Security of web application :Appearance and Challenges.

1st Sadia Afrin Laboni
17-33360-1
Computer Science and Engineering
sadiaafrinlaboni07@gmail.com

2nd Abdullah-Al-Mahadi
17-33324-1
Computer Science and Engineering
abdullahalmahadi5@gmail.com

3rd Iftekhar Rahman
17-33302-1
Computer Science and Engineering
iftekharshafi990@gmail.com

4th Md. Asiful Islam
17-33100-1
Computer Science and Engineering
asifuli751@gmail.com

***Abstract*—The web is totally essential piece of our lives. It is wide stage which is utilized for data sharing and administration over web. They are utilized for the money related, government, social insurance, instruction and numerous basic administrations. Ordinary billions of client buy things, move cash, recover data and convey over web with one another. Despite the fact that the web is closest companion of clients since it give whenever anyplace access to data and administrations simultaneously. Everything is made by human on the planet so it's existence that the things made by man are tad dangerous. So web applications are additionally made by human so it contains an excessive number of provisos. The prevalence of utilizations charm programmers towards them. Presently now a days securing and keeping up the sites against assault is hard and testing task. Discovering provisos in Web application, Computer framework or arrange and abusing them called hacking. New methodologies for web assaults are increasing everyday so the investigation of recognize and avert against web application assault and discovering arrangement is significant part in web world. In this paper we presented all web application based assault including two noteworthy assaults like XSS (Cross Site Scripting) and SQLI.**

***Index Terms*—Web Application Attacks, Web Security, Web Security, testing, Vulnerability.**

I. Introduction

Web application security and digital security is Current, demandable and hot profession field subject. The World Wide Web turns out to be most significant and worldwide data medium on the planet. It rapidly turns out to be most predominant approach to give access to online administrations. Client use electronic applications to look through information, trade message, connect with one another, direct business, and perform budgetary task and some more. For some clients the web is extremely simple to utilize and advantageous in light of the fact that it give 24 – hour whenever, anyplace access to data and administration for all concerned a lot is on the line: (A)Business: That get expanding salary from web business. (B)Users: Those who trust web application with touchy data. (C)Criminals: Those who can make enormous cash by taking installment subtleties or trading off financial balances.

II. Literature Survey

A. Studies carried out in Literature Survey

Garcia (1998)[2] addresses the use of performance metrics such as sensor detection rate, barrier delay times, and reaction force time results in an overall assessment of the safety effectiveness. This measure helps to link the risk to risks, goals, failure implications and the probability of attack at a facility.

Lyu R. Michael et al. (2000)[1] discussed the use of firewalls to protect network sites from external attacks and intrusion. It also addresses four key components in creating a firewall such as policy, advanced authentication, packet filtering, and gateway application.

Dima Alden et al. (1999)[3] addresses the use of programs for cryptographic module validation to improve and sustain web application security. An application can, for example, use cryptographic module to generate passwords etc. It also speaks about the use of COTS components as they can pose serious threats to the application's security aspect and thus become another justification for supporting open testing.

Choi Cheol et al. (2006)[4] addresses manipulation of parameters, manipulation of cookies, alteration or hijacking of a session with a user.

Wang Linzhang et al. (2007)[5] focuses on a security test methodology guided by threat model. They define danger as a state enabling attackers to breach security policy. Threats are behaviors that an attacker can present to the system and violate security properties such as authentication, authorisation, confidentiality and privacy.

Mendes Naaliel et al. (2008)[6] focuses on the problems and vulnerabilities caused by misconfigurations or the lack of some system for intrusion detection or firewalls. They also note the risks that can occur because of the use of off-shelf components.

B. Summary of Literature Survey

Author Name	Issue addressed
Mary Lynn Garcia [2]	Risk of attack
Michael R. Lyu et al. [1]	Use of firewalls to secure the network sites from external attacks and intrusion
Dima Alden et al. [3]	Usage of COTS modules, use of validation programs with cryptographic module
Choel Choi et al. [4]	Modification of parameters, modification of cookies, alteration or hijacking of a user session
Linzhang Wang et al. [5]	Focuses on threat model- security research approach. They define danger as a condition for attackers to breach security policy
Naaliel Mendes et al. [6]	Failure to detect intrusion or firewalls, use of shelf parts, misconfiguration

TABLE I

III. REVIEW PROCESS

A. Research Objective

A Systematic Literature Review was conducted following an adaptation of the process developed by Kitchenham et al. (2010), to identify the state-of - the-art how Security of web application was used for security purposes. SLR is an effective way to collect information on a subject or study issue, according to Kitchenham et al. (2010), once the most reliable evidence emerges from aggregating all empirical studies on a specific topic. Kitchenham et al. (2010) also claim that the suggested methodology for combining empirical research is an SLR approach that is present in several current studies, such as SLRs on the use and assessment of serious games (Calderón , Ruiz, 2015) (Petri , Gresse von Wangenheim, 2017). A SLR includes a systematic process (Figure 1) for identifying studies that address a given research problem, as well as a systematic presentation and synthesis of selected studies ' characteristics. We used systematic literature review to carry out this literature review. There are some individual tasks to complete in order to perform a systematic literature review. The SLR procedure is organized in the objective

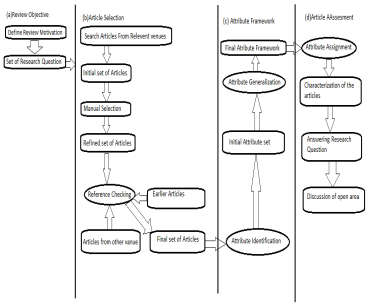

Fig. 1. Overview of systematic literature review

review phases, selection of article, framework of attributes, evaluation of article (Figure 1). First (Section a), an objective of review is defined to determine a research issue. Second, search articles from relevant locations, keywords and search string, digital libraries to search, final set of articles in article selection (Section b).Third, Attribute Framework,Attribute identification(section c) and fourth(section d),Article Assessment,Attribute assignment. The review process described in Figure 1 illustrates the tasks involved in this study's review protocol. The tasks will be discussed in the subsections that follow.

B. Research Questions

Q1: What are the security System testing vulnerabilities? Q2. Which are the challenges facing web testing for security? Q3: Which are the potential security aspects problems relevant to web testing?

Q4: Is automated testing useful to test Web applications for security?

Q5: Will the findings apply to essential factors for the practitioners?

C. Article Selection

Initially we did an automated keyword search(phase (b) in Figure 1) in the online libraries for article collection. After that, we made a manual collection from those results as well as a reference search for those posts. We maintained the criteria for inclusion in the selection of article. We took a few steps to preserve the requirements for inclusion.Initially, we looked for papers that had some strong relationship with security of web application monitoring. We confirmed the connection by analyzing the target, study area and methods of the researchers. We also included articles published in refereed journals and websites.

1) Inclusion criteria:: We have established the following inclusion criteria for selecting in advance in accordance with our study objective and questions that should be answered by the reviewed articles:

The subject area of the papers will show a strong focus on projects in Security of web application. Authors must state the study's goal (Security of Web application) clearly and provide detailed evidence of research methodology, data sets, and case study projects statistical information.

Using a manual analysis (discussed later in this section) of title, keywords, abstract, the suitability of the articles was determined against the above selection criteria. Conclusions are checked in case of doubt.

2) Keywords and Search String:: We first wanted to define the core concepts of this work in order to create the search string. Then we combined these search terms to form the search strings according to the searched digital library guidelines. The list of terms used for searching is as follows. Synonyms are also used in this process to complement the research conducted.

3) Digital Libraries to Search:: For the first collection of articles we did a widely automated keyword search, it is the first and foremost criterion to pick articles from online libraries. We searched the most common online libraries on the basis of the name, keyword and abstract. Five digital libraries were searched here for Systematic Literature Review articles: • IEEE Computer Society Digital Library; • The ACM Digital Library; • ScienceDirect; • SpringerLink; • Google Scholar; Such libraries are the famous repositories for research articles related to open source. All inquiries are concentrated on the name, abstract and keywords.

4) Manual selection:: Recent studies [1] [9] pointed out that (a) current digital libraries do not provide good support for automated keyword search due to lack of consistent set of keywords, and (b) the abstracts of articles are relatively poor in comparison to other disciplines. Thus it is possible that the 160 articles identified through automated search process might contain irrelevant ones and some relevant might be missing. Because of this, the manual selection was carried out on these papers by checking the name, keywords and abstract. The researchers analyzed the selected papers against the selection criteria in order to reduce the researcher bias in this selection process. This method has taken us to the 97 papers.

5) Reference checking:: As, we were aware of missing but relevant article we carried out a non-recursive search through the references of the 97 selected papers to ensure the inclusion of other relevant but missing articles .2 additional conference articles were found in this process.

6) Final set of Articles:: Finally, 7 articles ended up in the selection process of the article. A complete list of these articles can be found in our references along with year and venue wise distribution.

D. Attribute Framework:

The next step in the review protocol was to build a framework for attributes (phase (c) in Figure 1). Using this framework, the selected articles were characterized and the research questions answered. A brief description of this process is provided below.

1) Attribute identification:: At first we randomly studied some articles to identify the attribute and selected some attributes that used to describe and respond to research questions. We characterized the attribute in order to get the best answer to the question.

E. Article Assessment:

The article evaluation step consists of four separate activities as shown in Figure 1's phase (d) We're focusing on the first step in this section.

1) Attribute assignment:: We evaluated all the articles using the attribute framework and it helps us to gain clear knowledge of the articles and also helps to answer questions. To validate any claim, we also evaluated the data, resource and method.We characterized the reviewed articles and also evaluated the quality of the assignment in order to avoid any kind of bias.

IV. Security appearance and challenge

Application security issues lie not just in the risks and vulnerabilities of the application itself but also in the procedures and strategies taken within the enterprise to manage protection of the application. A nearer look at some of the pinnacle application safety can assist you avoid some of the most common mistakes. When more companies focus on the development of software, new versions of apps are released rapidly, not only making protection more relevant but also more difficult. Make sure you get the most value from the money that invests in protection for AppSec by providing it with information of the most popular threats and internal missteps.

A. Outside threats: The most common application security vulnerabilities

1) SQL Injections- : This sort of attack includes the insertion of a SQL question into the software so the attacker can examine sensitive statistics from the database, alter database facts, or perform different malicious activity. SQL Injection flaws make their way into programs when builders create dynamic database queries that encompass user-supplied input.

Having said that, avoiding writing dynamic queries is the only way to prevent SQL injection flaws. If this doesn't happen, you need to protect against (prevent) user-supplied input this contains malicious SQL from affecting the logic of the query being performed.

2) Remote command execution (RCE)— : RCE occurs when an attacker may add his or her own code for executing arbitrary commands. They can escalate their server privileges, insert malicious code or worse.

To secure your application:

• Stop as many calls as possible from the command line. (But then, use APIs.)

• Using shell commands (operating system commands will be executed), ensure that the input values do not contain malicious characters.

• A detailed analysis of the code.

• Run servers with limited permissions, allowing them to access only what they want.

3) Cross-site scripting : Cross-site scripting (XSS) is the second most common flaw in Web applications. XSS allows attackers, unknown to the user, to execute scripts in the visitor's browser on behalf of a compromised website. One can be routed to malicious sites or otherwise be subject to malicious behavior. The key point to bear in mind for defending against XSS is that it is easily detected with security testing tools for applications. So it is important to use these tools as part of the security testing process.

4) Inherited vulnerabilities : Software developers also use software frameworks built on long-established languages such as JavaScript in an attempt to remain. So developers can easily build and test applications. Yet these frameworks also rely on various overlapping dependencies and can pull components across the Internet from unknown sources — opening apps to threats. Developers appear to rely more on a given JavaScript library's popularity to assess its protection, making the mistaken assumption that if a lot of developers use it, then it must be secure. It is a faulty approach that bugs the query.

Preventive measures to avoid this form of attack should be taken:

• Take packages that is wanted to use and replicate them locally in your construction environment, instead of pulling them directly from the internet whenever you create. To put it another way, do not create and deploy from the Internet.

• Use program vulnerability tools to find vulnerable packages in your local registry, specifically Application Composition Analysis (SCA) software.

B. A look within: Internal application security challenges

1) Not using a variety of application security testing tools: It can only leave application vulnerable, depending on one or two methods for application security testing. It requires more than a few resources to ensure your application has complete coverage. A broad variety of testing methods including Static Application Security Testing (SAST) methods, Dynamic Application Security Testing (DAST) tools, Interactive Application Security Testing (IAST) tools, and SCA tools are the best approach. The best approach is one which combines manual testing and modeling of the hazard. Managing all these devices can, of course, become very complicated, and in itself a challenge. Fortunately, an application vulnerability manager can be employed to compare the tests and present them in one uniform format. Some of these tools also cross-reference findings from SAST and DAST tools help classify possible risks that can potentially be exploited.

2) Lack of progress tracking : Identifying threats and vulnerabilities is one thing, but if one does not delegate them out and ensure that they are addressed quickly, the application will suffer. Another advantage of an application vulnerability manager is that it interacts with common development environments and tools for issue tracking. Developers can be allocated problems within their chosen working environment, and progress can be easily monitored to ensure problems are solved immediately. To help track progress certain methods retain well-understood metrics. Making sure the use of the tool gives you the details needed.

3) Pressure of speed : Gartner recently pointed out the increased speed at which production of applications occurs. This speed trap also leads engineering teams down a road that places safety at the fringes. This takes only one failure and one loses the trust of current and potential customers. From day one, protection must be integrated in the design and development process, and it remains a priority – even if it slows you down a little. They're worth it.

4) Failure to build the right team : The protection of the applications goes beyond developers. This requires quality control, compliance monitoring, management buy-in and a dedication to organization. A developer does not make safety a priority if the management forces them to deliver updates in an unacceptable period of time. Leaders need to walk the walk and demonstrate their dedication to robust app protection.

5) Failure to develop a formal AppSec plan : The reliability of application cannot be achieved with a whim. There must be a structured plan in place, recording the methods used and operational requirements for security testing for applications. The strategy will be revisited periodically to ensure that it still meets one's organization's needs. Monitor and track performance so that one can see how the company is doing over time and change policies as appropriate. A little extra time and effort may be required to ensure that a safe and stable application is deployed. With a little preparation and planning, take appropriate measures to avoid the most popular security vulnerabilities in applications and ensure that AppSec takes the correct operational approach.

V. REVIEW RESULTS AND FUTURE DIRECTION

A. Review Results

Q1: What are the security System testing vulnerabilities?

ANS.: Web security testing vulnerabilities include SQL injection, content injection, file injection, XML injection, LDAP injection, XPATH injection, cookie manipulation, cookie sniffing, cross-site request forgery, cross-site scripting, session hijacking, authentication, disclosure of details, Clickjacking etc.

Q2. Which are the challenges facing web testing for security?

ANS: The development of automated tools for computer security testing, use of RIAs web applications, use of insecure cryptographic storage involve numerous challenges faced by security web testing. Security tester should protect itself against various unspecified attacks such as attacks on repudiation etc. Any trapdoor may become a potential threat to the application for a security tester.

Q3: Which are the potential security aspects problems relevant to web testing?

ANS: Broken or poor passwords, buffer overflows, secret field abuse, inappropriate use of authentication, cookie sniffing, server misconfiguration, bad session management, sensitive disclosure of data, abuse of parameters, social hacking, insufficient input validation, etc. are other potential issues related to the security aspect of web testing.

Q4: Is automated testing useful to test Web applications for security?

ANS: Developing automated tools has always been a challenge for evaluating protection for web applications. Construction of automated tools for security testing web applications is more difficult than checking the web application's functionality. The challenges faced by automated web security testing tools are to keep up with ever-evolving and changing technologies (such as RIAs) and to incorporate them effectively into existing development workflows.

Q5: Will the findings apply to essential factors for the practitioners?

ANS: We assume that concentrating on various issues and challenges related to web application security testing can yield significant dividends in recognizing various dangers, weaknesses, assaults, threats, viruses, etc. associated with web-based application security testing, which can thus be prevented when developing web application. It will also be beneficial to direct a safety tester to skillfully model the test framework and develop the appropriate test strategy.

B. Future Direction

Increasing security breaches have made safety testing an important part of the life cycle of web application creation. Web-based security research helps in emulating and identifying possible vulnerabilities and risks associated with the web application. It also tests to see if the application meets all security criteria when exposed to any malicious input data. The technology is growing at a faster pace so the testers

recognized the need to classify the robust testing capabilities in order to respond to the complex and heterogeneous nature of the web domain. The paper addresses numerous problems and challenges relating to the existing security testing scenario for web-based applications.

With the proliferation of web applications, there is a growing need for understanding, dedication and efficient quality assurance (QA) processes across the life cycle of the application. Security practitioners should ensure that there is proper culture and quality control systems for growth over the life cycle. If they outsource their app-dev, they should ensure that suppliers adhere to the appropriate level of quality assurance based on protection. Developers should keep safety in mind throughout the phases of application specifications, architecture, and design.

Ideally, developers and security professionals have some basic training (if not certification, yet) in security applications-even basics such as requiring security elements right from the start in the requirements stage, including input validation, error handling, safe handling of data and other features while writing code. Traditionally, security teams serve the role of the network and the infrastructure, with little engineering expertise or interest. Likewise, technology people have no background or interest in protection or network-that's the big gap in understanding protection of applications. Unless the top officials of the company do not take the responsibility for protection and technology policies, hackers will flourish. Top officials in the company will take measures to ensure security. There needs to be increased interest among developers by taking seminars, workshops, videos and making the work more interesting. Developers should also incorporate security safeguards into the program such as providing adequate validation of inputs to prevent the application from being targeted using command injection techniques.

Besides there are many who can write programs and learn very quickly, particularly the young. Yet they are also not attuned to write their codes securely or to guarantee consistency. They write beautiful apps with rich functions and features but miss the point that hackers can insert code to hijack or crash their devices. There has to be more emphasis on defense. Developers have the best intentions to write good code, but time, money and priorities demands make safety quality control take a back seat and an app goes into production with faults hackers find and exploit. Appropriate schedule and time should be set to ensure the best quality and protection.

This also articulates that research is based on implementing technology such that potential research techniques can keep track of all issues while creating careful test design and testing methods for the same as well as adjusting to the complex ubiquitous nature of the internet. This finding points out that there is an important need to create an efficient testing environment to perform security testing of web applications that may also present new challenges during testing.

Couple this with automated QA safety scanning and monitoring tools such as "black-box" and "white-box" to ensure thorough coverage of the application's robustness before development. As new threats proliferate, both developers and security people need to stay up-to-date on new problems, and be ready and easy to access and deploy patches to "old" problems that are unexpectedly discovered or exploited.

The best way to guarantee application protection is to have the programs written correctly in the first place, in order to find minimal post-production corrections, to be accepted by the protection and development leaders. Besides firewalls for networks, there's no other way to combat device attacks. Web application firewalls are very hard to create, configure and use-that's why there are so few such vendors at the moment; while using Web application firewall makes sense to guard against insecure vulnerabilities.

VI. CONCLUSION

In this paper, we've got attempted to spot various issues and challenges faced by security testing of web based applications. A security tester thus should keep track of all the problems while conducting testing of web application for security. Also the knowledge would be helpful for designing and modeling the effective test strategy and therefore the test application. While performing security testing, a tester should also incorporate implementation related information and issues while testing which can be helpful in eradicating various vulnerabilities associated with the protection testing of web applications.

REFERENCES

[1] Firewall Security: Policies, Testing and Performance Evaluation. Michael R. Lyu and Lorrien K. Y. Lau. Department of computer science and engineering. The Chinese University of Hong kong, Shatin, HK. 2000 IEEE.

[2] Automated Security Test Generation with Formal Threat Models Dianxiang Xu, Senior Member, IEEE, Manghui Tu, Michael Sanford, Lijo Thomas, Daniel Woodraska, and Weifeng Xu, Senior Member, IEEE. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2012.

[3] Raising the bar on software testing - Alden Dima, John Wack, and Shukri Wakid (1999)IEEE.

[4] Automatic Test Approach of Web Application for Security (AutoInspect). Kyung Cheol Choi and Gun Ho Lee, Springer-Verlag Berlin Heidelberg 2006.

[5] A Threat Model Driven Approach for Security Testing. Linzhang Wang, Department of Computer Science, Nanjing University, Eric Wong, Department of Computer Science, University of Texas at Dallas, Dianxiang Xu, Department of Computer Science, North Dakota State University. Third International Workshop on Software Engineering for Secure Systems (SESS'07). 2007 IEEE.

[6] Assessing and Comparing Security of Web Servers. Naaliel Mendes, Afonso Araújo Neto, João Durães, Marco Vieira, and Henrique Madeira CISUC, University of Coimbra. 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing.

[7] Web application security assessment tools- Mark Curphey and Rudolph Araujo 2006 IEEE security privacy.