Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Модель оцінки ризиків атак cryptojacking»

Виконавець: студент ІІ курсу, групи КБм-21

_____ **Максим ПАНЧЕНКО**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Тетяна БАБЕНКО	
Нормоконтроль	Лариса МИРУТЕНКО	

Київ 2024

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» листопада 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача КБм-21 (група) Панченка Максима Валерійовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Модель оцінки ризиків атак cryptojacking

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	процес оцінки ризиків типу cryptojacking на кінцевих системах.
Предмет досліджень	кількісні моделі оцінки ризиків кібербезпеки.
Мета	розробка моделі оцінки ризиків атак типу cryptojacking.
Вихідні дані для проведення роботи	дослідження атак cryptojacking, методи їх виявлення та блокування, методи оцінки ризиків.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна	розроблено вдосконалений алгоритм для оцінки ризиків в атаках cryptojacking, що відрізняється від існуючих застосуванням методу аналізу дерева відмов для визначення ймовірності атак та використанням симуляції Монте-Карло для передбачення розміру можливих фінансових збитків.
Практична цінність	можливість впровадження розробленої моделі під час оцінки ризиків атак cryptojacking та формування бюджету для засобів захисту в організаціях.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	**Строки виконання робіт (початок-кінець)**
Уточнення постановки задачі	17.11.2023 – 29.01.2024
Аналіз літературних джерел	30.01.2024 – 12.02.2024
Дослідження атак типу cryptojacking, методів виявлення та блокування	13.02.2024 – 21.02.2024
Аналіз методів оцінки ризиків інформаційної безпеки	22.02.2024 – 26.02.2024
Аналіз вхідних даних для оцінки ризиків атак cryptojacking	27.02.2024 – 04.03.2024
Розробка алгоритму кількісної оцінки ризиків атак типу cryptojacking	05.03.2024 – 10.03.2024
Розробка сценаріїв атак cryptojacking для оцінки ризиків	11.03.2024 – 17.03.2024
Розробка програмного засобу, що реалізує алгоритм оцінки ризиків	18.03.2024 – 19.03.2024
Формування вхідних даних для оцінки ризиків на прикладі організації та виконання симуляції Монте-Карло	20.03.2024 – 17.04.2024
Аналіз отриманих результатів та формування висновків	18.04.2024 – 25.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	26.04.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 18.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект	Зменшення збитків та кількості атак cryptojacking
Соціальний ефект	Покращення процесу оцінки ризиків атак cryptojacking завдяки впровадженню структурованого процесу сучасними методами

7. ДОДАТКОВІ ВИМОГИ

Завдання видала _____ Тетяна БАБЕНКО
(підпис) (Ім'я, ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Максим ПАНЧЕНКО
(підпис) (Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Модель оцінки ризиків атак cryptojacking»: 99 сторінок, 54 рисунки, 1 таблиця, 80 літературних джерел.

Об'єкт дослідження: це процес оцінки ризиків атак типу cryptojacking на кінцевих системах.

Мета роботи: розробка моделі оцінки ризиків атак типу cryptojacking.

Методи дослідження: аналіз, моделювання, синтез, експеримент, тестування.

У роботі проаналізовано атаки типу cryptojacking, що спрямовані на кінцеві точки, досліджено існуючі методи виявлення та блокування цих атак, а також методи оцінки ризиків, які дозволяють аргументувати впровадження нових засобів захисту. Запропоновано алгоритм кількісної оцінки ризиків атак cryptojacking, побудовано сценарії атак, розроблено програмний засіб, який реалізує алгоритм оцінки ризиків.

Практичне значення роботи полягає у розробленій моделі оцінки ризиків атак cryptojacking. Результати здійснених у кваліфікаційній роботі досліджень можуть бути використані під час формування стратегії засобів захисту в організаціях.

Наукова новизна дослідження полягає в розробці вдосконаленого алгоритму для оцінки ризиків в атаках cryptojacking, що відрізняється від існуючих застосуванням методу аналізу дерева відмов для визначення ймовірності атак та використанням симуляції Монте-Карло для передбачення розміру можливих фінансових збитків.

Напрямки подальших досліджень: уточнення розроблених сценаріїв атак, формування нових сценаріїв для конкретних систем, зокрема, хмарних середовищ.

Ключові слова: кібербезпека, кібератаки, криптовалюти, майнінг, cryptojacking, засоби захисту, управління ризиками, кількісна оцінка ризиків, метод Монте-Карло.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ALE	–	Annual Loss Expectancy
ARO	–	Annual Rate of Occurrence
ATA	–	Attack Tree Analysis
CART	–	Classification And Regression Tree
CMMI	–	Capability Maturity Model Integration
CNN	–	Convolutional Neural Network
CORAS	–	Consultative Objective Risk Analysis System
CPU	–	Central Processing Unit
CRAMM	–	CCTA Risk Analysis and Management Method
DDoS	–	Distributed Denial-of-Service
DPI	–	Deep Packet Inspection
EDR	–	Endpoint Detection and Response
ETA	–	Event Tree Analysis
FAIR	–	Factor Analysis of Information Risk
FRAP	–	Facilitated Risk Analysis Process
FTA	–	Fault Tree Analysis
GPU	–	Graphics Processing Unit
HTTPS	–	HyperText Transfer Protocol Secure
IoT	–	Internet of Things
IP	–	Internet Protocol
ISACA	–	Information Systems Audit and Control Association
IT	–	Information Technology
k-NN	–	k-Nearest Neighbors
LEC	–	Loss Exceedance Curve
LEF	–	Loss Event Frequency
LM	–	Loss Magnitude
LotL	–	Living-off-the-Land

LSTM	–	Long Short-Term Memory
OCC	–	One-Class Classification
OCTAVE	–	Operationally Critical Threat, Asset, and Vulnerability Evaluation
PERT	–	Project Evaluation and Review Technique
POW	–	Proof of Work
RF	–	Random Forest
RNN	–	Recurrent Neural Network
ROSI	–	Return on Security Investment
SLE	–	Single Loss Expectancy
SOC	–	Security Operations Center
SWOT	–	Strengths, Weaknesses, Opportunities, Threats
TCP	–	Transmission Control Protocol
TEF	–	Threat Event Frequency
TLS	–	Transport Layer Security
UDP	–	User Datagram Protocol
URL	–	Uniform Resource Locator
Wasm	–	Web Assembly
WMI	–	Windows Management Instrumentation

ЗМІСТ

ВСТУП

У сучасному світі кіберзагрози постійно еволюціонують, що призводить до появи нових форм атак. Однією із них є несанкціоноване використання зловмисником обчислювальних ресурсів для виконання майнінгу криптовалют. Такі атаки мають назву cryptojacking та стають дедалі частішими. Вони призводять до значних фінансових збитків організаціям та користувачам через збільшення витрат на електроенергію та сповільнення роботи пристроїв.

Розуміння потенційних наслідків атак cryptojacking дозволяє ефективно планувати заходи захисту. Тому актуальними є дослідження з розробки та вдосконалення моделей прогнозування та оцінки ризиків, які враховують різноманітні аспекти атак cryptojacking, включаючи їхній потенційний вплив на системні ресурси та фінансові збитки. Це дозволить організаціям ефективніше адаптуватися до змін та приймати інформовані рішення щодо заходів безпеки та реагування на потенційні загрози.

Метою роботи є розробка моделі оцінки ризиків атак типу cryptojacking.

Для досягнення зазначеної мети кваліфікаційної роботи поставлені окремі завдання:

- провести аналіз атак cryptojacking та засобів їх виявлення і блокування;
- визначити вхідні дані для оцінки ризиків атак cryptojacking;
- провести аналіз сучасних методів оцінки та прогнозування ризиків інформаційної безпеки;
- розробити алгоритм оцінки ризиків атак cryptojacking на основі обраних методів;
- розробити сценарії атак, що застосовуються під час оцінки ризиків;
- розробити програмний засіб, який реалізує алгоритм оцінки ризиків;
- оцінити ризики атак cryptojacking для обраного підприємства згідно розроблених сценаріїв для підтвердження доцільності використання моделі.

Об'єкт дослідження – це процес оцінки ризиків атак типу cryptojacking на кінцевих системах.

Предмет дослідження – кількісні моделі оцінки ризиків кібербезпеки.

Методи дослідження: аналіз, моделювання, синтез, експеримент, тестування.

Наукова новизна одержаних результатів:

- розроблено вдосконалений алгоритм для оцінки ризиків в атаках cryptojacking, що відрізняється від існуючих застосуванням методу аналізу дерева відмов для визначення ймовірності атак та використанням симуляції Монте-Карло для передбачення розміру можливих фінансових збитків.

Практична цінність роботи полягає у:

- розробці математичної моделі симуляції Монте-Карло для атак cryptojacking, що включає в себе розподіл ймовірності згідно характеру збитків внаслідок таких несанкціонованих дій.
- розробці моделі оцінки ризиків для атак cryptojacking, що дозволяє отримати прогнозовані значення збитків, які застосовуються для планування стратегії засобів захисту;
- розробці програмного засобу, який реалізує алгоритм оцінки ризиків.

Основні наукові положення і результати роботи доповідалися та обговорювалися на Міжнародній науково-практичній конференції «Проблеми кібербезпеки інформаційно-телекомунікаційних систем» (Київ, 2024). Основні положення кваліфікаційної роботи викладені в 1 науковій праці у матеріалах наукових конференцій.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ АТАК ТИПУ CRYPTOJACKING

1.1 Атаки cryptojacking

Криптовалюти – це цифровий спосіб оплати, який не має єдиного центру управління. З того дня, як у 2009 році було випущено біткойн, криптовалюти на основі блокчейну викликають дедалі більший інтерес за межами конкретних спільнот, таких як банки та комерційні установи [1]. Криптовалюти являють собою токени, що є зашифрованим рядком даних, облік яких ведеться в децентралізованій базі даних, яка має назву блокчейн, із записами усіх транзакцій у вигляді блоків. Процес розрахунку криптографічних операцій, що застосовується для підтвердження транзакцій та формування нових блоків називається майнінгом.

Блокчейн використовується як безпечний, приватний і надійний публічний архів [2, с. 4], тому криптовалюти більш захищені від шахрайства та крадіжки особистих даних. Будь-хто може надіслати криптовалюти у будь-який час і куди завгодно, без затримок або додаткових чи прихованих платежів від посередників.

Під час майнінгу криптовалют всі майнери конкурують, щоб знайти рішення дуже складної задачі, що має назву Proof of Work (POW) [3, с. 1], яка необхідна для завершення та запису всіх нових транзакцій валюти (рис. 1.1). Транзакції групуються в блоки, і лише перший майнер, який знаходить правильне рішення для блоку, отримує певну кількість валюти, що бере участь у транзакціях. Це означає, що майнер з обмеженими ресурсами зазвичай не зможе знайти рішення раніше за інших, і тому він ніколи не отримає винагороду. З цієї причини зазвичай створюються пули майнерів, які співпрацюють, щоб об'єднати свої зусилля та мати більше шансів стати першими, хто знайде рішення. Таким чином, вони можуть розділити між собою отримані винагороди. Тому пристрої та комп’ютери, підключені до Інтернету, хоч і недостатньо швидкі для обробки даних, все одно можуть бути використані для отримання прибутку від майнінгу.

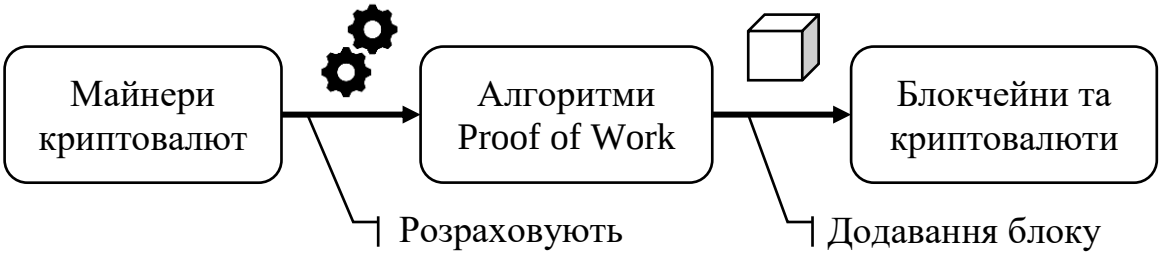

Рисунок 1.1 – Схема роботи криптомайнінгу

Для кіберзлочинців несанкціонований майнінг криптовалют є вигідним методом отримання прибутку. Такий тип атак має назву cryptojacking. Замість того, щоб витрачати кошти на власні криптоферми, зловмисники створюють шкідливе програмне забезпечення, яке виконує криптомайнінг на обчислювальних потужностях інформаційної системи жертви. У результаті зловмисник отримує прибутки у вигляді криптовалюти та не має жодних витрат на апаратне забезпечення. Крім того, зловмисне програмне забезпечення для майнінгу криптовалют одночасно заражає велику кількість пристроїв та формує ботнети для збільшення розміру винагороди [4].

Про прибутковість несанкціонованого майнінгу свідчить перехід деяких кіберзлочинців на цей тип атак, зокрема, група AstraLocker оголосила про припинення операцій з програмами-вимагачами та перехід на cryptojacking [5, с. 30].

Хоча несанкціоновані криптомайнери не мають настільки явного зловмисного впливу, як, наприклад, віруси-шифрувальники, вони все одно призводять до суттєвих збитків. Звичайні користувачі отримають збільшений рахунок за електроенергію та суттєве сповільнення роботи комп’ютера, а для організацій такі кібератаки можуть становити значно суттєвіші збитки. Зокрема, cryptojacking є суттєвою загрозою для банків, адже такі атаки можуть призвести до порушення операційної діяльності та спричинити фінансові збитки, тим самим завдаючи шкоди стабільності та репутації банку [6, с. 4]. Також для організацій, що використовують хмарну інфраструктуру це може суттєво вплинути на рахунки за хмарні послуги. Згідно дослідження Sysdig, вартість майнінгу однієї монети Monero на одному екземплярі AWS EC2 коштує у середньому $11 000. Також прибуток зловмисника розміром $1 спричинить втрати

для жертви розміром $53, але оскільки ціна криптовалют дуже нестабільна, реальні прибутки зловмисників можуть сильно змінюватися з часом, що ще більше викривляє співвідношення доходів зловмисників до витрат жертв. [7, с. 8].

Збитки від криптомайнінгу для організацій можуть завдавати не тільки кіберзлочинці. Це є одним із потенційних методів монетизації вебсервісів. Працівники можуть випадково або навмисно використовувати такі сервіси. Криптомайнінг на потужностях компанії також можуть виконувати недобросовісні співробітники.

Атаки cryptojacking можуть бути націлені на різні платформи, включаючи настільні комп'ютери, мобільні пристрої, прилади Інтернету речей (IoT) [8, с. 3], та навіть електромобілі [9]. Зокрема, під час досліджень окрім систем під керуванням Windows були виявлені випадки cryptojacking на таких пристроях [10]:

- пристрої iOS;
- пристрої, що працюють на операційній системі Ubuntu;
- домашні маршрутизатори;
- пристрої моніторингу середовища, що використовуються в дата-центрах;
- Smart TV і мобільні пристрої під управлінням Android;
- IP камери;
- сервери друку;
- ігрові консолі.

Несанкціонований майнінг може виконуватись із використанням самостійного зловмисного програмного забезпечення, або за допомогою вбудованих на сайтах скриптів [11]. Крім того, такі атаки можуть бути частиною іншого шкідливого програмного забезпечення, зокрема, ransomware. На рисунку 1.2 наведено приклад атаки cryptojacking шляхом несанкціонованого підключення до комп'ютера жертви з використанням інструменту Metasploit Framework, що є частиною Kali Linux [12].

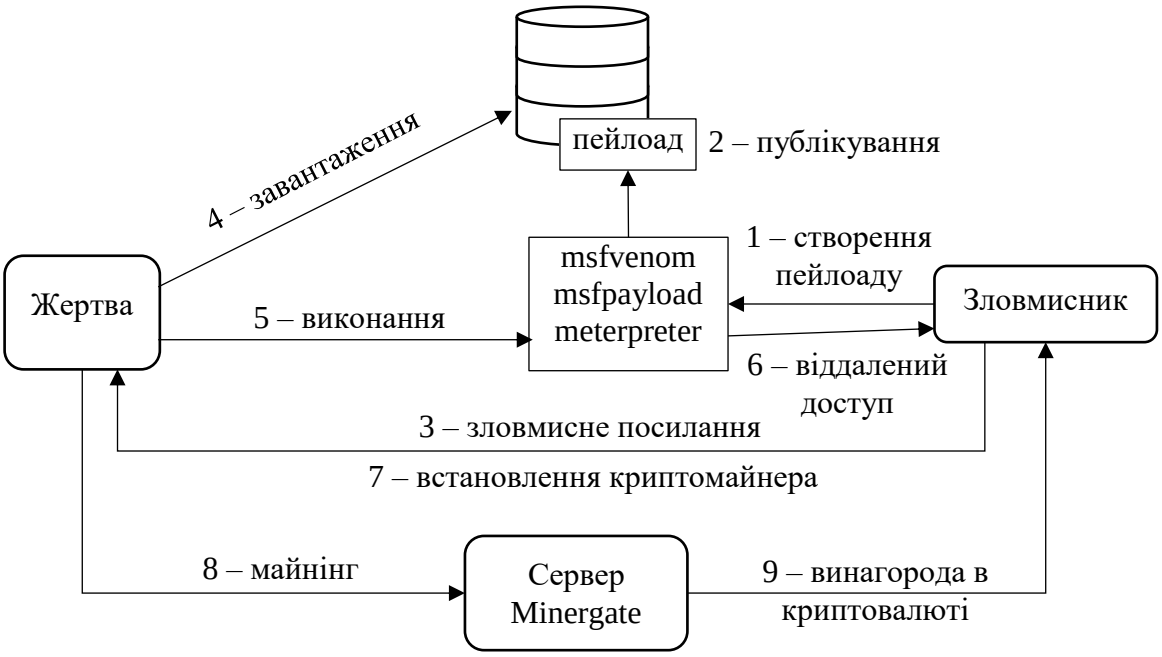

Рисунок 1.2 – Приклад атаки cryptojacking з отриманням віддаленого доступу

На схемі зображено такі кроки:

1. Зловмисник створює зловмисне програмне забезпечення з корисним навантаженням (пейлоад) meterpreter, що призначений для встановлення підключення із комп’ютером жертви за допомогою інструмента msfvenom, що є частиною Metasploit Framework.

2. Створене зловмисне програмне забезпечення завантажується на сервері зловмисника, що доступний в інтернеті.

3. Зловмисник надсилає посилання на свій сайт жертві, наприклад, за допомогою фішингу.

4. Жертва завантажує зловмисне програмне забезпечення.

5. Виконання зловмисного програмного забезпечення та запуск модуля meterpreter.

6. Встановлення з’єднання із зловмисником та надання віддаленого доступу до комп’ютера жертви. Підключення може бути замасковане під звичайне HTTPS з’єднання.

7. Зловмисник встановлює криптомайнер на комп’ютері жертви.

8. Запускається процес криптомайнінгу, під час якого відбувається обмін даними із майнінговим сервісом, наприклад, MinerGate.

9. За виконання майнінгу зловмисник отримує винагороду в криптовалюті.

Зазначимо, що під час цієї атаки зловмисник отримує повний доступ до комп'ютера жертви, а отже він може виконати будь-які додаткові атаки, наприклад, використовувати пристрій для розсилки спаму та фішингових листів, DDoS атак, викрадення даних користувача, тощо.

Атаки cryptojacking також можуть виконуватись через вебсайти. Коли користувач переходить на такий сайт відбувається завантаження коду JavaScript [13, с. 1] або WebAssembly (Wasm) [14, с. 5], який виконує криптомайнінг у браузері користувача. Такий спосіб криптомайнінгу вперше був представлений у 2013 році як підтвердження концепції. Це рішення було розроблено студентами Массачусетського технологічного інституту як потенційна альтернатива браузерній рекламі під час хакатону [15, с. 1]. Сьогодні зловмисники активно використовують цю розробку для отримання прибутку. Збільшення популярності таких атак відбулось після початку роботи сервісу браузерного криптомайнінгу Coinhive у 2017 році [16, с. 4]. Цей сервіс був розроблений як легальний спосіб отримання прибутку від сайтів замість реклами. Натомість, він активно використовувався зловмисниками для створення вебсайтів, що виконують майнінг без дозволу користувачів, або для зараження інших сайтів. Зокрема, прогнозована кількість сайтів у всій мережі інтернет, що виконували майнінг криптовалют у браузері без дозволу користувача у 2018 році, становила 0,011% та майже половина із них використовувала сервіс Coinhive [16, с. 14]. Цей сервіс був закритий у 2019 році, але такий тип атак досі є актуальним.

Шкідливі скрипти для майнінгу криптовалют можуть бути вбудовані на вебсайтах кількома способами [17, с. 2–3]:

- власники сайтів можуть вбудовувати такі скрипти на свої сайти та активувати їх без згоди відвідувачів;
- сторонні сервіси можуть впроваджувати такі скрипти, не повідомляючи про це ні власників сайтів, ні кінцевих користувачів;

- шкідливі розширення для браузерів можуть непомітно запускати майнери криптовалют у фоновому режимі;
- зловмисники можуть зламувати сервери, розширення для браузерів, сторонні сервіси та впроваджувати шкідливе програмне забезпечення для викрадення криптовалют;
- вразливі мережеві пристрої, такі як роутери, точки доступу тощо, можуть бути використані для модифікації вебтрафіку та ін'єкції зловмисних скриптів на сайтах, що не використовують протокол HTTPS [18].

Варто зазначити, що такий майнінг не завжди незаконний, адже деякі сайти використовують цю техніку для монетизації своїх сервісів [19; 20]. У будь-якому випадку кінцеві користувачі понесуть втрати через додаткове навантаження на свої пристрої. Схема атаки cryptojacking у браузері показана на рисунку 1.3.

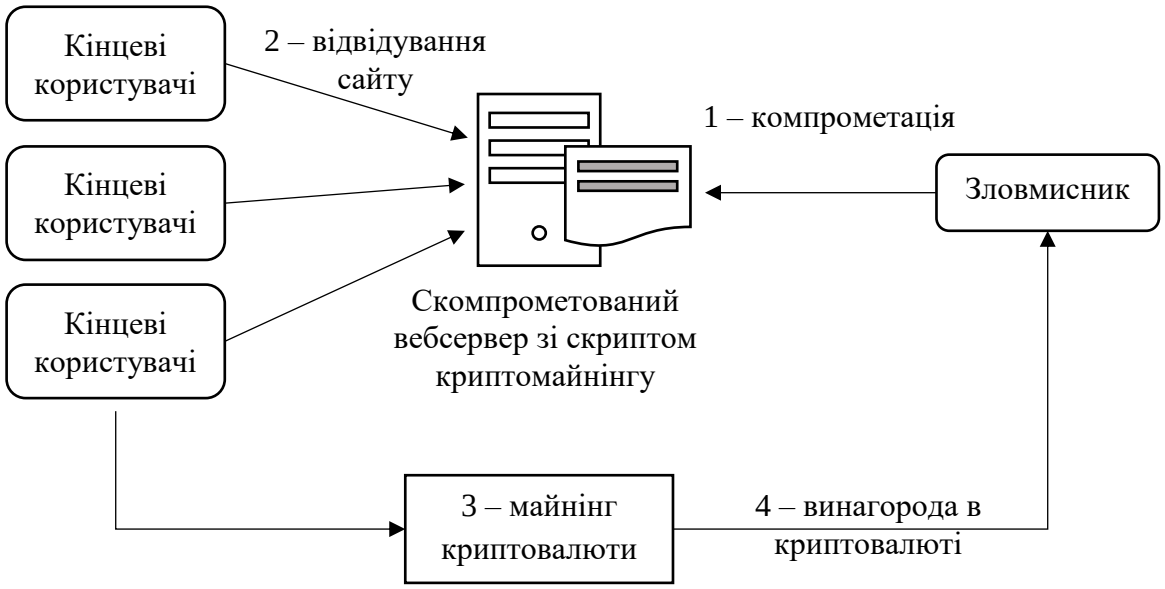

Рисунок 1.3 – Cryptojacking у браузері

На схемі показано такі кроки:

1. Зловмисник використовує вразливості на вебсайті для отримання доступу та завантажує скрипт криптомайнінгу на вебсервер.
2. Користувачі відвідують вебсайт за запускають скрипт у своїх браузерах.
3. Скрипт виконує криптомайнінг на комп'ютерах користувачів.

4. Зловмисник отримує винагороду в криптовалюті за виконаний майнінг.

У 2023 році кількість випадків атак cryptojacking перевищила річний показник 2022 року вже на початку квітня і продовжувала збільшуватись. До кінця року дослідники загроз SonicWall Capture Labs зафіксували 1,06 мільярдів випадків cryptojacking – на 659% більше, ніж у 2022 році. Цьому сприяли безпрецедентні обсяги атак у листопаді та грудні, коли було зафіксовано більше випадків атак, ніж за весь 2022 рік [21]. Порівняльний графік кількості таких атак за 2022 та 2023 рік показано на рисунку 1.4.

загальне споживання електроенергії виявлених випадків cryptojacking становила 278K кВт·год кожного дня у 2018 році, що еквівалентно споживанню населеним пунктом, у якому проживає 9300 мешканців [22, с. 13].

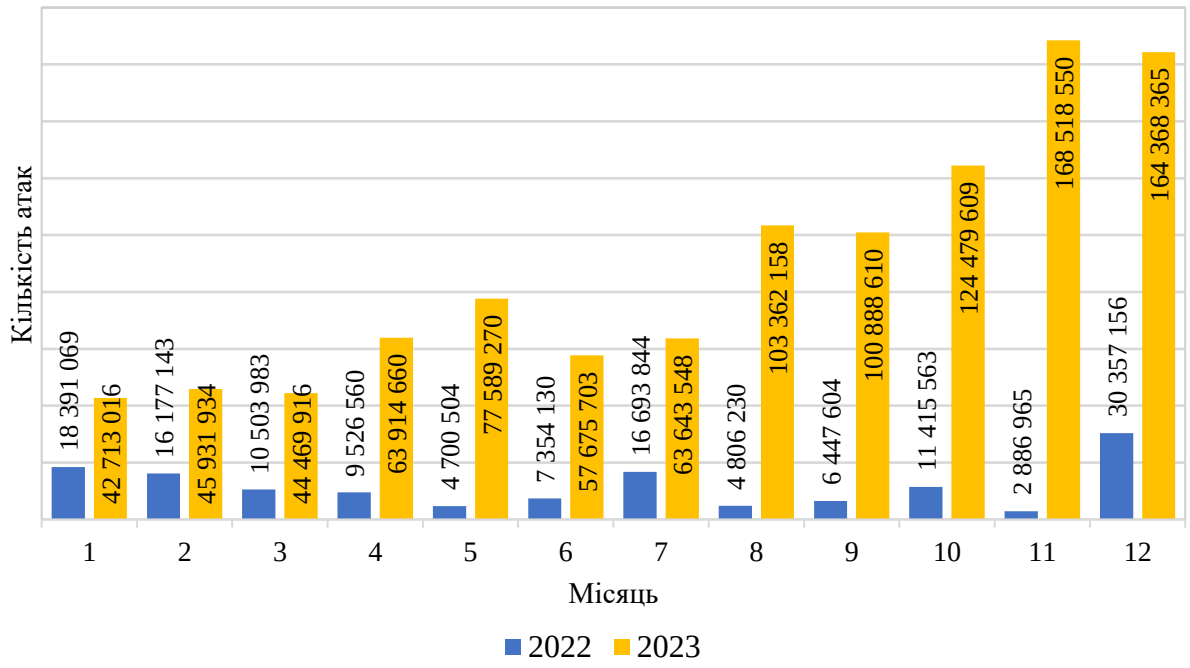

Рисунок 1.4 – Кількість атак cryptojacking за місяцями у 2022 та 2023 роках

За даними SonicWall, атаки cryptojacking становили одну шосту частину всього зловмисного програмного забезпечення у 2023 році [21]. Отже, надзвичайно актуальним є впровадження інструментів виявлення та блокування таких атак, а також оцінка можливих збитків.

1.2 Аналіз роботи поширених зловмисних криптомайнерів та технік обходу антивірусних засобів

Для виконання атак cryptojacking зловмисники використовують техніки для обходу антивірусного програмного забезпечення, зокрема:

- Шифрування та обфускація виконуваного файлу – через шифрування код програмного забезпечення буде недоступний для антивірусних програм, тому його неможливо буде виявити за сигнатурами. Обфускація виконуваного файлу може змінити структуру та назви функцій, змішати порядок інструкцій та вставити додаткові функції, що ускладнює аналіз коду. Це може ускладнити виявлення шкідливого ПЗ за допомогою традиційних антивірусних програм та статичного аналізу.
- Обмеження використання ресурсів системи (тротлінг) – якщо під час майнінгу буде використано 100% ресурсів процесора та відеокарти, тоді такі дії можуть бути одразу виявлені.

Також для уникнення виявлення антивірусними засобами зловмисник використовують безфайлове зловмисне ПЗ під час виконання атак cryptojacking [23]. На відміну від традиційних шкідливих програм, цей підхід не використовує для поширення виконувані файли або скрипти. Таке шкідливе програмне забезпечення напряму завантажується в оперативну пам'ять і може використовувати вразливості у легальному програмному забезпеченні. Крім того, використовується метод living-off-the-land (LotL), що передбачає використання функцій легітимних інструментів, таких як підписки Windows Management Instrumentation (WMI), PowerShell та макроси Microsoft Office, для запуску зловмисного коду та створення запланованих задач [23, с. 2], які забезпечують постійну присутність в системі.

Згідно звіту SonicWall, у 2023 році переважали атаки cryptojacking на кінцевих пристроях, у яких використовувався легітимний інструмент XMRig із відкритим вихідним кодом [21]. Такі зловмисні програмні засоби поширюється через фішинг, шкідливу рекламу, уразливості, зламані програми тощо. XMRig призначений для майнінгу криптовалюти Monero (також відому як XMR, яку часто обирають

кіберзлочинці через її конфіденційність) на відносно високій швидкості, не потребуючи надмірної кількості системних ресурсів. Тому це рішення може бути використано на широкому спектрі пристроїв, від персональних комп'ютерів і смартфонів до підключених до Інтернету побутових та промислових систем. Хоча такий майнінг не споживає велику кількість обчислювальних потужностей, він все одно суттєво завантажує процесор, оскільки постійно виконується у фоновому режимі.

Найпоширеніше безфайлове програмне забезпечення cryptojacking, яке наразі зустрічаються, включає Lemon Duck, Purple Fox, GhostMiner, PCASTLE, Tor2Mine, WannaMine [23]. Ці зловмисні програми використовують такі техніки для зараження хостів та обходу антивірусних засобів [24]:

1. Lemon Duck, націлений на системи Windows, використовує скрипти PowerShell та експлуатує вразливості, які дозволяють виконувати поширення в мережі. Цей зловмисний програмний засіб використовує Mimikatz для збору облікових даних, adfind.exe для сканування активних каталогів та багато інших методів, таких як планування завдань, читання та зміна реєстру, підписки на WMI забезпечення постійної присутності в системі. Він може проникати в мережі, починаючи з одного зараження і швидко поширюючись всією інфраструктурою, щоб використовувати ресурси для майнінгу криптовалюти. Особливістю є те, що зловмисні скрипти містять змінну під назвою "$Lemon_Duck".

2. Purple Fox використовує вразливості Windows за допомогою руткітів і тактики ухилення, використовуючи інструмент Invoke-ReflectivePEInjection з відкритим вихідним кодом для рефлексивного (безфайлового) завантаження і виконання стиснутих виконуваних файлів за допомогою інструменту GZIP в процесах PowerShell, що працюють в оперативній пам'яті.

3. GhostMiner впроваджує шкідливий JavaScript у вебсторінки для майнінгу криптовалюти на комп'ютерах відвідувачів цих вебсторінок. GhostMiner використовує об'єкти WMI як безфайлову процедуру для підтримки присутності на хостах.

4. PCASTLE використовує PowerShell та легітимні інструменти для прихованого майнінгу на операційній системі Windows. WannaMine використовує вразливість EternalBlue, імітує зловмисне програмне забезпечення WannaCry і застосовує безфайлові методи для прихованого майнінгу.

Отже, одним із основних легітимних інструментів, які використовують безфайлові зловмисні криптомайнери є PowerShell. Цей інструмент застосовується для виконання таких дій [24]:

- виконання шкідливого коду;
- маскування шкідливої активності;
- створення додаткових процесів;
- викрадення системних облікових даних;
- віддалено завантажувати та виконувати довільний код;
- налаштування завдань за розкладом;
- налаштовувати функції безпеки з відключенням або виключенням;
- звільнення системних ресурсів для виконання криптомайнінгу;
- відключити будь-які конкуруючі засоби криптомайнінгу;
- повторне зараження інших систем у скомпрометованій мережі.

1.3 Методи виявлення та блокування cryptojacking

Існують різні способи виявлення та блокування cryptojacking, які можна відокремити в такі групи [17]:

1. На рівні мережі – дозволяє виявляти будь-які види атак cryptojacking.
2. На рівні хостів (host-based) – дозволяє виявляти автономні майнери, які запускаються як шкідливе програмне забезпечення на хостах.
3. Виявлення у браузерах – призначене для атак cryptojacking, які виконуються через скрипти на вебсайтах.

До методів виявлення cryptojacking у мережі відноситься фільтрування URL-адрес. Цей метод полягає у використанні URL-фільтрації для блокування доступу до вебсайтів, які містять шкідливі скрипти для майнінгу криптовалют. Перевагою цього

методу є простота реалізації та можливість блокування доступу до шкідливих ресурсів у всій мережі. Така фільтрація виконується на рівні фаєрвола та доступна на популярних рішеннях Next-Generation Firewall від виробників Cisco, Fortinet, Palo Alto Networks, Juniper Networks, Huawei, Sophos, Forcepoint, Checkpoint тощо [25]. Рішення з фільтрування URL-адрес надають декілька варіантів налаштування цієї функції, зокрема [26]:

- фільтрування на основі категорій, які формуються виробником фаєрвола та включають в себе відомі сервіси криптомайнінгу;
- фільтрування на основі репутації, яке передбачає аналіз ймовірності використання URL-адреси для цілей, які можуть суперечити політиці безпеки організації. Репутація варіюється від високого ризику (рівень 1) до добре відомого (рівень 5);
- фільтрування вручну дозволяє самостійно налаштувати URL-адреси для блокування, використовуючи власний перелік зловмисних сервісів.

Однак, цей метод фільтрування можна обійти шляхом створення великої кількості різних доменних імен. Зловмисники активно використовують тактику динамічно згенерованих доменних імен та проксі-серверів для уникнення виявлення та блокування фаєрволами [17]. Також трафік криптомайнінгу може бути замаскований як легітимний трафік, що передається через стандартні мережеві протоколи.

Мережеві методи виявлення також можуть включати в себе аналіз трафіку з метою ідентифікації аномалій, характерних для атак cryptojacking. До таких методів відноситься deep packet inspection (DPI) [27; 28], який дозволяє виконувати перевірку вмісту пакетів, а не лише їх заголовків. Таким чином визначається тип сервісу, до якого належить конкретний сеанс, та застосовуються правила, визначені адміністратором. Таким чином DPI може виявити приховані загрози в потоці даних, наприклад спроби викрадання даних, порушення політики використання мережевих сервісів, зловмисне програмне забезпечення включаючи cryptojacking. DPI використовує правила на основі сигнатур, а також евристичні та статистичні технології для визначення протоколів, тому правила будуть застосовуватися навіть

якщо трафік передається на нестандартних портах [29], враховуючи що протоколи криптомайнінгу не мають єдиного загальноприйнятого порту [28, с. 1]. Але цей метод має недоліки, адже аналіз вмісту великого обсягу трафіку потребує великої кількісті обчислювальних ресурсів [28, с. 1]. Також під час криптомайнінгу може застосовуватись шифрування трафіку з використанням TLS (transport layer security), що забезпечує захист зв'язку між сервісами майнінгу та клієнтами. Вміст пакетів не буде доступний для читання і системи виявлення зловмисного трафіку, які використовують технологію DPI, не зможуть ефективно ідентифікувати атаки cryptojacking [30].

Для ефективного виявлення та блокування атак cryptojacking окрім статичних списків блокування URL-адрес необхідно використовувати алгоритми аналізу трафіку разом із методами машинного навчання, що дозволяють виявляти аномальну та підозрілу поведінку в мережі. Наприклад, ці методи можуть бути використані для аналізу даних NetFlow, щоб визначити протоколи, які використовуються під час криптомайнінгу [28; 31] та виявити характерні показники трафіку, такі як розміри пакетів у потоках, проміжки часу між пакетами [11]. NetFlow – це протокол, який виконує аналіз мережі на рівні потоків, агрегуючи весь трафік, що має однакові IP-адреси джерела та призначення, а також порти TCP/UDP. Для виконання аналізу трафіку розраховуються значення його обсягу по відношенню до часу потоку – пакети/секунду, біти/секунду, а також середній розмір пакетів (біт/пакет). Окрім частоти передачі даних, трафік криптомайнінгу має ще одну особливість, яка полягає в тому, що він є асиметричним. У той час як сервер передає багато даних клієнтам, клієнт надсилає менший обсяг даних серверу. Тому під час аналізу трафіку за допомогою моделей машинного навчання також використовуються відношення вхідних пакетів до вихідних пакетів та вхідних байтів до вихідних байтів [28, с. 2]. Загальна схема трафіку криптомайнінгу на прикладі протоколу Stratum показана на рисунку 1.5.

Результати досліджень ефективності різних моделей машинного навчання показують точність більшу за 90%, а також окремі моделі, зокрема класифікаційне та регресійне дерево (classification and regression tree, CART), дерево рішень C4.5,

наївний баєсовий класифікатор, однокласова класифікація (one-class classification, OCC) мають точність 96-97% [28, с. 5; 31].

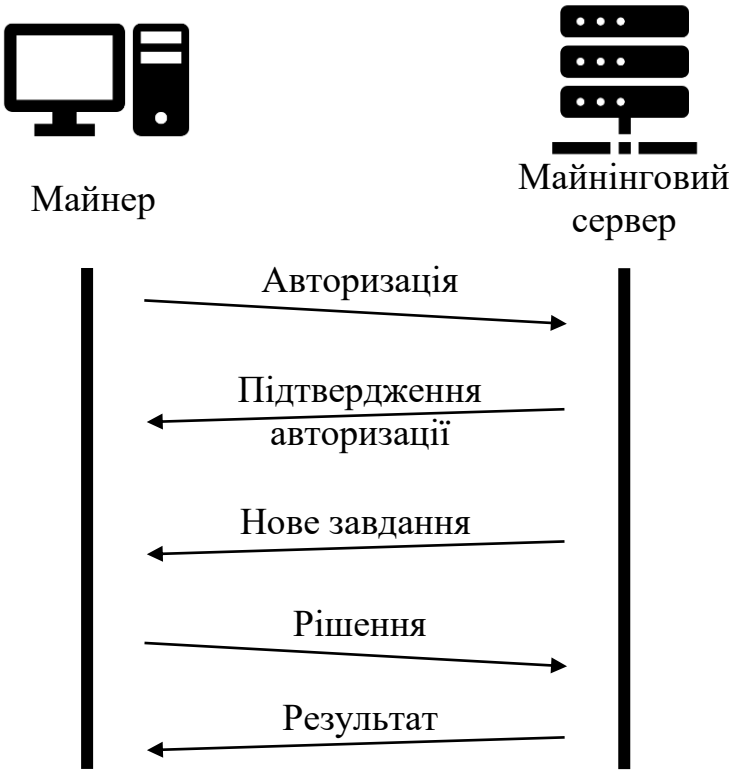

Рисунок 1.5 – Схема роботи протоколу криптомайнінгу

Методи host-based використовується для виявлення та блокування шкідливих дій на рівні окремих систем (хостів). Ці методи поділяються на такі категорії [32]:

- виявлення шкідливого програмного забезпечення на основі сигнатур;
- виявлення шкідливих програм на основі поведінки;
- евристичне виявлення шкідливого програмного забезпечення;
- виявлення шкідливого програмного забезпечення на основі перевірки моделей.

Також до окремих категорій можуть бути відокремлені методи виявлення атак cryptojacking у хмарі, на мобільних пристроях та на пристроях IoT [32].

До категорії виявлення на ріні хостів відноситься антивірусне програмне забезпечення та Endpoint Detection and Response (EDR). Зазвичай ці рішення використовують сигнатурні та поведінкові методи. Сигнатури являють собою

характеристику зловмисного коду, яка інкапсулює структуру програми та унікально ідентифікує кожне шкідливе програмне забезпечення [32, с. 5]. Сигнатури можуть включати в себе хеш виконуваних файлів, частини коду та інструкцій, текстові рядки або набір байтів, перелік функцій, які викликає програма та використані DLL-файли. Незважаючи на те, що ці методи досить швидкі та ефективні для створення підпису, вони дозволяють виявити лише вже відоме зловмисне програмне забезпечення, яке було проаналізовано та записано в базу даних антивірусних засобів, а також не є стійкими до методів обфускації. Наприклад, шкідливе програмне забезпечення може легко змінювати рядки та точки входу програми у своєму наборі інструкцій [32, с. 7].

Поведінковий метод виявлення дозволяє визначити функціональність зловмисного програмного забезпечення. Таким чином, навіть якщо послідовність інструкцій та хеш шкідливого програмного забезпечення можуть змінюватися, його функціональність залишиться однаковою. Індикатори поведінки включають в себе створення та зміну записів в реєстрі, відкриття мережевих з’єднань, тощо.

Однак, стандартні способи виявлення зловмисного програмного забезпечення, які використовують антивірусні програмні засоби, можуть бути недостатньо дієві для атак cryptojacking через поширене використання безфайлових технік. Тому для протидії таким тактикам застосовуються методи виявлення, які використовують засоби машинного навчання, що призначені для аналізу різних показників використання системних ресурсів [17]. Ці методи використовують алгоритми класифікації або кластеризації даних для ідентифікації аномалій, що характерні для атак cryptojacking. Зокрема, існує велика кількість досліджень, в яких застосовуються як легкі моделі машинного навчання k-nearest neighbors (k-NN), random forest (RF) [33, с. 2], так і моделі глибокого навчання, зокрема long short-term memory (LSTM) [33, с. 2], згорткова нейронна мережа (convolutional neural network, CNN) [13; 17; 34], рекурентна нейронна мережа (recurrent neural network, RNN) [35] тощо.

Один із параметрів системи, що аналізується моделями машинного навчання, є ресурси процесора. Зокрема, аналізуються окремі показники використання CPU, такі як споживання, продуктивність, використання і час роботи [8, с. 8]. Хоча такі показники можуть характеризувати роботу програм, які виконують криптомайнінг,

деякі легітимні програми також можуть бути ресурсоємними та спричиняти значні зміни у показниках використання процесора, спричиняючи помилкові спрацювання алгоритмів виявлення cryptojacking. Тому додатково аналізуються виклики системних функцій (syscalls), які виконують програми [36, с. 4], та показники використання оперативної пам’яті процесором [8, с. 10]. Ці показники можуть отримуватись використовуючи дані лічильників продуктивності [33]. Лічильники продуктивності операційної системи Windows надають інформацію про стан компонентів від рівня додатків до апаратного рівня, тобто цілісний стан системи. Вони являють собою невеликі програми, які підраховують, відстежують і вимірюють події в системі. Якщо одночасно запущено кілька сервісів або додатків, тоді лічильники надають інформацію про контекст роботи, тобто як саме змінилася інформація лічильника після запуску або завершення роботи додатків. Ці дані містять інформацію з чотирьох основних системних об'єктів (кожен з яких має кілька лічильників продуктивності): логічні диски, процесори, фізичні диски та пам'ять [33, с. 5]. Крім того, для збільшення точності виявлення показники на рівні окремих хостів можуть розглядатись разом із мережевими показниками NetFlow [37].

Загалом ефективність методів виявлення атак cryptojacking на основі машинного навчання є досить високою. Точність виявлення у більшості досліджень становить більше 90%, окремі методи досягають точності 97-98% [8; 36; 37]. Натомість, недоліком систем, які використовують машинне навчання та динамічний аналіз, що постійно працюють у фоновому режимі, є їх негативний вплив на продуктивність системи кінцевих користувачів [17].

Методи виявлення атак cryptojacking на рівні браузера включають в себе чорні списки доменів, які задіяні у несанкціонованому майнінгу криптовалют [17, с. 1]. Такі системи виявлення являють собою розширення для браузерів, зокрема до них відносяться NoCoin та MinerBlock, та містять функціонал, що ідентичний фільтруванню URL на фаєрволі. Ці засоби можуть бути реалізовані у вигляді доповнення браузера [38]. Загальний рівень виявлення за допомогою чорних списків є невисоким, адже зловмисники постійно створюють нові домени. Середній показник виявлення становить близько 50% [22, с. 9].

Для виявлення атак cryptojacking у браузері може виконуватись аналіз скриптів та коду, який завантажується на сайтах. Динамічний аналіз поведінки таких виконуваних файлів у браузерах відбувається на основі показників використання процесора, пам’яті, мережі. Рішення, що засноване на такому методі аналізу з використанням згорткових нейронних мереж, показало частку виявлення 93.8% [13, с. 8]. Також виконуваний код, зокрема, WebAssembly (Wasm), що є двійковим форматом інструкцій, може бути перетворений на зображення у відтінках сірого і класифікований з використанням згорткових нейронних мереж [17]. Згідно проведених досліджень, цей метод може забезпечити виявлення із точністю 98.97% [17, с. 3], а також він може бути стійкий до обфускації [17, с. 12]. Перевагою такого аналізу на відміну від динамічного є те, що він не вимагає постійного моніторингу подій процесора, пам'яті та мережі або підрахунку запущених інструкцій, тому він використовує набагато менше системних ресурсів. Ще один варіант аналізу скриптів у браузері – це виявлення певних викликів API, що надають доступ до CPU, GPU, пам’яті, та мережі [39]. Крім того, аналізуючи код може виконуватись ідентифікація криптографічних функцій, що застосовуються у криптомайнінгу [40, с. 11]. Наприклад, це може бути реалізовано за допомогою символічного виконання, що передбачає заміну конкретних значень у програмі символьними параметрами та імітує виконання програми так, щоб усі змінні містили символьні вирази. Таким чином визначаються особливості криптографічних алгоритмів з булевими формулами, які згодом використовуються як сигнатури для виявлення криптографічних алгоритмів в обфускованому двійковому коді [41, с. 15].

Окремо можуть бути виділені підходи до виявлення cryptojacking, що використовують особливості конкретного обладнання та систем, зокрема IoT. Використання показників використання системних ресурсів може бути недоступним на таких пристроях, оскільки більшість засобів IoT не дозволяють запрограмувати їх на збір цих характеристик [42, с. 13]. Тому для виявлення необхідно використовувати мережеві методи, що враховують сценарії атак, характерні для IoT. Дослідження таких способів виявлення також включають в себе аналіз трафіку в мережі зі змішаними типами пристроїв, що характерно для реальних ситуацій [42, с. 5].

Засоби виявлення атак cryptojacking можуть бути націлені на особливості хмарної інфраструктури, зокрема, віртуалізацію, що може спричинити шум в показниках використання системних ресурсів. Приклад такого інструмента є MineGuard, який реалізований на рівні гіпервізора, що не дозволить його скомпрометувати з віртуальної машини, доступ до якої може мати зловмисник [43, с. 3].

Отже, є досить широкий спектр засобів, які призначені для виявлення атак cryptojacking, тому важливо обрати правильний інструмент, який забезпечить такий рівень зменшення збитків, що буде перевищувати інвестиції. Якщо ж інвестиції будуть більшими за збитки, що можуть бути спричинені в результаті атаки, тоді такий засіб не буде виправданим для даного випадку.

1.4 Методи прогнозування та оцінки ризиків

Для створення стратегії заходів захисту інформаційної системи від кібератак використовуються методи оцінки ризиків, які дозволяють передбачити можливі втрати у випадку здійснення кіберзагроз та визначити оптимальний бюджет для впровадження заходів з кібербезпеки. Ці методи допомагають ідентифікувати потенційні загрози, визначити їх імовірність та вплив, а також розробляти ефективні стратегії реагування на кібератаки.

Відповідно до ISACA CMMI, ризик – це потенційна невизначена подія, яка може завдати шкоди або негативно вплинути на досягнення мети [44]. У контексті безпеки IT-систем ризик IT-систем є загальною мірою ймовірності та серйозності ситуації, у якій певна загроза використовує певну слабкість, спричиняючи втрату або пошкодження системних активів, отже, непрямі чи прямі збитки для організації [45].

Ризик визначається як результат поєднання ймовірності та впливу [46] та виражається за формулою:

$$R = I * E, \tag{1.1}$$

де I – ймовірність виникнення ризику, E – вплив ризику.

Водночас ймовірність включає в себе загрозу та вразливість і розраховується за формулою [47]:

$$I = T * V, \tag{1.2}$$

де T – рівень загрози, V – рівень вразливості.

Процес управління ризиками призначений для визначення, аналізу та реагування на ризики, які потенційно можуть поставити під загрозу заявлені цілі та завдання підприємства. Оцінка ризиків є невід'ємною частиною цього процесу та основним підходом для розуміння ризику та його ефективного управління. Процес управління ризиками згідно ISACA Risk IT Framework [48, с. 32] показано на рисунку 1.6.

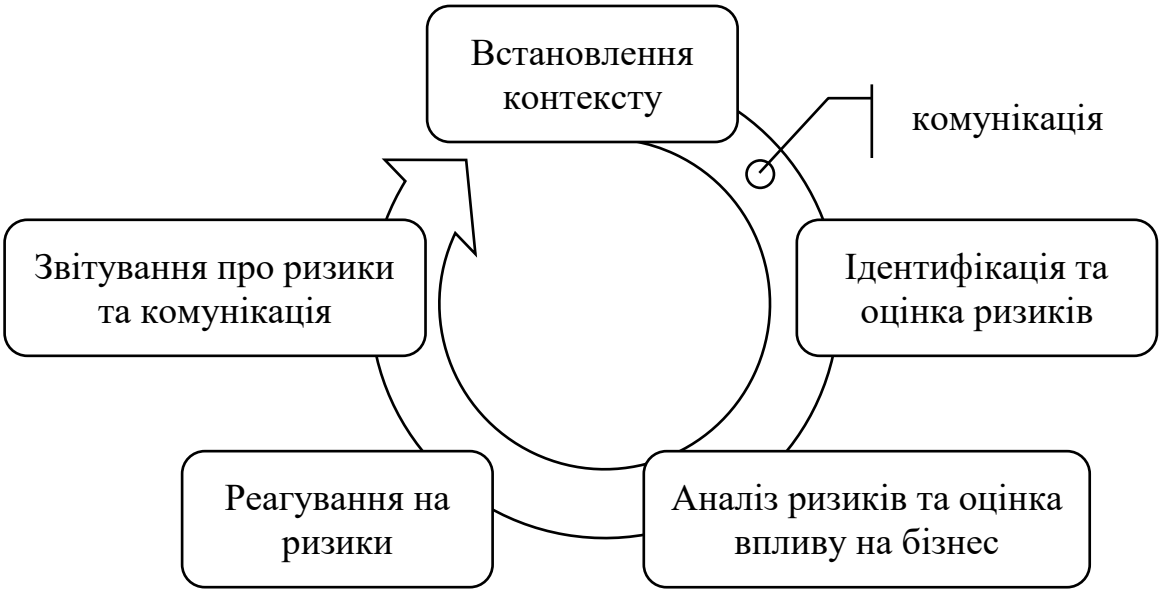

Рисунок 1.6 – Процес управління ризиками

Оцінка ризиків включає в себе якісні та кількісні методи. Якісні методи зазвичай базуються на експертній оцінці та не мають чітких числових значень. Вони класифікувати ризик на основі одного сценарію або групи сценаріїв як більший, ніж для іншого сценарію або групи сценаріїв. Коли всі сценарії з системи включено до

рейтингу, такий рейтинг може бути зроблений лише суб’єктивно. Зокрема, до якісних методів оцінки ризиків належать SWOT, CORAS, CRAMM, FRAP, OCTAVE, RiskWatch [49].

Компонентами якісних методів оцінки ризиків є матриця ризиків та опитувальники [50]. Прикладом якісної оцінки ризиків є побудова матриці осі якої є порядковими шкалами, і рівень ризику або рейтинг для кожного ризику розраховується як добуток імовірності та вплив [51]. Шкала оцінки може складатись із трьох значень – «низький», «середній», «високий» (таблиця 1.1). Після побудови матриці ризики розміщуються у відповідній клітинці залежно від оцінки їх ймовірності та впливу. Такий процес оцінки дозволяє ідентифікувати ті ризики, які потребують негайної уваги та ресурсів для їх мінімізації.

Якісні методи легкі для розуміння та застосування, але вони не надають можливість провести аналіз витрат і вигоди, оскільки ймовірність втрат не ґрунтується на точних числових значеннях.

Таблиця 1.1 – Матриця ризиків

		Ризик		
Ймовірність	висока	середній	високий	критичний
	середня	низький	середній	високий
	низька	низький	низький	середній
		низький	середній	високий
	Вплив			

З іншого боку, кількісні методи використовують конкретні числові значення для розрахунку ймовірностей виникнення ризиків та їхніх можливих наслідків. Такі методи значно підвищують обізнаність про IT-ризики з точки зору бізнесу та фінансів [48, с. 9]. Вони надають точні кількісні дані, що дозволяє більш об'єктивно оцінити ризики та приймати рішення згідно проведеного аналізу. Зокрема, на основі розміру збитків від атак, що визначений за кількісним методом, керівництво компанії може пріоритезувати певні напрямки забезпечення кібербезпеки та сформувати бюджет.

Підходи до аналізу та оцінки ризиків можуть бути [52, с. 7]:

- орієнтованим на загрози;
- орієнтований на активи або вплив;
- орієнтований на вразливість.

Зокрема, орієнтований на загрози підхід починається з ідентифікації джерел загрози та загрозливих подій і зосереджується на розробці сценаріїв загрози. Уразливості ідентифікуються в контексті загроз, вплив визначається на основі визначених цілей зловмисника.

Для оцінки впливу конкретних атак, зокрема, cryptojacking, варто використовувати кількісні методи оцінки ризику, адже вони забезпечують більшу конкретність та об'єктивність у визначенні потенційного впливу цих атак. Цей тип оцінки найкраще підходить для аналізу розміру інвестицій та вигоди для різних варіантів реагування на ризики або напрямків дій [52]. Атаки Cryptojacking можуть впливати на різні аспекти систем, включаючи використання ресурсів, ефективність роботи системи, економічний вимір та інші фактори, які краще виражаються числовими показниками. Кількісна оцінка надає можливість точніше визначити ймовірність та розмір фінансових збитків, а також забезпечує базу для обґрунтованого прийняття рішень.

Існують деякі базові показники, які використовуються в кількісній оцінці ризику. Однією із складових такої оцінки ризиків є концепція очікуваних річних збитків (annual loss expectancy, ALE). ALE для окремого типу події безпеки можна обчислити як добуток очікуваної одноразової втрати (single loss expectancy, SLE) і річної частоти виникнення (annual rate of occurrence, ARO) [53, с. 2]. SLE представляє ресурси, які очікується втратити, якщо інцидент трапиться один раз, ARO означає, скільки разів протягом річного інтервалу, як очікується, відбудеться інцидент. Відповідно, ALE виражається за формулою:

$$ALE = SLE * ARO. \quad (1.3)$$

ALE можна використовувати для обґрунтування вартості застосування контрзаходів для захисту активів або процесів. Зокрема, ALE надає можливість визначити розмір повернення інвестицій в кібербезпеку використовуючи метрику return on security investment (ROSI), що розраховується за формулою [53, с. 3]:

$$ROSI = ALE_0 - ALE_1 - \text{вартість}, \quad (1.4)$$

де $ALE_0 - ALE_1$ – зміна очікуваних річних збитків після інвестицій, $ALE_0 -$ очікуваний річний збиток до інвестицій в безпеку, $ALE_1 -$ очікуваний річний збиток після інвестицій в безпеку, вартість – інвестиції в засоби захисту.

ROSI може бути використано для підтримки рішення на користь або проти певного заходу безпеки. Якщо ROSI є позитивним, тоді інвестиція вважається вигідною, інакше витрати перевищують вигоду і таке рішення не варто впроваджувати [53].

Також існує інший спосіб розрахунку ROSI, що показує відношення цінності засобів захисту щодо показника зменшення ризику до їх вартості, та визначається за формулою [53, с. 3]:

$$ROSI = \frac{\text{Збиток від ризику} * \%\ \text{Зменшення ризику} - \text{Вартість рішення}}{\text{Вартість рішення}}. \quad (1.5)$$

Результат такої метрики є не грошовою одиницею, а співвідношенням, яке виражається у відсотках. Це представлення має такі переваги [53, с. 3]:

- дозволяє порівнювати різні заходи безпеки;
- організація може побачити, наскільки ефективно використовується її капітал, оскільки інвестиції в кібербезпеку можна порівняти з іншими інвестиційними проектами.

Більш розширені методи включають в себе розрахунки ймовірності та впливу. Серед методів кількісної оцінки ризиків, що застосовуються в кібербезпеці можна виділити такі:

- графічні методи, зокрема, fault tree analysis (FTA), event tree analysis (ETA), attack tree analysis (ATA);
- статистичні методи, зокрема аналіз баєсових мереж, метод моделювання Монте-Карло;
- метод аналізу чутливості (sensitivity analysis);

Графічні методи, засновані на моделі, використовують візуальні техніки, такі як FTA, ETA, аналіз дерева атак, аналіз дерева вразливостей і теоретико-системний аналіз процесів для представлення систем. Це логічні методи, які системно описують шляхи всередині системи для виявлення та класифікації відхилень. Ці графічні моделі ризику дуже ефективні в менших масштабах [54, с. 11]. Хоча всі ці методи спрямовані на аналіз подій та визначення ризиків у системах, вони мають різні підходи та особливості, що робить їх корисними для використання в різних контекстах та умовах.

Аналіз дерева відмов (FTA) – це графічна модель, яка представляє за допомогою символьних логічних операцій причинно-наслідкові зв’язки між комбінаціями подій, що призводять до визначеної основної небажаної події (рис. 1.7). FTA містить лише ті дії, які спричиняють кінцеву подію та може бути створена на основі допомогою кількісних або якісних методів. Кількісна FTA визначає ймовірність події на кожному кроці, що поширюється до кінцевої (найвищої) події для обчислення загальної ймовірності її виникнення. Незважаючи на те, що FTA є дещо ресурсомістким процесом, такий метод корисний для виявлення окремих точок збою, а також вразливостей і визначення потенційних засобів пом’якшення [54, с. 11].

Дерево атак (ATA) є різновидом FTA, у якому атака є головною подією, а не загальна системна помилка або аварія. Це дерево допомагає визначити потенційні шляхи атак та встановити заходи безпеки для їх запобігання. У дереві атак аналітики визначають шляхи, якими могли б слідувати зловмисники на основі відомих тактик, технік та процедур, і оцінюють імовірність виконання цих дій для певної атаки. Дерева атак корисні для виявлення слабких місць, однак їх важко використовувати у великих системах або установках через їх складність. Графіки атак схожі на дерева

атак, але використовують інший візуальний формат для позначення точок входу, точок виходу, вузлів і шляхів атаки [54, с. 12].

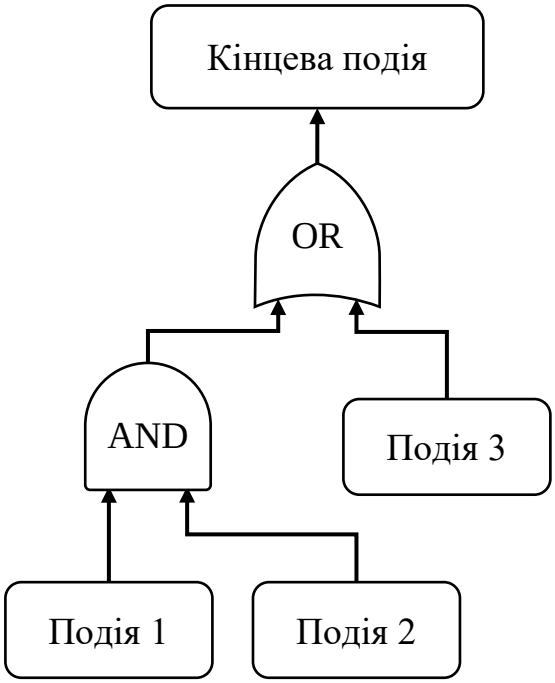

Рисунок 1.7 – Дерево відмов

Подібно до FTA, дерева вразливостей (vulnerability trees) – це підходи, які описують події зверху вниз, що разом компонують зв’язок між головною вразливістю та послідовністю вразливостей, які супротивник повинен використати, щоб досягти вершини. Дерева вразливостей допомагають інформувати сценарії атак, якими може скористатися противник. Однак, як і дерева атак, дерева вразливостей складні, і їх важко використовувати у великих системах [54, с. 12].

Ще один метод – event tree analysis (ETA). У той час як FTA є підходом зверху вниз, ETA є підходом знизу вгору (рис. 1.8). Він також є графічною логічною моделлю, яка, починаючи з початкової події, ідентифікує послідовність подій, що поширюються, та які призводять до декількох кінцевих небажаних подій або втрат. ETA може використовувати якісні або кількісні методи, має чіткий порядок від початку до кінця та може враховувати засоби пом’якшення. Однак ETA є складним, ресурсомістким і потребує нового дерева для кожної початкової події [54, с. 12].

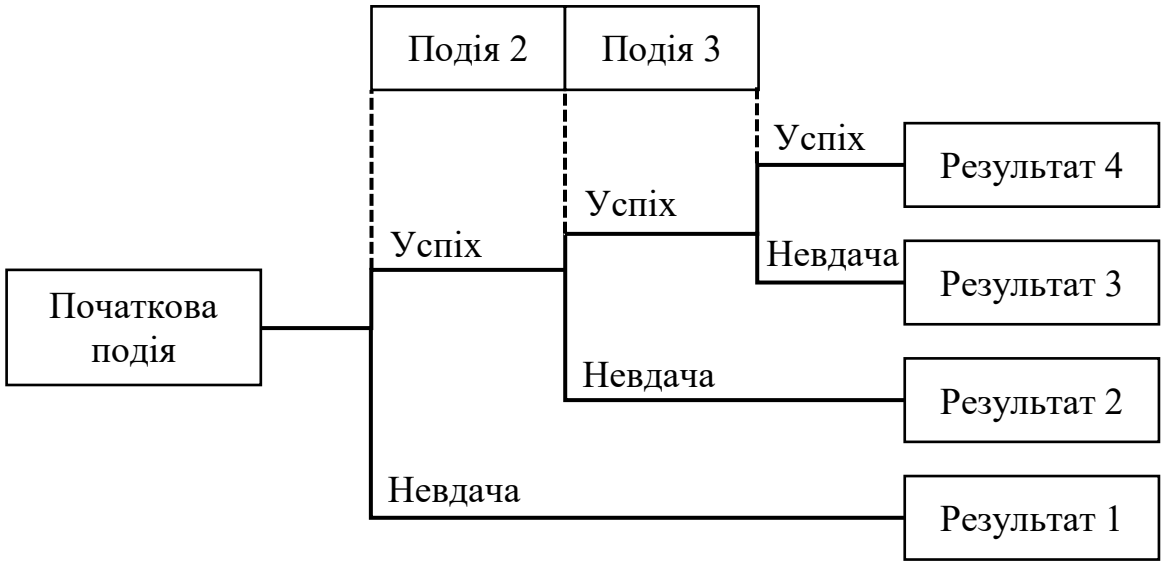

Рисунок 1.8 – Дерево подій

Під час застосування графічних моделей, зокрема АТА, для кількісної оцінки ризиків кібербезпеки створюється дерево ризиків шляхом визначення кроків, необхідних зловмиснику для досягнення мети. Ці кроки мають атрибути, за якими розраховується оцінка ризику. Такими значеннями є частота, вразливість і величина, які можуть бути визначені на основі історичних даних або експертної оцінки. [55, с. 7]. Перші вузли на шляху атаки утворюють «поверхню атаки», яка піддається постійним впливам від зловмисників (спробам атак). На цьому етапі необхідно визначити частоту цих впливів і відповідну вразливість (ймовірність проходження атаки). Потім атака продовжується у вигляді проміжних вузлів, яким потрібен лише показник вразливості, але вони вже можуть включати вплив. Останні вузли являють собою фактичну мету, яка визначає фактичні втрати.

Розрахунок ризиків за деревом атак складається із таких кроків [55, с. 8]:

1. Розділення дерева ризику на окремі шляхи від усіх вузлів входу до вузлів із визначеною величиною втрат.

2. Для кожного шляху: частоти атаки вхідного вузла множаться на вразливість усіх проміжних вузлів, доки не буде досягнуто цільового вузла.

3. Обчислення результуючого ризику у цільовому вузлі, помноживши отриману частоту на вплив.

4. Застосування розрахованого ризику до всіх вузлів і ребер на поточному шляху для відображення розміру втрат.

Приклад розрахунку ризиків за деревом атак зображено на рисунку 1.9 [55, с. 10]. На цьому дереві показано вхідні вузли зі значеннями частоти 5, 10, 1, а також значенням вразливості 50% значення ймовірності позначені як «f», значення вразливостей – «v» та значення впливу – «m». Проміжні вузли мають значення вразливості також 50%, кінцевий вузол має значення вразливості 50% та значення впливу – 1 000$. Результуючий ризик згідно такого дерева складає 2 062,5$.

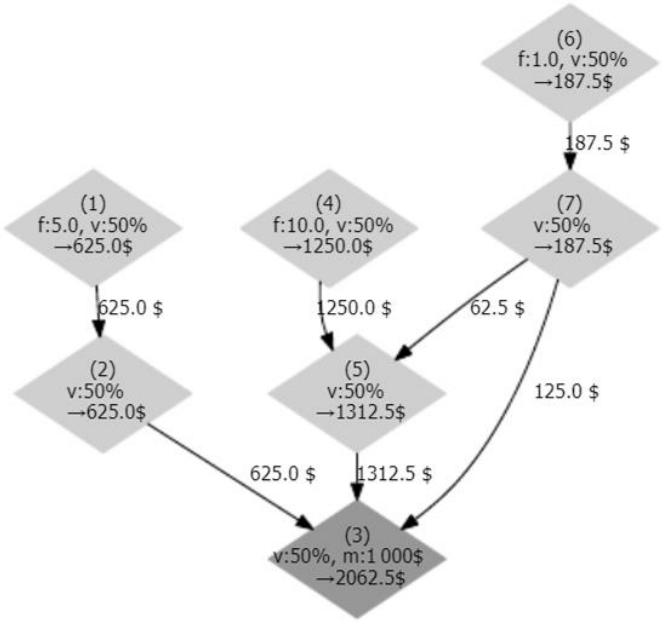

Рисунок 1.9 – Приклад дерева ризиків з різними шляхами атаки

Статистичні методи забезпечують найбільш точну оцінку ступеня ризику, але застосовні при наявності досить повної і достовірної інформації [56]. Одним із них є симуляція Монте-Карло, що проводиться для створення розподілу ймовірностей, що показує можливий діапазон втрат протягом даного періоду часу [57, с. 469]. Метод Монте-Карло виник у 20-му столітті. Незважаючи на це, на сьогоднішній день він вважається одним з найдосконаліших методів. Його широке застосування є результатом того, що цей метод можна просто адаптувати до поточних умов, а також завдяки зручності реалізації в сучасних програмних засобах [58]. Цей метод дозволяє виконати числовий аналіз ризиків, враховуючи різні фактори та їх взаємозв'язок, а

також дозволяє застосувати перелік історичних ризиків до всього активу. Метод Монте-Карло використовує випадкову вибірку та статистичне моделювання для оцінки математичних функцій та імітації роботи складних систем. Побудова моделі за цим методом починається з визначення функціональних залежностей у реальній системі. Після чого можна одержати кількісний розв'язок, використовуючи елементи теорії ймовірності та генератори випадкових чисел.

Симуляція Монте-Карло дозволяє врахувати невизначеність шляхом імітації випадкових варіацій станів системи і робочих параметрів, зокрема у динамічних системах [59, с. 6]. Тому такий метод особливо корисний у моделюванні атак, що є випадковими процесами. Крім того, метод Монте-Карло може бути поєднаним з іншими методами кількісної оцінки ризиків, зокрема із деревами подій та відмов [59, с. 6, 10].

Баєсові мережі, названі на честь теореми Баєса, є також інструментом моделювання з урахуванням невизначеності, що засновані на залежностях між різними змінними. Вони використовують принципи теорії ймовірностей для представлення набору змінних та їхніх умовних залежностей через спрямований ациклічний граф, у якому вузли представляють випадкові змінні, а ребра (стрілки) відображають причинно-наслідкові зв'язки між змінними. Кожен вузол має умовну ймовірність, яка кількісно оцінює вплив однієї змінної на іншу, що дозволяє моделювати складні взаємозалежності та невизначеності в реальному світі.

У контексті кількісної оцінки ризиків, баєсові мережі застосовуються для моделювання та аналізу ризиків, що дозволяє оцінити ймовірність та потенційний вплив різних подій на проєкти, системи або процеси. Використання баєсових мереж допомагає визначити, як зміни в одній частині системи можуть впливати на інші її частини, тим самим надаючи можливість проводити комплексний аналіз ризиків. Це особливо корисно у ситуаціях, коли доступна інформація неповна або нечітка, оскільки модель дозволяє врахувати невизначеність і надати більш обґрунтовані оцінки ризиків. Зокрема, баєсові мережі зазвичай використовуються для оцінки ризиків загроз і вразливостей мереж та забезпечують кількісну і якісну оцінку ризиків [60, с. 6]. Також вони можуть бути застосовані для аналізу дерева відмов щоб точно

розрахувати надійність дерева відмов за наявності статистичних залежностей між відмовами. У цьому випадку баєсова мережа виражається як послідовність стохастично залежних випадкових змінних $X_1, X_2, \ldots, X_n$, де X_i може залежати від X_j лише якщо $j < i$, тобто розподіл відмов залежить лише від розподілів відмов попередніх подій. Важливою особливістю аналізу баєсової мережі є те, що цей метод не лише надає можливість обчислити ймовірність верхньої події, заданої вузлами, але й обчислити ймовірності кожного з вузлів, заданого верхньою подією [61, с. 11].

Додатково можна виділити метод аналізу чутливості, що дозволяє виміряти, наскільки «чутливою» є модель до змін значень її керуючих параметрів. Він може бути застосований під час аналізу захищеності від атак промислових систем для визначення чутливих змінних до конкретних подій, наприклад, сигналів управління, або кібератак [62]. Таким чином може бути ідентифіковано ті параметри, які мають найбільший вплив та спрямовувати увагу на них під час проєктування систем виявлення вторгнень.

Однією із методологій, що описують застосування кількісної оцінки ризиків є система факторного аналізу інформаційних ризиків (factor analysis of information risk, FAIR), яка може бути застосована до багатьох ситуацій оцінки ризиків і загроз завдяки своїм ефективним і водночас простим практичним рекомендаціям [63, с. 2]. Також ця методологія обрана як міжнародний стандарт консорціумом The Open Group [64]. FAIR є поєднанням таксономії FAIR та статистичних методів. Ризик згідно FAIR являє собою фінансові втрати, що визначаються частотою подій (loss event frequency, LEF) та величиною втрат (loss magnitude, LM). LEF визначається як частота, з якою зловмисник буде завдавати шкоди інформаційному активу протягом певного періоду часу, і сама є функцією частоти подій загрози (threat event frequency, TEF) та вразливості. Методологія FAIR може бути застосована разом із такими методами, як симуляція Монте-Карло та баєсовими мережами. Зокрема, симуляція Монте-Карло виконується для проведення агрегації ризиків без використання додаткових методів прогнозування, що дозволяє уникнути внесення послідовної неточності [65, с. 6].

1.5 Аналіз вхідних даних для прогнозування та оцінки ризиків

На основі аналізу методів кількісної оцінки ризиків вхідні дані поділено на дві групи:

- значення ймовірності атак;
- значення впливу (втрат) від атак.

По-перше ймовірність виконання атак може бути визначено за історичними даними про попередні атаки, зокрема, інформацію про частоту, інтенсивність, тривалість та заходи реагування на них. Ці дані можуть міститись у журналах операційного центру кібербезпеки (security operations center, SOC), або його аналогу в організації, додатково така інформація міститься у звітах threat intelligence, публікаціях аналізу загроз. Зокрема, серед досліджень атак cryptojacking було визначено, що у 2018 році близько 0,011% всіх доменів в інтернеті активно виконували криптомайнінг без дозволу користувачів, із них 0,065% вебсайтів із списку Alexa Top 1M, що є найбільш часто використовуваними сайтами [16]. За іншими даними, у 2022 році серед 300 тисяч найбільш популярних вебсайтів несанкціонований майнінг виконували 1 813 [66, с. 1]. Крім того, у 2023 році серед вибірки сайтів розміром 50 000 було виявлено 42 вебсторінки, що виконують атаки cryptojacking [67, с. 10].

Також враховуються дані про відомі вразливості в системі, невстановлені патчі, вразливі програмні компоненти. Така інформація отримується за результатами сканування вразливостей, тестування на проникнення та діагностики. Окрім вразливостей під час оцінки ризиків також може бути врахована інформація про наявність легітимних інструментів, що можуть бути використані зловмисниками для виконання атаки. Аналіз цих даних дозволить визначити, які атаки можуть бути найбільш ймовірними та які частини системи потребують найбільшої уваги під час формування сценаріїв атак.

До вхідних даних, на основі яких формується ймовірність виконання атак також відноситься інформація про впроваджені заходи захисту. Зокрема, це загальні рішення для захисту від атак, такі як антивірусні програми, фаєрволи, моніторинг

мережі тощо, так і спеціальні засоби виявлення і блокування несанкціонованого майнінгу. Аналіз цих засобів допоможе визначити, ймовірність їх обходу зловмисником на основі виконаних досліджень, статистичних даних, діагностики методом експертного аналізу.

Під час формування вхідних даних для оцінки ризиків використовується інформація про технічні характеристики системи для формування значень збитків. Ці дані включають в себе деталі про конфігурацію апаратного забезпечення, зокрема, процесор та відеокарта, на основі яких може бути розраховано витрати на електроенергію, а також їх вартість. Додатково можуть враховуватись інші компоненти системи, що можуть вийти із ладу через перенавантаження для розрахунку витрат на їх заміну.

Для формування розміру збитків також застосовуються дані про використання ресурсів систем під час майнінгу. До них відносяться статистична інформація про навантаження на центральний процесор (CPU), графічний процесор (GPU), на основі досліджень та звітів попередніх атак cryptojacking, а також характеристик інструментів, які виконують майнінг. Зокрема, згідно досліджень, більшість майнінгових скриптів на вебсайтах використовують близько 25% CPU, що допомагає уникнути виявлення, але деякі сторінки можуть використовувати 100% ресурсів [68, с. 6]. Хоча таке споживання може збільшити прибутки зловмисника, це призведе до швидкого виявлення несанкціонованої активності.

Значну частину збитків становить втрачений час обчислювальних ресурсів, а також співробітників, що працюють із цими системами. Несанкціоноване використання ресурсів призводить до сповільнення роботи як самого пристрою, так і його користувачів. Тому додатковими вхідними даними є вартість часу співробітників компанії та відсоток того, наскільки атаки cryptojacking впливають на продуктивність, що напряму залежить від частки використання системних ресурсів під час майнінгу.

Під час комплексної оцінки ризиків, що включає різні атаки, додатково вхідними даними можуть бути ймовірні збитки через виконання інших атак, які супроводжують cryptojacking. Зокрема, під час аналізу цих атак було визначено, що

зловмисник може отримати повний доступ до системи користувача та виконувати інші несанкціоновані дії, наприклад, викрасти дані для входу в облікові записи, виконувати розсилку фішингових листів та спаму, DDoS-атаки на інші системи, та навіть шифрувати файли із вимогою викупу.

Висновки за розділом 1

У першому розділі було проаналізовано атаки cryptojacking, які передбачають несанкціонований майнінг криптовалют, розкриваючи основні механізми та методи, які використовують зловмисники. Такий вид кіберзлочинності спричиняє фінансові збитки жертвам та може бути націленим на різні пристрої. Було проаналізовано різні види зловмисного програмного забезпечення та техніки, що застосовуються для обходу антивірусних заходів.

Розглянуто методи виявлення та блокування cryptojacking, включаючи статичний та динамічний аналіз мережевого трафіку, а також аналіз поведінки виконуваних файлів і скриптів. Різні підходи розроблені для різних пристроїв та сценаріїв атак, зокрема для браузера, або для усієї системи чи мережі. Вони забезпечують різну точність виявлення, тому необхідно правильно спланувати стратегію захисту застосовуючи методи управління та оцінки ризиків.

Було проаналізовано способи оцінки ризиків та запропоновано використовувати для атак cryptojacking саме кількісні методи, які дозволяють визначити розмір фінансових збитків та спланувати інвестиції. Також проведено аналіз та визначено ключові параметри вхідних даних для розробки моделей прогнозування та оцінки ризиків таких атак.

РОЗДІЛ 2
РОЗРОБКА МОДЕЛІ ОЦІНКИ РИЗИКІВ АТАК CRYPTOJACKING

2.1 Алгоритм оцінки та прогнозування ризиків атак cryptojacking

Для зменшення збитків від атак cryptojacking необхідне створення стратегії заходів захисту інформаційної системи. Для цього використовуються методи оцінки ризиків, які дозволяють передбачити можливі втрати у випадку здійснення кіберзагроз та визначити оптимальний бюджет для впровадження заходів з кібербезпеки. Виконуючи оцінку ризиків атак cryptojacking недостатньо лише визначити якісні показники, адже вони спричиняють конкретні фінансові збитки, розмір яких залежить від кількості та потужності пристроїв, часу виконання атаки. Тому запропоновано використовувати саме кількісні методи, які дозволяють визначити влив цих атак для планування відповідних заходів захисту та обґрунтування розміру фінансування.

На основі дослідження існуючих методів оцінки ризиків кібербезпеки було обрано графічний метод аналізу дерева атак для розрахунку ймовірності атак cryptojacking, а також метод Монте-Карло для розрахунку кінцевих результатів оцінки ризиків, зокрема, прогнозування можливих втрат шляхом формування випадкових значень збитків. Схема розробленого алгоритму показана на рисунку 2.1.

Першим етапом алгоритму оцінки ризиків атак cryptojacking є визначення та аналіз можливих сценаріїв атак. Цей процес передбачає визначення потенційних векторів атак, які можуть бути використані зловмисниками для виконання несанкціонованого криптомайнінгу. Для кожного варіанту атаки розробляється відповідне дерево атак, що дозволяє візуалізувати можливі шляхи дій зловмисників для досягнення кінцевої мети – виконання майнінгу криптовалюти на обчислювальних потужностях жертви. Результатом цього етапу є структурований перелік сценаріїв атак з відповідними деталями, які стануть основою для подальшої кількісної оцінки ризиків. Визначення сценаріїв атак є критичним для забезпечення

ефективності подальших етапів алгоритму, оскільки дозволяє зорієнтуватися в потенційних загрозах і націлитись на мінімізацію конкретних ризиків, пов'язаних з cryptojacking.

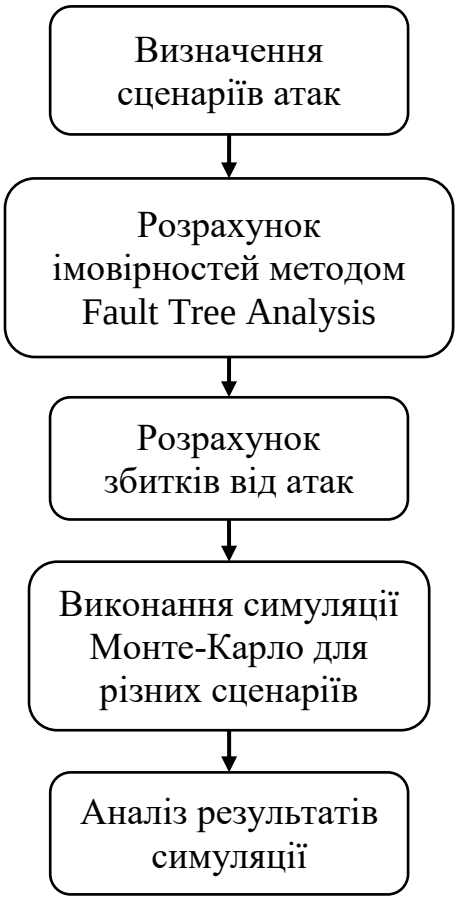

Рисунок 2.1 – Схема алгоритму оцінки та прогнозування ризиків атак cryptojacking

Наступним кроком є розрахунок ймовірностей визначених сценаріїв, що будуть використані для виконання симуляції Монте-Карло. Ймовірності визначаються на основі історичних даних про атаки, а також досліджень ефективності систем захисту. Кінцеве значення ймовірності сценарію визначається за методом дерева атак та залежить від імовірностей успішного виконання кожного етапу згідно сценаріїв атаки. Важливим компонентом є визначення ймовірності сценаріїв як для поточного стану кібербезпеки, так і для майбутнього після впровадження нових засобів захисту.

Далі виконується розрахунок впливу атак, тобто розмір втрат, що будуть спричинені внаслідок успішного виконання атаки. Для розрахунку впливу використовуються дані про види та кількість обладнання, його потужність, ціна

електрики, вартість простою обладнання через його використання не за призначенням. Необхідно визначити мінімальний та максимальний розмір втрат. Мінімальні втрати включають в себе вартість електрики, яка використана для виконання майнінгу, а також вартість втраченого часу відповідних обчислювальних потужностей, що міг бути використаний для виконання задач відповідно до цільового призначення обладнання. Максимальні втрати додатково включають в себе збитки від виведення з ладу обладнання та його заміни.

Після визначення усіх вхідних даних виконується симуляція Монте-Карло, яка на основі цих даних генерує розмір втрат у випадку успішного виконання атаки на основі випадкових чисел. Ця симуляція має велику кількість ітерацій, результатом яких є перелік різних розмірів втрат за кожним із сценаріїв атаки, а також для поточного і майбутнього стану захищеності організації.

На основі визначеного переліку втрат будується крива втрат (loss exceedance curve, LEC), яка показує прогнозовану ймовірність для кожного значення збитків. Також визначається різниця між ризиком для поточного стану зрілості засобів захисту та майбутнього. Таким чином визначається вигода від впровадження нових рішень безпеки, яка порівнюється із їх вартістю використовуючи метрику ROSI.

2.2 Побудова сценаріїв атак

Сценарії атак cryptojacking передбачають два основних варіанти – атаки на кінцевих точках та атаки через вебсайти. На основі цих двох напрямків побудовано основні варіанти дій зловмисника. Визначено такі сценарії атак cryptojacking:

1. Випадковий перехід на сайт, що виконує несанкціонований криптомайнінг.
2. Цільова атака, що передбачає створення зловмисником сайту.
3. Цільова атака зі зловмисним програмним забезпеченням на одному пристрої.
4. Розповсюдження зловмисного програмного забезпечення на всю мережу.
5. Виконання криптомайнінгу інсайдером на певній кількості пристроїв.

Перший сценарій атаки передбачає відкриття користувачем сайту, що під час використання виконує несанкціонований майнінг криптовалюти. Такий тип атаки має назву drive-by cryptojacking, для виконання якої зловмисник завантажує шкідливий скрипт, що виконує криптомайнінг, на власних сайтах, або на попередньо скомпрометованих. Основні кроки для цього сценарію показані на рисунку 2.2. Симуляцію на кожній ітерації необхідно виконувати окремо для кожного пристрою та зальні втрати розраховувати як суму окремих збитків, адже така атака може бути виконана незалежно на різних пристроях.

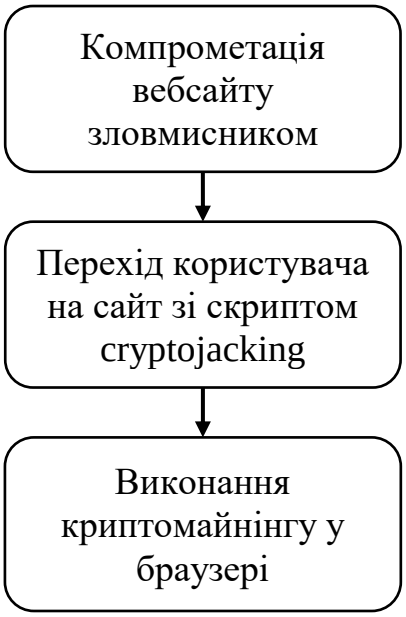

Рисунок 2.2 – Схема першого сценарію атаки

Другий сценарій передбачає цільову атаку на конкретне підприємство або користувача (рис. 2.3). Зловмисник створює копію вебсайту, яким користуються співробітники цієї організації, та надсилає посилання на цей сайт у фішинговому листі, або отримує доступ до легітимного сайту та завантажує на нього скрипт, що виконує майнінг. У результаті співробітники переходять з посиланням та запускають майнінг криптовалюти у браузері.

Наступний сценарій – це цільова атака на організацію, що передбачає виконання шкідливого програмного забезпечення на одному пристрої. Під час такої атаки зловмисник надсилає фішинговий лист співробітникам організації та один

користувач відкриває посилання у цьому листі, внаслідок чого завантажується шкідливе програмне забезпечення. Далі встановлюється підключення із сервером зловмисника для отримання інструкцій, після чого виконується майнінг криптовалюти на пристрої. Основні етапи сценарію показані на рисунку 2.4.

Рисунок 2.3 – Схема другого сценарію атаки

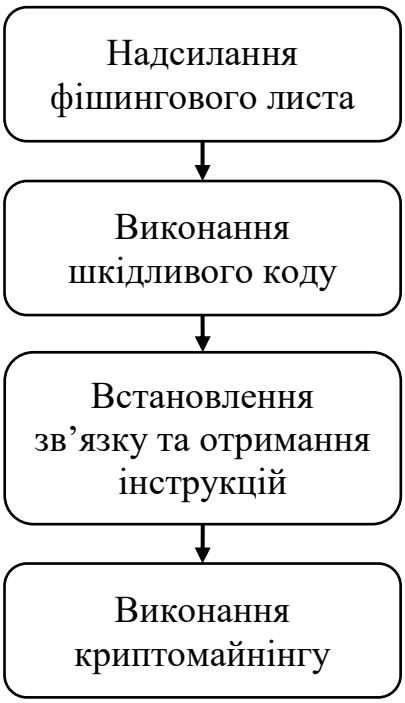

Рисунок 2.4 – Схема третього сценарію атаки

Після отримання доступу до одного пристрою, зловмисне програмне забезпечення може поширитись на інші пристрої в мережі. Для цього зловмисник виконує підвищення привілеїв, розвідку мережі організації. Далі може бути використано сервіси віддаленого доступу, сформовано фішингові листи від легітимного користувача, до пристрою якого вже отримано доступ тощо. У такому випадку схема атаки буде мати вигляд, що показаний на рисунку 2.5.

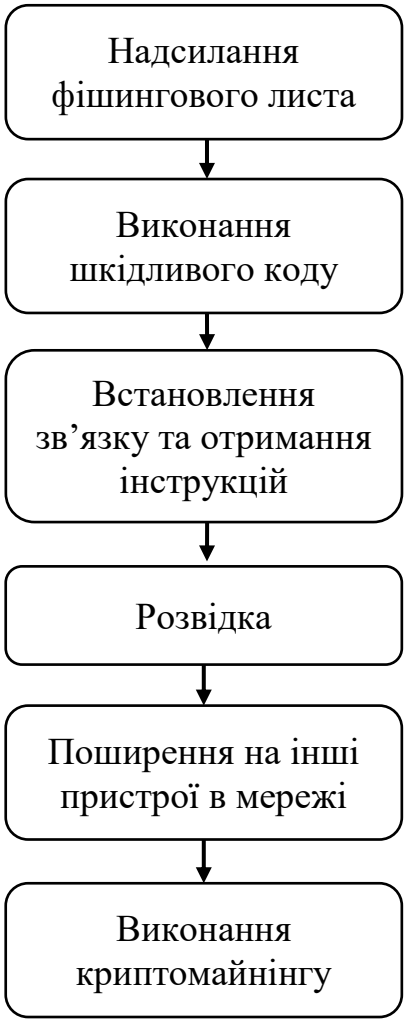

Рисунок 2.5 – Схема четвертого сценарію атаки

Окрім виконання атаки зовнішнім зловмисником, майнінг на пристроях організації також можуть виконувати самі співробітники із наявним доступом (атака інсайдера). У цьому випадку шкідливе програмне забезпечення може бути одразу встановлене на пристроях організації без необхідності отримання початкового доступу. Відмінністю від цільової атаки через вебсайти є те, що доступ у інсайдера

може бути одночасно до великого обсягу інфраструктури. Тому для моделювання такої атаки та розрахунку збитків необхідно визначити кількість уражених пристроїв на основі повноважень та рівня доступу співробітників.

2.3 Визначення ймовірності сценаріїв

Для визначення ймовірності виконання атаки запропоновано використовувати метод FTA, який передбачає побудову дерева подій, які мають відбутись для здійснення атаки cryptojacking згідно визначеного сценарію. Для кожної подій у дереві визначається окреме значення ймовірності на основі історичних даних про атаки, а також досліджень ефективності систем захисту методом експертної оцінки для поточного та для майбутнього стану. Кінцеве значення ймовірності сценарію розраховується на основі зв’язків між подіями у дереві, які визначаються відповідними операторами, зокрема [61, с. 4]:

- «AND» – вихідна подія відбувається, якщо відбулися всі вхідні події (рис. 2.6а).
- «OR» – вихідна подія відбувається, якщо відбувається будь-яка з вхідних подій (рис. 2.6б).

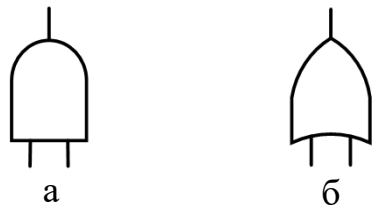

Рисунок 2.6 – Оператори «AND» (а) та «OR» (б)

Загальна ймовірність подій на вході оператора «AND» визначається як об’єднання ймовірностей окремих подій:

$$P(X) = \bigcap_{i}^{n} p(x_i) = p(x_1) \cap p(x_2) \cap \cdots \cap p(x_n), \quad (2.1)$$

де $x_n -$ вхідні події, $p(x_n) -$ ймовірність настання окремих вхідних подій, $P(X) -$ загальна ймовірність вхідних подій.

Так як вхідні події є незалежними, розрахунок загальної ймовірності виконується за формулою [69, с. 25]:

$$P(X) = \prod_{i}^{n} p(x_i) = p(x_1) \times p(x_2) \times \cdots \times p(x_n). \quad (2.2)$$

Загальна ймовірність подій на вході оператора «OR» означає появу хоча б однієї із вхідних подій та визначається як різниця між одиницею і добутком ймовірностей подій, що протилежні до вхідних. Це означає всі випадки окрім тих, коли всі вхідні події не здійснюються. Така ймовірність розраховується за формулою [69, с. 30]:

$$\begin{gathered} P(X) = 1 - p(\overline{x_1}) \times p(\overline{x_2}) \times \cdots p(\overline{x_n}) = \\ 1 - \big(1 - p(x_1)\big) \times \big(1 - p(x_2)\big) \times \cdots \times \big(1 - p(x_n)\big), \end{gathered} \quad (2.3)$$

де $P(X) -$ загальна ймовірність, $p(\overline{x_n}) -$ ймовірність настання події, що протилежна до x_n.

Сценарій, який передбачає випадковий перехід на сайт, що виконує несанкціонований майнінг криптовалют, має такі ключові події, які визначають ймовірність виконання атаки:

- перехід користувача на скомпрометований вебсайт;
- обхід системи виявлення cryptojacking на фаєрволі;
- обхід системи виявлення cryptojacking у браузері.

Сценарій буде успішним якщо усі події будуть виконані, тому для побудови fault tree використано оператор «AND» (рис. 2.7).

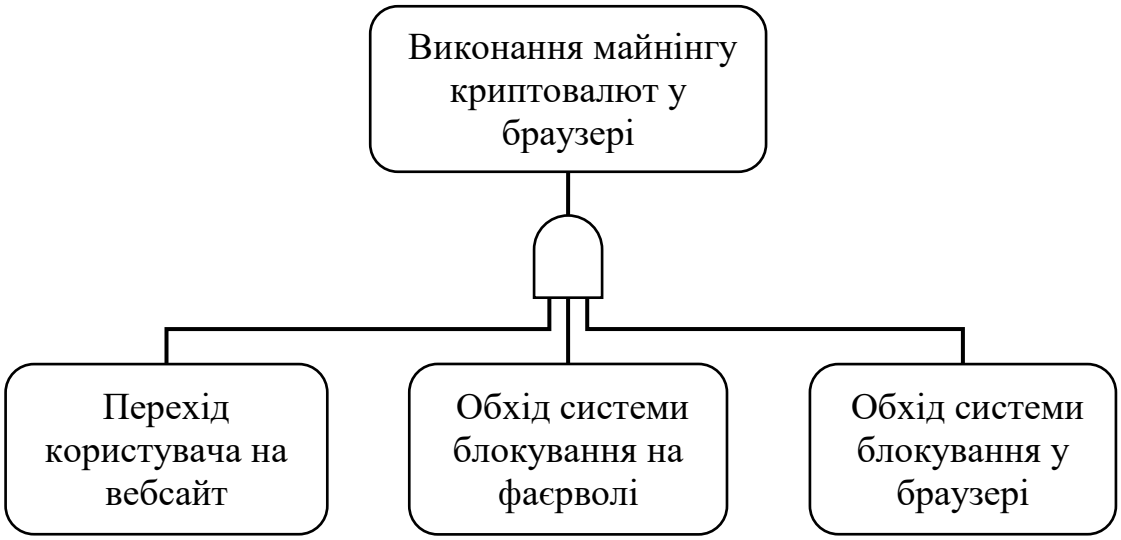

Рисунок 2.7 – Fault tree для першого сценарію атаки

Імовірність переходу користувача на вебсайт, що виконує майнінг криптовалюти визначимо на основі загального частки таких сайтів в інтернеті, що становить 0,011%. Ймовірність обходу системи блокування на фаєрволі та у браузері визначається згідно стану цих засобі захисту на підприємстві та можуть бути вилучені із дерева, якщо відповідні засоби відсутні. У випадку, коли використовується лише фільтрування URL адрес ймовірність обходу цього контролю становитиме 50% згідно проаналізованих досліджень. Якщо організація використовує засоби машинного навчання, що аналізують параметри використання системних ресурсів, тоді ймовірність успіху їх обходу буде 10%, враховуючи те, що такі засоби забезпечують в середньому виявлення 90% атак cryptojacking. Також якщо використовується комплексний аналіз параметрів системи та мережевого трафіку, тоді ймовірність успіху обходу цього засобу становитиме 2%, враховуючи, що він може виявити 98% атак cryptojacking.

Сценарій, що передбачає цільову атаку використовуючи вебсайт, яким користуються співробітники організації передбачатиме такі ключові події:

- надсилання користувачу посилання на створений сайт або компрометація сайту, яким користується жертва;
- обхід системи блокування на фаєрволі;
- обхід системи блокування у браузері.

Fault tree для цього сценарію буде мати такий самий вигляд як для попереднього, але у цьому випадку передбачається, що користувач точно перейде на зловмисний або скомпрометований сайт, тому ймовірність першої події становить 100%.

Сценарій цільової атаки, у якому відбувається виконання зловмисного програмного забезпечення, що виконує майнінг криптовалют на одному пристрої має такі ключові події:

- відкриття користувачем посилання у фішинговому листі;
- обхід засобів виявлення шкідливого програмного забезпечення;
- обхід системи блокування зловмисного трафіку на фаєрволі;
- обхід системи виявлення майнінгу на хості.

Для побудови fault tree розділимо ці події на три етапи. Перший етап успішного виконання атаки передбачає відкриття користувачем посилання або файлу у фішинговому листі та обхід засобів антивірусного захисту під час завантаження та виконання зловмисного програмного забезпечення. На наступному етапі виконується встановлення з’єднання із сервером зловмисника для управління та контролю за виконанням атаки, а також із серверами сервісів майнінгу. Цей трафік може бути заблокований на рівні фаєрволу, тому зловмисник має обійти цей засіб захисту. На останньому етапі виконується майнінг на пристрої користувача, що може бути виявлено та заблоковано системою виявлення таких дій на основі аналізу параметрів використання системних ресурсів. Для успіху атаки повинні бути виконані всі події у дереві, тому використано оператор «AND». Fault tree згідно цих етапів показано на рисунку 2.8.

Імовірність подій у дереві визначається на основі засобів захисту, що використовуються в організації, а також згідно оцінки стану захищеності методом експертного аналізу. Зокрема, ймовірність відкриття користувачем посилання у фішинговому листі може бути визначена на основі того, чи проводиться навчання користувачів щодо фішингу, а також згідно результатів розсилки тестових фішингових листів.

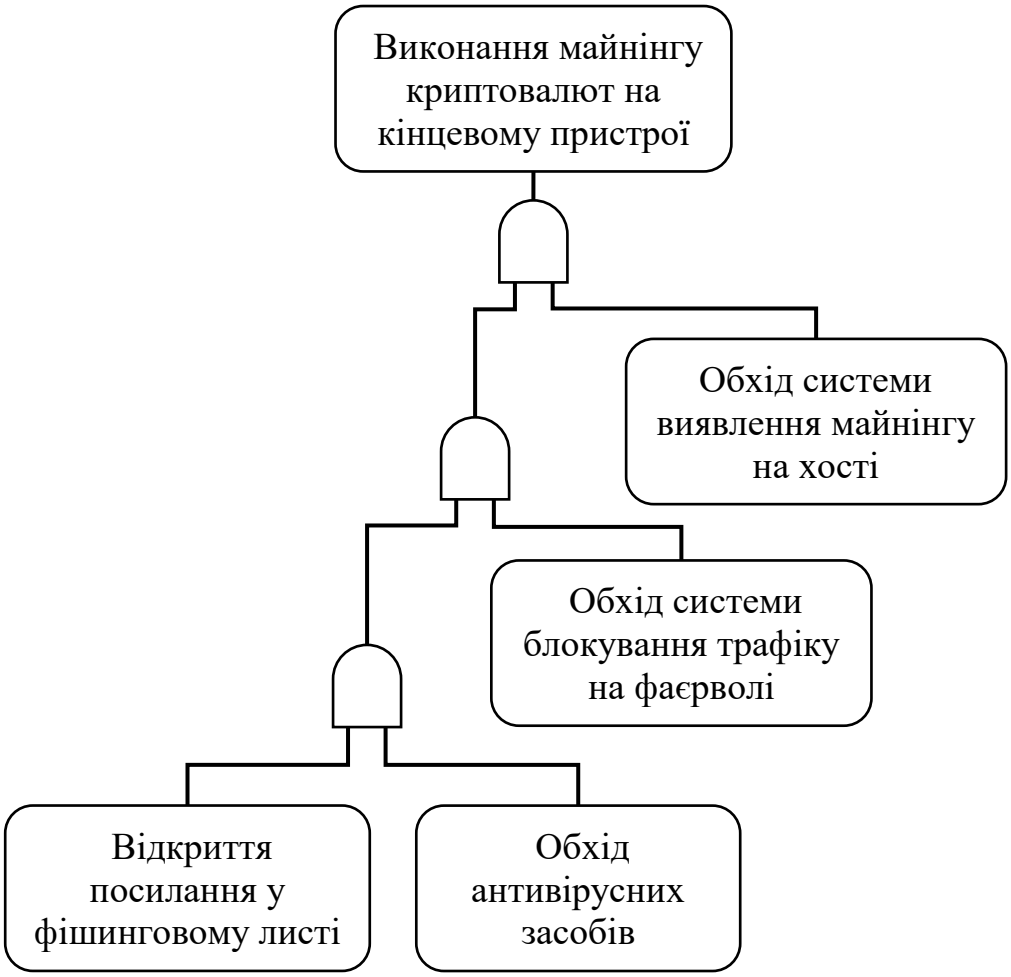

Рисунок 2.8 – Fault tree для третього сценарію атаки

Наступний сценарій є доповненням попереднього, коли атака поширюється в мережі на інші пристрої, що може передбачати такі ключові події:

- розвідка мережі організації;
- експлуатація вразливостей;
- викрадення даних для входу в облікові записи;
- підбір паролів до сервісів віддаленого доступу.

Ці події являють собою можливі шляхи, за якими зловмисник може розповсюдити зловмисне програмне забезпечення на інші пристрої. Вони можуть виконуватись окремо для успішного виконання атаки, тому для побудови fault tree використано оператор «OR» (рис. 2.9). Значення ймовірностей подій, що формують такий сценарій, визначаються на основі результатів тестувань на проникнення та експертної діагностики стану кібербезпеки. Зокрема, успішність розвідки мережі

визначається на основі того, чи виконана сегментація, а також чи обмежено використання мережевих інструментів на пристроях користувачів. Імовірність експлуатації вразливостей залежить від зрілості відповідного процесу в організації. Зокрема, чи виконується сканування вразливостей та як часто, які виконуються заходи з реагування на виявлені вразливості, а також які саме програмні засоби використовуються в організації, що може впливати, наприклад, на швидкість випуску оновлень та виправлень. Імовірність підбору паролів та викрадення даних для входу в систему визначається на основі парольної політики та впроваджених контролів, таких як двофакторна автентифікація, вимоги складності, періодична зміна паролів.

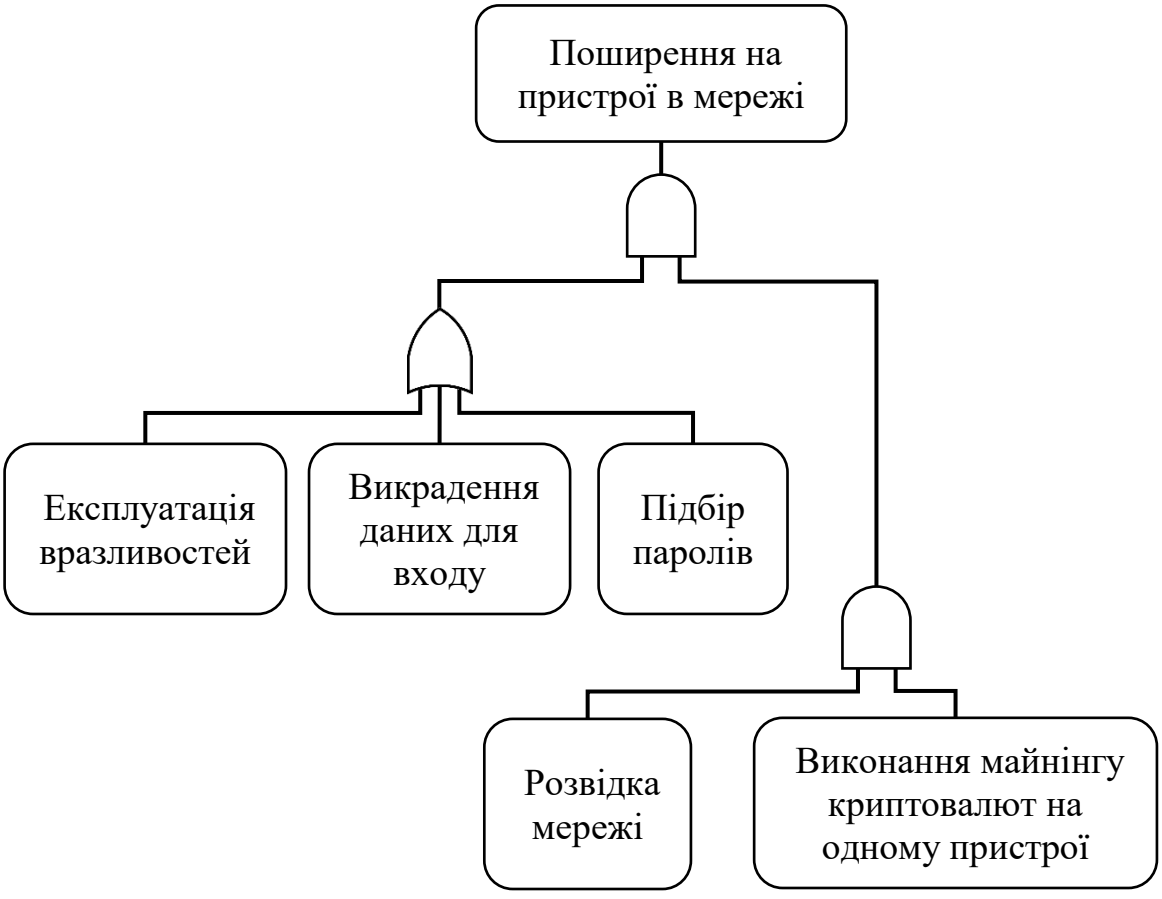

Рисунок 2.9 – Fault tree для четвертого сценарію атаки

Для сценарію, що передбачає атаку інсайдера, необхідно обійти засоби антивірусного захисту, а також системи виявлення та блокування майнінгу на фаєрволі та на кінцевих пристроях. Імовірність подій за цим сценарієм визначається

на основі експертної оцінки, а також згідно того, які саме використовуються засоби виявлення та блокування майнінгу на фаєрволі та кінцевих пристроях.

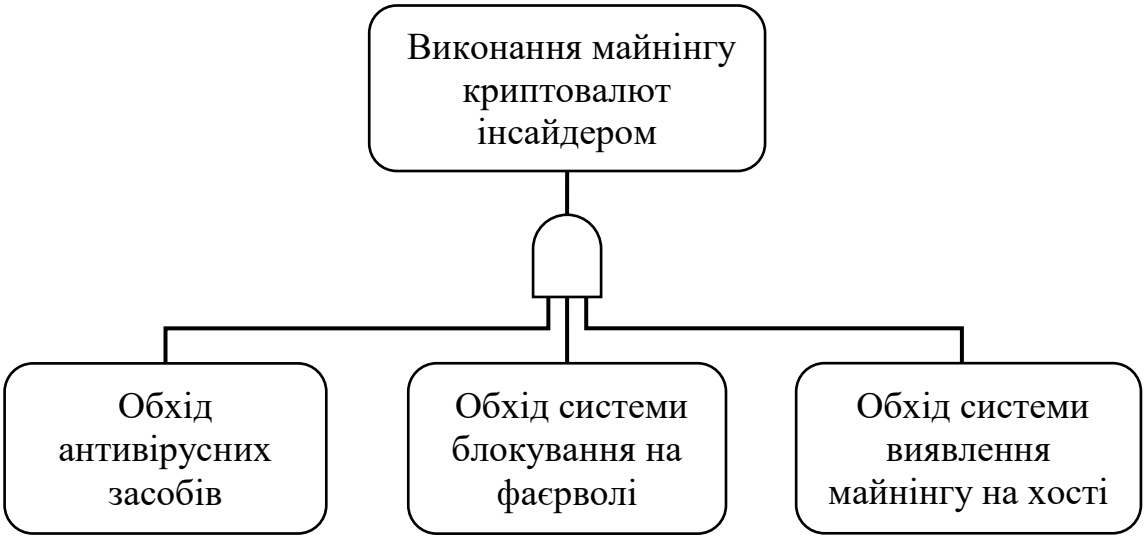

Рисунок 2.10 – Fault tree для п’ятого сценарію атаки

2.4 Симуляція Монте-Карло

Основним компонентом запропонованої моделі є симуляція Монте-Карло, що використовується для прогнозування розміру збитків. Цей метод передбачає формування переліку випадкових значень на основі розподілу ймовірностей, що показує можливий діапазон втрат протягом певного періоду часу. Також він дозволяє побудувати криву втрат (LEC), яка показує розміри втрат та їх імовірності.

Метод Монте-Карло обрано через випадкову природу атак cryptojacking, адже їх збитки можуть суттєво варіюватись – починаючи від надмірного використання електрики, що може бути протягом різного часового періоду, затримки виробничих процесів до виходу з ладу обладнання та його повної заміни.

Перший етап симуляції Монте-Карло призначений для визначення того, чи загроза здійсниться, чи ні [70, с. 5]. Для цього обирається випадкове число від 0 до 1 та порівнюється з імовірністю відповідного сценарію атаки. Якщо це число менше або дорівнює ймовірності атаки, тоді вона вважається успішною та виконуються

наступний етап симуляції. Після визначення успішності атаки формується випадкове значення збитків згідно обраного розподілу ймовірностей.

Нехай вхідними даними симуляції є:

- кількість ітерацій – «n»;
- ймовірність виконання атаки – «p».

Тоді блок-схема виконання симуляції Монте-Карло буде мати вигляд, що показаний на рисунку 2.11.

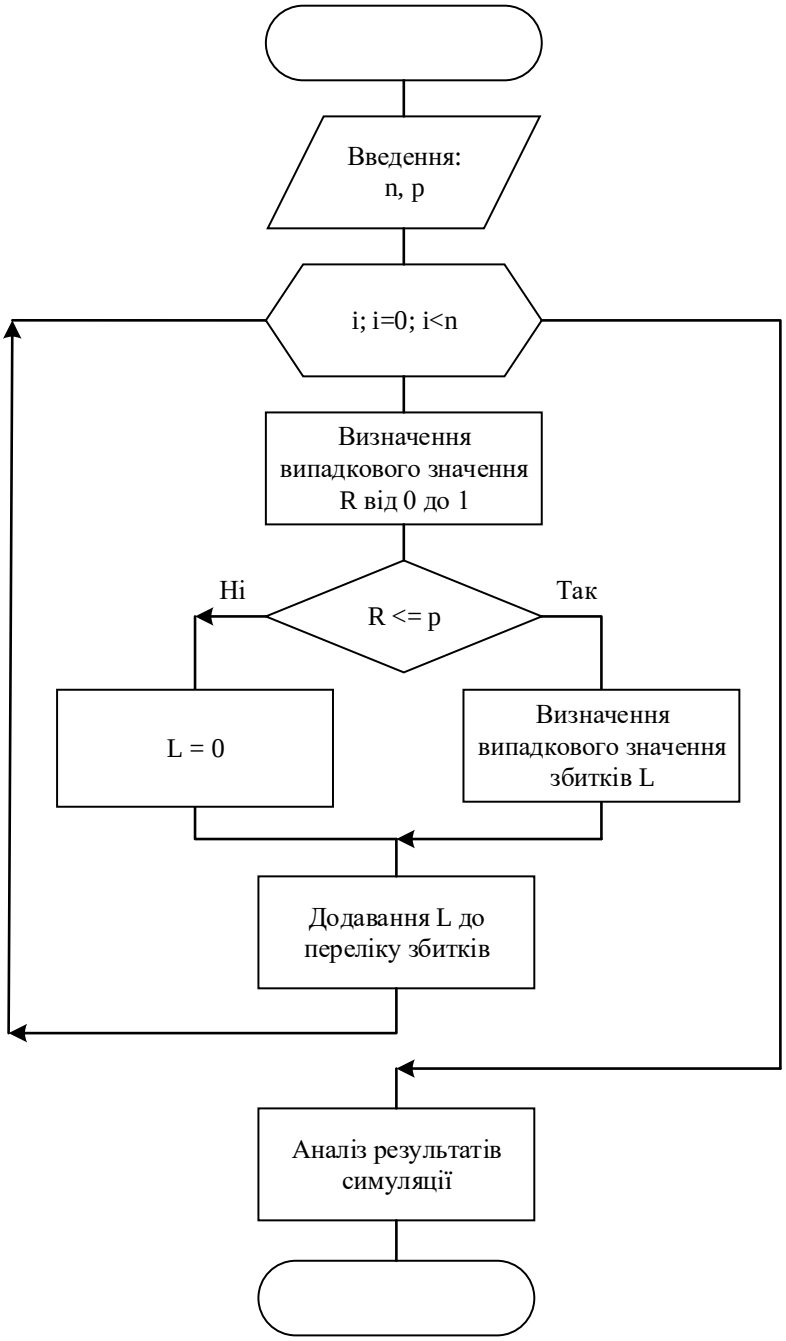

Рисунок 2.11 – Блок-схема симуляції Монте-Карло

Для оптимального моделювання втрат використовуючи цей метод необхідно визначити відповідний розподіл ймовірностей, за яким будуть обиратись випадкові значення. Зокрема, для методу Монте-Карло можуть використовуватись такі розподіли [71, с. 8]:

- нормальний, що має два параметри – середнє значення та відхилення;
- експоненціальний, що має два один параметр – середнє значення;
- логнормальний, що має два параметри – середнє значення та відхилення;
- рівномірний, що має два параметри – максимальне та мінімальне значення;
- трикутний, що має три параметри – максимальне, мінімальне та найбільш імовірне значення;
- розподіл PERT (project evaluation and review technique), що має три параметри – максимальне, мінімальне та найбільш імовірне значення;
- розподіл на основі гістограми, що має чотири і більше параметрів.

Для того, щоб обрати правильний розподіл ймовірностей побудуємо графіки щільності випадкової величини для кожного розподілу.

Щільність нормальної випадкової величини X з математичним сподіванням μ та дисперсією σ^2 має вигляд [72, с. 173]:

$$n(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}. \quad (2.4)$$

Графік щільності для середнього значення 5 та відхилення 1 показано на рисунку 2.12. Нормальний розподіл застосовується для моделювання великої кількості природних явищ. Він часто підходить для моделювання змінної, яка є адитивним добутком багатьох незалежних змінних, де жодна змінна не домінує [73, с. 774]. Для моделювання збитків, спричинених атаками cryptojacking, такий розподіл не надає достатньо можливостей контролю, адже він має лише два параметри, за якими неможливо точно встановити розміри втрат.

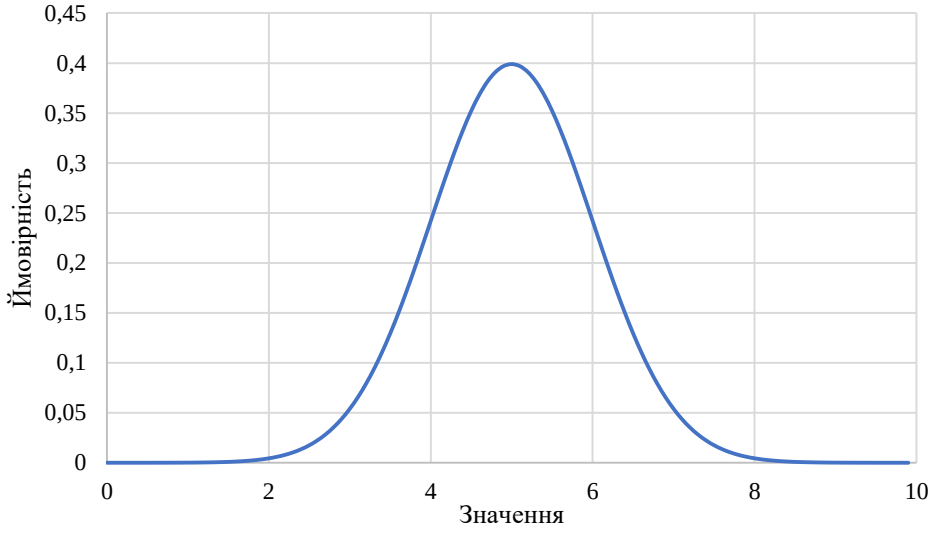

Рисунок 2.12 – Графік щільності нормального розподілу

Неперервна випадкова величина X має експоненціальний розподіл з параметром β, якщо її функція щільності має вигляд [72, с. 195]:

$$f(x;\beta) = \begin{cases} \frac{1}{\beta} e^{-x/\beta}, & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{2.5}$$

Побудуємо графік щільності експоненціального розподілу, обравши параметр β = 1 (рис. 2.13).

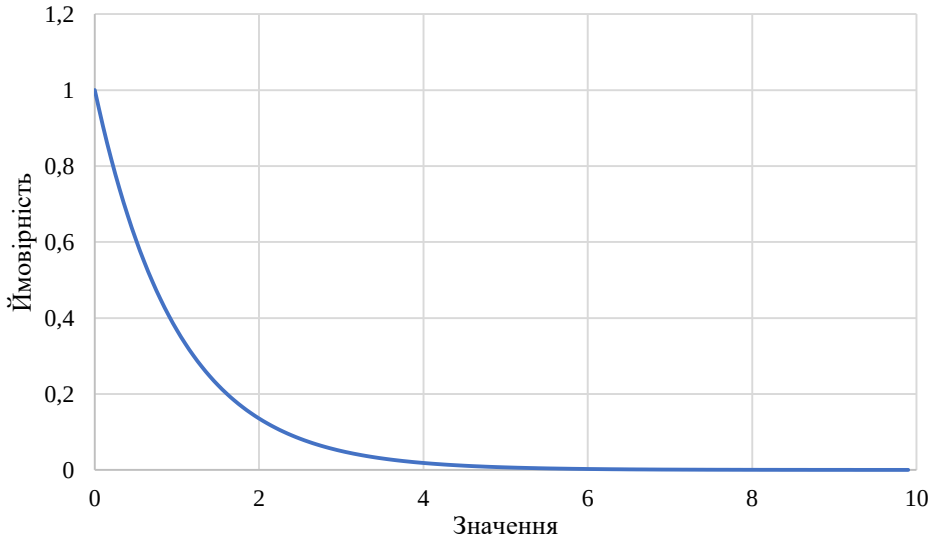

Рисунок 2.13 – Графік щільності експоненціального розподілу

Для симуляції атак cryptojacking експоненціальний розподіл не є оптимальним, адже вхідний параметр не надає достатньо контролю над формою функції щільності розподілу для формування випадкових значень збитків.

Неперервна випадкова величина X має логнормальний розподіл, якщо випадкова величина Y = ln(X) має нормальний розподіл із середнім значенням μ і стандартним відхиленням σ. Результуюча функція щільності розподілу випадкової величини X має вигляд [72, с. 201]:

$$f(x;\mu,\sigma) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{1}{2\sigma^2}[\ln(x)-\mu]^2}, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (2.6)$$

Для побудови графіку щільності логнормального розподілу оберемо середнє значення 1 та відхилення 1 (рис. 2.14).

Якщо встановити середнє значення 4 та залишити відхилення 1, тоді графік щільності на проміжку 0;10 буде мати вигляд, що показаний на рисунку 2.15. Отримані зміни на графіку показують, що такий розподіл не надає достатньо контролю над щільність ймовірностей, який необхідний для моделювання збитків від атак.

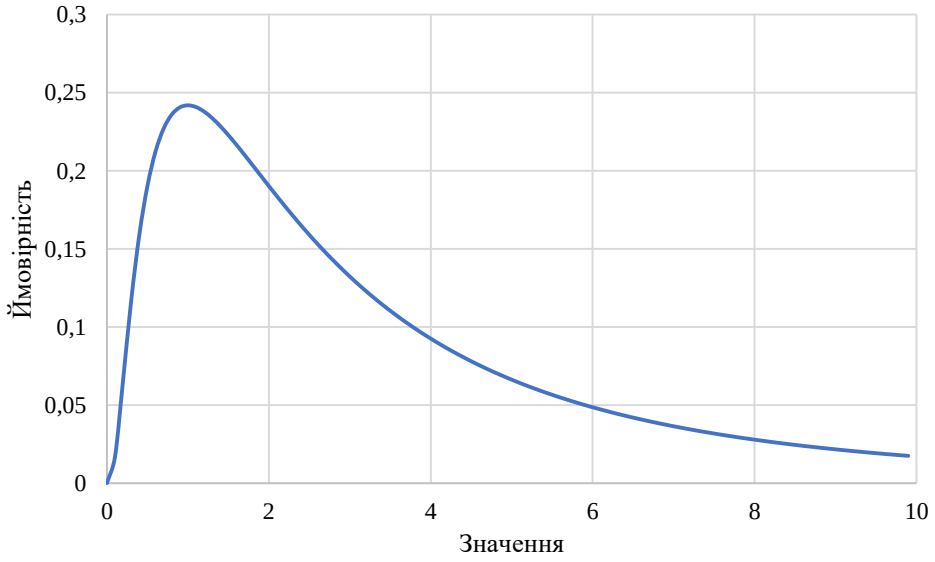

Рисунок 2.14 – Графік щільності логнормального розподілу

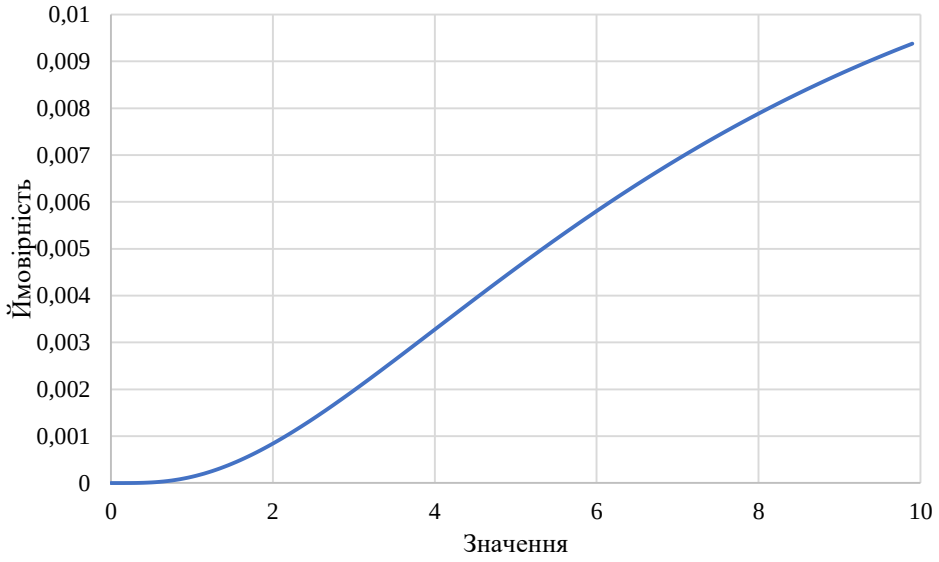

Рисунок 2.15 Графік щільності логнормального розподілу

Логнормальний розподіл може бути доречним, коли найбільше значення випадкової величини більше або дорівнює нулю. Крім того, такий розподіл часто підходить для моделювання змінної, яка є мультиплікативним добутком багатьох незалежних змінних, де жодна змінна не домінує. Користувач зазвичай вказує середнє значення розподілу (або медіану) і стандартне відхилення [73, с. 774].

Функція щільності неперервної однорідної випадкової величини X на проміжку [A, B] має вигляд [72, с. 171]:

$$f(x;A,B) = \begin{cases} \dfrac{1}{B-A}, & A \le x \le B, \\ 0, & \text{на інших проміжках.} \end{cases} \tag{2.7}$$

Графік щільності однорідного розподілу на проміжку 0;10 показано на рисунку 2.16. Він обмежений значеннями на проміжку, які рівномірно розподілені між кінцевими точками. Тобто, всі значення з однаковою ймовірністю зустрічаються на вказаному проміжку. Однорідний розподіл легко моделювати і візуалізувати, але він може не підходити для чітко визначених елементів [73, с. 775], наприклад, для переліку витрат внаслідок кібератак. Натомість, такий розподіл може бути

використаний для визначення успішності виконання атаки, адже цей процес є повністю випадковим.

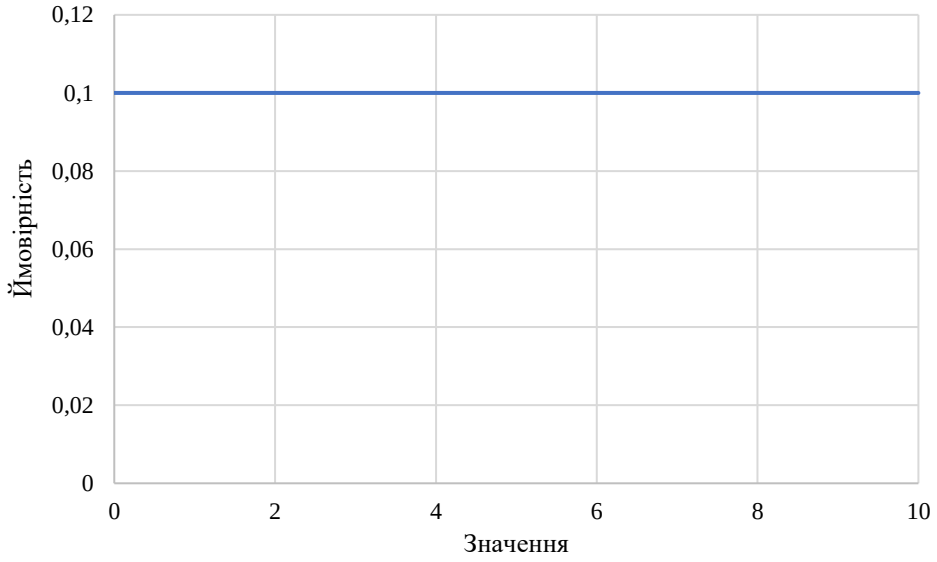

Рисунок 2.16 – Графік щільності однорідного розподілу

Трикутний розподіл має три параметри: a = мінімальне значення, c = найбільш ймовірне значення, b = максимальне значення, є неперервним розподілом ймовірностей з нижньою межею a, середнім значенням c і верхньою межею b. Функція щільності розподілу ймовірностей трикутного розподілу визначається за формулою [74, с. 4]:

$$f(x;a,b,c)=\begin{cases}\dfrac{2(x-a)}{(b-a)(c-a)}, & a\le x\le c,\\[2ex] \dfrac{2(b-x)}{(b-a)(b-c)}, & c<x\le b.\end{cases} \tag{2.8}$$

Для побудови графіку щільності трикутного розподілу оберемо максимальне значення 10, середнє значення 3 та мінімальне значення 0. Отриманий графік показаний на рисунку 2.17.

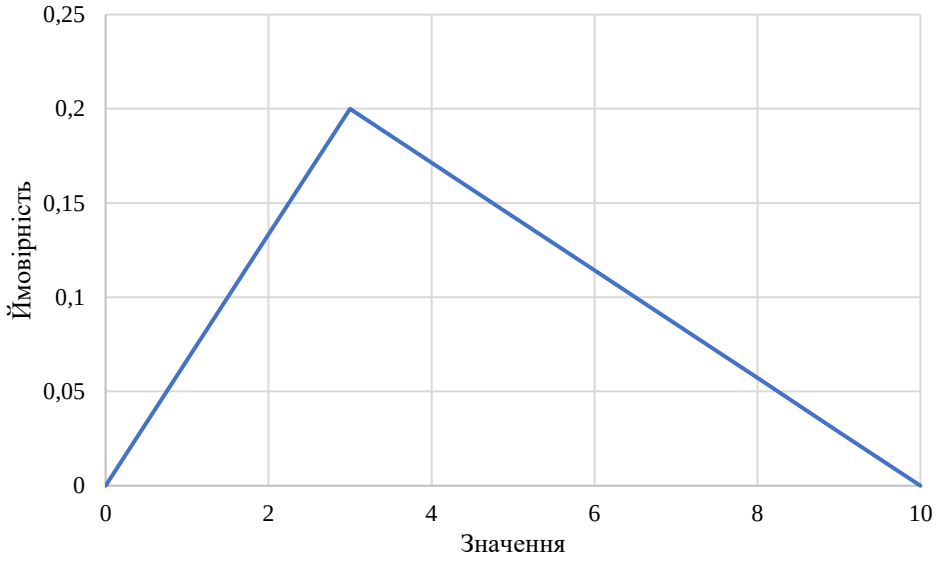

Рисунок 2.17 – Графік щільності трикутного розподілу

Трикутний розподіл може бути доречним, коли відомі найнижчі та найвищі значення (або можуть бути точно оцінені) разом із найбільш імовірним (середнім) значенням, взаємозв'язок між ними відомий, і дані безперервно розширюються від найнижчого значення до середнього і від середнього до найнижчого [73, с. 774].

Розподіл PERT є подібним до трикутного, але має більш плавний графік щільності. Він являє собою окрему форму бета-розподілу, та використовується під час виконання аналізу ризиків [73, с. 773]. Якщо маємо параметри, які використовувались для трикутного розподілу: a – мінімальне значення, b – найбільш імовірне та c – максимальне, тоді функція щільності розподілу PERT має вигляд [75, с. 3; 76, с. 4]:

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{(x-a)^{\alpha-1}(c-x)^{\beta-1}}{(c-a)^{\alpha+\beta-1}} = \frac{1}{B(\alpha,\beta)} \frac{(x-a)^{\alpha-1}(c-x)^{\beta-1}}{(c-a)^{\alpha+\beta-1}}, \quad (2.9)$$

де $\alpha = \frac{4b+c-5a}{c-a}$, $\beta = \frac{5c-a-4b}{c-a}$, Γ – гамма-функція, B – бета-функція.

Графік щільності розподілу PERT для максимального значення 10, середнього значення 3 та мінімального значення 0 показаний на рисунку 2.18.

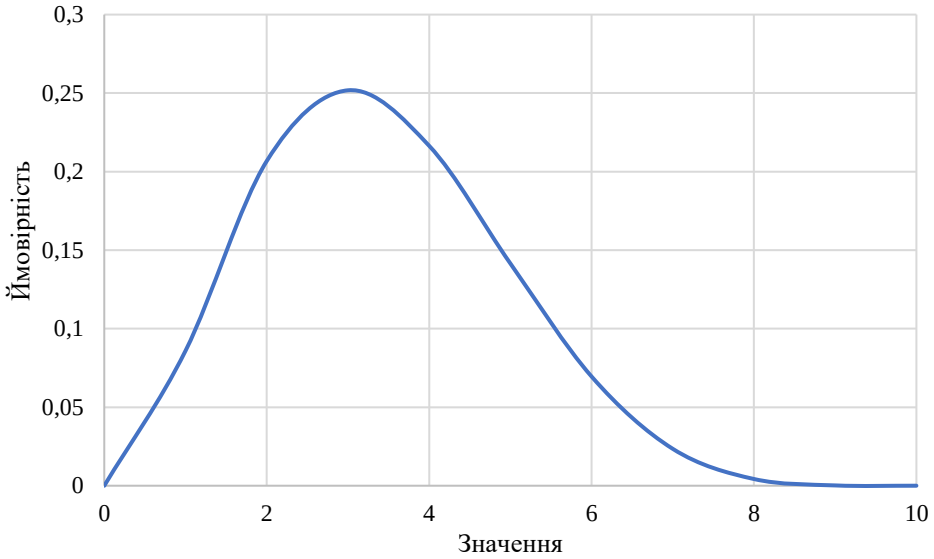

Рисунок 2.18 – Графік щільності розподілу PERT

Змінивши значення параметрів маємо можливість легко контролювати як трикутний розподіл ймовірностей, так і розподіл PERT, зокрема якщо встановити середнє значення 6, тоді графік щільності розподілу PERT буде мати вигляд, який показаний на рисунку 2.19.

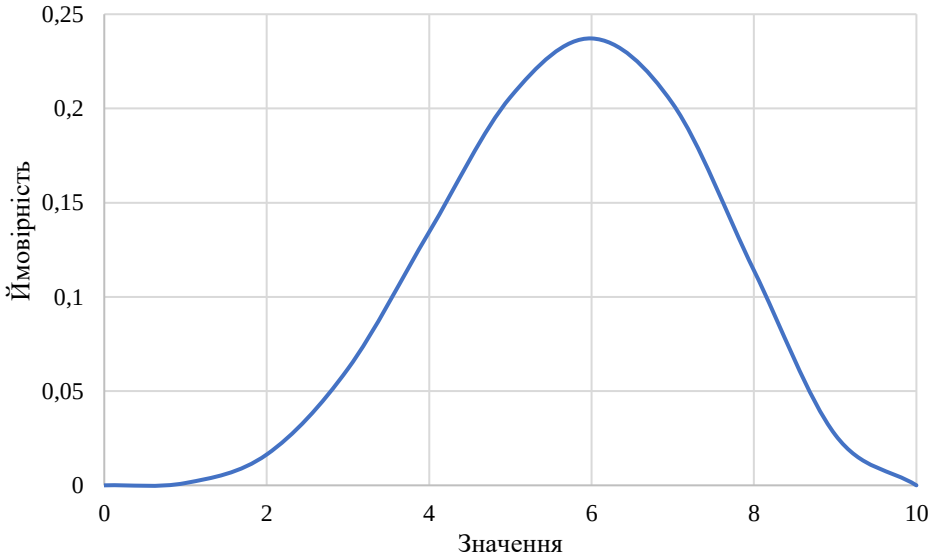

Рисунок 2.19 – Графік щільності розподілу PERT

Тому для атак cryptojacking доцільно використовувати саме трикутний розподіл або розподіл PERT, які дозволяють змінювати функцію щільності ймовірностей за

допомогою параметрів, що є максимальним, мінімальним та найбільш імовірним розміром збитків. Найбільш імовірний розмір збитків буде відповідати розміру витрат електроенергії та збитків від сповільнення основних задач. Мінімальний розмір збитків може бути розраховано як відхилення від середнього значення через те, що атаки можуть виконуватись протягом різного періоду часу. Максимальний розмір збитків додатково буде включати витрати на заміну обладнання у випадку його виходу з ладу через надмірне навантаження.

2.5 Розробка програми для оцінки ризиків

Для виконання оцінки ризиків розроблено програмний засіб на мові програмування Python. Під час розробки використано деякі напрацювання, що описані у роботі «Monte Carlo simulation for cyber threats portfolio» [77].

У програмі використано такі модулі Python:

- «random» – модуль, який надає функції, що призначені для генерації псевдовипадкових чисел.
- «matplotlib» – бібліотека, що надає можливість створювати графіки. У програмі використано такі модулі цієї бібліотеки: «pyplot» – включає в себе основні функції для створення графіків; «ticker» – модуль, що дозволяє керувати позначками на графіку, із якого використано клас «FuncFormatter», що призначений для встановлення формату підписів поділок на графіку.
- «numpy» – бібліотека Python, що призначена для роботи з масивами.
- «pertdist» – модуль, що надає генератор псевдовипадкових чисел за розподілом ймовірності PERT.

Дані для виконання симуляції зберігаються як об’єкти класу «Scenario», що має такі змінні:

- змінна «name» – назва сценарію;
- число «probability» – значення ймовірності відповідного сценарію атаки;
- число «devices» – кількість пристроїв, для яких потрібно окремо виконати симуляцію;

- список «lossv» – містить мінімальне, середнє та максимальне значення збитків;
- список «losses» – призначений для зберігання усіх згенерованих значень збитків під час симуляції.

У цьому класі змінні «name», «probability» та «lossv» є вхідними даними, список «losses» – результат симуляції.

В основній функції програми створюється список «scenarios», який містить об’єкти класу «Scenario» та присвоюються вхідні значення для кожного сценарію. Також у функції оголошується числова змінна «iterations», що містить кількість ітерацій симуляції. Далі викликаються функції «monte_carlo_simulation» та «analyse», яким передаються змінні «scenarios» та «iterations».

Виконання симуляції Монте-Карло поділено на такі функції:

- «monte_carlo_simulation» – основні функція симуляції, у якій для кожного сценарію у списку «scenarios» виконується цикл, який має кількість ітерацій, що визначена у змінній «iterations». У циклі викликається функція «simulate_attack» після чого повернене значення із цієї функції додається до списку «losses» відповідного сценарію.
- «simulate_attack» – функція, що описує виконання симуляції атаки – спочатку присвоюється значення «0» для локальної змінної «loss». Далі у циклі із кількість ітерацій, що рівна значенню змінної «devices» об’єкта класу «Scenario» викликається функція «is_attack_successful», яка повертає булеве значення. Якщо це значення «True», тоді викликається функція «calculate_loss», яка повертає згенероване значення збитків, що додається до змінної «loss», після чого повертається змінна «loss».
- «is_attack_successful» – функція, у якій генерується випадкове значення від 0 до 1 шляхом виклику функції «random» із модуля «random», після чого отримане значення порівнюється з імовірністю сценарію атаки, що передається як вхідна змінна. Якщо згенероване значення менше або дорівнює ймовірності, тоді повертається значення «True», інакше – «False».

- «calculate_loss» – функція, яка приймає значення змінної «lossv» відповідного сценарію та на основі цих даних генерує розмір збитків з розподілом ймовірності PERT шляхом виклику функції «PERT» із модуля «pertdist».

На рисунку 2.20 показано зв’язки між класом, функціями та змінними. Стрілки показують виклики відповідних функцій та класів, пунктирні стрілки показують використання змінних у функціях.

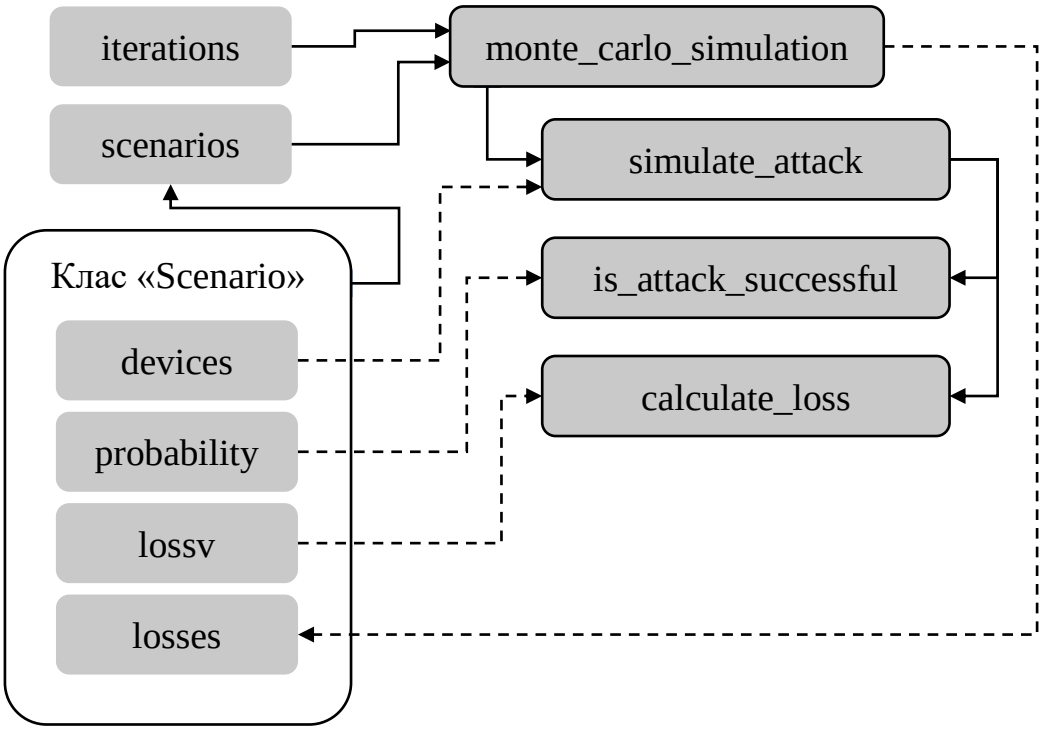

Рисунок 2.20 Схема роботи функцій симуляції Монте-Карло

Виведення та виконання аналізу отриманих даних описано у таких функціях:

- «analyse» – основна функція аналізу даних, у якій для кожного сценарію викликається функції «plot_losses» та «lec», що призначені для відображення графіків, яким передаються змінні «losses» та «name» об’єкта класу «Scenario». Далі виконується сортування списку «losses» для об’єкта класу «Scenario» використовуючи функцію «sort». Після цього видаляються нульові значення та отриманий результат зберігається як список «losses_v», розраховується довжина отриманого списку використовуючи функцію «len» та зберігається у змінній «events». Якщо кількість згенерованих значень збитків більша за 2, тоді викликається функція

«min_max_avg», якій передається список «losses_v», а також функція «ale», якій передаються змінні «losses_v», «events», «iterations».

- «plot_losses» – функція, що відображає графік, який показує розміри збитків за кожною ітерацією симуляції. Для цього використовується бібліотека «matplotlib» та її модулі «pyplot» і «ticker».
- «min_max_avg» – функція, яка виводить мінімальне, максимальне та середнє значення втрат без урахування нульових значень. Для розрахунку використовується модуль «numpy» та його функції «min», «max», «average».
- «ale» – функція, яка виводить значення очікуваних річних втрат (ALE) згідно формули (1.3). Для розрахунку ARO змінна «events» ділиться на змінну «iterations». Спочатку ALE розраховується для середнього значення збитків – під час розрахунку SLE сума усіх збитків ділиться на значення змінної «events». Також якщо всього згенерованих значень збитків більше за 3, тоді виконується розрахунок SLE для середнього значення із 10% найменших збитків та 10% найбільших збитків. У результаті отримується три значення ALE – середнє, мінімальне та максимальне.
- «lec» – функція, яка відображає графік loss exceedance curve на основі списку втрат «losses». Значення втрат ранжуються у порядку спадання, і для кожного значення розраховується його відсоток від загальної кількості втрат.

Висновки за розділом 2

У другому розділі було сформовано алгоритм кількісної оцінки ризиків cryptojacking та згідно нього розроблено модель, що включає в себе п’ять сценаріїв виконання атак для підприємства. Алгоритм включає в себе визначення ймовірності сценаріїв методом fault tree analysis та симуляцію Монте-Карло, що призначена для прогнозування розміру збитків.

Для кожного сценарію розроблено дерева відмов (fault trees), що складаються із подій, які мають відбутись для виконання атаки, а також описано визначення ймовірності подій.

Після визначення ймовірностей сценаріїв було проаналізовано різні розподіли ймовірностей, які застосовуються під час симуляції Монте-Карло для формування випадкових значень. Було запропоновано використовувати трикутний розподіл, або розподіл PERT, які надають можливість встановити мінімальне, середнє та максимальне значення для випадкових величин, що відповідають діапазону збитків від атак cryptojacking.

Також було розроблено програмний засіб на мові Python, який приймає на вході дані про сценарії атак та виконує для них симуляцію Монте-Карло. Отримані результати симуляції виводяться у вигляді графіків із прогнозованими розмірами збитків та loss exceedance curve для кожного сценарію. Крім того, виконується аналіз даних, що включає визначення максимального, середнього та мінімального значення збитків, а також розрахунок ALE. Було описано всі функції, змінні та класи, з яких складаються зазначені компоненти програмного засобу.

РОЗДІЛ 3
ОЦІНКА РИЗИКІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Розрахунок впливу атак cryptojacking

Визначення впливу атак cryptojacking виконано для організації, яка має наземну інфраструктуру, що включає в себе 100 робочих станцій та сервер. Ця організація займається 3D-моделюванням, тому потребує потужних обчислювальних ресурсів та використання їх не за функціональним призначенням призведе до сповільнення роботи, перевищення термінів виконання проектів та суттєвих фінансових збитків.

Організація має робочі станції, які призначені для роботи із програмним забезпеченням 3D-моделювання, що складаються із таких компонентів:

- CPU: Intel Core i7-10700K (8 ядер, 16 потоків), вартість приблизно $400 на одиницю.
- GPU: NVIDIA GeForce RTX 3080 (10 ГБ відеопам'яті), вартість приблизно $700 на одиницю.

Для розрахунку витрат електроенергії вважатимемо, що кожна робоча станція під час повного навантаження споживає приблизно 700 Вт електроенергії. Також припустимо, що зловмисник обмежує використання ресурсів під час майнінгу на рівні 60% для уникнення виявлення таких дій. Тоді споживання електроенергії несанкціонованим майнінгом на робочій станції буде становити 420 Вт.

Вартість обладнання однієї робочої станції становить:

($400 + $700) * 39,6702 грн = 43 637,22 грн.

Загальна вартість обладнання робочих станцій становить:

100 * ($400 + $700) = $110 000 = 110 000 * 39,6702 грн = 4 363 722 грн.

Сервер організації призначений для хостингу внутрішнього сервісу спільної роботи, а також для виконання рендерингу 3D-зображень. Він включає в себе такі компоненти:

- CPU: Intel Xeon Gold 6254 (18 ядер, 36 потоків), вартість приблизно $4 000.

- GPU: 2x NVIDIA Tesla V100 (32 ГБ відеопам'яті), вартість приблизно $8 000.

Сервер на повному навантаженні споживає приблизно 1600 Вт електроенергії. Враховуючи, що зловмисник використовує 50% потужностей, виконання майнінгу на сервері буде споживати 960Вт.

Загальна вартість обладнання серверу становить:

$3 000 + 2 * $8 000 = $19 000 = 19 000 * 39,6702 грн = 753 733,8 грн.

Збитки від майнінгу розраховуються за формулою:

Споживання * Час роботи * Ціна електроенергії за кВт * Кількість днів.

Припустимо, що пристрої працюють 8 годин на день та зловмисник виконує майнінг лише в робочі години для уникнення виявлення несанкціонованих дій. Ціна електроенергії для підприємств у Києві становить 5,3422 грн за кВт.год [78]. Враховуючи, що у році 256 робочих днів, тоді загальні збитки від використання електроенергії під час майнінгу на усіх пристроях організації протягом року становлять:

(420 * 100 + 960) / 1000 * 8 * 5,3422 * 256 = 470 017,867776 грн.

Окремо витрати на електроенергію від майнінгу на одній робочій станції становлять:

420 / 1000 * 8 * 5,3422 * 256 = 4 595,146752 грн.

Витрати на електроенергію від майнінгу на сервері становлять:

960 / 1000 * 8 * 5,3422 * 256 = 10 503,192576 грн.

Окрім електроенергії майнінг буде спричиняти витрати робочого часу, адже продуктивність роботи працівників знизиться. Для обчислення розміру збитків припустимо, що середня вартість робочої години 3D-моделювальника становить $30 [79]. Враховуючи, що майнінг буде займати 60% обчислювальних ресурсів, продуктивність співробітників також знизиться на 60%. Тоді збитки від використання обладнання однієї робочої станції не за функціональним призначенням протягом року становлять:

$30 * 39,6702 грн * 8 * 256 = 2 437 337,088 грн.

На основі розрахованих збитків визначимо вхідні дані розміру втрат за кожним сценарієм. Під час розрахунку врахуємо, що майнінг може виконуватись протягом

різного періоду часу. Тому для мінімальних збитків встановимо 20% робочого часу, для середніх – 50%, для максимальних – увесь робочий час. Перший сценарій на вході отримує втрати від однієї робочої станції:

- максимальні: 43 637,22 + 4 595,146752 + 2 437 337,088 = 2 485 569,454752 грн.
- середні: (4 595,146752 + 2 437 337,088) * 50% = 1 220 966,117376 грн.
- мінімальні: (4 595,146752 + 2 437 337,088) * 20% = 488 386,4469504 грн.

У другому сценарії відбувається цільова атака, внаслідок якої сайт, яким користуються співробітники для робочих задач, заражений скриптом, який виконує майнінг. Тому втрати розраховуються для всіх робочих станцій:

- максимальні: (4 595,146752 + 2 437 337,088) * 100 + 4 363 722 = 248 556 945,4752 грн.
- середні: (4 595,146752 + 2 437 337,088) * 100 * 50% = 12 209 661,17376 грн.
- мінімальні: (4 595,146752 + 2 437 337,088) * 100 * 20% = 4 883 864,469504 грн.

Третій сценарій передбачає виконання зловмисного програмного забезпечення на одній робочій станції, тому збитки будуть ті самі, що і для першого сценарію. Четвертий сценарій передбачає розповсюдження цього програмного забезпечення на всі пристрої в мережі, тому додатково враховуються втрати від використання сервера:

- максимальні: 470 017,867776 + 2 437 337,088 * 100 + 4 363 722 + 753 733,8 = 249 321 182,467776 грн.
- середні: (470 017,867776 + 2 437 337,088 * 100) * 50% = 122 101 863,333888 грн.
- мінімальні: (470 017,867776 + 2 437 337,088 * 100) * 20% = 48 840 745,3335552 грн.

Збитки за п’ятим сценарієм розраховуються залежно від того, до яких пристроїв має доступ інсайдер. Для оцінки ризиків оберемо ситуацію, коли співробітник має доступ до сервера та 40 робочих станцій. Тоді розмір збитків буде такий:

- максимальні: (4 595,146752 + 2 437 337,088 + 43 637,22) * 40 + 10 503,192576 + 753 733,8 = 100 187 015,182656 грн.

- середні: ((4 595,146752 + 2 437 337,088) * 40 + 10 503,192576) * 50% = 48 843 896,291328 грн.
- мінімальні: ((4 595,146752 + 2 437 337,088) * 40 + 10 503,192576) * 20% = 19 537 558,5165312 грн.

3.2 Розрахунок імовірності сценаріїв

Організація, для якої виконується оцінка ризиків має базову систему захисту, що складається із фаєрволу та антивірусного захисту на кінцевих пристроях. Мережа організації не сегментована та усі робочі станції можуть комунікувати одне з одним. Високорівнева схема мережі організації показана на рисунку 3.1.

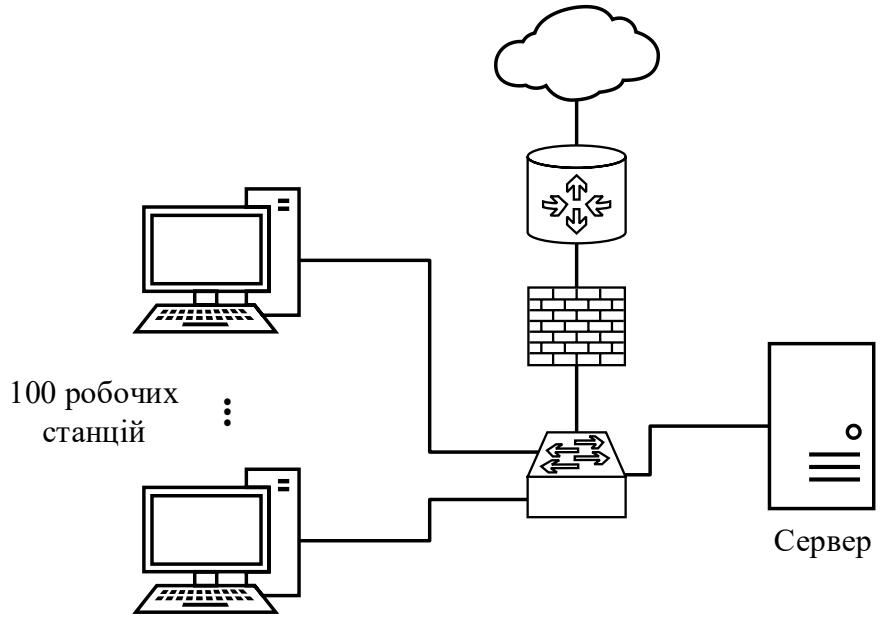

Рисунок 3.1 – Схема мережі організації

Окремі рішення з виявлення майнінгу не використовуються, але планується їх впровадження. На фаєрволі увімкнене фільтрування URL-адрес та у переліку категорій обрано «cryptomining». Майбутній стан засобів захисту передбачає використання системи виявлення та блокування, яка аналізує мережевий трафік використовуючи дані NetFlow та показники споживання системних ресурсів, що

забезпечить захист від 90% атак cryptojacking. Таким чином може бути виявлено майнінг у браузері, а також заблоковано зловмисні програми. Враховуючи, що за поточного стану додаткових засобів виявлення майнінгу у браузері не використовуються, за першим сценарієм імовірність виконання атаки становить:

$$0{,}011\% * 50\% = 0{,}0055\%.$$

Для майбутнього стану засобів захисту імовірність виконання атаки становить:

$$0{,}011\% * 50\% * 10\% = 0{,}00055\%.$$

Імовірність другого сценарію за поточного стану безпеки становить:

$$100\% * 50\% = 50\%.$$

Імовірність другого сценарію для майбутнього стану становить:

$$100\% * 50\% * 10\% = 5\%.$$

Співробітники організації ознайомлюються із базовою інструкцією щодо кібергігієни, але періодичних тренінгів з кібербезпеки не проводиться. Згідно тестової розсилки фішингових листів, успішність такої атаки становить 30%. На робочих станціях встановлений антивірусний захист Windows Defender, який працює за сигнатурами та базовим поведінковим аналізом. Співробітники не мають прав адміністратора, засоби блокування недозволених програм (Applocker) не використовуються. За даними звіту інституту SANS, антивірусний захист виявляє 47% атак [80, с. 9]. Враховуючи це, а також згідно діагностики стану кібербезпеки, результати експертного аналізу показали, антивірусний захист в організації може захистити від 40% загроз. Тому за поточного стану засобів захисту ймовірність третього сценарію становить:

$$30\% * 60\% * 50\% = 9\%.$$

За майбутнього стану безпеки ймовірність третього сценарію становитиме:

$$30\% * 60\% * 50\% * 10\% = 0{,}9\%.$$

Враховуючи, що мережа організації не сегментована та фаєрвол виконує фільтрацію лише зовнішнього трафіку, імовірність успішного виконання розвідки мережі становить 90%. Також результати експертної оцінки стану захищеності показали таке:

- ймовірність експлуатації вразливостей – 20%;

- ймовірність викрадення даних для входу – 10%;
- ймовірність успішного підбору паролів – 5%.

Тоді за поточного стану ймовірність четвертого сценарію становить:

$$9\% * 90\% * (1 – (1 – 20\%) * (1 – 10\%) * (1 – 5\%) = 2{,}5596\%.$$

За майбутнього стану безпеки ймовірність четвертого сценарію становитиме:

$$0{,}9\% * 90\% * (1 – (1 – 20\%) * (1 – 10\%) * (1 – 5\%) = 0{,}25596\%.$$

Імовірність успішного виконання атаки за п’ятим сценарієм для поточного стану засобів захисту становитиме:

$$60\% * 50\% = 30\%.$$

За майбутнього стану безпеки ймовірність п’ятого сценарію становитиме:

$$60\% * 50\% * 10\% = 3\%.$$

3.3 Виконання симуляції

Запишемо усі розраховані дані в об’єкти класу «Scenario» та запустимо симуляцію. Необхідно врахувати, що через недостатню кількість ітерацій деякі сценарії можуть не згенерувати ніяких збитків, що призведе до неможливості аналізу результатів. Тому враховуючи отримані значення ймовірності сценаріїв, встановимо кількість ітерацій 10000. Таким чином буде згенеровано розміри збитків для усіх сценаріїв. Під час симуляції використано розподіл ймовірності PERT. Для першого сценарію графік щільності розподілу ймовірності значень збитків показаний на рисунку 3.2.

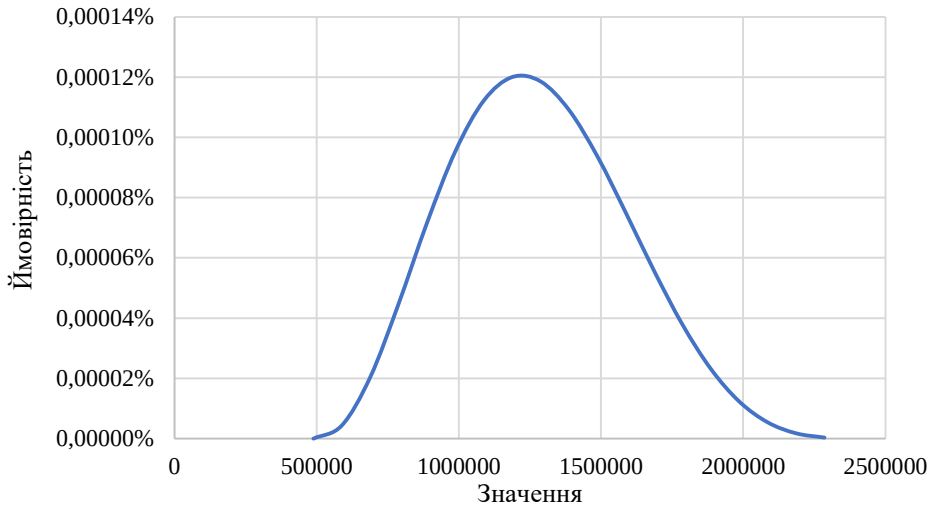

Рисунок 3.2 – Графік щільності розподілу ймовірності для першого сценарію

За результатами симуляції першого сценарію за поточного стану засобів захисту отримано перелік збитків, що показаний на рисунку 3.3. Бачимо, що кількість успішних атак невелика, а також розміри збитків суттєво відрізняються. Зокрема, отримано такі значення:

- мінімальне – 592 893 грн;
- середнє – 1 367 812 грн;
- максимальне – 2 266 616 грн.

Крива перевищення втрат показана на рисунку 3.4.

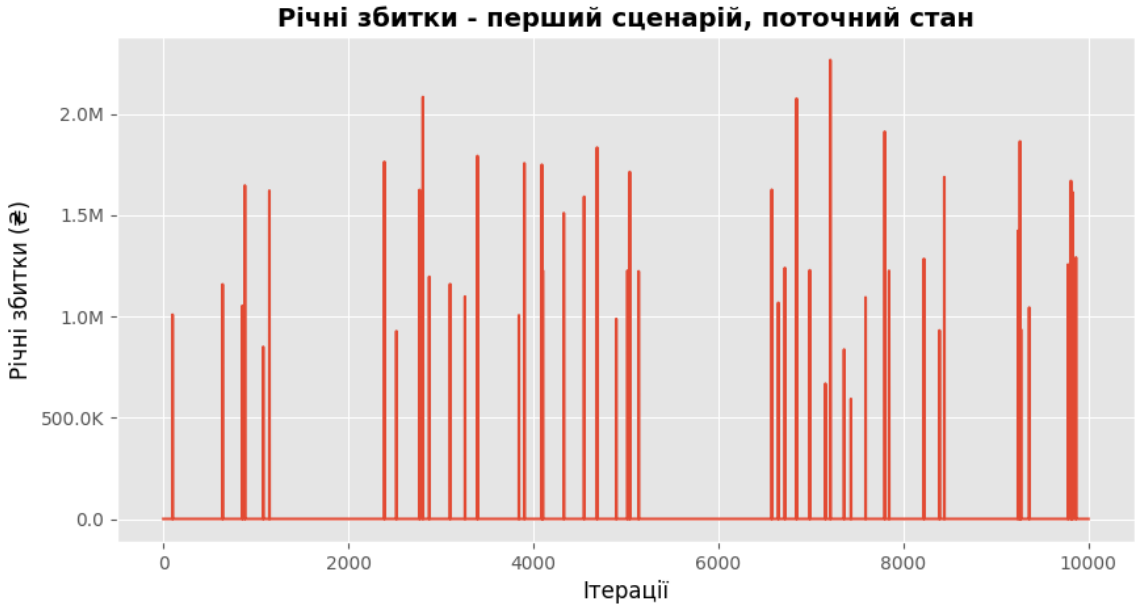

Рисунок 3.3 – Графік згенерованих збитків першого сценарію за поточного стану

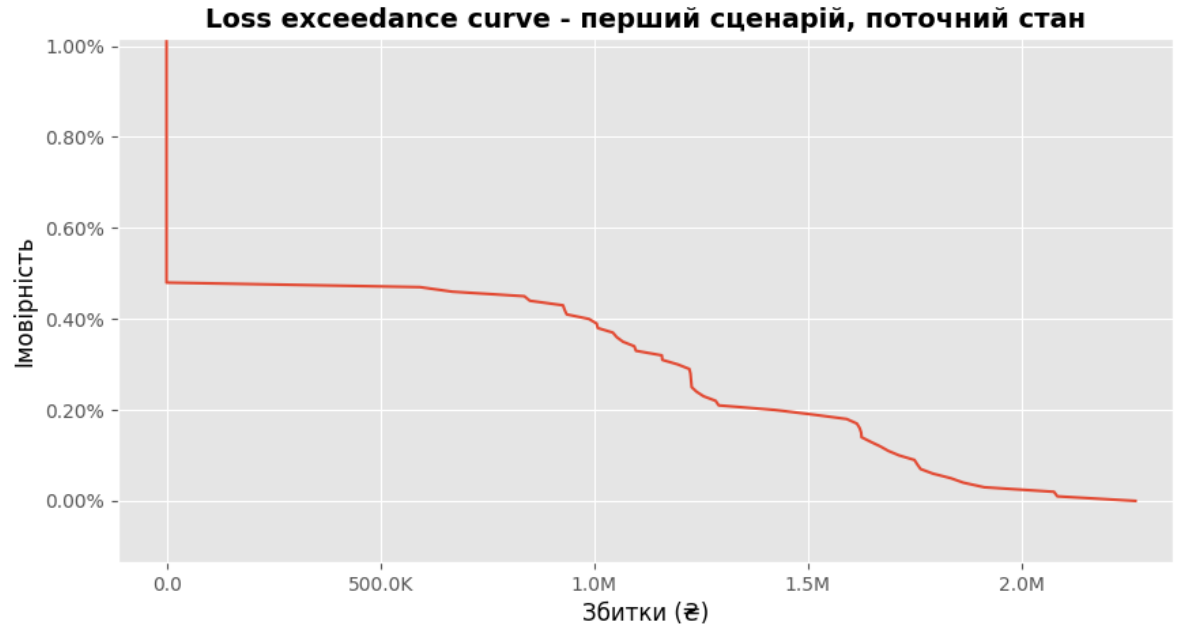

Рисунок 3.4 – Крива перевищення втрат першого сценарію за поточного стану

За результатами симуляції майбутнього стану засобів захисту для першого сценарію отримано набагато менше успішних атак (рис. 3.5).

Рисунок 3.5 – Графік згенерованих збитків першого сценарію за майбутнього стану

Всього згенеровано три значення збитків, серед яких:

- мінімальне – 1 423 129 грн;

- середнє – 1 694 960 грн;
- максимальне – 1 873 245 грн.

Крива перевищення втрат для цього сценарію показана на рисунку 3.6.

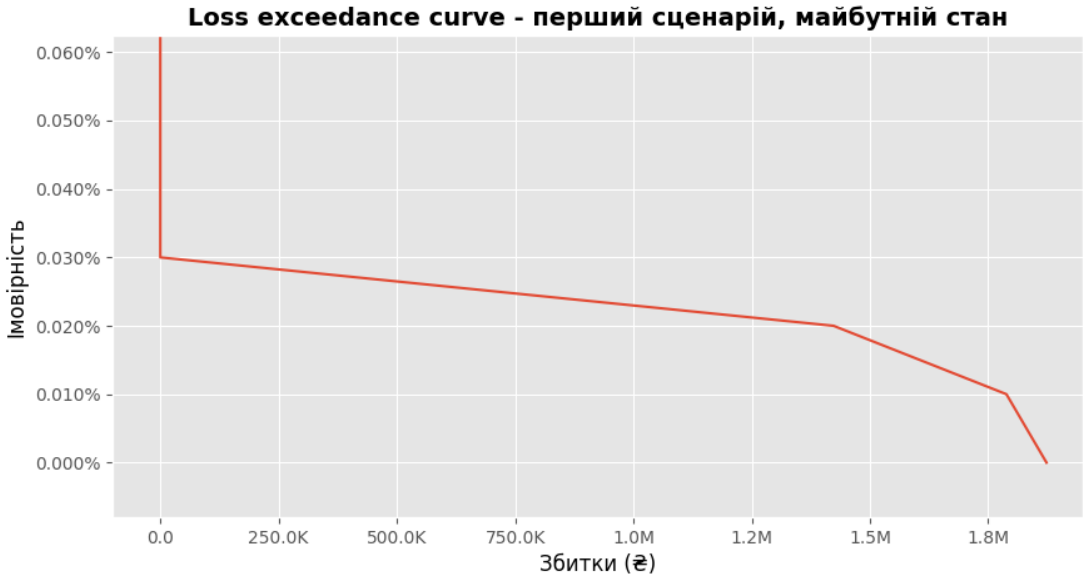

Рисунок 3.6 – Крива перевищення втрат першого сценарію за майбутнього стану

Для другого сценарію графік щільності розподілу ймовірності значень збитків показаний на рисунку 3.7.

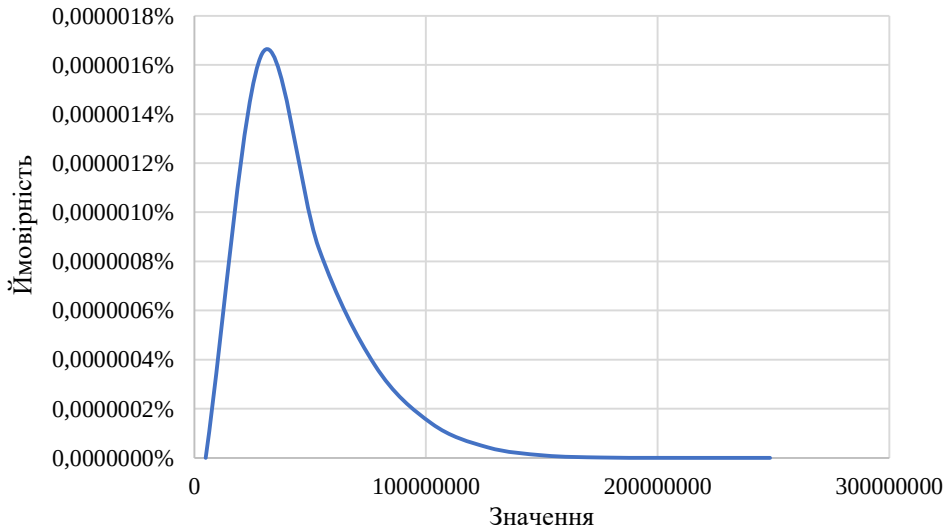

Рисунок 3.7 – Графік щільності розподілу ймовірності для другого сценарію

Отримані значення збитків за результатами симуляції показані на рисунку 3.8.

Зокрема, маємо такі розміри втрат:

- мінімальні – 4 896 032 грн;
- середні – 51 251 493 грн;
- максимальні – 207 362 069 грн.

Крива перевищення втрат для цього сценарію показана на рисунку 3.9.

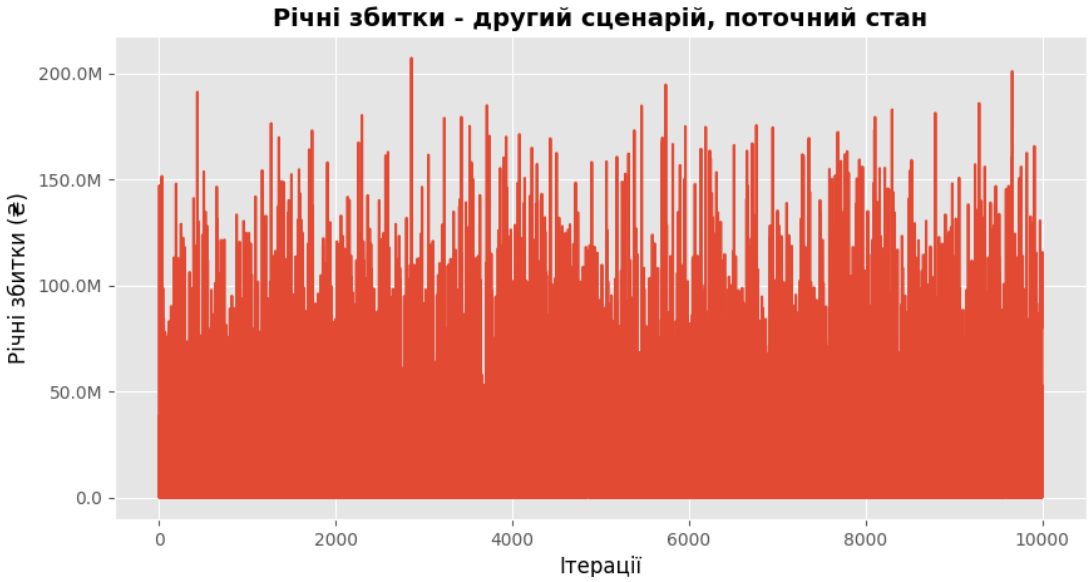

Рисунок 3.8 – Графік згенерованих збитків другого сценарію за поточного стану

Рисунок 3.9 – Крива перевищення втрат другого сценарію за поточного стану

Після впровадження рішення з виявлення та блокування майнінгу прогнозовані втрати показано на рисунку 3.10.

Зокрема, отримано такі значення:

- мінімальні збитки – 4 894 017 грн;
- середні збитки – 48 537 353 грн;
- максимальні збитки – 177 870 868 грн.

Бачимо, що ці дані майже не відрізняються від попередньої симуляції, але основним результатом впровадження системи виявлення є суттєве зменшення частоти атак cryptojacking, що показано на кривій перевищення втрат (рис. 3.11).

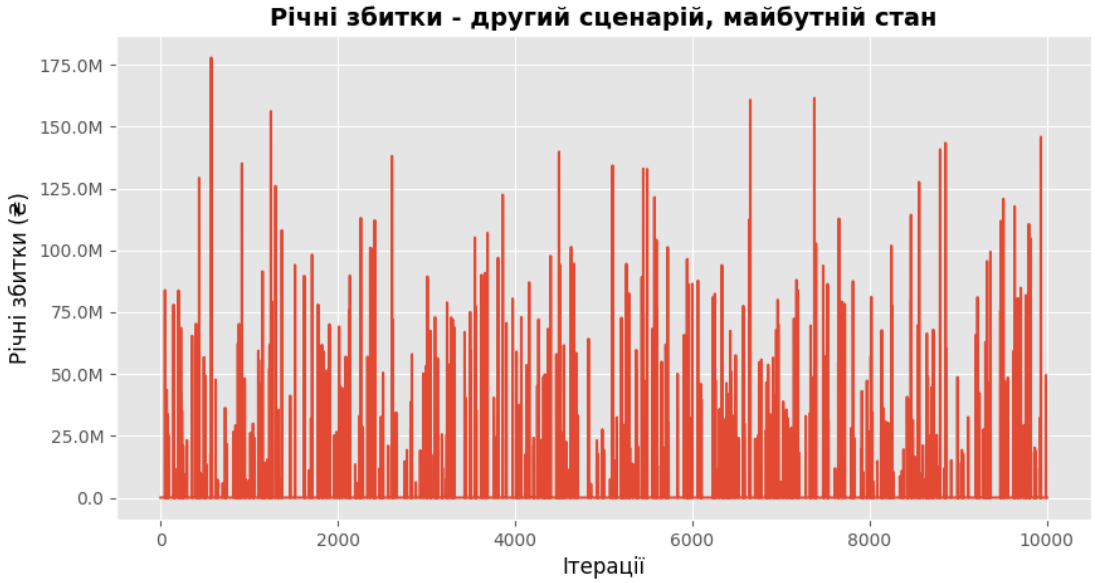

Рисунок 3.10 – Графік згенерованих збитків другого сценарію за майбутнього стану

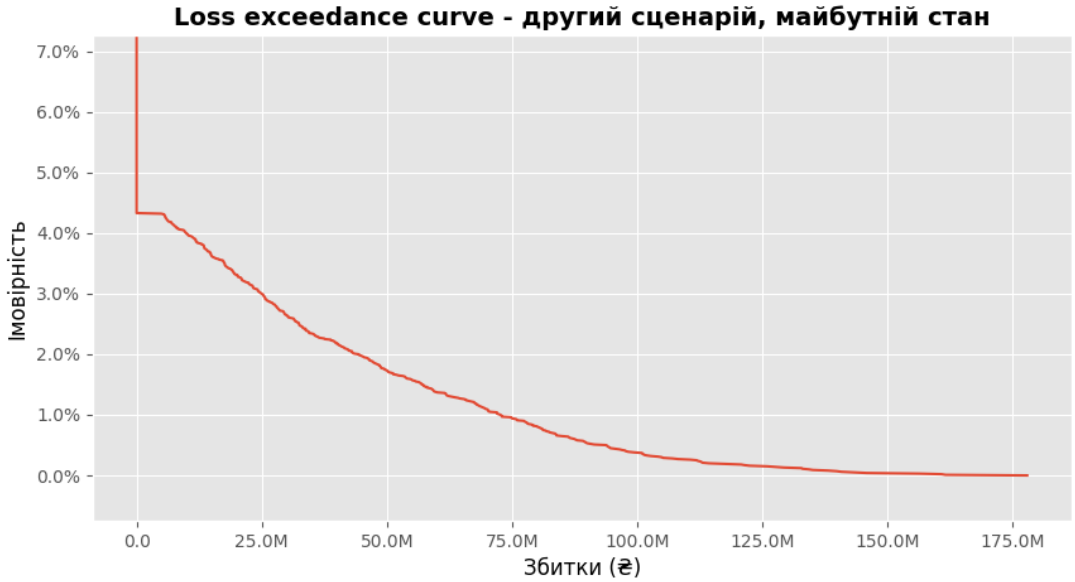

Рисунок 3.11 – Крива перевищення втрат другого сценарію за майбутнього стану

Для третього сценарію графік щільності розподілу ймовірності значень збитків має такий вигляд, як і для першого (рис. 3.2). Отримані значення збитків показані на рисунку 3.12. Зокрема, отримано такі показники:

- мінімальні збитки – 527 105 грн;
- середні збитки – 1 321 064 грн;
- максимальні збитки – 2 334 185 грн.

Крива перевищення втрат для цього сценарію показана на рисунку 3.13.

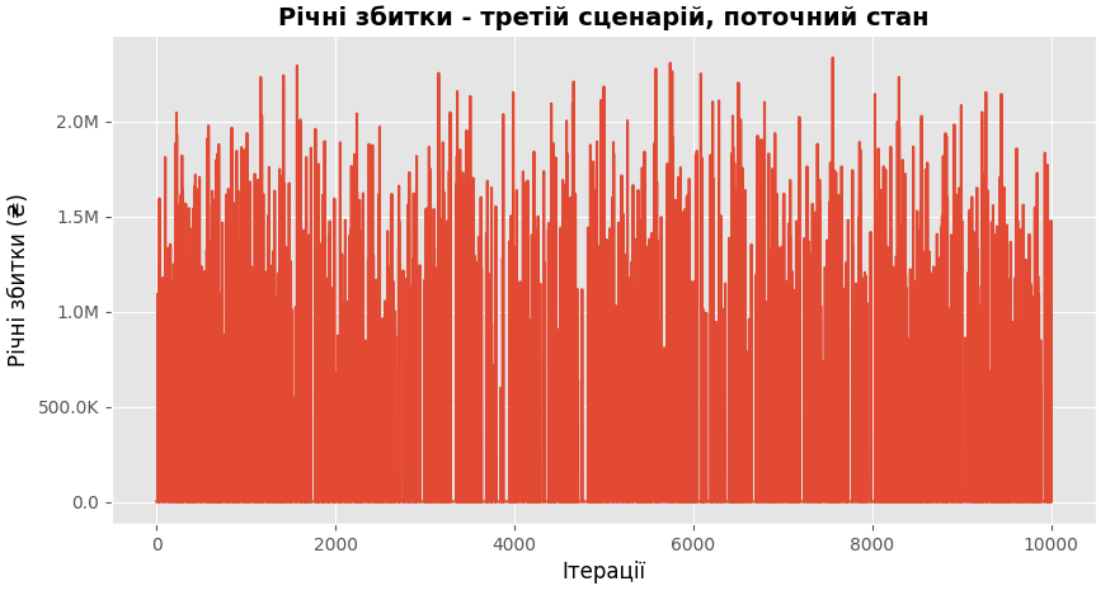

Рисунок 3.12 – Графік згенерованих збитків третього сценарію за поточного стану

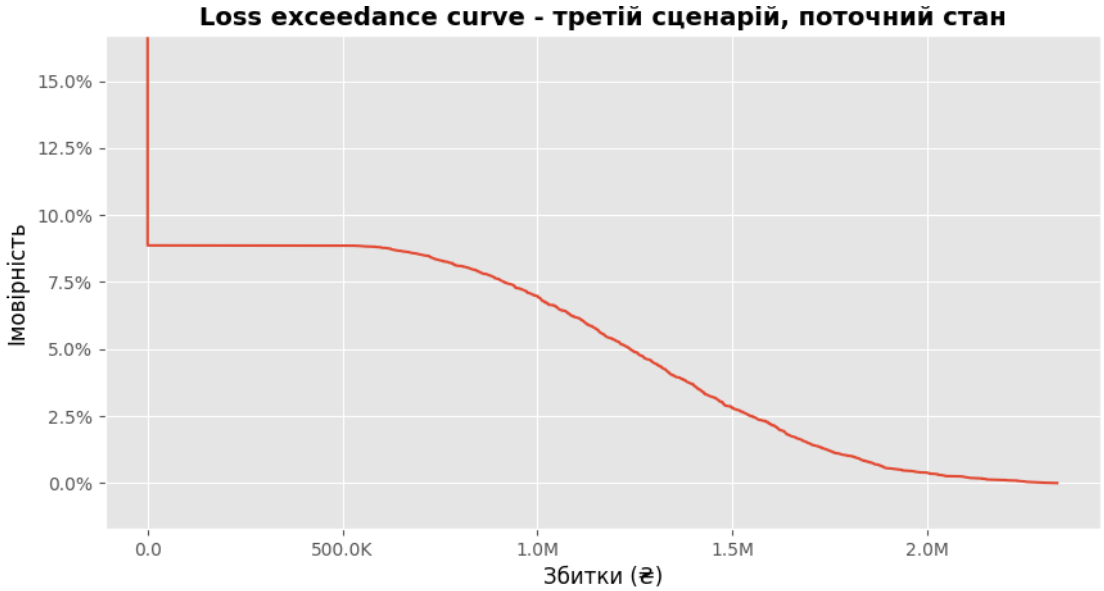

Рисунок 3.13 – Крива перевищення втрат третього сценарію за поточного стану

Згенеровані значення втрат за майбутнього стану засобів захисту показані на рисунку 3.14. Отримано такі показники:

- мінімальні збитки – 557 437 грн;
- середні збитки – 1 329 943 грн;
- максимальні збитки – 2 231 061 грн.

Бачимо, що за цим сценарієм збитки не змінились, але зменшилась імовірність успішного виконання атак cryptojacking (рис. 3.15)

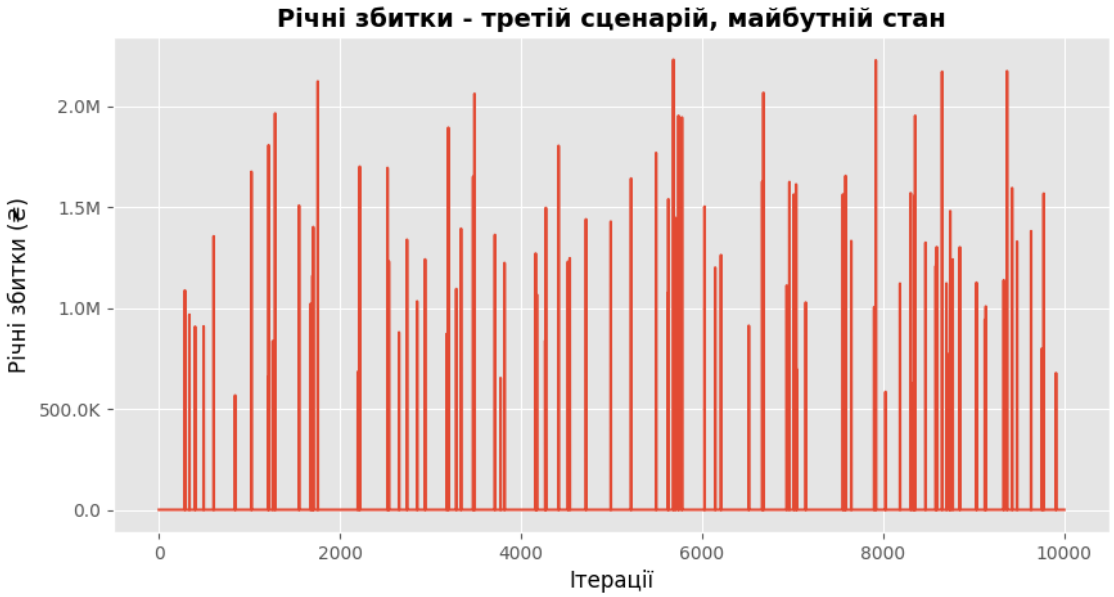

Рисунок 3.14 – Графік згенерованих збитків третього сценарію за майбутнього стану

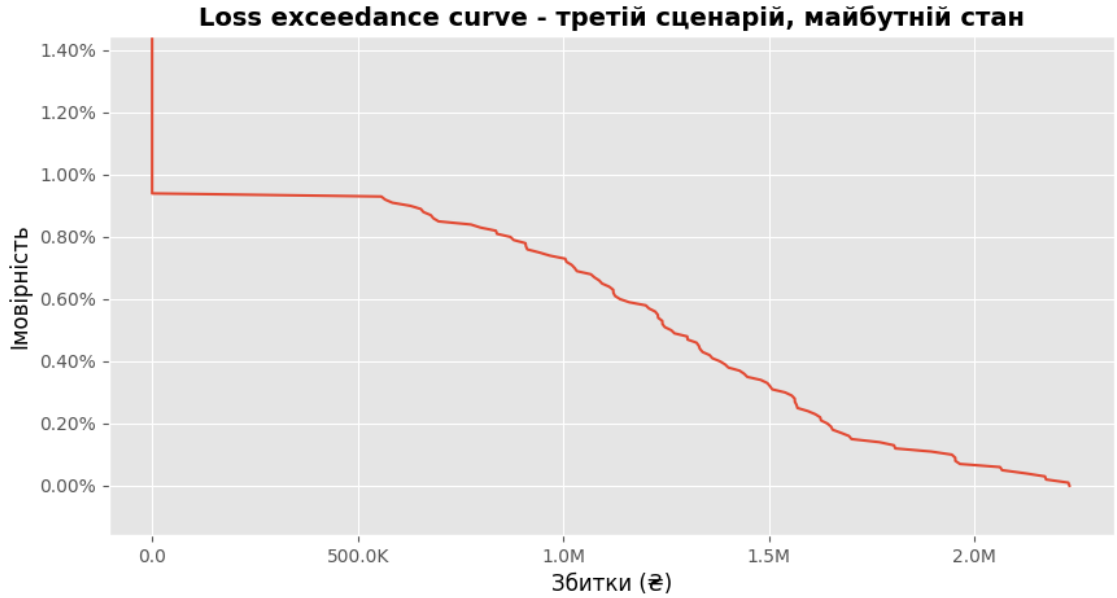

Рисунок 3.15 – Крива перевищення втрат третього сценарію за майбутнього стану

Для четвертого сценарію графік щільності розподілу ймовірності значень збитків показаний на рисунку 3.16.

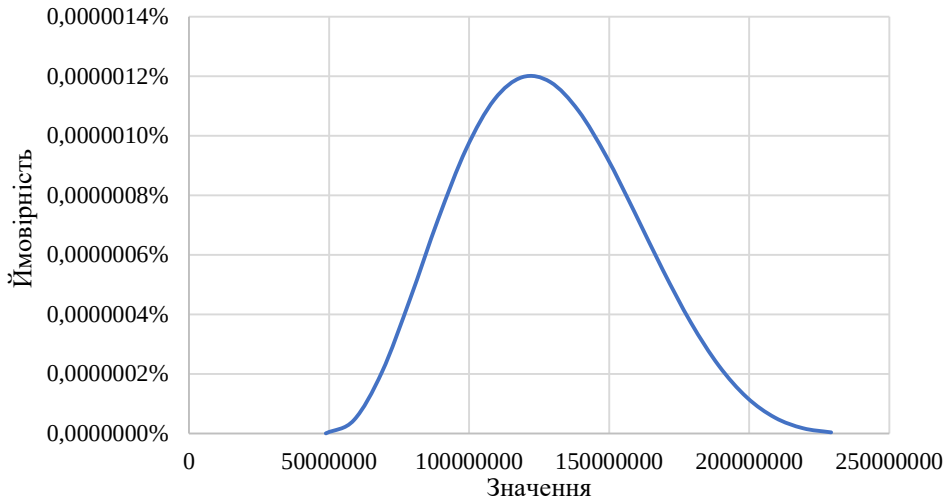

Рисунок 3.16 – Графік щільності розподілу ймовірності для четвертого сценарію

Після виконання симуляції для поточного стану засобів захисту отримано збитки, що показані на графіку річних збитків (рис. 3.17).

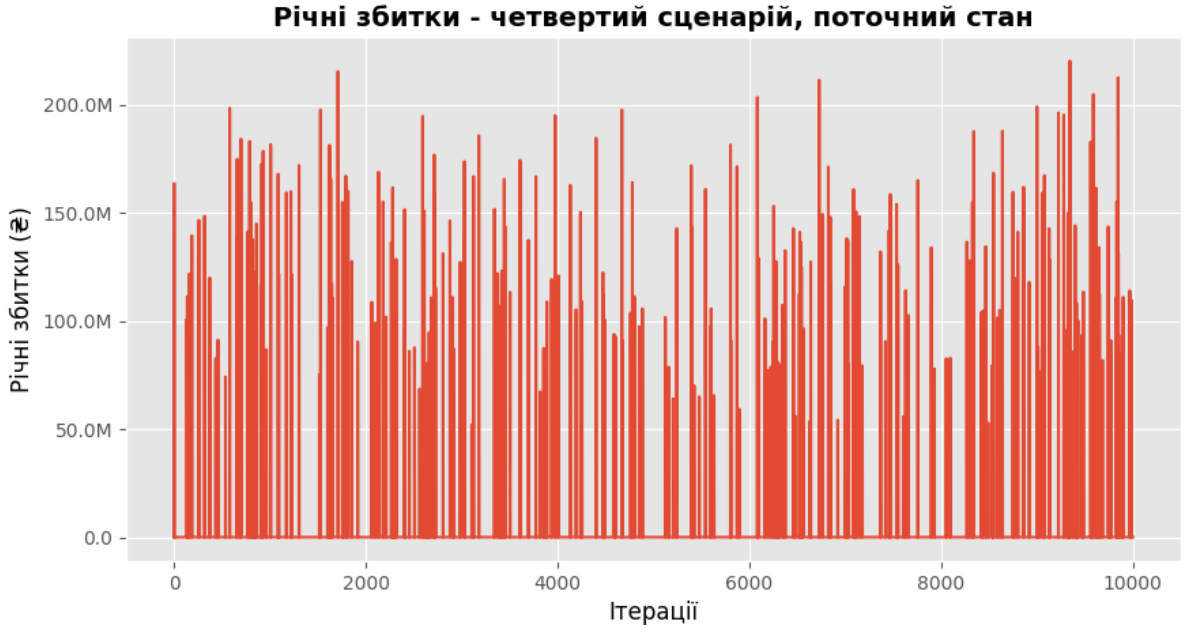

Рисунок 3.17 – Графік згенерованих збитків четвертого сценарію за поточного стану

Зокрема, отримано такі значення:

- мінімальні збитки – 52 004 934 грн;

- середні збитки – 125 641 434 грн;
- максимальні збитки – 220 315 637 грн.

Крива перевищення втрат для цього сценарію показана на рисунку 3.18.

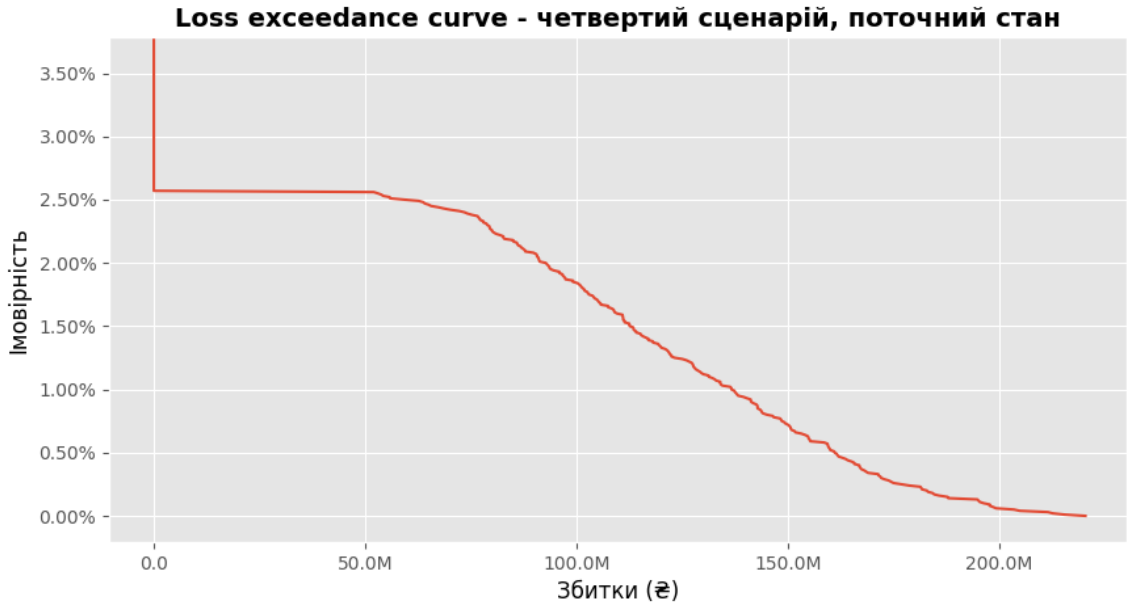

Рисунок 3.18 – Крива перевищення втрат четвертого сценарію за поточного стану

Прогнозовані збитки після зміни стану засобів захисту показано на рисунку 3.19.

Рисунок 3.19 – Графік згенерованих збитків четвертого сценарію за майбутнього стану

Отримано такі показники:

- мінімальні збитки – 64 547 350 грн;
- середні збитки – 131 744 097 грн;
- максимальні збитки – 194 679 173 грн.

Крива перевищення втрат показана на рисунку 3.20.

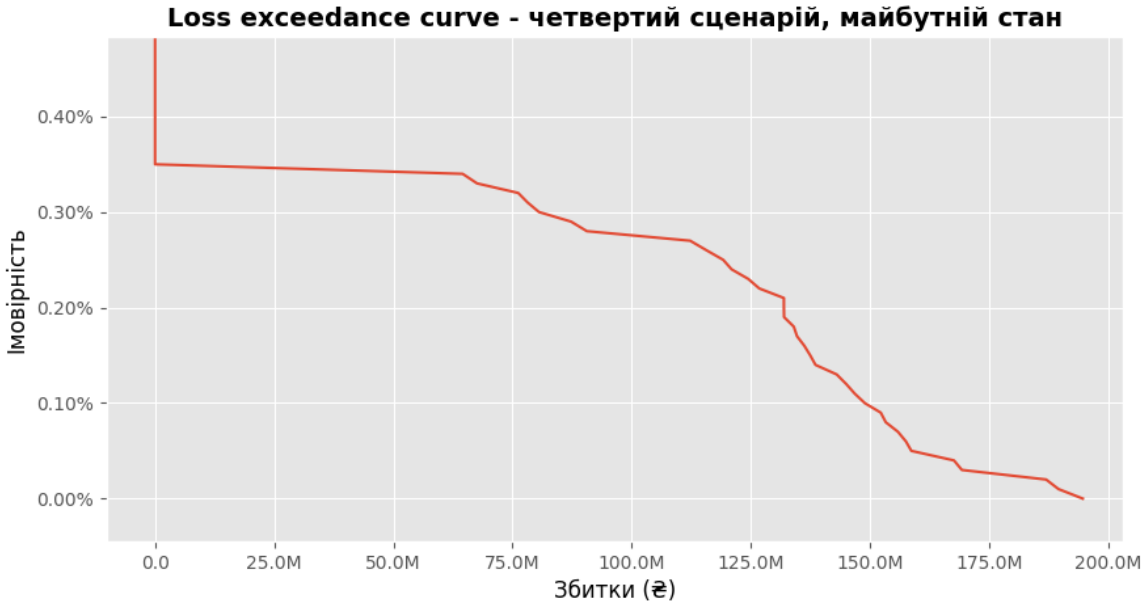

Рисунок 3.20 – Крива перевищення втрат четвертого сценарію за майбутнього стану

Для п’ятого сценарію графік щільності розподілу ймовірності значень збитків показаний на рисунку 3.21.

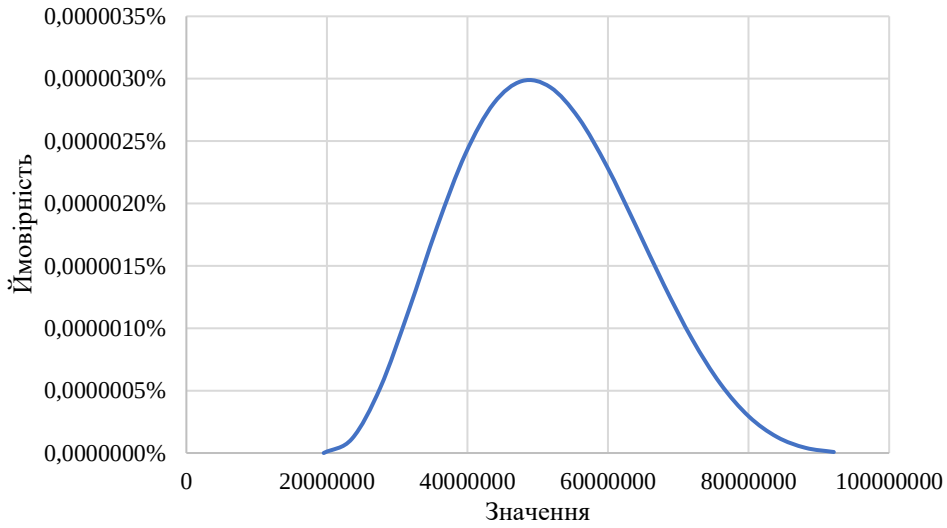

Рисунок 3.21 – Графік щільності розподілу ймовірності для п’ятого сценарію

Значення збитків за результатами симуляції п’ятого сценарію показані на рисунку 3.22. Отримано такі показники збитків:

- мінімальні – 20 463 879 грн;
- середні – 52 554 953 грн;
- максимальні – 95 280 715 грн.

Крива перевищення втрат для цього сценарію показана на рисунку 3.23.

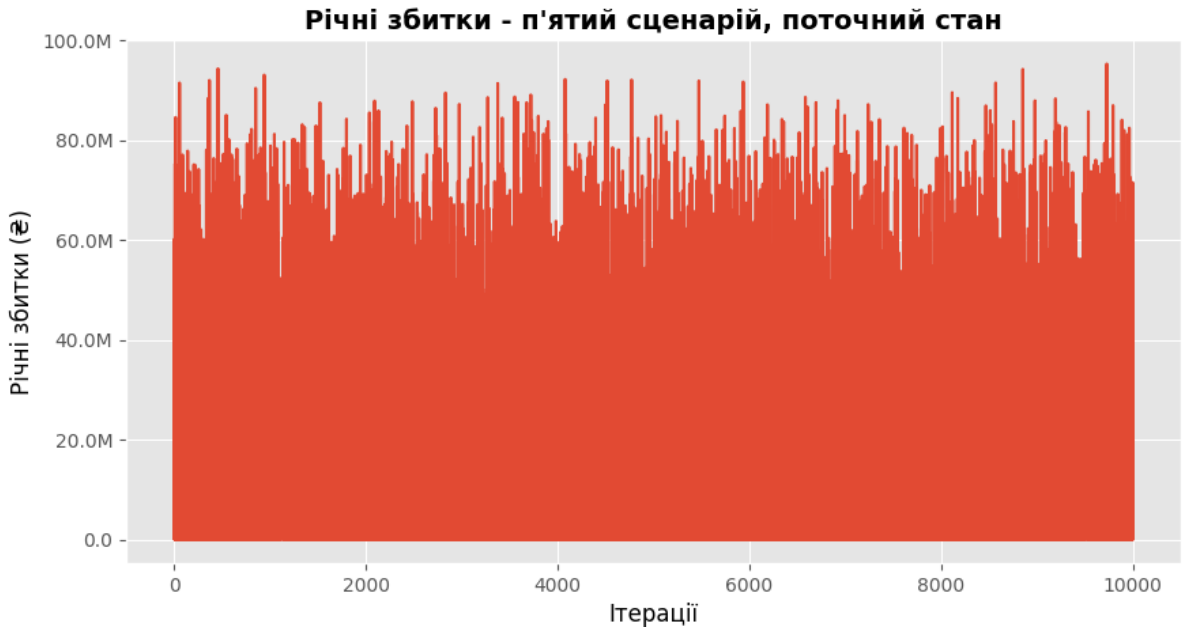

Рисунок 3.22 – Графік згенерованих збитків п’ятого сценарію за поточного стану

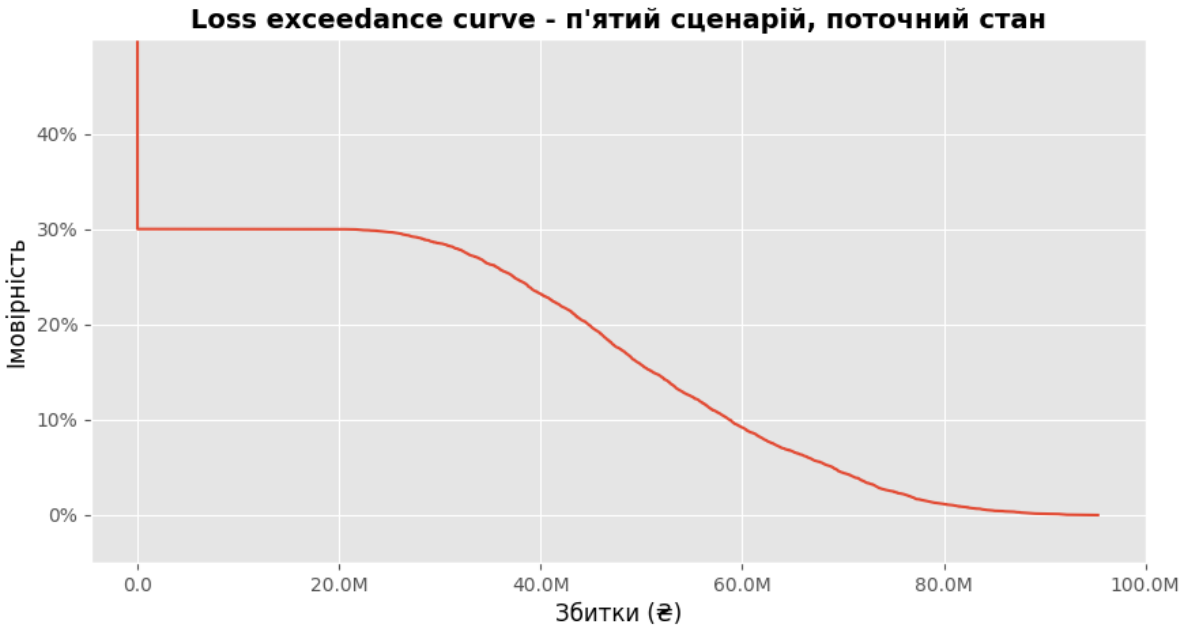

Рисунок 3.23 – Крива перевищення втрат п’ятого сценарію за поточного стану

Згенеровані значення втрат за майбутнього стану засобів захисту показані на рисунку 3.24.

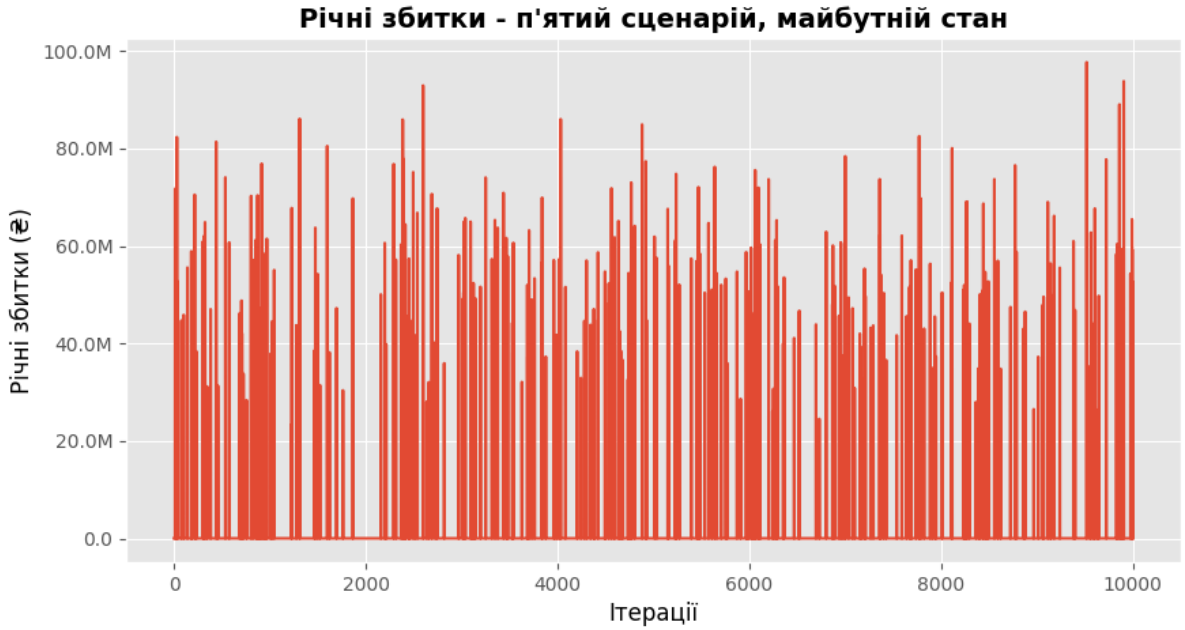

Рисунок 3.24 – Графік згенерованих збитків п’ятого сценарію за майбутнього стану

Зокрема, отримано такі показники:

- мінімальні збитки – 23 330 518 грн;
- середні збитки – 52 441 128 грн;
- максимальні збитки – 97 674 732 грн.

Крива перевищення втрат для майбутнього стану засобів захисту показана на рисунку 3.25. За результатами усіх симуляцій бачимо, що форма кривої втрат змінюється для сценаріїв із низькою ймовірністю успішного виконання атаки, що спричинено через малу кількість згенерованих збитків. Для формування більш плавної кривої втрат може бути встановлено більшу кількість ітерацій симуляції відповідних сценаріїв.

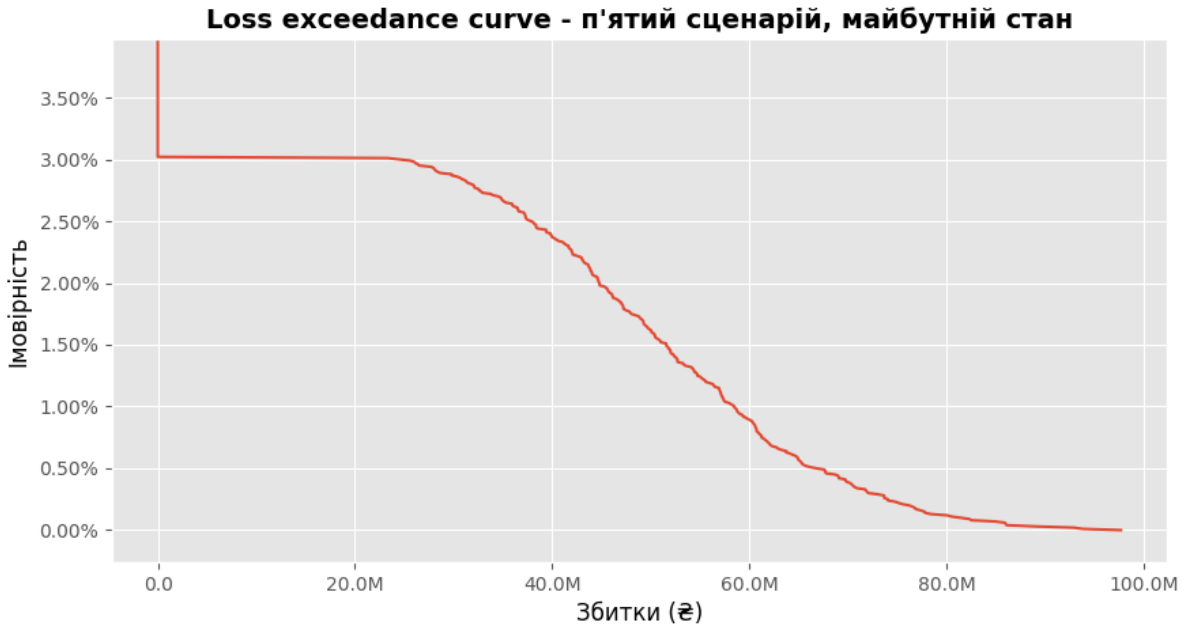

Рисунок 3.25 – Крива перевищення втрат п’ятого сценарію за майбутнього стану

3.4 Аналіз результатів симуляції

За отриманими даними після симуляції виконаємо аналіз результатів. Розрахуємо значення ALE для поточного та майбутнього стану засобів захисту та після цього визначимо ROSI за формулою (1.4). Для першого сценарію за поточного стану маємо такі дані:

- мінімальне ALE: 3 537 грн;
- середнє ALE: 6 565 грн;
- максимальне ALE: 9 796 грн.

Згідно результатів симуляції, після впровадження системи виявлення та блокування майнінгу успішних атак cryptojacking майже не залишиться, тому розраховано лише одне значення:

- середнє ALE: 508 грн.

Якщо припустити, що впровадження рішення з виявлення та блокування майнінгу за рік коштує $5 000 = 5 000 * 39,6702 грн = 198 351 грн, та взяти середнє значення ALE, тоді для цього сценарію отримаємо від’ємне значення ROSI:

ROSI = 6 565 – 508 – 198 351 = – 192 294 грн.

Отже, використання такого рішення лише для захисту від випадкового переходу на сайт, що виконує несанкціонований майнінг не є рентабельним.

Проаналізуємо результати симуляції інших сценаріїв. За поточного стану безпеки для другого сценарію отримано такі дані:

- мінімальне ALE: 4 267 760 грн;
- середнє ALE: 25 974 257 грн;
- максимальне ALE: 65 385 127 грн.

Після впровадження системи виявлення майнінгу маємо такі результати:

- мінімальне ALE: 348 687 грн;
- середнє ALE: 2 101 667 грн;
- максимальне ALE: 5 192 518 грн.

Розрахуємо ROSI для цього сценарію враховуючи середні значення ALE:

ROSI = 25 974 257 – 2 101 667 – 198 351 = 23 674 239 грн.

Для третього сценарію за поточного стану засобів захисту отримано такі дані:

- мінімальне ALE: 63 475 грн;
- середнє ALE: 117 178 грн;
- максимальне ALE: 177 175 грн.

Після впровадження системи виявлення майнінгу маємо такі результати:

- мінімальне ALE: 5 963 грн;
- середнє ALE: 12 501 грн;
- максимальне ALE: 19 673 грн.

Розрахуємо ROSI для третього сценарію враховуючи середні значення ALE:

ROSI = 117 178 – 12 501 – 198 351 = – 93 674 грн.

Бачимо, що для цього сценарію впровадження системи виявлення та блокування майнінгу також є збитковим.

Для четвертого сценарію за поточного стану засобів захисту отримано такі дані:

- мінімальне ALE: 1 716 243 грн;
- середнє ALE: 3 228 984 грн;
- максимальне ALE: 4 980 127 грн.

За майбутнього стану засобів захисту маємо такі результати:

- мінімальне ALE: 243 031 грн;
- середнє ALE: 461 104 грн;
- максимальне ALE: 647 971 грн.

Розрахуємо ROSI для четвертого сценарію враховуючи середні значення ALE:

ROSI = 3 228 984 – 461 104 – 198 351 = 2 569 529 грн.

Натомість, за четвертим сценарієм впровадження такого засобу є цілком доречним. Враховуючи, що він є доповненням третього сценарію можемо розрахувати загальне значення ROSI:

ROSI = 2 569 529 – 93 674 = 2 475 855 грн.

Отже, згідно з результатами двох пов’язаних сценаріїв повернення інвестицій є досить високими.

Для п’ятого сценарію за поточного стану засобів захисту отримано такі дані:

- мінімальне ALE: 8 831 535 грн;
- середнє ALE: 15 782 252 грн;
- максимальне ALE: 23 914 160 грн.

За майбутнього стану засобів захисту маємо такі результати:

- мінімальне ALE: 891 228 грн;
- середнє ALE: 1 583 722 грн;
- максимальне ALE: 2 416 233 грн.

Розрахуємо ROSI для п’ятого сценарію враховуючи середні значення ALE:

ROSI = 15 782 252 – 1 583 722 – 198 351 = 14 000 179 грн.

Отриманий результат показує, що за п’ятим сценарієм повернення інвестицій в безпеку є найбільшим. Усі сценарії передбачають впровадження однієї системи захисту, тому можемо розрахувати загальний розмір ROSI:

ROSI = – 192 294 + 23 674 239 – 93 674 + 2 569 529 + 14 000 179 = 39 957 979 грн.

Ці результати показують, що для дослідженої організації впровадження засобу виявлення та блокування майнінгу на основі машинного навчання дозволить суттєво зменшити ризик атак cryptojacking, а також забезпечить високий рівень повернення інвестицій згідно середніх показників ALE.

Висновки за розділом 3

У третьому розділі виконано оцінку ризиків атак cryptojacking та проведено аналіз результатів згідно розробленої моделі на прикладі деякого підприємства. Було розраховано діапазон збитків за кожним сценарієм атаки для поточного стану засобів захисту, а також для майбутнього разом із ймовірністю цих сценаріїв згідно розроблених дерев відмов.

На основі розрахованих значень було виконано симуляцію Монте-Карло та отримано графіки із прогнозованими втратами та loss exceedance curve для кожного сценарію. Отримані дані показали, за більшістю сценаріїв розміри збитків майже не змінились до та після впровадження рішення з виявлення cryptojacking, але суттєво змінилась імовірність успішного виконання цих атак.

Після виконання симуляції було розраховано значення ризиків у вигляді показника ALE, зокрема середнього, мінімального на основі 10% найменших збитків та максимального на основі 10% найбільших значень збитків. Ці розрахунки показали, що рівень ризику знизився для майбутнього стану засобів захисту. Також було розраховано показник повернення інвестицій в безпеку – ROSI. Отримані результати дозволяють зробити висновок, що за деякими сценаріями впровадження засобу виявлення атак cryptojacking на основі машинного навчання є невигідним, адже інвестиції перевищують рівень зменшення ризику. Натомість, інші сценарії показали значний розмір повернення інвестицій, що дозволяє аргументувати бюджет для обраного засобу захисту. Отже, отримані результати надають можливість визначити, як можна оптимізувати ресурси для захисту та для яких сценаріїв атак їх варто впроваджувати.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано актуальне наукове завдання щодо розробки нових моделей оцінки ризиків атак cryptojacking, що стають дедалі популярнішими серед кіберзлочинців. Під час розв’язання поставлених задач були отримані такі наукові та практичні результати:

1. Проведено аналіз атак cryptojacking та сучасних методів їх виявлення та блокування. Визначено показники точності різних підходів до ідентифікації несанкціонованого майнінгу на основі існуючих досліджень. Під час вибору та впровадження рішень з виявлення таких атак, а також планування інвестицій необхідно застосовувати методи управління та оцінки ризиків. Такий підхід дозволяє організаціям приймати обґрунтовані рішення.

2. Проаналізовано способи оцінки ризиків та запропоновано використовувати саме кількісні методи, що надають числові значення збитків, а також визначено вхідні дані.

3. Запропоновано алгоритм кількісної оцінки ризиків атак cryptojacking для підприємств, який включає в себе метод fault tree analysis та симуляцію Монте-Карло. Такий алгоритм дозволяє врахувати випадкову природу таких атак, адже вони можуть виконуватись на різних пристроях в організації протягом різного періоду часу, а також спричиняти різні збитки.

4. Розроблено модель для оцінки ризиків атак cryptojacking, що включає в себе п’ять сценаріїв, а також дерева рішень для визначення ймовірності виконання атак, що розроблені згідно методу fault tree analysis.

5. Для сформованої моделі запропоновано трикутний розподіл ймовірностей, або розподіл PERT, які застосовуються під час формування випадкових значень симуляції Монте-Карло.

6. Розроблено програмний засіб, який призначений для виконання симуляції Монте-Карло та аналізу отриманих даних, зокрема, побудову графіків згенерованих

значень збитків, перевищення витрат (loss exceedance curve), розрахунок значень ризику як показник ALE.

7. Проведено оцінку ризиків для обраного підприємства на основі розробленої моделі та розраховано показники повернення інвестицій в безпеку (ROSI), що дозволяють керівництву приймати рішення щодо вибору засобів захисту. Було підтверджено доцільність використання моделі оцінки ризиків атак cryptojacking.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tekiner E., Acar A., Uluagac A. S. et al. SoK: Cryptojacking Malware. *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. P. 120–139. DOI:10.1109/EuroSP51992.2021.00019.

2. Zhang R., Xue R., Liu L. Security and Privacy on Blockchain. *ACM Computing Surveys*. 2019. Vol. 52, Issue 3. P. 51:1-51:34. DOI:10.1145/3316481.

3. Schinckus C. Proof-of-work based blockchain technology and Anthropocene: An undermined situation? *Renewable and Sustainable Energy Reviews*. 2021. Vol. 152, 01.12.2021. P. 111682. DOI:10.1016/j.rser.2021.111682.

4. Borys A., Kamruzzaman A., Thakur H. N. et al. An Evaluation of IoT DDoS Cryptojacking Malware and Mirai Botnet. *2022 IEEE World AI IoT Congress (AIIoT)*. P. 725–729. DOI:10.1109/AIIoT54504.2022.9817163.

5. ENISA Threat Landscape 2023. *ENISA*. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023 (accessed 06.02.2024).

6. Kshetri N., Rahman M. M., Sayeed S. A. et al. cryptoRAN: A review on cryptojacking and ransomware attacks w.r.t. banking industry – threats, challenges, & problems. arXiv, 2023. DOI:10.48550/arXiv.2311.14783.

7. 2022 Cloud-Native Threat Report. *Sysdig*. URL: https://sysdig.com/resources/reports/2022-cloud-native-threat-report/ (accessed 06.05.2024).

8. Sanda O., Pavlidis M., Polatidis N. A deep learning approach for host-based cryptojacking malware detection. *Evolving Systems*. 2023. Vol. 15, 19.08.2023. P. 1–16. DOI:10.1007/s12530-023-09534-9.

9. Malik A. W., Anwar Z. Do Charging Stations Benefit from Cryptojacking? A Novel Framework for Its Financial Impact Analysis on Electric Vehicles. *Energies*. 2022. Vol. 15, Issue 16. P. 5773. DOI:10.3390/en15165773.

10. Huang K. Y. Security 101: The Impact of Cryptocurrency-Mining Malware - Security News. *Trend Micro*. 05.07.2017. URL: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/security-101-the-impact-of-cryptocurrency-mining-malware (accessed 06.05.2024).

11. Caprolu M., Raponi S., Oligeri G. et al. Cryptomining Makes Noise: a Machine Learning Approach for Cryptojacking Detection. *Computer Communications*. 2021. Vol. 171, 04.2021. P. 126–139. DOI:10.1016/j.comcom.2021.02.016.

12. Khiruparaj T. P., Abishek Madhu V., Sathia Bhama P. R. K. Unmasking File-Based Cryptojacking. *Intelligence in Big Data Technologies—Beyond the Hype*Singapore : Springer, 2021. P. 137–146. DOI:10.1007/978-981-15-5285-4_13.

13. Kelton C., Balasubramanian A., Raghavendra R. et al. Browser-Based Deep Behavioral Detection of Web Cryptomining with CoinSpy. *Workshop on Measurements, Attacks, and Defenses for the Web*. San Diego, CA : Internet Society, 2020. DOI:10.14722/madweb.2020.23002.

14. Sivaraju S. S. An Insight into Deep Learning based Cryptojacking Detection Model. *Journal of Trends in Computer Science and Smart Technology*. 2022. Vol. 4, Issue 3. P. 175–184. URL: https://irojournals.com/tcsst/article/view/4/3/6 (accessed 05.04.2024).

15. Carlin D., Burgess J., O'Kane P. et al. You Could Be Mine(d): The Rise of Cryptojacking. *IEEE Security & Privacy*. 2020. Vol. 18, Issue 2. P. 16–22. DOI:10.1109/MSEC.2019.2920585.

16. Bijmans H. L. J., Booij T. M., Doerr C. Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale. *28th USENIX Security Symposium (USENIX Security 19)*. (2019). P. 1627–1644. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/bijmans (accessed 13.04.2024).

17. Arış A., Naseem F., Babun L. et al. MINOS: A Lightweight Real-Time Cryptojacking Detection System. *Proceedings 2021 Network and Distributed System Security Symposium*. (23.02.2021). DOI:10.14722/ndss.2021.24444.

18. Bijmans H. L. J., Booij T. M., Doerr C. Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*New York, NY, USA : Association for Computing Machinery, 2019. P. 449–464. DOI:10.1145/3319535.3354230.

19. Sharma H. Implementing Cryptojacking as a Web Monetization Model for Increased Privacy : masters. Dublin, National College of Ireland, 2023. 39 p. URL: https://norma.ncirl.ie/6542/ (accessed 05.02.2024).

20. Rüth J., Zimmermann T., Wolsing K. et al. Digging into Browser-based Crypto Mining. *Proceedings of the Internet Measurement Conference 2018*P. 70–76. DOI:10.1145/3278532.3278539.

21. 2024 SonicWall Cyber Threat Report. *SonicWall*. URL: https://www.sonicwall.com/threat-report/ (accessed 23.03.2024).

22. Hong G., Yang Z., Yang S. et al. How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*New York, NY, USA : Association for Computing Machinery, 2018. P. 1701–1713. DOI:10.1145/3243734.3243840.

23. Varlioglu S., Elsayed N., ElSayed Z. et al. The Dangerous Combo: Fileless Malware and Cryptojacking. *SoutheastCon 2022*. P. 125–132. DOI:10.1109/SoutheastCon48659.2022.9764043.

24. Varlioglu S., Elsayed N., Varlioglu E. R. et al. The Pulse of Fileless Cryptojacking Attacks: Malicious PowerShell Scripts. arXiv, 2024. DOI:10.48550/arXiv.2401.07995.

25. Liang J., Kim Y. Evolution of Firewalls: Toward Securer Network Using Next Generation Firewall. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. P. 0752–0759. DOI:10.1109/CCWC54503.2022.9720435.

26. URL Filtering. *Cisco Firepower Management Center Configuration Guide, Version 6.3*. URL: https://www.cisco.com/c/en/us/td/docs/security/firepower/630/configuration/guide/fpmc-config-guide-v63/url_filtering.html (accessed 03.04.2024).

27. Hu X., Shu Z., Song X. et al. Detecting Cryptojacking Traffic Based on Network Behavior Features. *2021 IEEE Global Communications Conference (GLOBECOM)*. P. 01–06. DOI:10.1109/GLOBECOM46510.2021.9685085.

28. Muñoz J. Z. i, Suárez-Varela J., Barlet-Ros P. Detecting cryptocurrency miners with NetFlow/IPFIX network measurements. *2019 IEEE International Symposium on Measurements & Networking (M&N)*. P. 1–6. DOI:10.1109/IWMN.2019.8804995.

29. Song W., Beshley M., Przystupa K. et al. A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection. *Sensors*. 2020. Vol. 20, Issue 6. P. 1637. DOI:10.3390/s20061637.

30. Dong L., Li Z., Li X. et al. Identification technique of cryptomining behavior based on traffic features. *Frontiers in Physics*. 2023. Vol. 11, 2023. URL: https://www.frontiersin.org/articles/10.3389/fphy.2023.1269889 (accessed 18.02.2024).

31. Russo M., Šrndić N., Laskov P. Detection of illicit cryptomining using network metadata. *EURASIP Journal on Information Security*. 2021. Vol. 2021, Issue 1. P. 11. DOI:10.1186/s13635-021-00126-1.

32. Aslan Ö. A., Samet R. A Comprehensive Review on Malware Detection Approaches. *IEEE Access*. 2020. Vol. 8, 2020. P. 6249–6271. DOI:10.1109/ACCESS.2019.2963724.

33. Mani G., Pasumarti V., Bhargava B. et al. DeCrypto Pro: Deep Learning Based Cryptomining Malware Detection Using Performance Counters. *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. P. 109–118. DOI:10.1109/ACSOS49614.2020.00032.

34. Zhang S., Hu C., Wang L. et al. A Malware Detection Approach Based on Deep Learning and Memory Forensics. *Symmetry*. 2023. Vol. 15, Issue 3. P. 758. DOI:10.3390/sym15030758.

35. Yazdinejad A., HaddadPajouh H., Dehghantanha A. et al. Cryptocurrency malware hunting: A deep Recurrent Neural Network approach. *Applied Soft Computing*. 2020. Vol. 96, 01.11.2020. P. 106630. DOI:10.1016/j.asoc.2020.106630.

36. Karn R., Kudva P., Huang H. et al. Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning. *IEEE Transactions on Parallel and Distributed Systems*. 2021. Vol. 32, 01.03.2021. P. 674–691. DOI:10.1109/TPDS.2020.3029088.

37. Gomes G., Dias L., Correia M. CryingJackpot: Network Flows and Performance Counters against Cryptojacking. *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*. P. 1–10. DOI:10.1109/NCA51143.2020.9306698.

38. Shareef M. N., Khan J. H. M. K. A. A., Bari M. A. Crypto Jacking. *Mathematical Statistician and Engineering Applications*. 2023. Vol. 72, Issue 1. P. 1581–1586. DOI:10.17762/msea.v72i1.2387.

39. Rodriguez J. D. P., Posegga J. RAPID: Resource and API-Based Detection Against In-Browser Miners. *Proceedings of the 34th Annual Computer Security Applications Conference*New York, NY, USA : Association for Computing Machinery, 2018. P. 313–326. DOI:10.1145/3274694.3274735.

40. Romano A., Zheng Y., Wang W. MinerRay: semantics-aware analysis for ever-evolving cryptojacking detection. *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*New York, NY, USA : Association for Computing Machinery, 2021. P. 1129–1140. DOI:10.1145/3324884.3416580.

41. Xu D., Ming J., Wu D. Cryptographic Function Detection in Obfuscated Binaries via Bit-Precise Symbolic Loop Mapping. *2017 IEEE Symposium on Security and Privacy (SP)*. P. 921–937. DOI:10.1109/SP.2017.56.

42. Tekiner E., Acar A., Uluagac S. A Lightweight IoT Cryptojacking Detection Mechanism in Heterogeneous Smart Home Networks. *Proceedings 2022 Network and Distributed System Security Symposium*. 2022. 2022. DOI:10.14722/ndss.2022.24208.

43. Tahir R., Huzaifa M., Das A. та ін. Mining on Someone Else's Dime: Mitigating Covert Mining Operations in Clouds and Enterprises. *Research in Attacks, Intrusions, and Defenses*Cham : Springer International Publishing, 2017. С. 287–310. DOI:10.1007/978-3-319-66332-6_13.

44. ISACA Interactive Glossary. *ISACA*. URL: https://www.isaca.org/resources/glossary (accessed 07.05.2024).

45. Rot A. IT risk assessment: quantitative and qualitative approach. *Proceedings of the World Congress on Engineering and Computer Science 2008*. 2008. Issue March. P. 284. URL:

https://www.researchgate.net/publication/44262457_IT_Risk_Assessment_Quantitative_and_Qualitative_Approach (accessed 31.03.2024).

46. Schmidt M. Information security risk management terminology and key concepts. *Risk Management*. 2022. Vol. 25, 16.12.2022. DOI:10.1057/s41283-022-00108-8.

47. Interoperable EU Risk Management Framework. *ENISA*. URL: https://www.enisa.europa.eu/publications/interoperable-eu-risk-management-framework (accessed 25.03.2024).

48. Risk IT Framework. 2nd Edition. ISACA, 2020. URL: https://store.isaca.org/s/store#/store/browse/detail/a2S4w000004Ko9VEAS (accessed 31.03.2024).

49. Kuzminykh I., Ghita B. V., Sokolov V. et al. Information Security Risk Assessment. 2021. Vol. 1, 24.07.2021. P. 602–617. DOI:10.3390/encyclopedia1030050.

50. Munteanu A. Information Security Risk Assessment: The Qualitative Versus Quantitative Dilemma. *Proceedings of the 6th International Business Information Management Association (IBIMA) Conference*. 2006. 20.07.2006. P. 227–232. URL: https://papers.ssrn.com/abstract=917767 (accessed 20.02.2024).

51. Crotty J., Daniel E. Cyber threat: its origins and consequence and the use of qualitative and quantitative methods in cyber risk assessment. *Applied Computing and Informatics*. 2022. 01.01.2022. DOI:10.1108/ACI-07-2022-0178.

52. Initiative J. T. F. T. Guide for Conducting Risk Assessments. National Institute of Standards and Technology, 2012. DOI:10.6028/NIST.SP.800-30r1.

53. Böhme R., Nowey T. Economic Security Metrics. *Dependability Metrics: Advanced Lectures*. ed. I. Eusgeld., F. C. Freiling., R. Reussner. Berlin, Heidelberg : Springer, 2008. P. 176–187. DOI:10.1007/978-3-540-68947-8_15.

54. Eggers S., Le Blanc K. Survey of cyber risk analysis techniques for use in the nuclear industry. *Progress in Nuclear Energy*. 2021. Vol. 140, 01.10.2021. P. 103908. DOI:10.1016/j.pnucene.2021.103908.

55. Krisper M., Dobaj J., Macher G. et al. RISKEE: A Risk-Tree Based Method for Assessing Risk in Cyber Security. 2019. P. 45–56. DOI:10.1007/978-3-030-28005-5_4.

56. Степанюк З. А. Огляд сучасних методів кількісної оцінки ризиків та впровадження інформаційних технологій в систему ризик-менеджменту. *Економіка. Менеджмент. Бізнес*. 2017. Вип. 4. С. 173–181. URL: https://journals.dut.edu.ua/index.php/emb/article/view/1678 (дата звернення: 11.03.2024).

57. Sokri A. Cyber security risk modelling and assessment: A quantitative approach. *18th European Conference on Cyber Warfare and Security*. (Coimbra, Portugal, 2019). Coimbra, Portugal : University of Coimbra, 2019. P. 466–474. URL: https://www.proquest.com/openview/6e458070f7f2ffd24f6149787e472539/1?pq-origsite=gscholar&cbl=396497 (accessed 02.04.2024).

58. Senova A., Tobisova A., Rozenberg R. New Approaches to Project Risk Assessment Utilizing the Monte Carlo Method. *Sustainability*. 2023. Vol. 15, Issue 2. P. 1006. DOI:10.3390/su15021006.

59. Maidana R. G., Parhizkar T., Gomola A. et al. Supervised dynamic probabilistic risk assessment: Review and comparison of methods. *Reliability Engineering & System Safety*. 2023. Vol. 230, 01.02.2023. P. 108889. DOI:10.1016/j.ress.2022.108889.

60. Babeshko I., Giandomenico F. D. Safety and Cybersecurity Assessment Techniques for Critical Industries: A Mapping Study. *IEEE Access*. 2023. Vol. 11, 2023. P. 83781–83793. DOI:10.1109/ACCESS.2023.3297446.

61. Ruijters E., Stoelinga M. Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer Science Review*. 2015. Vol. 15–16, 01.02.2015. P. 29–62. DOI:10.1016/j.cosrev.2015.03.001.

62. Haller P., Genge B. Using Sensitivity Analysis and Cross-Association for the Design of Intrusion Detection Systems in Industrial Cyber-Physical Systems. *IEEE Access*. 2017. Vol. 5, 2017. P. 9336–9347. DOI:10.1109/ACCESS.2017.2703906.

63. Le A., Chen Y., Chai K. K. et al. Assessing Loss Event Frequencies of Smart Grid Cyber Threats: Encoding Flexibility into FAIR Using Bayesian Network Approach. *Smart Grid Inspired Future Technologies*Cham : Springer International Publishing, 2017. P. 43–51. DOI:10.1007/978-3-319-47729-9_5.

64. The Importance and Effectiveness of Cyber Risk Quantification. *FAIR Institute*. URL: https://www.fairinstitute.org/what-is-fair (accessed 01.05.2024).

65. Wang J., Neil M., Fenton N. A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model. *Computers & Security*. 2020. Vol. 89, 01.02.2020. P. 101659. DOI:10.1016/j.cose.2019.101659.

66. Hong H., Woo S., Park S. et al. Circuit: A JavaScript Memory Heap-Based Approach for Precisely Detecting Cryptojacking Websites. *IEEE Access*. 2022. Vol. 10, 2022. P. 95356–95368. DOI:10.1109/ACCESS.2022.3204814.

67. Xu G., Dong W., Xing J. et al. Delay-CJ: A novel cryptojacking covert attack method based on delayed strategy and its detection. *Digital Communications and Networks*. 2023. Vol. 9, Issue 5. P. 1169–1179. DOI:10.1016/j.dcan.2022.04.030.

68. Eskandari S., Leoutsarakos A., Mursch T. et al. A First Look at Browser-Based Cryptojacking. *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. P. 58–66. DOI:10.1109/EuroSPW.2018.00014.

69. Огірко О. І., Галайко Н. В. Теорія ймовірностей та математична статистика: навчальний посібник. Львів : ЛьвДУВС, 2017. 292 с. URL: https://dspace.lvduvs.edu.ua/handle/1234567890/629 (дата звернення: 22.04.2024).

70. Erola A., Agrafiotis I., Nurse J. R. C. et al. A system to calculate Cyber Value-at-Risk. *Computers & Security*. 2022. Vol. 113, 01.02.2022. P. 102545. DOI:10.1016/j.cose.2021.102545.

71. Hosseini Bamakan S. M., Dehghanimohammadabadi M. A Weighted Monte Carlo Simulation Approach to Risk Assessment of Information Security Management System. *International Journal of Enterprise Information Systems*. 2016. Vol. 11, 10.01.2016. P. 63–78. DOI:10.4018/IJEIS.2015100103.

72. Walpole R. E., Myers R. H., Myers S. L. et al. Probability and Statistics for Engineers and Scientists. 9th edition. Pearson Education, 2011. 812 p. ISBN 978-0-321-83144-6.

73. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 10th edition. John Wiley & Sons, 2009. 1122 p. ISBN 978-0-470-50383-6.

74. Zheng H., Tang Y. A Novel Failure Mode and Effects Analysis Model Using Triangular Distribution-Based Basic Probability Assignment in the Evidence Theory. *IEEE Access*. 2020. Vol. 8, 2020. P. 66813–66827. DOI:10.1109/ACCESS.2020.2986807.

75. Udoumoh E. F., Ebong D. W. A Review of Activity Time Distributions in Risk Analysis. *American Journal of Operations Research*. 2017. Vol. 07, Issue 06. P. 356. DOI:10.4236/ajor.2017.76027.

76. Zhang S., Wang X. Dynamic Probability Analysis for Construction Schedule Using Subset Simulation. *Advances in Civil Engineering*. 2021. Vol. 2021, 09.09.2021. P. 1–13. DOI:10.1155/2021/1567261.

77. Sullivan S. Monte Carlo simulation for cyber threats portfolio. *GitHub*. URL: https://github.com/etherpixie/data_science_for_cyber_security/blob/main/Monte%20Carlo%20for%20Cyber-Threats.ipynb (accessed 24.04.2024).

78. Ціни/Тарифи. *Київська обласна енергопостачальна компанія*. URL: https://koec.com.ua/page?root=23 (дата звернення: 27.04.2024).

79. Hire the best CG Artists in Kyiv, UA. *Upwork*. 26.04.2024. URL: https://www.upwork.com/hire/cg-artists/ua/kyiv/ (accessed 27.04.2024).

80. Neely L., Torres A. Endpoint Protection and Response: A SANS Survey. (06.2018). SANS Institute, 2018. URL: https://web.archive.org/web/20180710105541/https://www.sans.org/reading-room/whitepapers/analyst/endpoint-protection-response-survey-38460 (accessed 27.04.2024).

ДОДАТКИ

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Тези наукових доповідей:

1. Панченко М., Лебєдєва Н., Бабенко Т. Оцінка та прогнозування ризиків для атак cryptojacking. *Проблеми кібербезпеки інформаційно-телекомунікаційних систем (PCSITS)* : Тези науково-практ. конф., м. Київ, 26 квіт. 2024 р. Київ, 2024. С. 153–155.

ДОДАТОК Б
ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
import random
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import numpy as np
from pert import PERT

class Scenario():
    def __init__(self, name, probability, devices, lossv):
        self.name = name
        self.probability = probability
        self.devices = devices
        self.lossv = lossv
        self.losses = []

def is_attack_successful(probability):
    return random.random() <= probability

def calculate_loss(lossv):
    #return random.triangular(lossv[0],lossv[1],lossv[2])
    random=PERT(lossv[0],lossv[2],lossv[1])
    return random.rvs()[0]

def simulate_attack(scenario):
    loss = 0
    for i in range(scenario.devices):
        if is_attack_successful(scenario.probability):
            loss += calculate_loss(scenario.lossv)
    return loss

def monte_carlo_simulation(scenarios, iterations):
    for scenario in scenarios:
        for i in range(iterations):
            loss = simulate_attack(scenario)
            scenario.losses.append(loss)

def plot_losses(losses, name):
    plt.figure(figsize=(10,5))
    plt.style.use('ggplot')
    plt.grid(True)
    plt.plot(losses)
    plt.xlabel("Ітерації", fontsize=12, color='black')
    plt.ylabel("Річні збитки (₴)", fontsize=12, color='black')
    plt.title("Річні збитки - {0}".format(name), fontsize=14, fontweight='bold')
    mkfunc = lambda x, pos: '%1.1fM' % (x * 1e-6) if x >= 1e6 else '%1.1fK' % (x *
1e-3) if x >= 1e3 else '%1.1f' % x
    mkformatter = mtick.FuncFormatter(mkfunc)
    plt.gca().yaxis.set_major_formatter(mkformatter)
    plt.show()

def min_max_avg(losses):
    min_loss = np.min(losses)
    print("Minimum loss: ", min_loss)
    avg_loss=np.average(losses)
    print("Average loss: ", avg_loss)
    max_loss = np.max(losses)
```

```
    print("Maximum loss: ", max_loss)

def ale(losses,events,iterations):
    rate = (events/iterations)
    ale_avg = (sum(losses)/events)*rate
    print("Average ALE: ", ale_avg)
    if events > 3:
        tenp = int(len(losses)*0.1)
        if tenp==0: tenp=1
        ninetyp = int(len(losses)*0.9)
        if ninetyp==0: ninetyp=1
        ale_min = (sum(losses[:tenp])/tenp)*rate
        print("Minimum ALE: ", ale_min)
        ale_max = (sum(losses[ninetyp:])/(len(losses)-ninetyp))*rate
        print("Maximum ALE: ", ale_max)

def lec(losses, name):
    losses.sort()
    p = 1 - np.arange(1, len(losses) + 1) / len(losses)
    plt.figure(figsize=(10,5))
    plt.plot(losses, p)
    plt.xlabel("Збитки (₴)", fontsize=12, color='black')
    plt.ylabel("Імовірність", fontsize=12, color='black')
    plt.title("Loss exceedance curve - {0}".format(name), fontsize=14,
fontweight='bold')
    plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))
    mkfunc = lambda x, pos: '%1.1fM' % (x * 1e-6) if x >= 1e6 else '%1.1fK' % (x *
1e-3) if x >= 1e3 else '%1.1f' % x
    mkformatter = mtick.FuncFormatter(mkfunc)
    plt.gca().xaxis.set_major_formatter(mkformatter)
    plt.show()

def analyse(scenarios, iterations):
    for scenario in scenarios:
        print("Scenario {0}".format(scenario.name))
        plot_losses(scenario.losses,scenario.name)
        lec(scenario.losses, scenario.name)
        scenario.losses.sort()
        losses_v= [i for i in scenario.losses if i!=0 ]
        events=len(losses_v)
        if events > 2:
            min_max_avg(losses_v)
            ale(losses_v,events,iterations)

def main():
    scenarios = [
        Scenario( "перший сценарій, поточний стан", 0.000055, 100, [488386.4469504,
2485569.454752, 1220966.117376]),
        Scenario( "перший сценарій, майбутній стан", 0.0000055, 100, [488386.4469504,
2485569.454752, 1220966.117376]),
        Scenario( "другий сценарій, поточний стан", 0.5, 1, [4883864.469504 ,
248556945.4752 , 12209661.17376]),
        Scenario( "другий сценарій, майбутній стан", 0.05, 1, [4883864.469504 ,
248556945.4752 , 12209661.17376]),
        Scenario( "третій сценарій, поточний стан", 0.09, 1, [488386.4469504,
2485569.454752, 1220966.117376]),
        Scenario( "третій сценарій, майбутній стан", 0.009, 1, [488386.4469504,
2485569.454752, 1220966.117376]),
        Scenario( "четвертий сценарій, поточний стан", 0.025596, 1,
[48840745.3335552, 249321182.467776, 122101863.333888]),
        Scenario( "четвертий сценарій, майбутній стан", 0.0025596, 1,
[48840745.3335552, 249321182.467776, 122101863.333888]),
        Scenario( "п'ятий сценарій, поточний стан", 0.3, 1, [19537558.5165312,
100187015.182656, 48843896.291328]),
```

```
        Scenario( "п'ятий сценарій, майбутній стан", 0.03, 1, [19537558.5165312,
100187015.182656, 48843896.291328])
        ]
    iterations = 10000
    monte_carlo_simulation(scenarios, iterations)
    analyse(scenarios, iterations)

if __name__=='__main__':
    main()
```