# 02-2、提权

# 永久提权：Switching users with su

`su`（Switch User）是 Linux/Unix 系统中用于切换用户身份的核心命令，常用于临时获取目标用户的权限（尤其是 `root` 用户）。以下是其详细用法、选项及实际场景分析。

```
[abc@xnha ~]$ id abc
uid=1006(abc) gid=1007(abc) 组=1007(abc)
[abc@xnha ~]$ whoami
abc
[abc@xnha ~]$ useradd user999
useradd: Permission denied.                #权限不够
useradd：无法锁定 /etc/passwd，请稍后再试。
[abc@xnha ~]$ su - root
密码：
[root@xnha ~]# useradd user999
[root@xnha ~]#

成功
```

---

## 1. 基本语法

bash

```
su [选项] [用户名]
```

- **若不指定用户名**，默认切换到 `root` 用户。
- 常用选项：
  - `-` 或 `-l`：模拟完整登录（加载目标用户的环境变量）。
  - `-c "命令"`：以目标用户身份执行单条命令后退出。
  - `-s`：指定 Shell（如 `-s /bin/bash`）。

---

## 2. 核心功能与示例

### 2.1 切换到 root 用户

bash

```
su - root         # 完整登录，加载 root 环境变量（推荐）
su root           # 仅切换用户身份，环境变量不变
su -              # 等效于 su - root
[root@xnha ~]# su user01
[user01@xnha root]$ echo $PATH
/home/user01/.local/bin:/home/user01/bin:/usr/local/bin:/usr/local/sbin:/usr/bin
:/usr/sbin:/root/bin
[user01@xnha root]$ su root
密码：
```

```
[root@xnha ~]# echo $PATH
/home/user01/.local/bin:/home/user01/bin:/usr/local/bin:/usr/local/sbin:/usr/bin
:/usr/sbin:/root/bin
[root@xnha ~]# su - root
[root@xnha ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@xnha ~]#
```

**输入密码后**，终端提示符变为 `root@hostname`。

---

#### 2.2 切换到其他用户

bash

```
su - alice       # 切换到 alice 用户（需 alice 的密码）
su alice         # 仅切换身份，不加载环境变量
```

- root 用户切换到其他用户无需密码：

  bash

  ```
  su - alice      # root 用户无需输入密码
  ```

---

#### 2.3 执行单条命令后退出

bash

```
[root@xnha ~]# su - user01 -c "whoami"
user01
```

---

## 3. 关键细节解析

#### 3.1 `su` vs `su -`

| 命令 | 用户身份 | 环境变量 | 工作目录 |
| --- | --- | --- | --- |
| `su root` | root | 原用户 | 原目录 |
| `su - root` 或 `su -` | root | root | root 的 home 目录 |

---

#### 3.2 安全注意事项

- **密码暴露风险**：
  `su` 需要输入目标用户的密码，若多人共用 root 密码，难以审计操作来源。
  **推荐替代方案**：使用 `sudo` 实现权限最小化和操作审计。
- **禁用 root 登录**：
  在安全要求高的场景，可通过以下方式禁用 root 的 `su` 切换：

bash

```
# 编辑 /etc/pam.d/su
auth required pam_wheel.so use_uid
# 仅允许 wheel 组成员使用 su
```

---

## 4. 常见问题解答

### 4.1 如何退出 su 切换的用户?

- 输入 `exit` 或按 `Ctrl+D` 退回原用户。
- ```
  [root@xnha ~]# exit
  注销
  ```

### 4.2 切换用户后环境变量错误?

- 使用 `su -` 而非 `su`，确保加载完整环境。

### 4.3 普通用户无法切换到 root?

- 可能原因：
  1. root 密码错误。
  2. root 用户被禁用（如 `/etc/passwd` 中 Shell 设为 `/sbin/nologin`）。
  3. PAM 策略限制（如仅允许特定用户组使用 `su`）。

---

## 5. 实际应用场景

### 5.1 系统维护

bash

```
su - root -c "apt upgrade && apt autoremove"  # 以 root 身份更新系统
```

### 5.2 多用户协作

bash

```
su - alice   # 切换到开发人员 alice 的账户调试环境
```

### 5.3 脚本中临时提权

bash

```
#!/bin/bash
# 以 root 身份创建目录
su - root -c "mkdir /opt/custom_app"
chown user:user /opt/custom_app
```

---
```

### 6. 总结

| 场景 | 推荐命令 |
| --- | --- |
| 临时切换用户并加载环境 | `su - 用户名` |
| 执行单条特权命令 | `su -c "命令" root` |

合理选择 `su` 或 `sudo`，在便利性和安全性之间找到平衡。

---

# 临时提权：Running commands as root with sudo

## `sudo` 命令

`sudo`（Super User Do）是 Linux/Unix 系统中用于临时以特权身份（如 `root`）执行命令的核心工具，相比 `su` 更安全且支持细粒度权限控制。以下是其核心概念、配置及实际应用指南。

---

### 1. 核心特性

- **无需共享密码**：验证当前用户密码，而非目标用户（如 `root`）密码。
- **最小权限原则**：可限制用户仅能执行特定命令。
- **操作审计**：所有 `sudo` 操作记录在 `/var/log/auth.log` 或 `/var/log/secure`。
- **超时机制**：默认 15 分钟内无需重复输入密码（可配置）。

---

### 2. 基本语法

bash

```
sudo [选项] 命令
```

**常用选项**：

- `-u 用户`：以指定用户身份执行（默认 `root`）。
- `-i`：模拟完整登录（加载目标用户环境变量）。
- `-s`：启动目标用户的 Shell。
- `-l`：列出当前用户允许执行的命令。

---

### 3. 工作流程

1. **权限检查**：
   系统读取 `/etc/sudoers` 文件，验证用户是否有权执行该命令。
2. **密码验证**：
   输入当前用户密码（若未配置 `NOPASSWD`）。
3. **命令执行**：
   以目标用户身份执行命令，完成后权限自动回收。

---

## 4. 配置文件 `/etc/sudoers`

使用 `visudo` 命令编辑（避免语法错误）：

bash

```
visudo  # 安全编辑方式
```

### 4.1 配置语法

bash

```
用户/组    主机=(可切换身份)    可执行的命令 [参数]
```

- 字段说明：
  - **用户/组**：`%` 开头表示组（如 `%admin`）。
  - **主机**：ALL 表示所有主机。
  - **可切换身份**：ALL 表示任意用户，可指定为 `(root)`。
  - **命令**：需使用绝对路径（如 `/usr/bin/apt`）。

### 4.2 示例规则

bash

```
# 允许 wheel 组成员以 root 身份执行所有命令
%wheel  ALL=(ALL)  ALL
william  ALL=(ALL)  ALL
# 允许用户 alice 无需密码重启 Apache
alice   ALL=(root)  NOPASSWD: /usr/bin/systemctl restart httpd

# 允许用户 bob 在特定主机上管理用户
bob     host01=(root) /usr/sbin/useradd, /usr/sbin/userdel
```

### 4.3 实验

目的：使一个新用户user777具有删除用户的权限

- 方法一：将新用户加入wheel组

```
添加新用户user777,尝试使用sudo提权
[root@xnha ~]# useradd user777
[root@xnha ~]# passwd user777

切换用户user777,使用userdel命令删除test用户
[user777@xnha ~]$ userdel test
userdel: Permission denied.
userdel：无法锁定 /etc/passwd，请稍后再试。

使用sudo 提权尝试删除test用户
[user777@xnha ~]$ sudo userdel test
我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：
    #1) 尊重别人的隐私。
    #2) 输入前要先考虑(后果和风险)。
```

```
    #3) 权力越大，责任越大。
[sudo] user777 的密码：
user777 不在 sudoers 文件中。此事将被报告。                #没有权限使用sudo,只有在/etc/soduers中的用户才可以

新建用户user888，并添加到wheel组
[root@xnha ~]# useradd user888 -aG wheel
[root@xnha ~]# id user888
uid=1506(user888) gid=1507(user888) 组=1507(user888),10(wheel)
[root@xnha ~]# passwd user888

切换用户user888，别用su
[user888@xnha ~]$ sudo userdel -r test

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：

    #1) 尊重别人的隐私。
    #2) 输入前要先考虑(后果和风险)。
    #3) 权力越大，责任越大。

[sudo] user888 的密码：
[user888@xnha ~]$ grep test /etc/passwd        #未找到test用户，已被删除
```

- 方法二：编辑/etc/sudoers 添加特权

```
新建用户user999,重复方法一中user777的操作，发现无法删除test用户
编辑 /etc/sudoers 文件，将用户user999添加 userdel特权 （编辑前，先将文件进行备份！！！）
[root@xnha ~]# cp /etc/sudoers /tmp/sudoers
[root@xnha ~]# visudo
user999 ALL=(root) /usr/sbin/userdel,/usr/sbin/useradd

切换用户999，删除用户test
[user999@xnha ~]$ userdel -r test
userdel: Permission denied.
userdel：无法锁定 /etc/passwd，请稍后再试。
[user999@xnha ~]$ sudo userdel -r test

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：
    #1) 尊重别人的隐私。
    #2) 输入前要先考虑(后果和风险)。
    #3) 权力越大，责任越大。
[sudo] user999 的密码：
userdel：/var/spool/mail/test 并不属于 test，所以不会删除
userdel：/home/test 并不属于 test，所以不会删除
#删除成功
```

## 5. 实际应用场景

### 5.1 普通用户执行特权命令

bash

```
# 更新软件包列表
sudo apt update

# 查看系统日志（需 root 权限）
sudo tail -f /var/log/syslog
```

### 5.2 以其他用户身份执行命令

bash

```
# 以用户 mysql 身份启动进程
sudo -u mysql /usr/bin/mysqld_safe
```

### 5.3 进入特权 Shell 环境

bash

```
sudo -i   # 切换到 root 的完整环境（加载变量）
sudo -s   # 启动 root 的 Shell（保持当前目录）
```

### 5.4 查看用户权限

bash

```
sudo -l  # 列出当前用户可执行的命令
```

---

## 6. 安全最佳实践

1. **最小权限原则**：
   - 避免赋予 `ALL` 权限，按需授权特定命令。
   - 示例：仅允许用户管理服务：

     ```
     user1  ALL=(root) /usr/bin/systemctl restart nginx, /usr/bin/systemctl
     status nginx
     ```

2. **禁用密码（谨慎使用）**：

   ```
   # 允许用户无需密码执行命令
   user2  ALL=(root)  NOPASSWD: /usr/bin/apt update
   ```

3. **日志监控**：
   - 检查 `/var/log/auth.log` 或 `/var/log/secure`，追踪可疑操作。
4. **超时配置**：
   - 修改默认超时时间（单位：分钟）：

bash

```
Defaults timestamp_timeout=5  # 5 分钟后需重新输入密码
```

5. **禁用 root 登录**：
   - 结合 `sudo` 使用，增强安全性：

     bash

     ```
     # 禁用 root SSH 登录（/etc/ssh/sshd_config）
     PermitRootLogin no
     ```

---

## 7. 常见问题解决

### 7.1 用户无法使用 sudo

- **原因**：用户未在 `/etc/sudoers` 中配置。
- 解决：
  1. 使用 `root` 用户添加规则：

     bash

     ```
     usermod -aG wheel user3  # 将用户加入 wheel 组（需组已授权）
     ```

  2. 或直接编辑 `/etc/sudoers`：

     bash

     ```
     user3  ALL=(ALL)  ALL
     ```

### 7.2 命令路径不匹配

- **错误**：`Sorry, user user4 is not allowed to execute '/bin/systemctl restart httpd' as root on host01.`
- 解决：在规则中指定命令的绝对路径：

  bash

  ```
  user4  host01=(root)  /usr/bin/systemctl restart httpd
  ```

### 7.3 避免语法错误

- **风险**：直接编辑 `/etc/sudoers` 可能导致配置失效。
- 解决：始终使用 `visudo` 命令：

  bash

  ```
  sudo visudo  # 自动检查语法
  ```

---

## 8. 高级用法

### 8.1 限制命令参数

bash

```
# 允许 user5 仅能添加用户（禁止删除）
user5  ALL=(root)  /usr/sbin/useradd
user5 ALL=(root) /usr/sbin/useradd [a-z][a-z0-9_-]*
```

---

## 9. 总结对比：`sudo` vs `su`

| 特性 | `sudo` | `su` |
| --- | --- | --- |
| **密码验证** | 当前用户密码 | 目标用户密码 |
| **权限粒度** | 精细（命令级） | 粗放（用户级） |
| **日志审计** | 详细记录 | 无 |
| **安全性** | 高（无需共享 root 密码） | 低（依赖密码共享） |
| **典型场景** | 临时特权操作 | 长期切换用户身份 |

---

通过合理配置 `sudo`，可以在保障系统安全的前提下，高效完成日常管理任务。始终遵循最小权限原则，并定期审计权限分配。