Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Система оцінювання кібербезпеки корпоративних мереж»

КвРКІ.301163.23.01.18 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1 _____ Ігор РАМСЬКИЙ
Підпис Ім'я, прізвище

Керівник канд. екон. наук, доцент _____ Світлана САЧЕНКО
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КПС, д.ф., доц.
Ольга ПАВЛОВА
1 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма Освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 01 " 09 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Ігорю РАМСЬКОМУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система оцінювання кібербезпеки корпоративних мереж

Керівник проекту (роботи) Саченко С.І., к.е.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

аналіз відомих рішень щодо створення, захисту та підтримки розподілених комп'ютерних систем; архітектурні рішення для створення системи оцінювання кібербезпеки корпоративних мереж; функції оцінювання кібербезпеки корпоративних мереж та комп'ютерних станцій в них; метод синтезу системи оцінювання кібербезпеки корпоративних мереж, результати експерименту та моделювання роботи системи оцінювання кібербезпеки корпоративних мереж.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_01_»_____09_____2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування засобів для вирішення поставленої задачі, експериментальна частина	01.04.2026	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист КРМ	29.04.2025	виконано
10	Захист КРМ на засіданні ЕК	До 25.05.2025	

Студент _____ Ігор РАМСЬКИЙ
Підпис Ініціали, прізвище

Керівник роботи _____ Світлана САЧЕНКО
Підпис Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Система оцінювання кібербезпеки корпоративних мереж»

Автор роботи: Рамський Ігор Андрійович

Керівник роботи: Саченко С.І.

Пояснювальна записка: 77 с., 11 рис., 11 табл., 81 джерело.

КОРПОРАТИВНІ МЕРЕЖІ, РОЗПОДІЛЕНІ СИСТЕМИ, КІБЕРБЕЗПЕКА.

Об’єктом дослідження є кібербезпека в корпоративних мережах.

Предметом дослідження є методи та засоби кіберзахисту в корпоративних мережах.

Метою кваліфікаційної роботи магістра є покращення точності та ефективності оцінювання кібербезпеки корпоративних мереж.

Для розв’язання поставлених задач використовувалися методи оптимізації, теорія комп’ютерних систем.

Наукова новизна отриманих результатів:

- розроблено новий метод оцінювання кібербезпеки в корпоративних мережах.

На основі проведених досліджень розроблена система оцінювання кібербезпеки корпоративних мереж.

Практична значимість отриманих результатів полягає у розроблені математичного апарату для оцінювання кібербезпеки в корпоративних мережах.

У вступі подано об’єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи.

У першому розділі проведено аналіз відомих рішень щодо створення, захисту та підтримки розподілених комп'ютерних систем.

У другому розділі розроблені архітектурні рішення для створення системи оцінювання кібербезпеки корпоративних мереж.

У третьому розділі розроблені функції оцінювання кібербезпеки корпоративних мереж та комп’ютерних станцій в них.

У четвертому розділі розроблений метод синтезу системи оцінювання

кібербезпеки корпоративних мереж, проведено експеримент та моделювання роботи системи.

У висновках представлено підсумкову оцінку виконання завдань, сформульованих у ході дослідження.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

HTTP	HyperText Transport Protocol
FTP	File Transfer Protocol
XML	eXtensible Markup Language
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
SMTP	Simple Mail Transfer Protocol
DNS	Domain Name System
LDAP	Lightweight Directory Access Protocol
UEBA	User Entity and Behavior Analysis
SIEM	Security Information and Event Management
CVSS	Common Vulnerability Scoring System
EPSS	Exploit Prediction Scoring System
CVE	Common Vulnerability Exploit
ELK	Elasticsearch, Logstash, Kibana
DMZ	DeMilitarized Zone

ВСТУП

У сучасних умовах цифрової трансформації та зростання кількості кіберзагроз захист інформаційних ресурсів підприємств набуває критично важливого значення. Корпоративні мережі, які забезпечують взаємодію між численними інформаційними системами, стають основною ціллю для зловмисників, що використовують як традиційні методи атак, так і новітні засоби обходу захисних механізмів. У зв’язку з цим, питання своєчасного виявлення уразливостей, аналізу стану безпеки та оцінювання ризиків у корпоративному середовищі є актуальними як для дослідників, так і для практиків у сфері кібербезпеки.

Сучасні підходи до оцінювання кібербезпеки базуються на застосуванні аналітичних методів, автоматизованих систем моніторингу, а також на використанні формалізованих моделей, які враховують технічні, поведінкові та контекстуальні аспекти загроз. Проте, з огляду на динамічність мережевих середовищ та складність сучасної IT-інфраструктури, виникає потреба у розробці гнучких і масштабованих рішень, які можуть забезпечити об’єктивне оцінювання рівня захищеності корпоративної мережі.

Метою даної роботи є створення системи оцінювання кібербезпеки корпоративних мереж, що враховує технічні параметри комп’ютерних станцій, стан мережевих з’єднань, рівень вразливостей і актуальні загрози.

Для досягнення мети необхідно розв'язати такі основні завдання:

- здійснити аналіз існуючих методів і засобів захисту корпоративних мереж і розподілених систем, а також підходів до оцінювання рівня їх кібербезпеки;

- розробити новий метод оцінювання кібербезпеки в корпоративних мережах;

- реалізувати систему оцінювання кібербезпеки в корпоративних мережах відповідно до розробленого методу;

- дослідити роботу системи.

Об’єктом дослідження є кібербезпека в корпоративних мережах.

Предметом дослідження є методи та засоби кіберзахисту в корпоративних мережах.

Наукова новизна отриманих результатів:

- розроблено новий метод оцінювання кібербезпеки в корпоративних мережах.

На основі проведених досліджень розроблена система оцінювання кібербезпеки корпоративних мереж.

Практична значимість отриманих результатів полягає у розроблені математичного апарату для оцінювання кібербезпеки в корпоративних мережах.

За темою кваліфікаційної роботи опубліковано одну статтю у фаховому науковому журналі [81].

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ ОЦІНЮВАННЯ КІБЕРБЕЗПЕКИ У РОЗПОДІЛЕНИХ СИСТЕМАХ

1.1 Поняття про оцінювання кібербезпеки в корпоративних мережах

Оцінювання кібербезпеки в корпоративних мережах вимагає складних методів виявлення та реагування на різні загрози. Сучасні корпоративні мережі функціонують як розподілені системи з частковою централізацією, де прийняття рішень щодо виявлення шкідливого програмного забезпечення структуроване як децентралізована підсистема. Використання характерних індикаторів та аналітичних моделей дозволяє системі оцінювати складові стани та визначати відповідні реакції. Серед існуючих підходів є такий, що об'єднує кілька методів виявлення шкідливого програмного забезпечення, розглядаючи компоненти системи як інтегральні датчики [44][45].

Забезпечення стійкості до кібератак, зокрема ботнетів, є критично важливим аспектом оцінювання кібербезпеки. Переглянута література дає приклад самоадаптивної системи для переналаштування корпоративних мереж на основі сценаріїв безпеки, отриманих в результаті кластерного аналізу особливостей мережевого трафіку. Використовуючи напівконтрольований підхід кластеризації нечітких c-середніх (semi-supervised fuzzy c-means clustering), система виявляє кіберзагрози та вибирає стратегії безпеки для пом'якшення атак ботнетів, підвищуючи стійкість мережі [46]. Інша модель трирівневої системи виявлення ботнетів надає можливість ідентифікувати як відомі, так і невідомі ботнети, поєднуючи класифікацію Байєса на рівні хоста з розширеннями на рівні мережі. Такий підхід дозволяє ефективно обмінюватися інформацією в розподіленій системі і продемонстрував багатообіцяючі результати в точності виявлення ботнетів [47].

Розподілені атаки типу «відмова в обслуговуванні» (DDoS) є ще однією серйозною проблемою кібербезпеки, особливо в програмно-визначених мережах (SDN). Для виявлення та пом'якшення цих атак було розроблено фреймворк на основі машинного навчання, який використовує векторний класифікатор підтримки та класифікатор підвищення градієнта (SVC-GBC). З точністю 99,4% цей гібридний підхід значно підвищує безпеку SDN за рахунок уточнення деталізації виявлення та

посилення захисних механізмів [48]. Окрім виявлення вторгнень, виявлення аномалій у розподілених системах залишається проблемою через складні залежності між системними журналами. Мережа тимчасової логічної уваги (TLAN) на основі глибокого навчання була представлена для моделювання як шаблонів часових рядів, так і логічних залежностей, покращуючи продуктивність виявлення аномалій при одночасному зменшенні помилкових сигналів [49].

Надійність оцінювання кібербезпеки в розподілених системах додатково посилюється за рахунок механізмів виявлення збоїв. Ці механізми відстежують активність вузлів для виявлення несправностей і підвищення відмовостійкості системи. Систематичний аналіз детекторів відмов у розподілених середовищах підкреслює їх роль у забезпеченні надійності послуг шляхом вирішення проблем узгодження та відмов [50]. Виявлення аномалій на основі журналів (LAD) також відіграє важливу роль в оцінці кібербезпеки, використовуючи системні журнали для виявлення потенційних загроз і аномалій обслуговування. Загальна структура LAD для розподілених систем включає групування журналів і інтелектуальний аналіз за ознаками для підвищення ефективності виявлення, демонструючи його застосовність у реальних розподілених середовищах [51].

Крім того, питання конфіденційності в розподілених обчисленнях вимагають надійних систем безпеки. Дослідження конфіденційності в розподілених системах акцентує увагу на ризиках, пов'язаних з оцінюванням даних та відстеженням інформації, підкреслюючи актуальність моделей безпеки з нульовою довірою для безпечної реалізації систем в хмарних архітектурах [52]. Оскільки складність розподілених систем продовжує зростати, ефективні механізми аудиту систем, що об'єднують передову аналітику та штучний інтелект, стають важливими для моніторингу вразливостей та підвищення рівня безпеки [53]. Ці досягнення в сукупності сприяють створенню всеосяжної системи оцінювання кібербезпеки, що забезпечує стійкість корпоративних мереж до загроз, що розвиваються.

Оцінювання кібербезпеки в корпоративних мережах повинна вирішувати проблеми, пов'язані з надійністю, виявленням аномалій і дотриманням політики безпеки. Модель безпеки з нульовою довірою підкреслює необхідність перевірки

локальних серверів у корпоративних інтрамережах, проте існуючі методи сертифікації залишаються недоступними для невеликих організацій через вартість і складність. Ця прогалина призводить до залежності від самопідписаних сертифікатів, зростає вразливість до видавання себе за іншу особу та несанкціонованого доступу, що в кінцевому підсумку порушує принципи нульової довіри [54]. Для покращення виявлення загроз безпеці у великомасштабних розподілених системах був запропонований федеративний підхід, заснований на навчанні, що інтегрує мультимодальні великі мовні моделі. Ця система обробляє різноманітні джерела даних, досягаючи точності 96,4% при збереженні конфіденційності даних та обчислювальної ефективності, демонструючи значні покращення порівняно з традиційними методами виявлення [55].

Аномалії в розподілених системах створюють значні ризики через затримки в часі та погіршення якості даних. Було впроваджено систему оцінювання якості даних у реальному часі на основі глибокого навчання, яка використовує адаптивні нейронні мережі та паралельну обробку для забезпечення масштабованого виявлення аномалій з низькою затримкою. Оцінювання на великомасштабних наборах даних підтверджують ефективність системи в підтримці високої точності виявлення при обробці понад 1,2 мільйона подій в секунду [56]. У середовищах хмарних обчислень оптимізація розподілу ресурсів має вирішальне значення для підтримки ефективності. Підходи, засновані на машинному навчанні, що об'єднують глибоке навчання та генетичні алгоритми, були розроблені для покращення планування ресурсів, вирішуючи такі проблеми, як дисбаланс навантаження та низький рівень використання [57].

Подальший прогрес у розподілених обчисленнях зосереджений на підзвітності, виборі лідерів та генерації безпечної випадковості. Структура для підзвітних і реконфігурованих розподілених систем забезпечує безперебійну адаптацію у відповідь на збої, використовуючи абстракцію гратчастих угод. Крім того, інноваційні криптографічні протоколи покращують вибори лідерів у частково синхронних блокчейнах, покращуючи механізми консенсусу та стійкість системи [58]. Оскільки розподілені системи все частіше покладаються на моніторинг на основі

журналів для оцінювання безпеки, надійність моделей глибокого навчання проти зловмисних атак викликає все більше занепокоєння. Новий метод атак, LAM, маніпулює потоковими журналами, щоб уникнути виявлення аномалій, що підкреслює необхідність посилених заходів безпеки проти змагальних маніпуляцій [59].

Політики безпеки в розподілених системах також мають бути гнучкими та перевірятися в різних реалізаціях. Незалежна від мови система перевірки політик забезпечує відповідність політикам безпеки, аналізуючи поведінку введення-виведення замість того, щоб покладатися на обмеження мови програмування. Оцінювання демонструють його застосовність у реальних протоколах, що посилює потребу в адаптивних політиках безпеки [60]. Технологія блокчейн також сприяє кібербезпеці, підвищуючи прозорість і безпеку даних у розподілених системах управління. Однак такі проблеми, як масштабованість і сумісність, повинні бути вирішені, щоб в повній мірі використовувати потенціал блокчейна для захисту конфіденційних даних [61]. Нарешті, прогрес у глибокому навчанні для виявлення аномалій у журналах розподілених систем вводить моделі, які інтегрують глобальні просторово-часові особливості, значно підвищуючи точність виявлення загроз безпеці в складних середовищах [62]. Ці зміни в сукупності сприяють надійності та ефективності оцінювання кібербезпеки в корпоративних мережах.

Оцінювання кібербезпеки в корпоративних мережах повинна постійно адаптуватися до мінливих загроз і технологічного прогресу. Розподілені системи та обчислювальні підходи, включаючи технологію блокчейн та розподілені реєстри, пропонують значний потенціал для покращення запобігання фінансовим злочинам та кібербезпеки шляхом підвищення прозорості та зниження ризиків шахрайства. Однак такі проблеми, як відповідність нормативним вимогам, сумісність та інтеграція з існуючими інфраструктурами, повинні бути вирішені, щоб максимізувати ці переваги [63]. Проактивний підхід до безпеки необхідний у розподілених середовищах, а інтеграція методологій DevOps підвищує безпеку за рахунок вбудовування виявлення загроз у життєвий цикл розробки, автоматизації моніторингу та використання поведінкової аналітики для виявлення аномалій у режимі реального часу. Ця стратегія

сприяє формуванню культури спільної відповідальності за безпеку та дотримання правових стандартів [64].

Різноманіття систем є ще одним ключовим фактором підвищення надійності та безпеки розподілених мереж зв'язку. Аналітичні моделі, засновані на аналізі напруженості-сили, кількісно оцінюють ці поліпшення, надаючи цінну інформацію про стійкість системи [65]. У контексті інтелектуальних розподілених систем (SDS) забезпечення безпеки та сумісності даних має вирішальне значення для безперебійного обміну інформацією між такими галузями, як охорона здоров'я, комунальні послуги та ланцюги поставок. Встановлення глобальних стандартів безпеки може забезпечити основу для аутентифікації, співпраці та захисту від кіберзагроз у середовищах SDS [66]. Зростаюча інтеграція IoT з хмарними обчисленнями вносить нові вразливості, що вимагає комплексної структури безпеки, яка підвищує стійкість до кіберзагроз, зберігаючи при цьому масштабованість і адаптивність в розподілених середовищах [67].

Конфіденційність даних залишається серйозною проблемою, особливо в таких сферах, як освіта та охорона здоров'я. Розподілені обчислення пропонують покращення безпеки та часу відгуку, проте централізовані платформи часто перевершують розподілені системи за допомогою методів збереження конфіденційності, таких як k-анонімність, t-близькість та β-ймовірність. Порівняльний аналіз цих підходів виявляє компроміси в часі виконання, вимогах до пам'яті та рівнях придушення [68]. У сфері охорони здоров'я туманні обчислення є перспективним рішенням для моніторингу пацієнтів у режимі реального часу, але проблеми безпеки та конфіденційності повинні вирішуватися за допомогою шифрування, контролю доступу та методів аналізу даних, що зберігають конфіденційність [69]. Оцінка ризиків у розподілених інформаційних системах вимагає динамічного, багаторівневого підходу, який інтегрує кількісні, якісні та гібридні методології, використовуючи метрики безпеки для точних і надійних оцінок кібербезпеки [70].

Загрози кібербезпеці в інтелектуальних мережах підкреслюють важливість передових механізмів виявлення загроз. Традиційні методи навчання під наглядом

для виявлення кібератак вимагають різноманітних навчальних наборів даних, які не завжди можуть бути доступними. Підходи до інтелектуального аналізу даних без учителя, особливо для виявлення атак з використанням помилкових даних (FDIA), пропонують більш ефективну альтернативу, покладаючись виключно на звичайні дані про події для навчання моделей виявлення. Порівняльні дослідження демонструють, що неконтрольовані алгоритми перевершують методи контрольованого та глибокого навчання у виявленні невідомих шаблонів атак, підвищенні кібербезпеки в інфраструктурах інтелектуальних мереж [71]. Ці досягнення в сукупності сприяють зміцненню систем оцінювання кібербезпеки в корпоративних мережах, забезпечуючи стійкість до складних кіберзагроз.

Оцінювання кібербезпеки в корпоративних мережах повинна включати передові криптографічні методи для зниження ризиків витоку даних у розподілених середовищах. Хмарна криптографія відіграє вирішальну роль у захисті зберігання та передачі даних завдяки використанню механізмів шифрування, систем виявлення вторгнень і брандмауерів. Ці технології посилюють захист даних у хмарних розподілених системах, запобігаючи несанкціонованому доступу та проникненню шкідливого програмного забезпечення [72]. З розширенням хмарних і периферійних обчислень криміналістичні інструменти на основі штучного інтелекту стали ефективними рішеннями для виявлення та пом'якшення наслідків кіберінцидентів у режимі реального часу. Методи машинного навчання та глибокого навчання покращують криміналістичний аналіз, покращуючи масштабованість, точність та час реакції при виявленні кіберзагроз у розподілених системах [73].

Геополітичні міркування також впливають на стратегії кібербезпеки, оскільки кіберпростір стає спірною сферою за участю державних і недержавних суб'єктів. Рамки національної безпеки все частіше наголошують на партнерстві між державними та приватними структурами для протидії кіберзагрозам, що розвиваються, наголошуючи на необхідності адаптивних політик безпеки та стандартних рамок [74]. Контроль доступу залишається основним механізмом кібербезпеки, який гарантує, що неавторизовані суб'єкти не можуть перевищити свої дозволи. Останні досягнення в методах контролю доступу, особливо в хмарних

обчисленнях, блокчейні, IoT і програмно-визначених мережах, забезпечують підвищену безпеку при узгодженні зі стратегіями прийняття бізнесу і вимогами дотримання законодавства [75].

Інтеграція безпеки в пайплайни DevOps ще більше посилює захист розподілених систем за рахунок вбудовування найкращих практик кібербезпеки в життєвий цикл розробки. Автоматизоване тестування безпеки, безперервний моніторинг та механізми реагування на інциденти покращують пом'якшення вразливостей, зберігаючи при цьому гнучкість розгортання програмного забезпечення [76]. Поява розподілених хмарних систем створила нові загрози безпеці, включаючи атаки нульового дня та внутрішні загрози. Моделі глибокого навчання, такі як згорткові нейронні мережі (CNNs) і рекурентні нейронні мережі (RNNs), пропонують перспективні рішення для виявлення аномалій і запобігання вторгнень, хоча проблеми, пов'язані з інтерпретацією та затримкою, залишаються [77].

У міру розвитку кіберзагроз традиційні засоби захисту, такі як брандмауери та захист паролями, стають недостатніми, що вимагає передових методів, таких як розподілене статистичне висновування, виявлення аномалій та змагальне машинне навчання [78]. Інтеграція блокчейну в кіберфізичні системи підвищує стійкість за рахунок децентралізованої перевірки та безперервного моніторингу стану, проте залишаються проблеми в управлінні різнорідними промисловими інфраструктурами та оптимізації механізмів консенсусу для високої масштабованості [79]. Бездротові мережі зв'язку особливо вразливі до кібератак, оскільки зловмисники використовують складні методи для компрометації конфіденційних даних. Нові досягнення в галузі кібербезпеки, включаючи квантову криптографію та вдосконалені схеми управління ключами, надають перспективні рішення для зміцнення рамок безпеки від сучасних загроз [80]. Ці розробки в сукупності сприяють більш комплексній оцінці кібербезпеки, забезпечуючи надійний захист корпоративних мереж від дедалі складнішого ландшафту загроз.

Таким чином, аналіз існуючих досліджень у сфері оцінювання кібербезпеки корпоративних мереж дозволив виокремити ключові напрями розвитку відповідних систем, а також окреслити типові загрози, на які вони орієнтовані. Розглянуті підходи

продемонстрували, що ефективне виявлення уразливостей та аномалій потребує не лише технічної гнучкості, а й здатності адаптуватися до змінної структури мережі та поведінкових характеристик її елементів.

1.2 Методи створення розподілених систем для оцінювання кібербезпеки

У цьому параграфі розглянемо відомі дані про розподілені системи у контексті оцінювання кібербезпеки корпоративних систем, звернемо увагу на такі принципи і поняття, як масштабованість, відмовостійкість, надійність, узгодженість, моніторинг. Масштабованість і відмовостійкість, є критично важливими для автоматизованого оцінювання кібербезпеки корпоративної мережі, оскільки система повинна ефективно розширювати свої можливості для аналізу великих обсягів трафіку та залишатися працездатною навіть у разі відмови окремих вузлів. Постійний моніторинг дозволяє відстежувати стан мережі та виявляти потенційні загрози в режимі реального часу, забезпечуючи швидке реагування на інциденти. Завдяки високій надійності розподілена система забезпечує безперервну роботу без втрати даних, що є ключовим фактором для точного оцінювання кібербезпеки.

Масштабованість є фундаментальною характеристикою розподілених систем, що дозволяє їм динамічно регулювати продуктивність обчислень за рахунок модифікації наявних ресурсів і методів планування [27][3]. На відміну від монолітних архітектур, розподілені системи досягають масштабованості за рахунок горизонтального розширення, де додаткові вузли підвищують обчислювальну потужність, не перевантажуючи окремі компоненти [6]. Ця здатність особливо важлива для обробки великомасштабних даних, як це видно в розподілених файлових системах (DFS), які дозволяють ефективно зберігати файли на декількох взаємопов'язаних вузлах, зберігаючи при цьому безперебійний доступ і управління [5]. Сучасні впровадження DFS, включаючи Google File System (GFS) та Hadoop Distributed File System (HDFS), наголошують на масштабованості шляхом розподілу даних між кількома вузлами зберігання, забезпечуючи високу доступність та відмовостійкість при різних робочих навантаженнях [28].

Одним з ключових факторів масштабованих розподілених систем є розподілена обробка даних, яка дозволяє виконувати завдання одночасно на декількох вузлах, скорочуючи час обробки і підвищуючи загальну ефективність системи [7][21]. Такі технології, як Apache Spark і Apache Storm, полегшують як пакетну обробку даних, так і обробку даних в режимі реального часу, оптимізуючи розподіл робочого навантаження і покращуючи відмовостійкість. Ці фреймворки демонструють, як розподілена обробка даних дозволяє системам адаптуватися до коливань робочих навантажень без шкоди для продуктивності, що робить їх невід'ємною частиною сучасних обчислювальних інфраструктур.

Для забезпечення масштабованості розподілені системи використовують мікросервіси і безсерверні обчислення, які роз'єднують компоненти додатків і динамічно розподіляють ресурси в міру необхідності [6]. Ці архітектури пом'якшують вузькі місця в масштабованості, дозволяючи незалежному масштабуванню сервісів, дозволяючи розподіленим програмам ефективно реагувати на варіації попиту. Крім того, алгоритми, керовані штучним інтелектом, все частіше інтегруються в розподілені системи для оптимізації розподілу ресурсів і виявлення несправностей, що ще більше підвищує масштабованість і відмовостійкість [12][38]. Моделі машинного навчання, наприклад, можуть передбачати збої системи до того, як вони виникнуть, забезпечуючи проактивне управління несправностями та мінімізуючи час простою [29][39].

Масштабованість у розподілених системах також залежить від механізмів відмовостійкості, оскільки збої системи не повинні порушувати загальну продуктивність. Стратегії відмовостійкості, такі як реплікація, алгоритми консенсусу (наприклад, Paxos і Raft)[16][17] і прогнозне виявлення несправностей, допомагають підтримувати цілісність системи, незважаючи на збої компонентів [1][27]. Проактивний підхід використання машинного навчання для прогнозування несправностей ще більше підвищує надійність системи, дозволяючи вжити профілактичних заходів до того, як збої поширяться [29]. Крім того, прогрес в рекомендаційних системах полегшує адаптацію одномашинних проблем до розподілених архітектур, підвищуючи масштабованість і ефективність робочих

процесів обробки даних [40].

Відмовостійкість є критично важливим аспектом розподілених систем, який гарантує, що збої в окремих компонентах не порушать загальну роботу системи. З огляду на децентралізований характер розподілених середовищ, підтримка надійності вимагає балансування між узгодженістю, доступністю та толерантністю до розділів, що описується теоремою CAP [13].

Однією з фундаментальних проблем відмовостійкості є збереження узгодженості при забезпеченні доступності в багаторівневих розподілених обчислювальних системах, де компоненти працюють на пристроях IoT, платформах периферійних обчислень і хмарних середовищах [14]. Варіації затримки мережі вносять компроміси, які змушують системи жертвувати стабільністю або доступністю в умовах високої затримки. Для кількісної оцінки цих компромісів були розроблені математичні моделі, які допомагають розробникам систем приймати обґрунтовані рішення щодо обробки відмов та оптимізації продуктивності.

Щоб пом'якшити збої, викликані непередбачуваним часом виконання завдань, в розподілених обчисленнях зазвичай використовуються методи резервування, такі як реплікація, кодування і розбиття завдань [15]. Ці підходи підвищують надійність, дозволяючи виконувати завдання навіть у разі невдачі деяких завдань або затримок. Однак резервування створює компроміс між паралелізмом та ефективністю ресурсів, вимагаючи ретельного налаштування, щоб мінімізувати час виконання роботи та зберегти стійкість системи.

Оновлення системи є ще однією серйозною проблемою відмовостійкості, оскільки збої під час оновлення можуть призвести до тривалих перебоїв у обслуговуванні. Аналіз реальних збоїв оновлення в широко використовуваних розподілених системах виявив загальні закономірності збоїв і запропонував фреймворки проактивного тестування, такі як DUPTester і DUPChecker, які допомагають виявляти і запобігати збоям, викликаним оновленням, перед розгортанням [18]. Ці інструменти виявляють несумісність між версіями програмного забезпечення, зменшуючи ризик збоїв у розподілених середовищах, що постійно розвиваються.

Навіть в екстремальних умовах, таких як повністю дефектні асинхронні мережі, де всі лінії можуть зазнавати необмежених змін повідомлень, були розроблені механізми відмовостійкості для забезпечення надійного зв'язку. Методи, що використовують 2-краєві пов'язані структури графів і циклічну маршрутизацію повідомлень, дозволяють розподіленим системам функціонувати, незважаючи на тотальне пошкодження повідомлень, демонструючи надійність сучасних стратегій відмовостійкості [19]. Ці рішення підкреслюють необхідність проектування розподілених архітектур, здатних витримувати навіть найнесприятливіші умови.

Загрози безпеці, особливо атаки типу «відмова в обслуговуванні» (DoS), ще більше ускладнюють відмовостійкість у розподілених системах. Механізми розподілу ресурсів, такі як спалювання ресурсів (RB), захищають від DoS-атак, змушуючи зловмисників нести вищі витрати, ніж законні користувачі, стримуючи зловмисні збої [26]. Фреймворки на основі машинного навчання можуть динамічно налаштовувати рівні RB для оптимізації стійкості системи та мінімізації втрати ресурсів.

Моніторинг відіграє життєво важливу роль у забезпеченні надійності та продуктивності розподілених систем, забезпечуючи видимість їхньої діяльності в режимі реального часу. У зв'язку зі складністю асинхронних протоколів передачі повідомлень моніторинг повинен враховувати величезну кількість можливих шляхів виконання, що ускладнює характеристику набору досяжних станів системи. Одним з підходів до вирішення цієї проблеми є використання ймовірних інваріантів, які приблизно визначають досяжні стани системи на основі спостережень під час виконання. Вивчаючи та уточнюючи ці інваріанти, системи моніторингу можуть виявляти аномальні стани, які можуть вказувати на приховані помилки або неправильні конфігурації системи [4].

У хмарних розподілених програмах, керованих віртуалізацією, інструменти візуалізації програмного забезпечення допомагають розробникам керувати складністю, надаючи уявлення про використання системних ресурсів і взаємодію. Ці інструменти, особливо в середовищах Kubernetes, надають структурований спосіб аналізу поведінки додатків, хоча поточні рішення для візуалізації все ще охоплюють

лише частину доступних типів ресурсів. Покращення можливостей візуалізації матиме важливе значення для кращого моніторингу та усунення несправностей у великомасштабних розподілених середовищах [9]. Подібним чином, методи перевірки під час виконання пропонують систематичний метод моніторингу розподілених додатків, гарантуючи, що поведінка системи відповідає заздалегідь визначеним логічним обмеженням. Використовуючи лінійну часову логіку (LTL) і методи автоматизованої верифікації, системи можуть бути постійно оцінені на відповідність очікуваним властивостям, знижуючи ризик невиявлених збоїв [10].

Моніторинг на основі телеметрії набув популярності як метод виявлення системних аномалій та архітектурних проблем у розподілених програмах. Збираючи та аналізуючи дані про продуктивність системи, інженери можуть виявити закономірності, пов'язані зі збоями та неефективністю. Використання методів машинного навчання, таких як аналіз головних компонент (PCA), ще більше покращує виявлення аномалій, виявляючи відхилення в поведінці системи без вимоги заздалегідь визначених умов відмови. Ці досягнення сприяють автоматичній ідентифікації деградації системи, що дозволяє інженерам активно вирішувати виникаючі проблеми [31]. Крім того, фреймворки моніторингу помилок, які відстежують збої в окремих мікросервісах, дозволяють рано виявляти та повідомляти про проблеми, які в іншому випадку могли б поширюватися непомітно, покращуючи обслуговуваність системи та швидкість реагування [32].

У середовищах периферійних обчислень моніторинг є особливо складним через ненадійні умови мережі та потенційну можливість неправильної передачі даних. Щоб вирішити цю проблему, були запропоновані механізми виявлення несанкціонованого втручання на основі вставки функцій і теорії ігор, що дозволяють розподіленим системам перевіряти автентичність переданих даних, запобігаючи змові між скомпрометованими вузлами. Такий підхід гарантує, що периферійні пристрої надають достовірні дані, підвищуючи надійність розподілених фреймворків моніторингу [33]. Подібним чином моніторинг продуктивності мережі у великомасштабних розподілених системах спирається на аналітику великих даних для виявлення перевантажень, затримок і збоїв. Методи машинного навчання

допомагають класифікувати поведінку системи, дозволяючи розробляти стратегії оптимізації, які динамічно розподіляють ресурси та покращують доступність системи [35].

Інтегруючи ведення журналів, відстеження та збір показників, фреймворки для моніторингу надають глибоке розуміння поведінки системи, дозволяючи інженерам співвідносити події між розподіленими компонентами. Ці методи підвищують ефективність усунення несправностей, мінімізують час простою та покращують загальну відмовостійкість критично важливих додатків [37]. Оскільки складність розподілених систем продовжує зростати, поєднання вдосконаленої спостережуваності, виявлення аномалій та інтелектуальних рішень для моніторингу матиме важливе значення для підтримки високої доступності та стабільності роботи.

Ефективне управління ресурсами та оптимізація продуктивності мають вирішальне значення для розподілених систем, оскільки вони повинні збалансувати розподіл робочого навантаження, масштабованість та обчислювальну ефективність. Однією з ключових проблем в управлінні ресурсами є визначення оптимального розподілу робочих навантажень між декількома обчислювальними вузлами при збереженні гарантій продуктивності. Були досліджені різні стратегії розподілу ресурсів, включаючи парадигми розподіленого та централізованого управління, кожна з яких пропонує компроміси з точки зору адаптивності та ефективності. Гібридний підхід, який об'єднує обидві парадигми, забезпечує гнучкий механізм управління складними розподіленими середовищами, такими як інтелектуальні енергетичні мережі, за рахунок використання ієрархічного управління системою [8].

Хмарні обчислення значно змінили управління розподіленими ресурсами, забезпечивши динамічний розподіл робочого навантаження між географічно розподіленими центрами обробки даних. Передові алгоритми розподілу ресурсів, такі як розподілений розподіл ресурсів на основі справедливості (FDRA) та метод змінного напрямку множників (ADMM), були впроваджені для оптимізації розподілу ресурсів у різних областях, включаючи стільникові мережі, програмно-визначені мережі (SDN) та системи радіолокаційної візуалізації [22]. Еволюція хмарних обчислень і паралельних обчислень ще більше впровадила стратегії управління

ресурсами на основі штучного інтелекту, такі як нечітка мета-евристика, яка покращує операційну ефективність та економічну ефективність, одночасно вирішуючи такі проблеми, як складність впровадження та адаптивність системи [23].

Планування завдань у гетерогенних обчислювальних середовищах є ще одним фундаментальним аспектом управління ресурсами. Враховуючи складність задач планування, для оптимізації часу виконання при забезпеченні ефективного розподілу робочого навантаження зазвичай використовуються евристичні та метаевристичні алгоритми. Алгоритм Intelligent Harris Hawk Optimization (IHHO), наприклад, продемонстрував поліпшення в плануванні завдань шляхом подолання проблем ранньої збіжності та локальної оптимуми, що зробило його життєздатним рішенням для гетерогенних розподілених систем [25]. Ці підходи підкреслюють важливість адаптивних методів планування для підтримки продуктивності в динамічних розподілених архітектурах.

Розвиток мікросервісів також вплинув на управління ресурсами, сприяючи децентралізованому, незалежному розгортанню служб. У той час як мікросервіси підвищують масштабованість і гнучкість розробки, неправильна реалізація може призвести до розподілених монолітів, збільшуючи складність системи, а не підвищуючи ефективність. Правильне визначення меж послуг та оптимізація міжсервісної комунікації мають важливе значення для збереження переваг мікросервісів при мінімізації накладних витрат на ресурси [30].

Оцінка продуктивності в розподілених середовищах вимагає постійного моніторингу та аналізу часу виконання, використання ресурсів та ємності системи. Дослідження часу виконання в середовищах хмарних обчислень демонструють важливість мережевих моделей, таких як TCP-сокети та віддалений виклик методів (RMI), для оптимізації зв'язку між розподіленими компонентами. Ці методології підкреслюють взаємодію між розподіленими обчисленнями та хмарним виконанням, наголошуючи на важливості ефективної роботи в мережі для високопродуктивних розподілених систем [34].

Джерела [41], [42], [43] в деталях описують вплив бізнесу і його принципів на виклики у роботі з розподіленими системами.

Таким чином, огляд методів створення розподілених систем у контексті оцінювання кібербезпеки дозволив визначити ключові вимоги до їх архітектури, зокрема масштабованість, відмовостійкість, узгодженість і здатність до безперервного моніторингу. Особлива увага була приділена особливостям функціонування комп’ютерних мереж і процесів, що в них відбуваються, адже саме від коректної обробки розподілених даних і взаємодії між вузлами залежить своєчасне виявлення загроз.

1.3 Постановка задачі

Для досягнення поставленої мети передбачається виконання таких ключових завдань:

– здійснити аналіз існуючих методів і засобів захисту корпоративних мереж і розподілених систем, а також підходів до оцінювання рівня їх кібербезпеки;

– запропонувати метод оцінювання кібербезпеки корпоративних мереж;

– сформулювати цільову функцію, що характеризує рівень кібербезпеки корпоративної мережі;

– реалізувати прототип системи та провести дослідження ефективності запропонованого методу оцінювання.

1.4 Висновки до першого розділу

Проаналізовано відомі методи та засоби оцінювання кібербезпеки у розподілених системах, зосереджені на виявленні вразливостей, оцінці ризиків та використанні формалізованих метрик. Досліджено їх сильні та слабкі сторони, визначено актуальні підходи, що можуть бути адаптовані для розробки нової системи оцінювання.

2 АРХІТЕКТУРА СИСТЕМИ ОЦІНЮВАННЯ КІБЕРБЕЗПЕКИ У КОРПОРАТИВНИХ МЕРЕЖАХ

2.1 Архітектура системи

Система оцінювання кібербезпеки корпоративної мережі є стандартизованим технічним рішенням, призначеним для універсального застосування в різних корпоративних середовищах незалежно від їх галузевої специфіки, масштабів чи внутрішньої організаційної структури. Основною метою такої системи є забезпечення об’єктивного, системного та безперервного аналізу стану кібербезпеки шляхом збору, обробки та інтерпретації інформації, що стосується потенційних і фактичних загроз інформаційній інфраструктурі підприємства.

Універсальність системи досягається завдяки побудові її архітектури на основі типових ознак, спільних для більшості корпоративних мереж. До таких ознак належать наявність множини вузлів з мережевою взаємодією, централізоване або розподілене управління ресурсами, використання засобів автентифікації і контролю доступу, а також передбачувані правила взаємодії між компонентами мережі. Це дозволяє системі адаптуватися до різних технічних умов без потреби суттєвої модифікації її функціональних модулів.

Система реалізовуватиме модель оцінювання безпеки, що включає отримання даних з доступних джерел у межах корпоративної мережі, їх подальшу обробку для виявлення відхилень, потенційних уразливостей чи індикаторів компрометації, а також оцінювання ризику з точки зору впливу на цілісність, доступність та конфіденційність інформації. При цьому система не залежить від конкретних реалізацій програмного або апаратного забезпечення, а її механізми побудовані на узагальнених протоколах і стандартах взаємодії.

Оскільки жодна корпоративна мережа не є повністю статичною, система передбачає постійний моніторинг змін у мережевому середовищі. Вона повинна бути здатна виявляти появу нових вузлів, зміну конфігурацій, порушення політик безпеки, а також здійснювати адаптивне переналаштування на основі змінених вхідних умов. Цей підхід дозволяє зберігати актуальність оцінювання навіть за умови динамічного

розвитку мережевої інфраструктури.

У процесі побудови архітектури системи оцінювання кібербезпеки важливим етапом є впровадження уніфікованих механізмів і методик, які дозволяють формалізувати оцінку критичності виявлених вразливостей інформаційної інфраструктури. Одним із таких механізмів, що вже зарекомендував себе як ефективний інструмент у світовій практиці, є Common Vulnerability Scoring System. Інтеграція цієї методики до архітектури системи дозволить досягти узгодженості між результатами аналізу, які продукує система, та загальноприйнятими стандартами безпеки. CVSS забезпечує об’єктивне, кількісне оцінювання небезпеки кожної виявленої вразливості на основі ряду характеристик, що враховують як технічні параметри самої вразливості, так і контекст її експлуатації в межах корпоративної мережі.

Система, побудована з урахуванням використання CVSS, отримує можливість розглядати вразливості не лише з технічної точки зору, а й з позиції їхнього потенційного впливу на загальний рівень інформаційної безпеки підприємства. Зокрема, шкала CVSS включає оцінювання за такими параметрами, як вплив на конфіденційність, цілісність і доступність даних, легкість експлуатації, наявність методів захисту, що можуть зменшити ефективність атаки, а також вектор доступу до вразливості. Завдяки цьому система зможе формувати обґрунтовану аналітичну картину загроз і розставляти пріоритети реагування, враховуючи як інтенсивність потенційного впливу, так і ймовірність реалізації загрози в конкретному середовищі.

Крім того, використання CVSS у межах архітектури системи дозволяє досягти гнучкості та масштабованості в реалізації функцій з управління ризиками. Наприклад, обчислені бали CVSS можуть використовуватись як вхідні параметри для модулів кореляції подій, систем ухвалення рішень щодо автоматизованого реагування, або ж для побудови динамічних профілів ризику, які відображають поточний стан захищеності мережі. Завдяки уніфікації підходів до оцінювання, можливо не лише забезпечити співставність отриманих результатів у часі, але й інтегрувати систему з іншими платформами безпеки, які також використовують CVSS як стандартний інструмент оцінювання. Це підвищує рівень інтероперабельності архітектури,

розширює можливості її інтеграції у вже наявну інфраструктуру підприємства та сприяє формуванню більш повної та обґрунтованої системи прийняття рішень у сфері кіберзахисту.

Для забезпечення повноцінного функціонування системи оцінювання кібербезпеки, побудованої на основі методології Common Vulnerability Scoring System, необхідно передбачити наявність спеціалізованого програмного забезпечення, яке виконує ключові функції збору, обробки та інтерпретації даних про вразливості. Таке програмне забезпечення має бути інтегрованим компонентом архітектури системи й відповідати вимогам до масштабованості, автоматизованої взаємодії з зовнішніми джерелами інформації, високої точності розпізнавання вразливостей, а також здатності до обчислення числових показників за схемою CVSS. Зокрема, мова йде про програмні рішення, які здатні виконувати як активне, так і пасивне сканування об'єктів корпоративної мережі, формувати звіти про знайдені вразливості та здійснювати зіставлення кожного об'єкта з відповідною вразливістю з баз даних, що містять стандартизовану інформацію про відомі загрози. Це дає змогу отримувати базові, часові та контекстуальні метрики, які є фундаментом для розрахунку інтегрального показника ризику за шкалою CVSS.

На сучасному етапі існує низка широко використовуваних інструментів, які відповідають таким вимогам. Серед них варто виділити OpenVAS, Nessus, Qualys, Nexpose, а також комерційні та відкриті рішення, що підтримують обмін інформацією з базами даних на кшталт National Vulnerability Database, CERT Coordination Center, Exploit Database тощо. Ці сканери здатні автоматично виявляти широкий спектр вразливостей у програмному забезпеченні, операційних системах, мережевих сервісах і навіть апаратному забезпеченні, а також співвідносити знайдену інформацію з уніфікованими записами в CVE (Common Vulnerabilities and Exposures). На основі такого зіставлення система отримує унікальні ідентифікатори для кожної вразливості, а також доступ до відповідної CVSS-оцінки, яка вбудована в кожен запис. У випадку, коли значення CVSS не надається напряму або є застарілим, розроблювана система повинна містити механізм для обчислення цього значення

самостійно, використовуючи наявні метрики відповідно до формальної моделі CVSS, включаючи формули для базових, часових і контекстуальних балів.

Інтеграція з зазначеними інструментами повинна реалізовуватись через спеціальні інтерфейси прикладного програмування або механізми автоматичного імпорту звітів у стандартизованих форматах, таких як XML, JSON або CSV. Це дозволить побудувати гнучку архітектуру, яка підтримує регулярне оновлення інформації про стан системи без потреби у ручному втручанні. Автоматизація процесів збору та обробки вхідних даних відкриває можливості для створення безперервного циклу оцінювання, в якому кожна нова вразливість, що з’являється у мережі, негайно фіксується, класифікується та отримує відповідний рівень пріоритету. У подальшому ці дані можуть бути використані як вхідні параметри для модуля оцінювання ризиків, що дозволить враховувати не лише факт наявності вразливості, але й ступінь її критичності для конкретного середовища, враховуючи важливість активу, рівень доступу до нього, існування обхідних засобів захисту та ймовірність експлуатації.

У межах архітектури системи оцінювання кібербезпеки корпоративної мережі важливим елементом є модуль перевірки стану з'єднань між мережевими вузлами, який забезпечує виявлення потенційно вразливих або неправильно налаштованих каналів передавання даних. Такий модуль повинен виконувати низку функцій, пов’язаних із перевіркою відкритості портів, ідентифікацією використовуваних протоколів, виявленням фактів застосування або відсутності шифрування, а також з аналізом відповідності мережевої взаємодії встановленим політикам безпеки. Реалізація цього компонента є необхідною передумовою для формування цілісного уявлення про стан захищеності інформаційних потоків усередині корпоративної інфраструктури.

Найбільш базовим способом перевірки з'єднань є сканування портів, яке дозволяє визначити, які сервіси доступні на конкретному вузлі, а також які протоколи використовуються для взаємодії. Ця процедура дозволяє не лише ідентифікувати відкриті порти, але й зафіксувати відповідь сервера, зокрема версію програмного забезпечення, конфігураційні особливості та час відгуку. Така інформація є цінною з

точки зору виявлення служб, що можуть містити вразливості, зокрема ті, які не використовують шифрування або застосовують застарілі криптографічні алгоритми. У межах архітектури системи доцільним є використання інструментів, таких як Nmap або Masscan, які дозволяють здійснювати як швидке широкомасштабне сканування мережі, так і поглиблений аналіз окремих вузлів.

Крім визначення відкритості, система повинна забезпечувати можливість аналізу рівня захищеності з'єднань, що реалізується через перевірку застосування протоколів шифрування. Особливу увагу необхідно приділити TLS/SSL-з'єднанням, які є стандартом для забезпечення захищеної передачі даних. У межах перевірки здійснюється аналіз сертифікатів, зокрема їхньої чинності, довіри до сертифікаційного центру, використаних алгоритмів шифрування, довжини ключів і версії протоколу. Виявлення слабких конфігурацій, наприклад підтримки TLS 1.0 або RC4, може свідчити про потенційний ризик перехоплення або дешифрування трафіку. Для таких перевірок можуть бути використані інструменти на кшталт OpenSSL, testssl.sh або SSLyze, які дозволяють виконувати поглиблений криптографічний аудит у межах корпоративної мережі.

Ще одним важливим аспектом є виявлення незашифрованих протоколів, таких як HTTP, FTP або Telnet, особливо якщо вони використовуються для передавання конфіденційної інформації. У межах системи необхідно реалізувати механізм фіксації подібних з'єднань з подальшою передачею інформації до аналітичного модуля, який здійснює оцінку рівня ризику. Наявність таких протоколів у мережевому трафіку може бути критичним індикатором порушення політик безпеки або відсутності належного контролю з боку адміністрації мережі.

Крім цього, доцільно реалізувати можливості пасивного моніторингу трафіку, що дозволяє здійснювати аналіз без активного втручання в мережу. Такий підхід передбачає застосування інструментів, здатних перехоплювати та класифікувати мережеві пакети, з подальшим виявленням з'єднань, що не відповідають очікуваним характеристикам безпеки. Це дозволяє виявляти аномальні або несанкціоновані канали зв'язку, що можуть бути використані для витоку даних або внутрішнього переміщення зловмисника в межах мережі.

У контексті архітектури системи оцінювання кібербезпеки важливим доповненням до механізмів перевірки з’єднань є реалізація функціональності з аналізу фільтрації мережевого трафіку. Цей компонент відіграє ключову роль у виявленні порушень політик доступу, несанкціонованого трафіку або неправильно налаштованих міжмережевих екранів та систем контролю доступу. Перевірка фільтрації трафіку дозволяє оцінити ефективність захисних механізмів, які мають забезпечувати ізоляцію сегментів мережі, обмеження доступу до критичних ресурсів та блокування небажаного трафіку на рівні як вхідних, так і вихідних потоків.

У рамках архітектури система повинна бути здатна моделювати типові сценарії мережевої взаємодії та фіксувати результат спроб встановлення з'єднання між окремими вузлами або з зовнішніми сервісами. Зокрема, перевірки можуть проводитися з урахуванням очікуваних політик безпеки, які визначають допустимі маршрути трафіку, дозволені протоколи та порти, а також регламентовані напрями зв’язку між підсистемами. Порушення цих політик, наприклад відкритість адміністративного інтерфейсу в загальнодоступному сегменті мережі або можливість з’єднання з неконтрольованими зовнішніми адресами, мають бути зафіксовані як потенційні індикатори загрози.

Для реалізації перевірки фільтрації можуть бути використані як активні, так і пасивні методи. Активні методи передбачають генерацію тестових з’єднань, зокрема з використанням інструментів на кшталт Hping або nping, які дозволяють точно налаштовувати параметри пакетів, імітувати трафік від імені різних служб і перевіряти, чи досягає він цільового вузла. Такі перевірки дозволяють визначити, чи правильно налаштовані правила на міжмережевих екранах, маршрутизаторах або проксі-серверах. У свою чергу, пасивні методи базуються на моніторингу трафіку в реальному часі з використанням систем виявлення вторгнень або аналізаторів трафіку, які фіксують усі спроби з’єднання, незалежно від їх успішності, і виявляють аномалії, наприклад спроби обійти фільтрацію за допомогою нестандартних портів або тунелювання.

Важливим аспектом є й можливість перевірки фільтрації вихідного трафіку, який часто є менш захищеним, але може слугувати каналом для витоку даних у разі

компрометації внутрішнього вузла. Система повинна бути здатна моделювати ситуації, в яких внутрішній компонент намагається встановити з’єднання з контрольованим сервером за межами мережі, та фіксувати, чи блокується така активність. Зокрема, це дозволяє виявити відсутність обмежень на доступ до зовнішніх DNS-серверів, відкриті SMTP-з’єднання чи інші критичні вектори для організації атак типу Command and Control.

Результати перевірки фільтрації трафіку повинні інтегруватися у загальний механізм оцінювання ризику, де вони розглядаються у контексті поточної конфігурації мережі, рівня критичності доступних ресурсів та наявності інших вразливостей. У разі виявлення порушень система може не лише фіксувати їх як інциденти, але й формувати рекомендації щодо зміни конфігурації або автоматично ініціювати відповідні дії з обмеження доступу.

У межах проєктованої архітектури системи оцінювання кібербезпеки корпоративної мережі передбачається включення модуля виявлення аномальної активності, що виконує функції виявлення відхилень від типових параметрів поведінки користувачів, пристроїв та служб. Такий модуль має забезпечувати автоматизований моніторинг подій у реальному часі та виявлення нетипових змін у динаміці функціонування мережі, які можуть свідчити про компрометацію, зловмисну активність, порушення політик безпеки або порушення цілісності системних процесів. Основою його роботи є концепція поведінкового аналізу, яка не передбачає наявності наперед визначених сигнатур загроз, а замість цього орієнтується на фіксацію атипових подій щодо сформованих профілів стандартної активності.

Для побудови зазначеного функціонального компонента архітектура повинна передбачати механізми збору та обробки великої кількості подій з різних джерел, зокрема системних журналів, мережевих пакетів, даних з міжмережевих екранів, засобів автентифікації, систем зберігання логів, телеметрії від кінцевих пристроїв та мережевого обладнання. Інтеграція цих джерел повинна забезпечуватись через спеціалізовані інтерфейси, які дозволяють агрегувати інформацію в єдиному середовищі для подальшого аналізу. Важливим є забезпечення сумісності із

загальновживаними протоколами обміну повідомленнями, такими як Syslog, NetFlow, sFlow або IPFIX, що дозволяє реалізувати централізований збір даних у гетерогенному корпоративному середовищі.

У межах архітектури може бути передбачено використання спеціалізованого програмного забезпечення, яке забезпечує автоматизований аналіз аномалій за допомогою методів машинного навчання, кореляційного аналізу або статистичних моделей. Серед таких рішень варто виокремити програмні платформи класу SIEM, зокрема Splunk, IBM QRadar, ArcSight, LogRhythm, а також системи з підтримкою повноцінного UEBA-модуля, такі як Exabeam або Varonis. Зазначене програмне забезпечення здатне будувати динамічні профілі поведінки користувачів та об'єктів, фіксувати відхилення від типових сценаріїв, ранжувати виявлені події за критичністю та інтегрувати результати аналізу з іншими модулями системи. Також можливе використання рішень з відкритим вихідним кодом, зокрема Wazuh, OSSEC або Zeek, які можуть бути адаптовані для потреб конкретного середовища з урахуванням його масштабів і технічних особливостей.

Архітектурно система повинна підтримувати механізми навчання моделей на основі зібраної історичної інформації, а також їх адаптації у випадку змін у структурі мережі, облікових політиках або конфігурації доступу до ресурсів. Це дозволяє враховувати закономірності, властиві конкретному середовищу, та знижує ймовірність виникнення хибнопозитивних спрацювань. Застосування профілювання також дає змогу проводити порівняльний аналіз між активністю різних об'єктів, наприклад користувачів з однаковими повноваженнями, що підвищує ефективність виявлення прихованої активності або внутрішніх загроз. Виявлені аномалії повинні передаватись до аналітичного модуля, що оцінює потенційні наслідки фіксованих подій з точки зору ризику для конфіденційності, цілісності та доступності інформації.

Функціонування модуля виявлення аномальної активності, як і всієї системи оцінювання кібербезпеки загалом, передбачається в умовах сегментованої корпоративної мережі, що є ключовим принципом побудови її архітектури. Мережева сегментація дозволяє розмежовувати інформаційні потоки, обмежувати маршрути між підсистемами, ізолювати критичні компоненти інфраструктури та реалізовувати

диференційований контроль доступу до окремих ресурсів. У такій структурі кожен сегмент розглядається як окрема функціональна область з власними правилами взаємодії, а також із визначеними політиками щодо допустимих маршрутів, типів з’єднань, протоколів і обсягів трафіку.

У межах сегментованої мережі модуль виявлення аномалій виконує функцію контролю за межами дозволеної взаємодії між сегментами, а також фіксує відхилення у межах кожного з них. Завдяки наявності чітко визначених маршрутів і обмеженого числа сценаріїв доступу, система отримує змогу побудувати високоточні моделі нормальної поведінки, де навіть незначне відхилення від очікуваного профілю може розглядатися як потенційний індикатор загрози. Наприклад, поява міжсегментного трафіку, що не передбачений конфігурацією, або спроб доступу до сервісів, які мають бути недоступні з певного сегмента, означає потенційну атаку. Аналогічно, фіксація підозрілої активності всередині сегмента, де функціональне навантаження вузлів є стабільним і добре визначеним, дозволяє оперативно виявляти загрози з боку внутрішніх користувачів або скомпрометованих пристроїв.

Архітектура повинна враховувати, що в умовах сегментації можливе фізичне або логічне розмежування зон, що накладає вимоги на спосіб збору даних та синхронізацію аналітичних процесів між сегментами. У зв’язку з цим реалізація функціональності модуля має передбачати або розподілену структуру, де кожен сегмент обладнаний локальними компонентами моніторингу та обробки, або централізовану систему з можливістю безпечної агрегації даних у єдиному аналітичному середовищі. В обох випадках критично важливо забезпечити захищеність каналів передавання даних між сегментами, зокрема через використання шифрування, автентифікації на рівні сервісів, а також контроль за відповідністю обсягів трафіку очікуваним нормам.

Функціонування в умовах сегментованої мережі не лише підвищує точність виявлення аномалій, а й істотно знижує потенційний обсяг інциденту, ізолюючи його в межах одного сегмента та унеможливлюючи горизонтальне поширення загроз. Це створює умови для реалізації стратегії стримування та локалізації інцидентів, що у

сукупності з поведінковим аналізом дозволяє значно підвищити ефективність системи в цілому.

У межах архітектури системи також враховується наявність демілітаризованих зон, які використовуються для ізольованого розміщення сервісів із публічним доступом, зокрема веб-серверів, поштових шлюзів, DNS-серверів, проксі-серверів та інших компонентів, що взаємодіють із зовнішніми користувачами або організаціями. Демілітаризована зона (DMZ) створює додатковий рівень розмежування між зовнішнім середовищем і внутрішніми сегментами корпоративної мережі, дозволяючи обмежити можливість прямого доступу до критичних ресурсів ззовні та локалізувати вплив потенційних атак на зовнішні сервіси. Взаємодія між внутрішньою мережею, DMZ і зовнішнім середовищем регламентується правилами фільтрації на міжмережевих екранах і може супроводжуватись глибокою перевіркою вхідного та вихідного трафіку, включаючи застосування IDS/IPS-систем, шифрування даних і контроль автентичності.

У межах реалізації модуля виявлення аномальної активності передбачається окрема логіка аналізу подій, що виникають у демілітаризованій зоні. З урахуванням специфіки її функціонування, а саме підвищеного рівня зовнішньої взаємодії, частого доступу до загальнодоступних сервісів та великої кількості однотипних запитів, профілі нормальної поведінки для таких сегментів будуються за відмінними критеріями порівняно з внутрішніми сегментами. Система має бути чутливою не лише підозрілої активності з боку зовнішніх джерел, зокрема спроб перебору портів, сканування або ініціації нестандартних з’єднань, але й моніторинг потенційного некоректного використання зовнішніх сервісів з боку внутрішніх об'єктів, що може свідчити про намагання обійти політики доступу або про ознаки внутрішньої компрометації. Зібрані дані передаються до загального аналітичного модуля, що дозволяє забезпечити повноту оцінювання ризиків з урахуванням ролі DMZ як проміжного елементу між відкритим середовищем і захищеною внутрішньою мережею.

Таким чином, архітектура системи оцінювання кібербезпеки корпоративних мереж має забезпечувати гнучкість, масштабованість і здатність до інтеграції з

існуючою інфраструктурою підприємства. У центрі цієї архітектури лежить необхідність точного збору й аналізу інформації про вразливості, з урахуванням як технічного контексту, так і загроз, що постійно змінюються. Застосування уніфікованих підходів, зокрема методології CVSS, дозволяє формалізувати оцінювання ризиків і забезпечити зіставність результатів, що є критично важливим для прийняття обґрунтованих рішень у сфері кіберзахисту.

2.2 Параметри комп’ютерних станцій для оцінювання кібербезпеки, процеси в комп'ютерних мережах

Для забезпечення ефективної оцінки рівня кібербезпеки корпоративних мереж необхідним є належне технічне оснащення комп’ютерних станцій, що виконують функції аналізу, моделювання загроз та моніторингу мережевої активності. Основними параметрами таких станцій є центральний процесор (CPU), оперативна пам’ять (RAM) та тип і обсяг накопичувачів даних. Сучасні засоби кіберзахисту, зокрема системи аналізу трафіку, сканери вразливостей, емулятори атак або інструменти для тестування на проникнення (penetration testing), вимагають високої обчислювальної потужності, що зумовлює необхідність використання багатоядерних процесорів із частотою не менше ніж 2.5 ГГц та підтримкою багатопотокової обробки. Зокрема, рекомендованим є використання процесорів серій Intel Core i5/i7/i9 або AMD Ryzen 5/7/9 з архітектурною підтримкою віртуалізації (Intel VT-x, AMD-V), що є критично важливим при використанні гіпервізорів та емуляції мережевих середовищ. Обсяг оперативної пам’яті повинен становити не менше 8 ГБ для базового функціонування інструментів аналізу, однак для комплексної оцінки безпеки у віртуалізованому середовищі доцільно застосовувати конфігурації з 16 ГБ і більше. Це дозволяє одночасно розгортати декілька віртуальних машин, кожна з яких може імітувати окрему частину корпоративної мережі або виконувати специфічні рольові функції в межах сценаріїв Red Team/Blue Team. Накопичувачі повинні забезпечувати як достатній обсяг для збереження великих обсягів логів, дампів трафіку та баз даних вразливостей, так і високу швидкість читання/запису. Твердотільні накопичувачі

(SSD) з інтерфейсом NVMe значно перевищують за продуктивністю традиційні HDD, що дозволяє суттєво скоротити час обробки даних та підвищити загальну продуктивність системи. Мінімально рекомендований обсяг SSD – 512 ГБ, проте для повноцінної роботи з великими обсягами мережевих логів, реплік атак та баз загроз бажаним є використання накопичувачів від 1 ТБ і більше. У сукупності зазначені апаратні характеристики забезпечують стабільну роботу системи оцінювання кібербезпеки, уможливлюють застосування широкого спектру сучасного програмного забезпечення та сприяють підвищенню достовірності, швидкодії й ефективності діагностики кіберзагроз у корпоративному середовищі.

Вибір операційної системи для комп'ютерних станцій, призначених для оцінювання кібербезпеки, є критичним чинником, що безпосередньо впливає на ефективність, функціональність та сумісність інструментів, які використовуються в процесі аналізу інформаційної безпеки. Залежно від поставлених завдань доцільним є застосування як систем на базі Linux, так і Windows, оскільки кожна з платформ має свої переваги та обмеження у контексті кіберзахисту. Операційні системи сімейства Linux традиційно вважаються базовими для фахівців з кібербезпеки завдяки відкритому коду, гнучкості конфігурації, широкій підтримці мережевих утиліт, а також можливості глибокої модифікації ядра й компонентів системи. Серед найбільш поширених дистрибутивів, орієнтованих на інформаційну безпеку, слід виділити Kali Linux та Parrot OS, які містять попередньо встановлені інструменти для сканування портів, перехоплення трафіку, аналізу вразливостей, експлуатації відомих загроз та тестування на проникнення. Kali Linux є офіційним проектом Offensive Security і широко застосовується у професійних пентест-лабораторіях, завдяки своїй сумісності з Metasploit Framework, Burp Suite, Wireshark, Nmap, Aircrack-ng та іншими критично важливими інструментами. Parrot OS, у свою чергу, поєднує засоби для аналізу безпеки з інструментами для розробки, цифрової криміналістики та конфіденційного спілкування, що розширює сферу його застосування в рамках захищених мережевих середовищ.

Особливу увагу при формуванні апаратного забезпечення комп'ютерних станцій слід приділяти мережевим інтерфейсам, оскільки саме вони забезпечують

фізичне та логічне підключення до досліджуваного сегменту мережі та здійснення збору даних для подальшого аналізу. Мережеві адаптери повинні підтримувати функціональність роботи у режимі перехоплення трафіку, зокрема так званий "promiscuous mode", який дозволяє приймати пакети, адресовані не лише локальній мережевій адресі пристрою, а й увесь трафік, що циркулює в межах доступного домену колізій або сегменту мережі. Це критично важливо для ефективного пасивного моніторингу, аналізу мережевих протоколів, виявлення аномалій та підозрілої активності. Більшість сучасних мережевих карт підтримує даний режим, однак у практиці кібербезпеки рекомендується застосування адаптерів професійного рівня, які забезпечують апаратне прискорення обробки пакетів, низький рівень втрат при високонавантажених мережах, а також сумісність з широким спектром драйверів і операційних систем. Особливої актуальності набуває використання мережевих карт з підтримкою технології hardware timestamping, яка дозволяє точно синхронізувати часові мітки пакетів, що є необхідним при кореляційному аналізі подій у розподілених середовищах. Крім того, важливою є підтримка адаптером функцій відключення апаратного offloading'у (наприклад, TCP segmentation offload або checksum offload), які можуть перешкоджати коректному аналізу сирих мережевих даних засобами типу Wireshark, Zeek або Suricata. В окремих випадках, зокрема при виконанні активних тестів або імітації атак, може виникнути потреба у застосуванні адаптерів з підтримкою моніторингового режиму (monitor mode) для бездротових мереж, що дає змогу перехоплювати трафік на рівні радіоінтерфейсу. Всі зазначені особливості адаптерів повинні враховуватись у процесі формування технічних вимог до комп’ютерних станцій, які виконують завдання в межах системи оцінювання кібербезпеки, оскільки вони безпосередньо впливають на повноту та достовірність зібраної інформації, а отже – на якість виявлення вразливостей та реагування на потенційні інциденти.

Операційні системи на базі Windows, попри свою закриту архітектуру, також відіграють важливу роль у процесі оцінки безпеки корпоративних мереж, особливо у випадках, коли досліджуване середовище побудоване на технологіях Microsoft. Інструменти, як-от Windows Sysinternals, PowerShell з модулями безпеки, Active

Directory Explorer або специфічні сканери вразливостей, які інтегруються з корпоративними Windows-мережами, потребують безпосереднього запуску в середовищі Windows. Окрім того, тестування засобів соціальної інженерії, експлуатація специфічних для Windows API, емуляція поведінки кінцевого користувача або дослідження впливу шкідливого ПЗ також вимагають присутності повноцінного Windows-середовища. У сучасній практиці, оптимальним підходом є створення гібридних конфігурацій, де на базовій хост-системі (наприклад, Ubuntu або Windows 11) розгортається декілька віртуальних машин з різними ОС для комплексного моделювання мережевих взаємодій та багаторівневого аналізу.

Окрім центральних обчислювальних компонентів та засобів зберігання інформації, критично важливою складовою комп'ютерних станцій, призначених для оцінювання кібербезпеки, є здатність до ефективної віртуалізації. У сучасних умовах саме використання віртуалізованих середовищ дозволяє гнучко моделювати структури корпоративних мереж, створювати ізольовані зони для тестування атак, відтворювати сценарії інфікування або компрометації вузлів, а також проводити аналіз взаємодії компонентів без ризику завдання шкоди реальній інфраструктурі. Комп'ютерні станції повинні підтримувати технології апаратної віртуалізації, зокрема Intel VT-x або AMD-V, які дозволяють запускати гіпервізори першого та другого рівня з мінімальними накладними витратами ресурсів. Найбільш поширеними програмними платформами для реалізації таких середовищ є VMware Workstation Pro, Oracle VM VirtualBox та Microsoft Hyper-V. Кожна з них має свої особливості, однак усі забезпечують базові можливості із створення, конфігурації та ізоляції віртуальних машин, які імітують різні компоненти мережевої інфраструктури – від кінцевих станцій до серверів, маршрутизаторів, міжмережевих екранів та засобів реагування на інциденти. У процесі побудови віртуалізованого стенду доцільним є використання внутрішньомережевого з'єднання між віртуальними машинами, що імітує реальні сценарії розповсюдження загроз усередині локальної мережі. Також доцільним є створення окремих віртуальних сегментів з різним рівнем захисту для дослідження поведінки шкідливого ПЗ, експлуатації вразливостей або оцінки ефективності засобів виявлення загроз.

Програмне забезпечення, що використовується у процесі оцінювання кібербезпеки, повинно охоплювати як інструменти для активного аналізу, так і засоби пасивного моніторингу. До базового набору входить Wireshark – один із найпоширеніших інструментів для перехоплення та детального аналізу мережевого трафіку, що дозволяє фільтрувати пакети за різними параметрами, виявляти підозрілі сесії, визначати використані протоколи та потенційні спроби вторгнення. У свою чергу, Nmap забезпечує можливості сканування портів, виявлення активних хостів, типів операційних систем та служб, що працюють у мережі. Metasploit Framework слугує інструментом для моделювання атак, тестування вразливостей та перевірки ефективності механізмів захисту. Burp Suite застосовується переважно для аналізу веб-додатків, включаючи тестування на вразливості типу SQL injection, XSS, CSRF, а також для перехоплення й модифікації HTTP/HTTPS-запитів. У контексті оцінювання рівня вразливості інфраструктури часто використовуються системи типу OpenVAS, які забезпечують автоматизоване сканування та генерацію звітів щодо виявлених проблем. Для цілей мережевого моніторингу та виявлення аномалій важливе місце займають інструменти Suricata та Zeek, які реалізують функції IDS/IPS, глибокий аналіз трафіку та побудову поведінкових моделей. Suricata підтримує сигнатурний аналіз із високою продуктивністю, у той час як Zeek забезпечує більш гнучку логіку аналізу подій та зберігає структуровані журнали активності, що можуть бути інтегровані у централізовану систему збору та аналізу даних. Для обробки великих обсягів логів, візуалізації подій і побудови аналітичних дашбордів доцільно використовувати стек ELK, який дозволяє зберігати, обробляти й інтерпретувати події з різних джерел, в тому числі з IDS/IPS-систем, SIEM-платформ та серверів журналювання. Синергія зазначених програмних компонентів, розгорнутих на апаратно підготовлених комп’ютерних станціях, створює комплексне середовище для дослідження, оцінки та вдосконалення систем кіберзахисту в корпоративних мережах.

Не менш важливим аспектом при організації комп’ютерних станцій для оцінювання кібербезпеки є забезпечення їхнього власного захисту, оскільки такі станції самі по собі можуть стати об’єктами атак або джерелами витоку інформації у

разі компрометації. У зв’язку з цим, доцільно реалізовувати комплексну систему захисту, яка охоплює як організаційні, так і технічні заходи. Першочерговим є забезпечення фізичної та логічної ізоляції тестових середовищ, що дозволяє уникнути небажаного поширення активностей віртуалізованих загроз на продуктивну інфраструктуру або зовнішні мережі. Важливим є обмеження прав доступу до таких станцій: рекомендовано застосовувати багаторівневу автентифікацію, принцип найменших привілеїв та окремі облікові записи для адміністративних і звичайних операцій. Регулярне оновлення операційних систем та прикладного програмного забезпечення є ключовою умовою підтримки актуального рівня безпеки, з огляду на швидкий розвиток нових векторів атак і виявлення вразливостей у відомих продуктах. Окрім цього, на кожній робочій станції доцільно налаштувати ведення системних журналів подій із подальшою їх передачею у централізовану систему зберігання та аналізу (наприклад, SIEM або log-server), що дає змогу здійснювати ретроспективний аналіз інцидентів. Незважаючи на специфіку використання, станції повинні бути оснащені антивірусним або антималварним програмним забезпеченням, здатним виявляти як поширені, так і спеціалізовані загрози, особливо при роботі з потенційно шкідливими файлами або трафіком. Ще одним ефективним засобом підвищення безпеки є використання механізмів ізоляції процесів на рівні програмного забезпечення, таких як sandboxing, що дозволяє запускати потенційно небезпечні файли у контрольованому середовищі без доступу до основної системи. Комплексна реалізація вищезазначених заходів гарантує стабільність і безпеку роботи станцій, що беруть участь в оцінці кібербезпеки, та мінімізує ризики, пов’язані з виконанням активних досліджень або тестуванням шкідливих сценаріїв у межах корпоративної мережевої інфраструктури.

Розглянемо також процеси, що відбуваються в корпоративних мережах і цікаві з точки зору кібербезпеки. В першу чергу цікавим є мережевий трафік, оскільки через мережеву взаємодію відбуваються фактично всі кіберінциденти зовнішнього та внутрішнього характеру, тому він підлягає аналізу. Трафік може мати різну природу та походження, і його класифікація має важливе значення для вибору методів контролю, збору даних та виявлення потенційних загроз. До основних типів трафіків

відносять внутрішній, зовнішній та міжсегментний.

Внутрішній мережевий трафік є одним із ключових об’єктів дослідження в рамках оцінювання кібербезпеки корпоративних мереж, оскільки він відображає комунікацію між вузлами всередині організації, включаючи взаємодію користувацьких пристроїв, серверів, мережевого обладнання та інших системних компонентів. Саме в межах цього типу трафіку можуть реалізовуватись цільові атаки, що обходять периметрові засоби захисту, особливо у випадках, коли загроза походить зсередини – через компрометацію облікових записів, встановлення шкідливого програмного забезпечення або дії зловмисника, що має прямий доступ до внутрішньої інфраструктури. Внутрішній трафік є складним за своєю структурою, оскільки включає широкий спектр протоколів і сервісів: від звичних DNS-, DHCP- та SMB-запитів до специфічної міжсервісної взаємодії в мікросервісних архітектурах або системах з розподіленими базами даних. Його характер тісно пов’язаний із архітектурою мережі, політиками маршрутизації, логікою роботи сервісів і шаблонами поведінки користувачів, тому навіть незначні відхилення від нормального функціонування можуть сигналізувати про потенційні загрози.

З точки зору безпеки важливою є здатність ідентифікувати підозрілу активність на фоні легітимної взаємодії: наприклад, спроби горизонтального переміщення по мережі (lateral movement), сканування портів, нестандартні з’єднання між сегментами або аномальні обсяги переданих даних. Крім того, внутрішній трафік часто використовується як канал для розповсюдження шкідливих програм, таких як ransomware або spyware, а також як засіб ексфільтрації даних при реалізації складених атак типу APT (Advanced Persistent Threat). Моніторинг цього трафіку дозволяє не лише виявити активні фази атаки, але й отримати індикатори компрометації, що можуть бути використані для поширення правил виявлення на інші частини мережі. Для ефективного аналізу внутрішнього трафіку застосовуються як сигнатурні методи (виявлення за відомими шаблонами), так і поведінкові підходи, зокрема машинне навчання для виявлення відхилень від стандартних моделей взаємодії.

Зовнішній мережевий трафік охоплює всю мережеву взаємодію між внутрішніми ресурсами корпоративної мережі та зовнішніми об'єктами, що

знаходяться поза межами адміністративного контролю організації. Такий трафік є найбільш ризикогенним з точки зору проникнення загроз, оскільки саме через нього реалізується більшість атак ззовні: від автоматизованих сканувань портів до цільових спроб експлуатації вразливостей в інтернет-експонованих сервісах. Характер зовнішнього трафіку включає як ініційовані зсередини запити (наприклад, HTTP/HTTPS-з'єднання до вебресурсів, звернення до зовнішніх API, оновлення ПЗ), так і вхідні з’єднання, що можуть свідчити про спроби вторгнення, C2-комунікацію (command and control), або ж інші форми несанкціонованого доступу. З метою зменшення ризиків, пов’язаних із зовнішнім трафіком, застосовуються міжмережеві екрани, системи фільтрації на рівні додатків, а також проксі-сервери, які дозволяють не лише обмежити канали зв’язку, а й фіксувати й аналізувати вміст запитів та відповідей.

Зовнішній трафік також є джерелом критично важливої інформації для виявлення ранніх ознак компрометації, зокрема при наявності нетипових звернень до невідомих доменів, нестандартних портів або географічно нетипових регіонів. У таких випадках особливо ефективними є інструменти аналізу DNS-запитів, а також механізми порівняння звернень із базами відомих загроз, включаючи IOC (indicators of compromise) та TI-фіди (threat intelligence feeds). Для виявлення шкідливої активності в зовнішньому трафіку дедалі ширше застосовуються технології глибокої інспекції пакетів (DPI), що дозволяють аналізувати структуру запитів, SSL-сертифікати, заголовки протоколів і поведінкові характеристики з’єднань. Окрім цього, зовнішній трафік може виступати як канал ексфільтрації інформації – у прихованому або обфускованому вигляді, наприклад, через DNS-тунелювання, HTTP-запити або TOR-мережу, тому його ретельний аналіз є невід’ємною частиною системи кіберзахисту.

Міжсегментний трафік охоплює обмін даними між логічно або фізично відокремленими сегментами всередині корпоративної мережі, зокрема між підрозділами, серверними зонами, зонами з обмеженим доступом (DMZ), а також міжмережевими інтерфейсами різного призначення. Такий трафік є особливо важливим з точки зору кібербезпеки, оскільки саме через взаємодію між сегментами

можуть відбуватись несанкціоновані переміщення зловмисника у разі компрометації одного з вузлів, а також горизонтальне розповсюдження шкідливого програмного забезпечення, включаючи експлуатацію внутрішніх сервісів, які часто мають слабкі або неналаштовані механізми аутентифікації. У розвинених корпоративних мережах сегментація є базовою практикою, що забезпечує розмежування доступу, зменшує площину атаки та підвищує ефективність контролю трафіку. Проте на практиці саме міжсегментна взаємодія часто лишається недостатньо задокументованою, що ускладнює виявлення відхилень і потенційних аномалій.

Захист міжсегментного трафіку передбачає впровадження політик контролю доступу на рівні міжмережевих екранів внутрішнього периметра, застосування VLAN-ів, мікросегментації, а також контроль транзитних з'єднань із використанням мережевих шлюзів та інспекторів трафіку. Особливу цінність у контексті безпеки становить виявлення нетипових з'єднань між сегментами, які раніше не фіксувались у профілях звичної поведінки мережі, або ж створення нових маршрутів, що можуть свідчити про спроби обійти існуючі обмеження. У таких випадках важливим є контекстуальний аналіз: ідентифікація, хто саме (з якого джерела, за яким протоколом, з якими правами доступу) ініціював трафік між сегментами. Це дозволяє виявляти ознаки внутрішнього зловмисника, а також компрометацію сервісних облікових записів, що мають доступ до критичних вузлів мережі.

Крім класифікації мережевого трафіку, важливою складовою аналізу з позиції кібербезпеки є вивчення типових процесів, що відбуваються в комп'ютерних мережах і визначають характер їхньої повсякденної функціональності. Такі процеси не є випадковими або хаотичними – навпаки, вони формують передбачувану поведінкову модель системи, відхилення від якої може свідчити про наявність загроз, як внутрішніх, так і зовнішніх. Усвідомлення закономірностей мережевої активності дозволяє не лише виявляти інциденти в реальному часі, а й вбудовувати логіку аналізу в архітектуру системи захисту. До базових категорій процесів належать, зокрема, запити доступу до внутрішніх ресурсів: серверів файлів, поштових систем, баз даних, репозиторіїв, платформ спільної роботи, внутрішніх порталів, CRM- або ERP-систем. Кожна така взаємодія породжує характерний набір мережевих запитів,

відповідей і додаткових службових дій, які можуть бути типізовані, нормалізовані й перевірені на відповідність очікуваному патерну. У разі відхилення – наприклад, нетиповий час доступу, аномальна частота звернень або спроба отримати великі обсяги даних – система може зафіксувати це як підозрілу активність.

Окремий масив процесів формують механізми автентифікації та авторизації. Вони забезпечують перевірку користувачів, пристроїв і сервісів перед наданням доступу до мережевих ресурсів і тому мають критичне значення для побудови системи контролю доступу. Такі процеси реалізуються з використанням різних протоколів і механізмів, серед яких найпоширенішими є LDAP, Kerberos, NTLM, RADIUS або SAML. Вони можуть передбачати як однофакторну, так і багатофакторну автентифікацію, а також реалізацію механізмів Single Sign-On (SSO). Зловмисники часто намагаються маніпулювати саме цим етапом – наприклад, через підбір облікових даних, повторне використання скомпрометованих паролів або захоплення токенів автентифікації. Моніторинг процесів автентифікації дозволяє виявляти спроби доступу з невластивих геолокацій, нестандартних агентів користувача або ізолювати поведінку, що відрізняється від історичних шаблонів. Такі спроби можуть бути раннім індикатором проникнення або підготовки до більш масштабної атаки, і тому ці процеси мають фіксуватися з максимальною точністю та з подальшим аналізом в рамках системи виявлення загроз.

Важливо також розглядати процеси адміністрування, які включають усі дії, пов’язані з технічним супроводом мережі – налаштування обладнання, встановлення оновлень, модифікацію політик доступу, розгортання нових сервісів, ведення журналів, застосування скриптів автоматизації тощо. В умовах великої корпоративної мережі такі процеси часто відбуваються у визначені часові вікна, з передбачуваних джерел і під обліковими записами з відповідними правами доступу. Саме тому будь-які відхилення – зокрема позапланові адміністративні підключення, активація засобів керування з незвичних адрес або нетипові зміни в конфігураціях – можуть свідчити про несанкціоновану активність. Ще однією групою процесів є регулярна службова взаємодія між компонентами мережі: запити до DNS-, DHCP-, NTP- або оновлювальних серверів, перевірка цифрових підписів, синхронізація часу та

конфігурацій. Хоча такі процеси є фоновими, їхній аналіз дозволяє виявити нестандартні вектори атак, як-от використання DNS-тунелювання або підміна відповіді DHCP-сервера з метою маніпуляції маршрутизацією. До категорії регулярних, але потенційно вразливих процесів належать також резервне копіювання, синхронізація даних між дата-центрами, періодична передача логів на SIEM-сервери. Їхній обсяг, напрям і частота – сталі характеристики, тому будь-які спроби використання цих каналів для прихованої ексфільтрації даних або передачі команд шкідливим агентам підлягають обов'язковому виявленню й аналізу. У сукупності ці процеси формують фундамент повсякденної мережевої активності, яка в умовах кіберзагроз набуває характеру постійно контрольованого, змінного середовища, де навіть мінімальні відхилення можуть мати суттєві наслідки для інформаційної безпеки організації.

Таким чином, ефективність архітектури системи оцінювання кібербезпеки значною мірою залежить від точності збору, інтерпретації та оновлення інформації про параметри комп'ютерних станцій, зокрема конфігурації апаратного забезпечення, версій операційних систем, відкритих портів, активних служб і використаних протоколів. У динамічному середовищі корпоративних мереж, де постійно змінюється склад підключених вузлів та інтенсивність інформаційних потоків, надзвичайно важливим є безперервний аналіз мережевих процесів, включаючи маршрутизацію трафіку, стан каналів зв'язку та поведінкові шаблони взаємодії між компонентами. Недооцінка цих аспектів здатна призвести до неповного виявлення вразливостей і, як наслідок, до підвищення ризику несанкціонованого доступу або порушення цілісності даних у корпоративній інфраструктурі.

2.3 Висновки до другого розділу

У другому розділі було сформовано архітектуру системи оцінювання кібербезпеки корпоративних мереж, яка враховує специфіку мережевої взаємодії, динамічність інфраструктури та необхідність об'єктивного аналізу технічного стану вузлів. Розглянуто ключові функціональні компоненти системи, зокрема апаратне

забезпечення, механізми збору даних, виявлення вразливостей, аналізу мережевих з’єднань і перевірки політик доступу. Особливу увагу приділено використанню методології CVSS як основи для уніфікованого оцінювання ризиків. Розглянуто архітектурні рішення для створення системи оцінювання кібербезпеки.

3 МОДЕЛІ ЗАГРОЗ ТА ОЦІНЮВАННЯ КІБЕРБЕЗПЕКИ КОМП'ЮТЕРНИХ СТАНЦІЙ

3.1 Комп'ютерні атаки та зловмисне програмне забезпечення, які впливають на рівень кібербезпеки

Корпоративні мережі є складними, багаторівневими інформаційними системами, які забезпечують взаємодію внутрішніх та зовнішніх користувачів, автоматизованих систем управління, баз даних, сервісів аутентифікації та багатьох інших компонентів. Захист таких мереж вимагає постійного моніторингу активності, аналізу можливих векторів атак та виявлення аномалій у режимі реального часу. Оцінка кібербезпеки корпоративної мережі ґрунтується на визначенні рівня загроз, що можуть вплинути на конфіденційність, цілісність та доступність даних. В умовах стрімкого розвитку технологій атакуючі отримують у своє розпорядження нові методи компрометації систем, що робить необхідним впровадження інтелектуальних систем оцінювання кібербезпеки.

Зловмисне програмне забезпечення та комп'ютерні атаки є основними факторами, які впливають на рівень кібербезпеки корпоративних мереж. Використовуючи автоматизовані системи аналізу, можна визначити ймовірність успішної атаки в конкретний момент часу та вжити відповідних заходів для її запобігання. Основні типи атак та зловмисного ПЗ, що мають найбільший вплив на корпоративні системи, включають розширені атаки персистентних загроз, DDoS-атаки, використання уразливостей нульового дня, атаки на ланцюги постачання, фішингові атаки, впровадження шкідливих бекдорів, руткітів, ботнетів, вірусів-шифрувальників та маніпуляції з мережевими протоколами.

Розширені атаки персистентних загроз є складними багатоступінчастими атаками, що здійснюються протягом тривалого періоду часу. Головна мета таких атак – отримати тривалий доступ до мережі жертви, залишаючись непоміченим. Зазвичай вони починаються з фази розвідки, коли атакуючі збирають інформацію про систему, її користувачів, використовувані технології та захисні механізми. Після цього зловмисники використовують уразливості програмного забезпечення або соціальну

інженерію для проникнення в систему. Після первинного проникнення вони встановлюють бекдори або руткіти, які забезпечують їм стабільний доступ. Для захисту від таких атак корпоративна мережа повинна використовувати поведінковий аналіз трафіку та механізми раннього виявлення аномальної активності.

DDoS-атаки спрямовані на перевантаження серверів, що призводить до порушення доступності корпоративних сервісів. Вони можуть здійснюватися через ботнети, що складаються з великої кількості заражених пристроїв, які координовано надсилають запити до цільової системи. Такі атаки можуть тривати від кількох хвилин до кількох діб, що суттєво впливає на роботу компанії. Боротьба з DDoS-атаками вимагає використання систем фільтрації трафіку, балансування навантаження та механізмів швидкого виявлення аномальної активності.

Атаки нульового дня є особливо небезпечними, оскільки вони експлуатують уразливості, які ще не були виправлені розробниками програмного забезпечення. Оскільки такі уразливості невідомі антивірусним системам та інструментам моніторингу, їх виявлення вимагає аналізу поведінки виконуваних процесів. Після виявлення подібних атак компанії повинні негайно застосовувати патчі та оновлення системи безпеки.

Атаки на ланцюги постачання є складними багатоступінчастими атаками, коли атакуючі компрометують постачальників програмного або апаратного забезпечення, щоб згодом отримати доступ до цільових компаній. Уразливості в системах управління оновленнями або слабкі місця у процесах перевірки довірених сторонніх сервісів можуть сприяти успішності таких атак. Одним із методів запобігання є ретельний аналіз усіх вхідних оновлень, ізоляція критично важливих компонентів системи та впровадження механізмів перевірки достовірності коду.

Фішингові атаки є одним із найпоширеніших методів отримання несанкціонованого доступу до корпоративних мереж. Вони можуть здійснюватися через електронні листи, повідомлення у месенджерах або навіть телефонні дзвінки. Найбільш небезпечним є цільовий фішинг, коли зловмисники атакують конкретних співробітників, що мають доступ до критично важливих даних. Успішний фішинг може призвести до викрадення облікових даних, зараження системи шкідливим ПЗ

або маніпуляції користувачами. Для захисту необхідно впроваджувати багатофакторну аутентифікацію, навчати співробітників розпізнавати підозрілі повідомлення та використовувати системи аналізу вхідного трафіку.

Використання ботнетів дозволяє атакуючим організовувати масові атаки на корпоративну інфраструктуру або використовувати скомпрометовані ресурси для шахрайських дій. Ботнети можуть складатися як із серверних систем, так і з IoT-пристроїв, які мають слабкий рівень захисту. Виявлення ботнетів потребує аналізу мережевого трафіку, виявлення аномальних запитів та ізоляції скомпрометованих вузлів.

Руткіти та бекдори є небезпечними видами зловмисного ПЗ, які дозволяють атакуючим непомітно контролювати корпоративну мережу. Вони можуть змінювати системні журнали, обходити засоби антивірусного захисту та забезпечувати зловмисникам прихований віддалений доступ до системи. Найефективнішими методами захисту є використання механізмів контролю цілісності системи, моніторинг активності процесів та періодичний аудит конфігураційних файлів.

Віруси-шифрувальники можуть спричинити значні фінансові втрати компанії, блокуючи доступ до критичних даних. Вони поширюються через фішингові атаки, заражені оновлення або експлуатацію мережевих уразливостей. Після шифрування файлів зловмисники вимагають викуп, і навіть у разі його сплати немає гарантій відновлення даних. Єдиним надійним методом боротьби є створення регулярних резервних копій та впровадження строгих політик контролю доступу.

Маніпуляції з мережевими протоколами, такі як атаки MITM або DNS-спуфінг, дозволяють зловмисникам перехоплювати, підміняти або підслуховувати трафік. Це може призвести до викрадення облікових даних, змінення критичних бізнес-операцій або компрометації внутрішніх серверів. Захист від таких атак вимагає використання шифрування трафіку, впровадження VPN-рішень та активного моніторингу підозрілих змін у мережевих маршрутах.

SQL-ін'єкції залишаються однією з найнебезпечніших та найпоширеніших атак на корпоративні системи, оскільки бази даних є центральним сховищем критично важливої інформації, включаючи комерційні та фінансові записи, конфіденційні дані

клієнтів, внутрішні документи компанії та логіни користувачів. Основна ідея SQL-ін'єкції полягає у впровадженні шкідливих SQL-запитів через вхідні поля веб-додатків або API, що взаємодіють із базами даних. Це дозволяє зловмиснику виконувати довільні команди в системі управління базами даних (СУБД), отримуючи несанкціонований доступ до інформації, змінюючи її або повністю видаляючи. Вразливості такого типу виникають через неналежну валідацію вхідних даних, коли веб-додаток дозволяє виконувати введений користувачем SQL-код без відповідного екранування спеціальних символів. Найбільш критичними є атаки, що дозволяють не тільки отримувати дані з бази, а й змінювати їх, видаляти або створювати нові користувацькі облікові записи, що дає атакуючим можливість отримати повний контроль над системою. Для захисту від SQL-ін'єкцій рекомендується використовувати підготовлені запити (prepared statements), ORM (Object-Relational Mapping), багаторівневу аутентифікацію доступу до бази даних та регулярний моніторинг підозрілої активності запитів. Проте навіть при правильній архітектурі додатка людський фактор, помилки розробників або використання застарілих бібліотек можуть створювати ризики, які слід мінімізувати шляхом автоматизованого аналізу безпеки.

Атаки типу ransomware стали особливо руйнівними для корпоративних мереж, оскільки вони не тільки блокують доступ до критичних даних, а й можуть призвести до значних фінансових втрат, паралізуючи роботу компаній на тривалий період. Після проникнення в систему, шкідливе програмне забезпечення шифрує файли на серверах, робочих станціях та навіть резервних копіях, залишаючи постраждалих без можливості відновлення інформації без спеціального ключа дешифрування, який зловмисники пропонують викупити за криптовалюту. Віруси-шифрувальники поширюються через фішингові атаки, заражені вкладення електронної пошти, експлуатацію вразливостей у віддалених протоколах доступу, а також через компрометовані оновлення програмного забезпечення. Деякі сучасні версії ransomware також реалізують механізми подвійного вимагання, коли перед шифруванням дані ексфільтруються на сервери зловмисників, що дозволяє їм додатково шантажувати жертву загрозою витоку інформації у відкритий доступ.

Захист від таких атак вимагає регулярного створення зашифрованих резервних копій у відокремлених середовищах, жорсткого контролю доступу до критичних систем та застосування механізмів розширеного моніторингу активності файлів, які можуть сигналізувати про нетипові операції шифрування. Однак навіть такі заходи не дають абсолютної гарантії безпеки, оскільки зловмисники постійно вдосконалюють свої методи, обходячи традиційні засоби захисту.

Атаки типу "людина посередині" (Man-in-the-Middle, MITM), DNS-спуфінг та ARP-спуфінг є одними з найбільш витончених способів компрометації мережевого трафіку корпоративних систем. Використовуючи ці методи, атакуючі можуть підміняти трафік між користувачами та серверами, викрадати конфіденційну інформацію або навіть змінювати передані дані. У випадку MITM-атак зловмисник отримує можливість повністю контролювати мережеву комунікацію, що дозволяє йому впроваджувати шкідливий код, перехоплювати паролі або маніпулювати внутрішніми корпоративними системами. DNS-спуфінг передбачає підміну записів у системі доменних імен, що змушує жертв переходити на підроблені веб-сайти, де можуть викрадатися їхні облікові дані або поширюватися шкідливі програми. ARP-спуфінг використовується для перехоплення трафіку в локальних мережах шляхом підміни MAC-адрес у кеші ARP-запитів, що дозволяє атакуючим скеровувати трафік через власний пристрій. Для запобігання таким атакам слід використовувати шифрування TLS у всіх мережевих з'єднаннях, реалізовувати захист від підробки ARP-запитів та впроваджувати механізми моніторингу підозрілих змін у конфігурації мережевого середовища. Втім, багато атак MITM можна реалізувати навіть без прямого доступу до мережі компанії, використовуючи вразливості публічних Wi-Fi-зон або VPN-з'єднань, що додає ще більше загроз у корпоративному контексті.

Атаки через бекдори (Backdoor Injection) є ще одним способом довготривалої компрометації корпоративних мереж. Після початкового проникнення в систему зловмисники часто встановлюють бекдори – спеціальні приховані механізми, що дозволяють їм отримувати несанкціонований доступ навіть після усунення первинної уразливості. Такі механізми можуть бути вбудовані у зламані веб-додатки, серверні компоненти, драйвери пристроїв або навіть прошивки апаратного забезпечення.

Оскільки бекдори можуть бути налаштовані на активацію лише за певних умов, їхнє виявлення є надзвичайно складним завданням, що вимагає комплексного аналізу трафіку та поведінкової аналітики. Найнебезпечнішими є бекдори, вбудовані в критичні системи керування або безпекові компоненти корпоративних мереж, оскільки вони можуть залишатися непоміченими роками, надаючи зловмисникам прихований доступ до всієї інфраструктури компанії.

Використання ботнетів у корпоративних середовищах стає дедалі серйознішою загрозою, оскільки атакуючі можуть захоплювати контроль над великими групами серверів, що використовуються для масових атак або прихованого виконання шкідливих операцій. Заражені пристрої можуть використовуватися для розповсюдження зловмисного програмного забезпечення, проведення DDoS-атак або навіть шпигунства за корпоративною діяльністю. Одним із методів створення ботнетів є впровадження спеціального ПЗ через заражені оновлення або використання уразливостей в IoT-пристроях, що не мають належного рівня безпеки. Виявлення ботнет-активності вимагає аналізу аномальної поведінки пристроїв, а також використання систем IDS/IPS, які можуть ідентифікувати підозрілі комунікаційні патерни. Незважаючи на ці методи, сучасні ботнети часто використовують шифрування та децентралізовані структури керування, що значно ускладнює їх нейтралізацію.

Експлуатація уразливостей у хмарних середовищах стає все більш значущою загрозою для корпоративних мереж, оскільки все більше компаній переходять на використання хмарних технологій для зберігання та обробки даних. Основні ризики, пов'язані з хмарною інфраструктурою, включають витоки конфіденційної інформації, компрометацію облікових записів, використання неправильно налаштованих сховищ та атаки на API. Багато атакуючих експлуатують слабкі налаштування доступу до хмарних ресурсів, коли компанії залишають відкритими для загального доступу конфіденційні дані або резервні копії. Окрім цього, зловмисники можуть отримати доступ до хмарних сервісів через викрадені або вкрадені облікові дані, що дає їм змогу змінювати конфігурацію середовища, видаляти файли або впроваджувати шкідливе програмне забезпечення. Наприклад, атаки на незахищені API дозволяють

отримати доступ до внутрішніх сервісів компанії або маніпулювати функціоналом хмарних додатків. Для захисту від таких атак необхідно ретельно контролювати політики доступу, використовувати багатофакторну аутентифікацію, регулярно перевіряти налаштування безпеки хмарної інфраструктури та впроваджувати системи моніторингу підозрілої активності. Незважаючи на всі заходи безпеки, через складність і динамічність хмарних технологій навіть досвідчені IT-фахівці можуть не помітити певні вразливості, що робить їх привабливою мішенню для атакуючих.

Використання руткітів є однією з найнебезпечніших загроз для корпоративних систем, оскільки вони дозволяють зловмисникам приховано контролювати сервери, робочі станції та навіть інфраструктурні компоненти мережі. Руткіти можуть змінювати системні файли, перехоплювати мережевий трафік, маніпулювати процесами та залишатися непомітними для стандартних антивірусних рішень. Вони часто використовуються у поєднанні з бекдорами, забезпечуючи постійний доступ атакуючих до заражених пристроїв. Руткіти можуть бути впроваджені через експлуатацію вразливостей операційної системи, встановлення модифікованого драйвера або навіть через офіційні оновлення ПЗ, якщо їхні сертифікати безпеки були скомпрометовані. Найнебезпечнішими є руткіти на рівні ядра операційної системи, оскільки вони інтегруються в саму структуру ОС та можуть змінювати поведінку основних системних процесів. Виявлення та видалення руткітів є надзвичайно складним завданням, оскільки вони можуть використовувати методи обходу традиційних механізмів безпеки, такі як приховування своїх файлів, маніпуляція журналами подій та навіть підміна результатів запитів до системних команд. Для боротьби з руткітами необхідно використовувати спеціалізовані засоби аналізу системної активності, контролювати цілісність операційної системи та забезпечувати використання тільки довірених драйверів та програмного забезпечення. Проте навіть у разі виявлення руткіту його повне видалення може потребувати повного форматування системи та відновлення даних з чистого резервного копіювання.

Атаки на облікові записи користувачів є одним із найефективніших методів отримання несанкціонованого доступу до корпоративних ресурсів. Використання методів підбору паролів, атак типу brute force та password spraying дозволяє

атакуючим проникати в корпоративні системи, навіть якщо вони захищені паролями. Brute force атаки ґрунтуються на спробах перебору всіх можливих комбінацій паролів до моменту знаходження правильного значення, тоді як password spraying використовує підхід масового випробування популярних або слабких паролів на великій кількості облікових записів. Така стратегія допомагає уникнути блокування користувачів після надмірної кількості невдалих спроб входу. Якщо зловмисники отримують доступ до облікових даних адміністратора або користувача з високими привілеями, вони можуть змінювати політики доступу, зчитувати конфіденційну інформацію, впроваджувати шкідливий код або навіть видаляти критично важливі дані. Основними методами захисту є застосування багатофакторної аутентифікації, обмеження кількості невдалих спроб входу та впровадження політик складності паролів, однак багато компаній продовжують нехтувати цими заходами безпеки, що робить їх уразливими до атак на облікові записи.

Атаки на системи аутентифікації, такі як session hijacking та MFA bypass, дозволяють атакуючим обійти механізми безпеки та отримати доступ до корпоративних ресурсів без використання реальних облікових даних. Session hijacking передбачає викрадення або підміну сесій користувачів, що дозволяє зловмиснику використовувати активну сесію для виконання дій від імені жертви. Для цього атакуючі можуть використовувати методи перехоплення трафіку, атаки на незахищені файли cookie або експлуатацію уразливостей у механізмах управління сесіями. У свою чергу, MFA bypass передбачає обходи багатофакторної аутентифікації шляхом використання фішингових атак, компрометації токенів або маніпуляцій із механізмами відновлення паролів. Якщо атакуючий отримує контроль над сесією адміністратора, він може змінювати конфігурацію корпоративної мережі, видаляти або створювати нові облікові записи, впроваджувати шкідливий код у систему або змінювати критичні бізнес-процеси. Для запобігання таким атакам необхідно впроваджувати додаткові рівні перевірки безпеки, такі як поведінковий аналіз користувачів, аналіз IP-адрес та географічного розташування, а також шифрування токенів сесій та автоматичне їх завершення у разі підозрілої активності.

Маніпуляція оновленнями програмного забезпечення є ще одним критичним

вектором атак, який дозволяє зловмисникам впроваджувати бекдори, трояни та інші види шкідливого ПЗ через офіційні оновлення програмних продуктів. Атаки цього типу є особливо небезпечними, оскільки користувачі довіряють оновленням від офіційних постачальників та автоматично встановлюють їх у свої системи. Якщо атакуючий компрометує сервери оновлення або підміняє файли інсталяції, він може розповсюджувати шкідливе ПЗ у масштабах цілих корпоративних мереж. Наприклад, у випадку успішної атаки зловмисники можуть отримати можливість впровадження бекдорів, викрадення конфіденційних даних, виконання шкідливих команд на пристроях жертв або навіть повного руйнування корпоративної IT-інфраструктури. Захист від маніпуляцій із оновленнями включає перевірку цифрових підписів оновлень, використання сегментованих середовищ для тестування нових версій програмного забезпечення та моніторинг усіх змін у встановлених програмах. Однак навіть при цих заходах компанії, які покладаються на автоматичне оновлення ПЗ без попереднього аналізу, залишаються уразливими до подібних атак.

Використання заражених контейнерів у корпоративних DevOps-середовищах стає дедалі серйознішою загрозою для компаній, що активно використовують контейнеризацію та мікросервісну архітектуру. Docker-контейнери забезпечують ізоляцію процесів, гнучке масштабування та ефективне використання ресурсів, проте їхня популярність робить їх привабливою мішенню для атакуючих. Одним із головних ризиків є впровадження шкідливого коду у публічно доступні образи контейнерів, які розробники або системні адміністратори можуть використовувати без належної перевірки. Зловмисне ПЗ у контейнерах може містити приховані бекдори, криптомайнери, механізми викрадення облікових даних або навіть експлойти, що спрямовані на компрометацію базової інфраструктури, на якій запускається контейнеризоване середовище. Якщо заражений контейнер отримує доступ до корпоративної мережі, він може використовувати внутрішні API, проникати у бази даних, виконувати несанкціоновані запити або навіть здійснювати горизонтальне переміщення мережею, що робить його потужним інструментом для кібератак.

Зловмисники можуть використовувати кілька методів поширення заражених

контейнерів. Один із найбільш поширених – це публікація модифікованих образів у загальнодоступних сховищах контейнерів, таких як Docker Hub. Оскільки розробники часто використовують публічні образи для прискорення процесу розгортання сервісів, вони можуть випадково завантажити контейнер із прихованими загрозами. Інший метод полягає у компрометації власних приватних репозиторіїв компанії або атаках на CI/CD-пайплайни, де зловмисники можуть впроваджувати бекдори у процеси автоматизованої збірки контейнерів. Ще одним небезпечним вектором є атаки на Kubernetes-кластери, де через неправильні налаштування безпеки атакуючі можуть впроваджувати заражені контейнери у критичні частини корпоративної інфраструктури.

Компрометація пристроїв IoT є ще одним серйозним ризиком для корпоративних мереж, оскільки велика кількість організацій використовує інтернет-речей для моніторингу, автоматизації та керування технологічними процесами. Різноманітні IoT-пристрої, такі як інтелектуальні камери відеоспостереження, системи контролю доступу, сенсори та промислові контролери, часто мають слабкий рівень безпеки, що робить їх вразливими для атак. Основна проблема полягає у тому, що багато пристроїв IoT використовують застарілі прошивки, мають дефолтні паролі або некоректно налаштовані протоколи зв’язку, що дозволяє атакуючим отримати несанкціонований доступ. Одним із найбільш поширених методів атаки є використання ботнетів, таких як Mirai, які сканують мережу на наявність незахищених IoT-пристроїв, після чого їх заражають та використовують для DDoS-атак, прихованого прослуховування або несанкціонованого доступу до корпоративної інфраструктури.

Іншим небезпечним вектором є атаки на IoT через внутрішні мережі компаній. Оскільки багато пристроїв мають прямий доступ до корпоративних серверів, атакуючі можуть використовувати скомпрометовані IoT-пристрої як точку входу в мережу. Наприклад, якщо зловмисники отримують контроль над мережею, де працює система відеоспостереження, вони можуть не тільки здійснювати спостереження за діяльністю компанії, але й використовувати ці пристрої для атак на інші сегменти мережі. Ще одним варіантом є використання атак через бездротові технології зв’язку,

такі як Bluetooth, Zigbee або NFC, що дозволяє атакуючим отримати контроль над пристроями, не маючи безпосереднього доступу до внутрішньої мережі.

Небезпечним аспектом компрометації IoT є можливість проведення атак типу "pivoting", коли атакуючі використовують зламаний пристрій IoT для подальшого руху мережею. Це особливо актуально для промислових систем, де IoT-пристрої можуть бути інтегровані з критичними технологічними процесами, такими як SCADA-системи, автоматизовані виробничі лінії або системи моніторингу енергетичної інфраструктури. Якщо зловмисники отримують контроль над такими пристроями, вони можуть змінювати налаштування, фальсифікувати дані або навіть повністю вивести з ладу виробничі потужності.

Таким чином, у межах корпоративних мереж розподіленого типу особливу увагу слід приділяти зростаючій складності кіберзагроз, які проявляються через багаторівневі атаки, уразливості в компонентах систем та непередбачувану поведінку мережевого трафіку. Наявність великої кількості точок входу, асинхронна взаємодія між вузлами та потенційна недовіра до окремих елементів інфраструктури створюють передумови для широкого спектру загроз, включаючи атаки типу відмова в обслуговуванні, проникнення через незахищені протоколи та експлуатацію конфігураційних помилок. Усі ці чинники підкреслюють необхідність постійного, детального оцінювання стану кібербезпеки з урахуванням динаміки мережевого середовища, обмежень ресурсів і труднощів у досягненні повної прозорості розподілених систем.

3.2 Функція оцінювання кібербезпеки комп’ютерних станцій

Задамо дві функцію для оцінювання рівня захищеності мережі, де перша буде відображати ймовірність суттєвого втручання зловмисника в будь-яку критичну компоненту мережі.

Спочатку визначимо вразливість компоненти як ймовірність її компрометації незалежно від решти, присутніх в мережі. Вона буде задаватися формулою:

$$V = \omega_S S + \omega_P(1 - P) + \omega_U U, \quad (3.1)$$

де S – рівень вразливості ПЗ, нормалізований у діапазоні [0,1], P – ефективність політик безпеки (від 0 до 1, де 1 – максимальний рівень безпеки), U – ймовірність компрометації через людський фактор, $\omega_S, \omega_P, \omega_U$ – вагові коефіцієнти.

Розкриємо складові формули далі. P має бути визначено спеціалістами з кібербезпеки незалежно для кожного окремого випадку, оскільки у різних організаціях прийняті різні підходи до налагодження відповідних процесів. В контексті даної роботи визначатимемо U відповідно до частоти фішингових атак та інших ситуацій компроментування користувачів мережі в її історії. S визначатимемо за формулою

$$S = \sum_{k=1}^{N_e} \omega_k * \frac{CVSS_k}{10}, \quad (3.2)$$

де N_e – загальна кількість вразливостей на вузлі, $CVSS_k$ – оцінка критичності k-ї вразливості за шкалою CVSS (від 0 до 10), ω_k – ваговий коефіцієнт, що визначає вплив кожної вразливості (наприклад, 1 для активних експлойтів, 0.5 для застарілих вразливостей, що не мають активних атак).

Пошук вразливостей для обчислення S може бути організований за допомогою сканерів вразливостей.

Таким чином, формула вразливості однієї компоненти незалежно від решти мережі:

$$V = \omega_S \sum_{k=1}^{N_e} \omega_k * \frac{CVSS_k}{10} + \omega_P(1 - P) + \omega_U U, \quad (3.3)$$

Також необхідно враховувати, що компрометація одного хосту в мережі наражає на небезпеку також і інші компоненти мережі. Для цього задамо формулу для визначення ймовірності компрометації хоста j, якщо був скомпроментований хост i:

$$G_{ij} = \omega_T T_{ij} + \omega_F(1 - F_{ij}) + \omega_L(1 - L_{ij}), \quad (3.4)$$

де T_{ij} – рівень відкритості з'єднання (нормалізований в діапазоні [0,1]), де 1 означає повністю відкритий канал, а 0 – повністю ізольоване з'єднання, F_{ij} – ефективність фаєрволів і фільтрації трафіку (від 0 до 1, де 1 означає максимальний захист), L_{ij} – рівень шифрування (від 0 до 1, де 1 означає повне шифрування, а 0 – повністю відкритий трафік).

Зведемо разом ці дві формули, щоб для кожного хосту визначити ймовірність його компрометації і відповідно розрахуємо шанс компрометації будь-якого із важливих хостів.

$$CS = \prod_{i=1}^{M} \left(\left(1 - V_{a_i}\right) * \prod_{j=1}^{N} \left(1 - V_j G_{j,a_i}\right) \right), \quad (3.5)$$

де CS – загальний рівень кібербезпеки у корпоративній мережі, M – кількість важливих компонент мережі, a – список важливих компонент мережі.
Для прикладу розглянемо просту мережу з 4 хостами, з яких хости 1, 2 та 3 важливі.

Таблиця 3.1 – Вразливості хоста №1

№ вразливості	Ваговий коефіцієнт ω_k	CVSS
1	0.4	1
2	0.4	2
3	0.2	5

Таблиця 3.2 – Вразливості хоста №2

№ вразливості	Ваговий коефіцієнт ω_k	CVSS
1	0.4	2
2	0.4	3
3	0.1	5
4	0.1	1

Таблиця 3.3 – Вразливості хоста №3

№ вразливості	Ваговий коефіцієнт ω_k	CVSS
1	0.3	3
2	0.6	2
3	0.1	4

Таблиця 3.4 – Вразливості хоста №4

№ вразливості	Ваговий коефіцієнт ω_k	CVSS
1	0.3	7
2	0.4	6
3	0.3	4

На основі даних про відомі вразливості кожного хоста в системі можемо розрахувати для них S.

$$S_1 = \frac{(0.4 * 1 + 0.4 * 2 + 0.2 * 5)}{10} = 0.22$$

$$S_2 = \frac{(0.4 * 2 + 0.4 * 3 + 0.1 * 5 + 0.1 * 1)}{10} = 0.26$$

$$S_3 = \frac{(0.3 * 3 + 0.6 * 2 + 0.1 * 4)}{10} = 0.25$$

$$S_4 = \frac{(0.3 * 7 + 0.4 * 6 + 0.3 * 4)}{10} = 0.57$$

Маючи S для кожного хоста, додамо його до інших даних для прикладу.

На основі даних з таблиці 3.5, розрахуємо вразливість для кожного хоста:

$$V_1 = 0.5 * 0.22 + 0.3 * (1 - 0.9) + 0.2 * 0.2 = 0.18$$

$$V_2 = 0.5 * 0.26 + 0.3 * (1 - 0.8) + 0.2 * 0.2 = 0.23$$

$$V_3 = 0.5 * 0.25 + 0.3 * (1 - 0.7) + 0.2 * 0.3 = 0.275$$

$$V_4 = 0.5 * 0.57 + 0.3 * (1 - 0.6) + 0.2 * 0.1 = 0.425$$

Таблиця 3.5 – Вагові коефіцієнти та фактори безпеки мережі

№ хоста	Ваговий коефіці-єнт ω_k	Рівень вразливо-сті ПЗ S	Ваговий коефіці-єнт ω_k	Ефектив-ність політик безпеки P	Ваговий коефіці-єнт ω_k	Вплив людсько-го фактору U
1	0.5	0.22	0.3	0.9	0.2	0.2
2		0.26		0.8		0.2
3		0.25		0.7		0.3
4		0.57		0.6		0.1

Також розрахуємо матрицю ймовірності переходів між хостами G з даних для прикладу.

Таблиця 3.6 – Рівні відкритості з'єднання T:

Хост/хост	1	2	3	4
1	0	0.1	0.1	0.1
2	0.2	0	0.1	0.3
3	0.1	0.4	0	0.3
4	0.1	0.3	0.1	0

Таблиця 3.7 – Ефективності фаєрволів і фільтрації F

Хост/хост	1	2	3	4
1	1	0.6	0.9	0.5
2	0.1	1	0.7	0.9
3	0.1	0.7	1	0.3
4	0.4	0.6	0.8	1

Таблиця 3.8 – Рівні шифрування L

Хост/хост	1	2	3	4
1	1	0.8	0.6	0.8
2	0.7	1	0.7	0.7
3	0.9	0.8	1	0.9
4	0.9	0.7	0.8	1

Приймемо вагові коефіцієнти $\omega_T = 0.4$, $\omega_F = 0.3$, $\omega_L = 0.3$. За формулою $G_{ij} = \omega_T T_{ij} + \omega_F(1 - F_{ij}) + \omega_L(1 - L_{ij})$ розрахуємо значення G:

Таблиця 3.9 – Ймовірність переходу G

Хост/хост	1	2	3	4
1	0	0.22	0.19	0.25
2	0.44	0	0.22	0.24
3	0.34	0.31	0	0.36
4	0.25	0.33	0.16	0

Тепер, коли розраховані усі значення V та G, можна порахувати загальний рівень кібербезпеки у мережі із заданими характеристиками.

$$CS = \big((1 - V_1) * (1 - G_{21} * V_2) * (1 - G_{31} * V_3) * (1 - G_{41} * V_4)\big)$$
$$* \big((1 - V_2) * (1 - G_{12} * V_1) * (1 - G_{32} * V_3) * (1 - G_{42} * V_4)\big)$$
$$* \big((1 - V_3) * (1 - G_{13} * V_2) * (1 - G_{23} * V_2) * (1 - G_{43} * V_4)\big)$$
$$CS = \big((1 - 0.18)(1 - 0.22 * 0.44)(1 - 0.275 * 0.34)(1 - 0.425 * 0.25)\big)$$
$$* \big((1 - 0.22)(1 - 0.18 * 0.22)(1 - 0.275 * 0.31)(1 - 0.425 * 0.33)\big)$$
$$* \big((1 - 0.275)(1 - 0.18 * 0.19)(1 - 0.22 * 0.22)(1 - 0.425 * 0.16)\big)$$
$$\approx 0.2195$$

При даних, що використовувалися у прикладі, оцінка кібербезпеки виходить 0.2195. Це число означає ймовірність 0.2195 вистояти проти серйозних та/або частих кібератак, не скомпрометувавши жодного з важливих хостів.

Функція оцінювання кібербезпеки комп'ютерних станцій розроблена з урахуванням комплексного підходу до аналізу вразливостей та їхнього впливу на загальну стійкість мережевої інфраструктури. Враховуючи динамічний характер кіберзагроз та їхню еволюцію, цей підхід є не лише ефективним, а й необхідним для адекватного прогнозування можливих сценаріїв атак. Основна ідея методології полягає у створенні математичної моделі, яка б дозволяла оцінювати ризик компрометації окремих хостів та передбачати, яким чином атаки можуть поширюватися мережею, залежно від її архітектури та рівня захисту.

Актуальність такого підходу зумовлена тим, що у сучасних інформаційних системах рівень загроз постійно змінюється, зокрема через появу нових вразливостей, вдосконалення тактик атакувальників та еволюцію методів обходу існуючих заходів безпеки. Тому оцінка рівня кібербезпеки не може бути статичною величиною, а повинна відображати актуальний стан системи на конкретний момент часу. Це означає, що запропонована модель має враховувати не лише існуючі параметри безпеки, але й їхню динаміку, тобто можливість зміни рівня ризику унаслідок впровадження нових захисних заходів або появи нових загроз.

В основі методу лежить оцінювання вразливості кожного окремого вузла мережі, що здійснюється за допомогою формули, яка враховує три ключові складові: рівень вразливості програмного забезпечення, ефективність політик безпеки та ймовірність компрометації через людський фактор. Така структуризація була обрана з огляду на характер сучасних атак, які, як правило, спрямовані на один із цих аспектів або є комбінацією кількох. Важливо відзначити, що подібний підхід дозволяє не лише оцінити загальний рівень кіберзахисту, але й визначити, які саме чинники найбільше впливають на ймовірність успішного проникнення зловмисника. Це особливо важливо у контексті розподілу ресурсів на підвищення рівня безпеки, адже різні організації можуть мати різні пріоритети щодо того, які загрози для них є критичними.

Рівень вразливості програмного забезпечення (S) є одним із найважливіших параметрів, оскільки більшість атак використовує недоліки у коді програмних компонентів або некоректні налаштування системи. Саме тому його розрахунок базується на аналізі знайдених вразливостей та їхній критичності. У цій моделі використовується шкала CVSS (Common Vulnerability Scoring System), яка дозволяє оцінити серйозність кожної знайденої вразливості за десятибальною шкалою. Важливо враховувати, що не всі вразливості однаково впливають на безпеку системи, тому застосовуються вагові коефіцієнти, що диференціюють їхню значущість. Наприклад, якщо вразливість має активний експлойт у відкритому доступі, її вага буде вищою, ніж у випадку вразливості, для якої немає відомих способів експлуатації.

Таким чином, кінцеве значення S враховує не лише наявність слабких місць у програмному забезпеченні, а й їхню реальну загрозу для мережі.

Другим важливим показником є ефективність політик безпеки (P), яка визначає, наскільки добре налаштовані захисні механізми та процедури в організації. У межах запропонованої моделі цей параметр нормалізований у діапазоні від 0 до 1, де 1 відповідає найкращим можливим практикам у сфері безпеки, а 0 – повній відсутності захисних заходів. Визначення цього параметра вимагає участі експертів, оскільки оцінка має враховувати як технічні аспекти (наявність сучасних засобів захисту, регулярність оновлень), так і організаційні (політики управління доступом, наявність планів реагування на інциденти, контроль активності користувачів). Використання цього параметра дозволяє відобразити той факт, що навіть за наявності вразливостей їхня експлуатація може бути значно ускладнена при належній конфігурації захисних механізмів.

Останній параметр – ймовірність компрометації через людський фактор (U) – враховує вплив помилок користувачів на рівень безпеки системи. Дослідження показують, що значний відсоток атак стає успішним саме через дії користувачів, які можуть стати жертвами соціальної інженерії, використовувати слабкі паролі або несанкціоновано передавати конфіденційну інформацію. Для оцінки цього параметра використовується статистична інформація про попередні інциденти, кількість фішингових атак, які вдалося реалізувати, та рівень обізнаності персоналу щодо безпекових загроз. Як і попередні два параметри, U нормалізований у межах [0,1] і враховується при розрахунку загальної вразливості системи.

З урахуванням цих трьох складових визначається загальний рівень вразливості окремого вузла (V), однак для комплексного оцінювання рівня кібербезпеки необхідно враховувати імовірність поширення атаки мережею. Саме для цього у моделі введено функцію $G(i,j)$, що визначає ймовірність компрометації одного хоста за умови компрометації іншого. Вона враховує три основні параметри: рівень відкритості з’єднання (T), ефективність фільтрації трафіку (F), рівень шифрування (L). Вагові коефіцієнти у цій формулі дозволяють правильно визначити вплив

кожного з цих факторів на поширення атаки, що є важливим, оскільки деякі параметри можуть мати вирішальний вплив на безпеку мережі.

Нарешті, для визначення загального рівня кібербезпеки мережі (CS) використовується комбінація значень V та G, що дозволяє оцінити, наскільки захищена система від атак. Використання ймовірнісного підходу забезпечує гнучкість моделі та її адаптивність до різних мережевих конфігурацій. Отримане значення CS може бути використане не лише для оцінки поточного рівня безпеки, але й для прийняття рішень щодо необхідності впровадження додаткових заходів захисту. Таким чином, запропонована функція є ефективним інструментом для аналізу та прогнозування кіберзагроз, що робить її цінною у сфері управління інформаційною безпекою.

Функція оцінювання кібербезпеки комп'ютерних станцій ґрунтується на необхідності врахування множини параметрів, що впливають на рівень захищеності мережевої інфраструктури. Оцінка рівня безпеки не може базуватися лише на аналізі окремих компонент, оскільки інформаційні системи є складними взаємопов'язаними структурами, де ризик компрометації одного вузла може істотно підвищувати ймовірність атаки на інші. З огляду на це, застосування сукупності параметрів для оцінки безпеки є виправданим, адже воно дозволяє враховувати не тільки індивідуальні характеристики кожного хоста, а й загальну стійкість мережі до атак.

Підхід, що використовується у даній моделі, орієнтований на об'єктивну оцінку рівня безпеки, яка може бути отримана шляхом збору та обробки релевантних даних. Основні параметри, такі як рівень вразливості програмного забезпечення, ефективність політик безпеки та ймовірність компрометації через людський фактор, визначають ключові аспекти захищеності системи. Водночас, слід зазначити, що дана методологія має фундаментальну особливість: вона враховує взаємозв'язки між компонентами та дозволяє моделювати вплив одного вузла на інший, що є вкрай важливим при прогнозуванні можливих сценаріїв атак.

Однією з причин, чому саме така структура оцінювання була обрана, є емпіричний аналіз кіберінцидентів, який свідчить про те, що успішна компрометація інформаційної системи зазвичай є результатом поєднання кількох факторів, а не

ізольованого впливу однієї вразливості. Успішні атаки часто включають експлуатацію технічних вразливостей у поєднанні з соціальною інженерією та недостатньою ефективністю заходів безпеки. Таким чином, модель, що ґрунтується на трьох основних компонентах, дозволяє враховувати реальні загрози, з якими стикаються організації.

У межах запропонованого підходу критично важливим є правильне визначення вагових коефіцієнтів у кожній формулі. Використання коефіцієнтів дозволяє скоригувати вплив окремих факторів на кінцеве значення показника безпеки, що дає змогу зробити оцінку більш точною. Наприклад, якщо конкретна вразливість має відомий експлойт та активно використовується в атаках, то її вплив на показник вразливості системи буде значно більшим, ніж у випадку вразливості, яка ще не отримала широкого розповсюдження. Аналогічно, якщо організація використовує комплексні заходи безпеки, зокрема системи виявлення вторгнень, багатофакторну автентифікацію та автоматизовані системи аналізу поведінки користувачів, ефективність політик безпеки буде значно вищою, що зменшить загальний ризик компрометації.

Одним із найскладніших завдань у розробці таких моделей є коректний аналіз людського фактора, оскільки поведінкові аспекти важко формалізувати. Людська схильність до помилок, довірливість або недостатній рівень обізнаності щодо кіберзагроз можуть зробити навіть найбільш захищену систему вразливою. Тому оцінка цього параметра базується на історичних даних про попередні випадки успішних атак, проведених через соціальну інженерію, аналізі рівня навчання користувачів та частоті проходження ними тестувань на стійкість до фішингових атак. Наприклад, якщо компанія регулярно проводить тренінги з кібербезпеки, впроваджує навчальні програми та здійснює тестування персоналу щодо обізнаності про сучасні атаки, ймовірність компрометації через людський фактор буде нижчою.

Важливим аспектом аналізу мережевої безпеки є ймовірність розповсюдження загроз між хостами. У традиційних підходах оцінювання кібербезпеки часто розглядається кожен вузол окремо, що є хибним припущенням, адже у реальних мережах компрометація одного пристрою може суттєво підвищувати ймовірність

компрометації інших. Функція $G(i,j)$ визначає ризик зараження одного вузла від іншого. Формула враховує такі показники, як рівень відкритості з'єднання, ефективність засобів контролю трафіку, рівень шифрування та ефективність засобів виявлення аномалій. Сукупність цих факторів дозволяє зробити висновок про можливість подальшого поширення атаки у мережі.

Рівень відкритості з'єднання визначається як відношення дозволених з'єднань між вузлами до загальної кількості потенційних з'єднань у мережі. Чим вищий цей показник, тим більш імовірним є перехід атаки між хостами. У випадках, коли мережева архітектура передбачає сегментацію або використання ізольованих середовищ, рівень відкритості зменшується, що позитивно впливає на безпеку. Водночас ефективність фаєрволів і систем фільтрації трафіку відіграє вирішальну роль у блокуванні несанкціонованих з'єднань та мінімізації ризиків. Рівень шифрування переданих даних також є важливим показником: якщо комунікація між вузлами здійснюється через відкриті або слабко захищені канали, можливість перехоплення або підміни даних суттєво зростає. Нарешті, системи виявлення аномалій (IDS) дозволяють відстежувати підозрілу активність та блокувати її на ранніх стадіях, зменшуючи ризик поширення атаки.

Остаточна оцінка загального рівня кібербезпеки мережі (CS) отримується шляхом агрегування всіх попередньо розрахованих показників. Це значення є показником ймовірності того, що жоден із критичних хостів не буде скомпрометований у випадку здійснення кібератаки. Чим вище значення CS, тим вища загальна стійкість системи. Це значення може використовуватися як у стратегічному плануванні заходів кібербезпеки, так і в оперативному моніторингу стану захищеності.

Таким чином, запропонована методика дозволяє не лише оцінити поточний рівень кібербезпеки, але й виявити слабкі місця, спрогнозувати можливі сценарії розвитку атак і розробити оптимальні стратегії реагування. Використання математичного моделювання та ймовірнісного аналізу дає змогу значно підвищити точність оцінки стану безпеки мережі та підвищити ефективність захисних заходів. Це робить дану функцію оцінювання кібербезпеки універсальним інструментом для

аналізу захищеності комп’ютерних станцій у будь-яких типах організацій, від корпоративних середовищ до критичних державних інфраструктур.

3.3 Висновки до третього розділу

Визначено, що ефективність оцінювання кібербезпеки комп’ютерних станцій значною мірою визначається типами загроз, які на них спрямовані, зокрема комп’ютерними атаками та шкідливим програмним забезпеченням, а також здатністю системи точно вимірювати ступінь уразливості.

Було сформульовано функцію, яка узагальнює вплив технічних характеристик, поведінкових індикаторів та ризиків, пов’язаних із функціонуванням станцій, що дозволяє отримати кількісну оцінку рівня їх кіберзахищеності в умовах змінного загрозового середовища.

4 МЕТОД ОЦІНЮВАННЯ КІБЕРБЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ

4.1 Метод синтезу самоорганізованих систем оцінювання кібербезпеки комп’ютерних станцій

Метод синтезу самоорганізованих систем оцінювання кібербезпеки комп’ютерних станцій ґрунтується на побудові такої системи, яка у реальному часі забезпечує постійне спостереження за станом корпоративної мережі та окремих комп’ютерних станцій, здійснює збір відповідних метрик і проводить обчислення функції оцінювання кібербезпеки. Центральним елементом цієї системи є функція, що відображає поточний рівень захищеності інформаційної інфраструктури з урахуванням багатьох взаємозалежних чинників. Така функція має формуватись на основі агрегованих показників активності системних процесів, конфігураційної цілісності, стану мережевих з'єднань, а також ступеня вразливості, що випливає із відомих технічних характеристик програмного забезпечення та рівня застосування політик доступу.

Для розгортання системи оцінювання знадобиться початкова конфігурація коефіцієнтів та значень, що неможливо оцінити технічними методами з хорошою точністю.

У формулі 3.3 вагові коефіцієнти $\omega_S, \omega_P, \omega_U$, а також значення P та U за ідеальних обставин мають бути розраховані експертами з кібербезпеки у кожному конкретному випадку корпоративної мережі. Такий підхід передбачає індивідуальну настройку системи оцінювання з урахуванням специфіки архітектури, типів інформаційних активів, організаційної структури підприємства, а також можливих векторів атак, що притаманні певній галузі або регіону. Альтернативно, пропонується використовувати такі значення вагових коефіцієнтів:

Аналогічно, значення P та U також мають бути розраховані експертами з кібербезпеки (в ідеалі) на основі аудиту, що покаже відповідність політик безпеки мережі останнім стандартам та рівень обізнаності та коректності персоналу в користуванні комп’ютерами. Альтернативно, значення P можна приблизно визначити на основі таких компонентів: наявність документованих політик безпеки,

актуальність політик, контроль доступу, управління паролями, реакція на інциденти; значення U, відповідно на основі інших компонентів та історії: частота фішингових інцидентів за останній рік, рівень обізнаності персоналу (тести/опитування), наявність регулярного навчання, випадки втрати паролів/доступів, соціотехнічні симуляції (результати).

Таблиця 4.1 – Вагові коефіцієнти залежно від типу організації

Тип організації	ω_S	ω_P	ω_U
Техноцентрична організація	0.7	0.2	0.1
Установа з бюрократичною структурою	0.2	0.5	0.3
Компанія в умовах активного фішингу	0.3	0.2	0.5

Для знаходження решти значень формули – $\omega_k, N_e, CVSS_k$ – знадобиться спеціалізоване програмне забезпечення та інші ресурси. Для отримання $CVSS_k$ пропонується скористатися сканером вразливостей OpenVAS. Це безкоштовне ПЗ з відкритим кодом, що дозволяє бути впевненим у відсутності зловживання наданим до мережі доступом з боку розробника за умови регулярного перегляду змін у відкритому коді. Для роботи системи оцінювання кібербезпеки знадобиться регулярно запускати сканування комп'ютера на наявні вразливості, в результаті якого програма генерує звіт, значення CVSS з якого будуть використовуватися для розрахунків.

Для знаходження ω_k, N_e пропонується використовувати щоденно оновлювані дані від моделі Exploit Predition Scoring System (EPSS). Це система, яка оцінює ймовірність того, що певна вразливість буде експлуатуватися в реальному світі протягом найближчих 30 днів. Для отримання даних можна використовувати API або завантажувати звіти у форматі CSV. Кожним рядком файлу є трійка CVE (назва вразливості), EPSS (ймовірність експлуатації вразливості), Percentile (персентиль ймовірності для даної вразливості). В якості N_e використовуватимемо кількість вразливостей у звіті EPSS, а в якості ω_k – $EPSS_k$, нормалізоване таким чином, що

сума усіх дорівнює одиниці. Таким чином, вага вразливості буде пропорційна ймовірності цю вразливість зустріти.

Для формули 3.4 вагові коефіцієнти $\omega_T, \omega_F, \omega_L$ мають задаватися CISO (chief information security officer) або аналітиком. Залежно від типу мережі вони можуть сильно відрізнятися. Наприклад, у хмарному середовищі, де багато відкритих портів, але сильне шифрування – більше ваги для ω_T, менше для ω_L. Це може бути реалізовано як таблиця профілів ризику:

Таблиця 4.2 – Вагові коефіцієнти залежно від типу мережі

Тип мережі	ω_T	ω_F	ω_L
Хмарна інфраструктура	0.1	0.6	0.3
Корпоративна локальна мережа	0.1	0.5	0.4
Мінімальний контроль доступу	0.3	0.5	0.2

Також слід визначити T_{ij}, F_{ij}, L_{ij}. Розрахуємо T_{ij}:

$$T_{ij} = \frac{N_o}{N_a}, \tag{4.1}$$

де N_o – кількість відкритих портів, крім стандартних зашифрованих (наприклад, HTTPS), N_a – максимальна допустима кількість відкритих портів, типово – 10.

Розрахуємо F_{ij}. Для цього використовуєтєься періодичне активне тестування – генеруються запити, що імітують шкідливий трафік. Для генерації пропонується використовувати утиліту з відкритим кодом hping. Формула:

$$F_{ij} = \frac{N_{failed}}{N_{tests}}, \tag{4.2}$$

Де N_{failed} – кількість пропущених шкідливих запитів, згенерованих при тестуванні, та N_{tests} – загальна кількість тестів.

Розрахуємо L_{ij}. Пропонується використати програму SSLyze для сканування зв'язків мережі для перевірки надійності шифрування. За результатами сканування можна оцінити чисельне значення для використання у формулі 3.4. Оскільки на даний момент найнадійнішим шифруванням на транспортному рівні вважається протокол TLS 1.3, його оцінимо $L_{ij} = 1$, а SSL, як застарілий метод із давно відомими вразливостями, оцінимо як $L_{ij} = 0$. Для проміжних значень використаємо $L_{ij} = 0.7$ для TLS 1.2 та $L_{ij} = 0.3$ для TLS 1.1.

Отже, для розрахування значення функції кібербезпеки корпоративної мережі знадобляться різні програми та ресурси. Серед них є система OpenVAS – потужна відкрита платформа для сканування вразливостей, яка забезпечує автоматизований аналіз систем безпеки шляхом проведення активного тестування мережевих служб, операційних систем і програмного забезпечення на наявність відомих вразливостей. OpenVAS надає користувачеві розгорнуту інформацію щодо знайдених недоліків, а також рекомендації стосовно їх усунення. Ця робота передбачає використання OpenVAS для знаходження значень CVSS на кожній з комп’ютерних станцій. CVSS на основі багатьох факторів оцінює тяжкість вразливостей, наявних на комп’ютері, дає їм чисельне значення, яке можна використовувати для розрахунків.

Також робота передбачає використання Exploit Prediction Scoring System – системи, що дозволяє отримати дані для оцінювання вагомості кожної з вразливостей. Її головною метою є визначення ймовірності того, що конкретна вразливість буде використана зловмисниками у реальних умовах протягом найближчого часу, що значно підвищує точність оцінки ризиків у системі. Оскільки наявність вразливості не означає безумовно про те, що зловмисник спробує її експлуатувати, нижча ймовірність спроби вказує на більший шанс того, що корпоративна мережа вистоїть проти атаки. Завдяки цим оцінкам можна визначити пріоритети для усунення чи моніторингу знайдених вразливостей, зосереджуючи зусилля на тих, що з більшою ймовірністю можуть бути використані у шкідливих цілях. Таким чином, використання EPSS дозволяє не лише кількісно оцінити ступінь ризику, але й оптимізувати процес прийняття рішень у сфері кібербезпеки.

Інша програма, яка знадобиться для коректного оцінювання кібербезпеки – hping. Ця програма є однією з тих, що розширюють функціонал класичної утиліти ping, однак має значно ширші можливості. Hping дозволяє використовувати різні мережеві протоколи, зокрема TCP, UDP, ICMP та RAW-IP, що робить її універсальним інструментом для створення спеціалізованих мережевих пакетів. Завдяки високому рівню конфігурованості, вона може застосовуватися для широкого спектру задач у сфері кібербезпеки, включаючи зондування мережі, аудит безпеки, перевірку правил фаєрволу, симуляцію шкідливого трафіку, аналіз відповіді на нестандартні пакети, а також виявлення слабких місць у фільтрації. У контексті оцінювання кібербезпеки корпоративної мережі hping використовується для моделювання атак з метою перевірки ефективності фільтраційного обладнання та програмного забезпечення. Вона дає змогу на практиці протестувати, наскільки добре система здатна розпізнавати та блокувати небажаний або потенційно небезпечний трафік, що дозволяє зробити об’єктивні висновки щодо рівня захищеності мережевої інфраструктури.

Ще одна програма, яка необхідна для розрахування значення функції кібербезпеки – SSLyze. Вона є потужним інструментом для аналізу безпечності TLS-з'єднань, що дозволяє глибоко перевірити конфігурацію комп’ютерної станції після під’єднання до неї. SSLyze виконує сканування з метою перевірки якості налаштування протоколів шифрування, аналізує встановлені сертифікати, підтримувані набори шифрів, використовувані еліптичні криві, а також здатність системи протистояти відомим атакам на TLS-протоколи, таким як Heartbleed, ROBOT, DROWN та інші. Програма здатна виявити застарілі або небезпечні конфігурації, які можуть поставити під загрозу захищеність даних під час передавання в мережі. В контексті даної роботи таке сканування є надзвичайно важливим для перевірки безпеки з’єднання між вузлами корпоративної мережі, особливо якщо передача чутливої інформації відбувається через зовнішні або відкриті канали зв’язку. Використання SSLyze дозволяє своєчасно виявити слабкі місця у шифруванні, оновити конфігурації відповідно до актуальних стандартів та знизити

ризики перехоплення або підміни даних, що є критично важливим для забезпечення цілісності та конфіденційності інформаційної інфраструктури.

Усі запропоновані програми мають відкритий код, що дозволяє не витрачати додаткові кошти на придбання програмного забезпечення для використання у системі оцінювання, що особливо важливо для організацій з обмеженим бюджетом. Відкритий код означає, що програмне забезпечення є прозорим для користувача – кожен може ознайомитися зі структурою, алгоритмами та логікою роботи програми, переконатися у відсутності шкідливих або прихованих функцій, що можуть ставити під загрозу безпеку або конфіденційність корпоративної мережі.

Це створює додатковий рівень довіри, особливо у сфері кібербезпеки, де навіть невеликий прихований компонент може бути використаний для атаки або витоку даних.

Крім того, спільнота розробників постійно підтримує та оновлює такі інструменти, що дозволяє швидко реагувати на нові загрози та вразливості. Таким чином, використання програм з відкритим кодом не лише знижує фінансові витрати, а й підвищує загальний рівень надійності та контролю за безпекою інструментів, які інтегруються в систему оцінювання кіберзахисту.

Отже, для побудови самостійної системи оцінювання кібербезпеки необхідно встановити вищеописане програмне забезпечення та дозволити йому збирати дані про комп’ютери мережі. Результатом тривіальної обробки звітів, отриманих від ПЗ на основі зібраних даних є чисельні значення, на основі яких можна розрахувати значення функції, описаної формулою 3.5.

Деякі значення, як вагові коефіцієнти чи ефективність політик кібербезпеки, не можуть бути розраховані програматично, і вони мають бути сконфігуровані на етапі розгортання мережі експертами кібербезпеки, наприклад, CISO або аналітиками. В залежності від зовнішніх даних про мережу в цілому можна використати дані з таблиць, наведених у цьому розділі.

4.2 Дослідження ефективності методу оцінювання кібербезпеки корпоративних мереж

Для оцінки ефективності запропонованої моделі був проведени експеримент, що моделює роботу реалізованої системи оцінки кібербезпеки в умовах, наближених до реального середовища. Тестування охоплювало симуляцію активності мережевих вузлів протягом одного тижня з дискретністю в одну годину. У ході експерименту було реалізовано динамічне оновлення вхідних параметрів, що впливають на рівень вразливості окремих комп'ютерів та ймовірність їх компрометації в результаті взаємодії з іншими вузлами мережі.

Компоненти моделі, що формують функції вразливості та ймовірності компрометації, були налаштовані вручну на основі припущень щодо типових характеристик організаційного IT-середовища. Зокрема, вагові коефіцієнти для складових технічної, політичної та людської вразливості були встановлені у відповідності до умовно важливих пріоритетів безпеки. Аналогічно, ваги параметрів трафіку, фільтрації, шифрування та мережевої віддаленості були обрані так, щоб відобразити характерні ризики проникнення у мережу через взаємодію між окремими комп'ютерами. Значення сконфігурованих вручну параметрів: $\omega_S = 0.7$, $\omega_P = 0.2$, $\omega_U = 0.1$, $\omega_T = 0.1$, $\omega_F = 0.4$, $\omega_L = 0.3$, $P = 0.9$, $U = 0.1$.

Для окремих вузлів були вручну змодельовані поодинокі пікові події підвищеної вразливості, що відповідають потенційним інцидентам, які могли мати місце в окремі години тижня. Це дозволило оцінити реакцію моделі на локальні зміни ризику та проаналізувати вплив таких подій на загальний рівень кібербезпеки. В якості важливих вузлів були визначені комп'ютери №1, №2 та №3, для яких побудовано окремі графіки зміни рівня вразливості у часі, а також обчислено агрегований показник рівня безпеки з урахуванням взаємного впливу через мережеві зв'язки. Усі параметри були згенеровані з урахуванням варіативності, а максимальні значення деяких показників були нормалізовані для забезпечення узгодженості графічного представлення результатів.

Графіки зміни рівня вразливості V для окремих комп'ютерів демонструють динаміку накопичення ризиків, що формуються під впливом технічних, організаційних та людських чинників. Базові значення вразливості для кожного ПК були змодельовані із застосуванням нормального розподілу в межах допустимого інтервалу, що відповідає стабільній роботі системи за умов відсутності зовнішніх чи внутрішніх загроз.

У рамках експерименту для важливих вузлів – ПК №1, №2 та №3 – було вручну додано поодинокі пікові відхилення, які моделювали епізодичне зростання рівня ризику. Ці піки були реалізовані шляхом штучного додавання помітної кількості технічних вразливостей з високими оцінками, що еквівалентно ситуації, коли на певному хості виявляється новий набір критичних вразливостей, наприклад, через непроведене оновлення або свіжовиявлені уразливості в програмному забезпеченні. Результатом цього стали короткочасні, але різкі зростання показника V, які чітко відображаються на графіках на рисунках 4.1–4.5 .

Рисунок 4.1 – Рівень вразливості V ПК №1

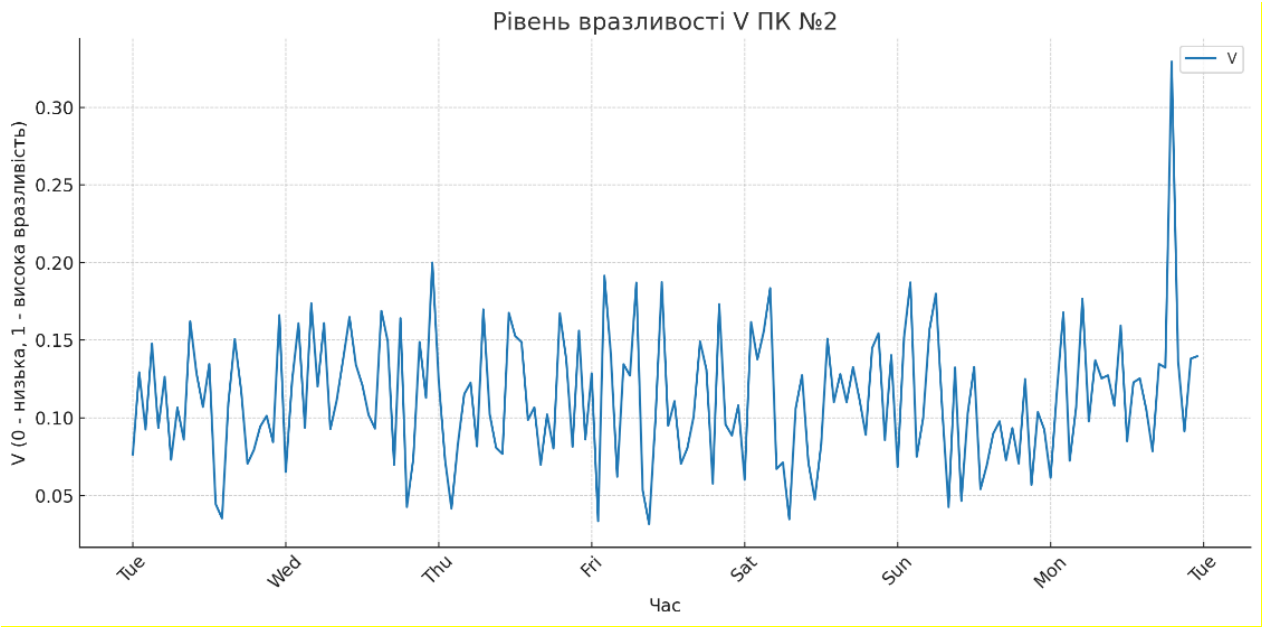


Рисунок 4.2 – Рівень вразливості V ПК №2

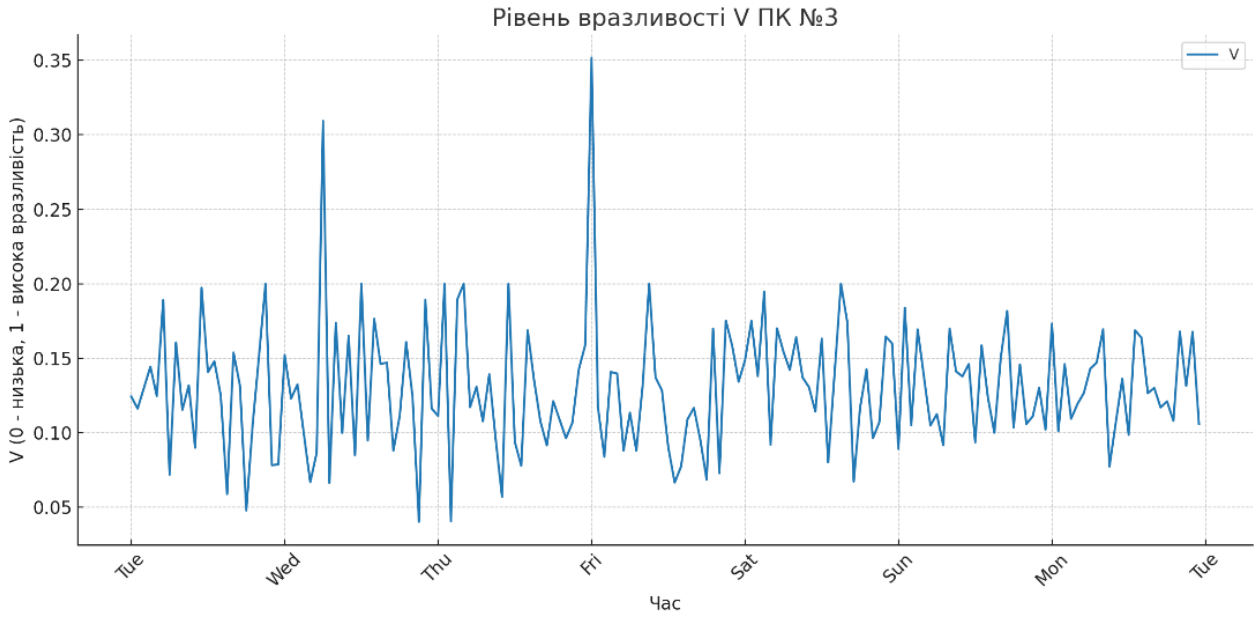


Рисунок 4.3 – Рівень вразливості V ПК №3

Графіки ймовірності компрометації G_{ij} відображають зміну ризиків, пов'язаних із взаємодією між окремими вузлами мережі, з урахуванням низки параметрів, що впливають на можливість розповсюдження атаки. При обчисленні G_{ij} було враховано інтенсивність трафіку між вузлами, рівень фільтрації, наявність або відсутність шифрування, а також логічну або фізичну віддаленість хостів один від одного. Кожен

із цих чинників має власну вагу у загальній формулі, що дозволяє гнучко відображати характер загроз у різних сценаріях.

Рисунок 4.4 – Рівень вразливості V ПК №4

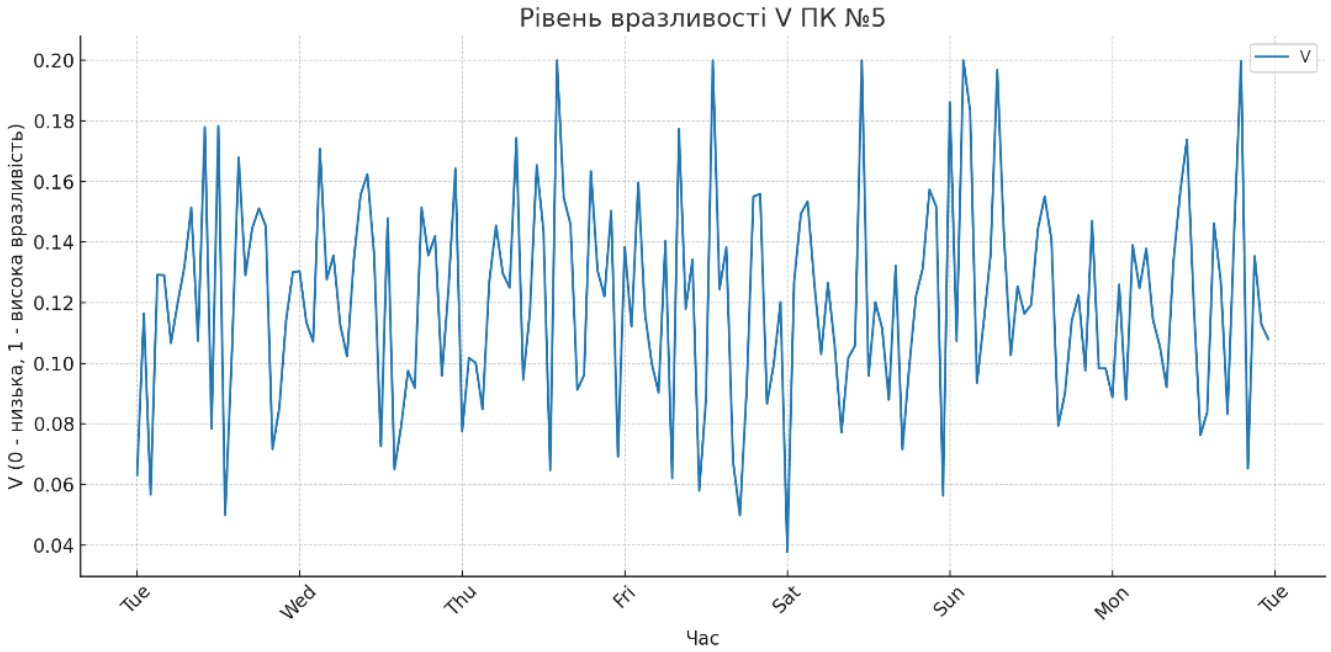


Рисунок 4.5 – Рівень вразливості V ПК №5

Значення ймовірності G_{ij} були отримані на основі даних, зібраних за допомогою спеціалізованого програмного забезпечення, описаного в параграфі 4.1,

яке дозволяє моделювати міжхостові зв’язки та параметри взаємодії в контрольованому середовищі. Це забезпечило можливість оцінки потенційних каналів компрометації як у напрямку від важливих ПК до інших елементів мережі, так і у зворотному напрямку.

Аналіз графіків на рисунках 4.6–4.10 показує, що в залежності від топології мережі та налаштувань безпеки інтенсивність ризику компрометації може суттєво відрізнятися навіть для хостів з подібним рівнем власної вразливості. Це підтверджує доцільність включення G_{ij} як обов’язкового компонента в загальну модель оцінки кіберризиків, оскільки врахування лише внутрішнього стану вузла не дозволяє повною мірою оцінити загрозу, що виникає внаслідок мережевої взаємодії.

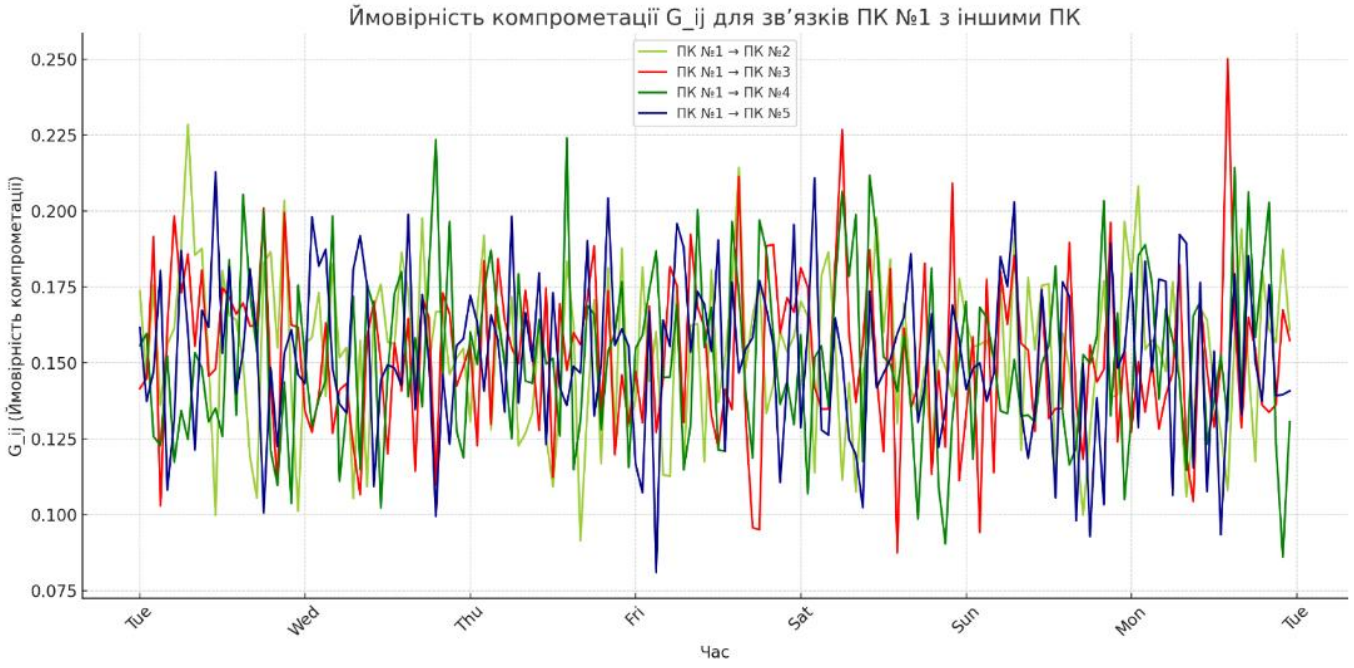


Рисунок 4.6 – Ймовірність компрометації G_{ij} ПК №1

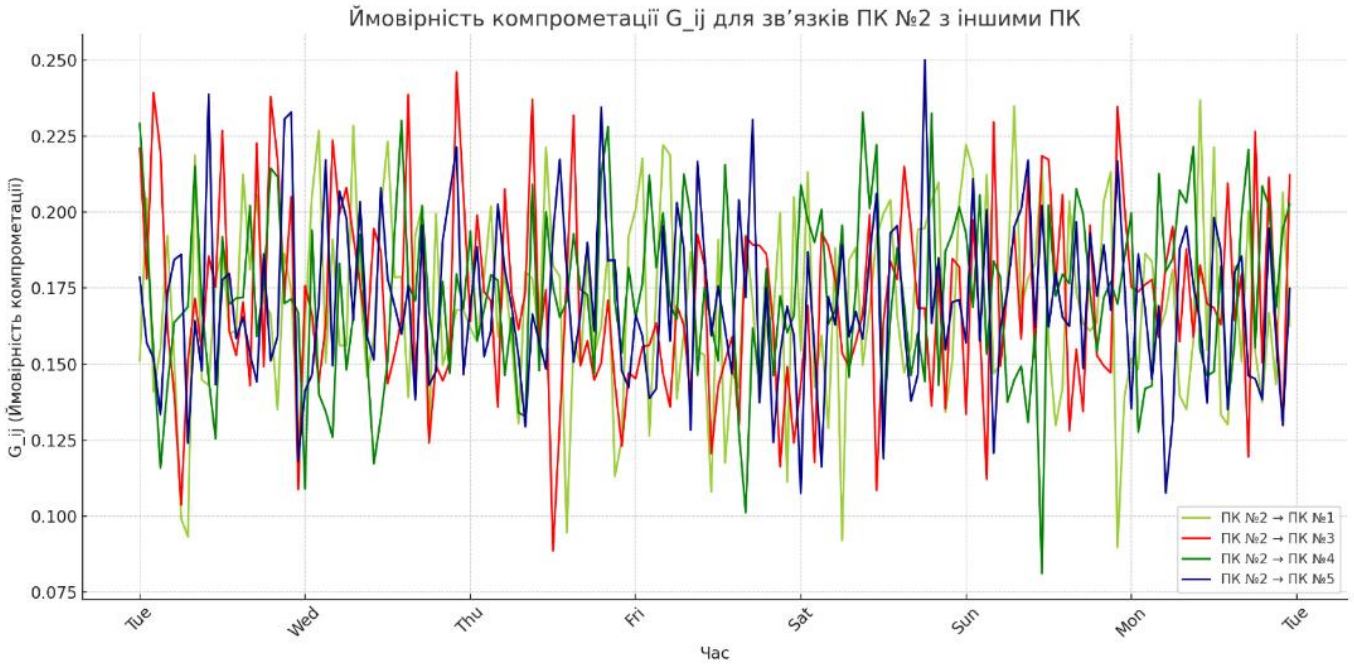


Рисунок 4.7 – Ймовірність компрометації G_{ij} ПК №2

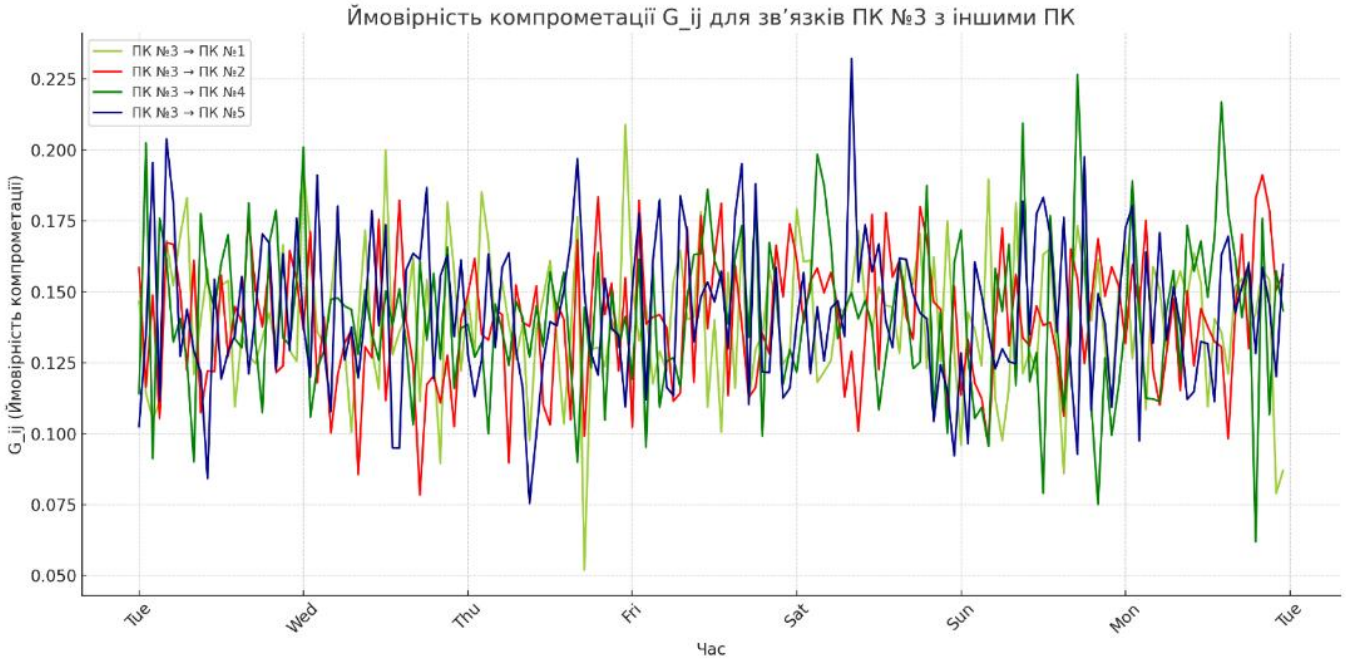


Рисунок 4.8 – Ймовірність компрометації G_{ij} ПК №3

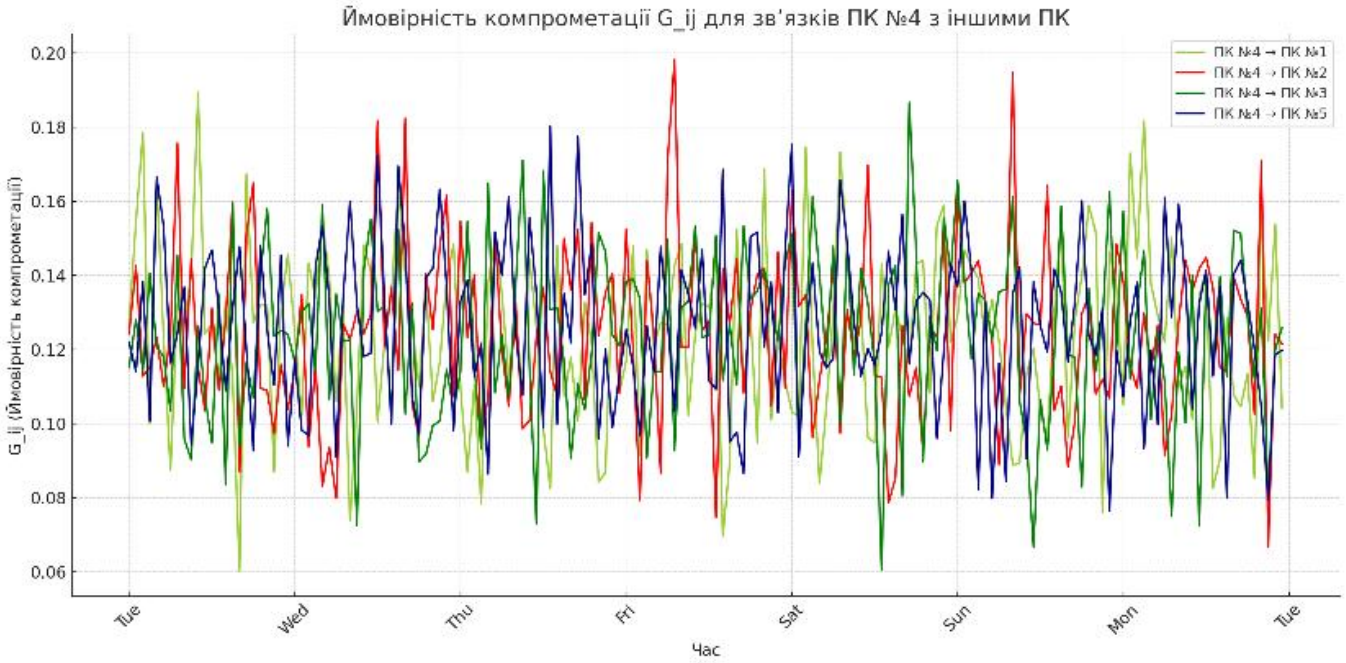


Рисунок 4.9 – Ймовірність компрометації G_{ij} ПК №4

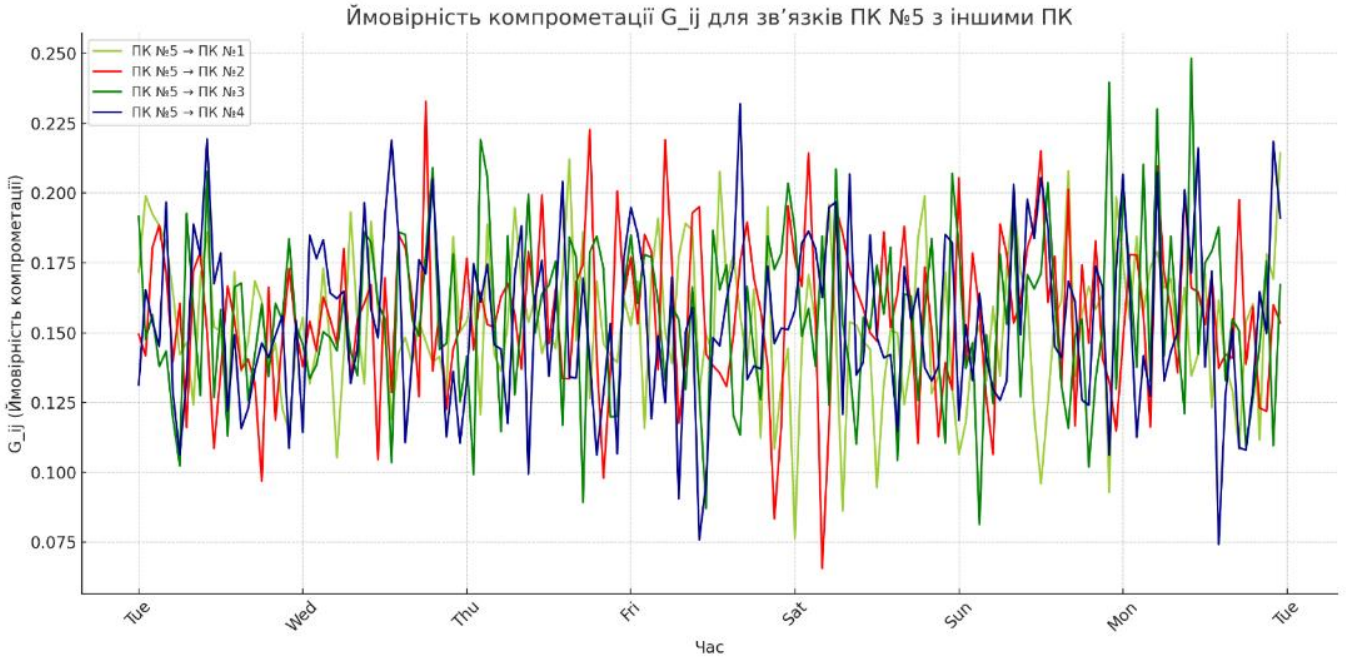


Рисунок 4.10 – Ймовірність компрометації G_{ij} ПК №5

Графік загального рівня кібербезпеки CS є ключовим аналітичним інструментом, який дозволяє комплексно оцінити безпекову ситуацію в мережі, враховуючи як локальні характеристики окремих вузлів, так і вплив міжвузлової взаємодії. Побудова цього показника базується на інтеграції оцінки вразливості важливих комп'ютерів із ймовірностями їх компрометації з боку інших елементів

системи. Такий підхід забезпечує багатовимірне бачення ризиків, що дозволяє отримати не лише ізольовану оцінку стану окремих хостів, а й відстежити системні залежності та потенційні ланцюги атак.

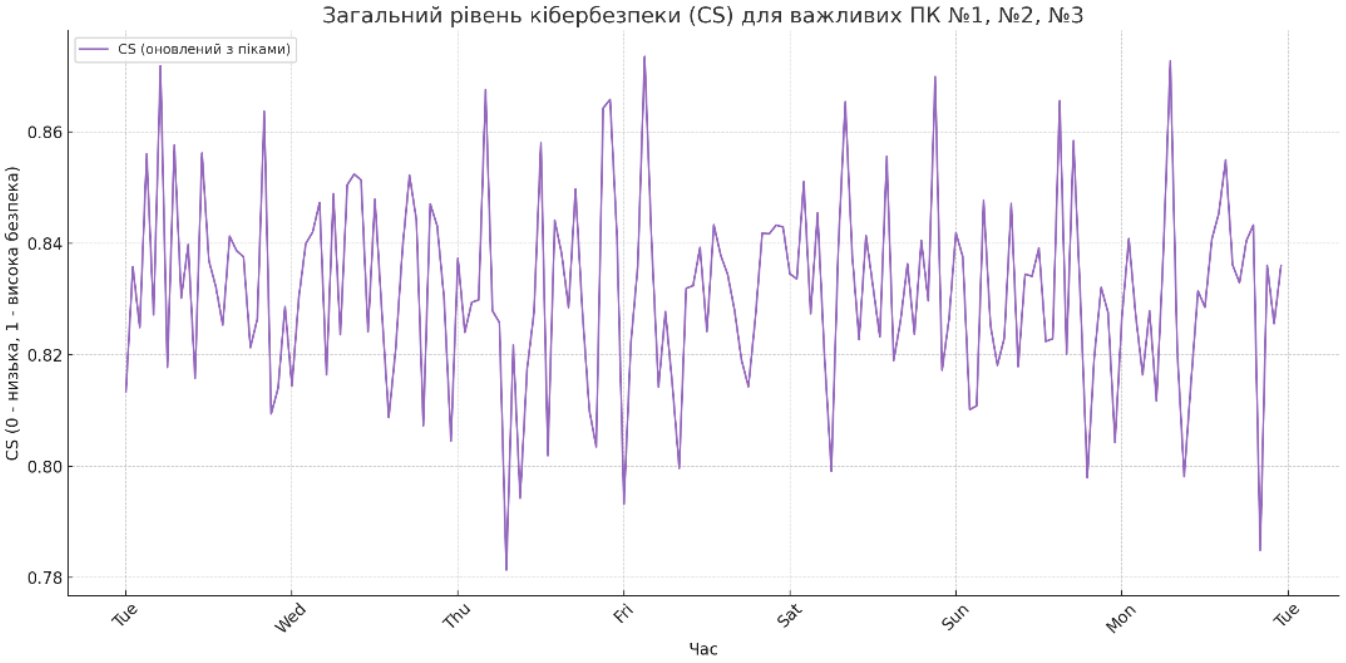


Рисунок 4.11 – Загальний рівень кібербезпеки CS

Особливу цінність графік графік на рисунку 4.11 має з точки зору оперативного моніторингу: він дає змогу ідентифікувати критичні часові ділянки, у яких фіксується різке зниження рівня безпеки, та прив'язати ці зміни до конкретних хостів із підвищеним рівнем вразливості або зростаючою загрозою компрометації. У сукупності з графіками V_i та G_{ij}, які деталізують джерела цих змін, графік CS дозволяє оператору миттєво оцінити загальну ситуацію в мережі, локалізувати проблемні точки та своєчасно вжити заходів для усунення вразливостей або зменшення ризиків поширення атаки.

Таким чином, візуалізація CS виступає ефективним механізмом прийняття рішень у реальному часі, що особливо важливо в умовах динамічного змінення загрозового ландшафту. Її інтеграція в систему управління безпекою дозволяє значно підвищити швидкість реагування та обґрунтованість дій з боку адміністратора або автоматизованих систем захисту.

4.3 Висновки до четвертого розділу

Розроблено метод оцінювання кібербезпеки комп'ютерних мереж. Він враховує параметри з’єднання та контекст виявлених вразливостей та людський фактор. Метод використовує оціночну функцію, побудовану на основі даних сканування, аналітичних показників ризику та інтеграції з CVSS та EPSS.

Результати експериментів показують ефективну та чутливу роботу системи оцінювання.

ВИСНОВКИ

У роботі розроблено новий метод оцінювання кібербезпеки у корпоративних мережах на основі використання оціночної функції. Отримані результати:

– здійснено аналіз існуючих методів і засобів захисту корпоративних мереж і розподілених систем, а також підходів до оцінювання рівня їх кібербезпеки;

– запропонувано метод оцінювання кібербезпеки корпоративних мереж;

– сформулювано цільову функцію, що характеризує рівень кібербезпеки корпоративної мережі;

– реалізувано прототип системи та проведено дослідження ефективності запропонованого методу оцінювання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Jain N. Distributed Systems: Basics. Medium. 2024. URL: https://medium.com/@naveenjain213.nj/distributed-systems-basics-78292e1e437e (дата звернення: 21.01.2025).

2. Uddin S., Bazgir E., Tanha T.T., Bhuiyan A.A., Haque E. Analysis of Distributed Systems. *SSRN*. 2024. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4976977 (дата звернення: 21.01.2025).

3. Ghosh R. K., Ghosh H. Distributed Systems: Theory and Applications. Hoboken, New Jersey. 2023. 532 p.

4. Xia Y., Sur D., Pingle A. S., Deshmukh J. V., Raghothaman M., Ravi S. Discovering Likely Invariants for Distributed Systems Through Runtime Monitoring and Learning. *Lecture Notes in Computer Science, vol 15529. Springer, Cham*. 2025. https://doi.org/10.1007/978-3-031-82700-6_1.

5. Pan X., Luo Z., Zhou L. Navigating the Landscape of Distributed File Systems: Architectures, Implementations, and Considerations. *Innovations in Applied Engineering and Technology*. 2023. Vol. 2, Issue 1, P. 1–12. https://doi.org/10.62836/iaet.v2i1.157.

6. Chippagiri S., Kassety N. Beyond the Monolith: Comprehensive Strategies for Architecting, Scaling, and Sustaining Resilient Distributed Systems. *International Journal of Research in Computer Applications and Information Technology (IJRCAIT)*. 2025. Vol. 8, Issue 1, P. 152–168. https://doi.org/10.34218/IJRCAIT_08_01_016.

7. Sarawagi S. Distributed Data Processing 101 – A Deep Dive. scaleyourapp.com. 2025. URL: https://scaleyourapp.com/distributed-data-processing-101-the-only-guide-youll-ever-need/ (дата звернення: 21.01.2025).

8. Kröher C., Gerling L., Schmid K. Combining Distributed and Central Control for Self-Adaptive Systems of Systems. *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Bologna, Italy. 2022. P. 109-112. https://doi.org/10.1109/ICDCSW56584.2022.00030.

9. Balreira D. G., Silva Araújo T., Petrillo F. Visualizing Kubernetes Distributed

Systems: An Exploratory Study. *2023 IEEE Working Conference on Software Visualization (VISSOFT)*, Bogotá, Colombia. 2023. P. 12-22. https://doi.org/10.1109/VISSOFT60811.2023.00011.

10. Ganguly R., Momtaz A., Bonakdarpour B. Runtime verification of partially-synchronous distributed system. *Form Methods Syst Des.* 2024. Vol. 64, P. 146–177. https://doi.org/10.1007/s10703-024-00450-5.

11. Ratana H., Syed-Mohamad S., Yong C. Software Model Checking Distributed Applications: A Hybrid Approach. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, Vol. 45(2), P. 152–167. https://doi.org/10.37934/araset.45.2.152167.

12. Govindaraaj J. The Convergence of Distributed Systems and AI-Driven Algorithms in Shaping Scalable and Resilient Computing Ecosystems. *International Journal of Information Technology Researchand Development (IJITRD)*. 2025. Vol. 6(2), P. 1-6. URL: https://www.researchgate.net/profile/Researcher-Iii/publication/389563040_The_Convergence_of_Distributed_Systems_and_AI-Driven_Algorithms_in_Shaping_Scalable_and_Resilient_Computing_Ecosystems/links/67c81595461fb56424f13189/The-Convergence-of-Distributed-Systems-and-AI-Driven-Algorithms-in-Shaping-Scalable-and-Resilient-Computing-Ecosystems.pdf (дата звернення: 21.01.2025).

13. Chandra P. Reliability-Driven Architecture Design for Distributed Systems: Key Principles and Practical Approaches. *International Journal of Research in Computer Applications and Information Technology (IJRCAIT).* 2025. Vol. 8(1), P. 2583-2597. https://doi.org/10.34218/IJRCAIT_08_01_187.

14. Lee E.A., Bateni S., Lin S., Lohstroh M., Menard C. Trading Off Consistency and Availability in Tiered Heterogeneous Distributed Systems. *Intelligent Computing*. 2023. Vol. 2. https://doi.org/10.34133/icomputing.0013.

15. Peng P., Soljanin E., Whiting P. Diversity/Parallelism Trade-Off in Distributed Systems With Redundancy. *IEEE Transactions on Information Theory*. 2022. Vol. 68(2), P. 1279-1295. https://doi.org/10.1109/TIT.2021.3127920.

16. Van Dame K.R., Bergmann T.B., Aichouri M., Pantoja M. A Comparative

Study of Consensus Algorithms for Distributed Systems. *Communications in Computer and Information Science*. 2022. https://doi.org/10.1007/978-3-031-04209-6_9.

17. Delporte-Gallet C., Fatourou P., Fauconnier H. When is recoverable consensus harder than consensus?. *Distrib. Comput.* 2025. https://doi.org/10.1007/s00446-025-00476-w.

18. Zhang Y., Yang J., Jin Z., Sethi U., Rodrigues K., Lu S., Yuan D. Understanding and Detecting Software Upgrade Failures in Distributed Systems. *SOSP '21: Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles.* 2021. P. 116–131. https://doi.org/10.1145/3477132.3483577.

19. Censor-Hillel K., Cohen S., Gelles R. Distributed computations in fully-defective networks. *Distrib. Comput.* 2023. Vol. 36, P. 501–528. https://doi.org/10.1007/s00446-023-00452-2.

20. Michail O., Skretas G., Spirakis P.G. Distributed computation and reconfiguration in actively dynamic networks. *Distrib. Comput.* 2022. Vol. 35, P. 185–206. https://doi.org/10.1007/s00446-021-00415-5.

21. Brânzan L. Formalization of distributed systems with semantic interoperability. *Technical Scientific Conference of Undergraduate, Master and PhD Students, Universitatea Tehnică a Moldovei.* 2024. Vol. 2, P. 898–906. URL: http://repository.utm.md/handle/5014/28194 (дата звернення: 21.01.2025).

22. HamaAli K.W., Zeebaree S.R.M. Resources Allocation for Distributed Systems: A Review. *International Journal of Science and Business, IJSAB International.* 2021. Vol. 5(2), P. 76–88. URL: https://ideas.repec.org/a/aif/journl/v5y2021i2p76-88.html (дата звернення: 21.01.2025).

23. Abdi A., Zeebaree S.R.M. Embracing Distributed Systems for Efficient Cloud Resource Management: A Review of Techniques and Methodologies. *Indonesian Journal of Computer Science.* 2024. Vol. 13(2), P. 1912–1933. URL: https://www.researchgate.net/publication/380577026_Embracing_Distributed_Systems_for_Efficient_Cloud_Resource_Management_A_Review_of_Techniques_and_Methodologies (дата звернення: 21.01.2025).

24. Mushtaq S.U., Sheikh S., Idrees S.M. Enhanced priority based task scheduling

with integrated fault tolerance in distributed systems. *International Journal of Cognitive Computing in Engineering*. 2025. Vol. 6, P. 152–169. https://doi.org/10.1016/j.ijcce.2024.12.006.

25. Habibpour Roudsari M.N. Improved task scheduling in heterogeneous distributed systems using intelligent greedy harris hawk optimization algorithm. *Evol. Intel.* 2024. Vol. 17, 4199–4226. https://doi.org/10.1007/s12065-024-00979-8.

26. Chakraborty T. Resource Sharing in Secure Distributed Systems. *ProQuest Dissertations & Theses, Mississippi State University*. 2024. URL: https://www.proquest.com/openview/c170eaf32f0228579712e7d98c761085/1?cbl=18750&diss=y&pq-origsite=gscholar (дата звернення: 21.01.2025).

27. Dhawan D., Ahmad F., Tripathi M.M. A System Model of Fault Tolerance Technique in the Distributed and Scalable System: A Review. *International Journal of Innovative Research in Computer Science & Technology*. 2022. https://doi.org/10.55524/ijircst.2022.10.1.14.

28. Malhotra S., Yashu F., Saqib M., Mehta D., Jangid J., Dixit S. Evaluating Fault Tolerance and Scalability in Distributed File Systems: A Case Study of GFS, HDFS, and MinIO. *arXiv*. 2025. https://doi.org/10.48550/arXiv.2502.01981.

29. Haroon M., Siddiqui Z.A., Husain M., Ali A., Ahmad T. A Proactive Approach to Fault Tolerance Using Predictive Machine Learning Models in Distributed Systems. *International Journal of Experimental Research and Review*. 2024. Vol. 44. https://doi.org/10.52756/ijerr.2024.v44spl.018.

30. Talaver O.V., Vakaliuk T.A. Reliable distributed systems: review of modern approaches. *Journal of Edge Computing*. 2023. Vol. 2(1). https://doi.org/10.55056/jec.586

31. Talaver O.V., Vakaliuk T.A. Telemetry to solve dynamic analysis of a distributed system. *Journal of Edge Computing*. 2024. Vol. 3(1). https://doi.org/10.55056/jec.728.

32. Nykänen P. Error monitoring in a distributed system. University of Jyväskylä. 2024. URL: https://urn.fi/URN:NBN:fi:jyu-202405153637 (дата звернення: 21.01.2025).

33. Zhao Z., Zeng Y., Wang J., Li H., Zhu H., Sun L. Detection and Incentive: A Tampering Detection Mechanism for Object Detection in Edge Computing. *2022 41st*

International Symposium on Reliable Distributed Systems (SRDS), Vienna, Austria. 2022. P. 166–177. https://doi.org/10.1109/SRDS55811.2022.00024.

34. Jghef Y.S., Zeebaree S.R.M., Ageed Z.S., Shukur H.M. Performance Measurement of Distributed Systems via Single-Host Parallel Requesting using (Single, Multi and Pool) Threads. *2022 3rd Information Technology To Enhance e-Learning and Other Application (IT-ELA), Baghdad, Iraq.* 2022. P. 38–43. https://doi.org/10.1109/IT-ELA57378.2022.10107923.

35. Wang Y. Network Monitoring and Performance Optimization of Large-scale Distributed System Based on Big Data Algorithm. *2024 Asia-Pacific Conference on Software Engineering, Social Network Analysis and Intelligent Computing (SSAIC), New Delhi, India*. 2024. P. 281–287. https://doi.org/10.1109/SSAIC61213.2024.00059.

36. Shah M., Hazarika A.V. An In-Depth Analysis of Modern Caching Strategies in Distributed Systems: Implementation Patterns and Performance Implications. *International Journal of Science and Engineering Applications.* 2025. Vol. 14(1), P. 9–13. https://doi.org/10.7753/IJSEA1401.1003.

37. Mahida A. Enhancing Observability in Distributed Systems – A Comprehensive Review. *Journal of Mathematical & Computer Applications.* 2023. Vol. 2(3), P. 1–4. https://doi.org/10.47363/JMCA/2023(2)135.

38. Aminizadeh S., Heidari A., Dehghan M., Toumaj S., Rezaei M., Navimipour N.J., Stroppa F., Unal M. Opportunities and challenges of artificial intelligence and distributed systems to improve the quality of healthcare service. *Artificial Intelligence in Medicine.* 2024. Vol. 149. https://doi.org/10.1016/j.artmed.2024.102779.

39. Zangana H.M., Zeebaree S.R.M. Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services. *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM).* 2024. Vol. 5(1), P. 11–30. https://doi.org/10.34010/injiiscom.v5i1.11883.

40. Orynbekova K., Kadyrov S., Bogdanchikov A., Oktamov S. A novel recommender system for adapting single machine problems to distributed systems within MapReduce. *Bulletin of Electrical Engineering and Informatics (BEEI).* 2024. Vol. 14(1). https://doi.org/10.11591/eei.v14i1.8370.

41. Zaman R., Jain T., Samara G., Jamali D. Corporate Governance Meets Corporate Social Responsibility: Mapping the Interface. *Business & Society*. 2020. Vol. 61(3). https://doi.org/10.1177/0007650320973415.

42. Rosário A.T., Raimundo R. Internet of Things and Distributed Computing Systems in Business Models. *Future Internet.* 2024. Vol. 16(10), P. 384. https://doi.org/10.3390/fi16100384.

43. Zhou S., Yuan B., Xu K., Zhang M., Zheng W. The Impact of Pricing Schemes on Cloud Computing and Distributed Systems. *Journal of Knowledge Learning and Science Technology*. 2024. Vol. 3(3), P. 193–205. https://doi.org/10.60087/jklst.v3.n3.p206-224.

44. 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. https://doi.org/10.1109/IDAACS58523.2023.10348773.

45. 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. No. 1, P. 152–175. https://doi.org/10.32620/reks.2024.1.13.

46. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive System for the Corporate Area Network Resilience in the Presence of Botnet Cyberattacks. *Computer Networks. CN 2018. Communications in Computer and Information Science*, Vol. 860. 2018. https://doi.org/10.1007/978-3-319-92459-5_31.

47. Savenko O., Sachenko A., Lysenko S., Markowsky G., Vasylkiv N. Botnet Detection Approach Based on the Distributed Systems. *International Journal of Computing*. 2020. Vol. 19(2), P. 190–198. https://doi.org/10.47839/ijc.19.2.1761.

48. Yadav A., Kaur M., Sharma C., Prashar D. Next-gen distributed denial-of-service detection and mitigation in software-defined networking using hybrid machine learning approach. *Soft Computing in Smart Manufacturing and Materials.* 2025. P. 97–133. https://doi.org/10.1016/B978-0-443-29927-8.00005-9.

49. Liu Y., Ren S., Wang X., Zhou M. Temporal Logical Attention Network for

Log-Based Anomaly Detection in Distributed Systems. *Sensors*. 2024. Vol. 24. https://doi.org/10.3390/s24247949.

50. Chaurasia B., Verma A., Verma P. An in-depth and insightful exploration of failure detection in distributed systems. *Computer Networks*. 2024. Vol. 247. https://doi.org/10.1016/j.comnet.2024.110432.

51. Wei X., Wang J., Sun C., Towey D., Zhang S., Zuo W., Yu Y., Ruan R., Song G. Log-based anomaly detection for distributed systems: State of the art, industry experience, and open issues. *Journal of Software: Evolution and Process*. 2024. Vol. 36(8). https://doi.org/10.1002/smr.2650.

52. Vankayalapati R.K. Zero-Trust Security Models for Cloud Data Analytics: Enhancing Privacy in Distributed Systems. *SSRN*. 2025. https://doi.org/10.2139/ssrn.5121185.

53. Di Pilla P., Pareschi R., Salzano F., Zappone F. Listening to what the system tells us: Innovative auditing for distributed systems. *Frontiers in Computer Science*. 2022. Vol. 4. https://doi.org/10.3389/fcomp.2022.1020946.

54. Botha-Badenhorst D., McDonald A.M., Barbour G.D., Buckinjohn E., Gertenbach W. On The Zero-Trust Intranet Certification Problem. *Proceedings of The 19th International Conference on Cyber Warfare and Security*. 2024. Vol. 19(1). https://doi.org/10.34190/iccws.19.1.2054.

55. Wang Y., Yang X. Design and implementation of a distributed security threat detection system integrating federated learning and multimodal LLM. *arXiv*. 2025. https://doi.org/10.48550/arXiv.2502.17763.

56. Zhang H., Jia X., Chen C. Deep Learning-Based Real-Time Data Quality Assessment and Anomaly Detection for Large-Scale Distributed Data Streams. *International Journal of Medical and All Body Health Research*. 2025. Vol. 6(1). https://doi.org/10.54660/IJMBHR.2025.6.1.01-11.

57. Wang B., He Y., Shui Z., Xin Q., Lei H. Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. *Applied and Computational Engineering*. 2024. Vol. 64(1), P. 89–94. https://doi.org/10.54254/2755-2721/64/20241350.

58. Freitas de Souza L. Achieving accountability, reconfiguration, randomness,

and secret leadership in byzantine fault tolerant distributed systems. *Distributed, Parallel, and Cluster Computing [cs.DC], Institut Polytechnique de Paris.* 2024. URL: https://hal.science/tel-04984550 (дата звернення: 21.01.2025).

59. Herath J.D., Yang P., Yan G. Real-Time Evasion Attacks against Deep Learning-Based Anomaly Detection from Distributed System Logs. *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (CODASPY '21).* 2021. P. 29–40. https://doi.org/10.1145/3422337.3447833.

60. Wolf F.A., Müller P. Verifiable Security Policies for Distributed Systems. *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24).* 2024. 4–18. https://doi.org/10.1145/3658644.3690303.

61. Chandan R.R., Torres-Cruz F., Figueroa E.N.T., Mendoza-Mollocondo C.I., Sisodia D.R., Alam T., Tiwari M. Revolutionizing Data Management and Security with the Power of Blockchain and Distributed System. *Meta Heuristic Algorithms for Advanced Distributed Systems.* 2024. Chapter 11. https://doi.org/10.1002/9781394188093.ch11.

62. Han P., Li H., Xue G., Zhang C. Distributed system anomaly detection using deep learning-based log analysis. *Computational Intelligence.* 2023. Vol. 39(3), P. 433–455. https://doi.org/10.1111/coin.12573.

63. Singh V.B.P., Singh P., Guha S.K., Shah A.I., Samdani A., Nomani M.Z.M., Tiwari M. The Future of Financial Crime Prevention and Cybersecurity with Distributed Systems and Computing Approaches. *Meta Heuristic Algorithms for Advanced Distributed Systems.* 2024. Chapter 19. https://doi.org/10.1002/9781394188093.ch19.

64. Allam A.R. Enhancing Cybersecurity in Distributed Systems: DevOps Approaches for Proactive Threat Detection. *Silicon Valley Tech Review.* 2023. Vol. 2(1), P. 54–66. URL: https://www.researchgate.net/publication/385886881_Enhancing_Cybersecurity_in_Distributed_Systems_DevOps_Approaches_for_Proactive_Threat_Detection (дата звернення: 21.01.2025).

65. Popov G., Popova A. Application of System Diversity for Increasing Security and Reliability of Distributed Systems. *2022 XXXI International Scientific Conference Electronics (ET), Sozopol, Bulgaria.* 2022. P. 1–4.

https://doi.org/10.1109/ET55967.2022.9920304.

66. Maher D.P., Ahatlan H.E., Poonegar A.D. A Standardized Trust Model for Enabling Data Security and Interoperability within Smart Distributed Systems. *2023 IEEE International Smart Cities Conference (ISC2), Bucharest, Romania.* 2023. P. 1–4. https://doi.org/10.1109/ISC257844.2023.10293630.

67. Raja M. Comprehensive Framework for Secure Cloud Computing and Distributed Systems with Integrated Cybersecurity and Information Assurance in the Era of Internet of Things. *International Journal of Information Technology Research and Development (IJITRD).* 2025. Vol. 6(2), P. 7–16. URL: https://ijitrd.com/index.php/home/article/view/IJITRD_6_2_2 (дата звернення: 21.01.2025).

68. Lamaazi H., Alneyadi A.M.M., Serhani M.A. Academic Data Privacy-Preserving using Centralized and Distributed Systems: A Comparative Study. *Proceedings of the 2024 6th International Conference on Big-data Service and Intelligent Computation (BDSIC '24).* 2024. P. 8–16. https://doi.org/10.1145/3686540.3686542.

69. Arora D., Sharma O. Fog Computing in Healthcare: Enhancing Security and Privacy in Distributed Systems. *Artificial Intelligence and Cybersecurity in Healthcare.* 2025. Chapter 3. https://doi.org/10.1002/9781394229826.ch3.

70. Palko D., Babenko T., Bigdan A., Kiktev N., Hutsol T., Kuboń M., Hnatiienko H., Tabor S., Gorbovy O., Borusiewicz A. Cyber Security Risk Modeling in Distributed Information Systems. *Applied Sciences.* 2023. Vol. 13(4), 2393. https://doi.org/10.3390/app13042393.

71. Pinto S.J., Siano P., Parente M. Review of Cybersecurity Analysis in Smart Distribution Systems and Future Directions for Using Unsupervised Learning Methods for Cyber Detection. *Energies.* 2023. Vol. 16(4), 1651. https://doi.org/10.3390/en16041651.

72. Dubey H., Kumar S., Chhabra A. Cyber Security Model to Secure Data Transmission using Cloud Cryptography. *Cyber Security Insights Magazine.* 2022. Vol. 2. URL: https://insights2techinfo.com/wp-content/uploads/2022/11/Cyber-Security-Model-to-Secure-Data-Transmission-using-Cloud-Cryptography_final_2.pdf (дата звернення: 21.01.2025).

73. Kyle J., Alexander D. AI-Driven Forensic Tools for Cloud and Edge Computing. *International Journal of Computational Intelligence in Digital Systems.* 2022. Vol. 11(1), P. 29–45. URL: https://www.researchgate.net/publication/388494481_AI-Driven_Forensic_Tools_for_Cloud_and_Edge_Computing (дата звернення: 21.01.2025).

74. Ibrar M., Yin S., Li H., Karim S., Laghari A.A. Comprehensive review of emerging cybersecurity trends and developments. *International Journal of Electronic Security and Digital Forensics.* 2024. Vol. 16(5), P. 633–647. https://doi.org/10.1504/IJESDF.2024.140762.

75. Golightly L., Modesti P., Garcia R., Chang V. Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. *Cyber Security and Applications.* 2023. Vol. 1, 100015. https://doi.org/10.1016/j.csa.2023.100015.

76. Manikyala A., Kommineni H.P., Allam A.R., Nizamuddin M., Sridharlakshmi N.R.B. Integrating Cybersecurity Best Practices in DevOps Pipelines for Securing Distributed Systems. *ABC Journal of Advanced Research.* 2023. Vol. 12(1), P. 57–70. https://doi.org/10.18034/abcjar.v12i1.773.

77. Dey S., Sarma W., Tiwari S. Deep learning applications for real-time cybersecurity threat analysis in distributed cloud systems. *World Journal of Advanced Research and Reviews.* 2023. Vol. 17(3), P. 1044–1058. https://doi.org/10.30574/wjarr.2023.17.3.0288.

78. Hero A., Kar S., Moura J., Neil J., Poor H.V., Turcotte M., Xi B. Statistics and Data Science for Cybersecurity. *Harvard Data Science Review.* 2023. Vol. 5(1). https://doi.org/10.1162/99608f92.a42024d0.

79. Liang X., Konstantinou C., Shetty S., Bandara E., Sun R. Decentralizing Cyber Physical Systems for Resilience: An Innovative Case Study from A Cybersecurity Perspective. *Computers & Security.* 2023. Vol. 124, 102953. https://doi.org/10.101/0167404822003455.

80. Kaur J., Ramkumar K.R. The recent trends in cyber security: A review. *Journal of King Saud University - Computer and Information Sciences.* 2022. Vol. 34(8, Part B), P. 5766–5781. https://doi.org/10.1016/j.jksuci.2021.01.018.

81. Ramskyi I., Drozd A., Lyhun. O. System of Cybersecurity Evaluation of

Corporate Networks. *Computer systems and information technologies*. 2025. Vol. 2.

ДОДАТОК А
(обов'язковий)

ПРЕЗЕНТАЦІЯ РОБОТИ

Система оцінювання кібербезпеки корпоративних мереж

Ігор РАМСЬКИЙ

Науковий керівник к.е.н., доцент Світлана САЧЕНКО

Хмельницький
2025

Актуальність роботи

У сучасних умовах цифрової трансформації та зростання кількості кіберзагроз захист інформаційних ресурсів підприємств набуває критично важливого значення. Корпоративні мережі, які забезпечують взаємодію між численними інформаційними системами, стають основною ціллю для зловмисників, що використовують як традиційні методи атак, так і новітні засоби обходу захисних механізмів. У зв’язку з цим, питання своєчасного виявлення вразливостей, аналізу стану безпеки та оцінювання ризиків у корпоративному середовищі є актуальними як для дослідників, так і для практиків у сфері кібербезпеки.

Сучасні підходи до оцінювання кібербезпеки базуються на застосуванні аналітичних методів, автоматизованих систем моніторингу, а також на використанні формалізованих моделей, які враховують технічні, поведінкові та контекстуальні аспекти загроз. Проте, з огляду на динамічність мережевих середовищ та складність сучасної IT-інфраструктури, виникає потреба у розробці гнучких і масштабованих рішень, які можуть забезпечити об’єктивне оцінювання рівня захищеності корпоративної мережі.

Актуальність роботи полягає в розробці методу створення систем оцінювання кібербезпеки корпоративних мереж з використанням інформації про вразливості комп'ютерів мережі, зв'язків між ними та потенційні небезпеки, пов'язані з людським фактором.

Метою даної роботи є створення системи оцінювання кібербезпеки корпоративних мереж, що враховує технічні параметри комп'ютерних станцій, стан мережевих з'єднань, рівень вразливостей і актуальні загрози.

Для досягнення мети необхідно розв'язати такі основні завдання:

- здійснити аналіз існуючих методів і засобів захисту корпоративних мереж і розподілених систем, а також підходів до оцінювання рівня їх кібербезпеки;

- розробити новий метод оцінювання кібербезпеки в корпоративних мережах;

- реалізувати систему оцінювання кібербезпеки в корпоративних мережах відповідно до розробленого методу;

- дослідити роботу системи.

Об'єктом дослідження є кібербезпека в корпоративних мережах.

Предметом дослідження є методи та засоби кіберзахисту в корпоративних мережах.

Наукова новизна отриманих результатів:

- розроблено новий метод оцінювання кібербезпеки в корпоративних мережах.

На основі проведених досліджень розроблена система оцінювання кібербезпеки корпоративних мереж.

Практична значимість отриманих результатів полягає у розроблені математичного апарату для оцінювання кібербезпеки в корпоративних мережах.

За темою кваліфікаційної роботи опубліковано одну статтю у фаховому науковому журналі CSIT (Computer Systems and Information Technologies).

Метод створення системи

Оцінювання кібербезпеки корпоративних мереж – важлива частина її забезпечення. Постійна робота системи дозволить звертати увагу відповідальних працівників на потенційні атаки, що дозволить їм своєчасно і точно реагувати на недоліки засобів захисту.

Для оцінювання кібербезпеки корпоративної мережі використовується відповідна система. Ця система хоститься на ресурсах цієї ж мережі.

Результатом роботи системи є наближене значення ймовірності корпоративної мережі успішно протидіяти атакам на важливі вузли - такі, що або суттєво завадять роботі, або приведуть до витоку конфіденційних даних.

Функція оцінювання кібербезпеки CS:

$$CS = \prod_{i=1}^{M}\left(\left(1 - V_{a_i}\right) * \prod_{j=1}^{N}\left(1 - V_j G_{j,a_i}\right)\right)$$

Для її розрахування треба розрахувати вразливість вузла мережі (комп'ютерної станції) V:

$$V = \omega_S \sum_{k=1}^{N_e} \omega_k * \frac{CVSS_k}{10} + \omega_P(1 - P) + \omega_U U$$

Та ймовірність розповсюдження успішної атаки з вузла i на вузол j G_{ij}:

$$G_{ij} = \omega_T T_{ij} + \omega_F(1 - F_{ij}) + \omega_L(1 - L_{ij})$$

Для розрахунків функції використовуються різні фреймворки та програми. Для розрахунку загальної вразливості вузла мережі V необхідно виявити слабкі місця на ньому (наприклад, в програмному забезпеченні). Для цього пропонується скористатися сканером вразливостей OpenVAS.

Для кожної знайденої за допомогою сканера вразливості розраховується числове значення небезпеки за допомогою фреймворка CVSS (Common Vulnerability Scoring System). Він враховує основні характеристики вразливості, як, наприклад, ступінь доступу чи вплив на конфіденційність.

Для оцінки ваги кожної з вразливостей використовуються дані від моделі EPSS (Exploit Prediction Scoring System). Ці дані включають загальну ймовірність, що експлойт буде використано для кожного з відомих.

Для розрахування ймовірності поширення атаки між вузлами G_{ij} потрібно також розрахувати показники відкритості з'єднання T_{ij}, шифрування L_{ij} та ефективності фаєрволів та фільтрування F_{ij}.

Для знаходження T_{ij} порахуємо відношення кількості відкритих портів крім стандартних зашифрованих (типу 80 для HTTPS) до максимально допустимої кількості.

Для знаходження L_{ij} використаємо програму сканування (пропонується SSLyze) зв'язків мережі на надійність шифрування – в першу чергу, протоколи TLS та SSL.

Для знаходження F_{ij} згенеруємо тести у вигляді запитів, що імітують шкідливий трафік, (наприклад, за допомогою утиліти hping). F_{ij} у цьому випадку є відношенням провальних тестів (пропущених шкідливих запитів) до загальної кількості тестів.

Робота системи полягає в тому, щоб регулярно перераховувати усі складові функції CS, збирати їх з усіх вузлів мережі та розраховувати значення функції.

Перед початком роботи системи оператор має виставити коефіцієнти, які складно порахувати програмно – наприклад, ті, що стосуються політик безпеки.

За результатами роботи оператор може робити висновки про стан мережі, наступні дії та потенційні виправлення вразливостей.

ВИСНОВКИ

У роботі розроблено новий метод оцінювання кібербезпеки у корпоративних мережах на основі використання оціночної функції. Отримані результати:

- здійснено аналіз існуючих методів і засобів захисту корпоративних мереж і розподілених систем, а також підходів до оцінювання рівня їх кібербезпеки;
- запропонувано метод оцінювання кібербезпеки корпоративних мереж;
- сформулювано цільову функцію, що характеризує рівень кібербезпеки корпоративної мережі;
- реалізувано прототип системи та проведено дослідження ефективності запропонованого методу оцінювання.

ДОДАТОК Б
(обов'язковий)

Наукова праця здобувача

UDC 004.49

IHOR RAMSKYI, ANDRIY DROZD, OLEKSII LYHUN,

Khmelnytskyi National University

SYSTEM FOR CYBERSECURITY EVALUATION OF CORPORATE NETWORKS

In the context of rapidly increasing cyber threats and the growing complexity of corporate IT infrastructure, ensuring a reliable and proactive approach to cybersecurity is becoming critically important for organizations of all sizes. Traditional cybersecurity assessment methods often fail to keep up with the dynamic nature of emerging threats – necessitating the development of more adaptive and intelligent evaluation systems. This article presents a comprehensive modular system for assessing the cybersecurity level of corporate networks – offering a holistic view of the security landscape by integrating both technical and organizational indicators.

The proposed system utilizes self-organizing analytical methods to dynamically process large volumes of data related to vulnerabilities, configuration states, and network behavior patterns. Through intelligent algorithms and adaptive learning, the system is capable of autonomously detecting anomalies, evaluating potential attack vectors, and correlating threats with the network's weak points. Additionally, the inclusion of organizational factors – such as policy compliance, user behavior, and access structures – enables a more contextual and in-depth risk assessment.

A key advantage of the system is its ability to perform real-time monitoring and dynamic risk evaluation – empowering decision-makers to take informed actions in response to incidents. The system's architecture supports scalability and compatibility with existing security tools and network management platforms.

To validate its effectiveness, the system was implemented and tested in a simulated corporate environment reflecting modern structural and operational challenges. The experimental results confirmed its capability to identify vulnerabilities, prioritize responses, and enhance overall cyber resilience.

This research contributes to the advancement of next-generation cybersecurity assessment tools – ensuring the continuous improvement of corporate defense mechanisms in an ever-changing cyber landscape.

Keywords: corporate networks, distributed systems, cybersecurity

І. А. Рамський, А. І. Дрозд, О. О. Лигун,

Хмельницький національний університет

СИСТЕМА ОЦІНЮВАННЯ КІБЕРБЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ

У контексті стрімкого зростання кіберзагроз та зростаючої складності корпоративної IT-інфраструктури забезпечення надійного та проактивного підходу до кібербезпеки стає критично важливим для організацій будь-якого масштабу. Традиційні методи оцінювання кібербезпеки часто не встигають за динамікою змін у загрозах, що зумовлює необхідність розробки більш адаптивних та інтелектуальних систем оцінки. У цій статті представлено комплексну модульну систему для оцінки рівня кібербезпеки корпоративних мереж, яка забезпечує цілісне бачення безпекової ситуації шляхом інтеграції як технічних, так і організаційних показників.

Запропонована система використовує самоорганізуючі аналітичні методи для динамічної обробки великих обсягів даних про вразливості, конфігураційні стани та поведінкові особливості мережі. Завдяки інтелектуальним алгоритмам та адаптивному навчанню система здатна автономно виявляти аномалії, оцінювати потенційні вектори атак і співвідносити загрози з вразливими місцями системи. Додатково, врахування організаційних факторів – таких як відповідність політикам, поведінка користувачів та структура доступу – забезпечує більш контекстуальну та глибоку оцінку ризиків.

Однією з ключових переваг системи є можливість здійснення моніторингу в реальному часі та динамічної оцінки ризиків, що дозволяє керівникам приймати обґрунтовані рішення для своєчасного реагування на інциденти. Архітектура системи передбачає масштабованість і сумісність з існуючими засобами захисту та платформами управління мережею.

Для підтвердження ефективності система була реалізована та протестована у моделюваному корпоративному середовищі, що відображає сучасні структурні та операційні виклики. Результати експерименту підтвердили її здатність виявляти вразливості, визначати пріоритети реагування та зміцнювати загальну кіберстійкість.

Це дослідження робить внесок у розвиток інструментів оцінювання кібербезпеки нового забезпечуючи постійне вдосконалення корпоративних механізмів захисту в умовах мінливого кіберсередовища.

Ключові слова: корпоративні мережі, розподілені системи, кібербезпека.

Introduction

In today's digitally interconnected world, corporate networks have become critical infrastructures that support core business operations, data exchange, and communication processes. As organizations increasingly rely on complex information systems, the potential attack surface expands, exposing networks to a broad range of cyber threats. These threats – ranging from malware and ransomware to advanced persistent threats and insider attacks – continue to grow in sophistication, frequency, and impact. Consequently, ensuring the cybersecurity of corporate networks has evolved from a technical challenge into a strategic necessity for maintaining operational continuity, protecting sensitive data, and preserving stakeholder trust.

Traditional cybersecurity assessment methods often rely on periodic audits, rule-based monitoring, or reactive measures that are insufficient in addressing modern, dynamic threat landscapes. Static approaches fail to capture real-time changes in network topology, user behavior, or system configurations, limiting their effectiveness in identifying and mitigating emerging threats. Furthermore, many existing solutions focus primarily on technical vulnerabilities while neglecting the organizational and procedural factors that also influence the overall security posture.

To address these limitations, there is a growing need for adaptive, comprehensive systems capable of continuously evaluating the cybersecurity state of corporate networks. Such systems should integrate both technical and organizational indicators, provide real-time insights, and support proactive risk management strategies.

This article presents a novel system for cybersecurity evaluation designed specifically for corporate networks. The system incorporates self-organizing analytical methods to interpret vulnerability data, configuration states, and behavioral patterns across the network. It enables real-time monitoring, dynamic risk assessment, and prioritization of mitigation efforts based on contextual analysis. The architecture is modular and scalable, allowing for seamless integration into diverse IT environments.

The following sections describe the system's design and implementation, followed by an evaluation of its performance within a simulated enterprise environment. The results demonstrate the system's ability to enhance situational awareness, support decision-making, and improve the overall cybersecurity resilience of corporate networks.

Related works

Assessing cybersecurity in corporate networks requires sophisticated methods for detecting and responding to various threats. Modern corporate networks function as distributed systems with partial centralization, where decision-making on malware detection is structured as a decentralized subsystem. The use of characteristic indicators and analytical models allows the system to evaluate the constituent states and determine the corresponding reactions. Among the existing approaches, there is one that combines several methods for detecting malware, treating system components as integral sensors [1][2].

Ensuring resilience to cyberattacks, particularly botnets, is a critical aspect of cybersecurity assessment. The reviewed literature provides an example of a self-adaptive system for reconfiguring corporate networks based on security scenarios obtained as a result of cluster analysis of network traffic features. Using a semi-supervised fuzzy c-means clustering approach, the system detects cyber threats and

selects security strategies to mitigate botnet attacks, increasing network resilience [3]. Another three-tier botnet detection system model provides the ability to identify both known and unknown botnets by combining host-level Bayes classification with network-level extensions. This approach allows for efficient exchange of information in a distributed system and has demonstrated promising results in the accuracy of botnet detection [4].

Distributed denial-of-service (DDoS) attacks are another major cybersecurity issue, especially in software-defined networks (SDNs). To detect and mitigate these attacks, a machine learning-based framework has been developed that uses the Support Vector Classifier and the Gradient Boost Classifier (SVC-GBC). With 99.4% accuracy, this hybrid approach significantly improves SDN security by refining detection granularity and strengthening defense mechanisms [5]. In addition to intrusion detection, anomaly detection in distributed systems remains a challenge due to complex dependencies between system logs. A deep learning-based Time Logical Attention Network (TLAN) has been introduced to model both time series patterns and logical dependencies, improving anomaly detection performance while reducing false signals [6].

The reliability of cybersecurity assessments in distributed systems is further enhanced by failure detection mechanisms. These mechanisms monitor the activity of nodes to identify faults and increase the fault tolerance of the system. Systematic analysis of fault detectors in distributed environments highlights their role in ensuring the reliability of services by solving matching and failure problems [7]. Log-based anomaly detection (LAD) also plays an important role in cybersecurity assessment, using system logs to identify potential threats and service anomalies. The overall structure of LAD for distributed systems includes logging grouping and feature mining to improve detection efficiency, demonstrating its applicability in real-world distributed environments [8].

In addition, privacy issues in distributed computing require robust security systems. The study of privacy in distributed systems focuses on the risks associated with data evaluation and information tracking, emphasizing the relevance of zero-trust security models for the secure implementation of systems in cloud architectures [9]. As the complexity of distributed systems continues to grow, effective system audit mechanisms that combine advanced analytics and artificial intelligence are becoming important for vulnerability monitoring and improving security [10].

These advances together contribute to the creation of a comprehensive cybersecurity assessment system that ensures the resilience of corporate networks to evolving threats. Cybersecurity assessments in corporate networks should address issues related to reliability, anomaly detection, and compliance with security policies. The zero-trust security model emphasizes the need to validate on-premises servers on corporate intranets, however, existing certification methods remain unavailable to small organizations due to cost and complexity. This gap leads to dependence on self-signed certificates, increasing vulnerability to impersonation and unauthorized access, which ultimately violates the principles of zero trust [11]. To

improve the detection of security threats in large-scale distributed systems, a federated approach based on learning has been proposed, integrating multimodal large language models. This system handles a variety of data sources, achieving 96.4% accuracy while maintaining data confidentiality and computational efficiency, demonstrating significant improvements over traditional detection methods [12].

Anomalies in distributed systems pose significant risks due to time delays and deterioration in data quality. A deep learning-based real-time data quality assessment system has been implemented, which uses adaptive neural networks and parallel processing to provide scalable, low-latency anomaly detection. Evaluations on large-scale datasets confirm the system's effectiveness in maintaining high detection accuracy when processing more than 1.2 million events per second [13]. In cloud computing environments, optimizing resource allocation is critical to maintaining efficiency. Machine learning-based approaches, combining deep learning and genetic algorithms, have been developed to improve resource planning, addressing issues such as load imbalances and low utilization [14].

Further advances in distributed computing focus on accountability, leadership selection, and safe randomness generation. The framework for accountable and reconfigured distributed systems enables seamless adaptation in response to failures using lattice agreement abstraction. In addition, innovative cryptographic protocols improve leadership elections on partially synchronous blockchains, improving consensus mechanisms and system resilience [15]. As distributed systems increasingly rely on log-based monitoring to assess security, the reliability of deep learning models against malicious attacks is a growing concern. A new attack method, LAM, manipulates streaming logs to avoid detecting anomalies, highlighting the need for enhanced security measures against adversarial manipulation [16].

Security policies in distributed systems also need to be flexible and validated in different implementations. A language-independent policy review system ensures compliance with security policies by analyzing I/O behavior instead of relying on programming language restrictions. Evaluations demonstrate its applicability in real-world protocols, which reinforces the need for adaptive security policies [17]. Blockchain technology also contributes to cybersecurity by increasing the transparency and security of data in distributed governance systems. However, issues such as scalability and interoperability must be addressed in order to fully exploit the potential of blockchain to protect sensitive data [18]. Finally, advances in deep learning to detect anomalies in distributed system logs introduce models that integrate global spatiotemporal features, greatly improving the accuracy of detecting security threats in complex environments [19]. These changes combine to contribute to the reliability and effectiveness of cybersecurity assessments in corporate networks.

Cybersecurity assessments in corporate networks must constantly adapt to changing threats and technological advancements. Distributed systems and computational approaches, including blockchain technology and distributed ledgers, offer significant potential to improve financial crime prevention and cybersecurity by increasing transparency and reducing fraud risks. However, issues such as regulatory

compliance, interoperability, and integration with existing infrastructures must be addressed to maximize these benefits [20]. A proactive approach to security is essential in distributed environments, and the integration of DevOps methodologies enhances security by embedding threat detection into the development lifecycle, automating monitoring, and using behavioral analytics to detect anomalies in real-time. This strategy contributes to the formation of a culture of shared responsibility for safety and compliance with legal standards [21].

The diversity of systems is another key factor in improving the reliability and security of distributed communication networks. Analytical models based on tension-force analysis quantify these improvements, providing valuable information about the stability of the system [22]. In the context of intelligent distributed systems (SDS), ensuring data security and interoperability is critical for the seamless exchange of information between industries such as healthcare, utilities, and supply chains. Setting global security standards can provide a framework for authentication, collaboration, and protection against cyber threats in SDS environments [23]. The growing integration of IoT with cloud computing introduces new vulnerabilities, requiring a comprehensive security framework that increases resilience to cyber threats while maintaining scalability and adaptability in distributed environments [24].

Data privacy remains a major concern, especially in areas such as education and healthcare. Distributed computing offers improvements in security and response times, however, centralized platforms often outperform distributed systems with privacy-preserving techniques such as k-anonymity, t-proximity, and β-probability. Comparative analysis of these approaches reveals trade-offs in runtime, memory requirements, and suppression levels [25]. In healthcare, foggy computing is a promising solution for real-time patient monitoring, but security and privacy concerns must be addressed through encryption, access control, and data analysis techniques that preserve privacy [26]. Risk assessment in distributed information systems requires a dynamic, multi-layered approach that integrates quantitative, qualitative, and hybrid methodologies, using security metrics for accurate and reliable cybersecurity assessments [27].

Cybersecurity threats in smart networks highlight the importance of advanced threat detection mechanisms. Traditional supervised learning methods for detecting cyberattacks require a variety of training datasets that may not always be available. Unsupervised data mining approaches, especially for detecting false data attacks (FDIA), offer a more efficient alternative, relying solely on conventional event data to train detection models. Comparative studies demonstrate that unsupervised algorithms are superior to supervised and deep learning methods in detecting unknown attack patterns, increasing cybersecurity in smart grid infrastructures [28].

These advances combine to strengthen cybersecurity assessment systems in corporate networks, ensuring resilience to sophisticated cyber threats. Cybersecurity assessments in corporate networks should include advanced cryptographic techniques to reduce the risks of data breaches in distributed environments. Cloud cryptography plays a crucial role in protecting data storage and transmission through the use of

encryption mechanisms, intrusion detection systems, and firewalls. These technologies strengthen data protection in cloud-based distributed systems, preventing unauthorized access and infiltration of malware [29]. With the expansion of cloud and edge computing, AI-powered forensic tools have become effective solutions for detecting and mitigating the effects of cyber incidents in real-time. Machine learning and deep learning techniques improve forensic analysis by improving scalability, accuracy, and response time when detecting cyber threats in distributed systems [30].

The function for evaluation fo cybersecurity of computer stations

Let's set two functions to assess the level of network security, where the first will reflect the likelihood of significant interference of an attacker in any critical component of the network.

First, let's define the vulnerability of a component as the probability of its compromise regardless of the rest present in the network. Corresponding formula is:

$$V = \omega_S S + \omega_P(1 - P) + \omega_U U, \quad (1)$$

where S is the software vulnerability level in range [0,1], P is the effectiveness of cybersecurity policies in range [0, 1], 1 standing for maximal security, U – probability of compromise due to a human error, $\omega_S, \omega_P, \omega_U$ are the weight coefficients.

Let's reveal the components of the formula further. P should be defined by cybersecurity professionals independently on a case-by-case basis, as different organizations have different approaches to setting up appropriate processes. In the context of this work, we will determine U according to the frequency of phishing attacks and other situations of compromise of network users in its history. S will be determined by the formula

$$S = \sum_{k=1}^{N_e} \omega_k * \frac{CVSS_k}{10}, \quad (2)$$

where N_e is the total number of vulnerabilities on the node, $CVSS_k$ is the assessment of the criticality of the k-th vulnerability on the CVSS scale (from 0 to 10), ω_k is the weighting coefficient, which determines the impact of each vulnerability.

Vulnerability search for S calculation can be organized using vulnerability scanners. Thus, the formula for the vulnerability of one component independently of the rest of the network:

$$V = \omega_S \sum_{k=1}^{N_e} \omega_k * \frac{CVSS_k}{10} + \omega_P(1 - P) + \omega_U U, \quad (3)$$

It should also be borne in mind that the compromise of one host in the network also endangers other components of the network. To do this, we will specify a formula to determine the probability of compromise of host j if host i was compromised:

$$G_{ij} = \omega_T T_{ij} + \omega_F(1 - F_{ij}) + \omega_L(1 - L_{ij}), \quad (4)$$

where T_{ij} is the the level of connection openness normalized in the range [0,1], where 1 means a fully open channel and 0 is a fully isolated connection, F_{ij} is the effectiveness of firewalls and traffic filtering (from 0 to 1, where 1 means maximum protection), L_{ij} is the encryption level (0 to 1, where 1 means full encryption and 0 means fully open traffic).

Let's put these two formulas together to determine the probability of its compromise for each host and, accordingly, calculate the chance of compromise of any of the important hosts.

$$CS = \prod_{i=1}^{M} \left(\left(1 - V_{a_i}\right) * \prod_{j=1}^{N} \left(1 - V_j P_{a_i}\right) \right), \quad (5)$$

where CS is the overall level of cybersecurity in the corporate network, M is the number of important network components, a is the list of important network components.

These formulas are based on comprehensive mathematical modeling that adequately accounts for both the internal characteristics of each host and the interdependencies between them. The vulnerability level of each node V is determined by three key parameters: software vulnerabilities S, the effectiveness of security policies P, and the probability of compromise due to human factors U. This structure aligns with modern cybersecurity threat analysis practices, where most incidents stem not only from technical flaws but also from social engineering and imperfect security administration. The use of weighting coefficients enables the model to reflect the relative importance of each factor in a given context, making the evaluation adaptable to the specific conditions of the network.

Further modeling of the probability of attack propagation across the network through the function $G(i, j)$ captures the probabilistic nature of inter-node interaction, where the risk of transmission depends on parameters such as connection openness, firewall effectiveness, encryption levels, and anomaly detection capabilities. This formula is crucial, as it accounts for not only the vulnerability of individual components but also their potential influence on other nodes—an essential distinction from traditional approaches that treat hosts in isolation.

The final stage involves the calculation of the overall cybersecurity level of the network CS, which is derived by combining all obtained V and G values. The formula for CS implements a multiplicative scheme that accurately reflects the cumulative nature of risks: even if a single host is highly vulnerable and located in a poorly protected segment, it can impact the security of the entire system. This approach allows for the estimation of the probability of a successful attack not only on isolated components but on critical infrastructure as a whole.

Taken together, the proposed formulas are not only mathematically sound but also effective in addressing the task of constructing a comprehensive cybersecurity evaluation model for corporate networks. They provide a high degree of accuracy, adaptability to changes in system configuration, and the ability to tailor

to specific threats and architectures, making the proposed methodology universally applicable across a wide range of practical implementations.

Practical implementation of the system

The method for synthesizing self-organizing systems for cybersecurity assessment of computer stations is based on constructing a system capable of real-time monitoring of the corporate network and individual computer stations. It continuously collects relevant metrics and computes a cybersecurity evaluation function. The central element of this system is a function that reflects the current level of protection of the information infrastructure, taking into account numerous interdependent factors. This function should be formed based on aggregated indicators of system process activity, configuration integrity, network connection status, and the degree of vulnerability derived from known technical software characteristics and the enforcement level of access control policies.

To deploy the evaluation system, an initial configuration of coefficients and values is required—parameters that cannot be accurately assessed using purely technical methods. Let us now consider Formula 3, which calculates the vulnerability of each individual computer in the network:

$$V = \omega_S \sum_{k=1}^{N_e} \omega_k * CVSS_k + \omega_P(1 - P) + \omega_U U$$

In this formula, the weighting coefficients $\omega_S, \omega_P, \omega_U$, as well as the values of P and U under ideal circumstances, should be determined by cybersecurity experts for each specific case of a corporate network. This approach assumes individual customization of the evaluation system, taking into account the architecture's specifics, the types of information assets, the organizational structure of the enterprise, as well as the potential attack vectors characteristic of a particular industry or region. Alternatively, the following values for the weighting coefficients are proposed:

Network Scenario	ω_S	ω_P	ω_U
Techno-centric organization	0.7	0.2	0.1
Institution with a bureaucratic structure	0.2	0.5	0.3
Company under active phishing conditions	0.3	0.2	0.5

Similarly, the values P and U should also be determined by cybersecurity experts (ideally) based on an audit that demonstrates the network's security policies comply with the latest standards and that personnel are knowledgeable and proficient in computer usage. Alternatively, the value of P can be roughly estimated based on components such as the existence of documented security policies, the currency of the policies, access control, password management, and incident response. Likewise, the value of U can be approximated based on other factors and historical data: the frequency of phishing incidents over the past year, the level of personnel awareness (tests/surveys), the availability of regular training, incidents of password/access loss, and the results of social engineering simulations.

To determine the remaining values in the formula ($\omega_k, N_e, CVSS_k$) specialized software and additional resources are required. To obtain $CVSS_k$, it is recommended to use the OpenVAS vulnerability scanner. This is a free and open-source software – which ensures there is no misuse of network access by the developers – provided that changes to the open code are regularly reviewed. For the cybersecurity evaluation system to function properly, it is necessary to regularly run vulnerability scans on the computer. As a result of these scans, the program generates a report, and the CVSS values extracted from it will be used for further calculations.

To determine ω_k, N_e , it is proposed to use daily updated data from the Exploit Prediction Scoring System (EPSS) model. This is a system that estimates the probability that a specific vulnerability will be exploited in the real world within the next 30 days. Data can be obtained via API or by downloading reports in CSV format. Each row in the file is a triplet: CVE (vulnerability identifier), EPSS (probability of exploitation), Percentile (probability percentile for the given vulnerability). N_e will be taken as the number of vulnerabilities in the EPSS report, and $\omega_k - EPSS_k$, normalized in such a way that the sum of all values equals one. In this way, the weight of a vulnerability will be proportional to the probability of encountering it.

Let us consider formula 4:

$$G_{ij} = \omega_T T_{ij} + \omega_F(1 - F_{ij}) + \omega_L(1 - L_{ij}) + \omega_D(1 - D_{ij}),$$

where T_{ij} is the level of openness within the range [0, 1], F_{ij} is the effectiveness of firewalls and traffic filtering within the range [0, 1], L_{ij} is the level of encryption within the range [0, 1], D_{ij} is the level of anomaly detection within the range [0, 1], $\omega_T, \omega_F, \omega_L, \omega_D$ – the weighting coefficients.

The weighting coefficients $\omega_T, \omega_F, \omega_L, \omega_D$ should be defined by the CISO (Chief Information Security Officer) or a security analyst. For example, in a cloud environment with many open ports but strong encryption – more weight should be assigned to ω_T, and less to ω_L, whereas in an environment without IDS/IPS (Intrusion Detection/Prevention Systems) – ω_D should be increased.

This can be implemented in the form of a risk profile table:

Scenario	ω_T	ω_F	ω_L	ω_D
Cloud infrastructure	0.1	0.5	0.2	0.2
Corporate local network	0.1	0.4	0.3	0.2
Minimal access control	0.1	0.2	0.1	0.4

It is also necessary to define T_{ij}, F_{ij}, L_{ij}, D_{ij}. Let us calculate T_{ij}:

$$T_{ij} = \frac{N_o}{N_a}, \tag{4.1}$$

where N_o is the number of open ports excluding standard encrypted ones (e.g., HTTPS), and N_a is the maximum allowable number of open ports, typically set to 10.

Let us calculate F_{ij}. This is done through periodic active testing – by generating requests that simulate malicious traffic. It is recommended to use the open-source tool **hping** to generate such traffic. The formula is:

$$F_{ij} = \frac{N_{failed}}{N_{tests}}, \tag{4.2}$$

where N_{failed} is the number of malicious test requests that were not blocked during testing, and N_{tests} – is the total number of tests conducted.

Let us calculate L_{ij}. It is proposed to use the tool **SSLyze** to scan network connections and assess the strength of encryption. Based on the scan results, a numerical value can be estimated for use in formula (4). Since TLS 1.3 is currently considered the most secure transport layer encryption protocol, it is rated as $L_{ij} = 1$. SSL, being outdated and known to contain vulnerabilities, is rated as $L_{ij} = 0$. . For intermediate values, we assign $L_{ij} = 0.7$ for TLS 1.2 and $L_{ij} = 0.3$ for TLS 1.1.

Results of the experiment

To evaluate the effectiveness of the proposed model, an experiment was conducted that simulates the operation of the implemented cybersecurity assessment system under conditions close to a real-world environment. The testing involved simulating the activity of network nodes over the course of one week with an hourly time step. During the experiment, dynamic updates of input parameters were implemented – these parameters influence the vulnerability level of individual computers and the probability of their compromise as a result of interaction with other nodes in the network.

The model components responsible for forming the vulnerability and compromise probability functions were manually configured based on assumptions about the typical characteristics of an organizational IT environment. In particular, the weight coefficients for the technical, policy-related, and

human vulnerability components were set according to conditionally prioritized security concerns. Similarly, the weights for traffic, filtering, encryption, and network remoteness parameters were chosen to reflect the characteristic risks of network intrusion through interactions between individual computers. The values of the manually configured parameters are as follows: $\omega_S = 0.7, \omega_P = 0.2, \omega_U = 0.1, \omega_T = 0.1, \omega_F = 0.4, \omega_L = 0.3, \omega_D = 0.2, P = 0.9, U = 0.1$.

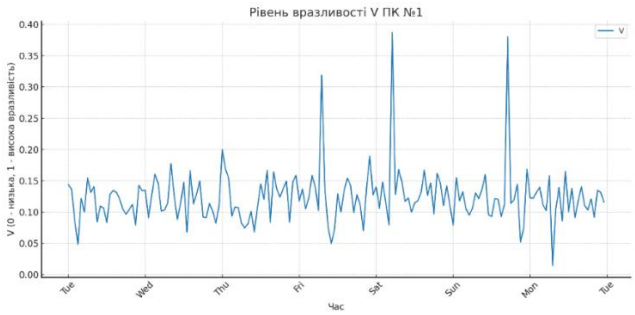


Fig. 1

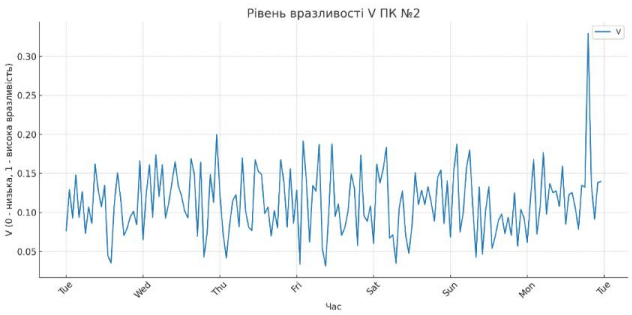


Fig. 2

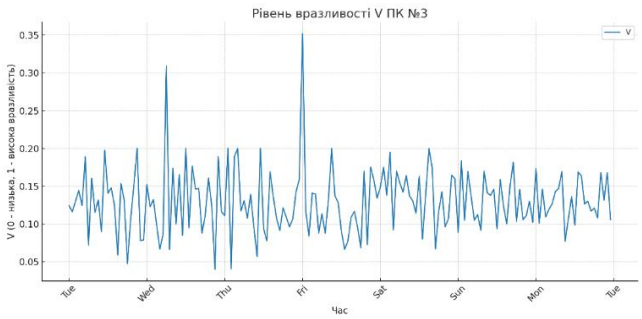


Fig. 3

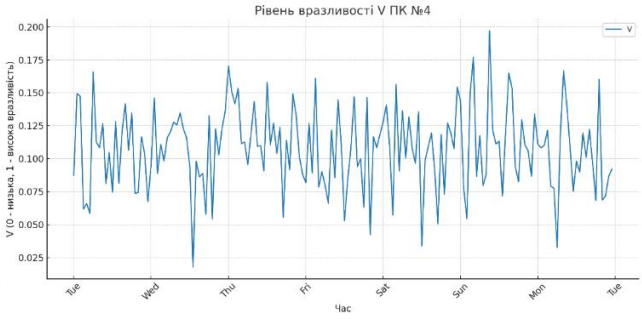


Fig. 4

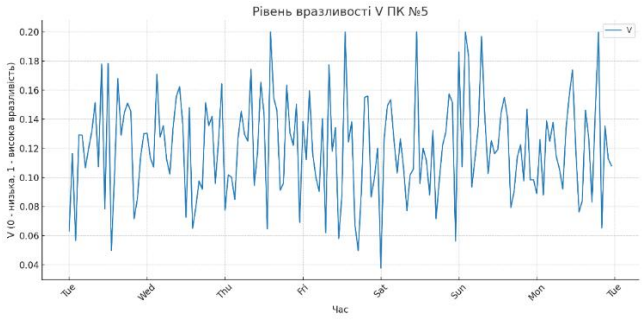


Fig. 5

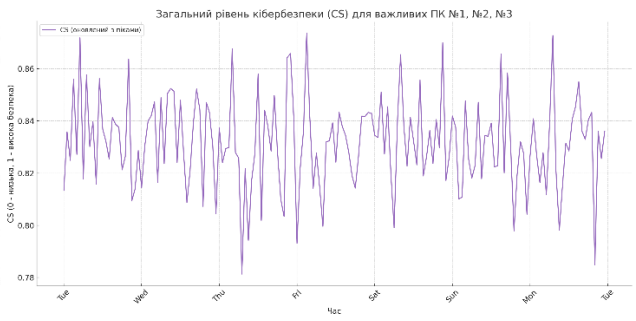


Fig. 6

As part of the experiment, isolated peak deviations were manually introduced for critical nodes – computers No. 1-3, – simulating episodic increases in risk level. These peaks were implemented by artificially adding a noticeable number of high-rated technical vulnerabilities, equivalent to a situation where a new set of critical vulnerabilities is discovered on a specific host, for example, due to a missed

update or newly identified software flaws. As a result, there were short-term but sharp increases in the V indicator, which are clearly visible in Fig. 1–3. Fig. 4-5 show vulnerability chart with no serious peaks.

The chart of the overall cybersecurity level CS (Fig. 6) serves as a key analytical tool that enables a comprehensive assessment of the security situation within the network, taking into account both the local characteristics of individual nodes and the impact of inter-node interactions. The construction of this indicator is based on integrating the vulnerability assessments of critical computers with the probabilities of their compromise by other elements of the system. This approach provides a multidimensional view of risks, allowing not only for isolated evaluations of individual hosts but also for tracking systemic dependencies and potential attack chains.

This chart holds particular value from the perspective of real-time monitoring – it makes it possible to identify critical time intervals during which a sharp decline in the security level is observed, and to correlate these changes with specific hosts exhibiting increased vulnerability or an escalating threat of compromise. In combination with the V_i graphs, which provide detailed insight into the sources of these changes, the CS graph enables the operator to instantly assess the overall network situation, localize problem areas, and take timely measures to eliminate vulnerabilities or reduce the risk of attack propagation.

Thus, CS visualization serves as an effective real-time decision-making mechanism, which is especially important in the context of a rapidly changing threat landscape. Its integration into the security management system significantly enhances the response speed and the rationality of actions taken by the administrator or automated defense systems.

Conclusions

The proposed system for cybersecurity evaluation of corporate networks effectively integrates technical, organizational, and human factors into a comprehensive framework. By employing adaptive mathematical modeling and real-time data analysis, it provides an accurate, dynamic assessment of a network's security posture. The approach's strength lies in its flexibility—allowing parameter customization based on the specifics of an organization—and its capability to evaluate not only isolated vulnerabilities but also interdependencies between network nodes. Experimental implementation demonstrated the model's practical applicability and its usefulness for identifying weak points, prioritizing response measures, and enhancing decision-making in security management. This system represents a significant step forward in proactive cybersecurity assessment, offering organizations a scalable and intelligent tool to fortify their digital infrastructure against evolving threats.

References

1. 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. 265–270. https://doi.org/10.1109/IDAACS58523.2023.10348773

2. 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. No. 1, 152–175. https://doi.org/10.32620/reks.2024.1.13
3. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive System for the Corporate Area Network Resilience in the Presence of Botnet Cyberattacks. *Computer Networks. CN 2018. Communications in Computer and Information Science,* Vol. 860. 2018. https://doi.org/10.1007/978-3-319-92459-5_31
4. Savenko O., Sachenko A., Lysenko S., Markowsky G., Vasylkiv N. Botnet Detection Approach Based on the Distributed Systems. *International Journal of Computing.* 2020. Vol. 19(2), 190–198. https://doi.org/10.47839/ijc.19.2.1761
5. Yadav A., Kaur M., Sharma C., Prashar D. Next-gen distributed denial-of-service detection and mitigation in software-defined networking using hybrid machine learning approach. *Soft Computing in Smart Manufacturing and Materials.* 2025. 97–133. https://doi.org/10.1016/B978-0-443-29927-8.00005-9
6. Liu Y., Ren S., Wang X., Zhou M. Temporal Logical Attention Network for Log-Based Anomaly Detection in Distributed Systems. *Sensors.* 2024. Vol. 24. https://doi.org/10.3390/s24247949
7. Chaurasia B., Verma A., Verma P. An in-depth and insightful exploration of failure detection in distributed systems. *Computer Networks.* 2024. Vol. 247. https://doi.org/10.1016/j.comnet.2024.110432
8. Wei X., Wang J., Sun C., Towey D., Zhang S., Zuo W., Yu Y., Ruan R., Song G. Log-based anomaly detection for distributed systems: State of the art, industry experience, and open issues. *Journal of Software: Evolution and Process.* 2024. Vol. 36(8). https://doi.org/10.1002/smr.2650
9. Vankayalapati R.K. Zero-Trust Security Models for Cloud Data Analytics: Enhancing Privacy in Distributed Systems. *SSRN.* 2025. https://doi.org/10.2139/ssrn.5121185
10. Di Pilla P., Pareschi R., Salzano F., Zappone F. Listening to what the system tells us: Innovative auditing for distributed systems. *Frontiers in Computer Science.* 2022. Vol. 4. https://doi.org/10.3389/fcomp.2022.1020946
11. Botha-Badenhorst D., McDonald A.M., Barbour G.D., Buckinjohn E., Gertenbach W. On The Zero-Trust Intranet Certification Problem. *Proceedings of The 19th International Conference on Cyber Warfare and Security.* 2024. Vol. 19(1). https://doi.org/10.34190/iccws.19.1.2054
12. Wang Y., Yang X. Design and implementation of a distributed security threat detection system integrating federated learning and multimodal LLM. *arXiv.* 2025. https://doi.org/10.48550/arXiv.2502.17763
13. Zhang H., Jia X., Chen C. Deep Learning-Based Real-Time Data Quality Assessment and

Anomaly Detection for Large-Scale Distributed Data Streams. *International Journal of Medical and All Body Health Research.* 2025. Vol. 6(1). https://doi.org/10.54660/IJMBHR.2025.6.1.01-11

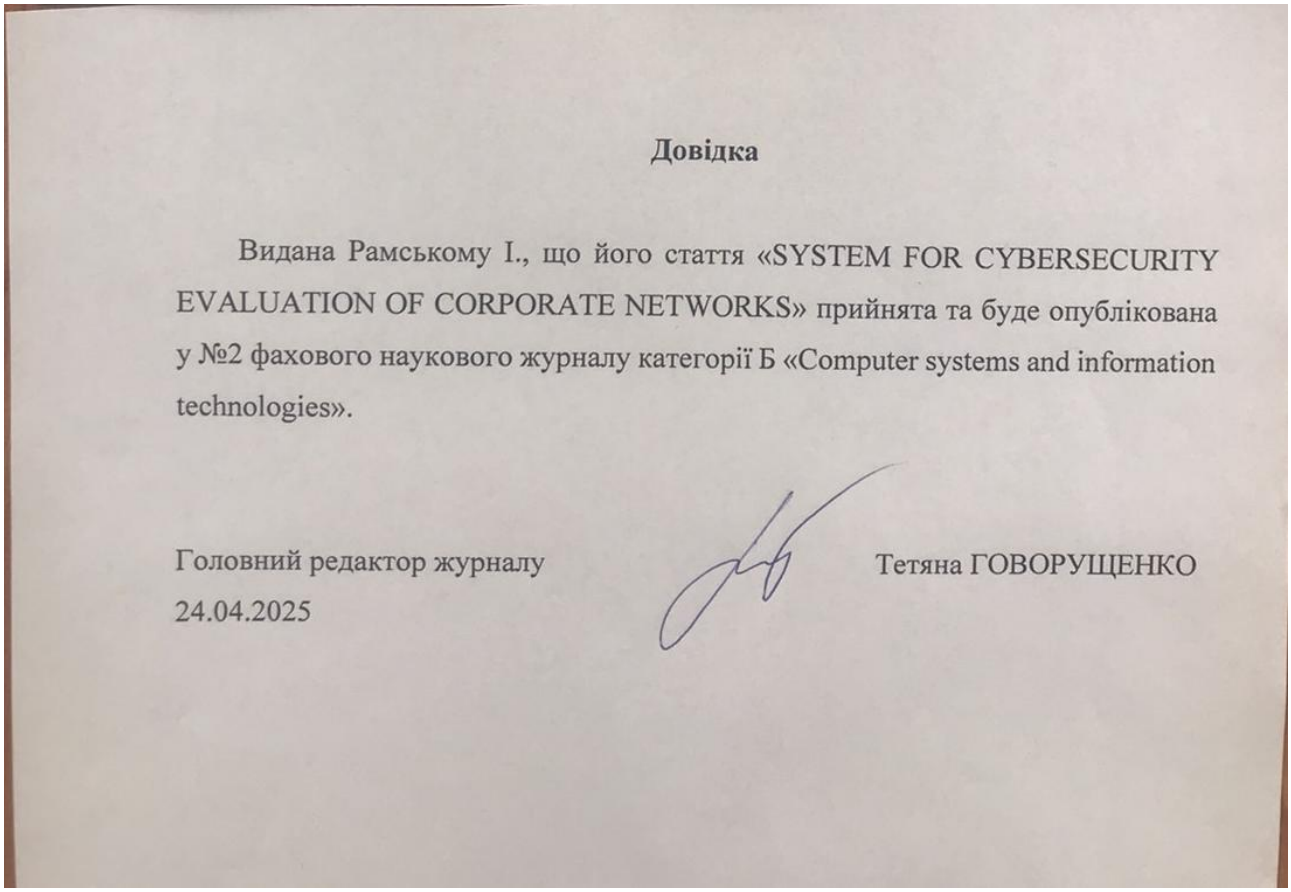
14. Wang B., He Y., Shui Z., Xin Q., Lei H. Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. *Applied and Computational Engineering.* 2024. Vol. 64(1), 89–94. https://doi.org/10.54254/2755-2721/64/20241350
15. Freitas de Souza L. Achieving accountability, reconfiguration, randomness, and secret leadership in byzantine fault tolerant distributed systems. *Distributed, Parallel, and Cluster Computing [cs.DC], Institut Polytechnique de Paris.* 2024. URL: https://hal.science/tel-04984550 (access date: 21.01.2025)
16. Herath J.D., Yang P., Yan G. Real-Time Evasion Attacks against Deep Learning-Based Anomaly Detection from Distributed System Logs. *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (CODASPY '21).* 2021. 29–40. https://doi.org/10.1145/3422337.3447833
17. Wolf F.A., Müller P. Verifiable Security Policies for Distributed Systems. *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24).* 2024. 4–18. https://doi.org/10.1145/3658644.3690303
18. Chandan R.R., Torres-Cruz F., Figueroa E.N.T., Mendoza-Mollocondo C.I., Sisodia D.R., Alam T., Tiwari M. Revolutionizing Data Management and Security with the Power of Blockchain and Distributed System. *Meta Heuristic Algorithms for Advanced Distributed Systems.* 2024. Chapter 11. https://doi.org/10.1002/9781394188093.ch11
19. Han P., Li H., Xue G., Zhang C. Distributed system anomaly detection using deep learning-based log analysis. *Computational Intelligence.* 2023. Vol. 39(3), 433–455. https://doi.org/10.1111/coin.12573
20. Singh V.B.P., Singh P., Guha S.K., Shah A.I., Samdani A., Nomani M.Z.M., Tiwari M. The Future of Financial Crime Prevention and Cybersecurity with Distributed Systems and Computing Approaches. *Meta Heuristic Algorithms for Advanced Distributed Systems.* 2024. Chapter 19. https://doi.org/10.1002/9781394188093.ch19
21. Allam A.R. Enhancing Cybersecurity in Distributed Systems: DevOps Approaches for Proactive Threat Detection. *Silicon Valley Tech Review.* 2023. Vol. 2(1), 54–66. URL: https://www.researchgate.net/publication/385886881_Enhancing_Cybersecurity_in_Distributed_Systems_DevOps_Approaches_for_Proactive_Threat_Detection (access date: 21.01.2025)
22. Popov G., Popova A. Application of System Diversity for Increasing Security and Reliability of Distributed Systems. *2022 XXXI International Scientific Conference Electronics (ET), Sozopol, Bulgaria.* 2022. 1–4. https://doi.org/10.1109/ET55967.2022.9920304

23. Maher D.P., Ahatlan H.E., Poonegar A.D. A Standardized Trust Model for Enabling Data Security and Interoperability within Smart Distributed Systems. *2023 IEEE International Smart Cities Conference (ISC2), Bucharest, Romania.* 2023. 1–4. https://doi.org/10.1109/ISC257844.2023.10293630
24. Raja M. Comprehensive Framework for Secure Cloud Computing and Distributed Systems with Integrated Cybersecurity and Information Assurance in the Era of Internet of Things. *International Journal of Information Technology Research and Development (IJITRD).* 2025. Vol. 6(2), 7–16. URL: https://ijitrd.com/index.php/home/article/view/IJITRD_6_2_2 (access date: 21.01.2025)
25. Lamaazi H., Alneyadi A.M.M., Serhani M.A. Academic Data Privacy-Preserving using Centralized and Distributed Systems: A Comparative Study. *Proceedings of the 2024 6th International Conference on Big-data Service and Intelligent Computation (BDSIC '24).* 2024. 8–16. https://doi.org/10.1145/3686540.3686542
26. Arora D., Sharma O. Fog Computing in Healthcare: Enhancing Security and Privacy in Distributed Systems. *Artificial Intelligence and Cybersecurity in Healthcare.* 2025. Chapter 3. https://doi.org/10.1002/9781394229826.ch3
27. Palko D., Babenko T., Bigdan A., Kiktev N., Hutsol T., Kuboń M., Hnatiienko H., Tabor S., Gorbovy O., Borusiewicz A. Cyber Security Risk Modeling in Distributed Information Systems. *Applied Sciences.* 2023. Vol. 13(4), 2393. https://doi.org/10.3390/app13042393
28. Pinto S.J., Siano P., Parente M. Review of Cybersecurity Analysis in Smart Distribution Systems and Future Directions for Using Unsupervised Learning Methods for Cyber Detection. *Energies.* 2023. Vol. 16(4), 1651. https://doi.org/10.3390/en16041651
29. Dubey H., Kumar S., Chhabra A. Cyber Security Model to Secure Data Transmission using Cloud Cryptography. *Cyber Security Insights Magazine.* 2022. Vol. 2. URL: https://insights2techinfo.com/wp-content/uploads/2022/11/Cyber-Security-Model-to-Secure-Data-Transmission-using-Cloud-Cryptography_final_2.pdf (access date: 21.01.2025)
30. Kyle J., Alexander D. AI-Driven Forensic Tools for Cloud and Edge Computing. *International Journal of Computational Intelligence in Digital Systems.* 2022. Vol. 11(1), 29–45. URL: https://www.researchgate.net/publication/388494481_AI-Driven_Forensic_Tools_for_Cloud_and_Edge_Computing (access date: 21.01.2025)

Ihor Ramskyi – Master's degree student, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: ramskyihor@gmail.com
https://orcid.org/0009-0007-6175-1923

Andriy Drozd – PhD student, Khmelnytskyi National University, Khmelnytskyi, Ukraine,
e-mail: andriydrozdit@gmail.com
https://orcid.org/0009-0008-1049-1911

Oleksii Lyhun – PhD student, Khmelnytskyi National University, Khmelnytskyi, Ukraine
e-mail: oleksii.lyhun@gmail.com
https://orcid.org/0009-0004-5727-5096



Довідка

Видана Рамському І., що його стаття «SYSTEM FOR CYBERSECURITY EVALUATION OF CORPORATE NETWORKS» прийнята та буде опублікована у №2 фахового наукового журналу категорії Б «Computer systems and information technologies».

Головний редактор журналу Тетяна ГОВОРУЩЕНКО
24.04.2025

ДОДАТОК В

Результати перевірки на плагіат

Tue Apr 15 14:31:40 EEST 2025, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.260 Educational

Максимальне співпадіння з одним документом 21.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 14%**

ID: 229498 Назва: МКР Система оцінювання кібербезпеки корпоративних мереж Додано в БД: 2025-04-15 Автора: Ігор РАМСЬКИЙ Керівники: Світлана САЧЕНКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	153141	945	33035 (22%)	203 (21%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
193937	Назва: Звіт з ПДП Система оцінювання кібербезпеки корпоративних мереж Додано в БД: 2025-03-24 Автора: Рамський І.А. Керівники: Гнатчук Є.Г. Консультанти: Опоненти:	31861 (21.0%)	188 (20.0%)

StrikePlagiarism.com

Дата звіту 4/15/2025

Дата редагування 4/15/2025

Документ прийнятий

Звіт подібності

метадані

Назва організації

Khmelnytskyi National University

Заголовок

РАМСЬКИЙ_Система оцінювання кібербезпеки корпоративних мереж

Автор

Ігор РАМСЬКИЙ Науковий керівник / Експерт

підрозділ

Кафедра комп'ютерної інженерії та інформаційних систем

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		21
Білі знаки		1
Парафрази (SmartMarks)		30

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

3.29% 3.29% КП 1

0.08% 0.08% КЦ

25
Довжина фрази для коефіцієнта подібності 2

19708
Кількість слів

160652
Кількість символів

ДОДАТОК Г

```
#include <iostream>
#include <string>
#include <vector>
#include <set>
#include <unordered_map>
#include <cstdlib>
#include <fstream>
#include <sstream>
#include <stdexcept>
#include <algorithm>
#include <chrono>
#include <thread>
#include <ctime>
#include <iomanip>


void inputNetwork(std::vector<std::string> &ips,
                  std::vector<bool>       &important)
{
    int N;
    std::cout << "Введіть кількість комп'ютерів у мережі: ";
    std::cin  >> N;

    ips.resize(N);
    for (int i = 0; i < N; ++i) {
        std::cout << "IP вузла #" << (i + 1) << ": ";
        std::cin  >> ips[i];
    }

    important.assign(N, false);

    int M;
    std::cout << "Введіть кількість важливих хостів: ";
    std::cin  >> M;
    std::cout << "Тепер введіть їх індекси (від 1 до " << N << "):\n";

    for (int j = 0; j < M; ++j) {
        int idx;
```

```
                std::cout << "  Індекс важливого хоста #" << (j + 1) << ": ";
                std::cin  >> idx;
                if (idx >= 1 && idx <= N) {
                    important[idx - 1] = true;
                } else {
                    std::cerr << "  Увага: індекс " << idx << " поза діапазоном,
пропускаємо.\n";
                }
            }
        }


    std::unordered_map<std::string,      double>      loadEPSSWeights(const
std::string &csvPath) {
            std::ifstream infile(csvPath);
            if (!infile.is_open()) {
                throw  std::runtime_error("Не  вдалося  відкрити  файл:  "  +
csvPath);
            }

            std::string line;
            bool headerSkipped = false;
            struct Entry { std::string cve; double epss; };
            std::vector<Entry> entries;

            while (std::getline(infile, line)) {
                std::stringstream ss(line);
                std::string cve, epssStr, percStr;
                if (!std::getline(ss, cve, ',') ||
                    !std::getline(ss, epssStr, ',') ||
                    !std::getline(ss, percStr, ',')) {
                    continue;
                }
                if (!headerSkipped && (cve == "CVE" || cve == "cve")) {
                    headerSkipped = true;
                    continue;
                }
                headerSkipped = true;
                try {
```

```
                double epss = std::stod(epssStr);
                entries.push_back({cve, epss});
            } catch (...) {}
        }
        infile.close();

        double sumEpss = 0.0;
        for (auto &e : entries) {
            sumEpss += e.epss;
        }
        if (sumEpss <= 0.0) {
            return {};
        }

        std::unordered_map<std::string, double> weights;
        weights.reserve(entries.size());
        for (auto &e : entries) {
            weights[e.cve] = e.epss / sumEpss;
        }
        return weights;
    }


    std::unordered_map<std::string,double>      runOpenVASAndLoadCvss(const
std::string &target) {
        std::system("openvas-start");

        std::string cmd = "openvas-scan --target=" + target + " --format=CSV
> openvas_report.csv";
        if (std::system(cmd.c_str()) != 0) {
            throw std::runtime_error("Помилка при запуску openvas-scan");
        }

        std::ifstream infile("openvas_report.csv");
        if (!infile.is_open()) {
            throw       std::runtime_error("Не       вдалося       відкрити
openvas_report.csv");
        }
```

```
        std::unordered_map<std::string,double> cvssMap;
        std::string line;
        bool header = true;
        while (std::getline(infile, line)) {
            if (header) { header = false; continue; }
            std::stringstream ss(line);
            std::string cve, cvssStr;
            if (!std::getline(ss, cve, ',')) continue;
            if (!std::getline(ss, cvssStr, ',')) continue;
            try {
                double cvss = std::stod(cvssStr);
                cvssMap[cve] = cvss;
            } catch (...) {}
        }

        return cvssMap;
    }


    struct Coeffs {
        double w_S;   // вага вразливості ПЗ
        double w_P;   // вага ефективності політик
        double w_U;   // вага людського фактора
        double P;     // ефективність політик (0..1)
        double U;     // ймовірність людського фактору (0..1)
    };


    std::vector<double> computeVVector(
        const std::vector<std::string>& ips,
        const Coeffs &c,
        const std::string &epssCsvPath)
    {
        auto epssMap = loadEPSSWeights(epssCsvPath);
        if (epssMap.empty()) {
            throw std::runtime_error("Не вдалося завантажити EPSS-ваги з "
+ epssCsvPath);
        }
```

```
        std::vector<double> V;
        V.reserve(ips.size());

        for (const auto &ip : ips) {
            auto cvssMap = runOpenVASAndLoadCvss(ip);
            if (cvssMap.empty()) {
                V.push_back(c.w_P * (1.0 - c.P) + c.w_U * c.U);
                continue;
            }

            double V_vuln = 0.0;
            for (auto &kv : cvssMap) {
                const auto &cve = kv.first;
                double cvss = kv.second;
                auto it = epssMap.find(cve);
                if (it != epssMap.end()) {
                    double omega_k = it->second;
                    V_vuln += omega_k * (cvss / 10.0);
                }
            }

            double Vi = c.w_S * V_vuln
                      + c.w_P * (1.0 - c.P)
                      + c.w_U * c.U;

            V.push_back(Vi);
        }

        return V;
    }


    double calculateTij(const std::string &targetIP, int maxAllowedPorts =
10) {
        static const std::set<int> excludedPorts = {443};

        std::string cmd = "nmap -p- --open -oG nmap_grep.txt " + targetIP +
" > /dev/null 2>&1";
        std::system(cmd.c_str());
```

```
        std::ifstream infile("nmap_grep.txt");
        if (!infile.is_open()) {
            throw std::runtime_error("Не вдалося відкрити nmap_grep.txt");
        }

        int openCount = 0;
        std::string line;
        while (std::getline(infile, line)) {
            auto pos = line.find("Ports:");
            if (pos == std::string::npos) continue;

            std::string portsList = line.substr(pos + 6);
            std::stringstream ss(portsList);
            std::string token;
            while (std::getline(ss, token, ',')) {
                std::stringstream ptok(token);
                std::string portStr;
                if (std::getline(ptok, portStr, '/')) {
                    int port = std::stoi(portStr);
                    if (excludedPorts.find(port) == excludedPorts.end()) {
                        ++openCount;
                    }
                }
            }
        }
        infile.close();

        double T = static_cast<double>(openCount) /
static_cast<double>(maxAllowedPorts);
        return std::min(T, 1.0);
    }


    std::vector<std::vector<double>> computeTMatrix(
        const std::vector<std::string>& ips,
        int maxAllowedPorts = 10)
    {
        int N = static_cast<int>(ips.size());
```

```
        std::vector<double> tValues(N);
        for (int j = 0; j < N; ++j) {
            tValues[j] = calculateTij(ips[j], maxAllowedPorts);
        }

        std::vector<std::vector<double>> T(N, std::vector<double>(N, 0.0));
        for (int i = 0; i < N; ++i) {
            for (int j = 0; j < N; ++j) {
                if (i == j) {
                    T[i][j] = 0.0;
                } else {
                    T[i][j] = tValues[j];
                }
            }
        }
        return T;
    }


    double calculateFij(const std::string &targetIP, int N_tests = 100) {
        std::string cmd =
            "hping3 -c " + std::to_string(N_tests) +
            " -S --fast " + targetIP +
            " > hping_result.txt 2>&1";
        if (std::system(cmd.c_str()) != 0) {
            throw std::runtime_error("Не вдалося виконати hping3");
        }

        std::ifstream infile("hping_result.txt");
        if (!infile.is_open()) {
            throw       std::runtime_error("Не       вдалося       відкрити
hping_result.txt");
        }

        int transmitted = 0, received = 0;
        std::string line;
        while (std::getline(infile, line)) {
            if (line.find("packets transmitted") != std::string::npos &&
                line.find("packets received")   != std::string::npos) {
```

```
                if (std::sscanf(
                        line.c_str(),
                        "%d packets transmitted, %d packets received",
                        &transmitted,
                        &received) == 2) {
                    break;
                }
            }
        }
        infile.close();

        if (transmitted <= 0) {
            throw    std::runtime_error("Невірна    кількість    переданих
пакетів");
        }

        double       F       =       static_cast<double>(received)       /
static_cast<double>(transmitted);
        return (F > 1.0) ? 1.0 : F;
    }


    std::vector<std::vector<double>> computeFMatrix(
        const std::vector<std::string>& ips,
        int N_tests = 100)
    {
        int N = static_cast<int>(ips.size());

        std::vector<double> fValues(N);
        for (int j = 0; j < N; ++j) {
            fValues[j] = calculateFij(ips[j], N_tests);
        }

        std::vector<std::vector<double>> F(N, std::vector<double>(N, 0.0));
        for (int i = 0; i < N; ++i) {
            for (int j = 0; j < N; ++j) {
                if (i == j) {
                    F[i][j] = 0.0;
                } else {
```

```
                    F[i][j] = fValues[j];
                }
            }
        }
        return F;
    }


    double calculateLij(const std::string &targetIP) {
        std::string cmd = "sslyze --regular " + targetIP + " > sslyze_result.txt 2>&1";
        if (std::system(cmd.c_str()) != 0) {
            throw std::runtime_error("Не вдалося виконати sslyze");
        }

        std::ifstream infile("sslyze_result.txt");
        if (!infile.is_open()) {
            throw std::runtime_error("Не вдалося відкрити sslyze_result.txt");
        }

        bool has1_3 = false, has1_2 = false, has1_1 = false;
        std::string line;
        while (std::getline(infile, line)) {
            if (line.find("TLS 1.3") != std::string::npos) {
                has1_3 = true;
            } else if (line.find("TLS 1.2") != std::string::npos) {
                has1_2 = true;
            } else if (line.find("TLS 1.1") != std::string::npos) {
                has1_1 = true;
            }
        }
        infile.close();


        if (has1_3)      return 1.0;
        else if (has1_2) return 0.7;
        else if (has1_1) return 0.3;
        else             return 0.0;
```

```
}

std::vector<std::vector<double>> computeLMatrix(
    const std::vector<std::string>& ips)
{
    int N = static_cast<int>(ips.size());
    std::vector<double> lValues(N);
    for (int j = 0; j < N; ++j) {
        lValues[j] = calculateLij(ips[j]);
    }

    std::vector<std::vector<double>> L(N, std::vector<double>(N, 0.0));
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j) {
            if (i == j) {
                L[i][j] = 0.0;
            } else {
                L[i][j] = lValues[j];
            }
        }
    }
    return L;
}


std::vector<std::vector<double>> computeGMatrix(
    const std::vector<std::vector<double>> &T,
    const std::vector<std::vector<double>> &F,
    const std::vector<std::vector<double>> &L)
{
    size_t N = T.size();
    std::vector<std::vector<double>> G(N, std::vector<double>(N, 0.0));

    for (size_t i = 0; i < N; ++i) {
        for (size_t j = 0; j < N; ++j) {
            if (i == j) {
                G[i][j] = 0.0;
            } else {
                G[i][j] = T[i][j] * (1.0 - F[i][j]) * (1.0 - L[i][j]);
```

```
            }
        }
    }
    return G;
}


double computeCS(
    const std::vector<double>           &V,
    const std::vector<std::vector<double>> &G,
    const std::vector<bool>             &important)
{
    size_t N = V.size();
    double CS = 1.0;
    for (size_t i = 0; i < N; ++i) {
        if (!important[i]) continue;
        for (size_t j = 0; j < N; ++j) {
            if (i == j) continue;
            CS *= (1.0 - V[i] * G[i][j]);
        }
    }
    return CS;
}

int main() {
    std::vector<std::string> ips;
    std::vector<bool> important;
    inputNetwork(ips, important);

    Coeffs coeffs;
    inputCoefficients(coeffs);

    std::string epssCsvPath;
    std::cout << "Введіть шлях до CSV-файлу EPSS: ";
    std::cin  >> epssCsvPath;

    const int hour_seconds = 3600;
    std::ofstream logFile("cs_log.txt", std::ios::app);
    if (!logFile.is_open()) {
```

```
            std::cerr << "Не вдалося відкрити cs_log.txt для запису\n";
            return 1;
        }

        while (true) {
            auto V = computeVVector(ips, coeffs, epssCsvPath);
            auto T = computeTMatrix(ips, 10);
            auto F = computeFMatrix(ips, 100);
            auto L = computeLMatrix(ips);
            auto G = computeGMatrix(T, F, L);

            double CS = computeCS(V, G, important);

            auto now = std::chrono::system_clock::now();
            std::time_t t_now = std::chrono::system_clock::to_time_t(now);
            logFile << std::put_time(std::localtime(&t_now), "%F %T")
                    << "  CS = " << CS << "\n";
            logFile.flush();


std::this_thread::sleep_for(std::chrono::seconds(hour_seconds));
        }

        return 0;
}
```

Завідувачу кафедри КІІС,
доктору філософії, доц. Ользі ПАВЛОВІЙ

Ігор РАМСЬКИЙ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25 квітня 2025 року

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Ігор РАМСЬКИЙ

Співавтор:

Назва: РАМСЬКИЙ_Система оцінювання кібербезпеки корпоративних мереж

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3.3%

Коефіцієнт подібності 2:1.2%

Мікропробіли: 21

Заміна букв: 0

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-04-15 13:27:42.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-04-15 Доцент Андрій Нічепорук

Дата експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система оцінювання кібербезпеки корпоративних мереж

Автор: Ігор РАМСЬКИЙ

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Світлана САЧЕНКО, к.е.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
2) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає менше 6% і адресується до джерел з інтернету та бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру завдання і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Світлана САЧЕНКО

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Ігор Рамський

Тема: Система оцінювання кібербезпеки корпоративних мереж

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень _-__; кількість сторінок записки 108_

1. Короткий зміст роботи та прийнятих рішень _У роботі розроблено метод створення систем оцінювання кібербезпеки корпоративних мереж

2. Висновок про відповідність роботи дипломному завданню _____
Кваліфікаційна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи.

У першому розділі проведено аналіз відомих рішень щодо створення, захисту та підтримки розподілених комп'ютерних систем.

У другому розділі розроблені архітектурні рішення для створення системи оцінювання кібербезпеки корпоративних мереж.

У третьому розділі розроблені функції оцінювання кібербезпеки корпоративних мереж та комп'ютерних станцій в них.

У четвертому розділі розроблений метод синтезу системи оцінювання кібербезпеки корпоративних мереж, проведено експеримент та моделювання роботи системи.

У висновках представлено підсумкову оцінку виконання завдань, сформульованих у ході дослідження.

4. Позитивні сторони роботи: ___

5. Негативні сторони роботи: ___немає.

6. Оцінка графічного оформлення та пояснювальної записки роботи: __-____

7. Відгук про роботу в цілому: Робота виконана на належному рівні.

8. Інші зауваження: –

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Корецька Людмила Олександрівна, к.т.н., доцент кафедри АКІТР ХНУ

"2" травня 2025р.