Day 30

Exploitation Analyst

User Management and PAM:

PAM:

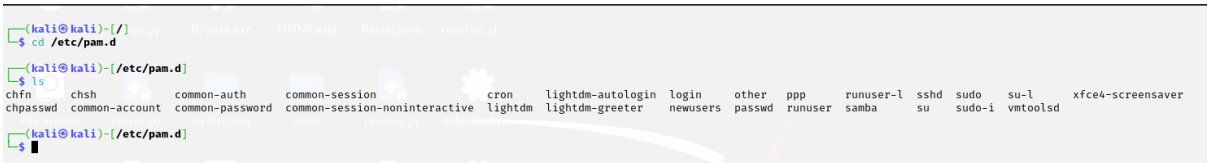
What is PAM?

PAM stands for Pluggable Authentication Modules.

It's a framework used in Linux to manage authentication tasks like login, sudo, SSH, and more. Instead of hardcoding how authentication works, PAM lets the system use configurable modules for things like:

- Username/password checks
- Account lockout after failed attempts
- Two-factor authentication
- Biometric or smart card login

Where to find this? It is located at the /etc/pam.d



What is the core logic of PAM?

In PAM (Pluggable Authentication Modules), the configuration uses a format like <module_type> <control_flag> <module> [options] to define how authentication is handled. The module_type tells when to apply the module (e.g., auth for password checks, session for login activities), the control_flag defines how its result affects the overall decision (required, optional, etc.), and the module is the actual plugin like pam_unix.so. For example, the line auth required pam_unix.so means the system must check the user's password using standard Linux methods, and if it fails, access will be denied after all checks are run. This modular and flexible system lets you build secure login policies with combinations like password checks, delays, or even 2FA.

How PAM Works (Step-by-Step):

1. **Application Requests Authentication**
   Example: You run sudo → PAM is triggered to authenticate you.
2. **PAM Reads Config File**
   It checks the relevant file in /etc/pam.d/ (e.g., /etc/pam.d/sudo) to see what modules to use and in what order.
3. **Executes PAM Modules in Order**
   Each line in the config defines:
   a. What to check (like password, account status)
   b. How to react (via required, requisite, optional, etc.)
   c. What module to use (e.g., pam_unix.so, pam_tally2.so, pam_google_authenticator.so)

4. **Collects Results Based on Control Flags**
    a. If a required module fails → final auth fails (after all modules run)
    b. If a requisite module fails → fails immediately
    c. sufficient or optional modules may be skipped based on logic
5. **Allows or Denies Access**
    Based on combined results, PAM tells the app to allow or deny the request.

Commonly Used PAM Modules:

PAM Module	Type(s)	Description
pam_unix.so	auth, account, password, session	Standard authentication using /etc/passwd and /etc/shadow.
pam_rootok.so	auth	Allows access **only if the user is root** (UID 0).
pam_deny.so	all	Always denies access (used to block actions or enforce policy ends).
pam_permit.so	all	Always permits access (used for testing or allowing fallback).
pam_securetty.so	auth	Restricts root login to secure terminals (defined in /etc/securetty).
pam_tally2.so	auth, account	Tracks failed login attempts and can **lock** accounts.
pam_faillock.so	auth, account	Similar to pam_tally2.so, newer and preferred on modern systems.
pam_env.so	session	Sets user environment variables at login (via /etc/environment).
pam_limits.so	session	Enforces resource limits (CPU, RAM, etc.) from /etc/security/limits.conf.
pam_motd.so	session	Displays the Message of the Day (/etc/motd) at login.
pam_lastlog.so	session	Shows last login info on terminal login.
pam_exec.so	all	Runs custom scripts/binaries during auth (can be risky if misused).
pam_google_authenticator.so	auth	Adds **2FA** using Google Authenticator or TOTP.
pam_wheel.so	auth	Restricts access to users in a specific group (e.g., only sudo users).

--The End--