Rose-Hulman Institute of Technology

Security and Integrity Analysis

Daniel Schepers, Robert Fendricks, Frank Huang

Table of Contents

1. Privacy Analysis

Due to the fact that our service will be used extensively by the Mafia, there is naturally quite a bit of information that could be considered personal and that should be protected. One of the most important tables that we have is the Employee table, which keeps a good deal of information about the "lawful" workers. In this table, the employee ID, their Family, their username, and their password are all personal, and if any of this information were to be leaked it could compromise them in various ways. For example, if an Employee is found to be associated with a certain Family, they may become the target of law enforcement or rival families, which could easily mean their imprisonment or worse. Only the employee and the head of their family should be able to access this data. To help facilitate this privacy, the username and password of the employee will be encrypted to make it more difficult for an unwanted user to access this sensitive data.

Under the Business table, there are two main columns that must be kept personal: the business's revenue and whether the business is legitimate or not. For our purposes, we are more worried about keeping the legitimacy of the business secret. If a business is found to be illegitimate, it could be shut down and subject to searches by the authorities, which could potentially compromise a Family or cripple them financially. Only a member of the family that owns a business should be able to know if the business is legitimate or not.

2. Security Analysis

After analyzing our system, we came up with three major vulnerabilities that could compromise our database. We have listed these vulnerabilities below and have given details on how we plan on avoiding them

1) Account accessing the database has full privileges

Currently the account that accesses the database has full privileges. If a malicious user were to gain access to this account, they would have limitless power over what they could see and what they could do. If this were to happen, all of the heads of all of the families could become exposed, forcing them to either hide or face legal actions. Furthermore, all of the details of employees working of the various families would be leaked. In short, this would be the end of organized crime. Alternatively, a malicious user could implant fake information into the database, or transfer control of one head of a family to another, causing chaos amongst the families.

2) User inputs are not sanitized

As it stands, our front-end will accept any input as valid without sanitizing it first. By sanitizing all user inputs, not only can SQL injects be prevented, but so can XSS attacks. At the moment, no information can be accessed this way, but data can be tampered with. It would be possible for a malicious user to do something like delete all of the information for a business, causing the owner to lose a substantial amount of money, and possible cause a family to lose track of what business they own or are extorting. Another possibility is that a malicious user could input information that shouldn't be in the database, or even alter information that is already there. Many of the same scenarios as described in the previous security concern could apply here and all of the stakeholders would be affected as well.

3) Traffic between front end and back end is unencrypted

Due to the sensitivity of the information on our database, it is important that there are no leaks anywhere. Currently a malicious user could sniff the traffic between the front end website going to the SQL server. Since none of this information is encrypted, they would easily be able to read all of the information and potentially gain access to confidential data. One possible example could be if a malicious user was sniffing the traffic while an employee was logging on. The malicious user would then have their username and password, and then would be able to easily login while appearing legitimate. At this point, he would have access to all of the information that that employee had access to, which could compromise that employee. To protect against this, we will be looking into possible ways of encrypting the traffic between the front end and the back end.

3. Entity Integrity

a) For the Project table, the ProjectID will be the primary key. Hours_worked will be a positive int, and cannot be null. Name will be a varchar of length 50, and cannot be null.

b) For the product table, ID will be non-zero positive int. It will also be the primary key. Name will be a 40 varchar field, and may not be null. Manf_Cost and Sale_Price must be positive, integers and may not be null.

c) The employee table will have several constraints. The username and password fields will be 40 varchars, and will be encrypted. They may not be null. The date of birth must be in the past. The name of the employee is not allowed to be null, and neither is the address. The primary key will be the Emp_ID, a positive int.

d) For the Prison table, the primary key will be a prisonID, which will be a positive integer. The number of guards will be a positive int. The name of the prison and its warden will be 40 varchar fields, and none of the above mentioned fields may be null

e) For the family table, the primary key will be the family Name. The name is a 50 varchar field. Each family will have a head, who is considered an employee of the family. The Head field may not be null.

f) The business table will have an ID as its primary key. The name will be a 50 varchar field, and revenue will be a positive integer. Neither of these may be null. The legit field is a Boolean, and tells whether a business is legitimate or not. This may not be null.

g) The primary key of Turf is composed of three elements: State, city, and region. State is a 2 varchar field, while city and region are each 40 varchars each. The mayor of a turf may not be null.

4. Referential Integrity Analysis

There are all sorts of references between various tables in our database. If an employee is deleted from the Employee table, we will cascade that change through the database because that employee has probably been killed and is not active in the mafia anymore. If a project is deleted, the projectID field of corresponding entries in the works_on table will be set to null, because that project has been finished or scrapped and there are no employees working on it anymore. If a prison is deleted and there are references to it, the change will be denied. A prison can't simply disappear. A similar tactic will be used for deletion of an entry in the family table – a mafia family can't be dead unless there aren't any employees working for it, it has no businesses, etc. Turf also cannot be deleted. While there may not be a controlling family, a geographic location can't disappear. Because we use IDs for foreign keys, updates will be trivial.

5. Business Rule Integrity Analysis

Because our system is not much of an order tracking system, there are no table quantities that affect other table quantities. Although we keep track of the products that each business sells, we do not keep track of the buyers and therefore we have no orders to track. This may change as time goes on, and we will use stored procedures to implement any business rules we come up with

6. Glossary

Mafia – any group of organized criminals, usually run by a family.

SQL – Standard Query Language

Primary Key – The unique identifier for a sql table.