1) To Execute and Simulate basic Network Security Commands

1. Ipconfig
2. Ipconfig/all
3. Ipconfig/release
4. Ipconfig/renew
5. Ping www.google.com
6. Nslookup www.google.com
7. Hostname
8. Getmac
9. Netstat
10. Arp-a

2) To implement and execute access control commands in MySql

```
mysql-ctl cli
mysql> create database testuser;
mysql> create user Account_manager;
mysql> create user Customer_service;
mysql> create user sales_reps;
mysql> USE testuser;
mysql> create table customers(Role varchar(255));
mysql> show tables;
mysql> GRANT select,insert,delete ON testuser.customers to Account_manager;
mysql> GRANT select ON testuser.customers to Customer_service;
mysql> GRANT select,insert,update ON testuser.customers to sales_reps;
mysql> show GRANTS for Account_manager;
mysql> show GRANTS for Customer_service;
mysql> show GRANTS for sales_reps;
```

3) To implement and execute security commands in Linux

Listing

- ls

Long listing

- ls -l

create file

- cat >abc
- how r u

view file

- cat abc

creating more files

- touch file1 file2 file3

adding user

- useradd friend
- passwd
- friend
- login friend
- pwd
- friend

giving permission

- chmod +x file3
- ls -l
- chmod -x file3
- ls -l
- chmod u-x,g+w,o-rw file 2
- ls -l

4) To implement security policies in Ubuntu OS environment

- #apt-get install nano
- #apt-get install libpam-pwquality 2
- Cd ../
- Ls -l
- Cd etc
- Ls -l
- Cd pam.d
- Nano common-password

Update this there

password requisite pam_pwquality.so retry=3

password requisite pam_pwquality.so minlen=10 maxrepeat=3

password requisite pam_pwquality.so ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1

password required pam_permit.so reject_username

password required pam_permit.so difok=3

- useradd user1
- passwd user1
- nano /etc/login.dfs

PASS_MAX_DAYS 90

PASS_MIN_DAYS 0

PASS_WARN_AGE 5

5) To implement IDS system using Snort tool in Linux Environment

ifconfig
apt-get install nano
apt-get install snort
cd /ect/snort
nano snort.conf
Update this there

- Ipvar HOME_NET 172.17.0.26

cd /ect/snort/rules
nano local.rules

- Alert tcp any any -> any any(msg:"Detection of TCP connection" sid:10001;)

snort -T -c /etc/snort/snort.conf
snort –A console -c /etc/snort/snort.conf (Dont copy paste it wont work)

New terminal

#apt-get install nmap
#apt-get install inetutils-ping
#ping 172.17.0.17

6) Create the three user defined roles that are shown in the following table and assign the specified permissions for the CUSTOMER table

ROLE	SELECT	INSERT	UPDATE	DELETE
Account_Manager	✓		✓	✓
Customer_ Service	✓			
Sales_Reps	✓	✓	✓	

```
mysql-ctl cli
mysql> create database testuser;
mysql> create user Account_manager;
mysql> create user Customer_service;
mysql> create user sales_reps;
mysql> USE testuser;
mysql> create table customers(Role varchar(255));
mysql> show tables;
mysql> GRANT select,insert,delete ON testuser.customers to Account_manager;
mysql> GRANT select ON testuser.customers to Customer_service;
mysql> GRANT select,insert,update ON testuser.customers to sales_reps;
mysql> show GRANTS for Account_manager;
mysql> show GRANTS for Customer_service;
mysql> show GRANTS for sales_reps;
```

7) Create a Database of Customer and Create the following application roles with corresponding Permissions

Role	Permission
Sales	**Select, Update, Insert into Customers and Orders, Order_Details**
Sales_Manager	**Sales and Delete on Orders and Order Details**

```
mysql-ctl cli
mysql> Create database Customer;
mysql> CREATE USER Sales IDENTIFIED BY 'root';
mysql> CREATE USER Sales_Manager IDENTIFIED BY 'root';
mysql> USE customer;
mysql> create table customers(Role varchar(255));
mysql> create table orders(Role varchar(255));
mysql> create table order_details(Role varchar(255));
mysql> GRANT Select,Update,Insert ON Customer.customers TO SALES;
mysql> GRANT Select,Update,Insert ON Customer.order TO SALES;
mysql> GRANT Select,Update,Insert ON Customer.order_details TO SALES;
mysql> GRANT Select,Update,Insert,Delete ON Customer.customers TO SALES_Manager;
mysql> GRANT Select,Update,Insert,Delete ON Customer.orders TO SALES_Manager;
mysql> GRANT Select,Update,Insert,Delete ON Customer.order_details TO SALES_Manager;
mysql> show GRANTS for sales;
mysql> show GRANTS for sales_Manager;
```

8) You have been hired as an Oracle/SQL Database Administrator by a small Company for the primary task of creating a security model which consists of three levels

READ Level

UPDATE and INSERT Level

DELETE Level

This Model will be using tables owned by HR Schema

```
mysql-ctl cli
mysql> Create database HRSCHEMA;
mysql> CREATE USER HR IDENTIFIED BY 'root';
mysql> CREATE USER EMP IDENTIFIED BY 'root';
mysql> USE HRSCHEMA;
mysql> create table Location(Place varchar(255), city varchar(255));
mysql> create table Job(Job_title varchar(255), job int);
mysql> create table Dept(Dept_title varchar(255), Dept int);
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Location TO HR;
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Job TO HR;
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Dept TO HR;
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Location TO Emp;
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Job TO Emp;
mysql> GRANT insert,Delete,Update,Select ON HRSCHEMA.Dept TO Emp;
mysql> show GRANTS for HR;
mysql> show GRANTS for Emp;
```

9) Using the Concepts in the Security data model based on application table, set up the following permission for the AUTHORS table in the Database

USER	ACCESS
John	Select, Update, Insert
Jane	Select, Update, Insert, Delete
Sally	Select

```
mysql-ctl cli

mysql> Create database Customer;
mysql> CREATE USER John IDENTIFIED BY 'root';
mysql> CREATE USER Jane IDENTIFIED BY 'root';
mysql> CREATE USER Sally IDENTIFIED BY 'root';
mysql> USE customer;
mysql> create table customers(Role varchar(255));
mysql> GRANT Select,Update,Insert ON Customer.customers TO John;
mysql> GRANT Select,Update,Insert,Delete ON Customer.order TO Jane;
mysql> GRANT Select ON Customer.order_details TO Sally;
mysql> show GRANTS for John;
mysql> show GRANTS for Jane;
mysql> show GRANTS for Sally;
```

10. Using Access level control, set up row and column level security on the Employee table, Clerks can only edit phone number and add new rows, but managers can perform all the DML operations in SQL Server

mysql-ctl cli
CREATE DATABASE EMP;
CREATE USER **CLERK** IDENTIFIED BY ‘ root’;
CREATE USER **MANAGER** IDENTIFIED BY ‘root’;
Show databases;
Use EMP;
CREATE TABLE EMPLOYEE(empid int,phone_number int,emp_details varchar(255));
GRANT **UPDATE,INSERT** ON EMP.EMPLOYEE TO CLERK;
GRANT **INSERT,UPDATE,DELETE** ON EMP.EMPLOYEE TO MANAGER;
SHOW GRANTS FOR CLERK;
SHOW GRANTS FOR MANAGER;