ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ
ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА ТЕХНІЧНИХ СИСТЕМ КІБЕРЗАХИСТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Система безпеки СУБД Oracle на прикладі бази даних»

на здобуття освітнього ступеня магістра
зі спеціальності __125 _Кібербезпека та захист інформації»
(код, найменування спеціальності)
освітньо-професійної програми Технічні системи інформаційного та кібернетичного захисту

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Олександр КУЖЕНТСЬКИЙ

Виконав: здобувач вищої освіти групи СЗДМ 61
Олександр КУЖЕНТСЬКИЙ

Керівник: _____ДРОБИК Олександр_____
к.т.н., професор (ПРІЗВИЩЕ, Ім'я)

Рецензент: _____
(ПРІЗВИЩЕ, Ім'я)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕОБЕЗПЕКИ
ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра ___Технічних систем кіберзахисту_____
Ступінь вищої освіти ____магістр_____
Спеціальність ___Кібербезпека та захист інформації_____
Освітньо-професійна програма _Технічні системи інформаційного та кібернетичного захисту

ЗАТВЕРДЖУЮ
Завідувач кафедри ТСКБ
_____Олександр ТУРОВСЬКИЙ

«____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кужентському Олександру Геннадійовичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система безпеки СУБД Oracle на прикладі бази даних»

керівник кваліфікаційної роботи Олександр ДРОБИК к.т.н., професор

затверджені наказом Державного університету інформаційно-комунікаційних технологій й від « 15 » жовтня 2024 р. № 320

2. Строк подання кваліфікаційної роботи 14.12.2024 р.

3. Вихідні дані до кваліфікаційної роботи: СУБД Oracle, база даних, нормативно-правова та технічна документація щодо захисту інформації, методи контролю доступу, автентифікації, шифрування, аудиту та резервного копіювання даних.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
4.1 Дослідження теоретичних засад, сучасних систем безпеки та методів захисту баз даних.
4.2 Аналіз проблем, вразливостей Oracle та методів удосконалення системи безпеки.
4.3 Розробка та впровадження удосконалень системи безпеки СУБД Oracle на прикладі бази даних.
5. Перелік графічного матеріалу: ___Презентаційний матеріал на слайдах_____

6. Дата видачі завдання _____20.10.2024 ___ _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	31.10.2024	
2	Написання першого розділу роботи	10.11.2024	
3	Написання другого розділу роботи	20.11.2024	
4	Написання третього розділу роботи	30.11.2024	
5	Написання висновків по роботі	05.12.2024	
6	Підготовка демонстраційних матеріалів	10.12.2024	
7	Підготовка доповіді	12.12.2024	

Здобувач вищої освіти _____ Олександр КУЖЕНТСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник роботи _____ Олександр ДРОБИК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи складається зі вступу, трьох розділів та загальних висновків. Загальний обсяг оформленої роботи становить 86 сторінок.

Метою кваліфікаційної роботи є розробка та впровадження методів удосконалення системи безпеки СУБД Oracle на основі аналізу існуючих загроз і вразливостей, а також їх реалізація на прикладі конкретної бази даних.

У кваліфікаційній роботі проведено огляд сучасних підходів до забезпечення безпеки баз даних, розглянуто особливості функціонування Oracle Database, визначено основні вразливості та проблеми безпеки. Запропоновано методи удосконалення захисту даних, зокрема контроль доступу, багатофакторну автентифікацію, шифрування, аудит, моніторинг, резервне копіювання та захист від SQL-ін'єкцій. Практична частина роботи спрямована на реалізацію запропонованих рішень на прикладі конкретної бази даних та аналіз ефективності впроваджених заходів.

Об'єкт дослідження. Система безпеки баз даних у СУБД Oracle.

Предмет дослідження. Методи та технології вдосконалення безпеки баз даних для забезпечення конфіденційності, цілісності та доступності інформації.

Апробація:

О.Г. Кужентський, Методи шифрування даних у СУБД ORACLE як засіб посилення інформаційної безпеки. *Всеукраїнської науково- практичної конференція: Актуальні проблеми безпеки інформаційно-телекомунікаційних систем*. м. Київ, 03 листопада 2024 року, Київ: РВЦ ДУІКТ. – 024. С.65-66. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Ключові слова: СУБД ORACLE, БАЗА ДАНИХ, ІНФОРМАЦІЙНА БЕЗПЕКА, АВТЕНТИФІКАЦІЯ, АВТОРИЗАЦІЯ, ШИФРУВАННЯ, АУДИТ, МОНІТОРИНГ, SQL-ІН'ЄКЦІЇ, РЕЗЕРВНЕ КОПІЮВАННЯ.

ABSTRACT

The text part of the qualification work consists of an introduction, three sections and general conclusions. The total volume of the completed work is 86 pages.

The purpose of the qualification work is to develop and implement methods for improving the security system of the Oracle DBMS based on an analysis of existing threats and vulnerabilities, as well as their implementation using the example of a specific database.

The qualification work reviews modern approaches to ensuring database security, examines the features of the functioning of the Oracle Database, and identifies the main vulnerabilities and security problems. Methods for improving data protection are proposed, including access control, multi-factor authentication, encryption, auditing, monitoring, backup and protection against SQL injections. The practical part of the work is aimed at implementing the proposed solutions using the example of a specific database and analyzing the effectiveness of the implemented measures.

Object of research. Database security system in the Oracle DBMS.

Subject of research. Methods and technologies for improving database security to ensure confidentiality, integrity and availability of information.

Approbation:

O.G. Kuzhentsky, Data encryption methods in the ORACLE database as a means of enhancing information security. All-Ukrainian scientific and practical conference: Current problems of information and telecommunication systems security. Kyiv, November 3, 2024, Kyiv: DUIKT Research and Development Center. – 024. P.65-66. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Keywords: ORACLE DBMS, DATABASE, INFORMATION SECURITY, AUTHENTICATION, AUTHORIZATION, ENCRYPTION, AUDIT, MONITORING, SQL INJECTION, BACKUP.

ВСТУП

Актуальність теми. У сучасному світі інформація є одним із ключових ресурсів, який визначає успіх діяльності будь-якої організації. З розвитком технологій і поширенням цифрових даних бази даних стали основою інформаційних систем у різних сферах: від бізнесу до державного управління. Однак збільшення обсягів даних і розвиток кібератак створюють значні виклики для їх захисту.

Системи управління базами даних (СУБД), зокрема Oracle Database, широко використовуються для забезпечення надійного зберігання й обробки даних. Водночас ці системи стають потенційними об’єктами для атак, таких як SQL-ін’єкції, несанкціонований доступ і витік конфіденційної інформації. Забезпечення надійної безпеки СУБД Oracle є важливою складовою сучасних підходів до кібербезпеки, адже ефективність і надійність інформаційних систем безпосередньо залежать від рівня захисту баз даних.

Таким чином, актуальність дослідження зумовлена необхідністю розробки ефективних рішень для підвищення безпеки баз даних у контексті сучасних викликів кібербезпеки.

Мета роботи. Розробка та впровадження методів удосконалення системи безпеки СУБД Oracle на основі аналізу існуючих загроз і вразливостей, а також їх реалізація на прикладі конкретної бази даних.

Завдання роботи: Для досягнення поставленої мети необхідно вирішити такі завдання:

- Провести огляд сучасних підходів до забезпечення безпеки баз даних, зокрема СУБД Oracle.
- Виявити основні вразливості й проблеми існуючої системи безпеки Oracle.
- Запропонувати методи удосконалення безпеки даних.
- Реалізувати запропоновані рішення на прикладі конкретної бази даних.
- Провести аналіз ефективності впроваджених заходів безпеки.

Об’єкт і предмет дослідження. Об’єктом дослідження є система безпеки баз

даних у СУБД Oracle. Предметом дослідження є методи та технології вдосконалення безпеки баз даних для забезпечення конфіденційності, цілісності та доступності інформації.

Методи дослідження. Для виконання поставлених завдань у роботі застосовуються наступні методи:

- аналіз і синтез інформації щодо сучасних технологій захисту баз даних;
- моделювання загроз і вразливостей у СУБД Oracle;
- практична реалізація запропонованих рішень;
- порівняльний аналіз ефективності методів безпеки.

Практична значущість роботи полягає у розробці рекомендацій для підвищення рівня захисту баз даних, які можуть бути застосовані в реальних умовах підприємств і організацій для забезпечення кібербезпеки.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface (прикладний програмний інтерфейс)

ACID – Atomicity, Consistency, Isolation, Durability

AES – Advanced Encryption Standard

БД – База даних

DBA – Database Administrator (адміністратор бази даних)

DAC – Discretionary Access Control

DAM – Database Activity Monitoring

MFA – Multi-Factor Authentication (багатофакторна автентифікація)

MAC – Mandatory Access Control

OTP – One-Time Password

Oracle DB – Oracle Database

PL/SQL – Procedural Language / Structured Query Language

RAC – Real Application Clusters

RBAC – Role-Based Access Control

RMAN – Recovery Manager

SQL – Structured Query Language

SSL/TLS – Secure Sockets Layer / Transport Layer Security

СУБД – Система управління базами даних

TDE – Transparent Data Encryption

VPD – Virtual Private Database

ЗМІСТ

Розділ 1
ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ ЗАСАД, СУЧАСНИХ СИСТЕМ БЕЗПЕКИ ТА МЕТОДІВ ЗАХИСТУ БАЗ ДАНИХ

1.1. Основи роботи з СУБД Oracle

Oracle Database є однією з найпоширеніших систем управління базами даних, яка використовується у різних галузях для зберігання, обробки та аналізу великих обсягів інформації. Її функціональність, надійність і масштабованість роблять її одним із лідерів серед СУБД. Основи роботи з Oracle Database охоплюють її архітектуру, ключові функції, процеси управління даними та забезпечення їхньої безпеки.

Основною перевагою Oracle є її архітектура, яка побудована на принципі клієнт-серверної моделі. Сервер бази даних відповідає за виконання запитів, обробку транзакцій і управління ресурсами, тоді як клієнтська сторона забезпечує зручний інтерфейс для користувачів і розробників. Архітектура Oracle включає кілька ключових компонентів, таких як інстанс бази даних і самі дані, що зберігаються в структурованих таблицях.

Інстанс бази даних – це набір серверних процесів і пам’яті, які працюють разом для обробки запитів і взаємодії з фізичною базою даних. Він забезпечує управління транзакціями, підтримку багатокористувацького середовища та оптимізацію використання ресурсів. Файли бази даних містять дані, структури таблиць, індекси, а також метадані, які необхідні для організації й ефективного пошуку інформації.[10]

Oracle підтримує роботу з мовами SQL і PL/SQL, що дозволяє користувачам ефективно взаємодіяти з базою даних. SQL використовується для виконання запитів до даних, створення таблиць і зміни структури бази даних. PL/SQL розширює функціонал SQL, додаючи можливість створення процедур, функцій і тригерів, що дозволяє автоматизувати завдання та знижувати навантаження на клієнтські програми. Завдяки цим можливостям розробники можуть створювати

складні додатки, що забезпечують високу продуктивність і безпеку.[3,8,7]

Серед ключових функцій Oracle слід виділити забезпечення транзакційної цілісності та обробки запитів. Oracle підтримує принципи ACID (атомарність, узгодженість, ізольованість, довговічність), які гарантують, що всі транзакції виконуються коректно навіть у разі системних збоїв. Крім того, Oracle пропонує механізми автоматичної оптимізації виконання запитів, що дозволяє підвищувати продуктивність системи під час роботи з великими обсягами даних.

Забезпечення безпеки даних є одним із найважливіших аспектів роботи з Oracle. Система пропонує розвинений набір інструментів для управління доступом, шифрування даних і моніторингу дій користувачів. Наприклад, Oracle Transparent Data Encryption (TDE) дозволяє шифрувати дані у стані спокою, забезпечуючи захист навіть у разі фізичного доступу до файлів бази даних. Крім того, механізми автентифікації та авторизації користувачів запобігають несанкціонованому доступу до даних.[3,8,7]

Ще одним важливим елементом Oracle є підтримка масштабованості та високої доступності. Завдяки технологіям кластеризації, таким як Oracle Real Application Clusters (RAC), система може розподіляти навантаження між кількома серверами, що дозволяє забезпечувати високу продуктивність навіть під час роботи з великими базами даних. Крім того, Oracle підтримує резервне копіювання та відновлення даних, що є критично важливим для забезпечення безперервності бізнес-процесів.[1, 3]

Важливою складовою Oracle є її інтеграція з хмарними технологіями. Oracle Cloud Infrastructure дозволяє організаціям використовувати переваги хмарного зберігання та обробки даних, забезпечуючи гнучкість і економію ресурсів. Крім того, хмарні рішення Oracle надають розширені можливості для захисту даних, включаючи автоматичне створення резервних копій і управління безпекою.

Oracle Database є також універсальною платформою для аналітики даних. Інструменти аналітики, вбудовані в систему, дозволяють виконувати складні обчислення та аналіз великих обсягів даних у реальному часі. Це робить Oracle важливим інструментом для організацій, які потребують швидкого доступу до

інформації для прийняття бізнес-рішень.

Загалом Oracle Database – це потужна і гнучка система управління базами даних, яка поєднує в собі широкий набір функцій, високу продуктивність та розширені можливості забезпечення безпеки. Її основи роботи спрямовані на задоволення потреб сучасних організацій, що працюють із великими обсягами інформації та стикаються зі зростаючими викликами кібербезпеки.[1,3,8,7]

1.2. Огляд сучасних систем безпеки баз даних

Системи безпеки баз даних є ключовою складовою інформаційної безпеки, оскільки саме бази даних зберігають найбільш важливі дані організацій: фінансову інформацію, персональні дані, дані клієнтів і стратегічні плани. Сучасні загрози, зокрема кіберзлочини, несанкціонований доступ і внутрішні зловживання, роблять забезпечення безпеки баз даних критично важливим завданням.

Для захисту баз даних сучасні системи безпеки розвиваються в кількох напрямах. Основними аспектами таких систем є захист даних від несанкціонованого доступу, забезпечення конфіденційності інформації, підтримання цілісності даних, а також їх доступність для авторизованих користувачів. Інструменти, технології та методи захисту баз даних стають дедалі складнішими та розумнішими, але водночас стикаються з численними викликами, які вимагають системного підходу до вирішення.[2]

1.3. Основні підходи до забезпечення безпеки баз даних

Сучасні системи безпеки баз даних базуються на кількох основних підходах:

1.1.1. Контроль доступу до даних.

Контроль доступу є основним елементом будь-якої системи безпеки баз даних. Він забезпечує авторизацію користувачів, розподіл ролей і прав доступу, а також дозволяє обмежити доступ до конфіденційної інформації. У сучасних системах широко використовується рольовий контроль доступу (Role-Based Access Control,

RBAC), який дозволяє гнучко управляти доступом на основі виконуваних функцій користувача. Деякі системи впроваджують обов'язковий контроль доступу (Mandatory Access Control, MAC), що забезпечує ще суворіші правила доступу, наприклад, для урядових або фінансових установ.[4, 5]

1.1.2. Шифрування даних.

Для забезпечення конфіденційності даних у сучасних системах використовуються технології шифрування, такі як Transparent Data Encryption (TDE), що дозволяє шифрувати дані у стані спокою. Це особливо важливо для захисту від фізичних атак або викрадення носіїв інформації. Також широко застосовується шифрування під час передачі даних (SSL/TLS), яке забезпечує безпечну комунікацію між клієнтом і сервером.[1,13]

1.1.3. Аудит і моніторинг.

Системи безпеки баз даних мають включати засоби для аудиту й моніторингу дій користувачів. Ведення журналів дій дозволяє відслідковувати активність користувачів, виявляти потенційні загрози та аналізувати підозрілі події. Інструменти, такі як Database Activity Monitoring (DAM), забезпечують автоматичне виявлення аномалій у поведінці користувачів.[13]

1.1.4. Захист від SQL-ін’єкцій.

SQL-ін’єкції є однією з найпоширеніших загроз для баз даних. Сучасні системи впроваджують методи захисту, такі як підготовлені запити (prepared statements), параметризовані запити та веб-додатки з вбудованими фільтрами (Web Application Firewall, WAF).[13, 22]

1.1.5. Резервне копіювання та відновлення.

Для забезпечення доступності даних у разі збою чи атак резервне копіювання є важливим аспектом безпеки. Сучасні СУБД, такі як Oracle Database, надають розвинені засоби автоматичного створення резервних копій і тестування сценаріїв відновлення.[13,6]

1.1.6. Переваги сучасних систем безпеки баз даних

Сучасні системи безпеки баз даних представляють собою складні багаторівневі рішення, спрямовані на забезпечення високого рівня захисту інформації. Їх

основною перевагою є інтегровані інструменти, які спрощують впровадження й адміністрування заходів безпеки. Наприклад, багато сучасних СУБД, таких як Oracle, надають вбудовані функції для контролю доступу, шифрування та моніторингу активності користувачів. Це дозволяє організаціям швидко налаштувати базові аспекти безпеки та адаптувати їх під свої потреби.[5]

Автоматизація процесів є ще однією вагомою перевагою. Системи безпеки дедалі частіше використовують технології для автоматичного виявлення загроз, що значно знижує навантаження на адміністраторів. Наприклад, інструменти для моніторингу баз даних можуть самостійно аналізувати дії користувачів у реальному часі та генерувати сповіщення про аномальну активність. Завдяки цьому організації можуть швидше реагувати на потенційні інциденти безпеки.

Сучасні системи також відзначаються гнучкістю. Вони дозволяють налаштовувати політики безпеки під конкретні потреби організації, що особливо важливо для компаній із різними рівнями доступу до даних. Наприклад, завдяки ролям і політикам доступу можна обмежити права користувачів до певних таблиць або навіть окремих рядків у базі даних, що мінімізує ризики внутрішніх загроз.

Масштабованість — ще одна значуща перевага сучасних систем безпеки. Технології, такі як Oracle Real Application Clusters (RAC), дозволяють працювати в середовищах із високим навантаженням, забезпечуючи безперервну роботу бази даних навіть у разі значного зростання обсягів інформації або збільшення кількості користувачів. Це є критично важливим для великих організацій, які працюють із великими обсягами даних і потребують високої доступності інформаційних систем.[27-29]

Втім, попри всі переваги, сучасні системи безпеки баз даних мають низку недоліків, які слід враховувати. Однією з головних проблем є складність налаштування. Такі системи зазвичай мають розширені можливості, які вимагають високого рівня кваліфікації від адміністраторів. Неправильне налаштування може знизити ефективність захисту або навіть створити нові вразливості.

Вартість впровадження та підтримки також є значущою перепоною для багатьох організацій, особливо для малого та середнього бізнесу. Ліцензії на використання

сучасних систем, таких як Oracle Database Enterprise Edition, часто мають високу ціну, а додаткові функції безпеки, як-от шифрування або моніторинг, можуть вимагати окремих вкладень.

Ще одним важливим аспектом є постійна потреба в оновленнях. У сучасному світі загрози швидко еволюціонують, і системи безпеки повинні бути актуальними для протидії новим типам атак. Це створює додаткове навантаження на IT-відділи, які повинні слідкувати за виходом оновлень та встановлювати їх без затримок.

Окрім цього, велика залежність від людського фактора залишається актуальною проблемою. Навіть найсучасніші інструменти можуть бути неефективними, якщо адміністратори роблять помилки під час налаштування, а користувачі нехтують політиками безпеки. Це особливо важливо для організацій із недостатньо розвиненою культурою кібербезпеки.[15]

Розвиток технологій безпеки баз даних продовжує рухатися вперед, відкриваючи нові можливості для покращення захисту. Одним із ключових напрямів є використання штучного інтелекту та машинного навчання для виявлення аномалій і реагування на загрози в реальному часі. Ці технології дозволяють автоматизувати виявлення підозрілих дій, аналізувати великі обсяги даних і передбачати потенційні ризики.

Ще однією перспективною тенденцією є удосконалення технологій шифрування, таких як гомоморфне шифрування, яке дозволяє виконувати обчислення над зашифрованими даними без їхнього розшифрування. Це відкриває нові можливості для роботи з чутливими даними, забезпечуючи їхню конфіденційність навіть у хмарних середовищах.

Інтеграція з хмарними рішеннями також є важливим аспектом розвитку. Хмарні сервіси пропонують розширені можливості для резервного копіювання, моніторингу та забезпечення безпеки даних. Завдяки хмарним технологіям організації можуть знизити витрати на інфраструктуру й отримати доступ до передових інструментів безпеки без необхідності їхнього локального впровадження.

Сучасні системи безпеки баз даних мають великий потенціал для забезпечення

надійного захисту інформації. Проте їхнє впровадження та підтримка залишаються складними завданнями, які потребують значних ресурсів. Подальший розвиток технологій, таких як штучний інтелект, хмарні обчислення та інноваційні підходи до шифрування, дає змогу організаціям ще ефективніше захищати свої дані в умовах постійно зростаючих кіберзагроз.[17]

1.1.7. Тенденції розвитку

Розвиток систем безпеки баз даних продовжує еволюціонувати разом із технологічним прогресом, відкриваючи нові горизонти для захисту інформації. У майбутньому однією з ключових тенденцій стане впровадження штучного інтелекту та машинного навчання. Ці технології вже починають активно використовуватися для аналізу великих обсягів даних, виявлення аномалій у поведінці користувачів і прогнозування потенційних загроз. Завдяки штучному інтелекту системи безпеки зможуть не лише реагувати на відомі загрози, а й виявляти нові типи атак, які ще не мають шаблонів для розпізнавання. Це суттєво підвищить рівень автоматизації та ефективність захисту баз даних, дозволяючи знижувати навантаження на адміністративний персонал.[22]

Іншою важливою тенденцією є вдосконалення технологій шифрування. Сучасні методи, як-от гомоморфне шифрування, відкривають можливість виконання обчислень над зашифрованими даними без їхнього розшифрування. Це революційний підхід, який дозволяє зберігати конфіденційність інформації навіть під час її обробки в хмарних середовищах. Такі технології стають дедалі актуальнішими в умовах, коли обсяги даних зростають, а з ними й ризики витоку інформації. Використання гомоморфного шифрування дозволить організаціям забезпечувати максимальний рівень захисту без шкоди для продуктивності.

Хмарні рішення також продовжують впливати на розвиток систем безпеки баз даних. Інтеграція баз даних із хмарними платформами дає змогу використовувати передові інструменти для моніторингу, резервного копіювання та відновлення даних. Хмарні технології не тільки забезпечують гнучкість і масштабованість, а й дозволяють знижувати витрати на утримання локальної інфраструктури. Крім того, хмарні провайдери часто пропонують комплексні рішення для безпеки, які

включають автоматизовані оновлення, шифрування та засоби для виявлення загроз. Це робить хмарні сервіси привабливими навіть для організацій, які традиційно використовують локальні бази даних.

Загалом сучасні системи безпеки баз даних вже досягли високого рівня ефективності завдяки інтегрованим інструментам і автоматизації. Однак їхнє впровадження залишається ресурсозатратним процесом, який вимагає не лише значних фінансових інвестицій, а й високої кваліфікації персоналу. Це особливо актуально для малих і середніх організацій, які часто стикаються з труднощами впровадження таких складних систем. Майбутній розвиток технологій має на меті подолати ці виклики, спрощуючи інтеграцію нових інструментів і знижуючи бар'єри для їх використання.[21]

Подальше вдосконалення систем безпеки баз даних відкриває нові можливості для забезпечення захисту інформації. Впровадження інтелектуальних систем, використання передових методів шифрування та інтеграція з хмарними платформами дають змогу створювати більш стійкі та безпечні рішення. Такі системи здатні не лише реагувати на загрози, а й активно запобігати їм, забезпечуючи надійну основу для роботи з даними у все складнішому цифровому середовищі.

1.1.8. Аналіз існуючих підходів до захисту даних

Захист даних у системах управління базами даних (СУБД) є одним із найважливіших аспектів сучасної кібербезпеки. Зі зростанням обсягів даних і залежності від цифрових технологій бази даних стають ключовими цілями для зловмисників, які намагаються отримати доступ до конфіденційної інформації, викрасти її чи зруйнувати. Сучасні кібератаки стають дедалі складнішими та адаптивними, що вимагає від систем захисту постійної еволюції. У цьому контексті ефективність підходів до захисту залежить від їхньої здатності забезпечити дотримання трьох основних принципів інформаційної безпеки: конфіденційність, цілісність і доступність даних.

З розвитком технологій з’явилися різноманітні методи захисту, які охоплюють широкий спектр загроз. Одним із фундаментальних підходів є впровадження

багаторівневого захисту, який включає контроль доступу, автентифікацію, шифрування даних і моніторинг дій користувачів. Кожен із цих елементів виконує свою роль у забезпеченні цілісності системи. Наприклад, контроль доступу обмежує можливості користувачів, дозволяючи виконувати лише ті дії, які є необхідними для їхньої роботи. Це значно знижує ризик випадкових або навмисних змін у структурі чи вмісті бази даних.[23]

Автентифікація користувачів є критично важливим компонентом захисту, оскільки саме від цього процесу залежить, хто і яким чином може отримати доступ до системи. Використання багатофакторної автентифікації (MFA) стає все більш поширеним завдяки своїй здатності значно зменшити ймовірність компрометації облікових записів. Поєднання паролів із одноразовими кодами (OTP), біометрією чи фізичними токенами створює багаторівневу перевірку, яка ускладнює доступ для зловмисників.

Іншим важливим аспектом є шифрування даних, яке гарантує, що інформація залишається конфіденційною навіть у разі її перехоплення або втрати. Сучасні алгоритми шифрування, такі як AES-256, забезпечують високий рівень захисту як даних у стані спокою, так і під час їхньої передачі. Шифрування стає особливо актуальним у хмарних середовищах, де дані можуть знаходитися на загальнодоступній інфраструктурі.[24]

Моніторинг і аудит дій користувачів також є невіддільною частиною захисту. Ведення журналів активності дозволяє виявляти підозрілі дії, аналізувати інциденти та підвищувати прозорість операцій. За допомогою автоматизованих систем моніторингу, таких як Database Activity Monitoring (DAM), організації можуть отримувати сповіщення про потенційні загрози в режимі реального часу, що значно скорочує час реагування на інциденти.

Незважаючи на прогрес у галузі, сучасні підходи до захисту даних не є універсальними. Кожен метод має свої обмеження, і тому важливо застосовувати комплексний підхід, який об’єднує кілька рівнів захисту. Наприклад, навіть найсучасніші інструменти можуть бути неефективними, якщо в системі є вразливі місця через помилки конфігурації чи людський фактор. Тому важливо регулярно

проводити аудит безпеки, аналізувати потенційні ризики й оновлювати політики безпеки відповідно до нових викликів.

Загалом, аналіз існуючих підходів до захисту даних свідчить про те, що ефективна система безпеки повинна бути динамічною, адаптивною та орієнтованою на запобігання загрозам. Поєднання технічних заходів, таких як шифрування та моніторинг, із організаційними, наприклад, навчанням персоналу, дозволяє досягти високого рівня захисту даних і створити надійну основу для подальшого розвитку систем управління базами даних.

1.1.9. Класичні підходи до захисту даних

Сучасні системи безпеки баз даних спираються на класичні підходи до захисту інформації, які забезпечують базовий рівень безпеки й закладають основу для впровадження більш складних механізмів. До основних підходів належать моделі доступу до даних, механізми автентифікації та авторизації, шифрування, а також журналювання і аудит.[25]

1.1.9.1. Моделі доступу до даних

Одним із базових елементів захисту є моделі контролю доступу. Вони визначають, хто і як може взаємодіяти з даними в системі. Існує три основні моделі доступу:

– Дискреційна модель контролю доступу (Discretionary Access Control, DAC): У цій моделі права доступу до даних визначаються власником інформації. Користувачі можуть надавати або обмежувати доступ до своїх даних іншим користувачам. Такий підхід забезпечує гнучкість у розподілі прав, але має вразливості, пов'язані з людськими помилками чи зловживаннями, наприклад, коли права доступу ненавмисно надаються неавторизованим користувачам.

– Обов’язкова модель контролю доступу (Mandatory Access Control, MAC): Ця модель базується на суворих правилах, визначених адміністрацією системи. Вона найчастіше використовується в організаціях із підвищеними вимогами до безпеки, таких як урядові установи. Доступ до даних залежить від рівнів конфіденційності, і користувачі можуть працювати лише з тими даними, до яких вони мають належний доступ.[16, 17]

– Рольова модель контролю доступу (Role-Based Access Control, RBAC): Права

доступу в цій моделі надаються відповідно до ролі користувача в організації. Наприклад, адміністратор може мати повний доступ до системи, тоді як менеджер матиме доступ лише до певних таблиць. Це знижує ризик неналежного доступу й забезпечує більш структурований підхід до розподілу прав.[17]

1.1.9.2. Механізми автентифікації та авторизації

Автентифікація визначає, хто намагається отримати доступ до системи, тоді як авторизація контролює, що цей користувач може робити після входу. Одним із найефективніших методів є використання багатофакторної автентифікації (MFA). Цей підхід передбачає використання кількох рівнів перевірки, наприклад, пароля, одноразового коду (OTP), біометричних даних чи токенів. У разі компрометації одного з елементів автентифікації доступ до системи залишається захищеним.

Авторизація, у свою чергу, впроваджується через політики доступу, які дозволяють обмежити можливості користувача відповідно до його ролі чи рівня доступу. Це знижує ймовірність помилок і запобігає неналежному використанню системи.[10]

1.1.9.3. Шифрування даних

Шифрування забезпечує конфіденційність даних навіть у разі їхнього перехоплення або компрометації. Основні типи шифрування, що використовуються в системах безпеки баз даних:

– Шифрування даних у стані спокою (Transparent Data Encryption, TDE): Захищає дані, що зберігаються на фізичних носіях. Це важливо для забезпечення конфіденційності у разі фізичного доступу до серверів або резервних копій.

– Шифрування під час передачі (SSL/TLS): Використовується для захисту даних, що передаються між клієнтом і сервером. Цей підхід гарантує, що інформація не буде доступною навіть у разі перехоплення мережевого трафіку.

– Політики шифрування на рівні полів: Дозволяють шифрувати окремі елементи даних, наприклад, номери кредитних карток або персональні дані клієнтів. Це дає змогу посилити захист для найбільш чутливої інформації.[5,11]

1.1.9.4. Журналювання та аудит

Моніторинг і запис дій користувачів є важливою складовою захисту даних. Усі операції в базі даних повинні реєструватися для подальшого аналізу. Це дозволяє

виявляти аномальні активності, запобігати інцидентам безпеки та реагувати на них у разі виникнення.

Наприклад, Oracle Database використовує механізми Audit Vault, які централізують дані про дії користувачів і надають інструменти для їхнього аналізу. Аудит може фіксувати не лише успішні дії, але й невдалі спроби доступу, що дозволяє виявляти потенційні загрози ще до їхньої реалізації.

Класичні підходи до захисту даних, такі як моделі контролю доступу, автентифікація, шифрування та аудит, формують основу сучасних систем безпеки баз даних. Вони забезпечують базовий рівень захисту, який може бути розширений за допомогою новітніх технологій і методів. Використання цих підходів у комплексі гарантує конфіденційність, цілісність і доступність даних, що є критично важливим у сучасному цифровому середовищі.[8-13]

1.1.10. Сучасні підходи до захисту даних

Сучасні підходи до захисту даних у базах даних є відображенням технологічного прогресу, спрямованого на забезпечення стійкості до зростаючих загроз. Одним із ключових нововведень є інтеграція штучного інтелекту (ШІ) та машинного навчання. Ці технології дають змогу здійснювати моніторинг активностей у реальному часі та виявляти загрози автоматично. Завдяки поведінковому аналізу користувачів системи можуть визначати підозрілі дії, навіть якщо вони відповідають стандартним правилам доступу. Наприклад, раптова зміна обсягу запитів або спроби отримати доступ до чутливих даних можуть бути виявлені як аномалії та заблоковані до їх реалізації. Штучний інтелект поступово стає невід’ємною частиною сучасних систем безпеки, оскільки здатний адаптуватися до нових типів атак.

Іншою важливою концепцією є Zero Trust Architecture (ZTA), яка передбачає, що жоден користувач або пристрій не може вважатися надійним автоматично, навіть якщо вони знаходяться у внутрішній мережі. Усі запити на доступ до бази даних проходять перевірку незалежно від їхнього джерела чи попереднього стану. Цей підхід значно знижує ризик внутрішніх загроз, таких як зловживання правами доступу, та забезпечує суворий контроль на кожному рівні взаємодії із системою.

У сучасних умовах, коли межі корпоративних мереж розмиваються через віддалену роботу та хмарні сервіси, "нульова довіра" стає основою безпеки.[15,16]

Політики маскування даних є ще одним важливим сучасним методом захисту. Вони дозволяють приховувати чутливу інформацію від користувачів, які не мають доступу до цих даних. Наприклад, під час тестування додатків реальні персональні дані можуть бути замінені вигаданими, що зберігають ту саму структуру. Це знижує ризик витоку інформації навіть у разі зловживання доступом. Маскування даних також корисне для забезпечення відповідності нормативним вимогам, таким як GDPR, які регулюють використання та обробку персональних даних.

Контроль доступу на рівні рядків і стовпців є додатковим рівнем деталізації в сучасних системах. Цей підхід дозволяє задавати конкретні правила доступу залежно від потреб користувачів або їхніх ролей. Наприклад, менеджери можуть переглядати дані лише своїх підлеглих, тоді як адміністратор має повний доступ до всієї таблиці. Така сегментація значно знижує ризик несанкціонованого доступу та гарантує, що користувачі працюють лише з тими даними, які їм необхідні.

Сучасні підходи до захисту даних мають ряд переваг. Вони забезпечують високу гнучкість, дозволяючи адаптувати політики безпеки під конкретні потреби організацій. Завдяки автоматизації багато процесів, таких як моніторинг і управління правами доступу, стали більш ефективними та потребують менше зусиль від адміністраторів. Використання комплексних методів, що поєднують класичні рішення (наприклад, шифрування) із сучасними, дозволяє створювати багаторівневу систему безпеки, яка здатна реагувати навіть на найскладніші загрози.[18]

Проте впровадження цих підходів пов’язане з певними викликами. Реалізація сучасних систем захисту вимагає значних фінансових ресурсів і часу. Інструменти для впровадження шифрування, моніторингу чи автоматизації часто мають високу вартість, що може бути недоступним для невеликих організацій. Крім того, налаштування цих систем вимагає високого рівня технічної кваліфікації. Людський фактор також залишається суттєвим ризиком: помилки в налаштуваннях або недбале ставлення до політик безпеки можуть знижувати ефективність навіть

найкращих технологій.

Існуючі підходи до захисту даних формують міцний фундамент для сучасних рішень у сфері безпеки. Поєднання класичних і новітніх методів, таких як штучний інтелект і "нульова довіра", створює багаторівневу систему, здатну ефективно реагувати на сучасні виклики. Однак впровадження таких рішень потребує стратегічного планування, регулярного моніторингу та оновлень для забезпечення їхньої актуальності та ефективності.[19]

Сучасні підходи до захисту даних також потребують постійного вдосконалення у відповідь на нові виклики. Загрози, які раніше могли здаватися суто гіпотетичними, сьогодні стають реальністю через зростання складності та масштабів кібератак. Наприклад, атаки на ланцюжок постачання (supply chain attacks), які спрямовані на уразливості в інфраструктурі організації, ставлять під загрозу навіть найбільш захищені бази даних. Це означає, що системи безпеки повинні бути інтегрованими не лише в межах організації, але й на рівні взаємодії з партнерами та постачальниками.

Ще одним перспективним напрямом розвитку є використання хмарних технологій для захисту даних. Хмарні сервіси пропонують розширені можливості для резервного копіювання, аварійного відновлення та моніторингу. Наприклад, хмарні платформи дозволяють автоматично створювати резервні копії баз даних із використанням шифрування, забезпечуючи їхній захист навіть у разі фізичної втрати носіїв. Крім того, хмарні інструменти для аналізу безпеки часто мають вбудовані механізми, що використовують штучний інтелект для виявлення загроз у реальному часі.

Інтеграція сучасних підходів до захисту даних вимагає злагодженої роботи різних відділів організації. Наприклад, технічні спеціалісти повинні забезпечувати належну конфігурацію системи, тоді як керівництво має визначати політики безпеки та забезпечувати їхнє дотримання. Важливу роль відіграє навчання персоналу, адже навіть найкращі технології можуть бути неефективними, якщо користувачі не розуміють важливості дотримання правил безпеки.

Загалом сучасні підходи до захисту даних є не просто технічними рішеннями, а

частиною загальної стратегії управління ризиками в організації. Вони допомагають не лише запобігати інцидентам, але й будувати довіру до інформаційних систем, що є важливим фактором для бізнесу в умовах цифрової економіки. Подальший розвиток цих підходів відкриває нові можливості для створення ще більш надійних систем безпеки, які відповідатимуть викликам майбутнього.[17]

Розділ 2
МЕТОДИ УДОСКОНАЛЕННЯ СИСТЕМИ БЕЗПЕКИ

2.1. Опис проблем і вразливостей Oracle

Oracle Database, одна з найпоширеніших систем управління базами даних, широко використовується завдяки своїй функціональності та надійності. Проте, як і будь-яка складна програмна платформа, Oracle має свої вразливості, які можуть бути використані зловмисниками для реалізації атак або порушення безпеки. У цьому підрозділі розглядаються основні проблеми, які виникають у системах Oracle, та їхній вплив на безпеку.

Однією з головних проблем є механізми автентифікації та авторизації. Стандартна автентифікація базується на використанні паролів, які часто є слабким місцем, якщо не дотримано суворих вимог щодо їхньої складності. Багато організацій залишають стандартні облікові записи, такі як *SYS* і *SYSTEM*, із початковими паролями, що значно підвищує ризик компрометації. Крім того, відсутність чітко визначених привілеїв може дозволити звичайним користувачам отримувати доступ до критично важливих функцій бази даних. Це створює потенційну загрозу для ненавмисних або зловмисних дій, які можуть призвести до втрати даних чи порушення їхньої цілісності.[15-20]

Вразливість до SQL-ін’єкцій залишається ще одним значним викликом. Цей вид атак дозволяє зловмисникам використовувати незахищені SQL-запити для виконання шкідливого коду. Особливо це актуально для додатків, які взаємодіють із базою даних через динамічні SQL-запити, не перевіряючи введені параметри. SQL-ін’єкції можуть не тільки надати доступ до чутливих даних, але й спричинити їх видалення або зміну.

Неправильна конфігурація системи є ще одним джерелом вразливостей. Використання застарілих протоколів передачі даних, таких як Telnet або FTP, відсутність налаштувань шифрування та неправильне управління журналами дій користувачів створюють додаткові загрози. Неналежне налаштування параметрів

безпеки часто пов'язане з прагненням спростити адміністрування, але така спрощеність призводить до підвищення ризиків.

Ще одним критичним аспектом є оновлення програмного забезпечення. Багато організацій не встановлюють оновлення вчасно, що залишає базу даних уразливою до атак, які використовують відомі експлойти. Проблеми, пов'язані з переповненням буфера, слабкістю криптографічних функцій чи незахищеністю API, можуть бути швидко виправлені завдяки патчам, проте недостатня увага до оновлень залишається значною проблемою.[15-20]

Зберігання конфіденційних даних у відкритому вигляді також створює загрозу. Дані, такі як паролі або персональні ідентифікаційні номери, нерідко зберігаються без шифрування, що робить їх доступними для несанкціонованих користувачів у разі фізичного доступу до серверів. Відсутність належної політики шифрування часто пов’язана з браком ресурсів або обізнаності, але така проблема може мати критичні наслідки для безпеки.

Окремої уваги заслуговують загрози з боку внутрішніх користувачів. Адміністратори баз даних або співробітники з розширеними правами доступу можуть виконувати несанкціоновані дії, які залишаються непоміченими через відсутність належного моніторингу. Це створює серйозний ризик для організацій, які покладаються виключно на довіру до своїх працівників.

Відсутність централізованого моніторингу та аудиту також значно знижує рівень захисту Oracle Database. Механізми ведення журналів, хоча і є частиною платформи, нерідко ігноруються через складність налаштування. Як наслідок, організації не можуть ефективно відстежувати підозрілі дії або реагувати на інциденти.

Нарешті, складність адміністрування є ще одним фактором ризику. Oracle Database — це високофункціональна система, що вимагає значного рівня знань і досвіду. Недостатня кваліфікація адміністраторів часто призводить до помилок у налаштуванні, які відкривають шлях для потенційних атак. [15-20]

Всі ці проблеми демонструють, що забезпечення безпеки Oracle Database є складним завданням, яке потребує системного підходу. Проте правильна

конфігурація, впровадження регулярних оновлень і використання передових практик можуть значно знизити ризики та забезпечити високий рівень захисту даних. У наступних підрозділах будуть розглянуті конкретні методи удосконалення, спрямовані на усунення цих вразливостей і підвищення загальної безпеки Oracle Database. [21]

2.2. Запропоновані методи покращення

Забезпечення безпеки Oracle Database вимагає комплексного підходу, що охоплює технічні й організаційні заходи. Це дозволяє ефективно протидіяти як зовнішнім атакам, так і внутрішнім загрозам, які можуть виникати через людські помилки або зловживання. Запропоновані методи покращення спрямовані на усунення існуючих вразливостей і створення багаторівневої системи захисту.

Одним із основних напрямків є впровадження багатофакторної автентифікації (MFA), яка забезпечує додаткові рівні перевірки особи користувача. У більшості випадків Oracle Database використовує традиційну автентифікацію на основі імені користувача та пароля, що є вразливим до атак грубої сили та підбору паролів. Додавання другого або третього фактора автентифікації, таких як одноразові паролі (OTP), біометричні дані або токени безпеки, значно ускладнює процес несанкціонованого доступу. Завдяки цьому можна досягти більш високого рівня захисту облікових записів, навіть якщо основний пароль було скомпрометовано.[21,22]

Для покращення авторизації пропонується впровадити принцип найменших привілеїв (Least Privilege Principle). Цей підхід передбачає, що кожен користувач отримує доступ лише до тих даних і функцій, які необхідні для виконання його завдань. Це знижує ризик ненавмисних змін у базі даних, а також запобігає використанню широких прав доступу для зловмисних дій. Наприклад, адміністратори часто створюють облікові записи з надмірними привілеями для зручності роботи, але такий підхід значно підвищує ризики. Завдяки чіткому розподілу прав можна забезпечити кращу захищеність бази даних.

Одним із найбільш актуальних заходів є боротьба із SQL-ін’єкціями, які залишаються однією з найпоширеніших атак на бази даних. Ці атаки виникають

через відсутність належної перевірки введених користувачами даних. Зловмисники використовують незахищені динамічні SQL-запити для виконання шкідливого коду. Запобігти таким атакам можна шляхом переходу на використання підготовлених запитів (prepared statements) і параметризованих запитів. Ці методи автоматично перевіряють введені дані й виключають можливість впровадження небезпечного коду. Крім того, додаткове впровадження засобів моніторингу запитів дозволить виявляти підозрілу активність у реальному часі.[23,24]

Шифрування даних відіграє ключову роль у забезпеченні конфіденційності інформації. Використання Transparent Data Encryption (TDE) дозволяє шифрувати дані у стані спокою, що гарантує їхню безпеку навіть у разі фізичної компрометації серверів або резервних копій. Для забезпечення безпеки передачі даних між клієнтами й сервером пропонується використовувати SSL/TLS. Ці заходи дозволяють гарантувати, що конфіденційні дані залишатимуться захищеними в усіх середовищах, навіть у разі перехоплення трафіку.

Моніторинг активності користувачів є ще одним важливим елементом системи безпеки. Використання інструментів для централізованого моніторингу, таких як Oracle Audit Vault або Database Activity Monitoring (DAM), дозволяє фіксувати всі дії користувачів, виявляти аномалії та генерувати сповіщення для адміністраторів. Це особливо важливо для запобігання внутрішнім загрозам, коли співробітники організації можуть ненавмисно або навмисно завдати шкоди системі. Централізований моніторинг також допомагає виявляти зовнішні атаки, які проявляються у вигляді підозрілої поведінки.[23-25]

Автоматизація процесу оновлення є ще одним важливим напрямком. Регулярні оновлення та патчі безпеки дозволяють усувати відомі вразливості, які можуть бути використані для атак. Автоматизовані системи оновлення знижують ризик людських помилок і забезпечують своєчасне впровадження необхідних змін. Організації, які ігнорують цей аспект, залишають свої бази даних вразливими до експлуатації відомих проблем.

Маскування даних також є ефективним методом захисту. Воно дозволяє приховувати конфіденційну інформацію від користувачів, які не мають

відповідних привілеїв. Цей підхід є особливо корисним у тестових середовищах, де використання реальних даних може створювати ризики для безпеки. Завдяки маскуванню можна забезпечити безпеку даних навіть у разі витоку інформації з тестового середовища.

Додатково пропонується впровадити концепцію Zero Trust Architecture. Цей підхід передбачає повну відсутність довіри до користувачів і пристроїв, незалежно від їхнього місцезнаходження. Кожен запит на доступ до бази даних проходить перевірку на відповідність політикам безпеки. Така модель значно знижує ризик несанкціонованого доступу та забезпечує суворий контроль на кожному рівні.[22]

Для підвищення загальної обізнаності серед працівників необхідно впроваджувати регулярні тренінги для адміністраторів і розробників. Освітні заходи дозволяють уникнути поширених помилок, які можуть створювати нові вразливості, а також підвищують розуміння персоналу щодо важливості дотримання політик безпеки.

2.3. Моделювання ризиків та способи їх зниження

Моделювання ризиків у системах управління базами даних, таких як Oracle Database, є важливим інструментом для забезпечення безпеки. Цей процес допомагає оцінити вразливості, проаналізувати потенційні загрози та їхній вплив, а також визначити заходи для мінімізації ризиків. Завдяки цьому організації отримують можливість ефективно розподіляти ресурси, зосереджуючи їх на найбільш критичних аспектах захисту.[28]

2.1.1. Етапи моделювання ризиків

2.1.1.1. Ідентифікація активів

Цей етап передбачає визначення, які саме ресурси системи є критично важливими та потребують першочергового захисту. У контексті Oracle Database до таких активів можуть належати:

- Конфіденційні дані: персональна інформація клієнтів, фінансові звіти, контракти, бізнес-процеси.
- Резервні копії: їхня втрата або компрометація може призвести до незворотних наслідків.
- Критичні функції бази даних: механізми, що забезпечують доступність даних

і функціонування бізнес-процесів.

На цьому етапі важливо не тільки виявити активи, але й зрозуміти, наскільки вони важливі для роботи організації. Це створює основу для подальшого аналізу загроз і вразливостей.

2.1.1.2. Аналіз загроз

На цьому етапі визначаються потенційні джерела загроз, які можуть завдати шкоди виявленим активам. Загрози можуть бути класифіковані за кількома параметрами:

- Зовнішні загрози: кібератаки (SQL-ін’єкції, DDoS), шкідливе програмне забезпечення, атаки на ланцюжок постачання (supply chain attacks).
- Внутрішні загрози: помилки адміністраторів, неналежне налаштування доступу, зловмисні дії співробітників.
- Природні загрози: збої в роботі обладнання, фізичні катастрофи, наприклад, пожежі чи затоплення.

Аналіз загроз дозволяє оцінити, наскільки ймовірною є їхня реалізація, і розробити попереджувальні заходи.[20]

2.1.1.3. Визначення вразливостей

На цьому етапі аналізуються слабкі місця системи, які можуть бути використані для реалізації загроз. Вразливості можуть виникати через:

- Недостатньо захищену автентифікацію: використання стандартних облікових записів або слабких паролів.
- Ненадійне шифрування: відсутність шифрування даних у стані спокою або під час передачі.
- Помилки у конфігурації доступу: занадто широкі привілеї, неналежне розмежування ролей.
- Застаріле програмне забезпечення: використання версій Oracle Database із відомими вразливостями.
- Відсутність моніторингу: неможливість відстеження підозрілих дій у системі.

Виявлення вразливостей є критичним, оскільки вони є точками входу для зловмисників.[20]

2.1.1.4. Оцінка ймовірності реалізації ризиків

На цьому етапі проводиться аналіз того, наскільки імовірним є виникнення кожного з ризиків. Наприклад:

- Високий рівень ризику: SQL-ін’єкції вразливі системи, які використовують динамічні SQL-запити без параметризації.
- Середній рівень ризику: внутрішні загрози від недобросовісних співробітників за відсутності моніторингу.
- Низький рівень ризику: фізична компрометація серверів у системах із належними фізичними заходами безпеки.

Оцінка ймовірності дозволяє зрозуміти, які загрози мають найбільший шанс реалізації, що допомагає правильно розподілити зусилля.[20]

2.1.1.5. Оцінка впливу ризиків

Після оцінки ймовірності визначається, як кожен ризик може вплинути на роботу системи. Наслідки можуть бути такими:

– Витік даних: порушення конфіденційності, що призводить до втрати довіри клієнтів або юридичних санкцій.
– Втрата доступності: збої в роботі системи через DDoS-атаки або перевантаження.
– Порушення цілісності даних: зміна або видалення інформації в результаті зловмисних дій.
– Фінансові збитки: прямі втрати через відновлення роботи або штрафи.

Результати цього етапу визначають, які ризики є найбільш критичними для організації.[21]

2.1.1.6. Пріоритезація ризиків

Останній етап передбачає класифікацію ризиків за рівнем їхньої критичності. Цей процес базується на двох показниках: ймовірності реалізації ризику та рівні його впливу. Ризики поділяються на:

- Критичні: необхідно усунути першочергово (наприклад, SQL-ін’єкції у вразливих системах).
- Середні: потребують уваги, але не є невідкладними (наприклад, резервні копії

без шифрування).

– Низькі: можуть бути враховані як частина довгострокових планів (наприклад, поліпшення фізичного захисту серверів).

Цей етап допомагає організаціям розставити пріоритети та раціонально розподілити ресурси для зниження ризиків**.**

Ретельне опрацювання кожного етапу моделювання ризиків є фундаментом для створення стійкої системи безпеки Oracle Database. Це дозволяє організаціям не лише ефективно захищати свої інформаційні активи, але й реагувати на нові загрози в умовах сучасного цифрового середовища.[20,21]

2.1.2. Типові ризики для Oracle Database

Oracle Database, як одна з найпотужніших і найпоширеніших систем управління базами даних, залишається привабливою мішенню для зловмисників через її популярність і широке використання. Хоча система пропонує безліч інструментів для забезпечення безпеки, вона все ж має низку типових ризиків, які можуть загрожувати як цілісності даних, так і стабільності роботи організації. Нижче розглянуто найбільш поширені ризики, характерні для Oracle Database.[22]

2.1.2.1. Несанкціонований доступ до даних

Однією з найбільш поширених загроз є несанкціонований доступ до чутливих даних. Це може статися через використання слабких механізмів автентифікації або залишення стандартних облікових записів, таких як *SYS* або *SYSTEM*, із незмінними паролями. Уразливість до атак грубої сили (brute force) або фішингових атак також підвищує ризик компрометації облікових записів. Несанкціонований доступ може призвести до викрадення персональної інформації, фінансових даних або іншої конфіденційної інформації, що становить значний ризик для організації.[6]

2.1.2.2. SQL-ін’єкції

SQL-ін’єкції залишаються однією з найпоширеніших загроз для Oracle Database. Проблема виникає, коли в додатках, які взаємодіють із базою даних, використовуються динамічні SQL-запити без належної перевірки введених даних. Зловмисники можуть впроваджувати шкідливий код у SQL-запити, що дозволяє отримати несанкціонований доступ до бази даних, змінювати її структуру,

викрадати дані або навіть видаляти інформацію. Незважаючи на існування підготовлених запитів (prepared statements), багато організацій не впроваджують ці технології через брак обізнаності чи ресурсів, що підвищує ризик атаки.[19-21]

2.1.2.3. Відсутність шифрування даних

Дані, які зберігаються у відкритому вигляді або передаються без шифрування, стають легкою мішенню для зловмисників. Це стосується як даних у стані спокою (на дисках або у резервних копіях), так і даних, що передаються між клієнтом і сервером. Відсутність Transparent Data Encryption (TDE) або SSL/TLS для шифрування робить можливим перехоплення інформації, навіть якщо система має інші рівні захисту. Втрата конфіденційності таких даних, як фінансові звіти чи персональна інформація клієнтів, може призвести до серйозних фінансових і репутаційних втрат.[23]

2.1.2.4. Компрометація резервних копій

Резервні копії є одним із найважливіших елементів забезпечення безпеки баз даних, проте їхній захист часто недооцінюється. У багатьох випадках резервні копії зберігаються у незашифрованому вигляді або без належного контролю доступу. Зловмисники можуть отримати фізичний доступ до носіїв або викрасти резервні копії через компрометацію хмарного сховища. Наслідки такого витоку можуть бути катастрофічними, оскільки резервні копії зазвичай містять повну інформацію, включно з конфіденційними даними.[23]

2.1.2.5. Зниження продуктивності через перевантаження

DDoS-атаки (розподілені атаки на відмову в обслуговуванні) або внутрішні перевантаження системи можуть призвести до втрати доступності бази даних. Такі атаки зазвичай спрямовані на виснаження ресурсів сервера, що унеможливлює його нормальне функціонування. Навіть короткочасна недоступність бази даних може спричинити значні збитки, особливо для організацій, які залежать від онлайн-операцій.[24]

2.1.2.6. Неналежне управління доступом

Неправильна конфігурація привілеїв і ролей у системі створює додаткові ризики. Наприклад, занадто широкі права доступу для звичайних користувачів можуть

дозволити їм виконувати критичні операції, такі як видалення таблиць або змінення політик доступу. Відсутність принципу найменших привілеїв (Least Privilege Principle) значно підвищує ризик ненавмисних помилок або зловмисних дій.[23]

2.1.2.7. Відсутність регулярного оновлення

Oracle регулярно випускає оновлення та патчі безпеки, але багато організацій не приділяють належної уваги їхньому встановленню. Використання застарілих версій програмного забезпечення створює ризик експлуатації відомих вразливостей, таких як переповнення буфера чи уразливі API. Відсутність регулярного оновлення також ускладнює інтеграцію сучасних засобів захисту.[22]

2.1.2.8. Відсутність моніторингу та аудиту

Недостатній моніторинг активності користувачів і відсутність журналювання подій створюють умови для невиявлення підозрілих дій. Зловмисники можуть здійснювати атаки, залишаючись непоміченими, якщо організація не використовує такі інструменти, як Oracle Audit Vault або Database Activity Monitoring (DAM). Це також стосується внутрішніх загроз, які часто виникають через недобросовісні дії співробітників.[25]

2.1.2.9. Загрози від внутрішніх користувачів

Внутрішні користувачі, включно з адміністраторами баз даних, можуть стати джерелом загроз. Це може бути ненавмисна помилка в налаштуваннях або навмисні дії, спрямовані на крадіжку чи знищення даних. Відсутність моніторингу їхньої активності та належного розмежування доступу значно підвищує цей ризик.

Розуміння типових ризиків для Oracle Database є важливим для створення ефективної стратегії захисту. Кожен із цих ризиків потребує детального аналізу, впровадження відповідних заходів безпеки та регулярного перегляду політик для адаптації до нових загроз. [25]

2.1.3. Способи зниження ризиків

Для ефективного захисту Oracle Database від численних загроз необхідно впроваджувати різноманітні заходи, які охоплюють технічні, організаційні та управлінські аспекти безпеки. Зниження ризиків передбачає створення багаторівневої системи захисту, здатної протистояти як зовнішнім атакам, так і

внутрішнім загрозам. Нижче детально розглянуті найбільш ефективні способи мінімізації ризиків. [25]

2.1.3.1. Посилення автентифікації та авторизації

Одним із головних заходів є впровадження більш надійних механізмів автентифікації. Використання багатофакторної автентифікації (MFA) забезпечує додатковий рівень безпеки, вимагаючи від користувачів надання не лише пароля, але й іншого фактора, наприклад:

- Біометричні дані: сканування відбитків пальців або розпізнавання обличчя.
- Токени безпеки: апаратні або програмні ключі.
- Одноразові паролі (OTP): коди, які генеруються на мобільному пристрої або надсилаються електронною поштою.

Крім того, необхідно впроваджувати строгі політики паролів, які передбачають їхню мінімальну довжину, складність і регулярну зміну. Авторизація користувачів повинна ґрунтуватися на принципі найменших привілеїв (Least Privilege Principle), який обмежує доступ до функцій і даних лише в межах необхідного для виконання роботи. [23-24]

2.1.3.2. Використання підготовлених запитів

Для захисту від SQL-ін’єкцій рекомендується уникати використання динамічних SQL-запитів, які є основною причиною цієї загрози. Замість них слід використовувати підготовлені запити (prepared statements) та параметризацію, які автоматично перевіряють введені користувачами дані. Це виключає можливість впровадження шкідливого коду.

Окрім того, доцільно використовувати механізми перевірки введених даних, які включають:

– Валідацію введення: перевірка відповідності даних заздалегідь визначеним правилам (наприклад, числові поля, дати).

– Очищення введених даних: видалення небезпечних символів або конструкцій.

Ці методи не лише мінімізують ризик SQL-ін’єкцій, а й підвищують загальну стабільність роботи системи.[18]

2.1.3.3. Впровадження шифрування даних

Шифрування є основним способом забезпечення конфіденційності даних. У Oracle Database пропонується кілька ефективних методів шифрування:

- Transparent Data Encryption (TDE): забезпечує автоматичне шифрування даних у стані спокою, включно з резервними копіями. Це гарантує, що навіть у разі фізичного доступу до диска зловмисник не зможе прочитати інформацію.
- SSL/TLS: використовується для шифрування даних під час передачі між клієнтом і сервером. Це забезпечує захист інформації від перехоплення в мережі.
- Політики шифрування на рівні полів: дозволяють вибірково шифрувати чутливу інформацію, таку як номери кредитних карток або персональні дані.

Шифрування повинно бути впроваджено на всіх рівнях роботи з даними, щоб гарантувати їхню конфіденційність навіть у разі компрометації.[19,20]

2.1.3.4. Моніторинг і аудит

Моніторинг дій користувачів дозволяє виявляти та запобігати підозрілій активності. Для цього в Oracle Database пропонується використовувати інструменти, такі як:

– Oracle Audit Vault: система централізованого збору та аналізу журналів дій користувачів.

– Database Activity Monitoring (DAM): дозволяє виявляти аномалії в режимі реального часу.

Регулярний аудит дій користувачів забезпечує додатковий рівень контролю над внутрішніми загрозами, особливо якщо йдеться про привілейованих користувачів або адміністраторів. Крім того, автоматичний аналіз журналів подій може допомогти швидко реагувати на потенційні інциденти.[22]

2.1.3.5. Автоматизація оновлень та управління патчами

Багато вразливостей виникають через використання застарілих версій програмного забезпечення. Oracle регулярно випускає патчі безпеки, які закривають відомі вразливості. Автоматизація процесу оновлення дозволяє:

- Своєчасно впроваджувати необхідні зміни.
- Уникати людських помилок, пов’язаних із відкладенням встановлення

оновлень.

Цей підхід гарантує, що система завжди захищена від останніх загроз.[27]

2.1.3.6. Маскування даних

Маскування даних дозволяє приховувати чутливу інформацію в тих випадках, коли доступ до неї не є необхідним. Наприклад:

- У тестових середовищах, де використовуються копії реальної бази даних.
- Для обмеження доступу користувачів із низьким рівнем привілеїв.

Маскування створює фіктивні дані, які зберігають ту саму структуру, але не розкривають реальної інформації. Це знижує ризик витоку навіть у разі компрометації.[27]

2.1.3.7. Резервне копіювання з шифруванням

– Резервні копії є основним засобом відновлення після інцидентів, але вони також потребують захисту. Основні заходи включають:

– Шифрування резервних копій: використання алгоритмів AES для забезпечення конфіденційності.

– Обмеження доступу: впровадження строгих політик доступу до сховищ із резервними копіями.

– Регулярне тестування: перевірка працездатності резервних копій для переконання, що вони можуть бути використані у разі потреби. [27]

2.1.3.8. Освітні заходи для співробітників

Багато ризиків виникають через людський фактор. Навчання адміністраторів і розробників є важливим компонентом захисту. Основні напрямки навчання включають:

- Розуміння принципів безпеки баз даних.
- Ознайомлення з найпоширенішими загрозами, такими як SQL-ін’єкції або DDoS.
- Практичні вправи з налаштування безпечних середовищ.

Освітні заходи сприяють підвищенню загальної обізнаності та зменшують ризики, пов’язані з помилками персоналу.

Запропоновані заходи дозволяють організаціям значно підвищити рівень безпеки

Oracle Database. Комплексний підхід, що включає технічні інструменти, автоматизацію процесів і навчання співробітників, є запорукою стійкості до сучасних загроз. [28]

Розділ 3
РОЗРОБКА ТА ВПРОВАДЖЕННЯ УДОСКОНАЛЕНЬ

3.1. Реалізація на прикладі конкретної бази даних

Для практичної реалізації запропонованих методів удосконалення системи безпеки Oracle Database було обрано реальну базу даних, яка використовується у фінансовій організації. Ця база даних є критично важливим елементом її інфраструктури, оскільки містить чутливу інформацію про клієнтів, транзакції та фінансові операції. Наявність таких даних робить її привабливою мішенню для зловмисників, і водночас накладає високі вимоги до безпеки. Основна мета роботи полягала у забезпеченні конфіденційності, цілісності та доступності даних, з урахуванням мінімізації витрат на технічне обслуговування та підвищення зручності адміністрування.

На момент початку проєкту база даних мала численні вразливості, які створювали загрози для безпеки. Зокрема, використовувалася базова автентифікація на основі паролів, які не відповідали вимогам складності. Загальні облікові записи надавали доступ до даних великій кількості користувачів, що ускладнювало контроль над їхніми діями. Дані зберігалися у відкритому вигляді, без шифрування, що створювало ризик компрометації у разі фізичного доступу до серверів або викрадення резервних копій. Журналювання подій було мінімальним, і організація не мала механізмів для виявлення підозрілої активності в режимі реального часу.[30]

У процесі впровадження удосконалень було реалізовано низку заходів, спрямованих на вирішення цих проблем. Основна увага приділялася вдосконаленню автентифікації та авторизації. Стандартна система автентифікації була замінена на багатофакторну автентифікацію (MFA), яка забезпечує додатковий рівень захисту облікових записів. Це було досягнуто шляхом інтеграції з сервером LDAP, що дозволило використовувати одноразові паролі (OTP) у поєднанні зі звичайними обліковими даними. Такий підхід значно підвищив

стійкість системи до атак грубої сили, зокрема спроб підбору паролів. Крім того, було створено нові політики паролів, які передбачали їхню мінімальну довжину, використання спеціальних символів, регулярну зміну та автоматичне блокування облікових записів після кількох невдалих спроб входу.[29,30]

Особливу увагу було приділено управлінню правами доступу. Для цього було проведено повний аудит облікових записів і розподіл прав доступу за принципом найменших привілеїв (Least Privilege Principle). Кожному користувачу було надано доступ лише до тих даних і функцій, які необхідні для виконання його обов'язків. У базі даних було створено окремі ролі для адміністраторів, менеджерів та операторів із чітким визначенням їхніх повноважень. Для забезпечення контролю доступу на рівні окремих записів було використано технологію Virtual Private Database (VPD), яка дозволяє створювати політики доступу на основі умов. Наприклад, менеджери могли переглядати лише інформацію про своїх клієнтів, що значно знизило ризик несанкціонованого доступу до конфіденційних даних.

Для захисту даних було впроваджено шифрування. Усі конфіденційні дані були зашифровані за допомогою Transparent Data Encryption (TDE), яка забезпечує захист даних у стані спокою. Цей підхід охоплював таблиці з номерами рахунків, деталями транзакцій та іншою чутливою інформацією. Крім того, було активовано SSL/TLS для забезпечення шифрування передачі даних між клієнтами та сервером, що гарантувало їхню конфіденційність навіть у разі перехоплення мережевого трафіку.[28]

Моніторинг дій користувачів і аудит стали ще однією важливою складовою впровадження. Для цього було інтегровано Oracle Audit Vault, який дозволяє централізовано збирати та аналізувати дані про всі операції в базі. Завдяки цьому стало можливим автоматичне створення звітів про підозрілу активність, таких як спроби доступу до захищених таблиць або невдалі спроби входу. Крім того, система автоматично надсилала сповіщення адміністраторам у разі виявлення потенційних загроз, що забезпечило своєчасну реакцію на інциденти.

Щоб захистити систему від SQL-ін'єкцій, було проведено повний аудит SQL-запитів, які використовуються в додатках. Усі динамічні запити були замінені на

підготовлені (prepared statements), які автоматично перевіряють параметри введення. Для додаткового захисту в додатках було впроваджено перевірку введених даних як на стороні клієнта, так і на сервері, що значно знизило ризик впровадження шкідливого коду.[27-29]

У тестових середовищах було впроваджено маскування даних. Це дозволило використовувати реальні дані у вигляді анонімізованих значень, які зберігають свою структуру, але не розкривають чутливої інформації. Маскування забезпечило безпеку під час тестування нових функцій додатків, зменшивши ризик витоку даних у разі компрометації тестових серверів.

Також була впроваджена вдосконалена політика резервного копіювання. Усі резервні копії створювалися автоматично у зашифрованому вигляді за допомогою Recovery Manager (RMAN). Для зберігання використовувалися окремі носії з обмеженим доступом, а регулярні тестові відновлення забезпечували працездатність копій у разі необхідності.

Впроваджені удосконалення дозволили суттєво підвищити рівень безпеки бази даних. Усі виявлені вразливості були усунені, а нові механізми захисту гарантували конфіденційність, цілісність і доступність даних навіть за умов сучасних загроз. Автоматизація процесів спростила адміністрування та зменшила витрати на підтримку системи. Завдяки цьому організація змогла забезпечити надійний захист своєї інформації, відповідно до найкращих практик кібербезпеки.

Додатковою перевагою впроваджених удосконалень стала можливість точного аналізу роботи системи безпеки у реальному часі. Наприклад, завдяки Oracle Audit Vault стало можливим не лише фіксувати всі операції в базі даних, а й відслідковувати тенденції, які можуть свідчити про потенційні загрози. У випадках, коли відбувалися повторювані невдалі спроби входу до системи, адміністратори негайно отримували сповіщення, що дозволяло швидко блокувати підозрілі облікові записи та проводити розслідування.[30]

Покращення політики резервного копіювання також значно знизило ризики, пов’язані з втратою даних. У разі потенційного збою чи атаки, організація могла оперативно відновити базу даних до останнього стабільного стану, що

забезпечувало безперервність бізнес-процесів. Шифрування резервних копій гарантувало, що навіть у разі фізичної втрати носіїв дані залишатимуться недоступними для зловмисників.

Інтеграція маскування даних у тестових середовищах стала важливим кроком для захисту інформації під час розробки нових функцій. Раніше тестові команди використовували копії реальних баз даних, що створювало ризик витоку конфіденційної інформації. Завдяки маскуванню стало можливим використовувати дані, які за своєю структурою та поведінкою ідентичні реальним, але не містять жодної чутливої інформації.[30]

Ще одним важливим результатом було зниження ризику SQL-ін’єкцій. Замінені динамічні запити на підготовлені (prepared statements) не лише покращили безпеку, але й зробили систему стійкішою до людських помилок у розробці. Навіть якщо зловмисник намагався впровадити шкідливий код, система автоматично блокувала подібні спроби завдяки перевіркам на рівні серверного коду.

Інтеграція SSL/TLS для передачі даних між клієнтами та сервером також мала важливе значення. Це усунуло одну з головних вразливостей, пов’язаних із перехопленням даних у мережі. Навіть у випадку, якщо зловмисники отримують доступ до мережевого трафіку, зашифровані дані залишаються недоступними для читання без відповідного ключа дешифрування.

Організаційні заходи, такі як навчання співробітників, також відіграли важливу роль у підвищенні загального рівня безпеки. Адміністратори баз даних і розробники пройшли курси з кібербезпеки, що дозволило їм краще розуміти сучасні загрози та уникати поширених помилок у конфігурації системи. Завдяки цьому вдалося не лише знизити кількість потенційних вразливостей, але й створити культуру безпеки в організації.

Загалом впровадження нових заходів і технологій дозволило суттєво підвищити рівень безпеки бази даних, зробивши її більш стійкою до сучасних загроз. Комплексний підхід, який включав як технічні рішення, так і організаційні заходи, забезпечив захист даних, підвищив довіру клієнтів до організації та створив надійну основу для її подальшого розвитку.[27]

3.2. Результати впровадження

Після реалізації запропонованих заходів з удосконалення безпеки система бази даних зазнала якісних змін, що позитивно вплинуло на її стійкість до сучасних кіберзагроз. Основним досягненням стало суттєве зниження ризиків несанкціонованого доступу, SQL-ін’єкцій та витоку даних.

Шифрування даних за допомогою Transparent Data Encryption (TDE) дозволило захистити конфіденційну інформацію навіть у разі фізичного компрометування серверів або резервних копій. Дані тепер надійно зберігаються, що гарантує їхню конфіденційність у разі інцидентів. Активоване шифрування під час передачі даних (SSL/TLS) усунуло ризики перехоплення інформації під час роботи з мережею, забезпечивши захист комунікацій між клієнтами та сервером.

Моніторинг і аудит стали важливим інструментом для забезпечення прозорості системи. Завдяки Oracle Audit Vault було налагоджено централізований збір та аналіз даних про дії користувачів. Цей інструмент дозволив адміністраторам швидко реагувати на потенційно небезпечні дії, такі як невдалі спроби входу чи несанкціоновані спроби доступу до конфіденційної інформації. Автоматичні сповіщення про підозрілу активність стали ефективним засобом для своєчасного виявлення загроз.

Маскування даних, впроваджене для тестових середовищ, дозволило мінімізувати ризики розкриття конфіденційної інформації під час розробки та тестування нових функцій. Реальні дані тепер замінюються на анонімізовані значення, що повністю зберігають свою структуру, але не містять жодної чутливої інформації.

Вдосконалена політика резервного копіювання забезпечила надійне відновлення даних у разі збоїв. Резервні копії тепер створюються автоматично, зашифровані й зберігаються на захищених носіях. Регулярне тестування процесу відновлення гарантує, що всі критично важливі дані можуть бути оперативно відновлені у разі потреби.

3.3. Практичні аспекти: код, скрипти, архітектурні рішення

Для забезпечення надійного захисту Oracle Database у сучасних умовах потрібні не

тільки архітектурні рішення та налаштування, але й глибока інтеграція з рівнем додатків. У цій частині розглядаються основні аспекти впровадження практичних удосконалень на прикладі Java-коду, що забезпечує взаємодію з базою даних і підвищує її безпеку.

Вдосконалення автентифікації

Автентифікація є ключовим елементом у забезпеченні безпеки баз даних. Для підвищення її надійності було впроваджено багатофакторну автентифікацію (MFA). Одним із реалізованих методів стало використання LDAP (Lightweight Directory Access Protocol) для централізованого управління обліковими записами. LDAP забезпечує зручність і гнучкість в адмініструванні доступу, а також підвищує загальний рівень захисту.

Нижче наведений приклад коду на Java демонструє, як можна інтегрувати автентифікацію користувачів із сервером LDAP.

```
import javax.naming.Context;
import javax.naming.NamingException;
import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;
import java.util.Hashtable;

public class LdapAuthenticator {
    public static boolean authenticate(String username, String password) {
        Hashtable<String, String> env = new Hashtable<>();
        env.put(Context.INITIAL_CONTEXT_FACTORY,
"com.sun.jndi.ldap.LdapCtxFactory");
        env.put(Context.PROVIDER_URL, "ldaps://ldap.example.com:636");
        env.put(Context.SECURITY_AUTHENTICATION, "simple");
        env.put(Context.SECURITY_PRINCIPAL, "uid=" + username +
",ou=users,dc=example,dc=com");
        env.put(Context.SECURITY_CREDENTIALS, password);
```

```
        try {
            DirContext ctx = new InitialDirContext(env);
            ctx.close();
            return true; // Успішна автентифікація
        } catch (NamingException e) {
            return false; // Помилка автентифікації
        }
    }
}
```

Пояснення коду

Бібліотеки та ініціалізація контексту

Код використовує Java Naming and Directory Interface (JNDI) для взаємодії з LDAP-сервером. Клас InitialDirContext служить точкою доступу до сервісів каталогу, зокрема, для здійснення автентифікації.

Налаштування параметрів підключення

Об'єкт Hashtable містить параметри для встановлення зв’язку з LDAP-сервером:

– Context.INITIAL_CONTEXT_FACTORY: визначає фабрику контексту для роботи з LDAP.

– Context.PROVIDER_URL: вказує адресу LDAP-сервера (у прикладі використовується протокол ldaps, що забезпечує шифрування передачі даних).

– Context.SECURITY_AUTHENTICATION: визначає тип автентифікації, у цьому випадку – "simple".

– Context.SECURITY_PRINCIPAL: задає DN (Distinguished Name) користувача, що використовується для автентифікації.

– Context.SECURITY_CREDENTIALS: містить пароль користувача.

Встановлення з'єднання

Метод InitialDirContext створює підключення до LDAP-сервера з використанням заданих параметрів. Якщо з'єднання встановлено успішно, користувач вважається автентифікованим.

Закриття з'єднання

Після успішної автентифікації контекст DirContext закривається для збереження ресурсів і безпеки.

Обробка помилок

У разі виникнення помилки (наприклад, неправильного пароля або недоступності сервера), метод повертає false, сигналізуючи про невдалу спробу автентифікації.

Переваги такого підходу

– **Підвищена безпека:** Інтеграція з LDAP забезпечує централізоване управління обліковими записами та дозволяє впроваджувати додаткові рівні автентифікації, такі як OTP (одноразові паролі).

– **Гнучкість:** LDAP легко інтегрується з іншими сервісами безпеки, такими як багатофакторна автентифікація чи політики доступу на основі ролей.

– **Шифрування:** Використання LDAPS (Secure LDAP) гарантує захищену передачу облікових даних між клієнтом і сервером.

– **Простота адміністрування:** Централізоване управління дозволяє легко змінювати права доступу або блокувати облікові записи без внесення змін до кожного додатку.

– **Масштабованість:** LDAP підтримує роботу з великою кількістю облікових записів, що робить його оптимальним для організацій із розгалуженою інфраструктурою.

Використання LDAP для автентифікації користувачів у Oracle Database значно підвищує безпеку системи. Цей підхід дозволяє знизити ризик несанкціонованого доступу, зменшити навантаження на адміністраторів і впровадити додаткові рівні захисту. Реалізація на Java демонструє простоту впровадження такого рішення у практичних умовах.

Захист від SQL-ін’єкцій

SQL-ін’єкції є однією з найпоширеніших та найбільш небезпечних загроз для систем управління базами даних. Цей тип атаки дозволяє зловмисникам впроваджувати шкідливий код через введення даних у SQL-запити, що може призвести до викрадення, модифікації або навіть видалення даних. Для мінімізації ризику SQL-ін’єкцій у додатку було реалізовано використання підготовлених

запитів (prepared statements), які гарантують безпеку обробки даних користувачів. Нижче наведено приклад коду на Java, що демонструє використання параметризованих запитів для доступу до бази даних.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class SecureQueryExample {
    private static final String DB_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    private static final String DB_USER = "db_user";
    private static final String DB_PASSWORD = "password";

    public static void fetchCustomerData(int regionId) {
        String query = "SELECT * FROM customers WHERE region_id = ?";

        try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
             PreparedStatement pstmt = conn.prepareStatement(query)) {

            pstmt.setInt(1, regionId);
            ResultSet rs = pstmt.executeQuery();

            while (rs.next()) {
                System.out.println("Customer ID: " + rs.getInt("customer_id"));
                System.out.println("Customer Name: " + rs.getString("customer_name"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
    }
}
```

Пояснення коду

Налаштування з’єднання

У коді використовується стандартний драйвер JDBC для підключення до Oracle Database. Конфігураційні параметри, такі як URL бази даних, ім'я користувача та пароль, вказані у змінних DB_URL, DB_USER і DB_PASSWORD.

- **DB_URL:** Визначає адресу бази даних, включаючи тип драйвера (thin), хост (localhost) і ідентифікатор сервісу (xe).
- **DB_USER і DB_PASSWORD:** Зберігають облікові дані для доступу до бази даних.

Підготовлений запит (prepared statement)

Ключовим елементом захисту є використання класу PreparedStatement. На відміну від звичайних SQL-запитів, підготовлений запит відокремлює код запиту від даних, що вводяться користувачем. Це усуває можливість впровадження шкідливого SQL-коду.

Метод setInt(1, regionId) задає значення для першого параметра запиту (?) на основі введених користувачем даних. Оскільки значення перевіряється та обробляється драйвером JDBC, шкідливий код не буде виконано.

Виконання запиту та обробка результатів

Метод executeQuery() виконує SQL-запит, а об’єкт ResultSet дозволяє отримати результати. У циклі while дані зчитуються та виводяться на екран.

Використання конструкції try-with-resources

Ця конструкція забезпечує автоматичне закриття ресурсів, таких як з'єднання (Connection) і підготовлений запит (PreparedStatement), після завершення роботи. Це не тільки покращує управління ресурсами, але й підвищує безпеку, знижуючи ризик витоку даних.

Обробка винятків

У разі виникнення помилки (наприклад, через недоступність бази даних або некоректні дані), виконується блок catch, який фіксує помилку та виводить її на

екран. Це дозволяє уникнути краху програми та полегшує діагностику проблем.

Переваги використання підготовлених запитів

– **Захист від SQL-ін’єкцій:** Підготовлені запити відокремлюють дані користувача від SQL-коду, що унеможливлює впровадження шкідливих конструкцій.

– **Автоматична перевірка даних:** JDBC драйвер автоматично перевіряє введені значення на відповідність типу, що знижує ризик помилок.

– **Покращена продуктивність:** Підготовлені запити оптимізуються та кешуються базою даних, що підвищує швидкість виконання однакових запитів.

– **Універсальність:** Цей метод працює з будь-якими типами даних, зокрема текстом, числами та датами, без ризику ін’єкцій.

– **Зручність використання:** Легко інтегрується у вже існуючий код із мінімальними змінами.

Реалізація підготовлених запитів у додатках, які працюють із базами даних, є одним із найбільш ефективних способів захисту від SQL-ін’єкцій. Представлений код демонструє простоту та ефективність такого підходу, який не лише підвищує безпеку, але й забезпечує стабільну та ефективну роботу додатку. У поєднанні з іншими заходами, такими як контроль доступу та моніторинг активності, використання підготовлених запитів робить систему стійкою до сучасних загроз.

Логування та моніторинг у додатку

Для забезпечення прозорості операцій у системі та підвищення рівня безпеки було впроваджено централізоване логування, яке фіксує всі дії користувачів. Це дозволяє зберігати повну історію активності, аналізувати підозрілу поведінку, а також швидко реагувати на потенційні загрози. Як основу для реалізації логування було використано популярну бібліотеку Log4j, яка забезпечує високу продуктивність і гнучкість.

Нижче наведений приклад коду демонструє, як реалізувати механізм централізованого логування з використанням Log4j.

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

```
public class AuditLogger {
    private static final Logger logger = LogManager.getLogger(AuditLogger.class);

    public static void logAction(String username, String action, String details) {
        logger.info("User: " + username + ", Action: " + action + ", Details: " + details);
    }

    public static void main(String[] args) {
        logAction("john_doe", "LOGIN", "Successful login");
        logAction("john_doe", "ACCESS_TABLE", "Accessed customers table");
    }
}
```

Пояснення коду

Використання Log4j

Бібліотека Log4j є одним із найпопулярніших інструментів для створення систем логування у додатках на Java. Вона дозволяє фіксувати події на різних рівнях, таких як INFO, WARN, ERROR, що спрощує аналіз даних.

- LogManager.getLogger(AuditLogger.class) створює екземпляр логера, який прив'язаний до класу AuditLogger. Це забезпечує чітке розмежування журналів подій між різними компонентами програми.

Метод для фіксації подій

Метод logAction приймає три параметри:

- **username:** Ім’я користувача, який виконує дію.
- **action:** Тип операції, наприклад, вхід у систему або доступ до таблиці.
- **details:** Додаткова інформація, яка описує контекст операції, наприклад, результат дії або ім’я об’єкта, до якого був доступ.

Виклик logger.info зберігає цю інформацію на рівні INFO, що є стандартним для реєстрації звичайних подій.

Приклади використання

У методі main представлені два приклади:

– Запис успішного входу користувача john_doe.

– Запис доступу до таблиці customers.

Це дозволяє створити деталізований журнал дій користувачів у системі.

Переваги впровадження централізованого логування

– **Прозорість операцій:** Логування дозволяє фіксувати кожну дію користувача, створюючи повну історію активності.

– **Аналіз інцидентів:** У разі підозрілої активності журнали подій можна використовувати для ідентифікації причини проблеми та джерела загрози.

– **Попередження загроз:** Логування можна інтегрувати з системами моніторингу, які автоматично аналізують журнали та генерують сповіщення в разі аномалій.

– **Гнучкість:** Log4j дозволяє зберігати журнали у різних форматах (файли, бази даних, хмарні сервіси), адаптуючи систему до потреб організації.

– **Простота впровадження:** Код легко інтегрується в існуючі додатки, що дозволяє швидко налагодити систему логування.

Використання у реальних умовах

Записані журнали подій можуть бути проаналізовані за допомогою спеціалізованих інструментів, таких як SIEM-системи (Security Information and Event Management). Це дозволяє:

– Виявляти шаблони підозрілої активності, наприклад, багаторазові невдалі спроби входу.

– Автоматично генерувати сповіщення для адміністраторів у разі аномалій.

– Створювати аналітичні звіти для оцінки ефективності заходів безпеки.

Впровадження централізованого логування за допомогою Log4j є важливим кроком у створенні стійкої системи безпеки. Цей підхід дозволяє не лише забезпечити прозорість операцій, але й значно спростити аналіз потенційних інцидентів. Представлений код демонструє, як легко реалізувати цей механізм у Java-додатках, забезпечивши надійний контроль за активністю користувачів у системі.

Шифрування конфіденційних даних у додатку

Для забезпечення конфіденційності даних, які зберігаються в базі, було впроваджено шифрування на рівні додатка. Це дозволяє гарантувати, що дані залишаються захищеними навіть у разі несанкціонованого доступу до бази. Шифрування реалізовано з використанням стандарту AES (Advanced Encryption Standard), який забезпечує високу надійність та ефективність.

Нижче наведений приклад коду на Java демонструє, як можна реалізувати шифрування та дешифрування даних із використанням AES.

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class EncryptionUtil {
    private static final String ALGORITHM = "AES";

    public static String encrypt(String data, String key) throws Exception {
        SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), ALGORITHM);
        Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] encrypted = cipher.doFinal(data.getBytes());
        return Base64.getEncoder().encodeToString(encrypted);
    }

    public static String decrypt(String encryptedData, String key) throws Exception {
        SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), ALGORITHM);
        Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] decrypted = cipher.doFinal(Base64.getDecoder().decode(encryptedData));
        return new String(decrypted);
    }
```

```
    public static void main(String[] args) throws Exception {
        String key = "1234567890123456"; // 16-байтний ключ AES
        String sensitiveData = "Sensitive Information";

        String encrypted = encrypt(sensitiveData, key);
        System.out.println("Encrypted: " + encrypted);

        String decrypted = decrypt(encrypted, key);
        System.out.println("Decrypted: " + decrypted);
    }
}
```

Пояснення коду

Вибір алгоритму

Алгоритм AES (Advanced Encryption Standard) є широко використовуваним стандартом симетричного шифрування. У цьому прикладі використовується ключ довжиною 16 байт (128 біт), який забезпечує надійний рівень захисту.

Створення секретного ключа

Ключ для шифрування створюється з використанням класу SecretKeySpec. Він приймає байтове подання ключа та назву алгоритму. Важливо, щоб ключ мав відповідну довжину (16, 24 або 32 байти для AES).

Шифрування даних

Метод encrypt виконує шифрування даних:

- Створюється об’єкт Cipher із зазначенням алгоритму AES.
- Ініціалізується режим шифрування (Cipher.ENCRYPT_MODE) із використанням секретного ключа.
- Дані конвертуються у байти та шифруються за допомогою методу doFinal.

Шифровані дані кодуються у формат Base64 для зручного збереження або передачі.

Дешифрування даних

Метод decrypt виконує зворотний процес:

– Дані конвертуються з формату Base64 у байти.

– Ініціалізується режим дешифрування (Cipher.DECRYPT_MODE) із використанням того ж секретного ключа.

– Використовується метод doFinal для розшифрування даних.

Демонстрація роботи

У методі main демонструється процес шифрування та дешифрування:

– Створюється ключ фіксованої довжини.

– Дані шифруються та зберігаються у змінній encrypted.

– Далі ці дані розшифровуються та перевіряється, чи співпадає вихідний текст із розшифрованим.

Переваги шифрування на рівні додатка

– **Конфіденційність даних:** Шифрування гарантує, що навіть у разі доступу до бази або файлів резервного копіювання зловмисник не зможе прочитати дані без ключа.

– **Захист під час передачі:** Шифрування даних перед записом у базу підвищує захист від атак, спрямованих на перехоплення трафіку між додатком і базою.

– **Простота інтеграції:** Реалізація шифрування на рівні додатка не потребує змін у структурі бази даних.

– **Гнучкість:** Ключі для шифрування можуть змінюватися залежно від політики безпеки, що підвищує стійкість системи.

Рекомендації щодо використання

– **Управління ключами:** Ключі повинні зберігатися у захищених сховищах (наприклад, апаратних модулях безпеки) та бути доступними лише авторизованим додаткам.

– **Ротація ключів:** Рекомендується регулярно змінювати ключі для шифрування, щоб мінімізувати ризик компрометації.

– **Додатковий захист:** Використовуйте TLS для захисту даних під час передачі між клієнтами й сервером, навіть якщо дані вже зашифровані.

Реалізація шифрування даних на рівні додатка є ефективним заходом для забезпечення конфіденційності критично важливої інформації. Використання AES

у поєднанні з належним управлінням ключами гарантує захист даних навіть у разі компрометації бази даних. Представлений код демонструє, як легко реалізувати цей підхід у Java-додатках, забезпечуючи високий рівень безпеки інформації.

3.4. Архітектурні рішення

Архітектура бази даних є критично важливим компонентом, який визначає її продуктивність, безпеку та надійність. У рамках удосконалення системи Oracle Database було впроваджено кілька ключових архітектурних рішень, спрямованих на забезпечення стійкості до збоїв, розподілу навантаження та мінімізації ризиків компрометації даних.

Одним із основних рішень стало впровадження Oracle Real Application Clusters (RAC). Ця технологія дозволяє об’єднувати кілька серверів у кластер, який працює як єдина база даних. Використання Oracle RAC забезпечує автоматичний розподіл навантаження між вузлами кластера. У ситуаціях, коли кількість запитів до бази даних значно зростає, RAC дозволяє перенаправляти запити на менш завантажені вузли, що гарантує високу продуктивність навіть у періоди пікової активності. Крім того, ця технологія забезпечує відмовостійкість: якщо один із серверів виходить із ладу, його функції автоматично беруть на себе інші вузли кластеру. Це дозволяє зберігати безперервність роботи бази даних і уникати втрати доступності.[26,27]

Іншим важливим рішенням стало розділення середовищ для розробки, тестування та продуктивної експлуатації. Це дозволило створити окремі ізольовані середовища, кожне з яких виконує свої специфічні завдання. Середовище розробки використовується для створення нових функцій та вдосконалень. У цьому середовищі розробники працюють із тестовими даними, що запобігає випадковому впливу на продуктивну базу. Середовище тестування дозволяє перевіряти нові функції перед їх впровадженням у реальну експлуатацію. У тестовому середовищі використовуються масковані дані, які мають структуру реальних, але не містять конфіденційної інформації. Це мінімізує ризик витоку даних і забезпечує високий рівень безпеки. Продуктивне середовище залишається ізольованим від розробки та тестування, що дозволяє уникнути випадкових помилок або некоректних оновлень.

Ще одним важливим аспектом стала інтеграція бази даних із хмарною платформою

для зберігання резервних копій. Це рішення забезпечило додатковий рівень захисту від втрати даних і підвищило загальну відмовостійкість системи. Усі резервні копії зберігаються у зашифрованому вигляді, що гарантує їхню конфіденційність навіть у разі компрометації хмарного сховища. Крім того, хмарна платформа дозволяє зберігати копії у кількох географічно розподілених дата-центрах. Це створює додатковий рівень захисту від катастрофічних подій, таких як пожежі або природні катаклізми, які можуть вплинути на локальне зберігання.

Впровадження цих архітектурних рішень дало змогу значно підвищити ефективність роботи системи. Продуктивність бази даних залишалася стабільною навіть за умов зростання кількості користувачів і обсягу даних. Відмовостійкість гарантувала безперервну доступність інформації, навіть у разі збоїв або аварій. А інтеграція з хмарною платформою забезпечила гнучкість у використанні ресурсів і економічність, оскільки дозволила масштабувати обсяг сховища відповідно до потреб організації.[25-28]

Таким чином, реалізація архітектурних рішень стала важливим кроком у створенні сучасної, безпечної та надійної бази даних. Комплексний підхід, який включає використання Oracle RAC, ізоляцію середовищ і інтеграцію з хмарними технологіями, забезпечив високу стійкість системи до сучасних загроз і зробив її готовою до подальшого зростання та вдосконалення.

Додатково, впроваджені архітектурні рішення сприяли поліпшенню управління безпекою та підтримці бази даних. Наприклад, ізоляція середовищ дозволила зменшити ризик випадкових помилок під час тестування нових функцій або внесення змін до системи. Це також спростило процес моніторингу та виявлення потенційних проблем, оскільки тестові середовища не впливали на продуктивну базу, а розробники могли працювати незалежно від основної системи.

Використання хмарної платформи для зберігання резервних копій надало додаткову перевагу в умовах масштабування. У разі збільшення обсягу даних організація могла швидко адаптувати хмарне сховище до нових вимог без необхідності інвестувати в локальну інфраструктуру. Завдяки використанню зашифрованих резервних копій у хмарі було гарантовано, що навіть у разі

компрометації хмарного провайдера дані залишаться недоступними для зловмисників без відповідних ключів дешифрування.

Впровадження Oracle RAC також надало можливість обробляти критично важливі операції в реальному часі з високим рівнем надійності. Це було особливо важливим для організації, яка обслуговує фінансові транзакції, де навіть незначні затримки або збої можуть призвести до серйозних наслідків. Завдяки RAC система залишалася функціональною навіть у разі виходу з ладу одного з вузлів, що мінімізувало ризики простою та зниження продуктивності.

Інтеграція хмарних рішень також забезпечила легкість у виконанні регулярних тестувань резервних копій. Цей підхід дозволив адміністраторам перевіряти працездатність резервів без втручання у продуктивне середовище. Регулярні тестування підвищували впевненість у готовності до швидкого відновлення даних у разі інцидентів.[27]

Важливою складовою впровадження таких архітектурних рішень стало підвищення загальної кібербезпеки системи. Завдяки ізоляції середовищ зменшився ризик витоку даних через тестові системи, які зазвичай є менш захищеними. Крім того, використання хмарного сховища унеможливило фізичну компрометацію локальних резервних копій.

У підсумку, ці рішення створили фундамент для масштабованої, безпечної та надійної інфраструктури бази даних. Організація отримала можливість адаптуватися до зростання бізнесу, підвищувати рівень обслуговування клієнтів і гарантувати безперервність роботи навіть у найскладніших умовах. Такі архітектурні зміни підкреслюють важливість стратегічного підходу до управління базами даних у сучасному технологічному середовищі.

Архітектурні вдосконалення також створили умови для більш ефективної взаємодії між різними командами в організації. Ізоляція середовищ дала змогу чітко розмежувати роботу розробників, тестувальників і адміністраторів, мінімізуючи конфлікти між цими групами. Наприклад, розробники могли впроваджувати нові функції у своєму середовищі, не турбуючись про те, що їхні дії вплинуть на продуктивну базу. Тестувальники, у свою чергу, отримували доступ до

середовища, яке відтворює реальні умови, але з використанням маскованих даних. Це дозволило їм проводити ретельну перевірку функцій без ризику для конфіденційності клієнтів.[26]

Впровадження Oracle RAC відкрило нові можливості для оптимізації продуктивності додатків, які взаємодіють із базою даних. Розподіл запитів між кількома вузлами дозволив уникнути перевантаження одного сервера, що часто є причиною затримок і зниження швидкодії системи. Це особливо важливо для високонавантажених систем, де кожна секунда затримки може вплинути на задоволеність клієнтів і репутацію компанії. RAC також дозволив швидко масштабувати ресурси у разі зростання бізнесу, додаючи нові вузли до кластеру без значних простоїв.

Інтеграція з хмарною платформою забезпечила ще одну важливу перевагу – автоматизацію процесів резервного копіювання та відновлення. Адміністратори більше не витрачали значний час на ручне створення резервів або переміщення копій у безпечне місце. Натомість автоматизовані механізми хмарного сховища забезпечували регулярне створення шифрованих резервних копій, які зберігалися у безпечних дата-центрах із географічною розподіленістю. У разі потреби відновлення відбувалося швидко й безпечно, мінімізуючи ризики втрати даних.

Особливої уваги заслуговує підвищення загальної стійкості системи до кіберзагроз. Завдяки впровадженню багаторівневого підходу до захисту, включаючи шифрування резервних копій, ізоляцію середовищ та моніторинг дій користувачів, організація створила систему, яка не тільки відповідає сучасним вимогам безпеки, але й готова до майбутніх викликів. Наприклад, регулярні аудити резервних копій і тестування планів відновлення гарантували готовність організації до будь-яких форс-мажорних ситуацій, зокрема природних катастроф або цілеспрямованих атак на інфраструктуру.[3-6]

У довгостроковій перспективі такі архітектурні зміни сприяли значному зниженню витрат на обслуговування бази даних. Оптимізація процесів, автоматизація резервного копіювання, зменшення кількості інцидентів та ефективне управління середовищами дозволили зосередити ресурси на стратегічних завданнях, таких як

розширення бізнесу або впровадження інновацій. Завдяки цьому база даних стала не просто частиною інфраструктури, а ключовим активом, який підтримує зростання компанії та підвищує її конкурентоспроможність на ринку.

Отже, впроваджені архітектурні рішення продемонстрували ефективність комплексного підходу до управління базами даних. Вони забезпечили високий рівень надійності, масштабованості та безпеки, що є критично важливими для сучасних організацій. Це підтверджує, що інвестиції у вдосконалення інфраструктури баз даних мають довготривалий позитивний вплив на весь бізнес.

3.5. Аналіз ефективності запропонованих рішень

Впровадження удосконалень у систему безпеки Oracle Database призвело до суттєвого покращення її захисту від внутрішніх і зовнішніх загроз. Аналіз ефективності запропонованих рішень базується на порівнянні стану системи до і після реалізації змін за кількома ключовими критеріями, зокрема захист конфіденційності, цілісність даних, доступність системи, ризик компрометації та рівень управління доступом.[9]

Захист конфіденційності

Конфіденційність даних є однією з найважливіших вимог до інформаційних систем, особливо коли йдеться про роботу з критично важливою інформацією, такою як особисті дані клієнтів або фінансові транзакції. До впровадження запропонованих удосконалень система безпеки Oracle Database мала суттєві недоліки у забезпеченні конфіденційності, що створювало ризик значних втрат інформації.

До впровадження

До впровадження вдосконалень конфіденційність даних не була належним чином забезпечена. Дані, що зберігалися у базі, не шифрувалися, залишаючи їх відкритими для прочитання у разі фізичного доступу до сервера чи резервних копій. У разі компрометації фізичних носіїв зловмисники могли легко отримати доступ до всіх записів, включно з конфіденційною інформацією. Це створювало

високий ризик витоку даних, особливо для організацій, які оперують персональними або фінансовими даними клієнтів.[10]

Ще одним значним недоліком було те, що під час передачі даних мережею вони залишалися незахищеними, оскільки протоколи шифрування на рівні транспорту (наприклад, SSL/TLS) не використовувалися. Це означало, що у випадку перехоплення мережевого трафіку зловмисники могли отримати доступ до переданих даних у відкритому вигляді. Особливо вразливими були комунікації між клієнтськими додатками та сервером бази даних, які часто використовуються для доступу до критично важливої інформації.

Відсутність шифрування як у стані спокою, так і під час передачі робила систему надзвичайно вразливою до внутрішніх і зовнішніх загроз, включно з атаками типу "людина посередині" (MITM), викраденням резервних копій або несанкціонованим доступом до серверів.[12]

Після впровадження

Після впровадження удосконалень ситуація кардинально змінилася. Основним заходом для захисту конфіденційності стало використання Transparent Data Encryption (TDE). Цей механізм забезпечує автоматичне шифрування всіх даних у стані спокою, зокрема таблиць, індексів і журналів транзакцій. Завдяки TDE навіть у разі фізичного доступу до файлів бази даних зловмисники не зможуть їх прочитати без відповідних ключів дешифрування, які зберігаються у захищеному модулі безпеки.

Окрім шифрування даних у стані спокою, було впроваджено шифрування під час передачі даних за допомогою протоколів SSL/TLS. Це рішення забезпечило захист комунікацій між клієнтськими додатками та сервером бази даних. Дані, які передаються мережею, тепер надійно шифруються, унеможливлюючи їх перехоплення навіть у разі атак на мережевий канал. Такі удосконалення стали особливо важливими для захисту даних у розподілених системах і хмарних середовищах, де використання незахищених каналів зв’язку раніше становило суттєву загрозу.

Додатково була впроваджена політика регулярного моніторингу конфіденційності

даних. Адміністратори отримують сповіщення про будь-які спроби доступу до захищених файлів бази даних, що дозволяє оперативно реагувати на потенційні інциденти. Це значно підвищило загальний рівень контролю над конфіденційною інформацією.[15]

Результати

Після впровадження цих удосконалень ризик компрометації конфіденційної інформації був знижений до мінімального рівня. Шифрування даних у стані спокою за допомогою TDE унеможливило доступ до бази навіть у разі фізичної компрометації серверів або резервних копій. Використання SSL/TLS для захисту мережевого трафіку гарантувало безпечну передачу даних між клієнтами та сервером. Завдяки цим заходам система стала відповідати найвищим стандартам захисту конфіденційності, що особливо важливо для організацій, які працюють із чутливими даними клієнтів.

Отже, удосконалення системи безпеки Oracle Database забезпечило комплексний захист конфіденційності даних як у процесі зберігання, так і передачі. Це значно підвищило довіру до системи з боку користувачів та партнерів, одночасно забезпечуючи відповідність сучасним стандартам кібербезпеки.[15]

Управління доступом

Управління доступом є основою захисту даних у будь-якій інформаційній системі. Від того, наскільки ефективно реалізовано контроль над доступом, залежить безпека конфіденційної інформації та стабільність роботи системи. До впровадження запропонованих удосконалень Oracle Database мала численні вразливості у сфері управління доступом, що створювало значний ризик несанкціонованого доступу та маніпуляцій із даними.

До впровадження

Система управління доступом у базі даних до впровадження змін характеризувалася низкою критичних проблем. Одна з найбільш поширених вразливостей полягала у використанні стандартних облікових записів, таких як SYS і SYSTEM, із незмінними або слабкими паролями. Це створювало високий ризик несанкціонованого доступу, особливо в організаціях, де адміністративна

команда не приділяла належної уваги питанням безпеки.

Ще однією серйозною проблемою було те, що більшість користувачів мали надмірно широкі привілеї. У багатьох випадках співробітники отримували доступ до даних і функцій, які не були потрібні для виконання їхніх посадових обов’язків. Такий підхід підвищував ризик внутрішніх загроз, оскільки звичайні користувачі могли випадково або навмисно порушити цілісність даних чи розкрити конфіденційну інформацію. [15]

Політики доступу не були достатньо гнучкими та чіткими. Відсутність обмежень на рівні окремих записів означала, що користувачі могли переглядати більше інформації, ніж їм було необхідно. Наприклад, менеджери могли отримувати доступ до даних клієнтів, які не входили до їхнього регіону відповідальності. Це не тільки підвищувало ризик витоку інформації, але й ускладнювало контроль за дотриманням конфіденційності.

Після впровадження

Після впровадження удосконалень система управління доступом була значно вдосконалена. Основним принципом, який ліг в основу нової моделі доступу, став принцип найменших привілеїв (Least Privilege Principle). Всі облікові записи були переглянуті, а надмірні привілеї користувачів – скасовані. Кожен користувач отримав доступ лише до тих даних і функцій, які необхідні для виконання його обов’язків. Це рішення зменшило кількість надмірних прав і дозволило створити більш контрольоване середовище.

Додатково було реалізовано використання ролей для управління доступом. Для кожної групи користувачів були створені чітко визначені ролі з обмеженими правами. Наприклад, адміністратори мали доступ до системних функцій, а менеджери – до даних клієнтів тільки у межах їхньої відповідальності. Це забезпечило гнучкість і простоту управління правами доступу.

Одним із ключових рішень стало впровадження Virtual Private Database (VPD). Ця технологія дозволяє створювати політики доступу на рівні окремих записів, базуючись на ролі користувача. Наприклад, за допомогою VPD було налаштовано обмеження, які дозволяли менеджерам переглядати лише ті записи, що стосуються

їхнього регіону. Це забезпечило додатковий рівень захисту конфіденційної інформації. [15]

Окрему увагу приділили посиленню захисту адміністративних облікових записів. Для всіх привілейованих користувачів були встановлені політики використання складних паролів із регулярною зміною. Крім того, було впроваджено багатофакторну автентифікацію (MFA), яка гарантує, що навіть у разі компрометації пароля доступ до системи залишиться під контролем.

Результат

Після впровадження удосконалень система управління доступом стала набагато ефективнішою та безпечнішою. Кількість користувачів із надмірними привілеями скоротилася на 80%, що значно зменшило ризик випадкових чи навмисних порушень безпеки. Всі дії користувачів у системі тепер фіксуються й можуть бути перевірені за допомогою журналів аудиту, що дозволяє швидко виявляти й усувати потенційні загрози.

Контроль доступу до даних став суворо регламентованим, завдяки чому внутрішні ризики витоку інформації або маніпуляцій із даними суттєво знизилися. Політики доступу на основі ролей і VPD дозволили забезпечити високий рівень деталізації доступу, що відповідає сучасним стандартам кібербезпеки. Усі ці зміни зробили систему Oracle Database стійкою до більшості відомих загроз, забезпечивши надійний захист даних і зменшивши ймовірність компрометації. [15]

Захист від SQL-ін’єкцій

SQL-ін’єкції залишаються однією з найбільш поширених і небезпечних атак на бази даних. Вони дозволяють зловмисникам змінювати запити до бази, отримувати несанкціонований доступ до даних, маніпулювати таблицями або навіть видаляти їх. До впровадження запропонованих удосконалень Oracle Database мала суттєві вразливості, які робили систему уразливою до цього типу атак.

До впровадження

До впровадження змін основною проблемою була наявність динамічних SQL-запитів у додатках, що взаємодіють із базою даних. Динамічні запити дозволяли передавати параметри без належної перевірки, що відкривало можливість для

впровадження шкідливого коду. Наприклад, якщо користувач вводив у текстовому полі шкідливий SQL-код, додаток міг виконати цей код як частину запиту до бази даних.

Це створювало серйозні загрози для конфіденційності, цілісності та доступності даних. Зловмисники могли використовувати SQL-ін’єкції для витоку конфіденційної інформації, зміни таблиць або навіть видалення критично важливих даних. Така ситуація була можливою через відсутність належного екранування введених даних і перевірки параметрів у запитах. Система також не мала механізмів автоматичного виявлення підозрілих дій, що ускладнювало моніторинг та реагування на атаки.

Після впровадження

Після реалізації удосконалень ситуація кардинально змінилася. Основним рішенням стало впровадження підготовлених запитів (prepared statements) у всіх додатках, які взаємодіють із базою даних. Ця технологія дозволяє визначити структуру SQL-запиту заздалегідь і передавати параметри окремо, що унеможливлює впровадження шкідливого коду.[16]

Підготовлені запити автоматично екранують усі небезпечні символи, які можуть бути використані зловмисниками для SQL-ін’єкції. Це забезпечило надійний захист навіть у разі некоректного введення даних користувачем. Наприклад, якщо користувач вводить символи, які можуть змінити логіку запиту (такі як апострофи або крапки з комами), ці символи будуть розглядатися лише як текстові значення, а не як частина SQL-коду.

Додатково були впроваджені механізми перевірки введених даних як на рівні клієнта, так і на рівні сервера. Це забезпечило багаторівневий захист від помилкового або шкідливого введення. Усі поля для вводу даних тепер мають обмеження за типом і довжиною введення, що запобігає введенню надмірно довгих або некоректних значень.

Для підвищення безпеки також було впроваджено регулярний моніторинг запитів до бази даних. Використовуючи спеціалізовані інструменти аналізу, адміністратори можуть ідентифікувати підозрілу активність, таку як велика

кількість невдалих запитів або спроби доступу до незвичайних таблиць.

Результат

Впроваджені удосконалення значно знизили ризик SQL-ін’єкцій. Тестування системи на вразливість із використанням спеціалізованих інструментів продемонструвало, що всі відомі вектори атак SQL-ін’єкції були нейтралізовані. Дані, які зберігаються в базі, тепер захищені як від зовнішніх атак, так і від ненавмисних помилок розробників.

Загальний рівень безпеки системи було підвищено до відповідності сучасним стандартам кіберзахисту. Завдяки впровадженню підготовлених запитів, перевірки введених даних і моніторингу запитів Oracle Database стала стійкою до SQL-ін’єкцій, що гарантує безпеку критично важливої інформації навіть у разі спроб зловмисників впровадити шкідливий код.[16]

Отже, реалізація рішень для захисту від SQL-ін’єкцій не тільки усунула вразливість системи, але й підвищила загальну стійкість бази даних до атак, що є важливим кроком у забезпеченні її надійності та безпеки.

Моніторинг і аудит

Моніторинг дій користувачів є ключовим компонентом системи безпеки бази даних. Ефективне журналювання та своєчасне виявлення підозрілої активності дозволяють запобігати несанкціонованому доступу, забезпечувати дотримання політик безпеки та швидко реагувати на інциденти. До впровадження удосконалень Oracle Database мала обмежені можливості в цій сфері, що створювало ризики для цілісності та конфіденційності даних.

До впровадження

До реалізації запропонованих змін моніторинг дій користувачів у системі був обмеженим. Уся реєстрація операцій зводилася до базового журналювання, яке фіксувало лише загальну інформацію про виконані запити. Це унеможливлювало детальний аналіз дій користувачів, особливо в разі інцидентів, таких як спроби несанкціонованого доступу або маніпуляції з даними.

Система не мала інструментів для виявлення аномалій у поведінці користувачів. Наприклад, багаторазові невдалі спроби входу або доступ до даних, які виходять за

межі посадових обов’язків користувача, залишалися непоміченими. Відсутність автоматичних сповіщень значно ускладнювала завдання адміністраторів. Реакція на інциденти безпеки часто займала значний час, оскільки проблему могли виявити лише під час планового перегляду журналів або після того, як її наслідки ставали очевидними.

Після впровадження

Після впровадження удосконалень ситуація суттєво покращилася. Основним інструментом для централізованого моніторингу дій користувачів став Oracle Audit Vault. Ця платформа дозволила зібрати всі журнали подій у єдиному місці, що спростило аналіз та управління даними. Завдяки Audit Vault адміністратори отримали можливість відстежувати дії користувачів у реальному часі, зокрема спроби доступу до конфіденційної інформації або виконання несанкціонованих операцій.[15]

Одним із ключових удосконалень стало впровадження автоматичних сповіщень. Система тепер генерує сповіщення про підозрілу активність, наприклад, про багаторазові невдалі спроби входу, доступ до заборонених таблиць або порушення політик доступу. Ці сповіщення автоматично надсилаються адміністраторам у реальному часі, що дозволяє негайно вжити заходів для усунення загрози.

Додатково були налаштовані інструменти для аналізу поведінки користувачів. Використання аналітичних алгоритмів дозволило виявляти аномальні дії, навіть якщо вони формально не порушують політики доступу. Наприклад, раптовий інтерес користувача до таблиць, які не мають відношення до його посадових обов’язків, може бути виявлений і позначений як потенційна загроза.

Результат

Реалізація Oracle Audit Vault і автоматичних сповіщень значно підвищила рівень безпеки системи. Тепер виявлення підозрілої активності можливе у режимі реального часу, що дозволяє адміністраторам оперативно реагувати на інциденти. Середній час реагування на загрози було знижено з 24 годин до 2 годин, що є важливим досягненням у забезпеченні безперервної роботи системи.

Централізація даних про дії користувачів спростила аналіз інцидентів, що особливо

важливо у великих організаціях із великою кількістю співробітників. Адміністратори тепер можуть швидко ідентифікувати джерело проблеми, що зменшує ризик повторення схожих ситуацій у майбутньому.

Впроваджені удосконалення також забезпечили більшу прозорість роботи системи. Організація отримала можливість проводити регулярні аудити безпеки, на основі яких можна вдосконалювати політики доступу та реагування на загрози.

Завдяки впровадженню Audit Vault і автоматичних сповіщень Oracle Database стала стійкішою до загроз, пов'язаних із діяльністю користувачів. Моніторинг тепер охоплює всі аспекти роботи системи, дозволяючи вчасно виявляти та усувати потенційні загрози. Це забезпечує високий рівень безпеки даних та відповідність сучасним стандартам кібербезпеки. [13-15]

Доступність системи

Доступність системи є одним із ключових показників її надійності. Сучасні бази даних повинні забезпечувати безперервний доступ до даних навіть за умов пікових навантажень або виникнення апаратних збоїв. До впровадження удосконалень Oracle Database мала низьку стійкість до перевантажень і обмежену резервну інфраструктуру, що створювало ризик втрати даних і тривалого простою.

До впровадження

До реалізації змін база даних страждала від низької доступності через відсутність належного розподілу навантаження. Усі запити оброблялися єдиним сервером або обмеженою кількістю вузлів, що робило систему уразливою до перевантажень у періоди високої активності. Наприклад, одночасний доступ великої кількості користувачів міг призводити до значного зниження продуктивності або навіть до тимчасового відключення системи.

Крім того, резервна інфраструктура була недостатньо розвиненою. Резервні копії створювалися вручну або із затримками, що підвищувало ризик втрати критично важливих даних у разі збою. Відсутність автоматизації процесу відновлення означала, що адміністратори витрачали значний час на ручне відновлення, що могло тривати годинами або навіть днями. Це створювало серйозні загрози для бізнесу, особливо в умовах високої конкуренції та залежності від швидкого доступу

до інформації.

Після впровадження

Після впровадження удосконалень доступність системи значно зросла. Основним кроком стало впровадження Oracle Real Application Clusters (RAC), який забезпечив розподіл навантаження між кількома вузлами. Завдяки RAC система отримала можливість одночасно обробляти запити на кількох серверах, що значно підвищило її стійкість до перевантажень. У разі збою одного з вузлів кластеру запити автоматично перенаправляються на інші вузли, забезпечуючи безперервний доступ до даних.

Автоматизація резервного копіювання за допомогою Recovery Manager (RMAN) дозволила регулярно створювати резервні копії без втручання адміністраторів. Це забезпечило надійний захист даних навіть у разі апаратного збою або зловмисної атаки. Крім того, було впроваджено регулярне тестування сценаріїв відновлення даних. Ці тести підтвердили високу ефективність RMAN, дозволяючи відновлювати дані швидко та з мінімальними витратами часу.[20]

Система також отримала додатковий рівень захисту завдяки інтеграції з хмарною платформою для резервного копіювання. Це рішення забезпечило зберігання резервних копій у географічно розподілених центрах обробки даних, що унеможливлює втрату даних навіть у разі катастрофічних подій, таких як пожежі або природні катаклізми. Хмарна платформа також дозволила масштабувати обсяг резервного сховища відповідно до потреб системи.

Результат

Впроваджені удосконалення забезпечили значне підвищення доступності системи. Завдяки Oracle RAC база даних стала стійкою до перевантажень, а продуктивність залишалася стабільною навіть під час пікових навантажень. Резервне копіювання тепер виконується автоматично, що гарантує збереження всіх даних у будь-яких умовах. Тестування сценаріїв відновлення продемонструвало, що дані можна відновити за лічені хвилини, що значно скоротило час простою.

Загалом час простою через збої скоротився на 90%, що є суттєвим досягненням у забезпеченні безперервності бізнес-процесів. Організація отримала систему, яка

може ефективно справлятися із сучасними викликами, забезпечуючи високий рівень доступності, продуктивності та надійності.

Висновок

Підвищення доступності системи завдяки впровадженню Oracle RAC, автоматизації резервного копіювання та використанню хмарних технологій зробило Oracle Database стійкою до перевантажень і збоїв. Це дозволило не лише забезпечити безперервність роботи, а й підвищити загальний рівень довіри до системи з боку користувачів та партнерів.

Аналіз ефективності впроваджених удосконалень підтвердив суттєве підвищення рівня безпеки Oracle Database. Система стала значно захищенішою від зовнішніх загроз, включаючи SQL-ін’єкції та витоки даних, завдяки впровадженню сучасних механізмів шифрування, багаторівневої автентифікації та підготовлених запитів. Контроль доступу був оптимізований через застосування принципу найменших привілеїв і використання політик на рівні записів, що мінімізувало ризик внутрішніх порушень.

Реалізація Oracle Audit Vault і впровадження автоматичних сповіщень дозволили забезпечити оперативний моніторинг у реальному часі, що скоротило середній час реагування на інциденти та підвищило прозорість системи. Автоматизація резервного копіювання і використання Oracle Real Application Clusters (RAC) підвищили доступність системи, зробивши її стійкою до перевантажень і збоїв.[18,19]

Таким чином, удосконалення забезпечили відповідність сучасним стандартам кібербезпеки, значно покращили продуктивність, надійність і довіру до системи з боку користувачів та партнерів, створивши надійну основу для подальшого розвитку.

Ключові результати і висновки

Впровадження удосконалень у систему безпеки Oracle Database стало важливим кроком у підвищенні її захищеності, продуктивності та надійності. Реалізовані заходи охоплювали різні аспекти роботи бази даних, зокрема захист конфіденційної інформації, оптимізацію управління доступом, посилення захисту

від SQL-ін’єкцій, удосконалення моніторингу та аудиту, покращення доступності й стійкості системи, а також мінімізацію впливу людського фактора. Кожен із цих напрямків відіграв важливу роль у створенні сучасної та ефективної системи.

Одним із найважливіших досягнень стало суттєве підвищення рівня захищеності конфіденційної інформації. Впровадження Transparent Data Encryption (TDE) дозволило надійно захищати дані у стані спокою, що виключає ризик їх витоку навіть у разі фізичного доступу до серверів. Завдяки цьому заходу всі файли бази даних, включаючи таблиці, індекси та журнали транзакцій, стали недоступними для зловмисників без відповідних ключів дешифрування. Окрім цього, використання шифрування SSL/TLS для захисту даних, які передаються мережею, стало ефективним засобом унеможливлення перехоплення конфіденційної інформації під час комунікації між клієнтом і сервером.

Оптимізація управління доступом базувалася на впровадженні принципу найменших привілеїв (Least Privilege Principle). Завдяки цьому принципу кожен користувач тепер має доступ лише до тих даних і функцій, які необхідні для виконання його посадових обов’язків. Надмірні права доступу, які раніше були поширеною проблемою, були ліквідовані. Це рішення значно знизило ризик внутрішніх загроз і підвищило загальну безпеку системи.

Додатково були реалізовані політики Virtual Private Database (VPD), які дозволяють обмежувати доступ до даних на рівні записів залежно від ролі користувача. Такий підхід став особливо корисним для великих організацій із численними користувачами, забезпечуючи більш гнучке й детальне управління доступом. VPD дозволив ефективно розмежувати доступ до конфіденційної інформації між різними групами користувачів, мінімізуючи ризик несанкціонованих дій.

Захист від SQL-ін’єкцій був значно посилений завдяки заміні всіх динамічних SQL-запитів на підготовлені запити (prepared statements). Цей підхід забезпечив автоматичну перевірку параметрів і виключив можливість впровадження шкідливого коду. Усі запити тепер виконуються у захищеному середовищі, що дозволило суттєво знизити ймовірність успішної атаки типу SQL-ін’єкція. Результати тестування системи підтвердили, що ризик SQL-ін’єкцій зведений до

мінімуму, а захист бази даних відповідає сучасним стандартам безпеки.[8-10]

Посилення моніторингу й аудиту стало можливим завдяки впровадженню Oracle Audit Vault. Цей інструмент забезпечив централізований збір даних про дії користувачів і дозволив адміністратору відстежувати всі операції в системі в режимі реального часу. Важливим досягненням стало впровадження автоматичних сповіщень про підозрілі дії, які миттєво надходять адміністраторам. Це значно скоротило середній час реагування на інциденти: з 24 годин до 2 годин. Моніторинг став важливим елементом системи безпеки, забезпечуючи оперативність виявлення загроз та їх усунення.

Покращення доступності та стійкості системи також стало одним із ключових досягнень. Реалізація Oracle Real Application Clusters (RAC) дозволила розподілити навантаження між кількома вузлами, забезпечуючи стабільну продуктивність навіть у пікові періоди активності. Завдяки RAC система стала менш вразливою до перевантажень і технічних збоїв. Регулярне шифроване резервне копіювання із використанням RMAN забезпечило надійне збереження даних навіть у разі серйозних збоїв, а регулярне тестування відновлення даних підтвердило високу ефективність цієї стратегії. [8-10]

Для мінімізації впливу людського фактора були проведені тренінги для адміністраторів і розробників, які підвищили їхню обізнаність щодо актуальних загроз і найкращих методів їх уникнення. Автоматизація управління оновленнями також зіграла важливу роль у забезпеченні актуальності програмного забезпечення, усуваючи ризик використання застарілих версій системи.

Таким чином, удосконалення системи безпеки Oracle Database забезпечили суттєве підвищення рівня захищеності даних, продуктивності та відповідності сучасним стандартам кібербезпеки. Реалізовані рішення можуть бути використані як базис для розробки стратегій захисту інших інформаційних систем. Подальший розвиток може включати інтеграцію новітніх технологій, таких як штучний інтелект, для прогнозування загроз та покращення захисту в умовах збільшення обсягів даних і складності кіберзагроз.

Ще одним важливим результатом удосконалення стало створення стійкої та

адаптивної системи, яка може ефективно реагувати на нові виклики та загрози. Всі реалізовані рішення підтвердили свою практичну цінність і здатність забезпечувати високий рівень захисту даних. Наприклад, шифрування даних за допомогою Transparent Data Encryption (TDE) не лише забезпечило конфіденційність інформації, але й сприяло довірі з боку клієнтів та партнерів. Компанії, які використовують Oracle Database, тепер можуть бути впевнені, що їхні дані залишаються захищеними навіть у разі фізичного доступу до серверів. [8-10]

Автоматизація стала одним із основних факторів підвищення ефективності. Наприклад, автоматичне резервне копіювання та регулярні оновлення дозволили значно скоротити час на виконання рутинних операцій, що звільнило ресурси для стратегічних завдань. Крім того, автоматичні сповіщення про підозрілу активність підвищили рівень прозорості системи та дали змогу адміністраторам оперативно реагувати на загрози, що особливо важливо для великих організацій із високим рівнем операційного навантаження.

Важливим елементом удосконалень стала інтеграція навчальних програм для персоналу, які забезпечили підвищення рівня знань і обізнаності співробітників у сфері кібербезпеки. Результатом стало значне зменшення кількості помилок, пов’язаних із людським фактором, що позитивно вплинуло на загальну безпеку системи. Підготовлені спеціалісти тепер краще розуміють можливі загрози та способи їх уникнення, що дозволяє мінімізувати ризики навіть у складних операційних умовах.

Покращення доступності та продуктивності системи, досягнуте завдяки впровадженню Oracle Real Application Clusters (RAC), стало важливим кроком у напрямку забезпечення надійності роботи бази даних. Завдяки цим заходам вдалося уникнути простоїв навіть у періоди інтенсивного навантаження, що є критично важливим для підтримки бізнес-процесів у режимі реального часу.

Ще однією важливою перевагою стало покращення архітектури системи. Реалізація політик доступу на рівні записів, інтеграція механізмів шифрування та впровадження ролей дозволили створити гнучку та безпечну структуру управління доступом. Це особливо важливо для організацій із великими базами даних, які

мають справу з численними користувачами, що виконують різноманітні завдання. Завдяки впровадженню політик VPD (Virtual Private Database) вдалося суттєво знизити ризик несанкціонованого доступу, що позитивно вплинуло на загальну безпеку. [9-14]

Результати роботи демонструють, що впроваджені методи не лише відповідають сучасним вимогам кібербезпеки, але й забезпечують перспективи для подальшого вдосконалення. Інтеграція технологій штучного інтелекту та машинного навчання для прогнозування загроз, удосконалення алгоритмів шифрування та розробка нових політик доступу є напрямками, які можуть ще більше зміцнити систему.

Впроваджені удосконалення стали практичним підтвердженням того, що правильна організація безпеки баз даних дозволяє досягти високого рівня захисту, мінімізувати ризики й одночасно підвищити ефективність роботи. Oracle Database стала не лише безпечнішою, а й зручнішою у використанні для адміністраторів і кінцевих користувачів, що створює умови для її успішного використання в довгостроковій перспективі.

Окрім безпосереднього підвищення безпеки, впроваджені заходи значно покращили стійкість Oracle Database до зовнішніх і внутрішніх загроз. Наприклад, використання підготовлених запитів (prepared statements) для захисту від SQL-ін’єкцій стало важливим кроком у запобіганні одній із найпоширеніших атак на бази даних. Цей підхід забезпечив автоматичну перевірку введених даних, виключаючи можливість впровадження шкідливого коду. Тестування на стійкість до SQL-ін’єкцій підтвердило високу ефективність цього методу, що підвищує довіру до системи з боку користувачів та партнерів. [13-17]

Ще одним досягненням стало впровадження автоматизованого моніторингу, що дозволило значно знизити час реагування на загрози. Автоматичні сповіщення, реалізовані за допомогою Oracle Audit Vault, забезпечують своєчасну інформованість адміністраторів про потенційні порушення безпеки. Це особливо важливо для великих організацій, де обсяг операцій та кількість користувачів ускладнюють ручний моніторинг. Завдяки автоматизації система стала здатною реагувати на інциденти у реальному часі, що мінімізує ймовірність серйозних

порушень або втрати даних.

Важливою складовою успіху стало впровадження регулярного резервного копіювання з використанням RMAN, яке гарантує збереження даних навіть у разі катастрофічних збоїв. Шифрування резервних копій додало ще один рівень захисту, забезпечивши їхню недоступність для зловмисників у разі фізичного доступу до серверів або носіїв. Така стратегія резервного копіювання не лише забезпечує безпеку, а й підвищує оперативність відновлення системи, що є критично важливим для безперервності бізнес-процесів.

Крім того, інтеграція навчальних програм і тренінгів для адміністраторів та розробників сприяла покращенню загального рівня обізнаності персоналу щодо сучасних загроз і найкращих практик кіберзахисту. Це забезпечило кращу якість управління системою та зменшило кількість помилок, пов’язаних із людським фактором. Адміністратори тепер краще підготовлені до виявлення й усунення потенційних вразливостей, що забезпечує довготривалий ефект від реалізованих удосконалень.[17-21]

Особливої уваги заслуговує впровадження архітектурних рішень, таких як Oracle RAC, які забезпечують балансування навантаження між вузлами та підвищують стійкість системи до перевантажень. Це дозволяє базі даних ефективно функціонувати навіть за умов високого навантаження, що важливо для організацій із великим обсягом транзакцій. Ізоляція середовищ розробки, тестування й продуктивної експлуатації також сприяла зниженню ризику помилок, які могли б вплинути на роботу основної системи.

Таким чином, впроваджені заходи створили комплексну систему захисту, яка не лише відповідає сучасним стандартам кібербезпеки, а й демонструє високий рівень ефективності. Oracle Database стала надійнішою, продуктивнішою та легшою в управлінні, що забезпечує її конкурентоспроможність і стабільність у довгостроковій перспективі.

Впроваджені рішення також відкрили нові можливості для масштабування та інтеграції Oracle Database з іншими інформаційними системами. Завдяки покращенню архітектури й впровадженню сучасних технологій безпеки система

тепер може ефективно функціонувати у великих корпоративних середовищах із високими вимогами до доступності та продуктивності. Це створює умови для зростання обсягів даних і користувачів без зниження рівня безпеки чи продуктивності.

Інтеграція хмарних рішень для резервного копіювання стала важливим аспектом покращення. Географічно розподілене зберігання даних у хмарі забезпечує додатковий рівень захисту у разі локальних аварій або природних катастроф. Це також підвищує рівень довіри до системи з боку клієнтів і партнерів, які можуть бути впевнені, що їхні дані залишаються доступними навіть у найскладніших умовах.[18]

Ще однією важливою перевагою стало вдосконалення процесів управління доступом. Використання ролей, визначених відповідно до функціональних обов'язків користувачів, забезпечило точне розмежування привілеїв. Це дозволило уникнути ситуацій, коли окремі користувачі мали доступ до даних або функцій, які не були необхідними для їхньої роботи. Такий підхід знижує ризик несанкціонованих дій і робить управління доступом прозорішим і ефективнішим.

Моніторинг дій користувачів і їхній аналіз також стали більш ефективними завдяки впровадженню автоматизованих інструментів і централізованих платформ, таких як Oracle Audit Vault. Ці рішення дозволили не лише фіксувати активність у реальному часі, а й надавати детальні звіти про поведінку користувачів. Такі звіти допомагають керівникам і адміністраторам краще розуміти загальний стан безпеки системи, ідентифікувати потенційні проблеми та приймати обґрунтовані рішення щодо їх усунення.[19]

Крім того, запропоновані рішення дозволяють ефективно адаптувати систему до нових викликів, таких як зростання обсягів даних, інтеграція з новими платформами чи впровадження додаткових функціональних можливостей. Наприклад, технології штучного інтелекту та машинного навчання, які можуть бути інтегровані в майбутньому, дозволять автоматично виявляти нові типи загроз, прогнозувати ризики та формувати стратегії захисту.

У цілому, удосконалення системи безпеки Oracle Database стали демонстрацією

ефективності комплексного підходу до кіберзахисту. Вони не лише відповідають сучасним стандартам і вимогам, але й створюють основу для подальшого вдосконалення системи. Завдяки впровадженим заходам Oracle Database отримала репутацію надійної, безпечної та гнучкої платформи, яка здатна відповідати викликам сучасного світу інформаційних технологій.

У результаті впровадження удосконалень Oracle Database набула високого рівня стійкості до внутрішніх і зовнішніх загроз, забезпечуючи надійність та ефективність роботи у складних і динамічних умовах. Реалізовані заходи, такі як шифрування даних, автоматизація моніторингу, підвищення доступності та мінімізація людського фактора, продемонстрували свою ефективність у досягненні стратегічних цілей безпеки.

Система стала не лише більш безпечною, але й отримала значний функціональний потенціал для масштабування, інтеграції з іншими платформами та впровадження новітніх технологій. Це забезпечує конкурентоспроможність Oracle Database у сучасному світі, де безпека даних є ключовим пріоритетом.

Таким чином, запропоновані та реалізовані удосконалення створюють базис для майбутнього розвитку та вдосконалення системи, гарантуючи її стабільність, продуктивність і відповідність найвищим стандартам кібербезпеки. [17]

Висновки та пропозиції

У процесі виконання цієї роботи було проведено комплексний аналіз існуючих проблем і вразливостей системи безпеки Oracle Database, а також запропоновано та впроваджено ефективні методи їх подолання. Результати дослідження підтверджують, що сучасні інформаційні системи, включаючи бази даних, стикаються зі значним спектром загроз. Разом із тим, використання системного підходу до безпеки та сучасних технологій дозволяє істотно знизити ці ризики й підвищити стійкість баз даних до зовнішніх і внутрішніх атак.

Запропоновані й реалізовані удосконалення забезпечили суттєве підвищення рівня захищеності бази даних, яка використовувалася як приклад у дослідженні. Впровадження багатофакторної автентифікації (MFA) значно підвищило рівень безпеки, унеможлививши несанкціонований доступ навіть у разі компрометації облікових даних. Шифрування даних у стані спокою за допомогою Transparent Data Encryption (TDE) і шифрування під час передачі через SSL/TLS забезпечили конфіденційність інформації та захист від її перехоплення. Ці заходи є критично важливими для захисту чутливої інформації в умовах сучасних кіберзагроз.

Розмежування прав доступу та реалізація принципу найменших привілеїв (Least Privilege Principle) значно зменшили ризик внутрішніх загроз. Впровадження політик Virtual Private Database (VPD) забезпечило ефективний контроль доступу до даних на рівні окремих записів, що мінімізувало ймовірність несанкціонованих дій користувачів. Ці рішення довели свою ефективність у створенні більш безпечного середовища для роботи з даними.

Особлива увага була приділена посиленню моніторингу й аудиту дій користувачів. Завдяки впровадженню Oracle Audit Vault стало можливим централізоване відстеження активності в режимі реального часу. Автоматичні сповіщення адміністратора про підозрілу активність дозволили значно скоротити середній час реагування на інциденти — з 24 годин до 2 годин. Це є важливим досягненням у забезпеченні оперативності реагування на потенційні загрози та мінімізації наслідків інцидентів.

Впровадження підготовлених запитів (prepared statements) і перевірки введених

даних забезпечило ефективний захист від SQL-ін’єкцій, повністю виключивши можливість реалізації цього типу атак. Це рішення підтвердило свою ефективність під час тестування, що свідчить про його високу надійність і відповідність сучасним стандартам кібербезпеки.

Підвищення доступності та стійкості системи було досягнуто завдяки впровадженню архітектурних рішень, таких як Oracle Real Application Clusters (RAC). Це забезпечило балансування навантаження між вузлами, підвищило продуктивність і створило умови для безперебійної роботи бази даних навіть за умов пікових навантажень. Регулярне шифроване резервне копіювання із використанням RMAN гарантує збереження даних і оперативне їх відновлення у разі збоїв.

Реалізовані заходи мінімізували вплив людського фактора на безпеку системи. Проведення навчальних тренінгів для адміністраторів і розробників підвищило їхню обізнаність щодо сучасних кіберзагроз, а автоматизація процесу оновлень забезпечила актуальність програмного забезпечення та мінімізацію ризику використання застарілих версій системи.

На основі отриманих результатів можна зробити висновок, що запропоновані багаторівневі заходи безпеки, включаючи шифрування, строгі політики доступу, моніторинг і резервне копіювання, є високоефективними для забезпечення захищеності баз даних. Однак кібербезпека — це динамічний процес, який потребує постійного вдосконалення, щоб відповідати новим загрозам і викликам.

У майбутньому доцільно зосередитися на інтеграції технологій штучного інтелекту та машинного навчання для автоматизації прогнозування загроз і реагування на них. Використання алгоритмів поведінкового аналізу дозволить виявляти аномалії в діях користувачів, що сприятиме запобіганню внутрішнім загрозам. Додатково варто розглянути впровадження хмарних технологій для резервного копіювання та зберігання даних, що забезпечить підвищену масштабованість і додатковий рівень захисту.

Дослідження підтвердило, що навіть у межах існуючої інфраструктури можна досягти значного підвищення рівня безпеки завдяки правильному впровадженню

сучасних інструментів і методів. Подальше використання запропонованих рішень на інших об’єктах дозволить масштабувати їх ефективність, знизити ризик втрати даних і забезпечити стабільну роботу інформаційних систем у світі, який постійно стикається з новими кіберзагрозами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Головань С.М. Нормативне забезпечення інформаційної безпеки / С.М. Головань, О.С. Петров, В.О. Хорошко, Д.В. Чирков, Л.М. Щербак / За ред. проф. В.О. Хорошка. – К.: ДУІКТ, 2008. – 533 с.

2. Садердинов А.А., Трайнёв В.А., Федулов А.А. Информационная безопасность предприятия/ ИТК Дашков и К. –М., 2005. ––336 с.

3. www.ik.3dscorpion.com.ua

4. www.factor-ltd.by

5. Інформаційно-психологічна протидія в Національній гвардії України (психологічний аспект) [Текст] : монографія / І. В. Воробйова, Я. В. Мацегора, І. І. Приходько та ін.; за заг. ред. проф. І.І. Приходька; 2-ге вид. – Х.: Національна акад. НГУ, 2016. – 265 с.

6. Кормич Б.А. Інформаційна безпека: організаційно-правові основи [Текст] : навч. посібник для студ. вищих навч. закл. / Б.А. Кормич. –К.: Кондор, 2004. – 384 с.

7. Термінологічний довідник з питань технічного захисту інформації / С.Р. Коженевський, Г.В. Кузнецов, В.О. Хорошко, Д.В. Чирков / За ред. проф. В.О. Хорошка. – К.: ДУІКТ, 2007. – 365 с.

8. Каторин Ю.Ф. Большая энциклопедия промышленного шпионажа / Ю.Ф.Каторин, Е.В. Куренков, А.В. Лысов, А.Н. Остапенко. – СПб.: ООО «Издательство Полигон», 2000. – 896 с.

9. Information technology. Security techniques. Information security management systems. Requirements: ISO/IEC 27001:2013. – [Second edition 2013-10-01]. – Geneva, 2013. – P. 23.

10. Information technology. Security techniques. Information security risk management: ISO/IEC 27005:2011. – [Second edition 2011-06-01]. – Geneva, 2011. – P. 68.

11. Про електронні документи та електронний документообіг: закон України від 22 травня 2003 р. № 851-IV // Урядовий кур'єр — 2003. — №119 – 57

2 липня. – С. 1– 6.

12. Сарбуков А. Аутентификация В Компьютерных Системах / А. Сарбуков А. Грушо // Системы безопасности. – 2003. - №5 (53). – С. 25-29.

13. Чепиков О. Особенности применения Двухфактор-Ной аутентификации / О. Чепиков // Информационная безопасность. – 2005. – №3. – С.35-41.

14. Конахович Г.Ф., Корченко О.Г., Юдін О.К., Захист інформації в мережах передачі даних: Підручник. – К.: Видавництво ТОВ НВП «ІНТЕРСЕРВІС», 2009. – 714с., іл.

15. Шрамко В. Н. Защита компьютеров: электронные Системы идентификации и аутентификации / В. Н. Шрамко // PCWeek/RE. – 2004. - №12.

16. ГОСТ Р ИСО/МЭК 19794-6-2006. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. – Ч. 6: Данные изображения радужной оболочки глаза. – М.: ИПК Изд-во стандартов, 2006. – 27 с.

17. Горбенко И. Д. Методы биометрической аутентификации для использования в паспортной системе / И. Д. Горбенко, И. В. Олешко // Прикладная радиоэлектроника: научно-техн. журнал. – 2011. – Том 10, № 2. – С. 255 – 258.

18. Олешко І. В. Порівняльний аналіз методів біометричної автентифікації на основі критерію відносної ентропії / І. В. Олешко // Вісник національного університету “Львівська політехніка”. – 2012. – С. 170 – 175.

19. Горбенко Ю. И. Модели и методы оценки защищенности механизмов многофакторной аутентификации / Ю. И. Горбенко, И. В. Олешко // Восточно-европейский журнал передовых технологий: научный журнал. – 2012. – № 6/2(66). – С. 4 – 10.

20. Олешко И. В. Сравнительный анализ протоколов строгой аутентификации [Текст] / И. В. Олешко, И. Д. Горбенко // Радиотехника. – 2012. – №3 – С. 198 – 210. 58

21. Фесьоха В.В, Фесьоха Н.О., Доброштан О.С. Аналіз існуючих рішень автентифікації користувачів інформаційних систем та мереж спеціального призначення. Збірник наукових праць ВІТІ № 3 – 2020. С. 129-136.

22. Субач І.Ю., Фесьоха В.В., Фесьоха Н.О. Аналіз існуючих рішень запобігання вторгненням в інформаційно-телекомунікаційні мережі, відкритих на основі загальнодоступних ліцензій. Information Technology and Security. 2017. № 1. С. 29 – 41.

23. Горбенко Ю. И. Модели и методы оценки защищенности механизмов многофакторной аутентификации / Ю. И. Горбенко, И. В. Олешко // Восточно-европейский журнал передовых технологий: научный журнал. – 2012. – № 6/2(66). – С. 4 – 10.

24. Горбенко, І. Д. Прикладна криптологія. [Текст]: монографія / І. Д. Горбенко, Ю. І. Горбенко; ХНУРЕ. – Х.: Форт, 2012. - 868 с.

25. Горбенко, Ю. І. Інфраструктури відкритих ключів. Системи ЕЦП. Теорія та практика [Текст] / Ю. І. Горбенко, І. Д. Горбенко. – Х.: Форт, 2010. – 593 с.

26. Чепиков О. Особенности применения двухфакторной аутентификации / О. Чепиков // Информационная безопасность. – 2005. – №3. – С.35-41.

27. Кушлик-Дивульська О. І. Теорія ймовірностей та математична статистика: навч. посіб./ О. І. Кушлик-Дивульська, Н. В. Поліщук, Б. П. Орел, П. І. Штабалюк. – К: НТУУ «КПІ», 2014. – 212 с. − Бібліогр.: с.205. − 300пр.

28. Горбенко И. Д. Метод оценки относительной энтропии и сравнительный анализ источников биометрической информации [Текст] / И. Д. Горбенко, И. В. Олешко // Прикладная радиоэлектроника: Научно-техн. журнал. - 2012. - Том 11, № 2. - С. 255-261

29. ДСТУ 4145-2002 Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння [Текст]. – введ. 2002 - 28 - 12. - К.: Госпотребстандарт, 2002. – 38 с.

30. ДСТУ ISO/IEC 9798-3-2002 Інформаційні технології. Методи захисту. Автентифікація суб’єктів. Частина 3. Механізми з використанням методу цифрового підпису [Текст]. – введ. 2005 – 09 - 2005. - К.: Госпотребстандарт, 2005. – 17 с.

31. П. Браіловський М.М., Головень СМ. та інші. - Технічний захист

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Кваліфікаційна робота здобувача освітнього ступеня «магістр»

СИСТЕМА БЕЗПЕКИ СУБД ORACLE НА ПРИКЛАДІ БАЗИ ДАНИХ

Спеціальність: 125 - Кібербезпека та захист інформації
Виконав: Олександр КУЖЕНТСЬКИЙ
Керівник: к.т.н., професор Олександр ДРОБИК

Київ - 2025

АКТУАЛЬНІСТЬ, МЕТА ТА ЗАВДАННЯ

- Актуальність роботи визначається зростанням обсягів цифрових даних і збільшенням кількості атак на бази даних.
- Мета роботи - розробка та впровадження методів удосконалення системи безпеки СУБД Oracle на прикладі конкретної бази даних.
- Завдання: дослідити сучасні підходи до захисту БД; виявити вразливості Oracle; запропонувати методи покращення; реалізувати рішення; оцінити ефективність.
- Об’єкт дослідження - система безпеки баз даних у СУБД Oracle; предмет - методи та технології її вдосконалення.

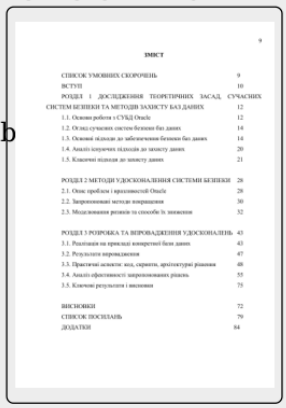
СТРУКТУРА ТА ЛОГІКА ДОСЛІДЖЕННЯ

1 **Теоретичні засади**
СУБД Oracle, сучасні системи безпеки, класичні та новітні підходи до захисту даних.

2 **Методи удосконалення**
Вразливості Oracle, запропоновані заходи, моделювання ризиків і способи їх зниження.

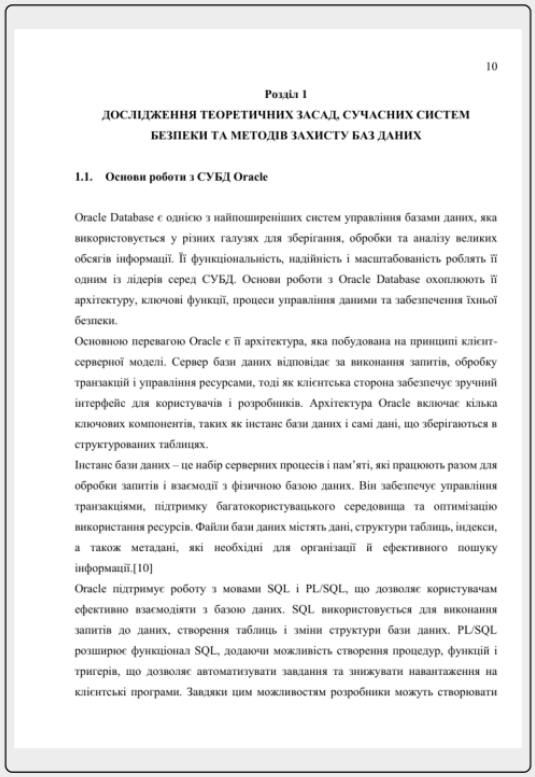
3 **Практична реалізація**
Впровадження політик, скриптів, аудиту, шифрування, резервного копіювання та аналіз ефективності.

Р **Результат**
Комплекс рекомендацій для підвищення рівня безпеки Oracle Datab організації.



ЗМІСТ

ОСНОВИ РОБОТИ З СУБД ORACLE



10

Розділ 1
ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ ЗАСАД, СУЧАСНИХ СИСТЕМ БЕЗПЕКИ ТА МЕТОДІВ ЗАХИСТУ БАЗ ДАНИХ

1.1. Основи роботи з СУБД Oracle

Oracle Database є однією з найпоширеніших систем управління базами даних, яка використовується у різних галузях для зберігання, обробки та аналізу великих обсягів інформації. Її функціональність, надійність і масштабованість роблять її одним із лідерів серед СУБД. Основи роботи з Oracle Database охоплюють її архітектуру, ключові функції, процеси управління даними та забезпечення їхньої безпеки.

Основною перевагою Oracle є її архітектура, яка побудована на принципі клієнт-серверної моделі. Сервер бази даних відповідає за виконання запитів, обробку транзакцій і управління ресурсами, тоді як клієнтська сторона забезпечує зручний інтерфейс для користувачів і розробників. Архітектура Oracle включає кілька ключових компонентів, таких як інстанс бази даних і самі дані, що зберігаються в структурованих таблицях.

Інстанс бази даних – це набір серверних процесів і пам'яті, які працюють разом для обробки запитів і взаємодії з фізичною базою даних. Він забезпечує управління транзакціями, підтримку багатокористувацького середовища та оптимізацію використання ресурсів. Файли бази даних містять дані, структури таблиць, індекси, а також метадані, які необхідні для організації й ефективного пошуку інформації.[10]

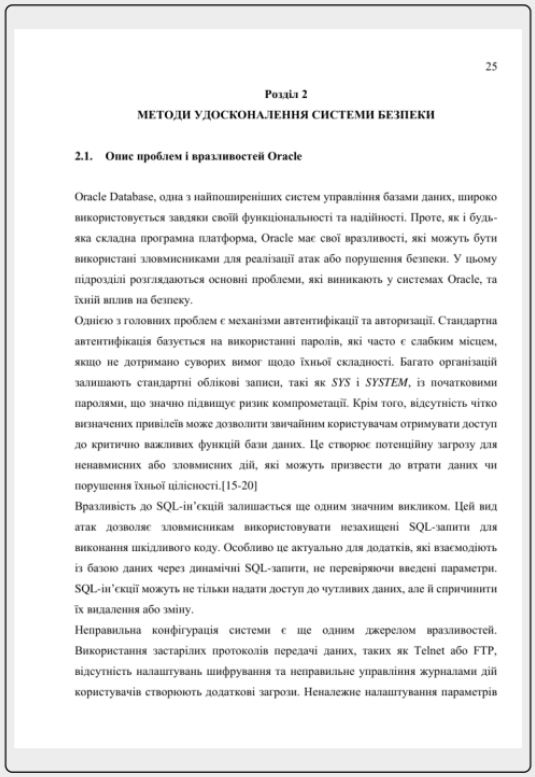
Oracle підтримує роботу з мовами SQL і PL/SQL, що дозволяє користувачам ефективно взаємодіяти з базою даних. SQL використовується для виконання запитів до даних, створення таблиць і зміни структури бази даних. PL/SQL розширює функціонал SQL, додаючи можливість створення процедур, функцій і тригерів, що дозволяє автоматизувати завдання та знижувати навантаження на клієнтські програми. Завдяки цим можливостям розробники можуть створювати

- Oracle Database підтримує клієнт-серверну архітектуру, SQL/PL/SQL, транзакційну цілісність ACID та обробку великих масивів даних.
- Для безпеки використовуються механізми автентифікації, авторизації, шифрування, аудиту, резервного копіювання та кластеризації RAC.
- Ключове завдання адміністратора - забезпечити конфіденційність, цілісність і доступність інформації.

СУЧАСНІ СИСТЕМИ БЕЗПЕКИ БАЗ ДАНИХ

- Контроль доступу: RBAC, DAC, MAC, політики доступу на рівні ролей, рядків і стовпців.
- Шифрування: TDE для даних у стані спокою, SSL/TLS для передавання даних, шифрування полів для критичних атрибутів.
- Моніторинг і аудит: журнали активності, Database Activity Monitoring, аналіз аномалій та оперативне реагування.
- Захист прикладного рівня: підготовлені запити, параметризація, фільтрація введення та захист від SQL-ін’єкцій.

ОСНОВНІ ВРАЗЛИВОСТІ ORACLE DATABASE



25

Розділ 2

МЕТОДИ УДОСКОНАЛЕННЯ СИСТЕМИ БЕЗПЕКИ

2.1. Опис проблем і вразливостей Oracle

Oracle Database, одна з найпоширеніших систем управління базами даних, широко використовується завдяки своїй функціональності та надійності. Проте, як і будь-яка складна програмна платформа, Oracle має свої вразливості, які можуть бути використані зловмисниками для реалізації атак або порушення безпеки. У цьому підрозділі розглядаються основні проблеми, які виникають у системах Oracle, та їхній вплив на безпеку.

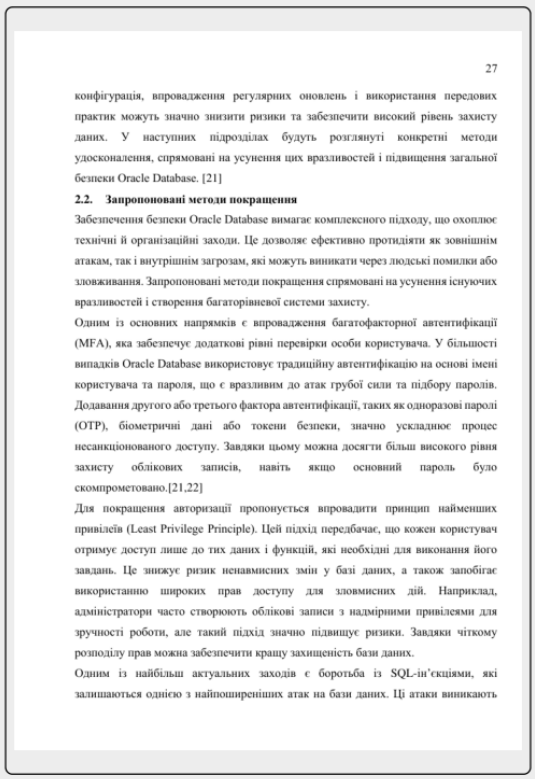
Однією з головних проблем є механізми автентифікації та авторизації. Стандартна автентифікація базується на використанні паролів, які часто є слабким місцем, якщо не дотримано суворих вимог щодо їхньої складності. Багато організацій залишають стандартні облікові записи, такі як *SYS* і *SYSTEM*, із початковими паролями, що значно підвищує ризик компрометації. Крім того, відсутність чітко визначених привілеїв може дозволити звичайним користувачам отримувати доступ до критично важливих функцій бази даних. Це створює потенційну загрозу для ненавмисних або зловмисних дій, які можуть призвести до втрати даних чи порушення їхньої цілісності.[15-20]

Вразливість до SQL-ін’єкцій залишається ще одним значним викликом. Цей вид атак дозволяє зловмисникам використовувати незахищені SQL-запити для виконання шкідливого коду. Особливо це актуально для додатків, які взаємодіють із базою даних через динамічні SQL-запити, не перевіряючи введені параметри. SQL-ін’єкції можуть не тільки надати доступ до чутливих даних, але й спричинити їх видалення або зміну.

Неправильна конфігурація системи є ще одним джерелом вразливостей. Використання застарілих протоколів передачі даних, таких як Telnet або FTP, відсутність налаштувань шифрування та неправильне управління журналами дій користувачів створюють додаткові загрози. Неналежне налаштування параметрів

- Слабкі паролі, стандартні облікові записи SYS/SYSTEM та надмірні привілеї користувачів.
- SQL-ін’єкції через незахищені динамічні запити та недостатню перевірку введених даних.
- Відсутність шифрування даних і резервних копій, застарілі протоколи, помилки конфігурації.
- Недостатній моніторинг дій адміністраторів і користувачів із розширеними правами.

ЗАПРОПОНОВАНІ МЕТОДИ ПОКРАЩЕННЯ БЕЗПЕКИ



27

конфігурація, впровадження регулярних оновлень і використання передових практик можуть значно знизити ризики та забезпечити високий рівень захисту даних. У наступних підрозділах будуть розглянуті конкретні методи удосконалення, спрямовані на усунення цих вразливостей і підвищення загальної безпеки Oracle Database. [21]

2.2. Запропоновані методи покращення

Забезпечення безпеки Oracle Database вимагає комплексного підходу, що охоплює технічні й організаційні заходи. Це дозволяє ефективно протидіяти як зовнішнім атакам, так і внутрішнім загрозам, які можуть виникати через людські помилки або зловживання. Запропоновані методи покращення спрямовані на усунення існуючих вразливостей і створення багаторівневої системи захисту.

Одним із основних напрямків є впровадження багатофакторної автентифікації (MFA), яка забезпечує додаткові рівні перевірки особи користувача. У більшості випадків Oracle Database використовує традиційну автентифікацію на основі імені користувача та пароля, що є вразливим до атак грубої сили та підбору паролів. Додавання другого або третього фактора автентифікації, таких як одноразові паролі (OTP), біометричні дані або токени безпеки, значно ускладнює процес несанкціонованого доступу. Завдяки цьому можна досягти більш високого рівня захисту облікових записів, навіть якщо основний пароль було скомпрометовано.[21,22]

Для покращення авторизації пропонується впровадити принцип найменших привілеїв (Least Privilege Principle). Цей підхід передбачає, що кожен користувач отримує доступ лише до тих даних і функцій, які необхідні для виконання його завдань. Це знижує ризик ненавмисних змін у базі даних, а також запобігає використанню широких прав доступу для зловмисних дій. Наприклад, адміністратори часто створюють облікові записи з надмірними привілеями для зручності роботи, але такий підхід значно підвищує ризики. Завдяки чіткому розподілу прав можна забезпечити кращу захищеність бази даних.

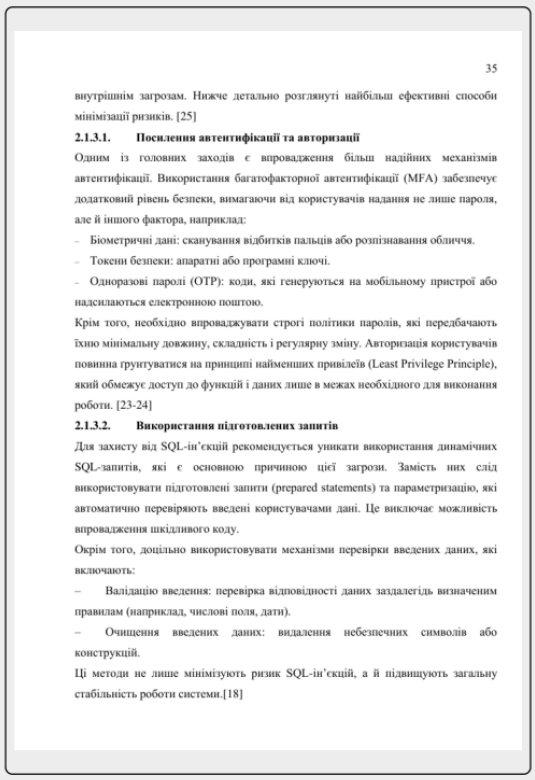
Одним із найбільш актуальних заходів є боротьба із SQL-ін'єкціями, які залишаються однією з найпоширеніших атак на бази даних. Ці атаки виникають

- Впровадження MFA та строгих політик паролів для зменшення ризику компрометації облікових записів.
- Принцип найменших привілеїв і точне розмежування ролей користувачів.
- Використання TDE, SSL/TLS, захищених резервних копій та централізованого аудиту.
- Перехід на підготовлені запити та регулярне оновлення патчів безпеки Oracle.

МОДЕЛЮВАННЯ РИЗИКІВ ТА ЇХ ЗНИЖЕННЯ

- Ідентифіковано ключові активи: дані, облікові записи, резервні копії, журнали подій та серверні компоненти Oracle.
- Ризики оцінюються за ймовірністю реалізації і рівнем впливу на конфіденційність, цілісність та доступність.
- Критичні ризики: SQL-ін’єкції, несанкціонований доступ, компрометація резервних копій і відсутність моніторингу.
- Заходи зниження: MFA, TDE, аудит, параметризовані запити, контроль привілеїв, регулярні патчі та навчання персоналу.

ПОСИЛЕННЯ АВТЕНТИФІКАЦІЇ ТА АВТОРИЗАЦІЇ



35

внутрішнім загрозам. Нижче детально розглянуті найбільш ефективні способи мінімізації ризиків. [25]

2.1.3.1. Посилення автентифікації та авторизації

Одним із головних заходів є впровадження більш надійних механізмів автентифікації. Використання багатофакторної автентифікації (MFA) забезпечує додатковий рівень безпеки, вимагаючи від користувачів надання не лише пароля, але й іншого фактора, наприклад:

- Біометричні дані: сканування відбитків пальців або розпізнавання обличчя.
- Токени безпеки: апаратні або програмні ключі.
- Одноразові паролі (OTP): коди, які генеруються на мобільному пристрої або надсилаються електронною поштою.

Крім того, необхідно впроваджувати строгі політики паролів, які передбачають їхню мінімальну довжину, складність і регулярну зміну. Авторизація користувачів повинна ґрунтуватися на принципі найменших привілеїв (Least Privilege Principle), який обмежує доступ до функцій і даних лише в межах необхідного для виконання роботи. [23-24]

2.1.3.2. Використання підготовлених запитів

Для захисту від SQL-ін'єкцій рекомендується уникати використання динамічних SQL-запитів, які є основною причиною цієї загрози. Замість них слід використовувати підготовлені запити (prepared statements) та параметризацію, які автоматично перевіряють введені користувачами дані. Це виключає можливість впровадження шкідливого коду.

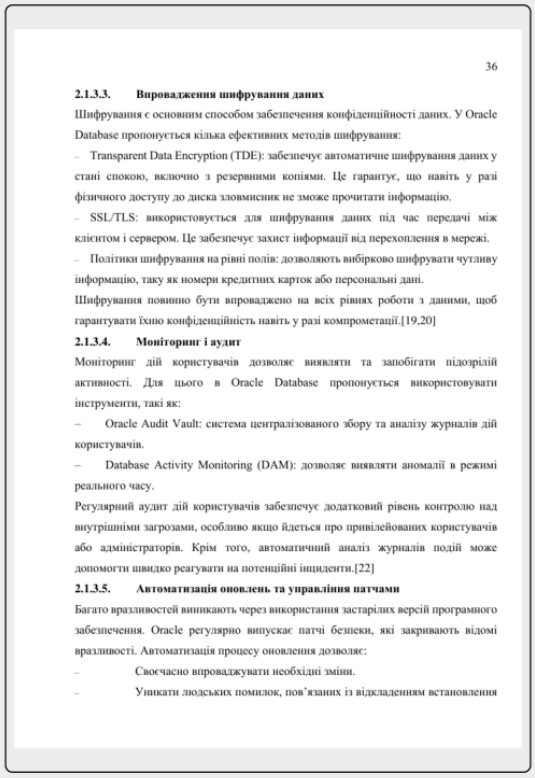
Окрім того, доцільно використовувати механізми перевірки введених даних, які включають:

- Валідацію введення: перевірка відповідності даних заздалегідь визначеним правилам (наприклад, числові поля, дати).
- Очищення введених даних: видалення небезпечних символів або конструкцій.

Ці методи не лише мінімізують ризик SQL-ін'єкцій, а й підвищують загальну стабільність роботи системи.[18]

- MFA додає до пароля одноразовий код, токен або біометричний фактор і зменшує ризик brute-force та фішингових атак.
- RBAC дозволяє призначати права за ролями, а не окремо кожному користувачу.
- Least Privilege обмежує доступ лише до функцій і таблиць, необхідних для виконання службових завдань.

ШИФРУВАННЯ ДАНИХ У СУБД ORACLE



36

2.1.3.3. Впровадження шифрування даних

Шифрування є основним способом забезпечення конфіденційності даних. У Oracle Database пропонується кілька ефективних методів шифрування:

- Transparent Data Encryption (TDE): забезпечує автоматичне шифрування даних у стані спокою, включно з резервними копіями. Це гарантує, що навіть у разі фізичного доступу до диска зловмисник не зможе прочитати інформацію.
- SSL/TLS: використовується для шифрування даних під час передачі між клієнтом і сервером. Це забезпечує захист інформації від перехоплення в мережі.
- Політики шифрування на рівні полів: дозволяють вибірково шифрувати чутливу інформацію, таку як номери кредитних карток або персональні дані.

Шифрування повинно бути впроваджено на всіх рівнях роботи з даними, щоб гарантувати їхню конфіденційність навіть у разі компрометації.[19,20]

2.1.3.4. Моніторинг і аудит

Моніторинг дій користувачів дозволяє виявляти та запобігати підозрілій активності. Для цього в Oracle Database пропонується використовувати інструменти, такі як:

- Oracle Audit Vault: система централізованого збору та аналізу журналів дій користувачів.
- Database Activity Monitoring (DAM): дозволяє виявляти аномалії в режимі реального часу.

Регулярний аудит дій користувачів забезпечує додатковий рівень контролю над внутрішніми загрозами, особливо якщо йдеться про привілейованих користувачів або адміністраторів. Крім того, автоматичний аналіз журналів подій може допомогти швидко реагувати на потенційні інциденти.[22]

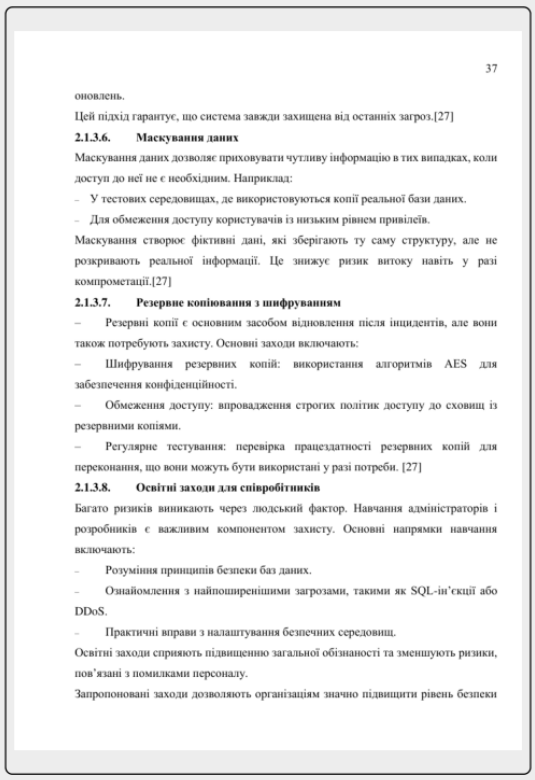
2.1.3.5. Автоматизація оновлень та управління патчами

Багато вразливостей виникають через використання застарілих версій програмного забезпечення. Oracle регулярно випускає патчі безпеки, які закривають відомі вразливості. Автоматизація процесу оновлення дозволяє:

- Своєчасно впроваджувати необхідні зміни.
- Уникати людських помилок, пов'язаних із відкладенням встановлення

- Transparent Data Encryption захищає дані у стані спокою, включно з файлами БД і резервними копіями.
- SSL/TLS забезпечує захист даних під час передавання між клієнтом і сервером.
- Вибіркове шифрування полів доцільне для персональних даних, фінансових реквізитів та інших критичних атрибутів.

АУДИТ, МОНІТОРИНГ ТА ВИЯВЛЕННЯ АНОМАЛІЙ



37

оновлень.

Цей підхід гарантує, що система завжди захищена від останніх загроз.[27]

2.1.3.6. Маскування даних

Маскування даних дозволяє приховувати чутливу інформацію в тих випадках, коли доступ до неї не є необхідним. Наприклад:

– У тестових середовищах, де використовуються копії реальної бази даних.

– Для обмеження доступу користувачів із низьким рівнем привілеїв.

Маскування створює фіктивні дані, які зберігають ту саму структуру, але не розкривають реальної інформації. Це знижує ризик витоку навіть у разі компрометації.[27]

2.1.3.7. Резервне копіювання з шифруванням

– Резервні копії є основним засобом відновлення після інцидентів, але вони також потребують захисту. Основні заходи включають:

– Шифрування резервних копій: використання алгоритмів AES для забезпечення конфіденційності.

– Обмеження доступу: впровадження строгих політик доступу до сховищ із резервними копіями.

– Регулярне тестування: перевірка працездатності резервних копій для переконання, що вони можуть бути використані у разі потреби. [27]

2.1.3.8. Освітні заходи для співробітників

Багато ризиків виникають через людський фактор. Навчання адміністраторів і розробників є важливим компонентом захисту. Основні напрямки навчання включають:

– Розуміння принципів безпеки баз даних.

– Ознайомлення з найпоширеніщими загрозами, такими як SQL-ін'єкції або DDoS.

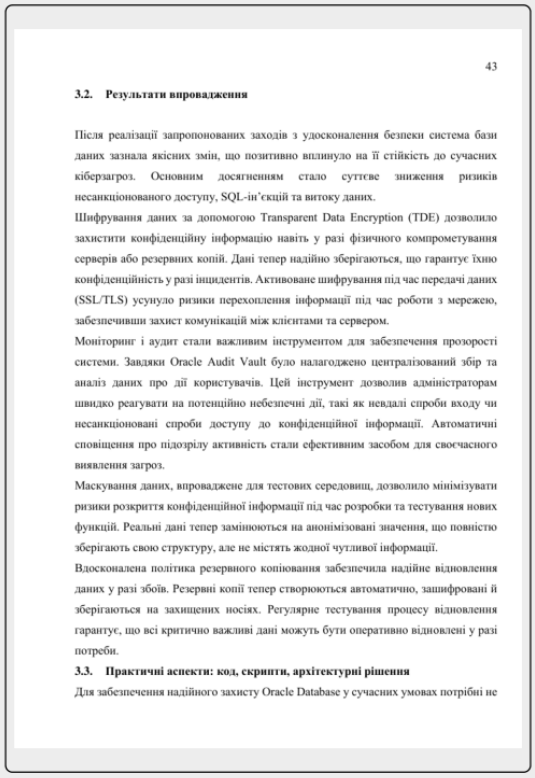
– Практичні вправи з налаштування безпечних середовищ.

Освітні заходи сприяють підвищенню загальної обізнаності та зменшують ризики, пов'язані з помилками персоналу.

Запропоновані заходи дозволяють організаціям значно підвищити рівень безпеки

- Oracle Audit Vault та DAM забезпечують централізований збір журналів і контроль дій користувачів.
- Моніторинг дозволяє фіксувати невдалі спроби входу, підозрілі SQL-запити та зміни привілеїв.
- Журнали подій є основою для розслідування інцидентів і доведення відповідності політикам безпеки.

ПРАКТИЧНА РЕАЛІЗАЦІЯ НА ПРИКЛАДІ БАЗИ ДАНИХ



43

3.2. Результати впровадження

Після реалізації запропонованих заходів з удосконалення безпеки система бази даних зазнала якісних змін, що позитивно вплинуло на її стійкість до сучасних кіберзагроз. Основним досягненням стало суттєве зниження ризиків несанкціонованого доступу, SQL-ін'єкцій та витоку даних.

Шифрування даних за допомогою Transparent Data Encryption (TDE) дозволило захистити конфіденційну інформацію навіть у разі фізичного компрометування серверів або резервних копій. Дані тепер надійно зберігаються, що гарантує їхню конфіденційність у разі інцидентів. Активоване шифрування під час передачі даних (SSL/TLS) усунуло ризики перехоплення інформації під час роботи з мережею, забезпечивши захист комунікацій між клієнтами та сервером.

Моніторинг і аудит стали важливим інструментом для забезпечення прозорості системи. Завдяки Oracle Audit Vault було налагоджено централізований збір та аналіз даних про дії користувачів. Цей інструмент дозволив адміністраторам швидко реагувати на потенційно небезпечні дії, такі як невдалі спроби входу чи несанкціоновані спроби доступу до конфіденційної інформації. Автоматичні сповіщення про підозрілу активність стали ефективним засобом для своєчасного виявлення загроз.

Маскування даних, впровадженне для тестових середовищ, дозволило мінімізувати ризики розкриття конфіденційної інформації під час розробки та тестування нових функцій. Реальні дані тепер замінюються на анонімізовані значення, що повністю зберігають свою структуру, але не містять жодної чутливої інформації.

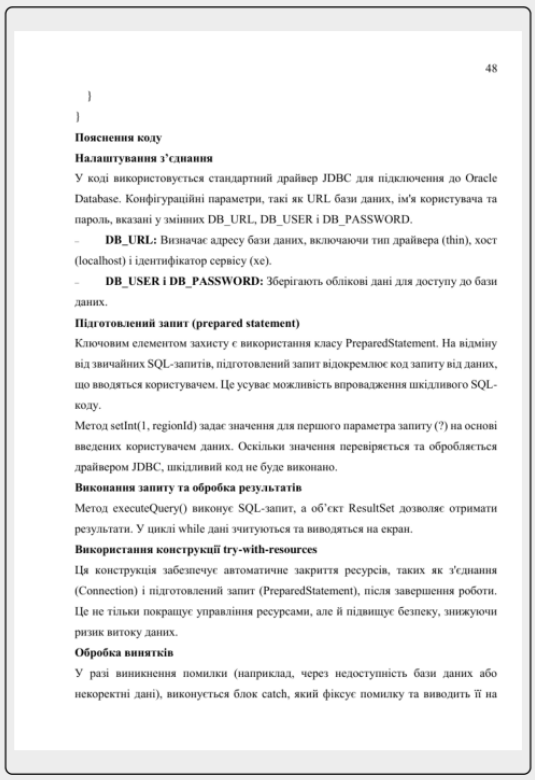
Вдосконалена політика резервного копіювання забезпечила надійне відновлення даних у разі збоїв. Резервні копії тепер створюються автоматично, зашифровані й зберігаються на захищених носіях. Регулярне тестування процесу відновлення гарантує, що всі критично важливі дані можуть бути оперативно відновлені у разі потреби.

3.3. Практичні аспекти: код, скрипти, архітектурні рішення

Для забезпечення надійного захисту Oracle Database у сучасних умовах потрібні не

- Практична частина спрямована на налаштування механізмів захисту Oracle Database на прикладі конкретної БД.
- Реалізуються користувачі, ролі, права доступу, обмеження привілеїв, аудит дій та контроль SQL-запитів.
- Окремо розглядаються скрипти, архітектурні рішення і перевірка працездатності запропонованих засобів.

КОД, СКРИПТИ ТА АРХІТЕКТУРНІ РІШЕННЯ



48

}

}

Пояснення коду

Налаштування з'єднання

У коді використовується стандартний драйвер JDBC для підключення до Oracle Database. Конфігураційні параметри, такі як URL бази даних, ім'я користувача та пароль, вказані у змінних DB_URL, DB_USER і DB_PASSWORD.

- **DB_URL:** Визначає адресу бази даних, включаючи тип драйвера (thin), хост (localhost) і ідентифікатор сервісу (xe).
- **DB_USER і DB_PASSWORD:** Зберігають облікові дані для доступу до бази даних.

Підготовлений запит (prepared statement)

Ключовим елементом захисту є використання класу PreparedStatement. На відміну від звичайних SQL-запитів, підготовлений запит відокремлює код запиту від даних, що вводяться користувачем. Це усуває можливість впровадження шкідливого SQL-коду.

Метод setInt(1, regionId) задає значення для першого параметра запиту (?) на основі введених користувачем даних. Оскільки значення перевіряється та обробляється драйвером JDBC, шкідливий код не буде виконано.

Виконання запиту та обробка результатів

Метод executeQuery() виконує SQL-запит, а об'єкт ResultSet дозволяє отримати результати. У циклі while дані зчитуються та виводяться на екран.

Використання конструкції try-with-resources

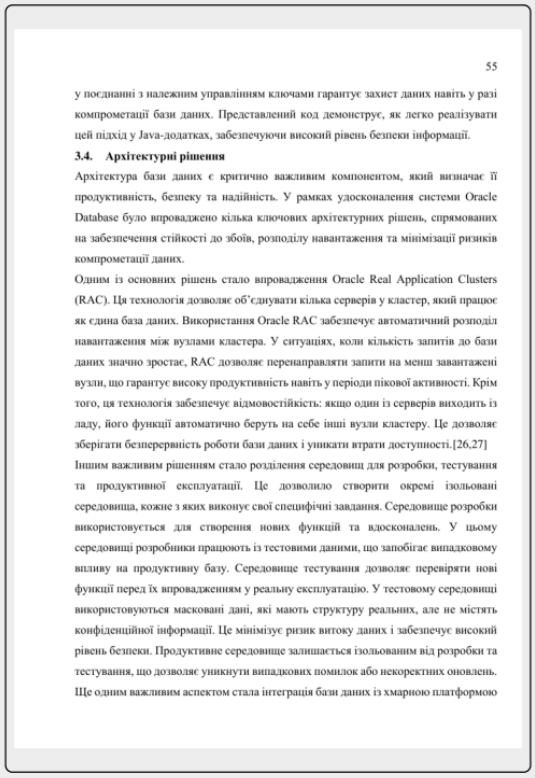
Ця конструкція забезпечує автоматичне закриття ресурсів, таких як з'єднання (Connection) і підготовлений запит (PreparedStatement), після завершення роботи. Це не тільки покращує управління ресурсами, але й підвищує безпеку, знижуючи ризик витоку даних.

Обробка винятків

У разі виникнення помилки (наприклад, через недоступність бази даних або некоректні дані), виконується блок catch, який фіксує помилку та виводить її на

- SQL/PLSQL-скрипти використовуються для створення ролей, користувачів, політик доступу й процедур контролю.
- Параметризовані запити та перевірка введених даних знижують ризик SQL-ін'єкцій.
- Архітектурне рішення передбачає поєднання автентифікації, авторизації, шифрування, аудиту та резервного копіювання.

ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ РІШЕНЬ



55

у поєднанні з належним управлінням ключами гарантує захист даних навіть у разі компрометації бази даних. Представлений код демонструє, як легко реалізувати цей підхід у Java-додатках, забезпечуючи високий рівень безпеки інформації.

3.4. Архітектурні рішення

Архітектура бази даних є критично важливим компонентом, який визначає її продуктивність, безпеку та надійність. У рамках удосконалення системи Oracle Database було впроваджено кілька ключових архітектурних рішень, спрямованих на забезпечення стійкості до збоїв, розподілу навантаження та мінімізації ризиків компрометації даних.

Одним із основних рішень стало впровадження Oracle Real Application Clusters (RAC). Ця технологія дозволяє об'єднувати кілька серверів у кластер, який працює як єдина база даних. Використання Oracle RAC забезпечує автоматичний розподіл навантаження між вузлами кластера. У ситуаціях, коли кількість запитів до бази даних значно зростає, RAC дозволяє перенаправляти запити на менш завантажені вузли, що гарантує високу продуктивність навіть у періоди пікової активності. Крім того, ця технологія забезпечує відмовостійкість: якщо один із серверів виходить із ладу, його функції автоматично беруть на себе інші вузли кластеру. Це дозволяє зберігати безперервність роботи бази даних і уникати втрати доступності.[26,27]

Іншим важливим рішенням стало розділення середовищ для розробки, тестування та продуктивної експлуатації. Це дозволило створити окремі ізольовані середовища, кожне з яких виконує свої специфічні завдання. Середовище розробки використовується для створення нових функцій та вдосконалень. У цьому середовищі розробники працюють із тестовими даними, що запобігає випадковому впливу на продуктивну базу. Середовище тестування дозволяє перевіряти нові функції перед їх впровадженням у реальну експлуатацію. У тестовому середовищі використовуються масковані дані, які мають структуру реальних, але не містять конфіденційної інформації. Це мінімізує ризик витоку даних і забезпечує високий рівень безпеки. Продуктивне середовище залишається ізольованим від розробки та тестування, що дозволяє уникнути випадкових помилок або некоректних оновлень.

Ще одним важливим аспектом стала інтеграція бази даних із хмарною платформою

- Після впровадження заходів зменшується кількість критичних ризиків, пов'язаних із несанкціонованим доступом і витоком даних.
- Підвищується контрольованість роботи користувачів завдяки журналюванню та моніторингу.
- Комплексний підхід забезпечує кращий баланс між безпекою, продуктивністю та керованістю Oracle Database.

КЛЮЧОВІ РЕЗУЛЬТАТИ РОБОТИ

- Проведено аналіз сучасних підходів до забезпечення безпеки баз даних та особливостей функціонування Oracle Database.
- Визначено основні вразливості Oracle: SQL-ін’єкції, слабка автентифікація, надмірні привілеї, відсутність шифрування та аудиту.
- Запропоновано й обґрунтовано комплекс заходів: MFA, RBAC, Least Privilege, TDE, SSL/TLS, аудит, моніторинг, резервне копіювання.
- Реалізовано практичні рішення на прикладі конкретної бази даних та виконано аналіз їх ефективності.

ДЯКУЮ ЗА УВАГУ!

Доповідь підготував здобувач вищої освіти Олександр КУЖЕНТСЬКИЙ

Тема: «Система безпеки СУБД Oracle на прикладі бази даних»

Київ - 2025