БАКАЛАВРСЬКА РОБОТА

БР. ІП - 03.00.00.000 ПЗ

Група ІП-21-4

Вахновський Ілля

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Вахновський Ілля Сергійович

(прізвище, ім'я, по батькові)

УДК 004.4

(індекс)

БАКАЛАВРСЬКА РОБОТА

Реалізація моделей захисту від несанкціонованого доступу

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Вахновський І.С.

(підпис, ініціали та прізвище здобувача)

Науковий керівник Процюк Василь Романович, к.т.н., доцент

(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ ____ ” _____2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Вахновському Іллі Сергійовичу

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) “ Реалізація моделей захисту від несанкціонованого доступу”

керівник проекту (роботи) Процюк В.Р., доцент, к.т.н.

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

2. Строк подання студентом проекту (роботи) 10 червня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

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

1. Аналіз області побудови моделей та засобів запобігання несанкціонованого доступу
2. Поточні виклики у виявленні внутрішніх загроз
3. Представлення моделі бази даних для розробки ПЗ захисту від несанкціонованого доступу
4. Програмна реалізація моделей захисту від несанкціонованого доступу в базах даних
5. Представлення результатів моделювання процесу несанкціонованого доступу

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Основні небезпеки інсайдерських загроз в БД (рис. 1.1)
2. Типи інсайдерських загроз в БД (рис. 1.2)
3. Класифікація інсайдерських загроз (рис. 1.3)
4. Види доступу (рис. 1.4)
5. Види інсайдерів (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання **28 квітня 2025 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз області побудови моделей та засобів запобігання несанкціонованого доступу	02.05.2025	виконано
2	Поточні виклики у виявленні внутрішніх загроз	08.05.2025	виконано
3	Представлення моделі бази даних для розробки ПЗ захисту від несанкціонованого доступу	18.05.2025	виконано
4	Програмна реалізація моделей захисту від несанкціонованого доступу в базах даних	28.05.2025	виконано
5	Представлення результатів моделювання процесу несанкціонованого доступу	03.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 83 сторінки, 24 рисунки, список використаних джерел із 34 найменуваннями.

Метою роботи є розробка моделі та засобів запобігання несанкціонованому доступу до баз даних, а також створення програмної реалізації, що дозволяє моделювати поведінку потенційних порушників і оцінювати ефективність існуючих механізмів контролю.

Об'єкт дослідження - процес забезпечення інформаційної безпеки в системах керування базами даних.

Предмет дослідження - методи та моделі виявлення і попередження несанкціонованого доступу до баз даних, спричиненого діями інсайдерів.

В першому розділі запропоновані основні рекомендації щодо виявлення внутрішніх загроз несанкціонованого доступу є важливим кроком у напрямку розробки ефективніших систем захисту

В другому розділі сформовано комплексне уявлення про модель бази даних, орієнтовану на забезпечення захисту від несанкціонованого доступу, зокрема атак з боку інсайдерів

В третьому розділі детально розглянуто процес програмної реалізації моделей захисту від несанкціонованого доступу в базах даних, що є логічним продовженням теоретичного моделювання, поданого в попередньому розділі

Висновок: реалізовано повноцінну імітацію несанкціонованих дій користувачів з метою виявлення вразливих місць у структурі бази даних та механізмах контролю доступу

КЛЮЧОВІ СЛОВА: КІБЕРБЕЗПЕКА, ВИТІК ДАНИХ, ВНУТРІШНІ ЗАГРОЗИ, ВИЯВЛЕННЯ ВНУТРІШНІХ ЗАГРОЗ, НЕСАНКЦІОНОВАНИЙ ДОСТУП, БАЗА ДАНИХ

ANNOTATION

The bachelor's thesis contains 83 pages, 24 figures, a list of used sources with 34 names.

The purpose of the work is to develop a model and means of preventing unauthorized access to databases, as well as create a software implementation that allows you to model the behavior of potential violators and evaluate the effectiveness of existing control mechanisms.

The object of the study is the process of ensuring information security in database management systems.

The subject of the study is methods and models for detecting and preventing unauthorized access to databases caused by insider actions.

The first section proposes basic recommendations for detecting internal threats of unauthorized access, which is an important step towards developing more effective protection systems.

The second section forms a comprehensive understanding of the database model, focused on ensuring protection against unauthorized access, in particular attacks from insiders.

The third section examines in detail the process of software implementation of protection models against unauthorized access in databases, which is a logical continuation of the theoretical modeling presented in the previous section.

Conclusion: a full-fledged simulation of unauthorized user actions has been implemented in order to identify vulnerabilities in the database structure and access control mechanisms.

KEYWORDS: CYBERSECURITY, DATA BREACH, INSIDER THREATS, INSIDER THREAT DETECTION, UNAUTHORIZED ACCESS, DATABASE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 9
ВСТУП 10

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОБУДОВИ МОДЕЛЕЙ ТА ЗАСОБІВ ЗАПОБІГАННЯ НЕСАНКЦІОНОВАНОГО ДОСТУПУ 13
1.1. Передумови розв'язання поставленої задачі роботи 13
1.2. Особливості внутрішніх загроз в базах даних 15
1.3. Представлення та класифікація внутрішніх загроз у вигляді таксономії 20
1.3.1. Особливості доступу інсайдера 21
1.3.2. Типи інсайдерів та методи 22
1.3.3. Мотивація інсайдера 24
1.3.4. Профілювання інсайдера 24
1.3.5. Вплив на властивості безпеки 26
1.3.6. Рівень інсайдера 26
1.4. Поточні виклики у виявленні внутрішніх загроз 28
1.4.1. Продуктивність 28
1.4.2. Набори даних про внутрішні загрози 29
1.4.3. Висока розмірність 29
1.4.4. Фізична та кіберповедінка 30
1.4.5. Інтервал аналізу 30
1.4.6. Обмеження статичної політики контролю доступу 31
1.4.7. Складність виявлення внутрішніх загроз 32
1.4.8. Внутрішні загрози в SCADA 33

1.5. Основні рекомендації щодо виявлення внутрішніх загроз несанкціонованого доступу 33
1.6. Висновки до розділу 37

РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МОДЕЛІ БАЗИ ДАНИХ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ІНСТРУМЕНТУ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ 39
2.1. Формальна модель загроз та складність атак інсайдерів на бази даних 39
2.1.1. Модель наміру 39
2.1.2. Модель загроз 41
2.2. Опис залежностей в базі даних. Матриця залежностей 46
2.2.1. Матриця залежностей для моделі даних 48
2.2.2 Опис обмежень залежностей 49
2.2.3. Граф обмежень і залежностей 50
2.3. Визначення залежностей на конфіденційну інформацію. Побудова універсальної схеми 51
2.4. Представлення та опис моделі бази даних 52
2.5. Висновки до розділу 56

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ В БАЗАХ ДАНИХ 58
3.1. Опис процесу імітації несанкціонованого доступу 58
3.2. Реалізація інтерфейсу системи 61
3.2.1 Введення схеми 61
3.2.2. Завантаження схеми 62
3.2.3. Проведення одиничної симуляції 63
3.2.4. Мультисимуляція 64
3.3. Представлення результатів моделювання процесу несанкціонованого доступу 65

3.3.1. Одиничний запуск схеми 65
3.3.2. Кількість користувачів у порівнянні з відхиленими транзакціями 67
3.3.3. Кількість транзакцій у порівнянні з відхиленими транзакціями 70
3.3.4. Кількість елементів, доступ до яких отримано, порівняно з відхиленими транзакціями 71
3.4. Виконання імітації запиту 72
3.5. Висновки до розділу 77

ВИСНОВКИ 79
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 80
БІБЛІОГРАФІЧНА ДОВІДКА

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

APT - Advanced Persistent Threat

CDG - Constraint and Dependency Graph

DM - Dependency Matrix

UEBA - User and Entity Behavior Analytics

DLP - Data Loss Prevention

ВСТУП

Комп'ютерні мережі та телекомунікації відіграють значну роль у обміні інформацією. Зростання цінності інформації та розвиток технологій призвели до збільшення загроз, як зовнішніх, так і внутрішніх. Внутрішні загрози можуть завдати значної шкоди репутації, фінансовим активам та інтелектуальній власності підприємств. Звіт за 2024 рік показує, що 53% загроз за останні 12 місяців виникли всередині організацій. Для захисту від недобросовісних інсайдерів організації повинні мати системи виявлення внутрішніх загроз, які можуть виявляти та нейтралізувати їх до того, як вони спричинять шкоду. Однак поле внутрішніх загроз недостатньо вивчене, і механізми або підходи для їх виявлення залишаються недослідженими.

Інформаційна безпека в умовах цифрової трансформації суспільства набуває особливої ваги, оскільки дедалі більше організацій покладаються на інформаційні системи для зберігання та обробки критичних даних. Одним із найбільш уразливих компонентів таких систем є бази даних, у яких зосереджено великі обсяги конфіденційної інформації. У зв’язку з цим виникає потреба у розробці ефективних засобів захисту від несанкціонованого доступу, особливо з боку інсайдерів — користувачів, які мають легітимний доступ до системи, але використовують його з шкідливою метою.

Актуальність роботи

У сучасних інформаційних системах бази даних є одним з основних об'єктів атак, зокрема з боку інсайдерів — осіб, які мають авторизований доступ до системи. Традиційні засоби захисту не завжди ефективні у виявленні внутрішніх загроз, що зумовлено складністю моделювання поведінки користувача та відсутністю достатньо гнучких механізмів доступу до конфіденційної інформації. Актуальність роботи зумовлена необхідністю створення інструментів, які дозволяють моделювати, виявляти та запобігати

несанкціонованому доступу до даних з урахуванням складної структури загроз, обмежень політик безпеки та поведінкових патернів користувачів.

Робота присвячена аналізу внутрішніх загроз у контексті інформаційної безпеки баз даних, побудові моделі для виявлення таких загроз та розробці програмного інструменту, який реалізує імітаційне моделювання потенційних атак. Запропонований підхід дозволяє виявляти структурні слабкості у політиках доступу та забезпечити підвищення рівня захищеності інформаційних систем.

Таким чином, дана робота є актуальною як з теоретичної, так і з практичної точки зору, оскільки пропонує системний підхід до аналізу внутрішніх загроз і реалізує інструментарій для дослідження та виявлення потенційних порушень у базах даних.

Метою роботи є розробка моделі та засобів запобігання несанкціонованому доступу до баз даних, а також створення програмної реалізації, що дозволяє моделювати поведінку потенційних порушників і оцінювати ефективність існуючих механізмів контролю.

Завдання дослідження:

1. Провести аналіз природи та класифікації внутрішніх загроз у контексті баз даних.
2. Побудувати формальні моделі наміру та загроз інсайдерів.
3. Сформувати матрицю залежностей та універсальну схему для конфіденційної інформації.
4. Реалізувати програмний інструмент, здатний моделювати та аналізувати несанкціонований доступ.
5. Провести тестування та симуляцію сценаріїв атак з оцінкою ефективності запропонованої моделі.

Об’єкт дослідження - процес забезпечення інформаційної безпеки в системах керування базами даних.

Предмет дослідження - методи та моделі виявлення і попередження несанкціонованого доступу до баз даних, спричиненого діями інсайдерів.

Методи дослідження:

- Системний аналіз предметної області.
- Методи моделювання інформаційної безпеки.
- Формальні методи опису залежностей і загроз.
- Метод симуляції сценаріїв атак.
- Емпіричне тестування програмної реалізації.

Наукова новизна

Запропоновано формалізовану модель виявлення внутрішніх загроз у базах даних, що враховує типи інсайдерів, рівень доступу, мотивацію, а також динамічну поведінку в контексті обмежень залежностей і політик безпеки, з подальшою реалізацією засобу симуляції атак і аналізу результатів.

Практичне застосування

Результати дослідження можуть бути використані для удосконалення систем контролю доступу в організаціях, що працюють з конфіденційними даними, зокрема у фінансових установах, державних організаціях та промислових системах SCADA.

Бакалаврська робота містить 83 сторінок, 24 рисунки, 3 розділи список використаних джерел із 34 найменуваннями.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОБУДОВИ МОДЕЛЕЙ ТА ЗАСОБІВ ЗАПОБІГАННЯ НЕСАНКЦІОНОВАНОГО ДОСТУПУ

1.1. Передумови розв’язання поставленої задачі роботи

Метою цієї роботи є аналіз процесу моделювання рішення, запропонованого на основі прогнозування та запобігання інсайдерським загрозам в системах реляційних баз даних. Основною передумовою, яка призвела до цього дослідження, було те, що інсайдери спричинили 52% порушень у 2024 році, більше, ніж кількість зовнішніх загроз для компаній та організацій [5]. Питання безпеки набувають все більшого значення, особливо щодо забезпечення захисту даних від «переривання, модифікації та фабрикації» [1]. Незважаючи на те, що було проведено велике дослідження щодо запобігання атакам сторонніх осіб, і все більше досліджень було присвячено проблемі інсайдерських атак, порівняно з цим було проведено відносно мало досліджень для вирішення проблеми інсайдерської загрози в реляційних базах даних. Інсайдер має авторизований доступ і привілеї, але може становити загрозу, порушуючи політику безпеки системи через законний доступ до інформації. Це відбувається за допомогою інформації, яка може бути виведена з існуючих знань про інші системні підрозділи. Отже, на інсайдерську загрозу в реляційних базах даних в першу чергу впливають залежності, які існують в даній базі даних.

У роботі [2] досліджується проблема отримання знань несанкціонованим інсайдером з використанням залежностей між об 'єктами в реляційних базах даних. Пропонуючи рішення для запобігання інсайдерській загрозі та доступу до інформації, у цій роботі представлені такі механізми, як графік обмежень та залежностей (CDG) та матриця залежностей, які використовуються для представлення залежностей та обмежень між об'єктами [2]. На основі цих графіків можна побудувати графік знань

інсайдерів, щоб показати базу знань користувача. Моделювання, яке є основною метою цієї роботи, використовує методи та процес, запропоновані при визначенні залежностей та обмежень, для визначення загроз та запобігання доступу до конфіденційної інформації несанкціонованих користувачів.

Інсайдерська загроза в реляційних базах даних (РБД) — це ризик компрометації, зловживання або втрати даних, спричинений особами, які мають легітимний доступ до системи. Ці особи можуть бути співробітниками компанії, підрядниками, тимчасовим персоналом або навіть колишніми працівниками, чиї облікові записи ще не були деактивовані.

Першим кроком у прогнозуванні та запобіганні внутрішнім загрозам у реляційних системах управління базами даних є визначення залежностей, які існують між елементами даних. Це тому, що внутрішня загроза в реляційних базах даних в основному залежить від залежностей, які існують між таблицями. Залежності, як визначено в цій роботі, є семантичними відносинами, які існують між атрибутами. Це виходить за межі типових функціональних залежностей, хоча й включає їх. Визначення залежностей в контексті первинних і зовнішніх ключів є першим кроком у створенні матриці залежностей. Однак відстеження залежностей також вимагає концептуального розуміння схеми, включаючи розуміння бізнес-правил і нормативів даної організації. Залежності використовуються, оскільки вони змінюються нечасто.

Мало змін відбувається в структурі таблиць, крім того, після встановлення бізнес-правил і створення моделі даних. Отже, відображення залежностей і обмежень між таблицями забезпечує надійний і послідовний спосіб відстеження умов загроз і конфіденційної інформації, яка існує для будь-якої даної бази даних.

Для цієї роботи використовується приклад схеми бази даних системи розрахунку заробітної плати. Обмеження і залежності, які існують у моделі

даних, використовуються для створення Матриці залежностей, яка, у свою чергу, буде використана для побудови графа знань інсайдера.

1.2. Особливості внутрішніх загроз в базах даних

Дослідження щодо внутрішніх загроз у реляційних системах управління базами даних проводилося для запобігання ситуаціям, коли "інсайдери можуть використовувати свої привілеї або знання різних одиниць системи для виведення інформації про інші одиниці системи, до яких вони не мають доступу". Основною метою існуючих досліджень було визначити стратегію прогнозування та запобігання внутрішнім загрозам у реляційних системах управління базами даних шляхом відстеження загального набуття знань користувачем. Стратегії були розроблені таким чином, щоб запобігти неавторизованому доступу до інформації без впливу на загальну продуктивність користувача.

За допомогою графів знань можна визначити кількість інформації, яку може вивести інсайдер. Це, у свою чергу, може допомогти системним адміністраторам визначити ефективний баланс між безпекою та чутливістю транзакції, яку вони призначають користувацькі дозволи. Призначення дозволів є критичним для захисту безпеки будь-якої системи, включаючи реляційні системи управління базами даних. Знаючи про існуючі проблеми безпеки та маючи уявлення про граф прогнозування загроз, адміністратори можуть призначати дозволи більш ефективно. В результаті користувачі можуть підтримувати високий рівень продуктивності, оскільки вони стикаються з меншою кількістю відхилених транзакцій.

Симуляції, які є основною увагою цієї роботи, слугують для визначення критичних областей для порушень безпеки в реляційній системі управління базами даних на основі залежностей і обмежень, які існують між таблицями для даної бази даних. Запуск симуляції дозволить адміністраторам

краще розуміти, як найкраще призначати дозволи, які дозволяють користувачам доступ до всієї необхідної інформації, але запобігають можливості виведення неавторизованої інформації. Декілька змінних, які можна змінювати в симуляції, від кількості користувачів і транзакцій до кількості елементів даних, до яких здійснюється доступ, дозволяють адміністраторам тестувати багато сценаріїв доступу користувачів до бази даних. Ця інформація може бути використана для визначення найкращого балансу між достатнім доступом до даних, який є життєво необхідним для продуктивності, і надмірним доступом до конфіденційної інформації.

Система розрахунку заробітної плати була використана як модель даних для прикладної схеми бази даних, оскільки вона надає систему, яка містить кілька прикладів конфіденційної інформації. Крім того, база даних розрахунку заробітної плати, ймовірно, буде знайдена в подібній формі в будь-якій організації або компанії і, отже, надає модель, яку можна легко зрозуміти та застосувати. Схема також пропонує кілька досить очевидних залежностей і обмежень між таблицями, що виявилося надзвичайно корисним при проходженні етапів створення графа обмежень і залежностей (CDG) і матриці залежностей.

Друга універсальна схема була створена шляхом випадкової генерації. Універсальна схема показала, що симуляцію можна виконувати на будь-якій схемі бази даних, за умови, що можна отримати залежності та обмеження, які призводять до набуття конфіденційної інформації для схеми. Універсальна схема відрізняється від схеми розрахунку заробітної плати тим, що містить меншу загальну кількість атрибутів і, відповідно, менше умов загроз. Наявність другої схеми надала можливість порівняти дві схеми між собою та визначити, як відмінності в обмеженнях, залежностях і умовах загроз впливають на загальне набуття знань і потенціал загроз, який представляє даний користувач.

Симуляції проводилися з припущенням, що користувачі не мали жодних спеціальних дозволів або авторизацій. Залежності між таблицями та умови загроз визначалися на рівні атрибутів для отримання найвищого рівня деталізації при визначенні потенційних загроз. Симуляція застосовує проактивний підхід, коли користувачам дозволяється доступ до будь-якої інформації, яку вони хочуть, до тих пір, поки доступ не має потенціалу для виведення конфіденційної інформації, до якої вони не мають доступу. У цей момент будь-які запити транзакцій користувачем, які мають потенціал порушити безпеку інформації, будуть відхилені. Залежності та обмеження, які існують для схеми розрахунку заробітної плати та універсальної схеми, детально описані нижче разом з умовами загроз для цих схем відповідно.

Рисунок 1.1 – Основні небезпеки інсайдерських загроз в БД

Інсайдерські загрози є особливо небезпечними з кількох причин (рис. 1.1). Розглянемо їх детальніше:

- Легітимний доступ. Інсайдери вже мають облікові записи, дозволи та знання про систему, що дозволяє їм обходити багато традиційних заходів безпеки (наприклад, міжмережеві екрани, системи виявлення вторгнень).

- Знання системи. Вони можуть знати слабкі місця системи, розташування цінних даних та способи їхнього отримання або маніпулювання без залишення очевидних слідів.

- Мотивація. Мотивація інсайдерів може бути різною: фінансова вигода (продаж даних, шантаж), помста, шпигунство, ідеологічні переконання або просто необережність.

- Складність виявлення. Дії інсайдерів можуть виглядати як звичайна робоча активність, що ускладнює їхнє виявлення системами моніторингу.

Рисунок 1.2 – Типи інсайдерських загроз в БД

Типи інсайдерських загроз у реляційних базах даних поділяються на наступні (рис. 1.2):

1. Зловмисні інсайдери. Ці особи навмисно використовують свій доступ для крадіжки, модифікації або знищення даних. Приклади:

- Крадіжка конфіденційної інформації (персональні дані клієнтів, комерційна таємниця).

- Несанкціонована зміна даних (фальсифікація фінансових звітів, зміна записів клієнтів).

- Видалення важливих даних або таблиць.

- Встановлення шкідливого програмного забезпечення через SQL-ін'єкції або інші методи.

- Надання несанкціонованого доступу зовнішнім зловмисникам.

2. Недбалі інсайдери. Ці особи ненавмисно створюють загрози через помилки, необережність або незнання політик безпеки. Приклади:

- Випадкове видалення або зміна даних.

- Збереження конфіденційної інформації на незахищених носіях.

- Передача облікових даних стороннім особам.

- Ігнорування попереджень системи безпеки.

- Перехід за фішинговими посиланнями.

3. Компрометовані інсайдери. Облікові записи легітимних користувачів можуть бути зламані зовнішніми зловмисниками, які потім використовують їх для доступу до бази даних. Хоча першопричина загрози зовнішня, подальші дії здійснюються під виглядом інсайдера.

Розглянемо основні заходи для запобігання та виявлення інсайдерських загроз у РБД:

- Принцип найменших привілеїв (Principle of Least Privilege), тобто надання користувачам лише тих прав доступу, які їм абсолютно необхідні для виконання їхніх посадових обов'язків.

- Суворий контроль доступу, що полягає у впровадженнв надійних механізмів аутентифікації та авторизації, регулярний перегляд та оновлення прав доступу.

- Аудит та моніторинг активності, а саме ретельне ведення журналів аудиту всіх дій користувачів у базі даних, моніторинг підозрілої активності.

- Шифрування даних, що зберігаються, так і даних, що передаються, для захисту від несанкціонованого доступу.

- Запобігання витоку даних (Data Loss Prevention - DLP), тобто впровадження інструментів та політик для контролю над переміщенням конфіденційної інформації.

- Навчання та підвищення обізнаності персоналу, а саме регулярне навчання співробітників з питань інформаційної безпеки, роз'яснення політик безпеки та наслідків порушень.

- Ретельна перевірка персоналу, проведення перевірок при наймі та періодичних перевірок діючих співробітників, особливо тих, хто має високий рівень доступу.

- Процедури звільнення та відкликання доступу, а саме негайне відкликання облікових записів та прав доступу звільнених співробітників.

- Аналіз поведінки користувачів (User and Entity Behavior Analytics - UEBA), використання інструментів на основі штучного інтелекту для виявлення аномальної поведінки користувачів, яка може свідчити про інсайдерську загрозу.

1.3. Представлення та класифікація внутрішніх загроз у вигляді таксономії

Класифікація різних аспектів внутрішніх загроз у відповідні класифікації та формування їх у набір таксономії є корисною. Це дослідження підсумовує внутрішні загрози в такій таксономії, як показано на рисунку 1.3. Області досліджень поділяються на дві основні категорії.

Прийом від принципу кібербезпеки, кульмінація безпеки є взаємодією людей, процесів і технологій. Отже, наш опис складається з: першого, який описує суб'єкта, який виконує дію. У цьому випадку потенційний зловмисник починає атаку з внутрішньої системи або авторизованої області. Ця категорія охоплює екосистему інсайдерів, відому як корінь доступу, типи інсайдерів, мотивацію, профілювання інсайдерів та властивості безпеки, які постраждали, а також рівень, метод і дії, вчинені.

Друга категорія описує технології та стандарти в галузі виявлення внутрішніх загроз. Ця категорія включає п'ять основних елементів:

аналізовані поведінки, техніки та методи, набори даних, методологію виявлення та матриці оцінки.

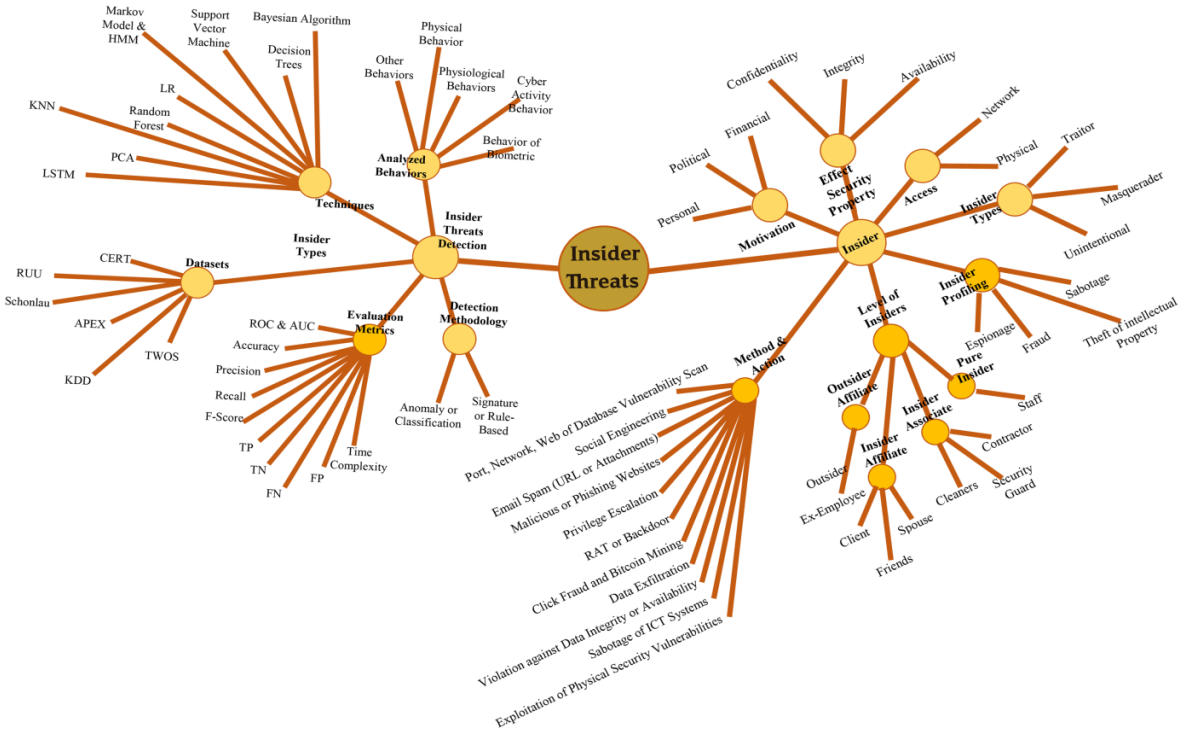

Рисунок 1.3 – Класифікація інсайдерських загроз

1.3.1. Особливості доступу інсайдера

За своєю природою інсайдери мають авторизований доступ до мереж, фізичних або обох, що дозволяє їм становити загрозу. Фізичний доступ описує недобросовісних інсайдерів, які використовують фізичний доступ для неправомірного використання даних систем. Мережевий доступ включає недобросовісних інсайдерів, які використовують свій доступ до даних/систем організації. Прикладом виявлення інсайдерів, які використовують мережевий доступ, є мережевий трафік, який може містити несанкціонований вміст або складатися з протоколів або адрес джерел і кінцевих точок, які можуть бути несанкціонованими. Зловмисне використання мережевого доступу може спричинити значну шкоду організації, і більшість внутрішніх загроз

виникають через неправомірне використання мережевого доступу, таке як саботаж і зміна інформації, зловмисне використання авторизованого мережевого доступу або встановлення шкідливого програмного забезпечення [19].

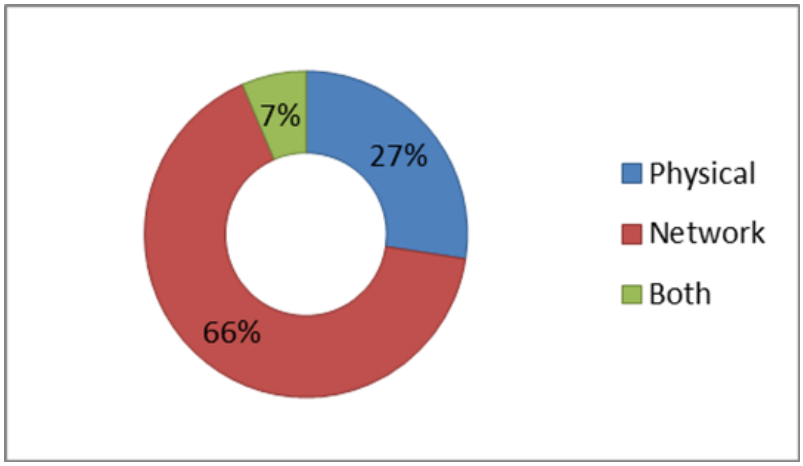

Рисунок 1.4 – Види доступу

Рисунок 1.4 показує, що більшість внутрішніх загроз пов'язані з мережевим доступом, що становить приблизно 66% вивчених випадків. Це пов'язано з тим, що інсайдери можуть легко отримати доступ до даних і систем за допомогою мережі компанії. У випадку витоку даних інсайдери можуть надсилати дані через електронну пошту або завантажувати їх у хмарну службу та використовувати дані поза межами компанії. Хоча більшість випадків інсайдерів потрапляють під сценарій мережевого доступу, фізичний доступ не можна ігнорувати, оскільки він може спричинити таку ж шкоду організації, коли інсайдери навмисно або ненавмисно використовують фізичні вразливості безпеки.

1.3.2. Типи інсайдерів та методи

Інсайдерів можна розділити на три типи: зрадник, маскувальник та ненавмисний. Зрадники складають основну категорію, і, як показано на рисунку 1.5, більшість загроз виникають від цього типу інсайдерів.

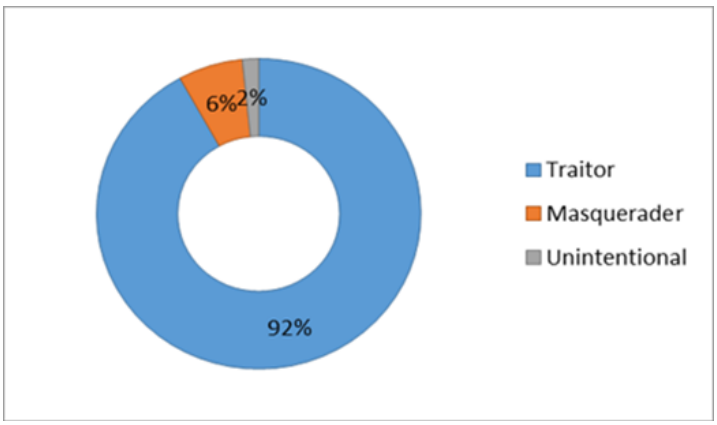

Рисунок 1.5 – Види інсайдерів

Зрадники також називаються недобросовісними користувачами; вони працюють всередині організації і не потребують маскування, але використовують свої привілеї доступу для неправомірного використання систем організації.

Маскувальник — це зовнішній зловмисник, який краде легітимну ідентифікацію інсайдера і використовує вкрадену ідентичність для неправомірного використання.

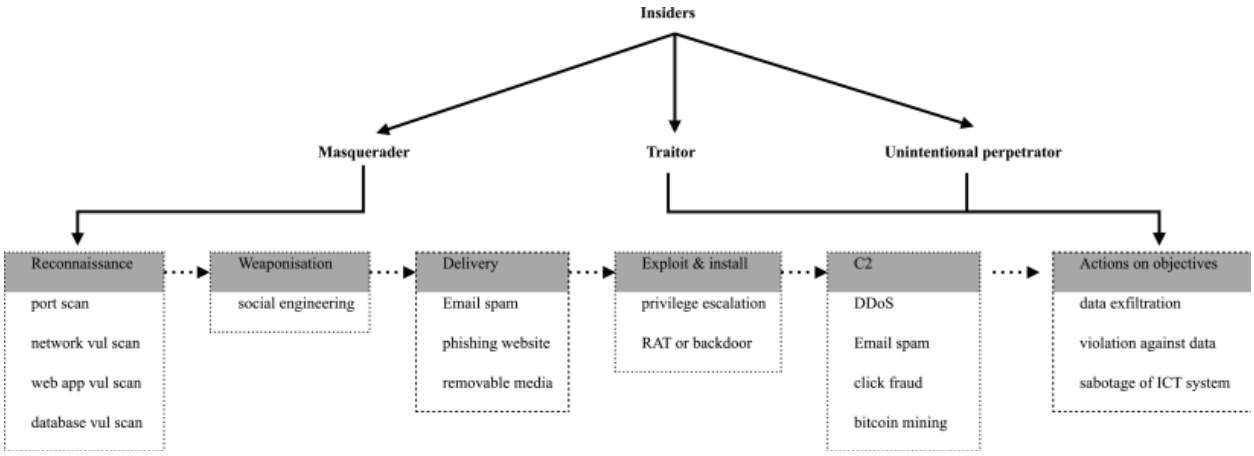

Рисунок 1.6 - Види і методи інсайдера

Ненавмисний тип стосується поточного співробітника, який ненавмисно завдає шкоди або збільшує ймовірність майбутньої шкоди організації. Для поєднання типу інсайдера з методом або загрозою,

використаною для вчинення атаки, запропоновано таксономію, яка відповідає типам інсайдерів методам, які вони використовують, або загрозам, які вони створюють. Для цього використовують ланцюг вторгнення Advanced Persistent Threat (APT) для моделювання всіх типів загроз, від ранніх до пізніх стадій. Рисунок 1.6 деталізує цей процес.

1.3.3. Мотивація інсайдера

Визначення мотивації інсайдера є надзвичайно важливим для полегшення виявлення, впровадження відповідних стратегій мінімізації та забезпечення можливості проведення форензичних досліджень. Переважно мотивацію поділяють на три основні фактори.

Фінансова мотивація є надзвичайно потужною, і вона спонукає людей діяти таким чином, якого ніхто не очікував. Тому не дивно, що фінансова мотивація є однією з основних причин, які спонукають недобросовісного інсайдера. Іншою мотивацією, яка спонукає людей, є їхні політичні погляди; вони сильно вірять у свої погляди, і якщо компанія працює проти їхніх інтересів, це може стати сильною причиною для того, щоб співробітник шкодив компанії або співпрацював із зовнішніми особами для завдання шкоди організації. Ще однією мотивацією є особиста — вона може проявлятися в різних формах або розмірах, але найчастіше це приймає форму шантажу. Зловмисник цілиться на когось із особистими таємницями, яких ця людина не хоче, щоб хтось дізнався, і погрожує розкрити ці особисті таємниці, якщо ціль не співпрацюватиме. Це дуже небезпечний спосіб і може спричинити багато проблем для людей і їхніх організацій.

1.3.4. Профілювання інсайдера

Профілювання інсайдерів поділяється на чотири категорії: саботаж, крадіж (інтелектуальної власності), шахрайство та шпигунство. Недобросовісний інсайдер використовує інформаційні технології для

саботажу або завдання конкретної шкоди організації або окремій особі. Такі недобросовісні інсайдери зазвичай незадоволені співробітники з технічними знаннями та авторизованим доступом. Прикладом цього типу профілювання є встановлення логічної бомби, яка може бути активована після звільнення співробітника [1].

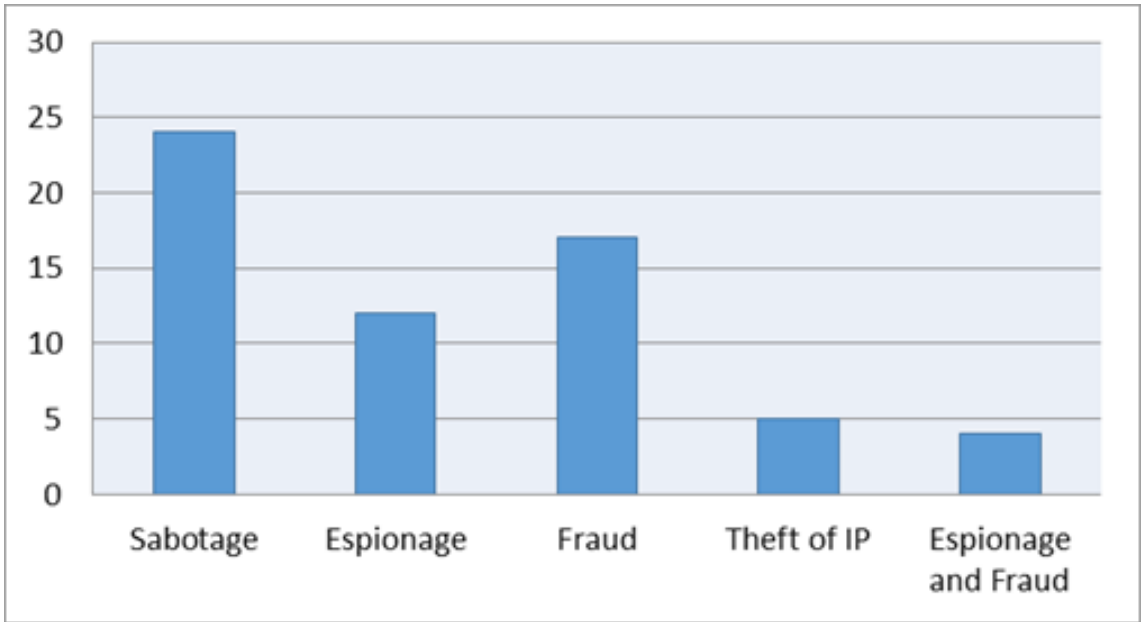

Рисунок 1.7 – Профілювання інсайдерів

Крадіжка інтелектуальної власності — це випадок, коли недобросовісний інсайдер краде інтелектуальну власність, до якої він має доступ під час виконання щоденної роботи, і бере її з собою за межі організації (наприклад, використання інтелектуальної власності для особистого бізнесу, передача її новому роботодавцю або передача конкурентній організації). Цей вчинок найчастіше здійснюють технічні (наприклад, розробники або інженери) або нетехнічні (наприклад, продавці або клерки) співробітники [1]. Шахрайство стосується використання авторизованого доступу для неправомірного використання фінансових ресурсів організації. Іншими словами, шахрайство — це спосіб крадіжки грошей з організації [25]. Нарешті, шпигунство стосується систематичного та

цілеспрямованого вилучення корпоративної інформації недобросовісним інсайдером, що надає недобросовісному інсайдеру стратегічні економічні, військові або міжнародні переваги [26,27].

Рисунок 1.7 показує, що більшість вивчених випадків потрапляють під саботаж і шахрайство, де незадоволені співробітники шкодять організації через помсту після звільнення або з наміром отримати фінансову вигоду, використовуючи свій авторизований доступ.

1.3.5. Вплив на властивості безпеки

Відповідно до типу доступу, який використовує інсайдер, кожен інсайдер має легітимний доступ до активів організації. Цей легітимний доступ може бути фізичним доступом, мережевим доступом або обома (наприклад, люди, які працюють в інформаційній системі в офісі). Ці різні типи доступу можуть створити загрози, які можуть бути вчинені навмисно зрадником або ненавмисно необережним співробітником. Розмежування є важливим, оскільки не всі внутрішні загрози створюються з наміром завдати шкоди компанії. Всі навмисно недобросовісні інсайдери та ненавмисні загрози інсайдерів можуть бути вчинені через неправомірне використання авторизованих дій щодо даних або через використання неавторизованих дій. Загрози можуть призвести до розголошення (загроза конфіденційності інформації), зміни (загроза цілісності інформації) або знищення та перешкод (загрози доступності інформації) [27].

1.3.6. Рівень інсайдера

На основі привілеїв доступу недобросовісні інсайдери поділяються на чотири категорії: чистий інсайдер, інсайдер-афіліат, інсайдер-асоціат та зовнішній афіліат. Перша категорія стосується користувачів з авторизованим доступом і пропусками або ключами до центрів даних організації. Користувач має доступ до всієї інформації про логічні або фізичні структури

даних високої чутливості та права доступу до таких даних. Порівняно з чистими інсайдерами, інсайдер-афіліати не мають причини або дозволу на доступ до ресурсів компанії. Інсайдер-афіліати можуть бути друзями, родичами або клієнтами компанії. У деяких випадках співробітники можуть відвідувати робочі місця і отримувати доступ до ресурсів, використовуючи облікові дані співробітників. Інсайдер-асоціат не працює в компанії, але може мати фізичний доступ до компанії замість мережевого доступу. Інсайдер-асоціат може бути бізнес-партнером, прибиральником, підрядником або охоронцем. Зовнішній афіліат не є частиною організації і не має легітимного доступу до ресурсів організації. Однак вони можуть спробувати отримати доступ до ресурсів через незахищені мережі. Зовнішній афіліат може нелегально отримати доступ до мережі для отримання облікових даних від організації.

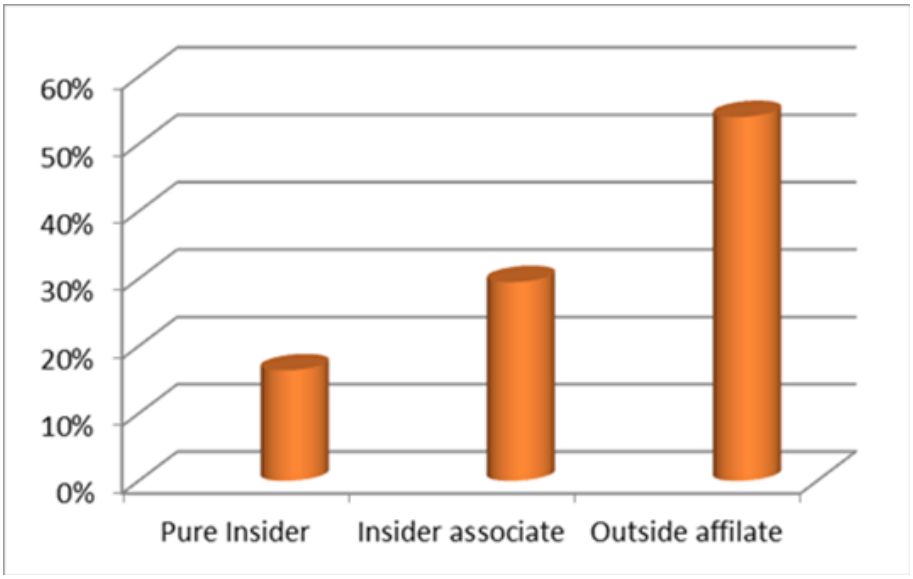

Рисунок 1.8 – Рівень інсайдера

На основі вивчених випадків рисунок 1.8 показує, що більшість інсайдерів потрапляють під категорію зовнішніх афіліатів, які зазвичай є колишніми співробітниками, яких звільнили, і вони становлять унікальну загрозу через їхні знання про організацію та їхні помстиві мотиви.

1.4. Поточні виклики у виявленні внутрішніх загроз

В даному розділі представляються поточні виклики у виявленні внутрішніх загроз, що поділені на одинадцять категорій, які обговорюються далі і подані на рисунку 1.9.

Рисунок 1.9 – Виклики у виявленні загроз

1.4.1. Продуктивність

Оскільки зловмисник є легітимним користувачем системи, важко провести чітку межу між легітимною та недоброзичливою поведінкою. Більшість існуючих підходів до виявлення внутрішніх загроз використовують підхід виявлення аномалій, який є наглядовим або ненаглядовим методом, який класифікує невеликі відхилення від нормальних патернів поведінки як аномалії та, отже, класифікує це як недоброзичливу поведінку. Однак більшість цих аномалій не є недоброзичливими діями. Ці методи схильні до помилок першого роду через це припущення, що робить ці підходи важкими для застосування в корпоративних середовищах. Іншими

словами, зменшення помилок I та II роду для виявлення внутрішніх загроз без впливу на точність виявлення залишається значною проблемою.

1.4.2. Набори даних про внутрішні загрози

Незважаючи на прогрес у дослідженнях внутрішніх загроз, виклики у перевірці та удосконаленні моделей виявлення залишаються через відсутність реальних даних від організацій. Відсутність фактичних даних про внутрішні загрози також є значною перешкодою для оцінки та розробки систем виявлення внутрішніх загроз. Крім того, синтетично створені набори даних, використані в оглянутих статтях, не були створені спеціально для внутрішніх загроз. Крім того, деякі з цих наборів даних не містять недоброзичливих даних, тоді як інші застаріли.

Незважаючи на зростання кількості випадків внутрішніх загроз, не всі організації повідомляють про такі випадки або дозволяють доступ до своїх даних, зазвичай через етичні та приватні занепокоєння. Питання доступу до реальних даних є критичним для виявлення внутрішніх загроз, яке продовжує бути значною перешкодою для перевірки та удосконалення ефективних та масштабованих систем виявлення. В результаті більшість існуючих систем виявлення перевіряються та оцінюються на синтетичних та симульованих наборах даних, з упередженнями, які це несе.

Щоб уникнути виявлення інструментами, такими як системи виявлення вторгнень, зловмисники можуть використовувати криптографію для маскування своїх атак. Така ситуація робить системи виявлення нездатними аналізувати шифрований трафік або шифровані дані, що є ще однією значною обмеженністю поточних систем виявлення вторгнень.

1.4.3. Висока розмірність

Здатність захоплювати журнали діяльності є перевагою, яка може надати інсайт у дії співробітників. Однак аналіз журналів діяльності

залишається складним для аналітиків через величезну кількість дій, які співробітники генерують щодня. Велика кількість персоналу організації вимагає моніторингу поведінкових характеристик, що призводить до необхідності обробки величезної кількості даних. Зростання цих даних перевищує здатність людських аудиторів та адміністраторів перетравлювати такі обсяги даних за допомогою ручного аналізу.

1.4.4. Фізична та кіберповедінка

Ще однією обмеженністю поточних підходів до виявлення внутрішніх загроз є те, що вони зосереджуються лише на кібербезпеці або фізичній безпеці всередині кібербезпеки. Більшість попередніх робіт не використовували обидві поведінки кібербезпеки та фізичної безпеки в аналізі виявлення внутрішніх загроз. Більшість дослідників прагнуть виявити інсайдерів, спостерігаючи за поведінкою або з кібербезпеки, або з фізичної безпеки. Однак щодо виявлення фізичних загроз більшість існуючих досліджень застосовують механізми контролю фізичного доступу, які можуть контролювати фізичний доступ неавторизованих користувачів до певної точки. Однак такі механізми неефективні проти атак інсайдерів.

1.4.5. Інтервал аналізу

Кілька систем виявлення внутрішніх загроз не можуть забезпечити реакцію в реальному часі, що підкреслює необхідність додаткових дослідницьких зусиль. У випадку офлайн інструментів для аналізу журналів виникає проблема, оскільки ці інструменти не можуть забезпечити підтримку та реагувати на аналіз журналів у реальному часі. Отже, більшість поточних систем все ще не мають інструментів реального часу, що запобігає подальшим діям щодо подолання цієї проблеми. Великі обсяги аудиторських даних збираються з корпоративних середовищ у формі журналів сервера, які потенційно можуть зіграти роль у прийнятті рішень щодо доступу. Однак ці

дані зазвичай використовуються лише для офлайн форензики, що призводить до ситуації "пізніше — це запізніло".

Одним із підходів до виявлення внутрішніх загроз є навчання даних для побудови моделі класифікації наглядового навчання. Однак більшість запропонованих методів виявлення побудовані на основі наглядового навчання. Отже, необхідні контекстні дані про користувачів та процес навчання для методів наглядового навчання, які є специфічними для виявлення внутрішніх загроз.

1.4.6. Обмеження статичної політики контролю доступу

Загалом, інсайдери знають політику і застосовують ці знання. Зазвичай політика пов'язана з правами доступу, які надаються інсайдерам, і вона спрямована на обхід правил регулювання. У цьому випадку інсайдер з недоброзичливими намірами може мати можливість знищити або вкрасти інформацію. Однак такі права доступу все частіше використовуються неуважними, ворожими, нелояльними та псевдонедоброзичливими інсайдерами. Отже, відсутність систем контролю доступу в системах виявлення внутрішніх загроз залишає організації вразливими до таких загроз.

Існуючі техніки контролю доступу розроблені на основі статичних політик, які зв'язують крипто-кредиталі з атрибутами, використаними правилами контролю доступу. Динамічні події, такі як зміни поведінки актора (наприклад, співробітник виконує незаконну дію всередині своїх привілеїв), підрив кредитацій (наприклад, крадіжка загальних карт доступу) та зміни в структурах документів (наприклад, редагування сторінок вікі), не можуть бути виявлені, що залишає системи вразливими протягом тривалого часу.

Як заходи проти внутрішніх загроз, правила контролю доступу є складнішими, ніж ті, що використовуються для заходів проти зовнішніх загроз. Крім того, виклик щодо того, де слід встановити точку контролю

доступу в мережі для контролю внутрішніх загроз, залишається невирішеним.

1.4.7. Складність виявлення внутрішніх загроз

Виявлення внутрішніх загроз стає все складнішою та важкою задачею з наступних причин. По-перше, інсайдери з довіреним доступом можуть виконувати неавторизовані дії. Отже, зовнішні мережеві системи безпеки, такі як брандмури, системи виявлення вторгнень та антивірусне програмне забезпечення, не можуть виявити недоброзичливого інсайдера. По-друге, атаки інсайдерів проявляються в багатьох формах. Наприклад, недоброзичливий інсайдер може встановити логічну бомбу для порушення роботи систем або крадіжки інтелектуальної власності. Різноманітність атак інсайдерів збільшує складність виявлення внутрішніх загроз. Нарешті, атаки інсайдерів часто здійснюються недоброзичливими інсайдерами під час звичайної роботи, що маскує аномальну поведінку недоброзичливих інсайдерів серед більшості нормальної поведінки співробітників.

Більшість існуючих рішень для внутрішніх загроз зосереджені на виявленні індивідуальних атак. Однак спільні атаки можуть бути організовані двома або більше інсайдерами, що важко виявити. Недоліком цих типів атак є те, що дії кожного інсайдера можуть виглядати доброзичливими, але в сукупності вони можуть скласти недоброзичливу дію. Отже, потрібні додаткові зусилля для подолання спільних атак.

Більшість організацій приділяють більше уваги зовнішнім загрозам, ніж недоброзичливим інсайдерам. Дослідники в галузі кібербезпеки розглядали та визначили багато різних загроз безпеки. Дослідники підкреслюють, що загрози від недоброзичливих інсайдерів є більш небезпечними, ніж зовнішні загрози; однак ця заява не отримала належної уваги. Ще однією проблемою є брак розуміння намірів та стратегії недоброзичливого інсайдера. Більшість дослідників у галузі безпеки

зосереджуються на нижніх шарах програмного забезпечення — тобто на видобуток даних на рівні мережі та хоста або на рівні вихідного коду. В результаті ці рішення зосереджуються в основному на певних підписах або категоріях загроз, які є природно тактичними. Поточна робота вказує на те, що загальна картина розуміння намірів та стратегії недоброзичливих інсайдерів залишається неповною.

1.4.8. Внутрішні загрози в SCADA

Системи керування та збору даних (SCADA) є чутливими частинами критичної інфраструктури. Кожен успішний зловмисний інцидент може спричинити значні матеріальні, економічні та людські збитки. Оператори відіграють ключову роль у SCADA системах, і їхні команди можуть мати значний вплив на надійність критичної інфраструктури. Отже, атаки інсайдерів та підходи до подолання недоброзичливого інсайдера в SCADA безпеці привертають все більше уваги. Всі SCADA середовища вразливі до атак інсайдерів, хоча для прямої атаки на віддалені термінали потрібен фізичний доступ до каналів зв'язку, але після отримання цього доступу, зазвичай, всі захисти будуть обійдені. Обладнання програмованих логічних контролерів (PLC) також особливо вразливе до віддалених атак через вразливість за проектування пристрою.

1.5. Основні рекомендації щодо виявлення внутрішніх загроз несанкціонованого доступу

Рекомендації щодо досліджень виявлення внутрішніх загроз поділені на вісім категорій, як показано на рисунку 1.10.

Для того щоб виявлення внутрішніх загроз було ефективним, важливо покращити показник відновлення без втрати точності. Вибір характеристик даних безпеки є потенційним підходом для покращення продуктивності.

Отже, потрібні додаткові зусилля для розробки технологій та інструментів захисту від внутрішніх загроз на основі вибірки характеристик. Одним із таких прикладів рекомендацій щодо вибору характеристик є використання складних моделей, таких як рекурентна нейронна мережа з довгою пам'яттю (LSTM) [96] та грати з воротами (GRU), щоб можна було навчитися багатій репрезентації поведінки користувача.

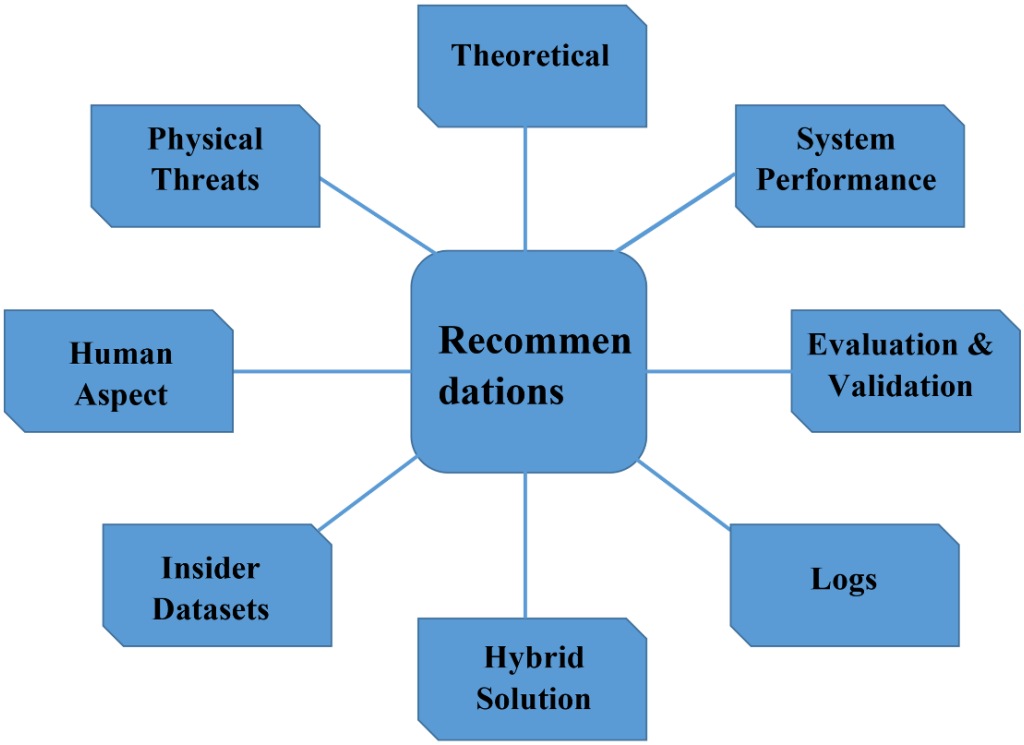

Рисунок 1.10 – Рекомендації щодо виявлення загроз

Попередні дослідження рекомендували створювати нові набори даних про внутрішні загрози з більшою кількістю даних та потоками. Важливо також підтвердити, чи можуть моделі нормально працювати з щоденними оновленнями, тоді як нормальна поведінка користувачів змінюється з часом. Новий набір даних повинен містити більше недоброзичливих даних, оскільки поточні набори даних містять лише кілька недоброзичливих даних, деякі з яких застаріли. Також слід враховувати спільні атаки при створенні нового

набору даних про внутрішні загрози для подальшого покращення та реалістичної перевірки методів виявлення [1]. Отже, важливо підтримувати оновлення наборів даних про внутрішні загрози з нормальними та недоброзичливими патернами поведінки для перевірки та оцінки запропонованих рішень для виявлення недавніх атак інсайдерів.

У загальному, поточні рішення в основному ґрунтуються на виявленні аномалій та ненаглядових підходах через нерівноваженість класів у наборах даних та інші проблеми, такі як атаки нульового дня. Однак хороша та міцна система виявлення внутрішніх загроз повинна використовувати комбінацію кількох незалежних підходів. На першій лінії захисту слід розглянути виявлення неправомірного використання для покриття відомих внутрішніх загроз. Однак на другій лінії слід використовувати виявлення аномалій та інші найкращі практики, такі як запобіжні та пом'якшувальні техніки.

Одним із кращих підходів до пом'якшення внутрішніх загроз є управління журналами, яке включає аналіз журналів та кореляцію подій. Аналіз журналів може визначити причину внутрішньої атаки та захистити мережу від порушень безпеки водночас. Поєднання інших джерел даних для кращого виявлення внутрішніх загроз включає журнали Windows, журнали Active Directory, журнали друку та журнали фізичної безпеки. Застосування інструментів великих даних та аналітики дозволяє захоплювати та керувати потоком журналів і забезпечувати доступність для обробки партій та потокової обробки в аналітичні інструменти, а також надавати інтерфейси для запитів ad hoc. Є відомий приклад обробки великих даних Splunk, який пропонує можливості, які полегшують кіберзахисникам створення кореляцій між різними частинами інформації журналів за допомогою спеціалізованої мови запитів. Отже, поєднання та аналіз цих типів журналів та застосування інструментів аналітики великих даних можуть бути напрямком для майбутніх досліджень внутрішніх загроз.

Наразі не існує рамки або стандарту для оцінки систем виявлення внутрішніх загроз [7]. Отже, вибір найкращої системи виявлення залишається складною задачею через багато критеріїв оцінки, таких як точність, відновлення, FPR, часова складність тощо. Тому рекомендується розробити універсальний фреймворк для оцінки систем виявлення внутрішніх загроз, щоб допомогти дослідникам та практикам у оцінці їхніх запропонованих систем. Що стосується покращення стандартів оцінки, гіккомендується співпраця між дослідниками в академічних колах та практиками в галузі для того, щоб дослідники могли отримати зворотний зв'язок від галузі щодо розроблених систем для оцінки, що може призвести до покращення адаптації та адаптації відповідних систем до реальних умов [7].

Більшість існуючих робіт зосереджені на технічному аспекті інсайдерів, мережі, машинах та системах. Однак нетехнічні аспекти проблеми інсайдера є критичними елементами будь-якої системи виявлення внутрішніх загроз. Отже, багато ефективних технік можуть бути застосовані для вирішення людського аспекту проблеми інсайдера, такі як людська комунікація (наприклад, тон голосу, мова тіла, ставлення до інших).

Хоча дослідники в галузі кібербезпеки, здається, усвідомлені про фізичні наслідки кіберзагроз, більшість досліджень проводяться або в "кібер" або "логічному" аспекті безпеки. Навіть у таких умовах багато кібербезпеки загроз (особливо в Інтернеті) виникають від фізичних втручань, і нам все ще потрібні підходи, які моделюють як кібер, так і фізичні аспекти.

Майбутні дослідження повинні надавати інформацію для всіх відповідальних департаментів управління та фахівців безпеки, надаючи глибоке розуміння характеристик інсайдерів, загроз, які вони створюють, потенційних ризиків внутрішніх загроз та можливих контрзаходів. Аналіз проблеми в цілому, включаючи розробку таксономії інсайдерів, атак та контрзаходів, вказує на конкретну загрозу інформаційній безпеці з прогнозами моделей розвитку [12].

Внутрішні загрози є однією з найскладніших загроз безпеки та головною проблемою для організацій усіх розмірів. Чисельні дослідження були проведені в цій галузі, і зусилля продовжуються зростати, хоча межі та описи внутрішніх загроз залишаються нечіткими. Отже, розуміння та отримання інсайтів щодо виявлення внутрішніх загроз є важливим напрямком досліджень. Цей огляд мав на меті надати всебічне уявлення та глибоке розуміння галузі внутрішніх загроз шляхом огляду та класифікації існуючої літератури. Разом із глибоким дослідженням існуючої літератури та аналізом реальних випадків було виділено значну інформацію шляхом інтенсивного огляду та аналізу остаточного набору переглянутих статей, таких як виклики та питання, з якими стикаються дослідники в галузі внутрішніх загроз. Крім того, важливі рекомендації щодо виявлення внутрішніх загроз, а також наборів даних та технік, які використовувалися, були запропоновані. Різні рекомендації можуть надати майбутнім дослідникам чітке уявлення про напрямок досліджень щодо виявлення внутрішніх загроз.

1.6. Висновки до розділу

В даному розділі розгляд проблеми виявив її безперечну актуальність в контексті забезпечення інформаційної безпеки сучасних баз даних. Специфіка внутрішніх загроз, обумовлена легітимним доступом інсайдерів до інформаційних ресурсів, суттєво ускладнює застосування традиційних механізмів контролю доступу, орієнтованих переважно на зовнішні атаки.

Розроблена таксономія внутрішніх загроз стала важливим інструментом для систематизації розуміння їхньої природи та ключових характеристик. Представлення загроз через призму особливостей доступу інсайдера, їхніх типологій та методів, мотиваційних чинників, профілювання,

впливу на властивості безпеки та рівня доступу забезпечує більш структурований підхід до аналізу ризиків та розробки контрзаходів.

Водночас, проведений аналіз виявив низку значущих викликів, що стоять на шляху ефективного виявлення внутрішніх загроз. До них належать питання продуктивності систем моніторингу, обмеженість доступних та релевантних наборів даних, проблема високої розмірності аналізованих даних, необхідність інтеграції даних про фізичну та кібернетичну поведінку користувачів, визначення оптимальних часових інтервалів для аналізу, а також фундаментальні обмеження статичних політик контролю доступу. Загальна складність виявлення внутрішніх загроз, особливо в критично важливих системах, таких як SCADA, підкреслює необхідність нових, більш інтелектуальних підходів

РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МОДЕЛІ БАЗИ ДАНИХ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ІНСТРУМЕНТУ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ

2.1. Формальна модель загроз та складність атак інсайдерів на бази даних

Атаки інсайдерів є однією з найнебезпечніших загроз для організації. На жаль, вони важко прогнозуються, виявляються та захищаються через довіру та обов'язки, покладені на співробітників. У цьому розділі ми спочатку визначаємо поняття наміру користувача та побудовуємо модель для найбільш поширеної сценарію загрози, використаного в літературі, який створює дуже високий ризик для конфіденційних даних, збережених у базі даних організації. Ми показуємо, що складність визначення псевдонамірів користувача в цій області є coNP-Complete, а запуск атаки хапальника всередині меж зазначеної моделі загроз займає лінійний час, тоді як цільова модель загроз є задачею NP-Complete.

2.1.1. Модель наміру

Запити SQL, які користувач надсилає на базу даних, можуть моделювати нормальну поведінку користувача [28]. Також запити, які схожі за своєю природою, можуть свідчити про те, що вони можуть бути видані для виконання схожих завдань. Однак розуміння наміру запиту вважається таким же складним, як створення нового запиту, і ще складнішим, коли запит є складним. Тому визначення наміру запиту часто є неоднозначним [4], і може бути різним підхід до вилучення характеристик. У літературі вилучення характеристик SQL для визначення наміру запиту досліджувалося для різних цілей, таких як оптимізація продуктивності [6], аналіз навантаження [4], рекомендації запитів [22] та цілі безпеки. З ростом потреби в доступі до

більш складної інформації конструкція запиту внутрішньо зростає в складності, що часто ускладнює розуміння отриманого запиту для людського читача. Зростання складності питання ускладнює порівняння схожості запитів щодо точності, навіть якщо вони створені для виконання одного і того ж завдання [4].

Коли ми дивимося на ці методи вилучення характеристик, вони фактично вилучають проєкції, вибірки, з'єднання, групування та сортування з SQL-запиту або їх підмножини для використання в порівнянні схожості запитів. Міра схожості може варіюватися залежно від мети методу вилучення; іноді різні запити генерують один і той же набір характеристик, а іноді запити, які мають на меті виконати одне і те ж завдання, можуть генерувати різні набори характеристик. Звичайно, результати запитів також залежать від даних, збережених у базі даних. Наприклад, запити

```
SELECT * FROM user WHERE username LIKE "A%"
SELECT * FROM user
```

дадуть абсолютно однаковий результат, якщо всі значення імен користувачів у таблиці user починаються з "A." Отже, запит можна інтерпретувати по-різному, що робить неможливим вилучити остаточний намір автора запиту з даного запиту, але все ще можна відчути нечітке уявлення про мети.

Ми прирівнюємо це нечітке уявлення до поняття псевдонаміру в аналізі формальних контекстів (FCA). Формальний контекст — це трійка K=(G,M,I), де G — це набір об'єктів, M — це набір атрибутів, а I — це відношення, яке асоціює кожен об'єкт g з атрибутами, задоволеними g. Щоб виразити, що об'єкт g знаходиться у відношенні I з атрибутом m, ми пишемо gIm [7].

Визначення того, чи є підмножина атрибутів псевдонаміром, показано як coNP-Complete [7]. Також важливо відзначити, що не всі набори атрибутів у контексті представляють псевдонамір; підрахунок кількості псевдонамірів

показано як #P-hard, тоді як знаходження кількості наборів, які не є псевдонамірами, показано як #P-Complete [31].

Отже, щоб зрозуміти псевдонамір, ми вилучаємо відповідні характеристики з SQL-запитів, і слідуючи визначенню, наданому FCA, ми визначаємо атрибути як характеристики SQL-запитів. Ці атрибути фактично є ресурсами, спожитими SQL-запитом. Ми будемо називати ці ресурси, коли ми кажемо про намір запиту (псевдонамір в області FCA), з цього моменту.

Визначення 2.1. (Намір запиту). Припускаючи, що ресурси, спожиті користувачем для виконання завдання, відображають намір користувача, ми визначаємо намір як скінченну множину ресурсів, позначену як $\phi=\{r1,r2,\ldots,r \mid \phi \mid \}$.

Це визначення безпосередньо відповідає методам вилучення характеристик, описаним вище, де ri — це вилучені характеристики запиту.

Визначення 2.2. (Активність користувача). Активність користувача A представлена користувачем $u \in U$, де U — це набір усіх користувачів, для періоду часу T, який починається з t0 і триває Δt, і набором намірів ϕ, виконаних u всередині T. Формально,

$$AuT=(aut0(\phi),aut1(\phi),\ldots,autu(\phi))$$

де auti представляє часову мітку діяльності, виконаної користувачем u.

Користувачі зазвичай створюють схожі навантаження на базу даних для виконання своїх щоденних завдань [54]. Ці навантаження можуть бути використані для створення ланцюга завдань для кожного користувача.

2.1.2. Модель загроз

Атаки безпеки класифікуються як пасивні та активні, і визначають їх наступним чином: "Пасивна атака намагається дізнатися або використовувати інформацію з системи, але не впливає на ресурси системи.

Активна атака намагається змінити ресурси системи або вплинути на їх роботу."

Тепер ми досліджуємо роботи, спрямовані на виявлення витоку даних інсайдерами за допомогою журналів запитів. Стандартні дослідження в цій галузі спрямовані на пасивних атакерів: цей тип атакерів запитує базу даних для вилучення конфіденційної інформації. Модель загроз припускає, що інсайдер не змінює дані або результати виконання на клієнтській стороні. Вони можуть отримувати доступ до системи бази даних через додаток на клієнтській стороні або через пряму взаємодію з сервером бази даних, при цьому запити спостерігаються монітором запитів.

Описувана система не блокує таку поведінку, а лише спостерігає за діяльністю запитів усіх користувачів і повідомляє персонал безпеки, якщо виявить підозрілу діяльність.

Однак наша система не розглядає активних атакерів, які змінюють дані або результати запитів, оскільки ми прагнемо запобігти витоку інформації, щоб запобігти крадіжці особистості та витоку інформації. Для запобігання зміни даних інсайдером можна використовувати техніки перевірки цілісності та аутентифіковані структури даних.

Наша модель спеціально ефективна проти атак інсайдерів на бази даних. Ми розглядаємо два пасивні сценарії атак: хапальник (також відомий як агрегатор) та цільовий (також відомий як індивідуальний) атакер. Хапальник — це супротивник, який збирає частину бази даних для дослідницького аналізу, запитуючи базу даних. Цільовий атакер — це супротивник, який отримує доступ до певної інформації без необхідності знати її для легітимних цілей, але обирає параметри атаки ретельно, щоб уникнути виявлення.

На відміну від зловмисника, який намагається отримати доступ до інформації ззовні, інсайдер зазвичай має знання про схему бази даних або доступ до програмних засобів, які працюють на базі даних без необхідності

знати основну базу даних, але не має інсайдерського знання про вміст даних у базі даних. Коли атакуючий не шукає конкретної інформації, він може надсилати дослідницькі запити до системи. Це може бути в двох формах: (1) без фільтрувальних умов або з невеликою кількістю фільтрувальних умов, або (2) багато запитів з фільтрувальними умовами. Оскільки цей тип атаки не має конкретної цільової інформації, у зловмисника є лише одна мета: вилучити якомога більше інформації. У найпростішій формі зловмисник може видати запити з підстановкою для кожної таблиці та отримати всю базу даних, виконуючи кількість таблиць $|R|$ запитів.

Розглянемо базу даних MY_BANK з відношеннями CUSTOMER, яка містить інформацію про клієнтів банку, ACCOUNT, яка містить всі рахунки, які обробляє банк, з багато-до-багатьох відношеннями, та CUSTOMER_ACCOUNTS, яка містить інформацію про те, які рахунки належать яким клієнтам, як показано на рисунку 2.1.

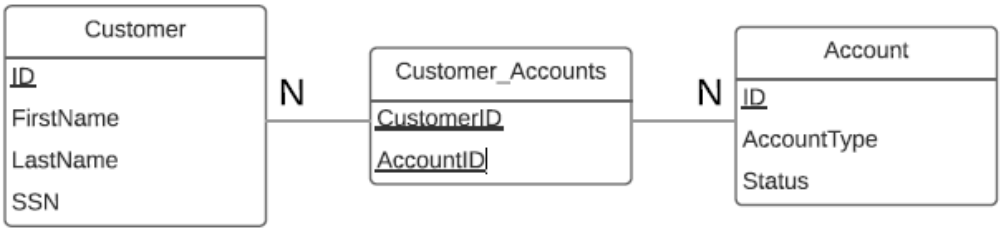

Рисунок 2.1 – Фрагмент діаграми БД My_BANK

Інсайдер, який має інформацію про схему MY_BANK, може отримати доступ до всіх даних у базі даних за допомогою лише 3 запитів: SELECT * FROM customer, SELECT * FROM account і SELECT * FROM customer_accounts.

Також можливо, щоб зловмисники використовували фільтрувальні умови, що призведе до збільшення кількості запитів, щоб уникнути виявлення. Однак механізм моніторингу запитів у підсумку зареєструє ті самі

ресурси в обох випадках. Отже, хоча може бути багато способів вилучити всі дані, складність завжди буде O(| R |).

Система моніторингу запитів (QMS) розроблена як модульна та гнучка, щоб мати змогу працювати з іншими застосунками безпеки. Вона може бути інтегрована в будь-яку СКБД лише для спостереження за рухом запитів, щоб не додавати часових витрат на обробку запитів. Після запуску та роботи будь-який додаток, який взаємодіє з базою даних, може бути відстежений.

Коли користувач додатку використовує веб- або настільний додаток на своєму комп'ютері (клієнтська сторона), додаток генерує запити та надсилає їх до бази даних. З іншого боку, кінцеві користувачі взаємодіють з базою даних через інтерфейс терміналу на своїх комп'ютерах, який безпосередньо з'єднує їх з сервером бази даних. База даних міститься на сервері бази даних (серверна сторона). QMS лише спостерігає за запитами, які надсилаються до бази даних, і не блокує або змінює їх. Будь-який запит, який надсилається до бази даних, захоплюється QMS, обробляється та реєструється там, а потім надсилається до бази даних. Хоча система не блокує жодних запитів, вона виявляє підозрілу діяльність і повідомляє про неї персонал безпеки. Загальна архітектура системи показана на рисунку 2.2, де QMS діє як спостерігач у системі.

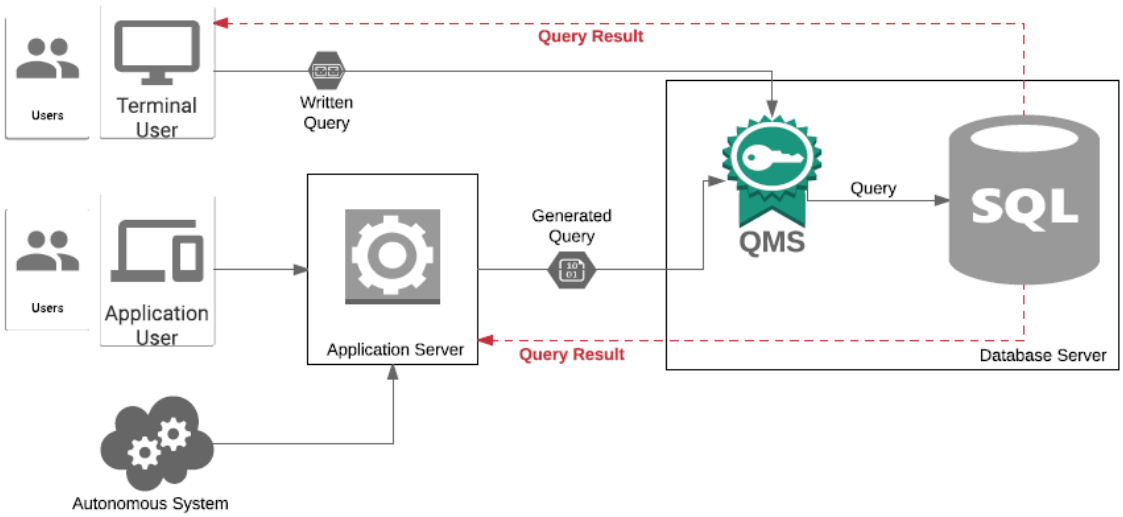

Рисунок 2.2 - Архітектура системи моніторингу запитів

Далі ми визначаємо модель загроз, проти якої побудована ця конструкція.

Атаки хапальників зосереджуються на зборі різноманітної інформації з бази даних, що призводить до поведінки пошуку, яка демонструє високий рівень різноманітності. Різноманітність вимірюється за допомогою схожості запитів та параметрів всередині сесії бази даних, тоді як широта вимірюється за допомогою різноманітності результатів повернення запитів всередині сесії бази даних.

Ми припускаємо, що супротивники всередині організації мають знання про схему бази даних або доступ до програмних засобів, які працюють на базі даних без необхідності знати основну базу даних, але не мають інсайдерського знання про вміст даних, збережених у базі даних. Коли атакуючі не шукають конкретної інформації, вони можуть надсилати дослідницькі запити до системи. Це може бути у двох формах: (1) без фільтрувальних умов або з невеликою кількістю фільтрувальних умов, або (2) багато запитів з фільтрувальними умовами. Оскільки цей тип атаки не має конкретної цільової інформації, у зловмисника є лише одна мета: вилучити якомога більше інформації. У найпростішій формі зловмисник може видати запити з підстановкою для кожної таблиці та отримати всю базу даних, виконуючи кількість таблиць $|R|$ запитів.

Розглянемо базу даних MY_BANK з відношеннями CUSTOMER, яка містить інформацію про клієнтів банку, ACCOUNT, яка містить всі рахунки, які обробляє банк, з багато-до-багатьох відношеннями, та CUSTOMER_ACCOUNTS, яка містить інформацію про те, які рахунки належать яким клієнтам, як показано на рисунку 2.1. Інсайдер, який має інформацію про схему MY_BANK, може отримати доступ до всіх даних у базі даних за допомогою лише 3 запитів: SELECT * FROM customer, SELECT * FROM account і SELECT * FROM customer_accounts.

Також можливо, щоб зловмисники використовували фільтрувальні умови, що призведе до збільшення кількості запитів, щоб уникнути виявлення. Однак механізм моніторингу запитів у підсумку зареєструє ті самі ресурси в обох випадках. Отже, хоча може бути багато способів вилучити всі дані, складність завжди буде O(｜R｜).

Аналіз запитів зазвичай покладається на структуру запитів, оскільки доступ до даних у СКБД може бути не завжди можливим для аудиторських систем або осіб, відповідальних за розслідування атак. Системи, які використовують кореляцію запитів, зазвичай використовують ресурси, до яких звертаються запити, отже, характеристики в наборі намірів можуть бути використані для вимірювання різноманітності. Дата-центричний порівняння запитів, з іншого боку, вимагає доступу до даних і може бути часомірним, оскільки він включає оцінку запитів.

Наприклад, часто підходять до аналізу журналів запитів з метою аналізу навантаження на систему. Вони вилучають терміни в селекторах, з'єднаннях, проєкціях, from, group-by та order-by окремо і записують їх частоту появи для кожного запиту в наборі даних. Вони створюють вектор характеристик, використовуючи частоту цих термінів, яку вони використовують для обчислення парної схожості запитів з функцією косинусної відстані. Метод вилучення характеристик може бути використаний для вимірювання різноманітності. Як ми вказали раніше, для механізму реєстрації немає різниці між використанням запитів з підстановкою та використанням назв ресурсів безпосередньо в запиті.

2.2. Опис залежностей в базі даних. Матриця залежностей

Схема розрахунку заробітної плати, яка використовувалася, була розроблена з метою бути якомога більш універсальною і, отже, мати схожість і застосування, подібні до моделі даних розрахунку заробітної плати

будь-якої типової організації або компанії. Умови загроз, які були визначені, припускали, що користувачі, які отримують доступ до інформації, не мали спеціальних дозволів або кваліфікацій, таких як бути співробітниками відділу кадрів. Отже, інформація, яку можна було вивести про базову заробітну плату, заробітну плату тощо, вважалася конфіденційною. Будь-які запити користувачів, які дозволили б або пряму інформацію, або інформацію, яку можна було б вивести щодо цих деталей, були відхилені.

Схема моделі даних розрахунку заробітної плати показана в наступному розділі. Короткий опис схеми, включаючи таблиці та їх атрибути, наведено нижче:

T1 - Співробітник

T2 - Посада

T3 - Заробітна плата співробітника

T4 - Календар періодів оплати

T5 - Корегування заробітної плати співробітника

T6 - Тип корегування

Першим кроком у визначенні залежностей і обмежень було створення матриці залежностей. Матриця залежностей показує залежності між різними таблицями, а також обмеження на такі залежності. Для цього проекту залежності розглядалися на рівні атрибутів окрім рівня таблиць. Всі типи залежностей спостерігаються на рівні таблиць, оскільки таблиця успадковує залежності, присутні на її рівнях атрибутів, тобто залежність між двома таблицями є залежністю між атрибутами, які належать їм. Отже, дві таблиці можуть мати більше одного типу залежності".

Оскільки рівень таблиці є найвищим рівнем гранульності, його найлегше побудувати. Ця збільшена гранульовість необхідна для повного вираження відносин між атрибутами і також надає більш реалістичне представлення запитів. Часто користувачі отримують доступ лише до атрибутів, які їх цікавлять, і не бачать цілі таблиці в запиті. З цього можна

легше побудувати залежності між атрибутами серед таблиць. Існує кілька відношень залежності між атрибутами в реляційній системі управління базами даних. Два найбільш поширені відношення залежності, які будуть обговорюватися в цій роботі, - це сильна і слабка залежності. Два елементи даних A і B мають відношення залежності між собою, якщо один з них залежить від іншого або якщо вони залежать один від одного. Відношення залежності між A і B позначається як A→BA→B, що означає, що B залежить від A. Відношення залежності класифікується за кількостю категорій, таких як сила, напрямок і транзитивність. Сила відношення залежності поділяється на два типи: слабка і сильна, які визначаються наступним чином.

Сильна залежність: при залежності A→BA→B, де A і B - два елементи даних, якщо зміна A призводить до зміни B, то це сильна залежність.

Слабка залежність: залежність A→BA→B називається слабкою, якщо зміна A не обов'язково призводить до зміни B. Значення, які генеруються при встановленні залежностей і обмежень між атрибутами в схемі, використовуються як вхідні дані в фактичній симуляції.

2.2.1. Матриця залежностей для моделі даних

Матриця залежностей, створена для моделі даних розрахунку заробітної плати, показана нижче.

	T_1	T_2	T_3	T_4	T_5	T_6
T_1	-	0	$(c_1, 2)$	0	$(c_2, 2)$	0
T_2	$(c_3, 2)$	-	$(c_4, 2)$	0	$(c_5, 1)$	0
T_3	0	0	-	0	$(c_6, 2)$	0
T_4	0	0	0	-	0	0
T_5	$(c_9, 1)$	$(c_{10}, 1)$	$(c_{11}, 1)$	0	-	0
T_6	0	0	0	0	$(c_{12}, 2)$	-

Рисунок 2.3 - Матриця залежностей для схеми розрахунку заробітної плати

Позначення (c_x,1) вказує на номер обмеження та ступінь залежності. Значення 1 вказує на сильну залежність, тоді як значення 2 вказує на слабку залежність. Описи нижче відповідають значенням обмежень вище.

2.2.2 Опис обмежень залежностей

(c_1,2) - Пряма залежність; employee_id в Employee_Salary є зовнішнім ключем до employee_id в Employee. Будь-які зміни до job_title_code, marital_status_code і dependents в Employee впливають на pay_period_id, net_pay і gross_pay в Employee_Salary. Знання marital_status впливають на net_pay в Employee_Salary.

(c_2,2) - Пряма залежність; employee_id в Employee_Pay_Adjustment є зовнішнім ключем до employee_id в Employee. adjustment_type_code і adjustment_amount будуть залежати від pay_per_period, marital_status_code і dependents. Знання marital_status дозволяє отримати інформацію про adjustment_amount в Employee_Pay_Adjustment.

(c_3,2) - Пряма залежність; job_title_code в Employee є зовнішнім ключем до job_title_code в Position_Title. job_title_code і base_pay в Position_Title впливають на pay-per-period в Employee. job_title_code буде корисним тільки якщо відома job_title.

(c_4,2) - Транзитивна залежність; base_pay в Position_Title відповідає gross_pay в Employee_Salary.

(c_5,1) - base-pay в Position_Title впливає на adjustment_amount для податків тощо в Employee_Pay_Adjustment.

(c_6,2) - Грос-плата Employee_Salary впливає на Employee_Pay_Adjustment.

(c_9,1) - На основі adjustment_type_code, adjustment_amount і adjustment_desc в Employee_Pay_Adjustment можна визначити marital_status, pay_per_period і кількість dependents в Employee.

(c_{10},1) - На основі adjustment_amount через податкові категорії тощо в Employee_Pay_Adjustment можна вивести base_pay і, відповідно, посаду конкретної особи.

(c_{11},1) - net_pay в Employee_Salary залежить від Employee_Pay_Adjustment.

(c_{12},2) - Пряма залежність; adjustment_type_code в Employee_Pay_Adjustment є зовнішнім ключем до adjustment_type_code в Adjustment_Type.

2.2.3. Граф обмежень і залежностей

На основі визначених обмежень набір атрибутів, які відповідають умовам загроз, був побудований, щоб виділити конфіденційну інформацію, яка буде розкрита неавторизованим користувачам, якщо будуть доступні всі вузли графа.

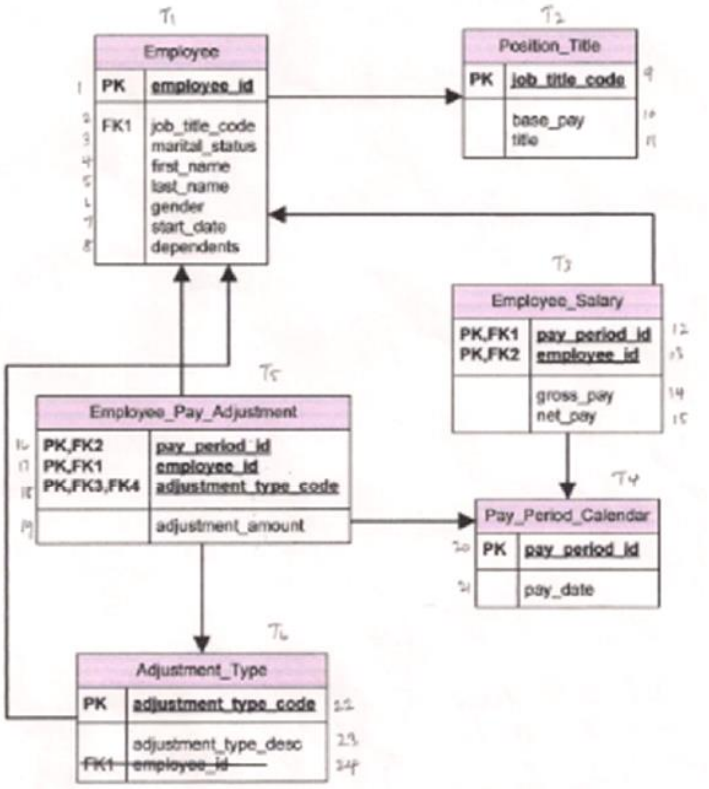

Рисунок 2.4 – Модель бази даних з пронумерованими атрибутами

Для цілей симуляції кожен атрибут моделі даних розрахунку заробітної плати був пронумерований, як показано на рисунку 2.4. Ці нумерації надають необхідну інформацію для запуску симуляції, яка створить базу знань інсайдерів і запобіжить будь-якому доступу до конфіденційної інформації. Оскільки припускається, що користувачі не мають жодних попередніх спеціальних авторизацій, вони не зможуть виконувати жодні транзакції, які дозволяють отримати відповідну інформацію щодо будь-яких загроз, перелічених вище.

Формат для переліку залежностей і обмежень, які розкривають конфіденційну інформацію, показаний нижче. Це буде введено в симуляцію для представлення схеми та потенційних загроз.

2.3. Визначення залежностей на конфіденційну інформацію. Побудова універсальної схеми

Визначимо залежності на конфіденційну інформацію:

Умова 1: Базова заробітна плата співробітника

Атрибути: 1,2,10,4,51,2,10,4,5

Умова 2: Грос-плата на основі дати початку

Атрибути: 1,2,10,4,5,71,2,10,4,5,7

Умова 3: Кількість осіб на утриманні конкретного співробітника

Атрибути: 17,22,23,10,1,4,5,8,1817,22,23,10,1,4,5,8,18

Умова 4: Чиста заробітна плата співробітника за конкретний місяць

Атрибути: 17,19,10,21,16,1,4,5,20,717,19,10,21,16,1,4,5,20,7

Умова 5: Сімейний стан співробітника

Атрибути: 1,4,5,3,17,18,19,22,231,4,5,3,17,18,19,22,23

Загальна кількість атрибутів для схеми: 23

Додаткові зауваження щодо умов загроз:

- Знання кількості осіб на утриманні можна використовувати для визначення суми страхування для конкретної особи.

- Знання сімейного стану особи дозволяє дізнатися про її податкову категорію, що може бути використано для виведення грос-плати для цього співробітника.

Універсальна схема була випадково згенерована після встановлення значень для загальної кількості атрибутів і кількості умов загроз. Універсальна схема була створена як зразок схеми для порівняння результатів симуляції з моделлю даних розрахунку заробітної плати. Універсальна схема порівняно з моделлю розрахунку заробітної плати має лише 12 атрибутів.

Обмеження залежностей для цієї схеми показані нижче:

Значення атрибутів загроз:

Умова 1: 1, 3, 5, 6, 7, 12

Умова 2: 2, 6, 7, 11, 12

Умова 3: 4, 5, 7, 1, 2, 9

У цій роботі симуляції, виконані для універсальної схеми, аналогічні симуляціям для моделі даних розрахунку заробітної плати. Єдиною відмінністю є те, що діапазон значень, які тестуються, змінюються відповідно до розміру схеми. Наприклад, для універсальної схеми встановлено менший діапазон кількості атрибутів, до яких здійснюється доступ для даної транзакції, через відносно невелику кількість атрибутів.

2.4. Представлення та опис моделі бази даних

Ця модель бази даних призначена для управління інформацією про заробітну плату співробітників. Вона складається з шести таблиць, пов'язаних між собою через первинні (PK) та зовнішні (FK) ключі (рис. 2.5).

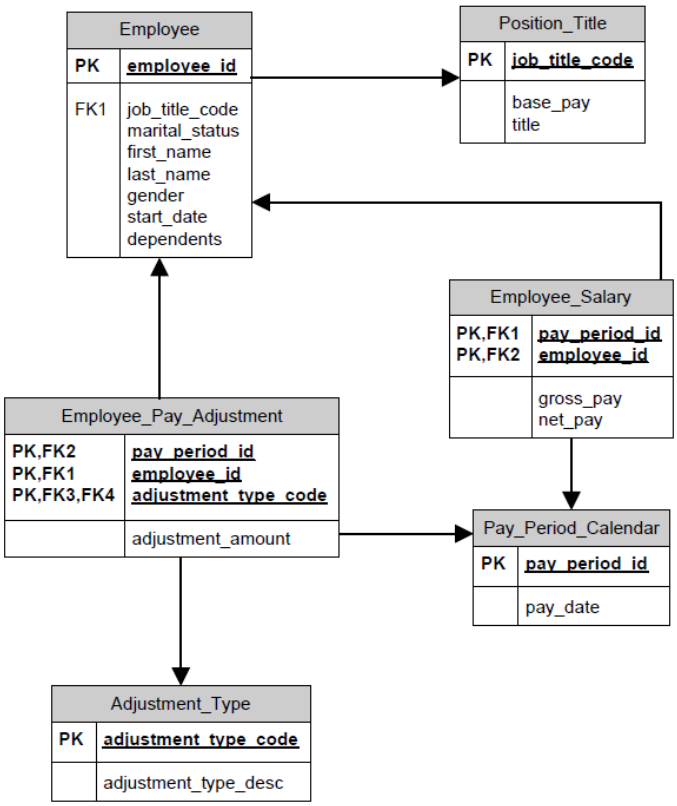

Рисунок 2.5 – Модель бази даних

Ось опис кожної таблиці та їхніх зв'язків:

1. Employee (Співробітник):

- PK employee_id: Первинний ключ, унікальний ідентифікатор кожного співробітника.

- job_title_code**: Зовнішній ключ (FK1), що посилається на таблицю Position_Title, вказує на посаду співробітника.

- marital_status: Сімейний стан співробітника.

- first_name: Ім'я співробітника.

- last_name: Прізвище співробітника.

- gender: Стать співробітника.

- start_date: Дата початку роботи співробітника.

- dependents: Кількість утриманців співробітника.

Зв'язки:

- Один співробітник може мати одну посаду (Employee до Position_Title - зв'язок "один до одного" або "багато до одного", залежно від того, чи може одна посада належати кільком співробітникам).

- Один співробітник може мати багато записів про коригування заробітної плати (Employee до Employee_Pay_Adjustment - зв'язок "один до багатьох").

- Один співробітник може мати багато записів про заробітну плату за різні періоди (Employee до Employee_Salary - зв'язок "один до багатьох").

2. Position_Title (Назва посади):

- PK job_title_code: Первинний ключ, унікальний код кожної посади.

- base_pay: Базова ставка оплати для цієї посади.

- title: Назва посади.

Зв'язки:

- Одна посада може бути пов'язана з багатьма співробітниками (Position_Title до Employee - зв'язок "один до багатьох").

3. Employee_Salary (Заробітна плата співробітника):

- PK, FK1 pay_period_id: Частина складеного первинного ключа та зовнішній ключ, що посилається на таблицю Pay_Period_Calendar, вказує на період оплати.

- PK, FK2 employee_id: Частина складеного первинного ключа та зовнішній ключ, що посилається на таблицю Employee, вказує на співробітника.

- gross_pay: Загальна сума нарахованої заробітної плати за період.

- net_pay: Сума заробітної плати до видачі після всіх відрахувань за період.

Зв'язки:

- Для кожного співробітника може бути багато записів про заробітну плату за різні періоди оплати (Employee до Employee_Salary - зв'язок "один до багатьох").

- Для кожного періоду оплати може бути багато записів про заробітну плату різних співробітників (Pay_Period_Calendar до Employee_Salary - зв'язок "один до багатьох").

4. Pay_Period_Calendar (Календар періодів оплати):

- PK pay_period_id: Первинний ключ, унікальний ідентифікатор кожного періоду оплати.

- pay_date: Дата виплати заробітної плати за цей період.

Зв'язки:

- Один період оплати може бути пов'язаний з багатьма записами про заробітну плату співробітників (Pay_Period_Calendar до Employee_Salary - зв'язок "один до багатьох").

- Один період оплати може бути пов'язаний з багатьма записами про коригування заробітної плати співробітників (Pay_Period_Calendar до Employee_Pay_Adjustment - зв'язок "один до багатьох").

5. Employee_Pay_Adjustment (Коригування заробітної плати співробітника):

- PK, FK2 pay_period_id: Частина складеного первинного ключа та зовнішній ключ, що посилається на таблицю Pay_Period_Calendar, вказує на період оплати, до якого відноситься коригування.

- PK, FK1 employee_id: Частина складеного первинного ключа та зовнішній ключ, що посилається на таблицю Employee, вказує на співробітника, для якого проводиться коригування.

- PK, FK3 adjustment_type_code: Частина складеного первинного ключа та зовнішній ключ, що посилається на таблицю Adjustment_Type, вказує на тип коригування.

- adjustment_amount: Сума коригування (може бути як позитивною, так і негативною).

Зв'язки:

- Для кожного співробітника може бути багато записів про коригування заробітної плати за різні періоди та різних типів (Employee до Employee_Pay_Adjustment - зв'язок "один до багатьох").

- Для кожного періоду оплати може бути багато записів про коригування заробітної плати різних співробітників та різних типів (Pay_Period_Calendar до Employee_Pay_Adjustment - зв'язок "один до багатьох").

- Для кожного типу коригування може бути багато записів про коригування заробітної плати різних співробітників та в різні періоди (Adjustment_Type до Employee_Pay_Adjustment - зв'язок "один до багатьох").

6. Adjustment_Type (Тип коригування):

- PK adjustment_type_code: Первинний ключ, унікальний код кожного типу коригування.

- adjustment_type_desc: Опис типу коригування (наприклад, премія, штраф, відрахування).

Зв'язки:

- Один тип коригування може бути пов'язаний з багатьма записами про коригування заробітної плати співробітників (Adjustment_Type до Employee_Pay_Adjustment - зв'язок "один до багатьох").

Ця модель бази даних дозволяє зберігати та керувати всією необхідною інформацією для розрахунку та виплати заробітної плати співробітникам, включаючи основні дані про співробітників, їхні посади, періоди оплати, нарахування та відрахування (через механізм коригувань).

2.5. Висновки до розділу

В даному розділі було здійснено всебічне представлення моделі бази даних, призначеної для розробки програмного інструменту захисту від несанкціонованого доступу. Розгляд формальної моделі загроз, зокрема

моделей наміру та загроз, дозволив чітко сформулювати потенційні сценарії зловживань з боку інсайдерів і визначити рівень складності можливих атак.

Подальший аналіз структури бази даних охопив побудову матриці залежностей, яка є ключовим елементом для розуміння взаємозв'язків між даними. Визначення обмежень, побудова графа залежностей та обмежень дали змогу формалізувати механізми контролю доступу на структурному рівні. Це забезпечує основу для подальшої реалізації політик безпеки.

Особливу увагу було приділено залежностям, пов'язаним із конфіденційною інформацією, що дозволило виокремити критичні елементи даних та побудувати універсальну схему їхнього захисту. На завершення, представлення повної моделі бази даних забезпечило цілісне бачення структури з урахуванням вимог до безпеки та цілісності даних.

Таким чином, результати розділу закладають концептуальну та технічну основу для подальшої реалізації ефективного засобу захисту баз даних від інсайдерських загроз

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ В БАЗАХ ДАНИХ

3.1. Опис процесу імітації несанкціонованого доступу

Імітація (симуляція) була розроблена з метою надати адміністраторам практичний спосіб застосування досліджень, проведених щодо прогнозування та запобігання загрозам, для визначення областей, де загрози найімовірніше виникнуть. На основі цієї інформації вони можуть надати права доступу, які надають користувачам максимальну свободу, зберігаючи при цьому безпеку системи. Симуляція також дає уявлення про те, як довго потрібно для отримання неавторизованих знань про елементи даних для різних типів реляційних схем баз даних, і виявляє, які області або умови найбільш сприйнятливі до внутрішніх загроз.

Такий процес зазвичай називають тестуванням на проникнення з позиції інсайдера або симуляцією інсайдерської загрози. Його мета - виявити слабкі місця в системі безпеки, які можуть бути використані співробітниками організації (або особами, які мають легітимний доступ до ресурсів) для отримання несанкціонованого доступу до конфіденційних даних.

Рисунок 3.1 - Етапи процесу імітації несанкціонованого доступу до БД

Наведемо типовий опис такого процесу (рис. 3.1):

1. Планування та підготовка (Planning and Preparation).

Насамперед визначаються конкретні цілі тестування (наприклад, отримати доступ до певної таблиці з персональними даними клієнтів, змінити записи про запаси на складі, експортувати фінансову звітність). Також окреслюється обсяг робіт, тобто які бази даних, системи та облікові записи будуть залучені.

Розробляється детальний сценарій дій інсайдера. Це може включати:

- Використання існуючого облікового запису з певними правами доступу.

- Спроби підвищення своїх привілеїв (наприклад, через експлуатацію вразливостей у програмному забезпеченні або помилки конфігурації).

- Зловживання легітимними правами доступу для виконання неавторизованих дій.

- Змова з зовнішнім зловмисником (хоча це може виходити за рамки суто "інсайдерського" тестування).

Потім формується команда фахівців з кібербезпеки, які будуть імітувати дії інсайдера. Важливо, щоб ці особи мали достатні технічні знання та розуміння внутрішніх процесів організації. Залежно від сценарію, можуть знадобитися спеціалізовані інструменти для сканування вразливостей, SQL-ін'єкцій, аналізу трафіку, а також тестові облікові записи та середовища.

Узгодження правил взаємодії (Rules of Engagement): Визначаються чіткі правила проведення тестування, включаючи дозволені дії, часові рамки, канали комунікації та процедури звітності. Важливо мінімізувати ризики для реальних систем та даних.

2. Виконання імітації (Execution).

Спочатку команда тестувальників починає виконувати запланований сценарій атаки, діючи подібно до реального інсайдера. Це може включати:

- Вхід в систему під існуючим або створеним тестовим обліковим записом.

- Вивчення структури бази даних, таблиць, прав доступу, процедур та інших об'єктів.

- Виконання SQL-запитів, спроби обходу механізмів контролю доступу, використання вразливостей у веб-інтерфейсах або інших точках доступу до БД.

- Спроби отримати вищі рівні доступу, ніж ті, що призначені для облікового запису.

- Читання, модифікація або видалення даних, виконання шкідливих процедур.

- Копіювання конфіденційної інформації.

- Фіксація всіх дій, щоб у подальшому проаналізувати можливість виявлення атаки.

3. Аналіз результатів та звітування (Analysis and Reporting).

Під час виконання імітації фіксуються всі успішні та неуспішні спроби несанкціонованого доступу, отримані дані, використані вразливості та інші важливі спостереження. Виявляються слабкі місця в системі безпеки, які дозволили інсайдеру здійснити атаку (наприклад, недостатній контроль доступу, слабкі паролі, відсутність моніторингу, помилки в конфігурації). Складається детальний звіт про результати тестування. Звіт зазвичай включає:

- Опис проведених дій.

- Перелік виявлених вразливостей.

- Опис успішно отриманого несанкціонованого доступу та виконаних дій.

- Оцінку ризиків, пов'язаних з виявленими вразливостями.

- Рекомендації щодо усунення виявлених проблем та покращення системи безпеки.

4. Подальші дії (Remediation and Follow-up).

На основі рекомендацій зі звіту вживаються заходи для усунення виявлених слабких місць у системі безпеки. Переглядаються та вдосконалюються політики та процедури контролю доступу, моніторингу, аудиту та реагування на інциденти. Після впровадження змін може бути проведене повторне тестування для перевірки ефективності вжитих заходів.

Важливо зазначити, що імітування несанкціонованого доступу інсайдера проводиться етично та з повним розумінням потенційних ризиків. Необхідно отримати дозвіл від керівництва організації та дотримуватися узгоджених правил взаємодії, щоб уникнути ненавмисного порушення роботи реальних систем або розкриття конфіденційної інформації.

Дана імітація була розроблена таким чином, щоб дозволити виконувати і зберігати для подальшого аналізу кілька схем. Імітація також дозволяє змінювати кілька параметрів для тестування, включаючи встановлення загальної кількості атрибутів. Будь-хто, хто використовує симуляцію, може змінювати кількість користувачів, які виконують транзакції, визначати загальну кількість дозволених транзакцій і вказувати мінімальну та максимальну кількість атрибутів, до яких здійснюється доступ для даної транзакції. Це випадково генерується на основі встановлених мінімальних і максимальних значень.

3.2. Реалізація інтерфейсу системи

3.2.1 Введення схеми

Вкладка введення схеми дозволяє користувачам надати деталі щодо схеми, на якій вони запускають імітації. Користувач спочатку вводить загальну кількість атрибутів, які є в таблицях для даної схеми бази даних. Комбінація атрибутів загроз відповідає списку атрибутів, які, якщо інсайдер без спеціальних дозволів має до них доступ, можуть розкрити

неавторизовану конфіденційну інформацію. Щоб надати комбінації атрибутів загроз, слід спочатку пронумерувати всі атрибути для даної схеми. На основі існуючих залежностей і обмежень атрибути, які при агрегації надають конфіденційну інформацію, потім вводяться, як показано нижче, кожна умова загрози відповідає одному рядку у списку вкладки Insider Threat Schema.

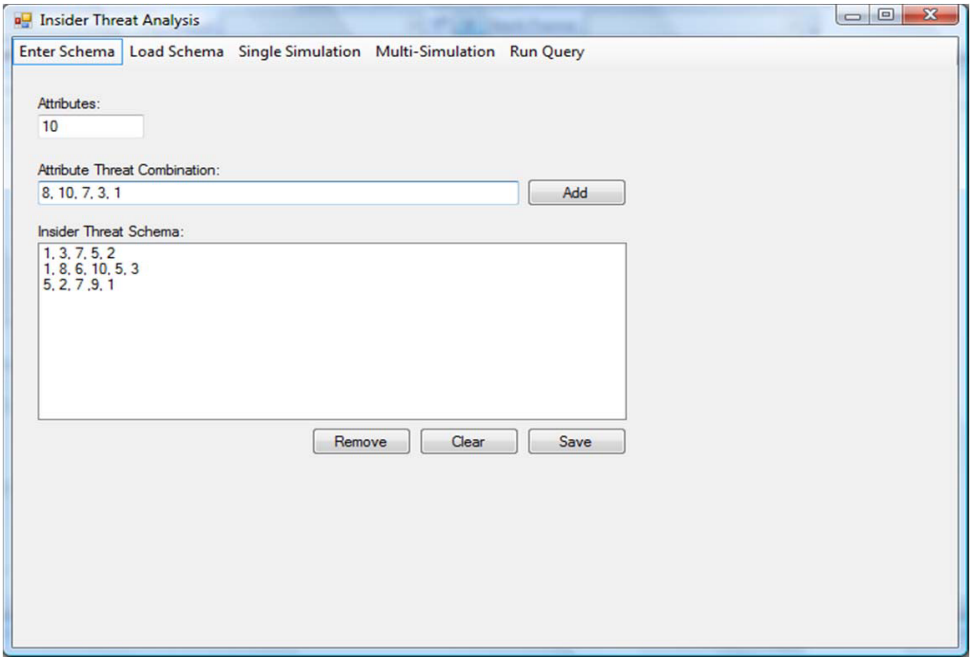

Рисунок 3.2 – Приклад введення схеми

Користувачам необхідно зберегти схему після її введення, щоб не потрібно було повторно вводити схему при запуску подальших симуляцій на вказаному списку умов загроз.

3.2.2. Завантаження схеми

Збережені схеми, які були введені на попередній вкладці, можна відкрити на вкладці завантаження схеми. Відкрита схема відображається для перевірки, чи відповідають умови загроз зазначеним. Перший рядок показує загальну кількість атрибутів, а умови загроз перелічені через кому в наступних рядках. Значення, надане для загальної кількості атрибутів,

використовується для випадкової генерації транзакцій читання для користувачів у симуляції. Схему потрібно відкрити та завантажити, перш ніж користувач зможе виконати або одиничну, або мультисимуляцію.

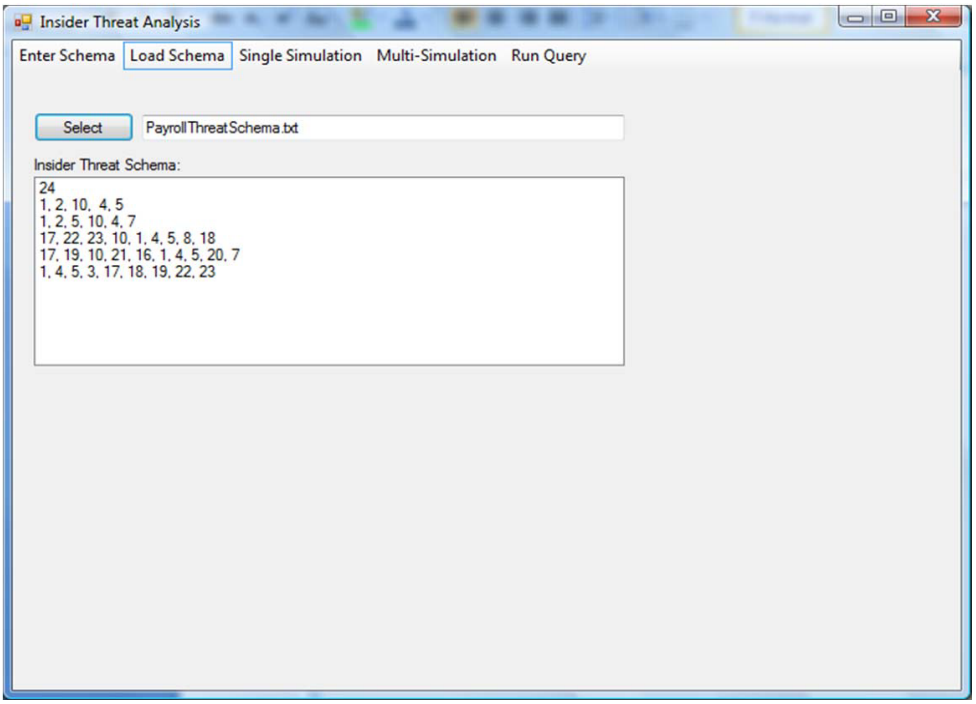

Рисунок 3.3 – Приклад завантаження схеми

3.2.3. Проведення одиничної симуляції

Після того, як користувач вказав схему для імітації, можна виконати одиничний запуск. Потрібно вказати наступні параметри: користувачі, транзакції та діапазон кількості атрибутів на транзакцію. Кількість користувачів, яку надає користувач, використовується для випадкової симуляції загальної кількості вказаних транзакцій. Кількість атрибутів на транзакцію також генерується випадково на основі вказаного діапазону. Отже, як показано на прикладі нижче, загальна кількість читань, які відбуваються для кожної транзакції для будь-якого користувача, дозволяє доступ до 5-7 елементів. Для цілей імітації ми розглядаємо транзакції лише з читанням, а не з записом, щоб уникнути додаткових складнощів, таких як проблеми з оновленнями тощо, які виходять за межі цієї роботи.

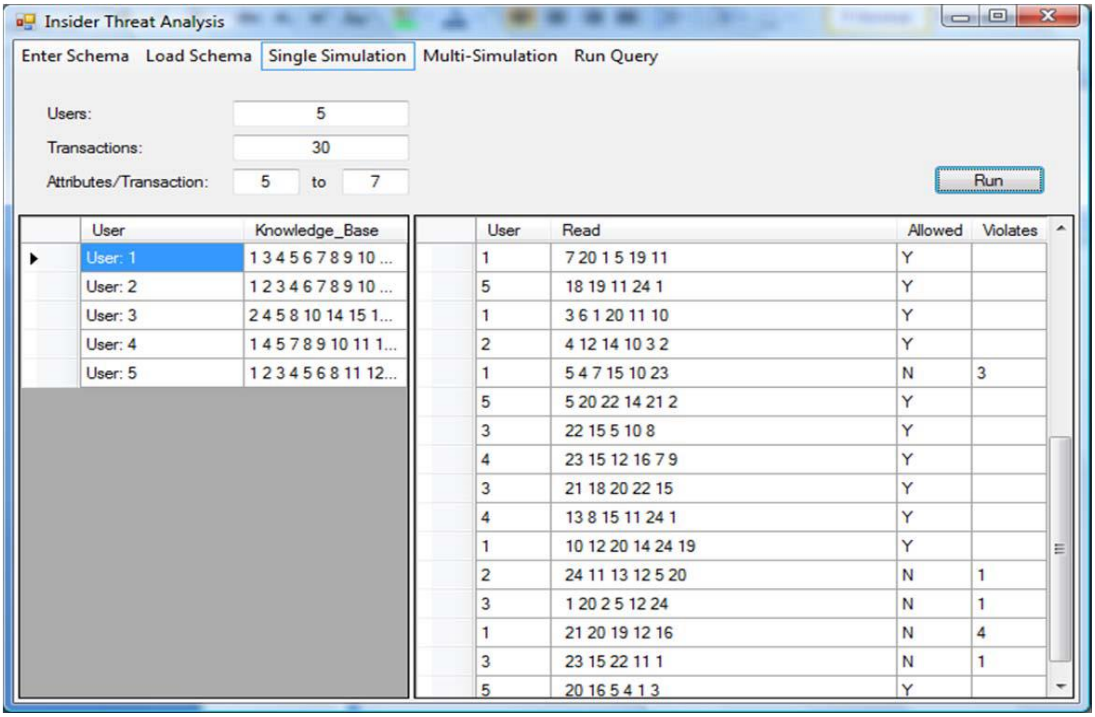

Рисунок 3.4 – Приклад проведення одиничної імітації

При виконанні одиничної імітації лівій частині відображається загальна база знань користувача на основі накопиченої інформації про доступ з усіх транзакцій для цього користувача. Права частина відображає випадкову імітацію загальної кількості транзакцій, розподілених випадково між користувачами. Стовпець "Дозволено" показує, чи була схвалена дана транзакція. Якщо транзакція була відхилена, стовпець "Порушує" показує, яка умова загрози була порушена, що завадило виконанню транзакції.

3.2.4. Мультисимуляція

Мультисимуляція використовує ті самі параметри, що й одинична імітація, за винятком того, що мультиімітація додає додаткову складність, дозволяючи змінювати дві змінні через вісь x і серії, тоді як третя змінна залишається сталою. Наприклад, на прикладі нижче кількість користувачів встановлено на вісь x, що змінюється з кроком 5 і в діапазоні від 5 до 25. Транзакції встановлено як змінна, яка змінюється через серії. У цьому прикладі загальна кількість транзакцій встановлено в діапазоні від 50 до 200 з

кроком 50. Нарешті, третя змінна, кількість атрибутів на транзакцію, встановлено, як і раніше, в діапазоні від 5 до 7. Поле для введення "Кількість запусків на точку даних" використовується для встановлення кількості виконань для кожного набору параметрів. Середнє значення береться з усіх запусків для кожного набору, щоб отримати найбільш послідовні та точні результати. Очікувалося, що збільшення кількості транзакцій для даної кількості користувачів призведе до більшої кількості відхилених транзакцій.

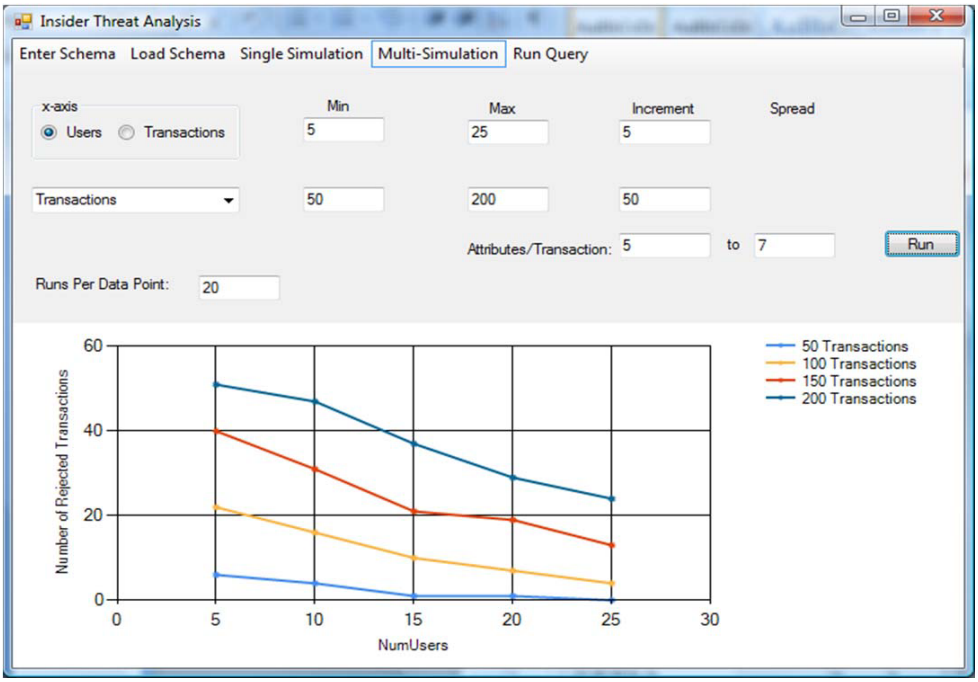

Рисунок 3.5 – Приклад виконання мультисимуляції

3.3. Представлення результатів моделювання процесу несанкціонованого доступу

3.3.1. Одиничний запуск схеми

Наступні дані показують детальніше виконання запуску симуляції загроз за допомогою схеми розрахунку заробітної плати. Наступні параметри були встановлені при запуску симуляції:

- Користувачі: 5

- Транзакції: 30

- Елементи, до яких здійснюється доступ на транзакцію: 5-7

Результати, отримані при запуску вищезазначеної симуляції, були наступними, що показані в таблиці 3.1.

Таблиця 3.1 - Приклад виконаних транзакцій на користувача

Користувач	**Читання**	**Дозволено**	**Порушує**
2	191418424	Так	
2	18122413	Так	
5	1315128514	Так	
5	12915218	Так	
4	1715461213	Так	
3	7121861911	Так	
2	1921051	Ні	1
1	71922010	Так	
3	1886139	Так	
3	11824218	Так	
2	1822624239	Так	
3	24651820	Так	
5	361311724	Так	
4	810471714	Так	
2	23415142018	Так	
2	487221520	Так	
4	12206225	Так	
5	981613511	Так	
4	13171812198	Так	
5	1311211618	Так	
2	2122323179	Так	
5	101618172	Так	
3	11323194	Так	
2	10165218	Ні	3
5	1691419218	Так	
3	1017152011	Так	
3	14192012222	Ні	1
2	172432191	Так	
4	221671514	Так	
3	1225146	Ні	1

База знань користувачів, яка була побудована під час читання кількох транзакцій і доступу до більшої кількості атрибутів, показана нижче, в таблиці 3.2.

Таблиця 3.2 - Приклад бази знань користувача

Користувач: 1	27101920
Користувач: 2	12346789131415171819202122 23
Користувач: 3	13456789101112131718192021 23
Користувач: 4	45678101213141516171819202 2
Користувач: 5	12356789101112131415161718 1921

Результати показують, що конфіденційна інформація з умови загрози 1 найімовірніше буде виявлена через випадковий доступ. Це узгоджується з обмеженнями та залежностями, накладеними на модель даних розрахунку заробітної плати. Умова загрози 1 складається лише з 5 атрибутів, що свідчить про те, що критична інформація може бути отримана, якщо доступ до всіх 5 атрибутів буде отримано. Інші умови загроз вимагають знання більше п'яти атрибутів. Отже, умови загроз, які можна отримати лише через агрегацію інформації з кількох атрибутів, мають меншу ймовірність бути порушеними через випадкові доступи.

3.3.2. Кількість користувачів у порівнянні з відхиленими транзакціями

Схема розрахунку заробітної плати та універсальна схема були симульовані в кількох запусках для визначення кількості відхилених транзакцій, зберігаючи кількість транзакцій сталою і змінюючи кількість користувачів. Кілька запусків дозволили порівняти результати для різної кількості виконаних транзакцій.

Результати кожного запуску були отримані шляхом усереднення 100 запусків на точку даних для забезпечення послідовності. Кількість атрибутів, до яких здійснюється доступ на транзакцію, залишалася сталою між діапазоном 3-5 і генерувалася випадково серед цих значень для кожної

транзакції в цій симуляції. Результати для схеми розрахунку заробітної плати показані нижче.

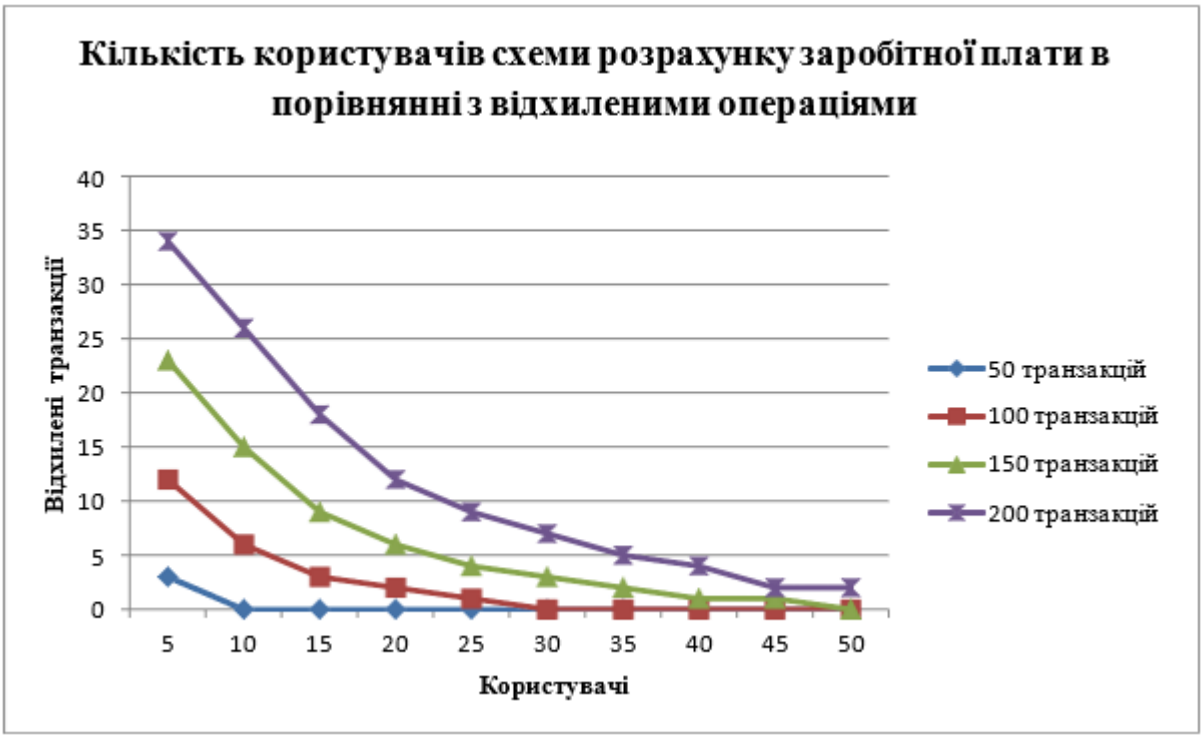

Рисунок 3.6 - Графік схеми нарахування заробітної плати, кількість користувачів у порівнянні з відхиленими трансакціями

Існує кореляція між відсотком відхилених транзакцій і загальною кількістю транзакцій, яка збільшується відносно кількості користувачів. Наприклад, кількість відхилених транзакцій для 5 користувачів, незалежно від збільшення загальної кількості транзакцій, становила приблизно 1/61/6 від усіх транзакцій. Виконання 150 транзакцій для 5 користувачів призвело до середнього 23 відхилених (23/150=15%), тоді як виконання 200 транзакцій призвело до 34 відхилених (17%).

Можливе пояснення цього полягає в тому, що після додавання більшості інформації зі схеми до бази знань, наші критерії для відхилення будь-якої транзакції, яка призведе до виявлення конфіденційної інформації, призводять до виключення одного атрибута з кожної умови загрози. Це

приблизно відповідає загальній кількості відхилених транзакцій. Оскільки було 24 атрибути для схеми даних розрахунку заробітної плати з 4 умовами загроз, (4/24=16%) надає досить точне число відхилених транзакцій відносно кількості виконаних транзакцій.

Ця інформація може бути корисною для створення порогового значення для доступу користувачів до бази даних. Знаючи, що 16% транзакцій відхиляються через те, що можна отримати конфіденційну інформацію, користувачам слід обмежити доступ лише до (100%−16%)=84% у випадкових доступах, щоб запобігти можливості виявлення будь-якої конфіденційної інформації. Результати для універсальної схеми показані нижче.

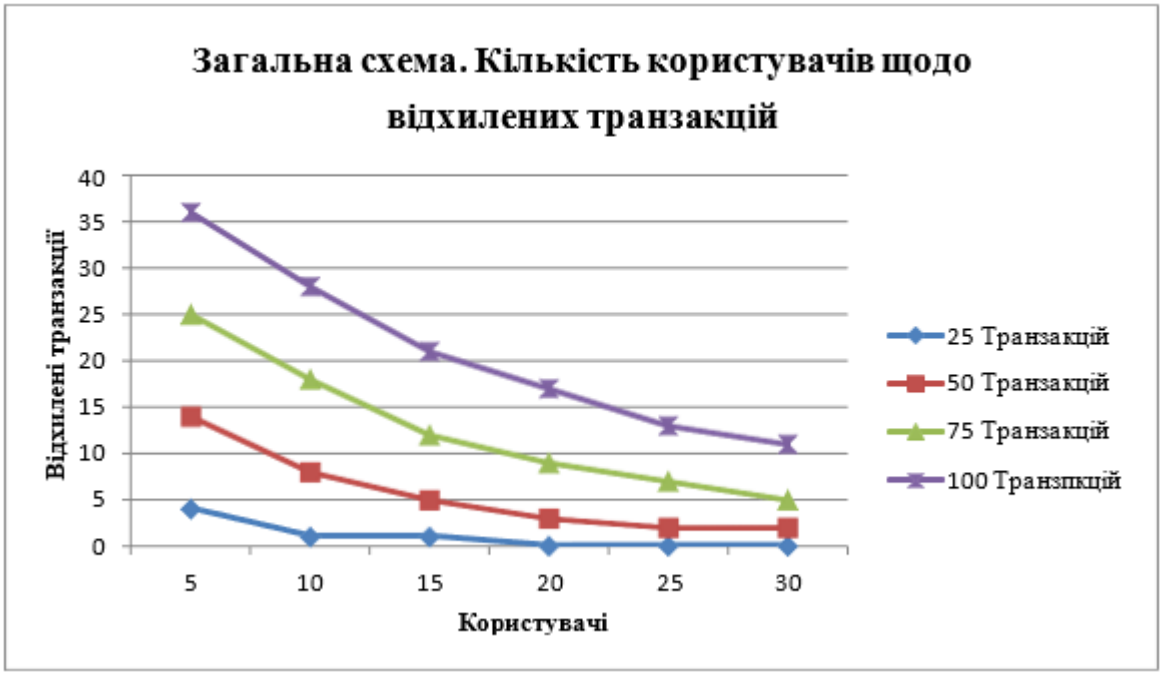

Рисунок 3.7 - Графік загальної схеми, кількість користувачів у порівнянні з відхиленими транзакціями

Загальні висновки, які можна зробити, порівнюючи результати, отримані за допомогою універсальної схеми з моделлю даних розрахунку заробітної плати, показують, що менша кількість користувачів, які

отримують доступ до бази даних, швидше отримують конфіденційну інформацію. Це можна побачити на рисунку 3.7, де коли кількість користувачів знаходиться в діапазоні від 5 до 15, більша кількість запитів транзакцій, ймовірно, буде відхилена. Це число вирівнюється більше, коли кількість транзакцій, які виконуються, збільшується відносно кількості користувачів.

3.3.3. Кількість транзакцій у порівнянні з відхиленими транзакціями

Симуляції були проведені для тестування зміни кількості транзакцій при сталій кількості користувачів. Очікувалося, що результати будуть зворотними до вищезазначених графіків для двох схем відповідно. Як і раніше, результати кожного запуску були отримані шляхом усереднення 100 запусків на точку даних, а кількість атрибутів, до яких здійснюється доступ на транзакцію, генерувалася випадково між діапазоном 3-5. Результати запуску симуляції для схеми розрахунку заробітної плати показані нижче.

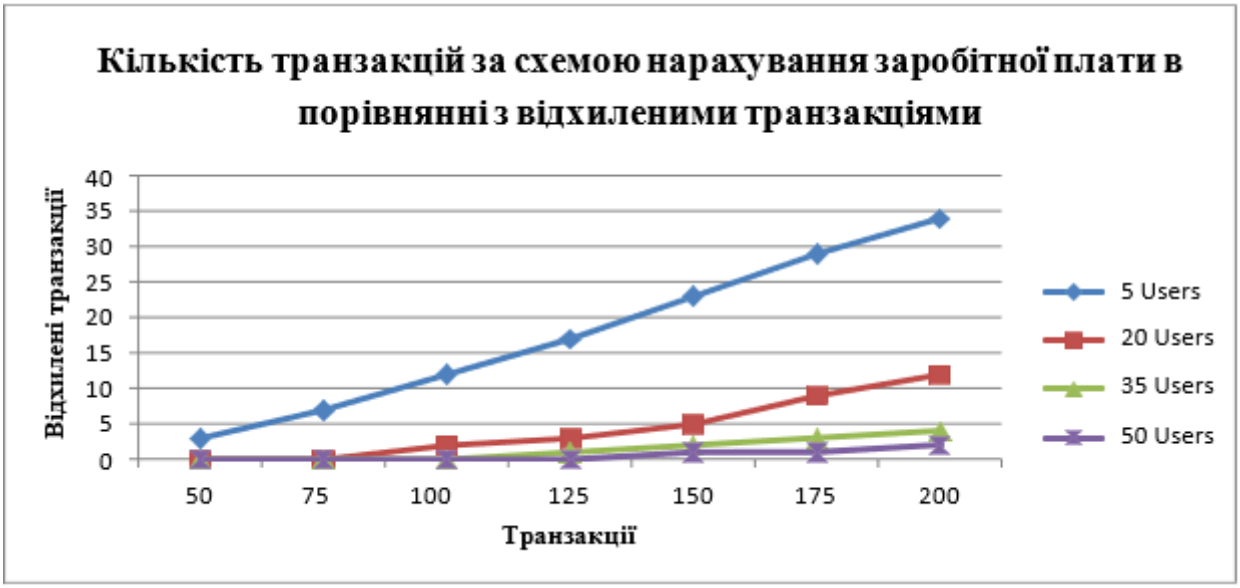

Рисунок 3.8 – Графік кількості транзакцій за схемою нарахування заробітної плати в порівнянні з відхиленими транзакціями

Результати запуску симуляції для універсальної схеми показані нижче.

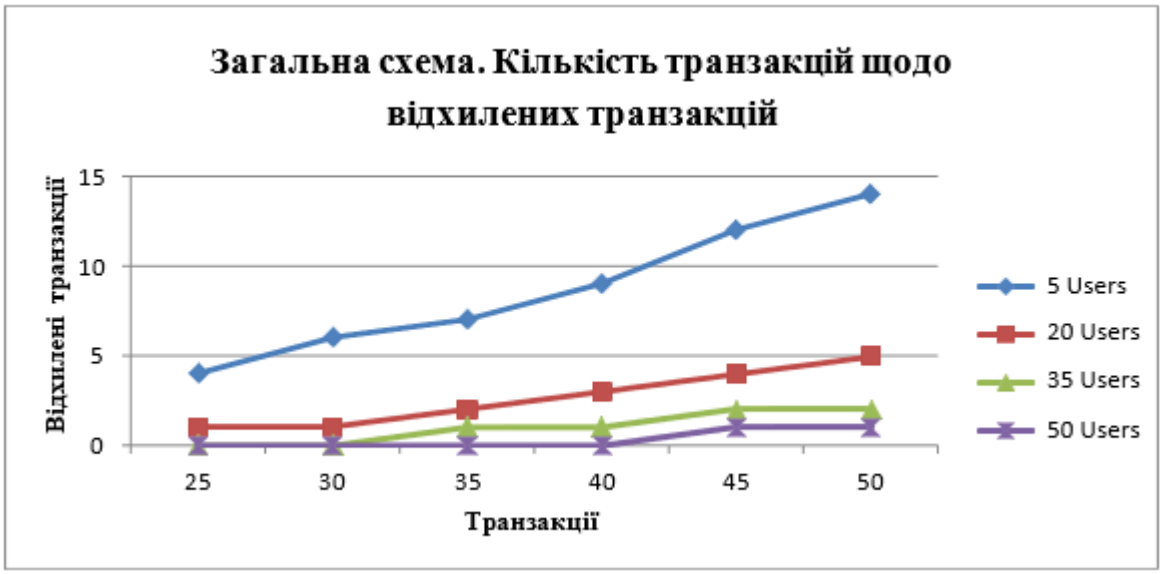

Рисунок 3.9 - Графік загальної схеми, кількість транзакцій у порівнянні з відхиленими транзакціями

Як очікувалося, аналіз значень показав, що моделювання, проведені для порівняння кількості транзакцій з загальною кількістю відхилених транзакцій, мали пряму зворотну залежність до кількості користувачів і кількості відхилених транзакцій. Це можна інтуїтивно зрозуміти, що збільшення кількості транзакцій для фіксованої кількості користувачів призводить до того, що вони швидше розвивають свою базу знань через випадкові доступи і, ймовірно, запитають транзакції, які дозволять їм визначити конфіденційну інформацію.

3.3.4. Кількість елементів, доступ до яких отримано, порівняно з відхиленими транзакціями

Було проведено тестове моделювання для визначення, чи існує кореляція між кількістю елементів, до яких здійснюється випадковий доступ в межах заданого діапазону, і загальною кількістю відхилених транзакцій. Ця симуляція була проведена з усіма іншими параметрами кількості користувачів проти відхилених транзакцій, за винятком зміни діапазону атрибутів, до яких здійснюється доступ на транзакцію, на 5-7 з 3-5.

Наведений нижче графік був створений в результаті запуску симуляції.

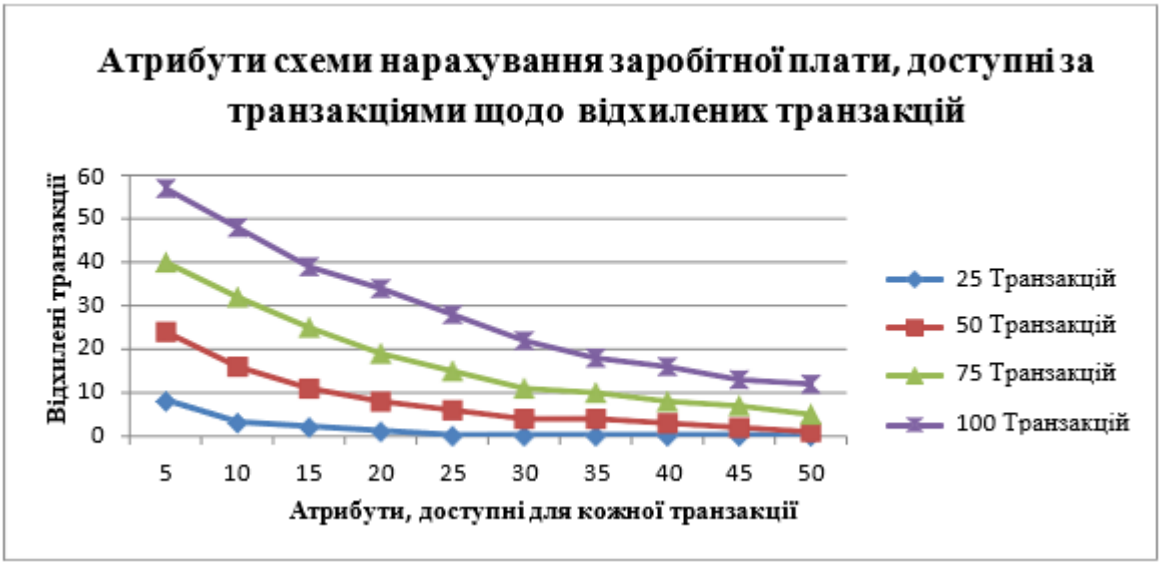

Рисунок 3.9 - Графік схеми нарахування заробітної плати, атрибути/транзакції в порівнянні з відхиленими транзакціями

Порівняння результатів графіка з початковим графіком, на якому було встановлено лише 3-5 елементів для випадкового доступу на транзакцію, показало, що більше транзакцій було відхилено для тієї ж кількості користувачів і загальної кількості транзакцій, коли кількість доступів до елементів даних знаходилася в діапазоні 5-7. Це узгоджується з припущенням, що чим більше атрибутів дозволяється користувачам доступати в даній транзакції, тим швидше вони будуть будувати свою базу знань і, відповідно, збільшувати ймовірність визначення конфіденційної інформації.

3.4. Виконання імітації запиту

Була розроблена симуляція запитів, яка аналізує та приймає або відхиляє SQL-запит користувача. На основі існуючої бази знань користувача запит або виконується і повертає очікувані результати, або не виконується,

якщо можна отримати конфіденційну інформацію. Якщо запит транзакції дозволено, він передається до бази даних, і SQL-запит виконується. Результати відображаються у вікні симуляції запитів. На даний момент підтримується лише оператор SELECT. У майбутньому розробки можуть включати підтримку операторів оновлення. Це забезпечує проактивний метод запобігання будь-яких порушень інсайдерів неавторизованою інформацією.

Симуляція запитів працює аналогічно одиничній симуляції та мультисимуляції, будуючи базу знань користувача під час надання доступу до даних. Виконання транзакції блокується, якщо запитані поля дозволяють користувачеві вивести будь-яку конфіденційну інформацію. Симуляція запитів використовує General SQL Parser, комерційно доступний SQL Parser, який визначає атрибути, до яких здійснюється доступ у SELECT і WHERE клаузах SQL-запиту. Поточна реалізація використовує SQL Server. На даний момент симуляція запитів є прототипом, але її можна реалізувати як мережеву службу, яка дозволить застосункам отримувати доступ до неї, а не просто надавати інтерфейс користувача.

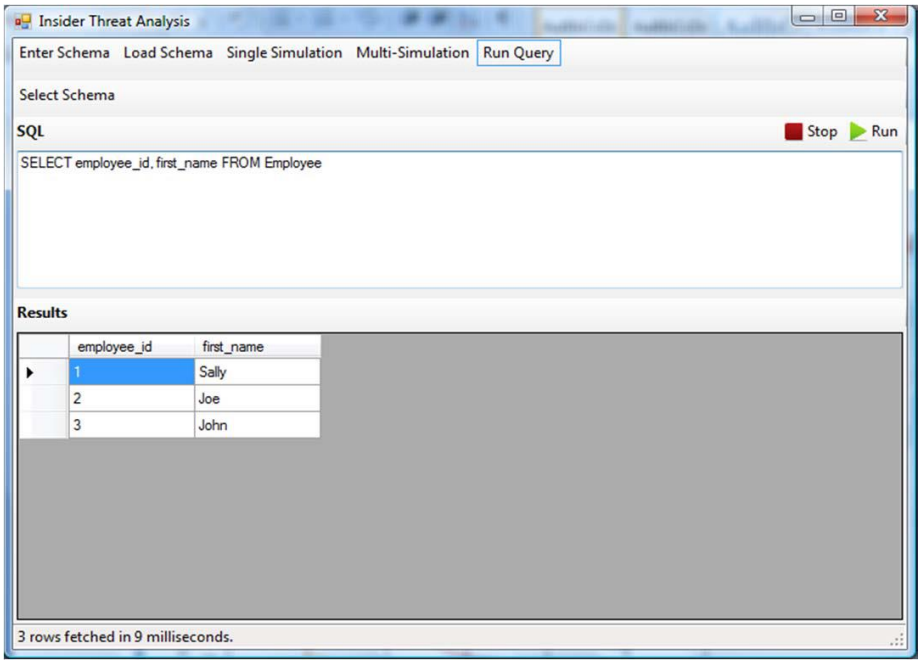

Рисунок 3.10 – Імітація запиту (приклад 1)

Наведений вище приклад показує базовий оператор SELECT. Результати, показані виконаним SQL-запитом, свідчать про те, що немає конфіденційної інформації, яку можна було б отримати для запитаної транзакції.

Наведений нижче приклад показує ще один оператор SELECT. Зверніть увагу, що база знань тепер включає всі атрибути, вибрані до цього часу, включаючи ті, що були вибрані в попередньому запиті.

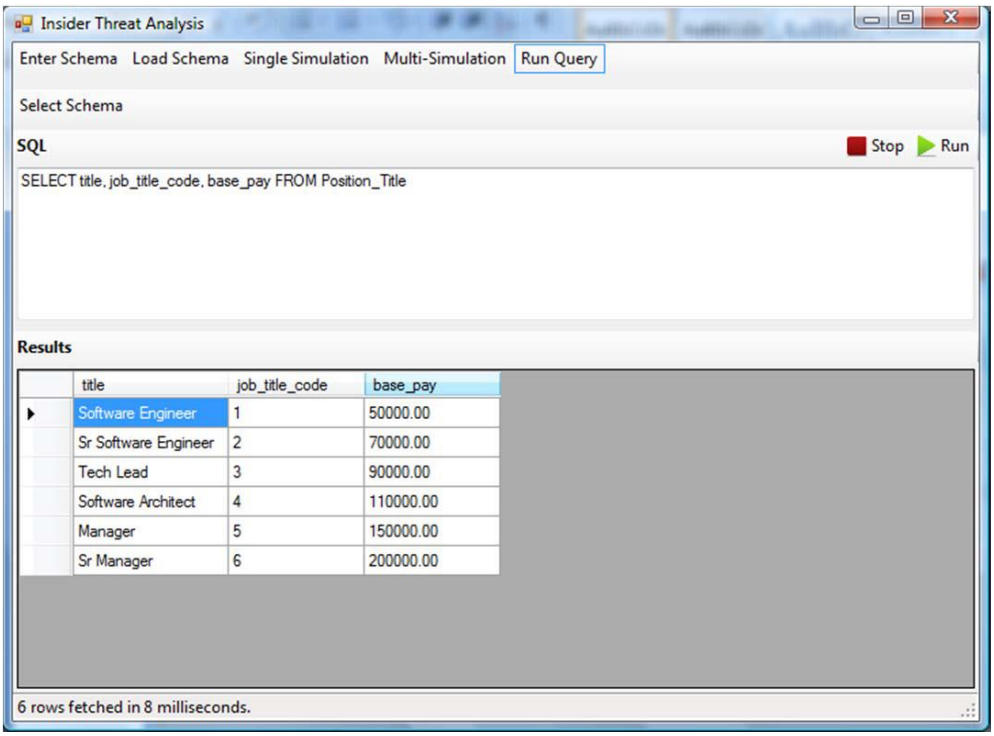

Рисунок 3.11 – Імітація запиту (приклад 2)

На основі бази знань, побудованої до цього моменту, наведений нижче запит призводить до відхилення запиту, оскільки можна отримати конфіденційну інформацію. Запит транзакції відхиляється, оскільки на основі отриманої інформації користувач може встановити зв'язок між конкретним співробітником і його базовою заробітною платою, яка є строго конфіденційною.

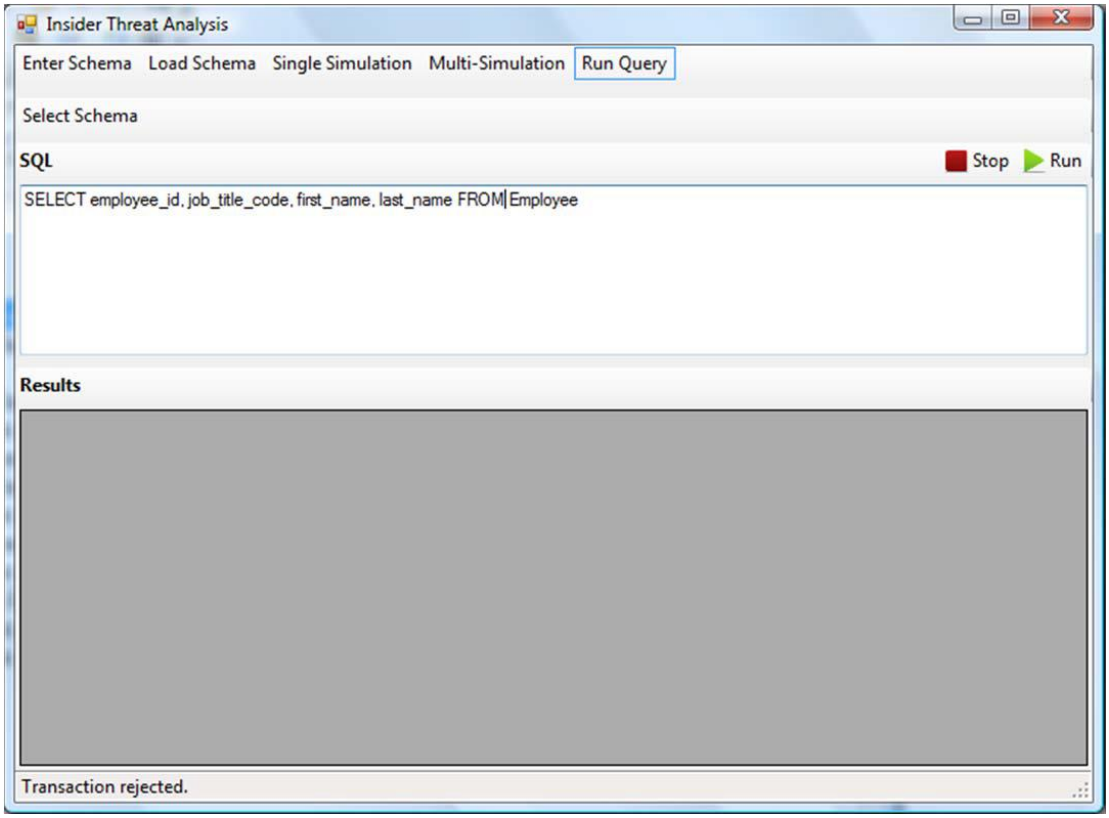

Рисунок 3.12 – Імітація запиту (приклад 3)

Симуляцію можна використовувати різними способами для отримання корисної інформації, яка може допомогти посилити безпеку та уникнути зловмисних внутрішніх атак у реляційних системах управління базами даних. Маючи схему, розбиту на різні умови загроз, симуляцію можна використовувати для визначення того, як надавати права доступу, які розкривають мінімум конфіденційної інформації. Крім того, знаючи про загрози, які найімовірніше будуть порушені, адміністратор системи може отримати інформацію для визначення найкращого балансу між наданням доступу до якомога більшої кількості даних і обмеженням можливості користувачів виводити неавторизовану інформацію.

На даний момент симуляція є проактивною для запобігання внутрішніх загроз до порушення. Іншим розвитком було б мати можливість сканувати журнали конкретної бази даних для створення бази знань, побудованої користувачами через їхні доступи. Це можна порівняти з умовами загроз для схеми, щоб визначити, які користувачі могли отримати доступ до яких

конфіденційних даних. Якщо певні користувачі не мали авторизації на таку інформацію, можна встановити обмеження та дозволи, щоб запобігти їхньому доступу до цієї інформації.

Отже, інсайдер реляційної системи управління базами даних визначається як "хтось, хто має авторизований доступ, привілеї або знання реляційної системи управління базами даних, якою він користується, і знайомий з залежностями між об'єктами даних, а також з відповідними відображеннями, і мотивований порушити політику безпеки системи через авторизований доступ". Метою роботи було надати метод, який дозволив би інсайдерам виконувати свої завдання якомога ефективніше без потенційних загроз. У випадку, коли внутрішня загроза повинна бути вирішена шляхом значного обмеження дозволів, доступність інформації буде обмежена, і користувачі не зможуть працювати так ефективно та ефективно, як це можливо. Стратегії та методи, рекомендовані в роботі, підкреслюють важливість прогнозування та запобігання, які дозволяють користувачам отримувати доступ до якомога більшої кількості інформації, доки це не втручається в будь-яку конфіденційну інформацію, до якої вони не мають доступу.

Схеми використовуються з припущенням, що атрибути змінюються нечасто і, отже, будуть надавати надійне джерело для генерування залежностей і обмежень. Залежності та обмеження на залежності, які існують у даній схемі, потім можуть бути використані для побудови графа знань користувача, який може бути використаний для прогнозування та запобігання загрозі, яку представляють інсайдери. Граф прогнозування загроз і база знань інсайдера, яка побудована під час запитів транзакцій користувача, можуть бути використані для визначення порогового значення щодо максимальної кількості інформації про конкретний атрибут, яку може отримати один інсайдер. Після досягнення або перевищення порогового значення користувачу може бути заблоковано доступ проактивно, або

адміністратору може бути надіслано попередження або скасовано доступ до певних атрибутів.

На даний момент, з роботи [2] можливо створити базу знань користувача. Тобто, маючи список того, до чого користувач має доступ, база знань побудована за допомогою графа обмежень і залежностей для відстеження того, скільки інформації про інші атрибути можна вивести. Це, однак, не надає розрахунку для умов загроз, які існують. Як виявилося при проведенні симуляції, знання про критичні елементи можна отримати лише через агрегацію знань з різних наборів. CDG не можна отримати зі схеми, оскільки концептуальні обмеження не можуть бути визначені без знання бізнес-правил і організації. База знань може бути побудована користувачем на основі висновків, які вимагають знання відносин, які виходять за межі відносин зовнішніх ключів тощо.

Пропонована симуляція надає практичне застосування методології запобігання несанкціонованого доступу і пропонує проактивне рішення для прогнозування та запобігання проблемі внутрішніх загроз у реляційних системах управління базами даних. Генерація графа знань і бази знань користувачів дозволяє системі відстежувати кількість отриманої інформації. Попередження можуть бути надіслані, якщо інсайдери мають можливість вивести значення елементів даних, до яких вони не мають авторизованого доступу. Симуляція демонструє модель того, як рішення для прогнозування та запобігання загрозам можуть бути реалізовані для будь-якої схеми бази даних і, отже, показують потенціал для застосування в галузі.

3.5. Висновки до розділу

В даному розділі особливу увагу приділено реалізації інтерфейсу системи, який забезпечує користувачу можливість вводити або завантажувати схеми бази даних, а також запускати симуляцію доступу – як

одиничну, так і багаторазову (мультисимуляцію). Такий підхід дає змогу оцінювати стійкість системи до загроз у різних сценаріях використання.

Результати моделювання було представлено в зручному форматі, що включає аналіз залежностей між кількістю користувачів, транзакцій, доступних елементів і кількістю відхилених транзакцій. Це дозволяє робити обґрунтовані висновки щодо ефективності реалізованих механізмів безпеки та адаптувати їх до змінних умов.

На завершення було реалізовано імітацію запитів до бази даних, що дає змогу оцінити реакцію системи на потенційно шкідливі дії та підтверджує дієвість запропонованої моделі захисту.

Таким чином, розділ демонструє повну програмну реалізацію концептуальної моделі захисту, що робить її придатною до практичного використання в умовах реальних інформаційних систем.

ВИСНОВКИ

В дипломній роботі було проведено комплексне дослідження проблематики запобігання несанкціонованому доступу до баз даних, зокрема з боку внутрішніх загроз, що є однією з найактуальніших проблем сучасної інформаційної безпеки.

У першому розділі було здійснено ґрунтовний аналіз предметної області, в якому розкрито особливості інсайдерських загроз, їх класифікацію у вигляді таксономії, мотивацію та профілювання потенційних зловмисників. Було також проаналізовано поточні виклики у виявленні таких загроз, що дало змогу виявити обмеження існуючих підходів до контролю доступу.

Другий розділ був присвячений розробці моделі бази даних, яка враховує залежності між об’єктами доступу та дозволяє виявити вразливі елементи в структурі інформаційної системи. Було створено матрицю залежностей, граф обмежень та універсальну схему, що відображає конфіденційність інформації. Це забезпечило основу для побудови ефективних механізмів захисту.

У третьому розділі було реалізовано програмну частину запропонованої моделі. Зокрема, розроблено інтерфейс користувача, механізми завантаження та обробки схем баз даних, а також імітаційні модулі для моделювання несанкціонованого доступу. Проведене моделювання дозволило візуалізувати результати у вигляді залежностей між кількістю користувачів, транзакцій та кількістю відхилених дій, що підтверджує практичну ефективність запропонованого підходу.

У результаті виконання роботи було досягнуто поставленої мети — розроблено концептуальну та програмну модель виявлення і запобігання несанкціонованому доступу, яка може бути використана як основа для побудови захищених інформаційних систем.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Yaseen, Q., Panda, B., “Knowledge Acquisition and Insider Threat Prediction in Relational Database Systems,” In: Proceedings of the International Workshop on Software Security Processes,. Vancouver, Canada, August, 2009, pp. 450-455
2. Yaseen, Q., Panda, B., “Predicting and Preventing Insider Threat in Relational Database Systems,” In the process of publication. Springer Link. http://www.springerlink.com/content/e43j7q17534hmk47/
3. Brackney, R., and Anderson, R., “Understanding the Insider Threat,” In: Proceedings of the March 2004 Workshop. Technical Report, RAND Corporation. Santa Monica, CA, March, 2004.
4. McCue, Andy. “Beware the Insider Security Threat,”. CIO Jury, April 17, 2008, Silicon.com. http://www.silicon.com/management/cio-insights/2008/04/17/beware-the-insider-security-threat-39188671/#comments
5. Gordon, L., Loeb, M., Lucyshyn, W., Richardson, R., “Computer Crime and Security Survey.” Available at http://www.cpppe.umd.edu/Bookstore/Documents/2005CSISurvey.pdf
6. Bertino, E., Sandhu, R. (2005). Database security—concepts, approaches, and challenges. IEEE Transactions on Dependable and Secure Computing, 2(1), 2–19.
7. Salem, M. B., Hershkop, S., & Stolfo, S. J. (2008). A survey of insider attack detection research. In Insider Attack and Cyber Security (pp. 69–90). Springer.
8. Bishop, M., & Gates, C. (2008). Defining the insider threat. In Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research (pp. 1–3). ACM.

9. Greitzer, F. L., & Frincke, D. A. (2010). Combining traditional cyber security audit data with psychosocial data: Towards predictive modeling for insider threat mitigation. Insider Threats in Cyber Security, 85–113.
10. Liu, A., Coman, I. D., Wijesekera, D., & Singhal, A. (2009). Detecting malicious insiders in database systems. In Proceedings of the 25th Annual Computer Security Applications Conference (pp. 277–286).
11. Bertino, E., Kamra, A., & Vakali, A. (2005). Intrusion detection in relational databases. In Data and Applications Security XIX (pp. 18–30). Springer.
12. Brackney, R., & Anderson, R. (2004). The insider threat: An introduction to the cyber-security challenge. Sandia National Laboratories.
13. Magklaras, G., & Furnell, S. (2002). Insider threat prediction tool: Evaluating the probability of IT misuse. Computers & Security, 21(1), 62–73.
14. Tripathi, N.; Hubballi, N. Exploiting DHCP server-side IP address conflict detection: A DHCP starvation attack. In Proceedings of the 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), Kolkata, India, 15-18 December 2015; pp. 1-3.
15. Chagarlamudi, A., Saxena, N., & Zhu, S. (2011). Privacy-preserving insider threat detection in cloud computing. In Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (pp. 93–100).
16. Fonseca, J., Vieira, M., & Madeira, H. (2007). Online detection of malicious database transactions. In Proceedings of the 2007 IEEE/IFIP International Conference on Dependable Systems and Networks (pp. 507–516).
17. Khattak, S., Latif, S., & Farooq, M. (2011). A taxonomy of insider threat mitigation techniques. In Proceedings of the 3rd International Workshop on Managing Insider Security Threats (pp. 47–56).

18. Tavallaee, M., Stakhanova, N., & Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 40(5), 516–524.
19. Zhang, Y., Wang, Y., Sun, Y., & Zhang, J. (2017). A behavior-based insider threat detection framework for relational databases. Computers & Security, 70, 199–211.
20. Hunker, J., & Probst, C. W. (2011). Insiders and insider threats—An overview of definitions and mitigation techniques. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2(1), 4–27.
21. Liu, Y., Yu, H., Wen, Q., & Li, W. (2015). An improved insider threat detection model based on user behavior. Security and Communication Networks, 8(18), 4279–4288.
22. Sabaliauskaite, G., Mathur, A. P., & Zhu, Q. (2013). A survey of cyber-physical systems security. In Proceedings of the 5th ACM Workshop on Cyber-Physical Systems Security and Privacy (pp. 45–58).
23. Probst, C. W., & Hansen, R. R. (2009). An extensible analysable system model. In Insider Threats in Cyber Security (pp. 191–206). Springer.
24. Colwill, C. (2009). Human factors in information security: The insider threat–Who can you trust these days?. Information Security Technical Report, 14(4), 186–196.
25. Althebyan, Q., & Panda, B. (2007). A novel model for intrusion detection system in relational databases. In 2007 ACM Symposium on Applied Computing (pp. 550–554).
26. Molok, N. N. A., Ahmad, A., & Chang, S. (2010). Information leakage through social networking sites: The insider threat perspective. In 2010 International Conference on Information Management and Engineering (pp. 643–647).

27. Beebe, N. L., & Rao, V. S. (2005). Using signal detection theory and ROC analysis to evaluate the performance of computer security systems. In Information Assurance and Security Workshop (pp. 17–24).
28. Hossain, M., & Mahmud, M. R. (2013). Machine learning based anomaly detection in database transactions. In Proceedings of the 8th International Conference on Internet Technology and Secured Transactions (pp. 312–318).
29. Liu, A., & Singhal, A. (2007). A framework for behavior-based anomaly detection in DBMS. In Proceedings of the 2007 IEEE Workshop on Information Assurance (pp. 178–185).
30. Althebyan, Q., & Panda, B. (2009). A dynamic multi-level anomaly detection model for database security. Computers & Security, 28(8), 675–685.
31. Khorrami, L.S.; Afshar, A. Attack detection in active queue management within large-scale networks control system with information of network and physical system units. In Proceedings of the 2016 24th Iranian Conference on Electrical Engineering (ICEE), Okinawa, Japan, 3-7 July 2016; pp. 714-719
32. Muchene, D.N.; Luli, K.; Shue, C.A. Reporting insider threats via covert channels. In Proceedings of the 2013 IEEE Security and Privacy Workshops Reporting, IEEE, San Francisco, CA, USA, 23-24 May 2013; pp. 68-71.
33. Wang, Y., & Wang, L. (2011). Using process mining for database transaction anomaly detection. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics (pp. 142–147).
34. Kandias, M., Mylonas, A., Virvilis, N., Theoharidou, M., & Gritzalis, D. (2010). An insider threat prediction model. In Proceedings of the 7th International Conference on Trust, Privacy and Security in Digital Business (pp. 26–37).

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “Реалізація моделей захисту від несанкціонованого доступу ”

Обсяг пояснювальної записки: 83 аркуші.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____