Level 0x09

Kernel Interfacing

Topics

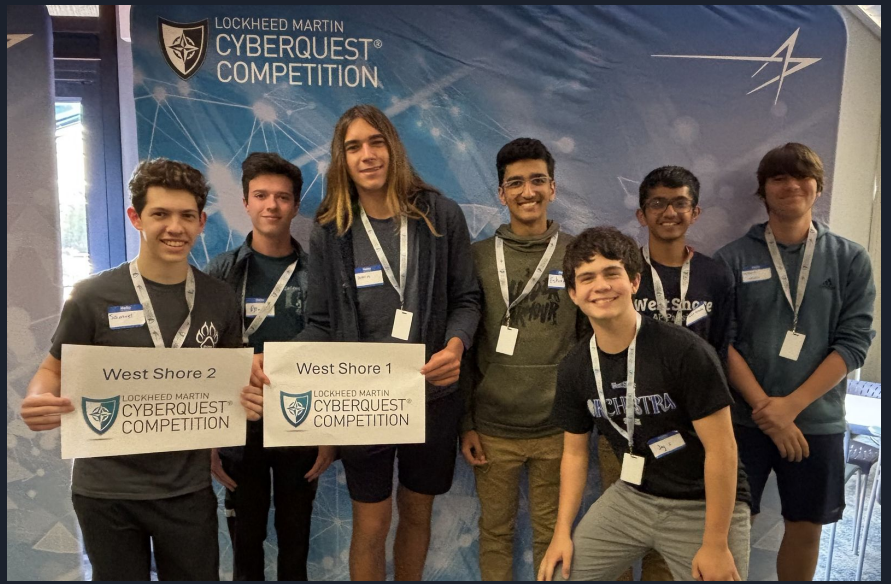
- Events
- Hacker History
- Linux syscalls

Ongoing Events

- December hacking contests
  - Advent of Cyber - TryHackMe.com
  - Sans Institute Holiday Hack Challenge - Less CTF, more game, probably more incident responder and defender focused
  - Advent of Code - Programming challenges. Used to be 25 2-parters, now reducing down to 12 days of challenges
  - Pwn.College is going to do an Advent of Pwn

Cyber Quest and Code Quest

- Code Quest
  - Saturday, Feb 28th
  - Registration Nov 17th -
- Spring Break
  - March 23rd - 27th
- Cyber Quest
  - Saturday, March 28th
  - Registration Jan 5th -



Richard Stallman aka rms

- Harvard Physics graduate
- Worked at MIT AI Lab… where a printer triggered him…
- Started the GNU Project (GNUs Not Unix)
  - GNU Compiler Collection (GCC), GNU make
  - GNU Debugger (GDB)
  - GPL License (Open Source License)
  - Free Software Foundation (Freedom: Libre vs Gratis)

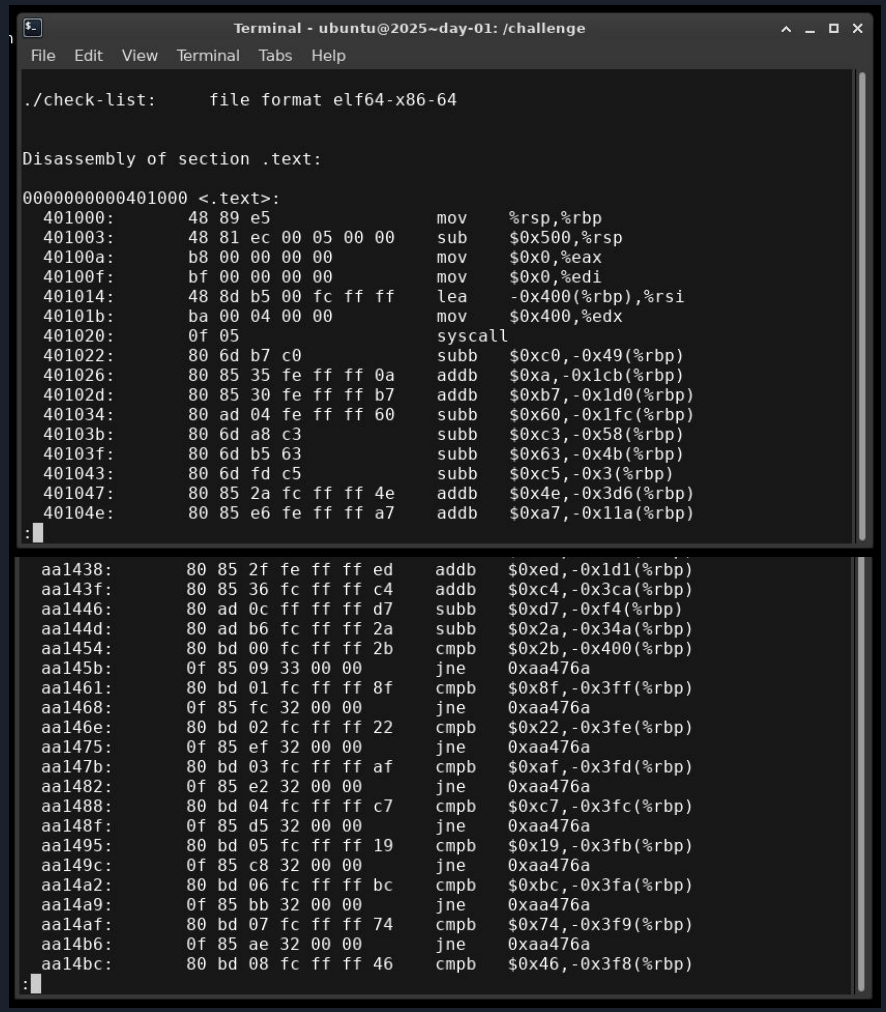
- MIT AI Lab had leadership involved with Jeffrey Epstein
- Has an insanely large speech rider
  - What pets are preferred at the place of stay
  - What codecs you are allowed to distribute speech in
  - Don’t even mention breakfast

Advent of Pwn Day 1

- Hint tells you to run **`objdump`**
  - -d is for disassemble

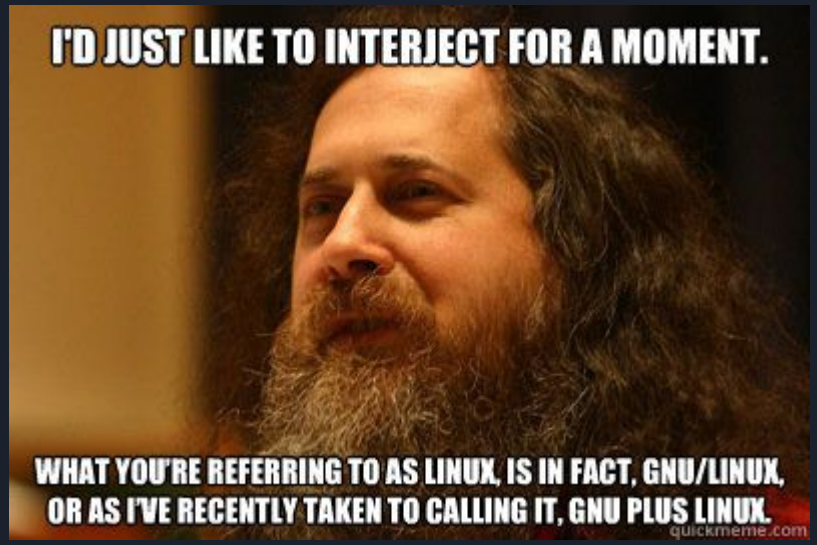
What we find in checklist:

- Some stuff
- syscall
- subb / addb instructions
  - Sooo many of these
  - Really, pages of them
- cmpb and jne instructions



What is inside an OS kernel

- Hardware Interfacing
  - Memory Access
  - SSD / Hard Drive control
  - Graphics Drivers
  - Keyboard / mouse interfacing
  - Ethernet / Wi-Fi
- Process control (threads)
- Filesystems
- Memory access
- Networking
- Enforces privilege

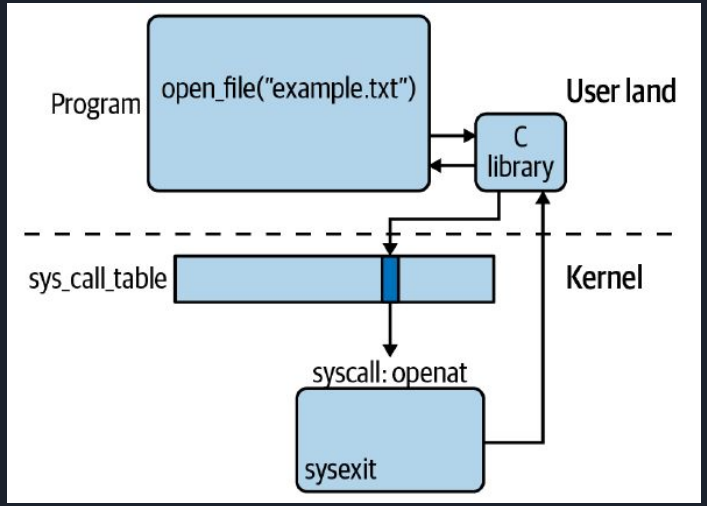


Trap / Kernel syscall

- We program / use applications in userspace
  - Code written in Java, Python, C/C++
  - Code is converted to machine code for our CPU
- We call libraries (other functions) to help us…
  - Read / write data from user or terminal
  - Draw images on the screen
  - GUI interfaces
- Libraries talk to kernel to do all of the above
- Kernel interfaces to the hardware

Example of reading data from a file

Trap (syscall) is special instruction, tells CPU to change privilege level and run some code as the kernel



System Call Table

- How does the kernel know what we want it to do we say “kernel do stuff” / syscall?

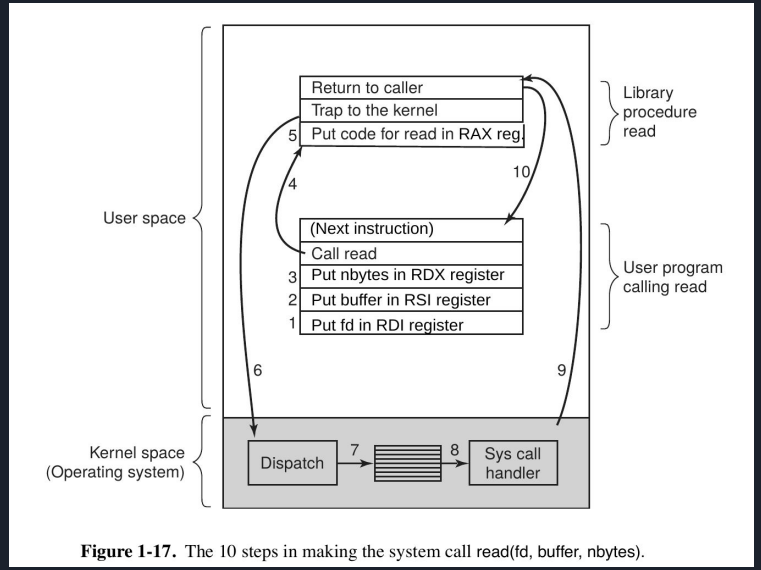


Figure 1-17. The 10 steps in making the system call read(fd, buffer, nbytes).

System Call Table

- How does the kernel know what we want it to do we say “kernel do stuff” / syscall?

- Pass arguments to kernel via CPU registers
  - RAX register is the sys call number

%rax	Name	Manual	Entry point
0	read	read(2)	sys_read
1	write	write(2)	sys_write
2	open	open(2)	sys_open
3	close	close(2)	sys_close
4	stat	stat(2)	sys_newstat
5	fstat	fstat(2)	sys_newfstat

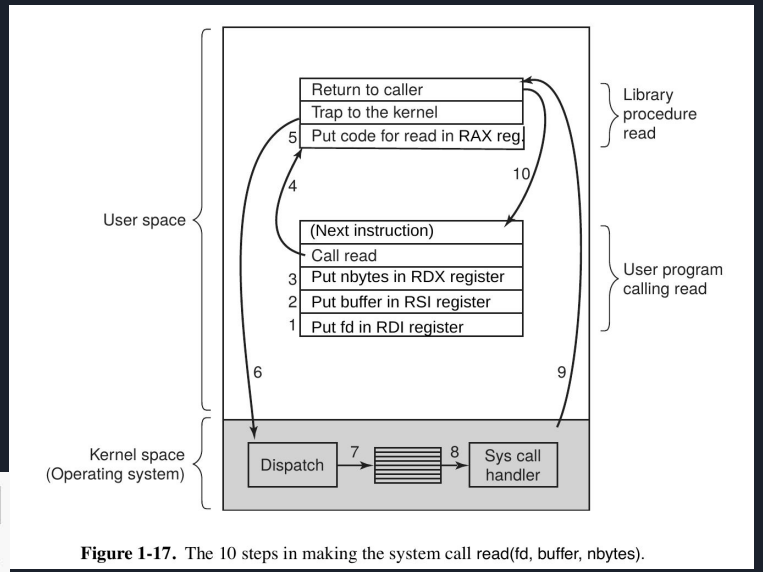


Figure 1-17. The 10 steps in making the system call read(fd, buffer, nbytes).

The syscall table is CPU arch specific. Many are similar, but each arch have differences

x86_64 (64-bit)

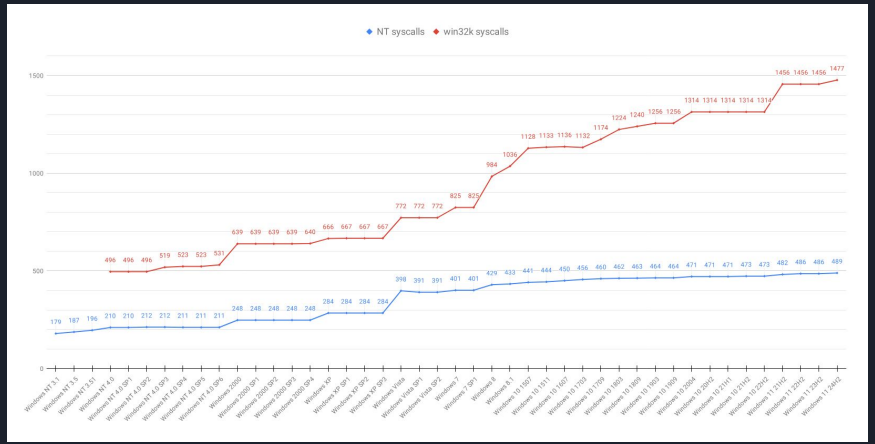
Compiled from Linux 4.14.0 headers.

NR	syscall name	references	%rax	arg0 (%rdi)	arg1 (%rsi)	arg2 (%rdx)	arg3 (%r10)	arg4 (%r8)	arg5 (%r9)
0	read	man/ cs/	0x00	unsigned int fd	char *buf	size_t count	-	-	-
1	write	man/ cs/	0x01	unsigned int fd	const char *buf	size_t count	-	-	-
2	open	man/ cs/	0x02	const char *filename	int flags	umode_t mode	-	-	-
3	close	man/ cs/	0x03	unsigned int fd	-	-	-	-	-
4	stat	man/ cs/	0x04	const char *filename	struct __old_kernel_stat *statbuf	-	-	-	-
5	fstat	man/ cs/	0x05	unsigned int fd	struct __old_kernel_stat *statbuf	-	-	-	-
6	lstat	man/ cs/	0x06	const char *filename	struct __old_kernel_stat *statbuf	-	-	-	-
7	poll	man/ cs/	0x07	struct pollfd *ufds	unsigned int nfds	int timeout	-	-	-
8	lseek	man/ cs/	0x08	unsigned int fd	off_t offset	unsigned int whence	-	-	-
9	mmap	man/ cs/	0x09	?	?	?	?	?	?
10	mprotect	man/ cs/	0x0a	unsigned long start	size_t len	unsigned long prot	-	-	-
11	munmap	man/ cs/	0x0b	unsigned long addr	size_t len	-	-	-	-
12	brk	man/ cs/	0x0c	unsigned long brk	-	-	-	-	-

Linux vs Windows

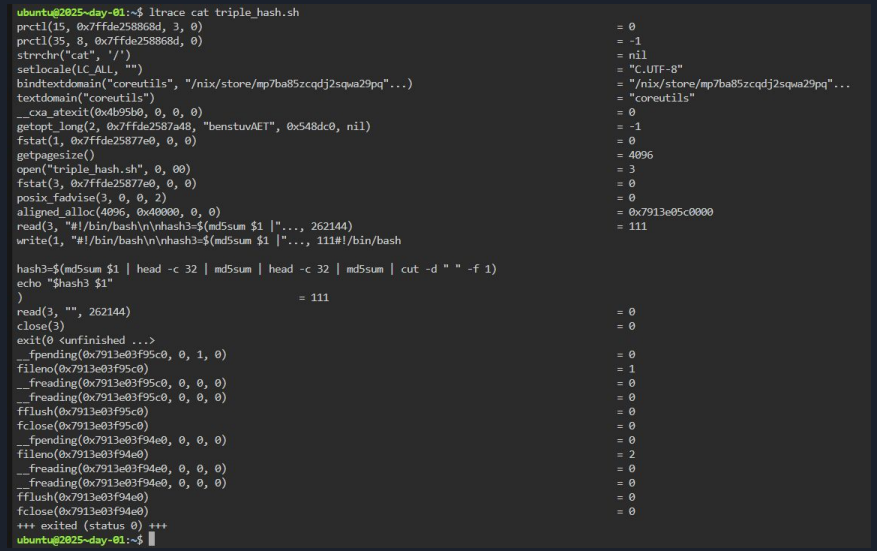
- Linux
  - 300-400 syscalls
  - They rarely ever get changed

- WinNT kernel
  - 400+ syscalls
  - Change a lot / don't use directly!
  - Closed source / not documented
- Win32 API / ntdll.dll
  - 1000s of functions
  - How developers should access the kernel

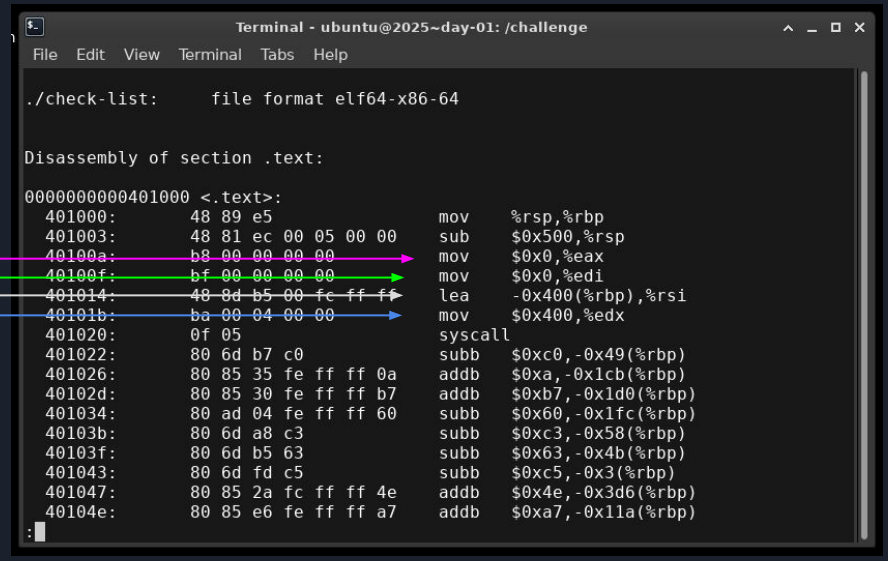


Linux Trace Commands

- **`strace`**
  - Displays which kernel sys calls are made
  - Parameters passed to kernel
  - Return value of the sys call
- **`ltrace`**
  - Displays which library functions are called
  - Only works if program is dynamically linked to other libraries at runtime



Back to Advent of Pwn...



x86_64 (64-bit)

Compiled from Linux 4.14.0 headers.

NR	syscall name	references	%rax	arg0 (%rdi)	arg1 (%rsi)	arg2 (%rdx)	arg3 (%r10)	arg4 (%r8)	arg5 (%r9)
0	read	man/ cs/	0x00	unsigned int fd	char *buf	size_t count	-	-	-
1	write	man/ cs/	0x01	unsigned int fd	const char *buf	size_t count	-	-	-
2	open	man/ cs/	0x02	const char *filename	int flags	umode_t mode	-	-	-
3	close	man/ cs/	0x03	unsigned int fd	-	-	-	-	-

Links

- Tanenbaum and Bos, Modern Operating Systems (book)
- Hausenblas, Learning Modern Linux (book)
- https://filippo.io/linux-syscall-table/
- https://chromium.googlesource.com/chromiumos/docs/+/master/constants/syscalls.md
- https://j00ru.vexillium.org/syscalls/win32k/64/
-