

Copilot

Certainly! Securing **file and directory permissions** for MySQL on Ubuntu according to the **CIS (Center for Internet Security) benchmarks** is crucial. Let's ensure a robust setup:

1. Understand CIS Benchmarks:

- **CIS Benchmarks** provide industry-accepted best practices for securing IT systems.
- They offer specific configuration recommendations tailored to various platforms.
- The **CIS MySQL Benchmark** focuses on MySQL Enterprise Edition 8.0.

2. Ubuntu 22.04 CIS Benchmark:

- The **CIS Benchmark for Ubuntu 22.04** provides guidelines for securing Ubuntu systems.
- It covers file permissions, encryption, and other security controls.

3. File and Directory Permissions:

- **Principle of Least Privilege:** Assign permissions based on the principle of least privilege.
- **MySQL Data Directory:**
 - Ensure that the MySQL data directory (/var/lib/mysql by default) has appropriate permissions.
 - Restrict access to the MySQL user and group.
 - Set permissions to **750** (owner: read-write-execute, group: read-execute, others: none).
- **Log Files:**
 - Secure MySQL log files (/var/log/mysql/error.log, etc.):
 - Set permissions to **640** (owner: read-write, group: read, others: none).
 - Limit access to authorized users.
- **Backup Files:**
 - If you store backups locally, secure them:
 - Set permissions to **600** (owner: read-write, group/others: none).
 - Encrypt backups if possible.
- **Configuration Files:**
 - Protect MySQL configuration files (/etc/mysql/mysql.conf.d/mysqld.cnf):
 - Set permissions to **640** (owner: read-write, group: read, others: none).
 - Limit access to authorized users.
- **Audit Logs:**
 - Enable MySQL audit logging.
 - Set appropriate permissions for audit log files.
 - Regularly review logs for anomalies.

4. Encryption:

- **Data at Rest Encryption:**
 - Consider using **full-disk encryption** for the entire system.
 - Encrypt the MySQL data directory using tools like **LUKS**.
- **SSL/TLS Encryption:**
 - Enable SSL/TLS for secure communication between clients and the MySQL server.
 - Generate SSL certificates and configure MySQL to use them.

5. Automate with Ansible:

- Use Ansible playbooks to automate security configurations based on CIS benchmarks.

Remember to adapt these guidelines to your specific environment and business needs. Regularly review the CIS benchmarks to stay current with security requirements. By following these steps, you'll enhance the security of your MySQL installation on Ubuntu. 🔒

For more details, refer to the [CIS Benchmark for Ubuntu Linux](#) and MySQL's official documentation on [security](#). .