Lesson 7.6: Race Conditions

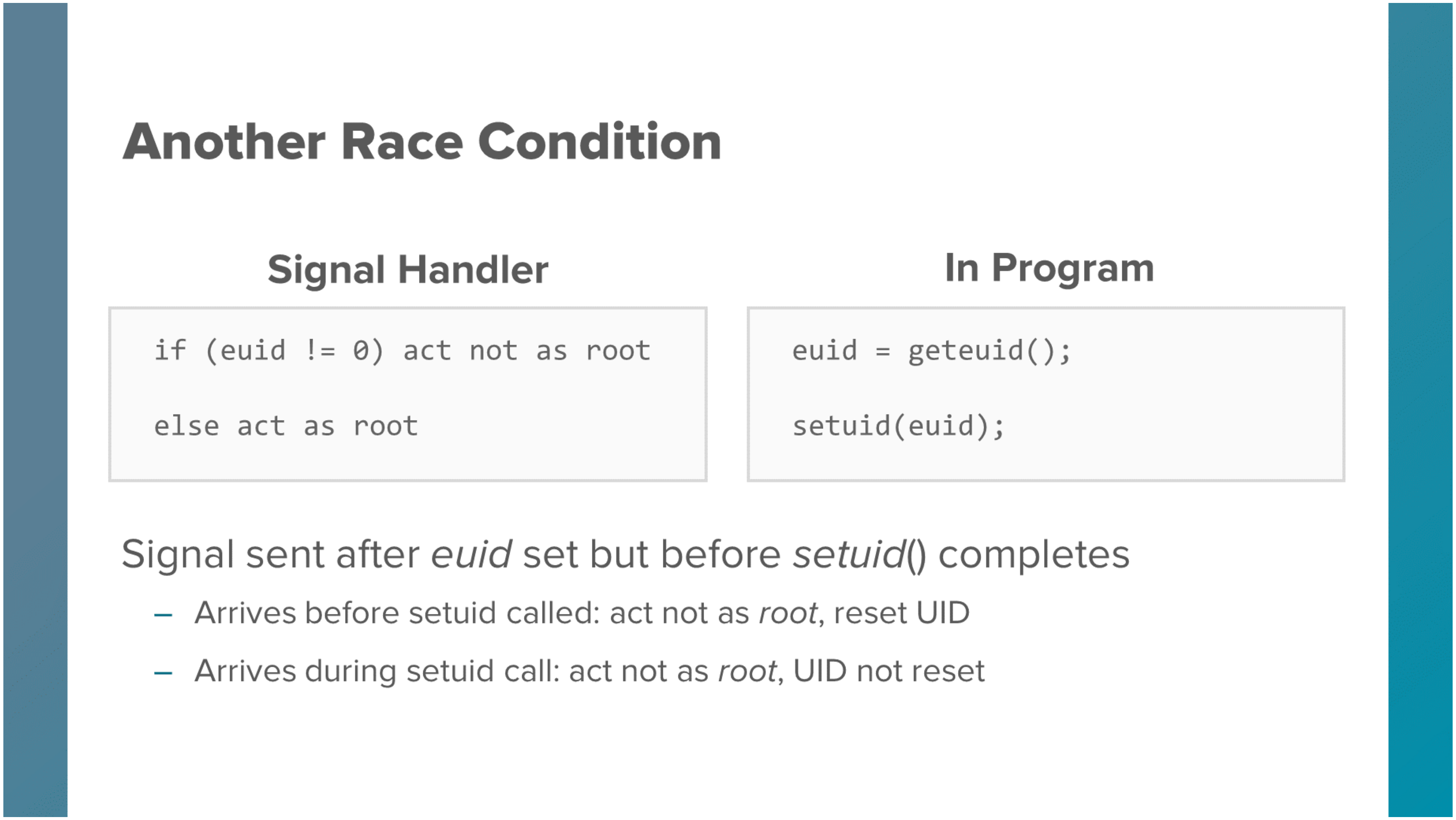
SECURITY VULNERABILITIES IN C/C++ PROGRAMMING

Race Conditions

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

UCDAVIS
Continuing and Professional Education

Slide 1: Another Race Condition



Another Race Condition

Signal Handler

```
if (euid != 0) act not as root

else act as root
```

In Program

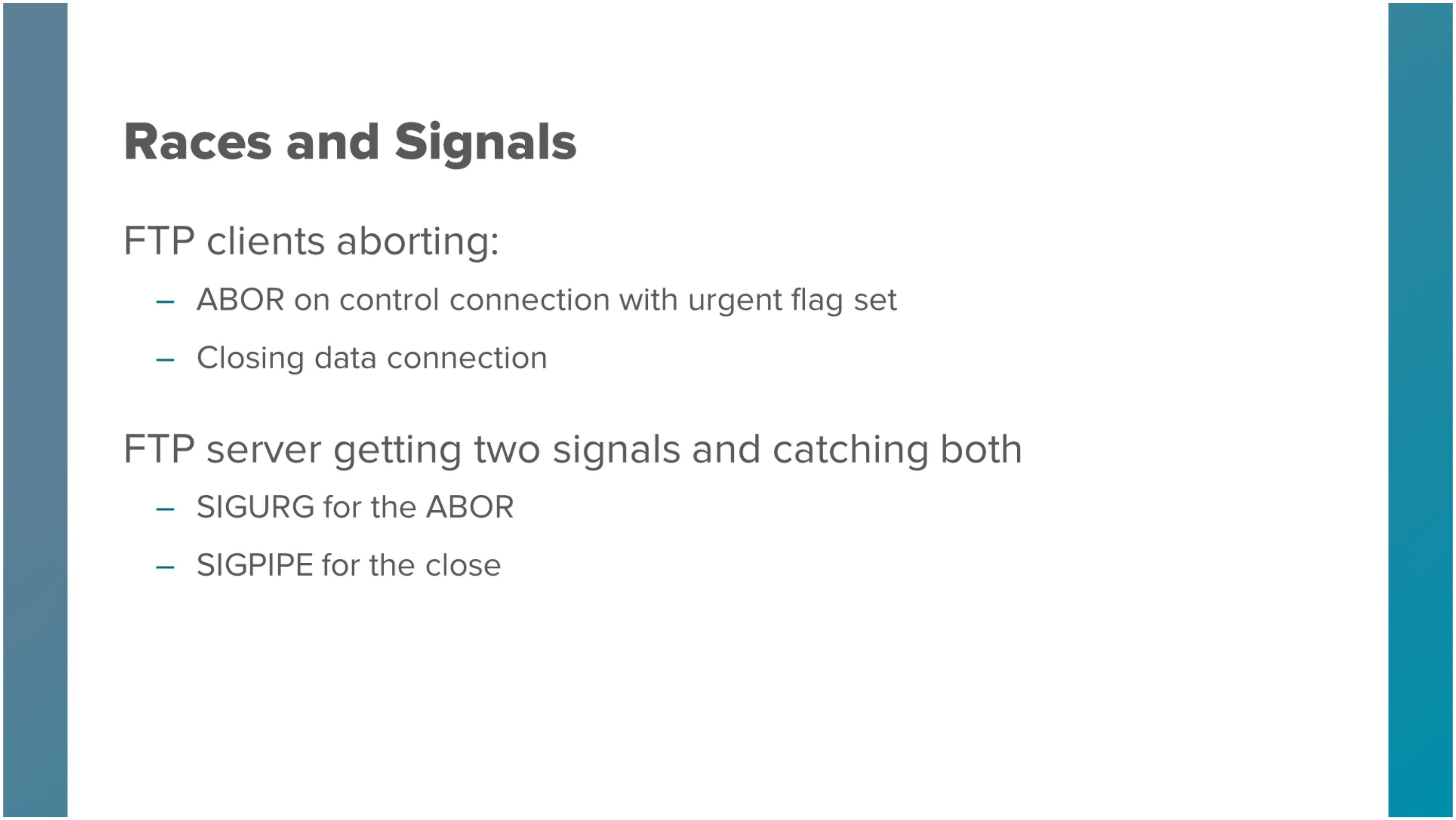
```
euid = geteuid();

setuid(euid);
```

Signal sent after *euid* set but before *setuid*() completes

- Arrives before setuid called: act not as *root*, reset UID
- Arrives during setuid call: act not as *root*, UID not reset

Slide 2: Races and Signals



Races and Signals

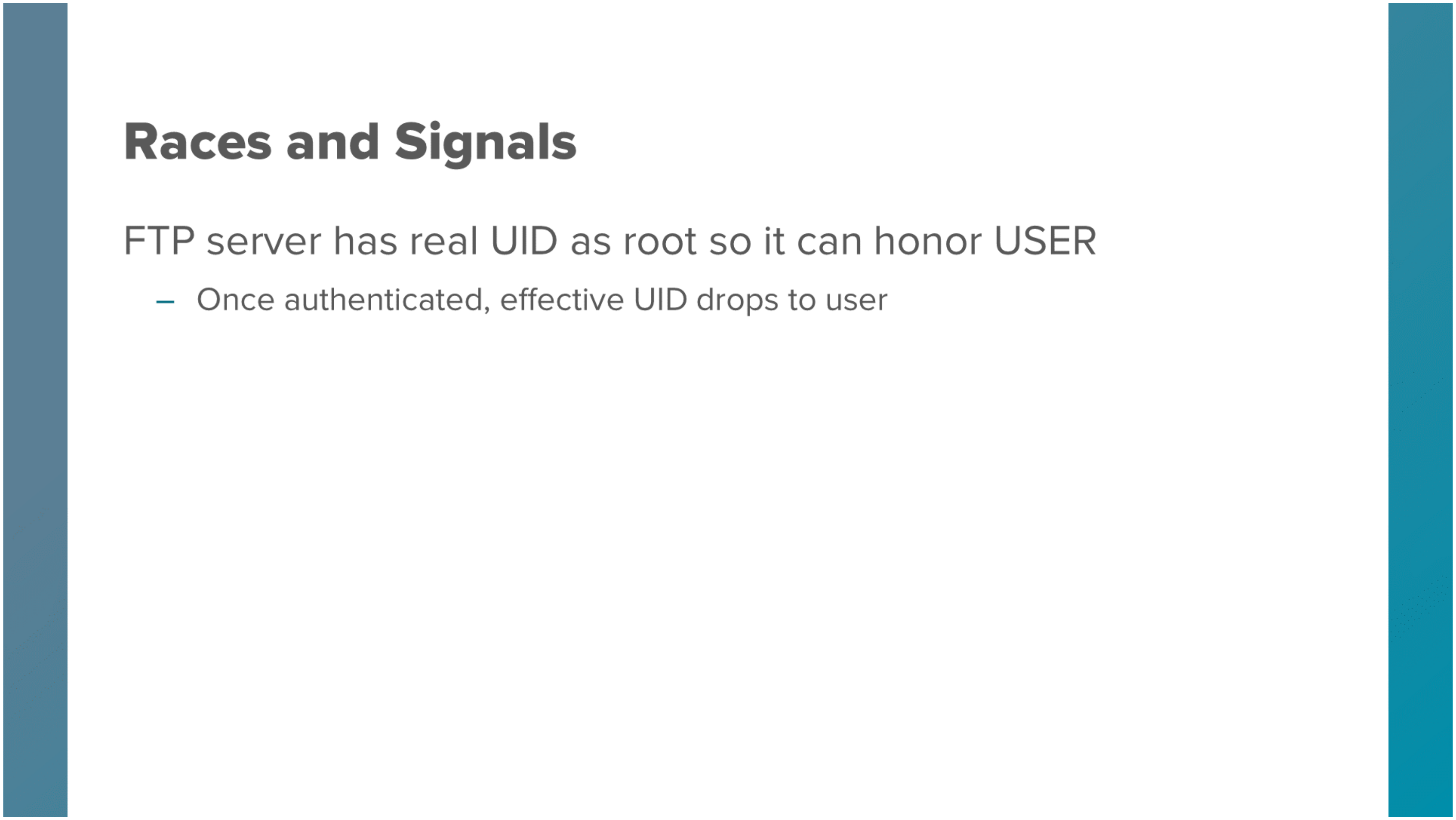
FTP clients aborting:

- ABOR on control connection with urgent flag set
- Closing data connection

FTP server getting two signals and catching both

- SIGURG for the ABOR
- SIGPIPE for the close

Slide 3: Races and Signals

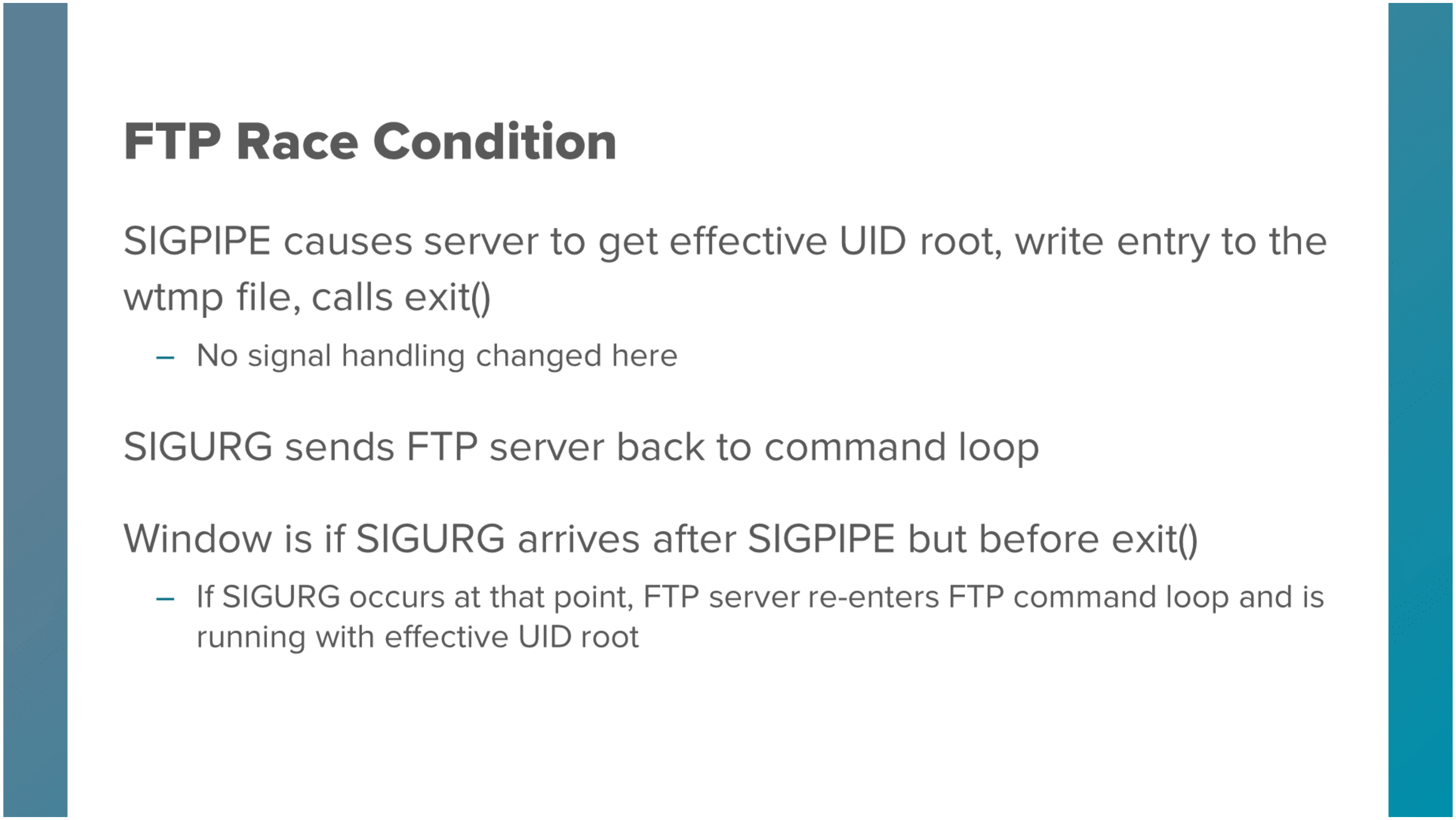


Races and Signals

FTP server has real UID as root so it can honor USER

- Once authenticated, effective UID drops to user

Slide 4: FTP Race Condition



FTP Race Condition

SIGPIPE causes server to get effective UID root, write entry to the wtmp file, calls exit()

- No signal handling changed here

SIGURG sends FTP server back to command loop

Window is if SIGURG arrives after SIGPIPE but before exit()

- If SIGURG occurs at that point, FTP server re-enters FTP command loop and is running with effective UID root