生产环境权限管理项目方案

作者：jsntian
原文链接：https://ld246.com/article/1574397803004
来源网站：链滴

1 问题现状

1.1 员工使用root权限操作生产环境

当前我们公司内部服务器实例有上百台，管理客户的服务器也将近有上百个服务器示例。各个服务器的管理人员很多（开发、运维、项目、工程人员等），在大家登录使用Linux服务器时，不同职能的工对Linux系统的熟悉程度不同，因此导致操作很不规范，root权限泛滥（几乎所有员工都使用root限操作），经常导致文件等莫名其妙的丢失，服务器文件、目录权限混乱等问题。这样使得公司内部客户服务器安全存在很大的不稳定性、及操作安全隐患。据调查企业服务器环境，50%以上的安全问都来源于内部，而不是外部。为了解决以上问题，单个用户管理权限过大现状，现提出针对Linux服器用户权限集中管理的解决方案。

1.2 操作记录不可追溯

当前我们公司管理着几大平台和数十个小平台。我们的客服、工程、研发人员都是直接连接SSH的，中的操作不可过滤、不可审计、不可追溯。如果出现了操作失误，将无法追溯，不知道是谁操作的，知道什么时候操作的，不知道操作了什么。

1.3 程序使用root权限运行

所有自主程序和部分第三方开源程序（如redis）都使用root权限来运行，使用root权限存在非常大的全隐患。比如：

1. 程序原本的功能是想删除自身的日志文件，如果程序有BUG，拼接路径时考虑不周全，可能会导致删了系统文件。严重时可导致系统奔溃，数据丢失。

2. 程序使用了开源框架有远程执行命令的安全漏洞，那么黑客可能会获得系统的root权限，既可为所为，公司、客户数据安全则面临着巨大的威胁。

3. redis的弱密码、空密码存在很大的安全隐患，如使用root运行redis，黑客很容易就拿下操作系统的oot权限。

2 项目需求

2.1 生产环境限制使用root权限操作

对于员工使用root权限操作这一现状，我们希望超级用户root密码掌握在少数或唯一的管理员手中，希望多个系统管理员或相关有权限的员工，能够完成更多更复杂的自身职能相关的工作，又不至于越操作导致系统出现安全隐患。

为了实现这一目标，需要遵循4个最小化原则：

1） 安装软件最小化

2） 运行服务最小化

3） 文件目录权限最小化

4） 操作权限最小化

那么，如何解决多个系统管理员都能管理系统而又不让超级权限泛滥呢？这就需要sudo管理来替代或

结合su命令来完成这样的苛刻且必要的服务器用户管理需求。

2.2 引入堡垒机

对于操作记录不可追溯这一现状，引入堡垒机系统，我们希望所有能SSH连接的服务器都能统一接入堡垒机系统。堡垒机能实现以下功能：

操作审计

- 运维操作记录： 操作失误、恶意操作、越权操作详细记录。
- Linux命令审计： 可提取命令符审计，支持命令定点回放。
- Windows操作录像： 远程桌面的操作，支持全程录像，包括键盘操作、鼠标操作、窗口打开等。
- 文件传输审计： SFTP上传下载文件审计。

职权管控

通过账号管控和权限组管理，实现分职权进行人员和资产的管理。

- 账号管控： 运维账号唯一，解决共享账号、临时账号、滥用权限等问题。
- 权组管理： 按照人员、部门组织、资源组，建立人员职责与资源分配的授权管理。

安全认证

- 支持Google Auth二次认证机制，控制账号密码泄露风险，防止运维人员身份冒用和复用。

高效运维

- 支持Web Terminal，无需安装任何工具即可连接SSH
- 支持SSH、RDP、SFTP协议

2.3 禁止程序使用root权限运行

对于程序使用root权限运行这一现状，我们希望能使用普通用户的权限来运行程序，以避免使用root限所带来的安全隐患。

为了实现这一目标，需要遵循3个普通和1个加强法则：

1） 使用系统普通账号运行程序

2） 使用MySQL普通账号操作MySQL数据库

3） 使用MongoDB普通账号操作MongoDB数据库

4） 加强密码强度

3 具体实现

3.1 生产环境限制使用root权限操作

针对公司不同部门，根据员工的具体工作职能（例如：开发，运维，工程）等级、分层次的实现对linu服务器的权限最小化、规范化。这样即减少了运维管理成本，消除了安全隐患，又提高了工作效率，现了高质量的、快速化的完成项目进度，以及日常系统维护。

3.2 引入堡垒机

针对客服（项目经理、运维、运营、工程人员），我们希望将公司服务器、客户服务器纳入堡垒机管，所有的操作都应经过堡垒机。

3.3 禁止程序使用root权限运行

针对自主程序，统一使用app（固定ID为999）用户运行。如果某些操作需要root权限操作，则使用sdo提权。

针对第三方程序，如使用rpm包安装的应用，则保持原有的用户不变，其它的统一使用app（固定ID999）用户运行。

4 限制使用root权限操作实施方案

4.1 信息采集

1. 召集相关各部门领导通过会议讨论或是与各组领导沟通权限管理方案的可行性。
2. 确定方案可行性后，会议负责人汇总、提交、审核所有相关员工对linux服务器的权限需求。
3. 按照需要执行的linux命令程序及公司业务服务来规划权限和人员对应配置。
4. 权限方案一旦实施后，所有员工必须通过《员工Linux服务器管理权限申请表》来申请对应的权限确定审批流程，规范化管理。

4.2 收集员工职能和对应权限

此过程是召集大家开会确定，或者请各领导安排人员进行统计汇总，员工及对应的职责，交给运维人，由运维人员优化职位所对应的系统权限。

命令用全路径，命令与命令之间用逗号隔开。

1） 运维组

级别	权限
初级运维	查看系统信息，查看和修改网络配置，看日志文件 /usr/bin/free,/usr/bin/iostat,/usr/bin/top,/bin/hostname,/sbin/ifconfig, /bin/netstat,/sbin/roue
高级运维	查看系统信息，查看和修改网络配置，程管理，软件包管理，存储管理，查看日志文件 /usr/bin/free,/usr/bin/iostat,/usr/bin/top,/bin/hostname,/sbin/ifconfig, /bin/netstat,/sbin/roue,/sbin/iptables,/etc/init.d/network,/bin/nice, /bin/kill,/usr/bin/kill,/usr/bin/killall,/bin/rpm,/usr/bin/yum,/bin/date, /sbin/fdisk,/sbin/sfdisk,/sbin/parted,/sbin/partprobe, /sbin/partx, /bin/mount,/bin/umount

运维组长 所有权限

ALL

2） 开发组

级别	权限
初级开发	查看对应服务的日志文件
/usr/bin/tail /home/log/,/usr/bin/tail /home//logs/,/bin/grep /home/log/,/bin/cat /home/*,/bin/s	
高级开发	查看对应服务的日志文件，重启对应服务
/sbin/service,/sbin/chkconfig,/usr/bin/tail /home/log/,/usr/bin/tail /home//logs/,/bin/grep /home/log/,/bin/cat /home/*,/bin/ls	
开发组长	所有权限，但不能修改root密码
ALL,/usr/bin/passwd [A-Za-z]*,!/usr/bin/passwd root, !/usr/sbin/visudo,!/bin/vi,!/usr/sbin/vim !/bin/su	

3） 工程

级别	权限
工程	查看对应服务的日志文件
/usr/bin/tail /home/log/,/usr/bin/tail /home//logs/,/bin/grep /home/log/,/bin/cat /home/*,/bin/s	

4.3 配置阶段

此步骤仅是一个简单的配置示例，不表示任何真实环境。

1） 新建组，如新建运维组如下：

```
groupadd -g 1000 ops
```

2） 根据收集到的员工信息，给每个员工新建一个专用账号，并将员工加到对应的组，如：

```
useradd -g 1000 -u 1001 rz-chen

useradd -g 1000 -u 1002 zt-lin
```

3） 新建命令别名

```
Cmnd_Alias JUNIOR_OPS_CMD=/usr/bin/free,/usr/bin/iostat,/usr/bin/top,/bin/hostname,/sbi/ifconfig,/bin/netstat,/sbin/route
```

4） 配置用户别名

```
User_Alias JUNIOR_OPS=rz-chen,zt-lin
```

5） 配置对应的权限

```
JUNIOR_OPS ALL=(root) JUNIOR_OPS_CMD
```

4.4 邮件通知

发邮件通知各位员工，必要时可以加以培训讲解。

4.5 制定权限申请流程及申请表

见单独文档：《员工Linux服务器管理权限申请表》

4.6 配置规范

4.6.1 名称定义或缩写

1. 级别

a) 初级：JUNIOR

b) 中级：INTERMEDIATE

c) 高级：SENIOR

2. 职位

a) 开发：DEV

b) 运维/运营：OPS

c) 测试：QA

d) 工程：NG

4.6.2 用户/组配置规范

l 组应以职位或职位缩写命名，如运维：ops

l 用户应以姓名拼音或缩写命名，如：rz-chen

l 用户归属的组应体现在职位上，如：rz-chen归属ops组

l 组ID从1000起，下一个组ID为1100，以此类推。如：ops组ID为1000，pm组ID为1100

l 用户ID应从组ID+1起，如：rz-chen和zt-lin归属于ops组，用户ID分别为1001和1002

4.6.3 sudoers配置规范

l 所有的自定义配置应放在 /etc/sudoers.d/ 下，尽量不修改 /etc/sudoers

l 以组为单位进行配置，如：所有关于ops组的配置，应配置到 /etc/sudoers.d/ops 文件内

l 配置文件的权限应为 440（-r--r-----）

4.6.4 命名规范

1. 命令别名（Cmnd_Alias）

字符全部大写，固定格式：级别_职位_CMD。如：

Cmnd_Alias JUNIOR_OPS_CMD=/usr/bin/free……

2. 用户别名（User_Alias）

字符全部大写，固定格式：级别_职位。如：

User_Alias JUNIOR_OPS=rz-chen,zt-lin

5 引入堡垒机实施方案

堡垒机系统早在2018年6月份已经搭建，由于员工已经习惯使用xshell连接服务器，堡垒机系统没有到广泛使用。本次实施方案，运维人员将给出一些措施，一步步将堡垒机推广开来，让运维工程师、营工程师、项目经理、销售工程师等都使用起来，而且堡垒机将是客户服务器唯一的连接入口。

5.1 推动堡垒机

1） 向目标受众（运维工程师、运营工程师、项目经理、销售工程师）宣贯引入堡垒机的好处

2） 宣贯案例，使用root权限所引起的严重后果

3） 告知目标受众，一星期后将不允许直接连接服务器SSH

5.2 禁止非堡垒机连接SSH

配置公司防火墙策略，禁止非堡垒机地址连接服务器SSH端口。

5.3 回收堡垒机root权限

回收操作root权限，启用普通权限账号：op。这个账号将只能做简单的查看系统状态，查看日志文件有限权限。

6 禁止程序使用root权限运行实施方案

该实施方案最终的实施者是开发组，运维人员协助开发进行权限的整理和设计。细节将不在此处说明。

目标是：

1) 禁止使用操作系统的root用户权限运行程序

2) 禁止使用MySQL的root用户权限连接MySQL数据库

3) 禁止使用MongoDB的root用户权限连接MongoDB数据库

4) 禁止文件、目录权限滥用