Тема 7

Безпека і захист

План

1. Базові поняття і принципи безпеки ОС
2. Автентифікація та авторизація
3. Аудит

1. Базові поняття і принципи безпеки ОС

Терміни “безпека” і “захист” не завжди розділяють.

[Silberchatz, Galvin, Gagne, 2018]

Безпека ОС (OS security) - міра впевненості, що цілісність системи і даних всередині неї буде збережено.

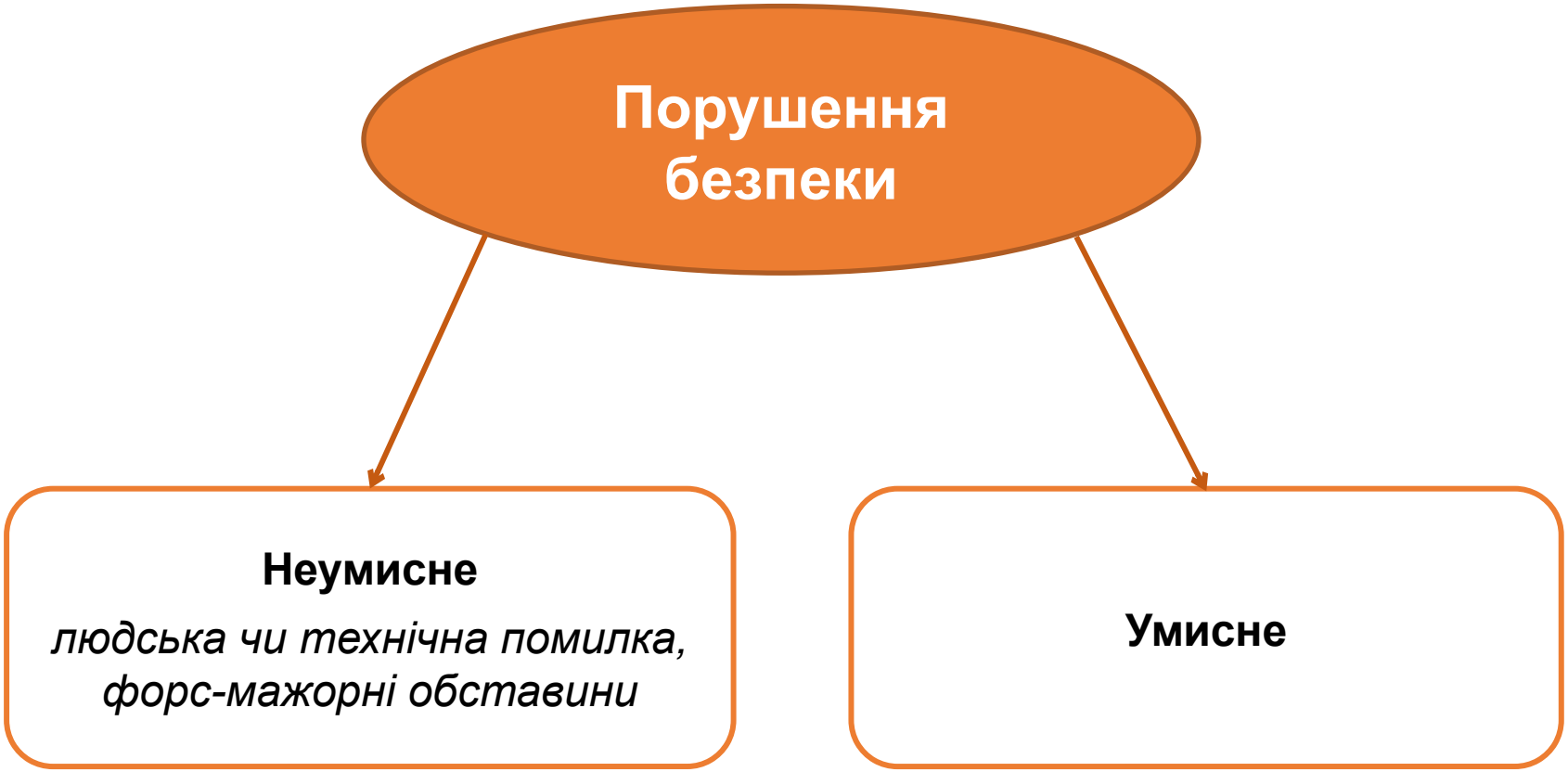
Захист ОС (OS protection) - механізм контролю доступу програм, процесів або користувачів до ресурсів комп'ютерної системи.

Контролю - за допомогою чого?

Цей механізм має надавати:

- засоби для визначення заходів контролю;
- засоби для примусового застосування цих заходів

1. Базові поняття і принципи безпеки ОС



1. Базові поняття і принципи безпеки ОС

Ресурси ОС
та їхні вразливості

Апаратне забезпечення
фізична втрата чи пошкодження

Програмне забезпечення
шкідливе ПЗ, зміна поведінки нешкідливого ПЗ

Дані
втрата, пошкодження, підміна, витік

Комунікації
виступають у ролі каналу для несанкціонованого доступу

1. Базові поняття і принципи безпеки ОС

? Що ОС може зробити для безпеки та захисту?

- *Зупинити несанкціонований доступ (автентифікація, розмежування доступу, шифрування)*
- *Вести аудит подій, які відбуваються у системі*
- *Запобігти втраті даних*
- *Попереджати про потенційно небезпечне ПЗ*
- *Поважати приватність користувача*
- *Протидіяти вірусам*
- *...*

1. Базові поняття і принципи безпеки ОС

? Є безпечна ОС, а є надійна ОС. В чому різниця?

Безпечна - захищена від загроз.

Надійна - коректно працює протягом передбачуваного періоду часу.

? Настільки строгим є твердження, що ОС безпечна?

What about an unhackable kernel?

Reminds me of that unsinkable ship.

(З форуму)

1. Базові поняття і принципи безпеки ОС

Заходи безпеки впроваджуються на чотирьох рівнях.

1. Фізичний рівень

Місце фізичного розміщення комп'ютерних систем має бути фізично захищене від збройного й таємного проникнення зловмисників.

2. Людський рівень

Протидія передачі авторизованими користувачами своїх реквізитів для доступу до системи стороннім особам, свідомого чи несвідомого.

3. Рівень операційної системи

ОС має захищати себе від порушень системи безпеки (неумисних та умисних).

4. Мережний рівень

Забезпечення працездатності мережних з'єднань, захист даних, які передаються по мережі, та систем, під'єднаних до мережі.

1. Базові поняття і принципи безпеки ОС

Механізми безпеки vs *політики безпеки*

Політики задають, ***що саме*** буде зроблено.
Механізми визначають, ***як саме*** щось буде зроблено.

Політики можуть відрізнятися залежно від обставин і змінюватися з часом.
Механізми в основі політики мають лишатися незмінними.

Політики можуть бути:

- закладені на етапі проектування ОС;
- встановлені у процесі адміністрування ОС;
- визначені окремими користувачами для захисту їхніх власних файлів та програм.

1. Базові поняття і принципи безпеки ОС

Основні вимоги до механізмів безпеки ОС

Автентифікація *(authentication):* кожна дія виконується певним суб'єктом, ідентичність якого встановлено системою.

Авторизація *(authorization):* система регулює, які дії дозволені тим чи іншим суб'єктам.

Аудит (*accounting*): система документує події, пов'язані з безпекою.

Застосовні не лише до ОС (напр., комп'ютерні мережі - AAA).

1. Базові поняття і принципи безпеки ОС

Частина цих принципів втілена у більшості сучасних систем, частина - лишається предметом дискусії

Принципи розробки систем безпеки і захисту

(за роботою Saltzer and Schroeder, 1975 + пізніші доповнення)

1. Механізми безпеки мають бути настільки простими і невеликими, наскільки це можливо.

2. Орієнтація передусім на дозволи, а не на заборони.

3. Кожний доступ до кожного об'єкта має перевірятися на предмет наявності повноважень.

4. Будова механізмів безпеки має бути відкритою, а не засекреченою.

5. Там, де це доцільно, для посилення механізми захисту варто використовувати ті, де передбачено доступ за двома ключами.

6. **Принцип мінімальних повноважень** *(principle of least privilege)*: кожній програмі чи користувачу у системі має бути надато мінімальний набір повноважень, необхідний для виконання поставлених перед ними завдань.

1. Базові поняття і принципи безпеки ОС

Принципи розробки систем безпеки і захисту

(Saltzer and Schroeder, 1975 + пізніші доповнення)

7. Психологічна доступність: механізми захисту повинні мати зручний для використання людиною інтерфейс, щоб користувачі могли регулярно й коректно використовувати ці механізми.

8. Порівнювати вартість обходу захисних механізмів з наявними в потенційного зловмисника ресурсами.

9. Надійні записи інформації про випадки, коли систему безпеки було скомпроментовано, можуть допомогти у виробленні надійніших механізмів.

1. Базові поняття і принципи безпеки ОС

Security vs *Accesibility*

Безпека проти *Доступності*

Тези:

- Один бік: кроки задля доступності можуть шкодити безпеці
- Інший бік: доступ до ресурсів повинні мати всі, для кого вони призначені

2. Автентифікація та авторизація

Типи автентифікації:

- **локальна**
  (перевірка легітимності входу - на комп'ютері, за яким працює користувач);
- **мережна**
  (перевірка легітимності входу - на віддаленому комп'ютері, дані передаються по мережі).

Форми автентифікації:

- автентифікація на основі паролю;
- автентифікація на основі фізичного об'єкту;
- автентифікація на основі біометричних параметрів.

2. Автентифікація та авторизація

Основні суб'єкти розмежування доступу - *користувачі* та *групи*.

Типові відомості облікового запису:

- **логін** користувача
- **ідентифікатор** користувача

  у Linux - **UID** (**U**ser **Id**entifier), у Windows - **SID** (**S**ecurity **Id**entifier)
- відомості про **пароль**

  сам пароль у зашифрованому вигляді або спеціальні відомості, які дозволяють перевірити правильність введеного паролю
- відомості про належність користувача до **груп**
- **обмеження на вхід** для даного користувача

  (термін дії облікового запису, паролю, дні чи години, в які користувачу дозволено входити до системи)
- шлях до **домашнього каталогу** користувача
- шлях до типового для користувача **командного інтерпретатора**

2. Автентифікація та авторизація

? Навіщо потрібні групи?

Керувати доступом до ресурсів за певним зразком.

Ідентифікатори груп:

- у Linux - **GID** (**G**roup **Id**entifier)
- у Windows - **SID** (**S**ecurity **Id**entifier) - тобто як і у користувачів

2. Автентифікація та авторизація

Паролі

? Яким має бути пароль?

- Бути довгим...
- ...Але не занадто довгим (користувачі будуть записувати).
- Містити літери, цифри, інші символи, у тому числі різного регістру.
- Мати обмежений термін існування.

На що варто звертати увагу:

- Якщо користувачі самі вибирають собі паролі, ці паролі мають бути надійними.
- Користувачі можуть записувати паролі.
- Рідко використовуваним обліковкам - особливу увагу.
- Підібрати оптимальний термін існування паролів.

2. Автентифікація та авторизація

Користувачі у Linux

Список користувачів зберігається у файлі

/etc/passwd

Загальний синтаксис окремого запису в */etc/passwd*:

ім'я_користувача:*пароль*:*UID*:*GID_основної_групи*:
відомості_про_користувача:*домашній_каталог*:
командний_інтерпретатор_за_замовчуванням

Поле *пароль* історичне, насправді зашифровані паролі зберігаються у файлі */etc/shadow*.

Приклади:

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

olena:x:1000:1000:Olena,,,:/home/olena:/bin/bash

2. Автентифікація та авторизація

Групи у Linux

Список груп зберігається у файлі

/etc/group

Загальний синтаксис окремого запису в */etc/group*:

ім'я_групи:*пароль*:*GID*:*перелік_членів_групи*

Поле *пароль* за замовчуванням порожнє (*x*), але може містити зашифрований пароль.

Приклади:

```
root:x:0:

daemon:x:1:

adm:x:4:syslog,olena

olena:x:1000:

user1:x:1001:
```

2. Автентифікація та авторизація

Основні підходи до розмежування доступу

Розмежування доступу передбачає існування певних **правил**, які пов'язували б:

- суб'єкта, що пройшов автентифікацію,
- об'єкт (тобто ресурс),
- дії, виконувані з об'єктом (ресурсом)

й однозначно давали б відповідь на питання:
чи дозволена деяка дія деякому суб'єкту щодо деякого ресурсу.

2. Автентифікація та авторизація

Основні підходи до розмежування доступу

Загальний випадок - **матриця доступу** (*access matrix*).

object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

Насправді значно більша

Але в реальності зберігати це у вигляді матриці незручно.

Як інакше:

- можна прив'язати відомості про розмежування доступу до *суб'єктів*,
- а можна до *об'єктів* (тобто до ресурсів).

2. Автентифікація та авторизація

Основні підходи до розмежування доступу

Тому частіше використовуються два основні підходи.

Списки керування доступом (ACL - Access Control List)

для кожного **ресурсу** задано список суб'єктів, яким дозволено використовувати цей ресурс.

Де: у більшості ОС.

Зокрема: у Windows (ACL - так і називаються),
у Linux (рядки повноважень - скорочений варіант ACL,
також можливе застосування повного варіанту ACL).

Переліки можливостей (Capabilities List, C-list)

для кожного **суб'єкта** задано перелік ресурсів, котрі йому дозволено використовувати.

Де: наприклад, у Mach, Hydra, Cambridge CAP System, елементи - у Windows (привілеї та права доступу облікових записів).

2. Автентифікація та авторизація

Розмежування доступу у Linux

Ролі:

- **u** - користувач-власник (user owner)
- **g** - група-власник (group owner)
- **o** - решта користувачів (others)
- **a** - всі користувачі (all)

Дозволи:

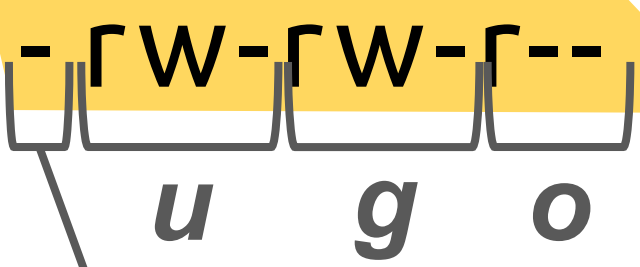
- **r** - читання (**r**ead)
- **w** - запис (**w**rite)
- **x** - виконання (e**x**ecute)
- **-** - дозвіл відсутній

2. Автентифікація та авторизація

Розмежування доступу у Linux

```
olena@ubuntu:~$ ls -l file1
-rw-rw-r-- 1 olena olena 0 лис 27 04:50 file1
```

рядок повноважень ***користувач-власник*** ***група-власник***



Перший символ рядка повноважень вказує на ***тип елемента***.
Найпоширеніші типи елементів:

- **-** - файл
- **d** - каталог *(**d**irectory)*
- **l** - символьне посилання *(symbolic **l**ink)*

2. Автентифікація та авторизація

Механізм *sudo* у Linux

Джерело картинки: *Nixcraft*

2. Автентифікація та авторизація

Механізм *sudo* у Linux

Традиційний обліковий запис адміністратора у Unix/Linux - *root*.

Обліковка ***root*** має необмежений доступ. Решта звичайних користувацьких обліковок традиційно має суттєво вужчі повноваження.

У сучасних Linux часто використовується механізм ***sudo****:*

- користувачі входять під обмеженими обліковими записами
- частина цих користувачів має право використовувати команду *sudo* та її аналоги

Обліковий запис *root* може бути відключений (часто, але не обов'язково).

Користувачі, яким дозволяється використовувати *sudo*, зазвичай є членами однойменної **групи *sudo***. Не додавати у групу ***sudo*** без необхідності!

Інші суб'єкти, яким також можна застосовувати *sudo*, вказуються у файлі /etc/sudoers (не рекомендовано редагувати вручну!)

Випадки невдалого використання sudo належать до подій, які фіксуються у журналах аудиту (наприклад, `/var/log/auth.log`).

2. Автентифікація та авторизація

Розмежування доступу у Windows

Кожному об'єкту відповідає *дискриптор захисту* (*security descriptor*).

Деякі елементи дискриптора захисту:

- SID власника об'єкта
- список керування доступом (**AC**

складається з **ACE** (***Access Control Entry***)

- тип ACE (дозвільний чи заборонний)
- SID користувача або групи
- набір прав доступу, які надаються чи забираються

2. Автентифікація та авторизація

Дозволи на доступ (*access permissions*) у Windows

Для файлів	Для папок
Читання (у т.ч. атрибутів)	**Читання** (переглядати вкладені папки і файли та їхні атрибути)
Запис (змінювати файл та його атрибути)	**Запис** (поміщати всередину нові файли та підпапки, змінювати атрибути)
–	**Список вмісту папки** (переглядати імена файлів та вкладених папок даної папки)
Читання і виконання (читати і запускати виконуваний файл)	**Читання і виконання** (отримувати доступ до файлів та вкладених папок, навіть якщо немає доступу до самої папки, а також див. *Читання* + *Список вмісту папки*)
Внесення змін (змінювати та видаляти файл + див. *Читання та виконання* + *Запис*)	**Внесення змін** (видаляти папку + див. *Читання* + *Читання та виконання*)
Повний доступ (все + ставати власником і змінювати дозволи)	**Повний доступ** (все + ставати власником і змінювати дозволи)
Особливі дозволи	**Особливі дозволи**

2. Автентифікація та авторизація

Дозволи на доступ (*access permissions*) у Windows: *особливі дозволи*

Для файлів	Для папок
Виконання файлу	Огляд папки
Читання даних	Вміст папки
Читання атрибутів	
Читання додаткових атрибутів	
Запис даних	Створення файлів
Дозапис даних	Створення папок
Запис атрибутів	
Запис додаткових атрибутів	
	Видалення підпапок і файлів
Видалення	
Читання дозволів	
Зміна дозволів	
Зміна власника	
Синхронізація	

3. Аудит

Аудит - документована перевірка роботи системи.

Політика аудиту - перелік подій, повідомлення про які підлягають документуванню.

Такі повідомлення називаються ***повідомленнями аудиту***.

Повідомлення найчастіше фіксуються у ***журналах аудиту*** (***logs***). За це відповідає спеціальний фоновий процес.

3. Аудит

? Для чого потрібні системні журнали?

Щоб фіксувати повідомлення про зміни в системі. Їх можна переглядати й аналізувати:

- *коли щось іде не так;*
- *задля профілактики (поки все так).*

Терміни: log data, log messages, log files (“logs”).

Можливі ситуації:

- Процес використовує стандартні системні процеси для аудиту.
- Процес здійснює власний аудит.

Логи також є стандартні, а є індивідуальні, створені окремими програмами.

3. Аудит

Linux

Назва журналу: ***системний журнал*** (*system log*).

Стандартні процеси аудиту у Linux (один із варіантів):

- *syslogd* + *klogd*
- *rsyslog*
- *journald* - також і двійкові файли

Стандартний каталог аудиту у Linux - ***/var/log*** (можуть бути й інші).

Windows

Назва журналу: ***журнал подій*** (*event log*).

Які події фіксуватимуться у журналі, визначається ***локальною політикою безпеки***.

Служба, відповідальна за ведення журналу, називається ***eventlog***.

Журнали подій зазвичай зберігаються у **C:\WINDOWS\system32\config**

Для самостійного читання

1. **[*Silberschatz, Galvin, Gagne, 2018*]** Chapters 14-15.
2. **[*Stollings, 2017*]** Chapters 15.
3. **[Tanenbaum*, 2014*]** Chapter 9.
4. **[*Шеховцов, 2009*]** Розділ 18.