Tools & Miscellaneous

Linux – *dmesg (I)*

- **possibility to analyse the kernel-level details**
  - analysis of the boot process
    - display messages from the kernel's ring buffer
    - system boot, hardware detection, driver initialization, and kernel errors
  - history of kernel interactions
    - kernel events like hardware connections, memory allocations, and peripheral issues.
  - system debugging (related to hardware)
    - USB devices, disk errors, CPU issues
    - hardware failures or driver errors are present in kernel logs
  - hardware configuration
    - detailed HW information, for configuration and/or troubleshooting

Linux – *dmesg (II)*

- filtering
  - cooperation with command-line tools
  - e.g. *Grep and dmesg* to focus on USB, CPU, memory-related logs, …
    - `dmesg | grep usb`
    - for USB device connections, helping to debug issues with USB peripherals.
- timeline reconstruction
  - `-T` option for human-readable timestamps for kernel messages

Linux – *kill*

- **It is 1) a function for signal delivery 2) a shell command to**
  - send signals to processes to instruct them to perform specific actions (e.g., terminate, pause, or continue execution).
  - targets specific processes (or process groups) using the process identifier (PID)
- **signals example**
  - SIGKILL (forceful termination), SIGTERM (graceful termination), SIGSTOP (pause), and SIGCONT (resume)
- **some signals (e.g., SIGTERM) can be caught and handled by the process, allowing it to perform cleanup tasks**
  - user must have the necessary permissions to send a signal to a process
    - In general, sending signals to processes owned by other users may not be allowed
- **essential for controlling processes in Linux environments**

Linux – *kill alteration*

- **incorrect process responses**
  - improperly handled signals drive to incorrect behavior (e.g. incomplete shutdowns)
- **privilege escalation**
  - If modified to bypass permission checks, allow unauthorized signals sending to processes owned by other users
    - critical services manipulation, breach of security policies
- **persistence**
  - Ignored/mishandled signals lead to runaway process (system degradation) and persiostent process (e.g. rootkits can instruct SIGKILL to be ignored)
- **monitoring compromission**
  - signals are used to trigger logging or monitoring actions (e.g., a service may reload its configuration upon receiving SIGHUP)