УДК 004.03:004.45:004.75
DOI https://doi.org/10.32782/2663-5941/2025.4.2/38

Стецюк Ю.В.
Хмельницький національний університет

Савенко О.С.
Хмельницький національний університет

ЧАСТКОВО-ЦЕНТРАЛІЗОВАНА СИСТЕМА БЕЗПЕКИ МЕРЕЖЕВИХ ОС З ДИНАМІЧНОЮ ПЕРЕДАЧЕЮ КЕРУВАННЯ МІЖ ЇЇ ВУЗЛАМИ

Проведено аналіз сучасного стану рівня загроз від зловмисного програмного забезпечення (ЗПЗ) сучасним захищеним спеціалізованим ОС, стосовно їх стійкості до витоків конфіденційної інформації. Представлена модель взаємодії захисних механізмів централізованої системи безпеки ОС мережевого вузла, яка дозволяє отримати архітектуру безпеки ОС, позбавлена проблеми витоку конфіденційної інформації при атаках на оперативну пам'ять системи, що досягається включенням до неї механізму маркування, особливістю якого є контроль каналів проходження конфіденційної інформації на рівні сторінок пам'яті. Розглянуті принципи побудови централізованих та частково-централізованих систем безпеки ОС та принципів організації роботи їх механізмів безпеки. Для збереження позитивних якостей централізованої системи безпеки і водно час уникнення проблем властивих будь-якій централізації, запропоновано метод динамічної передачі керування між вузлами комп'ютерної мережі, як способу підвищення її стійкості до витоку конфіденційної інформації та усунення вузького місця централізованої системи безпеки – єдиного центру управління, а також для протидії проведення розвідки зловмисниками комп'ютерної мережі з метою зриву атаки на її ресурси. Його особливістю є синхронізація та актуалізація центральної бази привілеїв та стану мережі шляхом реплікації локальних баз привілеїв мережевих вузлів в процесі передачі керування між ними. Ще одна особливість методу полягає в забезпеченні постійної наявності резервного керівного вузла на випадок збою. Кожен вузол, що передав керування мережею наступнику, визначеного в рандомний спосіб, переходить у резерв. Запропоновано узагальнений алгоритм, що реалізує кроки методу динамічної передачі керування між вузлами комп'ютерної мережі при штатній роботі мережі та для ситуації старту системи, або збою. Розроблено модель станів мережевого вузла в процесі динамічної передачі керування між вузлами, що дозволяє досліджувати його у всіх можливих ситуаціях. Виконано порівняльний аналіз ефективності централізованої та частково-централізованої систем безпеки.

Ключові слова: *система безпеки, централізована система, частково-централізована система, операційна система, модель станів.*

Постановка проблеми. Аналіз сучасного стану рівня загроз від ЗПЗ сьогоднішнім мережевим ОС підтверджує, що їх більша частина обумовлена реалізованим в ОС концептуальним підходом, в основі якого лежить реалізація схеми розподіленого адміністрування механізмів захисту. Таким чином, існуючі на сьогодні ОС не в змозі гарантувати стовідсоткову стійкість до витоку інформації через існування в них вразливостей з однієї сторони, а з іншої через їх знаходження під постійною загрозою зі сторони ЗПЗ, що тільки збільшується. Без впровадження нових підходів до захисту ОС, що основані на централізованих та частково-централізованих системах безпеки, протидія витоку інформації та, загалом, несанкціонованому доступу до конфіденційної інформації буде неефективною.

Аналіз останніх досліджень і публікацій. Операційна система (ОС) є основною, найбільш функціонально складною, програмною системою комп'ютерної системи (КС). Її призначення в управлінні всіма апаратними та програмними ресурсами КС, надання їх, відповідно до встановлених політик безпеки в розпорядження прикладних програм, забезпечення інтерфейсу як для користувачів так і прикладних програм. Сьогоднішні ОС до того ж мають добре розвинуті мережеві функції. Такими є урядові та військові системи, різноманітні фінансові установи та корпоративні ІС, де безпека даних має першорядне

значення. На сьогодні це не тільки чисто наукове або технічне поняття, але і юридичне. Вимоги до таких систем визначені стандартом США, розробленим Національним інститутом стандартів і технологій (NIST), а також в міжнародному стандарті [1].

На сьогодні існує немало систем реагування на інциденти, які можуть бути застосовані для підвищення рівня безпеки [2, 3] в мережах в плані протидії витоку конфіденційної інформації. Встановлено, що ОС, які працюють в складі кіберфізичних систем, які використовують канали передачі даних бездротового зв'язку СВТС э вразливими до атак, через низьку стійкість їх механізму виявлення вторгнень на основі машинного навчання [4, 5], що виявляє лише відомі атаки, які містяться в навчальних наборах даних, і є нечутливим до атак нового типу. З метою усунення проблеми в [6], запропоновано більш досконалий IDS-механізм на основі трансферного навчання для системи СВТС. Вона використовує оптимізований блок одновимірної згорткової нейронної мережі для автоматичного вилучення просторових і часових характеристик з вихідних даних та новий метод передачі знань, що дозволило виявляти і атаки нових типів, в тому числі атаки нульового дня.

В [7, 8] пропонується покращення IDS-систем при роботі ОС з даними в хмарах на основі використання комбінації ядра нечіткої кластеризації (KFCM) і оптимальної нечіткої нейронної мережі типу 2 (OT2FNN), а також механізму захисту E-SGX [9]. В [10] запропоновано розподілену мережеву систему з системою запобігання вторгненням, засновану на удосконаленій структурі алгоритму Spark і динамічній моделі теорії інформаційної безпеки. Ще подібні механізми пом'якшення атак на комп'ютерні мережі запропоновано в [11, 12]. В [13] запропоновано концептуальну модель багатокомп'ютерної системи, що забезпечує функціонування антивірусних приманок та пасток для виявлення шкідливих програм та комп'ютерних атак у мережах. В [14] запропоновано підхід до використання нових інструментів для виявлення шкідливих програм у корпоративних мережах, які є розподіленими системами з частковою централізацією.

Варіанти побудови систем раннього виявлення потенційних каналів ураження КС показано в [15, 16], архітектура якої включає агентів різних типів та певної кількості екземплярів, що спеціалізуються на вирішенні задач виявлення потенційних каналів ураження КС та взаємодіючих між собою, які спільно вирішують загальне завдання виявлення проникнення у КС [17]. В [18] запропоновано метод виявлення ЗПЗ на кінцевих точках на основі ідентифікаторів подій із використанням глибокого навчання. В [19] представлено самоадаптивну систему для забезпечення стійкості корпоративних мереж до кібератак ботнетів, де стійкість забезпечується адаптивною реконфігурацією мережі.

В [20, 21] пропонується використання інтелектуальних методів глибокого навчання в системах виявлення вторгнень в комп'ютерні мережі. В рамках цього підходу реалізовано метод WNIDS, в рамках якого функціонує контролер SDN, який ефективно відстежує та керує пристроями мережі, аналізує набори даних. В [22] представлені два варіанти алгоритму FTNPA: децентралізований та централізований. Перший варіант реалізує локального відновлення, коли мобільний вузол розміщується в певному місці, щоб допомогти постраждалій зоні. Другий варіант використовує централізований метод виявлення, де метод відновлення, створює альтернативні шляхи мобільних вузлів до вузла призначення. Багато уваги приділялось удосконаленню роботи захисних механізмів ОС, а саме її ядру [23]. Запропоновано модель централізованої системи безпеки ОС для нових архітектур ОС [24].

Кроком вперед є рішення доповнення систем безпеки ОС системою авторизації DiSAuth [25], яка забезпечує безпечний алгоритм, заснований на системі доменних імен (DNS) і блокчейні, що забезпечує її стійкість до втручання, зберігає конфіденційність, для захисту даних від витоку. В [26] піднімається питання мінімізації надання привілеїв доступу до ресурсів, як способу підвищення стійкості до витоків інформації, оскільки зловмисники часто використовують вразливості у файлових системах та драйверах дисків для витоку або маніпулювання вмістом файлів програм.

Удосконаленню архітектури систем безпеки сучасних мережевих ОС, що стосується нівелювання загроз витоку інформації, розглянуто в [27], метод динамічної передачі керування між вузлами мережі запропоновано в [28]. Розвитку систем з динамічною зміною центру керування без втручання користувача з метою ускладнення пошуку центру системи для зловмисників та встановлення принципів їх функціонування присвячені роботи [29, 30].

Інтеграція в комп'ютерні мережеві штучного інтелекту є однією з передових технологій. Очі-

кується, що це дозволить вирішити проблеми, з якими стикаються комп'ютерні мережі в протистоянні кібератакам [31].

Постановка завдання. Метою статті є розробка нових методів забезпечення ефективності роботи систем безпеки мережевих ОС, підвищення їх стійкості до витоків конфіденційної інформації шляхом випадкової динамічної передачі керування між вузлами мережі.

Виклад основного матеріалу. Розглянемо графову модель системи безпеки ОС, наведеної на рис. 1. Вершинам графа відповідають функціональні модулі системи безпеки ОС, а направлені ребра вказують на взаємодію між модулями, визначаючи алгоритм її роботи. Основні взаємодії:

– центральний модуль керування ЦМК передає політики безпеки, включаючи політики маркування конфіденційної інформації через модуль керування політиками безпеки МКПБ до периферійних модулів безпеки ПМБ1 – ПМБn;

– периферійні модулі безпеки ПМБ1 – ПМБn передають дані моніторингу через модуль контролю МК центральному модулю ЦМК для аналізу;

– модуль контролю МК передає результати перевірки коректності маркерів конфіденційності модулю маркування ММ;

– модуль маркування ММ передає агентам інформацію про маркування конфіденційної інформації периферійним модулям безпеки ПМБ1 – ПМБn для подальшого застосування на рівні сторінок пам'яті;

– модулі автентифікації МА, авторизації МАвт, шифрування МШ, мережевої безпеки МерМ взаємодіють з модулем маркування ММ при виконанні своїх функцій.

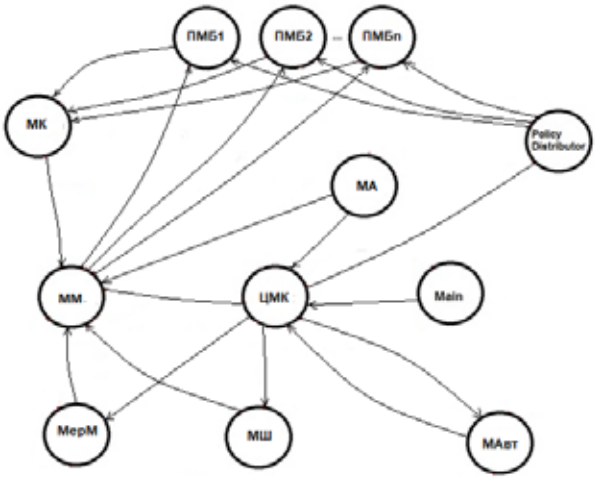

Рис. 1. Модель централізованої системи безпеки ОС з механізмом низькорівневого маркування конфіденційної інформації

Запропонована модель централізованої системи управління безпекою дозволяє отримати архітектуру безпеки ОС, яка буде позбавлена проблеми витоку конфіденційної інформації при низькорівневих атаках ЗПЗ. Це досягається включенням до неї механізму маркування, особливістю якого є контроль каналів проходження конфіденційної інформації на рівні сторінок пам'яті.

Якщо в якості базового вузла комп'ютерної мережі взяти станцію з операційною системою з наведеною моделлю системи безпеки, то отримаємо мережу, де в кожному окремому вузлу буде реалізовано централізоване керування привілеями, але сама мережа буде мати децентралізоване керування, зі всіма проблемами безпеки, що властиве даній моделі. Якщо керування мережею передати одному із вузлів мережі, то отримаємо мережну систему із централізованим керуванням. Така модель керування має ряд значимих переваг, таких як:

– уніфікація політик керування привілеями та керування ресурсами по однакових правилах, політики безпеки та протоколи встановлюються в одному місці, що забезпечує єдність управління мережею, полегшує адміністрування, оскільки зміни в політиках застосовуються одночасно для всієї мережі;

– спрощене управління, оскільки відсутня необхідність координувати дії між кількома керівними модулями – адміністрування здійснюється через один модуль ЦМК (Рис. 1);

– централізований моніторинг, означає, що вся інформація про стан мережі та її вузлів надходить до одного модуля, що значно полегшує виявлення загроз, спрощує аналіз та прийняття рішень.

Разом з тим централізована система керування має ряд потенційно загрозливих проблем. Основні з них:

– так звана «єдина точка відмови», коли при виході з ладу центрального модуля мережа стає вразливою, створюється високий ризик її безперервній роботі;

– вразливість до атак ЗПЗ. Сама наявність центрального вузла керування робить його головною ціллю для різного роду деструкцій, оскільки успішна атака може дати доступ до всієї мережі.

Метою цього дослідження є знаходження такого вирішення, яке б дозволило зберегти основні переваги централізованої системи, одночасно позбувшись її найбільших проблем.

В якості одного із способів подолання недоліків централізованої системи безпеки ОС із збереженням її позитивних якостей, пропонується метод динамічної передачі керування між вузлами мережі в процесі її роботи. В основі методу лежить постійна міграція керівного центра по вузлах мережі, що дозволяє позбутись основного недоліку чисто централізованого підходу – наявності вузького місця у вигляді вразливостей центрального модуля керування.

Згідно запропонованого методу комп'ютерна мережа представлена як множина вузлів $M_{цмк}$, чиї системи безпеки ОС мають модулі центрального керування ЦМК:

$$M_{ЦМК} = \{цмк_1, цмк_2, \ldots, цмк_n\} \quad (1)$$

де n – число вузлів мережі, $цмк_1, цмк_2, \ldots, цмк_n$ – модулі центрального керування вузлів мережі.

При цьому кожен модуль ЦМК вузла зв'язаний з базою привілеїв $B_{привілеїв}$ яка в поточному режимі слугує локальною базою вузла, а коли вузол в процесі динамічної передачі керування приймає функцію центрального модуля керування мережею, вона актуалізується до рівня основної бази привілеїв мережі. Всі локальні бази привілеїв мережевих вузлів можна представити множиною:

$$B_{привілеїв} = \{B_{привілеїв\,1}, B_{привілеїв\,2}, \ldots, B_{привілеїв\,n}\} \quad (2)$$

де n – число вузлів мережі; $B_{привілеїв\,1}, B_{привілеїв\,2}$, $\ldots, B_{привілеїв\,n}$ – локальні бази привілеїв вузлів мережі.

Визначимо стан системи $S_{системи}$ в момент часу t, що відповідає моменту, що передує передачі керування як:

$$S_{системи}(t) = \left(ЦМК_{акт.}(t), ЦМК_{резерв}(t), T_{керівний}\right) \quad (3)$$

де: $ЦМК_{акт.}(t) \in M_{ЦМК}$ – вузол з множини $M_{ЦМК}$, який в момент часу t володіє токеном (є центром керування мережею); $ЦМК_{резерв}(t) \in M_{ЦМК}$ резервний вузол керування в момент часу t в мережі; $T_{керівний}$ – керівний токен мережі.

Як видно з визначення стану мережі (формула 3), особливістю методу є передача функцій центрального модуля керування наступному вузлу шляхом відправки йому керівного такенна. Передачу керівного токена можна описати як перехід:

$$S_{системи}(t_k) = \left(ЦМК_{акт.}(t_k), ЦМК_{резерв}(t_k), T_{керівний}\right) \rightarrow S_{системи}(t_{k+1}) = \left(ЦМК_{акт.}(t_{k+1}), ЦМК_{резерв}(t_{k+1}), T_{керівний}\right) \quad (4)$$

де: $ЦМК_{акт.}(t_{k+1})$ – новий керівний вузол, випадково обраний із множини $ЦМК_{акт.}(t_{k+1}) \in M_{ЦМК} / \{ЦМК_k\}$, правило вибору якого задається формулою:

$$ЦМК_{k+1} \approx Uniform\left(M_{ЦМК / \{ЦМК_k\}}\right) \quad (5)$$

– $Uniform()$ – функція, яка забезпечує однакову ймовірність вибору в якості керівного вузла будь якого вузла з числа активних, за виключенням поточного вузла $ЦМК_k$; $ЦМК_k$ – поточний вузол керування, після передачі керування модулю $M_{ЦМК\,k+1}$, виконує функцію резервного вузла керування; $T_{керівний}$ – керівний токен.

Керівний токен має просту структуру (Рис. 2) і фізично є пакетом даних, який передається між вузлами мережі для визначення поточного керівного модуля (ЦМК). В один момент часу тільки один мережевий вузол володіє токеном $T_{керівний}$. Таким чином він слугує узгодженню стану мережі, передачі керування, уникненню конфліктів і запобігає одночасному виконанню функцій керування кількома вузлами. З метою збереження його цілісності токен підписується цифровим підписом.

Важливим моментом методу динамічної передачі керування є актуалізація основної бази привілеїв $B_{привілеїв\,i+1}$ для вузла, до якого переходить керування комп'ютерною мережею. Вона отримується в процесі процедури реплікації локальних баз привілеїв $B_{локальна\,i}$ активних вузлів мережі в момент зміни керівного центра мережі.

В поточному режимі роботи вузол максимально використовує локальну базу привілеїв,

Фізична структура керівного токена

Поле	Опис
TokenID	Унікальний ідентифікатор токена.
Timestamp	Час створення або останньої передачі токена.
IssuerID	Ідентифікатор вузла, що створив токен.
CurrentHolder	Ідентифікатор вузла, який зараз утримує токен.
Signature	Криптографічний підпис для підтвердження автентичності.
TTL (Time-to-Live)	Максимальний час дії токена.
Payload	Додаткові дані (наприклад, політики безпеки).

```
{ "TokenID": "12ABF13",
  "Timestamp": "2025-01-14 T12:00:00Z",
  "IssuerID": "192.168.0.12",
  "CurrentHolder": "192.168.0.11",
  "Signature": "f1a3c1d...ef23a12",
  "TTL": 720,
  "Payload": { "Policies": ["DAC=read",
  "MAC=yes"]  "Priority": 5 }
```

Рис. 2. Структура керівного токена для синхронізації передачі керування між вузлами мережі

а при передачі йому функцій центрального вузла керування активує основну базу. Передбачається, що на кожному вузлу зберігається його локальна база привілеїв, а основна база керівного модуля отримується шляхом їх реплікації. До реплікації допускаються локальні бази $B_{локальна\,i}$ множини активних вузлів $M_{ЦМК}$, створюючи множину баз привілеїв для реплікації $B_{привілеїв\ репл.}$, яка в загальному випадку може не співпадати з множиною $B_{привілеїв}$:

$$B_{привілеїв\ репл.} \leftarrow Filter\left(B_{локальна\,i}, Active\left(ЦМК_i\right)\right) \quad (6)$$

де: $Filter\left(B_{локальна\,i}, Active\left(ЦМК_i\right)\right)$ – фільтр локальних баз $B_{локальна\,i}$ по умові;

– $Conditium = Active\left(ЦМК_i\right)$ – умовою включення локальної бази $B_{локальна}$ – i-го вузла $ЦМК_i$ до множини реплікації при фільтрації локальних баз є її приналежності до активного вузла.

Реплікація буде успішною, якщо в ній приймають участь локальні бази, які є актуальними. Для деякої $B_{локальна}$ можна визначити її актуальність через функцію $Validate()$:

$Validate\left(B_{локальна\,i}\right) = 1$, тут значення функції рівне одиниці означає, що вона актуальна. Умовою актуальності бази $B_{локальна\,i}$ є її відпадання під наступні умови:

- $Sinc\left(B_{локальна\,i}\right) = 1$ – база синхронізована;
- $t_{оновлення}\left(B_{локальна\,i}\right) \le \Delta t_{max}$ – час останнього її оновлення, який не перевищує деякий встановлений максимальний термін Δt_{max};
- $Integririty\left(B_{локальна\,i}\right)$ – локальна база привілеїв не є пошкоджена.

З урахуванням того, що кожна локальна база привілеїв $B_{локальна}$ повинна знаходитись в актуальному стані, правило включення нових елементів до множини $B_{привілеїв\ репл.}$ буде тепер виглядати наступним чином:

$$B_{привілеїв\ репл.} \leftarrow Filter\left(B_{локальна\,i}, Active\left(ЦМК_i\right) \wedge Validate\left(B_{локальна\,i}\right) = 1\right) \quad (7)$$

де: $Filter(B_{локальна\,i}, Active\left(\text{ЦМК}_i\right)$ – умовою участі в формуванні центральної бази привілеїв $B_{\text{привілеїв репл.}}$ є приналежність $B_{\text{локальна}\,i}$ до активного мережевого вузла; $Validate\left(B_{\text{локальна}\,i}\right)$ – функція яка повертає значення актуальності локальної бази привілеїв $B_{\text{локальна}}$ і-го мережевого вузла.

З урахуванням цього правила отримаємо формулу для всіх елементів множини локальних баз привілеїв, що підлягають реплікації:

$$B_{привілеїв\ репл.} = \{B_{локальна\,i} | i \in M_{ЦМК}, Active\left(ЦМК_i\right) \wedge Validate\left(B_{локальна\,i}\right)\} \quad (8)$$

де: $B_{локальна\,i}$ – локальна база привілеїв, де $i \in M_{ЦМК}$ та її приналежність до активного мережевого вузла; $Validate\left(B_{\text{локальна}\,i}\right)$ – функція яка повертає значення актуальності локальної бази привілеїв $B_{\text{локальна}}$ і-го мережевого вузла.

Тепер, при виконані операції реплікації над елементами множини $B_{\text{привілеїв репл.}}$ буде отримана актуальна централізована база привілеїв для центрального вузла керування мережею, в якій відсутні некоректні дані локальних не актуальних баз привілеїв та локальних баз привілеїв неактивних вузлів:

$$B_{привілеїв\ центр.} = Merge\left(B_{привілеїв\ репл.}\right) \quad (9)$$

де: $Merge()$ – функція реплікації множини локальних баз привілеїв $B_{привілеїв\ репл.}$, що підпадають під реплікацію.

Передача керування від поточного центрального модуля до наступного буде успішною, якщо комп'ютерна мережева система буде знаходить в стані $S_{системи}\,t(i)$ та для наступного i+1-го вузла керування буде наявна актуальна основна база привілеїв $B_{привілеїв\ i+1}$. При цьому вузол, що тільки но передав керування наступному центральному вузлу, переходить у резерв, підтримуючи свою спроможність повернутись до функції центрального вузла.

Такий підхід в реалізації методу практично унеможливлює втрату бази привілеїв і не потребує постійного доступу до глобальної мережі, як у випадку її зберігання у хмарних сховищах, або реалізації у вигляді транзакцій в блокчейн – мережі підтримуючи високий рівень автономності, властивий для корпоративних мереж. При цьому також забезпечується висока стійкість комп'ютерної мережі в цілому до впливів різного роду деструкцій і, відповідно зменшення ймовірності витоку конфіденційної інформації. Таким чином метод забезпечує безперервність роботи мережі, узгодженість даних і захищеність системи.

Тепер можна сформулювати основні кроки методу динамічної передачі керування:

Крок 1. Підготовка до передачі керування. Оцінюється стан, в якому знаходиться поточний центральний модуль керування ЦМК_i (рис. 1) множини $M_{\text{ЦМК}}$, де перевіряється, наявність признаків (завершення часового терміну, перевантаження системи, планове технічне обслуговування). При активному значені хоча б одного з них, поточний центральний модуль має передати керівні функції іншому вузлу мережі ЦМК_{i+1}. При наявності хоча б одного признака, поточний модуль визначає наступний модуль ЦМК_{i+1} з числа активних вузлів, який прийме керування мережею. Затим поточний модуль ЦМК_i перевіряє актуальність

центральної бази привілеїв $B_{\text{привілеїв } i+1}$ вузла, який має прийняти функції центрального вузла керування. При необхідності виконується її реплікація або синхронізація.

Крок 2. Передача поточного стану системи, яка включає передачу активних сесій, політик безпеки та стану моніторингу системи, даних журналів подій. Поточний модуль ЦМК$_i$ ініціює перевірку коректності отриманих даних модулем-наступником ЦМК$_{i+1}$ та актуальності бази привілеїв $B_{\text{привілеїв } i+1}$ після той повідомляє поточний модуль ЦМК$_i$ про свою готовність виконувати функції центрального вузла.

Крок 3. Активація нового модуля. Поточний модуль ЦМК$_i$ відправляє наступнику сповіщення у вигляді керівного такенна $T_{\text{керівний}}$, при цьому всім іншим активним вузлам мережі передається інформація про нового модуля ЦМК$_{i+1}$. Важливою особливістю є те, що в кожен момент часу тільки один вузол з множини $M_{\text{ЦМК}}$ володіє керівним токеном $T_{\text{керівний}}$, який забезпечує його керівну роль у мережі.

Старий модуль ЦМК$_i$ деактивується, але при цьому залишається у готовності як резервний модуль, періодично синхронізуючи свою базу привілеїв та журналів подій у системі з базою та журналами нового центрального вузла.

Крок 4. Отримавши керівний токен $T_{\text{керівний}}$ (рис. 2), вузол активує себе в якості центрального модуля ЦМК$_{i+1}$, беручи на себе контроль за системою. Його першим кроком є фіксація в журналах всіх дій, пов'язаних з фактом передачі йому керування, включаючи час передачі, ідентифікатори модулів ЦМК активних вузлів, стан бази привілеїв. Крім того новий модуль ЦМК$_{i+1}$ сповіщає всі активні вузли системи, що він готовий виконувати функції керівника мережевої системи безпеки.

Крок 5. Моніторинг та контроль системи. На цьому кроці новий модуль ЦМК$_{i+1}$ здійснює моніторинг стану безпеки системи з метою виявлення можливих збоїв або деструкцій. Постійно інформує резервний модуль ЦМК$_i$ про зміни в основній базі привілеїв з метою їх підтримки в стані максимальної синхронізації. Відслідковує стан признаків передачі керування мережею наступному вузлу.

З метою більшої деталізації роботи запропонованого методу відобразимо кроки методу у вигляді алгоритму (рис. 3 та рис. 4).

При цьому слід наголосити, що даний алгоритм є законом функціонування кожного вузла комп'ютерної мережі, але в кожен момент часу вони знаходиться на різних стадіях його виконання. Важливою його особливістю в реалізації динамічної передачі керування між центральними модулями CSMM системи безпеки вузлів мережі ґрунтується на правилі безперервності роботи мережі та недопущені витоку конфіденційної інформації.

Загалом процес роботи комп'ютерної мережі, заснованої на методі динамічної передачі керування, поділяється на кілька етапів – процесу запуску мережі, основного етапу та етапу виходу із збою.

Основний етап це циклічний рух по траєкторії, що включає в себе вершини графа станів (рис. 5) 3 – 4 – 5 – 6 – 3 …. Тут третя вершина графа відповідає режиму очікування. Це звичайний режим, в якому вузол мережі знаходиться основну частину часу, виконуючи покладені на нього функції. Його модуль центрального керування ЦМК знаходиться в підпорядкуванні модуля ЦМК вузла, який в даний момент здійснює керування системою безпеки всієї мережі і, забезпечує локальне керування системою безпеки вузла відповідно до політик безпеки, які проводить центральний модуль.

З цього стану мережевий вузол може перейти в перехідний режим (вершина 4 рис. 3). В цьому режимі модуль ЦМК вузла отримав керівний токкен і знаходиться у процесі прийняття керування системою безпеки мережевої ОС – отримує інформацію про стан системи, активні сесії та базу привілеїв, перевіряє узгодженість даних між вузлами, що відповідає виконанню алгоритму (оператори 6–8 рис. 3).

З стану перехідного режиму вузол, при штатному розвитку подій переходить до стану, в якому він здійснює функції центрального модуля ЦМК (вершина 5 рис. 4), по іншому – керує системою безпеки мережевої ОС, що відповідає виконанню алгоритму (оператори 9–11 рис. 4). В цьому стані вузол знаходиться впродовж терміну, на який його призначено виконувати функції центрального модуля керування системою безпеки мережі, або до початку нештатної ситуації (ребро 5–7 рис. 5). Якщо при виконані алгоритму (оператор 11 рис. 3) буде виявлено хоча б один признак передачі керування (закінчення терміну, втручання адміністратора системи), то вузол переходить в стан резерву (вершина 6 рис. 3). В цьому режимі він знаходиться в стані готовності повернутись до виконання функцій центрального модуля ЦСМ.

Це означає, що основна база привілеїв цього вузла постійно підтримується в актуальному

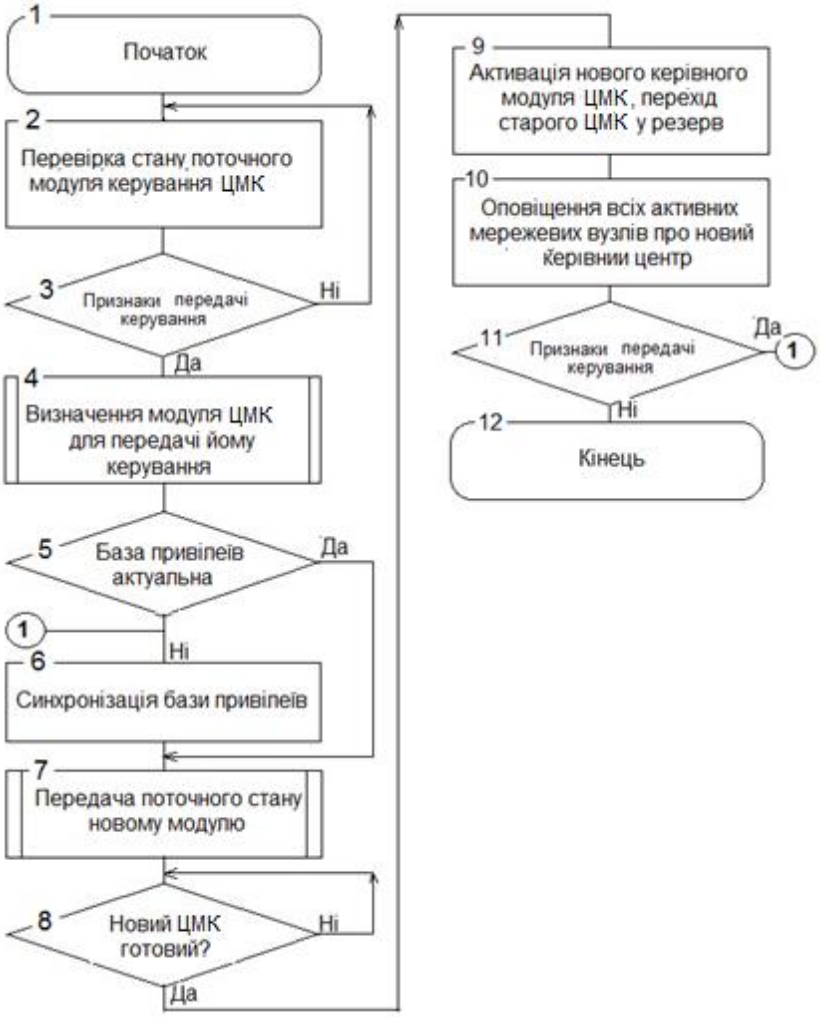

Рис. 3. Алгоритм передачі керування між вузлами комп'ютерної мережі

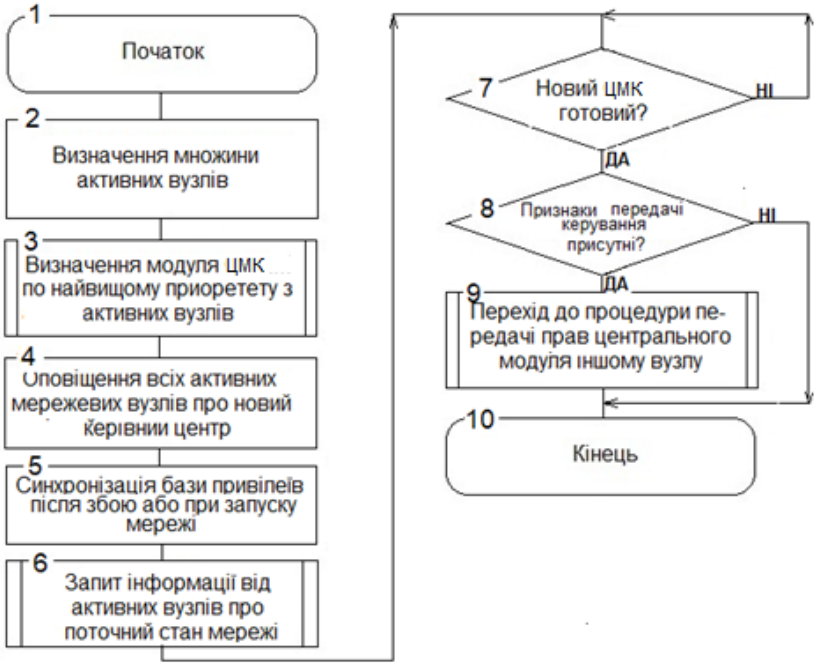

Рис. 4. Алгоритм встановлення керівного центра при старті або при збоях мережі

стані, що дозволяє максимально швидко замінити при необхідності поточний центральний вузол у випадку нештатного розвитку подій (ребро 6–5 рис. 5). При штатному розвитку подій, після наступної ітерації динамічної зміни керівного центра, вузол повернеться до режиму очікування (вершина 3 рис. 5), замикаючи основну послідовність штатних станів, в яких може знаходитись вузол комп'ютерної мережі.

При виникненні нештатної ситуації локального збою, будь який вузол, незважаючи на поточний стан в якому він знаходиться, перейде до аварійного стану (вершина 7 рис. 5). При цьому він буде виключений з числа активних вузлів мережі. З цього стану, відповідно до графа станів (рис. 5) вузол переходить в стан недоступності (вершина 1 рис. 5).

В цьому стані, після усунення причини збою очікує, повторного підключення до мережі. При підключенні до мережі автоматично переходить у стан синхронізації (вершина 2 рис. 5). В цьому стані виконується оновлення до актуального стану локальної бази привілеїв та політик, виконується перевірка узгодженості з іншими вузлами. Завершуються процеси синхронізації включенням вузла до списку активних, а сам вузол переходить у стан очікування (вершина 3 рис. 5).

При запуску мережі або при масштабних збоях відновлення роботи мережі виконується відповідно до алгоритму приведеного на рис. 3. Він враховує ситуації, які можуть виникнути при відновлені роботи мережі після збою. Оскільки при запускові мережі не всі вузли одночасно відновлюють свою роботу в мережі, то перший вузол, який візьме на себе керування буде визначатись як вузол з найвищим його пріоритетом серед вузлів, що на цей момент є активними (оператор 2 рис. 4).

Визначений в такий спосіб вузол, чий модуль ЦМК буде виконувати функцію центрального модуля системи безпеки ОС, генерує керівний токен і розсилає всім іншим активним вузлам мережі, що він є керівний центр системи (оператор 4 рис. 4).

В подальших кроках алгоритм передбачає синхронізацію бази привілеїв з локальними базами активних вузлів з метою відтворення актуальної основної бази привілеїв (оператор 5 рис. 4). Наступним кроком алгоритму є збір інформації про відкритті сесії та іншої інформації про стан мережі (оператор 6 рис. 3). В подальшому алгоритм роботи центрального модуля керування при запускові мережі після збою співпадає з алгоритмом приведеним на рис. 3.

Основна послідовність станів активного вузла мережі описується переходами між вершинами графа станів 3–4–5–6–3–… (Рис. 5). Таким чином кожен вузол має чотири основних стани. Для спрощення розуміння, вершини 4–5 графа можна умовно об'єднати в одну. Тоді основна послідовність переходів графа (Рис. 5) для вузла буде обмежена трьома станами: режим очікування $ЦМК_{Оч}$; режим керівного мережевого центра $ЦМК_{КМЦ}$; режим резервного модуля $ЦМК_{РЕЗ}$.

Інші стани графа переходів мають свою спеціалізації, яка лежить поза основною послідовністю. Так стани, що відповідають вершинам 1 та 2 графа (рис. 5) слугують для виведення вузла на основну траєкторію. Вершині 7 відповідає аварійний стан вузла, який в фізичній системі завжди має місце. Вихід з цього стану можливий через вершини 1 та 2 графа.

Частково-централізована система безпеки ОС базується на застосуванні запропонованого методу динамічної передачі керування між вузлами комп'ютерної мережі. Для оцінки його ефек-

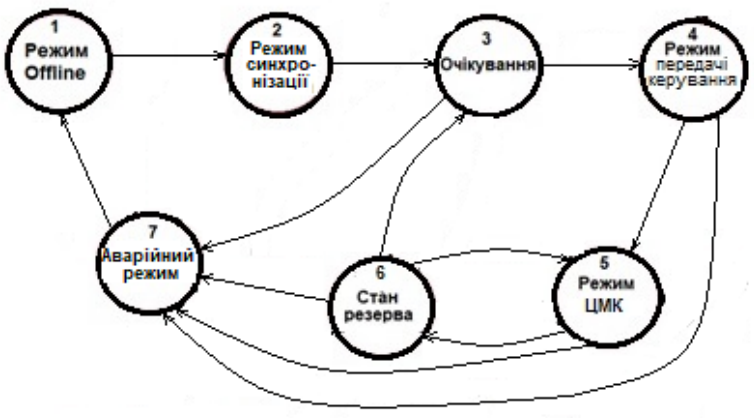

Рис. 5. Граф станів вузла мережі при динамічній передачі керування

тивності скористаємось сукупністю математичних та ймовірнісних моделей, що враховують такі показники: ймовірність витоку конфіденційної інформації, яка враховує ефективність захисних механізмів ОС мережевого вузла та вразливість КС $P_{витоку}$; час виявлення загрози $T_{вияв.\ загрози}$; час обробки операцій безпеки $T_{\text{обробки}}$.

З урахуванням вище сказаного формула ефективності комп'ютерної мережі $E_{системи}$, в якій будуть поєднані всі три параметри з урахуванням їх ваги та способу впливу на значення ефективності прийме вигляд:

$$E_{системи} = w_1 \times \left(1 - P_{витоку}\right) + \\ + w_2 \times \left(\frac{1}{T_{вияв.\ загрози}}\right) + w_2 \times \left(\frac{1}{T_{\text{обробки}}}\right) \quad (10)$$

де: w_1, w_2, w_3 – вагові коефіцієнти параметрів ефективності; $1 - P_{витоку}$ – ймовірність відсутності витоку конфіденційної інформації; $T_{вияв.\ загрози}$ – час виявлення загрози; $T_{\text{обробки}}$ – час обробки операцій безпеки керівним вузлом мережі.

Вираз формули (10) відповідає всім співвідношенням для централізованої системи безпеки ОС $E_{ЦС}$, тому перепишемо її з урахуванням прийнятих для неї позначень:

$$E_{ЦС} = w_1 * \left(1 - P_{витоку\ ЦС}\right) + \\ + w_2 * \frac{1}{T_{вияв.\ загрози\ ЦС}} + w_3 * \frac{1}{T_{обробки\ ЦС}}) \quad (11)$$

де: w_1, w_2, w_3 – вагові коефіцієнти параметрів ефективності; $1 - P_{витоку\ ЦС}$ – ймовірність відсутності витоку конфіденційної інформації в ОС з централізованою системою безпеки; $T_{вияв.\ загрози\ ЦС}$ – час виявлення загрози в системі з централізованою системою безпеки; $T_{\text{обробки ЦС}}$ – час обробки операцій безпеки керівним вузлом мережі з централізованою системою безпеки.

Оскільки робота частково-централізованої системи основана на використані метода динамічної передачі керування між вузлами мережі, цей факт має знайти своє відображення в формулі її ефективності. Цей метод зменшує вразливість системи безпеки ОС, але одночасно, зменшує в деякій мірі і продуктивність системи загалом. Динамічна зміна керівного центра потребує витрат часу на синхронізацію ресурсів безпеки, що підлягають централізованому керуванню. Позначимо його як $T_{\text{витрати синхр. ЧЦС}}$. З урахуванням цього параметра формула ефективності для частково-централізованої системи $E_{ЧЦС}$ прийме вигляд:

$$E_{ЧЦС} = w_1 * \left(1 - P_{витоку\ ЧЦС}\right) + w_2 * \frac{1}{T_{вияв.\ загрози\ ЧЦС}} + \\ + w_3 * \frac{1}{T_{обробки\ ЧЦС} + T_{\text{витрати синхр. ЧЦС}}} \quad (12)$$

де: w_1, w_2, w_3 – вагові коефіцієнти параметрів ефективності; $1 - P_{витоку\ ЧЦС}$ – ймовірність відсутності витоку конфіденційної інформації в ОС з частково-централізованою системою безпеки; $T_{вияв.\ загрози\ ЧЦС}$ – час виявлення загрози в системі з частково-централізованою системою безпеки; $T_{\text{обробки ЧЦС}}$ – час обробки операцій безпеки керівним вузлом мережі з частково-централізованою системою безпеки; $T_{\text{витрати синхр. ЧЦС}}$ – витрати часу на синхронізацію централізованих ресурсів безпеки.

Тепер порівняємо показники ефективності частково-централізованої та централізованої систем безпеки ОС:

$$E_{ЧЦС} - E_{ЦС} > 0 \quad (13)$$

Таким чином, якщо вираз (13) буде матиме значення більше нуля, то це означитиме, що метод динамічної передачі керівного центра між вузлами забезпечує більшу ефективність системи в порівняні з централізованою системою безпеки.

З метою визначення ефективності метода динамічної передачі керування було проведено кілька експериментів. Для цього було розгорнуто тестове середовище основане на використані віртуальної комп'ютерної мережі до якої входять віртуальні вузли ВМ01 – ВМ04, які слугували під час експериментів цільовими машинами, а також атакуюча машина ВМ00, з допомогою програмного забезпечення якої будуть симулюватись різнотипні атаки по можливих шляхах витоку конфіденційної інформації. На атакуючій машині встановлена ОС Kali Linux та симулятор атак Metasploit Framework. Цільові машини керуються ОС FreeBSD 13.1 з імплементованою централізованою системою безпеки. Всі віртуальні машини працюють під управлінням гіпервізора Virtual Box, що об'єднує їх у відокремлену локальну віртуальну мережу, необхідну для проведення експериментів.

Метою експерименту є отримання даних стосовно комплексної стійкості захисних механізмів під впливом загроз, що діють по можливих шляхах витоку інформації, які враховують такі параметри як ймовірність витоку інформації, час виявлення атаки ЗПЗ, продуктивність центрального вузла мережі та час передачі керування між вузлами мережі. Експеримент включає симуляцію

комп'ютерних різнотипних атак, направлених по всіх можливих шляхах витоку конфіденційної інформації всіх вузлів віртуальної мереж.

Для симуляції атак різних видів використовувався Metasploit Framework атакуючої машини ВМ00. Під його управлінням запускався скрипт-файл, який містить інструкції на мові Ruby, алгоритм якого включає циклічну симуляцію атак по всіх основних шляхах витоку по всіх вузлах мережі. В момент роботи скрипт-файла на всіх цільових машинах були відкриті файли /tmp/k1/test.txt, що імітують ресурси з конфіденційною інформацією. Результатами експерименту є інформація про кількість успішних та неуспішних атак на відповідні вузли віртуальної комп'ютерної мережі.

Окрім вище приведеної інформації на стороні атакуючої машини ВМ00 в файлі Result_atack.txt фіксувались дані початку кожної атаки, IP-адреса цільової машини, а також тип атаки. Якщо система безпеки відповідної цільової машини, на яку був спрямований вектор атаки, виявить загрозу, то після її блокування по факту цієї події зробить записи в своїх системних журналах security.log та audit.log. Різниця між часом старту симуляції зафіксованої в файлі Result_atack.txt атакуючої машини та часом її виявленням зафіксованим в журналі security.log відповідної цільової машини буде відповідати терміну виявлення загрози.

Результати співставлення даних з файлу Result_atack.txt атакуючої машини ВМ00 (рис. 6) та файлів security.log (рис. 7) відповідних цільових машин ВМ01-ВМ04 представлені в таблиці 1 в усередненому та унормованому вигляді. Оскільки атакуюча машина та цільові машини знаходяться в одному часовому просторі фізичної машини, то похибка вимірювання тривалості періоду виявлення атаки буде мінімальною. Ситуація коли запис початку симуляції, зафіксованій в файлі Result_atack.txt, немає зв'язаного з ним запису в журналах security.log всіх цільових машин, означає, що мала місце успішна атака. Приклад співставлення показано на рис. 6 та рис. 7.

Ще один етап експериментів мав на меті отримання даних, що дозволять визначити продуктивність роботи керівного вузла комп'ютерної мережі $T_{прод.ЦМК}$ в умовах централізованого керування. При цьому вимірювався час виконання запитів центральним вузлом мережі при виключених механізмах захисту в одному випадку і включених в другому, позначених як $T_{викл.\ МЗ}$ та $T_{вкл.\ МЗ}$ відповідно. Тоді продуктивність центрального вузла мережі визначиться як:

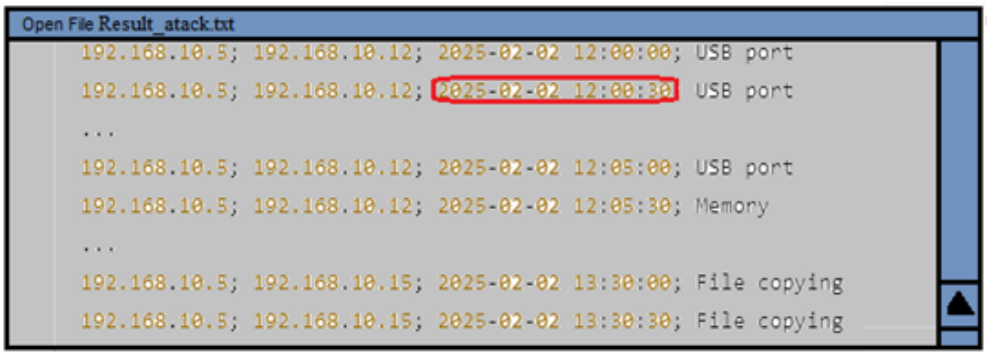

Рис. 6. Вміст файлу Result_atack.txt згенерованого скрипт-файлом симулятора Metasploit Framework

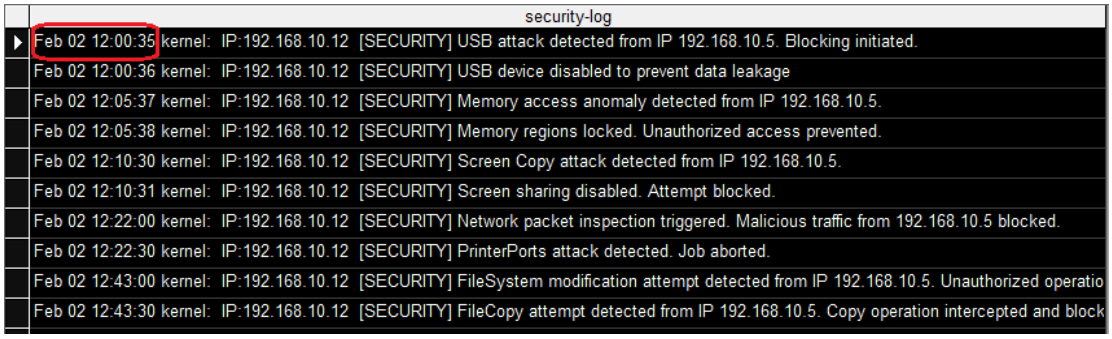

Рис. 7. Вміст файлу security.log цільової машини ВМ1

$$T_{прод.ЦМК} = \frac{T_{викл.\ МЗ}}{T_{вкл.\ МЗ}} \qquad (14)$$

Опрацьовані дані експерименту приведені в таблиці 1.

1.1. Розрахунки ефективності централізованої та частково-централізованої системи безпеки ОС

Скористаємось формулою (14) та даними експериментів розрахуємо продуктивність центрального вузла для централізованої системи $T_{прод.\ ЦС}$, для чого підставимо в неї відповідні значення з таблиці 1:

$$T_{прод.\ ЦС} = \frac{T_{викл.\ МЗ}}{T_{вкл.\ МЗ}} = \frac{0{,}126}{0{,}142} = 0{,}887 \qquad (15)$$

Так само розраховується продуктивність центрального вузла для частково-централізованої системи $T_{прод.\ ЧЦС}$:

$$T_{прод.\ ЧЦС} = \frac{T_{викл.\ МЗ}}{T_{вкл.\ МЗ}} = \frac{0{,}117}{0{,}127} = 0{,}921 \qquad (16)$$

Для розрахунку ефективності системи безпеки скористаємось формулою (10). Для цього визначимо значення вагових коефіцієнтів w_1 - w_3 присутніх в даній формулі. Щоб визначитись з їх вагою скористаємось методом лінійної ієрархії в рамках якого застосуємо шкалу важливості від одного до дев'яти: 1 – рівна важливість; 3 – слабка перевага одного фактора над іншим; 5 – сильна перевага; 7 – дуже сильна перевага; 9 – абсолютна перевага. Керуючись цією шкалою будуємо матрицю парних порівнянь:

Керуючись цією шкалою будуємо матрицю парних порівнянь:

$$M(w) = \begin{bmatrix} 1 & 3 & 5 \\ 1/3 & 1 & 3 \\ 1/5 & 1/3 & 1 \end{bmatrix}$$

Стовпці та рядки матриці є ваговими коефіцієнтами $w_1 - w_3$, а її елементи є результатом попарного співставлення. Виконавши подальші кроки отримаємо: $w_{1=0,633}$; $w_{2=0,261}$; $w_{3=0,106}$. Тепер можна безпосередньо розрахувати ефективність системи безпеки, для чого підставимо значення з таблиці 1 в формулу (11) для централізованої системи безпеки ОС:

$$E_{ЦС} = 0{,}633 * (1 - 0{,}0133) + \\ + 0{,}261 * \frac{1}{0{,}4} + 0{,}106 * \frac{1}{0{,}887} = 1{,}397$$

Аналогічно для частково-централізованої системи безпеки ОС скористаємось формулою (12):

$$E_{ЧЦС} = 0{,}633 * (1 - 0{,}0115) + 0{,}261 * \frac{1}{0{,}308} + \\ + 0{,}106 * \frac{1}{0{,}921 + 0{,}688} = 1{,}5389$$

Використаємо нерівність (13) для порівняння розрахованих значень ефективності двох варіантів систем безпеки ОС:

$$E_{ЧЦС} - E_{ЦС} = 1{,}5389 - 1{,}397 = 0{,}142$$

З неї видно, що ефективність частково-централізованої системи безпеки $E_{ЧЦС}$ в порівнянні з централізованою системою вища одиниці. Для зручності відобразимо отриманий результат у відсотках:

$$\left(1 - \frac{E_{ЦС}}{E_{ЧЦС}}\right) * 100\% = \left(1 - \frac{1{,}397}{1{,}539}\right) * 100\% = 9{,}2\%$$

Виконаний розрахунок показав, що ефективність частково – централізованої системи безпеки ОС, незважаючи на її складність, є помітно вищою. В свою чергу вища ефективність такої системи безпеки означає більшу стійкість до атак ЗПЗ, що в свою чергу забезпечує нижчу вразливість до витоку конфіденційної інформації.

Висновки. Запропонований метод динамічної передачі керування між вузлами мережевих вузлів з підтримкою реплікації основної бази привілеїв для частково-централізованої системи безпеки ОС дозволив зберегти основні переваги централізованої системи безпеки, особливо, що стосується стійкості системи до витоку інформації і, одночасно позбутись найбільшої її вади – вузького місця в системі безпеки у вигляді центрального

Таблиця 1

Усереднені дані експериментів (абсолютне значення/унормоване значення)

Система безпеки ОС	$P_{витоку}$	$T_{вияв.\ загрози}$ **сек**	$T_{викл.\ МЗ}$ **сек**	$T_{вкл.\ МЗ}$ **сек**	$T_{витрати\ синхр.}$ **сек**
Централізована	0,0133	5,2 (0,4)	0.126 (0,887)	0,142	-
Частково-централізована	0,0115	4,925 (0,358)	0,117 (0,921)	0,127	2.118 (0,688)

модуля безпеки, вихід з ладу якого призводить до втрати контролю над всією системою.

Ефективність роботи системи безпеки ОС, що використовує даний метод керування комп'ютерною мережею в порівнянні з чисто централізованим підходом вища щонайменше на 9,2 відсотка.

Ще однією перевагою цього методу динамічної передачі керування є те, що періодична рандомна зміна керівного вузла мережі з великою долею вірогідності робить неактуальними результати проведення розвідки мережі зловмисником, чим зриває атаки на мережу ще до їх початку. Це пояснюється тим, що час розвідки мережі значно більший періоду зміни керівного центра мережі, що дає йому можливість таким чином ухилятись від атак.

Список літератури:

1. Security and Privacy Controls for Information Systems and Organizations / *National Institute of Standards and Technology Walter Copan*, NIST Director and Under Secretary of Commerce for Standards and Technolog, Publication 800-53 Revision 5. DOI:10.6028/NIST.SP.800-53r5

2. Hajimirzaei B., Navimipour N.J. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. *ICT Express*, 2019. Vol 5(1). P. 56–59. DOI:10.1016/j.icte.2018.01.014

3. Kim Y., Park G., Kim H. Domain knowledge free cloud-IDS with lightweight embedding method. *J Cloud Comp.* 2024. № 13, p.143. DOI:10.1186/s13677-024-00707-8

4. Al-Ambusaidi M., Yinjun Z., Muhammad Y. ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications. *Soft Comput.* 2024. 28. p.1765–1784. DOI: 10.1007/s00500-023-09452-7

5. Pourardebil Khah Y., Hosseini Shirvani M., Motameni H. A hybrid machine learning approach for feature selection in designing intrusion detection systems (IDS) model for distributed computing networks. *J Supercomput.* 2025. Vol. 81. 254. DOI: 10.1007/s11227-024-06677-7

6. Lu H., Zhao Y., Song Y., Yang Y., He G., Yu H., Ren Y. A transfer learning-based intrusion detection system for zero-day attack in communication-based train control system. *Cluster Comput.* 2024. Vol. 27. p. 8477–8492. DOI: 10.1007/s10586-024-04376-9

7. Srilatha D., Shyam G.K., Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network. *Cluster Comput*. 2021. Vol 24, P. 2657–2672 DOI: 10.1007/s10586-021-03281-9

8. Hajimirzaei B., Navimipour N.J. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. *ICT Express.* 2019. Vol 5. № 1. P. 56–59 DOI: 10.1016/j.icte.2018.01.014

9. Lang F., Li H., Wang W., Lin J., Zhang F., Pan W., Wang Q. E-SGX: Effective Cache Side-Channel Protection for Intel SGX on Untrusted OS / Y. Wu, M. Yung et al. Information Security and Cryptology. Inscrypt 2020. *Lecture Notes in Computer Science*. 2021. Vol 12612. P. 221–243 DOI: 10.1007/978-3-030-71852-7_15

10. An L., Qiu J., Zhang H., Liu C. Design of distributed network intrusion prevention system based on Spark and P2DR models. *Cluster Comput*. 2024. P. 10757–10776. DOI: 10.1007/s10586-024-04487-3

11. Jain A.K., Shukla H., Goel D. A comprehensive survey on DDoS detection, mitigation, and defense strategies in software-defined networks. *Cluster Comput*. 2024. Vol. 27. P. 13129–13164 DOI:10.1007/s10586-024-04596-z

12. Mushtaq M., Yousaf M.M., Bhatti M.K., Lapotre V., Gogniat, G. The Kingsguard OS-level mitigation against cache side-channel attacks using runtime detection. *Ann. Telecommun*. 2022. Vol 77. p. 731–747. DOI: 10.1007/s12243-021-00906-3

13. Kashtalian A., Lysenko S., Savenko O., Nicheporuk A., Sochor, T., Avsiyevych V. Multi-computer malware detection systems with metamorphic functionality. *Radioelectronic and Computer Systems.* 2024. № 1. p. 152–175. DOI: 10.32620/reks.2024.1.13

14. 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). Germany. Dortmund. 2023. p. 265–270, DOI: 10.1109/IDAACS58523.2023.10348773

15. Soltani M., Khajavi K., Jafari Siavoshani M. A multi-agent adaptive deep learning framework for online intrusion detection. *Cybersecurity.* 2024. Vol. 7. № 9. P. 1–25 DOI:10.1186/s42400-023-00199-0

16. Wang S., Jing Y., Wang K., Wang X., Multi-Agent Systems for Collaborative Inference Based on Deep Policy Q-Inference Network. *J Grid Computing*. 2024. Vol. 22. № 38. DOI:10.1007/s10723-024-09750-w

17. Karataiev O. Towards multi-agent platform development. *Computer Systems and Information Technologies.* 2024. № 4, p. 98–106. DOI: 10.31891/csit-2024-4-12

18. N. Pradeep, Vaishnavi, V. Varsha, K. Vivek. The Application of Artificial Intelligence in Computer Network Technology. *International Journal of Advanced Research in Science Communication and Technology.* 2024. Vol. 4. P. 588–592. DOI: 10.48175/IJARSCT-22770

19. 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. № 860. P. 385–401.

20. Srividhya G., Nagarajan R. A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network. *Computing.* 2024. 106, p. 2613–2642 DOI: 10.1007/s00607-024-01295-w

21. Temene N., Naoum A., Sergiou C., Georgiou C., Vassiliou V. A fault tolerant node placement algorithm for WSNs and IoT networks. *Computer Networks.* 2024. Vol 254. № 110835. DOI: 10.1016/j.comnet.2024.110835

22. Zhang Y., Wang L., Liu D., Su Y., Zhu Y., Zhang X. Design of Information Security Protection System for Cloud Business System. Proceedings of International Conference on Artificial Intelligence and Communication Technologies. Singapore. 2023. Smart Innovation. *Systems and Technologies.* 2024. Vol 369. P. 105–122. DOI: 10.1007/978-981-99-6956-2_10

23. Yamauchi T., Akao Y., Yoshitani R., Nakamura Y., Hashimoto M. Additional kernel observer: privilege escalation attack prevention mechanism focusing on system call privilege changes. *Int. J. Inf. Secur*. 2021. № 20. P. 461–473. DOI: 10.1007/s10207-020-00514-7

24. Ying L., Jiuqi W., Ziyu F., Yufan F., Xiaodong L. DiSAuth: A DNS-based secure authorization framework for protecting data decoupled from applications. *Computer Networks*. 2024. Vol. 254. № 110774. DOI: 10.1016/j.comnet.2024.110774

25. Zhou Q., Jia X., Chen J., Huang Q., Du H. LightArmor: A Lightweight Trusted Operating System Isolation Approach for Mobile Systems. ICT Systems Security and Privacy Protection. SEC 2024. *IFIP Advances in Information and Communication Technology*. 2024. Vol. 710. P. 206–220. DOI: 10.1007/978-3-031-65175-5_15

26. Harmonizing Security and Performance in Microkernel File Servers |/ W.T. Li, Z.X. Wang, J.Y. Gu et al. *J. Comput. Sci. Technol.* 2025. № 40, p. 464–481. DOI: 10.1007/s11390-025-3762-3

27. Stetsyuk Y., Stetsyuk M., Kyrylo V., Paiuk V., Kvassay M. A model of a centralized security system, as an information technology for the synthesis of an OS architecture protected against the leakage of confidential information. 1st International Workshop on Advanced Applied Information Technologies (AdvAIT-2024) will be held in Khmelnytskyi, Ukraine and Zilina, Slovakia on December 5. 2024. Vol. 3899. P. 224–233.

28. Stetsyuk Y., Stetsyuk M., Savenko B. Kwiecien A., Kopania L. Method of random dynamic control transfer between network nodes for a partially centralized OS security system. The 6th International Workshop on Intelligent Information Technologies & Systems of Information Security. April 04. 2025. Khmelnytskyi. Ukraine. Vol. 3963. P. 264–283.

29. Kashtalian A., Lysenko S., Kysil T., Sachenko A., Savenko O., Savenko B. Method and Rules for Determining the Next Centralization Option in Multicomputer System Architecture. *International Journal of Computing.* 2025. № 24(1). P. 35–51. DOI: 10.47839/ijc.24.1.3875

30. Kashtalian A., Lysenko S., Sachenko A., Savenko B., Savenko O., Nicheporuk A. Evaluation criteria of centralization options in the architecture of multicomputer systems with traps and baits. *Radioelectronic and Computer Systems.* 2025. № 1. p. 264–297. doi: 10.32620/reks.2025.1.18

31. Ajish D. The significance of artificial intelligence in zero trust technologies: a comprehensive review. *Journal of Electrical Systems and Inf Technol.* 2024. 11. № 30. DOI: 10.1186/s43067-024-00155-z

Stetsyuk Yu.V., Savenko O.S. PARTIALLY CENTRALIZED NETWORK OS SECURITY SYSTEM WITH DYNAMIC CONTROL TRANSFER BETWEEN ITS NODES

The current state of the threat level from malicious software (MSS) to modern protected specialized OSs is analyzed, in relation to their resistance to leaks of confidential information. A model of interaction of protective mechanisms of a centralized security system of a network node OS is presented, which allows obtaining an OS security architecture devoid of the problem of leakage of confidential information during attacks on the system's RAM, which is achieved by including a marking mechanism in it, the feature of which is the control of channels for passing confidential information at the level of memory pages. The principles of building centralized and partially centralized OS security systems and the principles of organizing the operation of their security mechanisms are considered. To preserve the positive qualities of a centralized security system and at the same time avoid the problems inherent in any centralization, a method of dynamic transfer of control between computer network nodes is proposed as a way to increase its resistance to leakage of confidential information and eliminate the bottleneck of a centralized security system – a single control center, as well as to counteract reconnaissance by attackers of a computer network in order to disrupt attacks on its resources. Its feature is the synchronization and updating of the central database of privileges and the state of the network by replicating the local databases of privileges of network nodes in the process of transferring control between them. Another feature of the method is to ensure the constant availability of a backup control node in case of failure. Each node that has transferred control of the network to a successor determined in a random way goes

into reserve. A generalized algorithm is proposed that implements the steps of the method of dynamic transfer of control between computer network nodes during normal network operation and for the situation of system start or failure. A model of the states of a network node in the process of dynamic control transfer between nodes has been developed, which allows it to be studied in all possible situations. A comparative analysis of the effectiveness of centralized and partially centralized security systems has been performed.

Key words: *security system, centralized system, partially centralized system, operating system, state model.*

Дата надходження статті: 31.07.2025
Дата прийняття статті: 28.08.2025
Опубліковано: 27.10.2025