Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва інституту)

Кафедра комп'ютерних та інформаційних технологій і систем
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи

магістра

(ступінь вищої освіти)

на тему «Використання технологій BLOCKCHAIN в цифрових системах автоматизованого керування бізнес-процесами»

Виконав: студент II курсу, групи 602-ТН
спеціальності
122 Комп'ютерні науки
(шифр і назва спеціальності)

Рудь Артем Олександрович
(прізвище та ініціали)

Керівник к.т.н., доцент Чередников В.М.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Полтава – 2025 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки» на тему

«Використання технологій BLOCKCHAIN в цифрових системах автоматизованого керування бізнес-процесами»

Студент групи 602-ТН Рудь Артем Олександрович

Керівник роботи
к.т.н., доцент
Чередніков В. М.

Консультант
к.т.н., доцент Головко Г.В.

Завідувач кафедри
кандидат фізико-математичних наук,
Двірна О.А.

Полтава – 2025 рік

РЕФЕРАТ

Магістерська кваліфікаційна робота Рудь Артема Олександровича на тему «Використання технологій BLOCKCHAIN в цифрових системах автоматизованого керування бізнес-процесами» на здобуття ступеня вищої освіти магістра зі спеціальності 122 «Комп’ютерні науки».

В роботі вирішується актуальна наукова задача розробки інформаційно-технологічних моделей та методів захисту комп'ютерних систем і мереж з урахуванням специфіки діяльності малого та середнього бізнесу на основі методів зберігання та розповсюдження інформації, що базуються на технології блокчейн. Також важливим є завдання визначення архітектури інформаційної системи виявлення вторгнень для мереж МСБ з використанням блокчейн-компонентів. Об'єктом дослідження є інформаційні процеси в системах захисту від кіберзагроз та аномального трафіку в комп'ютерних мережах. Об'єктом дослідження є методи, моделі та елементи інформаційної технології спільного захисту від кібератак та аномального трафіку в комп'ютерних мережах підприємств малого та середнього бізнесу на основі технології блокчейн.

Метою дисертаційної роботи є підвищення ефективності захисту комп'ютерних мереж малого та середнього бізнесу на основі технології блокчейн. Завданням дослідження є побудова моделі розподіленої системи захисту комп'ютерних мереж на основі технології блокчейн на основі результатів аналізу основних загроз для комп'ютерних мереж, особливо мереж МСБ. Методологія дослідження базується на імітаційному моделюванні, UML-проектуванні компонентів технології блокчейн та методах математичного моделювання для визначення оптимальних параметрів підсистеми блокчейн. При побудові імітаційної моделі використовувалися методи експертних оцінок для точного вибору типових атак і корисного навантаження на систему, що атакується. Об'єктно-орієнтований аналіз та функціональне моделювання, зокрема методи проектування SADT, були використані як основа для концептуалізації бізнес-процесів у нотації IDEF0 та проектування інформаційної

технології для виявлення та аналізу аномальних подій з метою захисту комп'ютерних мереж МСП на основі блокчейнів.

Основним результатом та науковою новизною роботи є розробка моделі та алгоритмічної методології захисту комп'ютерних мереж МСБ на основі блокчейну. На основі аналізу існуючих загроз для комп'ютерних мереж МСБ визначено найбільш ефективні методи та заходи захисту таких мереж з урахуванням їх функціональних та експлуатаційних особливостей. Запропоновано перелік основних класифікаторів інформаційних технологій захисту комп'ютерних мереж, які можуть бути об'єднані в комплексний класифікатор для підвищення точності виявлення аномальних подій, невідомих розподіленим системам виявлення вторгнень. Розроблені методи підвищення ефективності використання ресурсів підсистемами блокчейну дозволяють прогнозувати ймовірність успішного створення блоків та зменшити споживання ресурсів підсистемами блокчейну. Запропонована архітектура розподіленої системи захисту комп'ютерних мереж на основі блокчейну дозволяє надійно захищати мережі МСБ від вторгнень з використанням даних, зібраних з різних мереж великою кількістю вузлів розподіленої системи. Побудовано UML-діаграму основних компонентів розподіленої системи захисту комп'ютерних мереж на основі блокчейну, деталізовано зв'язки між компонентами системи та структури самих компонентів системи. Кросплатформна реалізація блокчейн-компонентів та класифікаторів забезпечує можливість тестування на широкому спектрі апаратних засобів, що підтримують Unix-системи, незалежно від апаратних та програмних платформ.

Ключові слова: блокчейн, інформаційні технології, комп’ютерні мережі та системи, безпека інформаційних систем, вторгнення, системні реєстри.

ABSTRACT

Master's qualification work of Rud Artem Oleksandrovych on the topic "Use of BLOCKCHAIN technologies in digital systems of automated business process management" for obtaining a higher education degree of master in specialty 122 "Computer Science".

The work solves the current scientific problem of developing information technology models and methods for protecting computer systems and networks, taking into account the specifics of small and medium-sized businesses based on methods of storing and distributing information based on blockchain technology. Also important is the task of determining the architecture of an information system for detecting intrusions for SME networks using blockchain components. The object of the study is information processes in systems for protecting against cyber threats and anomalous traffic in computer networks. The object of the study is methods, models and elements of information technology for joint protection against cyber attacks and anomalous traffic in computer networks of small and medium-sized businesses based on blockchain technology.

The purpose of the dissertation is to improve the effectiveness of protecting small and medium-sized business computer networks based on blockchain technology. The task of the study is to build a model of a distributed computer network protection system based on blockchain technology based on the results of the analysis of the main threats to computer networks, especially SME networks. The research methodology is based on simulation modeling, UML design of blockchain technology components and mathematical modeling methods to determine the optimal parameters of the blockchain subsystem. When building the simulation model, expert assessment methods were used to accurately select typical attacks and payloads on the attacked system. Object-oriented analysis and functional modeling, in particular SADT design methods, were used as the basis for conceptualizing business processes in IDEF0 notation and designing information technology for detecting and analyzing anomalous events in order to protect SME computer networks based on blockchains.

The main result and scientific novelty of the work is the development of a model and algorithmic methodology for protecting SME computer networks based on blockchain. Based on the analysis of existing threats to SME computer networks, the most effective methods and measures for protecting such networks have been determined, taking into account their functional and operational features. A list of the main classifiers of information technologies for protecting computer networks has been proposed, which can be combined into a comprehensive classifier to increase the accuracy of detecting anomalous events unknown to distributed intrusion detection systems. The developed methods for increasing the efficiency of resource use by blockchain subsystems allow predicting the probability of successful block creation and reducing resource consumption by blockchain subsystems. The proposed architecture of a distributed system for protecting computer networks based on blockchain allows reliably protecting SME networks from intrusions using data collected from different networks by a large number of nodes of the distributed system. A UML diagram of the main components of a distributed computer network protection system based on blockchain has been constructed, the connections between the system components and the structure of the system components themselves have been detailed. Cross-platform implementation of blockchain components and classifiers provides the possibility of testing on a wide range of hardware supporting Unix systems, regardless of hardware and software platforms.

Keywords: blockchain, information technology, computer networks and systems, information system security, intrusion, system registries.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AAA – Authentication, Authorization, Accounting (Аутентифікація, авторизація та облік)

ASIC – Application-Specific Integrated Circuit (Інтегрована схема, спеціа-лізована для додатку)

DDOS – Distributed Denial of Service (Розподілена відмова в обслуговуванні)

DMZ – Demilitarized Zone (Демілітаризована зона)

DNS – Domain Name System (Система доменних імен)

DOS – Denial of Service (Відмова в обслуговуванні)

FTP – File Transfer Protocol (Протокол передачі файлів)

HIDS – Host-based Intrusion Detection System (Система виявлення вторгнень на основі хосту)

ICMP – Internet Control Message Protocol (Протокол керування повідо-мленнями Інтернету)

IDS – Intrusion Detection System (Система виявлення вторгнень)

IP – Internet Protocol (Протокол Інтернету)

IPS – Intrusion Prevention System (Система запобігання вторгненням)

NIDS – Network-based Intrusion Detection System (Система виявлення вто-ргнень на основі мережі)

PoW – Proof of Work (Доказ роботи)

PoS – Proof of Stake (Доказ власності)

PBFT – Practical Byzantine Fault Tolerance (Практична Візантійська відмо-востійкість)

SOHO – Small Office/Home Office (Малий/домашній офіс)

TCP – Transmission Control Protocol (Протокол керування передачею)

SMB – Small/Medium Business (Мале/середнє підприємство)

SPoF – Single Point of Failure (Єдина точка відмови)

VoIP – Voice over Internet Protocol (Протокол інтернет-телефонії)

ВСТУП

Актуальність дослідження пов’язана зі стрімким розвитком технологій та зростанням кіберзагроз, які стають дедалі складнішими, організованішими й добре фінансованими. Сучасні кібератаки не лише спрямовуються на великі корпорації, але й активно загрожують невеликим комп’ютерним мережам малого та середнього бізнесу (МСБ), які є найбільш вразливими через обмежені можливості захисту. Такі мережі, відомі як мережі SOHO та SMB, значно поступаються великим корпоративним системам у надійності засобів безпеки, що робить їх головними цілями зловмисників.

Основну роль у захисті комп'ютерних мереж відіграють системи виявлення та запобігання вторгнень. Проте їхня ефективність залежить від архітектурних рішень і правильного розміщення у мережі. Дослідження існуючих методів кібербезпеки демонструє необхідність подальших кроків у створенні нових моделей і підходів до захисту та виявлення загроз. Особливого значення ця проблема набуває у реаліях гібридної війни, коли кіберзагрози впливають на всі аспекти життя, вимагаючи забезпечення надійного захисту незалежно від масштабу чи інтенсивності трафіку мережі.

Наразі існує гостра потреба у впровадженні інформаційних технологій, що ґрунтуються на блокчейн-рішеннях, для виявлення та аналізу аномальних подій у мережах малих і середніх підприємств. Сучасні підходи до кіберзахисту збагачуються інноваціями у сфері обробки великих масивів даних і безпечного обміну інформацією. Значний внесок у розвиток механізмів виявлення вторгнень зробили такі дослідники, як Керол Дж. Фунг, Ольга Байсал, Вінод Єгнесваран, Ставрос Шіалес та інші. У своїх роботах вони активно досліджують інтеграцію блокчейн-технологій для підвищення рівня кіберзахисту.

Попри суттєві успіхи, використання блокчейну в задачах виявлення вторгнень здебільшого зосереджено на мережах IoT, які мають інші особливості порівняно з мережами загального призначення або МСБ. Проблематика забезпечення безпеки для таких мереж потребує окремого підходу. Зазвичай малі комп’ютерні мережі орієнтовані на локальні методи виявлення загроз, однак

інтеграція їх у ширші розподілені системи відкриває можливості для впровадження більш складних, скоординованих методів кібербезпеки.

У цьому дослідженні пропонується інформаційна технологія для виявлення та аналізу аномалій із використанням блокчейн-рішень, орієнтована на потреби комп’ютерних мереж МСБ. Розподіленість компонентів такої системи дозволяє підвищити ефективність аналізу за рахунок використання даних з різних сегментів мережі, підданих різним видам атак. Це відкриває перспективи розробки нових інформаційно-технічних моделей захисту з акцентом на особливості мереж МСБ і впровадження блокчейн-технологій для збереження та поширення інформації.

Одним із ключових завдань є визначення оптимальної архітектури системи виявлення та реагування на вторгнення з інтеграцією блокчейн-компонентів. Реалізація подібного підходу сприятиме підвищенню захисту малих і середніх підприємств у сучасному цифровому середовищі, забезпечуючи стійкість до новітніх кіберзагроз.

Дана робота спрямована на підвищення ефективності захисту комп'ютерних мереж за допомогою технології блокчейн. У рамках дослідження поставлено та вирішено низку завдань:

– проведення аналізу основних загроз кібербезпеці комп'ютерних мереж, а також методів і засобів їх захисту;

– розробка моделі розподіленої інформаційної системи для виявлення та аналізу аномальних подій у мережах підприємств малого та середнього бізнесу на базі блокчейн-технологій;

– формування методичного підходу до вибору протоколу консенсусу для створеної моделі інформаційної системи;

– адаптація протоколу консенсусу PoS (Proof of Stake) до розподіленої системи виявлення вторгнень із метою підвищення її ефективності;

– оцінка перспектив подальшого вдосконалення запропонованих рішень.

Реалізація зазначених завдань дозволяє створити надійну систему кіберзахисту для оптимізації безпеки бізнес-процесів підприємства.

РОЗДІЛ 1
АНАЛІЗ КІБЕРЗАГРОЗ У КОМП'ЮТЕРНИХ МЕРЕЖАХ І СИСТЕМАХ ТА ОСНОВНИХ МЕТОДІВ ЗАХИСТУ

1.1 Види загроз для комп'ютерних мереж і систем

На перший погляд може здаватися, що головними цілями хакерських атак зазвичай є мережі великих корпорацій, тоді як мережі SOHO (Small Office/Home Office) та підприємств малого і середнього бізнесу (МСП) не становлять значної загрози. Це створює помилкову впевненість у тому, що спеціалізовані заходи захисту для них не є необхідними. Проте аналіз останніх звітів про кіберінциденти демонструє протилежне: мережі SOHO та МСП дедалі частіше стають мішенню зловмисників. Більше того, статистика свідчить, що саме цей сектор мереж зазнав основної частки кібернападів упродовж останніх років.

Популярність SOHO та МСП серед зловмисників пояснюється кількома ключовими факторами. По-перше, такі мережі є значно численнішими, ніж корпоративні, що надає ширше поле для атак. Також вони охоплюють різні галузі, що розширює можливості для кіберзлочинців. По-друге, недостатня увага до кібербезпеки є типовою для цього сектору. Згідно зі звітами State of Cloud Security та Cybersecurity INSIDERS, близько половини організацій стикалися з витоками даних чи вразливостями через неправильну конфігурацію хмарної інфраструктури. Хоча в останні роки ситуація дещо покращилася, і витоки втратили статус головної загрози, приблизно чверть компаній все ще звітують про інциденти, пов'язані з хмарними технологіями. Значна частка цих інцидентів виникає через слабкі або скомпрометовані облікові записи.

Швидка цифровізація, прискорена масовим переходом компаній до онлайн-середовища, у шість разів збільшила кількість атак на комп'ютерні системи і мережі. Цей сплеск вразливостей ще більше ускладнився через воєнні дії з лютого 2022 року, які поставили мережі підприємств малого й середнього бізнесу під безпрецедентну загрозу. За наявною статистикою, більше половини

комп’ютерних систем і мереж сфери малого та середнього бізнесу виявляються зовсім не готовими до протидії подібним викликам.

Додатково варто відзначити, що понад половина малих і середніх підприємств взагалі не передбачає бюджету на кібербезпеку, і при цьому не використовують жодних інструментів захисту, а третина їх покладаються на безкоштовні технології. Такий підхід призводить до дефіциту кваліфікованого персоналу та слабкого рівня апаратного й програмного забезпечення. Безкоштовні засоби захисту, розроблені для домашнього використання, не є ефективними в умовах корпоративного середовища, оскільки вони орієнтовані на інші типи загроз та не здатні протистояти атакам, спрямованим на порушення роботи бізнес-сервісів.

Дослідження вказують на те, що відновлення після подібних атак у кожному третьому випадку займає понад тиждень. При цьому хибне уявлення про те, що відсутність цінних ресурсів знижує інтерес кіберзлочинців до таких мереж, часто стає причиною недооцінки ризиків. Навіть якщо не враховувати потенційну втрату конфіденційних даних чи інтелектуальної власності, компанії можуть бути використані як посередники для організації атак на інші цілі.

Таким чином, мережі SOHO та МСП залишаються в зоні ризику й потребують підвищеної уваги до кібербезпеки. Їхня захищеність має розглядатися комплексно, із особливим акцентом на вибір простих у налаштуванні, ефективних апаратних і програмних рішень для забезпечення безпеки.

1.2 Основні компоненти безпеки комп’ютерних систем та мереж

Хоча часто вважається, що безпека мережі залежить насамперед від використовуваних програмних і апаратних засобів захисту, навіть побіжний погляд на статистику показує, що концепція безпеки є комплексною і що жоден компонент сам по собі не може гарантувати надійний захист. Тому, перш ніж захистити мережу, необхідно зрозуміти основні фактори безпеки: Відповідно до

CyBOK [7], фактори безпеки можна розділити на дві категорії: ─ елементи, пов'язані з захистом від атак і запобіганням вторгнень, ─ елементи, пов'язані з людськими ресурсами, організацією та правовими аспектами. У той же час, інформаційну безпеку можна розділити на три основні елементи: безпека інфраструктури, системна безпека та безпека додатків і платформ. Безпека на рівні інфраструктури використовує спеціальні заходи для захисту каналів зв'язку та регулювання фізичної безпеки інфраструктури.

Хоча фізичні атаки є дорогими і більш поширеними у великих мережах і на підприємствах, низький рівень безпеки в мережах SOHO і МСБ може дати можливість зловмисникам здійснити фізичні атаки. Тому базовим рівнем безпеки на рівні інфраструктури для таких мереж має бути шифрування та захист мережевих з'єднань, а також обмеження фізичного доступу до критично важливих вузлів мережі, таких як сервери. Безпека на рівні системи включає використання захищеної операційної системи та безпечних методів автентифікації, авторизації та обліку як на рівні окремих вузлів, так і на рівні розподіленої системи (якщо це можливо). Для невеликих мереж цей рівень включає використання сучасних операційних систем з оновленнями безпеки, шифрування цінних даних, що зберігаються в мережі, і використання доступних інструментів AAA (автентифікації, авторизації та обліку). Безпека додатків і платформ включає використання безпечного програмного забезпечення. Це означає, що мережеве програмне забезпечення має бути актуальним, а оновлення безпеки повинні бути встановлені. Крім того, слід звернути увагу на життєвий цикл використовуваного програмного забезпечення та своєчасне втручання у виведення з експлуатації певних версій програмного забезпечення. Фактори, пов'язані з людськими ресурсами, можна розділити на кілька груп, основна з яких пов'язана з поведінкою користувачів мережі. Саме людський фактор є причиною ослаблення мережевої безпеки та витоку інформації. У таких випадках на перший план виходить поняття культури інформаційної безпеки [1]. Зловмисники часто використовують загрози, орієнтовані на людину, такі як фальшиві електронні листи, посилання, фальшиві профілі тощо, які користувачі

використовують під час роботи, що дозволяє зловмисникам проникнути в мережу. Таким чином, недбалість користувачів може призвести до серйозної шкоди та втрати інформації; на думку Niekerk [8], основною причиною цього є те, що користувачі часто не мають достатнього рівня знань, оскільки інформаційна безпека не має прямого відношення до їхньої роботи. Тому навчання користувачів основам інформаційної безпеки є вирішальним фактором у забезпеченні інформаційної безпеки в корпоративних мережах, особливо в мережах SOHO і SMB. У світлі вищесказаного, мережева безпека повинна бути комплексною і враховувати всі доступні рівні та елементи. При цьому захист мережі повинен реалізовуватися як на рівні політики безпеки, так і за допомогою рішень безпеки. Рішення безпеки не є еквівалентом політики безпеки. Рішення безпеки підтримують політики безпеки, але не замінюють їх. Ця відмінність може здатися очевидною, але вона має тенденцію розмиватися в процесі проектування, якщо компанія не має чітко визначеної політики. МСП, які не мають достатніх ресурсів для забезпечення внутрішньої мережевої безпеки своїх співробітників, швидше за все, не мають політики безпеки і покладаються на персонал, який займається розробкою рішень з безпеки [9]. Хоча обидва завдання є важливими для надійного захисту, розробка політики безпеки може мати інші юридичні наслідки, ніж розробка рішення з безпеки для її реалізації, а відсутність політики може негативно вплинути на якість розробленого рішення. Вона також може визначити групу людей (зазвичай системних адміністраторів або адміністраторів безпеки), відповідальних за мережеву безпеку. Від кваліфікації та сумлінної роботи цих людей залежить правильне створення та функціонування як самої мережі, так і функцій безпеки. Окрім людського фактору, значний вплив на мережеву безпеку має також архітектура мережі. Великі мережі зазвичай мають деревоподібну структуру і поділяються на домени, де політики безпеки визначаються відповідно до вимог і функцій мережі. Однак мережі SOHO і SMB зазвичай не містять такої великої кількості вузлів, що робить архітектуру простішою. Такі мережі, підключені до глобальної мережі Інтернет, зазвичай мають брандмауери або маршрутизатори з функцією

брандмауера, які відповідають за фільтрацію шкідливого трафіку ззовні. Крім того, якщо є окремі зони, функція поділу мережі на зони також покладається на брандмауер. Наприклад, веб-сервер, до якого потрібен доступ ззовні, повинен бути розміщений в DMZ (буферній зоні), де він не може отримати доступ до основної мережі, щоб зловмисники не могли використовувати його як плацдарм для атаки на інші вузли мережі. Інші вузли в таких мережах зазвичай підключаються через набір комутаторів і бездротових точок доступу [5]. Структура типової SMB-мережі показана на рисунку 1.1. На додаток до веб-серверів у демілітаризованій зоні, файлові сервери та мережеві принтери підключені до основної зони мережі.

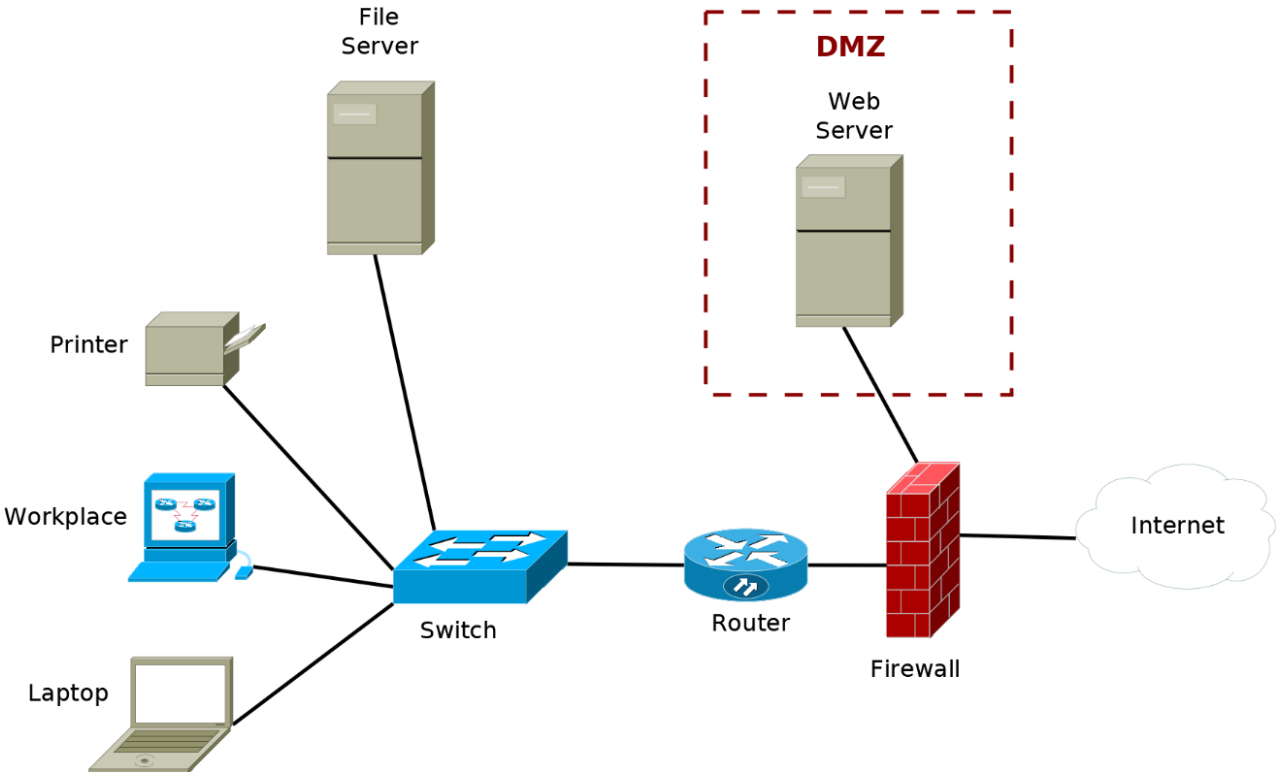


Рисунок 1.1 – Схема комп’ютерної мережі з буферною DMZ зоною

Буферна зона (DMZ) зазвичай є зоною захисту, яка забезпечує додатковий рівень безпеки для внутрішньої локальної мережі організації, блокуючи ненадійний трафік. Головною метою створення буферної зони DMZ є захист комп’ютерної мережі організації, одночасно дозволяючи доступ до загальнодоступного Інтернету для певних сервісів [11]. Ця буферна зона

зазвичай використовується для розміщення зовнішніх серверів і ресурсів, таких як DNS-системи, FTP-сервери, поштові сервери, проксі-сервери, VoIP і веб-сервери. Вони ізольовані від доступу до локальної мережі, проте можуть бути взаємо доступними через Інтернет, не відкриваючи прямий шлях до внутрішніх систем.

Такий підхід суттєво ускладнює для зловмисників проникнення до внутрішнього середовища організації через Інтернет. Буферні зони обмежуються за допомогою брандмауерів, які фільтрують трафік між DMZ та локальною мережею. Локальні (внутрішні) комп'ютерні системи в DMZ додатково захищені шлюзами безпеки, які блокують ненадійні запити ззовні [4]. Оптимальне розміщення буферної зони передбачає її розташування між двома брандмауерами: якщо хакер обійде один захисний механізм і потрапить до DMZ, він також мусить подолати внутрішній брандмауер перед доступом до конфіденційних даних.

Ключова перевага буферних зон DMZ полягає в тому, що вона мінімізує ризики доступу до закритих даних і серверів, створюючи граничну зону між відвідувачами Інтернету й внутрішньою комп'ютерною мережею організації. Це дозволяє значно підвищити рівень безпеки локальної комп'ютерної мережі. Буферна зона також надає такі переваги:

1. Контроль доступу. Організації можуть забезпечувати користувачам контрольований доступ до позамережевих сервісів через загальнодоступний Інтернет.

2. Ізоляція мережі. Така ізоляція унеможливлює доступ неавторизованих користувачів до закритих сервісів внутрішньої мережі.

3. Захисний буфер. DMZ виконує роль бар'єра між приватною та загальнодоступною мережею, додаючи проміжний рівень захисту.

Крім того, буферна зона DMZ часто використовує проксі-сервери для централізації трафіку, що спрощує його моніторинг та аудит. Це також допомагає запобігти мережевій розвідці, оскільки створений бар'єр обмежує можливості зловмисників досліджувати й виявляти уразливості у приватній

мережі. Хоча внутрішні мережі з буферними зонами DMZ є загальнодоступними, їхній трафік контролюється брандмауерами. Навіть у випадку компрометації систем DMZ, внутрішній брандмауер надійно відокремлює зону від локальної внутрішньої комп’ютерної мережі, запобігаючи проникненню у внутрішнє середовище.

Ще одним важливим аспектом є захист протоколу Інтернету. Зловмисники часто намагаються підмінити IP-адресу, видаючи себе за авторизований пристрій для доступу до системи. За допомогою DMZ можливе верифікування IP-адрес і блокування незаконних спроб доступу. Наявність сегментації мережевих потоків також забезпечує регулювання трафіку й полегшує віддалений доступ до сервісів внутрішньої мережі у контрольованому оточенні.

1.3 Забезпечення захисту комп’ютерних систем та мереж з використанням блокчейн-технологій

Блокчейн - це цифрові реєстри, що піддаються явній перевірці, які реалізовані у захищений від підробки, децентралізований спосіб (тобто без центрального сховища) і часто без централізованого контролю. На найпростішому рівні вони дозволяють спільноті користувачів реєструвати транзакції в книзі, яка є спільною для цієї спільноти, і за нормальної роботи мережі блокчейн транзакції не можуть бути змінені після їхньої публікації. У 2008 році ідея блокчейну була впроваджена в кілька інших технологій і об'єднала обчислювальні концепції, так народилася сучасна криптовалюта. Біткойн - перша така криптовалюта, заснована на блокчейні.63 Основні ідеї технології блокчейн з'явилися наприкінці 1980-х - на початку 1990-х років: У 1989 році Леслі Лампорт розробив протокол Paxos, а в 1990 році він подав статтю під назвою «Парламент за сумісництвом» [5] до ACM Transactions on Computer Systems. Ця стаття була опублікована у випуску 1998 року. У статті описується модель консенсусу для узгодження результатів у комп'ютерній мережі, де комп'ютери та мережа можуть бути ненадійними, 1991 р. Ланцюжок підписаної

інформації використовувався як електронна книга, а документи підписувалися цифровим підписом, щоб можна було легко показати, що жоден з підписаних документів у колекції документів не був підроблений [4]. Ці концепції були об'єднані та застосовані до електронних коштів у 2008 році в роботі «Біткойн»: ця робота була опублікована під псевдонімом Сатоші Накамото, який згодом заснував мережу блокчейн криптовалюти біткойн у 2009 році. Стаття Накамото містила концепції, яких дотримується більшість сучасних криптовалютних систем (хоча і з варіаціями та модифікаціями). Біткойн - це лише перший з багатьох додатків, заснованих на блокчейні. Використання блокчейну дозволило реалізувати біткойн децентралізовано, що сприяло його активному використанню, оскільки жоден користувач не контролює електронні кошти і немає єдиної точки відмови. Головною перевагою цього рішення було те, що воно дозволило здійснювати прямі транзакції між користувачами без необхідності залучення довіреної третьої сторони. Це також дозволило випускати нові криптовалюти відповідно до вбудованих алгоритмів користувачам, які випускали нові блоки і зберігали копію реєстру. Такі користувачі відомі в мережі Bitcoin (та інших криптовалютних мережах) як майнери. Автоматизація виплат майнерам дозволила децентралізувати систему без потреби в організації. Використовуючи блокчейн і підтримку на основі консенсусу, було створено механізм самоуправління, який гарантує, що в блокчейн додаються тільки дійсні транзакції і блоки. У мережі Біткоїн блокчейн дозволив користувачам бути псевдоанонімними. Це означає, що користувачі є анонімними, але ідентифікатори облікових записів - ні, а також те, що всі транзакції є публічними. Це дозволило Біткоїну запропонувати псевдоанонімність в реальності, оскільки облікові записи можна створювати без необхідності проходити процеси реєстрації, ідентифікації та автентифікації [6]. Оскільки біткойн є анонімним, було вкрай важливо мати механізм для створення довіри в середовищі, де користувачів не можна легко ідентифікувати. До появи технології блокчейн така довіра зазвичай передавалася через посередника, якому довіряли обидві сторони. За відсутності довіреного посередника, довіра

необхідна всередині мережі блокчейн, і це забезпечується чотирма ключовими особливостями технології, описаними нижче:

Реєстри - ця технологія використовує адитивний реєстр для забезпечення повної історії транзакцій. На відміну від традиційних баз даних, транзакції та значення, що зберігаються в блокчейні, не змінюються.

Безпека - блокчейн криптографічно захищений, тому дані в реєстрі не можуть бути підроблені, а дані в реєстрі перевіряються. Відкритість - реєстр ділиться між кількома учасниками. Це забезпечує прозорість для всіх вузлів мережі блокчейн.

Децентралізованість - блокчейн може бути децентралізованим, а це означає, що кількість вузлів у мережі може бути збільшена, щоб підвищити її стійкість до атак з боку користувачів. Збільшення кількості вузлів зменшує можливість зловмисників впливати на протоколи консенсусу, що використовуються в блокчейні.

У мережах блокчейн, де будь-хто може створити обліковий запис і брати участь анонімно (так звані мережі блокчейн без дозволу), ці функції можуть забезпечити рівень довіри між сторонами, навіть якщо вони не знають один одного заздалегідь. Ця довіра дозволяє приватним особам і організаціям здійснювати транзакції безпосередньо один з одним, прискорюючи їх здійснення і знижуючи вартість таких транзакцій. У блокчейн-мережах, де доступ більш жорстко контролюється (так звані блокчейн-мережі з контрольованим доступом), де між користувачами може існувати певний рівень довіри, така функція може допомогти побудувати довіру. Важливим компонентом технології блокчейн є використання криптографічних hash-функцій для багатьох транзакцій. Хешування - це метод застосування криптографічної hash-функції до даних для обчислення відносно унікального результату (який називається дайджест повідомлення або просто хеш) для вхідних даних практично будь-якого розміру (файл, текст, зображення тощо). Це гарантує, що дані не підроблені і не спотворені. Навіть невеликі зміни у вхідному сигналі (наприклад, зміна одного біта) можуть призвести до абсолютно різних вихідних дайджестів.

Криптографічні hash-функції мають наступні важливі властивості безпеки: їх прототипи важко знайти. Це означає, що вони є односторонніми. Тому неможливо обчислити правильне вхідне значення, знаючи деяке вихідне значення (наприклад, знайти x таке, що hash (x) = hash для заданого хешу). Існує другий опір прототипуванню. Тобто, неможливо знайти вхідний хеш для заданого вихідного. Іншими словами, криптографічні hash-функції розроблені таким чином, що для заданого входу неможливо знайти другий вхід, який дає такий самий результат (наприклад, для заданого x знайти y таким чином, щоб hash (x) = hash (y)). Єдиний можливий спосіб - це вичерпний перебір вхідного простору, що неможливо зробити за достатній час, щоб мати хоч якийсь шанс на успіх. Стійкість до зіткнень. Це означає, що неможливо знайти два входи з однаковим вихідним хешем. Тобто, обчислювально неможливо знайти два набори вхідних даних, які дають однаковий хеш (наприклад, знайти x і y такі, що hash (x) = hash (y)).

Криптографічна hash-функція, яка використовується в багатьох блокчейн-додатках, - це алгоритм безпечного хешування (SHA) (SHA-256) з вихідним розміром 256 біт. Багато комп'ютерів підтримують цей алгоритм у своєму апаратному забезпеченні і можуть швидко його обчислювати: Результат SHA-256 становить 32 байти (1 байт = 8 біт, 32 байти = 256 біт) і зазвичай відображається у вигляді шістнадцяткового рядка з 64 символів. Це означає, що може бути 2 256 ≈ 10 77 hash-значень. Алгоритм SHA-256, як і багато інших, визначений у Федеральному стандарті обробки інформації (FIPS) 180-4 [10]. На сайті NIST's Secure Hashing Website [11] перераховані специфікації FIPS для всіх затверджених NIST алгоритмів хешування. Оскільки існує нескінченна кількість можливих вхідних значень і скінченна кількість можливих вихідних hash-значень, можливо, але дуже малоймовірно, що хеш (x) = hash (y) (тобто, що два різних вхідних хеші дають один і той же хеш). У цьому випадку SHA-256 вважається стійким до колізій, оскільки для того, щоб знайти колізію в SHA-256, потрібно в середньому 2128 запусків алгоритму. У мережах блокчейн криптографічні hash-функції використовуються для вирішення низки завдань,

серед яких: захист даних блоку - публікація вузлами hash-даних блоку для генерації хешу, що зберігається в заголовку блоку. Захист заголовка блоку - вузол публікації хешує заголовок блоку. Оскільки заголовок блоку містить хешоване представлення даних блоку, самі дані блоку також захищені, коли хеш заголовка блоку зберігається в наступному блоці.

У технології блокчейн використовується ряд сімейств криптографічних hash-функцій, в тому числі Keccak (визнаний NIST кращим на створення стандарту хешування SHA-3) і RIPEMD-160 [12] (SHA-256 не є єдиним). Транзакції в блокчейні - це взаємодія між сторонами. У випадку криптовалют, наприклад, транзакція - це передача криптовалюти між користувачами мережі блокчейн. Кожен блок у блокчейні може містити одну або кілька транзакцій. Залежно від реалізації блокчейну, безперервний потік нових блоків (навіть за відсутності транзакцій) є важливим для підтримки безпеки мережі блокчейн. Безперервна трансляція нових блоків не дозволяє зловмиснику наздогнати і створити новий, довший, модифікований блокчейн. Хоча дані, які складають транзакцію, можуть відрізнятися в різних реалізаціях блокчейну, механіка транзакції в основному однакова. Інформація, що надсилається в мережу блокчейн, включає адресу відправника (або інший відповідний ідентифікатор), відкритий ключ відправника, цифровий підпис, вхідні та вихідні дані транзакції або дані, записані в транзакції. Технологія блокчейн використовує криптографію з асиметричним ключем. Криптографія з асиметричним ключем використовує математично пов'язану пару ключів - публічний і приватний ключі. Відкритий ключ є загальнодоступним без загрози для безпеки транзакції, в той час як закритий ключ повинен зберігатися в таємниці для криптографічного захисту даних; між цими двома ключами існує взаємозв'язок, але закритий ключ не може бути ефективно визначений на основі знання відкритого ключа. Можливе шифрування за допомогою закритого ключа і розшифрування за допомогою відкритого ключа. Або ж можливе шифрування за допомогою відкритого ключа і розшифрування за допомогою закритого ключа. Криптографія з асиметричним ключем забезпечує довірчі відносини між користувачами, які не знають і не

довіряють один одному, і надає механізм для перевірки цілісності та автентичності транзакцій, в той же час дозволяючи залишати транзакції відкритими. Для цього транзакції підписуються «цифровим підписом». Це означає, що для шифрування транзакції використовується приватний ключ, а розшифрувати її може будь-хто, хто має відкритий ключ. Оскільки відкриті ключі є загальнодоступними, шифрування транзакції за допомогою закритого ключа доводить, що особа, яка підписує транзакцію, має доступ до закритого ключа. Крім того, для шифрування даних можна використовувати відкритий ключ користувача, щоб розшифрувати дані тільки ті користувачі, які мають доступ до закритого ключа. Недоліком є те, що криптографія з асиметричним ключем часто є повільною в обчислювальному плані.

Найпоширеніші способи використання криптографії з асиметричним ключем у багатьох мережах блокчейн: приватні ключі для цифрового підпису транзакцій, відкриті ключі для отримання адрес і відкриті ключі для перевірки підписів, згенерованих закритими ключами.

Криптографія з асиметричним ключем надає можливість перевірити, що користувач, який надсилає значення іншому користувачеві, має приватний ключ, який може підписати цю транзакцію. Деякі мережі блокчейн можуть використовувати існуючу бізнес-інфраструктуру відкритих ключів для криптографії з асиметричним ключем для надання облікових даних користувачів, замість того, щоб кожен користувач мережі блокчейн мав власний асиметричний ключ. Це досягається завдяки використанню існуючих служб каталогів і використанню цієї інформації в мережі блокчейн.

Мережі блокчейн, що використовують існуючі служби каталогів, отримують доступ до каталогу за допомогою існуючих протоколів, таких як Lightweight Directory Access Protocol (LDAP) [13], і використовують інформацію каталогу локально. передається до внутрішнього центру сертифікації мережі блокчейн. Технологія блокчейн зазвичай використовує як розподілене право власності, так і розподілену фізичну архітектуру для захисту цілісності збережених даних. Розподілена фізична архітектура мереж блокчейн зазвичай

включає набагато більший набір комп'ютерів, ніж зазвичай використовується для розподіленої фізичної архітектури з централізованим управлінням. Зростаючий інтерес до розподіленого володіння бухгалтерськими книгами зумовлений потенційними загрозами довірі, безпеці та надійності, пов'язаними з централізованим володінням бухгалтерськими книгами. Переваги розподіленої системи блокчейн над системою з централізованим зберіганням даних показані в Таблиці 1.1.

Таблиця 1.1 – Переваги і недоліки форматів зберігання даних

Централізоване зберігання	**Розподілена блокчейн система**
Ризик втрати даних, уся відповідальність за резервне копіювання та цілісність на власнику обладнання	Безліч резервних копій розподілених між усіма учасниками блокчейн мережі значно підвищує надійність
Ризик однорідності архітектури, що спрощує проведення атак	Обладнання зазвичай різнорідне, що унеможливлює використання одних і тих же атак на всі вузли
Часто дані локалізовані в певному невеликому регіоні, ризик пошкодження усього обладнання одночасно.	Географічна різноманітність підвищує захищеність на випадок непередбачуваних подій
Можливість підробки транзакцій власником сховища, або не збереження деяких транзакцій	Криптографічна захищеність та цілісність даних гарантована алгоритмами блокчейн мережі

Централізовано збережені записи можуть бути втрачені або знищені. Користувачі повинні довіряти тому, що власник правильно створює резервні копії своїх даних. З іншого боку, мережі блокчейн децентралізовані і створюють кілька резервних копій, які автоматично оновлюються і синхронізуються, так що важлива інформація завжди реплікується між головними вузлами. Важливою перевагою технології блокчейн є те, що кожен користувач може керувати своєю власною копією книги. Щоразу, коли новий головний вузол приєднується до

мережі блокчейн, він знаходить інші головні вузли і завантажує повну копію книги в мережі блокчейн.

Централізований реєстр може зберігатися в однорідній мережі, де програмне забезпечення, апаратне забезпечення та мережева інфраструктура є однаковими. Ця особливість може знизити стійкість всієї системи, оскільки успішна атака на одну частину мережі може бути застосована до інших частин мережі. У той же час, мережі блокчейн є гетерогенними мережами з різним програмним і апаратним забезпеченням в пристроях і мережевій інфраструктурі. Через велику кількість відмінностей між вузлами мережі блокчейн, атака на один вузол не гарантує успіху при спробі атакувати інший вузол.

Централізовано збережені реєстри можуть знаходитися повністю в географічно обмеженій області (наприклад, в межах країни). Збій мережі в цьому місці може призвести до того, що послуги, які залежать від цього сховища, стануть недоступними. На відміну від централізованого сховища, мережа блокчейн може складатися з географічно віддалених вузлів по всьому світу. Тому однорангові блокчейн-мережі є стійкими до втрати будь-якого вузла або навіть цілої групи вузлів. Транзакції з централізованими сховищами не є прозорими, і легітимність транзакцій до таких сховищ може бути легко поставлена під сумнів, що вимагає від користувачів віри в те, що власник перевірив кожну отриману транзакцію. Мережа блокчейн автоматично перевіряє, чи всі транзакції є дійсними, виконуючи верифікацію. Якщо зловмисний вузол надсилає недійсну транзакцію, інші вузли виявляють це і ігнорують такі транзакції, запобігаючи поширенню недійсних транзакцій по всій мережі блокчейн. Список транзакцій у центральному реєстрі може бути неповним, і користувачі повинні довіряти тому, що власник реєстру включає всі отримані дійсні транзакції. З іншого боку, мережі блокчейн зберігають всі прийняті транзакції в своїх розподілених книгах. Створення нового блоку вимагає посилання на попередній блок. Якщо блок не містить правильного посилання на попередній блок, інші вузли відхилять його. Дані про транзакції в централізованому реєстрі можуть бути змінені, і користувачі повинні довіряти

тому, що власник реєстру не змінить попередні транзакції. Мережі блокчейн забезпечують зберігання даних з явними механізмами захисту від підробки записів за допомогою криптографічних механізмів, таких як цифрові підписи та криптографічні hash-функції.

Централізовані системи можуть бути ненадійними або недостатньо захищеними, і користувачі повинні довіряти тому, що комп'ютерні системи та мережі, які в них задіяні, отримують критичні виправлення безпеки. Мережі блокчейн, через свою децентралізовану природу, не мають центральної точки атаки. Як правило, інформація про мережу блокчейн є загальнодоступною і не дає підстав для крадіжки. Щоб атакувати користувача мережі блокчейн, зловмисник повинен націлитися на цього користувача індивідуально. Будь-яка спроба атакувати сам блокчейн зустріне опір з боку чесних вузлів системи. Якщо не замінити один вузол, то постраждає лише цей вузол, а не система в цілому. Транзакції додаються до блокчейну за допомогою майнінгових вузлів, які випускають блоки. Блок складається із заголовка та даних блоку. Заголовок блоку містить метадані для цього блоку. Дані блоку містять список перевірених і дійсних транзакцій, розміщених в мережі блокчейн. Дійсність і автентичність забезпечуються шляхом перевірки правильності форматування транзакцій і того, що постачальник цифрових активів кожної транзакції підписав транзакцію. Це підтверджує, що постачальник цифрових активів транзакції має доступ до приватного ключа, який може підписати цей цифровий актив або дані. Інші повні вузли перевіряють автентичність блоку і дійсність всіх транзакцій у випущеному блоці і не приймають блок, якщо він містить несанкціоновану транзакцію.

Оскільки кожен блок містить хеш-дайджест заголовка попереднього блоку, блоки з'єднуються в ланцюжок, утворюючи блокчейн. Якщо раніше опублікований блок змінюється, він матиме інший хеш. Це означає, що всі наступні блоки також матимуть різні хеші, що містять хеш попереднього блоку. Це полегшує виявлення та відхилення модифікованого блоку [14]. Коли користувачі приєднуються до мережі блокчейн, вони домовляються про початковий стан системи. Це записується в генезисі - заздалегідь визначеному

блоці. Кожна мережа блокчейн має публічний генетичний блок, і кожен блок повинен додаватися до блокчейну після генетичного блоку на основі узгодженої моделі консенсусу. Однак, незалежно від моделі, кожен блок повинен бути дійсним і незалежно перевірятися кожним користувачем мережі блокчейн.

Поєднання початкового стану і можливості перевірки кожного блоку дозволяє користувачам незалежно погоджувати поточний стан блокчейну. Якщо повний вузол отримує два валідних ланцюжка, механізм за замовчуванням у більшості блокчейн-мереж полягає в тому, що «довший» ланцюжок вважається правильним і приймається ланцюжок, створений з найбільшими зусиллями.

Важливою особливістю технології блокчейн є те, що немає необхідності в довіреній третій стороні для надання інформації про стан системи. Допускаються тимчасові розбіжності, але в кінцевому підсумку всі вузли повинні досягти спільної згоди для додавання нових блоків до блокчейну.

Блокчейн впливає на вдосконалення IDS, але не всі проблеми IDS можна вирішити за допомогою технології блокчейн. На їхню думку, технологія блокчейн може бути використана для підвищення продуктивності IDS, особливо CIDS, з точки зору обміну даними та обчислювальної надійності. Технологія блокчейн є новим рішенням, але вона все ще страждає від деяких притаманних їй проблем і обмежень:

– енергія і витрати: потужність комп’ютерної системи є проблемою для використання блокчейну. На прикладі майнінгу біткоїна для обчислення і перевірки транзакцій потрібна велика кількість енергії; обчислювальна потужність спочатку додається одним майнером, але зростає експоненціально в міру зростання мережі;

– захист і безпека: існуючі додатків на основі блокчейну використовують «розумні» транзакції та контракти, які не повинні бути анонімними, що викликає питання конфіденційності та безпеки даних, які зберігаються в загальному реєстрі. Крім того, сама технологія блокчейн може стати головною мішенню для кіберзлочинців, що призводить до різноманітних атак, включаючи розподілені атаки на відмову в обслуговуванні;

– тривалість і складність: внаслідок специфічної структури транзакції на основі блокчейну можуть тривати кілька годин, поки всі залучені сторони не оновлять свої реєстри. Ця затримка створює багато невизначеності для учасників транзакції і відкриває шлях для кіберзлочинців.

У цьому випадку кожна IDS обмінюється інформацією з іншими вузлами IDS і зовнішніми хостами. При цьому дані, пов'язані з безпекою, якими обмінюються учасники CIDN, зберігаються в блокчейні у вигляді довіреного ланцюжка, щоб запобігти втручанню зловмисних вузлів. Для використання в якості протоколу консенсусу пропонується комбінація PoW і PoS. Згідно з цим протоколом, довірені вузли IDS з більшою обчислювальною потужністю і вищими ставками з більшою ймовірністю будуть обрані для створення наступного блоку; згідно з філософією протоколу PoS (і, отже, PoW), вузол з найбільшою сумою ставки з більшою ймовірністю буде обраний для створення наступного блоку. Поєднання протоколів PoW і PoS в ланцюжку довіри дозволяє використовувати гібридний майнінг - метод вибіркового консенсусу. У гібридній схемі майнінгу ймовірність того, що IDS вибере довірений вузол в якості лідера, висока і залежить як від обчислювальної потужності вузла, так і від швидкості його обчислень. Перевага цієї гібридної схеми полягає в тому, що вона дозволяє уникнути ситуацій, коли довірений вузол IDS з великою часткою може безперервно генерувати всі блоки.

Ця розробка адаптує блокчейн для розповсюдження та створення спільної довіреної бази даних підписів. Ця архітектура підвищує надійність та ефективність виявлення вторгнень у мережах IoT порівняно з локальними IDS, а також підвищує стійкість систем виявлення вторгнень до флуд-атак. Блокчейн-компонент виконує три основні ролі:

Peer- 2- Peer-зв'язок. Цей компонент відповідає за зв'язок з іншими вузлами IDS в мережі та організацію робочих, адміністративних і фізичних комунікацій.

Common work. Цей компонент використовується IDS-вузлом для збору інформації, необхідної для оцінки надійності цільового вузла, і для відправки відповідного зворотного зв'язку на запити інших вузлів.

Trust mode. Цей компонент відповідає за довірчі обчислення та забезпечення репутації вузла. Наприклад, механізм довіри на основі викликів досліджує репутацію вузла, порівнюючи отриманий відгук з очікуваною відповіддю [15].

У даній роботі розглядаємо випадок, якщо зловмисник може контролювати один або декілька вузлів у CIDN, але не може контролювати велику кількість вузлів IDS за короткий час. Крім того, оскільки кожен вузол в CIDN має пару приватних і публічних ключів, ідентифікаторами не можна легко маніпулювати або дублювати. Це пов'язано з тим, що якщо зловмисник отримує контроль над великою кількістю вузлів в такій системі, стабільна робота блокчейну стає неможливою, або всередині розподіленої системи відбуваються постійні колізії, або блокчейн розпадається на кілька паралельних вузлів, що призводить до реальної фрагментації мережі.

РОЗДІЛ 2
МОДЕЛЬ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

2.1 Модель комп’ютерної системи виявлення загроз на основі технології блокчейн

Децентралізована система виявлення вторгнень для мереж малого та середнього бізнесу на основі блокчейну є частиною загальної системи управління кібербезпекою в організації. Для того, щоб побудувати децентралізовану систему захисту комп'ютерних мереж на основі блокчейну, необхідно розробити базові стандарти, що впливають на безпеку комп'ютерних мереж МСБ. Розробити обчислювальні моделі для виявлення та аналізу аномальних подій з метою захисту комп'ютерних мереж МСБ на основі блокчейну. Розробити імітаційні моделі для систем безпеки розподілених комп'ютерних мереж на основі блокчейну. Провести тестування системи у штучно створеному мережевому середовищі, подібному до виробничого, та в реальному мережевому середовищі.

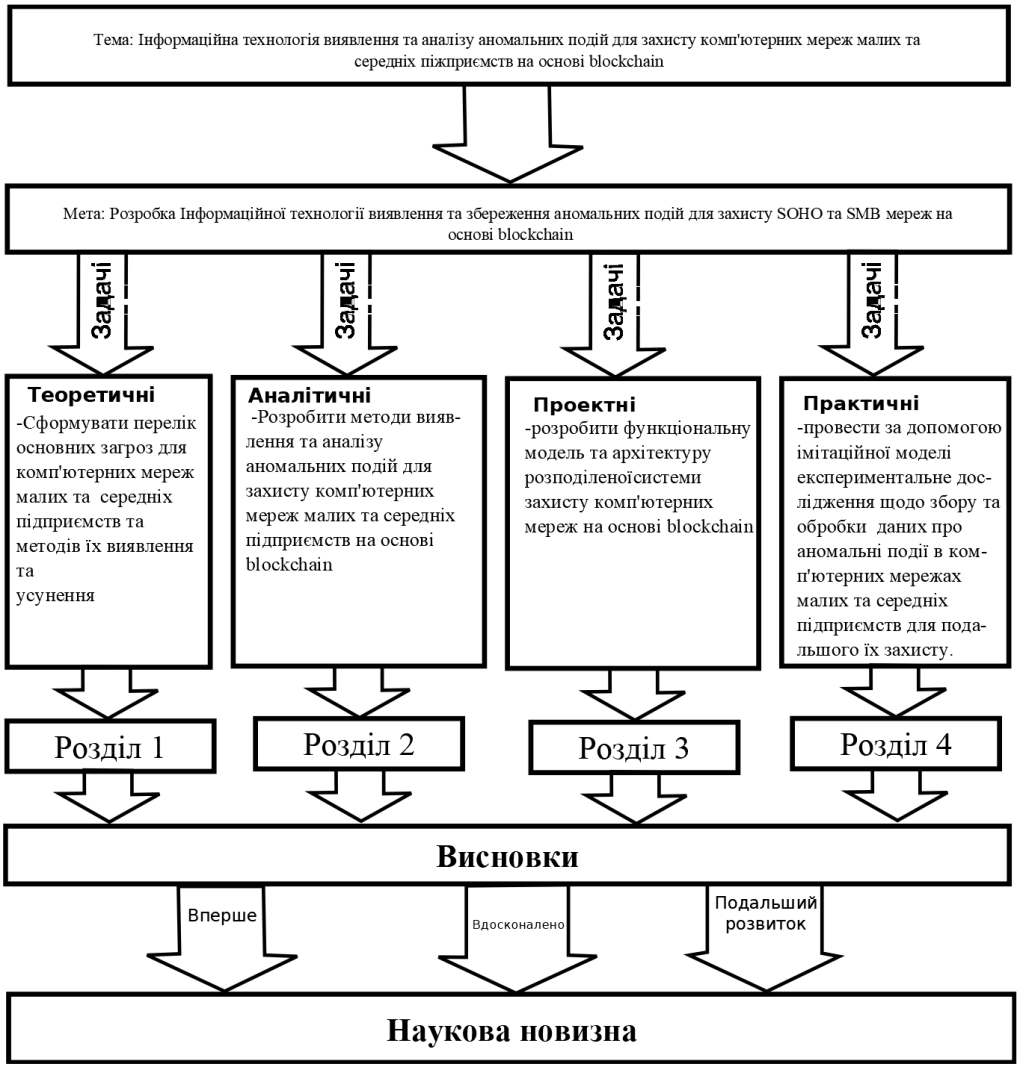


Рисунок 2.1 – Структурно-логічна модель дослідження

Розповсюджені комп’ютерні системи виявлення вторгнень, які зазвичай використовуються для захисту мереж SOHO і SMB, мають обмежені можливості для забезпечення належного рівня безпеки. Це пов'язано з тим, що вони зосереджені лише на локальних наборах правил і не враховують ширший контекст загроз. У сучасному світі, де постійно зростає кількість нових атак, важливо мати можливість швидко і адекватно реагувати на них.

Для вирішення цієї проблеми необхідно переосмислити архітектуру систем виявлення вторгнень. Вона має бути більш глобальною та охоплювати значну кількість локальних мереж та їх сегментів. Це забезпечить своєчасну передачу інформації між компонентами системи та раннє реагування на загрози. Наприклад, якщо одна частина системи виявить ознаки атаки, інші сегменти

мережі можуть бути краще підготовлені до мінімізації потенційних загроз і пом'якшення негативних наслідків. Важливим аспектом побудови такої системи є досягнення консенсусу між її компонентами.

Традиційно для цього використовуються централізовані сторонні вузли, які слугують надійним джерелом контролю. Однак для розподілених систем виявлення вторгнень такий підхід не є ефективним. Зі збільшенням кількості вузлів навантаження на централізований вузол значно зростає, що призводить до виникнення єдиної точки відмови (SPoF). Таким чином, централізована архітектура не тільки обмежує відмовостійкість системи, але і робить її більш вразливою до зовнішніх атак і внутрішніх збоїв. Тому, щоб побудувати ефективну і надійну розподілену систему виявлення вторгнень, необхідно перестати покладатися на централізовані вузли і впровадити децентралізований підхід, який забезпечує більшу стабільність і безпеку.

Тож структура комп’ютерної системи потребує максимальної децентралізації. Реалізувати цей підхід достатньо ефективно дозволяє технологія Blockchain.

У комп’ютерних розподілених системах і мережах виявлення загроз технологія блокчейну може слугувати ефективним засобом безпечного та надійного механізму обробки інформації між компонентами мережі.

Блокчейн представляє собою структуру даних у вигляді ланцюжка блоків, з'єднаних між собою за допомогою криптографічних хешів. Цей ланцюжок є незмінним — він може лише збільшуватися через додавання нових блоків. Кожен блок містить хеш попереднього блоку, що забезпечує цілісність інформації та унеможливлює її модифікацію у попередніх блоках. Крім цього, блок включає структуру даних у вигляді списку, яка дозволяє зберігати певний набір транзакцій. Хоча блокчейн найбільш відомий своїм застосуванням у криптовалютах, його можливості виходять далеко за межі цієї сфери.

У комп’ютерних системах виявлення вторгнень блокчейн може використовуватися для організації безпечного, децентралізованого та розподіленого сховища, яке зберігає сигнатурні бази, налаштування, а також

інформацію про виявлені вторгнення й аномальні дії. Подібні рішення пропонуються як для IoT-середовищ із врахуванням їх специфіки, так і для мереж загального призначення.

Слід зазначити, що блокчейн функціонує виключно у форматі запису даних у вигляді журналу: до такого сховища можна лише додавати нову інформацію, без можливості редагування чи видалення вже існуючих даних. Для систем виявлення вторгнень ця особливість є цілком прийнятною, оскільки дані, з якими працюють такі системи (сигнатурні бази, налаштування, журнали подій), здебільшого лише додаються до сховища.

На основі вищезазначеного можна виділити наступні переваги застосування технологій блокчейн для розподілених комп'ютерних систем і мереж виявлення загроз в SOHO та SMB:

– створення розподіленого сховища, яке не залежить від поведінки конкретного вузла або доступності іншої мережі чи сегмента, підключеного до розподіленої системи;

– перевірка цілісності даних на всіх кінцевих вузлах мережі SOHO або SMB;

– довірений обмін інформацією між вузлами розподіленої системи виявлення вторгнень без використання центрального довіреного вузла.

Серед недоліків використання технологій блокчейну для виявлення загроз в розподілених комп'ютерних системах і мережах можна віднести:

– зі збільшенням кількості розподілених мереж, підключених до розподіленої системи виявлення вторгнень, значно зростає і обсяг внутрішнього трафіку;

– оскільки неможливо виправити раніше записані дані, навіть у випадку хибної тривоги, необхідна ретельна і точна обробка наявної інформації;

– постійне збільшення розміру ланцюжка блокчейну є ще одним технічним обмеженням.

З цього можна зробити висновок, що блокчейн підходить для використання в децентралізованих системах виявлення вторгнень з великою

кількістю клієнтів, де облік і контроль клієнтів дуже складний, і прекрасно сумісний з децентралізованими системами виявлення вторгнень для мереж SOHO і SMB. Технологія блокчейн в системах виявлення вторгнень фактично виступає об'єднуючим компонентом, що забезпечує безпечний обмін і зберігання інформації. При цьому збільшення кількості вузлів блокчейну підвищує надійність та відмовостійкість системи. Концептуальна модель децентралізованої системи виявлення вторгнень на основі блокчейну наведена на рисунку 2.2, який відображає зв'язки між основними компонентами системи.

Для функціонування системи виявлення вторгнень на основі технології блокчейн потрібні спеціалізовані компоненти, зокрема повноцінні вузли блокчейн та легковагові клієнти. Повний блокчейн-вузол зберігає повну копію ланцюжка блоків, тоді як тонкий клієнт звертається до цих вузлів для отримання необхідних даних про блокчейн.

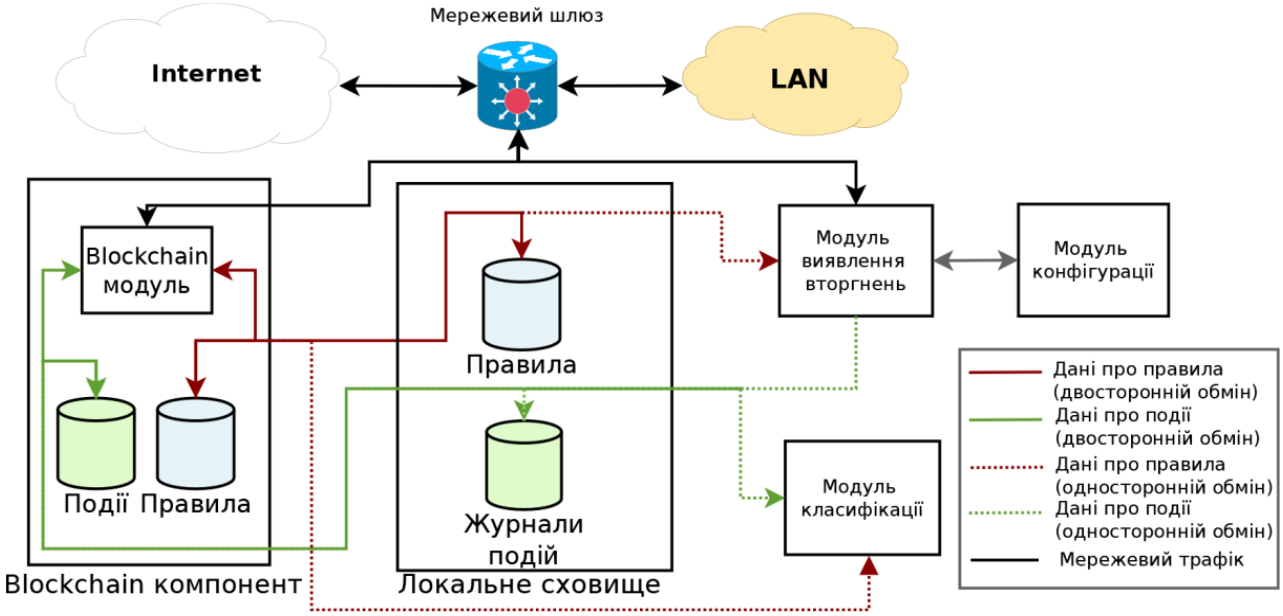


Рисунок 2.2 – Модель розподіленої комп’ютерної системи виявлення загроз на основі технології блокчейн для інформаційних мереж малих та середніх підприємств

У процесі функціонування блоки блокчейн зберігатимуть не лише інформацію про сигнатури атак, але й актуальні дані про стан кожного вузла та

підозрілу активність, яка була зафіксована хоча б одним із них. Крім того, передбачається можливість передачі службової інформації, яка має бути задокументована у журналі. Вузли системи виявлення вторгнень утворюватимуть децентралізовану peer-to-peer мережу, де кожен вузол функціонує автономно, але водночас взаємодіє з іншими вузлами, обмінюючись інформацією про стан захищеної мережі.

Одним із ключових елементів подібної системи є база даних правил із сигнатурами вторгнень. Це особливо важливо, оскільки методи виявлення на основі сигнатур є надзвичайно швидкими і потребують мінімальних обчислювальних ресурсів. Таким чином, їх можна застосовувати навіть на апаратних засобах нижчого класу. Проте зазначимо, що ефективність цього методу залежить від апаратних характеристик: занадто слабкі пристрої можуть не впоратися з обробкою трафіку навіть за використання сигнатурних методів, що може потребувати зменшення кількості активних правил.

Особливість методів виявлення на основі сигнатур полягає також у низькому рівні хибно-позитивних спрацювань порівняно з методами на основі аномалій або зі статистичними підходами. Однак ефективність роботи таких правил значною мірою залежить від попередньої обробки мережевого трафіку перед порівнянням із сигнатурами. Кожен протокол потребує специфічної процедури попередньої обробки. Наприклад, система Snort IDS використовує динамічні модулі обробки для цієї мети, причому алгоритми їхньої роботи суттєво складніші, ніж це можливо реалізувати у вигляді простого набору текстових правил.

Для забезпечення ефективності системи виявлення, заснованої на сигнатурах, необхідно здійснити її належне налаштування. Оптимальним підходом у цьому контексті є застосування спрощеного методу конфігурації, що дозволить досвідченим користувачам легко самостійно адаптувати систему. Насамперед, рекомендується класифікувати базу правил за протоколами, дозволяючи користувачам обирати лише ті протоколи, які застосовуються їхньою системою. Це сприятиме зменшенню обсягу бази даних активних правил,

а також дасть змогу відключити динамічні процесори трафіку для невикористовуваних протоколів, що знизить навантаження на процесорні ресурси.

Крім того, правила можуть бути диференційовані за рівнем чутливості. Для користувачів, які потребують максимального рівня безпеки, доречно використовувати правила з високим рівнем чутливості. Однак варто враховувати, що такі правила підвищують ризик хибно-позитивних помилок, оскільки вони можуть інтерпретувати нормальну діяльність як підозрілу. Натомість правила з нижчим рівнем чутливості краще підходять для більшості випадків використання, хоча вони можуть залишити без уваги деяку підозрілу активність.

Вимикання правил для невикористовуваних служб може створити вразливі місця в системі. Щоб уникнути цього, необхідно закрити всі порти, пов’язані з відключеними службами. Така міра значно знижує ймовірність експлуатації потенційних вразливостей у вимкнених службах. Ще одним важливим компонентом посилення безпеки є впровадження набору правил, які здатні фіксувати запити до вимкнених служб навіть за умови закритих портів. Наприклад, якщо на сервері заблоковано протокол HTTP у конфігурації IDS, а зловмисник намагається отримати доступ до цієї служби, система не лише закриє відповідні порти, але й повідомить про спробу атаки або підозрілу активність.

Ця функціональність дає змогу ідентифікувати нападників, які сканують сервіси та шукають їхні вразливості. Зокрема, якщо розподілена система IDS виявляє спроби однієї IP-адреси доступитися до портів і протоколів конкретної служби на багатьох захищених серверах, така IP-адреса може автоматично заноситися до чорного списку. Крім того, система може повідомити провайдера про виявлену шкідливу активність, що додатково посилює комплексний захист мережі.

Таким чином, постає необхідність впровадження підсистеми управління правилами. Насамперед це зумовлено використанням евристично створених правил, розроблених на основі списку підозрілих подій. Отже, список підозрілих

подій, що поширюється через блокчейн, може бути доступний для обробки усіма вузлами у випадку загальнодоступних подій (або лише обмеженим набором надійних вузлів у разі приватних подій). Вузол, який накопичує достатню кількість даних для автоматичного формування правила виявлення, створює його та додає до блокчейна як транзакцію з новим правилом. Автоматично сформовані правила відносяться до категорії високої чутливості й потребують ручної перевірки з боку фахівців із кібербезпеки, що дозволяє мінімізувати ризик підвищеного рівня хибно-позитивних спрацьовувань.

2.2 Модель блокчейн елемента розподіленої комп’ютерної системи

Розглянемо два види блокчейнів: без права доступу і з правом доступу. Блокчейни з контролем доступу можуть бути публічними або приватними (відкриті блокчейни за замовчуванням вважаються публічними). За критеріями контролю доступу сучасні блокчейн-системи можна умовно розділити на три категорії: публічні, консорціумні та приватні.

Відкриті блокчейни найчастіше використовують у криптовалютах. Їхня головна особливість полягає в тому, що будь-який пристрій, який відповідає технічним вимогам, може приєднатися до мережі та покинути її в будь-який момент. Водночас відсутній центральний орган, який контролює участь вузлів. Крім того, у таких системах зазвичай неможливо заборонити певному вузлу записувати дані в блокчейн.

У приватних блокчейнах участь у записі та читанні обмежена певним колом користувачів. Для цього створюється центральний орган, який визначає і видає вузлам права на доступ до цих операцій. Щоб забезпечити конфіденційність і кращий контроль, читачі та автори можуть працювати в окремих паралельних блокчейнах, пов'язаних між собою. У такій моделі доступ до інформації мають лише авторизовані вузли, тоді як у публічному блокчейні дані можуть переглядати та перевіряти всі охочі. Усі операції в блокчейні,

включно зі створенням і перевіркою блоків, здійснюються на основі протоколів консенсусу.

Цей механізм гарантує, що всі учасники системи мають централізований доступ до журналів записів. Тип протоколу консенсусу залежить від конкретної реалізації блокчейна і моделі загроз.

Одне з ключових завдань протоколів консенсусу - запобігти фальсифікації блоків у системах блокчейн. Для цього зазвичай використовують такі алгоритми, як Proof-of-Work, Proof-of-Stake і Proof-of-Elapsed-Time. Ці механізми обмежують неконтрольоване створення блоків і значно ускладнюють проведення атаки, якщо більше половини вузлів мережі є довіреними. Це гарантує високий рівень захисту даних у такому типі сховища. Для приватних блокчейнів набір протоколів консенсусу може бути найрізноманітнішим, серед яких часто використовується Practical Byzantine Fault Tolerance (PBFT). До переваг цього алгоритму можна віднести високу енергоефективність і менше споживання ресурсів. Однак у цього підходу є й недоліки, наприклад, складність масштабування системи.

Процес створення нових блоків у блокчейні контролюється протоколом консенсусу, який визначає механізм взаємодії між вузлами під час додавання нових даних.

Спосіб реалізації цього протоколу багато в чому залежить як від моделі загроз, так і від характеристик конкретної мережі блокчейн. Для забезпечення високого ступеня безпеки публічні блокчейни найчастіше ґрунтуються на обчислювальних алгоритмах, таких як Proof-of-Work, і підтвердженні прав власності на дефіцитні ресурси системи, наприклад, на моделі Proof-of-Stake. Водночас консорціуми і приватні блокчейни зазвичай використовують відмовостійкі алгоритми, засновані на візантійських угодах, такі як PBFT і SIEVE. Ці методи ефективніше нейтралізують згубний вплив потенційно недовірених вузлів. Проте, використання блокчейна не є універсальним рішенням усіх проблем безпечного розроблення застосунків. У деяких сценаріях використання цієї технології може навіть створити додаткові труднощі.

Наприклад, – загальний процес ухвалення рішень, що допомагає оцінити доцільність використання технології блокчейн і вибрати найбільш підходящий тип блокчейна для конкретного випадку. Основні критерії вибору включають необхідність підтримання стану застосунку, кількість користувачів і вузлів, що записують дані, і рівень довіри між ними. Як правило, використання блокчейна не рекомендується, якщо немає необхідності зберігати поточний стан системи або якщо є тільки один записуючий вузол.

У таких сценаріях прості бази даних працюють набагато ефективніше і не страждають від проблем, пов'язаних з незмінністю інформації, вже записаної в блокчейн. Однак якщо в системі працює велика кількість користувачів, які постійно створюють записи, блокчейн може допомогти забезпечити цілісність даних без необхідності залучення довіреної централізованої сторони, що має перебувати в режимі онлайн, аби стежити за точністю інформації, що вводиться. У цьому випадку децентралізована структура блокчейна гарантує підвищену відмовостійкість. Проте можливість відмови від блокчейна враховується, якщо всі учасники довіряють один одному і мають встановлені механізми контролю, навіть якщо кількість вузлів велика.

Перенісши цей підхід до вибору механізму в сферу систем виявлення вторгнень, ми отримаємо логіку дій, представлену у вигляді блок-схеми на рисунку 2.3.

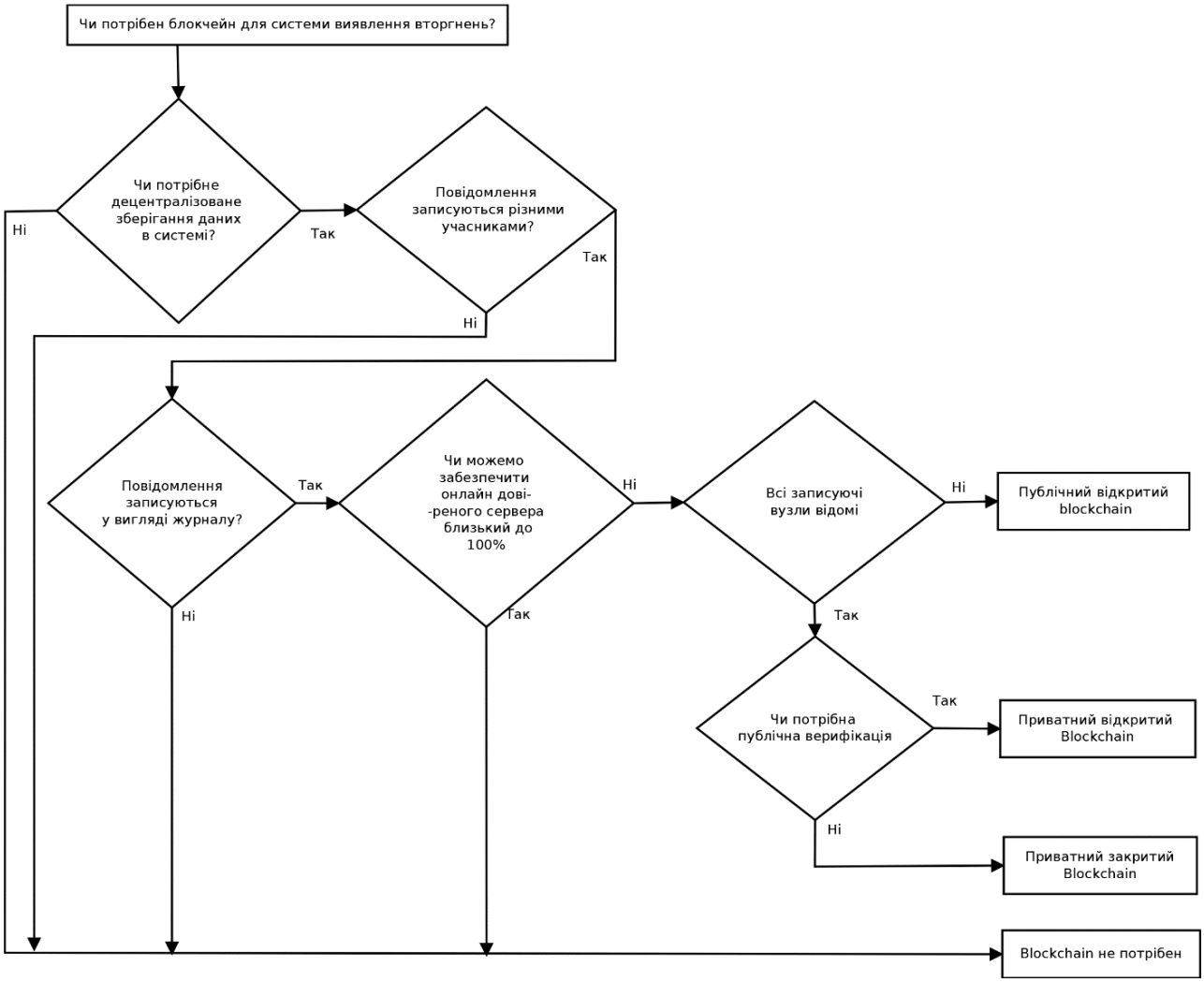


Рисунок 2.3 – Алгоритм застосування блокчейна для комп’ютерної системи виявлення загроз

Якщо склад вторгнення невідомий або може сильно змінитися, то блокчейн без контролю доступу - хороший варіант. У контексті використання блокчейну для децентралізованого виявлення мережевих вторгнень можна виокремити два підходи: перший стосується приватних мережевих вторгнень, які діють у межах однієї компанії чи між кількома компаніями, що мають взаємну довіру та можуть обмінюватися певною конфіденційною інформацією.

Другий стосується приватних систем виявлення мережевих вторгнень, що діють у межах однієї компанії або між кількома компаніями, які мають взаємну довіру і можуть обмінюватися певною конфіденційною інформацією. Блокчейн з контролем доступу найкраще підходить для цих цілей. Другий випадок

стосується використання систем виявлення вторгнень у децентралізованих, відкритих системах, у яких може брати участь будь-хто. У цьому випадку оптимальним є блокчейн без контролю доступу.

Інтеграція системи виявлення вторгнень (IDS) ґрунтується на використанні значного обсягу інформації для забезпечення ефективності її роботи. Насамперед, критично важливим є набір правил, який служить основою для швидкої обробки мережевого трафіку. Така база даних має тенденцію до постійного зростання, що робить застосування блокчейн-технології для зберігання та розповсюдження цього набору правил цілком обґрунтованим рішенням. Якщо певне правило втрачає актуальність, його можна позначити як застаріле, що дозволяє уникати непотрібного навантаження на систему.

Другим видом інформації в контексті розподілених IDS є журнал подій. Ці журнали також зазнають експоненційного зростання. Як і у випадку з правилами, їх можна поширювати через блокчейн. Проте якщо набір правил (наприклад, набір SNORT) є відносно невеликим за обсягом (менше 2 мегабайт) і достатньо компактним для інтеграції в блокчейн-інфраструктуру, то журнали подій становлять принципово іншу проблему. Їхній обсяг збільшується дуже швидко, що може спричинити значне перевантаження блокчейну. Крім того, більшість інформації у журналах подій має тимчасовий характер, і довгострокове зберігання таких даних може бути непрактичним, особливо для окремих видів пристроїв. Наприклад, одноплатні вузли (single-board nodes) часто обмежені в ресурсах пам’яті і не здатні забезпечувати збереження великих обсягів даних.

Зважаючи на ці обмеження, виникає потреба оптимізувати зберігання тимчасових даних. Це передбачає можливість вузлів визначити, чи доцільно зберігати конкретну інформацію протягом тривалого періоду. Найпростішим рішенням є перенесення даних за межі блокчейн-структури з подальшим застосуванням методу перевіреного володіння даними (Proof of Data Possession, PDP) для забезпечення цілісності. Метод PDP передбачає спочатку кодування даних M у нову форму M', де кожен блок даних mi в M' складається із s сегментів:

mi = (mi,1, mi,2,..., mi,s). Метадані σi обчислюються для кожного блоку mi і структуруються відповідно до визначеної схеми:

$$\sigma i = \left(H(name||i) \times \prod_{j=1}^{s} u_j^{m_{i,j}} \right)^{\alpha}$$

де α ключ доступу користувача uj ($1 \leq j \leq s$) обраний з масиву G.

Механізм PDP здебільшого використовується в мережах типу P2P для перевірки цілісності даних. Початковий метод PDP базувався на ймовірнісному підході до завершення перевірки, застосовуючи гомоморфні властивості RSA-підпису для агрегації доказів у компактну величину. Це дозволило суттєво знизити витрати на виконання протоколу. Водночас сучасні реалізації PDP переважно використовують хеш-системи [17].

В умовах використання блокчейн-технологій у розподілених системах виявлення вторгнень виникає необхідність обробки значного обсягу тимчасових даних, актуальних лише протягом певного періоду. Найбільш оптимальним підходом у такій ситуації є розподіл даних на дві категорії: постійні, які зберігаються у формі транзакцій всередині блоків, і тимчасові зовнішні, що зберігаються окремо від блоків, але криптографічно поєднані з ними.

Для реалізації цього підходу система хеш-систем у запропонованій блокчейн-архітектурі розділена на два компоненти. Перше дерево відповідає за верифікацію цілісності зовнішніх даних, які зберігаються поза межами блоку, а друге — за збереження хешів транзакцій, що безпосередньо входять до блоку. У заголовок кожного блоку включається загальний хеш, який формується шляхом комбінування кореневих хешів обох систем. Структуру модифікованих блоків із системою Merkle для роботи з тимчасовими даними ілюструє рисунок 2.4.

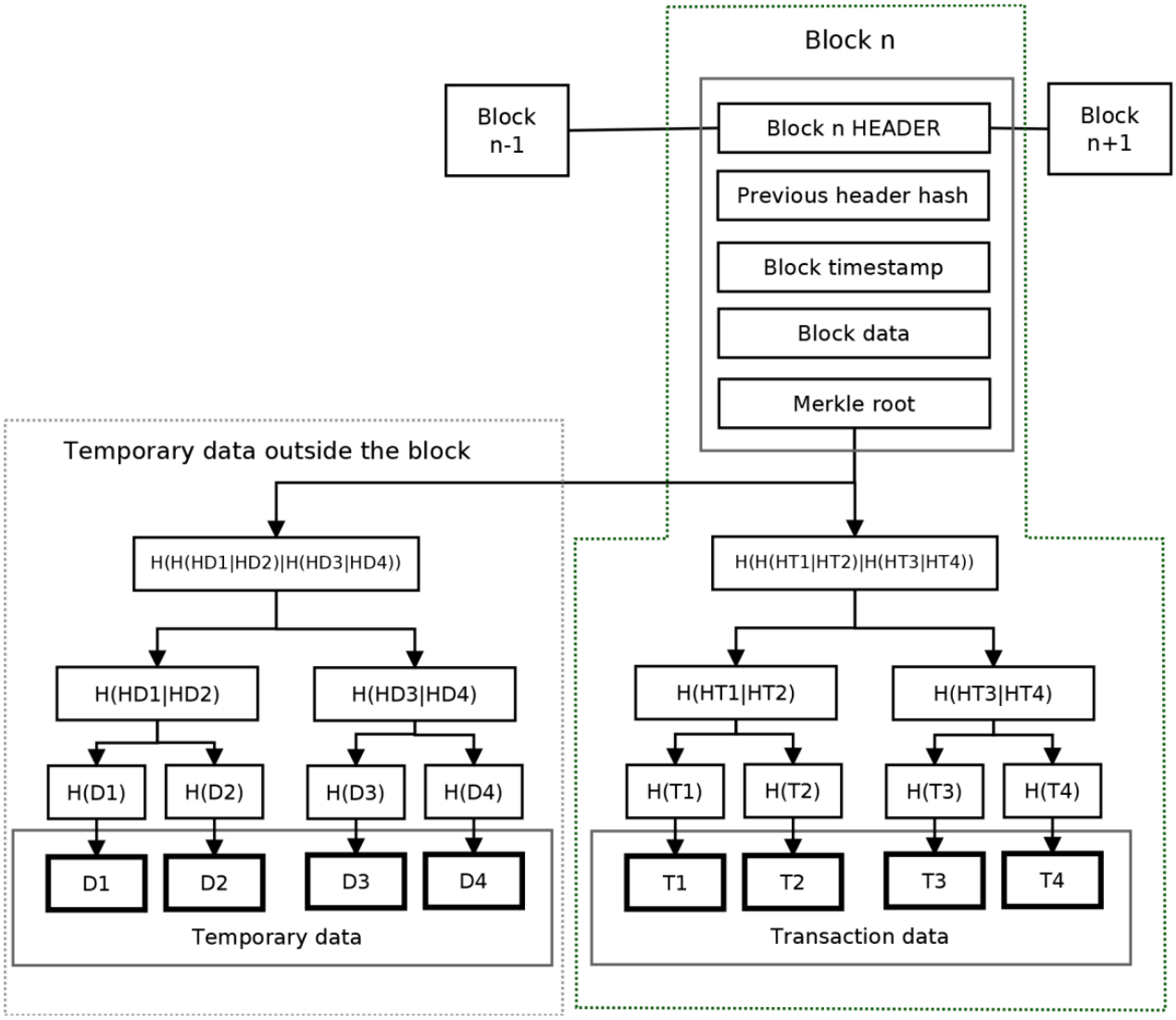


Рисунок 2.4 – Схема блока в блокчейні системи Merkle

Хеш-система Merkle являє собою структуру, де кожен нелистковий вузол визначається хешем його дочірніх вузлів, а листкові вузли містять хеші окремих блоків даних. У верхній частині дерева розташовується кореневий вузол, який об'єднує всі нижчі рівні. Одним із найпоширеніших хеш-алгоритмів для побудови дерева Merkle є Tiger. Цей алгоритм добре оптимізований для 64-розрядних процесорів і не має вразливостей, якими характеризується, наприклад, MD5 [16].

У розподілених системах виявлення вторгнень використання дерева Меркла має численні переваги. Ця структура забезпечує простоту реалізації та дозволяє поступово видаляти зайві дані з блоків, залишаючи лише їхні хеші. Окрім того, схема Merkle підтримує широкий спектр алгоритмів хешування, що

важливо для блокчейн-підсистем. Це дозволяє уникнути надмірного використання додаткових функцій і використовувати ту ж хеш-функцію, яка застосовується для хешування заголовків блоків. Завдяки цій особливості зберігається оптимальна продуктивність і знижується складність системи.

Система Merkle також спрощує пошук даних, збережених у вузлах мережі. Кожен вузол може швидко надавати доступ до своїх кореневих хешів, що дозволяє іншим вузлам перевіряти, чи містяться потрібні їм дані на цьому вузлі. Враховуючи такий підхід, тимчасове зберігання журналу подій може спричинити часткову втрату старих записів через певний час. Однак це навряд чи становитиме серйозну проблему, оскільки старі журнали поступово втрачають актуальність і рідко використовуються для класифікації подій. Аналіз великого обсягу даних із застарілих журналів не лише вимагає значних обчислювальних ресурсів, а й може знижувати точність класифікації через застарілість інформації.

Базова структура блоку, яка відповідає потребам розподіленої системи виявлення вторгнень, представлена на схемі 2.5. Основним елементом блоку є заголовок, що містить ключову інформацію: розмір блоку, його порядковий номер у ланцюжку, хеш дерева з даними всередині блоку, хеш дерева із зовнішніми даними, хеш заголовка попереднього блоку, мітку часу створення та версію протоколу консенсусу. Збереження версії протоколу є важливим для забезпечення сумісності зі старими правилами валідації у разі оновлення протоколу.

Криптографічна стійкість блоків до маніпуляцій обумовлена зв’язком між ними за допомогою хешів заголовків попередніх блоків. Перший блок у ланцюжку (генезис-блок) не має попередника, але кожен наступний блок посилається на свого попередника. Таким чином, будь-яка зміна блоку, що знаходиться в середині ланцюжка, потребує модифікації всіх наступних блоків. Це робить зміну даних технічно складною, а при достатньо високій потужності мережі — практично неможливою. У блокчейн-системах кількість блоків, які було створено після певного блоку, називається кількістю підтверджень цього

блоку. Для кожної системи встановлюється мінімальна кількість підтверджень, після досягнення якої дані вважаються незмінними та записаними назавжди.

2.3 Оцінювання ефективності застосування технології блокчейн в комп'ютерних системах і мережах

Блокчейн дозволяє вирішувати проблему довіри у розподілених системах, але при цьому слід враховувати його вплив на швидкість обміну інформацією. Саме тому процес оцінки ефективності використання апаратних ресурсів блокчейн-підсистемою з урахуванням різних параметрів мережі є критично важливим під час проєктування розподілених систем виявлення загроз.

Перед створенням моделі блокчейн-підсистеми необхідно розібратися у механізмі обробки транзакцій блокчейном. Транзакція у даному контексті — це атомарна одиниця інформації, що передається між вузлами і записується до блокчейну. Відповідно до правил більшості блокчейн-мереж, блоки генеруються із визначеною частотою. Через це транзакції не можуть бути негайно записані до блоку.

На початковому етапі створена транзакція зберігається вузлом у блоці пам'яті та передається сусіднім вузлам. При отриманні транзакції ці вузли також зберігають її у тимчасовому пулі пам'яті і далі транслюють іншим вузлам, з якими мають встановлені зв'язки. Таким чином, транзакція поступово поширюється по всій мережі.

Коли генерується новий блок, вузол, який його створює, записує транзакції з тимчасового пулу пам'яті до блоку. Потім цей блок передається сусіднім вузлам для подальшої перевірки. Після отримання нового блоку кожен вузол перевіряє його відповідність правилам протоколу консенсусу та коректність усіх записаних у блок транзакцій. Якщо перевірка успішна, вузол оновлює свій ланцюг блоків, видаляє з пулу пам'яті транзакції, що увійшли до блоку, і передає його іншим вузлам.

Основною проблемою блокчейну є обмежена кількість транзакцій, які можна записати до одного блоку. Через це здатність системи приймати й обробляти транзакції залежить від двох ключових чинників: максимальної кількості транзакцій у блоці та періодичності генерації блоків. Якщо середня кількість транзакцій за час створення блоку перевищує максимальну пропускну здатність системи, тимчасовий пул пам’яті постійно переповнюватиметься. Це може спричинити аварійне завершення роботи системи через нестачу пам’яті або втрату частини транзакцій.

Для оцінки продуктивності блокчейну була розроблена імітаційна модель на основі агентного підходу. Такий підхід обрано, оскільки блокчейн-системи зазвичай складаються з великої кількості автономних вузлів, що працюють незалежно, утворюючи єдиний розподілений механізм. У цій моделі кожен вузол представлено окремим агентом. Агенти взаємодіють між собою через однорангову мережу: більшість вузлів підключені лише до обмеженої кількості сусідніх вузлів (наприклад, у Bitcoin кожен вузол підтримує 8 вихідних та до 125 загальних підключень).

Для створення імітаційної моделі використано систему AnyLogic, що має значні можливості для симуляцій різноманітних типів систем. У моделюванні стратегії підключення агентів, яка відповідає блокчейн-мережам, застосовано стандартний тип мережі "small world". Цей тип характеризується тим, що агенти з’єднані із сусідами по кільцю, але також мають додаткові зв’язки з віддаленими вузлами для прискорення розповсюдження повідомлень мережею.

У представленій роботі пропонується спрощена модель блокчейн-системи, головною метою якої є оцінювання її швидкодії. Функціонування блокчейну аналізується у двох сценаріях: у нормальному стані системи та у стані, що піддається атакувальному навантаженню. Основний параметр для оцінювання — це частота генерування транзакцій за одиницю часу.

Для моделювання роботи системи в звичайному режимі було встановлено частоту створення сповіщень на рівні одного сповіщення кожні 33 секунди. Це вибір обґрунтований тим, що переважна більшість систем з відкритим доступом

до Інтернету та загальнодоступними IP-адресами регулярно піддаються незначним атакам, таким як сканування портів чи випадкові брутфорс атаки, спрямовані на злам стандартних паролів. Хоча такі загрози є рідкісними, цей показник, ймовірно, наближено характеризує реальну частоту шкідливих дій у мережевій системі під час її нормальної роботи.

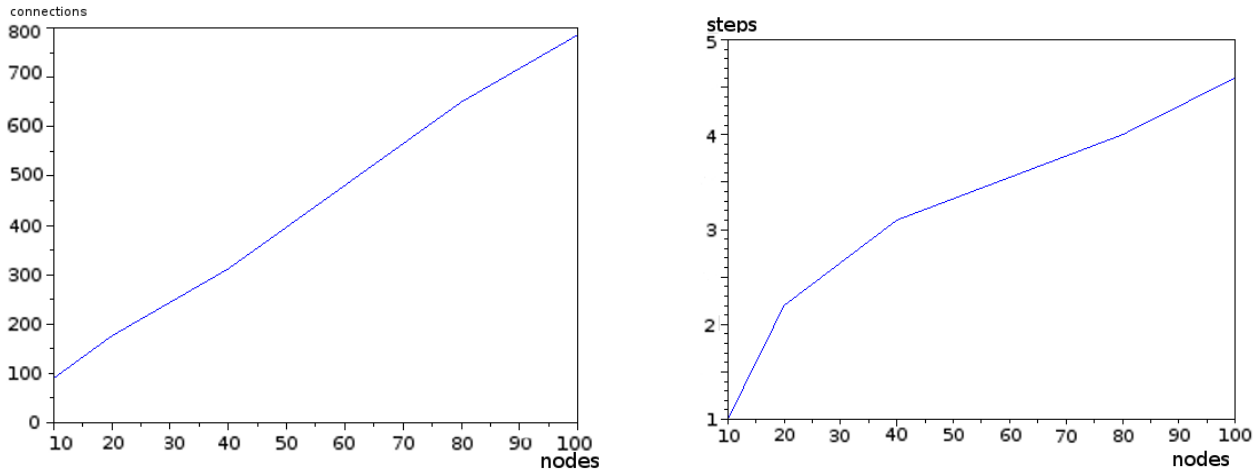
У моделі стану системи під атакою частота створення транзакцій підвищується до одного запису на секунду. Це спрощене припущення може змінюватися залежно від типу та масштабу атаки. Однак для запобігання перевантаженню системи транзакції не створюються для кожного окремого шкідливого пакета. У разі виявлення великої кількості таких подій сповіщення консолідуються: декілька близьких за типом подій, що надійшли в короткий часовий проміжок, групуються в одне повідомлення. Таким чином, використовується зазначена стратегія для більш ефективного представлення стану системи під час атаки.

Генерація сповіщень у моделі відбувається випадковим чином на мережевих вузлах з використанням експоненційного розподілу часу між подіями. Цей розподіл традиційно застосовується для моделювання випадкових вхідних дзвінків чи повідомлень, а в нашому випадку – сповіщень системи виявлення вторгнень. Створені сповіщення додаються до черги тимчасового пулу пам’яті кожного вузла, після чого поширюються на всі підключені вузли. Кожен вузол дублює отримані повідомлення до власної черги пулу пам’яті та розсилає їх своїм сусідам, імітуючи інформаційний потік у типовій блокчейн-мережі.

Окремим важливим аспектом моделі є симуляція процесу генерації блоків. У роботі використано поширений в більшості блокчейн-схем підхід – блоки створюються з фіксованим таймаутом. Простота моделі передбачає рівні ймовірності для всіх вузлів щодо генерації нового блоку, що забезпечує нейтральність у розподілі навантаження між вузлами. Вузол, обраний для генерації блоку, додає до нього певну кількість транзакцій із пулу пам’яті, але не більше встановленого максимального розміру блоку.

На завершення розглянемо ключові фактори, що здатні впливати на продуктивність розподілених блокчейн-мереж. Першим серед них є кількість вузлів, оскільки цей параметр безпосередньо визначає кількість з’єднань у мережі, а також кількість ітерацій (кроків), необхідних для повного розповсюдження транзакцій. При цьому крок визначається як процес отримання вузлом нового повідомлення із подальшою передачею цього повідомлення суміжним вузлам.

Застосування імітаційної моделі демонструє, що за умов незмінної кількості зв’язків на один вузол спостерігається лінійна залежність (рисунок 2.5 (а))



(а) (б)

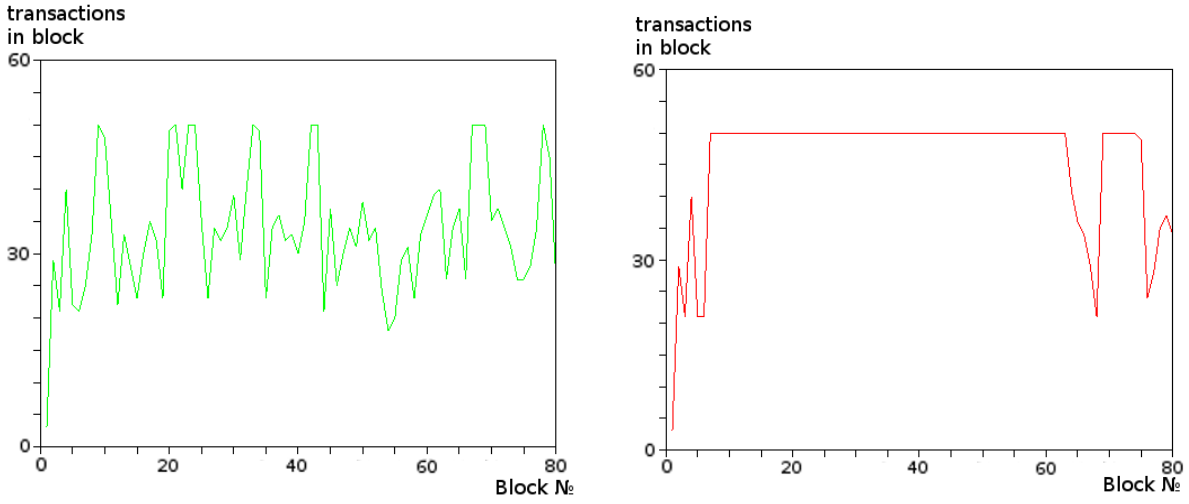
Рисунок 2.5 – Графіки залежності кількості з'єднань (а) та кроків повного доступу (b) від чисельності вузлів

Кількість з'єднань значною мірою впливає на продуктивність у централізованих системах, оскільки центральні вузли змушені обробляти всі з'єднання одночасно. Натомість у децентралізованій мережі кожен вузол має обмежену кількість з'єднань, що значно змінює підхід до обміну даними. Замість прямої передачі повідомлення всім вузлам мережі, інформація кілька разів передається й отримується сусідніми вузлами, доки вона не дійде до кінцевого пункту призначення та не розповсюдиться по всій мережі. У такій ситуації ключову роль починає відігравати кількість необхідних кроків, оскільки цей

показник пропорційний часу, за який повідомлення охопить всю мережу. Тож далі розглянемо, як залежить кількість кроків поширення повідомлень від загальної кількості вузлів у мережі (див. Рисунок 2.5 (б)).

Кількість кроків у цьому випадку демонструє логарифмічну залежність від числа вузлів, що дозволяє виконувати всього близько шести кроків для тисячі вузлів. Завдяки цьому такий тип децентралізованої мережі добре підходить для впровадження у великомасштабних системах без необхідності в потужних центральних вузлах, забезпечуючи швидкий обмін повідомленнями.

Однак слабким місцем блокчейн-систем залишається процес запису транзакцій (у нашій ситуації це дані про підозрілі події) у блоки. Особливу увагу потрібно приділити збереженню балансу між заповненням блоків і резервуванням ресурсів на випадок можливих атак або аномальних станів мережі.



(а) (б)

Рисунок 2.6 – заповнення компонентів блокчейну для вузлів у стані пошуку (а) та у стані вторгнення (б)

При фіксованому розмірі компонента збільшення об'єму блоку призводить до стрімкого зростання розміру блокчейну. Крім того, великі блоки важче передавати безперервно, оскільки існує ризик розривів з’єднання, а їх перевірка та обробка вимагають значного часу. Хоча обсяг дискового простору сьогодні вже не є критичним завдяки наявності жорстких дисків із місткістю понад 10 ТБ,

основною проблемою залишається поширення та перевірка блоків. Це особливо важливо, оскільки всі вузли повинні синхронно прийняти новий блок до початку генерації наступного.

Саме тому більшість криптовалют обирають менший розмір блоку для своїх блокчейнів. Наприклад, на поточний момент Bitcoin має теоретичний максимальний розмір блоку в 4 мегабайти, а нові блоки створюються кожні 10 хвилин. Такий блок може вмістити приблизно 8000 транзакцій обсягом 512 байт кожна, що еквівалентно приблизно 13 транзакціям на секунду. Ці характеристики є прийнятними для криптовалютних операцій, проте для розподіленої системи виявлення вторгнень блокчейн із такими властивостями не забезпечить необхідної швидкодії, що значно обмежує його ефективність у таких сценаріях.

Криптовалюти вирізняються тим, що зазвичай демонструють стабільний рівень транзакцій. Однак у контексті системи виявлення вторгнень ситуація змінюється: більшість блоків залишаються майже порожніми, але в разі атаки на один чи кілька вузлів вони різко наповнюються чисельними транзакціями.

У нашій моделі ми вирішили визначати блоки не за кількістю інформації, а за кількістю транзакцій, спрощуючи підхід і припускаючи, що всі транзакції мають однаковий розмір. Виходячи з цього, будемо аналізувати заповненість блоків та використання пам’яті пулу при змінній кількості вузлів. Для початку розглянемо сценарій із 20 вузлами (наведено на рисунку 2.7).

Розмір блоку, який ми обрали для цієї конфігурації (50 транзакцій у блоці та швидкість створення одного блоку за хвилину), доволі скромний для ситуації атаки, але наближається до граничного показника для нормального режиму з деякими спорадичними аномаліями. На основі отриманих графіків видно, що під час атак блоки повністю переповнюються транзакціями.

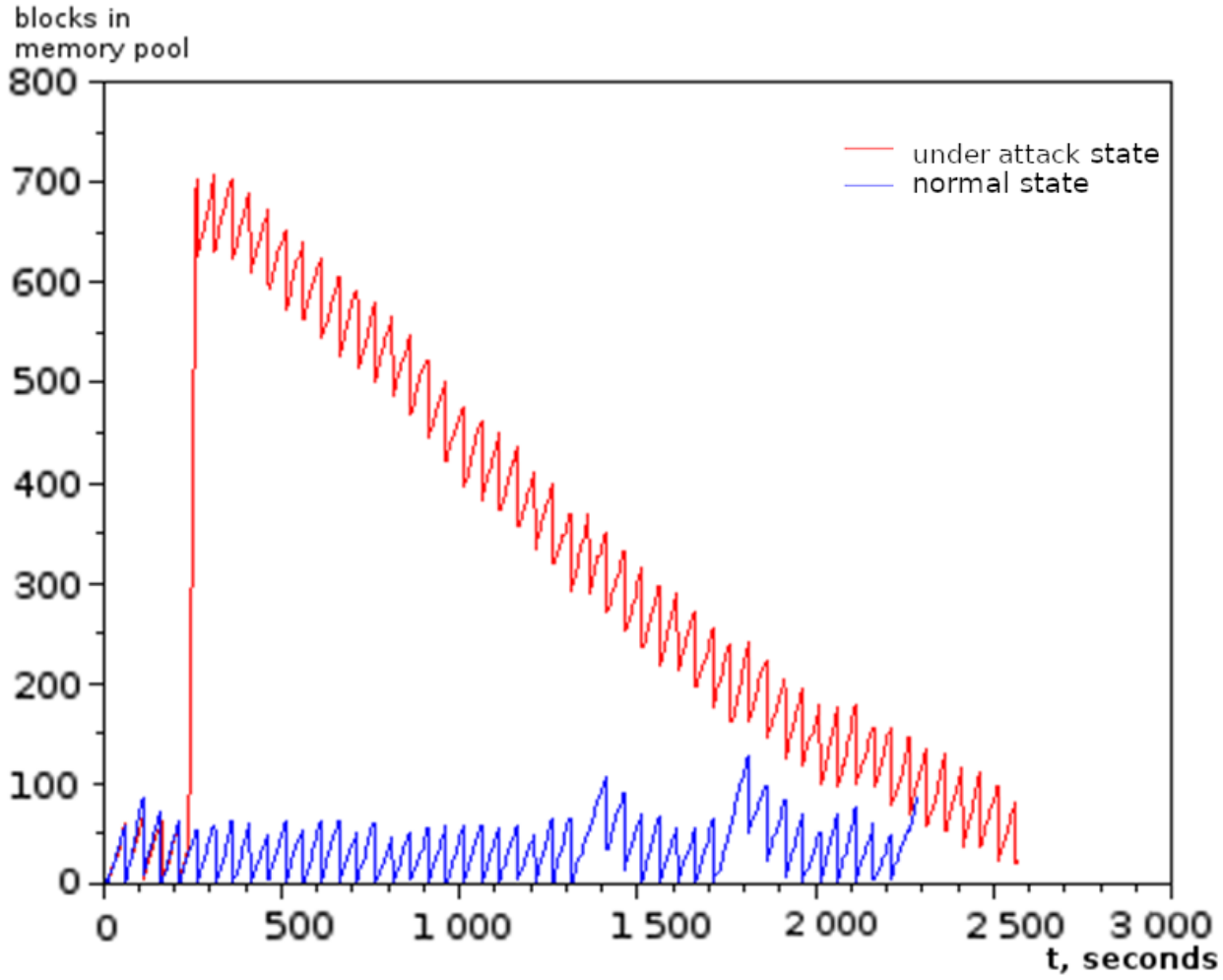


Рисунок 2.7 – Завантаженість блоків пам'яті при структурі системи у 20 вузлів блокчейну

У даній ситуації можна спостерігати аналогічну проблему: навіть для короткотривалих вторгнень вимагається значний обсяг часу для обробки всіх транзакцій (для описаних сценаріїв знадобиться близько 45 вузлів).

Такий стан справ є суттєво негативним для системи, що функціонує на основі блокчейн-технології, оскільки велика кількість неопрацьованих транзакцій у мемпулі призводить до дестабілізації системи. Більше того, подібна поведінка тимчасового пулу транзакцій сприяє збільшенню вразливості системи до тривалих атак. Це виникає через можливість переповнення мемпулу, що, у свою чергу, може призвести до аварійного завершення роботи вузла.

2.4 Структура системи захисту розподіленої компютерної мережі на основі технології блокчейн

Для досягнення мети дослідження пропонується архітектура розподіленого вузла системи виявлення загроз, що інтегрує технологію блокчейн. Ця структура зображена у вигляді блок схеми (рис. 2.7) і складається з п'яти основних компонентів, що забезпечують безпеку мережі:

- модуль мережевого фільтра: відповідає за проактивну фільтрацію трафіку та фільтрацію протоколів, які не використовуються в конкретному сегменті мережі.

- модулі виявлення загроз: працюють на основі складнішої системи правил, ніж між мережеві фільтри. Це дає змогу виявляти ширший спектр атак.

- модуль розпізнавання підозрілих звернень: використовує регресійні класифікатори та алгоритми на основі нейронних мереж для аналізу підозрілої активності. Цей модуль працює на основі даних, отриманих від модулів виявлення вторгнень і між мережевих фільтрів, а також інших вузлів системи безпеки розподіленої комп'ютерної мережі.

Генератор правил, спираючись на результати класифікації підозрілих подій, виконує функцію створення тимчасових правил, які надають можливість блокувати аналогічні атаки без необхідності постійної класифікації. Утворені правила направляються для обробки мережевим екраном та модулем виявлення вторгнень, а також розповсюджуються між вузлами захисної системи комп'ютерних мереж за допомогою блокчейн-технології.

Модуль блокчейн-підсистеми забезпечує передачу як згенерованих правил, так і інформації про виявлені аномальні події між вузлами системи. До його складу входять структурні елементи блокчейну та ключові компоненти для забезпечення функціонування: генератор блоків, механізм верифікації, а також мережевий елемент, який об'єднує блокчейн-модулі в єдину децентралізовану мережу. Цей модуль виконує роль основного посередника між вузлами розподіленої системи захисту комп'ютерних мереж.

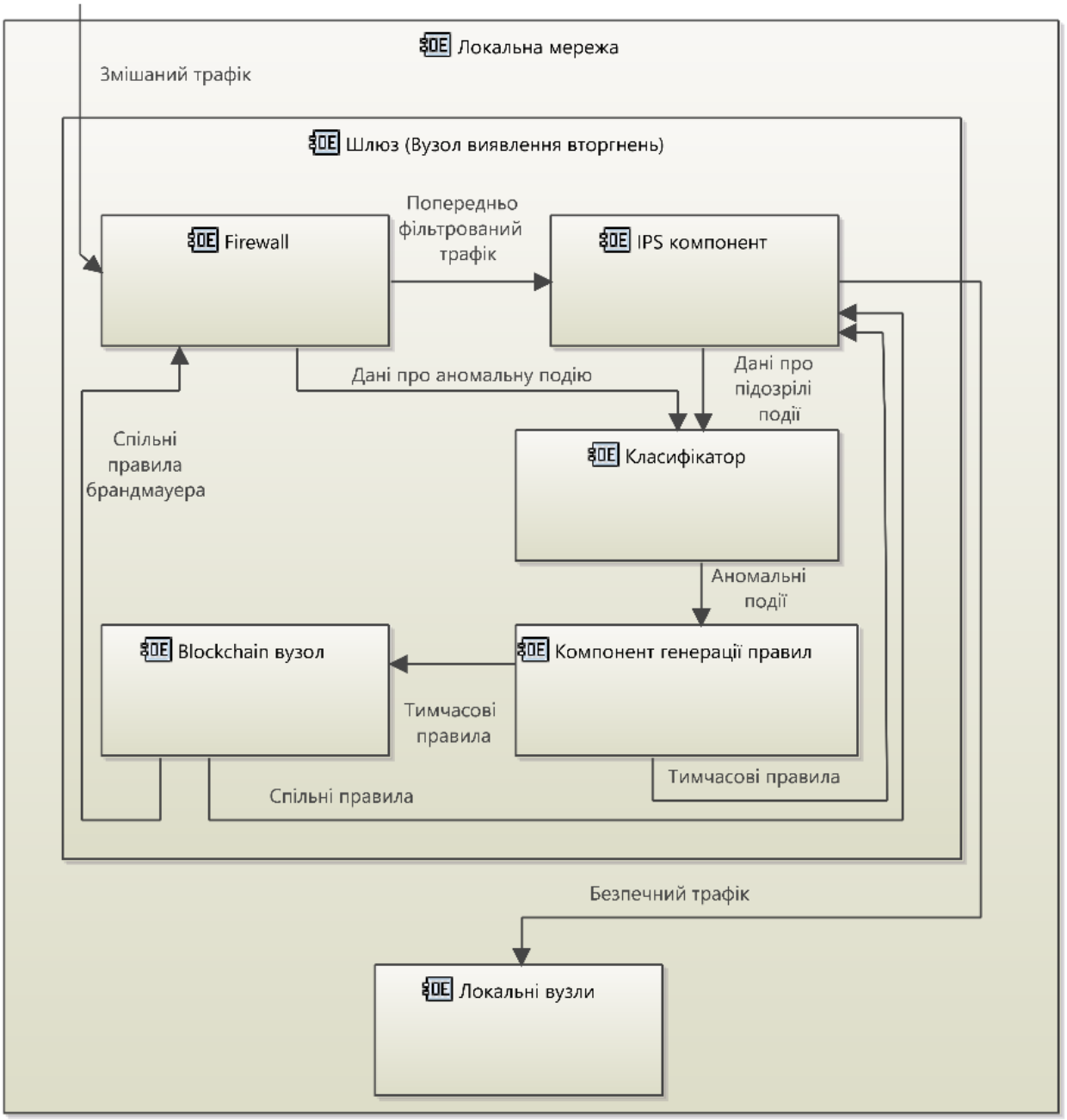


Рисунок 2.8 – Блок-схема архітектура розподіленого вузла системи виявлення загроз, що інтегрує технологію блокчейн

Детальне дослідження структури блокчейн-модуля демонструє, що для його повноцінної роботи необхідно залучати чотири базові компоненти. Водночас у спрощеному режимі — лише зчитування без створення нових блоків — достатньо двох компонентів. На перший погляд, таке рішення здається недоцільним; однак, враховуючи ресурсозатрати на створення нових блоків, для вузлів із обмеженою продуктивністю процесорів ця модель є оптимальною,

оскільки забезпечує ефективність функціонування шлюзу в рамках захисної інфраструктури.

Повна структура блокчейн-компонентів представлена на рисунку 2.9 і включає такі елементи:

1. Базовий блокчейн-компонент, який відповідає за підтримку зв’язку з іншими вузлами мережі блокчейну. Він забезпечує прийом і передачу як блоків, так і транзакцій, що є критично необхідним для функціонування підсистеми.

2. Компонент валідатора, завданням якого є перевірка достовірності отриманих блоків. Його впровадження є обов’язковим, оскільки використання неперевірених блоків пов’язане з високим ризиком уведення підроблених даних. Більше того, створення нових блоків без залучення механізму валідації неможливе через потенційну загрозу цілісності інформації.

3. Локальне сховище блокчейна - це ланцюжок блоків, який виступає як додатковий компонент інфраструктури блокчейна, забезпечуючи зберігання локальних копій даних блокчейна. Наявність цього елемента допомагає підвищити надійність зберігання даних у мережі блокчейн. Однак вузли з обмеженим обсягом пам'яті можуть відмовитися від використання цього компонента для оптимізації системних ресурсів.

4. Компонент майнінгу відіграє важливу роль у створенні нових блоків відповідно до затвердженого протоколу консенсусу. Цей модуль не є обов'язковим, але його інтеграція в кожен вузол бажана з погляду забезпечення відмовостійкості розподілених систем, призначених для захисту комп'ютерних мереж. Наявність майнінгового компонента на кожному вузлі підвищує загальну надійність системи і запобігає можливості злому і підміни блоків.

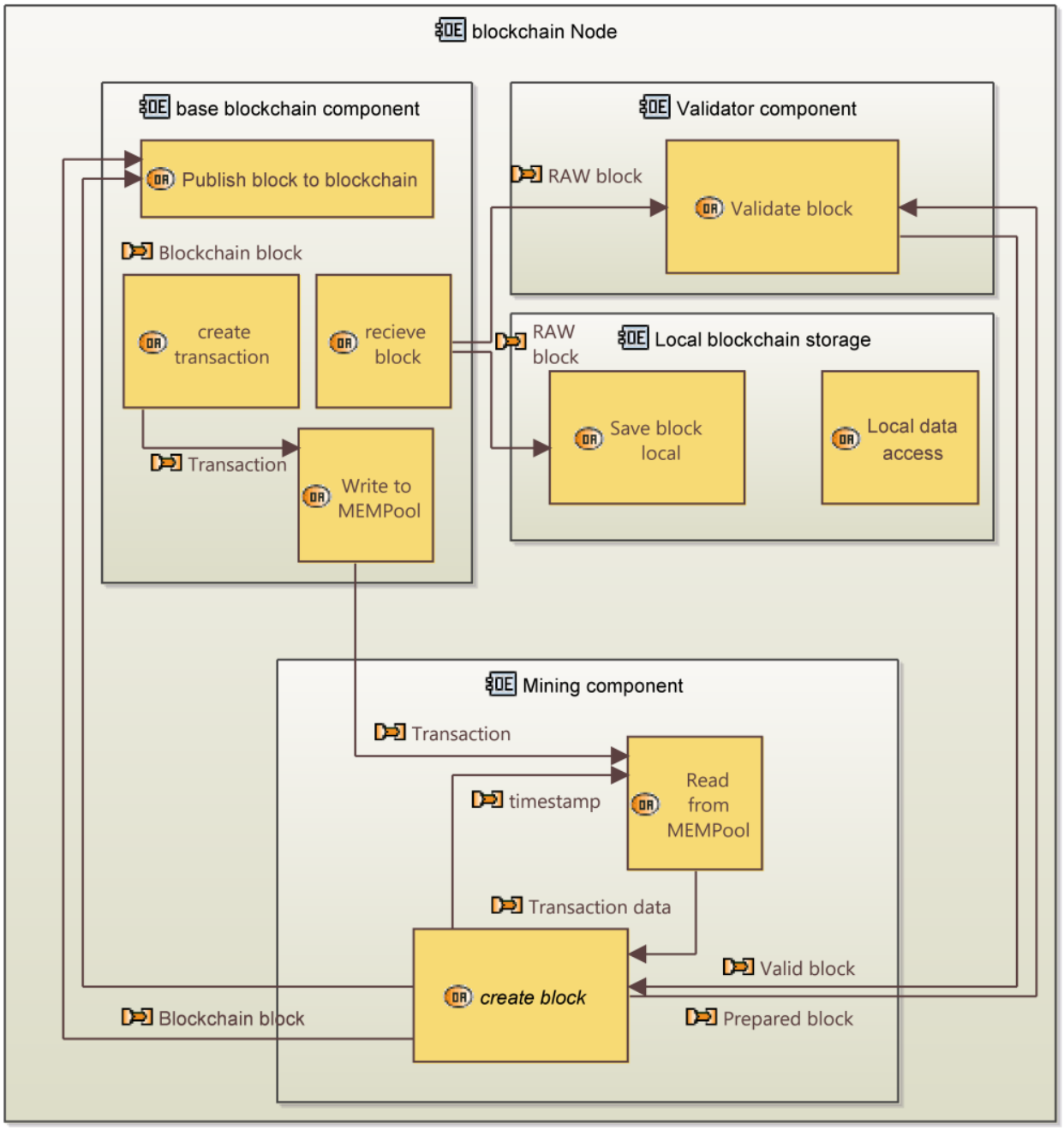


Рисунок 2.9 – Схема компонентів блокчейн розподіленої системи захисту комп’ютерних мереж

У системі безпеки децентралізованої комп'ютерної мережі блокчейн є важливим компонентом, який виконує роль сховища важливих даних, які необхідно зберігати і розподіляти в часі між вузлами системи виявлення вторгнень. Логічна структура блокчейн-компонента показана на рисунку 2.10. У цій структурі весь блокчейн складається з ланцюжка взаємопов'язаних блоків, починаючи з вихідного блоку (блок генезису). Блок генезису заздалегідь

визначений в програмному забезпеченні і вважається автоматично підтвердженим. Від цього блоку вузли системи в свою чергу починають будувати ланцюжок блоків, наповнених необхідними даними.

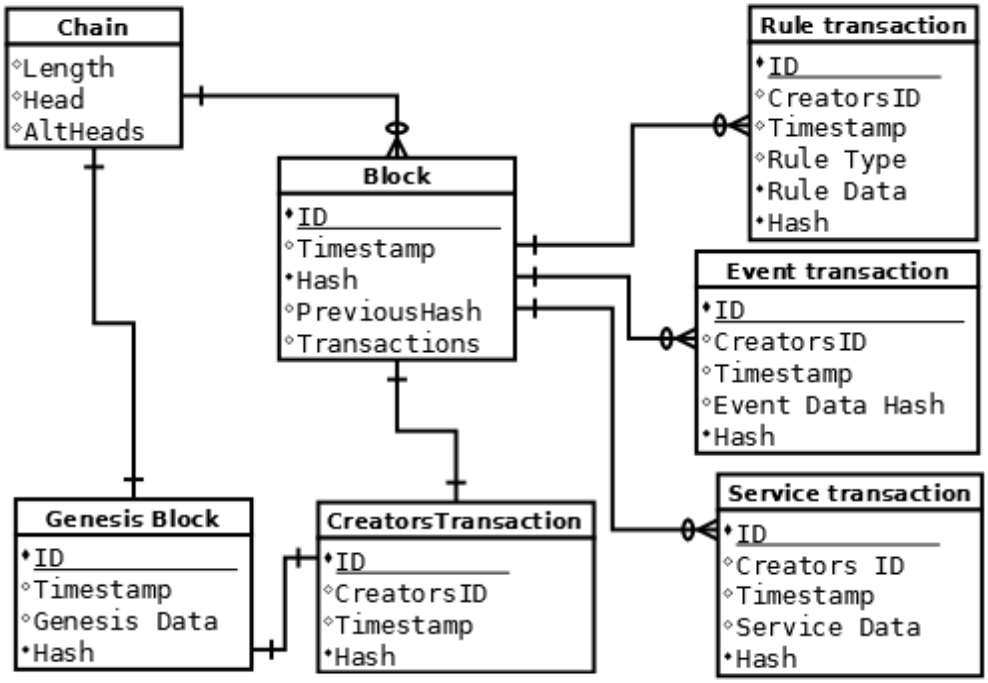


Рисунок 2.10 – Структура накопичення даних компонента блокчейн

Кожен блок у системі розпочинається із транзакції автора, яка виконує кілька важливих функцій. По-перше, вона фіксує вузол, що згенерував цей блок. По-друге, в ній зазначається час створення блоку. Ця інформація використовується для визначення проміжку часу, який минув від моменту генерації останнього блоку для кожного вузла. Завдяки цьому забезпечується максимально рівномірний розподіл блоків між вузлами. Водночас враховуються лише транзакції, що належать до основного ланцюжка блоків, у який додається новий блок; можливі розгалуження до уваги не беруться.

Дані блоку надалі формуються з транзакцій, відібраних із тимчасового пулу. Проте кількість транзакцій, що включаються до блоку, суворо регламентується максимально дозволеним числом. Таке обмеження запобігає перевищенню встановленого розміру блоку, що сприяє стабільній передачі даних із мінімальною ймовірністю розриву з’єднання.

Окрім транзакцій автора, у блок включаються три основні типи транзакцій. Перший тип — правило. У таких транзакціях фіксуються правила захисту комп’ютерних мереж, сформовані системою, і ця інформація записується до блокчейну. Другий тип — події. Ці транзакції містять дані про виявлені та потенційно загрозливі події. Оскільки така інформація швидко втрачає актуальність і займає вагомий обсяг дискового простору, зберігати її повністю в блокчейні недоцільно. Замість цього основні дані щодо таких транзакцій зберігаються у зовнішніх файлах, а в самій транзакції фіксується лише хеш MerkleTree, що посилається на ці дані. Третій тип — службові транзакції. Їхня основна функція полягає в обміні між вузлами службовою інформацією, необхідною для стабільної роботи розподіленої системи захисту комп’ютерних мереж.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ МОДЕЛЬ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

3.1 Програмно-апаратна платформа для захисту інформації на основі технології блокчейн

Першим кроком у побудові розподіленої системи виявлення вторгнень для мережі SOHO або SMB є визначення апаратних і програмних платформ, які легко розробляти, впроваджувати і використовувати. Оскільки функціональність такої системи передбачає перехоплення і фільтрацію мережевих пакетів, операційна система повинна мати відповідний мережевий стек. Оскільки фільтрація мережевих пакетів вимагає значних обчислювальних ресурсів, в якості апаратної платформи було обрано платформу, сумісну з архітектурою X86_64. Цей вибір має ряд переваг, таких як широкий вибір операційних систем та апаратних компонентів, висока продуктивність та широка доступність на ринку

Платформа ARM також розглядалася як альтернатива, але наразі важко знайти рішення на базі ARM, яке б відповідало вимогам щодо обчислювальної потужності та підтримки декількох мережевих адаптерів. Тому в якості апаратного забезпечення було обрано платформу X86_64, яка дозволяє використовувати стандартне серверне та побутове обладнання в якості основи для мережевої фільтрації. З точки зору операційного програмного забезпечення, Unix-подібні системи, особливо FreeBSD, є найкращим вибором завдяки своєму потужному мережевому стеку, наявності всіх необхідних компонентів та багатому програмному середовищу. FreeBSD є основою системи OpnSense, і варто зазначити, що її можна використовувати на платформах без моніторів, оскільки вона надає не тільки необхідну функціональність, але й зручний веб-інтерфейс управління.

Для вузлів, орієнтованих на підтримку компонентів блокчейну, підходящим вибором буде Unix-сумісна система. Однак у цьому випадку такі сервери не виконують функції фільтрації пакетів та виявлення вторгнень, що зменшує навантаження та оптимізує продуктивність при обробці транзакцій блокчейну. Основною мовою програмування для розробки блокчейн-підсистеми було обрано Python 3. Ця мова була обрана завдяки високій кросплатформенній сумісності та великій кількості готових бібліотек, доступних для реалізації всіх необхідних функцій. Це забезпечує сумісність з широким спектром операційних систем та апаратних платформ, що є важливим для побудови гнучких та масштабованих систем.

3.2 Програмні засоби фільтрації загроз

У нашому випадку найкращим варіантом є використання вбудованих в систему OpnSense компонентів PacketFilter та Suricata для фільтрації мережевого трафіку та виявлення вторгнень. Перевагою використання вбудованих компонентів у нашому випадку є те, що вони добре інтегровані в систему і не потребують додаткового встановлення, що спрощує загальне розгортання системи. Також вбудовані компоненти мають зручний веб-інтерфейс конфігурації, що спрощує процес налаштування.

PacketFilter - це міжмережевий екран, який використовується в системах FreeBSD та системах на їх основі. Він має досить широкий набір функцій мережевої фільтрації, включаючи традиційну фільтрацію, прозору фільтрацію другого рівня, NAT, балансування каналів, а також гнучке налаштування правил, включаючи макроси і мітки трафіку. PacketFilter складається з власне фільтра і утиліти адміністрування, яка використовується для запуску і зупинки оновлень конфігурації, а також для зупинки і т.д. і складається з двох компонентів: власне фільтра і утиліти адміністрування. Фільтр працює на рівні ядра за допомогою системного виклику ioctl [12], тому утиліта адміністрування не є обов'язковою

для роботи фільтра, але без неї було б дуже важко керувати роботою фільтра. Досить цікавою особливістю PacketFilter є таблиця адрес, яка дає PacketFilter перевагу над конкуруючими продуктами з точки зору продуктивності та гнучкості конфігурації. Таблиця може зберігати IP-адреси (як версії 4, так і 6), а можливість позначати адреси як винятки дозволяє визначати таким чином дуже складні топології. У той же час, використання таблиць може значно прискорити пошук адрес у порівнянні з використанням традиційних правил, дозволяючи використовувати більше записів для фільтрації без втрати продуктивності. Крім того, таблиці можна використовувати для ведення статистики по кожній записаній адресі або для використання умовних записів, коли в таблицю вносяться адреси, що перевищують певні кількісні параметри трафіку. Таблиця також може автоматично видаляти записи, термін дії яких закінчився.

Деякі сторонні утиліти, що мають непряме відношення до брандмауерів, можуть керувати таблицею брандмауера за допомогою системних викликів, наприклад, DHCP-сервер, який записує активні, вільні та заборонені адреси в таблицю. Основним завданням брандмауерів у розподілених системах виявлення вторгнень є попередня фільтрація трафіку і трансляція адрес для підключення локальних мереж, захищених розподіленою системою, до глобальної мережі.

Використовуючи вбудовані компоненти Suricata, як основні компоненти для виявлення вторгнень; Suricata є одним з найпопулярніших рішень для виявлення вторгнень з відкритим вихідним кодом, і, на відміну від Snort, це рішення краще працює на багатопотокових системах.

Однією з основних відмінностей між IDS і брандмауерами є те, що брандмауери не мають доступу до вмісту пакетів і вся фільтрація здійснюється лише на рівні заголовків пакетів, тоді як IDS мають доступ до глибокого аналізу пакетів і системи правил IDS пропонують набагато більше Процес обробки мережевих пакетів в SURICATA можна розділити на наступні етапи перехоплення пакетів - на цьому етапі пакети перехоплюються на всіх інтерфейсах з відповідною конфігурацією. Декодування пакетів - на цьому етапі система намагається отримати доступ до внутрішнього вмісту пакету для

подальшої обробки. Виявлення на основі правил - на основі всіх доступних параметрів можна визначити відповідність пакетів правилам. Реакція - на цьому етапі система реагує на певні події відповідно до правил; оскільки набір правил Suricata є стандартним і сумісним зі Snort, відкритий набір правил спільноти Snort може бути використаний як базовий набір правил для попереднього виявлення відомих атак Це призводить до створення кращого базового набору правил для попереднього виявлення відомих атак. Крім того, OpnSense включає в себе веб-інтерфейс управління Suricata, який охоплює основні функції системи виявлення вторгнень і надає гнучкі можливості управління набором правил, включаючи автоматичне завантаження і оновлення правил з реєстру.

Ще однією важливою перевагою SURICATA є можливість зберігати журнали у форматі JSON. В іншому випадку для розбору файлу журналу подій довелося б спочатку створювати обробник, а за відсутності стандартизованого формату такий процес був би дуже складним. Обробник файлу журналу подій побудований таким чином, що файл періодично перевіряється на наявність нових подій і при додаванні нової події ініціюється механізм створення нової транзакції. Іншими словами, блокчейн-компонент відстежує події, які повинні бути зафіксовані в блокчейні згідно з існуючою політикою безпеки, і при виявленні такої події створюється нова транзакція, а система журналу подій дає можливість поширити інформацію про подію серед всіх вузлів, що беруть участь в розподіленій системі виявлення вторгнень.

У рамках впровадження і тестування системи було реалізовано такі класифікатори: логістична регресія, класифікатор опорних векторів і ансамблевий класифікатор AdaBoostClassifier. Класифікатори на основі штучних імунних систем, процес навчання яких дуже складний і має бути адаптований до кожної конкретної системи та її В даному дослідженні його не реалізували, тому що він має бути адаптований до кожної конкретної системи та способу її використання. Процес створення класифікатора можна розділити на кілька етапів:

1) Підготовка навчальних і тестових наборів даних і розробка модулів навчання та попереднього опрацювання.

2) Навчання класифікатора.

3) Тестування класифікатора.

4) Створення остаточного модуля класифікації.

На першому етапі підготовки та попереднього опрацювання даних наявні інструменти спершу слід використовувати для проведення детального аналізу даних. Корисними варіантами для обробки даних, отриманих від модуля виявлення вторгнень, є Python та Jupyter Notebook.

Таке поєднання мови програмування та середовища виконання дає змогу використовувати потужний набір бібліотек Python для аналізу даних, зберігаючи водночас можливість інтерактивного виконання коду в графічному інтерфейсі, створюючи кінцевий модуль класифікації без істотних змін в Одержаний код можна використовувати без не всіх полів у звіті системи виявлення, що мають числовий формат, тож їх необхідно перетворити в числовий формат. Наприклад, для даних у текстовому форматі можна використовувати маркування з числовими мітками. Перший крок - створення кореляційної матриці параметрів, виявлення впливу кожного параметра на інші та видалення тих, які негативно впливають на узагальнення. До таких параметрів належать час передачі, IP-адреса і порт. У контексті одного запису такі параметри не приносять користі, оскільки не допомагають виявити шкідливу активність. Для перетворення нечислових даних у числові можна використовувати метод get_dummies, якщо набір можливих значень для поля обмежений. Цей метод створює поле, що містить 0 або 1 для кожного зі значень, знайдених у наборі, вказуючи, присутнє або відсутнє таке значення в полі.

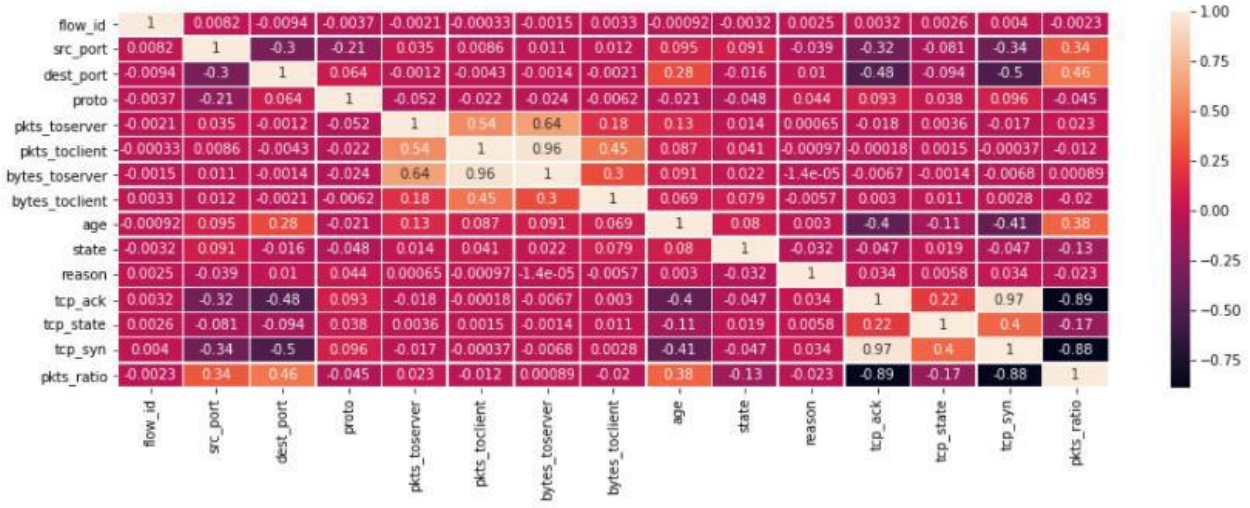


Рисунок 3.1 - Таблиця кореляції параметрів трафіку системи

Найкращим способом реалізації класифікатора на Python є використання бібліотеки Scikit-learn. Ця бібліотека містить набір реалізованих алгоритмів регресії та класифікаторів; Scikit-learn базується на бібліотеках NumPy та SciPy. Для реалізації логістичної регресії використовується клас LogisticRegression бібліотеки Scikit-learn.

Основними параметрами цього класу є multi_class - параметр для вибору типу класифікації «ovr» (один проти інших) або «multinomial» (багато проти багатьох), solver - алгоритм розв'язання оптимізаційної задачі, random_ state - ініціалізація генератора псевдовипадкових чисел для відтворюваності результатів та можливості оцінювання при підгонці моделі, n_jobs - задання кількості потоків, verbose - задання деталізації виведення інформації.

Код функції transaction Create FromLog

```
    def transactionCreateFromLog():
        if os.path.isfile("/var/log/suricata/eve.json"):
            file = open("/var/log/suricata/eve.json", "r")
            Block.threads_stopped[1]=1
            file.seek(0,2)
            while 1:
                where = file.tell()
                line = file.readline()
                if Block.app_stopped:
                    print("Transaction thread stopped")
                    Block.threads_stopped[1]=1
                    sys.exit(0)
                if not line:
                    time.sleep(1)
                    file.seek(where)
                else:
                    obj = json.loads(line)
                    t=Transaction(2, 222, int(time.time()))
                    t.gen(obj)
                    chain.addTransactionToPool(t)
                    for node in nodes:
                        try:
                            x=requests.post(node[1]+'transaction/',
data={'transaction':t.serialize()[1]}, timeout=2.5)
                        except:
                    transactions_self_to_csv(t)
        pass
```

Код класифікатора регресії

```
from sklearn.linear_model import LogisticRegression
clf   =   LogisticRegression(multi_class='multinomial',   solver='newton-cg',
random_state=42, n_jobs=-1, verbose=10)
clf.fit(X_train, y_train)
prediction = clf.predict(X_train)
accuracy_score(y_train, prediction)
```

Після створення класу необхідно провести його навчання на навчальній вибірці записів та оцінити точність моделі на тестовій вибірці. Такий метод навчання регресійного класифікатора дозволяє оцінити якість моделі на множині, відмінній від навчальної. Таким чином, він дозволяє уникнути перенавчання, коли модель добре працює на навчальній вибірці, але її точність різко падає на іншій вибірці.

Клас SVC використовується для реалізації класифікатора на основі опорних векторів. Деякі параметри в цьому класі відрізняються: tol - це параметр допуску, який зупиняє процес навчання, а C - параметр регуляризації. Навчання в цьому класі подібне до логістичної регресії. На тестовому наборі досягнуто 91% точності, що краще, ніж у логістичної регресії.

Код класифікатора SVC

```
from sklearn.svm import SVC
rf = SVC(random_state=42, tol=1e-4, verbose=10, C=1.5)
rf.fit(X_train, y_train)
prediction = rf.predict(X_train)
accuracy_score(y_train, prediction)
```

Ми використовуємо AdaBoostClassifier як класифікатор системи. Його єдиним параметром є base_estimator, який вказує, на основі якої моделі побудовано ансамбль. У нашому випадку ми обираємо DecisionTreeClassifier, який є звичайним деревом рішень. Параметр cv задає розподіл наборів для перехресної перевірки, що покращує точність класифікації, а параметр scoring визначає стратегію оцінки ефективності перехресної перевірки. Використання AdaBoostClassifier дозволяє підвищити точність оцінки на тестовому наборі до 92%, що краще, ніж значення, отримане з попереднім класифікатором.

Код класифікатора AdaBoostClassifier

```
abc = AdaBoostClassifier(base_estimator=DecisionTreeClassifier())
parameters = {'base_estimator__max_depth':[i for i in range(12,25,3)],
              'base_estimator__criterion':['entropy'],
              'n_estimators':[25, 50, 100, 250],
              'learning_rate':[0.01, 0.1],
              }
CV_clf = GridSearchCV(abc, parameters, verbose=10, cv=3, scoring='f1_macro',
n_jobs=-1)
CV_clf.fit(X_train, y_train)
best_estimator = CV_clf.best_estimator_
```

```
prediction = best_estimator.predict(X_train)
accuracy_score(y_train, prediction)
```

Дані, отримані за допомогою трьох моделей, можна об'єднати, щоб отримати простий модуль для класифікації подій, виявлених IDS. Таким чином, можна зменшити обчислювальне навантаження на систему виявлення, відфільтрувавши випадкові події або передавши явно шкідливі події на модуль генерації правил. Оскільки основним завданням блокчейн-компонента є зберігання та розподіл даних, то розробка блокчейн-компонента починається з блокчейн-сховища. Бінарні файли, що містять необроблені дані, використовуються для зберігання основних даних блоків, що зберігаються в ланцюжку, а бази даних levelDB використовуються для індексації блоків у ланцюжку, що може прискорити пошук блоків у бінарних файлах. Це пов'язано з тим, що розмір двійкових файлів з часом зростає, і завдання пошуку в неіндексованих файлах стають складнішими. База даних блоків містить записи ключ і значення, де ключ - це хеш блоку, а значення - адреса зсуву від початку файлу. Якщо потрібно обмежити максимальний розмір файлу, наприклад, при використанні файлової системи FAT32, максимальний розмір файлу можна використовувати як модуль. Значення зміщення, отримане модулем, вказує на зміщення файлу, в який записується певний блок, а частка від ділення значення зміщення на максимальний розмір файлу вказує на номер файлу, в який записується цей блок. Цей принцип, який стосується зберігання, є стандартним для більшості реалізацій блокчейну і показав хорошу продуктивність. При розгортанні блокчейну немає необхідності передавати базу даних, оскільки вона створюється на вузлі під час процесу валідації блоків. Вимога валідації блоку при отриманні значно уповільнює початкове завантаження блокчейну, але з міркувань безпеки такий підхід використовується майже у всіх блокчейн-зв'язках, оскільки весь блокчейн завантажується тільки при ініціалізації, не втручаючись в роботу додатків. Маніпуляції з базою даних в нашій системі реалізовані класом DBConnector, який містить базові методи для підключення до бази даних і маніпуляцій з її записами.

Код функції makeBlockRecord

```
    def makeBlockRecord(self, headerHash, header, blockid, trcount, offset):
            self.blockDB.put(b'b'+headerHash,  header+int(blockid).to_bytes(4,
'little')+int(trcount).to_bytes(4,          'little')+int(offset).to_bytes(4,
'little'), sync=True)
    def readBlockRecord(self, headerHash):
            rec = self.blockDB.get(b'b'+headerHash)
            if rec is not None:
                return (rec[0:71], rec[71:75], rec[75:79], rec[79:83])
            else:
                return (b'', b'', b'', b'')
```

Для формування структури об'єкта блоку створюється клас блоку, який визначає всі основні поля блоку та методи створення, запису, читання, вилучення, перевірки та серіалізації блоку для передачі по мережі. При записі у файл дані записуються у двійковому форматі на основі фіксованих зсувів та розмірів блоків, що задаються у самому файлі, як описано вище. Однак двійковий формат без надлишковості не дозволяє перевіряти узгодженість, що ускладнює перевірку, і цей формат не підходить для передачі через мережу. Тому, щоб полегшити передачу і подальше відтворення переданих блоків в інших вузлах, було вирішено вибрати формат Yaml, в якому об'єкти можуть бути закодовані в зручний спосіб і передані в текстовому форматі, а налагодження також є досить зручним в такому типі передачі даних. Конструктор приймає в якості параметра тільки індекс блоку, оскільки інші параметри, що задаються конструктором, описуються в коді і залежать від версії протоколу, попереднього блоку або вузла, який створив блок. Метод writeToFile Ви можете створити новий запис у двійковому файлі, що складається з блоків у відповідному форматі. Спочатку він перевіряє, чи існує відповідний файл, потім записує всі необхідні поля і на останньому етапі створює запис у базі даних.Метод ReadFromFile призначений для читання блоків з двійкового файлу. Цей метод отримує параметр, який є індексом блоку, що зчитується. Потім створюється новий об'єкт блоку з полями, прочитаними з двійкового файлу, а також всіма транзакціями, записаними в блок. Метод calcMerkleHash використовується для хешування

дерева транзакцій, доданих до блоку, і отриманий хеш може бути використаний для перевірки цілісності вмісту блоку.

Код функції makeBlockRecord

```
    def calcMerkleHash(self):
            transactionhashes=[]
            for t in self.transactions:
                transactionhashes.append(t.getHash())
            while len(transactionhashes)>1:
                transactionhashesnew=[]
                if (len(transactionhashes) % 2 != 0) :
                    transactionhashes.append(transactionhashes[-1])
                transactionhashes.reverse()
                while len(transactionhashes)>0:

transactionhashesnew.append(hashlib.sha256(transactionhashes.pop()+transacti
onhashes.pop()).digest())
                transactionhashes = transactionhashesnew
            return transactionhashes[0]
```

Застосування blockToYaml і blockFromYaml дозволє сформувати блок і передати його по мережі, а також відновити сформований блок після отримання, відповідно. Метод verify перевіряє відповідність блоку протоколу консенсусу, перевіряючи, чи відповідає блок протоколу консенсусу. Першим кроком є обчислення цільового значення даного блоку на основі того, який вузол створив блок і скільки часу пройшло. Потім хеш-значення заголовка порівнюється з цільовим значенням, і якщо хеш-значення менше, блок вважається перевіреним. Метод майнінгу реалізує POS-майнінг, принцип якого заснований на спробі створення блоків за часовий інтервал, який підходить для розміру ставки на створення блоку, що пройде перевірку Він заснований на спробі створення блоку за часовий інтервал, який підходить для розміру ставки на проходження перевірки [19]. Для збільшення ймовірності того, що блок буде створений одним з вузлів, список транзакцій, що записуються в блок, сортується у випадковому порядку. Клас Transaction реалізує об'єкт транзакції і має досить просту реалізацію. Він містить методи для створення, серіалізації та хешування

транзакцій у формат YAML. Конструктор транзакції приймає як параметри індекс транзакції, тип транзакції, ідентифікатор вузла та час створення транзакції. Основним типом транзакції є транзакція конструктора блоку, яка записується в блок першою і включає в себе конструктор блоку і час його створення. Такі транзакції не мають ніякого значення, але можуть використовуватися для тестування системи або для заповнення блоку для швидшої перевірки, коли відсутні реальні транзакції, а також для уникнення непередбачених затримок при створенні нових блоків. Основними типами транзакцій є ті, що містять інформацію про зловмисні або підозрілі події. І, нарешті, транзакції, що містять оновлення правил для підсистеми виявлення вторгнень. Цей метод викликається періодично для сканування всіх існуючих ланцюжків і вимірювання їх довжини.

Код функції chooseMainHead

```
      def chooseMainHead(self):
            if len(self.altHeads)>1:
                fl = self.calculateForkLength()
                print(("Fork length is: ", fl))
                headLength=[]
                for head in self.altHeads:
                    headLength.append(self.getBlocknoVerify(head).time     -
fl['timestamp'])
                hlc = headLength.copy()
                if fl['depth']<3:
                    return
                if len(hlc)>1:
                    hlc.sort(reverse=True)
                    diff=hlc[0]-hlc[1]
                    if (diff>120) or (fl['depth']>7):
                        try:
                            mainHead=self.altHeads[headLength.index(hlc[0])]
                            self.altHeads = [mainHead]
                            self.lastHash = mainHead
                            Block.lasthash = self.lastHash
```

Метод addBlock дає можливість включати нові блоки до ланцюжка. Це пов'язано з тим, що хоча блоки криптографічно пов'язані один з одним, для

правильної маніпуляції з блоками потрібен об'єкт ланцюжка. Цей метод приймає новий блок, що додається до ланцюжка, як параметр і встановлює поля існуючого об'єкта ланцюжка відповідно до даних цього блоку. addTransactionToPool і removeFromPool використовуються для додавання транзакцій до тимчасового пулу і видалення транзакцій з нього відповідно. AddTransactionToPool і removeTransactionFromPool додають транзакції до тимчасового пулу і видаляють транзакції з нього відповідно. Обидва методи отримують посилання на об'єкт транзакції як параметр. Для забезпечення обміну даними між вузлами блокчейну в мережі використовується фреймворк Flask, який дозволяє створювати відповідні API-інтерфейси на основі протоколу HTTP. Для створення такого API спочатку необхідно запустити мережевий сервіс, який прослуховує певний порт і встановлює з'єднання з іншими клієнтами. Також необхідно створити обробник маршруту, який викликає метод, що відповідає конкретному запиту з відповідною адресою. Як приклад, наведемо обробник запиту до блокчейну.

Код функції get_Chainc

```
@app.route('/chain/<head>', methods=['GET'])
def get_chainc(head):
    try:
        blockIndex=chain.getBlockIndex(unhexlify(head))
        return yaml.dump(blockIndex)
    except:
        return 'None'
```

Такий метод дає можливість приймати запити, які отримують блокчейн, що закінчується блоком із зазначеним хешем. Хеш передається з HTTP-запиту до відповідного методу обробника. Обробник перевіряє, чи існує запитуваний ланцюжок або блок, і надсилає відповідні дані, якщо він існує; якщо ні, то повертає повідомлення про помилку. Значення, повернуте методом-обробником, надсилається у відповідь на запит, так що вузол, який зробив запит, отримує необхідні дані або повідомлення про те, що він їх не отримав. Блокчейн-

підсистема розподіленої системи виявлення вторгнень використовує багатопотокову реалізацію, що дозволяє краще використовувати ресурси багатопроцесорної системи. Основні завдання розподіляються між потоками наступним чином. Перший потік відповідає за підтримку блокчейну в актуальному стані шляхом обробки та додавання нових блоків і синхронізації стану ланцюжка з іншими вузлами блокчейну; другий потік відповідає за генерацію транзакцій на основі даних, отриманих від модуля виявлення вторгнень; третій потік реалізує POS-майнер і відповідає за генерацію нових блоків [21]. Інші завдання, такі як обробка запитів до API та управління потоками, виконуються головним потоком.

Якщо роботу одного з вузлів підсистеми блокчейна необхідно завершити, то завершення операції повинно бути виконано таким чином, щоб гарантувати цілісність записаних даних. Для цього головний потік надсилає сигнал іншим потокам про необхідність завершення роботи і переходить у режим очікування відповіді. Отримавши сигнал про необхідність завершення роботи, інший потік доводить поточний етап виконуваного ним завдання до логічного завершення, якщо потік у цей час працює, або просто завершує свою роботу, якщо потік у цей час простоює. Головний потік, який отримав повідомлення про завершення роботи іншого потоку, також завершує свою роботу. Таке завершення дає змогу уникнути помилок у бінарному файлі через переривання запису і знижує ймовірність переривання передачі даних на інші вузли мережі. Цілісність бінарного файлу контролюється в разі аварійного вимкнення підсистеми блокчейна через збій живлення або деякі програмні збої в операційній системі. Такі події є скоріше винятком, ніж нормою, але можливість їхнього виникнення слід передбачити. Пошкодження бінарних файлів також може бути спричинене поганим станом носія, на якому зберігається база даних блокчейна. Спроби використовувати пошкоджені файли можуть призвести до збоїв у системі або помилок під час роботи. Це означає, що для запобігання подібних ситуацій необхідно своєчасно перевіряти та відновлювати цілісність блокчейна. Для такої перевірки цілісності бази даних блокчейна під час запуску сканується останній

або всі файли блокчейна, залежно від конфігурації. Під час сканування здійснюється пошук усіх доступних блоків у файлі та обчислення хешів для контролю цілісності. Якщо знайдено недійсний блок, увесь ланцюжок, створений після цього блоку, вважається недійсним, такі дані видаляються з файлу, а блок із потрібним порядковим номером завантажується заново.

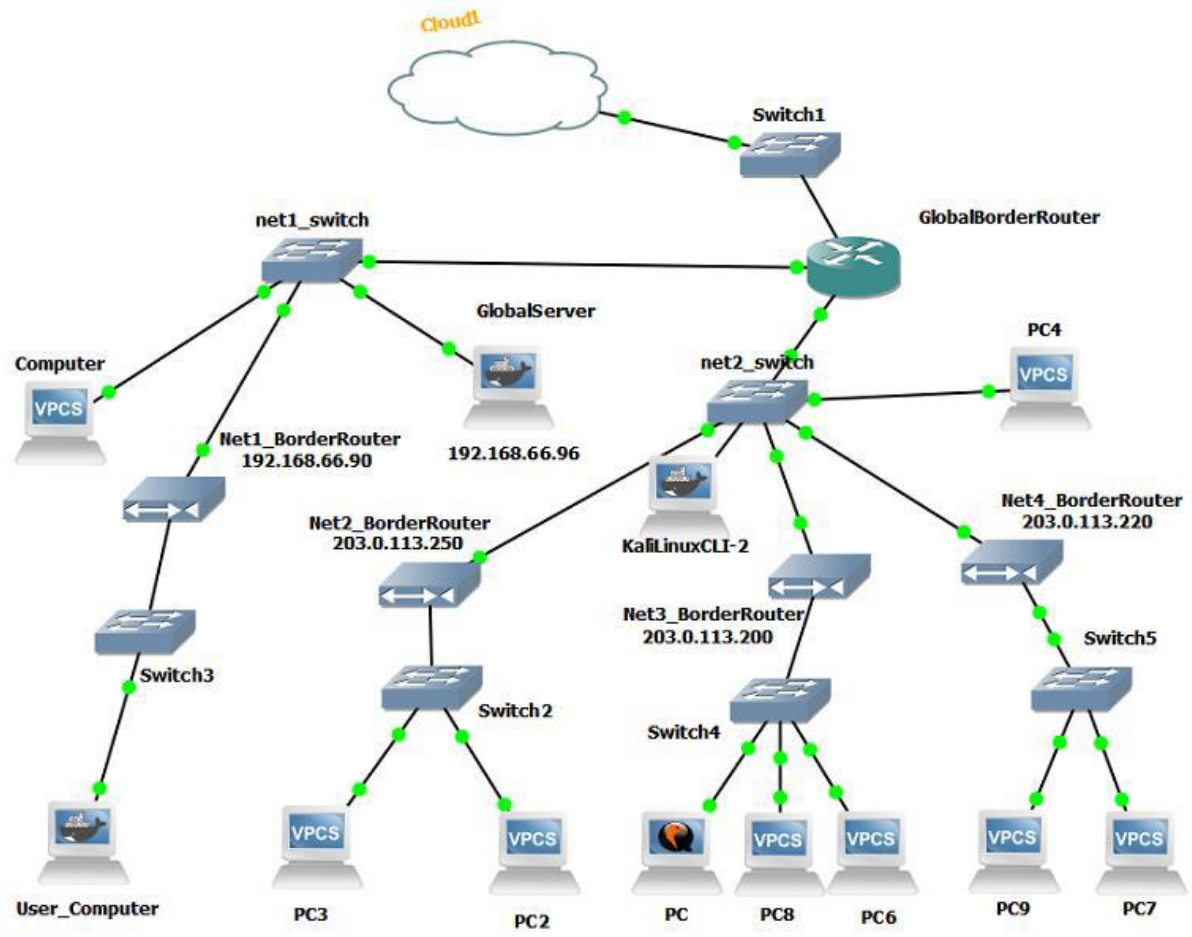


Рисунок 3.2 – Модель цифрової мережі розподіленої комп’ютерної системи для перевірки виявлення загроз

Щоб переконатися, що система буде правильно функціонувати в реальній мережі, було вирішено створити тестову мережу з аналогічною архітектурою в навчальній лабораторії з моделювання кіберзагроз. Експериментальна мережа містить п'ять вузлів, об'єднаних в один сегмент. Кожен вузол являє собою

низькорівневий сегмент мережі, а оскільки вузли, розташовані за маршрутизаторами, не мають істотного впливу на перебіг експерименту, їхні компоненти були створені за допомогою віртуалізації. На окремому комп'ютері також було встановлено KaliLinux для генерації шкідливого трафіку. Така модель мережі цілком відповідає роздільному поєднанню мереж SMB і SOHO. Зазвичай швидкість з'єднання в локальних мережах вища, а час опрацювання запитів менший, ніж під час взаємодії через Інтернет, але ці умови ближчі до реальних середовищ, ніж мережі на основі віртуальних машин.

Експеримент було налаштовано аналогічно віртуальному середовищу, але оскільки реалізувати перехоплення трафіку в реальній мережі набагато складніше, для оцінки поведінки в підсистемі блокчейну було передбачено логування відправлених та отриманих запитів. Оскільки результати запуску системи на реальному обладнанні подібні до результатів запуску на моделі віртуальної мережі, перехоплення трафіку набагато простіше здійснювати за допомогою моделі віртуальної мережі, враховуючи збільшення часу проходження мережевих пакетів.

ВИСНОВКИ

В результаті виконання поставленого у магістерській роботі завдання були отримані результати щодо побудови інформаційної технології виявлення та аналізу аномальних подій для захисту комп'ютерних мереж для малого та середнього бізнесу на основі блокчейну. За результатами магістерської роботи можна зробити наступні висновки: Результати аналізу основних загроз інформаційній безпеці комп'ютерних мереж підприємств малого та середнього бізнесу дозволили виявити існуючі методи та заходи захисту з урахуванням функціональних та експлуатаційних особливостей даного класу мереж. Розроблена модель децентралізованої інформаційної системи виявлення та аналізу аномальних подій на основі технології блокчейн є основою для побудови різноманітних децентралізованих систем захисту комп'ютерних мереж МСБ. Запропонований метод дає можливість вибору PoS-орієнтованих варіантів розподілених інформаційних систем виявлення та аналізу аномальних подій, зменшуючи навантаження на комп'ютери під час роботи системи та знижуючи енергоспоживання. При цьому знижуються вимоги до обчислювальних потужностей обладнання, що робить його придатним для використання в комп'ютерних мережах малих і середніх підприємств. Протокол консенсусу PoS, адаптований для децентралізованих систем виявлення вторгнень, забезпечує надійну і стабільну роботу компонентів блокчейну і споживає на 80% менше обчислювальних ресурсів, ніж базова версія протоколу. За результатами аналізу методів класифікації аномалій створено перелік базових класифікаторів, які об'єднано в комплексний класифікатор для підвищення точності виявлення аномальних подій, невідомих розподіленим системам виявлення вторгнень. У сукупності вищезазначені наукові результати складають нову інформаційну технологію виявлення та аналізу аномальних подій з метою захисту комп'ютерних мереж на основі блокчейн-технології. Запропонована інформаційна технологія може бути використана як розробниками систем захисту комп'ютерних мереж, так і мережевими адміністраторами. Розроблені бізнес-процеси та архітектура закладуть основу для створення більш потужних і функціональних розподілених систем виявлення вторгнень.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 8th edition ed., Pearson; , 2019, p. 832.

2. R. Shirey, " Internet Security Glossary, Version 2," Network Working Group, August 2007. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4949. [Accessed 20 10 2023].

3. Thanh Vu, S. N., Stege, M., El-Habr, P. I., Bang, J., & Dragoni, N., "A survey on botnets: Incentives, evolution, detection and current trends," Future Internet, vol. 13, p. 198, 2021.

4. Kumar, S., Pathak, S. K., & Singh, J., "A Comprehensive Study of XSS Attack and the Digital Forensic Models to Gather the Evidence," ECS Transactions, vol. 107, 2022.

5. І. Г. Бондарєв, Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи, Львів: Львівська політехніка, 2016.

6. Van Niekerk, J. F., Von Solms, R., "Information security culture: A management perspective.," Computers & Security, vol. 29(4), p. 476–486, 2010.

7. V. Lytvynov, N. Stoianov, I. Stetsenko, I. Skiter, O. Trunova, A. Hrebennyk, V. Nekhai, I. Burmaka, Attacks defense of computer nets by tools using extended information about environment : monograph, Chernihiv : Chernihiv Politechnic National University , 2021. - 212c..

8. Alvarez, J. R. N., Zamora, Y. P., Pina, I. B., & Angarita, E. N., "Demilitarized network to secure the data stored in industrial networks.," International Journal of Electrical & Computer Engineering, vol. 11, 2021.

9. Rababah, B., Zhou, S., & Bader, M., "Evaluation the Performance of DMZ.," International Journal of Wireless and Microwave Technologies, vol. 1, 2018.

10. Skiter, I., Burmaka, I., & Sigayov, A., "Design of Technical Methods for Analysing Network Security Based on Identification of Network Traffic Anomalies," Information & Security, vol. 47, no. 3, pp. 306-316, 2020.

11. N. H. Nguyen, Essential Cyber Security Handbook In Ukrainian, 2018.

12. D. K. James Michael Stewart, Network Security, Firewalls, and VPNs, Jones & Bartlett Learning, 2020.

13. Wes Noonan, Ido Dubrawsky, Firewall Fundamentals, Cisco Press, 2006.

14. H. Andrea, Cisco ASA Firewall Fundamentals, Amazon Digital Services, 2014.

15. Peter N. M. Hansteen, The Book of PF: A No-Nonsense Guide to the OpenBSD Firewall, No Starch Press, 2008.

16. І. Бурмака, "Класифікація систем виявленя вторгнень в розподілені інформаційні системи".Проблеми зняття з експлуатації об’єктів ядерної енергетики та відновлення навколишнього середовища (INUDECO 17): збірник матеріалів IV Міжнародної конференції (25–27 квітня 2017, м. Славутич). – Чернігів : ЧНТУ, 2017. – с. 59-63.

17. І. А. Бурмака, "Архітектура розподіленої системи виявлення вторгнень на основі blockchain технології".Проблеми зняття з експлуатації об’єктів ядерної енергетики та відновлення навколишнього середовища (INUDECO 2020) в режимі онлайн: збірник матеріалів V Міжнародної конференції (27–29 квітня 2020, м. Славутич). – Чернігів : ЧНТУ, 2020. с.54-59.

18. Burmaka, I., Dorosh, M., Skiter, I., & Lytvyn, S, "Architecture of Distributed Blockchain Based Intrusion Detecting System for SOHO Networks," Mathematical Modeling and Simulation of Systems (MODS'2020): Selected Papers of 15th International Scientific-practical Conference, MODS, 2021 June 28–July 01, Chernihiv, Ukraine. Springer Nature, pp. 313-326, 2021.

19. І. А. Бурмака, М. С. Дорош, "Оптимізація використання обчислювальних ресурсів розподіленою системою виявлення вторгнень на основі blockchain".Проблеми зняття з експлуатації об’єктів ядерної енергетики та відновлення навколишнього середовища (INUDECO 21) : збірникматеріалів VI Міжнародної конференції (27–29 квітня 2021,м. Славутич). – Чернігів : НУ «Чернігівська політехніка», 2021. – с. 47-50.

20. Burmaka, I. A., Lytvynov, V. V., Skiter, I. S., & Lytvyn, S. V., "Evaluating a blockchain-based network performance for the intrusion detection system.," Математичні машини і системи, vol. 1, pp. 99-109, 2020.

21. Zhang, S., & Lee, J. H., "Analysis of the main consensus protocols of blockchain," ICT express, vol. 6, no. 2, pp. 93-97, 2020.