Security strategy for vulnerabilities prevention in the development of web applications

Security strategy for vulnerabilities prevention in the development of web applications

S Vargas[1], **M Vera**[2], **and J Rodriguez**[3]
[1] Facultad de Administración y Negocios Internacionales, Universidad Simón Bolívar, San José de Cúcuta, Colombia
[2] Facultad de Ciencias Básicas y Biomédicas, Universidad Simón Bolívar, San José de Cúcuta, Colombia
[3] Departamento de Sistemas, Universidad Simón Bolívar, San José de Cúcuta, Colombia

E-mail: m.avera@unisimonbolivar.edu.co

Abstract. In recent years, Higher Education Institutions through their Systems departments have strengthened security for the development of applications on web environment, because of their vulnerability to possible computer attacks. This research proposes a security strategy to reduce the risk presented by the web applications developed in the systems department of the Simón Bolívar University, in San José de Cúcuta, Colombia, based on a diagnosis of the current state of its security policy compared to other institutions of the department of Norte de Santander, the analysis of current regulations and the state of the art of security in web applications, as an object of study. This strategy of safe web software development arises in order to establish the security parameters that should be applied by the web software developers of the Institution, shielding the developed applications and thus guaranteeing the integrity of the information that is manipulated through them. The strategy was validated through expert judgment in the field of web application development, emphasizing the importance of applying it to prevent vulnerabilities in institutional web software and thus provide greater reliability in the management of information.

1. Introduction

At present, the software transition processes developed for desktop environments to web environments are evident, due to the multiple advantages of this type of applications, including their multiplatform nature to run in different web browsers that guarantees quickly access without requiring download, installation and configuration. Likewise, the updates made to the application are available transparently to the user and it uses less resources. Further, web application developers require a guide to produce secure applications taking into account principles such as confidentiality, which allows access to data only to validated users [1].

The investigation arose from the need to provide to the systems department of the Simón Bolívar University (Unisimón), San José de Cúcuta, Colombia, with best security practices for software development. For this, it is based on international guidelines for the prevention of security incidents. In this research, are considered sensitive aspects that range from the taking of requirements to the tests and implementation linked to the life cycle of the development of secure software.

The Institution lacked security strategies for software development; because the security parameters were determined by the programmer at the time of carrying out the developments. For this reason, as a

first step in the investigation, the security parameters applied in other higher education institutions of the Norte de Santander, Colombia, were investigated, comparing the results with the current status of the Unisimón.

Taking into account the information collected in the state of the art and the theoretical framework, a secure web software development strategy was designed, with the aim of providing the Institution's developers with the patterns to follow when developing web applications, that guarantee security, avoiding effects of vulnerabilities such as injection attacks on applications, cross-site scripting (CSS) failures, broken sessions, insecure references to objects, cross-site request forgery (CSRF) attacks, insecure cryptography, among other failures.

With this approach, the aim is to guarantee the integrity of the information and strengthen the applications functioning that are developed and implemented in several Unisimón operative units.

The validation of the proposed strategy was carried out through expert judgment which, finally, determined the relevance and applicability of this approach in the web software development processes generated within the aforementioned Unisimón department.

2. Methodology

2.1. Design

In software engineering the goal is to build a product or improve an existing one. This is achieved thanks to an effective process that must provide standards for the development of efficient, safe and quality applications [2].

The design of the security architecture of an application must be done at the beginning of the software development cycle. After the security requirements analysis phase of an application, the selection for its archetype and design pattern must also be based on security in other aspects. All types of software may have vulnerabilities in the code of all levels of architecture, and design vulnerabilities in their platforms and architectural components.

The number of attacks an application may suffer depends on the vulnerabilities in [3]: Software architecture components and levels, platform and operating system, client software security including the operating system, application servers, net, database administration system, development technology, programming languages, online protection, security experience and knowledge of programmers.

Individual security objectives can be used to divide the application architecture, its subsequent analysis and support the identification of application vulnerabilities. This approach leads to a design that optimizes the activities specified in Table 1 [4].

Table 1. Activities contemplated in the design of security objectives.

Activity	Description
Authentication	The software definitely establishes the identity of the user trying to log in, usually, with credentials such as username and password
Authorization	Access control to resources and operations
Configuration management	Context execution, database connection, administration and protection of resources
Confidentiality	Protection of secrets and both the confidential data of the user and the application. Management of sensitive data using, generally, cryptographic algorithms
Integrity	Verification of alterations in data or libraries. Random values must be cryptographically strong
Availability	Maintaining the availability of information managed by a system or its resources
Not disavowal	Provide proof that a particular transmission or reception has been made, the receiver / transmitter cannot deny that it occurred

These security objectives can be used to make key decisions in the security design for an application and document them as part of the architecture. Figure 1, details the security problems identified in a typical Web application architecture.

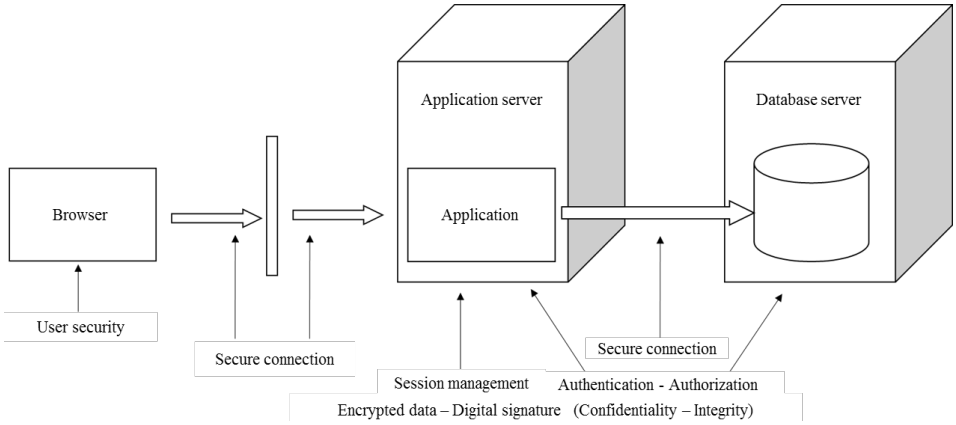

Figure 1. Security issues in a web applications architecture [3].

2.2. Secure software development life cycle

The secure software development life cycle is the basis for developing secure web applications. In this cycle, different technological processes are considered, such as, security requirements methods, risk analysis methods, safety checklists, security analysis tools, among others.

The tools and methods that interact during the secure software development process are intended to develop an application with the minimum-security vulnerabilities. Security controls ensure that the application works in a desired manner and provides defense against security threats.

In practice, security goes unnoticed in the first phases of the software life cycle, therefore one phase transfers its vulnerabilities to the next, which is structured in Figure 2 [5].

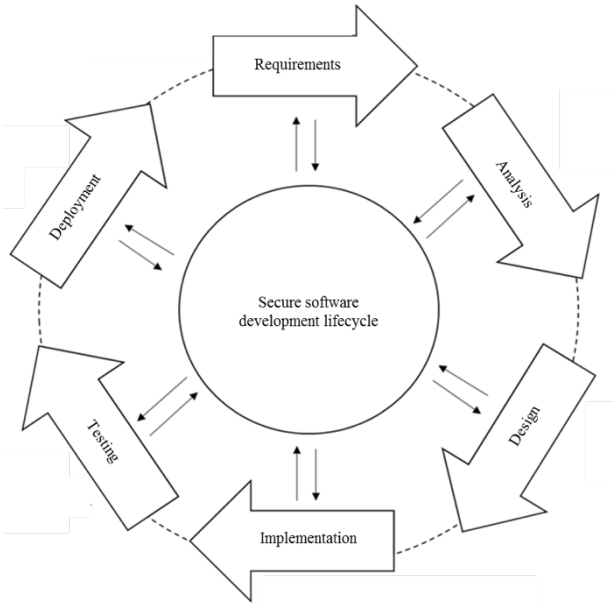

Figure 2. Secure software development lifecycle.

2.3. Security requirements analysis

Requirements engineering is critical to the success of any software project. Security requirements could be classified into three main categories: (i) functional, (ii) non-functional, (iii) derivatives. The functional ones list the functions that a system must perform; these refer to the inputs and outputs of a system. Non-functional, list the properties that a system must possess. Derivatives are those that arise from functional safety requirements [6].

These requirements could be treated take into account the user stories, which must be analyzed by development engineers who implement software security. User stories are an effective way to derive user requirements efficiently. It is important to anticipate abnormal behavior in a web application so that it is safe and reliable. For this reason, use cases should be created to mitigate abnormal behaviors [7].

2.4. Secure web software

The implementation phase of the software life cycle, achieves the development of the entire application code. There are several issues that must be developed in a reliable pattern to achieve security objectives during the implementation phase [3]. These contemplate the validation of input and output data with exception management and in the same way, session management and its audit and registration.

3. Results

The analytical method consists in split a whole, breaking it down into its parts to observe the causes, their nature and the effects [8]. The analysis is the observation and evaluation of a particular fact, it is necessary to know the nature of the object of study to know its essence. The analytical method was applied in this investigation, identifying the processes to be taken into account when developing secure web software and, thus, analyzing them based on the definition of indicators.

The structure of the strategy for secure web development is presented in Figure 3. At next, we describe the set of practices and guidelines proposed to strengthen the security of the web applications developed in the Unisimón.

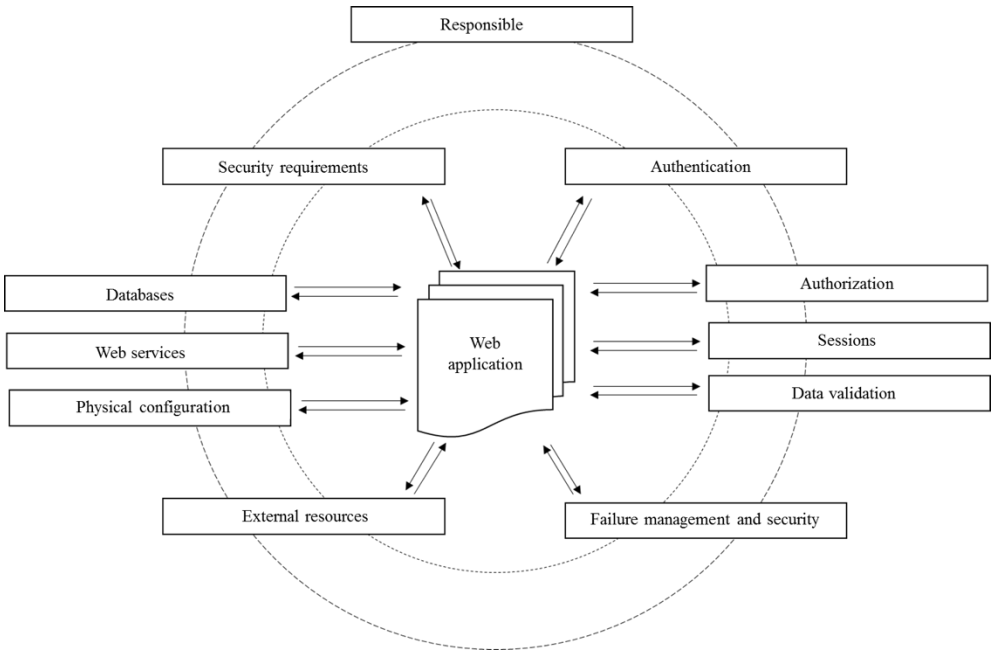

Figure 3. Secure web development strategy.

3.1. Responsible for web development

Responsible for web development without knowing it, implement web systems with security flaws, thus compromising the integrity of the application [9]. The head of systems of the Institution is responsible for defining, applying, evaluating and modifying the secure web development policy, as well as establishing the frequency of its update, guaranteeing information security.

The developers are responsible for applying the secure web development strategy and defining the controls that ensure that the development processes are guaranteeing the mitigation of security vulnerabilities.

3.2. Security requirements
For the development of a web application it is important to declare the security requirements at the beginning of each development phase. For this, the different available patterns must be taken into account to identify the appropriate one for the application requiring experience. This process includes security controls, identification, implementation, reviews and test environments [10].

3.3. Authentication
It is the procedure that allows validating that a user of the web application is authorized to access to the system. Modern websites use multiple methods to allow visitor authentication and thus grant different levels of access [11]. This process emphasizes the ideal selection of user names, assignment, security, change and reset of passwords, handling periods of inactivity, logins in different locations and account expiration.

3.4. Authorization
It is the validation of the functionalities that a user will have when accessing the system. The analysis of the problem of access and authorization provided by a web application to the available functionalities is addressed in [12], since once the attacker skips authentication, he would have available roles and privileges that would allow him to execute actions on institutional information. The implemented process incorporates the creation and management of profiles and roles, as well as access to resources.

3.5. Sessions
Sessions are fundamental in the construction of web systems. In [13] it describes session management vulnerabilities and possible attacks that could be exploited with that vulnerability. In the process, weak, permissive and exposed sessions are avoided, and include credentials on forms and session expiration.

3.6. Data validation
Data validation guarantees the integrity of the information used in the application interacts. It is necessary to validate the inputs and outputs thus checking their length, type of data or content to avoid possible injection of malicious code [3]. In this process, integrity control, validation of manipulated data and hidden fields, URL encryption and filtering of strings sent to the server were contemplated.

3.7. Fault management and security
Good fault handling ensures that the application does not show sensitive information to the user when something unexpected happens. It is important to implement mechanisms for handling errors and generating audit logs to monitor the different failures that may arise [14].

3.8. External resources
External resources are libraries implemented in the web application from external locations. These should be taken from reliable sources, as they could not only provide features to enrich web pages, but also threats when manipulated by third parties [15]. Otherwise, you must write the libraries that are required.

Sensitive information is protected by using the secure sockets layer / transport layer security protocol (SSL / TLS), which sets an encryption level that will be running until the session is closed with the server.

3.9. Databases
The database engines are responsible for storing all the information of the institution, so it must be designed applying integrity rules and a correct definition of the permits for each application. There are threats that can be presented in the databases, which is why they should be prevented once the web application is in production [16]. The structure of the databases includes the use of access credentials, definition of user privileges, creation of tables with integrity rules, implementation in a

network layer different from that of the application, authentication controls and authorization for entry and proper administration.

3.10. Web services

Web services allow the exchange of data between applications. For proper operation, mutual authentication and end-to-end security must be validated to ensure the integrity of the messages exchanged. There are good practices to maintain the integrity of the data shared through the use of roles and role certificates to manage the different intermediaries [17]. These provide for secure end-to-end communication, authentication between the user and the server, the integrity and confidentiality of the messages exchanged, verification of actions and traceability of the actions by audit.

3.11. Versions control

The documentation, source code and libraries developed and scripts of the database must be under change control and versioning procedures. The systems department must keep an updated record of the applications in production, with data regarding the version, date of last compilation and responsible for its support.

3.12. Physical configuration

It is the platform that supports the application at the software level (operating systems, database management system, web server, among others) and web application servers. It is important to keep the platform updated to minimize any vulnerability of the software that supports it [1]. Likewise, we must have a good configuration at the level of file and file permissions, user definition, software update, port enablement, necessary services and functions, maintenance windows for system log review and indexing through the file robots.txt.

4. Discussion

The nature of the proposed security strategy is defined as technical. This kind of strategy is directly related to the security architecture of technological systems and it combine rules and procedures in the configuration and / or maintenance of a system.

The types of vulnerabilities can be defined in terms of each phase of the software development life cycle [18]. In this sense, the research findings contribute from the disaggregation of these vulnerabilities to find in the design phase, specific procedures to counteract possible computer attacks.

5. Conclusions

The strategy fosters the development of secure web systems, executing good practices that reduce application vulnerabilities, minimizing the risks to which it is exposed. It is very important to provide the institution's developers with the patterns that they must take into account when starting the development process since the security of the applications will not depend exclusively on the considerations of the developer in charge, but on a policy institutional that establishes the steps to follow to guarantee the integrity of the information and the continuity of the academic and administrative activities that are part of the daily life of the institution.

The implementation of this strategy will reduce the risks of loss of productivity of information systems, since within the dynamics of quality concerning the continuous improvement and strengthening of information systems, it is imminent for the Institution to migrate all its systems to environments web, this in the face of the increasing competitiveness of the modern world, which has a tendency towards improvement and easy access to information. Likewise, a relevant procedure was proposed so that when starting the construction of new software products, the possible security risks on their applications are taken into account, and in this way they generate the protection mechanisms based on the modalities of attacks existing informatics, thus safeguarding the institutional

information of the dependencies that integrate it and profiling itself as a reference regarding the implementation of security strategies for web development in the Higher Education Institutions of the Norte Santander, Colombia.

References

[1] Canedo G, Flores M, Hill A, Martinez M, Papaleo M, Soarez N and Targetta C 2017 *Secure coding practices guide* (Lisbon: OWASP foundation)
[2] Booch G, Jacobson I and Rumbaugh J 2000 *El proceso unificado de desarrollo de software* (Madrid: Addison Wesley)
[3] Bermejo J R 2014 *Assessment methodology of web applications automatic security analysis tools for adaptation in the development life cycle* (Madrid: Universidad Nacional de Educación a Distancia)
[4] Goseva-Popstojanova K and Perhinschi A 2015 On the capability of static code analysis to detect security vulnerabilities *Information and Software Technology* **68** 18
[5] Daud M I 2010 Secure software development model: A guide for secure software life cycle *International Multi Conference of Engineers and Computer Scientists* (Hong Kong: IMECS) p 17
[6] Hope P and White P 2007 *Software security requirements the foundation for security* (Dulles: Cigital Inc.)
[7] Common Criteria for Information Technology Security Evaluation 2005 *Part 2: Security functional requirements, version 2.3* (United States and other countries: Common Criteria)
[8] Ruiz R 2006 *Historia y evolución del pensamiento científico* (México: Martínez Coll Ediciones)
[9] Smith M and Dehlinger J 2014 Enabling static security vulnerability analysis in PHP applications for novice developers with SSVChecker *Conference on Research in Adaptive and Convergent Systems* (New York: ACM DL) p 278
[10] Okubo T and Tanaka H 2008 Web security patterns for analysis and design *15th Conference on Pattern Languages of Programs* (Nashville: ACM DL) p 25
[11] Mundada Y, Feamster N and Krishnamurthy B 2016 Half-baked cookies: Hardening cookie-based authentication for the modern web *11th ACM on Asia Conference on Computer and Communications Security* (New York: ACM DL) p 675
[12] Neville-Neil G V 2007 Building secure web applications *ACM Queue* **5** 22
[13] Li X and Xue Y 2014 A survey on server-side approaches to securing web applications *ACM Computer Surveys* **46** 54
[14] Hernandez E 1999 *Auditoría en informática* (México: CECSA)
[15] Cao Y, Li Z, Rastogi V, Chen Y and Wen X 2012 Virtual browser: A virtualized browser to sandbox third-party javascripts with enhanced security *7th ACM Symposium on Information, Computer and Communications Security* (Seoul: ACM DL) p 8
[16] Mavromoustakos S, Patel A, Chaudhary K, Chokshi P and Patel S 2016 Causes and prevention of SQL injection attacks in web *applications 4th International Conference on Information and Network Security* (New York: ACM DL) p 55
[17] Yao D, Koglin Y, Bertino E and Tamassia R 2007 Decentralized authorization and data security in web content delivery *ACM Symposium on Applied Computing* (Seoul: ACM DL) p 1654
[18] Dowd M, McDonald J, Schuh J 2007 *The art of software security assessment: Identifying and preventing software vulnerabilities* (Mexico: Addison Wesley)