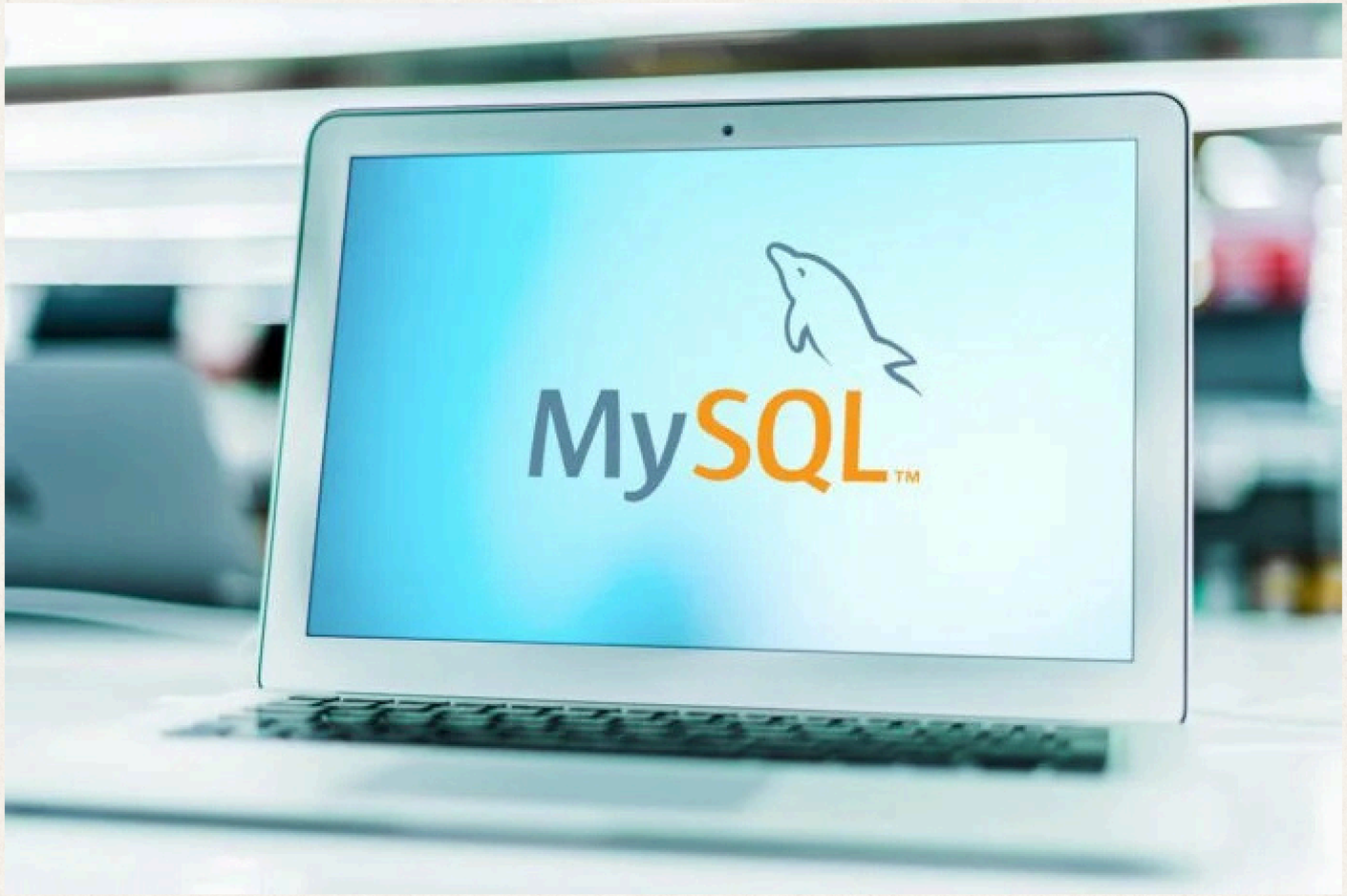
PRESENTED BY:
Taylor Dominey

ADVANCED NETWORK SYSTEMS 5100 PROJECT: APACHE AND MYSQL

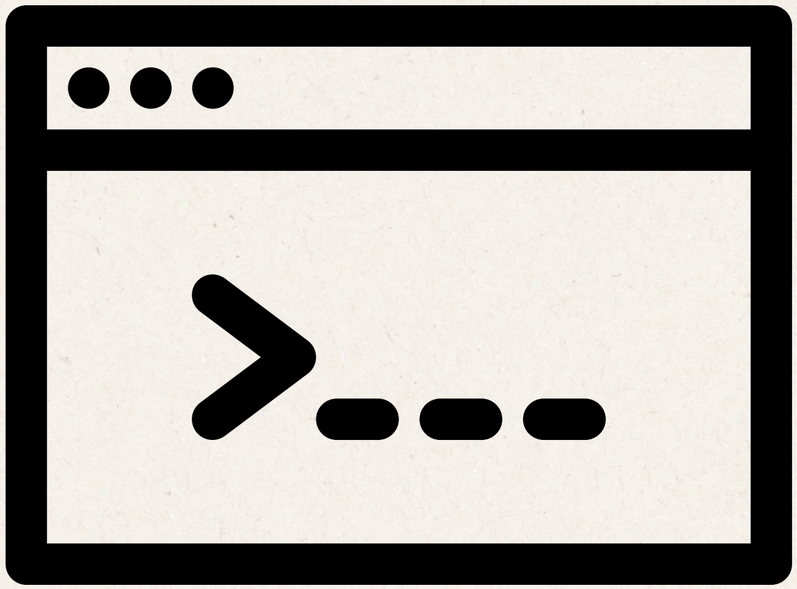


Project OverView

introduction

1. Introduction

This document provides a detailed overview of configuring Apache and MySQL for a class project. The objective of this project was to set up a secure and efficient environment, demonstrating the process of managing user permissions, securing directories, and setting up databases and users. In addition, security best practices were implemented to ensure the system's integrity, confidentiality, and availability.



2. Apache Configuration

2.1 Changing User and Group Directives

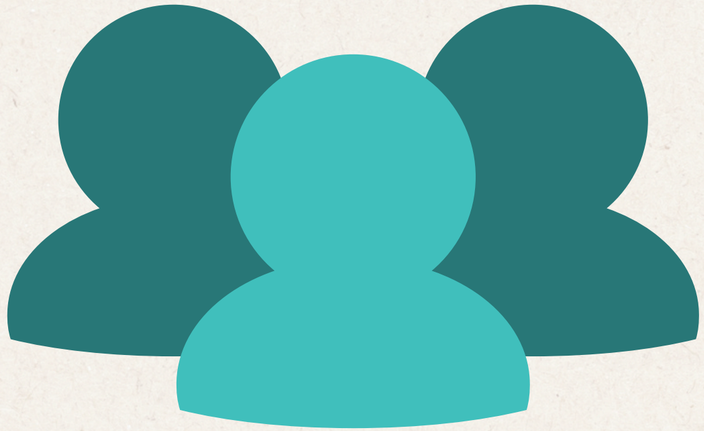
To enhance the security of the Apache server, user and group directives were modified to restrict permissions. By default, Apache runs under a generic user. Configuring specific users and groups helps control who has access to the server processes, improving security.

Steps:

1. Created a new group 'ApacheAdministrator' with Group ID (GID) 991, ensuring that only members of this group can manage the Apache process.
2. Added a user 'ApacheWebMaster' with User ID (UID) 555 and associated them with the 'ApacheAdministrator' group to limit permissions.
3. Modified the 'httpd.conf' file to set the User and Group directives as 'ApacheWebMaster' and 'ApacheAdministrator' respectively.
4. Adjusted permissions on the DocumentRoot to allow the 'ApacheWebMaster' user to read and write files, enhancing security by limiting access.

Commands:

```
# Create the group
sudo groupadd -g 991 ApacheAdministrator

# Create the user and add to the group
sudo useradd -u 555 -g ApacheAdministrator ApacheWebMaster

# Modify httpd.conf
# Change User and Group directives
sudo nano /etc/httpd/conf/httpd.conf

# Set the following values:
User ApacheWebMaster
Group ApacheAdministrator

# Ensure permissions on DocumentRoot
sudo chown -R ApacheWebMaster:ApacheAdministrator /var/www/html
sudo chmod -R 755 /var/www/html

# Restart Apache
sudo systemctl restart httpd
```



2.3 Options Directive

The 'Options' directive in Apache configuration controls the features that can be used within a directory. Different options were configured for directories to demonstrate how to control directory behaviors:
- `Indexes` allows directory listing; disabling this prevents users from seeing file names within directories.
- `FollowSymLinks` enables the server to follow symbolic links; restricting this can prevent malicious use of links.
Example Configuration:

```
<Directory "/var/www/html/dir1">
 Options Indexes FollowSymLinks
</Directory>

<Directory "/var/www/html/dir2">
 Options -Indexes +FollowSymLinks
</Directory>
```

2.4 Allow/Deny and .htaccess Configuration

To restrict access to specific directories, the 'Allow' and 'Deny' directives are used, often in combination with `.htaccess` files. This project included setting up an '/Admin' directory that should only be accessible by 'ApacheWebMaster'.
A sample `.htaccess` configuration can restrict access based on IP, or authenticated users.

3. MySQL Configuration

3.1 Database and User Creation

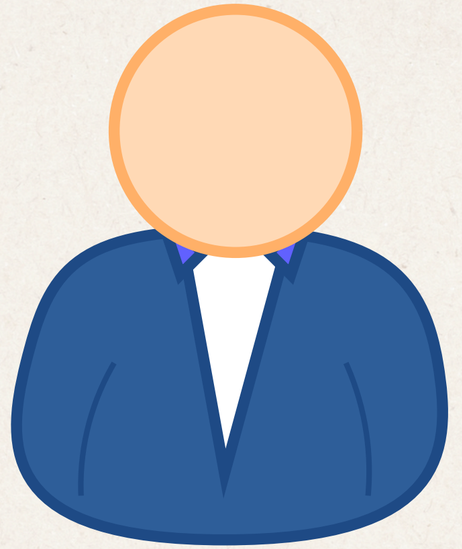
The MySQL server was configured with multiple databases and user accounts. Best practices for database user management were followed, including the use of strong passwords and restricted user privileges to minimize security risks.

Commands:

```
# Create databases
CREATE DATABASE assignment_db;
CREATE DATABASE test_user_db;

# Create users and set privileges
CREATE USER 'webAdmin'@'localhost' IDENTIFIED BY 'StrongPassword';
GRANT ALL PRIVILEGES ON assignment_db.* TO 'webAdmin'@'localhost';
```

3.2 Securing MySQL Root User

Securing the root user is a critical step in database security. The root user password was set to a strong, unique value, and remote access was disabled to prevent unauthorized access from external networks. By restricting access and applying strict permissions, the database server becomes less vulnerable to attacks.



4. Security Considerations

Security is a core aspect of server management. Some of the best practices followed during this project include:
- Regularly updating software to patch known vulnerabilities.
- Implementing firewall rules to block unauthorized access.
- Using strong, unique passwords for all user accounts.
- Regularly auditing access logs to monitor for suspicious activity.
- Run Apache as a Dedicated User:
Use a non-root, limited-privilege user (like ApacheWebMaster) to run Apache services. This minimizes the risk of system compromise if the web server is exploited.
- Use strong, complex passwords for all MySQL users, especially the root account. Enforce password policies that require complexity and expiration.

By adhering to these practices, we ensure a secure and reliable environment for the application.

5. Conclusion

This project provided a comprehensive hands-on experience in configuring and securing both Apache web servers and MySQL databases. Through the process of setting up user and group directives, I learned the importance of proper permission management to enhance system security and minimize potential vulnerabilities. By creating specific users and groups for Apache processes, I was able to restrict access and improve overall server security.

The project also required securing Apache directories, where I implemented various Options directives, such as disabling directory listings and controlling symbolic link usage. This helped me understand how to use Apache's configuration settings to limit unauthorized access and secure sensitive directories effectively. Additionally, configuring .htaccess files and using Allow and Deny directives taught me how to control access at a more granular level, reinforcing the principles of secure web server administration.

In the MySQL portion, I learned how to set up databases and users with specific privileges, highlighting the importance of giving users only the permissions they need. This is a critical practice to avoid accidental data exposure or modification. Securing the MySQL root user by setting a strong password and disabling remote access emphasized the need for careful management of administrative accounts, which can be potential entry points for attackers.

Overall, this project reinforced key best practices in server administration, including the importance of regular updates, proper user and permission management, and maintaining a proactive approach to security. These skills are essential for managing real-world IT infrastructure, and the knowledge gained through this project will be a valuable addition to my professional portfolio.

6. References

https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-22-04-with-a-lamp-stack

https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview

https://youtu.be/lD6vQBDHkqU?si=rNe5NjLhjv4HFi43

https://youtu.be/18rfWZYbS7o?si=mGgdccmkUxlce3Qb