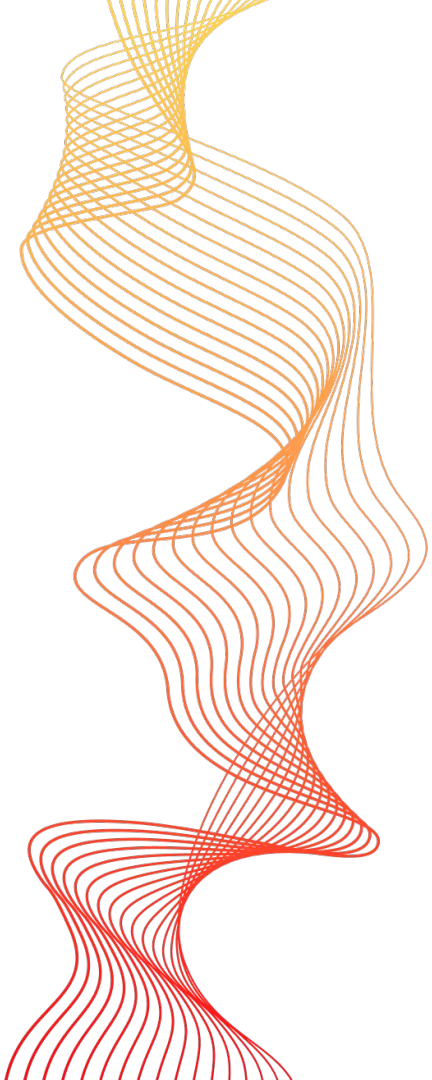
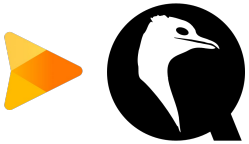
Kernel Debugging for Linux

An introduction to kernel debugging techniques for identifying vulnerabilities in the Linux kernel or developing exploits.

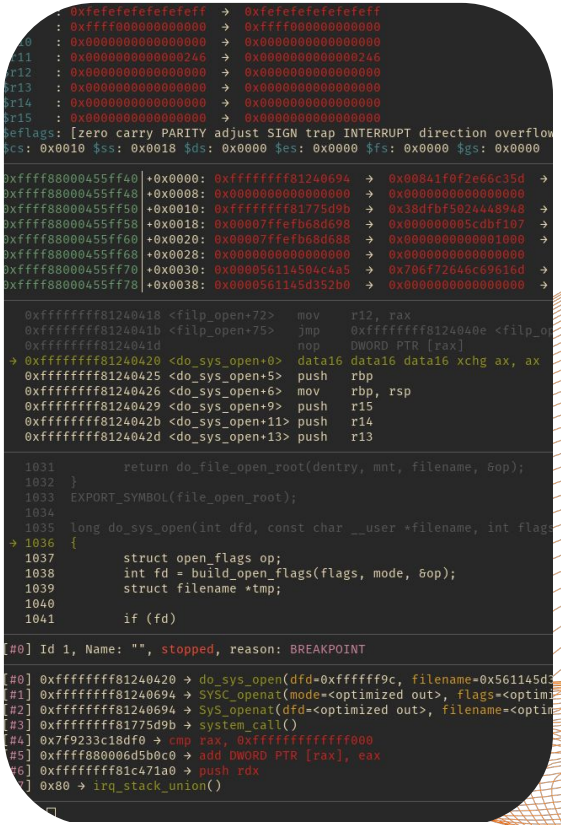




Kernel Debugging Techniques

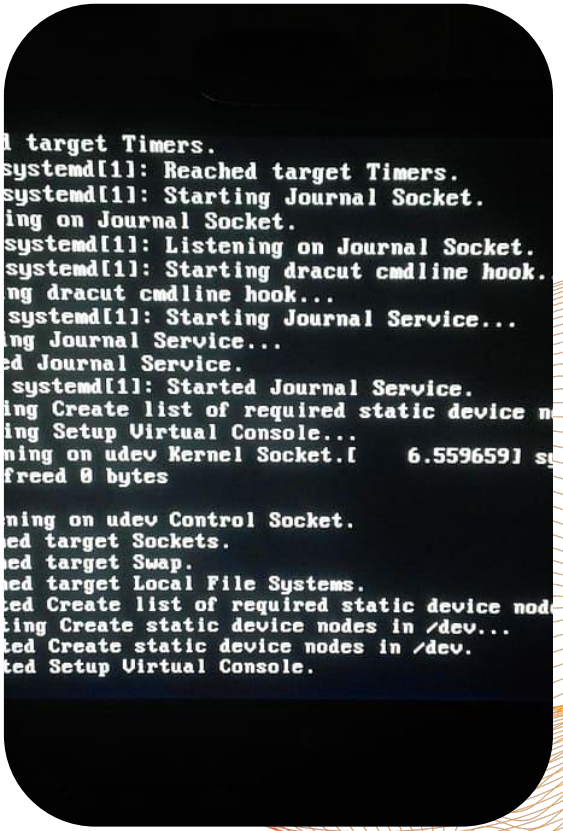
- Introduction to kernel debugging techniques for identifying vulnerabilities or developing exploits

- Practical exercises using prepared kernel modules with vulnerabilities



What is the Kernel?

- The kernel is a system program responsible for the main functions of the operating system
- From bootup to shutdown, the kernel controls hardware and loads applications
- The kernel has absolute authority in the system



Kernel Vulnerabilities

- Access to system resources is controlled by the kernel based on permissions
- A vulnerability in the kernel can allow a hacker to act beyond their privileges

Figure 4. Console output when height = 16

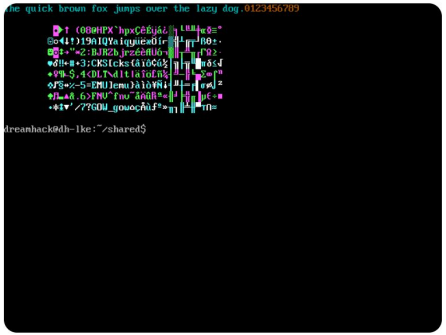
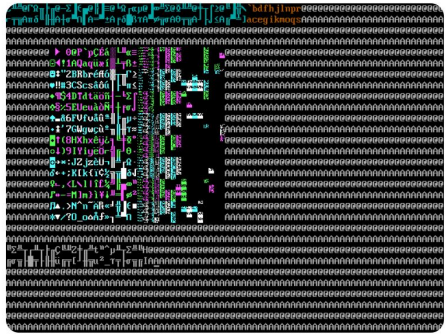


Figure 5. Console output when height = 32

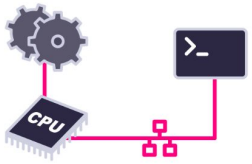
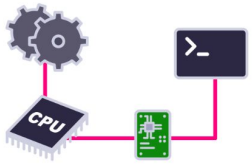
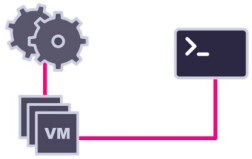


Importance of Patching

- Device drivers that need to be executed with the privileges of the kernel are frequently developed and patched

- If a serious security problem is found in the kernel, it must be patched immediately

Types of Kernel Debugging

Figure 1. Kernel Debugger	Figure 2. Hardware Debugger	Figure 3. Virtual Machine Debugger
		

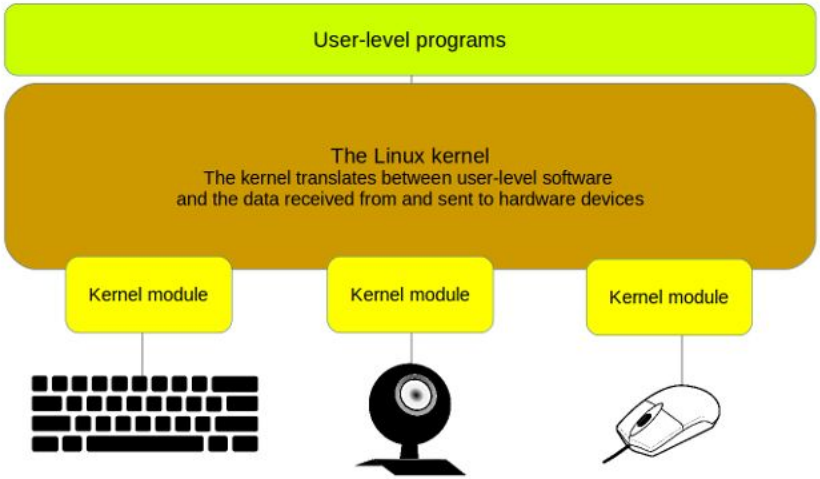
- Kernel Debugger - Issue debugging commands directly to the kernel.
- Hardware Debugger - Directly query and change the state of the circuit through JTAG, etc. used for hardware inspection. To use this method, separate debugging equipment is required.
- Virtual Machine Debugger - A way to utilize a virtual machine instead of running the kernel directly on top of hardware No special debugging equipment is required.

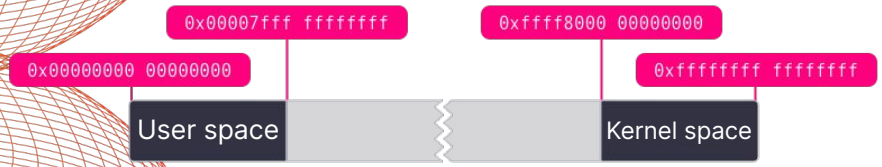
Loadable Kernel Modules (LKM)

LKM allows necessary functions to be attached or detached from the kernel to improve scalability

Kernel modules, when attached to the kernel, use the same

Vulnerabilities in kernel modules can lead to kernel





User Space and Kernel Space

- The kernel divides memory into user space and kernel space
- Running processes can only access user space in user mode
- In kernel mode, the processor has access to both user space and kernel space

Thank you for your time and attention 🙂