CYBER BUKIJUTSU: The Linux Offensive Terminal Toolkit Journey

Curated by Breeze for Dante Choppa

Phase 1: Linux Offensive Foundations (2-3 weeks)

Goals:

- Write C programs in Linux using syscalls
- Interact with the system through /proc, ptrace, and raw sockets

Core Concepts:

- Linux process model & /proc
- File descriptors and redirection
- Signals and signal handling
- Forking, exec, and zombie processes
- ptrace and process tracing
- Raw socket creation (ICMP, TCP)

Projects:

- ps_clone: Build your own `ps` using /proc
- mini_sh: A minimal shell with I/O redirection
- sig_catcher: Log caught signals and who sent them
- process_killer: Search for a process by name and kill it
- net_sniffer: Capture packets using raw sockets (start with ICMP)

Phase 2: Offensive Toolchain - Terminal Edition (3-4 weeks)

Goals:

- Build real tools you'd find in a Red Team toolbox - no GUIs, just raw terminal power

Core Concepts:

- Ptrace-based injection
- Keylogging through /proc and input/
- Reverse/bind shells in pure C
- Dynamic loading with dlopen
- Statically vs dynamically linked binaries
- Manual ELF loading (intro)

Projects:

- reverse_shell: Simple TCP reverse shell in C

- bind_shell: Server-mode shell listener
- ptrace_injector: Inject code into a running process via ptrace
- keylogger: Log keystrokes from /dev/input (non-root and root modes)
- stealth_ls: A fake ls command that hides certain files
- env_backdoor: A backdoor triggered by a specific environment variable

Phase 3: Advanced Linux Offensive Concepts (3-4 weeks)

Goals:

- Master stealth, evasion, and persistence on Linux
- Explore ELF, LD_PRELOAD, and rootkits

Core Concepts:

- Dynamic linker hijacking (LD_PRELOAD)
- Hiding processes/files
- Custom shellcode execution
- Manual ELF parsing and loading
- LKM-based rootkits (optional kernel dive)

Projects:

- preload_backdoor: Hijack libc functions to add a secret backdoor
- elf_patcher: Modify an existing ELF to add payload
- memory_shellcode_runner: Run shellcode from memory stealthily
- anti_ptrace: Add anti-debugging to your tools
- lkm_hider: Kernel module to hide processes/files (optional)

Parallel Windows Track (1 hour, 2-3x per week)

Goal: Maintain basic proficiency with Windows internals and offensive strategies.

Topics:

- Process hollowing
- PE parsing basics
- DLL injection and reflective loading
- API hooking
- Sysinternals tools (ProcMon, Process Hacker, etc.)

Tasks:

- Reverse a PE file weekly
- Build tiny proof-of-concepts for Windows injection techniques

- Read a chapter from Windows Internals every 1-2 weeks

Core Resources

- The Art of Unix Programming - Eric Raymond
- Linux System Programming - Robert Love
- man, strace, lsof, readelf, objdump, gdb - your best friends
- Offensive CTFs with Linux targets (e.g., pwn challenges)
- Hacking: The Art of Exploitation
- objdump -d, gdb, pwndbg
- Write your own shellcode in NASM