

User Management

Debian Linux: local accounts, authentication, lifecycle, and sudo policy

tsaimh (2022-2026, CC BY-SA)

lctseng (2019-2021, CC BY-SA)

? (1996-2018)

What You Will Learn

- Explain UID, primary GID, and supplementary groups.
- Read account records without exposing password hashes.
- Create, verify, disable, and remove local accounts safely.
- Distinguish password locking from revoking all access.
- Use su and sudo with an explicit privilege policy.

Course platform

Debian 13 (trixie). Examples use local accounts; NSS, PAM, and site policy can change effective access.

1. Account Records and Identity

Separate numeric identity, account metadata, authentication, and authorization.

UID, GID, and Supplementary Groups

```
$ id alice
uid=1001(alice) gid=1001(alice)
groups=1001(alice),2000(lab)

$ id root
uid=0(root) gid=0(root) groups=0(root)
```

- File ownership and process credentials use numeric IDs.
- Each account has a primary group and may belong to supplementary groups.
- UID 0 carries superuser identity regardless of the login name.

Illustrative output

Names and numbers above are examples. Avoid assigning UID 0 to another regular account.

Regular, System, and Directory Accounts

Account type	Purpose	Typical practice
Regular user	A person with an interactive session	Individual identity and home directory
Service/system user	Run one daemon with limited privileges	Dedicated identity and restricted login
Directory user	Centrally managed identity	Resolve through NSS/directory integration

Historical names

daemon, bin, sys, and nobody have platform-specific roles. Prefer a dedicated identity for each service.

/etc/passwd: Seven Fields

```
alice:x:1001:1001:Alice Example:/home/alice:/bin/bash
```

```
name:password-marker:uid:gid:gecos:home:shell
```

Field	Meaning
name, UID, GID	Login name and numeric identity
password marker	x refers to the protected shadow record on Debian
GECOS	Descriptive account information
home, shell	Home directory and configured login program

Lookup scope

getent passwd alice can include directory accounts. A local /etc/passwd file may not list every user.

Protected Password Records

Platform	Protected record	Management tools
Debian	/etc/shadow	passwd, adduser, usermod, vipw

FreeBSD comparison	/etc/master.passwd and databases	pw, passwd, vipw
--------------------	----------------------------------	------------------

- Stored password values are salted password hashes, not reversible encrypted passwords.
- Public account metadata and protected authentication records serve different purposes.
- Use account tools to keep records and derived databases consistent.

Teaching examples

Use redacted values in examples. Avoid printing or copying real password hashes into reports.

Password Hash Formats

Format / prefix	Teaching point
DES, MD5 (\$1\$)	Legacy formats unsuitable for new password storage
bcrypt (\$2...\$)	Cost-based hash, with implementation-specific input limits
SHA-512 crypt (\$6\$)	Salted format, configurable rounds
yescrypt (\$y\$)	Used by modern Debian password stacks; verify local PAM policy

Policy varies

Defaults and supported hashes depend on the OS, release, and PAM/crypt implementation. Do not hard-code historical defaults.

GECOS and the Login Shell

```
# Change personal account metadata  
chfn
```

```
# Inspect supported shells first  
cat /etc/shells  
chsh -s /bin/sh
```

- GECOS often stores a full name and optional contact fields.
- Only record personal details that the account policy needs.
- The login shell is a configured program, not necessarily the shell currently running.
- On Debian, verify the shell path and /etc/shells before changing it.

Shell restrictions

nologin blocks ordinary shell login but does not, by itself, describe every possible authentication path.

/etc/group and Group Membership

```
lab:x:2000:alice,bob
```

```
name:password-marker:gid:members
```

```
id alice
```

```
groups alice
```

- The member list describes supplementary membership.
- The primary group comes from the account record and may not list the user explicitly here.
- Existing processes retain their credentials. Test membership in a new login session.

Privilege groups

FreeBSD wheel membership is relevant to the default su policy. Linux su/sudo rules depend on local PAM and sudoers policy.

NSS and PAM on Debian

Layer	Role	Common configuration
NSS	Resolve users and groups	/etc/nsswitch.conf
PAM auth / account	Verify credentials and account policy	/etc/pam.d/
PAM session	Set up the login session	Service-specific PAM stack

- getent shows identities resolved through NSS.
- A passwd entry is not proof that every login method is allowed.

Keep the layers distinct

Identity lookup, authentication, account restrictions, and session setup have different jobs.

Resource Limits on Debian

```
# /etc/security/limits.d/90-lab.conf
@lab soft nfile 1024
@lab hard nfile 4096

# In a new applicable login session
ulimit -Sn
ulimit -Hn
```

- pam_limits applies limits when the PAM session stack includes it.
- Soft limits may be raised up to the hard limit.
- For systemd services, configure LimitNOFILE= or other unit resource controls.

Different entry points

PAM login limits do not automatically apply to every systemd service. Verify the effective process limits.

Debian shadow and Account Aging

```
alice:<hash>:20000:0:99999:7:::
```

```
# Inspect aging without displaying the hash  
sudo chage -l alice
```

Field group	Meaning
1-2	Login name and password hash/marker
3-6	Last change, minimum age, maximum age, warning days
7-9	Inactive days, account expiry date, reserved field

Different deadlines

Password expiration asks for a password change. Account expiration blocks access according to the authentication policy.

2. Account Lifecycle

Use the platform tools and verify the resulting account, ownership, and login behavior.

Creating a Local Account

- Choose a unique login name and the intended primary/supplementary groups.
- Use the account tool to create records, the home directory, and skeleton files.
- Set the initial authentication method through an approved channel.
- Apply quota, expiration, and privilege policy where needed.
- Verify identity, home ownership, shell, and actual login behavior.

Example convention

Commands use the fictional account `alice`. Administrative examples require root or authorized sudo access.

Debian: adduser and addgroup

```
# Create the teaching group
sudo addgroup lab

# Interactive user creation
sudo adduser alice

# Append supplementary membership
sudo adduser alice lab
getent passwd alice
id alice
```

- adduser is Debian's usual interactive administration helper.
- It applies local defaults and creates the home and skeleton files.
- Verify the identity, shell, home, and actual login.

Initial privileges

Do not add every new account to sudo. Grant administrative privileges only for the assigned work.

Debian: useradd as an Alternative

```
# Alternative to adduser, not a second step
sudo useradd -m -s /bin/bash alice
sudo passwd alice

getent passwd alice
id alice
stat -c "%U %G %a %n" /home/alice
```

- useradd is the lower-level account tool.
- Use explicit options when its defaults matter.
- Choose one creation path for each account.

Defaults to inspect

/etc/default/useradd and /etc/login.defs affect behavior. Compare results rather than assuming the tools are identical.

Supplementary Group Changes

```
# Append; do not replace the group list
sudo usermod -aG lab alice

# Debian helper alternative
# sudo adduser alice lab

# Verify in a new login session
id
groups
```

- -aG appends supplementary groups.
- -G without -a replaces the supplementary group list.
- id alice shows the account database; id in a session shows that process's credentials.

Refresh the session

Existing processes do not automatically acquire newly added groups. Open a new login and verify both views.

Home Directories and Skeleton Files

Debian location	Purpose
/etc/skel	Templates copied into a new home
/home/<user>	Usual home path for regular local users
.profile, .bashrc	Startup files with different shell/login contexts

- Account tools copy templates for new users. Later skeleton changes do not automatically update existing homes.
- Verify directory owner, group, permissions, and the configured home path.
- Use chown recursively only when the scope and existing contents are understood.

Connection to Shells

Startup files depend on the shell and session type. Keep configuration in the appropriate file.

Passwords and Filesystem Quotas

```
# Set the password interactively  
sudo passwd alice  
  
# After quota support is configured  
sudo edquota -u alice  
quota -u alice
```

- Filesystem quotas require support, mount policy, and initial setup.
- For an ext4 lab, verify quota configuration before using edquota.
- Soft limits allow a grace period; hard limits block more allocation.
- Block and inode limits control different resources.

Filesystem-specific setup

Do not copy FreeBSD UFS recipes. ext4, XFS, and Btrfs have different quota and recovery mechanisms.

Account Verification Checklist

Check	Evidence
Identity	id alice and the resolved account record
Membership	Primary group plus expected supplementary groups
Home	Correct path, owner, mode, and skeleton files
Authentication	A test login through each approved entry point
Policy	Expiration, quota, and sudo permissions
Audit	Creation record and responsible owner

Successful creation

An account-file entry alone does not prove that login, home access, or policy enforcement works.

Password Locking and Login Revocation

```
# Lock password authentication  
sudo passwd -l alice  
sudo passwd -S alice  
  
# Unlock only after policy review  
# sudo passwd -u alice
```

- Locking a password does not automatically revoke SSH keys or terminate sessions.
- Review account expiry, keys, tokens, directory policy, jobs, and active sessions.
- Use account tools; do not comment out passwd or shadow lines.

Revocation checklist

Test every approved entry point and handle existing processes and credentials explicitly.

Account Offboarding and Removal

```
# After access revocation and backup  
# Retain home by default  
sudo deluser alice  
  
# Lower-level alternative  
# sudo userdel alice  
  
# Delete home only after approval  
# sudo deluser --remove-home alice
```

- Inventory ownership, mail, cron jobs, keys, processes, and external credentials.
- Verify a backup and revoke access before removing the local record.
- Handle remaining files by old numeric UID before UID reuse.

Deletion scope

deluser options and local configuration determine data removal. Removing an account does not erase all its files.

3. Administrative Privileges

Use named accounts and grant only the operations required by the role.

The Superuser and Its Boundaries

- UID 0 grants broad authority over files, processes, devices, and system configuration.
- Examples include network changes, shutdown, ownership changes, and resource-limit adjustments.
- Linux capabilities and namespace boundaries can divide or constrain that authority.
- FreeBSD jails, securelevel, and security policies can also restrict operations.

Operational practice

Use a named account for daily work and elevate for a specific administrative task.

Root Login and SSH Policy

```
# Example SSH policy
PermitRootLogin no

# Check syntax before applying changes
sudo /usr/sbin/sshd -t
```

- Root console access and root SSH access have separate policy controls.
- PermitRootLogin no disables direct root SSH login.
- prohibit-password still allows permitted non-password methods such as public keys.
- Verify a named administrator can log in and elevate before restricting root access.

Configuration precedence

Include files and Match blocks can affect the effective policy. Keep an existing session and test a new connection.

su, su -, and sudo

Command	Purpose	Policy context
su alice	Switch identity while retaining much of the environment	Authentication and PAM/platform policy
su - alice	Start a login-style environment for the target	Target login settings and shell
sudo command	Run an authorized operation as another identity	sudoers or configured policy
sudo -u user command	Choose an explicit target identity	Allowed Runas user and command

Environment detail

Exact environment handling depends on implementation and configuration. Do not assume every variable changes or stays unchanged.

visudo and Policy Validation

```
# Safely edit and validate sudoers
sudo visudo
sudo visudo -c

# Inspect your effective permissions
sudo -l
```

- visudo locks the file and checks syntax before installation.
- Debian uses /etc/sudoers and included /etc/sudoers.d files.
- Inspect effective privileges with sudo -l.
- sudo logs activity through the configured logging facilities.

Debian policy files

Use /etc/sudoers and its included /etc/sudoers.d files. Validate with visudo and inspect effective permissions.

sudoers: User, Host, Runas, Command

```
# user    host = (runas-user) command with arguments
alice    ALL  = (root) /usr/bin/id -u
```

```
# Group names use a leading %
%lab     ALL  = (root) /usr/bin/id -u
```

Part	Question answered
User or %group	Who may invoke sudo?
Host	On which hosts does the rule apply?
Runas	As which identity may the command run?
Command and arguments	Which operation is allowed?

Example scope

Verify executable paths on the target host. A command-only entry without arguments can permit arbitrary arguments.

Aliases and a Narrow Allowlist

```
User_Alias LAB_TA = alice, bob  
Runas_Alias REPORT = nobody  
Cmnd_Alias WHOAMI = /usr/bin/id -u
```

```
LAB_TA ALL = (REPORT) WHOAMI
```

```
# Test the allowed operation  
sudo -u nobody /usr/bin/id -u
```

- Aliases name a set of users, hosts, target users, or commands.
- A narrow allowlist is easier to review than granting ALL and subtracting a few programs.
- Test both allowed and rejected invocations.

Service identities

nobody is used here only for demonstration. Real services should normally have dedicated identities.

Command Restrictions and Escape Paths

- Editors, pagers, interpreters, and some utilities can launch other commands.
- An allowed program may offer file writes or options that lead to more authority than expected.
- Denying a few shell paths does not constrain an account that otherwise has ALL privileges.
- Review arguments, writable executables, configuration files, and the program's full behavior.

Practical policy

For complex operations, use a reviewed root-owned helper with fixed behavior and tightly constrained inputs.

Password Prompts and Administrative Roles

```
# Broad admin rule, password required
%sudo ALL=(ALL:ALL) ALL

# Narrow passwordless rule
alice ALL=(root) NOPASSWD: /usr/bin/id -u
```

- NOPASSWD skips a password prompt for the matched rule. It does not reduce the granted authority.
- Granting NOPASSWD: ALL provides broad unattended administrative access.
- Audit group membership and effective sudo rules together.

Debian administrator group

The usual broad rule is %sudo ALL=(ALL:ALL) ALL. Review the installed policy and group membership together.

Lab: Create, Verify, and Revoke

- In a disposable VM, create alice with a home and a dedicated group.
- Compare id output with the account and group records.
- Add one supplementary group and verify it in a new login session.
- Grant only the example id -u operation and test it with sudo -l.
- Lock the password and explain which access paths still need review.
- Remove the account after verifying a backup and remaining ownership.

Deliverable

Record commands and observations. Include no passwords, private keys, or real password hashes.

Debian User Management Cheat Sheet

Task	Debian command
Create account / group	<code>adduser <user> / addgroup <group></code>
Append supplementary group	<code>adduser <user> <group> or usermod -aG</code>
Set / lock password	<code>passwd <user> / passwd -l <user></code>
Inspect aging	<code>chage -l <user></code>
Remove local account	<code>deluser <user></code>
Validate sudo policy	<code>visudo -c; sudo -l</code>

Final check

Confirm identity, ownership, a new login, and effective permissions after every change.

References

- Debian Reference: authentication and user administration
- Debian manuals: adduser(8), addgroup(8), deluser(8)
- Debian manuals: passwd(5), shadow(5), usermod(8), chage(1)
- Debian manuals: nsswitch.conf(5), pam(7), limits.conf(5)
- Privilege policy: sudoers(5), visudo(8), sshd_config(5)

Platform conventions

Debian local-account examples. FreeBSD is a comparison; consult the actual installed package manuals.