Security in Web applications

A. Santoyo-Sanchez[1], C. De Jesús-Velásquez[2],
L. I. Aguirre-Salas[3]

[1]Department of Computing, Universidad de Guadalajara –CUCEI, Guadalajara, Jalisco, México
[2]Compatibility Validation, Intel Tecnología de México S.A., Tlaquepaque, Jalisco, México
[3]Department of Engineering, Universidad de Guadalajara – CUCSUR, Autlán de Navarro, Jalisco, México
alejandra.santoyo@cucei.udg.mx

Abstract. The scientific and business communities today are particularly concerned to ensure that web applications containing less vulnerability on the issues of security and reliability of the information. This article proposes a guide with the aim of guiding the engineers Web application testing to find vulnerabilities within an application. This guide is based on standards ISO 9126:1991, IEEE 1233, IEEE 6190, IEEE 830, ISO / IEC 9646-1. The guide is currently incorporated in a wizard; some of the practical results obtained from using the same are presented in the case study.

Keywords:WEB application, quality, reliability, security, vulnerability.

1. Introduction

The use of Web applications to conduct business, banking, educational, information, e-business, issuance of certificates, among others, has prompted interest in ensuring the safety and reliability of the information displayed, processed, stored and/or transferred via the web.

The safety and reliability of Web applications has been approached from different perspectives and with different methods. Among the solutions are formal methods [1] and [2] which focus on specify, develop and test the quality of computer systems by implementing a rigorous mathematical notation. Unfortunately, formal methods are not generally used by the difficulty in practice in the use of such models.

In industry, semi-formal methods [3] [4] and informal [5], [6] are used for modeling and verifying aspects of security and reliability of Web applications due to its easy representation, however models obtained are difficult to analyze what aspects ensure security and reliability is a complex and imprecise. As different research groups have been building bridges between formal methods, semi-formal and informal as in [7].

Tendencias Tecnológicas en Computación
Research in Computing Science 66, 2013, pp. 161-174
Paper received 27-05-2013 and accepted 05-08-2013

Moreover according to [8] software vulnerabilities can be grouped into three categories: the design, development and implementation, and operation. Unfortunately, in practice, analysts and designers of Web applications do not specify security requirements, and often do not provide for vulnerabilities in their design.

This represents a challenge in the field on Information Technology Audit. So, for that reason in order to avoid these problematic this paper proposes a set of guidelines based on standards: ISO 9126:1991, IEEE 1233, IEEE 6190, IEEE 830, ISO / IEC 9646-1 which will be used in the testing stage of a development process of WEB software systems to help engineers in reducing the vulnerabilities of Web applications during the testing phase.

This paper is organized as follows. Section 2 describes how the security in Web application has addressed. Section 3 presents the contribution of this paper, i.e. the proposal guidance to achieve security and reliability within a Web application; whereas the section 4 presents a case study to illustrate the use of the guide. Finally, section 5 provides conclusions and future work.

2. Background

A Web application consists of a set of Web pages and components that interact to form a system running using server (s) Web, network (s), HTTP, and browser, in which its state changes to the information that users enter or request [9].

For quality Web applications from development methodologies, there are two main approaches 1) hypermedia community, and 2) the software engineering community [3]. The first takes care of all the methods and guidelines for writing, designing or writing content for Web publishing [10].

While the software engineering community in Web Engineering addresses the application of systematic methodologies, structured and measurable to develop, test and maintain Web applications [11] [12].

Unfortunately Web engineering has focused on the phases of life cycle analysis and design [13], while Web applications in addition require experts in the field of security to ensure data integrity, reliability, etc. [14].

According to ISO/IEC17799 [15] indicates that the security means a set of methodologies, practices and procedures that seek to protect information as a valuable asset and thereby reduce threats and risks (called information security) to ensure that resources system to be used the way it was decided (called security). In IEEE/std 610.12 [16] reliability is the ability of a system to perform the functions for which it was designed without failure, in other words intended to ensure that the information is not modified by anyone not authorized.

But despite numerous efforts to achieve security through code review and practice of software engineering, many professionally designed Web sites still suffer from security holes and make serious applications. This evidence suggests a need for tools and techniques to address the problem [17] [18].

Methods of developing security for the Web is a daunting task, partly because of security concerns arose after the development of the web application or after a security problem occurred. In [19] describes various tools that can be used for security within a Web application using a test, some of which are WEBGOAT, NETCAT, CURL and PROXIES. WEBGOAT created by OWASP (Open Web Application Security Project) is a Web application developed in J2EE designed with the objective of serving as a test bed and find vulnerabilities once Web application is deployed, it is supported by other tools like NETCAT which used to interact at http or any other protocol with the Web server, while CURL interacts at http command. Local PROXIES are used to intercept, modify and send requests to the server. Also looking for vulnerabilities or risk factors through the code, we can use a local copy of the web site. Additionally, in [19] is mentioned for these tools (NETCAT, CURL and proxies) to in order to work properly Firefox browser is required, these tools look for vulnerabilities simple as if you use the right browser, the right platform for the type of Web application, good 80/443 port operation (allowing http from any source, etc.). however these are used when the Web application is already on the server, which is why further investigations are focusing on addressing vulnerabilities in the design stage, although there are some mechanisms to verify single types of vulnerabilities, is not usually implemented if you try to avoid all the vulnerabilities that may appear in your web application, that is the reason we use [1] that describes a methodology for finding vulnerabilities in the design stage.

According to [1] AppScan DE is a method that looks for common vulnerabilities in Web applications and makes a list of them. Subsequently makes recommendations for solutions to ensure safety guidelines for reliability and quality in the application which focuses on UNIX platforms. Using ISO 9126:1991, IEEE 1233, IEEE 6190, IEEE 830, ISO / IEC 9646-1 and information on AppScan DE in the next section introduces the guide.

3. Guide on finding vulnerabilities

Vulnerabilities in Web applications in accordance with ISO / IEC 13335-1:2004 in [20] are a group of weaknesses that can be attacked by one or more threats, the most common according to the ISO 9126:1991 in [21], [22] are:

- *Usability*. It focuses on testing the effort needed for use, and individual assessment of such use by a system of users stated or implied.
- *Functionality*. Try the product's ability to provide features that meet specific needs.

- *Reliability*. Refers to test the ability of a product to maintain the level of performance under stated conditions for a period of time.
- *Efficiency*. Try the qualities that have to do with the behavior in time, resource utilization efficiency and compliance.
- *Portability*. Test focuses on the characteristics of adaptability of various settings, and instability co-existence with other platforms.
- *Maintainability*. Refers to prove the qualities of capacity for change, testing facility, stability and ease of maintenance.

On the other hand in [1] found a "perversion" of Web applications illustrated in Table 1, which shows which covers the tool AppScan DE yet within Windows platforms are others like that here are in Table 2.

The guide is based on questionnaires which indicate at each stage of development of a Web application security issues that must ensure the application in each stage. In this way the guide is intended to indicate that it has been validated.

Specifically in the stage of the questionnaires requirements help ensure the detailed definition in the life cycle of web application, where security is inherent.

Table 1. WEB "Perversion" that ensuring AppScan DE.

VULNERABILITY	HARDENED SERVERS	NETWORK SCANNERS	ACCESS CONTROL	SECURITY OF WEB APPLICATION WITH APPSCAN DE
HIDDEN HANDLING				✓
COOKIE POISONING				✓
BACKDOORS				✓
BUFFER OVERFLOW	✓			✓
STEALTH COMMANDS				✓
KNOWN VULNERABILITIES	✓	✓		✓
MY SETTINGS THIRD	✓	✓		✓
MANIPULATING PARAMETERS				✓
CROSS SITE SCRIPTING				✓
FORCED BROWSING			✓	✓

Table 2. Main issues of vulnerability found in Windows platforms.

WINDOWS SECURITY PROBLEMS	RESULT
PROTECTION OF POTENTIAL CUSTOMERS AND USERS OF VIRUSES THAT ENTER THROUGH THE VULNERABILITY IN THE WINDOWS META FILE CODE (WMF).	THE SECURITY FLAW BY DOWNLOADING AN INFECTED FILE ALLOWS A HACKER TO TAKE CONTROL OF YOUR PC, STEAL INFORMATION, PERSONAL PASSWORDS, ETC.
INTERNET EXPLORER ALLOWS REMOTE CODE EXECUTION.	ALLOWS THE VULNERABILITY IN THE BROWSER FOR REMOTE CODE EXECUTION.
UPDATES TO SNMP.	VULNERABILITY ALLOWS SIMPLE NETWORK MANAGEMENT PROTOCOL THAT REMOTE CODE EXECUTION.
INCREASED PRIVILEGES.	ALLOWS VULNERABILITY THAT ALLOWS PRIVILEGE ESCALATION.
INTERNET EXPLORER, ACCESS TO DOCUMENTS IN ANOTHER PAGE.	ERROR ON HOW TO HANDLE REDIRECTS THE URLS IN HTML MANAGER, WHICH ALLOWS ACCESSING DOCUMENTS SERVED FROM ANOTHER WEB PAGE.

In addition to ensuring they are implemented in the security policy and standards for the development team.

Some questions contained in the questionnaire are:

1. Is there a security protocol to avoid outside attacks and intrusions from hackers?
2. Do you avoid being seen, the program names and directories?
3. Does it protect the integrity of their programs and data?
4. Are the services offered are made via secure transaction channels?

Just to mention some of the questions you can ask in order to have a general idea of the aspects to be taken into account for accomplishing security designing a Web application.

In *analysis* and *design* phases looking for inconsistencies in design requirements, we recommend taking into account the following security mechanisms: user management, authentication, authorization, data confidentiality, integrity, reliability, session management, transport security, segregation system in levels, and privacy.

During *implementation* phase coding is completed in the design of a Web application. The goal is to generate a list of recommendations for developing and testing the sequence relationships between webpages in order to get a good robust application and ensure optimum performance. Issues that must meet a Web application according to ISO 9126:1991 with IEC (International Electrotechnical Commission) [21] and IEEE / Std 610.12, 1990 [16] are functionality, reliability and usability.

In *test* phase should check the requirements, analyze the design and code review. In case of errors or new requirements in Web development requires a budget to determine the cost, time and impact of a change in the existing product, document the change and verify the consistency changes.

As the test engineer must first identify all possible security holes site considering the characteristics tested. The issues proposed in this article for safety testing are as follows.

1) *DNS management*: Its main purpose is to detect whether the domain name is appropriate. More than one domain name can be assigned to a Web site. It is possible to add other names under the **.MX** or another, which will generate additional aliases for the site. It is recommended that all domains are redirected to the first screen corresponds to the official homepage of the Website.
2) *Protection of internal structure of the Website*: In this case you should check the site's internal structure, i.e. reducing the amount of information contained in the URL shown in the program as user data display, and directory names program, etc. It is recommended that mechanisms for transferring information between webpages is at the level of server objects, thereby preventing the customer to be responsible for the transfer of data between the server running sessions. Another way is to prevent access to elements of Web server addresses is associated with a session on or associated with *SessionId* or *UserId*, this because with simple steps you can know the session *token* and pretend that this is the same user returning to the site. Protections should be incorporated on the address source IP address.
3) *Protection against robots*: Determine if the search robots or spiders known as spiders from entering them. Not all site directories should be available for search bots to enter them. You must use the *robots.txt* file, which is a plain text file containing instructions in the meta-tags of the homepage to prevent their access.
4) *Managing privacy*: verify the privacy of users of the site permanently, according to site policy. There should be physical and logical protections on information. For example customers protect physical servers in different data storage, ideally separate interfaces including data query. Also incorporate mechanisms for data encryption to sensitive information.
5) *Secure channel*: Verify the encryption of the communication channel for transferring private information between users and the Web site, via the Internet. It recommends the establishment of the channel SSL (Secure Socket

Layer), although the mean delay in starting the initial connection, not subsequently results in increased bandwidth and server resources increases.

6) *Access control mechanisms*: Check the protection of user privacy on the web site content, in the context of keys and authentication as simple and advanced electronic signature which is a system that identifies the user when performing transactions through Internet or closed networks, another way is through authentication with username and password pair that should provide key feedback mechanisms, by providing answers to predefined users, and do so by email. You can also use hardware systems for authentication, in this case should incorporate mechanisms such as token cards, and security.
7) *Protection program*: Check the protection code and the server's internal programs. How to avoid the transfer of parameters across the direction of access to the pages, to avoid reading the executable from the directory server, in the case of scripts using code compaction.
8) *External versus self-Site Hosting*: an objective evaluation based on capabilities, support requirements and effective.
9) *To ensure minimal roles*: defining the various roles within the definition and design of the site. That is, choose the most competent staff that can meet them, for example, the architect who is responsible for making working configurations of servers and applications, the application manager, the quality control manager, security manager, among others.

Table 3 is a brief questionnaire that helps the verification of vulnerabilities and compliance.

To carry out a general and adequate guidance is recommended the following practices.

- *Test early and test often*: This practice focuses on testing before, during and after the Web application in which these tests should be performed frequently.
- *Understand the scope of security:* Classify information according to the degree of protection considering legal issues.
- *Understand the scope of study:* Having accurate documentation as architecture, data flow diagrams, use cases, etc., minimum infrastructure for monitoring and analysis of attacks and network applications.
- *Use the tools:* there are open source and commercial, they simplify and speed the security process, helping the security personnel in their tasks.
- *What is important is in the details*: Review results and eliminate errors.
- *Use the source code when available:* check the code to assist in this review and discover the vulnerabilities that are not detected in the inspection of code phases.

4. Case of study

The guide was used and tested during the development of a Web application for a company, but for reasons of confidentiality name is omitted.

The aim of the guide is to make relevant suggestions about safety and reliability in the testing process of Web applications on Windows platforms in a way that ensures the integrity and proper functioning of them. This guide is done in phases and each phase of life cycle contains the criteria that indicate when you can go from one phase to another.

Table 3. Format verification of security testing

Security Concepts	Meets	
	Yes	No
Does the site work properly and no failures to browse their pages or use your services? (Especially in the case of online transactions).		
The data entered by a user through forms; are validated before being sent and processed by the site server?		
Is there a security protocol to avoid outside attacks and intrusions from hackers?		

A. Inception.

In order to use the guide in the test site must have certain characteristics to determine whether or not, using the following evaluation criteria:

- Attendance of all involved to define the cost / time and scope.
- Agreement that the requirements have been established and captured, and there is a shared understanding of these.
- Agreement that the estimate of the cost / time priority, the risks and the development process are appropriate.
- All risks are identified and a mitigation strategy is each.

B. Elaboration.

At the end of the design phase objectives were examined and detailed scope of the system, the choice of architecture, and resolution of major risks.

C. Construction.

Once the product was ready for delivery it was necessary to do the following: ensure that all functionality has been developed and all tests are completed.

D. Safety Test.

The activities you can do to make safety tests are diverse and are directed to several areas in the tables 4 to 12.

The guide will enable and disable points to prove. After you define the effort devoted to each test, i.e. testing effort goals are conducted to verify the safety and reliability of the Web application.

Table 4. DNS test Management

DNS Management	Meets	
	Yes	No
Do all site links are associated page and appropriate content to link said?		
If a search within the site or any operation therein, are the results displayed properly?		

Access requirements. Site users must have access to the intranet and database.

Table 5. Protection of internal structure of the Web site

Protection of internal structure of the Web site	Meets	
	Yes	No
Do you avoid being seen the names of the programs and directories?		
Do they take steps to confirm that the same user returns to the site?		

Table 6. Protection against robots

Protection against Robots	Meets	
	Yes	No
Is it prevents access to protected directories?		
Do meta-tags used are appropriate to protect the site's directory?		

Table 7. Privacy Management Test

Privacy Management	Meets	
	Yes	No
Are private data, provided voluntarily by users are stored in a reserved manner?		
Do you offer a privacy policy of personal data and reported its existence in the relevant pages?		
Does it protect the integrity of their programs and data?		

Table 8. Secure Channels Test

Secure Channel	Meets	
	Yes	No
Are the services offered are made via secure transaction channels?		
Is incorporated encryption mechanisms of the communication channel for transferring private information between users and the website? (SSL, PGP, etc.).		

Table 9. Access Control Mechanisms Test

Access Control Mechanisms	Meets	
	Yes	No
Does the site work properly and no failures to browse their pages or use your services? (Especially in the case of online transactions)		
Are the data entered by a user through forms validated before being sent and processed by the site server?		
How the issues that require restricted access, the site provides a means to validate access, i.e. through a box with username and password?		

Table 10. Protection programs Test

Protection Programs	Meets	
	Yes	No
Is there a security protocol to avoid outside attacks and intrusions from hackers?		
Do you have a data backup policy in order to overcome effects of failures derived from the previous point?		

Table 11. External versus self-Site Hosting Test

External versus self-Site Hosting	Meets	
	Yes	No
Is allows combination of services and infrastructure to provide an optimal solution set in terms of cost-benefit?		
Is an assessment of the overall impact of all elements involved in the site?		

Table 12. Ensure minimal roles Test

Ensure minimal roles	Meets	
	Yes	No
Does the security policy implemented to validate the restricted access is adequate to the purposes of service or the institution?		
In the context of Web site operation: Do you have staff able to meet the diverse functional roles?		

Table 13. Type of test recommended

TYPE OF TEST	APPLIES	
	YES	NO
DNS MANAGEMENT		
PROTECTING YOUR WEB SITE'S INTERNAL STRUCTURE		
PROTECTION AGAINST ROBOTS		
PRIVACY MANAGEMENT		
SECURE CHANNELS		
ACCESS CONTROL MECHANISMS		
PROTECTION PROGRAM		
EXTERNAL HOSTING VS. OWN SITE		
ENSURE MINIMUM ROLES		

Resources for the effort test. For example: *HARDWARE*. Any computer with Internet access, network infrastructure at least 100 Mbps. *SOFTWARE*: Platform Windows installed with any browser. Or of *PERSONAL*: test engineer, application support (on request).

Table 13 shows the results after a series of tests with which ensure quality and application security.

5. Conclusions

This paper has presented a guide for test engineers in order to identify vulnerabilities in Web applications, i.e. reliability and safety issues to consider and we'll fix it before delivering the final product to the user, this is achieved by implementing a wizard in which based on responses from online questionnaires related to aspects of safety and reliability of Web applications being tested, the questionnaires are based on ISO 9126:1991 standards, IEEE 1233, IEEE 6190, IEEE 830, ISO / IEC 9646-1.

The usefulness of this method has been demonstrated in a case study, favorable results were obtained in the testing stage and help test engineers to see more clearly where there is more uncertainty in Web applications, also reduces the costs involved fees and better yet, it is easy to use, thus reducing training time cost.

It is important to mention that it is based on results that although sounds outdated, the test guide can be applied to current web application development. Since it is based on common practices, which every developer should know and apply.

As future work, the idea is to implement more projects in a wizard application, so you can find best practices, and according to their results being used as a base for projects more complex.

Acknowledgment

This work was partially supported by SEP in the scope of the project COECYT - Michoacán and ITSCH DELFIN 2008 and 2009.

References

1. C. A. Jerez Lugo. Seguridad para lograr Confiabilidad y Calidad de los Servicios Digitales en Internet. Bachelor dissertation, Ed. Universidad de las Américas: Departamento de Ingeniería en Sistemas Computacionales, Mayo 2004.
2. S. Debnath, P. Mitra, N. Pal, C. Lee Giles. Automatic Identification of Informative Sections of Web Pages. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 9, pp. 1233-1246, 2005.
3. S.M. Abrahão, O.Pastor, L. Olsina, J.J. Fons. Un método para medir el tamaño funcional y evaluar la calidad de sitios Web. IV Jornadas de Ingeniería del Software y Bases de Datos (JISBD), 2001, Almagro (Ciudad Real), Spain, pp. 477 – 490.
4. C.H. Liu Kung, P. Hsia, C.T. Hsu. Structural testing of WEB applications. Proc. 11th IEEE International Symposium on Software Reliability Engineering ISSRE, 2000, San Jose, CA, USA, pp. 84 – 96.
5. S. Zarei. Electronic Service Quality Evaluation Methods for Online-Banking System. International Journal of Computer Science and Technology (IJCST), Vol. 1, Issue 2, pp. 6 – 13, Ed. C/O Ayushmaan Technologies, 2010.
6. Y. Deshpande, S. Hansen. Web Engineering: Creating a Discipline among Disciplines. Vol.8, No. 2, pp.82 – 87, Ed. IEEE Multimedia, Abril – Junio 2001.
7. D. Andrés Silva, B. Mercerat. Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Megazine: Colombian Journal of Computation, Vol. 2, pp. 79 – 95, Dic. 2001.
8. Desarrollo Seguro de Aplicaciones. http://es.scribd.com/doc/54453491/6/Aplicaciones-inseguras, May 2011.
9. S. Sampath, V. Mihaylov, A. Souter, L. Pollock. Composing a framework to automate testing of operational Web-based software. Proc. 20th IEEE International Conference on Software Maintenance (ICSM), 2004, Chicago, IL, USA, pp. 104 – 113.
10. L. Rosefelf, M. Peter. Information architecture for the world wide Web. Cambridge, Mass, Ed. O'Reilly & Associates, 2002.
11. D. Lowe, B. Henderson-Sellers. Impacts on the development process of differences between web systems and conventional software systems. International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR), 2001, L'Aquila, Italy, pp. 1 – 12.

12. Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke, and B. White, Web Engineering. J. Web Eng., Vol. 1, No. 1, pp. 3 – 17, 2002.
13. M. Jose Escalona, G. Aragón. NDT. A Model-Driven Approach for Web Requirements. IEEE Transactions on Software Engineering, Vol. 34, No. 3, pp. 377 – 390, May-June 2008.
14. Y. Deshpande, S. Hansen. Web Engineering: Creating a Discipline among Disciplines. IEEE Multimedia, Vol.8, pp. 82 – 87, April–June 2001.
15. ISO/IEC 17799 (International Standar ISO/IEC 17799 Second edition 2005-06-15) htpp:/www.iso17799software.com/.
16. IEEE/std610.12 (Standard Glossary of Software Engineering Terminology) 1990, IEEE http: www.swen.uwaterloo.ca/~bpekilis/public/softwareEnGlossary.pdf.
17. A.D. Rubin, D.E. Geer Jr., A survey of WEB security. IEEE Computer Society Press, Vol. 31, pp. 34 – 41, Sep 1998.
18. D. Scott, R. Sharp. Developing secure WEB applications. IEEE Educational Activities Department, Vol. 6, pp.38 – 45, Nov. 2002.
19. H. M. Racciatti. Seguridad en aplicaciones Web. Revista @rroba, Vol. 96, Suplemento Hack paso a paso No.27, Noviembre 2007.
20. http://www.iso.org.
21. ISO 9126 (Software Engineering Product Quality part 1, 2, 3), http://www.iso.org.
22. J. L. García Cerpas. Sistema asistencial en el proceso de pruebas para microprocesadores y tarjetas madres. Master disertation, Ed. Universidad de Guadalajara Maestría en Sistemas de Información, Julio 2007.