Report - Software Security Requirements

UNIVERSIDADE D COIMBRA

Francisco Catarino Mendes - 2019222823
Department of Informatics Engineering
University of Coimbra

Introduction

In today's software world, a lot of attention is given to the functionalities of the system, from the user's perspective. On the other hand, not much focus is put on what the system should not do. Studies have shown that requirements engineering defects are always very costly financially to entities, going from 25 to 40 percent of the project budget [1]. Moreover, the return on investment when security analysis and secure engineering practices are introduced early in the development cycle ranges from 12 to 21 percent [2]. Organizations often still neglect quality requirements such as performance, safety, security, reliability, and maintainability, due to the desire to keep costs down and meet aggressive schedules, only prioritizing the functional end-user requirements [3]. Besides, when talking about security requirements, making functional requirements without taking into account security measures and the possible attacker's point of view can lead to considerable consequences.

This report has the intention of studying a home banking application with the objective of identifying, analyzing, and validating security requirements for the organization's software system. This process is going to be done following steps 2,3,4,6, and 7 of the **SQUARE Methodology**, a comprehensive procedure for security requirements engineering. The system provides the following functionalities:

1. Change Password, User ID, Security Phrase, and Challenge Questions and/or Answers for the Online Banking service;
2. Retrieve User ID;
3. Check account balances, review activity, and access account history;
4. Transfer funds between accounts and to other customers' accounts.

Regarding the architectural solution, a database-centric architecture is assumed, although it is considered that some operations may require access to external systems.

Step 2 - Identify Business and Security Goals

In this section, a business goal is defined, in a phrase, with the intention of giving the best possible characterization of what is intended with this application, followed by the security goals considered to be in more or less clear support of this business goal.

Usually, these pieces of information are asked and then received from the stakeholders, who use the help of the requirements engineering team, which means that this part of the report heavily depends on their feedback [3]. Table 1 shows both the business goal and the security goals obtained:

As one can see, the security goals are heavily focused on the CIA triad, the three most critical components

Business Goal
The system allows the client to check his financial data and status, as well as to make financial operations between his and other accounts.
Security Goals
G-01 All the users should be authenticated to be able to access the application, consult information, and make operations (authentication). G-02 All of the client's financial data and information exchanged with the system should only be accessible to authorized users (confidentiality). G-03 All types of data of the user should be real, accurate, and safeguarded from unauthorized modification (integrity). G-04 The system should be available whenever the clients need to access financial data or perform financial operations (availability).

Table 1: Business and Security Goals

of information security, and on the topic of financial data, as it is the type of data that this application deals with.

Step 3 - Develop Artifacts

Step 3 of the SQUARE Methodology is concerned with developing artifacts to serve as important inputs for the following steps. The artifacts that were focused on in this report were misuse cases and attack trees. With the help of knowledgeable engineers of the organization and the requirements engineering team, the stakeholders can then define these artifacts [3].

Attack Trees

Attack trees provide a formal, hierarchical way of describing the security threats to a system based on the types of attacks that could happen and how they could be realized. Attack tree diagrams represent attacks in a tree structure, where an attacker's goal is listed as the root node and tree leaves represent different ways to achieve that goal [3]. For our study, three attack trees are defined after considering which were the most impactful scenarios for the system. Figures 1, 2, and 3 show the results of this procedure:

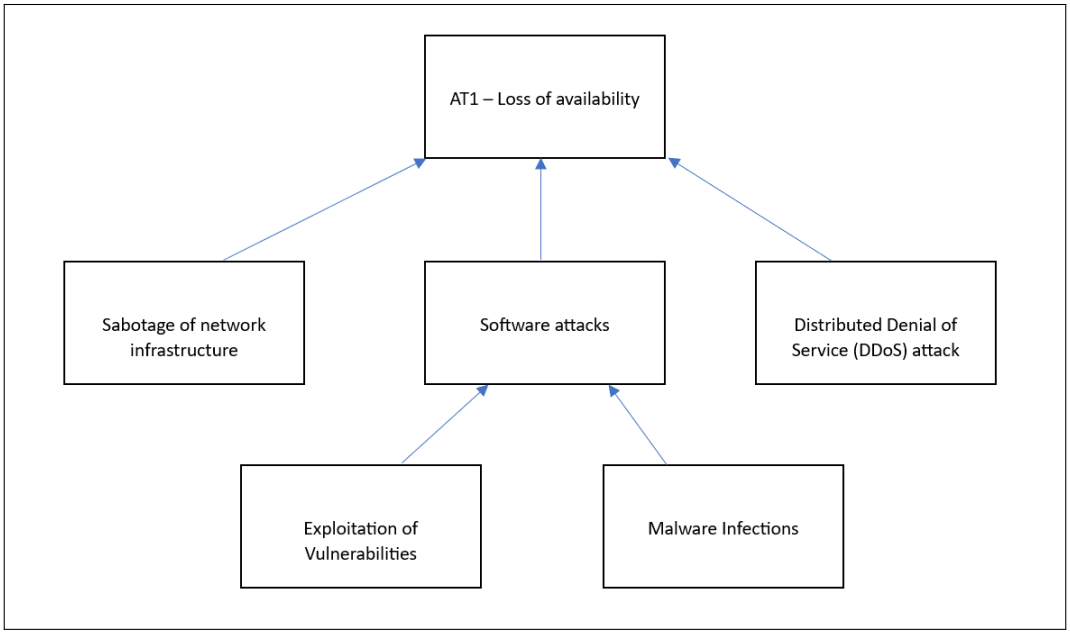

Figure 1: AT-01

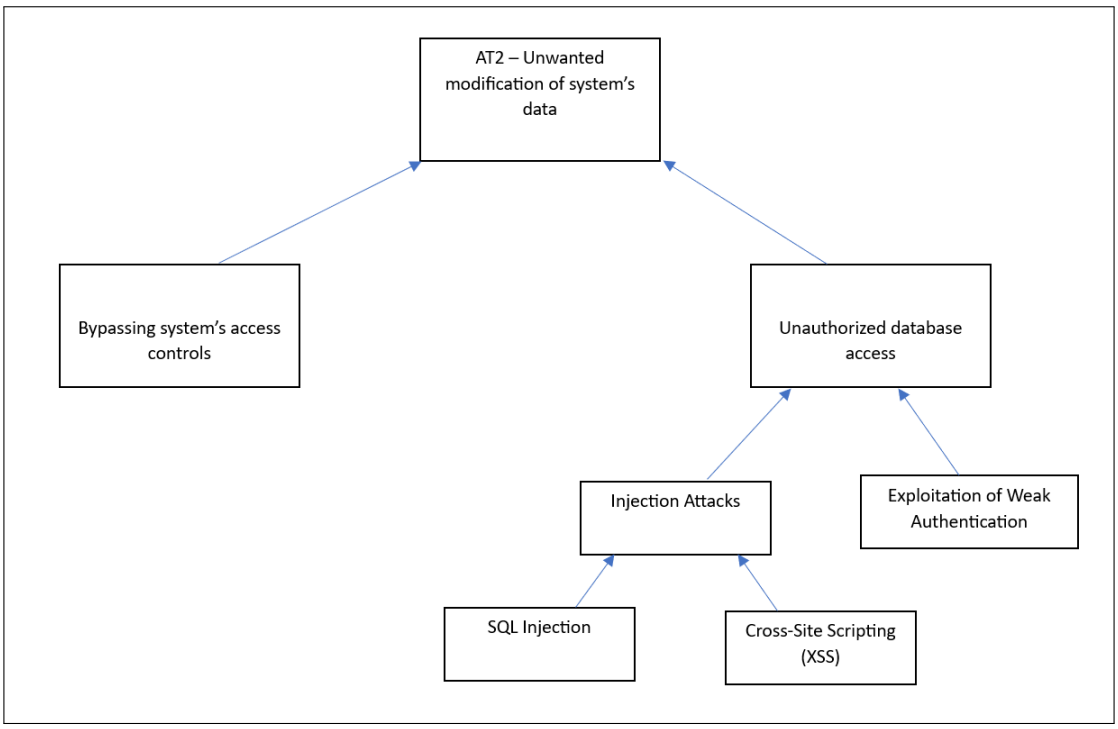

Figure 2: AT-02

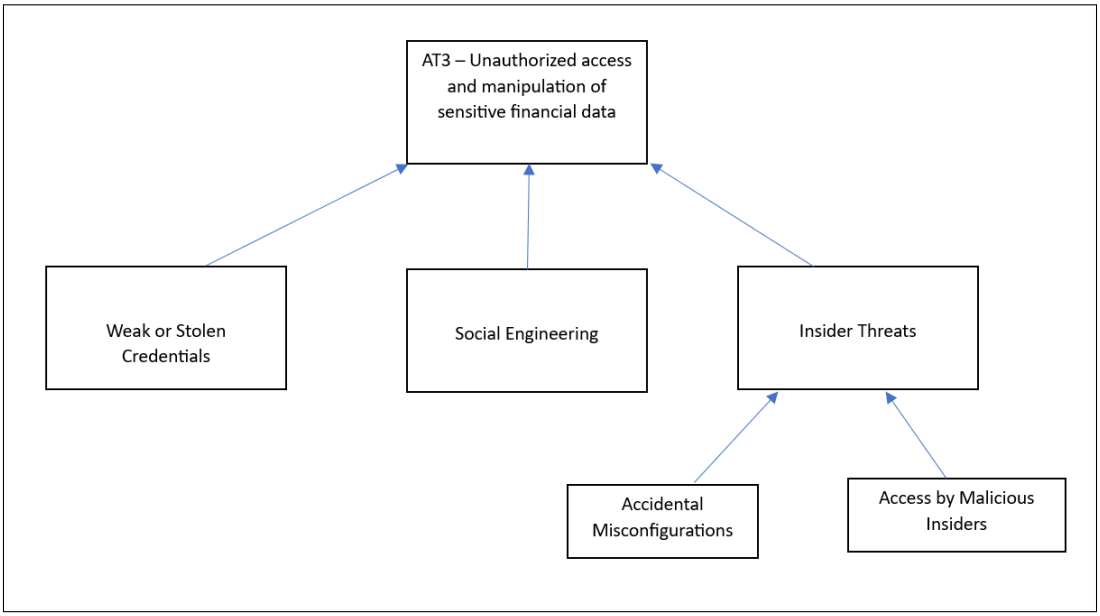

Figure 3: AT-03

As one can see, Figure 1 represents the various ways a loss of availability can happen, Figure 2 portrays how an attack on the integrity of the system can happen, and Figure 3 depicts some courses of action an attacker can take in order to gain unauthorized access to sensitive information.

In the end, these artifacts help in getting a better sense of the scope of threats that the system faces, as well as understanding the point of view of the attacker.

Misuse Cases

In order to have a detailed characterization of the most threatening situations the application can face, misuse cases are established, so that a better interpretation of the circumstances can be done. The misuse cases include the scope, which gives information about the area of the system we are in, the priority, so that an order of assessment can be made, the type of deployment environment, the security categories involved, pre-conditions that must be in place for the attack to take place, and finally post-conditions, that include having prepared a worst-case threat possibility mentality, and the most important prevention, detection, and recovery mechanism of the system.

Three misuse cases are set, that end up meeting the content of the attack trees earlier designated. Figures 4, 5, and 6 show the results of this process, while Table 2 shows the correlation between the misuse cases and the attack trees:

Number:	MC-01	
Name:	Unauthorized access and manipulation of sensitive financial data	
Scope:	Confidentiality and Authorization Concerns	
Priority:	( ) Low ( ) Medium (**X**) High	
Deployment Environment:	( ) Intranet (**X**) Extranet/Internet	
Security Attributes Affected:	(**X**) Confidentiality (**X**) Integrity ( ) Availability	
Description:	An attacker gains unauthorized access to an account, violating data privacy and potentially manipulating transactions.	
Pre-conditions:	The attacker has access to the network, either from the internet or within the organization's extranet. The attacker has discovered either a vulnerability in the system or the credentials of a user.	
Post-conditions:	Worst Case Threat:	The attacker successfully accesses sensitive data and executes unwanted financial operations.
	Wanted Prevention Guarantee:	Have strong authentication and access control measures.
	Wanted Detection Guarantee:	Intrusion detection and prevention systems.
	Wanted Recovery Guarantee:	Incident response procedures detailing steps to contain, investigate, and recover from incidents.

Figure 4: MC-01

Number:	MC-02	
Name:	Corruption of the database of the system, resulting in unauthorized modifications	
Scope:	System's Integrity Concerns	
Priority:	( ) Low ( ) Medium (**X**) High	
Deployment Environment:	(**X**) Intranet ( ) Extranet/Internet	
Security Attributes Affected:	( ) Confidentiality (**X**) Integrity ( ) Availability	
Description:	An attacker gains unauthorized access to the system's database and deliberately corrupts data.	
Pre-conditions:	The attacker has some level of access to the system, probably through a vulnerability. The attacker has the necessary knowledge or tools to manipulate the database.	
Post-conditions:	Worst Case Threat:	The attacker successfully corrupts the database, leading to significant data loss.
	Wanted Prevention Guarantee:	Always sanitize different types of queries and perform input validation.
	Wanted Detection Guarantee:	Real-time monitoring of the system's activity including log analysis.
	Wanted Recovery Guarantee:	Have regular backups of the database and a well-documented disaster recovery plan.

Figure 5: MC-02

Number:	MC-03	
Name:	Compromising the server that provides the availability of the application	
Scope:	System's Availability Concerns	
Priority:	( ) Low ( ) Medium (**X**) High	
Deployment Environment:	( ) Intranet (**X**) Extranet/Internet	
Security Attributes Affected:	( ) Confidentiality ( ) Integrity (**X**) Availability	
Description:	An attacker targets the server that maintains the application, disrupting its availability.	
Pre-conditions:	The server is exposed to the extranet, making it accessible from external networks. The attacker has knowledge of the server's existence and its potential vulnerabilities.	
Post-conditions:	Worst Case Threat:	The attacker successfully makes the server offline for an undetermined period of time.
	Wanted Prevention Guarantee:	Have strong network security measures like network segmentation and isolation techniques.
	Wanted Detection Guarantee:	Have continuous monitoring of the server's activities and network traffic along with logs to detect unusual or suspicious behaviour.
	Wanted Recovery Guarantee:	Possess a disaster recovery plan that outlines steps for recovering the server's functionality in case of a compromise.

Figure 6: MC-03

Misuse Case Name	Attack Tree
Unauthorized access and manipulation of sensitive financial data	AT-03
Corruption of the database of the system, resulting in unauthorized modifications	AT-02
Compromising the server that provides the availability of the application	AT-01

Table 2: Mapping Between Misuse Cases and Attack Trees

While the attack trees provide a general picture of potential attacks on the system, the misuse cases drill down to the details of the interactions between system components in the event of an attack [3].

Step 4 - Perform Risk Assessment

Concerning step 4 of the SQUARE Methodology, a risk assessment work of the system is required, so that concrete threats and vulnerabilities are identified, which serve as a baseline for the determination of the loss frequency and loss impact of an attack, and together with a value of uncertainty merge to form the risk indicator. This operation is done by an external risk expert, that receives the context of the system or organization that needs assessment by the requirements engineering team.

After some consideration and investigation, the framework chosen for the work was the NIST SP-800-30 - Guide For Conducting Risk Assessments [4]. Using a spreadsheet that will be linked at the end of this section, the outcome is described in the subsections that follow.

Threat Identification

(a) T1 - Malware (ransomware, viruses, worms);

(b) T2 - Injection Attacks (SQL injection, OS injection, XSS attacks);

(c) T3 - Social Engineering (phishing attacks, man-in-the-middle attacks);

(d) T4 - Insider Threats (malicious insiders access and operate sensitive financial data);

(e) T5 - Weak Authentication (inadequate authentication measures like allowing weak passwords);

Vulnerability Identification

(a) T1: Outdated and weak anti-malware software (V1); Unpatched Operating Systems (V2);

(b) T2: Failure to sanitize user inputs in SQL queries (V3); Lack of input validation (V4);

(c) T3: Lack of security awareness (V5); Easily accessible individual information (V6);

(d) T4: Weak access controls (V7); No user activity monitoring (V8);

(e) T5: Inadequate password policies (V9); Not using Multi-Factor Authentication (V10);

Calculating Risk

From the content shown in Table 3, vulnerability 2 - Unpatched Operating Systems, ranks the highest, meaning it probably has the biggest impact on the system among all of the vulnerabilities identified, and consequently should have the highest priority in terms of having prevention, detection, and recovery controls.

Asset	Asset Relative value	Vulnerability	Loss Frequency			Loss Magnitude (Impact)			Uncertainty	Risk
			Likelihood	Attack Success Prob.	Loss Freq	Asset Value	Probable Loss	Loss Magnitude		
A1	100	V1	5%	50%	3%	100	50%	50	20%	1.5
A1	100	V2	40%	70%	28%	100	70%	70	40%	27.44
A1	100	V3	30%	80%	24%	100	30%	30	20%	8.64
A1	100	V4	30%	80%	24%	100	30%	30	20%	8.64
A1	100	V5	10%	80%	8%	100	30%	30	50%	3.6
A1	100	V6	5%	90%	5%	100	30%	30	50%	2.025
A1	100	V7	20%	80%	16%	100	50%	50	70%	13.6
A1	100	V8	20%	80%	16%	100	50%	50	70%	13.6
A1	100	V9	20%	80%	16%	100	30%	30	20%	5.76
A1	100	V10	40%	80%	32%	100	30%	30	20%	11.52

Table 3: Vulnerability Risk

	NIST SP-800-30
Tier 1	V2 – Unpatched Operating Systems V7 – Weak access controls V8 – No activity monitoring
Tier 2	V3 – Failure to sanitize user inputs in SQL queries V4 – Lack of input validation V10 – Not using Multi-Factor Authentication
Tier 3	V1 – Outdated and weak anti-malware software V5 – Lack of security awareness V6 – Easily accessible individual information V9 – Inadequate password policies

Table 4: Vulnerability Risk Ranking

As for Table 4, it represents the ranking of the vulnerabilities in tiers, from the ones that have less risk (tier 3) to those that have the greater (tier 1). Insider threats (T4) rank highest along with vulnerability 2, and on the other hand, social engineering was put on the lowest tier due to the fact that awareness in tech organizations tends to be good.

Link for Risk Assessment Spreadsheet:

`https://docs.google.com/spreadsheets/d/1I-d-56gxWe386mTgB52aGHoS3rlq6FXv/edit?usp=sharing&ouid=105633069300502448317&rtpof=true&sd=true`

Step 6 - Elicit Security Requirements

Step 6 can be considered as the heart of the SQUARE Methodology. Here, the requirements engineering team, with the help and input of the stakeholders, comes up with the security requirements of the system. It is very important to mention that security requirements must not be non-verifiable, vague, or ambiguous, and should not be mistaken with implementations or architectural constraints. In case any of this happens along the process, this step must be retaken in order to correctly establish the security requirements.

Table 5 shows the security requirements made using the "Unstructured Interview" Elicitation Technique, where, in essence, the team does not use a specific elicitation technique to develop the requirements with the stakeholders.

R-01	The system is required to have strong and unique user passwords, with a minimum length of 12 characters, including a mix of upper and lower case letters, numbers, and special characters, as well as multi-factor authentication.
R-02	The system is required to have role-based access control, allowing administrators to define user roles and permissions, and users to have access only to the data and functionality relevant to their roles.
R-03	The system requires that sensitive data, both in transit and at rest, must be encrypted using industry-standard encryption protocols.
R-04	The system is required to maintain detailed audit logs of all user activities and system events, including successful and failed login attempts, data modifications, and critical system actions.
R-05	The system requires that data modification and deletion must be logged and require proper authorization, with a clear audit trail for accountability, and that data validation and integrity checks must be in place to prevent unauthorized data tampering.
R-06	The system is required to provide informative but non-revealing error messages, especially during login and authentication processes, to prevent information disclosure to potential attackers.
R-07	The system is required to have regular security awareness training for all employees, covering various topics like social engineering, password security, safe online behaviour, third-party components, and programming good practices, among others.
R-08	The system is required to perform and securely store regular data backups, with a documented disaster recovery plan in place for data restoration in the event of data loss or system failure.

Table 5: Security Requirements using Unstructured Interview

There are specific elicitation techniques, like IBIS, ARM, and JAD. All are valid, and probably better options to do this procedure in a more efficient way. However, all of them involve brainstorming sessions between the requirements engineering team and the stakeholders, in order to generate questions that, eventually, the final security requirements can answer. Therefore, the report will continue with the security requirements designated with the "Unstructured Interview" Elicitation Technique.

Step 7 - Categorize Requirements

The final step of the SQUARE Methodology that is covered by this report is step 7. In this section, the aim is to categorize the security requirements obtained in step 6, to better understand their scope in the system, and to make sure that they really are security requirements, because it can be easy to mistake them for architectural constraints. Once again, the stakeholders are responsible for this process, with the always-present help of the requirements engineering team.

Using a grouped categorization technique, that begins by fixing the possible categories, it then assigns each security requirement to the most appropriate and relevant category. The results are illustrated in Figure 7:

Group A: Confidentiality 1 - The system requires that sensitive data, both in transit and at rest, must be encrypted using industry-standard encryption protocols.	**Group B: Access Control** 1 – The system is required to have role-based access control, allowing administrators to define user roles and permissions, and users to have access only to the data and functionality relevant to their roles.
Group C: Data Integrity 1 - The system requires that data modification and deletion must be logged and require proper authorization, with a clear audit trail for accountability, and that data validation and integrity checks must be in place to prevent unauthorized data tampering. 2 - The system is required to perform and securely store regular data backups, with a documented disaster recovery plan in place for data restoration in the event of data loss or system failure.	**Group D: Manageability** 1 - The system is required to maintain detailed audit logs of all user activities and system events, including successful and failed login attempts, data modifications, and critical system actions. 2 - The system is required to have regular security awareness training for all employees, covering various topics like social engineering, password security, safe online behaviour, third-party components, and programming good practices, among others.
Group E: Usability 1 - The system is required to provide informative but non-revealing error messages, especially during login and authentication processes, to prevent information disclosure to potential attackers.	**Group F: Authentication** 1 - The system is required to have strong and unique user passwords, with a minimum length of 12 characters, including a mix of upper and lower case letters, numbers, and special characters, as well as multi-factor authentication.

Figure 7: Grouped Categorization of Security Requirements

All security requirements fit well in one of the groups. In the end, there is no need to add or delete any security requirements.

Conclusion

All in all, this document represents one possible way of executing the SQUARE Methodology. It shall be used whenever the organization needs it, in specific regard to developing security requirements for the home-banking application. A constant reassessment of this report should be made, to re-evaluate the state of the application.

References

[1] Wiegers, Karl E. Software Requirements. Redmond, WA: Microsoft Press, 2003.

[2] Berinato, Scott. “Finally, a Real Return on Security Spending.” CIO, April 8, 2002.

[3] N. Mead, E. Hough, T. R. Stehney II, “Security Quality Requirements Engineering (SQUARE) Methodology”, CMU/SEI-2005-TR-009, CMU, 2005.

[4] NIST - Joint Task Force Transformation Initiative, “Guide for Conducting Risk Assessments”, September 2012.