DATABASE AUDITING Mini - Project

(GC Media Advertising Company)



|  |  |  |
| --- | --- | --- |
| **PRN** | **Div.** | **Name** |
| 21030241052 | A | Vidit Gupta |
| 21030241102 | B | Suriya Narayanan R |
| 21030241154 | C | Satyabrata Saha |
| 21030241200 | D | Hritik Saxena |
| 21030241240 | D | Mohammad Adib Aslam |

1. The audited entity (*The name of the application*) :   
   GC Media Advertising Company.
2. Database Auditing Plan.

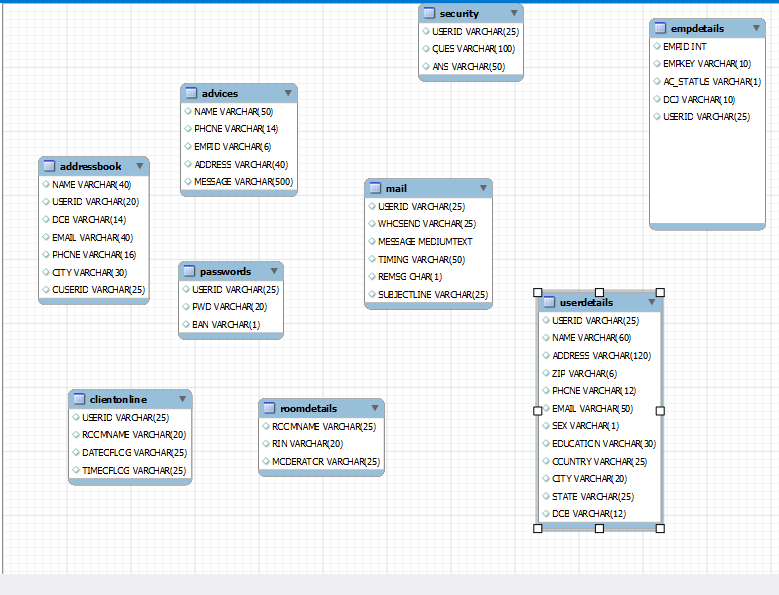
|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Database to be audited** | **Name of the database (for which application)** |
| 1. | Auditing team | Titans |
| 2. | No. of team members | 5 |
| 3. | Database Audit objectives | * To ensure that correct roles and responsibilities are assigned to the users * To ensure confidentiality and integrity of data * To ensure that the application maintains a log whenever the user accesses the database or application * To check the validity of the data stored in the tables |
|  |  |  |
|  |  |  |
| 4. | Start date | 07-10-2021 |
| 5. | End date | 11-10-2021 |
| 6. | Report submission date | 12-10-2021 |
| 7. | Auditing environment to be used |  |
|  | - Programming language used for developing the application | JAVA—JDK version 1.80\_111   1. JSPs(Java Server Pages) and Servlets 2. Core Java   FRAMEWORK:  Struts |
|  | - Type of database files to be audited | MySQL |
| 8. | Signature of team members |  |

1. Database Auditing Check-List.

|  |  |  |
| --- | --- | --- |
| **Sr**. No. | Item | Description(what to audit) |
| 1 | Access Control | * Login as root user and normal user and check if you can insert and select rows in tables. * Check if there are any user roles and privileges associated with the logged in user. * Identify the key differences in the access privileges between user, admin and guest user. |
| 2 | Logs | * Check if any log entry is there in tables .(user ID along with the timestamp should be associated with each and every activity) * Check for any loggers in application for tables in the database .i.e. check the application logs if the query being fired is getting logged or not. |
| 3 | User information | * Check how the user information is stored in tables. * Check the user activity if stored and logged or not. * Check the archival policy and deletion mechanism for user information |
| 4 | Tables | * Check the table creation * Check data types of columns of the table. * Observe the values inserted in the table. * Look for relationship between tables. * Check for encryption * Check the constraints for any column. * Validate any primary keys if present. * Validate any foreign keys |
| 5 | Stored functions | Check if any stored functions are being implemented. |
| 6 | Stored triggers | Check if any triggers or conditions are being implemented. |
| 7 | Stored procedures | Check if any Stored procedures are being executed. |
| 8 | Encryption of data | * Validate if the sensitive and confidential information like password, email, contacts and messages are being encrypted. * Validate if the connection to database is done in an encrypted format. |
| 9 | Links to the database | * Validate how the connection between application and database is getting established. * Look for drivers and check if they are outdated or not. |
| 10 | Sources of database access | * Check what all .jsp files or servlets are responsible for entries in tables of matrimony database. * Validate the flow of jsp and servlet and database and see how the logic is being implemented. |

1. Table Schema and their relationships:

|  |  |  |
| --- | --- | --- |
| Sr.No | Table name (*as in the application given to you for auditing)* | Description (*You could assume and describe what information does this table contain)* |
| 1 | Addressbook | This table is used for finding user information. This table contains the essential details of the Users such as Name, UserID, DOB, Email etc. |
| 2 | Advices | This table lists the employee id, name , phone and the address |
| 3 | Clientonline | This table shows the presence of Client. Date and time of Client logging. |
| 4 | Empdetails | This table is Account used for finding the employee information. i.e. ID, DOJ, account status. |
| 5 | Mail | This table houses the mailing details such as who sent, message line, timing etc. |
| 6 | Passwords | This table stores the password of the users in an unecrypted fashion |
| 7 | Room details | This table gives us the room details such as room number |
| 8 | Security | This table is used if user forgets the password. Security question and answer is stored |
| 9 | User details | This gives the entire particulars of user starting from name, sex education to phone number. |

5. Relationships Identified:

No relationship has been identified among 9 tables.

|  |  |  |
| --- | --- | --- |
| Name of the files to be Audited | | |
| 1 | Adminlogin.jsp | It hosts the login credentials to be fed |
| 2 | Myprofile2.jsp | It contains information about the user profiles |
| 4 | Changepass.jsp | It allows to you to change password |
| 5 | Loginroom.jsp | It displays info about the login |
| 6 | Clientlogin.jsp | It hosts the login credentials to be fed |
| 7 | Register.jsp | It contains the registration details |

1. Databases audited and the Results received:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr**. No. | Item | Description (what to audit) | Recommendations |
| 1 | Access Control | There are primarily two categories of users-  (a)Admin  (B)user  Role based Access Control is used. Only the Admin has the access to remove users and other functionalities | 1) The database has well defined role for different user types  2) Admin can define privileges for different users which is a very nice process. |
| 2 | Logs | 1) Logs for only client is maintained with data and time in client online table  2) No logs for activity of different users is maintained  3) Backup of logs not maintained | 1) Logs for activity of different user types must be maintained to keep a track of any unethical activity  2) Backup of logs must be maintained. If there’s an attack on DB and the attacker deletes the log, we would never be able to get it back. |
| 3 | User information | 1)User information isstored in viewprofile.jsp  2) Sources of information collected are mostly jsp and HTML forms  3) POST method is used to send data to database  4) No backup for user information found | 1) Information can be lost anytime due to any unwanted event so backup of User Information must be maintained. |
| 4 | Tables | 1) Following tables are in the database-   1. Addressbook 2. Advices 3. Clientonline 4. Empdetails 5. Mail 6. Passwords 7. Room details 8. Security 9. User details   2) No Primary key is defined any table.  3) Data types used to take input are VARCHAR, INT, LONG and CHAR  4) Validation of input data can’t be done as triggers are not used.  5) No Anomoly is found in the tables. So need of normalization. | 1) Primary Key must be defined in the table to give uniqueness to a row, prevent duplicates and be referenced in another table  2) Table creation access is only available to admin users which is a good process.  3) Use Stored Triggers |
| 5 | Stored functions | No Stored Functions are found | Stored Function helps to execute multiple queries with one statement. It should be used to decrease resource utilization and reduce time complexity. |
| 6 | Stored triggers | No Stored Triggers are found. | 1) Triggers must be used as they are helpful in validating input data and hashing the passwords to store in dataase.  2) Triggers must be implemented as they help in auditing. |
| 7 | Stored procedures | No Stored Procedures are found | Stored Procedures must be used as it can execute multiple queries in a single call. It helps in reducing network traffic and thus reducing network bottlenecks. |
| 8 | Encryption of data | Encryption mechanism is not in place.   1. Username and Passwords are stored in plain text format, not encrypted and hashed. 2. Personal information and sensitive information such as Name, UserID, Sex, Address, Phone number are stored in plain text format, not encrypted. 3. Security Question and answers are not encrypted | 1) Strong encryption algorithm must be used to protect Sensitive data.  2) Cryptographic salting with hashing must be used while storing password. |
| 9 | Links to the database | No links are found | External links are not exposed. It is a very good practice in security aspect |
| 10 | Sources of database access | Following files are having connection with database-   * Activateuser2.jsp * Addcontact2.java * Addemp.java * Addemp2.java * Addressbook.jsp * Adminlogin.jsp * Adminmainscreen.jsp * Banuser2.jsp * Changepass2.jsp * Changeroomkey2.jsp * Clientlogin.jsp * Conference.jsp * Conferencelogin.jsp * Createroom2.jsp * Deleteoffline.jsp. * Deleteroom.jsp * Deleteuser.jsp * Deleteuser2.jsp * Loginroom.jsp * Mm\_menu.js * Myprofile.jsp * Myprofil2.jsp * Myrooms.jsp * Offline.jsp * Readoffline.jsp * Register2.jsp * Registerfinal * Searchcontact2 * Statistics2 * Updatecontact2 * Viewprofile2 * ViewStatistics * Writeoffline2.jsp | 1) A good practice followed by the developers of this application is that they haven't given a direct link to the database.  2) A new session must be created each time database connection is established. |

1. Databases audited and the Results received:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Vulnerabilities Identified** | **Description** |
| **1** | Sensitive Data Exposure | Username and Passwords stored in plain text Any Threat Actor can enter the database and easily see all the username and passwords associated with it. Also the table names are kept such that any hacker would easily find where to get sensitive information |
| **2** | Insecure Direct Object Reference | Threat actor can traverse through the images used in website Even after logging out, users were able to click back and go back to logging page(No need for logging) |
| **3** | Insufficient Logging and Monitoring | No logging and Monitoring mechanism in place except checking for client |

1. Databases audited and the Results received:

|  |  |  |  |
| --- | --- | --- | --- |
| **Work allocation among team members** | | | |
|  | Member1 Vidit Gupta | Database Audit Plan  Server Deployment  Vulnerability Identification | |
| Member2 Suriya Narayanan R | Database Audit Plan  ER Diagram  Vulnerability Identification | |
| Member3 Satyabrata Saha | Accumulating Individual Contribution  Determining Audit Check-List Database Auditing Plan | |
| Member4 Hritik Saxena | Database Auditing Plan  Final report |  |
| Member5 Mohammad Adib Aslam | Indexing and Formatting of the final version. Specifying Audit Entity  Determining Audit Check-List | |

**In the given Database i.e., University Admission System, we came to definite conclusions:**

* The above database system is very poorly designed as the tables have not been identified with Primary keys (So concept of referential integrity) and all the confidential data is in plain texts.
* In ER Diagram, no relationships cannot be formed between the tables, as variable names are not properly defined and foreign key is not present to give reference for other tables.
* Stored functions, triggers and procedures should be implemented to reduce time complexity, have a check on Access and encryption
* No logging is in place. So, high chances of exploitation.
* Appropriate Access Control is defined for different users.

We were able to demonstrate our theoretical learning with this Project. As a bonus, we got exposed to server deployment too.