

7.2 SOLUTIONS AND GUIDED WALKTHROUGH

7.2.1 SOLUTIONS

1. *Secure MySQL Root Account*

MySQL is an application which has the capability for a user to harden the root account and set a strong password by running the command:

```
sudo mysql_secure_installation
```

This will begin an interactive dialog in which you will set the MySQL root account password.

2. *Harden MySQL Permissions*

To ensure **testuser** has read-only access on all tables in the *vulndb* database, run the following MySQL command:

```
GRANT SELECT ON vulndb.* TO 'testuser'@'localhost';
```

To ensure that this permission change takes place, run the following command:

```
FLUSH PRIVILEGES
```

This will reload the privileges table.

3. *Change Default Ports*

To change the default port which MySQL uses, open the file located at */etc/my.cnf* and change the line which indicates which port is being used or add a line **port = <xxxx>** if one does not already exist.

4. *Remove Unnecessary Databases or Users*

You can check the list of users on a MySQL server by running the MySQL command:

```
SELECT user, host FROM mysql.user;
```

Here you will find a user named **haxxor**, that is the malicious user. To remove a user, run the MySQL command:

```
DROP USER '<username>'@'<host>'
```

By navigating through the databases, you will find one titled *dontlook*. This database is empty and is the unnecessary table associated with this task. To remove a table, run the

MySQL command:

```
DROP TABLE <tablename>.
```

5. *Using MySQL*

For this task, reference the list of commands below, they should be sufficient to help you perform the tasks.

- Create a Database

```
CREATE DATABASE <dbname>;
```

- List all Databases

```
SHOW DATABASES;
```

- Enter a Database

```
USE <dbname>;
```

- List all Tables in a Database

```
SHOW TABLES;
```

- Create a Table

```
CREATE TABLE <name> (<field_1 type_1,...,field_n type_n>);
```

- Create a Table Entry

```
INSERT INTO <name> (<field_1,...,field_n>) VALUES (<value_1,...,value_n>);
```

- Delete an Entry

```
DELETE FROM <name> WHERE <field> = <value>;
```

- Add a Field to a Table

```
ALTER TABLE <name> ADD COLUMN <col_name col_type>;
```

6. *Querying MySQL Databases*

Queries in MySQL can be formatted in many ways. A common way to get results matching a query is to the following syntax:

```
SELECT <fields> FROM <name> WHERE <field> = <value>;
```

Answers:

- Jeanette Wise lives in New York.

- Jeremiah Houston's salary is \$50,000.
- Emma Castillo makes \$500,000 in Los Angeles.
- Tami Vasquez is 12 years old.
- Wanda Lloyd works as a Barista.
- The 40 year-olds are a teacher and a lawyer.

7.2.2 GUIDED WALKTHROUGH

In order to complete the challenges in this laboratory exercise, see the steps in this walk-through.

Databases are at the heart of many organizations. They allow for organized, electronic record storage. Additionally, using a database management system such as MySQL, you can perform powerful queries to manipulate specific data. Understanding how to manipulate the data in a database using MySQL and ensuring that your MySQL server is secure are critical.

In MySQL, databases are comprised of tables which hold records (data entries). A table is defined with specific fields (columns) of data which the individual records, or entries, will have. You can access the MySQL prompt by entering the following command in a terminal for a user named *sampleuser*:

```
$ mysql -u sampleuser
```

Challenge 1

A default installation of MySQL leaves the root user with no password, allowing anyone to access the databases with root privileges. You can set a password for the root user by running the following command:

```
$ sudo mysql_secure_installation
```

This will begin an interactive dialog in which you will set the MySQL root account password.

Challenge 2

MySQL users can have many different privileges on various databases in the system. MySQL allows you to configure what databases and tables specific users have certain privileges on. For this challenge, you are tasked with ensuring that **testuser** has read-only access on all tables in the *vulndb* database. To do this, enter the following command in the MySQL prompt:

```
GRANT SELECT ON vulndb.* TO 'testuser'@'localhost';
```

In this command, SELECT is the name of the privilege which will be granted, vulndb is the database which the privilege will be granted on, the asterisk (*) specifies that all tables in the preceeding database should be affected, and 'testuser'@'localhost' is the name and host of the user who will be affected by this granting of privileges. To ensure that the permission change takes place, run the following command in MySQL:

```
FLUSH PRIVILEGES;
```

This will reload the privileges table.

Challenge 3

The various services on a machine run on different ports. By default, MySQL runs on port 3306 (MySQL traffic goes in and out of port 3306). It can be helpful to change the default port to something else. To change the port, open the file located at **/etc/my.cnf**. In this file, find the line which specifies the port being used. The line should read: **port 3306**. If the line does not already exist, add the line, changing 3306 to something else. Be sure that the new port is not already in use.

Challenge 4

MySQL has various users who have different accesses to the data in a MySQL server. Ensuring that only necessary users have access to the data, follow the instructions below.

You can check the list of users on a MySQL server by running the MySQL command:

```
SELECT user, host FROM mysql.user;
```

Here you will find a user named **haxxor**, that is the malicious user. To remove a user, run the MySQL command:

```
DROP USER '<username>'@'<host>';
```

By navigating through the databases, you will find a table titled *dontlook*. This table contains just ID numbers and is the unnecessary table associated with this task. To remove a table, run the MySQL command:

```
DROP TABLE <tablename>;
```

Challenge 5

For this task, reference the list of commands below, they should be sufficient to help you

perform the tasks.

- Create a Database

```
CREATE DATABASE <dbname>;
```

- List all Databases

```
SHOW DATABASES;
```

- Enter a Database

```
USE <dbname>;
```

- List all Tables in a Database

```
SHOW TABLES;
```

- Create a Table

```
CREATE TABLE <name> (<field_1 type_1,...,field_n type_n>;
```

- Create a Table Entry

```
INSERT INTO <name> (<field_1,...,field_n>) VALUES (<value_1,...,value_n>;
```

- Delete an Entry

```
DELETE FROM <name> WHERE <field> = <value>;
```

- Add a Field to a Table

```
ALTER TABLE <name> ADD COLUMN <col_name col_type>;
```

To create a new database named **television**, use the following command:

```
CREATE DATABASE television;
```

To begin running commands within that database, use the following command:

```
USE television;
```

To create a table named **shows** with fields **name** and **startyear**, use the following command:

```
CREATE TABLE shows (name VARCHAR(255), startyear INT);
```

To add another field named **endyear** to the **shows** table, use the following command:

```
ALTER TABLE shows ADD COLUMN endyear INT;
```

To create the entries for the designated television shows, use the INSERT INTO command.

To remove the entry for Boy Meets World, use the DELETE FROM command.

Challenge 6

Queries in MySQL can be formatted in many ways. A common way to get results matching a query is to the following syntax:

```
SELECT <fields> FROM <name> WHERE <field> = <value>;
```

Answers:

- Jeanette Wise lives in New York. Query for location = “New York”.
- Jeremiah Houston’s salary is \$50,000. Query for name = “Jeremiah Houston”.
- Emma Castillo makes \$500,000 in Los Angeles. Query for location = “Los Angeles” and salary = “500000”
- Tami Vasquez is 12 years old. Query for name = “Tami Vasquez”.
- Wanda Lloyd works as a Barista. Query for name = “Wanda Lloyd”.
- The 40 year-olds are a teacher and a lawyer. Query for age = “40”.