Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів”

Виконав здобувач вищої освіти
ІІ курсу, групи КН-25МЗ
ОПП «Комп’ютерні науки»
спеціальності F3 «Комп’ютерні науки»
_____ Кривка А.С.
«____» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Кривка Артур Сергійович
Сертифікат: 3FAA9288358EC00304000000D3A731007E22E400
Дійсний з 13.05.2025 17:47:17 по 11.05.2027 17:47:17

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань _F_____*"Інформаційні технології"*_____
Спеціальність F3 *"Комп'ютерні науки"*
Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерні науки"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кривці Артуру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів*

2. Керівник роботи *Якименко Наталія Миколаївна, кандидат фізико-математичних наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 44-13 від 17.08.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна.*
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*8. Висновки.*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

8. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
8.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Кривка А.С. Дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів. ФЗ Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Метою розробки є дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Об'єктом дослідження є процес поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Предметом дослідження є методи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Методи дослідження базуються на методах поведінкового аналізу, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерні науки, поведінковий аналіз, обмеження активності додатків, моніторинг системних викликів

ABSTRACT

Kryvka A.S. Research and Software Implementation of a System for Behavioral Analysis and Application Activity Restriction Based on System Call Monitoring. F3 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification thesis for the second (master's) level of higher education, software has been developed for a system designed to perform behavioral analysis and restrict application activity based on the monitoring of system calls.

The objective of this development is the research and software implementation of a system for behavioral analysis and application activity restriction based on system call monitoring.

The object of the research is the process of behavioral analysis and application activity restriction based on system call monitoring.

The subject of the research consists of methods for behavioral analysis and the restriction of application activity based on system call monitoring.

The research methodology is based on techniques for behavioral analysis, mathematical statistics, and software development.

The outcome of the work is a software implementation of a system for behavioral analysis and application activity restriction based on system call monitoring.

During the development of the software model, an analysis of existing hardware and software tools was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed, and instructions for using the software tools are provided.

The program is designed for use on personal computers running the Windows 10/11 operating system.

The program was developed using the Python environment.

Keywords: computer science, behavioral analysis, application activity restrictions, system call monitoring

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	100	–	
2	А4	Робоча програма	25	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 9
ВСТУП..... 10
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 12
1.1 Призначення системи..... 12
1.2 Область застосування 12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 29
2.3 Розгорнута постановка завдання 30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 31
3.1 Опис функціонування системи 31
3.2 Розробка структурної схеми..... 36
3.3 Розробка функціональної схеми 42
3.4 Розробка діаграми процесів 44
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 46
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 75
6 НАУКОВА НОВИЗНА 81

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 82
7.1 Вступ 82
7.2 Аналіз умов праці 83
7.3 Техніка безпеки та протипожежна профілактика 87
7.4 Розробка заходів з охорони праці 89
7.5 Висновки до розділу 90
8 ОСНОВНІ ВИСНОВКИ 91
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

КМ	–	комп’ютерна мережа
КСАЗ	–	комплексна система антивірусного захисту
МЕ	–	міжмережний екран
ПЗ	–	програмне забезпечення
ПК	–	персональний комп’ютер
ACL	–	Access Control List
FTP	–	File Transfer Protocol
http	–	HyperText Transfer Protocol
POP3	–	Post Office Protocol Version 3
SMTP	–	Simple Mail Transfer Protocol
VLAN	–	Virtual Local Area Network

ВСТУП

Актуальність теми. Традиційні антивірусні програми (AV), які спиралися на певний набір сигнатур для виявлення шкідливої активності, стали застарілими. Ці інструменти порівнюють ці сигнатури з файлами в системі, щоб визначити, чи є файл пошкодженим. Хоча цей набір сигнатур постійно оновлюється, результат часто є грою в наздоганяння як для фахівців з безпеки, так і для користувачів AV. Натомість новітні технології ставить таке питання: що, якщо файл не є шкідливим за своєю суттю, але бере участь у шкідливій діяльності? Крім того, що, якщо шкідлива поведінка взагалі не пов'язана з файлом? Виявлення на основі поведінки є сучасним стандартом безпеки, оскільки традиційні антивірусні засоби виходять з моди. Фактично, переважна більшість шкідливих програм та зловмисної активності, яку розслідувала команда з цифрової криміналістики та реагування на інциденти (DFIR), не була виявлена традиційними антивірусними рішеннями, які були встановлені у жертв. Основне рішення цієї проблеми має два аспекти: антивірусне забезпечення наступного покоління (NGAV) та виявлення та реагування на кінцеві точки (EDR).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– Дослідження системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– Програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Об'єктом дослідження є процес поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Предметом дослідження є методи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Методи дослідження базуються на методах поведінкового аналізу, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– Розроблено вітчизняний продукт поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

NGAV поєднує найкраще з традиційного моніторингу на основі сигнатур із моніторингом процесів у режимі реального часу, щоб забезпечити чітке уявлення про ландшафт безпеки організації. EDR йде ще далі, надаючи централізовану консоль керування, з якої адміністратори можуть діяти, підключаючись до своїх мереж, щоб моніторити, розслідувати та реагувати на інциденти, а також користуватися перевагами безлічі функцій. Нові рішення, такі як рішення для керованого виявлення та реагування (MDR), що працює на передовій платформі XDR, йдуть ще далі.

Досі не впевнені, чи необхідне виявлення на основі поведінки для вашої організації? Пам’ятайте: майже в кожному розслідуваному нами витоку мережі правильно впроваджені рішення NGAV або EDR допомогли виявити активність на ранній стадії. Використання нашого керованого рішення XDR дозволило б нам виявити, стримати та усунути атаку, а також значно зменшити потенційну шкоду.

1.2 Область застосування

Приведемо декілька прикладів сучасних викликів безпеці інформації:

– Група вимагачів Blackbyte повертається з присутністю в Twitter та багаторівневим ціноутворенням. Групи програм-вимагачів продовжують посилювати тиск на своїх жертв, скорочуючи час, який вони мають для сплати викупу до того, як їхні дані стануть публічними. Переконайтеся, що комунікаційний план вашої організації містить чітку та рішучу реакцію на такі події.

– 35 000 репозиторіїв не зламано, але клони заполонили Github, розповсюджуючи шкідливе програмне забезпечення. Клоновані репозиторії GitHub («репозиторії») використовувалися для обману великої кількості користувачів, щоб вони мимоволі завантажували шкідливе програмне забезпечення. Захистіть своїх користувачів та свою організацію, забезпечивши використання лише перевірених репозиторіїв GitHub.

– Хакери з Північної Кореї помітили, як вони атакували шукачів роботи за допомогою шкідливого програмного забезпечення для MacOS. Було помічено, що підтримувана Північною Кореєю група Lazarus атакує шукачів роботи за допомогою шкідливого програмного забезпечення, здатного запускатися на комп'ютерах Apple з чіпсетами Intel та M1. Важливо пам'ятати, що комп'ютери Mac не застраховані від подій безпеки: організації повинні забезпечити їм такий самий рівень захисту, як і кінцеві точки Windows.

– Спонсоровані оголошення про роботу пропонують вакансії, щоб заманити жертв шпигунства. Фішингові атаки, спрямовані на людей, які шукають роботу, зростають. Будьте в курсі цієї тенденції та продовжуйте навчати своїх співробітників найновішим фішинговим кампаніям та тактикам.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Інструменти штучного інтелекту виконують реальну роботу в усьому бізнесі. Вони читають внутрішні документи, генерують результати та підключаються безпосередньо до систем, що зберігають конфіденційні дані. Такий доступ створює нові ризики для безпеки, особливо коли засоби контролю не відповідають тому, як ці інструменти використовуються. До 2028 року 25% корпоративних застосунків GenAI зіткнуться щонайменше з п'ятьма незначними інцидентами безпеки на рік, порівняно з 9% раніше. Водночас, Cyera виявила, що 83% підприємств вже використовують штучний інтелект, тоді як лише 13% мають чітке уявлення про те, як він взаємодіє з їхніми даними. Ця прогалина залишає конфіденційні дані без чіткого нагляду. У даному розділі розглядаються найкращі платформи безпеки на основі штучного інтелекту. Вона пояснює, що роблять ці інструменти, порівнює провідні варіанти та допомагає вам зробити вибір на основі ваших даних, використання штучного інтелекту та ризику.

Ключові висновки:

– Платформи безпеки на основі штучного інтелекту зосереджені на тому, як дані, моделі та користувачі взаємодіють у різних середовищах.

– Видимість даних та контроль доступу мають вирішальне значення для зниження ризиків, спричинених штучним інтелектом.

– Більшість рішень поєднують моніторинг, управління та виявлення загроз в одній платформі.

– Різні інструменти вирішують різні проблеми, від безпеки коду до захисту під час виконання.

Платформи безпеки на базі штучного інтелекту

Не всі платформи безпеки штучного інтелекту вирішують одну й ту саму проблему. Деякі зосереджені на захищенні коду та конвеєрів розробки, тоді як інші відстежують поведінку під час виконання, захищають кінцеві точки або контролюють, як системи штучного інтелекту отримують доступ до конфіденційних даних.

Для цього порівняння ми оцінили кожну платформу, використовуючи поєднання документації продукту, відгуків клієнтів та незалежних досліджень. Це включало такі джерела, як G2 та Gartner, а також матеріали постачальників та загальнодоступну технічну інформацію.

Ми зосередилися на тому, як кожна платформа обробляє дані, моніторинг активності штучного інтелекту, контроль доступу та виявлення загроз. Ми також розглянули інтеграцію з існуючими інструментами, підтримку гібридних середовищ та можливість масштабування зі зростанням використання штучного інтелекту. Платформи, які об'єднують дані, ідентифікацію та поведінку штучного інтелекту в єдине подання, отримали вищі рейтинги, оскільки вони допомагають командам переходити від виявлення до дій без додавання складності.

Платформи безпеки на основі штучного інтелекту допомагають організаціям контролювати, як системи штучного інтелекту отримують доступ до даних та використовують їх. Вони зосереджені на ризиках, пов'язаних з моделями, агентами, другими пілотами та даними, на які вони покладаються.

Ці платформи відстежують, де зберігаються конфіденційні дані та як інструменти штучного інтелекту взаємодіють з ними. Вони контролюють запити, виводи та потоки даних, щоб виявляти витоки, неправильне використання або небезпечну поведінку. Багато з них також застосовують політики в режимі реального часу, такі як блокування ризикованих запитів або обмеження доступу до певних даних.

Ключові характеристики сучасних платформ безпеки на базі штучного інтелекту

Сучасні платформи безпеки на основі штучного інтелекту створені для ризиків, для обробки яких не були розроблені старіші моделі безпеки. Традиційні інструменти зосереджені на мережах, кінцевих точках та відомих шаблонах атак. Штучний інтелект це змінює. Системи тепер отримують доступ до даних безпосередньо, обробляють ненадійні вхідні дані та виконують дії між інструментами без чітких меж:

– Видимість та класифікація даних: Визначає, де зберігаються конфіденційні дані в хмарі, SaaS та внутрішніх системах, і класифікує їх за контекстом, таким як власність, чутливість та ризик.

– Моніторинг активності ШІ та взаємодії з даними: відстежує запити, виводи та потоки даних між користувачами, агентами та моделями, щоб виявляти неправильне використання, витік або небезпечну поведінку.

– Управління доступом для ШІ: контролює, які інструменти ШІ, агенти та користувачі можуть отримувати доступ до певних даних, з політиками, пов'язаними з ідентифікатором, роллю та рівнем ризику.

– Захист від загроз та виконання: виявляє та блокує ризиковані дії в режимі реального часу, такі як оперативне впровадження, витік даних або аномальна поведінка штучного інтелекту.

– Управління безпекою даних (DSPM): виявляє та класифікує конфіденційні дані в різних середовищах, відображає, як до них здійснюється доступ та використовуються, а також забезпечує основу для управління доступом ШІ та зменшення ризиків у рамках DSPM.

Найкращі платформи безпеки на базі штучного інтелекту

Впровадження штучного інтелекту відбувається швидше, ніж більшість програм безпеки можуть встигати за цим. Ми вибрали наведені нижче платформи на основі того, наскільки добре вони забезпечують видимість даних та діяльності зі штучним інтелектом, забезпечують контроль доступу та знижують ризики, не уповільнюючи розробку чи операції.

Cyera

Найкраще рішення для підприємств, орієнтованих на дані, з багатохмарними робочими навантаженнями, керованими штучним інтелектом.

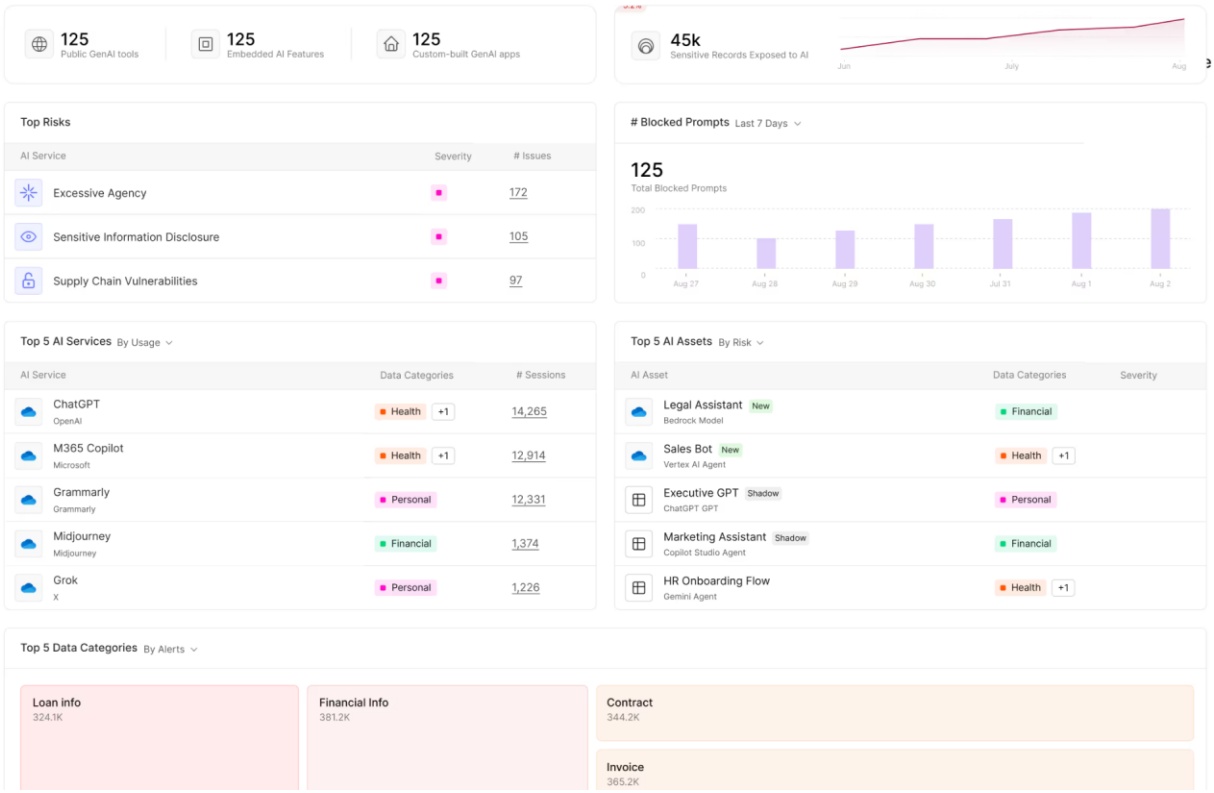

Рисунок 2.1 – Інтерфейс користувача Cyera

Cyera – це платформа безпеки даних на основі штучного інтелекту, створена для підприємств, які керують великими та складними середовищами даних. Вона зосереджується безпосередньо на даних, постійно ідентифікуючи та класифікуючи конфіденційну інформацію в хмарних, SaaS та локальних системах, а потім пов’язуючи ці дані з ідентифікаторами та шляхами доступу, щоб виявити реальний ризик. Це дає командам чітке уявлення про те, які дані існують, хто може до них отримати доступ і як зростає ризик в різних середовищах та робочих процесах зі штучним інтелектом.

Платформа об’єднує управління безпекою даних, безпеку штучного

інтелекту, управління доступом та виправлення в єдиній площині керування. Команди отримують швидкий доступ до конфіденційних даних і можуть вживати заходів, не покладаючись на ручні аудити чи фрагментовані інструменти. Вона працює в масштабі підприємства, скануючи великі середовища даних з високою точністю та відстежуючи, як здійснюється доступ до даних, як вони використовуються та як вони поширюються як між користувачами-людьми, так і системами штучного інтелекту.

Cyera також підтримує управління ризиками, пов'язаними зі штучним інтелектом. Вона виявляє тіньові інструменти штучного інтелекту, контролює надходження даних до систем штучного інтелекту та застосовує елементи керування для запобігання розкриттю конфіденційних даних у запитах та виводах. Її безагентне розгортання дозволяє організаціям швидко підключатися до середовищ та починати виявляти ризики без додаткової інфраструктури чи операційних витрат.

Ключові характеристики:

– DSPM на базі штучного інтелекту: ідентифікує та класифікує конфіденційні та власні дані в хмарних, SaaS та локальних середовищах, прив'язуючи ідентифікацію та контекст доступу до кожного набору даних.

– Безпека ШІ та засоби контролю AI-SPM: виявляє санкціоновані та тіньові інструменти ШІ, керує тим, як люди та агенти ШІ отримують доступ до даних, і зменшує ризик витоку даних у режимі реального часу.

– Механізм Omni DLP: Зменшує шум сповіщень, надає пріоритет значущим ризикам та рекомендує налаштовані політики, що зменшують кількість хибнопозитивних результатів.

– Видимість слідів доступу: відстежує, як користувачі та системи штучного інтелекту отримують доступ до конфіденційних даних, щоб підтримувати розслідування та готовність до аудиту.

– Безагентне розгортання: швидке розгортання без агентів та сканування великомасштабних середовищ з мінімальним впливом на експлуатацію.

Ціноутворення: Ціноутворення Cyera безпосередньо пов’язане з

результатами роботи клієнтів. Вони пропонують два комплексні плани для DSPM та DLP, кожен з яких побудований на їхній єдиній платформі безпеки зі штучним інтелектом.

Checkmarx

Найкращий варіант для команд, яким потрібне комплексне тестування безпеки додатків у конвеєрах DevSecOps.

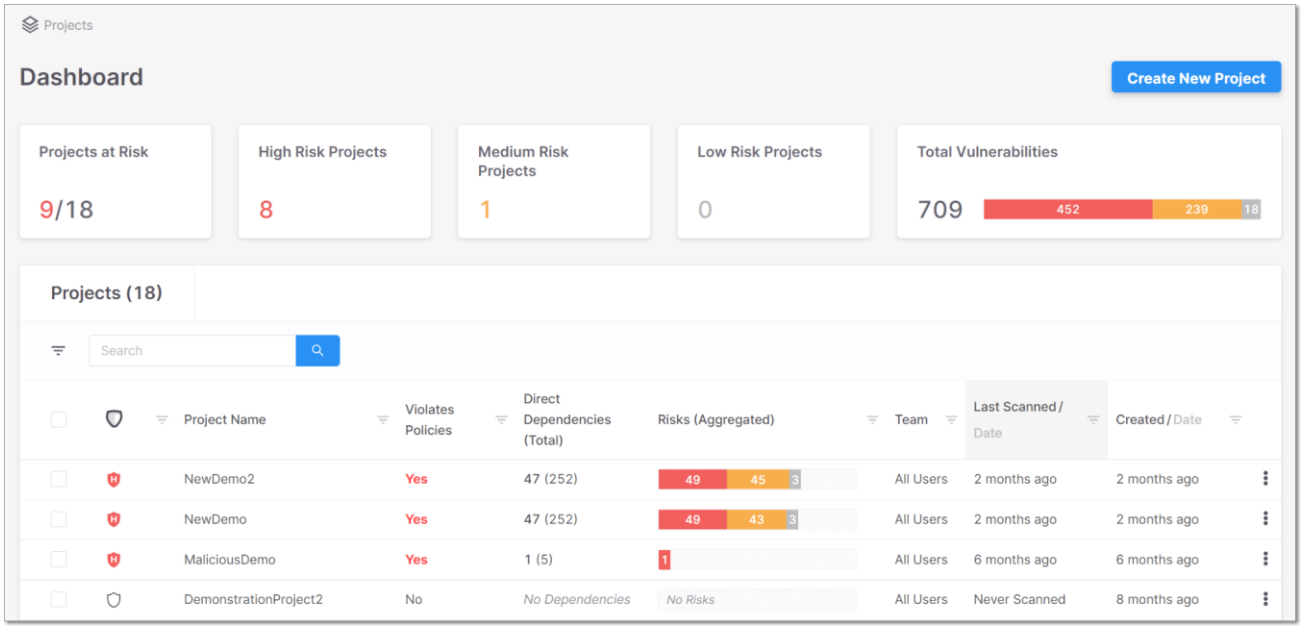

Рисунок 2.2 – Інтерфейс користувача Checkmarx

Checkmarx надає платформу безпеки додатків, яка захищає код, залежності та додатки, створені штучним інтелектом, протягом усього життєвого циклу розробки. Вона поєднує управління безпекою додатків з такими можливостями, як статичне та динамічне тестування, аналіз складу програмного забезпечення та безпека ланцюга поставок, надаючи командам видимість вразливостей у коді, конвеєрах та середовищах виконання.

Ключові характеристики:

– Керування станом безпеки застосунків: Виявляє та визначає пріоритети вразливостей у коді, залежностях та середовищах виконання.

– Аналіз коду, згенерованого штучним інтелектом: сканує та оцінює

ризики як у коді, написаному людиною, так і в коді, згенерованому штучним інтелектом.

– Статичне та динамічне тестування: поєднує SAST, DAST та аналіз складу програмного забезпечення для виявлення вразливостей протягом усього життєвого циклу розробки.

– Рекомендації щодо виправлення на основі штучного інтелекту: Надає контекстуальні пропозиції, які допомагають розробникам вирішувати проблеми в рамках їхнього робочого процесу.

Ціноутворення: Checkmarx пропонує різні тарифні плани, починаючи від базових і закінчуючи корпоративними.

Cisco AI Defense

Найкращий варіант для мережевих організацій, що забезпечують захист штучного інтелекту в усіх системах.

Cisco AI Defense – це платформа безпеки на основі штучного інтелекту, яка захищає програми, моделі та дані на основі штучного інтелекту в розподілених середовищах. Вона ідентифікує ресурси штучного інтелекту в хмарних середовищах та SaaS, контролює їх використання та виявляє такі ризики, як неправильні конфігурації, зловмисні атаки та витік даних. Cisco AI Defense застосовує політики та захисні механізми для контролю доступу, запобігання витоку конфіденційних даних та зменшення таких загроз, як оперативне впровадження та атаки типу «відмова в обслуговуванні» в режимі реального часу.

Ключові характеристики:

– Виявлення активів ШІ: Визначає моделі ШІ, програми, джерела даних та користувачів у різних середовищах.

– Виявлення ризиків, пов'язаних зі штучним інтелектом: виявляє вразливості, неправильні конфігурації та зловмисні загрози, спрямовані на системи штучного інтелекту.

– Захист під час виконання: Блокує впровадження запитів, витік даних та інші атаки під час виконання ШІ.

– Управління доступом: Контролює взаємодію користувачів і програм з інструментами та даними штучного інтелекту.

Ціноутворення: Ціни не вказані на веб-сайті.

CrowdStrike

Найкращий варіант для команд, орієнтованих на SOC, що розширює можливості виявлення загроз з урахуванням штучного інтелекту.

CrowdStrike надає платформу безпеки, яка захищає ідентифікаційні дані, хмарні робочі навантаження та середовища на основі штучного інтелекту за допомогою єдиної системи. Вона поєднує аналіз поведінки в режимі реального часу, захист ідентифікаційних даних, розвідку загроз та хмарну безпеку для виявлення та зупинки атак у різних доменах.

Ключові характеристики:

– Захист ідентифікаційних даних: моніторинг та захист людських та нелюдських ідентифікаційних даних у різних середовищах.

– Розвідка та полювання на загрози: відстежує активність зловмисника та підтримує розслідування активних загроз.

– Виявлення та реагування на основі штучного інтелекту: використовує штучний інтелект для аналізу поведінки, сортування сповіщень та автоматизації дій реагування.

Ціноутворення:

– Falcon Go: $59.99 на рік.

– Falcon Pro: $99.99 на рік.

– Falcon Enterprise: $184.99.

Darktrace

Найкращий варіант для команд, яким потрібен автономний штучний інтелект для виявлення загроз у режимі реального часу

Darktrace надає платформу кібербезпеки на основі штучного інтелекту, яка зосереджена на виявленні та реагуванні на загрози в мережевих, хмарних, електронних, ідентифікаційних та штучних середовищах. Вона використовує самонавчальний штучний інтелект для встановлення базової лінії нормальної

поведінки в системах та даних організації, а потім виявляє аномалії, які можуть свідчити про загрози, включаючи нові атаки та атаки, керовані штучним інтелектом.

Ключові характеристики:

– Виявлення загроз на основі поведінки: Виявляє аномалії, вивчаючи звичайні закономірності для користувачів, пристроїв та даних.

– Автономне реагування: вживає заходів для стримування загроз у режимі реального часу без ручного втручання.

– Міждоменна видимість: відстежує загрози в мережевих, хмарних, електронних, ідентифікаційних та штучних середовищах.

– Навчання за моделлю ШІ: Адаптується до середовища кожної організації для виявлення нових та постійно розвиваючихся загроз.

Ціноутворення: Ціни не вказані на веб-сайті.

GitHub Advanced Security

Найкраще для команд DevSecOps, які створюють безпечний код за допомогою штучного інтелекту

GitHub Advanced Security надає інструменти, які допомагають організаціям виявляти та виправляти вразливості, запобігати витокам секретних даних та керувати ризиками на рівні коду в різних репозиторіях. Він включає такі можливості, як сканування коду, аналіз залежностей та виявлення секретних даних, а також елементи керування, що блокують розкриття конфіденційних даних під час робочих процесів розробки.

Ключові характеристики:

– Сканування коду: Виявляє вразливості та проблеми з кодуванням за допомогою CodeQL або сторонніх інструментів.

– Секретне сканування: виявляє розкриті облікові дані, такі як ключі API та токени в репозиторіях.

– Захист від push-повідомлень: блокує коміти, що містять конфіденційні дані, перш ніж вони будуть додані.

– Перевірка залежностей: позначає вразливі залежності перед

об'єднанням змін коду.

Ціноутворення:

– Захист секрету GitHub: $19/місяць.

– Безпека коду GitHub: $30/місяць.

Microsoft Defender

Найкращий для середовищ Windows.

Microsoft Defender захищає пристрої, ідентифікаційні дані та дані в гібридних, багатохмарних та багатоплатформних середовищах. Він включає антивірусні засоби, засоби захисту від фішингу та можливості моніторингу загроз у режимі реального часу, які виявляють шкідливу активність і попереджають користувачів про потенційні ризики. Платформа підтримує захист кількох пристроїв і централізовану видимість, що дозволяє користувачам і командам відстежувати загрози, керувати налаштуваннями безпеки та реагувати на інциденти в різних системах з єдиного інтерфейсу.

Ключові характеристики:

– Сповіщення та інструкції щодо загроз: сповіщає користувачів про ризики та пропонує кроки для усунення загроз.

– Централізоване представлення інформації про безпеку: відображає стан пристрою, сповіщення та рекомендації в одному інтерфейсі.

– Захист, інтегрований у хмару: пов'язує безпеку пристрою з хмарними сервісами та обліковими записами користувачів.

Ціноутворення:

– Microsoft 365 персональний: $99.99/рік.

– Сімейство Microsoft 365: $129.99/рік.

– Преміум-підписка на Microsoft 365: 199,99 дол. США/рік.

Palo Alto Networks

Найкраще рішення для організацій, що впроваджують безпеку штучного інтелекту в широку хмарну платформу.

Palo Alto Networks – це платформа безпеки на основі штучного інтелекту, яка захищає мережі, хмарні середовища та системи штучного інтелекту за

допомогою єдиної архітектури. Вона виявляє та аналізує загрози в різних середовищах за допомогою інтегрованої телеметрії та застосовує автоматизоване виявлення та реагування для зменшення ризиків у режимі реального часу.

Ключові характеристики:

– Виявлення загроз на основі штучного інтелекту: Виявляє та аналізує загрози в мережі, хмарі та системах штучного інтелекту.

– Уніфікована платформа безпеки: об'єднує кілька функцій безпеки в рамках однієї архітектури.

– Захист хмари та мережі: Захищає інфраструктуру в гібридних та багатохмарних середовищах.

– Безпека застосунків на основі штучного інтелекту: захищає моделі, агентів та дані штучного інтелекту під час розробки та виконання.

Ціноутворення: Ціни не вказані на веб-сайті.

Splunk AI

Найкращий варіант для команд SOC та DevOps, які керують масштабними журналами для полювання на загрози в режимі реального часу

Splunk надає платформу на основі штучного інтелекту, яка аналізує машинні дані в середовищах безпеки, IT та спостереження для виявлення, розслідування та реагування на інциденти. Вона збирає та співвідносить дані з різних джерел, застосовує машинне навчання та моделі штучного інтелекту для виявлення закономірностей та аномалій, а також підтримує автоматизовані робочі процеси для реагування та усунення наслідків.

Ключові характеристики:

– Отримання та аналіз даних: Збирає та аналізує дані із середовищ безпеки, IT та прикладних систем.

– Виявлення на основі штучного інтелекту: Виявляє аномалії та загрози за допомогою машинного навчання та поведінкового аналізу.

– Запит природною мовою: дозволяє користувачам взаємодіяти з даними та генерувати аналітичні висновки за допомогою помічників штучного

інтелекту.

– Автоматизовані робочі процеси реагування: Запускає дії та кроки виправлення на основі виявлених подій.

Ціноутворення: Ціни не вказані на веб-сайті, але варіюються від цін на робоче навантаження до цін на основі активності

SentinelOne

Найкращий варіант для середовищ XDR на основі штучного інтелекту.

SentinelOne моніторить хмарні середовища, ідентифікаційні дані та активність штучного інтелекту для виявлення загроз та реагування на них. Він аналізує поведінку в різних системах для виявлення аномалій, співвідносить активність між доменами та автоматизує розслідування та реагування. Об'єднуючи ці можливості, платформа надає командам безпеки глибокий контекст та практичну інформацію про складні шляхи атак.

Ключові характеристики:

– Виявлення загроз на основі поведінки: Виявляє аномалії в хмарі та ідентифікаторах за допомогою поведінкового аналізу.

– Автономні дії реагування: автоматично стримує та пом'якшує загрози без ручного втручання.

– Міждоменна видимість: відстежує активність у хмарних середовищах, середовищах ідентифікації та штучного інтелекту.

– Аналітика на основі штучного інтелекту: аналізує великі обсяги даних безпеки для виявлення закономірностей та визначення пріоритетів ризиків.

Ціноутворення:

– Singularity Complete (базова безпека штучного інтелекту): $179.99.

– Singularity Commercial (Базовий захист штучного інтелекту): $229.99.

– Singularity Enterprise (Комплексна безпека на основі штучного інтелекту): Зверніться до відділу продажів.

Stellar Cyber

Найкращий варіант для консолідованих розгортань SIEM та Open XDR.

Stellar Cyber – це платформа SecOps на основі штучного інтелекту, яка

об’єднує дані безпеки з різних інструментів, середовищ та джерел даних в єдину систему для виявлення, розслідування та реагування. Вона поєднує SIEM, мережеве виявлення та реагування, аналітику поведінки користувачів та об’єктів, а також розширене виявлення та реагування в одній платформі

Ключові характеристики:

– Уніфікована платформа даних безпеки: агрегує телеметрію з хмари, мережі та сторонніх інструментів.

– Виявлення загроз на основі штучного інтелекту: Виявляє аномалії та співвідносить сигнали з кількох джерел даних.

– Автоматизоване сортування та реагування: Пріоритетність сповіщень та ініціювання дій реагування на основі ризику.

– Відкрита архітектура XDR: інтегрується з існуючими інструментами EDR та безпеки без заміни.

Ціноутворення: Ціни не вказані на веб-сайті.

Wiz

Найкраще для хмарних підприємств.

Wiz надає хмарну платформу безпеки, яка об'єднує код, хмарну інфраструктуру, ідентифікаційні дані та середовище виконання в єдиний граф безпеки. Вона ідентифікує ризики в різних середовищах, відображаючи зв'язки між вразливостями, неправильними конфігураціями, ідентифікаційними даними та ризиками вразливостей, надаючи командам єдине уявлення про шляхи атак та прогалини в безпеці. Платформа підтримує безперервний аналіз ризиків, їх виявлення та реагування в хмарних та штучно-інтелектуальних середовищах, допомагаючи командам визначати пріоритети та усувати проблеми на основі контексту та ризиків.

Ключові характеристики:

– Управління безпекою хмарних середовищ: Виявляє неправильні конфігурації, вразливості та ризики викриття в хмарних середовищах.

– Аналіз графів безпеки: відображає зв'язки між кодом, хмарними ресурсами, ідентифікаторами та даними з поверхневими шляхами атаки.

– Безагентне розгортання: підключається до хмарних середовищ без необхідності збору даних агентами.

– Пріоритетність ризиків: ранжує проблеми на основі можливості їх використання, вразливості та впливу.

Ціноутворення: Ціни не вказані на веб-сайті.

Як вибрати правильну платформу безпеки на базі штучного інтелекту

Почніть зі свого середовища, а не з інструменту. Платформи безпеки на основі штучного інтелекту різняться за тим, що вони охоплюють, тому перед порівнянням варіантів проаналізуйте свої ризики, місця розташування даних та існуючі засоби контролю. Зосередьтеся на тому, як кожна платформа ідентифікує, відстежує та контролює дані та активність штучного інтелекту в усьому вашому стеку.

– Оцініть свої основні зони ризику: визначте, де конфіденційні дані та використання штучного інтелекту створюють ризики, включаючи тіньовий штучний інтелект, ризики надмірного доступу та витоку даних. Наприклад, якщо співробітники ненавмисно діляться конфіденційними даними через підказки штучного інтелекту, вам потрібна платформа, яка відстежує взаємодію в режимі реального часу та позначає або блокує конфіденційні дані, перш ніж вони залишать ваше середовище.

– Оцінка видимості даних: переконайтеся, що платформа виявляє та класифікує дані в хмарних, SaaS, локальних та штучних системах з чітким контекстом доступу та використання. Наприклад, якщо конфіденційні дані зберігаються в SaaS-додатку, такому як CRM, ви повинні мати змогу бачити, які користувачі та інструменти штучного інтелекту мають до них доступ і як часто до них надходять запити.

– Пріоритетність інтеграції та масштабованості: перевірте підтримку існуючих інструментів, таких як SIEM, IAM, SOAR та DLP , і підтвердьте, що вони масштабуються залежно від обсягу даних та впровадження штучного інтелекту. Наприклад, виявлення ризику має запустити робочі процеси у вашому існуючому стеку, такі як відкриття заявки, сповіщення SOC або автоматичне

скасування доступу.

– Перевірка управління та постійної відповідності: Перегляньте, як платформа застосовує політики, відстежує доступ та створює готові до аудиту записи в різних середовищах. Наприклад, якщо регулятор запитує, хто отримав доступ до конфіденційних даних за допомогою інструментів штучного інтелекту, ви повинні мати змогу створити чіткий журнал аудиту з позначками часу, ідентифікаторами та вжитими діями.

Чому центральна на даних безпека штучного інтелекту стає стандартом

Інструменти, агенти та API штучного інтелекту взаємодіють з даними в хмарних, SaaS та внутрішніх середовищах, часто поза традиційним контролем. Підхід, орієнтований на дані, забезпечує важливий рівень безпеки, гарантуючи, що моделі штучного інтелекту отримують доступ лише до тих даних, які їм дозволено використовувати, і що конфіденційна інформація не розкривається через взаємодію зі штучним інтелектом.

– Ризик, пов'язаний зі штучним інтелектом, виникає на рівні даних: конфіденційні дані живлять моделі, запитують та виводять дані, тому вплив відбувається через доступ, переміщення та неправильне використання, а не лише через інфраструктуру..

– Системи штучного інтелекту обходять традиційні засоби контролю безпеки: агенти, інші пілоти та API взаємодіють з даними безпосередньо, що знижує ефективність захисту периметра та кінцевих точок.

– Вимоги до нормативних актів та довіри зростають: командам потрібні чіткі записи про те, де знаходяться конфіденційні дані, хто до них мав доступ і як вони використовуються в робочих процесах штучного інтелекту.

– Швидша, автоматизована безпека зараз є важливою: ручні аудити не можуть встигати за динамічними середовищами даних, тому платформи, які відстежують активність даних та запускають робочі процеси виправлення, допомагають зменшити ризики в міру їх виникнення.

Захистіть корпоративні системи за допомогою найкращих платформ безпеки на базі штучного інтелекту

Впровадження штучного інтелекту не сповільнюється, як і пов'язані з

ним ризики безпеки. Організації, які залишаються попереду, – це ті, які не чекають на порушення, щоб дізнатися, де зберігаються їхні конфіденційні дані, хто може отримати до них доступ і як їхні системи штучного інтелекту їх використовують. Правильна платформа безпеки на основі штучного інтелекту забезпечує вашій команді прозорість, забезпечує масовий контроль та автоматизує роботу, яка є занадто великою для ручного виконання. Йдеться не лише про запобігання інцидентам. Йдеться про створення основи безпеки, яка відповідає тому, як насправді працює ваш бізнес. Cyera створена саме для цього. Її безагентна платформа, що орієнтована на дані, забезпечує швидке розгортання, точну класифікацію, аналіз ризиків з урахуванням ідентифікації та автоматизоване виправлення в хмарних, SaaS та гібридних середовищах.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки програмних компонентів системи поведінкового аналізу та контролю активності додатків було обрано мову програмування Python. Вибір цього інструментарію зумовлений його гнучкістю, простотою синтаксису, наявністю ефективних структур даних високого рівня та підтримкою об'єктно-орієнтованої парадигми. Динамічна типізація, інтерпретований характер та лаконічність Python забезпечують високу швидкість прототипування і розробки складних алгоритмів моніторингу. Офіційний інтерпретатор та стандартна бібліотека платформи Python є відкритими, кросплатформовими та доступними для використання в будь-яких дослідницьких і прикладних проєктах. Важливою перевагою Python у контексті аналізу низькорівневої активності є можливість легкого розширення його функціонала за допомогою модулів, написаних на мовах C або C++. Це дозволяє ефективно поєднувати високу швидкість виконання критичних до продуктивності операцій перехоплення системних викликів із гнучкістю та зручністю керування логікою поведінкового аналізу на рівні мови високого рівня.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Фінансовий директор отримує голосове повідомлення, яке звучить точно так само, як і запит головного виконавчого директора: дозвольте терміновий переказ коштів, перш ніж конфіденційне придбання стане публічним. Запит надходить із правильною терміновістю, правильною лексикою та знайомим голосом. Це практична реальність тенденцій поведінкової кібербезпеки 2026 року. Найбільш серйозні атаки не залежатимуть виключно від зламу технологій. Вони використовуватимуть увагу, авторитет, звичку, втому та людську потребу бути корисним.

Для керівників з безпеки, слідчих, фахівців з дотримання вимог та фахівців з поведінки центральне питання змінюється. Йдеться вже не просто про те, чи можуть співробітники розпізнати підозрілий електронний лист. Йдеться про те, чи розуміє організація умови, які змушують хороших людей приймати ризиковані рішення під тиском.

Тенденції поведінкової кібербезпеки 2026: Поверхня людської атаки

Кібербезпека давно визнає людей потенційною точкою невдачі. Таке формулювання є неповним і часом контрпродуктивним. Працівники – це не слабкі ланки, які потрібно виправляти за допомогою щорічного навчання. Вони є особами, що приймають рішення, які працюють у складних середовищах, часто з неповною інформацією, конкуруючими пріоритетами та дедалі переконливішими цифровими сигналами.

У 2026 році поведінкова кібербезпека буде більше зосереджена на моментах перед дією: чому хтось приймає запит багатофакторної автентифікації, ділиться документом, змінює платіжні реквізити або обходить контроль. Цей вибір формується як соціальним контекстом, так і технічними знаннями.

Найбільш зрілі програми досліджуватимуть закономірності, а не окремі

помилки. Кластер ризикованих дій може відображати погано розроблений процес затвердження, заплутану мову безпеки, надмірний обсяг сповіщень або культуру, яка винагороджує швидкість, а не перевірку. Ця відмінність має значення. Якщо організації реагують лише звинувачуючи окремих осіб, вони зберігають умови, які зробили інцидент ймовірним.

Соціальна інженерія стане більш персоналізованою

Генеративний штучний інтелект знизив вартість створення переконливого, індивідуально підібраного обману. Зловмисники тепер можуть досліджувати публічні записи, професійні профілі, корпоративні оголошення та витік даних, щоб створювати повідомлення, що відображають роль, стосунки, поточні проекти та мовні моделі цілі. Синтетичне аудіо та відео додають ще один рівень достовірності шахрайству, пов'язаному з видаванням себе за керівника, та шахрайству з боку постачальників.

Визначальною рисою цих атак є не просто технічна складність. Це емоційна точність. Повідомлення може викликати страх пропустити дедлайн, лояльність до старшого керівника, занепокоєння за колегу або тривогу щодо потенційного дисциплінарного порушення. Зловмисник прагне звузити мислення цілі доти, доки перевірка не стане незручною або нелояльною.

Організації повинні реагувати, зробивши перевірку нормальною професійною поведінкою, особливо для змін платежів, запитів на отримання облікових даних, привілейованого доступу та незвичайних інструкцій керівництва. Другий канал підтвердження корисний лише тоді, коли працівники мають повноваження, час та психологічний дозвіл на його використання. Політика, яка існує на папері, але суперечить ієрархії на робочому місці, зазнає невдачі у критичний момент.

Безпека ідентифікації залежатиме від поведінкових сигналів

Паролі, токени та біометричні маркери залишаються важливими, але вони не відображають повної історії ідентифікації. Дійсний обліковий запис користувача може бути контрольований зловмисником. У відповідь організації розширюють поведінкову аналітику, щоб оцінити, чи нагадує сеанс звичайну

діяльність законного користувача.

Сигнали можуть включати час і послідовність дій, типові програми, ритм друку, контекст пристрою, шаблони доступу до даних або незвичайні зміни в поведінці транзакцій. За умови ретельного використання ці індикатори можуть підтримувати адаптивну автентифікацію та спонукати до додаткової перевірки, коли ризик зростає.

Компроміс є суттєвим. Поведінкові дані можуть бути конфіденційними, недосконалими та схильними до неправильного тлумачення. Відряджений працівник, аналітик нічної зміни або працівник, який використовує допоміжні технології, може здаватися аномальним, не будучи зловмисним. Служби безпеки повинні уникати трактування ймовірності як доказу.

Ефективне управління вимагає прозорої політики, обмеженого збору даних, чітких правил зберігання та перевірки людьми рішень, що приймаються в результаті. Поведінкові сигнали повинні зменшувати невизначеність, а не перетворюватися на прихований механізм карального спостереження. Це особливо важливо в багатонаціональних організаціях, де очікування щодо конфіденційності та правові зобов'язання відрізняються в різних юрисдикціях.

Програми управління внутрішніми ризиками вийдуть за межі підозр

Внутрішній ризик часто обговорюється так, ніби кожен працівник являє собою приховану загрозу. Такий підхід підриває довіру та може зменшити ймовірність повідомлення про порушення. У 2026 році сильніші програми розрізнятимуть зловмисні наміри, недбалу поведінку, скомпрометовані облікові записи та організаційні умови, які створюють ризик, якого можна уникнути.

Наприклад, багаторазове використання несанкціонованих інструментів для обміну файлами може свідчити про спробу обійти контроль. Це також може виявити, що схвалені системи занадто повільні або недоступні для необхідної роботи. Слідчі дії повинні бути пропорційними та заснованими на доказах, поєднуючи цифрові показники з розумінням очікувань щодо ролі, робочого навантаження, потреб у доступі та організаційної культури.

Поведінкова наука особливо цінна тут, оскільки вона протистоїть

спрощеним поясненням. Люди можуть бути відданими організації та все одно шукати спрощених шляхів. Вони можуть перебувати у фінансовому скрутному становищі, не створюючи внутрішньої загрози. Вони можуть приймати незвичайне рішення з цілком законної причини. Дослідникам потрібні гіпотези, а не припущення.

Культура безпеки вимірюватиметься поведінкою, а не рівнем виконання

Легко повідомити про рівень завершення модулів підвищення обізнаності, але це поганий показник готовності. Людина може пройти кожен призначений курс і все одно схвалити шахрайський запит під час напруженої зустрічі. Більш корисне питання полягає в тому, чи має місце безпечна поведінка, коли вона незручна, неоднозначна або соціально складна.

Це сприятиме ширшому використанню сценарного навчання, симуляцій та вправ для конкретних ролей. Фінансовим командам потрібна практика у сфері шахрайства з рахунками-фактурами та платежами. Фахівцям з управління персоналом необхідно розпізнавати запити на передачу даних на основі ідентифікаційних даних та конфіденційні дані. Технічним адміністраторам потрібні відрепетировані процедури для аномалій доступу та екстрених змін. Універсальний модуль фішингу не може підготувати кожну роль до її найімовірнішої точки прийняття рішення.

Поведінка керівництва також помітна. Якщо керівники обходять контроль, вимагають негайних дій без перевірки або ставляться до секретних питань як до перешкоди, співробітники зрозуміють, що дотримання вимог необов'язкове, коли йдеться про статус. І навпаки, керівники, які відкрито перевіряють неочікувані запити та дякують персоналу за ескалацію занепокоєння, створюють культуру, де обережність має професійну легітимність.

Тенденції поведінкової кібербезпеки 2026 року підвищать когнітивну стійкість

Когнітивна стійкість – це здатність приймати обґрунтовані рішення в умовах терміновості, невизначеності, відволікання уваги та маніпуляцій. Це не фіксована риса особистості. Організації можуть розробляти робочі процеси, які підтримують її або підривають.

Втома від попереджень – яскравий приклад. Коли працівники отримують забагато попереджень, їхня увага виснажується, і вони починають ігнорувати навіть законні підказки. Відповіддю не завжди є більше навчання. Це може бути менше сповіщень, які видаються краще вчасно; чіткіша мова; розумніші шляхи ескалації; та системи, які роблять безпечні дії легшими, ніж небезпечні скорочення.

Команди безпеки також звертатимуть пильнішу увагу на поведінкові наслідки постійних змін. Нові інструменти, реорганізації, домовленості про віддалену роботу та зміни правил доступу можуть створювати плутанину, якою можуть скористатися зловмисники. У перехідні періоди люди частіше погоджуються з незнайомими процесами або підкоряються очевидному авторитету. Тому плани управління змінами повинні включати цільову комунікацію щодо безпеки, а не лише технічні інструкції з розгортання.

Безпека штучного інтелекту вимагає розсудливості, а не повної заборони

Оскільки співробітники використовують інструменти штучного інтелекту для узагальнення документів, складання чернеток повідомлень, аналізу даних та написання коду, організації стикаються з неминучим викликом управління. Загальні заборони можуть зменшити безпосередній вплив, але вони також можуть заохочувати несанкціоноване використання. Необмежене впровадження створює інший набір ризиків, включаючи розголошення конфіденційних даних, небезпечний код, маніпулювання результатами та надмірну залежність від неточної інформації.

Практичний підхід полягає у визначенні затверджених варіантів

використання, встановленні меж обробки даних та навчанні персоналу, як перевіряти результати, згенеровані штучним інтелектом. Працівники повинні розуміти, що плавна відповідь не є доказом точності, авторизації чи безпеки. Це питання поведінки, а також технічне: люди схильні довіряти системам, які виглядають впевнено, особливо коли ці системи економлять час.

Фахівці з безпеки також повинні бути готові до зловмисного контенту зі штучним інтелектом, який є правдоподібним, але не ідеальним. Зловмисникам не потрібен бездоганний діпфейк чи повідомлення. Їм потрібна ціль, яка достатньо зайнята, достатньо довірлива або під тиском, щоб діяти, перш ніж перевіряти.

Підготовка фахівців до наступного етапу кіберризиків

На наступному етапі розвитку кібербезпеки будуть винагороджені фахівці, які зможуть пов’язати технічні докази з поведінковим контекстом. Це включає розпізнавання тактик переконання , проведення інтерв’ю без упередженого підтвердження, відповідальне тлумачення аномалій та розробку засобів контролю, що відповідають тому, як люди насправді працюють.

Цей досвід матиме значення в правоохоронних органах, розслідуванні шахрайства, корпоративній безпеці, комплаєнсі, управлінні персоналом та державній політиці. Він також вимагає скромності. Поведінкові показники можуть виявити ризик, але вони не можуть замінити належну правову процедуру, етичне судження або ретельно перевірені докази.

Організації, найкраще підготовлені до 2026 року, не вимагатимуть від працівників досконалості. Вони створять середовище, де паузи, запитання та перевірка є практичним вибором, за який людей поважатимуть.

3.2 Розробка структурної схеми

Виявлення на основі сигнатур: добре для початківців

Традиційні антивірусні рішення покладаються на виявлення на основі сигнатур, яке чудово працює доти, доки не перестане функціонувати.

Зловмисники можуть легко змінювати код шкідливого програмного забезпечення: змінити кілька рядків, перекомпілювати, і раптом ця «відома загроза» стає невидимою. Це як надіти штучні вуса та сподіватися обдурити розпізнавання обличчя.

Ситуація погіршується з безфайловими атаками. Ці загрози працюють виключно в пам'яті, запускаючи шкідливий код через легітимні системні інструменти, такі як PowerShell або інструментарій керування Windows. Не існує «поганого файлу» для сканування, оскільки шкідливе програмне забезпечення запозичує облікові дані вашої системи та проникає крізь неї.

Антивіруси наступного покоління з'явилися для вирішення цих проблем. У той час як традиційні антивіруси запитують: «Чи бачив я цю конкретну загрозу раніше?», рішення наступного покоління запитують: «Чи схожа ця поведінка на поведінку шкідливого програмного забезпечення?» Різниця має значення, коли зловмисники видозмінюють шкідливе програмне забезпечення швидше, ніж ви встигаєте оновити визначення.

Функції та можливості

Платформи антивірусного захисту та захисту кінцевих точок наступного покоління використовують штучний інтелект та моніторинг поведінки для виявлення загроз. Керовані платформи захисту кінцевих точок додають перевірку за участю людини , щоб зупинити загрози, перш ніж вони зможуть вплинути на ваші системи. Huntress Managed Antivirus пропонує цей керований людиною підхід до антивірусних програм наступного покоління, особливо для організацій, які використовують Microsoft Defender, поєднуючи автоматичне виявлення з цілодобовою перевіркою SOC, щоб виявити те, чого не можуть виявити автоматизовані інструменти.

Технології зустрічаються з людським досвідом

Найефективніші системи виявлення шкідливого програмного забезпечення на основі штучного інтелекту не покладаються виключно на алгоритми. Вони поєднують машинне навчання з аналітиками безпеки, які можуть виявити контекст, який пропускають машини.

Моделі штучного інтелекту та машинного навчання

Сучасні платформи антивірусного захисту та захисту кінцевих точок наступного покоління використовують машинне навчання для виявлення підозрілих закономірностей без необхідності попереднього спостереження за конкретною загрозою. Навчання на мільйонах зразків навчає ці моделі розпізнавати ДНК шкідливої поведінки, а не запам'ятовувати конкретні сигнатури.

Платформи захисту від шкідливого програмного забезпечення наступного покоління часто є хмарними, що дозволяє швидке розгортання на кінцевих точках за лічені години, а не місяці, одночасно усуваючи навантаження на обслуговування, таке як постійне оновлення сигнатур. Це не лише зменшує накладні витрати на ІТ, але й гарантує, що всі кінцеві точки залишаються в курсі останніх даних про загрози.

Штучний інтелект аналізує сотні атрибутів: як було створено файл, звідки він взявся, які процеси він запускає та які мережеві підключення він намагається встановити.

Поведінковий моніторинг у дії

Системи захисту кінцевих точок від шкідливого програмного забезпечення відстежують поведінку, яка вказує на атаку:

– Масове шифрування: Програми-вимагачі швидко шифрують файли. Звичайні програми рідко отримують доступ до сотень файлів за лічені секунди, що робить це серйозним тривожним сигналом.

– Спроби ескалації привілеїв: шкідливе програмне забезпечення часто вимагає прав адміністратора, щоб максимізувати шкоду. Інструменти наступного покоління відстежують ці запити та оцінюють, чи мають вони сенс.

– Незвичне виконання скрипта: несподіваний запуск PowerShell або запуск з дивних батьківських процесів є підозрілим.

– Швидка модифікація файлів: Шкідливе програмне забезпечення зазвичай модифікує, шифрує або видаляє файли зі швидкістю, до якої звичайні користувачі ніколи не досягнуть. Поведінкові системи можуть зупинити це до

того, як виникне серйозна шкода.

Поведінковий аналіз виявляє загрози незалежно від того, як вони упаковані. Зловмисник може змінити пакувальний папір, але якщо вміст все ще поводиться як шкідливе програмне забезпечення, його позначають.

Засоби зменшення експлойтів та індикатори атаки (IoA)

Сучасне антивірусне програмне забезпечення наступного покоління також відстежує IoA на кінцевих точках, щоб виявляти ранні ознаки компрометації. Трохи схоже на сусідське спостереження за вашою мережею, де кожна підозріла дія, така як шахрайський скрипт або підвищення привілеїв, реєструється та співвідноситься з активністю в інших місцях. Пов'язуючи ці точки, система може виявляти скоординовані атаки або горизонтальні переміщення, зупиняючи загрози до їх повного виконання. Це доповнює моніторинг поведінки, забезпечуючи як спостереження за поведінкою, так і розумні механізми для виявлення закономірностей у великих масштабах.

Керований антивірус: Відсутній елемент

Антивірусна технологія наступного покоління є потужною, але без людського нагляду хибнопозитивні результати накопичуються швидко, а реальні загрози можуть прослизнути крізь автоматичні відповіді. Керований антивірус Huntress поєднує цю прогалину, поєднуючи можливості Microsoft Defender наступного покоління з аналітиками SOC, які цілодобово перевіряють кожне сповіщення. Бізнес отримує антивірусний та EDR-захист корпоративного рівня без необхідності створювати внутрішню команду безпеки.

Можливості наступного покоління в дії

Антивірусне програмне забезпечення наступного покоління відстежує підозрілу поведінку та зупиняє атаки до їхньої ескалації.

Менше пропущених загроз

Виявлення на основі сигнатур є бінарним: або воно бачить загрозу, або ні. Антивірусне програмне забезпечення наступного покоління призначає оцінки ризику поведінці та співвідносить кілька низькорівневих індикаторів у високодостовірні сповіщення. Це дозволяє йому виявляти поліморфне шкідливе

програмне забезпечення, експлойти нульового дня та складні атаки, які пропускають традиційні інструменти. Малий та середній бізнес без центрів операцій безпеки тепер може зупиняти загрози, з якими вони ніколи раніше не стикалися.

Зменшення втоми при пильному спостереженні

Хоча поведінкових сповіщень більше, ніж у традиційних антивірусних систем, штучний інтелект фільтрує та розумно визначає пріоритети. Служби безпеки зосереджуються на справді підозрілій активності, тоді як автоматизовані реагування обробляють події з меншим ризиком, полегшуючи навантаження на вже перевантажені IT-команди.

Виявлення наступного покоління з перевіркою під керівництвом людини

Навіть найскладніший штучний інтелект можна обдурити або генерувати хибнопозитивні результати. Саме в цьому сяє Huntress – чи то завдяки керованому Microsoft Defender для комплексного антивірусного та EDR-захисту, чи то завдяки керованому EDR для розширеного пошуку загроз. Huntress поєднує поведінкове виявлення наступного покоління з цілодобовими аналітиками-людьми , які перевіряють загрози, перш ніж вони досягнуть вашої команди. Ви отримуєте автоматизовану швидкість і масштабування, а також людську оцінку, що дозволяє відрізняти справжні атаки від крайніх аномалій.

Платформа відстежує випадки шифрування, підвищення привілеїв та підозрілі скрипти, досліджуючи та часто усуваючи проблеми ще до того, як ви дізнаєтеся про їхню наявність. Це захист кінцевих точок, який не просто виявляє, а й захищає.

Структурна схема системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів, у вигляді антивірусного комплексу показана на рисунку 3.1.

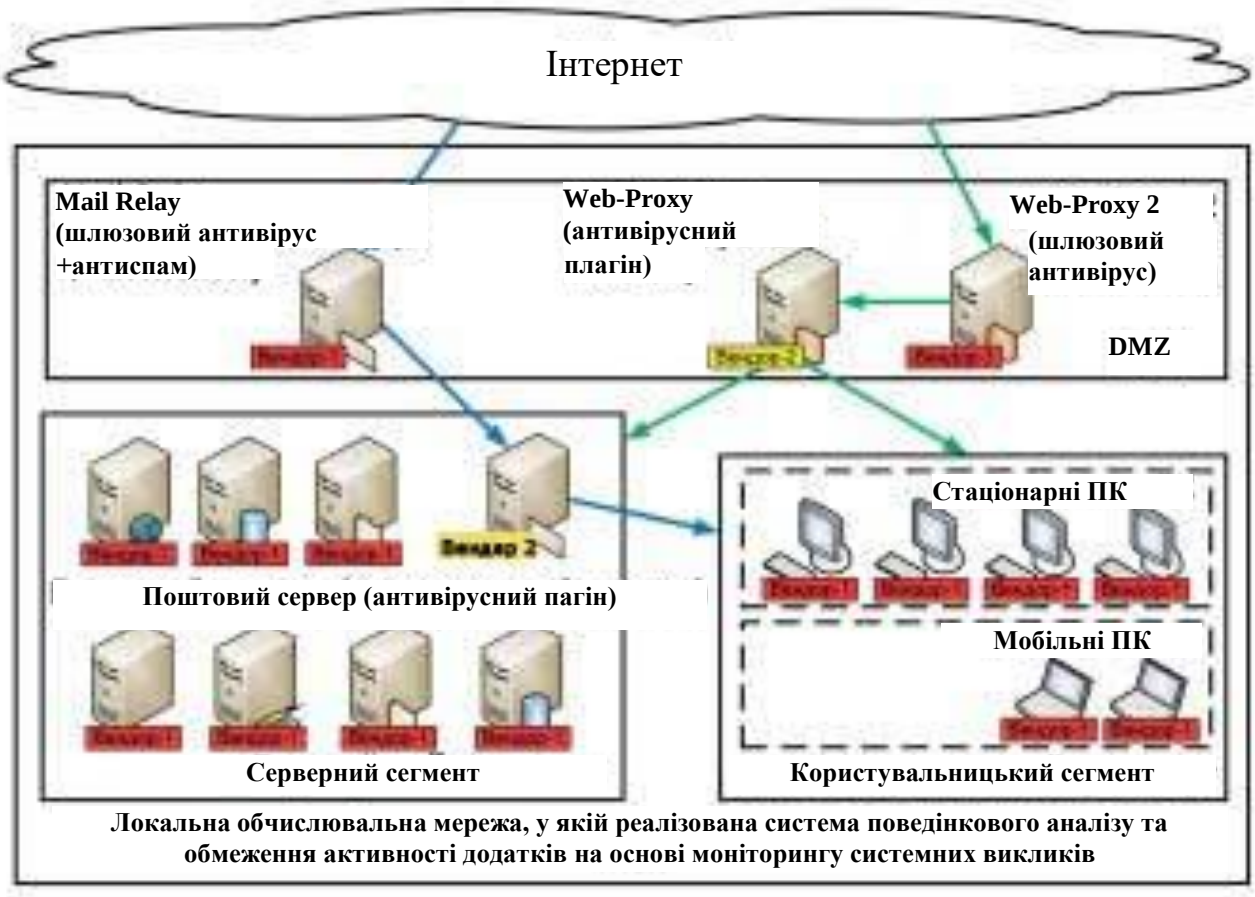

Рисунок 3.1 – Структурна схема системи

Створення комплексної системи антивірусного захисту – це багатоетапний процес, який зазвичай охоплює такі кроки:

– Аудит безпеки: збір вихідних даних для розробки плану впровадження.

– Формування вимог: підготовка технічного завдання для майбутньої системи захисту мережі.

– Проєктування: розробка технороботочого проєкту з описом архітектури, схем встановлення та налаштувань.

– Навчання: підготовка персоналу, який відповідатиме за адміністрування системи.

– Пусконалагодження та розгортання: тестування в пілотній зоні з наступним повномасштабним впровадженням у мережу.

– Технічна підтримка: подальше обслуговування системи в процесі її експлуатації.

Тривалість і склад цих етапів залежать від масштабу та розмірів мережі, що захищається. Усі роботи з розгортання та обслуговування систем поведінкового аналізу й моніторингу активності додатків підприємство може виконувати самостійно або за допомогою спеціалізованих компаній з інформаційної безпеки.

3.3 Розробка функціональної схеми

Розроблений у межах магістерської роботи програмний комплекс поведінкового аналізу та обмеження активності додатків функціонує як спеціалізована антивірусна система, призначена для захисту комп'ютерної системи від деструктивного програмного забезпечення шляхом постійного моніторингу системних викликів і контролю поведінки процесів. Архітектура комплексу базується на сучасних методах проактивного виявлення загроз та забезпечує стабільне функціонування операційної середовища з мінімальними накладними витратами на обчислювальні ресурси.

Основні функції розробленої системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів:

– Моніторинг і захист у реальному часі: безперервний перехоплення та аналіз системних викликів, завдяки чому виявляється підозріла активність виконуваних додатків до моменту завдання збитків ОС.

– Проактивний поведінковий аналіз: евристичний контроль динамічної поведінки програм, що дозволяє виявляти нові або невідомі шкідливі коди на основі аномальних послідовностей системних викликів.

– Миттєве блокування загроз: автоматичне обмеження активності, ізоляція або завершення процесів, чия поведінка відповідає сигнатурам атак або деструктивних шаблонів.

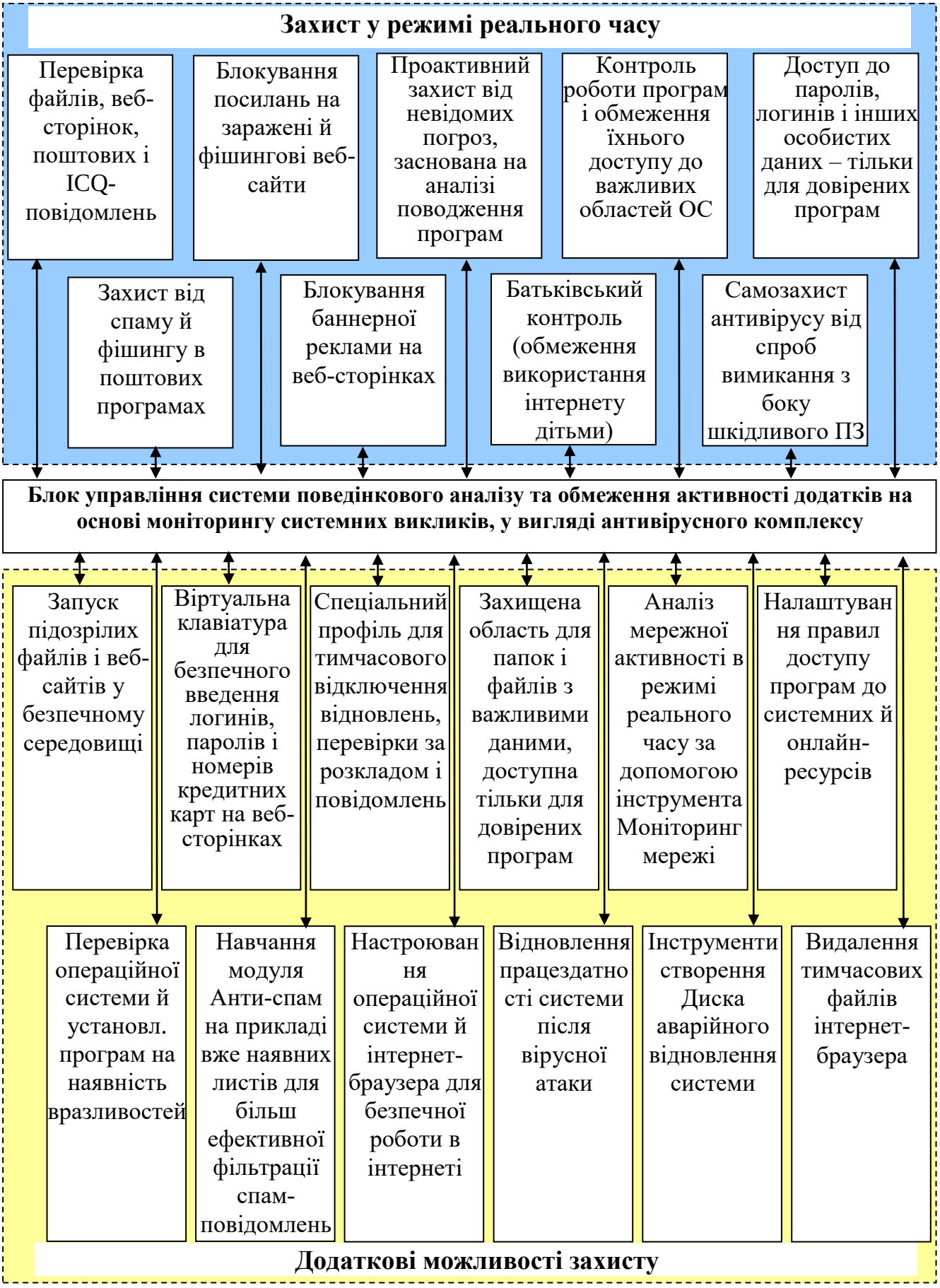

Рисунок 3.2 – Функціональна схема системи

– Захист мережевої активності: перевірка трафіку, блокування перенаправлень на фішингові та заражені вебсайти, а також захист при роботі з поштовими клієнтами.

– Самозахист антивірусного ядра: вбудовані механізми захисту власних процесів та служб від несанкціонованого завершення, модифікації або вимкнення з боку зловмисного ПЗ.

– Безпека робочого середовища: наявність інструментів віртуальної клавіатури для захисту конфіденційних даних під час їх введення, сканування операційної системи й встановленого софту на наявність уразливостей.

– Відновлення системи: функціонал усунення наслідків вірусних атак, очищення тимчасових файлів та відновлення пошкоджених параметрів конфігурації ОС і веббраузерів.

На рисунку 3.2 зображена функціональна схема системи. Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Процесну діаграму розроблюваної системи наведено на рисунку 3.3. Її детальний аналіз дозволяє простежити характер взаємодії внутрішніх компонентів та маршрутизацію інформаційних потоків у межах платформи поведінкового аналізу. У процесі проєктування застосовується модель графічного представлення потоків даних (Data Flow Diagram, DFD), яка наочно відображає рух, трансформацію та збереження інформації (зокрема, перехоплених системних викликів, параметрів активності додатків, логів безпеки та результатів проактивного аналізу).

Діаграми взаємодії процесів є фундаментальним інструментом структурного проєктування для візуалізації обробки даних. Традиційно розробка розпочинається з побудови контекстної діаграми потоків даних, яка ілюструє загальну взаємодію розроблюваної системи поведінкового аналізу та обмеження

активності із зовнішнім середовищем (користувачем, операційною системою та аналітичними модулями). У подальшому ця модель деталізується шляхом поетапної декомпозиції процесів і потоків даних з метою повної та всебічної демонстрації архітектури комплексу, що створюється.

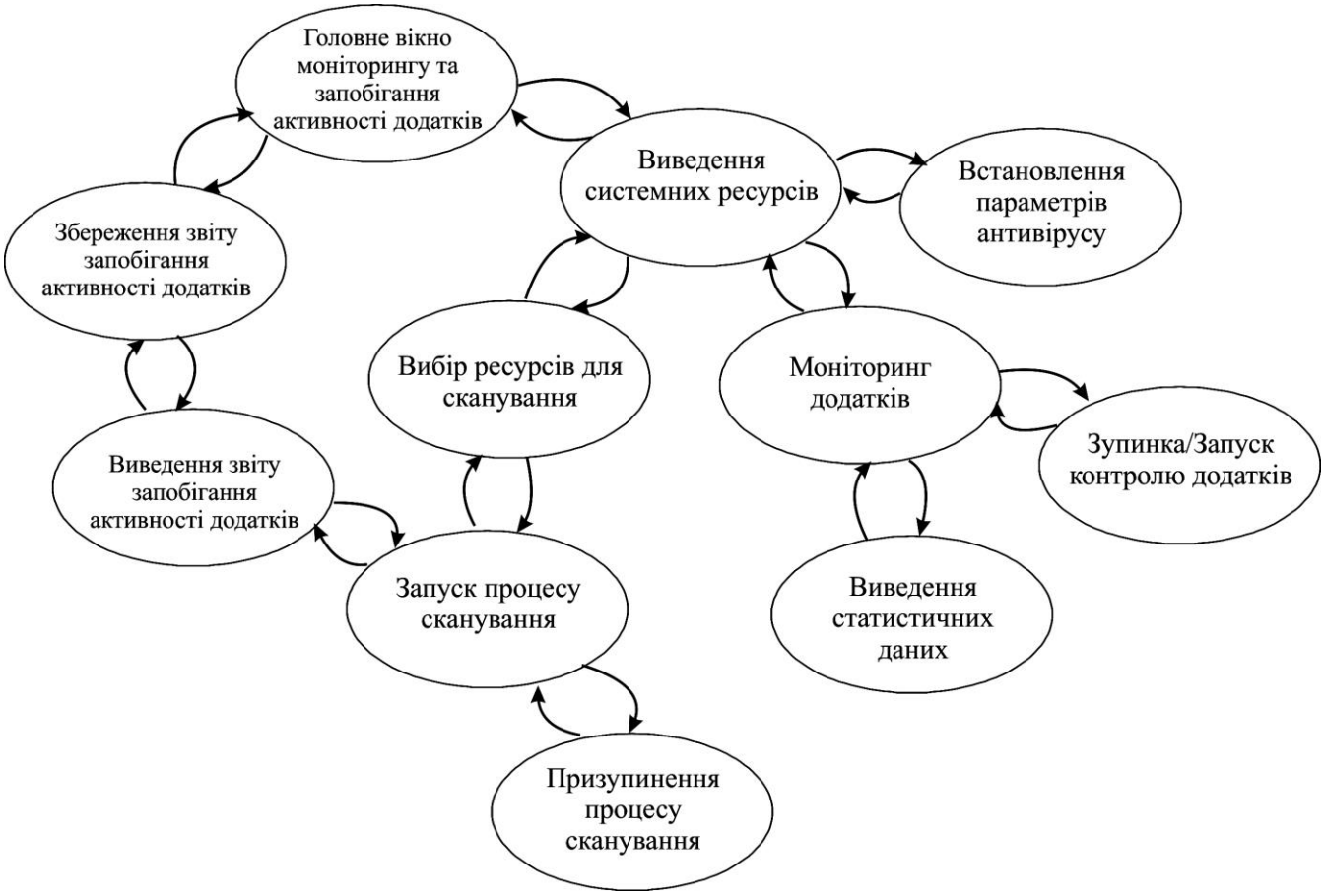

Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Опис алгоритмів функціонування системи

Алгоритмічне забезпечення системи поведінкового аналізу та обмеження активності додатків призначене для автоматизованого спостереження за взаємодією програмних процесів з операційною системою, визначення характерних моделей їхнього функціонування та виявлення потенційно небезпечних дій.

Основним джерелом інформації є системні виклики, за допомогою яких прикладні програми звертаються до функцій ядра операційної системи. Аналіз відповідних подій дозволяє досліджувати операції з файлами, мережевими з'єднаннями, процесами та іншими системними ресурсами. На основі отриманих відомостей передбачається формування поведінкових профілів і визначення умов застосування обмежень. Особливістю системи є поєднання пасивного моніторингу з контрольованим використанням штатних механізмів захисту операційної системи.

Загальний алгоритм функціонування програмного комплексу відображено на рисунку 4.1. Його робота починається з перевірки доступності компонентів моніторингу та завантаження конфігурації політик безпеки.

Після ініціалізації система отримує повідомлення про системні виклики контрольованих процесів. Зібрані дані нормалізуються та зіставляються з відповідними програмними об'єктами.

Далі формується поточний поведінковий профіль і виконується оцінювання відхилень від характерної активності.

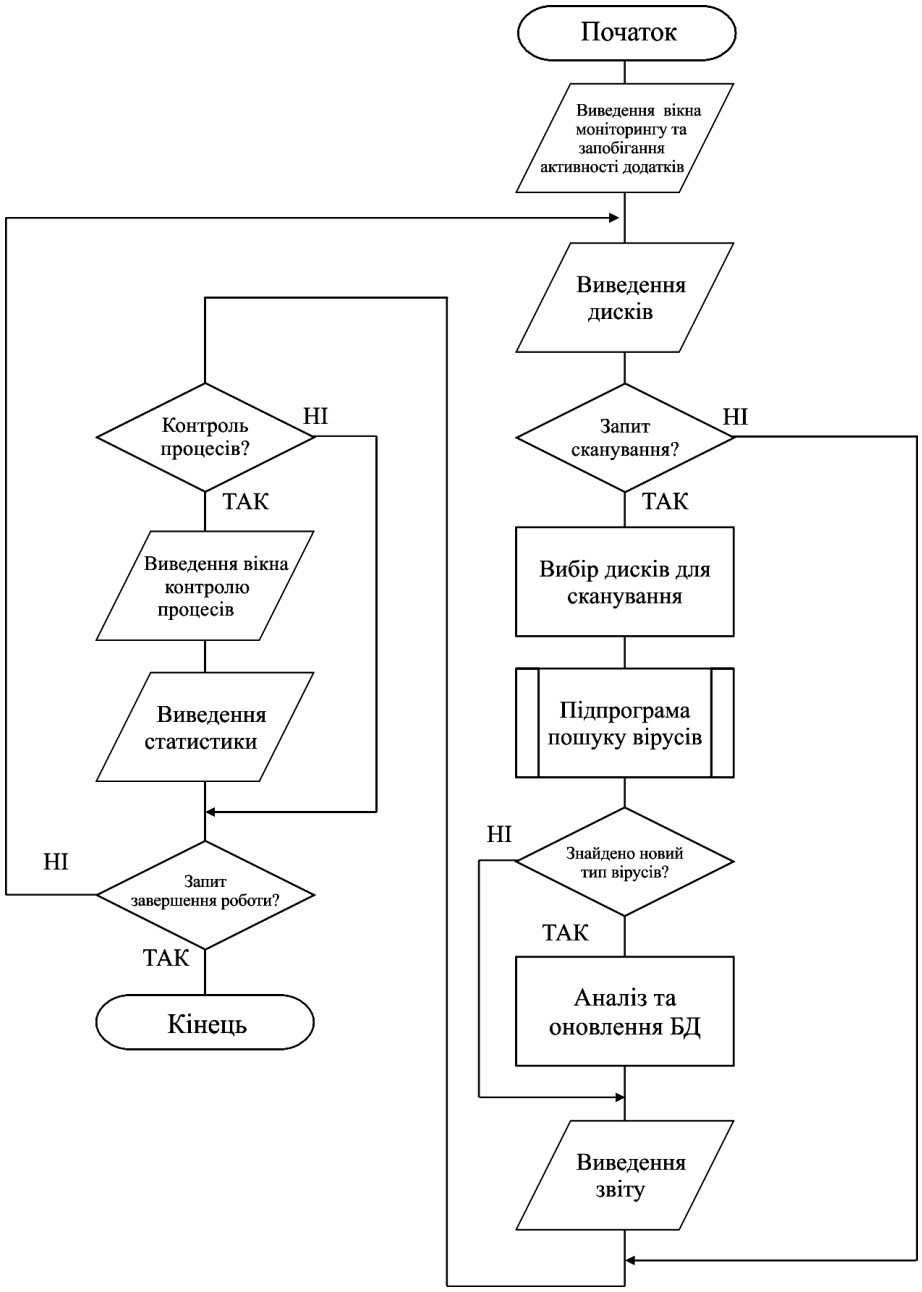

Рисунок 4.1 – Блок-схема основної програми

За результатами аналізу визначається рівень ризику та необхідність реагування. Події зберігаються в журналі аудиту, а інформація про стан додатків передається до користувацького інтерфейсу.

Архітектуру системи доцільно формувати з урахуванням розмежування привілейованих і непривілейованих компонентів. Перший рівень забезпечує отримання подій ядра та застосування дозволених обмежень.

Другий рівень виконує накопичення телеметрії, поведінковий аналіз і підготовку керувальних рішень.

На третьому рівні розміщуються засоби адміністрування, зберігання профілів і перегляду результатів. Така організація дозволяє зменшити обсяг логіки, яка виконується в привілейованому контексті.

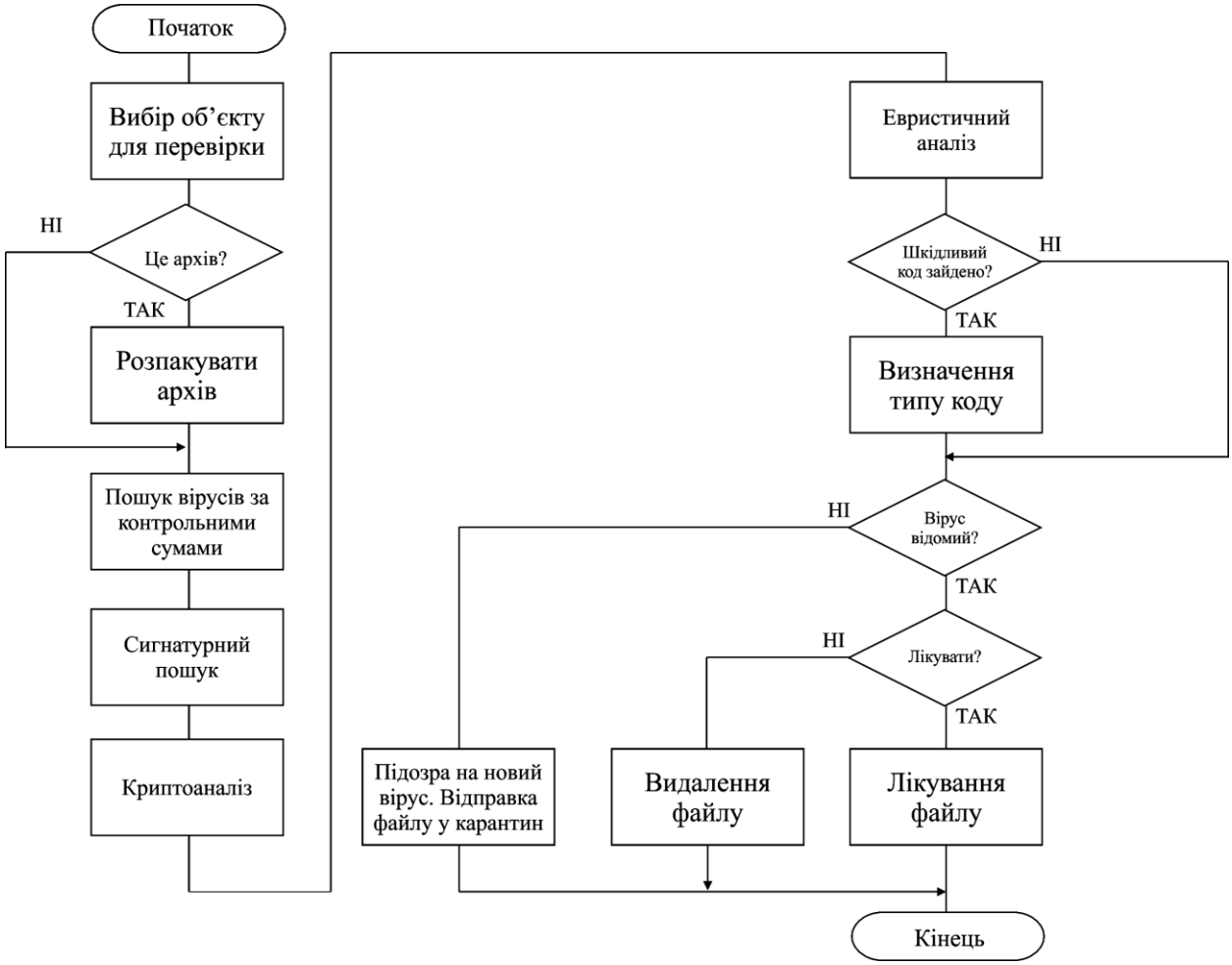

Рисунок 4.2 – Блок-схема роботи підпрограми

Аналітичні алгоритми можуть працювати в окремому процесі з обмеженими повноваженнями.

Як базове середовище дослідження доцільно розглядати Linux, оскільки його ядро надає декілька механізмів спостереження та обмеження процесів. Для отримання подій системних викликів можуть застосовуватися трасувальні точки ядра (Tracepoints), механізм eBPF або Linux Audit. Для обмеження набору доступних системних викликів використовується seccomp. Контроль доступу до

системних об'єктів може забезпечуватися механізмами Linux Security Modules (LSM). Зокрема, Landlock дозволяє додавати обмеження доступу до ресурсів для процесів, які підтримують відповідний спосіб ізоляції. Вибір технології визначається версією ядра, конфігурацією системи та вимогами до захисту.

Необхідно враховувати принципову різницю між спостереженням за системним викликом і його блокуванням. Програма eBPF, підключена до трасувальної точки, може реєструвати факт виконання відповідної операції, але звичайний механізм трасування не надає універсальної можливості забороняти її виконання. Для примусового обмеження необхідно використовувати окремі механізми контролю доступу, які підтримують прийняття рішення до завершення операції. Тому система повинна формувати аналітичне рішення незалежно від способу його технічного застосування.

Роботу підпрограми поведінкового оцінювання та визначення необхідності обмеження додатка представлено на рисунку 4.2. Підпрограма отримує нормалізовану подію, ідентифікатор процесу, поточний поведінковий профіль та активну політику безпеки. Спочатку визначається категорія системного виклику. Далі перевіряється відповідність операції дозволеній поведінці. За наявності відхилень обчислюється оцінка ризику з урахуванням частоти подій, їхньої послідовності та критичності ресурсів. Якщо встановлений поріг перевищено, формується рішення про сповіщення або застосування передбаченого обмеження. Остаточна дія залежить від технічних можливостей механізму захисту й поточного режиму роботи системи.

Для забезпечення коректного функціонування необхідно розрізняти початкову поведінку додатка, його звичайний робочий стан та періоди підвищеної активності. Наприклад, під час запуску програма може відкривати значну кількість бібліотек і конфігураційних файлів. У процесі звичайної роботи кількість файлових операцій може суттєво зменшуватися. Під час завершення можуть виконуватися операції збереження стану та очищення ресурсів. Якщо всі ці етапи об'єднати в один профіль без урахування контексту, система може формувати помилкові спрацювання.

Окремою вимогою є встановлення достовірного зв'язку між системними викликами та конкретним програмним об'єктом. Операційна система створює процеси, які можуть запускати дочірні процеси, виконувати інші програмні файли та використовувати декілька потоків. Тому одного ідентифікатора процесу недостатньо для довготривалого профілювання. Необхідно враховувати початок життєвого циклу процесу, його виконуваний файл, ідентичність користувача та належність до відповідної групи процесів.

Алгоритми аналізу повинні підтримувати роботу за неповної телеметрії. За високого навантаження частина повідомлень може втрачатися через переповнення буферів. Крім того, не всі операції прикладного програмного забезпечення обов'язково супроводжуються окремими системними викликами в момент спостереження. Тому відсутність зареєстрованої події не можна автоматично розглядати як підтвердження відсутності відповідної дії. Для кожного періоду необхідно оцінювати повноту спостереження.

Розглянуті особливості визначають послідовність функціонування програмного комплексу. Спочатку встановлюються контрольовані процеси та доступні джерела подій. Далі виконується нормалізація системних викликів і побудова поведінкових характеристик. Після формування профілів здійснюється оцінювання ризику. Механізми примусового обмеження використовуються лише після перевірки їхньої сумісності з відповідним додатком і підтвердження необхідності втручання.

Алгоритм ініціалізації системи моніторингу

Початковим етапом роботи є перевірка середовища виконання та визначення доступних засобів взаємодії з ядром. Вхідними даними є конфігурація програмного комплексу, перелік контрольованих додатків і параметри моніторингу.

На першому етапі визначається операційна система та її підтримувані функції. Далі перевіряється доступність механізмів трасування системних викликів. Якщо передбачається використання eBPF, необхідно встановити можливість завантаження відповідних програм і підключення до потрібних

трасувальних точок. Необхідні повноваження залежать від конфігурації ядра та встановлених правил безпеки.

Після перевірки можливостей середовища завантажується конфігурація політик. Для кожного контрольованого додатка визначаються режим моніторингу, категорії значущих операцій і допустимі способи реагування.

Наступним етапом є підготовка каналів передавання телеметрії. Для цього можуть використовуватися буфери eBPF Ring Buffer або інші підтримувані механізми обміну між ядром і користувацьким простором.

Після створення відповідних структур запускається колектор подій. Він перевіряє можливість отримання службових повідомлень і реєструє власний стан у журналі.

Якщо необхідний компонент недоступний, система повинна визначити можливість переходу до обмеженого режиму. Наприклад, відсутність привілеїв для eBPF може дозволяти застосування альтернативного джерела аудиту, але такий перехід не повинен приховувати зменшення функціональності.

Результатом алгоритму є підготовлене середовище моніторингу з визначеними можливостями спостереження та реагування.

Алгоритм реєстрації контрольованих додатків

Для аналізу поведінки необхідно сформувати реєстр програмних об'єктів, які підлягають моніторингу. Вхідними даними є перелік виконуваних програм, їхні системні ідентифікатори та відповідні політики безпеки.

На першому етапі визначається спосіб ідентифікації додатка. Для локального програмного забезпечення може використовуватися шлях до виконуваного файлу, його криптографічний відбиток і відомості про власника. Для контейнеризованих застосунків додатково враховується ідентифікатор контейнера або відповідної групи процесів.

Після реєстрації програми система встановлює, які процеси належать до відповідного додатка. Якщо застосунок створює дочірні процеси, вони можуть включатися до тієї самої логічної групи за встановленою політикою.

Особливу увагу необхідно приділити повторному використанню ідентифікаторів процесів. Після завершення процесу операційна система може призначити той самий PID іншому процесу. Тому тривала ідентифікація повинна використовувати додаткові характеристики, зокрема час створення процесу або інший стабільний атрибут його життєвого циклу.

Окремо перевіряється відповідність виконуваного файлу зареєстрованому програмному об'єкту. Якщо файл оновлено, його контрольне значення може змінитися. За таких умов необхідно визначити, чи відповідає нова версія дозволеній конфігурації.

Результатом алгоритму є актуальний реєстр додатків і пов'язаних із ними процесів, який використовується для групування системних викликів.

Алгоритм отримання подій системних викликів

Основним джерелом поведінкових даних є повідомлення, які виникають під час звернення процесів до функцій ядра. Для їхнього отримання можуть застосовуватися трасувальні точки входу та виходу із системних викликів.

Вхідними даними алгоритму є потік повідомлень ядра. На першому етапі визначається номер системного виклику, ідентифікатор процесу та потоку, час виникнення й доступні аргументи.

Для спостереження можуть використовуватися події сімейства sys_enter і sys_exit. Вони дозволяють реєструвати початок виконання виклику та отримання результату. Однак точний склад доступних полів залежить від обраного механізму трасування.

Після отримання повідомлення виконується перевірка належності процесу до контрольованої групи. Якщо подія не відповідає встановленому фільтру, її подальше оброблення може не виконуватися.

Для значущих викликів формується компактний запис. Він містить необхідні службові характеристики та передається до буфера. Доцільно уникати надлишкового копіювання даних між ядром і користувацьким простором.

Окремо необхідно враховувати архітектуру процесора. Номери системних викликів можуть відрізнятися між архітектурами та режимами виконання. Тому

для правильної інтерпретації необхідно враховувати відповідний інтерфейс системних викликів.

Результатом алгоритму є потік первинних повідомлень про активність контрольованих процесів.

Алгоритм зіставлення входу та завершення системного виклику

Для частини поведінкових характеристик недостатньо знати лише факт звернення до ядра. Необхідно визначити результат відповідної операції та тривалість її виконання.

Вхідними даними є події входу й виходу із системних викликів. На першому етапі повідомлення групуються за ідентифікатором потоку та інформацією про поточну операцію.

Після реєстрації входу система зберігає часову позначку й необхідні службові параметри. Коли надходить відповідна подія завершення, визначається результат виклику.

Для типового системного виклику від'ємне значення повернення може відображати помилку. Проте інтерпретація результату повинна враховувати особливості конкретної операції та її ABI.

Після отримання пари подій обчислюється тривалість виконання. Ця характеристика може бути використана для визначення незвичних затримок файлових або мережевих операцій.

Якщо відповідної події завершення немає, запис позначається як неповний. Причиною може бути втрата телеметрії, завершення процесу або специфічна поведінка системного виклику.

Результатом алгоритму є структурований запис виконаної операції з доступним результатом і часовими характеристиками.

Алгоритм нормалізації системних подій

Первинні повідомлення ядра містять технічні атрибути, які не завжди придатні для безпосереднього поведінкового аналізу. Тому необхідно перетворювати їх до уніфікованого представлення.

Вхідними даними є записи системних викликів, відомості про процеси та конфігурація відповідності системних ідентифікаторів.

На першому етапі визначається назва операції. Далі системному виклику призначається функціональна категорія. Наприклад, openat може належати до файлових операцій, connect до мережевих, а execve до операцій запуску програм.

Окремо нормалізуються часові позначки. Для внутрішнього зіставлення подій доцільно використовувати монотонне джерело часу. Для аудиторських записів може додатково зберігатися календарний час.

Після цього до повідомлення додається ідентифікатор програмного об'єкта. Якщо належність процесу не встановлена, подія не повинна довільно приписуватися найближчому зареєстрованому додатку.

Наступним етапом є перевірка коректності даних. Некоректні системні ідентифікатори, відсутні часові характеристики та неузгоджені повідомлення передаються до механізму діагностики.

Результатом алгоритму є структуровані події з однаковими полями, придатні для формування поведінкових профілів.

Алгоритм класифікації системних викликів

Для зменшення розмірності поведінкових даних доцільно групувати системні виклики за їхнім функціональним призначенням.

Вхідними даними алгоритму є нормалізовані повідомлення та таблиця відповідності системних викликів поведінковим категоріям.

До файлових операцій можуть належати openat, read, write та unlinkat. Для аналізу створення процесів можуть використовуватися події, пов'язані з clone, fork, vfork і execve. Мережева активність може характеризуватися викликами socket, connect, accept та sendto.

Окрему категорію становлять операції зміни прав доступу, використання просторів імен та взаємодії з іншими процесами. Їхнє значення залежить від повноважень процесу й поточної політики.

Необхідно враховувати, що однаковий системний виклик може використовуватися для різних прикладних завдань. Наприклад, write може

забезпечувати запис у звичайний файл, канал між процесами або мережевий сокет.

Тому класифікація лише за номером виклику може бути недостатньою. За наявності достовірного контексту доцільно враховувати тип відповідного файлового дескриптора або інші характеристики ресурсу.

Результатом алгоритму є категоризований потік подій, який використовується для статистичного та послідовнісного аналізу.

Алгоритм формування поведінкових часових вікон

Для визначення характерної активності необхідно аналізувати не лише окремі системні виклики, а й їхню сукупність протягом певного інтервалу.

Вхідними даними є нормалізовані події, ідентифікатор додатка та параметри часового вікна.

На першому етапі визначається тривалість інтервалу. Вона повинна відповідати типовій інтенсивності системних викликів контрольованого програмного забезпечення.

Далі події групуються за додатком і часовою належністю. Для кожного вікна визначається кількість викликів, частка окремих категорій та кількість помилок.

Окремо можуть обчислюватися характеристики послідовності. Наприклад, визначається частота переходів між файловими та мережевими операціями.

Для обліку нерівномірного навантаження доцільно враховувати активний час процесу. Порівняння абсолютної кількості викликів без урахування тривалості спостереження може призводити до некоректних висновків.

Якщо протягом інтервалу відбулися втрати телеметрії, відповідне вікно позначається як неповне. Його характеристики не повинні використовуватися для навчання базової моделі без додаткової перевірки.

Результатом алгоритму є послідовність поведінкових вікон із визначеними числовими характеристиками.

Алгоритм формування індивідуального профілю додатка

Для визначення відхилень необхідно мати представлення характерної поведінки кожного контрольованого програмного об'єкта.

Вхідними даними є історія часових вікон, відомості про версію додатка та встановлений режим його роботи.

На першому етапі формується навчальний набір. До нього включаються спостереження, отримані під час контрольованого виконання звичайних функцій програми.

Для кожної категорії системних викликів визначається характерна частота. Додатково обчислюються розподіл результатів виконання та типові послідовності операцій.

Окремо доцільно розрізняти етап запуску, активну роботу та завершення програми. Для кожного стану можуть формуватися незалежні характеристики.

Після обчислення статистичних показників створюється запис поведінкового профілю. Він містить версію програми, перелік використаних ознак, параметри нормалізації та відповідні межі звичайної активності.

Якщо навчальних даних недостатньо, профіль повинен отримувати статус неповного. У такому випадку автоматичне блокування на основі поведінкових відхилень доцільно не застосовувати.

Результатом алгоритму є базова модель активності додатка, призначена для подальшого оцінювання поточних подій.

Алгоритм аналізу частотних характеристик

Одним із найпростіших способів виявлення незвичної поведінки є порівняння поточної інтенсивності системних викликів з історичними показниками.

Вхідними даними є поточне поведінкове вікно та відповідний профіль додатка.

На першому етапі для кожної категорії визначається кількість зареєстрованих операцій за одиницю часу. Далі отримане значення порівнюється з характерним діапазоном.

Наприклад, значне збільшення кількості файлових операцій може свідчити про зміну характеру роботи програми. Однак воно не є безумовним доказом небезпечної активності.

Для зменшення впливу випадкових коливань можуть використовуватися медіана, міжквартильний розмах або інші стійкі статистичні характеристики.

Окремо аналізується частка помилкових системних викликів. Збільшення кількості відмов доступу може свідчити про неправильну конфігурацію або спроби виконання операцій за межами дозволених повноважень.

Результатом алгоритму є набір частотних відхилень, які використовуються під час комплексного оцінювання ризику.

Алгоритм аналізу послідовностей системних викликів

Окремі системні виклики можуть бути цілком звичайними, але їхня послідовність здатна відрізнятися від характерної поведінки додатка.

Для врахування таких залежностей доцільно використовувати аналіз послідовностей операцій.

Вхідними даними є впорядкована історія системних викликів у межах відповідного процесу або програмного об'єкта.

На першому етапі технічні ідентифікатори перетворюються на категоріальні значення. Далі формується послідовність визначеної довжини.

Для початкового аналізу може застосовуватися модель n-грам. Вона характеризує частоту появи коротких послідовностей системних викликів у навчальних даних.

Під час оброблення нового спостереження визначається наявність відповідної послідовності в поведінковому профілі. Рідкісні або раніше невідомі комбінації можуть отримувати підвищену оцінку незвичності.

Для складнішого аналізу можливе використання прихованих марковських моделей або нейромережевих архітектур, які враховують порядок подій. Вибір методу повинен ґрунтуватися на кількості доступних даних і результатах експериментального порівняння.

Важливо враховувати багатопотокове виконання. Системні виклики різних потоків можуть перемежовуватися залежно від планувальника операційної системи. Тому не кожна зміна глобального порядку повідомлень означає зміну логіки додатка.

Для зменшення відповідного впливу доцільно аналізувати послідовності окремих потоків або використовувати агреговані характеристики переходів між категоріями.

Результатом алгоритму є оцінка незвичності поточного порядку системних операцій.

Алгоритм оцінювання доступу до файлових ресурсів

Контроль файлової активності є важливою складовою поведінкового аналізу. Додатки можуть відкривати, змінювати та видаляти файли відповідно до свого функціонального призначення.

Вхідними даними алгоритму є події файлових операцій, характеристики процесу та відповідна політика доступу.

На першому етапі визначається тип операції. Далі за доступними відомостями встановлюється ресурс, до якого здійснюється звернення.

Необхідно враховувати, що системний виклик може отримувати шлях через покажчик на пам'ять користувацького процесу. Саме числове значення такого аргументу не є назвою файлу. Для отримання шляху потрібний відповідний механізм безпечного читання або спостереження на рівні файлових об'єктів.

Окремо слід ураховувати символьні посилання, відносні шляхи та зміну простору імен файлової системи. Через ці особливості порівняння лише текстових шляхів не завжди забезпечує коректний контроль доступу.

Для примусового обмеження доцільно використовувати механізми, які перевіряють права доступу на рівні відповідного системного об'єкта. Залежно від архітектури такими механізмами можуть бути Landlock або відповідні LSM.

Результатом алгоритму є оцінка файлової активності та визначення можливих порушень установлених правил доступу.

Алгоритм аналізу мережевої активності процесів

Мережеві системні виклики дозволяють досліджувати характер взаємодії додатків із зовнішніми ресурсами.

Вхідними даними є події створення сокетів, установлення з'єднань і передавання інформації.

На першому етапі визначається належність мережевої операції до відповідного процесу. Далі аналізуються доступні характеристики з'єднання, зокрема тип сокета та мережева адреса призначення, якщо її можна достовірно отримати.

Окремо визначається частота нових підключень. Незвичне зростання кількості спроб установлення з'єднань може розглядатися як поведінкове відхилення.

Далі поточна мережна активність порівнюється з профілем додатка. Наприклад, програма, яка зазвичай використовує лише локальні ресурси, може потребувати додаткової перевірки після появи зовнішніх мережевих підключень.

Водночас системні виклики не дозволяють автоматично визначати зміст зашифрованого прикладного трафіку. Для відповідного аналізу необхідні інші джерела інформації або спеціально передбачені механізми прикладного журналювання.

Для примусового мережевого обмеження можуть застосовуватися штатні засоби контролю доступу, зокрема мережеві політики LSM або окремі механізми фільтрації.

Результатом алгоритму є характеристика мережевої поведінки додатка з визначенням відхилень від встановленого профілю.

Алгоритм контролю створення та запуску процесів

Додатки можуть запускати допоміжні програми, створювати дочірні процеси та змінювати виконувані образи. Така поведінка потребує окремого аналізу.

Вхідними даними є події створення процесів, виконання програмних файлів і завершення обчислювальних процесів.

На першому етапі визначається батьківський процес. Далі встановлюється зв'язок новоствореного процесу з відповідним зареєстрованим додатком.

Для операцій виконання файлів можуть використовуватися події, пов'язані з execve та execveat. Під час аналізу необхідно враховувати, що виклик execve замінює програмний образ процесу, а не створює новий процес із новим PID.

Наступним етапом є перевірка відповідності виконуваного файлу встановленій політиці. Для додатків із передбачуваною структурою можуть визначатися дозволені допоміжні програми.

Якщо зафіксовано запуск невідомого об'єкта, система формує повідомлення про відхилення. Однак такий запуск може бути пов'язаний із оновленням або штатною функцією програми.

Тому для підтвердження небезпечності необхідно враховувати походження виконуваного файлу, параметри політики та додаткові події.

Результатом алгоритму є актуальна структура процесів додатка та перелік незвичних операцій запуску.

Алгоритм комплексного оцінювання поведінкового ризику

Після виконання окремих перевірок необхідно сформувати узагальнену оцінку поточної активності додатка.

Вхідними даними є частотні характеристики, оцінка незвичності послідовностей, результати перевірки політик та відомості про критичність ресурсів.

На першому етапі виконується нормалізація окремих показників. Значення повинні приводитися до порівнянного масштабу, щоб одна характеристика не впливала на результат лише через більший числовий діапазон.

Далі визначається значущість виявлених відхилень. Наприклад, невелике збільшення кількості файлових операцій може мати нижчу критичність, ніж підтверджена спроба доступу до захищеного системного об'єкта.

Окремо аналізується кількість незалежних ознак. Якщо протягом одного інтервалу спостерігаються незвичне створення процесів, нові мережеві

підключення та зростання кількості помилок доступу, така сукупність може потребувати підвищеної уваги.

Для об'єднання результатів допускається використання багатокритеріальної моделі. Її вагові коефіцієнти та порогові значення необхідно визначати на основі контрольованих тестових сценаріїв.

Далі оцінка порівнюється з установленими порогами. За низького ризику може виконуватися лише журналювання. Для підвищеного ризику формується повідомлення адміністратору. Критичні випадки можуть ініціювати процедуру застосування погодженого обмеження.

Результатом алгоритму є оцінка ризику та пояснення характеристик, які вплинули на прийняте рішення.

Алгоритм визначення політики реагування

Отримана оцінка ризику використовується для вибору допустимого способу реагування. Водночас статистичне відхилення не повинно автоматично призводити до припинення роботи програмного забезпечення.

Вхідними даними є оцінка ризику, характеристики додатка, активна політика та відомості про можливі наслідки блокування.

На першому етапі визначається режим роботи. У режимі спостереження система лише реєструє події та формує повідомлення.

У режимі попередження можливе додаткове інформування адміністратора про виявлені порушення. Для режиму примусового обмеження необхідно заздалегідь визначити дозволені дії.

Далі перевіряється критичність додатка. Для системних служб, від яких залежить робота операційної системи, автоматичне припинення процесу може бути неприйнятним.

Окремо встановлюється, чи має програмний комплекс технічну можливість застосувати відповідне обмеження. Якщо механізм захисту не підтримує потрібну операцію, система повинна сформувати рекомендацію без неправдивого повідомлення про виконане блокування.

Після перевірки обирається допустимий спосіб реагування. Для частини додатків може застосовуватися попередньо налаштований набір обмежень системних викликів. Для інших достатньо повідомлення про підозрілу активність.

Результатом алгоритму є рішення про подальшу дію з урахуванням встановленої політики та можливостей операційної системи.

Алгоритм обмеження системних викликів за допомогою seccomp

Механізм seccomp дозволяє обмежувати набір системних викликів, доступних процесу Linux. Його доцільно використовувати для програм, запуск яких здійснюється під контролем відповідного середовища.

Вхідними даними алгоритму є перелік дозволених системних викликів, параметри процесу та конфігурація фільтра.

На першому етапі формується політика. Для неї визначаються операції, необхідні для звичайного функціонування додатка.

Далі перевіряється сумісність політики з конкретною архітектурою системних викликів. Важливо враховувати, що одна прикладна функція може використовувати різні системні виклики залежно від версії бібліотеки та архітектури процесора.

Після перевірки фільтр може встановлюватися під час запуску відповідного процесу. Для звичайного непривілейованого процесу застосування фільтрів потребує дотримання встановлених ядром умов, зокрема використання no_new_privs.

Залежно від конфігурації seccomp може дозволяти системний виклик, повертати визначену помилку або застосовувати іншу підтримувану дію.

Необхідно враховувати, що seccomp не є універсальним механізмом ізоляції. Він контролює системні виклики та доступні значення аргументів, але не замінює перевірку прав доступу до файлових і мережевих ресурсів.

Додатково слід зазначити, що звичайні фільтри seccomp не можна довільно послаблювати після застосування. Для вже запущених процесів встановлення нової політики потребує відповідної підтримки та не еквівалентне зміні фільтра, який уже діє.

Результатом алгоритму є контрольований запуск додатка з попередньо визначеними обмеженнями системних викликів.

Алгоритм обмеження доступу до системних ресурсів

Не всі небезпечні дії можна коректно описати лише через номер системного виклику. Наприклад, одна й та сама операція відкриття файлу може бути дозволеною або забороненою залежно від конкретного об'єкта.

Для таких випадків необхідно використовувати механізми контролю доступу до ресурсів.

Вхідними даними алгоритму є політика додатка, характеристики файлових або мережевих ресурсів і доступні можливості операційної системи.

Одним із можливих механізмів є Landlock. Він дозволяє створювати додаткові обмеження, які застосовуються до процесу та відповідних дочірніх процесів.

На початковому етапі формується набір дозволів. Для файлової системи можуть визначатися каталоги та типи операцій, до яких додатку надається доступ.

Далі правила застосовуються до процесу через підтримуваний інтерфейс. При цьому необхідно враховувати доступну версію Landlock ABI, оскільки перелік підтримуваних обмежень залежить від можливостей ядра.

Важливо, що Landlock не призначений для довільного розширення повноважень. Додаткові правила можуть звужувати доступ, але не повинні скасовувати інші системні обмеження.

Для загальносистемного примусового контролю можуть застосовуватися відповідні LSM. Зокрема, BPF LSM дозволяє реалізовувати перевірки через підтримувані механізми ядра.

Результатом алгоритму є визначена конфігурація ресурсного доступу, яка доповнює поведінкове спостереження.

Алгоритм контрольованого обмеження активності процесу

У разі підтвердженої небезпечної поведінки може виникати потреба в тимчасовому обмеженні активності вже запущеного додатка.

Для цього доцільно передбачити окремий алгоритм реагування, який не залежить безпосередньо від механізму трасування.

Вхідними даними є ідентифікатор процесу, рішення політики, рівень критичності та дозволений спосіб впливу.

На першому етапі перевіряється актуальність процесу. Необхідно встановити, що він не завершився й не був замінений іншим процесом із повторно використаним PID.

Далі визначається характер обмеження. Для зменшення використання ресурсів можуть застосовуватися контрольовані обмеження CPU та пам'яті через cgroups.

Для припинення небажаних операцій доступу необхідно використовувати відповідні механізми контролю, які підтримують застосування політик до конкретного процесу або групи процесів.

Окремо перевіряється можливість безпечного завершення. Примусове припинення процесу може призвести до втрати незбережених даних або порушення роботи залежних компонентів.

Якщо ризик такого втручання є неприйнятним, система повинна обмежитися формуванням повідомлення та передаванням інциденту адміністратору.

Результатом алгоритму є виконане дозволене обмеження або обґрунтована відмова від автоматичного втручання.

Приклад програмної реалізації поведінкового оцінювання

Для пояснення принципу роботи аналітичної підсистеми може використовуватися функція мовою Python. Наведений фрагмент демонструє обчислення поведінкових відхилень на основі агрегованих системних викликів. Він не виконує безпосереднього блокування процесів і не є підтвердженням фактичного використання саме такого коду в розроблюваній системі.

У прикладі аналізуються кількість файлових, мережевих і процесних операцій, а також кількість помилок доступу. Для кожного показника

використовується відповідна базова характеристика. На основі сукупності відхилень формується умовна оцінка ризику.

```
from dataclasses import dataclass
from math import isfinite


@dataclass(frozen=True)
class ActivityWindow:
    # Кількість операцій у завершеному часовому вікні
    file_calls: int
    network_calls: int
    process_calls: int
    access_errors: int


@dataclass(frozen=True)
class BehaviorProfile:
    # Еталонні значення для однакової тривалості вікна
    file_baseline: float
    network_baseline: float
    process_baseline: float
    error_baseline: float


def evaluate_activity(
    window: ActivityWindow,
    profile: BehaviorProfile
) -> dict:
    """Оцінювання відхилень від звичайної поведінки додатка."""

    observed = {
        "file": window.file_calls,
        "network": window.network_calls,
        "process": window.process_calls,
        "errors": window.access_errors
    }

    baseline = {
        "file": profile.file_baseline,
        "network": profile.network_baseline,
        "process": profile.process_baseline,
        "errors": profile.error_baseline
    }

    # Перевіряємо коректність отриманих характеристик
    if any(value < 0 for value in observed.values()):
        return {
            "status": "INVALID_INPUT",
            "reason": "Від'ємна кількість операцій"
        }

    if any(
        not isfinite(value) or value < 0
        for value in baseline.values()
    ):
        return {
            "status": "INVALID_PROFILE",
            "reason": "Некоректний поведінковий профіль"
        }

    deviations = {}
```

```
for category, actual in observed.items():
    expected = baseline[category]

    # Обчислюємо відносне перевищення еталонного рівня
    denominator = max(expected, 1.0)

    deviation = max(
        0.0,
        (actual - expected) / denominator
    )

    deviations[category] = deviation

# Помилки доступу мають підвищену вагу в умовній оцінці
weights = {
    "file": 0.2,
    "network": 0.3,
    "process": 0.2,
    "errors": 0.3
}

risk_score = sum(
    weights[name] * min(value, 3.0)
    for name, value in deviations.items()
)

# Пороги демонструють принцип класифікації
if risk_score >= 1.5:
    severity = "HIGH"
elif risk_score >= 0.6:
    severity = "MEDIUM"
else:
    severity = "LOW"

return {
    "status": "EVALUATED",
    "risk_score": round(risk_score, 4),
    "severity": severity,
    "deviations": deviations,
    "recommended_mode": "REVIEW" if severity == "HIGH"
                        else "MONITOR"
}
```

У наведеному прикладі виконується перевірка коректності поточних і еталонних значень. Далі обчислюються відносні перевищення характерної інтенсивності операцій. Підсумкова оцінка визначається за допомогою вагової моделі.

Для випадків високого ризику функція повертає рекомендацію щодо додаткової перевірки. Вона не змінює права доступу та не припиняє роботу процесу.

Необхідно враховувати, що числові ваги й пороги є демонстраційними. У практичній системі відповідні параметри повинні визначатися за результатами експериментального аналізу нормальної та аномальної активності.

Також приклад передбачає однакову тривалість часових вікон. Для змінних інтервалів необхідно використовувати нормалізовану інтенсивність операцій за одиницю часу.

Алгоритм динамічного оновлення поведінкового профілю

Поведінка програмного забезпечення може змінюватися після оновлення, додавання нових функцій або зміни навантаження. Тому профіль не повинен залишатися незмінним протягом усього життєвого циклу додатка.

Вхідними даними алгоритму є поточний профіль, нові поведінкові вікна та результати їхнього оцінювання.

На першому етапі перевіряється повнота нових спостережень. Якщо під час збору телеметрії виникли втрати, такі дані не повинні автоматично використовуватися для навчання.

Далі аналізується рівень ризику. Спостереження з ознаками підозрілої поведінки доцільно тимчасово виключати з процедури оновлення.

Для допустимих даних може застосовуватися експоненційне згладжування. Нове значення профілю формується з урахуванням попереднього стану й останнього спостереження.

Швидкість адаптації необхідно обмежувати. Інакше поступове накопичення небезпечних операцій може призвести до закріплення відповідної поведінки як нормальної.

Окремо потрібно враховувати оновлення програмного забезпечення. Зміна версії додатка може потребувати повторного формування профілю або його перевірки на новому наборі даних.

Результатом алгоритму є актуалізований профіль із можливістю відновлення попередньої перевіреної версії.

Алгоритм оброблення втрат телеметрії

Система моніторингу повинна враховувати обмеження пропускної здатності каналів передавання подій. За високої інтенсивності системних викликів накопичення повідомлень може перевищувати можливості користувацького компонента.

Вхідними даними алгоритму є характеристики буферів, кількість отриманих подій та доступні лічильники втрат.

На першому етапі визначається поточне завантаження буфера. Якщо воно наближається до критичного значення, можуть застосовуватися додаткові фільтри для незначущих категорій подій.

При цьому важливо зберігати інформацію про критичні операції, які необхідні для контролю встановлених політик.

Якщо повідомлення втрачено, відповідний факт реєструється в службовій статистиці. Поведінкове вікно, на яке вплинула втрата, позначається як неповне.

Окремо контролюється затримка оброблення. Якщо події надходять швидше, ніж аналізуються, система повинна оцінити можливість тимчасового накопичення або зменшення деталізації моніторингу.

Результатом алгоритму є визначення повноти телеметрії та запобігання використанню неповних спостережень як достовірних характеристик звичайної поведінки.

Алгоритм формування журналів аудиту

Для подальшого аналізу необхідно зберігати результати поведінкового моніторингу та виконаних обмежень.

Вхідними даними є нормалізовані події, оцінки ризику та повідомлення керувальних компонентів.

На першому етапі визначається категорія запису. До неї можуть належати звичайна активність, поведінкова аномалія, порушення політики або результат застосованого обмеження.

Далі формується структурований запис, який містить ідентифікатор додатка, часові характеристики, причину спрацювання та відповідний результат.

Для зменшення обсягу сховища допускається агрегування повторюваних незначущих подій. Однак підтверджені порушення та результати примусових обмежень повинні зберігатися з необхідною деталізацією.

Окремо встановлюються правила доступу до журналів. Вони можуть містити інформацію про виконувані програми, файлові ресурси та мережеву активність користувачів.

За потреби використовуються криптографічний контроль цілісності та захищене передавання повідомлень до центрального сховища.

Результатом алгоритму є історія подій і керувальних дій, придатна для перевірки та аналізу.

Алгоритм оброблення помилок і відмов компонентів

Під час функціонування можливі порушення доступності джерела телеметрії, помилки оброблення подій, пошкодження профілю або недоступність механізму обмеження.

На першому етапі визначається категорія несправності. Далі перевіряється можливість продовження роботи в обмеженому режимі.

Якщо аналітичний компонент тимчасово недоступний, колектор може продовжувати накопичення подій за наявності достатніх ресурсів.

Якщо відмовив компонент зберігання, необхідно контролювати заповнення тимчасових буферів. За відсутності можливості надійного накопичення формується повідомлення про неповноту аудиту.

Окремо необхідно враховувати збої керувального механізму. Якщо обмеження не застосоване, система повинна відображати фактичний результат, а не лише намір виконати відповідну дію.

Політика реагування на відмову визначається призначенням контрольованого додатка. Для звичайних застосунків може бути допустиме збереження роботи в режимі моніторингу. Для спеціально ізольованих компонентів необхідно дотримуватися попередньо встановлених обмежень.

Результатом алгоритму є кероване реагування на технічні порушення з правильним відображенням стану захисту.

Особливості програмної реалізації системи

Програмна реалізація повинна враховувати відмінності між засобами ядра та користувацькими алгоритмами аналізу.

Для отримання подій eBPF доцільно використовувати спеціалізований компонент, сумісний із конфігурацією Linux. Він відповідає за підключення до трасувальних точок і передавання структурованої телеметрії.

Як можливі засоби реалізації можуть використовуватися libbpf та відповідні інструменти компіляції програм eBPF. Конкретний вибір залежить від цільової версії ядра та доступних можливостей.

Аналітичний компонент може бути реалізований мовою Python. Для оброблення статистичних даних можуть застосовуватися стандартні бібліотеки або NumPy. За потреби поведінкові моделі можуть реалізовуватися засобами scikit-learn.

Для локального зберігання профілів і журналів може використовуватися SQLite. Для централізованого розгортання можливе використання PostgreSQL або іншого сумісного сховища.

Важливо забезпечити відокремлення прав аналітичного компонента від прав керування політиками ядра. Передавання керувальних рішень повинно здійснюватися через визначений інтерфейс із перевіркою авторизації.

Окремо необхідно передбачити контроль версій політик. Перед застосуванням нових правил доцільно виконувати їхню перевірку на тестових сценаріях.

Забезпечення безпеки компонентів моніторингу

Система, яка отримує дані з ядра та керує обмеженнями процесів, сама становить важливий компонент інформаційної безпеки.

Тому її програмні модулі повинні використовувати мінімально необхідні повноваження.

Для компонента збору подій необхідно обмежити доступ до привілейованих інтерфейсів. Після завантаження потрібних механізмів він може передавати інформацію менш привілейованому процесу аналізу.

Окремо захищаються конфігурації політик. Їхня несанкціонована зміна може призводити до надмірних обмежень або послаблення захисту.

Для користувацького інтерфейсу передбачається автентифікація адміністратора та розмежування операцій перегляду і зміни правил.

Важливо враховувати конфіденційність телеметрії. Системні виклики можуть містити аргументи, які опосередковано розкривають назви файлів, параметри процесів або інші службові відомості.

Тому доцільно обмежувати склад збережених аргументів і не накопичувати вміст користувацьких буферів без визначеної необхідності.

Результатом відповідних заходів є зменшення ризику використання самого комплексу моніторингу як джерела додаткових загроз.

Перевірка коректності алгоритмів та оцінювання ефективності

Для дослідження запропонованого алгоритмічного забезпечення необхідно передбачити функціональні, поведінкові та навантажувальні випробування.

Початковим етапом є перевірка реєстрації процесів. Для цього можуть використовуватися контрольовані програми, які створюють дочірні процеси та виконують визначені файлові й мережеві операції.

Після виконання перевіряється правильність встановлення зв'язку між подіями системних викликів і відповідними програмними об'єктами.

Окремо оцінюється оброблення повторного використання PID. Необхідно підтвердити, що події нового процесу не додаються до профілю вже завершеного процесу лише через збіг числових ідентифікаторів.

Наступним етапом є перевірка отримання системних викликів. Для цього використовуються навчальні програми, які виконують визначені операції з файлами, процесами та мережевими з'єднаннями.

Для кожного сценарію формується очікуваний набір подій. Після виконання порівнюються фактичні результати та дані моніторингу.

Окремо перевіряється зіставлення входу й виходу із системного виклику. Для контрольованих операцій необхідно підтвердити правильність визначення результату та тривалості.

Під час випробувань необхідно враховувати, що кількість системних викликів може відрізнятися залежно від використовуваних бібліотек та

операційної системи. Тому очікувані характеристики повинні формуватися для конкретного тестового середовища.

Для перевірки поведінкового профілю можуть використовуватися декілька запусків однієї програми за однакових умов.

На основі частини спостережень формується базовий профіль. Наступні виконання використовуються для оцінювання відхилень.

Окремо створюються контрольовані сценарії зміненої поведінки. Наприклад, тестова програма може виконувати значно більшу кількість файлових операцій або створювати додаткові дочірні процеси.

Отримані результати дозволяють перевірити, чи правильно система визначає нетипову активність.

Для оцінювання якості виявлення можуть використовуватися Precision, Recall та F1-score за наявності достовірно розмічених тестових сценаріїв.

Окремо необхідно оцінювати частоту хибнопозитивних спрацювань. Надмірна кількість повідомлень про звичайну активність може ускладнювати експлуатацію комплексу.

Для порівняння алгоритмів можуть використовуватися прості порогові правила, статистичні профілі та аналіз послідовностей системних викликів.

Випробування повинні виконуватися на однакових наборах даних. Це дозволяє визначити, чи забезпечує використання складнішої поведінкової моделі практично значуще покращення.

Особливу увагу необхідно приділити тестуванню механізмів обмеження. Для цього використовуються спеціально підготовлені нешкідливі програми з визначеними операціями.

У першому сценарії перевіряється дозволена дія. У другому виконується операція, яку забороняє відповідна політика.

Після виконання необхідно переконатися, що заборонена операція не була виконана та що програма отримала очікуваний результат відмови.

Окремо перевіряються системні виклики, необхідні для нормального завершення програми. Надмірно сувора політика не повинна створювати неконтрольоване блокування стандартного життєвого циклу додатка.

Для механізмів файлового доступу необхідно передбачити перевірку відносних шляхів, символьних посилань та інших особливостей файлової системи.

Під час навантажувальних випробувань можуть вимірюватися кількість отриманих подій за секунду, використання процесорного часу та оперативної пам'яті.

Важливим показником є додаткова затримка, яку створює моніторинг. Для її визначення доцільно порівнювати виконання однакового навантаження без активного трасування та з увімкненим збором подій.

Окремо необхідно оцінювати вплив механізмів примусового контролю. Застосування політик ядра може створювати додаткові витрати, які відрізняються від витрат пасивного спостереження.

Для перевірки стійкості до перевантаження можуть формуватися інтенсивні потоки системних викликів. Під час аналізу визначається момент появи втрат і правильність реєстрації неповних часових вікон.

За результатами випробувань можуть встановлюватися допустима кількість контрольованих процесів, інтенсивність подій та вимоги до ресурсів програмного комплексу.

Окремого дослідження потребує відмовостійкість. Для цього моделюються зупинка аналітичного компонента, недоступність сховища та порушення роботи колектора.

Очікуваним результатом є правильне відображення стану системи й контрольоване відновлення оброблення без помилкового повідомлення про повноцінний моніторинг.

Кількісні висновки щодо точності поведінкового аналізу, ефективності обмежень і продуктивності системи можуть бути сформовані лише після проведення відповідних експериментів.

Розглянуте алгоритмічне забезпечення формує послідовний процес контролю активності додатків від отримання системних викликів до побудови поведінкових профілів і визначення необхідності застосування обмежень. Основними складовими є моніторинг процесів, нормалізація системних подій, аналіз частотних характеристик і послідовностей операцій, а також контрольоване реагування через штатні механізми операційної системи. Важливе значення має розмежування функцій спостереження й примусового контролю. Поєднання цих механізмів створює основу для програмної реалізації системи, призначеної для виявлення нетипової активності та підвищення керованості доступу додатків до системних ресурсів.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення системи поведінкового аналізу та обмеження активності додатків призначене для контролю взаємодії прикладних процесів з операційною системою на основі моніторингу системних викликів. Система забезпечує реєстрацію файлових, мережевих і процесних операцій, формування поведінкових профілів, виявлення нетипової активності та застосування дозволених політик обмеження. Отримані відомості використовуються для оцінювання ризиків і ведення журналів аудиту. Головне вікно програми показане на рисунку 5.1.

Впровадження програмного комплексу передбачає підготовку операційної системи, налаштування механізмів отримання подій ядра, встановлення аналітичних компонентів та перевірку працездатності засобів обмеження активності. Основними функціональними складовими є колектор системних викликів, модуль нормалізації подій, компонент поведінкового аналізу, служба керування політиками та сховище результатів моніторингу.

Як цільове середовище доцільно використовувати операційну систему Linux із підтримкою необхідних механізмів ядра. Для спостереження за системними викликами можуть застосовуватися Tracepoints, eBPF або Linux Audit. Конкретний спосіб отримання телеметрії визначається версією ядра, конфігурацією безпеки та доступними системними інтерфейсами. Перед встановленням необхідно перевірити підтримку відповідних можливостей і наявність потрібних повноважень.

Для реалізації компонента отримання подій eBPF як можливий технологічний варіант розглядається використання C та бібліотеки libbpf. Аналітична частина може реалізовуватися мовою Python. Вона забезпечує оброблення структурованих повідомлень, формування статистичних характеристик і оцінювання поведінкових відхилень. Для побудови складніших

моделей можуть використовуватися NumPy та scikit-learn. Зазначені засоби становлять рекомендований технологічний стек, конкретний склад якого необхідно визначати за фактичною реалізацією.

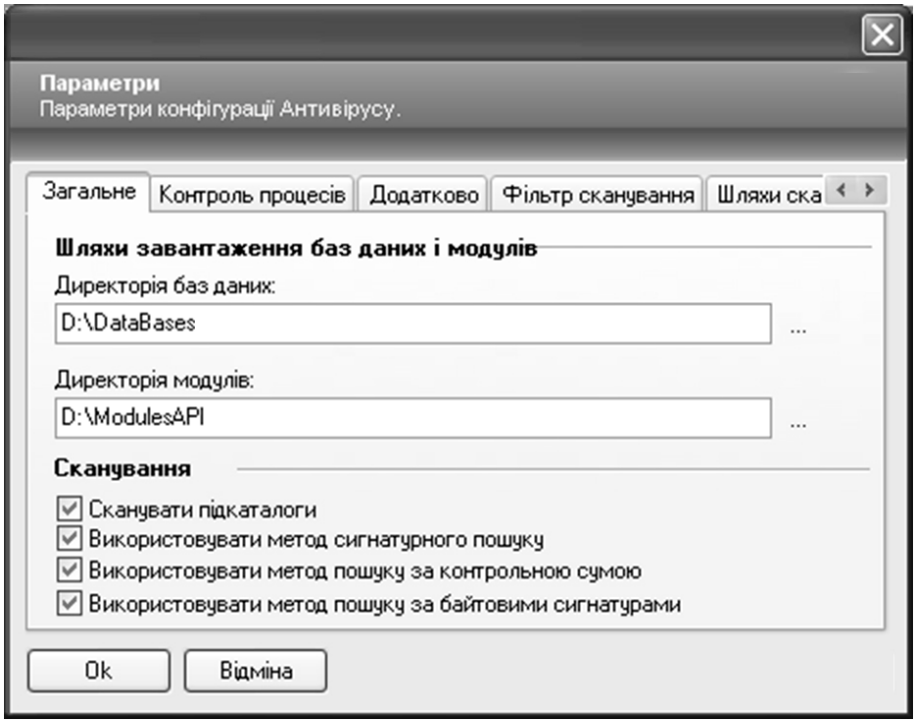

Рисунок 5.1 – Головне вікно ПЗ

Вимоги до апаратного забезпечення залежать від кількості контрольованих процесів та інтенсивності системних викликів. Особливу увагу необхідно приділити продуктивності процесора й обсягу оперативної пам'яті, оскільки оброблення великого потоку подій створює додаткове навантаження. Для зберігання профілів і журналів також необхідно передбачити достатній дисковий простір. Під час тривалої експлуатації доцільно встановити обмеження на обсяг накопиченої телеметрії.

Підготовка програмного середовища починається з перевірки доступності системних засобів трасування. Якщо передбачається використання eBPF, необхідно забезпечити сумісність програм із цільовим ядром та можливість їхнього завантаження. Для збирання відповідних компонентів можуть знадобитися компілятор Clang та інструменти роботи з BPF. У разі застосування

Python доцільно створити ізольоване середовище та встановити необхідні програмні залежності.

Після підготовки системи налаштовується компонент захоплення подій. Для нього визначаються категорії системних викликів, які підлягають моніторингу. До них можуть належати операції відкриття файлів, створення процесів, запуску виконуваних об'єктів і встановлення мережевих з'єднань. Передавання повідомлень із ядра до користувацького компонента може здійснюватися через підтримуваний буфер eBPF. Необхідно перевірити правильність отримання подій та можливість контролю втрат телеметрії.

Наступним етапом є реєстрація контрольованих додатків. Для кожного програмного об'єкта визначаються виконуваний файл, відповідні процеси та правила моніторингу. За потреби використовується криптографічний відбиток виконуваного файлу. Під час зіставлення подій необхідно враховувати, що операційна система може повторно використовувати PID після завершення процесу. Тому ідентифікація повинна спиратися також на характеристики його життєвого циклу.

Для роботи аналітичного компонента необхідно підготувати сховище поведінкових профілів. У локальному варіанті може використовуватися SQLite, а для централізованого розгортання PostgreSQL. Сховище повинно забезпечувати збереження характеристик додатків, статистики системних викликів, версій профілів та результатів оцінювання ризику. Окремо доцільно зберігати журнали застосування політик, щоб відрізняти виявлене порушення від фактично виконаного обмеження.

Початкове формування поведінкових профілів доцільно виконувати в режимі спостереження. Для цього контрольовані додатки запускаються за штатних умов із виконанням характерних функцій. Система накопичує статистику файлових, мережевих і процесних операцій. Отримані дані використовуються для визначення звичайної інтенсивності системних викликів та характерних послідовностей дій. До завершення початкового навчання автоматичне блокування на основі статистичних відхилень доцільно залишати вимкненим.

Після формування профілів налаштовуються правила оцінювання ризиків. Для цього визначаються допустимі відхилення, критичність окремих категорій операцій та умови формування сповіщень. Порогові значення необхідно уточнювати за результатами контрольованих випробувань. Окремо встановлюються правила роботи за неповної телеметрії, оскільки втрата подій не повинна сприйматися як підтвердження безпечного стану додатка.

Для обмеження системних викликів може використовуватися seccomp. Відповідні фільтри доцільно налаштовувати перед запуском контрольованої програми. Політика повинна враховувати системні виклики, необхідні для нормального функціонування додатка. Довільна зміна вже встановлених обмежень для активного процесу не повинна розглядатися як стандартна можливість seccomp. Перед застосуванням необхідно перевірити сумісність фільтра з програмою в тестовому середовищі.

Для обмеження доступу до файлових ресурсів може застосовуватися Landlock, якщо його підтримує цільове ядро. Він дозволяє звужувати доступ до визначених об'єктів файлової системи в межах підтримуваних механізмів. Для контролю використання процесорного часу та оперативної пам'яті можуть застосовуватися cgroups. Ці засоби виконують різні функції та повинні налаштовуватися відповідно до конкретної політики обмеження.

Важливо розмежувати функції моніторингу й примусового контролю. Отримання подій через звичайні трасувальні точки не означає можливості блокування відповідного системного виклику. Тому рішення аналітичного компонента передається до окремого механізму реагування. Для критичних системних служб автоматичне припинення роботи необхідно обмежувати, щоб уникнути порушення функціонування операційної системи.

Рекомендована послідовність запуску починається з перевірки доступності сховища та завантаження конфігурації моніторингу. Після цього активується компонент отримання системних подій і перевіряється передавання телеметрії. Далі запускається модуль нормалізації та завантажуються поведінкові профілі. Наступним етапом є активація механізмів оцінювання ризику й реєстрації

сповіщень. Завершується запуск перевіркою користувацького інтерфейсу та стану доступних засобів обмеження.

Первинне тестування доцільно проводити з використанням спеціально підготовлених нешкідливих програм. Для перевірки файлового моніторингу тестовий додаток може відкривати, читати та створювати файли у визначеному каталозі. Для перевірки мережевих подій використовується контрольоване підключення до тестового сервера. Після виконання необхідно підтвердити правильність реєстрації системних викликів, встановлення належності процесу та формування відповідних записів.

Окремо перевіряється виявлення поведінкових відхилень. Для цього тестова програма виконує контрольовану кількість додаткових операцій, які відрізняються від початкового профілю. Система повинна сформувати відповідну оцінку та відобразити причину спрацювання. Результат такого аналізу не повинен автоматично трактуватися як підтвердження шкідливої активності.

Перевірку примусових обмежень необхідно виконувати в ізольованому середовищі. Для цього формується політика, яка дозволяє частину тестових операцій і забороняє інші. Після запуску програми перевіряється фактичний результат виконання. Якщо обмеження застосовується через seccomp або Landlock, необхідно підтвердити відмову відповідної забороненої операції. Одночасно контролюється збереження працездатності дозволених функцій.

У процесі експлуатації можливі помилки завантаження eBPF, недостатні системні повноваження, переповнення буферів і недоступність сховища. Для діагностики використовуються службові журнали та показники стану компонентів. За відмови аналітичної підсистеми допускається продовження пасивного моніторингу, якщо це підтримується прийнятою архітектурою. При цьому фактичні обмеження, встановлені засобами ядра, повинні відображатися окремо від стану аналітичної служби.

Перед введенням комплексу в регулярну експлуатацію необхідно перевірити його роботу після перезапуску, правильність відновлення конфігурації, захист поведінкових профілів і розмежування адміністративних

повноважень. Додатково оцінюється вплив моніторингу на продуктивність додатків, повнота отримання подій та коректність застосування дозволених політик. Результати перевірок використовуються для визначення експлуатаційної готовності системи.

Для ідентифікації програмного продукту передбачається представлення відомостей про його розробника. Вони необхідні для документування авторства, визначення походження програмного забезпечення та організації подальшого технічного супроводження. Інформація про автора системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів представлена на рисунку 5.2.

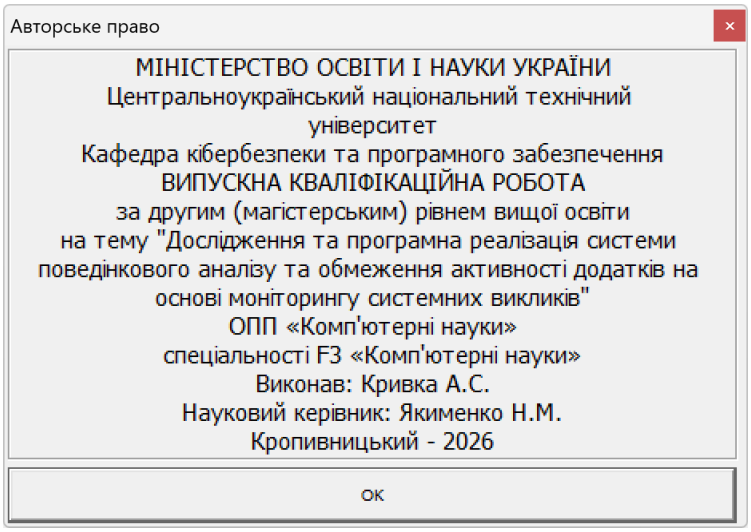

Рисунок 5.2 – Шаблонізована форма авторського права

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Метою розробки є дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Об'єктом дослідження є процес поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Предметом дослідження є методи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Методи дослідження базуються на методах поведінкового аналізу, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– Розроблено вітчизняний продукт поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи.

Згідно закону України “Про охорону праці” [1] кожна компанія впроваджує заходи з охорони праці. Реалізуються трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

– Інструкцій з охорони праці по кожній професії і загальні.

– Положення про охорону праці.

– Накази з охорони праці.

– Журнали реєстрації та інструктажу.

Законом України “Про охорону праці” [2] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема, Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [3], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [4].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки

(суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ), спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) при роботі апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Науково-технічний прогрес суттєво вплинув на умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

7.2 Аналіз умов праці

У приміщенні розташовано 3 робочих місця з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м^2, об'єм не менше 20 м^3. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м^2. Необхідна площа на 3 робочих місця із установленими ПК складає 18 м^2, що не перевищує фактичну. Обсяг приміщення на одного працюючого складає 31,5м^3, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м^3.

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань.

Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається бібліограф при роботі з ПК, приведені в таблиці 7.1.

Таблиця 7.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

За категорією вибухо- і пожежонебезпеки, дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії

пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» [3] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

За ступенем небезпеки поразки людей електричним струмом приміщення класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціювання в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціювання» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіН3.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин».

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м^3.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

Одним з найбільш поширеніших чинників зовнішнього середовища, який несприятливо впливає на людину, є шум. Вплив шуму на організм людини залежить від рівня звукового тиску, частотних характеристик, тривалості дії, а також індивідуальних особливостей людини.

При тривалій дії шуму у оператора ЕОМ виявляються симптоми утомленості, нервового збудження, що сприяють погіршенню працездатності і допущенні помилок при роботі. Для уникнення шкідливої дії шуму на організм

працюючого, необхідне дотримання нормованих параметрів, які не повинні перевищувати допустимих величин. При роботі на комп'ютері рівень шуму не повинен перевищувати 50 дБА. Приміщення розташоване вікнами у двір і знаходиться далеко від проїжджої частини вулиці. Основними джерелами шуму в приміщенні є устаткування і люди. Розглянута кімната не призначена для прийому відвідувачів і тому в ній не спостерігається великого скупчення людей. Тому основним джерелом шуму є комп'ютерна техніка.

Джерелами шуму при роботі ЕОМ є механічні частини принтера, що рухаються, і вентилятори ($L_{пк}$ = 35 дБА, L_{prn}= 48 дБА) При роботі вентиляційної системи, що забезпечує оптимальний температурний режим електронних блоків ЕОМ і вмонтована в задню панель, створюється аеродинамічний шум. Шум, створюваний працюючим комп'ютером, може бути охарактеризований як широко смуговий постійний з аперіодичним посиленням при роботі принтера. Час роботи ПЕОМ – 6 – 8 год. за добу; принтери працюють не більш 1,5-2 год. за добу.

При наявності великої кількості джерел шуму еквівалентне значення шуму $L_{ЕКВ}$, дБА розраховують за наступною формулою:

$$L_{екв}=10\cdot\lg\left(\frac{1}{T}\sum_{i=1}^{n}\left(t_i\cdot 10^{0.1\cdot L_i}\right)\right) \qquad (7.1)$$

де

L_i – рівень шуму і-го джерела (пристрою),

t_i – час роботи і-го джерела (пристрою),

T – загальний час роботи,

n – кількість джерел шуму даного типу;

Для даного приміщення необхідні змінні складають:

Загальний час роботи – робітник день, тобто T=8 годин.

Для фонового шуму (вентиляторів):

L_1 = 35 дБА, T_1 = 8 годин, n_1=15 (5х3);

Для лазерного принтера Lexmark Jet:

L_2 = 48дБА, T2 = 2 години, n_2=1, для сканера L_3 = 46дБА, T3 = 2 години.

Підставляємо отримані величини у формулу (7.1):

$$L_{\text{екв}} = 10 \cdot \lg\left(\frac{1}{8} \cdot \left(15 \cdot 8 \cdot 10^{0.1 \cdot 35} + 1 \cdot 2 \cdot 10^{0.1 \cdot 48} + 1 \cdot 2 \cdot 10^{0.1 \cdot 46}\right)\right) = 46{,}3 \text{ дБА}$$

Таким чином, еквівалентний рівень шуму в приміщенні за робочий день $L_{екв}$ = 46,3 дБА , тобто не перевищує норму 50 дБА.

7.3 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізольований кабель. Висота проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемами заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 В, а одна, забезпечує подачу напруги в 36 В. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежобезпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки.

До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6] приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однією з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

7.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропонується використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості, проведених відділом охорони праці.

Для робочої зони виробничих приміщень встановлюються оптимальні та допустимі мікрокліматичні умови з урахуванням важкості виконуваної роботи та періоду року. При одночасному виконанні в робочій зоні робіт різної категорії важкості рівні показників мікроклімату повинні встановлюватись з урахуванням найбільш чисельної групи працівників.

Як міри для зниження шуму можна запропонувати:

– облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);

– екранування робочого місця (встановленням перегородок, діафрагм);

– установка в комп'ютерних приміщеннях устаткування, що утворює

мінімальний шум;

– раціональне планування приміщення.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

7.5 Висновки до розділу

У даному розділі магістерської роботи були виконано аналіз умов праці користувачів ПК, які працюють у зазначеному приміщенні Проведено перевірка організації робочого місця із відповідними замірами параметрів мікроклімату, освітлення, рівня шуму та розрахунком рівня шуму.

Розроблені заходи щодо поліпшення умов праці дотримання техніки безпеки та проведення протипожежної профілактики дозволить створити умови, які будуть забезпечувати більш комфортну роботу.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– Досліджена система поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

– На основі отриманих результатів досліджень створена програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кривка А.С. Дослідження та програмна реалізація системи поведінкового аналізу та обмеження активності додатків на основі моніторингу системних викликів // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.

2. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.

3. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.

4. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

5. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

6. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

7. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

8. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

9. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

10. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

11. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, т. 2 № 34, С. 677-686.

12. Вінтенко Б.Ю., Смірнова Т.В., Миронець І.В., Смірнов О.А., Буравченко К.О. «Метод оцінки функціональної стійкості комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу АЕС». *Системи управління, навігації та зв'язку,* 2026, вип. 2(84), С. 62-69.

13. Смірнова Т.В., Усік П.С., Лисенко І.А., Буравченко К.О., Смірнов О.А. «Методологія підтримки технологічних процесів у критичній інфраструктурі з забезпеченням безпеки інформації на основі хмарних технологій». *Безпека інформації.* 2025. Том 31 № 1. С. 49-60.

14. Смірнова Т.В., Буравченко К.О., Добринчук О.А., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Хмарна технологія моніторингу ключових індикаторів ефективності технологічних процесів у критичній інфраструктурі». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. № 12(43), ч. II. С. 36-54.

15. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427.

16. Vintenko, B., Myronets, I., Smirnov, O., Smirnova, T. «Enhancing the Reliability of Input Data for Nuclear Power Plant Operator Support Systems». *International Conference on Sensors, Devices and Systems. SDaS 2025. Lecture Notes in Electrical Engineering,* 2026. vol 1617. pp 3-12/

17. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications,* 2026, pp. 587-633.

18. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

19. Chulinda L., Smirnov O., Shapenko L., Ustynova I., Bohatiuk I., Kelyp S. «The role of innovation in ensuring the safety of international civil aviation». *Ceur*

Workshop Proceedings, 2025, 4024, pp. 530–542.

20. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр "Кафедра", 2025.– 320 с.

21. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

22. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

23. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

24. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

25. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

26. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I.

«Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

27. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

28. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

29. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–188.

30. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

31. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

32. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

33. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

34. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T.

«Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings,* 2023, 3550, pp. 313-320.

35. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

36. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

37. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings,* Volume 3504, 2023, pp. 1-11.

38. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

39. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об’єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №3(19), 2023, С. 176-196.

40. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *ІІ Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

41. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична*

конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

42. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

43. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

44. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings,* Volume 3187, 2022,

45. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

46. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

47. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

48. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов

С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

49. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

50. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021,* Cracow, Poland, 22-25 September 2021. P. 414-418

51. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

52. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

53. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-58.*

54. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

55. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary

constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

56. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

57. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

58. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.