Lesson 3.5: Fixes

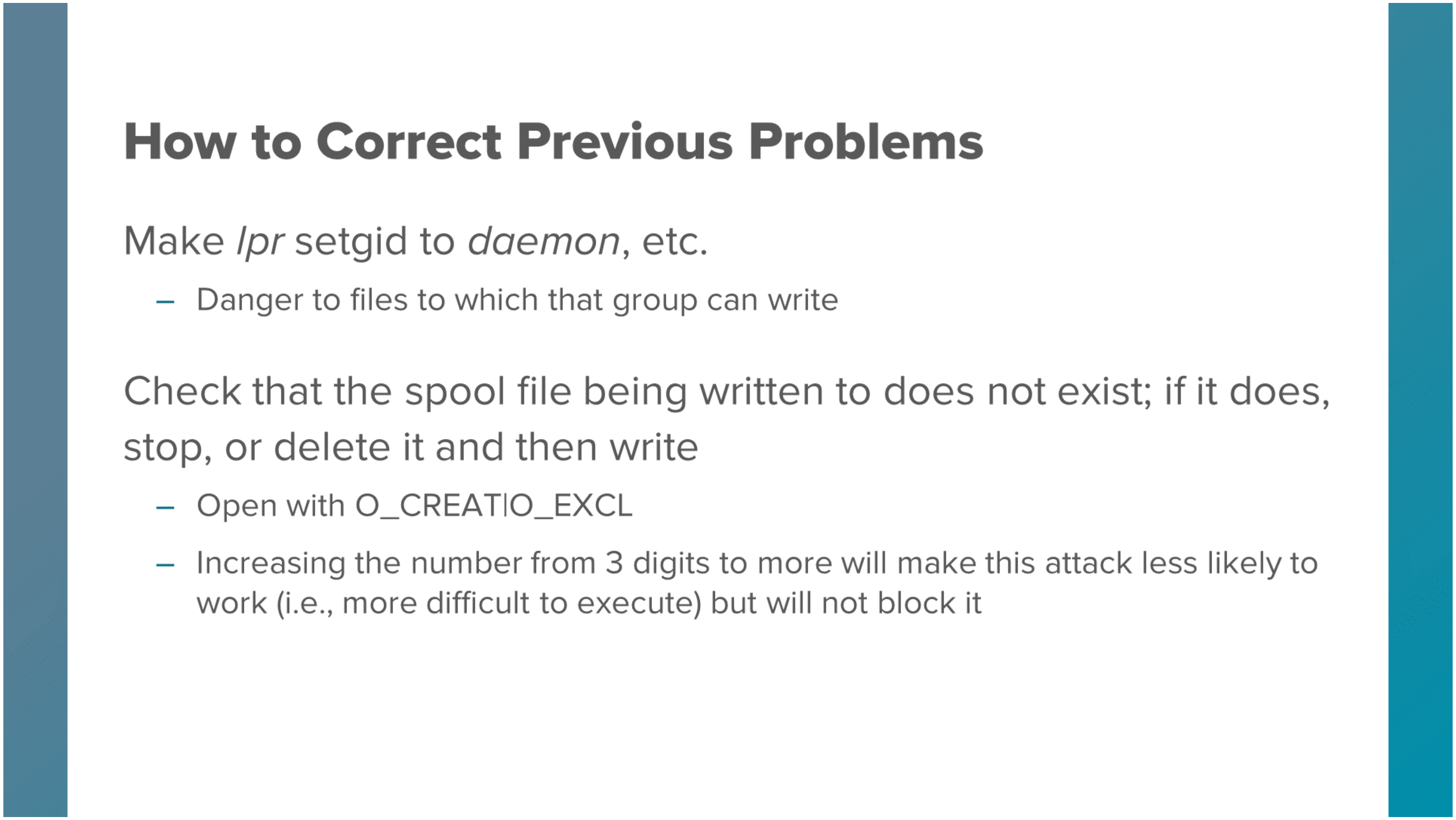
SECURITY VULNERABILITIES IN C/C++ PROGRAMMING

Fixes

Matt Bishop, Ph.D.
Professor of Computer Science,
UC Davis

UCDAVIS
Continuing and Professional Education

Slide 1: How to Correct Previous Problems



How to Correct Previous Problems

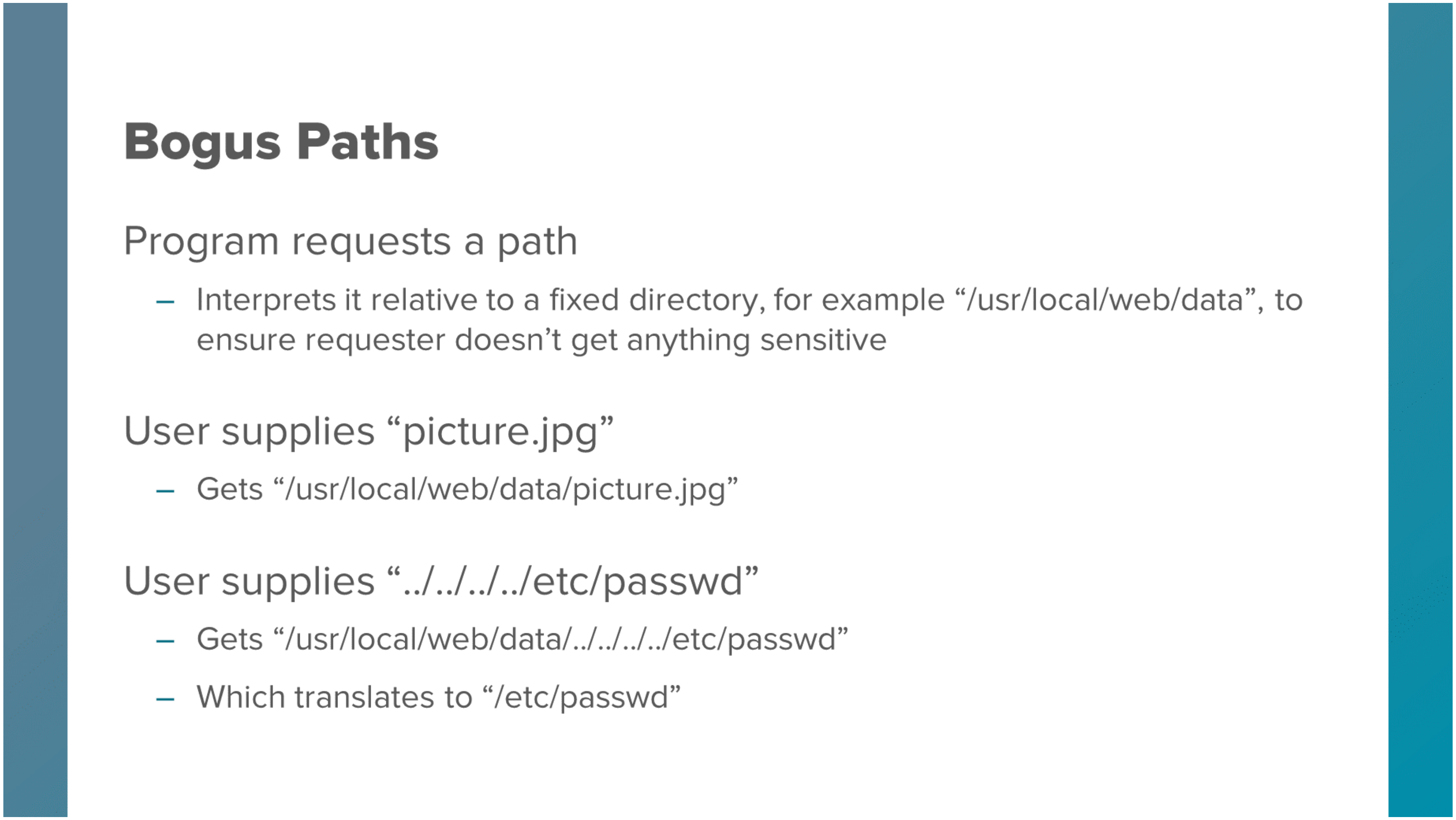
Make *lpr* setgid to *daemon*, etc.

- Danger to files to which that group can write

Check that the spool file being written to does not exist; if it does, stop, or delete it and then write

- Open with O_CREAT|O_EXCL
- Increasing the number from 3 digits to more will make this attack less likely to work (i.e., more difficult to execute) but will not block it

Slide 2: Bogus Paths



Bogus Paths

Program requests a path

- Interprets it relative to a fixed directory, for example “/usr/local/web/data”, to ensure requester doesn’t get anything sensitive

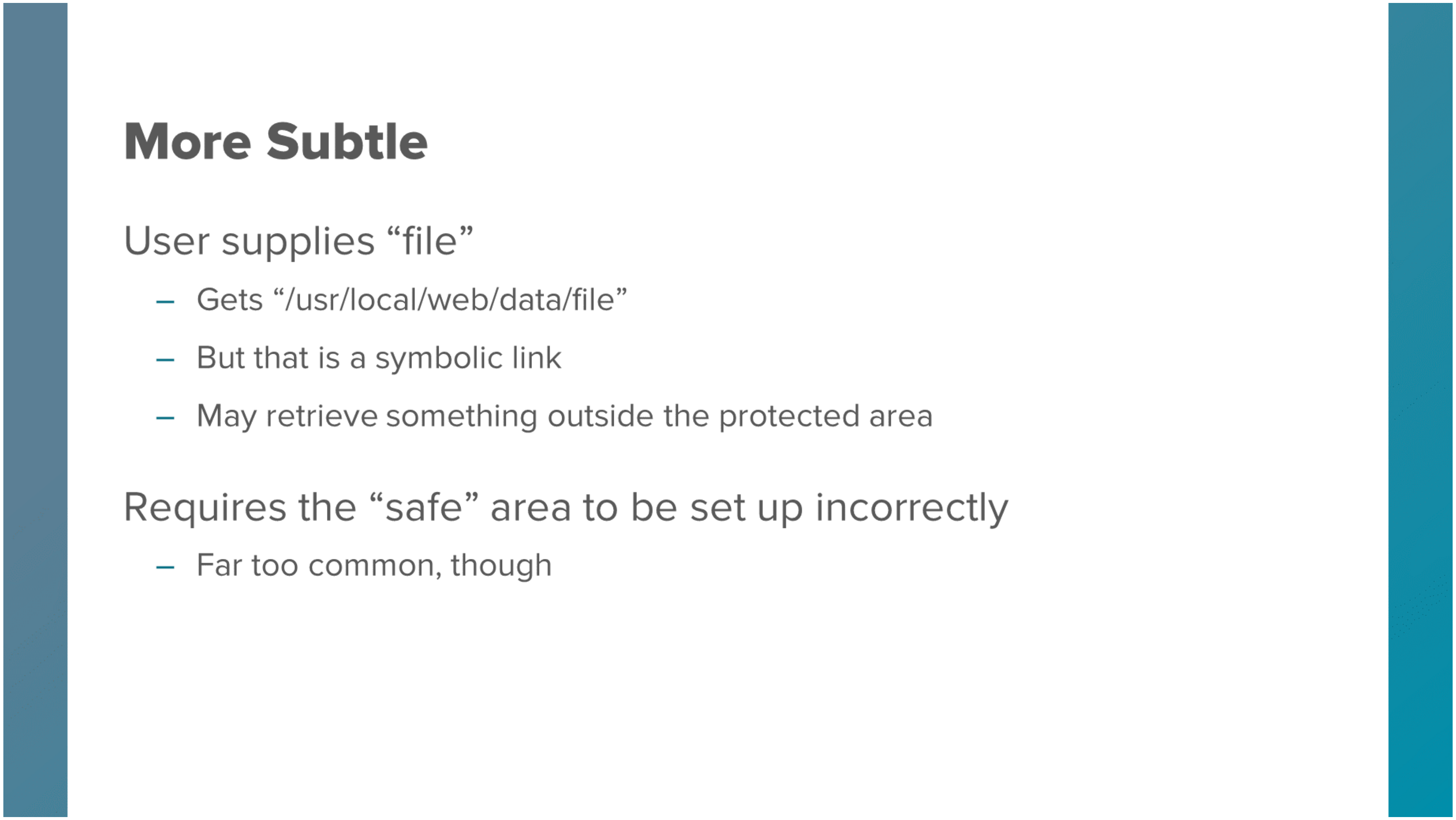
User supplies “picture.jpg”

- Gets “/usr/local/web/data/picture.jpg”

User supplies “../../../../etc/passwd”

- Gets “/usr/local/web/data/../../../../etc/passwd”
- Which translates to “/etc/passwd”

Slide 3: More Subtle



More Subtle

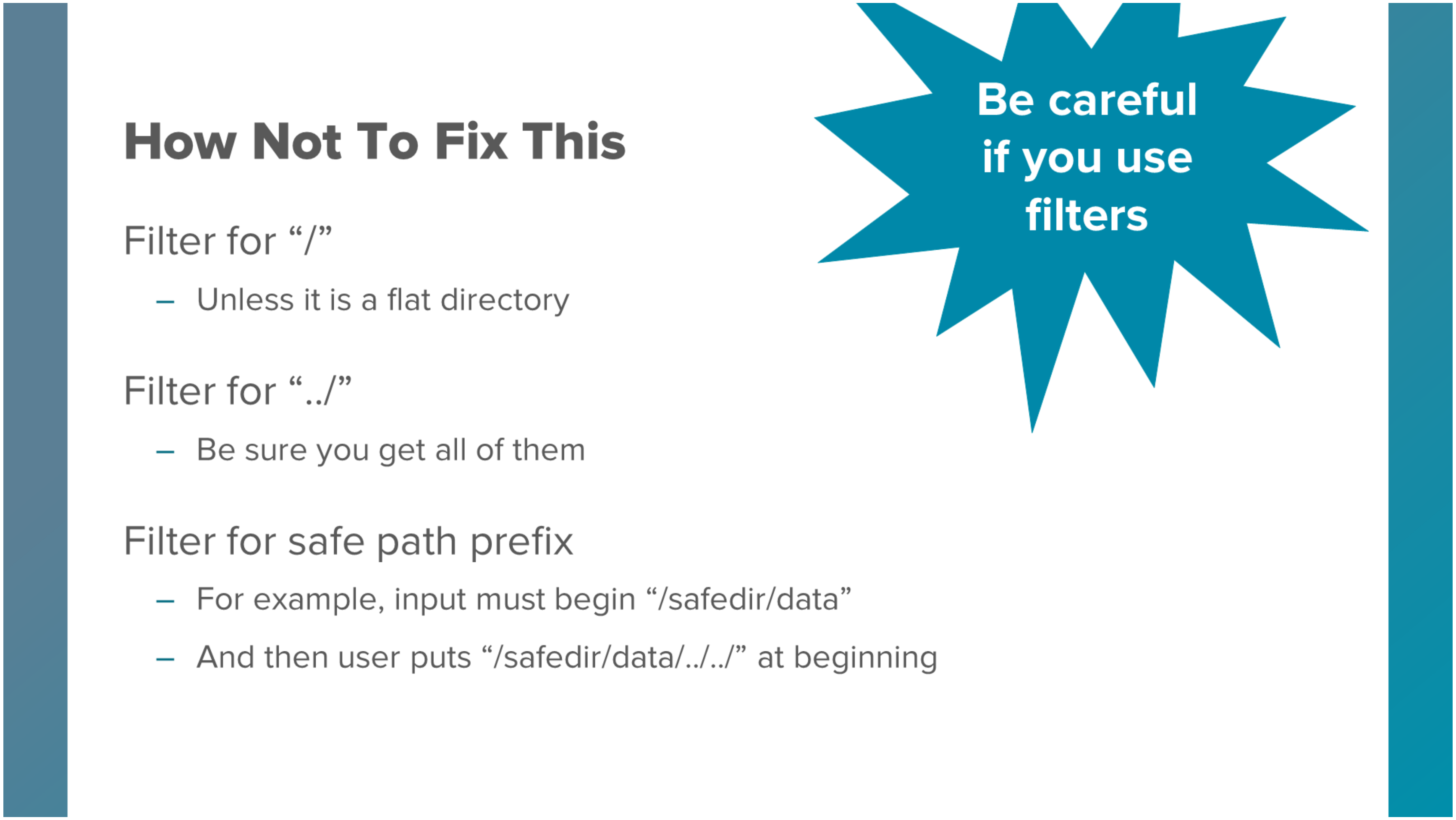
User supplies “file”

- Gets “/usr/local/web/data/file”
- But that is a symbolic link
- May retrieve something outside the protected area

Requires the “safe” area to be set up incorrectly

- Far too common, though

Slide 4: How Not To Fix This



How Not To Fix This

Be careful if you use filters

Filter for “/”

- Unless it is a flat directory

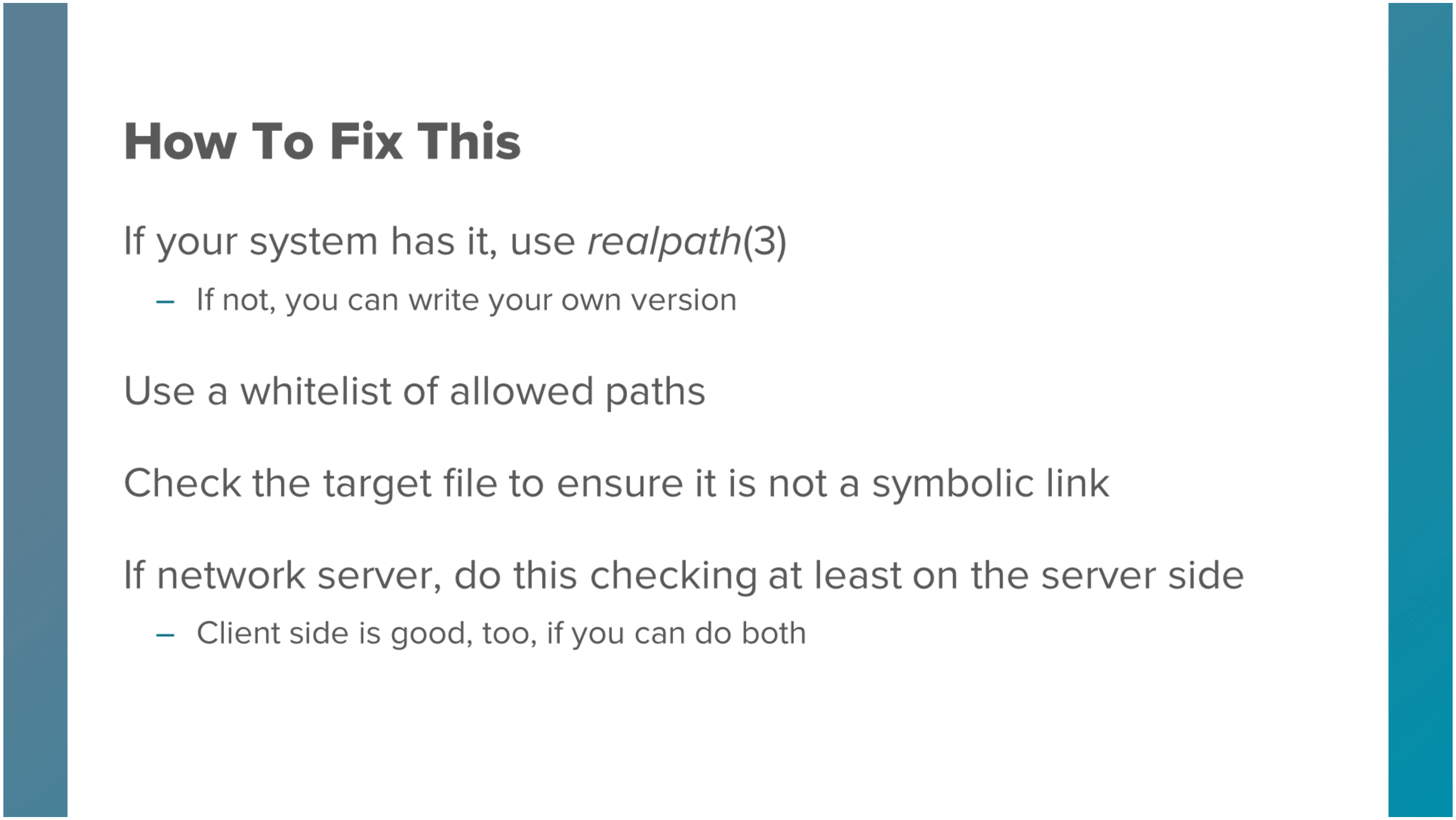
Filter for “../”

- Be sure you get all of them

Filter for safe path prefix

- For example, input must begin “/safedir/data”
- And then user puts “/safedir/data/../../” at beginning

Slide 5: How To Fix This



How To Fix This

If your system has it, use *realpath*(3)

- If not, you can write your own version

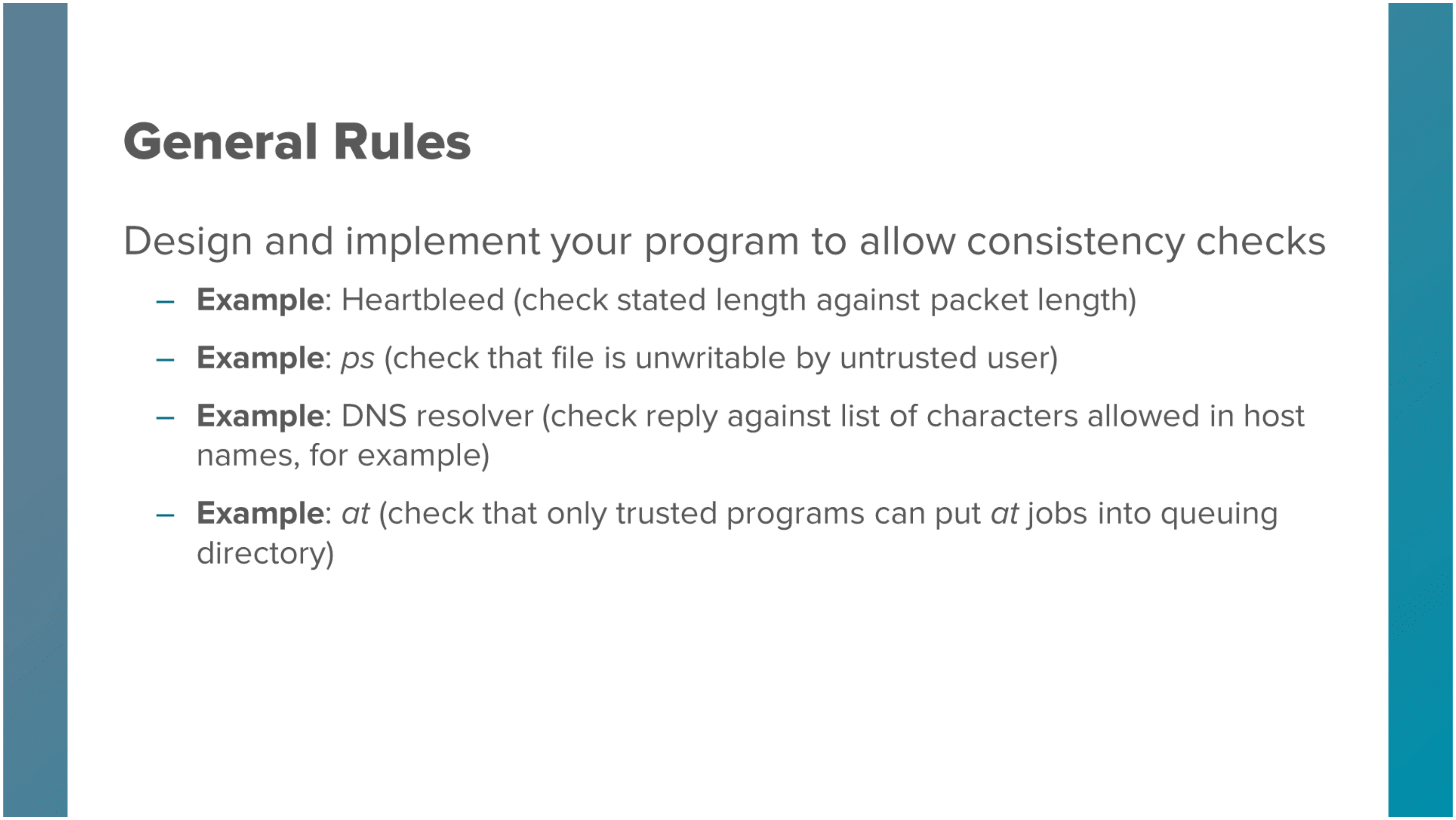
Use a whitelist of allowed paths

Check the target file to ensure it is not a symbolic link

If network server, do this checking at least on the server side

- Client side is good, too, if you can do both

Slide 6: General Rules



General Rules

Design and implement your program to allow consistency checks

- **Example**: Heartbleed (check stated length against packet length)
- **Example**: *ps* (check that file is unwritable by untrusted user)
- **Example**: DNS resolver (check reply against list of characters allowed in host names, for example)
- **Example**: *at* (check that only trusted programs can put *at* jobs into queuing directory)