Delivery Database Security Report

Introduction

Databases in the modern world, are a crucial component to the functioning of each institution. The vast amount of data stored within a database is a commodity that may be targeted by attackers from the outside, attempting to gain unauthorized access.

There are several occasions where databases may be faced with security threats. These threats can be categorized into human, software, hardware, and environment fields, where each, either accidentally or deliberately, poses a threat to the system. It is the responsibility of the database designers and administrator to design a system that is recoverable and with incorporated countermeasures, preventing unauthorized access.

The data in the delivery data base belongs to the Foodie.com startup and contains information relevant to orders, restaurants, and users from a typical month in operation.

While this data may not be highly confidential, a level of security and information structure must be maintained. It is also important to manage the level of access each authorized user has. While an authorized user should be able to view restaurants, food options and their prices, the same user should not be accessing information related to other users, their orders, their payments and addresses.

Preventing Unauthorized Access

Authentication and Monitoring

Authentication is the first step towards securing a database system. It is an initially step which some database security systems incorporate to verify the user's identity. By insuring the user behind the supposed credentials is the authorized individual, the database administrator is able to use their ID to track activity across the platform conducted by each user.

While the delivery database does not require authentication for its users at this stage, incorporating an authentication login process would limit the number of attackers taking advantage of unprotected user accounts and accessing database information via stolen credentials.

Spoofing

Spoofing is a threat that is hard to identify by the databases security system, as the unauthorized user is using compromised legitimate credentials to gain access. A log in or action reference is required to identify this threat as it will record activity while the official user wasn't present.

Multi-Factor Authentication

Based on the level of confidentiality of the data stored in Foodie.com's database, a multi-factor authentication should not be necessary. But could be a effective option for users recording reviews and ratings, where the systems vulnerability is increased, and requesting the users email, for instance, would be a good source for monitoring potential threats when the occasion of a malicious SQL injection would arise.

Authorization

Usually, the information stored in an authenticated user account, specifies the level of access the user has to data.

In Foodie.com's database, once a user is authenticated, access should be determined based on the role/service the user has in the system. Users serving as riders or employees will have access and view information relevant to several orders made from several customers, while customer users will primarily access restaurants, menus and their specific orders.

Roles

Roles can be used to allocate a bundle of privileges to individuals who require them to fulfill tasks defined by the role. A personalized role could be granted to customer, employee, and rider granting access to fulfill their requirement. The administrator and schema owners would hold further privileges for managing the system**.**

Abuse of Privilege

There not much a database designer or administrator can do about users who abuse the privilege granted to them. Abuse of privilege is an instance, where even authentication cannot prevent a malicious attack, since the user has decided to take advantage of the authorized credentials and granted access to compromise the database system

Database designers and administrator, can utilize the principle of least privilege when granting access to users, insuring that only a small portion of users has the power to potentially inflict harm or in other ways compromise the system, narrowing down the potential threat significantly.

In this case, a possible precaution could be periodic check-ins with staff with a higher degree of access to the system, to possibly identify a potential future threat before it unwraps.

SQL Injection

Any application which requests an input from the user may be subject to an attack in the form of SQL injection. SQL injection is a threat where the attacker gains unauthorized access due to their technical understanding of the SQL syntax and structure of the targeted database itself. The attacker may inject malicious SQL statement into a field required by the database, such as a log in, engineering the statement to access the database through the credential validation process, effectively finding a loophole in security system and retrieving unauthorized data or otherwise compromising the system.

It is the job of the database designer and administrator to instill countermeasures against SQL injection, and prevent maliciously engineered SQL statements from being accepted in specific fields.

In the delivery database, the only occasions where user input is requested, is the optional user login integration and user reviews. It is important for the database administrator to introduce the appropriate countermeasures that prevent the system from accepting any submissions in these fields, that do not match pre-determined criteria. These criteria should strictly identify appropriate input and prevent any attempt of merging SQL statement syntax into the submission. It is also good practice to monitor the users potentially posing as a threat via SQL injection and terminate or warn their account immediately.

Row level Security

As discussed previously, different users on the platform will have access to different information. When this degree of granularity is required, we use row-level security which allows us to specify information based on user identification rather than granting access to the entire table. While the master or administrator may view the data as a whole, individual users will be given access to rows relevant to their identification.

The database administrator can incorporate some of these countermeasures to prevent unauthorized access to the system, and the schema owner is responsible for assigning the appropriate level of access to individual users.