МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
Циклова комісія комп’ютерної інженерії та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙНУ ДЛЯ ЗАХИСТУ МЕРЕЖЕВОГО ТРАФІКА

Виконав: студент групи 2К-22
Спеціальності:
123 Комп’ютерна інженерія
Богдан РОЗПУТНІЙ
Керівник:
Маргарита МЕДОЛИЗ

Черкаси 2026

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ

Циклова комісія комп’ютерної інженерії та інформаційних технологій

Спеціальність 123 «Комп’ютерна інженерія»

Освітня програма Комп’ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувачка циклової комісії КІ та ІТ

_____Наталя ФАЛЬЧЕНКО

(підпис)

«_____» _____2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Розпутньому Богдану Олексійовичу

1. Тема кваліфікаційної роботи «Використання технології блокчейну для захисту мережевого трафіку»

   Керівник роботи Медолиз Маргарита Миколаївна, викладач вищої категорії, затверджені наказом закладу передвищої освіти від «06» жовтня 2025 року № 84/1у.
2. Строк подання студентом кваліфікаційної роботи 08.06.2026
3. Вихідні дані до кваліфікаційної роботи: нормативні документи та Стандарти захисту, джерела з інформацією про блокчейн та мережеву безпеку, наукові статті та електронні ресурси, середовище моделювання
4. Зміст кваліфікаційної роботи (питання, що необхідно розглянути): Теоретичні аспекти забезпечення цілісності мережевих журналів подій у корпоративних мережах підприємств; Аналіз уразливостей традиційних систем журналювання та реалізація системи захисту на основі блокчейну
5. Дата видачі завдання 15.10.2025 р

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	14.10.2025	
2	Розділ 1 Теоретичні аспекти забезпечення цілісності мережевих журналів подій у корпоративних мережах підприємств	09.12.2025	
3	Розділ 2 Аналіз уразливостей традиційних систем журналювання та реалізація системи захисту на основі блокчейну	10.03.2026	
4	Розділ 3 Реалізація прототипу системи забезпечення цілісності мережевих журналів на основі блокчейну та аналіз результатів	28.04.2026	
5	Висновки	12.05.2026	
6	Оформлення кваліфікаційної роботи (чистовий варіант)	26.05.2026	
7	Перевірка кваліфікаційної роботи на наявність ознак плагіату (за 10 днів до захисту)	05.06.2026	
8	Подання кваліфікаційної роботи на затвердження завідувачу ЦК (за 7 днів до захисту)	08.06.2026	

Студент _____ Богдан РОЗПУТНІЙ
(підпис)

Керівник роботи _____ Маргарита МЕДОЛИЗ
(підпис)

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ACL	Access Control List
APT	Advanced Persistent Threat
GDPR	General Data Protection Regulation
PCI DSS	Payment Card Industry Data Security Standard
SIEM	Security Information and Event Management
NetFlow	мережевий протокол експорту інформації про потоки трафіку
PoE	Power over Ethernet
QoS	Quality of Service
Syslog	System Logging Protocol

ЗМІСТ

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій корпоративні мережі підприємств стали критичною основою функціонування практично всіх бізнес-процесів. Постійна взаємодія між користувачами, серверами, мережевим обладнанням та інформаційними системами призводить до генерації величезних обсягів подій, пов’язаних з авторизацією, передачею даних, доступом до ресурсів, зміною конфігурацій та роботою сервісів. Журнали цих подій є одним з найважливіших джерел інформації для моніторингу стану мережі, виявлення аномалій, розслідування кіберінцидентів та проведення аудиту інформаційної безпеки.

Разом з тим традиційні централізовані системи журналювання, які домінують на більшості підприємств, мають суттєві системні недоліки. Вони зберігають усі логи на одному або обмеженій кількості серверів, що перетворює їх на єдину точку компрометації. Зловмисник, отримавши адміністративний доступ до такого сервера, може видалити, модифікувати або повністю очистити записи про свою діяльність, значно ускладнивши або зробивши неможливим подальше розслідування інциденту. Ця проблема особливо актуальна в умовах зростання кількості цілеспрямованих кібератак з боку професійних APT-груп, які свідомо спрямовують зусилля на приховування своїх слідів у системі.

Питання забезпечення цілісності мережевих журналів подій активно вивчається в науковій літературі та звітах провідних компаній з кібербезпеки. Однак більшість існуючих рішень зосереджена на посиленні захисту самих серверів або використанні розподілених систем зберігання, тоді як технологія блокчейн, яка дозволяє забезпечити криптографічну незмінність даних, вивчена ще недостатньо глибоко саме в контексті захисту мережевих журналів корпоративних мереж.

Актуальність обраної теми зумовлена необхідністю створення таких механізмів захисту, які б гарантували незмінність і достовірність журналів подій навіть у разі повної компрометації окремих компонентів інфраструктури. Це особливо важливо для підприємств критичної інфраструктури, банківської сфери, державних установ та компаній, які обробляють великі обсяги конфіденційної інформації, де втрата цілісності логів може призвести до серйозних фінансових втрат, юридичної відповідальності та втрати довіри клієнтів.

Метою кваліфікаційної роботи є розробка системи забезпечення цілісності мережевих журналів подій на основі технології блокчейн.

Для досягнення поставленої мети необхідно вирішити такі завдання: проаналізувати архітектуру сучасних корпоративних мереж підприємств та особливості генерації мережевих подій;

- розглянути традиційні системи журналювання подій та SIEM-системи, визначити їхні основні недоліки;
- дослідити нормативні вимоги до забезпечення цілісності журналів подій;
- проаналізувати реальні кіберінциденти з метою виявлення типових методів маніпуляції журналами;
- розробити архітектуру та алгоритм роботи системи захисту журналів подій на основі технології блокчейн;
- реалізувати програмний прототип системи та провести його тестування;
- проаналізувати отримані результати та визначити перспективи практичного застосування запропонованого рішення.

Об’єктом дослідження є процес забезпечення інформаційної безпеки мережевих журналів подій у корпоративних мережах підприємств.

Предметом дослідження виступають методи та технологічні рішення забезпечення криптографічної цілісності та незмінності цих журналів з використанням технології блокчейн.

Практичне значення роботи полягає в розробці додаткового механізму захисту цілісності мережевих журналів подій на основі технології приватного блокчейну, який може бути інтегрований як додатковий захисний шар до вже існуючих SIEM-систем. Запропоноване рішення дозволяє суттєво підвищити стійкість корпоративної інфраструктури до цілеспрямованих кібератак, зокрема до дій зловмисників, спрямованих на приховування слідів шляхом модифікації або видалення логів.

РОЗДІЛ 1 ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ МЕРЕЖЕВИХ ЖУРНАЛІВ ПОДІЙ У КОРПОРАТИВНИХ МЕРЕЖАХ ПІДПРИЄМСТВ

У цьому розділі розглядається класична архітектура корпоративної мережі підприємства на основі матеріалів офіційної документації компанії Cisco Systems. Дана модель широко використовується при проєктуванні сучасних комп'ютерних мереж підприємств та організацій.

1.1 Архітектура сучасних корпоративних мереж підприємств

Сучасні корпоративні мережі підприємств є складними високотехнологічними розподіленими системами, які забезпечують безперервну взаємодію між користувачами, серверами, мережевим обладнанням, системами зберігання даних та різноманітними прикладними сервісами. Для ефективного проєктування, масштабування та управління такими мережами вже протягом багатьох років застосовується трирівнева ієрархічна модель, розроблена компанією Cisco Systems. Ця модель, описана в документі Enterprise Campus 3.0 Architecture, стала промисловим стандартом і широко використовується на підприємствах різного масштабу та відображена на рисунку 1.1.

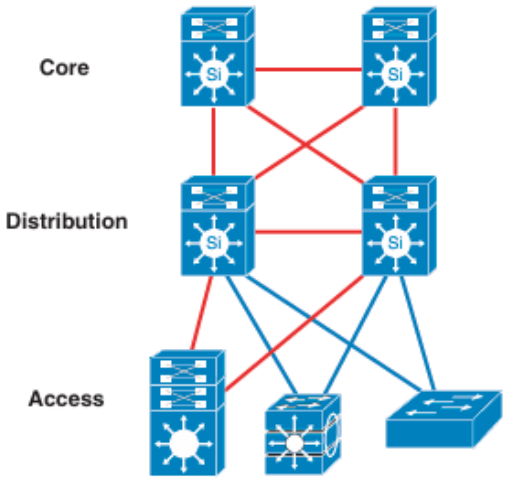

Рисунок 1.1 – Схема трирівневої архітектури корпоративної мережі

Трирівнева архітектура передбачає чіткий розподіл функцій між рівнями. Рівень доступу є найбільш численним і забезпечує безпосереднє підключення кінцевих пристроїв користувачів: персональних комп'ютерів, ноутбуків, принтерів, IP-телефонів, камер відеоспостереження, точок бездротового доступу та IoT-пристроїв. На цьому рівні відбувається первинна комутація трафіку на другому рівні моделі OSI, базовий контроль доступу за допомогою технологій Port Security та 802.1X, сегментація мережі за допомогою віртуальних локальних мереж VLAN, а також живлення кінцевих пристроїв за технологією Power over Ethernet. Типовими пристроями цього рівня є комутатори серій Cisco Catalyst 9200, 9300 та 2960.

Рівень розподілу виконує роль агрегаційного та політико-контрольного центру мережі. Він збирає трафік з кількох комутаторів рівня доступу, здійснює маршрутизацію між VLAN, застосовує політики безпеки через списки контролю доступу ACL, фільтрує трафік, контролює broadcast-трафік, забезпечує якість обслуговування QoS та резервізацію шляхів. На цьому рівні також відбувається агрегація посилань за технологією EtherChannel. Типовими пристроями є потужні комутатори та маршрутизатори Cisco Catalyst 9300, 9400 та Nexus серії.

Магістральний рівень є високошвидкісним ядром усієї мережі. Його головне завдання забезпечити максимально швидку та надійну передачу даних між різними сегментами мережі, між рівнями розподілу, а також підключення до центрів обробки даних та зовнішніх мереж WAN. На цьому рівні пріоритетом є висока продуктивність і мінімальна затримка, тому тут зазвичай не застосовуються складні політики фільтрації чи контролю доступу вони виконуються на рівні розподілу. Типовими пристроями є високопродуктивні маршрутизатори та комутатори Cisco Catalyst 9500, Nexus 9000 серії.

Така ієрархічна структура забезпечує чітке розмежування обов'язків, спрощує управління мережею, підвищує її масштабованість та відмовостійкість, порівняння рівнів мережі відображено в таблиці 1.1. У невеликих мережах рівні Core та Distribution часто об'єднують в один для

зниження вартості (модель Collapsed Core). У великих підприємствах і дата-центрах можуть застосовувати додаткові рівні, наприклад Spine-Leaf архітектуру, для ще більшої продуктивності та горизонтального масштабування.

Таблиця 1.1 – Порівняльна характеристика рівнів корпоративнох мережі

Рівень мережі	Основні функції	Типове обладнання	Основні генеровані події
Access Layer	Підключення кінцевих пристроїв, PoE, VLAN	Cisco Catalyst 9200, 9300	Підключення/відключення пристроїв, 802.1X, Port Security
Distribution Layer	Агрегація трафіку, маршрутизація, ACL, QoS	Cisco Catalyst 9300, 9400, Nexus	Зміни маршрутизації, застосування ACL, broadcast контроль
Core Layer	Високошвидкісна передача даних	Cisco Catalyst 9500, Nexus 9000	Перевантаження каналів, зміни топології

Важливою особливістю сучасних корпоративних мереж є їхня гібридність. Багато організацій поєднують локальну інфраструктуру з хмарними сервісами Microsoft Azure, Amazon AWS та Google Cloud. Це вимагає надійного захищеного підключення між локальною мережею та хмарою, що ще більше підвищує обсяг генерованих подій і вимоги до їхнього журналювання.

Таким чином, архітектура сучасних корпоративних мереж є багатошаровою, ієрархічною та високотехнологічною. Саме в такій інфраструктурі постійно генеруються великі обсяги мережевих подій, які потребують ефективного журналювання та надійного захисту їхньої цілісності. Від правильності побудови мережі безпосередньо залежить обсяг і якість інформації, яка надходитиме до систем моніторингу та журналювання.

1.2 Генерація та види мережевих подій

У процесі функціонування сучасної корпоративної мережі постійно генеруються значні обсяги подій, які відображають усі аспекти роботи інфраструктури. Кожна дія користувача, робота мережевого обладнання,

сервера чи прикладної системи супроводжується створенням відповідних записів у журналах. Ці події є основним джерелом інформації для моніторингу стану мережі, виявлення аномалій, розслідування інцидентів інформаційної безпеки та проведення аудиту.

Генерація подій відбувається на всіх рівнях мережевої архітектури. На рівні доступу комутатори фіксують підключення та відключення пристроїв, спроби авторизації через протокол 802.1X, порушення політики Port Security, помилки живлення PoE та інші локальні події. На рівні розподілу маршрутизатори та комутатори реєструють події маршрутизації між VLAN, застосування правил фільтрації ACL, зміни в таблицях MAC-адрес, контроль broadcast-трафіку та порушення політик безпеки. На магістральному рівні фіксуються події високошвидкісної комутації, перевантаження каналів, зміни в топології мережі та критичні системні помилки.

Найбільший обсяг подій генерують сервери. У середовищі Windows Server основним джерелом інформації є система Windows Event Logs, яка поділяється на кілька ключових журналів. Журнал Security фіксує події авторизації, доступу до об’єктів, зміни прав користувачів та аудит політики безпеки. Журнал System реєструє системні події, такі як запуск і зупинка служб, помилки драйверів, проблеми з апаратним забезпеченням та критичні системні збої. Журнал Application містить події, що генеруються прикладними програмами. Крім того, існують журнали Setup, ForwardedEvents та інші.

На мережевому обладнанні Cisco основним протоколом журналювання є Syslog. Він використовується для фіксації подій на комутаторах, маршрутизаторах, міжмережевих екранах та точках бездротового доступу. Syslog дозволяє передавати повідомлення з різними рівнями серйозності від критичних до налагоджувальних. Іншим важливим видом даних є NetFlow та IPFIX, які збирають детальну статистику про мережевий трафік: джерело та призначення, використаний протокол, обсяг переданих даних та тривалість сесії. Ця інформація особливо корисна для виявлення аномального трафіку, DDoS-атак та аналізу поведінки користувачів.

Значний обсяг даних також генерують прикладні системи: системи управління базами даних, наприклад SQL Server, Oracle, веб-сервери, наприклад IIS, Apache, Nginx, системи електронної пошти, наприкад Exchange, ERP-системи, CRM-платформи та системи контролю фізичного доступу. Кожна з цих систем створює власні журнали, які мають специфічну структуру, формат і рівень деталізації.

Усі ці види журналів суттєво відрізняються за обсягом, частотою генерації та рівнем критичності. Журнали Security вважаються найбільш цінними з точки зору безпеки, оскільки містять інформацію про спроби несанкціонованого доступу, зміни прав та аудит політики. Водночас журнали Application можуть бути дуже об’ємними, але менш критичними для оперативного реагування. У великих підприємствах загальний обсяг генерованих подій може сягати десятків і навіть сотень гігабайт на добу, що створює серйозні вимоги до систем зберігання, обробки та захисту цих даних.

Таким чином, генерація мережевих подій є постійним, масовим і багатошаровим процесом, який охоплює всі рівні мережевої інфраструктури. Різноманітність видів журналів створює великий обсяг інформації, що потребує ефективних механізмів централізованого збору, зберігання та особливо захисту їхньої цілісності. Саме ця потреба обумовлює актуальність пошуку нових технологічних рішень, таких як блокчейн, здатних забезпечити криптографічну незмінність даних.

1.3 Системи журналювання подій та SIEM

Для ефективної роботи з величезними обсягами подій, що генеруються в корпоративній мережі, підприємства застосовують спеціалізовані системи журналювання. На початковому етапі розвитку таких систем використовувалися прості централізовані сервери syslog, на які з різних пристроїв надсилалися повідомлення. Згодом ці рішення еволюціонували в потужні комплексні платформи, відомі як SIEM-системи.

SIEM-система являє собою комплексне програмно-апаратне рішення, призначене для централізованого збору, нормалізації, кореляції, аналізу та зберігання подій безпеки й операційних подій з усьєї IT-архітектури підприємства. Вона збирає дані з комутаторів, маршрутизаторів, серверів Windows, Linux, мережевих екранів, систем виявлення вторгнень, прикладних програм, баз даних та хмарних сервісів. Після отримання подій система виконує їх нормалізацію, тобто приводить дані з різних джерел до єдиного формату, що дозволяє проводити подальший аналіз.

Однією з ключових функцій SIEM є кореляція подій. Система здатна виявляти причинно-наслідкові зв’язки між подіями, які на перший погляд здаються не пов’язаними. Наприклад, кілька невдалих спроб авторизації з одного IP-адресу, за якими слідує успішний вхід і подальше копіювання великих обсягів даних, може бути автоматично розпізнано як спробу компрометації облікового запису. SIEM також забезпечує довгострокове зберігання подій, побудову візуальних звітів, моніторинг у реальному часі та генерацію сповіщень про інциденти.

На ринку існує чимало відомих SIEM-рішень. Microsoft Sentinel працює на базі хмарної платформи Azure і добре інтегрується з екосистемою Microsoft. Splunk є одним з лідерів ринку завдяки потужним можливостям пошуку та аналізу великих даних. ELK Stack є популярним відкритим рішенням, яке часто використовують через гнучкість і відносно низьку вартість. IBM QRadar, ArcSight, LogRhythm та SearchInform також займають значну частку ринку, особливо в державних структурах і великих корпораціях.

Незважаючи на високий рівень розвитку, традиційні SIEM-системи мають суттєві недоліки. Вони є централізованими рішеннями, тобто всі дані зберігаються в одному або кількох сховищах, які стають єдиною точкою відмови та привабливою ціллю для атак. Якщо зловмисник отримує доступ з правами адміністратора до SIEM-сервера або до бази даних з журналами, він може видалити, змінити або підчистити критичні записи, приховуючи сліди свого перебування в мережі. Саме ця вразливість централізованого зберігання

даних робить традиційні системи журналювання недостатньо надійними в умовах сучасних цілеспрямованих атак.

Таким чином, системи SIEM значно підвищують ефективність моніторингу та реагування на інциденти, але не забезпечують достатнього рівня гарантій цілісності та незмінності зібраних подій. Це створює потребу в додаткових технологічних механізмах, здатних забезпечити криптографічну незмінність журналів навіть у разі компрометації самої системи зберігання.

1.4 Нормативні вимоги та значення цілісності журналів подій

Забезпечення цілісності та достовірності журналів подій є не лише технічним завданням, а й обов’язковою нормативною вимогою для більшості сучасних організацій. Міжнародні та національні стандарти чітко визначають необхідність захисту аудиторських слідів від несанкціонованих змін, видалення або підміни.

Одним з основоположних міжнародних стандартів у сфері інформаційної безпеки є ISO/IEC 27001:2022 «Інформаційні технології. Методики захисту. Системи управління інформаційною безпекою. Вимоги». Стандарт встановлює обов’язок організацій здійснювати реєстрацію та моніторинг подій безпеки, а також забезпечувати захист журналів подій від несанкціонованого доступу, зміни та видалення. Крім того, організації повинні зберігати журнали в такому вигляді, який дозволяє перевірити їхню цілісність.

Аналогічні вимоги до захисту журналів подій містяться в стандарті PCI DSS (Вимога 10), європейському регламенті GDPR (статті 5 та 32), а також у національних нормативних документах України, зокрема в законах «Про захист інформації в інформаційно-телекомунікаційних системах» та «Про основні засади забезпечення кібербезпеки України».

Таким чином, забезпечення цілісності журналів подій є не лише технічним завданням, а й обов’язковою нормативною вимогою. Недотримання

цих вимог може призвести до втрати сертифікації, значних фінансових санкцій та неможливості доведення фактів під час розслідування інцидентів.

Для підприємств, які працюють з платіжними даними, критично важливим є стандарт PCI DSS. Вимога 10 «Track and monitor all access to network resources and cardholder data» зобов'язує організації зберігати журнали подій безпеки щонайменше протягом одного року, причому останні три місяці мають бути доступними для негайного аналізу. PCI DSS прямо вимагає захисту журналів від несанкціонованих змін і наявності механізмів, що дозволяють виявити будь-які спроби їх модифікації.

У Європейському Союзі GDPR також встановлює жорсткі вимоги. Стаття 5 визначає принцип «цілісності та конфіденційності», а стаття 30 зобов'язує вести записи про обробку персональних даних. Стаття 32 вимагає впровадження технічних заходів, що забезпечують цілісність даних. У разі розслідування порушення захисту даних відсутність достовірних журналів може призвести до значних штрафів.

В Україні нормативна база також чітко регулює це питання. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» та Закон «Про основні засади забезпечення кібербезпеки України» встановлюють обов'язок суб'єктів критичної інформаційної інфраструктури забезпечувати захист журналів подій. Для банківських установ діють нормативні документи Національного банку України, зокрема Положення про організацію захисту інформації в банках, яке вимагає ведення захищених аудиторських журналів з обов'язковим контролем їхньої цілісності. Державна служба спеціального зв'язку та захисту інформації України у своїх рекомендаціях також підкреслює необхідність застосування криптографічних методів захисту аудиторських слідів.

Таким чином, забезпечення цілісності журналів подій є не рекомендацією, а обов'язковою вимогою низки міжнародних і національних

нормативних документів. Недотримання цих вимог може призвести до втрати сертифікації, значних фінансових санкцій, неможливості доведення фактів у суді та зниження довіри клієнтів і партнерів. Саме тому традиційні централізовані системи журналювання, які не забезпечують криптографічно підтвердженої незмінності даних, стають все менш прийнятними для сучасних підприємств, що працюють у регульованих галузях.

РОЗДІЛ 2 УРАЗЛИВОСТІ ТА ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ У ТРАДИЦІЙНИХ СИСТЕМАХ ЖУРНАЛЮВАННЯ

2.1 Основні недоліки централізованих систем журналювання

Централізовані системи журналювання подій, які сьогодні домінують на більшості підприємств, мають ряд фундаментальних недоліків, що суттєво знижують їхню ефективність у сфері інформаційної безпеки. Головним недоліком є саме централізований принцип зберігання даних. Усі логи з серверів, комутаторів, firewall, робочих станцій та прикладних систем збираються на один або кілька центральних серверів, або кластер SIEM. Це створює так звану єдину точку компрометації.

Якщо зловмисник отримує доступ до цього центрального сервера, особливо з правами адміністратора, він автоматично отримує контроль над усією історією мережевих подій підприємства. У такому випадку він може не лише переглядати логи, але й видаляти, модифікувати або повністю очищати їх, приховуючи таким чином сліди своєї діяльності. У середовищі Windows Server це особливо небезпечно, оскільки з правами Domain Admin або Local Administrator зловмисник може використовувати вбудовані системні інструменти: wevtutil.exe, PowerShell-команди Clear-EventLog, WMI для швидкого і тихого видалення записів з журналів Security, System та Application.

Другий серйозний недолік відсутність криптографічної гарантії цілісності даних. Після того, як подія записана в базу даних SIEM або у файл, її можна змінити або видалити без будь-яких слідів. Традиційні реляційні бази даних: SQL Server, PostgreSQL, Elasticsearch – не забезпечують механізмів, які б унеможливлювали несанкціоновану модифікацію. Навіть якщо SIEM-система має функції аудиту, вони також зберігаються в тій самій централізованій базі і можуть бути підмінені.

Третій недолік вразливість до атак з підвищенням привілеїв та латерального руху. Сучасні APT-групи рідко починають атаку з масового викрадення даних. Вони спочатку намагаються закріпитися в мережі, підвищити привілеї до рівня адміністратора і лише потім працюють з журналами, щоб максимально приховати свою присутність. У результаті розслідування інцидентів стає надзвичайно складним або навіть неможливим через відсутність достовірної доказової бази.

Четвертий недолік обмежений термін зберігання логів. Багато компаній зберігають журнали лише 30–90 днів через високі витрати на зберігання. Це дозволяє досвідченим зловмисникам просто «перечекати» цей період, після чого сліди атаки зникають назавжди.

Крім того, централізовані системи мають проблеми з масштабованістю. При великій кількості джерел подій, наприклад при використанні тисячі пристроїв, навантаження на центральний сервер зростає експоненційно, що призводить до затримок в обробці подій у реальному часі та можливих втрат даних.

Таким чином, централізовані системи журналювання, незважаючи на свої переваги в зручності моніторингу та аналізу, демонструють фундаментальну слабкість у забезпеченні цілісності та незмінності даних. Ця слабкість особливо небезпечна в умовах зростання кількості цілеспрямованих атак з боку професійних хакерських груп. Саме тому виникає нагальна потреба у нових технологічних рішеннях, які могли б забезпечити криптографічну гарантію незмінності журналів навіть у разі повної компрометації центрального сервера.

2.2 Аналіз реальних кіберінцидентів

Реальні кіберінциденти останніх років яскраво демонструють критичні недоліки централізованих систем журналювання. Зловмисники все частіше використовують один і той самий підхід: після успішного проникнення вони

спрямовують основні зусилля саме на приховування своєї присутності через маніпуляцію з журналами подій.

Одним з найбільш показових прикладів є атака на SolarWinds Orion у 2020 році, відома як Solorigate або SUNBURST. Зловмисники групи APT29 (UNC2452) здійснили supply-chain атаку, вбудувавши backdoor у легітимні оновлення програмного забезпечення Orion Platform. Після встановлення оновлення тисячами клієнтів, включаючи державні установи США, зловмисники отримали тривалий прихований доступ до мереж жертв. Важливою частиною їхньої стратегії стало активне приховування слідів. Вони проводили розвідку, латеральний рух мережею та підвищення привілеїв, одночасно працюючи з журналами подій на скомпрометованих системах. За даними Microsoft та Mandiant, атакуючі використовували вбудовані інструменти Windows для очищення Windows Event Logs, що значно ускладнило виявлення та повне відновлення хронології атаки.

Ще одним яскравим прикладом стала атака китайської хакерської групи Storm-0558 у травні–червні 2023 року на електронну пошту Державного департаменту США. Зловмисники не ламали паролі в класичному розумінні, а використали підроблені токени аутентифікації Microsoft Azure Active Directory. Це дозволило їм видавати себе за легітимних користувачів і отримувати доступ до поштових скриньок високопосадовців. Після проникнення вони мали повний доступ до централізованих журналів Exchange Online. Логи авторизації, перегляду поштових скриньок та завантаження файлів зберігалися в єдиній системі, тому зловмисники могли видаляти або модифікувати записи про свою активність. Атака була виявлена лише через кілька тижнів, що свідчить про ефективність їхніх дій з приховування слідів.

Подібну тактику було зафіксовано під час атаки на Colonial Pipeline у 2021 році та при масовій експлуатації вразливості MOVEit Transfer у 2023 році. У всіх цих випадках зловмисники після початкового закріплення в мережі свідомо працювали з журналами, намагаючись видалити або спотворити записи про свою діяльність. Це дозволяє зробити висновок, що сучасні APT-

групи добре розуміють слабкі місця традиційних систем журналювання і розглядають централізовані логи як одну з найпривабливіших цілей.

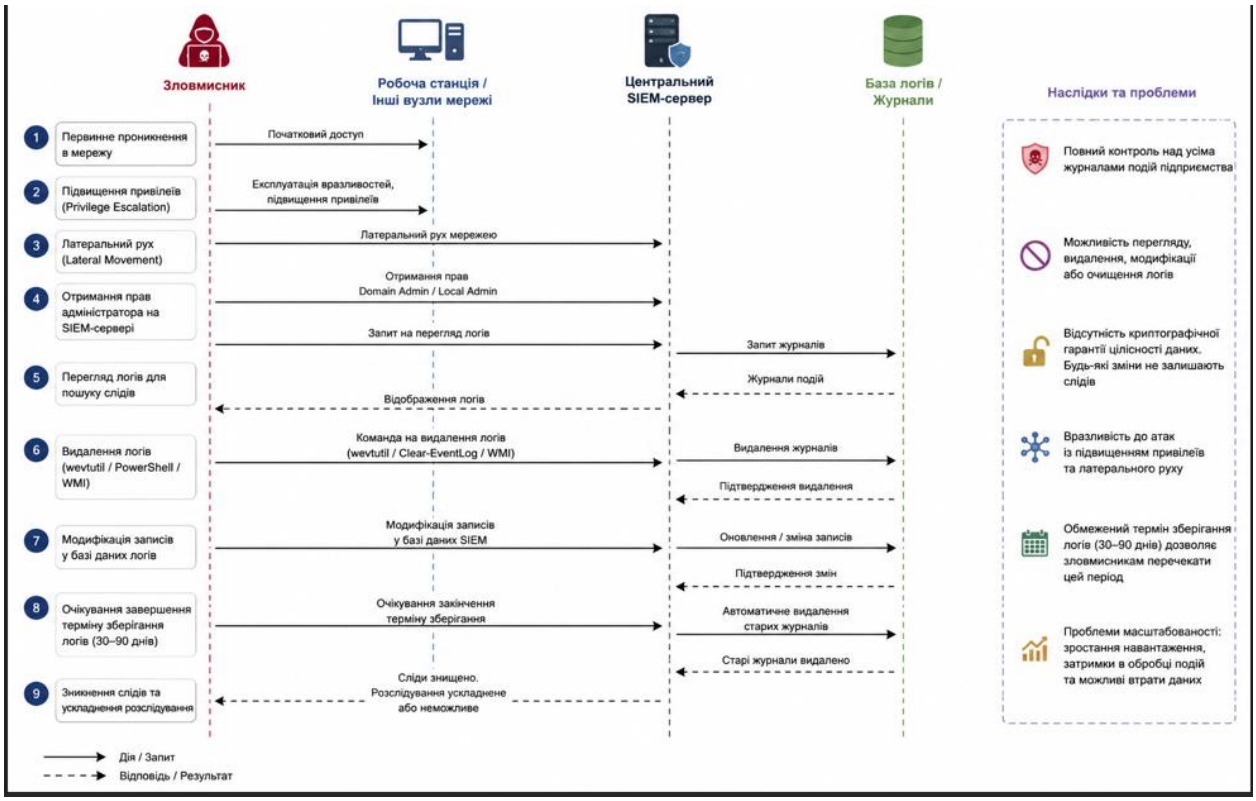

Рисунок 2.1 – Послідовність дій зловмисника при компрометації централізованої системи журналювання (SIEM)

Аналіз цих інцидентів показує кілька спільних закономірностей: Зловмисники рідко починають з масового викрадення даних. Спочатку вони намагаються закріпитися, провести розвідку і підвищити привілеї.

Одним з головних пріоритетів стає робота саме з журналами подій. Для маніпуляції логами часто використовуються вбудовані інструменти операційної системи, що ускладнює їх виявлення засобами захисту.

Після успішної атаки централізовані журнали часто втрачають свою доказову цінність, що робить повноцінне розслідування надзвичайно складним.

Таким чином, реальні кіберінциденти підтверджують, що традиційні централізовані системи журналювання, незважаючи на свої функціональні

переваги, демонструють фундаментальну вразливість у забезпеченні цілісності даних. Це обумовлює необхідність впровадження нових технологічних рішень, які можуть гарантувати незмінність журналів навіть у разі повної компрометації серверів.

2.3 Реалізація системи захисту на основі блокчейну

Для вирішення проблеми відсутності надійного захисту цілісності даних у традиційних системах журналювання, в рамках кваліфікаційної роботи була розроблена система на основі технології приватного блокчейну.

Головна ідея полягає не в повній заміні існуючих SIEM-систем, а в створенні додаткового незалежного захисного шару. Кожна подія, що виникає в корпоративній мережі, фіксується не тільки в центральній базі даних SIEM, а й записується в незмінний ланцюжок блоків. Таким чином, навіть якщо зловмисник отримає повний доступ до SIEM-сервера і спробує видалити або модифікувати логи, у системі залишиться криптографічно захищена копія подій, яку неможливо непомітно підробити.

Система має модульну архітектуру і складається з чотирьох основних компонентів. Перший модуль це збір і нормалізація подій. Він збирає інформацію з різноманітних джерел: Windows Event Logs, syslog з мережевого обладнання Cisco, NetFlow/IPFIX-потоків, а також журналів прикладних систем. Другий модуль, формування транзакцій. Тут кожна подія перетворюється в стандартизовану структуру: до неї додається точна тимчасова мітка, хеш попередньої транзакції та цифровий підпис для підтвердження автентичності. Третій модуль, блокчейн-реєстр, де створюється новий блок і додається до загального ланцюжка. Четвертий модуль це перевірка цілісності, який дозволяє в будь-який момент виконати незалежну перевірку всього ланцюжка блоків.

Система побудована на принципах permissioned blockchain, тобто приватного блокчейну. На відміну від публічних блокчейнів, таких як Bitcoin чи Ethereum, дана мережа є закритою і доступ до неї мають лише авторизовані

вузли всередині організації. Це дозволяє значно підвищити швидкість запису подій, зберігаючи при цьому високий рівень криптографічного захисту. Для обчислення хешів використовується алгоритм SHA-256, а автентичність кожної транзакції підтверджується цифровим підписом на основі асиметричної криптографії.

Завдяки механізму хеш-ланцюжка будь-яка, навіть найменша зміна даних в одному з попередніх блоків призводить до невідповідності всіх наступних хешів. Це дає можливість автоматично виявляти будь-які спроби фальсифікації або видалення журналів подій. Таким чином, система забезпечує математично доведену незмінність даних, чого неможливо досягти за допомогою традиційних реляційних баз даних або централізованих SIEM-рішень.

2.4. Переваги та особливості запропонованого рішення

Традиційні системи журналювання подій та сучасні SIEM-платформи, такі як Microsoft Sentinel, Splunk Enterprise, ELK Stack, IBM QRadar та ArcSight, вже протягом багатьох років залишаються основним інструментом моніторингу безпеки в корпоративному середовищі. Вони добре справляються із збором, кореляцією та візуалізацією подій у реальному часі. Однак при більш глибокому аналізі стає очевидним, що ці рішення мають системні недоліки, які суттєво обмежують їхню ефективність у забезпеченні справжньої цілісності даних.

Найсерйознішою проблемою є централізований характер зберігання інформації. Усі журнали, що надходять з серверів, комутаторів Cisco, робочих станцій та прикладних систем, зосереджуються в одному або кількох центральних сховищах. Така архітектура робить SIEM-сервер або базу даних єдиною точкою компрометації. Якщо зловмисник, як це часто відбувається під час APT-атак, отримує права адміністратора на цьому сервері, він отримує можливість не лише переглядати логи, але й видаляти, модифікувати або повністю очищати записи про свою діяльність. Реальні приклади атак на

SolarWinds, Colonial Pipeline та державні установи США чітко продемонстрували, наскільки ефективно зловмисники використовують цю вразливість.

Іншою важливою проблемою є відсутність надійних криптографічних механізмів захисту цілісності даних після їх запису. Навіть у таких потужних рішеннях, як Splunk з функцією Immutable Indexes або Microsoft Sentinel з Immutable Logs, захист працює лише до певної межі і часто залежить від правильної конфігурації та рівня привілеїв адміністратора. Якщо зловмисник отримує достатні права, він може обійти ці механізми або вимкнути їх. Крім того, багато компаній продовжують використовувати звичайні реляційні бази даних або Elasticsearch, де будь-який запис можна змінити без залишення криптографічних слідів.

Ще одним суттєвим обмеженням є залежність від вендора та хмарної інфраструктури. Багато сучасних SIEM-рішень (особливо Microsoft Sentinel та Splunk Cloud) активно просувають хмарне розгортання, що не завжди прийнятно для підприємств критичної інфраструктури, банків та державних організацій через вимоги до суверенітету даних та локального зберігання інформації. При цьому готові комерційні рішення на основі блокчейну для захисту логів на ринку практично відсутні або є надто дорогими та складними в інтеграції.

Саме тому в рамках цієї кваліфікаційної роботи було обрано розробку системи на основі технології приватного (permissioned) блокчейну. На відміну від традиційних SIEM-систем, запропоноване рішення створює додатковий незалежний захисний шар, у якому кожна подія фіксується в криптографічно пов’язаному ланцюжку блоків. Завдяки механізму хеш-ланцюжка та використанню алгоритму SHA-256 будь-яка спроба зміни або видалення вже записаної події призводить до порушення цілісності всього ланцюжка, що відразу виявляється системою.

Такий підхід не вимагає повної заміни існуючих SIEM-платформ, а працює паралельно з ними, забезпечуючи математично підтверджену

незмінність даних навіть у разі повної компрометації центрального сервера. Це дозволяє вирішити проблему, яку традиційні системи принципово не здатні вирішити, а саме гарантію цілісності журналів після їх запису. Крім того, приватний блокчейн дає можливість повністю контролювати інфраструктуру всередині організації, не залежачи від зовнішніх вендорів та хмарних провайдерів.

Таким чином, розроблена система не є альтернативою SIEM-рішенням, а їхнім важливим доповненням, яке закриває критичну прогалину в забезпеченні довгострокової цілісності та доказовості мережевих журналів подій.

РОЗДІЛ 3 ПРОТОТИП СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ МЕРЕЖЕВИХ ЖУРНАЛІВ НА ОСНОВІ БЛОКЧЕЙНУ

Для практичної перевірки запропонованого підходу в рамках кваліфікаційної роботи розроблено та протестовано програмний прототип системи захисту мережевих журналів подій на основі технології приватного блокчейну. Прототип реалізований мовою програмування Python.

3.1 Архітектура розробленого прототипу

Для практичної апробації запропонованого підходу в межах кваліфікаційної роботи був розроблений програмний прототип системи забезпечення цілісності мережевих журналів подій на основі технології приватного блокчейну. Прототип реалізований мовою програмування Python і призначений для демонстрації ключових механізмів створення незмінного ланцюжка даних. Повний код прототипу представлено в Додатку А.

Архітектура прототипу має чітку модульну структуру і складається з двох основних класів: Block та Blockchain (див. рис. 3.1 та Додаток А).

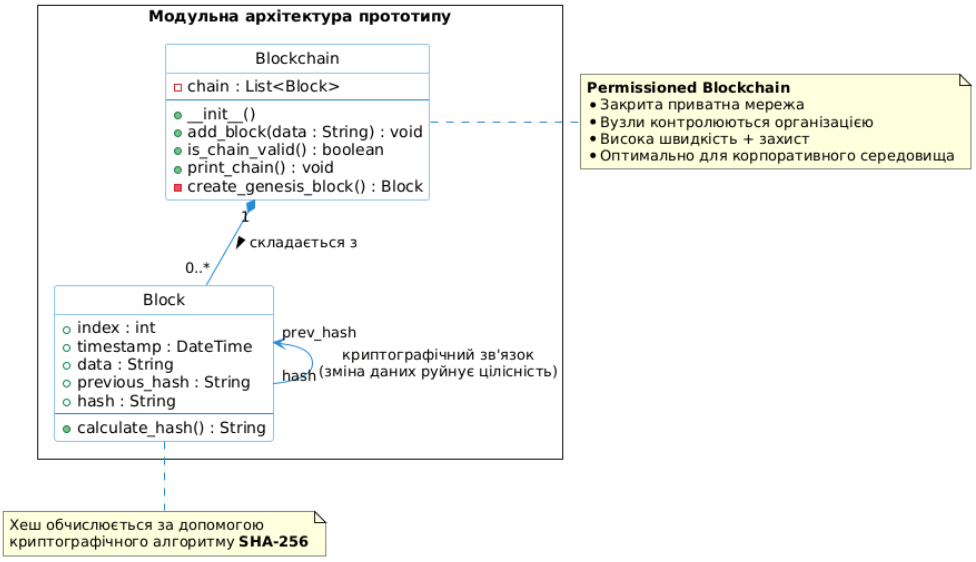

Рис.3.1 – Принцип роботи прототипу на основі технології приватного блокчейну

Клас Block є базовою одиницею ланцюжка і відповідає за структуру окремого блоку. Він містить такі атрибути:

- index - порядковий номер блоку в ланцюжку;
- timestamp - точна тимчасова мітка створення блоку;
- data - корисне навантаження (нормалізовані дані мережевої події);
- previous_hash - криптографічний хеш попереднього блоку;
- hash - власний хеш блоку, обчислений за допомогою алгоритму SHA-256.

Метод calculate_hash() забезпечує генерацію унікального хешу на основі всіх полів блоку. Це фундаментальний механізм, який робить неможливою непомітну модифікацію даних.

Клас Blockchain управляє всім ланцюжком блоків. При ініціалізації він автоматично створює genesis-блок (початковий блок), який не має попереднього хешу. Основні методи класу включають:

- add_block(data) - додає нову подію до ланцюжка, створюючи новий блок;
- is_chain_valid() - перевіряє цілісність усього ланцюжка шляхом послідовного порівняння хешів;
- print_chain() - виводить детальну інформацію про всі блоки для аналізу.

Прототип реалізований за принципом permissioned blockchain. На відміну від публічних блокчейнів, дана система є закритою, і доступ до неї мають лише авторизовані компоненти всередині організації. Такий підхід дозволяє значно підвищити швидкість запису подій, зберігаючи при цьому високий рівень криптографічного захисту.

Найважливішою особливістю архітектури є механізм хеш-ланцюжка. Кожен новий блок обов’язково містить хеш попереднього блоку. Завдяки цьому будь-яка, навіть незначна зміна даних в одному з попередніх блоків призводить до невідповідності хешів у всіх наступних блоках, що одразу

виявляється системою під час перевірки цілісності. Такий підхід забезпечує математично доведену незмінність (immutability) даних.

Таким чином, розроблена архітектура прототипу поєднує в собі відносну простоту реалізації з ключовими перевагами технології блокчейн: криптографічним захистом, децентралізацією довіри та можливістю незалежної перевірки цілісності даних. Вона може ефективно використовуватися як додатковий захисний шар до існуючих SIEM-систем, суттєво підвищуючи надійність зберігання мережевих журналів подій.

3.2 Алгоритм роботи прототипу

Алгоритм роботи розробленого прототипу побудований на послідовному виконанні чітко визначених етапів, що забезпечують надійну фіксацію та криптографічний захист кожної мережевої події. Логіка роботи системи наочно представлена на рисунку 3.2.

Процес починається з надходження події з різних джерел (Windows Event Log, Syslog, NetFlow та інших). Модуль збору подій отримує сирі дані, проводить їх первинну обробку, очищення та нормалізацію, приводячи інформацію до єдиного внутрішнього формату. Це дозволяє системі працювати з подіями незалежно від їхнього походження та структури.

Після нормалізації подія передається до модуля формування транзакції. На цьому етапі створюється структурована транзакція, до якої додається точна тимчасова мітка, основні дані події, хеш попередньої транзакції, а також виконується цифровий підпис для підтвердження автентичності.

Готова підписана транзакція надходить до блокчейн-модуля, де відбувається створення нового блоку. Кожен блок містить порядковий номер, тимчасову мітку, дані транзакції, хеш попереднього блоку та власний хеш, обчислений за допомогою алгоритму SHA-256. Після формування блок додається до ланцюжка, стаючи його невід’ємною частиною.

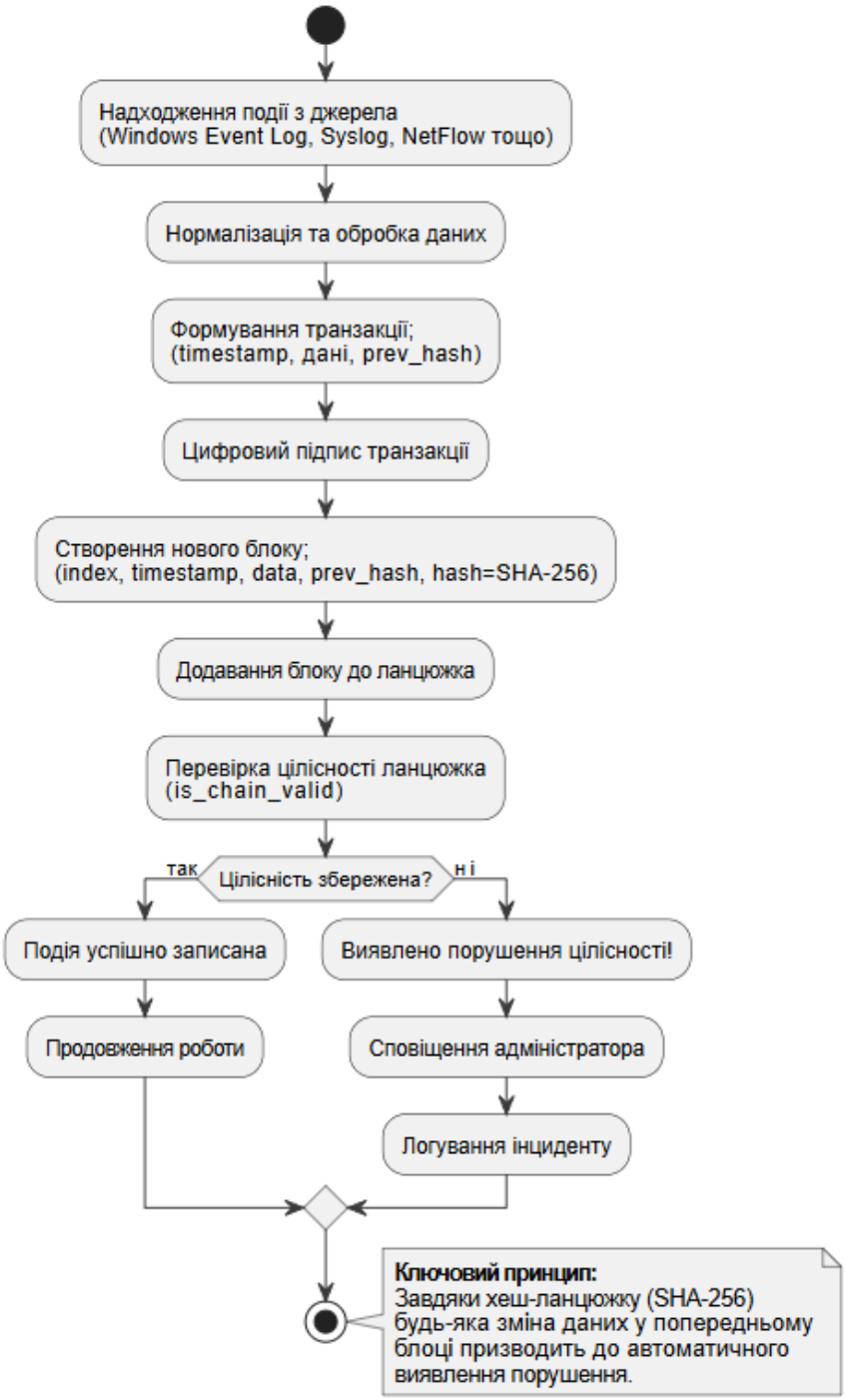

Рис 3.2 – Алгоритм роботи системи захисту журналів подій на основі технології приватного блокчейну

Завершальним етапом є перевірка цілісності ланцюжка. Спеціальний модуль послідовно проходить по всіх блоках, починаючи з genesis-блоку, перераховує хеші та порівнює їх зі збереженими значеннями. Якщо цілісність

збережена, подія вважається успішно записаною і система продовжує роботу. У разі виявлення будь-якої невідповідності (що свідчить про спробу несанкціонованої зміни даних) система фіксує порушення цілісності та надсилає сповіщення адміністратору.

Такий алгоритм забезпечує головну перевагу розробленої системи: після запису події в блокчейн її неможливо змінити або видалити без порушення цілісності всього ланцюжка. Це принципово відрізняє запропоноване рішення від традиційних централізованих систем журналювання, де дані залишаються вразливими до модифікації навіть після їх запису.

3.3 Результати тестування прототипу

Для практичної перевірки працездатності, надійності та стійкості до атак розробленого прототипу було проведено комплексне поетапне тестування. Тестування здійснювалося на персональному комп’ютері в середовищі Windows 10 з використанням Python 3.13. Основними цілями були: перевірка коректності запису подій, забезпечення криптографічного зв’язку між блоками, виявлення спроб несанкціонованої модифікації даних та оцінка загальної стабільності системи.

Етап 1. Запуск прототипу та запис тестових мережевих подій

На першому етапі було виконано ініціалізацію блокчейн-ланцюжка зі створенням Genesis блоку. Далі до ланцюжка було додано чотири тестові мережеві події різного типу. Кожна подія успішно пройшла етапи нормалізації, формування транзакції та запису в новий блок. Система коректно створила блоки №1–№4 (див. рис. 3.1).

Рисунок 3.1 – Створення блоків подій

Етап 2. Перевірка цілісності ланцюжка

Після завершення запису подій була виконана автоматична перевірка цілісності всього ланцюжка за допомогою методу is_chain_valid(). Система послідовно перерахувала хеші всіх блоків і порівняла їх зі збереженими значеннями. Результат перевірки: Ланцюжок валідний: True (див. рис. 3.2). Це підтвердило правильність роботи механізму хеш-ланцюжка.

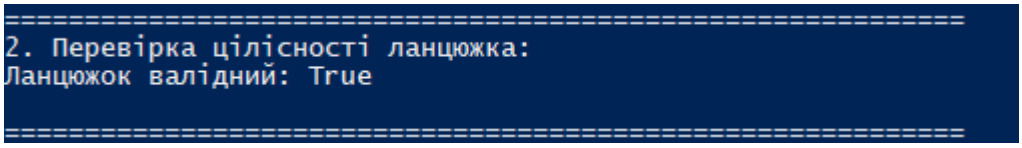

Рисунок 3.2 – Результат перевірки цілісності ланцюжка блоків після запису подій

Етап 3 Симуляція атаки (несанкціонована модифікація даних)

На третьому етапі була змодельована реальна загроза така як спроба несанкціонованої зміни даних в уже записаному блоці (блок №2). Після внесення змін (замість оригінального тексту події було вставлено повідомлення про атаку) система повторно перевірила цілісність ланцюжка. Результат перевірки: False. Порушення було виявлено відразу (див. рис. 3.3). Даний етап наочно демонструє ключову перевагу блокчейн-підходу: неможливість непомітної модифікації даних.

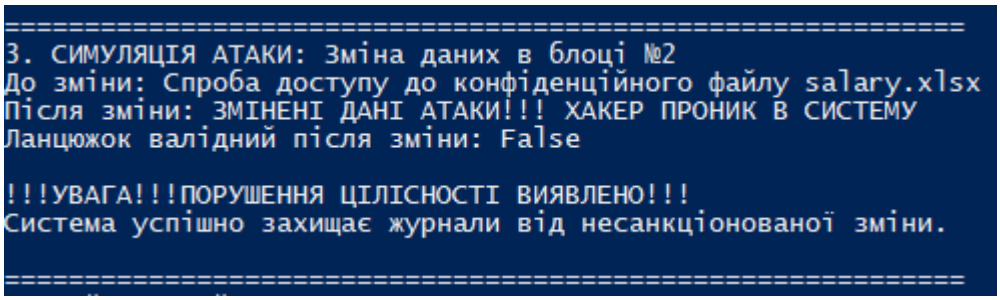

Рисунок.3.3 – Результат симуляції атаки виявлення порушення цілісності ланцюжка

Етап 4. Повний вивід блокчейн-ланцюжка

Для детального аналізу було виведено весь ланцюжок блоків з повною інформацією: номером блоку, часом створення, вмістом події та хешем (див. рис. 3.4).

Це дозволило візуально оцінити структуру даних і переконатися в коректній роботі механізму хеш-ланцюжка.

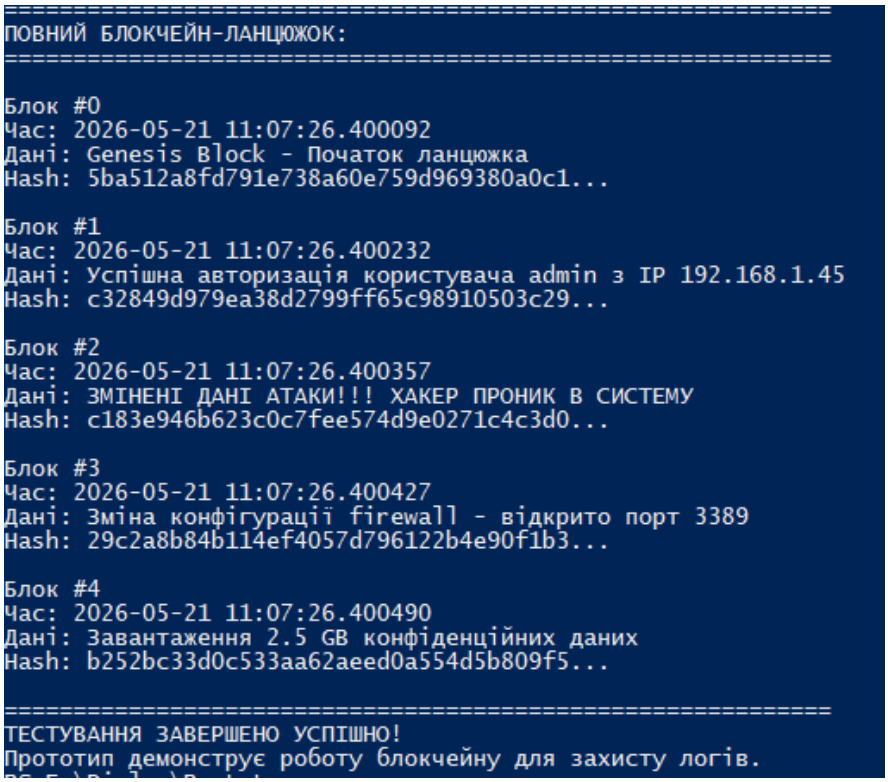

Рисунок.3.4 – Повний вивід блокчейн-ланцюжка після завершення тестування

Загальні результати тестування підтверджують, що розроблений прототип успішно виконує поставлені завдання. Він забезпечує стабільний запис подій, підтримує криптографічний зв’язок між блоками та ефективно виявляє будь-які спроби несанкціонованої модифікації даних. Отримані дані свідчать про працездатність запропонованого рішення та його потенціал для практичного застосування як додаткового захисного шару в корпоративних системах забезпечення цілісності мережевих журналів подій.

ВИСНОВКИ

У рамках кваліфікаційної роботи було проведено комплексне дослідження проблеми забезпечення цілісності мережевих журналів подій у корпоративних мережах підприємств. Сучасні корпоративні мережі, побудовані за трирівневою ієрархічною моделлю Cisco, постійно генерують величезні обсяги подій на всіх рівнях, від рівня доступу до магістрального. Для збору, зберігання та аналізу цих подій підприємства традиційно використовують централізовані SIEM-системи. Однак проведений теоретичний аналіз та вивчення реальних кіберінцидентів показали, що саме централізований принцип зберігання даних є їхнім головним недоліком. Сервер журналювання перетворюється на єдину точку компрометації, що дозволяє зловмисникам після успішного проникнення видаляти або модифікувати логи, повністю приховуючи сліди своєї діяльності.

Аналіз таких масштабних атак, як компрометація SolarWinds Orion у 2020 році, інцидент зі Storm-0558 у 2023 році та інші, чітко продемонстрував, що сучасні APT-групи добре розуміють цю слабкість і свідомо використовують її для приховування своєї присутності в системі. Традиційні SIEM-рішення – Microsoft Sentinel, Splunk, ELK Stack та інші, незважаючи на потужні можливості моніторингу та аналізу в реальному часі, не забезпечують криптографічної гарантії незмінності даних. Після компрометації центрального сервера вся історія подій втрачає свою доказову цінність, що значно ускладнює розслідування інцидентів та виконання нормативних вимог.

Для вирішення цієї критичної проблеми в роботі була запропонована та реалізована система забезпечення цілісності мережевих журналів подій на основі технології приватного блокчейну. У результаті виконання поставлених завдань створено програмний прототип, який демонструє ключові принципи роботи блокчейну: формування криптографічно пов’язаних блоків, використання хеш-ланцюжка, цифрові підписи та механізм перевірки цілісності. Прототип успішно пройшов комплексне тестування, під час якого

було підтверджено його здатність фіксувати події, підтримувати незмінність даних та виявляти будь-які спроби несанкціонованої модифікації.

Отримані результати свідчать, що використання технології блокчейн дозволяє суттєво підвищити рівень захисту мережевих журналів. Навіть у разі повної компрометації окремого сервера або вузла зловмисник не зможе непомітно змінити історію подій. Це дає підприємствам можливість мати достовірну, перевірену доказову базу для розслідування інцидентів, проведення внутрішніх та зовнішніх аудитів, а також виконання вимог регуляторів.

Запропоноване рішення має високий практичний потенціал. Воно може бути застосоване в банківській сфері, державних установах, компаніях критичної інфраструктури, медичних закладах та великих промислових підприємствах, скрізь, де потрібна гарантована незмінність журналів подій. Перспективи подальшого розвитку включають інтеграцію прототипу з існуючими SIEM-системами, підвищення продуктивності, створення розподіленої версії з кількома вузлами та розробку повноцінного програмного продукту, готового до промислового впровадження.

Таким чином, у ході виконання кваліфікаційної роботи теоретичні знання про архітектуру корпоративних мереж, системи журналювання та технологію блокчейн були успішно поєднані з практичною реалізацією. Розроблена система демонструє реальну можливість вирішення однієї з найактуальніших проблем сучасної інформаційної безпеки: забезпечення криптографічної цілісності мережевих журналів подій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Enterprise Campus 3.0 Architecture: Overview and Framework /Cisco Systems. – URL: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/campover.html (дата звернення: 12.05.2026).

2. MITRE ATT&CK. T1070.001 – Clear Windows Event Logs – URL: https://attack.mitre.org/techniques/T1070/001/ (дата звернення: 15.05.2026).

3. Китайські хакери зламали електронну пошту Держдепартаменту США //Суспільне Новини. – 2023. – URL: https://suspilne.media/527593-kitajski-hakeri-zlamali-elektronnu-postu-derzdepartamentu-ssa/ (дата звернення: 10.05.2026).

4. Основи та приклади застосування технології блокчейн: методичні матеріали / Харківський політехнічний інститут. – Харків, 2024. – 68 с. – URL: https://cybersecurity.khpi.edu.ua/wp-content/uploads/2024/09/ВП-2.3_256_Blockchain_основи-та-приклади-застосування.pdf (дата звернення: 08.05.2026).

5. Drescher D. Blockchain Basics: A Non-Technical Introduction in 25 Steps / Daniel Drescher. – New York: Apress, 2017. – 255 p.

6. SearchInform. Використання блокчейну в SIEM-системах для забезпечення цілісності даних – URL: https://searchinform.com/articles/cybersecurity/measures/siem/use-cases/blockchain/ (дата звернення: 11.05.2026).

7. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection – Information security management systems – Requirements. – Geneva: ISO, 2022. – URL: https://www.iso.org/standard/27001 (дата звернення: 05.05.2026).

8. Payment Card Industry Data Security Standard (PCI DSS) Version 4.0. – PCI Security Standards Council, 2022.

9. Regulation (EU) 2016/679 (General Data Protection Regulation) – URL: https://gdpr-info.eu/ (дата звернення: 12.05.2026).

10. Закон України «Про основні засади забезпечення кібербезпеки України» від 05.10.2017 № 2163-VIII. – URL: https://zakon.rada.gov.ua/laws/show/2163-19 (дата звернення: 10.05.2026).

11. Положення про організацію захисту інформації в банках України / Національний банк України. К., 2021.

12. Visual Studio Code Documentation URL: https://code.visualstudio.com/docs (дата звернення: 01.03.2026).

13. Python 3.12 Documentation URL: https://docs.python.org/3/ (дата звернення: 20.04.2026).

ДОДАТКИ

Додаток А – Лістинг прототипу

BlockChain.py

C: > Users > ASUS > Desktop > Дипломная работа > Прототип > BlockChain.py > Blockchain > is_chain_valid

```python
import hashlib
import datetime
import json

class Block:
    def __init__(self, index, timestamp, data, previous_hash):
        self.index = index
        self.timestamp = timestamp
        self.data = data                    # дані події (лог)
        self.previous_hash = previous_hash
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        block_string = json.dumps({
            "index": self.index,
            "timestamp": str(self.timestamp),
            "data": self.data,
            "previous_hash": self.previous_hash
        }, sort_keys=True)
        return hashlib.sha256(block_string.encode()).hexdigest()


class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, datetime.datetime.now(), "Genesis Block - Початок ланцюжка", "0")

    def add_block(self, data):
        previous_block = self.chain[-1]
        new_block = Block(
            index=len(self.chain),
            timestamp=datetime.datetime.now(),
            data=data,
            previous_hash=previous_block.hash
        )
        self.chain.append(new_block)
        print(f"Подія записана в блок #{new_block.index}")
        return new_block

```

BlockChain.py

C: > Users > ASUS > Desktop > Дипломная работа > Прототип > BlockChain.py > Blockchain > is_chain_valid

```
23  class Blockchain:
30      def add_block(self, data):
40          return new_block
41
42      def is_chain_valid(self):
43          for i in range(1, len(self.chain)):
44              current = self.chain[i]
45              previous = self.chain[i-1]
46
47              if current.hash != current.calculate_hash():
48                  return False
49              if current.previous_hash != previous.hash:
50                  return False
51          return True
52
53      def print_chain(self):
54          for block in self.chain:
55              print(f"\nБлок #{block.index}")
56              print(f"Час: {block.timestamp}")
57              print(f"Дані: {block.data}")
58              print(f"Hash: {block.hash[:20]}...")
59
60
61  # ====================== ТЕСТ ======================
62  if __name__ == "__main__":
63      bc = Blockchain()
64
65      print("Запуск прототипу Blockchain Logger\n")
66
67      # Додаємо тестові події
68      bc.add_block("Успішна авторизація користувача admin з IP 192.168.1.100")
69      bc.add_block("Спроба доступу до конфіденційного файлу salary.xlsx")
70      bc.add_block("Зміна конфігурації firewall - відкрито порт 3389")
71      bc.add_block("Завантаження великого обсягу даних з сервера")
72
73      print("\n" + "="*50)
74      print("Перевірка цілісності ланцюжка...")
75      print("Ланцюжок валідний:", bc.is_chain_valid())
76
77      # Спроба підміни
78      print("\nСпроба зміни даних в блоці #1...")
79      bc.chain[1].data = "ЗМІНЕНІ ДАНІ АТАКИ!!!"
80      print("Ланцюжок валідний після зміни:", bc.is_chain_valid())
```

Рисунок А.1 Основний код прототипу (BlockChain.py)