Systems & Software Security

COMSM0050
2020/2021

University of BRISTOL

Race condition: Examples
Dirty COW

bristol.ac.uk

Example 3: dirty cow

- Dirty Copy on Write
- CVE-2016-5195 (fixed October 2016)
- Linux vulnerability that could be exploited to gain root access
- Concurrency issue relating to how memory is managed

Example 3: dirty cow

- Local privilege escalation
- Allows an attacker to write to a file that is read only
- Can be used to gain a foot in a machine
    - e.g. by modifying /bin/bash
    - e.g. libraries
    - e.g. system configurations
    - e.g. overwriting /etc/passwd
    - etc…

Memory Mapped Files

- a.k.a mmap
    - Link to the man page on the course website
- maps files or devices to virtual memory of a process
- If changes are made and the mmaping is SHARED (e.g. MAP_SHARED flag) change to memory is copied to file
    - Minus some synchronization
    - Can be forced with msync
- More details in the man page.

mmap code example

```
int fd;
size_t size;
const char* mmaped;
fd = open (“file.txt", O_RDONLY);
// get file size e.g. with stat
mapped = mmap (0, size, PROT_READ, MAP_PRIVATE, fd, 0);
mmaped[10]; // content of the file at address 10
```

mmap code example

```
int fd;
size_t size;
fd = open (“file.txt", O_RDONLY);
// get file size e.g. with stat
mapped = mmap (0, size, PROT_READ, MAP_PRIVATE, fd, 0);
mmaped[10]; // content of the file at address 10
```

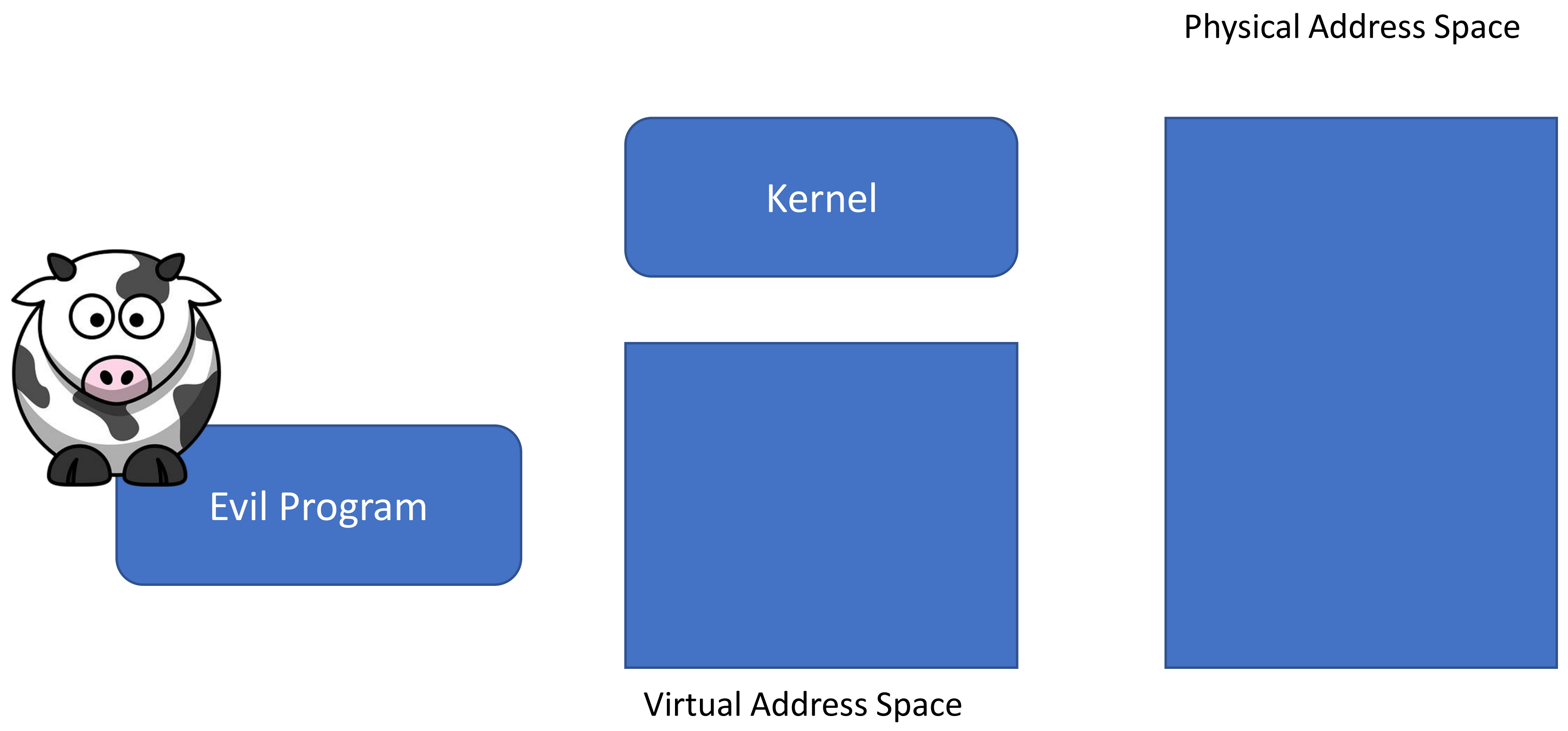
Exercise: try this at home!

Memory Mapped Files

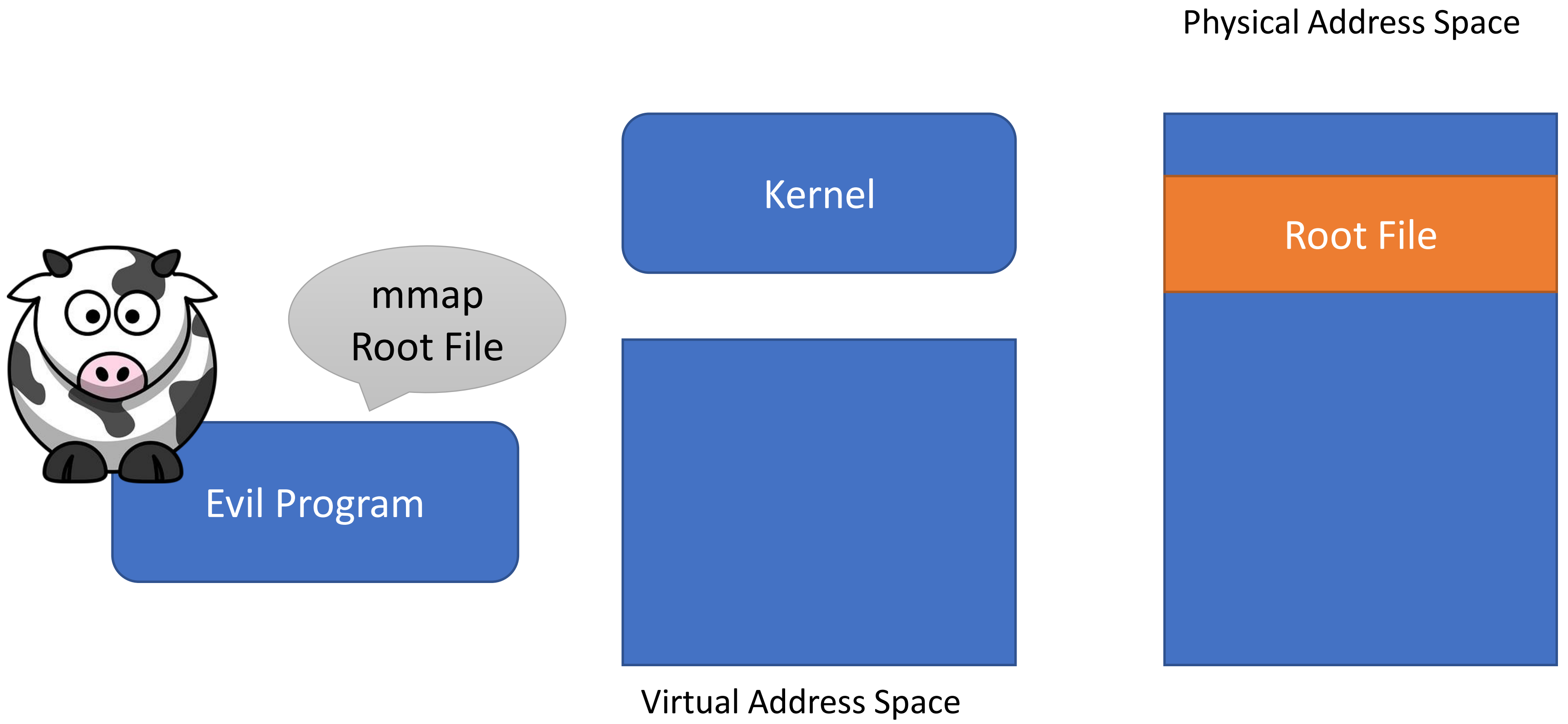
- File can be mmaped as **private**
- Change to privately mmaped file do not affect the underlying file
- Private mmaped file can be read/write regardless of access to the underlying file

- … but sometimes you could change the underlying file!
  - dirty COW!

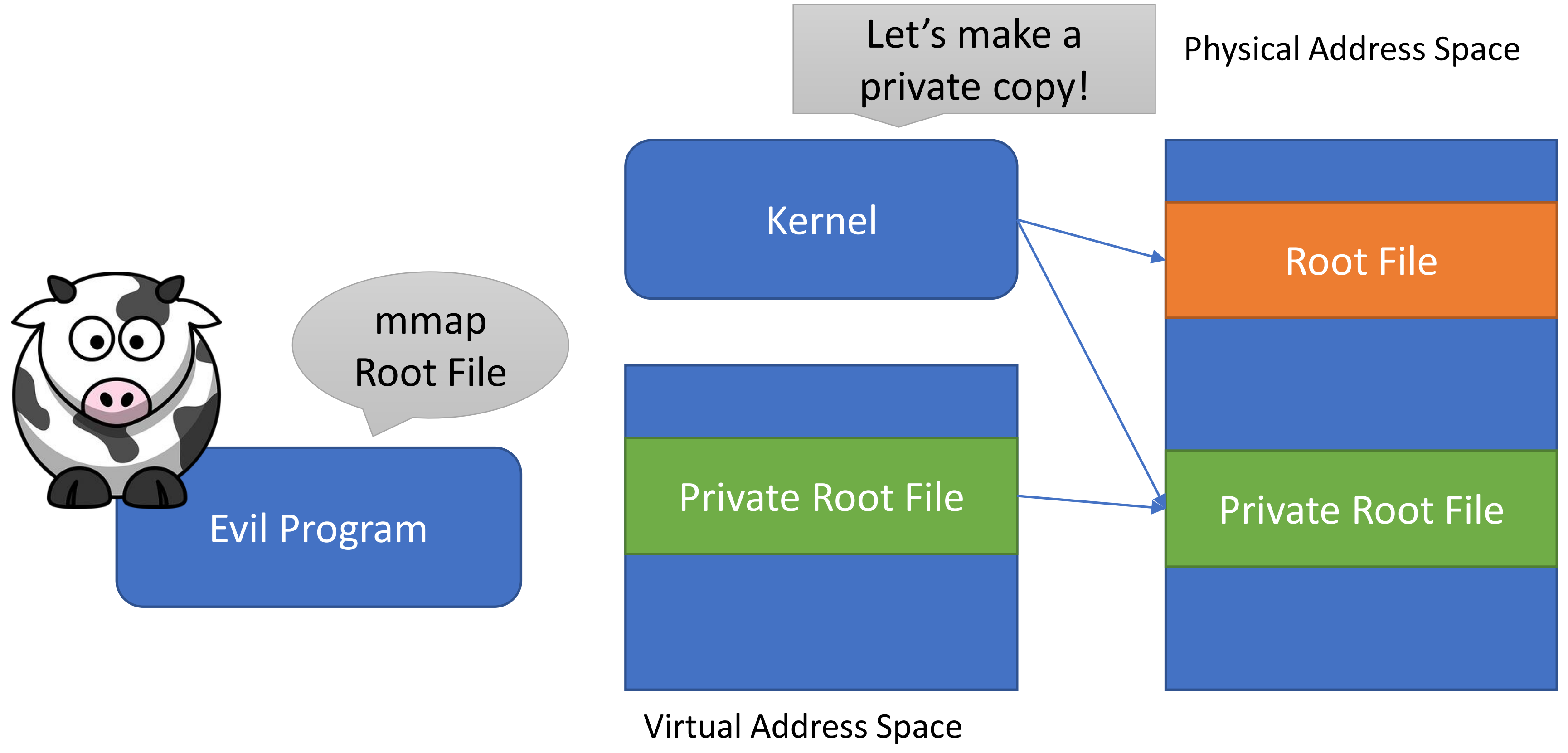
Dirty COW illustrated



Dirty COW illustrated

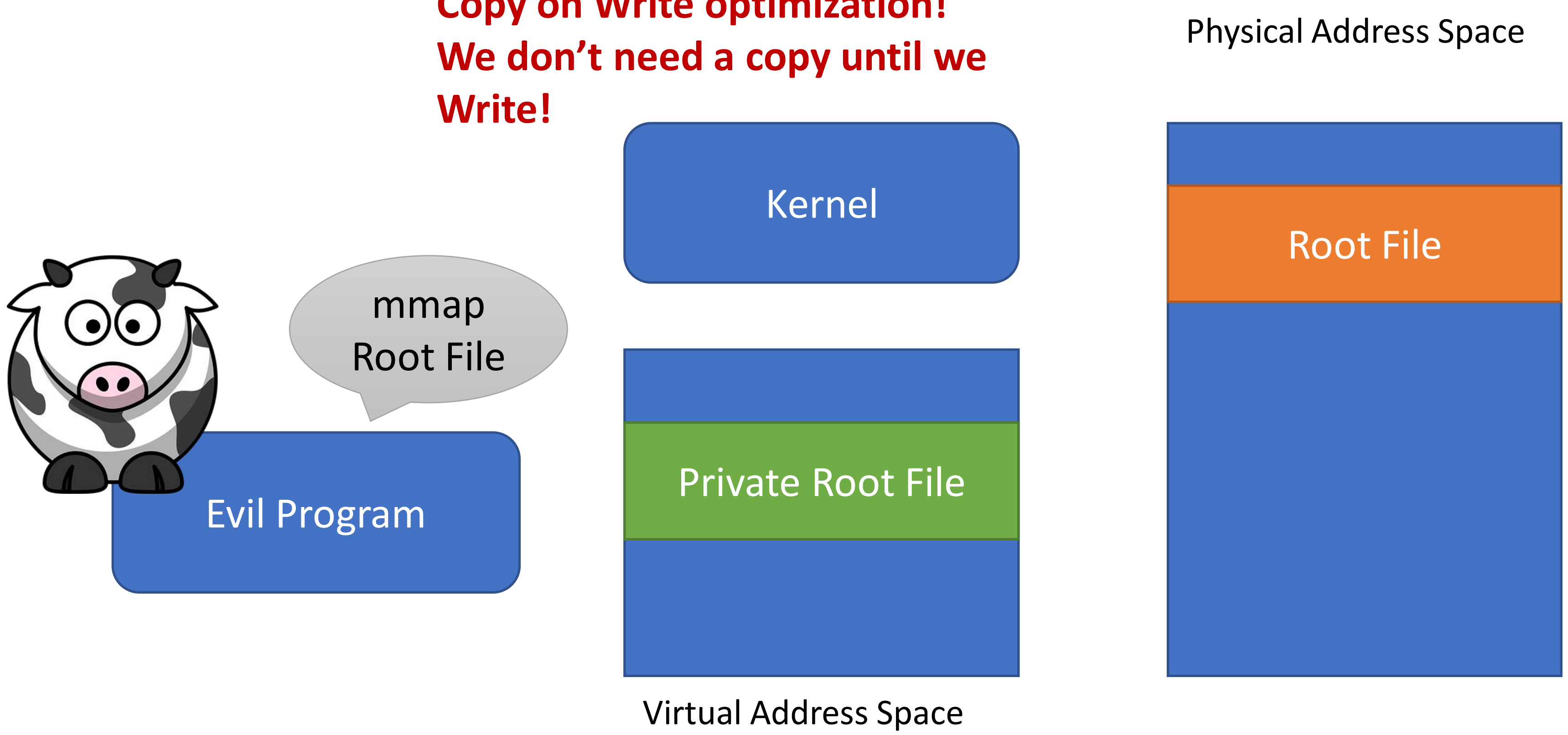


Dirty COW illustrated

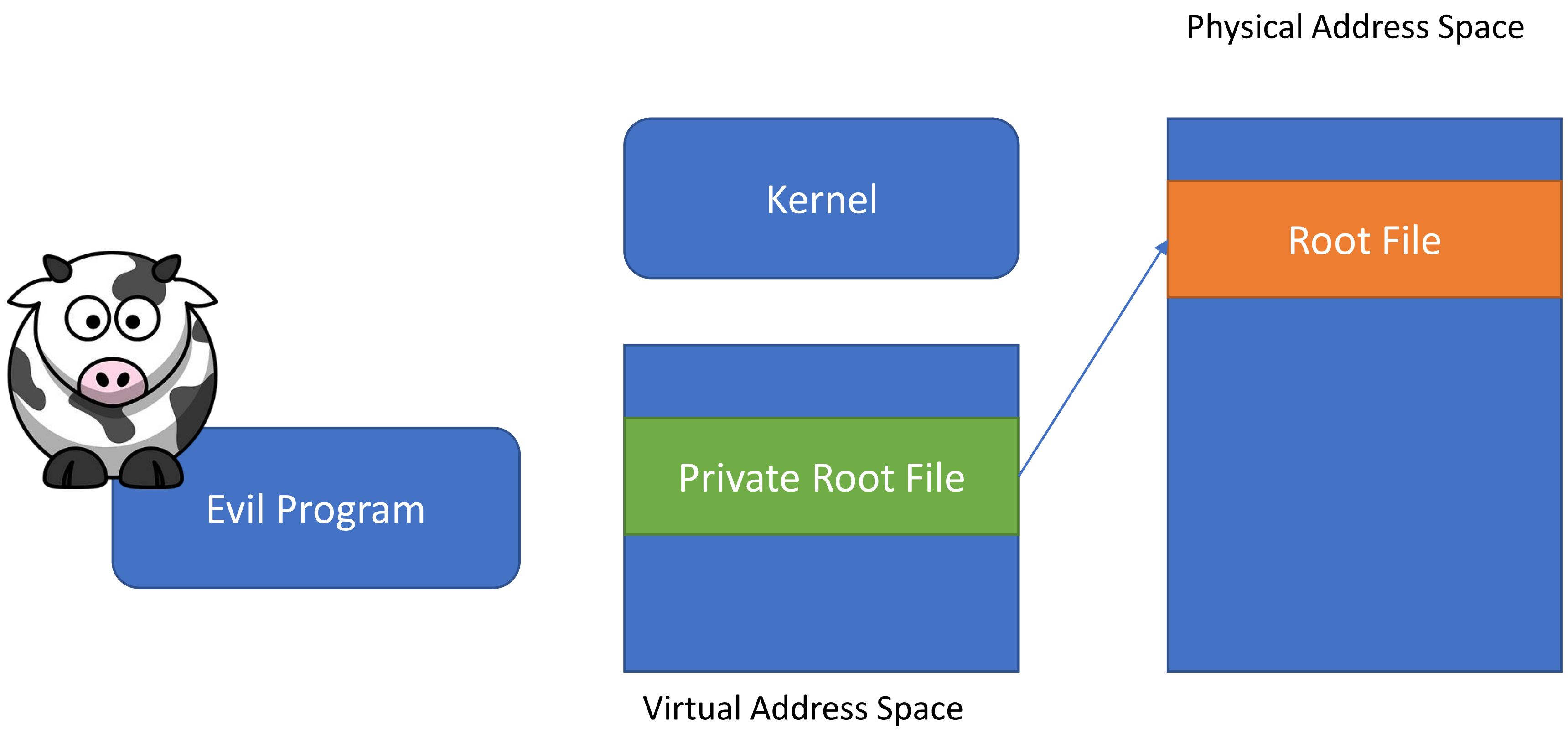


Dirty COW illustrated

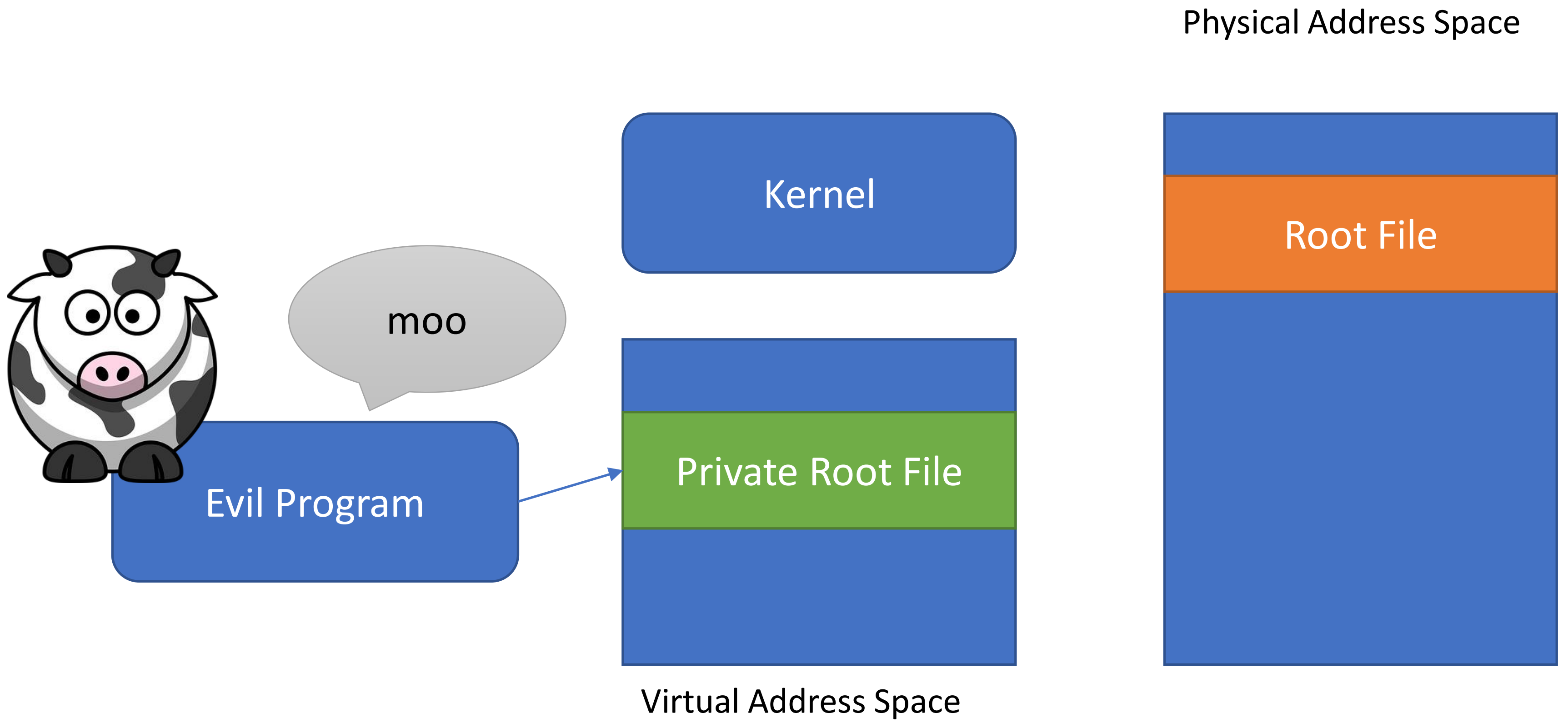
Copy on Write optimization! We don't need a copy until we Write!



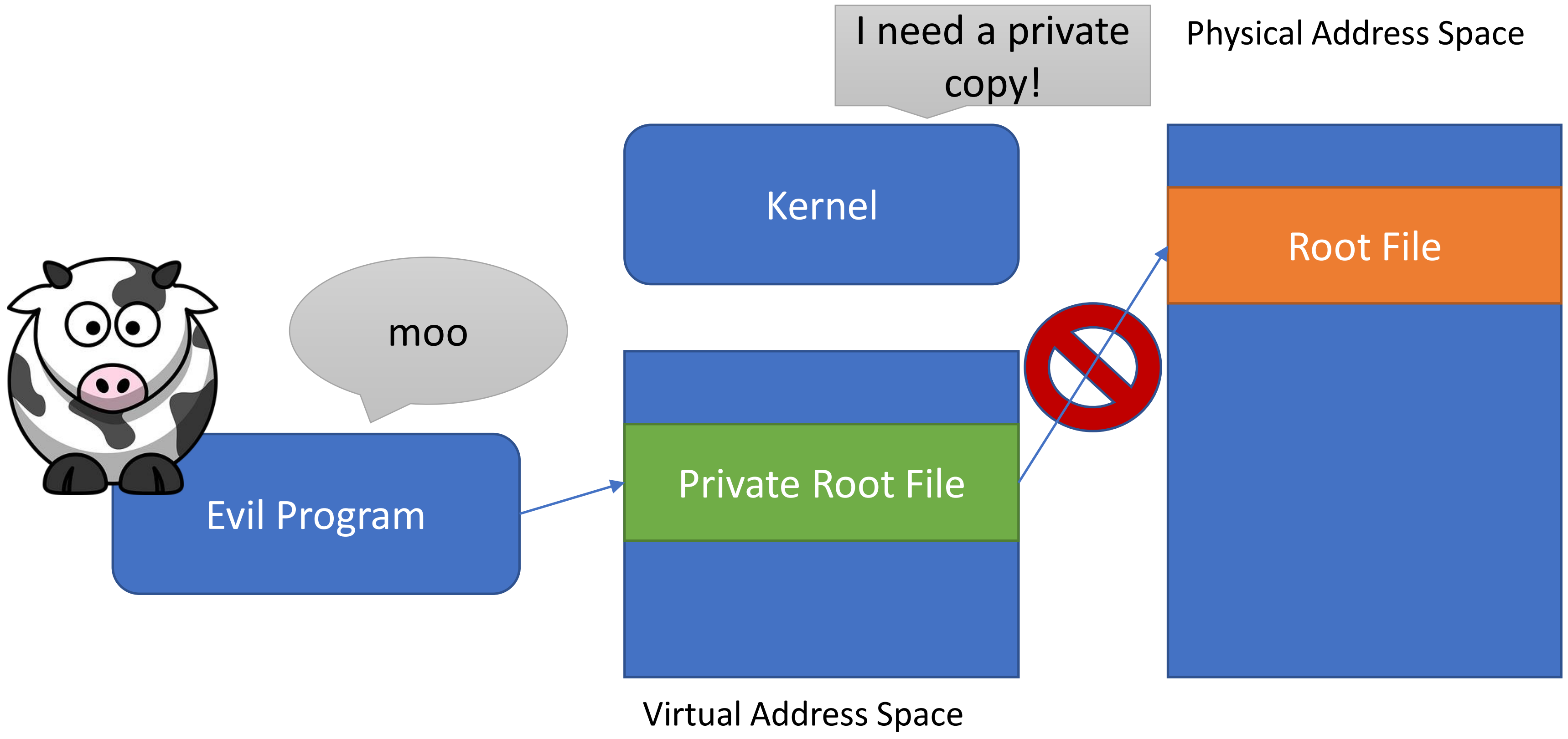
Dirty COW illustrated



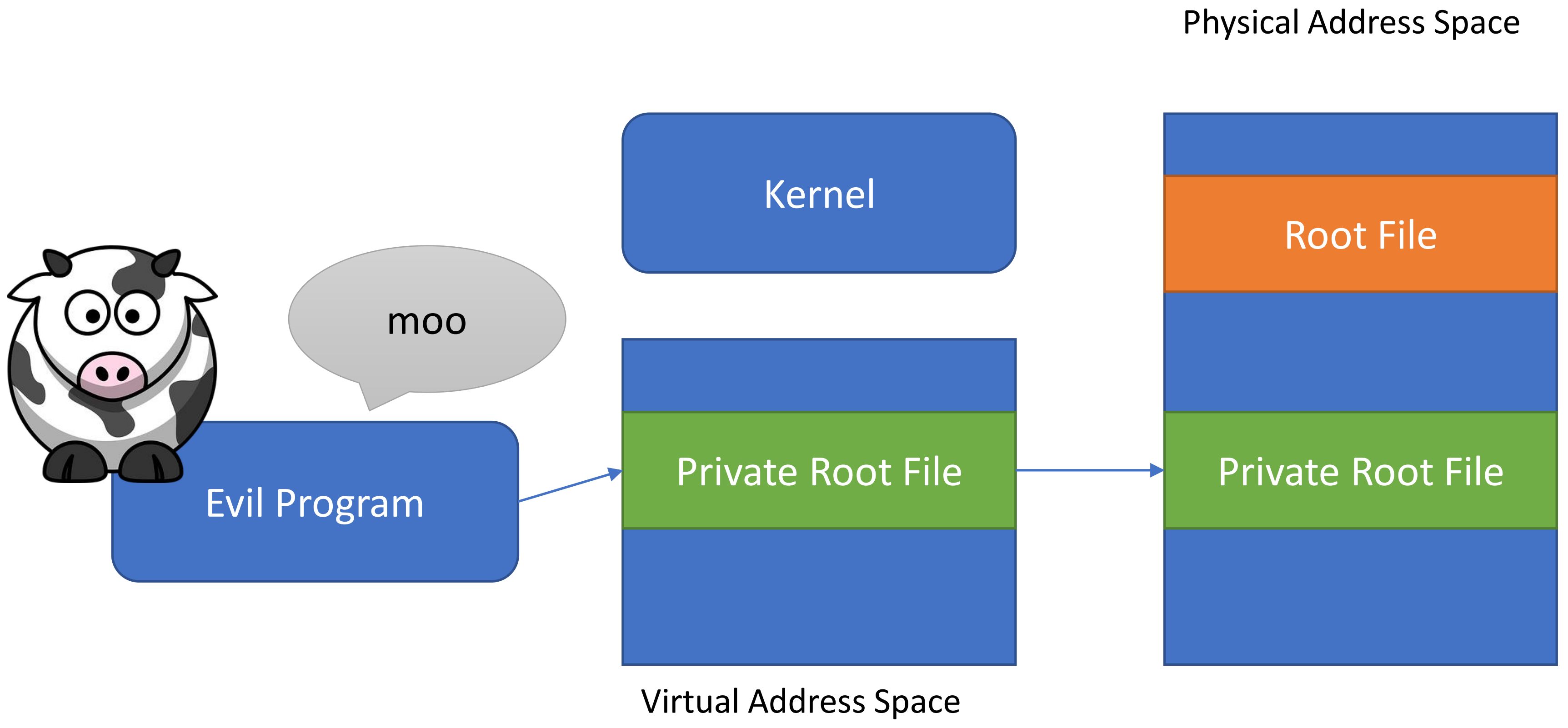
Dirty COW illustrated



Dirty COW illustrated

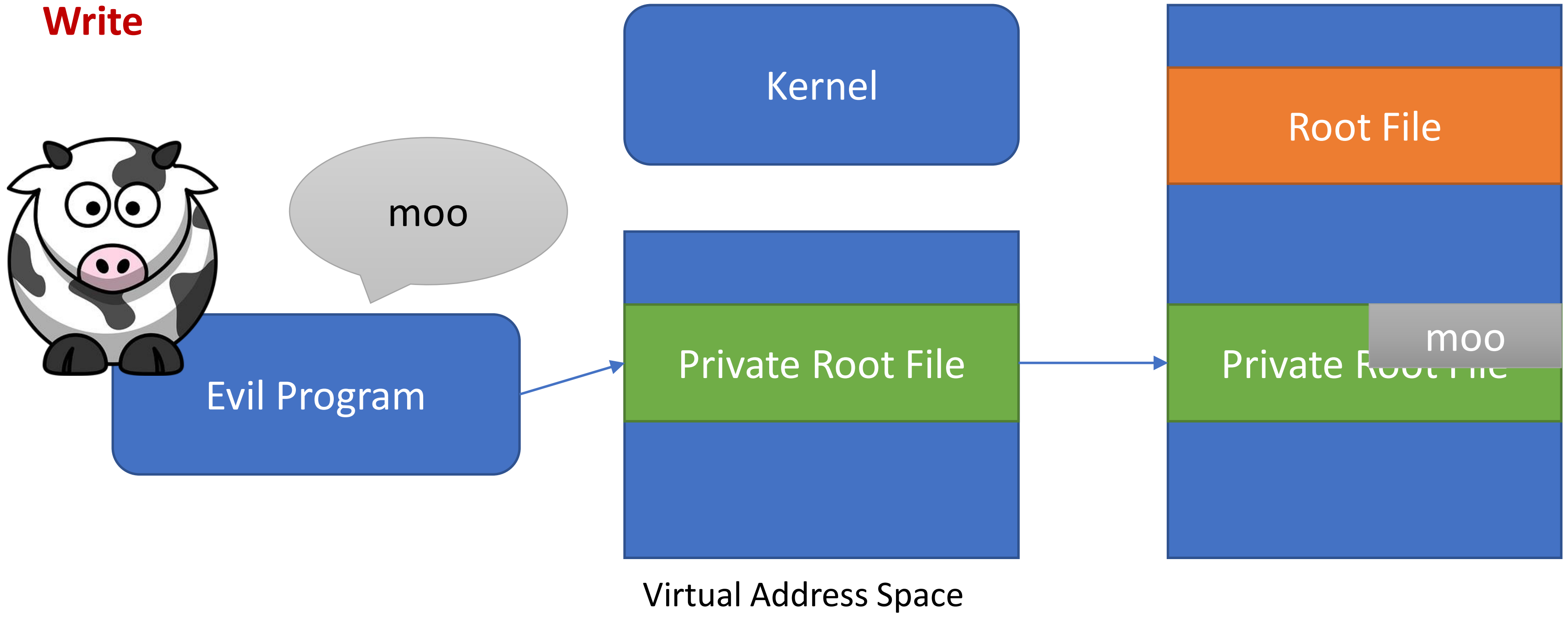


Dirty COW illustrated



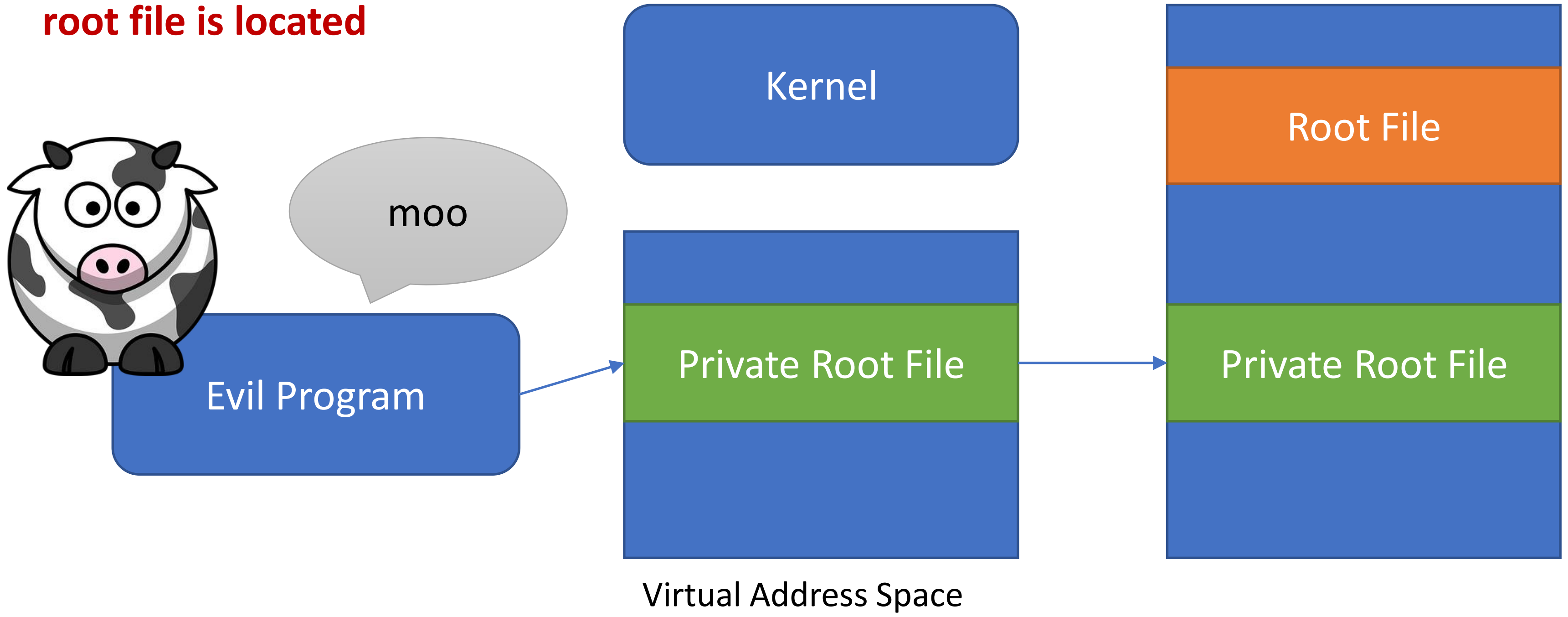
Dirty COW illustrated

TWO STEPS:
Locate the address
Write

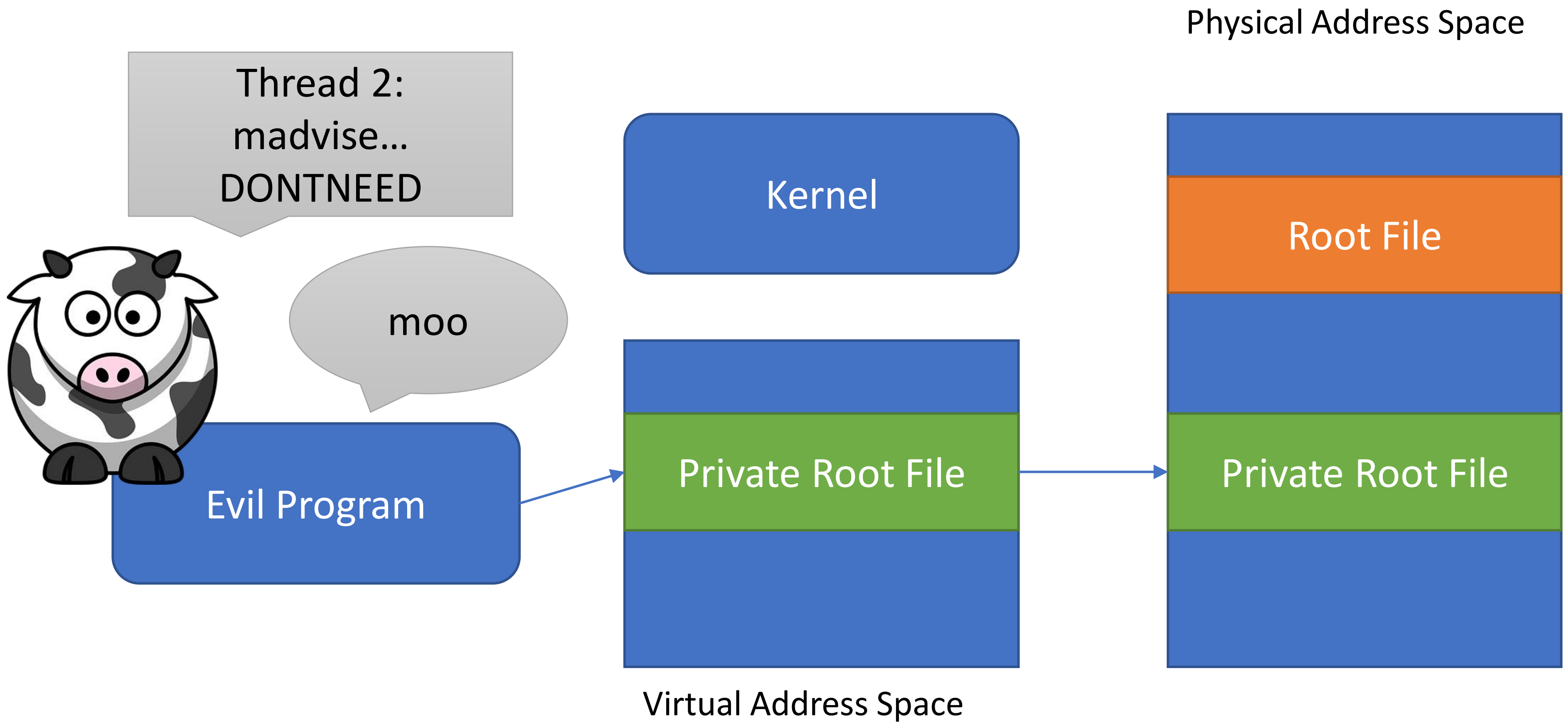


Dirty COW illustrated

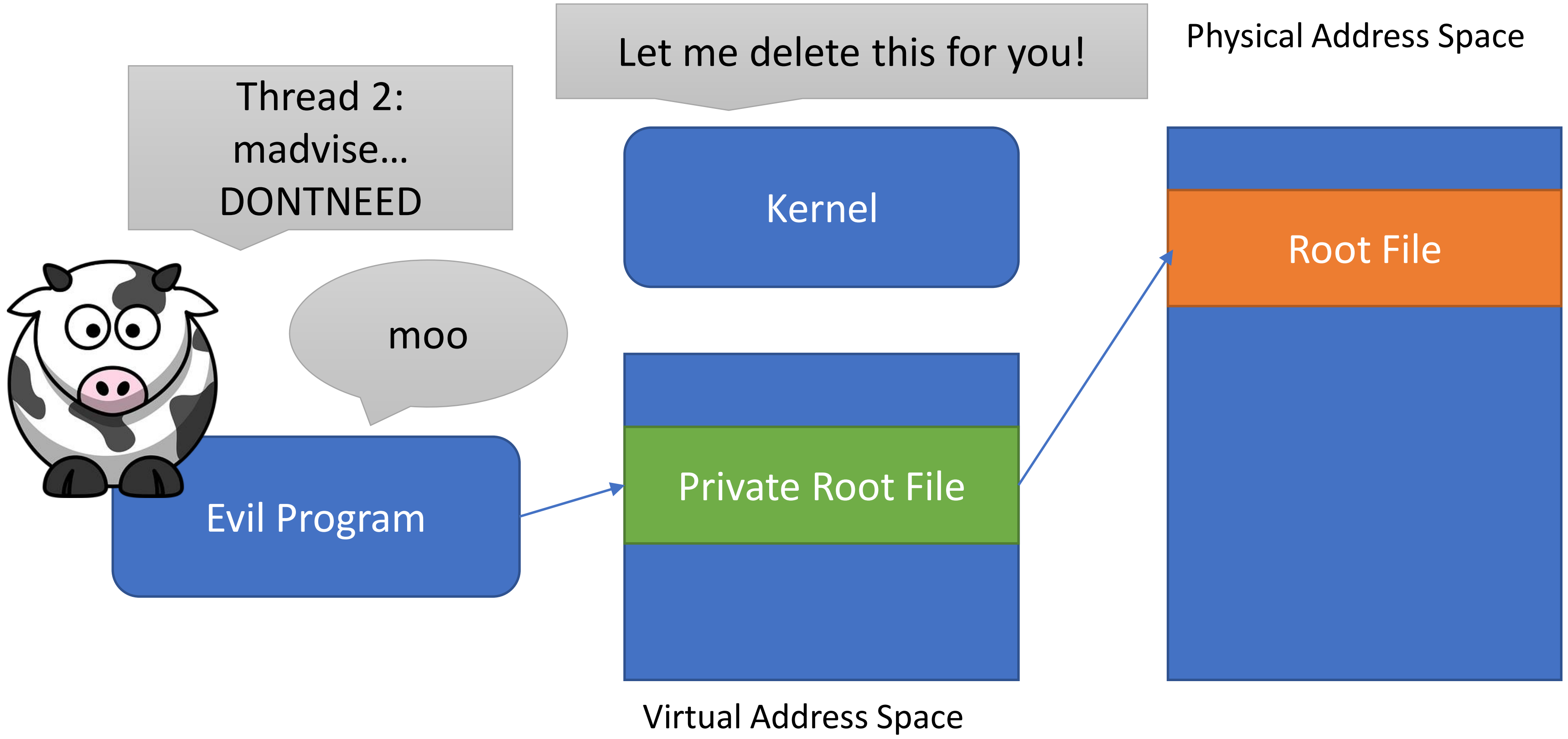
Let's break things!
Our evil program is writing, and the private root file is located



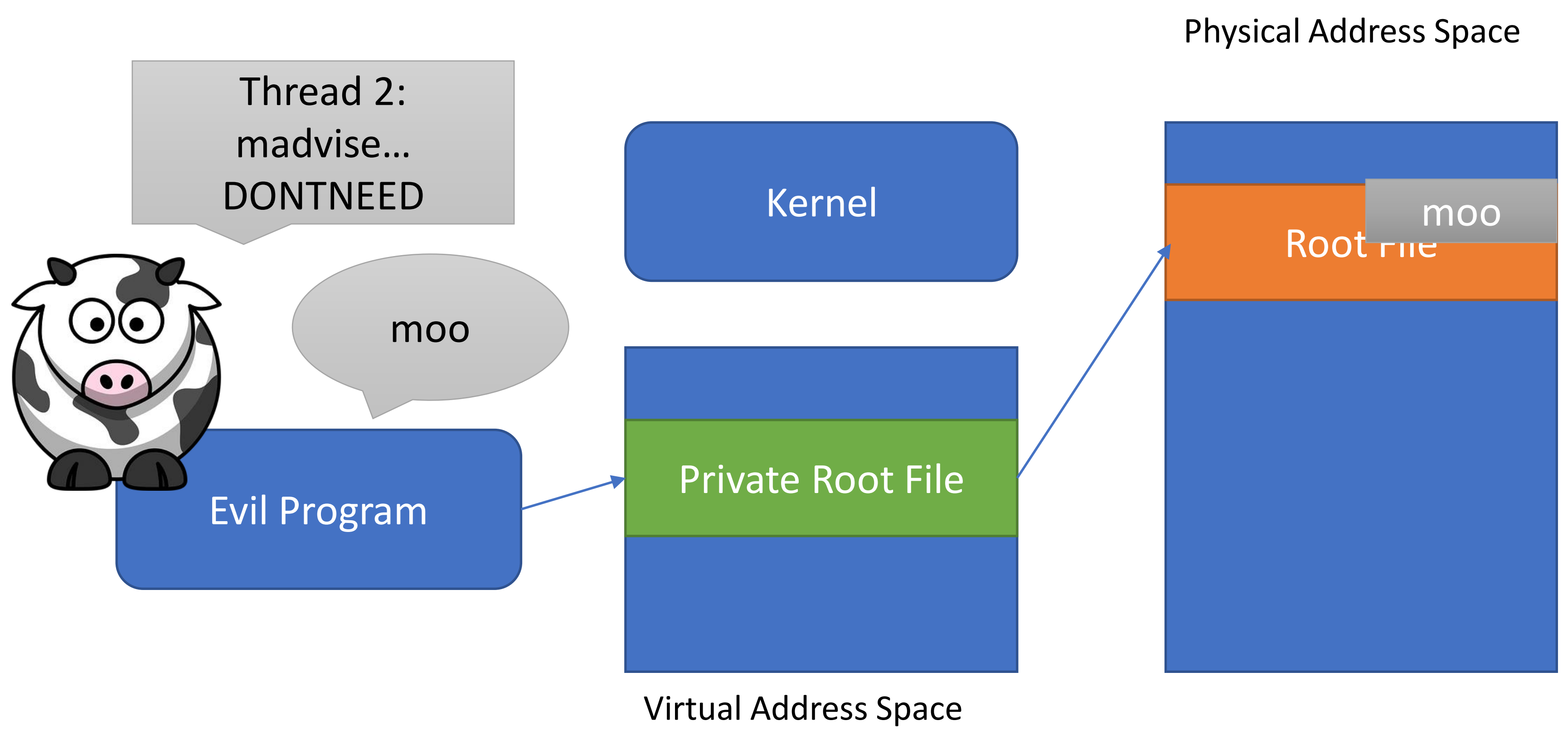
Dirty COW illustrated



Dirty COW illustrated



Dirty COW illustrated



Problem explained

- Locate and Write are not atomic!
  - First locate
  - Then Write!
- Concurrent thread can change where the virtual address point to!
- Write to the original physical address
- Even when you don’t have the privilege!

Problem explained

- Locate and Write are not atomic!
  - First locate
  - Then Write!
- Concurrent thread can change where the virtual address point to!
- Write to the original physical address
- Even when you don’t have the privilege!

Linux kernel: > 27 millions line of code

There is probably plenty of bugs…