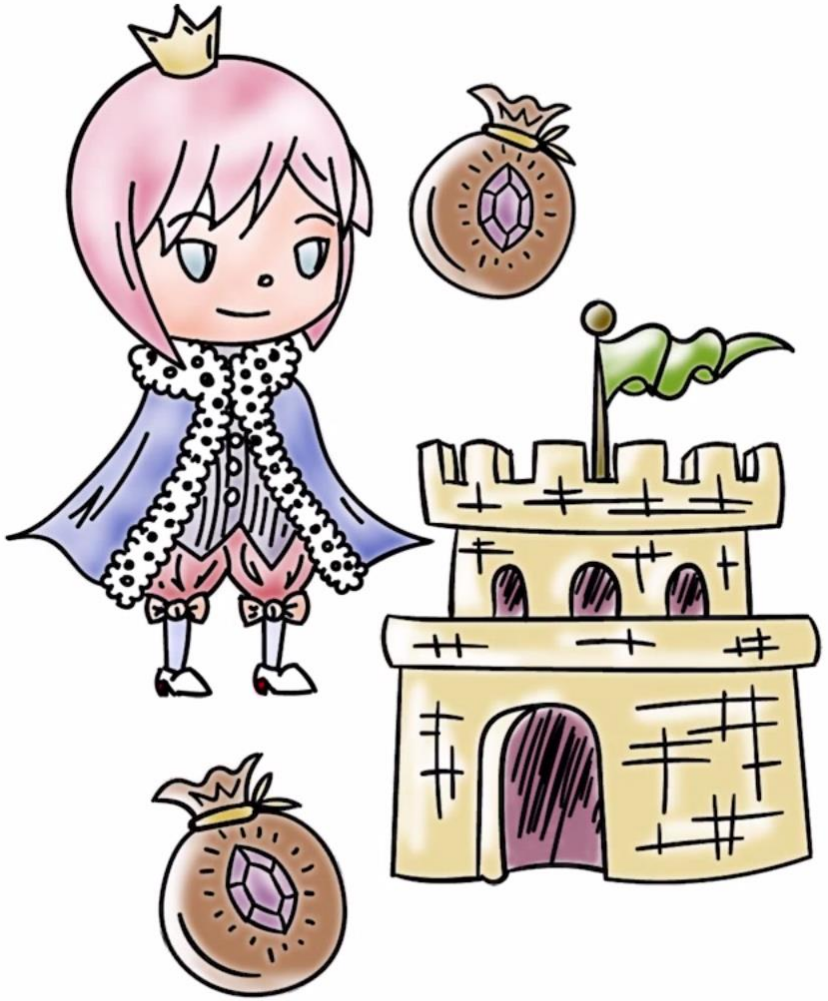
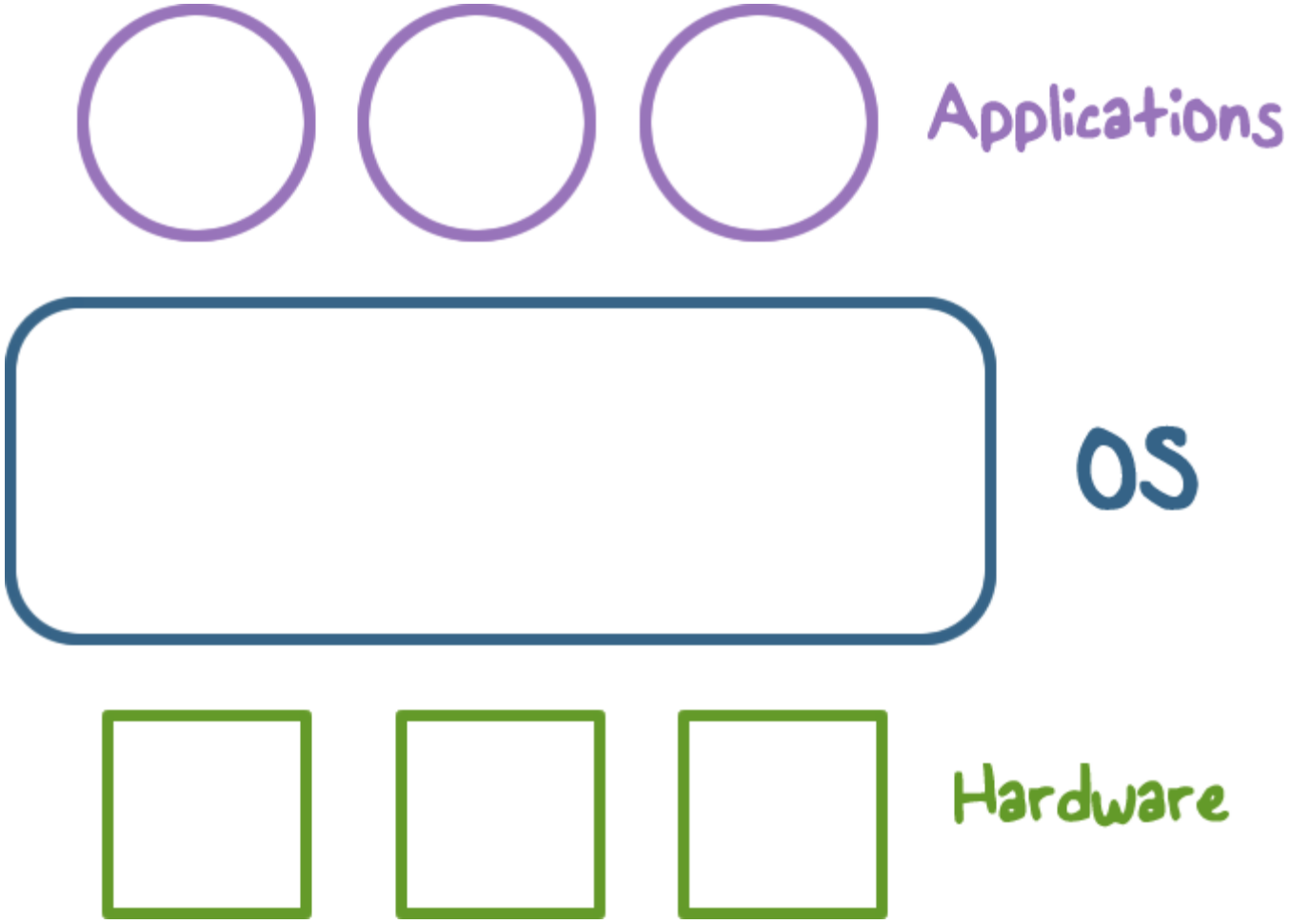
Operating Systems Security
Lesson Preview

- Understand the important role an **operating system** plays in computer security
- Learn about the **need for hardware support** for isolating OS from untrusted user/application code
- Understand **key trusted computing** base concepts

Operating Systems (OS)





Operating Systems

Operating System:

- Provides easier to use and high level abstractions for resources such as address space for memory and files for disk blocks.
- Provides controlled access to hardware resources.
- Provides isolation between different processes and between the processes running untrusted/application code and the trusted operating system.

Need for Trusting an Operating System

Why do we need to **trust** the operating system?
(AKA a **trusted computing base or TCB**)
What requirements must it meet to be trusted?

TCB Requirements:

1. Tamper-proof,
2. Complete mediation, and
3. Correct

TCB and Resource Protection

TCB Controls access to protected resources

- Must establish the source of a request for a resource (authentication is how we do it)
- Authorization or access control
- Mechanisms that allow various policies to be supported

Secure OS Quiz #1

A computer vendor ad claimed that its computers (including the OS they ran) were more secure. This claim could be based on one or more of the following:

- ☐ This vendor's more secure OS met TCB requirements while the others did not.
- ☐ The two OS were similar as far as security was concerned but one was not as big a target.
- ☐ The more secure OS could be much simpler than the other one.

Secure OS Quiz #2

A **system call** allows application code to gain access to functionality implemented by the OS. A system call is often called a protected procedure call.

Is the cost of a system call:

- ☐ the same as a regular call
- ☐ higher than a regular call

Secure OS Quiz #3

Complete mediation ensures that the OS cannot be bypassed when accessing a protected resource. **How does the OS know** who is making the request for the resource?

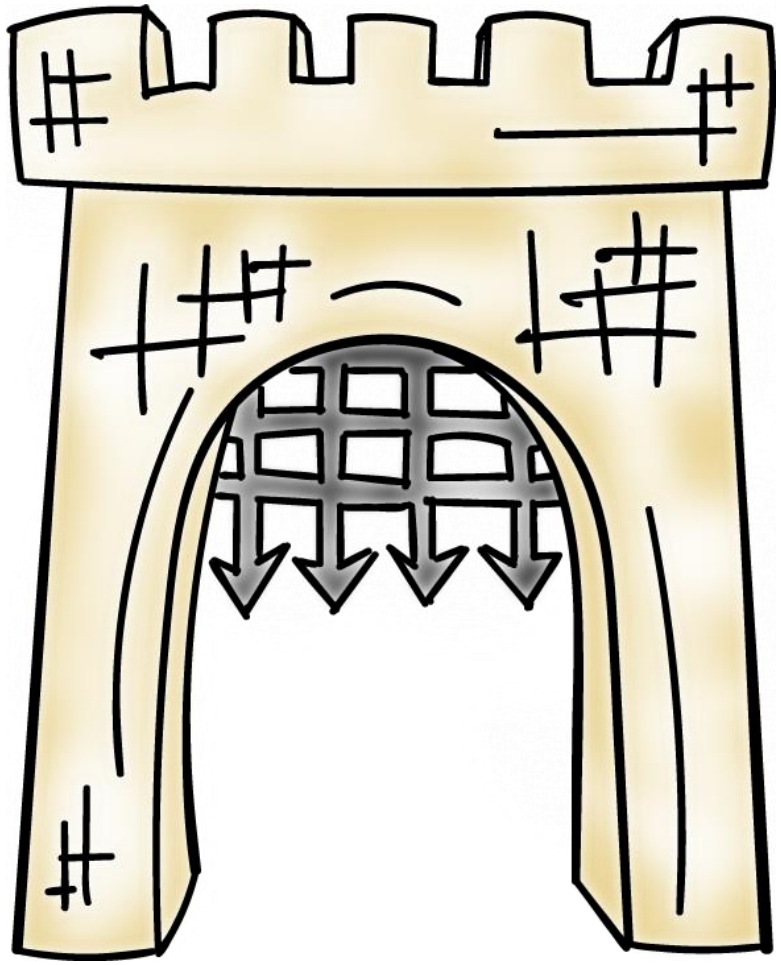
- ☐ Process runs on behalf of a user who must have previously logged in,
- ☐ Requested resource allows us to find out who must be requesting it

Isolating OS from Untrusted User Code

How do we meet the first requirement of a TCB (e.g., isolation or tamper-proof)?

- Hardware support for memory protection
- Processor execution modes (system AND user modes, execution rings)
- Privileged instructions which can only be executed in system mode
- System calls used to transfer control between user and system code

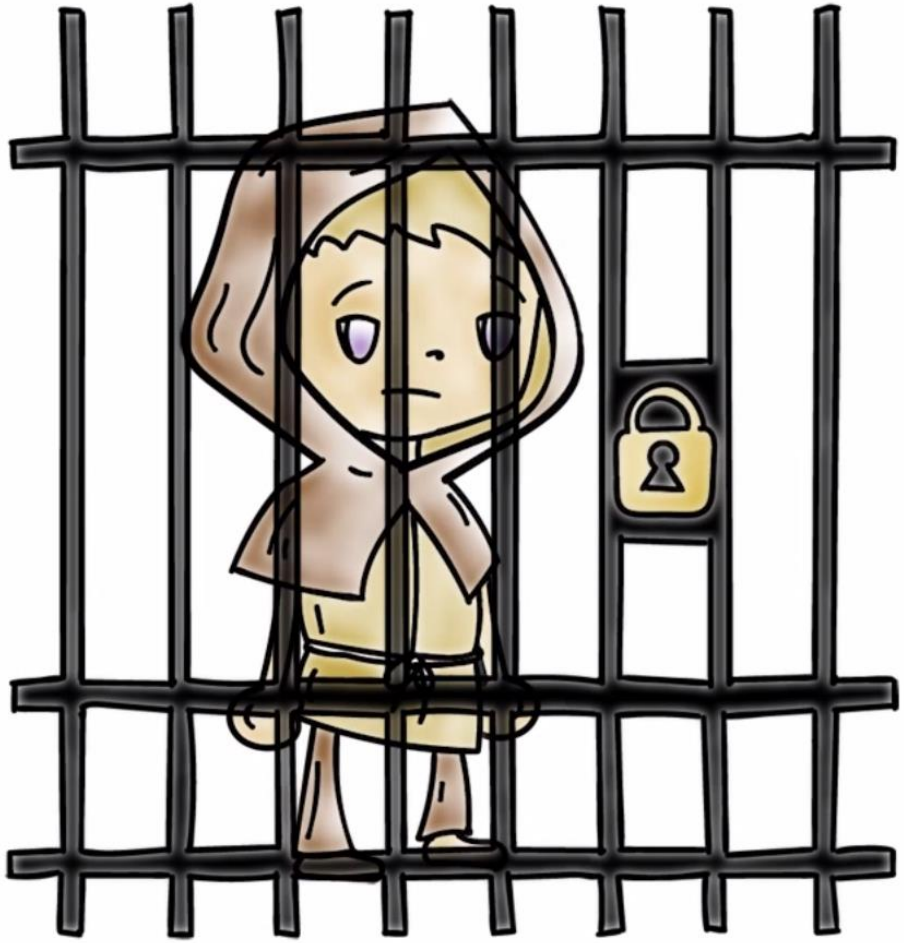
System Calls: Going from User to OS Code



System calls used to transfer control between user and system code

- Such calls come through **“call gates”** and return back to user code. The processor execution mode or privilege ring changes when call and return happen.
- x86 Sysenter/sysexit instructions

Isolating User Processes from Each Other



How do we meet the user/user isolation and separation?

OS uses hardware support for memory protection to ensure this.

Tampering with the OS Quiz

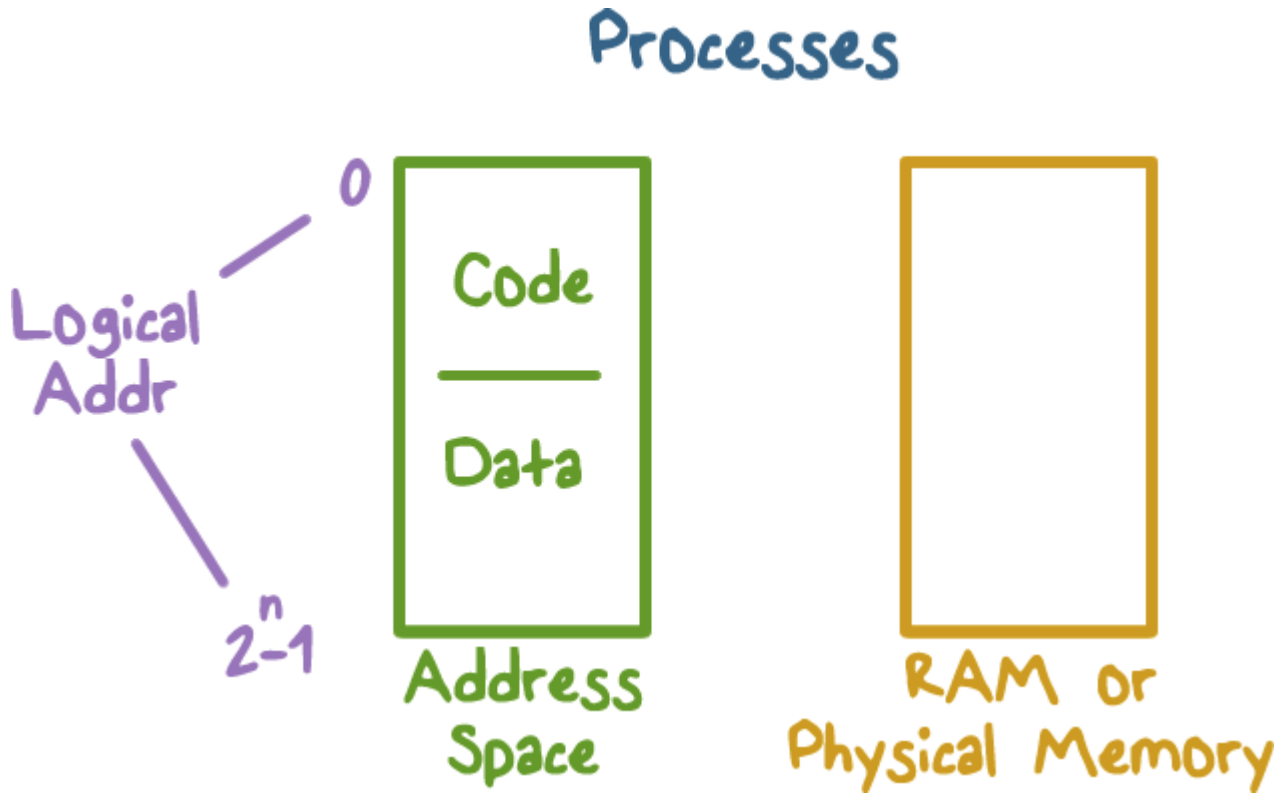
Which of these methods have been shown to allow hacker access to ‘secure’ memory belonging to the OS?

- ☐ Modification of firmware by Thunderstrike malware via malicious devices that connect via Mac’s Thunderbolt interface
- ☐ Exploiting the ‘refresh’ mechanism of a Dynamic RAM for privilege escalation
- ☐ Exploiting OS code buffer overflow vulnerability

Address Space: Unit of Isolation

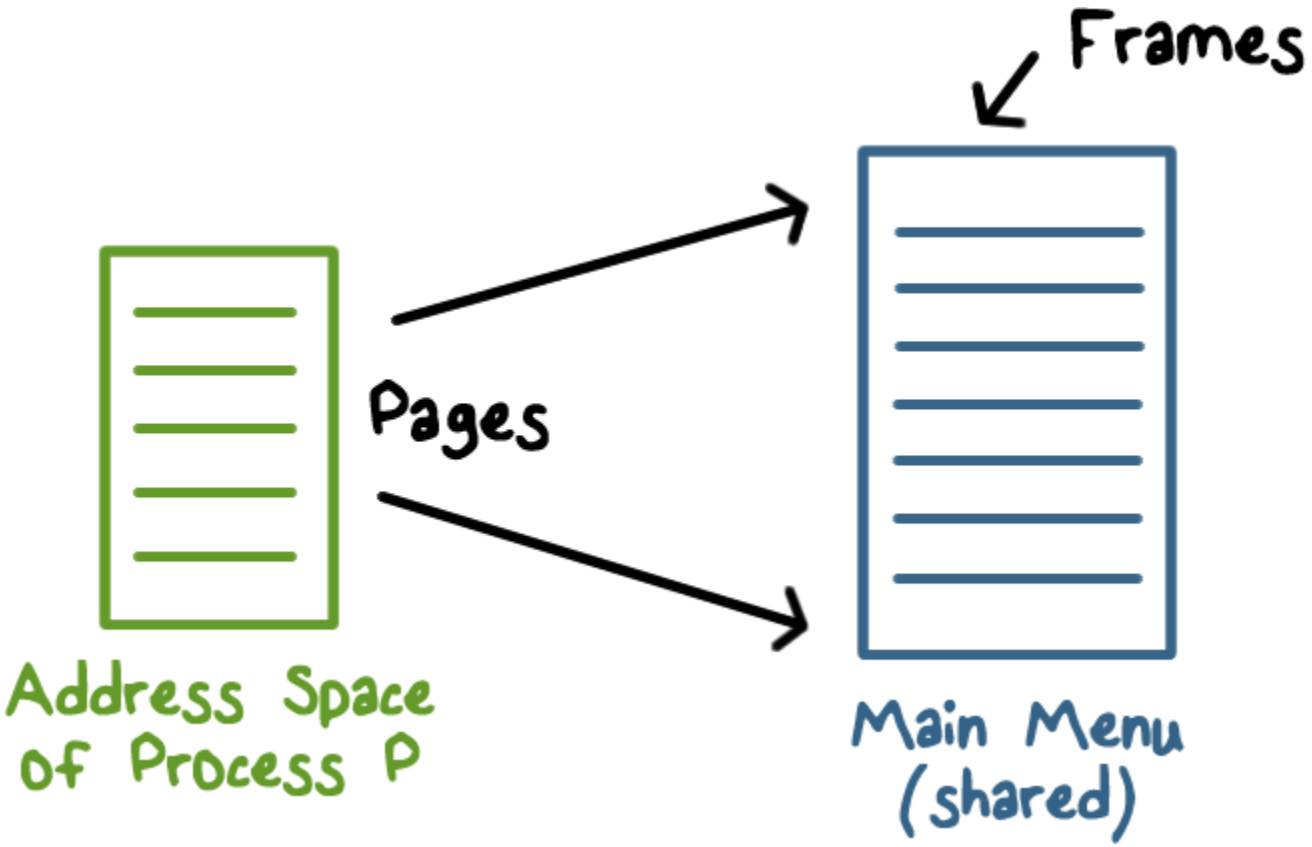
Processes view memory as **contiguous often larger than available physical memory**

- Usually 2^{32} or 2^{64} addresses
- Each process has its own mapping



Address Translation

Operating system **maps logical virtual addresses or pages onto physical memory frames**

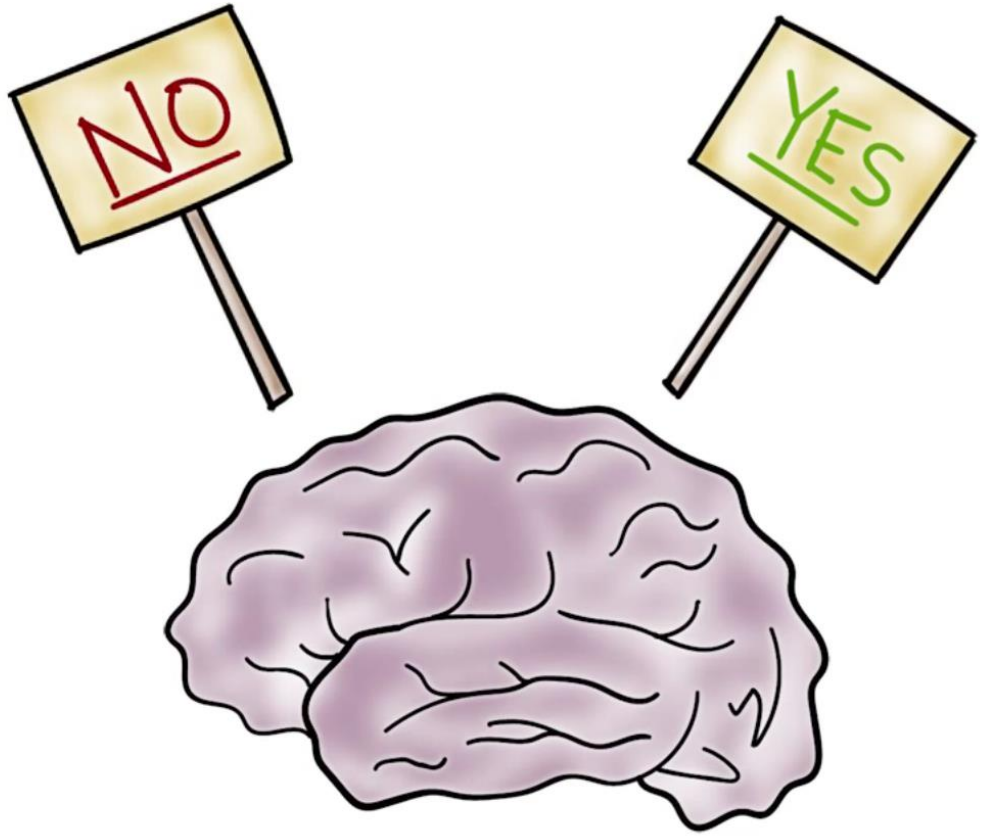


Process Data/Code Protection

OS will not map a **virtual page of process A** to a **physical page of process B** unless **explicit sharing** is desired.

- Process A **cannot access** process B's memory because it has no way to name/reach its memory.
- Page tables **managed by OS**

Process Protection through Memory Management



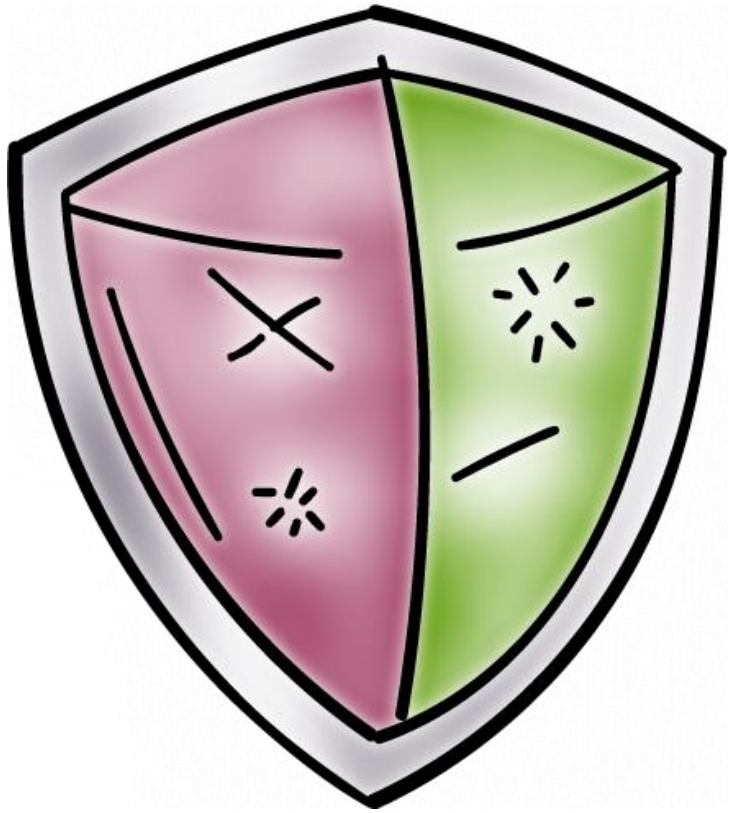
- Processor memory management unit (MMU) **uses page tables to resolve virtual addresses to physical addresses.**
- RWX bits on pages **limit type of access** to addressable memory

Revisiting Stack Overflow Quiz

The stack **can be exploited** through:

- ☐ Overflowing the buffer to change the return address to alter program execution
- ☐ Pushing data onto the stack to overflow the stack into the heap
- ☐ Popping data off the stack to gain access to application code.

Preventing Malicious Code Execution on the Stack through a Non-Executable Stack



Now think, how can we do a non-executable stack to **help prevent code injection via stack buffer?**

- **Used by Windows, OS X, Linux**

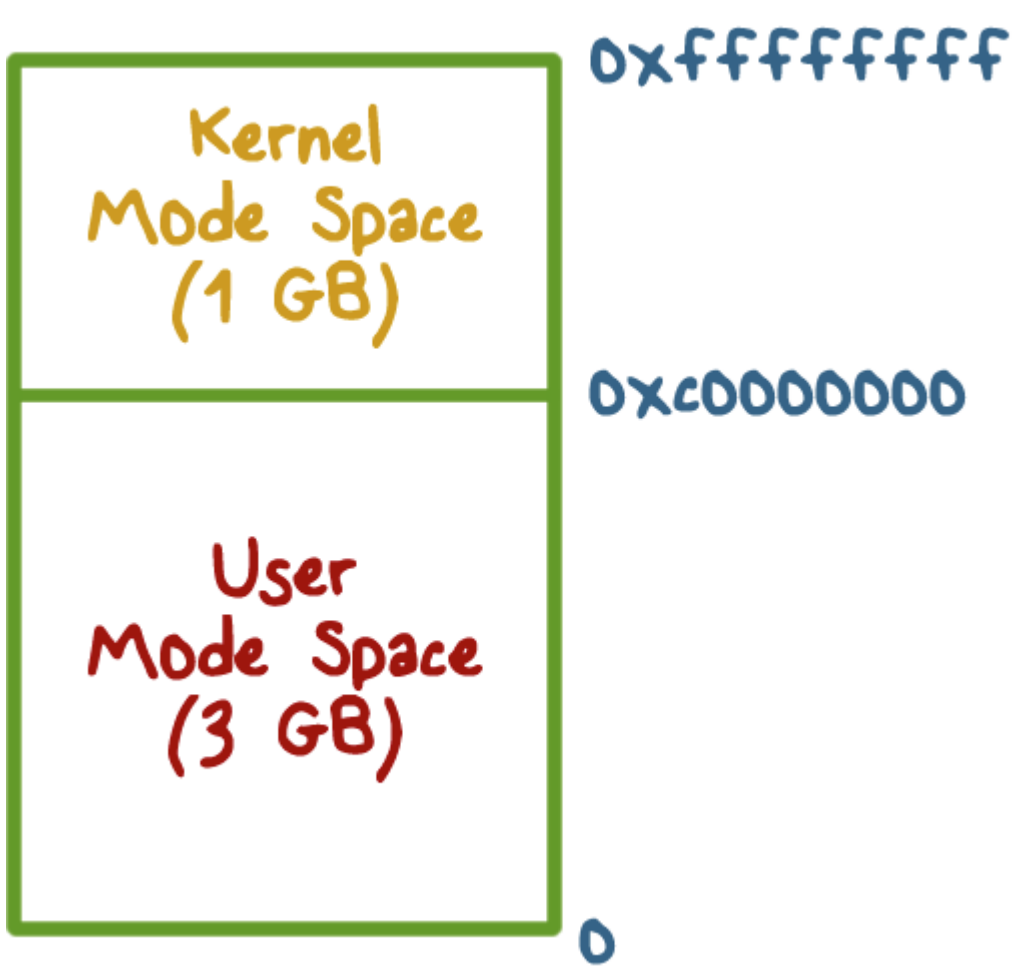
OS Isolation from Application Code

- OS (Kernel) resides in a portion of each process's address space.
- True for each process, **processes can cross the fence only in controlled/limited ways.**

OS Isolation from Application Code
Linux, DOS, OS X

- 32-bit Linux: Lower 3GB for user code/data, top 1GB for kernel
- Corresponds to x86 privilege ring transitions
- Windows and OS X similar
- DOS had no such fence, **any process could alter DOS and viruses could spread by hooking DOS interrupt handlers via kernel changes**

Linux User/Kernel Memory Split

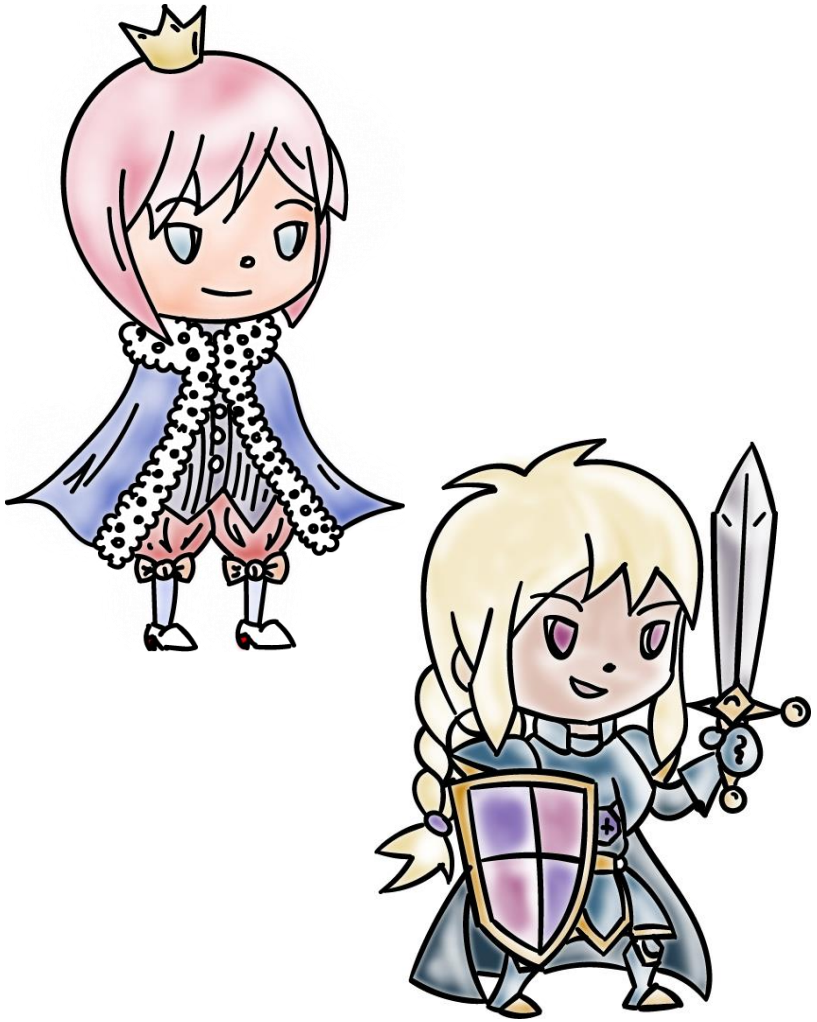


Execution Privilege Level Quiz

For the following described functions, Should it be executed in the operating system or if it can be executed in application code running in user mode?

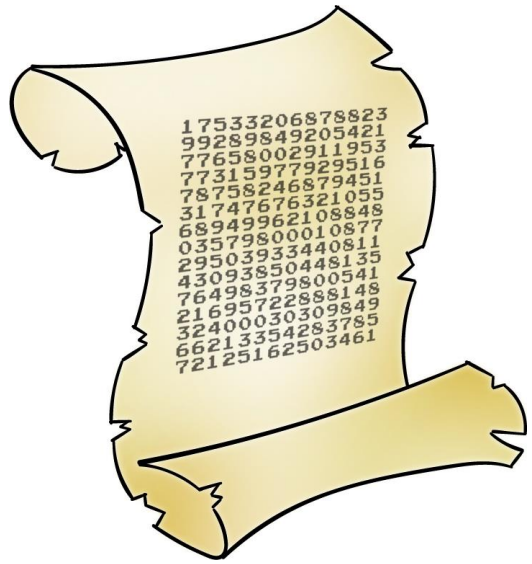
OS	User	
○	○	Switching CPU from one process to another when a process blocks.
○	○	Page fault handling
○	○	Changing who can access a protected resource such as a file
○	○	Setting up a new stack frame when an application program calls one of its functions

Complete Mediation: The TCB



- Make sure that no protected resource (e.g., memory page or file) could be accessed without going through the TCB
- TCB acts as a reference monitor that cannot be bypassed
- Privileged instructions

Complete Mediation: User Code



- User code cannot access OS part of address space without changing to system mode

- User code cannot access physical resources because they require privileged instructions (e.g. servicing interrupts) which can only be executed in system mode

Complete Mediation: OS

- OS virtualizes physical resources and provides an API for virtualized resources
- File for storing persistent data on disk
- Virtual resource must be translated to physical resource handle (e.g., file buffers) which can only be done by OS, which ensures complete mediation

Virtualization

- OS is large and complex, even different operating systems may be desired by different customers
- Compromise of an OS impacts all applications

Limiting the Damage of a Hacked OS

Use: Hypervisor, virtual machines, guest OS and applications

Compromise of OS in VM1 **only impacts applications running on VM1**

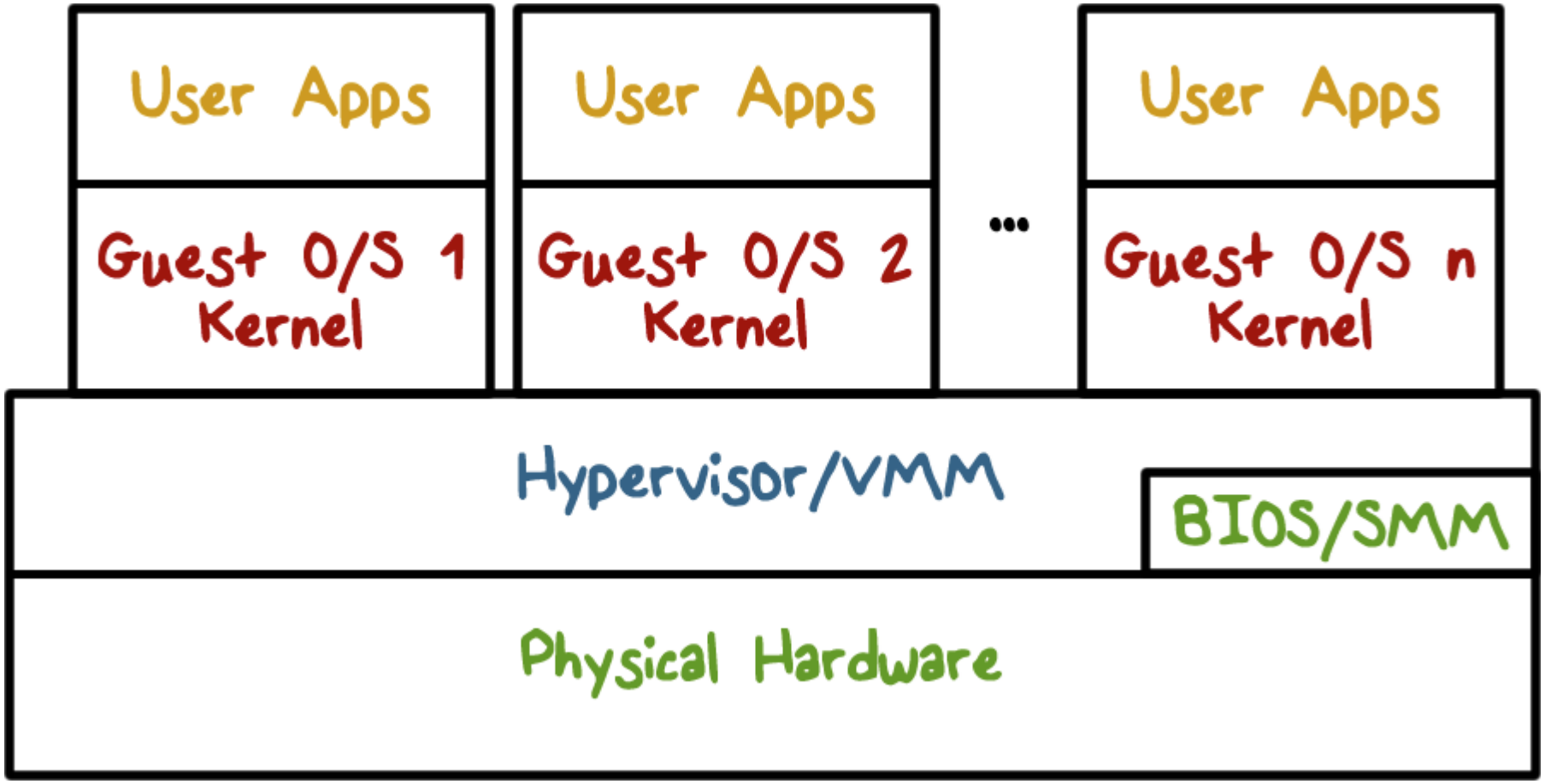
Limiting the Damage of a Hacked OS

Do your taxes in on VM1 while browsing potentially dangerous places on the web on VM2

What is the TCB here?

Hypervisor!

Virtualization Security Layers



Correctness:
The Final TCB Requirement

- Compromise of OS (TCB) means an **attacker has access to everything.**
- Getting the TCB right is extremely important
- **Smaller and simpler** (hypervisor only partitions physical resources among VMs and let us guest OS handle management)
- **Secure coding** is really important when writing the OS which typically is written in languages that are not type safe

TCB Requirements Quiz

An attack that exploits a vulnerability in an operating system **turns off the check** that is performed before access to a protected resource is granted.

What **TCB requirement is violated** as a result of this attack?

- ☐ Complete mediation
- ☐ Correctness
- ☐ Tamper-proof

Size of Security Code Quiz

Going from MS DOS to recent Windows operating systems, **what is a rough estimate for the multiplier** for the lines of code (e.g. multiplier is x if recent Windows OS is x times the number of lines of code in DOS)?

- ☐ Windows OS is **100x** larger than MS DOS
- ☐ Windows OS is **500x** larger than MS DOS
- ☐ Windows OS is **10,000x** larger than MS DOS

Hypervisor Code Size Quiz

The number of lines of code in a hypervisor is **expected to be smaller**. Xen is an open source hypervisor.

What is a rough estimate for the lines of code for the **Xen hypervisor**?

- ☐ 10,000
- ☐ 150,000
- ☐ 1,000,000

Operating Systems Security
Lesson Summary

- Understand the important role an OS plays in **protecting resources and applications**
- Understand how OS is **isolated from untrusted code** with hardware support for memory management
- Understand how **complete mediation** is provided.