SQL 安全手册

一、基础安全架构设计

1. 最小权限原则

账户分级：

系统管理员（sysadmin）：仅用于安装、备份等核心操作

数据库管理员（db_owner）：负责数据库架构管理

应用用户：通过角色（如 db_datareader/db_datawriter）分配最小权限

示例：使用 SQL Server 的 CREATE ROLE 创建特定业务角色 10

权限审计：

查询高权限账户

```
SELECT name, type_desc, create_date
FROM sys.server_principals
WHERE is_disabled = 0 AND (type = 'S' OR type = 'U')
```

2. 身份验证强化

多因素认证（MFA）：

对管理员账户强制启用 Windows Hello / 硬件令牌

应用层集成 OAuth 2.0/OpenID Connect

密码策略：

复杂度：8 位以上，包含大小写字母、数字、特殊字符

有效期：90 天强制更换

失败锁定：5 次尝试后锁定账户 30 分钟

MySQL 设置密码策略

```
SET GLOBAL validate_password.policy=1;
SET GLOBAL validate_password.length=10;
```

3. 网络层防护

防火墙规则：

限制数据库端口（如 MySQL 3306、SQL Server 1433）仅允许应用服务器 IP 访问

禁用 ICMP 协议防止探测

VPN 通道：

对远程管理启用 IPsec VPN 或 SSL VPN

示例：使用 OpenVPN 建立专用隧道

二、核心防御技术

1. SQL 注入防护

参数化查询：

所有用户输入必须通过预编译语句传递

python

Python 示例

```
cursor.execute("SELECT * FROM users WHERE id = %s", (user_id,))
```

避免拼接 SQL 字符串：

python

错误示例

```
cursor.execute("SELECT * FROM users WHERE id = " + user_id)
```

输入验证：

使用白名单验证输入格式（如正则表达式匹配邮箱）

过滤特殊字符：\ ' " ; -- #

MySQL 过滤函数

```
CREATE FUNCTION sanitize_input(input_str TEXT)
```

```
RETURNS TEXT DETERMINISTIC
```

```
BEGIN
```

```
    RETURN REPLACE(REPLACE(input_str, '\', '\\\''), '"', '\"');
```

```
END
```

2. 数据加密体系

静态加密：

使用透明数据加密（TDE）：

SQL Server 启用 TDE

```
USE master;
```

```
CREATE DATABASE ENCRYPTION KEY
```

```
WITH ALGORITHM = AES_256
```

```
ENCRYPTION BY SERVER CERTIFICATE TDECert;
```

```
ALTER DATABASE MyDB SET ENCRYPTION ON;
```

对敏感字段（如密码）使用 Always Encrypted

传输加密:

强制启用 SSL/TLS 连接:

```
[mysqld]
```

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

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

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

```
require_secure_transport=ON
```

3. 漏洞扫描与补丁管理

定期扫描:

使用 Nessus/Qualys 扫描数据库漏洞

重点检测 CVE-2023-23583（MySQL 身份验证绕过）等高危漏洞

补丁策略:

建立测试环境验证补丁兼容性

实施季度补丁更新周期

示例: Oracle Critical Patch Update (CPU) 每季度发布

三、高级安全实践

1. 审计与日志分析

审计策略:

记录所有权限变更、高危操作（如 ALTER DATABASE）

PostgreSQL 启用审计

```
CREATE EXTENSION pgaudit;
```

```
SET pgaudit.log = 'ddl,dml';
```

日志存储:

日志文件存储于只读介质

配置 Splunk/ELK 进行实时分析

```
json
```

```
{
```

```
  "index": "database_audit",
```

```
"type": "sql",
"fields": {
  "user": "admin",
  "action": "CREATE TABLE",
  "timestamp": "2025-08-01T12:00:00Z"
}
}
```

2. 容灾与备份安全

备份策略:

全量备份每周一次，增量备份每日两次

使用 AES-256 加密备份文件

PostgreSQL 备份命令

```
pg_dump -U postgres -d mydb -f backup.sql --encrypt-password
```

恢复测试:

每季度进行恢复演练

验证备份数据的完整性

3. 入侵检测与响应

异常检测:

监控登录失败率、慢查询等指标

使用机器学习模型识别异常行为

应急响应:

建立 72 小时响应机制

隔离受攻击数据库并进行取证分析

四、合规性要求

1. 中国法规遵循

网络安全法:

重要数据本地存储

三级等保要求日志留存 6 个月

数据安全法:

数据分类分级管理

跨境数据传输需通过安全评估

2. 国际标准参考

ISO 27001:

建立信息安全管理体

定期进行风险评估

GDPR:

个人数据加密存储

数据泄露 72 小时内报告

五、各数据库专项指南

1. MySQL 安全增强

禁用危险函数:

禁用 LOAD_FILE

```
SET GLOBAL local_infile=0;
```

限制并发连接:

```
[mysqld]
```

```
max_connections=500
```

2. SQL Server 深度防护

启用动态数据掩码:

对信用卡字段进行掩码处理

```
ALTER TABLE Customers
```

```
ADD CONSTRAINT mask_cc CHECK (CreditCard LIKE '****_****_****-%');
```

配置行级安全性:

```
CREATE SECURITY POLICY RLS_Policy
```

```
ADD FILTER PREDICATE dbo.fn_securitypredicate(Department)
```

```
ON dbo.Employees
```

```
WITH (STATE = ON);
```

3. Oracle 加固要点

撤销默认权限:

```
REVOKE CREATE PUBLIC SYNONYM FROM PUBLIC;
```

启用细粒度审计:

```
BEGIN
```

```
  DBMS_FGA.ADD_POLICY(  
    object_schema => 'HR',  
    object_name => 'EMPLOYEES',  
    policy_name => 'EMPLOYEES_AUDIT',  
    audit_condition => 'USER = ''ANONYMOUS''',  
    statement_types => 'SELECT'  
  );
```

```
END;
```

六、实施路线图

1. 阶段 1 (1-3 个月):

完成账户权限审计与密码策略部署

启用传输加密

2. 阶段 2 (4-6 个月):

实施参数化查询改造

部署入侵检测系统

3. 阶段 3 (7-12 个月):

完成静态数据加密

通过等保三级测评

七、工具推荐

类别	工具名称	功能特点
漏洞扫描	IBM AppScan	深度检测 SQL 注入等漏洞
日志分析	Elastic Stack	实时监控与威胁检测
加密管理	AWS KMS	密钥全生命周期管理

类别	工具名称	功能特点
权限管理	CyberArk Conjur	特权账户安全托管

通过实施上述体系化安全措施，可将数据库攻击面降低 80% 以上。