Security Maturity Model of Web Applications for Cyber Attacks

Renato Rojas
UPC
Av. Prolongación Primavera 2390
Lima, Peru
+51982604027,
renato2806@gmail.com

Ana Muedas
UPC
Av. Prolongación Primavera 2390
Lima, Peru
+51955528431
muedasanacristina@gmail.com

David Mauricio
UNMSM, UPC
Av. Prolongación Primavera 2390
Lima, Peru
+51971509987
pcsidmau@upc.edu.pe

ABSTRACT

Bearing in mind that the projections made for the area of information security point to an increase in attacks on the health sector, added to the lack or little diffusion of security maturity models that allow organizations to know the status of their website in terms of security and that the existing models lack a post-evaluation monitoring, it is necessary to propose a model of security maturity of web applications against cyber-attacks, oriented to the health sector, which is simple to apply. The proposed model will be based on the International Professional Practice Framework methodology and will include the main vulnerabilities published by the Open Web Application Security Project to propose attacks that identify the weakness of the evaluated web system, so that the client company has the possibility to reinforce its weaknesses. Guides will also be proposed to select strategies to improve critical points from a security perspective. As a result of the validation, it was found that, of the 14 tests applied, 5 were approved, positioning the web at level 3 of maturity, which means that there are validations in the structure of the web; however, they are partial or inefficient.

CCS Concepts

• Security and privacy~Penetration testing; • Security and privacy~Web application security; • Security and privacy~Access control; • Security and privacy~Authorization; • Security and privacy~Vulnerability scanners; • Security and privacy~Web protocol security; • Security and privacy~Social aspects of security and privacy

Keywords

Penetration tests; Vulnerability of the web application; Security information; Web attack; Cyber-attack; Cyber defense; Clinical records

ICCSP 2019, January 19–21, 2019, Kuala Lumpur, Malaysia

ACM ISBN 978-1-4503-6618-2/19/01…$15.00

DOI: https://doi.org/10.1145/3309074.3309096

1. INTRODUCTION

The growing competitiveness of the market generates an increasing difficulty in organizations to achieve success in their projects. This fact seeks to prioritize economic criteria, time, cost, quality and scope, which causes a lack of controls that result in security breaches in the company. In this way, security procedures such as web application testing are left in the background. These have vulnerabilities that can be used for malicious users to violate their protection mechanisms and gain access to private company information. According to an article published by SANS Institute, all organizations that maintain a presence on the web run the risk of being attacked, however, that danger is not prioritized, although security experts consider web-based attacks they are numerous, however, they are the least understood of all the risks related to confidentiality, availability and integrity.

According to predictions made by the market consultancy Gartner in 2015, by 2017, close to 25% of businesses would lose strength in terms of competitiveness as a result of not joining the trend of digital transformation [1]. Cyber security is among the 8 outstanding technologies for the year 2017, due to the increase in the number of high impact cyber security incidents, which causes the demand for security technologies and innovations [2]. According to a report issued by Reuters (2014), the private health sector has vulnerabilities in its major cyber security systems, compared to the financial and retail sectors, so according to the forecasts of the data breach industry, the industry Health will be the most sought after target for cyber-attacks in 2017, as the high value of electronic health records (EHRs) increasingly calls the attention of cybercriminals [3]. These records represent a greater source of income than if the card or bank account information was accessed. Stolen EHRs are used in multiple scams for longer periods of time, as they contain birth dates, social security numbers and a variety of health information, in addition to diagnostic codes, policy numbers and billing information.

The models of maturity for evaluation of web security existing in the market are scarce; in addition, they lack a post-evaluation accompaniment. The novelty of the proposed model is the possibility of a continuous re-evaluation flow with improvement process after each result obtained. A maturity model is proposed with a cyclic process of improvement validation. The model will allow the client to continue with the process of improvement and evaluation as many times as necessary until reaching an acceptable level for him. The process will determine the status of the web in terms of security through a series of tests and will cover the vulnerabilities found. To this end, controls proposed by OWASP have been used, where the points to be improved are addressed after an iteration in the maturity model and pentesting methodologies to determine the vulnerabilities of web applications.

The first section reviews the literature, where information is collected referring to articles published in the scientific community, all related to web security. In the second section, the proposed model will be explained. In the third section, the validation of the maturity model will be demonstrated. Finally, the conclusions of the proposed model will be made and a set of good practices and recommendations will be granted.

2. LITERATURE REVIEW

In [4] the problem of automatic classification of malicious web sessions is addressed. To allow realistic studies of the activities of the attackers, the honeypots must be executed with typical configurations and fully functional systems. As a result, it is concluded that web-based honeypots must be publicized to allow the use of search-based strategies, which seem to dominate the way attackers reach web servers. In [5] studied a special class of SQL injection attacks (SQLIA) where attackers can deduce the contents of the database by inspecting only the differences between the responses. The results show that the sequential search is the slowest, while the binary search and the bit by bit extraction are the fastest methods in case of a blind injection. [6] Analyzes the main web attacks in different categories. We identified studies where taxonomies related to web applications are established. In addition, study methods for web attacks such as honeypots are provided. It is concluded that in order to minimize the occurrence, the user must be educated and go beyond strengthening the application. [7] Performs a systematic review of the current state of research in the security of the web service to identify in which aspects of web service security current research is concentrated. It was found that 55.56% of the studies focus on web service attacks and 16.67% on vulnerabilities. For the authors, the result is positive, since not only must contingency measures be taken after an attack has occurred, but enough research must be done on how to prevent it.

In [8], two techniques are compared to analyze the presence of vulnerabilities in web services, the soapUI vulnerability scanner and the WSInject fault injector. It is concluded that the use of WSInject, in comparison with soapUI, improves the detection of the vulnerability, allows emulating the Cross-site Scripting (XSS) attack and generates variants. In [9] they propose a generic approach to design vulnerability testing tools, thus comparing the proposed tools with commercial scanners. The proposed tools present better results, in terms of coverage and false positives, surpassing any of the commercial tools. [10] Conducted a penetration test in a live production environment with the intention of collecting and analyzing the MITM attack traffic. It was found that ARP spoofing is more likely to change the header fields of the packet, resulting in less semi-duplicated pair. [11] presents VulScan, a web vulnerability scanner. VulScan discovered private SQL filtering and XSS vulnerabilities in some real websites protected by filters and WAF that could not be detected by a popular web vulnerability scanner. [12] Presents TestREx, a framework that allows automated and repeatable exploit tests in contexts. One will have to become familiar with all these tools and perform the experiments manually. [13] Presents security tests based on knowledge of web applications, a novel method to detect existing SQLI and XSS vulnerabilities in web applications.

Young-SuJan [14] presents a technique that modifies applications to protect against SQLIA. By applying its approach, new vulnerabilities are found that result from incorrect or incomplete disinfection. The main finding is that by using a substitute input variable and a disinfection based on the size of the query, it is possible to detect and prevent SQL queries that include injection vulnerabilities. Koning proposes CoreFlow and Bro data tools for data aggregation and Route Explorer for route calculation. CoreFlow focuses on the correlation of events already identified, through the use of data sources, such as NetFlow to create a more complete view of what happened and improve decision making. In the test it is concluded that it can lead to better alerts, however, it is necessary to investigate how to act on this new information obtained [15]. Kar presents a novel approach to detect SQL injection attacks, so, model SQL queries as a token graph. The approach was designed to work in the firewall layer of the database and was implemented in a prototype called SQLiGoT. The results obtained in five web applications show that they are completely vulnerable and confirm the effectiveness of the approach [16]. In [17] it proposes a framework for analyzing web application code and verifies unsafe information flows. The proposed framework was validated and tested in PHP web applications. As a result, the system's high success rate was shown in terms of detecting security failures at its source and in terms of reducing the false alarms reported. Razzaq proposes a method to detect and classify web application attacks. The proposed system was compared with other open source systems, such as ModSecurity and Profense. A generic rule generation mechanism could be created that provides interoperability with existing security solutions [18]. Bo Li presents a work that detects malicious network activities based on the analysis of web records. A multi-tasking learning approach, called MTLID, is proposed to detect web attacks. [19] Deepa investigates, from the injection vulnerabilities, the business logic. There are several approaches available to protect SQLI and XSS web applications, these attacks still prevail due to their impact and severity. There is no single solution to mitigate all defects. More research is needed in the area of correcting flaws in the source code of applications [20].

In [21] an information security maturity model based on ISO 27001 for software developers is presented. Its detailed construction is indicated in phases and then its evaluation by experts in the field. To evaluate the level of maturity, 9 Brazilian companies were used. The results of the evaluations showed that the model is a tool that can be used for companies to implement information security processes. In [22] it investigates existing models of security maturity, focusing on their characteristics and identifying their strengths and weaknesses. Finally, the document discusses and suggests measures for a robust and applicable cybersecurity model. [23] presents a maturity model for cyclical evaluation of information security. The model presents a series of steps to follow to obtain maturity periodically and improve this aspect through controls. Its maturity model consists of 5 levels that indicate the progress of security from the points described in ISO 27002 as Security policy, Organizational information security, Asset management, Human resources security, among others. It was determined that the model not only serves as an evaluation for SI, but as a system to manage security in it. [24] Seeks to determine which are the most widely used cybersecurity maturity models, for this purpose carried out a systematic review identifying the main models used in cybersecurity. All the models take as reference the CMM model; Current cybersecurity models are adaptations. It is concluded that the number of models mentioned by the authors is very small, thus, it is identified that there is a field that is not exploited at present on cybersecurity.

3. PROPOSED MATURITY MODEL

Considering that the International Professional Practice Framework (IPPF) methodology, among the other methodologies investigated, is the one that provides more details on how to build a maturity model, was chosen as the basis for constructing the proposed security maturity model. For this model, the vulnerabilities published in the Open Web Application Security Project (OWASP) are considered, a project that makes an application security standard that covers most common attacks and threats and is used to build several phases of the model, which they will be detailed below.

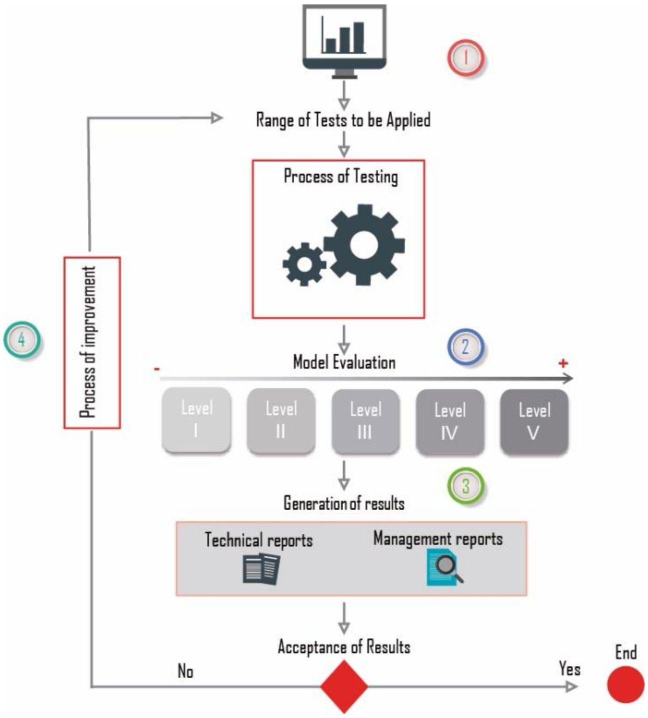


Figure 1. Web Applications Security Maturity Model

The proposed web application security maturity model includes the main vulnerabilities published by OWASP and proposes attacks to identify the weakness of the evaluated web system. The attacks are treated as penetration tests; each attack has a weight translated into a score for the test.

The model is made up of 4 phases: Tests to be carried out (1), Evaluation of the model (2), Generation of results (3) and Process of improvement (4). Its operation can be cyclical in the case that it is required to evaluate the web more than once.

To begin the evaluation process, the entire range of proposed tests is applied to the analyzed website (1). In the first phase it is necessary to mention that there are two possible scenarios: the page could apply for all the tests, as well as having some tests that are not applicable; for example, if the page was not created with a data serialization framework, the insecure Deserialization vulnerability could not be applicable. When this type of situation occurs, a full score will be awarded, since it is considered that, if the scenario could not be proven, the page would be protected against that attack, because it would not materialize.

After completing the tests, the results are evaluated and the level of the web analyzed is revealed (2).

Technical and managerial reports are generated where the situation found in the web application is indicated. These reports will allow the manager of the company to decide what action to take. If they are satisfied with the level of safety achieved, they will not carry out the recommendations offered in the improvement guide and will complete their evaluation cycle by model (3).

On the other hand, if the company wishes to improve its security level, it will enter into an improvement process in which it will follow the recommendations offered in the guide and will be evaluated again (4), and will be consecutively until reaching a desired level.

3.1 Phase 1: Testing Process

The tests applied in the proposed model come from OWASP publications in "Owasp top 10". By consolidating publications for 3 consecutive years, it was possible to identify the most recurrent attacks and vulnerabilities in the field of web pages, which will be used by the proposed maturity model. Next, the range of tests to be performed is indicated.

Table 1. Test matrix

Tests	Goals
Injection	Search for fields that interpret SQL queries
Loss of Access Control	Use functions that should not be accessible
URL access restriction failure	Access to unauthorized sections of the web through URL
Cross-site XSS script	Enter scripts to alter the behavior of the web
Exposure of sensitive data	Determine if sensitive data is adequately protected
Loss of authentication and session management	Partial or complete loss of keys, session tokens
Insecure deserialization	The web can interpret serialized information
Insufficient protection in the transport layer	Data transmitted as simple text in the transport layer
Non-validated redirects and resends	By means of a script, the sending of information to unsafe sites is altered
Unsafe cryptographic storage	Use of insecure algorithms for sensitive data and passwords
XML External entity (XXE)	Alteration of web operation by means of XML code entry
Insufficient Registry and Monitoring	Determine changes or attacks on the web in a set period
Falsification of cross-site requests CSRF	Force the use of unauthorized functions for the current user
Defective security settings	Use of non-configured security components
Use of components with known vulnerabilities	Use of components with unpatched CVE's

3.2 Phase 2: Evaluation

After applying the tests, the evaluation phase begins, where a score is assigned as a result of each attack. The assignment of scores to each concept follows the OWASP Risk Rating Methodology

(OWASP), which sets a score for each concept. The mentioned in table 2 is detailed

Table 2. Scoring system for evaluation concepts

Concept	Description	Score	Description
Reason	Motivation of attackers to exploit the vulnerability	1	Low or no reward
		4	Possible reward
		9	High reward
Opportunity	Resources and opportunities required to exploit vulnerability	0	Total access or expensive resources needed
		4	Special access or expensive resources needed
		7	Access or necessary resources
		9	Without access or necessary resources
Skill	Skill Necessary rating of attackers	1	Without technical skills
		3	some technical skills
		5	Advanced computer user
		6	Network skills and programming
		9	Security penetration skills
Size	How big is the group of attackers	2	Developers
		2	System administrators
		4	Intranet users
		5	Partners
		6	Authenticated users
		9	Anonymous Internet users

In Table 3, to qualify each test taking into account the concepts in Table 2, information is obtained on the impact of the attack on OWASP publications.

Table 3. Testing Quantification

ID	Vulnerability / Attack	Ability	Motive	Opportunity	Size	Score	Normalized Score
T1	Injection	3	9	9	9	7.5	10.3
T2	Loss of Access Control	5	1	7	6	3.5	4.8
T3	URL access restriction failure	5	1	7	6	3.5	4.8
T4	Cross-site XSS script	3	4	7	9	5.8	8
T5	Exposure of sensitive data	6	9	4	9	7	9.6
T6	Loss of authentication and session management	3	9	7	6	6.3	8.7
T7	Insecure deserialization	3	4	7	6	5	6.9
T8	Insufficient protection in the transport layer	6	4	4	6	5	6.9
T9	Non-validated redirects and resends	5	4	4	6	3.5	4.8
T10	Unsafe cryptographic storage	3	9	7	4	5.8	8
T11	XML External entity (XXE)	6	9	4	4	5.8	8
T12	Insufficient Registry and Monitoring	6	4	4	4	4.5	6.2
T13	Falsification of cross-site requests CSRF	5	4	4	5	3.3	4.5
T14	Defective security settings	5	4	4	5	3.3	4.5
T15	Use of components with known vulnerabilities	5	4	7	2	3.3	4.5

Total 72.8

As mentioned above, it can be the case that a test cannot be applied on the web, in the case of that scenario is assigned a full score, since it is considered to be resistant to vulnerability. For the cases where the tests are applicable, if the page approves the attack, it is assigned the complete score (see table 3), in case it does not approve, it is assigned 0 (zero). The final score of the test is divided by the sum of the maximum scores of the tests (a transformation is performed so that the value of the total sum (72.8) equals 100 and thus dividing that value in the 5 proposed levels. performed so that the maximum score is equivalent to a scale of 100 as maximum sum). The sum of the normalized scores obtained by each test is obtained and the final level is obtained, which serves as an indicator of how prepared the web is against latent vulnerabilities in thousands of web pages.

The formula to define the level achieved by the web, both for cases where not all tests are applicable and in which all the tests will be developed, is shown below, and is applied in the indicated sequence:

$$\text{Score} = \frac{\text{Ability} + \text{Motive} + \text{Oportunity} + \text{Size}}{4}$$

$$\text{Normalized Score} = \frac{100}{72.8}\text{Score}$$

$$\text{Final Score} = \text{E Normalized Score}$$

The final score will be a value between 0 and 100 and will place the web in one of the 5 levels mentioned below in table 4.

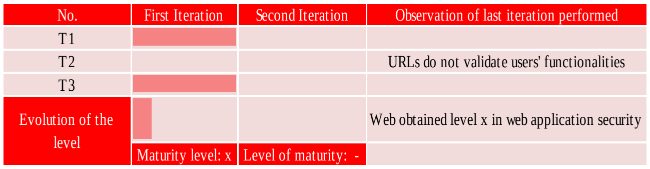
Table 4. Maturity Levels

Level	Name	Description	Final score range
1	Incipient	At this point the web is vulnerable more than 80% of known attacks	0-20
2	Basic	Few controls are taken into account for web components	21-40
3	Intermediate	There are validations; however, they are partial or inefficient.	41-60
4	Strategic	Has managed the risks that involve the strategic processes of the company	61-80
5	Optimized	The risk of the tests recognized by OWASP has been reduced	81-100

3.3 Phase 3: Generation of results

In this phase, the results of phase 2 are compiled to be presented in the form of reports. They detail the results of each test and the vulnerability that a cyber-attack can suffer from. Taking these points in terms of risk to the organization, improvement decisions can be made more effectively.

The technical results of the tests are shown in figure 2, where the procedure of the same is detailed, as well as the tools used during its execution. In addition, it works as an outlet for the improvement process. The technical report that would be generated after evaluating a client's website is shown.



No.	First Iteration	Second Iteration	Observation of last iteration performed
T1	🟥		
T2			URLs do not validate users' functionalities
T3	🟥		
Evolution of the level	🟥		Web obtained level x in web application security
	Maturity level: x	Level of maturity: -	

Figure 2. Technical report with consolidated tests

A management report is also generated where the vulnerabilities found in the form of risk for the organization are presented in figure 3 and 4, which indicates the cost of operating with the vulnerabilities found and how to determine the most relevant risks to be mitigated. For the prioritization of risks, the client company must:

1. Focus on the attacks marked in red that represent those disapproved in the validation.

2. As a next step, verify the impact of risk, the higher the impact, the more relevant it is to attend to the risk.

3. Classes of controls are classified as manual or automatic. The automatic ones have a higher cost.

4. Then the frequency of occurrence of that risk must be observed, while more frequent, it represents a higher cost.

5. Once the attacks to be mitigated have been selected, a monetary cost criterion is entered.

Table 5. Management Report with the risks of the tests

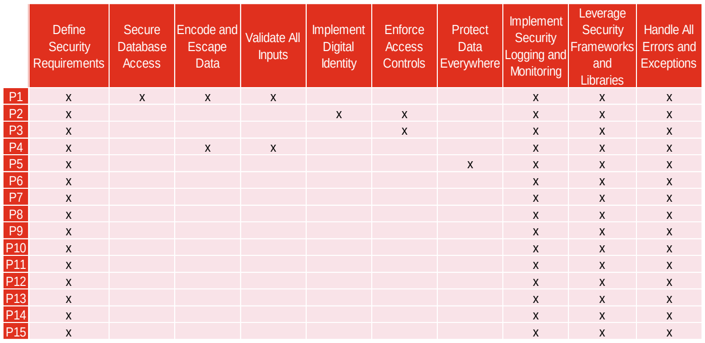
No.	Consequences	Probability	Impact	Inherent Risk
T1	Read sensitive information, make changes or even delete this information	Almost sure	Moderate	Extreme

Table 6. Management Report with the risks of the tests

No.	Controls to Apply	Impact on Risk	Control class	Frequency	Cost mh	Software cost
T1	Parameterize queries	15	Manual	When it's requested	72hh	X
T2	Verify early safety and often	4	Automatic	Daily	2hh	X

3.4 Phase 4: Process of improvement

After obtaining the level of maturity of the website, recommendations are generated from controls suggested by OWASP. Controls will be applied if the test is not satisfactory. In table 5, the control is crossed with the test, for example, in the case of not approving the injection test, it will be recommended to comply with the related controls, such as "validate all entries", which indicates having the necessary considerations for the information data entries.



	Define Security Requirements	Secure Database Access	Encode and Escape Data	Validate All Inputs	Implement Digital Identity	Enforce Access Controls	Protect Data Everywhere	Implement Security Logging and Monitoring	Leverage Security Frameworks and Libraries	Handle All Errors and Exceptions
P1	x	x	x	x				x	x	x
P2	x				x	x		x	x	x
P3	x					x		x	x	x
P4	x		x	x				x	x	x
P5	x						x	x	x	x
P6	x							x	x	x
P7	x							x	x	x
P8	x							x	x	x
P9	x							x	x	x
P10	x							x	x	x
P11	x							x	x	x
P12	x							x	x	x
P13	x							x	x	x
P14	x							x	x	x
P15	x							x	x	x

Figure 3. Controls proposed by OWASP for vulnerabilities

4. VALIDATION

4.1 Organization

The validation of the model was carried out in an internationally recognized health sector company. This company is responsible for providing hospital, pharmaceutical and medical management software solutions. The website used as a validation scenario belongs to that company and pools information on more than 20 health entities among clinics and hospitals, which have diversified and structured care capable of handling up to 20 medical specialties. This fact allows the website to store a large number of clinical histories, which makes it a well-liked target for cybercriminals.

Due to the increasing rise of cyber-attacks to health organizations the companies needs to measure the security of its website, since the flank of web applications tends to be a vulnerable and simple point of attack for cybercriminals. That is why they need to identify the level of security of their website and the vulnerabilities that it has.

4.2 Application of Model

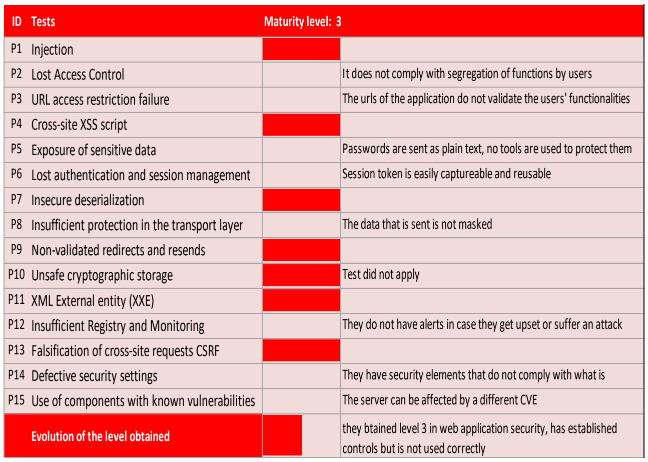
In the application procedure of the model, it is necessary to determine the tests that can be performed on the evaluated web. After having the relationship of these, we proceed to the execution of attacks on the modules that make up the web. As a second step, the level of maturity of the web application is calculated and the results of the tests are entered in the technical report. In addition, the necessary controls are determined to improve the level of security in the management report.

4.3 Model Phases

In phase 1 of the evaluation process proposed by the security maturity model, the tests that should be applied to the web were determined. Of the tests proposed by the model, the test "P10-Unsafe cryptographic storage" could not be applied, because the web does not have its information encrypted, thus, it was not possible to evaluate whether the encryption is safe or not.

4.4 Result

Phase 2 of the model allows evaluating the results obtained in each test and determining at what level of maturity is the web. Of the 14 tests applied, it is observed that the page approved 5 tests. Next, we have the findings obtained as a result of the attacks figure 4.



ID	Tests	Maturity level: 3	
P1	Injection	🟥	
P2	Lost Access Control		It does not comply with segregation of functions by users
P3	URL access restriction failure		The urls of the application do not validate the users' functionalities
P4	Cross-site XSS script	🟥	
P5	Exposure of sensitive data		Passwords are sent as plain text, no tools are used to protect them
P6	Lost authentication and session management		Session token is easily captureable and reusable
P7	Insecure deserialization	🟥	
P8	Insufficient protection in the transport layer		The data that is sent is not masked
P9	Non-validated redirects and resends	🟥	
P10	Unsafe cryptographic storage	🟥	Test did not apply
P11	XML External entity (XXE)	🟥	
P12	Insufficient Registry and Monitoring		They do not have alerts in case they get upset or suffer an attack
P13	Falsification of cross-site requests CSRF	🟥	
P14	Defective security settings		They have security elements that do not comply with what is
P15	Use of components with known vulnerabilities		The server can be affected by a different CVE
	Evolution of the level obtained	🟥	they btained level 3 in web application security, has established controls but is not used correctly

Figure 4. Result security test / tests

It was identified that the web is at level 3 security maturity, which means that there are validations in the structure of the web, however, are partial or inefficient.

Technical and managerial reports were issued, as indicated in phase 4 of the model; these reports were exposed to the board of directors of the company, which decided to maintain the level reached by not applying the proposed controls to obtain a higher level of security in a subsequent evaluation.

Table 7. Management Report with the risks of the tests

No.	Consequences	Probability	Impact	Inherent Risk
T1	Read sensitive information, make changes or even delete this information	Almost sure	Moderate	Extreme
T2	Unauthorized access, information theft	Rare	Less	Low
T3	Loss of sensitive information	Rare	Less	Low
T4	Alteration in the normal flow of the page	Almost sure	Moderate	Extreme
T5	Read sensitive information, make changes or even delete this information	Almost sure	Moderate	Extreme
T6	Unauthorized access, information theft	Almost sure	Moderate	Extreme
T7	Read sensitive information, make changes or even delete this information	Unlikely	Moderate	Moderate
T8	Read sensitive information, make changes or even delete this information	Unlikely	Moderate	Moderate
T9	Alteration in the normal flow of the page	Rare	Less	Low
T10	Loss of sensitive information	Probable	Moderate	High
T11	Alteration in the normal flow of the page	Probable	Moderate	High
T12	Loss of sensitive information	Unlikely	Moderate	Moderate
T13	Alteration in the normal flow of the page	Rare	Less	Low
T14	Read sensitive information, make changes or even delete this information	Rare	Less	Low
T15	Unauthorized access, information theft	Rare	Less	Low

5. CONCLUSION

In the present work a security maturity model was developed that allows to indicate the level of protection against cyber-attacks from a health sector website. This model is simple to apply, and can be implemented by a person with basic knowledge of information security. Through the test guide, it is possible to apply them in 6 hours and generate the results automatically, entering the values obtained in the technical report file, which added with the management report, will allow to create decision criteria on implementing the proposed changes. The improvement guide or finish the evaluation of the web with the result of only one iteration.

With the validation in the proposed scenario it was possible to identify that the website is at an intermediate security level (level 3), since it was possible to enter reserved sessions for web administrators and visualize sensitive and personal information of doctors and patients, There are validations, for example, of data entry, however, they are partial or inefficient. During the execution of the tests, it was necessary to have the support of a staff capable of identifying the segregation of functions for the different types of users who use the web, in order to determine up to what level of information could be reached with an account and analyze if that level was contemplated for the use of the user's role. The validation process clearly and truthfully demonstrates the security flaws. The biggest faults are in the logic of the application's functionalities and in the delimitation of the access scope for each user role on the web.

Taking into account that the model fits into a field that changes with speed, it is important to keep it updated with the new trends of cyber attacks. That is why we conclude that the model can be enriched by studies that complement the methodologies of choice of penetration tests dictated by the OWASP, the knowledge can be used other testing methodologies such as PTES or OSSTMM. In addition, to implement techniques that reduce the cases of false positives during the penetration tests.

Table 8. Management Report at the level of decision criteria

No.	Controls to Apply	Impact on Risk	Control class	Frequency	Costo hh	Software Cost
T1	Parameterize queries	15	Manual	When it's requested	72hh	x
T2	Verify early safety and often	4	Automatic	Daily	2hh	x
T3	Take advantage of security frameworks and libraries	5	Manual	Sporadic / Surprise	72-144hh	x
T4	Implement intruder registration and detection	12	Automatic	Daily	x	$2000
T5	Protect the data	14	Manual	Permanent	144hh	x
T6	Verify early safety and often	13	Automatic	Daily	2hh	x
T7	Take advantage of security frameworks and libraries	9	Manual	Sporadic / Surprise	72-144hh	x
T8	Implement intruder registration and detection	8	Automatic	Daily	x	
T9	Handling errors and exceptions	6	Manual	Permanent	72-144hh	x
T10	Verify early safety and often	11	Automatic	Daily	2hh	x
T11	Implement intruder registration and detection	10	Automatic	Daily	x	$2000
T12	Handling errors and exceptions	7	Manual	Permanent	72-144hh	x
T13	Take advantage of security frameworks and libraries	1	Manual	Sporadic / Surprise	72-144hh	x
T14	Verify early safety and often	2	Automatic	Daily	2hh	x
T15	Handling errors and exceptions	3	Manual	Permanent	72-144hh	x

1. $Cost\ hh = \$20$

6. ACKNOWLEDGMENTS

The authors thank UPC for the partial financing of this work.

7. REFERENCES

[1] Perez, M. (8 de marzo de 2017). Pymes a tiempo para comenzar su digitalización. Recuperado el 26 de agosto de 2017, de The e Mag: https://www.the-emag.com/theitmag/2017/03/08/pymes-a-tiempo-para-comenzar-su-digitalizacion

[2] Castilla, A. (9 de marzo de 2017). 10 TENDENCIAS TECNOLÓGICAS DEL GRUPO GARTNER PARA 2017. Recuperado el 26 de agosto de 2017, de Ciencia y Tecnología: https://cienciaytecnologia.fundaciontelefonica.com/2017/03/09/10-tendencias-tecnologicas-del-grupo-gartner-para-2017/

[3] REUTERS. (25 de septiembre de 2014). Historias clínicas, el nuevo objetivo de los piratas informáticos. Recuperado el 26 de agosto de 2017, de El tiempo: http://www.eltiempo.com/archivo/documento/CMS-14593475

[4] Goseva-Popstojanova, K., Anastasovski, G., Dimitrijevikj, A., Pantev, R., & Miller, B. (2014). Characterization and classification of malicious Web traffic. Computers & Security, 92-115.

[5] Stampar, M. (2016). Inferential SQL Injection Attacks. International Journal of Network Security, 316-325.

[6] Kaur, D., & Kaur, D. P. (2015). Empirical Analysis of Web Attacks. International Conference on Information Security & Privacy (págs. 298 – 306). Nagpur: Procedia Computer Science.

[7] Mouli, V. R., & K.P.Jevitha. (2016). Web Services Attacks and Security- A Systematic Literature Review. Procedia Computer Science, 870-877.

[8] Salas, M., & Martins, E. (2014). Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-Security. Electronic Notes in Theoretical Computer Science, 133-154.

[9] Antunes, N., & Vieira, M. (2017). Designing vulnerability testing tools for web services: approach, components, and tools. International Journal of Information Security, 435–457.

[10] Calvert, C., Khoshgoftaar, T. M., Najafabadi, M. M., & Kemp, C. (2017). A Procedure for Collecting and Labeling Man-in-the-Middle Attack Traffic. International Journal of Reliability, Quality and Safety Engineering.

[11] Huang, H.-C., Zhang, Z.-K., & Cheng, H.-W. (2017). Web Application Security: Threats, Countermeasures, and Pitfalls. IEEE Computer Society, 81-85.

[12] Dashevskyi, S., Santos, D. R., Massacci, F., & Sabetta, A. (2017). TestREx: a framework for repeatable exploits. International Journal on Software Tools for Technology Transfer, 1–15.

[13] Zech, P., Felderer, M., & Breu, R. (2017). Knowledge-based security testing of web applications by logic programming. International Journal on Software Tools for Technology Transfer, 1-61.

[14] Jang, Y.-S., & Choi, J.-Y. (2014). Detecting SQL injection attacks using query result size. Computers & Security, 104-118.

[15] Koning, R., Buraglio, N., Laat, C., & Grosso, P. (2018). CoreFlow: Enriching Bro security events using network traffic monitoring data. Future Generation Computer Systems, 235-242.

[16] Kar, D., Panigrahi, S., & Sundararajan, S. (2016). SQLiGoT: Detecting SQL Injection Attacks using Graph of Tokens and SVM. Computers & Security, 206-225.

[17] El-Hajj, W., Brahim, G. B., Hajj, H., Safa, H., & Adaimy, R. (2015). Security-by-construction in web applications development via database annotation. Computers & Security, 151-165.

[18] Razzaq, A., Latif, K., Ahmad, H. F., Hur, A., Anwar, Z., & Bloodsworth, P. C. (2014). Semantic security against web application attacks. Information Sciences, 19-38.

[19] Li, B., Lin, Y., & Zhang, S. (2017). Multi-Task Learning for Intrusion Detection on Web Logs. Journal of Systems Architecture, 92-100.

[20] Deepa, G., & Thilagam, P. S. (2016). Securing Web Applications from Injection and Logic Vulnerabilities:Approaches and Challenges. Information and Software Technology, 160-180.

[21] Silva, M. P., & Barros, v. M. (2017). Maturity Model of Information Security for. IEEE LATIN AMERICA TRANSACTIONS, 1994 - 1999.

[22] Le, N. T., & Hoang, D. B. (2016). Can maturity models support cyber security? Performance Computing and Communications Conference (IPCCC), 2016 IEEE 35th International.

[23] Rigon, E. A., Merkle Westphall, C., Dos Santos, D. R., & Becker Westphall, C. (2014). A cyclical evaluation model of information security maturity. Information Management & Computer Security, 265 - 278.

[24] Rea-Guaman, A., Sánchez-García, I., San Feliu, T., & Calvo-Manzano, J. (2015). Maturity Models in Cybersecurity: a systematic review. Technology and Society (ISTAS), 2015 IEEE International Symposium on (págs. 1-6). Dublin: IEEE.