Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ БЕЗПЕКИ З ВИКОРИСТАННЯМ HONEYPOTS

SOFTWARE IMPLEMENTATION AND RESEARCH OF A SECURITY SYSTEM USING HONEYPOTS

спеціальність 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КНм-21
Веремій Ілля Ігорович

_____
(підпис)

Керівник: к.т.н., доцент
Кошелюк Віктор Андрійович

_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Валерій Олександрович

_____
(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерних наук*
Ступінь вищої освіти: *магістр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *122 Комп'ютерні науки*
Освітня програма: *«Комп'ютерні науки»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Валерій ЛІЩИНА
«14» травня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА
ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Веремій Ілля Ігорович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Програмна реалізація та дослідження системи безпеки з використанням honeypots»
Керівник к.т.н., доцент Кошелюк Віктор Андрійович
затверджені наказом закладу вищої освіти від «*14*» *травня* *2025* р. № 255/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *«05» грудня 2025 р.*

3. Вихідні дані до роботи: _____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, експерементальне дослідження результативності предмету дослідження.
_____

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз проблематики за темою роботи та постановка завдань дослідження	*Кошелюк В.А.*		
Теоретичне дослідження та практична реалізація предмету дослідження	*Кошелюк В.А.*		
Експерементальне дослідження результативності предмету дослідження	*Кошелюк В.А.*		
Показник запозичень тексту	_____ %		
Інструментальна перевірка	*Кошелюк В. А.*		
Нормоконтроль	*Сачук В. О.*		
Гарант ОПП	*Ліщина В. О.*		

7. Дата видачі завдання *«14» травня 2025 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	*Провести огляд літературних джерел по темі кваліфікаційної роботи*	*до 30.06.2025 р*	
2	*Провести аналіз загальної проблеми і вибір напрямків дослідження*	*до 01.09.2025 р.*	
3	*Розробити функціональну схему роботи програмного продукту*	*до 01.10.2025 р*	
4	*Описати засоби розробки об'єкта проектування*	*до 15.10.2025 р.*	
5	*Практична реалізація об'єкта проектування*	*до 10.11.2025 р.*	
6	*Провести експерементальне дослідження результативності предмету дослідження*	*до 25.11.2025 р.*	
7	*Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру*	*до 05.12.2025 р.*	

Здобувач вищої освіти _____ Ілля ВЕРЕМІЙ

Керівник роботи _____ Віктор КОШЕЛЮК

АНОТАЦІЯ

Веремій І. І. Програмна реалізація та дослідження системи безпеки з використанням Honeypots. Рукопис. Кваліфікаційна робота магістра за спеціальністю 122 Комп'ютерні науки. Луцький національний технічний університет. Луцьк, 2025. 60 с.

Кваліфікаційна робота магістра складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

У роботі досліджено та проведено аналіз використання властивостей blockchain для реалізації системи безпеки з використанням honeypots. Під час виконання поставлених завдань було проаналізовано методи та засоби безпеки з інтеграцією honeypots, досліджено технології використання blockchain для підвищення рівня інформаційної безпеки, здійснено конфігурацію та реалізацію динамічної системи безпеки інформації з використанням blockchain.

Ключові слова: honeypots, blockchain, безпека, виявлення загроз, ELK stack.

ANNOTATION

Ilya Veremiy. Software implementation and research of a security system using Honeypots. Manuscript. Master's Qualification Thesis in the field of 122 Computer Science. Lutsk National Technical University, 2025. 60 pages.

The master's thesis consists of an introduction, three sections, conclusions, a list of used sources, appendices.

The paper investigates and analyzes the use of blockchain properties for implementing a security system using honeypots. In the course of the research, security methods and tools with honeypot integration were analyzed, technologies for using blockchain to improve information security were investigated, and a dynamic information security system using blockchain was configured and implemented.

Keywords: honeypots, blockchain, security, threat detection, ELK stack.

ЗМІСТ

ВСТУП

Актуальність дослідження. У сучасному цифровому середовищі питання захисту мережевих систем набуває критичного значення. Організації зберігають цінні дані та ресурси, які потребують надійного захисту від кіберзагроз. Одним із ефективних інструментів забезпечення безпеки є технологія honeypots.

Honeypot являє собою спеціально сконфігурований ресурс інформаційної системи, призначений для приваблення та дослідження дій потенційних зловмисників. Згідно з визначенням проекту Honeynet, це «ресурс безпеки, цінність якого полягає у його дослідженні, атаці або компрометації». Фактично, honeypot функціонує як імітація реальної системи, розміщена в мережі для залучення атакуючих.

Технічно honeypots реалізуються як віртуальні машини, що емулюють справжні сервери з активними службами, відкритими портами та додатками, характерними для типових мережевих систем. Вони створюють переконливу ілюзію цінних цілей для хакерів.

Основна стратегія використання honeypots полягає у перенаправленні уваги зловмисників з критичної IT-інфраструктури організації на контрольовані пастки. Коли атакуючий виявляється і взаємодіє з honeypot, він витрачає час та ресурси на некритичну систему, що дозволяє фахівцям з безпеки аналізувати методи атак та розробляти відповідні контрзаходи. Таким чином, інтеграція honeypot-систем із блокчейном створює нову концепцію інтелектуального захисту, де інформація про інциденти безпеки фіксується у децентралізованому середовищі, що унеможливлює її підробку чи видалення.

Проте honeypots не повинні розглядатися як єдиний захисний механізм. Організаціям необхідно інтегрувати їх у комплексну стратегію кібербезпеки, що включає впровадження кращих практик захисту, розробку політик безпеки та постійне вдосконалення IT-інфраструктури для забезпечення багаторівневого захисту від сучасних кіберзагроз.

Об’єктом дослідження є процеси виявлення, аналізу та протидії кіберзагрозам у інформаційних системах з використанням технологій honeypot та блокчейн.

Предметом дослідження є методи та програмні засоби інтеграції honeypot-систем із блокчейн-технологіями з метою підвищення рівня безпеки, достовірності збереження даних про атаки та неможливості їх модифікації зловмисниками.

Метою магістерської роботи є розробка та дослідження програмної систему кібербезпеки, що базується на використанні honeypot-технологій у поєднанні з технологією блокчейн для забезпечення підвищеного рівня захисту інформаційних систем від кібератак, а також для створення прозорого та достовірного механізму фіксації інцидентів безпеки.

Завдання дослідження. Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати сучасні підходи до виявлення та моніторингу кібератак із використанням honeypot-технологій;

– дослідити можливості застосування технології блокчейн для збереження та перевірки достовірності журналів подій безпеки;

– розробити архітектуру програмної системи, що поєднує honeypot і блокчейн для фіксації атак у незмінному вигляді;

– реалізувати прототип системи та провести експериментальне дослідження її ефективності;

– проаналізувати результати експериментів та порівняти запропонований підхід із традиційними системами безпеки;

– сформулювати висновки та рекомендації щодо подальшого вдосконалення систем захисту з використанням honeypot та блокчейн-технологій.

Методи дослідження. У процесі роботи планується використати методи аналізу та синтезу для вивчення існуючих підходів до виявлення атак і збереження журналів безпеки; моделювання для створення архітектури системи

з інтеграцією honeypot і блокчейн; методи програмної інженерії для реалізації та тестування прототипу програмної системи.

Наукова новизна роботи полягає у реалізації вдосконаленого підходу до фіксації подій інформаційної безпеки шляхом їх збереження у розподіленому блокчейн-реєстрі, що унеможливлює підробку або видалення записів. Поряд з тим запропоновано інтегровану архітектуру honeypot-блокчейн системи, яка поєднує моніторинг атак і децентралізовану фіксацію даних.

Практична цінність дослідження полягає у створенні прототипу, що може бути використаний для навчальних, дослідницьких або промислових цілей у сфері інформаційної безпеки. Розроблена система дозволяє автоматизовано фіксувати кібератаки, забезпечуючи незмінність та надійність журналів подій. Запропоновані рішення можуть бути інтегровані у SIEM-системи, SOC-платформи або інші інструменти моніторингу безпеки.

Апробація результатів дослідження:

– IX Міжнародна науково-практична конференція «Development of science: theories, methodology, practice and technologies» (28-31 жовтня 2025 р.), Париж, Франція. International Science Group. 2025 [1].

– 2 Міжнародна науково-практична конференція «Modern Perspectives on Science and Economic Progress» (5-7 листопада, 2025 р.), Вільнюс, Литва. International Scientific Unity. 2025 [2].

– Андрущак І., Кошелюк В., Веремій, І. Технології обману в кібербезпеці: інтеграція cowrie та ELK Stack для виявлення атак на мережевий трафік. International Science Journal of Engineering and Agriculture. ISJEA, 2025. Pp. 1-14. [3].

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМАТИКИ ВИКОРИСТАННЯ HONEYPOTS ТА ОГЛЯД НАЯВНИХ РІШЕНЬ

1.1 Огляд і аналіз предметної області проблеми, результати існуючих теоретичних та експериментальних досліджень

Сучасний цифровий світ переживає безпрецедентну епоху interconnectivity – до глобальної мережі Інтернет сьогодні під'єднано набагато більше різноманітних систем, пристроїв та обладнання, ніж це було на будь-якому попередньому етапі розвитку інформаційних технологій. Смартфони, планшети, розумні побутові прилади, промислові контролери, медичне обладнання, автомобільні системи та численні IoT-пристрої формують величезну екосистему взаємопов'язаних об'єктів.

Однак цією стрімко зростаючою кількістю технологічних рішень дедалі частіше користуються звичайні користувачі, які не мають глибоких технічних знань, спеціалізованої освіти чи професійного досвіду в галузі інформаційних технологій. Більше того, рівень їхньої обізнаності щодо потенційних загроз кібербезпеці, можливих векторів атак та необхідних превентивних заходів захисту виявляється ще нижчим. Багато людей просто не усвідомлюють масштабів ризиків, які супроводжують їхню щоденну цифрову активність.

Ситуація ускладнюється тим, що безпекові налаштування в більшості сучасних систем часто встановлюються неправильно, залишаються на заводських параметрах або взагалі ігноруються користувачами. Це створює сприятливе середовище, де сучасний Інтернет перетворюється на ідеальний полігон для кіберзлочинців – своєрідний майданчик з практично необмеженою кількістю вразливих пристроїв, які можуть бути легко зламані, скомпрометовані та використані в злочинних схемах.

За таких обставин надзвичайно актуальним та критично важливим стає питання своєчасного виявлення й ефективної протидії широкому спектру кіберзагроз. Це стосується як давно відомих, класичних методів злому та

перевірених часом вірусів, так і найновіших, ще невивчених типів атак, інноваційних методик проникнення, складного шкідливого програмного забезпечення нового покоління та інших форм зловмисної діяльності в кіберпросторі, що постійно еволюціонують.

Сучасні кібератаки демонструють стійку тенденцію до зростання як за кількістю інцидентів, так і за різноманітністю застосовуваних методів. Цей феномен безпосередньо пов’язаний із безперервним розвитком та імплементацією інноваційних технологічних рішень у цифровому просторі. Парадоксально, але ефективність більшості кібератак визначається не стільки високою технічною кваліфікацією зловмисників, скільки недостатнім рівнем цифрової грамотності та неуважністю потенційних жертв до питань інформаційної безпеки.

Детальний аналіз успішних кібератак демонструє, що значна частина з них не відзначалася особливою технічною складністю або використанням витончених методів проникнення. Навпаки, переважна більшість інцидентів відбувалася через експлуатацію загальновідомих вразливостей у системах безпеки, які залишалися невиправленими через недбалість або відсутність своєчасного оновлення програмного забезпечення.

За характером спрямованості кібератаки поділяються на цілеспрямовані та ненавмисні. Останні характеризуються тим, що жертва не є конкретною метою зловмисників, а потрапляє під удар випадково, часом стаючи одним із численних об’єктів масованої атаки. Найбільш розповсюдженим мотивом для здійснення ненавмисних кібератак виступає соціальна інженерія – психологічна маніпуляція, спрямована на отримання конфіденційної інформації. Класичним прикладом такого підходу є фішингові атаки через електронну пошту, коли зловмисники видають себе за легітимні організації.

Структура та послідовність етапів реалізації кібератаки варіюються залежно від комплексу чинників: первинної мотивації нападників (фінансова вигода, шпіонаж, саботаж), профілю атакуючого (індивідуальний хакер, організована злочинна група, державні структури) та специфічних

характеристик обраного типу атаки. Розуміння цих аспектів є критично важливим для розробки ефективних стратегій кіберзахисту.

Для ефективного протидіяння сучасним кіберзагрозам та забезпечення комплексного захисту інформаційних систем розроблено спеціалізовані технології, серед яких особливе місце посідають системи-пастки, відомі в міжнародній практиці як honeypots (медові пастки). Ці системи являють собою високотехнологічні рішення, побудовані на базі спеціалізованого програмного забезпечення, яке створює привабливі для потенційних зловмисників цілі в корпоративній мережі або окремих сегментах інфраструктури.

Принцип роботи honeypots полягає у створенні контрольованого середовища, що імітує реальні виробничі системи, сервіси чи дані, проте насправді призначене виключно для виявлення, документування та детального аналізу хакерських атак. Ці системи володіють потужними можливостями логування та моніторингу, що дозволяє фахівцям з кібербезпеки в режимі реального часу відстежувати всі етапи несанкціонованого проникнення – від початкового сканування портів до спроб отримання привілейованого доступу та викрадення інформації.

Відомий фахівець у галузі інформаційної безпеки та засновник міжнародного проєкту Honeynet Project Ленс Шпіцнер надає таке визначення концепції: Honeypot – це «безпековий ресурс, основна цінність якого полягає саме в тому, аби стати об’єктом дослідження, атаки або компрометації з боку зловмисників» [4].

У своїй основі технологія Honeypots реалізує інноваційну стратегію кібер-обману (cyber deception). Як детально описано в дослідженні Кліффа та колег, фундаментальні компоненти механізму кібер-обману включають навмисно підготовлені захисниками інформаційні активи та дані, які спеціально розроблені для дезінформації атакуючих, а також помилкові дії та рішення, які приймає зловмисник внаслідок впливу цього обману [5]. Така методологія дозволяє не лише виявляти загрози, але й вивчати тактику, техніки та процедури

хакерів для подальшого вдосконалення систем захисту. На рисунку 1.1 проілюстровано частину мережевого сегменту з програмною приманкою.

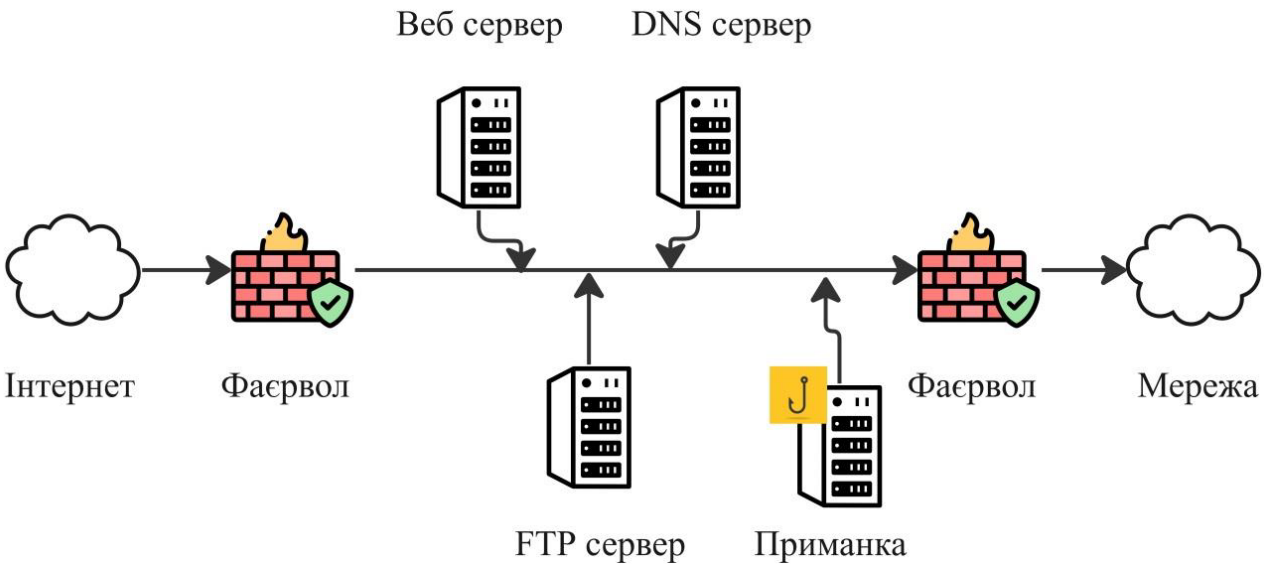

Рисунок 1.1 – Мережевий сегмент з honeypot [6]

Honeypots являють собою спеціалізовані системи кібербезпеки, які функціонують як своєрідні цифрові пастки для потенційних зловмисників. По суті, це ретельно сконструйовані імітації реальних комп'ютерних систем, серверів або мережевих пристроїв, які навмисно створюються для залучення уваги кіберзлочинців. Основне призначення таких систем полягає у виявленні, моніторингу та детальному аналізі тактики, методів і стратегій, які використовують хакери під час спроб несанкціонованого проникнення в мережеву інфраструктуру.

На відміну від пасивних засобів захисту, таких як файрволи чи антивірусне програмне забезпечення, honeypots відносяться до категорії активних превентивних механізмів безпеки. Їхня унікальна особливість полягає в тому, що вони спеціально розгортаються з єдиною метою – стати об'єктом кібератаки. Ці системи не містять жодних критично важливих даних і не використовуються для легітимних бізнес-процесів, тому будь-яка взаємодія з ними автоматично розглядається як потенційно шкідлива активність.

Критичним фактором ефективності honeypot є його здатність бути максимально привабливим для зловмисників. Концепція «привабливості» у даному контексті має багатовимірний характер. По-перше, система повинна точно імітувати цільові ресурси, які традиційно цікавлять хакерів – наприклад, сервери баз даних, платіжні системи або сховища конфіденційної інформації. По-друге, honeypot має демонструвати ознаки уразливості, створюючи ілюзію легкої можливості для експлуатації – будь то застарілі версії програмного забезпечення, слабкі паролі чи невідповідні налаштування безпеки. По-третє, система повинна пропонувати зловмисникові уявну високу цінність – доступ до «цінних» даних або можливість використання ресурсів для подальших атак, що мотивує кіберзлочинця витрачати час та зусилля на спробу компрометації саме цього об’єкта.

Ступінь інтерактивності пасток-приманок (honeypots) відіграє критичну роль у визначенні можливостей взаємодії зловмисників із системою та, як наслідок, впливає на обсяг і характер інформації, що може бути отримана в процесі моніторингу. Діапазон взаємодії охоплює широкий спектр – від елементарного встановлення мережевого з’єднання до надання повноцінної можливості завантаження, інсталяції та виконання шкідливого програмного забезпечення в контрольованому середовищі.

Важливо розуміти, що між вартістю впровадження та ефективністю роботи honeypot існує пряма пропорційна залежність. Це означає, що високоінтерактивні системи-пастки потребують значних фінансових інвестицій не лише на етапі первинного розгортання, але й для подальшого технічного обслуговування, моніторингу та регулярного оновлення.

Для забезпечення ефективного функціонування будь-яка система-приманка повинна володіти двома ключовими характеристиками: здатністю підтримувати певний рівень взаємодії з потенційним атакуючим та можливістю непомітного спостереження за всіма його діями в режимі реального часу. У сучасній практиці кібербезпеки виділяють три основні класифікаційні категорії інтерактивності honeypots: низькоінтерактивні (low-interaction),

високоінтерактивні (high-interaction) та середньоінтерактивні (medium-interaction) системи, кожна з яких має свої специфічні переваги та обмеження [7].

Впровадження технології Honeynet створює унікальну можливість для розгортання спеціалізованої мережевої інфраструктури активного захисту з високим рівнем контролю та моніторингу. Ця система функціонує як своєрідна пастка, де кожна дія потенційного зловмисника, кожна спроба проникнення та всі тактики атаки стають повністю прозорими та доступними для детального аналізу безпековими фахівцями. Інформація, яку можна зібрати, спостерігаючи за реальною поведінкою хакерів у спеціально підготовленому, але водночас необмеженому мережевому середовищі, має виняткову практичну цінність. Ці дані допомагають організаціям, які прагнуть вдосконалювати свої механізми кібербезпеки, передбачати нові загрози та розробляти ефективні захисні заходи

Проте застосування Honeynet супроводжується суттєвими ризиками, які не можна ігнорувати. Головним недоліком є те, що така система за своєю природою містить навмисно створені вразливості. Це призводить до значного збільшення потенційних точок компрометації та відмови системи. Якщо зловмисник зможе використати ці вразливості для виходу за межі контрольованого середовища, наслідки можуть бути катастрофічними для всієї корпоративної мережі, перетворюючи захисний інструмент на джерело загрози. Питання етичності застосування приманок у кібербезпеці викликає чимало дискусій серед фахівців. Проте переважна більшість експертів дійшла згоди, що використання honeypot-технологій є морально виправданим, враховуючи їхнє призначення та контекст впровадження [8]. Ключовим аргументом на користь етичності приманок виступає їхня первинна мета – забезпечення захисту інформаційних систем від злочинної діяльності. Honeypot створюються виключно для виявлення, аналізу та протидії протиправним діям хакерів, які самі порушують закон та етичні норми. Таким чином, застосування приманок розглядається як легітимний оборонний механізм, спрямований на запобігання кіберзлочинам та мінімізацію шкоди від атак зловмисників, що робить їх використання етично обґрунтованим у контексті інформаційної безпеки.

1.2 Огляд і аналіз методів та засобів безпеки з використанням honeypots для вирішення проблеми дослідження

Honeypot по праву вважається першим втіленням та фундаментальною основою технології Deception. Ці новаторські рішення з'явилися ще наприкінці вісімдесятих – на початку дев'яностих років минулого століття, започаткувавши революційний підхід до виявлення кіберзагроз. За своєю суттю, Honeypot являє собою спеціально створений мережевий об'єкт або систему-пастку, єдиною та головною метою якого є привернути увагу потенційного зловмисника та спровокувати його на проведення атаки.

Коли кіберзлочинець здійснює атаку на Honeypot, система автоматично реєструє цю подію та детально фіксує всі дії, техніки та методи, які використовує зловмисник. Зібрана інформація надалі стає цінним джерелом даних для безпекових аналітиків, дозволяючи детально вивчити тактику атакуючого, його інструментарій та траєкторію руху в мережі [9]. Друга важлива функція Honeypot полягає у затримці та уповільненні просування зловмисника корпоративною інфраструктурою, змушуючи його витрачати дорогоцінний час на дослідження та експлуатацію підробленого, безпечного ресурсу.

Проте традиційні Honeypots мають суттєві обмеження. Вони не здатні ефективно протидіяти фішинговим атакам так само природно, як це роблять реальні користувачі, тому не можуть спровокувати та виявити атаки через цей поширений вектор проникнення. На відміну від класичних Honeypots, технології обману наступного покоління демонструють значно вищу ефективність. Вони здатні автоматично та динамічно змінювати середовище приманки, імітуючи природну поведінку реальної корпоративної мережі, де дані користувачів та конфігурації систем постійно оновлюються. Сучасні deception-технології виявляють присутність зловмисника надзвичайно швидко – всього за три-чотири кроки його руху в інфраструктурі, навіть без розгортання елементів обману на кожному мережевому вузлі [10].

Інноваційні технології обману нового покоління революціонізують підходи до кібербезпеки, надаючи організаціям унікальні можливості для захисту критичної інфраструктури. Ці передові рішення забезпечують миттєве виявлення кіберзагроз у режиму реального часу, супроводжуване детальним криміналістичним аналізом та збором цифрових доказів. Ключовою перевагою таких систем є мінімальна кількість хибних спрацьовувань, що суттєво підвищує ефективність служб безпеки. Найважливішою характеристикою залишається непомітність для зловмисників – атакуючі не усвідомлюють, що перебувають під спостереженням фахівців з кібербезпеки. На рисунку 1.2 наведено типова схема технології обману, що використовується в інформаційних системах.

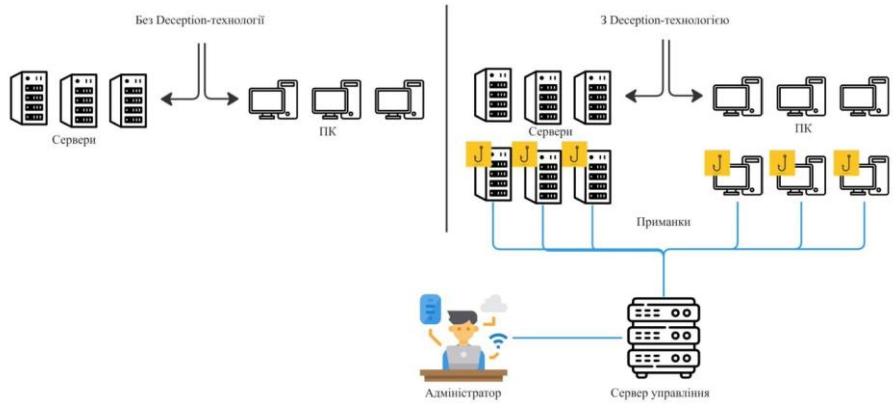

Рисунок 1.2 – Типова схема deception technologies [11]

Традиційні honeypot-системи, хоча й демонструють певну ефективність у виявленні шкідливої активності, значно поступаються сучасним deception-технологіям за багатьма параметрами. Класичні приманки характеризуються обмеженими можливостями детектування реальних загроз, генерують надмірну кількість помилкових тривог та не забезпечують глибокого криміналістичного аналізу справжніх вузлів мережі, через які зловмисники здійснюють компрометацію інфраструктури.

Deception-технології належать до класу Intrusion Detection System (IDS) – систем виявлення несанкціонованого проникнення. Їхнє фундаментальне призначення полягає в ідентифікації та блокуванні небажаних спроб отримання

доступу до корпоративної мережі. Якщо honeypot являє собою ізольований мережевий ресурс, пасивно очікуючий атаки для фіксації дій порушника, то deception представляє централізовану платформу управління множиною фальшивих мережевих об’єктів – так званих пасток або приманок, які створюють комплексну екосистему для виявлення та аналізу кіберзагроз.

Одним з перспективних напрямків посилення безпеки honeypot-систем є інтеграція технології Blockchain. Blockchain являє собою розподілену децентралізовану мережу, яка може об’єднувати мільйони користувачів по всьому світу, створюючи надійну інфраструктуру для зберігання та обміну інформацією. Кожен учасник такої мережі має можливість додавати нові дані до Blockchain, при цьому вся інформація надійно захищається за допомогою сучасних криптографічних алгоритмів, що практично унеможливлює її несанкціоноване змінення чи підробку [12].

Ключовою особливістю Blockchain є механізм колективної верифікації даних, коли кожен учасник мережі бере на себе відповідальність за перевірку достовірності інформації, яка додається до ланцюга. Цей процес реалізується через систему асиметричного шифрування з використанням трьох типів ключів: приватного, публічного та ключа одержувача. Така криптографічна архітектура дозволяє учасникам не лише перевіряти автентичність даних, але й однозначно ідентифікувати їх джерело, забезпечуючи повну прозорість походження інформації. На рисунку 1.3 продемонстровано шифрування передачі даних в Blockchain.

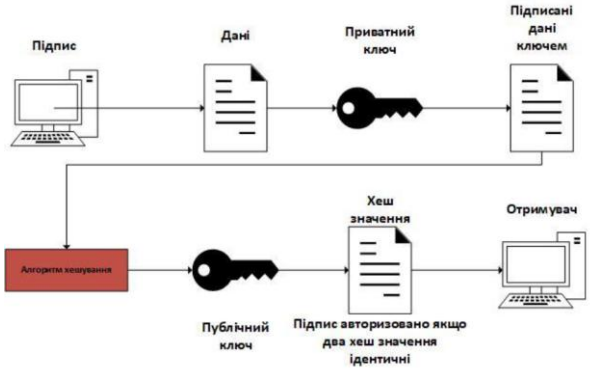

Рисунок 1.3 – Шифрування передачі даних в Blockchain [13]

Найважливішою перевагою технології Blockchain є її повна незалежність від централізованих посередників або довірених третіх сторін. Завдяки децентралізованій та розподіленій архітектурі, система функціонує автономно. Кожен учасник мережі володіє унікальним приватним ключем, який використовується для створення цифрових підписів та авторизації транзакцій. Така модель забезпечує високий рівень безпеки honeypot-систем, роблячи їх стійкими до атак та маніпуляцій з боку зловмисників.

Принцип оновлення інформації в блокчейн-технології базується на створенні незмінного ланцюга взаємопов'язаних блоків даних. Коли власник інформації потребує оновити певний фрагмент, він не змінює існуючий блок, а натомість додає новий блок поверх попереднього, формуючи унікальний криптографічний ланцюжок. Кожен блок містить спеціальний код, який математично пов'язаний з попереднім блоком, створюючи безперервну послідовність записів.

Найважливішою особливістю цієї системи є її надзвичайна чутливість до будь-яких змін. Навіть найменша модифікація даних – зміна одного символу, коми чи пробілу – миттєво призводить до зміни криптографічного коду не лише в конкретному блоці, а й у всіх наступних блоках мережі. Це створює ефект доміно, який автоматично поширюється по всьому ланцюгу.

Така архітектура забезпечує абсолютну прозорість та відстежуваність. Кожна транзакція, кожна зміна, кожне оновлення залишають постійний цифровий слід у системі. Жодна інформація не може бути безслідно видалена чи втрачена, оскільки користувачі завжди мають можливість повернутися до попередніх версій блоків і порівняти їх з поточними, визначивши точні відмінності між версіями.

Ця метикулозна система ведення записів надає блокчейну унікальні переваги в питаннях безпеки. Система автоматично виявляє блоки з помилковими, сфальсифікованими або пошкодженими даними, оскільки такі блоки не відповідатимуть криптографічним правилам ланцюга. Це ефективно

запобігає несанкціонованим змінам, втраті інформації та корупції даних, роблячи блокчейн надійною технологією для зберігання критично важливої інформації.

Однією з фундаментальних особливостей технології Blockchain є можливість для користувачів зберігати повну копію всієї мережі безпосередньо на власних комп'ютерах. Ця практика, яку активно використовує значна частина учасників блокчейн-спільноти, створює унікальну екосистему розподіленого зберігання даних із численними перевагами.

Такий підхід до зберігання інформації приносить користувачам подвійну вигоду. З одного боку, вони отримують можливість монетизувати свої обчислювальні ресурси, здаючи в оренду невикористаний простір на жорсткому диску та забезпечуючи тим самим додатковий дохід. З іншого боку, кожен учасник, який зберігає копію ланцюга блоків, автоматично стає гарантом цілісності та безперервності функціонування всієї системи, що робить мережу надзвичайно стійкою до збоїв та атак.

Система захисту Blockchain побудована на принципі консенсусу та колективної верифікації. Коли відбувається спроба несанкціонованого втручання – наприклад, зловмисник намагається підробити або змінити дані в окремому блоці – мережа автоматично ініціює процес перевірки. Система порівнює кожен блок із тисячами його копій, розподілених серед учасників мережі, шукаючи розбіжності. Блок, що відрізняється від більшості інших копій, негайно ідентифікується як скомпрометований і відхиляється системою, що унеможливлює його інтеграцію в ланцюг.

Принципова архітектура Blockchain побудована на відсутності централізованого управління або єдиного сховища даних. Натомість кожен учасник мережі виконує критично важливу функцію зі збереження повної або часткової копії блокчейну. Усі користувачі спільно відповідають за валідацію інформації, що зберігається або передається в системі. Це створює надійний механізм захисту, який унеможливлює додавання фальсифікованих даних та запобігає видаленню існуючої верифікованої інформації, забезпечуючи

незмінність історії транзакцій. На рисунку 1.4 представлено загальну структуру та організацію блоків в blockchain.

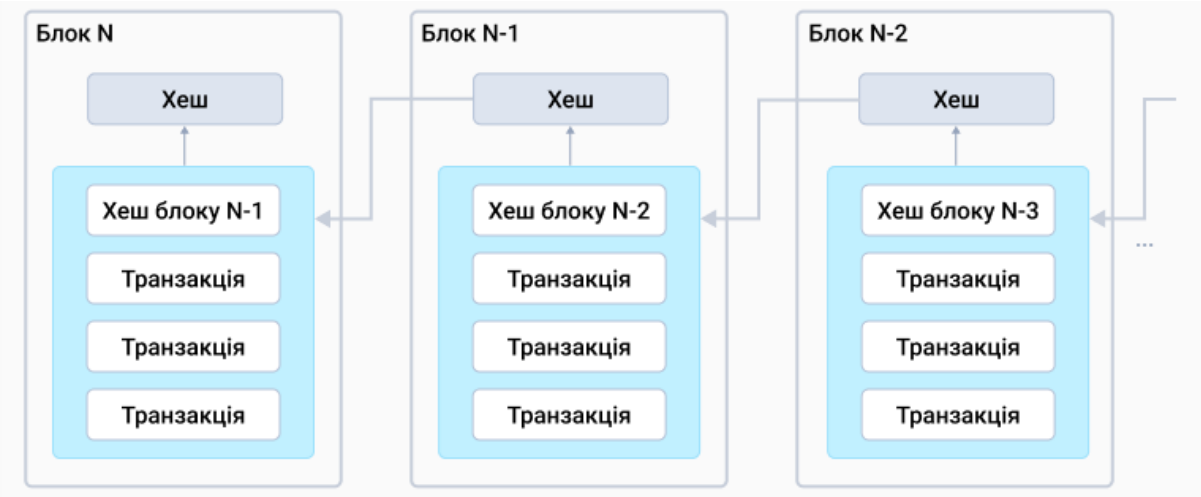

Рисунок 1.4 – Приклад загальної структури та організації блоків [14]

Технологія blockchain структурує дані у вигляді послідовних блоків, що формують надійний розподілений реєстр. Кожен окремий блок являє собою контейнер для зберігання певного обсягу інформації, і коли його ємність вичерпується, система автоматично створює наступний блок. Таким чином формується безперервний ланцюжок взаємопов'язаних блоків, звідки й походить назва технології.

Ключовою особливістю blockchain є використання криптографічних хешів – унікальних цифрових ідентифікаторів для кожного блоку. Хеш представляє собою складну комбінацію цифр та літер, яка може досягати 64 символів у довжину. Як зазначає Вікас Агарвал [15], партнер консалтингової практики PwC у галузі фінансових послуг, це унікальний код, що дозволяє елементам системи органічно поєднуватися один з одним, наче частини складної мозаїки.

Функціональність хешування виконує подвійну роль у забезпеченні безпеки blockchain. По-перше, криптографічний хеш надійно захищає інформацію всередині блоку від несанкціонованого доступу – лише користувачі з відповідними ключами можуть отримати доступ до даних. По-друге, хеш гарантує цілісність усього ланцюжка, оскільки кожен блок містить посилання на

хеш попереднього блоку. Це створює міцний зв’язок між блоками та робить практично неможливою зміну історичних записів без виявлення.

Одна з найважливіших характеристик технології Blockchain полягає в тому, що після внесення інформації до мережі та її криптографічного захисту через хеш-функції, ці дані набувають властивості незмінності та постійності. Кожен учасник мережі, або вузол, зберігає повну копію всієї історії транзакцій від самого початку створення блокчейну, що забезпечує прозорість та надійність системи.

Цей розподілений принцип зберігання створює потужний захисний механізм. Якщо зловмисник спробує скомпрометувати один окремий комп’ютер у мережі та внести несанкціоновані зміни до даних заради власної вигоди, це не вплине на інформацію, збережену на тисячах інших вузлів. Підроблений запис миттєво виявляється, оскільки він не співпадає з версією, що зберігається у більшості учасників мережі. Завдяки консенсусному механізму, хибні дані автоматично відхиляються, а правильна інформація відновлюється.

Крім того, сама архітектура системи робить її практично непроникною для атак. Обчислювальна потужність, необхідна для зворотного інжинірингу криптографічних хеш-функцій або для одночасного компрометування більшості вузлів мережі, є астрономічно великою. Зловмиснику довелося б контролювати понад 51% усієї обчислювальної потужності мережі, що в випадку великих блокчейнів, таких як Bitcoin або Ethereum, вимагає неймовірних фінансових та технічних ресурсів. Саме тому технологія блокчейн вважається однією з найбезпечніших систем зберігання даних у сучасному цифровому світі, забезпечуючи надійність через децентралізацію та криптографічний захист.

Технологія блокчейн сьогодні вважається одним із найнадійніших та найефективніших механізмів забезпечення інформаційної безпеки. Вона пропонує революційний підхід до захисту цифрових даних від несанкціонованого доступу, кіберзлочинності та різноманітних шахрайських схем, суттєво знижуючи ризики викрадення, фальсифікації чи компрометації критично важливої інформації.

Унікальність архітектури блокчейн полягає в її децентралізованій природі. На відміну від традиційних централізованих систем зберігання даних, інформація в блокчейн-мережі розподіляється між численними учасниками. Щоб скомпрометувати або зруйнувати такий ланцюжок, зловмиснику необхідно було б одночасно атакувати та знищити дані на комп’ютерах абсолютно всіх користувачів глобальної мережі. Враховуючи, що мова може йти про мільйони пристроїв, розташованих у різних куточках планети, кожен з яких зберігає повну або часткову копію реєстру транзакцій, виконання такого завдання стає практично нездійсненним.

Навіть якщо хакеру вдасться вивести з ладу частину мережі, решта комп’ютерів-вузлів автоматично продовжить функціонувати, підтримуючи цілісність системи. Ці непошкоджені вузли постійно здійснюють верифікацію та синхронізацію всіх записів, забезпечуючи безперервність роботи блокчейну.

Рівень захищеності мережі прямо пропорційний кількості її активних учасників. Чим більша кількість користувачів підтримує функціонування блокчейн-системи, тим експоненційно складнішою стає будь-яка спроба злому. Масштабні блокчейн-платформи з мільйонами учасників демонструють практично абсолютну стійкість до кібератак завдяки надзвичайній технічній складності проникнення в таку розгалужену інфраструктуру.

1.3 Постановка завдання на кваліфікаційну роботу магістра

Сучасний розвиток інформаційних технологій супроводжується постійним зростанням загроз кібербезпеки. Щодня зловмисники використовують все більш складні методи проникнення в інформаційні системи, що ускладнює їх виявлення та протидію. Традиційні засоби захисту, такі як антивірусні програми, міжмережеві екрани та системи виявлення вторгнень, часто не здатні ефективно виявляти нові або приховані атаки, особливо ті, що застосовують нестандартні вектори доступу. У зв’язку з цим особливу актуальність набувають технології

honeypot – спеціально налаштовані системи, що імітують вразливі об’єкти для залучення зловмисників та збору інформації про їхню активність.

Основна цінність honeypot-систем полягає у здатності виявляти атаки на ранньому етапі та отримувати детальну інформацію про методи та інструменти, які застосовують зловмисники. Проте важливим аспектом залишається збереження і забезпечення достовірності зібраних даних, оскільки журнали подій можуть бути модифіковані або видалені при спробах вторгнення. Для вирішення цієї проблеми доцільно використовувати технологію блокчейн, яка забезпечує децентралізоване та незмінне зберігання даних, гарантує їх цілісність і прозорість.

Однак у процесі збирання й обробки даних виникає проблема достовірності та цілісності інформації, що може бути підроблена або змінена зловмисниками після виявлення системи-пастки. З метою вирішення цієї проблеми пропонується використати технологію блокчейн, яка забезпечує незмінність, розподілене зберігання та криптографічний захист даних. Завдяки децентралізованому характеру блокчейну, записи про інциденти безпеки не можуть бути змінені або видалені без порушення всієї структури ланцюга блоків, що гарантує повну прозорість і надійність даних.

Таким чином, інтеграція honeypot-систем із блокчейн-технологією дозволяє створити нову модель безпеки, де інформація про кібератаки автоматично фіксується в незмінному вигляді та може бути використана для подальшого аналізу, цифрової форензіки або аудиту подій.

Кваліфікаційна робота передбачає розробку програмної системи, яка реалізує архітектуру інтегрованого honeypot-блокчейн рішення. В рамках досягнення поставленої мети передбачається вирішення низки ключових завдань:

– дослідити існуючі підходи до побудови honeypot-систем та оцінити їх ефективність;

– визначити можливості впровадження блокчейн для реєстрації інцидентів безпеки;

– створити прототип системи, який буде здатний збирати, верифікувати та фіксувати події безпеки у розподіленому середовищі;

– провести експериментальні дослідження з метою оцінювання продуктивності, масштабованості та достовірності запропонованого рішення.

Для оцінки результатів передбачено проведення експериментальних досліджень, у ході яких буде виявлено кількість атак, зафіксованих системою, точність і надійність збереження інформації в блокчейні, а також швидкодію обробки подій. Окрему увагу буде приділено порівнянню запропонованого підходу з традиційними методами фіксації інцидентів, що дозволить визначити переваги інтегрованого рішення.

Отже, дана кваліфікаційна робота спрямована на створення та дослідження інтегрованої системи безпеки, яка поєднує інтелектуальні механізми виявлення загроз (honeypot) із технологіями гарантованої довіри та незмінності даних (blockchain). Реалізація такого підходу сприятиме підвищенню рівня кіберстійкості інформаційних систем, а також розвитку сучасних методів активного моніторингу та аналізу кібератак.

Висновки до розділу 1

У межах цього розділу здійснено огляд і аналіз предметної області проблеми, результати існуючих теоретичних та експериментальних досліджень; огляд і аналіз методів та засобів безпеки з використанням honeypots для вирішення проблеми дослідження.

Окрему увагу приділено використанню технології блокчейну для реалізації deception technologies при реалізації honeypot (кіберпасток). Кваліфікаційна робота присвячена комплексному вивченню гібридної системи кібербезпеки, що органічно інтегрує передові технології honeypot для активного виявлення та аналізу потенційних загроз із розподіленою blockchain-архітектурою, яка забезпечує криптографічну незмінність і верифікацію критичних даних.

РОЗДІЛ 2
ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ БЕЗПЕКИ З ВИКОРИСТАННЯМ HONEYPOTS

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

Проведений детальний аналіз сучасних систем обману (Deception) та технології Honeypot продемонстрував значні перспективи їхнього подальшого розвитку та еволюції. Дослідження виявило численні можливості для розширення функціональності цих технологій у контексті сучасної кібербезпеки. Проте, незважаючи на їхню ефективність, обидві системи – як Deception, так і Honeypot – базуються на централізованій архітектурі, що створює суттєві вразливості та обмеження.

Ключовою проблемою централізованих систем захисту залишається наявність єдиного сервера управління, який виступає критичною точкою відмови. У випадку, коли досвідчений зловмисник ідентифікує цю основну ланку інфраструктури, він отримує можливість адаптувати свою тактику атаки, обходити встановлені захисні механізми або навіть повністю компрометувати всю систему моніторингу. Це створює серйозні ризики для організацій, які покладаються виключно на такі рішення.

Для ефективної нейтралізації зазначених ризиків необхідно побудувати принципово нову систему захисту, яка не залежатиме від функціонування одного централізованого вузла. Технологія Blockchain пропонує революційний підхід до вирішення цієї проблеми. Blockchain є розподіленою багатовузловою системою, в якій кожен незалежний вузол мережі повинен здійснити перевірку та підтвердження інформації, що надходить до будь-якої з ланок, перш ніж ця інформація буде інтегрована в загальний потік даних та стане частиною незмінного ланцюга записів.

Унікальні властивості Blockchain, зокрема механізми динамічної зміни конфігурації мережі та автоматичного валідування вузлів, можуть багаторазово

посилити існуючі системи захисту. Це дозволить усунути критичну проблему централізованого керування та створити більш стійку інфраструктуру безпеки. На основі цього підходу необхідно розробити та змоделювати динамічну розподілену систему управління, яка використовуватиме властивості Blockchain, а також провести комплексне дослідження параметрів та ефективності даної гібридної системи в реальних умовах експлуатації.

Пропонована система являє собою інноваційну динамічну розподілену модель програмної приманки (honeypot), побудовану на основі N незалежних хостів та трьох ключових сервісів. Архітектура передбачає взаємодію двох типів учасників: потенційного зловмисника (хакера) та легітимного користувача. Важливою особливістю є те, що авторизований користувач постійно синхронізований з реальним продуктивним сервісом, завдяки чому клієнт має можливість відстежувати актуальне місцезнаходження справжніх ресурсів і володіє точною інформацією про їхнє розташування в мережі.

Фундаментом системи виступає приватна блокчейн-мережа, сформована з N хостів, що функціонує за принципом peer-to-peer (P2P) топології. Критичною характеристикою цієї мережі є її закритість – вона не надає доступу до зовнішнього світу, що забезпечує додатковий рівень безпеки та ізоляції від несанкціонованих втручань.

У якості базової платформи нижнього рівня використовується Multichain – одна з найпотужніших блокчейн-платформ сучасності. N хостів колективно формують приватний блокчейн-ланцюг, створюючи децентралізовану P2P-мережу. Кожен хост має можливість брати участь у процесі майнінгу: обчислюючи криптографічне хеш-значення блоку, вузол може згенерувати новий потенційний блок та інтегрувати його до існуючого ланцюга. Цей механізм консенсусу гарантує справжню розподіленість та децентралізацію всієї архітектури розгортання системи.

Координацію роботи забезпечує тимчасовий головний хост, який виконує спеціалізований алгоритм розподілу сервісів між вузлами мережі. Після прийняття рішень щодо розподілу, головний вузол передає відповідну

зашифровану інформацію іншим хостам через захищені канали зв’язку, забезпечуючи конфіденційність та цілісність даних у системі.

Вузол мережі, який успішно видобуває черговий блок, отримує тимчасову роль координатора децентралізованої системи. Цей координуючий вузол несе відповідальність за генерацію конфігураційних даних про розподіл сервісів між усіма учасниками мережі. На основі алгоритму псевдовипадкової генерації система визначає, які саме служби має запускати кожен окремий хост – справжні робочі сервіси або спеціалізовані honeypot-пастки, призначені для виявлення та аналізу зловмисної активності.

Конфігураційна інформація структурується у вигляді ідентифікаторів сервісів та бінарних кодів (послідовностей нулів і одиниць), які визначають тип служби для запуску. Для забезпечення конфіденційності та захисту від несанкціонованого втручання ці дані піддаються шифруванню за допомогою асиметричного алгоритму RSA з довжиною ключа 2048 біт, що гарантує високий рівень криптографічної стійкості.

Зашифрований пакет конфігураційних даних розповсюджується координуючим вузлом всім іншим хостам у приватному блокчейн-сегменті мережі. При отриманні зашифрованого повідомлення кожен вузол виконує процедуру дешифрування з використанням свого приватного ключа, отримуючи відкриті конфігураційні дані. Інтерпретація бінарного коду відбувається наступним чином: значення «0», сигналізує про необхідність активації honeypot-сервісу (програмної пастки), тоді як «1» вказує на запуск легітимної робочої служби.

Після побітового аналізу отриманих інструкцій система автоматично ініціалізує відповідні сервіси, завершуючи процес конфігурації. Для авторизованих користувачів реалізовано механізм синхронізації, який забезпечує безперервний доступ до потрібних ресурсів. Сервер передає легітимним користувачам зашифровані дані про розташування справжніх сервісів, що дозволяє їм отримувати стандартні послуги без переривань. Також передбачено можливість активного запиту: користувач може надіслати

зашифрований запит для отримання актуальної адреси необхідної служби. Завдяки такому підходу авторизовані користувачі зберігають повноцінний доступ до реальних системних ресурсів, незалежно від поточної конфігурації honeypot-інфраструктури.

В кваліфікаційній роботі запропоновано метод побудови динамічної системи захисту, заснованої на використанні програмних приманок (honeypots). Фундаментальною основою представленого методу є спеціально розроблена модель, практична реалізація якої здійснюється через послідовне виконання наступних структурованих кроків та етапів взаємодії компонентів системи:

– процес класифікації та перенаправлення загроз. Коли вхідний трафік надходить до мережевої інфраструктури, він спочатку проходить ретельну перевірку через брандмауер. Програмне забезпечення, інтегроване в маршрутизатор, виконує глибокий аналіз кожного запиту, використовуючи складні алгоритми для виявлення потенційно шкідливої активності. У момент ідентифікації підозрілого трафіку система автоматично ініціює механізм розгортання спеціалізованих програмних приманок (honeypots). Це відбувається в режимі реального часу без втручання адміністратора. Після виявлення загрози всі шкідливі запити автоматично перенаправляються на щойно створені honeypot-сервіси. Цей процес відбувається прозоро для зловмисника, який не підозрює про те, що взаємодіє не з реальною системою, а з контрольованою пасткою, призначеною для збору інформації про методи атаки;

– блокчейн-інфраструктура для координації. Основою координації всієї системи виступає приватний блокчейн на базі платформи Multichain. Ця децентралізована база даних використовується для безпечного обміну критичною інформацією про розгортання всіх сервісів і приманок у розподіленій мережі. Блокчейн забезпечує незмінність записів, прозорість операцій та синхронізацію даних між усіма учасниками мережі;

– динамічне розгортання сервісів. Система реалізує інноваційний підхід до розподілу сервісів: кожен вузол мережі отримує випадковий набір служб, що робить інфраструктуру непередбачуваною для атакуючих. Програма розподілу,

яка функціонує на тимчасовому головному сервері, керує цим процесом, використовуючи адреси блокчейн-акаунтів серверів як вхідні параметри. Кожному сервісу призначається унікальний бінарний код, що визначає його стан. Наприклад, для чотирьох сервісів (Nginx, Node.js та двох honeypot-служб) код може виглядати як «0101», де «0» позначає активний стан, а «1» деактивований. Така система кодування дозволяє ефективно управляти конфігурацією всієї мережі;

– смарт-контракти та ротація ролей. Деталі розгортання надійно зберігаються в блокчейні через спеціальні смарт-контракти, які забезпечують контрольований доступ до інформації. Кожен сервер функціонує як повноцінний вузол приватного блокчейну, що гарантує синхронізацію даних у режимі реального часу. Критично важливою особливістю системи є механізм періодичної ротації: через визначені інтервали часу один з клієнтських серверів автоматично отримує роль головного сервера, а попередній лідер переходить у статус клієнта. Новий головний сервер негайно ініціює повне перерозподілення всіх сервісів випадковим чином, що додатково підвищує непередбачуваність та безпеку інфраструктури.

Фундаментальним принципом комунікації в технології Blockchain є механізм розподіленого консенсусу. Цей підхід передбачає, що спеціалізовані вузли-валідатори здійснюють ретельну верифікацію інформації, яка надходить від інших учасників мережі. Процес прийняття рішення відбувається через децентралізоване голосування, де валідатори колективно визначають, чи заслуговує отримана інформація на включення до розподіленого реєстру.

Оскільки цілісність даних, що циркулюють між вузлами мережі, є критичною для стабільного функціонування всієї системи, їхній аналіз має першочергове значення для забезпечення кібербезпеки. Потенційно шкідливі дані можуть містити команди, спрямовані на виконання деструктивних операцій. Хост-генератор такої шкідливої інформації автоматично класифікується як скомпрометований. Для глибокого аналізу загроз безпеці система використовує

абстраговані моделі передачі даних, зосереджуючись на виявленні та запобіганні потенційним атакам.

Ефективність впровадження honeypot-технологій у контексті забезпечення інформаційної безпеки визначається їхньою здатністю відволікати та перенаправляти кіберзагрози від критичної корпоративної інфраструктури. Фундаментальною передумовою успішності цих систем виступає створення високоавтентичного цифрового середовища, яке максимально точно імітує реальні виробничі системи та ресурси підприємства.

Архітектурна досконалість honeypot-рішень фундаментально базується на глибокому розумінні принципів конструювання віртуальних приманок, які переконливо відтворюють цінні корпоративні активи з високим рівнем технологічної достовірності та реалістичності поведінки. Коректно спроектовані та налаштовані honeypot-системи трансформуються в інтелектуальні безпекові механізми, що виконують подвійну стратегічну функцію: по-перше, ефективно відволікають увагу потенційних зловмисників від справжніх критичних інформаційних активів, створюючи ілюзію вразливих точок входу, по-друге, забезпечують оперативне детектування, аналіз та детальне документування несанкціонованої активності в реальному часі з можливістю подальшого форензичного дослідження.

Застосування такої стратегічної методології дозволяє організаціям не тільки ідентифікувати існуючі вразливості безпеки на ранніх етапах атаки, але й формувати комплексні практичні рекомендації щодо побудови захищеної контейнерної інфраструктури з урахуванням актуальних векторів загроз. Це набуває особливої критичної актуальності в умовах широкого розповсюдження хмарних обчислень, контейнерних технологій та мікросервісних архітектурних підходів, де традиційні периметральні методи захисту виявляються недостатньо ефективними для протидії сучасним складним багатовекторним кіберзагрозам та цілеспрямованим атакам.

2.2 Практична реалізація об’єкта проектування

Сучасні організації та компанії повсюдно впроваджують хмарні обчислення у свою операційну діяльність. Оскільки працівники всіх підрозділів безпосередньо взаємодіють з кіберпростором, критично важливі бізнес-процеси стають уразливими перед загрозами з боку зловмисників та кіберзлочинців. Для забезпечення безперебійного виконання операційних завдань необхідно приділяти особливу увагу конфіденційності даних та мережевій безпеці IT-інфраструктури.

Ефективний контроль роботи IT-відділу та впровадження комплексних захисних заходів мінімізують вплив потенційних ризиків на критичну інфраструктуру компанії. Одним із найдієвіших інструментів захисту є систематичний аналіз журналів логування IT-інфраструктури. Ці журнали фіксують кожну дію в системі, дозволяючи фахівцям з кібербезпеки своєчасно виявляти підозрілу активність та запобігати атакам.

Для ефективного моніторингу інфраструктури в кваліфікаційний роботі використовується пакет Elastic Stack, який інтегрує комплексний набір спеціалізованих інструментів: Elasticsearch для індексації, Logstash для обробки, Kibana для візуалізації та Filebeat для збору даних. На рисунку 2.1 продемонстровано функціональна схема ELK Stack.

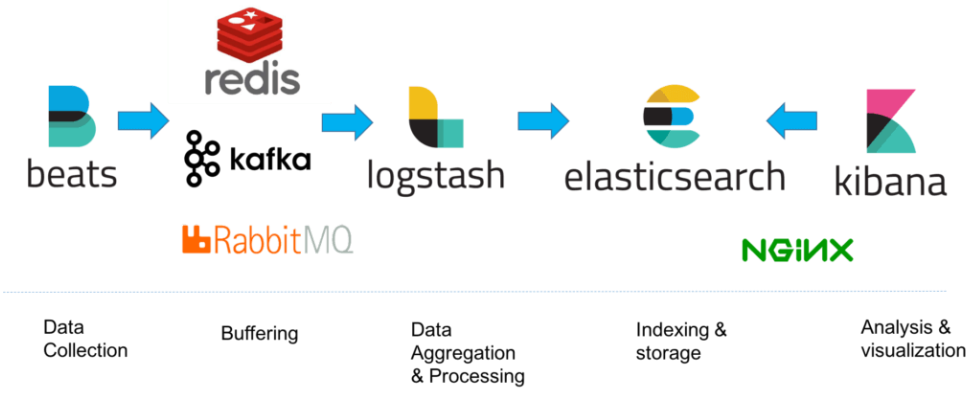

Рисунок 2.1 – Функціональна схема ELK Stack [16]

Реалізація моніторингу Elastic Stack потребує налаштування всіх компонентів. Elasticsearch забезпечує індексування та масштабований пошук документів у реальному часі з підтримкою мультиарендності. Система використовує розподілену архітектуру: індекси діляться на сегменти з реплікацією, кожен вузол координує операції, автоматично балансуючи навантаження. Метою розгортання є моніторинг інфраструктури та контроль інформаційних потоків через open source сервіси: elasticsearch, logstash, kibana, filebeat. Автоматизація конфігурації системи з хмарною приманкою виконана за спеціальним сценарієм, що відображено на лістингу 2.1.

Лістинг 2.1 – Розгортання ELK Stack

```
      sudo apt update && sudo apt upgrade -y
      sudo apt install apt-transport-https curl gnupg -y
      curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch
      echo "deb [signed-by=/usr/share/keyrings/elastic.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo
tee /etc/apt/sources.list.d/elastic-8.x.list
      sudo apt install elasticsearch -y
      sudo systemctl start elasticsearch
      sudo apt install logstash -y
      sudo systemctl start logstash
      sudo apt install kibana -y
      sudo systemctl start kibana
```

кінець лістингу 2.1

Після виконання всіх кроків ми отримаємо те, що elasticsearch слухає на 9200; logstash приймає події та пересилає їх до elasticsearch; kibana доступна на 5601 і відображає лог-дані в реальному часі. Перевірка логів здійснюється з використанням службових команд, що наведено на лістингу 2.2.

Лістинг 2.2 – Сценарій перевірки логів

```
      sudo journalctl -u elasticsearch -f
      sudo journalctl -u kibana -f
      sudo journalctl -u logstash -f
```

кінець лістингу 2.2

Рисунок 2.2 демонструє перевірку активності elasticsearch та logstash для підтвердження їх коректної роботи та готовності до використання.

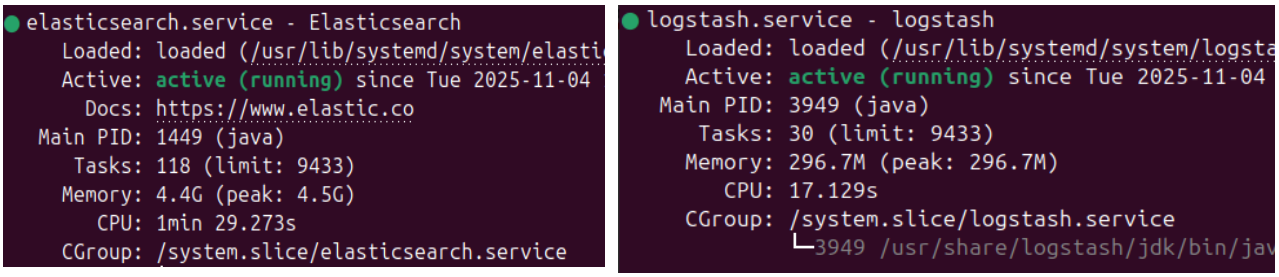

Рисунок 2.2 – Верифікація elasticsearch та logstash

Візуалізація даних Elasticsearch реалізується через Kibana – плагін з відкритим кодом, що дозволяє створювати гістограми, лінійні, точкові діаграми та карти на великих обсягах даних. Для перевірки функціональності Kibana потрібно відкрити веб-браузер і перейти за адресою http://10.0.2.15:5601. Після завантаження відкриється повнофункціональне середовище Elastic Stack, через яке здійснюється централізований моніторинг системних подій, конфігурацій, активності користувачів та роботи агента. Kibana забезпечує зручні інструменти для візуалізації даних, створення інтерактивних дашбордів та аналізу логів у реальному часі. На рисунку 2.3 показано інтерфейс Elastic Stack та вікно Kibana.

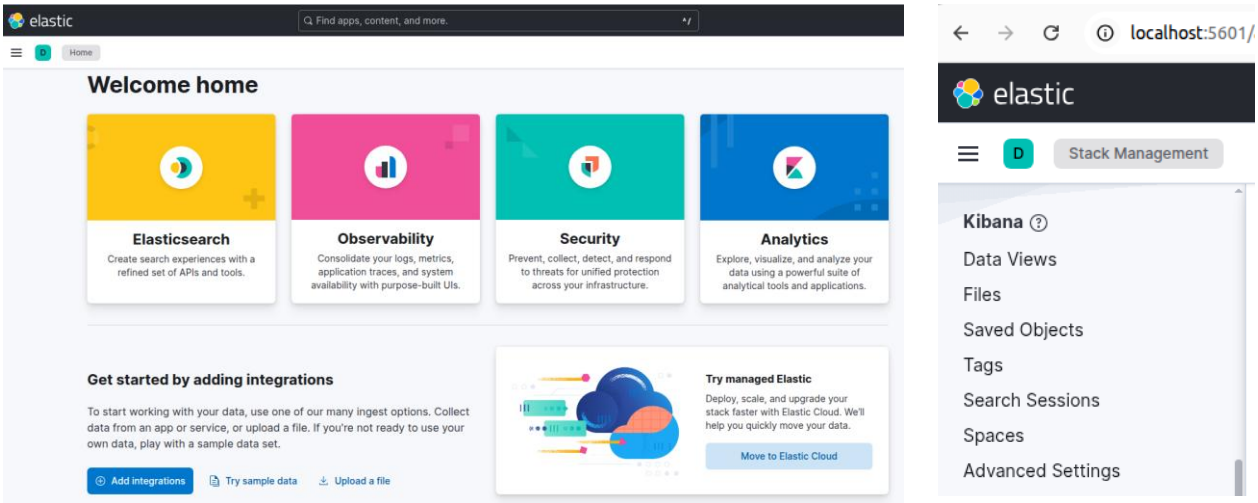

Рисунок 2.3 – Середовище Elastic та панель kibana

У дослідженні для ефективного виявлення та превентивного запобігання кібератакам на хмарну інфраструктуру застосовується комплексний підхід, що поєднує потужну систему централізованого моніторингу ELK Stack із спеціалізованою honeypot-пасткою cowrie. Хмарна приманка являє собою високоефективний інструмент для глибокого аналізу шаблонів використання різноманітних сервісів, безпечного відстеження зловмисних дій у повністю ізольованому віртуальному середовищі та систематичного збору критичних даних про компрометовані облікові записи, включаючи логіни, паролі, виконувані shell-команди та інші важливі артефакти атак. Окремі honeypot-системи додатково обладнані функціоналом для перехоплення та збереження виконуваних файлів і бінарних об’єктів, що завантажуються на сервер зловмисниками, а їх подальше детальне дослідження дозволяє ідентифікувати раніше невідомі зразки шкідливого програмного забезпечення. Налаштування та конфігурацію cowrie здійснюємо у відповідності до лістингу 2.3.

Лістинг 2.3 – Конфігурація cowrie

```
sudo apt update
sudo apt install -y python3-venv git
git clone https://github.com/cowrie/cowrie.git /opt/cowrie
cd /opt/cowrie
python3 -m venv cowrie-env
source cowrie-env/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
cp etc/cowrie.cfg.dist etc/cowrie.cfg
```

кінець лістингу 2.3

Налаштування honeypot cowrie передбачає взаємодію з платформою ELK Stack. Дашборд у Kibana дає змогу аналізувати події, зафіксовані cowrie, тобто спроби SSH/Telnet підключень, введені команди, логіни, IP-адреси атакувальників та інші показники. Це допомагає відстежувати динаміку атак; ідентифікувати географічні джерела загроз; бачити найбільш часті команди, які використовують зловмисники. Процес створення дашборду Kibana для

моніторингу honeypot-cowrie передбачає кілька послідовних етапів налаштування візуалізації даних про кібератаки.

Спочатку необхідно налаштувати index pattern у kibana, який вказуватиме на індекс elasticsearch, що містить логи Cowrie. Зазвичай це індекси з префіксом «cowrie-*». Після створення індексного шаблону важливо переглянути доступні поля та їх типи даних для коректної побудови візуалізацій. На рисунку 2.4 відображено створення патерну для приманки.

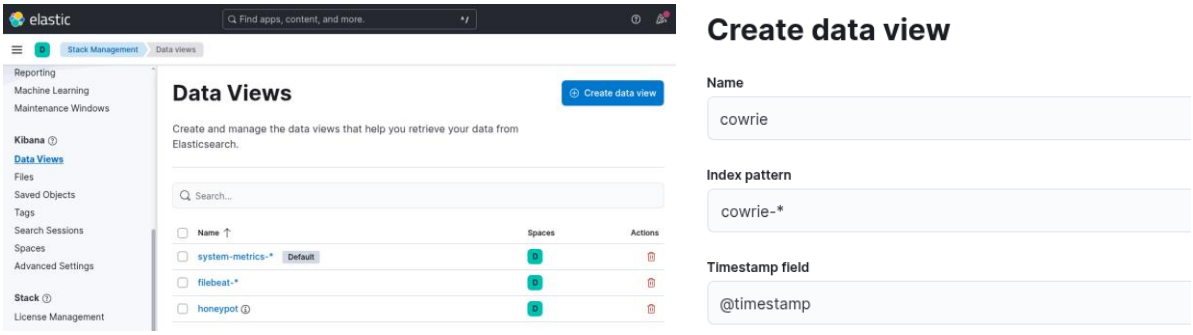

Рисунок 2.4 – Панель data view

Створення дашборду було здійснено з використанням панелі Kibana → Data Views → Create data views. Для ефективної візуалізації додано наступні віджети: visualization type: area chart / line chart; X-axis: @timestamp (date histogram); Y-axis: count; показує кількість атак у часі. На рисунку 2.5 проілюстровано журнал активності подій, використання транспортних та мережевих протоколів.

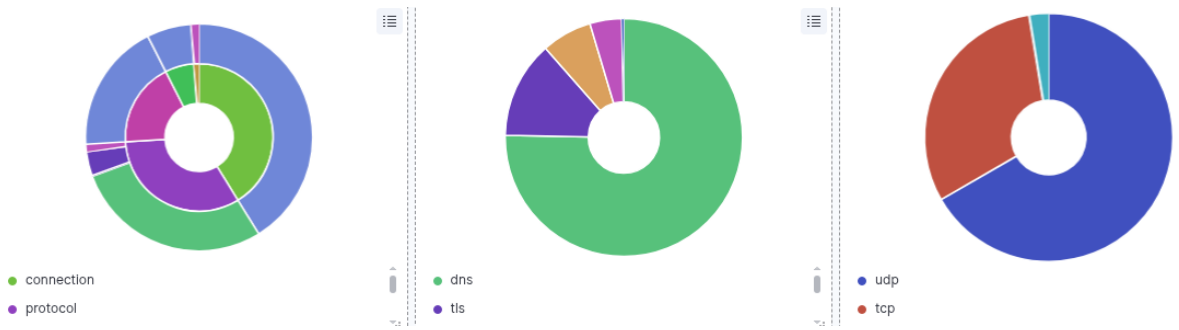

Рисунок 2.5 – Панель візуалізацій filebeat

Наступний етап практичної реалізації системи був зосереджений на детальному налаштуванні та конфігурації blockchain-інфраструктури. Для побудови децентралізованої платформи було обрано Multichain – сучасне блокчейн-рішення, що повністю підтримує індустрійні стандарти ERC-20 для взаємозамінних токенів та ERC-721 для унікальних цифрових сертифікатів.

Під час вибору блокчейн-платформи для реалізації проєкту було ретельно проаналізовано декілька альтернативних варіантів, включаючи Binance Smart Chain та Solana.

Binance Smart Chain привертає увагу завдяки суттєво нижчим комісіям за проведення транзакцій порівняно з іншими мережами, що робить її фінансово привабливою для користувачів. Проте детальний аналіз виявив значні обмеження: недостатня підтримка сучасних стандартів токенів, присутність централізованих елементів в архітектурі мережі та потенційні ризики для безпеки й децентралізації системи знижують її придатність для довгострокового використання в серйозних проєктах.

Solana демонструє вражаючу продуктивність із можливістю обробки тисяч транзакцій за секунду. Однак її екосистема залишається відносно молодою та менш стабільною, інфраструктура продовжує розвиватися, а документація й інструменти для розробників значно поступаються за якістю та повнотою.

Натомість Multichain виокремлюється завдяки своєму широкому глобальному розповсюдженню, потужній підтримці активної спільноти розробників, бездоганній інтеграції з численними бібліотеками та фреймворками, а також перевіреній часом надійності, що зробило цю платформу оптимальним і виваженим вибором для реалізації поставлених завдань.

Стандарт ERC-20 забезпечує високий рівень уніфікації при роботі з токенами, що критично важливо для безшовної інтеграції із внутрішніми модулями системи винагород та зовнішніми застосунками. Цей стандарт дозволяє створювати цифрові активи із чітко визначеними правилами емісії, передачі та обігу, гарантуючи прозорість усіх транзакцій.

Паралельно, стандарт ERC-721 використовується для генерації унікальних, невзаємозамінних токенів (NFT), які представляють цифрові сертифікати досягнень. Кожен такий сертифікат має унікальний ідентифікатор та метадані, що робить неможливим його підробку. Завдяки блокчейн-технології, автентичність і дійсність кожного сертифіката можна миттєво перевірити через публічний реєстр, забезпечуючи повну прозорість та довіру до системи винагород. На рисунку 2.6 проілюстровано архітектурна схема використання блокчейну на основі Multichain для honeypot cowrie.

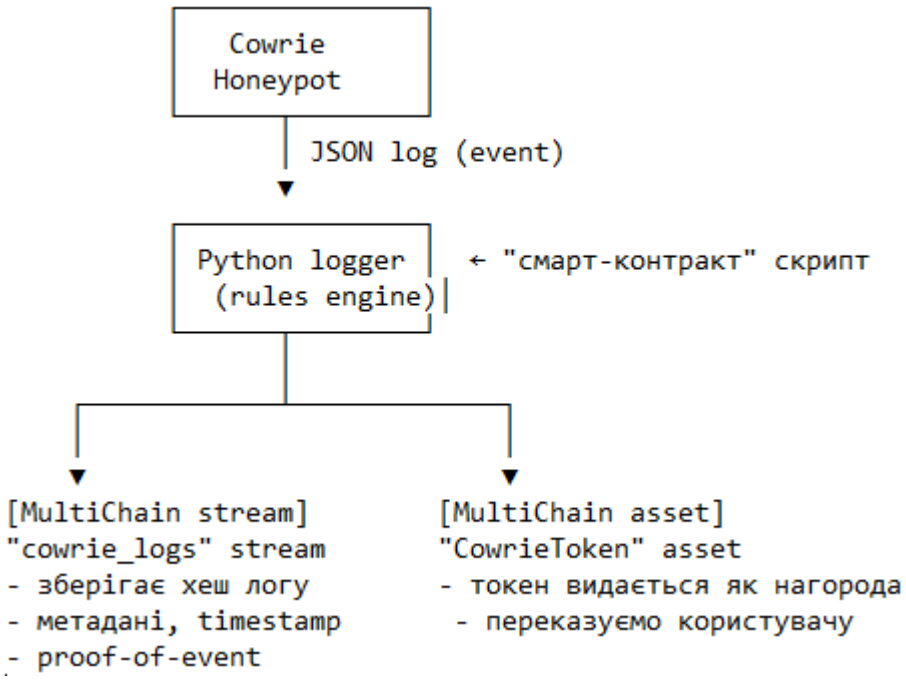

Рисунок 2.6 – Архітектурна схема

Для ефективної взаємодії honeypot cowrie з блокчейн-мережею Multichain було реалізовано комплексну систему налаштувань, що забезпечує надійне збереження та верифікацію зібраних даних про кібератаки.

Базова конфігурація підключення передбачає встановлення з’єднання з вузлом Multichain через JSON-RPC інтерфейс, використовуючи захищений протокол передачі даних. У конфігураційному файлі cowrie визначаються параметри підключення: адреса вузла blockchain, порт для комунікації, облікові дані для автентифікації та назва цільового потоку даних для запису інформації

про атаки. Механізм буферизації забезпечує тимчасове збереження даних у локальній черзі на випадок тимчасової недоступності blockchain-вузла, гарантуючи безперервність моніторингу без втрати критичної інформації про інциденти безпеки. На лістингу 2.4 проілюстровано основні команди для конфігурації Multichain.

Лістинг 2.4 – Налаштування Multichain

```
      sudo apt update && sudo apt upgrade -y
      sudo apt install -y wget unzip
      wget https://www.multichain.com/download/multichain-
2.3.3.tar.gz
      tar -xvzf multichain-2.3.3.tar.gz
      cd multichain-2.3.3
      sudo mv multichaind multichain-cli multichain-util
/usr/local/bin/
      multichaind --version
      MultiChain 2.3.3 Daemon (protocol 10008-10012, 20004-20013)
```

кінець лістингу 2.4

Реалізація механізм консенсусу передбачає виконання наступних кроків:

– створення блокчейну за допомогою службових команд;

– створення токена (asset) за допомогою службових команд;

– створення stream для логів за допомогою службових команд;

– перевірка та аудит за допомогою службових команд.

Службові команди механізму консенсусу відображено на лістингу 2.5

Лістинг 2.5 – Реалізація механізм консенсусу

```
      multichain-util create cowriechain
      multichaind cowriechain -daemon
      multichain-cli cowriechain issue COWRIE_TOKEN 1000
      multichain-cli cowriechain getaddressbalances
      multichain-cli cowriechain create stream cowrie_logs true
      multichain-cli cowriechain subscribe cowrie_logs
      multichain-cli cowriechain liststreamitems cowrie_logs
      multichain-cli cowriechain getaddressbalances
```

кінець лістингу 2.5

Для реалізації системи моніторингу та збереження даних про кібератаки через honeypot cowrie було розроблено спеціалізований смарт-контракт на базі Multichain з використанням нативних токенів та потоків даних (додаток В). Налаштовано мультипідписну схему з трьома адміністраторами, де для критичних операцій потрібні підписи мінімум двох учасників. Адреса honeypot-сервера отримує ексклюзивні права на запис (write) до всіх потоків, тоді як аналітичні вузли мають права лише на читання (read). Смарт-контракт автоматично перевіряє цифрові підписи вхідних даних, валідує формат JSON-структур та зберігає Merkle-корені для пакетів даних, забезпечуючи криптографічне підтвердження незмінності записів про інциденти безпеки.

Висновки до розділу 2

У межах даного розділу було проведено комплексне теоретичне дослідження та здійснено практичну реалізацію системи безпеки на основі honeypot-технологій. Теоретична частина роботи включала детальний аналіз сучасних методів виявлення кіберзагроз та обґрунтування вибору оптимальних технологічних рішень. Особливу увагу приділено вивченню алгоритмів збору та аналізу даних про атаки, а також методів їх незмінного зберігання.

Практична реалізація проекту передбачала декілька ключових етапів: розгортання та детальну конфігурацію платформи ELK Stack для централізованого збору, індексації та візуалізації логів; налаштування honeypot cowrie для емуляції SSH/Telnet сервісів та збору інформації про спроби несанкціонованого доступу; розробку інноваційної архітектурної схеми інтеграції блокчейн-технології на базі платформи Multichain для забезпечення незмінності та прозорості зібраних даних про атаки. Така архітектура дозволяє створити надійну систему моніторингу загроз з гарантованою цілісністю forensic-даних.

РОЗДІЛ 3
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ БЕЗПЕКИ З ВИКОРИСТАННЯМ HONEYPOTS

3.1 Методика проведення дослідження

В цьому розділі детально розглядається метод побудови динамічної системи кібербезпеки, що базується на використанні програмних приманок (honeypots) у поєднанні з технологією розподіленого реєстру. Запропонований підхід являє собою комплексне рішення для виявлення, ізоляції та аналізу шкідливого трафіку в мережевому середовищі. Методологія включає математичний опис та модель функціонування системи, що реалізується через послідовність взаємопов’язаних етапів.

Механізм класифікації та перенаправлення шкідливого трафіку. Першочерговим етапом роботи системи є інтелектуальна обробка вхідних запитів на рівні мережевої інфраструктури. Коли мережевий трафік проходить через брандмауер, спеціалізоване програмне забезпечення, інтегроване безпосередньо в маршрутизатор, здійснює аналіз кожного запиту в режимі реального часу. Система класифікації використовує набір евристичних правил та сигнатур для ідентифікації потенційно шкідливих запитів. При виявленні підозрілої активності автоматично ініціюється процес розгортання відповідних програмних приманок, які імітують легітимні сервіси. Після ідентифікації загрози система здійснює автоматичне перенаправлення всіх шкідливих запитів на щойно розгорнуту приманку. Цей механізм дозволяє ізолювати потенційно небезпечну активність від критичних ресурсів мережі, одночасно забезпечуючи можливість детального аналізу поведінки зловмисників та застосовуваних ними тактик, технік і процедур.

Архітектура на базі приватного блокчейн Multichain [17]. Для забезпечення надійного та безпечного обміну інформацією між компонентами системи реалізовано приватну мережу на базі блокчейн Multichain. Ця високопродуктивна платформа розподіленого реєстру використовується для

централізованого зберігання та синхронізації деталей розгортання всіх сервісів та приманок у мережевій інфраструктурі. Використання технології блокчейн гарантує цілісність даних, незмінність історії змін та автоматичну реплікацію інформації між усіма вузлами системи.

Динамічне розгортання сервісів та механізм випадкового розподілу. Ключовою особливістю запропонованого підходу є система динамічного розгортання служб, яка функціонує на кожному вузлі мережі абсолютно випадковим чином. Програма розподілу служб, що виконується на тимчасовому головному сервері, керує цією функціональністю шляхом взаємодії з блокчейн. Система отримує адресу облікового запису кожного сервера в ланцюзі блоків та призначає унікальний код служби, представлений у вигляді бінарної послідовності, де довжина коду дорівнює загальній кількості доступних служб. Для ілюстрації розглянемо практичний приклад: якщо в системі функціонують чотири основні служби (Nginx, Node.js, Tomcat) та відповідні їм програмні приманки, то код служби матиме формат чотирьох бінарних розрядів (наприклад, 0000), де значення «0» вказує на активний стан служби, а «1» – на деактивований.

Смарт-контракти та розподілене зберігання даних. Програма розподілу автоматично генерує деталі конфігурації розгортання, які повинні бути безпечно розподілені між усіма серверами-учасниками системи. Blockchain виступає як розподілене сховище для реалізації цієї критично важливої функціональності. Кожен сервер представляє окремий вузол приватної мережі блокчейн, що забезпечує автоматичну синхронізацію даних. Таким чином, інформація, збережена з одного вузла, стає миттєво доступною для всіх інших учасників. Для ефективного збереження та отримання даних з блокчейн описано та імплементовано спеціалізовані смарт-контракти, розгорнуті в приватній мережі. Ці програмні модулі забезпечують надійне збереження конфігураційних даних та оперативне їх отримання за необхідності, гарантуючи автоматизацію процесів без потреби в централізованому управлінні.

Механізм ротації ролей та перерозподілу. Для підвищення рівня безпеки та забезпечення відмовостійкості системи реалізовано механізм періодичної ротації ролей. Через визначені часові інтервали один із тимчасових клієнтських серверів автоматично переходить у статус тимчасового головного сервера, тоді як попередній ведучий вузол стає клієнтом. Новопризначений тимчасовий сервер ініціює повний перерозподіл усіх служб у випадковому порядку, що суттєво ускладнює можливість прогнозування конфігурації системи потенційними зловмисниками та підвищує загальний рівень захищеності інфраструктури.

Служби, що функціонують на хості, організовуються та керуються через абстракцію подій введення та виводу даних. Вихідні дані сервісів представлені двома типами подій: generate (data), яка відповідає за генерацію інформації, та send (data, c_i), що забезпечує передачу даних з додатковою контекстною інформацією. Натомість, подія receive (data, c_i) моделює процес отримання та прийому вхідних даних від інших компонентів системи.

З позиції забезпечення інформаційної безпеки, кожен хост може одночасно функціонувати у двох принципово різних операційних режимах. Нормальний режим роботи характеризується відсутністю шкідливого програмного забезпечення або зловмисних даних, що дозволяє хосту підтримувати стабільну та передбачувану роботу відповідно до встановлених специфікацій. На противагу цьому, скомпрометований режим сигналізує про те, що хост функціонує під впливом зловмисного коду, виконує несанкціоновані операції та може завдавати шкоди як власній системі, так і довіреним користувачам.

Стани функціонуючого хості класифікуються за трьома основними категоріями згідно з дослідженням [18]: service_n означає нормальний режим роботи, service_c позначає скомпрометований режим, а найкритичніший стан service_b відображає повну поломку системи, коли хост припиняє будь-яку операційну діяльність. Хост зберігає свій поточний операційний режим протягом певного часу, доки не відбудеться подія переходу, яка моделює динамічні відношення та можливі трансформації між трьома різними режимами

функціонування системи, формуючи складну модель поведінки хоста в різних умовах безпеки. На рисунку 3.1 продемонстровано перехідні стани ланок.

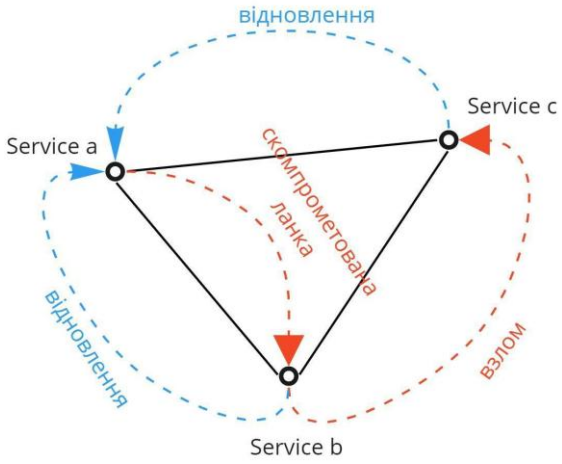

Рисунок 3.1 – Перехідні стани ланок [18]

У запропонованому підході основою взаємодії між елементами системи виступає механізм досягнення консенсусу, притаманний технології блокчейн. Комунікація відбувається між вузлами-валідаорами, які отримують та перевіряють інформацію, надіслану іншими вузлами мережі. Після аналізу отриманих даних валідаори шляхом голосування приймають колективне рішення щодо допуску цієї інформації до спільного реєстру або її відхилення. Такий підхід гарантує високий рівень довіри між учасниками мережі та мінімізує можливість маніпуляції даними.

Якщо певний вузол надсилає інформацію, що не проходить верифікацію або суперечить даним інших учасників, система автоматично вважає такий вузол потенційно скомпрометованим. Це означає, що мережа здатна виявляти аномалії та помилки на етапі обміну інформацією без додаткових зовнішніх механізмів контролю. Завдяки цьому навіть внутрішні атаки, коли зловмисник уже має доступ до інфраструктури, значно ускладнюються, оскільки непомітно вплинути на дані або змінити інформацію без реакції інших вузлів практично неможливо.

Передані дані мають важливе значення для підтримки коректного функціонування системи, тому відіграють ключову роль у механізмах безпеки.

Передбачається, що кожен фрагмент інформації створюється одним конкретним хостом, який несе відповідальність за його достовірність. Однак існує можливість, що хост може бути зламаним і сформувати шкідливі дані, наприклад, команди, які призводять до виконання небезпечних дій. У такому випадку система автоматично визначає джерело компрометації, дозволяючи вчасно локалізувати загрозу та запобігти її поширенню.

Запропонована модель децентралізованої комунікації між вузлами мережі базується на вдосконаленому математичному апараті, який забезпечує ефективну оптимізацію двох критично важливих процесів: класифікації потенційно шкідливих запитів та інтелектуального розподілу сервісних служб між вузлами розподіленої мережі. Система використовує унікальну адресу облікового запису, яку кожен сервер у блокчейн-архітектурі застосовує для ідентифікації та автентифікації, приймаючи її як вхідні параметри. Кожному сервісу присвоюється спеціалізований код служби, що являє собою послідовність двійкових кодів, які забезпечують однозначну ідентифікацію та маршрутизацію запитів.

Архітектура клієнтської частини передбачає створення двох незалежних комунікаційних потоків з відкриттям відповідних мережевих портів, завдяки чому клієнт функціонує в гібридному режимі, виконуючи одночасно роль сервера. Така двостороння конфігурація надає клієнту можливість не лише отримувати службові дані від серверів, але й ініціювати власні запити до серверної інфраструктури, забезпечуючи повноцінну двосторонню комунікацію.

Для отримання актуальної інформації про доступні сервіси система реалізує два різні комунікаційні шаблони. Перший – активний (позитивний) шаблон – передбачає проактивну поведінку клієнта, який самостійно ініціює запити формату «whois + ім’я сервера» до сервера для визначення IP-адреси необхідної служби. Другий – пасивний шаблон – функціонує за принципом очікування: клієнт перебуває в режимі прослуховування, очікуючи на вхідні дані від сервера, який у свою чергу зобов’язаний проаналізувати список підключених клієнтів і розіслати їм актуальні IP-адреси всіх доступних сервісів у мережі.

Безпека комунікації забезпечується криптографічною системою на основі двох ключових файлів (publicKey і privateKey), що містять 2048-бітні ключі шифрування за алгоритмом RSA. Файл publicKey призначений для шифрування даних на етапі, що передує безпосередньому процесу передачі інформації, тоді як файл privateKey використовується для розшифрування отриманих від сервера зашифрованих даних. Після успішного отримання достовірної інформації про необхідний сервіс, клієнт автоматично ініціює з'єднання з відповідною IP-адресою та здійснює доступ до реальних ресурсів, забезпечуючи безперервну та захищену взаємодію в децентралізованій мережевій екосистемі.

3.2 Обробка та аналіз отриманих результатів

У експериментальній частині дослідження представлено детальні результати проведеної симуляції кібератак на спеціально підготовлене вразливе тестове середовище. Отримані в процесі симуляції результати можуть бути передані до спеціалізованої системи дослідження кіберзлочинів для подальшого комплексного багатофакторного аналізу, виявлення закономірностей злочинної активності та ідентифікації характерних патернів поведінки зловмисників.

Експериментальне середовище для генерування та збору журналів подій безпеки побудовано на основі розподіленої системи інтелектуальних приманок (honeypot), яка використовує інноваційну технологію Blockchain для забезпечення криптографічної цілісності, незмінності та достовірності зібраних даних про атаки. Основними функціональними вузлами розробленої системи є високопродуктивні веб-сервіси на базі Nginx, які виконують роль реверсивних проксі-серверів, забезпечуючи інтелектуальну маршрутизацію мережевого трафіку та первинну багаторівневу фільтрацію підозрілих запитів. Окрім цього, побудована інфраструктура включає повністю ізольовані контейнеризовані екземпляри навмисно вразливого веб-застосунку OWASP JuiceShop, який містить широкий спектр типових та найпоширеніших вразливостей сучасних

веб-додатків, що дозволяє імітувати реалістичні сценарії атак та досліджувати методи їх експлуатації зловмисниками.

Для проведення комплексного експериментального аналізу та об'єктивної оцінки ефективності захисних механізмів було організовано та реалізовано симуляції трьох основних категорій кібератак. Перша категорія включала атаки з використанням автоматизованих інструментів сканування мережевої інфраструктури та експлуатації виявлених вразливостей програмного забезпечення. Друга категорія охоплювала розподілені атаки типу «відмова в обслуговуванні» (DDoS) різної інтенсивності та тривалості. Третя категорія складалася з цілеспрямованих ін'єкційних атак, зокрема SQL-ін'єкцій для компрометації баз даних та міжсайтового скриптингу (XSS) для виконання шкідливого коду (рис. 3.2).

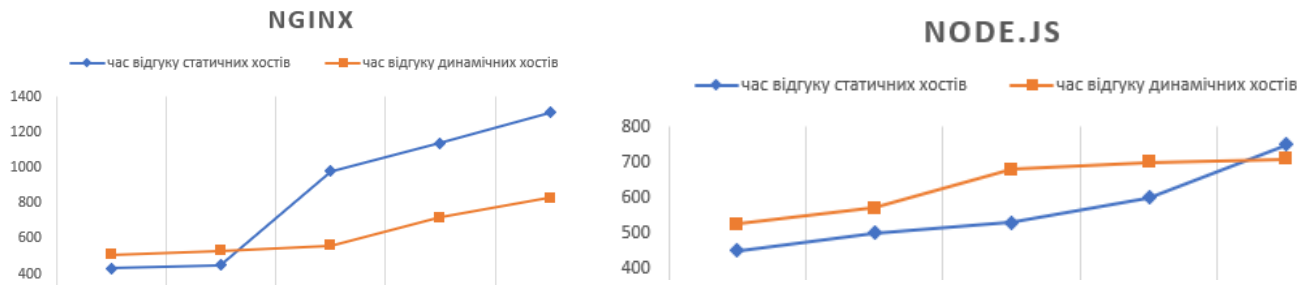

Рисунок 3.2 – Час відгуку сервісів [1]

В рамках експериментального дослідження було організовано однакову кількість однотипних кібератак на дві принципово різні архітектури систем інформаційного захисту: традиційну централізовану систему із статичними програмними приманками та інноваційну динамічну систему захисту, спроектовану та розроблену на основі запропонованого адаптивного методу виявлення та протидії загрозам.

У рамках проведеного науково-дослідного експерименту було реалізовано та ретельно проаналізовано загалом 50 цілеспрямованих спроб кібератак різної складності. Методологія дослідження передбачала рівномірний розподіл атакуючих сценаріїв між двома принципово відмінними архітектурними

рішеннями мережевої інфраструктури: 25 випадків зловмисних втручань були спрямовані проти системи безпеки з інтегрованою авторською моделлю багаторівневого захисту, тоді як решта 25 атак здійснювалися на традиційну централізовану архітектуру з класичними засобами протидії загрозам.

Кожен окремий атакуючий сценарій представляв собою ретельно спланований комплексний вплив, що органічно поєднував три принципово різні, але взаємодоповнюючі методики зловмисного проникнення. По-перше, застосовувалися потужні DDoS-атаки, метою яких було критичне перевантаження системних ресурсів та порушення доступності сервісів. По-друге, впроваджувалася реалістична симуляція ін’єкційних атак для системного виявлення потенційних вразливостей програмного забезпечення та баз даних. По-третє, здійснювалося активне автоматизоване сканування мережевої інфраструктури з метою детальної ідентифікації відкритих портів, запущених сервісів та можливих точок входу. Критично важливим аспектом оцінювання була умова, що атака вважалася повністю відбитою та нейтралізованою виключно за умови успішного блокування всіх трьох незалежних векторів загрози одночасно.

Необхідно окремо наголосити на специфічній ролі honeypot-систем у контексті централізованих моделей безпеки. Як було детально описано у попередніх розділах дослідження, ці програмні приманки не функціонують як безпосередній захисний бар’єр або активний елемент протидії загрозам. Натомість вони виконують стратегічну розвідувальну функцію в загальній архітектурі інформаційної безпеки. Їхнє фундаментальне призначення полягає у цілеспрямованому відволіканні уваги потенційного атакувальника від критичних ресурсів, забезпеченні безперервного детального моніторингу його тактики та методів, а також систематичному документуванні кожного кроку зловмисника у спеціалізованих журналах подій для подальшого форензік-аналізу.

З 25 реалізованих атакуючих сценаріїв проти централізованої системи лише 6 випадків були повноцінно заблоковані за всіма трьома встановленими

критеріями оцінювання, що становить приблизно 0.24 від загального обсягу тестових втручань. Детальний статистичний аналіз отриманих експериментальних результатів демонструє чітку та послідовну закономірність: показники ефективності захисних механізмів демонструють стійку тенденцію до прогресивного зниження пропорційно до зростання загальної інтенсивності та кількості атакуючих дій на систему.

Особливої уваги заслуговує критичний факт, що приблизно 0.24 від загального числа здійснених атак досягли повного тактичного успіху та змогли успішно проникнути через усі наявні рівні багатошарового захисту централізованої інфраструктури. Це відбулося внаслідок того, що досвідчені зловмисники змогли ідентифікувати програмну приманку, правильно розпізнати її справжнє призначення, технічно обійти цей захисний механізм і стратегічно перенаправити свої зусилля безпосередньо на атаку легітимної production-мережевої інфраструктури.

Комплексний аналіз розподілу захисних механізмів за типами кібернетичних атак демонструє чітку диференціацію у рівнях ефективності протидії різним векторам загроз. У контексті DDoS-атак, спрямованих на порушення доступності сервісів через перевантаження ресурсів, система продемонструвала здатність успішно ідентифікувати та нейтралізувати 15 випадків із загальної кількості 25 зафіксованих інцидентів, що відповідає показнику ефективності на рівні 0.60. Це свідчить про достатню, але водночас потребуючу подальшого вдосконалення спроможність протистояти атакам відмови в обслуговуванні.

Що стосується спроб несанкціонованого сканування відкритих портів та виявлення вразливих сервісів, захисні механізми виявились більш результативними, забезпечивши блокування 18 атак із 25 спроб проникнення, досягнувши тим самим 72-відсоткового рівня захисту. Найвищу ефективність система продемонструвала у протидії симульованим ін’єкційним атакам, які є одним із найнебезпечніших векторів компрометації: 22 успішно заблокованих

інциденти з 25 можливих, що еквівалентно вражаючим 0.88 ефективності нейтралізації загроз цього типу.

Для забезпечення об’єктивності оцінювання та створення наочної візуалізації захисного потенціалу запропонованої методології, референтні показники швидкості відповіді чотирьох критично важливих служб в умовах штатного функціонування (за відсутності активних кібератак) були ретельно задокументовані, систематизовані та структуровані у табличному форматі для подальшого компаративного дослідження та порівняльного аналізу продуктивності. В таблиці 3.1 наведено швидкість відгуку сервісів за відсутності атак.

Таблиця 3.1 – Швидкість відгуку сервісів за відсутності атак

Сервіс	Статичний хост, мс	Динамічний хост, мс
Nginx	430	510
Node.js	450	525
Tomcat	505	590

Статичні хости демонструють стабільно кращі показники швидкості відгуку порівняно з їхніми динамічними аналогами, що є очікуваним результатом з огляду на архітектурні особливості. Навіть за умови ретельного налаштування та оптимізації параметрів обчислювальної складності у конфігураційному файлі genesis, спрямованого на мінімізацію ресурсних витрат, інтегрований механізм досягнення консенсусу, який відповідає за процеси формування, верифікації та валідації блоків у розподіленому реєстрі, об’єктивно сповільнює швидкість обробки вхідних запитів динамічними сервісами через необхідність криптографічних обчислень та синхронізації стану.

Проте ключовою і незаперечною перевагою динамічних хостів залишається їхня суттєво підвищена стійкість до DDoS-атак та інших форм зловмисних дій, спрямованих на порушення доступності та працездатності служб. Це робить їх значно надійнішим рішенням у критичних сценаріях

високого навантаження та цілеспрямованих атак. Важливо зауважити, що традиційні динамічні архітектури зазвичай мають централізовану структуру управління, що створює єдину точку відмови (single point of failure) та підвищує ризики несанкціонованої модифікації даних. Розроблена децентралізована blockchain-схема ефективно усуває ці критичні вразливості. На відміну від існуючих рішень, де динамічність застосовується виключно до приманок (honeypots), інноваційна схема забезпечує періодичну ротацію як приманок, так і автентичних сервісів, унеможливлюючи ідентифікацію справжніх вузлів зловмисником та гарантуючи комплексний захист інфраструктури.

Поряд з тим в експериментальному дослідженні ми зосереджуємось на детальному аналізі ефективності впровадження honeypot-систем як інноваційного інструменту для превентивного запобігання та оперативного виявлення DDoS-атак на ранніх стадіях їх розвитку. Центральну увагу приділено вивченню передових методів безперервного моніторингу мережевого трафіку, алгоритмам автоматичного виявлення аномальної поведінки та стратегіям ефективного використання накопичених даних для формування адаптивних моделей захисту, здатних еволюціонувати відповідно до змін у тактиках зловмисників. Практичні результати дослідження формують надійний фундамент для розробки інтелектуальних систем кіберзахисту нового покоління, спроможних проактивно ідентифікувати потенційні загрози ще до їх активації та суттєво підвищувати загальну стійкість критичних інформаційних інфраструктур до різноманітних типів DDoS-атак.

У межах науково-практичної частини роботи здійснюється багатоаспектний комплексний аналіз ефективності функціонування мережевої інфраструктури в екстремальних умовах реалізації SYN-флуд атаки. Всебічному оцінюванню підлягають ключові показники продуктивності як традиційних статичних хостів, так і гнучких динамічних серверів, які працюють відповідно до запропонованої інноваційної архітектурної схеми. Розроблена методологія тестування передбачає точну симуляцію реальних умов кібератаки шляхом безперервного автоматизованого генерування та цілеспрямованого відправлення

великої кількості SYN-пакетів із гнучко варійованою інтенсивністю трафікового потоку. Такий підхід уможливлює об’єктивне визначення критичних параметрів граничного навантаження, прецизійне вимірювання часу відгуку системи за різноманітними сценаріями атак та комплексне оцінювання загальної стійкості й надійності мережевої інфраструктури до аналогічних кіберзагроз.

Для забезпечення всебічного та детального аналізу функціонування мережевої інфраструктури під впливом значних навантажень було ретельно визначено та налаштовано специфічні параметри тестування. У рамках дослідження розмір SYN-пакетів для імітації атаки в утиліті Hping3 третьої версії встановлено на рівні 80 байт, що уможливлює ефективну сегментацію мережевого трафіку на окремі TCP-пакети заданого розміру. Детальний розподіл структури пакету передбачає загальний обсяг 80 байт, з яких 40 байт призначено для корисного навантаження (payload), тоді як інші 40 байт формують базовий заголовок без додаткових даних. Така конфігураційна схема вважається оптимальною для генерування максимально можливої кількості пакетів протягом визначеного часового інтервалу, що має критичне значення для проведення стрес-тестування мережевого обладнання та з’єднань.

Для проведення високоточних вимірювань основних показників продуктивності мережі, таких як пропускна здатність каналу зв’язку, тимчасові затримки передачі даних та відсоток втрачених пакетів, застосовується спеціалізований програмний інструмент Iperf третьої версії. Цей засіб забезпечує отримання комплексної статистичної інформації про роботу мережевого з’єднання в режимі реального часу з високою точністю та деталізацією.

У початковому стані мережевої інфраструктури, за повної відсутності зловмисного втручання та нульової інтенсивності атакуючих пакетів (базова конфігурація без шкідливого трафіку), всі категорії мережевих вузлів демонструють оптимальні експлуатаційні характеристики та досягають максимальних технічних показників ефективності. У цьому режимі пропускна здатність TCP-з’єднань досягає граничного значення 800 Мбайт/с, водночас

швидкість передачі TCP-трафіку стабілізується на позначці 100 Мбіт/с, що відповідає нормативним параметрам функціонування системи.

Графічні матеріали та діаграми, що наведені на рисунках 3.3 та 3.4, наочно ілюструють кореляційну залежність між інтенсивністю проведення DDoS-атаки та ключовими параметрами функціонування мережевої інфраструктури, зокрема ефективною пропускною здатністю та загальним обсягом переданого TCP-трафіку.

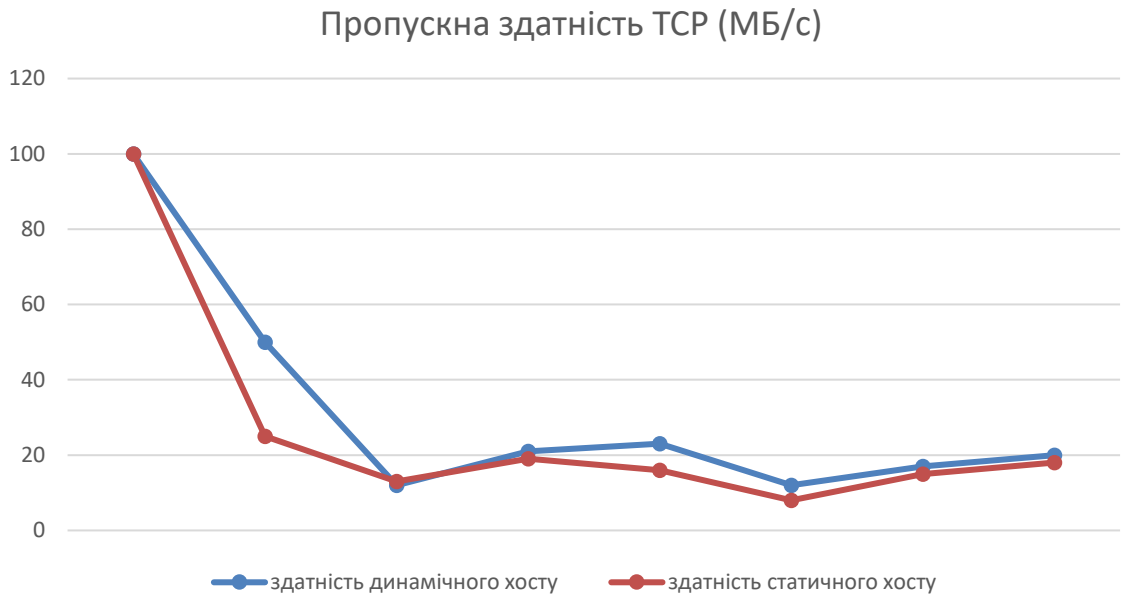

Рисунок 3.4 – Порівняння пропускної здатності TCP [2]

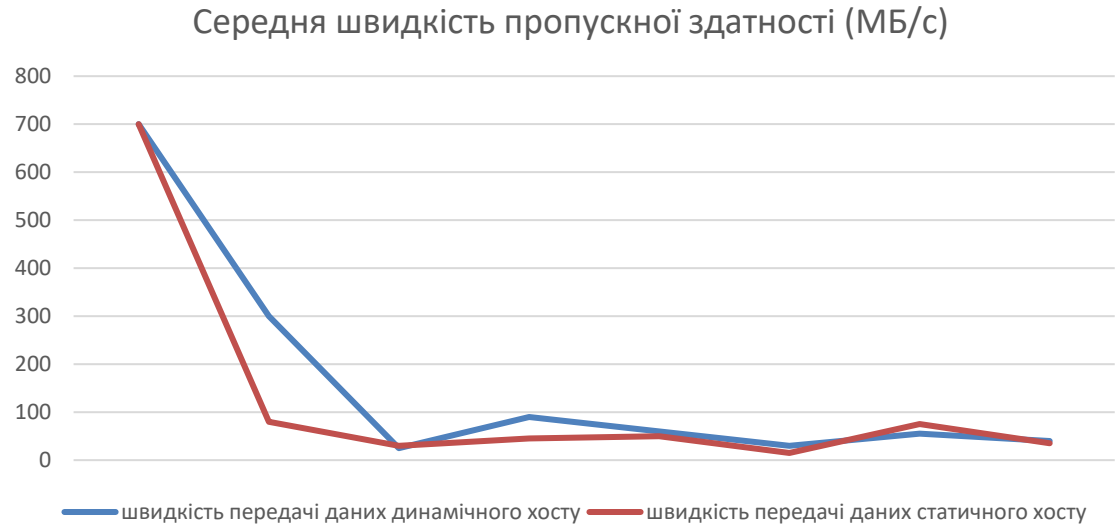

Рисунок 3.5 – Порівняння середньої швидкості пропускної здатності [2]

Однак із поступовим збільшенням інтенсивності шкідливого впливу та наростанням навантаження атакуючими пакетами фіксується критична деградація експлуатаційних характеристик мережевої інфраструктури. Найбільш катастрофічне погіршення продуктивності спостерігається в критичному діапазоні від 0 до 1000 пакетів за секунду. У цьому інтервалі зареєстровано різке, майже обвальне падіння обох досліджуваних метрик, що безпосередньо вказує на високу вразливість та недостатню стійкість мережевої архітектури до розподілених атак подібного типу та масштабу впливу.

Проводячи ретельний аналіз експериментальних результатів, представлених у графічному вигляді, можна дійти обґрунтованого висновку щодо істотних відмінностей у поведінці двох систем. Зокрема, крива продуктивності для статичних хостів демонструє різке падіння з більш вираженим кутом нахилу, тоді як траєкторія динамічних хостів характеризується нелінійною, ламаною конфігурацією з меншою інтенсивністю спаду показників.

Детальніше вивчаючи динаміку зміни параметрів на графіку, можна помітити поступове та порівняно повільне зростання навантаження в діапазоні від 1000 до 3000 пакетів за секунду. Принципово важливим спостереженням є той факт, що за будь-яких умов експерименту показники ефективності динамічних хостів стійко перевищують аналогічні значення статичних систем впродовж усього досліджуваного періоду.

Отримані емпіричні дані переконливо підтверджують, що динамічна архітектура програмних приманок (honeypots) володіє безперечною перевагою та демонструє значно вищу ефективність порівняно зі стаціонарними рішеннями. Це особливо помітно проявляється при оцінці критично важливих параметрів продуктивності мережевої інфраструктури та її спроможності обробляти інтенсивне навантаження за екстремальних умов експлуатації.

Для забезпечення об’єктивної оцінки та наочної демонстрації ефективності функціонування різних типів мережевих систем, важливо здійснити детальний порівняльний аналіз пропускної здатності між динамічним хостом (ДХ) та статичним хостом (СХ). Подібний методологічний підхід уможливлює

комплексне визначення того, якою мірою кожна з досліджуваних систем спроможна забезпечувати стабільність роботи та підтримувати високий рівень продуктивності в умовах активного проведення різноманітних кібератак. З метою об’єктивізації результатів дослідження застосовується методика розрахунку відсоткового співвідношення між показниками пропускних здатностей обох типів хостів.

Результати проведеного аналізу виявили, що отримане середнє відсоткове значення перевищення становить приблизно 0.43, що переконливо засвідчує суттєву технологічну перевагу динамічного хоста над статичним. Це свідчить про те, що в умовах активних кібернетичних атак динамічна система демонструє в середньому на 0.43 вищі показники ефективності у підтриманні та збереженні пропускної здатності мережі порівняно зі статичною конфігурацією. Такий значний результат обґрунтовується насамперед високою гнучкістю та адаптивністю архітектури динамічного хоста, який володіє можливістю швидко пристосовуватися до змін мережевого навантаження, оперативно реагувати на виявлені загрози безпеці та ефективно мінімізувати втрати трафіку. Натомість статичний хост характеризується жорстко фіксованими параметрами конфігурації, що істотно обмежує його функціональні можливості щодо протидії динамічним атакам у режимі реального часу.

Узагальнюючи результати проведеного дослідження, можна стверджувати, що експериментальні дані наочно підтверджують суттєву перевагу розробленої динамічної системи програмних приманок над традиційними статичними методами захисту інформаційної безпеки. Хоча інтеграція технології blockchain дійсно потребує залучення додаткових обчислювальних потужностей і певною мірою впливає на швидкодію динамічних хостів, цей вплив виявляється незначним і цілком прийнятним для практичного застосування. Більше того, запропонована архітектура демонструє помітне зниження загального навантаження на мережеву інфраструктуру у порівнянні з класичними фіксованими рішеннями, досягаючи оптимального співвідношення між рівнем безпеки та витратами системних ресурсів.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи представлено детальний аналіз отриманих експериментальних результатів, а також описано методику проведення дослідження, що дозволила забезпечити об’єктивність і достовірність висновків. Особлива увага приділяється оцінюванню сучасних технологій обману (deception technologies), які сьогодні відіграють важливу роль у побудові гнучких та адаптивних механізмів кіберзахисту. У ході практичної частини дослідження було встановлено, що використання honeypot-систем є одним з найбільш результативних підходів для виявлення, моніторингу та подальшого аналізу розподілених атак типу DDoS, що становлять значну загрозу для корпоративних та державних мережевих інфраструктур.

Під час роботи honeypot збирає цінні дані: IP-адреси джерел атак, часові характеристики, типи інструментів, застосовані скрипти, вектори проникнення, техніки обходу захисту та інші артефакти атакувальної активності. Отримана інформація аналізується та застосовується для розробки більш ефективних політик мережевого захисту, удосконалення механізмів виявлення вторгнень, раннього оповіщення про загрози та побудови превентивних стратегій. Таким чином, результати дослідження підтверджують значний потенціал honeypot-технологій як важливого інструменту підвищення рівня інформаційної безпеки.

Таким чином, використання honeypot-систем не тільки знижує ймовірність успішного проведення DDoS-атак, але й забезпечує формування гнучкої та адаптивної системи оборони, здатної вдосконалюватися завдяки аналізу реальних кіберінцидентів. У перспективі поява більш інтелектуальних honeypot-рішень, що базуються на алгоритмах машинного навчання, дасть змогу в автоматичному режимі ідентифікувати типи атак, оперативно реагувати на загрози та оптимізувати політики захисту. Отже, honeypot-технології стають ключовим компонентом сучасної стратегії кібербезпеки, орієнтованої не лише на виявлення, а й на запобігання DDoS-атакам.

ВИСНОВКИ

У кваліфікаційній роботі магістра було запропоновано програмну реалізацію та досліджено систему безпеки з використанням технологій honeypot та блокчейн. Для виконання поставлених завдань кваліфікаційної роботи було здійснено огляд і аналіз предметної області проблеми, результати існуючих теоретичних та експериментальних досліджень; огляд і аналіз методів та засобів безпеки з використанням honeypots для вирішення проблеми дослідження; обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання.

В кваліфікаційній роботі було виконано основні завдання дослідження, а саме:

– на основі аналізу сучасних підходів виявлено тенденцію до інтеграції honeypot з іншими технологіями для підвищення надійності збереження доказів атак;

– проаналізовано властивості блокчейн-технології: незмінність, децентралізація, прозорість, криптографічний захист та визначено переваги блокчейн для журналів безпеки: захист від підробки, можливість аудиту, timestamp-верифікація, розподілене зберігання;

– спроектовано смарт-контракти для автоматизованої реєстрації та верифікації інцидентів та розроблено механізми взаємодії між компонентами через API та черги повідомлень;

– розроблено прототип системи, який успішно продемонстрував технічну можливість інтеграції honeypot та блокчейн-технологій. Поряд з тим система показала високу надійність фіксації атак з нульовим відсотком втрат подій під час тестування;

– виявлено компроміс між швидкістю обробки подій та рівнем захисту: блокчейн додає затримку, але забезпечує абсолютну незмінність. Під час аналізу було виявлено, що гібридний підхід, який поєднує традиційні SIEM для

оперативного реагування та блокчейн для архівування критичних подій, виявився найбільш оптимальним;

– встановлено, що поєднання honeypot та блокчейн створює якісно новий рівень довіри до даних безпеки, що особливо важливо для форензики та судових розслідувань. Криптографічна гарантія цілісності журналів робить систему стійкою до атак типу «anti-forensics» та внутрішніх зловмисників.

Проведене дослідження продемонструвало, що інтеграція honeypot-систем з блокчейн-технологіями дозволяє вирішити ключові проблеми традиційних систем безпеки. Honeypot-компонент забезпечує активне виявлення загроз шляхом імітації вразливих сервісів та збору даних про методи атак, тоді як блокчейн гарантує незмінність та достовірність зібраних логів, що є критично важливим для судової експертизи та аналізу інцидентів.

Програмна реалізація системи показала її ефективність у детектуванні різних типів атак, включаючи спроби несанкціонованого доступу, сканування портів та експлуатацію вразливостей. Використання розподіленого реєстру для збереження інформації про інциденти безпеки унеможливлює підробку або видалення даних зловмисниками, навіть у випадку компрометації окремих вузлів системи.

Експериментальні дослідження підтвердили працездатність розробленої системи та виявили її переваги порівняно з традиційними підходами. Зокрема, система демонструє високу швидкість реагування на загрози, мінімальний рівень хибних спрацювань та можливість масштабування для захисту розподілених інфраструктур.

Водночас дослідження виявило певні обмеження, зокрема накладні витрати на запис транзакцій у блокчейн та необхідність балансування між детальністю логування і продуктивністю системи. Ці аспекти потребують подальшої оптимізації та можуть стати напрямком майбутніх досліджень.

Для підвищення ефективності виявлення та класифікації загроз рекомендується впровадити алгоритми машинного навчання та штучного інтелекту. Це дозволить системі автоматично розпізнавати нові типи атак на

основі аналізу поведінкових патернів, виявляти аномалії в трафіку та прогнозувати потенційні загрози. Доцільним є використання методів глибокого навчання для аналізу великих обсягів даних, зібраних honeypot-системою, що значно покращить точність детектування складних багатоетапних атак.

З метою зменшення накладних витрат на запис даних у блокчейн рекомендується впровадити механізми агрегації та пакетної обробки транзакцій. Доцільним є використання рішень другого рівня (Layer 2) або сайдчейнів для зберігання детальних логів з періодичною фіксацією хешів у основному блокчейні. Це дозволить суттєво знизити вартість операцій та підвищити швидкість обробки даних без втрати переваг незмінності та прозорості.

Для підвищення рівня захисту даних рекомендується інтеграція сучасних криптографічних технологій, таких як гомоморфне шифрування для аналізу зашифрованих даних, zero-knowledge proofs для підтвердження достовірності інформації без розкриття її змісту, та квантово-стійкі алгоритми для захисту від майбутніх загроз, пов'язаних з квантовими обчисленнями.

Доцільним є створення децентралізованої системи репутації для обміну інформацією про загрози між різними організаціями. Використання блокчейну для зберігання індикаторів компрометації (IoC) та даних про відомі загрози дозволить сформувати довірене середовище для колективного захисту. Організації зможуть отримувати актуальну інформацію про нові методи атак, зберігаючи при цьому конфіденційність власних даних.

Отримані результати мають як теоретичне, так і практичне значення для розвитку систем кібербезпеки. Розроблена архітектура може бути адаптована для захисту критичної інфраструктури, корпоративних мереж та хмарних сервісів. Перспективи подальшого розвитку включають впровадження машинного навчання для автоматизованого аналізу паттернів атак, інтеграцію з системами Security Information and Event Management (SIEM) та розробку розподілених honeypot-мереж з консолідованим блокчейн-реєстром загроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кошелюк В., Веремій І. Інтеграція Blockchain у механізми кіберпасток (Honeypots). IX Міжнародна науково-практична конференція «Development of science: theories, methodology, practice and technologies» (28-31 жовтня 2025 р.), Париж, Франція. International Science Group. 2025. С. 68-76.

2. Кошелюк В., Веремій І. Синергія honeypot і аналітики трафіку для ранньої детекції DDOS. 2 Міжнародна науково-практична конференція «Modern Perspectives on Science and Economic Progress» (5-7 листопада, 2025 р.), Вільнюс, Литва. International Scientific Unity. 2025. С. 166-170.

3. Андрущак І., Кошелюк В., Веремій, І. Технології обману в кібербезпеці: інтеграція cowrie та ELK Stack для виявлення атак на мережевий трафік. International Science Journal of Engineering. ISJEA, 2025. Pp. 1-14.

4. Mastering Honeypots: Art of deception for cybersecurity defense. Choudhary M. URL: https://www.business-vox.com/catalog/book/88961676?_ locale =en (accessed: 05.10.2025)

5. Blockchain and Machine Learning for IoT Security. Azrour M. et al. URL: https://www.taylorfrancis.com/books/edit/10.1201/9781003438779/blockchain-machine-learning-iot-security-mourade-azrour-jamal-mabrouki-azidine-guezzaz-said-benkirane (accessed: 05.10.2025)

6. Vasylyshyn S., Lakhno V., Alibiyeva N., Alibiyeva Z., Sauanova K., Pleskach V., Lakhno M. Information technologies for the synthesis of rule databases of an intelligent lighting control system. M. Journal of Theoretical and Applied Information Technologythis link is disabled. 2022. Vol. 100(5). Pp. 1340-1353.

7. Blockchain for Cybersecurity in Cyber-Physical Systems. Maleh Y. URL: https://dokumen.pub/blockchain-for-cybersecurity-in-cyber-physical-systems-9783031255052-9783031255069.html (accessed: 08.10.2025)

8. Blockchain Technology for Cyber Defense, Cybersecurity, and Countermeasures: Techniques, Solutions, and Applications. Naresh K. et al. URL: https://www.routledge.com/Blockchain-Technology-for-Cyber-Defense-Cybersecu

rity-and-Countermeasures-Techniques-Solutions-and-Applications/Kshetri-Pandey-Ahmed/p/book/9781032583037 (accessed: 08.10.2025)

9. A Comprehensive Guide for Web3 Security: From Technology, Economic and Legal Aspects (Future of Business and Finance). Huang K. et al. URL: https://www.springerprofessional.de/en/a-comprehensive-guide-for-web3-security/26568874 (accessed: 13.10.2025)

10. Vasylyshyn S., Opirskyy I., Shevchenko S. Honeypot Security Efficiency versus Deception Solution. Paper presented at the CEUR Workshop Proceedings. 2021. Vol. 3188. Pp. 229-236.

11. The Complete Guide to the ELK Stack. Horovits D. URL: https://logz.io/learn/complete-k/?utm_source=chatgpt.com (accessed: 15.10.2025)

12. Markowitch O., Dricot J-M. IoT Security: Threat Detection, Analysis and Defense. MDPI reprint, 2025. 254 p.

13. Blockchain Security and Its Application in Internet of Things. Chun-Ta Li. URL: https://www.mdpi.com/books/reprint/10958-blockchain-security-and-its-application-in-internet-of-things (accessed: 03.11.2025)

14. Blockchain and Machine Learning for IoT Security. Azrour M. et al. URL: https://www.taylorfrancis.com/books/edit/10.1201/9781003438779/blockchain-machine-mabrouki-azidine-guezzaz-said -benkirane (accessed: 04.11.2025)

15. Sharma S. et al. Federated Learning and Blockchain: A Cross-Domain Convergence. 3rd International Conference on Technological in Computational Sciences (ICTACS). Pp. 1121-1127.

16. Learning ELK Stack: Build mesmerizing visualizations, analytics, and logs from your data using Elasticsearch, Logstash, and Kibana. Chhajed S. URL: https://www.scholarvox.com/catalog/88853379?_locale=en (accessed: 08.11.2025)

17. Enterprise blockchain. That actually works. URL: https://www.multichain.com/ (accessed: 10.11.2025)

18. Vasylyshyn S. et al. Information technologies for the synthesis of rule databases of an intelligent lighting control system. Journal of Theoretical and Applied Information Technologythis link is disabled, 2022. 100(5), Pp. 1340–1353

ДОДАТКИ

ДОДАТОК А

Апробація результатів дослідження

ДОДАТОК Б

International Science Journal of Engineering & Agriculture
2025; 4(6): 1-14
https://isg-journal.com/isjea
doi: 10.46299/j.isjea.20250406.01
ISSN: 2720-6319

INTERNATIONAL SCIENCE GROUP

Технології обману в кібербезпеці: інтеграція cowrie та ELK Stack для виявлення атак на мережевий трафік

Ігор Андрущак
Луцький національний технічний університет, Луцьк, Україна
ORCID 0000-0002-8751-4420

Віктор Кошелюк
Луцький національний технічний університет, Луцьк, Україна
ORCID 0000-0002-4136-5087

Ілля Веремій
Луцький національний технічний університет, Луцьк, Україна
ORCID 0009-0003-1381-9920

Анотація: У статті досліджуються технології обману (Deception Technologies) у сфері кібербезпеки з метою підвищення ефективності виявлення атак на мережевий трафік. Основна мета роботи полягає у практичному дослідженні інтеграції системи Cowrie, що забезпечує емуляцію серверів протоколів Secure Shell та Telnet, із платформою Elastic Stack (Elasticsearch, Logstash, Kibana) для збору, обробки та візуалізації даних про підозрілу активність. Дослідження спрямоване на розробку методики налаштування honeypot-системи та її інтеграції в централізовану аналітичну систему з метою своєчасного реагування на потенційні загрози. Методи дослідження включають аналіз сучасних технологій обману та їх порівняння, конфігурацію та оптимізацію cowrie для збору логів про мережеві атаки, а також інтеграцію отриманих даних у Elastic Stack. Для оцінки ефективності системи використовувався експериментальний підхід із фіксацією типів атак, джерел вторгнень та поведінки зловмисників у мережі. Візуалізація даних у Kibana дозволила здійснювати детальний аналіз активності та формувати оперативні звіти про інциденти безпеки. Результати дослідження показали, що інтегрована система cowrie та Elastic Stack ефективно фіксує та класифікує спроби несанкціонованого доступу, атаки на паролі та інші типові мережеві загрози. Було встановлено, що комбінування емуляції пасток із централізованим аналізом даних дозволяє підвищити швидкість виявлення атак та покращити розуміння поведінки зловмисників. Налаштування логування та кореляції даних забезпечує гнучкий підхід до моніторингу та аналітики безпеки. Наукова новизна дослідження полягає у розробці інтегрованого підходу до застосування технологій обману в кібербезпеці з використанням honeypot cowrie та платформи аналізу даних Elastic Stack для виявлення, моніторингу та аналізу атак на мережевий трафік. Запропоновано методику автоматизованого збору та кореляції даних про несанкціоновані дії, що дозволяє підвищити точність детекції та швидкість реагування на загрози. Дослідження демонструє ефективність інтеграції емульованих пасток із системою візуалізації та аналітики, забезпечуючи більш глибоке розуміння поведінки атакуючих та можливість прогнозування потенційних сценаріїв вторгнень у мережеву інфраструктуру. У висновках підкреслюється, що технології обману є ефективним інструментом протидії кіберзагрозам, а їх інтеграція з платформами аналітики, такими як Elastic Stack, значно підвищує спроможність організацій реагувати на інциденти. Рекомендації включають впровадження подібних систем у корпоративних та хмарних середовищах, регулярне оновлення конфігурацій honeypot та вдосконалення методів візуалізації й аналізу даних. Майбутні напрями розвитку передбачають автоматизацію реагування на інциденти на основі зібраних даних, інтеграцію з системами управління інформацією та подіями безпеки, а також використання машинного навчання для прогнозування та попередження нових типів атак.

Ключові слова: cowrie, deception technologies, honeypot, ELK Stack, кіберзагрози.

1. Вступ

Незважаючи на багаторічні дослідження та розвиток технологій у сфері комп'ютерної безпеки, вона залишається складною і вразливою галуззю, у якій традиційні підходи часто не забезпечують повного захисту від сучасних загроз. Хронічні проблеми безпеки інформаційних систем підкреслюють необхідність пошуку інноваційних методів, здатних доповнювати класичні механізми контролю та захисту. Одним із перспективних напрямів є застосування стратегій обману (deception technologies) як активного інструменту кіберзахисту. У повсякденному житті обман використовується для створення ілюзії безпеки, наприклад, увімкнене світло в будинку може дати зловмиснику хибне уявлення про присутність мешканців. В IT-системах обман застосовується значно рідше і часто має прихований характер, проте його потенціал для зміни поведінки нападника є суттєвим [1, 2].

На відміну від традиційних методів безпеки, які здебільшого спрямовані на виявлення або блокування атак, обман діє через моделювання середовища та маніпулювання сприйняттям зловмисника, змушуючи його здійснювати дії, вигідні захиснику. Такий підхід дозволяє компенсувати слабкі сторони класичних механізмів і створює додатковий рівень стійкості системи. Інтеграція обману із традиційними засобами безпеки може суттєво підвищити ефективність захисту, роблячи інформаційні системи більш адаптивними та важкодоступними для кібератак.

У сфері кібербезпеки використання обману залишається недостатньо розвиненим, хоча він може стати критично важливим інструментом захисту [3]. Дослідження військових стратегій обману демонструють, що ефективне введення супротивника в оману є складною професійною навичкою, яка потребує глибокого розуміння принципів, методів та психології дезінформації. У контексті інформаційних систем обман може застосовуватися для створення кібер-пасток, таких як honeypot-системи, які впливають на процеси прийняття рішень зловмисниками, відволікають їхні дії або збирають інформацію про атаки. Такі технології обману забезпечують додатковий рівень захисту, підвищують ефективність превентивних заходів і заповнюють прогалини традиційних методів кібероборони, стаючи ключовим елементом сучасної стратегії інформаційної безпеки.

Термін «IT-безпека» охоплює широкий спектр технічних, організаційних та процедурних заходів, спрямованих на захист визначеного набору інформаційних активів від різноманітних загроз. Ці активи зазвичай класифікують відповідно до ключових цілей безпеки, відомих як CIAA: конфіденційність, цілісність, автентифікація та доступність. Конфіденційність гарантує, що інформація доступна лише уповноваженим користувачам; цілісність забезпечує недоторканість даних і запобігає їх несанкціонованим змінам; автентифікація підтверджує особу користувачів і джерел даних; доступність гарантує безперервність доступу до інформаційних ресурсів. Класична IT-безпека здебільшого зосереджується на досягненні цих цілей у традиційних IT-системах і на мінімізації ризику несанкціонованих дій [4, 5].

Deception Technologies (DT) – сучасний підхід, що не завжди безпосередньо відображає класичні цілі CIAA, але орієнтований на більш абстрактні та проактивні завдання, такі як виявлення вторгнень, моніторинг підозрілої активності та аналіз поведінки потенційних зловмисників. DT особливо ефективні в хмарних середовищах, де масштабованість та динамічність ресурсів дозволяють створювати пастки та фальшиві системи, які імітують реальні сервіси. Однією з ключових особливостей DT є використання концепції Security by Obscurity (SbO) – навмисного ускладнення або приховування деталей системи, що ускладнює зловмиснику розуміння її структури. Хоча SbO не рекомендується як основний метод у традиційній безпеці, у DT він слугує ефективним інструментом уповільнення атакуючих і відволікання від реальних цілей. Особливо це важливо у хмарних інфраструктурах, де невідомість наявності пастки значно підвищує шанси на своєчасне виявлення зловмисної активності та мінімізацію ризиків для критично важливих сервісів.

ДОДАТОК В

TCP-honeypot

```
import socket
import threading
from logger import log_event
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv("honeypot_log.csv", names=["timestamp", "ip",
"port", "data"])
top_attackers = df['ip'].value_counts().head(10)

plt.bar(top_attackers.index, top_attackers.values)
plt.xticks(rotation=45)
plt.ylabel("Кількість атак")
plt.title("Топ 10 IP, що атакували honeypot")
plt.show()
HOST = "0.0.0.0"
PORT = 2222   #
def handle_client(conn, addr):
    ip, port = addr
    log_event("tcp_connect", ip, port, "Connection
established")

    with conn:
        try:
            while True:
                data = conn.recv(1024)
                if not data:
                    break
                decoded = data.decode(errors="ignore")
                log_event("tcp_data", ip, port, decoded)
                # фейкова відповідь зловмиснику
                conn.sendall(b"OK\r\n")
        except Exception as e:
            log_event("tcp_error", ip, port, str(e))
def start_honeypot():
    print(f"TCP Honeypot listening on {HOST}:{PORT}")
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as
s:
        s.bind((HOST, PORT))
        s.listen()
        while True:
            conn, addr = s.accept()
            threading.Thread(target=handle_client,
                             args=(conn, addr),
```

```
                                    daemon=True).start()
    if __name__ == "__main__":
        start_honeypot()
```

Логування підключень у файл

```
    import csv
    def log_connection(addr, data):
        with open("honeypot_log.csv", mode="a", newline="") as
file:
            writer = csv.writer(file)
            writer.writerow([datetime.now(), addr[0], addr[1],
data])
    # у функції handle_client
    log_connection(addr, data.decode(errors='ignore'))
    from http.server import BaseHTTPRequestHandler, HTTPServer
    class HoneypotHTTPRequestHandler(BaseHTTPRequestHandler):
        def do_GET(self):
            print(f"[{datetime.now()}] GET від
{self.client_address}: {self.path}")
            self.send_response(200)
            self.end_headers()
            self.wfile.write(b"<h1>Welcome to Python
Honeypot</h1>")
        def do_POST(self):
            content_length = int(self.headers.get('Content-
Length', 0))
            post_data = self.rfile.read(content_length)
            print(f"[{datetime.now()}] POST від
{self.client_address}: {post_data.decode(errors='ignore')}")
            self.send_response(200)
            self.end_headers()

    server_address = ('', 8080)
    httpd = HTTPServer(server_address, HoneypotHTTPRequestHandler)
    print("HTTP Honeypot запущено на порті 8080")
    httpd.serve_forever()
```

Інтеграція з Cowrie (SSH/FTP honeypot)

```
    import json
    LOG_FILE = "/path/to/cowrie/log/cowrie.json"
    def parse_cowrie_log():
        with open(LOG_FILE, "r") as f:
            for line in f:
```

```
                try:
                    entry = json.loads(line)
                    if entry.get('eventid') ==
"cowrie.session.connect":
                        print(f"Підключення від
{entry['src_ip']}:{entry['src_port']}")
                except json.JSONDecodeError:
                    continue
    parse_cowrie_log()
```

Фальшивий HTTP сервер

```
    from http.server import BaseHTTPRequestHandler, HTTPServer
    from logger import log_event
    from datetime import datetime
    class HoneypotHandler(BaseHTTPRequestHandler):
        def log_request_data(self, method):
            length = int(self.headers.get("Content-Length", 0))
            body = self.rfile.read(length) if length > 0 else b''
            data = body.decode(errors="ignore")
            ip, port = self.client_address

            log_event(f"http_{method.lower()}", ip, port,
                      f"path={self.path},
headers={dict(self.headers)}, body={data}")

        def do_GET(self):
            self.log_request_data("GET")
            self.send_response(200)
            self.end_headers()
            self.wfile.write(b"<h1>Welcome to IoT Honeypot</h1>")

        def do_POST(self):
            self.log_request_data("POST")
            self.send_response(200)
            self.end_headers()

    server = HTTPServer(("0.0.0.0", 8080), HoneypotHandler)
    print("HTTP Honeypot running on port 8080")
    server.serve_forever()
```