eFontys Banking Website

Final Report

SePr 2016

Radu Stoica, Armin Roushan, Jan-Niklas Schneider, Georgiana Manolache, Nena O’Driscoll, Raditya Pratama

Table of Contents

[1 Introduction 2](#_Toc464747651)

[2 Design Overview 3](#_Toc464747652)

[2.1 Introduction 3](#_Toc464747653)

[2.2 User Group 3](#_Toc464747654)

[2.2.1 Customer 3](#_Toc464747655)

[2.2.2 Administrator 3](#_Toc464747656)

[2.3 Network Architecture 3](#_Toc464747657)

[3 Threat analysis 5](#_Toc464747658)

[3.1 Assets management 6](#_Toc464747659)

[3.2 Authorization requirements 7](#_Toc464747660)

[3.3 Risk analysis 7](#_Toc464747661)

[4 Secure design 9](#_Toc464747662)

[4.1 Attack Trees 9](#_Toc464747663)

[4.2 Misuse Cases 9](#_Toc464747664)

[4.2.1 Injection of malicious SQL string 9](#_Toc464747665)

[4.2.2 Insertion of XSS code 9](#_Toc464747666)

[4.2.3 Cross-Site Request Forgery 9](#_Toc464747667)

[4.2.4 Brute force attack 9](#_Toc464747668)

[4.2.5 Software mitigation 9](#_Toc464747669)

[5 Implementation choices 12](#_Toc464747670)

[6 Code example 12](#_Toc464747671)

[7 Group peer assessment 12](#_Toc464747672)

[8 Personal Evaluation 12](#_Toc464747673)

[8.1 Radu 12](#_Toc464747674)

[9 References 12](#_Toc464747675)

# Introduction

Security breaches have been increasing considerably over the last decade, becoming an everyday problem for both low and high-profile organizations. In fact, hacked websites have become so common that typically only the biggest data breaches capture enough attention to make headlines (WhiteHat Security, 2015). Breach reports are being posted weekly, presenting in detail information about data exposure events. For example ITRC[[1]](#footnote-1) (IDTheftResourceCenter), which performs breach reports on United States websites, cover a wide range of categories where data might be put at risk. According to ITRC report for 2015 (Data Breach Report, 2015), banking and fanatical breaches have been ranked third most vulnerable, after business and medical and healthcare.

In this report, we present possible banking breaches and best-practices to improve security of an online financial service. We have implemented a simple online banking system and enhanced its security infrastructure. Finally, pen-testing is added to further test the security of our website.

# Design Overview

## Introduction

This chapter will cover the pre-knowledge, final design decisions and security regarding the SPA Fontys e-banking system called eFontys. The application is written in PHP, HTML5, CSS3 and MySQl.

## User Group

The application allows for two different roles, namely the administrator and the customer. A new customer can be added only through the administrator page.

### Customer

The customer has an account with the ability to:

* Edit account details
* Check current balance
* Deposit money into own account
* Transfer money from own account to recipient
* Logout

### Administrator

The administrator’s account has the same abilities as the customer but can also:

* Add and remove users
* Edit other account details

## Network Architecture

The network architecture looking as follows. With eFontys being hosted on the Athena server. The database and server are connected through the internet.

Figure ‑ Network Architecture Diagram



As shown on the Network Architecture, you can see that the Fontys bank application is hosted by a private server, which is a hosting service provided by Fontys called Athena.

The web application itself had really basic banking functionalities which is:

* Login
* Check balance
* Deposit Money
* Transfer Money

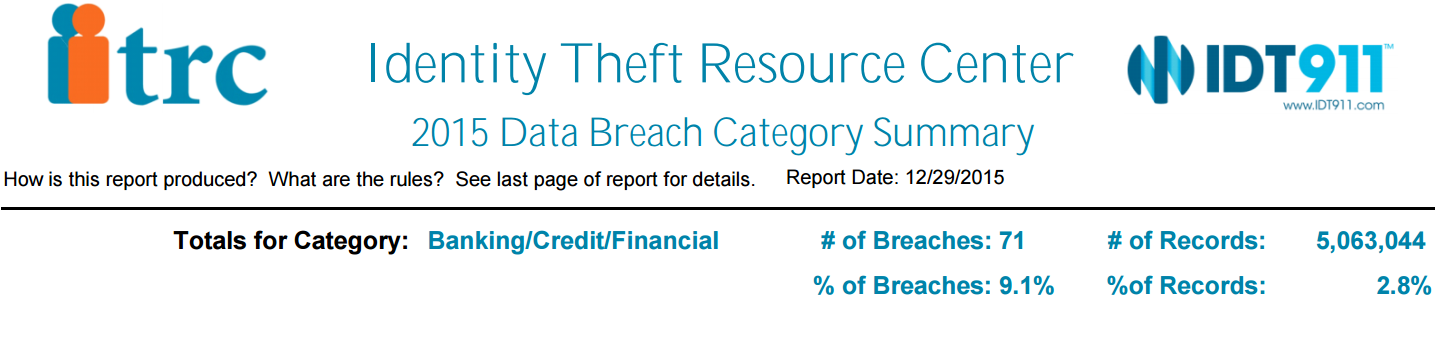
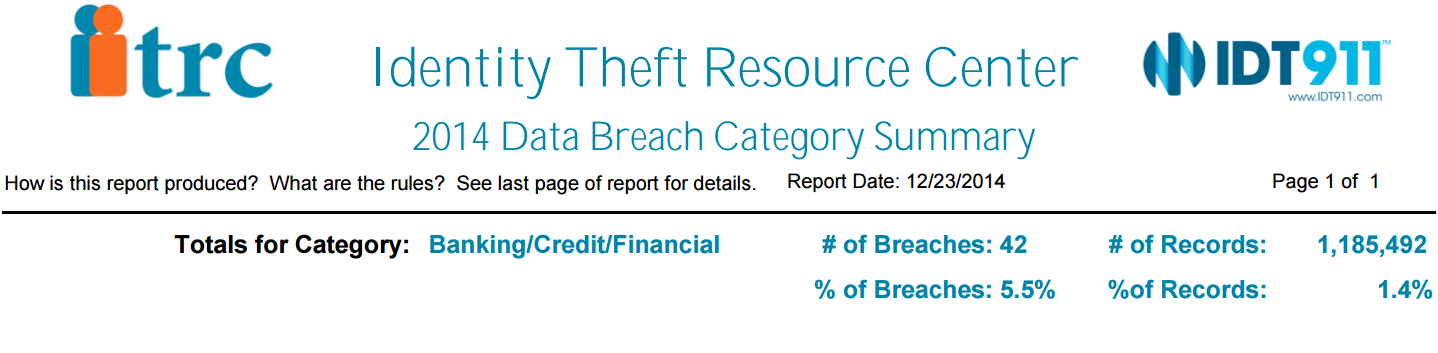
The web application is using MySQL to record all the data. The database has two tables which is ‘user’ and ‘account’ table that contain all the user records.

# Threat analysis

Financial services and especially online banking services have always been a target for cyber criminals. The number of reported data security violation continues to increase while becoming more diverse and sophisticated.  According to ITCR (Identity Theft Resource Center) Annual Breach Reports, only in the United States in 2015 the recorded banking sector braches - 71 in total - were nearly double in number compared to 2014 with 42 breaches (Figure 4-1). Between 2010-2014, banks experienced attacks with losses over 30.000 records due to hacking or poor security as shown in Figure 4-2. Among the affected financial services organizations there is JP Morgan Chase, European Central Bank, US Federal Reserve Bank of Cleveland, Citigroup (World's Biggest Data Breaches, 2016). In a study conducted by the Verizon Business RISK Team in 2009, it was reported that the cyber criminals focus on the theft of personal identification number (PIN) information and their associated credit and debit account information (2009 Data Breach Investigations Report, 2009).

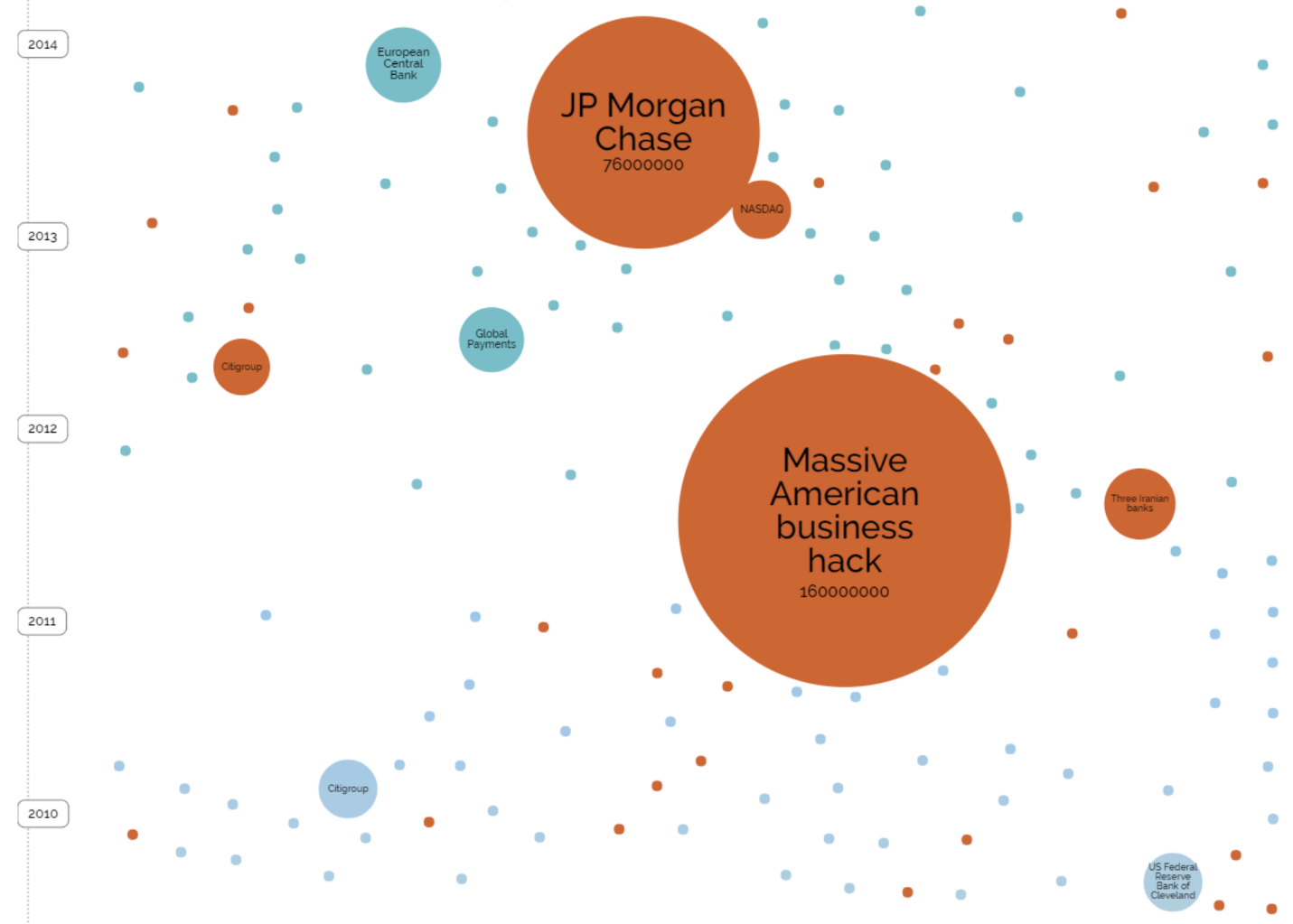
Protecting financial data and assets is, thus, vital for a financial organization. The following subchapters describe the risk analysis of the online banking web application.

Figure ‑ ITRC Stats Summary 2014-2015



*Source: Identity Theft Resource Center Breach* [*http://www.idtheftcenter.org/ITRC-Surveys-Studies/2015databreaches.html*](http://www.idtheftcenter.org/ITRC-Surveys-Studies/2015databreaches.html)

Figure ‑ Major financial data breaches 2010-2014 with losses greater than 30.000 records.

**

*Source: World’s Biggest Data Breaches* [*http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/*](http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/)

## Assets management

The online banking application must maintain the majority of information at a high security level. Apart from the log record, data elements are strictly confidential, private, integral and availability is minimal. Table 4-1 describes the choses level of each data sets.

Table ‑ CIA Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bank data | Data elements | Data classification  (C, I, A) | Privacy | Explanation |
| User account | Username  Password (hash) | **H, H, H** | **H** | User credentials are strictly private and should not be available to anyone but user. All data classifications are important especially for bank reputation and financial loss. |
| User information | Name  Email  Phone number(s) Address | **H, H, H** | **H** | User information is highly private and confidential. All information must be confidential. |
| Account information | IBAN  Amount | **H, H, H** | **H** | The user IBAN is available at user permission (given by user himself) as transactions are performed to other users by specifying these data. |
| Log record | Date  Time  Type  Sender  Recipient | **H, H, M** | **M** | Log reports are available to the web administrator for auditing. Mainly for auditing. |

Legend: H = High, M = Medium.

## Authorization requirements

Within the online banking application, the web admin has highest authority rights over the data; a web administrator is responsible for ***creating*** a new user account, ***read*** data, ***update*** data on user request or ***delete*** user account when customer decides to withdraw his account from the banking service. The same authorization is assigned to the rest of the data groups existent within the web application: user information, account information and log records. The user has limited authority (read, update) over data, to avoid misusages and potential fake users. Table 4-2 gives an overview of the access-rights per user groups.

Table ‑ Authorization Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Group data | User | Web administrator | Anonymous  (website visitor) |
| User account | **U** | **CRUD** | **-** |
| User information | **R, U** | **CRUD** | **-** |
| Account information | **R, U** | **CRUD** | **-** |
| Log record | **R** | **CRUD** | **-** |

Legend: C = Create, R = Read, U = Update, D = Delete.

## Risk analysis

According to SafeNet (Top Online Banking Threats to Financial Service Providers in 2010 , 2010), leading market in protecting financial transactions, there are four majorly applied attacks. Most frequent is ***Phishing***, typically carried out through e-mail or instant messaging, providing links or instructions that direct the user to fraudulent Web sites portrayed as legitimate ones. ***Password Database Theft*** is a threat in which hackers get possession of costumer data from other less protected websites assuming user uses similar user ID and password. By collecting personal information**,** cyber criminals can assume individual identity, also known as ***Identity Theft*** threat. ***Man-in-the-Middle (MitM)*** is this type of threat in which the attacker can actively inject messages of its own into the traffic between the user's machine and the authenticating server. ***Man-in-the-Browser (MitB)*** is a variant of the MitM attack, that infects the user internet browser and inserts itself between the user and the Web browser, modifying and intercepting data sent by the user before it reaches the browser’s security mechanism.

Table 4-3 gives an overview of the possible risks that could occur on an online banking website.

Table ‑ Risk analysis

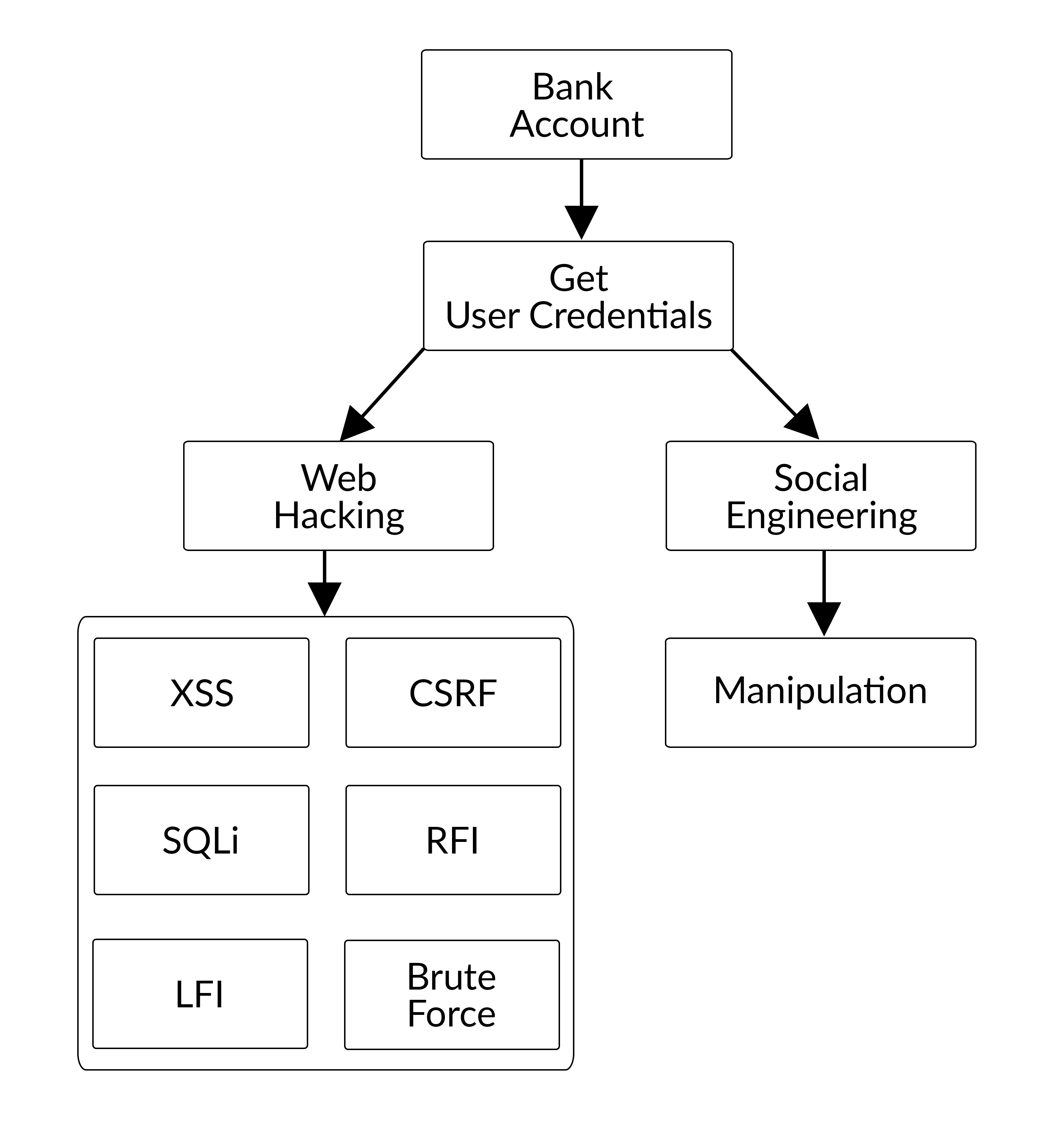
|  |  |  |  |
| --- | --- | --- | --- |
| Threat | Risk | Impact | Needed Security level Risk impact |
| Phishing | **High** | **High** | **High** |
| Password Database Theft | **High** | **High** | **High** |
| Man-in-the-Middle (MitM), Man-in-the-Browser (MitB) | **High** | **High** | **High** |
| Identity Theft | **High** | **High** | **High** |

# Secure design

Before proceeding to the implementation of the website, security design consideration was taken into account. This step was necessary to have better overview of the parts which could be at risk.

## Attack Trees

Figure ‑ Attack trees



## Misuse Cases

### Injection of malicious SQL string

An attacker might try to insert malicious SQL command into the provided inputs on the web application in order to change the balance. SQL injection could also retrieve sensitive information such as user information, or get access to the admin account which is the administrator of the website.

### Insertion of XSS code

The attacker may insert different JavaScript or PHP code lines in order to cause damages to the website or to take control over the web application.

### Cross-Site Request Forgery

The website is creating a session in order for the user to log in. This can be a potential vulnerability of the website. The hacker may try to take advantage of it in order to get sensitive information.

### Brute force attack

By brute forcing the password people may try to retrieve a user password.

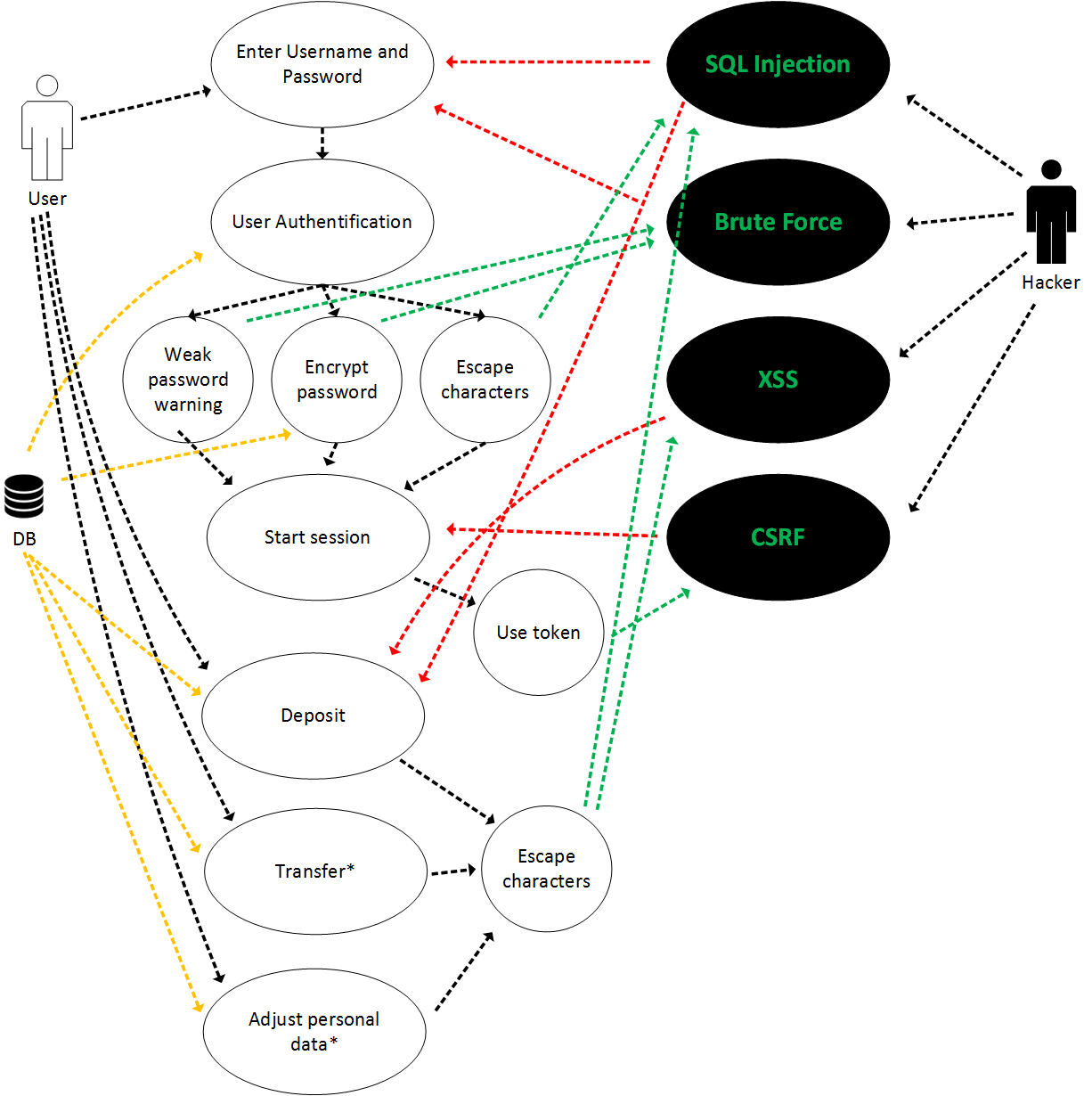
### Software mitigation

To prevent and block unwanted requested from hackers were implemented next functions

* Use escape special characters for inputs in order to prevent SQL injection.
* Use token for PHP session in order to prevent CSRF.
* Encrypt passwords in order to decrease the risk of finding the real password in case of a data leak.
* Check the data inserted in the inputs in order to prevent XSS code.

Below is shown image with possible hacker’s attacks and mitigations:

Figure ‑ Misuses case diagram

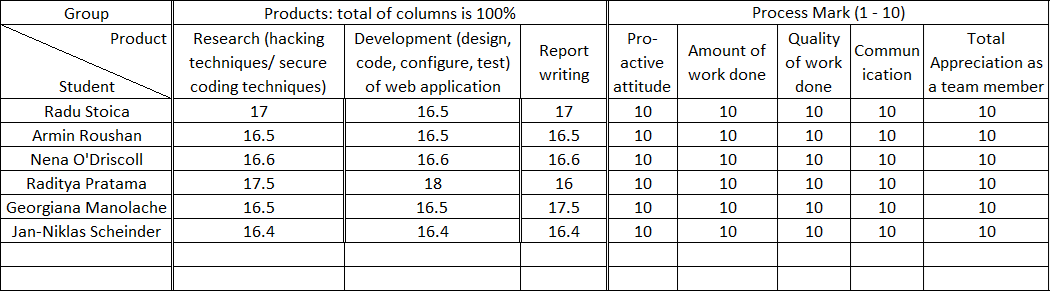


# Implementation choices

The implementation was performed according to breach and hacking prevention research.

# Code example

# Group peer assessment



# Personal Evaluation

## Radu

This course has helped me to expand my knowledge about web development and web site vulnerabilities. I have learned how to operate the Kali Linux OS and how to use some of the software provided by this OS in order to analyse and exploit vulnerabilities on a specific website. I think the pen-testing session was really helpful as it gave us the opportunity to apply our knowledge from this course in a “real” environment. Also, the experience of group work for the web application and the presentations has helped me to develop both technical and soft skills. Another very helpful skill achieved during this course is writing more secure software/web application as now I know what kind of attacks to expect on my applications in the future and how to mitigate them.

## Georgiana

This course has helped me raising awareness in the way I handle security in my own code. It has taught me how to tackle, design and implement a higher secure software/web application. I have learned how to ‘think as a hacker’ during the pen-testing session and learn from other people’s mistakes. Thought the course I was also able to improve my team work and presentation skills.

## Armin

Before I started this course, I did not have any knowledge about secure programing and hacking techniques, but this course helped to enlighten my knowledge in this aspect of web development. I also would like to add that it was nice to get familiar with the tools we used during the course and apply some the techniques in practical.

## Jan-Niklas

The course secure programming showed me that security in web applications shouldn’t be taken lightly. An unsecured website can be very easily targeted by people with the right tools and knowledge. The techniques I learned in this course will help me in future building secure web applications and securing it against hackers. Overall, the course gave good insight into a big issue in modern web technology, security, how to protect against it but also giving practical knowledge.

## Nena

This course not only taught me that exploiting on the web isn’t just for Eliot’s or other modern day wizards, it reconfirmed how every team member has a valuable different view of solving a problem. The problem in this case being noticing vulnerabilities, and consequently either exploit or patch them. In future projects I hope to pay far more attention to the security of the application, as it is indeed a wild wide web.

# References

*2009 Data Breach Investigations Report*. (2009). Retrieved from Verizon Enterprise: http://www.verizonenterprise.com/resources/security/reports/2009\_databreach\_rp.pdf

*Data Breach Report*. (2015). Retrieved from ID Theft Center: http://www.idtheftcenter.org/images/breach/DataBreachReports\_2015.pdf

*Top Online Banking Threats to Financial Service Providers in 2010* . (2010). Retrieved from http://www2.gemalto.com/: http://www2.gemalto.com/email/2010/mitb-2010/whitepaper\_top-threat-to-financial-service-providers-in-2010\_final.pdf

WhiteHat Security. (2015). *Website Security Statistics Report*. Retrieved from Whitehat Security: https://info.whitehatsec.com/rs/whitehatsecurity/images/2015-Stats-Report.pdf

*World's Biggest Data Breaches*. (2016, September). Retrieved from Information is beautiful: http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

1. ITRC = ID Theft Resource Center [↑](#footnote-ref-1)