# Threat Modelling and Risk Management Framework for SecureHelp Application

TDT4237 Group (097)
Stefano Iannello, Eleni Mandana

**Abstract**

An application that aims to help refugees get access to a shelter, food and a safe environment needs to be deployed in a week. An analysis of the application was conducted following the principles of threat modelling and risk management framework, in order to identify potential risks and the countermeasures to mitigate them. The report presents the identified business assets, goals and risks, two misuse cases with their attack trees schemas followed by the identified technical risks with related business risk. In the final section a test plan of ten related technical risk is presented.

*Keywords:* Risk Management Framework, Threat Modelling, Web Application, OWASP Testing Guide, Risk Analysis.

# Contents



# List of Tables



# List of Figures

# 1. Introduction

The goal is to create a safe and efficient digital environment for both the users and the organization.

We started by identifying business assets, goals and risks. We continued with misuse cases and attack tree examples to demonstrate some potential threats and the possible outcomes these might have on the system. Finally we identified technical risks and created a test plan to mitigate them.

# 2. Part 1: Risk management framework

## 2.1. Identified Business Assets

| Business Assets | |
|---|---|
| ID | Description |
| BA1 | Personal information of volunteers |
| BA2 | Personal information of refugees |
| BA3 | Medical equipment/supplies |
| BA4 | Refugee camps |
| BA5 | Sponsors |
| BA6 | QRs |

Table 1: Business Assets

## 2.2. Identified Business Goals

| Business Goals | |
|---|---|
| ID | Description |
| BG1 | Obtain financial sustainability |
| BG2 | Achieve high prestige |
| BG3 | Provide help to the refugees |
| BG4 | Raise people's awareness |
| BG5 | Trusted environment |
| BG6 | Attract volunteers |
| BG7 | Attract sponsors |

Table 2: Business Goals

## 2.3. Identified Business Risks

| Business Risks | | | | |
|---|---|---|---|---|
| ID | Description | Likelihood | Impact | Risk ranking |
| 1 | Users data leaks | Medium | High | High |
| 2 | Slow system responses | Medium | Low | Low |
| 3 | System unavailable | Low | Extreme | High |
| 4 | System too complicated and difficult to use | Low | Low | Low |
| 5 | Phishing | Low | High | Medium |
| 6 | User impersonation | Medium | Extreme | High |
| 7 | System untrusted | Medium | Extreme | High |
| 8 | Black market | Low | Extreme | High |

Table 3: Business Risks

## 2.4. Two misuse cases examples

In the figure 1, the grey circles represent the identified misuse cases, the white circles represent legit interactions with the application.

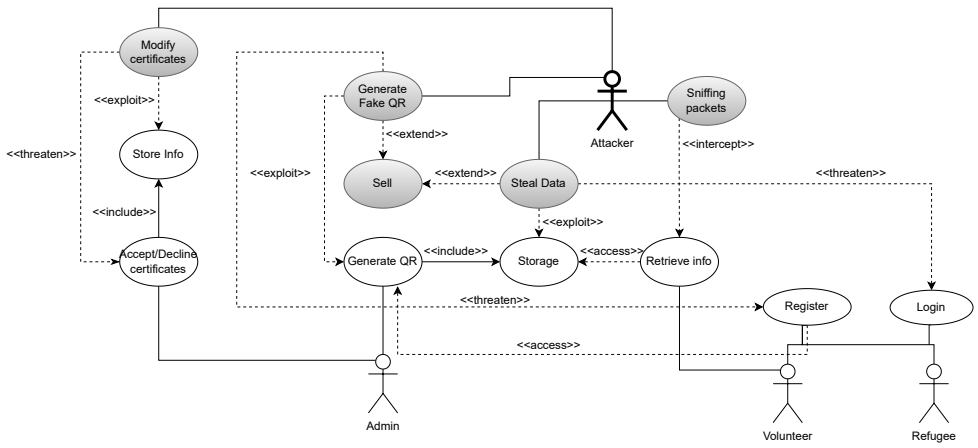


Figure 1: Misuse cases.

## 2.5. Two attack tree examples

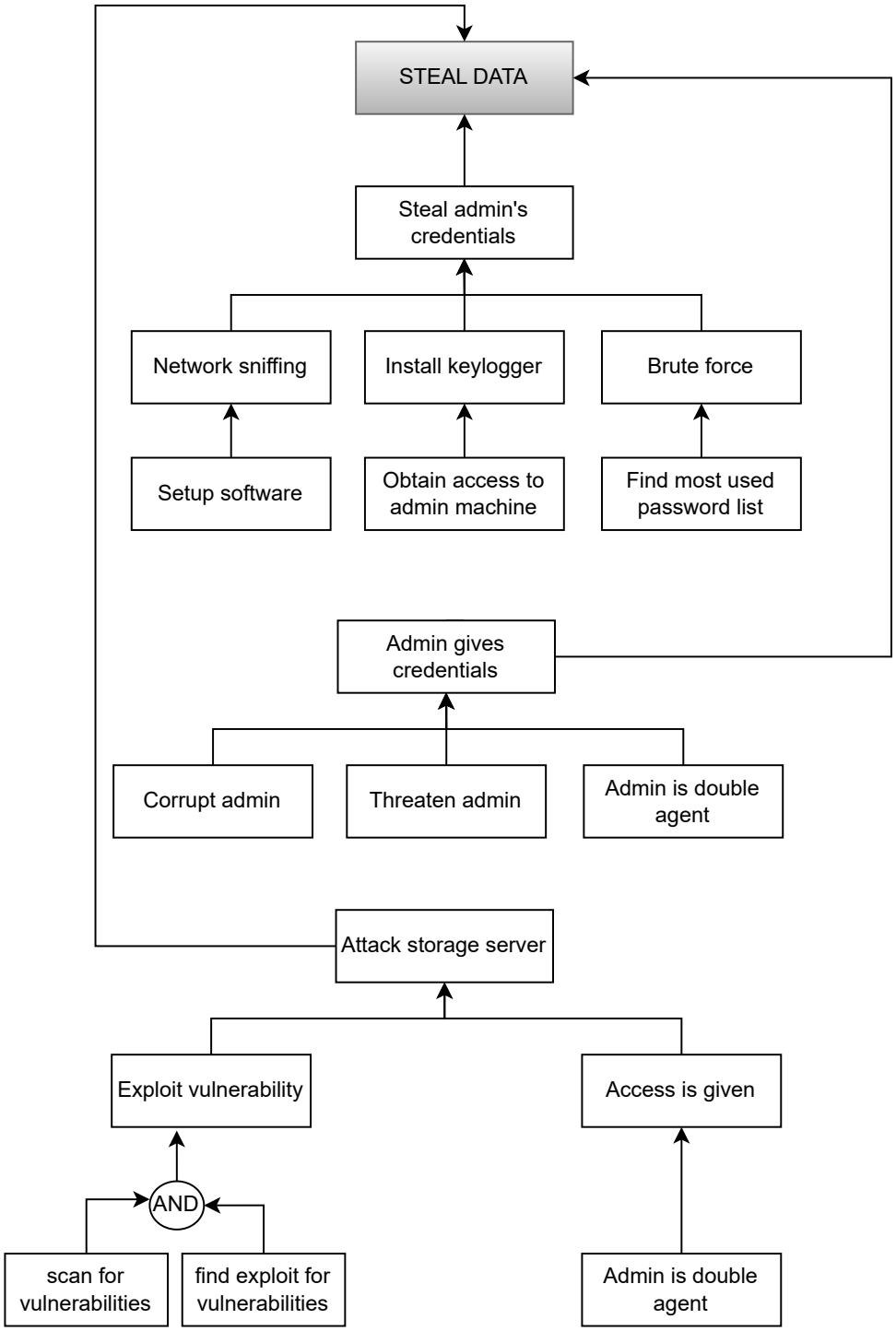


Figure 2: Attack tree 1.

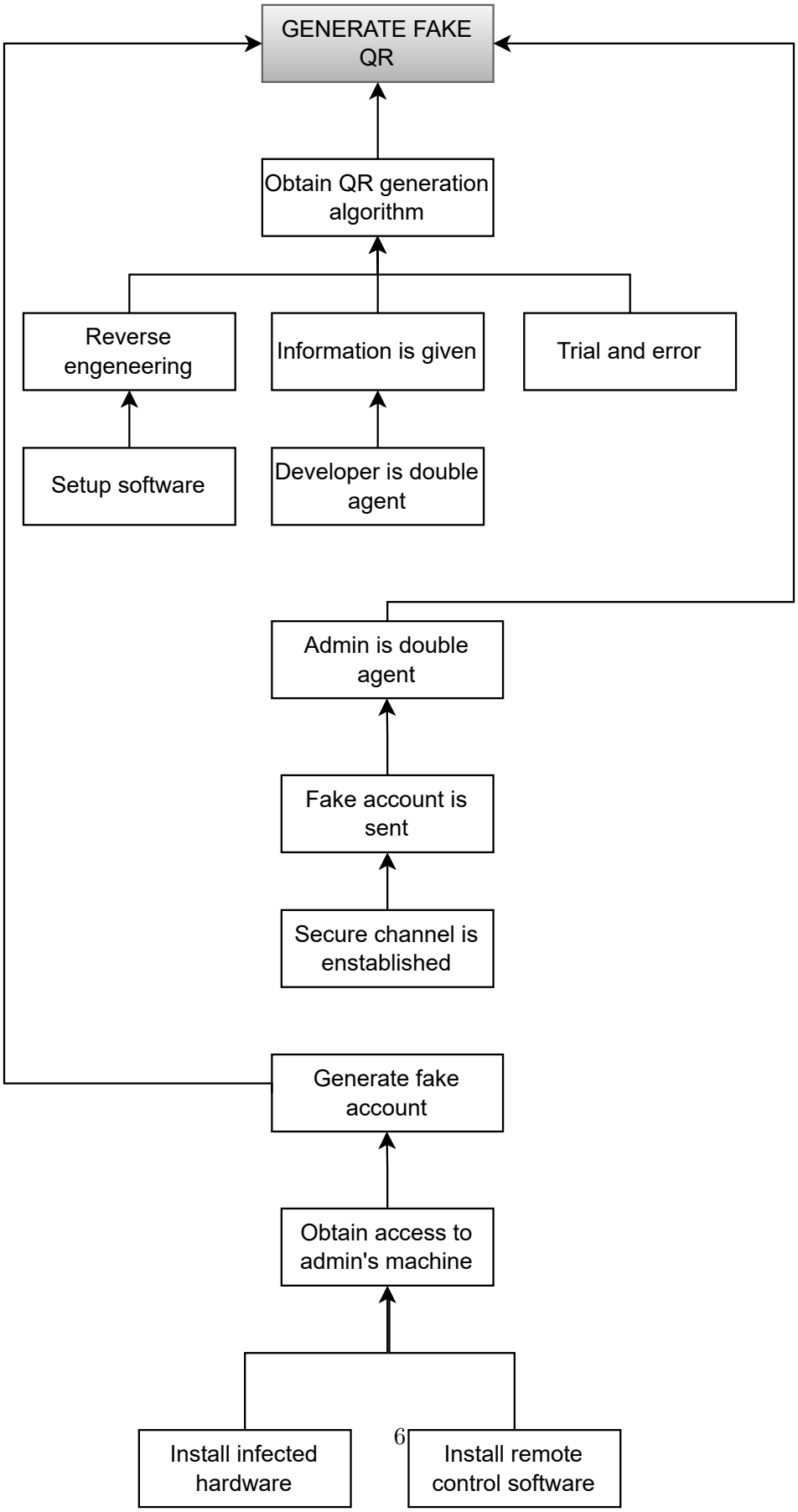




Figure 3: Attack tree 2.

## 2.6. Identified Technical Risks

When an app needs to be deployed in such a small period of time, a technical risk analysis should be a priority. In this section, we will explore some of the most critical technical risks associated with the application, how likely they are to happen, what their impact will be, what security measures could potentially lower the chances of the risks to happen and the business risk they are related to. The following table 4 is a summary of the above information displaying the technical risks ordered by likelihood.

One of the most prevalent security threats [1] that can potentially result in significant damage to any application is a dictionary attack. The most commonly used passwords [2] [3] [4] can be easily found via a simple internet search (e.g. rockyou.txt). To mitigate this risk, the application should be designed to reject common passwords and lock out users doing suspicious actions such as trying brute force attacks.

This situation presents other significant challenges, one of which is the risk of unwanted users lockout. It is not fair to penalize genuine users for an attack they did not initiate. Two-factor authentication (2FA) is a viable solution that could mitigate the risk of locking out legitimate users [5]. Another potential issue is Distributed Denial of Service (DDoS) attacks, which can cause severe damage to an application. To prevent DDoS attacks, numerous countermeasures need to be implemented. First and foremost, implementing firewalls and intrusion detection/prevention systems to identify and block DDoS traffic before it reaches the application is a priority. However, given that attacks cannot always be prevented, therefore developing a backup recovery plan in the event of a successful DDoS attack should also be considered [6].

Servers are not threatened only by DDoS attacks. To prevent server crashes caused by invalid input, (e.g. validate that the input in a numerical field is a numerical value) measures such as input validation must be implemented. Monitoring the system using log files [7] and anti-malware software can be beneficial to protect the machine [8] [9]. Additionally, implementing a backup server and backing up data can mitigate the impact of hardware failures or data loss due to malware attacks or external events [10].

Injections are also a matter that needs to be addressed. To avoid SQL injections, every user input needs to be escaped/sanitized and queries should be parameterized instead of building them dynamically using user's input [11]. A similar strategy applies also for mitigating XSS attacks: Filtering user input, encoding output data, using appropriate response headers, and implementing a Content Security Policy (CSP) [12].

Finally, to prevent session token thefts, CSRF attacks and data sniffing, developers must ensure that all sensitive information is sufficiently encrypted and that recommended protocols such as TLS are utilized [13]. Furthermore, the usage of a VPN can provide a secure and encrypted connection between the user's device and the server, thereby reducing the risk of token theft [14]. In conclusion, a sound understanding of basic cybersecurity principles can enhance security measures and mitigate the identified security threats.

| Technical Risks | | | | | |
|---|---|---|---|---|---|
| ID | Description | Likelihood | Impact | Security Requirements | Related Business Risk |
| TR6 | Dictionary Lockout | High | Genuine user is locked out | 2FA Server logs | 3 |
| TR13 | Dictionary Attack | High | Get access to user's account | Disallow common passwords | 3 |
| TR2 | SQL injection | High | User's info could be stolen | Sanitize input | 1 |
| TR3 | XSS attack | High | Sensitive information theft | Input Validation | 6 |
| TR12 | Web server crashes on invalid input | High | Process delay | Input validation | 2 |
| TR10 | Network sniffing | High | Data could be stolen | TLS VPN | 1 |
| TR1 | DoS Attack | Medium | System unavailability | Implement anti DoS technologies | 3 |
| TR11 | Session Token Theft | Medium | Data theft of Refugees | HTTPS | 1 |
| TR5 | Slowloris attack | Medium | Resource drain | Timeout Server logs | 2 |
| TR7 | CSRF attack | Medium | Users could be scammed | Educate users | 5 |
| TR4 | Poor user verification | Low | Attacker is volunteer | Camera verification | 7 |
| TR8 | Hardware failure | Low | No service | Backup server | 3 |
| TR9 | Malware on server | Low | Severe security breaches | Log activities Maintenance | 7 |

Table 4: Technical Risks

## 2.7. Test plan

In the following section we aim to provide a comprehensive description of the test plans required to mitigate the aforesaid technical risks. The tests are ranked in a scale from one to three based on their priority to be tested. One corresponds to higher priority.

When creating a test plan, it is important to prioritize the tests based on the potential risks and impact of a failure.

In this case, the highest priority should be given to DDoS, SQL injections, and XSS attacks, which are common and can cause significant damage to the system. Dictionary lockout, Dictionary Attacks and CSRF attacks should also be high priority as they can lead to unauthorized access and data theft.

Hardware failure and malware on the server should be prioritized next, as they can cause downtime and loss of data.

Network sniffing and session token theft should be given low priority, as they can be mitigated with encryption and secure protocols. However, they should still be thoroughly tested to ensure the system's overall security.

The table below 5 contains a comprehensive description of a test plan for most of the identified technical risks. The order of the table is corresponding to the testing priority of each test.

| Test Plan | | | |
|---|---|---|---|
| Related Technical Risk | ID | Test Priority (1-3) | Test Description |
| DoS Attack | TR1 | 1 | Attempt to overload the app by sending a large number of requests (e.g. log in attempts) in a short amount of time, and monitor the app to identify any signs of slowdowns or failure. |
| SQL Injection | TR2 | 1 | Check if unsanitized input (e.g. 'OR 1==1') gives unathorized access. |
| XSS Attack | TR3 | 1 | Test if <script>alert('XSS');</script> is allowed in input fields in the app. |
| Dictionary Attack | TR13 | 1 | Do not allow common passwords (e.g. qwerty) and hash passwords with salt and pepper for each user. |
| Dictionary Lockout | TR6 | 1 | Attempt to log in to the application using a list of common passwords and usernames. After a certain number of failed attempts, verify that the account or IP address is locked out. Attempt to log in again with the correct credentials and verify that the account is unlocked |
| CSRF Attack | TR7 | 1 | Test [15] if you can access refugee information from a basic html page with the following script while you are logged in the site in a different tab <img src="http://site.com/scanned>. |
| Hardware failure | TR8 | 2 | Test backup and recovery procedures to ensure that they are effective in restoring system functionality after a hardware failure. |
| Malware on Server | TR9 | 2 | Install anti-malware software on the server side. |
| Network Sniffing | TR10 | 3 | Use Wireshark to capture and analyze if sensitive data can be found as a plain text. |
| Session Token Theft | TR11 | 3 | Use Burp Suite to capture session tokens in the requests, and then with the repeater tool try to make a request with the stolen token. |

Table 5: Test plan

## 3. Summary of Findings

In conclusion, the application contains vulnerabilities that need to be taken into consideration in order to protect the integrity of their user's information. An analysis has been conducted in order to find the right counter-measures and prevent worst scenarios.

The interest of the application corresponds to the business assets that were identified, with the most important of them being the refugees' information, therefore the analysis is primarily centered on protecting the confidentiality and integrity of this sensitive data, as well as preserving the organization's reputation.

We have identified several significant technical risks that have the potential to harm the organization and the users. Data of the refugees is the most targeted asset among all the threats.

The majority of these technical risks can be mitigated by implementing the test plan outlined above, thereby improving the safety of the application.

## References


[1] "10 most common types of cyber attacks." https://www.crowdstrike.com/cybersecurity-101/cyberattacks/most-common-types-of-cyberattacks/.

[2] "Top 200 most common passwords." https://nordpass.com/most-common-passwords-list/.

[3] "Wikipedia:10,000 most common passwords." https://en.wikipedia.org/wiki/Wikipedia:10,000_most_common_passwords.

[4] "20 most hacked passwords in 2023: Is yours here?." https://www.safetydetectives.com/blog/the-most-hacked-passwords-in-the-world/.

[5] "What is two-factor authentication and why is it used?." https://www.techtarget.com/searchsecurity/definition/two-factor-authentication.

[6] "What is a ddos attack?." https://aws.amazon.com/shield/ddos-attack-protection/.

[7] "The importance of website server logs." https://blog.sucuri.net/2022/08/importance-of-website-logs-for-security.html.

[8] "Protect servers from malware in four steps." https://help.deepsecurity.trendmicro.com/10/0/configure-anti-malware.html.

[9] "5 ways to protect your servers from ransomware attack." https://stonefly.com/blog/5-ways-to-protect-your-servers-from-ransomware-attacks.

[10] "How to prevent data loss from hardware failure." https://invenioit.com/continuity/prevent-data-loss-from-hardware-failure/.

[11] "Protecting against sql injection." https://www.hacksplaining.com/prevention/sql-injection.

[12] "Cross-site scripting." https://portswigger.net/web-security/cross-site-scripting.

[13] "What is packet sniffing? what are the ways to protect against sniffing?." https://in.norton.com/blog/privacy/what-is-packet-sniffing-and-ways-to-protect-against-sniffing.

[14] "What is session hijacking, and how can it be prevented?." https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/how-to-prevent-session-hijacking-attacks/.

[15] "How to test for cross-site request forgery?." https://brightsec.com/blog/cross-site-request-forgery-testing/.