Lesson 6.3: Subprocess Environment

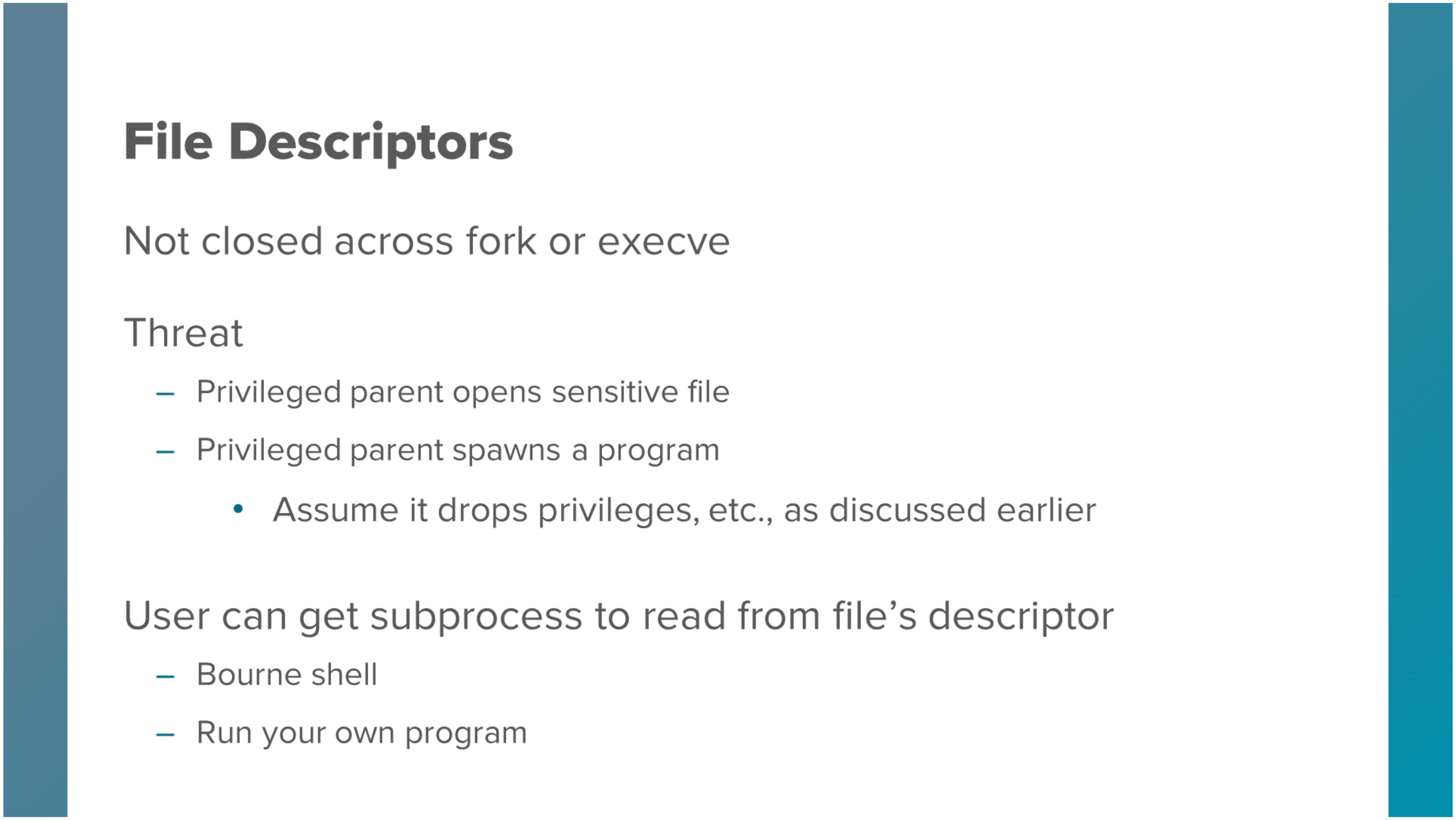
SECURITY VULNERABILITIES IN C/C++ PROGRAMMING

Subprocess Environment

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

UCDAVIS
Continuing and Professional Education

Slide 1: File Descriptors



File Descriptors

Not closed across fork or execve

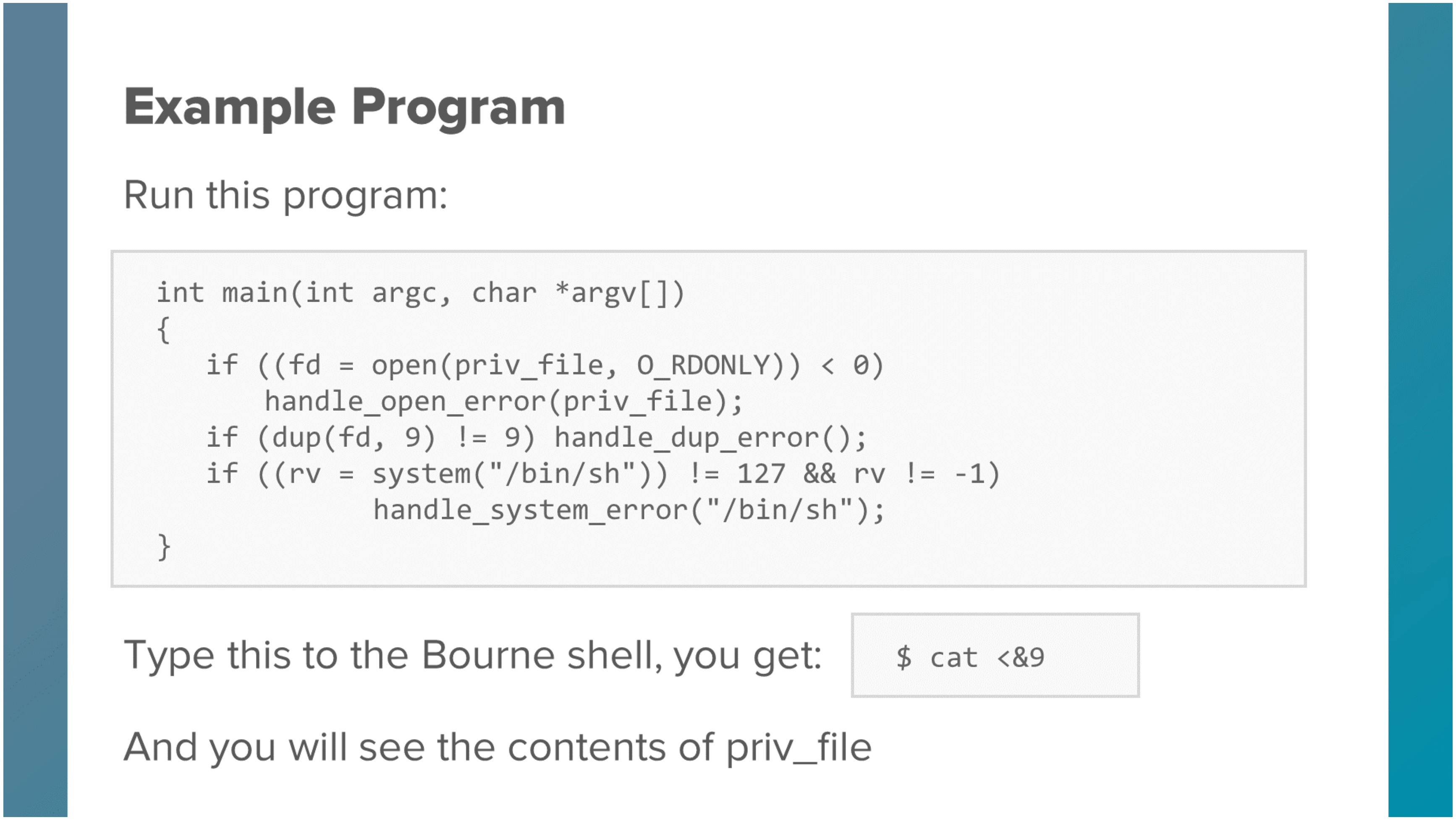
Threat

- Privileged parent opens sensitive file
- Privileged parent spawns a program
    - Assume it drops privileges, etc., as discussed earlier

User can get subprocess to read from file’s descriptor

- Bourne shell
- Run your own program

Slide 2: Example Program



Example Program

Run this program:

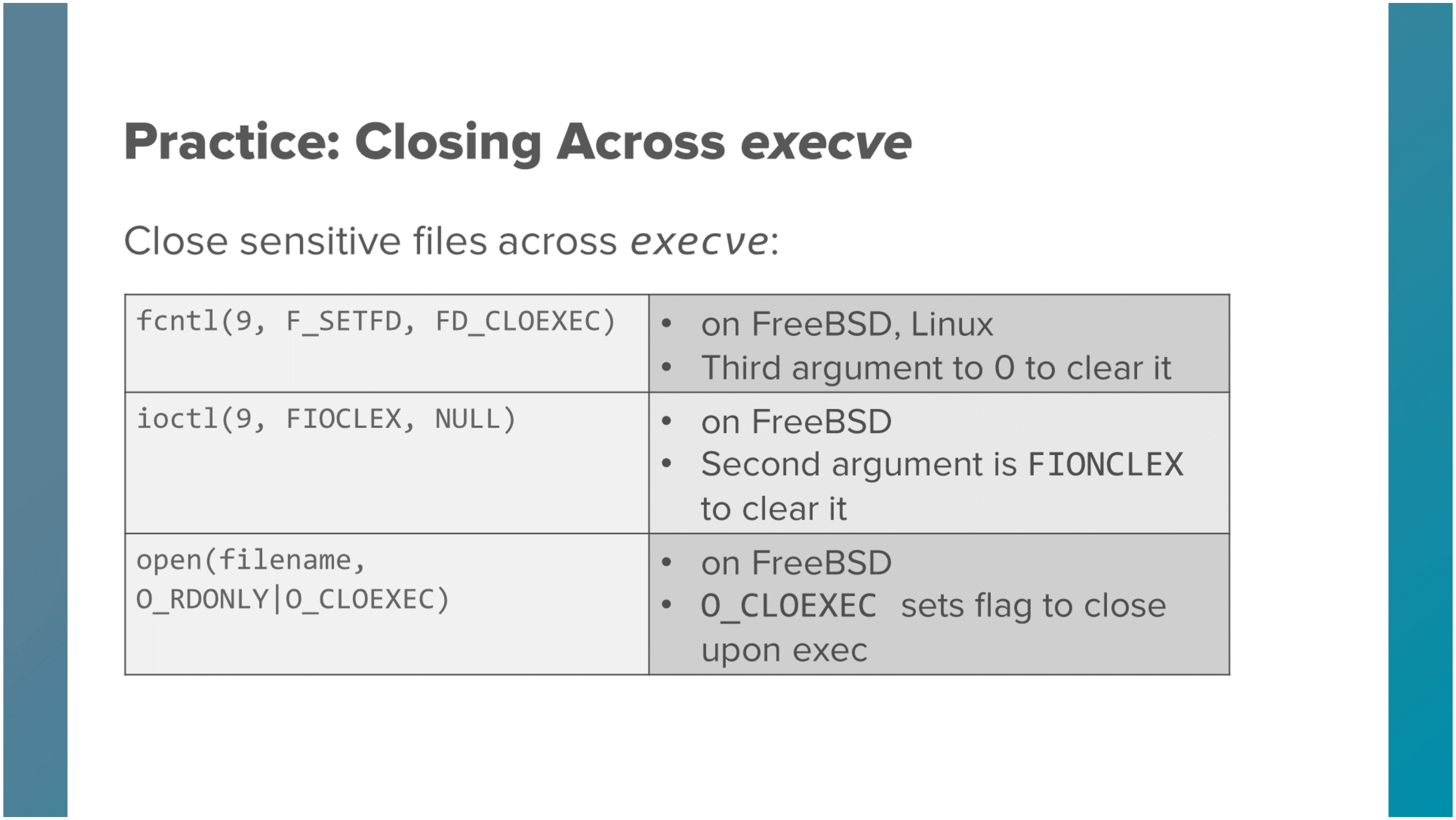
```
int main(int argc, char *argv[])
{
   if ((fd = open(priv_file, O_RDONLY)) < 0)
       handle_open_error(priv_file);
   if (dup(fd, 9) != 9) handle_dup_error();
   if ((rv = system("/bin/sh")) != 127 && rv != -1)
             handle_system_error("/bin/sh");
}
```

Type this to the Bourne shell, you get:

```
$ cat <&9
```

And you will see the contents of priv_file

Slide 3: Practice: Closing Across execve

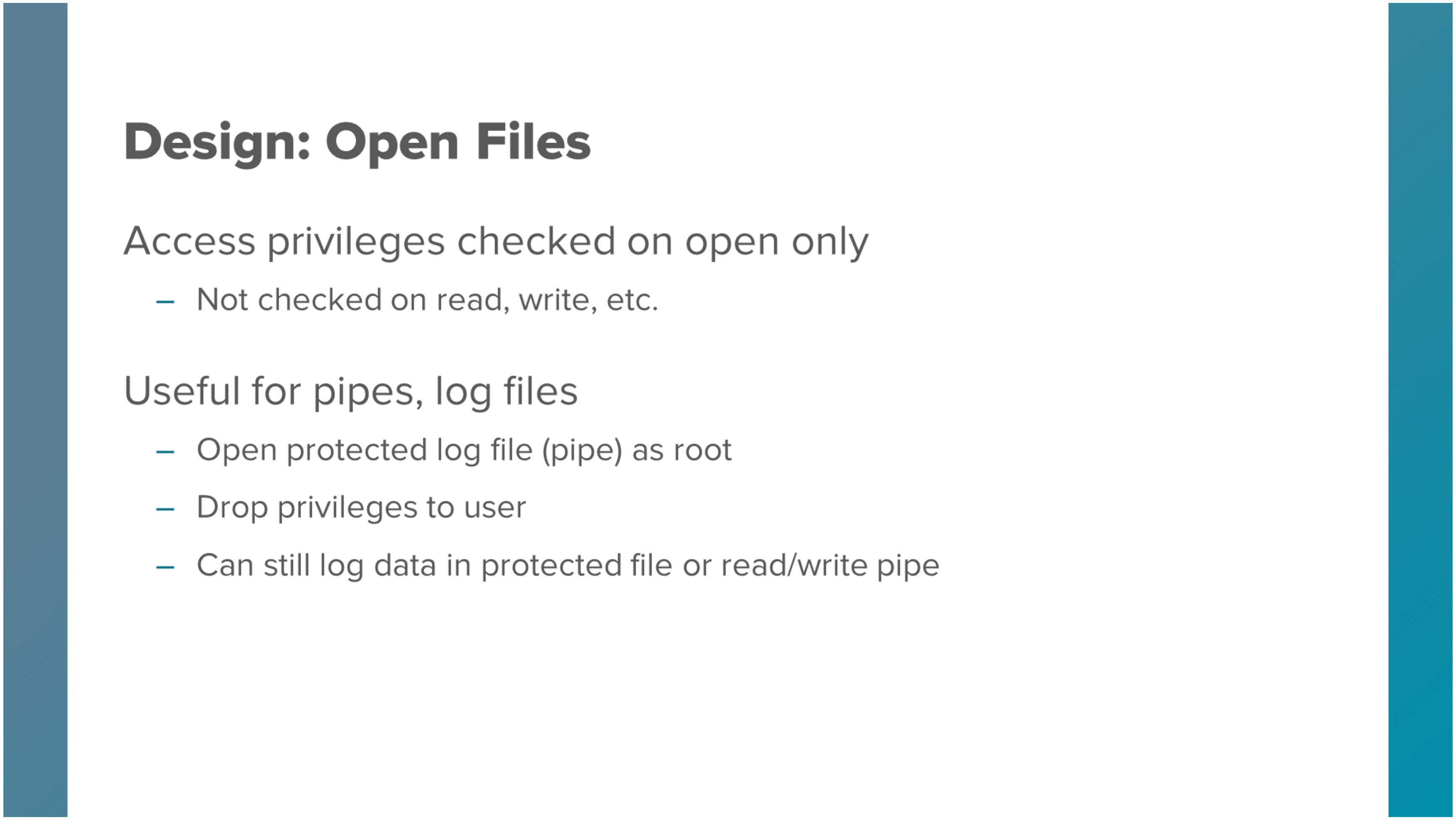


Practice: Closing Across *execve*

Close sensitive files across *execve*:

`fcntl(9, F_SETFD, FD_CLOEXEC)`	• on FreeBSD, Linux • Third argument to 0 to clear it
`ioctl(9, FIOCLEX, NULL)`	• on FreeBSD • Second argument is `FIONCLEX` to clear it
`open(filename, O_RDONLY|O_CLOEXEC)`	• on FreeBSD • `O_CLOEXEC` sets flag to close upon exec

Slide 4: Design: Open Files



Design: Open Files

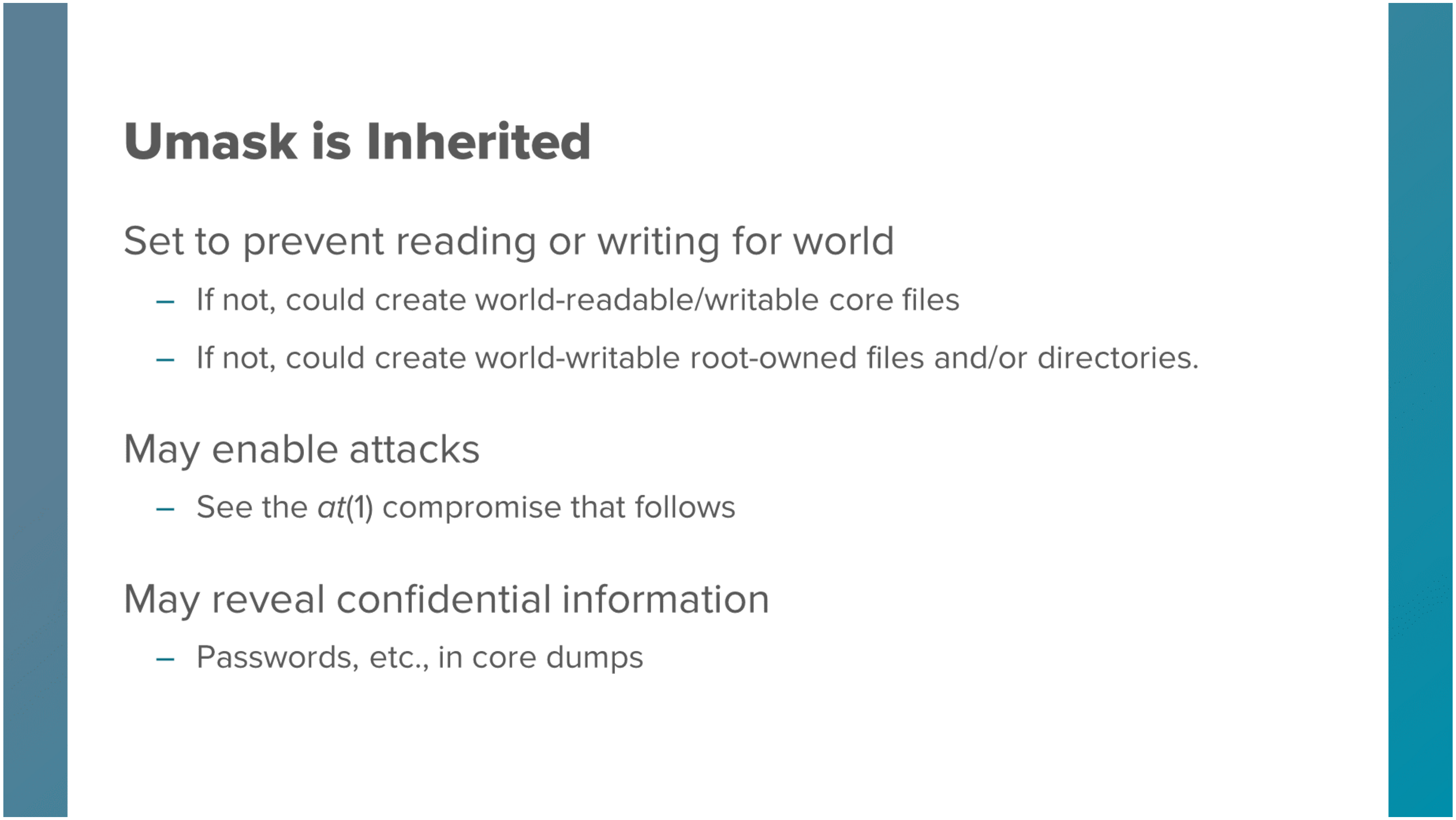
Access privileges checked on open only

- Not checked on read, write, etc.

Useful for pipes, log files

- Open protected log file (pipe) as root
- Drop privileges to user
- Can still log data in protected file or read/write pipe

Slide 5: Umask is Inherited



Umask is Inherited

Set to prevent reading or writing for world

- If not, could create world-readable/writable core files
- If not, could create world-writable root-owned files and/or directories.

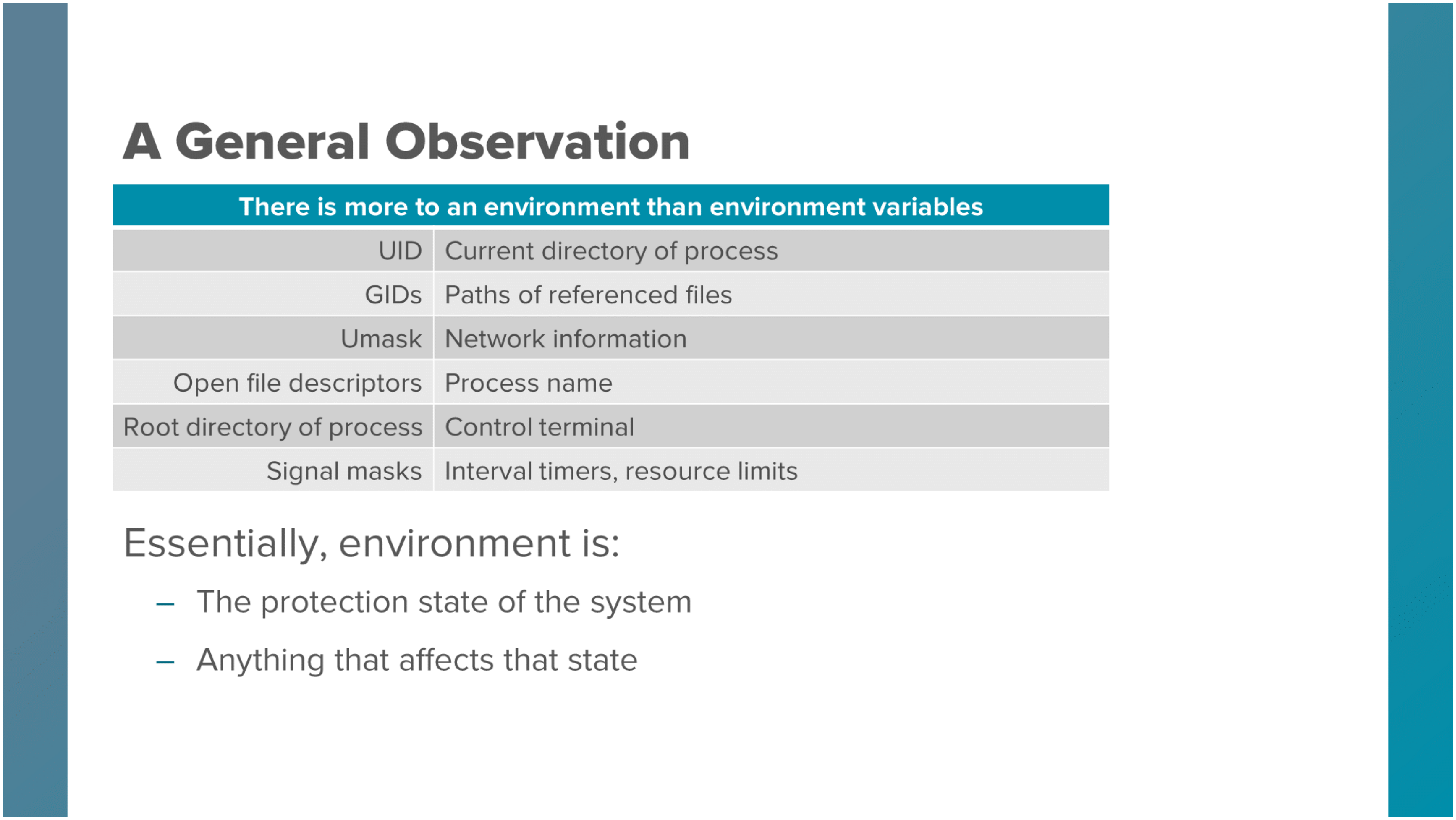
May enable attacks

- See the *at*(1) compromise that follows

May reveal confidential information

- Passwords, etc., in core dumps

Slide 6: A General Observation



A General Observation

There is more to an environment than environment variables	
UID	Current directory of process
GIDs	Paths of referenced files
Umask	Network information
Open file descriptors	Process name
Root directory of process	Control terminal
Signal masks	Interval timers, resource limits

Essentially, environment is:

- The protection state of the system
- Anything that affects that state