sudo Run as another user

CONCEPTS

sudo runs one command with another identity (root by default) if the sudoers rules allow it, and asks for your own password, not root's.

Redirections happen in your shell before sudo starts, so sudo echo x > /etc/f still writes as you.

RUNNING COMMANDS

`sudo <command>`
Run a command as root

`sudo -u <user> <command>`
Run as another user

`sudo !!`
Repeat the last command with sudo

SHELLS

`sudo -i`
Root login shell with root's environment

`sudo -s`
Root shell keeping your environment

`sudo -iu <user>`
Login shell as another user

PRIVILEGES AND PASSWORD CACHE

`sudo -l`
List what you may run here

`sudo -v`
Refresh the cache without running anything

`sudo -k`
Invalidate the cache for this session

ENVIRONMENT

`sudo -E <command>`
Keep your environment variables

`sudo <VAR>=<value> <command>`
Set a variable for one command

ENVIRONMENT (cont.)

PATH is replaced by secure_path on many systems, so tools in your own PATH may not be found under sudo.

EDITING FILES

`sudo -e <file>`
Edit a copy and write it back as root

REDIRECTION AND PIPES

`echo '<text>' | sudo tee <file>`
Write a root-owned file

`echo '<text>' | sudo tee -a <file>`
Append to a root-owned file

`sudo sh -c '<command> > <file>'`
Redirect inside a root shell

SUDOERS RULES

`sudo visudo`
Edit /etc/sudoers with a syntax check

`sudo visudo -f /etc/sudoers.d/<name>`
Edit a drop-in file

`sudo visudo -c`
Check the syntax only

Rule: who where = (as whom) what

Part	Example	Meaning
`user or %group`	`alice, %admin`	who may run it
`host`	`ALL`	on which machine
`(runas)`	`(root), (ALL)`	as which user
`tags: commands`	`NOPASSWD: /usr/bin/apt`	what, and options

`%wheel ALL=(ALL) ALL`
Full rights for a group

Never edit /etc/sudoers with a plain editor: one syntax error can lock out every sudo user. Keep a root shell open while testing.