Web Application Vulnerabilities and Detection

Kangan(Student)[1] **and Monika(Assistant Professor)**[2]
[1]U.I.E.T, Panjab University, Chandigarh, U.T, India
[2] U.I.E.T, Panjab University, Chandigarh, U.T, India

Abstract - *Web applications cover a range of activities, such as e-banking, webmail, online shopping, community websites, blogs, vlogs, network monitoring and bulletin boards. Web security testing verifies whether Web based applications are vulnerable or secure when they are subjected to malicious input data. This paper presents taxonomy of various security testing techniques and mainstream software tools used for a particular type of security testing. We also provide the brief description of various types of attacks and the security testing tools used to detect these attacks.*

Keywords: web application security; security testing tools; types of security testing; security risks;

1 Introduction

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results [1]. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The purpose of testing can be quality assurance, verification and validation, or reliability estimation [2]. There is a plethora of testing techniques [3] as shown in figure 1.

Today, software becomes more complicated and large-scale, which results in more software security problems increasingly. Software security is the ability of software to provide required function when it is attacked [4]. Security testing of any developed system is all about finding out all the potential loopholes and weaknesses of the system, which might result into loss/theft of highly sensitive information or destruction of the system by an intruder/outsider. Security Testing helps in finding out all the possible vulnerabilities of the system and help developers in fixing those problems.

Now a day, almost all organizations across the world are equipped with hundreds of computers connected to each other through intranets and various types of LANs inside the organization itself and through Internet with the outer world and are also equipped with data storage & handling devices. The information that is stored in these storage devices and the applications that run on the computers are highly important to the organization from the business, security and survival point of view.

Any organization small or big in size, need to secure the information it possesses and the applications it uses in order to protect its customer's information safe and suppress any possible loss of its business.

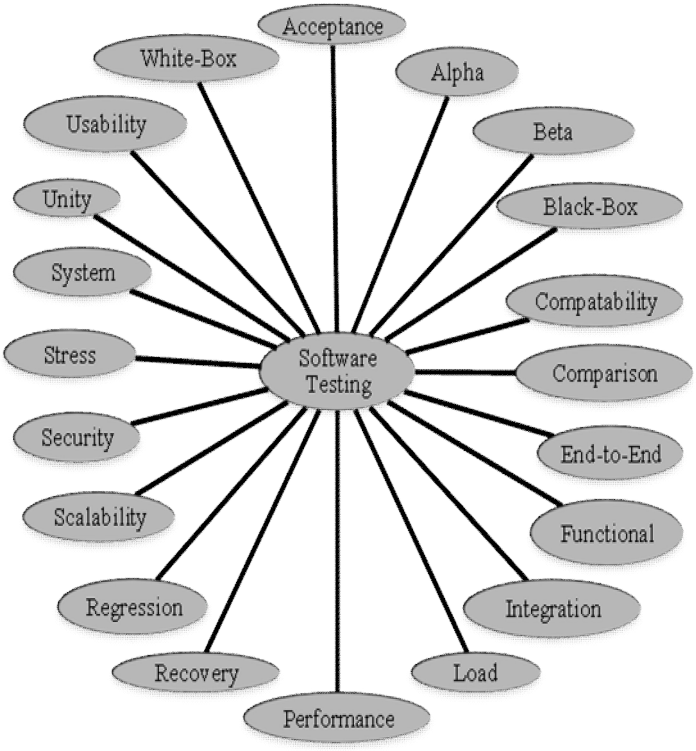

Figure 1. Types of Security Testing

Security testing ensures that the systems and applications used by the organizations are secure and not vulnerable to any type of attack. Finding and fixing security flaws in a legacy web application typically requires detailed knowledge of its behavior. This knowledge is a result of understanding high-level design artifacts combined with an analysis of the source code of the web application. However, it is well known that manual effort spent towards analysis of the source code is labor and cost-intensive and is often error-prone.

Additionally, design level artifacts are often unavailable for legacy web applications and the only available resource is the source code. While source code is the most accurate

description of the behavior of a web application, this description is expressed in low-level program statements. Due to its inherent low-level nature, source code does not readily over a high-level understanding of an application's intended behavior which is necessary to identify and fix security flaws. So, there arises the need for security testing of web application.

Software testing is the process that determines that confidential data stays confidential and users can perform only those tasks that they are authorized to perform.

In the next section we have describes the needs of security testing. Section III presents the different software for various types of security testing. Section IV presents various security risks. Section V provides the security testing tool for each type of security risk. Section VI concludes by highlighting research opportunities resultant from this work.

2 Needs of Security Testing

Exposing systems to the internet increases the risk that security weaknesses in those systems will be leveraged to compromise the system or the underlying data. So, there arises the need for security testing of web application. The various needs of security testing are [10]:

- Security test helps in finding out loopholes that can cause loss of important information and allow any intruder enter into the systems.
- Security Testing helps in improving the current system and also helps in ensuring that the system will work for longer time .
- Security Testing doesn't only include conformance of resistance of the systems your organization uses, it also ensures that people in your organization understand and obey security policies. Hence adding up to the organization-wide security.
- If involved right from the first phase of system development life cycle, security testing can help in eliminating the flaws. This is beneficial to the organization almost in all aspects (financially, security and even efforts point of view).

3 Needs of Security Testing

The purpose of security testing is to identify the vulnerabilities and subsequently repairing them. It ensures that the systems and applications used by the organizations are secure and not vulnerable to any type of attack. Typically, security testing is conducted after the system has been developed, installed and is operational. Security testing can be further classified into various types as shown in table I [10]. It presents the types of security testing, faults detected by them and various tools used for each type of security testing.

TABLE I. TYPES OF SECURITY TESTING

Type of Security Testing	**Fault Detected**	**Mainstream Commercial Software Tools used**
Security Scanning	•Detect areas of vulnerability in the OS, applications and network	Nessus (software), ISS
Network Scanning	•Detect active devices • Detect open ports and associated services/ application	Amap, AutoScan, Netdiscover, Nmap, P0f, Umit etc
Vulnerability Scanning	•Detects hosts and open ports •Detect known vulnerabilities	Firewalk, GFI LANguard, Hydra, Metasploit, Nmap, Paros Proxy etc
Password Cracking	• Detect weak passwords and password policies	Hydra, John the Ripper, Rcrack, SIPcrack, SIPdump, TFTPBrute, THC etc
Log Review	•Provides historical information on system use, configuration, and modification •Could reveal potential problems and policy deviations	Snort IDS sensor
File Integrity Checkers	•Detect changes to important files; •Detect certain forms of unwanted files, such as well-known attacker tools	Autopsy, Foremost, RootkitHunter, and Sleuthki
Anti-Virus and Malicious Code Detection	•Prevent attacks	avast! Free Antivirus, Veracode
Penetration Testing	• Tests security using the same methodologies and tools that attackers employ •Verifies vulnerabilities	Driftnet, Dsniff, Ettercap, Kismet, Metasploit, Nmap, Ntop, SinFP, SMB Sniffer, and Wireshar
Modem Security/ war dialing	• Detect the use of unauthorized modems that might be used to bypass existing security measures.	Toneloc (freeware), THC-SCAN (freeware), SecureLogix Telesweep Secure (commercial)
Risk Assessment	•Find out and prepare possible backup-plan for any type of potential risk	CRAMM, CORA, COBRA, Risk Check, RiskPAC,
Ethical Hacking	•Detect potential security weaknesses for a client	Cain and abel, Legion, Brutus,Ec-Council
Security Auditing	•Detect common security /config errors	LSAT, Flawfinder, RATS

4 Security Risks

In recent years the development of such applications has been considerable, and today rich internet applications offer complex, real-time interactions with users. For instance, web operating systems such as eyeOS offer much functionality that was previously available only with traditional operating

systems. While web applications have become ubiquitous, they also present new security risks. It is important to identify and understand these risks when developing, hosting or simply using these applications [5]. There are two main reasons that web applications are vulnerable to attack.

First, it is generally difficult for the service manager to keep up to date with security patches. This is a common issue for services in general, but it may be particularly challenging for web applications. This could be improved by better design and packaging but it is often impossible to upgrade web applications automatically.

Second, web applications are often easy targets for attackers. As a relatively recent development, they use non-mature code compared to traditional network services. Unfortunately exploits – malicious code that exploits software vulnerabilities –are generally easy to prepare, remotely executable, cross-platform, and require no compilation. This helps attackers to design effective and scalable automated attacks. Vulnerable installations can be found quickly, easily and silently by using search engines to detect known vulnerable patterns, generally filenames, of specific web applications. In table 2 we focused on identifying the most common vulnerabilities.

In our study we get details for each of the Open Web Application Security Project Top Ten 2011 [6] vulnerability. Table II shows the variations in top 10 security risks from 2007 to 2011.

TABLE II. TOP 10 SECURITY RISKS

Risk Level	Top 10 Security Risks in	
	2007	Till 2011
1	Cross Site Scripting (XSS)	Injection
2	Injection Flaws	Cross-Site Scripting (XSS)
3	Malicious File Execution	Broken Authentication and Session Management
4	Insecure Direct Object Reference	Insecure Direct Object References
5	Cross Site Request Forgery (CSRF)	Cross-Site Request Forgery (CSRF)
6	Information Leakage and Improper Error Handling	Security Misconfiguration(NEW)
7	Broken Authentication and Session Management	Insecure Cryptographic Storage
8	Insecure Cryptographic Storage	Failure to Restrict URL Access
9	Insecure Communications	Insufficient Transport Layer Protection
10	Failure to Restrict URL Access	UnvalidatedRedirects and Forwards (NEW)

Added risks are Security Misconfiguration and UnvalidatedRedirects and Forwards. Security Misconfiguration issue was A10 in the Top 10 from 2004, but was dropped in 2007 because it wasn't considered to be a software issue. However, from an organizational risk and prevalence perspective, it clearly merits re-inclusion in the Top 10; so now it's back. UnvalidatedRedirects and Forwards issue is making its debut in the Top 10. The evidence shows that this relatively unknown issue is widespread and can cause significant damage.

Removed risks are Malicious File Execution and Information Leakage and Improper Error Handling. Malicious File Executions is still a significant problem in many different environments. However, its prevalence in 2007 [9] was inflated by large numbers of PHP applications having this problem. PHP now ships with a more secure configuration by default, lowering the prevalence of this problem. Information Leakage and Improper Error Handling issue is extremely prevalent, but the impact of disclosing stack trace and error message information is typically minimal.

5 Security Testing Tools

Many web application security vulnerabilities result from generic input validation problems. Examples of such vulnerabilities are SQL injection and Cross-Site Scripting (XSS). Although the majority of web vulnerabilities are easy to understand and to avoid, many web developers are, unfortunately, not security-aware[7]. As a result, there exist a large number of vulnerable applications and web sites on the web.

A web application scanner is an automated security testing that examines web applications for security vulnerabilities. In addition to searching for web application specific vulnerabilities, the tools also look for software coding errors, such as illegal input strings and buffer overflows [8]. Most of these scanners are commercial tools (e.g., Acunetix Web Vulnerability Scanner and HP WebInspect), but there are also some free application scanners (e.g., Foundstone WSDigger and wsfuzzer) with limited use.

A method is method to evaluate and benchmark automatic web vulnerability scanners using software fault injection techniques [8]. Software faults are injected in the application code and the web vulnerability- scanning tool under evaluation is executed, showing their strengths and weaknesses concerning coverage of vulnerability detection and false positives. However, this study was focused on a various security testing tools or vulnerability scanner for a particular type of tool. In table III various security testing tools [12] used to detect OWA SP Top Ten attacks are presents and how these attacks occur. Security testing tools presented in table III are both open source and commercially available tools.

TABLE III. TYPES OF ATTACKS AND TOOLS USED FOR THEM

Type of Attack	How they occur	Security Testing Tools
1. SQL Injection[15]	SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands	ZAP, Francois Larouche, Antonio Parata, icesurfer, ilo
2. Reflected Cross-Site Scripting (XSS) [13]	In this attack doesn't load with the vulnerable web application but is originated by the victim loading the offending URL	XSS-Proxy, ratproxy, Burp Proxy, WebScarab, PHP Charset Encoder(PCE)
3. Stored Cross Site Scripting (XSS)[17]	Occurs when a web application gathers input from a user which might be malicious, and then stores that input in a data store for later use	PHP Charset Encoder(P CE) , Hackvertor, BeEF, XSS-Proxy, Backframe, Burp, WebScarab
4. DOM-based Cross-Site Scripting [16]	Occur when active content, such as a JavaScript function, is modified by a specially crafted request such that a DOM element that can be controlled by an attacker.	The DOMinator Tool, DOM XSS Wiki, DOM Snitch
5. Broken Authentication & Session Management Testing [12]	Attacker uses leaks or flaws in the authentication or session management functions (e.g., exposed accounts, passwords, session IDs) to impersonate users.	HackBar
6. Insecure Direct Object References [12]	Occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter.	Burp Suite
7. Cross-Site Request Forgery (CSRF) [12]	Occurs when attacker link or script in a page that accesses a site to which the user is known (or is supposed) to have been authenticated.	Tamper Data, monkeyfist
8. Security Misconfiguration [14]	Occur when the system admins, DBAs, and developers leave security holes in the configuration.	Watobo, Microsoft Baseline Security Analyzer
9. Insecure Cryptographic Storage [12]	Occur when web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing.	N/A
10. Failure to Restrict URL Access [12]	Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.	Nikto/Wikto
11. Insufficient Transport Layer Protection [12]	When Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic	Calomel
12. UnvalidatedRedirects and Forwards [12]	Common website functions, such as search results or account logins, frequently use redirects or forwards to send visitors to another destination. If the website doesn't verify the destination, redirects or forwards might be vulnerable.	Watcher

6 Conclusion

In this paper we have provided a review of various security risks to which web applications are vulnerable. A brief description of which type of security testing tool should be used for a particular type of attack is given. Various studies that are given in this paper show that the effectiveness of security testing tools in detection of vulnerabilities varies a lot. So researchers are still focusing on this major issue so as to make the web application more secure.

7 References

[1] Hetzel, William C., The Complete Guide to Software Testing, 2nd ed. Publication info: Wellesley, Mass : QED Information Sciences, 1988.

[2] Jiantao Pan, 18-849b Dependable Embedded Systems,"Software Testing",Spring1999.

[3] Types of security testing , Available : http://www.softwaretestingnow.com/types-of-software-testing

[4] Gary McGraw, Bruce Potter. "Software Security Testing"[J]. IEEE Security & Privacy, 2004, 2(5):81-85.

[5] OWASP Top Ten Project. Open Web Application Security Project. [Online]. Available: http://www.owasp.org /index.php/Category:OWASP_Top_Ten_Project

[6] Romain Wartel. Security Risks. [Online]. Available: http://cern.ch/security.

[7] Jan-Min Chen; Chia-Lun Wu; Dept. of Inf. Manage., Yu Da Univ., Miaoli, Taiwan, "An Automated Vulnerability Scanner for Injection Attack Based on Injection Point" . In Computer Symposium (ICS), Dec 2010 .

[8] Elizabeth Fong; Vadim Okun.,"Web Application Scanners: Definitions and Functions". In System Science, 40th Annual Hawaii International Conference on Jan 2007.

[9] Security risks 2007, Available: https://www.owasp.org /index.php/Top_10_2007

[10] Security Testing, Available: http://www.buzzle.com/ editorials/7-14-2006-102344.asp.

[11] Security risks 2010, Available: https://www.owasp.org/ index.php/Top_10_2010-Main.

[12] Security Testing Tools, Available: http://resources. infosecinstitute.com/owasp-top-10-tools-and-tactics.

[13] Cross-Site Scripting, Available: http://www.opensource testing.org/security.php.

[14] Securitymisconfigurations, Available: http://www.make useof.com/tag/test-computer-securitymisconfigurations –microsoft-baseline-security-analyzer.

[15] Tesing SQL Injection, Available: https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OWASP-DV-005).

[16] DOM-based Cross-Site Scripting, Available: https:// www.owasp.org/index.php/DOM_Based_XSS.

[17] Cross site scripting tools, Available: https://www.owasp.org/index.php/Testing_for_Cross_site_scripting.