项目申请书

项目名称: openGauss权限扫描

项目主导师: 崔永泉 yqcui1977@hust.edu.cn

申请人: 陈贤文

日期: 2023年5月9日

邮箱: mcxw@hust.edu.cn

1.项目背景

openGauss是一种高性能、高可靠、高安全性的开源数据库系统，由华为公司发起和维护。在实际应用中，由于复杂的配置和管理需求，openGauss数据库可能存在各种违规操作和安全隐患。因此，需要根据openGauss的数据库配置和用户权限，检查数据库中是否存在违规操作的可能和安全隐患。

技术要求：

1. 了解openGauss的基础功能 2. 了解访问控制

项目产出：

1. 探索openGauss中的访问控制模型，完成技术洞察博客一篇。
2. 设计用户权限检查项，开发扫描程序，实现数据库权限扫描功能，扫描数据库中是否有违规操作的可能和安全隐患，完成设计文档。

2.技术方法及可行性

该扫描程序将基于Java语言和openGauss JDBC驱动程序实现，使用Spring Boot框架构建。

- Java语言和JDBC API：用于与openGauss数据库交互。
- Spring Boot框架：用于构建和管理Web应用程序。

对JDBC开发有项目经历，曾基于Spring Boot框架，Mybatis和Spring JDBC开发小程序，用Thymeleaf模板引擎，设计web应用。

3.项目实现细节梳理

3.1 探索openGauss中的访问控制模型

openGauss是一种开源的、关系型数据库管理系统，它提供了多种安全机制来控制用户对数据库资源的访问。在openGauss中，访问控制模型包括认证、授权和审计等功能。本文将探索openGauss中的访问控制模型，并编写实现细节。

认证

在openGauss中，认证是指验证用户的身份和凭据。openGauss支持多种认证方式，包括密码认证、Kerberos认证和证书认证等。以下是密码认证的实现细节：

1. 用户输入用户名和密码。
2. 系统根据用户名从pg_shadow表中获取该用户的加密口令。
3. 系统通过加密算法将用户输入的密码转换为加密口令。
4. 系统比较两个口令，如果相同，则认证通过。

授权

在openGauss中，授权是指授予用户对数据库资源的权限。openGauss支持多级授权，包括数据库级别、模式级别、表级别和列级别。以下是表级别授权的实现细节：

1. 系统管理员使用GRANT命令授权给普通用户，例如：GRANT SELECT, UPDATE ON table_name TO user_name;
2. 系统管理员可以使用REVOKE命令撤销已有的授权，例如：REVOKE SELECT, UPDATE ON table_name FROM user_name;
3. 用户可以使用SHOW GRANTS命令查看自己的权限，例如：SHOW GRANTS FOR user_name;
4. 管理员可以使用ALTER DEFAULT PRIVILEGES命令设置默认权限，例如：ALTER DEFAULT PRIVILEGES IN SCHEMA schema_name GRANT SELECT, UPDATE ON TABLES TO role_name;

审计

在openGauss中，审计是指记录用户对数据库资源的访问信息。openGauss支持多种审计方式，包括日志审计和审计策略等。以下是日志审计的实现细节：

1. 管理员使用ALTER SYSTEM SET参数来配置审计参数，例如：ALTER SYSTEM SET audit_trail = 'DB';
2. 系统管理员使用CREATE AUDIT POLICY命令创建审计策略，例如：CREATE AUDIT POLICY policy_name ACTIONS ALL;
3. 系统管理员使用ALTER USER命令启用或禁用用户的审计功能，例如：ALTER USER user_name AUDIT ALL;
4. 系统管理员使用SELECT * FROM pg_audit_all_tables()命令查看所有表的审计日志。

以上是openGauss中访问控制模型的一些实现细节，这些机制可以使得openGauss更加安全和可靠。

3.2 用户权限检查

设计用户权限检查项，开发扫描程序，实现数据库权限扫描功能，扫描数据库中是否有违规操作的可能和安全隐患，完成设计文档。

在openGauss数据库中，存在各种类型的用户和角色，并且每个用户/角色都有不同的权限和访问级别。为了确保数据库安全性，我们需要开发一个程序来检查数据库中的违规操作和安全隐患。以下是我们将采取的实现细节：

设计用户权限检查项

为了检查数据库中是否存在违规操作和安全隐患，我们需要设计一系列用户权限检查项。以下是一些示例检查项：

- 是否存在default账户？
- 是否存在弱口令？
- 是否存在超级用户？
- 是否允许远程连接？
- 是否启用密码策略？
- 是否设置了正确的SSL证书？
- 是否过期的账号和密码？
- 是否关闭了未使用的服务？
- 是否启用了审计功能？

开发扫描程序

为了实现openGauss数据库权限扫描功能，我们将使用Java编写扫描程序。以下是主要步骤：

1. 读取配置文件：从配置文件中读取数据库连接信息和需要扫描的检查项。
2. 连接到数据库：使用JDBC连接到openGauss数据库。
3. 执行查询语句：执行相应的SQL查询或命令，如SELECT、CREATE等。
4. 解析结果：解析查询结果，并确定数据库是否存在违规行为。
5. 输出结果：将扫描结果输出到日志文件或控制台。

实现openGauss数据库权限扫描功能

为了实现openGauss数据库权限扫描功能，我们将采用以下技术：

- JDBC API：用于与数据库交互。
- Spring Boot框架：用于构建和管理Web应用程序。
- Thymeleaf模板引擎：用于生成报告和分析。
- Bootstrap框架：用于设计和排版Web界面。

以下是大致步骤：

1. 用户在Web界面上选择需要扫描的检查项。
2. 程序读取配置文件中的数据库连接信息，并使用JDBC连接到openGauss数据库。
3. 程序根据用户选择的检查项执行相应的查询语句，如SELECT、CREATE等。
4. 程序解析查询结果，并确定数据库是否存在违规行为。
5. 程序将扫描结果输出到Web界面上，并提供详细的报告和分析。

设计文档

为了记录openGauss数据库权限扫描功能的设计和实现细节，我们需要编写设计文档。该文档应包括以下内容：

- 项目背景和目标；
- 技术架构和实现细节；
- 用户权限检查项和扫描程序设计；
- 数据库权限扫描功能实现方法；
- 代码示例和测试结果；
- 总结和未来工作。

通过编写设计文档，我们可以更好地管理和维护openGauss数据库权限扫描功能，并为后续的开发工作提供参考。

4.规划

1.项目研发第一阶段

（07 月 01 日 - 08 月01 日）：

在这个阶段，需要探索openGauss中的访问控制模型，并撰写技术洞察博客。以下是实现细节：

- 阅读openGauss官方文档，了解openGauss的基本架构和特性。
- 探索openGauss中的访问控制模型，包括用户、角色和权限等组件。
- 实践openGauss，在虚拟机或云环境中安装并配置openGauss。
- 撰写技术洞察博客，介绍openGauss的访问控制模型和基本使用方法。

2.项目研发第二阶段

（08 月 02日 - 08 月 30 日）：

在这个阶段，将设计和开发openGauss数据库权限扫描程序，以便检查数据库中的违规操作和安全隐患。以下是实现细节：

- 设计用户权限检查项，确定需要检查的数据库资源和操作。
- 使用Java编写扫描程序，连接到openGauss数据库并执行相应的查询语句。
- 解析查询结果，确定是否存在违规行为和安全隐患。
- 输出扫描结果到日志文件或Web界面。
- 编写设计文档，记录openGauss数据库权限扫描功能的实现细节和测试结果。

3.项目测试和部署

（08 月 30日 - 09 月 30 日）：

在这个阶段，将对openGauss数据库权限扫描功能进行测试，并部署到生产环境中。以下是实现细节：

- 编写单元测试，确保程序能够正确执行和输出必要的结果。
- 进行系统测试，检查是否存在任何问题和异常情况。
- 部署程序到生产环境，确保其可以在实际运行中正常工作。
- 提供用户培训和支持，以便他们能够使用和理解开发的程序。
- 编写用户文档，介绍openGauss数据库权限扫描功能的基本用法和注意事项。