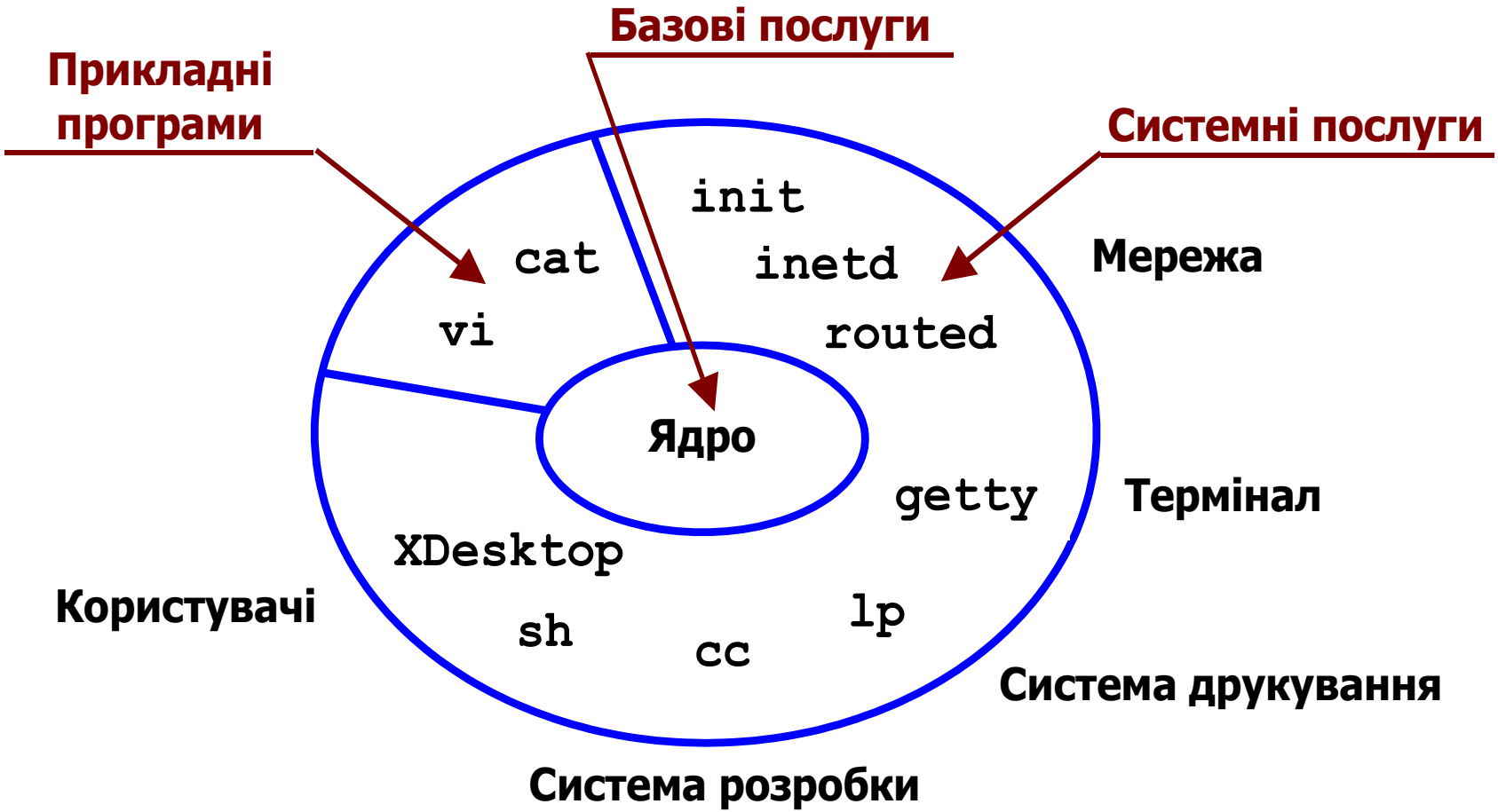
Безпека операційних систем і комп'ютерних мереж

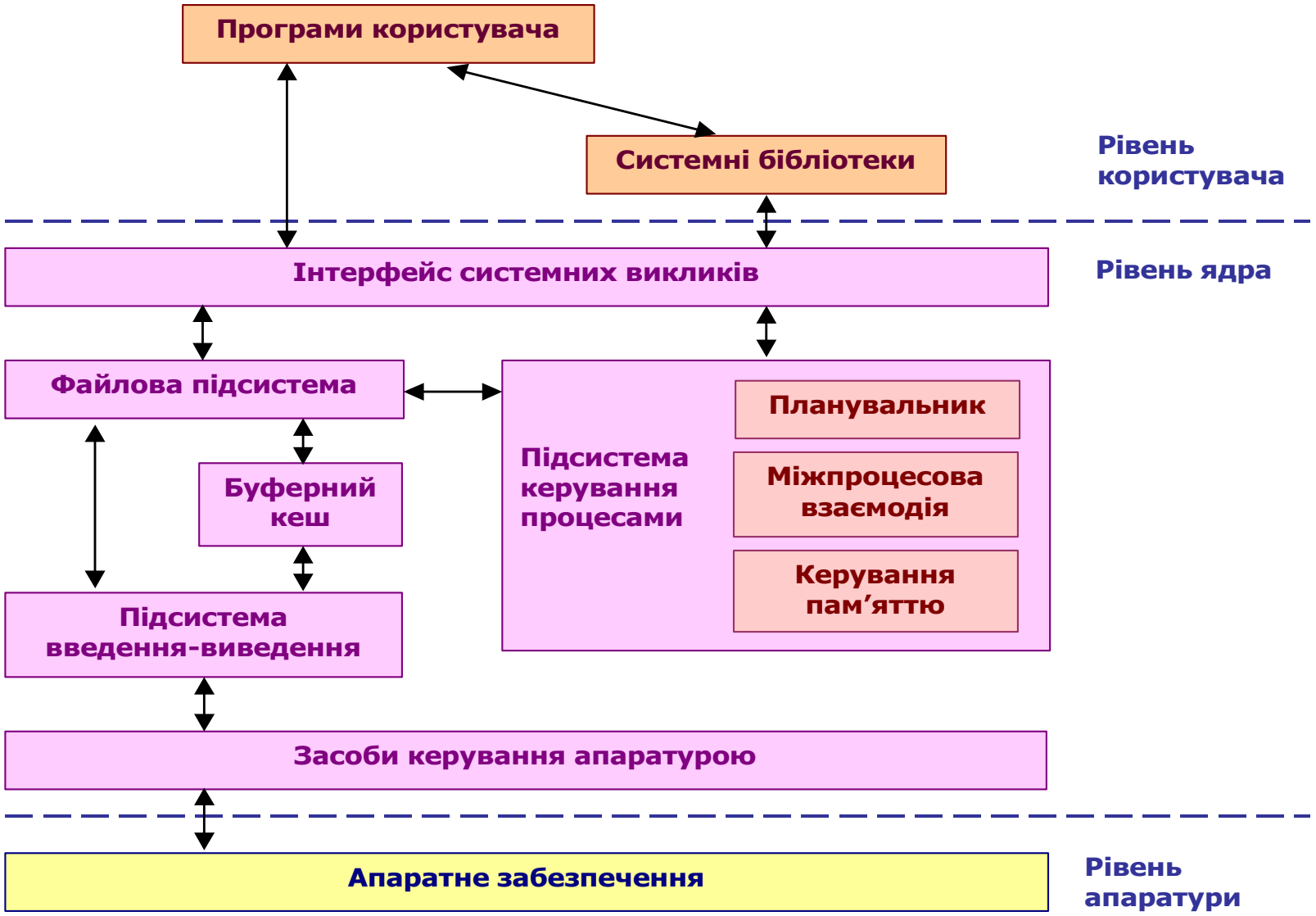
Лекція 9

Засоби захисту UNIX-подібних систем (зокрема, Linux)

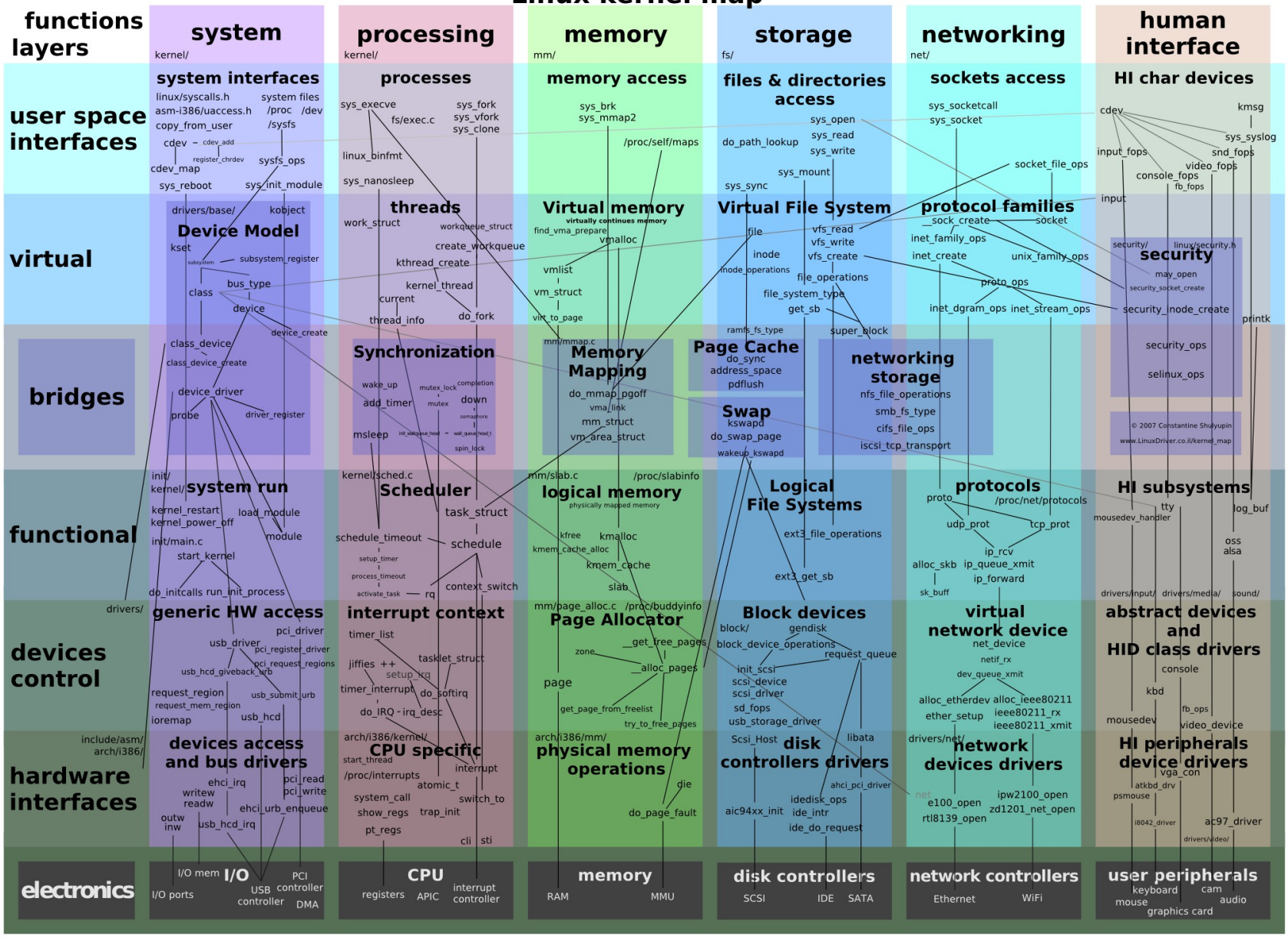
Архітектура системи UNIX



Структура ядра UNIX (Linux)

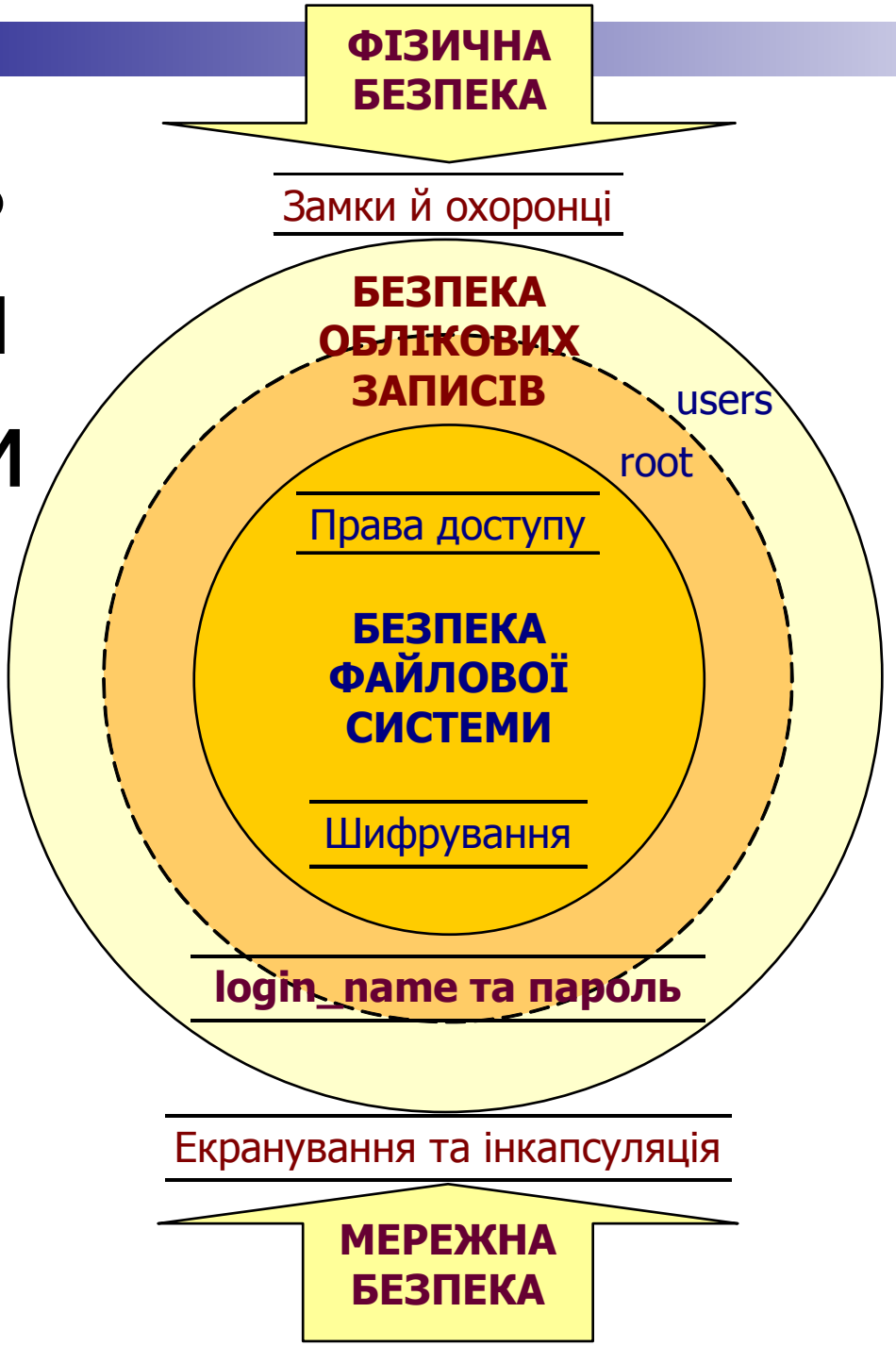


Linux kernel map



Ver 0.6, 1/1/2008

Модель безпеки системи UNIX



Підсистема ідентифікації та автентифікації (традиційна)

- Облікові записи: файл */etc/passwd*

  login_name:*:UID:GID:user_info:home:shell

  iv13:*:1013:1013:Ivan Ivanenko:/usr/home/iv13:/usr/bin/bash
- Хеш-образи паролів (DES, MD5, SHA1, SHA2):

  /etc/shadow (Solaris, Linux, інші системи)

  /etc/master.passwd (BSD)
- Автентифікація: утиліта *login*
- Перехід в інший обліковий запис: утиліта *su*
- Зміна пароля: утиліта *passwd*
- Авторизація: запуск програми, що вказана у файлі */etc/passwd*, від імені користувача
- Додавання користувачів: *adduser* (або *useradd*)
- Редагування облікових записів: *vipw*

Підсистема ідентифікації та автентифікації (сучасна)

- PAM — Pluggable Authentication Modules (рос. - Подключаемые модули аутентификации, ПМА)
  - Механізм PAM поєднує різні низькорівневі схеми автентифікації в API високого рівня, що дозволяє створювати програми, які використовують автентифікацію незалежно від схеми автентифікації, що застосовується
  - Зокрема, сучасні версії утиліт login, passwd, su самостійно з паролями не працюють, а натомість звертаються до PAM
- PAM був розроблений Sun Microsystems у 1995 році
- Реалізований у більшості сучасних UNIX-подібних систем
- Реалізації PAM:
  - Linux-PAM
  - OpenPAM (FreeBSD, NetBSD, Mac OS X)
  - Java™ PAM или JPam

Налаштування PAM

- PAM складається з бібліотек-модулів, які знаходяться у каталозі */lib/security/*
  - Кожний модуль реалізує певний механізм автентифікації
- Для кожної з програм, що використовує PAM, є її специфічний сценарій
  - Ці сценарії розміщені у каталозі */etc/pam.d*
- Ім'я кожного сценарію в цьому каталозі збігається з ім'ям програми, для якої він призначений
  - Наприклад, сценарій для *login* має адресу */etc/pam.d/login*
  - Для деяких модулів є специфічні файли конфігурації, які знаходяться у каталозі */etc/security*
- Якщо немає каталогу */etc/pam.d* (у сучасних системах він є), то налаштування PAM здійснюється за допомогою файлу */etc/pam.conf*

Модулі PAM

- Модулі PAM класифікують за типами. Кожний модуль повинен виконувати функції хоча б одного з чотирьох типів:

  auth — модуль автентифікації, використовується для автентифікації користувачів або створення і вилучення облікових даних

  account — модуль керування обліковими записами, виконує дії, що пов'язані з доступом, терміном придатності облікових даних чи записів, правилами і обмеженнями для паролів тощо

  session — модуль керування сеансами, використовується для створення і завершення сеансів

  password — модуль керування паролями, виконує дії, що пов'язані зі змінами і оновленнями паролів

- PAM забезпечує різні функціональні можливості, такі як автентифікація з однократною реєстрацією, керування доступом та інші

Приклади модулів PAM

- **pam_access** забезпечує керування входом в систему у вигляді служби, що протоколюється, за допомогою імені користувача і домену в залежності від правил, заданих у файлі **/etc/security/access.conf**
- **pam_cracklib** перевіряє паролі на відповідність правилам для паролів
- **pam_env sets/unsets** встановлює і скидає змінні середовища з файлу **/etc/security/pam_env_conf**
- **pam_debug** виконує зневадження PAM
- **pam_deny** блокує модулі PAM
- **pam_echo** виводить повідомлення
- **pam_exec** виконує зовнішню команду
- **pam_ftp** модуль для анонімного доступу
- **pam_localuser** перевіряє наявність ім'я користувача у файлі **/etc/passwd**
- **pam_unix** виконує звичайну автентифікацію на основі пароля з файлу **/etc/passwd**

Приклад /etc/pam.d/login

```
#%PAM-1.0
auth       requisite    /lib/security/pam_unix.so       nullok #set_secrpc
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       required     /lib/security/pam_mail.so
account    required     /lib/security/pam_unix.so
password required     /lib/security/pam_unix.so       strict=false
session    required     /lib/security/pam_unix.so       none # debug or trace
session    required     /lib/security/pam_limits.so
```

Підсистема розмежування доступу

- Суб’єкти доступу – користувачі (групи користувачів)
  - /etc/passwd – первинна група користувача
  - /etc/group – інші групи
- Об’єкти доступу – файли
  - звичайні файли
  - каталоги
  - посилання (link)
  - канали (pipe)
  - спеціальні файли – пристрої
- Методи доступу (закріплені у стандарті POSIX):
  - зчитування (Read)
  - записування (Write)
  - виконання (eXecute)

Опис прав доступу до файлу

			власник			група			всі решта		
SUID	SGID	Sticky	r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1	4	2	1

-rw-rw-r- - 664
drwxr-xr-x 755

- Зміна власника: команда *chown*
- Зміна прав доступу: команда *chmod*

```
% ls –l clients.db
-rw-rw----   1 ivanych  managers  65536 Jul 10   2006 clients.db
% chmod 640 clients.db
% ls –l clients.db
-rw-r-----   1 ivanych  managers  65536 Jul 10   2006 clients.db
%
```

Списки керування доступом (Solaris, Linux)

- Команди: getfacl та setfacl

```
% ls -l clients.db
-rw-rw----   1 ivanych  managers  65536 Jul 10   2006 clients.db
% getfacl clients.db

# file: clients.db
# owner: ivanych
# group: managers
user::rw-
group::rw-              # effective:r--
mask:r--
other:---
```

Списки керування доступом (Solaris, Linux)

- Команди: getfacl та setfacl

```
% setfacl -m user:petrovych:rw-clients.db
% getfacl clients.db

# file: clients.db
# owner: ivanych
# group: managers
user::rw-
user:petrovych:rw-              # effective:rw-
group::rw-              # effective:r--
mask:r--
other:---

% ls -l clients.db
-rw-rw----+  1 ivanych  managers  65536 Jul 10   2006 clients.db
%
```

Суперкористувач в UNIX

- Обліковий запис:
  - login_name root (традиційно, але не принципово)
  - UID 0 (принципово — лише за умови стандартної автентифікації)
  - **У випадку PAM-автентифікації наявність прав root у користувача може бути зовсім не очевидною**
- Обмеження доступу до системи:

  файл /etc/defaults/login (Solaris)

  `CONSOLE=/dev/console` – лише локальний доступ root

  `CONSOLE=` – root взагалі не може інтерактивно ввійти в систему
- Тимчасове отримання прав суперкористувача:
  - su – необхідно знати пароль root
  - sudo – для підтвердження необхідно ввести свій пароль, дозволені дії налаштовуються у файлі /etc/sudoers
  - **Налаштування PAM можуть змінити поведінку su і sudo щодо автентифікації**

Захист від використання вразливостей коду — ASLR

- Linux застосовує технологію випадкового розташування програми в адресному просторі (address space layout randomization, ASLR)
  - Успіх атак, подібних до переповнення стеку, заснований на знанні зловмисником взаємного розташування різних модулів програми, яку атакують
  - Тому випадкова перестановка цих модулів (яка, зрозуміло, повинна не порушувати цілісність і функціональність самої програми) зробить більшість таких атак марними
    - У найгіршому випадку зловмисник зможе викликати аварійне завершення процесу, але не зможе отримати контроль над ним
    - Крім того, «падіння» процесу внаслідок атаки, ймовірно, притягне більше уваги системного адміністратора, ніж успішне і непомітне впровадження коду
- Технологія ASLR впроваджена у ядро Linux, починаючи з версії 2.6.12 (2005 рік — значно раніше, ніж у Windows)
- Для рандомізованого розміщення у пам'яті образу виконуваного файлу він має бути скомпільований у режимі Position-independent executable

Основні недоліки традиційної моделі безпеки Linux і Unix

- Недостатня вибірковість системи розмежування доступу.
  - Дуже часто права доступу надаються за принципом «або усім, або лише одному». Це стосується як користувачів, так і інших агентів системи.
  - Внаслідок цього користувач часто змушений приймати права root (або через suid-біт, або командою su), з усіма повноваженнями суперкористувача, для виконання завдань, що насправді вимагають значно менших прав
- Надання усіх можливих прав і привілеїв одному обліковому запису суперкористувача
  - Користувач, що несанкціоновано отримав доступ до прав root, одержує повний контроль над системою
- Відсутність розмежування прав доступу до деяких важливих об’єктів системи
  - Наприклад, з будь-якого сеансу користувача можна запустити сервер TCP/IP (якщо тільки порт вже не зайнятий іншим застосуванням)
  - Для більшості користувачів така функціональність є зайвою і може мати наслідком порушення безпеки