https://doi.org/10.31891/2307-5732-2025-353-57
УДК 004.03:004.45:004.75

СТЕЦЮК ЮРІЙ
Хмельницький національний університет
https://orcid.org/0000-0003-0312-2276
e-mail: yuriy.stetsuk@khmnu.edu.ua
САВЕНКО ОЛЕГ
Хмельницький національний університет
https://orcid.org/0000-0002-4104-745X
e-mail: savenko_oleg_st@ukr.net

МОДЕЛЬ ЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ БЕЗПЕКИ ОПЕРАЦІЙНИХ СИСТЕМ СТІЙКИХ ДО ВИТОКІВ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ

Розглядається побудова моделей підсистеми децентралізованої та централізованої системи безпеки ОС призначеної для роботи в складі захищеної системи для обробки конфіденційної інформації в багатомашинній мережевій комп'ютерній системі. Виконано аналіз публікацій стосовно побудови стійких до витоку конфіденційної інформації ОС та загалом захисту оброблюваної в них інформації. Узагальнено підходи по покращенню основних компонентів безпеки мережевих ОС. Розглянуті їх механізми захисту та способи покращення ефективності їх роботи в рамках систем безпеки ОС.

Розглянуті основні аспекти побудови децентралізованих та централізованих систем безпеки ОС та принципів організації роботи їх механізмів безпеки. Представлено математичні моделі децентралізованої та централізованої систем безпеки ОС, які враховують вплив множини загроз на ресурси системи та протидію їм відповідних захисних механізмів ОС і можуть використовуватись для дослідження розроблюваних систем безпеки ОС. При цьому їх набір параметрів може адаптуватись до вимог розроблюваної системи.

Виконано порівняльний аналіз ефективності централізованих систем безпеки та децентралізованих систем. Приведені їх основні недоліки та переваги. Ключовим аспектом, відповідно до прийнятого підходу, є знаходження збалансованої архітектури підсистеми безпеки ОС, яка може ефективно забезпечувати стійкість ОС до витоку конфіденційної інформації та її захисту загалом.

Побудовані моделі та проведені тести показали, що централізовані системи безпеки мають кращі показники виявлення та реагування на інциденти, що знижує ризики невиявлених витоків інформації, але мають вищі ризики при компрометації центрального вузла за рахунок наявності «єдиної точки відмови». У децентралізованих систем вона відсутня, але їх стійкість до витоків значно менша. Подальші дослідження будуть направлені на розробку моделей систем, з метою успадкування ними найкращих показників централізованих та децентралізованих систем безпеки.

Ключові слова: централізована система, децентралізована система, система безпеки.

STETSYUK YURIY
SAVENKO OLEG
Khmelnytskyi National University

MODEL OF A CENTRALIZED SECURITY SYSTEM FOR OPERATING SYSTEMS RESISTANT TO LEAKS OF CONFIDENTIAL INFORMATION

The construction of subsystem models of a decentralized and centralized OS security system designed to work as part of a protected system for processing confidential information in a multi-machine network computer system is considered. An analysis of publications on the construction of OSs resistant to leakage of confidential information and the protection of information processed in them in general is performed. Approaches to improving the main components of network OS security are summarized. Their protection mechanisms and methods for improving the efficiency of their operation within the framework of OS security systems are considered.

The main aspects of the construction of decentralized and centralized OS security systems and the principles of organizing the operation of their security mechanisms are considered. Mathematical models of decentralized and centralized OS security systems are presented, which take into account the impact of a set of threats on system resources and the counteraction of the corresponding OS protection mechanisms to them and can be used to study the developed OS security systems. At the same time, their set of parameters can be adapted to the requirements of the developed system.

A comparative analysis of the effectiveness of centralized security systems and decentralized systems has been performed.Their main disadvantages and advantages are given. The key aspect, according to the adopted approach, is finding a balanced architecture of the OS security subsystem, which can effectively ensure the OS's resistance to leaks of confidential information and its protection in general.

The built models and conducted tests have shown that centralized security systems have better indicators of incident detection and response, which reduces the risks of undetected information leaks, but have higher risks when compromising the central node due to the presence of a "single point of failure". Decentralized systems do not have it, but their resistance to leaks is much lower. Further research will be aimed at developing system models in order to inherit the best indicators of centralized and decentralized security systems.

Keywords: centralized system, decentralized system, security system

Стаття надійшла до редакції / Received 10.05.2025
Прийнята до друку / Accepted 25.05.2025

Вступ

Розвиток інформаційних технологій призвів до того, що практично всі аспекти людської діяльності стали критично залежати від різноманітних досягнень, таких як багаточислені електронні пристрої, обчислювальні системи та їх математичне забезпечення. Їх успішна робота є заложником таких якостей, як надійність, відмовостійкість і саме основне - безпека інформації.

В основі роботи таких систем лежать різного типу і призначення операційні системи. Їх фундаментальність полягає в абстрагуванні апаратного забезпечення від користувача інформаційної системи. ОС дозволяє користувачу не відчувати всю складність багатоликої апаратної платформи сучасної комп'ютерної системи, зосереджуючись на вирішені своєї прикладної задачі. Керуючи роботою комп'ютерної системи, ОС вирішує дуже важливі загальносистемні задачі. Тут і ефективний розподіл ресурсів апаратної частини, багатозадачність, стандартизовані інтефейси для комунікацій, забезпечення продуктивності роботи і саме головне - забезпечення інформаційної безпеки.

Як показує практика, ОС дуже часто стають об'єктами атак зловмисного програмного забезпечення (ЗПЗ). І все вказує на те, що цей процес боротьби за виживання є непервним, змінюється лише рівень її складності по мірі удосконалення ЗПЗ

Постановка проблеми

Невпинний розвиток комп'ютерних інформаційних технологій привів до необхідності широкого використання захищених мережевих ОС, призначених для обробки конфіденційних даних. Це в свою чергу потребує нового підходу в побудові їх підсистем безпеки. Вони мають забезпечувати свою функціональність з одночасно високими рівнями відмовостійкості, живучості та захисту інформації, що обробляється в них.

В даний час досить добре вивчені питання побудови підсистем захисту ОС, напрацьовано немало їх складових механізмів захисту, які мають високу відмовостійкість та живучість. Однак без використання збалансованих архітектур на основі централізованих систем безпеки сьогоднішні ОС, в умовах все зростаючих інформаційних потоків, не зможуть гарантувати постійно високу надійність їх захисту.

Аналіз досліджень та публікацій

З появою перших комп'терів і перших ОС почалась боротьба на виживання. ОС система є найскладнішою програмною системою, оскільки поєднує в собі вирішення кількох нетривіальних задач: адаптивне керування апаратними засобами та їх розподіл між запитами користувацьких та системних процесів, управління політиками безпеки, управління самими процесами. І все це із забезпеченням максимальної продуктивності роботи системи в умовах постійної боротьби за живучість, відмовостійкість та захист оброблюваної в її додатках і ній самій інформації в умовах впливу ЗПЗ та інших деструкцій.

Зразу стало зрозумілим, що тільки випереджаючий розвиток захисних механізмів ОС дозволить їм забезпечувати свій функціонал без значних інформаційних втрат. Пошук стійких до різного роду деструкцій моделей архітектури ОС став постійним процесом. Були представлені кілька абстрактних моделей систем захисту, які стали фундаментальними основами побудови ОС. Одна з перших - модель Біба (Biba), відповідно до якої всі суб'єкти та об'єкти деякої системи попередньо поділяються на кілька рівнів доступу, з накладенням обмежень на їх взаємодії [1]. Наступним кроком в розвитку абстрактних моделей систем безпеки ОС стала модель Гогена-Мезігера (Goguen-Meseguer) 1982 року, заснована на теорії автоматів [2]. В 1986 році представлена Сазерлендська модель захисту, яка робить акцент на взаємодії суб'єктів та потоків інформації. Ця модель дозволяє досліджувати поведінку множинних композицій функцій переходу з одного стану в інший [3]. Важливу роль в теорії захисту інформації відіграє модель захисту Кларка-Вільсона (Clark-Wilson), опублікована в 1987 році і модифікована в 1989 [4, 5], заснована на повсюдному використанні транзакцій і та на виваженому надані прав доступу суб'єктів до об'єктів.

Окрім чисто абстрактних розробок моделей побудови систем захисту ОС, представлено немало практичних розробок, втілених в фізичних ОС. Процес розвитку операційних систем, як класу програмного забезпечення, почався з універсальних ОС і призвів до виокремлення в них підкласів по принципу збільшення значимості деяких експлуатаційних параметрів, або збільшення їх спеціалізації. Так перевага такої властивості в ОС, як захист інформації та безпека, призвела до появи підкласу, захищених ОС, що широко використовуються в критично важливих середовищах, де втрата інформації неприпустима ні при яких обставинах.

Захищена ОС повинна відповідати певним стандартам та використовувати спеціалізовані механізми для протидії загрозам [6]. В стандарті США, що розроблений Національним інститутом стандартів і технологій (NIST), також приводиться визначення захищеної ОС в контексті вимог до інформаційних систем федерального рівня [7]. Також вимоги до захищених ОС приводяться в міжнародному стандарті [8]. Така увага до захищених ОС зі сторони законодавців всього, технологічно розвинутого світу, з однієї сторони показує, наскільки важливим питанням є створення захищених ОС для інформаційного суспільства.

Методи забезпечення дотримання встановлених політик безпеки розглянуті в [9]. Завдяки тому, що всі дії та події, що стосуються безпеки ресурсів, якими опікується ОС, механізм аудиту допомагає переконатися, що всі дії в системі виконуються відповідно до встановлених політик безпеки. У разі їх порушення система може заблокувати користувача, процес або відповідним чином сформувати повідомлення для адміністратора.

Засоби виявлення несанкціонованої діяльності та попередженню назріваючих потенційних загроз запропоновані в [10]. Методи відстеження активності та поведінки користувачів в [11, 12].

Ядро складає центральну частину операційної системи і слугує інтерфейсом між апаратними ресурсами КС та додатками. До складу ядра входять модулі забезпечення функцій, які вирішують внутрішньосистемні задачі організації обчислювального процесу, такі як перемикання контекстів процесів, завантаження/вивантаження сторінок пам'яті та керування пам'яттю КС [13] загалом, процесами введення-виведення, файловою системою, обробки переривань процесора, організації взаємодії та диспетчеризації процесів, оброблення команд і т.д [14].

Функції ОС, що виконуються модулями ядра, є найбільш часто застосовуваними і тому швидкість їх виконання визначає продуктивність всієї системи в цілому. Для забезпечення високої швидкості роботи ОС всі модулі ядра або велика їх частина постійно знаходяться в оперативній пам'яті, тобто є резидентними [15].

Додатки і процеси не мають прямого доступу до цих функцій. Вони доступні їм в режимі системного виклику ядра відповідно до їх прав. Такий режим роботи ядра ОС забезпечує високий рівень захищеності ОС, дозволяє тримати під контролем всі основні операції, пов'язані з використанням ресурсів КС. Разом з тим, як і будь яка централізація, це зменшує продуктивність ОС [16].

Ядро, при завантаженні ОС перемикає процесор в захищений (привілейований) режим, закріпляючи за собою повний доступ до всіх ресурсів КС, в той час як всі інші програми працюють в користувацькому режимі, де такого доступу немає. Таким чином ядро завжди знаходиться в привілейованому стані по відношенню до всіх прикладних програм, що не дає їм можливості втрутитись в роботу ядра так і інших прикладних програм [17, 18].

Багато уваги приділено ізоляції процесів [19, 20]. В ОС він забезпечується апаратними засобами сучасних процесорів, таких як підтримка віртуалізації та механізми захисту пам'яті. Це стало можливим завдяки розвитку архітектури сучасних процесорів, які апаратно підтримують ці два режими своєї роботи - **режим користувача (user mode)** і **режим ядра (kernel mode).** В користувацькому режимі процеси обмежені в привілеях і не можуть виконувати критичні операції, які можуть пошкодити систему. Перемикання між режимами виконується тільки через контрольовані точки входу (системні виклики), що захищає систему від некоректних дій користувацьких процесів. Ізоляція процесів є основоположним підходом забезпечення безпеки, стабільності та ефективності захищених операційних систем. На цьому базується захист КС від шкідливих або некоректних дій користувацьких програм.

Захищена ОС включає механізми, які перешкоджають атакам через пам'ять. Основною метою захисту пам'яті є запобігання доступу до пам'яті, яка знаходиться в розпорядженні ОС і поки, що не виділена під поточний процес. Це не дає можливості програмним помилкам чи діям ЗПЗ впливати на інші процеси, або саму операційну систему. Сучасні процесори надають можливість виконувати апаратний контроль виходу процесу за відведені межі пам'яті [21]. Спроба доступу до пам'яті, яка не належить даному процесу, спричиняє апаратне переривання, яке називають помилкою сегментації. Ця помилка, зазвичай, спричиняє аварійне завершення роботи процесу, що призвів до переривання. Захист пам'яті включає додаткові можливості для забезпечення безпеки комп'ютера такі, як захист виконуваного простору та будь яке місце розташування адресного простору процесу [22, 23]. В [24] пропонується метод визначення стану компонентів систем з метою виявлення шкідливого програмного забезпечення.

Багато уваги приділено механізмам шифрування в ядрі ОС, що гарантує збереження даних на дисках або інших носіях, а також даних, які передаються через мережу або інші канали зв'язку, запобігаючи перехопленню та зміні під час передачі [25]. Шифрування також може використовуватися в процесі завантаження ОС для перевірки цілісності та автентичності компонентів системи. Це дозволяє запобігти завантаженню модифікованих або шкідливих компонентів. Також до ядра ОС можуть бути включені механізми захисту криптографічних ключів та сертифікатів за допомогою шифрування. Це захистить їх від витоку або компрометації під час їх використання в системі. В [26] пропонується механізм раннього виявлення ЗПЗ під назвою CryptoSniffer, дозволяє значно нівелювати можливі наслідки атак. В [27] пропонується використання шифрування для перевірки цілісності компонентів самого ядра ОС, що унеможливлює несанкціоновані його зміни або встановленню шкідливих модулів, які можуть модифікувати критичні частини системи.

В [28] запропоновано ряд механізмів резервного копіювання. Це забезпечує відновлення інформації або важливих компонент системи на випадок спотворення або пошкодження. В захищених ОС наявність цього механізму є обов'язковою, оскільки він суттєво підвищує стійкість ОС до всіх видів збоїв. Цей механізм захисту ОС опирається на ряд нормативних актів, вимагають регулярного резервного копіювання для захисту конфіденційних даних та забезпечення безпеки вразливої інформації [29].

Забезпеченню контролю вхідного та вихідного трафіку за заданими правилами присв'ячено роботу [30]. Зі збільшенням кількості сервісів, які активно використовують мережі для забезпечення взаємозв'язку інформаційних систем стало фіксуватись більше подій, у яких вони наражаються на все зростаючі загрози безпеки. Отже, безпека мережевих систем стала обов'язковою. Для забезпечення інформаційної безпеки хмарних систем в [31] пропонується комплексний підхід, що враховує

класифікацію та управління безпекою даних, захист безпеки даних, стратегії та системи управління безпекою, установи та персонал управління безпекою, управління конструюванням безпеки, керування експлуатацією та обслуговуванням безпеки, топологію мережі, контроль доступу, виявлення та запобігання вторгненням, дані захист і аварійне відновлення. В [32] представлено підхід до виявлення ботнетів в розподілених системах. Він базується на розробленій трирівневій моделі, яка включає основні компоненти ботнету, а в [33] запропоновано нову інформаційну технологію виявлення ботнетів, засновану на аналізі їхньої поведінки в корпоративній мережі. Самоадаптивна система для протистояння ботнетам в корпоративних мережах розглянута в [34]

Підходи з мінімізації привілеїв запропоновано в [35], що утруднює зловмисникам використання вразливостей для проникнення в систему. Ефективні методи розслідування інцидентів безпеки, що виявляють причини порушень та вразливостей, а також визначають винуватців приводяться в [36].

Формулювання цілей статті

Метою цього дослідження є вирішення задача, яка полягає в знаходженні такої моделі архітектури підсистеми захисту ОС, яка б інтегрувала у собі найбільш ефективні механізми забезпечення її стійкості до витоків конфіденційної інформації, захисту інформації загалом, що обробляється в спеціалізованих інформаційних системах під її керуванням на основі використання централізованої системи безпеки керування захисними механізмами ОС.

Таким чином вирішувану наукову задачу можна охарактеризувати як актуальну і як ту, що має досить широке практичне застосування.

Виклад основного матеріалу

Модель децентралізованої системи безпеки ОС базується на принципі розподілення функцій безпеки між різними компонентами та сегментами системи. В такій архітектурі кожен компонент незалежно керує своїми політиками безпеки, механізмами захисту, моніторингом та управлінням привілеями. Відсутність централізованого управління підвищує гнучкість та масштабованість, але в той же час ускладняється координація між складовими системи, що підвищує вірогідність витоку інформації та успішності впливів ЗПЗ.

Для децентралізованих систем безпеки ОС характерним є наявність локальних систем управління доступом (Local Access Control Systems), при якій кожен компонент системи (сервер бази даних, додаток або процес) має свої власні політики доступу, які визначають, хто і що може робити з ресурсами, що реалізується через локальні списки контролю доступу (ACL). Забезпечення прав доступу здійснюється на основі ролей або атрибутів з використанням локальних механізмів автентифікації. Прикладом можуть слугувати різного роду сервери, що працюють під управлінням ОС та файлові системи цих же ОС які використовують власні політики доступу для кожного користувача (Рис.1).

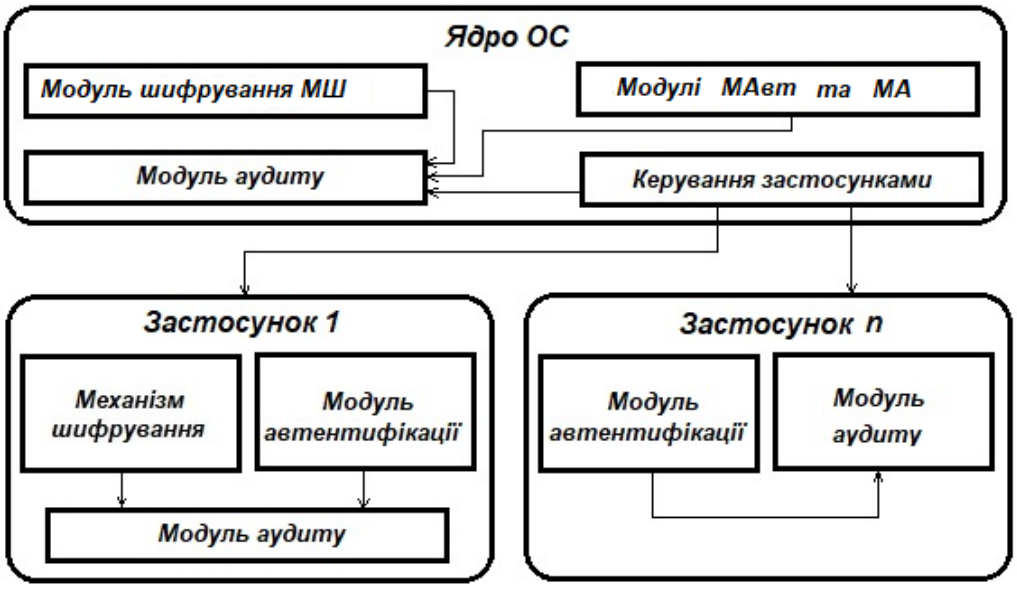

Рис. 1. Модель децентралізованої системи безпеки ОС

Наявність деякої множини локальних політик безпеки (Local Security Policies), що реалізуються застосунками та ядром ОС ускладнює систему безпеки системи загалом і може бути джерелом помилок при надані або позбавлені прав доступу до ресурсів, вести до збільшення прихованих каналів витоку конфіденційної інформації.

Децентралізовані системи безпеки ОС суттєво ускладнюють роботу деяких механізмів захисту. Через локальність їх реалізації вони можуть бути досить вразливими. Насамперед це стосується

шифрування, через наявність кількох центрів управління ключами. Надзвичайно ускладняється задача контролю за вхідним і вихідним трафіком.

Основна особливість децентралізованої моделі полягає в тому, що в ній відсутній єдиний центр керування безпекою. Ситуація, коли кожен компонент відповідає за свій захист має деякі позитивні моменти, але загалом недоліки їх переважають:

- ускладняється управління безпекою загалом. Відсутність централізованого управління робить координацію між компонентами системи набагато складнішою, а кожен сегмент потребує окремого адміністрування;

- підвищена ймовірність помилок конфігурації. Незалежне управління може призводити до невідповідностей або помилок у політиках безпеки між різними компонентами;

- обмежена координація. Відсутність єдиного механізму моніторингу ускладнює своєчасне виявлення і реагування на загрози, що можуть впливати на кілька сегментів одночасно.

У децентралізованій системі безпеки ОС кожен компонент самостійно виявляє загрози і захищає свої дані від витоку. Для опису загроз можна скористаємось ймовірнісними моделями, що базуються на можливості витоку конфіденційної інформації через конкретні компоненти системи. Нехай $T = \{t_1, t_2, \dots, t_n\}$ - множина загроз витоків інформації, а $V(r_i, t_j)$ - функція, яка відображає вразливість ресурсу r_i до загрози t_j, де $V(r_i, t_j) \in [0,1]$. Тоді імовірність витоку інформації через деякий компонент системи $P_{\text{витоку}}(r_i)$ може бути оцінена через зважену функцію ймовірностей атак на конкретний ресурс:

$$P_{\text{витоку}}(r_i) = \sum_{j=1}^{p} P(t_j) \cdot V(r_i, t_j), \quad (1)$$

де p- загальна кількість врахованих загроз; r_i – і-й шлях або ресурс, через який може відбутися витік інформації; t_j – j-та загроза, яка може впливати на ресурс r_i; $P(t_j)$ - ймовірність реалізації загрози t_j у відношенні ресурсу r_i; $\sum_{j=1}^{p}$ - сумарний вплив всіх загроз на ресурс r_i. Тоді загальна ймовірність витоку інформації в ОС з децентралізованою системою $P_{\text{витоку ДЦС}}$ буде обчислюватись, як добуток ймовірностей витоків но всіх компонентах системи:

$$P_{\text{витоку ДЦС}} = 1 - \prod_{i=1}^{p}(1 - P_{\text{витоку}}(r_i)), \quad (2)$$

де p – загальне число загроз, що діє у відношенні ресурсу r_i; $1 - P_{\text{витоку}}(r_i)$ – ймовірність того, що витоку через ресурс r_i не станеться; $\prod_{i=1}^{p}(1 - P_{\text{витоку}}(r_i))$ - ймовірність того, що жодна з p загроз не спричинить витоку інформації. Ця формула враховує захист кожного ресурсу та ймовірність витоку інформації через різні компоненти системи. Чим краще кожен компонент системи буде захищений, що відповідає меншим значенням $P_{\text{витоку}}(r_i)$, то і загальна ймовірність витоку конфіденційної інформації буде зменшена.

Таке представлення математичної моделі децентралізованої системи безпеки, що протидіє витоку інформації, базується на ймовірнісних оцінках доступу, шифрування, вразливостей та механізмів виявлення загроз. В її рамках кожен компонент відповідає за свій захист і моніторинг, а загальна ймовірність витоку інформації (формула 2) розраховується на основі захисту окремих сегментів системи (формула 1), що дозволяє оцінити ризики витоку інформації та ефективність захисних заходів у децентралізованій системі.

Тепер сконцентруємось на побудові моделі централізованої підсистеми безпеки деякої абстрактної мережевої ОС. Розглянемо основні складові централізованої підсистеми безпеки такої ОС. Для зосередження управління безпекою ОС в одному місці системи, включимо до складу її архітектури центральний модуль керування безпекою (ЦМКБ). Цей модуль є основним компонентом, який відповідає за управління всією системою захисту ОС. До його функцій відноситься визначення політик безпеки, моніторинг подій в системі, реагування на інциденти та аудит.

Для того щоб центральний модуль безпеки ЦМКБ отримував інформацію про стан безпеки зі всіх важливих вузлів ОС введемо до їх архітектури периферійні модулі безпеки ПМБ1 - ПМБn. Вони знаходяться у кожному вузлу мережевої ОС та забезпечують виконання локальних функцій безпеки, таких як перевірка прав користувачів і процесів на доступ до системних ресурсів на основі політик, отриманих від ЦМКБ, збір інформації про активність на локальному вузлу, яку передає до ЦМКБ для аналізу, а також здійснюють захист локальних ресурсів, таких як файли, пам'ять, мережеві з'єднання, канали переміщення інформації в комп'ютерній системі від несанкціонованого доступу. Так на один із ПМБ покладена функція мережевої безпеки для забезпечення захисту ОС від мережевих загроз шляхом використання між мережевого екрану (Firewall) та системи виявлення вторгнень (IDS) з метою контролю вхідного та вихідного трафіку на основі визначених політик центральним модулем ЦКМБ.

Для перевірки ідентичності користувачів та процесів до архітектури безпеки включено модуль автентифікації МАвт. Його функції полягатимуть в надані ЦКМБ інформації автентифікацію користувача (за допомогою пароля, біометричних даних, смарт-картки або іншого методу) та процесів (через перевірку цифрових підписів виконуваних файлів). Для перевірки прав користувачів та процесів до підсистеми безпеки ОС включено модуль авторизації МА. Його функції полягають в призначенні

користувачам та процесам певних ролей, що визначають їхні права доступу до різних ресурсів системи відповідно до політики RBAC.

Важливу роль в забезпечені безпеки ОС відіграє модуль шифрування МШ, який є обов'язковим для захищених ОС. Він працює під керівництвом центрального модуля безпеки ЦКМБ та призначається для шифрування даних на диску, у пам'яті, забезпечує безпечне зберігання та управління ключами шифрування. Модулі механізмів захисту ОС, що включені до складу моделі централізованої підсистеми безпеки ОС, не є вичерпними. Модель може бути розширена шляхом включення інших механізмів безпеки при потребі.

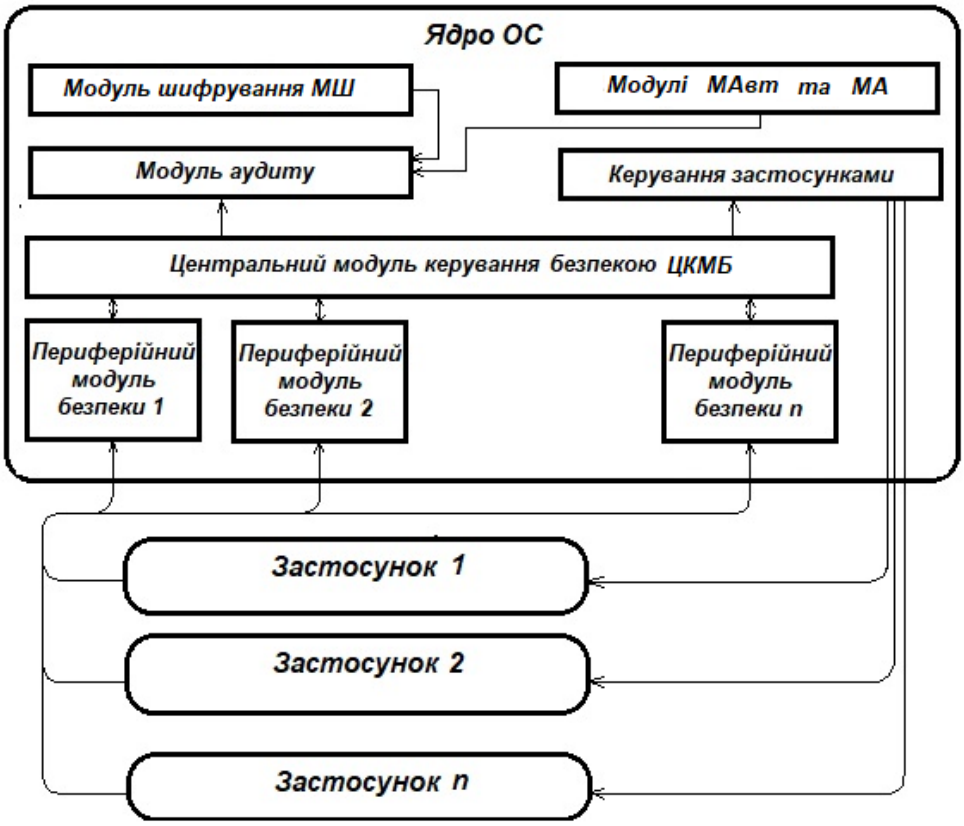

Рис. 2. Модель централізованої системи безпеки мережевої ОС

Описана модель представлена на рис. 2. В ній показані основні модулі централізованої підсистеми безпеки ОС, а також їх функціональні впливи, що вказують на відповідну взаємодію між модулями, визначаючи алгоритм її роботи:

- ЦКМБ надсилає політики безпеки периферійним модулям безпеки ПСМ1 - ПСМn;
- модулі ПСМ1 - ПСМn контролюють мережевий трафік та можливі загрози по каналах проходження інформації в комп'ютерній системі, надсилають дані моніторингу та звіти до центрального модуля керування ЦКМБ, до центрального модуля ЦКМБ;
- модуль автентифікації МАвт надсилає результати автентифікації до центрального модуля ЦКМБ;
- центральний модуль керування ЦКМБ запитує та отримує інформацію про права доступу з модуля авторизації МА.
- ЦКМБ Центральний модуль керує процесом шифрування через модуль шифрування МШ;
- модуль шифрування МШ надсилає ключі або інші дані для шифрування до ПСМ1 – ПСМn.

На рис. 3 представлена ще одна модель централізованої системи безпеки ОС особливістю якої включення до її складу механізму низькорівневого маркування конфіденційної інформації. Його наявність суттєво покращує стійкість ОС до витоку інформації за рахунок контролю інформації на рівні сторінок оперативної пам'яті. Основні модулі моделі ті ж, що показані на рис 2., але їх функції змінені з урахуванням механізму низькорівневого маркування МНМ:

- центральний модуль керування ЦКМБ передає політики безпеки, включаючи політики маркування конфіденційної інформації до периферійних модулів безпеки ПСМ1 - ПСМn.
- периферійні модулі безпеки ПСМ1 - ПСМn забезпечують безпеку проходження інформації по каналах в комп'ютерній системі та передають дані моніторингу центральному модулю ЦКМБ для аналізу;
- модуль маркування МНМ перевіряє коректність маркерів конфіденційності та передає інформацію про маркування конфіденційної інформації периферійним модулям безпеки ПСМ1 - ПСМn для подальшого застосування на рівні сторінок пам'яті.

- модулі автентифікації МАвт, авторизації МА, шифрування МШ, та ПСМ, що відповідає за мережеву безпеку взаємодіють з модулем маркування МНМ при виконанні своїх функцій.

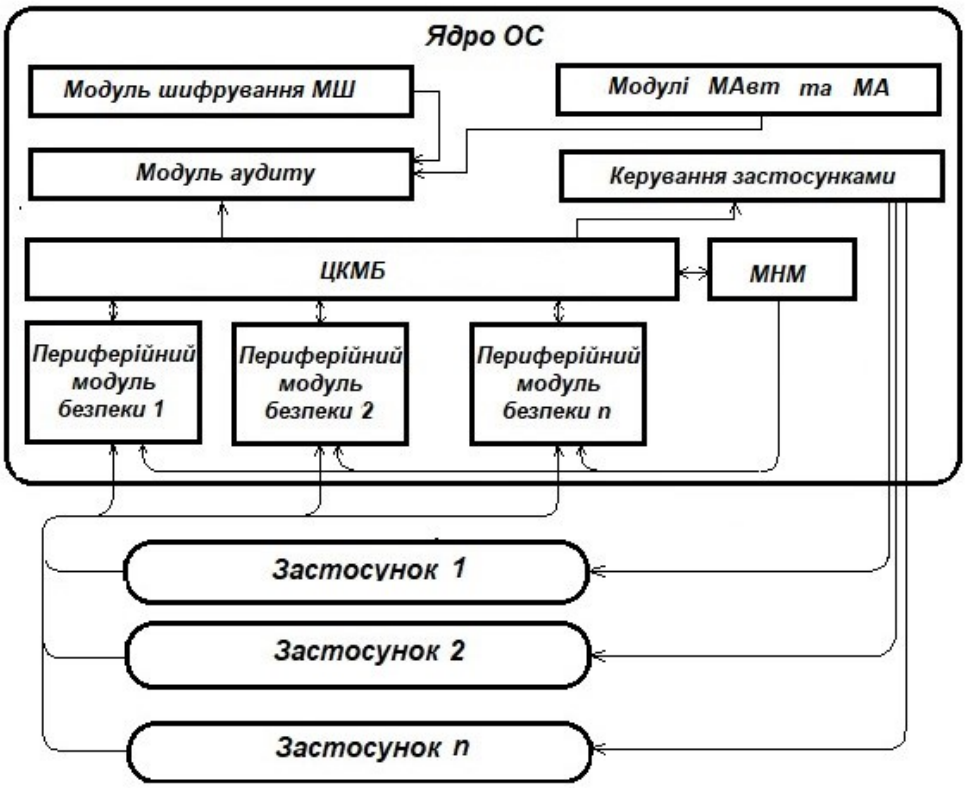

Рис. 3. Модель централізованої системи безпеки ОС з механізмом низькорівневого маркування конфіденційної інформації

Математично, модель централізованої системи безпеки, що протидіє витоку інформації, можна представити як взаємодія компонентів, керованих центральним модулем безпеки ЦКМБ. До її складу входять множини користувачів $U = \{u_1, u_2, \dots, u_n\}$, процесів $P = \{p_1, p_2, \dots, p_m\}$ та ресурсів $R = \{r_1, r_2, \dots, r_j\}$. Оскільки в ОС з централізованою системою безпеки доступ до ресурсів контролюється централізовано, то матриця доступу до ресурсів буде, на відміну від децентралізованої системи загальносистемною. Представимо її як $A(i,j) \in [0,1]$, де $A(i,j) = 1$ означає, що і-й користувач чи процес має доступ до j-го ресурсу.

Для централізованої системи безпеки характерно, що виконання захисних функцій (наприклад шифрування даних) управляється центральним модулем керування безпекою ЦКМБ. Позначимо ефективність захисту деякого ресурсу r_i як $E(r_i)$, де $E(r_i) \in [0,1]$. При цьому захист тим ефективніший, чим ближче значення $E(r_i)$ до 1. Тоді ймовірність успішного витоку інформації через деякий скомпрометований ресурс r_i можна представити як:

$$P_{\text{витоку}}(r_i) = (1 - E(r_i)) \cdot P_{\text{дискр.}}(r_i), \quad (3)$$

де i – число можливих ресурсів, шляхів де можливий витік інформації; 1-$E(r_i)$- імовірність того, що захисні механізми ОС не подолають вплив загроз; $P_{\text{дискр.}}(r_i)$ - ймовірність компрометації ресурсу r_i через фізичний доступ або в інший спосіб.

Одна із переваг централізованої системи безпеки полягає в тому, що моніторинг стану безпеки ведеться в режимі реального часу. При цьому аналізуються всі події в системі та виявляються потенційні загрози. Аналіз включає виявлення аномальної активності в різних компонентах та спроби несанкціонованого доступу. Позначимо ймовірність виявлення загрози або несанкціонованого доступу до ресурсу r_i як $D(r_i)$, де $D(r_i) \epsilon [0,1]$. Тепер ймовірність успішного витоку інформації через деякий ресурс, не виявлену системою моніторингу централізованої системи безпеки з урахуванням формули (3), можна представити так:

$$P_{\text{не виявл.витоку}}(r_i) = (1 - D(r_i)) \cdot P_{\text{витоку}}(r_i), \quad (4)$$

де i-загальне число ресурсів (шляхів), через які може статись витік інформації; $(1 - D(r_i))$ - загроза або несанкціонований доступ до ресурсу r_i не будуть виявлені; $P_{\text{витоку}}(r_i)$ - ймовірність успішного витоку інформації через деякий скомпрометований ресурс r_i. Добуток ймовірностей (формула 4), враховує той момент, що навіть якщо ресурс скомпрометовано, то витік відбудеться тільки у випадку, коли не спрацювали захисні механізми ОС.

Централізоване управління привілеями мінімізує можливість витоку інформації через зловживання правами доступу. Усі користувачі мають лише ті права, які необхідні для виконання їхніх

завдань, що значно зменшує ризики зловживання правами. Позначимо ймовірність того, що користувач u_n має необхідні привілеї для доступу до ресурсу r_j як $P_{\text{привілеї}}(u_n, r_j)$. Тоді ймовірність витоку через зловживання правами можна представити як:

$$P_{\text{зловж.}}(u_n, r_j) = P_{\text{привілеї}}(u_n, r_j) \cdot P(r_j), \quad (5)$$

де $P_{\text{привілеї}}(u_n, r_j) \cdot -$ фактор, що визначає, наскільки ймовірно, що користувач взагалі має доступ до ресурсу. Якщо доступу немає (значення 0), зловживання неможливе; $P(r_j)$ – фактор, що оцінює, наскільки ресурс вразливий до витоку, якщо хтось має доступ. Він враховує технічні характеристики ресурсу та стан його захисту.

Тоді для централізованої системи безпеки загальна ймовірність витоку інформації по всіх компонентах може бути описана як функція ймовірностей витоку через окремі ресурси та ефективність захисту кожного компонента:

$$P_{\text{витоку ЦС}} = 1 - \prod_{i=1}^{p}(1 - P_{\text{не виявл.витоку}}(r_j) \cdot P_{\text{зловж.}}(u_n, r_j)), \quad (6)$$

де r_j – ресурс (шлях) системи, де можливий витік інформації; u_n – користувач, що взаємодіє з ресурсом r_j ; $p -$ загальне число загроз; $P_{\text{не виявл.витоку}}(r_j) \cdot P_{\text{зловж.}}(u_n, r_j)$ –враховує ситуацію невиявленого витку інформації в ОС через r_j унаслідок зловживання користувача u_n ; $\prod_{i=1}^{p} -$ добуток, що враховує одночасність дії всіх загроз.

Як видно з формули (6), імовірність витоку інформації з використанням централізованої системи безпеки ОС тим нижча, чим кращі управління привілеями та ефективність роботи ЦКМБ щодо управління механізмами захисту ресурсів ОС.

Перейдемо до аналізу ефективності архітектури з огляду на їх стійкість до витоку конфіденційної інформації. Єдиний центр управління доступом і політиками безпеки дозволяє забезпечити узгодженість усіх правил доступу до конфіденційної інформації. Політики можна легко застосовувати до всіх користувачів і ресурсів, що зменшує ймовірність помилок налаштування, які можуть призвести до витоку інформації. Централізоване керування привілеями дозволяє обмежити доступ до конфіденційних даних на основі ролей, що мінімізує ризики несанкціонованого доступу. Саме цим вирізняється ОС з централізованою системою безпекию.

Архітектури безпеки які базуються на використані централізованих систем безпеки ОС показують високу ефективність роботи всіх механізмів безпеки. Централізоване управління дозволяє відслідковувати всі події в системі, що полегшує виявлення потенційних витоків інформації. Система моніторингу (наприклад, MaxPatrol SIEM в Windows Server 2019) дозволяє виявляти аномальну активність у режимі реального часу та оперативно реагувати на загрози. Централізоване ведення журналів подій і аудиту надає більш повну картину доступу до конфіденційних даних. Єдиний стандарт шифрування дозволяє забезпечити захист даних на всіх рівнях системи. При цьому спрощується управління шифруванням та ключами. Крім цього централізована система безпеки ОС дозволяє водночас всім компонентам ОС використовути найсучасніші алгоритми для захисту конфіденційної інформації.

Механізм резервування в рамках централізованої системи безпеки дозволяє більш повно, ніж в децентралізованій системі забезпечити захист конфіденційних даних в разі збоїв або атак ЗПЗ. Усі резервні копії будуть формуватись єдиним чином відповідно до єдиних політик, що значно полегшує їх відновлення інформації після можливих проблем.

Централізована система безпеки ОС дозволяє більш оперативно реагувати на атаки ЗПЗ та змінювати політики безпеки у всій системі одночасно, адаптувати механізми захисту ОС до нових загроз.

Також ефективність централізованої системи безпеки проявляється в більш раціональному використані ресурсів завдяки єдиному центру управління, що проявляється в менших витратах на адміністрування, ніж в децентралізованих системах.

Таким чином, підсумовуючи вище приведене, можна константувати, що централізовані системи безпеки ОС забезпечують більш високу ефективність в роботі своїх механізмів захисту, повноту ситуативного контролю та управлінні всіма аспектами захисту від витоку інформації, значно менші витрати на адміністрування і при цьому дозволяють більш оперативно реагувати на атаки ЗПЗ та забезпечувати узгодженість політик.

Проте всі централізовані системи є вразливими стосовно свого центрального вузла. Це в повній мірі стосується і централізованих систем безпеки ОС. Проте, як завжди є важливий нюанс - забезпечити захист центрального вузла завжди легше, ніж багатьох в децентралізованих системах. Тому, не зважаючи на приведену вразливість, централізовані системи безпеки ОС забезпечують більш ефективний захист інформації загалом та від витоку конфіденційності інформації зокрема.

Були проведені експерименти з двома варіантами ОС, що передбачали тестування контролю доступу, шифрування, контролю виявлення загроз та реагування системи, керування привілеями, аудиту та журналювання. Всі вони підтвердили кращу ефективність та стійкість ОС до витоку інформації із застосуванням централізованої системи безпекию

Висновки

Централізована система безпеки ОС не є абсолютним рішенням, але при націленості ОС на протидію витоку конфіденційної інформації та загалом захист інформації, що обробляється її додатками, вона, як показали експерименти, є кращим рішенням. Її ключова роль у протидії витоку конфіденційної інформації полягає в використанні інтегрованих механізмів захисту та централізованому управлінню всіма аспектами безпеки. Це гарантує координацію заходів безпеки, всебічний, на відміну від децентралізованих систем, моніторинг активності, управління доступом та контроль за діями користувачів і процесів, що значно знижує ризик несанкціонованого доступу до ресурсів або витоку даних.

Таким чином запропоновані моделі централізованої системи безпеки ОС з імплементованими в них захисним механізмом, розробленого методу маркування конфіденційної інформації дозволив створювати інформаційні технології для побудови спеціалізованих захищених ОС, стійких до витоку інформації шляхом деструктивних впливів ЗПЗ та комп'ютерних атак.

В результаті використання перелічених заходів було отримано архітектуру централізованої системи безпеки ОС, яка може використовуватись в захищених ОС призначених для обробки конфіденційної інформації в інформаційних системах, що працюють під їх управлінням.

Результати проведених досліджень інформаційної технології побудови централізованої системи безпеки ОС підтверджують покращений рівень стійкості до витоку конфіденційної інформації, спрощення керування механізмами задання прав доступу до ресурсів.

Література

1. Liu G., Zhang J., Liu J., Zhan Y. Improved Biba model based on trusted computing. *Security and Communication Networks.* 2015. Vol. 8. p. 2793 – 2797. https://doi.org/10.1002/sec.1201
2. Goguen J. A., Meseguer J. Unwinding and Inference Control*, IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1984, p. 75-75, doi: 10.1109/SP.1984.10019.
3. Hahn M.A., Oestreicher D.R., Stevenson R.J. The Evans & Sutherland view of tomorrow's supercomputing. *Digest of Papers. COMPCON Spring 89.* Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, San Francisco, CA, USA, 1989, p. 300-303, doi: 10.1109/CMPCON.1989.301945.
4. Fatima H., Messaoud A., Rachid D., Mounir B.M. Formal Modelling and Implementation of Clark-Wilson Security Policy with FoCaLiZe. *6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, EL OUED, Algeria, 2024, pp. 1-5, doi: 10.1109/PAIS62114.2024.10541223
5. Avorgbedor F., Liu J., Enhancing User Privacy Protection by Enforcing Clark-Wilson Security Model on Facebook. *IEEE International Conference on Electro Information Technology (EIT)*, Chicago, IL, USA, 2020, p. 155-161, doi: 10.1109/EIT48999.2020.9208279.
6. Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 3rd edition. 2020. p. 1232 ISBN-13:978-1119642787
7. Minimum Security Requirements for Federal Information and Information Systems/Federal Information Processing Standards. National Institute of Standarts and Technology USA. FIPS PUB 200. 2006.
8. Information security, cybersecurity and privacy protection - Information security management systems – Requirements. ISO/IEC 27001:2022, Published (Edition 3, 2022).
9. Calcatinge A., Balog J. Mastering Linux Administration - Second Edition: Take your sysadmin skills to the next level by configuring and maintaining Linux systems. Packt Publishing, 2nd ed. Edition. 2024. p. 764 ISBN 978-1837630691
10. Tevault D.A. Mastering Linux Security and Hardening: A practical guide to protecting your Linux system from cyber attacks. Packt Publishing, 3rd Edition. 2023. p. 618 ISBN 978-1837630516
11. Wojslaw D., Adamowicz G. The Linux DevOps Handbook: Customize and scale your Linux distributions to accelerate your DevOps workflow. Packt Publishing. 2023. p. 428 ISBN 978-1803245669
12. Yin J., Ishikawa Y., Takefusa A. A Linux Audit and MQTT based Monitoring Framework for IoT Devices and Its Evaluation. *Journal of Information Processing*. Vol. 32. 2024. p. 586 – 595. DOI:10.2197/ipsjjip.32.586
13. Scott M.L., Brown T. Shared-Memory Synchronization. Springer Cham. 2024. p. 243 https://doi.org/10.1007/978-3-031-38684-8
14. Calavera D., Fontana L. Linux Observability with BPF: Advanced Programming for Performance Analysis and Networking. 1st Edition, O'Reilly, 2019. p. 128 ISBN 978-1492050209
15. Zhao S., Yu X., Luo J., Xie G. Speculation-Free Function Table Construction in LLVM IR for Fine-Grained Control Flow Integrity. *Journal of Circuits, Systems and Computers*. №16. 2023. https://doi.org/10.1142/s021812662350281x
16. Silberschatz A., Galvin P.B., Gagne G. Operating system concepts. Hoboken, NJ : Wiley. 10th. 2018. p 1278. ISBN 9781119320913
17. Kuo H.C., Chen J., Mohan S., Xu T. Set the Configuration for the Heart of the OS: On the Practicality of Operating System Kernel Debloating. *Communications of the ACM*. vol. 65, №5. 2022. p. 101

– 109 http://dx.doi.org/10.1145/3524301

18. Song Y., Dai H., Jiang J., Zhang W. Multikernel: Operating system solution to generalized functional safety. *Security and Safety*. Vol. 2, 2023. p. 14 https://doi.org/10.1051/sands/2023007

19. Gerhorst L., Herzog B., Reif S., Schröder-Preikschat W., Höni T. Fast and Flexible System-Call Aggregation, *11thWorkshop on Programming Languages and Operating Systems (PLOS '21)*, October 25, 2021, Virtual Event, Germany, vol 3487267. 2021. p. 6 https://doi.org/10.1145/3477113

20. Cai P., Karsten M. Kernel vs. User-Level Networking: Don't Throw Out the Stack with the Interrupts. *Proc. ACM Meas. Anal. Comput. Syst*. 2023. vol. 7. article 49. p. 23 https://doi.org/10.1145/3626780

21. Bae B., Kim T., Lee W., Shin Y. Exploiting Memory Page Management in KSM for Remote Memory Deduplication Attac, *International Conference on Information Security Applications 2024*, January 2024; p.16 DOI:10.1007/978-981-99-8024-6_19

22. Kuzuno H., Yamauchi T. Mitigation of privilege escalation attack using kernel data relocation mechanism. *International Journal of Information Security*. 2024. p. 18. https://doi.org/10.1007/s10207-024-00890-4

23. Mushtaq M., Yousaf M.M., Bhatti, M.K. et al. The Kingsguard OS-level mitigation against cache side-channel attacks using runtime detection. *Ann. Telecommun*. 2022. №77, p.731–747. https://doi.org/10.1007/s12243-021-00906-3

24. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:* Technology and Applications (IDAACS), Dortmund, Germany, 2023, p. 265-270, doi: 10.1109/IDAACS58523.2023.10348773

25. da Rocha M., Valadares D.C.G., Perkusich A., Gorgonio K.C., Pagno R.T., Will N.C. Trusted Client-Side Encryption for Cloud Storage. In: Ferguson D., Pahl C., Helfert M. (eds) *Cloud Computing and Services Science. CLOSER 2020. Communications in Computer and Information Science*. 2021. vol 1399. Springer, Cham, p. 1-24 https://doi.org/10.1007/978-3-030-72369-9_1

26. Enomoto S., Kuzuno H., Yamada H. et al. Early mitigation of CPU-optimized ransomware using monitoring encryption instructions. *Int. J. Inf. Secur*. 2024. №23. p.3393–3413. https://doi.org/10.1007/s10207-024-00892-2

27. Seddigh M., Esfahani M., Bhattacharya S., Aref M.R., Soleimany H. KASLR on mobile devices without any use of cache memory (extended version). *J Cryptogr Eng*. 2024. vol 14. p.281–294. https://doi.org/10.1007/s13389-023-00344-y

28. Krishnamurthi S., Hopkins P.W., McCarthy J. *et al.* Implementation and use of the PLT scheme Web server. *Higher-Order Symb Comput.* 2007. №20. p. 431–460. https://doi.org/10.1007/s10990-007-9008-y

29. General Data Protection Regulation (GDPR) Statement The Plan's Processing of Personal Data Under the European Union (EU) GDPR, and Your Rights, Pittsburgh, PA 15222. 2021. p. 4 https://member.allmyhealth.com/GDPR.pdf - (Date of access 13.08.2025). – Screen name.

30. De Oliveira D.B., Casini D., Cucinotta T. Operating System Noise in the Linux Kernel. *IEEE Transactions on Computers*. 2023. №1. p. 196-207. https://doi.org/10.1109/tc.2022.3187351

31. Zhang Y., Wang L., Liu D., Su Y., Zhu Y., Zhang X. Design of Information Security Protection System for Cloud Business System. In: Kountchev R., Patnaik S., Nakamatsu K., Kountcheva R. (eds) *Proceedings of International Conference on Artificial Intelligence and Communication Technologies (ICAICT 2023). ICAICT 2023*. Smart Innovation, Systems and Technologies. 2024. vol. 369. p. 105–122. https://doi.org/10.1007/978-981-99-6956-2_10

32. Savenko O., Sachenko A., Lysenko S., Markowsky G., Vasylkiv N. Botnet detection approach based on the distributed systems. *International Journal of Computing.* 2020. №19(2), p. 190-198. https://doi.org/10.47839/ijc.19.2.1761

33. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A., Savenko B. Information technology for botnets detection based on their behaviour in the corporate area network. *Communications in Computer and Information Science*. 2017. vol. 718. p. 166–181. ISSN: 1865–0929

34. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks. *Communications in Computer and Information Science*. 2018. vol. 860. p. 385-401.

35. Akhtar Z.B. Securing Operating Systems (OS): A ComprehensiveApproach to Security with Best Practices and Techniques. International Journal of Advanced Network. *Monitoring and Controls*. vol 09(1). p. 100-111. doi: 10.2478/ijanmc-2024-0010

36. Tanenbaum A., Bos H. Modern Operating Systems. 5th Edition. Global Edition. Pearson Education. 2023. p. 387

References

1. Liu G., Zhang J., Liu J., Zhan Y. Improved Biba model based on trusted computing. Security and Communication Networks. 2015. Vol. 8. p. 2793 – 2797. https://doi.org/10.1002/sec.1201

2. Goguen J. A., Meseguer J. Unwinding and Inference Control*, IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1984, p. 75-75, doi: 10.1109/SP.1984.10019.

3. Hahn M.A., Oestreicher D.R., Stevenson R.J. The Evans & Sutherland view of tomorrow's supercomputing. *Digest of*

Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, San Francisco, CA, USA, 1989, p. 300-303, doi: 10.1109/CMPCON.1989.301945.

4. Fatima H., Messaoud A., Rachid D., Mounir B.M. Formal Modelling and Implementation of Clark-Wilson Security Policy with FoCaLiZe. *6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, EL OUED, Algeria, 2024, pp. 1-5, doi: 10.1109/PAIS62114.2024.10541223

5. Avorgbedor F., Liu J., Enhancing User Privacy Protection by Enforcing Clark-Wilson Security Model on Facebook. *IEEE International Conference on Electro Information Technology (EIT)*, Chicago, IL, USA, 2020, p. 155-161, doi: 10.1109/EIT48999.2020.9208279.

6. Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 3rd edition. 2020. p. 1232 ISBN-13:978-1119642787

7. Minimum Security Requirements for Federal Information and Information Systems/Federal Information Processing Standards. National Institute of Standarts and Technology USA. FIPS PUB 200. 2006.

8. Information security, cybersecurity and privacy protection - Information security management systems – Requirements. ISO/IEC 27001:2022, Published (Edition 3, 2022).

9. Calcatinge A., Balog J. Mastering Linux Administration - Second Edition: Take your sysadmin skills to the next level by configuring and maintaining Linux systems. Packt Publishing, 2nd ed. Edition. 2024. p. 764 ISBN 978-1837630691

10. Tevault D.A. Mastering Linux Security and Hardening: A practical guide to protecting your Linux system from cyber attacks. Packt Publishing, 3rd Edition. 2023. p. 618 ISBN 978-1837630516

11. Wojslaw D., Adamowicz G. The Linux DevOps Handbook: Customize and scale your Linux distributions to accelerate your DevOps workflow. Packt Publishing. 2023. p. 428 ISBN 978-1803245669

12. Yin J., Ishikawa Y., Takefusa A. A Linux Audit and MQTT based Monitoring Framework for IoT Devices and Its Evaluation. *Journal of Information Processing*. Vol. 32. 2024. p. 586 – 595. DOI:10.2197/ipsjjip.32.586

13. Scott M.L., Brown T. Shared-Memory Synchronization. Springer Cham. 2024. p. 243 https://doi.org/10.1007/978-3-031-38684-8

14. Calavera D., Fontana L. Linux Observability with BPF: Advanced Programming for Performance Analysis and Networking. 1st Edition, O'Reilly, 2019. p. 128 ISBN 978-1492050209

15. Zhao S., Yu X., Luo J., Xie G. Speculation-Free Function Table Construction in LLVM IR for Fine-Grained Control Flow Integrity. *Journal of Circuits, Systems and Computers*. №16. 2023. https://doi.org/10.1142/s021812662350281x

16. Silberschatz A., Galvin P.B., Gagne G. Operating system concepts. Hoboken, NJ : Wiley. 10th. 2018. p 1278. ISBN 9781119320913

17. Kuo H.C., Chen J., Mohan S., Xu T. Set the Configuration for the Heart of the OS: On the Practicality of Operating System Kernel Debloating. *Communications of the ACM*. vol. 65, №5. 2022. p. 101 – 109 http://dx.doi.org/10.1145/3524301

18. Song Y., Dai H., Jiang J., Zhang W. Multikernel: Operating system solution to generalized functional safety. *Security and Safety*. Vol. 2, 2023. p. 14 https://doi.org/10.1051/sands/2023007

19. Gerhorst L., Herzog B., Reif S., Schröder-Preikschat W., Höni T. Fast and Flexible System-Call Aggregation, *11thWorkshop on Programming Languages and Operating Systems (PLOS '21)*, October 25, 2021, Virtual Event, Germany, vol 3487267. 2021. p. 6 https://doi.org/10.1145/3477113

20. Cai P., Karsten M. Kernel vs. User-Level Networking: Don't Throw Out the Stack with the Interrupts. *Proc. ACM Meas. Anal. Comput. Syst*. 2023. vol. 7. article 49. p. 23 https://doi.org/10.1145/3626780

21. Bae B., Kim T., Lee W., Shin Y. Exploiting Memory Page Management in KSM for Remote Memory Deduplication Attac, *International Conference on Information Security Applications 2024*, January 2024; p.16 DOI:10.1007/978-981-99-8024-6_19

22. Kuzuno H., Yamauchi T. Mitigation of privilege escalation attack using kernel data relocation mechanism. *International Journal of Information Security*. 2024. p. 18. https://doi.org/10.1007/s10207-024-00890-4

23. Mushtaq M., Yousaf M.M., Bhatti, M.K. et al. The Kingsguard OS-level mitigation against cache side-channel attacks using runtime detection. *Ann. Telecommun*. 2022. №77, p.731–747. https://doi.org/10.1007/s12243-021-00906-3

24. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:* Technology and Applications (IDAACS), Dortmund, Germany, 2023, p. 265-270, doi: 10.1109/IDAACS58523.2023.10348773

25. da Rocha M., Valadares D.C.G., Perkusich A., Gorgonio K.C., Pagno R.T., Will N.C. Trusted Client-Side Encryption for Cloud Storage. In: Ferguson D., Pahl C., Helfert M. (eds) *Cloud Computing and Services Science. CLOSER 2020. Communications in Computer and Information Science.* 2021. vol 1399. Springer, Cham, p. 1-24 https://doi.org/10.1007/978-3-030-72369-9_1

26. Enomoto S., Kuzuno H., Yamada H. et al. Early mitigation of CPU-optimized ransomware using monitoring encryption instructions. *Int. J. Inf. Secur*. 2024. №23. p.3393–3413. https://doi.org/10.1007/s10207-024-00892-2

27. Seddigh M., Esfahani M., Bhattacharya S., Aref M.R., Soleimany H. KASLR on mobile devices without any use of cache memory (extended version). *J Cryptogr Eng*. 2024. vol 14. p.281–294. https://doi.org/10.1007/s13389-023-00344-y

28. Krishnamurthi S., Hopkins P.W., McCarthy J. *et al.* Implementation and use of the PLT scheme Web server. *Higher-Order Symb Comput.* 2007. №20. p. 431–460. https://doi.org/10.1007/s10990-007-9008-y

29. General Data Protection Regulation (GDPR) Statement The Plan's Processing of Personal Data Under the European Union (EU) GDPR, and Your Rights, Pittsburgh, PA 15222. 2021. p. 4 https://member.allmyhealth.com/GDPR.pdf - (Date of access 13.08.2025). – Screen name.

30. De Oliveira D.B., Casini D., Cucinotta T. Operating System Noise in the Linux Kernel. *IEEE Transactions on Computers*. 2023. №1. p. 196-207. https://doi.org/10.1109/tc.2022.3187351

31. Zhang Y., Wang L., Liu D., Su Y., Zhu Y., Zhang X. Design of Information Security Protection System for Cloud Business System. In: Kountchev R., Patnaik S., Nakamatsu K., Kountcheva R. (eds) *Proceedings of International Conference on Artificial Intelligence and Communication Technologies (ICAICT 2023). ICAICT 2023*. Smart Innovation, Systems and Technologies. 2024. vol. 369. p. 105–122. https://doi.org/10.1007/978-981-99-6956-2_10

32. Savenko O., Sachenko A., Lysenko S., Markowsky G., Vasylkiv N. Botnet detection approach based on the distributed systems. *International Journal of Computing.* 2020. №19(2), p. 190-198. https://doi.org/10.47839/ijc.19.2.1761

33. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A., Savenko B. Information technology for botnets detection based on their behaviour in the corporate area network. *Communications in Computer and Information Science*. 2017. vol. 718. p. 166–181. ISSN: 1865–0929

34. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks. *Communications in Computer and Information Science*. 2018. vol. 860. p. 385-401.

35. Akhtar Z.B. Securing Operating Systems (OS): A ComprehensiveApproach to Security with Best Practices and Techniques. International Journal of Advanced Network. *Monitoring and Controls*. vol 09(1). p. 100-111. doi: 10.2478/ijanmc-2024-0010

36. Tanenbaum A., Bos H. Modern Operating Systems. 5th Edition. Global Edition. Pearson Education. 2023. p. 387