# Resumen - Vulnerabilidad de MySQL/MariaDB (usuario con contraseña débil)

Hallazgo del laboratorio: se creó intencionalmente un usuario de base de datos con una contraseña débil. Este resumen documenta la evidencia observada, el impacto y la conclusión solicitada por el proyecto.

## Evidencia recolectada

Durante la revisión en la VM Debian, se verificó la existencia del usuario y sus privilegios mediante consultas directas a MariaDB.

```
Comando:
  SELECT user, host, plugin FROM mysql.user;

Resultado relevante:
  - wordpressuser  | localhost | mysql_native_password
  - user           | localhost | mysql_native_password

Comando:
  SHOW GRANTS FOR 'wordpressuser'@'localhost';

Resultado relevante:
  GRANT USAGE ON *.* TO `wordpressuser`@`localhost`
  GRANT ALL PRIVILEGES ON `wordpress`.* TO `wordpressuser`@`localhost`

Comando:
  sudo ss -tulpn | grep 3306

Resultado observado:
  mariadbd escuchando en 127.0.0.1:3306 (solo localhost)
```

## Cómo esta configuración compromete la seguridad

- Permite acceso no autorizado por adivinación/fuerza bruta si las credenciales son simples o reutilizadas.
- Una vez autenticado, el atacante puede leer y modificar datos sensibles (usuarios, correos, hashes, configuraciones de la aplicación).
- En este caso, el usuario wordpressuser tiene ALL PRIVILEGES sobre wordpress.*, lo que facilita la toma de control de WordPress desde la base de datos (por ejemplo, cambiar roles o credenciales).
- Aunque el servicio escuche en localhost (127.0.0.1), si un atacante obtiene acceso al sistema (p. ej., vía SSH o una falla web), puede conectarse localmente y explotar estas credenciales.

## Riesgo (resumen)

**Severidad:** Alta. La combinación de credencial débil + privilegios amplios (ALL PRIVILEGES) incrementa el impacto. El alcance remoto directo es menor porque la BD escucha en localhost, pero el riesgo sigue siendo significativo ante una intrusión inicial.

## Recomendaciones de corrección

- Rotar la contraseña del usuario débil por una contraseña fuerte y única.
- Aplicar el principio de mínimo privilegio: otorgar solo los permisos necesarios sobre la base de datos (evitar ALL PRIVILEGES si no es imprescindible).
- Mantener el alcance de conexión restringido a localhost y evitar cuentas con host='%' salvo necesidad real.
- Proteger credenciales en la aplicación (wp-config.php): permisos restrictivos y bloqueo de acceso directo desde Apache.
- Monitorear y auditar: registrar intentos de autenticación y cambios en usuarios/permisos.

## Conclusión (respuesta directa)

La configuración es insegura porque permite que un atacante, al obtener o adivinar la contraseña del usuario wordpressuser, tenga control completo sobre la base de datos wordpress debido a ALL PRIVILEGES. Esto habilita exfiltración y manipulación de datos y puede derivar en compromiso total de WordPress. La exposición remota de la BD es limitada por estar en localhost, pero el riesgo permanece alto ante un acceso inicial al servidor.