

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ**

ЕЛЕКТРОННИЙ НАВЧАЛЬНО-НАОЧНИЙ ПОСІБНИК

**СХЕМОКОНСПЕКТ ЛЕКЦІЙ
з навчальної дисципліни
«БЕЗПЕКА ТА ЗАХИСТ ОПЕРАЦІЙНИХ СИСТЕМ»
для студентів всіх форм навчання
галузі знань 12 Інформаційні технології**

Луцьк, 2024

Схемоконспект лекцій з навчальної дисципліни «Безпека та захист операційних систем» для студентів всіх форм навчання галузі знань 12 Інформаційні технології : навчально-наочний посібник / уклад. Я.Ю. Дорогий. – Луцьк : ДонНТУ, 2024. – 896 с.

Навчальна дисципліна «Безпека та захист операційних систем» є актуальною і затребуваною в контексті підготовки сучасних фахівців у галузі інформаційних технологій та кібербезпеки та з огляду на формування когнітивних, афективних та моторних компетентностей в контексті вивчення та розуміння принципів організації та реалізації безпеки та захисту операційних систем.

Пропонований студентам для опанування схемоконспект лекцій з цієї дисципліни має за мету забезпечити їм певну допомогу під час вивчення ними ключових положень базових тем курсу. Схематична форма конспекту акцентує увагу студентів на закріпленні основних фахових знань, формуванні відповідних умінь і в цілому отриманні ними кваліфікаційних компетентностей.

Укладач: _____ Я. Ю. Дорогий, доцент, д. т. н., професор кафедри ПМІ

Рецензент: _____ В. В. Цуркан, доцент, к. т. н., доцент спеціальної кафедри №5 ІСЗЗІ КПІ ім. Ігоря Сікорського,

Відповідальний за випуск: Н. О. Маслова, доцент, к. т. н., зав. каф. ПМІ

Затверджено навчально-методичним відділом ДонНТУ, протокол № 5 від 16.01.2024 р.

Розглянуто на засіданні кафедри прикладної математики і інформатики, протокол № 13 від 27.12.2023 р.

ЗМІСТ

ВСТУП		6
Тема 1.	Мета, задачі, зміст курсу. Основні визначення.	
	Лекція 1. Вступ до дисципліни	7
Тема 2.	Модель загроз для операційної системи.	
	Лекція 2. Модель загроз.	29
Тема 3.	Розроблення захищених операційних систем.	
	Лекція 3. Розроблення захищених операційних систем.	64
Тема 4.	Стандарти оцінювання захищеності операційних систем.	
	Лекція 4. Стандарти оцінювання захищеності операційних систем.	98
Тема 5.	Тема 5. Нормативно-правове регулювання в Україні.	
	Лекція 5. Нормативно-правове регулювання в Україні.	131
Тема 6.	Стандарт ISO/IEC 15408.	
	Лекція 6. Стандарт ISO/IEC 15408.	159

Тема 7.	Засоби захисту ОС Windows.	
	Лекція 7. Модель загроз Windows. Компоненти K33 Windows.	190
	Лекція 8. Криптографічна підтримка та захист даних користувача у Windows	219
	Лекція 9. Ідентифікація та автентифікація. Захист функцій K33.	265
	Лекція 10. Технологія Active Directory.	309
	Лекція 11. Групові політики Windows.	359
	Лекція 12. Аудит у Windows.	414
Тема 8.	Засоби захисту ОС Linux.	
	Лекція 13. Модель загроз Linux.	438
	Лекція 14. Архітектура ПЗ Linux.	465
	Лекція 15. Функціональність безпеки Linux.	499
	Лекція 16. Протокол LDAP.	545
	Лекція 17. Основи SELinux	597
	Лекція 18. Основи AppArmor.	647

	Лекція 19. Аудит в Linux.	681
Тема 9.	Архітектура безпеки і механізми захисту Android.	
	Лекція 20. Механізми захисту Android.	719
	Лекція 21. Безпека ОС Android.	749
Тема 10.	Архітектура безпеки і механізми захисту MacOS.	
	Лекція 22. Механізми захисту MacOS.	786
	Лекція 23. Безпека ОС MacOS.	814
	Лекція 24. Налаштування безпеки в MacOS.	849
	СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ДЛЯ САМОПІДГОТОВКИ	888

У сучасних умовах, що характеризуються високою мінливістю та невизначеністю, функціонування держави та суб'єктів господарювання пов'язане з підвищеним ризиком. Це зумовлено виникненням та діями різноманітних загроз як у зовнішньому, так і у внутрішньому середовищі.

Негативні наслідки реалізації загроз можуть бути руйнівними, тому їх нейтралізація є актуальним завданням. Для цього необхідно своєчасно ідентифікувати загрози, розробити та реалізувати ефективні заходи захисту.

Одним із важливих елементів забезпечення безпеки є захист операційних систем (ОС). ОС є базовим елементом критично інформаційної інфраструктури, від надійності та безпеки якої залежить ефективність діяльності в інформаційних просторах держави та суб'єктів господарювання.

Курс «Безпека та захист операційних систем» спрямований на формування у студентів знань та навичок, необхідних для забезпечення безпеки та захисту ОС.

Для формування практичних навичок студентам пропонується виконання лабораторних робіт та самостійних завдань.

Стисле, але вичерпне подання матеріалу курсу сприятиме формуванню у студентів необхідних знань, умінь та компетентностей у сфері захисту ОС.

Донецький національний технічний університет



Тема 1. Мета, задачі, зміст курсу. Основні визначення.

Лекція 1. Вступ до дисципліни

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

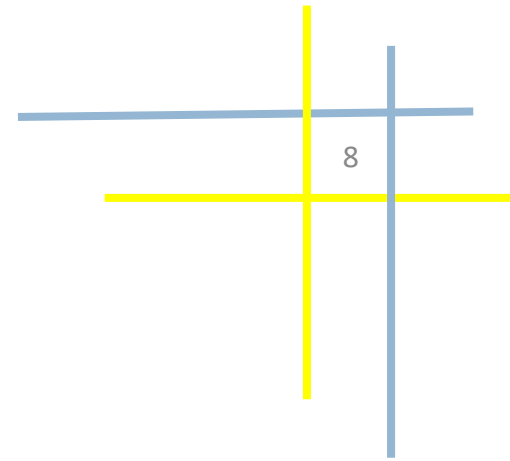
Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета курсу
- Основна термінологія
- Структура курсу





Мета курсу

- формування у студентів навичок аналізування проблем і загроз безпеці програмного забезпечення, які виникають під час обробки інформації в інформаційно-комунікаційних системах
- надання студентам здатностей до обґрунтованого вибору засобів захисту для операційних систем відповідно до вимог національних та міжнародних стандартів
- розвиток умінь налаштування засобів захисту в операційних системах та оцінювання рівня захищеності інформації відповідно до встановлених критеріїв



Автоматизована система (АС)

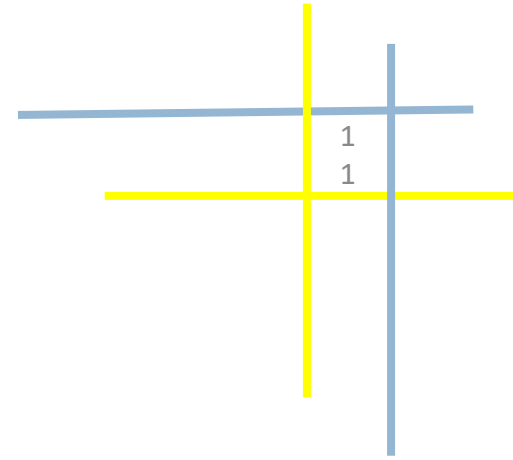
- це організаційно-технічна система, що реалізує інформаційну технологію і поєднує в собі:

- ✓ Обчислювальну систему
- ✓ Фізичне середовище
- ✓ Персонал
- ✓ Інформацію





Інформаційно-комунікаційна система (ІКС)



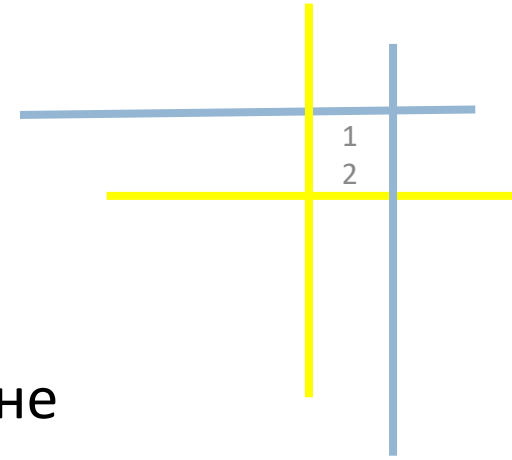
До ІКС відносяться:

- Інформаційна система (ІС) - організаційно-технічна система, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів
- інформаційно-комунікаційна система (ІКС) - сукупність інформаційних та електронних комунікаційних систем, які у процесі обробки інформації діють як єдине ціле

Визначення наведено відповідно до ЗУ «Про захист інформації в інформаційно-комунікаційних системах»



Інформація

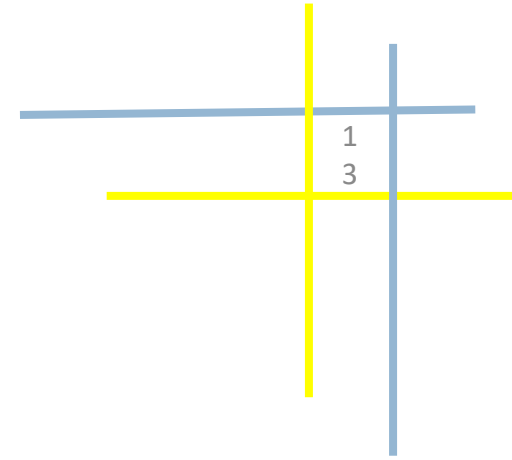


Відповідно до ЗУ "Про інформацію" термін інформація має наступне визначення:

- **це будь-які відомості та/або дані, які можуть бути збережені на матеріальних носіях або відображені в електронному вигляді.**



Властивості Інформації



☐ Конфіденційність

- лише уповноважені користувачі можуть ознайомитись з інформацією

☐ Цілісність

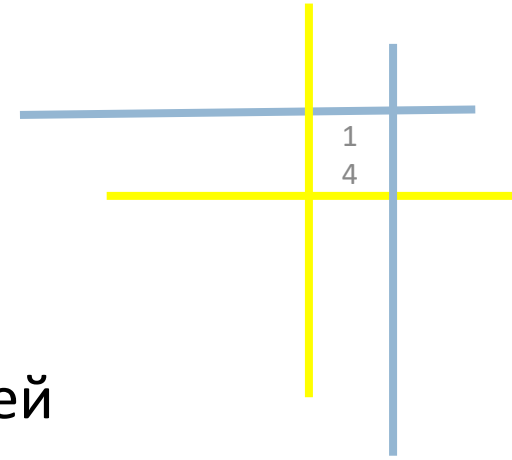
- лише уповноважені користувачі можуть модифікувати інформацію

☐ Доступність

- уповноважені користувачі можуть отримати доступ до інформації, не очікуючи довше за заданий (малий) час



Безпека інформації



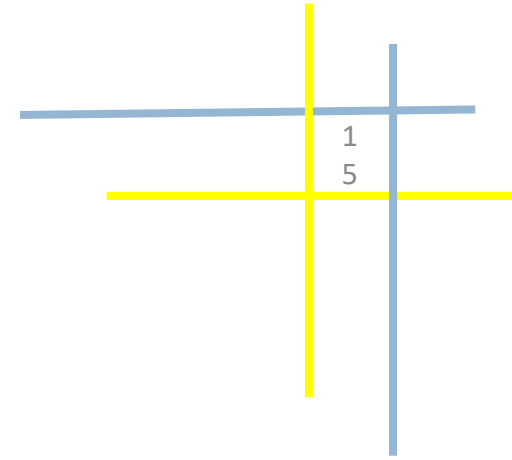
- стан інформації, в якому забезпечується збереження властивостей інформації, визначених політикою безпеки

Відповідно до ЗУ «Про засади інформаційної безпеки України» термін «інформаційна безпека» наступний:

- стан захищеності життєво важливих інтересів людини і громадянина, суспільства і держави, при якому запобігається завдання шкоди через неповноту, несвоєчасність та недостовірність поширюваної інформації, порушення цілісності та доступності інформації, несанкціонований обіг інформації з обмеженим доступом, а також через негативний інформаційно-психологічний вплив та умисне спричинення негативних наслідків застосування інформаційних технологій



Загроза, атака, вразливість, порушник



❑ **Загроза**

- *будь-які обставини або події, що можуть бути причиною порушення політики безпеки інформації та (або) нанесення збитків ІКС*

❑ **Атака**

- *спроба реалізації загрози*

❑ **Вразливість системи**

- *нездатність системи протистояти реалізації певної загрози або сукупності загроз*

❑ **Порушник, зловмисник**

- *фізична особа, яка порушує політику безпеки (навмисно або ні)*



Моделі

16

☐ **Модель [політики] безпеки**

- Абстрактний формалізований або неформалізований опис політики безпеки

☐ **Модель загроз**

- Абстрактний формалізований або неформалізований опис методів і засобів здійснення загроз

☐ **Модель порушника**

- Абстрактний формалізований або неформалізований опис порушника



Захищена комп'ютерна система

17

Захищена комп'ютерна система (англ. — trusted computer system) — комп'ютерна система, що здатна забезпечувати захист оброблюваної інформації від визначених загроз.



Спостережність

18

Спостережність (англ. — *accountability*) — властивість ІКС, що дає змогу фіксувати діяльність користувачів і процесів, використання пасивних об'єктів, а також однозначно встановлювати ідентифікатори причетних до певних подій користувачів і процесів із метою запобігання порушенню політики безпеки і (або) забезпеченню відповідальності за певні дії.



Захист інформації в ІКС

19

Під захистом інформації в ІКС розуміють діяльність, спрямовану на забезпечення безпеки оброблюваної в ІКС інформації та ІКС в цілому, що дає змогу запобігти реалізації загроз або унеможливити її, та зменшити ймовірність завдання збитків від реалізації загроз.

Захист інформації в ІКС полягає у створенні системи технічних (інженерних, програмно-апаратних) та нетехнічних (правових, організаційних) заходів та в підтримці її працездатності.

Система таких заходів – комплексна система захисту інформації (КСЗІ)!



Комплекс засобів захисту (КЗЗ)

Під **КЗЗ** розуміють сукупність програмно-апаратних засобів, що забезпечують реалізацію політики безпеки інформації. КЗЗ є складовою обчислювальної системи (див. рис. на сл. 3).



Об'єкти системи (ОС)

21

ОС – це елемент ресурсів обчислювальної системи, який знаходиться під керуванням КЗЗ і характеризується визначеними атрибутами й поведженням.

Розрізняють наступні види об'єктів:

- ✓ Пасивні
- ✓ Об'єкти-користувачі
- ✓ Об'єкти-процеси



Доступ

22

Доступ – взаємодія двох об'єктів обчислювальної системи, коли один із них (той, що здійснює доступ) виконує дії над іншим (тим, до якого здійснюється доступ). Результатом доступу є змінення стану системи або утворення інформаційного потоку.

Правила розмежування доступу – складова політики безпеки, що регламентує правила доступу користувачів і процесів до пасивних об'єктів.



Розпізнавання об'єктів

23

Ідентифікація – процес розпізнавання об'єктів за їх мітками або ідентифікаторами.

Ідентифікатор – унікальний атрибут об'єкта, який дозволяє вирізнити його з поміж інших об'єктів.

Ідентифікація – процес присвоєння ідентифікаторів об'єктам.



Автентифікація, авторизація

24

Автентифікація – перевірка запропонованого ідентифікатора на відповідність об'єкту, пересвідчення в його справжності.

Авторизація – процедура надання користувачу визначених повноважень у системі.



Відмови

25

Відмова від авторства – заперечення причетності до створення або передавання якого-небудь документа чи повідомлення.

Відмова від одержання – заперечення причетності до отримання якого-небудь документа чи повідомлення.



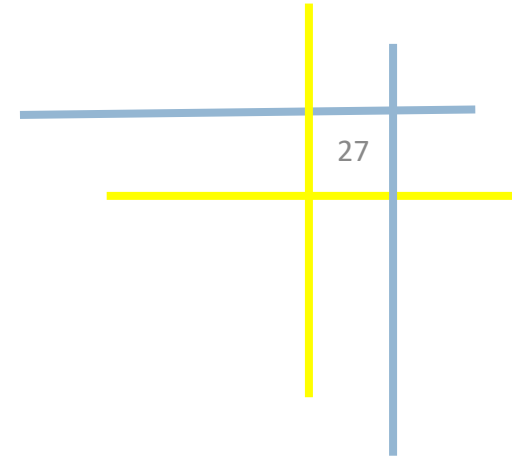
Адекватність

26

Адекватність – показник реального гарантованого рівня безпеки, що відображає ступінь ефективності та надійності реалізованих засобів захисту та їхніх відповідностей поставленим задачам.



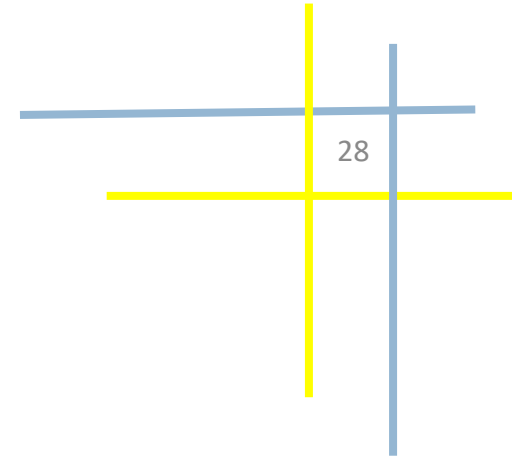
Структура курсу



- 1) Модель загроз для ОС
- 2) Концепція розробки захищених ОС
- 3) Комплекс засобів захисту ОС
- 4) Стандарти оцінювання безпеки ОС
- 5) Нормативно-правове регулювання в галузі
- 6) Засоби захисту ОС Windows
- 7) Засоби захисту UNIX-подібних систем
- 8) Архітектура безпеки і механізми захисту Android
- 9) Архітектура безпеки і механізми захисту iOS



Цикл лабораторних робіт



Курсом дисципліни передбачено 13 лабораторних робіт:

- 9 – на базі операційної системи Linux
- 2 – на базі операційної системи Windows
- 1 – на базі операційної системи Android
- 1 – на базі операційної системи iOS

Донецький національний технічний університет



Тема 2. Модель загроз для операційної системи. Лекція 2. Модель загроз

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Класична модель загроз ОС
- Модель STRIDE



Мета лекції

- Розглянути класичну модель загроз ОС
- Отримати уявлення про модель STRIDE



Етапи побудови засобу захисту

32

- 
1. Формування вимог
 2. Розроблення моделі загроз
 3. Розроблення політики безпеки, моделі безпеки
 4. Розроблення технічного завдання (ТЗ)
 5. Проектування
 6. Реалізація
 7. Тестування (сертифікація)
 8. Впровадження
 9. Супроводження



Модель загроз для операційної системи (класична)

33

- ☐ Сканування файлової системи
- ☐ Викрадення ключової інформації
- ☐ Підбирання паролів
- ☐ Збирання сміття
- ☐ Перевищення повноважень
- ☐ Програмні закладки
- ☐ Жадібні програми



Сканування файлової системи

34

- ☐ *Ця загроза полягає в тому, що порушник переглядає (сканує) всю файлову систему, і намагається прочитати усі файли поспіль*
- ☐ *Сканування файлової системи – це в першу чергу атака на політику безпеки*



Сканування файлової системи (продовження)

35

- ❑ ***В подальшому ми взагалі не будемо розглядати ті ОС, які не здійснюють контроль доступу до файлів***
 - Але й ті ОС, що такий контроль (керування, аудит) здійснюють, можуть бути атакованими
- ❑ ***Можливість НСД до об'єктів файлової системи визначається наявністю її вбудованої системи розмежування доступу і коректністю її адміністрування***
- ❑ ***Джерелом атаки може бути будь-який легальний користувач системи***
 - Цю атаку може здійснювати анонімний користувач (гість), якщо можливість анонімного входу в систему не заблокована



Викрадення ключової інформації

36

- ❑ *На рівні ОС використовується різна інформація, яку можна віднести до ключової. Найпоширеніший вид такої інформації – паролі доступу до системи*
- ❑ *Такий спосіб атак на паролі називають “не-електронні атаки” (non-electronic attacks)*
- ❑ *Типові способи викрадення паролів:*
 - Соціальна інженерія (social engineering)
 - Підглядання пароля в момент, коли користувач його набирає (shoulder surfing)
 - Викрадення паролів, що записані або роздруковані на папері (dumpster diving)
 - При тимчасовому фізичному доступі до комп'ютера — отримання паролів, що ненадійно зберігаються



Підбирання паролів

37

- ☐ ***Підбирання паролів передбачає використання засобів автентифікації для знаходження того пароля, який буде прийнятий як правильний***
- ☐ ***Будь-яка система передбачає можливість того, що користувач під час введення пароля зробив помилку, тому введення пароля можна повторити***



Підбирання паролів – активні атаки онлайн

38

❑ **Активні атаки онлайн (*Active Online Attacks*)**

- ❑ Вгадування паролів (**Password Guessing**) — в загальному випадку передбачає певні знання про жертву, що дозволяють передбачити, який пароль він(вона) могли б обрати
- ❑ Атака за словником (**Dictionary Attack**) — зазвичай дуже ефективна, якщо жертва не дуже вибаглива у вгадування паролів
- ❑ Повний перебір (**Brute Forcing Attack**) — в решті-решт дасть результат, але потрібний на це час може бути неприйнятно тривалим
- ❑ Також до активних атак онлайн відносять фішинг (**phishing**), ін'єкції хешу (**hash injection**), **LLMNR/NBT-NS poisoning**, застосування троянів, **spyware**, клавіатурних шпигунів (**keyloggers**) тощо



Підбирання паролів – пасивні атаки онлайн

39

□ **Пасивні атаки онлайн (*Passive Online Attacks*)**

- В таких атаках порушник прослуховує або записує дані, що проходять комунікаційним каналом, а потім використовує ці дані для проникнення в систему
 - Sniffing
 - Man-in-the-Middle
 - Атака відтворення (Replay Attack)



Підбирання паролів –атаки офлайн

40

□ Атаки оффлайн (*Offline Attacks*)

- В таких атаках порушник намагається відтворити пароль з (раніше отриманого) файлу хешів
 - Для підвищення ефективності застосовують **Rainbow Tables**



Підбирання паролів — протидія

41

❑ В усіх сучасних системах автентифікації вживаються заходи, які суттєво послаблюють цю загрозу

❑ Основні заходи наступні:

- обмеження швидкості введення паролів шляхом введення штучних затримок в процедуру їх перевірки
- обмеження кількості спроб введення паролів (як правило до 3...5), після чого система автентифікації на певний час блокує подальші спроби введення паролів з тієї консолі
- реєстрація спроб автентифікації
- негайне повідомлення адміністратора про повторні невдалі спроби автентифікації, тощо



Збирання сміття

- ☐ ***Збирання сміття – це отримання даних, які залишаються у об'єктах, що звільняються ОС після їх використання***
- ☐ ***У більшості файлових систем файли після їх видалення не знищуються фізично, а лише помічаються як знищені***



Збирання сміття – тимчасові файли

43

- ☐ ***Простий спосіб збирання сміття – копіювання і вивчення тимчасових файлів***
- ☐ ***Велику цінність має вміст “файлу підкачування”, який створюється для реалізації механізму віртуальної пам’яті***
- ☐ ***Деякі дуже інформативні для порушників файли можуть виникати під час збоїв***
- ☐ ***У більшості сучасних систем для мінімізації наслідків помилкового вилучення файли, які вилучають, копіюють у спеціальне сховище – “кошик для сміття” (trash bin), звідки їх можна повністю відновити***



Збирання сміття в оперативній пам'яті

44

- ☐ ***Зазвичай, прості ОС не займаються очищенням оперативної пам'яті***
- ☐ ***Коли прикладна програма запитує додаткову пам'ять, їй виділяють ділянки пам'яті, що продовжують зберігати дані попередніх програм, які ці ділянки використовували***



Перевищення повноважень

45

- ❑ ***Ця загроза полягає у тому, що порушник якимось чином отримує повноваження, що перевищують ті, які йому надані згідно політики безпеки***
- ❑ ***Перевищення повноважень можливе***
 - або через помилки в розробці й реалізації політики безпеки (наприклад, невірні налаштування системи розмежування доступу)
 - або шляхом використання наявних вразливостей в програмному забезпеченні, яке входить до складу ОС



Перевищення повноважень користувачів Unix-подібних ОС

46

❑ **Категорії користувачів в Unix-подібних ОС:**

- Адміністратор
- Спеціальний користувач
- Звичайний користувач
- Псевдокористувач

❑ **Перевищення повноважень реалізується при будь-якому несанкціонованому переході користувача з нижчої категорії до вищої. Типовими загрозами є перехід:**

- з категорії 3 до 1 (звичайний користувач отримав права адміністратора),
- з 4 до 3 (псевдокористувач отримав можливість інтерактивно працювати в системі з правами користувача),
- і з 4 до 1 (те ж саме, але з правами адміністратора)



Програмні закладки

47

- ☐ ***До програмних закладок відносять програми або окремі модулі програм, які протягом тривалого часу функціонують в комп'ютерній системі, здійснюючи заходи щодо приховування свого існування від користувача***
 - Найгірший варіант — rootkit
- ☐ ***Програмні закладки можуть впроваджуватись вірусом, “троянським конем”, мережним хробаком або безпосередньо користувачем-зловмисником***



Жадібні програми

48

- ❑ ***Жадібні програми*** – це шкідливі програми, які захоплюють значну частину ресурсів комп'ютера, внаслідок чого робота інших користувачів та/або процесів помітно утруднюється або взагалі стає неможливою
 - Жадібні програми можуть призводити до краху ОС
- ❑ ***Здебільшого, жадібні програми належать до класу “троянських коней”***
- ❑ ***Жадібні програми можуть бути Web-застосунками, які запускаються при заході браузером на певні Web-сторінки***



Доповнення класичної моделі загроз

- ☐ ***Шкідливі функції прикладних програм***
- ☐ ***Атаки на систему оновлення***
- ☐ ***Атаки на криптографічну підсистему***



Методика STRIDE

❑ **Фактично, *STRIDE* — це саме модель загроз**

- 1) Підміна мережних об'єктів (**Spoofing identity**)
- 2) Модифікація даних (**Tampering with data**)
- 3) Відмовлення від авторства (**Repudiation**)
- 4) Розголошення інформації (**Information disclosure**)
- 5) Відмова в обслуговуванні (**Denial of service**)
- 6) Підвищення привілеїв (**Elevation of privilege**)

❑ **Питання:**

- 1) Чи можна її застосувати до ОС безпосередньо?
- 2) На якій стадії?



Методика STRIDE насправді

❑ ***Це методика розроблення безпечних програм, яку пропонують компанії Microsoft і Symantec***

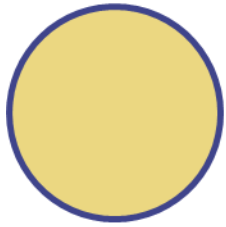
❑ ***Складається з певних етапів:***

- 1) Декомпозиція програми на складові
 - Застосування DFD – Data Flow Diagrams
- 2) Визначення (моделювання) загроз
 - Застосування методики STRIDE
- 3) Ранжування загроз (оцінка ризиків)
 - Застосування методики DREAD
- 4) Запобігання загрозам



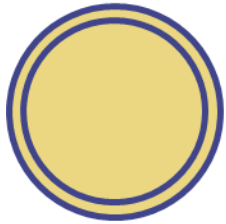
DFD – діаграми потоків даних – умовні позначення

52



Процес

Перетворення
або оброблення
даних



Група процесів

Перетворення
або оброблення
даних



Сховище даних

Місце зберігання
тимчасових або
постійних даних



Межа

Межі між машинами,
областями з різним
рівнем безпеки,
адресного простору
тощо



Зовнішній суб'єкт

Зовнішнє джерело
даних

Автентифіковані
дані



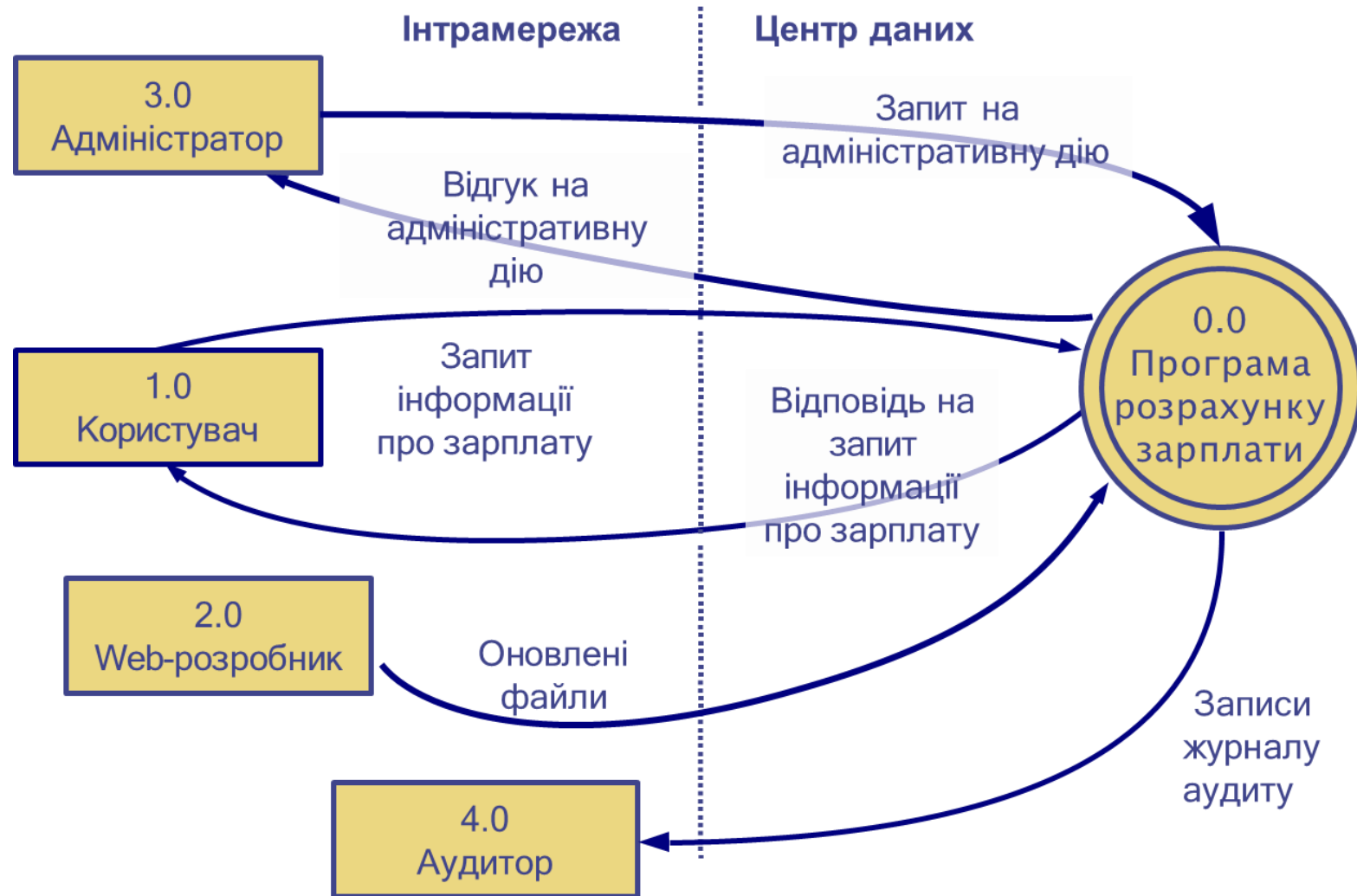
Потік даних

Описує потоки даних
зі сховищ даних,
процесів і зовнішніх
суб'єктів



Застосування DFD-діаграм₁

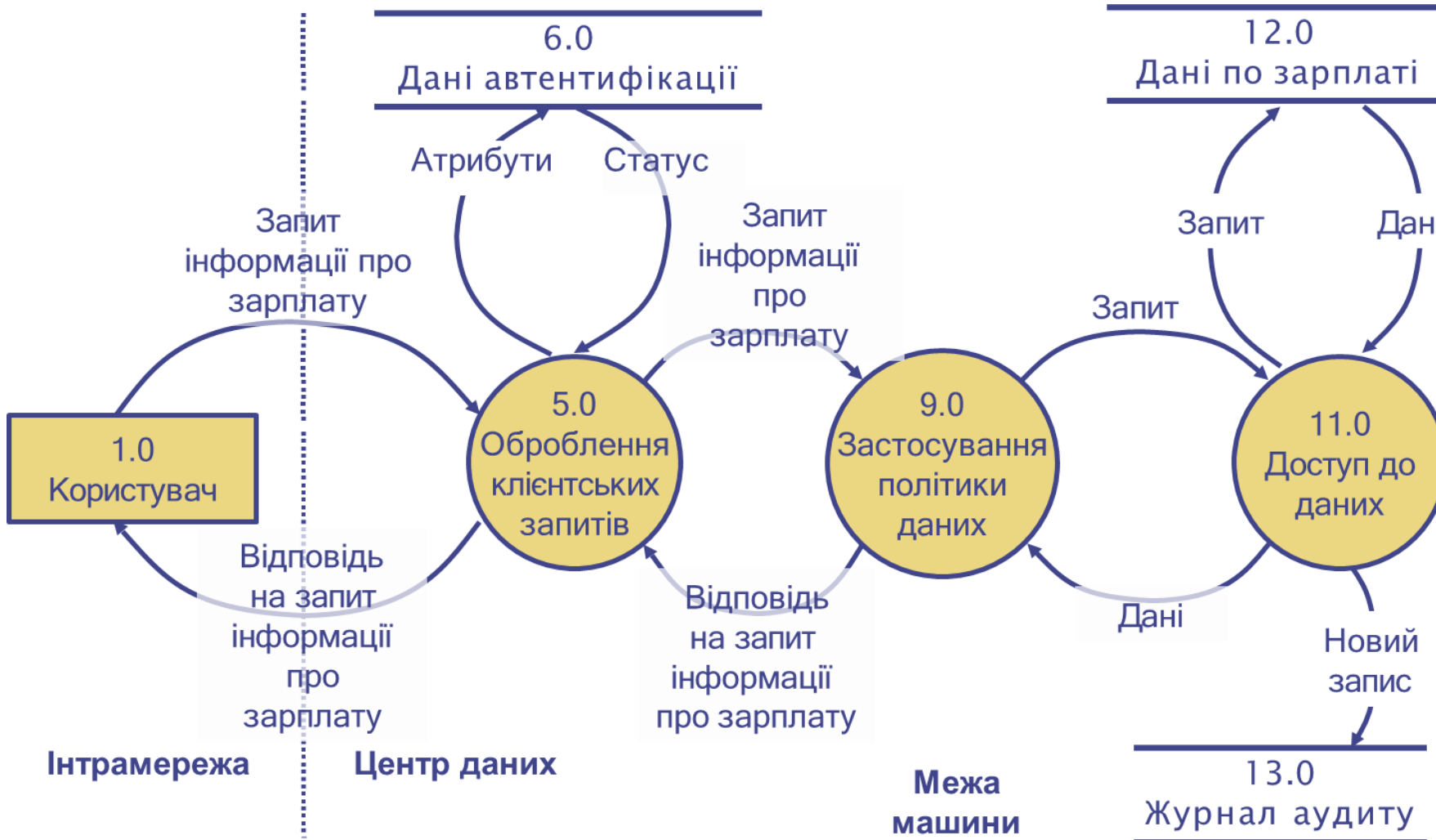
53





Застосування DFD-діаграм₂

54





Підхід Common Criteria (ISO/IEC 15408)

55

- ☐ **Критерії, що застосовують для оцінювання безпеки продуктів інформаційних технологій (ПІТ), а також заснований на них міжнародний стандарт, що застосовують для сертифікації цих продуктів (зокрема, операційних систем)**
- ☐ **Модель загроз для продукту описують у документах ST (Security Target — Завдання з безпеки) та PP (Protection Profile — Профіль захисту) у відповідних розділах**
- ☐ **Сам стандарт не містить переліку можливих загроз — їх формують розробники названих документів**



Controlled Access Protection Profile (CAPP) — 1999

56

- ☐ Був розроблений як заміна класу C2 зі стандарту TCSEC (“Оранжева книга”)
- ☐ Усі задачі захисту (*security objectives*) визначені з організаційних політик безпеки (*organizational security policies*)
- ☐ Модель загроз у явному вигляді не сформульована
- ☐ На сьогодні є застарілим і для сертифікації сучасних систем не використовується



Operating Systems Protection Profile (OSPP) — 2010

57

□ **Визначені:**

- Активи (**Assets**), що підлягають захисту
 - Порушники (**Threat Agents**)
 - Загрози (**Threats**), яким має протидіяти об'єкт оцінювання (**Target of Evaluation, TOE**)
-
- Надалі: **TSF = TOE security functions** =(нашою мовою) **КЗЗ** — комплекс засобів захисту



АКТИВИ

58

□ **Активи, що підлягають захисту:**

- Постійні об'єкти сховища, що застосовуються для зберігання даних користувача і КЗЗ, у випадках де ці дані мають бути захищені від будь-якої з названих подій:
 - Несанкціонований (unauthorized) доступ на зчитування
 - Несанкціонована модифікація
 - Несанкціоноване вилучення об'єкту
 - Несанкціоноване створення нових об'єктів
 - Несанкціоноване керування атрибутами об'єкту
- Тимчасові об'єкти сховища, включаючи дані з мережі
- Функції КЗЗ і пов'язані з ними дані КЗЗ
- Ресурси під керуванням КЗЗ, що застосовуються для зберігання вищеназваних об'єктів, включаючи метадані, необхідні для керування цими об'єктами



Загрози, яким протидіє ОО₁

59

❑ **T.ACCESS.TSFDATA** (Доступ до даних КЗЗ)

- Агент загроз може читати або модифікувати дані КЗЗ без необхідної авторизації, коли ці дані зберігаються або передаються

❑ **T.ACCESS.USERDATA** (Доступ до даних користувача)

- Агент загроз може отримати доступ до даних користувача, що зберігаються, обробляються, або передаються ОО, без належної авторизації у відповідності до політики безпеки ОО

❑ **T.ACCESS.TSFFUNC** (Доступ до функцій КЗЗ)

- Агент загроз може використовувати або модифікувати функціональність КЗЗ без необхідних привілеїв для надання собі або іншим несанкціонованого доступу до даних КЗЗ або користувачів

❑ **T.ACCESS.COMM** (Доступ до каналу зв'язку)

- Агент загроз може отримати доступ до каналу зв'язку (communication channel), що встановлює довірені відносини між ОО й іншою віддаленою довіреною інформаційною системою, або удавати із себе (masquerade) іншу віддалену довірену інформаційну систему



Загрози, яким протидіє ОО₂

60

❑ ***T.RESTRICT.NETTRAFFIC*** (*Доступ до мережного трафіку*)

- Агент загроз може отримати доступ до інформації або передавати інформацію іншим абонентам через мережні канали зв'язку без належної авторизації для такої спроби обміну з боку політики керування потоками інформації

❑ ***T.IA.MASQUERADE*** (*"Маскарад"*)

- Агент загроз може удавати з себе (masquerade) авторизовану сутність, включаючи сам ОО, або частину ОО, для отримання несанкціонованого доступу до даних користувача, даних КЗЗ, або ресурсів ОО

❑ ***T.IA.USER*** (*Неналежна автентифікація користувачів*)

- Агент загроз може отримати доступ до даних користувача, даних КЗЗ, або ресурсів ОО, за винятком загальнодоступних об'єктів, без ідентифікації та автентифікації



Protection Profile for General Purpose Operating Systems

- ☐ *Це — сучасний профіль захисту (остання редакція — 2018)*
- ☐ *Вимагає лише рівень гарантій 1 (найнижчий!)*
- ☐ *В якості агента загроз розглядають зловмисника (an attacker)*



Модель загроз PPGPOS₁

62

❑ **T.NETWORK_ATTACK** (*Мережна атака*)

- Зловмисник знаходиться у каналі зв'язку або десь в мережній інфраструктурі
- Зловмисники можуть вступати в інформаційний обмін з прикладними програмами і сервісами, що працюють під керуванням або є частиною ОС, з метою компрометації
- Такий обмін може бути модифікуванням встановлених легітимних обмінів

❑ **T.NETWORK_EAVESDROP** (*Прослуховування мережі*)

- Зловмисник знаходиться на каналі зв'язку або десь в мережній інфраструктурі
- Зловмисники можуть здійснювати моніторинг і отримувати доступ до даних, якими обмінюються прикладні програми і сервіси, що працюють під керуванням або є частиною ОС



Модель загроз PPGPOS₂

63

❑ **T.LOCAL_ATTACK** (*Локальна атака*)

- Зловмисник може скомпрометувати прикладні програми, що виконуються під керуванням ОС
- Скомпрометована прикладна програма може надавати спеціально (зловмисно) сформовані дані для ОС через низку каналів обміну, включаючи непривілейовані системні виклики і обмін повідомленнями через файлову систему

❑ **T.LIMITED_PHYSICAL_ACCESS** (*Обмежений фізичний доступ*)

- Зловмисник може намагатись здійснити доступ до даних ОС в умовах обмеженого часу з фізичним пристроєм

Донецький національний технічний університет



Тема 3. Розроблення захищених операційних систем.

Лекція 3. Розроблення захищених операційних систем

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Розроблення захищених ОС
- Принципи створення захищених ОС
- Типова архітектура КЗЗ



Мета лекції

- Розглянути питання розроблення захищених ОС та принципи побудови таких ОС
- Розглянути типову архітектуру КЗЗ, функції підсистем



Захищена операційна система

- ❑ Будемо вважати **захищеною** таку **ОС**, яка передбачає захист від основних загроз, визначених в ухваленій моделі загроз
- ❑ Якщо ОС передбачає захист лише від деяких загроз з ухваленої моделі, то таку ОС іноді називають **частково захищеною**



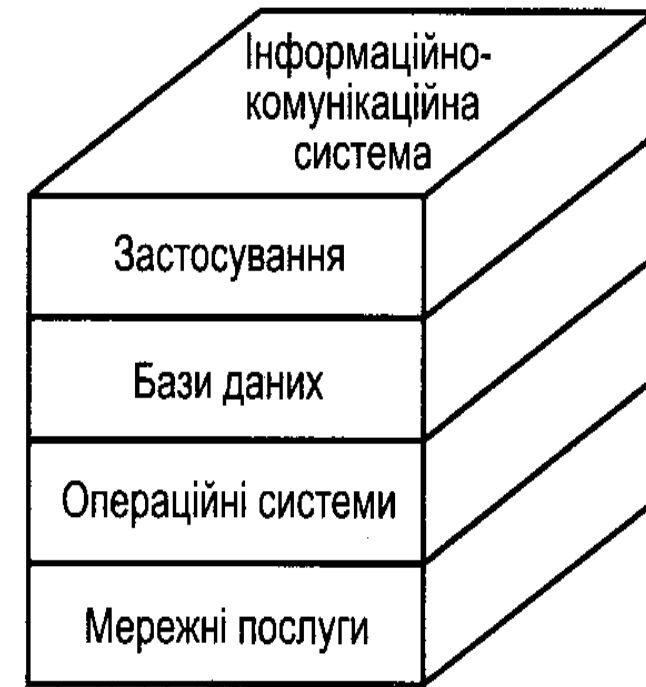
Підходи до побудови захищених операційних систем

- ☐ Розроблення захищених систем “з нуля”
- ☐ Побудова так званих “довірених” (trusted) версій шляхом модернізації систем, що існують



Рівні ІКС

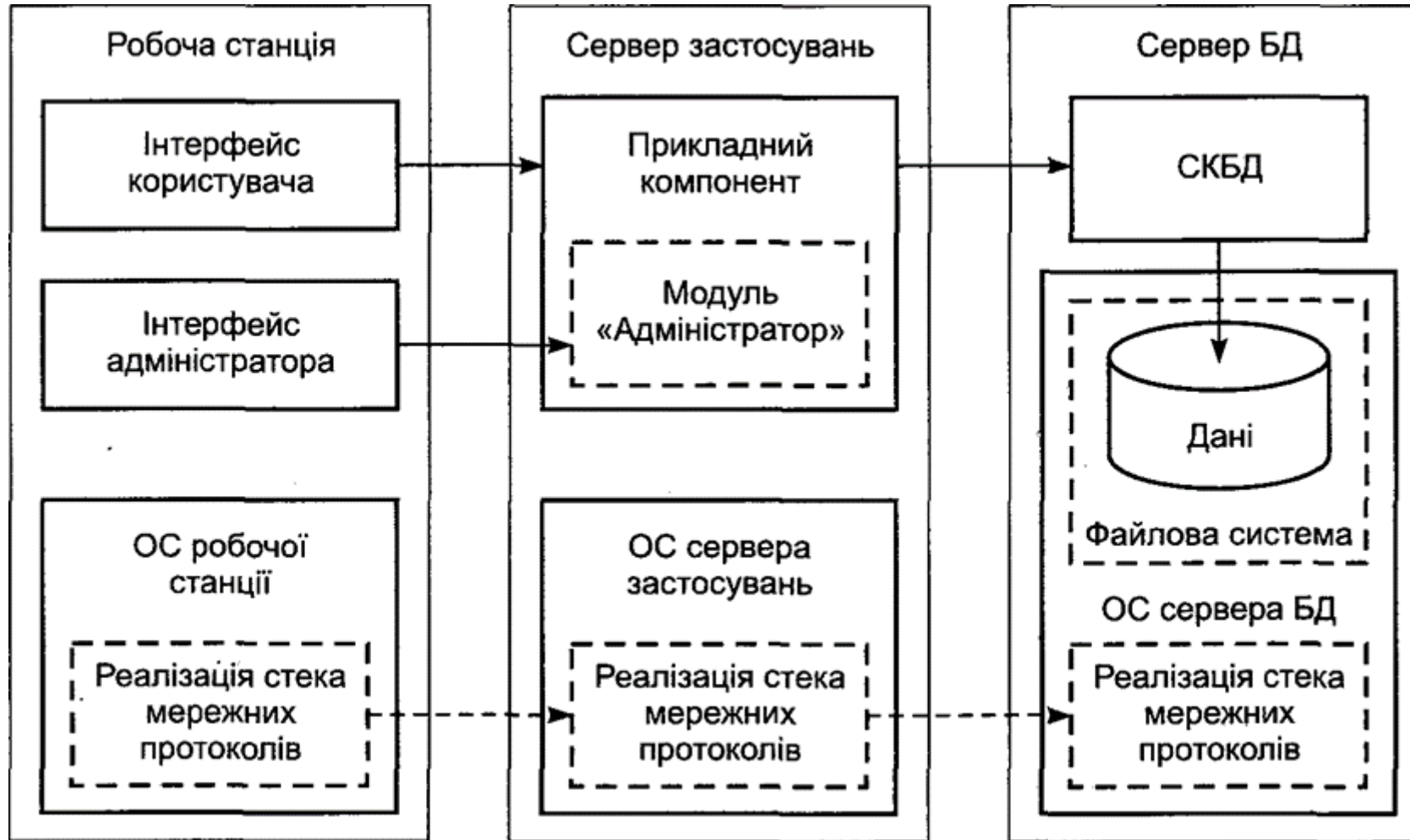
- ☐ ***Рівень мережі***
- ☐ ***Рівень операційних систем***
- ☐ ***Рівень СКБД***
- ☐ ***Рівень прикладного ПЗ***





Приклад типової ІКС

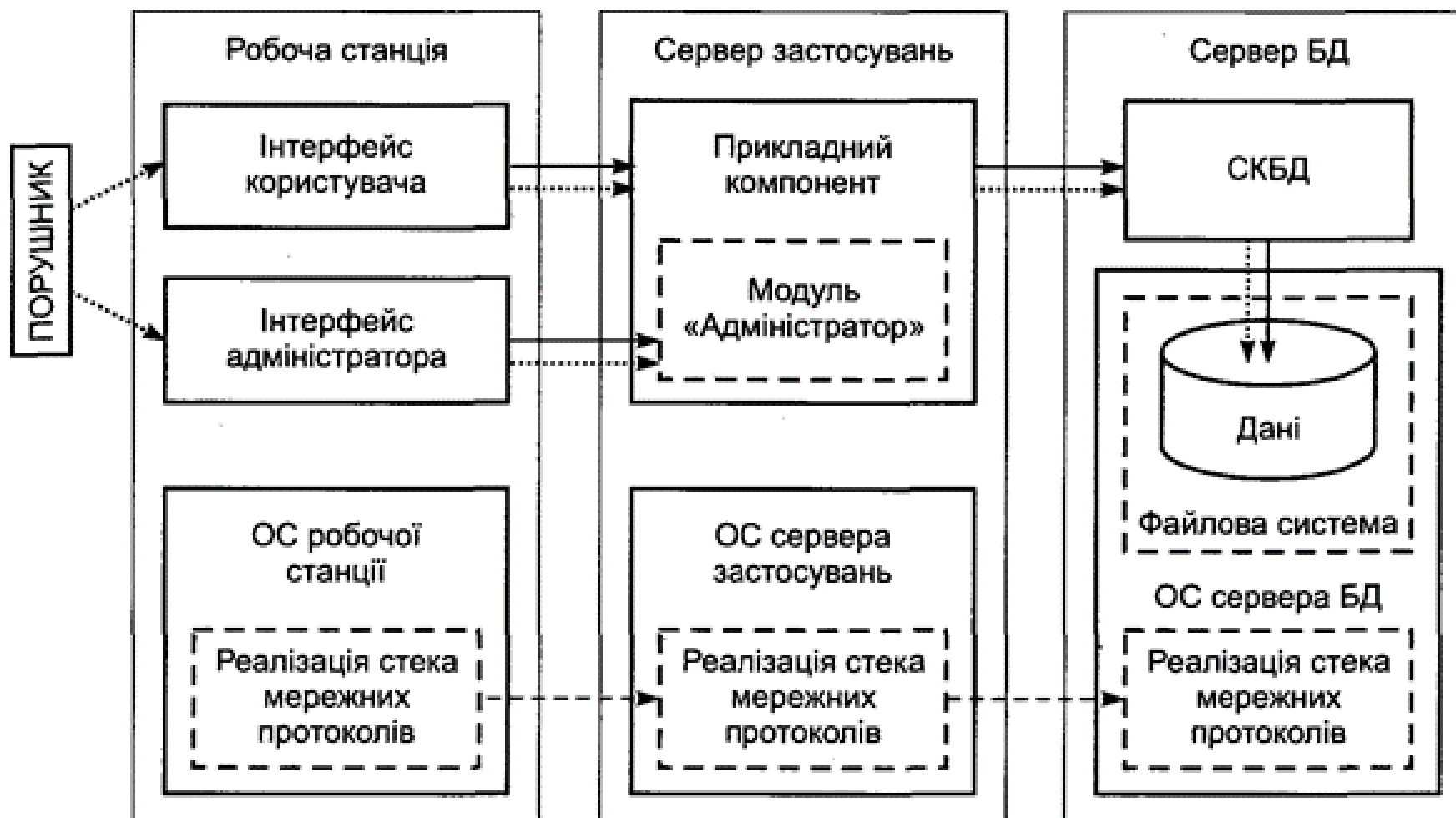
70





Атака на прикладне ПЗ

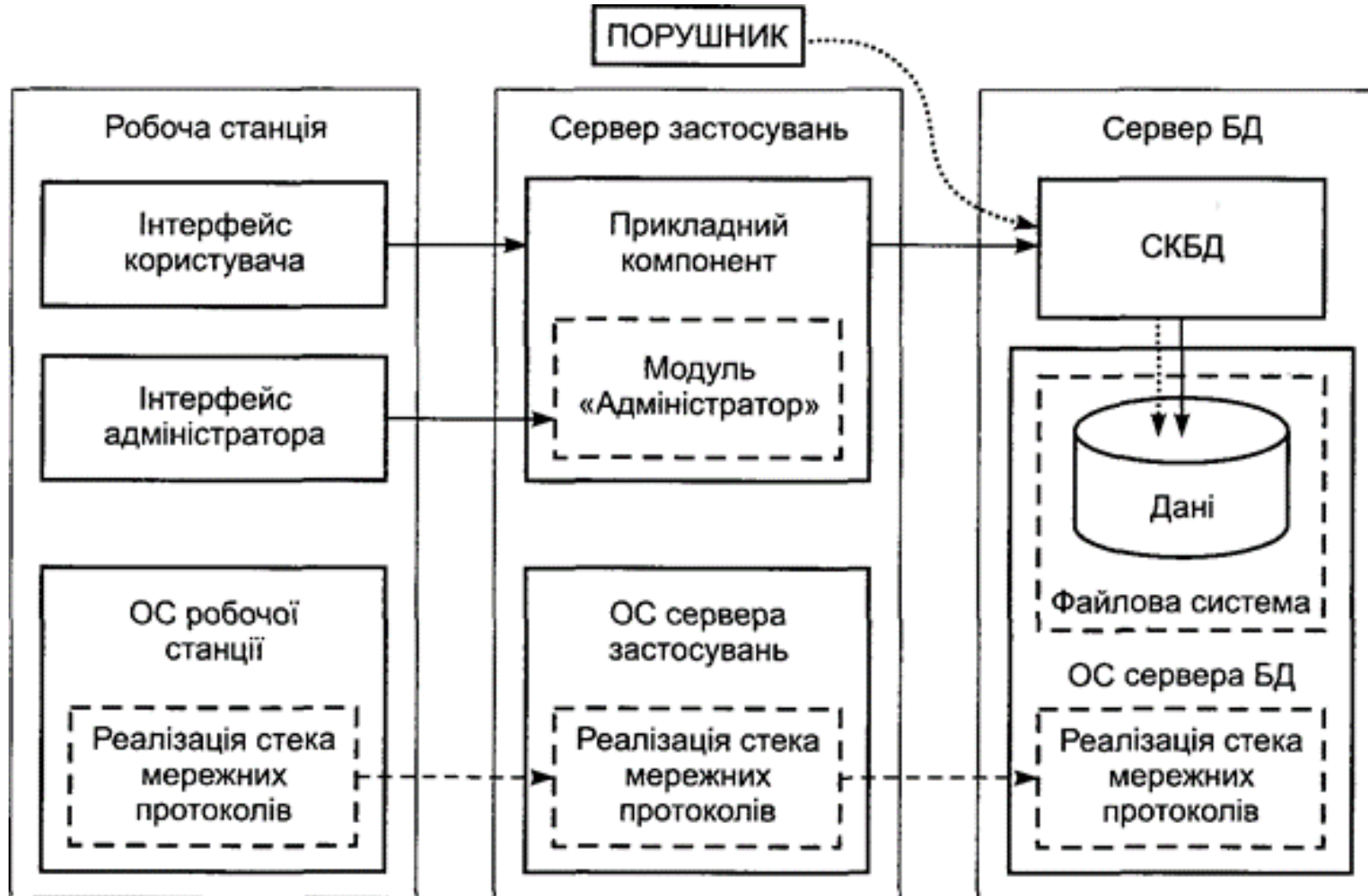
71





Атака на рівні СКБД

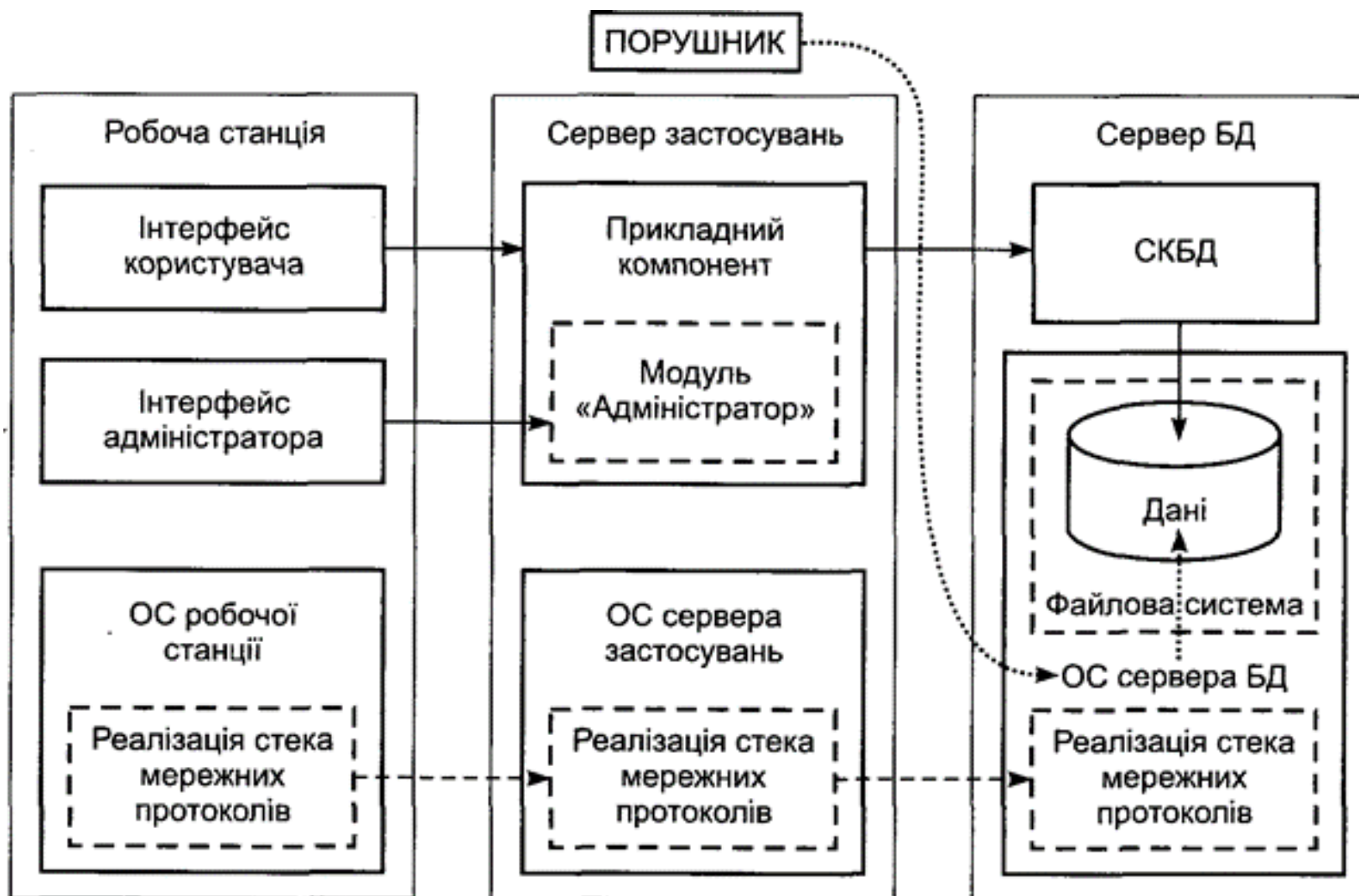
72





Атака на рівні ОС

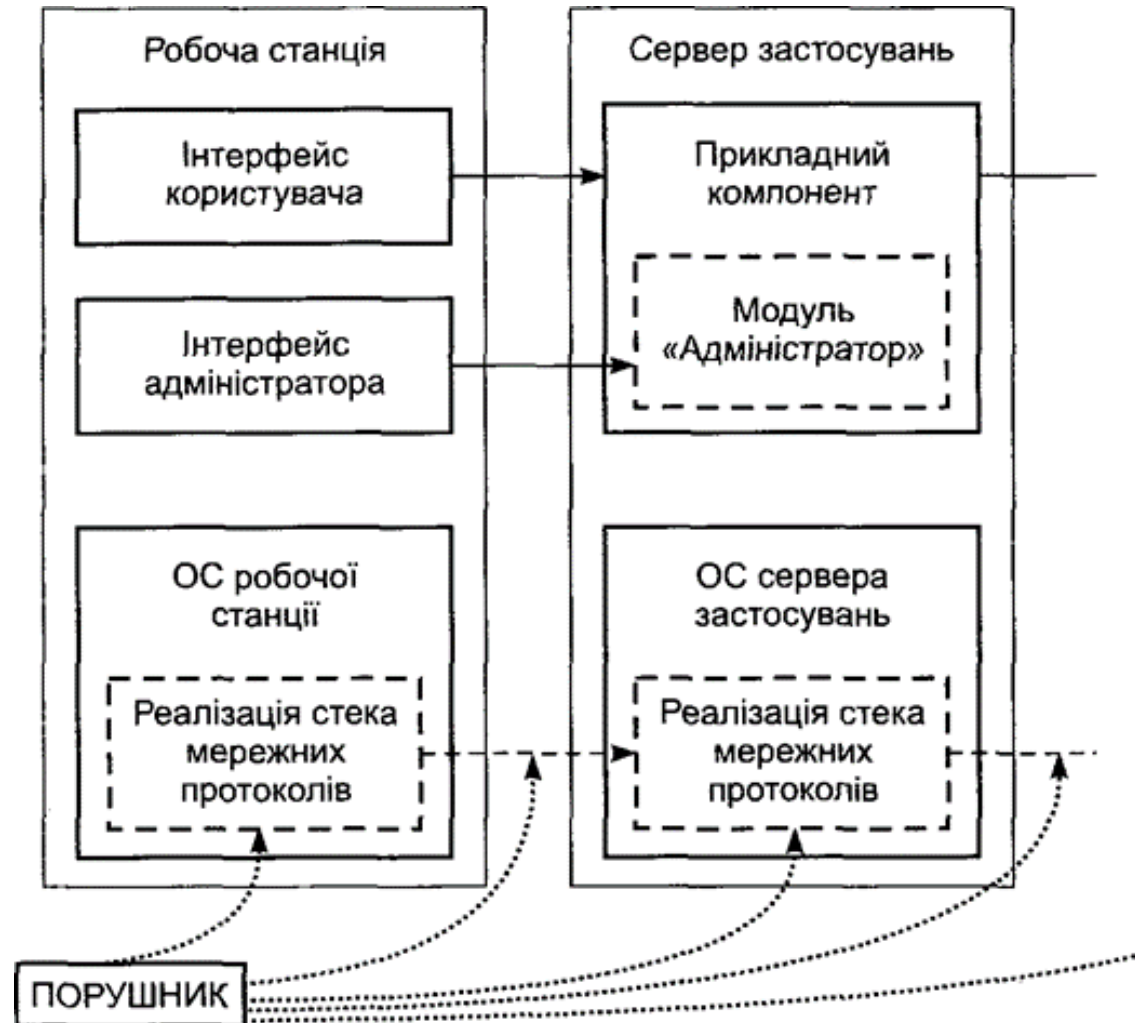
73





Атака на рівні мережі

74





Розроблення захищених систем

75

- ☐ *При розробці захищених систем “з нуля” на етапі проектування закладаються усі функціональні можливості та архітектурні рішення, що розраховані на сертифікацію за встановленим профілем і рівнем гарантій*
- ☐ *Головною рисою цього підходу є розробка методів гарантованої реалізації встановлених вимог*
- ☐ *Застосовується класична схема проектування захищених систем*



Побудова “довірених” версій шляхом модернізації систем, що існують

76

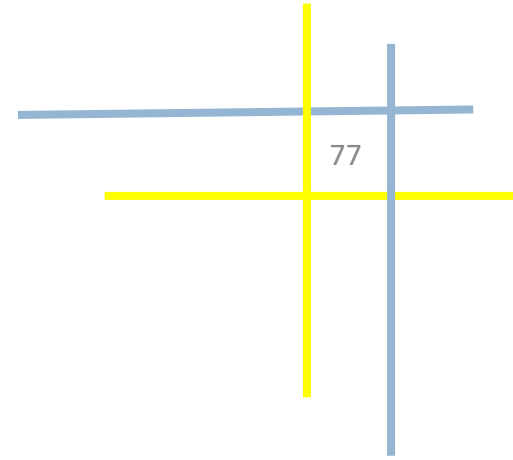
❑ При побудові “довірених” версій шляхом модернізації систем, що існують, як правило:

- ❑ додають функції шифрування і цифрового підпису (в сучасних ОС ці функції вже впроваджені),
- ❑ підсилюють керування доступом шляхом впровадження мандатного (адміністративного) керування (**MAC — Mandatory Access Control**),
- ❑ розподіляють обов’язки адміністратора системи між різними обліковими записами або “ролями” (на практиці це означає, що в системі також впроваджують рольове керування доступом, **RBAC — Role-Based Access Control**),
- ❑ впроваджують додаткові засоби ідентифікації й автентифікації, аудиту та моніторингу.



Принципи створення побудови захищених операційних систем

- ☐ ***Принцип інтегрованості***
- ☐ ***Принцип інваріантності***
- ☐ ***Принцип уніфікації***
- ☐ ***Принцип гарантованості***
- ☐ ***Принцип коректності***





Принцип інтегрованості

78

❑ Засоби захисту повинні бути вбудовані в систему таким чином, щоби усі без винятку механізми взаємодії знаходились під їх контролем

- ❑ Найпростішим методом, що реалізує цей принцип при створенні ОС, є максимальне обмеження числа механізмів взаємодії та інтеграція засобів захисту безпосередньо в ці механізми
- ❑ Додавання “сторонніх” засобів, що не повністю інтегруються в ОС, як правило не дозволяє досягти повного контролю механізмів взаємодії



Принцип інваріантності

79

- ☐ ***Засоби захисту не повинні залежати від особливостей реалізації утиліт і прикладних програм, і не повинні враховувати логіку їх функціонування***
- ☐ ***Засоби захисту повинні бути універсальними для усіх типів взаємодій***
 - Для ОС інваріантність засобів захисту може бути досягнута шляхом застосування строго регламентованої парадигми функціонування програм, що обмежує способи взаємодій



Принцип уніфікації

80

- Засоби захисту мають бути універсальними, що дозволяє використовувати їх без змін як для реалізації різних моделей безпеки, так і для керування доступом до об'єктів різної природи**
- Повинна існувати однозначна відповідність між взаємодіями суб'єктів і об'єктів, що контролюються, та операціями доступу, керування якими описується моделями безпеки
 - При розробленні ОС слідування цьому принципу приводить до необхідності створення універсального інтерфейсу доступу, що об'єднує всі способи взаємодій між суб'єктами й об'єктами, всі функції якого однозначним чином відображуються на множину операцій, що описуються моделлю безпеки



Принцип гарантованості

81

□ Для забезпечення реальної здатності протидіяти атакам необхідно виключити усі чинники, які спричиняють виникнення вразливостей

- Усі механізми реалізації атак базуються на використанні наявних вразливостей
- Однією з головних причин появи вразливостей є непослідовність в реалізації керування доступом



Принцип коректності

82

❑ Засоби захисту повинні реалізовувати керування доступом відповідно до формальних моделей

❑ Наявність несутеречливої моделі безпеки:

- дозволяє формально обґрунтувати безпеку системи,
- надає об'єктивний критерій коректності її роботи,
- може бути основою для побудови вичерпних тестів, що перевіряють правильність роботи засобів захисту в усіх режимах і обставинах.



Типова архітектура комплексу засобів захисту операційних систем

83

Основні підсистеми КЗЗ ОС





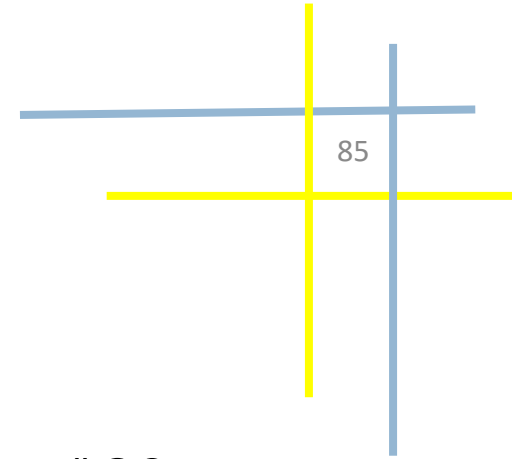
Основні підсистеми КЗЗ ОС₁

□ **Розмежування доступу**

- Кожному користувачеві надається доступ лише до тих захищених об'єктів, доступ до яких дозволений йому політикою безпеки
- Ця підсистема безпосередньо реалізовує політику безпеки



Основні підсистеми КЗЗ ОС₂



□ **Ідентифікація й автентифікація**

- Жодний користувач не може розпочати роботу в середовищі захищеної ОС, не надавши системі свого ідентифікатора і не підтвердивши справжність наданого ідентифікатора за допомогою додаткової інформації, що автентифікує цього користувача
- Система повинна ідентифікувати усі об'єкти, а також здійснювати автентифікацію об'єктів, що не знаходяться під її безпосереднім контролем (мережні об'єкти)



Основні підсистеми КЗЗ ОС_з

86

□ **Реєстрація та аудит**

- В захищеній ОС здійснюється реєстрація всіх подій, що є потенційно небезпечними
- Підсистема аудиту здійснює захист журналів, в яких відбувається реєстрація, від НСД
- Також ця підсистема може надавати засоби для аналізу журналів і відстеження джерел тих чи інших подій



Основні підсистеми КЗЗ ОС₄

87

□ **Антивірусний захист**

- Як правило, під підсистемою антивірусного захисту розуміють сукупність програм, які надають можливість виявляти і знешкоджувати відомі шкідливі програми, що відносяться як до вірусів (у широкому розумінні – включаючи троянських коней, мережних хробаків, шпигунські програми), так і до засобів здійснення атак
- Без антивірусного захисту в наш час неможливо підтримувати ОС у безпечному стані, особливо якщо вона встановлена на комп'ютері, що підключений до мережі
- Часто антивірусні засоби не входять до складу ОС, а постачаються окремо



Основні підсистеми КЗЗ ОС₅

88

□ **Забезпечення цілісності**

- Будь-яка сучасна ОС надає додаткові засоби для захисту цілісності даних не лише від НСД, але й від випадкових помилок, а також від аварій і збоїв системи
- Насамперед, це стосується даних у файлових системах
- Такі засоби реалізують можливості відкату, а також автоматизацію процесу створення резервних копій і відновлення з них



Основні підсистеми КЗЗ ОС₆

89

□ **Оновлення програм**

- Сучасні технології розроблення програмних продуктів — як прикладних програм, так і ОС, не дозволяють гарантувати відсутність уразливостей. Тому сучасний підхід до забезпечення захищеного стану інформаційно-телекомунікаційних систем передбачає обов'язкове постійне оновлення програм
- Підсистема оновлення програм повинна гарантувати встановлення оновлень лише з надійних джерел



Основні підсистеми КЗЗ ОС₇

□ **Криптографічні функції**

- Криптографічні функції застосовуються для захисту конфіденційності і цілісності інформації, для автентифікації і забезпечення неможливості відмовлення від авторства
- Криптографічні функції можуть використовуватись в якості самостійних засобів захисту (шифрування файлових систем), або в якості допоміжних механізмів в інших засобах (криптографічний захист цілісності журналів реєстрації)



Основні підсистеми КЗЗ ОС₈

91

□ **Мережні функції**

- Мережний обмін є можливим шляхом доступу зловмисників до системи і найбільш критичним джерелом проникнення в систему шкідливих програм. Тому мережні функції відносять до комплексу засобів захисту
- Мережні функції повинні протистояти мережним атакам, насамперед — атакам на протоколи



Основні підсистеми КЗЗ ОС₉

92

❑ **Керування політикою безпеки**

- Захищена ОС повинна надавати інтерфейси, які дозволяють адміністраторам ефективно вирішувати завдання з підтримання адекватної політики безпеки
- Обов'язково надаються інтерфейси для налаштування підсистем розмежування доступу, ідентифікації й автентифікації, аудиту

❑ **Апаратні засоби**

- КЗЗ ОС спирається на функції захисту, реалізовані в апаратних засобах - процесорі і системній платі



Особливості архітектури КЗЗ ОС

93

- ☐ ***КЗЗ ОС практично завжди є сукупністю значної кількості програмних модулів***
- ☐ ***В деяких ОС, де використовується більша кількість рівнів привілеїв процесів (кілець захисту), будується ієрархія засобів захисту***
- ☐ ***В сучасних ОС КЗЗ чітко виділяється в архітектурі системи***
 - наприклад, Windows
- ☐ ***Зустрічаються такі ОС, в яких підсистеми і окремі компоненти КЗЗ розпорошені по всій системі***
 - наприклад, традиційні системи Unix і Linux
- ☐ ***Як правило, підсистема захисту дозволяє додавати додаткові модулі, які реалізують підсилені функції захисту, для чого передбачаються відповідні інтерфейси***



Адміністративні заходи захисту

- ☐ ***Постійний контроль коректності функціонування ОС, зокрема, її підсистеми захисту***
- ☐ ***Організація й підтримання адекватної політики безпеки***
- ☐ ***Навчання користувачів***
- ☐ ***Створення резервних копій***
- ☐ ***Постійний контроль змін в конфігураційних даних і політиці безпеки ОС***



Адекватна політика безпеки

95

- ☐ ***Адекватна політика безпеки в ОС – це така політика безпеки, яка забезпечує захист від визначеної множини загроз з достатньою надійністю***
- ☐ ***Вибір адекватної політики безпеки – завдання адміністратора безпеки, підтримання цієї політики безпеки – одна з найважливіших задач адміністратора ОС***
- ☐ ***Не слід вважати адекватною політику безпеки, яка забезпечує максимальний можливий захист***
 - Як правило, підвищення захищеності пов'язано з обмеженням функціональності і утрудненням роботи користувачів
- ☐ ***Єдиної адекватної політики безпеки на всі можливі випадки не існує й існувати не може***



Основні етапи визначення і підтримання адекватної політики безпеки₁

- ☐ ***Аналіз загроз***
- ☐ ***Формування вимог до політики безпеки***
- ☐ ***Формальне визначення політики безпеки***



Основні етапи визначення і підтримання адекватної політики безпеки₂

☐ ***Втілення політики безпеки***

☐ ***Підтримання і корекція політики безпеки***

Донецький національний технічний університет



Тема 4. Стандарти оцінювання захищеності операційних систем.

Лекція 4. Стандарти оцінювання захищеності операційних систем

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



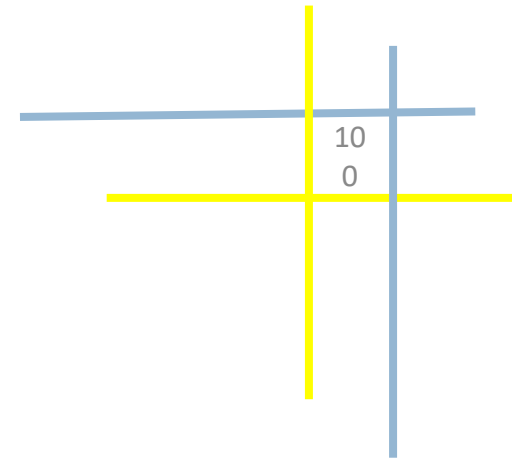
Лекційні питання

- Мета лекції
- Стандарти оцінювання захищеності операційних систем



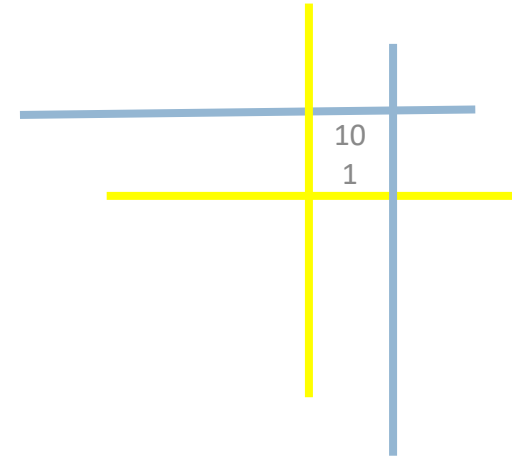
Мета лекції

- Розглянути основні стандарти оцінювання захищеності операційних систем





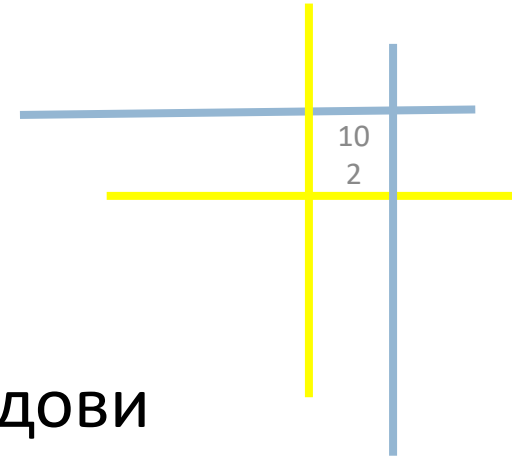
Призначення стандартів



- ❑ **Стандарт повинен регламентувати**
 - концепції інформаційної безпеки, підходи до її досягнення
 - вимоги до систем і шляхи їх реалізації
 - систему критеріїв оцінювання захищеності систем, процедури оцінювання систем за цими критеріями
- ❑ **Стандарти створюють основу для погодження вимог**
 - розробників систем
 - споживачів (користувачів систем і власників інформації, що в них обробляється)
 - експертів, які оцінюють захищеність інформації в системах (а в разі необхідності – також і державних або відомчих органів, які видають дозвіл на експлуатацію системи і обробку в ній певної інформації)



Оптимістичний погляд на стандарти інформаційної безпеки



- ☐ Викорінення вад захисту – це тактичний шлях побудови захищених систем, чи підхід “знизу”
- ☐ Стандарти визначають стратегічний підхід до побудови захищених систем, або підхід “зверху”
- ☐ Стандарти лише частково вирішують питання захищеності систем
- ☐ Підготовка системи до оцінювання за стандартом (compliance) позитивно впливає на якість розроблюваної системи



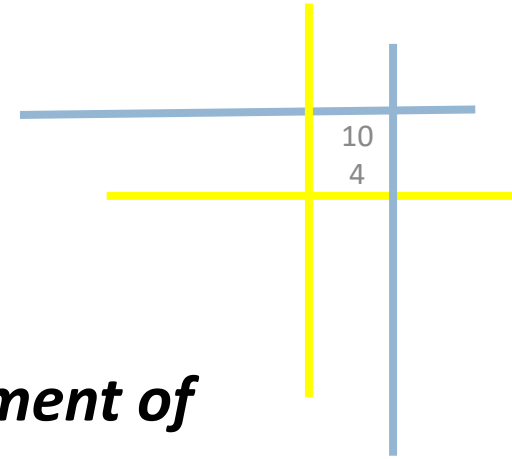
Розвиток стандартів інформаційної безпеки

103

- ❑ ***TCSEC – Критерії оцінювання захищених комп'ютерних систем Міністерства оборони США («Оранжева книга»), 1983 р.***
- ❑ ***ITSEC – Європейські критерії безпеки інформаційних технологій, 1991 р.***
- ❑ ***FCITS – Федеральні критерії безпеки інформаційних технологій США, 1992 р.***



TCSEC (“Помаранчева книга”)



- ❑ ***Trusted Computer System Evaluation Criteria.*** – ***US Department of Defense.*** – ***CSC-STD-001-83, 1983.***
- ❑ ***Trusted Network Interpretation.*** – ***National Computer Security Center.*** ***NCSC-TG-005 Version 1, 1987.***
- ❑ ***Trusted Database Management System Interpretation.*** – ***National Computer Security Center.*** ***NCSC-TG-021 Version 1, 1991.***
- ❑ ***The Interpreted Trusted Computer System Evaluation Criteria Requirements.*** – ***National Computer Security Center.*** ***NCSC-TG-007-95, 1995.***



TCSEC



□ Визначення безпечної системи

- система, що підтримує керування доступом таке, що лише уповноважені користувачі або процеси, що діють від їх імені, одержують можливість читати, записувати, створювати і знищувати інформацію

□ 6 вимог до безпечної системи

- Політика безпеки
- Мітки безпеки
- Ідентифікація і автентифікація
- Реєстрація і облік
- Коректність засобів захисту
- Неперервність захисту



TCSEC: перелік вимог₁

□ Політика безпеки

- Дискреційне керування доступом



ТСЕС: перелік вимог₂

☐ Політика безпеки

- Дискреційне керування доступом
- Повторне використання об'єктів



ТСЕСЕС: перелік вимог₃

□ Політика безпеки

- Дискреційне керування доступом
- Повторне використання об'єктів
- Мітки безпеки
 - Цілісність міток безпеки
 - Експорт поміченої інформації
 - Мітки повноважень суб'єктів
 - Мітки пристроїв



ТСЕСЕС: перелік вимог₃

□ Політика безпеки

- Дискреційне керування доступом
- Повторне використання об'єктів
- Мітки безпеки
 - Цілісність міток безпеки
 - Експорт поміченої інформації
 - Мітки повноважень суб'єктів
 - Мітки пристроїв



ТСЕС: перелік вимог₄

□ Політика безпеки

- Дискреційне керування доступом
- Повторне використання об'єктів
- Мітки безпеки
 - Цілісність міток безпеки
 - Експорт поміченої інформації
 - Мітки повноважень суб'єктів
 - Мітки пристроїв
- Мандатне керування доступом



ТСЕСЕС: перелік вимог₅

□ Аудит

- Ідентифікація і автентифікація
 - Пряма взаємодія з КЗЗ
- Реєстрація і облік подій



TCSEC: перелік вимог₆

□ Коректність

- Коректність функціонування
 - Архітектура системи
 - Цілісність системи
 - Аналіз прихованих каналів
 - Керування безпекою
 - Відновлення
- Коректність розробки
 - Тестування безпеки
 - Розробка і верифікація специфікацій
 - Керування конфігурацією
 - Дистрибуція



TCSEC: перелік вимог₇

□ Документація

- Настанова з безпеки користувача
- Настанова адміністратора безпеки
- Документування процесу тестування
- Документування процесу розроблення



TCSEC: класифікація систем

Групи	Класи	
A	A1	Верифікований захист
B	B1 B2 B3	Мандатний захист
C	C1 C2	Дискреційний захист
D	D1	Мінімальний захист



Основні вимоги рівня С2 щодо операційних систем

- ☐ Засіб безпечного входу в систему
- ☐ Дискреційне керування доступом
- ☐ Контроль (аудит) безпеки
- ☐ Захист від повторного використання об'єкта



Деякі важливі вимоги рівня В щодо операційних систем

- ☐ ***Механізм “довіреного маршруту”***
- ☐ ***Довірений засіб керування***



Висновки по TCSEC

117

- ☐ *Перша спроба створити єдиний стандарт безпеки, що розрахований на розробників, споживачів і спеціалістів із сертифікації КС*
- ☐ *Орієнтований на системи спеціального (військового) застосування, головним чином на операційні системи (домінують вимоги з конфіденційності)*
- ☐ *Критерії гарантій реалізації засобів захисту і адекватності політики безпеки розроблені недостатньо*
- ☐ *Оцінка системи зводиться до перевірки виконання вимог одного з наперед визначених класів. Шкала класів є ієрархічною*



ITSEC – Європейські критерії

118

- ❑ ***Information Technology Security Evaluation Criteria. Harmonized Criteria Of France-Germany- Netherlands-United Kingdom. – Department of Trade and Industry. – London, 1991.***
- ❑ ***Спільна розробка уповноважених служб Великої Британії, Нідерландів, Німеччини та Франції***
- ❑ ***Документ базується на “Помаранчевій книзі” і може розглядатись як її розвиток і доповнення***

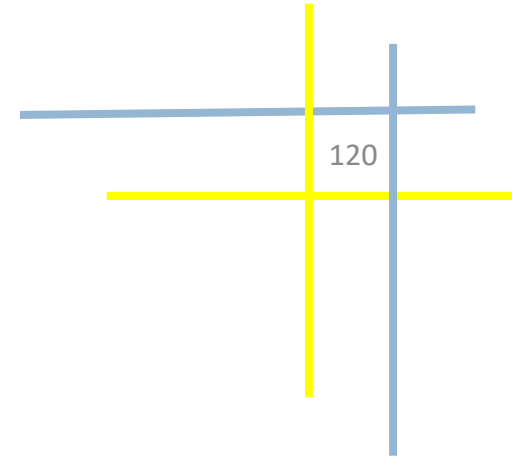


ITSEC – класи безпеки

- ☐ ***F-C1, F-C2, F-B1, F-B2, F-B3 – відповідають TCSEC***
- ☐ ***F-IN – підвищені вимоги до забезпечення цілісності (СКБД)***
- ☐ ***F-AV – підвищені вимоги до забезпечення працездатності (промислові системи керування)***
- ☐ ***F-DI – розподілені системи з підвищеними вимогами до цілісності***
- ☐ ***F-DC – розподілені системи з підвищеними вимогами до конфіденційності***
- ☐ ***F-DX – розподілені системи з підвищеними вимогами до конфіденційності, цілісності і неможливості відмови від авторства***



ITSEC – критерії ефективності



□ **Критерії ефективності**

- Відповідність набору засобів захисту заданим цілям
- Взаємна узгодженість різних засобів і механізмів захисту
- Здатність засобів захисту протистояти атакам
- Можливість практичного використання недоліків архітектури засобів захисту
- Простота використання засобів захисту
- Можливість практичного використання функціональних недоліків засобів захисту



ITSEC – критерії коректності

☐ **Критерії коректності**

- Процес розроблення
- Середовище розроблення
- Експлуатаційна документація
- Середовище експлуатації



Висновки по ITSEC

122

- ☐ ***ITSEC тісно пов'язані з TCSEC (не повністю самостійний документ)***
- ☐ ***Головне досягнення – введення поняття гарантій захисту і визначення окремої шкали для критеріїв гарантій***
- ☐ ***Відмова від єдиної ієрархічної шкали оцінювання систем***
- ☐ ***Визнання можливості наявності недоліків у сертифікованих системах і введення критерію можливості використання недоліків захисту***



FCITS – «Федеральні критерії»: мета розробки

123

□ *Federal Criteria for Information Technology Security. – National Institute of Standards and Technology & National Security Agency. Version 1.0, 1992.*

- Визначення універсального й відкритого для подальшого розвитку набору основних вимог безпеки, що висувають до сучасних інформаційних технологій
- Удосконалення вимог і критеріїв безпеки, що існують
- Узгодження між собою вимог і критеріїв безпеки інформаційних технологій, що прийняті у різних країнах
- Нормативне закріплення головних принципів інформаційної безпеки



FCITS – етапи розроблення

124

- 1. Розроблення і аналіз *профілю захисту***
- 2. Розроблення і кваліфікаційний аналіз продуктів ІТ**
- 3. Компонування і сертифікація системи оброблення інформації в цілому**



FCITS – структура профілю

125

☐ **Опис**

- Класифікаційна інформація, що необхідна для ідентифікації профілю у спеціальній картотеці

☐ **Обґрунтування**

- Опис середовища експлуатації, загроз безпеці, що передбачаються, і методів використання продукту ІТ

☐ **Функціональні вимоги до продукту ІТ**

☐ **Вимоги до технології розроблення продукту ІТ**

☐ **Вимоги до процесу кваліфікаційного аналізу продукту ІТ**



FCITS – функціональні вимоги

126

☐ **Реалізація політики безпеки**

- Політика аудиту
 - ◆ Ідентифікація і автентифікація
 - ◆ Реєстрація в системі
 - ◆ Забезпечення прямої взаємодії з КЗЗ
 - ◆ Реєстрація і облік подій
- Політика керування доступом
 - ◆ Дискреційне керування доступом
 - ◆ Мандатне керування доступом
 - ◆ Контроль прихованих каналів
- Політика забезпечення працездатності
 - ◆ Контроль за розподілом ресурсів
 - ◆ Забезпечення стійкості до відмов
- Керування безпекою

☐ **Моніторинг взаємодій**

☐ **Логічний захист КЗЗ**

☐ **Фізичний захист КЗЗ**

☐ **Самоконтроль КСЗ**

☐ **Ініціалізація і відновлення КЗЗ**

☐ **Обмеження привілеїв під час роботи з КЗЗ**

☐ **Простота використання КЗЗ**



FCITS – Вимоги до технології розробки

❑ *Процес розробки*

- Визначення множини функцій КЗЗ у відповідності до функціональних вимог
- Реалізація КЗЗ
- Гарантії реалізації КЗЗ
- Тестування і аналіз КЗЗ

❑ *Середовище розробки*

- Інструментальні засоби
- Засоби керування процесом розробки
- Процедура дистрибуції



FCITS – Вимоги до технології розробки

128

□ **Документація**

- Документація функцій КЗЗ
- Повна документація на продукт ІТ (інтерфейси, компоненти, модулі, структура КЗЗ, методика проектування, вихідні тексти і специфікація апаратних засобів)
- Документація тестування і аналізу продукту ІТ
 - ◆ Документація процесу тестування функцій
 - ◆ Документація аналізу можливостей порушення безпеки
 - ◆ Документація аналізу прихованих каналів
- Документація середовища і процесу розробки

□ **Супроводження**

- Документація користувача
- Настанова з адміністрування системи безпеки
- Процедура оновлення версій і виправлення помилок
- Процедура інсталяції



FCITS – Вимоги до процесу кваліфікаційного аналізу

☐ **Аналіз**

- Аналіз архітектури
- Аналіз реалізації

☐ **Контроль**

- Контроль середовища розробки
- Контроль процесу супроводження продукту ІТ

☐ **Тестування**

- Тестування функцій КЗЗ виробником
- Незалежне тестування функцій КЗЗ



FCITS – Висновки

130

- ❑ ***Вперше запропонована концепція профілю захисту***
- ❑ ***У стандарті визначені три незалежні групи вимог***
 - Функціональні вимоги безпеки добре структуровані
 - Вимоги до технології розроблення змушують виробників застосовувати сучасні технології програмування, що дозволяють підтвердити безпеку продукту
 - Вимоги до процесу кваліфікаційного аналізу узагальнені і не містять конкретних методик
- ❑ ***Відсутня загальна оцінка рівня безпеки за допомогою універсальної шкали, запропоновано незалежне ранжування вимог кожної групи***

Донецький національний технічний університет



Тема 5. Нормативно-правове регулювання в Україні.

Лекція 5. Нормативно-правове регулювання в Україні

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Нормативно-правове регулювання в Україні



Мета лекції

- Розглянути основні нормативно-правові акти, що регулюють сферу інформаційного захисту в Україні



НД ТЗІ з питань захисту інформації в комп'ютерних системах від НСД₁

134

- Загальні положення із захисту інформації в комп'ютерних системах від несанкціонованого доступу (НД ТЗІ 1.1-002-99)
- Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу (НД ТЗІ 1.1-003-99)
- Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу (НД ТЗІ 2.5-004-99)
- Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу (НД ТЗІ 2.5-005-99)
- Методичні вказівки по розробці технічного завдання на створення комплексної системи захисту інформації в автоматизованій системі (НД ТЗІ 3.7-001-99)
- Типове положення про службу захисту інформації в автоматизованій системі (НД ТЗІ 1.4-001-2000)



НД ТЗІ з питань захисту інформації в комп'ютерних системах від НСД₂

135

- **Технічний захист інформації. Комп'ютерні системи. Порядок створення, впровадження, супроводження і модернізації засобів технічного захисту інформації від несанкціонованого доступу. (НД ТЗІ 3.6-001-2000)**
- **Вимоги із захисту конфіденційної інформації від несанкціонованого доступу під час оброблення в автоматизованих системах класу 2 (НД ТЗІ 2.5-008-02)**
- **Вимоги до захисту інформації WEB–сторінки від несанкціонованого доступу (НД ТЗІ 2.5-010-03)**
- **Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі (НД ТЗІ 3.7-003-05)**
- **Методичні вказівки з оцінювання функціональних послуг безпеки в засобах захисту інформації від несанкціонованого доступу (НД ТЗІ 2.7-009-09)**
- **Методичні вказівки з оцінювання рівня гарантій коректності реалізації функціональних послуг безпеки в засобах захисту інформації від несанкціонованого доступу (НД ТЗІ 2.7-010-09)**



Термінологія в галузі захисту інформації в КС від НСД₁

~~Суб'єкт~~ → Об'єкт-користувач
Об'єкт-процес

Доступ:

користувач → процес → пасивний об'єкт

Керування доступом:

~~Дискреційне~~ - - - - -> Довірче

~~Мандатне~~ - - - - -> Адміністративне



Термінологія в галузі захисту інформації в КС від НСД₂

- ☐ ***Довірче керування доступом***
- ☐ ***Адміністративне керування доступом***



Критерії оцінки захищеності інформації в КС від НСД

138

☐ ***Критерії конфіденційності***

☐ ***Критерії цілісності***

☐ ***Критерії доступності***

☐ ***Критерії спостережності***

☐ ***Критерії гарантій***

Функціональні
критерії



Концепція функціонального профілю в НД ТЗІ

139

□ Послуга (сервіс) безпеки

- Сукупність функцій, що забезпечують захист від певної загрози або від множини загроз
- Для кожної послуги запропонована окрема шкала оцінок, яка визначає рівень реалізації цієї послуги

□ Функціональний профіль

- Перелік рівнів функціональних послуг, які реалізуються комп'ютерною системою
- Функціональний профіль оформлюється певним чином
 - зокрема, специфіковано порядок, у якому повинні бути названі послуги



Критерії конфіденційності

140

❑ **Довірча конфіденційність (КД-1...4)**

- дозволяє користувачу керувати потоками інформації від захищених об'єктів, що належать його домену, до інших користувачів

❑ **Адміністративна конфіденційність (КА-1...4)**

- дозволяє адміністратору або спеціально авторизованому користувачу керувати потоками інформації від захищених об'єктів до користувачів



Критерії конфіденційності₂

141

❑ **Повторне використання об'єктів (КО-1)**

- дозволяє забезпечити коректність повторного використання розділюваних об'єктів, тобто таких, які по чергово виділяються різним користувачам та/або процесам
- гарантує, що коли розділюваний об'єкт виділяється новому користувачу або процесу, він не містить інформації, яка залишилась від попереднього користувача або процесу

❑ **Аналіз прихованих каналів (КК-1...3)**

- забезпечує виявлення і усунення потоків інформації, які існують, але не контролюються іншими послугами

❑ **Конфіденційність при обміні (КВ-1...4)**

- забезпечує захист об'єктів від несанкціонованого ознайомлення з інформацією, яку вони містять, під час їх передавання через незахищене середовище



Критерії цілісності

142

❑ **Довірча цілісність (ЦД-1...4)**

- дозволяє користувачу керувати потоками інформації від інших користувачів до захищених об'єктів, що належать його домену

❑ **Адміністративна цілісність (ЦА-1...4)**

- дозволяє адміністратору або спеціально авторизованому користувачу керувати потоками інформації від користувачів до захищених об'єктів

❑ **Відкат (ЦО-1...2)**

- забезпечує можливість відмінити операцію або послідовність операцій і повернути (відкотити) захищений об'єкт до попереднього стану

❑ **Цілісність при обміні (ЦВ-1...3)**

- забезпечує захист об'єктів від несанкціонованої модифікації інформації, яку вони містять, під час їх передавання через незахищене середовище



Критерії доступності

143

☐ **Використання ресурсів (ДР-1...3)**

- забезпечує керування використанням користувачами послуг і ресурсів

☐ **Стійкість до відмов (ДС-1...3)**

- гарантує доступність КС (можливість використання інформації, окремих функцій або КС в цілому) після відмови її компонента

☐ **Гаряча заміна (ДЗ-1...3)**

- дозволяє гарантувати доступність КС (можливість використання інформації, окремих функцій або КС в цілому) в процесі заміни окремих компонентів

☐ **Відновлення після збоїв (ДВ-1...3)**

- забезпечує повернення КС у відомий захищений стан після відмови або переривання обслуговування



Критерії спостережності₁



☐ **Реєстрація (НР-1...5)**

- дозволяє контролювати небезпечні для КС дії шляхом реєстрації і аналізу подій, що мають відношення до безпеки

☐ **Ідентифікація й автентифікація (НИ-1...3)**

- дозволяє КЗЗ визначити і перевірити особистість користувача, що намагається одержати доступ до КС

☐ **Достовірний канал (НК-1...2)**

- дозволяє гарантувати користувачу можливість безпосередньої взаємодії з КЗЗ

☐ **Розподіл обов'язків (НО-1...3)**

- дозволяє зменшити потенційні збитки від навмисних або помилкових дій користувача і обмежити авторитарність керування

☐ **Цілісність комплексу засобів захисту (НЦ-1...3)**

- визначає міру здатності КЗЗ захищати себе і гарантувати свою спроможність керувати захищеними об'єктами



Критерії спостережності₂



☐ **Самотестування (НТ-1...3)**

- дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій КС

☐ **Ідентифікація й автентифікація при обміні (НВ-1...3)**

- дозволяє одному КЗЗ ідентифікувати інший КЗЗ (встановити і перевірити його ідентичність) і забезпечити іншому КЗЗ можливість ідентифікувати перший, перш ніж почати взаємодію

☐ **Автентифікація відправника (НА-1...2)**

- дозволяє забезпечити захист від відмови від авторства і однозначно встановити належність певного об'єкта певному користувачу, тобто той факт, що об'єкт був створений або відправлений даним користувачем

☐ **Автентифікація одержувача (НП-1...2)**

- дозволяє забезпечити захист від відмови від одержання і дозволяє однозначно встановити факт одержання певного об'єкта певним користувачем



Критерії гарантій

☐ **Архітектура**

☐ **Середовище розроблення:**

- процес розроблення
- керування конфігурацією

☐ **Послідовність розроблення:**

- функціональні специфікації (політика безпеки)
- функціональні специфікації (модель політики безпеки)
- проект архітектури
- детальний проект
- реалізація

☐ **Середовище функціонування**

☐ **Документація**

☐ **Випробування комплексу засобів захисту**



Обов'язковість послуг

147

КД-1. Мінімальна довірча конфіденційність	КД-2. Базова довірча конфіденційність	КД-3. Повна довірча конфіденційність	КД-4. Абсолютна довірча конфіденційність
Політика довірчої конфіденційності, що реалізується КЗЗ, повинна визначати множину об'єктів КС, до яких вона відноситься		Політика довірчої конфіденційності, що реалізується КЗЗ, повинна відноситись до всіх об'єктів КС	
КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу			
процесу і захищеного об'єкта	користувача і захищеного об'єкта		користувача, процесу і захищеного об'єкта
Запити на зміну прав доступу до об'єкта повинні оброблятися КЗЗ на підставі атрибутів доступу користувача, що ініціює запит, і об'єкта			
КЗЗ повинен надавати користувачу можливість для кожного захищеного об'єкта, що належить його домену, визначити			
конкретні процеси і/або групи процесів, які мають право одержувати інформацію від об'єкта	конкретних користувачів і/або групи користувачів, які мають право одержувати інформацію від об'єкта	конкретних користувачів (і групи користувачів), які мають, а також тих, які не мають права одержувати інформацію від об'єкта	конкретних користувачів і процеси (і групи користувачів і процесів), які мають, а також тих, які не мають права одержувати інформацію від об'єкта
—	КЗЗ повинен надавати користувачу можливість для кожного процесу, що належить його домену, визначити		
	конкретних користувачів і/або групи користувачів, які мають право ініціювати процес	конкретних користувачів (і групи користувачів), які мають, а також тих, що не мають права ініціювати процес	
Права доступу до кожного захищеного об'єкта повинні встановлюватись в момент його створення або ініціалізації. Як частина політики довірчої конфіденційності повинні бути представлені правила збереження атрибутів доступу об'єктів під час їх експорту та імпорту			
НЕОБХІДНІ УМОВИ: НИ-1		НЕОБХІДНІ УМОВИ: КО-1, НИ-1	



Обмеження на функціональний профіль

148

- ☐ **Обов'язково має виконуватись послуга “Цілісність комплексу засобів захисту” (щонайменше, рівень НЦ-1)**
- ☐ **Мають бути задоволені усі вимоги специфікацій послуг щодо виконання інших послуг**
 - КД, КА, ЦД, ЦА, ЦО, НР, НО, НА, НП → НИ-1 (ідентифікація, автентифікація)
 - КА, ЦА, КВ-2-4, ЦВ-2,3, ДР, ДС, ДЗ, ДВ, НР-2,5, НЦ-1, НТ-2 → НО-1 (виділення адміністратора)
 - КД-3,4, КА-3,4, ЦД-3,4, ЦА-3,4, КК → КО-1 (повторне використання об'єктів)
 - КК-2, КВ-4, НЦ-1 → НР-1 (реєстрація)
 - КВ-3,4, ЦВ-3 → НВ-1 (ідентифікація й автентифікація при обміні)
 - ДЗ-2,3 → ДС-1 (стійкість до відмов)
 - НИ-2,3 → НК-1 (достовірний канал)
 - КК, КВ-4 → Г-3



Класифікація АС і стандартні функціональні профілі ЗІ в КС

149

☐ **Клас 1: одномашинний однокористувачевий комплекс**

- Окрема робоча станція, що не підключена до мережі

☐ **Клас 2: локалізований багатомашинний багатокористувачевий комплекс**

- Локальна мережа
- Багатотермінальний сервер
- Кілька комп'ютерів, що не поєднані у мережу, але знаходяться у спільному приміщенні і виконують спільні завдання

☐ **Клас 3: розподілений багатомашинний багатокористувачевий комплекс**

- Головна ознака – принципова неможливість повного контролю території, на якій знаходиться АС



Семантика профілю

150

☐ **Опис профілю складається з трьох частин:**

- буквено-числового ідентифікатора
- знака рівності
- переліку рівнів послуг, взятого в фігурні дужки

☐ **Ідентифікатор у свою чергу включає:**

- позначення класу АС (1, 2 або 3)
- буквену частину, що характеризує види загроз, від яких забезпечується захист (К, і/або Ц, і/або Д)
- номер профілю і необов'язкове буквене позначення версії

☐ **Всі частини ідентифікатора відділяються одна від одної крапкою**

- Наприклад, 2.К.4 — функціональний профіль номер чотири, що визначає вимоги до АС класу 2, призначених для обробки інформації, основною вимогою щодо захисту якої є забезпечення конфіденційності

☐ **Версія може служити, зокрема, для вказівки на підсилення певної послуги всередині профілю**

- Наприклад, нарощування можливостей реєстрації приведе до появи нової версії

☐ **Внесення деяких істотних змін, особливо додання нових послуг, може або привести до появи нового профілю, або до того, що профіль буде відноситись до іншого класу чи підкласу АС**



Приклади стандартних функціональних профілів захищеності

151

- **1.Ц.2 = { ЦА-2, ЦО-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1 }**
- **1.Д.2 = { ДР-2, ДС-1, ДЗ-1, ДВ-2, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-2 }**
- **1.Д.3 = { ДР-2, ДС-2, ДЗ-2, ДВ-2, НР-3, НИ-2, НК-1, НО-1, НЦ-2, НТ-2 }**
- **1.КЦ.2 = { КА-1, КО-1, ЦА-2, ЦО-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1 }**
- **2.КЦ.2 = { КД-2, КО-1, ЦД-1, ЦО-1, НР-2, НИ-2, НК-1, НО-1, НЦ-2, НТ-1 }**
- **2.КЦД.2 = { КД-2, КА-2, КО-1, ЦД-1, ЦА-2, ЦО-1, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2 }**
- **3.КЦД.2 = { КД-2, КА-2, КО-1, КВ-2, ЦД-1, ЦА-2, ЦО-1, ЦВ-2, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }**



НД ТЗІ 2.5-008-02

152

- ☐ **Повна назва документа - Вимоги із захисту конфіденційної інформації від несанкціонованого доступу під час оброблення в автоматизованих системах класу 2**
- ☐ **Мета документа – надання нормативно-методологічної бази під час розроблення комплексів засобів захисту від НСД до конфіденційної інформації, яка обробляється в АС класу 2, створення КСЗІ, проведення аналізу та оцінки захищеності інформації від несанкціонованого доступу в системах такого класу, а також рекомендацій для визначення необхідного функціонального профілю захищеності інформації в конкретній АС**
- ☐ **В документі наведені:**
 - загальні вимоги до захисту конфіденційної інформації, які впливають з вимог чинного законодавства
 - характеристики типових умов функціонування АС класу 2. Розглянуті обчислювальна система, фізичне середовище, в якому вона знаходиться і функціонує, користувачі АС та оброблювана інформація, у тому числі й технологія її оброблення.
 - детальні вимоги щодо захисту інформації в АС класу 2
 - докладно розібрана політика реалізації послуг безпеки інформації в АС класу 2
- ☐ **Фактично, документ є детальною настановою з розроблення технічного завдання на створення КСЗІ в АС класу 2**



Політика безпеки згідно НД ТЗІ 2.5-008-02

153

- ❑ **Як основний принцип політики безпеки визначено адміністративний принцип розмежування доступу, який забезпечується послугами безпеки КА і ЦА**
- ❑ **Реалізація послуг безпеки, що базуються на довірчому принципі розмежування доступу (КД і ЦД), може здійснюватися у випадках:**
 - коли політикою безпеки передбачено створення груп користувачів з однаковими повноваженнями щодо роботи з конфіденційною інформацією для розмежування доступу до об'єктів, що таку інформацію містять, у межах цих груп
 - для розмежування доступу до об'єктів, які потребують захисту, але не містять конфіденційної інформації
- ❑ **Визначені такі стандартні функціональні профілі захищеності оброблюваної інформації:**
 - 2.К.3 = {КД-2, КА-2, КО-1, НР-2, НК-1, НЦ-2, НТ-2, НИ-2, НО-2}
 - 2.КЦ.3 = {КД-2, КА-2, КО-1, ЦД-1, ЦА-2, ЦО-1, НР-2, НК-1, НЦ-2, НТ-2, НИ-2, НО-2}
 - 2.КД.1а = {КД-2, КА-2, КО-1, ДР-1, ДС-1, ДЗ-1, ДВ-1, НР-2, НК-1, НЦ-2, НТ-2, НИ-2, НО-2}
 - 2.КЦД.2а = {КД-2, КА-2, КО-1, ЦД-1, ЦА-2, ЦО-1, ДР-1, ДС-1, ДЗ-1, ДВ-1, НР-2, НК-1, НЦ-2, НТ-2, НИ-2, НО-2}
- ❑ **Мінімальним достатнім рівнем гарантій реалізації КЗЗ АС класу 2 є рівень Г–2**
- ❑ **Реалізація послуги безпеки КО-1 є обов'язковою у тому випадку, коли в АС класу 2 усі користувачі допущені до обробки конфіденційної інформації одного рівня**



НД ТЗІ 2.5-010-03

154

- ❑ **Повна назва документа – Вимоги до захисту інформації WEB–сторінки від несанкціонованого доступу**
- ❑ **Документ встановлює вимоги до технічних та організаційних заходів захисту інформації Web-сторінки (Web-сайту) в мережі Інтернет**
 - Зокрема, мінімально необхідний перелік послуг безпеки інформації та рівнів їх реалізації у комплексах засобів захисту інформації Web-сторінки від несанкціонованого доступу згідно з визначеними НД ТЗІ 2.5-004-99 специфікаціями
- ❑ **Мета документа – надання нормативно-методологічної бази для розроблення комплексу засобів захисту від несанкціонованого доступу до інформації Web- сторінки під час створення КСЗІ**
- ❑ **В документі наведені:**
 - загальні вимоги до захисту інформації Web-сторінки, які впливають з вимог чинного законодавства
 - актуалізація розміщених на Web-сторінці інформаційних ресурсів та керування доступом до них здійснюється за допомогою АС, в якій повинна бути створена КСЗІ
 - характеристики типових умов функціонування такої АС
 - детальні вимоги щодо захисту інформації Web-сторінки
 - докладно розібрана політика реалізації послуг безпеки інформації в АС, що забезпечує функціонування Web-сторінки



Категорії інформації Web-сторінок згідно НД ТЗІ 2.5-010-03

155

☐ **Інформація Web-сторінки поділяється на дві категорії: загальнодоступна і технологічна**

- Загальнодоступна – інформація, що розміщується на Web-сторінці з метою надання можливості користування нею будь-яким фізичним або юридичним особам, які мають доступ до мережі Інтернет
- Технологічна інформація Web-сторінки – технологічна інформація КСЗІ та технологічна інформація щодо адміністрування та управління обчислювальною системою АС і засобами обробки інформації
 - ☐ дані про мережні адреси,
 - ☐ імена, персональні ідентифікатори та паролі користувачів,
 - ☐ їхні повноваження та права доступу до об'єктів,
 - ☐ інформація журналів реєстрації дій користувачів,
 - ☐ інша інформація баз даних захисту,
 - ☐ встановлені робочі параметри окремих механізмів або засобів захисту,
 - ☐ інформація про профілі обладнання та режими його функціонування,
 - ☐ робочі параметри функціонального ПЗ тощо.

☐ **Розміщення на Web-сторінці конфіденційної інформації діючим законодавством не припускається**

- Якщо інформація не є власністю держави або інформацією з обмеженим доступом, вимога щодо захисту якої встановлена законом, то власник інформації може запроваджувати правила доступу до неї на свій розсуд, але при цьому вимоги НД ТЗІ 2.5-010-03 не дотримуються



Технології доступу до інформації згідно НД ТЗІ 2.5-010-03

156

☐ **Web-сторінка може бути розміщена**

- на території власника інформації
- на території сторонньої організації (наприклад, оператора мережі доступу до Інтернет)
 - Власник інформації, виходячи з чинного законодавства, визначає правила доступу до інформації
 - Власник системи здійснює захист інформації, зокрема, забезпечуючи відповідне розмежування доступу до інформації

☐ **Доступ до технологічної інформації та передавання даних для актуалізації загальнодоступної інформації може здійснюватись у два способи:**

- Технологія T1: з робочої станції, розміщеної на тій самій території, що і Web-сервер (установи-власника Web-сторінки або оператора) або з консолі Web-сервера
- Технологія T2: з робочої станції, яка розміщена на території установи-власника Web-сторінки, до Web-сервера, що розміщений на території оператора, з використанням мереж передачі даних

☐ **Технологія T2 відрізняється від технології T1:**

- наявністю незахищеного середовища, яке не контролюється
- додатковими вимогами щодо ідентифікації та автентифікації між K33 робочої станції й K33 Web-сервера під час спроби розпочати обмін інформацією
- додатковими вимогами щодо забезпечення цілісності інформації при обміні



Функціональні профілі захищеності згідно НД ТЗІ 2.5-010-03

157

- ☐ ***Коли доступ до технологічної інформації та передавання даних для актуалізації загальнодоступної інформації здійснюється за технологією T1, мінімально необхідний функціональний профіль визначається:***
 - {КА-2, ЦА-1, ЦО-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1}
- ☐ ***Коли передбачається можливість доступу до технологічної інформації та/або передавання даних для актуалізації загальнодоступної інформації за технологією T2, мінімально необхідний функціональний профіль визначається:***
 - {КА-2, КВ-1, ЦА-1, ЦО-1, ЦВ-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1, НВ-1}
- ☐ ***Мінімальним достатнім рівнем гарантій реалізації K33 Web-сторінки є рівень Г–2***



Оцінювання за критеріями

158

☐ ***Передбачено дві процедури:***

■ ***Сертифікація засобів захисту***

- Порядок проведення робіт із сертифікації засобів забезпечення технічного захисту інформації загального призначення
 - Наказ Держстандарту України та Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 9.07.2001 № 329/32

■ ***Державна експертиза КСЗІ***

- Положення про державну експертизу в сфері технічного захисту інформації
 - Наказ Адміністрації Державної служби спеціального зв'язку та захисту інформації України від 16.05.2007 № 93

☐ ***Реально побудована система проведення державної експертизи, як для КСЗІ, так і для засобів захисту***

Донецький національний технічний університет



Тема 6. Стандарт ISO/IEC 15408.

Лекція 6. ISO/IEC 15408 (Common Criteria)

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Нормативно-правове регулювання в Україні



Мета лекції

- Розглянути основні нормативно-правові акти, що регулюють сферу інформаційного захисту в Україні



Історія розробки₁

- ❑ 1990 рік: Міжнародна організація із стандартизації (ISO) розпочинає роботи із створення стандарту у сфері оцінки безпеки інформаційних технологій (ІТ).
- ❑ Червень 1993 року: організації із стандартизації та забезпечення безпеки США, Канади, Великої Британії, Франції, Німеччини та Нідерландів об'єднали свої зусилля в рамках проекту із створення єдиної сукупності критеріїв оцінки безпеки ІТ
- ❑ 1996 рік: Завершена розробка версії 1.0 Загальних критеріїв
- ❑ 1998 рік: Оприлюднена версія 2.0 ЗК
- ❑ 1999 рік: На основі версії 2.0 ЗК прийнято міжнародний стандарт ISO/IEC 15408



Історія розробки₂

163

- ❑ Вже після прийняття стандарту з урахуванням досвіду його використання з'явився ряд інтерпретацій ЗК
- ❑ 2005 рік: Версія 2.2 ЗК стала основою для внесення змін у стандарт ISO
- ❑ Протягом певного часу паралельно проводили урахування інтерпретацій (була завершена версія 2.4 ЗК і розроблювали версію 2.6 ЗК) і вели розробку версії 3.0 ЗК, яка має дещо змінену (спрощену) структуру вимог
- ❑ 2008 рік: в якості стандарту ISO/IEC 15408:2008 ухвалили версію 3.0 ЗК
- ❑ Методологія застосування ЗК оформлена у вигляді окремого документа “Загальна методологія оцінювання безпеки інформаційних технологій” і також набула статусу стандарту, діюча версія: ISO/IEC 18045:2008



Основні відомості

164

- ❑ У застосуванні до оцінки безпеки продуктів інформаційних технологій (ПІТ) стандарт задає систему понять, в термінах яких повинна проводитись оцінка
- ❑ Стандарт містить відносно повний каталог вимог безпеки (функціональних та гарантій), але не надає конкретних наборів вимог та критеріїв для тих чи інших типів **об'єктів оцінювання (Target of Evaluation, TOE)**, виконання яких необхідно перевіряти
- ❑ Вимоги та критерії формуються у **профілях захисту (Protection Profile, PP)** та **завданнях з безпеки (Security Target, ST)**
- ❑ Офіційно прийняті профілі захисту утворюють побудовану на основі ЗК нормативну базу, що використовується на практиці у сфері інформаційної безпеки



Структура Загальних Критеріїв

165

ЗК є сукупністю самостійних, але взаємопов'язаних частин:

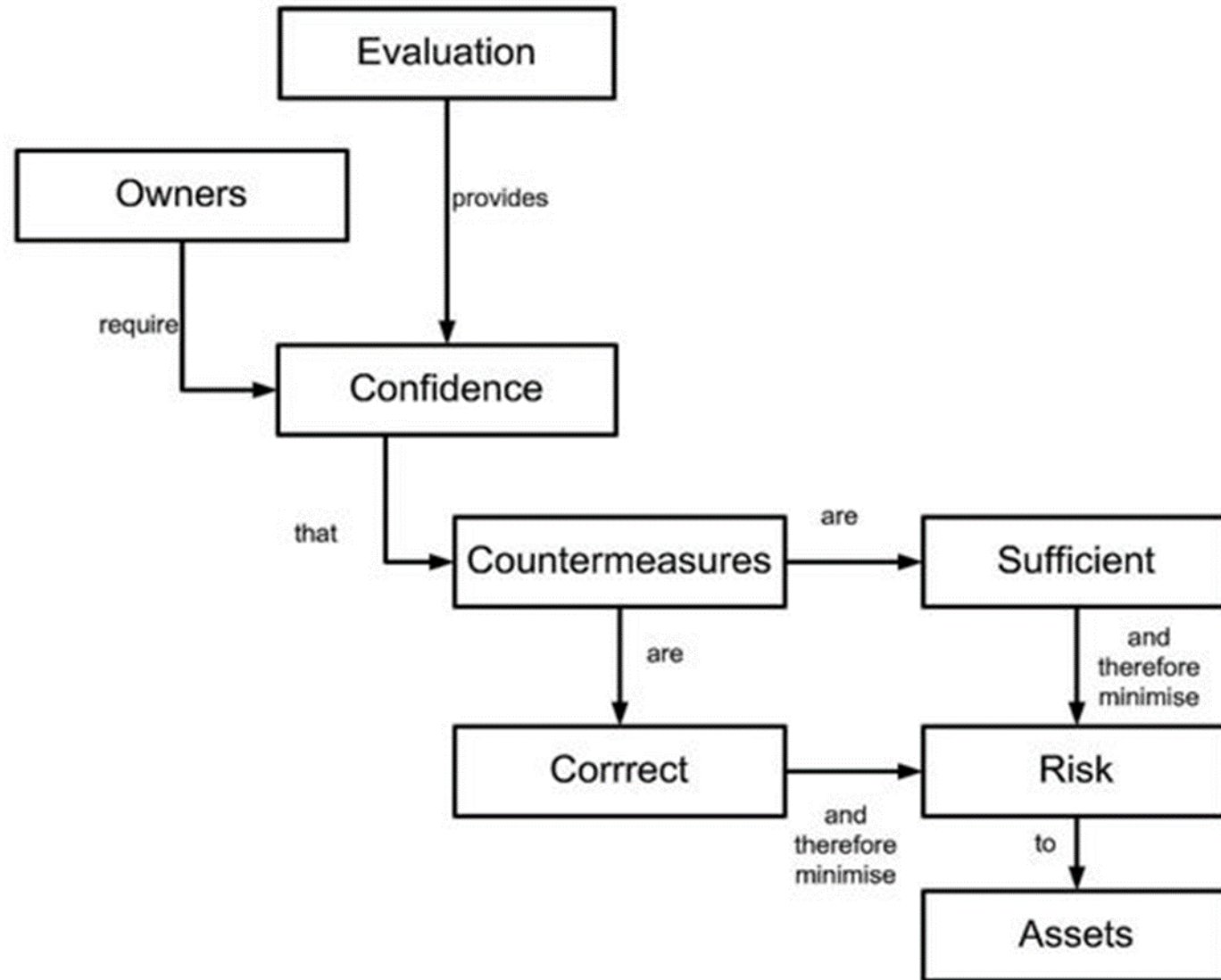
- ❑ Розділ **Представлення і загальна модель** – визначає загальну концепцію й принципи оцінки безпеки ІТ і подає загальну модель оцінки, а також конструкції для:
 - формування задач захисту ІТ
 - вибору й визначення вимог безпеки ІТ
 - опису специфікацій високого рівня для продуктів і систем
- ❑ Розділ **Вимоги до функцій безпеки** – встановлює набір функціональних компонентів як стандартний шлях формулювання функціональних вимог до об'єктів оцінки
- ❑ Розділ **Вимоги гарантій безпеки** – включає компоненти вимог гарантій оцінки, а також рівні гарантій оцінки, які визначають ранжирування за ступенем задоволення вимог

У версії 1.0 був передбачений розділ **Визначені профілі захисту**, що містив приклади профілів захисту, які включали функціональні вимоги безпеки та вимоги гарантій оцінки, що були ідентифіковані у діючих на той час стандартах (**ITSEC, CTCPEC, FC, TCSEC**), а також інші профілі. Цей розділ був вилучений із ЗК починаючи з версії 2.0 (тобто, до складу стандарту ISO не увійшов)



Поняття з оцінювання та їх відношення

166





Базові поняття

- ❑ **Об'єкт оцінювання (*Target of Evaluation, TOE*)**
- ❑ **Задачі захисту (*Security Objectives*)**
- ❑ **Профіль захисту (*Protection Profile*)**
- ❑ **Завдання з безпеки (*Security Target*)**



Три стадії процесу кваліфікаційного аналізу

168

1. Аналіз Профілю захисту на предмет

- його повноти,
- несуперечності,
- можливості реалізації та
- можливості використання як набору вимог для ОО

2. Аналіз Завдання з безпеки на предметПерелік рівнів функціональних послуг, які реалізуються комп'ютерною системою

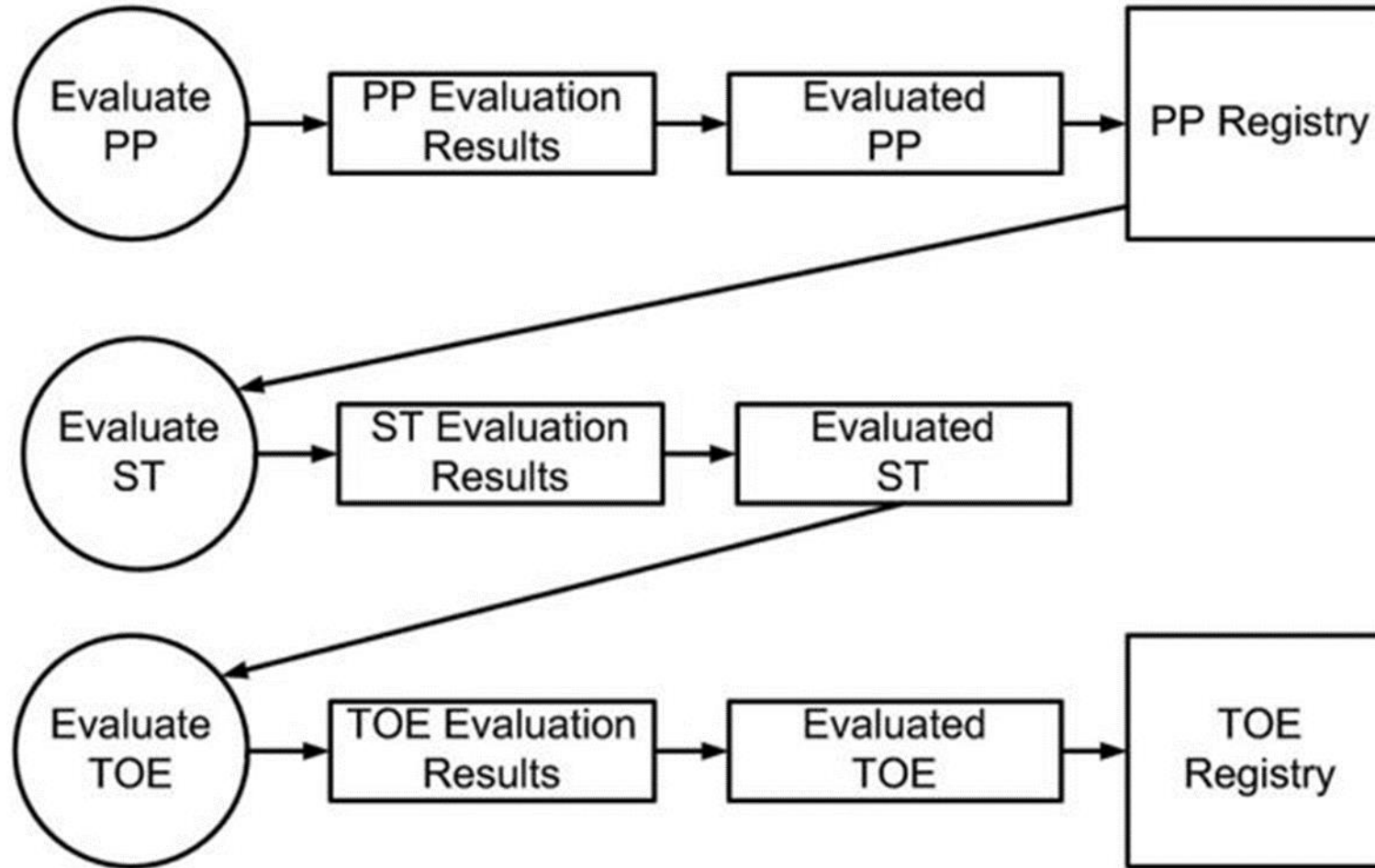
- його відповідності вимогам профілю захисту,
- повноти,
- несуперечності,
- можливості реалізації та
- можливості використання як опису ОО

3. Аналіз ОО на предмет відповідності Завданню з безпеки



Процес кваліфікаційного аналізу

169





Структура вимог

Функціональні вимоги (частина 2) та вимоги гарантій (частина 3) подані в одному спільному стилі та використовують одну і ту саму організацію та термінологію:

☐ ***Клас***

☐ ***Сімейство***

☐ ***Компонент***

☐ ***Елемент***



Набори структур

171

- ☐ **ЗК визначають набори структур, що поєднують компоненти вимог безпеки**
- ☐ **Пакет** — це проміжна комбінація компонентів
- ☐ **Рівні гарантій оцінки (EAL)** – це визначені пакети вимог гарантій
- ☐ **Профіль захисту (Protection Profile)**
- ☐ **Завдання з безпеки (Security Target)**



Перетворення компонентів

172

- ☐ ***Компоненти можуть бути перетворені за допомогою дозволених дій, щоби забезпечити виконання певної політики безпеки або протистояти певній загрозі***
- ☐ ***До дозволених дій належать:***
 - Призначення
 - Вибір
 - Обробка



Таксономія вимог

Функціональні вимоги – класи₁

- ❑ **FAU** – Аудит безпеки
- ❑ **FCO** – Інформаційний обмін
- ❑ **FCS** – Криптографічна підтримка
- ❑ **FDP** – Захист інформації користувача
- ❑ **FIA** – Ідентифікація й автентифікація
- ❑ **FMT** – Керування безпекою
- ❑ **FPR** – Конфіденційність доступу до системи



Таксономія вимог

Функціональні вимоги – класи₂

- ☐ **FPT** – *Захист функцій безпеки*
- ☐ **FRU** – *Контроль за використанням ресурсів*
- ☐ **FTA** – *Контроль доступу до системи*
- ☐ **FTP** – *Забезпечення прямої взаємодії*



Таксономія вимог

Вимоги гарантій – класи

- ☐ **APE** – Оцінка профілю захисту
- ☐ **ASE** – Оцінка завдання з безпеки
- ☐ **ADV** – Розробка
- ☐ **AGD** – Документація
- ☐ **ALC** – Підтримка життєвого циклу
- ☐ **ATE** – Тестування
- ☐ **AVA** – Оцінка вразливості
- ☐ **ACO** – Інтеграція



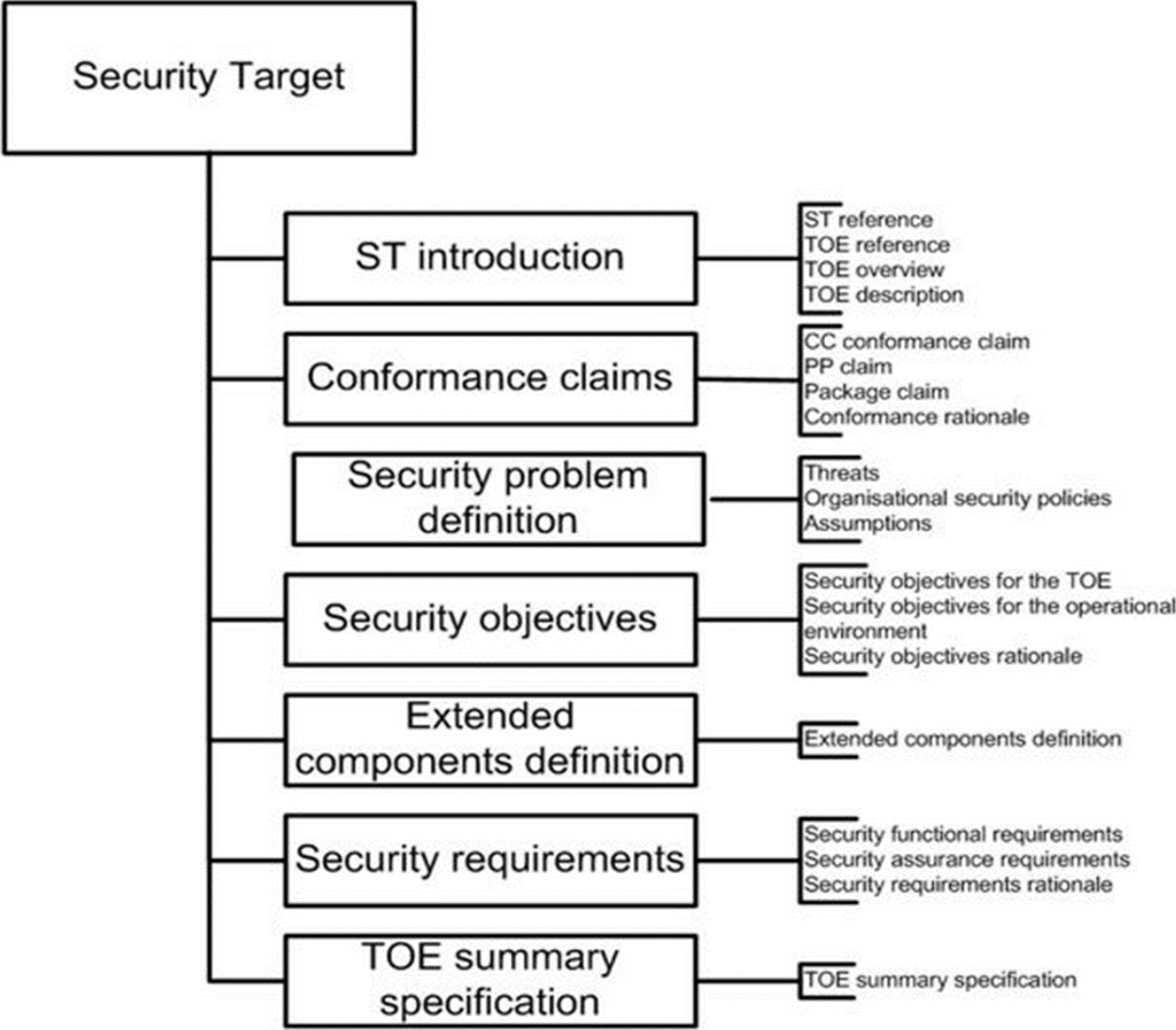
Обов'язковий зміст Завдання з безпеки

□ ***Завдання з безпеки повинно містити:***

- вступну частину;
- заявку на відповідність;
- визначення проблеми безпеки (**a security problem definition**);
- задачі захисту (**security objectives**);
- (опціонально) визначення розширених компонентів;
- вимоги безпеки;
- підсумкову специфікацію ОО.



Зміст «Завдання з безпеки»





Структура «Завдання з безпеки» згідно ЗК версії 2

- ☐ ***Вступ***
- ☐ ***Опис ОО***
- ☐ ***Задачі захисту***
- ☐ ***Вимоги безпеки***
- ☐ ***Загальні специфікації ОО***
- ☐ ***Обґрунтування***



Структура «Завдання з безпеки» згідно ЗК версії 3

- ☐ ***Вступ***
- ☐ ***Заявка на відповідність***
- ☐ ***Визначення проблеми безпеки***
- ☐ ***Вимоги безпеки***
- ☐ ***Задачі захисту***
- ☐ ***Визначення розширених компонентів***
- ☐ ***Вимоги безпеки***
- ☐ ***Підсумкова специфікація ОО***



Як має використовуватись «Завдання з безпеки»

180

Завдання з безпеки виконує дві функції:

- ☐ ***До і під час кваліфікаційного аналізу, Завдання з безпеки визначає “що має бути оцінено”***
- ☐ ***Після завершення кваліфікаційного аналізу, Завдання з безпеки визначає “що було оцінено”***



Як «Завдання з безпеки» використовуватись не повинно

Дві функції (з багатьох), які Завдання з безпеки не повинно мати:

- ☐ ***Детальної специфікації***
- ☐ ***Повної специфікації***



Обов'язковий зміст профілю захисту

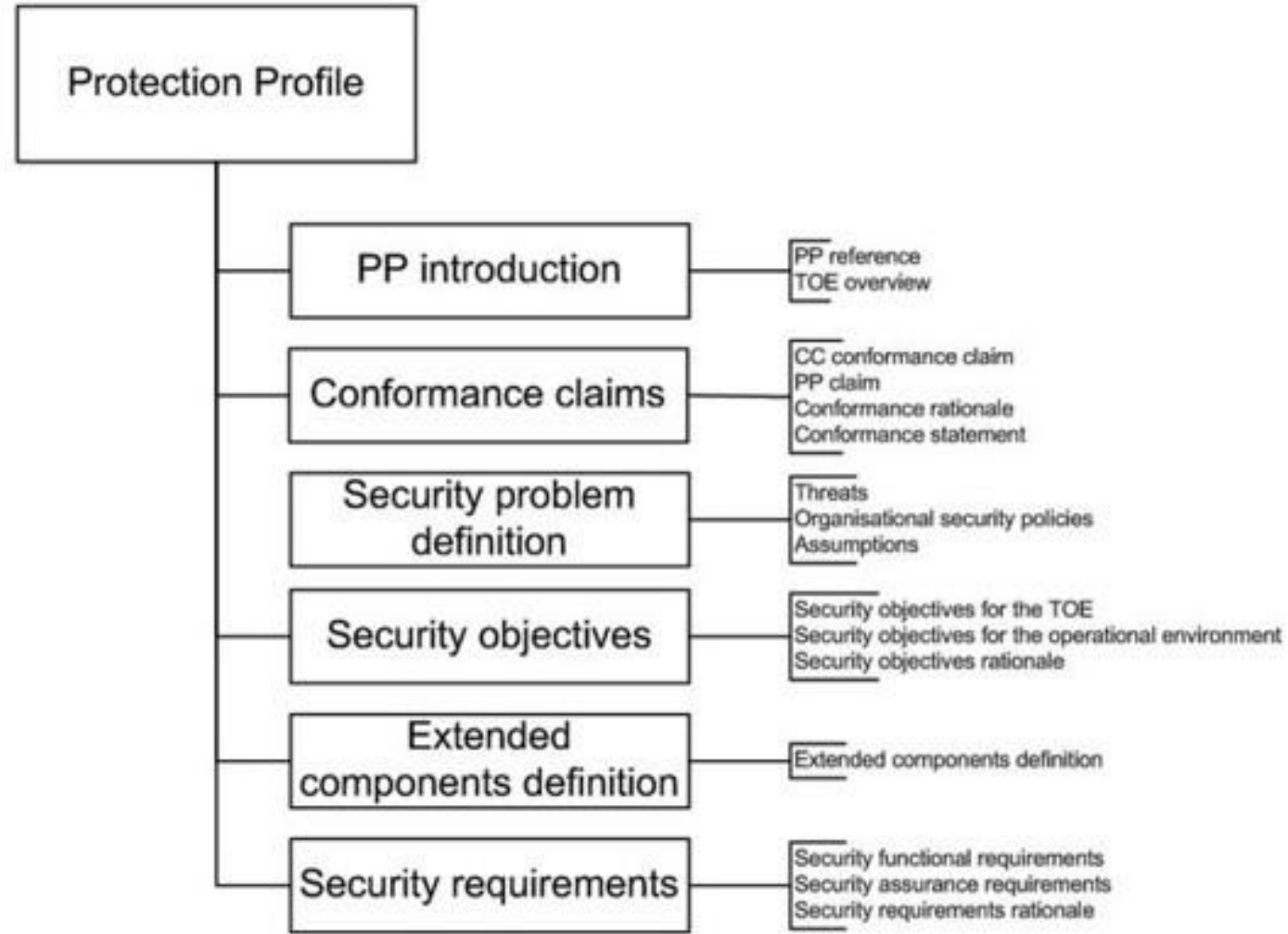
Профіль захисту містить:

- ☐ ***вступну частину;***
- ☐ ***заявку на відповідність;***
- ☐ ***визначення проблеми безпеки;***
- ☐ ***задачі захисту;***
- ☐ ***визначення розширених компонентів;***
- ☐ ***вимоги безпеки.***



Зміст Профілю захисту

183





Структура профілю захисту згідно ЗК версії 2

- ☐ ***Вступ***
- ☐ ***Опис ОО***
- ☐ ***Середовище експлуатації***
- ☐ ***Задачі захисту***
- ☐ ***Функціональні вимоги***
- ☐ ***Вимоги гарантій***
- ☐ ***Вимоги до середовища експлуатації***
- ☐ ***Додаткові відомості***
- ☐ ***Обґрунтування***



Структура профілю захисту гідно ЗК версії 3

- ☐ ***Вступ***
- ☐ ***Заявка на відповідність***
- ☐ ***Визначення проблеми безпеки***
- ☐ ***Задачі захисту***
- ☐ ***Визначення розширених компонентів***
- ☐ ***Вимоги безпеки***



Як має використовуватись «Профіль захисту»

186

- ☐ ***Зазвичай, Профіль захисту є формулюванням потреб, коли співтовариство користувачів, регуляторний орган, або група розробників визначають загальний набір потреб захисту***
- ☐ ***Профіль захисту надає споживачам засіб посилення на такий набір, і полегшує подальшу оцінку на відповідність цим потребам***
- ☐ ***Таким чином, Профіль захисту зазвичай використовують як:***
 - ☐ частину специфікацій вимог конкретного споживача або групи споживачів, які розглядатимуть придбання певного типу продуктів ІТ лише якщо він задовольняє вимогам цього профілю захисту;
 - ☐ частину директивних вимог певного регуляторного органу, який дозволить використання певного типу продуктів ІТ якщо він задовольняє вимогам цього профілю захисту;
 - ☐ деякі базові вимоги, визначені групою розробників ІТ, які домовляються, що усі продукти ІТ визначеного типу, що вони виробляють, будуть задовольняти цим вимогам,

хоча це не заперечує інші можливості застосування.



Як профіль захисту використовуватись не повинен

187

Три функції (з багатьох), які Профіль захисту виконувати не повинен:

- ☐ ***бути детальною специфікацією***
- ☐ ***бути повною специфікацією***
- ☐ ***бути специфікацією одного продукту***



Роль сертифікації

- ☐ ***Сертифікація за Загальними критеріями не може гарантувати безпеку***
- ☐ ***Натомість, сертифікація може гарантувати, що заяви щодо властивостей безпеки оціненого продукту були перевірені незалежно***



Критика

189

❑ У різних джерелах формулювали такі зауваження до процесу і методології оцінювання за Загальними критеріями:

- ❑ Оцінювання – дорогий процес**
- ❑ Різні рівні**
- ❑ Для підготовки оцінювання необхідно багато часу**
- ❑ Є прив'язка до каскадної моделі ПЗ**
- ❑ Немає належного контролю за ПЗ після сертифікації**

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 7. Модель загроз Windows.

Компоненти K33 Windows.

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Загальний опис профілю захисту Windows
- Модель загроз Windows
- Опис K33 Windows
- Аудит безпеки Windows



Мета лекції

- Ознайомитися з загальним описом профілю захисту Windows
- Розглянути модель загроз Windows
- Розглянути компоненти K33 Windows
- Ознайомитися з методами аудиту безпеки Windows



Надійні джерела

193

❑ Windows Internals, 7th Edition by Mark E. Russinovich, Andrea Allievi, Alex Ionescu, David A. Solomon

- Windows Internals, Seventh Edition, Part 1: System architecture, processes, threads, memory management, and more by Pavel Yosifovich, Alex Ionescu, Mark E. Russinovich, and David A. Solomon. - Microsoft Press, 2017. - ISBN: 978-0-7356-8418-8
- Windows Internals, Seventh Edition, Part 2 by Andrea Allievi, Alex Ionescu, Mark E. Russinovich, and David A. Solomon. - Microsoft Press, 2021. - ISBN: 978-0-13-546240-9
- Найдетальніший опис того, як Windows влаштована і працює
- Фактично, це декларація можливостей Windows від розробника

❑ Microsoft Windows 10, Windows Server (April 2018 Update) Security Target

- Документ, підготовлений фахівцями Microsoft для сертифікації чергової версії системи за стандартом ISO/IEC 15408:2008
- Фактично, це декларація можливостей Windows з урахуванням необхідності підтвердження заявлених функцій у ході випробувань незалежними експертами
- З іншого боку, Security Target описує лише функції безпеки, і не є ані повною, ані детальною специфікацією



Коротка характеристика

194

TOE Overview за Security Target

☐ Об'єкт оцінювання включає:

- Операційну систему Windows 10,
- Операційну систему Windows Server,
- Програми, необхідні для керування, підтримки та конфігурування операційної системи

☐ Windows 10 і Windows Server можуть постачатись попередньо встановленими на новому комп'ютері або можуть бути скачані з вебсайту Microsoft

☐ Усі редакції Windows 10 і Windows Server, що узагальнено називають "Windows", є багатозадачними (з підтримкою витісняльної — **preemptive** багатозадачності), багатопроцесорними і багатокористувацькими операційними системами



Використання Windows

- ☐ Windows 10 призначена для десктопів, ноутбуків і конвертованих комп'ютерів
- ☐ Windows Server побудований для навантажень у діапазоні від відділу до підприємства і до хмарних сервісів
- ☐ Windows надає інтерактивний інтерфейс користувача — User Interface (UI) — а також мережний інтерфейс



Визначення проблеми

згідно Security Target

- ❑ Визначення проблеми безпеки складається з:
 - Загроз безпеці
 - Організаційних політик безпеки
 - (не визначені для GP OS PP та WLAN EP)
 - Припущень щодо використання
- ❑ Припущення, загрози та політики скопійовані з:
 - the General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (“GP OS PP”)
 - the General Purpose Operating Systems Protection Profile/ Mobile Device Fundamentals Protection Profile Extended Package (EP) Wireless Local Area Network (WLAN) Clients (“WLAN EP”)



Модель загроз для Windows

- ☐ **T.NETWORK_ATTACK** (*Мережна атака*)
- ☐ **T.NETWORK_EAVESDROP** (*Прослуховування мережі*)
- ☐ **T.LOCAL_ATTACK** (*Локальна атака*)
- ☐ **T.LIMITED_PHYSICAL_ACCESS** (*Обмежений фізичний доступ*)



Додаткова модель загроз для Windows (WLAN EP Threats)

- ☐ ***T.TSF_FAILURE (Відмова K33)***
- ☐ ***T.UNAUTHORIZED_ACCESS (Несанкціонований доступ)***
- ☐ ***T.UNDETECTED_ACTIONS (Невиявлені дії)***



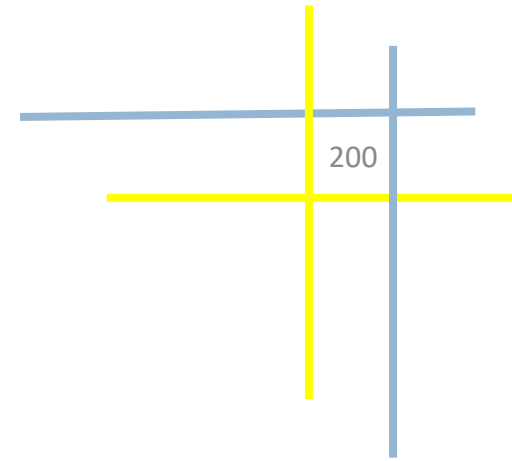
Припущення щодо безпечного використання Windows

- ☐ A.PLATFORM (Платформа)
- ☐ A.PROPER_USER (Належний користувач)
- ☐ A.PROPER_ADMIN (Належний адміністратор)



Припущення WLAN EP щодо безпечного використання Windows

- ☐ A.NO_TOE_BYPASS (Немає обходу Windows)
- ☐ A.TRUSTED_ADMIN (Довірений адміністратор)





Компоненти КЗЗ Windows₁

201

Згідно “Windows Internals”

- ❑ Монітор безпеки (**Security Reference Monitor, SRM**) - Ntoskrnl.exe
- ❑ Підсистема локальної безпеки (**Local Security Authority Subsystem Service, Lsass**) - Lsass.exe, Lsassrv.dll
- ❑ LSAIso.exe (Відомий також як **Credential Guard**)
- ❑ База даних політики Lsass (HKLM\SECURITY)



Компоненти K33 Windows₂

Згідно “Windows Internals”

- ☐ Диспетчер облікових записів безпеки (**Security Accounts Manager, SAM**) - Samsrv.dll
- ☐ База даних SAM (HKLM\SAM)
- ☐ Active Directory - Ntdsa.dll
- ☐ Пакети автентифікації



Компоненти K33 Windows₃

203

Згідно “Windows Internals”

- ☐ Інтерактивний диспетчер входу в систему (**Winlogon**)
- ☐ Користувацький інтерфейс входу в систему — Logon user interface (**LogonUI**) - LogonUI.exe
- ☐ Постачальники облікових даних — **Credential providers (CP)** - authui.dll, SmartcardCredentialProvider.dll, Bio.Cred.Prov.dll та FaceCredentialProvider.dll
- ☐ Служба мережного входу до системи (**Netlogon**) - Netlogon.dll



Компоненти K33 Windows₄

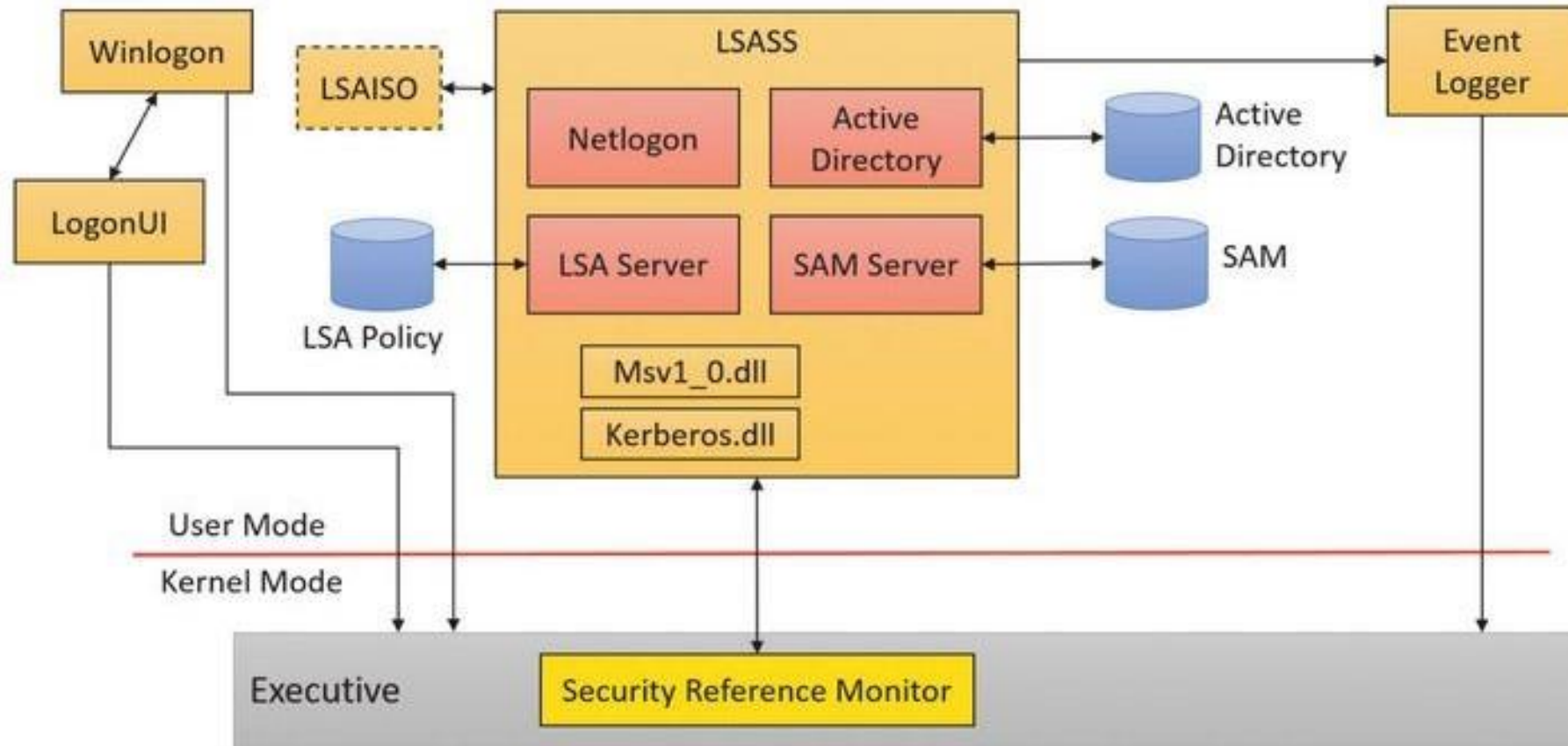
Згідно “Windows Internals”

- ❑ Kernel Security Device Driver (**KSecDD**) - Ksecdd.sys
- ❑ AppLocker - AppId.sys, AppIdSvc.dll



Взаємодія компонентів й БД системи безпеки

205



Ілюстрація з *Windows Internals, Part 1, 7th edition*

Укладач: проф. Дорогий Я.Ю.



Захист на основі віртуалізації

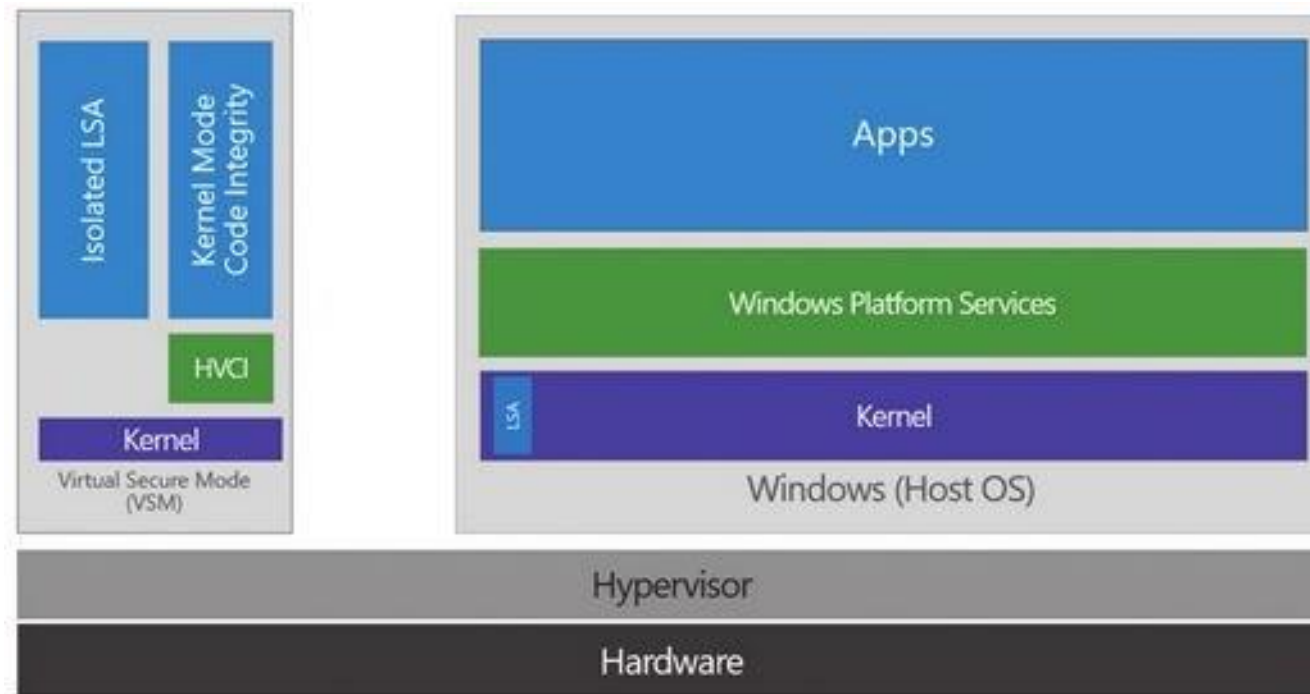
206

- ❑ Windows 10 and Server 2016 включають архітектуру безпеки на основі віртуалізації (**Virtualization-Based Security, VBS**), що забезпечує додатковий ортогональний рівень віртуального захисту (**virtual trust level, VTL**)
- ❑ Звичайний код режиму користувача і код ядра виконуються в **VTL 0**, і нічого не знають про існування **VTL 1**
- ❑ **Credential Guard і Device Guard** підтримують VTLs для захисту даних користувача і надання додаткового рівня безпеки на основі обладнання для цифрового підписування коду
- ❑ **Kernel Patch Protection (KPP)** надається компонентом PatchGuard і підсилюється технологією **HyperGuard** на основі VBS



Архітектура безпеки на основі віртуалізації VBS

207



На схемі показані основні компоненти VBS:

- **Hypervisor-Based Code Integrity (HVCI)** та **Kernel-Mode Code Integrity (KMCI)**, які є основою для **Device Guard**
- **LSA (Lsass.exe)** та **isolated LSA (Lsalso.exe)**, які є основою для **Credential Guard**

Ілюстрація з Windows Internals, Part 1, 7th edition



K33 Windows₁

208

згідно Security Target

- ❑ Концептуально Windows можна розглядати як набір сервісів безпеки
- ❑ Ці сервіси надаються компонентами Windows:
 - Диспетчер завантаження (**Boot Manager**), який активується кодом завантаження комп'ютера
 - Завантажувач Windows (**Windows Loader**), який завантажує операційну систему в пам'ять комп'ютера
 - Ядро Windows (**Windows Kernel**), що містить драйвери файлової системи, шифрування розділів (**full volume encryption**), фільтр аварійного дампу (**crash dump filter**), і криптографічну бібліотеку режиму ядра
 - Стек IPv4 / IPv6 в ядрі
 - Захищений інсталятор Windows (**Windows Trusted Installer**), який встановлює оновлення операційної системи Windows



K33 Windows₂

209

згідно Security Target

- ☐ Підсистема локальної автентифікації (**Local Security Authority Subsystem**), яка ідентифікує та автентифікує користувачів перед їх входом у систему і генерує події для журналу аудиту безпеки
- ☐ Криптографічні алгоритми, ухвалені FIPS (**FIPS- Approved cryptographic algorithms**), для захисту даних користувача і системи
- ☐ Служба захисту ключів (**Key Isolation Service**), яка захищає таємні та приватні ключі
- ☐ Локальні та віддалені інтерфейси адміністрування (**administrative interfaces**) для управління безпекою
- ☐ **Windows Explorer**, який може використовуватись для управління операційною системою та перевірки цілісності файлів і оновлень Windows



Сервіси безпеки Windows (TOE Security Services)

згідно Security Target

- ☐ Аудит безпеки (Security Audit)
- ☐ Криптографічна підтримка (Cryptographic Support)
- ☐ Захист даних користувача (User Data Protection)
- ☐ Ідентифікація й автентифікація (Identification and Authentication)
- ☐ Захист КЗЗ (Protection of the TOE Security Functions)
- ☐ Блокування сеансу (Session Locking)
- ☐ Доступ до ОО (TOE Access)
- ☐ Захищений канал взаємодії (Trusted Path for Communications)
- ☐ Управління безпекою (Security Management)



Аудит безпеки

☐ ***Цей сервіс здійснює:***

- Збирання даних аудиту
- Селективний аудит
- Захист журналу аудиту від переповнення
- Захист обмеженого доступу до журналу аудиту



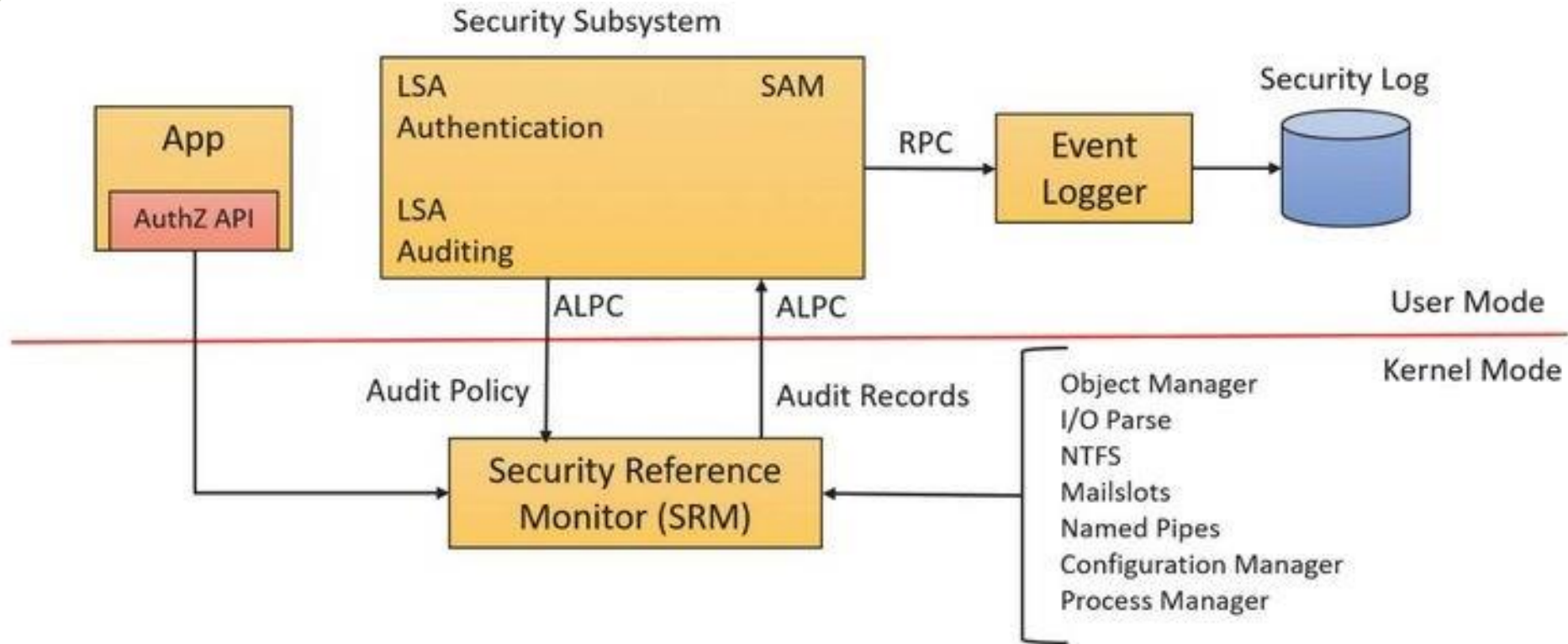
Збирання даних аудиту

- ☐ Служба **Windows Event Log** створює журнал подій безпеки (**security event log**)
- ☐ Сервер **Local Security Authority** (LSA)
- ☐ Обмежень на розмір окремого запису аудиту немає



Потік даних аудиту

213



Ілюстрація з Windows Internals, Part 1, 7th edition



Дані аудиту

214

- ☐ ***Кожний запис аудиту містить такі стандартні поля:***
 - ☐ Дата події;
 - ☐ Час події;
 - ☐ SID користувача, причетного до події;
 - ☐ Ідентифікатор події (унікальний для кожної категорії);
 - ☐ Джерело — компонент Windows, що згенерував подію;
 - ☐ Результат (чи є подія результатом успішної чи неуспішної спроби);
 - ☐ Категорія
- ☐ ***Кожний запис аудиту може також містити дані, специфічні для категорії цієї події, що містяться у тілі запису***



Категорії подій

Служба LSA визначає такі категорії подій аудиту в журналі безпеки:

- ☐ Система (**System**)
- ☐ Вхід / Вихід (**Logon / Logoff**)
- ☐ Доступ до об'єкта (**Object Access**)
- ☐ Доступ до служби каталогів (**Directory Service Access**)
- ☐ Використання привілеїв (**Privilege Use**)
- ☐ Детальне відстеження процесу (**Detailed Process Tracking**)
- ☐ Зміна політики (**Policy Change**)
- ☐ Управління обліковими записами (**Account Management**)
- ☐ Вхід в обліковий запис (**Account Logon**)



Де збирають події аудиту безпеки

216

В КЗЗ є два місця, де збирають події аудиту безпеки:

1) В ядрі монітор безпеки (**Security Reference Monitor — SRM**), що є частиною виконавчої системи (**NT Executive**), відповідає за генерування усіх подій аудиту таких категорій:

- Доступ до об'єктів
- Використання привілеїв
- Детальне відстеження процесу

Компоненти Windows можуть запросити **SRM** згенерувати запис аудиту і надати усі елементи цього запису крім системного часу, який надає виконавча система

2) За одним винятком, події аудиту для інших категорій подій генерують різні служби, які

- Або співіснують в сервері **LSA**,
- Або, маючи привілей **SeAuditPrivilege**, викликають інтерфейси **AuthZ Report Audit**, імплементовані в субкомпоненті **LSA Policy**

☐ Винятком є те, що сама служба **Event Log Service** створює запис події, коли

- Здійснюють очищення журналу безпеки
- Або журнал безпеки перевищує рівень попередження, встановлений авторизованим адміністратором



Політика аудиту₁

- ❑ Сервер LSA підтримує в своїй базі даних політику аудиту, що визначає, які категорії подій повинні збиратися
 - Визначення і модифікація політики аудиту дозволені лише авторизованому адміністратору
- ❑ Єдиний інший компонент КЗЗ, що застосовує політику аудиту — це SRM, який веде аудит доступу до об'єктів, використання привілеїв, і детального відстеження процесів
 - LSA і SRM мають спільний приватний порт зв'язку, який застосовується для того, щоби передати SRM політику аудиту



Політика аудиту₂

218

- ❑ Додатково до визначення політики аудиту для усієї системи, можливо визначити політику аудиту для окремих користувачів (per-user audit policy), використовуючи **auditpol.exe**
 - Це дозволяє включати або відключати окремі категорії подій аудиту для окремих користувачів
 - Windows не дозволить локальному адміністратору відключити аудит для облікових записів локальних адміністраторів
- ❑ У кожній категорії аудит може здійснюватись для подій успішних, неуспішних, або усіх
- ❑ Для подій доступу до об'єктів аудит може далі контролюватись на основі ідентифікації користувача/групи і правах доступу, використовуючи системні списки керування доступом (**System Access Control Lists — SACL**), що асоційовані з об'єктами

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 8. Криптографічна підтримка та захист даних користувача у Windows

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Мета лекції
- Криптографічна підтримка у Windows
- Захист даних користувача у Windows



Мета лекції

- Розглянути питання криптографічної підтримки у Windows
- Розглянути основні механізми захисту даних користувача у Windows



Криптографічна підтримка

- ☐ Криптографічні алгоритми та операції
- ☐ Валідація криптографічних алгоритмів
- ☐ Мережний обмін (**TLS**)
- ☐ Захист даних за допомогою **DPAPI**



Огляд функцій₁

☐ Windows надає перевірені FIPS 140-2 CAVP криптографічні функції, які підтримують:

- шифрування / дешифрування,
- криптографічні підписи,
- криптографічне хешування,
- договір криптографічного ключа, і
- генерацію випадкових чисел

☐ Windows додатково забезпечує:

- підтримку відкритих ключів,
- управління обліковими даними та функції перевірки сертифікатів,
- підтримку криптографічних алгоритмів Агенції національної безпеки (NSA) Suite B



Огляд функцій₂

☐ Windows також надає:

- широку аудиторську підтримку криптографічних операцій,
- можливість замінити криптографічні функції та генератори випадкових чисел альтернативними реалізаціями,
- послугу ізоляції ключів, призначену для обмеження потенційного розкриття секретних та приватних ключів.

☐ Крім використання криптографії для власних функцій безпеки, Windows пропонує доступ до функцій криптографічної підтримки програм користувача та режиму ядра

☐ Сертифікати відкритих ключів, створені та використані Windows, перевіряють автентичність користувачів та машин, а також захищають як дані користувачів, так і системи, під час передавання мережею



Cryptography API: Next Generation

225

- ❑ API криптографії наступного покоління (**CNG**) розроблений як розширюваний на багатьох рівнях та агностичний для наборів криптографічних алгоритмів
- ❑ Важливою особливістю **CNG** є його власна реалізація алгоритмів **Suite B**
- ❑ Такі протоколи, як **Internet Key Exchange (IKE)** та **Transport Layer Security (TLS)**, використовують еліптичну криву Діффі-Хеллмана (**ECDH**), включену в **Suite B**, а також функції хешування



Детермінована генерація випадкових бітів

- ❑ Детермінована генерація випадкових бітів (**Deterministic Random Bit Generation — DRBG**) реалізована відповідно до Спеціальної публікації NIST 800-90
- ❑ Windows має різні джерела ентропії (детерміновані та недетерміновані), які виробляють дані ентропії, які використовуються для генерації випадкових чисел



Джерела ентропії в Windows

- ☐ Час переривань
- ☐ TPM (англ. Trusted Platform Module)
- ☐ RDRAND/RDSEED
- ☐ Seed файл
- ☐ Зовнішня ентропія
- ☐ ACPI-OEMO
- ☐ Дані з драйверів
- ☐ UEFI
- ☐ Відмітка про час старту системи
- ☐ Унікальне (не випадкове) число Hyper-V



Пул ентропії

228

❑ Пул ентропії (Entropy Pool) заповнюється з використанням таких значень:

- ☐ Початкове значення ентропії із насінневого файлу, що надається завантажувачу ОС Windows під час завантаження (512 біт ентропії)
 - ☐ Розраховане значення на основі лічильника циклів процесора з високою роздільною здатністю, який спрацьовує після кожних 1024 переривань (безперервне джерело, що забезпечує 16384 біти ентропії).
 - ☐ Випадкові значення, що періодично збираються з модуля довіреної платформи (TPM), (320 біт ентропії при завантаженні, 384 біти після цього на вимогу на основі таймера ОС).
 - ☐ Випадкові значення періодично збираються, викликаючи команду процесора RDRAND, (256 біт ентропії).
- ❑ Кожне джерело ентропії не залежить від інших джерел і не залежить від часу**



Захист криптографічних функцій

229

- ❑ K33 Windows захищає від фальсифікації джерел генерації випадкових чисел (**Random Number Generation — RNG**) / генерації псевдовипадкових чисел (**Pseudorandom Number Generation — PRNG**), інкапсулюючи їх використання в драйвері пристрою безпеки ядра — **Kernel Security Device Driver**
- ❑ Постачальником **CNG** для генерації випадкових чисел є **AES_CTR_DRBG**, коли Windows вимагає використання солі, вона використовує Windows **RBG**
- ❑ Операції шифрування та дешифрування виконуються незалежними модулями, відомими як постачальники криптографічних послуг (**Cryptographic Service Providers — CSP**)



Інші криптографічні операції

- ☐ На додаток до послуг шифрування та дешифрування, TSF надає інші криптографічні операції
- ☐ Хешування використовують інші алгоритми, схвалені FIPS, реалізовані в Windows
- ☐ Коли Windows потребує встановлення спільного секретного ключа на основі RSA, він може виступати як відправник, так і одержувач



Стандарти криптографічних алгоритмів Windows та методи оцінки₁

231

Криптографічна операція	Стандарт	Метод оцінки
Шифрування/Дешифрування	FIPS 197 AES	NIST CAVP # 5847, # 5859, # 5860, # 5861
	NIST SP 800-38A CBC mode	# 5847, # 5861
	NIST SP 800-38C CCM mode	# 5847, # 5859
	NIST SP 800-38E XTS mode	# 5847
	NIST SP 800-38F KW mode	# 5860
Цифровий підпис (генерація ключів)	NIST SP 800-38D GCM mode	# 5847
	FIPS 186-4 RSA	NIST CAVP # 3079, # 3080
	FIPS 186-4 RSA	NIST CAVP # 3079, # 3080, # 3082
	FIPS 186-4 RSA	NIST CAVP # 3079, # 3080, # 3081, # 3082
	FIPS 186-4 DSA	NIST CAVP # 1479



Стандарти криптографічних алгоритмів Windows та методи оцінки₂

232

Криптографічна операція	Стандарт	Метод оцінки
Цифровий підпис (генерація ключів)	FIPS 186-4 ECDSA	NIST CAVP # 1563, # 1564, # 1565
Цифровий підпис (генерація ключів, генерація і перевірка підпису)	FIPS 186-4 ECDSA	NIST CAVP # 1563, # 1564
Хешування	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP # 4633
Код автентифікації повідомлень на основі хешу з ключем	FIPS 198-2 HMAC	NIST CAVP # 3858, # 3859
Генерація випадкових чисел	NIST SP 800-90 CTR_DRBG	NIST CAVP # 2435, # 2436
Узгодження ключів	NIST SP 800-56A ECDH	NIST CAVP # 200, # 201
Встановлення ключів	NIST SP 800-56B RSA	NIST CVL # 21 1, # 21 2
Виведення ключа на основі ключа	SP800-108	NIST CAVP # 242, # 243
IKEv1, IKEv2, TLS	SP800-135	NIST CVL # 21 0



Сховище ключів

- ☐ CNG включає послугу ізоляції ключів у режимі користувача, розроблену спеціально для розміщення секретних та приватних ключів у захищеному процесі для зменшення фальсифікації або доступу до важливих матеріалів ключів для процесів у режимі користувача
- ☐ CNG виконує перевірку виявлення помилок ключа при кожній передачі ключа
- ☐ CNG запобігає архівуванню (приватних) ключів підпису, термін дії яких завершився, та знищує непостійні криптографічні ключі



Типи ключів, що використовує Windows₁

234

Ключі	Опис
Symmetric encryption/decryption keys	Keys used for AES (FIPS 197) encryption/decryption for IPsec ESP, TLS, Wi-Fi.
HMAC keys	Keys used for HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512 (FIPS 198-1) as part of IPsec
Asymmetric ECDSA Public Keys	Keys used for the verification of ECDSA digital signatures (FIPS 186-4) for IPsec traffic and peer authentication.
Asymmetric ECDSA Private Keys	Keys used for the calculation of ECDSA digital signatures (FIPS 186-4) for IPsec traffic and peer authentication.
Asymmetric RSA Public Keys	Keys used for the verification of RSA digital signatures (FIPS 186-4) for IPsec, TLS, Wi-Fi and signed product updates.
Asymmetric RSA Private Keys	Keys used for the calculation of RSA digital signatures (FIPS 186-4) for IPsec, TLS, and Wi-Fi as well as TPM-based health attestations. The key size can be 2048 or 3072 bits.



Типи ключів, що використовує Windows₂

235

Ключі	Опис
Asymmetric DSA Private Keys	Keys used for the calculation of DSA digital signatures (FIPS 186-4) for IPsec and TLS. The key size can be 2048 or 3072 bits.
Asymmetric DSA Public Keys	Keys used for the verification of DSA digital signatures (FIPS 186-4) for IPsec and TLS. The key size can be 2048 or 3072 bits.
DH Private and Public values	Private and public values used for Diffie-Hellman key establishment for TLS.
ECDH Private and Public values	Private and public values used for EC Diffie-Hellman key establishment for TLS.
DPAPI master secret	512-bit random value used by DPAPI
DPAPI master AES key	256-bit encryption key that protects the DPAPI master secret
DPAPI AES key	256-bit encryption key used by DPAPI
DRBG seed	The seed for the main DRBG, zeroized during reseeding



Мережний обмін (TLS)

236

- ❑ Windows реалізує TLS як надійний мережний шлях, який використовується для автентифікації клієнта та сервера, а також HTTPS
- ❑ Протоколи описані в документах:
 - Профіль безпеки транспортного рівня (TLS) MS-TLSP
 - RFC 2246 Протокол TLS, версія 1.0
 - RFC 3268 AES набори шифрів для TLS
 - RFC 3546 Розширення безпеки транспортного рівня (TLS)
 - RFC 4366 Розширення безпеки транспортного рівня (TLS)
 - RFC 4492 ECC набори шифрів для TLS
 - RFC 4681 TLS Розширення картографування користувачів
 - RFC 5246 - Протокол безпеки транспортного рівня (TLS), версія 1.2
 - RFC 5289 - Набори TLS ECC із SHA-256/384 та AES GCM



TLS RFCs впроваджені в Windows

237

RFC#	Назва	Як використовується
2246	The TLS Protocol Version 1.0	Specifies requirements for TLS 1.0.
3268	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)	Specifies additional ciphersuites implemented by Windows.
3546	Transport Layer Security (TLS) Extensions	Updates RFC 2246 with TLS 1.0 extensions implemented by Windows.
4346	The Transport Layer Security (TLS) Protocol Version 1.1	Specifies requirements for TLS 1.1.
4366	Transport Layer Security (TLS) Extensions	Obsoletes RFC 3546 Requirements for TLS 1.1 extensions implemented by Windows.
4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)	Specifies additional ciphersuites implemented by Windows.
4681	TLS User Mapping Extension	Extends TLS to include a User Principal Name during the TLS handshake.
5246	The Transport Layer Security (TLS) Protocol Version 1.2	Obsoletes RFCs 3268, 4346, and 4366. Specifies requirements for TLS 1.2.
5289	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)	Specifies additional ciphersuites implemented by Windows.
SSL3	The SSL Protocol Version 3	Specifies requirements for SSL3.



Особливості реалізації TLS₁

238

- ☐ Під час узгодження набору шифрів еліптичної кривої TLS 1.2 Windows автоматично включатиме як частину повідомлення клієнта Hello
- ☐ З іншого боку, за замовчуванням алгоритмами підпису в повідомленні клієнта Hello є: SHA1, SHA256, SHA384 та SHA512

Ілюстрація з Windows Internals, Part 1, 7th edition

Укладач: проф. Дорогий Я.Ю.



Особливості реалізації TLS₂

239

- ❑ Кожен компонент Windows, який використовує TLS, перевіряє, чи ідентифікаційна інформація у сертифікаті відповідає очікуваному.
- ❑ Ідентифікатор DN та будь-яке альтернативне ім'я суб'єкта у сертифікаті перевіряється на наявність ідентифікації DNS- запису або IP-адреси віддаленого комп'ютера, щоб переконатися, що вони збігаються, як описано в [http://technet.microsoft.com/en-us/library/cc783349\(v=WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc783349(v=WS.10).aspx) зокрема розділ «Повідомлення про сертифікат сервера»

Ілюстрація з Windows Internals, Part 1, 7th edition

Укладач: проф. Дорогий Я.Ю.



Особливості реалізації TLS₃

240

- ❑ Референсним ідентифікатором у Windows для TLS є ім'я DNS або IP-адреса віддаленого сервера, яке порівнюється з іменем DNS у вигляді представленого ідентифікатора в альтернативному імені суб'єкта (SAN) або імені суб'єкта сертифіката
- ❑ Сертифікат, який використовує шаблон у крайній лівій частині ідентифікатора ресурсу (тобто *.contoso.com), може бути прийнятий для автентифікації, інакше сертифікат буде визнано недійсним
- ❑ Windows реалізує HTTPS, як описано в RFC 2818, так що Магазин Windows та системні програми, що виконуються на ОО, можуть надійно підключатися до зовнішніх серверів за допомогою HTTPS



Бездротова мережа

241

- ❑ Windows має власні реалізації IEEE 802.11-2012 та IEEE 802.11ac-2013 для забезпечення безпечних бездротових локальних мереж (Wi-Fi)
- ❑ Windows реалізує обгортання та розгортання ключів (wrapping and unwrapping) відповідно до специфікації NIST SP 800-38F (режим «KW») і таким чином розгортає тимчасовий ключ групи Wi-Fi (GTK), який було надіслано точкою доступу

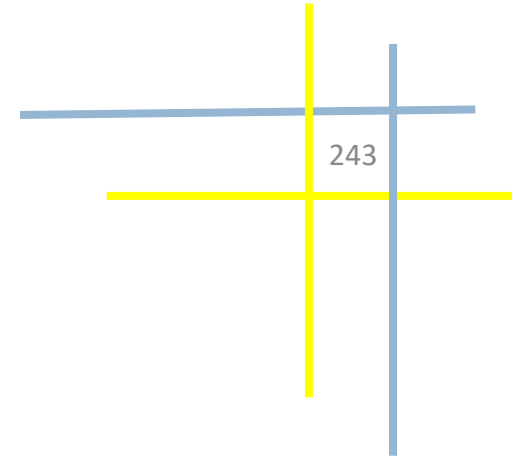


Захист даних з DPAPI

- ❑ Windows надає API захисту даних DPAPI, який компоненти Windows, власні та сторонні програми можуть використовувати для захисту будь-яких збережених даних, які розробник вважає чутливими



Захист даних користувача



- ☐ Дискреційне керування доступом
(Discretionary Access Control — DAC)
- ☐ Клієнт VPN



Дискреційне керування

- ☐ Виконавчий компонент в ядрі Windows опосередковує доступ між суб'єктами та об'єктами даних користувача, також відомими як іменовані об'єкти
- ☐ Суб'єкти складаються з процесів з одним або кількома потоками, що виконуються від імені користувачів
- ☐ Хоча політика дискреційного керування доступом Windows керує декількома різними типами іменованих об'єктів, профіль захисту, який є основним для цієї оцінки, зосереджується на файлах NTFS та об'єктах каталогів NTFS



Атрибути суб'єктів DAC

- ☐ Маркери (токени) доступу Windows (Access Tokens) містять атрибути безпеки суб'єкта
- ☐ Токени пов'язані з процесами та потоками, що працюють від імені користувача
- ☐ Адміністратор може змінити все це, за винятком SID облікового запису користувача та SID входу



Уособлені маркери (Impersonation Tokens)

246

- ❑ Поток може бути призначений “уособлений маркер” — маркер видавання себе за іншу особу, — який буде використовуватися замість основного маркера процесу під час здійснення перевірки доступу та генерації даних аудиту
- ❑ Маркер доступу може також містити список обмежувальних SID, які використовуються для обмеження доступу до об'єктів
- ❑ Рішення щодо доступу приймаються з використанням уособленого маркера потоку, якщо він існує, а в іншому випадку — основного маркера процесу потоку (який завжди існує)



Атрибути об'єкта DAC

- ☐ Дескриптори безпеки (**Security Descriptors — SD**) містять усі атрибути безпеки, пов'язані з об'єктом
- ☐ Усі іменовані об'єкти мають пов'язаний **SD**
- ☐ Атрибутами безпеки **SD**, що використовуються для дискреційного керування доступом, є:
 - **SID** власника об'єкта, який визначає власника дескриптора безпеки;
 - прапор присутності списку дискреційного керування доступом (**Discretionary Access Control List — DACL**);
 - сам **DACL**, коли він присутній.
- ☐ **DACL** містять список записів керування доступом (**Access Control Entries — ACE**)
- ☐ Кожен **ACE** вказує
 - ☐ тип **ACE**,
 - ☐ **SID**, що представляє користувача або групу,
 - ☐ маску доступу, що містить набір прав доступу.
- ☐ Кожен **ACE** має пов'язані з ним атрибути успадкування, які вказують, чи застосовується **ACE** лише до асоційованого об'єкта, лише до його дочірніх об'єктів або як до його дочірніх об'єктів, так і до асоційованого об'єкта



Два типи ACE

248

□ Існує два типи ACE, які застосовуються до дискреційного контролю доступу:

■ ДОЗВОЛИТИ ДОСТУП

- ACCESS_ALLOWED_ACE: використовується для надання доступу користувачеві або групі користувачів.
- ACCESS_ALLOWED_OBJECT_ACE: (для об'єктів служби каталогів), що використовується для надання доступу користувачеві або групі до властивості або набору властивостей об'єкту служби каталогів, або для обмеження наслідування ACE до вказаного типу дочірнього об'єкта. Цей тип ACE підтримується лише для об'єктів служби каталогів.

■ ЗАБОРОНИТИ ДОСТУП

- ACCESS_DENIED_ACE: використовується для заборони доступу користувачеві або групі користувачів.
- ACCESS_DENIED_OBJECT_ACE: (для об'єктів служби каталогів) використовується для заборони доступу користувача або групи до властивості або набору властивостей об'єкту служби каталогів, або для обмеження наслідування ACE до вказаного типу дочірнього об'єкта. Цей тип ACE підтримується лише для об'єктів служби каталогів.



Маска доступу

249

- ☐ В ACE маска доступу містить права доступу до об'єкта, надані (або заборонені) SID, що представляють користувача або групу
- ☐ Також маска доступу використовується
 - щоб вказати бажаний доступ до об'єкта при доступі до об'єкта
 - ідентифікувати наданий доступ, пов'язаний з відкритим об'єктом
- ☐ Кожен біт у масці доступу представляє певне право доступу
- ☐ Існує чотири категорії прав доступу: стандартні, специфічні, спеціальні та загальні (generic)
 - Стандартні права доступу (Standard Access Rights) застосовуються до всіх типів об'єктів
 - Специфічні права доступу (Specific Access Rights) мають різне семантичне значення залежно від типу об'єкта
 - Спеціальні права доступу (Special Access Rights) використовуються в масках бажаного доступу для запиту спеціального доступу або запиту всіх дозволених прав
 - Загальні права доступу (Generic Access Rights) — це зручні поєднання конкретних та стандартних прав доступу
- ☐ Кожен тип об'єкта забезпечує своє власне відображення між загальними правами доступу та стандартними та спеціальними правами доступу



Таблиці дескрипторів (Handle Tables)

250

- ❑ Для більшості об'єктів суб'єкт запитує доступ до об'єкта (наприклад, відкриває його) і отримує у відповідь вказівник на дескриптор (a pointer to a handle)
- ❑ K33 пов'язує надану маску доступу з кожним відкритим дескриптором
- ❑ Для об'єктів режиму ядра:
 - Дескриптори підтримуються в таблиці дескрипторів режиму ядра
 - Для кожного процесу існує одна таблиця дескрипторів
 - Кожен запис у таблиці дескрипторів ідентифікує відкритий об'єкт та права доступу, надані цьому об'єкту
- ❑ Для серверів K33 режиму користувача:
 - Дескриптор — це контрольований сервером вказівник контексту, пов'язаний із зв'язком між суб'єктом та сервером
 - Сервер використовує цей контекстний дескриптор так само, як і в режимі ядра
- ❑ (тобто, щоб знайти відкритий об'єкт і пов'язану з ним маску дозволеного доступу).
- ❑ В обох випадках (об'єкти режиму користувача та ядра) SRM приймає всі рішення щодо керування доступом



Права доступу DAC та іменовані об'єкти

251

• Каталог NTFS

- ACCESS_SYSTEM_SECURITY
READ_CONTROL
WRITE_DAC
- WRITE_OWNER
SYNCHRONIZE
FILE_LIST_DIRECTORY
FILE_ADD_FILE
FILE_ADD_SUBDIRECTORY
FILE_DELETE_CHILD
FILE_READ_ATTRIBUTES
FILE_WRITE_ATTRIBUTES
FILE_DELETE_CHILD |
FILE_ADD_FILE
- DELETE

Файл NTFS

ACCESS_SYSTEM_SECURITY
READ_CONTROL
WRITE_DAC
WRITE_OWNER
SYNCHRONIZE
FILE_WRITE_DATA
FILE_READ_DATA
FILE_APPEND_DATA
FILE_WRITE_EA
FILE_EXECUTE
FILE_READ_ATTRIBUTES
FILE_WRITE_ATTRIBUTES
FILE_WRITE_DATA and FILE_WRITE_ATTRIBUTES
DELETE
FILE_WRITE_DATA | FILE_READ_DATA
FILE_READ_DATA | FILE_EXECUTE
FILE_READ_DATA | FILE_EXECUTE |
FILE_WRITE_DATA
FILE_WRITE_DATA | FILE_WRITE_EA |
FILE_WRITE_ATTRIBUTES



Алгоритм DAC (огляд)

252

- ☐ КЗЗ застосовує політику DAC до об'єктів на підставі
 - SID і привілеїв в маркері запитувача,
 - бажаної маски доступу, що запитується,
 - дескриптора безпеки об'єкта.
- ☐ Для надання доступу всі права доступу, зазначені у бажаній масці доступу, повинні бути надані одним із наступних кроків
 1. Перевірка привілеїв
 2. Перевірка власника
 3. DACL відсутній
 4. DACL присутній, але порожній
 5. Ітеративна обробка кожного ACE у тому порядку, в якому вони відображаються в DACL
 6. Перевірка доступу щодо обмежуючих SID
- ☐ В кінці будь-якого кроку, якщо всі запитувані права доступу надані, доступ надається
- ☐ В кінці алгоритму, якщо будь-яке запитуване право доступу не було надано, доступ заборонено



Алгоритм DAC₁

253

1. Перевірка привілеїв:
 - a) Перевірка привілею SeSecurity: Це потрібно, якщо **ACCESS_SYSTEM_SECURITY** (доступ до даних аудиту) знаходиться у масці бажаного доступу
 - Якщо запитується ACCESS_SYSTEM_SECURITY, а запитувач не має цього привілею, доступ заборонено
 - В іншому випадку надається ACCESS_SYSTEM_SECURITY
 - b) Перевірка привілею SeTakeOwner:
 - Якщо маска бажаного доступу має право доступу WRITE_OWNER, а привілей знайдено в маркері запитувача, тоді доступ WRITE_OWNER надається
 - c) Перевірка привілею SeBackupPrivilege: привілей резервного копіювання файлів та каталогів дозволяє суб'єкту читати файли та об'єкти реєстру для операцій резервного копіювання незалежно від їх ACE в DACL
 - Якщо процес має привілей SeBackupPrivilege, а операція вимагає привілею, подальша перевірка не проводиться, і доступ надається
 - В іншому випадку ця перевірка неактуальна, і перевірка доступу триває



Алгоритм DAC₂

254

- d) Перевірка привілею SeRestorePrivilege: привілей «Відновлювати файли та каталоги» дозволяють процесу писати файли та об'єкти реєстру для операцій відновлення незалежно від їх ACE в DACL
- Якщо процес-суб'єкт має привілей SeRestorePrivilege, і операція вимагає привілею, подальша перевірка не проводиться, і доступ дозволений
 - В іншому випадку ця перевірка неактуальна, і перевірка доступу триває

2. Перевірка власника:

- a) Якщо DACL містить один або більше ACE з OwnerRights SID, ці записи, разом із усіма іншими застосовними ACE для користувача, використовуються для визначення прав власника
- b) В іншому випадку перевіряють усі SID в маркері, щоб визначити, чи існує збіг із власником об'єкта. Якщо так, то права READ_CONTROL та WRITE_DAC надаються якщо були запитані

3. DACL відсутній:

- Усі подальші запитовані права доступу надаються

4. DACL присутній, але порожній:

- Якщо вимагаються будь-які додаткові права доступу, у доступі відмовляється



Алгоритм DAC₃

255

5. Ітеративно обробляють кожен ACE у тому порядку, в якому вони відображаються в DACL, як описано нижче:

- а) Якщо атрибути успадкування ACE вказують, що ACE застосовується лише до дочірніх об'єктів асоційованого об'єкта, ACE пропускається
- б) Якщо SID в ACE не відповідає жодному SID у маркері доступу запитувача, ACE пропускається
- в) Якщо знайдено збіг SID, а маска доступу в ACE відповідає доступу в бажаній масці доступу (див. Наступний слайд)

■ ACE типу “Доступ дозволено”:

- Якщо ACE має тип ACCESS_ALLOWED_OBJECT_ACE, і ACE включає GUID, що представляє набір властивостей або властивість, пов'язану з об'єктом, тоді доступ надається набору властивостей або конкретній властивості, представлений GUID (а не до цілого об'єкта)
- В іншому випадку (ACCESS_ALLOWED_ACE) ACE надає доступ до всього об'єкта

■ ACE типу “Доступ заборонено”:

- Якщо ACE має тип ACCESS_DENIED_OBJECT_ACE, і ACE включає GUID, що представляє набір властивостей або властивість, пов'язану з об'єктом, тоді доступ заборонено до набору властивостей або конкретної властивості, представленої GUID
- В іншому випадку (ACCESS_DENIED_ACE) ACE забороняє доступ до всього об'єкта
- Якщо ACE конкретно відмовляє в запитуваному доступі, тоді весь запит на доступ є невдалим



Алгоритм DAC₄

6. Перевірка обмежувальних SID:

- Якщо всі запитані права доступу надані, але маркер запитувача має принаймні один обмежувальний SID, проводиться повна перевірка доступу щодо обмежувальних SID
- Якщо ця друга перевірка доступу не надає бажаного доступу, тоді весь запит на доступ є невдалим



Захист DAC за замовчуванням

257

- ❑ КЗЗ забезпечує процес, який гарантує, що DACL застосовується за умовчанням до всіх нових об'єктів
 - Коли створюються нові об'єкти, створюється відповідний DACL
- ❑ Методи, що використовуються для побудови DACL для нового об'єкта служби каталогів та для нових іменованих об'єктів ядра, дещо відрізняються. Є дві ключові відмінності, які полягають у наступному:
 - Для об'єктів служби каталогів правила створення DACL розмежовують загальні спадкові ACE та специфічні об'єктно-успадкокувані ACE у SD батьківського об'єкта.
 - Загальні спадкові ACE можуть успадковуватися всіма типами дочірніх об'єктів.
 - Об'єктно-специфічні спадкові ACE можуть успадковуватися лише за типом дочірнього об'єкта, до якого вони застосовуються
 - Визначення схеми Active Directory для об'єкта може включати SD
 - Кожен клас об'єктів, визначений у схемі, має атрибут defaultSecurityDescriptor
 - Якщо ні процес створення, ні успадкування від батьківського об'єкта не надають DACL для нового об'єкта Active Directory, Windows використовує DACL у типовому SD, визначеному схемою



Правила встановлення DACL в SD для нових об'єктів служби каталогів₁

258

□ DACL об'єкта — це DACL із SD, визначеного процесом, що створює

- Windows об'єднує будь-які успадковувані ACE в DACL, якщо SE_DACL_PROTECTED не встановлено у прапорах керування SD
- Потім Windows встановлює прапорець керування SE_DACL_PRESENT у SD

□ Якщо в процесі створення не вказано SD

- Windows перевіряє DACL батьківського об'єкта на наявність успадкованих об'єктно-залежних ACE, які застосовуються до типу створюваного об'єкта
- Якщо батьківський об'єкт має успадковувані специфічні об'єкти власного використання для типу об'єкта, Windows будує DACL об'єкта із спадкоємних елементів управління активами, включаючи загальні та специфічні об'єкти управління
- Потім Windows встановлює прапор керування SE_DACL_PRESENT у SD



Правила встановлення DACL в SD для нових об'єктів служби каталогів₂

259

□ Якщо батьківський об'єкт не має успадковуваних специфічних ACE для типу створюваного об'єкта

- Windows використовує DACL за умовчанням зі схеми AD для цього типу об'єкта
- Потім встановлює прапори керування SE_DACL_PRESENT та SE_DACL_DEFAULTED у SD

□ Якщо схема AD не визначає типову DACL за типом для об'єкта

- Windows використовує DACL за умовчанням у маркері доступу суб'єкта
- Потім встановлює прапори керування SE_DACL_PRESENT та SE_DACL_DEFAULTED у SD
- Маркер доступу суб'єкта завжди має DACL за умовчанням, який встановлюється підкомпонентом LSA при створенні маркера



DAC Management

260

□ *Нижче наведено чотири методи зміни DACL:*

- **Власник об'єкта:** має неявний доступ до **WRITE_DAC**
- **Явний доступ до зміни DACL:** користувач, якому надано явний доступ **WRITE_DAC** до DACL, може змінити DACL
- **Стати власником, використовуючи явний доступ:** Користувач, якому надано явний доступ **WRITE_OWNER** до DACL, може отримати право власності на об'єкт, а потім використовувати неявний доступ власника **WRITE_DAC**
- **Стати власником, використовуючи привілей:** Користувач із привілеєм **SeTakeOwner** може отримати право власності на об'єкт, а потім користуватися неявним доступом власника до **WRITE_DAC**



Арбітраж посилань (Reference Mediation)

261

- ❑ Доступ до об'єктів у системі в цілому заснований на отриманні дескриптора (handle) об'єкта
 - Дескриптори зазвичай отримуються в результаті відкриття або створення об'єкта
 - Дескриптори також можуть бути успадковані від батьківського процесу або безпосередньо скопійовані (з відповідним доступом) з іншого суб'єкта
 - Дескриптор завжди має асоційовану маску наданого доступу
- ❑ У деяких випадках до об'єктів звертаються безпосередньо за назвою без проміжного кроку попереднього отримання дескриптора



VPN Client₁

262

❑ Клієнт **IPsec VPN** у Windows може бути налаштований локальним адміністратором пристрою

- Адміністратор може налаштувати IPsec VPN-клієнт, щоб весь IP-трафік проходив через тунель IPsec, за винятком:
 - Трафіку IKE, що використовується для встановлення тунелю VPN
 - Трафіку IPv4 ARP для визначення локальних адрес мережного рівня та встановлення локальної адреси
 - Трафіку IPv6 NDP для визначення локальних адрес мережного рівня та встановлення локальної адреси

❑ **IPsec VPN** — це наскрізна (**end-to-end**) технологія роботи в мережі Інтернет

- Сеанси VPN можуть встановлюватися над протоколами фізичних мереж, таких як бездротова локальна мережа (Wi-Fi), або локальних мереж



VPN Client₂

263

□ Компоненти, що відповідають за маршрутизацію IP-трафіку через VPN-клієнт:

- **Мережний стек IPv4 / IPv6** у ядрі обробляє вхідний та вихідний мережний трафік
- **Служба модулів ключів (IPsec та IKE та AuthIP Keying Modules)**, яка розміщує модулі ключів **IKE** та **AuthIP (Authenticated Internet Protocol)**
 - Ці модулі ключів використовуються для автентифікації та обміну ключами в IPsec
- **Драйвер пристрою служби віддаленого доступу в ядрі (Remote Access Service device driver)**, який використовується переважно для з'єднань VPN; відомий як **"RAS IPsec VPN"** або **"RAS VPN"**
- **Служба агента політики IPsec (IPsec Policy Agent service)**, яка забезпечує реалізацію політик IPsec



VPN Client₃

- ❑ Розробники Універсальних програм для Windows (**Universal Windows App**) можуть реалізувати власний клієнт VPN, якщо корпорація Майкрософт дозволить використовувати можливість (**capability**) **networkingVpnProvider**, що включає встановлення політики блокування мережного трафіку, як описано вище

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 9. Ідентифікація та автентифікація. Захист функцій КЗЗ.

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



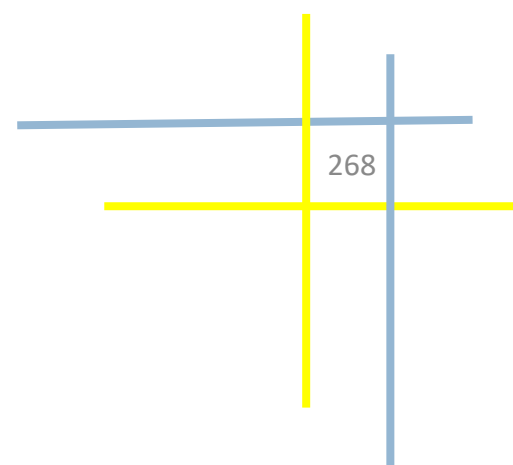
Лекційні питання

- ☐ Мета лекції
- ☐ Ідентифікація та автентифікація
- ☐ Захист функцій КЗЗ
- ☐ Блокування сеансу
- ☐ Захищений шлях для зв'язку
- ☐ Управління безпекою



Мета лекції

- ☐ Ознайомитися з питаннями ідентифікації та автентифікація в Windows
- ☐ Розглянути підходи до захисту функцій КЗЗ
- ☐ Ознайомитися з процесом блокування сеансу
- ☐ Розглянути захищений шлях для зв'язку
- ☐ Ознайомитися з підходами управління безпекою в Windows



Ідентифікація та автентифікація



Ідентифікація та автентифікація

- ☐ Основи політики ідентифікації та автентифікації
- ☐ Способи ідентифікації та автентифікації
- ☐ Результат ідентифікації та автентифікації
- ☐ Захист від підбирання паролів
- ☐ Перевірка (validation) та генерування сертифікатів X.509
- ☐ Сховище сертифікатів



Основи політики ідентифікації та автентифікації

- ☐ Кожен користувач Windows повинен бути ідентифікований та автентифікований на основі визначеної адміністратором політики перед виконанням будь-яких функцій, опосередкованих КЗЗ
- ☐ Інтерактивний користувач звертається до надійного шляху, щоб захистити свою ідентифікаційну інформацію
- ☐ Windows підтримує бази даних облікових записів, включаючи їх ідентифікаційні дані, інформацію про автентифікацію, асоціації з групами та асоціації привілеїв та прав входу
- ☐ Функції політики облікового запису Windows включають можливість визначити мінімальну довжину пароля, кількість невдалих спроб входу, тривалість блокування та вік пароля



Способи ідентифікації та автентифікації₁

- ❑ Windows 10, Windows 10 S та Windows Server можуть автентифікувати користувачів на основі:
 - імені користувача та паролю
 - за допомогою PIN-коду Windows Hello, який підтримується модулем TPM
- ❑ Windows 10 і Windows Server також можуть використовувати фізичну або віртуальну смарт-карту, таким чином підтримуючи багаторазову автентифікацію користувачів
- ❑ Усі входи в систему по суті обробляються однаково незалежно від їх джерела:
 - інтерактивний вхід,
 - мережний інтерфейс,
 - внутрішньо ініційований вхід до служби
- ❑ і починаються з
 - імені облікового запису,
 - доменного імені (яке може бути NULL; із зазначенням локальної системи),
 - та облікових даних, які повинні бути надані K33

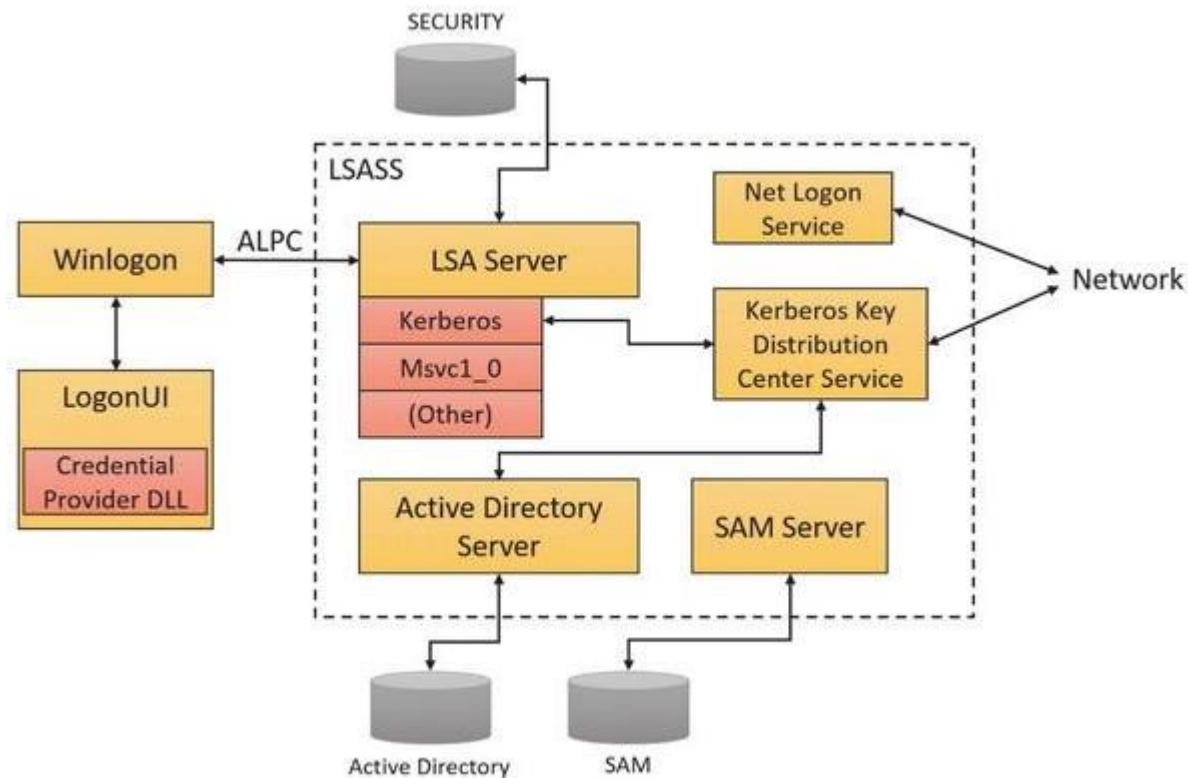


Способи ідентифікації та автентифікації₂

- ☐ Автентифікація на основі пароля для Windows успішна, коли надані користувачем дані відповідають збереженому захищеному поданню пароля
- ☐ Автентифікацію за допомогою пароля можна використовувати для:
 - інтерактивних входів,
 - сервісних входів,
 - входів в мережу,
 - і запуску екрану "змінити пароль"
- ☐ Windows Hello та смарт-картки використовують автентифікацію на основі PIN-коду, щоб розблокувати захищений ресурс, приватний ключ
 - збережене представлення PIN-коду захищене ядром безпеки (Secure Kernel)
- ☐ PIN-код Windows Hello, фізичні та віртуальні смарт-картки можна використовувати для інтерактивних входів
- ☐ PIN-код Windows Hello використовується для повторної автентифікації користувача, коли користувач вирішить змінити свій PIN-код



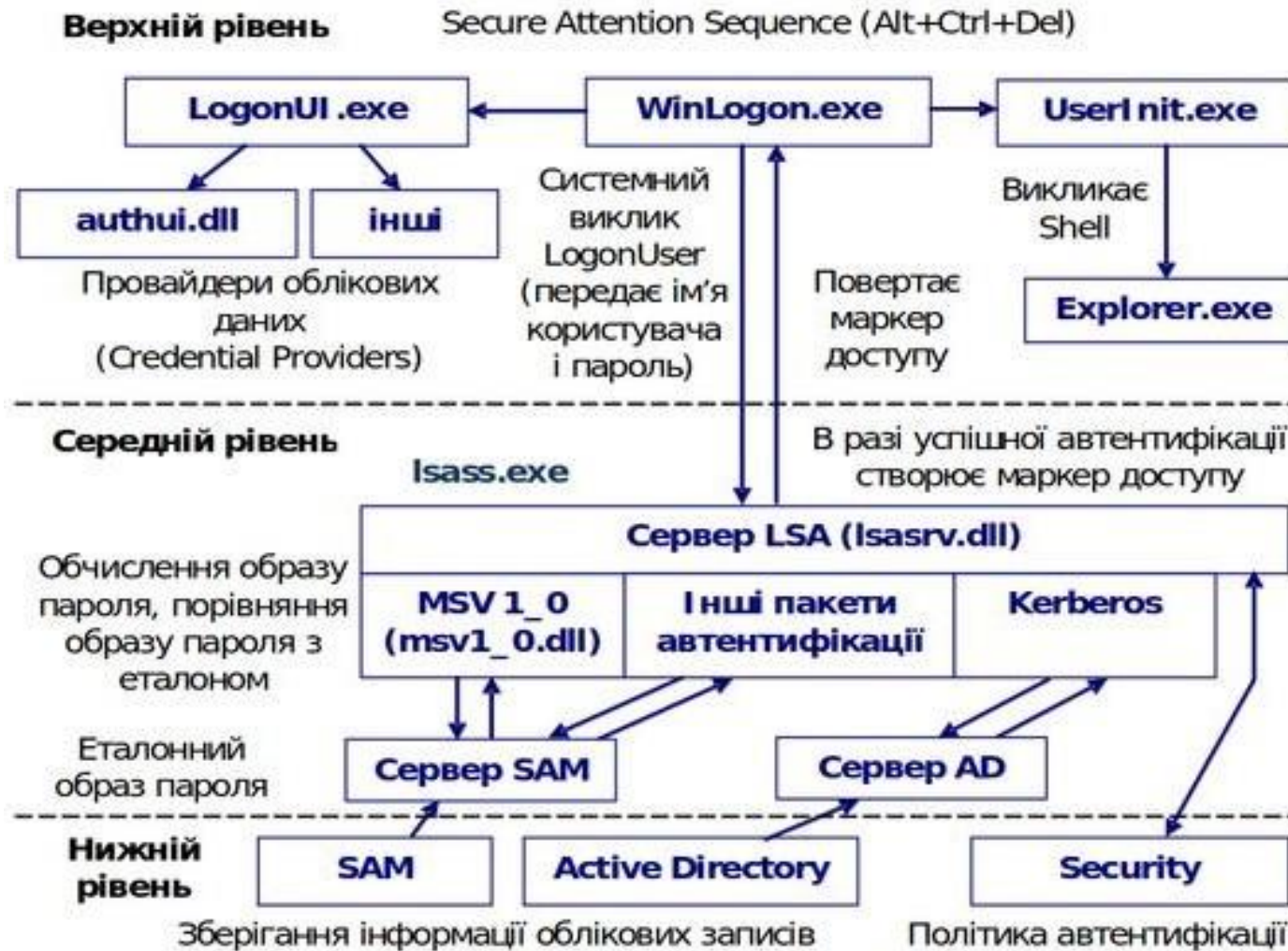
Компоненти, що беруть участь в ідентифікації та автентифікації₁



Ілюстрація з *Windows Internals, Part 1, 7th edition*



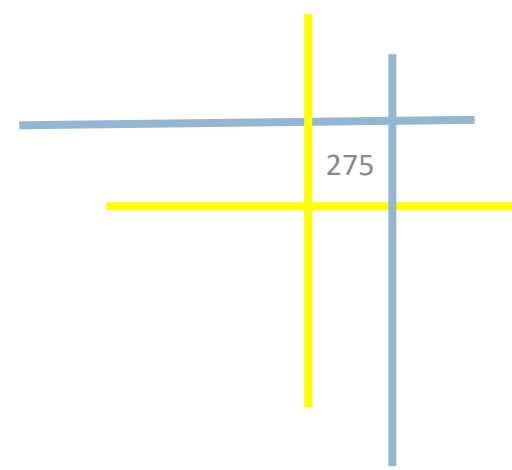
Компоненти, що беруть участь в ідентифікації та автентифікації₂





Результат ідентифікації та автентифікації

- ❑ Коли автентифікація успішна,
- користувач увійде на свій робочий стіл,
 - його екран буде розблоковано,
 - або зміняться його фактори автентифікації
- залежно від того, чи**
- користувач увійшов на комп'ютер,
 - дисплей був заблокований,
 - або потрібно було змінити PIN-код або пароль





Захист від підбирання паролів

- ❑ Локальна підсистема безпеки — **Local Security Authority (LSA)** у Windows підтримує підрахунок послідовних невдалих спроб входу в систему деякого облікового запису від їх останньої успішної автентифікації
- ❑ Інтерактивні входи виконуються на захищеному робочому столі, що не дозволяє запускати інші програми, а отже, перешкоджає автоматичному вгадуванню пароля
- ❑ Крім того, компонент входу в систему Windows примушує затримку в одну секунду між кожним невдалим входом із збільшенням затримки після декількох послідовних збоїв входу



Перевірка (validation) та генерування сертифікатів X.509₁

- ☐ Кожен компонент Windows, який використовує сертифікати X.509, відповідає за перевірку сертифікатів
- ☐ Кожен компонент, який використовує сертифікати X.509, матиме сховище публічних сертифікатів і буде вибирати сертифікат на основі певних критеріїв
- ☐ Компонент Windows використовуватиме ті самі типи інформації разом із шляхом сертифікації та списками довіри сертифікатів як частину рішення про прийняття сертифіката



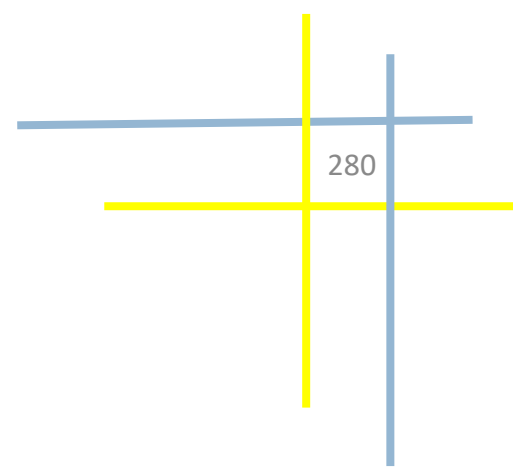
Перевірка (validation) та генерування сертифікатів X.509₂

- ☐ Якщо перевірка сертифіката не вдається або якщо Windows не може перевірити статус перевірки сертифіката, Windows не встановить надійний мережний канал
- ☐ Коли Windows потрібно сформулювати запит на реєстрацію сертифіката, він буде включати певні атрибути



Сховище сертифікатів

- ☐ В ОС Windows збережені сертифікати, відомі як надійні кореневі сертифікати, містяться в сховищах сертифікатів
- ☐ Кожен користувач має власне сховище сертифікатів, а також існує сховище сертифікатів для облікового запису комп'ютера
- ☐ Сертифікати, які використовуються програмами, наприклад, TLS, також розміщуються в сховищах сертифікатів для користувача
- ☐ На додаток до стандартних процесів відкриття сертифікатів, сертифікати програми можна завантажувати за допомогою адміністративних інструментів, таких як **certutil.exe**, зміни довірених кореневих сертифікатів можна внести за допомогою Списків довіри сертифікатів (**Certificate Trust Lists**)



Захист функцій КЗЗ



Захист функцій КЗЗ

- ☐ Відокремлення та ізоляція домену
- ☐ Захист бінарних файлів ОС, даних аудиту та конфігурації
- ☐ Захист від недоліків реалізації
- ☐ Цілісність платформи Windows та цілісність коду
- ☐ Оновлення Windows та програм



Захист функцій КЗЗ

- ☐ Windows захищає від несанкціонованого розкриття та зміни даних за допомогою набору стандартних протоколів Інтернету, включаючи IPsec, IKE та ISAKMP
- ☐ Windows забезпечує безпеку ізоляції процесів для всіх процесів через приватні віртуальні адресні простори, контекст виконання та контекст безпеки
- ☐ Структури даних Windows, що визначають адресний простір процесу, контекст виконання, захист пам'яті та контекст безпеки, зберігаються у захищеній пам'яті режиму ядра
- ☐ Windows містить функції самоперевірки, які забезпечують цілісність виконуваних образів програми та її криптографічних функцій
- ☐ Нарешті, Windows надає надійний механізм оновлення для оновлення самих бінарних файлів Windows



Розділення та ізоляція доменів

- ❑ КЗЗ забезпечує домен безпеки для власного захисту та забезпечує ізоляцію процесів
- ❑ Домени безпеки, які використовуються всередині та за допомогою КЗЗ, складаються з наступного:
 - Апаратне забезпечення
 - Розділи віртуалізації
 - Програмне забезпечення режиму ядра
 - Захищені (trusted) процеси режиму користувача
 - Процес адміністративних засобів (Administrative tools) режиму користувача
 - Контейнери для прикладних програм



КЗЗ: Апаратне забезпечення і програмне забезпечення режиму ядра

- ☐ Апаратне забезпечення КЗЗ управляється програмним забезпеченням КЗЗ режиму ядра і не може бути модифіковано ненадійними суб'єктами
- ☐ Програмне забезпечення КЗЗ режиму ядра захищене від модифікацій апаратним станом виконання та захистом як фізичної пам'яті, так і пам'яті, виділеної розділу
- ☐ Апаратне забезпечення КЗЗ надає інструкцію програмного переривання, яка спричиняє зміну стану з режиму користувача на режим ядра всередині розділу
- ☐ Програмне забезпечення КЗЗ режиму ядра відповідає за обробку всіх переривань і визначає, чи дозволений виклик режиму ядра, що здійснюється



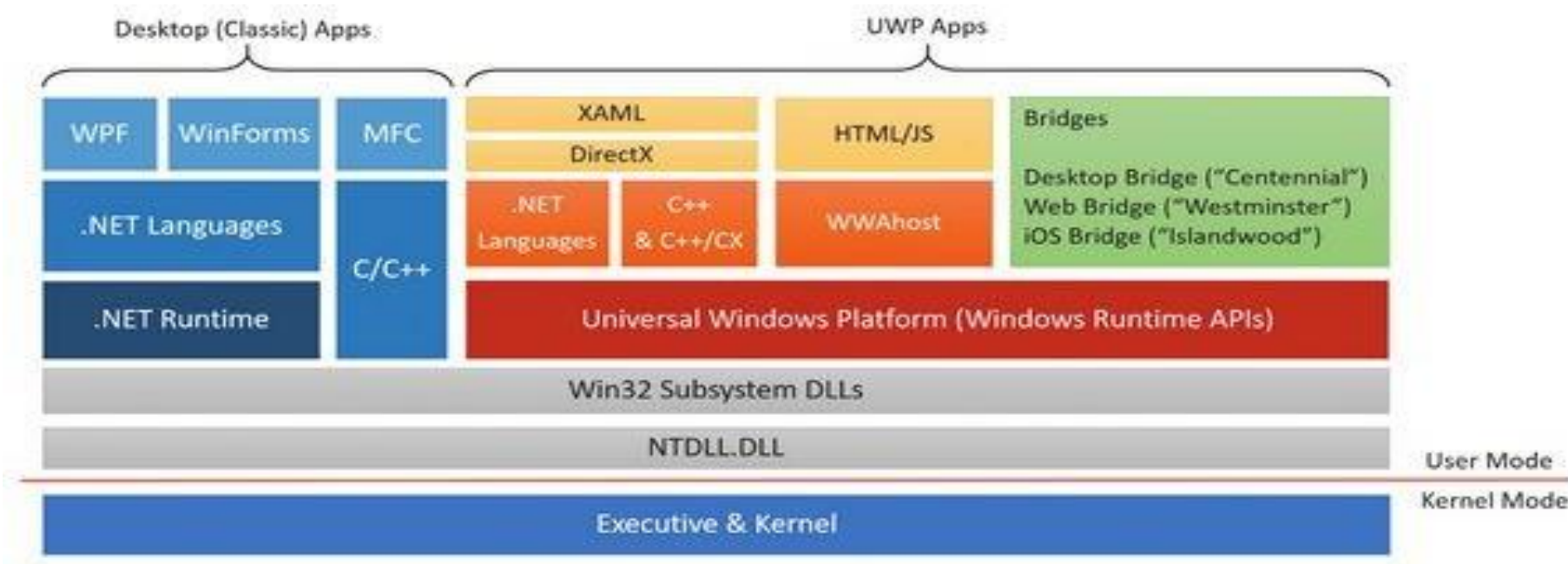
Ізоляція процесів режиму користувача

- ❑ K33 забезпечує ізоляцію для всіх процесів режиму користувача
- ❑ Структури даних, що визначають адресний простір процесу, контекст виконання та контекст безпеки, зберігаються у захищеній пам'яті режиму ядра
- ❑ Як і процеси K33, процеси користувача також мають приватний адресний простір та контекст процесу
- ❑ Інструменти адміністратора режиму користувача виконуються з контекстом безпеки процесу, який працює від імені уповноваженого адміністратора
- ❑ Крім того, K33 має додаткову можливість захисту сторінок пам'яті за допомогою запобігання виконанню даних (**Data Execution Prevention — DEP**)



Контейнери прикладних програм

- ❑ Контейнери прикладних програм ("**App Containers**") надають середовище виконання для універсальних програм Windows (**Universal Windows Applications**)



Ілюстрація з Windows Internals, Part 1, 7th edition



Криптографічний захист

- ❑ КЗЗ реалізує криптографічні механізми в межах окремого процесу режиму користувача, де до його послуг можна отримати доступ як з компонентів ядра, так і з режиму користувача
- ❑ Крім того, КЗЗ включає функцію перевірки цілісності коду (**Code Integrity Verification feature**), також відому як підписування коду в режимі ядра (**Kernel- mode code signing — KMCS**), за допомогою якої драйвери пристроїв завантажуватимуться лише в тому випадку, якщо вони підписані цифровим підписом або Microsoft, або надійного кореневого центру сертифікації, визнаного Microsoft



Захист бінарних файлів ОС, даних аудиту та конфігурації

- ❑ За умовчанням операційна система Windows встановлюється в каталозі `\Windows\` першого завантажувального розділу сховища даних комп'ютера
- ❑ Ядро, драйвери пристрою (файли `.sys`), системні виконувані файли (файли `.exe`) та динамічно завантажувані бібліотеки (файли `.dll`) зберігаються в каталозі `%systemRoot%\system32` і підкаталогах `system32`
- ❑ Кореневий каталог журналів аудиту — `%systemRoot%\system32\winevt`
- ❑ Основним сховищем даних конфігурації для Windows є `реєстр`



Захист від недоліків реалізації₁

- ❑ Ядро Windows, програми режиму користувача та всі програми магазину Windows реалізують рандомізацію адресного простору (**Address Space Layout Randomization — ASLR**) для завантаження виконуваного коду за непередбачуваними базовими адресами
- ❑ Виконавча система Windows також надає можливість захисту від переповнення буфера в стеку (**stack buffer overrun protection capability**), яка припиняє процес після того, як Windows виявить потенційне перевищення буфера у стеку потоку, перевіряючи значення “канарійок” у прологу та епілогу функцій, а також впорядковуючи стек
- ❑ Усі бінарні файли Windows та програми магазину Windows реалізують захист від переповнення буфера стека



Захист від недоліків реалізації₂

- ❑ Щоб увімкнути ці засоби захисту за допомогою середовища розробки Microsoft Visual Studio, програми компілюють з
 - опцією **/DYNAMICBASE** для ASLR,
 - і додатково з **/HIGHENTROPYVA** для 64-розрядних ASLR,
 - або **/NXCOMPAT:NO**, щоб відмовитися від DEP, заснованого на програмному забезпеченні,
 - та **/GS** (увімкнено за замовчуванням) для захисту від переповнення буфера в стеку
- ❑ Програми Магазину Windows компілюються за допомогою параметра **/APPCONTAINER**, який створює виконуваний файл для запуску в контейнері програм Windows для роботи із захистом у режимі користувача, описаним вище



Цілісність платформи Windows та цілісність коду

- ❑ Операційна система Windows перевіряє цілісність програмного коду Windows, використовуючи комбінацію можливостей безпечного завантаження (**Secure Boot**) та цілісності коду (**Code Integrity**) у Windows
- ❑ На комп'ютерах з модулем надійної платформи (**Trusted Platform Module — TPM**) перед завантаженням Windows комп'ютер перевірить цілісність компонентів раннього завантаження
- ❑ Ця можливість відома як **Secure Boot**



Підписування коду режиму ядра

- ❑ Підписування коду режиму ядра (**Kernel-mode code signing — KMCS**) запобігає завантаженню драйверів пристроїв режиму ядра, таких як мережний драйвер TCP/IP (**tcpip.sys**), якщо вони не опубліковані та не підписані розробниками, які перевірені одним із декількох довірених центрів сертифікації (ЦС)



Захист файлів Windows

- ❑ Захист файлів Windows (**Windows File Protection**) підтримує набір захищених файлів, які зберігаються у кеші разом із криптографічними хешами кожного з цих файлів
- ❑ Після ініціалізації системи **Захист файлів Windows** завантажується і сканує захищені файли, щоб переконатися, що вони мають дійсні криптографічні хеші
- ❑ Захист файлів Windows також реєструється для отримання сповіщень у разі зміни будь-якого із захищених файлів, щоб він міг повторно перевірити криптографічну контрольну суму в будь-який момент, коли система працює
- ❑ Якщо будь-яка з перевірок криптографічного хешу не вдалася, відповідний файл буде відновлено з кешу



Оновлення Windows і програм₁

- ❑ Оновлення Windows постачаються у вигляді файлів окремого пакета оновлень Майкрософт — **Microsoft Update Standalone Package files** — (файли **.msu**), підписані Майкрософт двома цифровими підписами
- ❑ Цифровий підпис підписується корпорацією Майкрософт із шляхом сертифікації через сертифікат підписання коду Майкрософт (**Microsoft Code Signing certificate**) і, зрештою, кореневий центр сертифікації Майкрософт (**Microsoft Root Certification Authority**)
- ❑ Операційна система Windows перевірить, чи є сертифікат дійсним і не був відкликаний, за допомогою стандартної PKI CRL
- ❑ Зауважте, що інсталятор Windows не встановить оновлення, якщо файли в упаковці мають менші номери версій, ніж встановлені файли



Оновлення Windows і програм₂

- ❑ Цілісність сертифіката підписання коду Майкрософт (**Microsoft Code Signing certificate**) на комп'ютері захищається кореневим ключем сховища в TPM, а також перевіреною цілісністю бінарних файлів Windows завдяки **Secure Boot** та **Code Integrity**
- ❑ Оновлення операційної системи Windows, програм Windows та десктопних програм Майкрософт доставляються через
 - **Windows Update** (для Windows) і
 - **Microsoft Update** (для десктопних програм Майкрософт),
- ❑ Після цього користувач може перевірити правильність підпису, переглянувши дані про цифровий підпис файлу з Windows Explorer, або за допомогою командлета PowerShell **Get-AuthenticodeSignature**



Програми магазину Windows

- ❑ Програми Універсальної платформи Windows (**Universal Windows Platform — UWP**) можна завантажити з магазину Майкрософт, а їх інсталяційні пакети перевірити за допомогою цифрового підпису корпорації Майкрософт із використанням підпису коду
- ❑ Служба розгортання AppX (**AppX Deployment Service**) перевірить цифровий підпис RSA SHA-256 для карти блоків та інших метаданих AppX на початку завантаження пакету (або колекції) AppX



Розповсюдження оновлень

❑ Існує кілька каналів розповсюдження оновлень для ОС Windows та програм Windows:

- **Windows Update**: Windows Update — це веб-сервіс для доставки оновлень Windows безпосередньо споживачам
- Служби оновлення Windows Server (**Windows Server Update Services — WSUS**): WSUS — це роль сервера у Windows Server, яку IT-адміністратори можуть використовувати для розповсюдження оновлень програм серед користувачів на своєму підприємстві
- Магазин Windows (**Windows Store**): Магазин Windows — це веб-сервіс для доставки оновлень до програм універсальної платформи Windows (**Universal Windows Platform**), які спочатку були встановлені з Магазину Windows



Блокування сеансу (Session Locking)



Блокування сеансу (Session Locking)

- ☐ Можливість заблокувати сеанс одразу або через певний проміжок часу
- ☐ Блокування комп'ютера після встановленого періоду бездіяльності



- ❑ Windows надає користувачеві можливість заблокувати свій інтерактивний сеанс входу за власним бажанням або після визначеного користувачем тайм-ауту бездіяльності
- ❑ Коли користувач має сеанс на робочому столі, він може викликати функцію блокування сеансу, використовуючи ту ж комбінацію клавіш, яка використовується для виклику довіреного шляху — **Secure Attention Sequence (Ctrl+Alt+Del)**



Блокування за інтервалом бездіяльності

- ☐ Windows надає адміністратору можливість вказати інтервал бездіяльності, після якого сеанс буде заблоковано.
- ☐ Windows постійно відстежує активність миші, клавіатури, сенсорного дисплея та датчика орієнтації, щоб визначити, чи вони неактивні протягом заданого періоду часу.
- ☐ Після цього Windows заблокує робочу станцію та запустить заставку, якщо користувач не переглядає відео, наприклад, фільм.



Розблокування сеансу

- ☐ Після того, як комп'ютер був заблокований, щоб розблокувати сеанс, користувач натискає клавішу або проводить пальцем по дисплею.
- ☐ Потім користувач повинен повторно ввести свої дані автентифікації, які були кешовані локальною системою від початкового входу, після чого дисплей користувача буде відновлено, а сеанс відновиться.
- ☐ В рамках встановлення інтерактивного сеансу входу в систему Windows можна налаштувати на відображення банера для входу, визначеного адміністратором, який користувач повинен прийняти перед встановленням сеансу.
- ☐ Як описано в інструкціях для адміністраторів, авторизований адміністратор може вказати, до яких мереж Wi-Fi (SSID) може бути підключений комп'ютер.



Захищений шлях для зв'язку (Trusted Path for Communications)



Захищені (Trusted) мережні канали

- ☐ Windows надає захищені мережні канали для зв'язку з підтримуючою ІТ-інфраструктурою або програмами
- ☐ Щоб встановити надійний канал, ці комунікації захищені, як описано вище
- ☐ Windows впроваджує IEEE 802.11-2012, IEEE 802.1X та EAP-TLS для забезпечення автентифікованих сеансів бездротових мереж за запитом користувача



Управління безпекою (Security Management)



Управління безпекою (Security Management)

306

- ☐ Windows має кілька функцій для управління політиками безпеки
- ☐ Управління політикою здійснюється за допомогою поєднання керування доступом, членства в групах адміністраторів, та привілеїв



Управління безпекою₁

Функція управління	Адміні-стратор	Кори-стувач
Активувати/деактивувати блокування екрану	✓	✓
Задати тайм-аут неактивності для блокування екрану	✓	✓
Задати ємність локального сховища даних аудиту	✓	
Задати мінімальну довжину пароля	✓	
~~Задати мінімальну кількість спеціальних символів в паролі~~		
~~Задати мінімальну кількість цифрових символів в паролі~~		
~~Задати мінімальну кількість символів верхнього регістру в паролі~~		
~~Задати мінімальну кількість символів нижнього регістру в паролі~~		
Задати тайм-аут неактивності віддаленого з'єднання	✓	
~~Дозволити/заборонити логін без автентифікації~~		
Конфігурувати політику блокування для неуспішних спроб автентифікації (тайм-ауті між спробами, обмеження кількості спроб за певний період часу)	✓	
Конфігурувати мережний екран на хості	✓	
Задати ім'я/адресу сервера каталогів для зв'язку з ним	✓	



Управління безпекою₂

Функція управління	Адміністратор	Користувач
Задати ім'я/адресу віддаленого сервера керування, з якого отримувати налаштування управління	✓	
~~Задати ім'я/адресу сервера аудиту/реєстрації, на який надсилати записи аудиту/реєстрації~~		
Конфігурувати правила аудиту	✓	
Задати ім'я/адресу мережного сервера часу	✓	
Активувати/деактивувати автоматичні оновлення програмного забезпечення	✓	
Конфігурувати інтерфейс Wi-Fi	✓	
Активувати/деактивувати інтерфейс Bluetooth	✓	
Конфігурувати інтерфейси USB	✓	
Активувати/деактивувати інтерфейс локальної мережі	✓	

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 10. Технологія Active Directory.

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- ☐ Об'єкти Active Directory
- ☐ Ідентичність та доступ
- ☐ Компоненти Active Directory
- ☐ Логічні структури
- ☐ Фізична структура
- ☐ Концепції Active Directory



Ідентичність та доступ (IDA)

- ☐ Інфраструктура IDA повинна:
 - Зберігати інформацію про користувачів, групи, комп'ютери та інші особи.
- ☐ Ідентичність — це представлення сутності, яка виконуватиме дії на сервері.
- ☐ Компонент IDA — це сховище ідентифікаційних даних, яке містить властивості, які унікально ідентифікують об'єкт, наприклад:
 - Ім'я користувача
 - Ідентифікатор безпеки (SID)
 - Пароль
- ☐ Сховище даних Active Directory (AD) є сховищем ідентифікаційних даних.
- ☐ Сам каталог розміщується на контролері домену – сервері, який виконує роль служби домену Active Directory (AD DS) і керується НИМ.



Обов'язки IDA

☐ Аутентифікація

- AD використовує автентифікацію Kerberos

☐ Управління доступом

- Підтримує список контролю доступу (ACL)
- Відображає політику безпеки, що складається з дозволів, які визначають рівні доступу для певних ідентифікаторів.

☐ Аудит

- Дозволяє відстежувати зміни та дії в інфраструктурі IDA



Технології IDA, що підтримуються AD

- ☐ Ідентичність
- ☐ Застосунки
- ☐ Довіра
- ☐ Цілісність
- ☐ Партнерство



Ідентичність

□ Доменні служби Active Directory (AC DC)

- Центральне сховище для керування ідентифікацією.
- Надає послуги автентифікації та авторизації через групову політику.
- Надає служби керування інформацією та обміну даними, що дозволяє користувачам знаходити будь-який компонент шляхом пошуку в каталозі.



Застосунки

❑ Полегшені служби каталогів застосунків (AD LDS)

- По суті, це автономна версія AD
- Зберігає та копіює лише інформацію, пов'язану з застосунком.
- Зазвичай використовується застосунками, які вимагають сховища каталогу, але не вимагають такої широкої реплікації інформації, як для всіх контролерів домену.
- Дозволяє розгортати спеціальну схему для підтримки застосунку без зміни схеми AD DS.
- Формально відомий як Active Directory Application Mode (ADAM)



Довіра (траст)

❑ Служби сертифікації Active Directory (AD CS)

- Використовується для налаштування центру сертифікації для видачі цифрових сертифікатів як частини інфраструктури відкритих ключів (PKI), яка прив'язує особу, пристрій або послугу до відповідного закритого ключа.
- Якщо використовується AD CS для надання цих послуг зовнішнім спільнотам, тоді AD CS слід пов'язати із зовнішнім відомим центром сертифікації.



Цілісність

- ❑ Служби керування правами Active Directory (AD RMS)
- ❑ Технологія захисту інформації, яка дозволяє впроваджувати шаблони постійних політик використання, які визначають авторизоване та неавторизоване використання
 - напр., можна налаштувати шаблон, який дозволяє користувачам читати документ, але не друкувати чи копіювати його вміст.



Партнерство

❑ Служби федерації Active Directory (AD FS)

- Дозволяє організації розширювати IDA на кількох платформах, включаючи середовища Windows і не-Windows.
- Передає ідентифікаційні дані та права доступу надійним партнерам через межі безпеки.
- Підтримує єдиний вхід (SSO).



Поза межами IDA

- ❑ AD пропонує більше, ніж рішення IDA
- ❑ AD надає механізми для підтримки, керування та налаштування ресурсів у розподіленому мережевому середовищі.
 - Схема
 - Адміністрування на основі політики
 - Послуги реплікації



Схема

- ❑ Набір правил, що визначає класи об'єктів і атрибутів, які можуть міститися в каталозі.
 - напр., той факт, що AD має об'єкти користувача, які містять ім'я користувача та пароль, пояснюється тим, що схема визначає клас об'єкта користувача, два атрибути та зв'язок між класом об'єкта та атрибутами.



Адміністрування на основі політик

- ☐ Забезпечує єдину точку для налаштування параметрів, які потім розгортаються в кількох системах.
- ☐ Такі політики включають:
 - Групові політики;
 - Політики аудиту;
 - Точну політика паролів.



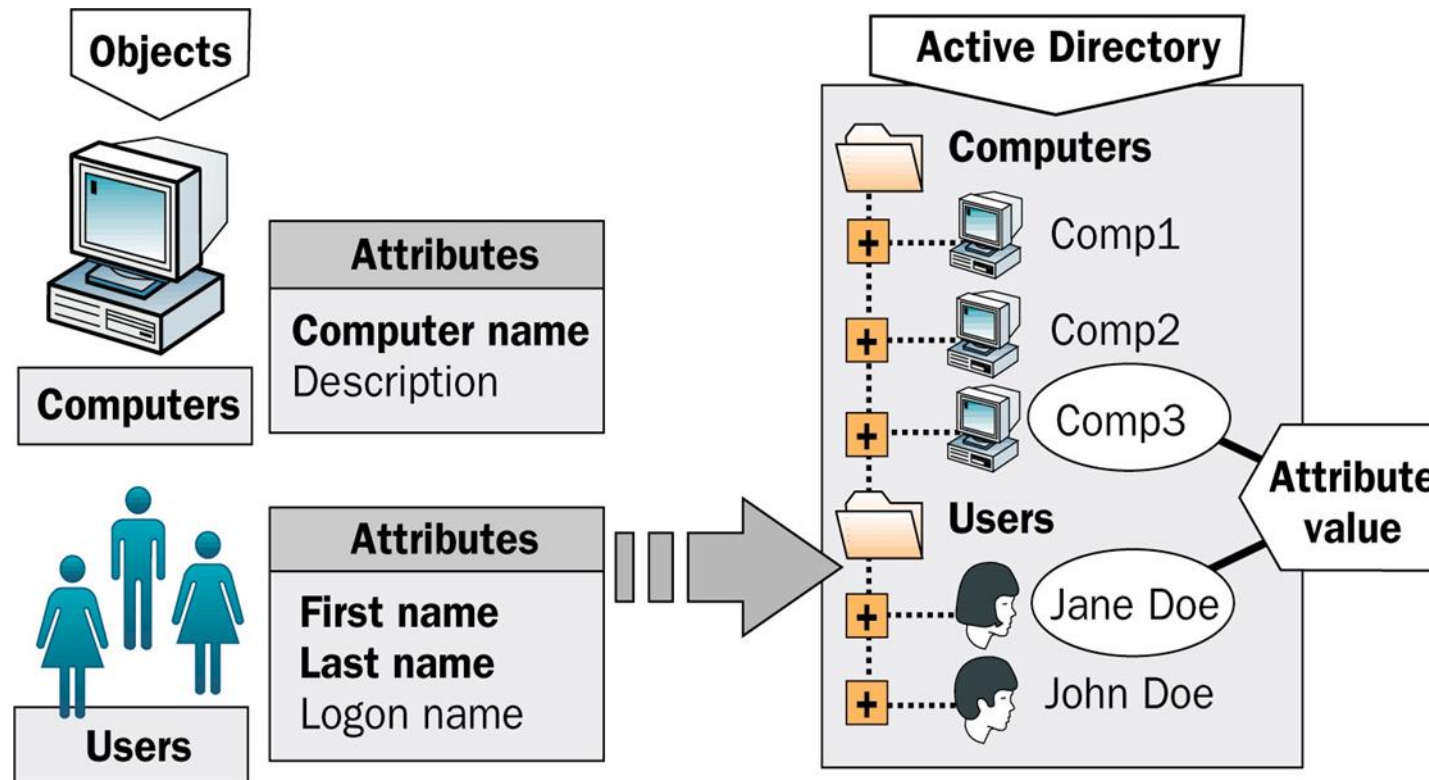
Сервіси реплікації

☐ Розповсюдження даних каталогу в мережі

- Це стосується як самого сховища даних, так і даних, необхідних для впровадження політик і конфігурації, включаючи сценарії входу.



Об'єкти та атрибути Active Directory





Визначення Active Directory

324

- ❑ Ресурси, що зберігаються в каталозі, такі як дані користувача, принтери, сервери, бази даних, групи, комп'ютери та політики безпеки, відомі як **об'єкти**.
- ❑ **Об'єкт** — це окремий іменований набір атрибутів, який представляє мережевий ресурс.
- ❑ **Атрибути** - це характеристики об'єктів у каталозі.
- ❑ Об'єкти організовані в **класи**, які є логічними групами об'єктів.
- ❑ Об'єкти, відомі як **контейнери**, можуть містити інші об'єкти.



Атрибути та класи

325

□ Атрибут:

- Визначається окремо від класів
- Визначається лише один раз і може використовуватися в кількох класах.
- Зберігає інформацію, що описує об'єкт

□ Класи:

- Це колекції атрибутів.
- Описують можливі об'єкти, які можна створити.
- Їх також називають класами об'єктів.
- Кожен об'єкт є екземпляром класу об'єктів.



Компоненти Active Directory

326

□ Логічна структура

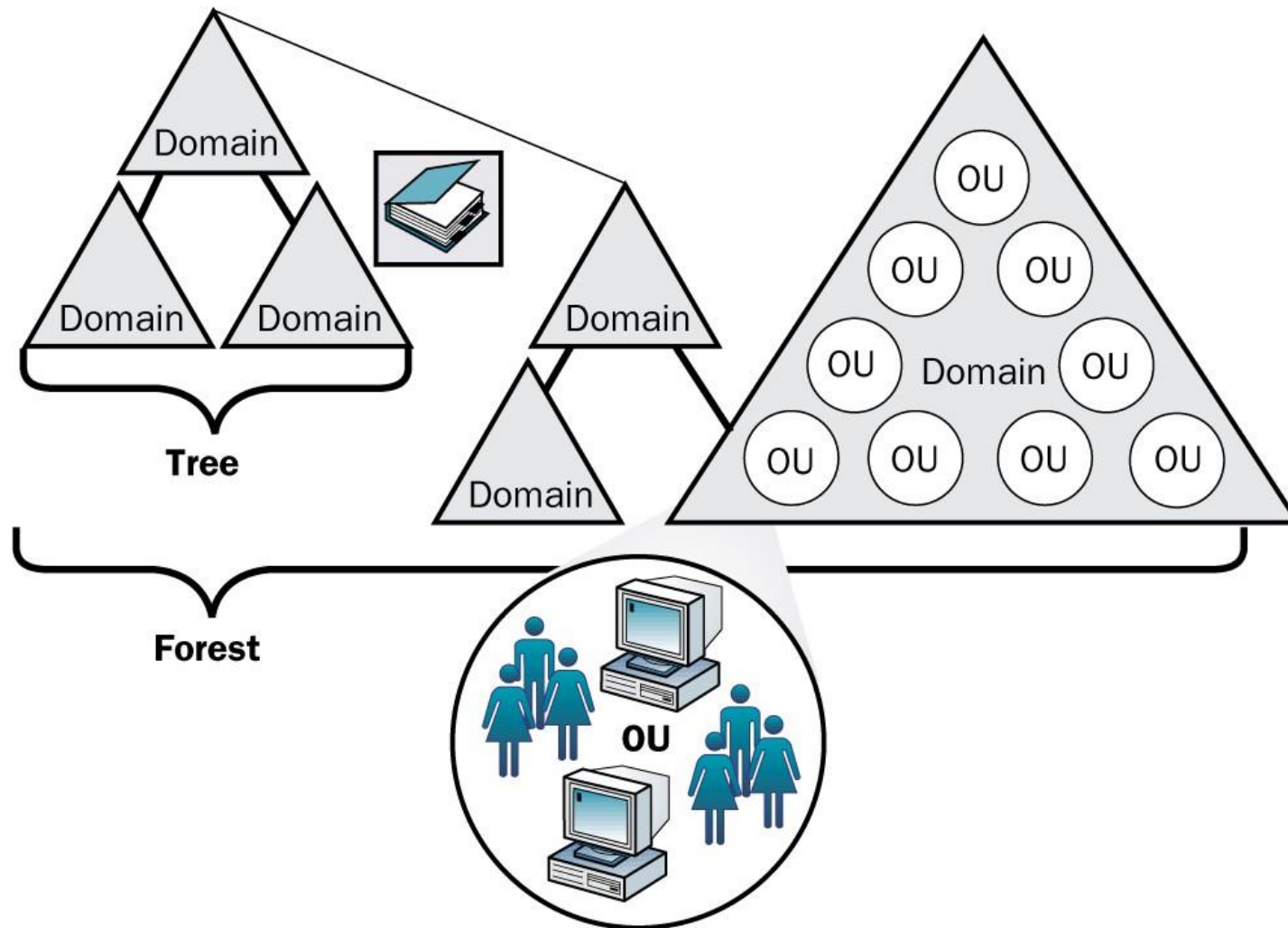
- Домени
- Організаційні одиниці
- Функціональний рівень
- Деревя
- Ліси

□ Фізична структура

- Сайти
- Контролери домену
- Сховище даних AD



Логічна ієрархічна структура





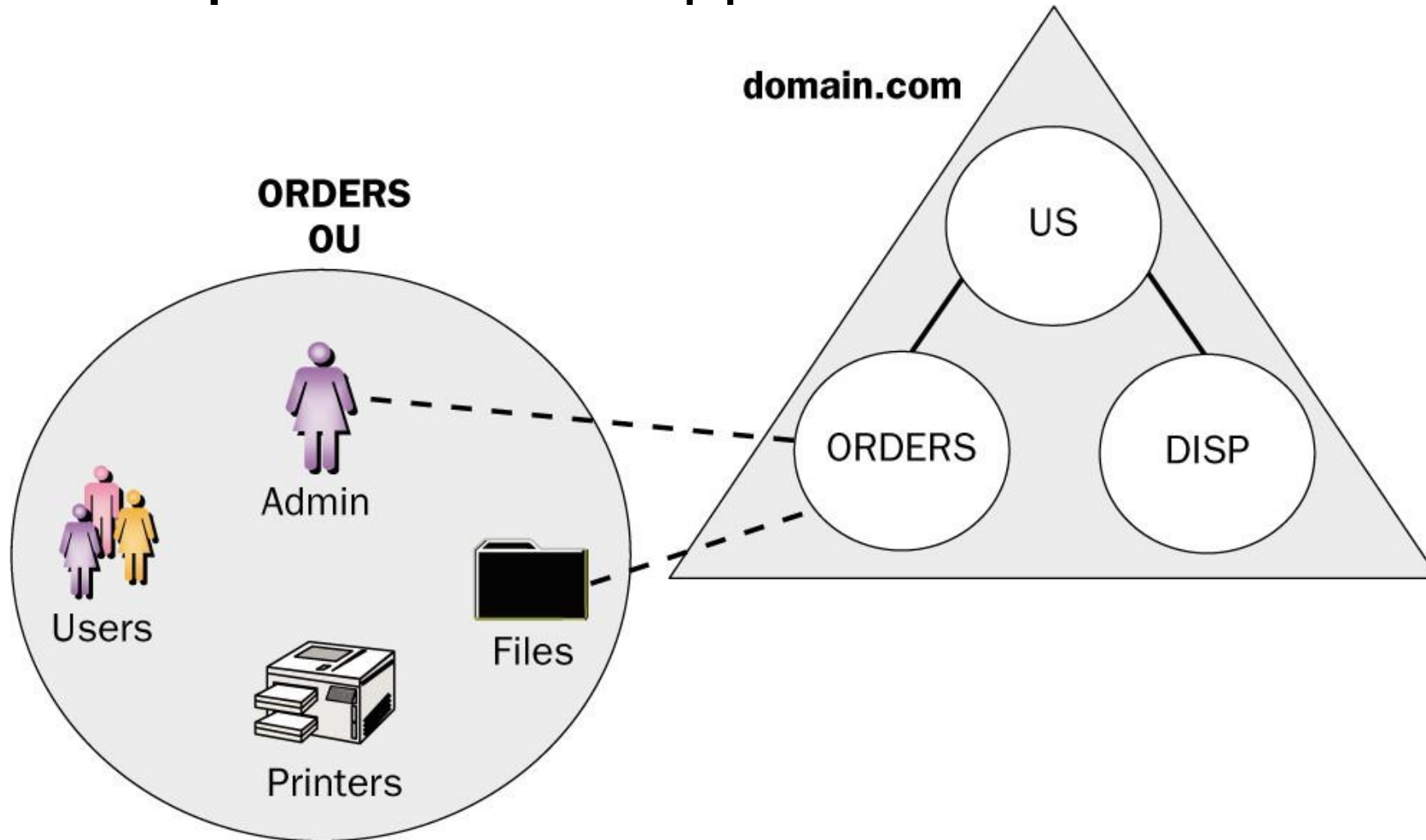
Логічна структура

328

- ☐ Ресурси повинні бути організовані в логічну структуру, яка відображає логічну структуру організації.
- ☐ Логічне групування ресурсів дозволяє користувачам і адміністраторам знаходити ресурси за назвою, а не за фізичним розташуванням.
- ☐ Фізична структура мережі прозора для користувачів.



Використання OU для виконання адміністративних завдань



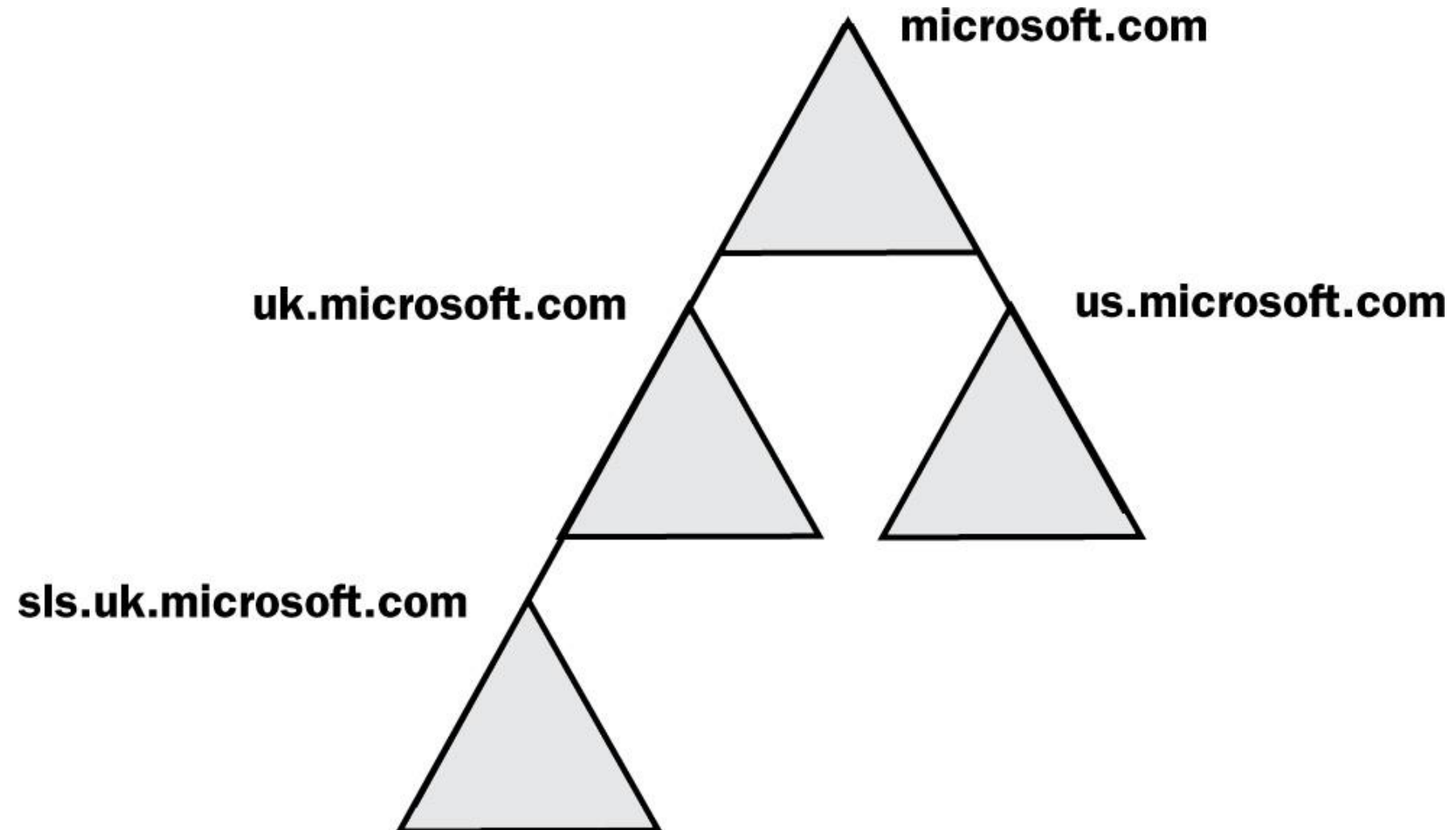


Функціональний рівень

- ☐ Функціональність, доступна в домені або лісі AD, залежить від його функціонального рівня.
- ☐ Сім функціональних рівнів домену для Windows Server 2008:
 - Windows Server 2022
 - Windows Server 2019
 - Windows Server 2016
 - Windows Server 2012 R2
 - Windows Server 2012
 - Windows Server 2008 R2
 - Windows Server 2008
- ☐ Три функціональні рівні для Windows Server 2016:
 - Windows Server 2022
 - Windows Server 2019
 - Windows Server 2016
- ☐ Функціональний рівень визначає версії Windows, дозволені на контролерах домену.

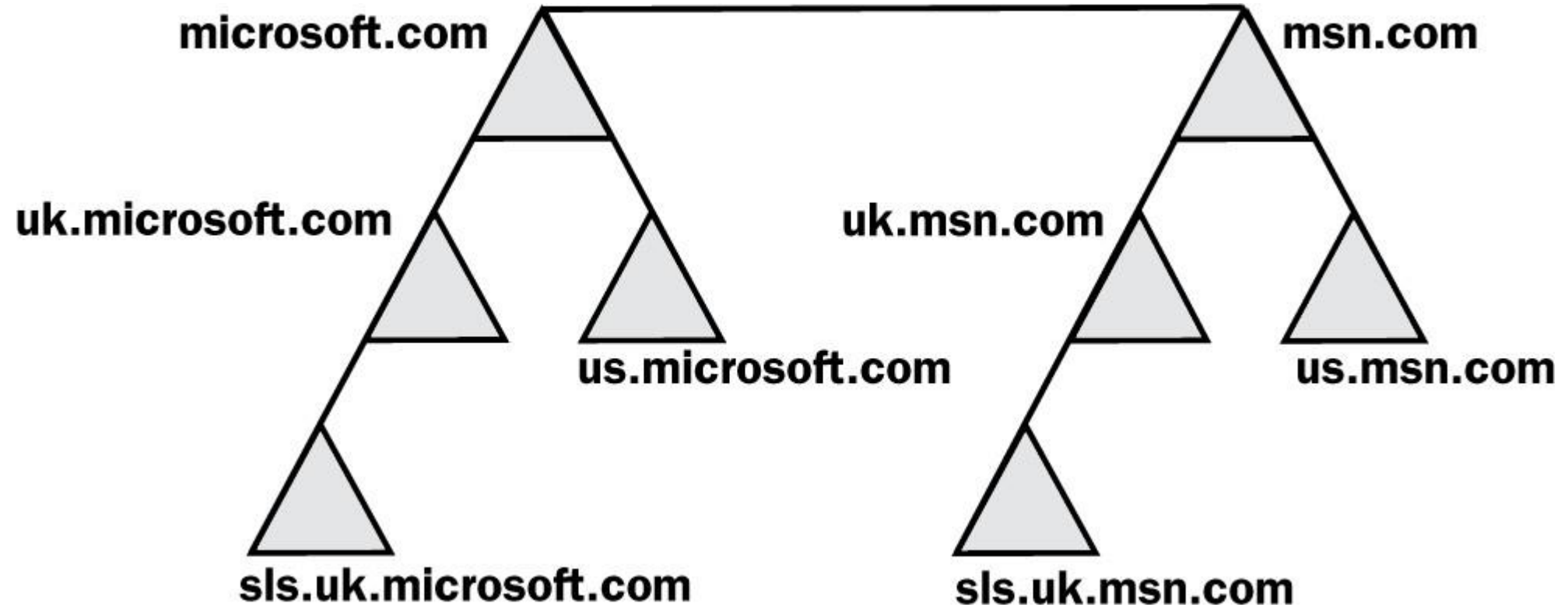


Дерево доменів





Ліс дерев





Сайти

- ☐ Комбінація однієї або кількох IP-підмереж, з'єднаних високонадійним і швидким з'єднанням, щоб локалізувати якомога більше мережевого трафіку.
- ☐ Як правило, має ті самі межі, що й локальна мережа.
- ☐ Групуючи підмережі в мережі, об'єднуйте лише ті підмережі, які мають швидкі, недорогі та надійні мережеві з'єднання одна з одною.
- ☐ Не є частиною простору імен.
- ☐ Містить лише комп'ютерні об'єкти та об'єкти підключення, які використовуються для налаштування реплікації між сайтами.

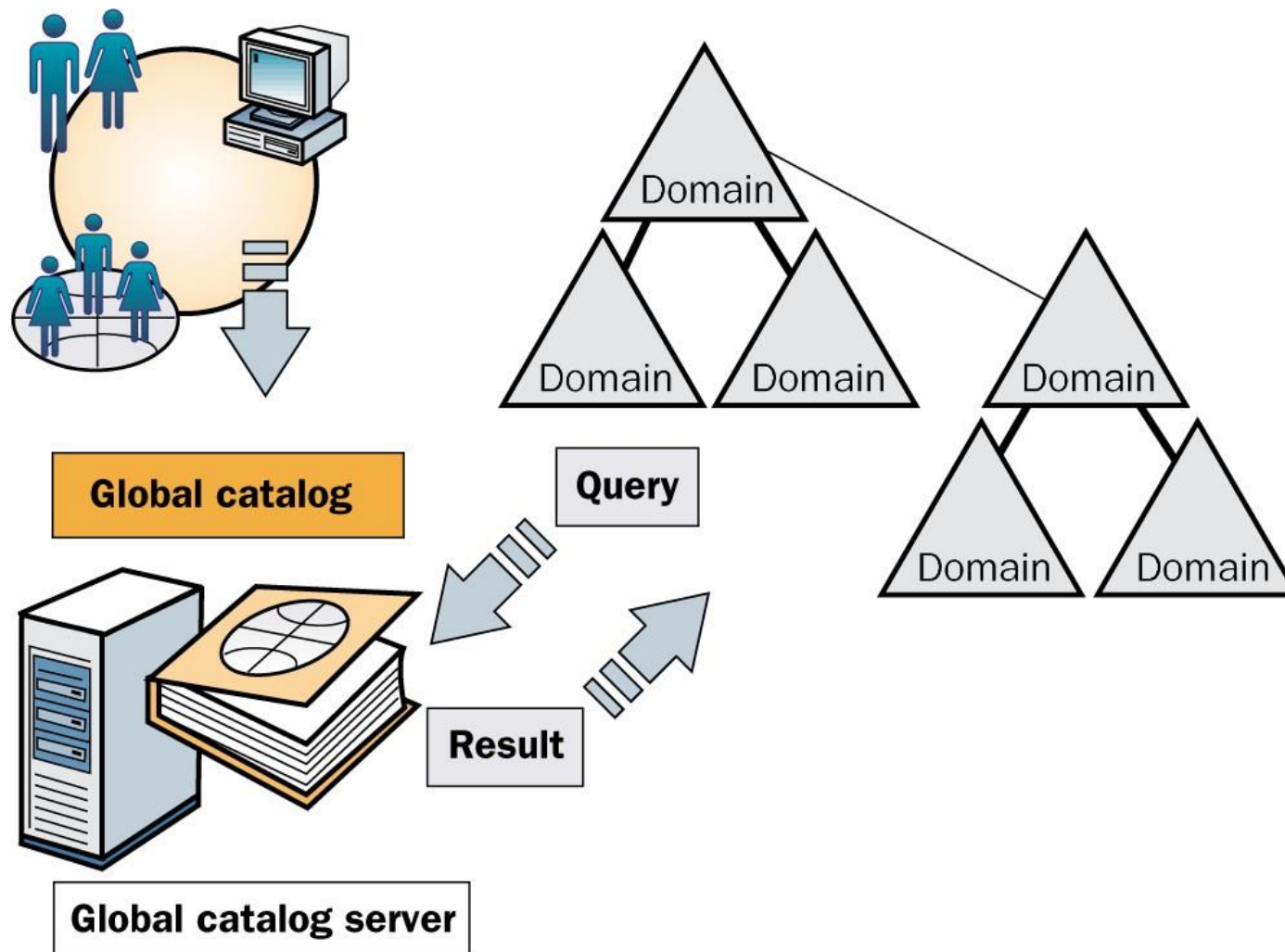


Концепції Active Directory

- ☐ Глобальний каталог
- ☐ Реплікація
- ☐ Довірчі відносини
- ☐ Простір імен DNS
- ☐ Сервери імен
- ☐ Правила іменування



Глобальний каталог





Ключові ролі каталогу

- ☐ Дозволяє вхід у мережу, надаючи інформацію про членство в універсальній групі контролеру домену під час запуску процесу входу.
- ☐ Дозволяє знаходити інформацію каталогу незалежно від того, який домен у лісі насправді містить дані.

Універсальне членство в групі

- ☐ Якщо в домені існує лише один контролер домену, контролер домену та глобальний каталог є одним і тим же сервером.
- ☐ Якщо в мережі існує кілька контролерів домену, глобальний каталог — це контролер домену, налаштований як такий.
- ☐ Якщо глобальний каталог недоступний, в разі, якщо користувач ініціює процес входу в мережу, він зможе увійти лише на локальному комп'ютері.



Розділи каталогу

□ Інформація про схему

- Визначає об'єкти, які можна створити в каталозі, і атрибути, пов'язані з цими об'єктами.

□ Інформація про конфігурацію

- Описує логічну структуру розгортання, що містить таку інформацію, як структура домену або топологія реплікації.
- Загальна для всіх доменів у дереві чи лісі доменів.

□ Дані домену

- Описує всі об'єкти в домені.
- Специфічна для домену та не поширюється на інші домени.
- Підмножина властивостей для всіх об'єктів у всіх доменах зберігається в глобальному каталозі.



Контролер доменів

□Зберігає та реплікує:

- Інформацію про схему для дерева або лісу домену.
- Інформацію про конфігурацію для всіх доменів у дереві або лісі доменів.
- Усі об'єкти каталогу та властивості для його домену.
- Підмножину властивостей усіх об'єктів у домені (репліковану до глобального каталогу)



Глобальний каталог

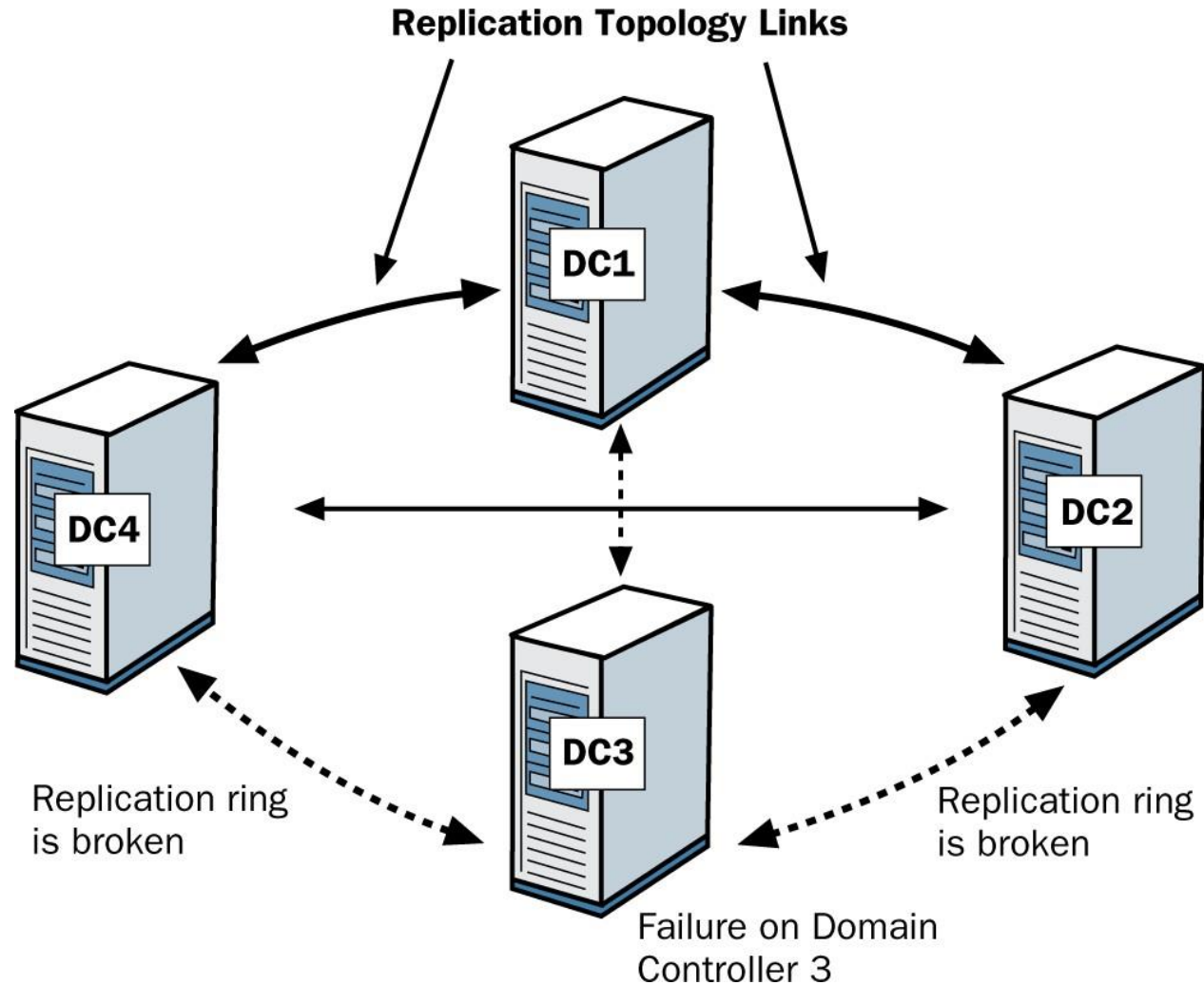
339

□Зберігає та реплікує:

- Інформацію про схему для лісу.
- Інформацію про конфігурацію для всіх доменів у лісі.
- Підмножину властивостей для всіх об'єктів каталогу в лісі (тиражується лише між серверами глобального каталогу).
- Усі об'єкти каталогу та всі їхні властивості для домену, у якому розташований глобальний каталог.



Топологія реплікації





Реплікація в межах сайту

- ☐ Active Directory автоматично створює топологію для реплікації між контролерами домену в одному домені за допомогою кільцевої структури.
- ☐ Топологія визначає шлях для передачі оновлень каталогу від одного контролера домену до іншого, доки всі контролери домену не отримають оновлення каталогу.
- ☐ Кільцева структура гарантує існування принаймні двох шляхів реплікації від одного контролера домену до іншого.
- ☐ Active Directory періодично аналізує топологію реплікації на сайті, щоб переконатися, що вона все ще ефективна.
- ☐ Якщо контролер домену додається або видаляється з мережі чи сайту, Active Directory переналаштовує топологію відповідно до змін.



Реплікація між сайтами

- ☐ Щоб забезпечити реплікацію між сайтами, Active Directory має бути налаштована на реплікацію інформації за допомогою посилань на сайти для представлення мережеских з'єднань.
- ☐ Active Directory використовує інформацію про мережеве підключення для створення об'єктів підключення, які забезпечують ефективну реплікацію та відмовостійкість.
- ☐ Надається інформація про використовуваний протокол реплікації, вартість посилання на сайт, час, коли посилання доступне для використання, і частоту використання посилання.
- ☐ Active Directory використовує цю інформацію, щоб визначити, яке посилання на сайт використовуватиметься для копіювання інформації.



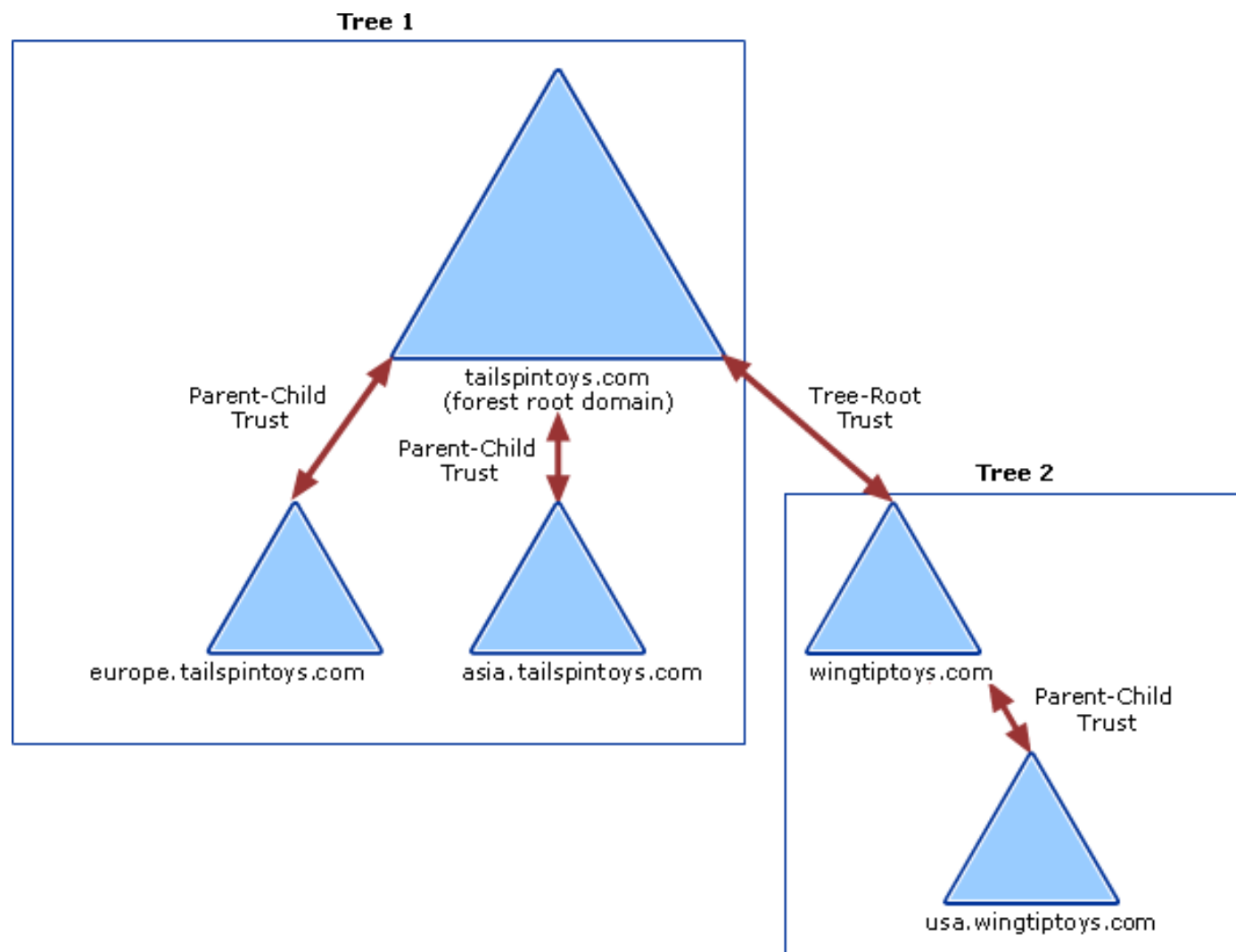
Потоки відношень довіри (трастів)

- ☐ Потік захищених комунікацій через трасти визначає еластичність трасту.
- ☐ Те, як створюється або налаштовується довіра, визначає, наскільки далеко поширюється взаємодія у межах лісу чи між ними.
- ☐ Потік взаємодії через трасти визначається напрямком трасту.
- ☐ Трасти можуть бути односторонніми та двосторонніми, а також транзитивними та нетранзитивними.



Трасти

344





Односторонні та двосторонні трасти

- ☐ Довірчі відносини дозволяють отримати доступ до ресурсів як в односторонньому, так і двосторонньому порядках.
- ☐ Одностороння довіра — це односпрямований шлях автентифікації, створений між двома доменами. В односторонній довірі між доменом А та доменом В користувачі домену А можуть отримати доступ до ресурсів домену В. Однак користувачі домену В не можуть отримати доступ до ресурсів домену А.
- ☐ У двосторонній довірі домен А довіряє домену В, а домен В довіряє домену А. Ця конфігурація означає, що запити на автентифікацію можна передавати між двома доменами в обох напрямках.
- ☐ Усі трасти домену в лісі AD DS є двосторонніми транзитивними трастами.
- ☐ Коли створюється новий дочірній домен, між новим дочірнім доменом і батьківським доменом автоматично створюється двостороння транзитивна довіра.



Транзитивні та нетранзитивні трасти

- ☐ Транзитивність визначає, чи можна розширити довіру за межі двох доменів, з якими вона була створена.
- ☐ Транзитивну довіру можна використовувати для розширення довірчих відносин з іншими доменами.
- ☐ Нетранзитивний довірчий зв'язок можна використовувати для заборони довірчих відносин з іншими доменами.
- ☐ Транзитивні довірчі відносини протікають вгору через дерево доменів у міру його формування, створюючи транзитивні довірчі відносини між усіма доменами в дереві доменів.
- ☐ Запити на автентифікацію слідують цим довірчим шляхам, тому облікові записи з будь-якого домену в лісі можуть бути автентифіковані будь-яким іншим доменом у лісі.
- ☐ За допомогою процесу єдиного входу (**single sign in**) облікові записи з відповідними дозволами можуть отримати доступ до ресурсів у будь-якому домені в лісі.

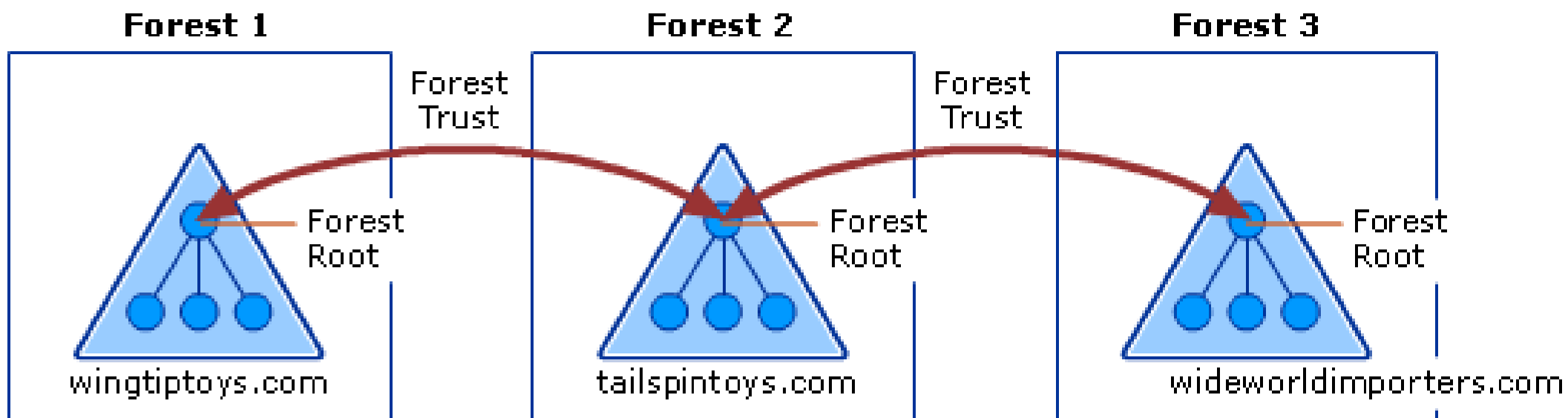


Трасти між лісами

- ☐ Лісові трасти допомагають керувати сегментованою інфраструктурою AD DS і підтримують доступ до ресурсів та інших об'єктів у кількох лісах.
- ☐ Лісові трасти корисні для постачальників послуг, компаній, які здійснюють злиття або поглинання, спільних бізнес-екстранетів і компаній, які шукають рішення для адміністративної автономії.
- ☐ Використовуючи лісові трасти, можна зв'язати два різні ліси, щоб сформувати односторонній або двосторонній транзитивний довірчий зв'язок.
- ☐ Довіра до лісу дозволяє адміністраторам з'єднати два ліси AD DS за допомогою єдиного довірчого зв'язку, щоб забезпечити безперебійну автентифікацію та авторизацію в лісах.
- ☐ Довіру до лісу можна створити лише між кореневим доменом лісу в одному лісі та кореневим доменом лісу в іншому лісі.
- ☐ Лісові трасти можуть бути створені лише між двома лісами і не можуть бути неявно поширені на третій ліс.

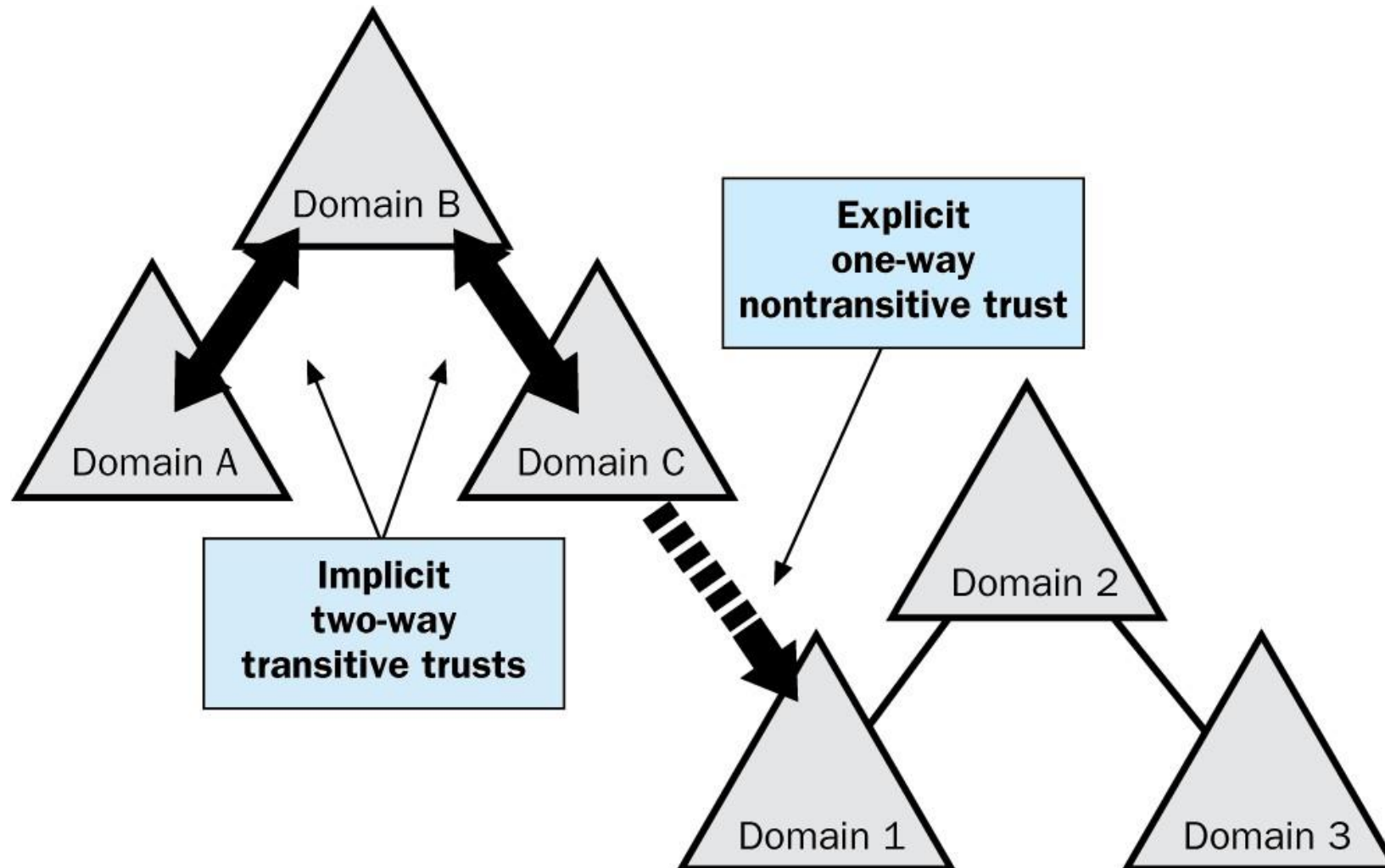


Лісовий траст (приклад)





Два типи довірчих відносин





Неявна двостороння транзитивна довіра (Implicit Two-Way Transitive Trust)

350

- ☐ Довірчі відносини між батьківським і дочірнім доменами в дереві та між доменами верхнього рівня в лісі.
- ☐ Встановлюється та підтримується автоматично.
- ☐ Функція протоколу автентифікації Kerberos.
- ☐ Якщо домен А довіряє домену В, а домен В довіряє домену С, то домен А довіряє домену С.



Явна одностороння нетранзитивна довіра (Explicit One-Way Nontransitive Trust)

- ☐ Довірчі відносини між доменами, які не є частиною одного дерева.
- ☐ Обмежена двома доменами в довірчих відносинах і не перетікає в інші домени в лісі.
- ☐ Це єдина можлива форма довіри між:
 - Доменом Microsoft Windows Server 2000/2003/2008/2012/2016/2019/2022 і доменом Windows NT.
 - Доменом Windows Server 2000/2003/2008/2012/2016/2019/2022 в одному лісі та доменом Windows Server 2000/2003/2008/2012/2016/2019/2022 в іншому лісі.
 - Доменом Windows Server 2000/2003/2008/2012/2016/2019/2022 і областю MIT Kerberos V5.



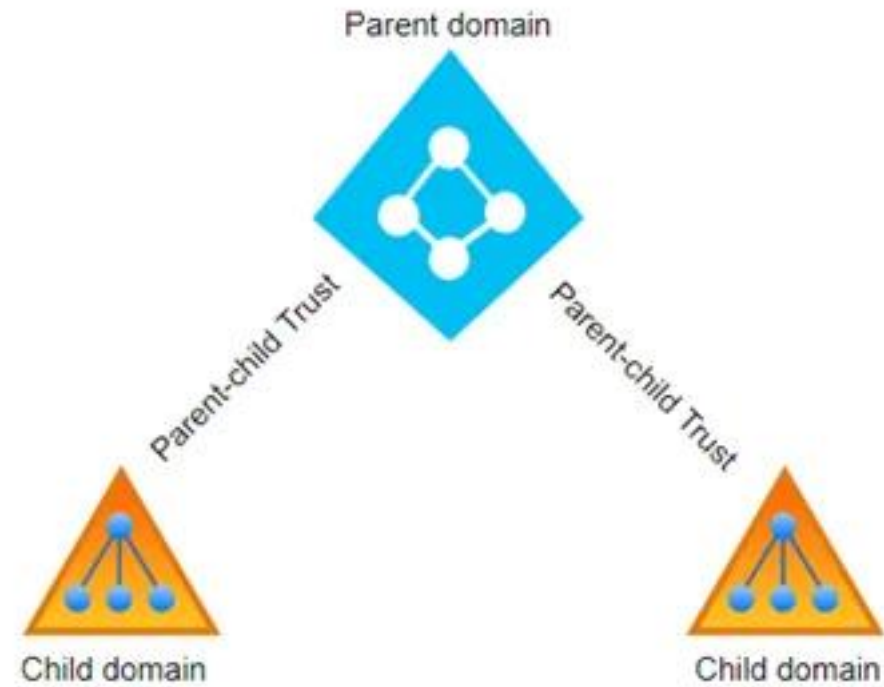
Резюме – 5 типів трастів

- ☐ Parent-child
- ☐ Tree-Root
- ☐ Forest
- ☐ Shortcut
- ☐ Realm
- ☐ External



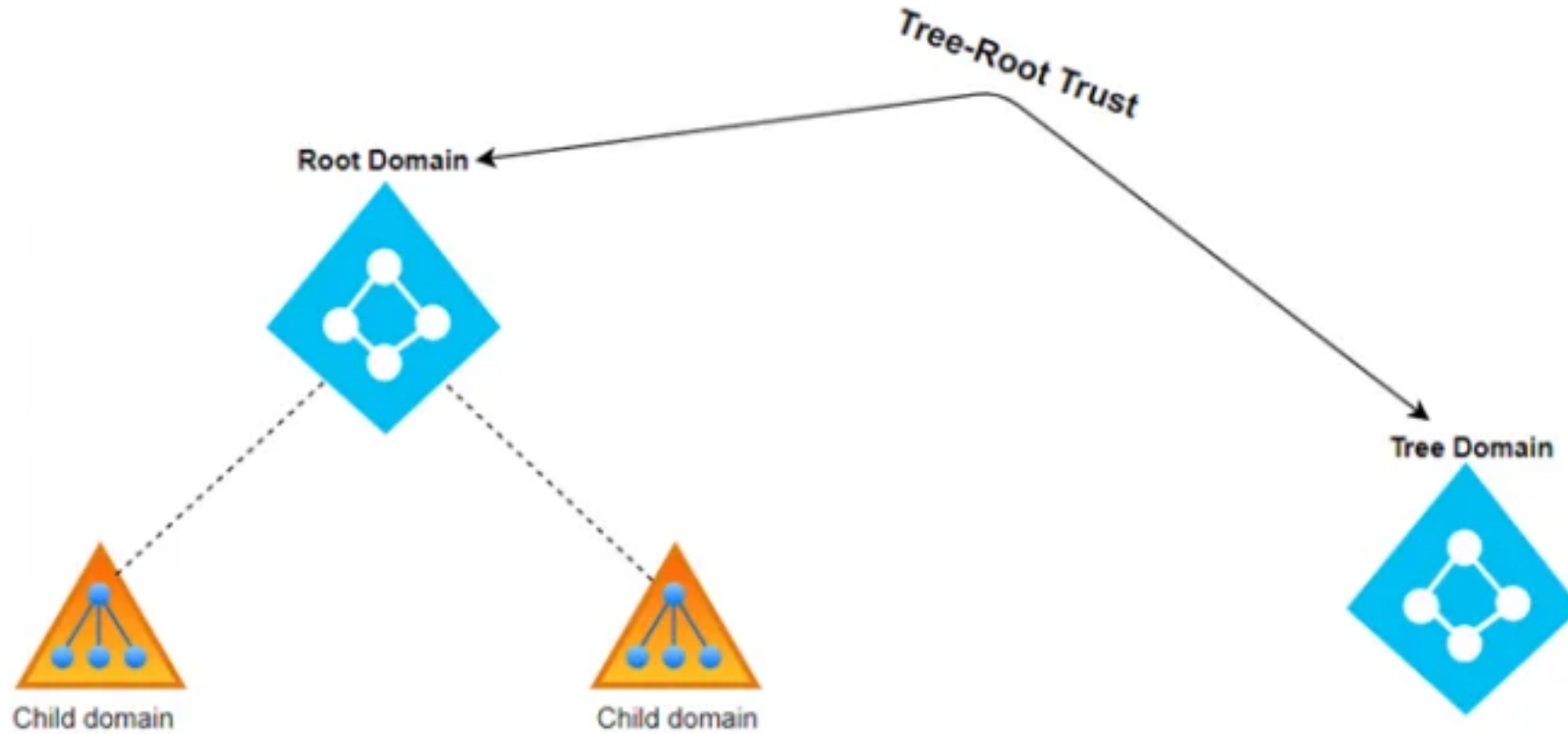
Траст Parent-child

353



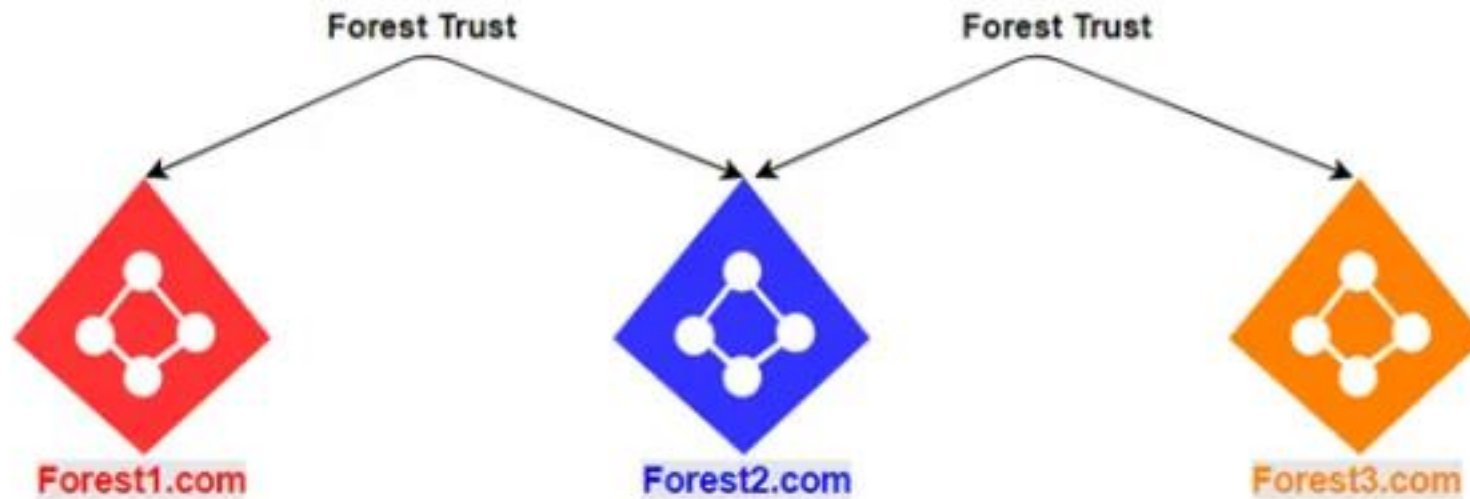


Траст Tree-Root



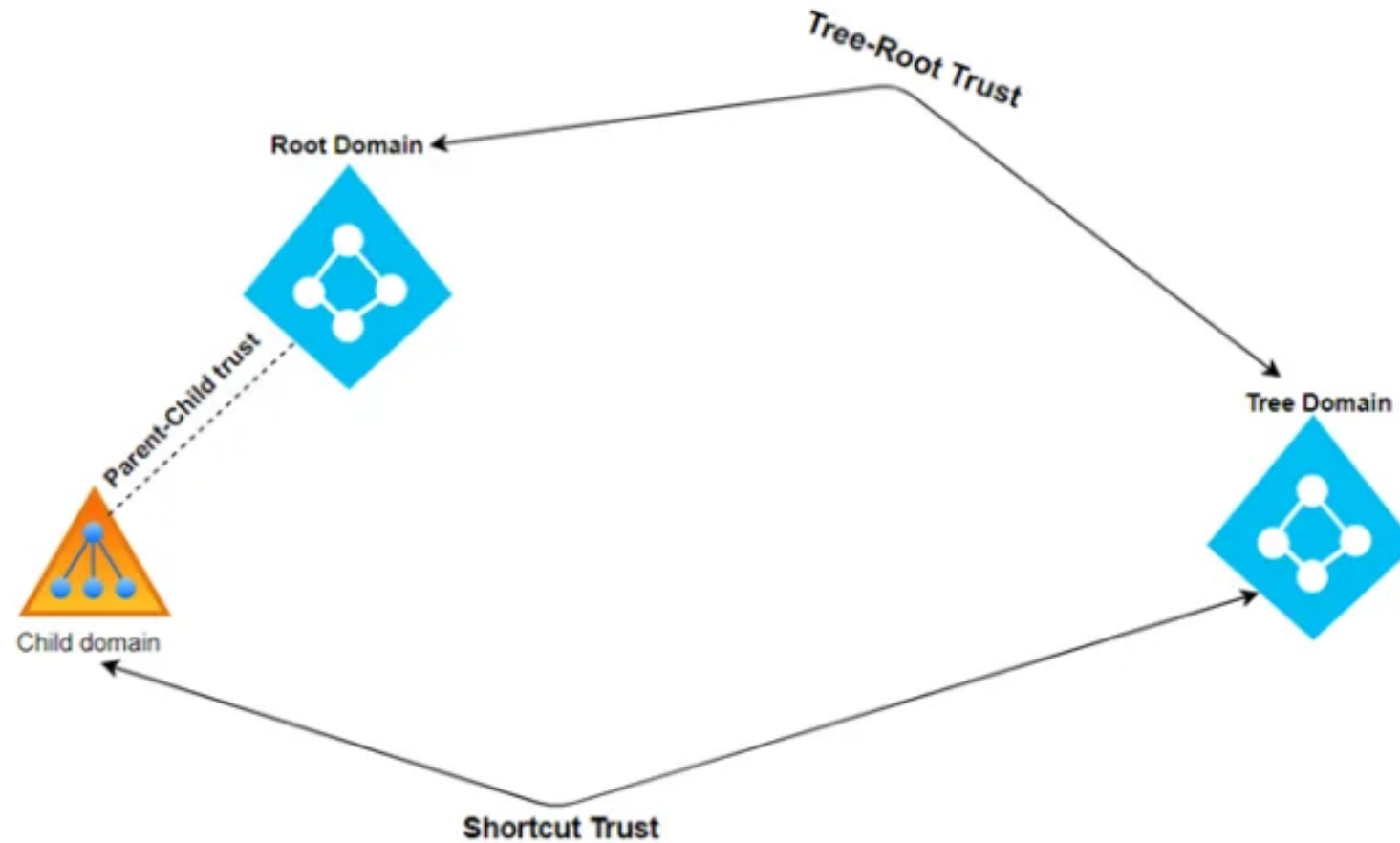


Траст Forest Trust





Траст Shortcut





Траст External





Траст Realm

- ❑ Ці види довіри між доменом або лісом з іншим доменом і лісом, який не базується на Windows Active Directory.
- ❑ Realm Trust може бути встановлений для забезпечення доступу до ресурсів і крос-платформної сумісності між доменом AD DS і не-windows Kerberos v5 Realm.

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 11. Групові політики Windows.

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- Архітектура та обробка об'єктів групової політики
- Конфігурація налаштувань групової політики
- Робота з шаблонами безпеки
- Управління та моніторинг групових політик
- Налаштування параметрів групової політики



Архітектура групової політики

□ Архітектура та функції групової політики включають такі компоненти:

- GPO

- Об'єкт, що містить параметри політики, які впливають на робоче середовище та безпеку користувача та комп'ютера. Можуть бути локальними або об'єктами AD

- Реплікація

- Гарантує, що всі контролери домену мають поточну копію кожного GPO

- Сфера застосування та успадкування

- Область дії групової політики визначає, на яких користувачів і комп'ютери впливають її налаштування

- Створення та зв'язування

- Об'єкти групової політики створюються на консолі керування груповою політикою та можуть бути пов'язані з одним або кількома контейнерами AD



Об'єкти групової політики (GPO)

- ❑ Об'єкт групової політики містить параметри політики для керування багатьма аспектами контролерів домену, рядових серверів, рядових комп'ютерів і користувачів
- ❑ Два основних типи GPO:
 - Локальні GPO
 - Об'єкти групової політики домену



Локальні GRO

- ☐ Локальні GRO зберігаються на локальних комп'ютерах і редагуються за допомогою оснастки редактора об'єктів групової політики.
- ☐ Налаштування в локальних об'єктах групової політики, успадкованих від об'єктів групової політики домену, не можна змінити на локальному комп'ютері.
- ☐ Лише параметри, які не визначені або не налаштовані GRO домену, можна редагувати локально.



Нові локальні GPO у Windows Vista та Server 2008

- ❑ Нові політики дозволяють установлювати різні політики залежно від того, хто входить до комп'ютера:
 - Local Administrators GPO
 - Local Non-Administrators GPO
 - User-specific GPO
- ❑ Якщо ці політики використовуються, вони обробляються в наведеному вище порядку, особливо для вирішення конфліктів (останній параметр політики має пріоритет)



Об'єкти групової політики домену

- ☐ Об'єкти групової політики домену зберігаються в Active Directory на контролерах домену.
- ☐ Складається з двох окремих частин: шаблону групової політики (GPT) і контейнера групової політики (GPC).
- ☐ GPT і GPC мають спільну структуру імен і структуру папок.
- ☐ Для вирішення проблем важливо знати структуру GPO.



Шаблони групової політики

- ❑ Шаблон групової політики містить усі параметри політики, які складають GPO, а також пов'язані файли, наприклад сценарії, і міститься в спільному ресурсі **Sysvol** на контролері домену
- ❑ Під час створення GPO створюється кілька файлів і вкладених папок (точна кількість може відрізнятися), але кожна папка GPT міститиме принаймні три елементи:
 - GPT.ini
 - Машину
 - Користувача



Контейнери групової політики

- ☐ Зберігається в папці **System\Policies**
- ☐ Містить властивості GPO та інформацію про стан, але не містить параметрів політики
- ☐ Подібний до GPT тим, що він використовує **GUID GPO** для назви папки
- ☐ Інформація, що міститься в GPC:
 - Назва GPO
 - Шлях до GPT
 - Версія
 - Статус



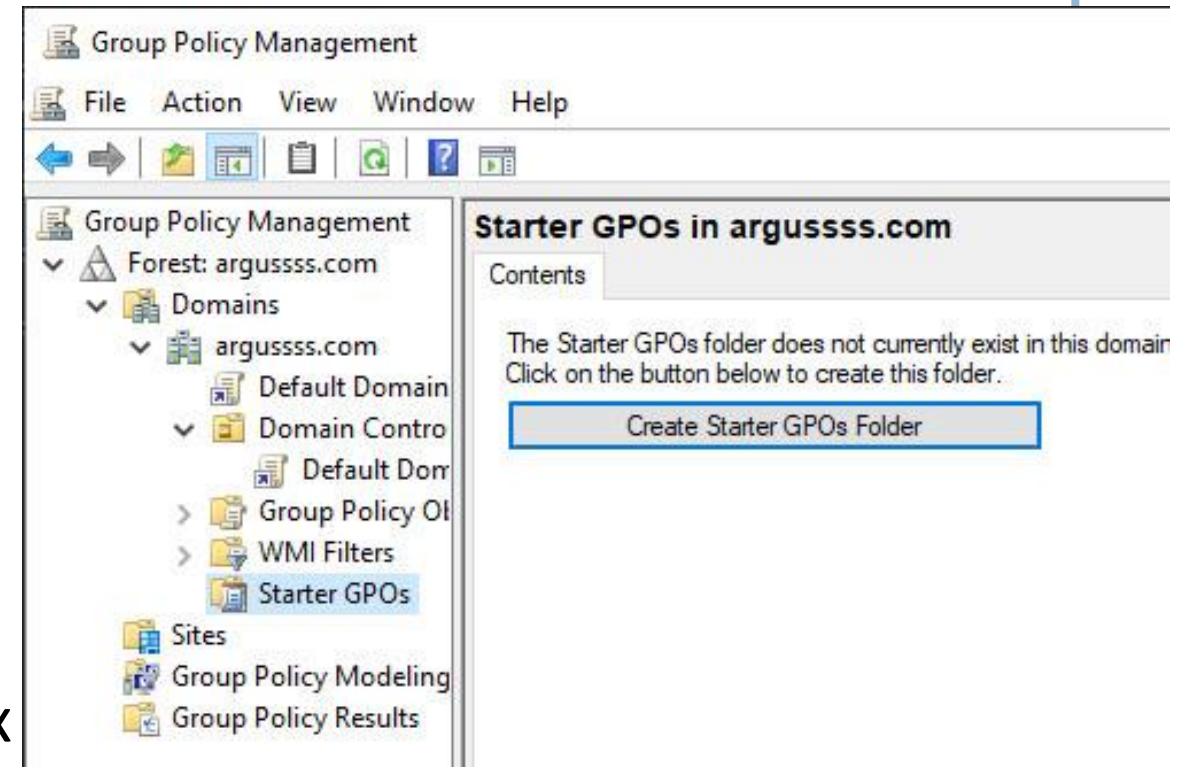
Реплікація групової політики

- ☐ GPC реплікуються за допомогою Active Directory.
- ☐ GPT реплікуються одним із таких методів:
 - Служба реплікації файлів (FRS)
 - Використовується під час роботи в змішаному середовищі різних операційних систем Windows Server
 - Реплікація розподіленої файлової системи (DFSR)
 - Використовується, коли всі DC працюють під керуванням Windows Server 2019/2022
- ☐ DFSR є ефективнішим і надійнішим.
- ☐ GPC і GPT можуть не синхронізуватися.



Створення та зв'язування GPO

- ❑ Основними інструментами для керування, створення та редагування GPO є консоль керування груповою політикою (GPMC) і редактор керування груповою політикою (GPME).
- ❑ У разі редагування GPO, який уже пов'язано з контейнером, зміни в параметрах політики набудуть чинності, щойно клієнти їх завантажуть.
- ❑ Перш ніж вводити кілька змін до політики одночасно, протестуйте їх окремо.





Редагування існуючого GPO

- ☐ Щоб відредагувати, клацніть правою кнопкою миші GPO в GRMC і клацніть «Редагувати», після чого відкриється GPO в GRME.
- ☐ Можна внести зміни в Політику домену за замовчуванням, але це не бажано.
- ☐ Рекомендований спосіб внесення змін до політик домену – створити новий GPO та зв'язати його з доменом.
- ☐ Об'єкти групової політики застосовуються до об'єктів у зворотному порядку зв'язування.

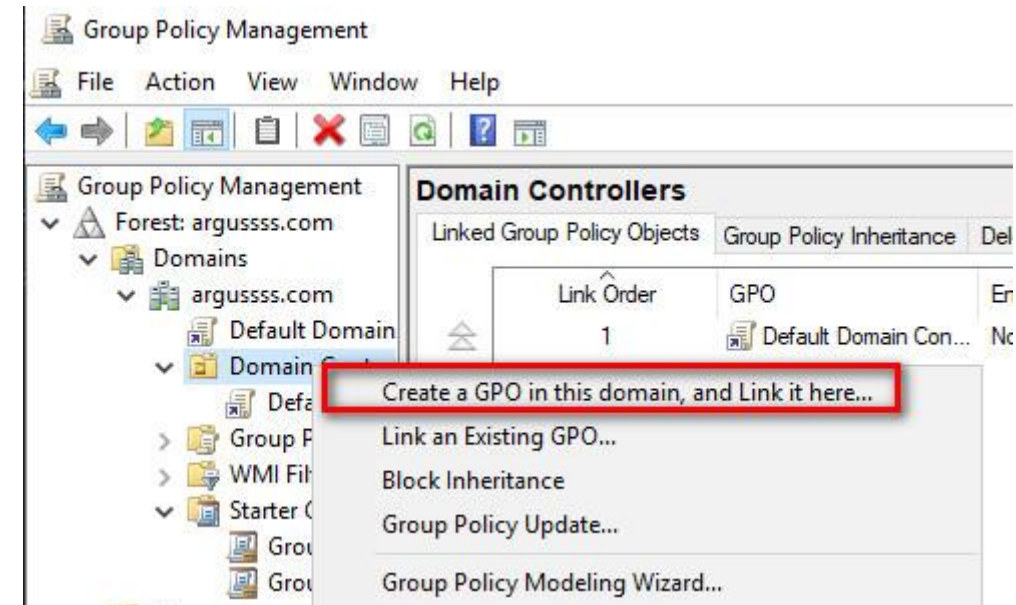


Створення нового GPO

❑ Два способи створення нового GPO за допомогою GPMC:

- Клацніть правою кнопкою миші на контейнер, з яким ви пов'язуєте GPO, і виберіть «**Create a GPO in this domain, and Link it here**».
- Клацніть правою кнопкою миші папку «Об'єкти групової політики» та натисніть «Створити».

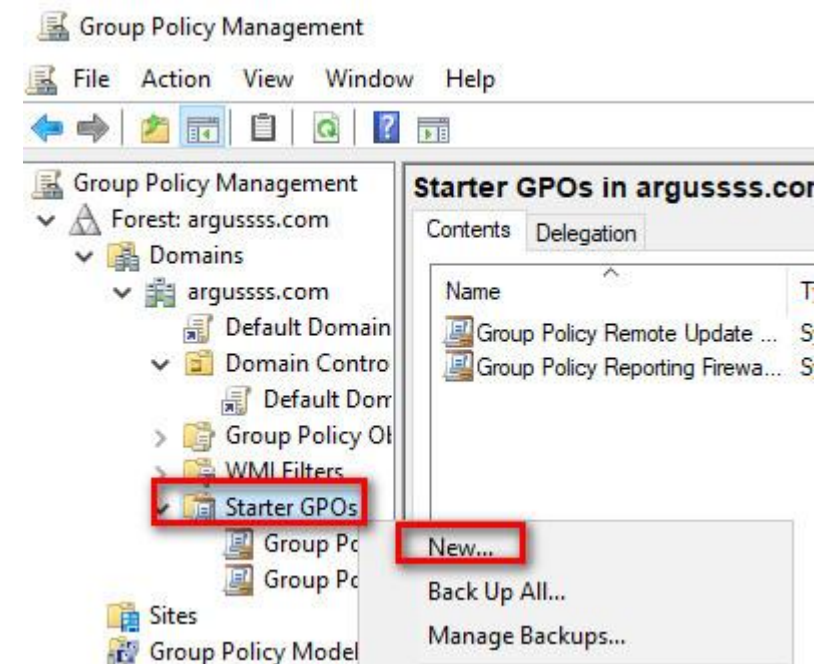
❑ Найкраще створити об'єкти групової політики, які зосереджені на категорії параметрів, а потім назвати об'єкт групової політики відповідно.





Використання об'єктів групової політики Starter

- ❑ Starter GPO — це шаблон для створення GPO (не GPT).
- ❑ Новий майстер GPO містить опцію використання Starter GPO.
- ❑ Зберігається в папці Starter GPO у GPMC.
- ❑ Щоб використати Starter GPO, виберіть його у списку Source Starter GPO у майстрі створення нових об'єктів групової політики або клацніть правою кнопкою миші на початковому об'єкті групової політики в папці початкових об'єктів групової політики та клацніть «Новий об'єкт групової політики» з початкового об'єкта групової політики.
- ❑ Щоб створити Starter GPO, клацніть правою кнопкою миші папку Starter GPO і виберіть New.





Сфера застосування та успадкування групової політики

- ❑ Область групової політики визначає, на які об'єкти в AD впливають налаштування політики.
- ❑ Якщо два об'єкти групової політики застосовано до об'єкта, а параметр налаштовано для одного GPO, але не для іншого, буде застосовано налаштований параметр.
- ❑ Політики застосовуються в такому порядку:
 - Локальна політика
 - Об'єкти групової політики, пов'язані з сайтом
 - Об'єкти групової політики, пов'язані з доменом
 - GPO, пов'язані з OU



Розуміння GRO, пов'язаних із сайтом

- ☐ Об'єкти групової політики, пов'язані з об'єктом сайту, впливають на всіх користувачів і комп'ютери, фізично розташовані на сайті.
- ☐ Можна використовувати для налаштування різних політик для мобільних користувачів.
- ☐ У середовищі окремого сайту та домену краще використовувати доменні GRO.
- ☐ Об'єкти групової політики сайтів можуть заплутати користувачів мобільних пристроїв, якщо між сайтами досить різко змінюється політика.



Розуміння доменних GPO

- ☐ Об'єкти групової політики на рівні домену повинні містити параметри, які застосовуються до всіх об'єктів у домені.
- ☐ Політику облікового запису можна визначити лише на рівні домену.
- ☐ Найкращі практики пропонують установку політик облікових записів і кількох критичних політик безпеки на рівні домену.



Розуміння GPO, пов'язаних із OU

- ☐ Точне налаштування групових політик слід виконувати на рівні OU.
- ☐ Користувачі та комп'ютери з подібними вимогами до політики мають бути розташовані в одному підрозділі.
- ☐ Оскільки OU можуть бути вкладеними, GPO також можуть бути вкладеними.
- ☐ Об'єкти групової політики, застосовані до вкладених організаційних підрозділів, слід використовувати для винятків із політик, встановлених на вищому рівні.



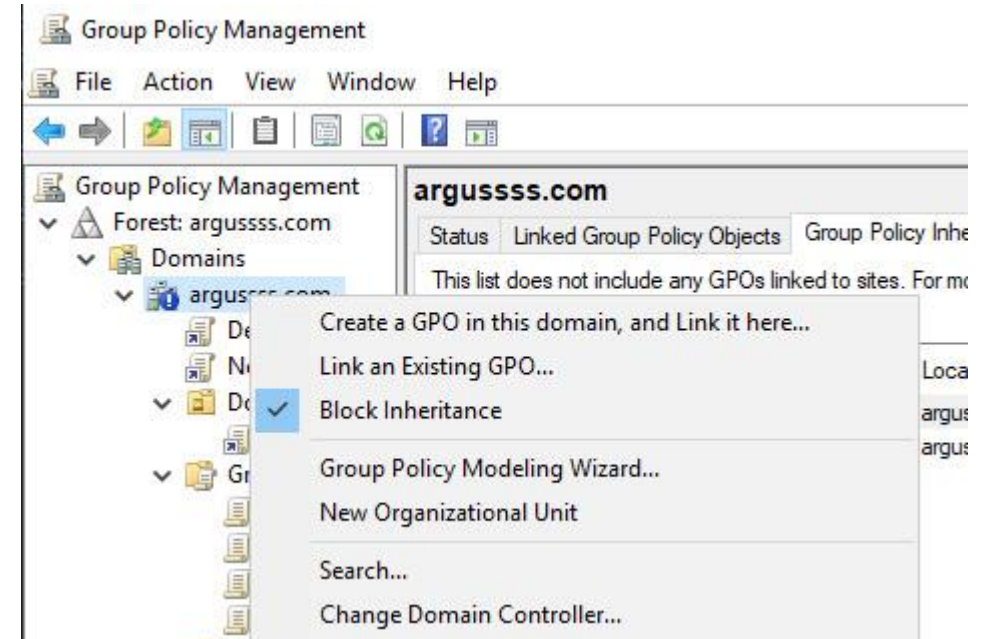
Зміна типової поведінки успадкування GPO

- ☐ Спадкування GPO увімкнено за замовчуванням.
- ☐ Щоб побачити, звідки успадковуються політики, виберіть контейнер на лівій панелі GRMC і натисніть вкладку успадкування групової політики на правій панелі.
- ☐ Є кілька способів вплинути на успадкування GPO:
 - Блокування успадкування
 - Примусове забезпечення успадкування
 - Фільтрація GPO
 - Обробка політики зворотного зв'язку



Блокування успадкування GPO

- ❑ Запобігає впливу GPO, пов'язаних із батьківськими контейнерами, на дочірні контейнери.
- ❑ Щоб заблокувати успадкування групової політики, у GPMC клацніть правою кнопкою миші дочірній домен або організаційний підрозділ і виберіть **Block Inheritance**.
- ❑ Якщо блокування ввімкнено, організаційний підрозділ або об'єкт домену відображається із **синім знаком оклику**.
- ❑ Часте блокування означає можливу несправність дизайну OU.





Примусове успадкування GPO

- ❑ Примусове успадкування групової політики скасовує будь-які конфліктні конфігурації на більш глибокому рівні.
- ❑ Якщо примусово застосовано кілька GPO, у конфлікті буде застосовано GPO найвищого рівня.
- ❑ Приклад: якщо об'єкт групової політики, пов'язаний з організаційним підрозділом, і об'єкт групової політики, пов'язаний із доменом, налаштовано на примусове застосування, об'єкт групової політики, пов'язаний із доменом, має більший пріоритет.



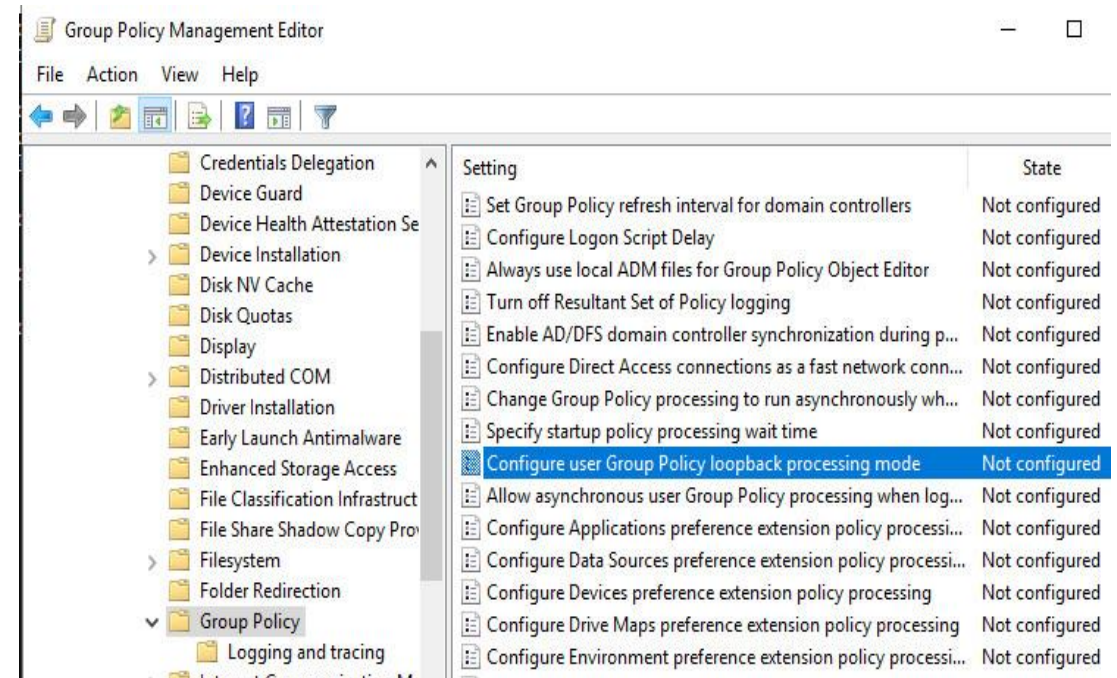
Фільтрація GPO

- ❑ Фільтрація GPO дозволяє змінювати успадкування на основі кожного об'єкта.
- ❑ Два типи фільтрації GPO:
 - Фільтрація безпеки
 - Фільтр інструментарію керування Windows (WMI).
- ❑ Фільтрація безпеки використовує дозволи для обмеження доступу об'єктів до GPO.
- ❑ Фільтрація WMI використовує запити для вибору групи комп'ютерів на основі певних атрибутів, а потім застосовує або не застосовує політики на основі результатів запиту.



Обробка політики зворотного зв'язку

- ❑ Зазвичай політики, які впливають на налаштування користувача, стежать за тим, на якому комп'ютері вони ввійшли.
- ❑ Обробка політики зворотного зв'язку дозволяє застосовувати параметри у вузлі конфігурації користувача GPO до всіх користувачів, які входять до комп'ютера.
- ❑ Щоб використовувати, увімкніть політику «User group policy loopback processing mode» у вузлі «**Configuration\Policies\Administrative Templates\System\Group Policy**».





Параметри групової політики

- ❑ Параметри в конфігурації комп'ютера мають пріоритет над параметрами в конфігурації користувача, якщо виникне конфлікт.
- ❑ Три папки в папці «Політики»:
 - Налаштування програмного забезпечення (**Software Settings**)
 - Параметри Windows (**Windows Settings**)
 - Адміністративні шаблони (**Administrative Templates**)
- ❑ Параметри політики можуть бути керованими або некерованими.
 - Керовані політики скидаються до «не налаштовано», коли об'єкт виходить за межі політики
 - Некеровані політики є постійними



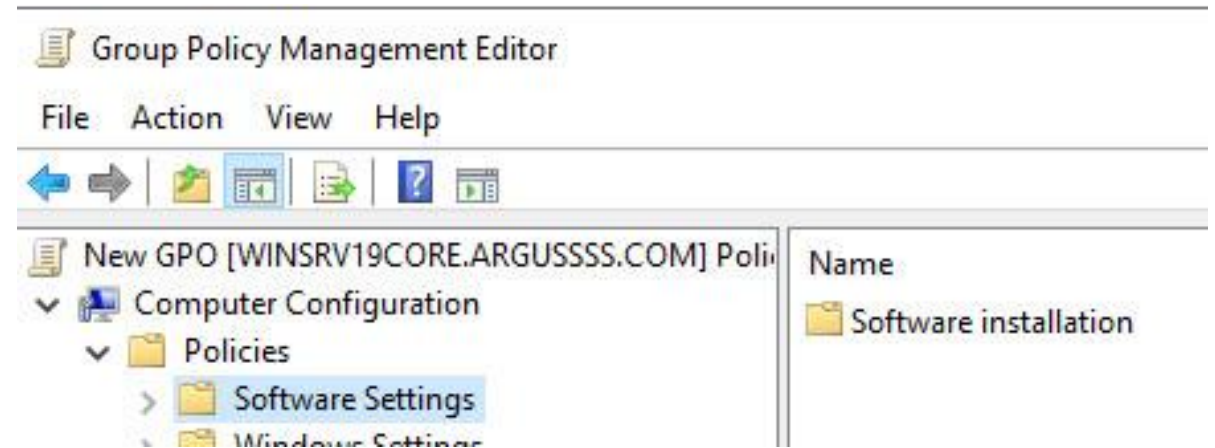
Політики у вузлі конфігурації комп'ютера

- ☐ Застосовується до комп'ютерів незалежно від того, хто входить до комп'ютера.
- ☐ Містить більшість налаштувань, пов'язаних із безпекою, у вузлах «Політика облікового запису», «Призначення прав користувача», «Політика аудиту» та «Параметри безпеки».
- ☐ Політики конфігурації комп'ютера завантажуються на комп'ютер під час запуску ОС і оновлюються кожні 90 хвилин після цього.
- ☐ Деякі зміни політики можуть потребувати перезапуску.



Конфігурація комп'ютера: налаштування програмного забезпечення

- ☐ Містить розширення Software Installation, яке можна налаштувати для дистанційного встановлення пакетів програмного забезпечення.
- ☐ Програми розгортаються за допомогою служби Windows Installer, яка використовує файли MSI.
- ☐ Пакети програмного забезпечення призначаються цільовим комп'ютерам, що робить встановлення обов'язковим під час наступного запуску комп'ютера.





Розширені параметри розгортання програми

- ☐ Під час розгортання програм натисніть кнопку «Додатково» в діалоговому вікні «Розгортання програмного забезпечення».
- ☐ Відкриється вікно властивостей із такими вкладками:
 - Вкладка «Розгортання»
 - Вкладка «Оновлення».
 - Вкладка «Категорії»
 - Вкладка «Зміни»
- ☐ Якщо в пакет внесено зміни, він не встановлюється знову за замовчуванням. Однак пакет можна легко повторно розгорнути.

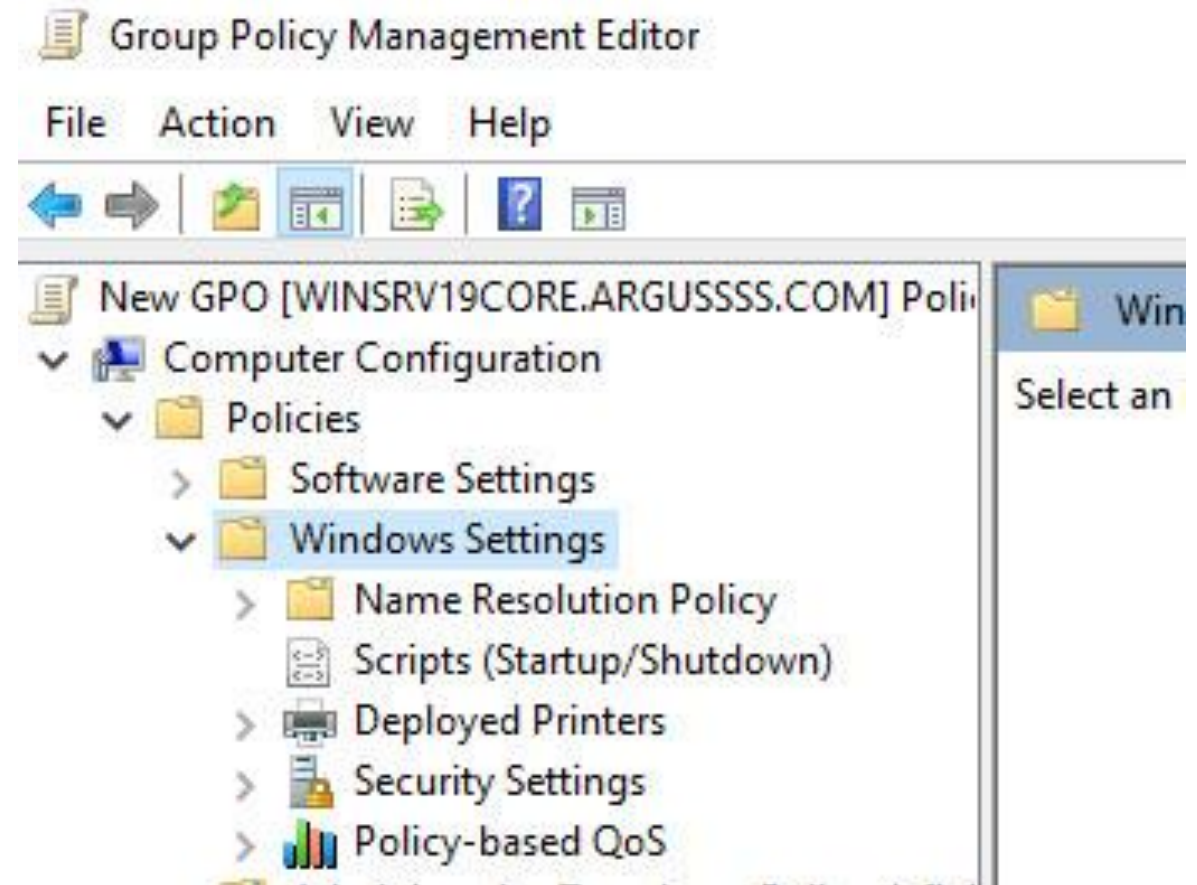


Конфігурація комп'ютера: налаштування Windows

□ Папка Windows Settings

містить чотири підвузли:

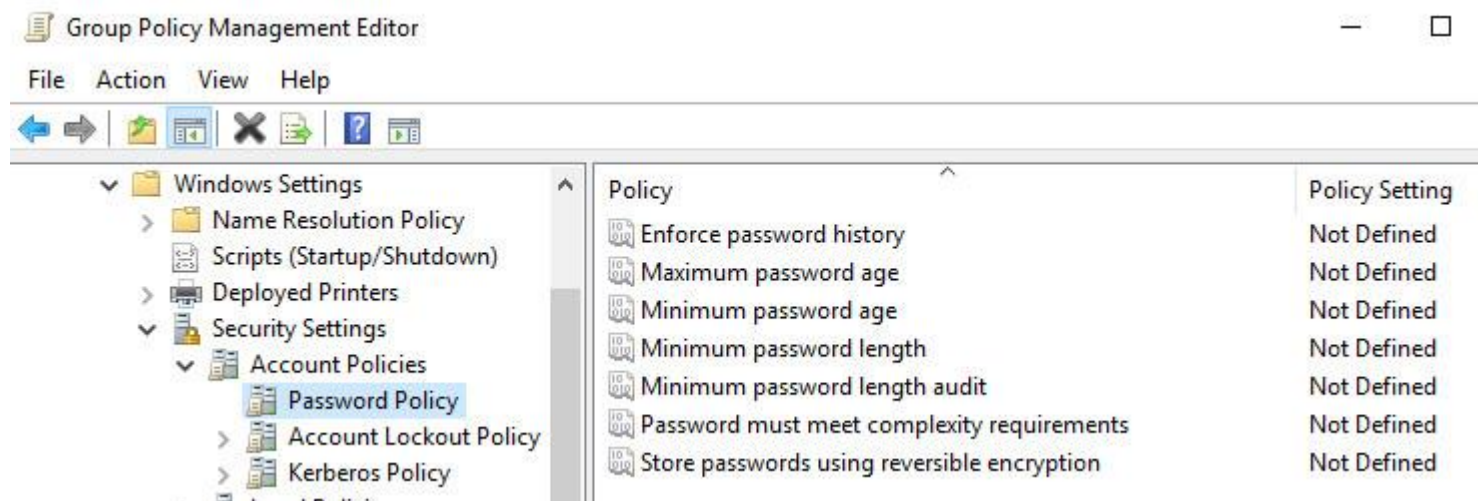
- Сценарії (запуск/вимкнення)
 - Дозволяє створювати сценарії для запуску під час запуску або завершення роботи
- Розгорнуті принтери
 - Можна розгортати принтери на комп'ютері, вказавши шлях UNC до спільного принтера
- Параметри безпеки
 - Містить вузли для встановлення політик безпеки, наприклад тих, що стосуються облікових записів
- QoS на основі політики
 - Дозволяє адміністраторам керувати використанням пропускну здатності мережі





Підвузол налаштувань безпеки: правила облікового запису₁

- ❑ Щоб діяти, правила облікового запису мають бути пов'язані з доменом.
- ❑ Політика облікового запису містить три підвузли:
 - Політика паролів
 - Застосовувати історію паролів
 - Максимальний термін дії пароля
 - Мінімальний термін дії пароля
 - Мінімальна довжина пароля
 - Пароль має відповідати вимогам складності
 - Зберігати паролі за допомогою оборотного шифрування

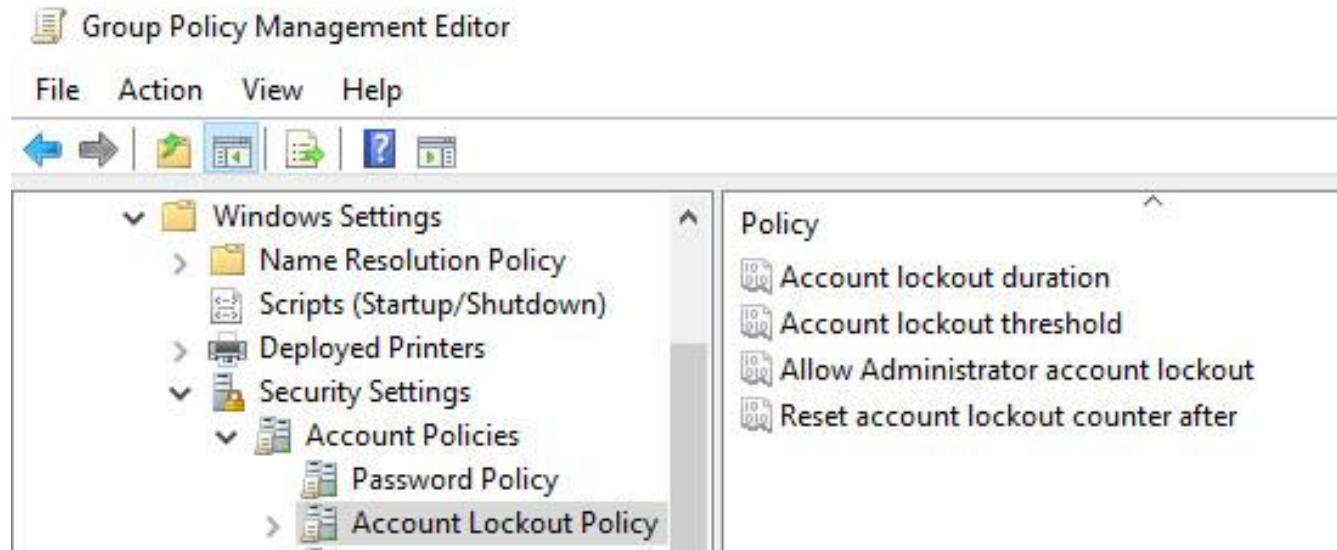




Підвузол налаштувань безпеки: правила облікового запису₂

- Політика блокування облікового запису
 - Тривалість блокування облікового запису
 - Поріг блокування облікового запису
 - Дозволити блокування облікового запису адміністратора
 - Скинути лічильник блокувань облікового запису після

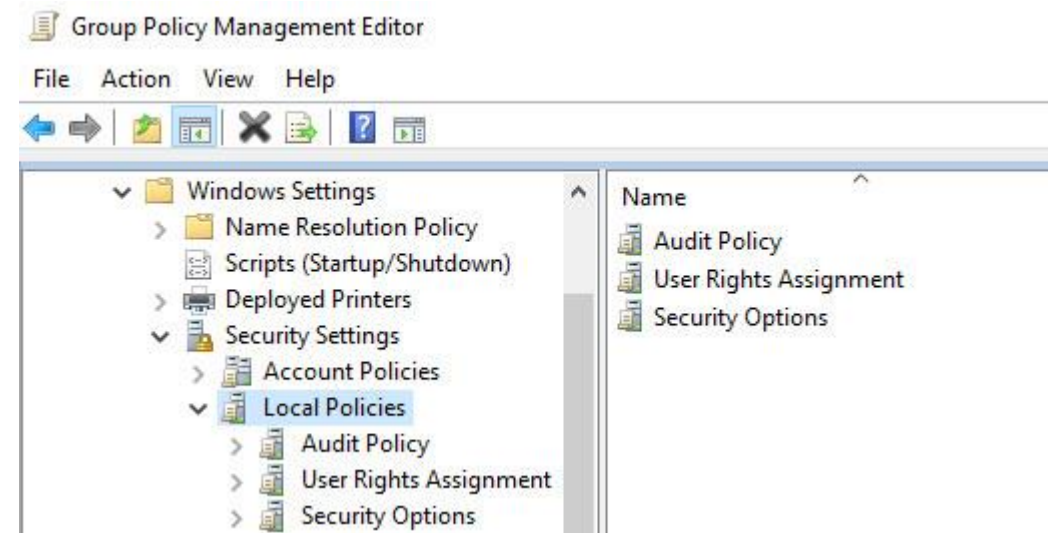
□ Політика Kerberos





Підвузол параметрів безпеки: локальні політики

- ❑ Застосовується до того, що користувачі можуть і не можуть робити на локальному комп'ютері, на якому вони входять.
- ❑ Зазвичай визначається в GPO, пов'язаних з OU, що містять облікові записи комп'ютерів.
- ❑ Три підвузли локальних політик:
 - Політика аудиту
 - Призначення прав користувача
 - Параметри безпеки





Аудит доступу до об'єктів

☐ Два кроки для аудиту об'єктів:

- Увімкніть політику доступу до об'єкта аудиту для успіху, невдачі або обох.
- Увімкніть аудит цільових об'єктів на успіх, невдачу або обидва.

☐ Аудит передбачає значні накладні витрати. Доступ до одного об'єкта може створити кілька записів журналу.

☐ Windows Server реєструє події успішного входу та деякі інші події за замовчуванням, навіть якщо аудит вимкнено.



Точна політика паролів

- ❑ Точні політики паролів дозволяють установлювати різні правила блокування паролів і облікових записів для цільових користувачів і груп.
- ❑ Створюється шляхом визначення об'єкта налаштувань пароля (PSO) у контейнері налаштувань пароля (PSC).
- ❑ Для створення PSO можна використовувати два інструменти:
 - Редагування ADSI
 - LDIFDE



Додаткові підвузли параметрів безпеки

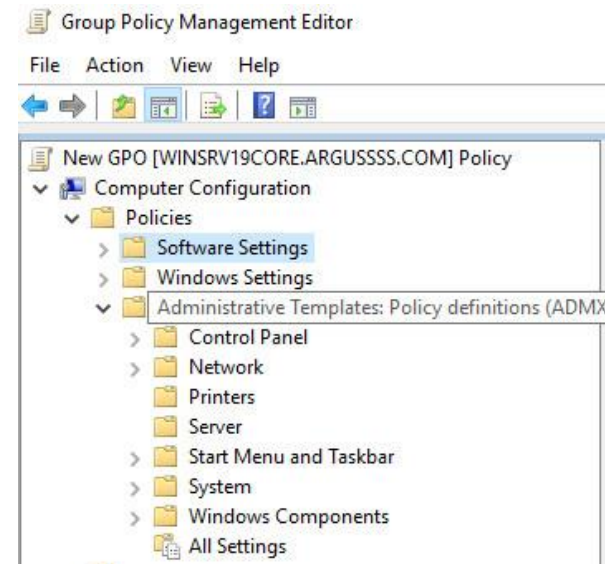
Ще 14 підвузлів у налаштуваннях безпеки:

- Журнал подій (**Event Log**)
- Реєстр системних служб обмежених груп (**Restricted Groups**)
- Системні сервіси (**System Services**)
- Реєстр (**Registry**)
- Файлова система (**File System**)
- Правила дротової мережі (IEEE 802.3) (**Wired Network (IEEE 802.3) Policies**)
- Брандмауер Windows із розширеною безпекою (**Windows Firewall with Advanced Security**)
- Менеджер політики списку мереж (**Network List Manager Policies**)
- Політики бездротової мережі (IEEE 802.11) (**Wireless Network (IEEE 802.11) Policies**)
- Політики відкритого ключа (**Public Key Policies**)
- Політики обмеження програмного забезпечення (**Software Restriction Policies**)
- Політики контролю програм (**Application Control Policies**)
- Політики безпеки IP в Active Directory (**IP Security Policies on Active Directory**)
- Розширена конфігурація політики аудиту (**Advanced Audit Policy Configuration**)



Конфігурація комп'ютера: адміністративні шаблони

- ❑ Впливає на розділ **HKEY_LOCAL_MACHINE** реєстру комп'ютера.
- ❑ Файли адміністративних шаблонів – це файли формату XML, які визначають політики в папці адміністративних шаблонів у GPO.
- ❑ Використовує формат файлу **.admx** або **.adml** для певної мови. Усі файли ADMX і ADML знаходяться в **%systemroot%\PolicyDefinitions**.
- ❑ Папка Адміністративні шаблони містить такі підвузли:
 - Панель управління
 - Мережа
 - Принтери
 - Сервер
 - Меню «Пуск» і система панелі завдань
 - Компоненти Windows
 - Усі налаштування





Політики у вузлі конфігурації користувача

- ☐ Політики, установлені у вузлі «Конфігурація користувача» (**User Configuration**), слідує за користувачем, де б він не ввійшов.
- ☐ Немає більшості параметрів безпеки та політик облікових записів.
- ☐ Політики у вузлі «Конфігурація користувача» більше зосереджені на середовищі користувача, наприклад на функціях Windows, до яких можна або не можна отримати доступ.



Конфігурація користувача: налаштування програмного забезпечення (**Software Settings**)

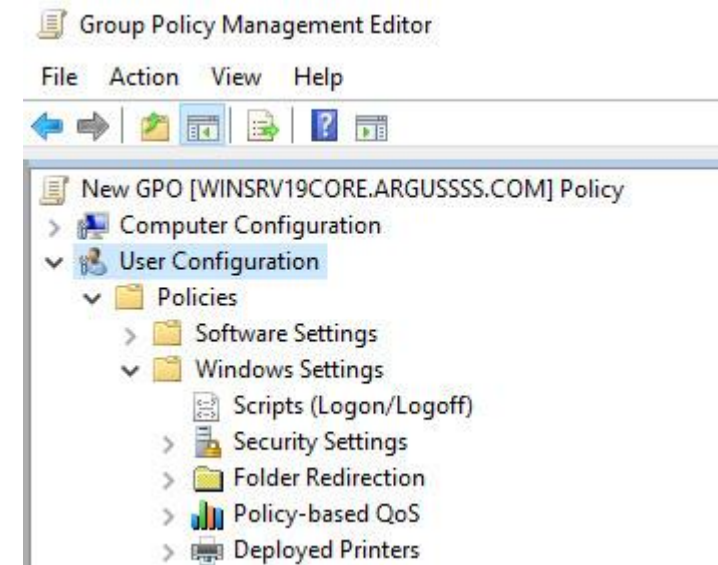
- ☐ Виконує ті самі функції, що й у конфігурації комп'ютера, але з істотними відмінностями в параметрах і виконанні.
- ☐ Пакет програмного забезпечення можна призначити лише комп'ютеру, але є два варіанти:
 - Опубліковано
 - Не встановлюється автоматично; містить посилання на програму в розділі «Програми та функції» або «Установка/видалення програм».
 - Призначений
 - Програми рекламуються як посилання в меню «Пуск».



Конфігурація користувача: Параметри Windows (**Windows Settings**)

□ Параметри Windows містить п'ять підвузлів:

- Сценарії (вхід/вихід)
- Параметри безпеки
- Перенаправлення папок
- QoS на основі політики
- Розгорнуті принтери





Підвузол налаштувань безпеки: політики обмеження програмного забезпечення

- ☐ Розроблений, щоб заборонити користувачам запускати певні програми або дозволити користувачам запускати лише певні програми.
- ☐ Папка Security Levels містить три правила:
 - Disallowed
 - Basic User
 - Unrestricted
- ☐ Папка додаткових правил призначена для винятків і містить чотири способи визначення винятків:
 - Hash
 - Certificate
 - Path
 - Network zone
- ☐ Можна налаштувати три політики:
 - Enforcement
 - Designated File Types
 - Trusted Publishers



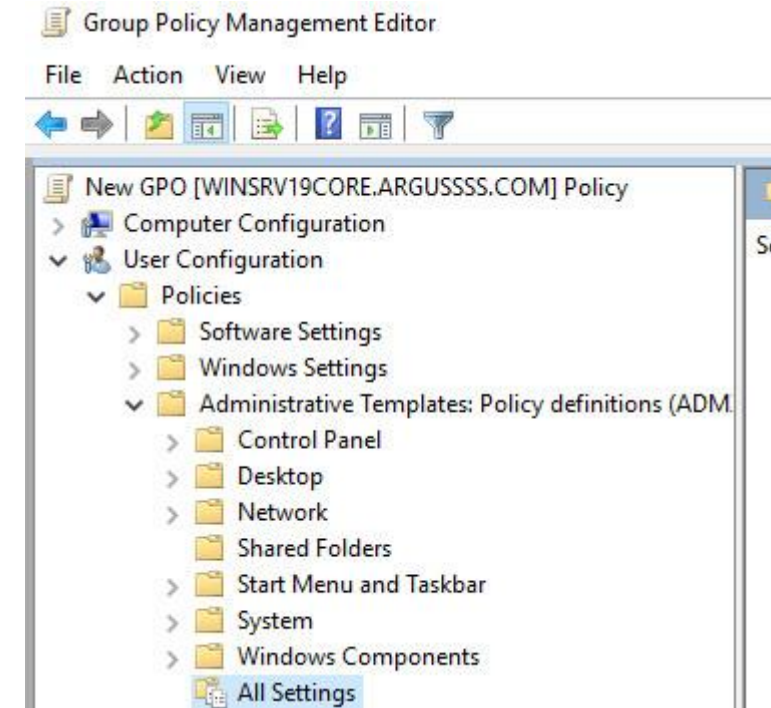
Підвузол перенаправлення папок (Folder Redirection)

- ☐ Дозволяє перенаправляти одну або декілька папок у каталозі профілю користувача.
- ☐ Корисно для забезпечення резервного копіювання документів користувача на сервер без втручання з боку користувача.
- ☐ Може допомогти зменшити використання пропускну здатності, коли використовуються роумінгові профілі.



Конфігурація користувача: адміністративні шаблони

- ❑ Впливає на розділ HKEY_CURRENT_USER реєстру комп'ютера.
- ❑ Дуже схожий на адміністративні шаблони у вузлі «Конфігурація комп'ютера».
- ❑ Містить такі додаткові підвузли:
 - Панель управління
 - Робочий стіл
 - Мережа
 - Спільні папки
 - Меню «Пуск» і панель завдань
 - Система
 - Компоненти Windows
 - Усі налаштування





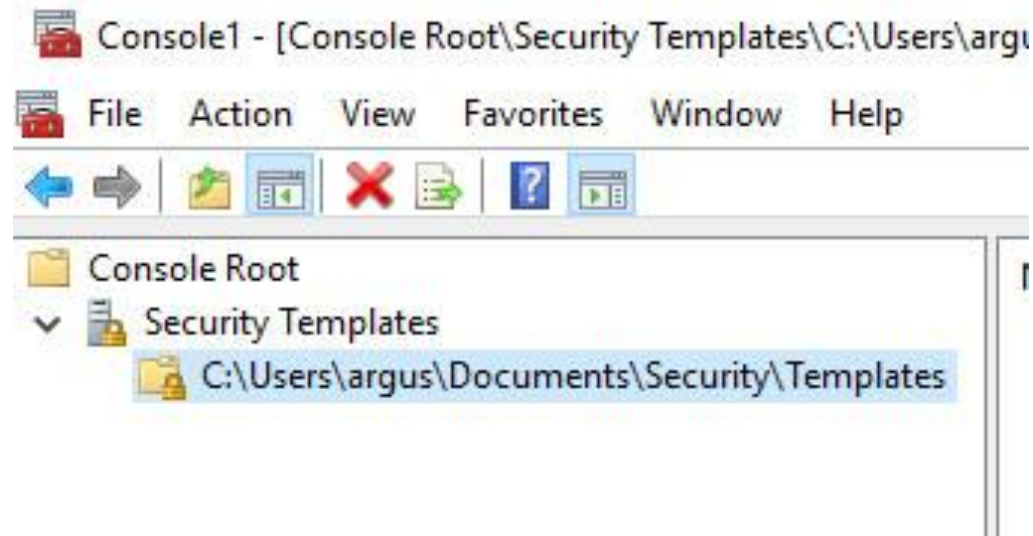
Використання шаблонів безпеки

- ❑ Шаблони безпеки — це текстові файли з розширенням **.inf**, які містять інформацію для визначення параметрів політики у вузлі «Параметри безпеки».
- ❑ Можна використовувати для перевірки поточних налаштувань безпеки на комп'ютері на відповідність налаштуванням у шаблоні.
- ❑ Три інструменти для роботи з шаблонами безпеки:
 - Оснастка шаблонів безпеки (**Security Templates snap-in**)
 - Оснастка конфігурації та аналізу безпеки (**Security Configuration and Analysis snap-in**)
 - **Secedit.exe**



Оснастка шаблонів безпеки

- ❑ Можна використовувати для створення шаблонів безпеки для використання з комп'ютерами, які вимагають інших параметрів безпеки, наприклад серверів із різними ролями.
- ❑ Коли користувач створює шаблон, він зберігається в папці «**Documents**» користувача в розділі «**Security\Templates**».





Оснастка конфігурації та аналізу безпеки

- ☐ Корисно для перевірки існуючих налаштувань безпеки комп'ютера на відомі налаштування у файлах шаблонів безпеки.
- ☐ Можна також застосувати шаблон безпеки до комп'ютера.
- ☐ Аналіз поточних параметрів безпеки за шаблоном створює звіт. Для кожного параметра політики існує п'ять можливих результатів:
 - X у червоному колі вказує на невідповідність.
 - Галочка зеленого кольору вказує на збіг.
 - Знак питання в білому колі означає, що політику не визначено або користувач не має дозволу на доступ до політики.
 - Знак оклику в білому колі означає, що політика не існує на цьому комп'ютері.
 - No індикатор вказує на те, що політика не була визначена в шаблоні.



Secedit.exe

- ☐ Програма командного рядка, яка виконує багато тих самих функцій, що й оснащення конфігурації та аналізу безпеки.
- ☐ Можна автоматизувати за допомогою сценаріїв і пакетних файлів.
- ☐ Можна імпортувати або експортувати деякі або всі налаштування між базою даних безпеки та файлом шаблону.
- ☐ Може порівнювати параметри між базою даних безпеки та поточними параметрами комп'ютера або застосовувати базу даних до комп'ютера.



Керування GPO за допомогою GPMCS₁

❑ Делегування GPO – 8 можливих дозволів можна застосувати до GPO та об'єктів-контейнерів, з якими вони пов'язані через делегування:

- Створити GPO
- Зв'язати GPO
- Виконати аналіз моделювання групової політики
- Читати дані результатів групової політики
- Читати
- Читати (з фільтрації безпеки)
- Редагувати налаштування, видалення, зміна безпеки
- Редагувати налаштування



Керування GPO за допомогою GPMC₂

❑ Після створення GPO він може перебувати в одному з таких станів:

- Link status: unlinked
- Link status: enabled
- Link status: disabled
- GPO status: Enabled
- GPO status: User Configuration Settings Disabled
- GPO status: Computer Configuration Settings Disabled
- GPO status: All Settings Disabled



Резервне копіювання та відновлення GPO

- ☐ Резервне копіювання GPO створює резервну копію налаштувань політики, але також створює резервну копію налаштувань фільтрації безпеки, налаштувань делегування та посилок фільтрів WMI.
- ☐ Не створює резервних копій файлів фільтрів WMI, політик IPSec і посилок на контейнер GPO.
- ☐ Процедура відновлення GPO залежить від того, чи необхідно:
 - Відновити попередню версію
 - Відновити видалений GPO
 - Імпортувати налаштування



Міграція GPO

- ☐ Міграція корисна, якщо кілька доменів мають однакові вимоги політики або якщо необхідно налаштувати тестове середовище.
- ☐ Об'єкти групової політики можна перенести між доменами в тому самому чи різних лісах, додавши домен до GPMC.
- ☐ GPO також можна перенести за допомогою процедури резервного копіювання та імпорту.



Результати та моделювання групової політики₁

- ☐ Майстер результатів групової політики створює звіт, щоб показати адміністраторам, які параметри політики застосовуються до користувача, комп'ютера чи обох.
- ☐ Надає ту саму інформацію, що й оснастка **Resultant Set of Policy** (RSOP).
- ☐ Після завершення роботи майстра звіт матиме три вкладки:
 - Резюме
 - Налаштування
 - Події політики



Результати та моделювання групової політики₂

- ☐ **Gpresult.exe** виконує подібне завдання, що й майстер результатів групової політики.
- ☐ Моделювання групової політики дозволяє адміністратору перевіряти результати налаштувань політики, фактично нічого не застосовуючи.
- ☐ Замість вкладки «Події політики» у ньому є вкладка «Запит», яка показує варіанти створення звіту в моделюванні групової політики.



Центральне сховище ADMX

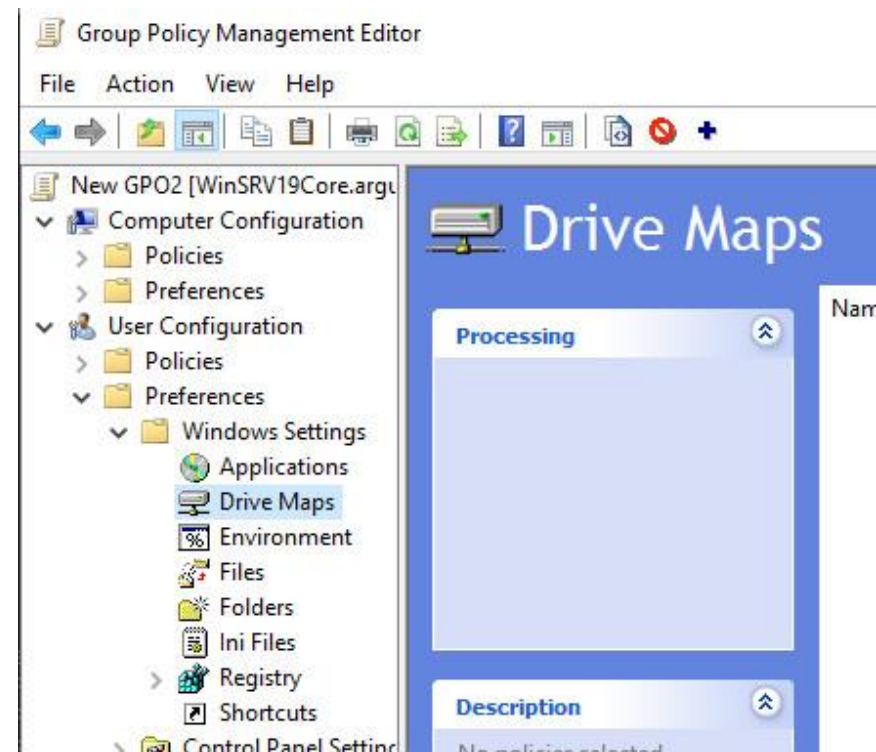
- ❑ Центральне сховище ADMX — це централізоване місце для зберігання файлів ADMX.
- ❑ Щоб створити центральне сховище, створіть папку під назвою **PolicyDefinitions** у **%systemroot%\SYSVOL\sysvol\domainname\policies**, а потім створіть папку для певної мови, яка використовує двосимвольний стандарт ISO для мов.
- ❑ Нарешті, скопіюйте файли ADMX у розташування сховища.



Параметри групової політики (**Group Policy References**)

411

- ❑ Створює стандартизоване середовище, одночасно дозволяючи користувачам вносити зміни в налаштовані параметри.
- ❑ За допомогою налаштувань групової політики можна виконувати такі завдання, як:
 - Створення та змінення локальних користувачів і груп.
 - Увімкнення та вимкнення пристроїв на комп'ютері.
 - Створення зіставлення дисків.
 - Керування параметрами живлення.
 - Створення та керування файлами, папками та ярликами.
 - Створення та змінення принтерів.
 - Налаштування параметрів застосунку.





Висновки₁

- ❑ Архітектура та функції групової політики включають такі компоненти: GPO, реплікацію, область дії та успадкування, а також створення та зв'язування GPO.
- ❑ Об'єкти групової політики домену складаються з GPT, що зберігається в спільному ресурсі Sysvol, і GPC, що зберігається в Active Directory.
- ❑ Реплікацію GPO обробляє реплікація Active Directory для GPC і FRS або DFSR для GPT.
- ❑ Використовуємо GPMS для створення, зв'язування та керування GPO, а GPME – для редагування GPO.



Висновки₂

- ❑ Starter GPO схожі на файли шаблонів.
- ❑ Об'єкти групової політики можна пов'язувати із сайтами, доменами та організаційними підрозділами. Політики застосовуються в такому порядку, і останній застосований параметр політики має перевагу, якщо існують конфлікти.
- ❑ Успадкування групової політики за замовчуванням можна змінити за допомогою блокування успадкування, примусового застосування, фільтрації групової політики та обробки політики зворотного зв'язку.
- ❑ Вузли «Конфігурація комп'ютера» та «Конфігурація користувача» містять три підвузли:
 - «Параметри програмного забезпечення»,
 - «Параметри Windows» і
 - «Адміністративні шаблони».

Донецький національний технічний університет



Тема 7. Засоби захисту ОС Windows.

Лекція 12. Аудит у Windows

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

- ☐ Що таке моніторинг змін Active Directory?
- ☐ Аудит для відстеження змін Active Directory
- ☐ Розширений аудит для відстеження змін Active Directory
- ☐ Логи безпеки у Event Viewer
- ☐ Традиційний моніторинг і аудит Active Directory
- ☐ Справжній безперервний моніторинг і аудит Active Directory



Що таке моніторинг змін Active Directory?

❑ Відстеження всіх змін, які відбуваються з об'єктами в Active Directory

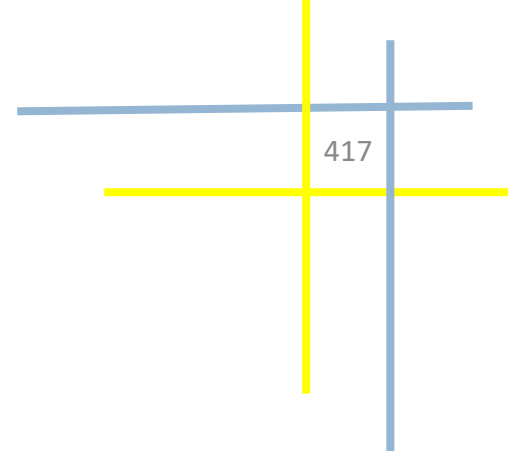
- Користувачі
- Групи
- Комп'ютери
- Групова політика
- Політика паролів
- тощо



Що таке моніторинг змін Active Directory?

❑ Відстеження всіх деталей щодо змін об'єктів в Active Directory

- Хто вніс зміни
- Який об'єкт було змінено
- Коли внесено зміни
- Яке нове налаштування
- Якою була стара настройка



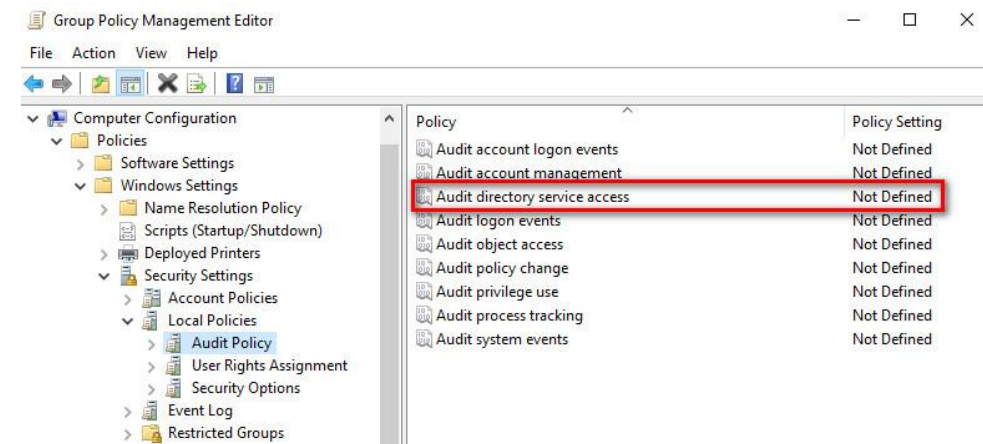


Аудит для відстеження змін Active Directory₁

418

❑ На кожному контролері домену має бути ввімкнено аудит:

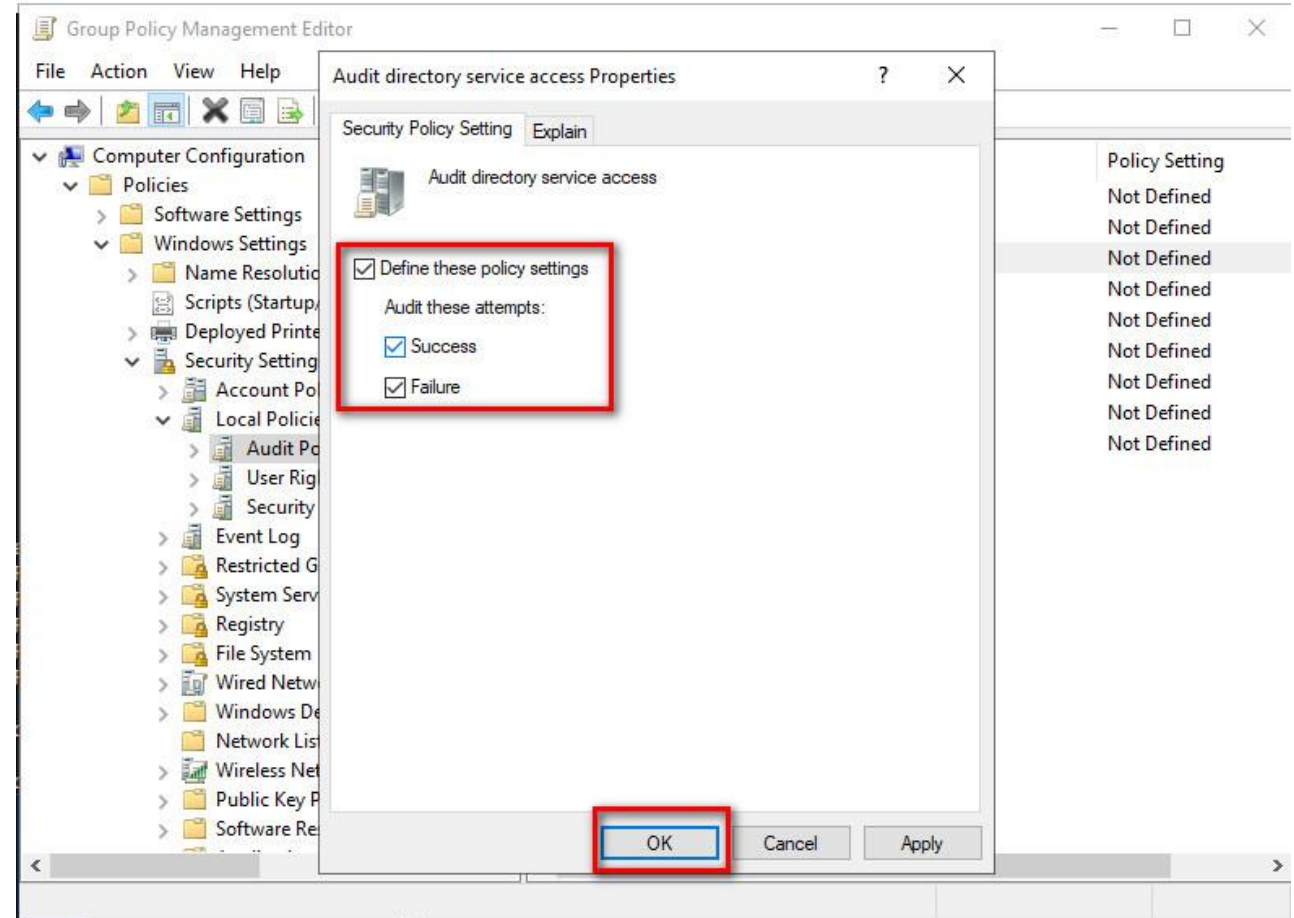
- Увімкнути аудит AD через групову політику
- Налаштувати політику контролерів домену за замовчуванням АБО створити новий GPO та зв'язати його з OU контролерів домену
- Аудит знаходиться в: **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Audit Policy**





Аудит для відстеження змін Active Directory₂

- ☐ Успіх (**Success**) – відстежує успішні зміни в AD
- ☐ Помилка (**Failure**) – відстежує відмови для зміни AD





Аудит для відстеження змін Active Directory₃

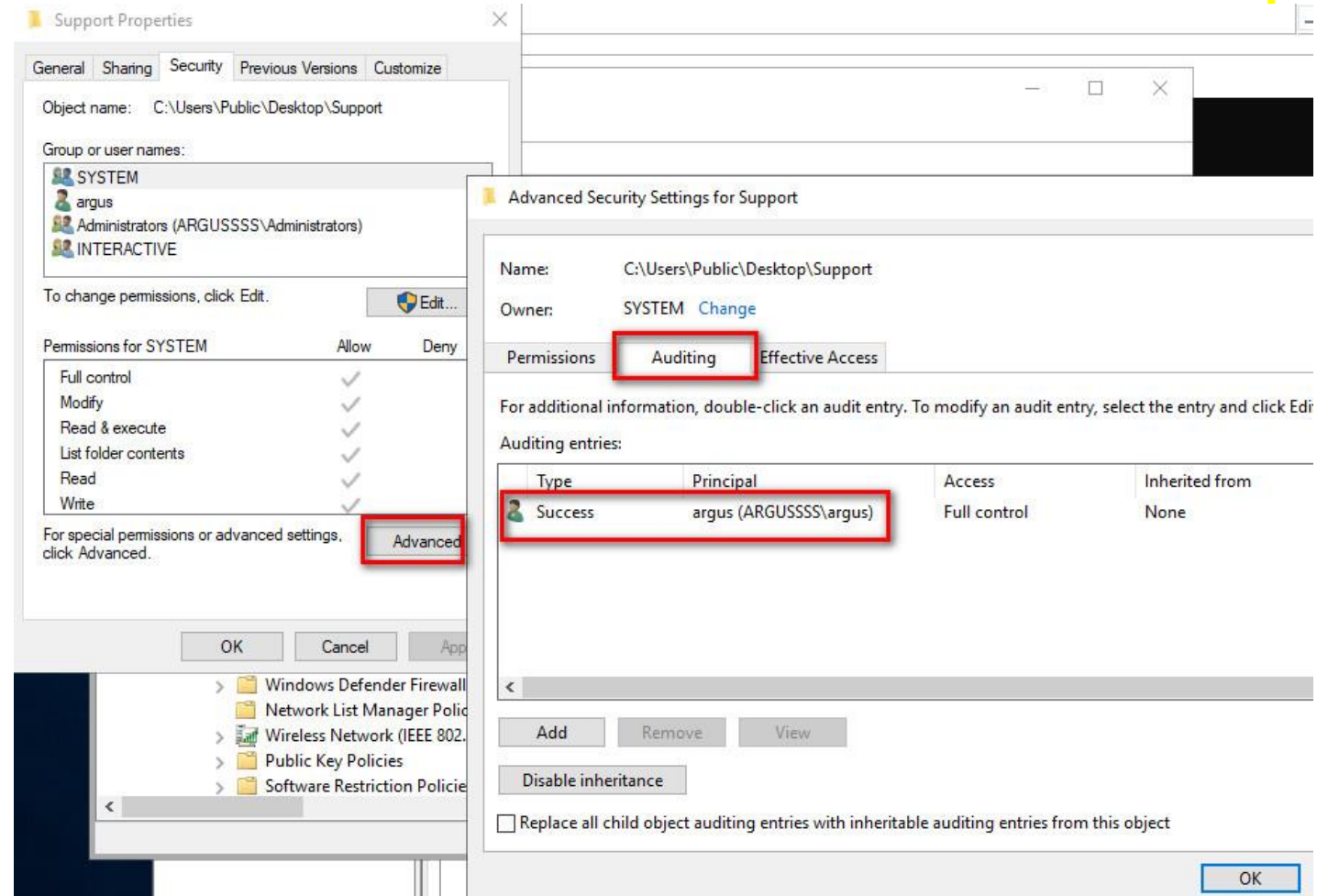
- ☐ Перевірені події зберігаються у засобі Event Viewer.
 - Відстежені зміни зберігаються в журналі безпеки на DC, де сталася подія.
 - Кожен DC має унікальний журнал безпеки.
 - Щоб переглянути всі події, необхідно переглянути кожен DC або консолідувати логи.
- ☐ Деякі події, згенеровані аудитом доступу до служби каталогів.
- ☐ Деякі події, згенеровані Auditing Account Management.



Аудит для відстеження змін Active Directory₄

□ Аудит будь-якого об'єкту:

- Увімкніть доступ до служби каталогів аудиту
- Налаштуйте вкладку «Аудит», клацнувши вкладку «Безпека» у властивостях об'єкта
- Потрібно вибрати «кожну властивість», яку потрібно відстежувати!





Аудит для відстеження змін Active Directory₅

❑ Не забудьте налаштувати всі всі області:

- Користувачі
- Групи
- Комп'ютери
- Схема
- Групова політика
 - GPC
 - GPT

422



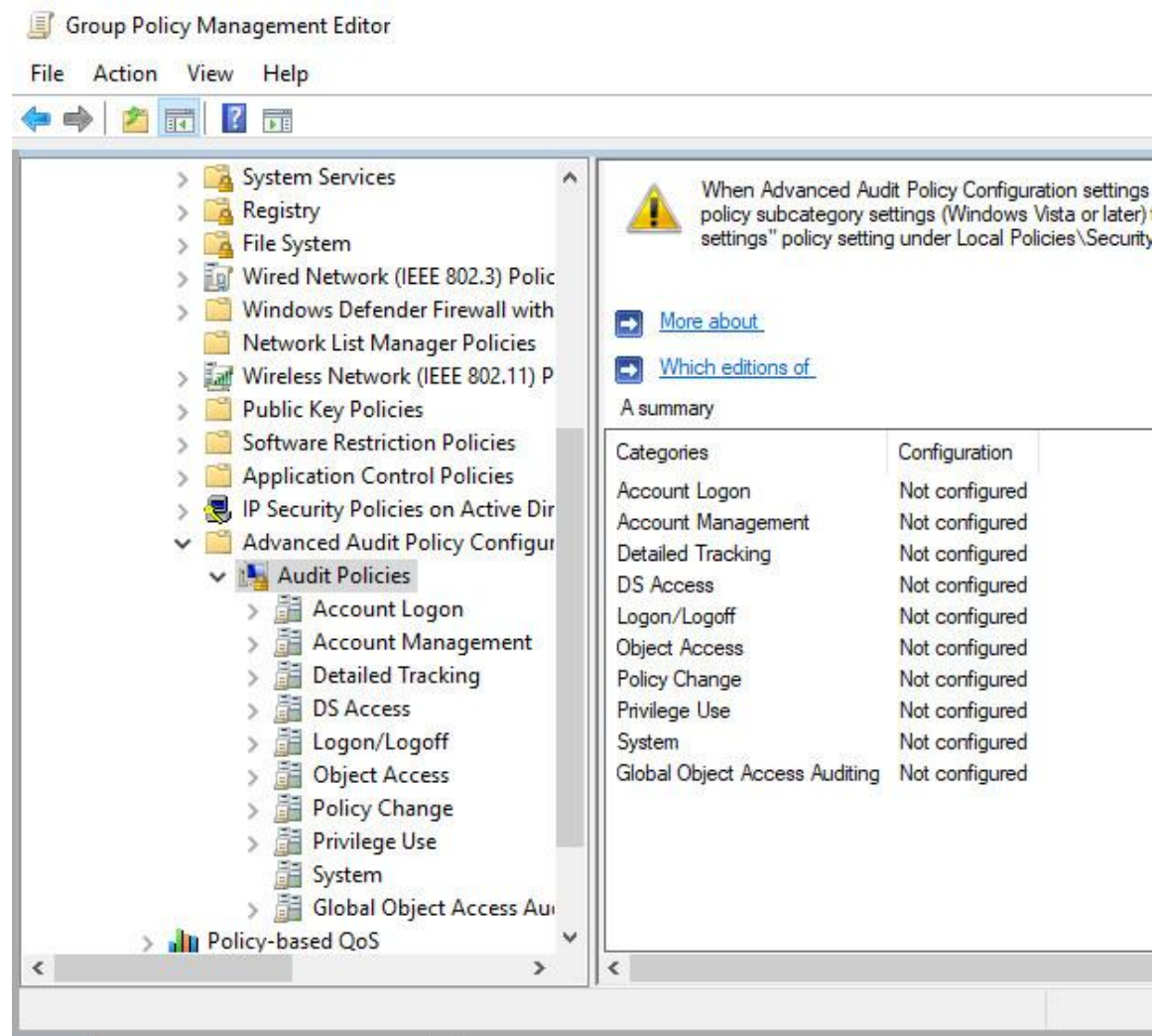
Розширений аудит для відстеження змін AD₁

- ☐ Розширений аудит для усунення несправностей, аудиторів і спеціалістів із безпеки
- ☐ Надає деталі для більшості зобов'язань щодо відповідності:
 - cse.mitre.org
 - Збігається з Security Compliance Manager (SCM)
- ☐ Продовжує надсилати інформацію про перевірені події в журнал безпеки.



Розширений аудит для відстеження змін AD₂

424





Розширений аудит для відстеження змін AD₃

425

DS Access–Directory Service Changes	Повідомляє про зміни в об'єктах у доменних службах Active Directory (AD DS). Типи змін, про які повідомляється, це операції створення, зміни, переміщення та відновлення видалення, які виконуються над об'єктом. Аудит DS Change, якщо це доречно, вказує на старі та нові значення змінених властивостей об'єктів, які були змінені.
DS Access–Directory Service Replication	Повідомляє про початок та кінець реплікації між двома контролерами домену.
DS Access–Detailed Directory Service Replication	Повідомляє детальну інформацію про реплікацію інформації між контролерами домену. Ці події можуть бути дуже великими.
DS Access–Directory Service Access	Звітує про доступ до об'єкта AD DS. Лише об'єкти з SACL викликають генерацію подій аудиту, і лише тоді, коли доступ до них здійснюється у спосіб, який відповідає їх SACL. Ці події подібні до подій доступу до служби каталогів у попередніх версіях



Розширений аудит для відстеження змін AD₄

☐ Локальний GPO

- Configuration\Windows Settings\Security Settings\Advanced Audit Policy Configuration\System Audit Policy

☐ Об'єкт групової політики AD GPO в GPMC

- Computer Configuration\Policies\Windows Settings\Security Settings\Advanced Audit Policy Configuration\System Audit Policy



Розширений аудит для відстеження змін AD₅

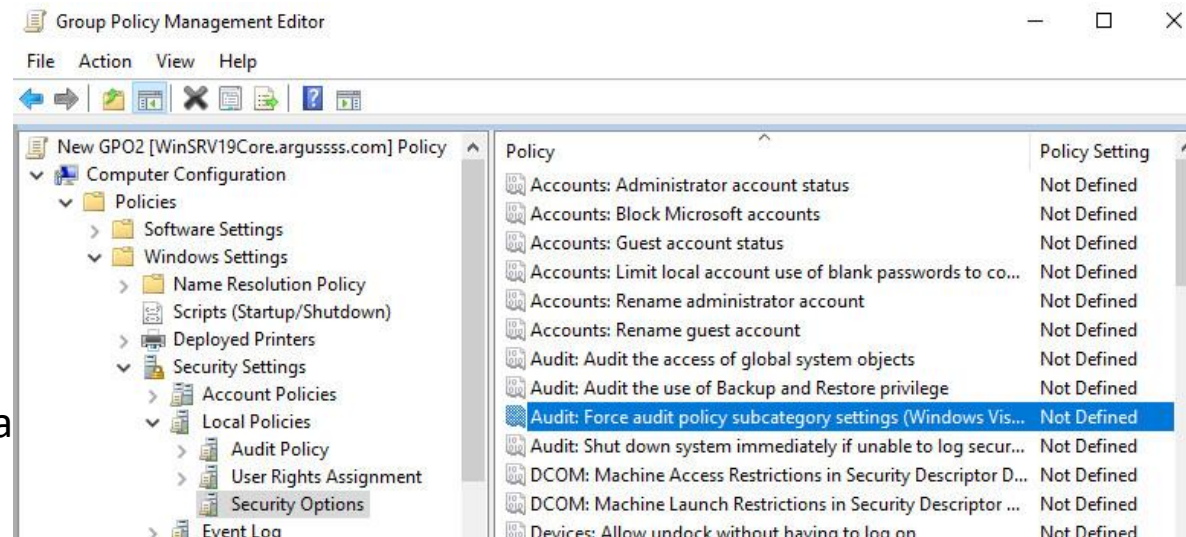
427

❑ Проблеми сумісності

- Застаріла політика аудиту
- Новий розширений аудит

❑ Якщо існує застаріла політика аудиту... вона переможе над новою вдосконаленою політикою аудиту... якщо тільки не встановити:

- **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Security Options**
- **«Audit: Force audit policy subcategory settings...»** для перевизначення налаштування категорії політики аудиту.
 - Увімкнено: ігноруватиме попередні налаштування
 - Вимкнено: використовуватиме поведінку за замовчуванням





Логи безпеки в Event Viewer₁

❑ Налаштуйте права користувача на керування аудитом і журналом безпеки:

- Аудит файлів, папок, реєстру тощо.
- Перегляд перевірених подій у журналі безпеки.
- Перегляд та очищення журналу безпеки.
- Збереження журналу безпеки.



Логи безпеки в Event Viewer₂

❑ Ідентифікатори подій

- Старіші версії – 3-значні ідентифікатори
- Новіші версії – 4-значні ідентифікатори

❑ <https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/>



Логи безпеки в Event Viewer₃

❑ Створіть спеціальне представлення «багатьох журналів» або «багатьох джерел» в «один журнал»

- Контролери домену
- Адміністративні заходи
- Ролі сервера
- Доменні служби Active Directory
- Сервер DHCP
- DNS-сервер
- Файловий сервер
- Політика мережі та служби доступу
- Веб-сервер



Логи безпеки в Event Viewer₄

431

☐ Користувацькі параметри перегляду

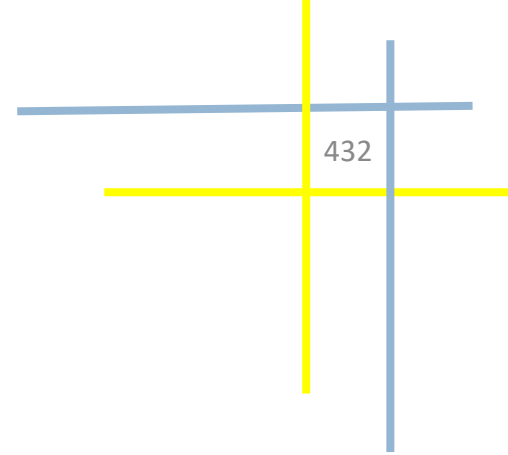
- Фільтрувати за журналом
- Зареєстровано (діапазони дати/часу)
- Рівень події (тип журналу)
- Переглянути параметри
 - За журналом(ами)
 - За джерелом(ами)
- Категорія завдання
- Ключові слова



Логи безпеки в Event Viewer₅

❑ Після створення спеціального представлення:

- До представлення можна додати фільтр
- Можна прикріпити завдання до представлення
- Представлення можна експортувати
- Представлення можна скопіювати





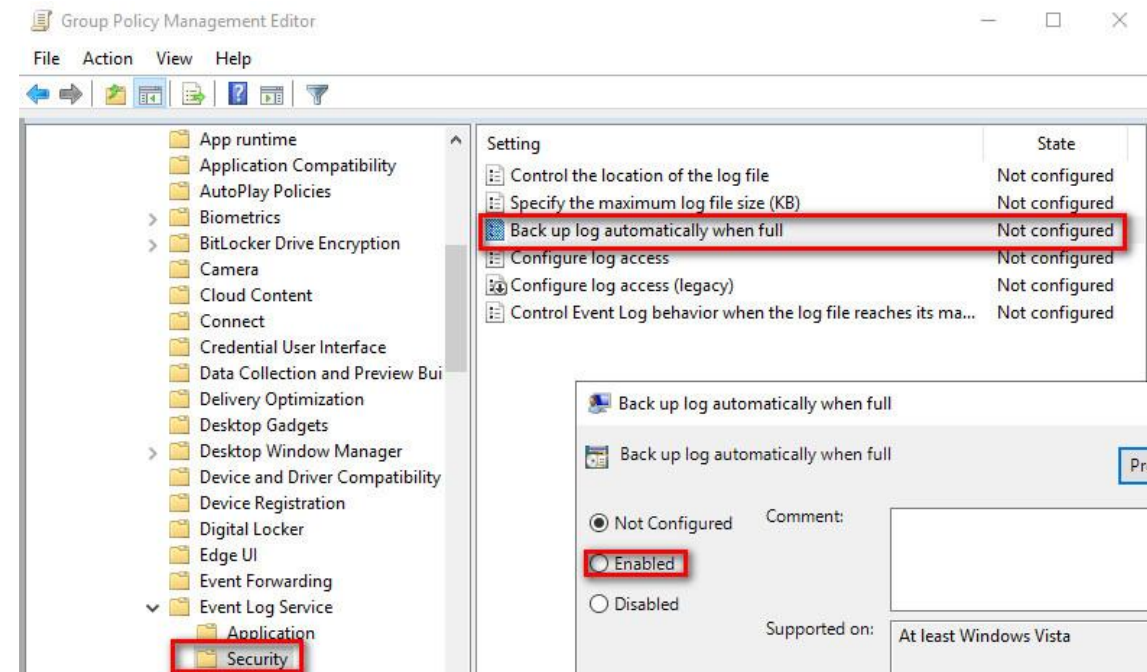
Логи безпеки в Event Viewer₆

433

❑ Резервне копіювання журналу безпеки

■ Computer Configuration\Policies\Administrative Templates\Windows Components\Event Log Service\Security

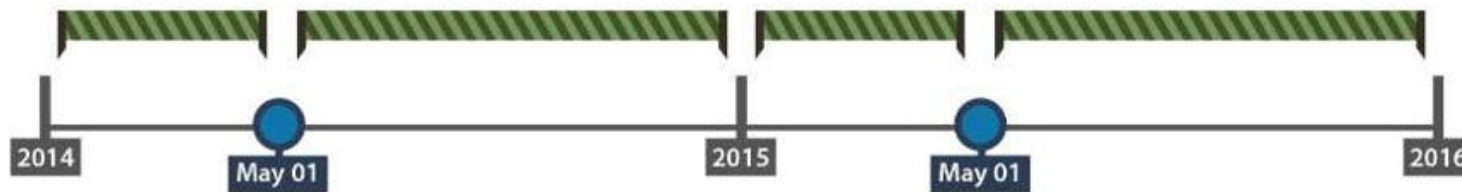
- Автоматичне резервне копіювання журналів (**Back up log automatically when full**)
- Контроль розташування лог файлу (**Control the location of the log file**)



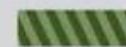


Стандартний аудит/моніторинг AD

- ☐ Зазвичай проводиться один раз на рік
- ☐ Інформація збирається на серверах/DC
- ☐ Інформація надається за один момент часу
- ☐ Зміни можна вносити безпосередньо до та в будь-який час після збору інформації



Legend



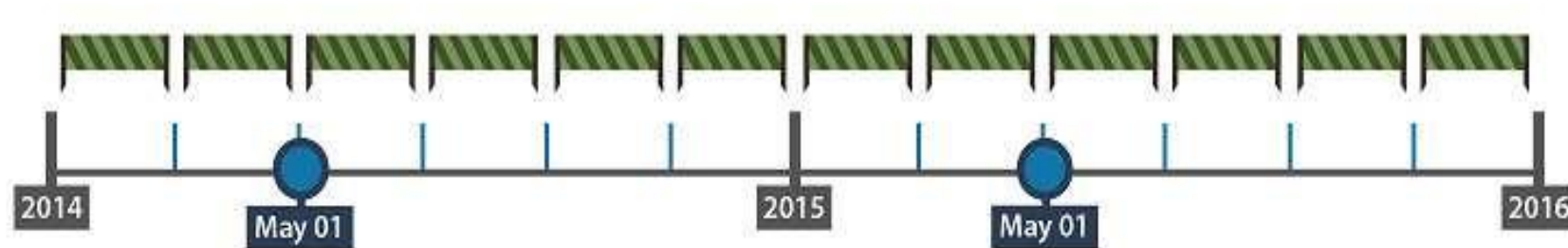
Unknown Risk



Point-in-time Annual Audit



Безперервний аудит/моніторинг AD

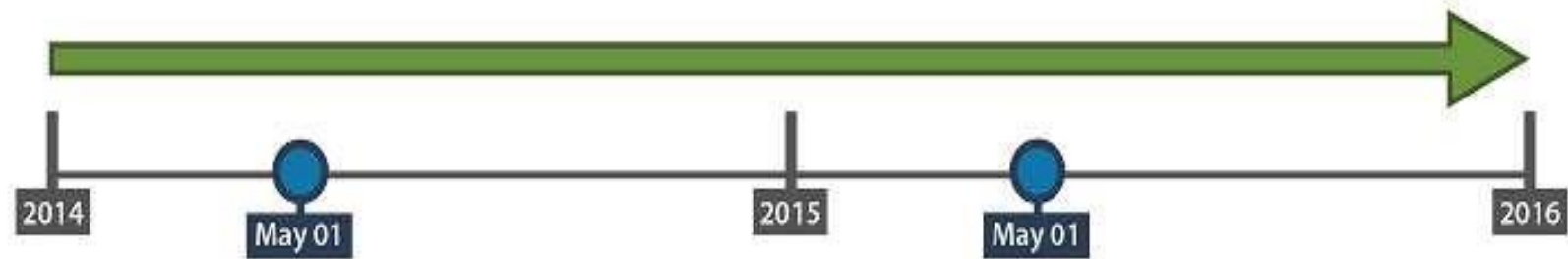


Legend

-  Unknown Risk
-  Point-in-time Annual Audit
-  Manual or Automated Continuous Audits



Справжній постійний моніторинг АД



Legend



True continuous auditing
and reporting



Point-in-time Annual Audit



Проблеми з Event Viewer

- ☐ Замалий розмір журналів безпеки
- ☐ Інтерфейс не передбачає звітності
- ☐ Події важко розшифрувати та проаналізувати
- ☐ Події реєструються на DC, де відбувається подія... кілька журналів
- ☐ Оповідження недостатньо детальне

Слід подумати про використання для аудиту спеціалізованого програмного забезпечення!!!

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux.

Лекція 13. Модель загроз Linux

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

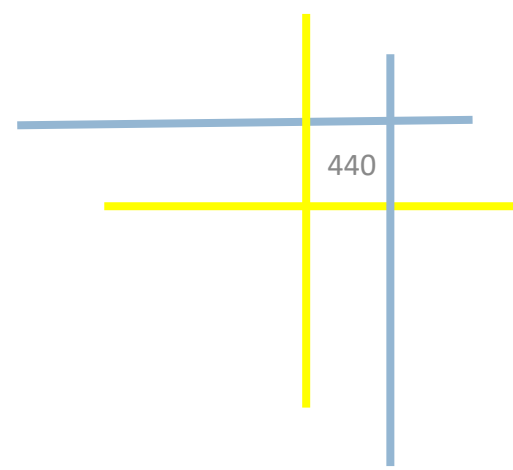
Укладач: проф. Дорогий Я.Ю.



Лекційні питання

439

- ☐ Загальний опис ST
- ☐ Модель загроз
- ☐ Модель політики безпеки
- ☐ Ідентифікація КЗЗ
- ☐ Інтерфейси КЗЗ



Загальний опис системи



Вступ

- ❑ Як і для Windows, розглянемо функції безпеки Linux за документами Security Target (ST), що готували для оцінювання окремих дистрибутивів Linux за стандартом Common Criteria
- ❑ Оцінювання (з наступною сертифікацією) проводили для різних дистрибутивів, серед яких
 - Red Hat Enterprise Linux (v.8.2)
 - SUSE Linux Enterprise Server (v.15)
 - Oracle Linux (v.7.6)
 - Canonical Ubuntu Server (18.04.4)
- ❑ Багато підрозділів опису функцій безпеки зазначених продуктів в різних ST повторюються дослівно, що свідчить про те, що за основу береться один документ, розроблений для Linux взагалі, а не для окремих дистрибутивів
- ❑ Однак є суттєві відмінності, пов'язані як з відмінностями функцій дистрибутивів, так і з комплектацією відповідних ТОВ
- ❑ Ця презентація зроблена на основі ST на відповідність профілю OSPP (EAL 4+) — Red Hat Enterprise Linux v.7 та SUSE Enterprise Server v.12
 - ST на відповідність GPOS PP (EAL 1) не містять детального опису політики безпеки



Загальний опис системи₁

442

- ☐ Об'єктом оцінювання є дистрибутив Linux версії, що вказана в ST, що виконується на обладнанні вказаному в ST (далі будемо ОО називати просто "Linux")
- ☐ Кілька систем Linux можуть бути пов'язані фізично захищеною локальною мережею (LAN)
- ☐ Програми користувача ініціюють мережні запити відсилаючи іншому комп'ютеру повідомлення протоколів TCP (Transmission Control Protocol) або UDP (User Datagram Protocol)



Загальний опис системи₂

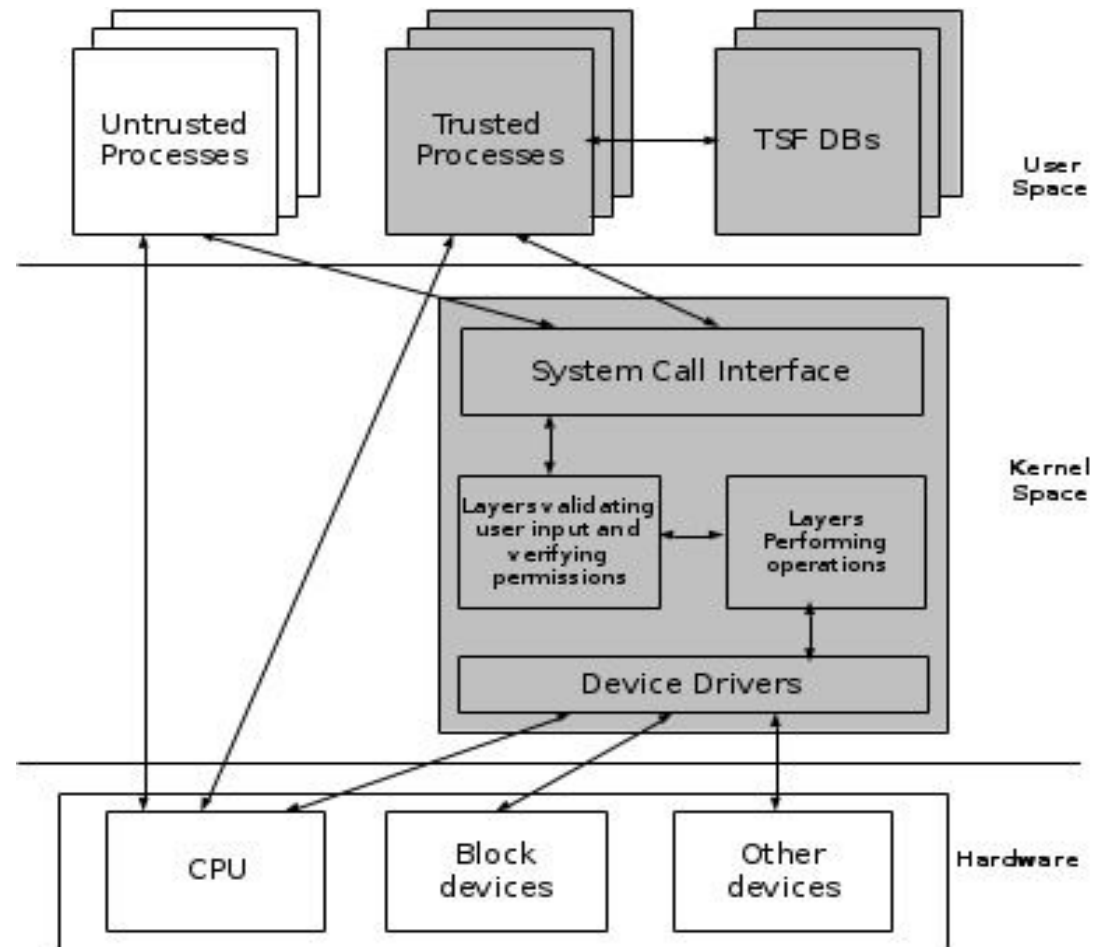
443

- ❑ Linux забезпечує механізм ідентифікації й автентифікації користувачів, вимагаючи від кожного користувача зайти в систему з правильним паролем на локальному комп'ютері і на кожному віддаленому комп'ютері, де користувач може вводити команди в командну оболонку (наприклад, віддалені сесії SSH)
- ❑ Кожен комп'ютер забезпечує набір таких політик:
 - Політика дискреційного керування доступом (DAC), заснована на бітах доступу у стилі UNIX
 - Опціонально — списки керування доступом (ACL)
 - Політика мандатного керування доступом (MAC) із використанням розширень SELinux (Security Enhanced Linux) для іменованих об'єктів, що перебувають під його контролем.
 - У системах гілки Debian (зокрема, Ubuntu) замість SELinux застосовують AppArmor з подібною функціональністю



Структура хост-комп'ютера

444





Сервіси, що надає кожен хост-комп'ютер

- ☐ Локальні сервіси користувачам, які в даний час увійшли в систему за допомогою
 - локальної консолі комп'ютера
 - віртуальних консолей
 - термінальних пристроїв, що підключені через фізично захищені послідовні лінії
- ☐ Локальні сервіси попереднім користувачам через відкладені завдання
 - прикладом є демон cron
- ☐ Локальні сервіси користувачам, які отримали доступ до локального хосту через мережу за допомогою протоколу, такого як SSH, який запускає оболонку користувача на локальному хості
- ☐ Мережні сервіси для потенційно кількох користувачів або на локальному хості, або на віддалених
- ☐ Надаються середовища віртуалізації, що дозволяють ненадійному програмному забезпеченню виконуватись у користувацькому режимі процесора



Локальні та мережні сервіси, що надає Linux₁

446

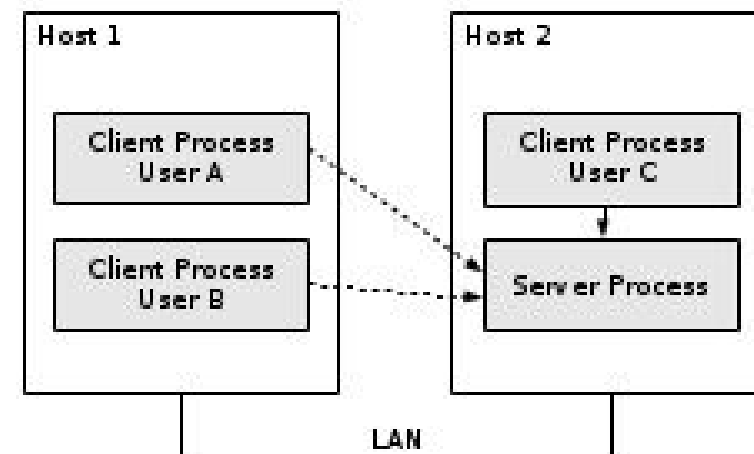
- ☐ Користувач може увійти в систему на локальному хост-комп'ютері та робити запити файлової системи або запити управління пам'яттю для служб за допомогою системних викликів до ядра локального хосту
- ☐ Мережні служби, такі як SSH або FTP, включають архітектуру клієнт-сервер і протокол рівня мережного сервісу

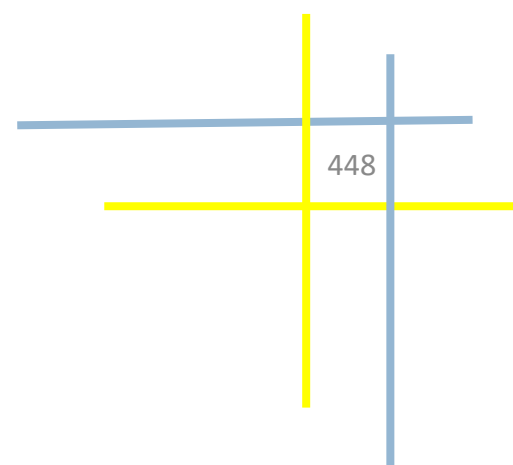


Локальні та мережні сервіси, що надає Linux₂

447

- ❑ Користувач А може увійти на хост 1, а потім використати SSH для входу на хост 2
- ❑ Також зверніть увагу, що мережний клієнт і сервер можуть бути в одній хост-системі





Політика безпеки



Модель загроз₁

449

- ☐ T.ACCESS.TSFDATA
- ☐ T.ACCESS.USERDATA
- ☐ T.ACCESS.TSFFUNC
- ☐ T.ACCESS.COMM
- ☐ T.RESTRICT.NETTRAFFIC
- ☐ T.IA.MASQUERADE
- ☐ T.IA.USER
- ☐ T.ACCESS.COMPENV
- ☐ T.INFOFLOW.COMP
- ☐ T.COMM.COMP



Модель загроз₂

- ☐ T.ROLE.SNOOP
- ☐ T.ROLE.DELEGATE
- ☐ T.UNOBSERVED_AUDIT
- ☐ T.ACCESS.CP.USERDATA

450



Модель політики безпеки₁

❑ Політика безпеки для Linux визначається функціональними вимогами безпеки

❑ Далі наведено перелік суб'єктів та об'єктів, які беруть участь у політиці

❑ Суб'єкти:

- Процеси, що діють від імені людини-користувача або технічної сутності
- Процеси, що діють від імені людини-користувача або технічної сутності, що забезпечує середовище віртуальної машини

❑ Іменовані об'єкти:

- Об'єкти файлової системи в дозволених (підтримуваних) файлових системах
 - Перелік підтримуваних файлових систем див. у Довідкових матеріалах, слайди 10-11
- Об'єкти міжпроцесової взаємодії (IPC):
 - Семафори, Спільна пам'ять, Черги повідомлень, Іменовані канали, Спеціальні файли сокетів домену UNIX



Модель політики безпеки₂

□ Дані КЗЗ:

- Виконуваний код КЗЗ
- Метадані суб'єктів — усі дані, що використовуються для суб'єктів, крім даних, які не інтерпретуються КЗЗ і не реалізують частини КЗЗ (ці дані називаються даними користувача)
- Метадані іменованих об'єктів — усі дані, що використовуються для відповідних об'єктів, крім даних, які не інтерпретуються КЗЗ і не реалізують частини КЗЗ (ці дані називаються даними користувача)
- Облікові записи користувачів, включаючи атрибути безпеки Записи аудиту

□ Дані користувача:

- Виконуваний код, що не належить КЗЗ, і використовується для керування поведінкою суб'єктів
- Дані, які не інтерпретуються КЗЗ і зберігаються або передаються з використанням іменованих об'єктів



Політика безпеки. Користувачі

- ☐ Користувач — це уповноважена особа, що має обліковий запис
- ☐ Користувачі можуть використовувати систему в декілька способів
- ☐ Користувач повинен увійти в локальну систему, щоб отримати доступ до захищених ресурсів системи
- ☐ Після автентифікації користувача він може отримати доступ до файлів або виконувати програми на локальному комп'ютері, або робити мережні запити на інші комп'ютери в системі



Політика безпеки. Суб'єкти та об'єкти

! ☐ Єдиними суб'єктами в системі є процеси

- Процес складається з адресного простору з контекстом виконання
- Процес обмежений комп'ютером; відсутній механізм диспетчеризації процесу для віддаленого запуску (через TCP/IP) на іншому хості
- Кожен процес має ідентифікатор процесу (PID), який є унікальним на своєму локальному комп'ютері-хості, але PID не є унікальними для всіх систем
 - Як приклад, кожен хост у системі має процес init з PID 1

☐ Об'єкти — це пасивні сховища даних

- У Linux визначено три типи об'єктів:
 - іменовані об'єкти, що є ресурсами, такі як файли та об'єкти IPC, якими можуть керувати багато користувачів, використовуючи угоду про іменування, визначену в інтерфейсі K33;
 - об'єкти зберігання, що підтримують доступ як для читання, так і для запису різними недовіренними суб'єктами; і
 - публічні об'єкти — об'єкти, які можуть публічно читати недовірені суб'єкти, але записувати їх можуть лише довірені суб'єкти.
- Відповідно до цих визначень, усі іменовані об'єкти також класифікуються як об'єкти зберігання, але не всі об'єкти зберігання є іменованими об'єктами



Політика безпеки Керування доступом

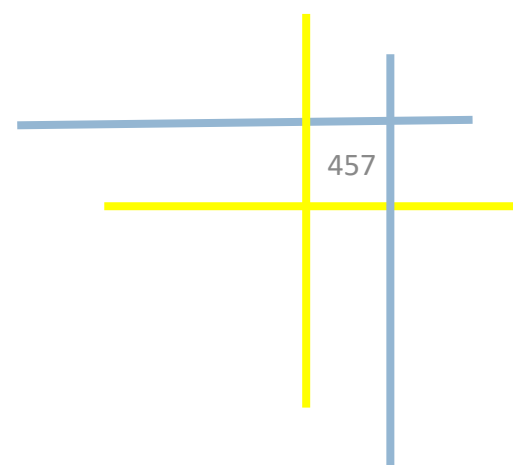
455

- ☐ Linux застосовує політику дискреційного керування доступом для всіх іменованих об'єктів, що знаходяться під її контролем, і політику повторного використання об'єкта для всіх об'єктів зберігання, що знаходяться під її контролем
- ☐ На додаток до політики дискреційного керування, Linux також застосовує політику мандатного керування доступом для всіх іменованих об'єктів, що знаходяться під її контролем
- ☐ Для забезпечення дотримання правил керування доступом усі користувачі повинні бути ідентифіковані та їх ідентифікаційні дані повинні бути автентифіковані



Політика безпеки

- ☐ Linux використовує як апаратні, так і програмні механізми захисту
- ☐ Користувач може ввійти в систему на консолі, на інших безпосередньо підключених терміналах або через мережне підключення
- ☐ Автентифікація базується на введеному користувачем паролі та даних автентифікації, що зберігаються у захищеному файлі, або через інші типи облікових даних, наприклад, криптографічні ключі при використанні SSH
- ☐ Архітектура системи забезпечує механізми самозахисту K33 та ізоляції процесів



Інтерфейси КЗЗ



Ідентифікація K33 (з яких причин пакети або окремі файли відносять до K33)

458

- ☐ Він містить код, такий як ядро, модуль ядра та драйвери пристроїв, що працює у привілейованому режимі процесора
- ☐ Він забезпечує політику безпеки системи
- ☐ Він дозволяє SUID або SGID до привілейованого користувача (наприклад, root) або групи
- ☐ Йому надані можливості (capabilities) файлової системи, що передбачають підвищення привілеїв під час виконання
- ☐ Він надає одну або кілька можливостей подолати обмеження, пов'язані з безпекою, які застосовуються системою для користувача, що його викликає
- ☐ Він був запущений як демон, що виконується з правами root або з системним UID/GID
- ☐ Це програмне забезпечення, яке повинно функціонувати належним чином для підтримки механізмів безпеки системи
- ☐ Він потрібен для системного адміністрування
- ☐ Він складається з даних K33 або файлів конфігурації
- ☐ Він складається з бібліотек, пов'язаних з програмами K33
- ☐ Він був запущений як програма з іншими привілеями, ніж той, що його викликав
 - Найпомітнішими прикладами є init, modprobe та v86d, використовувані драйвером iuvesafb
- ☐ Він надає користувачеві, який його викликає, одну або кілька можливостей подолання багаторівневої безпеки (MLS)



Програмні компоненти K33 Linux

- ☐ Ядро Linux, що включає
 - Базове ядро
 - Окремо-завантажувані модулі ядра і драйвери пристроїв
- ☐ Ядро реалізує інтерфейс системних викликів Linux
- ☐ Програмне забезпечення K33, що не входить до ядра, включає програми, що працюють з адміністративними привілеями, такі як демони `sshd` та `systemd`
- ☐ K33 також включає конфігураційні файли, які визначають
 - авторизованих користувачів,
 - групи користувачів,
 - послуги, що надаються системою,
 - та інші дані конфігурації
- ☐ Не входять до складу K33
 - оболонки, що використовуються адміністраторами,
 - та стандартні утиліти, що викликаються адміністраторами



Інтерфейси K33₁

- ❑ Інтерфейси K33 включають будь-який інтерфейс, який можливий між недовіреним програмним забезпеченням та K33
- ❑ Інтерфейси K33 включають:
 - локальні інтерфейси, що надаються кожним хост-комп'ютером
 - мережні інтерфейси клієнт-сервер, що забезпечуються парами хост-комп'ютерів
- ❑ Локальні інтерфейси, що надаються хост-комп'ютером, включають такі надані ядром інтерфейси:
 - Системні виклики довірених та недовірених програм до привілейованого програмного забезпечення режиму ядра
 - Наступні інтерфейси ядра слід розглядати як незалежні інтерфейси під час аналізу безпеки (незважаючи на те, що вони є семантичним розширенням системних викликів `open`, `ioctl` або пов'язаних з сокетом):
 - Файли пристроїв
 - Віртуальні файлові системи
 - Мережні сокети, що використовують протокол `AF_NETLINK` або `PF_KEY` або `AF_ALG`
 - Будь-який аналізатор/парсер мережних протоколів, реалізований ядром:
 - Ethernet
 - ARP
 - Сімейства протоколів TCP/IP
 - IPSec
 - Помічена комунікація з використанням протоколів IPSec



Інтерфейси K3Z₂

□ Наступні інтерфейси реалізовані довіреними процесами в просторі користувача:

- Мережні інтерфейси, що надаються парами хост-комп'ютерів:
 - SSHv2 (RFC 4252)
 - IKEv1 та IKEv2: (IKE — RFC 2409 та RFC 5996, доступні групи Діффі-Хеллмана - RFC 2409, RFC 3526, RFC 5114)
 - TLS v1.1 та TLS v1.2: (RFC 4346 та RFC 5246)
- Масиви символів `argv` та `envp`, які є аргументами системного виклику `execve`
 - Ці два масиви можуть бути оцінені виконуваною прикладною програмою
 - Ряд змінних середовища, які можуть бути надані `envp`, реалізуються бібліотеками, які зазвичай завантажуються більшістю, якщо не всіма програмами (наприклад, C-бібліотекою)
- Файли, які є частиною бази даних K3Z, що визначають параметри конфігурації, що використовуються функціями захисту
- Інтерфейси взаємодії між процесами, експортовані прикладною програмою
 - Тип міжпроцесового інтерфейсу зв'язку включає підтримку DBus
 - DBus — це протокол, який працює на основі механізмів міжпроцесового зв'язку, що надаються ядром
- Виклики гіпервізора та емуляція та моделювання інструкцій, реалізовані ядром Linux для підтримки середовища віртуалізації KVM
 - Це включає моделювання та емуляцію апаратного забезпечення, що постачається з ядром Linux



Інтерфейси, що не входять до K33₁

- ❑ Інтерфейси між процесами, що не належать до K33, та базовим обладнанням
- ❑ Інтерфейси між різними частинами K33, які невидимі для звичайних користувачів (наприклад, між підпрограмами всередині ядра)
 - Інтерфейс є внутрішнім для довіреної частини Linux і не може бути викликаний поза цими частинами
- ❑ Прошивка, хоча і є частиною Linux, не розглядається як така, що забезпечує інтерфейси K33
 - тому що вони не допускають прямих непривілейованих операцій до них



Інтерфейси, що не входять до КЗЗ₂

- ☐ Винятки процесорів, що передають прошивці, не вважаються інтерфейсами TSF
 - Вони не стосуються безпеки, оскільки надають доступ до прошивки, яка не реалізує жодних функцій безпеки
- ☐ У випадку, якщо ІТ-середовище забезпечує середовище віртуалізації, таке як
 - Хост-системи KVM,
 - z/VM або PR/SM на системах s390x,
 - POWER LPAR на системах IBM POWER,
- ☐ архітектурні інтерфейси гіпервізора середовища віртуалізації (як-то виклики гіпервізора) не вважаються інтерфейсом TSF
- ☐ Зверніть увагу, що про функціонал гіпервізора, реалізований ядром Linux та його інтерфейсами, тут мова не йде
- ☐ Стан SMM процесорів на базі Intel не стосується безпеки, оскільки програмне забезпечення, що виконується в цьому стані, не реалізує жодних функцій безпеки



Що не дозволено (політикою безпеки)

- ❑ Існує різниця між програмним забезпеченням режиму користувача, що не є частиною КЗЗ, яке можна завантажувати та запускати в системі, та програмним забезпеченням, яке повинно бути виключене із системи
- ❑ Наступні методи використовуються для того, щоб гарантувати, що виключене програмне забезпечення не можна буде використовувати для порушення політики безпеки системи:
 - Додавання модулів ядра не дозволяється
 - Програмне забезпечення для інсталяції може змінити конфігурацію (наприклад, біти режиму), так що програма не зможе порушити політику безпеки
 - Додавання програм із увімкненим бітом SUID, якими володіє UID root або системний UID, заборонено
 - Додавання програм із увімкненим бітом SGID, якими володіє GID root або системний GID, заборонено
 - Додавання ввімкнених бітів можливостей (capabilities) програм не дозволяється
 - Додавання демонів, що виконуються з UID/GID root або системним, не дозволяється
 - Зміни набору правил фреймворків, які дозволяють появу процесів з різними привілеями, такими як udev або PolicyKit, не дозволяються
 - Додавання програм із увімкненими можливостями (capabilities) подолання обмежень MLS не дозволяється

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux.

Лекція 14. Архітектура ПЗ Linux

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

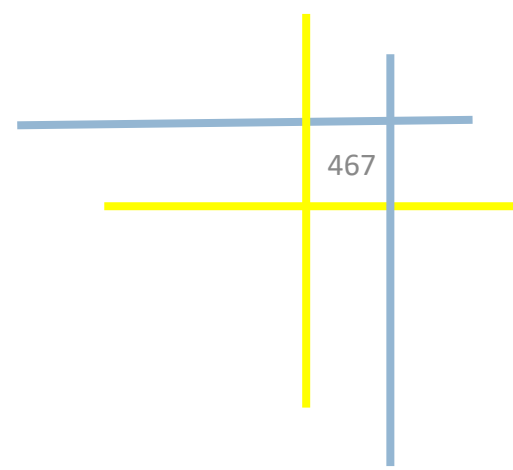
Укладач: проф. Дорогий Я.Ю.



Лекційні питання

466

- ☐ Архітектура ПЗ
- ☐ Управління доступом
- ☐ Можливості
- ☐ Ядро як компонент КЗЗ



Архітектура програмного забезпечення



Апаратні привілеї — рівень привілеїв x86

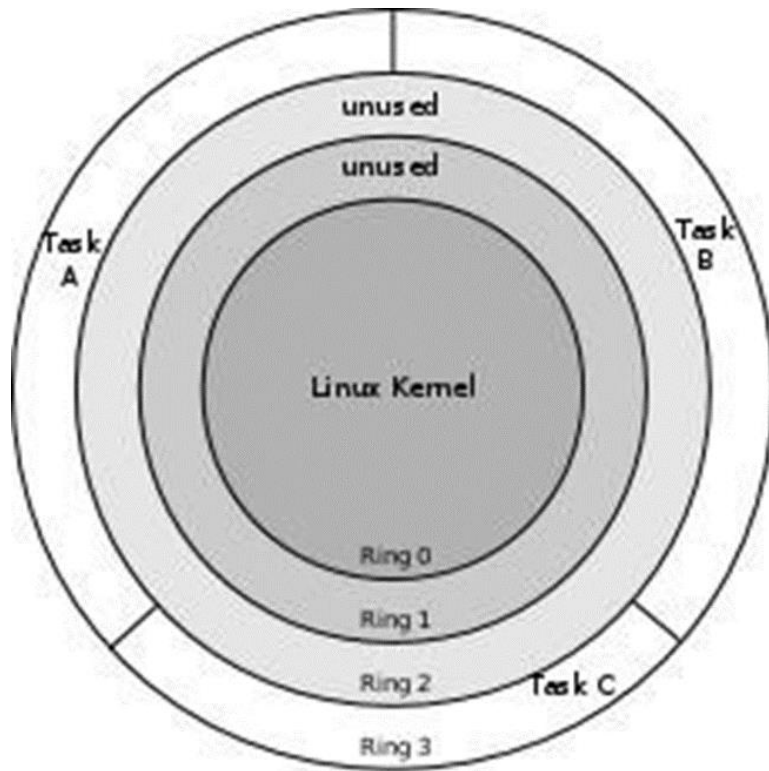
468

- ❑ Концепція привілею реалізується шляхом присвоєння значення від 0 до 3 ключовим об'єктам, які розпізнає процесор
 - Це значення називається рівнем привілеїв.
- ❑ Наступні об'єкти, які розпізнає процесор, містять рівні привілеїв:
 - Дескриптори містять поле, яке називається рівнем привілеїв дескриптора (descriptor privilege level — DPL)
 - Селектори містять поле, яке називається рівнем привілеїв запитувача (requestor's privilege level — RPL)
 - RPL призначений для представлення рівня привілеїв процедури, яка ініціює селектор
 - Внутрішній реєстр процесора записує поточний рівень привілеїв (current privilege level — CPL)
 - Зазвичай CPL дорівнює DPL сегмента, який зараз виконує процесор CPL змінюється, коли керування передають сегментам з різними DPL



Рівні привілеїв, трактовані як рівні захисту

46
9



- ❑ Центр призначений для сегментів, що містять найбільш критичне програмне забезпечення, як правило, ядро операційної системи
- ❑ Зовнішні шари призначені для сегментів менш критичного програмного забезпечення
- ❑ Ядро Linux, як і більшість інших UNIX-подібних ядер, використовує лише два з цих режимів виконання
 - Найвищий, з рівнем привілеїв процесора 0, відповідає режиму ядра
 - Найнижчий, із рівнем привілеїв процесора 3, відповідає режиму користувача



Реалізація апаратного привілею

- ☐ Коли процесор перебуває в режимі ядра, програма має апаратний привілей
- ☐ Таким чином, будь-який код, який працює в режимі ядра, виконується з апаратними привілеями
- ☐ Програмне забезпечення, що працює з апаратними привілеями, включає:
 - Базове ядро Linux
 - Окремо завантажені модулі ядра, такі як багато модулів драйверів пристроїв
- ☐ Усе інше програмне забезпечення в системі зазвичай працює в режимі користувача, без апаратних привілеїв, включаючи процеси користувача, такі як оболонки, мережне клієнтське програмне забезпечення та редактори
- ☐ Процеси в режимі користувача виконуються з апаратними привілеями, коли вони ініціюють системний виклик



Рівні привілеїв PowerPC та SystemZ

□ Рівень привілеїв PowerPC

- Ця архітектура процесора забезпечує три режими виконання, ідентифіковані бітом PR (біт 49) та бітом HV (біт 3) Регістру стану машини (**Machine State Register — MSR**) процесора
 - Значення 0 для бітів PR та HV вказують на **режим гіпервізора**
 - Значення біта HV 1 і значення біта PR 0 означають **режим супервізора або ядра**
 - Значення біта HV 1 і значення біта PR 1 вказують на **режим користувача**

□ Рівень привілеїв SystemZ

- Системи SystemZ забезпечують два режими виконання, визначені бітом стану проблеми (або задачі) — **Problem State bit** (біт 15) слова стану програми (**Program Status Word — PSW**) процесора
 - Значення 0 вказує на **режим супервізора або ядра**
 - Значення 1 вказує на **режим стану проблеми або користувача**



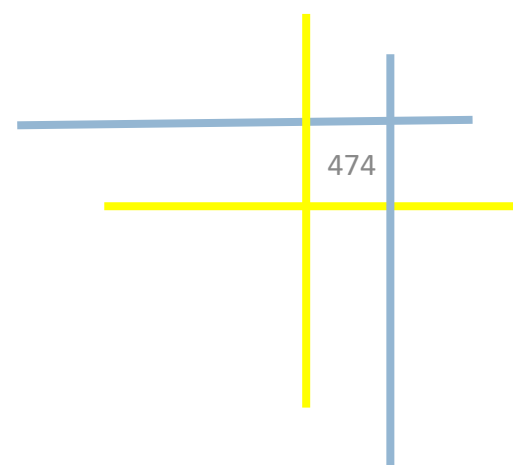
Розгляд питання віртуалізації

- ☐ Коли Linux використовується як хост-система для віртуалізації KVM, на додаток до двох згаданих вище використовується третій стан привілеїв: режим гіпервізора
- ☐ Ядро Linux використовує цей режим, коли активована віртуалізація KVM
- ☐ Для реалізації стану гіпервізора використовуються такі механізми процесора:
 - На базі Intel: VT-x
 - На базі AMD: SVM
 - SystemZ: інструкція SIE
- ☐ Ядро Linux не забезпечує підтримку віртуалізації для інших процесорів, окрім названих



Програмні привілеї

- ❑ Привілеї програмного забезпечення в Linux передбачають можливість подолати механізми керування доступом ядра
- ❑ Linux реалізує такі моделі керування доступом:
 - Дискреційне керування доступом (**Discretionary Access Control — DAC**), що застосовується до об'єктів зберігання
 - Перевірки безпеки на основі можливостей (**Capability-based security checks**) для обмеження виконання системних викликів або функціональних підмножин системних викликів для програм, що мають відповідні можливості
 - Перевірка доступу модуля безпеки Linux (**Linux Security Module — LSM**) — мандатні політики керування доступом (**Mandatory Access Control — MAC**), реалізовані за допомогою LSM
- ❑ При доступі до іменованих об'єктів спочатку застосовується дискреційне керування доступом, а перевірка доступу LSM застосовується тоді і тільки тоді, коли DAC надає доступ
- ❑ Окрім програмних привілеїв керування доступом, можливості (**Capabilities**) — це ще один програмний привілей, який може надаватися програмам



Управління доступом



Дискреційне керування доступом₁

475

- ❑ Модель DAC дозволяє власнику об'єкта вирішувати, хто і яким чином може отримати доступ до цього об'єкта
- ❑ Суб'єкти та об'єкти
 - Суб'єктами в Linux є регулярні процеси та потоки ядра
 - Вони обидва представлені структурою `task_struct`
 - Потоки ядра виконуються лише в режимі ядра і не обмежуються політикою DAC
 - Усі об'єкти зберігання, такі як звичайні файли, символьні та блокові файли пристроїв, каталоги, сокети, об'єкти IPC та кільця ключів ядра знаходяться під контролем політики DAC



Дискреційне керування доступом₂

476

□ Атрибути

- Атрибутами суб'єкта, що використовуються для забезпечення політики DAC, є UID процесу, GID, додаткові групи та можливості процесу
- Атрибутами об'єкта, що використовуються для забезпечення політики DAC, є власник, група-власник, біти дозволів та списки керування доступом (Access Control Lists — ACL) POSIX.1e для об'єктів файлової системи

□ Правила керування доступом

- Правила керування доступом DAC визначають, як певний процес із відповідними атрибутами DAC може отримати доступ до об'єкта з набором атрибутів захисту DAC
- Крім того, правила керування доступом DAC також визначають, як атрибути захисту суб'єкта та об'єкта переходять на нові значення та за яких умов

□ Програмні привілеї

- Привілеї програмного забезпечення для політики DAC засновані на можливостях

□ CAP_DAC_OVERRIDE, CAP_DAC_READ_SEARCH, які можна призначити процесу

- Процес, що має ці можливості, отримує програмний привілей щодо DAC, оскільки він може обійти політику керування доступом системи



Мандатне керування доступом SELinux LSM

- ❑ При Мандатному керуванні доступом (**Mandatory Access Control — MAC**), реалізованому за допомогою модуля безпеки Linux (**Linux Security Module — LSM**) SELinux (**Security Enhanced Linux**), саме політика безпеки системи (на відміну від власника в DAC) контролює, кому повинен бути наданий доступ і до якої інформації
- ❑ Політика MAC, що надається SELinux, базується на двох типах політик:
 - Застосування типу (Type Enforcement — TE), яке використовується для реалізації керування доступом на основі ролей
 - Багаторівнева безпека (Multi-Level Security — MLS) із ієрархічною політикою маркування у стилі Белла-ЛаПадула
- ❑ Суб'єкти та об'єкти
 - Суб'єкти в MAC — це звичайні процеси та потоки ядра (такі ж, як у DAC)
 - Потоки ядра виконуються лише в режимі ядра і не обмежуються політикою MAC
 - Усі іменовані об'єкти, такі як звичайні файли, символьні та блокові файли пристроїв, каталоги, сокети та об'єкти IPC, перебувають під контролем політики MAC
 - На додаток до іменованих об'єктів, політика MAC може також керувати доступом до певних структур даних ядра, таких як дескриптори файлів, повідомлення IPC та файлові системи, щоб дозволити детальний вираз системної політики безпеки



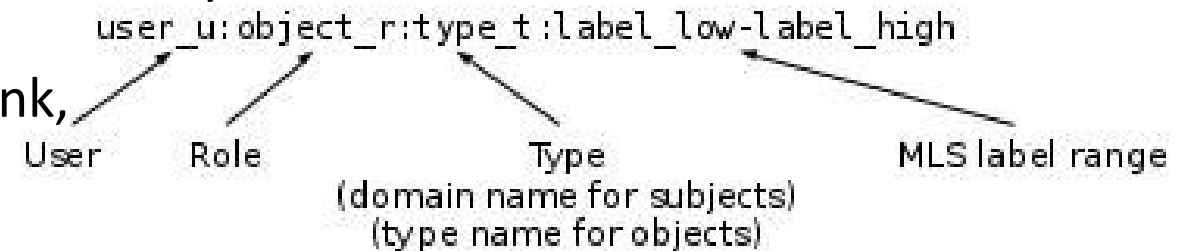
Атрибути SELinux

478

- ❑ Атрибути безпеки суб'єкта та об'єкта, також звані контекстом безпеки SELinux, мають однаковий формат
- ❑ У зручній для читання та зовнішній формі контекстом безпеки є рядок ASCII
 - Ядро з міркувань продуктивності перетворює рядок у 32-розрядне ціле число під час виконання

❑ Атрибути SELinux зберігаються:

- у структурі процесу `task_struct` для суб'єктів,
- у структурі даних `inode` в ядрі і на диску для об'єктів файлової системи,
- у структурі даних `kern_ipc_perm` для об'єктів System V IPC,
- у структурі даних `sock` для сокетів,
- у структурі `xfrm_state` для сокетів netlink,
- у структурі даних `key` для ключів.





Контексти безпеки SELinux

□ Контекст безпеки складається з чотирьох розділених двокрапкою компонентів

- **User** (Користувач): ім'я користувача SELinux
- **Role** (Роль): роль SELinux, що відповідає суб'єкту чи об'єкту
 - Для суб'єктів роль використовується для здійснення керування доступом на основі ролей, дозволяючи ролі контролювати доступ до доменів
 - Для об'єктів компонент ролі не використовується, і це завжди object_r
- **Type** (Тип): домен суб'єкта або тип об'єкта
 - Домени та типи — це еквівалентні класи для процесів та ресурсів відповідно
 - Рішення щодо доступу в ядрі приймаються на основі домену суб'єкта та типу об'єкта
- **MLS label range** (Діапазон міток MLS): дві повні мітки MLS.
 - Кожна мітка MLS складається з двох компонентів, рівня ієрархічної класифікації (або рівня чутливості) та неієрархічного набору категорій

□ SELinux записує недійсні контексти як

system_u:object_r:unlabeled_t:s0



Мітки MLS

- ☐ Кожна мітка MLS складається з двох компонентів
 - ієрархічний рівень класифікації (або рівень чутливості)
 - неієрархічний набір категорій
- ☐ Діапазон міток MLS містить дві повні мітки MLS
 - Вони розташовані з нижньою міткою ліворуч і верхньою міткою, що домінує над нижньою міткою, праворуч. Дві мітки розділені тире
- ☐ Для суб'єктів
 - Нижня мітка діапазону відповідає ефективній мітці MLS суб'єкта
 - Верхня мітка відповідає “допуску” (clearance) суб'єкта (максимальному рівню дозволу для цього суб'єкта)
 - Допуск суб'єкта відображує допуск користувача, від імені якого суб'єкт діє
- ☐ Для таких об'єктів, як звичайні файли, політика безпеки наказує, що нижня мітка та верхня мітка рівні
- ☐ Такі об'єкти, як каталоги, пристрої та сокети, можуть мати верхню мітку, яка не дорівнює нижній мітці
 - Ці об'єкти є багаторівневими, і їхній діапазон міток MLS вимагає, щоб рівень чутливості будь-якої інформації, яка проходить через них, домінував над нижньою міткою, а над ним домінувала верхня мітка



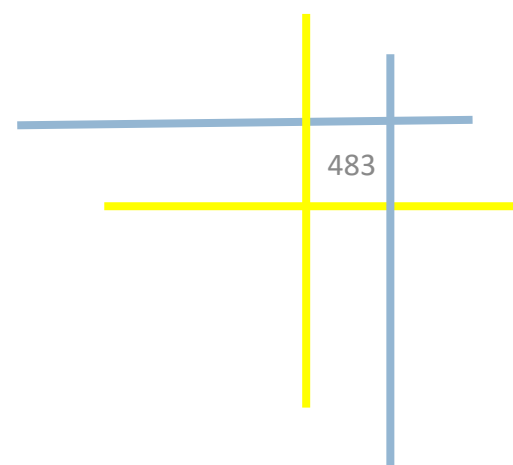
Правила керування доступом

- ❑ Реалізація керування доступом MAC базується на застосуванні типу (TE)
 - TE забезпечує детальне керування доступом для суб'єктів та об'єктів
 - Правила TE та MLS визначаються політикою SELinux
 - Правила TE визначаються як перевірки керування доступом, тоді як політика MLS виражається як обмеження поверх TE
- ❑ Отже, загальна перевірка керування доступом відбувається в три логічні етапи, наступним чином:
 - Перевірка DAC за допомогою атрибутів DAC
 - Перевірка TE з використанням доменів і типів контексту безпеки SELinux
 - Перевірка політики MLS з використанням діапазону MLS контексту безпеки SELinux
- ❑ Кожна перевірка є неавторитетною, тобто дозволи лише зменшуються на кожному етапі
 - заборона DAC не може бути скасована TE,
 - і заборона TE не може бути скасована MLS.



Програмні привілеї для MAC

- ❑ Програмні привілеї для політики MAC реалізуються з доменом процесу та правилами політики безпеки, пов'язаними з цим доменом
 - Наприклад, процесам, що виконуються в домені `init_t`, дозволено читати об'єкти, мітка чутливості яких домінує над міткою чутливості процесу
 - Це подолання загальної політики досягається атрибутами домену
- ❑ Правило політики описує доступ домену до типу з перевизначенням (подоланням) загальних правил, пов'язаним з атрибутом домену
 - Наприклад, наступне правило політики може бути використано, щоб дозволити процесу перевизначити обмеження моделі багаторівневої безпеки "не читати зверху" при спробі прочитати звичайний файл:
`mlsconstrain {file} {read} ((l1 dom l2) or (t1 == mlsfileread));`
 - Вищезазначений оператор встановлює політику безпеки, згідно з якою мітка **l1** (що належить суб'єкту) повинна домінувати над міткою **l2** (що належить об'єкту) для успішного виконання операції зчитування, або для цього домен процесу повинен мати атрибут **mlsfileread**
- ❑ Політика безпеки системи для різних типів суб'єктів та об'єктів описується з функціональним описом цих суб'єктів та об'єктів



Можливості



Можливості — Capabilities₁

- ❑ Ядро Linux має фреймворк для забезпечення привілеїв програмного забезпечення для багатьох різних функцій, що надаються ядром Linux за допомогою можливостей (Capabilities)
 - Ці можливості дозволяють розділити привілей програмного забезпечення ядра, пов'язаний з нульовим ідентифікатором користувача (UID = 0), на набір дискретних привілеїв на основі операцій, які намагаються здійснити
- ❑ Можливості інтегруються з ідентифікаторами користувачів (UID) наступним чином:
 - Якщо виконується SUID програма з ідентифікатором власника root, для цього процесу увімкнені усі можливості
 - Якщо використовується сімейство системних викликів **set*uid** для зміни UID процесу, що викликає, на ідентифікатор, що не дорівнює 0, усі можливості цього процесу видаляються
 - Процеси, породжені користувачем root, спочатку мають усі можливості
 - Процеси, породжені користувачами, що не є root, спочатку не мають увімкнених можливостей



Можливості — Capabilities₂

□ Програмний привілей

- Ядро обмежує використання багатьох системних викликів або функціональних областей, що викликаються системними викликами, до процесів, що мають певні можливості
- Якщо процес виклику не має необхідних можливостей, виконання зачепленої області коду не дозволяється, а процесу, що викликає, повертається помилка **EPERM**
 - Наприклад, якщо процес намагається створити спеціальний файл пристрою, викликаючи системний виклик **mknod**, ядро перевіряє, чи здатний процес створювати спеціальні файли пристрою, перевіряючи, чи має процес можливість **CAP_MKNOD**
- За відсутності спеціальних модулів ядра, які визначають і використовують можливості, перевірка можливостей повертається назад до надання привілеїв програмного забезпечення ядра на основі UID процесу

□ Можливості засновані на проекті **POSIX.1e** плюс ряд додаткових можливостей

- Повний перелік усіх можливостей, включаючи їх значення, повністю описаний у файлі вихідного коду ядра Linux **include/linux/capability.h**



Особливості AppArmor (Ubuntu)

- ❑ В той час, коли дистрибутиви Linux гілки Red Hat (RHEL, SUSE) для реалізації MAC використовують **SELinux**, Ubuntu та інші дистрибутиви гілки Debian використовують **AppArmor**
 - AppArmor — це також LSM
 - AppArmor абсолютно несумісний з SELinux
 - AppArmor також несумісний з Capabilities, бо фактично цей модуль підтримує власну систему Capabilities
- ❑ На відміну від SELinux, AppArmor не надає політику багаторівневої безпеки, але надає повноцінний аналог застосуванню доменів/типів
- ❑ Політика AppArmor формується у вигляді профілів для кожного виконаного файлу
 - Профіль — це текстовий файл, який можна створювати вручну або з допомогою інструментів AppArmor
 - У профілі, зокрема, прописуються можливості (capabilities) цієї програми, а також права доступу (дозволи) до усіх доменів (каталогів, окремих файлів), до яких ця програма може мати доступ



Застосування AppArmor для реалізації MAC в Ubuntu

487

□ Ubuntu підтримує мандатне керування доступом на основі такої концепції:

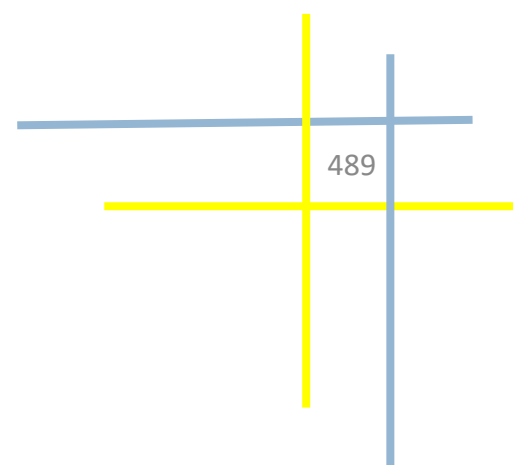
- Для розділення віртуальних машин та їхніх ресурсів під час виконання, застосовуються правила AppArmor, визначені політиками AppArmor
- Фреймворк управління віртуальними машинами (virtual machine management framework) призначає кожній віртуальній машині унікальну мітку (label)
- Згідно цієї політики, кожному ресурсу, призначеному цій віртуальній машині, призначають ту саму мітку
- Політика AppArmor на основі цих міток гарантує, що віртуальна машина може отримати доступ лише до своїх власних ресурсів
 - Ресурси інших віртуальних машин, операційної системи хосту, і ресурси інших користувачів є недосяжними

□ Фреймворк управління віртуальними машинами застосовує AppArmor прозоро для адміністратора



Програми з програмними привілеями

- ❑ Прикладами програм, що працюють із програмним привілеєм, є:
 - Програми, які запускаються системою, такі як демони cron і init
 - Програми, які запускаються довіреними адміністраторами для виконання системного адміністрування
 - Програми, які працюють із правами привілейованого користувача шляхом виконання програмних файлів SUID/SGID можливостями файлової системи
 - Програми, що виконуються довіреними супердемонами. Присутні такі супердемони з можливістю породження прикладних програм з різними привілеями від імені користувачів, що їх викликають:
- ❑ Все програмне забезпечення (ПЗ), яке працює з апаратними або програмними привілеями, є частиною КЗЗ
- ❑ У системі, яку адмініструють належним чином:
 - Непривілейоване ПЗ підпорядковується політиці безпеки системи і не має жодних засобів обходу цих механізмів — це називається недовіреним (або ненадійним) ПЗ
 - Довірені процеси, які не реалізують жодної функції безпеки, повинні бути захищені від несанкціонованого втручання за допомогою функцій безпеки Linux, щоби гарантувати, що вони не виконують жодної функції, яка порушує політику безпеки Linux



Ядро як компонент КЗЗ



Ядро як компонент КЗЗ

- ❑ Ядро є центральною частиною операційної системи. Воно:
 - Взаємодіє безпосередньо з апаратним забезпеченням,
 - Здійснює спільний доступ до ресурсів, надання загальних послуг програмам та
 - Забороняє програмам отримувати безпосередній доступ до апаратно-залежних функцій
- ❑ Ядро Linux — це повністю витісняльне (preemptible) ядро
 - У невитісняльних ядрах код ядра працює до завершення
 - У витісняльних ядрах можна витіснити завдання в будь-який момент, якщо ядро перебуває у стані, в якому безпечно перепланувати його
- ❑ Сервіси, що надаються ядром, включають наступне:
 - Керування виконанням процесів, дозволяючи їх створення, завершення або призупинення, а також зв'язок
 - Виділення основної пам'яті для виконуваного процесу
 - Обслуговування віртуальних машин протягом життєвого циклу
 - Обслуговування файлової системи



Сервіси, що надаються ядром

- ☐ Керування виконанням процесів, дозволяючи їх створення, завершення або призупинення, а також зв'язок
- ☐ Виділення основної пам'яті для виконуваного процесу
- ☐ Обслуговування віртуальних машин протягом їх життєвого циклу
- ☐ Обслуговування файлової системи



Логічні компоненти ядра₁

492

❑ Файлова підсистема та підсистема вводу-виводу:

- Ця підсистема реалізує функції, пов'язані з об'єктами файлової системи
- Реалізовані функції включають функції, які дозволяють процесу створювати, підтримувати, взаємодіяти та видаляти об'єкти файлової системи
- Ці об'єкти включають звичайні файли, каталоги, символічні посилання, жорсткі посилання, спеціальні файли для пристроїв, іменовані канали та сокети

❑ Підсистема керування процесами:

- Ця підсистема реалізує функції, пов'язані з управлінням процесами та потоками
- Реалізовані функції включають функції, що дозволяють створювати, планувати, виконувати та видаляти процеси та потоки

❑ Підсистема керування пам'яттю:

- Ця підсистема реалізує функції, пов'язані з управлінням ресурсами пам'яті системи
- Реалізовані функції включають функції, що створюють віртуальну пам'ять та керують нею, включаючи управління таблицями сторінок та алгоритмами підкачки

❑ Мережна підсистема:

- Ця підсистема реалізує сокети доменів UNIX та Інтернету, а також алгоритми планування мережних пакетів



Логічні компоненти ядра2

493

❑ Підсистема IPC:

- Ця підсистема реалізує функції, пов'язані з механізмами IPC
- Реалізовані функції включають функції, що полегшують контрольований обмін інформацією між процесами, дозволяючи їм обмінюватися даними та синхронізувати їх виконання, щоб взаємодіяти із загальним ресурсом

❑ Підсистема фреймворку ядра:

- Ця підсистема реалізує інфраструктуру для підтримки ядром різних механізмів
- Ця підсистема включає такі функції, серед інших:
 - Підтримка завантажуваних модулів: реалізовані функції включають функції завантаження, ініціалізації та вивантаження модулів ядра
 - Обробка винятків, таких як перемикання контексту, завантаження системних викликів тощо

❑ Аудит:

- Підсистема аудиту реалізує функції, пов'язані із записом критично важливих подій в системі
- Реалізовані функції включають функції, які фіксують кожен системний виклик для запису критично важливих подій та функції, що реалізують збір та запис даних аудиту



Логічні компоненти ядра₃

494

❑ Розширення безпеки Linux:

- Розширення безпеки Linux (Linux Security Extensions) реалізують різні аспекти, пов'язані з безпекою, які надаються всьому ядру, включаючи фреймворк модуля безпеки Linux (Linux Security Module — LSM)
- Фреймворк LSM забезпечує агностичну основу для модулів для реалізації різних політик безпеки, включаючи SELinux
 - SELinux є важливою логічною підсистемою
 - Ця підсистема реалізує функції мандатного керування доступом для опосередкування доступу між усіма суб'єктами та об'єктами

❑ Підсистема драйверів пристроїв:

- Ця підсистема реалізує підтримку різних апаратних та програмних пристроїв через загальний, незалежний від пристроїв інтерфейс

❑ Підсистема KVM:

- Ця підсистема реалізує підтримку життєвого циклу віртуальної машини
- Вона включає виконання інструкцій для інструкцій, що вимагають лише невеликих перевірок
- Для виконання будь-яких інших інструкцій KVM викликає компонент простору користувача QEMU

❑ Кристо API:

- Ця підсистема забезпечує внутрішню криптографічну бібліотеку ядра для всіх компонентів ядра
- Вона надає криптографічні примітиви



Виконувані компоненти ядра₁

□ Базове ядро

- Базове ядро включає код, який виконується для надання сервісу, наприклад, обслуговування системного виклику користувача або обслуговування події переривання або винятку
- Більшість компільованого коду ядра підпадає під цю категорію

□ Потоки ядра

- Для виконання певних рутинних завдань, таких як очищення кеш-пам'яті диска або відновлення пам'яті шляхом заміни невикористаних фреймів, ядро створює внутрішні процеси або потоки
- Потоки ядра плануються так само, як і звичайні процеси, але вони не мають контексту в режимі користувача
- Потоки ядра знаходяться в просторі ядра і працюють лише в режимі ядра



Виконувані компоненти ядра₂

□ Модулі ядра та драйвери пристроїв

- Модулі ядра — це фрагменти коду, які можна завантажувати в ядро та вивантажувати та з нього за запитом
- Драйвер пристрою — це особливий тип модуля ядра, який дозволяє ядру отримувати доступ до апаратного забезпечення, підключеного до системи



Неядерне ПЗ і бази даних КЗЗ

- ☐ Неядерне ПЗ КЗЗ складається з довірених програм, які використовуються для реалізації функцій безпеки
- ☐ *Зверніть увагу, що спільні бібліотеки, включаючи модулі РАМ, в деяких випадках використовуються довіреними програмами. Однак немає жодного випадку, коли спільна бібліотека сама по собі вважається довіреною сутністю*
- ☐ Довірені команди можна згрупувати наступним чином
 - Ініціалізація системи
 - Ідентифікація та автентифікація
 - Мережні програми
 - Пакетна обробка
 - Управління системою
 - Аудит рівня користувача
 - Криптографічна підтримка
 - Підтримка віртуальної машини
 - Обробка авторизації в користувальницькому просторі
- ☐ Бази даних КЗЗ — це файли конфігурації довірених програм
 - Жодна з цих баз даних не може бути змінена користувачем, крім адміністративного користувача
 - Керування доступом здійснюється компонентом файлової системи ядра Linux



Апаратне забезпечення та прошивка

498

□ Апаратне забезпечення

- Апаратне забезпечення складається з фізичних ресурсів, таких як
 - ЦП,
 - основна пам'ять, регістри,
 - кеші,
 - та пристрої, які ефективно складають комп'ютерну систему

□ Вбудоване програмне забезпечення (прошивка, firmware)

- Прошивка складається з програмного забезпечення, що знаходиться в апаратному забезпеченні, яке запускається, коли система переходить у режим скидання живлення
- На додаток до ініціалізації апаратного забезпечення та запуску операційної системи, на платформах, що підтримують логічне розділення (partitioning), прошивка забезпечує також підтримку логічного розділення

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux. Лекція 15. Функціональність безпеки Linux

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

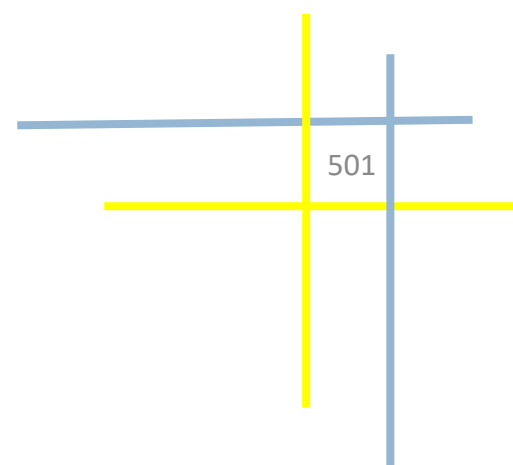
Укладач: проф. Дорогий Я.Ю.



Лекційні питання

500

- ☐ Аудит
- ☐ Криптографічна підтримка
- ☐ Пакетний фільтр
- ☐ Ідентифікація та автентифікація



Аудит



Аудит

- ❑ Легка система аудиту (Lightweight Audit Framework — LAF) розроблена для того, щоб бути системою аудиту для Linux, яка відповідає вимогам Загальних критеріїв
 - LAF може перехоплювати всі системні виклики, а також отримувати записи журналу аудиту з привілейованих програм простору користувача
 - Підсистема дозволяє конфігурувати події, які фактично будуть підлягати аудиту, з набору всіх подій, які можна піддати аудиту
 - Ці події конфігуруються у певному конфігураційному файлі, а потім ядро отримує повідомлення про створення власної внутрішньої структури для подій, які підлягають аудиту
- ❑ **auditd** — це компонент простору користувача системи аудиту Linux
 - Він відповідає за запис записів аудиту на диск.
 - Перегляд журналів здійснюється за допомогою утиліт ausearch або aureport
 - Налаштування системи аудиту або правил завантаження здійснюється за допомогою утиліти
- ❑ **auditctl**
 - Під час запуску auditctl читає правила в /etc/audit/audit.rules і завантажує їх в ядро
 - Крім того, існує також програма augenrules, яка читає правила, розташовані в /etc/audit/rules.d/, та компілює їх у файл audit.rules
 - Сам демон аудиту має деякі параметри конфігурації, які адміністратор, можливо, захоче налаштувати. Вони знаходяться у файлі auditd.conf



Функціональність аудиту₁

❑ Ядро Linux реалізує основу функціональності LAF

- Воно збирає всі події, що підлягають аудиту, аналізує ці події на основі правил аудиту та пересилає ті події, які вимагається піддати аудиту, до демона аудиту, що виконується в просторі користувача
- Події аудиту генеруються в різних місцях ядра
- Програма простору користувача може створювати записи аудиту, які потрібно подавати в ядро для подальшої обробки

❑ Функціональність аудиту ядра Linux налаштовується програмами простору користувача, які взаємодіють з ядром за допомогою спеціального каналу зв'язку **netlink**

- Цей канал зв'язку також повинен використовуватись програмами, які хочуть надіслати подію аудиту до ядра

❑ Інтерфейс ядра netlink використовується лише програмами, що мають такі можливості:

- **CAP_AUDIT_CONTROL**: Виконання операцій управління, таких як додавання або видалення правил аудиту, встановлення або отримання параметрів аудиту;
- **CAP_AUDIT_WRITE**: Надсилання записів аудиту до ядра, яке, в свою чергу, пересилає записи аудиту до демона аудиту.



Функціональність аудиту₂

- ☐ Виходячи з правил аудиту, ядро вирішує, чи подія аудиту ігнорується, чи вона надсилається до демона аудиту простору користувача для збереження її в протоколі аудиту
- ☐ Ядро знову відправляє повідомлення демону аудиту за допомогою згаданого вище каналу зв'язку **netlink**
- ☐ Доступ до даних аудиту звичайним користувачам заборонений функцією DAC
 - Доступ до журналу аудиту та файлів конфігурації аудиту дозволений лише системному адміністратору
- ☐ Системний адміністратор може визначити події, що підлягають аудиту, використовуючи прості вирази фільтру
 - Системний адміністратор також може визначити набір ідентифікаторів користувачів, для яких аудит активний, або ж набір ідентифікаторів користувачів, аудит яких не ведеться
 - Окремі файли можна налаштувати для перевірки, додавши їх до списку спостереження, який завантажується в ядро



Функціональність аудиту₃

- ☐ Правила аудиту можна вказати для формування аудиторських даних на основі великої кількості різних атрибутів
- ☐ Система аудиту може бути налаштована на дії, якщо протокол аудиту заповнений або досягає заданого обсягу дискового простору
 - Дії, які можна налаштувати, включають зупинку роботи системи, запобігання подальшим контрольованим діям, сповіщення адміністратору чи виконання налаштованої команди
- ☐ Linux надає програму управління, яка використовує згаданий інтерфейс netlink
 - Ця програма використовується під час завантаження системи для завантаження правил аудиту з конфігураційного файлу `/etc/audit/audit.rules`
 - Правила аудиту можуть бути змінені під час роботи системи



Протокол аудиту (Audit trail)₁

- ❑ Запис аудиту складається з одного або декількох рядків тексту, що містять поля у форматі з тегами «**ключове_слово=значення**»
- ❑ Наступна інформація міститься в усіх рядках записів аудиту:
 - **Type** (Тип): вказує джерело події, наприклад SYSCALL, PATH, USER_LOGIN або LOGIN
 - **Timestamp** (Мітка часу): Дата та час створення аудиторського запису
 - **Audit ID**: унікальний числовий ідентифікатор події
 - **Login ID**: Ідентифікатор користувача, автентифікований системою ("auid") — незалежно від того, чи змінив користувач свій реальний та/або ефективний ідентифікатор згодом
 - **Effective user and group ID**: ефективний ідентифікатор користувача та групи процесу на момент створення події аудиту
 - Успіх або невдача (де це доречно)
 - **Ідентифікатор процесу суб'єкта**, що спричинив подію (**PID**)
 - **Ім'я хосту або термінал**, який суб'єкт використав для виконання операції
 - Інформація про заплановану операцію



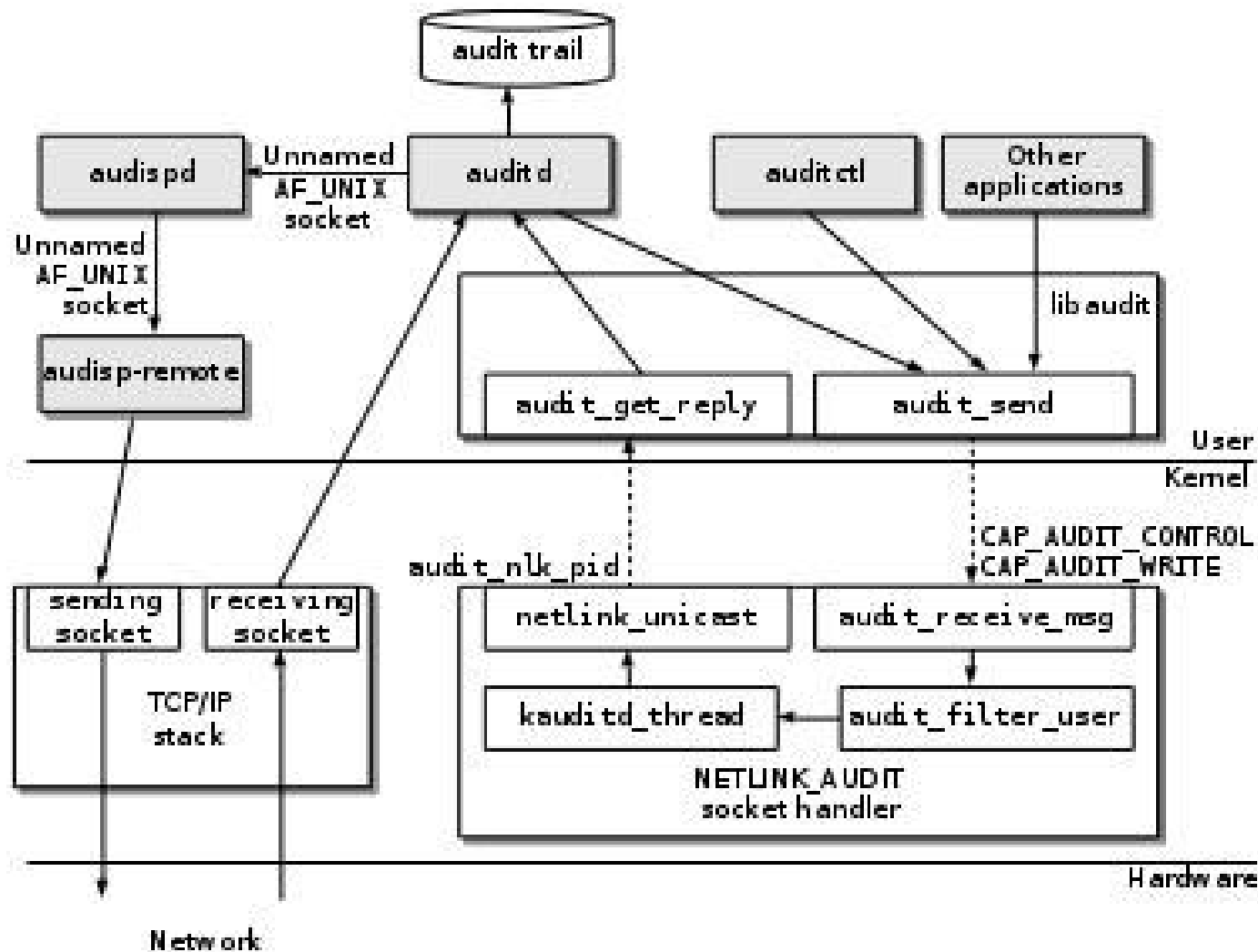
Протокол аудиту (Audit trail)₂

- ☐ Зазначена вище інформація супроводжується даними, специфічними для конкретної події
- ☐ У деяких випадках, таких як записи подій **SYSCALL**, що включають об'єкти файлової системи, для однієї події буде сформовано кілька текстових рядків, всі вони мають однакову мітку часу та ідентифікатор аудиту, щоб забезпечити легку кореляцію
- ☐ Протокол аудиту зберігається у вигляді тексту ASCII
- ☐ Протокол аудиту зберігається у файлах, доступ до яких має лише адміністратор



Фреймворк аудиту

508





Питання, що не включені до цієї лекції

509

(рекомендується ознайомитись з документацією)

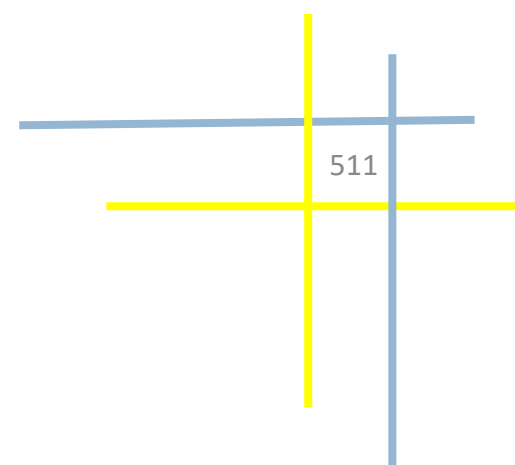
- ☐ Інтерфейс ядро-користувацький простір
- ☐ Розширення структури завдань для аудиту
- ☐ Аудит системних викликів
- ☐ Формування записів аудиту викликів сокета та IPC
- ☐ Аудит файлової системи
- ☐ Аудит інших дій ядра
- ☐ Ініціалізація аудиту ядра
- ☐ Формат запису аудиту
- ☐ Підтримка аудиту IPTables
- ☐ Підтримка аудиту OpenSSH
- ☐ Підтримка міток часу



auditd vs rsyslogd

510

- ❑ **Syslog** (rsyslog, syslog-ng) та **Audit** мають різні цілі
- ❑ Щоби переконатися, що у системі, де працюють обидва, вони не роблять одне і те саме, можна перевірити конфігураційні файли кожного:
 - **/etc/audit/auditd.conf** — файл конфігурації для auditd
 - **/etc/audit/audit.rules** — правила аудиту для auditd, завантажені під час запуску
 - **/etc/rsyslog.conf** — файл конфігурації для rsyslogd
- ❑ Взагалі, є сенс мати обидва демони в роботі
 - rsyslogd можна використовувати локально та віддалено для реєстрації, він працює через сокети Інтернету та Unix
 - auditd має налаштовані правила, які запускаються під час запуску та завантажуються в ядро
 - Якщо видалити rsyslogd, ведення журналу не буде працювати
 - Якщо видалити auditd, не будуть реєструватись повідомлення ядра (зазвичай — у **/var/log/messages** та(або) **/var/log/audit**)



Криптографічна підтримка



Криптографічні сервіси

- ☐ Linux забезпечує криптографічно захищені мережні канали зв'язку, що дозволяє віддаленим користувачам взаємодіяти з системою
- ☐ Користуючись одним із наступних криптографічно захищених мережних каналів, користувач може запитувати такі сервіси:
 - OpenSSH
 - Libvirt
 - VNC
 - IPSec
- ☐ На додаток до криптографічно захищених каналів зв'язку, Linux також забезпечує криптографічні алгоритми для загального користування
- ☐ Криптографічні примітиви для реалізації вищезазначених протоколів криптографічного зв'язку забезпечуються OpenSSL



Протокол SSHv2

- ❑ Linux підтримує такі функції безпеки протоколу SSH v2.0:
 - Встановлення захищеного каналу зв'язку за допомогою наступних криптографічних функцій, передбачених протоколом SSH v2.0
 - Виконання автентифікації користувача за допомогою стандартного методу автентифікації на основі пароля, який Linux забезпечує для користувачів (метод автентифікації за паролем, як визначено у **главі 5 RFC4252**)
 - Виконання автентифікації користувача за допомогою методу автентифікації на основі ключів RSA, DSA або ECDSA (метод автентифікації з відкритим ключем, як визначено в **главі 5 RFC4252**)
 - Перевірка цілісності повідомлень, якими обмінюються, та переривання з'єднання у випадку виявлення помилки цілісності
- ❑ Застосунки OpenSSH sshd, ssh та ssh-keygen використовують генератор випадкових чисел **OpenSSL**, ініціалізований **/dev/random** або **/dev/urandom**, для створення криптографічних ключів
 - OpenSSL забезпечує різні DRNG (цифрові генератори випадкових чисел), залежно від того, чи в системі ввімкнено режим **FIPS 140-2**



Сімейство протоколів IPSEC та IKEv2₁

- ❑ Linux реалізує сімейство протоколів **IPSEC** та **IKE**
- ❑ Ядро Linux проводить обробку **ESP/АН**, а користувацька програма **strongSwan** охоплює обробку набору протоколів IKE наступним чином:
 - Обмін ключами в Інтернеті (**Internet Key Exchange — IKE**)
 - Протокол IKE встановлює спільний сеансовий ключ, який використовується для шифрування зв'язку
 - Крім того, як частина протоколу IKE виконується ключова угода для **ISAKMP SA**, яка захищає всю комунікацію IKE
 - Протокол IKEv2 (описаний у RFC5996) реалізований демоном **charon** і забезпечений виключно кодом простору користувача
 - Криптографічні примітиви, необхідні для IKE, надаються OpenSSL
- ❑ **IPSec**:
 - Після обміну або узгодження ключів для **IPSEC SA**, шифрування та дешифрування фактичних даних, що передаються мережею, забезпечуються протоколами **IPSec ESP**, що потенційно підтримується АН
 - У Linux ядро реалізує виключно протоколи IPSec, використовуючи ключі, встановлені з протоколом IKE



Сімейство протоколів IPSEC та IKEv2₂

- ☐ Демон **charon** реалізує протокол IKEv2 (відповідно до RFC5996)
- ☐ Реалізація IPSec ядра підтримує і транспортний, і тунельний режими
 - Це дозволяє конфігурувати сценарії взаємодії користувач-користувач, користувач-мережа та мережа-мережа
- ☐ Для реалізації сімейства протоколів IPSec підтримуються такі RFC:
 - **RFC4301, RFC4303**: Визначення SPD/SAD, SA, ESP
 - **RFC4306, RFC4307**: IKEv2 з ISAKMP, група Діффі-Хеллмана
 - **RFC3526, RFC5114, RFC6954**: Групи Діффі-Хеллмана
 - RFC4106: Режим AES GCM для IPSec
 - **RFC4303, RFC4307**: Пізні типи шифру для IPSec
 - **RFC4309**: Режим AES CCM для IPSec
 - **RFC5996**: IKEv2
- ☐ Криптографічні реалізації забезпечують належне обнуління конфіденційних даних перед звільненням пов'язаної пам'яті



Сховище даних із захистом конфіденційності₁

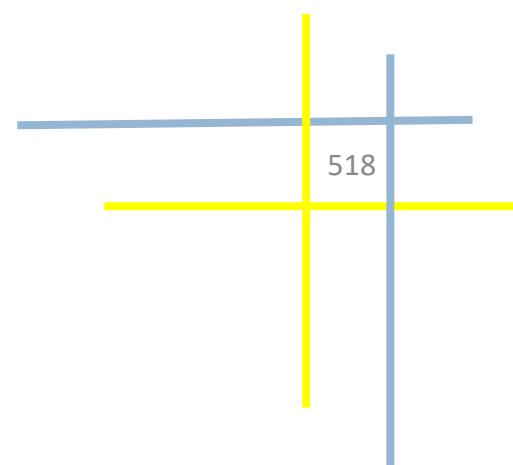
- ☐ Об'єкти файлової системи зберігаються на блокових пристроях, таких як розділи на жорсткому диску
- ☐ Операційні системи Linux пропонують використання додаткового рівня між файловими системами та фізичним блоковим пристроєм для шифрування та дешифрування будь-яких даних, що передаються між файловою системою та блоковим пристроєм
- ☐ Функціонал **dm_crypt** використовує перетворювач (mapper) пристроїв Linux для забезпечення такої операції шифрування та дешифрування, яка є прозорою для файлової системи, а отже і для користувача
- ☐ Перш ніж монтувати блоковий пристрій, який захищений схемою шифрування **dm_crypt**, власник зашифрованого блокового пристрою повинен надати парольну фразу
- ☐ Якщо адміністратор отримує прямий доступ до обладнання, на якому розміщений блоковий пристрій, можна буде прочитати лише зашифровані дані



Сховище даних із захистом конфіденційності₂

517

- ☐ Операція шифрування та дешифрування пристрою реалізована ядром
- ☐ Щоб розблокувати зашифрований ключ розділу, що зберігається на захищеному блоковому пристрої, програма **cryptsetup** виконує такі дії:
 - отримує пароль користувача
 - застосовує механізм отримання ключів **LUKS** до паролі фрази
 - зчитує зашифрований ключ розділу з пристрою
 - розшифровує ключ розділу за допомогою ключа, отриманого з паролі фрази користувача
 - вводить в ядро розшифрований ключ розділу та налаштовує відображення між блоковим пристроєм та ключем розділу
- ☐ Використовуючи інструмент **cryptsetup**, ключ розділу можна також передати, зашифрувавши його за допомогою іншої паролі фрази, яка може бути надана іншому користувачеві
- ☐ Аналогічно, інструмент **cryptsetup** можна використовувати для стирання місця зберігання одного зашифрованого ключа розділу
- ☐ Ключовий матеріал, який використовується **cryptsetup** для механізму шифрування диска, походить з **/dev/urandom** та **XORing** 13 байт з **/dev/random** у генератор випадкових чисел стану
- ☐ У режимі FIPS 140-2 для генерації ключів використовується **libcrypt DRBG**, ініціалізований **/dev/urandom** XORing 13 байт з **/dev/random**
- ☐ Інсталятор, за винятком IBM System Z, дозволяє конфігурувати схему шифрування всього диска, де весь диск захищений, крім розділу **/boot**



Пакетний фільтр



Пакетний фільтр

- ❑ Реалізація мережного стеку ядра Linux відповідає структурі рівнів мережних протоколів
- ❑ Реалізовано код для обробки канального (data link, 2) і мережного (network, 3) рівнів
- ❑ Для цих рівнів передбачений незалежний механізм фільтрації:
 - Канальний рівень: **ebtables** реалізує механізм фільтрації для мостів
 - Мережний рівень: **netfilter/iptables** реалізує механізм фільтрації для інтерфейсів, що не є мостами
- ❑ Правила фільтрації пакетів можуть бути введені в ядро Linux для примусового застосування процесами, що мають можливість **CAP_NET_ADMIN**



Фільтрація мережного рівня — Netfilter

520

❑ **Netfilter** — це фреймворк для керування пакетами, реалізований у мережному стеку ядра Linux, який обробляє мережний рівень

❑ Структура netfilter складається з наступних частин:

- Стек IP визначає п'ять хуків, які є чітко визначеними точками на шляху проходження мережним пакетом стека протоколу IP
- Фреймворк netfilter забезпечує реєстрацію функцій для інших частин ядра для прослуховування різних хуків
- Коли пакет позначений для черги до простору користувача, фреймворк netfilter обробляє асинхронний зв'язок з користувацьким простором



П'ять хуків фреймворку Netfilter

- ☐ Коли пакет входить в мережний рівень реалізації стеку протоколів і після застосування деяких перевірок коректності, але перед тим, як звернутися до таблиці маршрутизації, спрацьовує хук **NF_IP_PRE_ROUTING**
- ☐ Коли пакет генерується локально, хук **NF_IP_LOCAL_OUT** спрацьовує перед зверненням до таблиці маршрутизації
- ☐ Після проходження таблиці маршрутизації, якщо код маршрутизації позначає, що пакет буде спрямовано на інший хост, спрацьовує хук **NF_IP_FORWARD**
- ☐ Після проходження таблиці маршрутизації, якщо код маршрутизації позначає, що пакет буде спрямовано в локальну систему, спрацьовує хук **NF_IP_LOCAL_IN**
- ☐ Коли пакет пройшов весь мережний стек і готовий до відправлення, спрацьовує хук **NF_IP_POST_ROUTING**



Фільтрація мережного рівня — IPTables

- ❑ Усім зв'язком на мережному рівні може керувати фреймворк IPTables
- ❑ Поєднання IPTables та netfilter реалізує фільтр пакетів, який забезпечує **stateful**- і **stateless**- фільтрування пакетів шляхом перевірки IP-заголовка, заголовка TCP, заголовка UDP та/або заголовка ICMP кожного мережного пакету, що проходить через мережний стек
- ❑ Система відбору пакетів під назвою IPTables використовує фреймворк netfilter для реалізації фактичної логіки фільтрації пакетів на мережному рівні для сімейства протоколів TCP/IP
 - Примітка: IPTables може виконувати трансляцію мережних адрес (NAT), а також трансляцію адрес портів (PAT) як для простих, так і для більш складних протоколів
 - Крім того, за допомогою IPTables надається підтримка маніпуляцій з пакетами
- ❑ IPTables реєструє всі хуки, надані фреймворком netfilter
 - Механізм NAT/PAT використовує хуки pre-routing та post-routing
 - Можливість фільтрації пакетів застосовується на хуках local-in, local-out та forwarding



Архітектура IPTables

523

❑ IPTables складається з наступних двох компонентів:

- Застосування пакетного фільтру в ядрі:
 - Компонент ядра IPTables використовує фреймворк netfilter
 - Механізмом ядра забезпечуються три списки правил фільтрації пакетів: один для кожного хуку фреймворку netfilter, що застосовується для фільтрації пакетів
 - Кожен список містить нуль або більше правил, які перевіряються послідовно
 - Правило складається з частини відповідності (також званої "розширенням відповідності") та частини дії (також званої "розширенням цілі")
- Програма конфігурації простору користувача:
 - Програма користувацького простору iptables дозволяє конфігурувати компоненти ядра IPTables
 - Програма дозволяє вказувати одне правило для кожного виклику, де правило містить згадані вище частину відповідності та частину дії
 - Цей інструмент також дозволяє змінювати або видаляти правила, що існують, а також конфігурувати дію за замовчуванням



Фільтрація канального рівня – ланцюжки ebtables

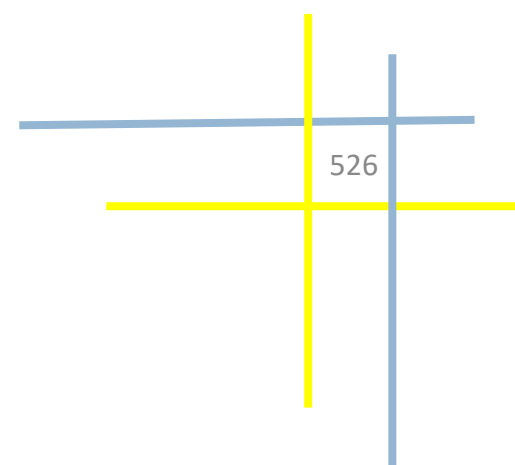
524

- ❑ Код канального рівня, що обробляє функціональність мосту, реалізує ланцюжки, які використовуються **ebtables** для застосування фільтрації
- ❑ Для реалізації ланцюжків ebtables використовується фреймворк netfilter у наступних точках логіки проходження пакетів:
 - Коли пакет потрапляє на канальний рівень Linux і після застосування деяких перевірок коректності, але до прийняття рішення мосту, запускається ланцюжок **BROUTING**
 - Після проходження ланцюжка **BROUTING**, але все ще до прийняття рішення мосту, спрацьовує ланцюг **PREROUTING**
 - Коли код мосту вирішує, що кадр не призначений для мосту, він пересилається на мережний рівень і обробка на канальному рівні припиняється
 - Після прийняття рішення мостової маршрутизації, якщо код маршрутизації позначає пакет, як призначений для іншого хоста, запускається ланцюжок **FORWARD**
 - Після прийняття рішення мостової маршрутизації, якщо код маршрутизації позначає пакет, як спрямований до локальної системи, спрацьовує ланцюжок **INPUT**
 - Якщо пакет генерується локально, і кадр позначено, як призначений для мосту, спрацьовує ланцюжок **OUTPUT**
 - Коли пакет пройшов всю логіку моста і готовий до відправлення, спрацьовує ланцюжок **POSTROUTING**



Фільтрація канального рівня – правила фільтрації ebttables

- ❑ Система вибору пакетів під назвою **ebtables** використовує мережний фільтр netfilter для реалізації фактичної логіки фільтрації пакетів для кадрів Ethernet, що проходять через мости
- ❑ Мостові мережні комунікації видно лише на канальному рівні
 - Оскільки хост використовує мости для підключення віртуальних машин до навколишнього середовища, для керування потоком трафіку можна використовувати лише ebttables
 - Дані, інкапсульовані в кадрах Ethernet, не передаються через мережний стек хост-системи, а тому не можуть бути проаналізовані та контрольовані вищеописаною структурою IPTables
 - Незважаючи на те, що мережний стек хост-системи не аналізує мостові кадри Ethernet, фреймворк ebttables приймає весь кадр і йому дозволено працювати на цьому кадрі
 - Отже, ebttables може аналізувати протокол усередині кадру Ethernet.
- ❑ Фреймворк ebttables може застосовувати фільтри на основі таких концепцій:
 - Відповідність кадру/пакету: Збіги базуються на протоколах Ethernet або MAC-адресах джерела та/або призначення
 - Дія, яку потрібно виконати на обраних кадрах: Подібно до IPTables, ebttables має бути налаштований для виконання дії з відповідним фреймом
 - Дія може полягати як у відхиленні пакету, так і в його прийнятті



Ідентифікація та автентифікація



Механізми ідентифікації та автентифікації на основі PAM₁

- ❑ Linux використовує набір бібліотек під назвою "Змінні модулі автентифікації" (**Pluggable Authentication Modules — PAM**), які дозволяють адміністративному користувачеві вибирати, як програми з підтримкою PAM автентифікують користувачів
- ❑ Linux надає модулі PAM, які реалізують усі функції безпеки для:
 - Забезпечення контролю входу та встановлення всіх UID, GID та ідентифікатора входу для суб'єкта
 - Забезпечення якості паролів
 - Застосування обмеження для облікових записів (наприклад, кількість максимально допустимих одночасних сеансів для користувача)
 - Застосування зміни паролів через налаштований час, включаючи контроль якості паролів
 - Примусового блокування облікових записів після невдалих спроб входу
 - Обмеження використання облікового запису root для певних терміналів
 - Обмеження використання команд su та sudo



Механізми ідентифікації та автентифікації на основі PAM₂

- ☐ Обробка входу встановлює реальний UID, UID файлової системи, ефективний UID і UID для входу, а також реальний GID, ефективний GID, GID файлової системи та набір додаткових GID суб'єкта, що створюється
- ☐ Клієнтська програма, яку зазвичай надає віддалена система, повинна правильно захистити введення користувачем пароля (наприклад, надавати лише приховану інформацію, як-то символи на екрані)
- ☐ Після успішної ідентифікації та автентифікації, Linux ініціює сеанс для користувача та створює початкову оболонку входу як перший процес, з яким користувач може взаємодіяти
- ☐ Linux забезпечує механізм блокування сеансу або автоматично після настрюваного періоду бездіяльності для цього сеансу, або за запитом користувача
- ☐ Облікові записи користувачів зберігаються у файлах конфігурації (/etc/passwd та /etc/shadow)



Підсистема ідентифікації та автентифікації (що було до PAM)

- ☐ 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`
- ☐ Автентифікація
 - Вхід в систему: 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`



Змінний модуль автентифікації

- ☐ PAM відповідає за підсистему ідентифікації та автентифікації
- ☐ PAM забезпечує централізований механізм автентифікації всіх служб
- ☐ PAM дозволяє обмежувати доступ до програм та альтернативні методи автентифікації, що налаштовуються
 - Для отримання більш детальної інформації про PAM див. Веб-сайт проекту PAM за адресою <http://www.kernel.org/pub/linux/libs/pam>
- ☐ PAM складається з набору спільних бібліотечних модулів, які надають програмам відповідні сервіси автентифікації та аудиту
 - Програми оновлюються, щоб передати їх код автентифікації та аудиту в PAM, що дозволяє системі застосовувати послідовну політику ідентифікації та автентифікації, а також створювати відповідні записи аудиту
 - Наступні програми удосконалені для використання PAM:
 - login
 - passwd
 - su, sudo
 - useradd, usermod, userdel
 - groupadd, groupmod, groupdel
 - sshd
 - chage
 - chfn
 - chsh
 - newrole



Компоненти 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 (як правило) 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 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



Автентифікація з РАМ₁

532

1. Програма із підтримкою РАМ здійснює виклик до РАМ для ініціалізації певних структур даних
 - З ініціалізацією програма, що здійснює виклик, надає РАМ ім'я, яке в кінцевому підсумку використовується для пошуку файлу конфігурації стека автентифікації в `/etc/pam.d/`
 - Зазвичай це ім'я дорівнює імені програми
2. Модуль РАМ знаходить файл конфігурації для цієї програми з `/etc/pam.d/application_name` і отримує список модулів РАМ, необхідних для обслуговування цієї програми
 - Якщо він не може знайти файл конфігурації для певної програми, він використовує `/etc/pam.d/other`
3. Залежно від порядку, зазначеного у файлі конфігурації, РАМ завантажує відповідні модулі для операції РАМ, яку вимагає програма, що викликає
 - РАМ забезпечує один зворотний виклик для кожного типу модуля — тип модуля узгоджується з розділами "auth", "session", "password" та "account" у файлах конфігурації РАМ



Автентифікація з РАМ₂

533

4. Код модуля автентифікації виконує запитувану операцію залежно від типу модуля
 - Модуль може вимагати введення від користувача
 - Модуль може виконувати операції, які навряд чи мають щось спільне з автентифікацією, але дії яких необхідні для налаштування середовища користувача
5. Кожен модуль автентифікації виконує свою дію і передає результат назад програмі
6. Бібліотека РАМ модифікована для створення запису аудиту типу **USER_AUTH**, щоб занотувати успіх або невдачу модуля автентифікації
7. Програма здійснює відповідні дії на основі сукупних результатів усіх модулів автентифікації



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 імені 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_secrcp		
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	



Зміна ідентифікатора користувача - команда su

536

- ☐ Команда **su** призначена для переключення на інший ідентифікатор, який встановлює новий сеанс входу та створює нову оболонку з новим ідентифікатором
- ☐ При виклику **su**, користувач повинен надати облікові дані, пов'язані з цільовою ідентифікацією — тобто, коли користувач хоче перейти на інший ідентифікатор користувача, він повинен надати пароль, що захищає обліковий запис цільового користувача
- ☐ Основне використання команди **su** в Linux полягає у наданні належним чином уповноваженим особам можливості приймати ідентифікатор root для виконання адміністративних дій
- ☐ Зауважте, що коли користувач виконує програму, у якій встановлений біт **suid**, лише ефективний ідентифікатор користувача та ідентифікатор файлової системи змінюються на ідентифікатор власника файлу, що містить програму, тоді як реальний ідентифікатор користувача залишається ідентифікатором того, хто викликає програму
- ☐ Ідентифікатор входу (login ID) не змінюється ні командою **su**, ні виконанням програми, яка має біт **suid** або **sgid**, оскільки він використовується для аудиту



Зміна ідентифікатора користувача — команда `sudo`

- ☐ Команда `sudo` призначена для надання користувачам дозволів на виконання команд з іншим ідентифікатором користувача
- ☐ При виклику `sudo` користувач повинен пройти автентифікацію за допомогою власних облікових даних
- ☐ `sudo` пов'язано зі складним набором правил, який можна задіяти, щоб вказати, який:
 - початковий ідентифікатор користувача
 - що походить з якого хосту
 - може отримати доступ до команди, команди з певними прапорами конфігурації або до всіх команд у каталозі
 - з яким новим ідентифікатором користувача.
- ☐ При перемиканні ідентифікаторів, реальний, файлової системи та ефективний ідентифікатори користувача, а також реальний, файлової системи та ефективний ідентифікатори групи змінюються на ті, що відповідають користувачу, зазначеному в команді (після успішної автентифікації)



Управління даними автентифікації₁

538

- ❑ Кожен екземпляр Linux підтримує власний набір користувачів із їхніми паролями та атрибутами
 - Незважаючи на те, що один і той самий користувач може мати облікові записи на різних серверах, взаємопов'язаних мережею, що працюють під керуванням екземпляру Linux, ці облікові записи та їх параметри не синхронізуються в різних екземплярах Linux
 - Як результат, один і той же користувач може мати різні імена користувачів, різні ідентифікатори користувачів, різні паролі та різні атрибути на різних машинах у мережному середовищі
- ❑ Кожен екземпляр Linux в мережі підтримує власну адміністративну базу даних, вносячи всі адміністративні зміни в локальний екземпляр Linux
 - Файл **/etc/passwd** містить для кожного користувача ім'я користувача, ідентифікатор користувача, індикатор, чи дійсний пароль користувача, ідентифікатор основної групи користувача та іншу (не пов'язану з безпекою) інформацію
- ❑ Файл **/etc/shadow** містить для кожного користувача хеш пароля користувача, ідентифікатор користувача, час останньої зміни пароля, час закінчення терміну дії, а також термін дії пароля та деяку іншу інформацію



Управління даними автентифікації₂

539

- ❑ Користувачам дозволено змінювати свої паролі за допомогою команди **passwd**
 - Ця програма може читати та змінювати вміст **/etc/shadow** для введення пароля користувача, який, як правило, буде недоступним для непривілейованого процесу користувача
 - Користувачів також попереджають про необхідність зміни їх паролів під час входу, якщо термін дії пароля скоро закінчиться, і забороняють входити в систему, якщо термін дії пароля закінчився
- ❑ Час останнього успішного входу реєструється в каталозі **/var/log/tallylog**, де зберігається один файл на користувача
- ❑ TOE відображає користувачам інформативні банери до або під час входу в систему
- ❑ Банери можна визначити у файлах **/etc/issue** для входів через фізичну консоль або **/etc/issue.net** для віддалених входів, наприклад через SSH



Управління даними автентифікації₃

540

- ☐ SSSD — це системний демон з основною функцією забезпечення доступу до віддаленого ресурсу ідентифікації та автентифікації через загальний фреймворк, який може забезпечити кешування та автономну підтримку системи
- ☐ SSSD можна налаштувати на використання власного домену LDAP (тобто постачальника ідентифікатора LDAP з автентифікацією LDAP) або постачальника ідентифікатора LDAP з автентифікацією Kerberos
- ☐ Однією з основних переваг SSSD є автономна автентифікація
- ☐ У системі SSSD користувачеві потрібно керувати лише одним обліковим записом
- ☐ SSSD інтегрується з фреймворками PAM та NSS і тому може використовуватися разом із модулями PAM для локальних сховищ облікових даних



Автентифікація на основі ключів SSH

541

- ☐ На додаток до автентифікації на основі PAM, сервер OpenSSH може виконувати автентифікацію на основі ключів
- ☐ Щоб встановити автентифікацію на основі ключа, користувач спочатку має сформувати пару ключів RSA або ECDSA
- ☐ Коли виконується операція входу, протокол SSHv2 намагається виконати автентифікацію "publickey", використовуючи приватний ключ на стороні клієнта та відкритий ключ, знайдений на стороні сервера
 - Операції, що виконуються під час автентифікації за допомогою відкритих ключів, визначені в главі 7 RFC4252
- ☐ Користувачі повинні захищати частину свого приватного ключа так само, як захищати пароль
 - Необхідні відповідні налаштування дозволу для файлу, що містить приватний ключ
- ☐ Для посилення захисту приватного ключа користувач може зашифрувати ключ там, де пароль служить ключем для операції шифрування



Блокування сеансу

- ☐ Linux використовує програму **screen(1)**, яка блокує поточний сеанс користувача або після встановленого адміністратором часу бездіяльності, або за запитом користувача
- ☐ Щоб розблокувати сеанс, користувач повинен надати свій пароль
- ☐ **screen** використовує PAM для перевірки пароля і дозволяє користувачеві отримати доступ до його сеансу після успішної перевірки



Захист від використання вразливостей коду — ASLR

543

- ☐ 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
- ☐ Технологія ASLR впроваджена у ядро Linux, починаючи з версії 2.6.12 (2005 рік — значно раніше, ніж у Windows)



Основні недоліки традиційної моделі безпеки Linux (і Unix)

544

- ☐ Недостатня вибірковість системи розмежування доступу
- ☐ 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

Донецький національний технічний університет



Тема 8. Засоби захисту Linux.

Лекція 16. Протокол LDAP.

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

546

☐ Розуміння LDAP

- Інформаційна структура
- Найменування
- Функції/Операції
- Безпека

☐ Модель протоколу

☐ Відображення на транспортних послугах

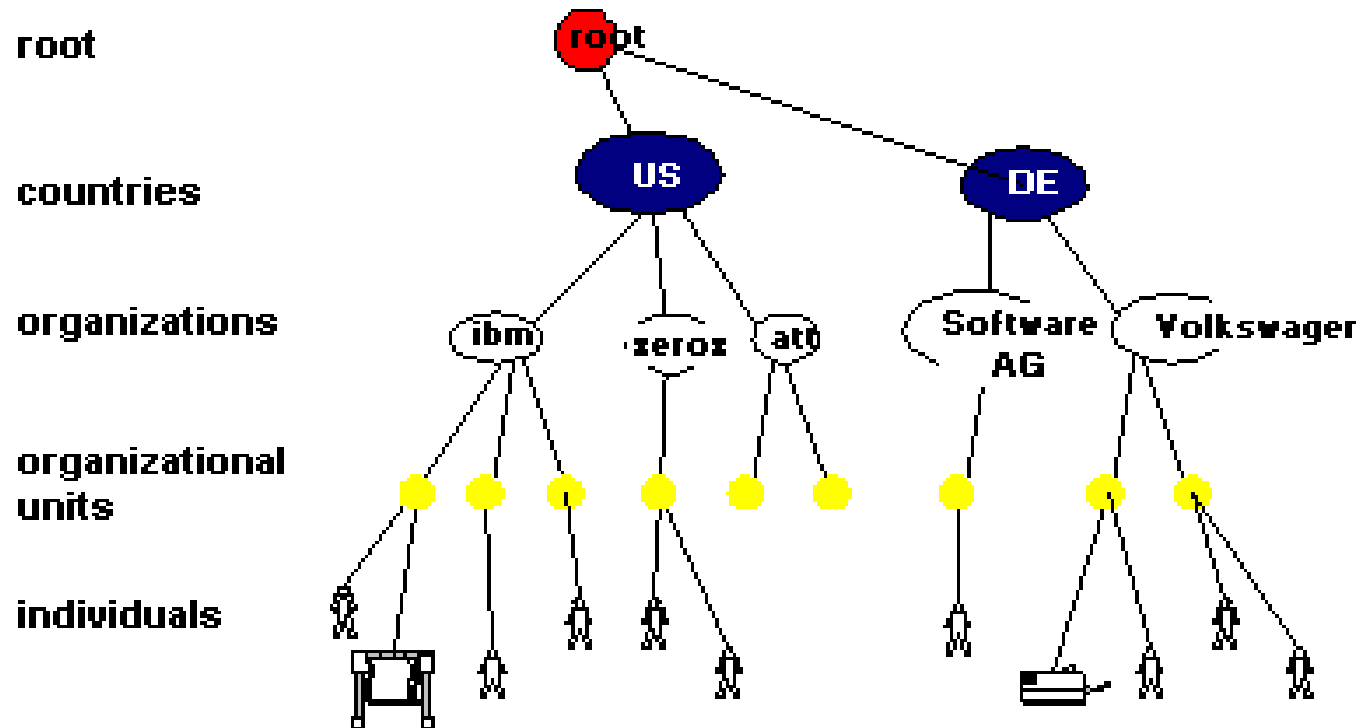
☐ Кодування елементів протоколу



Стандарт X.500₁

547

- ❑ X.500 standard. CCITT 1988
 - ISO 9594 – X.500-X.521 1990





Стандарт X.500₂

548

- ☐ Організовує записи каталогу в ієрархічному просторі імен
- ☐ Потужні можливості пошуку
- ☐ Часто використовується для взаємодії несумісних служб каталогів
- ☐ Використовував DAP для с/s зв'язку
- ☐ Для роботи DAP (App. Layer) потрібен ВСЬ стек OSI
- ☐ Занадто важкий для невеликих середовищ



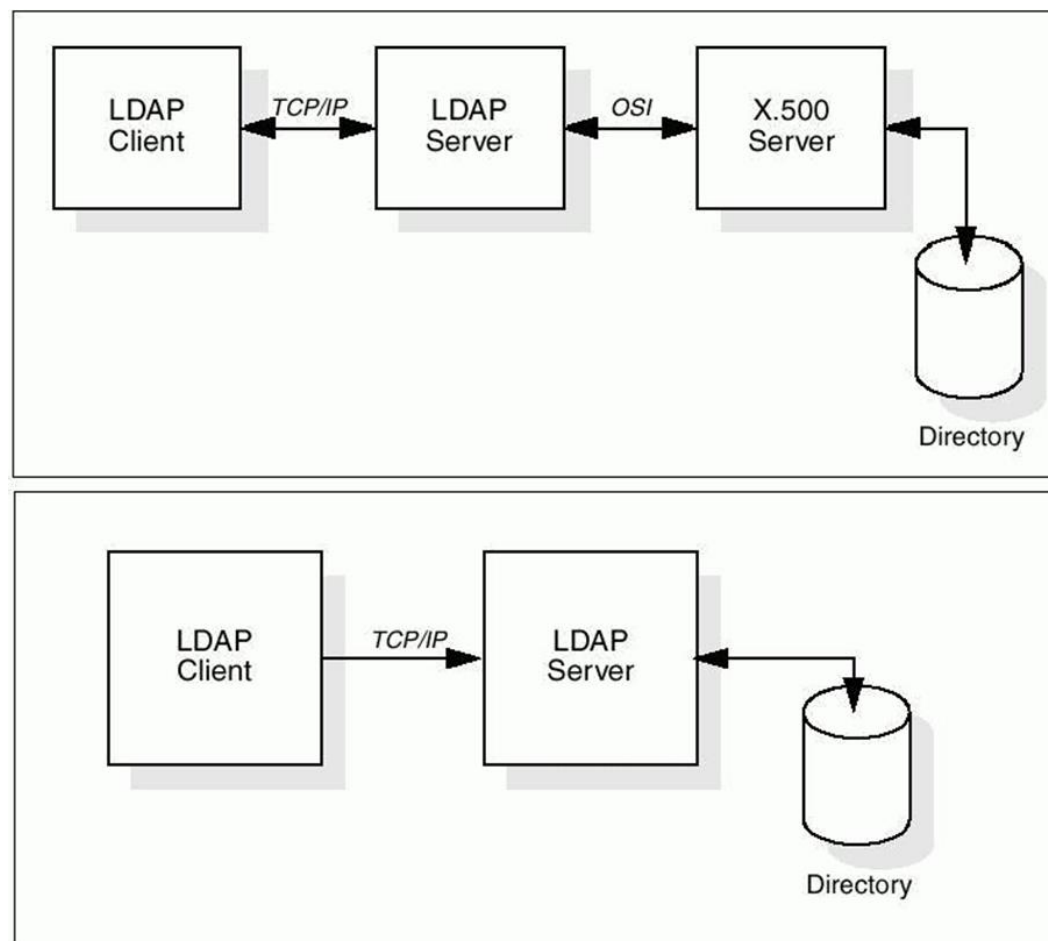
Що таке LDAP?

- ☐ Полегшений протокол доступу до каталогу (**Lightweight Directory Access Protocol**)
- ☐ Використовується для доступу та оновлення інформації в каталозі, побудованому на моделі X.500
- ☐ Специфікація визначає вміст повідомлень між клієнтом і сервером
- ☐ Включає операції для встановлення та відключення сеансу від сервера



LDAP Server: G/S

550





Розуміння LDAP

551

- ☐ Легка альтернатива DAP
- ☐ Використовує TCP/IP замість стеку OSI
- ☐ Спрощує певні функції та пропускає інші
- ☐ Для представлення даних використовує рядки, а не нотацію ASN.1 DAP



LDAP

552

☐ Інформація

- Структура інформації, що зберігається в каталозі LDAP

☐ Найменування

- Як інформація організована та ідентифікована

☐ Функціональний/операції

- Описує, які операції можна виконувати з інформацією, що зберігається в каталозі LDAP

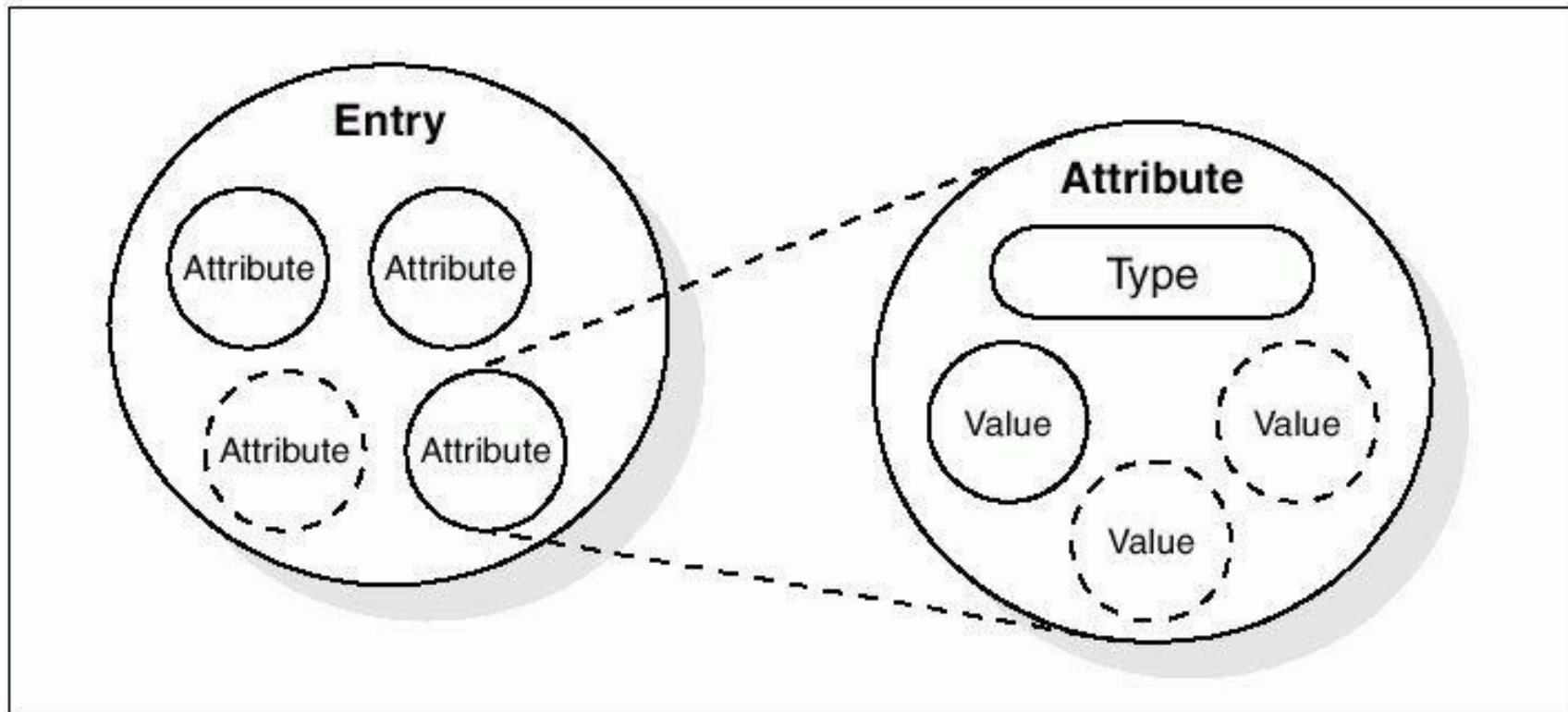
☐ Безпека

- Описує, як можна захистити інформацію від несанкціонованого доступу



Сховище інформації LDAP₁

553





Сховище інформації LDAP₂

- ☐ Кожен атрибут має тип/синтаксис і значення
- ☐ Може визначати поведінку значень під час пошуку/операцій з каталогом
- ☐ Синтаксис: bin, ces, cis, tel, dn тощо
- ☐ Обмеження використання: ssn – тільки один, jpegPhoto – 10K



Сховище інформації LDAP₃

☐ Кожен «запис» описує об'єкт (клас)

- Особа, сервер, принтер тощо

☐ Приклад запису:

- InetOrgPerson(cn, sn, ObjectClass)

☐ Приклади атрибутів:

- cn (cis)
- sn (cis)
- phoneNumber (tel)
- ou (cis)
- owner (dn)
- jpegPhoto (bin)



LDAP Naming₁

556

- ☐ DN складаються з послідовності відносного DN
 - cn=argussss,ou=OraDiv,o=ArCorp c=UA
(Leaf 2 Root) (~використовуйте \ для спеціального)
- ☐ Інформаційне дерево довідників (DIT)
- ☐ Дотримуйтеся географічної чи організаційної схеми
- ☐ Псевдоніми: Tree-like
- ☐ Псевдоніми можуть пов'язувати вузли, які не є листовими



LDAP Naming₂

557

- ☐ Реферали: може не зберігати весь DIT (v3)
- ☐ Реферали
 - objectClass=реферал, атрибут=посилання, значення=LDAPurl
- ☐ Реалізація відрізняється
 - Реферали/ланцюжки (постачальник)
 - RFC 1777: очікується з'єднання серверів.



LDAP Naming₃

558

☐ Схема

- Визначає дозволені класи об'єктів
- Де вони зберігаються
- Які вони мають атрибути (objectClass)
- Які атрибути є необов'язковими (objectClass)
- Тип/синтаксис кожного атрибута (objectClass)

☐ Сервер запитів для інформації: DN нульової довжини

☐ Схема LDAP повинна бути читабельною



Приклади LDAP Naming

Тип атрибута	Строка
CommonName	CN
LocalityName	L
StateorProvinceName	ST
OrganizationName	O
OrganizationalUnitName	OU
CountryName	C
StreetAddress	STREET
domainComponent	DC
Userid	UID



Функції/операції LDAP

560

☐ Аутентифікація

- BIND/UNBIND
- ABANDON

☐ Запит

- Пошук
- Порівняння записів

☐ Оновлення

- Додавання записів
- Видалення записів (лише листові вузли, без псевдонімів)
- Зміна записів
- Зміна DN/RDN



Взаємодія клієнта та сервера

☐ Клієнт встановлює сеанс із сервером (BIND)

- Ім'я хосту/IP та номер порту
- Безпека
 - Автентифікація на основі ідентифікатора користувача/паролю
 - Анонімне підключення - права доступу за замовчуванням
 - Також підтримується шифрування/Kerberos

☐ Клієнт виконує операції

- Read/Update/Search
- SELECT X,Y,Z FROM PART_OF_DIRECTORY

☐ Клієнт завершує сесію (UNBIND)

☐ Клієнт може ABANDON сесію



BIND/UNBIND/ABANDON

- ☐ Запит містить версію LDAP, ім'я, яке клієнт хоче прив'язати, тип автентифікації
 - Простий (паролі з відкритим текстом, анонімний)
 - Kerberos v4 до сервера LDAP (krbv42LDAP)
 - Kerberos v4 до сервера DSA (krbv42DSA)
- ☐ Сервер відповідає індикацією статусу
- ☐ UNBIND: завершує сеанс протоколу
 - UnbindRequest ::= [APPLICATION 2] NULL
- ☐ ABANDON



Пошук/порівняння

☐ Запит включає:

- baseObject: LDAPDN
- Область: скільки рівнів для пошуку
- derefAliases: обробка псевдонімів
- sizeLimit: максимальна кількість повернутих записів
- timeLimit: максимальний час, дозволений для пошуку
- attrOnly: повертає типи атрибутів АБО також значення
- Filter: cond. які потрібно виконати під час пошуку
- Атрибути: список атрибутів запису, які потрібно повернути

☐ Читання та список реалізовано як пошук

☐ Порівняння: схоже на пошук, але повертає T/F



ADD/MODIFY/DELETE

☐ Запит ADD

- Запис: LDAPDN
- Список атрибутів та значень (або наборів значень)

☐ Запит MODIFY

- Використовується для додавання, видалення, модифікації атрибутів
- Запит включає:
 - Об'єкт: LDAPDN
 - Список модифікацій
 - ✓ Додавання, видалення, заміна

☐ Запит DELETE

- Об'єкт: LDAPDN

☐ Модифікація RDN: LDAPDN, newRDN



Елементи протоколу₁

□ LDAPMessage (MessageID unique)

```
LDAPMessage ::=
  SEQUENCE {
    messageID      MessageID,
    protocolOp     CHOICE {
      bindRequest      BindRequest,
      bindResponse     BindResponse,
      unbindRequest    UnbindRequest,
      searchRequest     SearchRequest,
      searchResponse    SearchResponse,
      modifyRequest     ModifyRequest,
      modifyResponse    ModifyResponse,
      addRequest        AddRequest,
      addResponse       AddResponse,
      delRequest        DelRequest,
      delResponse       DelResponse,
      modifyRDNRequest  ModifyRDNRequest,
      modifyRDNResponse ModifyRDNResponse,
      compareDNRequest  CompareRequest,
      compareDNResponse CompareResponse,
      abandonRequest    AbandonRequest
    }
  }

MessageID ::= INTEGER (0 .. maxInt)
```



Елементи протоколу₂

LDAPString ::= OCTET STRING

LDAPDN ::= LDAPString

RelativeLDAPDN ::= LDAPString

AttributeValueAssertion ::= Sequence {
 attributeType attributeValue,
 attributeValue attributeValue
}

attributeType ::= LDAPString

attributeValue ::= OCTET STRING



Елементи протоколу₃

❑ Результат LDAP

❑ Помилки

▪ Надсилається скорочена послідовність DIT RDN

- noSuchObject
- aliasProblem
- invalidDNSyntax
- isLeaf
- інші

```
LDAPResult ::=
  SEQUENCE {
    resultCode      ENUMERATED {
      success              (0),
      operationsError      (1),
      protocolError        (2),
      timeLimitExceeded    (3),
      sizeLimitExceeded    (4),
      compareFalse         (5),
      compareTrue          (6),
      authMethodNotSupported (7),
      strongAuthRequired   (8),
      noSuchAttribute      (16),
      undefinedAttributeType (17),
      inappropriateMatching (18),
      constraintViolation   (19),
      attributeOrValueExists (20),
      invalidAttributeSyntax (21),
      noSuchObject         (32),
      aliasProblem          (33),
      invalidDNSyntax      (34),
      isLeaf                (35),
      aliasDereferencingProblem (36),
      inappropriateAuthentication (48),
      invalidCredentials    (49),
      insufficientAccessRights (50),
      busy                  (51),
      unavailable           (52),
      unwillingToPerform    (53),
      loopDetect            (54),
      namingViolation       (64),
      objectClassViolation  (65),
      notAllowedOnNonLeaf   (66),
      notAllowedOnRDN       (67),
      entryAlreadyExists    (68),
      objectClassModsProhibited (69),
      other                  (80)
    },
    matchedDN      LDAPDN,
    errorMessage   LDAPString
  }
```



Безпека LDAP₁

- ☐ Підтримується поточна версія LDAP
 - Чисті текстові паролі
 - Аутентифікація KERBEROS версії 4
- ☐ Інші методи автентифікації можливі в майбутніх версіях (березень 1995 р.)
- ☐ У версії 3 додано підтримку SASL
 - Kerberos вважається сильнішим за SASL



Безпека LDAP₂

569

- ☐ Безпека на основі моделі BIND
- ☐ Чистий текст - версія 1
- ☐ Kerberos - версії 1,2,3 (depr)
- ☐ SASL - версія 3
 - проста автентифікація та рівень безпеки
 - використовує один із багатьох методів автентифікації
- ☐ Пропозиція щодо безпеки транспортного рівня
 - На основі SSL v3 від Netscape



Безпека LDAP₃

570

☐ Без аутентифікації

☐ Проста аутентифікація

- Надаються DN і пароль
- Чистий текст або кодування Base 64

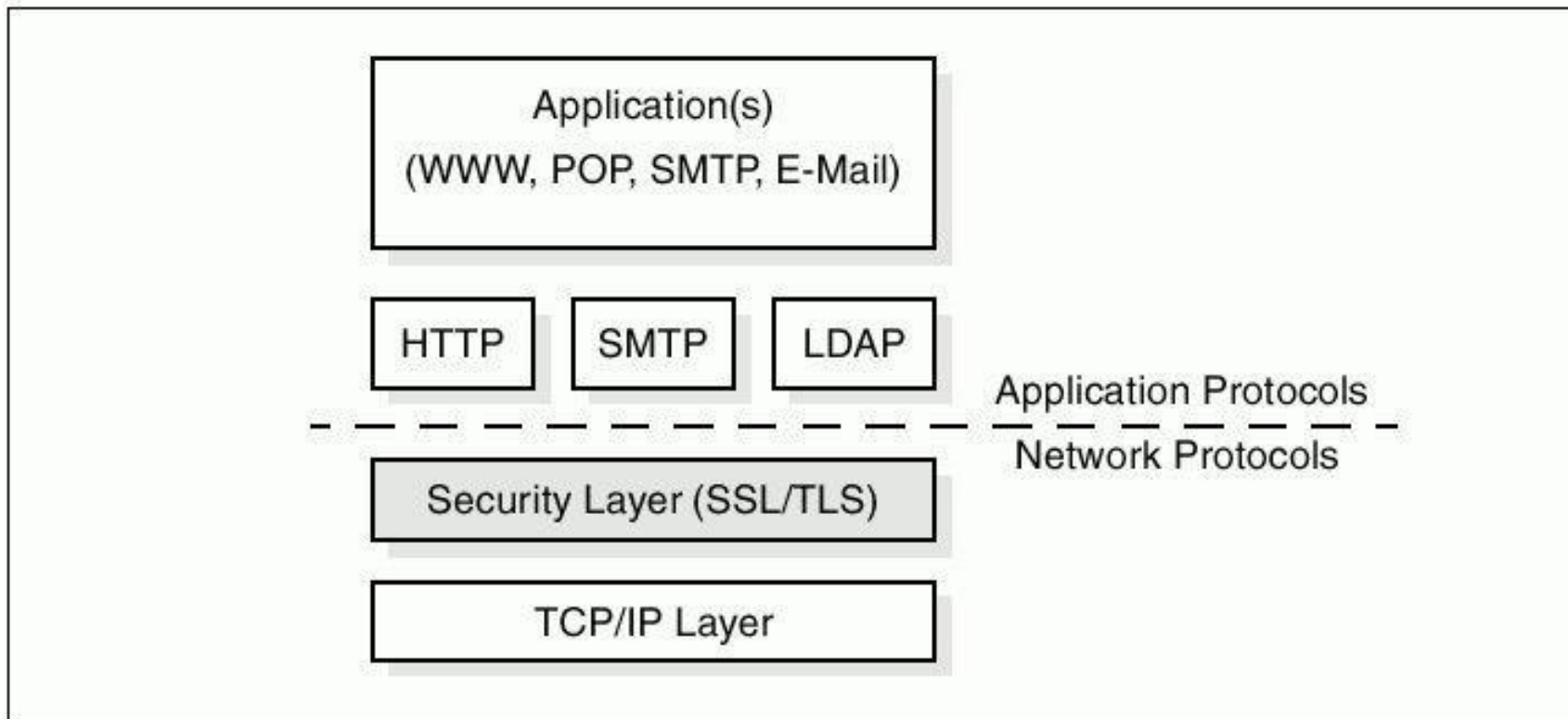
☐ SASL (RFC 2222)

- Параметри: DN, механізм, облікові дані
- Забезпечує міжпротокольні автентифікаційні виклики
- Шифрування можна узгодити за бажанням
- `ldap_sasl_bind()` (виклик версії 3)
- `Ldap://<ldap_server>/?supportedsaslm mechanisms`



Безпека LDAP₄

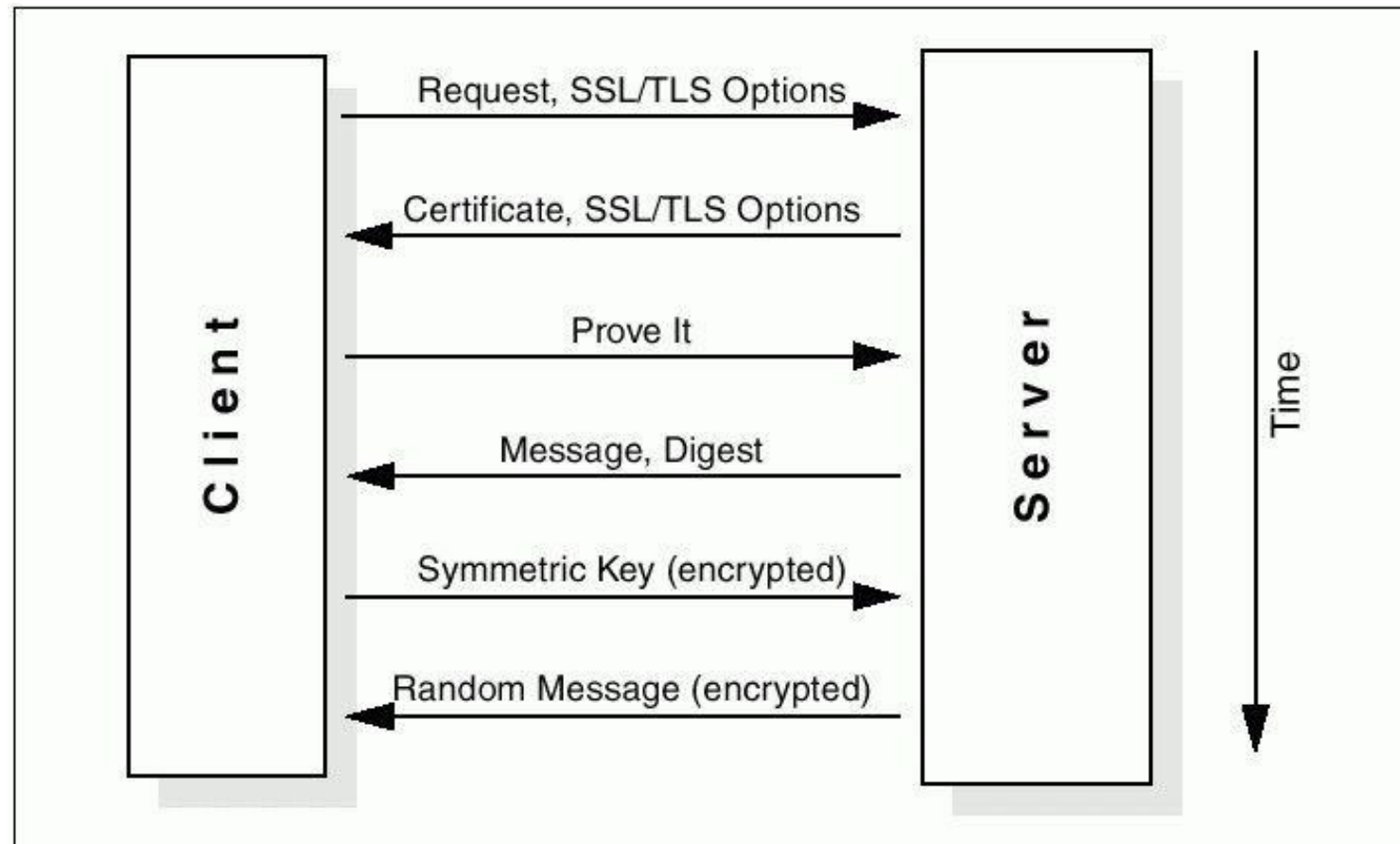
□ LDAP з використанням SASL та SSL/TLS





Безпека LDAP5

□ SSL/TLS Handshake





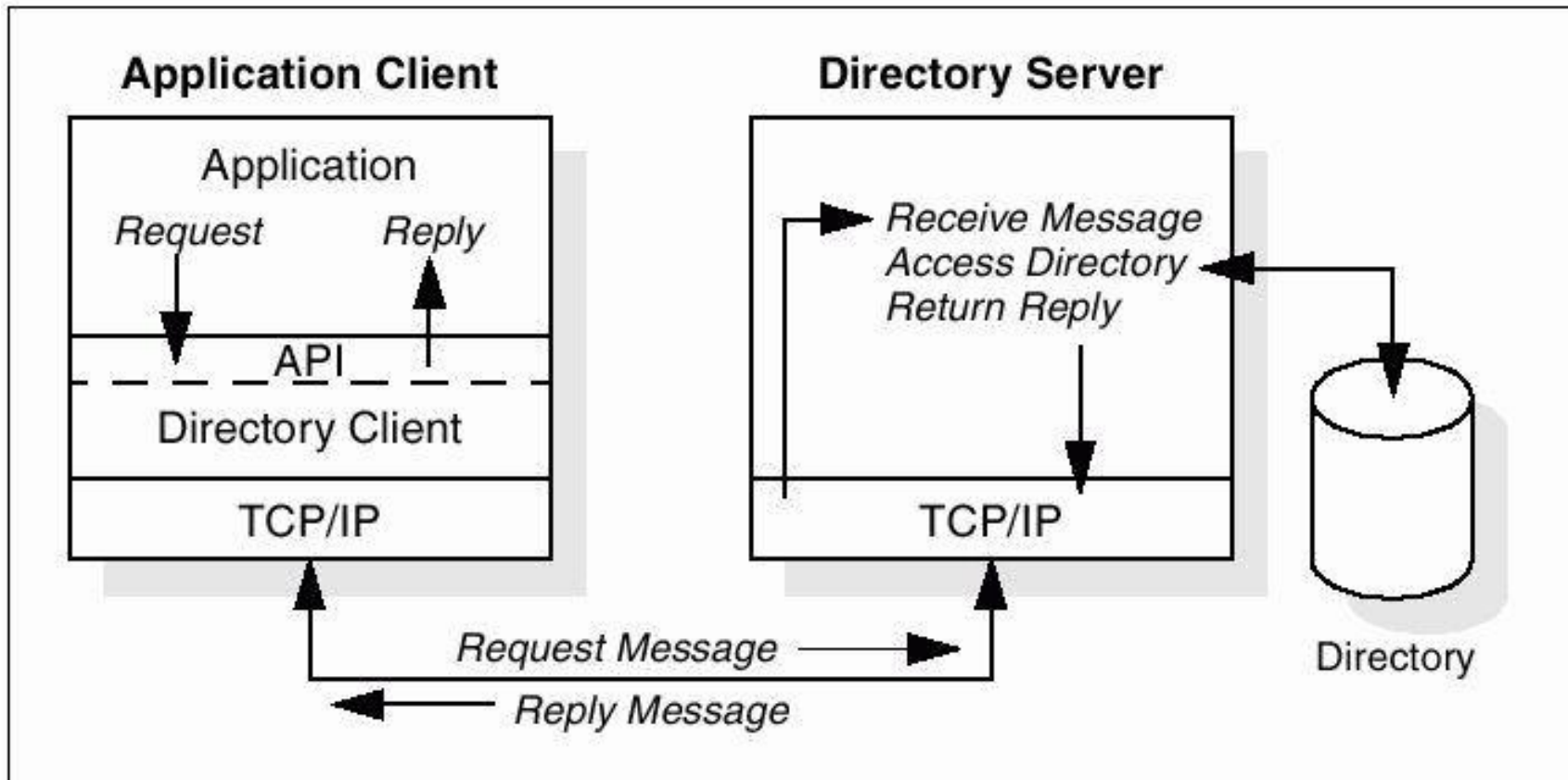
Модель протоколу

573

- ☐ Клієнти, що виконують операції протоколу на серверах
 - Клієнт надсилає запит протоколу на сервер
 - Сервер виконує операції з каталогом
 - Сервер повертає відповідь (результати/помилки)
- ☐ Поведінка асинхронного сервера



Взаємодія клієнт/сервер з каталогу





Відображення на транспорт

- ☐ Використовує надійний транспорт, орієнтований на підключення
- ☐ TCP
 - LDAPMessage PDU, зіставлений на потік байтів TCP
 - Слухач LDAP на порту 389
- ☐ Connection Oriented Transport Service (COTS)
 - LDAP PDU відображається безпосередньо на T-Data



Кодування елементів протоколу

- ☐ Закодовано для Exchange за допомогою BER (основні правила кодування)
- ☐ BER визначено в абстрактній синтаксичній нотації один (ASN.1)
- ☐ Високі накладні витрати для BER
 - Обмеження, введені для підвищення продуктивності
 - Лише певна форма кодування довжини
 - Бітові рядки/октетні рядки та всі типи рядків символів, закодовані лише в примітивній формі



Імплементації LDAP

- ☐ C Library API
 - LDAPv2 - RFC 1823 'The LDAP API'
 - LDAPv3
- ☐ Java JNDI
- ☐ LDAP v3
- ☐ шлюз HTTP to LDAP
- ☐ LDAP to X.500 шлюз – ldapd



OpenLDAP

☐ Встановлення

- `apt install openldap-server`

☐ slap.conf

- Порожні рядки та рядки, що починаються зі знака фунта (#), ігноруються
- Параметри та пов'язані з ними значення розділені пробілами
- Продовженням попереднього вважається рядок із пробілом у першому стовпчику



slap.conf

```
include                /usr/local/etc/openldap/schema/core.schema
pidfile                /var/run/openldap/slapd.pid
argsfile               /var/run/openldap/slapd.args
loglevel               256
modulepath             /usr/local/libexec/openldap
moduleload             back_mdb
moduleload             back_ldap
# ACL rules here for global
database              mdb
maxsize                1073741824
suffix                 "dc=nap,dc=nctucs,dc=net"
rootdn                 "cn=Manager,dc=nap,dc=nctucs,dc=net"
roopw                  <generated by slappasswd>
directory              /var/db/openldap-data

# Indices to maintain
index                  objectClass eq
# ACL rules here for specify database
```



Directory ACL₁

```
access to dn.exact="cn=Manager,dc=nap,dc=nctucs,dc=net"  
    by peername.ip="127.0.0.1" auth  
    by users none  
    by anonymous none  
    by * none  
  
access to attrs=userPassword  
    by self write  
    by anonymous auth  
    by dn.base="cn=Manager,dc=nap,dc=nctucs,dc=net" write  
    by * none  
  
access to attrs=englishname,birthdate  
    by self write  
    by users read  
    by anonymous read
```



Directory ACL₂

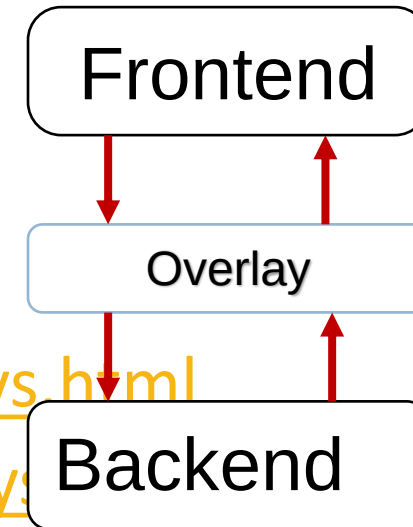
Level	Privileges	Description
none =	0	no access
disclose =	d	needed for information disclosure on error
auth =	dx	needed to authenticate (bind)
compare =	cdx	needed to compare
search =	scdx	needed to apply search filters
read =	rscdx	needed to read search results
write =	wrscdx	needed to modify/rename
manage =	mwrscdx	needed to manage

<http://www.openldap.org/doc/admin24/access-control.html>



Overlay

- ❑ Компоненти програмного забезпечення, які забезпечують підключення до функцій, аналогічних тим, що надаються серверними програмами, які можна помістити поверх внутрішніх викликів і як зворотні виклики поверх відповідей серверної частини, щоб змінити їх поведінку.
- ❑ Frontend
 - забезпечує доступ до мережі та обробку протоколів
- ❑ Backend
 - суто стосується зберігання даних



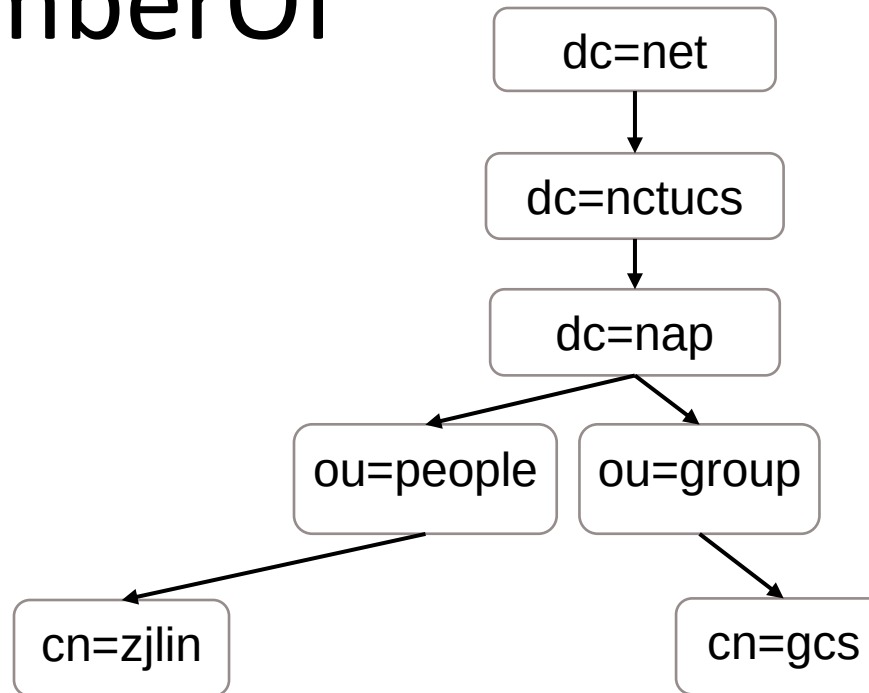
<https://www.openldap.org/doc/admin24/overlays.html>

<https://en.wikipedia.org/wiki/OpenLDAP#Overlay>



Overlay - memberOf

❑ Членство



objectClass: posixGroup
objectClass: top
objectClass: posixAccount
cn: zjlin
gidNumber: 1120

objectClass: posixGroup
objectClass: top
cn: gcs
displayName: gcs
description: Domain Unix group
gidNumber: 1120



Overlay - memberOf

❑ slapd.conf

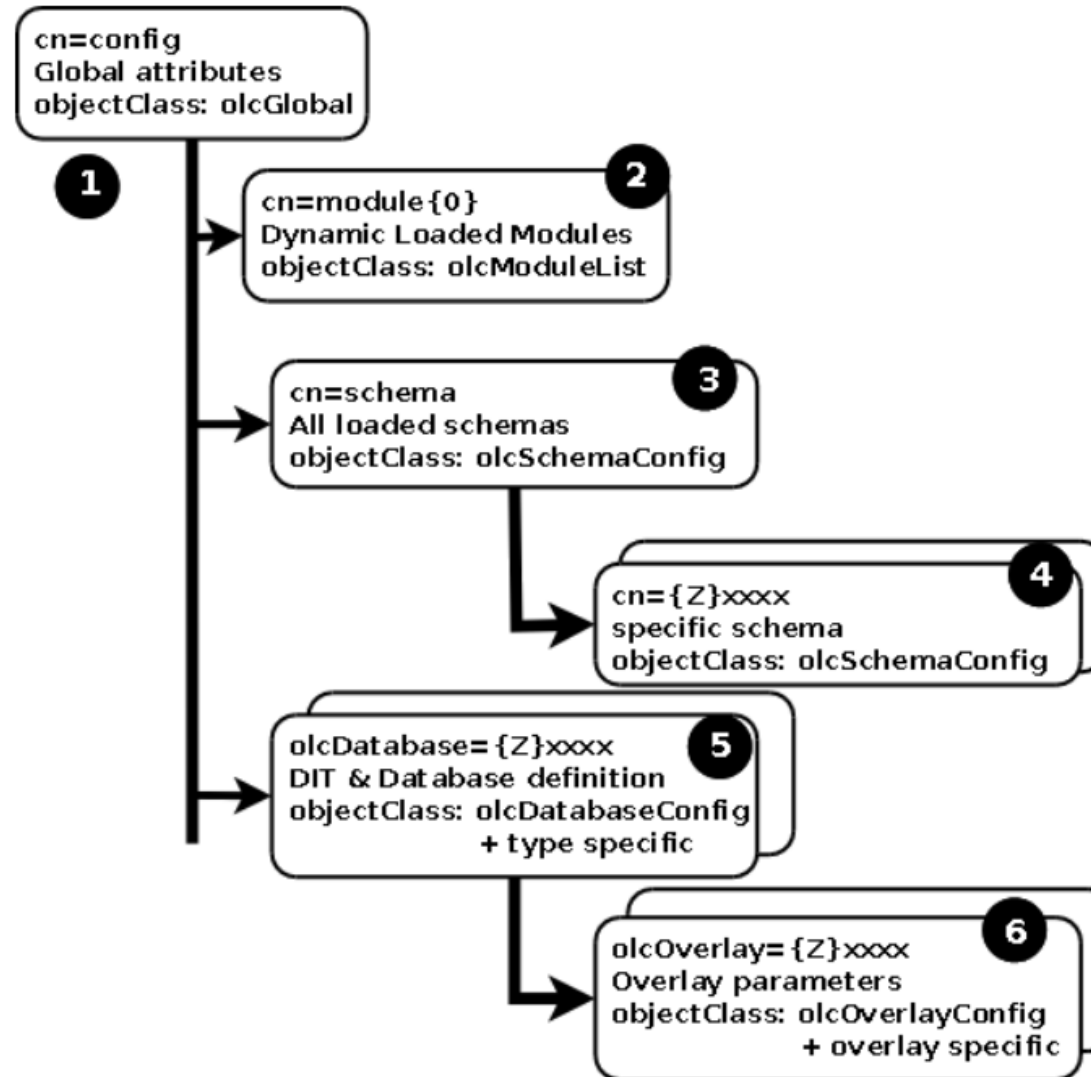
```
166 # MemberOf
167 overlay memberof
```

```
dn: cn=testgroup,ou=OraDiv,dc=nap,dc=argussss,dc=com
objectclass: groupOfNames
cn: testgroup
member: cn=zjlin,ou=People,dc=nap,dc=argussss,dc=com
```

<https://www.openldap.org/doc/admin24/overlays.html>



OLC - on-line configuration





OLC - on-line configuration

```
# {1}mdb, config
dn: olcDatabase={1}mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {1}mdb
olcDbDirectory: /var/db/openldap-data/cs
olcSuffix: dc=cs,dc=nctu,dc=argussss,dc=com
olcAddContentAcl: FALSE
olcLastMod: TRUE
olcMaxDerefDepth: 15
olcReadOnly: FALSE
olcRootDN: cn=Manager,dc=nap,dc=argussss,dc=com
olcRootPW: password
```



Slapd tools

☐ slapcat

- Цей інструмент читає записи з бази даних slapd і записує їх у файл або стандартний вихід

☐ slapadd

- Цей інструмент читає записи LDIF із файлу або стандартного введення та записує нові записи до бази даних slapd

☐ slapindex

- Цей інструмент регенерує індекси в базі даних slapd

☐ slappasswd

- Цей інструмент генерує хеш пароля, придатний для використання як Lq у slapd.conf



LDAP tools

☐ Idapsearch

- Цей інструмент надсилає пошукові запити LDAP до серверів каталогів

☐ Idapadd, Idapmodify

- Ці інструменти надсилають оновлення на сервери каталогів

☐ Idapcompare

- Цей інструмент просить сервер каталогів порівняти два значення

☐ Idapdelete

- Цей інструмент видаляє записи з каталогу LDAP



Ldapsearch

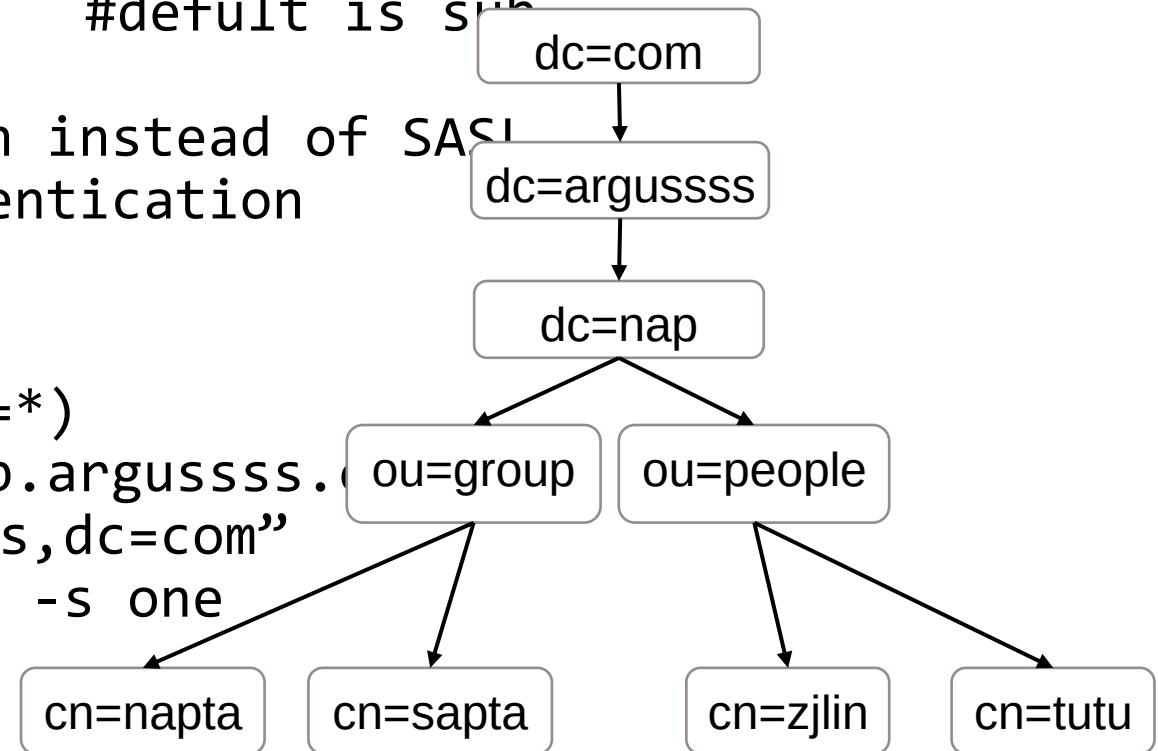
Options

- -b searchbase
- -s {base|one|sub|children} #default is sub
- -D binddn
- -x #Use simple authentication instead of SASL
- -W #password for simple authentication
- -H ldapuri

Ldapsearch [options] filter

- default filter, (objectClass=*)
- ldapsearch -H ldap://ldap.nap.argussss.com
- D "cn=zjlin,dc=nap,dc=argussss,dc=com"
- b "dc=nap,dc=argussss,dc=com" -s one

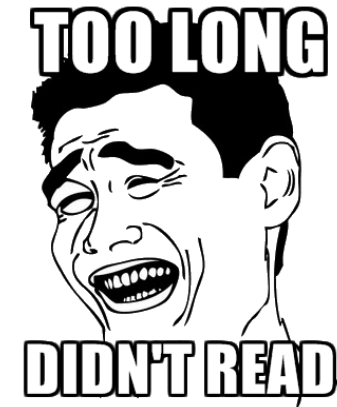
man ldapsearch





ldap.conf

```
❑ ldapsearch -H ldap://ldap.nap.argussss.com  
-b "dc=nap,dc=argussss,dc=com" cn=zjlin
```



```
❑ Edit /usr/local/etc/openldap/ldap.conf
```

```
# See ldap.conf(5) for details  
# This file should be world readable but not world writable.  
BASE      dc=nap,dc=argussss,dc=com  
URI        ldaps://ldap.nap.argussss.com
```

=> ldapsearch -x "cn=zjlin"



Idapsearch – база пошуку vs filter

☐ Пошук за dn

```
# ldapsearch dn='cn=zjlin,dc=nap,dc=argussss,dc=com'
```

- Не працює!

```
# ldapsearch -b 'cn=zjlin,dc=nap,dc=argussss,dc=com' -s base
```

- Працює!

☐ Чому?

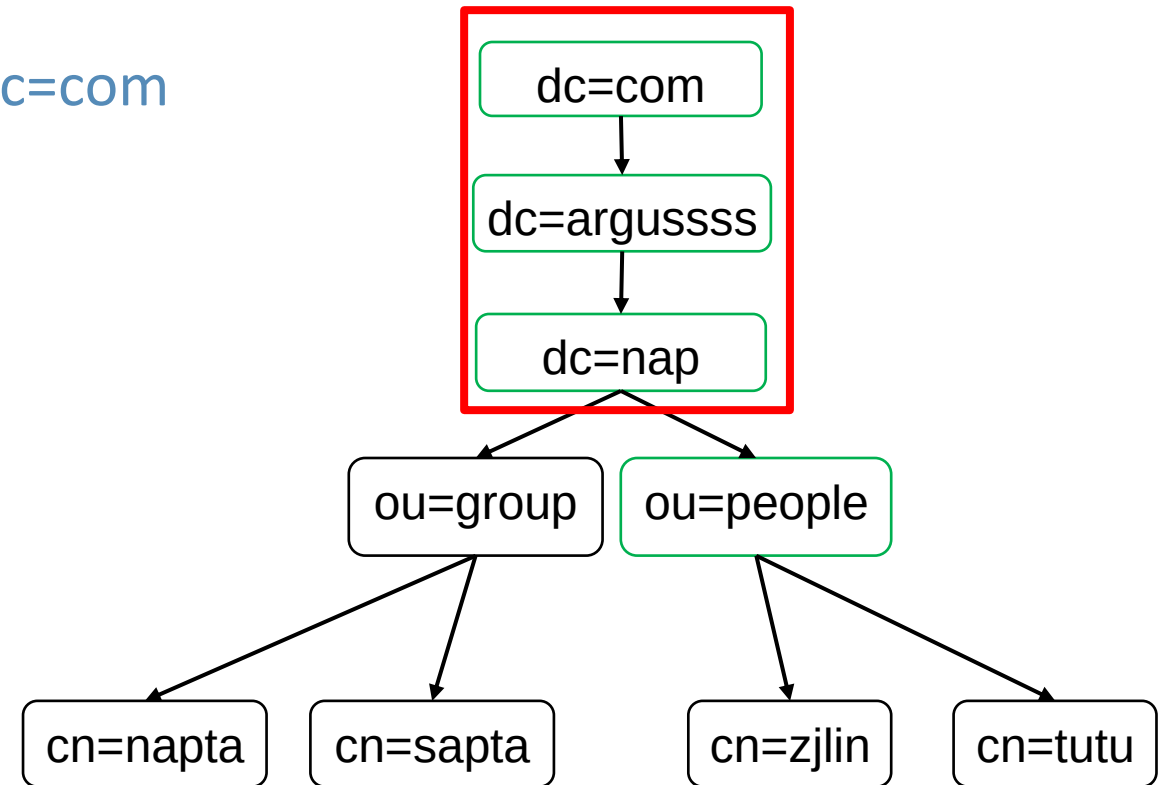
- Ви отримали повний dn, не потрібно шукати.



Idapsearch – база пошуку vs filter

□ База пошуку

- `dc=nap,dc=argussss,dc=com`
- `ou=People, dc=nap,dc=argussss,dc=com`

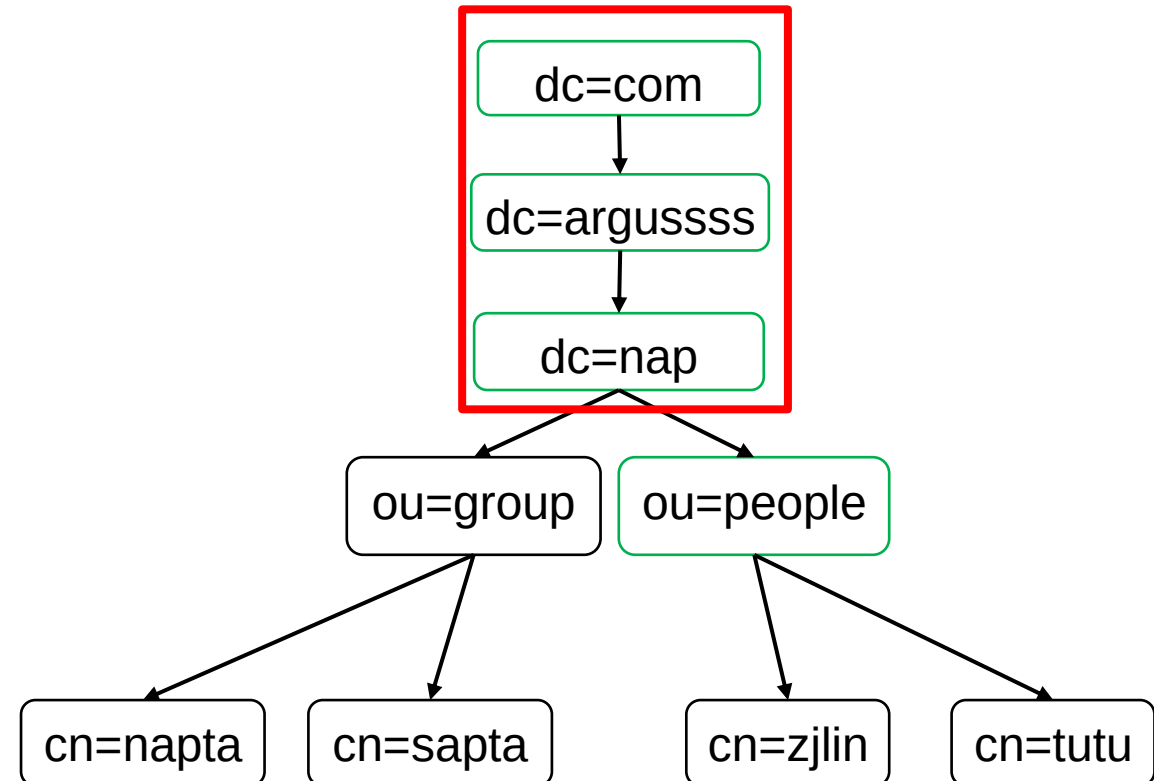




Idapsearch – база пошуку vs filter

❑ filter – пошук за фільтром в базі пошуку

- **cn=napta**
- cn=napta -> can't find





LDAP аутентифікація₁

- ☐ Встановити nss-pam-ldapd
- ☐ Відредагувати /usr/local/etc/nslcd.conf
- ☐ Відредагувати /etc/nsswitch.conf
- ☐ Відредагувати /etc/pam.d/system



LDAP аутентифікація₂

❑ Відредагувати `/usr/local/etc/nslcd.conf`

```
# The user and group nslcd should run as.  
uid nslcd  
gid nslcd  
uri ldap://ldap.nap.nctucs.net  
base dc=nap,dc=argussss,dc=com
```



LDAP аутентифікація₃

❑ Відредагувати /etc/nsswitch.conf

<https://www.freebsd.org/doc/en/articles/ldap-auth/client.html>

```
# nsswitch.conf(5) - name service switch configuration file
# $FreeBSD: releng/11.1/etc/nsswitch.conf
group: files ldap
passwd: files ldap
```

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux.

Лекція 17. Основи SELinux

Предмет: Безпека та захист ОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

598

- ☐ Основи SELinux
- ☐ Основні концепції SELinux
- ☐ Ролі
- ☐ Домени
- ☐ Інструменти
- ☐ Налаштування SELinux на Ubuntu



Що таке SELinux?

- ☐ SELinux - це розширена система безпеки для Linux
- ☐ SELinux заснована на концепції ролей і доменів безпеки
- ☐ SELinux призначена для підвищення безпеки Linux-систем



Історія SELinux

- ☐ SELinux був розроблений Агентством національної безпеки США (NSA) у 1999 році
- ☐ Розробка SELinux була спрямована на підвищення безпеки Linux-систем
- ☐ SELinux був вперше впроваджений у ядро Linux 2.6.0 у 2003 році. З тих пір SELinux став стандартною функцією ядра Linux
- ☐ SELinux продовжує розвиватися та розширюватися. У 2013 році було випущено третє покоління SELinux (SELinux 3.0), яке включає ряд нових функцій і поліпшень
- ☐ SELinux широко використовується в корпоративних і урядових системах.
- ☐ SELinux є одним з найпотужніших інструментів для підвищення безпеки Linux-систем.



Концепції SELinux

- ❑ Роль – це сукупність дозволів і обмежень, які застосовуються до об'єктів
- ❑ Домен безпеки – це сукупність ролей, які застосовуються до об'єктів
- ❑ Об'єкт – це будь-який ресурс в системі, наприклад, файл, каталог, процес або мережевий інтерфейс



Роль

□ Роль – це сукупність дозволів і обмежень, які застосовуються до об'єкта.

- Наприклад, роль `user_u` дозволяє користувачам виконувати звичайні операції, такі як запуск програм і доступ до файлів.
- Роль `httpd_r` дозволяє веб-серверу доступ до файлів веб-сайту.



Основні ролі SELinux

Роль	Опис
<code>unconfined_u</code>	Дозволяє будь-які дії.
<code>system_u</code>	Пов'язана з системними процесами.
<code>user_u</code>	Пов'язана з користувачами.
<code>admin_u</code>	Пов'язана з адміністраторами.
<code>daemon_u</code>	Пов'язана з системними службами.
<code>mail_smtpd_r</code>	Пов'язана з поштовим сервером SMTP.
<code>httpd_r</code>	Пов'язана з веб-сервером Apache.



Домен безпеки

- ❑ Домен безпеки – це сукупність ролей, які можуть бути застосовані до об'єкта
- ❑ Домен безпеки визначає, які дії об'єкт може виконувати
- ❑ Приклад:
 - Файл `/etc/passwd` має роль `system_u:object_r:passwd_file_t`.
 - Це означає, що файл може бути прочитаний тільки об'єктами з роллю `system_u`, які мають дозвіл на читання файлів.



Які є типи доменів безпеки?

- Існує кілька типів доменів безпеки, наприклад:
- Домени користувачів
 - Домени процесів
 - Домени мережевих інтерфейсів



Що таке об'єкт SELinux?

- ❑ Об'єкт – це будь-який елемент системи, який може бути підданий контролю доступу SELinux
- ❑ До об'єктів SELinux відносяться:
 - Процеси
 - Файлові системи
 - Файлові об'єкти
 - Клавіатура
 - Миша
 - Диски
 - Мережеві інтерфейси
 - Сервіси



Контекст безпеки

- ❑ Кожному об'єкту SELinux присвоюється контекст безпеки, який визначає роль і домен безпеки об'єкта.
- ❑ Контекст безпеки складається з двох частин:
 - Роль
 - Домен
- ❑ Приклад:
 - Процес веб-сервера Apache має контекст безпеки `httpd_r:system_u:object_r:httpd_sys_content_t`.
 - Це означає, що процес може виконувати будь-які дії, дозволені ролі `httpd_r`, і що до процесу можуть бути застосовані тільки ролі, пов'язані з доменом `system_u`.



Контексти безпеки і управління доступом

- ☐ Контексти безпеки використовуються SELinux для управління доступом до об'єктів.
- ☐ Коли об'єкт намагається виконати дію, SELinux перевіряє контекст безпеки об'єкта і контекст безпеки об'єкта, до якого він намагається отримати доступ.
- ☐ Якщо контексти безпеки дозволяють дію, то вона виконується.
- ☐ Якщо контексти безпеки забороняють дію, то вона блокується.



Контексти безпеки і політика безпеки

- ☐ Контексти безпеки визначаються політикою безпеки SELinux.
- ☐ Політика безпеки – це набір правил, які визначають, які дії можуть виконувати об'єкти SELinux.
- ☐ Адміністратори системи можуть налаштовувати політику безпеки SELinux, щоб забезпечити необхідний рівень безпеки.



Правила SELinux

- ❑ Правила SELinux визначають, які дії дозволені для процесів у системі, та керують взаємодією між цими процесами та об'єктами (файли, мережа, пристрої тощо). Вони є основою політики безпеки SELinux і забезпечують контроль доступу.
- ❑ Типи правил:
 - Правила дозволу (allow)
 - Правила заборони (deny)
 - Правила обов'язку (must-pass)



Структура правила

□ Правила SELinux мають певну структуру, яка визначає суб'єкт, об'єкт, дію та умови, за яких правило застосовується. Типова структура:

allow <subjekt> <object> <class>:<permission> [<condition>];

- **<subjekt>**: визначає процес або групу процесів з певним контекстом безпеки
- **<object>**: визначає тип об'єкта, на який поширюється правило (файл, каталог, мережевий порт тощо)
- **<class>**: визначає клас об'єкта, який додатково уточнює тип об'єкта
- **<permission>**: визначає конкретну дію, яку дозволяє або забороняє правило
- **<condition>**: додаткові умови, за яких застосовується правило



Приклади правил

- ❑ `allow user_u file_t:home_dir read;`: дозволяє процесам у ролі `user_u` читати файли в своєму домашньому каталозі
- ❑ `deny httpd_r net_t:tcp_port write;`: забороняє процесам веб-сервера Apache (`httpd_r`) відкривати з'єднання TCP на будь-якому порту
- ❑ `must-pass user_u file_t:appraisal { appraisal = 'trusted'; };`: вимагає, щоб файл мав атрибут '`appraisal`' зі значенням '`trusted`', щоб процес у ролі `user_u` міг його виконувати



Правила перенаправлень SELinux

- ❑ Правила перенаправлень SELinux використовуються для перенаправлення дій з одного суб'єкта на інший. Вони можуть використовуватися для забезпечення додаткового рівня безпеки або для спрощення політики безпеки.
- ❑ Типи правил перенаправлень:
 - **Правила перенаправлення дозволу (allow-to):** Ці правила дозволяють дії, яка зазвичай заборонена, якщо вона виконується суб'єктом з певним контекстом безпеки. Наприклад, правило може дозволити процесу в ролі **user_u** відкрити файл, який зазвичай заборонено відкривати користувачам.
 - **Правила перенаправлення заборони (deny-to):** Ці правила забороняють дії, яка зазвичай дозволена, якщо вона виконується суб'єктом з певним контекстом безпеки. Наприклад, правило може заборонити процесу в ролі **httpd_r** читати файли в домашньому каталозі, які зазвичай дозволено читати.



Структура правил перенаправлень

□ Правила перенаправлень SELinux мають певну структуру, яка визначає суб'єкт, об'єкт, дію, до якого суб'єкта перенаправляється, та умови, за яких правило застосовується. Типова структура:

`allow <subject> <object> <class>:<permission> to <subject>:<domain> [ <condition> ];`

- `<subject>`: визначає процес або групу процесів з певним контекстом безпеки, для яких правило застосовується
- `<object>`: визначає тип об'єкта, на який поширюється правило (файл, каталог, мережевий порт тощо)
- `<class>`: визначає клас об'єкта, який додатково уточнює тип об'єкта
- `<permission>`: визначає конкретну дію, яку дозволяє або забороняє правило
- `<subject>:<domain>`: визначає суб'єкт, на який перенаправляється дія
- `<condition>`: додаткові умови, за яких застосовується правило



Приклади правил перенаправлень

- ❑ `allow httpd_r file_t:file { open; read; };`: дозволяє процесам веб-сервера Apache (`httpd_r`) відкривати і читати файли
- ❑ `deny-to user_u file_t:home_dir { read; };`: забороняє процесам у ролі `user_u` читати файли в своєму домашньому каталозі
- ❑ `allow user_u file_t:file { read; } to system_u:systemd_r:sd_daemon_t;`: дозволяє процесам у ролі `user_u` читати файли, які зазвичай заборонені, якщо вони виконуються процесом `systemd`



Інструменти для роботи з SELinux

❑ Існує кілька інструментів для роботи з SELinux, наприклад:

- semanage
- semodule
- selinuxedit
- sestatus
- seinfo
- seaudit



Інструмент `semanage`

❑ `semanage` – це інструмент командного рядка, який використовується для управління політикою безпеки SELinux. Він дозволяє адміністраторам системи створювати, змінювати та видаляти правила SELinux.

❑ Основні команди `semanage`:

- `semanage fcontext` - використовується для управління контекстами безпеки файлів і каталогів
- `semanage login` - використовується для управління контекстами безпеки користувачів і груп
- `semanage port` - використовується для управління контекстами безпеки мережевих портів
- `semanage service` - використовується для управління контекстами безпеки служб
- `semanage boolean` - використовується для управління дозволами і обмеженнями, які застосовуються до процесів



Приклади використання semanage₁

❑ Налаштування контексту безпеки файлу:

- `semanage fcontext -a -t httpd_sys_content_t /var/www/html`
- Ця команда встановить контекст безпеки `httpd_sys_content_t` для всіх файлів у каталозі `/var/www/html`.

❑ Налаштування контексту безпеки користувача:

- `semanage login -a -u user -t user_u`
- Ця команда встановить контекст безпеки `user_u` для користувача `user`.



Приклади використання semanage₂

❑ Налаштування контексту безпеки мережевого порту:

- `semanage port -a -t http_port_t -p tcp 80`
- Ця команда встановить контекст безпеки `http_port_t` для мережевого порту 80.

❑ Налаштування контексту безпеки служби:

- `semanage service -a -s httpd -t httpd_r`
- Ця команда встановить контекст безпеки `httpd_r` для служби `httpd`.

❑ Налаштування дозволу для процесу:

- `semanage boolean -m httpd_can_network_connect`
- Ця команда дозволить процесам у ролі `httpd_r` підключатися до мережі.



Інструмент `semodule`

❑ `semodule` – це інструмент командного рядка, який використовується для управління модулями політики безпеки SELinux. Він дозволяє адміністраторам системи встановлювати, оновлювати та видаляти модулі політики безпеки.

❑ Основні команди `semodule`:

- `semodule load` - використовується для завантаження модуля політики безпеки
- `semodule unload` - використовується для вивантаження модуля політики безпеки
- `semodule update` - використовується для оновлення модуля політики безпеки
- `semodule delete` - використовується для видалення модуля політики безпеки



Приклади використання `semodule`

❑ Завантаження модуля політики безпеки:

- `semodule load httpd.pp`
- Ця команда завантажить модуль політики безпеки `httpd.pp`.

❑ Вивантаження модуля політики безпеки:

- `semodule unload httpd.pp`
- Ця команда вигрузить модуль політики безпеки `httpd.pp`.



Приклади використання `semodule`

❑ Оновлення модуля політики безпеки:

- `semodule update httpd.pp`
- Ця команда оновить модуль політики безпеки `httpd.pp`.

❑ Видалення модуля політики безпеки:

- `semodule delete httpd.pp`
- Ця команда видалить модуль політики безпеки `httpd.pp`.



Інструмент `selinuxedit`

❑ `selinuxedit` – це інструмент командного рядка, який використовується для редагування контекстів безпеки об'єктів SELinux. Він дозволяє адміністраторам системи змінювати контексти безпеки файлів, каталогів, мережевих портів, служб і користувачів.

❑ Основні команди `selinuxedit`:

- `**selinuxedit -f <файл>` - використовується для редагування контекстів безпеки об'єктів, які перелічені у файлі
- `**selinuxedit -a <об'єкт> -t <контекст>` - використовується для встановлення контексту безпеки об'єкта
- `**selinuxedit -d <об'єкт>` - використовується для видалення контексту безпеки об'єкта



Приклади використання `selinuxedit`

❑ Редагування контекстів безпеки файлів і каталогів:

- `selinuxedit -f /etc/selinux/targeted/contexts/files/file_contexts`
- Ця команда відкриє файл `/etc/selinux/targeted/contexts/files/file_contexts` для редагування.

❑ Встановлення контексту безпеки файлу:

- `selinuxedit -a /etc/passwd -t httpd_sys_content_t`
- Ця команда встановить контекст безпеки `httpd_sys_content_t` для файлу `/etc/passwd`.

❑ Видалення контексту безпеки файлу:

- `selinuxedit -d /etc/passwd`
- Ця команда видалить контекст безпеки для файлу `/etc/passwd`.



Інструмент `sestatus`

❑ `sestatus` – це інструмент командного рядка, який використовується для відображення стану політики безпеки SELinux. Він дозволяє адміністраторам системи переглядати контексти безпеки об'єктів, дозволи і обмеження, які застосовуються до процесів.

❑ Основні команди `sestatus`:

- `sestatus` - використовується для відображення загального стану політики безпеки SELinux
- `sestatus -a` - використовується для відображення контекстів безпеки всіх об'єктів
- `sestatus -b` - використовується для відображення дозволів і обмежень, які застосовуються до процесів



Приклади використання `sestatus`

❑ Відображення загального стану політики безпеки SELinux:

- `sestatus`

- Ця команда виведе наступну інформацію:

```
SELinux status: enabled
```

```
Current mode: enforcing
```

```
Mode from config file: enforcing
```

```
Policy version: 31
```

```
Policy type: targeted
```



Приклади використання `sestatus`

❑ Відображення контекстів безпеки всіх об'єктів:

- `sestatus -a`

- Ця команда виведе наступну інформацію:

```
SELinux status: enabled
```

```
Current mode: enforcing
```

```
Mode from config file: enforcing
```

```
Policy version: 31
```

```
Policy type: targeted
```

Contexts:



Приклади використання `sestatus`

❑ Відображення дозволів і обмежень, які застосовуються до процесів:

- `sestatus -b`
- Ця команда виведе наступну інформацію:
SELinux status: enabled
Current mode: enforcing
Mode from config file: enforcing
Policy version: 31
Policy type: targeted

Processes:



Інструмент `seinfo`

❑ `seinfo` – це інструмент командного рядка, який використовується для відображення інформації про політику безпеки SELinux. Він дозволяє адміністраторам системи переглядати правила, дозволи, атрибути та інші аспекти політики безпеки SELinux.

❑ Основні команди `seinfo`:

- `seinfo` - використовується для відображення загальної інформації про політику безпеки SELinux
- `seinfo -a` - використовується для відображення всіх правил політики безпеки SELinux
- `seinfo -b` - використовується для відображення всіх дозволів і обмежень, які застосовуються до процесів
- `**seinfo -c <клас>`` - використовується для відображення всіх правил політики безпеки SELinux для заданого класу
- `**seinfo -d <домен>`` - використовується для відображення всіх правил політики безпеки SELinux для заданого домену
- `**seinfo -r <роль>`` - використовується для відображення всіх правил політики безпеки SELinux для заданої ролі



Приклади використання `seinfo`

❑ Відображення загальної інформації про політику безпеки SELinux:

- `seinfo`
- Ця команда виведе наступну інформацію:
SELinux policy version: 31
SELinux policy type: targeted
SELinux policy configuration file:
/etc/selinux/config
SELinux policy modules:



Приклади використання `seinfo`

□ Відображення всіх правил політики безпеки SELinux:

- `seinfo -a`

- Ця команда виведе наступну інформацію:

```
SELinux policy version: 31
```

```
SELinux policy type: targeted
```

```
SELinux policy configuration file: /etc/selinux/config
```

```
SELinux policy modules:
```

```
allow httpd_r file_t:file { open; read; };
```

```
allow httpd_r net_t:tcp_socket { connect; listen; accept; };
```

```
allow httpd_r network_r:netlink_route_socket { read; write; };
```

```
...
```



Приклади використання `seinfo`

❑ Відображення всіх дозволів і обмежень, які застосовуються до процесів:

- `seinfo -b`

- Ця команда виведе наступну інформацію:

```
SELinux policy version: 31
```

```
SELinux policy type: targeted
```

```
SELinux policy configuration file: /etc/selinux/config
```

```
SELinux policy modules:
```

```
process_execve:
```

```
allow init_t init_t:process { execve };
```

```
allow init_t *:process { execve };
```

```
...
```




Приклади використання `seinfo`

❑ Відображення всіх правил політики безпеки SELinux для заданого класу:

- `seinfo -c file`

- Ця команда виведе наступну інформацію:

```
SELinux policy version: 31
```

```
SELinux policy type: targeted
```

```
SELinux policy configuration file: /etc/selinux/config
```

```
SELinux policy modules:
```

```
allow root file_t:dir { open; read; };
```

```
allow root file_t:file { open; read; };
```

```
allow httpd_r file_t:file { open; read; };
```

```
...
```



Приклади використання `seinfo`

❑ Відображення всіх правил політики безпеки SELinux для заданого домену:

- `seinfo -d system_u`
- Ця команда виведе наступну інформацію:

```
SELinux policy version: 31
```

```
SELinux policy type: targeted
```

```
SELinux policy configuration file: /etc/selinux/config
```

```
SELinux policy modules:
```

```
allow system_u file_t:dir { open; read; };
```

```
allow system_u file_t:file { open; read; };
```

```
allow system_u net_t:tcp_socket { connect; listen; accept;  
};
```

```
...
```



Приклади використання `seinfo`

❑ Відображення всіх правил політики безпеки SELinux для заданої ролі:

- `seinfo -r httpd_r`

- Ця команда виведе наступну інформацію:

```
SELinux policy version: 31
```

```
SELinux policy type: targeted
```

```
SELinux policy configuration file: /etc/selinux/config
```

```
SELinux policy modules:
```

```
allow httpd_r file_t:file { open; read; };
```

```
allow httpd_r net_t:tcp_socket { connect; listen; accept; };
```

```
...
```



Інструмент `seaudit`

❑ `seaudit` – це інструмент командного рядка, який використовується для відображення аудиту політики безпеки SELinux. Він дозволяє адміністраторам системи переглядати записи аудиту, які були створені внаслідок виконання правил SELinux.

❑ Основні команди `seaudit`:

- `seaudit` - використовується для відображення загального аудиту політики безпеки SELinux
- `seaudit -a` - використовується для відображення всіх записів аудиту політики безпеки SELinux
- `**seaudit -b <клас>`` - використовується для відображення всіх записів аудиту політики безпеки SELinux для заданого класу
- `**seaudit -d <домен>`` - використовується для відображення всіх записів аудиту політики безпеки SELinux для заданого домену
- `**seaudit -r <роль>`` - використовується для відображення всіх записів аудиту політики безпеки SELinux для заданої ролі



Приклади використання `seaudit`₁

❑ Відображення загального аудиту політики безпеки SELinux:

- `seaudit`

- Ця команда виведе наступну інформацію:

```
SELinux audit status: enabled
```

```
SELinux audit log file: /var/log/audit/audit.log
```

```
SELinux audit rules:
```



Приклади використання `seaudit`

❑ Відображення всіх записів аудиту політики безпеки SELinux:

- `seaudit -a`

- Ця команда виведе наступну інформацію:

```
SELinux audit status: enabled
```

```
SELinux audit log file: /var/log/audit/audit.log
```

```
SELinux audit rules:
```

```
<Запис аудиту 1>
```

```
<Запис аудиту 2>
```

```
...
```



Алгоритм налаштування SELinux для Ubuntu₁

1. Перевірте поточний статус SELinux.

- `sestatus`

□ Якщо SELinux вимкнено, увімкніть його за допомогою наступної команди:

- `setenforce enforcing`

2. Встановіть політику безпеки SELinux.

- `apt install selinux-policy-default`
- Ця команда встановить політику безпеки SELinux для стандартного типу.



Алгоритм налаштування SELinux для Ubuntu₂

3. Налаштуйте контексти безпеки для файлів і каталогів.

- `selinuxedit -f /etc/selinux/targeted/contexts/files/file_contexts`
- Ця команда відкриє файл `/etc/selinux/targeted/contexts/files/file_contexts` для редагування.

4. Налаштуйте контексти безпеки для служб.

- `semanage service -a -s <служба> -t <контекст>`
- Наприклад, щоб встановити контекст безпеки `httpd_r` для служби `httpd`, використовуйте наступну команду:
 - `semanage service -a -s httpd -t httpd_r`



Алгоритм налаштування SELinux для Ubuntu₃

5. Налаштуйте дозволи для процесів.

- `semanage boolean -m <boolean>`
- Наприклад, щоб дозволити процесам у ролі `httpd_r` підключатися до мережі, використовуйте наступну команду:
 - `semanage boolean -m httpd_can_network_connect`

6. Перевірте політику безпеки SELinux.

- `seinfo`
- Ця команда виведе список всіх правил політики безпеки SELinux.



Алгоритм налаштування SELinux для Ubuntu₃

7. Перевірте аудит політики безпеки SELinux.

- `seaudit`
- Ця команда виведе список всіх записів аудиту політики безпеки SELinux.



Реальний приклад конфігурації SELinux

643

```
# Налаштування контекстів безпеки  
semanage fcontext -a -t file_t /  
semanage fcontext -a -t dir_t /  
semanage fcontext -a -t net_t /etc/resolv.conf  
semanage fcontext -a -t httpd_sys_content_t /var/www/html
```

```
# Налаштування дозволів  
allow system_r:system_u:file_t:file { open; read; write; };  
allow system_r:system_u:file_t:dir { open; read; write; };  
allow system_r:system_u:net_t:file { open; read; write; };
```

```
# Налаштування аудиту  
auditd -w / -k all
```



Переваги SELinux

- ☐ Покращена безпека: SELinux може допомогти захистити систему від атак, обмежуючи доступ процесів до об'єктів.
- ☐ Більша прозорість: SELinux дозволяє адміністраторам системи краще зрозуміти, як об'єкти в системі взаємодіють один з одним.
- ☐ Легкість налаштування: SELinux можна налаштувати відповідно до потреб конкретної системи.



Недоліки SELinux

- ❑ Додаткове навантаження на систему: SELinux може додавати додаткове навантаження на систему, оскільки йому потрібно перевіряти правила політики безпеки для кожної операції.
- ❑ Складність налаштування: SELinux може бути складним у налаштуванні, особливо для систем з великою кількістю програм і служб.
- ❑ Несумісність з деякими програмами: Деякі програми можуть бути несумісні з SELinux, тому їх може знадобитися налаштувати або перекомпілювати.



Резюме

□ Загалом, SELinux – це потужний інструмент, який може допомогти підвищити безпеку системи. Однак важливо розуміти переваги та недоліки SELinux перед його використанням.

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux.

Лекція 18. Основи AppArmor

Предмет: БЗОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

648

- ☐ Основи AppArmor
- ☐ Основні концепції SELinux
- ☐ Ролі
- ☐ Домени
- ☐ Інструменти
- ☐ Налаштування SELinux на Ubuntu



Що таке AppArmor?

- ❑ AppArmor – це система керування доступом для Linux, яка заснована на профілях. Профіль AppArmor - це набір правил, які визначають, які ресурси може використовувати програма. AppArmor використовується для обмеження доступу програм до системних ресурсів, таких як файли, каталоги, мережні з'єднання та пристрої.
- ❑ Розробники та історія розвитку
 - AppArmor був розроблений компанією Immunix у 2000 році. У 2007 році компанія Immunix була придбана компанією Novell, яка в свою чергу була придбана компанією SUSE. У 2014 році SUSE передала розробку AppArmor спільноті Linux.
 - AppArmor є частиною ядра Linux з версії 2.6.32.

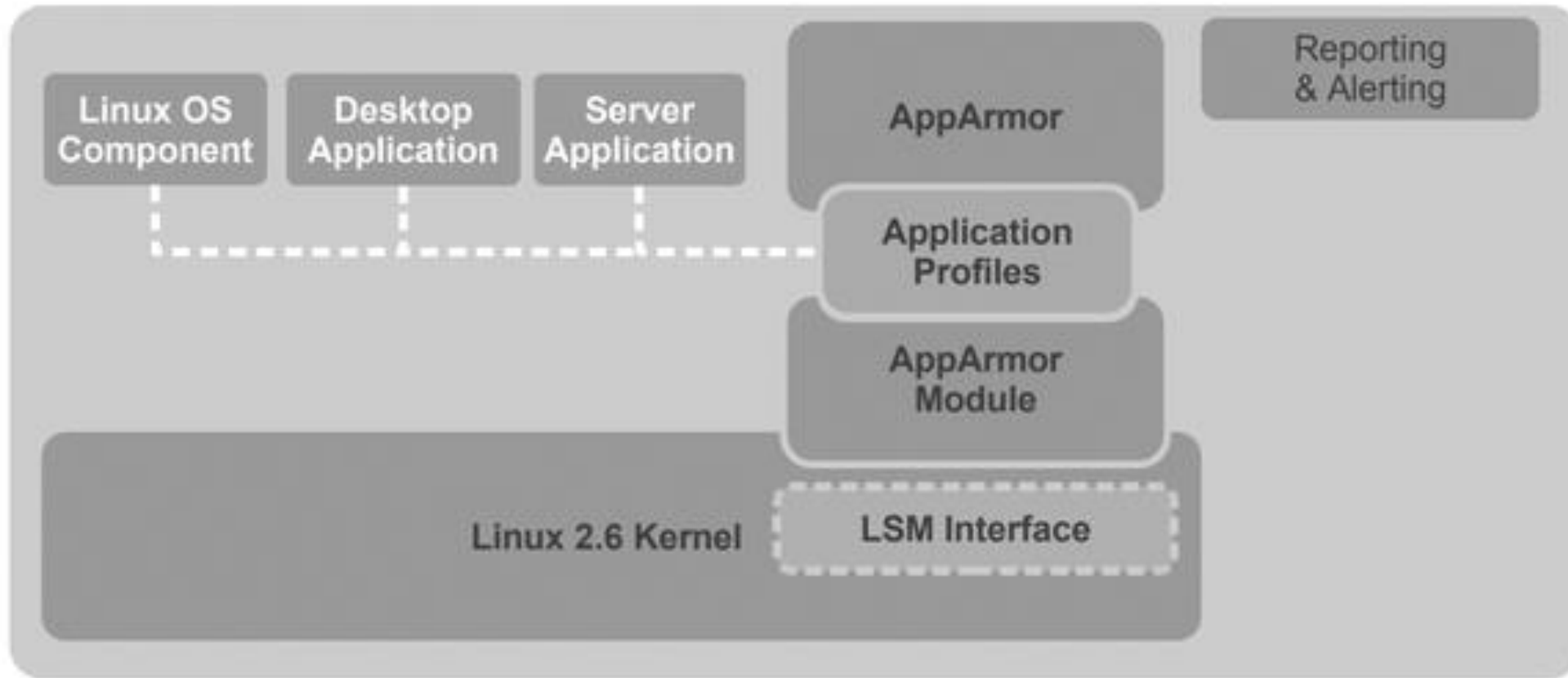


Використання AppArmor

- ❑ AppArmor заснований на концепції дискреційних дозволів. Це означає, що кожній програмі надається набір дозволів, які визначають, які ресурси вона може використовувати.
- ❑ AppArmor можна використовувати для захисту від різних типів атак, таких як:
 - Захист від несанкціонованого доступу до файлів і каталогів
 - Захист від несанкціонованого використання мережевих ресурсів
 - Захист від несанкціонованого виконання програм
- ❑ AppArmor є порівняно легкою в налаштуванні системою.



Архітектура AppArmor₁





Архітектура AppArmor₂

- AppArmor складається з наступних основних компонентів:
- Ядро Linux: Ядро Linux забезпечує базові функції безпеки, необхідні для роботи AppArmor.
 - Модуль AppArmor: Модуль AppArmor ядра Linux відповідає за виконання політики AppArmor.
 - Профілі AppArmor: Профілі AppArmor містять правила, які визначають, які ресурси може використовувати програма.



Принципи роботи AppArmor

- ❑ AppArmor заснований на концепції дискреційних дозволів. Це означає, що кожній програмі надається набір дозволів, які визначають, які ресурси вона може використовувати.
- ❑ AppArmor визначає, які ресурси може використовувати програма, на основі її профілю безпеки. Профіль безпеки - це набір правил, які визначають, які операції програма може виконувати з кожним типом ресурсу.



Ядро Linux та AppArmor₁

- ❑ AppArmor інтегрується в ядро Linux за допомогою модуля ядра AppArmor. Модуль AppArmor відповідає за виконання політики AppArmor.
- ❑ Модуль AppArmor інтегрується в ядро Linux на етапі його завантаження. Модуль AppArmor завантажується в ядро Linux і реєструється як модуль керування доступом.
- ❑ Взаємодія з ядром для контролю доступу
 - AppArmor взаємодіє з ядром Linux для контролю доступу до системних ресурсів.
 - Коли програма намагається отримати доступ до ресурсу, ядро Linux передає запит на модуль AppArmor. Модуль AppArmor перевіряє профіль безпеки програми, щоб визначити, чи має програма дозволу на доступ до ресурсу.
 - Якщо програма має дозвіл, модуль AppArmor дозволяє програмі отримати доступ до ресурсу. Якщо програма не має дозволу, модуль AppArmor відмовляє програмі в доступі до ресурсу.



Ядро Linux та AppArmor₂

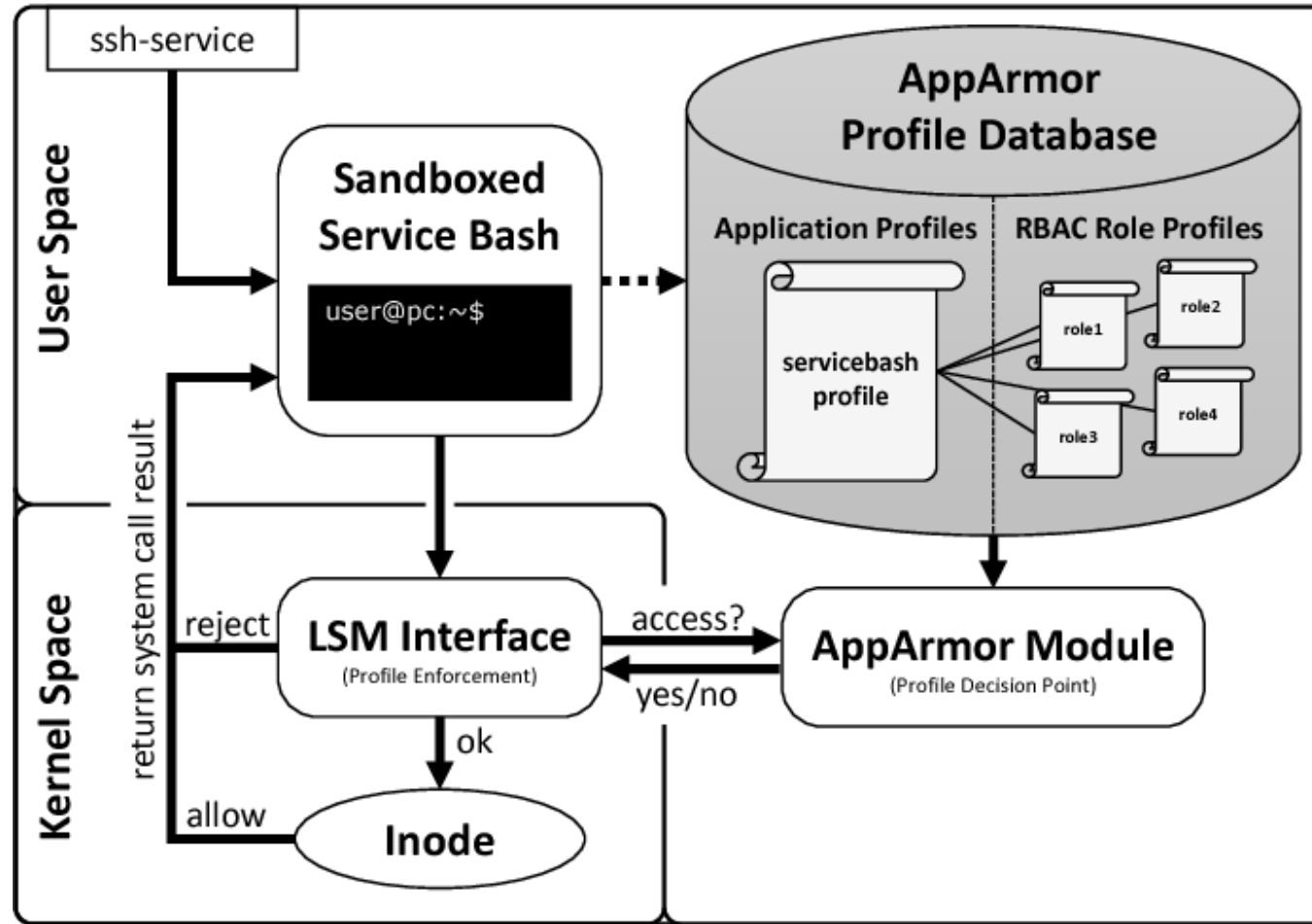


Fig. 3. Architecture of implementation.



Профілі безпеки AppArmor

- ❑ Профіль безпеки AppArmor - це набір правил, які визначають, які ресурси може використовувати програма.
- ❑ Профілі безпеки AppArmor можна налаштувати для різних типів програм, включаючи:
 - Програми користувача: Програми, які запускаються звичайними користувачами.
 - Служби: Програми, які працюють у фоновому режимі і надають послуги іншим програмам.
 - Ядро Linux: Ядро Linux виконує основні функції операційної системи.



Правило AppArmor

- ❑ Правило AppArmor – це конфігураційний елемент в профілі захисту для програмного забезпечення під керуванням AppArmor. Воно визначає обмеження та політику безпеки для конкретного об'єкта, такого як виконуваний файл або процес. Правила AppArmor дозволяють адміністраторам системи визначити, які ресурси та операції можуть бути використані або виконані додатком чи процесом.
- ❑ Кожне правило AppArmor визначає, які дозволи чи заборони встановлені для конкретного об'єкта та визначає умови, за яких ці обмеження будуть застосовані. Наприклад, правило може визначати, що конкретний процес має доступ тільки до певних файлів у файловій системі або може використовувати обмежений набір системних викликів.
- ❑ Правила AppArmor грають ключову роль у створенні безпечних середовищ виконання для додатків, дозволяючи детально налаштовувати обмеження та забезпечувати захист від потенційних загроз безпеки.



Синтаксис правил AppArmor

❑ Правила AppArmor мають наступний синтаксис:

```
<правило> {  
    <налаштування>;  
    <налаштування>;  
    ...  
}
```



Налаштування правила AppArmor

□ Налаштування правила визначає, які операції програма може виконувати з ресурсом. Налаштування правила можуть бути наступними:

- `allow`: дозволяє програмі виконувати операцію
- `deny`: забороняє програмі виконувати операцію
- `limit`: обмежує кількість разів, коли програма може виконувати операцію

□ **Наприклад, наступне правило дозволяє програмі `httpd` читати файли в каталозі `/var/www/html`:

```
/var/www/html {  
    allow read;  
}
```



Структура правил AppArmor

- ❑ Правила AppArmor можуть бути організовані в дерева. Корневим елементом дерева є файл, на який посилається правило. Правила, які стосуються одного файлу, можуть бути згруповані в один розділ.
- ❑ **Наприклад, наступний розділ визначає доступ для файлів, які належать до веб-сервера `httpd`:

```
/var/www/html {  
    allow read;  
    allow write;  
    allow execute;  
}
```



Налаштування доступу до файлів

- ❑ Правила AppArmor можна використовувати для налаштування доступу до файлів. Наприклад, наступне правило дозволяє програмі `httpd` читати файли в каталозі `/var/www/html`, але забороняє програмі писати в ці файли:

```
/var/www/html {  
    allow read;  
    deny write;  
}
```



Налаштування доступу до мережі

- ❑ Правила AppArmor можна використовувати для налаштування доступу до мережі. Наприклад, наступне правило дозволяє програмі httpd отримувати доступ до всіх мережевих ресурсів:

```
/usr/sbin/httpd {  
    allow network;  
}
```



Налаштування доступу до інших ресурсів

- ❑ Правила AppArmor можна використовувати для налаштування доступу до інших ресурсів, таких як пристрої, файлові системи та процеси. Наприклад, наступне правило дозволяє програмі httpd запускати інші програми:

```
/usr/sbin/httpd {  
    allow process;  
}
```



Інструменти управління AppArmor

□ AppArmor надає два основних інструменти для управління профілями безпеки:

- `aa-enforce`: запускає профіль безпеки в режимі жорсткого контролю. У цьому режимі AppArmor блокує будь-яку спробу програми отримати доступ до ресурсу, який не дозволений профілем безпеки.
- `aa-complain`: запускає профіль безпеки в режимі жорсткого контролю, але не блокує спроби програми отримати доступ до ресурсу, який не дозволений профілем безпеки. У цьому режимі AppArmor генерує повідомлення про помилку, але дозволяє програмі продовжити роботу.



Інструменти командного рядка для перегляду та зміни налаштувань

□ AppArmor надає ряд інструментів командного рядка для перегляду та зміни налаштувань профілів безпеки:

- `aa-status`: виводить список профілів безпеки, які активні в системі
- `aa-log`: виводить журнал AppArmor
- `aa-rules`: виводить список правил профілю безпеки
- `aa-genprof`: створює новий профіль безпеки
- `aa-updateprof`: оновлює існуючий профіль безпеки



Використання AppArmor в консольних та графічних інтерфейсах

- ☐ AppArmor можна використовувати в консольних та графічних інтерфейсах.
- ☐ У консольному інтерфейсі можна використовувати інструменти командного рядка, які надає AppArmor.
- ☐ У графічних інтерфейсах можна використовувати такі програми, як:
 - AppArmor GUI: графічний інтерфейс для управління AppArmor.
 - GNOME AppArmor: графічний інтерфейс для управління AppArmor, який входить до складу GNOME.



Використання aa-enforce та aa-complain

❑ Наприклад, щоб запустити профіль безпеки веб-сервера в режимі жорсткого контролю, можна виконати наступну команду:

- `aa-enforce /etc/apparmor.d/usr.sbin.apache2`

❑ Щоб запустити профіль безпеки веб-сервера в режимі жорсткого контролю, але генерувати повідомлення про помилку, якщо програма намагається отримати доступ до ресурсу, який не дозволений профілем безпеки, можна виконати наступну команду:

- `aa-complain /etc/apparmor.d/usr.sbin.apache2`



Використання інструментів командного рядка для перегляду та зміни налаштувань₁

- ☐ Наприклад, щоб вивести список профілів безпеки, які активні в системі, можна виконати наступну команду:
 - `aa-status`
- ☐ Щоб вивести журнал AppArmor, можна виконати наступну команду:
 - `aa-log`
- ☐ Щоб вивести список правил профілю безпеки, можна виконати наступну команду:
 - `aa-rules /etc/apparmor.d/usr.sbin.apache2`



Використання інструментів командного рядка для перегляду та зміни налаштувань₂

❑ Щоб створити новий профіль безпеки, можна виконати наступну команду:

- `aa-genprof /etc/apparmor.d/<назва_профілю>`

❑ Щоб оновити існуючий профіль безпеки, можна виконати наступну команду:

- `aa-updateprof /etc/apparmor.d/<назва_профілю>`



Використання AppArmor в графічних інтерфейсах

❑ Щоб відкрити AppArmor GUI, можна виконати наступну команду:

- `apparmor-gui`

❑ Щоб відкрити GNOME AppArmor, можна виконати наступну команду:

- `gnome-apparmor-properties`



Алгоритм налаштування AppArmor на Ubuntu₁

1. Встановіть AppArmor

❑ `sudo apt install apparmor`

2. Створіть профілі безпеки

❑ Вручну: відкрийте текстовий редактор і введіть правила для профілю безпеки.

❑ За допомогою `aa-genprof`: виконайте команду `aa-genprof /etc/apparmor.d/<назва_профілю>`



Алгоритм налаштування AppArmor на Ubuntu₂

3. Активуйте профілі безпеки

- ☐ Вручну: виконайте команду `sudo aa-enforce /etc/apparmor.d/<назва_профілю>` для кожного профілю безпеки, який ви хочете активувати.
- ☐ За допомогою `aa-enforce`: виконайте команду `sudo aa-enforce` для активації всіх профілів безпеки, які є в каталозі `/etc/apparmor.d/`.

4. Перевірте правильність конфігурації

- ☐ Використовуйте інструменти AppArmor для перегляду та зміни налаштувань.
- ☐ Протестуйте програми, які захищені AppArmor.
- ☐ Перегляньте журнал AppArmor, щоб переконатися, що AppArmor блокує несанкціоновані дії.



Приклад налаштування AppArmor для веб-сервера Apache на Ubuntu₁

- ☐ Веб-сервер Apache є одним з найпоширеніших веб-серверів в світі. Він є стандартною програмою в дистрибутивах Linux, включаючи Ubuntu.
- ☐ Щоб захистити веб-сервер Apache за допомогою AppArmor, необхідно створити профіль безпеки для цієї програми.



Приклад налаштування AppArmor для веб-сервера Apache на Ubuntu₁

- ☐ Веб-сервер Apache є одним з найпоширеніших веб-серверів в світі. Він є стандартною програмою в дистрибутивах Linux, включаючи Ubuntu.
- ☐ Щоб захистити веб-сервер Apache за допомогою AppArmor, необхідно створити профіль безпеки для цієї програми.



Приклад налаштування AppArmor для веб-сервера Apache на Ubuntu₂

❑ Відкрийте текстовий редактор і введіть наступний текст:

```
/usr/sbin/apache2 {  
    # Дозволи на файли  
    allow read;  
    allow write;  
    allow execute;  
  
    # Дозволи на мережу  
    allow network;  
}
```

❑ Цей профіль безпеки дозволяє веб-серверу Apache отримувати доступ до файлів, виконувати інші програми та використовувати мережу.



Приклад налаштування AppArmor для веб-сервера Apache на Ubuntu₃

- ☐ Виконайте команду `sudo aa-enforce /etc/apparmor.d/usr.sbin.apache2.`
- ☐ Після цього веб-сервер Apache буде захищений AppArmor.



Приклад налаштування AppArmor для веб-сервера Apache на Ubuntu₄

- ❑ Залежно від ваших потреб, ви можете додати додаткові правила до профілю безпеки Apache. Наприклад, ви можете заборонити веб-серверу Apache доступ до певних файлів або каталогів.
- ❑ Щоб заборонити веб-серверу Apache доступ до каталогу `/etc/passwd`, додайте наступне правило до профілю безпеки:

```
/usr/sbin/apache2 {  
    # ...  
  
    # Заборонити доступ до каталогу /etc/passwd  
    deny /etc/passwd;  
}
```



Переваги AppArmor

- ❑ *Ефективний захист системи від несанкціонованих дій.*
AppArmor дозволяє контролювати доступ програм до системних ресурсів, що допомагає запобігти атакам на систему.
- ❑ *Легка налаштування.* AppArmor надає простий синтаксис для створення профілів безпеки.
- ❑ *Гнучкість.* AppArmor дозволяє налаштовувати правила безпеки з урахуванням конкретних потреб системи.



Недоліки AppArmor

- ❑ *Потрібні знання Linux.* Для ефективного використання AppArmor необхідно мати базові знання Linux.
- ❑ *Може бути несумісний з деякими програмами.* AppArmor може блокувати роботу деяких програм, якщо вони не були розроблені з урахуванням цієї технології.



Резюме

680

- ❑ AppArmor є ефективним засобом захисту системи від несанкціонованих дій. Він простий у налаштуванні та гнучкий. Однак, для ефективного використання AppArmor необхідно мати базові знання Linux.

Донецький національний технічний університет



Тема 8. Засоби захисту ОС Linux.

Лекція 19. Аудит в Linux

Предмет: БЗОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

682

- ☐ Можливості аудиту
- ☐ Типи правил
- ☐ Створення звітів
- ☐ Візуалізація даних аудиту



5 золотих правил безпеки

683

- ☐ Знай свою систему
- ☐ Принцип найменших привілеїв
- ☐ Захист в глибину
- ☐ Захист є ключовим, але виявлення є обов'язковим
- ☐ Знай свого ворога



Аудит

684

- ☐ Аудит Linux допомагає зробити вашу систему більш безпечною, надаючи вам засоби для детального аналізу того, що відбувається у вашій системі.
- ☐ Однак він сам по собі не забезпечує додаткової безпеки — він не захищає вашу систему від збоїв у роботі коду чи будь-яких експлойтів.
- ☐ Натомість Audit корисний для відстеження цих проблем і допомагає вжити додаткових заходів безпеки, як-от SELinux, щоб запобігти їм.
- ☐ Аудит складається з кількох компонентів, кожен з яких вносить важливу функціональність у загальну структуру.
- ☐ Модуль ядра аудиту перехоплює системні виклики та записує відповідні події.



Демон та утиліти

- ☐ Демон auditd записує звіти аудиту на диск
- ☐ Різні утиліти командного рядка піклуються про відображення, запити та архівування контрольного сліду



Можливості аудиту в Linux₁

□ Аудит дозволяє зробити наступне:

- Пов'язати користувачів із процесами
 - Аудит зіставляє процеси з ідентифікатором користувача, який їх запустив.
 - Це дає змогу адміністратору або офіцеру безпеки точно відстежити, який користувач володіє яким процесом і потенційно виконує зловмисні операції в системі.
- Переглядати аудиторський слід
 - Аудит Linux надає інструменти, які записують звіти про аудит на диск і переводять їх у формат, зрозумілий людині.
- Огляд окремих подій аудиту
 - Аудит надає утиліту, яка дозволяє фільтрувати звіти аудиту за певними подіями, що представляють інтерес



Можливості аудиту в Linux₂

❑ Можна фільтрувати за такими сутностями як:

- Користувач
- Група
- Ідентифікатор аудиту
- Ім'я віддаленого хоста
- Адреса віддаленого хосту
- Системний виклик
- Аргумент системного виклику
- Файл
- Файлова операція
- Сеанс
- Успіх чи невдача



Можливості аудиту в Linux₃

❑ Вибірковий аудит

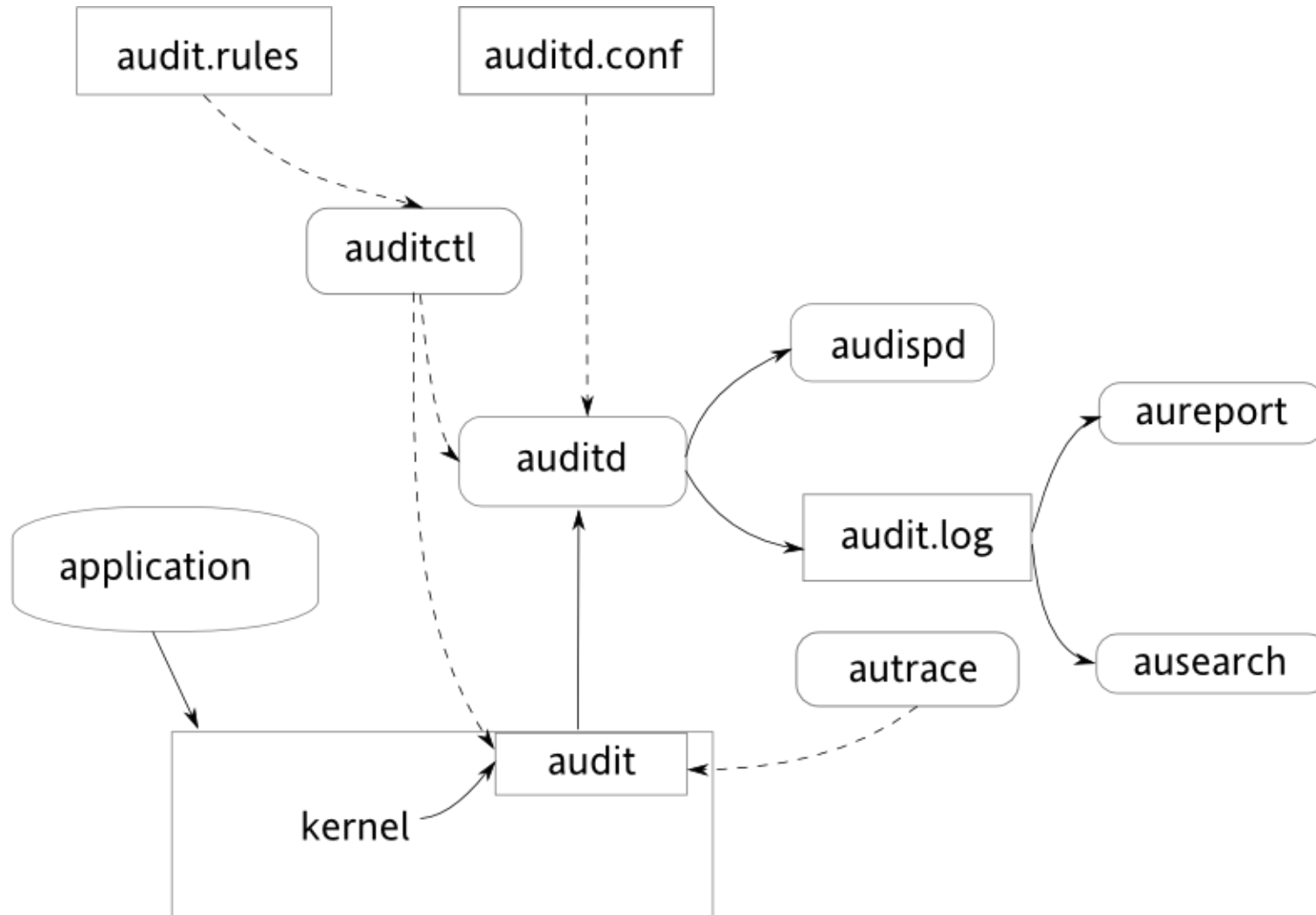
- Аудит надає засоби фільтрації звітів аудиту для подій, що цікавлять, а також налаштування аудиту для запису лише вибраних подій.
- Можна створити свій власний набір правил і дозволити демонстру аудиту записувати лише ті, які вас цікавлять.

❑ Запобігання втраті даних аудиту

- Аудит надає кілька механізмів для запобігання втраті даних аудиту в разі втрати системних ресурсів.



Компоненти аудиту в Linux





Кроки налаштування аудиту

- ❑ Налаштування аудиту розділено на три частини:
 - Конфігурація демона аудиту
 - Правила аудиту
 - Конфігурація демона Audispd



/etc/audit/auditd.conf

691

- ❑ Конфігураційний файл `/etc/audit/auditd.conf` визначає, як функціонує система аудиту після запуску демона.
- ❑ Директиви вказують `auditd`, куди розміщувати файли журналу аудиту, як проводити очищення записів аудиту, керувати файлами журналу аудиту та виконувати обробку помилок.
- ❑ Для більшості випадків використання налаштувань за замовчуванням, які постачаються з пакетом, має бути достатнім.



Приклад auditd.conf

```
log_file =  
/var/log/audit/audit.log  
log_format = RAW  
log_group = root  
priority_boost = 4  
flush = INCREMENTAL  
freq = 20  
num_logs = 5  
dispatcher = sbin/audispd  
disp_qos = lossy  
name_format = hostname  
##name = mydomain
```

```
max_log_file = 6  
max_log_file_action = ROTATE  
space_left = 75  
space_left_action = SYSLOG  
action_mail_acct = root  
admin_space_left = 50  
admin_space_left_action = SUSPEND  
disk_full_action = SUSPEND
```



/etc/audit/audit.rules

693

- ❑ `audit.rules` — це файл, що містить правила аудиту, які завантажуватимуться сценарієм запуску демона аудиту під час кожного запуску демона.
- ❑ Програма `auditctl` використовується сценарієм запуску для виконання цієї операції.
- ❑ Правила аудиту бувають 3 різновидів:
 - управління
 - файлової системи
 - системний виклик



Правила управління (контролю)

- ☐ Правила управління, як правило, передбачають налаштування системи аудиту, а не вказують їй, що спостерігати.
- ☐ Ці правила, розташовані у верхній частині файлу правил, зазвичай включають:
 - Видалення всіх правил (-D)
 - Встановлення розміру резервної черги ядра (-b)
 - Встановлення режиму збою (-f)
 - Встановлення обмеження частоти подій (-r)
 - Встановлення режиму ігнорування синтаксичних помилок в правилах (-i) і (-c)
 - Увімкнути/вимкнути аудит (-e)



Правила файлової системи

- ❑ Правила файлової системи іноді називають спостереженнями.
- ❑ Ці правила використовуються для перевірки доступу до певних файлів або каталогів, які можуть вас зацікавити.
- ❑ Якщо шлях, указаний у правилі, є каталогом, то правило, яке використовується, є рекурсивним до нижньої частини дерева каталогів, за винятком будь-яких каталогів, які можуть бути точками монтування.
- ❑ Синтаксис цих правил зазвичай має такий формат:
 - `-w` шлях до файлу `-r` дозволи `-k` ім'я ключа
 - де дозвіл є будь-яким із наступного:
 - `r` – читання файлу
 - `w` – запис у файл
 - `x` – виконання файлу
 - `a` – зміна атрибута файлу
 - `keyname` — це довільний рядок тексту, який використовується для унікальної ідентифікації записів аудиту, створених правилом



Перевірка цілісності

❑ Використовуючи спостереження за файлами є сенс написати список правил для виявлення змін у файлах правил аудиту та створених правилах аудиту в ядрі:

- `-w /etc/audit/auditd.conf -p wa -k audit-config`
- `-w /etc/audit/auditd.rules -p wa -k audit-config`
- `-w /sbin/auditctl -p x -k audit-config`



Секрети налаштувань спостережень

- ☐ Якщо вам потрібні детальні записи, пов'язані з файлами, увімкніть окремі спостереження для всіх цікавих файлів.
- ☐ Спостереження за каталогами створюють менш детальні журнали, ніж спостереження за конкретними файлами.
- ☐ Аудит не підтримує глобалізацію шляхів будь-якого типу. Завжди використовуйте точні імена шляхів.
- ☐ Аудит можна виконувати лише для існуючих файлів.
- ☐ Будь-які файли, додані під час роботи демона аудиту, ігноруються, доки набір правил аудиту не буде оновлено для перегляду нових файлів.
- ☐ Пам'ятайте: спрацьовує перший матчинг!



Аудит виконання двійкових файлів Setuid/Setgid₁

- ☐ Припустімо, що з метою відповідності вам потрібно перевірити виконання двійкових файлів setuid/setgid у вашій системі.
- ☐ Як це зробити?
- ☐ Спочатку запустіть сценарій під час завантаження з `/etc/rc.local` надсилаючи виведення до тимчасового файлу, наприклад, `/tmp/snorf`.



Скрипт

```
#!/bin/bash
# Пошук усіх локально змонтованих файлових систем
for i in ` /bin/egrep /etc/fstab | /bin/awk '{print $2}' `
'(ext4|ext3|ext2)' do
# Знайти усі файли у файловій системі, знайдений вище, і роздрукувати їх
# і правило аудиту для нього
/usr/bin/find $i -xdev -type f \( -perm -4000 -o -perm -2000 \) -print | \
/bin/sort | /bin/awk '{ print "-w " $1 " -p x -k privileged -k ids-exec-high"
}'

done
```



Результат роботи скрипта

```
-a -w /bin/cgclassify -p x -k privileged -k ids-exec-high
-a -w /bin/cgexec -p x -k privileged -k ids-exec-high
-a -w /bin/ping -p x -k privileged -k ids-exec-high
-a -w /bin/ping6 -p x -k privileged -k ids-exec-high
-a -w /bin/su -p x -k privileged -k ids-exec-high
-a -w /sbin/mount.nfs -p x -k privileged -k ids-exec-high
-a -w /sbin/netreport -p x -k privileged -k ids-exec-high
-a -w /sbin/pam_timestamp_check -p x -k privileged -k ids-exec-high
-a -w /sbin/unix_chkpwd -p x -k privileged -k ids-exec-high
-a -w /usr/bin/chage -p x -k privileged -k ids-exec-high
-a -w /usr/bin/chfn -p x -k privileged -k ids-exec-high
-a -w /usr/bin/chsh -p x -k privileged -k ids-exec-high
-a -w /usr/bin/crontab -p x -k privileged -k ids-exec-high
-a -w /usr/bin/gpasswd -p x -k privileged -k ids-exec-high
-a -w /usr/bin/ksu -p x -k privileged -k ids-exec-high
```



Аудит виконання двійкових файлів

Setuid/Setgid₂

- ❑ Потім вкажіть `auditctl` на тимчасовий файл, щоб додати щойно створені правила аудиту.
- ❑ Програма `auditctl` використовується для контролю поведінки, отримання статусу та додавання чи видалення правил у системі аудиту ядра.

```
/sbin/auditctl -R /tmp/snorf
```

- ❑ Кілька речей про `auditctl`:
 - `auditctl` не є фільтром, тому вихідні дані не можуть бути передані в нього
 - файли правил для `auditctl` мають належати `root`



Правила системних викликів₁

- ❑ Правила системних викликів завантажуються у відповідний механізм, який перехоплює кожен системний виклик, який здійснюють усі програми в системі.
- ❑ Тому дуже важливо використовувати правила системного виклику лише тоді, коли це необхідно, оскільки вони впливають на продуктивність. Чим більше правил, тим більший удар по ефективності.
- ❑ Однак можна підвищити продуктивність, об'єднавши системні виклики в одне правило, коли це можливо.
- ❑ Пам'ятайте: виграє перший матчінг!
- ❑ Правила Syscall мають загальну форму:
 - -a дія, список -S системний виклик -F поле=значення -k ім'я ключа



Правила системних викликів₂

- ❑ Параметр -а повідомляє механізму відповідності правил ядра, що ми хочемо додати правило та кінець списку правил.
- ❑ Але нам потрібно вказати, до якого списку правил воно входить і яку дію виконувати, коли воно запускається.
- ❑ Дія та список розділені комою, але без пробілу між ними.
Дійсні списки: `task, entry, exit, user, exclude`.
- ❑ Допустимі дії:
 - `always` - завжди створювати подію
 - `never` - ніколи не створюйте події



Правила системних викликів₃

- ❑ Наступним у правилі є параметр -S, що вказує назву або номер системного виклику. Зазвичай ім'я використовується майже завжди. Ви можете надати більше ніж один системний виклик у правилі, вказавши інший параметр -S.
- ❑ Після вказівки системного виклику ви, як правило, матимете один або кілька параметрів -F, які точно налаштовують, з чим матчити.
- ❑ Система аудиту розглядає uid як числа без знаку та використовує число -1, щоб вказати, що loginuid не встановлено. Це означає, що роздрукований номер виглядає як 4294967295.
- ❑ Остання річ щодо правил системного виклику полягає в тому, що можна додати ім'я ключа, яке є текстовим рядком у довільній формі, який необхідно вставити в подію, щоб допомогти в майбутньому визначити його призначення.



Приклади правил системних викликів

❑ Для перегляду файлів, відкритих визначеним користувачем:

- `-a exit,always -S open -F auid=1337`

❑ Для перегляду неуспішних відкритих викликів:

- `-a exit,always -S open -F success=0`



Фіксація спроб зміни системного часу

- ❑ Хакери часто користуються системним часом, щоб приховати свої дії. Використовуйте аудит системних викликів, щоб створити правила аудиту для запису змін системного часу.

```
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k  
-a always,exit -F arch=b64 audit_time_rules  
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -k -audit_time_rules  
-a always,exit -F arch=b64 -S clock_settime -k audit_time_rules  
-S clock_settime -k audit_time_rules
```



Створення звітів з аудиту₂

❑ Використовуйте `aureport` для створення лаконічних, зрозумілих для людини звітів.

❑ Деякі з корисних варіантів:

- `--summary`
- `--failed`
- `--start i --end` (`aureport` розуміє `today, yesterday, now, recent, this-week, this-month, this-year`)
- `--auth, --avc, --login, --user, --executable, --syscall`

❑ Для початку виконайте `aureport --summary`, і отримаєте щось на зразок цього:



Створення звітів з аудиту₃

Summary Report

=====

Range of time in logs: 08/09/2015 02:28:36.498 - 10/02/2015 18:07:38.737

Selected time for report: 08/09/2015 02:28:36 - 10/02/2015 18:07:38.737

Number of changes in configuration: 0

Number of changes to accounts, groups, or roles: 4

Number of logins: 16

Number of failed logins: 13134

Number of authentications: 59

Number of failed authentications: 46567

Number of users: 9

Number of terminals: 18

Number of host names: 1139

Number of executables: 14

Number of files: 0

Number of AVC's: 0

Number of MAC events: 0

Number of failed syscalls: 0

Number of anomaly events: 0

Number of responses to anomaly events: 0

Number of crypto events: 50419

Number of keys: 0

Number of process IDs: 25791

Number of events: 129193

708



Створення звітів з аудиту₄

□ Давайте розглянемо деякі з невдалих входів за допомогою `aureport -auth --failed:`

Login Report

=====

date time auid host term exe success event

```
1. 09/23/2015 19:53:04 send 195.4.234.9 sshd /usr/sbin/sshd no 2469819
2. 09/23/2015 19:53:06 send 195.4.234.9 sshd /usr/sbin/sshd no 2469821
3. 09/23/2015 19:53:10 root 195.4.234.9 sshd /usr/sbin/sshd no 2469825
4. 09/23/2015 19:53:12 ts 195.4.234.9 sshd /usr/sbin/sshd no 2469828
5. 09/23/2015 19:53:14 ts 195.4.234.9 sshd /usr/sbin/sshd no 2469830
6. 09/23/2015 19:56:37 root 218.65.30.217 sshd /usr/sbin/sshd no 2469852
7. 09/23/2015 19:56:40 root 218.65.30.217 sshd /usr/sbin/sshd no 2469854
8. 09/23/2015 19:56:42 root 218.65.30.217 sshd /usr/sbin/sshd no 2469856
9. 09/23/2015 21:34:52 ubnt 222.186.21.154 sshd /usr/sbin/sshd no 2470589
10. 09/23/2015 21:34:54 ubnt 222.186.21.154 sshd /usr/sbin/sshd no 2470591
11. 09/24/2015 00:25:15 root 218.4.117.26 sshd /usr/sbin/sshd no 2471855
12. 09/24/2015 00:25:17 pi 218.4.117.26 sshd /usr/sbin/sshd no 2471858
13. 09/24/2015 00:25:19 pi 218.4.117.26 sshd /usr/sbin/sshd no 2471860
14. 09/24/2015 00:25:21 test 218.4.117.26 sshd /usr/sbin/sshd no 2471863
15. 09/24/2015 00:25:23 test 218.4.117.26 sshd /usr/sbin/sshd no 2471865
```



Пошук в глибину з `ausearch`₁

- ❑ `aureport` дозволяє створювати загальні підсумки того, що відбувається в системі, але якщо необхідно глибше розібратися в деталях конкретної події, інструментом для використання є `ausearch`.
- ❑ `ausearch` дозволяє шукати в журналах аудиту за допомогою спеціальних ключів і пошукових фраз, які стосуються більшості позначок, які з'являються в повідомленнях про події в `/var/log/audit/audit.log`
- ❑ Методологія для використання полягає в тому, щоб знайти цікавий клас події за допомогою `aureport`, а потім детально розібратися в дрібницях за допомогою `ausearch`.
- ❑ Наприклад, використовується `aureport --syscall --failed`, щоб побачити невдалі системні виклики. Використайте далі `ausearch` та один із ідентифікаторів подій, щоб отримати більше інформації.



Пошук в глибину з ausearch₂

```
❏ aureport --syscall --fail
```

Syscall Report

```
=====
# date time syscall pid comm auid event
=====
1. 10/02/2015 15:26:16 2 5630 cp 25016 306512
```

```
❏ ausearch -i -a 306512
```

```
type=PATH msg=audit(10/02/2015 15:26:16.299:306512) : item=0
name=/etc/shadow inode=394070 dev=fd:00 mode=file,000 ouid=root ogid=root
rdev=00:00 obj=system_u:object_r:shadow_t:s0 nametype=NORMAL
type=CWD msg=audit(10/02/2015 15:26:16.299:306512) : cwd=/home/dr-horrible
type=SYSCALL msg=audit(10/02/2015 15:26:16.299:306512) : arch=x86_64 syscall=open
success=no exit=-13(Permission denied) a0=0x7ffde2dd8739 a1=O_WRONLY|O_TRUNC a2=0x0
a3=0x7ffde2dd6bd0 items=1 ppid=27491 pid=5630 auid=dr-horrible uid=dr-horrible
gid=users euid=dr-horrible suid=dr-horrible fsuid=dr-horrible egid=users sgid=users
fsgid=users tty=pts0 ses=16292 comm=cp exe=/bin/cp
subj=unconfined_u:unconfined_r:unconfined_t:s0- s0:c0.c1023 key=identity
```



Пошук в глибину з ausearch₃

❑ Корисна функція – пошук за мітками.

❑ Наприклад, `ausearch -ts today -i -k identity і ми`
отримаємо:

```
type=PATH msg=audit(10/02/2015 15:26:16.299:306512) : item=0
name=/etc/ shadow inode=394070 dev=fd:00 mode=file,000 ouid=root
ogid=root rdev=00:00 obj=system_u:object_r:shadow_t:s0
nametype=NORMAL
type=CWD msg=audit(10/02/2015 15:26:16.299:306512) : cwd=/home/dr-
horrible
type=SYSCALL msg=audit(10/02/2015 15:26:16.299:306512) : arch=x86_64
syscall=open success=no exit=-13(Permission denied) a0=0x7ffde2dd8739
a1=0_WRONLY|O_TRUNC a2=0x0 a3=0x7ffde2dd6bd0 items=1 ppid=27491
pid=5630 auid=dr-horrible uid=dr-horrible gid=users euid=dr-horrible
suid=dr-horrible fsuid=dr-horrible egid=users sgid=users fsgid=users
tty=pts0 ses=16292 comm=cp exe=/bin/cp
subj=unconfined_u:unconfined_r:unconfined_t:s0- s0:c0.c1023
key=identity
```




Evil Sysadmin₁

❑ Ви підозрюєте, що системний адміністратор, який діє як «root», перебирає файлову систему /home у пошуках інтелектуальної власності для продажу конкурентам. Напишіть правило(а) аудиту, яке реєструватиме тралення під root.

❑ Спосіб зробити це — використати опцію -C для створення правила порівняння між полями:

```
-a exit,always -F dir=/home/ -F uid=0 -C  
auid!=obj_uid -k admin-access
```



Evil Sysadmin₂

- ❑ Оскільки в правилі не було явного посилання на системний виклик, усі системні виклики перевіряються на відповідність цьому правилу.
- ❑ Це може призвести до значного зниження продуктивності.
Кращий спосіб це зробити:

```
-a always,exit -F arch=b32 -S open -S openat -S -S truncate -S  
open_by_handle_at ftruncate -F dir=/home/ -F uid=0 -C auid!=4294967295 -k  
auid!=obj_uid -F auid>=500 -F admin-access  
-a always,exit -F arch=b64 -S open -S openat -S -S truncate -S  
open_by_handle_at ftruncate -F dir=/home/ -F uid=0 -C auid!=4294967295 -k  
auid!=obj_uid -F auid>=500 -F admin-access
```



Аудит програм незалежно від розташування

- ☐ Згадайте сценарій, який записує правила аудиту для пошуку виконуваних файлів `setuid`. Як записати виконання програми `setuid`, розміщеної в неочікуваному місці?
- ☐ Напишіть правило аудиту, щоб відстежувати виконання програми `setuid` незалежно від її розташування.

```
-a always,exit -F arch=b64 -S execve -F  
euid=0 -C uid!=euid
```



Візуалізація даних аудиту в Linux

- ❑ Існує безліч інструментів для візуалізації даних аудиту в Linux. Деякі з найбільш поширених інструментів включають:
- SIEM (Security Information and Event Management) - система, яка збирає і аналізує дані з різних джерел, включаючи дані аудиту.
 - Matplotlib - бібліотека Python для створення графічних зображень.
 - Seaborn - бібліотека Python, яка надає додаткові функції для створення візуалізацій даних.



SIEM₁

717

- ❑ SIEM-системи, як правило, мають більш потужні функції візуалізації, ніж Auditd. SIEM-системи можуть використовуватися для створення візуалізацій даних аудиту в різних форматах, таких як графіки, таблиці та карти.
- ❑ Однією з популярних SIEM-систем є Splunk. Splunk надає широкий спектр функцій візуалізації, включаючи:
 - Графіки
 - Таблиці
 - Карти
 - Діаграми
 - Візуалізації часу
 - Візуалізації простору
 - Візуалізації поведінки



SIEM₂

718

❑ Щоб створити візуалізацію даних аудиту в Splunk, виконайте наступні кроки:

- Створіть запит для даних аудиту, які необхідно візуалізувати.
- Виберіть тип візуалізації, який потрібно створити.
- Налаштуйте візуалізацію відповідно до потреб.



Донецький національний технічний університет



Тема 9. Архітектура безпеки і механізми захисту Android.

Лекція 20. Механізми захисту Android

Предмет: БЗОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

720

- ☐ Вступ
- ☐ Апаратно-забезпечена безпека
- ☐ Android Keystore System
- ☐ Біометрія



ОС мобільних пристроїв: специфічні вимоги

- ❑ Мобільні пристрої мають обмежені ресурси: обчислювальна потужність, пам'ять, обсяг сховища даних, ємність батареї
- ❑ Мобільні пристрої мають різні форм-фактори і можливості: розміри і роздільну здатність екранів, клавіатури (або їх відсутність взагалі), різні бездротові інтерфейси, інтегровані і зовнішні периферійні пристрої, різні архітектури процесорів (ARM, x86) тощо
- ❑ ОС має бути орієнтована на непрофесійного користувача: має бути легкою в користуванні, інтуїтивно зрозумілою, не вимагати значних зусиль в обслуговуванні, але забезпечувати достатню безпеку, зокрема протидію шкідливим програмам
- ❑ Ринок вимагає позитивних перших вражень, швидкого розвитку, і не пробачає помилок



Принципи, виведені з вимог

- ☐ Ядро ОС має бути компактним і вимагати мінімальний обсяг ресурсів
- ☐ ОС не буде застосовувати витіснення коду і даних з оперативної пам'яті на зовнішній пристрій і повинна працювати і керувати прикладними програмами у фіксованому обсязі ОП
- ☐ Повинні бути надані API і механізми, що дозволять пристрою “спати” будь-коли це можливо для економії батареї
- ☐ Прикладні програми мають бути абстраговані від ОС і апаратури
- ☐ Прикладні програми мають виконуватись у пісочниці для мінімізації впливу шкідливих програм
- ☐ Стандартні компоненти, що існують, повинні застосовуватись максимально
- ☐ Необхідно надати можливість стороннім розробникам створювати безліч прикладних програм
- ☐ Треба створити власний магазин прикладних програм (App store) і зробити його привабливим для розробників



Реалізація принципів в ОС Android₁

1.Основою платформи Android є модифіковане ядро Linux

- ☐ Linux є надійною масштабованою технологією, працює на мільйонах вбудованих пристроїв, має потужну підтримку різноманітної апаратури і розвинені засоби безпеки
- ☐ Завдяки своїй історії постійного дослідження, атак та виправлень з боку тисяч розробників, Linux перетворився на стабільне та безпечне ядро, якому довіряють багато корпорацій та фахівців з безпеки
- ☐ Плюс, це ядро з відкритим кодом

2.Java для прикладних програм

- ☐ Java є доведеною технологією популярною у розробників: Enterprise (JSP), Mobile (J2ME) та інші
- ☐ Програми Java виконуються у віртуальній машині, тобто можуть бути незалежними від ОС і обладнання
- ☐ Для мінімізації вимог до процесору і пам'яті бібліотеки Java і середовище виконання Java (VM/runtime) були суттєво перероблені, тому "Android Java" не є 100% еквівалентною "Sun/Oracle Java"



Реалізація принципів в ОС Android₂

724

3. Оптимізована Java VM в якості середовища виконання

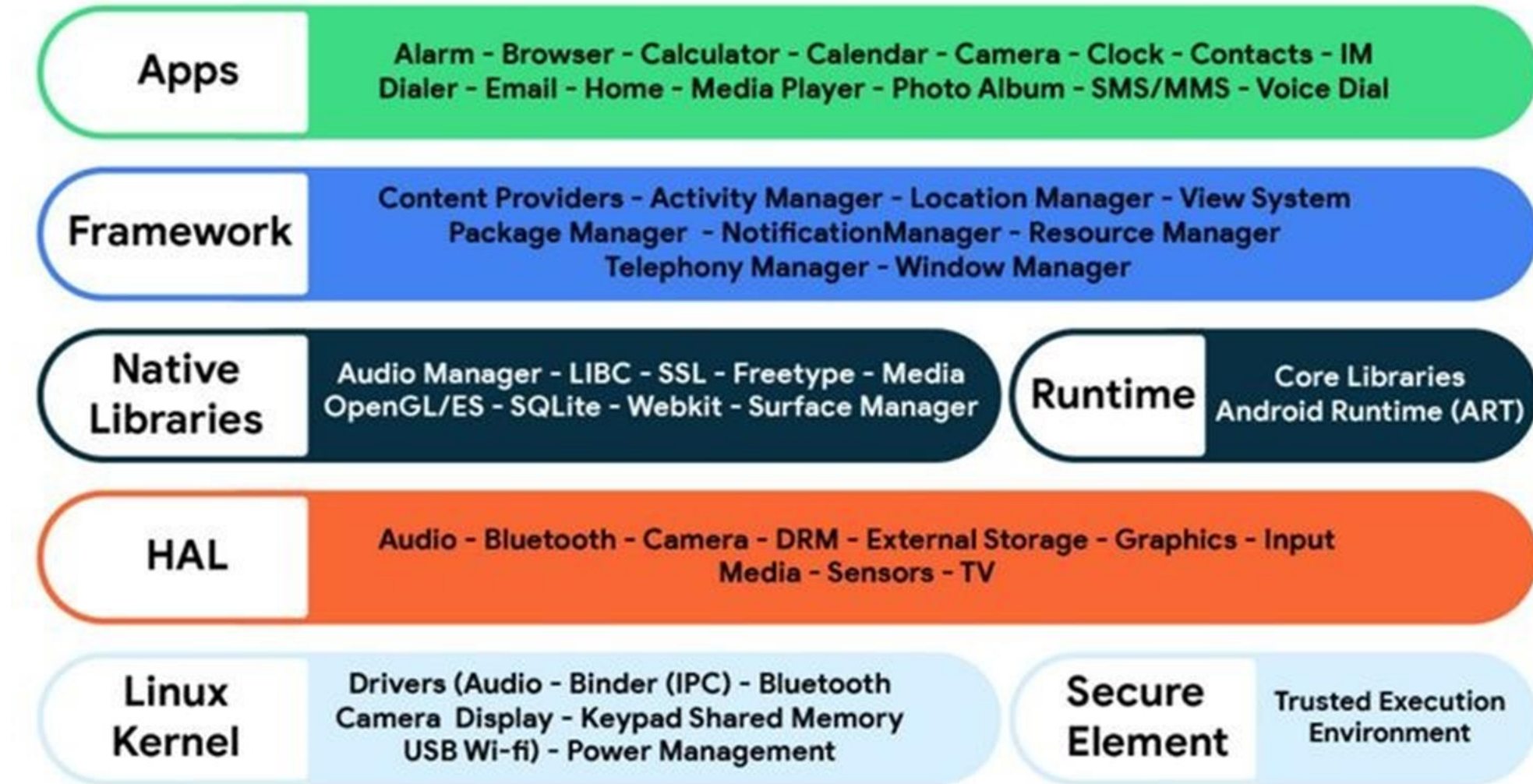
- Розробники Android написали власну “легку” Java VM і середовище виконання, оптимізоване для мобільних пристроїв
 - Перша версія, що застосовувалась до Android 4.4, називалась Dalvik
 - Нова версія середовища виконання називається ART (Android RunTime)
- Оптимізована архітектура VM включає специфічні для мобільних пристроїв риси, такі як поділювана пам’ять, швидкий старт, а також втрачає найбільш “важкі” вимоги Java VM (на жаль, це включає і суттєву частину безпеки)

4. Пісочниці

- Програми, що працюють на Android, підписуються та ізолюються в пісочницях, пов’язаних із їх підписом
 - Пісочниця визначає привілеї, доступні для програми
- Програми, як правило, побудовані для виконання в середовищі виконання Android та взаємодії з операційною системою через фреймворк, що описує системні служби, інтерфейси програмування платформи (API) та формати повідомлень
- Різноманітні мови високого та нижчого рівня, такі як Java, Kotlin та C/C ++, дозволені та працюють в одній і тій же пісочниці



Програмний стек Android





Безпека за дизайном (огляд)₁

726

- ❑ Android використовує апаратні та програмні засоби захисту для досягнення надійного захисту
- ❑ Безпека починається на апаратному рівні, де користувач автентифікується за допомогою облікових даних блокування екрана
- ❑ Перевірене завантаження (**Verified Boot**) гарантує, що системне програмне забезпечення не було підроблено, а апаратне шифрування та обробка ключів допомагають захистити дані при передачі та в стані зберігання
- ❑ Захист, вбудований на програмному рівні, є надзвичайно важливим для забезпечення безпеки пристроїв Android
- ❑ **Google Play Protect** — це найпоширеніша у світі служба виявлення загроз, яка щодня активно сканує понад 50 мільярдів програм на пристроях для відстеження шкідливої поведінки
 - **Play Protect** сканує всі програми, включаючи загальнодоступні програми від Google Play, системні програми, оновлені виробниками та постачальниками, і програми зі сторонніх джерел



Безпека за дизайном (огляд)₂

727

- ❑ Пісочниця ізолює та захищає програми Android, перешкоджаючи доступу шкідливих програм до приватної інформації
- ❑ Android також захищає доступ до внутрішніх компонентів операційної системи, щоб запобігти експлуатації вразливостей
- ❑ Обов'язкове постійне шифрування допомагає захистити дані, навіть якщо пристрої потрапляють у чужі руки
 - Шифрування захищено ключами **Keystore**, які зберігають криптографічні ключі в контейнері, що ускладнює вилучення з пристрою
 - Розробники можуть безпечно та легко використовувати **Android KeyStore** із **Jetpack Security**
 - **Adiantum** надає можливості шифрування для пристроїв з меншим рівнем потужності, які не мають апаратної реалізації інструкцій **Advanced Encryption Standard (AES)** у процесорі, що потенційно дозволяє компаніям використовувати дешевші пристрої без втрати криптографічної безпеки



Апаратно-забезпечена безпека



Апаратно-забезпечена безпека

729

- ☐ Перевірене завантаження (**Verified Boot**)
- ☐ Довірене середовище виконання (**Trusted Execution Environment**)
- ☐ Система зберігання ключів (KeyStore) Android
 - Атестація ключів KeyStore
 - “Брелок” (**KeyChain**)
 - Jetpack Security
 - Розшифровка ключа на розблокованих пристроях
 - Прив'язка версії (**Version Binding**)
- ☐ Підтримка обладнання, що захищає від втручання (**Tamper-resistant**)
 - Біометрія
 - Автентифікація за відбитками пальців
 - Автентифікація обличчя
 - Додаткові методи автентифікації
 - Захищене підтвердження (**Protected Confirmation**)



Verified Boot₁

730

- ❑ Підтверджене завантаження (**Verified Boot**) — це безпечний процес завантаження Android, який перевіряє системне програмне забезпечення перед його запуском
 - Це ускладнює збереження програмних атак при перезавантаженні та забезпечує користувачам безпечніший стан під час завантаження
- ❑ Кожен етап перевіреного завантаження має криптографічний підпис
 - Кожна фаза процесу завантаження перевіряє цілісність наступної фази перед виконанням цього коду
 - Починаючи з Android 7.0, повне завантаження сумісного пристрою із заблокованим завантажувачем триває, лише якщо ОС задовольняє перевірку цілісності
- ❑ Стан перевіреного завантаження використовується як вхід у процесі отримання ключів шифрування диска
 - Якщо стан перевіреного завантаження змінюється (наприклад, користувач розблоковує завантажувач), захищене обладнання забороняє доступ до даних, що використовуються для отримання ключів шифрування диска, які використовувались, коли завантажувач був заблокований



Verified Boot₂

731

- ❑ Підтверджене завантаження на сумісних пристроях під управлінням Android 9.0 і новіших версій вимагає захисту від відкату (**rollback**)
 - Це означає, що компрометація ядра (або фізична атака) не може помістити стару, більш уразливу версію ОС в систему та завантажити її
 - Крім того, стан захисту від відкату зберігається в захищеному сховищі (**tamper-evident storage**)
- ❑ Підприємства можуть перевірити стан перевіреного завантаження за допомогою атестації ключів KeyStore
 - Атестація ключів KeyStore потрібна для сумісних пристроїв, що постачаються з Android 8.0 і новішими версіями



Trusted Execution Environment

❑ Пристрої Android, які підтримують екран блокування, мають вторинне, ізольоване середовище, яке називається **Trusted Execution Environment (TEE)**

- Це дозволяє подальше відокремлення від будь-якого ненадійного коду
- Можливості, як правило, реалізуються з використанням безпечного обладнання, такого як технологія **ARM TrustZone®**

❑ **TEE** відповідає за деякі операції на пристрої, найкритичніші з міркувань безпеки, включаючи:

- Перевірка коду блокування екрана: доступна на пристроях, які підтримують захищений екран блокування та поставляються з Android 7.0 і новішими версіями. Перевірка блокованого екрану надається TEE, якщо немає ще більш безпечного середовища
- Відповідність шаблону відбитків пальців: доступна на пристроях, які мають датчик відбитків пальців і поставляються з Android Marshmallow 6.0 і вище
- Захист та керування ключами KeyStore: доступно на пристроях, що підтримують захищений екран блокування і поставляються з Android 7.0 і вище
- Захищене підтвердження: використовує захищений апаратним забезпеченням користувальницький інтерфейс, який називається Trusted UI, щоб забезпечити високу надійність критичних транзакцій, доступних на пристроях під управлінням Android 9.0 і новіших версій
- Управління цифровими правами (DRM): розширюваний фреймворк, що дозволяє програмам керувати контентом, захищеним правами, відповідно до ліцензійних обмежень, пов'язаних із контентом



733

Android Keystore System



Android Keystore System₁

734

- ❑ Система **Android KeyStore** є основою захисту даних на пристроях
 - Зберігає криптографічні ключі в контейнері, що ускладнює вилучення їх із пристрою
 - Змушує програми визначати дозволене використання своїх ключів, а потім застосовує ці обмеження поза процесами цих програм
 - Це зменшує можливість несанкціонованого використання ключових матеріалів на пристрої Android
- ❑ Починаючи з Android 7.0 — атестація ключів (**Keymaster 2**)
 - дозволяє службам поза пристроєм перевіряти, що ключі, що використовуються в програмах, зберігаються в сховищі ключів, що підтримується апаратним забезпеченням
- ❑ Починаючи з Android 8.0 — атестація ідентифікаторів (**Keymaster 3**)
 - дозволяє пристрою надати підтвердження своїх апаратних ідентифікаторів, таких як серійний номер або IMEI
- ❑ Пристрої з Android 9.0 і новіших версій використовують апаратно підтримуваний **Keymaster 4**
 - пропонує додатковий захист від фальсифікації



Android Keystore System₂

735

❑ KeyStore підтримує

- симетричні криптографічні примітиви:
 - AES (Advanced Encryption Standard)
 - HMAC (Keyed-Hash Message Authentication Code)
- асиметричні криптографічні алгоритми

❑ Елементи керування доступом визначаються під час генерації ключів і застосовуються протягом усього терміну дії ключа

- Ключі можуть бути обмежені:
 - для використання лише після автентифікації користувача
 - лише для визначених цілей
 - або із зазначеними криптографічними параметрами

❑ Для пристроїв, які підтримують захищений екран блокування та постачаються з Android 7.0 або новішою версією, KeyStore повинен бути реалізований у захищеному обладнанні

- Це гарантує, що навіть у випадку компрометації ядра ключі KeyStore не можна витягти із захищеного обладнання



Атестація ключів KeyStore

736

- ❑ Атестація ключів надає серверу можливість отримати впевненість у властивостях ключів
 - Атестацію ключів підтримують сумісні пристрої, що постачаються з Android 8.0 і вище
- ❑ Пристрої, що підтримують **Google Play**, оснащують на заводі ключем атестації, згенерованим Google
 - Захищене обладнання на таких пристроях може підписувати за допомогою ключа атестації специфікації, що засвідчують властивості ключів, захищених захищеним обладнанням
 - наприклад, те, що ключ створено і не може залишити захищене обладнання
- ❑ Атестація ключів краще захищає розташування важливих властивостей пристрою, таких як версія ОС, рівень виправлення та проходження перевіреного завантаження (**Verified Boot**)



KeyChain (“Брелок”)

- ❑ Клас **KeyChain** забезпечує доступ до приватних ключів та відповідних ланцюжків сертифікатів у сховищі облікових даних
- ❑ **KeyChain** часто використовується Chrome, віртуальними приватними мережами (VPN) та корпоративними програмами для доступу до ключів, імпортованих користувачем або програмою управління мобільними пристроями
- ❑ KeyStore призначений для неподілюваних ключів для конкретних програм, **KeyChain** — для ключів, які призначені для спільного використання в профілі
 - Наприклад, агент управління мобільними пристроями може імпортувати ключ, який Chrome використовуватиме для корпоративного веб-сайту
- ❑ Android 10 представляє кілька вдосконалень API **KeyChain**
 - Коли програма викликає **KeyChain.choosePrivateKeyAlias**, пристрої тепер фільтрують список сертифікатів, які користувач може вибрати на основі емітентів та ключових алгоритмів, зазначених у дзвінку
 - **KeyChain** більше не вимагає від пристрою блокування екрана перед тим, як імпортувати ключі або сертифікати центру сертифікації (CA)



Jetpack Security

738

- ❑ Розробники можуть використовувати Android KeyStore за допомогою **Jetpack Security**
- ❑ **MasterKeys** дозволяє розробникам створювати безпечний ключ AES 256 GCM нестандартно або для випадків розширеного використання, які визначають параметри для контролю авторизації ключа
- ❑ **Jetpack Security** також забезпечує криптографічні абстракції вищого рівня для шифрування файлів (**EncryptedFile**) та **SharedPreferences** (**EncryptedSharedPreferences**)
- ❑ Рекомендується використовувати **Jetpack Security** усіма контролерами **Device Policy (DPC)**, які контролюють локальні політики пристроїв та системні програми на пристроях, корпоративні програми, загальнодоступні та приватні програми



Розшифрування ключів на розблокованих пристроях

739

- ☐ Android 9.0 впровадив прапор **unlockedDeviceRequired**
 - Цей параметр визначає, чи вимагає KeyStore розблокування екрану перед тим, як дозволити використання зазначеного ключа
- ☐ Ці типи ключів добре підходять для шифрування конфіденційних даних для зберігання на диску, таких як дані про стан здоров'я або дані підприємства
- ☐ Прапор надає користувачам більшу впевненість у тому, що дані не можна розшифрувати, коли пристрій заблоковано у разі втрати або викрадення телефону



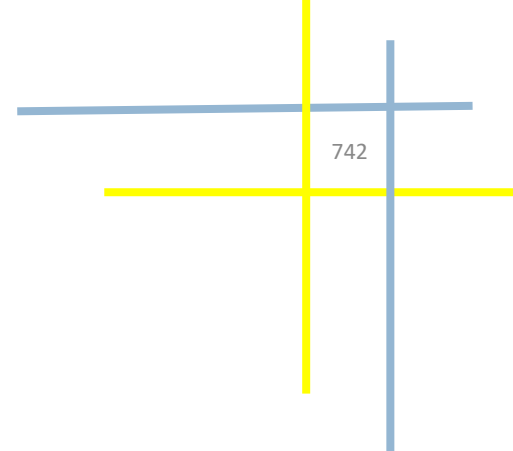
Прив'язка версії (Version Binding)

- ☐ У **Keymaster 2 і 3** всі ключі також прив'язані до операційної системи та рівня виправлення образу системи
- ☐ Це гарантує, що зловмисник, який виявив слабкі місця у старій версії системи або програмного забезпечення TEE, не зможе повернути пристрій до вразливої версії та використовувати ключі, створені з новою версією
- ☐ Крім того, коли на пристрої, який оновлено до нової версії або рівня виправлення, використовується ключ із заданою версією та рівнем виправлення, ключ модернізується перед тим, як його можна буде використовувати, а попередня версія ключа втрачає силу
 - Таким чином, у міру оновлення пристрою, ключі будуть просуватись вперед разом із пристроєм, але будь-яке повернення пристрою до попереднього випуску призведе до того, що ключі стануть непридатними



Підтримка обладнання, що захищає від втручання (Tamper-resistant hardware support)

- ☐ Починаючи з Android 8.0, сумісні пристрої можуть додатково використовувати обладнання, що захищає від втручання, для перевірки паролю блокування екрану
- ☐ У разі успішної перевірки це обладнання повертає секрет високої ентропії, який можна використовувати для отримання ключа шифрування диску



Біометрія



Біометрія₁

743

- ❑ В Android 9.0 і вище система **BiometricPrompt** API надає діалоги біометричної автентифікації, що можуть використовуватися від імені програми
 - Це створює відповідний вигляд, відчуття та розміщення діалогового вікна та надає користувачам більшої впевненості у їх автентифікації за допомогою біометричних даних за допомогою надійного відстежувача облікових даних
- ❑ Цей API використовується разом із системою Android Keystore
 - Захист біометричних даних здійснюється за допомогою апаратного модуля захисту **Strongbox Keystore**, який надійно зберігає та обробляє криптографічні ключі на пристрої
- ❑ Пристрої можуть використовувати біометричну автентифікацію для захисту приватної інформації та важливих корпоративних даних, доступних через пристрої, що використовуються в корпоративних умовах
- ❑ **BiometricPrompt** API доступний розробникам для інтеграції біометричної автентифікації у свої програми



Біометрія₂

744

- ☐ Фреймворк Android включає біометричну автентифікацію обличчя та відбитків пальців
 - Android можна налаштувати для підтримки інших форм біометричної аутентифікації, таких як Iris
- ☐ Усі біометричні реалізації повинні відповідати вимогам безпеки та мати високий рейтинг, щоб брати участь у класі **BiometricPrompt**
- ☐ Методи розблокування на основі біометрії, як правило, оцінюються на основі коефіцієнта помилкового прийняття (**False Accept Rate — FAR**)
 - Android використовує дві додаткові метрики, щоб допомогти виробникам пристроїв оцінити свою безпеку: коефіцієнт прийняття самозванця (**Imposter Accept Rate — IAR**) та коефіцієнт прийняття підроблених даних (**Spoof Accept Rate — SAR**)
- ☐ Крім того, виробники пристроїв Android можуть отримати доступ до рекомендацій щодо найкращих практик системної безпеки щодо використання біометричної автентифікації
- ☐ Біометричні датчики класифікуються на основі рівня їх прийняття та підміни, а також безпеки біометричного конвеєра
- ☐ Доступна методологія тестування, яка допоможе виміряти реалізацію цих методів розблокування



Методи біометричної автентифікації₁

Автентифікація за відбитками пальців

- ❑ На пристроях із датчиком відбитків пальців користувачі можуть зареєструвати один або кілька відбитків пальців і використовувати ці відбитки пальців для розблокування пристрою та виконання інших завдань
- ❑ Android використовує мову визначення інтерфейсу апаратного забезпечення відбитків пальців (**Hardware Interface Definition Language — HIDL**) для підключення до бібліотеки та обладнання конкретного постачальника, наприклад датчика відбитків пальців
- ❑ Android підтримує Gatekeeper для автентифікації за відбитками пальців та PIN/шаблоном/паролем
 - Підсистема Gatekeeper виконує цю автентифікацію у довіреному середовищі виконання, реєструючи та перевіряючи паролі за допомогою коду автентифікації повідомлень на основі хеш-коду (**Hash-Based Message Authentication Code — HMAC**) за допомогою апаратно-захищеного секретного ключа
- ❑ Документ визначення сумісності Android визначає вимоги щодо впровадження біометричної безпеки



Методи біометричної автентифікації₂

Автентифікація обличчя

- ❑ Автентифікація обличчя дозволяє користувачам розблокувати свій пристрій, просто подивившись на передню частину пристрою
- ❑ Android 10 додає підтримку нового стека автентифікації обличчя, який може безпечно обробляти кадри камери, зберігаючи безпеку та конфіденційність під час автентифікації обличчя на підтримуваному обладнанні
- ❑ Android 10 також пропонує метод для реалізації, що відповідає безпеці, щоб забезпечити інтеграцію у програми для транзакцій, такі як Інтернет-банкінг або інші послуги



Методи біометричної автентифікації₃

Додаткові методи автентифікації

- ❑ Android підтримує фреймворк Trust Agent для розблокування пристрою
- ❑ **Google SmartLock** використовує цей фреймворк, щоб дозволити пристрою залишатися розблокованим до тих пір, поки він залишатиметься з користувачем, що визначається певною присутністю користувача чи іншими сигналами
 - Однак **SmartLock** не відповідає такому рівню надійності, як інші методи розблокування на Android, і йому не дозволено розблоковувати ключі KeyStore, пов'язані з автентифікацією
- ❑ Організації можуть вимкнути довірчих агентів, використовуючи прапор **KEYGUARD_DISABLE_TRUST_AGENTS** у політиках EMM



Захищене підтвердження (Protected Confirmation)

748

- ☐ **Android Protected Confirmation** використовує апаратно захищений користувальницький інтерфейс (**Trusted UI**) для здійснення критичних транзакцій поза операційною системою на пристроях з ОС Android 9.0 або новішої версії
 - Це захищає операції від шахрайських програм або скомпрометованої операційної системи
- ☐ Коли програма викликає захищене підтвердження, керування передається довіреному інтерфейсу, де відображаються дані транзакцій та отримується підтвердження користувачем правильності даних
- ☐ Після підтвердження намір криптографічно підтверджується та захищається від фальсифікації, коли його передають надійній стороні
- ☐ Загалом транзакція має вищий захист та безпеку порівняно з іншими формами вторинної автентифікації
- ☐ Існує кілька прикладів використання (use cases) захищеного підтвердження Android, такі як грошові перекази від особи до особи, автентифікація користувача чи інновації, такі як підтвердження правильних ін'єкцій інсулінової помпи

Донецький національний технічний університет



Тема 9. Архітектура безпеки і механізми захисту Android. Лекція 20. Безпека ОС Android

Предмет: БЗОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

750

- ☐ Безпека ОС Android
- ☐ Конфіденційність користувачів та даних
- ☐ Безпека мережі
- ☐ Безпека прикладних програм



Безпека операційної системи (Operating System Security)

751

- ❑ Android використовує багат шаровий захист операційної системи
 - З кожною версією Android операційна система додатково зміцнюється, щоб мати належний захист від постійних загроз, з якими стикаються підприємства
- ❑ Цілісність пристрою (**Device Integrity**)
- ❑ Функції цілісності пристрою захищають мобільний пристрій від скомпрометованої та(або) несанкціоновано модифікованої операційної системи
- ❑ Android приймає кілька заходів для гарантування цілісності пристрою, включаючи:
 - Пісочницю (**sandboxing**)
 - **SELinux**
 - **Seccomp**
 - Права доступу Unix
 - Рівень апаратної абстракції (**Hardware Abstraction Layer – HAL**)



Sandboxing

752

- ❑ Android запускає всі програми всередині пісочниць, щоб запобігти зловмисному або помилковому коду програм, не порушуючи інші програми або компоненти системи
- ❑ Оскільки пісочниця програми застосована в ядрі, вона охоплює всю програму незалежно від конкретного середовища розробки, API або використовуваної мови програмування
- ❑ За замовчуванням програми не можуть взаємодіяти між собою та мають обмежений доступ до операційної системи
- ❑ Подібним чином, системні компоненти працюють у найменш привілейованих пісочницях, щоб запобігти впливу компрометації в одному компоненті на інші
- ❑ Наприклад, віддалено доступні компоненти, такі як медіа-сервер та **WebView**, ізольовані у власній обмеженій пісочниці



SELinux

753

- ❑ Android використовує **Security-Enhanced Linux (SELinux)** для забезпечення мандатного керування доступом (MAC) над усіма процесами, включаючи ті, що мають права root/суперкористувача
- ❑ SELinux дозволяє Android краще захищати та обмежувати системні служби, контролювати доступ до даних програм та системних журналів, зменшувати вплив шкідливого програмного забезпечення та захищати користувачів від потенційних вад коду на мобільних пристроях
- ❑ SELinux працює за принципом відмовлення за умовчанням: все, що явно не дозволено, забороняється
- ❑ Android включає SELinux та відповідну політику безпеки для компонентів в AOSP
- ❑ Заборонені дії запобігаються, а всі спроби порушень реєструються за допомогою інструментів Linux: **dmesg** друкує буфер повідомлень ядра, а **logcat** — це інструмент командного рядка, який вивантажує журнал системних повідомлень
- ❑ З архітектурою системи Android, SELinux використовується для забезпечення розділення між фреймворком Android та компонентами постачальника, специфічними для пристрою, так що вони працюють у різних процесах і взаємодіють між собою через набір дозволених інтерфейсів постачальників, реалізованих як шари абстракції апаратури (HAL)



Фільтр Seccomp

754

- ❑ У поєднанні з SELinux Android використовує Seccomp для подальшого обмеження доступу до ядра, блокуючи доступ до певних системних викликів
 - Починаючи з Android 7.0, Seccomp застосовувався до процесів у медіа-фреймворках
 - Починаючи з Android 8.0, фільтр Seccomp застосовується до всіх програм, застосовуючи білий список дозволених системних викликів
- ❑ Програми можуть додатково забезпечити власний фільтр сескомп для подальшого зменшення набору дозволених системних викликів



Права доступу Unix

755

- ☐ Android використовує права доступу Linux/Unix для подальшої ізоляції ресурсів програм
- ☐ Android призначає унікальний ідентифікатор користувача (UID) кожній програмі та запускає кожного користувача в окремому процесі
- ☐ За замовчуванням програми не можуть отримувати доступ до файлів або ресурсів одна одної так само, як різні користувачі в Linux ізольовані один від одного



Anti-exploitation

756

- ❑ Android пропонує захист від експлуатації, такий як **Control Flow Integrity** і **Integer Overflow Sanitization**
- ❑ На основі компілятора були додані нові зменшення загроз, щоб ускладнити використання помилок та запобігти перетворенню певних класів помилок у вразливості
- ❑ Це розширює зменшення загроз, що існують, компілятором, який спрямовує операції часу виконання на безпечне переривання процесів, коли виникає невизначена поведінка



Конфіденційність користувачів та даних



Конфіденційність користувачів та даних₁

- ☐ Захист конфіденційності користувачів є фундаментальним для Android
- ☐ В ОС Android 9.0 основні аспекти конфіденційності включили:
 - обмеження доступу фонових програм до сенсорів пристроїв,
 - обмеження інформації, отриманої при скануванні Wi-Fi,
 - нові правила дозволів та групи дозволів, пов'язані з телефонними дзвінками, станом телефону та сканування Wi-Fi
- ☐ Ці зміни стосуються всіх програм, що працюють на Android 9, незалежно від цільової версії SDK
- ☐ Android 10 розширює конфіденційність та засоби керування, які користувачі мають над даними та можливостями програм
 - Загалом вони надають користувачам та IT-адміністраторам кращу ясність щодо того, як можна отримати доступ до даних та розташування користувачів
- ☐ Розробникам рекомендується забезпечити, щоб їх програми відповідали останнім змінам конфіденційності
- ☐ Android 10 встановлює обмеження на доступ до даних та ідентифікаторів системи, доступ до інформації про камери та мережі та вносить кілька змін до моделі дозволів



Конфіденційність користувачів та даних₂

- ❑ Робочий профіль створює окремий автономний профіль на пристроях Android, який ізолює корпоративні дані від особистих програм та даних
 - Це можна додати до персональних пристроїв у налаштуваннях BYOD або на корпоративному пристрої, що використовується як для роботи, так і для особистих цілей
 - За допомогою цього окремого профілю особисті програми та дані користувача не підпадають під контроль ІТ
 - Для забезпечення чіткої видимості користувача, коли робочий профіль застосовується до пристрою, DPC представляє умови використання та деталізує дані, які фіксуються та записуються
 - Користувач повинен переглянути та прийняти ліцензійну угоду користувача для налаштування робочого профілю
- ❑ Користувачі можуть переглянути налаштування робочого профілю через
Налаштування --> Облікові записи (Settings --> Accounts)



Обмеження доступу до ідентифікаторів пристрою

760

- ☐ Пристрої, що працюють під управлінням **Android 8.0** і новіших версій, використовують випадкові MAC-адреси при зондуванні нових мереж, якщо в даний час вони не пов'язані з мережею
- ☐ На **Android 9.0** пристрій може використовувати рандомізовану MAC-адресу під час підключення до мережі Wi-Fi, якщо це дозволено опцією розробника
- ☐ В **Android 10** система передає рандомізовані MAC-адреси за умовчанням
- ☐ Крім того, доступ до IMEI пристрою та серійних номерів неможливий



Location control

761

- ❑ Програми можуть надавати відповідну інформацію користувачеві, використовуючи API розташування
 - Наприклад, якщо програма допомагає користувачеві орієнтуватися на маршруті доставки, їй потрібно постійно отримувати доступ до місця розташування пристрою, щоб надати потрібну допомогу
- ❑ Розташування корисно в багатьох сценаріях — Android надає розробникам інструменти для запиту необхідних дозволів, одночасно надаючи користувачам можливість вибору, що саме вони дозволяють
- ❑ Програми, які використовують служби локації, повинні запитувати дозволи на отримання розташування, щоб користувач мав видимість і контроль над цим доступом
- ❑ В Android 10 користувачі бачать діалогове вікно, що повідомляє їх про те, що програма бажає отримати доступ до їх розташування
 - Цей запит може бути на доступ лише під час використання програми або постійно
 - Користувач може дозволити програмі постійний доступ до місця розташування пристрою
 - Коли програма звертається до місця розташування пристрою у фоновому режимі вперше після того, як користувач зробив такий вибір, система планує надіслати користувачеві повідомлення
 - Це сповіщення нагадує користувачеві, що він дозволив програмі постійно отримувати доступ до місцезнаходження пристрою



Доступ до зовнішнього сховища

762

- ❑ Щоб надати користувачам більше контролю над своїми файлами та обмежити захаращення файлами, програмам, орієнтованим на Android 10 і новіших версій, за замовчуванням надається обмежений доступ до зовнішнього пристрою зберігання даних
 - Такі програми можуть бачити лише свій каталог для конкретної програми, доступ до якого здійснюється за допомогою `Context.getExternalFilesDir()`, а також певні типи носіїв
- ❑ Розробникам рекомендується використовувати такий доступ як найкращу практику
- ❑ Адміністратори EMM можуть не дозволяти користувачам своєї організації отримувати доступ до зовнішнього сховища, такого як SD-карта, підключена до їх пристрою, щоб додатково зменшити потенціал будь-якої втрати даних



Обмежений доступ до фонових сенсорів

- ❑ Android 9.0 обмежує можливість фонових програм отримувати доступ до даних користувача та даних сенсора
- ❑ Якщо програма працює у фоновому режимі на пристрої під управлінням Android 9.0 і новіших версій, система застосовує до програми такі обмеження:
 - Програма не може отримати доступ до мікрофона або камери
 - Сенсори, які використовують режим безперервної звітності, такі як акселерометри та гіроскопи, не отримують подій
 - Сенсори, які використовують режими звітності "за зміни" або "один знімок", не отримують подій
- ❑ Якщо програмі потрібно виявляти події сенсорів на пристроях з ОС Android 9.0, вона повинна використовувати службу переднього плану (**foreground**)



Режим блокування (Lockdown mode)

- ☐ Користувач може ввімкнути функцію блокування, щоб додатково обмежити доступ до пристрою
- ☐ Цей режим демонструє опцію кнопки живлення, яка вимикає **Smart Lock**, біометричне розблокування та сповіщення на екрані блокування
- ☐ Його можна ввімкнути через **Налаштування --> Налаштування екрану блокування --> Режим блокування (Settings --> Lock screen preferences --> Lockdown mode)**
- ☐ Адміністратори підприємства можуть віддалено заблокувати робочий профіль і вилучити ключ шифрування з пам'яті на корпоративних пристроях, використовуючи цю можливість



765

Безпека мережі



Безпека мережі

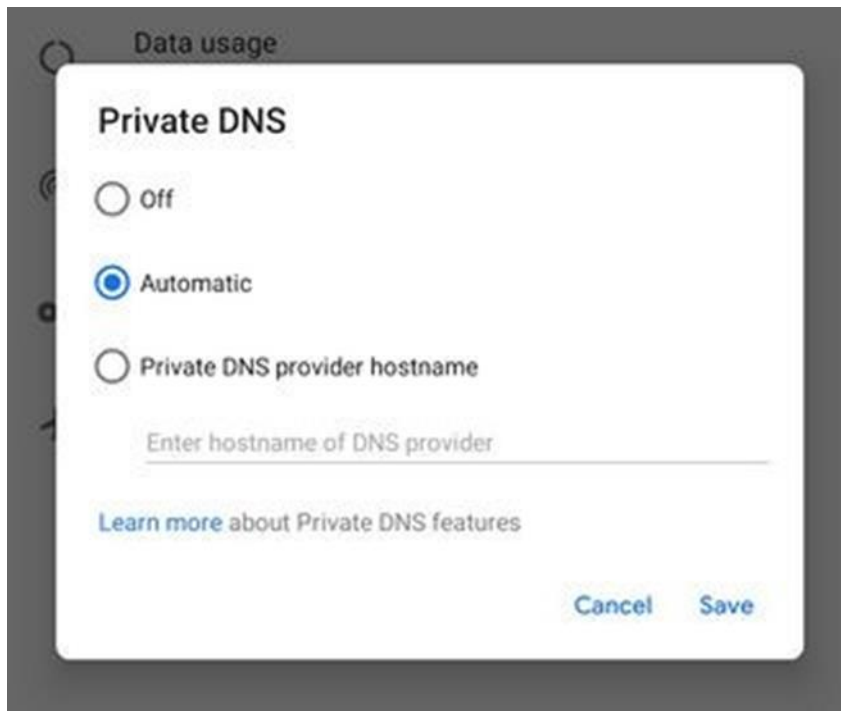
766

- ❑ На додаток до захисту даних у стані спокою — захисту інформації, що зберігається на пристрої — Android забезпечує мережну безпеку для захисту даних під час передачі даних, що надсилаються на пристрої Android та з них
- ❑ Android забезпечує безпечний обмін через Інтернет для перегляду веб-сторінок, електронної пошти, обміну миттєвими повідомленнями та інших Інтернет-програм, підтримуючи безпеку транспортного рівня (**Transport Layer Security — TLS**)
 - DNS over TLS (Android 9.0 та вище)
 - TLS за умовчанням
 - Wi-Fi VPN
 - Обробка сертифікатів



DNS over TLS

767



- ❑ Android 9.0 і вище включає вбудовану підтримку DNS через TLS
 - Користувачі або адміністратори можуть увімкнути режим приватного DNS у налаштуваннях мережі та Інтернету
- ❑ Android 10 розширює можливості адміністраторів для запобігання змінам налаштувань DNS користувачами, запобігаючи витоку запитів DNS
- ❑ Пристрої автоматично переходять на DNS через TLS, якщо настроєний DNS-сервер підтримує його, але користувачі можуть вимкнути його, якщо захочуть



TLS за умовчанням

768

- ❑ Android допомагає захистити дані, захищаючи мережний трафік, який надходить або виходить із пристрою, підтримуючи безпеку транспортного рівня (**Transport Layer Security — TLS**)
- ❑ В ОС Android 9.0 і новіших версій Конфігурація безпеки мережі за умовчанням блокує весь трафік з відкритим текстом (незашифрований HTTP)
 - Розробники повинні явно ввімкнути певні домени, щоб використовувати трафік з відкритим текстом у своїх програмах
 - Android Studio також попереджає розробників, коли їхня програма містить потенційно небезпечну конфігурацію безпеки мережі
- ❑ Щоб запобігти випадковому незашифрованому з'єднанню, атрибут маніфесту **android: usesCleartextTraffic** дозволяє програмам вказувати, що вони не мають наміру надсилати мережний трафік без шифрування
- ❑ Android 10 використовує TLS 1.3 за умовчанням для всіх з'єднань TLS
 - TLS 1.3 — це суттєвий перегляд стандарту TLS з перевагами продуктивності та підвищеною безпекою
 - TLS 1.3 також є більш конфіденційним, оскільки він шифрує більшу частину процесу рукописання та пропонує посилену безпеку, більше не підтримуючи сертифікати, підписані за допомогою SHA 1
 - Тестування швидкодії свідчать, що за допомогою TLS 1.3 захищені з'єднання можна встановити на 40 відсотків швидше порівняно з TLS 1.2



Wi-Fi₁

769

❑ Android 10 підтримує стандарти Wi-Fi Alliance Wi-Fi Protected Access версії 3 (**WPA3**) та Wi-Fi Enhanced Open

- WPA3 та Wi-Fi Enhanced Open покращують загальну безпеку Wi-Fi, забезпечуючи кращу конфіденційність та стійкість (robustness) проти відомих атак

❑ **WPA3** — це новий стандарт безпеки WFA для персональних та корпоративних мереж, що використовує сучасні алгоритми безпеки та надійніші набори шифрів

❑ **WPA3** має дві частини: персональну та корпоративну

- **WPA3-Enterprise** пропонує більш ефективні методи автентифікації та шифрування на каналному рівні та додатковий 192-розрядний режим безпеки для чутливих середовищ безпеки
- **WPA3-Personal** використовує одночасну автентифікацію рівних (**Simultaneous Authentication of Equals — SAE**) замість спільного ключа (**Pre-shared Key — PSK**), забезпечуючи користувачам більш надійний захист від таких атак, як offline атаки за словником, відновлення ключів та підробка повідомлень



Wi-Fi₂

770

- ❑ Wi-Fi Enhanced Open — це новий стандарт безпеки WPA для загальнодоступних мереж, заснований на “опортуністичному” бездротовому шифруванні (**Opportunistic Wireless Encryption — OWE**)
 - Він забезпечує шифрування та конфіденційність у відкритих мережах, не захищених паролем, у таких місцях як кафе, готелі, ресторани та бібліотеки
 - Enhanced Open не забезпечує автентифікацію
- ❑ Android також підтримує протокол WPA2-Enterprise (802.11i), розроблений для корпоративних мереж, і може бути інтегрований у широкий спектр серверів автентифікації служби віддаленої автентифікації (**Remote Authentication Dial-In User Service — RADIUS**)
 - Підтримка протоколу WPA2-Enterprise використовує шифрування з автентифікацією AES- 128-CCM
- ❑ В Android 10 QR-коди та дані NFC, що використовуються для авторизації пристрою, можуть містити конфігурацію та облікові дані розширеного протоколу автентифікації (**Extensible Authentication Protocol — EAP**), включаючи сертифікати
 - Коли особа сканує QR-код або натискає тег NFC, пристрій автоматично автентифікується у локальній мережі Wi-Fi за допомогою EAP і запускає процес авторизації без додаткового введення вручну



VPN₁

771

- ❑ Android підтримує безпечне підключення до корпоративної мережі за допомогою VPN: Always-on VPN, Per User VPN, Per Application VPN
- ❑ Постійно ввімкнена VPN (**Always-on VPN**) — VPN можна налаштувати так, щоб програми не мали доступу до мережі, доки не буде встановлено VPN-з'єднання, що перешкоджає програмам надсилати дані через інші мережі
 - В ОС Android 7.0 і вище, Always-on VPN підтримує клієнтів VPN, які реалізують
- ❑ **VpnService**. Система автоматично запускає VPN після завантаження пристрою
 - Завжди ввімкнену VPN можна активувати для програм у корпоративному середовищі за допомогою **DevicePolicyManager#setAlwaysOnVpnPackage**
 - Власники пристроїв та власники профілів можуть вимагати, щоб робочі програми завжди підключалися через зазначену VPN
 - Крім того, користувачі можуть вручну налаштувати постійні клієнти VPN, які реалізують методи **VpnService**, використовуючи **Settings --> More --> VPN**
 - Можливість увімкнути **Always-on VPN** у налаштуваннях доступна, лише якщо клієнт VPN націлений на рівень API 24 або вище



VPN₂

772

- ❑ VPN для кожного користувача (**Per User VPN**) — на багатокористувацьких пристроях VPN застосовуються для кожного користувача Android, тому весь мережний трафік маршрутизується через VPN, не зачіпаючи інших користувачів пристрою
 - VPN застосовуються до кожного робочого профілю, що дозволяє IT-адміністратору вказати, що через корпоративну мережу VPN-профілю проходить лише їхній мережний трафік підприємства, а не особистий мережний трафік користувача
- ❑ VPN для кожної програми (**Per Application VPN**) — Android 5.0 впровадила підтримку для забезпечення VPN-з'єднань у дозволених програмах та для запобігання VPN-з'єднанням у заборонених програмах
- ❑ В Android 10 програми VPN можуть встановлювати проксі HTTP для свого VPN-з'єднання
 - Перед викликом **VpnService.Builder.setHttpProxy()** програма VPN має налаштувати екземпляр **ProxyInfo** з хостом і портом
 - Система та багато мережних бібліотек використовують ці налаштування проксі-сервера, але система не змушує програми надсилати через проксі запити HTTP



Режими сервісу VPN

773

- ❑ Тепер програми VPN також можуть виявити, чи сервіс працює через постійно ввімкнену VPN та чи активний режим блокування (**Lockdown mode**)
- ❑ Нові методи, додані в Android 10, можуть допомогти розробникам налаштувати інтерфейс користувача
 - Наприклад, розробники можуть вимкнути кнопку відключення з'єднання в програмі VPN, коли постійно ввімкнена VPN контролює життєвий цикл сервісу

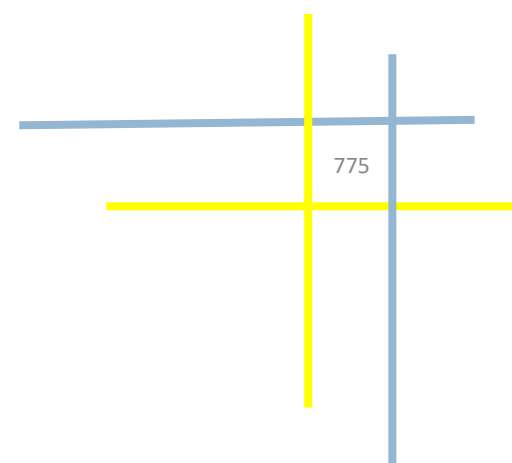
Режими блокування VPN

- ❑ Режими блокування дозволяють адміністраторам блокувати мережний трафік, який не використовує VPN, і звільняти програму, що дозволяє їй використовувати будь-яку доступну мережу, якщо мережа VPN відключена або недоступна
 - Адміністратори також можуть явно заборонити доступ до всіх мереж для програми, і це дозволяє лише спілкуватися через VPN



Користування сертифікатами

- ❑ Починаючи з ОС Android 7.0, усі нові пристрої мають поставлятися з одним і тим самим сховищем центру сертифікації (ЦС)
- ❑ Центри сертифікації є важливою складовою інфраструктури відкритих ключів, яка використовується для створення безпечних сеансів зв'язку за допомогою TLS
 - Визначення надійних центрів сертифікації, а зрештою, які цифрові сертифікати, підписані певним ЦС, є надійними — має вирішальне значення для безпечного спілкування через мережу
- ❑ Цей захист додатково покращується шляхом запобігання програмам, націленим на Android 9.0, дозволяти незашифровані з'єднання за умовчанням
 - Це слідує за різними змінами, зробленими роками для кращого захисту користувачів Android
- ❑ В ОС Android 7.0 та вище сумісні пристрої довіряють лише стандартизованим системним ЦС, що підтримуються в AOSP
 - Прикладні програми також можуть довіряти ЦС, доданим користувачами або адміністраторами
 - Довіру можна призначити всій програмі або лише для підключення до певних доменів
- ❑ Коли потрібні центри сертифікації для конкретного пристрою, наприклад, програма-оператор, якій потрібен безпечний доступ до компонентів інфраструктури оператора (наприклад, шлюзи SMS/MMS), ці програми можуть включати приватні центри сертифікації до самих компонентів/програм



Безпека прикладних програм



Безпека прикладних програм

- ❑ Прикладні програми є невід'ємною частиною будь-якої мобільної платформи, і користувачі все більше покладаються на мобільні прикладні програми для вирішення основних завдань продуктивності та спілкування
- ❑ Android надає декілька рівнів захисту програм, що дозволяє користувачам завантажувати програми для роботи чи особистого користування на свої пристрої, не забуваючи, що вони отримують високий рівень захисту від шкідливого програмного забезпечення, експлойтів та атак
 - **Підписування програм** — Android вимагає, щоб перед встановленням усі програми мали цифровий підпис ключем розробника
 - **Дозволи для програм** — центральний момент проектування архітектури безпеки Android полягає в тому, що жодна програма за умовчанням не має дозволу виконувати будь-які операції, які негативно впливають на інші програми, операційну систему або користувача
 - **Google Play Protect** — Постійно активна служба вбудована в усі пристрої, які мають Google Play, і захищає понад 2,5 мільярда пристроїв
 - **Перевірка програм Google Play** — перед тим, як програми стануть доступними в Google Play, вони проходять процес розгляду програм (review), щоб підтвердити їх відповідність політиці Google Play
 - **SafetyNet** — набір служб та API, які розробники можуть використовувати для захисту програм від загроз безпеці



Підписування програм

- ❑ Android вимагає, щоб перед встановленням усі програми мали цифровий підпис ключем розробника
 - Ротація ключів APK, що підтримується в Android 9.0, дає програмам можливість змінювати ключ підпису в рамках оновлення APK
 - Для підтримки ротації ключів схему підпису APK оновлено з v2 до v3, щоб дозволити використовувати старі та нові ключі
- ❑ Android використовує відповідний сертифікат для ідентифікації автора програми
 - Коли система встановлює оновлення для програми, вона порівнює сертифікат у новій версії з сертифікатом у версії, що встановлена, та дозволяє оновлення, якщо сертифікат відповідає
- ❑ Android дозволяє програмам, підписаним тим самим ключем, працювати в одному процесі, якщо програми цього вимагають, так що система розглядає їх як єдину програму
 - Android забезпечує застосування дозволів на основі підписів, щоб програма могла відкрити функціональність іншій програмі, підписаній тим самим ключем
 - Підписуючи кілька програм одним і тим же ключем і використовуючи дозволи на основі підписів, програма може безпечно ділитися кодом та даними



Дозволи для програм₁

778

- ❑ Дозволи захищають конфіденційність користувачів Android і забезпечують прозорість щодо того, до яких ресурсів чи інформації програми хочуть отримати доступ
 - Щоб програми мали доступ до системних функцій, таких як камера та Інтернет, або до даних користувача, таких як контакти та SMS, програма Android має явно запитувати дозвіл
 - Ці запити на дозвіл розроблені таким чином, щоб користувач мав чітке розуміння запиту та можливість його схвалити або відхилити
- ❑ Центральним моментом архітектури безпеки Android є те, що жодна програма за умовчанням не має дозволу на виконання будь-яких операцій, які негативно впливають на інші програми, операційну систему чи користувача
 - Це включає читання або записування особистих даних користувача (наприклад, контактів чи електронних листів), читання або записування файлів іншої програми, виконання доступу до мережі, утримання пристрою від режиму сну та ін.



Дозволи для програм₂

779

❑ Програми, націлені на рівень API 23 (Android 6.0) і вище, використовують дозволи під час виконання

- Ці діалогові вікна вимагають від користувача надання доступу до зазначеного дозволу
- Цей підхід спрощує процес встановлення та оновлення програми, оскільки користувачеві не потрібно надавати дозволи під час встановлення або оновлення програми
- Це також дає користувачеві більший контроль над функціональністю програми; наприклад, користувач міг би надати програмі камери доступ до камери, але не до розташування пристрою
- Користувачі можуть відкликати дозволи в будь-який час, навіть якщо програма націлена на нижчий рівень API

❑ Android 8.0 і вище також включають вдосконалення, що дають користувачам кращий контроль над використанням ідентифікаторів

- Стійкі ідентифікатори пристроїв, які впливають на конфіденційність, або більше недоступні, або закриті за дозволом під час виконання
- Наприклад, API, які мають доступ до MAC-адреси Wi-Fi, були видалені, за винятком повністю керованих пристроїв

❑ На корпоративних пристроях програми керування пристроями (DPC) можуть відмовляти у дозволах від імені користувача за допомогою **setPermissionPolicy API**, функції керованої Google Play



Google Play Protect

780

❑ **Google Play Protect** (Захист Google Play) — це потужна служба виявлення загроз, яка активно контролює пристрій для захисту його, його даних та програм від шкідливого програмного забезпечення

- Постійно активна служба вбудована в усі пристрої, які мають Google Play, і захищає понад 2,5 мільярда пристроїв

❑ Служба **Google Play Protect Verify Apps** раз на день перевіряє пристрої на наявність шкідливої поведінки або ризиків для безпеки

- Якщо ця служба виявляє програму, що містить шкідливе програмне забезпечення, вона сповіщає користувача
- Google Play Protect також може автоматично видаляти або вимикати шкідливі програми в рамках своєї ініціативи запобігання та використовувати зібрану інформацію для покращення виявлення потенційно шкідливих програм (**Potentially Harmful Applications — РНА**)
- Крім того, користувач може вибрати опцію, щоб невідомі програми надсилалися в Google для аналізу



Перевірка програм Google Play1

- ❑ **Play Store** має політику щодо захисту користувачів від зловмисників, які намагаються поширювати РНА
- ❑ Розробників перевіряють у два етапи
 - Вони вперше перевіряються, коли вони створюють свій обліковий запис розробника на основі їхнього профілю та кредитних карт
 - Потім розробники перевіряються з використанням додаткових сигналів після надсилання програми
- ❑ Перш ніж програми стануть доступними в Google Play, вони проходять процедуру перевірки (**review**), щоб підтвердити їх відповідність політиці Google Play
 - Google розробила автоматизований аналізатор ризиків програм, який виконує статичний та динамічний аналіз файлів **.apk** для виявлення потенційно шкідливої поведінки програм
 - Коли аналізатор ризиків у програмах (**Google's application risk analyzer**) виявляє щось підозріле, він позначає програму, яка порушує правила, і передає її аналітику з питань безпеки для перевірки вручну
 - Google призупиняє облікові записи розробників, які порушують програмну політику розробників



Перевірка програм Google Play₂

- ❑ Розробники негайно повідомляють, якщо його програму відмічено через проблему безпеки
 - Вони отримують подробиці про те, як покращити програму, і посилання на деталі сторінки підтримки для отримання додаткових вказівок
 - Це сповіщення зазвичай містить терміни доставки покращення
 - У деяких випадках необхідно покращити безпеку програм, перш ніж розробник зможе публікувати будь-які подальші оновлення
- ❑ Ще одним ключовим елементом мінімізації ризику є використання оновлених API
 - Заохочення розробників використовувати найновіші API підтримує підтримку найновіших функцій, створюючи оптимальну безпеку та продуктивність
 - Щоб відповідати вимогам API, нові програми та їх оновлення мають бути націлені принаймні на Android 9.0 або на рівень API 28
- ❑ Кожна нова версія Android вносить зміни, які приносять значні поліпшення безпеки та продуктивності — та покращують загальний досвід користування Android
 - Деякі з цих змін стосуються лише програм, які явно заявляють про підтримку через атрибут маніфесту **targetSdkVersion**, також відомий як цільовий рівень **API (target API level)**



SafetyNet₁

783

❑ **SafetyNet** — це набір служб та API, які розробники можуть використовувати для захисту програм від загроз безпеці

- Вони можуть пом'якшити втручання в пристрої, неправильні URL-адреси, PHA та підроблених користувачів

❑ **SafetyNet Attestation API** — це API проти зловживань, який дозволяє розробникам програм оцінювати пристрій Android, на якому працює їхня програма

- Цей API забезпечує криптографічно підписану атестацію, що оцінює цілісність пристрою
- Щоб створити атестацію, API вивчає програмне та апаратне середовище пристрою, шукає проблеми цілісності та порівнює їх із довідковими даними для затверджених пристроїв Android
- Згенерована атестація пов'язана з нонсом, який програма надає
- Атестація також містить позначку часу генерації та метадані про програму, що атестацію запросила

❑ **SafetyNet Safe Browsing API** пропонує послуги для визначення того, чи URL-адреса була позначена Google як відома загроза

- SafetyNet реалізує клієнта для Мережного протоколу безпечного перегляду v4 (Safe Browsing Network Protocol v4), розробленого Google
- І клієнтський код, і мережний протокол v4 були розроблені для збереження конфіденційності користувачів та мінімізації споживання акумулятора та пропускну здатності
- Розробники можуть використовувати цей API, щоб у повній мірі скористатися перевагами служби безпечного перегляду Google на Android найбільш оптимізованим для ресурсів способом і без реалізації мережного протоколу



- ❑ Сервіс **SafetyNet** також містить **SafetyNet reCAPTCHA API**, який захищає програми від шкідливого трафіку
 - Цей API використовує вдосконалений механізм аналізу ризиків для захисту програм від спаму та інших образливих дій
 - Якщо служба підозрює, що користувач, який взаємодіє з програмою, може бути ботом, а не людиною, він надає CAPTCHA, яку людина повинна вирішити, перш ніж програма зможе продовжувати виконання
- ❑ **SafetyNet Verify Apps API** дозволяє програмі взаємодіяти програмно з **Google Play Protect**, щоб перевірити, чи не встановлені відомі потенційно шкідливі програми
 - Якщо програма обробляє конфіденційні дані користувача, наприклад фінансову інформацію, розробники повинні підтвердити, що поточний пристрій захищений від шкідливих програм і вільний від програм, які можуть видавати себе за нього або виконувати інші шкідливі дії
 - Якщо безпека пристрою не відповідає мінімальній позиції безпеки, розробники можуть вимкнути функціональні можливості програми, щоб зменшити небезпеку для користувача



Захист даних

785

- ❑ Android використовує провідні в галузі функції безпеки для захисту даних користувачів
- ❑ Платформа надає інструменти та послуги для розробників, які допомагають забезпечити конфіденційність, цілісність та доступність даних користувачів
 - Шифрування
 - Шифрування на основі файлів — Дозволяє шифрувати області зберігання різними ключами та доступна для використання на пристроях починаючи з Android 7.0
 - Повне дискове шифрування — Пристрої з ОС Android 5.0 до 9.0 можуть використовувати шифрування повного диска замість шифрування на основі файлів
 - Шифрування резервних копій — Пристрої під управлінням Android 9.0 і новіших версій тепер можуть підтримувати покращене шифрування резервних копій
 - Оновлення безпеки Android
 - Оновлення партнерів OEM
 - Оновлення системи Google Play
 - Conscrypt — модуль Conscrypt прискорює покращення безпеки
 - **Adiantum** — метод шифрування, призначений для пристроїв з ОС Android 9.0 і вище, процесори яких не мають інструкцій AES

Донецький національний технічний університет



Тема 10. Архітектура безпеки і механізми захисту MacOS

Лекція 22. Механізми захисту MacOS

Предмет: БЗ0С

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

787

- ☐ Вступ
- ☐ Довірений кеш
- ☐ Перетворення
- ☐ Розширення ядра MacOS
- ☐ Пісочниця



Надійність macOS

- ❑ macOS працює на різних пристроях, включаючи Intel та Apple чіпи. Вона підтримує різні сценарії використання, від базових застосунків до втручання в ядро для коду.
- ❑ Під час розробки, Apple використовувала інтегрований підхід до обладнання та програмного забезпечення для забезпечення безпеки та спрощення управління, але зберігає можливості налаштування для користувача.
- ❑ macOS включає ключові технології безпеки для захисту корпоративних даних та інтеграції в корпоративні мережі.



Функції, які допомагають безпечно задовольнити різні потреби користувачів macOS

- ☐ Безпека підписаного системного тому
- ☐ Захист цілісності системи та довірені кеші
- ☐ Захист периферійних пристроїв
- ☐ Підтримка та безпека Rosetta 2 (автоматичне перетворення) для комп'ютерів Mac з чіпом Apple
- ☐ Підтримка та засоби захисту прямого доступу до пам'яті (DMA)
- ☐ Підтримка та безпека розширень ядра
- ☐ Підтримка та безпека додаткових ПЗУ
- ☐ Безпека прошивки UEFI для комп'ютерів Mac з процесором Intel



Захист цілісності системи

- ❑ Функція захисту цілісності системи (SIP) в MacOS використовує дозволи ядра для обмеження можливості запису в критично важливі системні файли. Це окрема функція, і вона доповнює функцію апаратного захисту цілісності ядра (KIP), доступну на комп'ютері Mac з чіпом Apple, яка захищає від зміни ядра в пам'яті.
- ❑ Для надання цього засобу захисту та ряду інших на рівні ядра, включаючи пісочницю та Data Vault, використовуються обов'язкові засоби контролю доступу.



Обов'язкові засоби контролю доступу

- ❑ В MacOS використовуються обов'язкові засоби контролю доступу, які встановлюють обмеження, що не можна обійти. Вони відрізняються від ініціативного контролю доступу, де користувач може змінювати політики безпеки.
- ❑ Ці засоби не видні користувачам, але допомагають реалізувати функції, як пісочниця, батьківський контроль, керовані налаштування та захист цілісності системи.



Захист цілісності системи

- ☐ Захист цілісності системи в MacOS обмежує доступ до критичних місць файлової системи, щоб запобігти змінам шкідливим кодом. Ця функція включена за замовчуванням у OS X 10.11 та новіших.
- ☐ Відключення цієї функції на Mac з Intel призводить до зняття захисту з усіх розділів зберігання. macOS застосовує цю політику до всіх процесів, незалежно від їхніх прав доступу.



Довірені кеші

- ☐ Статичний довірений кеш в macOS - це запис двійкових файлів Mach-O, який є частиною безпечного завантаження.
- ☐ При виконанні двійкового файлу, його каталог коду хешується і перевіряється на наявність в кеші.
- ☐ Неплатформні двійкові файли вимагають дійсних ланцюгів сертифікатів.
- ☐ Кеш прив'язує двійкові файли до версії MacOS, запобігаючи перенесенню файлів зі старих ОС.



Код платформи, що постачається окремо від операційної системи

- ❑ Деякі двійкові файли від Apple, такі як для Xcode та стеку інструментів розробника, не підписані з використанням ідентифікатора платформи. Незважаючи на це, вони можуть виконуватися з повноваженнями платформи на комп'ютері Mac з чіпом Apple та комп'ютері Mac з чіпом Apple T2.
- ❑ Оскільки це програмне забезпечення платформи постачається незалежно від macOS, на нього не поширюється процедура відкликання, пов'язана зі статичним довіренням кешем.



Завантажені довірені кеші

- ❑ Деякі програмні пакети Apple мають завантажувані довірені кеші, які мають таку ж структуру даних, як і статичний довірений кеш. Ці кеші проходять аутентифікацію або за тим же механізмом, що і завантажувальна прошивка, або як об'єкти з глобальним підписом.
- ❑ Прикладом персоналізованого довіреного кешу може бути кеш, що постачається з образом диска для діагностики комп'ютера Mac з чіпом Apple.
- ❑ Довірений кеш з глобальним підписом постачається, наприклад, з оновленнями програмного забезпечення MacOS.



Безпека периферійних процесорів на комп'ютерах Mac

- ☐ Сучасні комп'ютерні системи мають багато вбудованих периферійних процесорів для різних завдань.
- ☐ Якщо використання окремих процесорів з власною прошивкою є обов'язковим, приймаються заходи, щоб запобігти злоумиснику утриматися в цьому процесорі.
- ☐ Це забезпечується одним з двох способів: процесор запускається, щоб завантажити перевірену прошивку з основного процесора при запуску; або периферійний процесор реалізує власну безпечну послідовність завантаження, перевіряючи свою прошивку при кожному завантаженні Mac.

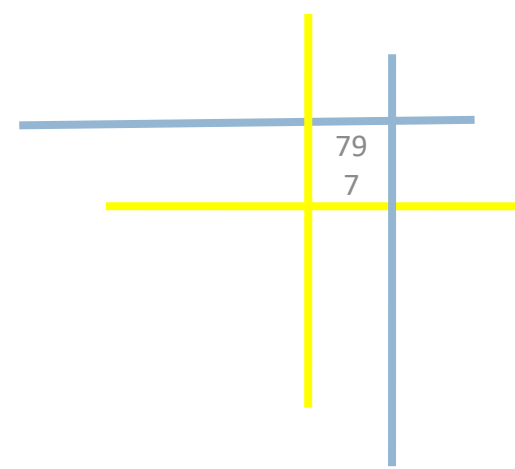


Rosetta 2 на комп'ютері Mac з чіпом Apple

- ❑ Комп'ютер Mac з чіпом Apple може запускати код, скомпільований для набору інструкцій x86_64, за допомогою механізму перетворення під назвою Rosetta 2.
- ❑ Пропонуються два типи перетворення: точно в потрібний момент (JIT) та заздалегідь.



Rosetta 2





Перетворення точно у потрібний момент (JIT)

- ☐ При використанні JIT, ядро передає управління загрузці перетворення Rosetta.
- ☐ Загрузка виконує перетворення сторінок x86_64 під час виконання образу.
- ☐ Ядро перевіряє хеші коду кожної сторінки на відповідність підпису коду, прикріпленого до двійкового файлу, коли виникає збіг сторінки.



Завчасне перетворення

- ☐ При використанні завчасного перетворення (АОТ) двійкові файли x86_64 читаються з сховища в оптимальний час для швидкості відгуку цього коду.
- ☐ Перетворені артефакти записуються в сховище як спеціальний файл об'єкта Mach.
- ☐ Процес АОТ починається аналогічно процесу JIT, але потім середовище виконання Rosetta надсилає в системну службу Rosetta запит про доступність преобразування АОТ для поточного виконуваного образу.
- ☐ Перетворені артефакти знаходяться в сховищі Data Vault, доступ до якого обмежений службою Rosetta.



Вміст статичного довіреного кешу

- ☐ macOS 11 або новіший має мультиархітектурні двійкові файли Mach, які містять фрагменти коду x86_64 та arm64.
- ☐ Користувач Mac з чіпом Apple може виконати фрагмент x86_64 системного двійкового файлу через процес Rosetta.
- ☐ Для підтримки цього підходу статичний довірений кеш, який постачається з macOS, містить три хеші каталога коду на кожний файл об'єкта Mach.
- ☐ Процедура перетворення AOT Rosetta детермінована. Під час збірки macOS кожний файл об'єкта Mach проходить через процес перетворення AOT Rosetta, і отриманий хеш каталога коду записується в довірений кеш.



Непідписаний код x86_64

- ☐ Комп'ютер Mac з чіпом Apple не дозволяє виконання нативного коду arm64 без підпису коду. Цей підпис може бути простим, як спеціальний ad-hoc підпис коду.
- ☐ Двійковому коду x86_64 дозволено виконуватися через Rosetta, навіть не маючи інформації про підпис.
- ☐ Процедура підпису Secure Enclave для конкретного пристрою не ідентифікує цей код.



Засоби захисту прямого доступу до пам'яті комп'ютерах Mac

- ❑ Для досягнення високої пропускної здатності швидкісних інтерфейсів, таких як PCIe, FireWire, Thunderbolt та USB, комп'ютери повинні підтримувати прямий доступ до пам'яті (DMA) з периферійних пристроїв.
- ❑ З 2012 року на комп'ютерах Mac реалізовано багато технологій для захисту DMA, що дозволило отримати найефективніший та повний набір засобів захисту DMA порівняно з будь-яким іншим ПК.



Засоби захисту DMA для Mac з чіпом Apple

- ☐ В системі на кристалі Apple є модуль управління пам'яттю вводу-виводу (IOMMU) для кожного агента DMA в системі, включаючи порти PCIe та Thunderbolt.
- ☐ Периферійні пристрої мають доступ лише до тієї області, яка явно відповідає їх використанню.
- ☐ Якщо модуль IOMMU виявляє спробу периферійного пристрою отримати доступ до області пам'яті, яка не призначена для нього, він видає критичну помилку ядра.



Засоби захисту прямого доступу до пам'яті для Mac з процесором Intel

- ❑ Комп'ютери Mac з процесором Intel та технологією віртуалізації Intel для спрямованого вводу-виводу (VT-d) ініціалізують модуль IOMMU для повторного відображення DMA.
- ❑ Модуль IOMMU Apple починає роботу з політикою, яка накладає заборону за замовчуванням, тому система автоматично починає блокувати всі запити DMA від периферійних пристроїв
- ❑ Починаючи з MacOS 11, на всіх комп'ютерах Mac з чіпом безпеки Apple T2 використовуються драйвери UEFI, які використовують DMA в обмеженому середовищі кільця 3.



Розширення ядра в macOS

- ☐ Починаючи з macOS 11, якщо дозволені сторонні розширення ядра, їх неможливо завантажити в ядро за запитом. На Mac з чіпом Apple вимірювання AuxKC записується в політиці LocalPolicy.
- ☐ Перебудова AuxKC вимагає підтвердження користувача та перезавантаження macOS.
- ☐ Важливо! В macOS більше не рекомендується використовувати розширення ядра.
- ☐ Розширення ядра ставлять під ризик цілісність та надійність операційної системи, тому Apple рекомендує віддати перевагу рішенням, які не вимагають розширень.



Розширення ядра на комп'ютері Mac з чіпом Apple

- ☐ На Mac з чіпом Apple розширення ядра повинні бути дозволені за допомогою тривалого натискання кнопки живлення при запуску для переходу в режим однієї справжньої середовища відновлення (1TR).
- ☐ Після авторизації користувачем завантаження розширень ядра застосовується потік, затверджений користувачем.
- ☐ Після створення AuxKC її вимірювання надсилається в Secure Enclave на підпис і включення в структуру даних Image4.
- ☐ Відповідь розширення ядра використовується такими підсистемами, як Apple Pay, для визначення наявності будь-яких завантажених розширень ядра.



Альтернативи розширенням ядра (macOS 10.15 та новіші)

- ☐ У MacOS 10.15 розробники можуть розширювати можливості macOS, встановлюючи та використовуючи системні розширення, які виконуються в користувацькому просторі, а не на рівні ядра.
- ☐ Розширення, які працюють в користувацькому просторі, підвищують стабільність та безпеку macOS.
- ☐ Розробники можуть використовувати доступні програмні середовища для створення драйверів USB та інструментів захисту кінцевих точок, а також інструментів VPN та мережевих засобів, для цього не потрібні розширення ядра.



Завантаження розширень ядра, затверджених користувачем

- ☐ Для підвищення безпеки перед завантаженням розширень ядра, встановлених під час або після встановлення macOS 10.13, потрібна згода користувача. Цей процес називається затвердженням користувачем завантаження розширень ядра.
- ☐ Розширення ядра не вимагають затвердження в деяких випадках.
- ☐ Починаючи з MacOS 10.13.2, через систему MDM можна задати список розширень ядра, які завантажуються без дозволу користувача.



Безпека додаткових ПЗУ на комп'ютері Mac з чипом безпеки Apple T2

- ☐ Пристрої Thunderbolt та PCIe можуть мати додаткове ПЗП (OROM), фізично підключене до пристрою.
- ☐ У системах на основі UEFI ця прошивка зазвичай є драйвером UEFI, який зчитується прошивкою UEFI та виконується. Однак, оскільки пристрої OROM зазвичай є перезаписуваними, якщо зломисник перезапише OROM справжнього периферійного пристрою, то код зломисника буде виконаний на ранньому етапі процесу завантаження.
- ☐ На комп'ютерах Mac з MacOS 10.12.3, проданих після 2011 р., OROM не виконується за замовчуванням під час завантаження Mac, якщо не було натиснуто спеціальне поєднання клавіш.

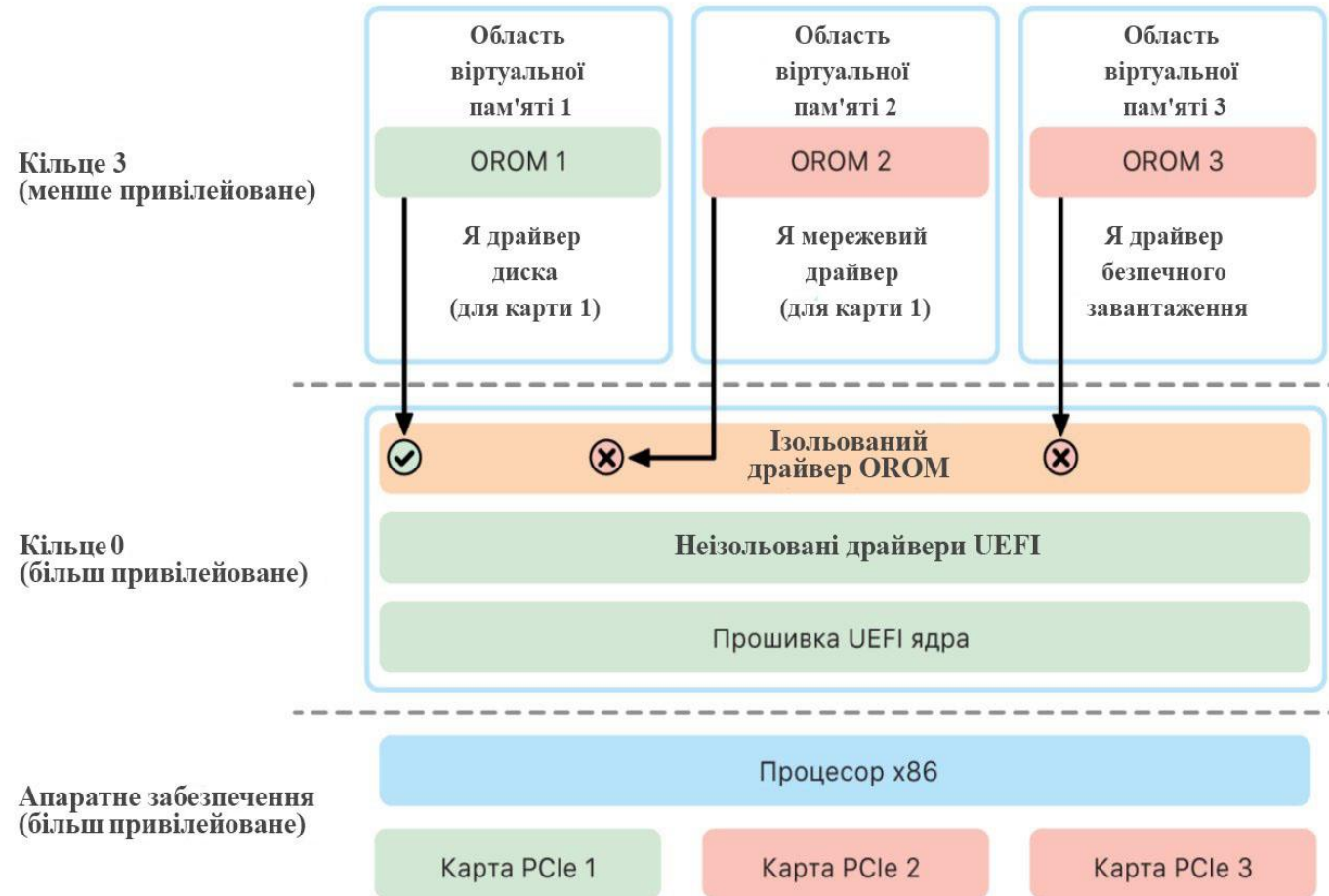


Безпека пісочниці OROM

- ☐ У прошивці UEFI в macOS 10.15 було додано механізм розміщення OROM у ізолюваному середовищі (пісочниці) та скасування повноважень OROM.
- ☐ Зазвичай прошивка UEFI виконує весь код, включаючи OROM, з максимальним рівнем прав процесора, називаємим "кільце 0".
- ☐ Пісочниця OROM скасовує повноваження OROM, використовуючи розділення віртуальної пам'яті аналогічно до того, як це робить ядро, і переводячи OROM в кільце 3.
- ☐ Ця схема має важливу перевагу: шкідливі OROM більше не можуть виконувати запис безпосередньо в пам'ять кільця 0.



Схема пісочниці





Підсистема Intel Management Engine (ME)

- ❑ Однією з підсистем, які зберігаються в прошивці UEFI, є прошивка Intel Management Engine (ME).
- ❑ Підсистему ME можна використовувати для віддаленого керування, захищеної передачі аудіо та відео та підвищення безпеки на Mac, на якому встановлено лише графічний процесор Intel.
- ❑ Комп'ютери Mac з процесором Intel використовують спеціальну прошивку ME, з якої було видалено більшість компонентів, щоб зменшити атакуєму поверхню підсистеми.



Режим керування системою (SMM)

- ❑ У процесорів Intel є спеціальний режим виконання, який відрізняється від звичайної роботи. Цей режим, називається режимом керування системою (SMM), спочатку розроблявся для виконання таких операцій з жорсткими часовими обмеженнями, як керування живленням.
- ❑ Однак для виконання таких дій на комп'ютерах Mac завжди використовувався окремий мікроконтролер, називається контролером керування системою (SMC).
- ❑ SMC більше не є окремим мікроконтролером: він інтегрований в чіп T2.

Донецький національний технічний університет



Тема 10. Архітектура безпеки і механізми захисту MacOS.

Лекція 23. Безпека ОС MacOS

Предмет: БЗОС

Розділ: Кібербезпека

Галузь: Інформаційні технології

Укладач: проф. Дорогий Я.Ю.



Лекційні питання

815

- Вступ
- Апаратні функції безпеки
- Secure Enclave
- Біометрія



Заходи щодо безпеки платформи Apple

816

- ☐ Продукція Apple = провідні технології + комплексні методи для захисту інформації
- ☐ Програма Apple Security Bounty
- ☐ Аудит безпеки та тестування продукції
- ☐ Моніторинг загроз



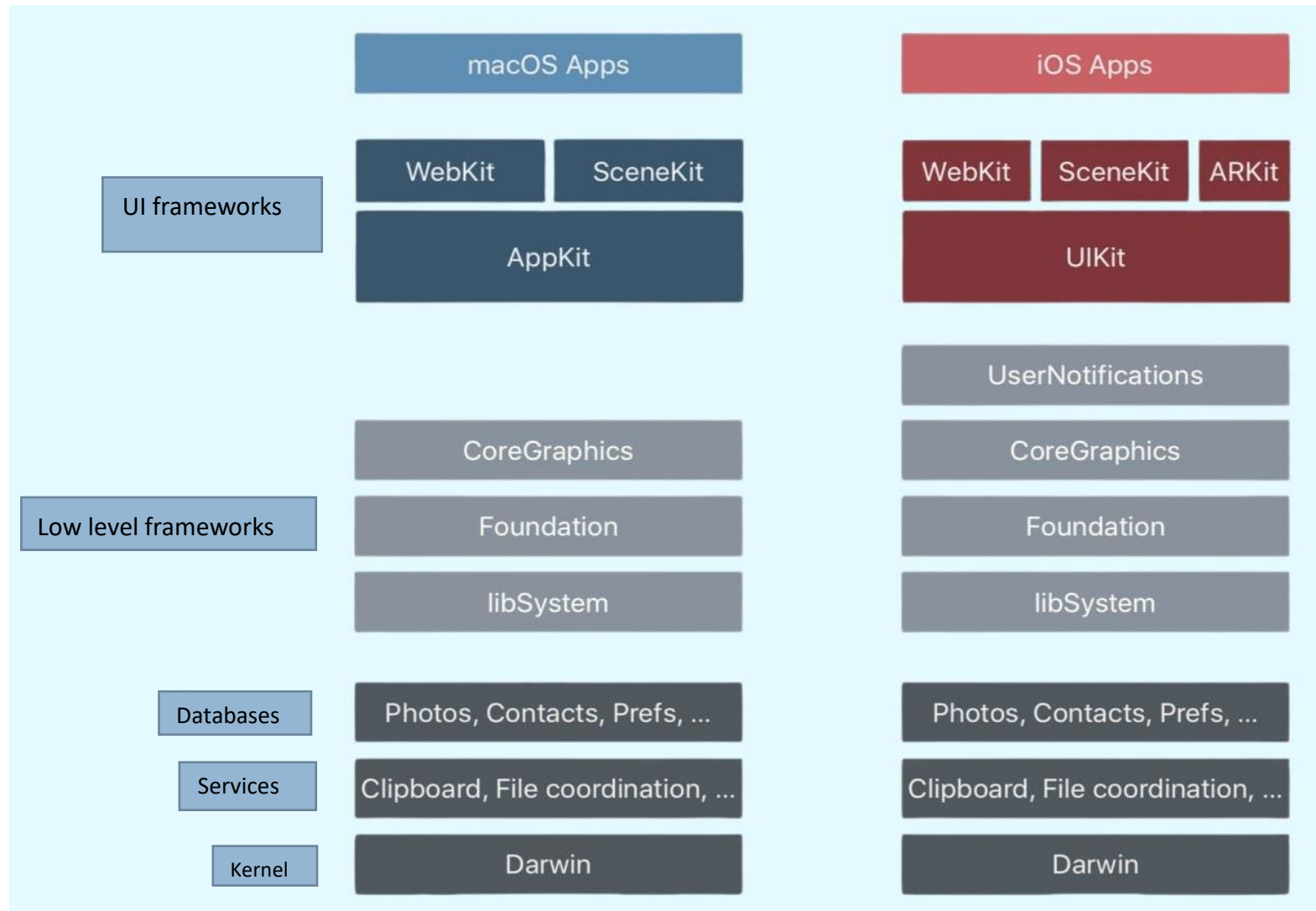
Збереження конфіденційності на Mac

- ❑ Конфіденційність – одне з основних прав людини.
- ❑ В macOS вбудовані функції безпеки, що допомагають підвищити рівень конфіденційності
 - Використання Екранного часу
 - Налаштування в Safari
 - Контроль персональної інформації, що передаються за допомогою застосунків
 - Налаштування брандмауера





Програмний стек





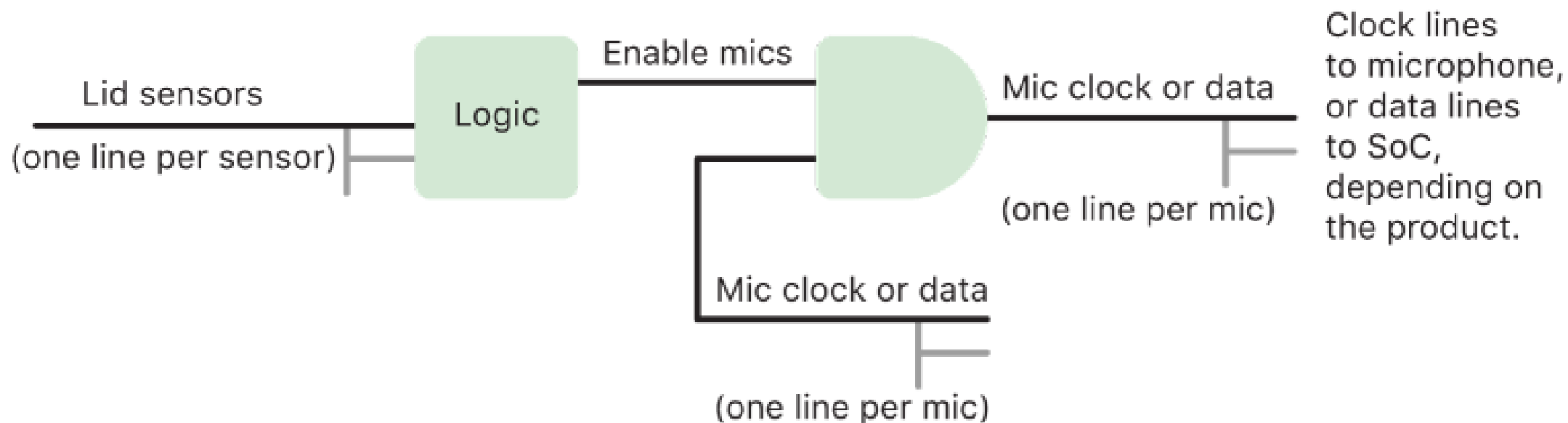
Апаратні функції безпеки

819

- ☐ Апаратне відключення мікрофона
- ☐ Безпечне завантаження
- ☐ Secure Enclave

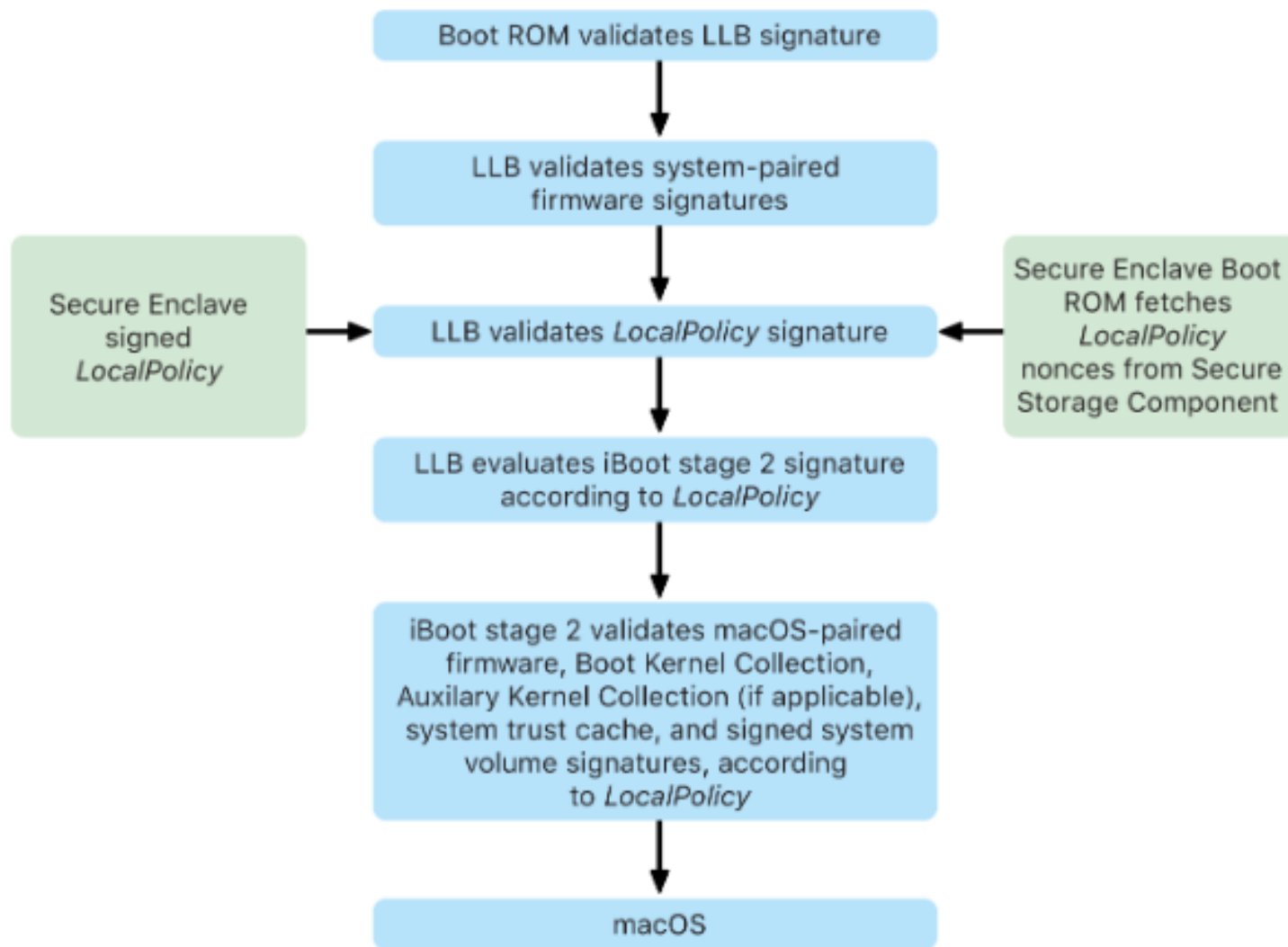


Апаратне відключення мікрофона





Безпечне завантаження





822

Рівні налаштувань безпеки ОС



Вищій рівень безпеки

823

- ❑ У цьому режимі допускається завантаження тільки програмного забезпечення, яке вже було заздалегідь найактуальнішим на момент встановлення.



Знижений рівень безпеки

- ❑ Низькорівневому завантажувачу вказується довіряти "глобальним" підписам, які поставляються в комплекті з операційною системою. Завдяки цьому на комп'ютері можна використовувати більш ранні версії macOS. Оскільки в старих версіях macOS неминуче є неусунуті вразливості, через це і обумовлена його назва.



Низький рівень безпеки

❑ Якщо обрано цей рівень безпеки, система працює так само, як і в режимі зниженого рівня безпеки щодо використання глобальних підписів для iBoot і не тільки, однак у цьому режимі iBoot пропонується приймати деякі об'єкти завантаження, підписані Secure Enclave за допомогою того самого ключа, який використовувався для підписання політики LocalPolicy. Цей рівень політики підтримує створення, підписання і завантаження користувачами власних ядер XNU.



826

Secure Enclave



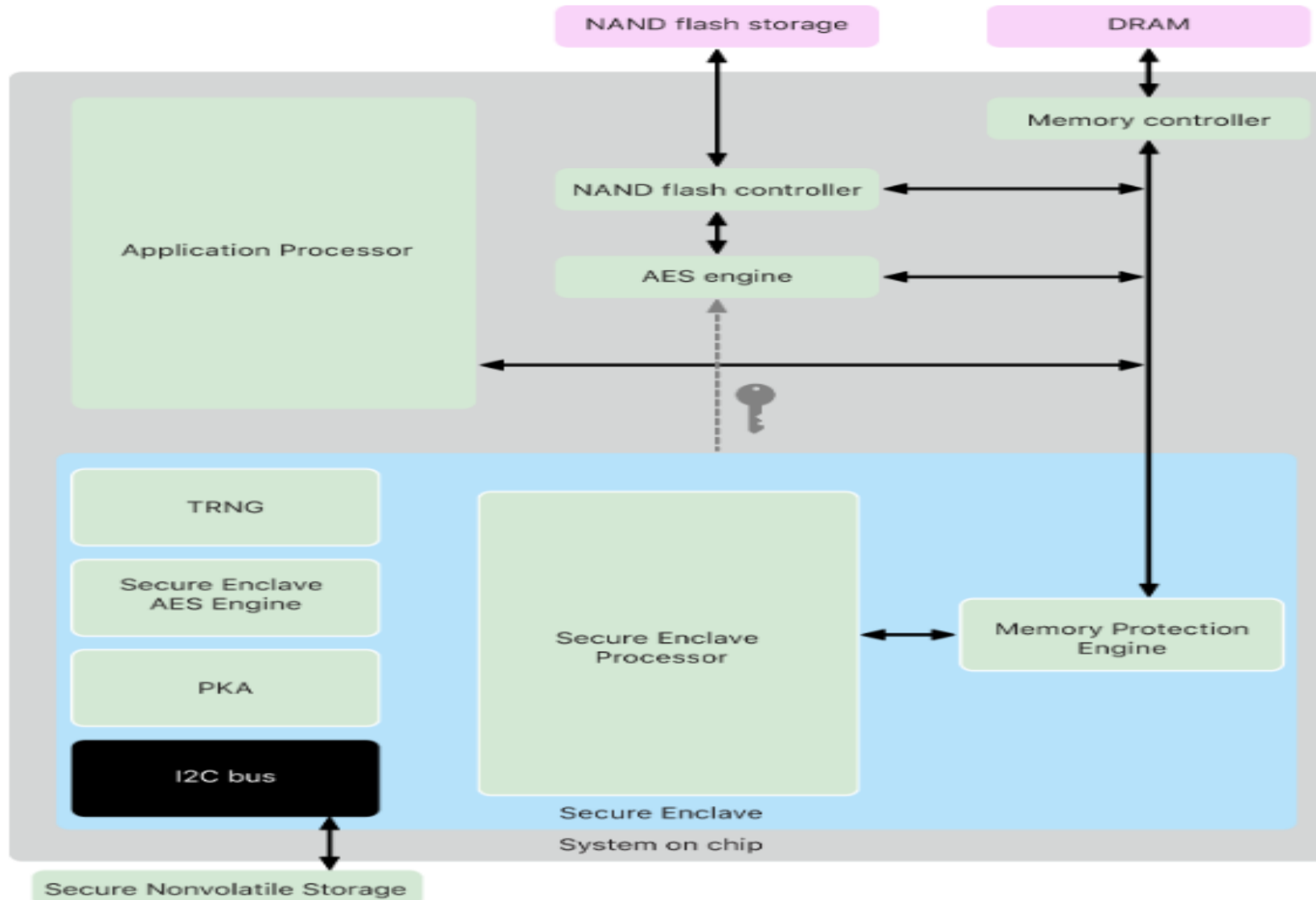
Secure Enclave₁

❑ Підсистема **Secure Enclave** є основою захисту на пристрої з MAC OS.

- Підсистема зберігає інформацію в підключеному сховищу окремо від NAND, яку використовує процесор та ОС.
- Підсистема зберігає криптографічні протоколи в окремому контейнері.



Secure Enclave₂





Secure Enclave₃

829

❑ Secure Enclave підтримує

- Симетричні криптографічні протоколи
 - AES (Advanced Encryption Standard)
- Асиметричні криптографічні протоколи

❑ Елементи керування доступом визначаються під час генерації ключів і застосовуються протягом усього терміну дії ключа



Кореневі криптографічні ключі₁

❑ **Secure Enclave** містить кореневий криптографічний ключ унікального ідентифікатора (UID)

- UID є унікальним для кожного окремого пристрою і не пов'язаний з якимось ще ідентифікатором на пристрої



Кореневі криптографічні ключі₂

- ❑ Випадково згенеровані UID вшиваються в систему на кристал під час збірки пристрою починаючи з кристалу A9
 - Ключ генерується випадково за допомогою TRNG
 - Процес генерації робить UID не видимим за кордонами пристрою. Це означає що ні Apple, ні її постачальники не матимуть доступ до нього



Модуль AES Secure Enclave₁

- ❑ Модуль AES представляє собою апаратний блок, котрий виконує симетричне шифрування на основі шифру AES.
 - Модуль виклює витік інформації через аналіз тактової частоти і статистичного аналізу живлення (SPA)
 - В системах на кристалі M1 і новіше модуль має компоненти протидії динамічному захисту аналізу живлення (DPA)



Модуль AES Secure Enclave₂

- ❑ Модуль AES підтримує апаратні і програмні ключі
 - Апаратні ключі виходять із ідентифікаторів UID або GID Secure Enclave
 - Ці ключи не видні навіть постачальнику ПЗ serOS
- ❑ В системах на кристалі M1 та новіше модуль AES має блоковані початкові біти, котрі змінюють ключ UID або GID Secure Enclave
 - Наприклад, початкові біти використовуються для того щоб закрити доступ до даних, які захищені паролем, при завантаженні в режимі оновлення прошивки пристрою (DFU)



Захищене енергозалежне сховище₁

- ❑ Для підключення захищеного енергозалежного сховища до Secure Enclave використовується виділена окремо шина I2C, тому доступ до сховища має тільки Secure Enclave
- ❑ Корені усіх ключів шифрування даних користувачів ентропійно закодовані і зберігаються енергонезалежному сховищі



Захищене енергозалежне сховище₂

□ Захищене енергонезалежне сховище використовується для всіх служб антиповтору, ці служби використовуються для ануляції даних, котрі відмічають границі автоповтору включаючи ці ситуації

- Зміни код паролю
- Додавання або видалення лица та відбитку пальця, ввімкнення або вимкнення, та збросу Face ID або Touch ID
- Додавання або видалення карти Apple Pay
- Видалення всього контенту та налаштувань



Монітори живлення і тактової частоти

- ☐ Secure Enclave оснащується мікросхемою для контролю напруги
- ☐ Ці мікросхеми допомагають зберігати характеристики електроживлення в кордонах робочого діапазону
- ☐ При виявленні забороненого режиму роботи тактові генератори Secure Enclave автоматично зупиняються і не перезапускаються до наступного збросу системи на кристалі



Обзор функций Secure Enclave

☐ Модуль захисту пам'яті:

- Шифування, автентифікація і механізм антиповтору

☐ Захищене сховище

- Компонент захищеного сховища 2 поліття

☐ Модуль AES

- Захист DPA і блоковані початкові біти

☐ РКА

- Ключ прив'язки до ОС і монітор завантаження



838

Біометрія



Біометрія₁

- ☐ Біометрична автентифікація є більш зручним для користувача так, як не потрібно вводити по декілька разів пароль та не зменшує безпеку пристрою
- ☐ Біометрична автентифікація підтримує безпеку надійного код паролю, навіть піднімає її рівень
- ☐ Face ID та Touch ID не замінюють пароль, або код пароль



Біометрія₂

- ❑ Архітектура біометричної безпеки заснована на строгому розділенні відповідальності між біометричним сенсором і Secure Enclave
- ❑ Сенсор фіксує біометричне зображення і передає Secure Enclave на безпечному для них з'єднанні
 - При реєстрації Secure Enclave зберігає, шифрує данні відповідних шаблонів Face ID і Touch ID
 - Secure Enclave зіставляє вхідні данні, які отримує від біометричного сенсору із збереженими біометричними шаблонами



Безпека Face ID₁

- ❑ Безпечна автентифікація виконується за допомогою камери TrueDepth, яка точно розпізнає лице використовуючи передові технології
- ❑ Face ID використовує нейромережі для розпізнавання обличчя користувача зіставляючи його.
 - Це дає можливість розблокувати пристрій навіть при використанні маски
 - Face ID автоматично пристосовується до змін зовнішності користувача і ретельно зберігає конфідесійність і безпеку біометричних даних



Безпека Face ID₂

- ☐ Функція активна в тому випадку, коли погляд направлений на камеру
 - Це необхідно щоби автентифікувати користувача з малою долею помилок і предзастереження електронних і фізичних засобів несанкціонованого доступу
- ☐ Коли камера TrueDepth підтверджує наявність лиця і увагу користувача вона створює двохвимірне інфрачервоне зображення
- ☐ Ці данні відправляються в Secure Enclave



Безпека Face ID₃

- ❑ Для застереження витоку даних двовимірні інфрачервоні зображення відправляються у випадковому порядку проєціює унікальну для кожного пристрою схему
- ❑ Secure Enclave створює математичне представлення цих даних та порівнює його з даними зареєстрованого зображення лица



Безпека Touch ID₁

- ☐ Touch ID система зчитування відбитку пальця, котра прискорює та спрощує процес безпечного доступу до пристрою
- ☐ Touch ID не позбавляє від необхідності створення код-паролю, який необхідно вводити при кожному при перезавантаженні або виходу з системи



Безпека Touch ID₂

- ☐ В деяких випадках Touch ID необхідно для розблокування захищених додатків код паролем або захищених паролем нотаток
- ☐ Touch ID використовується при видаленні коду пароля додавання або видалення відбитку пальця



Безпека Touch ID₃

- ❑ Коли датчик відбитку пальців розпізнає дотик пальця то він сканує його і відправляє зображення в Secure Enclave
- ❑ Канал, що використовується для захисту цього з'єднання залежить від того, вбудований сенсор в пристрій з Secure Enclave або є в окремому периферійному пристрої



Безпека Touch ID₄

- ☐ Вбудований сенсор Touch ID обмінюється даними по шині послідовного периферійного пристрою
- ☐ Для шифрування і автентифікації використовується ключ сеансу
- ☐ Ключ кожного сенсору є надійним випадково згенерованим і унікальним
- ☐ При обміні ключем використовується алгоритм AES



Безпека Touch ID₅

848

- ❑ В процесі векторизації відбитку зображення тимчасово зберігається в зашифрованому сегменті пам'яті Secure Enclave, а потім видаляється
- ❑ При реєстрації отримана карта вузлів зберігається в зашифрованій формі, яку може прочитати тільки Secure Enclave, причому вона зберігається лише у вигляді шаблону та не містить в собі ніякої інформації
- ❑ Ці дані ніколи не покидають пристрій, вони не відправляються в Apple та не включаються в резервні копії пристрою

Донецький національний технічний університет



Тема 10.Архітектура безпеки і механізми захисту MacOS

Лекція 24. Налаштування безпеки в MacOS

Предмет: БЗ0С

Розділ: Кібербезпека

Галузь: Інформаційні технології

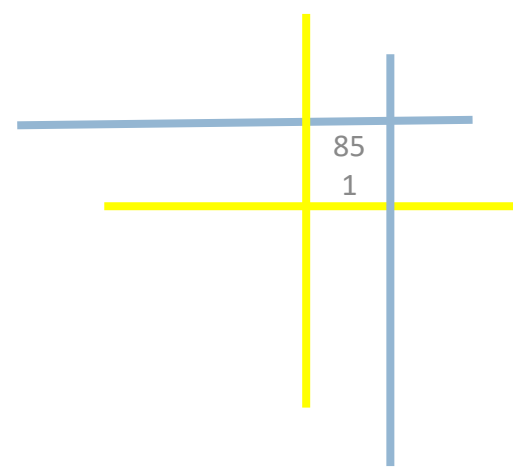
Укладач: проф. Дорогий Я.Ю.



Лекційні питання

850

- ☐ Безпечне завантаження
- ☐ Оновлення безпеки
- ☐ FileVault
- ☐ Безпека додатків



Безпечне завантаження

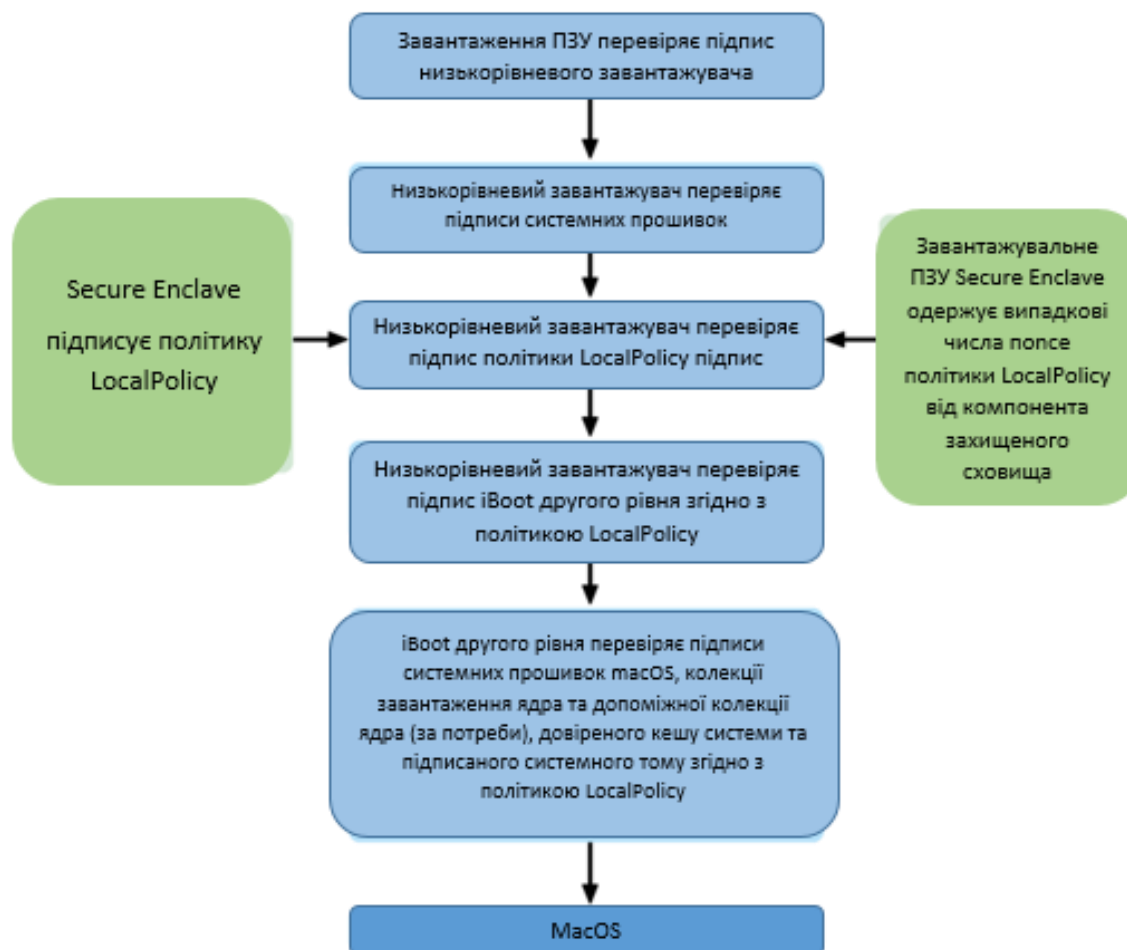


Безпечне завантаження

- ❑ **Безпечне завантаження** - це процес, який використовується в пристроях iOS та iPadOS, що забезпечує перевірку цілісності програмного забезпечення під час завантаження.
- ❑ Він використовує компоненти, які підписані Apple, для перевірки цілісності та перевірки ланцюжки довіри.
- ❑ Цей процес включає в себе загрузчики системи, ядро, розширення ядра та прошивку для немодульованої передачі по мобільній мережі.



Процес завантаження в MacOS





Політика “Вищий рівень безпеки”₁

- ❑ Ця політика встановлена за замовчуванням і взаємодіє аналогічно до систем iOS та iPadOS, щоб забезпечити вищий рівень безпеки для користувачів Mac.
- ❑ **Система персоналізованих підписів:** На відміну від глобального підпису, macOS під час завантаження та встановлення програмного забезпечення звертається до сервера підпису Apple, щоб отримати "персоналізований" підпис. Цей підпис містить унікальний ідентифікатор чіпа ECID, забезпечуючи додатковий рівень ідентифікації.
- ❑ **Захист від атак методом відкату:** Використання інтернет-сервера підпису забезпечує ефективний захист від атак методом відкату, надаючи більшу стійкість та безпеку порівняно з традиційними глобальними підписами.



Політика “Вищий рівень безпеки”₂

- ❑ **Унікальний ідентифікатор чіпа ECID:** Персоналізований підпис включає унікальний ідентифікатор процесора Apple, ускладнюючи можливість атак і забезпечуючи більший контроль над безпекою системи.
- ❑ **Захист від розповсюдження вразливостей:** При виявленні вразливості після зміни епохи безпеки, зловмисник не може легко використовувати вразливе програмне забезпечення з однієї системи для атаки іншої, забезпечуючи додатковий шар захисту.
- ❑ **Вибір користувача:** Незважаючи на стандартну політику безпеки, власник Mac завжди може вибрати інший рівень безпеки, який найкращим чином відповідає його потребам, залишаючи контроль у руках користувача.



Політика “Знижений рівень безпеки”₁

- ❑ Ця політика схожа на середній рівень безпеки на Mac з процесором Intel і чіпом T2. Постачальник (Apple) використовує "глобальний" цифровий підпис для коду, який можна використовувати на будь-якому Mac із встановленою політикою "Знижений рівень безпеки".
- ❑ **Обмежені можливості захисту:** Знижений рівень безпеки не гарантує захисту від атак методом відкату. Несанкціоновані зміни в операційній системі можуть призвести до недоступності даних користувача.
- ❑ **Ризики встановлення розширень ядра:** Помимо запуску старіших версій macOS, цей рівень безпеки необхідний для встановлення сторонніх розширень ядра, що може викласти систему користувача на ризик.



Політика “Знижений рівень безпеки”₂

- ❑ **Загрози від розширень ядра:** Розширення ядра мають такі ж дозволи, як саме ядро, тому їхні вразливості можуть стати прямою загрозою для всієї операційної системи.
- ❑ **Рекомендація для розробників:** Розробникам настійно рекомендується переходити на розширення системи, оскільки підтримка розширень ядра буде припинена для майбутніх комп’ютерів Mac з чіпом Apple.



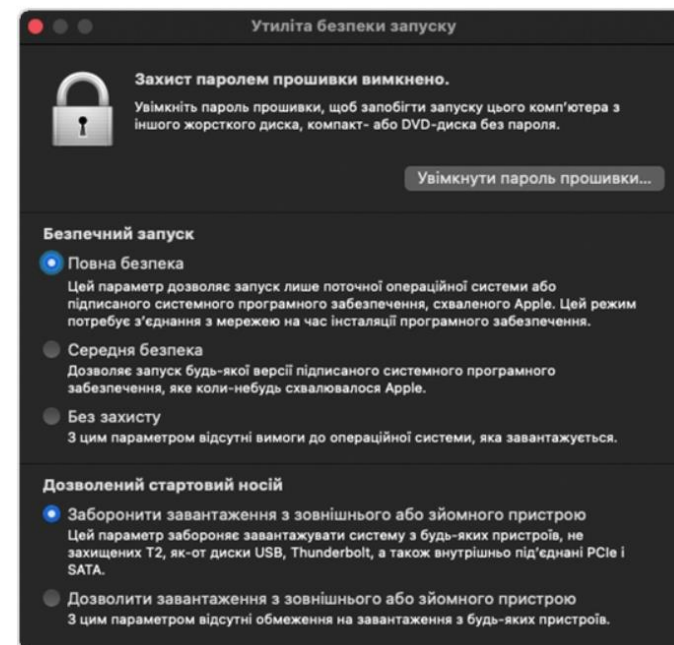
Політика "Знижений рівень безпеки"₃

- ❑ Ця політика схожа на середній рівень безпеки на комп'ютері Mac з процесором Intel та чіпом T2. Постачальник (Apple) створює цифровий підпис для коду, щоб підтвердити його походження. Цей "глобальний" підпис Apple можна використовувати на будь-якому комп'ютері Mac за умови, що встановлена політика "Знижений рівень безпеки".
- ❑ **Дії та ризики:** Цей рівень безпеки не лише дозволяє запускати старі версії macOS, але й потрібен для встановлення розширень ядра. Однак це може підвищити ризик, оскільки розширення ядра мають доступ до тих самих можливостей, що й саме ядро, тож вони можуть стати потенційною загрозою для всієї системи.
 - **Увага!** Знижений рівень безпеки не гарантує захист від атак методом відкату. Несанкціоновані зміни у операційній системі можуть призвести до недоступності користувацьких даних.



Утилита безопасного завантаження

- ❑ На комп'ютері Mac з процесором Intel та чіпом безпеки Apple T2 Утилита безпечно завантаження керує рядом параметрів політики безпеки.
- ❑ Для доступу до утиліти необхідно завантажити в режимі recoveryOS і вибрати Утилиту безпечно завантаження в меню «Утиліти».
- ❑ Утилита захищає підтримувані параметри безпеки, ускладнюючи маніпуляції з ними сторони зловмисника





Вищий рівень безпеки

- ❑ Це політика завантаження за замовчуванням, яка схожа на механізм роботи iOS і iPadOS або на вищий рівень безпеки на Mac з чіпом Apple.
- ❑ При завантаженні та підготовці до встановлення програмного забезпечення воно “персоналізується” за допомогою підпису, який містить універсальний ідентифікатор чіпа (ECID), тобто ідентифікатор чіпа T2. Це допомагає запобігти атакам методом відкату.



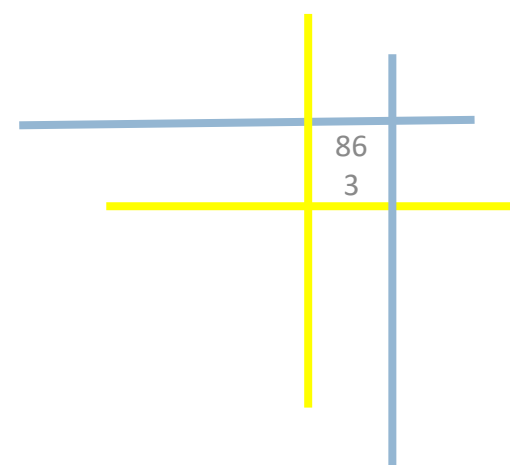
Середній рівень безпеки

- ❑ Ця політика завантаження схожа на традиційне безпечне завантаження UEFI, коли постачальник (у даному випадку Apple) генерує цифровий підпис для коду, щоб підтвердити, що він отриманий від постачальника.
- ❑ Ми називаємо цей підпис “глобальним”, оскільки його можна використовувати на будь-якому комп’ютері Mac протягом будь-якого періоду часу, якщо наразі на комп’ютері встановлена політика “Середній рівень безпеки”.
- ❑ При такому налаштуванні не вживаються жодні заходи для запобігання атакам методом відкату.



Функції безпеки відключені

- ☐ Процесор Intel не виконує оцінку безпечного завантаження, та Користувач може завантажувати все, що він хоче.



Безпечне оновлення програмного забезпечення



Безпечне оновлення ПЗ

- ❑ Безпека вимагає постійних зусиль. Недостатньо просто забезпечити безпечне завантаження версії операційної системи, встановленої на заводі. Також повинен існувати механізм для швидкого та безпечного отримання найновіших оновлень безпеки.
- ❑ Apple регулярно випускає оновлення програмного забезпечення відповідно до виникаючих проблем безпеки. Користувачі iOS та iPadOS отримують сповіщення про наявність оновлень прямо на пристрої. На Mac оновлення доступні в Системних налаштуваннях.
- ❑ Оновлення передаються через бездротову мережу, що прискорює впровадження найновіших виправлень, пов'язаних з безпекою.



Процес оновлення₁

- ☐ Під час оновлення використовується той же апаратний корінь довіри, що і при безпечному завантаженні. Це допомагає гарантувати, що встановлюється лише код, підписаний Apple.
- ☐ При фізичному підключенні оновлюваного пристрою до комп'ютера Mac завантажується і встановлюється повна копія iOS або iPadOS. При бездротовому оновленні завантажуються лише ті компоненти, які необхідні для оновлення системи.
- ☐ Крім того, оновлення ПЗ можуть кешуватися на комп'ютері Mac під управлінням macOS 10.13 і новіше з включеною функцією «Кешування контенту», щоб пристроям iOS і iPadOS не доводилося заново завантажувати необхідні оновлення з Інтернету.



Процес оновлення₂

- ☐ Під час оновлення встановлюється з'єднання з сервером авторизації встановлення Apple, яке включає список криптографічних показників для кожної частини встановлюваного пакета, випадкове значення (nonce) для функції антиповтору та унікальний ідентифікатор пристрою (ECID).
- ☐ Сервер авторизації перевіряє отримані виміри з версіями, встановлення яких дозволено. Якщо відповідність знайдено, сервер додає ECID до вимірів і підписує результат.
- ☐ Під час оновлення сервер передає на пристрій повний комплект підписаних даних.

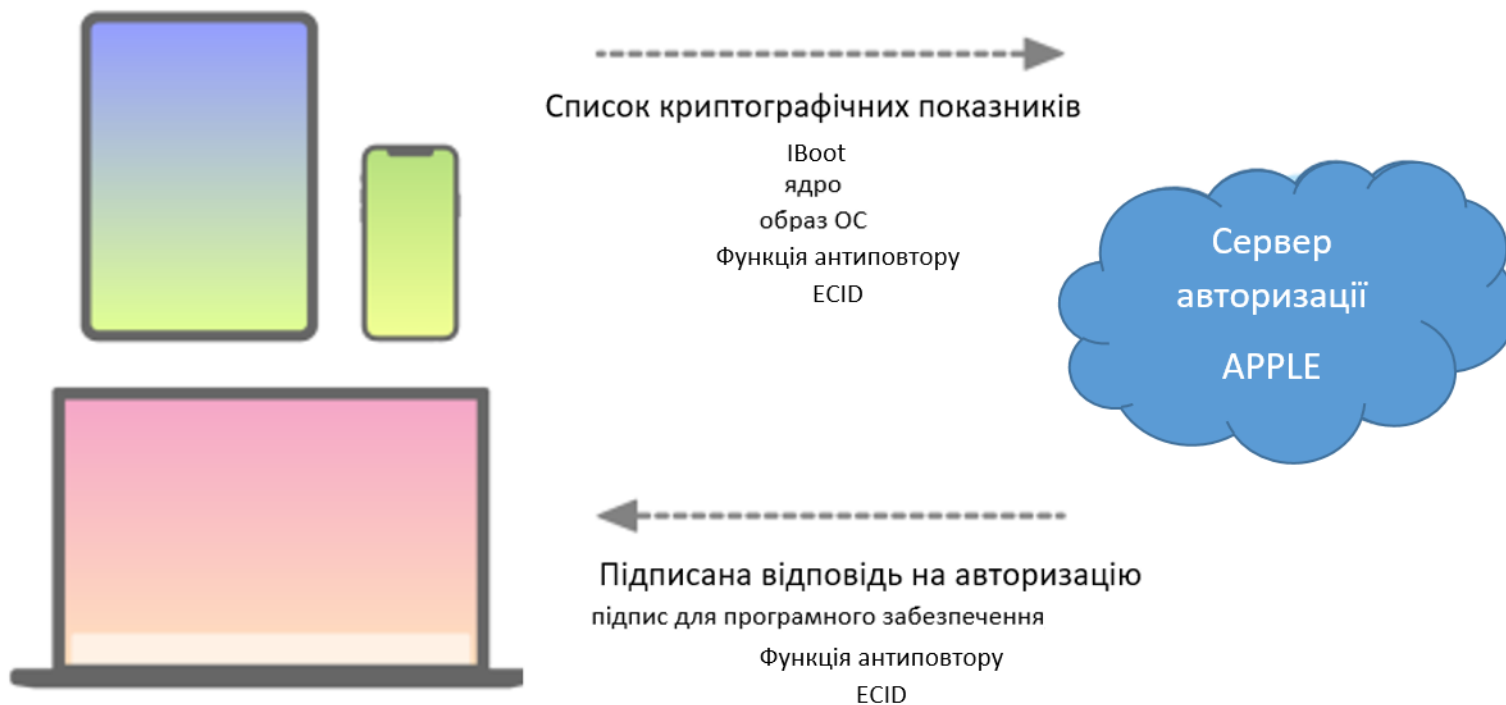


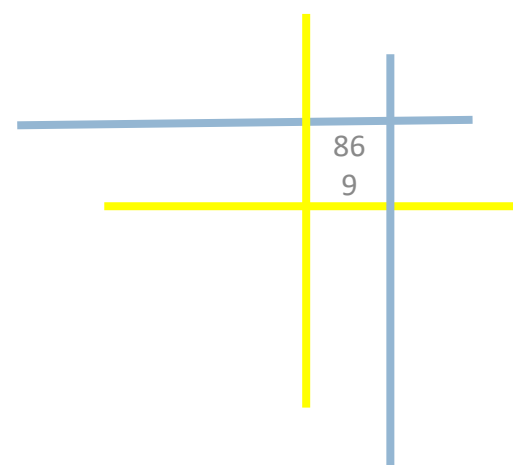
Безпека оновлення

- ❑ Додавання ECID необхідне, щоб зробити авторизацію запитуючого пристрою унікальною. Авторизація та підписання лише при наявності встановлених вимірів допомагають серверу гарантувати, що оновлення буде відбуватися саме так, як передбачено Apple.
- ❑ Перевірка ланцюга довіри в момент завантаження дозволяє переконатися, що підпис отримано від Apple, а значення вимірів об'єкта, завантаженого з пристрою зберігання, в поєднанні з ECID пристрою, збігаються з тими, на які поширюється підпис.



Персоналізований процес оновлення





FileVault



FileVault

- ❑ **FileVault** - це вбудована функція шифрування, що забезпечує захист усіх даних на комп'ютерах Mac. Її основа - алгоритм AES-XTS, що шифрує весь том як на внутрішніх, так і на знімних пристроях зберігання.
- ❑ FileVault реалізований з використанням технології захисту даних, ключа тому і апаратних функцій безпеки, які надає Secure Enclave. Якщо користувач увімкнув FileVault на Mac, під час завантаження потрібно ввести облікові дані. Це запобігає несанкціонованому доступу до зашифрованих томів APFS, навіть якщо витягнути фізичний пристрій зберігання з комп'ютера і підключити його до іншого комп'ютера.
- ❑ FileVault також дозволяє швидко і надійно стерти дані шляхом видалення криптографічних ключів.

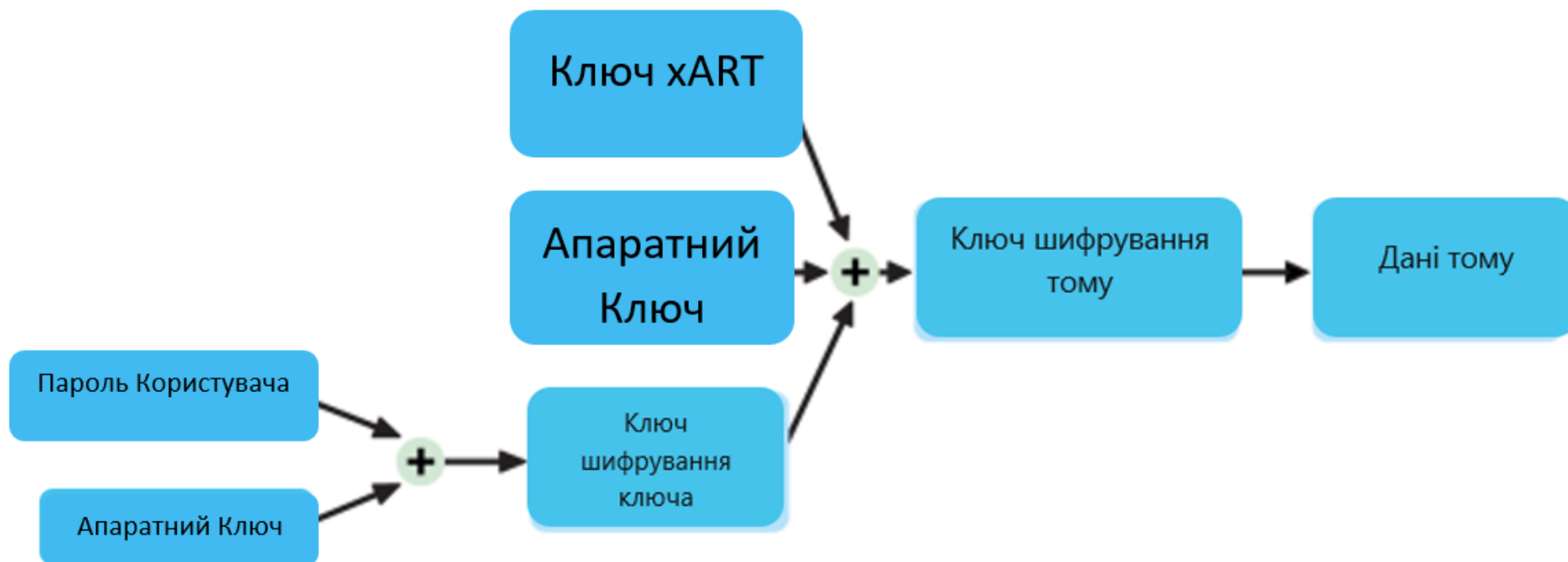


Внутрішнє сховище з увімкненою функцією FileVault

- Без дійсних облікових даних для входу або криптографічного ключа відновлення, внутрішні томи APFS залишаються зашифрованими і захищеними від несанкціонованого доступу. Це стосується і системного тому, і тому даних. Шифрування внутрішнього тому на Mac з чіпом Apple або T2 здійснюється шляхом побудови і контролю ієрархії ключів, і воно базується на технологіях апаратного шифрування, вбудованих в чіп. Ця ієрархія ключів призначена для одночасного вирішення наступних чотирьох завдань:
- вимога пароля користувача для розшифрування;
 - захист системи від атаки методом перебору;
 - надання можливості швидкого і надійного стирання даних шляхом видалення необхідних криптографічних ключів;
 - надання користувачеві можливості змінити свій пароль (і, в свою чергу, криптографічні ключі, які використовуються для захисту його файлів) без необхідності повторного шифрування всього тому.



Схема шифрування з увімкненою функцією FileVault



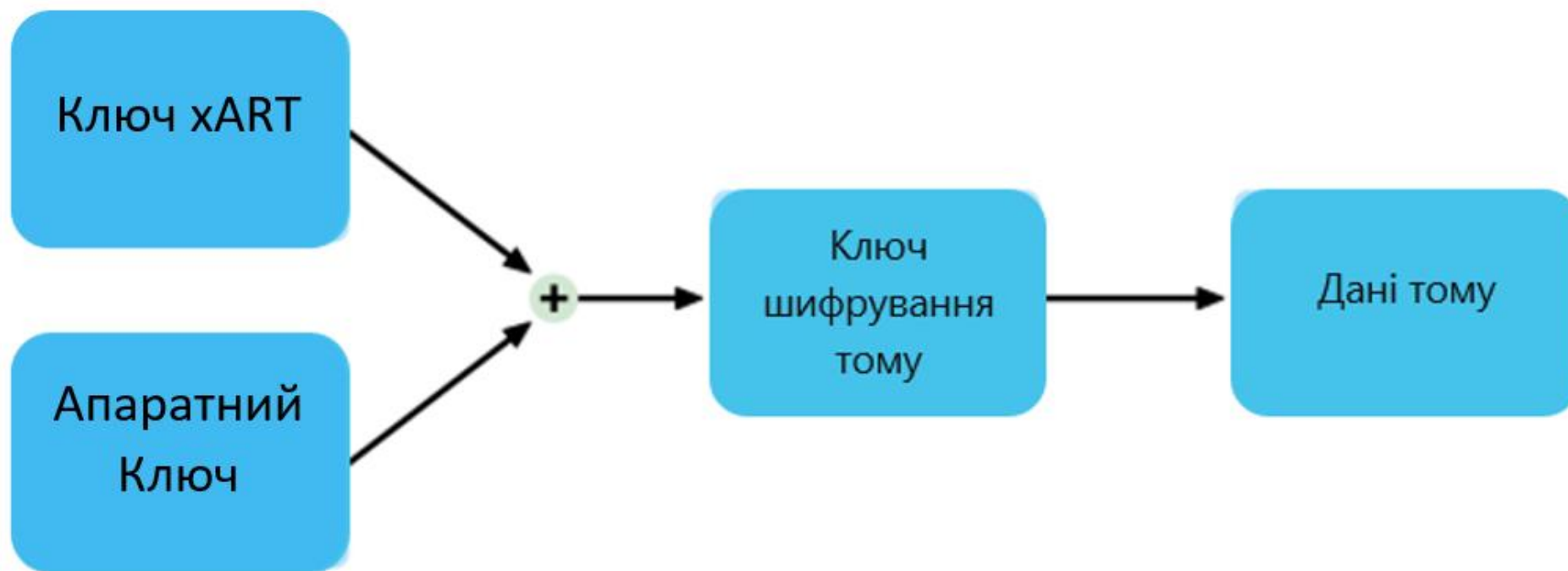


Внутрішнє сховище з вимкненою функцією FileVault

- ❑ Якщо на комп'ютері Mac з чіпом Apple або комп'ютері Mac з чіпом T2 під час початкової налаштування в Асистенті налаштування не включити FileVault, том все ще буде зашифрований, але ключ шифрування тому в Secure Enclave буде захищений лише за допомогою апаратного UID.
- ❑ Якщо включити FileVault пізніше (що буде виконано миттєво, оскільки дані вже зашифровані), механізм антиповтору дозволяє запобігти використанню старого ключа (основаного лише на апаратному UID) для розшифрування тому.
- ❑ Том буде захищений за допомогою комбінації пароля користувача і апаратного UID, як було описано раніше.



Схема шифрування з вимкненою функцією FileVault





Безпечне видалення тому

- ❑ Копроцесор Secure Enclave: Під час видалення тому, цей копроцесор безпечним чином видаляє ключ шифрування тому, що блокує подальший доступ навіть для самого Secure Enclave.
- ❑ Ключі шифрування в пакунку: Усі ключі шифрування томів упаковуються в ключ носія. Цей ключ не додає додаткової конфіденційності, але гарантує швидке та надійне видалення даних, унеможливлючи їх розшифрування.
- ❑ Гарантоване стирання на чіпі Apple та T2: Ключ носія на комп'ютерах з цими чіпами гарантовано стирається за допомогою технології, що підтримується Secure Enclave, забезпечуючи криптографічну недоступність тома.



Управління FileVault у macOS

- ❑ macOS дозволяє організаціям керувати FileVault за допомогою:
- безпечного токена
 - токена ініціалізації



Використання безпечного токена

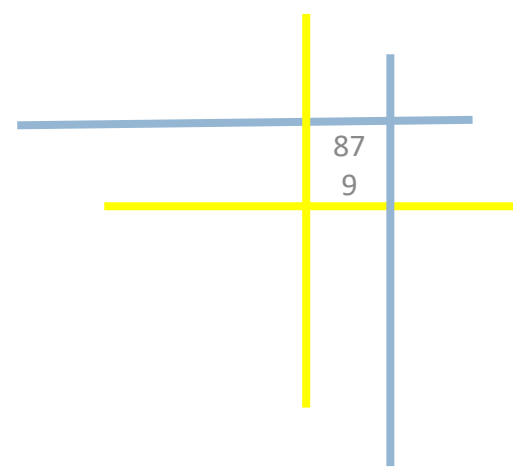
❑ **Безпечний токен** - це версія ключа шифрування ключа (KEK), захищена паролем користувача. При розгортанні FileVault в APFS користувач зберігає можливість використовувати існуючі інструменти та процеси, такі як особистий ключ відновлення (PRK), який може бути збережений в системі управління мобільними пристроями (MDM); створення та використання корпоративного ключа відновлення (IRK); перенесення включення FileVault на момент входу користувача в систему або виходу з системи. В macOS 11 при встановленні початкового пароля першого користувача на комп'ютері Mac цьому користувачу надається безпечний токен. Щоб виключити це, додайте параметр ;DisabledTags;SecureToken в атрибут AuthenticationAuthority користувача, перш ніж встановлювати пароль користувача.

```
sudo dscl . append /Users/<user name> AuthenticationAuthority  
";DisabledTags;SecureToken"
```



Використання токена ініціалізації

- ☐ У macOS 10.15 представлено нову функцію - токен ініціалізації. Він допомагає надавати безпечні токени для мобільних облікових записів та додаткового облікового запису адміністратора, створеного під час реєстрації пристрою.
- ☐ У macOS 11 за допомогою токена ініціалізації можна надати безпечний токен будь-якому користувачеві, який входить в систему на комп'ютері Mac. Для використання функції токена ініціалізації в macOS 10.15 або новішій версії потрібно зареєструвати Mac в MDM через Apple School Manager або Apple Business Manager.
- ☐ У macOS 10.15.4 або новішій версії сгенерований токен ініціалізації передається в систему MDM під час першого входу будь-якого користувача, у кого включений безпечний токен. У macOS 11 токен ініціалізації також можна використовувати не тільки для надання безпечного токена обліковим записам користувачів.
- ☐ На комп'ютері Mac з чіпом Apple токен ініціалізації, якщо він є, можна використовувати для авторизації встановлення як розширень ядра, так і оновлень програмного забезпечення за допомогою MDM.



Безпека додатків в macOS



Безпека застосунків в macOS

- ☐ В macOS реалізовано декілька рівнів безпеки застосунків, які дозволяють запускати лише підписані та надійні додатки з App Store.
- ☐ macOS має засоби виявлення та видалення шкідливого ПЗ та забезпечує додаткові заходи безпеки, щоб запобігти доступу ненадійних додатків до даних користувачів.
- ☐ Користувачі macOS можуть вибрати будь-яку модель безпеки, навіть якщо вона передбачає запуск непідписаного та ненадійного коду.



Процес підпису коду застосунків в macOS

- ❑ **Підпис коду та завірення** - це різні процеси в macOS. Розробник підписує код своїм сертифікатом, пов'язаним з унікальним ідентифікатором, тоді як завірення може проводити будь-який учасник ланцюга поширення програмного забезпечення.
- ❑ В App Store всі додатки мають підпис Apple, що гарантує їхню автентичність та цілісність. Цей підпис використовується для впевненості у тому, що додатки не підроблені чи змінені. Навіть додатки, що постачаються з пристроями Apple, також мають підпис від компанії.



Gatekeeper та захист під час виконання в macOS

□ **Gatekeeper** - це безпекова технологія в macOS, що гарантує, що лише довірене програмне забезпечення запускається на комп'ютері Mac користувача.

- Перевірка автентичності: Gatekeeper перевіряє додатки, плагіни та пакети встановлювачів на походження від підтверджених розробників або засвідчення від Apple.
- Контроль користувацької згоди: Користувачі отримують запит на згоду перед відкриттям завантаженого програмного забезпечення, щоб уникнути випадкового виконання шкідливого коду.
- Гнучкість налаштувань: Користувачі та організації можуть змінювати політику Gatekeeper для дозволу встановлення лише з App Store або встановлення підписаного іншими ідентифікаторами програмного забезпечення.



Gatekeeper та захист під час виконання в macOS

❑ Gatekeeper не лише перевіряє походження програмного забезпечення, а й захищає від потенційних загроз під час виконання

- Боротьба з шкідливими плагінами: Gatekeeper запобігає автоматичному завантаженню шкідливих плагінів через додатки, використовуючи заходи захисту доступу до випадкових місць.
- Підтримка організаційних налаштувань: Організації можуть використовувати системи управління мобільними пристроями (MDM) для зміни параметрів Gatekeeper та контролю над дозволами для встановлення програмного забезпечення.



Завірення застосунків в macOS

884

❑ **Завірення** - це сервіс від Apple, що служить для виявлення та перевірки наявності шкідливого програмного забезпечення.

- Перевірка додатків: Розробники, які не використовують App Store для розповсюдження своїх додатків, надсилають їх на перевірку до Apple, де проводиться аналіз на наявність шкідливого ПЗ.
- Квиток завірення: Якщо програмне забезпечення не має ознак шкідливого ПЗ, Apple видає квиток завірення для його запуску.



Робота системи завірення в macOS

- ❑ Система завірення використовується для швидкого блокування потенційно шкідливих додатків у macOS:
 - Квиток ануляції: Якщо виявлено шкідливе ПЗ, навіть після завірення, Apple може випустити квиток ануляції, щоб негайно заборонити їх запуск.
 - Автоматичне оновлення: macOS регулярно перевіряє нові квитки ануляції, щоб забезпечити найактуальнішу інформацію для Gatekeeper, забезпечуючи надійний захист від шкідливого програмного забезпечення.



XProtect в macOS

❑ **XProtect** - вбудована антивірусна система в macOS, яка виявляє та блокує шкідливе програмне забезпечення за допомогою бази сигнатур YARA.

- Виявлення шкідливого ПЗ: XProtect порівнює підозрілі файли з базою сигнатур, регулярно оновлюваною Apple, для виявлення відомих шкідливих програм.
- Автоматична блокування загроз: Система блокує виконання відомого шкідливого ПЗ та виявляє його у різних ситуаціях, включаючи зміни файлів та оновлення сигнатур.

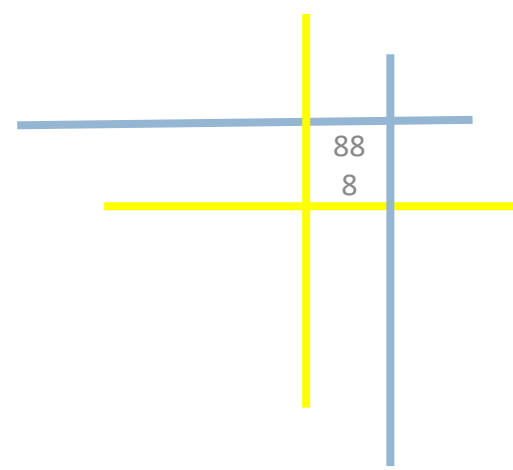


Робота XProtect в macOS

❑ XProtect вживає заходів для захисту та усунення шкідливих програм:

- Інтегрована захист: Якщо шкідливе ПЗ проникає на Mac, XProtect виявляє, блокує та усуває інфекцію за допомогою автоматичних оновлень.
- Оновлення безпеки: Apple надає регулярні оновлення для XProtect на основі останніх даних про загрози, забезпечуючи захист користувачів.

❑ Ця система дозволяє macOS ефективно боротися з потенційними загрозами, забезпечуючи безпеку користувачів.



Список рекомендованих джерел для самопідготовки



Список рекомендованих джерел для самопідготовки₁

1. М. В. Грайворонський, О. М. Новіков. Безпека інформаційно-комунікаційних систем. – К.: Видавнича група ВНУ, 2009. – 608 с.
2. Бондаренко М.Ф., Качко О.Г. Операційні системи. Навч. посібник. – Х.: Компанія СМІТ, 2018. – 432 с. ISBN 978-966-2028-02-7.
3. Яковина В.С. Операційні системи. Конспект лекцій. – Національний Університет "Львівська політехніка", Львів, 2016. 128 с.
4. Авраменко В. С., Авраменко А. С. Основи операційних систем. Навчальний посібник. – Черкаси: ЧНУ імені Богдана Хмельницького, 2018. – 524 с.: іл. ISBN 966-552-157-8.
5. Глоба Л.С. Розробка інформаційних ресурсів та систем. Том 1: Розподілені системи. – [Електронний ресурс] – Режим доступу: http://www.dut.edu.ua/uploads/l_1690_29298415.pdf
6. Глоба Л.С. Розробка інформаційних ресурсів та систем. Том 2: Розподілені системи. – [Електронний ресурс] – Режим доступу: http://www.dut.edu.ua/uploads/l_1690_27125554.pdf



Список рекомендованих джерел для самопідготовки₂

7. Rist O. Getting Started with VirtualBox / Oliver Rist. – Packt Publishing, 2018. -190 p.
8. Preece R. VirtualBox 6.0: Beginners Guide on how to Install, Configure, and Use VirtualBox / Robert Preece. – Independently published, 2019. – 70 p.
9. Russell D. Oracle VM VirtualBox: Beginner's Guide / Russel Dusty. – Packt Publishing, 2018. – 284 p.
10. Herter N. VirtualBox 6.0 – The Ultimate Beginner's Guide / Nicolas Herter. – Independently published, 2019. – 97 p.
11. Silverman. Oracle VM VirtualBox: An Ultimate Guide Book on Oracle Virtualization Essentials / Steve Silverman. – Independently published, 2018. – 87 p.
12. Lacroix J. Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04, 2nd Edition / J. Lacroix.
13. Tanenbaum A. Modern Operating Systems, 4th Edition / A. Tanenbaum, H. Bos. – Pearson, 2015. – 1104 p. ISBN-10: 0136006639, ISBN-13: 978-0136006633.



Список рекомендованих джерел для самопідготовки₃

13. Nemeth E. UNIX and Linux System Administration Handbook, 5th Edition / Evi Nemeth, Garth Snyder, Trent Hein, Ben Whaley, Dan Mackin. – Addison-Wesley Professional, 2017. – 1232 p. ISBN-10: 0134277554, ISBN-13: 978-0134277554.
14. Kerrisk M. The Linux Programming Interface: A Linux and UNIX System Programming Handbook / M. Kerrisk. – No Starch Press, 2010. – 1552 p. ISBN-10: 1593272200, ISBN-13: 978-1593272203.
15. Johnson C., Varma J. Pro Bash Programming, Second Edition: Scripting the GNU/Linux Shell, 2nd Edition / Chris Johnson, Jayant Varma. – Apress, 2015. – 279 p. ISBN-10: 1484201221, ISBN-13: 978-1484201220.
16. Зайцев, В. Г. Операційні системи [Електронний ресурс] : навчальний посібник для студентів спеціальності 123 «Комп'ютерна інженерія» / В. Г. Зайцев, І. П. Дробязко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,22 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 240 с. – Назва з екрана.
17. Погребняк Б.І. Операційні системи: навч. посібник / Б.І.Погребняк, М.В.Булаєнко; Харків. нац. ун-т міськ. госп-ва ім. О.М. Бекетова. – Харків: ХНУМГ ім. О.М. Бекетова, 2018. – 104с.



Список рекомендованих джерел для самопідготовки₄

18. Федотова-Півень І.М. Операційні системи: навчальний посібник. [за ред. В.М. Рудницького] / І.М. Федотова-Півень, І.В. Миронець, О.Б. Півень, С.В. Сисоєнко, Т.В. Миронюк; Черкаський державний технологічний університет. – Харків: ТОВ «ДІСА ПЛЮС», 2019. – 216 с.

19. Микитишин А.Г. Операційні системи: консп. лекц. / укл. А.Г. Микитишин, І.В. Чихіра. – Тернопіль: ТНТУ імені Івана Пулюя, 2016. – 107 с.

20. Docker Overview – Docker Docs [Електронний ресурс]. – Режим доступу: <https://docs.docker.com/get-started/overview/> (дата звернення: 24.10.2023 р.).

21. Docker Compose Overview – Docker Docs [Електронний ресурс]. – Режим доступу: <https://docs.docker.com/compose/> (дата звернення: 24.10.2023 р.).

22. OpenLDAP, Documentation [Електронний ресурс]. – Режим доступу: <https://www.openldap.org/doc/> (дата звернення: 24.10.2023 р.).

23. LDAP – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/LDAP> (дата звернення: 24.10.2023 р.).



Список рекомендованих джерел для самопідготовки₅

24. Configuring authentication and authorization in RHEL [Електронний ресурс]. – Режим доступу: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_authentication_and_authorization_in_rhel/index (дата звернення: 6.11.2023 р.).

25. . Команда AWK в Unix/Linux з прикладами [Електронний ресурс]. – Режим доступу: <https://devzone.org.ua/post/komanda-awk-v-unixlinux-z-prikladami> (дата звернення: 7.11.2023 р.).

26. Awk: Навчання сценарію оболонки за допомогою команди awk Terminal [Електронний ресурс]. – Режим доступу: <https://blog.desdelinux.net/uk/awk-shell-scripting-comando/> (дата звернення: 7.11.2023 р.).

27. AWK Tutorial [Електронний ресурс]. – Режим доступу: Посилання (дата звернення: 7.11.2023 р.).

28. Мова програмування GNU awk [Електронний ресурс]. – Режим доступу: https://docs.linux.org.ua/LDP/Bash_beginners_guide/the_gnu_awk_programming_language/ (дата звернення: 7.11.2023 р.).



Список рекомендованих джерел для самопідготовки₆

29. Що таке bash в Linux? Гайд по написанню bash-скриптів [Електронний ресурс]. – Режим доступу: <https://acode.com.ua/bash-in-linux/> (дата звернення: 7.11.2023 р.).
30. DHCP Failover Setup Guide (Microsoft Windows Server 2022) [Електронний ресурс]. – Режим доступу: <https://www.dannymoran.com/dhcp-failover-setup-guide-windows-server/> (Дата звернення: 12.12.2023).
31. Microsoft Windows Server 2022 Installation Guide Video [Електронний ресурс]. – Режим доступу: <https://www.dannymoran.com/microsoft-windows-server-2022-installation-guide/> (Дата звернення: 12.12.2023).
32. How to Install Windows Server 2019? [Електронний ресурс]. – Режим доступу: <https://www.servermania.com/kb/articles/how-to-install-windows-server-2019> (Дата звернення: 12.12.2023).
33. Install Active Directory on Windows Server 2012, 2016, and 2019 [Електронний ресурс]. – Режим доступу: <https://www.webhi.com/how-to/install-active-directory-on-windows-server-2012-2016-and-2019/> (Дата звернення: 12.12.2023).



Список рекомендованих джерел для самопідготовки₇

34. Guide: How to Install Active Directory in Windows Server 2019 Using PowerShell [Електронний ресурс]. – Режим доступу: <https://petri.com/how-to-install-active-directory-in-windows-server-2019-using-powershell/> (Дата звернення: 12.12.2023).
35. Group Policy Essentials [Електронний ресурс]. – Режим доступу: https://cdn.ttgtmedia.com/rms/editorial/gpo_chapter1.pdf (Дата звернення: 12.12.2023).
36. Group Policy Management in Active Directory [Електронний ресурс]. – Режим доступу: <https://woshub.com/group-policy-active-directory/> (Дата звернення: 12.12.2023).
37. Windows Group Policy (GPO) Explanation and Tutorials [Електронний ресурс]. – Режим доступу: <https://woshub.com/group-policy/> (Дата звернення: 12.12.2023).
38. How To Implement a Secure Group Policy [Електронний ресурс]. – Режим доступу: <https://mentorcruise.com/blog/how-to-implement-a-secure-group-policy-4d8ff/> (Дата звернення: 12.12.2023).
39. Windows Server 2019: Active Directory, GPO, R. Server Access [Електронний ресурс]. – Режим доступу: <https://www.udemy.com/course/windows-server-2019-active-directory-and-group-policies-gpo/> (Дата звернення: 12.12.2023).



Список рекомендованих джерел для самопідготовки₉

40. macOS-Security-and-Privacy-Guide [Електронний ресурс]. – Режим доступу: <https://github.com/drduh/macOS-Security-and-Privacy-Guide> (дата звернення: 17.12.2023).
41. MacOS Hardening Guide [Електронний ресурс]. – Режим доступу: <https://newosxbook.com/files/moxii3/AppendixA.pdf> (дата звернення: 17.12.2023).
42. Master Your Mac Security & macOS Complete Security Guide! [Електронний ресурс]. – Режим доступу: Посилання (дата звернення: 17.12.2023).
43. Best Practices To Hardening Your MacOS Security in 2022 [Електронний ресурс]. – Режим доступу: <https://www.realinfosec.net/trending-infosec-news/best-practices-to-hardening-your-macos-security-in-2022/> (дата звернення: 17.12.2023).