Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

751

group: read and run

others: run, but not read

2.2

ACL: stored with each object, listing users and their rights (a column of the grid)

capability: held by each user, a token naming an object and rights (a row)

2.3

device access is privileged: it works only in kernel mode

so the kernel can check permissions and stop one program from corrupting another's files

2.4

the copy runs past `buf` and overwrites the saved return address, so an attacker can choose where the function returns

a stack canary detects the overwrite

a non-executable stack or ASLR makes the attack much harder; or use a bounded copy

3.1

(a) only root may write the password file; setuid makes `passwd` run as root whoever starts it

(b) any bug in it runs as root, so an attacker who exploits it controls the whole machine

(c) a small helper or service that does only one thing, changing one user's own entry, and checks the old password; the rest runs as the user

(d) setuid, with owner `rwx`, group `r-x`, others `r-x`

3.2

(a) $10^8 \times 1\ \mu s = $ **100 s**

(b) $10^8 / 65\,536 \approx$ **1526** calls; about **1.5 ms**

(c) it switches to kernel mode and back, and the kernel saves registers and checks every argument

3.3

(a) any student's code can read and change every other site's files, e.g. steal database passwords or deface pages

exercises.lol exercises.com.cn

(b) each site's code runs as its student's own user; folders `750`, so other students cannot read them

(c) each site gets its own view of the files and its own process list, with limits on CPU and memory so one site cannot slow the rest

(d) one account per student, inside a container with resource limits: least privilege isolates students from each other, and limits stop one buggy site from taking down 299 others; the cost is a little more set-up