch, a real time system crawls the URL & they are submitted to web services & it directs to spam message. Monarch can give an accurate result & give real time protection & they cannot generalize across web sites. In recent years internet plays a major role in web services including social networking, video sharing and the customer can share millions of viewers. Monarch is a real time system that crawls the URL from the web site and it directs the spam content. Monarch consist of three main elements front end accepts the URL submitted by the web services, & many of the browser that host the URL & to extract the content. It has some basic fundamental functionality arise from spam content. Monarch has a feature collection it gives millions of URLs are collected from email & Twitter .Monarch deploys the implementation to demonstrate accuracy, scalability & classify the tweet & spam content. The system flow can be an internal system flow, the URL is collected & they are associated with raw data that can be include the content page, behaviour and they have some hosting architecture**.**

Feature selection-The system visit the URL with the help of Firefox web browser and collect all the content of the pages include HTML pages, behaviour of the web page and manage all the windows activity. Single monitor manages many numbers of copies to aggregate results & restart the processes that has been previously visited. A web browser gives the basic idea for collecting features for classification. Some of the source web pages are initial and landing URL, redirects, source URLs, Frame URLs, HTML content, page links, java script events, pop-up windows, HTTP headers, DNS, Geolocation and Routing data are some of the source data in the web content.

10. WEB SERVICES THREATS, VULNERABILITIES, AND COUNTERMEASURES

Secure web application has some basic components in the client application, it intracts with the SOAP processor, XML parser &HTTP processor in the client side. Web application contains SOAP processor, XML parser and web server. Web server is connected to the HTTP processor. This task is very difficult, because vulnerability present in the web application is too large. Security architecture is designed by various requirment analysis and security planning are maintained. Here they discussed about threat, vulnerability, attacks and countermeasures.threat is an organizational asserts or individuals through an information system through an authorized access. Vulnerability is an weakness in the informatiuon system, it is defines as the flaw in a software component. Countermeasure is risk among the vulnerable application.

TABLE IV

VULNERABILITIES APPLICATION CATEGORIES

S.NO	APPLICATION VULNERABILITY	DESCRIPTION
1	Input validation	Controls all the web application should perform on its input: correctly.
2	Configuration Management	Web application uses its own data, are they stored and retrieved.
3	Authentication	Web application properly authenticate the sender input of the user
4	Session Management	Sessions are properly Protected here.
5	Exception management	Web application properly manage error information and reported to the user to avoid leakage information
6	Sensitive Data	Application use this techniques to protect confidentiality and integrity of data

11. WEB SERVER SECURITY AND SURVEY ON WEB APPLICATION SECURITY

Comparative Analysis and Suggestions between SQLIA, CSRF, XSS are discussed in this paper. Threats to Web Server and Countermeasures are discussed mainly. Some of the guide lines for developers are

- Use Repeatable Security Processes and Standard Security Controls
- Application Security Requirements
- Application Security Architecture
- Standard Security Controls

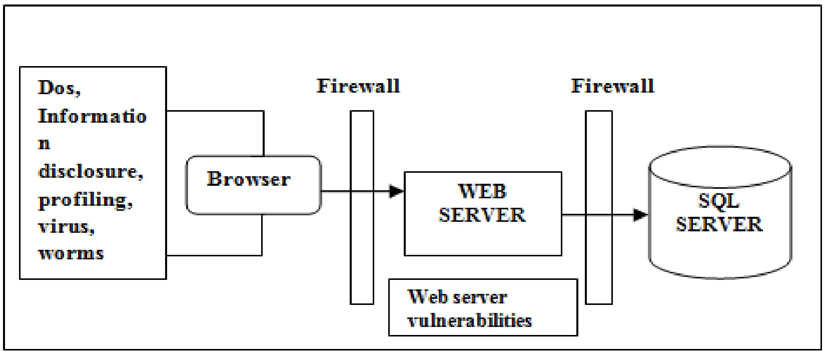
Dos, Informatio n disclosure, profiling, virus, worms
Firewall
Firewall
Browser
WEB SERVER
SQL SERVER
Web server vulnerabilities

Fig 4: web server threats

This figure 4 shows the various web server threads and some od the web server vulnerabilities and how they are working many attacks and threat the ever-growing numbers of attackers.

CONCLUSION

This paper surveys about different web application scanner and some important security properties of web application. Reports from different scanners produce different test results/report which reduces the vulnerability of web application. A good scanner should find as many numbers of vulnerabilities as possible. The main problem with Web scanners is that most of them give false positives results. A good scanner should avoid reporting false positives as much as possible. A single vulnerability opens a door for many attacks. False positives are hard to resolve so testers have to waste a lot of time to test manually and identify the vulnerabilities. We conclude a scanner is really good; we need to test the scanners against many test cases. Test cases are like users plan that testing a scanner against a vulnerable application. Automated scanning is to find vulnerabilities but it is not a correct solution for finding the vulnerability. Web servers, application servers, and web application environments are susceptible to broken authentication and session management.

REFERENCES

[1] Gopal R. Chaudhari, Prof. Madhav V. Vaidya"A Survey on Security and Vulnerabilities of Web Application ", Department of Information Technology, SGGS IE & T, Nanded, Maharashtra, India-431606.

[2] Xiaowei Li and Yuan Xue" A Survey on Web Application Security", Department of Electrical Engineering and Computer Science Vanderbilt University xiaowei.li, yuan.xue@vanderbilt.edu

[3] VSRD-IJCSIT, Vol. 2 (4), 2012, 356-364" Securing Web Application Using Broken Authentication & Session Management, Cross Site Request Forgery & Scripting Attacks and SQL Injection", Dinesh Chandra Misra, PankajAgrawal and Amit Kr. Srivastava.

[4]Web application scanners: A review of related articles(Elizabeth Fong and VadimOkun), Web application scanners: A review of related articles(Andreas Wiegenstein, FrederikWeidemann, Markus Schumacher, and Sebastian Schinzel), Web application scanners: A review of related articles(Gary Mc-Graw)

[5]Jason Bau, ElieBursztein, Divij Gupta, "State of the Art: Automated Black-BoxWeb Application Vulnerability Testing", John Mitchell Stanford University Stanford, CA {jbau,divijg}@stanford.edu, {elie,mitchell}@cs.stanford.edu

[6]Volume 3, Issue 11, November 2013 ISSN: 2277 128X , International Journal of Advanced Research in Computer Science and Software Engineering Research Paper Available online at: www.ijarcsse.com.'CAPTCHA Based Web Security: An Overview',SushamaKulkarni* Dr. H. S. Fadewar *,Department of Computational Science, S. R. T. M. University, Nanded, Maharashtra, India.*

[7] *Research Article'*Vulnerability Assessment of IPv6 Websites to SQL Injection andOther Application Level Attacks*',* Ying-Chiang Cho and Jen-Yi Pan *Department of Electrical Engineering, National Chung Cheng University, Chia-Yi 62102, Taiwan*silvergun@mail2000.com.twReceived 14 October 2013; Accepted 2 December 2013,Academic Editors: S. K. Bhatia and A. K. Misra.

[8]Design and Evaluation of a Real-Time URL Spam Filtering Service Kurt Thomas*, Chris Grier*y, Justin Ma*, Vern Paxson*y, Dawn Song* fkthomas, grier, jtma, vern, dawnsongg@cs.berkeley.edu* University of California, Berkeley y International Computer Science Institute.

[9]Web Services Threats, Vulnerabilities, and Countermeasures E. Bertino et al., *Security for Web Services and Service-Oriented Architectures*,DOI 10.1007/978-3-540-87742-4 3, c_ Springer-Verlag Berlin Heidelberg 2010

[10]Web Server Security and Survey on Web Application Security ShaikhBushraAlmin, *Department of Information Technology, PIIT, New Panvel. University of Mumbai, India.* skbushra78691@gmail.com

[11]Web Application Vulnerabilities and In-secure Software Root Causes: The OWASP Top 10, Cincinnati Chapter Meeting February 26th, 2008Marco.Morana@OWASP.ORG

[12]Antunes, N.; Vieira, M. "Detecting SQL Injection Vulnerabilities in Web Services". Fourth Latin-American Symposium on Dependable Computing, pp.17-24, 1-4 Sept. 2009.

[13]Kals, S; Kirda, E; Kruegel, C.; Jovanovic, N. SecuBat: a web vulnerability scanner. In Proceedings of the 15th international conference on World Wide Web. ACM, NY, USA, 247-256, 2006.

[14]Teodoro, N.; Serrao, C. "Web application security: Improving critical web-based applications quality through in-depth security analysis". International Conference on Information Society (i-Society), pp.457-462, 27-29 June 2011.

[15]Vieira, M.; Antunes, N.; Madeira, H. "Using web security scanners to detect vulnerabilities in web services". IEEE/IFIP International Conference on Dependable Systems & Networks,pp.566-571, 2009.

[16]ACUNETIX. Website Security - Acunetix Web Security Scanner. URL: http://www.acunetix.com.

Últimoacessoemfevereiro de 2012.

[17]OWASP Foundation: SQL Injection, http://www.owasp.org/index.php/SQL_injection

[18] OWASP Foundation: http://www.owasp.org

[19]Fear the EAR: Discovering and Mitigating Execution After Redirect Vulnerabilities Adam Doupé, Bryce Boe, Christopher Kruegel, and Giovanni VignaUniversity of California, Santa Barbara{adoupe, bboe, chris, vigna}@cs.ucsb.edu.

[20] XIAOWEI LI" A Survey on Server-sideApproachesto Securing Web Applications"Vanderbilt University YUAN XUE Vanderbilt University ACM Transactions on Computing Surveys, Vol. V, No. N, November 2013, Pages 1–0??.