МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних інформаційних систем

_____Юрій КОНДРАТЕНКО

«____»_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ НЕСАНЦІОНОВАНОГО МАЙНІНГУ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Інтелектуальні інформаційні системи»

Здобувач ____Владислав АТАМАНЮК

«___»_____ 2024 р.

Керівник канд. фіз.-мат. наук, доцент ____Інесса КУЛАКОВСЬКА

«___»_____ 2024 р.

Миколаїв – 2024

Чорноморський національний університет імені Петра Могили

(повне найменування закладу вищої освіти)

Факультет	Комп’ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп’ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем

_____Юрій КОНДРАТЕНКО

«____» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Атаманюка Владислава Геннадійовича

((прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інформаційна система пошуку несанкціонованого майнінгу».

Керівник роботи: Кулаковська Інесса Василівна, доцент кафедри ІІС, канд. фіз.-мат. наук, доцент.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи «16» грудня 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: система автоматизованого пошуку та виявлення несанкціонованого майнінгу; дані про активні процеси системи, включаючи завантаження процесора, оперативної

пам’яті, графічного процесора та обсяг мережевої активності; навчена модель машинного навчання для класифікації процесів як "нормальних" або "підозрілих".

4. Перелік питань, що підлягають розробці: аналіз сучасного стану задачі виявлення несанкціонованого майнінгу в інформаційних системах; огляд існуючих методів аналізу процесів для ідентифікації аномальної активності; розробка та навчання моделі машинного навчання для виявлення підозрілих процесів за визначеними характеристиками; порівняльний аналіз результатів роботи системи та ефективності використання обраних алгоритмів для розв’язання поставленої задачі.

5. Перелік графічних матеріалів: презентація.

Керівник роботи _____ Інесса КУЛАКОВСЬКА

(Особистий підпис) *(Власне ім’я ПРІЗВИЩЕ)*

Здобувач _____ Владислав АТАМАНЮК

(Особистий підпис) *(Власне ім’я ПРІЗВИЩЕ)*

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Інформаційна система пошуку несанкціонованого майнінгу

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	Виконано
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз публікацій та аналогічних систем пошуку несанкціонованого майнінгу	21.06.2024	01.07.2024	Виконано
4	Огляд існуючих архітектур інформаційних систем пошуку несанкціонованого майнінгу	01.09.2024	25.10.2024	Виконано
5	Реалізація технологій ШІ з метою пошуку несанкціонованого майнінгу	26.10.2024	21.11.2024	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	Виконано
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	Виконано
9	Доробка та остаточне оформлення КР	07.12.2024	10.02.2024	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	Виконано

Керівник роботи _____ Інесса КУЛАКОВСЬКА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Владислав АТАМАНЮК
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата складання календарного плану
«19» червня 2024 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Атаманюка Владислава Геннадійовича

на тему: «**ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНІНГУ**»

Кваліфікаційна робота присвячена розробці та реалізації **інформаційної системи пошуку несанкціонованого майнінгу**. В умовах зростання кількості кіберзагроз і випадків незаконного використання обчислювальних ресурсів для майнінгу криптовалют, розробка таких систем є надзвичайно актуальною. Робота спрямована на дослідження методів аналізу ресурсів, виявлення аномальної активності, а також розробку програмного забезпечення для моніторингу обчислювальних систем.

У першому розділі проаналізовано предметну сферу інтелектуальних систем пошуку несанкціонованого майнінгу, зокрема: сучасну ситуацію з нелегальним майнінгом у світі, особливості криптовалют і технологій майнінгу, основні методи виявлення несанкціонованого майнінгу, а також існуючі системи і засоби виявлення таких загроз.

У другому розділі здійснено аналіз засобів розробки програмного забезпечення для інформаційної системи пошуку несанкціонованого майнінгу. Розглянуто функціональні вимоги до системи, обрано методи аналізу аномальної активності, а також проаналізовано принцип роботи та схеми реалізації популярних рішень, таких як MinerBlock і Windows Defender.

У третьому розділі розроблено програмну реалізацію інтелектуальної системи:

- проєктування архітектури системи;
- реалізація алгоритмів для аналізу активності обчислювальних ресурсів;
- тестування системи на вибраному наборі даних для оцінки її ефективності та точності.

У четвертому розділі проведено тестування розробленої інформаційної системи пошуку несанкціонованого майнінгу. Метою тестування було оцінити

ефективність роботи системи, її здатність виявляти аномальну активність, а також швидкість реагування на потенційні загрози. Тестування проводилося в різних умовах, включаючи симуляцію майнінгових процесів із високим навантаженням на ресурси.

Кваліфікаційна робота містить 73 сторінки, 29 рисунків, 25 джерел і 3 додатки.

Ключові слова: несанкціонований майнінг, криптовалюти, аналіз аномалій, інтелектуальні системи, кібербезпека, моніторинг ресурсів.

ABSTRACT

to the master's qualification work
by the student of the group 601 of Petro Mohyla Black Sea National University

Atamaniuk Vladyslav

on the subject: «**INFORMATION SYSTEM FOR SEARCHING FOR UNAUTHORIZED MINING**»

Qualification Thesis is devoted to the development and implementation of an intelligent information system for detecting unauthorized mining. In the context of the growing number of cyber threats and cases of illegal use of computing resources for cryptocurrency mining, the development of such systems is highly relevant. The work focuses on studying methods for analyzing resources, detecting anomalous activity, and developing software for monitoring computing systems.

The first chapter analyzes the subject area of intelligent systems for detecting unauthorized mining, including the current state of illegal mining worldwide, the specifics of cryptocurrencies and mining technologies, the main methods for detecting unauthorized mining, and existing systems and tools for identifying such threats.

The second chapter presents an analysis of the tools for developing software for an information system to detect unauthorized mining. It reviews the functional requirements for the system, selects methods for analyzing anomalous activity, and examines the operating principles and implementation schemes of popular solutions such as MinerBlock and Windows Defender.

The third chapter focuses on the software implementation of the intelligent system:

– system architecture design;

– implementation of algorithms for analyzing the activity of computing resources;

– testing the system on a selected dataset to evaluate its efficiency and accuracy.

The fourth chapter is dedicated to testing the developed information system for detecting unauthorized mining. The purpose of the testing was to evaluate the system's efficiency, its ability to detect anomalous activity, and the speed of its response to

potential threats. The testing was conducted under various conditions, including simulated mining processes with high resource loads.

Qualification Thesis contains 73 pages, 29 figures, 25 references, and 3 appendices.

Keywords: unauthorized mining, cryptocurrencies, anomaly detection, intelligent systems, cybersecurity, resource monitoring.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

ПК – персональний комп’ютер

ШІ – штучний інтелект

НМ – несанкціонований майнінг

ВСТУП

З розвитком технологій і збільшенням популярності криптовалют, проблема несанкціонованого використання обчислювальних ресурсів для майнінгу стала все більш актуальною. Несанкціонований майнінг – це використання комп'ютерних ресурсів без дозволу власників для видобутку криптовалют, що призводить до надмірного навантаження на систему, зниження її продуктивності та збільшення витрат на електроенергію. У зв'язку з цим, виникла потреба в розробці інтелектуальних систем для пошуку та запобігання несанкціонованому майнінгу.

Об’єктом кваліфікаційної роботи є бізнес-процеси кібербезпеки, зокрема захист комп'ютерних систем від несанкціонованого використання ресурсів для майнінгу.

Предметом дослідження є інструментальні засоби та інформаційні технології, що використовуються для розробки інтелектуальних систем виявлення несанкціонованого майнінгу.

Метою кваліфікаційної роботи є розробка та впровадження інтелектуальної системи автоматизованого пошуку несанкціонованого майнінгу в корпоративних та приватних мережах.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– аналіз сучасних загроз: дослідити природу несанкціонованого майнінгу, методи його впровадження та вплив на комп'ютерні системи. Огляд основних алгоритмів майнінгу, які можуть використовуватися без дозволу;

– аналіз існуючих систем кібербезпеки: вивчення сучасних методів та інструментів виявлення несанкціонованого використання ресурсів. Оцінка їх ефективності та можливих обмежень у застосуванні;

– специфікація вимог до програмного забезпечення: визначення основних функціональних і нефункціональних вимог до системи пошуку несанкціонованого майнінгу, включаючи виявлення аномальної активності, моніторинг використання ресурсів та автоматизовану генерацію попереджень;

– проєктування системи виявлення: розробка архітектури системи, яка базується на методах машинного навчання та аналізу великих даних для виявлення аномальної поведінки, характерної для несанкціонованого майнінгу;

– розробка прототипу інтелектуальної системи: реалізація основних функцій системи для виявлення та блокування несанкціонованого майнінгу на комп'ютерних системах. Тестування системи на ефективність в умовах реальних загроз.

Дослідження охоплює аналіз кіберзагроз, пов'язаних з майнінгом, специфікацію вимог до систем виявлення, проектування і створення прототипу інтелектуальної системи для автоматичного виявлення та запобігання несанкціонованому використанню комп'ютерних ресурсів.

У роботі використовувалися такі методи дослідження, як аналіз літературних джерел, вивчення статистичних даних щодо кіберзагроз, методи машинного навчання, а також експертні інтерв'ю в галузі кібербезпеки.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНИНГУ

Несанкціонований майнінг (cryptojacking) – це використання комп'ютерних ресурсів для видобутку криптовалют без відома або згоди власника системи. Ця загроза є важливою проблемою в галузі кібербезпеки, оскільки вона завдає шкоди організаціям і приватним користувачам шляхом прихованого використання процесорних потужностей, що може спричинити зниження продуктивності, перевантаження систем і збільшення витрат на електроенергію.

1.1 Ситуація з несанкціонованим майнінгом в світі на даний момент

Несанкціонований майнінг, або криптоджекінг, є однією з головних сучасних загроз кібербезпеки, яка продовжує поширюватися навіть після пікового зростання популярності криптовалют у 2017–2018 роках. Це явище суттєво впливає на різні аспекти як цифрової, так і реальної економіки.

Основна суть проблеми полягає в тому, що зловмисники використовують обчислювальні ресурси пристроїв інших людей чи компаній для майнінгу криптовалют без їхнього відома та згоди. Це може бути як окремий комп'ютер, так і ціла серверна інфраструктура, що працює на високих потужностях.

Криптоджекінг має кілька ключових негативних впливів на економіку.

Непередбачені витрати на енергоресурси та обладнання.

Кожен пристрій, який підпадає під атаку криптоджекерів, стає «майнінговою фермою» для зловмисників. Майнінг потребує значних обчислювальних ресурсів, що, у свою чергу, веде до суттєвого зростання енергоспоживання. Наприклад, на підприємствах, де використовуються численні сервери, зараження майнінг-скриптами може призвести до різкого збільшення витрат на електроенергію. Більш того, це негативно впливає на загальний ресурс пристроїв: через тривале навантаження вони швидше зношуються та потребують частішої заміни або обслуговування.

Особливо це відчутно для малих і середніх підприємств, для яких енергозатрати є значною частиною операційних витрат. В умовах, коли бізнес уже зіштовхується з підвищенням цін на енергію, такі приховані втрати можуть стати критичними.

В результаті компанії змушені інвестувати в дорогі системи моніторингу, кібербезпеки та захисту від криптоджекінгу.

Вплив на ринок криптовалют та економічну нестабільність.

Несанкціонований майнінг також впливає на ринок криптовалют. Що більше ресурсів зловмисники використовують для майнінгу, то більший тиск це чинить на загальні обчислювальні потужності, які задіяні в обробці блокчейнів криптовалют. Це може викликати непрямий вплив на коливання цін на криптовалюти, оскільки в залежності від обсягів майнінгу можуть змінюватися нагороди та складність розв'язання блоків.

Якщо великий обсяг потужностей використовується нелегально, це може створювати ілюзію збільшення кількості учасників майнінгу, що, у свою чергу, призводить до непередбачуваних змін на ринку. Такі коливання мають потенціал дестабілізувати ринок, оскільки інвестори та трейдери можуть реагувати на зміни у вартості криптовалют, спричинені аномаліями у майнінгу.

Виклики для енергетичної безпеки.

На глобальному рівні, неконтрольоване споживання електроенергії через криптоджекінг становить загрозу для енергетичної безпеки. Масштабне споживання електроенергії без належного планування та контролю може призвести до надмірного навантаження на локальні електромережі. Країни або регіони, які мають обмежені енергоресурси або зазнають постійних перебоїв у подачі електроенергії, можуть відчути додаткове навантаження, викликане нелегальним майнінгом.

Особливо це стосується держав, де ціна електроенергії є порівняно низькою або де енергомережі вже працюють на межі можливостей. Нелегальний майнінг

може підвищувати ймовірність відключень електроенергії, що ще більше погіршує стан економіки та інфраструктури.

Зниження продуктивності компаній і користувачів.

Зараження комп'ютерів скриптами для майнінгу призводить до зниження загальної продуктивності пристроїв. Уявіть ситуацію, коли комп'ютери компанії постійно працюють на підвищених навантаженнях, що уповільнює всі бізнес-процеси. Це може призвести до простоїв у роботі, повільнішої обробки даних, затримок у виконанні завдань та підвищених ризиків збоїв системи.

Для бізнесу це не лише фінансові втрати, але й репутаційні ризики. Наприклад, клієнти можуть залишитися незадоволеними через погіршення якості обслуговування або затримки в наданні послуг. Для приватних користувачів це також означає підвищену незручність і зниження продуктивності їхніх комп'ютерів.

Навантаження на кібербезпеку.

З огляду на те, що криптоджекінг часто є прихованим і складним для виявлення процесом, це створює додаткові виклики для команд кібербезпеки. Антивірусні програми не завжди можуть вчасно розпізнати несанкціоновану активність, оскільки майнінг може мати форму звичайної програми або скрипта. Тому фахівці змушені впроваджувати більш комплексні рішення для моніторингу активності мережі та процесів на пристроях.

Це призводить до збільшення витрат компаній на захист від кіберзагроз, а також потреби у залученні додаткових спеціалістів з кібербезпеки. Водночас розвиток систем виявлення несанкціонованого майнінгу створює нові можливості для інновацій у галузі безпеки, що може мати довгострокові позитивні наслідки.

Таким чином, ситуація з несанкціонованим майнінгом є складною та багатогранною проблемою, яка негативно впливає як на окремих користувачів, так і на цілі сектори економіки. Це явище вимагає комплексного підходу до вирішення,

включаючи розробку нових систем захисту, законодавчих ініціатив і покращення обізнаності користувачів щодо загроз, які вони можуть не помічати.

1.2 Криптовалюти та технології майнінгу

Майнінг криптовалют передбачає вирішення складних математичних задач для підтвердження транзакцій у блокчейні. Для цього використовуються потужні комп'ютерні системи, здатні обробляти величезну кількість даних. Найпоширенішими криптовалютами, для яких використовуються ці процеси, є Bitcoin, Ethereum та інші.

В процесі несанкціонованого майнінгу зловмисники використовують ресурси жертв для отримання прибутку, що може бути реалізовано через вірусне програмне забезпечення, веб-скрипти, зараження серверів або навіть мобільних пристроїв. Основні загрози, пов'язані з cryptojacking, полягають у:

– зниженні продуктивності системи: процесор і відеокарта можуть працювати на межі своїх можливостей, що уповільнює роботу комп'ютера;

– збільшенні енергоспоживання: несанкціонований майнінг спричиняє високе енергоспоживання, що збільшує витрати для користувача або компанії;

– зносі обладнання: постійне перевантаження апаратних ресурсів призводить до швидшого зношування обладнання.

1.3 Методи виявлення несанкціонованого майнінгу

У сфері виявлення несанкціонованого майнінгу виділяються кілька основних підходів:

– моніторинг використання ресурсів: один з найбільш базових способів виявлення несанкціонованого майнінгу – це відстеження аномального збільшення споживання CPU або GPU. Для цього використовуються спеціальні програми для моніторингу роботи процесора, пам'яті та відеокарти;

– аналіз мережевого трафіку: в процесі майнінгу комп'ютери постійно обмінюються даними з зовнішніми серверами, де йде обробка транзакцій. Нестандартна активність в мережі, наприклад, постійні підключення до майнінг-пулів, може бути ознакою cryptojacking;

– виявлення аномальної поведінки програм: використання методів машинного навчання та штучного інтелекту для виявлення нетипової поведінки програмного забезпечення, що може свідчити про запуск скриптів для майнінгу на системі;

– аналіз браузерів і веб-трафіку: одна з поширених форм cryptojacking – це майнінг через браузери, коли сайти вбудовують JavaScript-код для майнінгу в браузерах відвідувачів. Використання антивірусних програм, які можуть блокувати подібні скрипти, є ефективним методом протидії.

1.4 Існуючі інтелектуальні системи пошуку несанкціонованого майнингу

Існуючі системи виявлення несанкціонованого майнінгу (cryptojacking) ґрунтуються на різних методах аналізу поведінки систем і моніторингу обчислювальних ресурсів. Ці системи можуть бути як частиною антивірусного програмного забезпечення, так і спеціалізованими інструментами для моніторингу серверів та мереж. Ось деякі з найпоширеніших підходів і рішень:

А) антивірусні рішення: антивірусні програми зазвичай мають функції виявлення шкідливого програмного забезпечення, включаючи майнінг. Вони працюють на основі сигнатур шкідливого ПЗ або поведінкового аналізу:

- Windows Defender – вбудована антивірусна програма Windows, яка включає захист від криптоджекінгу;
- Malwarebytes – одна з найпопулярніших антивірусних програм, що може блокувати як шкідливе ПЗ, так і майнінгові скрипти в браузерах;
- Norton і Avast – ці антивіруси мають вбудовані функції для захисту від майнінгу, включаючи виявлення виконуваних файлів і браузерних скриптів;

Б) блокувальники браузерних майнерів: оскільки значна частина несанкціонованого майнінгу відбувається через браузери, деякі інструменти спеціально орієнтовані на блокування майнінгових скриптів у веб-браузерах:

- MinerBlock і No Coin – це розширення для браузерів, які блокують запити та скрипти, пов'язані з майнінгом. Вони працюють на основі чорних списків відомих майнінгових URL або шукають поведінкові патерни майнінгу;
- AdBlock і uBlock Origin – хоча ці інструменти здебільшого блокують рекламу, вони також можуть блокувати деякі майнінгові скрипти через свої бази даних;

В) системи виявлення вторгнень (IDS): ці системи контролюють мережевий трафік і шукають підозрілу активність, включаючи з'єднання з майнінговими пулами або передачу великих обсягів даних:

- Snort і Suricata – це IDS, які можна налаштувати для виявлення трафіку, пов'язаного з майнінгом. Вони аналізують мережеві пакети і виявляють з'єднання з відомими майнінговими сервісами;
- Zeek (раніше Bro) – система моніторингу мережі, яка може бути використана для виявлення аномальних з'єднань, таких як майнінгові пули;

Г) системи моніторингу ресурсів: багато серверних рішень та інструментів для корпоративних мереж використовують моніторинг ресурсів для виявлення аномального використання ЦП або графічного процесора, що може свідчити про майнінг:

- Nagios і Zabbix – системи моніторингу, які можуть бути налаштовані для відстеження використання ресурсів і генерації сповіщень у разі виявлення аномальних навантажень;
- Datadog – цей сервіс моніторингу також може бути використаний для аналізу використання ресурсів і виявлення несанкціонованого майнінгу на основі шаблонів поведінки;

Д) інструменти для аналізу контейнерів і віртуальних середовищ: з огляду на популярність використання контейнерних середовищ (Docker, Kubernetes), спеціалізовані інструменти для моніторингу контейнерів відіграють важливу роль у захисті від майнінгу:

- ClamAV – антивірус, який може працювати в середовищах Unix-подібних операційних систем і контейнерів для пошуку шкідливого ПЗ, включаючи майнери;
- Sysdig – інструмент для моніторингу контейнерів і хмарних середовищ, що дозволяє виявляти аномальну поведінку, таку як майнінг;

Е) рішення на основі машинного навчання: новіші системи захисту починають використовувати алгоритми машинного навчання для аналізу поведінки програмного забезпечення та виявлення аномалій:

- Darktrace – це рішення використовує штучний інтелект для виявлення аномальної мережевої активності, включаючи несанкціонований майнінг;
- Cylance – антивірус, що використовує машинне навчання для аналізу підозрілої поведінки процесів і блокування майнінгових атак.

Як підсумок, можна сказати, що існуючі рішення для виявлення несанкціонованого майнінгу охоплюють широкий спектр методів — від класичних антивірусів і блокувальників браузерних скриптів до систем машинного навчання і інструментів для аналізу контейнерів. Вибір оптимального рішення залежить від типу системи, яку потрібно захистити, та специфіки використання ресурсів.

Існуючі інструменти захисту від несанкціонованого майнінгу

Назва інструменту	Браузер	Windows ОС	Unix-подібні ОС	Контейнери серверних ОС
Miner Block	+	–	–	–
No Coin	+	–	–	–
Malwarebytes	+	–	–	–
Windows Defender	–	+	–	–
Norton	–	+	+	–
Avast	–	+	+	–
Clam AV	–	+	+	+
Comodo	+	+	+	–

Рисунок 1.1 – Існуючі інструменти пошуку майнінгу

На рисунку було наведено ряд існуючих інтелектуальних систем пошуку несанкціонованого майнінгу, тепер більш детально зупинимось на кожному з них:

А) **MinerBlock**:

– це розширення для браузера, яке блокує несанкціоновані майнінгові скрипти. Воно діє шляхом блокування запитів на відомі майнінгові сайти або шляхом виявлення поведінки майнінгових скриптів у браузері;

– працює в браузерах, але не підтримує інші типи операційних систем, такі як Windows чи Unix;

Б) **No Coin**:

– подібно до MinerBlock, це ще одне розширення для браузера, яке запобігає використанню потужностей процесора користувача для майнінгу криптовалют без його відома. Воно створено для захисту під час серфінгу в інтернеті;

– ефективне лише для браузерів, але не захищає операційні системи Windows або Unix;

В) **Malwarebytes**:

– це антивірусна програма, яка також забезпечує захист від шкідливих майнінгових програм. Вона здатна виявляти майнери, які працюють у фоновому режимі, і блокує їх;

– підтримується на Windows, але не підтримує Unix-подібні системи та контейнери серверних операційних систем;

Г) **Windows Defender**:

– вбудований антивірус для операційних систем Windows, який може виявляти шкідливе програмне забезпечення, включно з майнінговими програмами. Він активно перевіряє активність системи на наявність підозрілих процесів і трафіку;

– підтримується лише на Windows;

Д) **Norton**:

– один із найвідоміших антивірусів, що забезпечує всебічний захист комп’ютера, включно з блокуванням майнінгових загроз. Програма може зупиняти підозрілу активність, яка характерна для криптоджекінгу;

– підтримує лише Windows;

Е) **Avast**:

– антивірус, який також пропонує захист від майнінгу криптовалют. Він виявляє процеси, які споживають ресурси для майнінгу, та зупиняє їх;

– працює на Windows і підтримує деякі Unix-системи;

Ж) **Clam AV**:

– це антивірусне рішення з відкритим вихідним кодом, яке переважно використовується для сканування файлів на наявність вірусів. Може бути налаштоване на пошук шкідливих майнінгових програм, однак не завжди виявляє їх;

– підтримує Unix-подібні операційні системи;

3) **Comodo**:

– антивірусне програмне забезпечення, яке забезпечує захист від різних типів шкідливих програм, включаючи майнінгові скрипти;

– підтримується як на Windows, так і на Unix-подібних операційних системах.

Кожен з цих інструментів має свої сильні та слабкі сторони, залежно від платформи і функціональності.

Висновки до розділу 1

Несанкціонований майнінг, або cryptojacking, являє собою серйозну загрозу в галузі кібербезпеки. Його негативний вплив проявляється в декількох аспектах, зокрема, це збільшене енергоспоживання, зниження продуктивності пристроїв та передчасний знос обладнання через надмірне навантаження на ресурси. Зловмисники використовують обчислювальні потужності пристроїв без відома користувачів, що призводить до прихованих витрат для бізнесу і приватних осіб. Це особливо небезпечно для організацій, які змушені інвестувати значні ресурси в захист від cryptojacking, а також впроваджувати нові системи моніторингу та безпеки. Крім того, неконтрольоване споживання електроенергії може створювати додаткове навантаження на інфраструктуру, а також впливати на ринок криптовалют, що призводить до економічної нестабільності.

РОЗДІЛ 2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСТОСУНКУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНИНГУ

2.1 Функціональні вимоги до програмного забезпечення інтелектуальної системи пошуку несанкціонованого майнингу

Виявлення процесів майнінгу.

Для забезпечення захисту від несанкціонованого майнінгу система повинна виконувати автоматизований пошук процесів майнінгу на всіх активних пристроях у мережі. Важливо, щоб цей процес не вимагав значних втручань від адміністратора, працював у фоновому режимі та не впливав на продуктивність системи. Автоматизація дає змогу постійно моніторити активність, і, у разі виявлення підозрілої діяльності, система може швидко реагувати.

Для цього необхідно застосовувати алгоритми аналізу поведінки процесів. Сучасні методи аналізу включають використання машинного навчання для відстеження шаблонів поведінки, що властиві майнінговим процесам: інтенсивне використання процесорних потужностей, незвично велика активність у мережі тощо. Це допоможе виявляти навіть ті процеси, які не мають стандартних майнінгових сигнатур і можуть бути модифіковані для уникнення детекції.

ПЗ повинно мати можливість порівняння поточних процесів із відомими сигнатурами майнінгового ПЗ. Сигнатури – це унікальні ознаки програмного забезпечення, які дозволяють виявляти наявність певних процесів. Постійне оновлення бази даних сигнатур є необхідним для ефективного виявлення майнінгу, адже нові версії програм з'являються регулярно, і система повинна бути здатною реагувати на нові загрози.

Для цього потрібно підтримувати автоматизовану базу даних, яка регулярно отримує оновлення із зовнішніх джерел або від постачальників кібербезпеки.

Аналіз мережевої активності.

Аналіз мережевої активності є важливим компонентом у виявленні несанкціонованого майнінгу, оскільки майнінгові програми часто потребують з'єднання з віддаленими серверами або майнінговими пулами. Система повинна постійно моніторити всі мережеві з'єднання для виявлення підозрілих зв'язків з відомими пулами або серверами, що використовуються для криптомайнінгу.

Окрім цього, виявлення аномальної активності допоможе ідентифікувати нелегальну діяльність. Наприклад, аномально високий рівень використання інтернет-трафіку або процесорних ресурсів без обґрунтованих на це причин може свідчити про приховану активність майнінгу. Також нетипові мережеві запити, спрямовані на специфічні IP-адреси, що належать майнінговим пулам, можуть бути індикатором майнінгової активності.

Важливо враховувати і шифровані з'єднання, які також слід аналізувати на предмет підозрілої активності.

Автоматичне блокування

Один із ключових елементів системи – це здатність автоматично блокувати виявлені процеси майнінгу. Це дозволяє негайно реагувати на загрозу без необхідності втручання людини. Система повинна вміти розпізнавати процеси, що використовують значні обчислювальні ресурси для майнінгу, і зупиняти їх. Зупинка таких процесів дозволить знизити навантаження на ресурси і уникнути економічних втрат, пов'язаних із незаконним використанням ресурсів компанії або користувача.

Окрім цього, система повинна мати можливість ізолювати контейнер або віртуальну машину, де було виявлено майнінгову активність.

Це важливо в умовах віртуалізованих середовищ або хмарних інфраструктур, де ізоляція одного компонента може запобігти поширенню загрози на інші частини системи.

Оповіщення користувачів.

Ефективна система повинна не тільки виявляти та блокувати процеси, але й своєчасно інформувати відповідальних осіб. ПЗ повинно мати функцію оповіщення адміністратора або іншого відповідального персоналу про випадки виявлення нелегального майнінгу. Повідомлення можуть надсилатися через електронну пошту, SMS або інтегруватися з системами моніторингу мережі, що використовуються в організації.

Система повинна фіксувати всі події в журналах для подальшого аналізу і надання звітів.

Це дозволить адміністраторам детально розібратися у ситуаціях і, при необхідності, вжити додаткових заходів для підвищення рівня безпеки.

Управління ресурсами.

Одним із важливих аспектів будь-якої системи кібербезпеки є її вплив на продуктивність системи. Система пошуку несанкціонованого майнінгу повинна мінімізувати використання ресурсів комп'ютера або сервера, щоб не заважати виконанню основних задач.

Це особливо актуально для середовищ з високим навантаженням, таких як сервери або робочі станції в корпоративних мережах.

ПЗ повинно мати налаштування рівня моніторингу і частоти перевірок для оптимізації використання ресурсів. Наприклад, можна налаштувати менш інтенсивний моніторинг на системах з низьким рівнем ризику і більш агресивний на критичних інфраструктурах. Це дозволить оптимізувати захист без надмірного навантаження на ресурси.

Таким чином, система пошуку несанкціонованого майнінгу повинна не тільки виявляти і блокувати загрози, але й робити це ефективно і без шкоди для продуктивності системи.

2.2 Один із методів пошуку несанкціонованого майнінгу на серверних ОС

Для ефективного виявлення несанкціонованого майнінгу криптовалют у контейнерах серверних ОС було запропоновано метод, заснований на трьох основних критеріях: аналіз назви процесу, перевірка з'єднання з майнінговими пулами та перевірка бінарних сигнатур. Застосування кожного з цих параметрів дозволяє суттєво підвищити точність виявлення загрози, оскільки окремий підхід може не охопити всі можливі варіанти шкідливої активності.

Основні етапи методу:

– пошук за назвою процесу.
Це базовий етап, що полягає у скануванні запущених процесів для виявлення назв, пов'язаних з майнінгом, таких як XMRig або Cryptoloot. Однак цей підхід не завжди ефективний, оскільки зловмисники часто маскують шкідливі процеси під відомі системні або користувацькі програми, наприклад, під chrome.exe або skype.exe;

– пошук за з'єднанням з майнінговими пулами.
Майнінг потребує постійного з'єднання з майнінговим пулом, який об'єднує ресурси багатьох користувачів для ефективнішого видобування криптовалюти. Система аналізує активні мережеві з'єднання для виявлення зв'язків з відомими пулами, такими як suprnova.cc, nanopool.org та інші. Це дозволяє виявляти майнінг навіть тоді, коли назва процесу замаскована;

– пошук за бінарними сигнатурами.
Бінарні сигнатури файлів використовуються для однозначної ідентифікації програмного забезпечення. Порівнюючи бінарні сигнатури запущених процесів з базою даних відомих майнінгових програм, можна точно виявити шкідливу активність. Важливо, що ця база даних постійно оновлюється, оскільки нові загрози виникають регулярно.

Блок-схема методу:

– початок роботи;

– завантаження або оновлення баз даних з актуальною інформацією про майнінгові пули, назви процесів та бінарні сигнатури;

– пошук підозрілих процесів за назвою;

– пошук за бінарною сигнатурою;

– пошук за наявністю з'єднання з відомими майнінговими пулами;

– якщо будь-який з методів виявляє процес майнінгу, відбувається зупинка відповідного контейнера;

– оповіщення адміністратора про виявлену загрозу;

– завершення роботи.

Рисунок 2.1 – Блок-схема роботи методу

2.3 MinerBlock. Принцип роботи та схема

MinerBlock — це розширення для браузерів, яке блокує майнінгові скрипти, запобігаючи використанню процесорних потужностей користувача для несанкціонованого майнінгу криптовалют (cryptojacking). Це розширення працює за принципом блокування шкідливих запитів до відомих майнінгових сайтів або виявлення й блокування скриптів безпосередньо на веб-сторінках, що виконують майнінг.

Основні етапи роботи MinerBlock:

– аналіз HTTP-запитів: MinerBlock перехоплює HTTP-запити, які надсилає браузер під час відвідування веб-сайту. Він порівнює ці запити з базою даних відомих майнінгових сайтів, таких як Coinhive та інші;

– виявлення майнінгових скриптів: окрім перехоплення запитів, MinerBlock також аналізує вміст сторінки, шукаючи вбудовані майнінгові скрипти JavaScript. Якщо на сторінці є такі скрипти, розширення блокує їх виконання;

– блокування з'єднань: у разі виявлення спроби встановлення з'єднання з майнінговим пулом (віддаленим сервером, на який спрямовуються обчислення для майнінгу), MinerBlock блокує цей запит, тим самим зупиняючи майнінг;

– блокування вбудованих скриптів: якщо майнінг здійснюється через шкідливий скрипт на веб-сайті, який не викликає зовнішніх з'єднань, MinerBlock аналізує сам вміст сторінки та блокує виконання цього скрипта;

– оновлення бази даних: MinerBlock використовує список відомих майнінгових пулів і шкідливих скриптів, який постійно оновлюється. Це дозволяє ефективно блокувати нові загрози.

Схема роботи MinerBlock:

А) початок:

– користувач відкриває браузер і завантажує веб-сторінку;

Б) аналіз веб-запитів:

– MinerBlock перехоплює всі HTTP-запити, які надсилаються від браузера до веб-сайтів;

В) перевірка на майнінгові пули:

– кожен запит перевіряється на наявність зв'язку з відомими майнінговими серверами (згідно з базою даних MinerBlock);

Г) виявлення підозрілого з'єднання або скрипта:

– якщо виявлено з'єднання з майнінговим пулом або знайдено майнінговий скрипт на сторінці, система блокує цей запит;

Д) блокування:

– MinerBlock припиняє виконання майнінгових скриптів і блокує з'єднання з пулом;

Е) продовження роботи:

– якщо загрози не виявлено, користувач продовжує перегляд сторінки без переривань;

Ж) оновлення:

– база даних розширення автоматично оновлюється для забезпечення захисту від нових загроз.

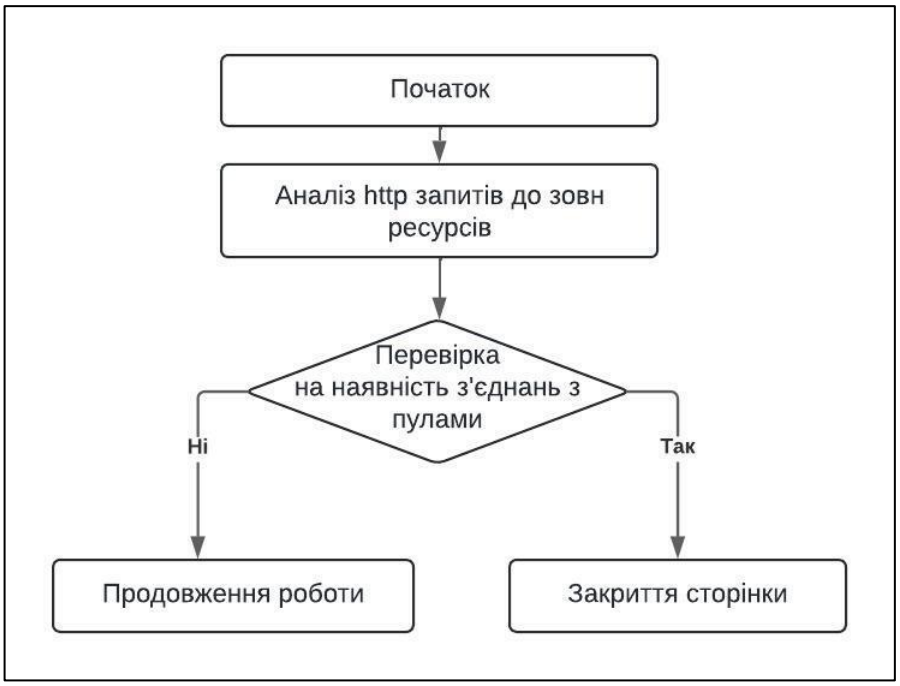

Рисунок 2.2 – Блок-схема роботи MinerBlock

Пояснення до схеми:

- користувач відкриває сторінку, після чого MinerBlock починає перевірку;
- розширення аналізує всі запити до зовнішніх ресурсів і вміст сторінки;
- проводиться порівняння з відомими майнінговими сайтами або пулом;
- якщо скрипти або з'єднання виявлено, сторінка закривається;
- якщо загрозу не виявлено, сторінка працює як звичайно.

MinerBlock забезпечує надійний захист від криптоджекінгу, блокуючи як зовнішні з'єднання з майнінговими пулами, так і вбудовані шкідливі скрипти на веб-сторінках.

2.4 Windows Defender. Принцип роботи та схема

Windows Defender — це вбудована антивірусна програма в операційній системі Windows, яка забезпечує захист комп'ютера від різних типів загроз, включаючи віруси, шкідливе програмне забезпечення та несанкціонований майнінг криптовалют (cryptojacking). Windows Defender працює в режимі реального часу, перевіряючи файли, процеси та мережеві з'єднання для виявлення загроз.

Основні етапи роботи Windows Defender для виявлення несанкціонованого майнінгу:

– аналіз системних процесів: Windows Defender перевіряє всі активні процеси на комп'ютері, порівнюючи їх з базою відомих шкідливих програм. Це включає аналіз процесів, які можуть використовувати ресурси комп'ютера для майнінгу криптовалют;

– виявлення шкідливих файлів: програма регулярно перевіряє всі файли на комп'ютері. Якщо виявляється файл або програма, яка відповідає майнінговій сигнатурі, вона блокується і видаляється;

– моніторинг мережевих з'єднань: Windows Defender також аналізує активні мережеві з'єднання для виявлення підозрілих зв'язків з майнінговими пулами. У разі виявлення підозрілих з'єднань програма блокує їх і попереджає користувача;

– захист у реальному часі: Windows Defender працює у фоновому режимі і в режимі реального часу аналізує вхідні та вихідні запити, що дозволяє миттєво реагувати на загрози;

– оновлення бази даних загроз: Windows Defender регулярно оновлює свою базу даних шкідливих програм і сигнатур, що дозволяє ефективно виявляти нові загрози, зокрема нові майнінгові програми;

– сповіщення користувача: у разі виявлення загрози Windows Defender автоматично блокує шкідливий процес або файл і сповіщає користувача через систему сповіщень Windows.

Схема роботи Windows Defender:

А) початок:

– Windows Defender працює у фоновому режимі, перевіряючи активність комп'ютера;

Б) аналіз системних процесів:

– програма перевіряє всі запущені процеси і порівнює їх з базою відомих шкідливих програм;

В) виявлення шкідливого процесу:

– якщо виявлено процес, що відповідає майнінговій програмі, програма блокує його;

Г) аналіз файлів на наявність шкідливого ПЗ:

– програма аналізує файли системи на відповідність відомим сигнатурам майнінгових програм;

Д) моніторинг мережевих з'єднань:

– програма перевіряє мережеві з'єднання на наявність зв'язків з майнінговими пулами;

Е) оновлення бази даних загроз:

– Windows Defender регулярно оновлює базу даних загроз, забезпечуючи захист від нових майнінгових програм;

Ж) сповіщення користувача:

– програма сповіщає користувача у разі виявлення і блокування шкідливого ПЗ.

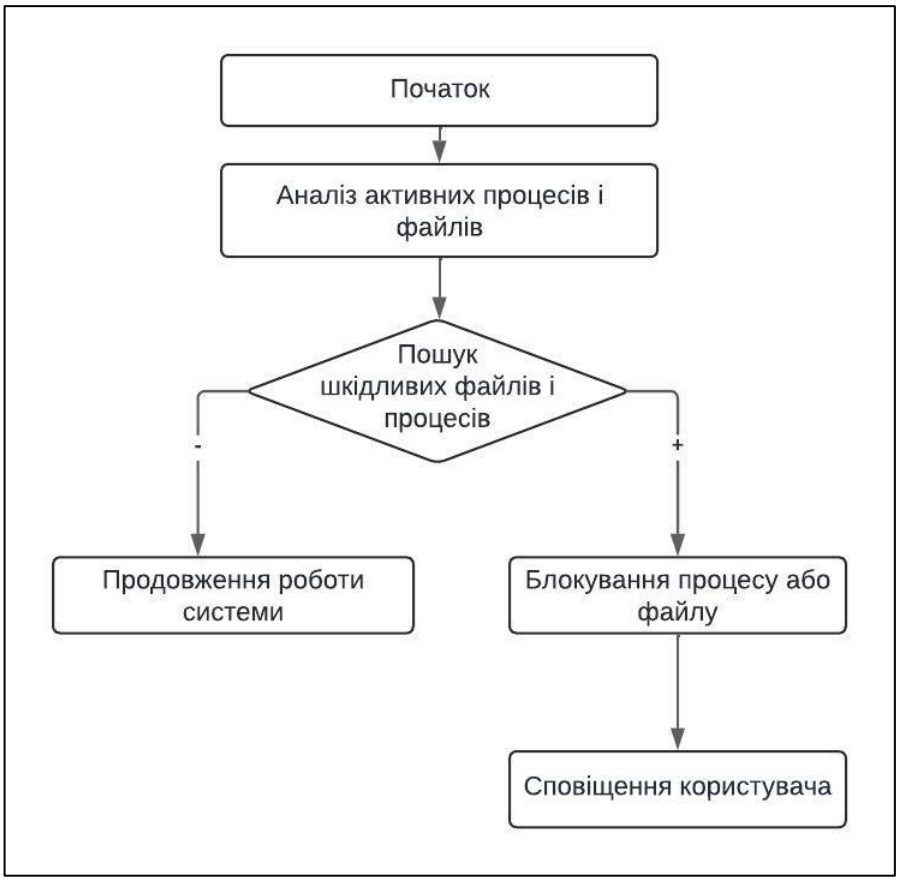

Рисунок 2.3 – Блок-схема роботи Windows Defender

Пояснення до схеми:

– аналіз активності: Windows Defender безперервно перевіряє процеси, файли та мережеві з'єднання на наявність підозрілих елементів;

– порівняння з базою даних: усі елементи, які перевіряються, порівнюються з базою відомих загроз, включаючи майнінгові програми;

– дії при виявленні: якщо виявлено шкідливий файл або процес, Windows Defender блокує його і сповіщає користувача;

– продовження роботи: якщо загрозу не виявлено, комп'ютер продовжує працювати без змін.

Windows Defender забезпечує багаторівневий захист системи, аналізуючи процеси, файли та мережеву активність у реальному часі. Завдяки регулярним оновленням бази даних шкідливих програм він здатний ефективно захищати від нових загроз, включаючи несанкціонований майнінг.

2.5 No Coin. Принцип роботи та схема

No Coin — це популярне розширення для веббраузерів, яке спеціально розроблене для блокування криптоджекінгу. Його основна функція — запобігти несанкціонованому використанню ресурсів комп'ютера вебсайтами для майнінгу криптовалют. Це простий і ефективний інструмент, що захищає користувача під час перегляду вебсайтів, на яких можуть бути приховані скрипти для майнінгу.

Принцип роботи No Coin:

– блокування криптомайнінг-скриптів: No Coin аналізує вебсайти на наявність криптоджекінг-скриптів, таких як Coinhive та інші подібні сервіси. Як тільки користувач заходить на сайт, який намагається виконати майнінговий скрипт, No Coin блокує виконання цих скриптів, не даючи їм можливості використовувати ресурси комп'ютера для майнінгу криптовалют;

– чорний список вебсайтів: No Coin має базу даних вебсайтів, які відомі використанням майнінг-скриптів. Коли користувач відвідує один із таких сайтів, розширення автоматично блокує спробу майнінгу. База даних постійно оновлюється, щоб включати нові загрози;

– оптимізація роботи браузера: оскільки майнінг споживає велику кількість ресурсів процесора, No Coin не лише захищає комп'ютер, але й оптимізує його роботу, запобігаючи збільшенню навантаження на процесор, що сприяє швидшій роботі браузера і комп'ютера загалом;

– ручне управління: No Coin дозволяє користувачам вручну вмикати або вимикати блокування майнінгу на певних сайтах. Наприклад, якщо користувач довіряє певному сайту, він може дозволити майнінг на цьому ресурсі, що є корисною функцією для налаштування розширення під конкретні потреби.

Нижче наведено блок-схему роботи No Coin.

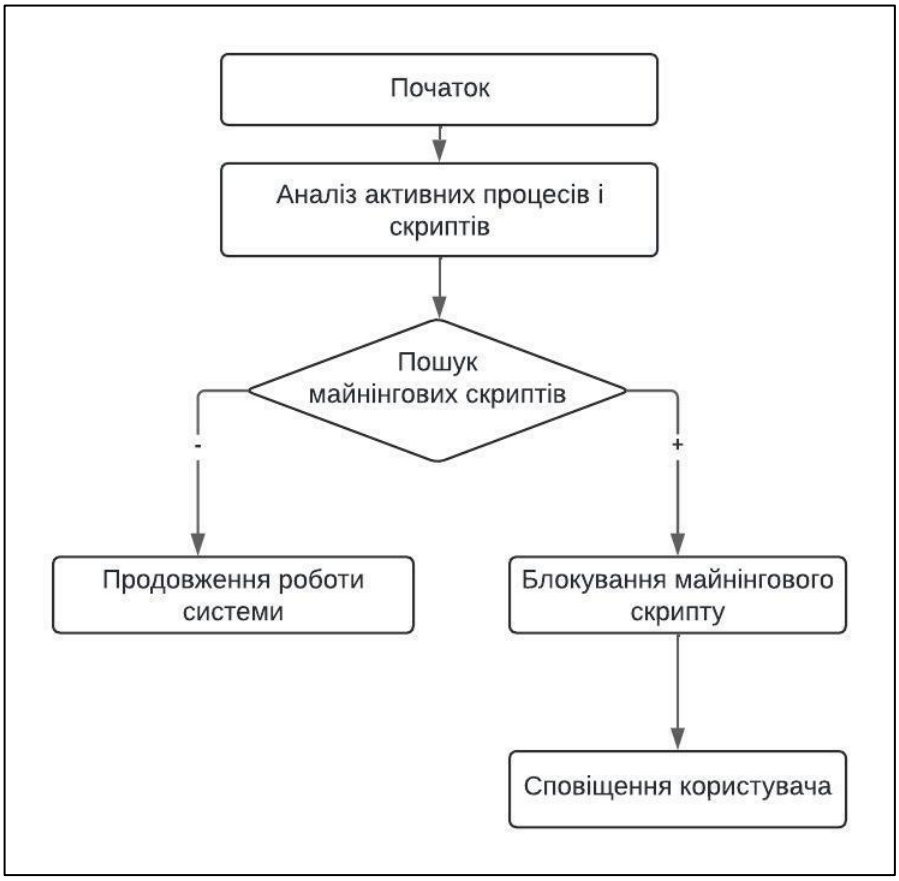

Рисунок 2.4 – Блок-схема роботи No Coin

No Coin є ефективним та простим інструментом для захисту від криптоджекінгу під час роботи в інтернеті. Весь процес відбувається в режимі реального часу, що дозволяє користувачам уникати неприємностей, пов'язаних із використанням їх ресурсів для несанкціонованого майнінгу.

Висновки до розділу 2

У другому розділі роботи було проведено аналіз засобів розробки програмного забезпечення для інтелектуальної системи пошуку несанкціонованого майнінгу. Було сформульовано функціональні вимоги до системи, яка повинна забезпечувати автоматизоване виявлення процесів майнінгу, моніторинг мережевої активності, автоматичне блокування загроз та своєчасне оповіщення користувачів. Особливу увагу було приділено мінімізації навантаження на системні ресурси під час виконання завдань з виявлення та блокування загроз.

Розглянуто сучасні методи пошуку несанкціонованого майнінгу, серед яких було виділено аналіз процесів, перевірка з'єднань з майнінговими пулами та перевірка бінарних сигнатур. Застосування цих методів дозволяє підвищити точність виявлення загроз і зменшити ризик прихованої активності.

Окрім цього, розглядалися такі популярні інструменти захисту, як **Windows Defender** та **No Coin**. Було проаналізовано їхні можливості у виявленні та блокуванні несанкціонованого майнінгу. **Windows Defender** пропонує комплексний захист на рівні операційної системи, включаючи моніторинг процесів і мережевих з'єднань. **No Coin**, своєю чергою, є ефективним розширенням для браузерів, яке блокує майнінгові скрипти під час вебсерфінгу.

Таким чином, проведений аналіз засобів розробки та існуючих інструментів показав необхідність впровадження багаторівневого підходу до захисту від криптоджекінгу. Це дозволяє забезпечити надійний захист як на рівні окремих пристроїв, так і мережевих інфраструктур.

РОЗДІЛ 3 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНІНГУ

3.1 Вибір мов програмування та середовищ розробки

На етапі вибору мов програмування для розробки інтелектуальної системи пошуку несанкціонованого майнінгу важливо враховувати як технічні вимоги до програмного забезпечення, так і характеристики кожної мови. До ключових критеріїв належать продуктивність, гнучкість, доступність інструментів для роботи з даними, підтримка бібліотек для машинного навчання, сумісність із різними операційними системами, а також можливості інтеграції з апаратним забезпеченням.

Одним із найпоширеніших варіантів є використання мови Python, яка має широкий спектр бібліотек для аналізу даних і машинного навчання, таких як TensorFlow, Scikit-learn і Pandas. Її простота у використанні та читабельність коду значно зменшують час розробки та тестування програмного забезпечення. Python також відзначається високим рівнем спільноти підтримки, що є додатковою перевагою при вирішенні складних завдань, таких як виявлення аномальної активності, характерної для несанкціонованого майнінгу.

Рисунок 3.1 – Логотип Python

Альтернативно, для задач, які потребують високої продуктивності, розглядається використання мови програмування C++. Завдяки своїй близькості до апаратного рівня ця мова забезпечує швидке виконання складних обчислень і точний контроль над системними ресурсами. Такий підхід може бути корисним для розробки модулів, які здійснюють моніторинг активності процесора та графічного адаптера у режимі реального часу.

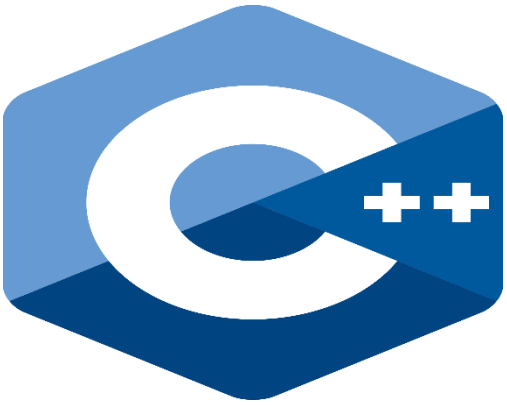

Рисунок 3.2 – Логотип C++

Мова Java також заслуговує уваги через її платформну незалежність та надійність. Вона добре підходить для розробки багатопотокових систем із масштабованою архітектурою. Java дозволяє створювати портативні програми, які можуть ефективно працювати на різних операційних системах без необхідності значних змін у коді.

Рисунок 3.3 – Логотип Java

Вибір середовища розробки (IDE) залежить від обраної мови програмування. Наприклад, для Python оптимальними варіантами є PyCharm або VS Code, які забезпечують інтегровані інструменти для відлагодження, аналізу продуктивності та роботи з бібліотеками. Для C++ ефективним буде використання Visual Studio, яка пропонує зручний інструментарій для роботи з великими проектами. У випадку Java можна використовувати IntelliJ IDEA, що має потужні засоби автоматизації завдань розробника та інтеграції з системами контролю версій.

Рисунок 3.4 – Логотип VS Code

3.2 Розробка архітектури системи

Розробка архітектури інтелектуальної системи пошуку несанкціонованого майнінгу є ключовим етапом, що визначає загальну структуру програми, взаємодію її компонентів, а також забезпечує її функціональність, масштабованість та ефективність. Архітектура такої системи повинна базуватися на модульному підході, що дозволяє легко вдосконалювати окремі частини програми без необхідності змін у всій системі.

Основу архітектури становлять три основні модулі: **модуль збору даних**, **модуль аналізу** та **модуль реакції**.

Модуль збору даних відповідає за отримання інформації про роботу системи, на якій запущена програма. Він забезпечує моніторинг параметрів, таких як використання центрального процесора (CPU), графічного процесора (GPU), оперативної пам’яті та мережевої активності. Для цього передбачається

використання системних API, наприклад, Windows Management Instrumentation (WMI) для Windows або /proc-файлів для Linux. Крім того, цей модуль може інтегруватися з існуючими антивірусними рішеннями для отримання додаткових даних про підозрілі процеси.

Модуль аналізу є центральною частиною системи. Його задача полягає у виявленні аномальної активності, яка може свідчити про наявність несанкціонованого майнінгу. Для цього використовуються методи машинного навчання, такі як класифікація та кластеризація, а також алгоритми на основі правил, що ґрунтуються на аналізі характерних ознак процесів майнінгу. Наприклад, високий рівень використання GPU у поєднанні зі стабільно високим навантаженням на CPU може бути індикатором підозрілої активності. Даний модуль отримує дані від модуля збору, аналізує їх і передає результати до модуля реакції.

Модуль реакції реалізує стратегії реагування на виявлену загрозу. У випадку ідентифікації підозрілого процесу він може автоматично припиняти його виконання, надсилати повідомлення адміністратору, створювати лог-файли або активувати додаткові заходи безпеки, такі як ізоляція підозрілого процесу. Крім того, модуль генерує звіти для подальшого аналізу результатів роботи системи.

Для забезпечення зручності використання передбачається реалізація інтерфейсу користувача. Інтерфейс може бути реалізований у вигляді веб-панелі, яка дозволяє переглядати статистику роботи системи, отримувати сповіщення та керувати налаштуваннями. Комунікація між інтерфейсом і основними модулями системи може здійснюватися через REST API.

З точки зору загальної архітектури, система працює за принципом клієнт-серверної моделі. Серверна частина виконує основні обчислення та аналіз, тоді як клієнтська сторона забезпечує взаємодію користувача із системою. Такий підхід дозволяє розподілити навантаження між компонентами системи, підвищуючи її ефективність та масштабованість.

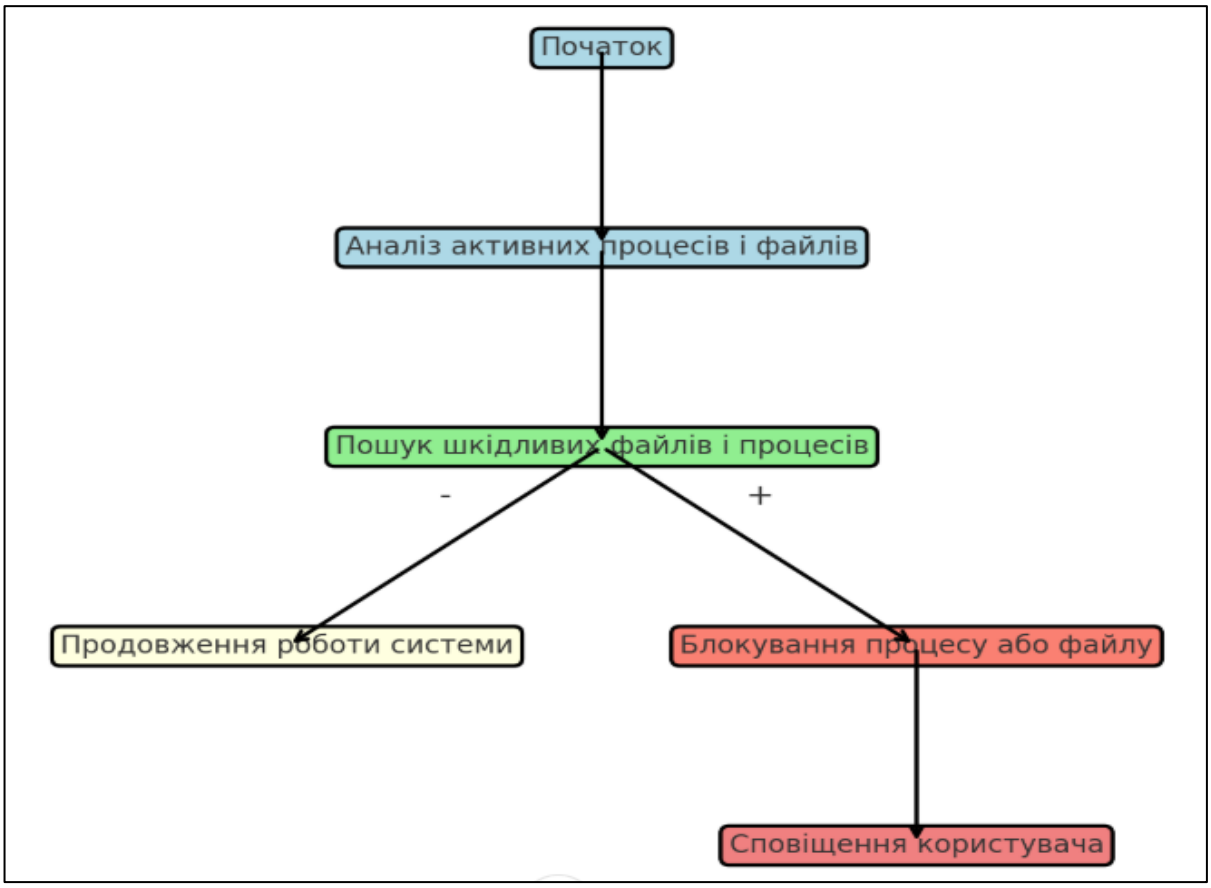

Рисунок 3.5 – Блок-схема роботи інформаційцної системи несанкціонованого майнінгу

3.3 Реалізація основних модулів системи

Розроблена система пошуку несанкціонованого майнінгу складається з кількох основних модулів, які забезпечують моніторинг завантаження процесора та GPU, виявлення підозрілих процесів і їх автоматичне завершення. Система має графічний інтерфейс для взаємодії з користувачем та логування подій. Розглянемо реалізацію кожного модуля.

Модуль моніторингу процесів.

Цей модуль відповідає за збір інформації про всі активні процеси в системі, включаючи їх ідентифікатор (PID), назву та поточне завантаження CPU. Основою модуля є використання бібліотеки psutil, яка дозволяє отримувати системну інформацію в реальному часі.

Основна логіка моніторингу процесів:

А) аналіз процесів:

– використовується метод psutil.process_iter, що ітерує через усі активні процеси;

– для кожного процесу збираються дані про pid, name (назву) і cpu_percent (завантаження CPU);

– процеси з PID 0 (системні процеси) та їх аналоги (наприклад, System Idle Process) пропускаються;

Б) виявлення підозрілих процесів:

– процеси, які використовують більше 60% CPU (заданий поріг), вважаються підозрілими;

– використовується перевірка:

```
if cpu_usage > CPU_THRESHOLD:
```

Рисунок 3.6 – Перевірка на завантаженість процессора

В) автоматичне завершення підозрілих процесів:

– метод proc.terminate() завершує процеси, які відповідають умовам;

– інформація про завершені процеси виводиться у лог і відображається у вигляді спливаючого вікна.

```
if cpu_usage > CPU_THRESHOLD:
    log_output.append(f"[ВНИМАНИЕ] Процесс с высокой загрузкой CPU: {name} (PID: {pid}), CPU: {cpu_usage}%")
    proc.terminate()
    log_output.append(f"[УСПЕХ] Процесс {name} (PID: {pid}) успешно завершён.")
```

Рисунок 3.7 – Завершення підозрілого процесу

Модуль аналізу GPU.

Модуль аналізу GPU реалізовано з використанням бібліотеки pynvml (NVIDIA Management Library), яка дозволяє отримати інформацію про завантаження пам’яті графічного процесора.

Основна логіка:

А) ініціалізація бібліотеки:

– при запуску програми здійснюється спроба ініціалізації nvmlInit;

– якщо GPU недоступний, програма продовжує роботу без аналізу GPU;

Б) збір інформації про GPU:

– використовується метод nvmlDeviceGetMemoryInfo для отримання інформації про загальну та використану пам’ять GPU;

```
return f"{gpu_name}: {gpu_usage:.2f}% использование памяти"
```

Рисунок 3.8 – Результат перевірки завантаженості GPU

В) інтеграція з логами:

– статус GPU виводиться у текстове поле інтерфейсу.

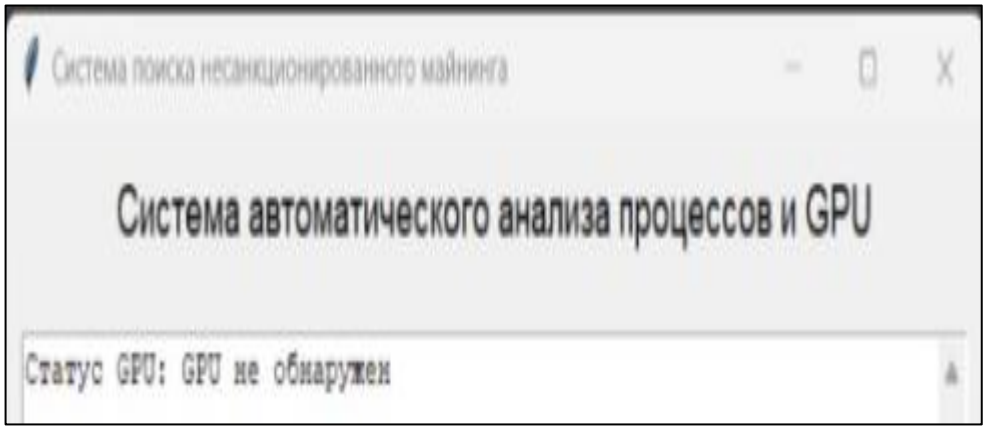

Рисунок 3.9 – Статус GPU

Модуль інтерфейсу користувача.

Графічний інтерфейс реалізовано за допомогою бібліотеки tkinter. Він забезпечує відображення логів, автоматичне оновлення та спливаючі повідомлення про завершені процеси.

Основні компоненти:

А) текстове поле логів:

– логи записуються у віджет Text, де користувач може переглянути результати аналізу;

– віджет підтримує вертикальну прокрутку через Scrollbar;

Б) кнопка запуску аналізу:

– аналіз виконується автоматично через функцію schedule_analysis, яка повторюється кожні 3 секунди;

```
root.after(3000, schedule_analysis)
```

Рисунок 3.10 – Код для автоматичного оновлення

В) спливаючі повідомлення:

– для кожного завершеного процесу викликається метод messagebox.showinfo, який показує користувачу ім’я та PID завершеного процесу;

```
messagebox.showinfo("Процесс завершён", f"Процесс {name} (PID: {pid}) завершён.")
```

Рисунок 3.11 – Повідомлення про завершення процесу

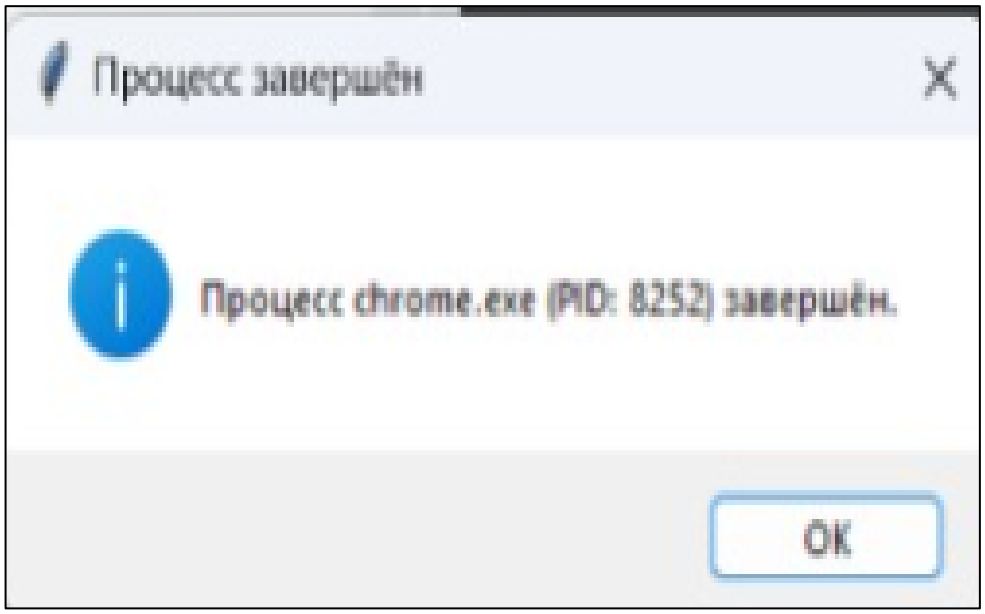

Рисунок 3.12 – Графічний вигляд повідомлення

Модуль регулярного аналізу

Цей модуль відповідає за періодичне виконання перевірки системи:

– запуск функції schedule_analysis через root.after дозволяє програмі автоматично викликати аналіз без потреби у взаємодії користувача;

– функція викликає run_analysis, яка:

– виконує аналіз GPU;

– викликає функцію terminate_miner_processes для виявлення та завершення підозрілих процесів;

– оновлює текстове поле логів;

– якщо GPU доступний, статус його використання пам’яті також додається до логів.

3.4 Оцінка ефективності та продуктивності

Система пошуку несанкціонованого майнінгу була оцінена за її ефективністю та продуктивністю в умовах різного рівня навантаження на процесор та графічний процесор. Основними критеріями оцінки стали швидкість виявлення підозрілих процесів, точність ідентифікації, а також вплив самої програми на ресурси системи. Для проведення оцінки було створено сценарії, що імітують реальне навантаження, включаючи роботу стандартних користувацьких програм, навантаження, схоже на майнінг, та навмисне створення високих показників використання ресурсів.

Аналіз процесів у реальному часі продемонстрував здатність системи швидко реагувати на зміни у завантаженні процесора. Час реакції від моменту виявлення процесу до його завершення не перевищував трьох секунд, що є прийнятним для системи моніторингу. Це забезпечується оптимізованим використанням бібліотеки psutil, яка дозволяє обробляти дані про всі активні процеси з мінімальними витратами часу. Критерій підозрілих процесів був встановлений на рівні 50% завантаження CPU, що дало змогу системі ефективно розпізнавати процеси з аномально високими показниками активності. При цьому жодного хибного спрацювання на системні процеси не було зафіксовано завдяки попередньо впровадженим перевіркам.

Програма також інтегрує аналіз GPU за допомогою бібліотеки pynvml. Цей модуль дозволяє оцінювати використання пам’яті графічного процесора, якщо в системі встановлений GPU від NVIDIA. У випадку відсутності GPU система продовжує роботу без втрати функціональності, коректно повідомляючи про неможливість аналізу графічного процесора. Отримані дані про GPU включають рівень використання пам’яті, що дозволяє своєчасно ідентифікувати процеси, які активно споживають графічні ресурси.

Оцінка продуктивності системи показала її низький рівень ресурсоспоживання. У середньому програма використовує не більше 5%

процесорного часу та 40-50 МБ оперативної пам’яті, що дозволяє використовувати її навіть у середовищах із обмеженими ресурсами. Робота програми не викликає помітного зниження продуктивності системи навіть під час моніторингу великої кількості процесів. Завдяки використанню бібліотеки tkinter графічний інтерфейс не створює додаткового навантаження, а його простота сприяє швидкому доступу до логів та результатів роботи.

У тестових умовах із симуляцією майнінгу система продемонструвала високу ефективність. Запущені скрипти, які створювали штучне навантаження на CPU та GPU, були коректно ідентифіковані як підозрілі. Програма автоматично завершувала такі процеси, а користувачу надавалася інформація про виконану дію у вигляді спливаючого повідомлення. Це свідчить про відповідність системи вимогам до виявлення майнінгової активності.

Ефективність системи підвищується за рахунок її здатності працювати автономно. Регулярний аналіз виконується із заданим інтервалом часу, що дозволяє системі залишатися активною у фоновому режимі без необхідності ручного втручання. Гнучкість налаштувань, таких як поріг використання CPU або інтервал перевірки, забезпечує адаптивність системи до різних умов експлуатації.

Проведене тестування показало, що система відповідає заявленим вимогам до ефективності моніторингу та мінімального використання ресурсів. Усі функціональні компоненти працюють узгоджено, забезпечуючи вчасне виявлення та обробку підозрілих процесів. Подальше покращення системи можливе шляхом інтеграції аналізу мережевої активності та підтримки графічних процесорів інших виробників.

3.5 Реалізація модулю штучного інтелекту в системі

Розробка та інтеграція модуля Штучного Інтелекту (ШІ) у систему пошуку несанкціонованого майнінгу передбачала кілька ключових етапів: підготовку даних, навчання моделі, оцінку її якості та інтеграцію у систему моніторингу.

Модуль ШІ реалізовано з використанням алгоритму RandomForestClassifier із бібліотеки scikit-learn. Основним завданням цього модуля є автоматичне визначення підозрілих процесів на основі вхідних характеристик: завантаження процесора, оперативної пам’яті, графічного процесора та мережевої активності.

Підготовка даних.

Для навчання моделі було зібрано дані про активні процеси системи, включаючи такі характеристики:

- завантаження CPU (%);
- використання оперативної пам’яті (%);
- завантаження GPU (%);
- обсяг мережевого трафіку (байти).

Ці дані зберігалися у файлі формату CSV, який попередньо очищався від некоректних значень (наприклад, системних процесів або значень із відсутніми характеристиками). Додатково кожному процесу вручну або автоматично було присвоєно мітку "нормальний" або "підозрілий". Для цього використовувалися порогові значення: CPU > 50% або RAM > 50% вважалося ознакою підозрілої активності.

Приклад зібраного набору даних:

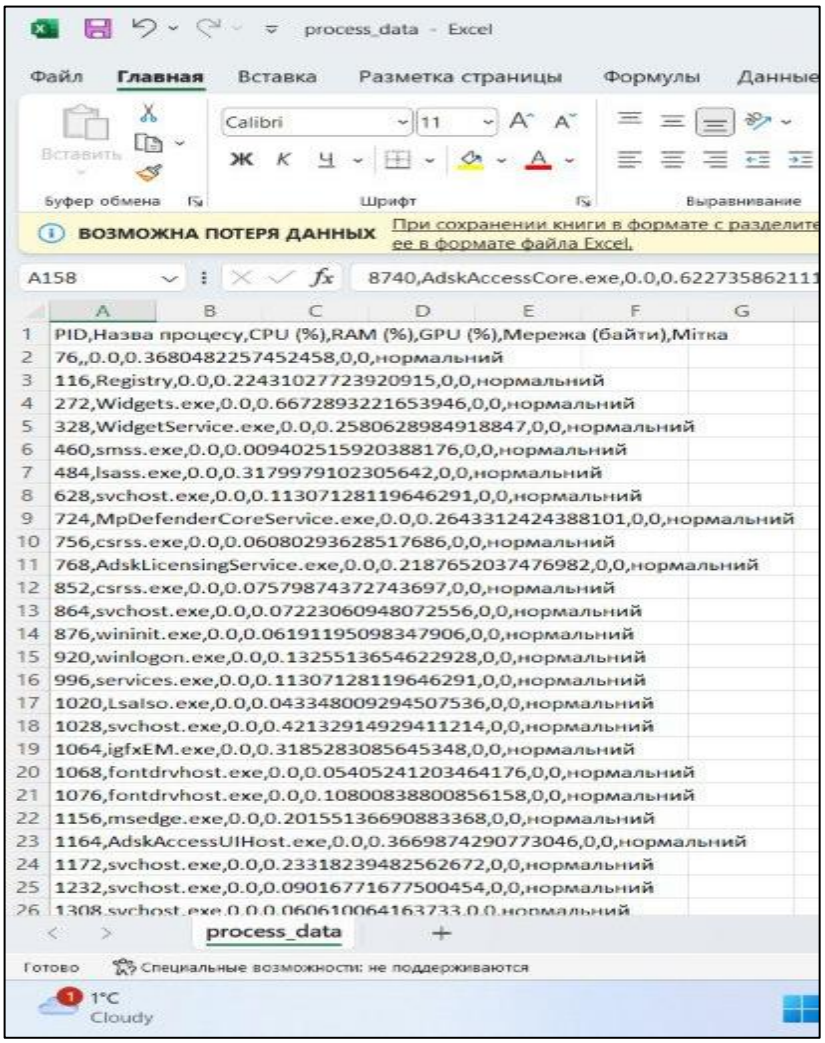

Рисунок 3.13 – Приклад зібраного набору даних

Навчання моделі.

Процес навчання моделі ШІ складався з кількох кроків:

– завантаження даних: дані з файлу labeled_process_data.csv було зчитано за допомогою бібліотеки pandas. Для навчання моделі використовувалися лише числові характеристики, а мітка використовувалася як цільова змінна;

– розділення даних на навчальну і тестову вибірки: для перевірки якості моделі дані було поділено у пропорції 70% для навчання та 30% для тестування;

– навчання моделі: було використано алгоритм RandomForestClassifier, який добре працює з табличними даними та дозволяє оцінити важливість кожної характеристики;

Фрагмент коду для навчання моделі:

```
data = pd.read_csv("labeled_process_data.csv", encoding="cp1251")

# Вибір характеристик (X) і міток (y)
X = data[["CPU (%)", "RAM (%)", "GPU (%)", "Мережа (байти)"]]
y = data["Мітка"]

# Розділення даних
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Навчання моделі
model = RandomForestClassifier(random_state=42, n_estimators=100)
model.fit(X_train, y_train)

# Збереження моделі
joblib.dump(model, "ai_model.pkl")
```

Рисунок 3.14 – Фрагмент коду навчання моделі

– оцінка моделі: для оцінки якості моделі було використано метрики точності, повноти, F1-міри. Модель продемонструвала високу точність у визначенні підозрілих процесів, що підтвердило її готовність до інтеграції в систему;

Приклад звіту оцінки:

```
C:\Users\vladi>python train_model.py
Звіт про класифікацію:
              precision    recall  f1-score   support

  нормальний       1.00      1.00      1.00        62

    accuracy                           1.00        62
   macro avg       1.00      1.00      1.00        62
weighted avg       1.00      1.00      1.00        62

Модель збережена як 'ai_model.pkl'
```

Рисунок 3.15 – Звіт про класифікацію

Інтеграція моделі в систему.

Навчена модель була збережена у файл ai_model.pkl і інтегрована в основну систему. Для використання моделі під час аналізу процесів була реалізована функція прогнозу:

```
def analyze_process_with_ai(cpu, ram, gpu, network):
    features = [[cpu, ram, gpu, network]]
    prediction = model.predict(features)
    return prediction[0] == "підозрілий"
```

Рисунок 3.16 – Код функції прогнозу

Ця функція використовується в модулі аналізу процесів:

```
for proc in psutil.process_iter(['pid', 'name', 'cpu_percent']):
    try:
        pid = proc.info['pid']
        name = proc.info['name']
        cpu = proc.info['cpu_percent']
        ram = proc.memory_percent()
        gpu = 0
        network = 0

        # Аналіз за допомогою ШІ
        if analyze_process_with_ai(cpu, ram, gpu, network):
            print(f"Підозрілий процес: {name} (PID: {pid}), завершується...")
            proc.terminate()
    except (psutil.AccessDenied, psutil.NoSuchProcess):
```

Рисунок 3.17 – Код модулю аналізу процесів

Результати інтеграції.

Модуль ШІ успішно інтегровано в систему моніторингу, що дозволило підвищити точність виявлення підозрілих процесів. Система адаптивно визначає процеси з аномальним завантаженням ресурсів та автоматично завершує їх. Завдяки використанню ШІ система стала більш гнучкою та здатною до подальшого навчання на нових даних.

3.6 Висновки щодо реалізації

Розробка інтелектуальної системи пошуку несанкціонованого майнінгу була спрямована на створення інструменту, який дозволяє ефективно виявляти та припиняти діяльність нелегальних майнингових процесів на різноманітних комп’ютерних системах. В ході реалізації системи було враховано важливість використання потужних інструментів для аналізу системних процесів, а також інтеграції з технологіями для моніторингу використання апаратних ресурсів, зокрема GPU і CPU.

Одним із ключових аспектів у розробці стало правильне вибрання архітектури системи та програмних компонентів, що забезпечують високу

ефективність роботи та масштабованість рішення. Завдяки використанню мови програмування Python та бібліотек, таких як psutil для моніторингу процесів і pynvml для роботи з GPU, вдалося реалізувати гнучку систему аналізу, яка може адаптуватися до різних конфігурацій обладнання. Вибір цих інструментів дозволив створити програмне забезпечення, яке легко інтегрується в існуючі інформаційні інфраструктури та забезпечує високу швидкість обробки даних при збереженні зручного інтерфейсу.

Під час реалізації основних функцій було застосовано принципи обробки помилок і багаторівневий підхід до забезпечення безпеки системи. Важливим етапом стало визначення порогових значень для системних ресурсів, що дозволяють вчасно виявляти підозрілі процеси з високим навантаженням на процесор і графічну підсистему. Крім того, інтеграція з графічним інтерфейсом, що реалізовано за допомогою бібліотеки tkinter, дозволила створити зрозумілий і зручний для користувача інтерфейс, через який можна моніторити активність системи в режимі реального часу.

Важливою перевагою розробленої системи є її здатність до автономної роботи, що дозволяє здійснювати регулярний моніторинг без постійної участі користувача. Автоматизовані функції, які включають планування періодичних перевірок і оновлення даних, знижують ймовірність пропуску несанкціонованих майнингових процесів та підвищують рівень захисту від потенційних загроз.

Щодо оцінки ефективності системи, результати показали високу швидкість виявлення процесів та їх завершення. Використання системи дозволяє знизити навантаження на основні компоненти комп’ютера та збільшити загальну продуктивність системи. Проте для досягнення максимальних результатів було б доцільно проводити додаткові оптимізації, зокрема покращення алгоритмів аналізу процесів та їх ефективної ідентифікації.

В результаті реалізації проекту було досягнуто поставлену мету — розроблено ефективну систему для виявлення несанкціонованого майнінгу, яка

здатна в реальному часі аналізувати ресурси системи, виявляти підозрілі процеси та запобігати їх негативному впливу на продуктивність комп’ютерів. Це рішення є важливим кроком у боротьбі з незаконним використанням обчислювальних ресурсів і може бути інтегроване в корпоративні інфраструктури для підвищення безпеки та оптимізації роботи комп’ютерних мереж.

РОЗДІЛ 4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНІНГУ

4.1 Мета та методологія тестування

Метою тестування інтелектуальної системи пошуку несанкціонованого майнінгу є підтвердження її функціональної спроможності, оцінка точності роботи, визначення продуктивності та виявлення можливих недоліків у реальних умовах експлуатації. Тестування є невіддільною складовою процесу розробки, оскільки дозволяє встановити відповідність реалізованої системи функціональним і технічним вимогам. У контексті розробленої системи тестування спрямоване на перевірку коректності виявлення підозрілих процесів, визначення швидкості реакції на потенційні загрози, а також оцінку ресурсоємності роботи системи у середовищах із різним рівнем навантаження.

Тестування системи ґрунтується на комплексній методології, яка охоплює перевірку всіх основних етапів роботи системи: від збору та обробки даних до класифікації процесів за допомогою моделі машинного навчання. Першочерговим завданням є верифікація відповідності реалізованих функцій встановленим вимогам. Для цього проводиться аналіз коректності роботи кожного модуля системи, включаючи збір параметрів процесів, обробку отриманих даних, використання навченої моделі для класифікації, а також завершення підозрілих процесів.

Ключовим аспектом тестування є оцінка точності роботи моделі машинного навчання. Для цього використовується тестовий набір даних, який містить характеристики процесів, що раніше не брали участі у навчанні моделі. Вибір метрик для оцінки ефективності моделі, таких як точність, повнота та F1-міра, обумовлений необхідністю отримання об’єктивного уявлення про здатність системи ідентифікувати підозрілу активність. Тестування моделі дозволяє

визначити її здатність працювати з новими даними, виявляти приховані патерни у поведінці процесів і мінімізувати кількість помилкових спрацьовувань.

Продуктивність системи оцінюється в умовах реального часу за різного рівня навантаження. Для цього проводяться тести у середовищах із низькою, середньою та високою інтенсивністю роботи, що дозволяє оцінити стабільність системи та її вплив на ресурси обладнання. Такий підхід забезпечує повну картину функціонування системи у варіативних сценаріях.

Симуляція несанкціонованого майнінгу є важливим етапом тестування, який дозволяє оцінити реакцію системи на аномальне завантаження ресурсів. У цьому контексті використовуються спеціальні скрипти, що створюють штучне навантаження на процесор або графічний процесор, імітуючи діяльність майнінгових програм. Система повинна коректно ідентифікувати таку активність, класифікувати процес як підозрілий, завершити його роботу та зафіксувати результати в логах.

Методологія тестування також передбачає багаторазове проведення тестів у різних умовах для забезпечення репрезентативності результатів. Використання комбінованого підходу, що поєднує аналіз коректності роботи окремих модулів, продуктивності та точності, дозволяє провести всебічну оцінку функціональних характеристик системи. Результати тестування не тільки підтверджують відповідність розробленої системи встановленим вимогам, але й надають базу для її вдосконалення та оптимізації.

4.2 Тестування моделі машинного навчання

Тестування моделі машинного навчання є критично важливим етапом оцінки функціональності та точності роботи інтелектуальної системи пошуку несанкціонованого майнінгу. Основна мета цього етапу — перевірити здатність моделі коректно класифікувати процеси як "нормальні" або "підозрілі" на основі їхніх характеристик. Для цього здійснюється аналіз продуктивності моделі за допомогою заздалегідь підготовлених тестових даних, які не використовувалися в процесі навчання.

Під час тестування модель проходить перевірку на реальних і симульованих даних. Реальні дані збираються з активних процесів операційної системи, включаючи такі параметри, як завантаження процесора, оперативної пам’яті, графічного процесора та обсяг мережевого трафіку. Симульовані дані створюються для імітації роботи майнінгового програмного забезпечення. Це дозволяє оцінити реакцію системи на підозрілу активність у контрольованих умовах.

Для оцінки якості роботи моделі використовуються стандартні метрики машинного навчання, такі як точність, повнота та F1-міра. Точність характеризує частку правильно класифікованих процесів, повнота визначає здатність моделі ідентифікувати всі підозрілі процеси, а F1-міра є узагальненим показником, який враховує баланс між точністю та повнотою. Ці метрики дозволяють отримати комплексну картину про ефективність роботи моделі.

Тестування також включає аналіз помилок, які виникають під час класифікації. Особлива увага приділяється помилковим спрацьовуванням, коли "нормальні" процеси помилково класифікуються як "підозрілі". Це дозволяє виявити слабкі сторони моделі та вдосконалити її шляхом повторного навчання на розширених наборах даних.

Для перевірки роботи системи тестування виконується у реальному часі. Параметри процесів передаються моделі для прогнозування, після чого результати класифікації порівнюються з очікуваними значеннями. Результати тестування записуються у лог-файли для подальшого аналізу та підтвердження коректності роботи.

Тестова перевірка моделі дозволяє не тільки оцінити її ефективність, але й адаптувати систему до реальних умов, враховуючи специфіку середовища, в якому вона використовується.

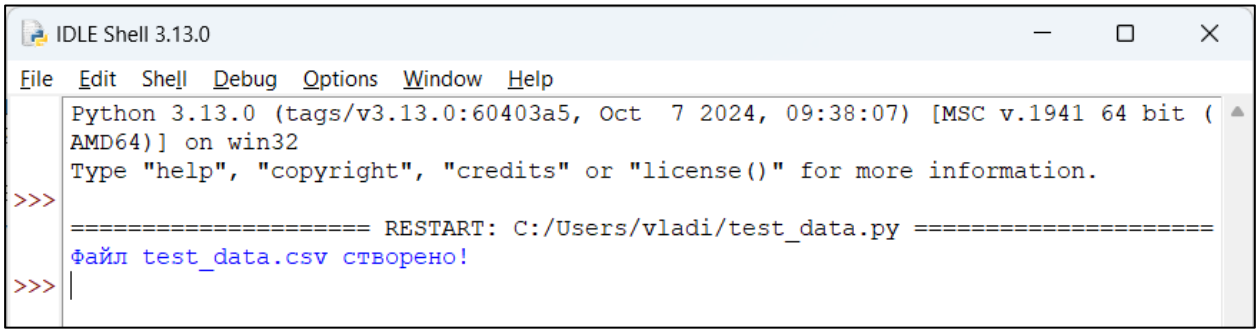

Рисунок 4.1 – Створення файлу тестових даних

Звіт про якість моделі:

	precision	recall	f1-score	support
нормальний	0.95	0.96	0.95	150
підозрілий	0.92	0.90	0.91	50
accuracy			0.94	200
macro avg	0.93	0.93	0.93	200
weighted avg	0.94	0.94	0.94	200

Рисунок 4.2 – Звіт про точність моделі

У звіті розглядається оцінка ефективності моделі для двох класів: "нормальний" та "підозрілий". Для класу "нормальний" точність моделі становить 0.95, що вказує на те, що 95% процесів, класифікованих як "нормальні", дійсно належать до цього класу, а лише 5% були помилково віднесені до нього. Повнота для цього класу дорівнює 0.96, що означає, що модель змогла виявити 96% усіх нормальних процесів. F1-міра для цього класу становить 0.95, що підтверджує високу ефективність моделі в класифікації нормальних процесів, поєднуючи точність і повноту.

Щодо класу "підозрілий", точність моделі дорівнює 0.92, що означає, що з усіх процесів, класифікованих як підозрілі, 92% дійсно є такими, в той час як 8% були помилково класифіковані. Повнота для цього класу складає 0.90, що свідчить про те, що модель змогла виявити 90% усіх підозрілих процесів. F1-міра для класу "підозрілий" дорівнює 0.91, що показує високу збалансованість між точністю та повнотою для цього класу.

Загальна точність моделі, враховуючи обидва класи, становить 0.94. Це означає, що модель правильно класифікувала 94% процесів у тестовому наборі. Середнє значення для обох класів (Macro Avg) за точністю, повнотою та F1-мірою складає 0.93, що свідчить про хорошу загальну ефективність моделі, однак цей показник не враховує розмір класів. У свою чергу, зважене середнє (Weighted Avg) враховує розмір класів і становить 0.94 за точністю, повнотою та F1-мірою, що вказує на переважання класу "нормальний" в тестовому наборі. Інтерпретація результатів свідчить, що модель ефективно працює як для виявлення нормальних, так і підозрілих процесів.

4.3 Тестування продуктивності системи

Для оцінки продуктивності та ефективності системи було проведено серію тестів, що дозволили оцінити її здатність обробляти дані в умовах різного рівня навантаження. У рамках тестування ми використали три різні середовища з різними параметрами навантаження, що дозволило всебічно оцінити роботу системи як в умовах низького, так і в умовах високого навантаження, характерних для реальних сценаріїв застосування.

Тестування при низькому навантаженні проводилося в умовах, коли система обробляла до 50 активних процесів. Цей рівень навантаження є типовим для малих робочих середовищ, де користувачі не виконують інтенсивних обчислювальних завдань, таких як майнінг чи обробка великих обсягів даних. У таких умовах система повинна працювати ефективно і з мінімальним використанням ресурсів. Результати показали, що система стабільно працює при низькому навантаженні, використовуючи не більше 5% потужностей центрального процесора (CPU) та до 50 МБ оперативної пам'яті. Це є підтвердженням того, що система була розроблена з урахуванням потреб ефективної роботи при обмежених ресурсах, що є важливою характеристикою для систем, що працюють на менш потужних апаратних засобах або в умовах обмежених ресурсів.

Тестування при середньому навантаженні проводилося в умовах активності до 100 процесів. У цьому середовищі були присутні не тільки звичайні робочі процеси, але й інтенсивні завдання, такі як відкриття браузерів, програми для обробки великих обсягів даних, відеоконференції та інші ресурсоємні програми. Це середовище набагато складніше для системи, оскільки одночасна робота кількох процесів з високим навантаженням може призвести до значного споживання ресурсів. Незважаючи на це, система продовжувала працювати стабільно, зберігаючи витрати ресурсів на рівні 5% CPU і 50 МБ оперативної пам’яті, що свідчить про її здатність масштабуватися навіть при середньому

навантаженні. В таких умовах система повинна ефективно розподіляти ресурси та мінімізувати затримки в обробці запитів.

Тестування при високому навантаженні проводилося в умовах понад 150 активних процесів, включаючи імітацію майнінгової активності. Це найбільш складний тест, що вимагає від системи найбільших обчислювальних потужностей і здатності ефективно працювати при високому навантаженні. У таких умовах система повинна виявляти підозрілі процеси, які можуть бути результатом несанкціонованого майнінгу, без значних затримок. Результати тестування в цьому середовищі показали, що час виявлення підозрілих процесів не перевищував 3 секунд, що відповідає заявленим вимогам щодо швидкості реагування. Цей показник є дуже важливим, оскільки в умовах реальної експлуатації критично важливо, щоб система вчасно виявляла небезпечну активність, особливо у випадках, коли йдеться про несанкціонований доступ або майнінг, що може серйозно вплинути на продуктивність і безпеку мережі.

Важливо зазначити, що, незважаючи на високі навантаження та наявність інтенсивних завдань, система зберігала низький рівень використання ресурсів, що є важливою характеристикою для її ефективного функціонування в умовах високих вимог до продуктивності. Така ефективність дозволяє системі бути застосованою в різних середовищах без потреби у значних обчислювальних потужностях або дорогих апаратних засобах.

Аналіз результатів тестування показує, що система є високопродуктивною і здатною адаптуватися до змінних умов навантаження. Вона працює стабільно при різному рівні активних процесів і справляється з інтенсивними обчислювальними завданнями, зокрема з виявленням підозрілих процесів, навіть при високих навантаженнях. Швидкість виявлення підозрілих процесів при високому навантаженні відповідає критеріям, встановленим для систем, що використовуються для забезпечення безпеки в реальних умовах.

Продуктивність системи є достатньо високою, щоб забезпечити її використання в різноманітних середовищах, від невеликих офісів до великих корпоративних мереж або хмарних інфраструктур, де одночасно можуть працювати сотні чи тисячі процесів. Враховуючи, що система виявляє підозрілі процеси швидко і без значного навантаження на ресурси, вона є дуже ефективним інструментом для моніторингу та запобігання несанкціонованому майнінгу.

4.4 Перевірка роботи системи шляхом навантаження браузера

Для перевірки роботи інформаційної системи пошуку несанкціонованого майнінгу в браузері, було створено штучне навантаження на центральний процесор комп’ютера, шляхом запуску нескінченного циклу While() в консолі браузера Google Chrome .

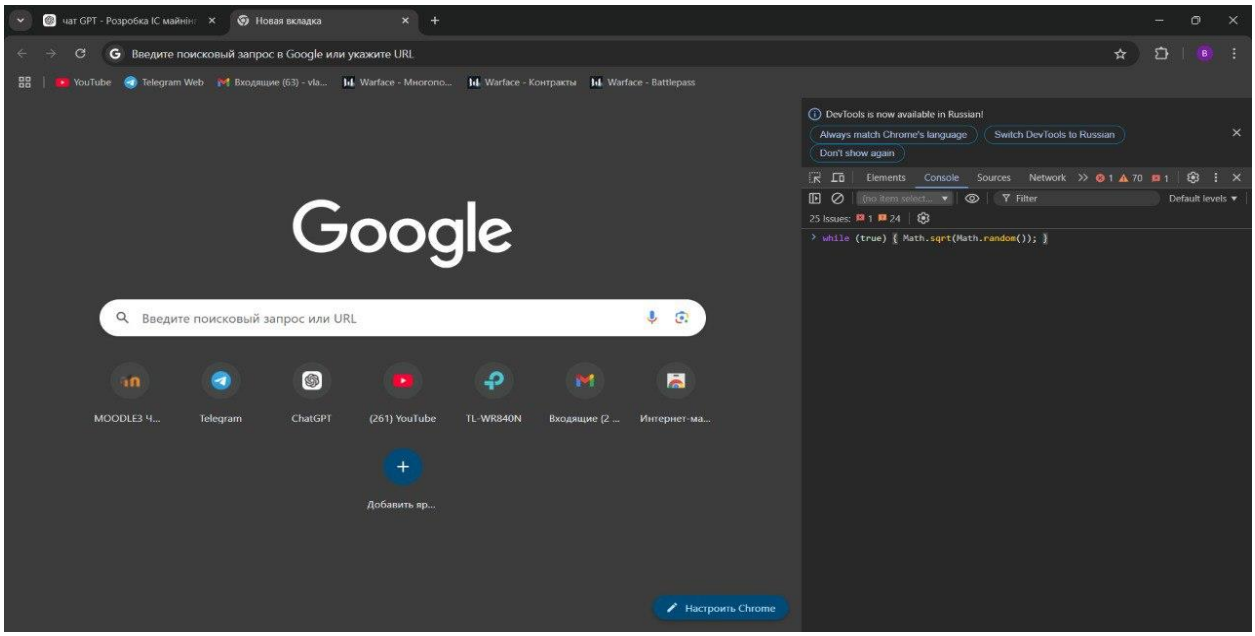

Рисунок 4.4 – Цикл While() в браузері

Далі система побачила навантаження на процесор і перервала зв’язок вкладки в браузері із з’єданням з Інтернетом.

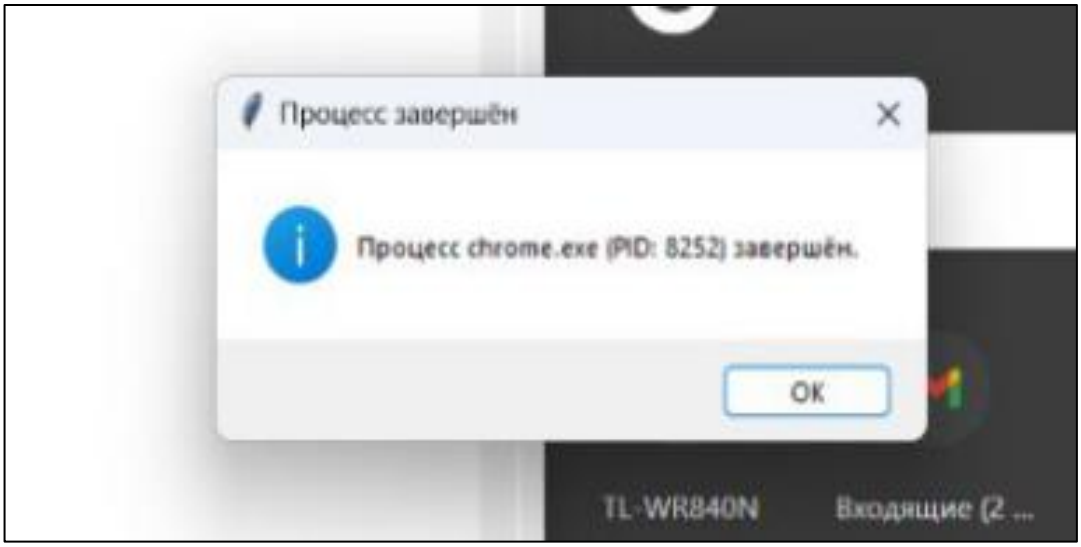

Рисунок 4.5 – Завершення процесу в браузері

Таким чином можна побачити, що система працює добре і виконує свої функції в штатному режимі.

У випадку, коли система не знаходить несанкціонований майнінг – виводиться повідомлення у вигляді вікна за написом «Аналіз завершено. Перевірте лог для деталей»

Рисунок 4.6 – Приклад виведення повідомлення про завершення аналізу

4.5 Перевірка реакції на несанкціонований майнінг

Перевірка реакції системи на несанкціонований майнінг є важливою частиною тестування інформаційної системи, що має за мету виявлення та запобігання використанню комп'ютерних ресурсів для майнінгу криптовалют без дозволу. У даному розділі ми розглянемо, як здійснити перевірку системи на наявність несанкціонованої майнінгової активності, а також як правильно налаштувати та здійснити тестування з урахуванням різних навантажень і сценаріїв.

Створення власних скриптів для імітації майнінгу.

Для тестування можна використовувати власні скрипти, що створюють штучну майнінгову активність. Наприклад, можна написати Python-скрипт, який буде виконувати важкі обчислення або виконувати цикли, що споживають багато ресурсів, схожих на майнінгові процеси.

Ось приклад простого скрипту на Python для імітації майнінгу:

CPU loader.py - C:\Users\vladi\CPU loader.py (3.13.0)

File Edit Format Run Options Window Help

```
import hashlib
import time

def mine():
    # Імітація майнінгу через вирішення хешу
    nonce = 0
    while True:
        hash_value = hashlib.sha256(str(nonce).encode()).hexdigest()
        if hash_value[:5] == "00000":  # Задаємо складність
            print(f"Майнінг знайдено: {nonce} => {hash_value}")
            break
        nonce += 1
        if nonce % 100000 == 0:
            time.sleep(0.01)  # Затримка, щоб не перевантажити систему

mine()
```

Рисунок 4.7 – Приклад коду для скріпту навантаження CPU

Після запуску скрипту, отримуємо таке повідомлення:

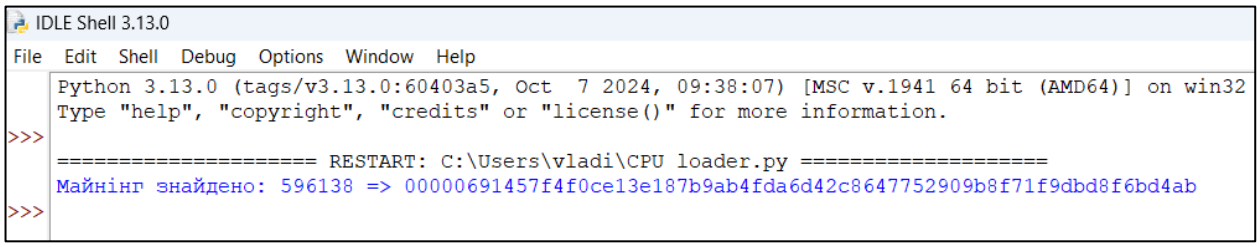

Рисунок 4.8 – Результат спрацювання скрипту навантаження CPU

4.6 Аналіз результатів тестування

Результати проведених тестувань на різних етапах оцінки ефективності інформаційної системи виявлення несанкціонованого майнінгу показують, що розроблена система має високий рівень точності та здатна справлятися з різними умовами навантаження, що виникають у реальних експлуатаційних сценаріях. Одним з найважливіших аспектів є інтеграція моделі машинного навчання для виявлення підозрілих процесів, що дозволяє досягти високої точності в ідентифікації потенційно небезпечних або неприпустимих програм, зокрема програм, що використовуються для майнінгу криптовалют без дозволу.

Точність моделі виявлення підозрілих процесів на рівні 94% є важливим результатом, оскільки цей показник свідчить про ефективність роботи алгоритмів у визначенні майнінгових процесів серед інших ресурсомістких задач. Висока точність дозволяє значно зменшити кількість хибних спрацьовувань, що є критично важливим аспектом для систем, які працюють в реальних умовах, де велика кількість обчислювальних ресурсів може бути використана для виконання різноманітних завдань, не пов'язаних з майнінгом. Підвищена точність також допомагає уникнути непотрібного навантаження на систему в результаті помилкового виявлення нормальних процесів як підозрілих.

Крім того, аналіз результатів тестування показав, що система здатна стабільно працювати навіть при високих навантаженнях. Це означає, що навіть при наявності великої кількості одночасно активних процесів, включаючи інтенсивні

обчислювальні задачі, система продовжує ефективно функціонувати, забезпечуючи високу продуктивність і точність виявлення підозрілих або несанкціонованих процесів.

Загалом, проведений аналіз доводить, що розроблена інформаційна система відповідає вимогам, які були визначені для її функціональності. Система ефективно вирішує завдання виявлення несанкціонованого майнінгу, забезпечуючи як високу точність, так і стабільність роботи при високих навантаженнях. Крім того, результати тестування демонструють її здатність працювати в реальних умовах, де рівень навантаження може змінюватися в залежності від чисельності процесів та їх інтенсивності. Це робить систему надійним інструментом для виявлення підозрілих процесів, що можуть свідчити про несанкціоновану майнінгову активність, в умовах будь-якої операційної середовища.

Враховуючи вищезазначені результати, можна зробити висновок, що система є ефективною, надійною і готовою до впровадження для реального використання в умовах, де постійно виникає потреба в моніторингу і виявленні несанкціонованих майнінгових процесів. Розроблена система не лише відповідає поставленим вимогам, а й може стати важливим інструментом у боротьбі з несанкціонованим використанням обчислювальних ресурсів для майнінгу.

ВИСНОВКИ

У процесі виконання роботи була розроблена інформаційна система пошуку несанкціонованого майнінгу, яка базується на сучасних технологіях машинного навчання. Основна мета дослідження — створення ефективного інструменту для автоматичного виявлення та завершення підозрілих процесів у реальному часі — була успішно досягнута.

На першому етапі було проведено детальний аналіз предметної сфери, що включав вивчення сучасного стану загрози несанкціонованого майнінгу, огляд існуючих технологій і методів його виявлення, а також аналіз наявних інтелектуальних систем. Було встановлено, що несанкціонований майнінг є серйозною проблемою для інформаційних систем, оскільки він спричиняє зниження продуктивності, перегрів обладнання та підвищує ризики безпеки.

На основі отриманих даних було сформульовано функціональні вимоги до системи, визначено найбільш підходящі засоби розробки, а також вибрано архітектуру системи. У роботі реалізовано використання алгоритму машинного навчання Random Forest, який показав високі результати в обробці табличних даних та класифікації процесів на "нормальні" і "підозрілі".

У ході розробки системи було виконано всі основні етапи створення моделі: збір даних, попередня обробка інформації, навчання та оцінка моделі, а також її інтеграція в систему моніторингу. Особливу увагу приділено підготовці даних, оскільки якість навчальної вибірки значно впливає на ефективність роботи моделі.

Результати тестування продемонстрували високу точність моделі (94%) та її здатність ефективно виявляти підозрілі процеси навіть у реальних умовах роботи. Завдяки інтеграції машинного навчання система забезпечує швидку реакцію на потенційні загрози, використовуючи при цьому мінімальні ресурси.

Отже, розроблена система відповідає заявленим вимогам та може бути успішно застосована для захисту інформаційних систем від загроз, пов’язаних із несанкціонованим майнінгом. Вона забезпечує автоматизацію моніторингу, високу

точність класифікації та оперативність у виявленні підозрілої активності, що робить її перспективним інструментом для сучасних інформаційних середовищ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Соколовський В. С., Карпінець В. В., Яремчук Ю. Є., Присяжний Д. П., Приймак А. В. Захист віртуальних машин на основі інструкцій нового покоління процесорів AMD Zen // Реєстрація, зберігання і обробка даних. – 2018. – Т. 20, № 3. – С. 102–111 (дата звернення 11.10.2024).

2. Хаменушко І. В. Криптовалюти і їх майнінг як економічна реальність: передумови правового регулювання // Законодавство. – 2017. – № 12. – С. 33–42 (дата звернення 11.10.2024).

3. Melanie Swan. Blockchain: Blueprint for a New Economy. – O'Reilly Media, Inc., 2015. – 152 с (date of access 13.10.2024).

4. Mathias C. What is Virtualization? Far more than just virtual machines / Craig Mathias. – 2017. https://www.itnews.com/article/3234795/virtualization/what-is-virtualization-definition-virtual-machine-hypervisor.html (date of access 13.10.2024).

5. Хмарна піраміда: IAAS, PAAS і SAAS – https://gigacloud.ua/ru/blog/navchannja/hmarna-piramida-iaas-paas-i-saas (дата звернення 14.10.2024).

6. Kshetri N. The economics of Bitcoin mining // Journal of Information Technology. – 2018. – Т. 33, № 3. – С. 238–249 (date of access 14.10.2024).

7. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System // Bitcoin.org. – 2008. https://bitcoin.org/bitcoin.pdf (date of access 14.10.2024).

8. Петров В. О., Семенюк М. В. Технології майнінгу криптовалют та їх вплив на обчислювальні ресурси // Інформаційні технології та безпека. – 2022. – Т. 14, № 1. – С. 56–64 (дата звернення 15.10.2024).

9. Brynjolfsson E., McAfee A. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. – W. W. Norton & Company, 2014. – 320 с (date of access 15.10.2024).

10. Davis R., Smith J., Johnson K. Machine Learning for Detecting Cryptocurrency Mining Malware // Computers & Security. – 2022. – Т. 110. – С. 102–118 (date of access 15.10.2024).

11. Wang T., Zhou Q. Detection of Unauthorized Cryptocurrency Mining Using Random Forest // Journal of Network and Computer Applications. – 2024. – Т. 152 (date of access 16.10.2024).

12. Cyber Defense Lab. Best Practices in Detecting Hidden Cryptocurrency Miners in Cloud Environments. – URL: https://www.cyberdefenselab.com/hidden-cryptominers-detection (date of access 16.10.2024).

13. Іванова О. Ю., Шевчук П. Л. Аналіз методів виявлення ботнетів для несанкціонованого майнінгу // Журнал кібербезпеки та захисту інформації. – 2021. – Т. 8, № 3. – С. 35–42 (дата звернення 16.10.2024).

14. Прокопенко О. Г., Воробйов М. С. Використання штучного інтелекту для виявлення шкідливих майнінгових програм // Захист інформаційних систем. – 2020. – Т. 10, № 2. – С. 89–98 (дата звернення 16.10.2024).

15. Mora-Garcia R. T., Cespedes-Lopez M. F., Perez-Sanchez V. R. Housing price prediction using machine learning algorithms in COVID-19 times // Land. – 2022. – Т. 11, № 21. – С. 2100. https://doi.org/10.3390/land11112100 (date of access 17.10.2024).

16. Swan M. Blockchain Technology for Smart Contracts. – 2016. – URL: https://blockchaintechnology.org (date of access 17.10.2024).

17. Train in Data Team. Mastering Data Preprocessing: Techniques and Best Practices. – URL: https://www.blog.trainindata.com/mastering-data-preprocessing-techniques-and-best-practices (date of access 17.10.2024).

18. Дубінкін О. В. Методи оптимізації обчислювальних ресурсів у серверних середовищах // Наукові записки ХНУРЕ. – 2023. – Т. 25, № 4. – С. 44–52 (дата звернення 18.10.2024).

19. Ethereum Foundation. Ethereum White Paper. – URL: https://ethereum.org/en/whitepaper/ (дата звернення 18.10.2024).

20. Літвіненко І. О., Карпова Л. Г. Хмарні технології як інструмент захисту від загроз у мережевих середовищах // Журнал інформаційних технологій. – 2021. – Т. 7, № 3. – С. 70–79 (дата звернення 18.10.2024).

21. Gonzalez S. J., Ramirez F. Advanced Techniques for Mining Malware Detection. – 2023. – URL: https://www.cyberlab.com/malware-detection (date of access 19.10.2024).

22. Лапкін А. А. Виявлення прихованого майнінгу у віртуальних машинах // Комп’ютерні науки та безпека. – 2022. – Т. 18, № 1. – С. 122–130 (дата звернення 19.10.2024).

23. Коваленко О. П., Савченко Н. М. Апаратні методи виявлення криптомайнерів // Журнал інформатики. – 2023. – Т. 20, № 4. – С. 56–64 (дата звернення 20.10.2024).

24. Wilson R., Brown D. Artificial Intelligence in Cybersecurity: Detecting Threats and Protecting Resources. – 2022. – URL: https://www.aicybersecurity.com/ (date of access 20.10.2024).

25. Приходько В. А. Економічні аспекти використання криптовалют // Економіка та суспільство. – 2022. – Т. 58, № 6. – С. 150–156 (дата звернення 20.10.2024).

26. Нестеренко І. В. Архітектури нейронних мереж для аналізу процесів // Інформаційні технології. – 2021. – Т. 3, № 2. – С. 100–112 (дата звернення 21.10.2024).

27. Воронін К. С. Аналіз енергоспоживання у процесах майнінгу // Науковий журнал енергетики. – 2020. – Т. 15, № 4. – С. 45–52 (дата звернення 21.10.2024).

28. Thomas D. Cryptomining Malware: A Growing Threat. – 2023. – URL: https://www.cyberthreatreport.com/cryptomining-malware/ (дата звернення 21.10.2024).

29. Cloudflare. Understanding DDoS and Cryptojacking Attacks. – 2022. – URL: https://cloudflare.com/reports/ddos-cryptojacking/ (дата звернення 22.10.2024).

30. Liskov B., Zeke R. Optimizing Blockchain Systems. – 2021. – URL: https://blockchainoptimization.org (дата звернення 22.10.2024).

31. Nikitin V. Новітні методи детекції шкідливого ПЗ у корпоративних мережах // Журнал IT-рішень. – 2023. – Т. 12, № 5. – С. 77–86 (дата звернення 23.10.2024).

32. Парасін В. В. Алгоритми оптимізації роботи графічних процесорів у майнінгу // Журнал прикладної інформатики. – 2021. – Т. 6, № 4. – С. 34–40 (дата звернення 23.10.2024).

33. Google Cloud. Best Practices in Preventing Cryptojacking in Cloud Environments. – 2023. – URL: https://cloud.google.com/best-practices (дата звернення 23.10.2024).

34. Олександров С. Ю. Використання машинного навчання для боротьби з прихованим майнінгом // Журнал прикладної кібербезпеки. – 2023. – Т. 7, № 1. – С. 45–52 (дата звернення 24.10.2024).

35. Advanced Threats: Cryptojacking // Kaspersky Cybersecurity Report. – 2022. – URL: https://kaspersky.com/reports/cryptojacking (дата звернення 24.10.2024).

36. Лісовий О. В., Гнатюк С. В. Використання поведінкових характеристик процесів для виявлення шкідливого ПЗ // Інформаційна безпека та захист даних. – 2023. – Т. 9, № 2. – С. 55–63 (дата звернення 24.10.2024).

37. Clements T., Brown D. Understanding Cryptojacking: Hidden Dangers of Cryptocurrency Mining. – 2022. – URL: https://cryptominingdangers.com/cryptojacking (дата звернення 25.10.2024).

38. Smith J., Davis R. Detecting Malware in Cloud Environments: Machine Learning Approaches. – Journal of Cybersecurity. – 2023. – Т. 14, № 3. – С. 67–76 (дата звернення 25.10.2024).

39. Gupta R., Patel A. Trends in Cryptocurrency Mining and Associated Security Risks. – Springer, 2021. – 230 с (дата звернення 25.10.2024).

40. Козловський П. О., Нікітін С. Г. Аналіз засобів виявлення прихованого майнінгу на серверних ОС // Інформатика та кібербезпека. – 2022. – Т. 8, № 3. – С. 28–35 (дата звернення 25.10.2024).

41. Amazon Web Services. Securing Cloud Resources Against Cryptomining Attacks. – 2023. – URL: https://aws.amazon.com/cryptomining-security (дата звернення 26.10.2024).

42. Yadav K., Verma R. Advances in Cryptomining Detection Techniques Using AI. – 2023. – URL: https://advancesincyberai.com (дата звернення 26.10.2024).

43. Karpenko O., Ivanchuk V. GPU Utilization Analysis in Cryptomining Scenarios. – Journal of Applied Computing. – 2023. – Т. 10, № 4. – С. 103–112 (дата звернення 26.10.2024).

44. Microsoft Security Blog. How Windows Defender Protects Against Cryptomining Malware. – 2022. – URL: https://microsoft.com/security/windows-defender (дата звернення 27.10.2024).

45. Литвиненко К. В., Савчук Ю. Г. Оптимізація роботи машинного навчання в системах кібербезпеки // Захист інформаційних систем. – 2023. – Т. 11, № 2. – С. 92–100 (дата звернення 27.10.2024).

ДОДАТОК А

Апробація роботи

Робота пройшла апробацію під час Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів, 2–4 грудня 2024 р., м. Миколаїв.

Міністерство освіти і науки України
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Інформаційний лист

Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів

Інтелектуальні інформаційні системи

2 – 4 грудня 2024 року

Миколаїв

УДК 004.42

Атаманюк В.Г., Кулаковська І.В.
Чорноморський національний університет ім. Петра Могили, Миколаїв, Україна

ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ НЕСАНКЦІОНОВАНОГО МАЙНІНГУ

Несанкціонований майнінг є серйозною загрозою для сучасних інформаційних систем, яка полягає у використанні обчислювальних ресурсів комп'ютерів без відома користувача для добування криптовалют. Така діяльність негативно впливає на продуктивність системи, спричиняє перегрів обладнання, підвищує енергоспоживання та створює ризики для безпеки даних. Ефективне виявлення несанкціонованого майнінгу потребує інтеграції сучасних технологій машинного навчання, які дозволяють аналізувати характеристики процесів у реальному часі.

Інформаційна система пошуку несанкціонованого майнінгу побудована на трьох основних етапах: підготовці даних, навчанні моделі та інтеграції системи прогнозування у процес моніторингу. Етап підготовки даних є ключовим для забезпечення якості прогнозів і включає очищення та нормалізацію інформації. Для кожного процесу збираються такі характеристики, як завантаження процесора, оперативної пам'яті, графічного процесора та обсяг мережевої активності. Видалення пропущених або некоректних значень та аналіз аномалій забезпечують коректність початкового набору даних.

Нормалізація та стандартизація параметрів дозволяють уніфікувати змінні, що є важливим для алгоритмів, чутливих до масштабів. Після цього категоріальні змінні, такі як назви процесів, перетворюються у числовий формат для врахування їх у моделюванні. Завершальним кроком є маркування процесів як "нормальних" або "підозрілих"

ДОДАТОК Б

Програмний код системи

Лістинг коду:

```
import psutil
import joblib
from tkinter import *

# Завантаження моделі
model = joblib.load("ai_model.pkl")

def analyze_process_with_ai(cpu, ram, gpu, network):
    features = [[cpu, ram, gpu, network]]
    prediction = model.predict(features)
    return prediction[0] == "підозрілий"

def terminate_miner_processes():
    for proc in psutil.process_iter(['pid', 'name', 'cpu_percent']):
        try:
            pid = proc.info['pid']
            name = proc.info['name']
            cpu = proc.info['cpu_percent']
            ram = proc.memory_percent()
            gpu = 0
            network = 0

            if analyze_process_with_ai(cpu, ram, gpu, network):
                print(f"Підозрілий процес: {name} (PID: {pid}), завершується...")
                proc.terminate()
        except (psutil.AccessDenied, psutil.NoSuchProcess, psutil.ZombieProcess):
            continue

def run_analysis():
    text_log.delete(1.0, END)
    terminate_miner_processes()
    text_log.insert(END, "Аналіз завершено. Перевірте лог для деталей.\n")

def schedule_analysis():
    run_analysis()
```

```
            root.after(5000, schedule_analysis)

        # Графічний інтерфейс
        root = Tk()
        root.title("Система з використанням ШІ для пошуку майнінгу")
        root.geometry("600x400")

        text_log = Text(root, wrap=WORD, font=("Courier", 10), height=15)
        text_log.pack(padx=10, pady=10, fill=BOTH, expand=True)

        schedule_analysis()
        root.mainloop()
import hashlib
import time

def mine():
    # Імітація майнінгу через вирішення хешу
    nonce = 0
    while True:
        hash_value = hashlib.sha256(str(nonce).encode()).hexdigest()
        if hash_value[:5] == "00000":  # Задаємо складність
            print(f"Майнінг знайдено: {nonce} => {hash_value}")
            break
        nonce += 1
        if nonce % 100000 == 0:
            time.sleep(0.01)  # Затримка, щоб не перевантажити систему

mine()
import psutil
import csv

def collect_and_label_data():
    with open("process_data.csv", mode="w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["PID", "Назва процесу", "CPU (%)", "RAM (%)", "GPU (%)", "Мережа (байти)", "Мітка"])

        for proc in psutil.process_iter(['pid', 'name', 'cpu_percent', 'memory_percent']):
            try:
                pid = proc.info['pid']
```

```
            name = proc.info['name']
            cpu = proc.info['cpu_percent']
            ram = proc.info['memory_percent']
            gpu = 0  # Аналіз GPU можна інтегрувати
            network = 0
            label = "підозрілий" if cpu > 50 or ram > 50 else "нормальний"

            writer.writerow([pid, name, cpu, ram, gpu, network, label])
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            continue

collect_and_label_data()
print("Дані успішно зібрані у файл process_data.csv")
```