Chapter One: Introduction to Threat Model

Introduction

Software Development Lifecycle comprises of several phases: Requirement Gathering and Analysis, Design, Implementation or Coding, Testing, Deployment and Maintenance. Intuitively it could be understood that eliminating small threats in the earlier stages of software development would prevent encountering a very complex threat at the later stages.

Threat modelling is a structured approach to identify and manage threats which can jeopardize the system. Threat models can be applied to software, networks and even business processes.

Objective

The objective of this document is to identify where the most effort should be applied to keep the system Result Management System, RMS secure. The assets and vulnerabilities of RMS, threats to the system would be identified.

Threat Model and its Purpose

As stated before, threat modelling is a structured approach to identify and nullify or eliminate threats which can jeopardize the system and cause loss of assets. Below we state some more reasons of why threat modelling should be done.

* It is better to find security flaws when there is time to fix them.
* It can save time, revenue and the reputation of your company.
* To build a secure application.
* To bridge the gap between developers and security.
* It provides a document of all the identified threats and rated threats.
* It offers knowledge and awareness of the latest risks and vulnerabilities.

Steps of Building Threat Model

The steps to build a Threat Model are as follows.

* Identifying Assets
* Describe Architecture
* Decompose Application
* Identification of Threats
* Documenting the Threats
* Rate Threats

Conclusion

Chapter Two: Threat Model for RMS

Introduction to RMS

Result Management System or RMS is a system intended to assist the result management system of University of Dhaka. There are three types of users involved in the system: Admin, Teacher and Student. RMS consists of three subsystems. These are: Authentication, Marks Maintenance and Result Calculation and Publication.

Admin and Faculty Member are the registered users of the system.

Security Requirement of RMS

The security requirements are stated as follow.

* Confidentiality
* Integrity
* Availability
* Access Control
* Authentication
* Non-repudiation

Confidentiality

Data written by one faculty member should be hidden from another faculty member who is not authorized the particular asset.

Integrity

Data written by one faculty member should not be overwritten by another faculty member.

Availability

The system needs to be fully functioning and services should be available upon request.

Access Control

Account and information managed by another user should not be accessible by an unauthorized person.

Authentication

Users should be able to access their account upon entering correct credentials. In no way, an unauthorized person should be allowed to enter another individual’s account.

Non-repudiation

User should be protected from denial by another party in a communication. For example admin assigns faculty member to a course. A faculty member should not be able to deny that he/she was assigned.

Identifying Assets for Threat Model

The first step of threat Modelling is to identify the assets. The assets of RMS is listed below.

* database server
* file servers
* data lake stores
* Active Directory
* REST calls
* configuration screens
* Azure portal
* authenticated and anonymous web user
* Azure AAD client apps
* database users
* DB administrators

Description of Architecture

* Framework: .NET MVC 5
* Tool: Visual Studio 2016 Community Edition
* Architecture layers: There are three layers:
  + Presentation layer
  + Business layer
  + Database layer**Presentation layer**

Presentation layer or client layer is the top most layer of an application. The main functionality of this layer is to communicate with Application layer. For example, login page of RMS where an end user could see input text boxes and buttons to enter username, password and to click on sign-in. This is also called view of the application.

**Business layer**

Business logic layer or logical layer is where data from client layer and database layer is processed. In RMS, faculty members enter marks of students into an auto generated excel sheet. This input from presentation layer is stored in database. When result sheet is being computed, data is read from database and internal computations are done. These computations are the internal business logic of RMS. Afterwards, the computed result is sent to database layer. It can be stated that this layer acts as a mediator between the Presentation and the Database layer.

**Database layer**

All data inserted into the presentation layer and processed by business layer is stored in the database layer. Business layer communicates with Database layer to retrieve the data. It contains methods that connects the database and performs required action e.g.: insert, update, delete etc.

Decomposing the Application

Data Flow Diagram is used to breakdown the application regarding its process, including all the sub-processes that are running the application.  We show DFD level 0 and level 1 in this subsection.

DFD Level 0 gives a view of the users that interact with system which is represented as a single process.

In Level 1 DFD, the system is broken down into 3 processes and the individual user interaction of user with those processes is shown. The data that flows relevant to the interaction is also displayed.

Threat Identification

The STRIDE mnemonic is used to identify the threats in RMS. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

**Spoofing**: Spoofing is the act of pretending to be someone an individual is not.

**Tampering**: Tampering is modifying something you’re not supposed to modify. It can include packets on the wire (or wireless), bits on disk, or the bits in memory.

**Repudiation**: Repudiation means claiming you didn’t do something (regardless of whether you did or not).

**Denial of Service**: Denial of Service are attacks designed to prevent a system from providing service, including by crashing it, making it unusably slow, or filling all its storage.

**Information Disclosure**: Information Disclosure is about exposing information to people who are not authorized to see it.

**Elevation of Privilege**: Elevation of Privilege is when a program or user is technically able to do things that they’re not supposed to do.

**SPOOFING THREATS**

|  |  |  |
| --- | --- | --- |
| **THREAT TARGET** | **MITIGATION STRATEGY** | **MITIGATION TECHNIQUE** |
| Spooﬁng a person | Identiﬁcation and authentication (usernames and something you know/have/ are) | Identiﬁcation and authentication (usernames and something you know/have/ are) Usernames, real names, or other identiﬁers:   * Passwords * Tokens * Biometrics   Enrollment/maintenance/expiry |
| Spooﬁng a “ﬁle” on disk | Leverage the OS | * Full paths * Checking ACLs * Ensuring that pipes are created properly |
| Cryptographic authenticators | Digital signatures or authenticators |
| Spooﬁng a network address | Cryptographic | * DNSSEC * HTTPS/SSL * IPsec |
| Spooﬁng a program in memory | Leverage the OS | Many modern operating systems have some form of application identiﬁer that the OS will enforce. |

**TAMPERING THREATS**

|  |  |  |
| --- | --- | --- |
| **THREAT TARGET** | **MITIGATION STRATEGY** | **MITIGATION TECHNIQUE** |
| Tampering with a file | Operating System | * ACLS |
| Cryptographic | * Digital Signatures * Keyed MAC |
| Racing to create a file | Using a directory that’s protected from arbitrary user tampering | * ACLs * Using private directory structures   (Randomizing filenames makes it annoying to execute an attack) |
| Tampering with a network packet | Cryptographic | * HTTPS/SSL * IPsec |
| Anti-pattern | * Network isolation |