# *nix Security

## 1. SELinux (Security-Enhanced Linux)

- What It Is
  - **A Linux kernel security module** that provides **Mandatory Access Control (MAC) to enforce strict security policies**.
- Key Features
  - **Labels and Policies**
    - Every file, process, and resource has a **security context** (e.g., user_u:role_r:type_t).
  - **Enforcing Modes**
    - Enforcing: Policies are applied, and violations are blocked.
    - Permissive: Violations are **logged but not blocked** (useful for debugging).
    - Disabled: SELinux is turned off.
  - **Fine-Grained Controls**
    - Restricts processes to only the actions defined in the security policy.
- Use Cases
  - Isolating services (e.g., confining web servers to specific files and ports).
  - Preventing privilege escalation through misconfigured or vulnerable applications.

## 2. Kernel, Userspace, and Permissions

- **Kernel**
  - The **core component** of Unix/Linux responsible for managing hardware and system resources.
  - Provides an interface for user applications to interact with hardware via system calls.
- **Userspace**
  - **Non-kernel processes and applications** running on the system.
  - Includes user applications, libraries, and daemons.
- **Permissions**
  - Unix uses a three-level permission model:
    - **Owner**: The user who owns the file or directory.
    - **Group**: A group of users who can access the file.
    - **Others**: Everyone else.
  - Modes
    - Read (r), Write (w), Execute (x).
    - Permissions are displayed as a 10-character string (e.g., -rw-r--r--).
  - Command: chmod, chown, ls -l.

## 3. MAC vs DAC

| Aspect | Mandatory Access Control (MAC) | Discretionary Access Control (DAC) |
|---|---|---|
| Control | Enforced system-wide by administrators. | Decentralized; owners control permissions. |
| Flexibility | Less flexible; predefined policies. | More flexible but less secure. |
| Example | SELinux, AppArmor | Standard Unix permissions (chmod, chown). |

| Aspect | Mandatory Access Control (MAC) | Discretionary Access Control (DAC) |
| --- | --- | --- |
| Use Case | High-security environments (e.g., servers) | General-purpose systems. |

## 4. /proc

- What It Is
  - **A virtual filesystem that provides information about processes and system resources**.
- Common Directories
  - **/proc/**: Contains details about a specific process.
    - cmdline: Command-line arguments used to start the process.
    - status: Process status and memory usage.
  - /proc/cpuinfo: Information about the CPU.
  - /proc/meminfo: Information about system memory.
  - /proc/net: Network statistics.
- Forensic Uses
  - Monitoring running processes and their behavior.
  - Identifying rogue processes by inspecting cmdline and fd.

## 5. /tmp

- Purpose
  - **A directory for storing temporary files**. It's **world-writable**, meaning any user can create files here.
- Security Concerns
  - **Code Execution**
    - Attackers can save malicious scripts or binaries in /tmp and execute them.
  - **Symbolic Link Attacks**
    - Creating symbolic links in /tmp to sensitive files for privilege escalation.
- Mitigations
  - **Mount /tmp with the noexec option** to prevent execution of binaries
  - Regularly **clean /tmp** to remove potentially harmful files.

```
mount -o remount,noexec /tmp
```

## 6. /shadow

- What It Is
  - A file that **stores hashed passwords for user accounts**.
  - Located at /etc/shadow and **readable only by the root user**.
- Structure
  - Each line corresponds to a user:

```
username:hashed_password:last_change:min_days:max_days:warn_days:inactive_
days:expire
```

- Security Concerns:
  - If compromised, attackers can use tools like **John the Ripper** or **Hashcat** to crack the hashes.
  - Common hash formats:
    - $6$: SHA-512.
    - $5$: SHA-256.
    - $1$: MD5.
- Best Practices:
  - **Use strong password policies** and **hashing algorithms (e.g., SHA-512)**.
  - Limit access to **/etc/shadow**.

## 7. LDAP (Lightweight Directory Access Protocol)

- What It Is
  - **A protocol for accessing and managing directory information**.
  - **Commonly used for authentication and user management in Unix environments**.
- How It Works
  - Centralized management of user credentials and information.
  - Users can authenticate across multiple services (e.g., email, VPN) using a single password.
- LDAP vs. Active Directory
  - **LDAP is a protocol**, while **Active Directory is a Microsoft directory service that uses LDAP**.
  - LDAP is more lightweight and platform-agnostic, making it ideal for Unix systems.
- Security Considerations
  - **Encrypt LDAP traffic using LDAPS or StartTLS**.
  - Implement **access controls** to restrict unauthorized access.

## Summary

| Concept | Details |
|---|---|
| SELinux | Provides MAC, enforcing strict security policies. |
| Kernel/Userspace | Kernel manages resources; userspace includes applications and user-level processes. |
| MAC vs DAC | MAC offers stricter security; DAC provides more flexibility. |
| /proc | Virtual filesystem for process and system information. |
| /tmp | Temporary file storage; vulnerable to code execution without noexec. |
| /shadow | Stores hashed passwords; critical for system security. |
| LDAP | Centralized authentication protocol similar to AD for Unix systems. |

**Unix/Linux systems rely on robust security mechanisms like SELinux, file permissions, and secure configuration of directories like /tmp and /shadow**. By understanding these components and implementing best practices, administrators can build resilient systems that resist common attacks while maintaining operational flexibility.