Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
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 апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних”

Виконав здобувач вищої освіти
II курсу, групи КІ-25М
ОПП «Комп’ютерна інженерія»
спеціальності F7 «Комп’ютерна інженерія»
_____ Устинович М.Є.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
«____» _____ 2026 р.

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

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Устинович Максим Євгенович
Сертифікат: 514B5C86A1E5DA11040000003AB0CE001B121505
Дійсний з 18.12.2025 13:29:07 по 18.12.2026 13:29:06

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _F_____*"Інформаційні технології"*_____

Спеціальність F7 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

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

Устиновичу Максиму Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних*

2. Керівник роботи *Смірнов Сергій Анатолійович, кандидат технічних наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 41-13 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 апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних*

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

Устинович М.Є. Дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних. F7 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Метою розробки є дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Об'єктом дослідження є процес апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Предметом дослідження є методи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Методи дослідження базуються на методах управління базами даних, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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 Visual C#.

Ключові слова: комп'ютерна інженерія, моніторинг активності (DAM), аудит безпеки, мережеві бази даних

ABSTRACT

Ustynovych M.Ye. Research and Software Implementation of a Database Activity Monitoring (DAM) and Security Auditing System for Networked Databases. F7 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This final qualification thesis, submitted for the second (master's) level of higher education, presents the development of software designed for a Database Activity Monitoring (DAM) and security auditing system for networked databases.

The objective of this project is the research and software implementation of a Database Activity Monitoring (DAM) and security auditing system for networked databases.

The object of the research is the process of Database Activity Monitoring (DAM) and security auditing for networked databases.

The subject of the research consists of the methods used for Database Activity Monitoring (DAM) and security auditing of networked databases.

The research methodology draws upon database management techniques, mathematical statistics, and software development practices.

The outcome of the work is the software implementation of a Database Activity Monitoring (DAM) and security auditing system for networked databases.

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 operating the software tools are provided. The program can be used on PCs running the Windows 10/11 operating system.

The program was developed in the Visual C# environment.

Keywords: computer engineering, activity monitoring (DAM), security auditing, network databases

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	87	–	
2	А4	Робоча програма	18	–	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 13
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 15
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 15
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 24
2.3 Розгорнута постановка завдання 26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 28
3.1 Опис функціонування системи 28
3.2 Розробка структурної схеми..... 31
3.3 Розробка функціональної схеми 37
3.4 Розробка діаграми процесів 40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 42
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 63
6 НАУКОВА НОВИЗНА 68

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 69
7.1 Вступ 69
7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером 70
7.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 71
7.4 Розробка заходів з умов поліпшення охорони праці 74
7.5 Розрахункова частина 75
8 ОСНОВНІ ВИСНОВКИ 78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

КСЗ	–	комплексна система захисту
ПЕОМ	–	персональна електронно-обчислювальна машина
СУБД	–	система управління базами даних
PGP	–	Pretty Good Privacy

ВСТУП

Актуальність теми. Кожне сучасне підприємство працює на власних базах даних. Записи клієнтів, фінансові транзакції, історії пацієнтів, інтелектуальна власність – майже кожен актив, який має значення для бізнесу, зберігається на певному рівні в реляційній базі даних. І все ж у більшості організацій система безпеки навколо цих баз даних вражаюче низька. Моніторинг активності баз даних (DAM) заповнює цю прогалину. Моніторинг активності бази даних (DAM) – це практика безперервного спостереження, запису та аналізу кожної дії, що виконується всередині бази даних: запитів, змін схеми, входів у систему, надання привілеїв та експорту даних, щоб підозрілу активність можна було виявити, дослідити та зупинити до того, як вона стане порушенням безпеки. Найпростіший спосіб уявити DAM – це систему камер безпеки для ваших баз даних. Щоразу, коли хтось отримує доступ до даних, змінює їх або видаляє, платформа DAM записує це, аналізує на відповідність усталеним поведінковим моделям і подає сигнал тривоги, якщо щось виглядає не так. Платформа DAM класу 2026 йде далі: вона не просто фіксує активність, а розуміє її за допомогою поведінкових базових показників та оцінки ризиків на основі штучного інтелекту, щоб відрізнити планове технічне обслуговування від реальної загрози.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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

– Огляд існуючих систем апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– Дослідження системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– Програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Об'єктом дослідження є процес апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Предметом дослідження є методи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Методи дослідження базуються на методах управління базами даних, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– Розроблено вітчизняний продукт апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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 апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних, 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

Більшість реляційних баз даних Oracle, Microsoft SQL Server, PostgreSQL, MySQL постачаються з певною формою вбудованого журналювання аудиту. Тож навіщо організаціям окрема платформа DAM?

Вбудовані журнали аудиту мають три основні обмеження:

1. Вони не пов'язані між собою. Кожен тип бази даних створює власний формат аудиту, у своєму власному місці та зберігається протягом свого власного часового вікна. Команда безпеки, відповідальна за сотні баз даних від різних постачальників, не має єдиного уявлення.

2. Вони пасивні. Файл журналу не сповіщає вас, коли старший адміністратор баз даних раптово експортує всю таблицю клієнтів о 2-й годині ночі. Хтось має читати журнали в режимі реального часу та знати, на що звертати увагу.

3. Їм бракує контексту. Запис у журналі говорить: «користувач X виконав SELECT для таблиці Y у момент часу Z». Там не сказано: «цей користувач ніколи раніше не звертався до цієї таблиці, і це статистично аномально».

Платформа DAM існує для вирішення всіх трьох проблем: вона об'єднує дані аудиту з усіх типів баз даних в єдине нормалізоване представлення, оцінює кожну подію в режимі реального часу на відповідність правилам сповіщень та базовим рівням поведінки, а також надає контекст, який визначає, чи є подія бази даних рутинною чи небезпечною.

1.2 Область застосування

Основні можливості сучасної платформи DAM:

1. Збір даних аудиту кількох баз даних у режимі реального часу. Сучасна платформа DAM одночасно підключається до Oracle, MSSQL, PostgreSQL та MySQL і збирає аудиторські записи з налаштованими інтервалами, навіть кожні 10 секунд у середовищах з високою чутливістю. Кожен запит, вхід та зміна схеми фіксуються та зберігаються у нормалізованому форматі з можливістю пошуку.

2. Виявлення конфіденційних даних на основі штучного інтелекту. Ви не можете захистити те, про існування чого не знаєте. Найбільша прогалина в більшості програм безпеки баз даних полягає в тому, що просто невідомо, які стовпці містять особисту ідентифікаційну інформацію (PII), захищену медичну інформацію (PHI), фінансові дані або секрети автентифікації. Сучасні платформи DAM передають схему вашої бази даних до моделі великої мови програмування (LLM), яка автоматично класифікує конфіденційні стовпці, навіть якщо назви стовпців незрозумілі, наприклад USR_EML_V2, або TX_AMT_GROSS.

3. Поведінкова аналітика та оцінка ризиків. Статичне правило, таке як «сповіщення про доступ після робочого часу», спрацьовує для кожного законного завдання з технічного обслуговування, яке виконується вночі. Поведінкова аналітика відрізняється: вона створює базовий рівень нормальної поведінки для кожного користувача, а потім оцінює кожну нову дію відносно цього базового рівня. Якщо певний користувач жодного разу не звертався до цієї системи після 19:00 за 200 сеансів історії, платформа позначає це. Якщо адміністратор бази даних, який ніколи не запускав аудит DROP TABLEпротягом двох років, раптово видає його, платформа позначає це. Таким чином, DAM переходить від журналів аудиту до точного виявлення внутрішніх загроз.

4. Інтелектуальні сповіщення та звіти про відповідність вимогам у форматі PDF. Сповіщення – це не просто записи на панелі інструментів.

Платформа DAM надсилає сповіщення електронною поштою у форматі HTML щойно правило збігається з правилами, а також генерує звіти у форматі PDF, готові до відповідності, за розкладом. Це важливо для регульованих галузей: банківської справи, охорони здоров'я, урядового сектору, фінтех-технологій.

Будь-яка організація, яка зберігає регульовані, фінансові або конфіденційні дані клієнтів, повинна використовувати DAM. Найкращі варіанти відповідності включають:

– Банківські та фінансові послуги. Контролюйте, хто має доступ до фінансових записів клієнтів після робочого часу. Будь-який доступ до даних про транзакції поза звичайним робочим часом через незаплановану пакетну обробку є негайною аномалією.

– Постачальники медичних послуг. Переконайтеся, що медичні дані пацієнтів (PHI) не експортуються масово. Адміністратор баз даних, який виконує великі SELECT *запити до таблиць пацієнтів о 23:00, – це саме той сигнал, який висвітлює DAM.

– Платформи електронної комерції. Виявлення та відновлення після випадкового видалення розробником критичної робочої таблиці. Функція Kill Switch активується негайно та блокує колектор, перш ніж можна буде торкнутися журналу аудиту.

– Уряд та державний сектор. Ведення незмінних журналів аудиту з часовими мітками для регуляторних перевірок із записом повного SQL-тексту для кожної події.

– SaaS-компанії. Продемонструйте корпоративним клієнтам та аудиторам SOC 2, що привілейований доступ до бази даних контролюється, а аномалії виявляються та реагуються.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних, 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 DAM, орієнтовані на безпеку

Ці інструменти спеціально розроблені для забезпечення безпеки баз даних, таких як контроль доступу, виявлення загроз, захист даних та аудит відповідності на рівні запитів.

Cloudanix DAM

Що він робить: Cloudanix DAM працює як проксі-сервер запитів, розташований між користувачами та базами даних. Кожен SQL-запит проходить через багатоетапний конвеєр: SQL-аналіз (AST-парсинг з походженням стовпців), оцінка політик, застосування (блокування/дозвіл/перезапис), виконання та маскування після виконання. Небезпечні запити блокуються до того, як вони досягнуть бази даних, і не виявляються після виконання.

Ключові можливості:

– Превентивне блокування запитів: умови DROP, TRUNCATE, DELETE без WHERE та тавтологічні умови (1=1) перехоплюються під час виконання перевірки та зупиняються перед виконанням.

– Динамічне маскування ідентифікаційних даних: кілька стратегій маскування, що застосовуються після виконання, на основі ідентифікаційних даних користувача та класифікації даних. Інженери налагоджують виробничі проблеми, не бачачи ідентифікаційних даних клієнта.

– Відстеження походження стовпців: механізм аналізу SQL відстежує стовпці через CTE, підзапити, JOIN, псевдоніми та функції до їхньої вихідної таблиці. Маскування не можна обійти, обгортаючи стовпці виразами.

– Вбудований JIT-доступ: сеанси бази даних з обмеженим часом,

робочими процесами затвердження Slack/Teams та автоматичним закінченням терміну дії облікових даних.

– 100% суверенітет даних: Розгортається як контейнерний компонент у власному хмарному середовищі клієнта (ECS). Журнали аудиту зберігаються у S3, що належить клієнту. Вихід даних до інфраструктури Cloudanix відсутня.

– Безключовий доступ до бази даних: інженери підключаються через свою рідну IDE (DBeaver, DataGrip, TablePlus, pgAdmin), не знаючи пароля бази даних.

Підтримувані бази даних: PostgreSQL, MySQL, MSSQL (всі три бази даних готові до роботи)

Розгортання: Контейнеризоване, розгорнуте в хмарному обліковому записі клієнта. Налаштування займає 30 хвилин.

Найкраще підходить для: хмарних команд середнього бізнесу (200–2000 співробітників), яким потрібна безпека та відповідність вимогам без витоку даних постачальників. Особливо підходить для організацій із вимогами HIPAA, PCI DSS, SOC 2 або DPDPA, які також хочуть підтримувати швидкість розробки.

Зауваження: Наразі підтримується три діалекти баз даних. Організаціям, які використовують Oracle, MongoDB або Snowflake як основну базу даних, може знадобитися підтвердити охоплення дорожньою картою.

IBM Guardium

Що вона робить: IBM Guardium – це платформа безпеки корпоративних даних, яка працює на ринку DAM вже понад десять років. Вона забезпечує моніторинг активності баз даних, оцінку вразливостей, виявлення та класифікацію даних, а також управління життєвим циклом ключів шифрування. Guardium підтримує як агентний, так і безагентний моніторинг у локальних, гібридних та багатохмарних середовищах.

Ключові можливості:

– Широке охоплення платформ: Підтримує понад 100 джерел даних, включаючи реляційні бази даних, платформи великих даних, файлові системи та

мейнфрейми

– Застосування політик у режимі реального часу: блокування несанкціонованого доступу та сповіщення про порушення політик

– Виявлення та класифікація даних: знаходить та класифікує конфіденційні дані по всій спадщині

– Оцінка вразливостей: сканує бази даних на наявність відомих вразливостей та неправильних конфігурацій

– Guardium DDR (Виявлення даних та реагування): Новіша можливість для пріоритетного виявлення загроз з автоматичним реагуванням.

– Крипто-спритність: IBM Cryptography Manager для постквантової готовності та управління ключами шифрування

Підтримувані бази даних: понад 100 джерел даних, включаючи Oracle, DB2, SQL Server, PostgreSQL, MySQL, MongoDB, Hadoop, Teradata, бази даних мейнфреймів та хмарні сервіси

Розгортання: на основі агентів (агенти S-TAP на серверах баз даних), безагентне для деяких хмарних джерел, локальні пристрої. Зазвичай вимагає виділеної інфраструктури та навчених адміністраторів.

Найкраще підходить для: великих підприємств з гібридними та мейнфрейм-сервісами, існуючими інвестиціями в екосистему IBM та широкими вимогами до відповідності, що охоплюють як локальні, так і хмарні системи. Організацій зі спеціалізованими командами безпеки баз даних, які можуть керувати операційною складністю.

Міркування: Складні накладні витрати на розгортання та управління. Терміни впровадження зазвичай вимірюються місяцями, а не днями. Зона впливу агентів на сервери баз даних додає операційного навантаження. Ціноутворення відповідає корпоративному рівням. Платформа значно розвинулася; деякі клієнти повідомляють про фрагментацію функцій у сімействі продуктів Guardium (захист даних, DDR, оцінка вразливостей, Key Lifecycle Manager – це окремі продукти).

Imperva Data Security Fabric

Що вона робить: Imperva (тепер частина Thales) пропонує Data Security Fabric (DSF) – широку платформу безпеки даних, яка включає моніторинг активності бази даних як один із компонентів, а також виявлення даних, класифікацію, маскування, шифрування та аналітику ризиків. Вона прагне стати багатохмарним рішенням корпоративного масштабу для захисту структурованих та неструктурованих даних.

Ключові можливості:

– Моніторинг активності даних: безперервний моніторинг та аудит у сховищах даних

– Аналіз ризиків даних: Поведінковий аналіз для виявлення невідповідних або шкідливих моделей доступу

– Виявлення та класифікація даних: Автоматизоване виявлення конфіденційних даних у різних середовищах

– Маскування статичних даних: зміна конфіденційних даних для невиробничих середовищ

– Інтеграція екосистеми: підключення до SIEM, системи продажу квитків та хмарних сервісів

– Широке охоплення баз даних: Підтримка реляційних баз даних, NoSQL, великих даних, хмарних сервісів та файлових сховищ

Підтримувані бази даних: Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, Amazon RDS, Azure SQL, Google Cloud SQL, Snowflake та багато інших

Розгортання: Моніторинг на основі агентів за допомогою шлюзових пристроїв. Хмарне та локальне використання. Потрібна виділена інфраструктура.

Найкраще підходить для: великих підприємств, яким потрібна повноцінна платформа безпеки даних, що виходить за рамки DAM та охоплює виявлення, класифікацію, маскування та шифрування даних. Організацій у сфері фінансових послуг (6 з 10 провідних телекомунікаційних компаній, 4 з 5

провідних банків США є клієнтами) та охорони здоров'я зі складними вимогами до дотримання вимог, що охоплюють різні типи даних.

Міркування: Ціноутворення на рівні підприємства та складність впровадження. Платформа зросла завдяки придбанню (тепер є частиною групи Thales), що може означати навігацію по кількох лінійках продуктів. В першу чергу, це модель виявлення та оповіщення, а не превентивне блокування на рівні запитів. Для повної видимості потрібне розгортання агента на серверах баз даних.

Varonis for Databases

Що він робить: Varonis забезпечує керування безпекою даних за допомогою універсального конектора бази даних, який інтегрується з мережевими базами даних для виявлення даних, класифікації та моніторингу активності. Він зосереджений на пошуку конфіденційних даних, розумінні того, хто має до них доступ, та виявленні підозрілої поведінки за допомогою UEBA (аналітики поведінки користувачів та сутностей).

Ключові можливості:

– Універсальний з'єднувач бази даних: інтегрується з будь-якою мережевою базою даних для виявлення та класифікації

– Виявлення та класифікація даних: класифікація на основі штучного інтелекту та правил у великих масштабах у керованих та некерованих базах даних

– Моніторинг активності бази даних: відстежує шаблони доступу та веде журнал аудиту, зручний для читання людиною.

– Динамічне маскування даних: автоматично застосовує маски для захисту критично важливої інформації від несанкціонованого доступу.

– UEBA: Виявляє аномальну поведінку користувачів, включаючи незвичайні шаблони запитів та час доступу

– Управління станом: Виявляє неправильні конфігурації, такі як загальнодоступні порти та незашифровані резервні копії

Підтримувані бази даних: керовані та некеровані бази даних у хмарі та

локально через універсальний конектор. AWS RDS, Azure SQL та інші хмарні бази даних.

Розгортання: SaaS-проектування з агентами для підключення до джерел даних. Покриття хмарних та локальних ресурсів.

Найкраще для: організацій, яким потрібні можливості керування станом безпеки даних (DSPM) з охопленням баз даних як частина ширшої стратегії захисту даних. Особливо ефективний для середовищ із тіньовими базами даних, некерованими екземплярами баз даних та складними вимогами до управління доступом.

Міркування: Основна сильна сторона Varonis – це неструктуровані дані (спільні файлові ресурси, M365, SaaS); охоплення баз даних розширюється, але може не відповідати глибині спеціалізованих інструментів DAM для блокування на рівні запитів або маскування в режимі реального часу. Платформа більше зосереджена на DSPM та управлінні, ніж на превентивному контролі запитів.

Інструменти моніторингу баз даних, орієнтовані на продуктивність

Ці інструменти зосереджені на оптимізації продуктивності та спостережливості бази даних. Вони не є інструментами DAM безпеки, вони не блокують запити, не маскують дані та не створюють журнали аудиту відповідності. Вони включені сюди, оскільки користувачі, які шукають «моніторинг активності бази даних», часто стикаються з ними та повинні розуміти різницю.

Якщо вашою основною потребою є продуктивність програм, оптимізація запитів або спостереження за інфраструктурою, ці інструменти чудово підійдуть. Якщо ж вам потрібна безпека, відповідність вимогам або захист даних, перегляньте розділ Security DAM вище.

Datadog Database Monitoring

Що вона робить: Datadog DBM є частиною платформи спостереження Datadog. Вона забезпечує видимість продуктивності бази даних у режимі реального часу, допомагаючи розробникам додатків та адміністраторам баз даних вирішувати проблеми з повільними запитами, виявляти конфлікти

ресурсів та оптимізувати ефективність бази даних. Вона об'єднує телеметрію бази даних, додатків та інфраструктури в одній платформі.

Ключові можливості:

– Аналіз продуктивності запитів: відстеження середньої затримки, часу виконання, сканованих рядків та подій очікування

– Аналіз першопричин за допомогою штучного інтелекту: Автоматизована діагностика регресій затримки запитів

– Кореляція APM: Підключення повільних запитів до бази даних до уражених служб додатків за допомогою розподіленого трасування

– Візуалізація блокувальних запитів: Визначення ланцюжків блокувань та кореневих блокувальних запитів

– Рекомендації щодо оптимізації: Автоматизовані пропозиції щодо перезапису запитів, додавання індексів та оновлення схеми

Підтримувані бази даних: PostgreSQL, MySQL, SQL Server, Oracle, MongoDB, Amazon RDS, Aurora, Azure SQL, Google CloudSQL, Supabase, ClickHouse

Найкраще підходить для: команд DevOps/SRE та розробників додатків, яким потрібно оптимізувати продуктивність запитів та вирішувати проблеми з вузькими місцями в базі даних. Особливо ефективний, якщо ви вже працюєте в екосистемі Datadog та хочете мати єдину спостережуваність на рівнях додатків, інфраструктури та бази даних.

Не підходить для: дотримання вимог безпеки, маскування даних, блокування запитів, контролю доступу або створення журналу аудиту для нормативних вимог.

SolarWinds Database Observability

Що він робить: SolarWinds пропонує інструменти спостереження за базами даних, спрямовані на моніторинг, діагностику та оптимізацію продуктивності бази даних. Доступний як у власному розпорядженні (Database Performance Analyzer), так і в SaaS (SolarWinds Observability), він створений в першу чергу для адміністраторів баз даних, які керують критично важливою

інфраструктурою баз даних.

Ключові можливості:

– Аналіз часу очікування: Визначення запитів на ресурси, на які вони очікують

– Міжплатформний моніторинг: підтримка SQL Server, Oracle, PostgreSQL, MySQL та хмарних баз даних

– Виявлення аномалій: Визначення відхилень продуктивності від базових показників

– Консультанти запитів: рекомендації експертів щодо налаштування таблиць та оптимізації запитів

– Власне розміщення або SaaS: Гнучкі варіанти розгортання

Підтримувані бази даних: SQL Server (найглибша підтримка через SQL Sentry), Oracle, PostgreSQL, MySQL, MariaDB, Amazon RDS, Aurora, Azure SQL

Найкраще підходить для: команд адміністраторів баз даних у традиційних підприємствах, яким потрібен спеціалізований моніторинг продуктивності бази даних із глибокими можливостями діагностики. Особливо ефективний для середовищ SQL Server.

Не підходить для: дотримання вимог безпеки, маскування даних, блокування запитів або контролю доступу. Це інструмент оптимізації продуктивності.

Middleware Database Monitoring

Що воно робить: Middleware забезпечує єдину спостережуваність з моніторингом бази даних як частину повноцінної платформи, що охоплює APM, журнали, інфраструктуру та моніторинг реальних користувачів. Воно зосереджено на видимості продуктивності запитів у хмарному, локальному та Kubernetes середовищах.

Ключові можливості:

– Аналіз запитів у режимі реального часу: моніторинг SQL-запитів у режимі реального часу в різних середовищах баз даних

– Оптимізація запитів: Визначення довготривалих запитів за допомогою

практичних рекомендацій

– Підтримка різних постачальників: Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, Cassandra, Couchbase та хмарні бази даних

– Повна кореляція: співвіднесення показників бази даних із відповідними трасуваннями та журналами APM

– Усунення несправностей на основі штучного інтелекту: OpsAI для автоматизованого аналізу першопричин

Підтримувані бази даних: Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, Cassandra, Couchbase, Amazon Aurora, Google Cloud SQL, Azure SQL Database

Найкраще для: Невеликих команд, яким потрібен доступний повноцінний моніторинг із видимістю бази даних як частина єдиної платформи. Гарний варіант для команд, яким потрібен моніторинг бази даних разом із APM, керуванням журналами та моніторингом інфраструктури без ціноутворення корпоративного рівня.

Не підходить для: дотримання вимог безпеки, маскування даних або контролю доступу на рівні запитів.

Checkmk Database Monitoring

Що він робить: Checkmk – це рішення з відкритим кодом для моніторингу інфраструктури, яке включає можливості моніторингу баз даних. Воно забезпечує перевірку доступності, показники продуктивності та моніторинг стану серверів баз даних як частину ширшого моніторингу IT-інфраструктури.

Ключові можливості:

– Автоматичне виявлення: автоматично виявляє та контролює бази даних без ручного налаштування.

– Широка підтримка баз даних: Oracle, MSSQL, MySQL/MariaDB, PostgreSQL, SAP HANA, MongoDB, IBM DB2

– Попередньо налаштовані пороги: готовий до використання моніторинг із розумними налаштуваннями за замовчуванням

– Користувацькі запити: Активний моніторинг за допомогою користувацьких SQL-запитів

– Локальні та хмарні: моніторинг як локальних баз даних, так і екземплярів, керованих хмарою (Amazon RDS, Azure SQL)

Підтримувані бази даних: Oracle (від 10.2), MSSQL, MySQL/MariaDB, PostgreSQL, SAP HANA, MongoDB, IBM DB2, IBM Informix

Найкраще підходить для: команд системних адміністраторів, що працюють з локальною інфраструктурою та потребують спрощеного моніторингу стану бази даних як частини ширшого моніторингу інфраструктури. Організацій, які шукають моніторинг з відкритим кодом на власному хостингу без прив'язки до постачальника SaaS.

Не підходить для: дотримання вимог безпеки, маскування даних, блокування запитів або будь-якого іншого випадку використання DAM, орієнтованого на безпеку. Це моніторинг стану інфраструктури.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення апаратно-програмного комплексу моніторингу активності та аудиту безпеки мережевих баз даних (DAM) реалізовано мовою програмування Visual C#. Вибір цієї мови зумовлений її архітектурними перевагами: Visual C# є суворо типізованою об’єктно-орієнтованою мовою, розробленою для створення захищених, масштабованих та високоефективних застосунків у середовищі виконання .NET. Наявність потужних інструментів розробки, розвиненого середовища та вбудованих бібліотек значно спрощує проектування складних багатопотокових систем, що взаємодіють із реляційними та нереляційними базами даних.

Особливості мови та її застосування в DAM-системах

Синтаксис Visual C# є інтуїтивно зрозумілим і виразним, успадковуючи найкращі практики сімейства мов C/C++. Водночас він пропонує розширені

інструменти для роботи з даними, що критично важливо для парсингу SQL-запитів, аналізу мережевого трафіку та обробки логів аудиту:

– LINQ (Language-Integrated Query): забезпечує можливість виконання суворо типізованих запитів безпосередньо до колекцій та джерел даних, що підвищує читабельність і надійність коду обробки подій безпеки.

– Делегати та події: дозволяють реалізувати ефективний асинхронний механізм сповіщень про критичні інциденти та несанкціоновані спроби доступу до баз даних.

– Об’єктно-орієнтована парадигма: підтримка інкапсуляції, спадкування та поліморфізму дає змогу побудувати гнучку модульну архітектуру комплексу, де кожен компонент (модуль перехоплення трафіку, аналізатор правил, модуль звітності) чітко розмежований.

– Механізми взаємодії (Interop) та низькорівневий контроль: за потреби Visual C# дозволяє інтегруватися з нативними бібліотеками (DLL) та об'єктами COM, а також використовувати безпечні й «небезпечні» (unsafe) блоки коду з прямим доступом до пам'яті, що є важливим для оптимізації швидкодії високо навантажених мережевих зондів.

Архітектура середовища виконання .NET

Програмні модулі DAM-системи виконуються в середовищі .NET, яке містить віртуальну систему виконання (CLR – Common Language Runtime) та багатий уніфікований набір бібліотек класів.

Вихідний код Visual C# компілюється в проміжну мову (IL – Intermediate Language), яка під час запусків додатку перетворюється на машинний код за допомогою механізму JIT-компіляції (Just-In-Time). Середовище CLR забезпечує виконання низки критично важливих служб:

– Кероване середовище виконання (Managed Code): автоматичне керування пам'яттю (збирання сміття – Garbage Collection) і надійна обробка виключень мінімізують ризики витоків пам'яті під час безперервного цілодобового моніторингу активності баз даних.

– Безпека типів і пам'яті: сувора типізація запобігає багатьом поширеним

вразливостям на рівні програмного забезпечення (наприклад, переповнення буфера), що є критичним для систем класу Security/DAM.

– Масштабованість та інтеграція: завдяки відповідності специфікації CLI, розроблені компоненти можуть легко взаємодіяти з іншими модулями, написаними різними мовами платформи .NET.

Використання мови Visual C# та платформи .NET забезпечує створення надійного, продуктивного, захищеного та легкого в підтримці програмного ядра апаратно-програмного комплексу моніторингу і аудиту безпеки мережевих баз даних.

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 апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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

DAM, PAM та SIEM є взаємодоповнюючими, а не конкуруючими рівнями програми безпеки підприємства:

– Керування привілейованим доступом (PAM) контролює, хто може підключатися до систем і що їм дозволено робити під час сеансу.

– DAM записує та аналізує те, що насправді відбувається в базі даних під час та після цього сеансу.

– SIEM співвідносить сигнали з усього вашого середовища DAM, PAM, EDR, брандмауера, IdP з поверхневими інцидентами на рівні організації.

DAM – це рівень, який бачить саму базу даних. PAM контролює вхід; DAM спостерігає за тим, що відбувається, коли хтось опиняється всередині.

Під час складання короткого списку постачальників послуг моніторингу активності баз даних найважливішими питаннями є:

1. Підтримка кількох баз даних. Чи може він моніторити Oracle, MSSQL, PostgreSQL та MySQL з однієї консолі, чи для кожної з них потрібен окремий продукт?

2. Виявлення даних на основі штучного інтелекту. Чи автоматично класифікує він особисті дані, захищені медичні дані, фінансові дані та дані автентифікації за допомогою справжнього LLM, чи покладається на шаблони регулярних виразів, що підтримуються вручну?

3. Поведінкова аналітика. Чи будує вона базові показники та відхилення оцінок для кожного користувача, чи просто відповідає статичним правилам?

4. Оповіщення в режимі реального часу. Чи надсилає система критичні оповіщення в момент їх спрацьовування, чи чекає на наступний пакетний звіт?

5. Готовність до проведення судово-медичної експертизи. Чи існує вбудований механізм для збереження доказів у момент виявлення руйнівної

події?

6. Гнучкість розгортання. Чи може воно працювати локально для забезпечення суверенітету даних, у хмарі або в обох варіантах? Чи може постачальник штучного інтелекту бути самостійно розміщеним (наприклад, Ollama) для середовищ, де дані схеми не можуть залишити мережу?

Моніторинг активності бази даних у 2026 році – це не «приємна річ», це рівень видимості, від якого залежить усе інше. Відповідність вимогам, виявлення внутрішніх загроз, судово-медична експертиза порушень та забезпечення нульової довіри – все це передбачає, що хтось або щось спостерігає за базою даних. Сучасна платформа DAM на базі штучного інтелекту робить це можливим у масштабі підприємства.

Якщо ви оцінюєте платформи DAM, зверніть увагу на SWOT DAM 3.0 від Rudratic, керовану штучним інтелектом, створену для Oracle, MSSQL, PostgreSQL та MySQL, з функцією виявлення даних за допомогою штучного інтелекту, поведінкової аналітики та судово-медичного збереження даних за допомогою Kill Switch в одній уніфікованій консолі. Або замовте демонстрацію у нашої команди.

Вбудований аудит баз даних (вбудований в Oracle, SQL Server, PostgreSQL) реєструє події в тій самій базі даних, яку він контролює, тобто привілейований адміністратор баз даних може змінювати або видаляти журнал аудиту. DAM надсилає потоки аудиту на окрему, незалежну платформу, до якої адміністратор баз даних не має доступу. DAM також забезпечує сповіщення в режимі реального часу та виявлення аномалій у поведінці, до яких не мають вбудованих журналів аудиту. Для цілей відповідності вимогам аудиторам потрібен незалежний, захищений від несанкціонованого доступу журнал аудиту, який надає лише DAM.

Які бази даних моніторить платформа DAM?. Платформи Enterprise DAM підтримують Oracle Database, Microsoft SQL Server, PostgreSQL, MySQL, IBM Db2 та SAP HANA. Сучасні платформи також підтримують хмарні бази даних Amazon RDS, Azure SQL, Google Cloud SQL. SWOT DAM 3.0 охоплює всі

основні корпоративні та хмарні механізми баз даних з єдиної консолі моніторингу, без необхідності встановлення окремого агента для кожного типу бази даних.

Чи може DAM виявляти внутрішні загрози? Так, виявлення внутрішніх загроз є одним із основних варіантів використання DAM. Платформи DAM на базі штучного інтелекту створюють базову поведінкову модель для кожного користувача: які таблиці вони зазвичай запитують, в який час і з якими обсягами. Відхилення, коли співробітник запитує таблицю клієнтів о 2 годині ночі та експортує 50 000 рядків, викликають сповіщення в режимі реального часу. Привілейовані користувачі (адміністратори баз даних, системні адміністратори) охоплені системою, навіть якщо вони обходять елементи керування прикладним рівнем, підключаючись безпосередньо до бази даних.

Які системи відповідності вимагають моніторингу активності бази даних? Вимога PCI DSS 10 (реєстрація аудиту всіх доступів до даних власників карток), правило безпеки HIPAA §164.312(b) (контроль аудиту для ePHI), розділ 404 SOX (загальні засоби контролю IT, що вимагають доказів моніторингу доступу до фінансових даних), стаття 32 GDPR (відповідні технічні заходи безпеки) та NIST SP 800-53 AU-2 та AU-12 (реєстрація подій аудиту) – усі вони вимагають контролю, яким задовольняє належна платформа DAM.

Чи впливає DAM на продуктивність бази даних? Продукти моніторингу активності баз даних використовують різні архітектури колекцій, включаючи агенти на основі хоста та моніторинг мережі. Вплив на ресурси залежить від методу моніторингу, версії бази даних, робочого навантаження та конфігурації. Використовуйте репрезентативне тестування для оцінки накладних витрат для конкретного середовища.

DAM на базі штучного інтелекту виходить за рамки оповіщень на основі правил (які генерують втому від оповіщень через тисячі щоденних хибних спрацьовувань) і до виявлення поведінкових аномалій. Платформа вивчає звичайні шаблони запитів кожного користувача та надсилає сповіщення лише про справжні відхилення. Штучний інтелект також дозволяє автоматизовано

виявляти конфіденційні дані (класифікуючи стовпці PII та PHI без ручного позначення тегами) та оцінювати ризики, що визначає пріоритетність оповіщень за серйозністю. Результатом є значно менша кількість хибних спрацьовувань та швидше реагування на інциденти.

3.2 Розробка структурної схеми

Моніторинг активності бази даних (DAM) – це будь-яке рішення, яке активно відстежує та аналізує активність бази даних. Він є критично важливим для стратегії безпеки даних організації, допомагаючи командам виявляти несанкціонований доступ, запобігати витоку даних та дотримуватися вимог відповідності. Інструменти DAM є багатоцільовими для виявлення загроз, судово-медичних розслідувань, контролю доступу та звітності перед регуляторними органами.

Чому DAM важливий у 2026 році

Обсяг даних, що зберігаються, та складність захисту баз даних зростають експоненціально. Регуляторний контроль посилюється, а хмарні додатки повинні працювати у дедалі складніших середовищах. Бази даних тепер обслуговують різні додатки та бізнес-операції, а доступ має бути жорстко контрольований, щоб відповідати політикам та стандартам відповідності. Штучний інтелект тепер є помножувачем сили для викриття.

Сучасні інструменти знижують вартість розвідки та соціальної інженерії, водночас створюючи більше шляхів доступу до конфіденційних даних через автоматизовані агенти, конвеєри та інтеграції. Водночас організації перейшли від локальних баз даних одного постачальника до гібридних, багатохмарних та повністю керованих сервісів, де можливості вбудованого аудиту є непослідовними та часто затримуються, що виключає можливість запобігання в режимі реального часу. Регулятори відреагували суворішими правилами розкриття інформації та вищими штрафами, зробивши журнали аудиту такими, що перевіряються, незмінними, а також таблицями швидкого реагування на

основі доказів. Традиційні рішення DAM часто не справляються з роботою в хмарних середовищах, де бази даних використовуються як керовані SaaS-сервіси. Немає доступу до інфраструктури, місця для агентів і мережевого рівня для використання. Натомість DAM має перехоплювати запити на кінцевій точці даних і асинхронно пересилати їх до зовнішнього сервісу, такого як Splunk.

Огляд моніторингу активності бази даних

Більшість баз даних не реєструють всю активність за замовчуванням. Коли це робиться, ці журнали часто зберігаються всередині бази даних, що робить їх вразливими. Якщо зловмисник отримає доступ до облікових даних із правами на запис, він може просто видалити докази свого вторгнення.

Наприклад:

```
DELETE FROM audit_log WHERE user_id = 'attacker';
```

Можливість редагування журналів є причиною, чому власному веденню журналів не можна довіряти як джерелу достовірної інформації. Журнали можуть бути скорочені або змінені, що унеможливлює визначення того, чи відбулося порушення або як воно сталося. Служби безпеки не можуть проводити судово-медичні розслідування без достовірних даних про активність, повідомляти постраждалих користувачів або визначати, як зловмисники отримали доступ.

DAM вирішує цю проблему, фіксуючи та пересилаючи активність бази даних до зовнішньої, незмінної системи, такої як інструмент управління інформацією та подіями безпеки (SIEM) або судово-медична служба, гарантуючи, що журнал аудиту залишається неушкодженим та захищеним від несанкціонованого доступу. Така видимість є основоположною для дотримання вимог, таких як PCI DSS, HIPAA та SOX, а також для швидкого та точного реагування на порушення.

DAM дозволяє організаціям спостерігати за активністю бази даних у режимі реального часу, що важливо для розуміння того, як і хто отримує доступ до даних. Така видимість є основоположною для дотримання таких вимог, як PCI DSS, HIPAA та SOX, а також для швидкого та точного реагування на

порушення безпеки. Для збереження цілісності доказів, дані про активність слід записувати в незмінне сховище (наприклад, WORM/S3 Object Lock) з криптографічним хешуванням/атестацією та чіткими політиками зберігання. Це забезпечує надійний ланцюг зберігання для аудитів та розслідувань. Хоча заплановані реплікації (cron) можуть допомогти зі звітністю, вони створюють прогалини, коли передачі не вдаються, журнали очищуються або події відбуваються між циклами, що робить їх недостатніми для надійного виявлення та криміналістики.

Командам безпеки все частіше доводиться проводити судово-медичні розслідування, щоб повідомляти постраждалих користувачів та визначати, як зловмисники отримали доступ. Це спонукало багато організацій впроваджувати рішення DAM, які записують всю активність запитів до бази даних, що дозволяє проводити аналіз після розслідування та проактивно виявляти загрози.

Моделі розгортання та компроміси продуктивності

DAM можна розгорнути кількома способами, кожен з яких має свої недоліки:

Агентний моніторинг

Агентна DAM передбачає встановлення програмних агентів безпосередньо на серверах баз даних. Вона забезпечує глибокий огляд активності на рівні запитів, включаючи контекст користувача та поведінку програм. Оскільки вона працює в середовищі хоста, вона може захоплювати зашифрований трафік та підтримувати сповіщення в режимі реального часу та застосування політик. Однак вона додає обчислювальне та мережеве навантаження до кожного запиту, що може погіршити продуктивність бази даних, особливо за великих обсягів транзакцій. Вона також вимагає оновлень та перевірок сумісності з версіями баз даних та патчами ОС, а керування агентами у великому парку баз даних може бути операційно ресурсоємним.

Мережевий моніторинг

Цей метод використовує мережеві перехоплення або захоплення пакетів (PCAP) для перевірки трафіку між програмами та базами даних. Він

ненав'язливий, не вимагає встановлення програмного забезпечення на серверах баз даних і його легше розгортати в гетерогенних середовищах з централізованою архітектурою моніторингу. З недоліків, він може пропускати зашифрований трафік (наприклад, TLS), якщо не ввімкнено розшифрування SSL. Він не завжди може співвідносити запити з певними користувачами або програмами та може не розкривати внутрішню активність бази даних, таку як збережені процедури або заплановані завдання.

Запланована реплікація (наприклад, завдання Cron)

Цей підхід періодично копіює журнали або таблиці аудиту з баз даних до централізованої системи моніторингу. Він легкий та простий у впровадженні, не впливає на продуктивність запитів у реальному часі та корисний для звітності про відповідність та аналізу історії. Однак події безпеки можуть залишатися непоміченими до наступного циклу реплікації, можуть виникати прогалини в даних, якщо реплікація не вдається або журнали очищуються передчасно, і він не підходить для середовищ, що потребують негайного реагування на загрози.

Асинхронне пересилання в режимі реального часу

Асинхронно, найефективніші рішення DAM пересилають дані про активність бази даних майже в режимі реального часу. Це гарантує, що виконання запитів не затримується операціями реєстрації.

```
/**
def intercept_query(query, user_context):
  log_entry = {
   "query": query,
   "user": user_context["username"],
   "timestamp": datetime.now()
  }
  send_to_siem_async(log_entry)
  execute_query(query)
```

Ця функція Python перехоплює та реєструє запити користувачів з контекстом та міткою часу для моніторингу SIEM перед виконанням.

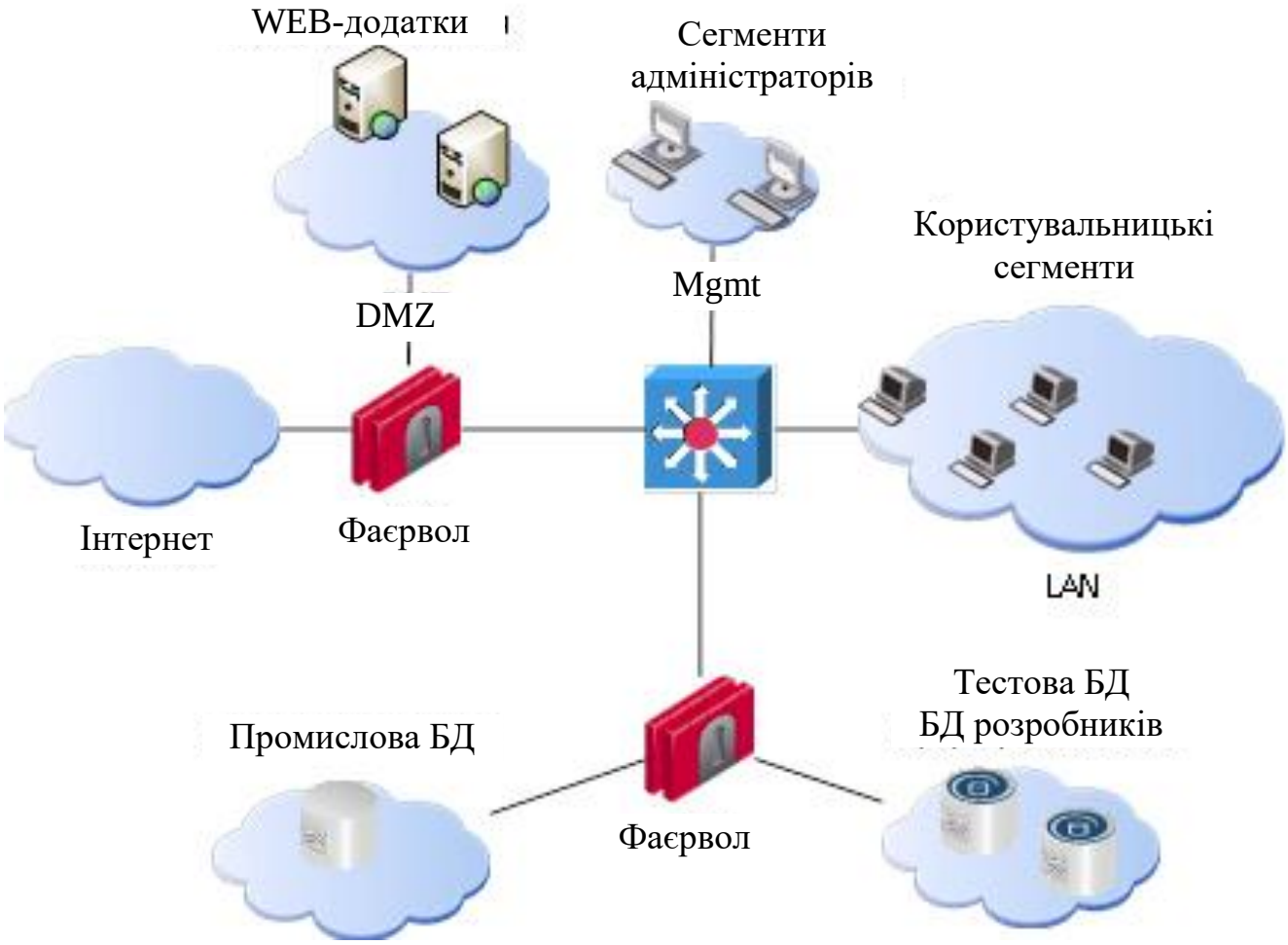

Рисунок 3.1 – Структурна схема системи

Безагентне, бездержавне перехоплення (хмарне/кероване)

Перехоплення запитів на кінцевій точці даних або через легкий вбудований проксі-сервер, боковий сервер або шлюз, який асинхронно пересилає збагачену телеметрію до аналітики або криміналістики. Він працює з керованими базами даних, де доступ до хоста або мережі недоступний, створює узгоджену збагачену телеметрію (ідентифікація, чутливість, середовище) в локальній та хмарній мережах, а також мінімально ускладнює роботу, оскільки хости баз даних не потребують встановлення чи виправлення агентів. Однак він вимагає високодоступної конструкції та незначної затримки в шляху передачі даних, повинен підтримувати гетерогенні граматики та протоколи (SQL, NoSQL,

потоки) та нормалізацію в масштабі, а також потребує чітких шаблонів для відкриття/закриття при збоях та обробки зворотного тиску.

SaaS-аналітика з приватною, легкою колекцією

Завдяки легкому колектору необроблені дані залишаються у вашому середовищі, водночас використовуючи аналітику SaaS для виявлення, кореляції та звітності. Це спрощує оновлення та аналітику, щоб команди могли зосередитися на зниженні ризиків, а не на обслуговуванні інструментів. Конфіденційність на першому місці, оскільки необроблені дані зберігаються у вашому клієнті. Водночас до площини SaaS надходять лише телеметрія та функції, що спрощує стандартизацію виявлень та панелей інструментів у багатохмарних сховищах. Однак це вимагає чітких меж виведення даних, ізоляції оренди та шифрування під час передачі та зберігання, а також має відповідати вимогам щодо місця розташування даних, суверенітету та політикам юридичного зберігання.

Моніторинг активності даних у хмарі

Застарілі інструменти DAM були створені для локальних баз даних на основі SQL. Але сучасні дані неоднорідно розподілені між SQL, NoSQL та тематичними репозиторіями. Впровадження «товстого» шару перехоплення може серйозно вплинути на продуктивність. Потрібен «тонкий» шар, який працює в режимі реального часу з незначною затримкою та підтримує сучасні граматики та протоколи баз даних.

DAM наступного покоління має нормалізувати телеметрію в різних гетерогенних сховищах – реляційних, документних, ключ-значення, стовпцевих, аналітичних сховищах та потокових/тематичних платформах – і працювати безпосередньо з керованими базами даних, де мережеві крани та агенти нежиттєздатні. Це особливо важливо в хмарних середовищах, де інфраструктура абстрагована, а традиційні методи моніторингу на основі хоста або мережі є непрактичними або неможливими.

Тонкий шар перехоплення в режимі реального часу, який додає незначну затримку та збагачує події контекстом ідентифікації та чутливості, є важливим

для збереження продуктивності та максимальної точності. У хмарних архітектурах цей підхід забезпечує сумісність з керованими сервісами, масштабується в розподілених середовищах та відповідає сучасним операційним моделям, що робить його найефективнішим та найстійкішим шляхом моніторингу активності баз даних.

3.3 Розробка функціональної схеми

Функціональну схему розробленого апаратно-програмного комплексу моніторингу активності та аудиту безпеки мережевих баз даних наведено на рисунку 3.2. Відповідно до схемного рішення, архітектура системи охоплює такі ключові функціональні модулі:

– головне інтерфейсне вікно комплексу контролю доступу та аудиту транзакцій у розподілених базах даних;

– модуль розмежування прав доступу;

– модуль управління обліковими даними (менеджер паролів);

– підсистема журналювання та аудиту подій безпеки;

– довідково-інформаційний блок;

– криптографічні модулі шифрування й дешифрування інформації на основі симетричного алгоритму *3DES*.

Детальна характеристика функціональних блоків комплексу

1. Головне вікно системи та інтерфейс взаємодії. Інтерфейсний модуль забезпечує оператору швидкий доступ до базових функцій моніторингу, панелей управління та структурованого головного меню. Програмне рішення побудоване за принципом головного та незалежних дочірніх вікон, розташування і розміри яких адаптуються користувачем. Усі зміни інтерфейсних налаштувань автоматично зберігаються для наступних сеансів роботи. Ергономіка управління підтримується системою контекстних спливаючих підказок для кожного елемента керування.

2. Блок розмежування доступу. Призначений для організації захищеного та контрольованого доступу користувачів і адміністраторів до захищених ресурсів корпоративних баз даних. Модуль реалізує такі можливості:

– забезпечення санкціонованого доступу до корпоративних інформаційних ресурсів, структурованих схем і довідників з урахуванням ролей;

– синхронізацію облікових записів та ідентифікаторів у межах контрольованого середовища;

– динамічне керування правами доступу (оперативне надання, зміна або відкликання при зміні повноважень співробітників);

– підтримку відповідності галузевим і корпоративним стандартам інформаційної безпеки.

У межах цього блоку реалізовано підсистему самообслуговування облікових записів (управління профілями, зміна реквізитів автентифікації, перевірка політик безпеки), механізми запитів і затверджень ресурсів (з можливістю накладання цифрового підпису), а також рольове керування доступом із підтримкою політик поділу обов'язків (SoD – Segregation of Duties).

3. Менеджер облікових даних (паролів). Забезпечує захищене зберігання, керування та генерацію автентифікаційних даних для множинних облікових записів у багатокористувацькому середовищі. Бази даних облікових записів ізольовано та зашифровано. Додатково модуль підтримує функцію пошуку, використання макросів, оперативне резервне копіювання баз даних, швидке перемикання сеансів користувачів та генерацію криптостійких паролів.

4. Блок журналювання подій (Audit Logging). Призначений для безперервної реєстрації всіх подій, транзакцій та дій користувачів у системі. Модуль містить підсистему архівації логів, яка дозволяє формувати захищені архіви за обраний період (із забезпеченням мінімального терміну зберігання історії дій не менше ніж один місяць). Після ініціації процедури очищення журналу система автоматично генерує зашифрований файл звіту, записує посилання для адміністратора та видаляє оброблені записи з оперативної бази даних. У разі виникнення помилок збереження архіву цілісність оперативних даних журналювання гарантовано зберігається.

5. Довідкова підсистема. Забезпечує користувача контекстною документацією щодо експлуатації комплексу, а також містить технічні дані про версію програмного забезпечення, дату випуску та розробників.

6. Блоки криптографічного захисту (3DES). Призначені для апаратного або програмного шифрування та дешифрування чутливих даних і транзакцій, до яких звертаються користувачі, відповідно до встановлених рівнів прав доступу та політик безпеки DAM-системи.

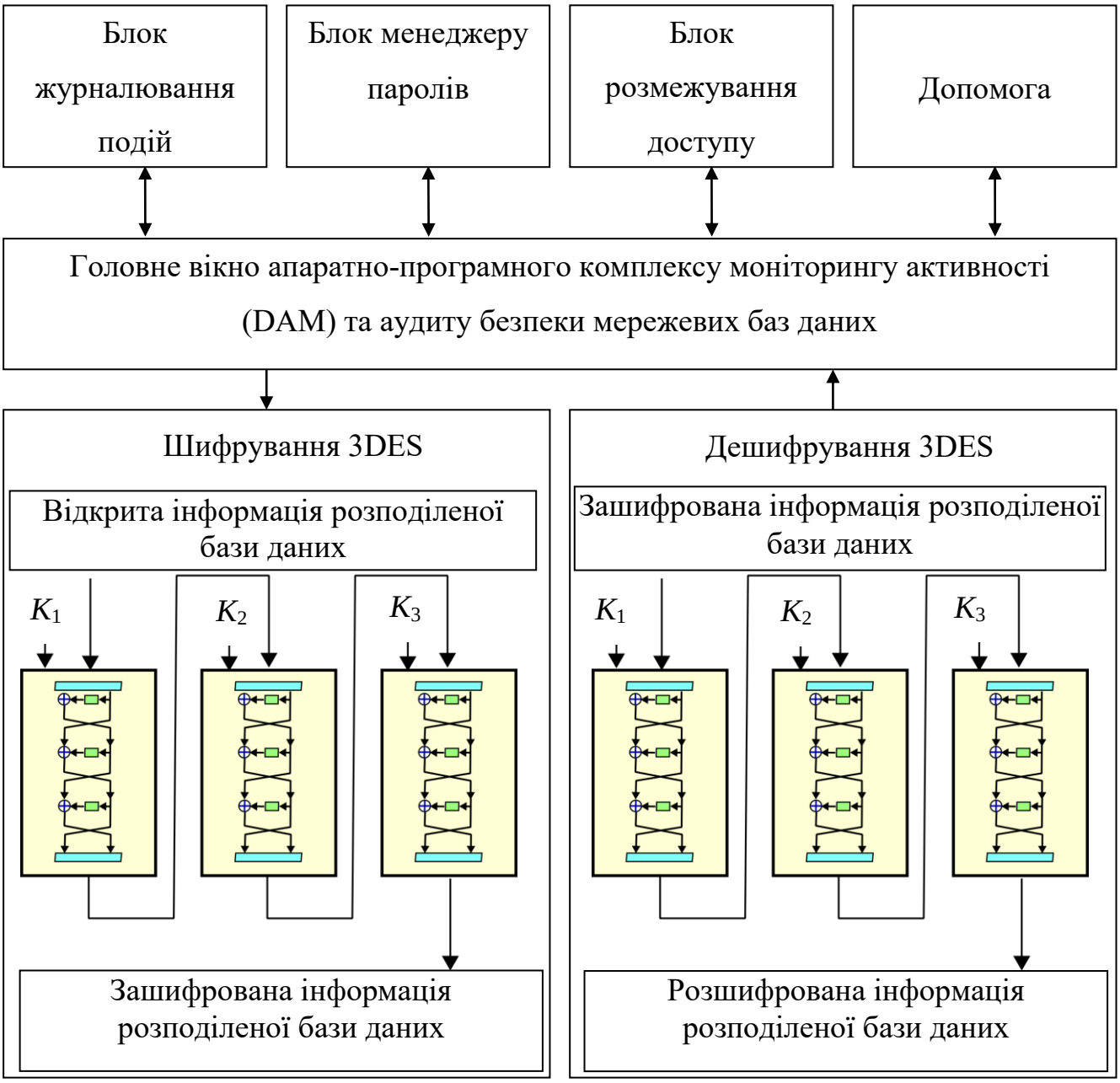

Рисунок 3.2 – Функціональна схема системи

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграму процесів розробленої системи наведено на рисунку 3.3. Її детальний аналіз дозволяє простежити логіку взаємодії внутрішніх компонентів та маршрутизацію потоків інформації в межах апаратно-програмного комплексу.

У процесі проєктування застосовується модель графічного представлення потоків даних (Data Flow Diagram, DFD), яка відображає рух і трансформацію даних в інформаційній системі.

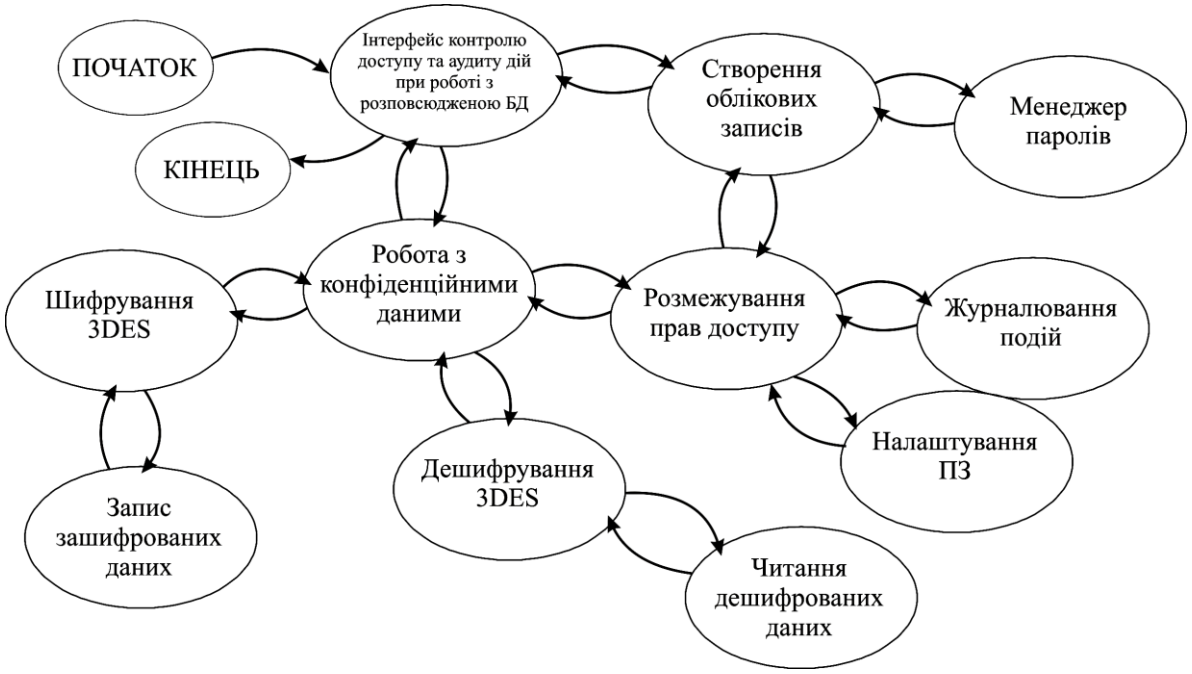

Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми взаємодії процесів є базовим інструментом структурного проєктування для візуалізації обробки даних. Традиційно розробка розпочинається з побудови контекстної діаграми потоків даних, яка ілюструє загальну взаємодію системи моніторингу та аудиту із зовнішнім середовищем.

Надалі ця модель деталізується шляхом поетапної декомпозиції процесів і потоків даних з метою повної та всебічної демонстрації архітектури розроблюваного комплексу захисту баз даних.

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

Алгоритмічне забезпечення апаратно-програмного комплексу Database Activity Monitoring (DAM) призначене для контролю активності користувачів і програмних застосунків під час звернення до мережевих баз даних, виявлення потенційно небезпечних операцій та формування достовірної історії подій інформаційної безпеки. Основним завданням є автоматизоване отримання відомостей про підключення до систем керування базами даних (СКБД), аналіз виконуваних SQL-операцій, перевірка відповідності активності встановленим політикам та визначення ознак несанкціонованого використання інформаційних ресурсів. Особливістю комплексу є поєднання апаратних засобів спостереження за мережевим трафіком із програмними механізмами оброблення, кореляції та зберігання аудиторської інформації.

Загальну послідовність функціонування комплексу відображено на рисунку 4.1. Основний алгоритм передбачає ініціалізацію джерел моніторингу, отримання мережевих пакетів і подій аудиту СКБД, визначення параметрів користувацьких сеансів та нормалізацію зібраної інформації. Після підготовки даних виконуються перевірки правил безпеки та аналіз відхилень від характерної активності. За результатами оцінювання формується повідомлення про потенційну загрозу або звичайну операцію. Усі значущі події передаються до централізованого сховища аудиту. Подальше функціонування здійснюється в режимі безперервного спостереження за підключеними інформаційними ресурсами.

Архітектура апаратно-програмного комплексу передбачає логічне розділення засобів отримання даних і компонентів їхнього аналізу.

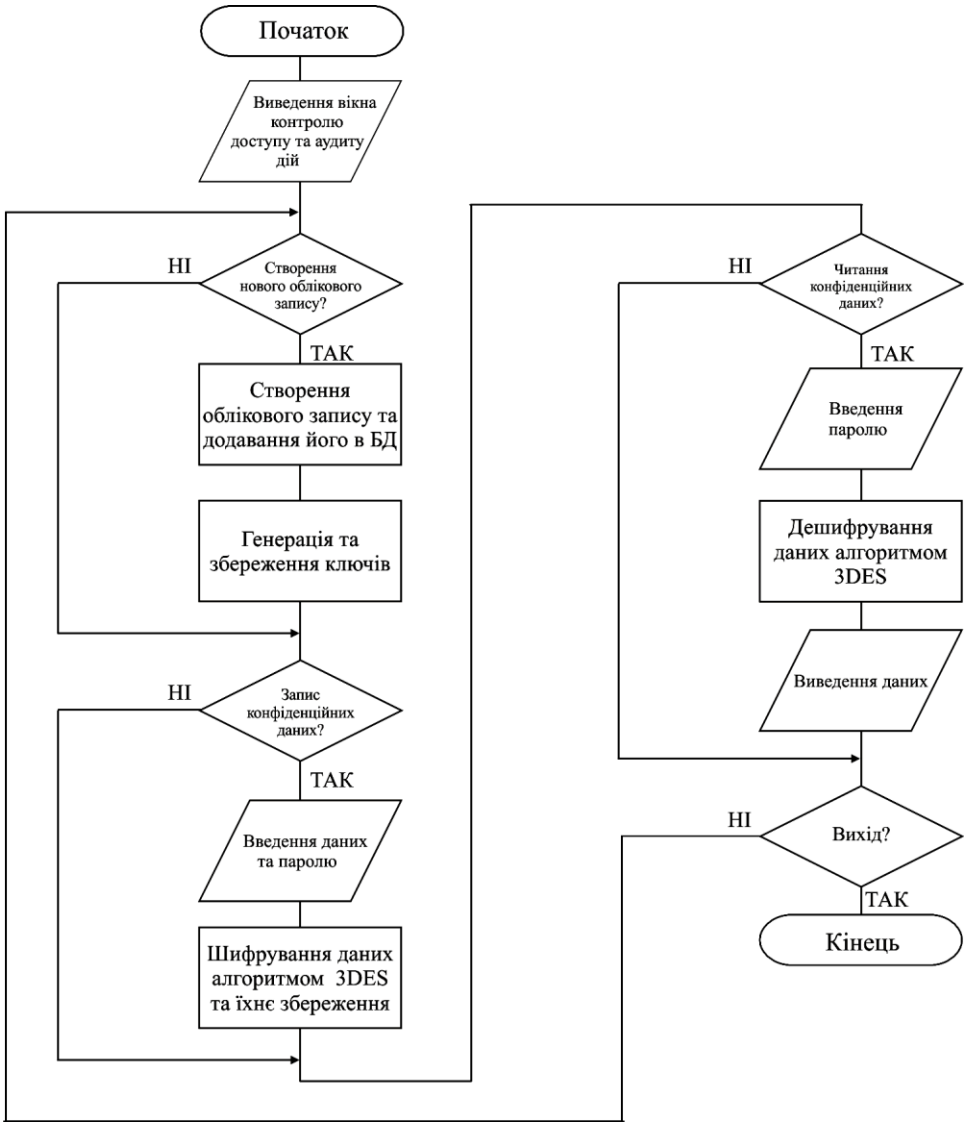

Рисунок 4.1 – Блок-схема основної програми

Апаратна частина може включати окремий сервер моніторингу з мережевими адаптерами для приймання копій трафіку. До його функцій належать отримання пакетів і передавання доступної інформації програмним компонентам. Програмна частина забезпечує реконструкцію мережевих сеансів, аналіз подій СКБД, застосування політик безпеки, виявлення підозрілої активності та формування звітів. Таке розділення дозволяє розміщувати сенсори поблизу контрольованих серверів, а аналітичну підсистему використовувати для централізованого опрацювання даних.

Для організації мережевого спостереження доцільно використовувати копіювання трафіку з комутатора через механізм SPAN або спеціалізований мережевий відгалужувач Network TAP.

У першому випадку копії пакетів передаються на визначений порт комутатора. У другому використовується окремий пристрій, який забезпечує доступ до копії мережевого потоку.

Конкретний спосіб підключення залежить від топології мережі та характеристик обладнання. Для програмного отримання пакетів можуть застосовуватися механізми libpcap або відповідні інтерфейси операційної системи.

Окремим функціональним елементом є підпрограма аналізу активності сеансу бази даних. Принцип її роботи представлено на рисунку 4.2.

Алгоритм підпрограми передбачає отримання нормалізованої події, визначення користувача та цільового інформаційного ресурсу, класифікацію операції й перевірку відповідності правилам доступу. Далі аналізуються частота звернень, характеристики SQL-команди та контекст виконання. За наявності ознак порушення формується відповідна знахідка з визначеним рівнем критичності. Результат передається до компонента аудиту й подальшого реагування.

Важливим обмеженням мережевого моніторингу є використання шифрованих з'єднань між клієнтом і сервером бази даних. Якщо передавання здійснюється через TLS, пасивний мережевий сенсор зазвичай не має доступу до відкритого тексту SQL-запитів.

У такому випадку можуть аналізуватися мережеві адреси, порти, обсяги переданих даних, тривалість сеансів та інші доступні характеристики.

Для отримання відомостей про фактичні SQL-операції необхідно додатково використовувати журнали аудиту СКБД, дозволені серверні агенти або контрольовані шлюзи доступу.

Тому алгоритмічна модель комплексу передбачає можливість поєднання мережевих і серверних джерел без порушення конфіденційності транспортного з'єднання.

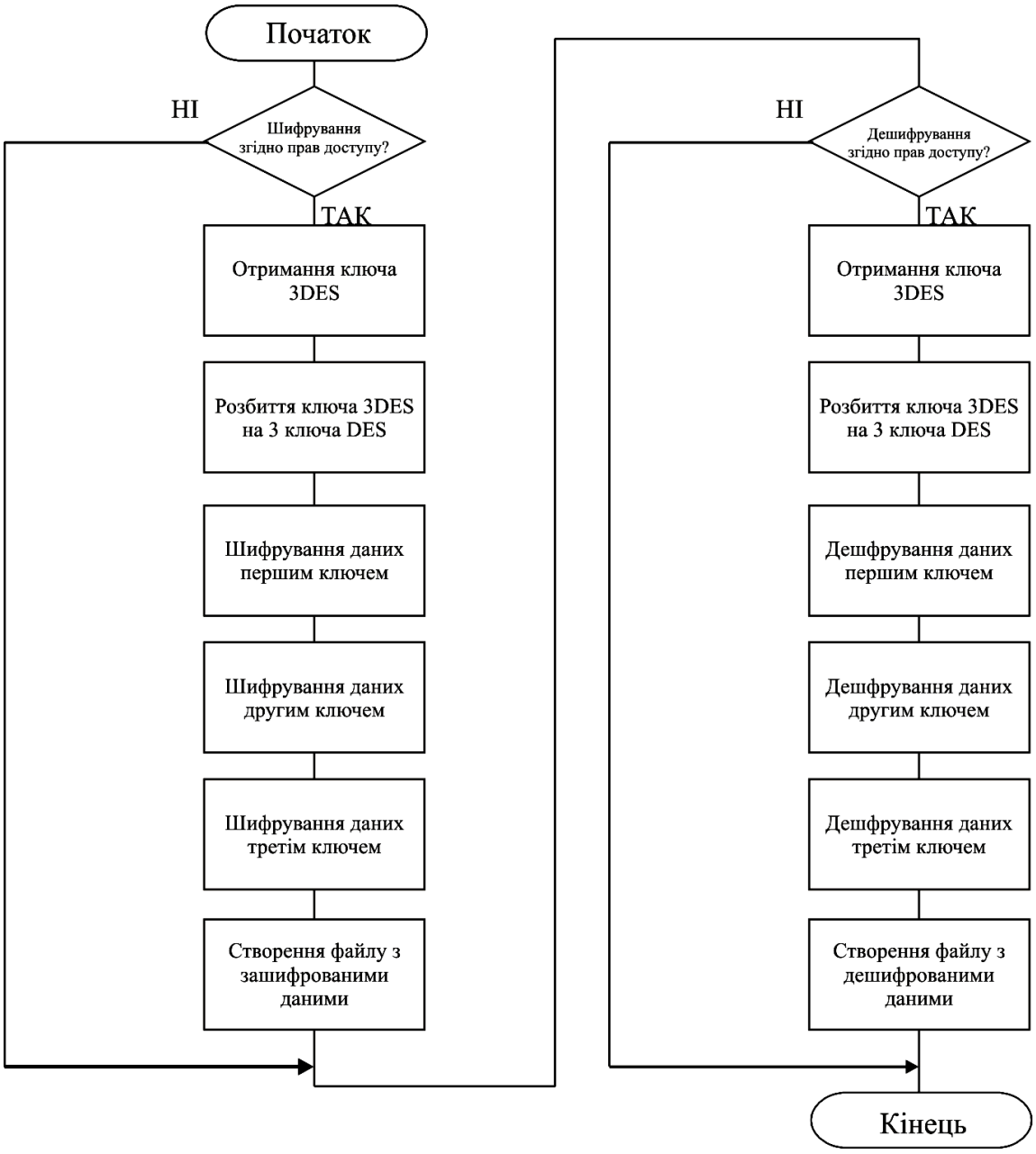

Рисунок 4.2 – Блок-схема роботи підпрограми

Алгоритм виявлення та реєстрації контрольованих баз даних

Початковим етапом роботи комплексу є формування переліку баз даних, активність яких необхідно контролювати. У корпоративній мережі можуть одночасно використовуватися декілька СКБД із різними мережевими протоколами та механізмами автентифікації. Для правильної інтерпретації подій необхідно визначити характеристики кожного сервера та спосіб отримання інформації про його роботу.

Вхідними даними алгоритму є перелік мережевих вузлів, параметри підключення до доступних інтерфейсів моніторингу та відомості про використовувані СКБД. На першому етапі перевіряється правильність зазначених адрес і портів. Далі визначається тип сервера, його функціональне призначення та належність до певного сегмента корпоративної інфраструктури.

Для кожного контрольованого ресурсу формується внутрішній ідентифікатор. Він повинен залишатися стабільним після зміни мережевої адреси або перезапуску сервера. До реєстру додаються тип СКБД, назва інформаційної системи, категорія оброблюваних даних та доступні джерела аудиту.

Наступним етапом є перевірка працездатності підключеного джерела. Якщо мережевий сенсор отримує пакети, але серверні журнали недоступні, система повинна відображати частковий стан моніторингу. Відсутність SQL-подій за таких умов не можна вважати підтвердженням відсутності операцій із даними.

Результатом алгоритму є актуальний реєстр контрольованих баз даних, який використовується під час маршрутизації повідомлень і визначення політик безпеки.

Алгоритм отримання мережевого трафіку апаратним сенсором

Завданням апаратного сенсора є отримання інформації про мережеву взаємодію клієнтських застосунків із серверами баз даних. Для цього може використовуватися окремий мережевий інтерфейс у режимі приймання копій трафіку.

На початковому етапі перевіряється доступність мережевого адаптера та відповідність його параметрів конфігурації моніторингу. Після цього активується механізм захоплення пакетів. Для зменшення обчислювальних витрат можуть застосовуватися фільтри Berkeley Packet Filter, які обмежують потік даних необхідними адресами, портами або протоколами.

Отримані пакети передаються до програмного компонента попередньої обробки. Для кожного пакета фіксується час отримання, джерело спостереження, мережеві адреси, транспортні порти та доступні службові характеристики. Якщо

аналізується TCP-трафік, додатково враховуються номери послідовності та прапорці керування з'єднанням.

Особливу увагу необхідно приділити втратам під час захоплення. За високої інтенсивності мережевого обміну буфери сенсора можуть переповнюватися. Крім того, SPAN-порт не гарантує збереження всіх пакетів за умов перевантаження. Тому алгоритм повинен контролювати доступні лічильники втрат і передавати інформацію про неповноту спостереження.

У випадку використання декількох сенсорів необхідно враховувати можливість отримання копій того самого мережевого потоку. Відповідні пакети не повинні автоматично трактуватися як незалежні операції користувача.

Результатом алгоритму є потік мережевих даних із визначеним джерелом і контрольованими характеристиками повноти захоплення.

Алгоритм реконструкції мережевих сеансів

Окремий мережевий пакет не завжди містить достатньо інформації для визначення операції бази даних. Тому наступним етапом є відновлення логічних з'єднань між клієнтами та серверами.

Вхідними даними алгоритму є пакети TCP та їхні службові поля. Для групування можуть використовуватися адреси й порти обох сторін, транспортний протокол та ідентифікатор джерела спостереження.

На першому етапі визначається, чи належить пакет до вже зареєстрованого з'єднання. Якщо відповідний сеанс існує, оновлюється його поточний стан. Для нового потоку створюється окремий запис із часом початку спостереження.

Під час оброблення TCP необхідно враховувати порядок передавання сегментів. Пакети можуть надходити на сенсор із затримкою, повторюватися або спостерігатися не в початковій послідовності. Тому реконструкція повинна використовувати транспортні номери послідовності та контроль дублювання.

Якщо необхідна частина потоку відсутня, повне відновлення прикладного повідомлення може бути неможливим. У такому випадку алгоритм не повинен формувати вигаданий SQL-запит. Відповідний сеанс позначається як частково спостережуваний.

Після завершення мережевого обміну або закінчення встановленого періоду неактивності запис про сеанс переводиться до стану завершення. Для нього можуть визначатися загальна тривалість, обсяг переданих даних і кількість отриманих пакетів.

Результатом алгоритму є структурований опис мережевого сеансу. За наявності незашифрованого підтримуваного протоколу він може використовуватися для подальшого прикладного аналізу.

Алгоритм отримання подій аудиту СКБД

Для забезпечення контролю SQL-операцій доцільно використовувати можливості аудиту, які надає сама СКБД. Такий підхід дозволяє отримувати інформацію про дії користувачів без необхідності дешифрувати мережевий трафік.

Вхідними даними алгоритму є доступні журнали сервера баз даних, параметри їхнього формату та правила відбору подій. Залежно від використовуваної СКБД можуть застосовуватися штатні засоби журналювання або спеціалізовані модулі аудиту.

Для PostgreSQL одним із можливих варіантів є pgAudit. Він забезпечує детальніший аудит сеансів і операцій із визначеними об'єктами бази даних через механізми журналювання PostgreSQL. Відповідна функціональність передбачає окреме налаштування категорій подій і параметрів їхнього записування.

Для MariaDB може використовуватися Server Audit Plugin. Він підтримує реєстрацію подій підключення, SQL-запитів та доступу до таблиць відповідно до встановленої конфігурації. Для інших СКБД необхідно використовувати сумісні засоби аудиту.

Після отримання запису виконується перевірка його структури. Далі визначаються час події, ідентифікатор сеансу, користувач, тип SQL-операції та результат виконання. Якщо джерело передає текст запиту, він обробляється відповідно до політики захисту конфіденційних даних.

Важливо враховувати, що деякі режими аудиту можуть записувати значення параметрів запитів. Такі дані можуть містити паролі або персональну

інформацію. Тому необхідно обмежувати склад журналів і застосовувати відповідні механізми маскування.

Результатом алгоритму є потік подій СКБД, придатний для аналізу дій користувачів і програмних сервісів.

Алгоритм нормалізації подій баз даних

Події мережевого сенсора та журнали різних СКБД мають неоднакову структуру. Для спільного аналізу необхідно перетворити їх до уніфікованого формату.

Вхідними даними алгоритму є записи про мережеві сеанси, події автентифікації та повідомлення про SQL-операції. На першому етапі визначається тип джерела. Далі застосовується відповідний механізм виділення полів.

Для кожної події доцільно зберігати час виникнення, ідентифікатор сервера, користувача, тип операції, назву бази даних, доступний об'єкт та результат виконання. Окремо фіксується джерело отримання інформації.

Часові позначки необхідно приводити до спільного представлення. Для розподіленої корпоративної інфраструктури доцільно використовувати UTC та забезпечувати синхронізацію часу серверів. При цьому потрібно зберігати початкову часову позначку, якщо вона доступна.

У разі відсутності певного поля його значення не повинно визначатися довільно. Наприклад, мережевий сенсор може встановити IP-адресу клієнта, але не мати достовірних відомостей про користувача СКБД.

Після нормалізації виконується перевірка коректності типів даних і допустимості значень. Некоректні записи передаються до окремого механізму діагностики.

Результатом алгоритму є структуровані події, які можуть оброблятися спільними правилами безпеки незалежно від способу їхнього отримання.

Алгоритм зіставлення мережевих і серверних подій

Поєднання мережевого моніторингу з аудитом СКБД дозволяє отримувати повніше представлення активності. Однак одна операція може бути зареєстрована

кількома джерелами. Тому необхідно уникати її помилкового повторного врахування.

Вхідними даними алгоритму є нормалізовані події мережевого та серверного рівнів. На початковому етапі визначаються спільні характеристики. До них можуть належати адреса клієнта, адреса сервера, часовий інтервал, користувач та ідентифікатор сеансу.

Якщо серверний журнал містить стабільний ідентифікатор підключення, він може використовуватися як основна ознака зіставлення. За відсутності такого значення застосовується сукупність доступних атрибутів.

Особливу увагу необхідно приділити трансляції мережевих адрес та використанню пулів підключень. Якщо декілька користувачів працюють через один прикладний сервер, мережева адреса не дозволяє однозначно встановити ініціатора SQL-операції.

Для таких випадків можуть використовуватися додаткові прикладні ідентифікатори, якщо застосунок передає їх до механізму аудиту. Водночас ідентичність кінцевого користувача не можна автоматично виводити лише з назви облікового запису сервісу.

Після встановлення відповідності події об'єднуються в один логічний запис або пов'язану групу. При цьому зберігаються первинні повідомлення та відомості про походження кожного атрибута.

Результатом алгоритму є узгоджене представлення активності без необґрунтованого дублювання SQL-операцій.

Алгоритм класифікації SQL-операцій

Для визначення характеру взаємодії з базою даних необхідно класифікувати виконувані SQL-команди. Різні типи операцій мають неоднакові наслідки для конфіденційності, цілісності та доступності інформації.

Вхідними даними алгоритму є нормалізований запис аудиту з доступним типом SQL-команди або її текстом. Якщо СКБД уже надала достовірну категорію операції, вона може використовуватися без повторного аналізу тексту.

Для команд читання визначаються операції SELECT та інші відповідні способи отримання інформації. Для змін даних аналізуються INSERT, UPDATE, DELETE та сумісні конструкції. Окремі категорії становлять зміни структури бази даних, керування правами доступу та адміністративні операції.

Під час аналізу тексту необхідно враховувати складність SQL-синтаксису. Визначення типу команди лише за першим словом не завжди є достатнім. Наприклад, конструкція WITH може містити різні типи операцій залежно від підтримуваного діалекту.

Для точнішого аналізу доцільно використовувати синтаксичний парсер, сумісний із конкретною СКБД. Окремо необхідно враховувати виклики збережених процедур. Один виклик може ініціювати декілька внутрішніх операцій, які не обов'язково відображаються в тексті початкової команди.

Після класифікації визначається категорія доступу та можливий вплив операції. Результатом алгоритму є структурована характеристика SQL-активності, яка використовується для перевірки правил безпеки.

Алгоритм перевірки політик доступу до баз даних

Контроль активності передбачає зіставлення фактичних дій користувачів з установленими правилами доступу. Для цього доцільно використовувати централізований механізм перевірки політик DAM.

Вхідними даними є класифікована подія, ідентичність користувача, характеристики цільової бази та відповідна політика. На першому етапі визначається належність ресурсу до певної категорії захисту.

Далі аналізується тип операції. Наприклад, для службового облікового запису може бути передбачено виконання лише визначених запитів читання. Спроба змінити структуру бази даних або права користувачів за таких умов розглядається як порушення політики.

Окремо можуть контролюватися звернення до таблиць із персональними даними, фінансовими записами або іншою конфіденційною інформацією. Для цього необхідно підтримувати реєстр відповідних об'єктів.

Алгоритм повинен враховувати, що наявність технічного дозволу СКБД не завжди означає відповідність операції корпоративній політиці безпеки. Наприклад, адміністратор може мати широкі повноваження, але його використання чутливих даних поза визначеним регламентом потребуватиме додаткової перевірки.

За результатами аналізу формується рішення про відповідність або порушення політики. Оскільки пасивний DAM-сенсор не знаходиться на шляху виконання SQL-запиту, він не може самостійно блокувати вже виконану операцію. Для активного запобігання необхідний окремий контрольований механізм на рівні СКБД або шлюзу доступу.

Алгоритм виявлення аномальної активності користувачів

Частина небезпечних дій може виконуватися з використанням чинних облікових записів і дозволених SQL-команд. Тому аналіз лише явних порушень правил не завжди дозволяє виявити потенційне зловживання.

Для розширення можливостей моніторингу передбачається алгоритм визначення відхилень від характерної поведінки користувачів і сервісів. Вхідними даними є історія SQL-операцій, тривалість сеансів, час активності та характеристики доступу до інформаційних ресурсів.

На початковому етапі формується базовий профіль активності. Для цього аналізуються попередні періоди роботи конкретного користувача або групи однотипних облікових записів.

До характеристик профілю можуть входити звичайна кількість запитів за одиницю часу, типові таблиці, використовувані операції та періоди найбільшої активності. Для службових облікових записів додатково можуть враховуватися характерні прикладні сервери.

Після формування профілю поточні показники порівнюються з історичними. Наприклад, різке збільшення кількості запитів SELECT до конфіденційних таблиць може розглядатися як підозріла активність. Аналогічно перевіряються незвичні адміністративні операції та звернення з нових мережевих вузлів.

Для зменшення впливу короткочасних коливань доцільно використовувати статистичні характеристики часових інтервалів. Вибір параметрів залежить від характеру робочого навантаження.

Важливо враховувати, що відхилення від звичного профілю не є безумовним доказом порушення. Воно може бути пов'язане з плановим резервним копіюванням, оновленням застосунку або іншими технологічними операціями.

Результатом алгоритму є перелік аномальних спостережень, які передаються до модуля кореляції та оцінювання ризику.

Алгоритм аналізу масового отримання та експорту даних

Одним із важливих завдань DAM є виявлення операцій, які можуть свідчити про спробу масового отримання конфіденційної інформації. Такий аналіз повинен ураховувати характер SQL-команд, обсяг оброблюваних даних і контекст користувацької активності.

Вхідними даними є події читання, експорту, копіювання та доступні відомості про кількість повернених записів. На першому етапі визначаються операції, пов'язані з конфіденційними таблицями.

Далі аналізується інтенсивність звернень. Значна кількість послідовних запитів до однієї таблиці або декількох пов'язаних інформаційних ресурсів може бути ознакою автоматизованого вилучення даних.

Якщо СКБД надає відомості про кількість повернених рядків, вони можуть використовуватися для уточнення оцінки. Проте такі показники доступні не для всіх типів журналювання. Тому алгоритм не повинен визначати точний обсяг отриманої інформації лише за кількістю мережевих байтів.

Окремо перевіряються характерні команди експорту, зокрема підтримувані конкретною СКБД операції копіювання даних. За наявності відповідної інформації визначаються цільовий об'єкт і користувач, який ініціював операцію.

Результатом є оцінка потенційно небезпечного отримання даних. Вона використовується для формування сповіщення, але потребує уточнення з урахуванням дозволених технологічних процесів.

Алгоритм контролю адміністративних змін

Зміни структури бази даних, прав доступу та параметрів облікових записів можуть суттєво впливати на безпеку інформаційної системи. Тому доцільно передбачити окремий алгоритм моніторингу адміністративних операцій.

Вхідними даними є події, які відповідають змінам структури, створенню та видаленню користувачів, призначенню повноважень і модифікації параметрів СКБД.

На початковому етапі алгоритм визначає категорію адміністративної дії. Далі перевіряється користувач, від імені якого вона виконана, та відповідність операції встановленій політиці.

Особливу увагу необхідно приділяти змінам, які розширюють повноваження облікових записів. Наприклад, призначення адміністративної ролі звичайному прикладному користувачу може потребувати негайної перевірки.

Наступним етапом є зіставлення часу операції з дозволеними інтервалами технічного обслуговування. Якщо система має доступ до інформації про погоджені зміни, вона може використовуватися як додатковий контекст.

За результатами перевірки формується подія аудиту з визначеним рівнем критичності. Для змін, які можуть впливати на конфіденційність або цілісність даних, доцільно передбачити підвищений пріоритет аналізу.

Алгоритм оцінювання ризику активності бази даних

Після виконання перевірок необхідно визначити значущість виявленої події. Однакова SQL-команда може мати різні наслідки залежно від користувача, цільового ресурсу та умов виконання.

Вхідними даними алгоритму є класифікована подія, результати перевірки політик, ознаки аномальності та характеристики інформаційного ресурсу.

На першому етапі визначається тип операції. Адміністративні зміни, масове отримання даних та спроби доступу до захищених таблиць можуть отримувати вищу початкову критичність, ніж звичайні дозволені запити читання.

Далі враховується важливість бази даних. Для систем, які обробляють персональні або фінансові відомості, застосовуються відповідні вимоги до контролю.

Наступним етапом є аналіз користувацького контексту. Звернення з незвичної мережевої адреси або виконання нетипової операції може підвищувати пріоритет події. Наявність декількох незалежних ознак розглядається як підстава для поглибленої перевірки.

Для прийняття рішення може застосовуватися багатокритеріальна модель оцінювання. Вона враховує тип операції, критичність ресурсу, результат перевірки політик та характер виявленого відхилення.

Результатом алгоритму є визначений рівень ризику з поясненням чинників, які вплинули на оцінку. Конкретні порогові значення повинні уточнюватися за результатами експериментальної перевірки.

У практичній реалізації алгоритм може використовувати зовнішні конфігурації політик, профілі користувачів та статистичні характеристики активності. Для забезпечення відтворюваності результату доцільно зберігати версію правил, яка застосовувалася під час оцінювання конкретної події.

Алгоритм кореляції пов'язаних подій безпеки

Окрема подія може не містити достатньо інформації для встановлення потенційного порушення. Тому необхідно аналізувати послідовності дій, які виконуються одним користувачем або стосуються спільного інформаційного ресурсу.

Вхідними даними алгоритму є нормалізовані події, результати оцінювання ризику та характеристики відповідних сеансів.

На початковому етапі події групуються за спільними ознаками. Для цього можуть використовуватися ідентифікатор користувача, сеансу, сервера або інформаційного об'єкта.

Після групування аналізується часова послідовність. Наприклад, декілька невдалих підключень, подальша успішна автентифікація та масове читання конфіденційної таблиці можуть утворювати сценарій підозрілої активності.

Для кореляції використовуються часові вікна, які визначають допустиму відстань між пов'язаними подіями. Їхня тривалість залежить від характеру контрольованого сценарію.

Якщо повідомлення надходять із затримкою, алгоритм повинен ураховувати початковий час виникнення. Інакше пов'язані події можуть бути помилково розділені.

Результатом є група взаємопов'язаних записів, яка використовується для формування єдиного інциденту або уточнення вже зареєстрованої загрози.

Алгоритм централізованого зберігання аудиторської інформації

Для забезпечення подальшого аналізу необхідно зберігати результати моніторингу у структурованому вигляді. При цьому важливо забезпечити відокремлення аудиторських даних від продуктивних баз, діяльність яких контролюється.

Вхідними даними алгоритму є нормалізовані події, результати кореляції, рівні ризику та службові відомості про джерела.

На першому етапі перевіряється коректність запису та відповідність його встановленій структурі. Далі визначається категорія зберігання. Для детальних подій, агрегованої статистики та інцидентів можуть використовуватися різні правила життєвого циклу.

Як можливий варіант сховища розглядається PostgreSQL або інша СКБД, яка підтримує необхідні операції пошуку та фільтрації. Для значних обсягів подій можуть застосовуватися спеціалізовані системи зберігання журналів.

Після записування події формується підтвердження успішного збереження. Якщо сховище тимчасово недоступне, повідомлення можуть залишатися в проміжній черзі до повторного передавання.

Окремо необхідно контролювати захист аудиторських записів. Користувач, активність якого контролюється, не повинен мати можливості довільно змінювати або видаляти відповідні журнали.

Результатом алгоритму є доступна для аналізу історія подій із визначеним джерелом і часом реєстрації.

Алгоритм контролю цілісності журналів аудиту

Достовірність результатів DAM залежить від можливості встановити, чи змінювалися записи після їхнього отримання. Тому доцільно передбачити окремий механізм контролю цілісності аудиторської інформації.

На початковому етапі кожний запис перетворюється до визначеного канонічного представлення. Це необхідно для отримання однакового результату обчислення контрольного значення незалежно від порядку технічних полів.

Після цього може формуватися криптографічний хеш запису. Для створення послідовності пов'язаних повідомлень допускається включення хешу попереднього запису до наступного.

Такий підхід дозволяє виявляти окремі зміни всередині послідовності за умови збереження довіреного контрольного значення. Однак просте локальне хешування не гарантує захисту від зловмисника, який має можливість змінювати і записи, і всі контрольні значення.

Тому для підвищення достовірності доцільно використовувати зовнішньо захищені контрольні точки, цифрові підписи або сховища з обмеженням модифікації раніше записаної інформації.

Під час перевірки журналу алгоритм повторно обчислює контрольні значення та зіставляє їх зі збереженими. У разі невідповідності формується подія порушення цілісності.

Результатом є можливість виявлення несанкціонованої зміни аудиторських записів у межах гарантій обраного механізму захисту.

Алгоритм формування сповіщень та інцидентів безпеки

Після визначення потенційно небезпечної активності система повинна надати відповідну інформацію адміністратору або фахівцю з інформаційної безпеки.

Вхідними даними є оцінені події, результати кореляції та встановлені правила сповіщення. На першому етапі перевіряється наявність раніше створеного інциденту, пов'язаного з тією самою активністю.

Якщо відповідний інцидент існує, нові події додаються до його історії. За потреби виконується повторне оцінювання критичності. Це дозволяє зменшити кількість окремих повідомлень про одну послідовність дій.

Якщо пов'язаного інциденту немає, створюється новий запис. Він містить час виявлення, відповідну базу даних, користувача, тип активності та причину спрацювання.

Для керування інцидентами доцільно передбачити стани реєстрації, перевірки, підтвердження та завершення. Перехід між ними виконується на основі результатів аналізу й дій уповноваженого користувача.

Окремі події можуть потребувати негайного повідомлення. Наприклад, спроба несанкціонованої зміни прав доступу або масове звернення до конфіденційних таблиць може мати підвищений пріоритет.

Результатом алгоритму є сформовані сповіщення й керовані записи інцидентів, придатні для подальшого розслідування.

Алгоритм контролю працездатності апаратно-програмного комплексу

Безперервність моніторингу залежить від роботи мережевих сенсорів, серверних агентів, колекторів і централізованого сховища. Відмова одного з компонентів може призвести до втрати частини інформації про активність користувачів.

Для запобігання непомітним порушенням передбачається алгоритм контролю стану засобів моніторингу.

Вхідними даними є відомості про доступність компонентів, час останнього отриманого повідомлення, стан внутрішніх черг і доступні показники втрат пакетів.

На першому етапі перевіряється працездатність сенсорів. Далі визначається актуальність повідомлень із кожного контрольованого джерела. Якщо тривалий час не надходять події, система аналізує можливість порушення каналу моніторингу.

При цьому необхідно враховувати характер роботи бази даних. Відсутність SQL-запитів у малонавантаженій системі може бути нормальною. Тому для

контролю доцільно використовувати окремі службові повідомлення про працездатність.

Далі перевіряється можливість передавання подій до сховища. Якщо внутрішня черга зростає, формується повідомлення про потенційне перевантаження.

Результатом алгоритму є актуальний стан компонентів DAM і відомості про повноту доступного спостереження.

Алгоритм оброблення високої інтенсивності SQL-подій

У корпоративних системах із великим навантаженням інтенсивність SQL-запитів може бути значною. Суцільне зберігання всіх доступних повідомлень здатне створювати суттєві вимоги до продуктивності сенсорів, мережі та сховища.

Для забезпечення керованого оброблення передбачається використання проміжних черг і розподілення завдань між аналітичними компонентами.

Вхідними даними алгоритму є потік нормалізованих подій та відомості про завантаження системи. На першому етапі визначається поточна інтенсивність надходження.

Якщо аналітичний компонент не може обробляти всі події безпосередньо, вони тимчасово накопичуються в черзі. Для цього може застосовуватися брокер повідомлень або інший механізм асинхронного передавання.

Під час паралельної обробки необхідно зберігати можливість коректної кореляції подій одного сеансу. Тому записи можуть розподілятися за стабільним ключем, пов'язаним із відповідним підключенням.

За значного перевантаження доцільно контролювати затримку оброблення. Якщо вона перевищує допустиме значення, система формує експлуатаційне повідомлення.

Вибіркове відкидання подій може застосовуватися лише за визначеною політикою. Особливо важливо зберігати інформацію про адміністративні операції та підтверджені порушення безпеки.

Результатом алгоритму є контрольований процес оброблення телеметрії з визначеними обмеженнями пропускної здатності.

Особливості програмної реалізації та інформаційної безпеки

Програмна реалізація апаратно-програмного комплексу може передбачати окремий сервіс захоплення трафіку, колектор аудиторських журналів, модуль нормалізації, аналітичний компонент і підсистему зберігання результатів. Конкретні технології визначаються фактичною архітектурою програмного продукту.

Важливо обмежувати права компонентів DAM. Мережевий сенсор не повинен автоматично отримувати адміністративний доступ до сервера бази даних. Колектор журналів повинен використовувати мінімально необхідні повноваження.

Передавання аудиторських подій до центрального компонента доцільно захищати TLS або іншим відповідним механізмом. При цьому необхідно перевіряти автентичність взаємодіючих сторін.

Окрему увагу слід приділити конфіденційності аудиторської інформації. SQL-запити можуть містити персональні дані, паролі або інші чутливі значення. Тому доцільно передбачити маскування параметрів і обмеження доступу до детальних журналів.

Для зменшення обсягу збережених даних можуть використовуватися нормалізовані шаблони SQL-запитів без конкретних значень параметрів. Такий підхід дозволяє аналізувати типові операції без необхідності зберігати весь вміст кожного запиту.

Перевірка коректності алгоритмів та оцінювання ефективності

Для визначення правильності функціонування комплексу необхідно передбачити функціональні й навантажувальні випробування. Початковим етапом є перевірка реєстрації контрольованої бази даних та отримання аудиторських подій.

Для цього може використовуватися тестова СКБД з визначеним набором користувачів і таблиць. Під час випробування виконуються контрольовані операції підключення, читання, зміни даних і керування правами доступу.

Після завершення операцій перевіряється наявність відповідних записів у системі DAM. Особливу увагу необхідно приділити правильності визначення користувача, часу виконання та типу SQL-команди.

Наступним етапом є перевірка мережевого сенсора. Для цього аналізується можливість отримання копій трафіку та формування записів про мережеві сеанси.

Окремо перевіряється робота за використання TLS. У такому сценарії необхідно підтвердити правильність оброблення доступних мережевих метаданих та отримання SQL-подій із серверного джерела аудиту.

Для перевірки алгоритму оцінювання ризику можуть використовуватися контрольовані сценарії. До них належать виконання нетипових адміністративних операцій, багаторазові невдалі підключення та звернення до захищених таблиць.

Результати порівнюються з очікуваними категоріями подій. Це дозволяє визначати правильність спрацювання встановлених правил і виявляти випадки необґрунтованої класифікації.

Під час навантажувальних випробувань можуть оцінюватися кількість отриманих подій за одиницю часу, затримка оброблення, використання оперативної пам'яті та інтенсивність записування до сховища.

Окремим показником є вплив аудиту на продуктивність контрольованої СКБД. Для його визначення доцільно порівнювати виконання однакового навантаження за різних режимів аудиту. Мережеве пасивне спостереження та серверне журналювання можуть мати різний вплив на систему, тому їх необхідно оцінювати окремо.

Для перевірки відмовостійкості можуть моделюватися зупинка колектора, втрата зв'язку із сенсором і тимчасова недоступність сховища. Очікуваним результатом є правильне відображення порушення та контрольована обробка накопичених повідомлень після відновлення.

Кількісні висновки щодо продуктивності, повноти аудиту та точності виявлення небезпечної активності можуть бути сформовані лише після проведення відповідних вимірювань.

Запропоноване алгоритмічне забезпечення апаратно-програмного комплексу DAM формує послідовний процес контролю активності мережевих баз даних від отримання мережевих і серверних подій до оцінювання ризиків та формування аудиторських записів. Основне значення має поєднання пасивного спостереження за мережевими сеансами з аналізом подій самої СКБД. Це дозволяє враховувати обмеження шифрованих з'єднань і забезпечувати інформативність моніторингу за наявності відповідних джерел аудиту. Подальша програмна реалізація та експериментальна перевірка повинні визначити повноту отримання подій, продуктивність комплексу й придатність запропонованих алгоритмів для корпоративних мережевих середовищ.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення апаратно-програмного комплексу моніторингу активності баз даних (Database Activity Monitoring, DAM) призначене для контролю мережевих підключень, аналізу SQL-операцій, виявлення підозрілої активності користувачів та централізованого аудиту інформаційної безпеки. Його застосування дозволяє об'єднувати відомості мережевих сенсорів і серверних журналів СКБД для подальшого оцінювання потенційних загроз. Головне вікно програми показане на рисунку 5.1.

Практичне впровадження комплексу передбачає підготовку апаратного сенсора, розгортання програмних компонентів аналізу та підключення контрольованих серверів баз даних.

Основними функціональними складовими є компонент захоплення мережевого трафіку, колектор аудиторських подій, модуль нормалізації, підсистема оцінювання ризиків і сховище результатів. За потреби ці компоненти можуть розміщуватися на різних фізичних або віртуальних серверах.

Для роботи апаратної частини доцільно використовувати окремий сервер із мережевими адаптерами для приймання копій трафіку та керувальної взаємодії. Обсяг оперативної пам'яті, продуктивність процесора та характеристики накопичувачів визначаються інтенсивністю мережевих з'єднань і кількістю контрольованих баз даних. Мережевий адаптер повинен забезпечувати приймання необхідного обсягу пакетів без перевищення допустимих втрат.

Як програмне середовище може застосовуватися серверна операційна система Linux. Для отримання мережевих пакетів доцільно використовувати libpcap або сумісний механізм захоплення трафіку. Алгоритми нормалізації, перевірки політик і оцінювання подій можуть реалізовуватися мовою Python. Конкретні програмні залежності та їхні версії визначаються фактичною структурою розробленого програмного комплексу.

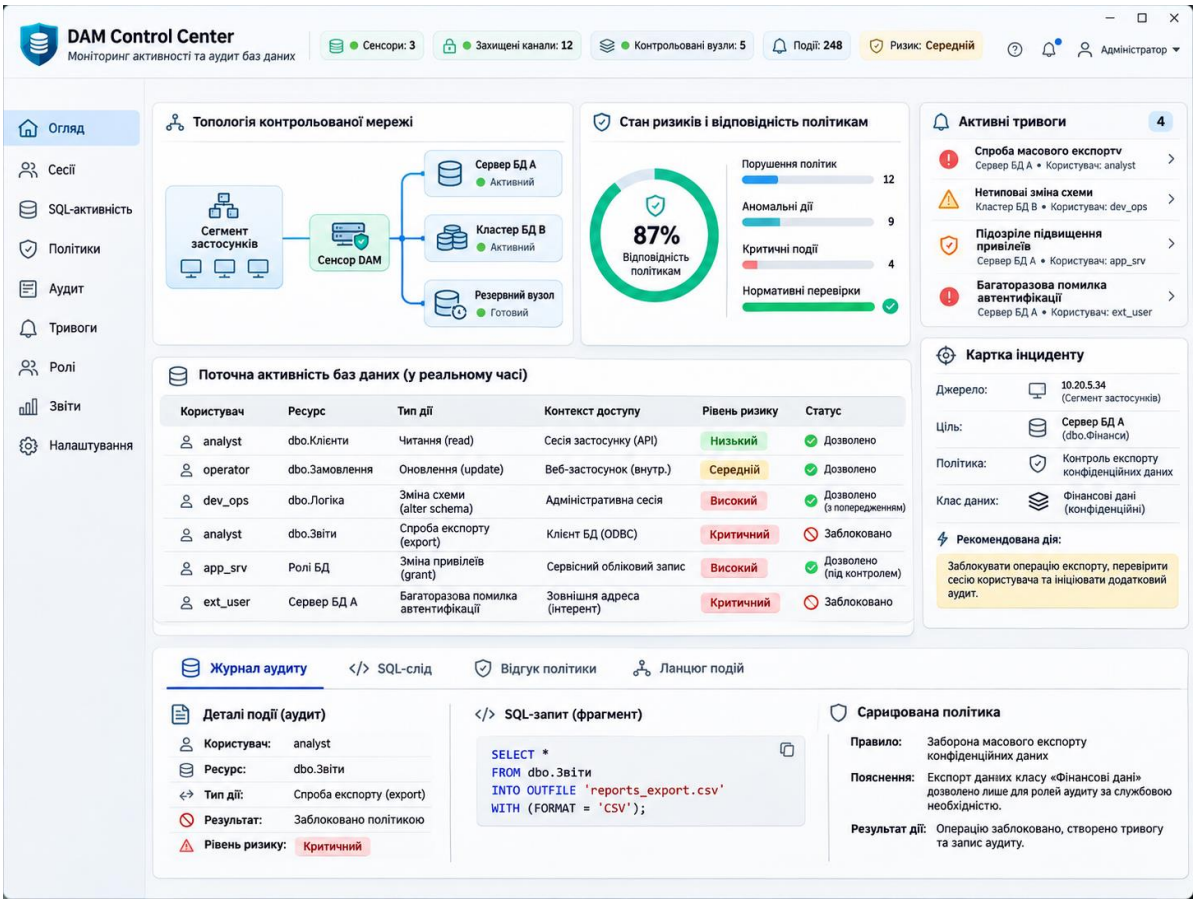

Рисунок 5.1 – Головне вікно ПЗ

Підготовка мережевого сенсора починається з визначення ділянки корпоративної мережі, у якій проходить обмін між клієнтськими застосунками та серверами баз даних. Для передавання копій пакетів може використовуватися SPAN-порт керованого комутатора або спеціалізований пристрій Network TAP. Після фізичного підключення перевіряється доступність мережевого інтерфейсу та можливість отримання трафіку в обох напрямках.

Для сенсорного інтерфейсу необхідно налаштувати режим приймання пакетів відповідно до обраного способу підключення. Якщо використовується SPAN, важливо перевірити правильність вибору джерельних портів або VLAN. Окремо контролюється завантаження порту моніторингу. За перевищення його пропускної здатності частина пакетів може втрачатися, що впливає на повноту аналізу мережевих сеансів.

Наступним етапом є визначення переліку контрольованих СКБД. Для кожного сервера встановлюються мережеві адреси, використовувані порти, тип системи керування базами даних і спосіб отримання аудиторських повідомлень. Ці відомості передаються до конфігурації програмного комплексу. Доступ до керувальних інтерфейсів серверів повинен надаватися лише в межах необхідних повноважень.

Для PostgreSQL як можливий засіб отримання детальних подій може використовуватися pgAudit. Його налаштування передбачає визначення категорій SQL-операцій, які підлягають реєстрації, та параметрів серверного журналювання. Якщо контролюється MariaDB, аналогічні функції може забезпечувати Server Audit Plugin. У разі використання інших СКБД необхідно застосовувати відповідні сумісні механізми аудиту.

Особливу увагу необхідно приділити використанню TLS між клієнтами та серверами баз даних. Пасивний мережевий сенсор не має можливості безпосередньо аналізувати відкритий текст SQL-запитів усередині належно захищеного з'єднання. Тому для таких сеансів мережевий компонент використовується переважно для отримання метаданих. Відомості про SQL-операції надходять через налаштовані журнали СКБД.

Після підготовки джерел налаштовується передавання подій до центрального компонента. Для кожного типу повідомлень визначається відповідний спосіб оброблення. Необхідно забезпечити узгоджене представлення часових позначок, ідентифікаторів користувачів, назв баз даних і типів операцій. Для коректної кореляції подій рекомендується синхронізувати системний час серверів і сенсорів.

Для централізованого зберігання аудиторських записів може використовуватися PostgreSQL або інша СКБД із підтримкою необхідних операцій пошуку та фільтрації. Перед запуском аналітичного компонента необхідно підготувати відповідне сховище, перевірити доступність його мережевого інтерфейсу та налаштувати облікові дані службового підключення.

Аудиторські записи доцільно зберігати окремо від продуктивних баз даних, активність яких контролюється.

Наступним етапом є налаштування правил моніторингу. Для контрольованих баз визначаються категорії конфіденційних об'єктів, допустимі адміністративні операції та умови формування сповіщень. Окремо встановлюються параметри виявлення нетипової активності користувачів. Порогові значення необхідно уточнювати за результатами тестування, оскільки інтенсивність SQL-запитів суттєво залежить від призначення інформаційної системи.

Після визначення політик налаштовується реєстрація подій безпеки. Для кожного запису необхідно передбачити збереження часу, джерела, користувача, типу операції та результату оцінювання. Якщо журнали містять параметри SQL-запитів, необхідно контролювати можливість розкриття конфіденційної інформації. Для зменшення відповідного ризику доцільно застосовувати маскування чутливих значень і обмежувати доступ до деталізованих записів.

Рекомендована послідовність запуску програмного комплексу починається з перевірки працездатності мережевого сенсора та засобів аудиту СКБД. Далі запускається центральне сховище подій і перевіряється можливість записування службової інформації. Після цього активуються колектор журналів і компонент нормалізації. Завершальним етапом є запуск аналітичного модуля, механізмів сповіщення та користувацького інтерфейсу.

Первинну перевірку працездатності доцільно виконувати на тестовій базі даних із заздалегідь визначеними користувачами та об'єктами. Під час тестового сеансу здійснюються підключення до СКБД, читання записів і контрольована зміна даних. Після цього перевіряється надходження відповідних повідомлень до системи DAM. Особливу увагу необхідно приділити правильності визначення типу SQL-операції та її зв'язку з відповідним користувацьким сеансом.

Для перевірки механізмів безпеки можуть виконуватися тестові операції, які порушують установлені правила. Наприклад, звернення до захищеної таблиці від користувача без відповідних повноважень або спроба неузгодженої

адміністративної зміни. Необхідно перевірити формування сповіщення, визначення критичності та збереження пов'язаних подій. Такі випробування повинні проводитися у контрольованому середовищі.

Ідентифікація створеного програмного продукту передбачає представлення відомостей про його розробника. Такі дані необхідні для встановлення авторства та організації подальшого технічного супроводження апаратно-програмного комплексу DAM. Інформацію про автора розробленого програмного забезпечення наведено на рисунку 5.2.

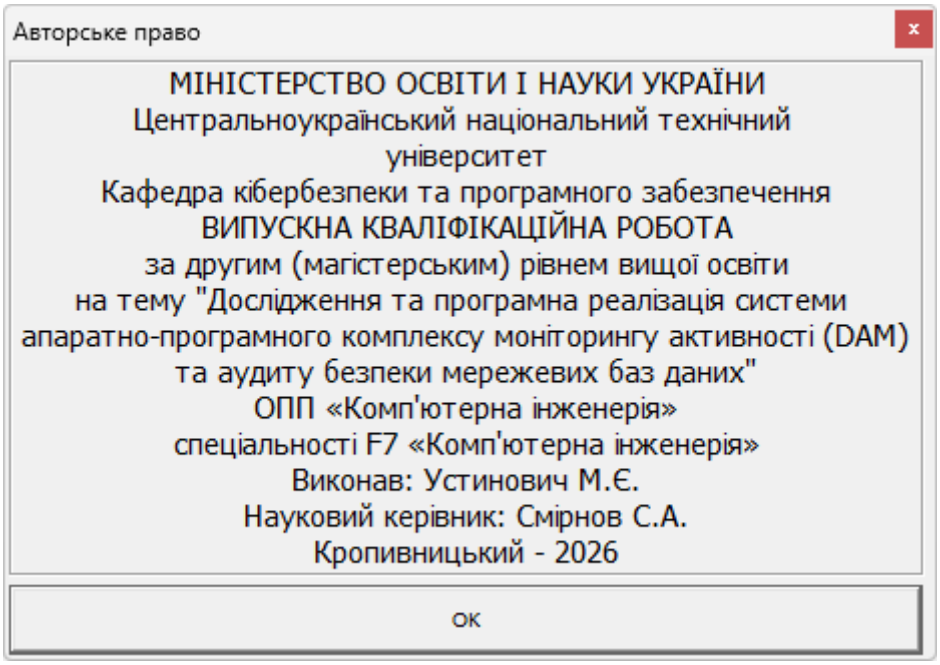

Рисунок 5.2 – Авторське право

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Метою розробки є дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Об'єктом дослідження є процес апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Предметом дослідження є методи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Методи дослідження базуються на методах управління базами даних, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– Розроблено вітчизняний продукт апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Законом України “Про охорону праці” [3] 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» [5], 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 3.3.2-007-97.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робитників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-17.

Умови праці програміста включають наступні фактори:

– параметри повітряного середовища в приміщенні;

– вентиляція приміщення;

– освітлення приміщення;

– параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

7.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-97.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м’язовий апарат.

Ці фактори можуть викликати у працівника певні розлади здоров’я, зокрема підвищення артеріального тиску, кон’юктивіти, тендовагініти та інші захворювання.

7.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 7.1.

Таблиця 7.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,38
Довжина	5,95
Висота	2,8

Таблиця 7.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	8
Об'єм, V	$м^3$	не менше 20.0	22,4

*Згідно ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

У зазначеному приміщенні працюють четверо людей. За даними, які наведено у табл. 7.1- 7.2, можна зробити висновок, що площа та об’єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Таним чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають нормативним вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація.

Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри.

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 7.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії важкості робіт 1а, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 7.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло-гість,%	Швидкість повітря, м/с	Температура, °С	Воло-гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом

(під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк [1]/ Крім того, все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

7.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов’язковою

наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

7.5 Розрахункова частина

Початкові дані для розрахунку захисного штучного заземлення: опір заземлювача, який нормується: $R_{ЗН}$ = 4 Ом.

Для захисного штучного заземлення застосовуються вертикальні електроди з металевого прутка діаметром 30 мм (D = 30 мм = 0,03 м) довжиною L=2,5 м та горизонтальний електрод – металева полоса з перетином 40х4 мм. тип ґрунту – глина (питомий опір 40 Ом·м). Відстань між вертикальними заземлювачами (електродами) А=3 м.

Глибина закладення горизонтального контуру заземлення t = 0,8 м.

Умовна товщина верхнього шару ґрунту: Н=0,4 м. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 7.1).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Завданням э визначення необхідної кількості вертикальних заземлювачів та довжини полоси (горизонтального заземлювача).

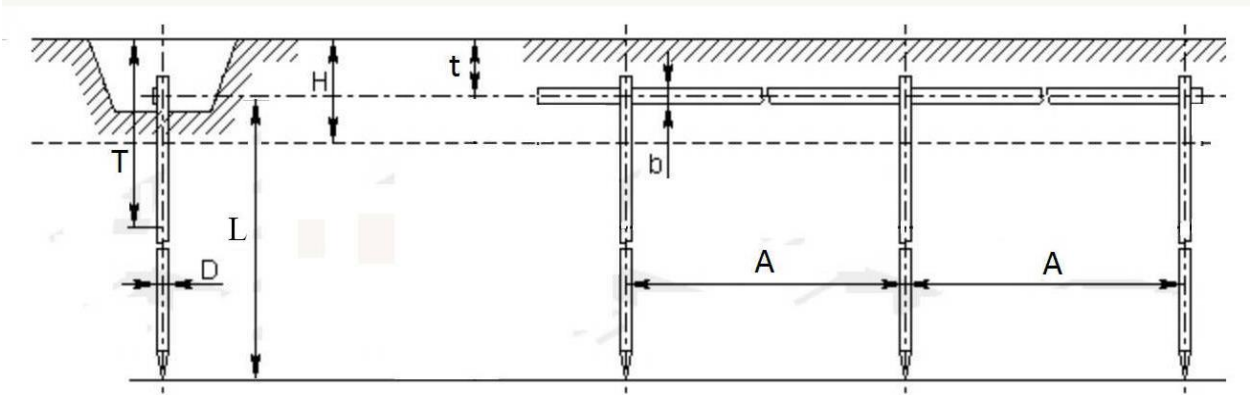

Рисунок 7.1 – Схема штучного заземлення

Виконаємо розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0,8 + 2,5/2 = 2,05 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho_2 = 1,36 \cdot 40 = 54,5 \text{ Ом·м,}$$

де

$\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [8];

$\rho_1 = 50$ Ом·м – табличне значення питомого опору верхнього шару ґрунту [6];

$\rho_2 = 40$ Ом·м – табличне значення питомого опору нижнього шару ґрунту [8].

Опір розтіканню електричного струму одного електрода вертикального заземлювача [8]:

$$R_o = 0,366\frac{\rho}{L}\left(\lg\frac{2L}{D} + \frac{1}{2}\lg\frac{4T + L}{4T - L}\right) = 0,366\frac{54,5}{2,5}\left(\lg\frac{2 \cdot 2,5}{0,03} + \frac{1}{2}\lg\frac{4 \cdot 2,05 + 2,5}{4 \cdot 2,05 - 2,5}\right) = 20,1 \text{ Ом.}$$

Відношення A/L=3/2,5=1,2.

Визначаємо коефіцієнт екранування вертикальних електродів $К_{ев}$=0,8 при попередній (орієнтовній) кількості вертикальних електродів, яке дорівнює 4 [8].

Визначаємо необхідну кількість вертикальних заземлювачів (без врахування горизонтального заземлювача), при $R_{ЗН}$ = 4 Ом :

$$N=R_O /(К_{ев} \cdot R_{ЗН}) = 20,1 / (0,8 \cdot 4) = 5,87 \approx 6 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси [8]:

$$L_П =1,05 \cdot A \cdot N= 1,05 \cdot 3 \cdot 6 =18,8 \approx 19 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси [8]:

$$R_П =0,366 \cdot (\rho_2 \cdot K_П/L_П) \cdot \lg(2(L_П \cdot L_П)/(К \cdot t))=$$

$$=0,366 \cdot (40 \cdot 5/16) \cdot [\lg(2 \cdot 16 \cdot 16)/(0,04 \cdot 0,8)] =16,5 \text{ Ом.}$$

де $K_П$ = 5 – табличне значення коефіцієнта сезонності для відповідної кліматичної зони з'єднуючої полоси [8].

Загальний опір розтіканню електричного струму заземлювача [8]:

$$R = (R_O \cdot R_П)/(R_O \cdot \eta_П + N \cdot R_П \cdot К_{ев})=$$

$$=(20,1 \cdot 16,5)/(20,1 \cdot 0,75 + 6 \cdot 16,5 \cdot 0,8) = 3,32 \text{ Ом.}$$

де $\eta_П$ = 0,75 – табличне значення коефіцієнта екранування з'єднуючої полоси [8].

Умова $R \leq R_{ЗН}$ виконується ($3,32 \leq 4$ Ом).

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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 апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– Досліджена система апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

– На основі отриманих результатів досліджень створена програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних.

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 Visual C#. 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. Устинович М.Є. Дослідження та програмна реалізація системи апаратно-програмного комплексу моніторингу активності (DAM) та аудиту безпеки мережевих баз даних // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.
2. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
3. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
4. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
5. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
6. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
7. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
8. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
9. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
10. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
11. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
12. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.

13. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.

14. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.

15. 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.

16. Смірнов, О., Ткачук, Р., Козірова, Н., Константинова, Л., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С.. «Дослідження застосування векторних баз даних у генеративному штучному інтелекті». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 1 № 33, 667-682.

17. 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.

18. 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.

19. 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.

20. Лисенко, І., Минайленко, Р., Смірнов, С., Буравченко, К., Якименко,

Н., Смірнов, О. «Дослідження інструментів штучного інтелекту для інтелектуального аналізу даних». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2025, № 3(31), С. 227–241.

21. 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 Springer, Cham.

22. Смірнов О., Заріцкий В., Буравченко К., Конопліцька-Слободенюк О., Константинова Л., Якименко Н., Смірнов С. «Оптимізація розпізнавання облич за допомогою CUDA прискореної бібліотеки dlib». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2026, Том 4, № 32, С. 573-582.

23. 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.

24. 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, 2025.

25. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

26. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

27. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп’ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. Вип. 11(42), ч. ІІ. С.52-62.

28. Смірнов, О.А., Константинова, Л.В., Конопліцька-Слободенюк, О.К., Козірова, Н.В, Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка.* 2025. №3(27), С. 429–448.**)**

29. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings* Volume 3664, 2024, Pages 11-23.

30. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Конопліцька-Слободенюк, О., Смірнова, Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об'єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 6-26.

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. Kuznetsov O., Ilchenko O., Kryvinska N., Buravchenko K., Smirnov O., Savchenko Iu. «An Empirical Assessment of Leading Blockchain Financial Services». *2023 IEEE 1st Ukrainian Distributed Ledger Technology Forum (UADLTF)*, Kyiv, Ukraine, 2023, pp. 1-6,

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. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

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. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

38. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

39. 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.

40. 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.

41. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

42. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного

забезпечення комп’ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

43. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

44. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.

45. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

46. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

47. 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.

48. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

49. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

50. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

51. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

52. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

53. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

54. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.

55. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

56. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

57. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing

Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

58. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

59. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

60. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

61. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

62. Смірнов О.А., Дрєєва Г.М., Дрєєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

63. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).