Database Security Lab Project

Professor: Michael Martin

Due Day: December 11, 2003

Course: CSMN 668

University of Maryland University College

The Problem Description

Recently, the ABC University has encountered the growing number of database security breaches. Too often, data is stored in the university's student registration database as "clear text" plainly visible to outsiders, dismissed employees and other potential attackers. In addition, the database has been constantly experienced security problems such as data redundancy and duplication due no access control, data were viewed by unauthorized personnel, and unauthorized logon to the database from outsiders.

Having the problems reported to the university's senior management, the designated Data Security manager was tasked to find the solution to the database security issues, which has become vulnerable to attack from both external and internal sources. After conducting full risk assessment and analysis, the senior leadership has decided to establish security practices to protect the student registration information from the database. The following are primary implementation of database security measures, which involve:

? Security policies,
? Security procedures,
? Users' account with Password,
? Assigning various roles for users based on their job activities,
? Establishing access Control,
? Granting users to access to the Oracle database.

Listing of tables and views

? There are 5 primary tables namely department, offering, student, faculty, and enrollment in the database.

```
SQL> SELECT TABLE_NAME FROM USER_TABLES;
TABLE_NAME
---------------------------
DEPARTMENT
ENROLLMENT
FACULTY
OFFERING
STUDENT

5 rows selected.

SQL> DESC DEPARTMENT;
 Name                              Null?    Type
 --------------------------------- -------- ----
 DEPT_ID                                    VARCHAR2(20)
 DEPARTMENT                                 VARCHAR2(15)
 DEPT_LOC                                   VARCHAR2(10)

SQL> DESC ENROLLMENT;
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----
 OFFERING                                           NUMBER(10)
 STUDENT_ID                                NOT NULL NUMBER(10)

SQL> DESC FACULTY
 Name                                      Null?    Type
 ----------------------------------------- -------- ----
 FACULTY_ID                                NOT NULL NUMBER(10)
 NAME                                               VARCHAR2(20)
 DEPARTMENT                                         VARCHAR2(15)
 POSITION                                           VARCHAR2(15)

SQL> DESC OFFERING
 Name                                      Null?    Type
 ----------------------------------------- -------- ----
 OFFERING                                  NOT NULL NUMBER(10)
 COURSE                                             VARCHAR2(5)
 FACULTY_ID                                         NUMBER(10)
 TERM                                               VARCHAR2(10)
 YEAR                                               VARCHAR2(4)
 TIME                                               VARCHAR2(5)

SQL> DESC STUDENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----
 STUDENT_ID                                NOT NULL NUMBER(10)
 NAME                                               VARCHAR2(30)
 ADDRESS                                            VARCHAR2(20)
 CITY                                               VARCHAR2(25)
 STATE                                              VARCHAR2(2)
 ZIPCODE                                            VARCHAR2(9)
 MAJOR                                              VARCHAR2(30)
 STATUS                                             VARCHAR2(10)
 AGE                                                NUMBER(2)
 GPA                                                NUMBER(4,3)

Display table contents:
     SQL> SELECT * FROM DEPARTMENT;

     DEPT_ID              DEPARTMENT      DEPT_LOC
     -------------------- --------------- ----------
     P1000                SA              Building2
     B2000                IS              Building5
     P3000                IM              Building2
     M4000                IT              Building5
```

```
SQL> SELECT * FROM ENROLLMENT;

 OFFERING STUDENT_ID
--------- ----------
     1111        100
     1233        500
     2222        300
     3333        400

SQL> SELECT * FROM FACULTY;

FACULTY_ID NAME                 DEPARTMENT      POSITION
---------- -------------------- --------------- ---------------
       980 MARTIN               IM              Professor
      5430 SEAVER               IS              Professor
      7650 LOONEY               IT              Instructor
      9870 MILLS                SA              Lecturer

SQL> SELECT * FROM OFFERING;

 OFFERING COURS FACULTY_ID TERM       YEAR TIME
--------- ----- ---------- ---------- ---- -----
     1111 IS320       5430 Fall       2001 10 AM
     1233 IS320        980 Fall       2001 11 AM
     2222 IS460       7650 Spring     2002 10 AM
     3333 IT480       5430 Spring     2002 11 AM

SQL> SELECT * FROM STUDENT;

STUDENT_ID NAME                           ADDRESS
CITY                      ST
---------- ------------------------------ -------------------- -
---------------------- --
ZIPCODE   MAJOR                          STATUS           AGE
GPA
--------- ------------------------------ ---------- --------- --
-------
       100 ABLE                           12 Kinds Rd
Rockville                 MD
20850     History                        SR                21
3

       200 BAKER                          89 TwinK Ct
Bethesda                  MD
20877     Accounting                     JR                19
2.7
```

```
       300 CHARLES                          440 Rock Rd
Germantown                MD
20827     Math                             SR                22
3.5

       400 DRAKE                            34 Lock Rd
Bowie                     MD
29845     Computer Science                 FR                18
2.8

       500 ELLIOT                           66 Montrose Rd
Rockville                 MD
20850     Biology                          SM                19
3.25
```

There are 11 views, which created from above base tables:
SQL> select view_name from user_views;

VIEW_NAME

FACULTY_VIEW1
FACULTY_VIEW2
FACULTY_VIEW3
FACULTY_VIEW4
OFFERING_VIEW
STUDENT_MAJOR_STATUS
STUDENT_VIEW1
STUDENT_VIEW2
STUDENT_VIEW3
STUDENT_VIEW4
STUDENT_VIEW5

11 rows selected.

There are 5 synonyms, which created from the above base tables depend upon offering table:
OFFERING table is acted as synonyms for 5 student user accounts, namely ABLE12, BAKER13, CHARLES14, DRAKE15, ELLIOT16
OFFERING as synonyms for 5 faculty member accounts, namely MARTIN22, SEAVER23, LOONEY24, MILLS25

This lab project, which is described with security policy, procedures, and lab results with examples, is based on the level of security for each scenario**.**

Security Policy:
Management shall create users' accounts for student and faculty member's accounts with passwords in order to connect and view any objects in Oracle database base on their job activities.

Procedures:

? Enforcement of user's names must contain at least 1 numeric and 1 character for the user name and at least 6 character length,
? Passwords must be at least 8-character lengths plus combinations of digits and characters.
? Various users are assigned to certain objects such as tables, views, and synonyms based on their access roles.
? Granting users with appropriate privilege and roles to the database.

Lab report from Oracle database.

? From the DBA user account (yyzz), creating student accounts to access Oracle database.

User name: ABLE12

SQL> create user ABLE12 identified by ab12q6t2;
User created
User Name: BAKER13
SQL statement: SQL> create user BAKER13 identified by bk13c9e3;
User created
User Name: CHARLES14
SQL> create user CHARLES14 identified by ca14t6u2;
User Name: DRAKE15
SQL statement: SQL> create user DRAKE15 identified by dk15r3d2;
User Name: Elliot16
SQL> create user ELLIOT16 identified by eT16k3r4;

****

Example: Try to logon to the database without grant connect and resource to users
Example: user CHARLES14 can not logon to the database due to lack of creating session privilege.

/class/mn668a/07% sqlplus

SQL*Plus: Release 9.0.1.4.0 - Production on Sun Dec 7 23:05:37 2003

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Enter user-name: CHARLES14
Enter password:
ERROR:
ORA-01045: user CHARLES14 lacks CREATE SESSION privilege; logon denied

Procedures:

? DBA shall grant privileges to Students, so they shall have the authority to access Oracle database.

SQL> grant connect, resource to ABLE12, BAKER13, CHARLES14, DRAKE15, ELLIOT16;

SQL> /
Grant succeeded.
****

Example: User DRAKE15 successfully logon to the Oracle database after issuing above sql statement.

Enter user-name: DRAKE15
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.0.1.4.0 - Production
****

Procedures:

? DBA shall grant privileges to Students, so they shall have the authority to access specific tables through views or synonyms for their own record:

```
SQL> create view student_view1 as select * from student
        where name = 'ABLE12';

SQL>/
View created

SQL> create view student_view2 as select * from student
        where name = 'BAKER13';

SQL>/
View created

SQL> create view student_view3 as select * from student
        where name = 'CHARLES14';

SQL>/
View created

SQL> create view student_view4 as select * from student
        where name = 'DRAKE15';

SQL>/
View created

SQL> create view student_view5 as select * from student
        where name = ELLIOT16 '';
```

```
SQL>/
View created

SQL> grant SELECT ON STUDENT_VIEW1 TO ABLE12;

Grant succeeded.

Test grant statement:
GRANT SELECT ON STUDENT_VIEW3 TO CHARLES14
SQL> /

Grant succeeded.

/class/mn668a/07% sqlplus

SQL*Plus: Release 9.0.1.4.0 - Production on Mon Dec 8 01:18:02 2003

(c) Copyright 2001 Oracle Corporation.  All rights reserved.

Enter user-name: CHARLES14
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.0.1.4.0 – Production

**************************************************************
Example from Output: user Charles14 can only view his own information.
SQL> select * from yyzz.student_view3;
SQL> /

STUDENT_ID NAME                      ADDRESS
---------- ---------------------------- --------------------
CITY                 ST ZIPCODE   MAJOR                     STATUS
-------------------------- -- --------- ------------------------------ ----------
     AGE       GPA USERNAME
---------- ---------- ------------------------------
     300 CHARLES14                440 Rock Rd
Germantown             MD 20827     Math                      SR
      22       3.5 CHARLES14
      23
```

Procedures:

? DBA shall create student role for all students and grant privileges to 5 Students so they can access OFFERING TABLE information from DBA user (yyzz)'s account:

SQL>create role student;
SQL>GRANT SELECT ON OFFERING TO student;
SQL>GRANT student ON OFFERING TO ABLE12, BAKER13, CHARLES14, DRAKE15, Elliot16;

Lab report from Oracle database.

From user ABLE12's account, creating synonyms from OFFERING TABLE as below:
SQL>CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

From BAKER13's account, creating synonyms from OFFERING TABLE as below:
SQL>CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

From CHARLES14's account, creating synonyms from OFFERING TABLE as below:
SQL>CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

From DRAKE15's account, creating synonyms from OFFERING TABLE as below:
SQL>CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

From Elliot16's account, creating synonyms from OFFERING TABLE as below:
SQL>CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

Example: Go to any student's account to check that students can select offering relation from their accounts:
SQL> select * from offering;

OFFERING	COURS	FACULTY_ID	TERM	YEAR	TIME
1111	IS320	5430	Fall	2001	10 AM
1233	IS320	980	Fall	2001	11 AM
2222	IS460	7650	Spring	2002	10 AM
3333	IT480	5430	Spring	2002	11 AM

Lab report from Oracle database.

? From the DBA user account (yyzz), creating faculty member accounts to access Oracle database.
  ? Create accounts with passwords:

SQL statement: SQL> create user MARTIN22 identified by mt28w5u7;

SQL statement: SQL> create user SEAVER23 identified by sv60g2e6;

SQL statement: SQL> create user LOONEY24 identified by ln30y1j2;

SQL statement: SQL> create user MILLS25 identified by mi25t3k2;

? Grant faculty authority to access oracle database and to create tables, procedures, and triggers;

SQL> grant connect, resource to MARTIN22, SEAVER23, LOONEY24, MILLS25;

Grant succeeded.

****

Example: Try to Logon any faculty user account, in this case, is Dr. Martin, to test output
Enter user-name: MARTIN22
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.0.1.4.0 - Production

Security Policy:
Management shall provide certain authority to modify the offering table.

Procedures:
Faculty members shall have the authority to create, update, and delete any record in the offering table

SQL> CREATE SYNONYM OFFERING FOR yyzz.offering;

Synonym created.

? Before any modification, the table offering looks like:

SQL> select * from offering;

OFFERING	COURS	FACULTY_ID	TERM	YEAR	TIME
1111	IS320	5430	Fall	2001	10 AM
1233	IS320	980	Fall	2001	11 AM
2222	IS460	7650	Spring	2002	10 AM
3333	IT480	5430	Spring	2002	11 AM

after insert a record to offering:
SQL> insert into offering values (4444, 'BI229', '9999', 'Summer', '2001', '12 P
M');

1 row created.

SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
      4444 BI229       9999 Summer     2001 12 PM
```

after update a record to offering table:
SQL> update offering set course = 'EN888' where offering = 4444;

1 row updated.

SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
      4444 EN888        9999 Summer     2001 12 PM
```

Delete this record where offering number equal 4444;
SQL> delete offering where offering = 4444;

1 row deleted.
Testing the output from MARTIN22 user account:
SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
```

Testing effect on yyzz account to see that 1 more record has been added after commit from faculty MARTIN22:
SQL> commit;

Commit complete.

SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
      4444 EN888        9999 Summer     2001 12 PM
```

Delete this record from faculty MARTIN22's account:
Delete this record where offering number equal 4444;
SQL> delete offering where offering = 4444;

1 row deleted.
Testing the output from MARTIN22 user account:
SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
```

Issuing commit in MARTIN22's account:
SQL> commit;

Commit complete.
Testing the change again in the yyzz account and show:
SQL> select * from offering;

```
 OFFERING COURS FACULTY_ID TERM       YEAR TIME
---------- ----- ---------- ---------- ---- -----
      1111 IS320       5430 Fall       2001 10 AM
      1233 IS320        980 Fall       2001 11 AM
      2222 IS460       7650 Spring     2002 10 AM
      3333 IT480       5430 Spring     2002 11 AM
```

Security Policy:

Management shall provide certain privileges for faculty member to view certain object(s) in the database, and grant 1 faculty member with DBA privileges,

Management shall provide certain privileges for students to view certain object(s) in the database.

Procedures:

? Faculty members shall have the authority to view all attributes of the faculty relation, but only for their own record;

Creating views based on each faculty 's account
SQL>create view FACULTY_VIEW1 AS SELECT * FROM FACULTY WHERE NAME = 'MARTIN22';
View created.

SQL> SELECT * FROM FACULTY_VIEW1;

FACULTY_ID	NAME	DEPARTMENT	POSITION
980	MARTIN22	IM	Professor

SQL> SELECT * FROM FACULTY_VIEW2;

FACULTY_ID	NAME	DEPARTMENT	POSITION
5430	SEAVER23	IS	Professor

SQL> SELECT * FROM FACULTY_VIEW3;

FACULTY_ID	NAME	DEPARTMENT	POSITION
7650	LOONEY24	IT	Instructor

SQL> SELECT * FROM FACULTY_VIEW4;

FACULTY_ID	NAME	DEPARTMENT	POSITION
9870	MILLS25	SA	Lecturer

Procedures:

? Grant the following views to each faculty members:
SQL> GRANT SELECT ON FACULTY_VIEW1 TO MARTIN22;

Grant succeeded.

SQL> GRANT SELECT ON FACULTY_VIEW2 TO SEAVER23;

Grant succeeded.

SQL> GRANT SELECT ON FACULTY_VIEW3 TO LOONEY24;

Grant succeeded.

SQL> GRANT SELECT ON FACULTY_VIEW4 TO MILLS25;

Grant succeeded.

****
Example: Test by logon randomly as one of 4 users SEAVER23 can successfully access Oracle database
Enter user-name: SEAVER23
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.0.1.4.0 - Production
****

SQL> select * from yyzz.faculty_view2;

FACULTY_ID	NAME	DEPARTMENT	POSITION
5430	SEAVER23	IS	Professor

Procedures:

? Dr. Martin has the authority to grant and revoke user accounts and passwords from yyzz account, granting dr. martin create and drop user account privileges

SQL>grant create user, drop user to MARTIN22;

grant succeed

****
Example: checking out the see whether dr. martin has the privileges on the privileges by logon
martin22 account:
Enter user-name: MARTIN22
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.4.0 - 64bit Production
With the Partitioning option

JServer Release 9.0.1.4.0 - Production

Procedures:

? Grant each and all students;
1. the authority to view their own info in the student, offering and enrollment relations
2. grant students with juniors and senior standings to change their majors:

```
FROM YYZZ ACCOUNT
CREATE VIEW STUDENT_MAJOR_STATUS AS SELECT MAJOR FROM
STUDENT
  2  WHERE STATUS = 'SR'
  3* or STATUS = 'JR'
SQL> /

View created.
SQL> GRANT SELECT ON STUDENT_MAJOR_STATUS TO ABLE12, BAKER13,
CHARLES14;

Grant succeeded.
SQL> GRANT UPDATE ON STUDENT_MAJOR_STATUS TO ABLE12, BAKER13,
CHARLES14;

Grant succeeded.

****************************************************************
Example: TESTING OUTPUT FROM 1 OF THE Student (able12)'s account:
  1* SELECT * FROM YYZZ.STUDENT_MAJOR_STATUS
SQL> /

MAJOR
------------------------------
History
Accounting
Math

after update take place:

 1  UPDATE  YYZZ.STUDENT_MAJOR_STATUS
  2  SET MAJOR = 'Music'
  3  WHERE MAJOR =
  4* 'History'
SQL> /

1 row updated.

SQL> select * from yyzz.student_major_status;
```

MAJOR

Music
Accounting
Math