KINGBASE™
人大金仓

KingbaseES V008R003

安全指南

北京人大金仓信息技术股份有限公司

约定

以下文本约定适用于本文档：

- 中括号（[和]）表示包含一个或多个可选项。不需要输入中括号本身。
- 花括号（{和}）表示包含两个以上（含两个）的候选，必须在其中选取一个。不需要输入花括号本身。
- | 为分割中括号或者花括号中的两个或两个以上选项。不需要输入“|”本身。
- 点（. . .）表示其之前的元素可以被重复。

安全指南

欢迎来到KingbaseES V008R003（以下简称KingbaseES V8R3）教程。本教程从安装与更新、SQL和PL/SQL、管理、开发、性能、安全和高可用七部分对KingbaseES V8R3数据库进行全方位的介绍。

本部分主要内容为KingbaseES V8R3 安全指南。数据安全是信息安全的基础环节。为应对纷繁复杂的多样化数据安全保护需求，金仓数据库KingbaseES 完全遵照安全数据库国家标准GB/T 20273-2006的结构化保护级(第四级)技术要求进行了完整系统的安全功能研制。通过全新的结构化系统设计和强化的多样化强制访问控制模型框架，金仓数据库KingbaseES 在身份鉴别、用户权限，以及数据访问、存储和传输等方面的安全增强提高了数据库系统的整体安全性，提供了包括强化身份鉴别、自主访问控制、安全标记、强制访问控制、特权分立、资源限制、客体重用，以及程序运行和数据存储完整性、数据存储透明加密、数据传输加密等在内的主要安全功能和控制手段，可以从容应对复杂多样的安全业务场景，保障敏感数据的安全。

在国产数据库厂家中，它率先通过公安部计算机信息系统安全产品质量监督检验中心的强制性安全认证，并获得销售许可证。

KingbaseES通过全新的结构化系统设计和强化的多样化强制访问控制模型框架，自主开发了多个高等级的安全特性，并完整实现包括多重身份鉴别、入侵检测与报警、可信路径、推理控制、形式化证明及隐蔽信道分析等在内的全部结构化保护级的技术和功能要求。

目录

Part I. 快速入门

数据安全是信息安全的基础环节。为应对纷繁复杂的多样化数据安全保护需求，KingbaseES V8R3完全遵照安全数据库国家标准GB/T 20273-2006结构化保护级(第四级)技术要求进行了完整系统的安全功能研制。金仓数据库在身份鉴别、用户权限，以及数据访问、存储和传输等方面的安全增强提高了数据库系统的整体安全性。

目录

KingbaseES V8R3安全策略

前言

在数据库安全方面，国外数据库厂商在我国销售的数据库产品的安全级别只能达到TCSEC C1或C2级，个别产品虽在C2安全级别基础上增加了部分如强制访问控制、数据加密等高等级的安全特性，但其整体安全级别均不能达到B1级或更高，从而不能很好满足如军队、军工、政府、金融行业、保险行业及电信行业等涉密部门的高安全要求。因此，研制具有自主产权的高等级数据库安全产品非常必要。

KingbaseES自主研发高等级的数据库安全产品，它完全符合国家安全数据库标准GB/T 20273-2006的结构化保护级(即第四级)的技术要求，该级别近似等同于TCSEC B2级。在国产数据库厂家中，它率先通过公安部计算机信息系统安全产品质量监督检验中心的强制性安全认证，并获得销售许可证。

KingbaseES通过全新的结构化系统设计和强化的多样化强制访问控制模型框架，自主开发了多个高等级的安全特性，并完整实现包括特权分立、身份鉴别、多样化访问控制、用户数据保护、审计等在内的全部结构化保护级的技术和功能要求。

1 特权分立

KingbaseES 采用了三权分立的安全管理体制，数据库三权分立是为了解决数据库超级用户权力过度集中的问题，参照行政、立法、司法三权分立的原则来设计的安全管理机制。KingbaseES把数据库管理员分为数据库管理员、安全管理员、审计管理员三类。

- 数据库管理员

  主要负责执行数据库日常管理的各种操作和自主存取控制。

- 安全管理员

  主要负责强制存取控制规则的制定和管理。

- 审计管理员

  主要负责数据库的审计，监督前两类用户的操作。

这三类用户是相互制约又相互协作共同完成数据库的管理工作。安全管理员可以授权用户查看某些敏感数据（强制存取控制授权），但是并不意味着这个用户就可以看到这些敏感数据，它还需要得到数据库管理员的授权（自主存取控制授权）。同理，如果只有数据库管理员的自主存取控制授权而没有安全管理员的强制存取控制授权，用户还是无法看到它不应当看到的敏感数据。审

计管理员拥有一套机制，可以保护审计记录数据不会被数据库管理员或者安全管理员删除或者篡改。

这三类用户彼此隔离，互不包容，各自维护自己权限许可范围内的对象，不能跨范围操作，也不能相互授权。数据库管理员不能对安全、审计相关的用户及数据库对象进行操作，不能将任何用户修改为安全员或审计员，不能授予、回收安全员、审计员的权限，不能切换到安全员、审计员的许可认证；安全员只能管理安全员和安全相关的系统对象，同理，审计员只能管理审计员和审计相关的系统对象。

三权分立堵住了以前滥用数据库超级用户特权的安全漏洞，进一步提高了数据库的整体安全性。

2 身份鉴别

KingbaseES支持基于强化口令的身份鉴别，它包括对数据库用户施加口令复杂度检查、帐户和口令有效期限设置、帐户锁定等安全策略管理等机制。此外，KingbaseES还支持基于Kerberos、Radius、LDAP认证协议和CA等技术在内的与第三方身份认证产品相结合的外部统一身份鉴别或集中化身份认证方式。

通过强化身份鉴别与SSL安全传输相结合，KingbaseES建立了服务器与客户端的可信路径安全通道，实现了双向可信认证，杜绝假冒用户身份或者假冒数据库服务器的恶意行为，并进一步防范这些恶意行为可能引发的重放攻击。

3 多样化访问控制

KingbaseES采用多种控制手段确保用户访问数据的合法性和安全性，并有效防止非法用户的未授权访问。

这些访问控制技术具体如下：

3.1. 自主访问控制

KingbaseES采用ACL(存取控制列表)技术实现了用户对于自有表和列(字段)的自主访问控制。用户可自主授权其他用户和角色对自有数据对象上的操作权限。当一个用户访问某个数据表/列时，自主访问控制将依据 ACL检查该用户对访问对象的访问权限是否合法，用以决定接受还是拒绝本次用户的访问行为。

对用户和角色的存取权限的定义称为授权，这主要通过GRANT和REVOKE语句来实现。可授予用户的权限包括系统权限和对象权限，而对象权限包括表级权限和列级权限等。

3.2. 强制访问控制

KingbaseES按照Bell-La Padula安全保密模型，并基于实用性和灵活性的考虑对强制存取模型策略进行了扩充。所谓强制访问控制是指主体用户对数据库客体对象访问的安全策略是由系统强制实施的，任何用户都无法突破系统定义的强制安全策略而非法访问未授权的数据库客体对象。与自主访问控制不同的是，强制访问控制由安全管理员管理而不是数据对象属主，并它直接在受控主客体上施加安全标记，而不再从ACL中获取权限信息。

除了支持简单保密性原则—“从下读、平行写”外，KingbaseES还可以支持“区间写”。为了支持一些特殊的管理性工作，系统还提供了特权功能。拥有特权的主体能绕过相应的标记仲裁规则。

3.3. 推理控制

在数据库系统中，访问控制限制了用户对敏感信息的访问，即只允许用户访问授权信息，而禁止用户访问未授权信息。这种方式虽然防止了敏感信息的直接泄露，但却无法防止这些信息的间接泄露，例如：

假设某个公司信息系统中，姓名和职务属于授权信息，而工资属于未授权信息。那么，用户A有权访问姓名和职务信息，而无权访问工资信息。此外，还假设用户B职务和A相同，存在函数依赖职务（授权）->工资（未授权），A的工资是1000元（假设每个用户都知道自己的工资）， 则用户A可推出B的工资也是1000元。上述这类问题无法通过传统的访问控制方式解决，需借助推理控制技术。

KingbaseES实现静态函数依赖、用户已知敏感信息定义、动态函数依赖、动态敏感关联，禁止用户推导出自己无权访问的高安全等级信息，保证数据的安全性。

4 用户数据保护

用户数据是数据库系统的核心资源，其安全性对整个系统至关重要。通常，用户数据中包含具有保密要求的重要信息，如商业秘密信息。一旦这些信息被窃取或篡改，将给用户造成极大损失，甚至带来灾难性的后果。因此，保障用户数据安全是非常必要的。

KingbaseES为用户数据提供多方位的保护，有效杜绝对恶意用户对存储数据的非法窃取，及外部破坏和篡改，这些保障措施具体如下：

4.1. 数据存储完整性保护

通过在每个数据块头增加"数据水印"，KingbaseES实现了数据存储过程中的完整性校验和保护。每次读磁盘时，自动进行数据水印校验。每次写磁盘时，自动更新数据水印。这样，可以有效杜绝来自数据库外操作系统层的非法篡改。整个过程对用户完全透明。

4.2. 数据备份安全保护

备份数据是否安全可靠一直是数据库安全中容易被忽视的技术点和管理点。针对这个问题，KingbaseES提供带密钥的加密备份和还原，同时施加完整性数据水印，且支持备份压缩，可充分保证了数据备份和恢复的安全性和可信性。

4.3. 数据透明加密存储

KingbaseES实现了数据在写到磁盘上时对其进行加密，当授权用户重新读取数据时再对其进行解密。无需对应用程序进行修改，授权用户甚至不会注意到数据已经在存储介质上加密，加密解密过程对用户都是透明的。

5 审计

KingbaseES实现了审计功能记录用户的行为，主要实现了服务器实例级别审计、语句级别审计、模式对象级别审计、细粒度级别审计。

Part II. KingbaseES 安全细则

目录

用户管理

1. 用户定义

KingbaseES中，除了在系统初始化过程中可以创建的三个用户：数据库管理员、安全管理员、审计管理员外，还可以创建普通用户来访问数据库，运行数据库应用。

2. 三权分立

2.1. 概述

KingbaseES V8安全版本支持将管理特权三权分立为三个管理员，并在初始化的时候创建数据库管理员、安全管理员和审计管理员。三权分立的安全管理体制是为了解决数据库超级用户权力过度集中的问题，参照行政、立法、司法三权分立的原则来设计的安全管理机制。

KingbaseES中，由于三权分立的约束，数据库管理员，安全管理员，审计管理员各自维护自己权限许可范围内的用户，不同目的的用户应由相应的管理员创建。即：数据库管理员用户不能创建和修改安全员和审计员，也不能将一个普通用户修改为安全员或者审计员，安全管理员只能创建和修改安全员并且不能将安全员修改为非安全员，审计管理员只能创建和修改审计员并且不能将审计员修改为非审计员。

2.2. 特权用户

a. 系统管理员（SYSTEM)

   主要负责执行数据库日常管理的各种操作和自主存取控制。

b. 安全管理员（SYSSSO）

   主要负责强制存取控制规则的制定和管理，且不能创建和操作普通对象。

c. 审计管理员（SYSSAO）

   主要负责数据库的审计，监督前两类用户的操作。且不能创建和操作普通对象。

2.3. 与其他模块关系

a. 审计

   i. 系统管理员

系统管理员不可以修改审计参数。系统管理员不可以审计规则。系统管理员不可以查看审计记录。

ii. 安全管理员

安全员只可以设置及删除对审计员和普通用户的审计规则。安全员只可以查看普通用户和审计员的审计记录。

iii. 审计管理员

只可以由审计员开启审计开关GUC参数。审计员可以设置及删除对安全员和系统管理员的审计规则。审计员可以查看安全员和系统管理员的审计记录。

b. 安全

i. 系统管理员

不可以进行安全功能的操作。

ii. 安全管理员

只可以由安全员开启安全开关GUC参数。只可以由安全员设置及删除标记，策略等安全功能。

iii. 审计管理员

不可以进行安全功能的操作。

2.4. 相关参数

a. separate_user

```
separate_user 为安全员参数，只能由安全员进行修改。
ON:
1）SYSTEM创建用户不能带with语句。（with password，with superuser等）
2）只有SYSSSO用户可以修改除SYSTEM和SYSSAO以外的用户，普通用户本身自
己只可以修改自己的密码
3）SYSSSO/SYSSAO/SYSTEM只可以修改自己除createrole/nocreaterole,
createdb/nocreatedb,superuser/nosuperuser之外的属性
4）只有SYSSSO可以重命名用户名,但是不可以修改SYSTEM/SYSSAO。
OFF:
1）separate_user参数默认为OFF
2）SYSTEM正常操作除SYSSSO和SYSSAO以外的用户
3）SYSSAO/SYSSSO只可以修改自己的密码
普通用户分两种情况：
1）无createrole权限，只可以修改自己的密码。
2）有createrole权限，可以修改一些更多的属性，例如createdb,connection limit
注意：
1） 使用三权分立必须打开separate_user选项。
2） 在安全版中separate_user参数默认值为ON。
```

2.5. 相关限制

a. CREATE

SYSSSO和SYSSAO不可以创建用户及普通对象

b. ALTER USER/ROLE

详情见separate_user参数控制

c. DROP USER/ROLE

依赖于是否有CREATEROLE的权限，并且SAO和SSO不可以被删除。

d. ALTER … OWNER TO

SYSTEM和普通用户可以执行此命令，但目标用户不可以是SSO和SAO。
SAO和SSO不可以执行此命令。

e. REASSIGN OWNED

只有SYSTEM可以执行此命令，但目标用户不可以是SSO和SAO。

f. DROP OWNED

系统管理员和普通用户可以执行此命令，但目标不允许是SSO和SAO。
SAO和SSO不可以执行此命令。

g. SET SESSION AUTHORIZATION

系统管理员可以切换到除SAO和SSO以外系统管理员角色与其他用户。
审计员、安全员和普通用户会话的切换目标只能是自己本身。
执行主体指的是登录时的用户，并不是当前用户。

h. ALTER SYSTEM SET

系统管理员不可以修改审计和安全的参数
审计员只可以修改审计参数
安全员只可以修改安全参数
普通用户不可以修改

i. ALTER DATABASE SET

系统管理员和数据库属主可以执行此命令。
SSO和SAO不可以执行。

j. SET

```
系统管理员不允许修改审计和安全相关的参数
审计员只可以修改普通用户可配置参数
安全员只可以修改普通用户可配置参数
普通用户只可以修改普通用户可配置参数
安全参数和审计参数改为了系统级别，所以不能使用SET进行设置。
```

3. 用户访问控制

KingbaseES 中，管理员可以配置用户有效期，限制账号的使用期限，还可以设定密码规则，设置用户密码最大连续错误次数。

身份鉴别

KingbaseES支持基于强化口令的身份鉴别，包括对数据库用户的口令复杂度进行检查、帐户口令的有效期限设置、帐户异常登录锁定、账户登录信息显示等安全策略的管理机制。

1. 口令的复杂度管理

口令的复杂度检查是由数据库管理员对口令的最小长度，所包含的数字、英文字母、特殊符号的数目进行设置后，在创建和修改用户时，自动对口令进行相关方面的检查。如果口令不满足指定的条件，那么创建用户将不成功。

KingbaseES通过插件的方式来进行口令的复杂度管理。这种方式更为灵活，当数据库的实用场景需要进行口令的复杂度管理时，加载插件即可。而不需要该功能时，卸载插件即可。

KingbaseES中通过 4 个全局级参数配合插件来实现用户口令复杂度管理。

1.1. 加载插件：

修改 kingbase.conf 文件中 `shared_preload_libraries` 参数后重启数据库。

```
shared_preload_libraries = 'passwordcheck'
```

1.2. 参数配置：

`passwordcheck.password_length`

口令的最小长度，取值范围为[8，63]，缺省为 8。

```
SHOW passwordcheck.password_length;
 passwordcheck.password_length
-------------------------------
 8
(1 row)

SET passwordcheck.password_length = 10;
SHOW passwordcheck.password_length;
 passwordcheck.password_length
-------------------------------
 10
(1 row)
```

```
CREATE USER u_pwd PASSWORD '123ab';
ERROR:  password length 5 is too short, should be longer than min
password length 10.

CREATE USER u_pwd PASSWORD '1234567890ab';
CREATE ROLE
```

passwordcheck.password_condition_letter

口令至少包含几个字母，取值范围为[2，61]，缺省为 2。

```
SHOW passwordcheck.password_condition_letter;
 passwordcheck.password_condition_letter
-----------------------------------------
 2
(1 row)

SET passwordcheck.password_condition_letter = 3;
SHOW passwordcheck.password_condition_letter;
 passwordcheck.password_condition_letter
-----------------------------------------
 3
(1 row)

ALTER USER u_pwd PASSWORD '1234567890';
ERROR:  Password should contain at least 3 letter and the current
number is 0

ALTER USER u_pwd PASSWORD '1234567890ab';
ERROR:  Password should contain at least 3 letter and the current
number is 2

ALTER USER u_pwd PASSWORD '1234567890abC';
ALTER ROLE
```

passwordcheck.password_condition_digit

口令至少包含几个数字，取值范围为[2，61]，缺省为 2。

```
SHOW passwordcheck.password_condition_digit;
 passwordcheck.password_condition_digit
----------------------------------------
 2
(1 row)

SET passwordcheck.password_condition_digit = 3;
SHOW passwordcheck.password_condition_digit;
```

```
 passwordcheck.password_condition_digit
----------------------------------------
 3
(1 row)

ALTER USER u_pwd PASSWORD 'abcdefghij';
ERROR:  Password should contain at least 3 digit and the current
number is 0

ALTER USER u_pwd PASSWORD 'abcdefghij123';
ALTER ROLE
```

`passwordcheck.password_condition_punct`

口令至少包含几个特殊字符，取值范围为[0，59]，缺省为 0。其中特殊符号为除空白符、英文字母、单引号和数字外的所有可见字符。

```
SHOW passwordcheck.password_condition_punct;
 passwordcheck.password_condition_punct
----------------------------------------
 0
(1 row)

SET passwordcheck.password_condition_punct = 2;
SHOW passwordcheck.password_condition_punct;
 passwordcheck.password_condition_punct
----------------------------------------
 2
(1 row)

ALTER USER u_pwd PASSWORD '1234567890abc.';
ERROR:  Password should contain at least 2 punct and the current
number is 1

ALTER USER u_pwd PASSWORD '1234567890abc./';
ALTER ROLE
```

1.3. 卸载插件：

修改 kingbase.conf 文件中 `shared_preload_libraries` 参数后重启数据库。

```
shared_preload_libraries = ''
```

2. 口令有效期设置

KingbaseES的用户管理中含有口令有效期这一属性，当用户的口令过期后，用户仍能正常登录，但会提示用户密码过期，如用户不修改密码，则不能进行任何其它操作。一旦修改密码后，不需要退出即可正常执行操作。

2.1. 参数配置：

`password_change_interval`

口令的更换周期，单位是天，0表示无限制，取值范围为[0，2147483647]，缺省为 0。

> **注意：**安全版中该参数取值范围为[1,7] 。

指定密码创建用户时，可通过 password expire 选项指定该用户的口令更换周期，指定的口令更换周期必须晚于当前时间且早于更换周期 `password_change_interval` 指定的时间。

对已创建成功且已拥有密码的用户，也可通过alter语句的 password expire 选项修改其口令更换周期，但仅安全管理员有这个权限，其它用户无法修改自己及他人的口令更换周期。

若在创建用户或修改用户密码时未显示的通过password expire选项指定该用户的密码有效期，那么系统会根据 `password_change_interval`参数设定的值自动为其计算密码有效期。

```
SHOW password_change_interval;
 password_change_interval
--------------------------
 0
(1 row)

SET password_change_interval to 5;
SHOW password_change_interval;
 password_change_interval
--------------------------
 5
(1 row)

CALL now();
              NOW
--------------------------------
 2019-06-27 18:17:51.045933+08
(1 row)

CREATE USER u_pwd_et PASSWORD EXPIRE '2019-06-28';
ERROR:  option "PASSWORD EXPIRE" can not be specified if "PASSWORD"
is not specified

CREATE USER u_pwd_et PASSWORD '1234567890abC/.' PASSWORD EXPIRE '
```

```
2019-06-28';

SELECT USENAME, PWDEXPIRETIME FROM SYS_USER WHERE USENAME =
'U_PWD_ET';

 USENAME   |      PWDEXPIRETIME
-----------+-------------------------
 U_PWD_ET | 2019-06-28 00:00:00+08
(1 row)

ALTER USER u_pwd_et PASSWORD '/.1234567890abC';
SELECT USENAME, PWDEXPIRETIME FROM SYS_USER WHERE USENAME =
'U_PWD_ET';

 USENAME   |          PWDEXPIRETIME
-----------+--------------------------------
 U_PWD_ET | 2019-07-02 18:22:03.801357+08
(1 row)
```

3. 帐户异常登录锁定

帐户异常登录锁定是指如果用户连续若干次不能正确的登录数据库，那么这个用户的帐户将被系统禁用。系统允许的用户连续错误登录次数由数据库管理员指定。被禁用的帐户可以由安全员利用 SQL 命令使其重新可用或者等待一段时间自动解锁。

3.1. 参数配置：

`error_user_connect_times`

允许的连续错误最大次数，超过则封锁，0 表示无限制，取值范围为[0，2147483647]，缺省为 0。

> **注意：** 安全版中该参数取值范围为[1,5] 。

设置密码连续最大失败次数为 6。

```
ALTER SYSTEM SET error_user_connect_times = 6;
CALL sys_reload_conf();
```

`error_user_connect_interval`

被封锁用户的自动解锁时间，单位是分钟，超过时间间隔自动解除用户封锁，0 表示关闭自动解锁功能，取值范围为[0，2147483647]，缺省为 30。

设置被封锁用户的自动解封时间为 1 小时。

```
ALTER SYSTEM SET error_user_connect_interval = 60;
CALL sys_reload_conf();
```

3.2. 解除锁定：

超过时间间隔自动解除用户封锁。

安全员可通过enable user解除对用户的封锁。

```
ENABLE USER username;
```

3.3. 相关接口：

系统表 sys_audit_blocklog：

当封锁用户的功能被打开，即系统参数 `error_user_connect_times` 的值大于0时，可通过系统表 sys_audit_blocklog 查看用户被封锁的信息；该系统表记录从上一次成功登录开始到现在的登录密码错误次数信息，如果没有登录密码错误的情况，则在该系统表中不存在对应用户的记录。

查看用户USER1的封锁信息

```
select * from sys_audit_blocklog where user = 'USER1';
```

4. 账户登录信息显示

账户登录信息显示则会在用户登录数据库时给出用户一些提示信息，如本次成功登录的用户名、IP和登录时间，上次成功登录的用户名、IP和登录时间，两次成功登录之间的失败登陆次数和最近两次的登录失败记录以及口令到期时间等信息。

4.1. 参数配置：

`user_logonlog_level`

数据库对用户登录信息记录的级别，取值范围为[0，2]，缺省为0。

0 表示不记录任何用户的登录信息；

1 表示只记录用户成功登录的信息；

2 表示既记录用户登录成功的信息，也记录用户登录失败的信息。

设置既记录用户成功登陆信息又记录失败登录信息。

```
ALTER SYSTEM SET user_logonlog_level = 2;
CALL sys_reload_conf();
```

4.2. 相关接口：

系统表 sys_audit_userlog：

用户的登录信息保存在系统表sys_audit_userlog中，管理员可查看所有用户的登录信息；普通用户可通过系统视图sys_user_audit_userlog查看本用户的登录信息。

管理员查看最近登录的10条信息。

```
select * from sys_audit_userlog order by audtimestamp desc limit 10;
```

系统函数 sys_del_user_logonlog_before_days：

安全员可删除指定天数之前的登录信息。

安全员删除30天以前的用户登录信息。

```
SELECT sys_del_user_logonlog_before_days(30);
```

系统函数 sys_del_user_logonlog_by_ip：

安全员可删除指定IP地址的用户登录信息。

安全员删除IP为127.0.0.1的用户登录信息。

```
SELECT sys_del_user_logonlog_by_ip('127.0.0.1');
```

系统函数 sys_del_user_logonlog_by_name：

安全员可删除指定用户的登录信息。

安全员删除用户 USER1 的登录信息。

```
SELECT sys_del_user_logonlog_by_name('USER1');
```

以 SYSTEM 身份使用 ksql 连接数据库的登录信息显示示例：

```
ksql -USYSTEM -W123 TEST
ksql (V008R003)
Type "help" for help.

This time logon information:
            User Name: SYSTEM
                 Host: [local]
           Logon Time: 2019-06-27 18:33:52.371487+08

The last time successful logon information:
            User Name: SYSTEM
                 Host: [local]
```

```
           Logon Time: 2019-06-27 18:28:11.841052+08

The fail times between this logon and the last logon: 3

This latest [1] attempt to logon information:
            User Name: SYSTEM
                 Host: [local]
           Logon Time: 2019-06-27 18:33:48.677542+08

This latest [2] attempt to logon information:
            User Name: SYSTEM
                 Host: [local]
           Logon Time: 2019-06-27 18:33:48.677542+08

The password's expire time: 2019-07-02 18:28:01.558175+08

The number of days from the password expires: 4 days 23:54:09

TEST=#
```

客户端认证

当一个客户端应用连接一个数据库服务器时，它将指定以哪个KingbaseES 数据库用户名连接，就像我们以一个特定用户登录一台 Unix 计算机一样。在 SQL 环境中，活动的数据库用户名决定对数据库对象的访问权限。因此，它本质上是哪些数据库用户可以连接。

> **注意：**如上所释，KingbaseES实际上以"角色"来进行权限管理。在本章中，用数据库用户表示"拥有LOGIN权限的角色"。

认证是数据库服务器建立客户端身份的过程，并且服务器决定客户端应用（或者运行客户端应用的用户）是否被允许以请求的数据库用户名来连接。

KingbaseES提供多种不同的客户端认证方式。被用来认证一个特定客户端连接的方法可以基于（客户端）主机地址、数据库和用户来选择。

KingbaseES数据库用户名在逻辑上是和服务器运行的操作系统中的用户名相互独立的。如果一个特定服务器的所有用户在那台服务器的机器上也有帐号，那么分配与操作系统用户名匹配的数据库用户名是有意义的。不过，一个接受远程连接的服务器可能有许多没有本地操作系统帐号的用户，并且在这种情况下数据库用户和操作系统用户名之间不必有任何联系。

1. sys_hba.conf文件

客户端认证是由一个配置文件（通常名为sys_hba.conf并被存放在数据库集簇目录中）控制（HBA表示基于主机的认证）。在initdb初始化数据目录时，它会安装一个默认的sys_hba.conf文件。

sys_hba.conf文件的常用格式是一组记录，每行一条。空白行将被忽略，#注释字符后面的任何文本也被忽略。记录不能跨行。一条记录由若干用空格和/或制表符分隔的域组成。如果域值用双引号包围，那么它可以包含空白。在数据库、用户或地址域中引用一个关键字（例如，all或replication）将使该词失去其特殊含义，并且只是匹配一个有该名字的数据库、用户或主机。

每条记录指定一种连接类型、一个客户端 IP 地址范围（如果和连接类型相关）、一个数据库名、一个用户名以及对匹配这些参数的连接使用的认证方法。第一条匹配连接类型、客户端地址、连接请求的数据库和用户名的记录将被用于执行认证。这个过程没有"落空"或者"后备"的说法：如果选择了一条记录而且认证失败，那么将不再考虑后面的记录。如果没有匹配的记录，那么访问将被拒绝。

记录可以是下面七种格式之一：

```
local      database  user  auth-method  [auth-options]
host       database  user  address  auth-method  [auth-options]
hostssl    database  user  address  auth-method  [auth-options]
hostnossl  database  user  address  auth-method  [auth-options]
host       database  user  IP-address  IP-mask  auth-method
[auth-options]
hostssl    database  user  IP-address  IP-mask  auth-method
[auth-options]
hostnossl  database  user  IP-address  IP-mask  auth-method
[auth-options]
```

各个域的含义如下：

`local`

这条记录匹配企图使用 Unix 域套接字的连接。如果没有这种类型的记录，就不允许 Unix 域套接字连接。

`host`

这条记录匹配企图使用 TCP/IP 建立的连接。`host`记录匹配SSL和非SSL的连接尝试。

> **注意：**除非服务器带着合适的listen_addresses配置参数值启动，否则将不可能进行远程的 TCP/IP 连接，因为默认的行为是只监听在本地环回地址`localhost`上的 TCP/IP 连接。

`hostssl`

这条记录匹配企图使用 TCP/IP 建立的连接，但必须是使用SSL加密的连接。

要使用这个选项，编译服务器的时候必须打开SSL支持。此外，在服务器启动的时候必须通过设置ssl配置参数打开SSL。

`hostnossl`

这条记录的行为与`hostssl`相反；它只匹配那些在 TCP/IP上不使用SSL的连接企图。

`database`

指定记录所匹配的数据库名称。值`all`指定该记录匹配所有数据库。值 `sameuser`指定如果被请求的数据库和请求的用户同名，则匹配。值`samerole`指定请求的用户必须是一个与数据库同名的角色中的成员（`samegroup`是一个已经废弃了，但目前仍然被接受的`samerole`同义词）。对于一个用于`samerole`目的的角色，超级用户不会被考虑为其中的成员，除非它们是该角色的显式成员（直接或间接），而不是由于超级用户的原因。值`replication`指定如果一个复制连接被请求则该记录匹配（注意复制连接不指定任何特定的数据库）。在其它情况里，这就是一个特定的KingbaseES数据库名字。可以通过用逗号分隔的方法指定多个数据库，也可以通过在文件名前面放`@`来指定一个包含数据库名的

文件。

user

指定这条记录匹配哪些数据库用户名。值all指定它匹配所有用户。否则，它要么是一个特定数据库用户的名字或者是一个有前导+的组名称（回想一下，在KingbaseES里，用户和组没有真正的区别，+实际表示"匹配这个角色的任何直接或间接成员角色"，而没有+记号的名字只匹配指定的角色）。出于这个目的，如果超级用户显式的是一个角色的成员（直接或间接），那么超级用户将只被认为是该角色的一个成员而不是作为一个超级用户。多个用户名可以通过用逗号分隔的方法提供。一个包含用户名的文件可以通过在文件名前面加上@来指定。

address

指定这个记录匹配的客户端机器地址。这个域可以包含一个主机名、一个 IP 地址范围或下文提到的特殊关键字之一。

一个 IP 地址范围以该范围的开始地址的标准数字记号指定，然后是一个斜线（/） 和一个CIDR掩码长度。掩码长度表示客户端 IP 地址必须匹配的高序二进制位位数。在给出的 IP 地址中，这个长度的右边的二进制位必须为零。 在 IP 地址、/和 CIDR 掩码长度之间不能有空白。

这种方法指定一个 IPv4 地址范围的典型例子是： 1720.143.89/32用于一个主机，1720.143.0/24用于一个小型网络， 10.6.0.0/16用于一个大型网络。 一个单主机的 IPv6 地址范围看起来像这样：::1/128（IPv6 回环地址）， 一个小型网络的 IPv6 地址范围则类似于：fe80::7a31:c1ff:0000:0000/96。 0.0.0.0/0表示所有 IPv4 地址，并且::0/0表示所有 IPv6 地址。要指定一个单一主机，IPv4 用一个长度为 32 的 CIDR 掩码或者 IPv6 用 长度为 128 的 CIDR 掩码。在一个网络地址中，不要省略结尾的零。

一个以 IPv4 格式给出的项将只匹配 IPv4 连接并且一个以 IPv6 格式给出的项将只匹配 IPv6 连接，即使对应的地址在 IPv4-in-IPv6 范围内。请注意如果系统的 C 库不支持 IPv6 地址，那么 IPv6 格式中的项将被拒绝。

你也可以写all来匹配任何 IP 地址、写samehost来匹配任何本服务器自身的 IP 地址或者写samenet来匹配本服务器直接连接到的任意子网的任意地址。

如果指定了一个主机名（任何除 IP 地址单位或特殊关键字之外的都被作为主机名处理），该名称会与客户端的 IP 地址的反向名字解析（例如使用 DNS 时的反向 DNS 查找）结果进行比较。主机名比较是大小写敏感的。如果匹配上，那么将在主机名上执行一次正向名字解析（例如正向 DNS 查找）来检查它解析到的任何地址是否等于客户端的 IP 地址。如果两个方向都匹配，则该项被认为匹配（sys_hba.conf中使用的主机名应该是客户端 IP 地址的地址到名字解析返回的结果，否则该行将不会匹配。某些主机名数据库允许将一个 IP 地址关联多个主机名，但是当被要求解析一个 IP 地址时，操作系统将只返回一个主机

名）。

一个以点号（.）开始的主机名声明匹配实际主机名的后缀。因此.example.com将匹配foo.example.com（但不匹配example.com）。

当主机名在sys_hba.conf中被指定时，你应该保证名字解析很快。建立一个类似nscd的本地名字解析缓存是一种不错的选择。另外，你可能希望启用配置参数log_hostname来在日志中查看客户端的主机名而不是 IP 地址。

这个域只适用于host、hostssl和hostnossl记录。

> 用户有时候会疑惑为什么这样处理的主机名看起来很复杂，因为需要两次名字解析（包括一次客户端 IP 地址的反向查找）。在客户端的反向 DNS 项没有建立或者得到某些意料之外的主机名的情况下，这种方式会让该特性的使用变得复杂。这样做主要是为了效率：通过这种方式，一次连接尝试要求最多两次解析器查找，一次逆向以及一次正向。如果有一个解析器对于该地址有问题，这仅仅是客户端的问题。一种假想的替代实现是只做前向查找，这种方法不得不在每一次连接尝试期间解析sys_hba.conf中提到的每一个主机名。如果列出了很多名称，这就会很慢。并且如果主机名之一有解析器问题，它会变成所有人的问题。
>
> 另外，一次反向查找也是实现后缀匹配特性所需的，因为需要知道实际的客户端主机名来与模式进行匹配。
>
> 注意这种行为与其他流行的基于主机名的访问控制实现相一致，例如 Apache HTTP Server 和 TCP Wrappers。

IP-address
IP-mask

这两个域可以被用作*IP-address/ mask-length*记号法的替代方案。和指定掩码长度不同，实际的掩码被指定在一个单独的列中。例如，255.0.0.0表示 IPv4 CIDR 掩码长度 8，而255.255.255.255表示 CIDR 掩码长度 32。

这些域只适用于host、hostssl和hostnossl记录。

auth-method

指定当一个连接匹配这个记录时，要使用的认证方法。下面对可能的选择做了概述。

trust

无条件地允许连接。这种方法允许任何可以与KingbaseES数据库服务器连接的用户以他们期望的任意KingbaseES数据库用户身份登入，而不需要口令或者其他任何认证。详见第 3.1 节。

`reject`

无条件地拒绝连接。这有助于从一个组中"过滤出"特定主机，例如一个`reject`行可以阻塞一个特定的主机连接，而后面一行允许一个特定网络中的其余主机进行连接。

`sha-256`

以SCRAM-SHA-256加密解析的方式验证用户的密码。详见第 3.2 节。

`md5`

以SCRAM-SHA-256或MD5加密解析的方式验证用户的密码。详见第 3.2 节。

`password`

要求客户端提供一个未加密的口令进行认证。因为口令是以明文形式在网络上发送的，所以我们不应该在不可信的网络上使用这种方式。详见第 3.2 节。

`gss`

用 GSSAPI 认证用户。只对 TCP/IP 连接可用。详见第 3.3 节。

`sspi`

用 SSPI 来认证用户。只在 Windows 上可用。详见第 3.4 节。

`ident`

通过联系客户端的 ident 服务器获取客户端的操作系统名，并且检查它是否匹配被请求的数据库用户名。Ident 认证只能在 TCIP/IP 连接上使用。当为本地连接指定这种认证方式时，将用 peer 认证来替代。详见第 3.5 节。

`peer`

从操作系统获得客户端的操作系统用户，并且检查它是否匹配被请求的数据库用户名。这只对本地连接可用。详见第 3.6 节。

`ldap`

使用LDAP服务器认证。详见第 3.7 节。

`radius`

用 RADIUS 服务器认证。详见第 3.8 节。

`cert`

使用 SSL 客户端证书认证。详见第 3.9 节。

`pam`

使用操作系统提供的可插入认证模块服务（PAM）认证。详见第 3.10 节

使用操作系统提供的可插入认证模块服务（PAM）认证。详见第 3.10 节。

`bsd`

使用由操作系统提供的 BSD 认证服务进行认证。详见第 3.11 节。

auth-options

在*auth-method*域的后面，可以是形如*name=value*的域，它们指定认证方法的选项。关于哪些认证方法可以用哪些选项的细节请见下文。

除了下文列出的与方法相关的选项之外，还有一个与方法无关的认证选项`clientcert`，它可以在任何`hostssl`记录中指定。当被设置为`1`时，这个选项要求客户端在认证方法的其他要求之外出示一个有效的（可信的）SSL 证书。

用`@`结构包括的文件被读作一个名字列表，它们可以用空白或者逗号分隔。注释用`#`引入，就像在`sys_hba.conf`中那样，并且允许嵌套`@`结构。除非跟在`@`后面的文件名是一个绝对路径， 文件名都被认为是相对于包含引用文件的目录。

因为每一次连接尝试都会顺序地检查`sys_hba.conf`记录，所以这些记录的顺序是非常关键的。通常，靠前的记录有比较严的连接匹配参数和比较弱的认证方法，而靠后的记录有比较松的匹配参数和比较强的认证方法。 例如，我们希望对本地 TCP/IP 连接使用`trust`认证，而对远程 TCP/IP 连接要求口令。在这种情况下为来自于 127.0.0.1 的连接指定`trust`认证的记录将出现在为一个更宽范围的客户端 IP 地址指定口令认证的记录前面。

在启动以及主服务器进程收到SIGHUP信号时，`sys_hba.conf`文件会被读取。 如果你在活动的系统上编辑了该文件，你将需要通知 kingbase（使用`sys_ctl reload`或`kill -HUP`）重新读取该文件。

> **提示:** 要连接到一个特定数据库，一个用户必须不仅要通过`sys_hba.conf`检查，还必须要有该数据库上的`CONNECT`权限。如果你希望限制哪些用户能够连接到哪些数据库，授予/撤销`CONNECT`权限通常比在`sys_hba.conf`项中设置规则简单。

例 1-1中展示了`sys_hba.conf`项的一些例子。不同认证方法的详情请见下一节。

例 1-1. 示例 `sys_hba.conf` 项

```
# 允许本地系统上的任何用户
# 通过 Unix 域套接字以任意
# 数据库用户名连接到任意数据库
#（本地连接的默认值）。
#
# TYPE  DATABASE    USER    ADDRESS       METHOD
local   all         all                   trust

# 相同的规则，但是使用本地环回 TCP/IP 连接。
#
```

```
# TYPE  DATABASE    USER    ADDRESS       METHOD
host    all         all     127.0.0.1/32 trust

# 和前一行相同，但是使用了一个独立的掩码列
#
# TYPE  DATABASE    USER    IP-ADDRESS    IP-MASK           METHOD
host    all         all     127.0.0.1     255.255.255.255   trust

# IPv6 上相同的规则
#
# TYPE  DATABASE    USER    ADDRESS        METHOD
host    all         all     ::1/128        trust

# 使用主机名的相同规则（通常同时覆盖 IPv4 和 IPv6）。
#
# TYPE  DATABASE    USER    ADDRESS        METHOD
host    all         all     localhost      trust

# 允许来自任意具有 IP 地址
# 19168.93.x 的主机上任意
# 用户以 ident 为该连接所
# 报告的相同用户名连接到
# 数据库 "kingbase"
#（通常是操作系统用户名）。
#
# TYPE  DATABASE    USER      ADDRESS         METHOD
host    kingbase    all       19168.93.0/24   ident

# 如果用户的口令被正确提供，
# 允许来自主机 19168.110
# 的任意用户连接到数据库 "kingbase"。
#
# TYPE  DATABASE    USER       ADDRESS         METHOD
host    kingbase    all        19168.110/32    md5

# 如果用户的口令被正确提供，
# 允许 example.com 中主机上
# 的任意用户连接到任意数据库。
#
# TYPE  DATABASE    USER       ADDRESS         METHOD
host    all         all        .example.com    md5

# 如果没有前面的 "host" 行，这两
# 行将拒绝所有来自 19168.54.1
# 的连接（因为那些项将首先被匹配），
# 但是允许来自互联网其他任何地方的
# GSSAPI 连接。零掩码导致主机IP 地址中的所有位都不会被考虑，
# 因此它匹配任意主机。
#
```

```
# TYPE  DATABASE    USER      ADDRESS           METHOD
host    all         all       19168.54.1/32     reject
host    all         all       0.0.0.0/0         gss

# 允许来自 19168.x.x 主机的用户
# 连接到任意数据库，如果它们能够
# 通过 ident 检查。例如，假设 ident
# 说用户是 "bryanh" 并且他要求以
# KingbaseES 用户 "guest1" 连接，
# 如果在 sys_ident.conf 有一个映射
# "omicron" 的选项说 "bryanh" 被
# 允许以 "guest1" 连接，则该连接将被允许。
#
# TYPE  DATABASE    USER    ADDRESS          METHOD
host    all         all     19168.0.0/16     ident map=omicron

# 如果这些是本地连接的唯一三行，
# 它们将允许本地用户只连接到它们
# 自己的数据库（与其数据库用户名
# 同名的数据库），不过管理员和角
# 色 "support" 的成员除外（它们可
# 以连接到所有数据库）。文件
# $KINGBASE_DATA/admins 包含一个管理员
# 名字的列表。在所有情况下都要求口令。
#
# TYPE  DATABASE    USER      ADDRESS           METHOD
local   sameuser    all                         md5
local   all         @admins                     md5
local   all         +support                    md5

# 上面的最后两行可以被整合为一行：
local   all         @admins,+support            md5

# 数据库列也可以用列表和文件名：
local   db1,db2,@demodbs  all                   md5
```

2. 用户名映射

当使用像 Ident 或者 GSSAPI 之类的外部认证系统时，发起连接的操作系统用户名可能不同于用来连接的数据库用户。在这种情况下，一个用户名映射可被用来把操作系统用户名映射到数据库用户。要使用用户名映射，在`sys_hba.conf`的选项域指定`map=`*`map-name`*。此选项支持所有接收外部用户名的认证方法。由于不同的连接可能需要不同的映射，在`sys_hba.conf`中的*`map-name`*参数中指定要被使用的映射名，用以指示哪个映射用于每个个体连接。

用户名映射定义在 ident 映射文件中，默认情况下它被命名为`sys_ident.conf`并被存储在集簇的数据目录中。ident 映射文件包含的行的一般格式：

```
map-name system-username database-username
```

以在sys_hba.conf中同样的方式处理注释和空白。*map-name*是一个任意名称，它将被用于在sys_hba.conf中引用该映射。其他两个域指定一个操作系统用户名和一个匹配的数据库用户名。相同的*map-name*可以被反复地用在同一个映射中指定多个用户映射。

对于一个给定操作系统用户可以对应多少个数据库用户没有限制，反之亦然。因此，一个映射中的项应该被看成意味着"这个操作系统用户被允许作为这个数据库用户连接"，而不是按时它们等价。如果有任何映射项把从外部认证系统获得的用户名和用户要求的数据库用户名配对，该连接将被允许。

如果*system-username*域以一个斜线（/）开始，域的剩余部分被当做一个正则表达式。正则表达式可以包括一个单一的捕获，或圆括号子表达式，然后它可以在*database-username*域中以\1（反斜线一）被引用。这允许在单个行中多个用户名的映射，这特别有助于简单的语法替换。例如，这些项

```
mymap   /^(.*)@mydomain\.com$       \1
mymap   /^(.*)@otherdomain\.com$    guest
```

将为用户移除以@mydomain.com结束的系统用户名的域部分，以及允许系统名以@otherdomain.com结束的任意用户作为guest登入。

> **提示:** 记住在默认情况下，一个正则表达式可以只匹配字符串的一部分。如上例所示，使用^和$来强制匹配整个系统用户名通常是明智的。

在启动以及主服务器进程收到SIGHUP信号时，sys_ident.conf文件会被读取。 如果你在活动的系统上编辑了该文件，你将需要通知 kingbase（使用sys_ctl reload或kill -HUP）重新读取该文件。

例 2-1中展示了一个可以联合sys_hba.conf文件使用的sys_ident.conf文件。在这个例子中，对于任何登入到 19168 网络上的一台机器的用户， 如果该用户没有操作系统用户名bryanh、ann或robert，则他不会被授予访问权限。只有当 Unix 用户robert尝试作为KingbaseES用户bob（而不是作为robert或其他人）连接时，他才被允许访问。ann只被允许作为ann连接。用户bryanh被允许以bryanh或者guest1连接。

例 2-1 一个示例 sys_ident.conf 文件

```
# MAPNAME       SYSTEM-USERNAME         KDB-USERNAME

omicron         bryanh                  bryanh
omicron         ann                     ann
# bob 在这些机器上有用户名 robert
omicron         robert                  bob
# bryanh 也可以作为 guest1 连接
omicron         bryanh                  guest1
```

3. 认证方法

下列小节更详细地描述认证方法。

3.1. 信任认证

当trust认证被指定时，KingbaseES假设任何可以连接到服务器的人都被授权使用他们指定的任何数据库用户名（即使是超级用户）访问数据库。当然，在database和 user列中设置的限制仍然适用。只有当在操作系统层对进入服务器的连接有足够保护时，才应该使用这种方法。

trust认证对于单用户工作站的本地连接是非常合适和方便的。通常它本身*不*适用于一台多用户机器。不过，只要你利用文件系统权限限制了对服务器的 Unix 域套接字文件的访问，即使在多用户机器上，你也可以使用trust。 要做这些限制，你可以设置连接和认证中描述的unix_socket_permissions配置参数（可能还有unix_socket_group）。 或者你可以设置unix_socket_directories配置参数来把 Unix 域套接字文件放在一个经过恰当限制的目录中。

设置文件系统权限只能有助于 Unix 套接字连接。本地 TCP/IP 连接不会被文件系统权限限制。因此，如果你想利用文件系统权限来控制本地安全，那么从sys_hba.conf中移除host ... 127.0.0.1 ...行，或者把它改为一个非trust认证方法。

如果通过指定trust的sys_hba.conf行让你信任每一个被允许连接到服务器的机器上的用户，trust认证只适合 TCP/IP 连接。为任何不是来自localhost（127.0.0.1）的 TCP/IP 连接使用trust很少是合理的。

3.2 口令认证

基于口令的认证方式包括sha-256、md5和password。这些认证方法的启用操作都一样，区别在于在服务器上用户密码的存储方式以及客户端连接发送密码的方式有所不同

sha-256

sha-256认证方式是一种挑战-响应架构的认证方式，可防止密码在不可信连接上被嗅探，并支持以密码散列的形式将密码存储在服务器上。

这是目前提供的口令认证方法中最安全的方法，但旧版数据库并不支持。

md5

md5认证方式使用自定义安全性较低的质询-响应机制。它也可以防止密码嗅探，并避免以纯文本形式将密存储在服务器上，但如果攻击者设法从服务器窃取密码哈希，则密码将面临被破解的风险。所以，MD5哈希算法现在已经不再被认为是完全安全的加密方式了。

md5认证方式不能使用db_user_namespace特性。

为了简化从md5认证方式到新的sha-256认证方式的转换，如果在pg_hba.conf中指定md5为口令认证方式，但服务器上用户的密码是以sha-256方式加密的， 那么自动选择基于sha-256的认证方式。

password

password认证方式将以明文形式发送密码，因此易受密码嗅探攻击。如果可能，应始终避免使用它。不过如果连接受到SSL加密保护，那么password的认证方式就很安全。（那样的话SSL，SSL证书认证是更好的选择）。

口令认证方式的可用性取决于用户的密码在服务器上的加密方式。这是在设置密码时由配置参数password_encryption控制的。如果使用sha-256加密方式对密码进行加密，那么可用的口令认证方式为sha-256和password（后一种情况下密码将以纯文本形式传输）。而此时若认证方式为md5，则会自动切换到sha-256，所以它也可以工作。如果密码是使用md5设置加密的，那么它只能用md5和password认证方式（同样，在后一种情况下是用明文传输密码的）。

KingbaseES数据库口令独立于操作系统用户口令。每个数据库用户的口令被存储在sys_authid系统目录中。口令可以用 SQL 命令CREATE USER和ALTER ROLE管理，例如CREATE USER foo WITH PASSWORD 'secret'。如果没有为一个用户设置口令，那么存储的口令为空并且对该用户的口令认证总会失败。

3.3. GSSAPI 认证

GSSAPI是用于 RFC 2743 中定义的安全认证的一个工业标准协议。KingbaseES根据 RFC 1964

支持带Kerberos认证的GSSAPI。GSSAPI为支持它的系统提供自动认证（单点登录）。认证本身是安全的，但通过数据库连接发送的数据将不被加密，除非使用SSL。

当编译KingbaseES时，GSSAPI 支持必须被启用。

当GSSAPI使用Kerberos时， 它会使用格式为 *servicename*/*hostname*@*realm*的标准 principal。KingbaseES服务器将接受该服务器所使用的 keytab 中包括的任何 principal，但是在从使用krbsrvname连接参数的客户端建立连接时要注意指定正确的 principal 细节。安装的默认值kingbase可以在编译时使用./configure --with-krb-srvnam=*其他值*修改。在大部分的环境中，这个参数从不需要被更改。某些 Kerberos 实现可能要求一个不同的服务名，例如 Microsoft Active Directory 要求服务名是大写形式（KINGBASE）。

*hostname*是服务器机器的被完全限定的主机名。服务 principal 的 realm 是该服务器机器的首选 realm。

客户端 principal 可以被通过sys_ident.conf映射到不同的KingbaseES数据库用户名。例如，username@realm可能会被映射到username。或者，你可以使用完整的username@realm当事人作为KingbaseES中的角色而无需任何映射。

KingbaseES也支持一个参数把 realm 从 principal 中剥离。这种方法是为了向后兼容性，并且我们强烈反对使用它，因为这样就无法区分具有相同用户名却来自不同 realm 的不同用户了。要启用这种方法，可将include_realm设置为 0。对于简单的单 realm 安装，这样做并且设置krb_realm参数（这会检查 principal 的 realm 是否正好匹配krb_realm中的参数）仍然是安全的。但比起在sys_ident.conf中指定一个显式映射来说，这种方法的能力较低。

确认你的服务器的 keytab 文件是可以被KingbaseES服务器帐户读取的。密钥文件的位置由配置参数krb_server_keyfile指定。默认是/usr/local/kingbase/etc/krb5.keytab（或者任何在编译的时候作为sysconfdir的目录）。出于安全原因，推荐对KingbaseES服务器使用一个独立的 keytab而不是开放系统 keytab 文件的权限。

keytab 文件由 Kerberos 软件生成，详见 Kerberos 文档。下面是 MIT 兼容的 Kerberos 5 实现的例子：

```
kadmin% ank -randkey kingbase/server.my.domain.org
kadmin% ktadd -k krb5.keytab kingbase/server.my.domain.org
```

当连接到数据库时，确保你有一个匹配被请求数据库用户名的 principal 的票据。例如，对于数据库用户名fred，principal fred@EXAMPLE.COM将能够连接。要也允许 principal fred/users.example.com@EXAMPLE.COM，可使用一个用户名映射，如第 2 节中所述。

下列被支持的配置选项用于GSSAPI：

include_realm

如果设置为 0，在通过用户名映射之前（第 2 节），来自已认证用户 principal 的 realm 名称会被剥离掉。我们不鼓励这样做，这种方法主要是为了向后兼容性而存在的，因为它在多 realm 环境中是不安全的（除非也使用 krb_realm）。推荐用户让include_realm设置为默认值（1）并且在sys_ident.conf中提供一条显式的映射来把 principal 名称转换成KingbaseES用户名。

map

允许在系统和数据库用户名之间的映射。详见第 2 节。对于一个 GSSAPI/Kerberos 原则，例如username@EXAMPLE.COM（或者更不常见的username/hostbased@EXAMPLE.COM），用于映射的用户名会是username@EXAMPLE.COM（或者username/hostbased@EXAMPLE.COM，相应地），除非include_realm已经被设置为 0，在那种情况下username（或者username/hostbased）是映射时被视作系统用户名的东西。

krb_realm

设置 realm 为对用户 principal 名进行匹配的范围。如果这个参数被设置，只有那个 realm 的用户将被接受。如果它没有被设置，任何 realm 的用户都能连接，服从任何已完成的用户名映射。

3.4. SSPI 认证

SSPI是一种用于带单点登录的安全认证的Windows技术。 KingbaseES在negotiate模式中将使用 SSPI，它在可能的情况下使用Kerberos并在其他情况下自动降回到NTLM。只有在服务器和客户端都运行着Windows时，SSPI才能工作。或者在非 Windows 平台上GSSAPI可用时，SSPI也能工作。

当使用Kerberos认证时，SSPI和GSSAPI的工作方式相同，详见第 3.3 节。

下列被支持的配置选项用于SSPI：

include_realm

如果设置为 0，在通过用户名映射之前（第 2 节），来自已认证用户 principal 的 realm 名称会被剥离掉。我们不鼓励这样做，这种方法主要是为了向后兼容性而存在的，因为它在多 realm 环境中是不安全的（除非也使用krb_realm）。推荐用户让 include_realm 设置为默认值（1）并且在sys_ident.conf中提供一条显式的映射来把 principal 名称转换成KingbaseES用户名。

compat_realm

如果被设置为 1，该域的 SAM 兼容名称（也被称为 NetBIOS 名称）被用于include_realm选项。这是默认值。如果被设置为 0，会使用来自 Kerberos 用户主名的真实 realm 名称。

不要禁用这个选项，除非你的服务器运行在一个域账号（这包括一个域成员系统上的虚拟服务账号）下并且所有通过 SSPI 认证的所有客户端也在使用域账号，否则认证将会失败。

upn_username

如果这个选项和compat_realm一起被启用，来自 Kerberos UPN 的用户名会被用于认证。如果它被禁用（默认），会使用 SAM 兼容的用户名。默认情况下，对于新用户账号这两种名称是一样的。

注意如果没有显式指定用户名，libkci会使用 SAM 兼容的名称。如果你使用的是libkci或者基于它的驱动，你应该让这个选项保持禁用或者在连接字符串中显式指定用户名。

map

允许在系统和数据库用户名之间的映射。详见第 2 节。对于一个 GSSAPI/Kerberos 原则，例如username@EXAMPLE.COM（或者更不常见的username/hostbased@EXAMPLE.COM），用于映射的用户名会是username@EXAMPLE.COM（或者username/hostbased@EXAMPLE.COM，相应地），除非include_realm已经被设置为 0，在那种情况下username（或者username/hostbased）是映射时被视作系统用户名的东西。

krb_realm

设置领域为对用户 principal 名进行匹配的范围。如果这个参数被设置，只有那个领域的用户将被接受。如果它没有被设置，任何领域的用户都能连接，服从任何已完成的用户名映射。

3.5. Ident 认证

ident 认证方法通过从一个 ident 服务器获得客户端的操作系统用户名并且用它作为被允许的数据库用户名（和可选的用户名映射）来工作。它只在 TCP/IP 连接上支持。

> **注意：**当为一个本地（非 TCP/IP）连接指定 ident 时，将实际使用 peer 认证（见第 3.6 节）。

下列被支持的配置选项用于ident：

`map`

允许系统和数据库用户名之间的映射。详见第 2 节。

"Identification Protocol（标识协议）"在 RFC 1413 中描述。实际上每个类 Unix 操作系统都带着一个默认监听 TCP 113 端口的 ident 服务器。ident 服务器的基本功能是回答类似这样的问题："哪个用户从你的端口*X*发起了连接并且连到了我的端口*Y*？"。因为当一个物理连接被建立后，KingbaseES既知道*X*也知道*Y*， 所以它可以询问尝试连接的客户端主机上的 ident 服务器并且在理论上可以判断任意给定连接的操作系统用户。

这个过程的缺点是它依赖于客户端的完整性：如果客户端机器不可信或者被攻破，攻击者可能在 113 端口上运行任何程序并且返回他们选择的任何用户。因此这种认证方法只适用于封闭的网络， 这样的网络中的每台客户端机器都处于严密的控制下并且数据库和操作系统管理员操作时可以方便地联系。换句话说，你必须信任运行 ident 服务器的机器。注意这样的警告：

标识协议的本意不是作为一种认证或访问控制协议。

--RFC 1413

有些 ident 服务器有一个非标准的选项，它导致返回的用户名是被加密的，使用的是只有原机器管理员知道的一个密钥。当与KingbaseES配合使用 ident 服务器时，一定不要使用这个选项，因为KingbaseES没有任何方法对返回的字符串进行解密以获取实际的用户名。

3.6. Peer 认证

Peer 认证方法通过从内核获得客户端的操作系统用户名并把它用作被允许的数据库用户名（和可选的用户名映射）来工作。这种方法只在本地连接上支持。

下列被支持的配置选项用于peer：

`map`

允许在系统和数据库用户名之间的映射。详见第 2 节。

Peer 认证只在提供`getpeereid()`函数、`SO_PEERCRED`套接字参数或相似机制的操作系统上可用。这些 OS 当前包括Linux、大部分的BSD包括OS X以及Solaris。

3.7. LDAP 认证

这种认证方法操作起来类似于password， 只不过它使用 LDAP 作为密码验证方法。LDAP 只被用于验证用户名/口令对。因此，在使用 LDAP 进行认证之前，用户必须已经存在于数据库中。

LDAP 认证可以在两种模式下操作。在第一种模式中（我们将称之为简单绑定模式），服务器将绑定到构造成*prefix username suffix*的可区分名称。通常，*prefix*参数被用于指定cn=或者一个活动录环境中的*DOMAIN*\。*suffix*被用来指定非活动目录环境中的DN的剩余部分。

在第二种模式中（我们将称之为搜索与绑定模式），服务器首先用一个固定的用户名和密码（用*ldapbinddn*和*ldapbindpasswd*指定）绑定到 LDAP 目录，并为试图登入该数据库的用户执行一次搜索。如果没有配置用户名和密码， 将尝试一次匿名绑定到目录。搜索将在位于*ldapbasedn*的子树上被执行，并将尝试做一次*ldapsearchattribute*中指定属性的精确匹配。一旦在这次搜索中找到用户，服务器断开并且作为这个用户重新绑定到目录，使用由客户端指定的口令来验证登录是正确的。这种模式与在其他软件中的 LDAP 认证所使用的相同，例如 Apache mod_authnz_ldap 和 pam_ldap。这种方法允许位于目录中用户对象的更大灵活性，但是会导致建立两个到 LDAP 服务器的独立连接。

下列配置选项被用于两种模式：

ldapserver

连接的 LDAP 服务器的名称或 IP 地址。可以指定多个服务器，用空格分隔。

ldapport

要连接的 LDAP 服务器的端口号。如果没有指定端口，LDAP 库的默认端口设置将被使用。

ldaptls

设置为 1 以使 KingbaseES 和 LDAP 服务器之间的连接使用 TLS 加密。请注意，这里仅加密到 LDAP 服务器的流量 — 到客户端的连接将不被加密，除非使用 SSL。

下列选项只被用于简单绑定模式：

ldapprefix

当做简单绑定认证时，前置到用户名形成要用于绑定的 DN 的字符串。

ldapsuffix

做简单绑定认证时，前置到用户名形成要用于绑定的 DN 的字符串。

下列选项只被用于搜索与绑定模式：

ldapbasedn

当做搜索与绑定认证时，开始搜索用户的根 DN。

ldapbinddn

当做搜索与绑定认证时，用户要绑定到目录开始执行搜索的 DN。

ldapbindpasswd

当做搜索与绑定认证时，用户用于绑定到目录开始执行搜索的口令。

ldapsearchattribute

当做搜索与绑定认证时，在搜索中用来与用户名匹配的属性。如果没有指定属性，将会使用uid属性。

ldapurl

一个 RFC 4516 LDAP URL。这是一种用更紧凑和标准的形式书写某些其他 LDAP 选项的可选方法。格式是

```
ldap://host[:port]/basedn[?[attribute][?[scope]]]
```

*scope*必须是base、one、sub之一，通常是后者。只有一个属性会被使用，并且某些标准 LDAP URL 的其他部件（如过滤器和扩展）不被支持。

对于非匿名绑定，ldapbinddn和ldapbindpasswd必须被指定为独立选项。

要使用加密的 LDAP 连接，在ldapurl之外还必须使用ldaptls选项。ldaps URL 模式（直接 SSL 连接）不被支持。

LDAP URL 当前只支持 OpenLDAP，而不支持 Windows。

将简单绑定的选项中混合用于搜索与绑定的选项是一种错误。

这里是一个简单绑定 LDAP 配置的例子：

```
host ... ldap ldapserver=ldap.example.net ldapprefix="cn="
ldapsuffix=", dc=example, dc=net"
```

当请求一个作为数据库用户someuser到数据库服务器的连接时，KingbaseES 将尝试使用cn=someuser, dc=example, dc=net和客户端提供的口令来绑定到 LDAP 服务器。如果那个连接成功，将被授予数据库访问。

这里是一个搜索与绑定配置的例子：

```
host ... ldap ldapserver=ldap.example.net ldapbasedn="
dc=example, dc=net" ldapsearchattribute=uid
```

当请求一个作为数据库用户someuser到数据库服务器的连接时，KingbaseES 将尝试匿名绑定

（因为没有指定ldapbinddn）到 LDAP 服务器，在指定的基础 DN 下执行一次对于(uid=someuser)的搜索。如果找到一个项，则它将尝试使用找到的信息和客户端提供的口令进行绑定。如果第二个连接成功，将被授予数据库访问。

这里是被写成一个 URL 的相同搜索与绑定配置：

```
host ... ldap ldapurl="ldap://ldap.example.net/
dc=example,dc=net?uid?sub"
```

一些支持根据 LDAP 认证的其他软件使用相同的 URL 格式，因此很容易共享该配置。

> **提示:** 如例子中所示，由于 LDAP 通常使用逗号和空格来分割一个 DN 的不同部分，在配置 LDAP 选项时通常有必要使用双引号包围的参数值。

3.8. RADIUS 认证

这种认证方法的操作类似于password，不过它使用 RADIUS 作为密码验证方式。RADIUS 只被用于验证 用户名/密码对。因此，在 RADIUS 能被用于认证之前，用户必须已经存在于数据库中。

当使用 RADIUS 认证时，一个访问请求消息将被发送到配置好的 RADIUS 服务器。这一请求将是Authenticate Only类型，并且包含参数user name、password（加密的）和NAS Identifier。该请求将使用一个与服务器共享的密钥加密。RADIUS 服务器将对这个服务器响应Access Accept或者Access Reject。不支持RADIUS accounting。

下列被支持的配置选项用于 RADIUS：

radiusserver

连接到 RADIUS 服务器的名称或IP地址。此参数是必需的。

radiussecret

和 RADIUS 服务器秘密交谈时会用到共享密钥。这在 KingbaseES 和 RADIUS 服务器之间必须有完全相同的值。我们推荐用一个至少 16 个字符的字符串。这个参数是必需的。

> **注意：**如果KingbaseES编译为支持OpenSSL，所用的加密向量将只是强密码。在其他情况下，到 RADIUS 服务器的传输应该被视为应该被视为被混淆的、不安全的。如有必要，应采用外部安全措施。

radiusport

用于连接到 RADIUS 服务器的端口号。如果没有指定端口，则使用默认端口1812。

radiusidentifier

在 RADIUS 请求中字符串被用作NAS Identifier。 这个参数可以被用作第二个参数标识例如该用户试图以哪个数据库用户进行认证，它可以被用于 RADIUS 服务器上的策略匹配。如果没有指定标识符，默认使用kingbase。

3.9. 证书认证

这种认证方法使用 SSL 客户端证书执行认证。因此，它只适用于 SSL 连接。当使用这种认证方法时，服务器将要求客户端提供一个有效的、可信的证书。不会有密码提示将被发送到客户端。证书的cn（通用名）属性将与被请求的数据库用户名进行比较，并且如果匹配将允许登录。用户名映射可以被用来允许cn与数据库用户名不同。

下列被支持的配置选项用于 SSL 证书认证：

map

允许在系统和数据库用户名之间的映射。详见第 2 节。

在一条指定证书认证的sys_hba.conf记录中，认证选项clientcert被假定为1，并且它不能被关掉，因为这种方法中一个客户端证书是必需的。cert方法对基本clientcert证书验证测试所增加的东西是检查cn属性是否匹配数据库用户名。

3.10. PAM 认证

这种认证方法操作起来类似password， 只不过它使用 PAM （插入式验证模块）作为认证机制。默认的 PAM 服务名是kingbase。PAM 只被用于验证用户名/口令对并且可以有选择地验证已连接的远程主机名或 IP 地址。因此，在使用 PAM 进行认证之前，用户必须已经存在于数据库中。有关 PAM 的更多信息，请阅读 Linux-PAM 页面。

下列被支持的配置选项用于 PAM：

pamservice

PAM服务名称。

pam_use_hostname

判断是否通过PAM_RHOST项把远程 IP 地址或者主机名提供给 PAM 模块。默认情况下会使用 IP 地址。把这个选项设置为 1 可以使用解析过的主机名。主机名解析可能导致登录延迟（大部分的 PAM 配置不使用这些信息，因此只有使用为利用这种信息而特别创建的 PAM 配置时才需要考虑这个设置）。

> **注意：** 如果 PAM 被设置为读取/etc/shadow，认证将会失败，因为 KingbaseES 服务器是由一个非 root 用户启动 。然而，当 PAM 被配置为使用 LDAP 或其他认证验证方法时这就不是一个问题。

3.11. BSD 认证

这种认证方法操作起来类似于password，不过它使用 BSD 认证来验证口令。BSD 认证只被用来验证用户名/口令对。因此，在 BSD 认证可以被用于认证之前，用户的角色必须已经存在于数据库中。BSD 认证框架当前只在 OpenBSD 上可用。

KingbaseES中的 BSD 认证使用auth-kingbase登录类型，如果login.conf中定义了kingbase登录分类，就会用它来认证。默认情况下这种登录分类不存在，KingbaseES将使用默认的登录分类。

> **注意：** 要使用 BSD 认证，KingbaseES 用户账号（也就是运行服务器的操作系统用户）必须首先被加入到auth组中。在 OpenBSD 系统上默认存在auth组。

4. 认证问题

认证失败以及相关的问题通常由类似下面的错误消息显示：

```
FATAL:  no sys_hba.conf entry for host "123.123.123.123",
user "andym", database "testdb"
```

这条消息最可能出现的情况是你成功地连接了服务器，但它不愿意与你沟通。就像消息本身所建议的，服务器拒绝了连接请求，因为它没有在其sys_hba.conf配置文件里找到匹配项。

```
FATAL:  password authentication failed for user "andym"
```

这样的消息表示你连接了服务器，并且它也愿意和你沟通，但是你必须通过sys_hba.conf文件中指定的认证方法。检查你提供的口令，或者如果错误消息提到了 Kerberos 或 ident 认证类型，检查那些软件。

```
FATAL:  user "andym" does not exist
```

指示的数据库用户没有被找到。

```
FATAL:  database "testdb" does not exist
```

你试图连接的数据库不存在。请注意如果你没有声明数据库名，默认会用数据库用户名作为数据库名，这可能正确也可能不正确。

> **提示:** 服务器日志可能包含比报告给客户端的更多的有关认证失败的信息。如果困惑于认证失败的原因，请检查服务器日志。

连接和认证

1. 连接设置

`listen_addresses` (`string`)

指定服务器在哪些 TCP/IP 地址上监听客户端连接。值的形式是一个逗号分隔的主机名和/或数字 IP 地址列表。特殊项`*`对应所有可用 IP 接口。项`0.0.0.0`允许监听所有 IPv4 地址并且`::`允许监听所有 IPv6 地址。如果列表为空，服务器将根本不会监听任何 IP 接口，在这种情况中只能使用 Unix 域套接字来连接它。默认值是localhost，它只允许建立本地 TCP/IP "环回"连接。虽然客户端认证允许细粒度地控制谁能访问服务器，`listen_addresses`控制哪些接口接受连接尝试，这能帮助在不安全网络接口上阻止重复的恶意连接请求。这个参数只能在服务器启动时设置。

`port` (`integer`)

服务器监听的 TCP 端口；默认是 54321 。请注意服务器会同一个端口号监听所有的 IP 地址。这个参数只能在服务器启动时设置。

`max_connections` (`integer`)

决定数据库的最大并发连接数。默认值通常是 100 个连接，但是如果内核设置不支持（initdb时决定），可能会比这个 数少。这个参数只能在服务器启动时设置。

当运行一个后备服务器时，你必须设置这个参数等于或大于主服务器上的参数。否则，后备服务器上可能无法允许查询。

`superuser_reserved_connections` (`integer`)

决定为KingbaseES超级用户连接而保留的连接"槽"数。 同时活跃的并发连接最多max_connections个。任何时候，活跃的并发连接数最多为`max_connections`减去`superuser_reserved_connections`，新连接就只能由超级用户发起了，并且不会有新的复制连接被接受。

默认值是 3 。这个值必须小于`max_connections`的值。 这个参数只能在服务器启动时设置。

`unix_socket_directories` (`string`)

指定服务器用于监听来自客户端应用的连接的 Unix 域套接字目录。通过列出用逗号分隔的多个目录可以建立多个套接字。项之间的空白被忽略，如果你需要在名字中包括空白或逗

号，在目录名周围放上双引号。一个空值指定在任何 Unix 域套接字上都不监听，在这种情况中只能使用 TCP/IP 套接字来连接到服务器。默认值通常是/tmp，但是在编译时可以被改变。这个参数只能在服务器启动时设置。

除了套接字文件本身（名为.s.PGSQL.*nnnn*，其中*nnnn*是服务器的端口号），一个名为.s.PGSQL.*nnnn*.lock的普通文件会在每一个unix_socket_directories目录中被创建。任何一个都不应该被手工移除。

Windows下没有 Unix 域套接字，因此这个参数与 Windows 无关。

unix_socket_group (string)

设置 Unix 域套接字的所属组（套接字的所属用户总是启动服务器的用户）。可以与选项unix_socket_permissions一起用于对 Unix域连接进行访问控制。默认是一个空字符串，表示服务器用户的默认组。这个参数只能在服务器启动时设置。

Windows 下没有 Unix 域套接字，因此这个参数与 Windows 无关。

unix_socket_permissions (integer)

设置 Unix 域套接字的访问权限。Unix 域套接字使用普通的 Unix 文件系统权限集。这个参数值应该是数字的形式，也就是系统调用chmod和umask接受的 形式（如果使用自定义的八进制格式，数字必须以一个0（零）开头）。

默认的权限是0777，意思是任何人都可以连接。合理的候选是0770（只有用户和同组的人可以访问， 又见unix_socket_group）和0700（只有用户自己可以访问）（请注意，对于 Unix 域套接字，只有写权限有麻烦，因此没有对读取和执行权限的设置和收回）。

这个参数只能在服务器启动时设置。

这个参数与完全忽略套接字权限的系统无关，尤其是自版本10以上的Solaris。 在那些系统上，可以通过把unix_socket_directories指向一个把搜索权限 限制给指定用户的目录来实现相似的效果。 因为 Windows 下没有 Unix 域套接字，因此这个参数也与 Windows 无关。

bonjour (boolean)

通过Bonjour广告服务器的存在。默认值是关闭。 这个参数只能在服务器启动时设置。

bonjour_name (string)

指定Bonjour服务名称。空字符串''（默认值）表示使用计算机名。 如果编译时没有打开Bonjour支持那么将忽略这个参数。这个参数只能在服务器启动时设置。

tcp_keepalives_idle (integer)

指定不活动多少秒之后通过 TCP 向客户端发送一个 keepalive 消息。 0 值表示使用默认值。这个参数只有在支持TCP_KEEPIDLE或TCP_KEEPALIVE符号的系统或 Windows 上才可以使用。在其他系统上，它必须为零。在通过 Unix 域套接字连接的会话中，这个参数被忽略并且总是读作零。

> **注意:** 在 Windows 上，值若为 0，系统会将该参数设置为 2 小时，因为 Windows 不支持读取系统默认值。

tcp_keepalives_interval (integer)

指定在多少秒之后重发一个还没有被客户端告知已收到的 TCP keepalive 消息。0 值表示使用系统默认值。这个参数只有在支持TCP_KEEPINTVL符号的系统或 Windows 上才可以使用。在其他系统上，必须为零。在通过 Unix域套接字连接的会话中，这个参数被忽略并总被读作零。

> **注意:** 在 Windows 上，值若为 0，系统会将该参数设置为 1 秒，因为 Windows 不支持读取系统默认值。

tcp_keepalives_count (integer)

指定与客户端的服务器连接被认为死掉之前允许丢失的 TCP keepalive 数量。0 值表示使用系统默认值。这个参数只有在支持TCP_KEEPCNT符号的系统上才可以使用。在其他系统上，必须为零。在通过 Unix 域套接字连接的会话中，这个参数被忽略并总被读作零。

> **注意:** Windows 不支持该参数，且必须为零。

2. 安全和认证

authentication_timeout (integer)

完成客户端认证的最长时间，以秒计。如果一个客户端没有在这段时间里完成 认证协议，服务器将关闭连接。这样就避免了出问题的客户端无限制地占有一个连接。默认值是 1分钟（1m）。这个参数只能在服务器命令行上或者在kingbase.conf文件中设置。

ssl (boolean)

启用SSL连接。请在使用这个参数之前阅读*服务器管理指南-8. 用 SSL 进行安全的 TCP/IP 连接*。默认是off。 这个选项只能在服务器启动时设置。SSL通信只能和 TCP/IP 连接一起使用。

ssl_ca_file (string)

指定包含 SSL 服务器证书颁发机构（CA）的文件名。默认值为空，表示不载入 CA 文件，并且不执行客户端证书验证。相对路径是相对于数据目录。这个参数只能在服务器启动时设置。

ssl_cert_file (string)

指定包含 SSL 服务器证书的文件名。默认值是server.crt。相对路径是相对于数据目录的。这个参数只能在服务器启动时设置。

ssl_crl_file (string)

指定包含 SSL 服务器证书撤销列表（CRL）的文件名。默认值为空，意味着不载入 CRL 文件。相对路径是相对于数据目录。这个参数只能在服务器启动时设置。

ssl_key_file (string)

指定包含 SSL 服务器私钥的文件名。默认值为server.key。 相对路径是相对于数据目录。这个参数只能在服务器启动时设置。

ssl_ciphers (string)

指定一个SSL密码列表，用于安全连接。这个设置的语法和所支持的值列表可以 参见OpenSSL包中的 ciphers手册页。默认值是 HIGH:MEDIUM:+3DES:!aNULL。它通常是合理的，除非你有特别的安全性需求。

默认值的解释：

HIGH

使用来自HIGH组的密码的密码组（例如 AES, Camellia, 3DES）

MEDIUM

使用来自MEDIUM组的密码的密码组（例如 RC4, SEED）

+3DES

OpenSSL 对HIGH的默认排序是有问题的，因为它认为 3DES 比 AES128 更高。这是错误的，因为 3DES 提供的安全性比 AES128 低，并且它也更加慢。 +3DES把它重新排序在所有其他HIGH和 MEDIUM密码之后。

!aNULL

禁用不做认证的匿名密码组。这类密码组容易收到中间人攻击，因此不应被使用。

可用的密码组细节可能会随着 OpenSSL 版本变化。可使用命令 openssl ciphers -v 'HIGH:MEDIUM:+3DES:!aNULL'来查看 当前安装的OpenSSL版本的实际细节。注意这个列表是根据服务器密钥类型 在运行时过滤过的。

ssl_prefer_server_ciphers (bool)

指定是否使用服务器的 SSL 密码首选项，而不是用客户端的。默认为真。

ssl_ecdh_curve (string)

指定用在ECDH密钥交换中的曲线名称。它需要被所有连接的客户端支持。 它不需要与服务器椭圆曲线密钥使用的曲线相同。默认值是prime256v1。

OpenSSL 命名了最常见的曲线： prime256v1 (NIST P-256)、 secp384r1 (NIST P-384)、 secp521r1 (NIST P-521)。

openssl ecparam -list_curves命令可以显示可用曲线的完 整列表。不过并不是所有的都在TLS中可用。

password_encryption (boolean)

当在CREATE USER或ALTER ROLE中指定了一个密码，而没有写ENCRYPTED或UNENCRYPTED时，这个参数决定是否密码会被加密。默认值是on（加密密码）。

krb_server_keyfile (string)

设置 Kerberos 服务器密钥文件的位置。这个参数只能在 kingbase.conf文件中或服务器命令行上进行设置。

krb_caseins_users (boolean)

设置 Kerberos 和 GSSAPI 用户名是否应区分大小写。默认是off（区分大小写）。这个参数只能在 kingbase.conf文件中或服务器命令行上进行设置。

db_user_namespace (boolean)

允许针对每个数据库的用户名。默认是关闭的。这个参数只能在 kingbase.conf文件中或服务器命令行上进行设置。

如果打开这个参数，你应该以username@dbname的方式创建用户。 当一个username被连接着的客户端传递时，@和数据库名被增加到用户名中并且那个数据库相关的用户名会被服务器查找。注意，当你在 SQL 环境里创建包含@的用户名时， 你需要用引号包围用户名。

打开这个参数之后，你还是能够创建普通的全局用户。只要在客户端指定用户名时附加一个@，例如joe@。 在服务器查找这个用户名之前，这个@会被剥除。

db_user_namespace导致客户端和服务器的用户名表示变得不同。 认证检查总是使用服务器用户名来完成，因此认证方法必须为服务器的用户名配置，而不是客户端的用户名。因为在客户端和服务器上md5都使用用户名作为盐粒，md5不能和db_user_namespace一起使用。

数据访问控制

前言

概述

KingbaseES支持自主访问控制和强制访问控制。

术语定义

- "标记 "——标记是由等级和范围构成，并且，标记是可以比较的，其比较结果的含义是代表了数据的敏感程度。如果赋予用户某些标记，然后给数据再赋予标记，这样，将用户的标记和数据的标记进行比较，按照一定规则来决定用户是否可以对数据进行访问。
- "强制访问控制 "——强制访问控制（Mandatory Access Control——MAC），用于将系统中的信息分密级和类进行管理，以保证每个用户只能访问到那些被标明可以由他访问的信息的一种访问约束机制。通俗的来说，在强制访问控制下，用户（或其他主体）与文件（或其他客体）都被标记了固定的安全属性（如安全级、访问权限等），在每次访问发生时，系统检测安全属性以便确定一个用户是否有权访问该文件。

1. 自主访问控制

自主访问控制（DAC）作用是对主体（如用户）操作客体（如表）进行授权管理，简记为DAC。DAC主要包括权限授予，回收及传播。KingbaseES DAC采用存取控制列表（ACL）技术实现。

一旦一个对象被创建，它会被分配一个所有者。所有者通常是执行创建语句的角色。对于大部分类型的对象，初始状态下只有所有者（或者超级用户）能够对该对象做任何事情。为了允许其他角色使用它，必须分配权限。

有多种不同的权限：SELECT、INSERT、UPDATE、DELETE、TRUNCATE、REFERENCES、TRIGGER、CREATE、及USAGE。可以应用于一个特定对象的权限随着对象的类型（表、函数等）而不同。KingbaseES所支持的不同类型的完整权限信息请参考GRANT。下面的章节将简单介绍如何使用这些权限。

通常只有所有者才有修改或销毁一个对象的的权限。

一个对象可以通过该对象类型相应的ALTER命令来重新分配所有者，例如ALTER TABLE。超级用户总是可以做到这点，普通角色只有同时是对象的当前所有者（或者是拥有角色的一个成员）以及新拥有角色的一个成员时才能做同样的事。

要分配权限，可以使用GRANT命令。例如，如果joe是一个已有角色，而accounts是一个已有表，更新该表的权限可以按如下方式授权：

```
GRANT UPDATE ON accounts TO joe;
```

用ALL取代特定权限会把与对象类型相关的所有权限全部授权。

一个特殊的名为PUBLIC的"角色"可以用来向系统中的每一个角色授予一个权限。同时，在数据库中有很多用户时可以设置"组"角色来帮助管理权限。

为了撤销一个权限，使用REVOKE命令：

```
REVOKE ALL ON accounts FROM PUBLIC;
```

对象拥有者的特殊权限（即执行DROP、GRANT、REVOKE等的权力）总是隐式地属于拥有者，并且不能被授予或撤销。但是对象拥有者可以选择撤销他们自己的普通权限，例如把一个表变得对他们自己和其他人只读。

一般情况下，只有对象拥有者（或者超级用户）可以授予或撤销一个对象上的权限。但是可以在授予权限时使用"with grant option"来允许接收人将权限转授给其他人。如果后来授予选项被撤销，则所有从接收人那里获得的权限（直接或者通过授权链获得）都将被撤销。更多详情请见GRANT和REVOKE参考页。

2. 强制访问控制

KingbaseES支持标记和强制访问控制，保护用户数据，防止非法窃取。

强制访问控制（MAC）与DAC相比，MAC提供更严格和灵活的控制方式。MAC首先为所控制的主体和客体指派安全标记，然后依据这些标记进行访问仲裁。并且，只有主体标记能支配客体标记时才允许主体访问。

2.1 概念

KingbaseES的强制访问控制规则遵循简单保密模型，即"向下读，区间写"模型，实现信息流向总是向上的保密信息传递。下读规则是指用户只能读和自己等级一样或者等级比自己低的数据。区间写是上写模型，上写规则是指用户只能写入和自己等级一样或者等级比自己高的数据。

2.2 控制参数

参数enable_mac(boolean)控制强制访问控制是否打开，默认为false，表示不启用强制访问控制。

2.3 访问控制模型

在KingbaseES数据库中，数据能否被访问的仲裁结果取决于以下三个要素：数据标记、用户的会话标记和用户的特权。

2.3.1 数据标记

在KingbaseES中，安全管理员可以根据数据的具体密级为数据定义标记。标记包含两个元素：等级和范围

等级代表数据的敏感度，它关系到数据的安全性，用以防止未授权的用户查询和修改高密级的数据。即，拥有低等级的用户无法访问高密级的数据。例如，可以根据敏感度将数据分为如下等级：机密、秘密和普通。

范围用于将数据分类，例如，可以用范围将数据分为陆军信息、海军信息和空军信息。

标记可以是以下两种形式：

标记只包含等级元素，例如：“机密：”；

标记包含一个等级和范围集合，例如：“秘密:空军,陆军”。

2.3.2 授予权限

安全员可以给用户授予标记权限，以决定用户可以何种形式（读或写）访问那些数据。当表中的元组被标记标识后，只有用户被授予了访问该标记的权限后，可以读取到该元组或者写该元组。当元组具有多个数据标记时，用户必须具有所有标记的权限才可以访问该元组。

2.3.3 策略特权

策略下的特权允许用户绕开部分标记仲裁规则来访问数据。安全员可以给用户授予策略下的特权，使得用户可以不依靠会话标记对数据执行读写操作。

KingbaseES的强制访问控制规则遵循简单保密模型，即"向下读，区间写"模型，实现信息流向总是向上的保密信息传递。下读规则是指用户只能读和自己等级一样或者等级比自己低的数据。区间写是上写模型，上写规则是指用户只能写入和自己等级一样或者等级比自己高的数据。

2.4 策略特权

KingbaseES为一些特殊的操作提供特权，以提高数据访问控制的灵活性，下面列举了所有能够被授予可信用户的策略特权。

READ

当用户具有READ特权时，用户可以读取安全策略所保护的所有数据，此外拥有READ特权的用户仍然可以根据标记仲裁写相应的数据。

FULL

当用户具有FULL特权，用户可以读取安全策略所保护的所有数据，此外还可以写所有数据，KingbaseES对具有FULL特权的用户所进行的操作，不进行强制访问控制。

WRITEUP

WRITEUP特权允许用户提升数据标记的等级，但用户只能将等级提升到用户最大等级，并且不能修改数据的范围。例如用户的最大等级为秘密，用户不能将数据的等级提升到机密这个级别。

WRITEDOWN

WRITEDOWN特权允许用户降低数据标记的等级，但用户只能将等级降低到用户最小等级，并且不能修改数据的范围。

WRITEACROSS

WRITEACROSS特权允许用户修改数据标记的范围，但不能修改数据标记的等级。此外新的范围必须仍然是用户最大写范围的子集，并且在上写规则中，用户最小写范围必须是新范围的子集。

2.5 具体实现

2.5.1 策略管理

强制访问控制--策略管理

```
注意：1).设置ENABLE_MAC = on；2).只有安全员可以操作策略相关语句
```

创建策略

语法格式：

```
CREATE_POLICY (policy_name NAME,column_name NAME,
hide_column BOOL);
```

功能：

创建新的安全策略，定义策略列的名称，并指定策略选项。默认情况下，策略创建后都处于启用状态.

参数说明：

policy_name：指定策略的名称，该名称必须在数据库中唯一。

column_name：当策略应用于表上时，将自动在表上增加一列，该参数就是用于指定策略列的名称。该名称必须在数据库中唯一。

hide_column：指定当策略用于表上时，是否隐藏策略列。TRUE代表隐藏，FALSE代表不隐藏。

示例：

```
CALL CREATE_POLICY ('PA','PA_C',FALSE);
```

禁用策略

语法格式：

```
DISABLE_POLICY(policy_name NAME);
```

功能：

禁用安全策略，而不是将策略从表上移除。策略禁用后，系统将不再对受策略保护的表实施访问控制。

参数说明：

policy_name：指定要禁用的安全策略名称。

示例：

```
CALL DISABLE_POLICY ('PA');
```

启用策略

语法格式：

```
ENABLE_POLICY(policy_name NAME);
```

功能：

启用安全策略。策略启用后，系统将继续对受策略保护的表实施访问控制。

参数说明：

policy_name：指定需要被启用的安全策略名称。

示例：

```
CALL ENABLE_POLICY('P1');
```

删除策略

语法格式：

```
DROP_POLICY (policy_name NAME,drop_column BOOL);
```

功能：

删除数据库中的安全策略，并同时删除该策略下的所有数据标记和用户会话标记。此外用户可以指定是否删除受到该安全策略保护的表中的策略列。

参数说明:

policy_name：指定要删除的安全策略名称。

drop_column：指定是否删除受到安全策略保护的表中的策略列。TRUE代表删除，FALSE代表不删除。

示例：

```
CALL DROP_POLICY('P1');
```

2.5.2 标记元素管理

强制访问控制--标记元素管理

```
注意：1).设置ENABLE_MAC = on; 2).只有安全员可以操作标记元素相关语句
```

创建等级

语法格式：

```
CREATE_LEVEL(policy_name NAME, level_name NAME, levid SMALLINT)
```

功能：

创建指定策略名下的等级，并指定等级的ID。用户指定的ID值的大小决定了等级的敏感性级别，数值越大级别越高。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

level_name：指定等级的名称，等级名称在策略下唯一。

levid：指定等级的ID值，等级ID的取值范围为[1,9999]，并在策略下唯一。

示例：

```
CALL CREATE_LEVEL('P1','L1',10);
```

删除等级

语法格式：

```
DROP_LEVEL(policy_name NAME, level_name NAME)
DROP_LEVEL(policy_name NAME, levid SMALLINT)
```

功能：

删除指定策略下的等级，当等级被标记引用时，不能删除该等级。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

level_name：指定等级的名称。

levid：指定等级的ID值。

示例：

```
CALL DROP_LEVEL ('P1', 'L1');
CALL DROP_LEVEL ('P1', 10);
```

创建范围

语法格式：

```
CREATE_COMPARTMENT (policy_name NAME,comp_name NAME,comp_id
SMALLINT)
```

功能：

创建指定策略下的范围，并指定范围ID。用户指定的范围ID值决定了范围在标记字符串中出现的先后顺序。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

comp_name：指定范围的名称，范围名称在策略下唯一。

comp_id：指定范围的ID值，范围ID的取值范围为[1,9999]，并在策略下唯一。

示例：

```
CALL CREATE_COMPARTMENT ('P1','PA',100);
```

删除范围

语法格式：

```
DROP_COMPARTMENT(policy_name NAME,comp_name NAME)
DROP_COMPARTMENT(policy_name NAME,comp_id SMALLINT)
```

功能：

删除指定策略下的范围，当范围被标记引用时，不能删除该范围。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

comp_name：指定范围的名称。

comp_id：指定范围的ID值。

示例：

```
DROP_COMPARTMENT('P1', 'PA');
DROP_COMPARTMENT('P1', 100);
```

2.5.3 标记管理

强制访问控制--标记管理

注意：1).设置ENABLE_MAC = on；2).只有安全员可以操作标记相关语句

创建标记

语法格式：

```
CREATE_LABEL(policy_name NAME, label TEXT,labelid INT)
```

功能：

创建指定策略下的标记，并指定标记ID。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

label：指定标记字符串，关于标记的字符串的语法格式请参见“标记的语法规定”小节。

labelid：指定标记的ID值，标记ID的取值范围是[1,99999999]，并且必须是同一数据库下唯一。

示例：

```
CALL CREATE_LABEL('P1','L1:C4',11);
```

修改标记

语法格式：

```
ALTER_LABEL(policy_name NAME, label TEXT,newlabel TEXT)
ALTER_LABEL(policy_name NAME, labelid INT, newlabel TEXT)
```

功能：

修改指定策略下的标记。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

label：指定要修改的标记字符串。

labelid：指定要修改的标记ID。

newlabel：指定新的标记字符串。

示例：

```
CALL ALTER_LABEL('P1','L1:C4','L1:C5');
CALL ALTER_LABEL('P1',11,'L1:C5');
```

删除标记

语法格式：

```
DROP_LABEL(policy_name NAME, label TEXT)
DROP_LABEL(policy_name NAME,labelid INT)
```

功能：

删除指定策略下的标记。删除标记后，用该标记ID标识的数据标记都将视为无效的标记ID。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

label：指定要删除的标记字符串。

labelid：指定要删除的标记ID。

示例：

```
CALL DROP_LABEL('P1','L1:C4');
CALL DROP_LABEL('P1',11);
```

标记ID与标记的转换

语法格式：

```
VARCHAR LABEL_TO_CHAR (labelid INT);
INT CHAR_TO_LABEL (policy_name NAME,label_string TEXT);
```

功能：

函数LABEL_TO_CHAR通过参数标记ID返回对应的标记字符串，CHAR_TO_LABEL通过策略名称和标记字符串返回标记ID。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

label_string：指定要转换的标记字符串。

labelid：指定要转换的标记ID。

示例：

```
SELECT A,P1_C,LABEL_TO_CHAR(P1_C) FROM T1;
SELECT A,P1_C,LABEL_TO_CHAR('P1', 'L1:C4') FROM T1;
```

2.5.4 用户标记和特权管理

强制访问控制--用户标记和特权管理

安全员为用户授予等级

语法格式：

```
SET_LEVELS(
policy_name TEXT,
user_name TEXT,
max_level TEXT,
min_level TEXT,
def_level TEXT,
row_level TEXT
 )
```

功能：

指定用户的最大等级和最小等级，以及用户登陆数据库时默认的会话等级和写入等级。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

max_level：指定读写访问的最高等级。

min_level：指定写访问的最低等级，min_level必须小于等于max_level。如果min_level为NULL，那么系统自动将其设置为指定策略下的最低等级。

def_level：指定用户登录系统的默认会话等级。def_level必须小于等于 row_level，如果def_level为NULL，那么系统自动将其值设置为row_level

row_level：指定用户的写入等级。row_level必须小于等于max_level， 如果row_level为NULL，那么系统自动将其值设置为max_level。

示例：

```
CALL SET_LEVELS('P1','U1','L5','L2','L4','L3');
```

安全员为用户授予范围

语法格式：

```
SET_COMPARTMENTS(
```

```
    policy_name NAME,
    user_name NAME,
    read_compartments TEXT,
    max_write_compartments TEXT,
    defaut_compartments TEXT,
    row_compartments TEXT,
    min_write_compartments TEXT
    )
```

功能：

指定用户的读写范围，以及用户登陆数据库时默认的会话范围和写入范围。安全员在给用户授予范围之前必须先给用户授予等级。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

read_compartments：指定用户具有读权限的范围列表，范围之间以逗号分隔。

max_write_compartments：指定用户具有最大写权限的范围列表，范围之间以逗号分隔。最大写范围集合必须是读范围集合的子集。 如果max_write_compartments为NULL，那么系统自动将其值设置为read_compartments。

defaut_compartments：指定用户登录系统后默认的具有读权限的范围列表，范围之间以逗号分隔。默认的范围集合必须是默认写入范围集合的子集。如果defaut_compartments为NULL，那么系统自动将其设置为row_compartments。

row_compartments：指定默认的写入范围列表，范围之间以逗号分隔。写入范围集合必须是最大写范围集合的子集。如果row_compartments为NULL，那么系统自动将其值设置为max_write_compartments。

min_write_compartments：指定用户具有最小写权限的范围列表，范围之间以逗号分隔。最小写范围集合必须是默认写入范围集合的子集。如果min_write_compartments为NULL，那么最小写范围集合为空，系统不会自动设置其值。

示例：

```
CALL SET_COMPARTMENTS('P1','U1','C1,C2,C3,C4,C5','C1,C2,C3',
'C1,C2,C4', 'C1,C2');
```

安全员为用户授予标记

语法格式：

```
    SET_USER_LABELS(
```

```
    policy_name NAME,
    user_name NAME,
    max_read_label TEXT,
    max_write_label TEXT,
    min_write_label TEXT,
    def_label TEXT,
    row_label TEXT
    )
```

功能：

通过给用户授予会话标记，来设置用户的等级和范围。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

max_read_label：指定用户的最大读标记，该标记包含授予用户的最大等级和具有读权限的范围集合。

max_write_label：指定用户的最大写标记，该标记包含授予用户的最大等级和具有写权限的范围，该参数必须满足如下条件：
a) max_write_label.level= max_read_label.level。
b) max_write_label.comps为 max_read_label.comps的子集。
如果该参数为空，那么系统自动将其值设置为max_read_label。

row_label：指定用户的写入标记，其值必须满足如下条件：
a) row_label.level <= max_write_label.level。
b) row_label.level >= min_write_label.level。
c) row_label.comps为max_write_label.comps的子集。
如果该参数为空，那么系统将其值设置为max_write_label；

min_write_label：指定用户的最小写标记，该标记包含授予用户的最小等级和最小写范围，该参数必须满足如下条件：
a) min_write_label.level <= row_label.level。
b) min_write_label.comps为row_label.comps的子集。
如果该参数为空，那么系统将其值设置为策略内的最小等级。

def_label：用户的默认会话标记，其值必须满足如下条件：
a) def_label.level <= row_label.level。
b) def_label.comps为 row_label.comps的子集。
如果该参数为空，那么系统将其值设置为row_label。

示例：

```
CALL SET_USER_LABELS('P1', 'U1', 'L5:C1,C2,C3,C4,C5', 'L5:C1,C2,C3',
```

```
'L2:', 'L4:C1,C2,C4', 'L3:C1,C2');
```

安全员为用户设置默认标记

语法格式：

```
SET_DEFAULT_LABEL(
policy_name NAME,
username NAME,
label TEXT
)
```

功能：

设置用户登录系统时默认的会话标记。安全员在给用户设置默认标记之前必须先给用户授予标记。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

label：设置用户登陆系统时的初始化会话标签。 其值必须满足如下条件：
a) label.level <= 用户的写入标记的等级。
b) label.comps集合是写入标记范围集合的子集。

示例：

```
CALL SET_DEFAULT_LABEL('P1',U1,'L1:C4');
```

安全员为用户设置默认写入标记

语法格式：

```
SET_DEF_ROW_LABEL(
policy_name NAME,
user_name NAME,
label TEXT
)
```

功能：

设置用户登录系统时默认的写入标记。安全员在给用户设置默认写入标记之前必须先给用户授予标记。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

label：指定用户用户的写入标记。其值必须满足如下条件：
a) label.level >= 用户的最小等级。
b) label.level <= 用户的最大等级。
c) label.comps集合是最大写范围的子集。
此外，该值必须仍然可以支配用户的默认标记和最小写标记，并满足如下条件：
a) label.level >= 用户默认标记的等级。
b) label.level >= 用户最小写标记的等级。
c) 默认标记的范围集合仍然是label.comps集合的子集。
d) 最小写标记的范围集合仍然是label.comps集合的子集。

示例：

```
CALL SET_DEF_ROW_LABEL('P1',U1,'L1:C4');
```

安全员为用户授予特权

语法格式：

```
SET_USER_PRIVS (
policy_name NAME,
user_name  NAME,
privileges TEXT
)
```

功能：

为用户授予系统预设的策略特权。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

privileges：指定用户的特权列表，权限间由“,”隔开。当privilege为NULL时，相当于回收用户的所有策略特权。

示例：

```
CALL SET_USER_PRIVS('P1','U2','FULL');
```

安全员回收用户的所有权限

语法格式：

语法格式：

```
DROP_USER_ACCESS(
policy_name NAME,
user_name NAME
)
```

功能：

收回用户指定策略下所有的权限和特权。

参数说明：

policy_name：指定数据库中已经创建的安全策略名称。

user_name：指定用户名称。

示例：

```
CALL DROP_USER_ACCESS('P1',U1);
```

数据访问保护

KingbaseES支持数据页面的一致性保护，同时支持用户使用加密函数保护数据。

1. 数据页面一致性

数据页面包括CRC校验码，在读数据时，首先完成一致性校验。若发现问题，及时报错，阻止错误蔓延，阻止错误升级。

KingbaseES 提供配置参数zero_damaged_pages，若设置为true，在读取数据发现页面错误时，服务器将页面初始化为初始状态，继续处理。这个设置适用于对数据一致性要求不高，允许数据部分丢失的场合。

2. 加密函数

KingbaseES 提供了典型的加密函数，用户可以使用加密函数存储关键敏感数据。支持的加密函数参见下表。

表 2-1. 使用和不用 OpenSSL 的功能总结

功能	内建	使用 OpenSSL
MD5	yes	yes
SHA1	yes	yes
SHA224/256/384/512	yes	yes (注意 a)
Blowfish	yes	yes
AES	yes	yes (注意 c)
DES/3DES/CAST5	no	yes
SM4	yes	no
RC4	yes	no

加密示例

SM4/rc4加密示例，支持使用sm4()和sm4_ex()函数进行SM4的加/解密运算。
sm4_ex()在sm4()函数的基础上支持可选填充方式。

```
create extension kbcrypto;
set bytea_output to escape;
```

1. sm4()函数/rc4()函数

参数分别为：加/解密数据；密钥；加密/解密标识。

```
-- 加密：
select sm4('123456abcdef','0123456789ABCDEF',0);
select rc4('123456abcdef','0123456789ABCDEF',0);
-- 解密
select sm4(sm4('123456abcdef','0123456789ABCDEF',0), '0123456789ABCDEF',1);
select rc4(rc4('123456abcdef','0123456789ABCDEF',0), '0123456789ABCDEF',1);
```

2. sm4_ex()函数（rc4没有ex函数扩展）

参数分别为：加/解密数据；密钥；加密/解密标识；填充模式。

1）填充模式选1，数据按16字节倍数强制填充，缺m个字节则填充m个字节的m值(m最大值为16)。

```
-- 加密：
select sm4_ex('123456abcdef','0123456789ABCDEF',0,1);
-- 解密
select sm4_ex(sm4_ex('123456abcdef','0123456789ABCDEF',0,1), '0123456789ABCDEF',1
```

2）填充模式选0，数据按16字节倍数非强制填充0x0，同sm4。

```
-- 加密
select sm4_ex('123456abcdef','0123456789ABCDEF',0,0);
-- 解密
select sm4_ex(sm4_ex('123456abcdef','0123456789ABCDEF',0,0), '0123456789ABCDEF',1
```

注意：

a. SHA2 算法是在版本 0.9.8 时被加入到 OpenSSL 的。

b. OpenSSL 支持的任何摘要算法都是自动选取的。这对于使用密码来说是不可能的，因为需要被显式地支持。

c. AES 从版本 0.9.7 就开始被包括在 OpenSSL 中。

数据安全传输

KingbaseES支持通过SSL协议实现客户端和服务器之间的安全数据传输，使得数据在传输过程中难以被窃听、篡改、重放和伪造。

1. SSL 支持

KingbaseES本地支持使用SSL连接加密客户端/服务器通信以提高安全性。

libkci读取系统范围的OpenSSL配置文件。默认情况下，这个文件被命名为`openssl.cnf`并且存放在`openssl version -d`报告的目录中。可以通过设置环境变量`OPENSSL_CONF`把这个默认值覆盖为想要的配置文件的名称。

1.1. 服务器证书的客户端验证

默认情况下，KingbaseES将不会执行服务器证书的任何验证。这意味着可以在不被客户端知晓的情况下伪造服务器身份（例如通过修改一个 DNS 记录或者接管服务器的 IP 地址）。为了阻止哄骗，必须使用SSL证书验证。

如果参数`sslmode`被设置为`verify-ca`，libkci 将通过检查证书链一直到一个可信的证书机构（CA）来验证服务器是可信的。如果`sslmode`被设置为`verify-full`，libkci 将还会验证服务器主机名是否匹配它的证书。如果服务器证书不能被验证，SSL 连接将失败。在大部分对安全敏感的环境中，建议使用`verify-full`。

在`verify-full`模式下，主机名与证书的Subject Alternative Name（主题备用名称)属性进行匹配，或者如果没有类型为dNSName的主题备用名称，则与Common Name（公用名称）属性进行匹配。如果证书的名称属性以星号（ * ）开头，则星号将被视为通配符，其将匹配除了点(.)之外的所有字符。这意味着证书将不会匹配子域。如果使用IP地址而不是主机名进行连接，则将匹配IP地址（不进行任何DNS查找）。

要允许服务器证书验证，一个或多个可信的CA必须被放置在用户home目录下的文件`~/.kingbase/root.crt`中。如果中间CA出现在`root.crt`中，该文件必须也包含到它们的根CA的证书链 （在微软 Windows 上该文件被命名为`%APPDATA%\kingbase\root.crt`）。

如果文件`~/.kingbase/root.crl`存在 （微软 Windows 上的`%APPDATA%\kingbase\root.crl`）， 也会检查证书撤销列表（CRL）项。

根证书文件和 CRL 的位置可以通过设置连接参数sslrootcert和 sslcrl或环境变量KCISSLROOTCERT和KCISSLCRL改变。

> **注意:** 为了与 KingbaseES 的早期版本达到向后兼容，如果存在一个根 CA 文件，sslmode=require的行为将与 verify-ca相同，意味着服务器证书根据 CA 验证。不鼓励依赖这种行为，并且需要证书验证的应用程序应该总是使用verify-ca或者verify-full。

1.2. 客户端证书

如果服务器要求一个可信的客户端证书，libkci将发送用户主目录中~/.kingbase/kingbase.crt文件存储的证书。该证书必须由一个受服务器信任的证书机构（CA）签发。也必须存在一个匹配的私钥文件~/.kingbase/kingbase.key。该私钥文件不允许全局或组用户的任何访问，可以通过命令chmod 0600 ~/.kingbase/kingbase.key实现。在微软 Windows 上这些文件被命名为%APPDATA%\kingbase\kingbase.crt和%APPDATA%\kingbase\kingbase.key，不会有特别的权限检查，因为该目录被假定为安全。证书和密钥文件的位置可以使用连接参数sslcert和sslkey或者环境变量KCISSLCERT和KCISSLKEY覆盖。

在一些情况下，客户端证书可以由"中间"证书机构签名，而不是由服务器直接信任的证书机构。要使用这样一个证书，将签发机构的证书加入到kingbase.crt文件，然后是它的上级机构的证书，并且一直到一个受服务器信任的证书机构（"根"机构或者"中间"机构），即由该服务器的root.crt文件中的一个证书签发。

注意客户端的~/.kingbase/root.crt列出了被认为可信的能用于签发服务器证书的顶层 CA。原则上不需要列出签发客户端证书的 CA，大部分情况下这些 CA 也被信任可以用于服务器证书。

1.3. 不同模式中提供的保护

sslmode参数的不同值提供了不同级别的保护。SSL 能够针对三类攻击提供保护：

窃听

如果一个第三方能够检查客户端和服务器之间的网络流量，它能读取连接信息（包括用户名和口令）以及被传递的数据。SSL使用加密来阻止这种攻击。

中间人（MITM）

如果一个第三方能对客户端和服务器之间传送的数据进行修改，它就能假装是服务器并且因此能看见并且修改数据，即使这些数据已被加密。然后第三方可以将连接信息和数据传送给原来的服务器，使得它不可能检测到攻击。这样做的常用载体包括

DNS 中毒和地址劫持，借此客户端被定向到预期之外的不同的服务器。还有几种其他的攻击方式能够完成这种攻击。SSL使用证书验证让客户端认证服务器，就可以阻止这种攻击。

模仿

如果第三方可以伪装成一个授权的客户端，那么它能够轻松访问它本不能访问的数据。通常这可以由不安全的口令管理所致。SSL使用客户端证书来阻止这种情况，即确保只有持有合法证书的客户才能访问服务器。

对于一个已知安全的连接，在连接被建立之前，必须在客户端和服务器端都进行SSL配置。如果只在服务器上配置，客户端在知道服务器要求高安全性之前可能会结束发送敏感信息（例如口令）。在 libkci 中，可以通过将`sslmode`参数设置为`verify-full`或`verify-ca`来确保安全连接，并且为系统提供一个根证书用来验证。这类似于使用`https` URL进行加密网页浏览。

一旦服务器已经被认证，客户端可以传递敏感数据。这意味着直到这一点，客户端都不需要知道是否证书将被用于认证，这样只需要在服务器配置中指定就比较安全。

所有SSL选项都带来了加密和密钥交换的开销，因此必须在性能和安全性之间做出平衡。表 1-1说明不同`sslmode`值所保护的风险，以及关于安全和开销所做出的声明。

表 1-1. SSL 模式描述

sslmode	窃听保护	MITM保护	声明
`disable`	否	否	我不关心安全性，并且我不想承担加密的开销。
`allow`	可能	否	我不关心安全性，但如果服务器坚持，我会承担加密开销 。
`prefer`	可能	否	我不关心加密，但如果服务器支持，我希望承担加密开销。
`require`	是	否	我希望我的数据加密，我接受开销。我相信该网络将确保我始终连接到想要连接的服务器。
`verify-ca`	是	取决于CA-策略	我希望我的数据加密，我接受开销。我想要确保我连接到的是我信任的服务器。
`verify-full`	是	是	我希望我的数据加密，我接受开销。我想要确保我连接到的是我信任的服务器，并且就是我指定的那一个。

`verify-ca`和`verify-full`之间的区别取决于根CA的策略。如果使用了一个公共CA，`verify-ca`允许连接到那些可能已经被其他人注册到该CA的服务器。在这种情况下，总是应该使用`verify-full`。如果使用了一个本地CA或者甚至是一个自签名的证书，使用`verify-ca`通常就可以提供足够的保护。

`sslmode`的默认值是`prefer`。如表中所示，从安全角度来看这样做是没有意义的，并且它只承诺可能的性能开销。提供它作为默认值只是为了向后兼容，在安全部署中不建议使用。

1.4. SSL 客户端文件使用

表 1-2总结了与客户端 SSL 设置相关的文件。

表 1-2. Libkci/客户端 SSL 文件用法

文件	内容	影响
~/.kingbase/kingbase.crt	客户端证书	由服务器要求
~/.kingbase/kingbase.key	客户端私钥	证明由所有者发送客户端证书，并不表示证书所有者是可信的
~/.kingbase/root.crt	受信任的证书颁发机构	检查服务器证书是由一个可信的证书机构签发
~/.kingbase/root.crl	被证书颁发机构撤销的证书	服务器证书必须不在这个列表中

1.5. SSL 库初始化

如果初始化应用libssl或libcrypto库以及libkci编译来支持SSL，应该调用KCIinitOpenSSL来告诉libkci：libssl或libcrypto库已经被你的应用初始化，这样libkci将不会再初始化这些库。

KCIinitOpenSSL

允许应用选择要初始化哪个安全库。

```
void KCIinitOpenSSL(int do_ssl, int do_crypto);
```

当do_ssl是非零时，libkci 将在第一次打开数据库连接前初始化OpenSSL库。当do_crypto是非零时，libcrypto库将被初始化。默认情况下（如果没有调用KCIinitOpenSSL），两个库都会被初始化。当 SSL 支持没有被编译时，这个函数也存在但是什么也不做。

如果你的应用使用并且初始化OpenSSL或者它的底层libcrypto库，必须在第一次打开数据库连接前以合适的非零参数调用这个函数。同时要确保在打开一个数据库连接前已经完成了初始化。

KCIInitializeSSLEnv

允许应用选择要初始化哪个安全库。

```
void KCIInitializeSSLEnv(int do_ssl);
```

这个函数等效于KCIinitOpenSSL(do_ssl, do_ssl)。这对于要么初始化OpenSSL以及libcrypto要么都不初始化的应用足够用了。

在KingbaseES V8R2 中已经支持 KCIInitializeSSLEnv因此KCIInitializeSSLEnv可能对那些需要与旧版本libkci一起工作的应用来说更合适。

透明存储加密

前言

概述

KingbaseES提供的数据透明加密保护,保护存储在磁盘中的数据不被非法窃取。

术语定义

- "数据加密 "——在密码学中，加密是将明文信息隐匿起来，使之在缺少特殊信息时不可读。在网络通讯中,为了保证数据传输和存储的保密性，用密钥对文件进行加密变换，使之成为不可识别的格式。根据密钥类型不同可以将现代密码技术分为两类：对称加密技术和非对称加密技术。
- "透明加密 "——透明加密是指数据在写到磁盘上时对其进行加密，当授权用户重新读取数据时再对其进行解密。无需对应用程序进行修改，授权用户甚至不会注意到数据已经在存储介质上加密，加密解密过程对用户都是透明的。

1 透明存储加密

1.1 概念

透明存储加密是指数据在写到磁盘上时对其进行加密，当授权用户重新读取数据时再对其进行解密。无需对应用程序进行修改，授权用户甚至不会注意到数据已经在存储介质上加密，加密解密过程对用户都是透明的。

1.2 级别

透明存储加密支持表级加密，指定的表中的所有数据进行加密。当索引建立在加密表或加密列上时，该索引将隐式被加密，且该索引也属于表级加密。

1.3 算法

透明存储加密支持RC4算法对数据进行加密。

1.4 密钥管理

透明存储加密使用的密钥随机生成，每个表使用的密钥不同，密钥保存在系统表 SYS_CLASS 中。

1.5 语法

透明存储加密在创建表时指定，具体参考 CREATE TABLE 语法。

数据库审计

1. 审计概述

对数据库系统中发生的动作（或称为事件），将其对应的操作对象、操作时间等信息记录下来的过程，称为审计。

任何系统的安全保护措施都不是完美无缺的，蓄意盗窃，破坏数据的人总是想方设法打破控制，审计功能将用户对数据库的所有操作自动记录下来放入审计日志中，审计员（SAO）可以通过对审计日志的分析，对潜在的威胁提前采取有效地措施加以防范。KingbaseES数据库提供了一套完整的审计机制，用来保证对数据库中的各种行为进行监控，进而为数据库的安全、可靠和有效提供有力的保障。

2. 审计开关

KingbaseES数据库中对审计功能设置了开关。审计功能关闭时，数据库不会发生审计动作，也无法将审计设置应用到数据库中。审计员（sao）如果要使 KingbaseES 系统中的审计设置生效，必须打开该审计开关。

`sysaudit.enable`

设置审计功能是否开启的总开关，有on和off两种选择，缺省为 on。

- on表示打开审计功能；
- off表示关闭审计功能；

打开（或关闭）审计开关的方式有如下两种方式：

- 在KingbaseES启动之前，通过修改 kingbase.conf 文件，将 `sysaudit.enable`设置为on（或off）
- 在KingbaseES启动后，以审计员用户连接数据库，利用ALTER SYSTEM命令将 `sysaudit.enable` 设置为on|off。打开（或关闭）审计开关，重载配置后，数据库系统中的审计功能立即生效（或失效）。
  打开（或关闭）审计开关并重载的命令：

```
ALTER SYSTEM SET sysaudit.enable = on | off;
CALL sys_reload_conf();
```

3. 审计设置

KingbaseES的审计设置分为三个级别：服务器级别审计、语句级别审计、模式对象级别审计。

- 服务器级别审计：审计数据库服务器级别发生的事件，包含以下三种：数据库服务器的启动、数据库服务器的停止、数据库服务器的配置文件重新加载。简称为：服务器级审计或服务器审计。
- 语句级别审计：也称为 "STATEMENT AUDITING"，指在 DBMS 范围内，对DBMS拥有的结构或模式对象进行操作时引发的事件进行审计，此类结构或模式对象并不指具体的某个结构或模式对象，而是一类结构或模式对象的泛称。通常，包括 DBMS 提供的 DDL、DML、DQL、DCL、TCL 语句部分引发的事件。简称为：语句级审计或语句审计。
- 模式对象级别审计：也称为 "SCHEMA OBJECT AUDITING"，指在某个确定的模式对象上进行 SELECT 或 DML 操作时引发的事件进行审计。模式对象包括表、视图、存储过程、函数、序列、包。模式对象不包括有依附关系的对象，如依附于表的索引、约束、触发器、分区表等。简称为：模式对象级审计或对象审计。

  KingbaseES的审计参数设置如下：

  `sysaudit.systemobjects`

  是否审计系统对象，默认false。

  `sysaudit.serverevent`

  服务器事件审计开关，包括：服务器启动、关闭、重载配置，服务器启动时设置，默认true。

  `sysaudit.userevent`

  用户事件审计开关，包括：客户端登录和退出，默认true。

  `sysaudit.enable`

  设置审计功能是否开启的总开关，有on和off两种选择，缺省为 off。

  `sysaudit.syntaxerror`

  语言错误审计开关，设置是否审计发生语法错误的语句，默认off 此类错误也包括：事务块中发生错误之后，未经语法分析而直接报错的语句。

  `sysaudit.semanticerror`

  SELECT类/DML类语句的语义错误审计开关，设置是否审计进入执行阶段之前发生的语义错误，默认true。

  `sysaudit.parameters`

审计参数，是否在审计结果中输出Prepared语句的参数，默认false。

sysaudit.audit_table_hostaddr

存储审计日志的数据库的IP地址，默认为本机。

sysaudit.audit_table_port

存储审计日志的数据库的端口。

sysaudit.audit_table_user

存储审计日志的数据库的用户，默认为审计管理员sao。

sysaudit.audit_table_password

存储审计日志的数据库的用户密码。

sysaudit.local_sao_password

数据库的审计管理员sao用户密码。

3.1. 服务器审计

只要打开审计开关，总是在审计中记录发生的服务器级别的事件，不必进行审计设置。

3.2. 语句审计

启用语句审计：

```
bigint set_audit_stmt(audit_type text,
audit_users text,
audit_schema text,
audit_objs text
```

参数说明：

audit_type

语句级审计策略，支持DML、DDL、TCL语句等。

既可以指定语句分组，审计此分组内的所有SQL命令，也可以指定审计某一种SQL命令。

详见*表13-2 审计策略与SQL命令对应表*。

audit_users

审计的用户名，null表示审计所有用户。

audit_schema

审计对象的模式名。可以为空，表示审计所有模式下的此类对象。

audit_objs

审计对象的名称，只有select table、insert table、update table、delete table、truncate table、drop table这六种语句支持设置表名。

返回值说明：

无返回值

取消审计：

```
void remove_audit (rule_id bigint)
```

参数说明：

rule_id

审计策略编号（创建后可从all_audit_rules中查询）

语句审计示例：

审计system用户的表创建动作。

```
SELECT set_audit_stmt('CREATE TABLE', 'system', null, null);
```

审计用户user_a的表修改动作

```
SELECT  set_audit_stmt('alter table', 'user_a', null, null);
```

取消编号为10001的审计策略。

```
CALL remove_audit (10001);
```

3.3. 对象审计

模式对象级审计发生在具体的对象上的DML/SELECT操作，需要指定模式名及其对象名。只有对相应对象执行正确的设置才会触发审计，比如：为函数对象设置一个INSERT的审计策略将无法生效，因为数据库并不支持INSERT到一个函数。

启用对象审计：

```
bigint set_audit_object(audit_type text,
audit_users text,
```

```
audit_schema text,
audit_objs text)
```

参数说明：

audit_type

审计对象的类型，支持：TABLE、VIEW、MATERIALIZED VIEW、PROCEDURE、FUNCTION，大小写不敏感。

audit_users

审计的用户名，null表示审计所有用户。

audit_schema

审计对象的模式名。可以为空，表示审计所有模式下的此类对象。

audit_objs

审计对象的名称，比如为表名、视图名、存储过程名。可以为空，表示审计指定模式下的所有此类对象。

返回值说明：

无返回值。

取消对象审计：

```
void remove_audit (rule_id bigint)
```

对象审计示例：

审计 user1 用户对 public 模式下的表 t1 的任意操作。

```
SELECT set_audit_object('table', 'user1', 'public', 't1');
```

取消id为10002的审计设置。

```
CALL remove_audit (10002);
```

3.4. 审计设置查询

审计策略设置可通过系统视图 all_audit_rules 查询，结构如下表所示：

列名	数据类型	说明
audit_id	bigint	审计策略编号
audit_target	smallint	审计目标（事件、语句、对象）
audit_type	text	审计类型
audit_users	text	审计用户
audit_schema	text	审计对象所在模式
audit_objnames	text	审计对象名称
audit_objoid	oid	审计对象id
creator_name	text	策略设置者名

4. 审计信息存储

审计信息存储在数据库SECURITY中，审计管理员sao可通过视图sysaudit_record_sao查看，安全管理员sso可通过视图sysaudit_record_sso查看。

5. 审计记录查询

审计管理员sao或安全管理员sso通过视图查询审计记录。

例13-1. 视图结构如下表

列名	数据类型	说明
session_id	text	会话id
proc_id	int	进程号
vxid	text	虚事务id
xid	text	事务id
user_id	oid	用户id
username	text	用户名
remote_addr	text	ip地址/端口
db_id	oid	数据库id
db_name	text	数据库名
rule_id	bigint	审计策略编号
rule_type	text	审计类型
opr_type	text	操作类型
obj_type	text	对象类型
schm_id	oid	模式id
schm_name	text	模式名
obj_id	oid	对象id
obj_name	text	对象名
sqltext	text	SQL文本
params	text	参数
errcode	text	SQLSTATE错误码
errmsg	text	错误消息内容
audit_ts	timestamp with time zone	操作时间
result	text	操作成功是success，失败是failure
record_type	smallint	审计记录类型（预留）
aud_client	text	客户端名字
server_type	text	集群时服务器类型，M是主机，S是备机

表13-2 审计策略与SQL命令对应表（括号内为缩写）

具体的 SQL 命令内容可参考*SQL 语句参考手册*

更改

分组名称	SQL命令	说明
ALTER	ALTER TABLE	更改表定义
	ALTER INDEX	更改索引定义
	ALTER VIEW	更改视图定义
	ALTER MATERIALIZED VIEW	更改物化视图定义
	ALTER SEQUENCE	更改序列定义
	ALTER FUNCTION	更改函数定义
	ALTER PROCEDURE	更改过程定义
COPY	COPY FROM	从文件中向表复制数据
	COPY TO	从表或查询向文件复制数据
CREATE	CREATE TABLE	定义表
	CREATE INDEX	定义索引
	CREATE VIEW	定义视图
	CREATE MATERIALIZED VIEW	定义物化视图
	CREATE SEQUENCE	定义序列
	CREATE FUNCTION	定义函数
	CREATE PROCEDURE	定义过程
DML	DELETE TABLE	删除记录
	INSERT TABLE	插入记录
	UPDATE TABLE	更新记录
	DROP TABLE	移除表
	DROP INDEX	移除索引
	DROP VIEW	移除视图

分组名称	SQL命令	说明
DROP	DROP MATERIALIZED VIEW	移除物化视图
	DROP SEQUENCE	移除序列
	DROP FUNCTION	移除函数
	DROP PROCEDURE	移除过程
FUNCTION	CREATE FUNCTION	定义函数
	ALTER FUNCTION	更改函数定义
	DROP FUNCTION	移除函数
INDEX	CREATE INDEX	定义索引
	ALTER INDEX	更改索引定义
	DROP INDEX	移除索引
MATERIALIZED VIEW	CREATE MATERIALIZED VIEW	定义物化视图
	ALTER MATERIALIZED VIEW	更改物化视图定义
	DROP MATERIALIZED VIEW	移除物化视图
PROCEDURE	CREATE PROCEDURE	创建过程
	ALTER PROCEDURE	更改过程定义
	DROP PROCEDURE	移除过程
SELECT	SELECT TABLE	查询语句。从数据库中检索数据的过程或命令叫做查询。在SQL 语句里SELECT命令用于指定查询。

分组名称	SQL命令	说明
SEQUENCE	CREATE SEQUENCE	定义序列
	ALTER SEQUENCE	更改序列定义
	DROP SEQUENCE	移除序列
SYSTEM	ALTER SYSTEM	更改服务器配置参数
TABLE	CREATE TABLE	定义表
	ALTER TABLE	更改表定义
	DROP TABLE	移除表
	TRUNCATE	清空一个或者一组表
TRANSACTION	BEGIN	开始事务块
	COMMIT	提交当前事务
	ROLLBACK	中止当前事务
	SAVEPOINT	在当前事务中定义保存点
	ROLLBACK TO	回滚到指定保存点
	RELEASE	释放指定的保存点
	START TRANSACTION	开始事务块
VIEW	CREATE VIEW	定义视图
	ALTER VIEW	更改视图定义
	DROP VIEW	移除视图

注意

> 数据库在审计过程中，需要先读取审计规则，之后使用审计规则判断，才能生成审计记录。如果在读取审计规则前，就出错，可能无法生成审计记录。

表空间低限报警

1. 概述

表空间低限报警功能的设计实现是在原数据库系统基础上进行的，主要根据通过GUC参数设定表空间的预警值和表空间的最大值，表空间低限报警分为两类：普通表空间报警和审计日志报警，当在日常操作中造成的普通表空间或审计日志的实际大小超过预定的百分比就会报警。

表空间低限报警功能是KingbaseES V8项目中安全性功能需求中的一项子功能需求，此需求是安全版和JY版指定技术指标之一。

2. 参数说明

表空间低限报警涉及的参数如下：

```
threshold_alarm_tablespace_percent//普通表空间预警百分比
threshold_alarm_audit_tablespace_percent//审计表空间百分比
threshold_alarm_audit_tablespace_maxsize//审计表空间最大值(以MB为单位)
```

3. 使用说明

a. 审计表空间报警设置：
通过修改threshold_alarm_audit_tablespace_maxsize参数为审计表空间设置审计表空间的最大值(以MB为单位)

```
ALTER SYSTEM SET threshold_alarm_audit_tablespace_maxsize TO 10;
```

再修改threshold_alarm_audit_tablespace_percent参数设置当审计表空间的使用率达到一定百分比时开始报警。

```
ALTER SYSTEM SET threshold_alarm_audit_tablespace_percent TO 50;
```

b. 普通表空间报警设置：
通过命令修改表空间的大小(以MB为单位)

```
ALTER TABLESPACE TS SET MAXSIZE = 10;
```

再修改threshold_alarm_tablespace_percent参数设置当表空间的使用率达到一定百分比时开始报警。

```
ALTER SYSTEM SET threshold_alarm_tablespace_percent TO 50;
```

版权申明

联系我们

欢迎您针对此手册提出宝贵意见和建议，您的意见和建议将成为完善此手册的重要部分。

如果您发现了手册中的错误，或有更好的意见和建议，可通过以下方式发送给我们。

电子邮箱：support@kingbase.com.cn

传真：86-10-59111032

服务热线：4006011188

通信地址：北京市朝阳区容达路7号E座2层