Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Ingeniería Informática

Proyecto Final

Web platform for questionnaire-based DREAD risk analysis and penetration testing

Autor del proyecto: Esther Pérez Jerez

Director del proyecto: Jaime Font Burdeus

Zaragoza, 11 de septiembre de 2020

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Ingeniería Informática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma Fecha **11 de septiembre de 2020**

Dedicatoria y agradecimientos

A mis padres por apoyarme en todas mis decisiones por muy descabelladas que sean, al final no fue ni Oxford ni Harvard, fue Zaragoza. Acabo mi carrera donde empezó todo, no podía ser de otra manera.

A Jaime Font, director de este proyecto, por su ayuda y acompañamiento constantes y por confiar en mí, espero tener la oportunidad de volver a trabajar contigo de nuevo.

A mis amigas por sus incansables ánimos y por recordarme que soy más capaz de lo que creo cuando más lo necesito, incluso a kilómetros de distancia.

A Bryan Pérez por aportarme la calma que siempre me falta y por contagiarme de su dedicación y constancia que junto con su apoyo me han empujado a acabar este proyecto.

Table of contents

Table Index

Illustration Index

Resumen

Este proyecto se ha desarrollado como Proyecto Fin de Grado para la titulación "Grado en Ingeniería Informática" de la Universidad San Jorge con la colaboración de la empresa GEMED Soluciones.

La idea de este proyecto surge de la necesidad de mostrar a las empresas la importancia de la seguridad de la información y el peligro al que se exponen si no toman las acciones pertinentes. Por esto se desarrollan dos aplicaciones con la finalidad de proporcionar a las empresas de las herramientas necesarias para autoevaluarse y entender un poco mejor su nivel de seguridad a través de un Análisis de riegos y una Prueba de penetración.

El Análisis de riesgos consiste en la evaluación de diversas amenazas, divididas en cuatro pilares fundamentales, mediante una serie de cuestionarios con la aplicación de la metodología DREAD.

La Prueba de penetración consiste en el lanzamiento de ataques automatizado de algunas para las vulnerabilidades web más comunes contra la web del usuario. Además, cuenta con un apartado didáctico que contiene una colección de ejemplos de estos ataques con el fin de enseñar el funcionamiento de estos y las medidas de prevención necesarias para cada uno.

Abstract

This project has been developed as a Final Degree Project for the degree "Grado en Ingeniería Informática" of Universidad San Jorge with the collaboration of the company GEMED Soluciones.

The idea of this project arises from the need to show companies the importance of information security and the danger to which they are exposed if they do not take the relevant actions. For this reason, two applications are developed with the purpose of providing companies with the necessary tools to self-assessment and understand a little better their level of safety through a Risk Analysis and a Penetration Test.

Risk Analysis consists of the assessment of various threats, divided into four main pillars, through a series of questionnaires with the application of the DREAD methodology.

The Penetration Test consists of the launch of automated attacks of some of the most common web vulnerabilities against the user's web. In addition, it has a didactic section containing a collection of examples of attacks in order to explain the inner working of each attack and show the measures to prevent them.

1. Introduction

1.1. Gemed Soluciones

This project will be developed in conjunction with GEMED Soluciones [1], one of the enterprises that forms Grupo GEMED. GEMED Soluciones was born in 2011 from the Engineering area of GEMED Canarias S.L. (nowadays GEMED Suministros) which since 1998 has been present in the Canarian market as an engineering and technological company.

Their main activity is to offer services for Information Security assurance, which includes security convergence, ISMS (Information Security Management System) implementation, Monitoring via SOC (Security Operation Centre), Continuous training, Security audits, Red Team and SIEM (Security Information and Event Managements), among others.

At GEMED Soluciones, from now on GEMED, they understand safety from four fundamental pillars:

- Cybersecurity, which covers vulnerabilities of systems, computers, networks, mobile devices, architectures and services.
- Social engineering, which covers human factor vulnerabilities due to unsafe use of information system, breaches of security-related regulations, policies, procedures, etc.
- Information Security Management, which covers ISMS vulnerabilities by weak, deficient or non-existent policies, procedures, rules, records, etc.
- Business continuity, which covers vulnerabilities in contingency or recovery plans for security incidents due to deficiencies or lack thereof.

These four pillars might be balanced in order to maintain a good overall security. If one of these pillars decompensates it could lead to serious security problems because as it is commonly said in the field of cybersecurity "A system is as strong as its weakest link".

1.2. Project justification

Among small and medium-sized companies, the prevailing thought is that information security is something that is not within their reach and that only large companies can afford it. In addition, these companies usually also think that hackers won’t even be interested in them because they are small businesses that don’t have the impact that, for example, a multinational has, so they don’t need to worry about a possible threat.

This is the aim of GEMED to eradicate this thought about cybersecurity by bringing some knowledge about its importance. That's why we propose for this project the creation of a web application that anonymously and free allows anyone to self-assess, in a generic way, the security status of their company. The purpose of this application is to provide an overview and some awareness of the importance of the security of their companies, the real harm they might be exposed to and how easily it would be exploited. This project will consist in two different parts: Risk analysis application and a Penetration test application.

Risk analysis application will be a questionnaire-based application where customers will be asked to answer several questions related to their enterprise like, for example, how is their data stored or which password protocols do they have. These questionnaires will be evaluated with DREAD, a risk analysis evaluation method, created by Microsoft and used by GEMED, in order to assign a score to the security level of customer's enterprise.

Penetration testing application will be an educational tool for bringing some knowledge about the most common web vulnerabilities, how are they exploited and how to prevent them, also, in this tool there will be a section where users could execute an automated search to identify these vulnerabilities in their own website.

2. Context

In this chapter, we will do a short introduction of some key concepts, tools and methodologies that would be useful along the development of this project.

Information Security is a complex field about which many people do not have enough knowledge, that's why we are going to start by defining some key terms that we will find during this project.

Following this, we are going to introduce DREAD, the risk analysis methodology chosen by GEMED to evaluate their customers security level, and the OWASP, an organization focused on web vulnerabilities and their prevention.

Finally, we will discuss about some tools available on the Internet that will help us in the development of this project like some charts libraries and vulnerable websites to test our Penetration test application.

2.1. Key concepts

Attending to [2] we are going to introduce several concepts starting by information technology (IT) security which is the practice of defending digital information and IT assets against threats, this defense includes the detection, prevention and response to such threats.

We understand threat, as a potential breach of security, any circumstance or event that may adversely affect a system through unauthorized access, destruction, disclosure or modification of data, or denial of service. These threats may or may not be intentional: Intentional threat, a possibility of an attack by an intelligent entity; Accidental threat, possibility of human error or omission, unintentional malfunction of equipment or natural disaster.

A distinction needs to be made between threat and vulnerability; a threat is a potential hazard that could potentially exploit a vulnerability. Being a vulnerability, a defect or weakness in the design, implementation or operation and management of a system that could be exploited to violate the security policy of the system. Most systems have one or more vulnerabilities, but this does not mean that the systems are too defective for use. Not all threats result in an attack, and not all attacks are successful.

We understand an attack as an intentional act by which an entity attempts to evade security services and violate a system's security policy. That is, a real assault on system security that stems from an intelligent threat. The success of these attacks depends on the degree of vulnerability, the strength of the attacks and the effectiveness of any countermeasure in use.

One interesting topic we should also mention related to attacks is payload, which is the concrete component of an attack that has the goal to harm the victim.

Therefore, we perform what is known as Risk Analysis, an assessment process that systematically identifies valuable system resources and threats to those resources, quantifies exposures to losses based on estimated frequencies and costs of occurrence.

One of the most common methods for analyzing these risks is Penetration testing which consists of a system test in which evaluators attempt to circumvent the safety features of a system using tools and techniques available to adversaries. This evaluation can be carried out in different modalities depending on the knowledge about the victim:

- Black box, you only know what can be found out by anyone (e.g. googling).
- Grey box, you have some knowledge about the victim or its system.
- White box, when you know everything about the victim, or you have full access to their system (for example an open source site where the attacker has access to the source code of the website).

2.2. DREAD methodology

DREAD [3] model is a quantitative risk analysis form pioneering by Microsoft that involves rating the severity of a cyber threat. DREAD is an acronym of Damage, Reproducibility, Exploitability, Affected users and Discoverability which are the five key points to evaluate for each security threat.

This qualification system is used to identify, compare and prioritize risk's value of the different threats analysed. Final value is calculated as an average of the previous mentioned key points, which are scored from 1 to 10 and rated as showed in the following table:

Risk rating	Result
High	7 – 10
Medium	4 – 6
Low	1 – 3

Table 1 – Risk score rating for DREAD analysis

For making a more understandable explanation of this model and the different key points that need to be evaluated let's introduce a running example:

> *"Our company has no passwords policy. Our employees can choose whatever password they'd like to, no length, characters nor capitals restrictions are applied to them. There is neither password changes required at any time so employees can use the same password forever."*

So, let's analyse:

> *"What is the risk of an attacker finding out our passwords?"*

2.2.1. Damage Potential

At this key point it is analysed the impact of the vulnerability in case that it is exploited. This damage needs to be determined within three different aspects: confidentiality, integrity and availability. To evaluate this the following question needs to be answered: How much impact would this threat have, if it is exploited? following the criteria exposed on Table 2:

Score	**Situation**
0	No damage will occur.
3	Individual user data is compromised, affected or availability denied.
5	All individual tenant user data is compromised, affected or availability denied.
7	All tenant data is compromised, affected or availability denied.
7	Availability of a concrete component/service is denied.
8	Availability of all components/services are denied.
9	Underlying management and infrastructure data are compromised or affected.
10	Complete system or data destruction, failure or compromise.

Table 2 – Score assignment rules for Damage Potential

For our company in the concrete case that we want to analyse we would need to assign a mark of 10. In the worst case that our administrator user password is guessed, all the system will be affected, since this would give attackers full access to our system and data.

2.2.2. Reproducibility

This key point says nothing about the preconditions needed to launch the attack, just how well it works once you trigger it. So, the question here is "How difficult is this threat to be replicated?" Which evaluation criteria is shown in Table 3:

Score	Situation
0	Very hard or impossible, even for administrators.
5	Attack can be reproduced, may need to be an authorized user.
10	Unauthenticated users can trivially and reliable exploit it.

Table 3 – Score assignment rules for Reproducibility

In this concrete case, a brute force attack for trying to crack our administrator's password can be executed by anyone and, obviously, authentication is not needed. This is why in this point we will need to assign a score of 10, again.

2.2.3. Exploitability

At this key point, knowledge and preconditions needed to exploit the vulnerability and how likely is this threat to be taken advantage of? are measured and evaluated under the following criteria:

Score	Situation
0	Advanced programming and networking knowledge, with custom or advanced attack tools.
1	Even with direct knowledge of the vulnerability we do not see a viable path for exploitation.

2	Advanced techniques required, custom tooling. Only exploitable by authenticated users.
5	Exploit is available/understood, usable with only moderate skill by authenticated users.
7	Exploit is available/understood, usable by non-authenticated users.
10	Trivial, just a web browser is needed

Table 4 – Score assignment rules for Exploitability

There are a lot of free tools focused on password cracking available on the internet and, in our concrete case, as we mention in the previous section, authentication is not needed. Although this attack could be managed with just a web browser it's not the common way to proceed since it would need a lot of time and a lot of effort, this password cracking is normally executed through some preexistent tool and this is why we will assign a score of 7 in this point.

2.2.4. Affected users

This category is pretty self-explanatory, how many users would be impacted by this threat? And the following evaluation criteria:

Score	Situation
0	None of them
5	Some of them
10	All of them

Table 5 – Score assignment rules for Affected users

Focusing in our example and as we mentioned in previous sections if I were administrator's password gets cracked all our users would be affected, so a score of 10 would be assigned to this point.

2.2.5. Discoverability

This is the most controversial category, in fact, there is a variation of this system called as DREAD-D, which subtracts it, but is good information to have. At this category we measure how likely is this threat to be found out? And the score is assigned as follows:

Score	Situation
0	Hard or impossible, needs administrator permissions. Needs access to source code or similar to find it out.
5	Can be figured out by monitoring network traces or by guessing.
9	Vulnerability details could be found at public domain by using a search engine.
10	Information is visible at the web browser bar or in a form.

Table 6 – Score assignment rules for Discoverability

Regarding our example company situation, no having password protocols is not a public information, just our employees have knowledge about it, but it could be guessed by someone just by trying brute force attacks and that's why in this point we should assign a score of 5.

Now that we have finally assign a mark to each of the five key points, we can proceed with the calculation of the final DREAD score, that will be as follows:

$$DREAD\ Score = \frac{10 + 10 + 7 + 10 + 5}{5} = 8{,}4$$

A final score of 8,4 on the DREAD model would be classify as a high-risk threat, following *Table 1* criteria.

DREAD methodology pretends to evaluate a threat in a very generic way, but the main problem with this is to assign a value to each of its key points. It is such generic that sometimes it will be really hard to find an appropriate score for some of them.

2.3. Data visualization libraries

One of the most important aspects of this project is to allow our users to know the current state of their company's security at a glance. For this, we have chosen to represent the results of the risk analysis graphically. With this purpose, we carried out a small research to find different libraries that could help us with this task, and, in turn, could provide a good visual aesthetic to our website. The following alternatives were taken into account:

2.3.1. ChartJS

ChartJS is an open source *JavaScript* library for plotting charts, among its possibilities there are 8 different chart types, which are: bar, line, area, doughnut, pie, scatter plots, bubble and radar.

For the representation of these charts, *ChartJS* uses canvas, and these plots are responsive to screen resize to maintain the proportions of the data represented. In addition, all the plots are displayed with animations. Also, there is a wide amount of customizable options including colours, labels, event handling, and axis among others.

Its repository is updated regularly, which permits it to be kept up-to-date and maintain a really good compatibility with all the browsers in every moment.

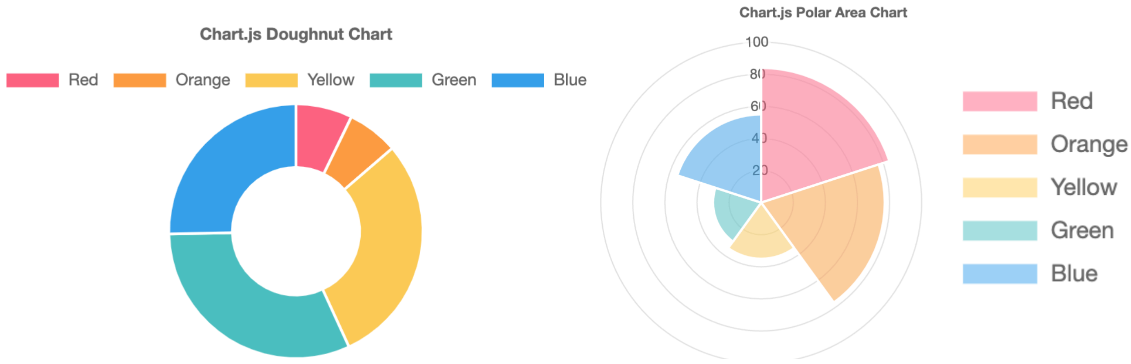


Illustration 1 - ChartJS example charts

2.3.2. Morris chart

Morris.js is an open source *JavaScript* and *CoffeeScript* library which can be used to represent line, area, bar and doughnut charts.

Representation with this library is performed using SVG and it is quite simple; we just need a call to the proper function of each chart type, which receives the chart's options as arguments. Among these options we can specify data, labels, colours, axis and other properties that vary depending on the kind of chart.

However, it seems that the repository has not been updated for the last 4 years, which could mean some compatibility problems somehow.

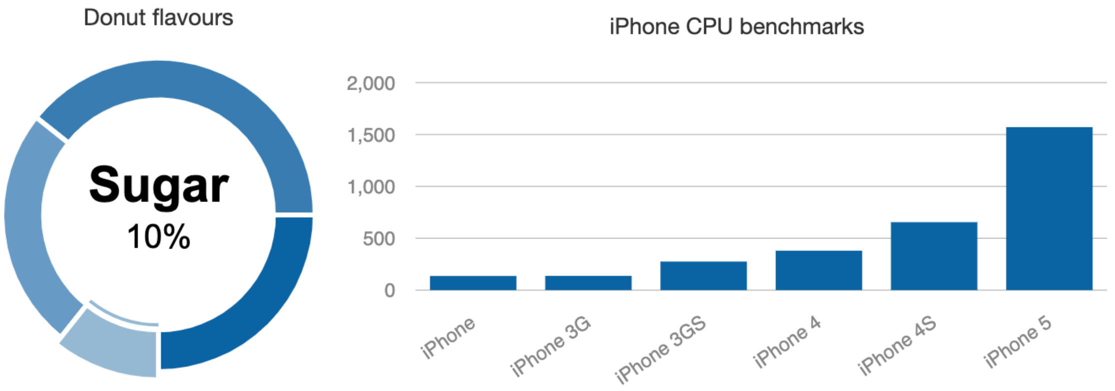


Illustration 2 - Morris.js example charts

2.3.3. Final decision

In the next table we can see a brief comparison between these two chart libraries:

Library	**Language**	**Graphic technology**	**Chart Types**	**Active Community**	**Price**	**Documentation**
ChartJS	JavaScript	Canvas	9 with multiple variations	Yes	Free	Well documented and up to date
Morris.js	JavaScript and CoffeeScript	SVG	4	No	Free	Little documentation about chart properties usage

Table 7 - Chart libraries comparation

Finally, we decided to implement the *ChartJS* approach due to its pretty charts styling but also for the customizable options that it provides. Although the code is larger than *Morris's* was easy to understand and modify to get done the features that we wanted to add, in a short period of time. Also, *ChartJS* is a more up-to-date project, which is still being updated while *Morris* last update was years ago.

2.4. OWASP

OWASP [4], Open Web Applications Security Project, is a non-profit foundation dedicated to elaborate and provide guides, tools and knowledge about security software, specifically for web applications.

Through its open source projects leaded by the community, being the reference in multiple educational and training conferences, OWASP foundation states the main guidelines for developers to secure the web.

One of the main projects of this organization is the OWASP Top Ten, which is the one we are going to focus on for the implementation of this project.

2.4.1. OWASP - Top Ten

OWASP Top Ten [5] project is a consciousness document for web application security. At the moment of the elaboration of this project the latest version is 2017, which gathers the most critical web security risks discovered up to that year.

1. Injection, has been as the major security risk since 2010 edition, occurs when unreliable data is sent to an interpreter as part of a command or query with the intention of executing it, for example, to access data without authorization. There are several types depending on the underlaying technology used to build the web: SQL, NoSQL, OS, and LDAP injection.

2. Broken authentication, takes place when keys, passwords or session tokens are compromised allowing attackers to steal other users' identity.

3. Sensitive Data Exposure, sensitive data should not be kept as far as possible, if it needs to, some protection is needed such as encryption at rest or in transit, and requires special precautions when exchanged with the browser, although it may be compromised or stolen.

4. XML External Entities, many older XML processors allow the specification of an external entity, a URI without reference and evaluated during XML processing. This failure allows data extraction, remote code execution, internal port scanning, and denial of service attacks among others.

5. Broken Access Control, users' permissions are often not properly constrained so attackers can exploit this situation to access unauthorized functionality and/or data, like accessing other users' account, alter other users' data or modify access rights.

6. Security Misconfiguration, commonly occurs due to default, incomplete or ad hocs configurations, misconfigured HTTP headers, open cloud storage and verbose error messages containing sensitive information.

7. Cross-Site Scripting (XSS), takes place when untrusted data is included into a web page without proper validation or escaping. Attackers add scripts before data is sent and executed at users' browser to hijack sessions, redirect to malicious sites or deface pages.

8. Insecure Deserialization, regularly ends in remote code execution or they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

9. Using components with known vulnerabilities, frameworks, libraries and other software modules run with the same privileges as the application. If any of them is vulnerable, could be exploited and used to facilitate serious data loss or server takeover.

10. Insufficient Logging and Monitoring, improperly recording flaws, lack of alerts and blockages allow the attacker to loop through vulnerabilities until one is exploitable.

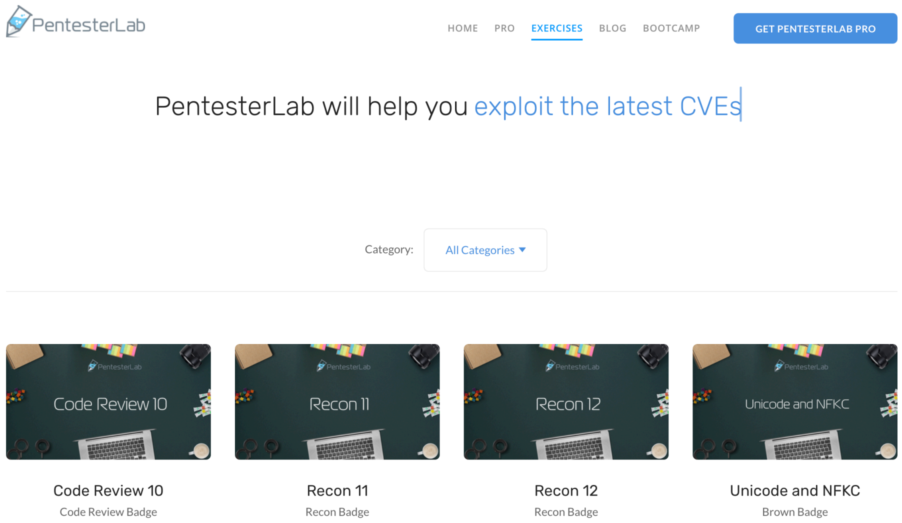
This top ten will be used as a basis to know what type of attacks we can implement and test with our penetration testing application.

2.5. Vulnerable websites

In this section we will analyze different websites that deliberately present some of the vulnerabilities that we have just mentioned. The goal of these vulnerable websites is purely didactic and are focused on ethical hacking. In our case we analyze these websites in search of potential "victims" in order to test the vulnerability tool safely and legally. Moreover, taken as a whole, they will allow us to understand the various ways in which a vulnerability can be exploited in order to be able to abstract from such attacks in order to generalize them and thus give our tool greater reach.

2.5.1. PentesterLab

PentesterLab [6] is a web resource that consists of a large compilation of exercises with various topics within computer security but focused on penetration testing. Each of these exercises is formed by an *iso* image, where the victim can be launched, and a bunch of videos and hints in order to be able to exploit whatever vulnerability is the exercise about.



PentesterLab will help you exploit the latest CVEs

Code Review 10
Code Review Badge

Recon 11
Recon Badge

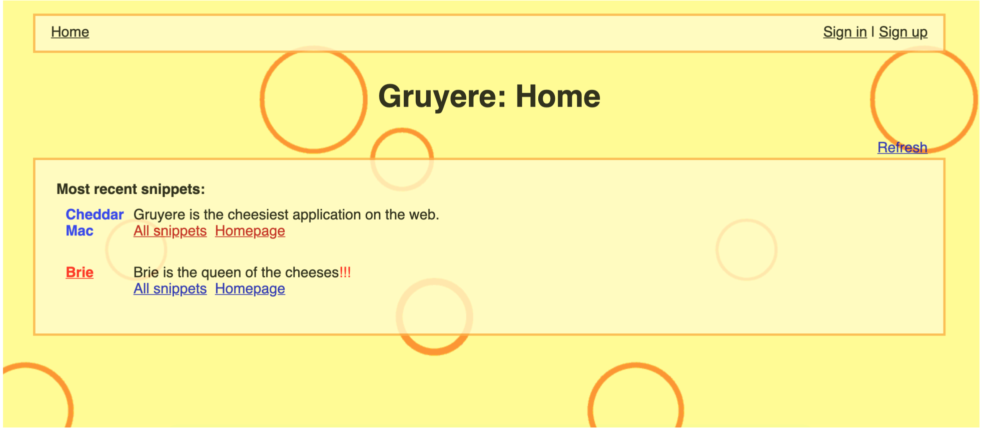
Recon 12
Recon Badge

Unicode and NFKC
Brown Badge

Illustration 3 - PentesterLab main page

2.5.2. Google Gruyere

Google Gruyere [7] is a free code lab focused on web application vulnerabilities, how to exploit them and how to protect against them. It gives their users the possibility of learning about these vulnerabilities by attacking a real application. It also offers the chance to see the source code of its page giving its users to attack these vulnerabilities in two ways: black box or white box.



Gruyere: Home

Most recent snippets:

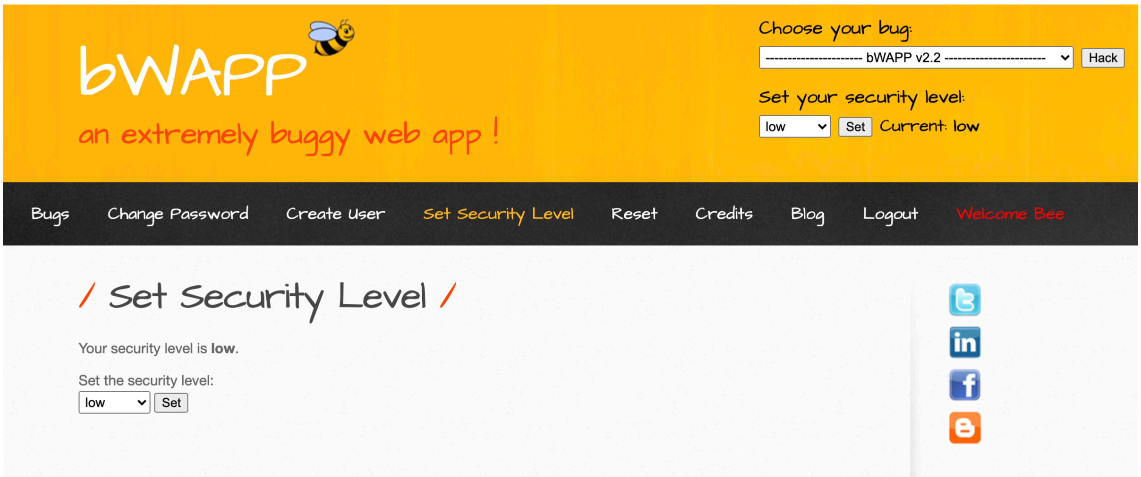
Cheddar Mac Gruyere is the cheesiest application on the web.

Brie Brie is the queen of the cheeses!!!

Illustration 4 - Google Gruyere main page

2.5.3. bWAPP

bWAPP [8] is a free and open source deliberately insecure web application. It is programmed in PHP with a MySQL database connection and covers all the risks listed in the OWASP Top Ten project. It needs to be launched at a local server such as Apache or with XAMPP.



/ Set Security Level /

Your security level is **low**.

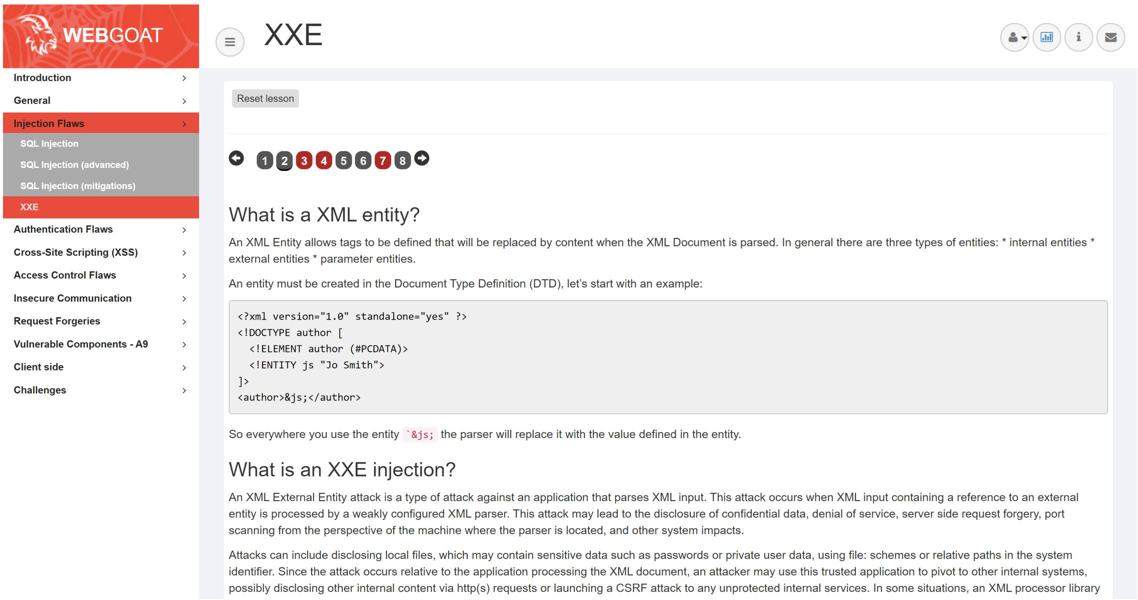
Set the security level:

low Set

Illustration 5 - bWAPP main page

2.5.4. WebGoat

WebGoat [9] is a deliberately insecure web application, created and maintained by OWASP, for testing Java-based applications vulnerabilities that use common and popular open source components. This program is a demonstration of common server-side application flaws. The exercises are intended to be used by people to learn about application security and penetration testing techniques.



What is a XML entity?

An XML Entity allows tags to be defined that will be replaced by content when the XML Document is parsed. In general there are three types of entities: * internal entities * external entities * parameter entities.

An entity must be created in the Document Type Definition (DTD), let's start with an example:

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE author [
  <!ELEMENT author (#PCDATA)>
  <!ENTITY js "Jo Smith">
]>
<author>&js;</author>
```

So everywhere you use the entity `` `&js; `` the parser will replace it with the value defined in the entity.

What is an XXE injection?

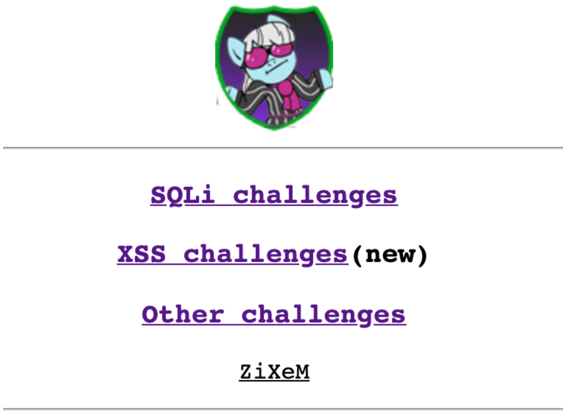
An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.

Attacks can include disclosing local files, which may contain sensitive data such as passwords or private user data, using file: schemes or relative paths in the system identifier. Since the attack occurs relative to the application processing the XML document, an attacker may use this trusted application to pivot to other internal systems, possibly disclosing other internal content via http(s) requests or launching a CSRF attack to any unprotected internal services. In some situations, an XML processor library

Illustration 6 - WebGoat main page

2.5.5. *Zixem*

Zixem is a vulnerable webpage focused on two types of attacks: SQL injections and Cross-Site Scripting. Currently, they are working in adding some other challenges which at the moment are based on remote code execution. There is no source code available and it can be easily accessed through any web browser. It also presents some "rules" like, for example, SQL injections are limited to "union" based statements.



SQLi challenges

XSS challenges(new)

Other challenges

ZiXeM

Illustration 7 - Zixem main page

3. Objectives

The main objectives that were defined for this project, which are included in the proposal for this [Annex 1], are:

1) Creation of a web application that conducts customer surveys and adapts them based on the responses obtained.

2) Risk analysis with DREAD methodology of the responses obtained.

3) Creation of a specific tool for website vulnerability analysis.

4) Present the results of the previous analysis to the client as well as certain tips or recommendations for improvement.

At the beginning of the project these objectives were analyzed, and project scope was defined. During the project development we considered that the previous objectives were adequate but decided to add a fifth objective:

5) Modified Objective 3, so that the Risk analysis application could have a didactic component, could grow dynamically and include the possibility of performing penetration tests automatically.

4. Methodology

This project is developed for Final Degree Project, a subject estimated in 12 ECTS [10] credits, which means approximately 320 hours of work to be carried out by, in this case, a single person. There is a delivery date for this project, and it is September 11th, 2020, so the flexibility of the iterations has a limit.

This project is combined with other subjects, so full time cannot be devoted to its development. The availability to work on this project will change from week to week depending on the external workload at the time.

Taking into account the described characteristics, Extreme Programming (XP) [11] is chosen as the methodology to apply to this project.

4.1. Extreme Programming (XP)

The choice of XP is due to the fact that it is an agile methodology that is based on code refactoring and feedback, in addition to promoting the use of good practices and that involves the use of various elements that were considered of great interest for the development of this project.

Given that this project is developed for GEMED, an IT company, we believe that the fact that XP promotes refactoring and feedback throughout development can be very beneficial for the project growth, maintenance and update.

The promotion of the use of good practices is one of the main reasons why this methodology is chosen since, being the first project developed in a professional manner, we believe that it is a good way to adopt good customs.

Among the elements used by this methodology, the ones that were found most useful for this development are user stories, acceptance tests, estimates, iteration planning and record keeping of communications between the client and the developers.

On the other hand, XP is defined as particularly suitable for projects with imprecise and highly changing requirements, and where there is a high technical risk [12], as in this case it may be the lack of experience in software development.

4.2. Methodology adaption

As most of the agile XP methodologies are designed for developers' teams, small in this case, so the first change that should be considered in the use of this methodology is the fact that this development team is made up of one person. This entails the elimination of daily meetings or stand ups.

The figure of the client is very important in XP, in this case this role will be assumed by GEMED, whose assessment and feedback, will guide the development of this project.

Another relevant change is that the iterations will not be defined for a number of days, as we have already mentioned, the time that can be used for this project is variable, so at the beginning of each iteration an estimate will be made of the time that can be spent in depending on the external workload you have and based on it, the duration of the next iteration will be defined.

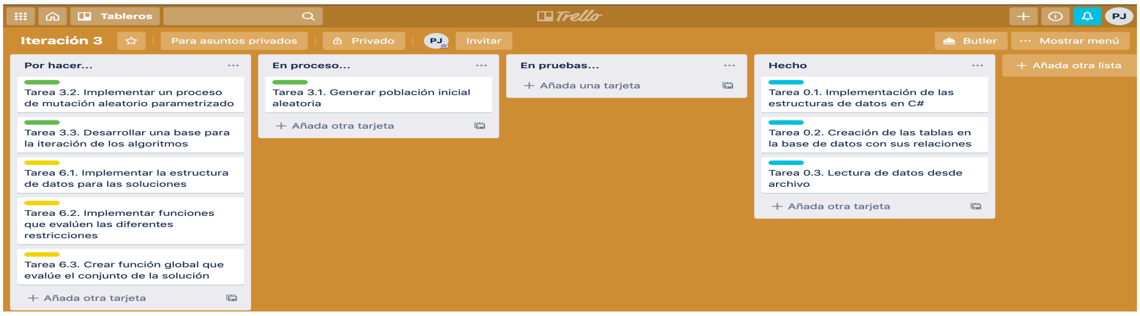
4.3. Workflow tracking

Apart from the elements used by XP for software development, other tools will be used to facilitate project monitoring.

4.3.1. Trello

This tool consists of a board that allows the creation of several columns in which cards can be created, moved and deleted, thus allowing in a very visual way to know what the current status is in the development of a project.

In our case we will use Trello [13] in the following way: at the beginning of each iteration the user stories to be carried out will be divided into small tasks, each task will be a card and an estimate and a priority will be added to it, as the iteration develops, these cards or tasks will change columns or states until they are completed.



Por hacer...
- Tarea 3.2. Implementar un proceso de mutación aleatorio parametrizado
- Tarea 3.3. Desarrollar una base para la iteración de los algoritmos
- Tarea 6.1. Implementar la estructura de datos para las soluciones
- Tarea 6.2. Implementar funciones que evalúen las diferentes restricciones
- Tarea 6.3. Crear función global que evalúe el conjunto de la solución

+ Añada otra tarjeta

En proceso...
- Tarea 3.1. Generar población inicial aleatoria

+ Añada otra tarjeta

En pruebas...

+ Añada una tarjeta

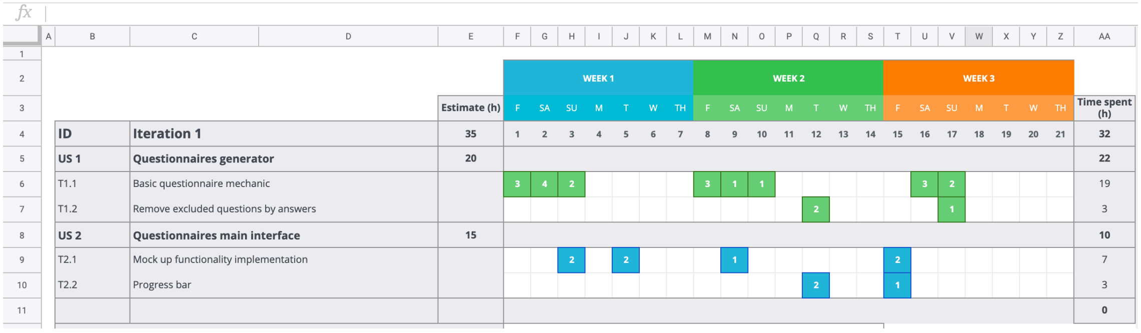
Hecho
- Tarea 0.1. Implementación de las estructuras de datos en C#
- Tarea 0.2. Creación de las tablas en la base de datos con sus relaciones
- Tarea 0.3. Lectura de datos desde archivo

+ Añada otra tarjeta

Illustration 8 - Tracking with Trello

4.3.2. Working hours tracking

To track the involvement in working hours of this project, template [14] was adapted in Excel from a Gantt chart. It documents the time spent on each of the iteration tasks, in order to record and compare the estimated time with the real-time load and thus improve future estimates throughout the project.



			WEEK 1							WEEK 2							WEEK 3							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 1**	**35**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	**32**
US 1	**Questionnaires generator**	**20**																						**22**
T1.1	Basic questionnaire mechanic		3	4	2					3	1	1						3	2					19
T1.2	Remove excluded questions by answers													2					1					3
US 2	**Questionnaires main interface**	**15**																						**10**
T2.1	Mock up functionality implementation				2		2				1						2							7
T2.2	Progress bar													2			1							3
																								0

Illustration 9 - Working hours tracking

Each iteration will be displayed on a different sheet of Excel, all iterations are timeless which means that they have not necessarily been developed one immediately after another. Therefore, each sheet will contain a list of generic weeks that will refer to the time that has elapsed since the iteration began, so we will always see "Week 1" at the beginning of each chronology, and this will refer to the first working week of said iteration.

5. Development

At the beginning of the project it was decided to divide it in two different subprojects, the risk analysis application and the penetration testing tool. This decision arose from the fact that GEMED wanted to create a risk analysis questionnaire-based application for evaluating their customers enterprises security level and we suggest to them to complement that risk analysis with a penetration testing tool, to give a more global image of safety. Risk analysis is based on the responses of a human, penetration testing is based on the "response" of the system itself.

It is kept separate because the company still does not know if it wants to incorporate it into the risk analysis page or have it separately. For this reason, the implementation was separated in time and started by the questionnaire application.

5.1. Iteration 0 – Risk analysis

5.1.1. DREAD Analysis

At the beginning of the project, GEMED indicated that they understand safety from four fundamental pillars, that might be balanced in order to maintain a good overall security. If one of these pillars decompensates it could lead to serious security problems because as it is commonly said in the field of cybersecurity "A system is as strong as its weakest link".

The design of the implementation of the DREAD methodology to a questionnaire system was challenging. At first, we had the four pillars on the one hand, on which the safety of the company is based, and on the other hand we had the DREAD methodology, with which we intend to evaluate these pillars. We were tasked with merging these two concepts into a questionnaire-based application, so that each question would relate to a pillar and receive a DREAD-based assessment depending on the chosen response.

In the next diagram is shown the design for the application of DREAD methodology to questionnaires for one concrete pillar being, in turn, equivalent for the rest of them.

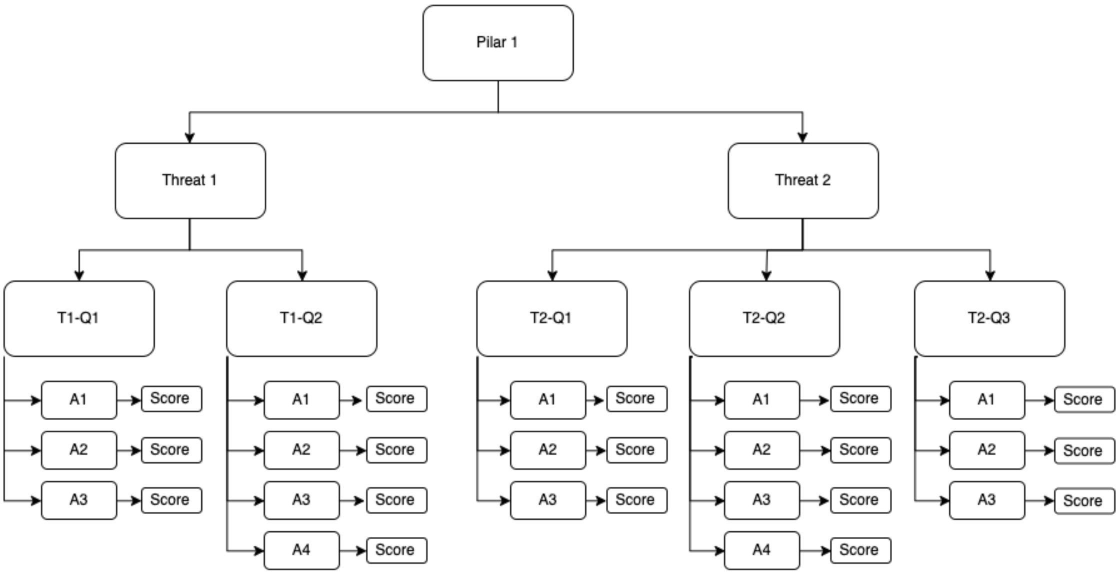


Illustration 10 - Design of DREAD application to questionnaires

First, each pillar will be associated with a set of potential threats that we would like to assess. To do this, each threat will be linked to a group of questions whose purpose will be to establish a score, based on the DREAD methodology, for that threat. Finally, each pillar will get a note that will result from the calculation of the average score of the threats with which it is related.

5.1.2. Questionnaires' interface

The following illustration shows the appearance of the mock-up elaborated for the design of questionnaires' main page. This was prepared before in order to be taken to the first meeting with the client.

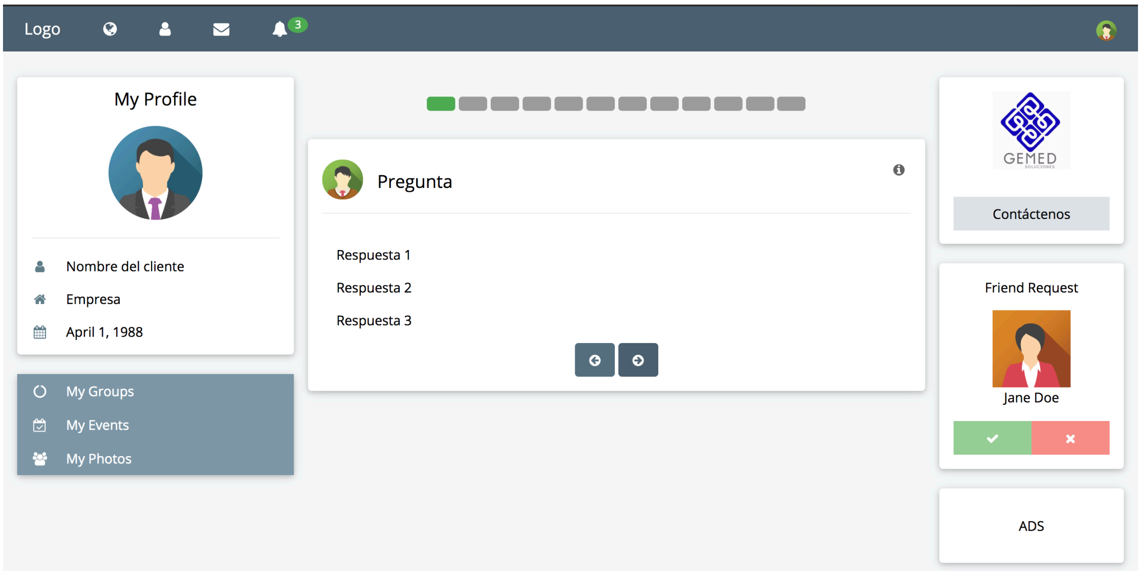


Illustration 11 - Questionnaire page mockup

In addition, the following database structure was designed from the structure shown in the Illustration 10:

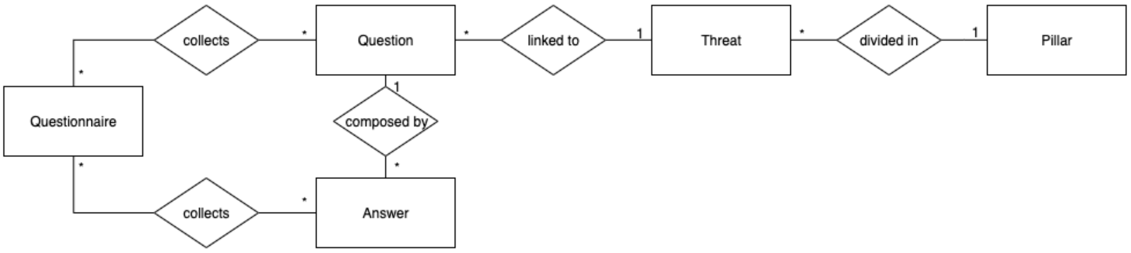


Illustration 12 - Database entity-relationship diagram

At Illustration 12 we can appreciate the different relations that these entities would maintain between them. Pillars will be divided in threats, each of them will be linked to several questions and at the questionnaire entity we will keep an instance of each question with its respective selected answer.

5.2. Iteration 1 – Risk analysis

In this iteration were implemented the user stories US01 and US02 that can be checked in Table 8 and Table 9, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 01	Title: Questionnaires’ generator	
Description: Development of a question-and-answer management system that, separating by pillar, generates surveys and is able to establish an exclusion criterion based on the responses obtained by the interviewee. This includes the creation of the database to contain all information relating to the questionnaires.		
Priority: High	Risk: Medium	Effort: 20 h

Table 8 - US01 Questionnaires' generator

ID: 02	Title: Questionnaires' main interface	
Description: Design and implementation of an interface formed by different blocks that will mainly be: • Assistant (only design should be implemented not functionality) • Questionnaire • Contact / Corporate Information In addition, the central part (questionnaires) will have a progress bar.		
Priority: High	Risk: Low	Effort: 15 h

Table 9 - US02 Questionnaires' main interface

5.2.1. Tasks

US01 – Questionnaire's generator

1.1. Database creation and connection

1.2. Basic questionnaire mechanic

1.3. Remove excluded questions by answers

US02 – Questionnaires' main interface

2.1. Mock-up functionality implementation

2.2. Progress bar

5.2.2. Acceptance tests

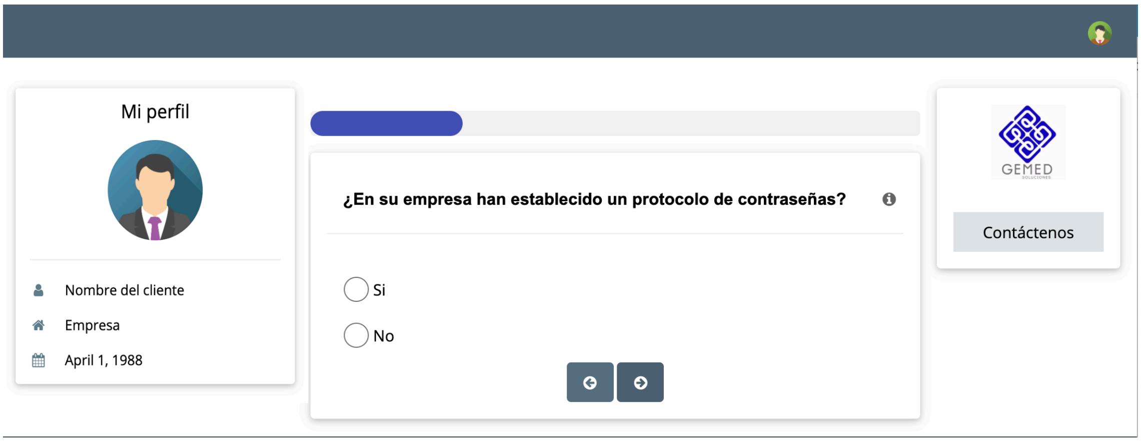
- AT1 - Questionnaire's page is divided in three sections throughout all the questionnaire.
- AT2 - When clicking on right arrow and no answer is selected an error message should be displayed.
- AT3 - Right arrow advances a question in the questionnaire.
- AT4 - Right arrow remains block on the last question.
- AT5 - Left arrow goes back a question in the questionnaire.
- AT6 - Left arrow remains block on first question.

- AT7 - When an answer that excludes a question is selected, that question is removed from the questionnaire.
- AT8 - Right arrow makes progress bar to increase.
- AT9 - Left arrow makes progress bar to decrease.
- AT10 - When last question is submitted a finalization message is displayed.
- AT11 - When contact button is clicked the contact page of GEMED should open in a new tab.

5.2.3. Development

During this iteration, the implementation of the basic mechanics of the questionnaire generator was carried out, mainly focused on the "dynamic" elimination of questions based on the answers obtained by the user. For example, if to the question 'Do you have any password protocol?' the user chooses the answer "No", no further questions related to passwords protocols will be performed.

In addition to this, the interface designed to support the course of the questionnaires was jointly developed so that the user can easily navigate through them and answer them. The following illustration shows the final aspect of this interface:


Mi perfil

Nombre del cliente
Empresa
April 1, 1988

¿En su empresa han establecido un protocolo de contraseñas?

○ Si
○ No

Illustration 13 - Questionnaire's page

We can see that certain changes were made with respect to the original mock-up. The progress bar was changed to a single piece that changes in size over time as the user advances with their answers or questions are eliminated from the total.

On the other hand, some blocks that were not considered necessary were removed from the page, ending with three main blocks or columns. The left column serves to display the user profile information. The central column collects the questions themselves. The column on the right allows the user to contact GEMED by redirecting it to its contact page.

During this iteration, the database was also created and connected to questionnaires' main page. In the following illustration we can appreciate its final structure:

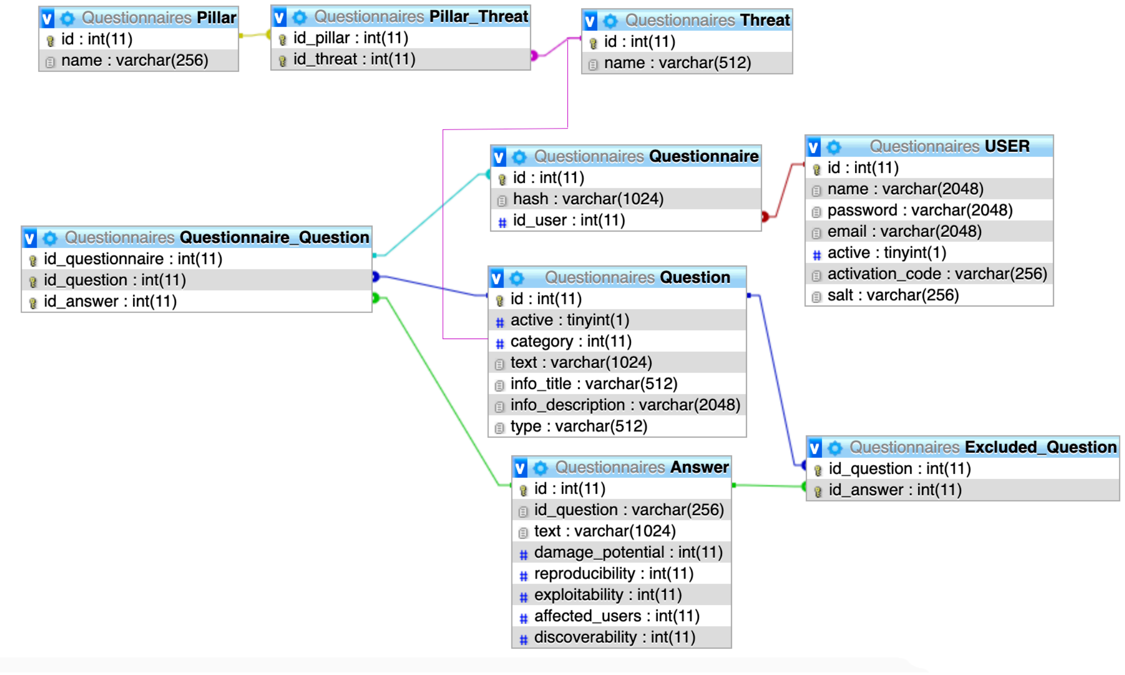


Illustration 14 - Database structure

As we can see in Illustration 14 the final structure of the database matches with the one designed previously on Illustration 12.

5.2.4. *Iteration validation*

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was that sometimes the following tests failed:

- AT6 - Left arrow remains block on first question.
- AT10 - When last question is submitted a finalization message is displayed.

The rest of the tests were successfully passed. The failing acceptance tests were reunited in a new user story US2.1 – Questionnaires' interface issues.

5.2.5. Duration

The work assigned for this iteration was estimated at 35 working hours, there were no serious problems during the development of these user stories and could be completed in less time than estimated, 32 hours.

5.3. Iteration 2 – Risk Analysis

In this iteration were implemented the user story US09 that can be checked in Table 10, and for its implementation it has been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 09	Title: Questionnaires' administrator	
Description: Implementation of an editor that allows page administrators to view, add, edit, and remove pillars, question and answer threats from the system in order to enable its updating.		
Priority: Medium	Risk: Low	Effort: 25 h

Table 10 – US09 Questionnaires' administrator

5.3.1. Tasks

US09 – Questionnaire's administrator

9.1. Creation form for each data structure

9.2. Edition form for each data structure

9.3. Addition connection with database

9.4. Edition connection with database

9.5. Deletion connection with database

5.3.2. Acceptance tests

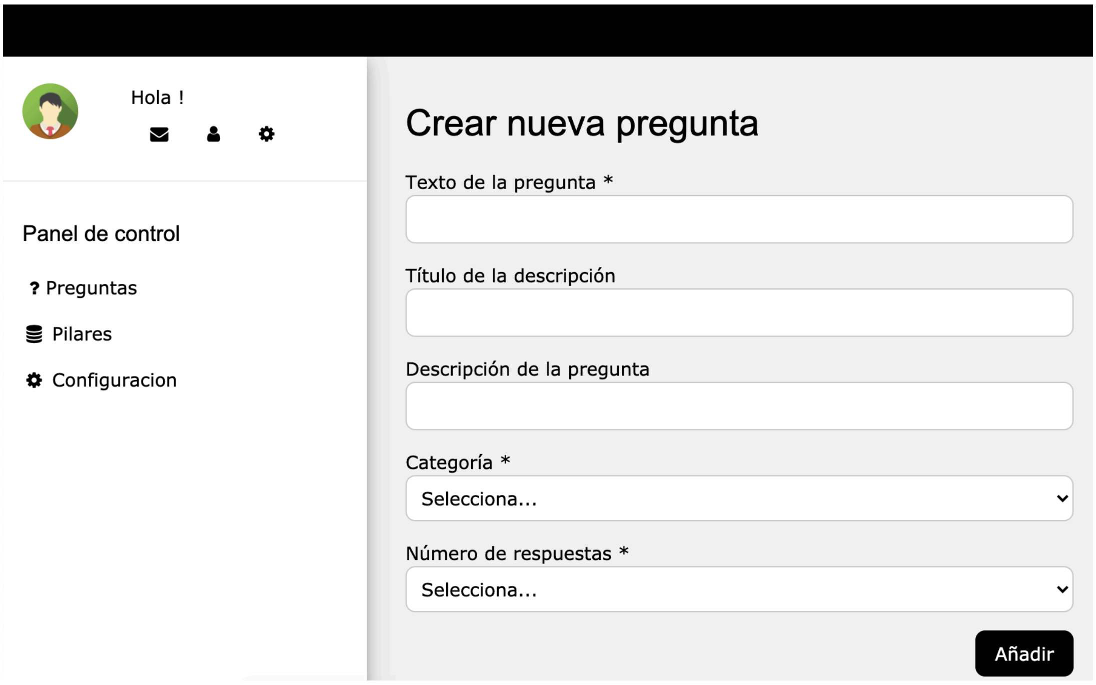
- AT12 - When submitting create question form, a new tuple is added at "Question" database table.
- AT13 - When submitting create question form, one tuple for each determined answer is added to "Answer" database table.

- AT14 - When submitting create question form, one tuple for each determined answer is added to "Question_Answer" database table.
- AT15 - When submitting create question form, a new tuple is added at "Threat_Question" database table.
- AT16 - When submitting create pillar form, a new tuple is added at "Pillar" database table.
- AT17 - When submitting create threat form, a new tuple is added at "Threat" database table.
- AT18 - When submitting create threat form, a new tuple is added at "Pillar_Threat" database table.
- AT20 - When submitting edit question form, the concrete tuple is updated at "Question" database table.
- AT21 - When submitting create question form, the concrete tuple for each determined answer is updated to "Answer" database table.
- AT22 - When submitting edit question form, the concrete tuple is updated at "Threat_Question" database table.
- AT23 - When submitting edit pillar form, the concrete tuple is updated at "Pillar" database table.
- AT24 - When submitting edit threat form, the concrete tuple is updated at "Threat" database table.

5.3.3. Development

Administrator's page was implemented separately from the questionnaire's page in order to give it the privacy it deserves. In this page the person in charge of updating the questionnaires would be able to create, edit and remove pillars, threats and questions.

In the following illustration is shown the final appearance of this admin's page:



Crear nueva pregunta

Texto de la pregunta *

Título de la descripción

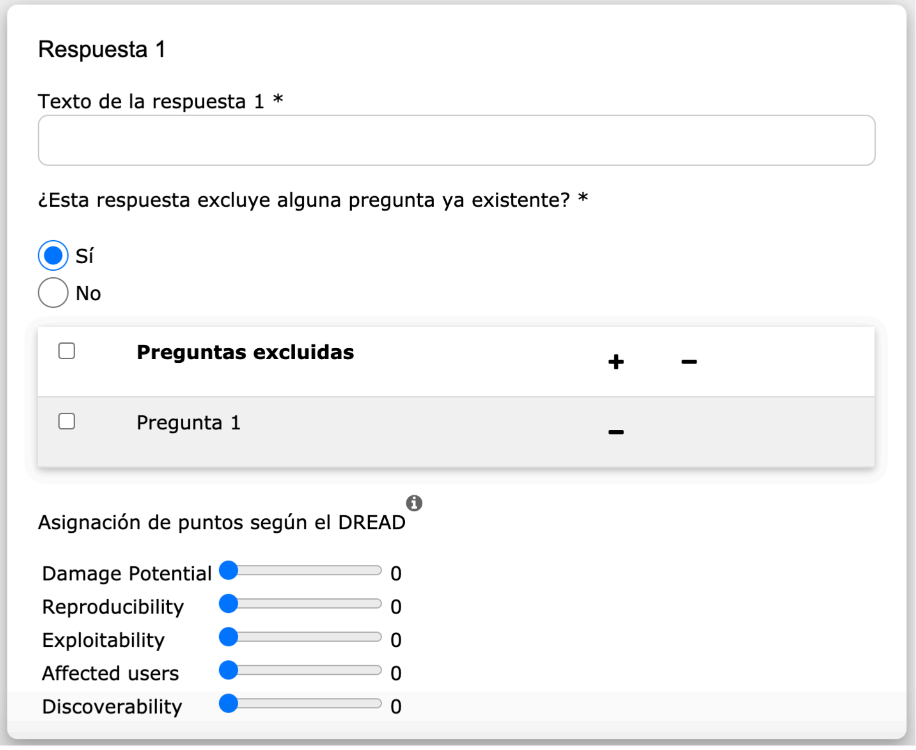
Descripción de la pregunta

Categoría *
Selecciona…

Número de respuestas *
Selecciona…

Illustration 15 - Questionnaire's administrator page

As we can see in the previous image, there is a sidebar from which we can choose between the three main components of our application: Questions, Pillars and Threats. By clicking in each of them a list with all the existent ones will be displayed. In that table we will be able to choose between adding, editing or removing. Illustration 15 shows the add question form where we are asked to introduce all the relevant data of the new question to be created. Also, we are asked to assign a threat to assess to this new question. In addition, when establishing the number of answers dynamically a form is generated for each of them, as shown in Illustration 16.



Respuesta 1

Texto de la respuesta 1 *

¿Esta respuesta excluye alguna pregunta ya existente? *

● Sí
○ No

□	**Preguntas excluidas**	+	−
□	Pregunta 1	−	

Asignación de puntos según el DREAD

Damage Potential	0
Reproducibility	0
Exploitability	0
Affected users	0
Discoverability	0

Illustration 16 - Answers' attributes form

Answers' form included a section for adding excluded questions, eventually some answers would remove unnecessary questions from the questionnaire list. When clicking on the [+] button of the "excluded questions" table a modal is opened where user can search for an existent question and add it to the table. Also, there is a section for applying a DREAD score for that concrete answer in relation to the threat linked to this new question.

5.3.4. *Iteration validation*

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was that all of the tests were successfully passed.

In addition, a meeting was held with the client to show the results of the user stories developed so far: US01, US02 and US09. Several suggestions with regard to development were addressed during the meeting:

- C1: Users can register or not, so the user profile column would not be the best.
- C2: For those users who are not registered, allow by means of a hash the resumption of the questionnaire, if they do not have the necessary time to complete it at one time.

- C3: Client propose to fill in the profile gap with an assistant that provides some relevant information about the current question to help the user in their understanding. In addition, they suggest providing with some intelligence to the assistant, such as, for example, interacting with the user if he has been inactive for a long time with help messages or reporting the time remaining to finish the questionnaire.

The following actions were taken regarding these suggestions:

- US2.1: as a result of the C1 we added some tasks to preexistent US2.1 that gathers these suggestions on the interface and the modifications needed to fix the problems that lead to the failure of certain acceptance tests.

ID: 2.1	Title: Questionnaires' interface issues	
Description: • AT6 - Left arrow remains block on first question • AT10 - When last question is submitted a finalization message is displayed • Remove user profile column		
Priority: Low	Risk: Low	Effort: 5 h

Table 11 - US2.1 Questionnaires' interface issues

- US1.1: as a result of the C2 proposal a new user story was created for the implementation of a follow-up survey system from a hash code.

ID: 1.1	Title: Resumption of questionnaire by hash code	
Description: Implementation of the resumption of questionnaires from a hash code. This includes the restoration of that questionnaire for the first question unanswered by the user.		
Priority: Medium	Risk: Medium	Effort: 10 h

Table 12 - US1.1 Resumption of questionnaire by hash code

- US03: as a result of the C3 proposal a user story is created intended for the creation of an assistant with some intelligence to support and assist the user throughout the questionnaires.

ID: 03	Title: Assistant	
Description: Character, who contributes certain information and "animation" to the course of the questionnaires, among their functions are: • Provide information about the questions to be answered by the respondent. • Indicate the estimated survey duration. • Animations such as falling asleep if the user does not answer any questions in a couple of minutes		
Priority: Low	Risk: Medium	Effort: 20 h

Table 13 - US03 Assistant

5.3.5. Duration

The work assigned for this iteration was estimated at 25 working hours, there were no serious problems during the development of these user stories and could be completed in less time than estimated, 22 hours.

5.4. Iteration 3 – Risk analysis

In this iteration were implemented the user stories US04 and US05 that can be checked in Table 14 and Table 15, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 04	Title: DREAD Analysis	
Description: Each of the answers given by the respondent will be evaluated and scored according to the DREAD method. In turn, the score of the answers will be grouped according to the threat and pillar of the question to which it is associated to provide a note to that threat and its corresponding pillar.		
Priority: High	Risk: Low	Effort: 10 h

Table 14 – US04 DREAD Analysis

ID: 05	Title: Results visualization	
Description: The results will be shown to the user with some library that allows the generation of "dynamic" charts that can help to better understand the user the level of security in which your company is located.		
Priority: Medium	Risk: High	Effort: 15 h

Table 15 – US05 Results visualization

5.4.1. Tasks

[US04 – DREAD Analysis]

4.1. Method for calculating desired scores

[US05 – Results visualization]

5.1. Chart library implementation and adaptation

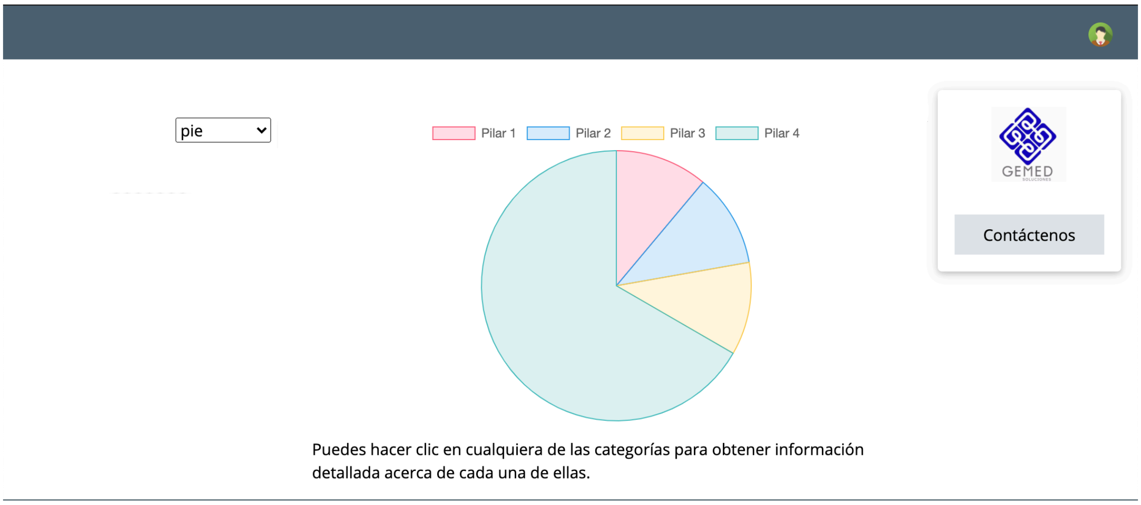
5.2. Visualization of the specific results and options

5.4.2. Acceptance tests

- AT25 - When "View results" button is clicked DREAD score for each threat and pillar should be calculated.
- AT26 - When "View results" button is clicked main charts page should be displayed.
- AT27 - When clicked on one concrete pillar of the chart a new chart with the threats' scores should be displayed.
- AT28 - When selecting another chart type from the dropdown the concrete chart type should be displayed.

5.4.3. Development

When all the questions have been answered, a "View results" button appears, by clicking it user will be redirected to the following chart page:


pie

Puedes hacer clic en cualquiera de las categorías para obtener información detallada acerca de cada una de ellas.

Illustration 17 - General results visualization

This is the page intended to visualize the overall score of the questionnaire, divided by pillar. The main goal of this pie graph is to show to the user how balanced the different pillars are and which one they need to take care of. However, there is the possibility to change the type of chart by the dropdown located at the top left of the page.

In addition, it is possible to see a more detailed score of each of the threats that form each of these pillars by clicking on them. Once any of the sections of this graph is clicked the following chart is displayed:

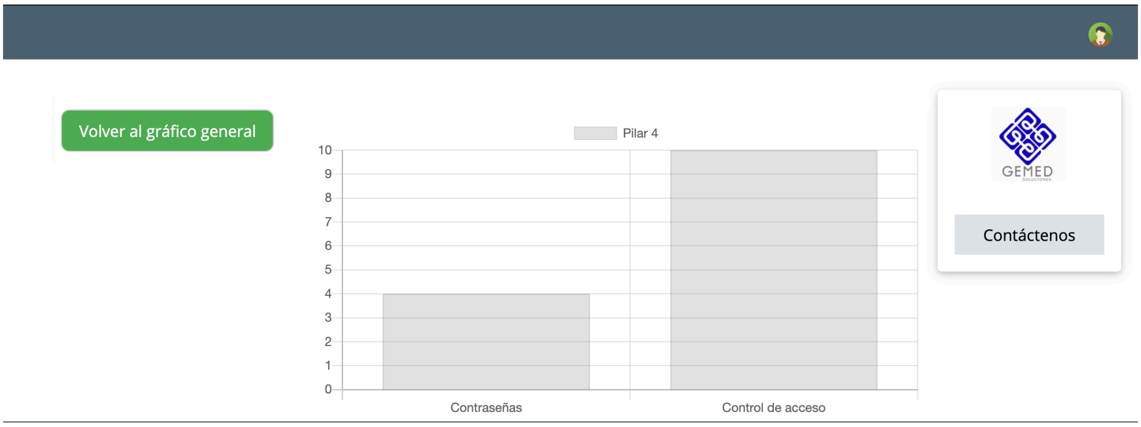


Illustration 18 - Concrete results visualization

In this detailed view the user can get more information about which specific aspects are the ones that generate the most risk within this pillar, in order to be able to better identify which is the problem and put a solution. Also, it is possible to navigate back to the previous graph by using the top left back button.

5.4.4. Iteration validation

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was that all of the tests were successfully passed.

Also, decided to add some functionality to charts by including some comparations with the average score reached by other users, this was included into US5.1 - Results visualization comparison.

5.4.5. Duration

The work assigned for this iteration was estimated at 25 working hours, some problems were encountered when modifying the graphics library, but it was possible to complete the work of this iteration in a time not much higher than estimated, 28 hours.

5.5. Iteration 4 – Risk analysis

In this iteration were implemented the user stories US06, US6.1 and US5.1 that can be checked in Table 16, Table 17 and Table 18, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 06	Title: Access control and information pages	
Description: Users will be able to register and log in with their account. In addition, this registration/login system must have the relevant security measures: • Password hashing • Salt application • Confirmation mail		
Priority: Medium	Risk: Low	Effort: 20 h

Table 16 – US06 Access control and information pages

ID: 6.1	Title: Access control - Login as guest	
Description: Acceder como invitado a los cuestionarios.		
Priority: Medium	Risk: Low	Effort: 3 h

Table 17 – US6.1 Access control – Login as guest

ID: 5.1	Title: Results visualization comparison	
Description: Create a chart that allows comparing the results of the user's questionnaire with the average score of responses of the rest of the data stored.		
Priority: Low	Risk: Low	Effort: 10 h

Table 18 – US5.1 Results visualization comparison

5.5.1. Tasks

US06 – Access control and information pages

6.1. Log in and register

6.2. Landing page

US6.1 – Access control – Login as guest

6.1.1. Guest access

US5.1 - Results visualization comparison

5.1.1. General chart comparison
5.1.2. Detailed chart comparison

Refactor and minor defects

5.5.2. Acceptance tests

- AT29 - When filling the register form a confirmation mail is sent to the specified email address.
- AT30 - When clicking on the link of the received email the user account should be activated at the database "User" table.

- AT31 - When filling the register form a new user is added to the database "User" table.
- AT32 - When missing one or more fields of the register form an error message should be displayed.
- AT33 - When clicking on the "I have an account" link the login form should be displayed.
- AT34 - When introducing a valid username and password should be redirect to application main page.
- AT35 - When introducing an invalid username and/or password an error message should be displayed.
- AT36 - When clicking on "Forgot password" link a new form should be display for introducing an email address.
- AT37 - When clicking on "No account yet" link the register form should be displayed.
- AT38 - Introducing a registered email address at the "Forgot password" form a new email should be sent to that address for updating the password.
- AT39 - At the "Change password" form when the same password is introduced twice that new user password should be updated at the database "User" table.
- AT40 - At the "Change password" form when both passwords don't match an error message should be displayed.

5.5.3. Development

During this iteration, a full registration and login system was implemented. This includes the corresponding account activation email and the form to change the password, with the corresponding sending of a link to the email. In addition, at the request of GEMED, a button was added to log in as a guest.

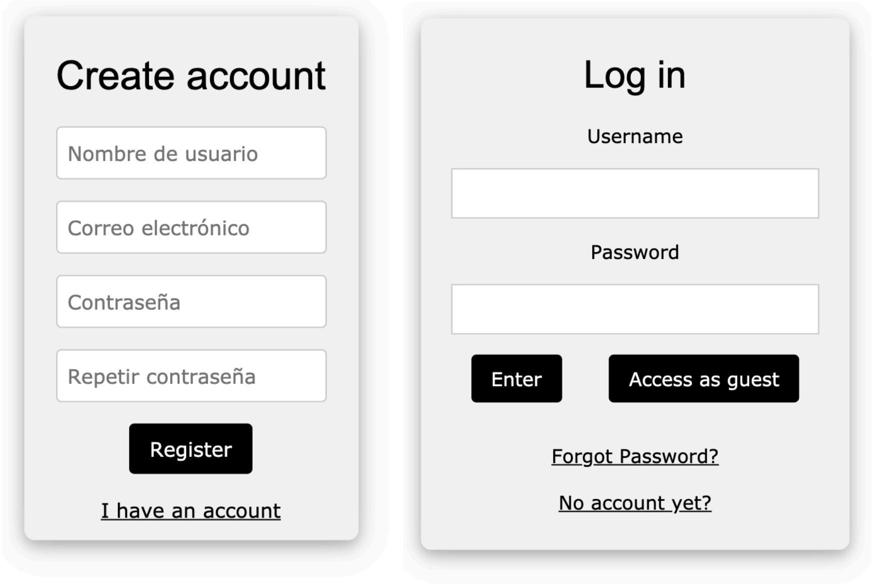


Illustration 19 – Register and log in forms

Also, some security measures were taken into account during this login and register system implementation. Passwords should never be saved in plain text, to avoid that if someone gains access to our database, they can know which passwords our users use (since they are often repeated across different applications). For this, we will use a Hash function, these are one-way functions that converts a string into a code, which means that our password will always result in the same hash code but there is no way to do it inversely. An example of this can be seen in the top part of Illustration 20.

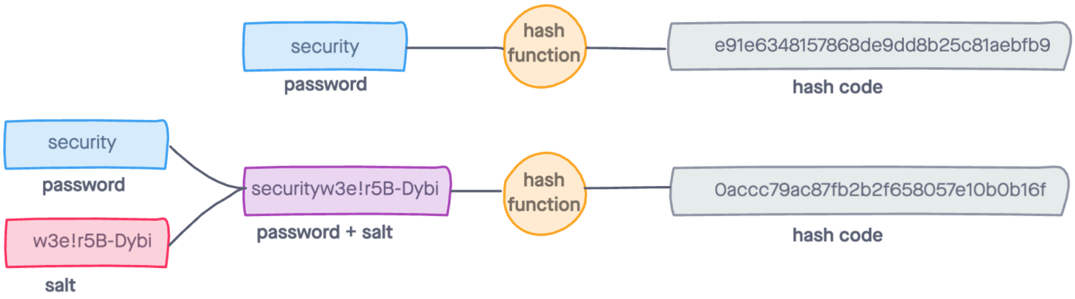
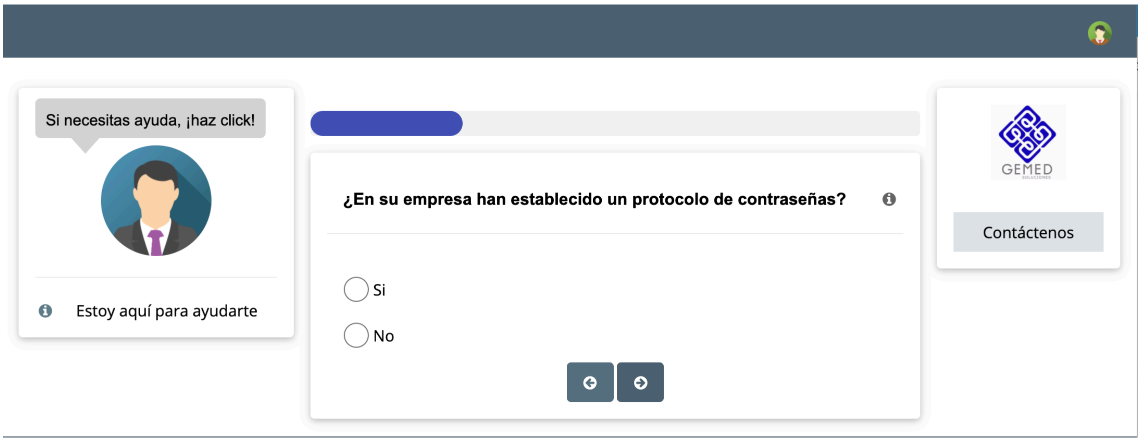


Illustration 20 - Hashing with and without salt

However, hash functions can be compromised by using dictionaries, which contain lists of common words and their corresponding hash codes. If our user’s password is a simple word, as security, it will be easily found in any dictionary. We can’t force the user not to use simple words as passwords, but we can spice up their password with salt. Salt comprises random bits that are aggregated to the password before hashing it for increasing. An example can be seen at the bottom of the Illustration 20, the blue string can be found in one of the dictionaries accessible to attackers, but the purple string, does not exist in any dictionary.

In addition to this, during this iteration several necessary modifications of the interface were also carried out. As we can see in the Illustration 21, the left column, where the user profile details were previously, was changed to a very simple prototype of what could be US03 assistant. At the moment it just offers relevant information for the current question when clicking over him.



Si necesitas ayuda, ¡haz click!

Estoy aquí para ayudarte

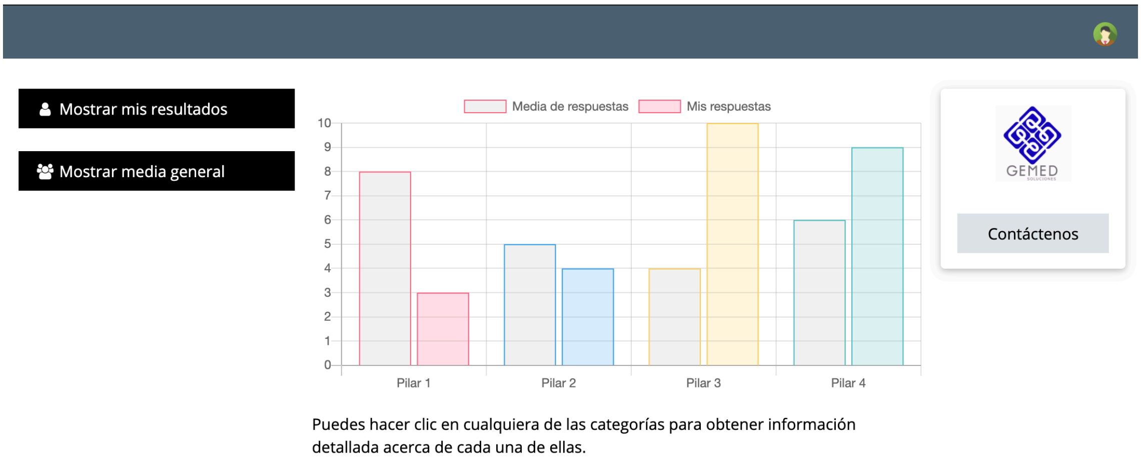
¿En su empresa han establecido un protocolo de contraseñas?

○ Si

○ No

Illustration 21 - Final questionnaires' page design

Other interface modification that was carried out through this iteration was the addition of some comparative charts to the results of the questionnaire.



Puedes hacer clic en cualquiera de las categorías para obtener información detallada acerca de cada una de ellas.

Illustration 22 - Comparative chart of general scores

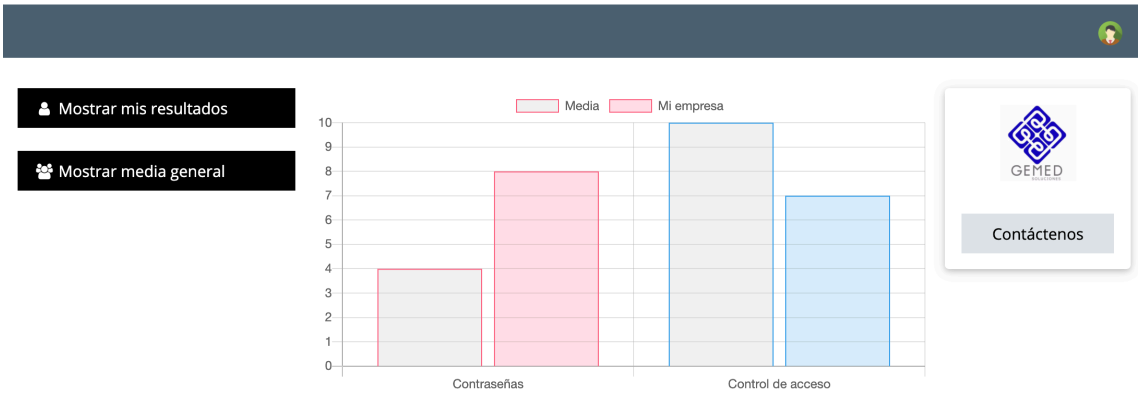


Illustration 23 - Comparative chart of detailed scores

At Illustration 22 users can compare their results with the average results of the rest of users that have completed the questionnaires. Also, as in the personal results view by clicking in any of the pillars score a detailed chart is displayed showing the score of the different threats of that pillar, in this case, comparing it with the average scores as shown at Illustration 23.

Finally, appropriate modifications were made to arrange acceptance tests that had not been successfully passed before.

5.5.4. Iteration validation

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was that all of the tests were successfully passed.

Also, a meeting with GEMED was scheduled for showing the final result of the first part of the project. The overall aspect and functionality of the Risk analysis application was warmly welcomed, but they wanted to make some changes related to charts.

- C4: DREAD score may be confusing for customers since 10 is the worst score reachable but in our daily basis we think about this score as the optimum one. So, they decided that graphs should be modified and show the score in negative values to be more clarifying and avoid confusions.

This change will be added to the backlog inside a new user story, US5.2 – Results visualization modifications.

5.5.5. Duration

The work assigned for this iteration was estimated at 28 hours of work, it took a little more time to refactorize the code, but it was possible to complete the work of this iteration in a time not much higher than estimated, 29 hours.

5.6. Iteration 0 – Penetration test

5.6.1. Penetration test web design

For the Penetration test page, the following layout was designed:

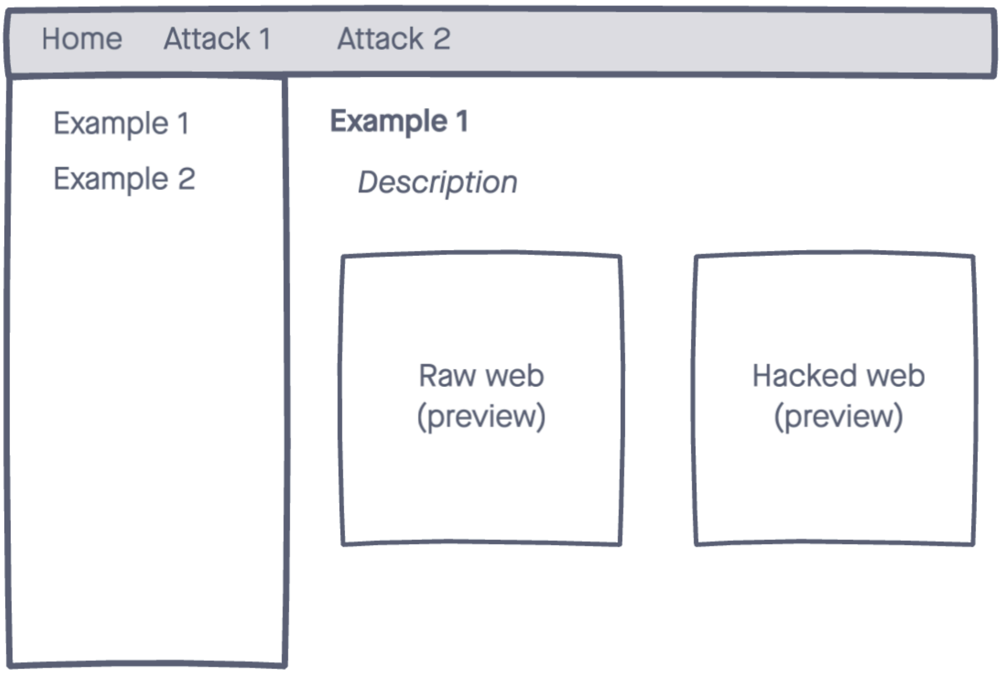


Illustration 24 - Penetration test interface design

In this layout, we can appreciate a navbar at the top with the different types of attacks and at the left a sidebar with the available examples about the selected attack. The main content of each of these examples is posed as two `<iframe>` where one shows the website without altering and the second shows the website after suffering the attack.

5.6.2. Web environment workflow

Throughout the implementation of this second part we will try to perform attacks that exploit various web vulnerabilities, this requires a clear understanding of how this environment works and what function each of the components interacting in it performs. The following illustration shows the typical web environment where a server-side language, PHP, generates the contents:

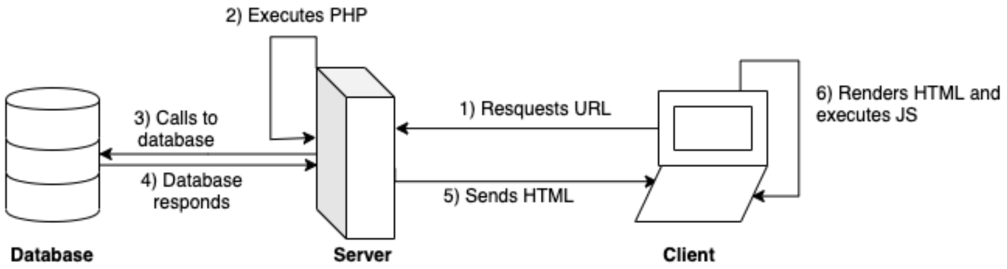


Illustration 25 - Web environment workflow

The user requests through a URL a resource that is located on the server, which searches it and executes it, thus generating the relevant calls to the database. This execution ends with the generation of an HTML webpage that is sent back to the client, the browser renders that webpage and the relevant JavaScript functions are executed. It should be noted that the browser is not able to execute PHP code, it remains always on the server and is not visible to clients. In the same way the client's browser is in charge of executing the corresponding JavaScript functions, the server is not capable, nor will it ever execute JavaScript code.

5.7. Iteration 5 – Penetration test

In this iteration were implemented the user stories US8 and US8.1 that can be checked in Table 19 and Table 20, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 08	Title: Common vulnerabilities research	
Description: Study of the most common basic vulnerabilities based on the OWASP Top Ten and the audit format that the client follows when assessing companies. This also includes to select which vulnerabilities will be implemented in the first version of the project.		
Priority: High	Risk: High	Effort: 15 h

Table 19 – US08 Common vulnerabilities research

ID: 8.1	Title: Collect and attack potential victims	
Description: Investigate and select between the different vulnerable webpages which are more suitable for testing the vulnerabilities that are going to be implemented in projects' first version.		
Priority: Medium	Risk: High	Effort: 20 h

Table 20 – US8.1 Collect and attack potential victims

5.7.1. Tasks

US08 – Common vulnerabilities research

8.1. Research vulnerability replication

8.2. Research vulnerability exploitation

US8.1 – Collect and attack potential victims

8.1.1. Find vulnerable pages

8.1.2. Launch attacks over them

5.7.2. Development

In this iteration, a thorough investigation was carried out on the different web vulnerabilities, described in the OWASP Top Ten project. Mainly the goal of this research was to clarify the exploitability and discoverability of these common vulnerabilities and how much effort will it cost to us to exploit them remotely.

- Injections: we decided to implement this vulnerability because it has been one of the most common for a long time, and it does not need any external tools or specific applications to be exploited. We only need to inject the payload into the website URL to carry out the attack. Specifically, we will focus on SQL injections as it is the most widespread and about which we have more knowledge.

- Broken authentication: we decided not to implement, in this first version of the project, this vulnerability because of the complexity it presents. The processing and submission of credentials can be carried out in a wide variety of ways and would require a great deal of technical effort. On the other hand, the purpose of this project is that users can self-assessment free of charge their sites, so GEMED considered that this type of attack should be done in a more professional way and under their supervision.

- Sensitive data exposure: because of similar reasons as the previous vulnerability and the fact that we did not find any vulnerable page that presents it, we decided to not implement this one neither, in this first version of the project.

- XML External Entities: exploitability of this vulnerability is really complex since it relies on old XML processors and we also decided it has a high degree of complexity. That's why we decided to not implement it, at the moment.

- Broken Access Control: For being able to reproduce this vulnerability we will need to register or have some credentials for being able do try to access unauthorized functionality. So, we decided to not implement this vulnerability, in this first version of the project, for the same reasons as the broken authentication.

- Security Misconfiguration: This vulnerability will need deep research about our victim for being able to "guess" which kind of misconfiguration it could has. For the purpose of automation, we thought this will be really hard to provide this type of detailed search in an abstract way, so decided to not go for this one neither, at the moment. Anyway, we think this could be done in a second stage of the project once we have already performed some automated attacks and see how it goes.

- Cross-Site Scripting (XSS): After some research about this vulnerability we decided to implement it, since, as the SQL injections, it doesn't need any external application and it can also be injected at the URL of the website.

- Insecure deserialization: For exploiting this vulnerability we will need to perform a deep research about the website we are attacking and being sure that is processing some type of serialized object like JSON files for example. This will be really complex and time consuming, so we decided to not implement this, in this first version of the project.

- Known vulnerabilities: There are several web pages on Internet that allows you to know what technologies, and version, are behind any website. Also, there are several pages where you can check for known vulnerabilities for these technologies. We decided that it could be interesting to join this to type of pages for directly telling what known vulnerabilities are behind a website, but it is not going to be implemented, at the moment.

- Insufficient logging and monitoring: Since these are things internally managed by website owners, we decided it is out of our scope for this project and we are not going to implement it, in this first version of the project.

After analyzing which vulnerabilities will be implemented account was taken of the vulnerable pages referred to in section [3.Vulnerable websites], to choose which ones will fit better to us:

- PentesterLab: Within all the possibilities offered by the *PentesterLab* we decided to use the exercise called *Web for pentester I* as a potential victim for the attacks that we want to automate since it has specifically numerous SQL injection and XSS sample exercises.

- Google Gruyere: Although it offers the possibility to execute XSS attacks we decided not to use this website as a victim because its construction, it only works with file uploading, would be quite difficult for us to attack. Also, it has no SQL injection vulnerable examples.

- bWAPP: We decided to use this vulnerable website as a victim, because it presents all the vulnerabilities collected in the OWASP Top Ten, it is easy to use and has different pages for each example which simplifies a lot the attack process.

- WebGoat: After trying to install this victim we found several difficulties; a lot of work was needed and also it was only possible to launch with *Docker,* so we decided to stop trying it and not to use this site as a victim for this project.

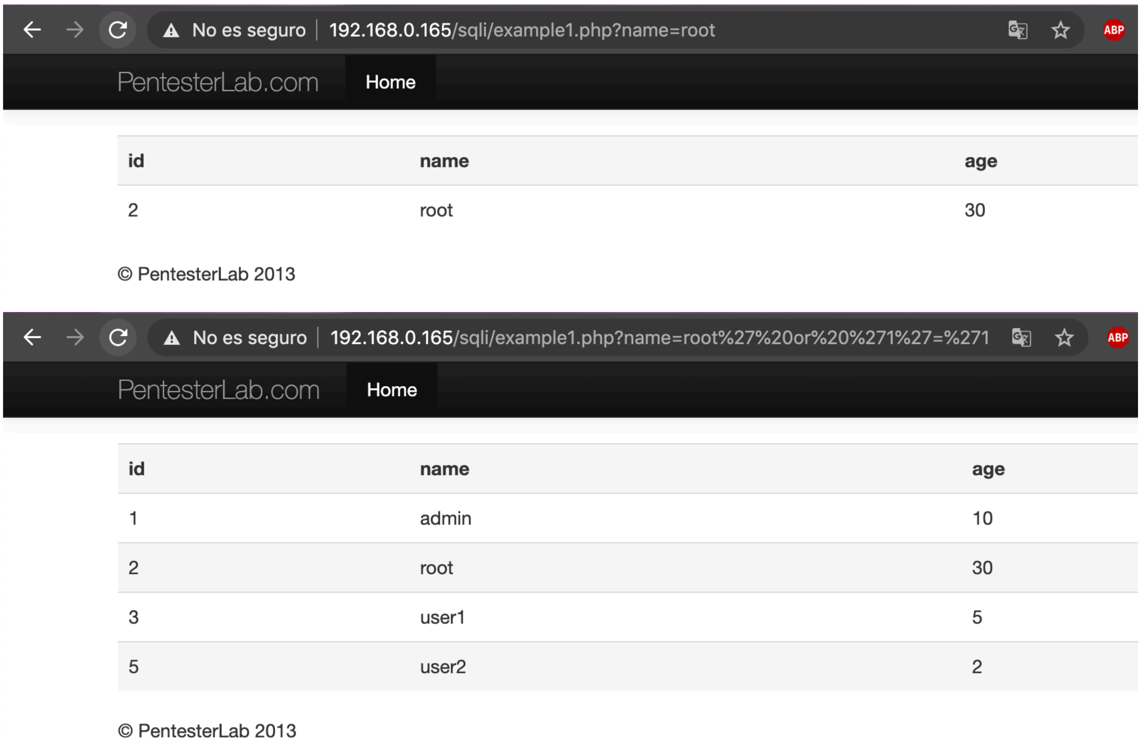
- Zixem: This website is focused specifically in SQL injections and XSS, it is simply to use and also has different pages for each example. We included it as a victim for our project.

Once we have already research about the vulnerabilities that most suit us and decided which vulnerable web pages we are going to use as victims, we can start attacking them. For this purpose, SQL injections and XSS attacks were investigated in detail, with both concluding that the best way to execute these attacks is through the insertion of payloads into the URL of the website in question.

Let's remember what we explained at Illustration 25, at web environment client requests resources through URLs and in many times these requests may be accompanied by extra information. There are two types of requests that a user can perform: GET and POST. The only difference between these two is that GET variables are hardcoded in the URL, which is the case that we will take advantage of in this project, while POST variables are sent "hidden" of the user's view.

SQL injections potentially takes place when some data is asked to the user and sent to the server through GET or POST requests, normally this input is used on a database request and if it is not processed correctly can end up in data exposure, corruption or even deletion.

In this project SQL injections we will try to show more information from the database than expected by injecting SQL queries to GET variables in the website URL. We can see an example in the following illustration:



192.168.0.165/sqli/example1.php?name=root

id	name	age
2	root	30

© PentesterLab 2013

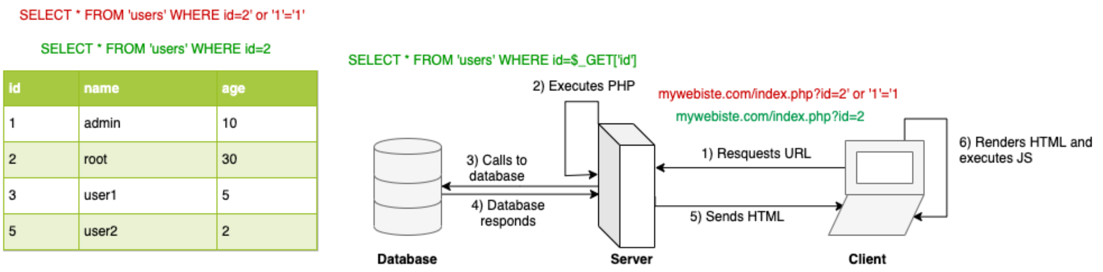
192.168.0.165/sqli/example1.php?name=root%27%20or%20%271%27=%271

id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

© PentesterLab 2013

Illustration 26 - Before and after of a SQL injection at Web for pentester I

In the previous images we can see the result of a SQL injection attack but let's take a closer look into why this is happening:


SELECT * FROM 'users' WHERE id=2' or '1'='1'
SELECT * FROM 'users' WHERE id=2
SELECT * FROM 'users' WHERE id=$_GET['id']
2) Executes PHP
mywebiste.com/index.php?id=2' or '1'='1
mywebiste.com/index.php?id=2
1) Resquests URL
6) Renders HTML and executes JS
3) Calls to database
4) Database responds
5) Sends HTML
Database
Server
Client

id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

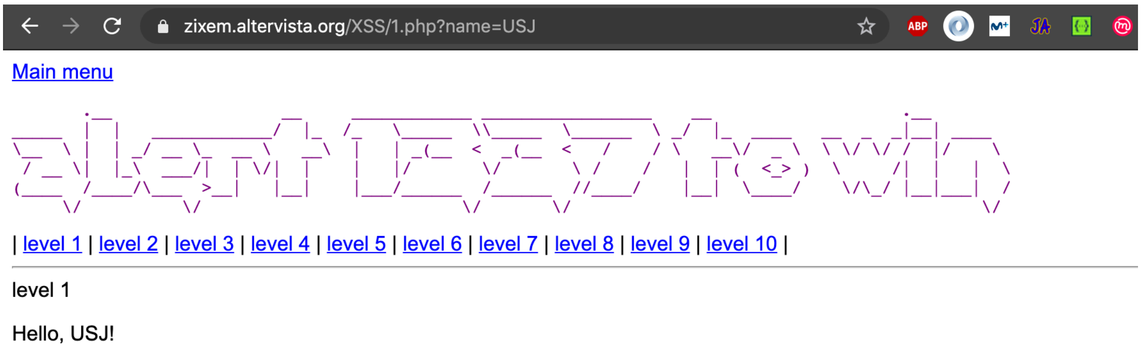
Illustration 27 - SQL injection functioning

At Illustration 27 we can see an explanation of the SQL injection that we just performed. Normally users will request the webpage as marked in green at 1), but in our case we added a payload (red). Once this "injected" URL arrives to the server, PHP is not filtering user input for variable `$_GET['id']` so it is directly sent to the database which, instead of executing the green query as it is expected, ends up executing the red one, resulting in a data exposure at client side, as we previously saw at Illustration 26.

Now let's talk about XSS attacks, which can be classified in two different categories depending on how it is exploited: [15]

- Reflected XSS: It consists in modifying values that the web application uses to pass variables between two pages.
- Stored XSS: It consists in inserting dangerous HTML code on sites that allow it; in this way it will be visible to users who enter the modified site

In this project we will implement Reflected XSS attacks that will consist in the remote execution of an `alert()` message by injecting payload to GET variables in the website URL. We can see an example in the following illustration:

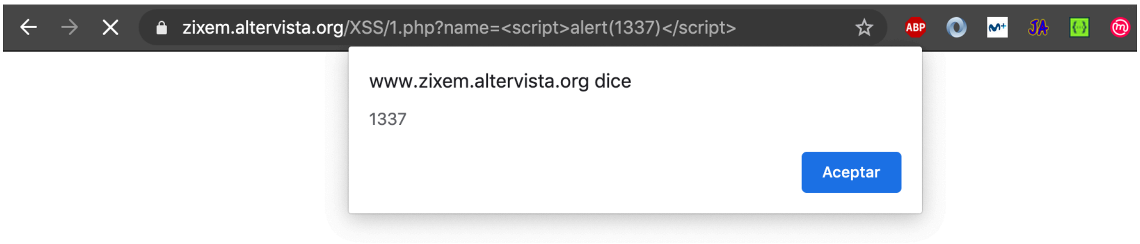

zixem.altervista.org/XSS/1.php?name=USJ

Main menu

| level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 | level 9 | level 10 |

level 1

Hello, USJ!



zixem.altervista.org/XSS/1.php?name=<script>alert(1337)</script>

www.zixem.altervista.org dice

1337

Illustration 28 – Before and after of an XSS attack at Zixem's website

Now that we have seen the result of performing an XSS attack let's take a closer look at why this script execution is taking place:

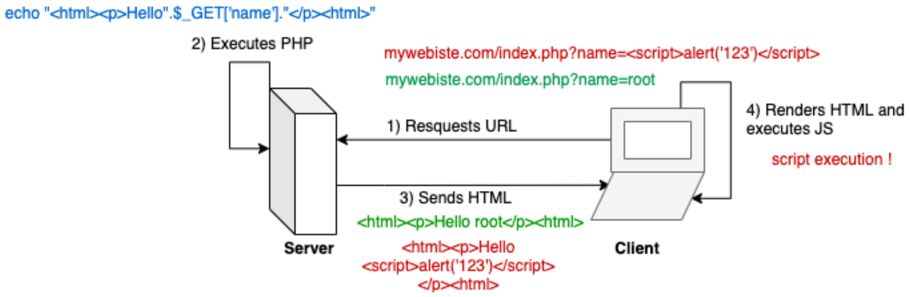

echo "<html><p>Hello".$_GET['name']."</p><html>"
2) Executes PHP
mywebiste.com/index.php?name=<script>alert('123')</script>
mywebiste.com/index.php?name=root
1) Resquests URL
4) Renders HTML and executes JS
script execution !
3) Sends HTML
<html><p>Hello root</p><html>
<html><p>Hello <script>alert('123')</script> </p><html>
Server
Client

Illustration 29 - XSS attack functioning

At Illustration 29 we can see an explanation of the XSS attack that we just performed. In this case we "inject" a script in the `$_GET['name']` variable which, once again, is not being processed by the PHP script, so it is directly concatenated into the response HTML code resulting in its execution at client's browser.

Each of the different websites, with all the available examples, was tested to see which attacks were successful in each of them and a compilation of these attacks was made.

5.7.3. Iteration validation

Following the development of this iteration, a meeting was held with the project manager where we discussed about the results of the research done and about the decisions taken through it.

5.7.4. Duration

The work assigned for this iteration was estimated at 35 working hours, due to the large number of vulnerabilities to be studied and the level of complexity, mostly greater than expected, it took longer than expected, 47 hours.

5.8. Iteration 6 – Penetration test

In this iteration were implemented the user stories US10 that can be checked in Table 21, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 10	Title: Attacks automation	
Description: Implementation of the launch of attacks in an automated manner and checking whether or not they have been successful by a script. This will require a process of abstraction to generalize attacks as widely as possible in order to serve as many victims as possible.		
Priority: Medium	Risk: Low	Effort: 25 h

Table 21 - US10 Attacks automation

5.8.1. Tasks

US10 – Attacks automation

10.1. Attacks abstraction and generalization

10.2. Launch automated attacks

10.3. Check automatically attacks result

5.8.2. Acceptance tests

- AT41 – When launching the XSS attack and resulting html contains the corresponding string a message of succeed should be displayed.
- AT42 – When launching the XSS attack and resulting html does not contain the corresponding string a message of "not vulnerable" should be displayed.
- AT43 – When launching the SQL injection attack and resulting html contains more <tr> tags than the original html a message of succeed should be displayed.
- AT44 – When launching the SQL injection attack and resulting html does not contain more <tr> tags than the original html a message of "not vulnerable" should be displayed.

5.8.3. Development

After studying the different available victims and the successful attacks in each of them, the process of abstraction and automation began.

During the implementation of the automation process several problems were encountered:

- P1: On the one hand, launching such attacks against any page without the consent of its owner and its host is illegal. Although these pages are designed to test and launch attacks we do not know if they will "tolerate" a massive attack, as it could be interpreted the automation we are planning, since the number of attacks per unit of time would not be human "possible".

- P2: On the other hand, the biggest problem we've found is checking. Automation needs a method to corroborate that the attack has succeeded, however, the result obtained after each one of these exploits will change depending on the content of the web that is being attacked and how it is programmed. For this reason, the process of abstraction to find an overall result was very complicated.

- P3: Finally, not all the victims that had been collected were valid for the automation process. Many of these websites are designed to offer a complete learning about the various web vulnerabilities and therefore have a login system, which, although is not "secure" at all, they cause that to be able to attack the URLs of the different exercises it is necessary to include a username and a password.

The following ideas were raised for trying to solve these problems:

- I1.1: Creating our own vulnerable webpages launched locally for being able to attack without legal risks.

- I1.2: Some of the webpages are valid like *PentesterLab* since they offer iso images that are launched locally.

- I1.3: Implement a cooldown between attacks to emulate human requests timing.

- I2.1: For SQL injections it was established that in general the most common result when an attack succeed full table tuples are displayed. So, we can check the number of `<tr>` tags in the raw victim's URL and compare with the number of tags at the attacked victim's URL.

- I2.2: For XSS attacks, the goal is to append a `<script>` tag the html page so we can search for it directly to see if the attack succeeds.

- I2.3: For XSS attacks, instead of executing an `alert()` message we could try to execute a `document.body.innerHTML = “”` for changing the victim’s html completely.

- I3: It could be easily added to the get request of the victim’s URL the necessary variables for logging in.

When trying out all these ideas to find out which suits better for each problem, a new issue arises:

- P4: XSS attacks cannot be properly checked because they are JavaScript code intended to execute in client-side of the application so requesting the URL with the payload injected from a PHP won’t launch the attack.

After analyzing and testing the different ideas, the following solutions were applied:

- S1: Developing our own vulnerable pages would involve a lot of research to generate these vulnerabilities which means a huge amount of time and decided that it is not worth it. PentesterLab will be one of our victims since is the only one that provides us the safety to launch the attacks against it. Zixem would be our other victim, although it is an external hosted website, we found out that one of the exercises that this site proposes asked for creating an automated brute force attack, so its server won’t be alarmed by the big number of requests. This means not cooldown needed neither.

- S2: I2.1. is the best option and the only one that, for the moment, would allow us to verify that the SQL injection attack succeed. Related to XSS attacks idea I2.3. would have been the best solution to be able to verify that the attack had worked. But we could not make use of it because when performing the URL request from a PHP script, that script function is not executed at any time, so it has no effect. The same happens for I2.2. but we decided to go for it, although there is the possibility of finding matches with this tag even if the attack has not been successful. For example, if the website treats the exploit as a plain text string.

- S3: we decided not to implement any solution to this problem. Taking into account the goal of this project it makes no sense to ask customers to introduce their credentials in order to test their sites. They won’t do it and we can’t ask them so. We can check if the automation works with different pages with ones mentioned before.

5.8.4. Iteration validation

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was that all tests succeed.

In this meeting we also took care about the problems that appeared during the implementation and discussed about the solutions that have been finally taken to solve them.

5.8.5. Duration

The work assigned for this iteration was estimated at 25 working hours, but the numerous problems that appeared during its implementation it finally took 30 hours to complete all the tasks.

5.9. Iteration 7 – Penetration test

In this iteration were implemented the user stories US07 that can be checked in Table 22, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 07	Title: Penetration test interface	
Description: Design and development of the appearance of a tool that will provide support and feedback to users on the study of different web vulnerabilities and how to prevent them. It will also have a separate form that will collect the data needed to launch the automated attacks.		
Priority: Medium	Risk: Medium	Effort: 20 h

Table 22 - US07 Penetration test interface

5.9.1. Tasks

[US07 – Penetration test interface]

7.1. Vulnerabilities detailed pages

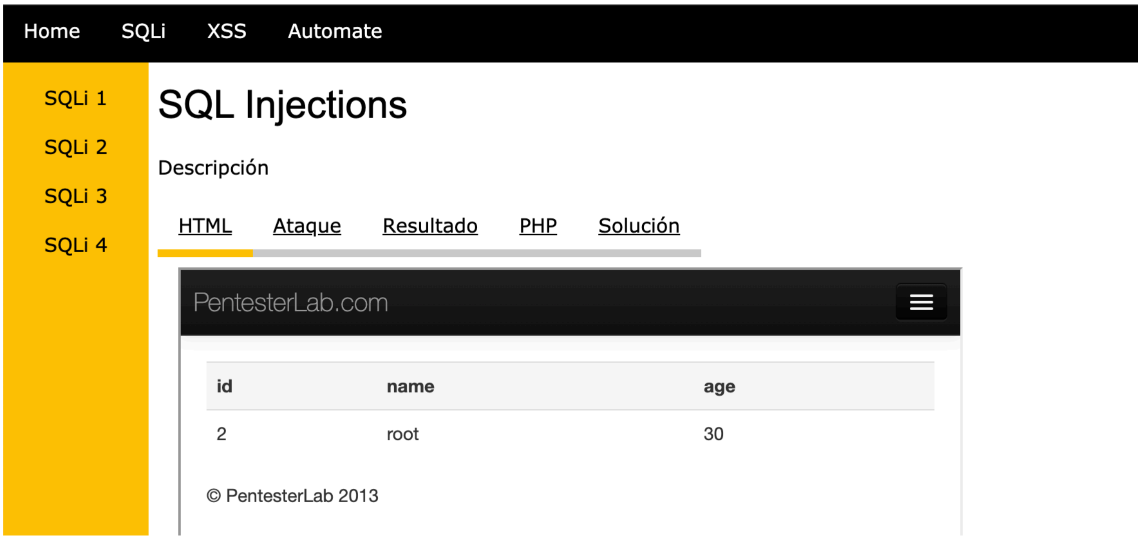
7.2. Automated attacks form page

5.9.2. Acceptance tests

- AT45 - When automation script is executed all of the attack are launched.
- AT46 - When the resulting html of an XSS attack has a match with the alert injected it should be marked as successful.
- AT47 - When the resulting html of an XSS attack has no match with the alert injected it should be marked as failed.
- AT48 - When the resulting html of an SQL injection attack has a greater number of <tr> tags than the original one it should be marked as successful.
- AT49 - When the resulting html of an SQL injection attack has less or the same number of <tr> tags than the original one it should be marked as failed.
- AT50 - When automation script is executed a list with the vulnerabilities found is generated.

5.9.3. Development

At the beginning of this iteration the main appearance of the vulnerabilities educational page was implemented. As shown previously in its design, it will consist in a navigation bar where the different type of attacks could be found and once one of these attacks type is selected a sidebar would appear offering several examples. Differently from the initial layout, each of these examples will consist in a brief description and several tabs that will contain different relevant information like the source code of the victim or an explanation of the attack, why it is working and how to prevent it. We can see the final result of this interface in the following illustration:



SQL Injections

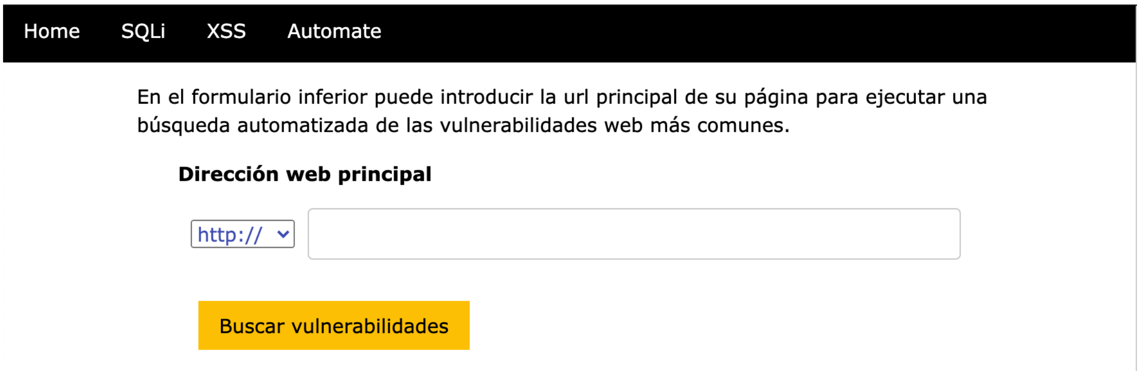
Descripción

id	name	age
2	root	30

© PentesterLab 2013

Illustration 30 - Vulnerabilities detailed page

Also, the navbar of this web page will contain an option for the automation that will be in charge of executing all our attacks against a specified URL. The resulting appearance of these automation form is shown in the following illustration:



En el formulario inferior puede introducir la url principal de su página para ejecutar una búsqueda automatizada de las vulnerabilidades web más comunes.

Dirección web principal

Illustration 31 - Automated vulnerability search

In order to make this automation a little bit more complete we implemented a library called PHP XML Sitemap Generator [16] that will be in charge of finding out all the available URLs under the main one that is specified in the previous form. This library will generate an XML file with all the routes, under this main URL, them will be analyzed by our automation PHP script that will be responsible for launching all our attacks against each of these routes. When all these has finished and message indicating the number of vulnerabilities found will be displayed.

5.9.4. Iteration validation

After the development of this iteration, a meeting was held with the project director where the acceptance tests were carried out in order to validate the work done. The result was that all tests succeed.

Furthermore, we thought it will be interesting to provide this didactic website the capacity of growing easily so we decided to create a new user story US11 – Penetration test extensibility for allowing the addition of new technical examples in a more comfortable way.

5.9.5. Duration

The work assigned for this iteration was estimated at 20 working hours, no problems were encountered during implementation and all tasks could be completed in 18 hours.

5.10. Iteration 8 – Penetration test

In this iteration were implemented the user stories US11 that can be checked in Table 23, for their implementation they have been divided into a series of simple tasks and acceptance tests have been created that the implementation must pass to be considered correct.

ID: 11	Title: Penetration test extensibility	
Description: To promote the growth of the tool you want to develop a dynamic structure that facilitates the expansion of the content of the tool in a simple and comfortable way.		
Priority: Medium	Risk: Low	Effort: 20 h

Table 23 – US11 Penetration test extensibility

5.10.1. Tasks

[US11 – Penetration test extensibility]

11.1. Dynamic structure to facilitate page extensibility

5.10.2. Acceptance tests

- AT51 – When a new folder is added at root folder, a new attack is added to the navbar.
- AT52 – When a subfolder is added, a new example is added to the sidebar.
- AT53 - When a new folder named starting by "_" is added at root folder, a new attack is not added to the navbar.
- AT54 – When a subfolder named starting by "_" is not is added, a new example is not added to the sidebar.
- AT55 – When creating a file called `content.json`, with proper content format, inside a subfolder, its content should be rendered.
- AT56 – When creating a file called `content.json`, with not proper content format, inside a subfolder, its content should not be rendered.

5.10.3. Development

In this iteration, the vulnerability web was restructured in order to automate its growth and development in the future. The following system was designed for this purpose:

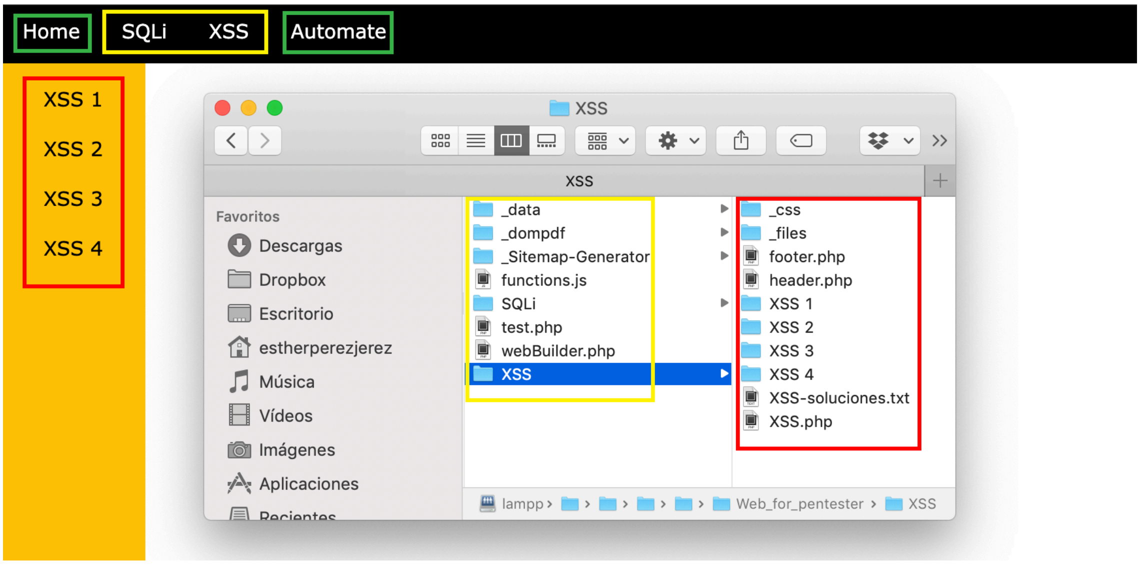
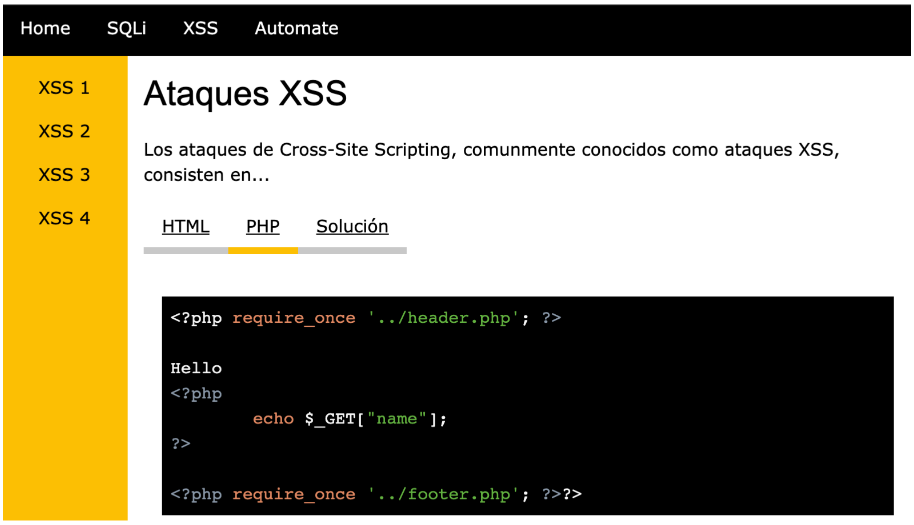


Illustration 32 - Vulnerabilities' web structure

This structure is formed by three main components. Firstly, we have some fixed buttons (green) on the navigation bar such as "Home" and "Automate", they will be always there. Secondly, we have different types of attacks (yellow), for adding one of these types to the webpage we just need to create a folder at root, notice that files and underscored folders are not going to be added to the navbar. Thirdly, we found the examples (red) for these previous attacks that corresponds to the subfolders inside the concrete attack folder, once again files and underscored folders won't be added to this sidebar.

Once any of these examples is clicked its main content would be displayed as follows:



Ataques XSS

Los ataques de Cross-Site Scripting, comunmente conocidos como ataques XSS, consisten en...

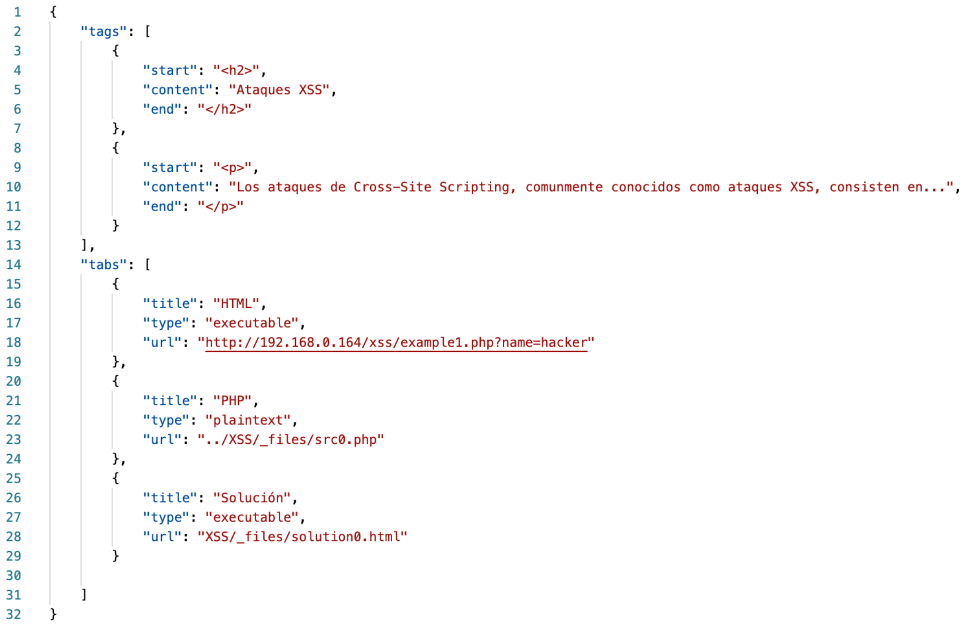
```
<?php require_once '../header.php'; ?>

Hello
<?php
        echo $_GET["name"];
?>

<?php require_once '../footer.php'; ?>?>
```

Illustration 33 - Attack example content visualization

For the content above to be displayed each of the subfolders will contain a file named as `content.json` where the content of these examples would be describe for subsequently be displayed on the page. In the following illustration an example of this content format is shown:



```
{
    "tags": [
        {
            "start": "<h2>",
            "content": "Ataques XSS",
            "end": "</h2>"
        },
        {
            "start": "<p>",
            "content": "Los ataques de Cross-Site Scripting, comunmente conocidos como ataques XSS, consisten en...",
            "end": "</p>"
        }
    ],
    "tabs": [
        {
            "title": "HTML",
            "type": "executable",
            "url": "http://192.168.0.164/xss/example1.php?name=hacker"
        },
        {
            "title": "PHP",
            "type": "plaintext",
            "url": "../XSS/_files/src0.php"
        },
        {
            "title": "Solución",
            "type": "executable",
            "url": "XSS/_files/solution0.html"
        }

    ]
}
```

Illustration 34 - content.json example

This `content.json` will be divided in two sections: tags and tabs. Tags section will result in the text before the tabs sections, it is completely customizable since it accepts any HTML tag. Tabs section is focused on the different aspects of the attacks that we will like to show, there are two types: executable, for indicating that the code should be processed; and plaintext, for indicating that the content should be displayed without executing it. For any of these types of tabs a "url" needs to be specified, it can be a file path or a website URL.

5.10.4. Iteration validation

Following the development of this iteration, a meeting was held with the project manager where acceptance tests were carried out in order to validate the work done. The result was all tests succeed.

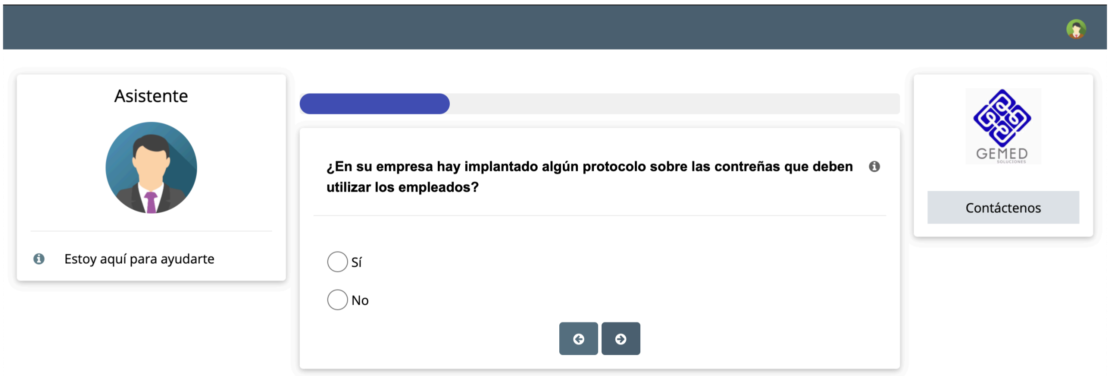
5.10.5. Duration

The work assigned for this iteration was estimated at 20 working hours, but the work of this iteration was completed in 15 hours.

6. Results

As a result of this project two software products have been produced:

Risk analysis application, programmed in PHP and JavaScript with a connection to SQL database, implemented in the first half of the project. Its purpose is to allow users, through the completion of some questionnaires, to know the general state of safety of their company and the level of danger they may face. It resulted in a valid and practical conversion of DREAD from an assignment of values to threats into a questionnaire-based tool that assigns these values while progressing through questions. Furthermore, questionnaires are dynamic, extensible and easily adaptable to different scenarios through the administration interface.


Asistente

Estoy aquí para ayudarte

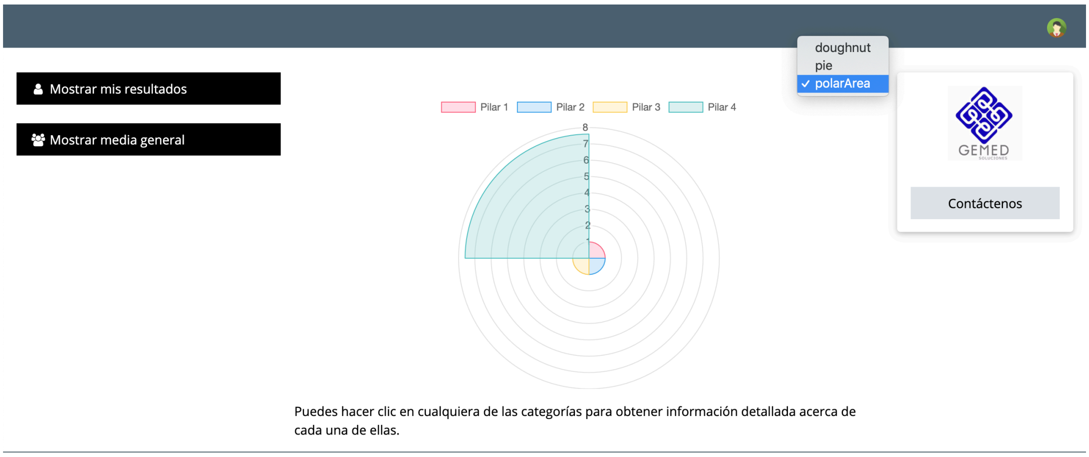
¿En su empresa hay implantado algún protocolo sobre las contreñas que deben utilizar los empleados?

○ Sí

○ No

Illustration 35 - Questionnaires' page

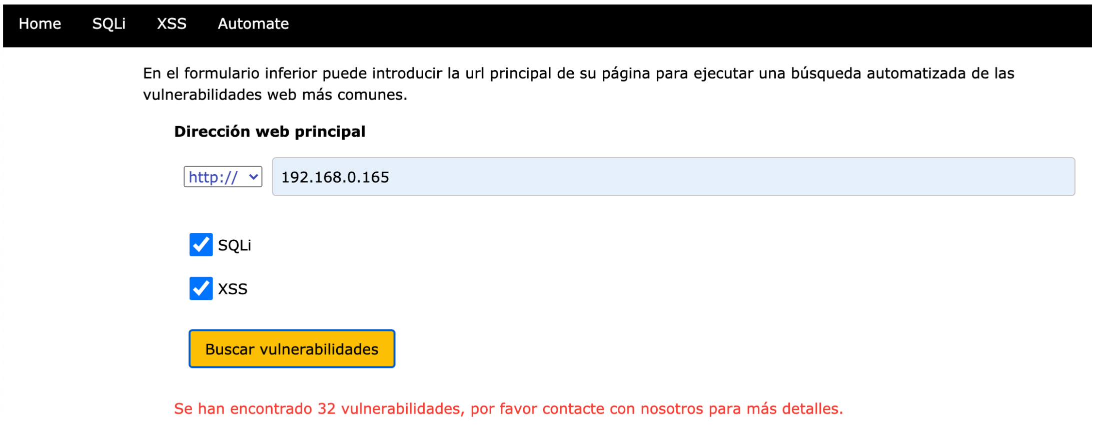
In addition to this, at the end of the questionnaires, the user is shown, visually, through various charts which note he has obtained and a comparison with other users' average scores. It is possible to switch between different types of charts and to obtain detailed information about the different pillars' threats scores.



Puedes hacer clic en cualquiera de las categorías para obtener información detallada acerca de cada una de ellas.

Illustration 36 - Risk analysis application results

Automated penetration test application, also programmed in PHP and JavaScript, with no database connection, implemented during the second half of the project. Its main goal is to provide visibility on web vulnerabilities, why they are caused and how to prevent them. This application is divided in two functionalities: firstly, a demonstrative section which main goal is to provide visibility on web vulnerabilities, why they are caused and how to prevent them; secondly, a penetration test tool focused on identifying possible SQL injections and XSS vulnerabilities at a specific website.



En el formulario inferior puede introducir la url principal de su página para ejecutar una búsqueda automatizada de las vulnerabilidades web más comunes.

Dirección web principal

http:// 192.168.0.165

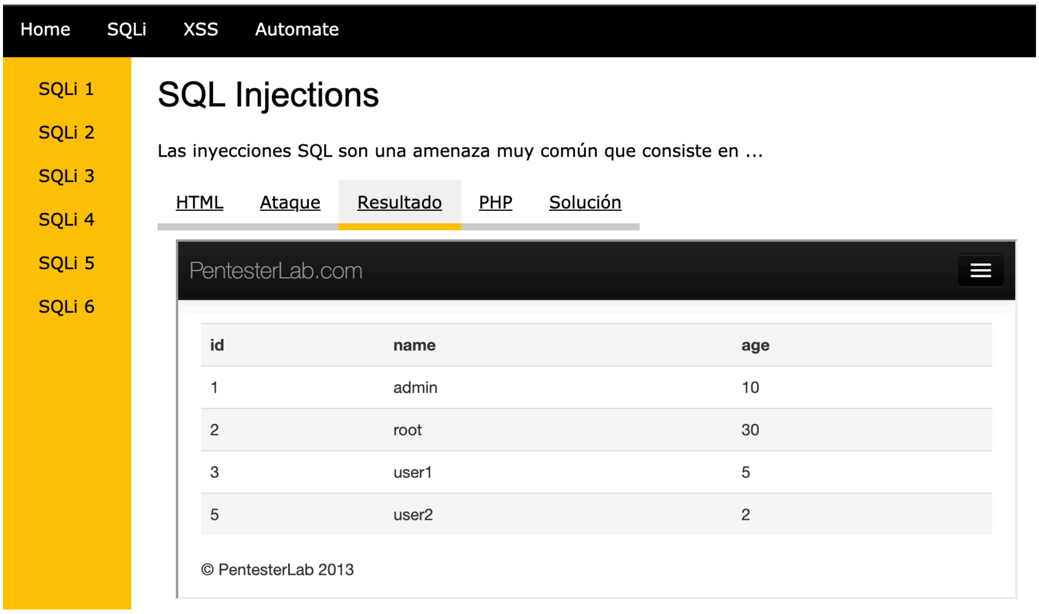
☑ SQLi

☑ XSS

Se han encontrado 32 vulnerabilidades, por favor contacte con nosotros para más detalles.

Illustration 37 - Automated penetration test results

In addition, for this second product, a structure was also created to promote the growth of the penetration testing didactic functionality. This structure allows to easily add new types of attacks, with their respective examples and giving the possibility to the user to show in a personalized way such examples.



SQL Injections

Las inyecciones SQL son una amenaza muy común que consiste en ...

id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

© PentesterLab 2013

Illustration 38 - Demonstrative attacks content page

Furthermore, automated attacks ended up with good accuracy results, it was tested against `Web for pentester I`, which has 40 known vulnerabilities, our collection of attacks should contain exploits for 29 of them and finally we find 32 vulnerabilities but 3 of them are confirmed false positives.

As for the outcome of the project planning, in the following illustrations we can see how the initial planning was affected and what the final result was:

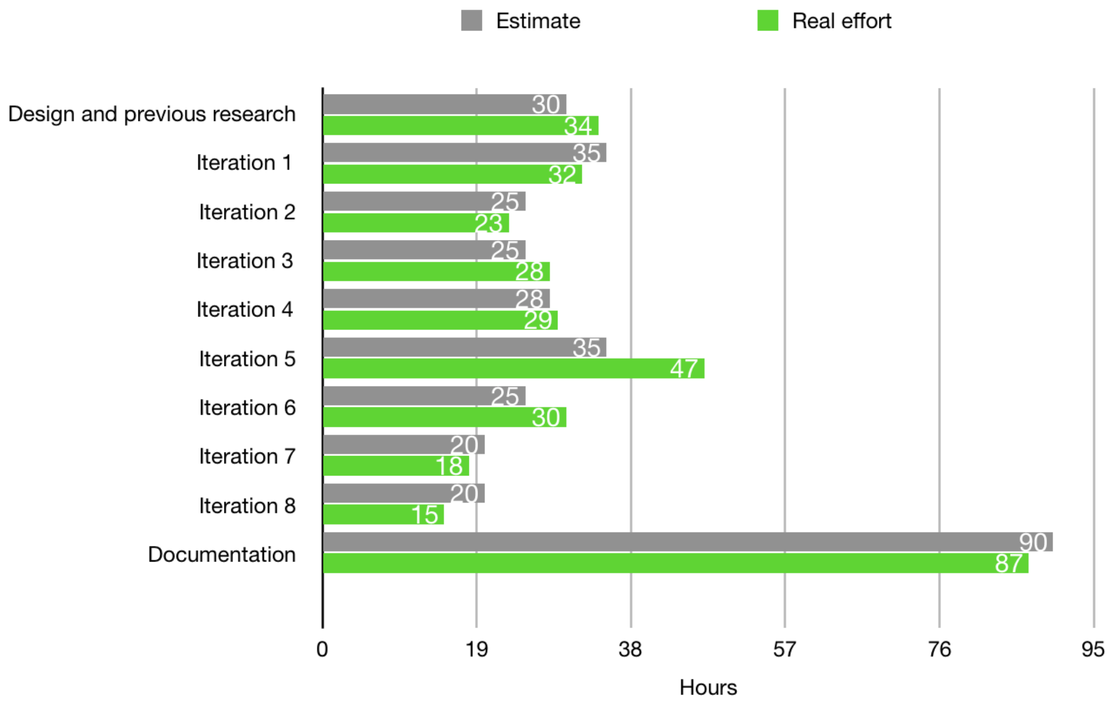


Illustration 39 - Comparison between estimate and real work time per iteration

The implementation proceeded normally, there were some changes in the initial planning, which are shown in Illustration 39, but no insurmountable obstacle was found that paralyzed the course of the project.

Finally, the total time spent on this project as follows:

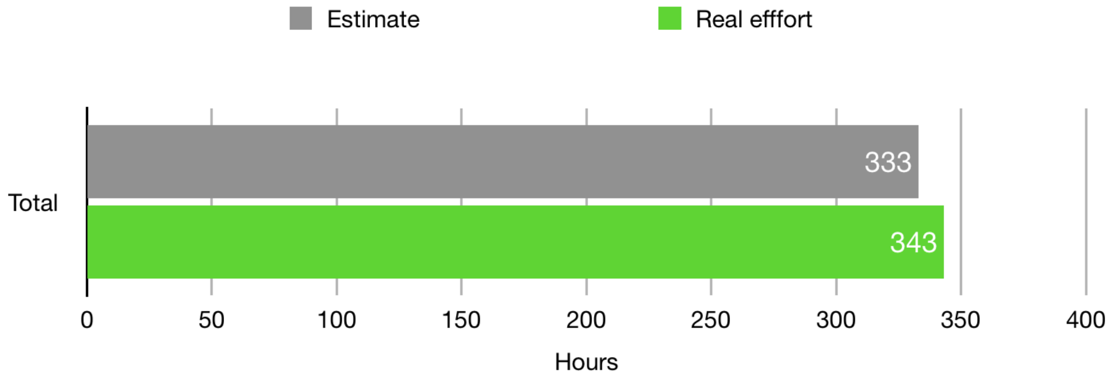


Illustration 40 - Comparison between total estimate and total real work time

Finally, in this project we have implemented 15 user stories, divided in 28 tasks and their corresponding acceptance test which add up to a total of 56 tests in total of which all have been passed successfully.

7. Economic study

7.1. Costs breakdown

7.1.1. Material costs

Within the materials used for the development of this project the costs are distributed as follows:

	Price	Lifetime	Airtime	Real cost
Macbook pro 13''	1.200€	5 years [17]	2 months	40€
Peripherals	60€	6 years [18]	2 months	1,66€
25'' Screen	109€	30.000 hours [19]	343 hours	1,19€
			Total	**42,85€**

Table 24 - Material costs

7.1.2. Human costs

The costs derived from the personnel necessary for the implementation of the project are shown in the next table, according to the roles played and the time spent in each phase of the development:

	Cost / Hour	Time spent (hours)	Total cost
Designer	15€/hour [20]	34	510€
Developer	20€/hour [21]	309	6.180 €
	Total	**343**	**6.690€**

Table 25 - Human costs

7.1.3. Infrastructure costs

During the 2 months in which the project was developed, a space was needed to work and consequently this results in some infrastructure costs which are listed below:

	Cost / month	Airtime	Total cost
Rent	200€/month	2 months	400€
Water	20€/month	2 months	40€
Electricity	50€/month	2 months	100€
Internet	50€/month	2 months	100€
		Total	**640 €**

Table 26 - Infrastructure costs

7.1.4. Total costs

Finally, we can calculate the total sum of the costs associated with the development of this project:

Material costs	Human costs	Infrastructure costs	Total costs
42,85€	6.690€	640 €	7.372,85€

Table 27 - Total costs

In any case, it should be remembered that what was developed during this project is only a prototype, but that it has a great potential and can be improved to achieve a complete final product. For this, taking into account the pending user stories and the improvement proposals, it is estimated that 50-70 more hours of work would be needed, which would mean investing around 15-20% more of effort and the total cost of the final product would be about 8.478,78-8.847,42€. For completing the Risk analysis, Penetration test could be extended endlessly.

8. Conclusion

The two solutions provided as result of this project fully met the initial objectives proposed. The company is happy with the adaptation of the risk analysis that they requested and think that the tool for penetration testing is a good complement for their purposes. The two products have great potential and good prospects of being used and further developed by the company in the future.

I personally enjoyed developing this project, especially the Penetration test application, since I had a lot of interest in getting started in the world of cybersecurity and wanted to learn first-hand guided by professionals from the sector. I have greatly enriched myself in terms of knowledge throughout its development and I now have a better understanding of how the web environment works and of the prevention measures I should put in place in future developments.

In relation to the Risk analysis application, what I least liked was the DREAD methodology. It is too generic and now I understand why Microsoft and other developers stopped using it, sometimes it is frustrating to try to find a score for one of the key points of a very specific threat or a nonvirtual environment. Although I found it challenging and enjoyed solving it, I would not choose DREAD for risk analysis again.

Finally, I would love to be able to continue developing a complete automated attack tool that covers a wide variety of vulnerabilities. I know it would be a very enriching experience, whether my career ends up focusing on cybersecurity or on software development, because it provides very specific insights into how systems or different programming languages work and you learn to identify potential vulnerabilities simply by having a better understanding of how they perform.

8.1. Future work

Although it has great functionality, this project is only a prototype and we can still take various actions to improve it and make it a more complete product:

- Firstly, it would be nice to be able to refine the attack automator a bit to improve its accuracy and avoid false positives, as well as increase its collection of attacks so that it is able to find more vulnerabilities than it can find right now.
- US1.1 - Resumption of questionnaire by hash code, could not be carried out in this first version of the project but it is customer request and it is considered that it could contribute to the application.

- US03 – Assistant, was also not developed in this first phase but we thought it would be a good way to remove boredom from the questionnaires to have a character that makes this monotonous experience more enjoyable.

- [NEW] US12 – Risk analysis recommendations, after users receive their results it will be nice to provide them some feedback on how they can reduce their risk by a series of recommendations for mitigating the possible threats. This means the generation of a pdf with these advices. With an estimation of 20 hours.

9. Bibliography

[1] «GEMED Soluciones,» [En línea]. Available: https://www.gemedsoluciones.es/. [Último acceso: May 2020].

[2] "Definitions," [Online]. Available: https://tools.ietf.org/html/rfc4949. [Accessed August 2020].

[3] "DREAD," [Online]. Available: https://www.logixconsulting.com/2019/12/18/what-is-the-dread-cybersecurity-model/. [Accessed July 2020].

[4] "OWASP," [Online]. Available: https://owasp.org/. [Accessed May 2020].

[5] "OWASP Top Ten," [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed May 2020].

[6] "PentesterLab," [Online]. Available: https://pentesterlab.com/. [Accessed August 2020].

[7] "Google Gruyere," [Online]. Available: https://google-gruyere.appspot.com/. [Accessed August 2020].

[8] "bWAPP an extremely buggy web app!," [Online]. Available: http://www.itsecgames.com/. [Accessed August 2020].

[9] "OWASP WebGoat," [Online]. Available: https://owasp.org/www-project-webgoat/. [Accessed August 2020].

[10] E. C. E. Y. E. S. E. ESPAÑOL, "Espacio Europeo de Educación Superior," [Online]. Available: http://www.eees.es/pdf/credito-europeo.pdf. [Accessed Junio 2020].

[11] "Extreme Programming - A Gentle introduction," [Online]. Available: www.extremeprogramming.org. [Accessed Junio 2020].

[12] "Extreme Programming Ingeniería de Software," [Online]. Available: http://ingenieriadesoftware.mex.tl/52753_xp---extreme-programing.html. [Accessed Junio 2020].

[13] "Trello," [Online]. Available: https://trello.com/. [Accessed Junio 2020].

[14] "Diagrama De Gantt, Microsoft Excel, Plantilla PNG," [Online]. Available: https://www.freepng.es/png-4tx15x/. [Accessed Octubre 2019].

[15] [Online]. Available: https://www.welivesecurity.com/la-es/2015/04/29/vulnerabilidad-xss-cross-site-scripting-sitios-web/. [Accessed August 2020].

[16] [Online]. Available: http://iprodev.github.io/PHP-XML-Sitemap-Generator/. [Accessed August 2020].

[17] «Airtime macbook pro,» [En línea]. Available: https://www.ticpymes.es/tecnologia/noticias/1111246049504/vida-util-de-tecnologia-apple-puede-llegar-12-anos.1.html#:~:text=La%20vida%20%C3%BAtil%20de%20la,a%C3%B1os%20%7C%20Noticias%20%7C%20Tecnolog%C3%ADa%20%7C%20TicPymes&text=Si%20hay%20algo%20que%20ca. [Último acceso: julio 2020].

[18] «Airtime keyboard,» [En línea]. Available: https://www.wikiversus.com/informatica/cuanto-tiempo-equivalen-millones-de-pulsaciones/. [Último acceso: julio 2020].

[19] «Airtime screen,» [En línea]. Available: https://hardzone.es/2019/01/13/vida-util-monitor-antes-rompa/. [Último acceso: julio 2020].

[20] "Glassdoor - UX designer," [Online]. Available: https://www.glassdoor.es/Sueldos/ux-designer-sueldo-SRCH_KO0,11.htm. [Accessed August 2020].

[21] "Glassdoor - Programador junior," [Online]. Available: https://www.glassdoor.es/Sueldos/programador-junior-sueldo-SRCH_KO0,18.htm. [Accessed August 2020].

[22] "CVE Details," [Online]. Available: https://www.cvedetails.com/cve/CVE-2007-5900/. [Accessed July 2020].

[23] "CVE Details," [Online]. Available: https://www.cvedetails.com/cve. [Accessed July 2020].

[24] B. Leban, M. Bendre and P. Tabriz, "Google Gruyere," [Online]. Available: https://google-gruyere.appspot.com/. [Accessed August 2020].

[25] "Glassdoor - Project Manager," [Online]. Available: https://www.glassdoor.es/Sueldos/project-manager-junior-sueldo-SRCH_KO0,22.htm. [Accessed August 2020].

[26] "Security definition," [Online]. Available: https://searchsecurity.techtarget.com/definition/security. [Accessed August 2020].

10. Annex

10.1. Annex 1: Project proposal

Nombre alumno:	Esther Pérez Jerez
Titulación:	Grado en Ingeniería Informática
Curso académico:	2019 - 2020

1. TÍTULO DEL PROYECTO

Test de seguridad de la información para empresas

2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TEMA A TRATAR

El proyecto consiste en la elaboración de un aplicativo web dirigido a empresas que, por un lado, realice mediante una serie de encuestas un análisis de riesgos de la misma, basado en la metodología DREAD, y por otro lado, un módulo específico que analizará los sitios web de la entidad en busca de vulnerabilidades, basado en el OWASP Top Ten.

3. OBJETIVOS DEL PROYECTO

- Creación de una aplicación web que realice encuestas al cliente y las adapte en función de las respuestas obtenidas.
- Análisis de riesgos con metodología DREAD de las respuestas obtenidas.
- Creación de una herramienta específica para análisis de vulnerabilidades de sitios web.
- Presentar los resultados de los análisis anteriores al cliente así como ciertos consejos o recomendaciones de mejora.

4. METODOLOGÍA

La metodología se establecerá en las primeras fases del proyecto.

5. PLANIFICACIÓN DE TAREAS

Las tareas se definirán de acuerdo a los objetivos. Serán fijadas de forma concreta durante el desarrollo del proyecto.

6. OBSERVACIONES ADICIONALES

El tutor del proyecto será Jaime Font Burdeus

10.2. Annex 2: Meeting notes

REUNIÓN: 00

Fecha:	25/02/2020		
Hora comienzo:	17:30	**Hora finalización:**	18:16
Lugar:	Universidad San Jorge		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Validación de las historias de usuario.	
5	División del proyecto en dos aplicaciones diferenciadas	
6	Validación diseño inicial de la estructura de la primera parte del proyecto	
7	Elección de las historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	001
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Implementación de las historias de usuario US01 y US02	No aplica	Esther Pérez
002			
003			
004			

REUNIÓN: 01

Fecha:	10/03/2020		
Hora comienzo:	17:15	**Hora finalización:**	18_06
Lugar:	Universidad San Jorge		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 1.	
5	Fallaron los siguientes tests y se añaden a la historia de usuario US2.1 – Questionnaires' interface issues.	001
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	002
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Redacción de la historia de usuario US2.1	No aplica	Esther Pérez
002	Implementación de la historia de usuario US09		
003			
004			

REUNIÓN: 2.1

Fecha:	20/04/2020		
Hora comienzo:	12:00	**Hora finalización:**	13:15
Lugar:	Skype		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Sergio Barboza, José Ramón Ansó, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Demostración de las historias de usuarios implementadas hasta la fecha.	
5	Comentarios recibidos por parte del cliente: ○ C1: Los usuarios pueden registrarse o no, por lo que la columna del perfil de usuario no sería la mejor. ○ C2: Para aquellos usuarios que no estén registrados, permitir por medio de un hash la reanudación del cuestionario, si no tienen el tiempo necesario para completarlo al mismo tiempo. ○ C3: El cliente propone rellenar el hueco del perfil con un asistente que proporciona información relevante sobre la pregunta actual para ayudar al usuario en su comprensión. Además, sugieren proporcionar cierta inteligencia al asistente, como, por ejemplo, interactuar con el usuario si ha estado inactivo durante mucho tiempo con mensajes de ayuda o informar del tiempo restante para terminar el cuestionario.	
7	**Otros asuntos**	
8	**Próxima reunión** A determinar	

REUNIÓN:	**2.2**

Fecha:	20/04/2020		
Hora comienzo:	18:02	**Hora finalización:**	18:47
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	**Asunto**	**Acuerdo**
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 2.	
5	Comentarios recibidos por parte del cliente. ○ C1: US2.1 – Questionnaires' interface issues ○ C2: US1.1 – Resumption of questionnaires by hash code ○ C3: US03 – Assistant	001
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	002
7	**Otros asuntos**	
8	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	**Acuerdo**	**Plazo**	**Responsable**
001	Redacción de las historias de usuario derivadas de los cambios propuestos por el cliente.	No aplica	Esther Pérez
002	Implementación de las historias de usuario US04 y US05	No aplica	Esther Pérez
003			
004			

REUNIÓN: 03	

Fecha:	04/05/2020		
Hora comienzo:	17:13	**Hora finalización:**	18:04
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 3.	
5	Redacción de una nueva historia de usuario US5.1 - Results visualization comparison, para aportar una comparativa a los gráficos de los resultados.	
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	001
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Implementación de las historias de usuario US06, US6.1 y US5.1	No aplica	Esther Pérez
002			
003			
004			

REUNIÓN: 4.1

Fecha:	26/06/2020		
Hora comienzo:	12:00	**Hora finalización:**	13:32
Lugar:	Skype		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Sergio Barboza, José Ramón Ansó, Esther Pérez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Demostración del resultado de la primera parte del proyecto.	
5	Se recibe el siguiente comentario de mejora: ○ C4: La puntuación DREAD puede ser confusa para los clientes ya que 10 es la peor puntuación alcanzable, pero en nuestra base diaria pensamos en esta puntuación como la óptima. Por lo tanto, decidieron que los gráficos deben ser modificados y mostrar la puntuación en valores negativos para ser más esclarecedor y evitar confusiones.	
6		
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

REUNIÓN: 4.2

Fecha:	26/06/2020		
Hora comienzo:	17:30	**Hora finalización:**	18:40
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 4.	
5	Comentarios recibidos por parte del cliente. ○ C4: US5.2 – Results visualization modifications.	001
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	002
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Redacción de la nueva historia de usuario US5.1	No aplica	Esther Pérez
002	Implementación de las historias de usuario US08 y US8.1	No aplica	Esther Pérez

REUNIÓN: 05

Fecha:	13/07/2020		
Hora comienzo:	18:05	**Hora finalización:**	18:36
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 5.	
5		
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	01
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Implementación de la historia de usuario US10	No aplica	Esther Pérez
002			
003			
004			

REUNIÓN: 06

Fecha:	03/08/2020		
Hora comienzo:	17:00	**Hora finalización:**	17:23
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 6.	
5		
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	002
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Implementación de la historia de usuario US07	No aplica	Esther Pérez
002		No aplica	Esther Pérez
003			
004			

REUNIÓN:	**07**

Fecha:	18/08/2020		
Hora comienzo:	18:00	**Hora finalización:**	18:43
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 7.	
5		
6	Historias de usuarios a desempeñar en la siguiente iteración y redacción de sus tests de aceptación	002
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Implementación de la historia de usuario US11	No aplica	Esther Pérez
002		No aplica	Esther Pérez
003			
004			

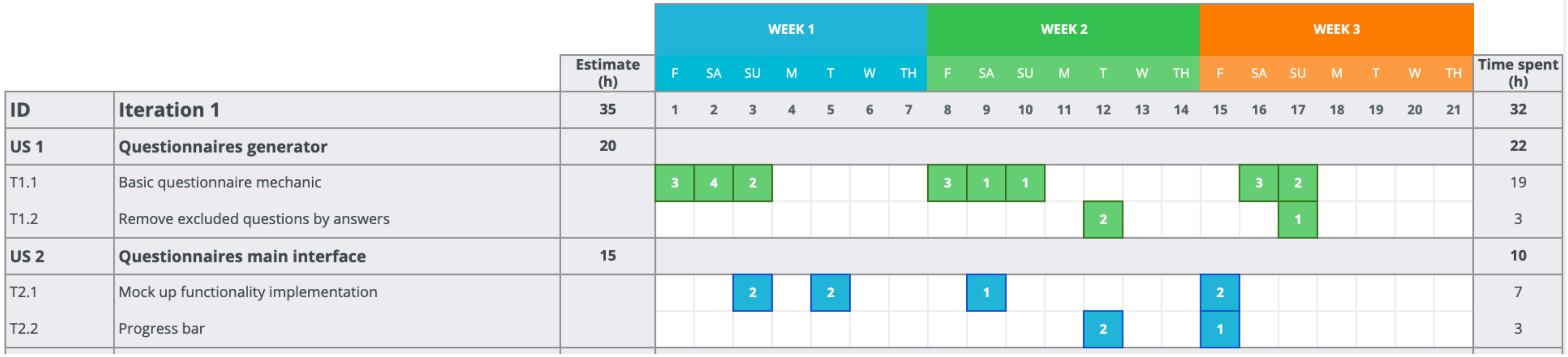
REUNIÓN: 08

Fecha:	01/09/2020		
Hora comienzo:	18:00	**Hora finalización:**	18:34
Lugar:	Microsoft Teams		
Elabora acta:	Esther Pérez Jerez		
Convocados:	Jaime Font Burdeus, Esther Pérez Jerez		

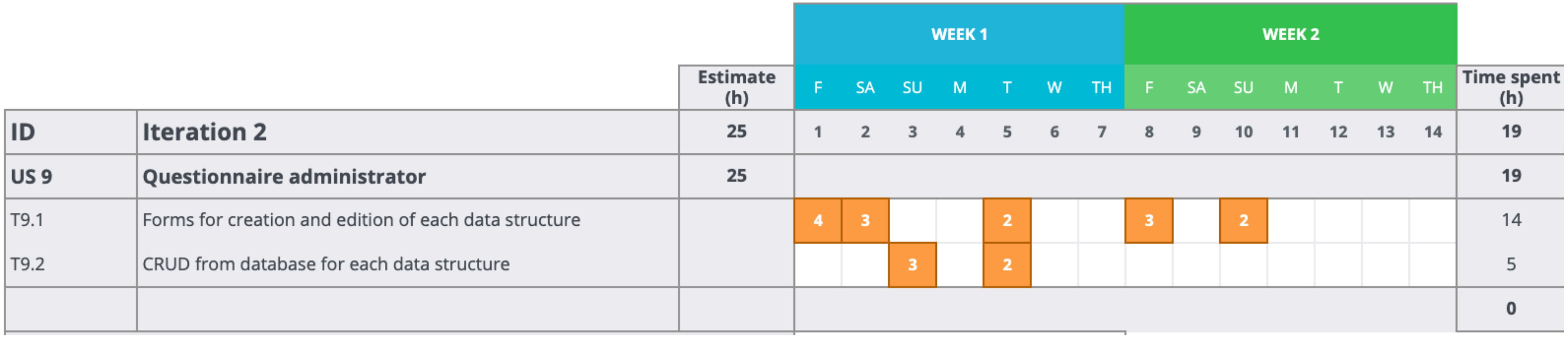
Orden del día / Acta

No.	Asunto	Acuerdo
1	**Disculpas por ausencia**	
2	**Aprobar última acta**	
3	**Asuntos pendientes última acta**	
4	Ejecución de los tests de aceptación correspondientes al trabajo de la iteración 8.	
5	Demostración del estado final de la segunda parte del proyecto.	
6		
7		
8		
9	**Otros asuntos**	
10	**Próxima reunión** A determinar	

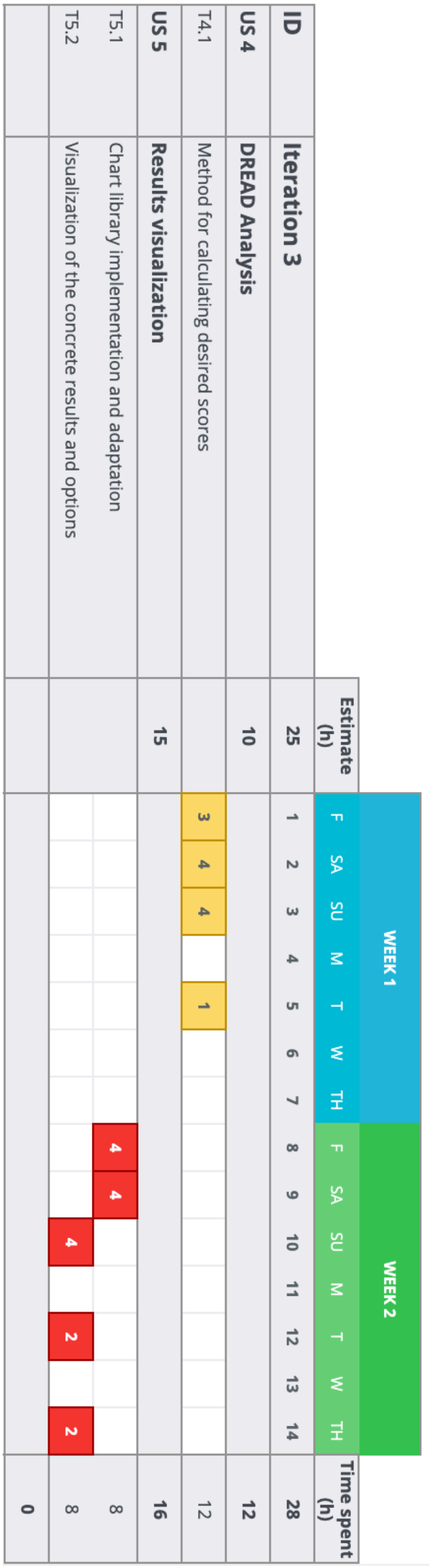
10.3. Annex 4: Workflow



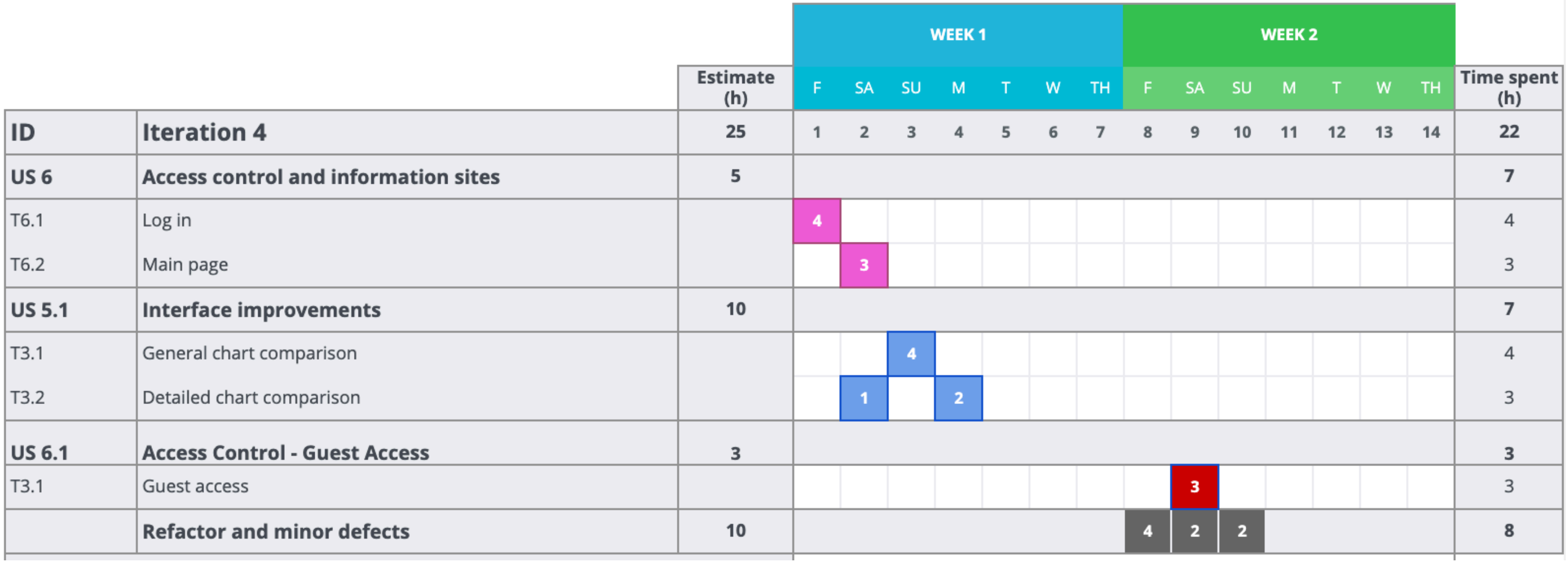
			WEEK 1							WEEK 2							WEEK 3							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 1**	**35**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	**32**
US 1	**Questionnaires generator**	**20**																						**22**
T1.1	Basic questionnaire mechanic		3	4	2					3	1	1						3	2					19
T1.2	Remove excluded questions by answers													2					1					3
US 2	**Questionnaires main interface**	**15**																						**10**
T2.1	Mock up functionality implementation				2		2				1						2							7
T2.2	Progress bar													2			1							3



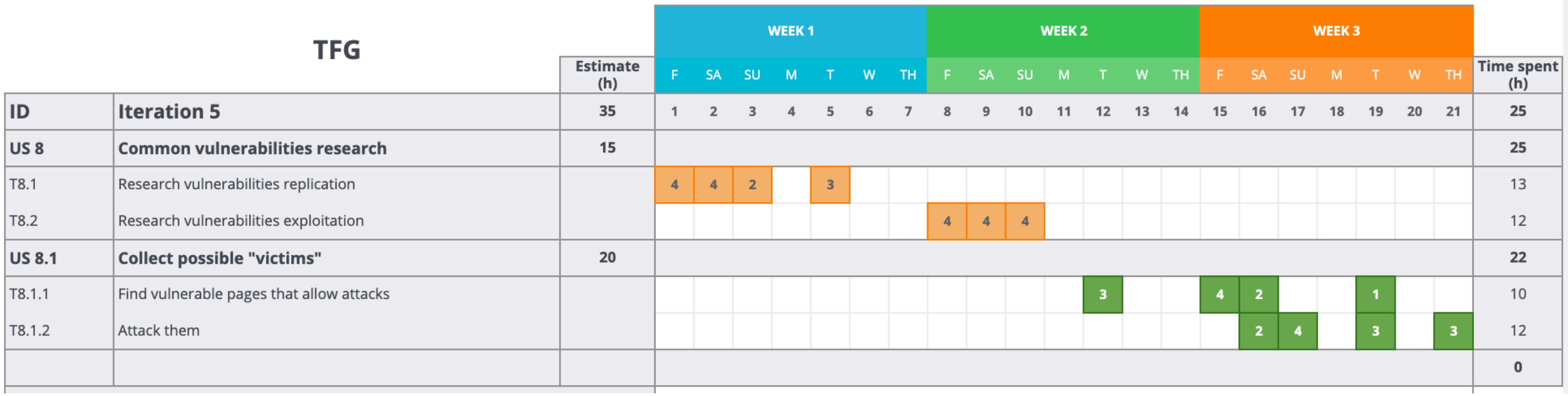
			WEEK 1							WEEK 2							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 2**	**25**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**19**
US 9	**Questionnaire administrator**	**25**															**19**
T9.1	Forms for creation and edition of each data structure		4	3			2			3		2					14
T9.2	CRUD from database for each data structure				3		2										5
																	0



			WEEK 1							WEEK 2							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 3**	**25**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**28**
US 4	**DREAD Analysis**	**10**															**12**
T4.1	Method for calculating desired scores		3	4	4		1										12
US 5	**Results visualization**	**15**															**16**
T5.1	Chart library implementation and adaptation									4	4						8
T5.2	Visualization of the concrete results and options											4		2		2	8
																	0

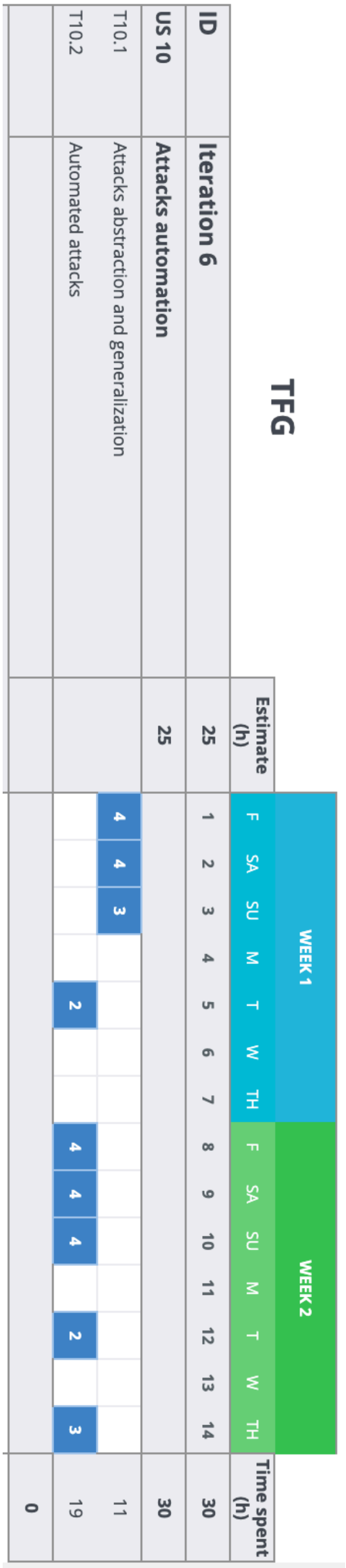


			WEEK 1							WEEK 2							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 4**	**25**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**22**
US 6	**Access control and information sites**	**5**															**7**
T6.1	Log in		4														4
T6.2	Main page			3													3
US 5.1	**Interface improvements**	**10**															**7**
T3.1	General chart comparison				4												4
T3.2	Detailed chart comparison			1		2											3
US 6.1	**Access Control - Guest Access**	**3**															**3**
T3.1	Guest access										3						3
	Refactor and minor defects	**10**								4	2	2					**8**



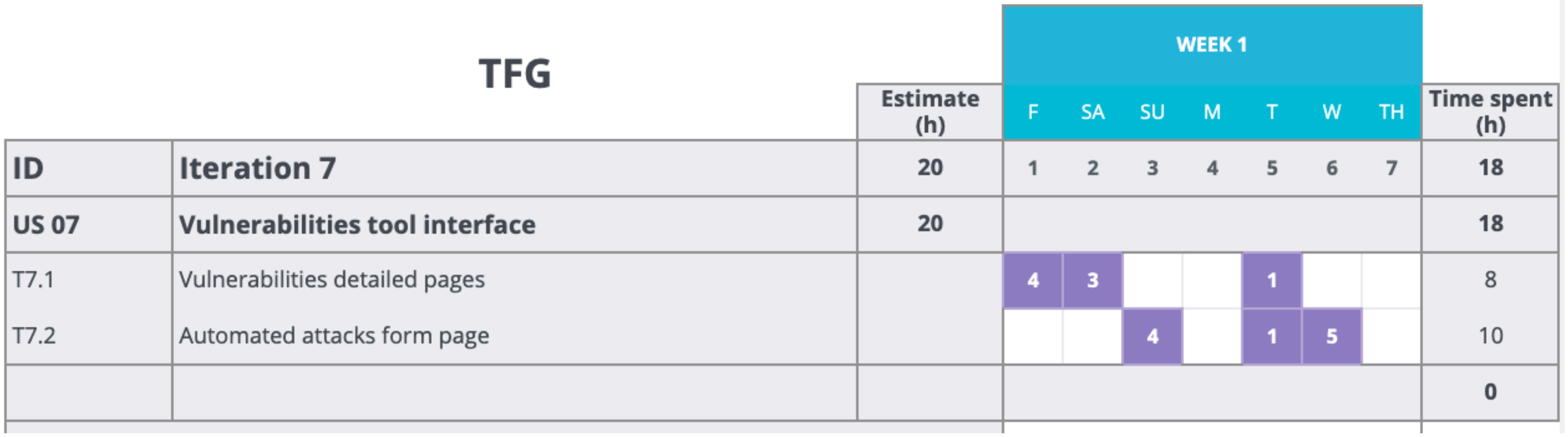
TFG

			WEEK 1							WEEK 2							WEEK 3							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 5**	**35**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	**25**
US 8	**Common vulnerabilities research**	**15**																						**25**
T8.1	Research vulnerabilities replication		4	4	2		3																	13
T8.2	Research vulnerabilities exploitation									4	4	4												12
US 8.1	**Collect possible "victims"**	**20**																						**22**
T8.1.1	Find vulnerable pages that allow attacks													3			4	2			1			10
T8.1.2	Attack them																	2	4		3		3	12
																								0



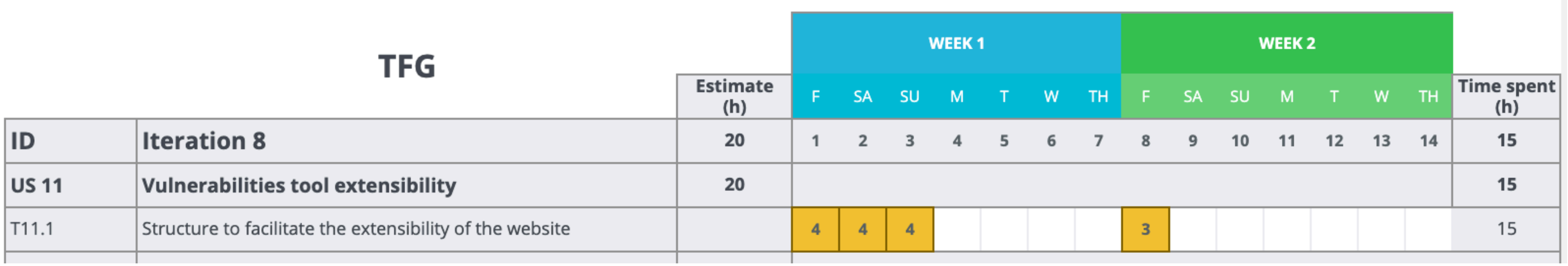
TFG

			WEEK 1							WEEK 2							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 6**	**25**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**30**
US 10	**Attacks automation**	**25**															**30**
T10.1	Attacks abstraction and generalization		4	4	3												11
T10.2	Automated attacks						2			4	4	4		2		3	19
																	0



TFG

			WEEK 1							
		Estimate (h)	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 7**	**20**	1	2	3	4	5	6	7	**18**
US 07	**Vulnerabilities tool interface**	**20**								**18**
T7.1	Vulnerabilities detailed pages		4	3			1			8
T7.2	Automated attacks form page				4		1	5		10
										0



TFG

			WEEK 1							WEEK 2							
		Estimate (h)	F	SA	SU	M	T	W	TH	F	SA	SU	M	T	W	TH	Time spent (h)
ID	**Iteration 8**	**20**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**
US 11	**Vulnerabilities tool extensibility**	**20**															**15**
T11.1	Structure to facilitate the extensibility of the website		4	4	4					3							15