Auditoría y Fortalecimiento de la Seguridad en Bases de Datos de Codearts Solutions

🔹 Fase 1: Análisis de vulnerabilidades en la base de datos

```
# Habilitar logs generales y slow query en MySQL

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

# Añadir o modificar:

general_log = ON

general_log_file = /var/log/mysql/mysql.log

slow_query_log = ON

slow_query_log_file = /var/log/mysql/mysql-slow.log

long_query_time = 2

# Reiniciar MySQL

sudo systemctl restart mysql


# Ver usuarios con privilegios en MySQL

mysql -u root -p -e "SELECT user, host, authentication_string FROM mysql.user;"


# Para PostgreSQL, revisar configuración de logging en postgresql.conf

sudo nano /etc/postgresql/14/main/postgresql.conf

# Ajustar:

logging_collector = on

log_directory = 'pg_log'

log_filename = 'postgresql-%Y-%m-%d.log'

log_statement = 'all'

# Reiniciar PostgreSQL

sudo systemctl restart postgresql
```

Usar SQLmap para detectar inyección SQL

sqlmap -u "http://targetsite.com/page.php?id=1" --batch --risk=3 --level=5

Comprobar cifrado TLS entre cliente y servidor

openssl s_client -connect dbserver:3306 -starttls mysql

🔹 Fase 2: Implementación de medidas de seguridad

Crear roles y asignar permisos mínimos en MySQL

mysql -u root -p

CREATE USER 'appuser'@'%' IDENTIFIED BY 'securepassword';

GRANT SELECT, INSERT, UPDATE ON dbname.* TO 'appuser'@'%';

FLUSH PRIVILEGES;

Habilitar cifrado en tránsito (MySQL)

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Añadir:

ssl-ca=/etc/mysql/certs/ca.pem

ssl-cert=/etc/mysql/certs/server-cert.pem

ssl-key=/etc/mysql/certs/server-key.pem

Reiniciar MySQL

sudo systemctl restart mysql

Restringir acceso remoto en MySQL (en my.cnf)

bind-address = 127.0.0.1

En PostgreSQL, editar pg_hba.conf para restringir IPs

sudo nano /etc/postgresql/14/main/pg_hba.conf

Añadir línea para permitir sólo IPs autorizadas, ejemplo:

host all all 192.168.1.0/24 md5

Reiniciar PostgreSQL

sudo systemctl restart postgresql

Configurar firewall UFW para permitir solo IPs específicas

sudo ufw allow from 192.168.1.0/24 to any port 3306

sudo ufw deny 3306

Automatizar copia de seguridad cifrada con cron y OpenSSL

echo "0 2 * * * mysqldump -u root -p'STRONG_PASS' dbname | openssl enc -aes-256-cbc -out /backups/db_backup_\$(date +\%F).sql.enc -k 'backupkey'" | sudo tee -a /etc/crontab

---

🔹 Fase 3: Pruebas de seguridad y simulación de ataques

Realizar prueba de inyección SQL controlada (usar entorno de pruebas)

sqlmap -u "http://testapp.local/login.php?user=admin&pass=1234" --batch --risk=3 --level=5

Simular ataque DoS enviando consultas pesadas (PostgreSQL)

psql -U postgres -d dbname -c "SELECT pg_sleep(10);" &

Monitorizar intentos de acceso fallidos (MySQL)

sudo tail -f /var/log/mysql/mysql.log | grep "Access denied"

En PostgreSQL

sudo tail -f /var/log/postgresql/postgresql-*.log | grep "FATAL"

🔹 Fase 4: Monitoreo y auditoría de accesos

```
# Activar logs avanzados y rotación en MySQL (logrotate ya instalado)
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
# Confirmar general_log y slow_query_log activados (ver Fase 1)

# Configurar alertas en OSSEC o Wazuh para MySQL (ejemplo)
# En el agente Wazuh, añadir reglas para monitorizar logs MySQL y PostgreSQL
# Consultar documentación Wazuh para configuración detallada

# Monitorear logs en tiempo real con alertas básicas
sudo tail -f /var/log/mysql/mysql.log | grep --line-buffered "Access denied" | while read line; do
  echo "Alerta de acceso denegado: $line" | mail -s "Alerta DB" admin@codearts.com
done
```