**Vastbase G100 V2.2**

**(Build 10)**

**安全加固指南**

【版权声明】

©2007-2022 北京海量数据技术股份有限公司 版权所有

本文档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、 修改、抄袭、传播全部或部分本文档内容。

北京海量数据技术股份有限公司保留所有的权利。

【服务声明】

本文档意在向客户介绍海量数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本文档内容不做任何明示或模式的承诺或保证。

**目录**

[1. 概述 6](#_Toc28017)

[2. 安全配置规则 7](#_Toc25324)

[3. 操作系统特殊配置 8](#_Toc19701)

[3.1. 系统资源 8](#_Toc30691)

[3.1.1. 设置kernel.shmmax 8](#_Toc8873)

[3.1.2. 设置kernel.shmall 9](#_Toc11154)

[3.1.3. 配置NTP服务器 9](#_Toc4444)

[3.1.4. 配置安全SSH协议 10](#_Toc18914)

[3.1.5. 配置安全SSH服务监听IP地址 10](#_Toc24661)

[3.1.6. 配置安全SSH服务允许组 11](#_Toc9180)

[3.1.7. 配置安全SSH服务root用户允许登录方式 11](#_Toc12243)

[3.1.8. 配置安全SSH服务密码认证方式 12](#_Toc1719)

[3.1.9. 配置安全SSH服务互信IP访问限制 12](#_Toc22113)

[3.2. 服务器欺骗 13](#_Toc23639)

[3.2.1. 确保套接字文件存在 13](#_Toc21958)

[3.2.2. 确保server.key文件存在 13](#_Toc18675)

[3.2.3. 确保server.crt文件存在 14](#_Toc3243)

[4. 数据库配置 15](#_Toc30392)

[4.1. 数据库补丁与运行环境 15](#_Toc12835)

[4.1.1. 数据库拒绝不受限的连接 15](#_Toc1947)

[4.2. 数据库文件权限 16](#_Toc21176)

[4.2.1. 限制数据库的home目录权限 16](#_Toc9923)

[4.2.2. 限制${GAUSSHOME}/share目录权限 16](#_Toc689)

[4.2.3. 限制${GAUSSHOME}/bin目录权限 16](#_Toc15914)

[4.2.4. 限制data目录权限 17](#_Toc22438)

[4.2.5. 限制postgresql.conf文件权限 17](#_Toc14955)

[4.2.6. 限制pg\_hba.conf文件权限 18](#_Toc2978)

[4.3. 数据库模板 18](#_Toc21974)

[4.3.1. 没有非文档化的数据库模板存在 18](#_Toc32353)

[4.3.2. 禁止与数据库模板产生新连接 18](#_Toc27963)

[4.4. 数据库扩展 19](#_Toc11627)

[4.4.1. 没有未文档化的扩展存在 19](#_Toc13332)

[4.5. 连接设置 19](#_Toc16922)

[4.5.1. 监听地址不允许包括\*或0.0.0.0 19](#_Toc2804)

[4.5.2. 端口使用非默认端口 20](#_Toc3283)

[4.5.3. 配置数据库的最大并发连接数 20](#_Toc4702)

[4.5.4. 配置系统管理员使用的连接数 21](#_Toc16117)

[4.5.5. 配置UNIX域套接字的访问权限 22](#_Toc9592)

[4.5.6. Vastbase中数据库节点的pg\_hba.conf仅包含Vastbase内部节点连接配置 22](#_Toc330)

[4.5.7. pg\_hba.conf检查所有使用trust方法的local条目 23](#_Toc820)

[4.5.8. pg\_hba.conf无hostnossl条目 23](#_Toc13058)

[4.5.9. pg\_hba.conf没有host条目的数据库指定为all 24](#_Toc25062)

[4.5.10. pg\_hba.conf没有host条目的用户指定为all 24](#_Toc15741)

[4.5.11. pg\_hba.conf没有host条目的源地址指定为all或0.0.0.0/0 24](#_Toc18529)

[4.5.12. pg\_hba.conf除Vastbase内部节点外没有host条目使用trust方法 25](#_Toc2927)

[4.5.13. pg\_hba.conf没有host条目使用md5认证方式 25](#_Toc8455)

[4.5.14. pg\_hba.conf设置允许某时间段访问 26](#_Toc26151)

[4.6. 开发选项 26](#_Toc24191)

[4.6.1. 建议关闭系统表结构修改配置 26](#_Toc20988)

[4.7. 安全认证设置 27](#_Toc1769)

[4.7.1. 配置客户端认证超时时间 27](#_Toc2858)

[4.7.2. 配置认证加密迭代次数 27](#_Toc25021)

[4.7.3. 配置SSL协议 28](#_Toc3758)

[4.7.4. 配置SSL加密算法 28](#_Toc9673)

[4.8. 版本平台兼容性设置 29](#_Toc14046)

[4.8.1. 配置服务器拒绝带反斜杠转义的引号 29](#_Toc12256)

[4.9. 用户角色 30](#_Toc29128)

[4.9.1. 没有未知的角色存在 30](#_Toc20534)

[4.9.2. 默认用户名必须移除 30](#_Toc1961)

[4.9.3. 检查id为10的管理员角色 31](#_Toc27528)

[4.9.4. 检查CREATEROLE权限 31](#_Toc23031)

[4.9.5. 检查CREATEDB权限 32](#_Toc22132)

[4.9.6. 配置用户角色的有限期限 32](#_Toc2000)

[4.9.7. 配置密码复杂度检查 33](#_Toc17589)

[4.9.8. 配置密码存储加密方式 33](#_Toc2765)

[4.9.9. 配置密码的不可重用天数 33](#_Toc13087)

[4.9.10. 配置帐户登录失败尝试次数 34](#_Toc14220)

[4.9.11. 配置帐户锁定后自动解锁时间 34](#_Toc10889)

[4.9.12. 限制用户角色链接 35](#_Toc10096)

[4.9.13. 修改初始创建用户的密码 35](#_Toc27408)

[4.10. 权限控制 36](#_Toc32097)

[4.10.1. PUBLIC角色不允许拥有pg\_catalog.pg\_authid表的权限 36](#_Toc19024)

[4.10.2. 普通角色在PUBLIC下不允许拥有CREATE权限 36](#_Toc17812)

[4.10.3. 所有对象权限不允许授予PUBLIC 37](#_Toc9349)

[4.10.4. 控制拥有WITH GRANT OPTION权限的角色 38](#_Toc28843)

[4.10.5. 控制SECURITY DEFINER函数的执行权限 38](#_Toc32701)

[4.10.6. 控制SECURITY INVOKER函数的执行场景 39](#_Toc11430)

[4.10.7. 开启三权分立配置 39](#_Toc14715)

[4.11. 安全审计 40](#_Toc12922)

[4.11.1. 开启数据库审计 40](#_Toc6774)

[4.11.2. 开启登录注销审计 40](#_Toc5450)

[4.11.3. 开启数据库启动、停止、恢复和切换审计 41](#_Toc18783)

[4.11.4. 开启数据库锁定和解锁审计 41](#_Toc18591)

[4.11.5. 开启权限授予和回收审计 42](#_Toc7380)

[4.11.6. 对数据库对象的添加、删除、修改进行审计 42](#_Toc32008)

[4.11.7. 对数据库对象的查询进行审计 42](#_Toc3558)

[4.11.8. 配置审计优先策略 43](#_Toc12039)

[4.11.9. 配置单个审计文件的最大记录时间 43](#_Toc22499)

[4.11.10. 配置单个审计日志文件的最大容量 44](#_Toc25154)

[4.11.11. 配置所有审计日志文件占用的最大磁盘空间 44](#_Toc30522)

[4.11.12. 配置审计日志文件最小保存时间 44](#_Toc4936)

[4.11.13. 配置审计日志文件的最大数目 45](#_Toc2225)

[4.12. 统一审计 45](#_Toc31850)

[4.12.1. 开启安全审计并及记录日志 45](#_Toc21334)

[4.12.2. 对数据库对象的添加、删除、修改进行安全审计 46](#_Toc28282)

[4.12.3. 对表对象添加注释进行安全审计 47](#_Toc23237)

[4.12.4. 对表对象分析进行安全审计 47](#_Toc17522)

[4.12.5. 对权限操作进行安全审计 47](#_Toc1203)

[4.12.6. 对会话属性操作进行安全审计 48](#_Toc29000)

[4.12.7. 对登陆注销等操作进行安全审计 48](#_Toc29677)

[4.12.8. 对检查点事件进行安全审计 48](#_Toc24963)

[4.12.9. 对事务提交回滚进行安全审计 49](#_Toc3108)

[4.12.10. 对表数据插入、更新、删除、查询进行安全审计 49](#_Toc30278)

[4.12.11. 对复制表数据进行安全审计 49](#_Toc27965)

[4.12.12. 对函数执行进行安全审计 50](#_Toc19953)

[4.12.13. 对表重建索引进行安全审计 50](#_Toc27134)

[4.12.14. 对清空表数据进行安全审计 50](#_Toc28294)

[4.12.15. 对指定的对象进行安全审计 51](#_Toc23892)

[4.12.16. 对指定审计范围进行安全审计 51](#_Toc18543)

[4.12.17. 指定安全审计的响应动作 51](#_Toc6781)

[4.13. 错误报告和日志设置 52](#_Toc31846)

[4.13.1. 开启日志收集器 52](#_Toc22497)

[4.13.2. 配置日志名称 52](#_Toc25983)

[4.13.3. 配置日志文件权限 53](#_Toc12643)

[4.13.4. 禁用同名日志文件覆写 53](#_Toc12459)

[4.13.5. 配置单个日志文件最大记录时间 54](#_Toc1638)

[4.13.6. 配置单个日志文件最大容量 54](#_Toc21805)

[4.13.7. 配置客户端日志消息等级 54](#_Toc22754)

[4.13.8. 配置服务器日志等级 55](#_Toc7636)

[4.13.9. 开启用户注销时日志记录功能 56](#_Toc15932)

[4.13.10. 配置导致错误条件的SQL语句服务器日志级别 56](#_Toc12541)

[4.13.11. 开启用户登录时日志记录功能 57](#_Toc14268)

[4.13.12. 配置写入服务器的日志级别 57](#_Toc13736)

[4.13.13. 配置日志不记录主机名 57](#_Toc14296)

[4.13.14. 开启访问历史记录 58](#_Toc26146)

[4.14. 可靠性和归档设置 58](#_Toc681)

[4.14.1. 配置WAL信息记录级别 58](#_Toc9490)

[4.14.2. 开启归档模式 59](#_Toc19276)

[4.15. 数据存储加密 60](#_Toc4080)

[4.15.1. 开启数据存储加密 60](#_Toc15609)

# 概述

范围

本指南规定了运行于Linux/Unix操作系统之上的Vastbase数据库的安全配置规范。

本指南适用于公司所有应用Vastbase数据库的产品。

简介

本指南主要描述Vastbase数据库的安全配置规则，包括Vastbase数据库和数据库所运行的操作系统相关模块的安全配置规则。

本指南从数据库以及操作系统两个角度讲述加固的方法，以便给Vastbase数据库提供更加安全和稳定的运行环境。

在数据库完成安装之后开始执行安全加固。

关键字

Vastbase

# 安全配置规则

安全配置规则是指依据相关标准和公司要求，为加强Vastbase数据库安全性而制定的一系列安全配置规则。

配置规则中选项的详细说明请参见表1-1。

选项说明

| 名称 | 说明 |
| --- | --- |
| 配置说明 | 配置项的描述信息、配置的原因和依据等信息。 |
| 配置方法 | 配置的详细步骤。 |
| 检查方法 | 检查数据库或者操作系统当前配置的详细方法。 |
| 推荐值 | 满足安全要求的预期值。 |
| 风险等级 | 描述该检查项的风险影响程度，分为高，中，低三个等级。 |
| 参考 | 配置规则参考的标准，比如CVE或者CCE编号。 |

# 操作系统特殊配置

表1-2为Vastbase的变量说明，请根据实际环境进行调整。

变量参数说明

| 变量名称 | 参数说明 | 取值 |
| --- | --- | --- |
| GAUSSDATA | Vastbase的数据目录。 | 存储节点或者数据主节点的data目录。 |
| GAUSSHOME | Vastbase的安装目录。 | - |
| GAUSSUSER | 安装Vastbase指定的用户。 | - |
| GAUSSGROUP | 安装Vastbase指定用户所属的用户组。 | - |

[3.1 系统资源](#_ZH-CN_TOPIC_0245189877" \o " )

[3.2 服务器欺骗](#_ZH-CN_TOPIC_0245189887" \o " )

## 系统资源

### 设置kernel.shmmax

**配置说明：**

参数kernel.shmmax表示共享内存段的最大尺寸，单位为字节。增大参数kernel.shmmax的值可以提高Vastbase的性能。

共享内存和信号量属于系统进程间的通讯方法，设置合适的RAM以满足数据库应用程序需求。



将{SHMMAX\_VALUE}替换为本地环境中合适的值。

**配置方法：**

在/etc/sysctl.conf文件中配置参数kernel.shmmax。

**vim /etc/sysctl.conf**

**检查方法：**

**grep -P '^[^#]\*kernel.shmmax\s\*=' /etc/sysctl.conf**

**推荐值：**{SHMMAX\_VALUE}

**风险等级：**中

**参考：**

管理Kernel资源-[https://www.postgresql.org/docs/9.1/static/kernel-resources.html](https://www.postgresql.org/docs/9.1/static/kernel-resources.html" \o " )

### 设置kernel.shmall

**配置说明：**

参数kernel.shmall表示共享内存总量，单位为页。增大参数kernel.shmall的值可以提高Vastbase的性能。

共享内存和信号量属于系统进程间通讯方法，设置合适的RAM以满足数据库应用程序需求。



将{SHMALL\_VALUE}替换为本地环境中合适的值

**配置方法：**

在/etc/sysctl.conf文件中配置参数kernel.shmall。

**vim /etc/sysctl.conf**

**检查方法：**

**grep -P '^[^#]\*kernel.shmall\s\*=' /etc/sysctl.conf**

**推荐值：**{SHMALL\_VALUE}

**风险等级：中**

**参考：**

管理Kernel资源- [https://www.postgresql.org/docs/9.1/static/kernel-resources.html](https://www.postgresql.org/docs/9.1/static/kernel-resources.html" \o " )

### 配置NTP服务器

**配置说明：**

NTP（Network Time Protocol，网络时间协议）是同步网络中的各个计算机时间的一种协议。配置NTP可以使计算机的时钟同步到国际标准时间UTC（Universal Time Coordinated，世界协调时），同步多种操作系统的系统时钟。



将{NTP\_SERVER}替换为本地环境中合适的值。

**配置方法：**

在/etc/ntp.conf文件中配置合适的NTP服务器。

**vim /etc/ntp.conf**

**检查方法：**

**grep -P '^\s\*server\s+{NTP\_SERVER}.\*$' /etc/ntp.conf**

**推荐值：**{NTP\_SERVER}

**风险等级：**低

**参考：**无

### 配置安全SSH协议

**配置说明：**

SSH（Secure Shell，安全外壳协议）是建立在应用层和传输层基础上的安全协议。专为远程管理会话和其他网络服务提供安全性的协议。利用SSH协议可以有效防止远程管理过程中的信息泄露问题。



将Protocol参数赋值为本地环境中合适的值。

**配置方法：**

以root身份在/etc/ssh/sshd\_config文件中配置参数Protocol，设置之后需要重启SSH服务生效。

**su - root**   
**vim /etc/ssh/sshd\_config**   
   
**service sshd restart #重启SSH服务**

**检查方法：**

**grep -P '^[^#]\*Protocol\s\*' /etc/ssh/sshd\_config**

**推荐值：**2

**风险等级：**中

**参考：**无

### 配置安全SSH服务监听IP地址

**配置说明：**

ListenAddress用于设置sshd服务绑定的IP地址。默认情况下，会侦听所有接口的SSH要求。添加该参数限制用于防止未经授权访问系统、未经授权访问或者窃听数据、未经授权使用服务或资源、不易察觉的可行攻击。



将需要使用SSH服务的IP接口添加到参数ListenAddress中。

**配置方法：**

以root身份在/etc/ssh/sshd\_config文件中配置参数ListenAddress，设置之后需要重启SSH服务生效。

**su - root**   
**vim /etc/ssh/sshd\_config**   
   
**service sshd restart #重启SSH服务**

**检查方法：**

**grep -P '^[^#]\*ListenAddress\s\*' /etc/ssh/sshd\_config**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置安全SSH服务允许组

**配置说明：**

AllowGroups用于设置sshd服务允许登录的组，其它所有组将被拒绝。默认情况下，允许所有组登录。添加该参数限制，用于防止未经授权访问系统。



将需要使用SSH服务的组添加到参数AllowGroups中。

**配置方法：**

以root身份在/etc/ssh/sshd\_config文件中配置参数AllowGroups，设置完成之后需要重启SSH服务生效。

**su - root**   
**vim /etc/ssh/sshd\_config**   
   
**service sshd restart #重启SSH服务**

**检查方法：**

**grep -P '^[^#]\*AllowGroups\s\*' /etc/ssh/sshd\_config**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置安全SSH服务root用户允许登录方式

**配置说明：**

PermitRootLogin用于设置sshd服务是否允许root登录。默认情况下，允许root登录，建议该参数设置为no，不允许以root用户登录。添加该参数限制用于防止未经授权访问系统、未经授权访问或窃听数据、未经授权修改数据、未经授权使用服务或资源、可用性中断。



将PermitRootLogin参数设置为no。

**配置方法：**

以root身份在/etc/ssh/sshd\_config文件中配置参数PermitRootLogin，设置完成之后需要重启SSH服务。

**su - root**   
**vim /etc/ssh/sshd\_config**   
   
**service sshd restart #重启SSH服务**

**检查方法：**

**grep -P '^[^#]\*PermitRootLogin\s\*' /etc/ssh/sshd\_config**

**推荐值：**no

**风险等级：**中

**参考：**无

### 配置安全SSH服务密码认证方式

**配置说明：**

PasswordAuthentication用于设置sshd服务是否允许使用基于密码的认证。默认情况下，该值为yes，允许使用基于密码的认证，建议该参数设置为no。添加该参数限制用于防止未经授权访问或窃听数据、未经授权修改数据、可用性中断。



将PasswordAuthentication参数设置为no。

**配置方法：**

以root身份在/etc/ssh/sshd\_config文件中配置参数PasswordAuthentication，设置完成之后需要重启SSH服务。

**su - root**   
**vim /etc/ssh/sshd\_config**   
   
**service sshd restart #重启SSH服务**

**检查方法：**

**grep -P '^[^#]\*PasswordAuthentication\s\*' /etc/ssh/sshd\_config**

**推荐值：**no

**风险等级：**中

**参考：**无

### 配置安全SSH服务互信IP访问限制

**配置说明：**

安全SSH服务中的authorized\_keys文件记录用于免密码登录用户帐户的SSH密钥。该文件中可以指定哪些主机可以免密码登录本机的SSH服务。建议只允许使用互信服务的主机登录。添加该参数限制用于防止未经授权访问系统。



在使用互信的用户帐户家目录.ssh目录下的authorized\_keys文件中添加from字段，限制访问主机。

**配置方法：**

以用户帐户身份在~/.ssh/authorized\_keys文件中配置from字段。

**vim ~/.ssh/authorized\_keys**

配置内容见如下示例（在建立互信后检查该文件中，是否配置有from字段，如没有，请添加需要使用互信的主机IP和主机名称）：

from="10.180.40.100,10.180.40.101,10.180.40.102,plat1,plat2,plat3" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArMU/cd5QDIkA/trmLpfj1W1zBk2pW8zyDjMEvc7JiAdtOlsOIFyRpdmdMv39WuQfZ6G/+zbAc9JJwYLP8/3aYgkNSJ10+c4VcyQJ8jBsLJyKLW/Pt2t3IlY+zG35UUyhyDo9P8Bt/uZ0d1XbgRrWuVrO8idWIFncallJYrmaTvd5b7CULKMSrFI+CMys9FLYOk5lGHVdQSQUE0uNor/k63plsyyL7+fVqG2ZDRbB77nVC+ah5PMfogBym24g1ouiuxQPIihOTt2dDPeqnnWpC0/WzhB/PjIoeHCK+HbGFg92eWdi7CJexb7H6OiG/Ylvej2ED4yc8WqEnCSZmWQDow== xijie\_trunk@plat1

**检查方法：**

**cat ~/.ssh/authorized\_keys**

**推荐值：**无

**风险等级：**中

**参考：**无

## 服务器欺骗

### 确保套接字文件存在

**配置说明：**

某些应用程序可能在/tmp目录下查找socket文件，欺骗服务器。系统启动期间默认创建一个socket文件/tmp/${USER}\_vastbase/.s.Pvsql.${PGPORT}，用以防止某些应用程序在/tmp目录下去自行创建或覆盖socket文件，其中${USER}为安装用户名，${PGPORT}为节点端口号。socket文件所在路径与实际安装时的路径配置有关，可以参考${GAUSSDATA}/postgresql.conf配置文件中unix\_socket\_directory参数配置。

**检查方法：**

**find /tmp/${USER}\_vastbase/.s.Pvsql.${PGPORT}**

**推荐值：**无

**风险等级：**中

**参考：**

Preventing Server Spoofing - [https://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html](https://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html" \o " )

### 确保server.key文件存在

**配置说明：**

阻止TCP连接服务器欺骗的解决方案是使用SSL证书，并确保客户端检验服务器的证书。因此，服务器必须仅配置hostssl连接，并存在SSL的server.key（密钥）和server.crt（证书）。

**配置方法：**

确定文件${GAUSSDATA}/server.key存在，其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**find ${GAUSSDATA}/server.key**

**推荐值：**无

**风险等级：**中

**参考：**

Preventing Server Spoofing - [https://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html](https://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html" \o " )

### 确保server.crt文件存在

**配置说明：**

阻止TCP连接服务器欺骗的最佳解决方案是使用SSL证书，并确保客户端会检验服务器的证书。因此，服务器必须仅配置hostssl连接，并存在SSL的server.key（密钥）和server.crt（证书）。

**配置方法：**

确定文件${GAUSSDATA}/server.crt存在，其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**find ${GAUSSDATA}/server.crt**

**推荐值：**无

**风险等级：**中

**参考：**

Preventing Server Spoofing - [http://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html](http://www.postgresql.org/docs/9.1/static/preventing-server-spoofing.html" \o " )

# 数据库配置

[4.1 数据库补丁与运行环境](#_ZH-CN_TOPIC_0245189892" \o " )

[4.2 数据库文件权限](#_ZH-CN_TOPIC_0245189894" \o " )

[4.3 数据库模板](#_ZH-CN_TOPIC_0245189901" \o " )

[4.4 数据库扩展](#_ZH-CN_TOPIC_0245189904" \o " )

[4.5 连接设置](#_ZH-CN_TOPIC_0245189906" \o " )

[4.6 开发选项](#_ZH-CN_TOPIC_0245189921" \o " )

[4.7 安全认证设置](#_ZH-CN_TOPIC_0245189923" \o " )

[4.8 版本平台兼容性设置](#_ZH-CN_TOPIC_0245189929" \o " )

[4.9 用户角色](#_ZH-CN_TOPIC_0245189931" \o " )

[4.10 权限控制](#_ZH-CN_TOPIC_0245189945" \o " )

[4.11 数据库审计](#_ZH-CN_TOPIC_0245189953" \o " )

[4.12 错误报告和日志设置](#_ZH-CN_TOPIC_0245189967" \o " )

[4.13 可靠性和归档设置](#_ZH-CN_TOPIC_0245189981" \o " )

## 数据库补丁与运行环境

### 数据库拒绝不受限的连接

**配置说明：**

为了控制访问数据库的会话数量，所以应该限制连接数据库的会话数量。推荐配置会话数量限制在1024个以内。



若设置参数datconnlimit为-1，表示不限制连接数据库的会话数量。

**配置方法：**

**vsql -d postgres -p** 8000

**UPDATE pg\_database SET datconnlimit=***<CONN\_LIMIT\_VALUE>* **WHERE datname=***<DATABASE\_NAME>***;**

**检查方法：**

**SELECT datname FROM pg\_database WHERE datistemplate = false AND datconnlimit = -1;**

**推荐值：**1000

**风险等级：**低

**参考：**无

## 数据库文件权限

### 限制数据库的home目录权限

**配置说明：**

${GAUSSHOME}是Vastbase的安装目录，为了防止安装包被恶意篡改或破坏给客户网络安全造成威胁。所以此目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod 0700 ${GAUSSHOME}**

**检查方法：**

**find ${GAUSSHOME} -prune \( ! -user ${GAUSSUSER} -o ! -group ${GAUSSGROUP} -o -perm /g=rwx,o=rwx \)**

环境变量${GAUSSUSER}和${GAUSSGROUP}需要配置为Vastbase的安装用户和用户组。

**推荐值：**无

**风险等级：**中

**参考：**无

### 限制${GAUSSHOME}/share目录权限

**配置说明：**

${GAUSSHOME}/share目录包含了Vastbase的共享组件，为了防止共享组件被恶意篡改或破坏，所以此目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod og-w ${GAUSSHOME}/share**

**检查方法：**

**find ${GAUSSHOME}/share -prune -perm /g=w,o=w**

**推荐值：**无

**风险等级：**中

**参考：**无

### 限制${GAUSSHOME}/bin目录权限

**配置说明：**

${GAUSSHOME}/bin目录包含了数据库的二进制文件，为了防止二进制文件被恶意篡改或破坏给客户信息安全造成威胁。所以此目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod og-w ${GAUSSHOME}/bin**

**检查方法：**

**find ${GAUSSHOME}/bin -prune -perm /g=w,o=w**

**推荐值：**无

**风险等级：**中

**参考：**无

### 限制data目录权限

**配置说明：**

data目录包含了用户数据文件，为了防止数据文件被恶意篡改或破坏给客户的数据信息安全造成威胁。所以此目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod 0700 ${GAUSSDATA}**

其中GAUSSDATA为数据库节点的data目录。

**检查方法：**

**find ${GAUSSDATA} -prune \( ! -user ${GAUSSUSER} -o ! -group ${GAUSSGROUP} -o -perm /g=rwx,o=rwx \)**

环境变量${GAUSSUSER}和${GAUSSGROUP}需要配置为Vastbase的安装用户和用户组。

**推荐值：**无

**风险等级：**中

**参考：**无

### 限制postgresql.conf文件权限

**配置说明：**

配置文件postgresql.conf包含数据库的默认配置，为了防止配置文件的参数被恶意篡改给客户的信息安全造成威胁。所以此文件目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod 0600 ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库节点的data目录。

**检查方法：**

**find ${GAUSSDATA}/postgresql.conf \( ! -user ${GAUSSUSER} -o ! -group ${GAUSSGROUP} -o -perm /u=x,g=rwx,o=rwx \)**

环境变量${GAUSSUSER}和${GAUSSGROUP}需要配置为Vastbase的安装用户和用户组。

**推荐值：**无

**风险等级：**中

**参考：**无

### 限制pg\_hba.conf文件权限

**配置说明：**

配置文件pg\_hba.conf包含了连接数据库的配置信息，为了防止配置文件的参数被恶意篡改给客户的信息安全造成威胁。所以此文件目录应该受到保护，不允许未认证的用户访问。

**配置方法：**

**chmod 0600 ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库节点的data目录。

**检查方法：**

**find ${GAUSSDATA}/pg\_hba.conf \( ! -user ${GAUSSUSER} -o ! -group ${GAUSSGROUP} -o -perm /u=x,g=rwx,o=rwx \)**

环境变量${GAUSSUSER}和${GAUSSGROUP}需要配置为Vastbase的安装用户和用户组。

**推荐值：**无

**风险等级：**中

**参考：**无

## 数据库模板

### 没有非文档化的数据库模板存在

**配置说明：**

必须将所有数据库模板文档化。应该仔细检查任何具有未知用途的数据库模板。

**配置方法：**

删除不相关的数据库模板。

**检查方法：**

**SELECT datname FROM pg\_database WHERE datistemplate = true AND datname<>'template0' AND datname<>'template1';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 禁止与数据库模板产生新连接

**配置说明：**

为了防止对数据库模板的不当修改，所以在系统表pg\_database中设置字段datallowconn为false，表示不允许用户连接到此数据库。将数据库模板template1设置为不允许用户连接会影响数据库升级，需要在升级前检查此配置。如果升级前在系统表pg\_database中template1对应的字段datallowconn为false，需要先设置为true，等升级完成后再将配置改为false。

**配置方法：**

**vsql -d postgres -p** 8000

**UPDATE pg\_database SET datallowconn =false WHERE datname=***<DATABASE\_NAME>***;**

**检查方法：**

**SELECT datname FROM pg\_database WHERE datistemplate = true AND datallowconn = true;**

**推荐值：**无

**风险等级：**低

**参考：**无

## 数据库扩展

### 没有未文档化的扩展存在

**配置说明：**

应该将所有安装的扩展文档化。应该仔细检查任何未识别的扩展。

**配置方法：**

**vsql -d postgres -p** 8000

**DELETE FROM pg\_extension where EXTNAME not in ('plpvsql', 'dist\_fdw', 'file\_fdw', 'hdfs\_fdw', 'gc\_fdw', 'log\_fdw', 'hstore', 'packages', 'dimsearch', 'uuid-ossp', 'tsdb');**

**检查方法：**

**SELECT extname, extversion FROM pg\_extension where EXTNAME not in ('plpvsql', 'dist\_fdw', 'file\_fdw', 'hdfs\_fdw', 'gc\_fdw', 'log\_fdw', 'hstore', 'packages', 'dimsearch', 'uuid-ossp', 'tsdb');**

**推荐值：**空

**风险等级：**中

**参考：**无

## 连接设置

### 监听地址不允许包括\*或0.0.0.0

**配置说明：**

监听地址不允许包含特殊字符“\*”或“0.0.0.0”，因为“\*”或“0.0.0.0”表示监听所有可用的IP地址。

**配置方法：**

在postgresql.conf配置文件中修改参数listen\_addresses为“localhost”，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**gs\_om -t stop**   
**gs\_om -t start**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'listen\_addresses' AND (setting = '\*' OR setting = '0.0.0.0');**

**推荐值：**无

**风险等级：**中

**参考：**无

### 端口使用非默认端口

**配置说明：**

为了防止非法连接数据库，对数据进行非法操作，请及时修改默认端口号。



默认端口为8000。

**配置方法：**

在postgresql.conf配置文件中修改服务器监听端口参数port为非8000的值，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**gs\_om -t stop**   
**gs\_om -t start**

**检查方法：**

SELECT name,setting FROM pg\_settings WHERE name = 'port' AND setting = '8000';

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置数据库的最大并发连接数

**配置说明：**

参数max\_connections控制数据库的最大并发连接数。增加这个参数值可能引起Vastbase请求更多的System V共享内存或者信号量，会导致超出操作系统默认配置允许的值，并且增加max\_connections时要同步调整参数max\_pool\_size，保证参数max\_pool\_size的值大于参数max\_connections的值。



修改{MAX\_CONNECTIONS}为适合本地环境的值。

参数max\_pool\_size控制连接池的最大连接数。

**配置方法：**

在postgresql.conf配置文件中修改参数max\_connections为800，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D {GA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'max\_connections';**

**推荐值：**[1..800]

**风险等级：**中

**参考：**无



Vastbase数据库属于底层基础软件，其访问端口完全对外开放。攻击者可利用伪造的IP地址构造泛洪攻击，即向被攻击端发送大量的连接请求，导致被攻击端在等待关闭该连接的过程中消耗资源，影响正常业务。因此建议解决方案在部署Vastbase数据库时，在路由器层配置TCP拦截或采用Anti-DDos设备、防火墙等方案。具体可由上层业务部署或用户依据自身安全网络组网进行统一规划。

### 配置系统管理员使用的连接数

**配置说明：**

参数sysadmin\_reserved\_connections表示预留给Vastbase系统管理员的最少连接数。此参数值必须小于max\_connections的值。

**配置方法：**

在postgresql.conf配置文件中修改参数sysadmin\_reserved\_connections为3，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='sysadmin\_reserved\_connections';**

**推荐值：**[1..3]

**风险等级：**中

**参考：**无

### 配置UNIX域套接字的访问权限

**配置说明：**

参数unix\_socket\_permissions设置了UNIX域套接字的访问权限。默认权限为0777表示所有人都可以连接。推荐配置应该是0770（只有用户和组允许）或者0700（只有用户允许）。

**配置方法：**

在postgresql.conf配置文件中修改参数unix\_socket\_permissions为0770，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'unix\_socket\_permissions';**

**推荐值：**0770、0700

**风险等级：**中

**参考：**无

### Vastbase中数据库节点的pg\_hba.conf仅包含Vastbase内部节点连接配置

**配置说明：**

Vastbase节点部署于安全内网环境中，数据库节点不提供对外连接访问，仅允许内部Vastbase节点通讯连接。数据库节点的pg\_hba.conf文件中仅包含Vastbase内部节点的连接配置以及可信赖的内部安全连接。

**配置方法：**

在VastbaseDN上的pg\_hba.conf文件中去掉除Vastbase内部节点以及可信赖内部安全连接之外的host配置方式。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+.+[Tt][Rr][Uu][Ss][Tt]\s\*$' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**只有内部节点

**风险等级：**中

**参考：**无

### pg\_hba.conf检查所有使用trust方法的local条目

**配置说明：**

Trust认证方法假定所有可以连接到服务器的用户都可以访问数据库。在单用户的工作站上此方法非常方便，但不适用于多用户机器。所以应检查所有使用trust认证方法的local条目。

**配置方法：**

在配置文件pg\_hba.conf中配置所有非内部通信项为非trust认证方法。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\* \s+[Tt][Rr][Uu][Ss][Tt]\s\*$' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening

### pg\_hba.conf无hostnossl条目

**配置说明：**

hostnossl条目指定连接不使用SSL加密。从安全角度考虑，尽可能使用SSL安全连接。

目前SSL连接只支持CN与对外之间的连接，不支持内部节点之间的连接。

**配置方法：**

在配置文件pg\_hba.conf中删除所有hostnossl条目。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*hostnossl' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

### pg\_hba.conf没有host条目的数据库指定为all

**配置说明：**

host连接条目若指定数据库为all，则允许连接用户连接到任何数据库。推荐host条目的取值仅为需要连接的数据库。

**配置方法：**

在配置文件pg\_hba.conf中删除数据库所有非内部连接项中host、hostssl、hostnossl为all的条目。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+[Aa][Ll][Ll]' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**低

**参考：**无

### pg\_hba.conf没有host条目的用户指定为all

**配置说明：**

host条目配置user为all允许任意用户连接到数据库。推荐host条目的取值仅为需要连接数据库的用户。

**配置方法：**

在配置文件pg\_hba.conf中删除所有非内部连接项中host、hostssl、hostnossl用户为all的条目。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+\S+\s+[Aa][Ll][Ll]' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**低

**参考：**无

### pg\_hba.conf没有host条目的源地址指定为all或0.0.0.0/0

**配置说明：**

host条目配置源地址为all或0.0.0.0/0，表示允许任意IP连接到数据库。推荐host条目配置的源地址仅为需要连接数据库的IP。

**配置方法：**

在配置文件pg\_hba.conf中删除所有host、hostssl、hostnossl地址为all或0.0.0.0/0的条目。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+\S+\s+\S+\s+[Aa][Ll][Ll]' ${GAUSSDATA}/pg\_hba.conf**   
**grep '0.0.0.0/0' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**低

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/owasp_backend_security_project_postgresql_hardening" \o " )

### pg\_hba.conf除Vastbase内部节点外没有host条目使用trust方法

**配置说明：**

Vastbase节点部署在安全内网环境中，仅允许内部Vastbase节点间使用trust认证方式通讯。trust认证假定所有可以连接到Vastbase服务器节点的用户都可以访问数据库。此配置仅适用于机器上的每个用户都允许访问数据库的TCP/IP连接。

**配置方法：**

在配置文件pg\_hba.conf中配置host项为非trust方法（Vastbase内部节点除外）。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+.+[Tt][Rr][Uu][Ss][Tt]\s\*$' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**只有内部节点

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

### pg\_hba.conf没有host条目使用md5认证方式

**配置说明：**

md5认证方法为不安全的认证方法，推荐使用SHA256认证方式。



Vastbase保留md5认证和密码存储，是为了便于第三方工具的使用（比如TPCC评测工具）。如果用户需要使用第三方工具还要重新配置pg\_hba.conf文件中的内容。

**配置方法：**

在配置文件pg\_hba.conf中配置host项为SHA256方法。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+.+[Mm][Dd][5]\s\*$' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**中

**参考：**无

### pg\_hba.conf设置允许某时间段访问

**配置说明：**

当期望建立一个数据库会话时，Vastbase会根据访问配置文件pg\_hba.conf中配置的访问时间段判断是否允许建立数据库会话。

**配置方法：**

在配置文件pg\_hba.conf中配置所需限制项的时间段，如配置host项的时间段为”[2020-10-01,2020-10-08]”。

**vim ${GAUSSDATA}/pg\_hba.conf**

其中GAUSSDATA为数据库主节点的data目录。

**检查方法：**

**grep -P '^[^#]\*host(ssl|nossl)?\s+.+\d{4}-\d{2}-\d{2}' ${GAUSSDATA}/pg\_hba.conf**

**推荐值：**无

**风险等级：**中

**参考：**无

## 开发选项

### 建议关闭系统表结构修改配置

**配置说明：**

参数allow\_system\_table\_mods允许修改系统表的结构。开发选项参数和Vastbase内部协同工作，在某些情况下可以协助恢复遭到严重破坏的数据库。在生产条件下的数据库中，不建议使用该配置。

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'allow\_system\_table\_mods';**

**推荐值：**off

**风险等级：**中

**参考：**无

## 安全认证设置

### 配置客户端认证超时时间

**配置说明：**

参数authentication\_timeout控制完成客户端认证的时间上限，默认为一分钟。此参数可以避免出现问题的客户端长时间占用连接通道。

**配置方法：**

在postgresql.conf配置文件中修改参数authentication\_timeout为1min，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'authentication\_timeout';**

**推荐值：**[1..60]

**风险等级：**中

**参考：**无

### 配置认证加密迭代次数

**配置说明：**

认证加密信息生成过程中使用的迭代次数。

**配置方法：**

在postgresql.conf配置文件中修改参数auth\_iteration\_count为10000，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'auth\_iteration\_count';**

**推荐值：**10000

**风险等级：**中

**参考：**无

### 配置SSL协议

**配置说明：**

SSL协议是为网络通信提供安全及数据完整性的一种安全协议。在使用数据库过程中尽可能使用SSL进行TCP/IP连接。

**配置方法：**

在postgresql.conf配置文件中修改参数ssl为on，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

有关创建服务器私钥和证书的细节信息，请参考《管理员指南》。

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='ssl';**

**推荐值：**on

**风险等级：**中

**参考：**无

### 配置SSL加密算法

**配置说明：**

参数ssl\_ciphers定义了一系列安全连接需要的SSL加密算法。Vastbase支持如下算法：

* DHE-RSA-AES256-GCM-SHA384
* DHE-RSA-AES128-GCM-SHA256
* DHE-DSS-AES256-GCM-SHA384
* DHE-DSS-AES128-GCM-SHA256
* DHE-RSA-AES256-SHA256
* DHE-RSA-AES128-SHA256
* DHE-DSS-AES256-SHA256
* DHE-DSS-AES128-SHA256
* DHE-RSA-AES256-CCM
* DHE-RSA-AES128-CCM
* DHE-RSA-AES256-SHA
* DHE-RSA-AES128-SHA
* DHE-DSS-AES256-SHA
* DHE-DSS-AES128-SHA

配置参数ssl\_ciphers为ALL，Vastbase默认会选择DHE-RSA-AES256-GCM-SHA384。

**配置方法：**

在postgresql.conf配置文件中修改参数ssl\_ciphers为ALL，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='ssl\_ciphers';**

**推荐值：**"ALL"

**风险等级：**中

**参考：**无

## 版本平台兼容性设置

### 配置服务器拒绝带反斜杠转义的引号

**配置说明：**

参数backslash\_quote控制字符串里面\'是否可以代替引号。由于历史原因PostgreSQL接受“\'”，然而使用“\'”会引起安全风险，有可能发生SQL注入攻击。配置服务器为了避免这个风险，拒绝带反斜杠转义的引号的查询。推荐使用SQL标准方法，使用一个引号写两遍（''）的方法。

**配置方法：**

在postgresql.conf配置文件中修改参数backslash\_quote为safe\_encoding，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='backslash\_quote';**

**推荐值：**"safe\_encoding" || "off"

**风险等级：**中

**参考：**无

## 用户角色

### 没有未知的角色存在

**配置说明：**

检查任何未知的角色。对于每个查询出来的角色，检查此角色的用途，如果不是必须存在，则删除这些角色。

**检查方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 检查角色。

**SELECT rolname FROM pg\_roles;**

----结束

**应对策略：**

**DROP ROLE** *<ROLE\_NAME>***;**

**推荐值：**无

**风险等级：**中

**参考：**无

### 默认用户名必须移除

**配置说明：**

不要使用众所周知的用户名，比如：postgres。

**检查方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 检查默认用户名。

**SELECT rolname FROM pg\_roles WHERE rolname='postgres';**

----结束

**应对策略：**

删除用户名“postgres”或者使用其他用户名替代“postgres”。

**推荐值：**无

**风险等级：**中

**参考：**无

### 检查id为10的管理员角色

**检查说明：**

数据库id为10的系统管理员具有数据库的最高权限，并且具有所有的系统权限和对象权限。建议仅将此用户作为DBA管理用途，而非业务应用。应该仔细检查此系统管理员的操作记录。

**检查方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 检查id为10的系统管理员。

**SELECT rolname FROM pg\_roles WHERE rolsuper=true;**

----结束

**推荐值：**系统安装时所指定的用户名

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

### 检查CREATEROLE权限

**检查说明：**

只有拥有CREATEROLE权限的角色才能创建、修改或者删除其他角色。拥有此权限也可以授予或者取消其他角色的权限。为了避免随意为其他角色授予权限，请删除不需要拥有CREATEROLE权限的角色。

**检查方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 检查拥有CREATEROLE权限的角色。

**SELECT rolname FROM pg\_roles WHERE rolcreaterole=true AND rolsuper=false;**

----结束

**应对策略：**

**DROP ROLE** *<ROLE\_NAME>***;**

**推荐值：**无

**风险等级：**中

**参考：**无

### 检查CREATEDB权限

**检查说明：**

拥有CREATEDB权限的角色才能创建数据库，也可以创建和定义数据库的所有用户。为了避免随意创建数据库，请删除不需要拥有CREATEDB权限的角色。

**检查方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 检查拥有CREATEDB权限的角色。

**SELECT rolname FROM pg\_roles WHERE rolcreatedb=true AND rolsuper=false;**

----结束

**应对策略：**

**DROP ROLE** *<ROLE\_NAME>***;**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置用户角色的有限期限

**配置说明：**

创建角色时，使用VALID BEGIN关键字设置角色的有效开始时间，VALID UNTIL关键字设置角色的有效结束时间。如果忽略这两个关键字，角色会长期有效。Vastbase各节点的角色过期时间以各节点操作系统的时钟作为依据，用户部署Vastbase时可以使用NTP确保各集群节点的时间保持一致，否则会存在各节点登录帐户过期时间不一致风险。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 修改角色有效期。

**alter role** *<rolename>* **valid begin** *'2012-1-1 0:0:0'* **valid until** *'2014-1-1 0:0:0'***;**

----结束

**检查方法：**

**SELECT rolname,rolvalidbegin,rolvaliduntil FROM pg\_roles WHERE rolsuper=false AND (rolvalidbegin IS NULL OR rolvaliduntil IS NULL);**

**推荐值：**无

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening -[https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

### 配置密码复杂度检查

**配置说明：**

为了保证数据库的使用安全，创建用户或者修改密码时都会进行检查密码复杂度校验，Vastbase严禁使用不满足密码复杂度要求的密码。

**配置方法：**

在postgresql.conf配置文件中修改参数**password\_policy**为1，然后重载数据库。

**Vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='password\_policy' and setting !=1;**

**推荐值：**1

**风险等级：**中

**参考：**无

### 配置密码存储加密方式

**配置说明：**

Vastbase支持MD5、SHA256以及SM3共三种密码储存加密方式。MD5方式为不安全的加密方式，不应该配置，保留此方式仅为兼容开源第三方工具，应该配置为SHA256方式（默认配置）。

**配置方法：**

在postgresql.conf配置文件中修改参数**password\_encryption\_type**为2，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='password\_encryption\_type' and setting !=2;**

**推荐值：**2

**风险等级：**中

**参考：**无

### 配置密码的不可重用天数

**配置说明：**

在数据库使用过程中，推荐配置用户密码不可重用天数，以避免用户反复使用相同的密码，导致密码被破解。配置参数password\_reuse\_time后，密码只有超过此参数值后才允许被重用。

**配置方法：**

在postgresql.conf配置文件中修改参数**password\_reuse\_time**为60，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='password\_reuse\_time';**

**推荐值：**60

**风险等级：**中

**参考：**无

### 配置帐户登录失败尝试次数

**配置说明：**

在数据库使用过程中，推荐配置帐户登录失败尝试次数，防止密码被暴力破解。配置参数failed\_login\_attempts后，当失败次数超过此参数值后，帐户将被自动锁定。

**配置方法：**

在postgresql.conf配置文件中修改参数**failed\_login\_attempts**为10，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='failed\_login\_attempts';**

**推荐值：**10

**风险等级：**中

**参考：**无

### 配置帐户锁定后自动解锁时间

**配置说明：**

用户需要配置帐户发生异常被锁定后，自动解锁的时间。

**配置方法：**

在postgresql.conf配置文件中修改参数**password\_lock\_time**为1，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='password\_lock\_time';**

**推荐值：**1

**风险等级：**中

**参考：**无

### 限制用户角色链接

**配置说明：**

不应该允许用户角色无限制数量的连接。如果参数rolconnlimit设置为-1，表示允许无限制数量的并发连接。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 修改用户角色连接数。

**alter role** *<ROLENAME>* **connection limit** *<connection\_num>***;**

----结束

**检查方法：**

**SELECT rolname,rolconnlimit FROM pg\_roles WHERE rolconnlimit=-1;**

**推荐值：**无

**风险等级：**低

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

### 修改初始创建用户的密码

**配置说明：**

初始创建用户是Vastbase中权限最高、id为10的系统管理员，该用户不能被修改和锁定，因此，数据库安装完成后就应该修改其密码。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 使用以下命令修改密码。

**ALTER ROLE** *<ROLENAME>* **IDENTIFIED BY** *<newpassword>* **REPLACE** *<oldpassword>***;**

----结束

使用新密码重新登录判断能否修改成功。

如果成功，表示初始密码已修改成功。

否则，请重新设置密码。

**推荐值：**无

**风险等级：**高

**参考：**无

## 权限控制

### PUBLIC角色不允许拥有pg\_catalog.pg\_authid表的权限

**配置说明：**

pg\_catalog.pg\_authid表中包含了所有的系统角色，为了防止更改表中系统角色，PUBLIC角色不允许拥有这个表的任何权限。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 回收PUBLIC角色拥有系统表pg\_authid的权限。

**REVOKE ALL ON pg\_authid FROM PUBLIC;**

----结束

**检查方法：**

**SELECT relname,relacl FROM pg\_class WHERE relname = 'pg\_authid' AND CAST(relacl AS TEXT) LIKE '%,=%}';**

**推荐值：**无

**风险等级：**高

**参考：**无

### 普通角色在PUBLIC下不允许拥有CREATE权限

**配置说明：**

普通用户可以在PUBLIC下创建与系统函数的同名恶意函数，使其他用户不小心误调用到恶意函数，以破坏数据库安全。

另外如果普通角色拥有PUBLIC的CREATE权限，则任何用户都可以在PUBLIC表空间下创建表或者其他数据库对象。其他用户也可以查看和修改这些表和数据库对象，因此不建议普通角色拥有PUBLIC的CREATE权限。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 回收PUBLIC角色拥有的CREATE权限。

**REVOKE CREATE ON SCHEMA PUBLIC FROM public;**

----结束

**检查方法：**

**SELECT CAST(has\_schema\_privilege('public','public','CREATE') AS TEXT);**

**推荐值：**"false"

**风险等级：**中

**参考：**

OWASP Backend Security Project PostgreSQL Hardening - [https://www.owasp.org/index.php/OWASP\_Backend\_Security\_Project\_PostgreSQL\_Hardening](https://www.owasp.org/index.php/OWASP_Backend_Security_Project_PostgreSQL_Hardening" \o " )

A Guide to CVE-2018-1058: Protect Your Search Path - [https://wiki.postgresql.org/wiki/A\_Guide\_to\_CVE-2018-1058%3A\_Protect\_Your\_Search\_Path](https://wiki.postgresql.org/wiki/A_Guide_to_CVE-2018-1058%3A_Protect_Your_Search_Path" \o " )

### 所有对象权限不允许授予PUBLIC

**配置说明：**

PUBLIC角色属于任何用户，为了保障数据库数据的安全，此角色应该拥有尽可能少的权限，所以应该控制PUBLIC角色的权限。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 回收PUBLIC角色拥有的对象权限。

**REVOKE ALL ON** *<OBJECT\_NAME>* **FROM public;**

----结束

**检查方法：**

**SELECT relname,relacl FROM pg\_class WHERE CAST(relacl AS TEXT) LIKE '%,=%}' OR CAST(relacl AS TEXT) LIKE '{=%}';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 控制拥有WITH GRANT OPTION权限的角色

**配置说明：**

拥有WITH GRANT OPTION权限的用户可以把对象的权限授权给其他人，为了保障数据库数据的安全，所以要控制拥有此权限的用户。

WITH GRANT OPTION与WITH ADMIN OPTION的差别在于前者用于把对象的权限赋给用户，后者用于把用户的权限赋给用户，需要区别使用。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 回收PUBLIC角色拥有的WITH GRANT OPTION权限。

**REVOKE GRANT OPTION FOR ALL ON** *<OBJECT\_NAME>* **FROM** *<ROLE\_NAME>***;**

----结束

**检查方法：**

**SELECT relname,relacl FROM pg\_class WHERE CAST(relacl AS TEXT) LIKE '%=%\*%/%';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 控制SECURITY DEFINER函数的执行权限

**配置说明：**

SECURITY DEFINER函数是以创建它的用户的权限执行的，所以一定确保这样的函数不被滥用。为了安全，应该设置search\_path排除可以被任何不信任用户更改的模式。这能避免恶意用户用生成的对象来替代函数中使用的对象的攻击方法。



SECURITY DEFINER和AUTHID DEFINER的功能相同。

**配置方法：**

1. 连接数据库。

**vsql -d postgres -p** 8000

1. 设置search\_path。

**SET search\_path = admin, pg\_temp;**

1. 回收PUBLIC角色的SECURITY DEFINER函数。

**REVOKE EXECUTE ON FUNCTION** *<FUNCTION\_NAME>* **FROM PUBLIC;**

----结束

**检查方法：**

**show search\_path;**   
**SELECT CAST(has\_function\_privilege('public', '***function\_name()***', 'EXECUTE') AS TEXT);**

**推荐值：**admin, pg\_temp

**风险等级：**中

**参考：**

[http://postgres-xc.sourceforge.net/docs/1\_1/sql-createfunction.html](http://postgres-xc.sourceforge.net/docs/1_1/sql-createfunction.html" \o " )

### 控制SECURITY INVOKER函数的执行场景

**配置说明：**

SECURITY INVOKER函数是以调用它的用户的权限来执行，所以在调用非自身创建的这类函数时，一定要先检查函数执行内容，避免造成函数创建者借助执行者的权限执行了越权的操作。



重点需要检查的场景：

* 普通用户或管理员执行非自身创建的SECURITY INVOKER函数。
* 普通用户或管理员对非自身创建的含有单机触发器的表执行INSERT/UPDATE/DELETE/TRUNCATE操作。

**配置方法：**

无

**检查方法：**

1、普通用户或管理员执行非自身创建的SECURITY INVOKER函数。

**SELECT prosrc FROM pg\_proc a WHERE proname = '*function\_name*'and (upper(a.prosrc) like '%ALTER USER%' or upper(a.prosrc) like '%ALTER ROLE%' or upper(a.prosrc) like '%GRANT%');**

2、普通用户或管理员对非自身创建的含有单机触发器的表执行INSERT/UPDATE/DELETE/TRUNCATE操作。

**SELECT a.prosrc FROM PG\_PROC a INNER JOIN PG\_TRIGGER c ON a.oid = c.tgfoid INNER JOIN PG\_CLASS b ON b.oid = c.tgrelid AND b.relname = '*table\_name*' and (upper(a.prosrc) like '%ALTER USER%' or upper(a.prosrc) like '%ALTER ROLE%' or upper(a.prosrc) like '%GRANT%');**

**推荐值：**空

**风险等级：**中

**参考**：无

### 开启三权分立配置

**配置说明：**

如果设置参数enable\_separation\_of\_duty为on，则会开启三权分立配置，限制系统管理员的权限。不允许系统管理员创建和更改用户配置，且不能访问私有用户schema下的数据，这样可以有效控制系统管理员访问用户私有数据。建议用户关闭GUC参数enable\_copy\_server\_files来控制系统管理员的copy权限，防止系统管理员通过copy命令读取或者修改用户配置文件。

三权分立特性的详细说明请参见《管理员指南》。

**配置方法：**

修改参数enable\_separation\_of\_duty为on，然后重启数据库。

**Vi ${GAUSSDATA}/postgresql.conf**  
**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'enable\_separation\_of\_duty'** **and setting != 'on';**

**推荐值：**on

**风险等级：**中

**参考：**无

## 安全审计

### 开启数据库审计

**配置说明：**

审计日志记录是安全事件中事后追溯、定位问题原因及划分事故责任的重要手段。推荐开启审计日志选项audit\_enabled。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_enabled**为on，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload** -D ${GAUSSDATA}

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_enabled' and setting != 'on';**

**推荐值：**on

**风险等级：**中

**参考：**无

### 开启登录注销审计

**配置说明：**

参数audit\_login\_logout决定是否审计用户登录和注销，如果开启此选项可以追溯有哪些用户曾经登录数据库，何时注销登录。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_login\_logout**为7，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_login\_logout';**

**推荐值：**7

**风险等级：**中

**参考：**无

### 开启数据库启动、停止、恢复和切换审计

**配置说明：**

参数audit\_database\_process决定是否对数据库的启动、停止、切换和恢复进行审计，如果开启此选项可以追溯数据库的运行状态变化。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_database\_process**为1，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_database\_process';**

**推荐值：**1

**风险等级：**中

**参考：**无

### 开启数据库锁定和解锁审计

**配置说明：**

参数audit\_user\_locked决定是否对审计数据库用户的锁定和解锁，如果开启此选项可以追溯数据库的帐号锁定变化。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_user\_locked**为1，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_user\_locked';**

**推荐值：**1

**风险等级：**中

**参考：**无

### 开启权限授予和回收审计

**配置说明：**

参数audit\_grant\_revoke决定是否对审计数据库用户权限授予和回收的操作，如果开启此选项可以追溯数据库帐户的权限变化。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_grant\_revoke**为1，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_grant\_revoke';**

**推荐值：**1

**风险等级：**低

**参考：**无

### 对数据库对象的添加、删除、修改进行审计

**配置说明：**

参数audit\_system\_object决定是否对数据库对象的CREATE、DROP、ALTER操作进行审计，数据库对象包括DATABASE、USER、SCHEMA、TABLE等，如果开启此选项可以追溯这些数据库对象信息的变化。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_system\_object**为12295，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_system\_object';**

**推荐值：**12295

**风险等级：**低

**参考：**无

### 对数据库对象的查询进行审计

**配置说明：**

参数audit\_dml\_state\_select决定是否对数据库对象的SELECT操作进行审计，如果开启此选项可以追溯用户对数据库的查询操作。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_dml\_state\_select**为1，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_dml\_state\_select';**

**推荐值：**1

**风险等级：**低

**参考：**无

### 配置审计优先策略

**配置说明：**

控制审计日志的保存策略，以空间还是时间限制为优先策略。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_resource\_policy**为on，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_resource\_policy';**

**推荐值：**on

**风险等级：**低

**参考：**无

### 配置单个审计文件的最大记录时间

**配置说明：**

参数audit\_rotation\_interval设置单个审计日志文件的最长日志记录时间。如果超过这个时间，将自动创建一个新的审计日志文件。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_rotation\_interval**为1440，单位为min，即为1天（1d），然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_rotation\_interval';**

**推荐值：**1440

**风险等级：**低

**参考：**无

### 配置单个审计日志文件的最大容量

**配置说明：**

参数audit\_rotation\_size设置单个审计日志文件最大的容量。当审计日志文件达到最大容量时，将自动创建一个新的日志文件。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_rotation\_size**为10240，单位为KB，即为10MB，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_rotation\_size';**

**推荐值：**10240

**风险等级：**低

**参考：**无

### 配置所有审计日志文件占用的最大磁盘空间

**配置说明：**

参数audit\_space\_limit设置审计文件所占的最大磁盘空间。当审计文件总量超过最大值时，则写入警告信息到数据库日志文件中，同时删除最早的审计文件，并记录审计文件删除信息到审计日志中。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_space\_limit**为1048576，单位为KB，即为1GB，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_space\_limit';**

**推荐值：**1048576

**风险等级：**中

**参考：**无

### 配置审计日志文件最小保存时间

**配置说明：**

参数audit\_file\_remain\_time设置审计日志最小保存时间。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_file\_remain\_time**为90，单位为天，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_file\_remain\_time';**

**推荐值：**90

**风险等级：**中

**参考：**无

### 配置审计日志文件的最大数目

**配置说明：**

参数audit\_file\_remain\_threshold设置审计日志文件最大数目。当审计文件总数超过指定值时，则警告信息写入数据库日志文件中，同时删除最早的审计文件，并记录审计文件删除信息到审计日志中。

**配置方法：**

在postgresql.conf配置文件中修改参数**audit\_file\_remain\_threshold**为1048576，然后重载数据库。

**vim ${GAUSSDATA}/postgresql.conf**

**vb\_ctl reload -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name = 'audit\_file\_remain\_threshold';**

**推荐值：**1048576

**风险等级：**中

**参考：**无

## 统一审计

### 开启安全审计并及记录日志

**配置说明：**

安全审计功能是对数据库用户行为进行记录、分析和阻断的功能。推荐开启安全审计功能和开启审计日志收集。

**配置方法：**

1. 修改rsyslog服务配置文件，添加以下内容。

vi /etc/rsyslog.conf

template(name="vbaudit" type="list") {

property(name="timereported" dateformat="year")

constant(value="-")

property(name="timereported" dateformat="month")

constant(value="-")

property(name="timereported" dateformat="day")

constant(value=" ")

property(name="timereported" dateformat="hour")

constant(value=":")

property(name="timereported" dateformat="minute")

constant(value=":")

property(name="timereported" dateformat="second")

constant(value="|")

property(name="msg")

constant(value="\n")

}

local0.\* /var/log/vbaudit.log;vbaudit

1. 重启rsyslog服务。

systemctl stop rsyslog.service

systemctl start rsyslog.service

1. 创建审计日志文件。

touch /var/log/vbaudit.log

1. 修改数据库配置文件。

vi postgresql.conf

enable\_security\_policy=on

shared\_preload\_libraries = 'pg\_stat\_statements,security\_plugin'

1. 重启数据库。

Vb\_ctl restart

----结束

**检查方法：**

**show** enable\_security\_policy**;**

**推荐值：**on

**风险等级：**中

**参考：**无

### 对数据库对象的添加、删除、修改进行安全审计

**配置说明：**

通过创建审计策略，可审计数据库对象的添加、删除、修改操作，并记录到审计日志。可审计的对象类型包括：模式、表、视图和函数。

**配置方法：**

**CREATE AUDIT POLICY poltest\_1 PRIVILEGES CREATE,ALTER,DROP;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_1';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对表对象添加注释进行安全审计

**配置说明：**

通过创建审计策略，可审计表对象添加注释操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_2 PRIVILEGES COMMENT;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_2';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对表对象分析进行安全审计

**配置说明：**

通过创建审计策略，可审计表对象分析操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_3 PRIVILEGES ANALYZE;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_3';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对权限操作进行安全审计

**配置说明：**

通过创建审计策略，可审计表对象分析操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_4 PRIVILEGES GRANT,REVOKE;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_4';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对会话属性操作进行安全审计

**配置说明：**

通过创建审计策略，可审计会话属性操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_5 PRIVILEGES SET,SHOW;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_5';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对登陆注销等操作进行安全审计

**配置说明：**

通过创建审计策略，可审计登陆注销等操作，并记录到审计日志。

**配置方法：**

* 审计任何登陆事件

**CREATE AUDIT POLICY poltest\_6 PRIVILEGES LOGIN\_ANY;**

* 审计登陆成功事件

**CREATE AUDIT POLICY poltest\_7 PRIVILEGES LOGIN\_SUCCESS;**

* 审计登陆成功事件

**CREATE AUDIT POLICY poltest\_8 PRIVILEGES LOGIN\_FAILURE;**

* 审计登出（退出）事件

**CREATE AUDIT POLICY poltest\_9 PRIVILEGES LOGOUT;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_6';**

**select \* from gs\_auditing\_policy where polname = 'poltest\_7';**

**select \* from gs\_auditing\_policy where polname = 'poltest\_8';**

**select \* from gs\_auditing\_policy where polname = 'poltest\_9';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对检查点事件进行安全审计

**配置说明：**

通过创建审计策略，可审计检查点事件操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_10 PRIVILEGES CHECKPOINT;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_10';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对事务提交回滚进行安全审计

**配置说明：**

通过创建审计策略，可审计检查点事件操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_11 PRIVILEGES COMMIT,ROLLBACK;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_11';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对表数据插入、更新、删除、查询进行安全审计

**配置说明：**

通过创建审计策略，可审计表数据的插入、更新、删除、查询操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_12 ACCESS INSERT,UPDATE,DELETE,SELECT;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_12';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对复制表数据进行安全审计

**配置说明：**

通过创建审计策略，可审计复制表数据操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_13 ACCESS COPY;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_13';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对函数执行进行安全审计

**配置说明：**

通过创建审计策略，可审计函数执行操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_14 ACCESS EXECUTE;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_14';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对表重建索引进行安全审计

**配置说明：**

通过创建审计策略，可审计表重建索引操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_15 ACCESS REINDEX;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_15';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对清空表数据进行安全审计

**配置说明：**

通过创建审计策略，可审计表重建索引操作，并记录到审计日志。

**配置方法：**

**CREATE AUDIT POLICY poltest\_16 ACCESS TRUNCATE;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_16';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对指定的对象进行安全审计

**配置说明：**

通过创建资源池和审计策略，可审计指定数据库对象的指定操作。可审计的对象类型包括：模式、表、视图和函数。

**配置方法：**

**CREATE RESOURCE LABEL l1 ADD TABLE (t1);**

**CREATE AUDIT POLICY poltest\_17 ACCESS TRUNCATE ON LABEL (l1);**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_17';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 对指定审计范围进行安全审计

**配置说明：**

通过创建审计策略，可指定属性来控制要审计的时间范围。每个审计策略包括以下属性：审计事件类型、IP地址、用户、客户端工具和响应动作。

**配置方法：**

**CREATE AUDIT POLICY poltest\_18 ACCESS TRUNCATE FILTER ON IP ('127.0.0.1') AND ROLES (vastbase\_test) AND APP (vsql);**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_18';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 指定安全审计的响应动作

**配置说明：**

通过创建审计策略，可指定安全审计的响应动作。响应动作包括：记录审计日志（普通日志和告警日志）和阻断会话，默认为记录审计日志（普通日志）。

**配置方法：**

* 响应动作为记录审计日志（普通日志）

**CREATE AUDIT POLICY poltest\_19 ACCESS TRUNCATE normally;**

* 响应动作为记录审计日志（告警日志）

**CREATE AUDIT POLICY poltest\_20 ACCESS TRUNCATE warning;**

* 响应动作为阻断会话

**CREATE AUDIT POLICY poltest\_21 ACCESS TRUNCATE block;**

**检查方法：**

**select \* from gs\_auditing\_policy where polname = 'poltest\_19';**

**select \* from gs\_auditing\_policy where polname = 'poltest\_20';**

**select \* from gs\_auditing\_policy where polname = 'poltest\_21';**

**推荐值：**无

**风险等级：**中

**参考：**无

## 错误报告和日志设置

### 开启日志收集器

**配置说明：**

参数logging\_collector控制日志收集。此后台进程抓取发送到stderr或csvlog的日志消息并写入日志文件。避免syslog信息遗漏。

**配置方法：**

在postgresql.conf配置文件中修改参数logging\_collector为on，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='logging\_collector' AND setting != 'on';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置日志名称

**配置说明：**

参数log\_filename设置服务器运行日志文件的名称。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_filename为“postgresql-%Y-%m-%d\_%H%M%S.log”，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_filename' AND setting NOT LIKE 'postgresql-%.log';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置日志文件权限

**配置说明：**

当开启参数logging\_collector时，可以使用参数log\_file\_mode设置服务器日志文件的权限。因为日志文件中可能含有敏感数据，所以日志文件访问必须受到限制。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_file\_mode为0600，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_file\_mode' AND setting != '0600';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 禁用同名日志文件覆写

**配置说明：**

参数log\_truncate\_on\_rotation设置Vastbase是否删除或者覆盖（而不是追加）已存在的同名日志文件。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_truncate\_on\_rotation为off，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_truncate\_on\_rotation' AND setting != 'off';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置单个日志文件最大记录时间

**配置说明：**

参数log\_rotation\_age设置单个日志文件的最长日志记录时间。当日志记录时间超过最大值后，服务器将自动创建新的日志文件。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_rotation\_age为1d，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_rotation\_age' AND setting != '1440';**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置单个日志文件最大容量

**配置说明：**

参数log\_rotation\_size设置单个日志文件的最大容量。当日志文件达到最大容量时，服务器将自动创建新的日志文件。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_rotation\_size为20MB，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_rotation\_size' AND cast(setting as int) != 20480;**

**推荐值：**无

**风险等级：**中

**参考：**无

### 配置客户端日志消息等级

**配置说明：**

参数client\_min\_messages控制消息按设置的等级发送到客户端。

有效的值包括DEBUG5、DEBUG4、DEBUG3、DEBUG2、DEBUG1、LOG、NOTICE、WARNING、ERROR、FATAL和PANIC，消息等级至少配置为NOTICE。每个等级都包括排在他后面的所有级别中的日志，级别越低，发送给客户端的消息就越少。在实际设置过程中，如果设置的级别大于error，例如fatal或panic，系统会默认将级别转为error。

**配置方法：**

在postgresql.conf配置文件中修改参数client\_min\_messages为notice，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='client\_min\_messages';**

**推荐值：**

"NOTICE" 或 "LOG"

**风险等级：**中

**参考：**无

### 配置服务器日志等级

**配置说明：**

参数log\_min\_messages控制消息按的等级写入到服务器日志。

有效的值包括DEBUG5、DEBUG4、DEBUG3、DEBUG2、DEBUG1、INFO、NOTICE、WARNING、ERROR、LOG、FATAL和PANIC，消息等级至少配置为WARNING。每个级别都包含排在他后面的所有级别中的信息，级别越低，服务器运行日志中记录的消息就越少。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_min\_messages为warning，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_min\_messages';**

**推荐值：**

"WARNING" 或 "NOTICE" 或 "INFO"

**风险等级：**中

**参考：**无

### 开启用户注销时日志记录功能

**配置说明：**

与log\_connections类似，在会话结束的时候在服务器日志里输出一行，并且包含会话持续时间。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_disconnections为on，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_disconnections' AND setting !='on';**

**推荐值：**on

**风险等级：**中

**参考：**无

### 配置导致错误条件的SQL语句服务器日志级别

**配置说明：**

参数log\_min\_error\_statement控制哪些引起错误的SQL语句记录到服务器日志中。对于大于等于当前配置安全等级的和SQL语句消息，会记录到日志里。

有效值包括DEBUG5、DEBUG4、DEBUG3、DEBUG2、DEBUG1、INFO、NOTICE、WARNING、ERROR、LOG、FATAL和PANIC，消息等级至少配置为ERROR。

由于有的SQL语句中包含用户个人信息，如果不需要记录出错的SQL语句，可以将log\_min\_error\_statement设置为PANIC。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_min\_error\_statement为error，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_min\_error\_statement';**

**推荐值：**

"ERROR"或 "FATAL"或"PANIC"

**风险等级：**中

**参考：**无

### 开启用户登录时日志记录功能

**配置说明：**

参数log\_connections可以记录每次尝试连接到服务器的日志，以及成功完成客户端连接认证的日志。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_connections为on，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_connections' AND setting !='on';**

**推荐值：**on

**风险等级：**中

**参考：**无

### 配置写入服务器的日志级别

**配置说明：**

参数log\_error\_verbosity控制写入服务器日志的级别。有效值包括TERSE、DEFAULT和VERBOSE。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_error\_verbosity为default，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_error\_verbosity';**

**推荐值：**"DEFAULT" 或 "VERBOSE"

**风险等级：**中

**参考：**无

### 配置日志不记录主机名

**配置说明：**

默认情况下，连接日志消息只显示连接主机的IP地址。若设置参数log\_hostname为on，则会记录主机名。由于解析主机名可能需要一定的时间，这个可能会带来性能开销。

**配置方法：**

在postgresql.conf配置文件中修改参数log\_hostname为off，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D ${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='log\_hostname' AND setting != 'off';**

**推荐值：**off

**风险等级：**中

**参考：**无

### 开启访问历史记录

**配置说明：**

参数vastbase\_login\_info控制用户登陆时是否记录用户的访问信息，并且在用户成功登陆后是否显示访问历史信息。开启后，用户可在系统表pg\_vastbase\_login\_info中查看用户历史访问记录。

**配置方法：**

在postgresql.conf配置文件中修改vastbase\_login\_info=true，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='**v**astbase\_login\_info' AND setting != 'true';**

**推荐值：**无

**风险等级：**中

## 可靠性和归档设置

### 配置WAL信息记录级别

**配置说明：**

wal\_level决定了写入WAL的信息量。最小的wal\_level不包含足够根据备份文件和WAL日志重建数据库的信息，所以必须使用archive或hot\_standby级别，并启用WAL归档(archive\_mode)和流复制。wal\_level至少配置为archive。

**配置方法：**

在postgresql.conf配置文件中修改参数wal\_level为archive，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='wal\_level';**

**推荐值：**

"archive" 或"hot\_standby"

**风险等级：**低

**参考：**无

### 开启归档模式

**配置说明：**

wal\_level决定了写入WAL的信息量。最小的wal\_level不包含足够根据备份文件和WAL日志重建数据库的信息，所以必须使用archive或hot\_standby级别，并启用WAL归档(archive\_mode)和流备份。

**配置方法：**

在postgresql.conf配置文件中修改archive\_mode=on，然后重启数据库。

**vim ${GAUSSDATA}/postgresql.conf**

其中GAUSSDATA为数据库主节点的data目录。

**vb\_ctl restart -D** **${GAUSSDATA}**

**检查方法：**

**SELECT name,setting FROM pg\_settings WHERE name='archive\_mode' AND setting != 'on';**

**推荐值：**无

**风险等级：**中

**参考：**无

## 数据存储加密

### 开启数据存储加密

**配置说明：**

vastbase采用软加密方式实现数据加解密功能，当用户向数据库写数据时，数据被自动加密；当用户从数据库查询数据时，数据被自动解密。

**配置方法：**

1、对密钥进行base64编码

生成存储加密密钥：

echo -n "12345abcde@ABCDE"|base64

生成主密钥：

echo -n "23456abcde@ABCDE"|base64

2、初始化实例（-K指定存储加密密钥，--masterkey指定主密钥）

vb\_initdb -D testdb --nodename='testdb' --algorithm=SM4-CTR-128 -K MTIzNDVhYmNkZUBBQkNERQ== --masterkey=MjM0NTZhYmNkZUBBQkNERQ==

3、配置密钥

gs\_encrypt -v ABCDEFGHIJKLMNOP -B MTIzNDVhYmNkZUBBQkNERQ== TRANS\_ENCRYPT\_SAMPLE\_STRING

4、将上一步结果写入配置文件中

Transparent\_encrypted\_string='上一步的结果'

5、启动实例

在出现提示“please input the master key:”时，输入masterkey：MjM0NTZhYmNkZUBBQkNERQ==

**检查方法：**

1、vsql登录数据库

修改初始化用户密码

alter user vastbase password 'Aa123456';

创建表并插入数据

vastbase=# create table test\_pwd (id int,name text);

CREATE TABLE

vastbase=# insert into test\_pwd values (1,'加密');

vastbase=# select \* from test\_pwd;

id | name

----+------

1 | 加密

(1 row)

2、进入数据目录（$PGDATA/base/）查看是否加密成功：

[klk@db1 15659]$ grep -iR '加密' ##没返回则表示加密成功;

**推荐值：**无

**风险等级：**中

**参考：**无



电话：010-82838118

地址：北京市海淀区学院路30号科大天工大厦B座6层

官网：www.vastdata.com.cn