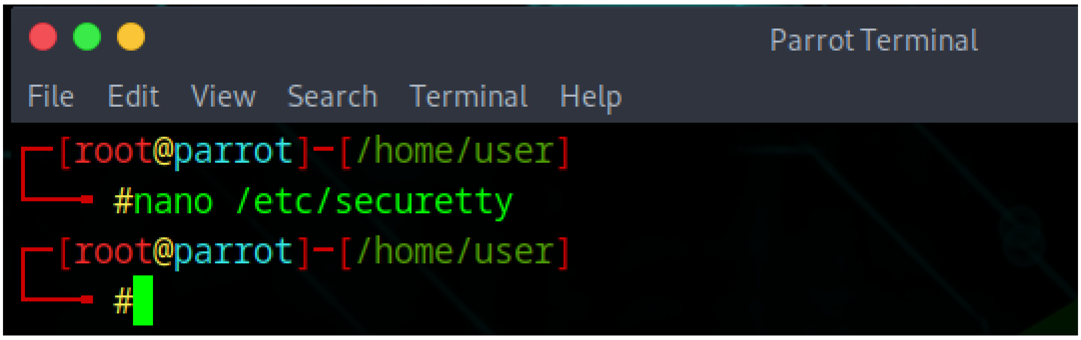
Day 37

Exploitation Analyst

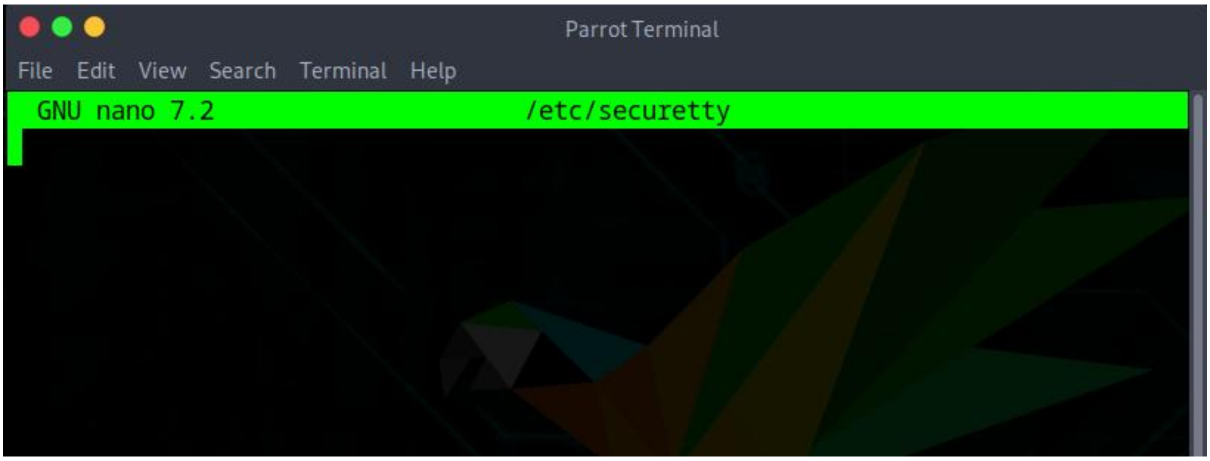
User Management and PAM:

/etc/securetty – Root Login Control

- **Purpose:**
  Defines which TTY (terminal devices) the root user is allowed to log in from.
- **Before Editing:**
  Contains entries like tty1, tty2, etc. → Root can log in locally at those consoles (e.g., physical console or Ctrl+Alt+F1..F6).
- **After Editing (empty file or removed):**
  Root cannot log in directly from any console.
  Root can still be accessed indirectly via su or sudo from another account.
- **Important Notes:**
  - Does not affect SSH logins (that is handled by /etc/ssh/sshd_config).
  - Removing or commenting out entries disables root login on those specific terminals.



```
┌─[root@parrot]─[/home/user]
└──╼ #nano /etc/securetty
┌─[root@parrot]─[/home/user]
└──╼ #
```



```
  GNU nano 7.2                    /etc/securetty
```

Why keeping /etc/securetty empty is a good idea

- **Security Principle:** Prevents root from logging in directly on any local terminal (console).
- **Benefit:**
  - Forces administrators to log in as a normal user first and then elevate with sudo or su.

    - Adds an extra barrier against unauthorized access if someone gains physical access to the machine.
- **Impact:**
    - Reduces attack surface by removing direct root logins.
    - Root account is still usable indirectly via privilege escalation, so system management is not blocked.
- **Conclusion:** Keeping /etc/securetty empty enforces the **principle of least privilege** and improves security by disabling direct root console access.

--The End--