|  |  |
| --- | --- |
| Penetration Test Report Prepared for Hotel Dorsey | Corporate Logo Haverbrook security lab corporate logo  Name: Michael R. Lee  Team Number: 0  Student Number: 3 |

**Introduction**

Good day! My name is Michael Lee, and I am pleased to present a penetration test report on my findings since my team’s initial report a short while ago. The scope of my test will be attempting to gain access to port 3306, running Hotel Dorsey’s MySQL database, to see if I am able to uncover any sensitive or confidential information. In this test I will be using OpenVAS (vulnerability scanner), Nmap (network mapper/scanner), and Metasploit (penetration testing framework module). I will outline in repeatable steps how MySQL can be exploited on Hotel Dorsey’s Network.

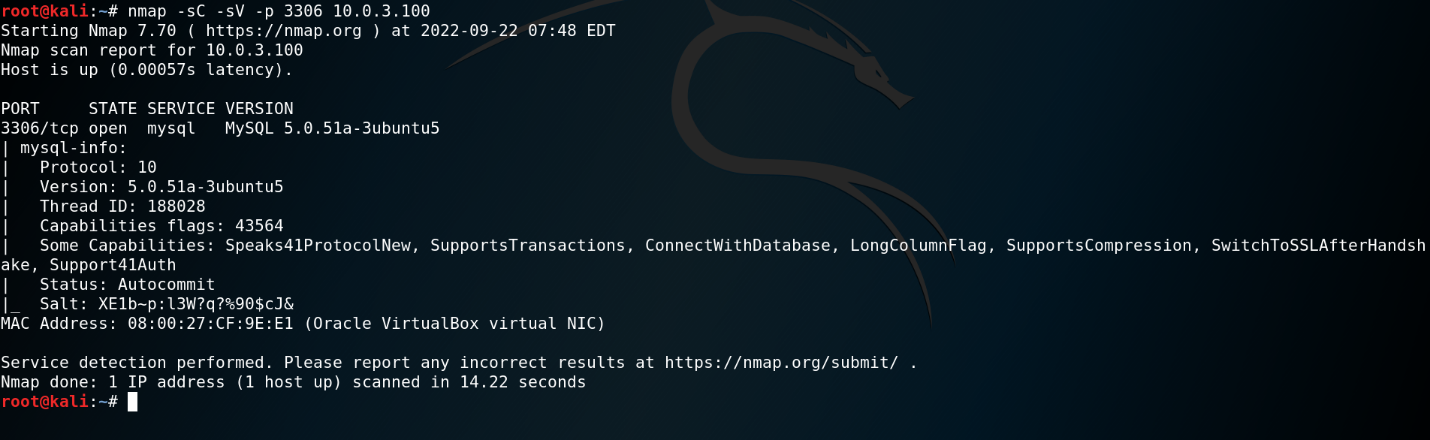
**Target**

The target IP address for this test is 10.0.3.100 and the IP address for the machine I will be using to test the network is 10.0.3.50. In my previous report, I stated the number of open ports on IP address 10.0.3.100. The following are the open ports associated with this IP address:

|  |  |  |
| --- | --- | --- |
| Port | Service | Explanation |
| 21 | FTP | File Transfer Protocol |
| 22 | SSH | Secure Shell, Secure Logins, Port Forwarding |
| 23 | TELNET | Telnet Protocol-unencrypted text communications |
| 25 | SMTP | Simple Mail Transfer Protocol-email routing between mail servers |
| 53 | DOMAIN | Domain Name System name resolver |
| 80 | HTTP | Hypertext Transfer Protocol, connection to the internet |
| 111 | RCPBIND |  |
| 139 | NETBIOS-SSN | NetBIOS Session Service |
| 445 | MICROSOFT-DS | Samba |
| 512 | EXEC | Netkit-rsh |
| 513 | LOGIN | OpenBSD or Solaris rlogind |
| 514 | SHELL | Netkit rshd |
| 1099 | RMIREGISTRY | Java RMI Registery |
| 1524 | INGRESLOCK | Metasploitable root shell |
| 2049 | NFS | Network file system 2-4 |
| 3306 | MYSQL | MySQL database system |
| 5432 | POSTGRESQL | PostgreSQL database system |
| 6667 | IRC | Internet Relay Chat |
| 8009 | AJP13 | Apache Jserv |
| 8180 | UNKNOWN | Regular Nmap scan pulled no results for this port |

Table 1: List of All Open Ports

As I stated above, I will be testing port 3306. The following is a screenshot of my scan using Nmap:

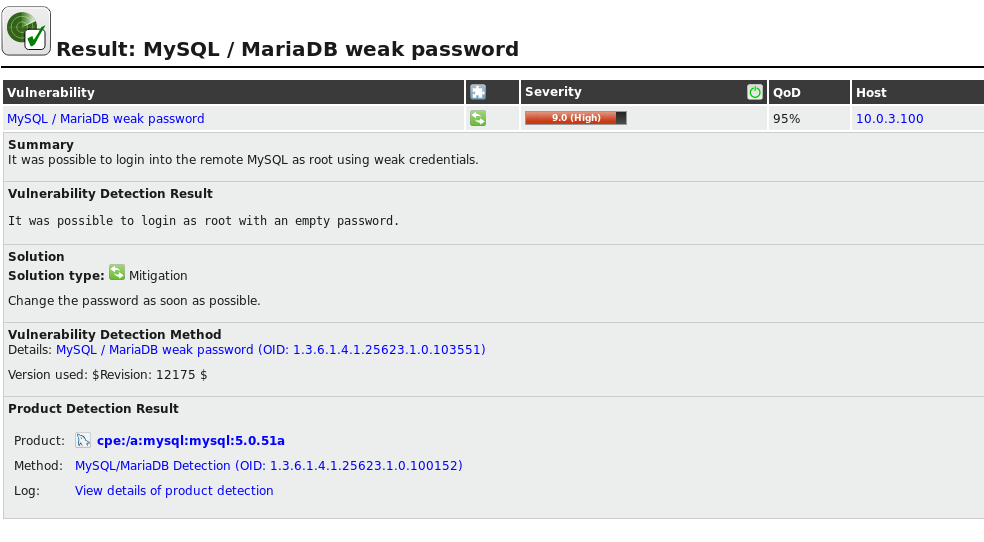


Screenshot 1: Nmap Scan of Port 3306 (MySQL)

In the above screenshot you can see that port 3306 is open, which means it is accessible to any machine on the network. In the sections below I will outline how I am able to exploit this service and gain access to privileged information.

**Vulnerability**

The vulnerabilities I am exploiting in my demonstration are empty passwords and accounts with unnecessary escalated privileges. I started by using OpenVAS to get a vulnerability analysis of port 3306:

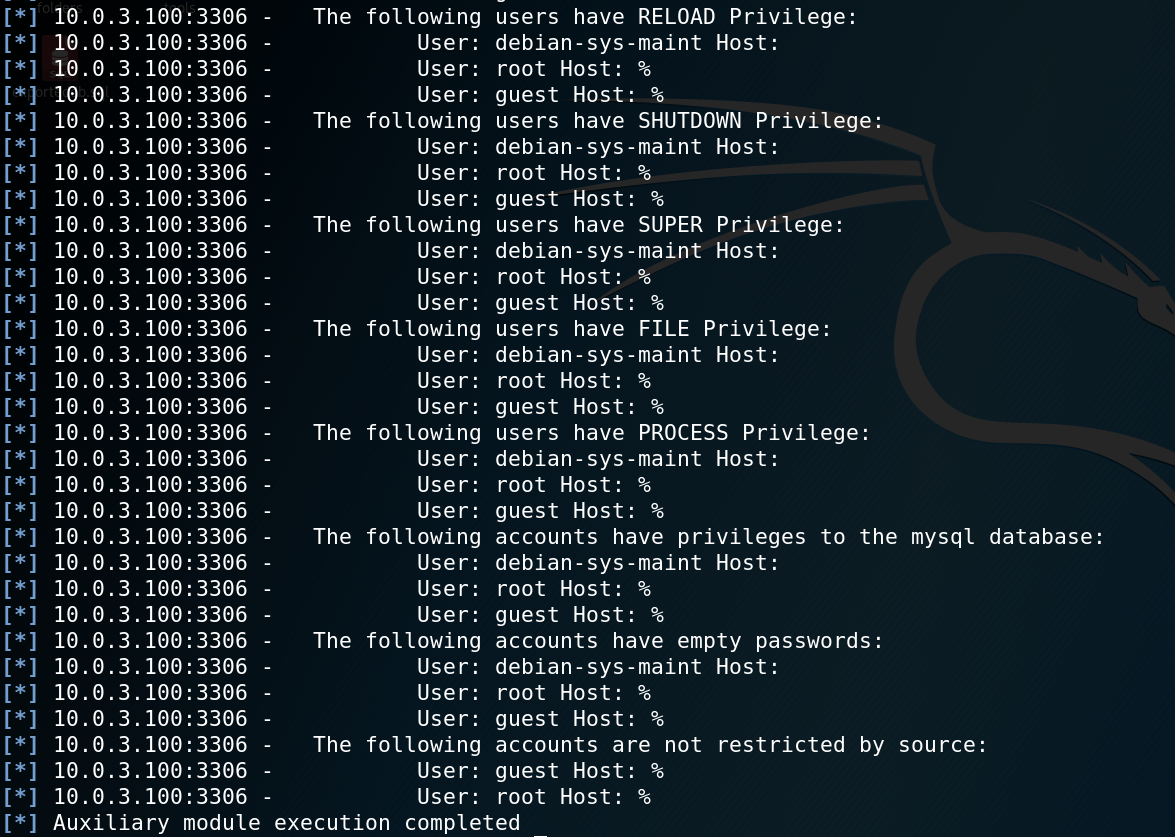
Screenshot 2: OpenVAS Results for MySQL port 3306

As you can see in the above screenshot, the result of the scan shows that root (the account with the most privileges in the system) is possible to be logged in to with an empty, or no password. This leaves your database extremely vulnerable. Before attempting to log in with the information I uncovered above I will use Metasploit to enumerate port 3306 or simply put, check to see if there are any other vulnerable areas. The following screenshot shows the results of the Metasploit scan:

Text

Description automatically generated

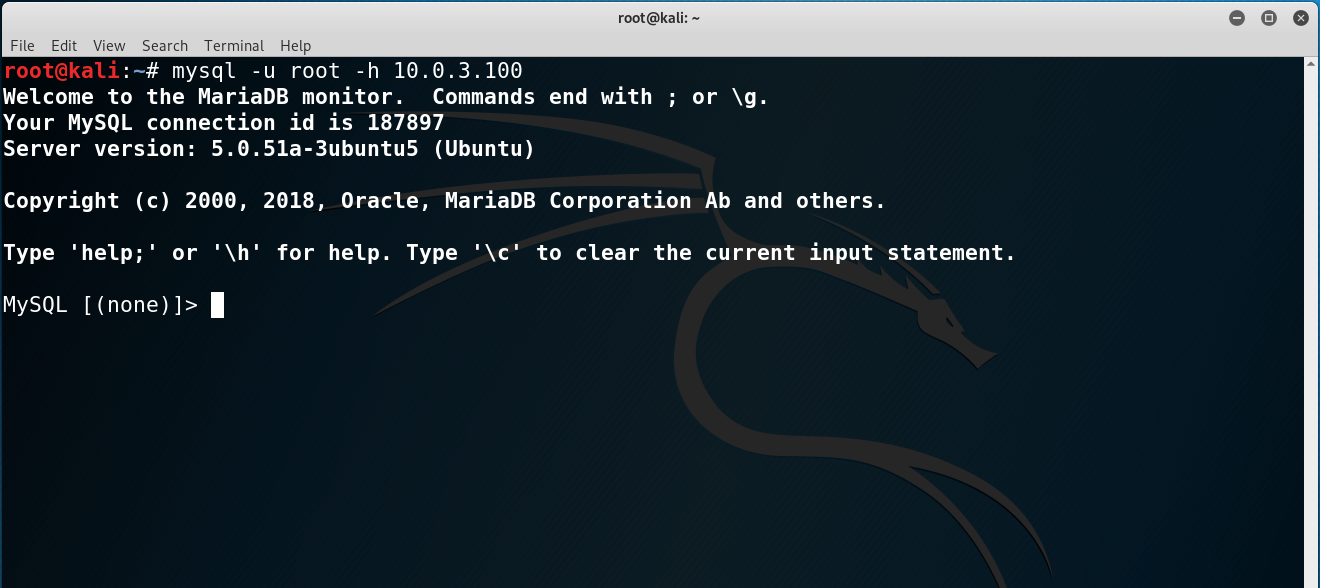
Screenshot 3: Metasploit Results Page 1



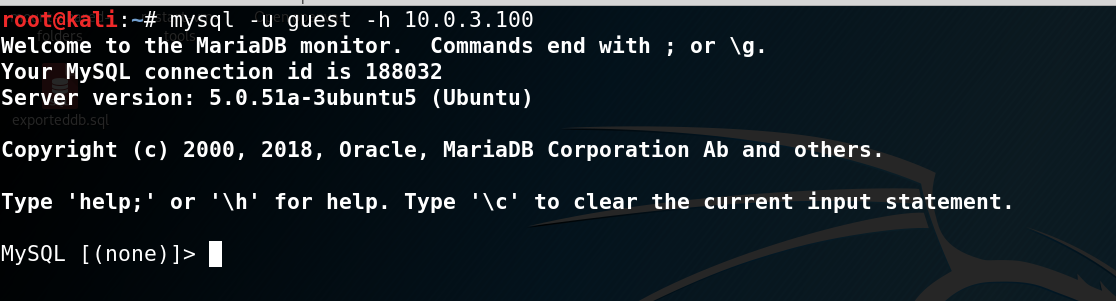
Screenshot 4: Metasploit Results Page 2

In the screenshots above I would like to direct your attention to the areas under enumerating accounts starting on page 1 of the scan results. Metasploit searched for accounts on port 3306 and found three accounts: root, guest, and debian-sys-maint host. All three accounts do not have a password hash otherwise known as empty passwords and two accounts have all privileges to make changes to the database.

With this data I will proceed to gain access to the MySQL server. I will be focusing on both root and guest users since both have all privileges. The following screenshot shows me successfully logging in with no password:



Screenshot 5: Logging in to MySQL as root

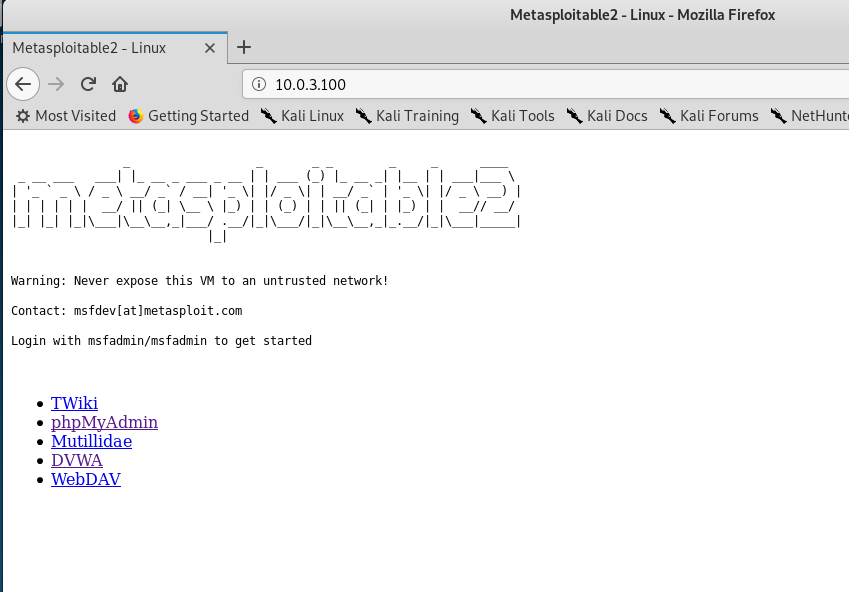


Screenshot 6: Logging in to MySQL as guest

For simplicity, I will be explaining the remainder of the databases vulnerabilities from the web browser. The information I received from Metasploit and OpenVAS means I can now try to find information I can extract from the database.

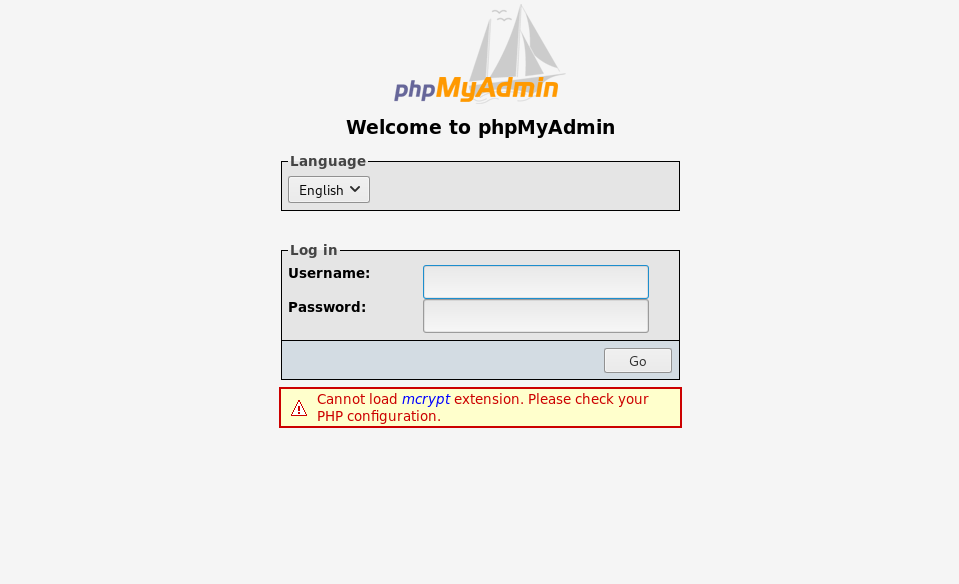
**Data Exfiltration**

To extract any data from the database, I first need to login.



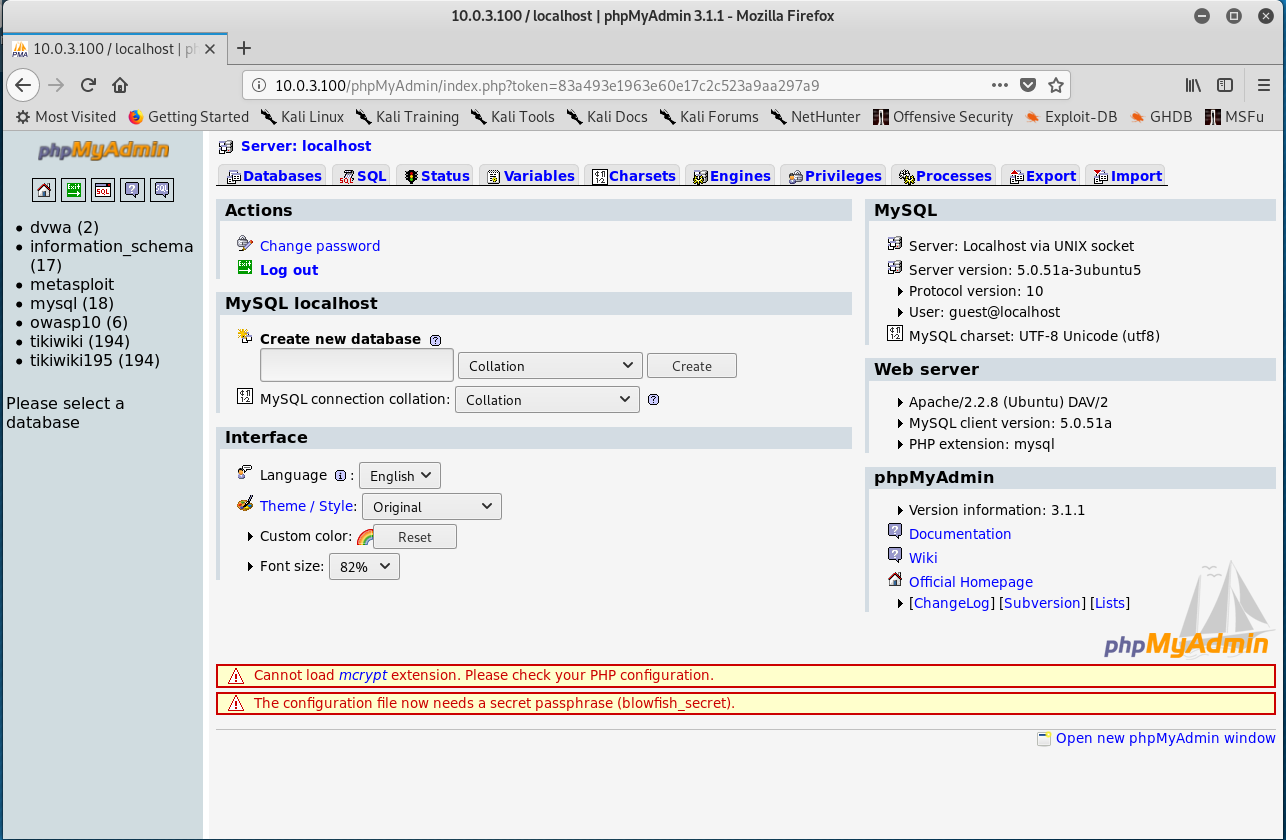
Screenshot 7: Opening the Web Browser for Target IP Address

Next, I selected phpMyAdmin:



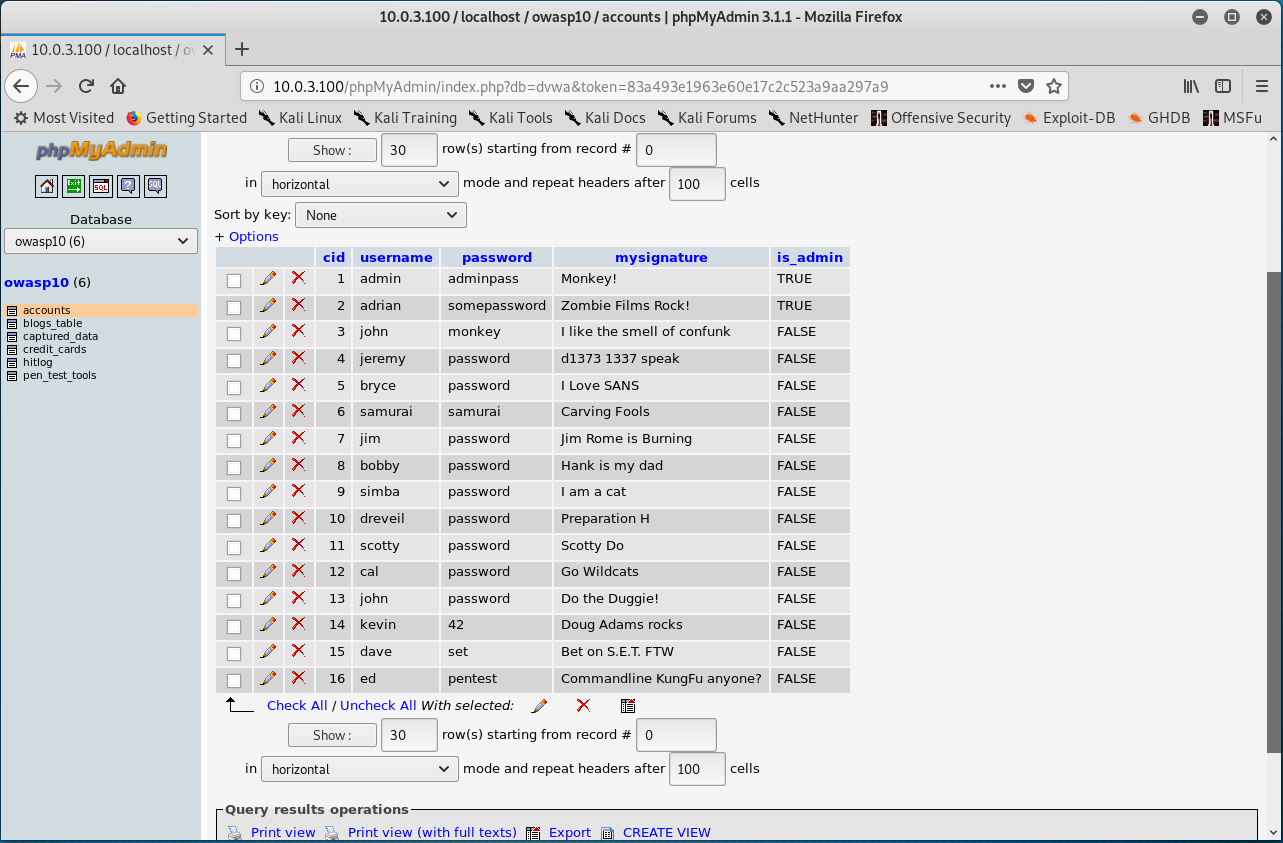
Screenshot 8: phpMyAdmin login

With our discovery of empty passwords, I should be able to type just the username to get into the database as I did in the terminal. It turns that I am not able to login to the web browser with root and an empty password. There is no issue however, because I was able to login as guest with an empty password instead.

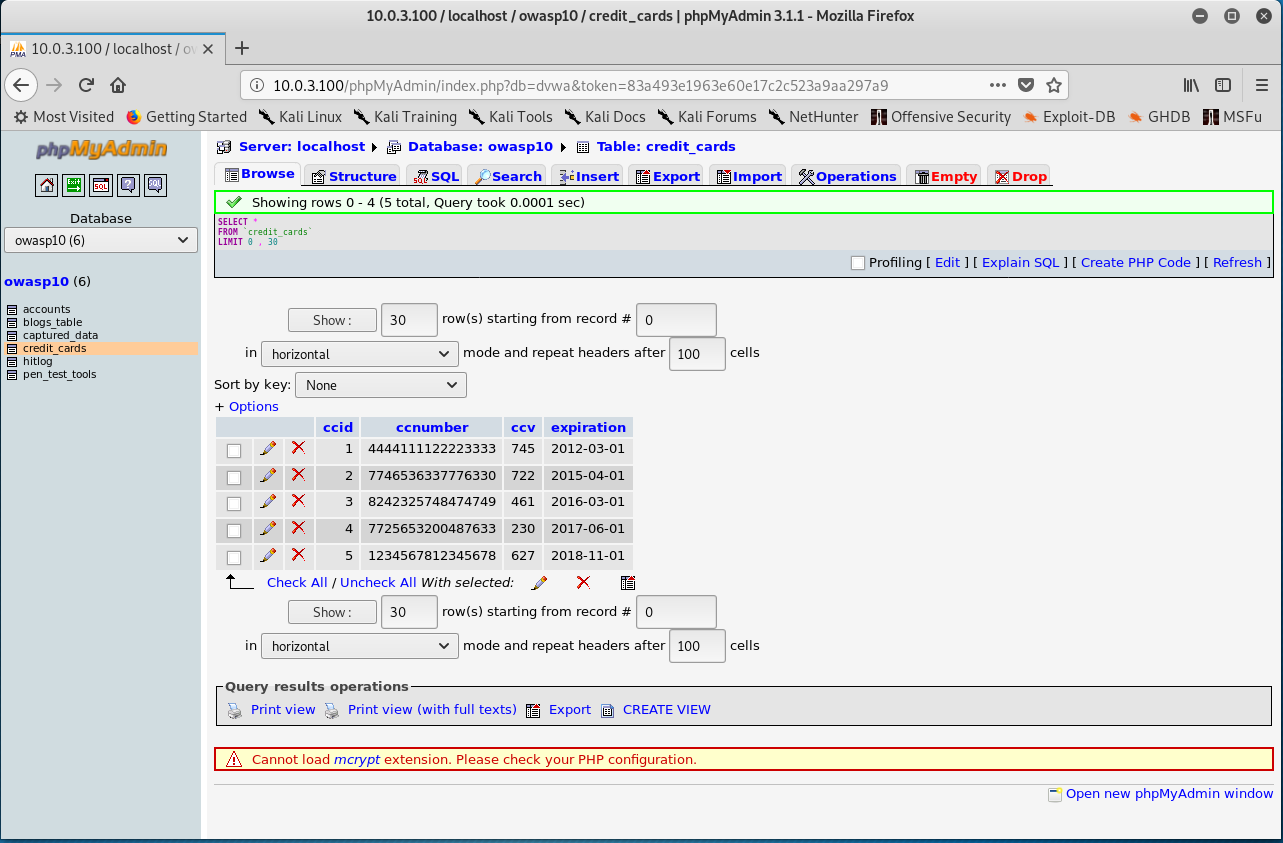


Screenshot 9: Successful “guest” Login

From here I can look at all the databases and tables as well as the information about accounts and passwords.

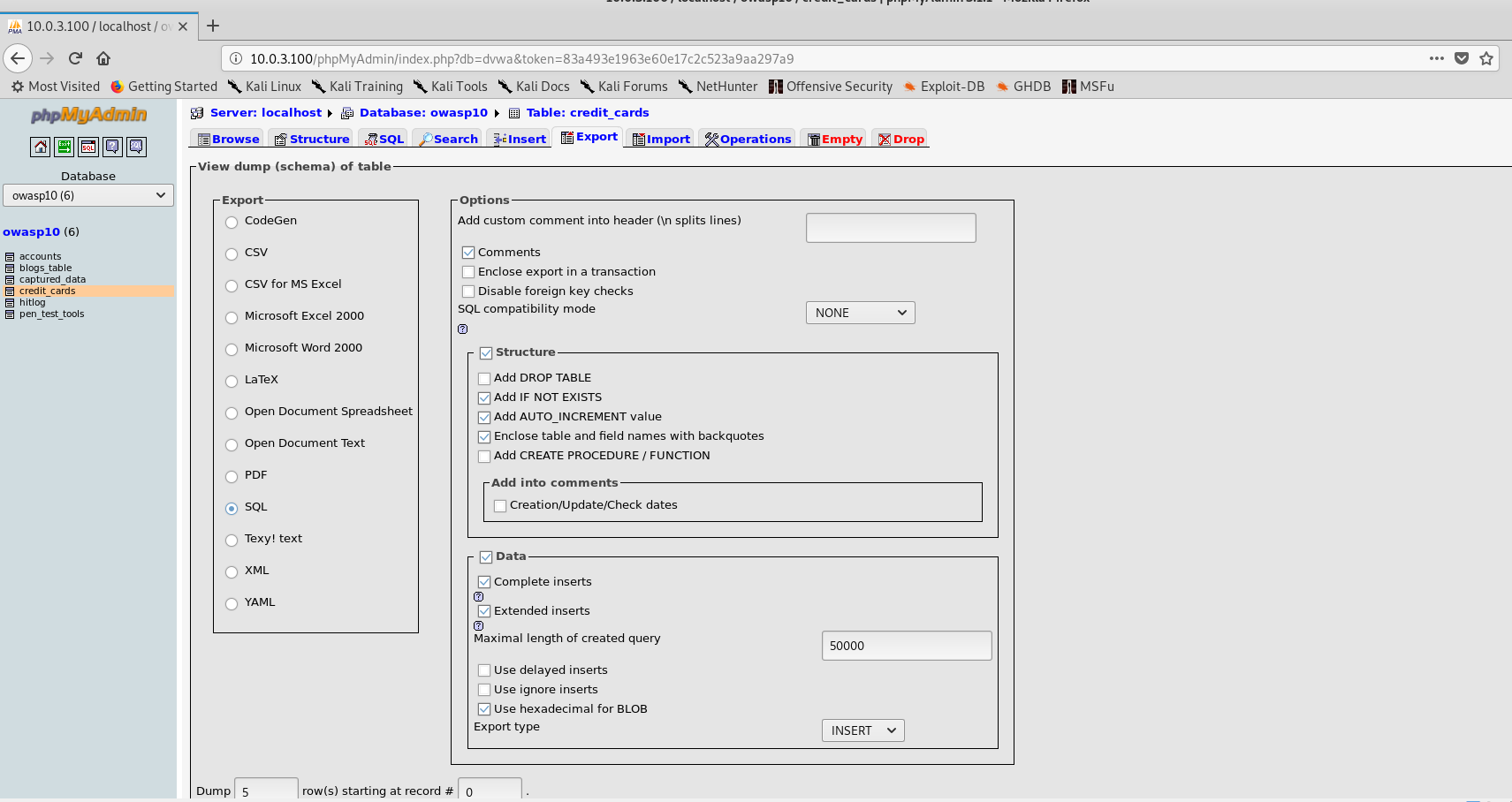


Screenshot 10: Accounts Table



Screenshot 11: Credit Card Table

All data can be exported from this database once an attacker gains access. There is an export tab at the top of the page [1]. If an attacker clicks that tab, they will have an easy way to take whatever they want in whatever format they want.



Screenshot 12: Export Tab

The implications of individuals having access to the information in this database are heavy. The cost could be thousands of dollars’ worth of losses in data and sensitive company information. With an empty password, a threat could simply login and set passwords to lock the owners of the database out of their own system.

**Recommendations**

My recommendations for Hotel Dorsey are to not have multiple accounts with escalated privileges and to ensure that each account has a password policy that effectively prevents easy login and brute force attacks. A start for password length is 8 characters [2], with a rule of thumb being to have uppercase and lowercase letters, numbers, and special characters in the password. Hotel Dorsey should implement these changes, then check to ensure that the same problems aren’t present in a week for all known usernames in the database. Members revealed in the accounts tab should also change their passwords as their credentials could have been harvested at a time prior to this test. Thank you for your attention, I hope you have a wonderful rest of your day!

## **References**

[1] Mediquant, “MySQL DB Extraction: 5 ways to extract data from a mysql database,” *MediQuant*, 21-Apr-2022. [Online]. Available: https://www.mediquant.com/mysql-db-extraction/. [Accessed: 27-Sep-2022].

[2] C. Colby and S. Profis, “Keep your passwords strong and secure with these 9 rules,” *CNET*, 08-May-2022. [Online]. Available: https://www.cnet.com/tech/mobile/keep-your-passwords-strong-and-secure-with-these-9-rules/. [Accessed: 29-Sep-2022].