 DOI: https://doi.org/10.12349/iser.v7i3.9476

Research on Identification Countermeasures of Database Privacy Data Leakage in Software Development

Yanhui Li

Shanghai Shanda College, Shanghai, 201209, China

Abstract

As digital transformation continues to deepen, software databases contain increasingly extensive information. The leakage of personal data and critical information within these databases poses significant threats to individual rights. In this context, it is essential to establish comprehensive privacy data leakage identification and prevention strategies throughout the entire software development lifecycle. Effective application of technologies such as sensitive data auto-identification, abnormal access and operational behavior detection, data flow and transmission leakage analysis, and security vulnerability identification can enhance privacy data protection capabilities. Building on this foundation, implementing core control measures during critical phases—including development, testing, deployment, and operations—will strengthen the overall protection of privacy data.

Keywords

software development; database; privacy data breach; identification and prevention

软件开发数据库隐私数据泄露识别对策研究

李彦会

上海杉达学院，中国 · 上海 201209

摘　要

在数字化转型持续深化的背景下，软件数据库中囊括的信息变得越来越多，数据库中的个人信息和重要数据一经泄露则很容易会对个人权益造成较大的影响和威胁。在这样的背景下立足软件开发全过程明确隐私数据泄露识别与防控策略是十分必要的。可通过敏感数据自动识别技术、异常访问与操作行为识别技术、数据流向与传输泄露识别技术、安全漏洞与脆弱性识别技术等相应技术的有效应用提高隐私数据泄露识别能力。在此基础之上紧抓开发阶段、测试阶段、上线部署阶段、运维阶段等各个阶段明确核心管控措施，提高隐私数据的保护能力。

关键词

软件开发；数据库；隐私数据泄露；识别与防范

1 引言

现阶段软件应用范围在不断扩大，在政务、金融、医疗、民生等相应领域都有所应用。而在软件应用中数据库作为数据存储与管理的核心载体，容纳了大量数据，这其中也包含个人信息或关乎公众利益的重要数据，一经泄露所带来的影响和损失是不容忽视的，因此必须加强数据库隐私数据泄露识别，做好隐私数据的保护，可从如下几点着手提高隐私数据泄露识别能力。

2 数据库隐私数据泄露识别技术

2.1 敏感数据自动识别技术

想要更好的识别隐私数据泄露问题，首要基础和重中之重则是做好敏感数据的识别，只有这样才可以为后续的监测与防护奠定良好的基础和保障。可根据《中华人民共和国个人信息保护法》将个人信息划分为一般个人信息和敏感个人信息，敏感个人信息又包含宗教信仰、生物识别、特定身份、医疗健康、金融账户、行踪轨迹等等。针对该类隐私数据可采用结构化字段匹配、语义识别、规则引擎结合的方式进行数据库信息扫描完成公开数据、内部数据、敏感数据、核心机密数据的识别，并生成统一的敏感数据资产清单，提高敏感数据识别能力。此外，为了更好的保障系统运行的流畅性，可遵循最小必要原则标记与业务相关的隐私数据。

2.2 异常访问与操作行为识别技术

异常访问与操作行为识别技术可以及时的发现泄露风险，其原理是通过对数据库的全流程访问，及时的识别违背正常策略的行为。在用户登录应用软件时该项技术可实时监测用户的身份、IP 地址、操作时间、SQL 语句类型、数据访问量、数据导出频次等相应的数据信息，构建正常行为基

【作者简介】李彦会（1988–），女，中国广东深圳人，硕士，助教，从事质量管理，软件开发技术研究。

线，若用户的行为偏离基线则会自动预警。例如可以精准识别非工作时间批量查阅敏感数据、越权访问未授权表、高频次小批量导出数据、执行高危 SQL 语句、境外 IP 访问核心数据库等等。可通过日志采集、协议解析、行为分析模型的构建提高异常访问与操作行为的识别能力，及时发现异常行为，保障异常行为识别的准确性。

2.3 数据流向与传输泄露识别技术

在隐私数据泄露识别上需注意数据跨节点传输和流转是泄露的高发节点，因此必须抓住这一关键要点进行流量监测和数据血缘追踪，确保隐私数据全链路可见、可控。可借助数据流向与传输识别技术精准识别数据流出行为，有效监测敏感数据从生产库流向测试库、开发环境、外部系统、终端设备的全过程。若在检测的过程中发现未授权数据流转行为，可采用密码技术配合网络边界防护设备及时阻断违规流向，保障数据安全。

2.4 安全漏洞与脆弱性识别技术

在软件开发阶段很有可能会出现配置缺陷、代码漏洞等相应问题，这些问题也会导致隐私数据泄露。为更好的规避这类问题则需要借助安全漏洞与脆弱性识别技术，通过技术识别及时发现 SQL 注入漏洞、权限配置漏洞、加密实现缺陷、日志审计缺失、默认口令未修改等相应问题，保障隐私数据安全。在具体落实上可借助静态代码检测、动态安全测试、配置基线核查实现开发测试阶段的漏洞识别，生成漏洞清单，并为相关工作人员提供修复建议，严格按照国家标准要求进行漏洞审核，从源头上降低隐私数据泄露风险 [1]。

不同技术的核心功能和识别对象是存在鲜明差异的，需结合隐私数据泄露识别需求来进行技术选择和技术应用，如表 1 所示。

表 1：数据库隐私数据泄露识别核心技术

技术维度	核心功能	识别对象
敏感数据识别	分类分级标记	隐私数据字段
异常行为识别	实时监测预警	违规访问操作
数据流向识别	全链路追踪	跨域数据传输
漏洞识别	脆弱点检测	代码配置缺陷

3 软件开发全流程隐私数据泄露管控对策

软件开发是一项系统性工作，需要根据不同阶段存在的数据泄露隐患针对性的落实管控工作，明确管控目标和核心管控措施，如表 2 所示。

表 2：不同阶段的管控目标及核心管控措施

流程阶段	核心管控目标	关键措施
开发阶段	源头风险防控	代码审核、数据隔离
测试阶段	测试数据安全	脱敏处理、权限管控
上线阶段	配置安全加固	权限最小化、加密审计
运维阶段	持续安全运行	实时监测、应急处置

3.1 开发阶段

开发阶段是隐私数据安全管控的重心，可以从源头上减少隐私数据泄露的问题。而在开发阶段相关工作人员首先需要从需求分析、架构设计、代码编写等各个环节出发，将安全要求嵌入到不同环节当中。例如在需求分析阶段需要通过数据分析整合明确应用软件在投入使用以后隐私数据的收集范围、使用目的、存储期限，坚持最小必要与知情同意原则，在满足软件应用的实际需求同时最大化的减少不必要或与业务无关的个人信息。而在架构设计阶段则可通过数据库敏感字段加密、访问权限隔离、数据脱敏规则的设计保护隐私数据。在代码编写阶段应严格按照相应规范要求来编写 SQL 语句，并且通过漏洞检测及时的发现问题，配合各种现代化技术的有效应用实现敏感数据访问日志自动记录，并禁止硬编码数据库凭证与敏感信息。还可通过代码安全审核机制的构建进行隐私风险核查，若核查未通过则不得进入下一环节 [2]。

3.2 测试阶段

测试阶段是验证软件功能与安全性的关键环节，可以及时的发现隐私泄露风险，有效规避数据滥用、越权访问、违规留存、随意拷贝等相应行为。在该阶段需确保测试过程不触碰、不扩散、不遗留真实隐私数据，确保测试数据的生成与使用符合于规范要求，坚决杜绝在测试阶段，从生产库抽取未经处理的真实个人信息，用于压力测试、接口测试和功能测试。进入测试环境的敏感数据可通过标识化、加密、掩码替换等相应脱敏处理方法，在确保数据信息仍旧保留业务逻辑有效性的同时确保隐私数据处理后无法识别特定自然人。此外，在测试的过程中可通过访问控制机制的优化和调整确保测试人员仅可访问相关数据，禁止越权访问非相关数据库表、敏感字段及后台配置信息，避免在测试阶段出现数据泄露的情况。同时可借助区块链技术等相应现代化技术将测试操作等相应数据信息统一整合并进行审计。为确保测试工作落实的规范性、专业性，有效发现漏洞的同时避免隐私信息的丢失，可签订数据安全与保密协议，在协议中明确相关工作人员禁止私自拷贝、截图、传输、存储测试环境中的隐私数据，不可将隐私数据应用于测试以外的任何用途。在测试结束以后还需按照规范标准对临时文件、数据、日志、缓存信息等相应数据信息进行全面且彻底的清理，避免出现隐私数据残留的情况。此外，在测试的过程中还需同步落实数据安全专项监测，从敏感数据识别、访问控制、加密脱敏、异常阻断等安全机制出发，分析该类安全机制是否能够发挥其应有的作用和影响。若发现存在隐私数据泄露隐患应及时反馈给相关工作人员并落实整改工作，确保系统上线前其安全能力达标 [3]。

3.3 上线部署阶段

上线部署阶段是软件系统从开发测试环境转向生产运

行环境的关键节点，在该阶段进行安全防控也是十分必要的，否则则很容易会出现配置疏漏、权限泛滥、加密失效、审计不全等相应问题，进而使隐私数据处于开放暴露的状态。而在上线部署阶段可通过强化安全基线、收紧访问权限、启动密码保护、完善审计机制等多种方式确保隐私数据安全。这就需要相关工作人员根据国家安全标准与数据安全法规的相关条例确定安全配置基线，做好身份鉴别、访问控制、协议配置、服务端口、日志记录等相应配置的调整，并且关闭不必要的存储过程、远程调用功能以及管理端口，同时需要消除宽松权限、默认账号、弱口令等潜在的安全隐患。在生产库权限配置的过程中需要遵循最小权限原则，即根据不同岗位工作人员的工作内容、工作职责来确定授权权限，可以根据软件应用对象将其划分为运维、管理、业务、开发等不同人员群体，然后针对性的确定系统权限。对于敏感数据则可以采用加密储存机制，结合国家密码管理要求确定加密算法，确保隐私数据在静止状态下并不会被明文读取。配合数据库审计功能有效识别、监测登录行为、查询操作、修改操作、删除操作、批量导出操作等相应的操作信息，将信息登记并共享至云端平台。在上线前还需要通过安全评估和配置核查，分析隐私数据防护措施是否到位，是否建立了漏洞闭环管理机制，权限设置是否合理等等，坚决杜绝未经过安全核查即上线运行 [4]。

3.4 运维阶段

数据库的开发周期相较于运营周期是相对较短的，而在数据库运行期间也很有可能会因外部攻击、内部违规、误操作等相应因素的影响出现隐私数据泄露，这时则需要通过日常运维、运行监测等相应工作的落实来确保隐私数据安全。在日常运维中应以安全巡检、漏洞修复、权限审计、策略优化为重点，定期落实数据库配置、账号权限、连接状态、存储空间、运行日志的检查，若在检查中发现漏洞和问题应及时进行整改。同时在系统运行的过程中可能会涉及到业务变更、架构调整、系统升级等相应问题，这时需要第一时间对系统进行检测和分析，判断是否存在新风险。在运行监测方面可借助数据库隐私数据泄露识别技术及时发现异常访问、高频查询、批量导出、境外 IP 访问等相应问题，并自动预警，提高泄露风险的发现、响应、干预和处置能力。此外，在软件系统运行期间还需要做好数据备份，确保在发生数据损坏、丢失、非法篡改等相应恶性事件以后可以及时恢复数据，避免影响后续各项业务及工作的顺利推进和正常开展。

4 结语

软件开发数据库隐私泄露识别工作的有效落实是保障个人权益的重要前提，必须引起关注和重视。可通过敏感数据自动识别技术、异常访问与操作行为识别技术、数据流向与传输泄露识别技术、安全漏洞与脆弱性识别技术等相应现代化技术的有效应用提高数据库隐私数据泄露识别的能力。在此基础之上还需要从开发阶段、测试阶段、上线部署阶段、运维阶段出发明确不同阶段的隐私数据泄露管控要点，避免隐私数据丢失、泄露等相应情况的出现。

参考文献

[1] 李世博. 基于人工智能的软件开发数据库隐私数据泄露识别技术 [J]. 信息与电脑, 2025, 37 (07): 35-37.

[2] 伍帅. 软件开发数据库隐私数据泄露识别技术研究 [J]. 信息与电脑, 2025, 37 (05): 118-120.

[3] 张杰. 油田开发数据库隐私数据泄露风险识别方法 [J]. 无线互联科技, 2025, 22 (03): 25-28.

[4] 于平. 基于概率主题模型的软件开发数据库隐私数据泄露识别方法研究 [J]. 河北软件职业技术学院学报, 2024, 26 (02): 19-23.