Харківський національний університет імені В.Н. Каразіна

Факультет комп'ютерних наук

Безпека інформаційних систем і технологій

«Допущено до захисту»

Зав.кафедрою БІСТ

Сватовський І.І._____

« » червня 2023р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

спеціальність: 125 Кібербезпека

на тему: «Аналіз та дослідження хмарних сервісів для зберігання та синхронізації даних»

оцінка « » Керівник к.т.н Єсіна М. В.

(прізвище та ініціали/підпис)

Голова ЕК Рецензент к.т.н Бобух В. А.

(прізвище та ініціали/підпис)

Лемешко О.В._____ Виконавець студентка групи КБ-41

Кравченко С. О.

(прізвище та ініціали/підпис)

Харків – 2023

РЕФЕРАТ

Пояснювальна записка містить 59 сторінок, 4 рисунки, 2 таблиці, 2 додатки, 56 джерел**.**

Метою дипломної роботи є дослідження хмарних сервісів, їх особливості та моделі, досвід їх використання у різних сферах; аналіз та дослідження синхронізації даних, особливості процесу синхронізації, моделі та інструменти реалізації синхронізації, аналіз та дослідження збереження даних у хмарних сервісах, опис процесу збереження даних та методів збереження; аналіз моделей розгортання, загроз та безпеки у хмарних сервісах.

Об’єктом дослідження дипломної роботи є хмарні сервіси та їхні моделі, синхронізація даних, збереження даних, моделі загроз для постачальників та користувачів, модель порушника та модель безпеки у хмарних сервісах.

Методи дослідження: аналіз наукової літератури, аналіз досвіду користувачів та постачальників хмарних сервісів.

Результатами проведеної роботи є аналіз та дослідження хмарних сервісів та їх використання в повсякденному житті, а також існуючих моделей розгортання та обслуговування, які відрізняються за рівнем надання послуг та рівнем відповідальності, що значно спрощує життя користувачів, також обмежують коло людей яким доступна ця хмара. Аналіз та дослідження хмарних сервісів для синхронізації та зберігання даних, їх особливості, методи та інструменти. Проведення аналізу моделей загроз, порушника та безпеки у хмарних сервісах, важливість використання яких допомагає запобігати загрозам та зменшувати можливі втрати через витік даних.

Ключові слова: МОДЕЛЬ БЕЗПЕКИ, МОДЕЛЬ ЗАГРОЗ, МОДЕЛІ ОБСЛУГОВУВАННЯ, МОДЕЛЬ ПОРУШНИКА, МОДЕЛІ РОЗГОРТАННЯ, ОБРОБКА ДАНИХ, СИНХРОНІЗАЦІЯ ДАНИХ, ХМАРНІ СЕРВІСИ.

ABSTRACT

The explanatory note contains 59 pages, 4 figures, 2 tables, 2 appendices, 56 sources.

The aim of the thesis is to study cloud services, their features and models, experience of their use in various fields; analysis and research of data synchronization, features of the synchronization process, models and tools for implementing synchronization, analysis and research of data storage in cloud services, description of the data storage process and storage methods; analysis of deployment models, threats and security in cloud services.

The object of research of the thesis is cloud services and their models, data synchronization, data storage, threat models for providers and users, the model of the violator and the model of security in cloud services.

Research methods: analysis of scientific literature, analysis of the experience of users and providers of cloud services.

The results of this work are the analysis and research of cloud services and their use in everyday life, as well as existing models of deployment and maintenance, which differ in the level of service provision and the level of responsibility, which greatly simplifies the lives of users, and also limits the number of people who can access this cloud. Analysis and research of cloud services for data synchronization and storage, their features, methods and tools. Conducting analysis of threat, intruder and security models in cloud services, the importance of using which helps to prevent threats and reduce possible losses due to data leakage.

Keywords: CLOUD SERVICES, DATA PROCESSING, DATA SYNCHRONIZATION, DEPLOYMENT MODELS, SECURITY MODEL, SERVICE MODELS, THREAT MODEL, VIOLATOR MODEL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

ACID – Атомарність, послідовність, ізоляція, довговічність
API – Прикладний програмний інтерфейс
DDoS – Розподілена атака на відмову в обслуговуванні
ETL – Витяг, перетворення, завантаження
GFL – Файлова система Google
HDFS – Розподілена файлова система Hadoop
HTTP – Протокол передачі гіпертексту
IaaS – Інфраструктура як послуга
NIST – Національний інститут стандартів і технології
PaaS – Платформа як послуга
RPC – Виклик віддалених процедур
SaaS – Програмне забезпечення як послуга
SQL – Мова структурованих запитів
UNIX – Сімейство багатозадачних операційних систем
АС – Автоматизована система
БД – Бази даних
ІКТ – Інформаційно-комунікаційні технології
ІТ – Інформаційні технології
ІТС – Інформаційно-технологічний супровід
ОС – Операційна система
ПК – Персональний комп’ютер
ЦП – Центральний процесор
ЦРУ – Конфіденційність, цілісність, доступність

ВСТУП

Хмарні сервіси є достатньо молодою технологією, яка швидко набула своєї популярності та продовжує це робити. Хмарні сервіси стали неодмінною складовою сучасного світу інформаційних технологій, надаючи широкий спектр послуг для збереження, обробки та обміну даними. Завдяки їх високій масштабованості, доступності та ефективності, хмарні сервіси знаходять все більше застосувань у різних галузях, від бізнесу до особистого використання.

Майже кожна людина чула про такі популярні хмарні сервіси, як Dropbox, Google Drive, Microsoft OneDrive та Amazon Web Services. Хтось використовує хмарні сервіси просто для комфортного зберігання інформації, наприклад, фотографія та відео, хтось здобуває освіту за допомогою використання хмарних сервісів, тому, що вони надають багато корисних послуг, для індивідуального або групового навчання та інше.

Багато організацій впровадили хмарні сервіси у свою роботу, що допомогло покращити ефективність роботи, оскільки працівники мають змогу працювати з будь-якої точки світу, де є Інтернет-з’єднання, з будь-якого пристрою, такі як ПК, смартфон, планшет та інші.

Однак, разом зі зростанням популярності хмарних сервісів, з'являються й нові виклики щодо безпеки та надійності збереження та синхронізації даних. Питання щодо конфіденційності, цілісності та невідмовності даних стають особливо актуальними. Навіть найменша помилка або вразливість можуть призвести до втрати даних, порушення конфіденційності або незабезпеченості даних від несанкціонованого доступу.

Важко визначити вразливі місця хмарних сервісів, оскільки помилка, загроза або атака може трапитись на будь-якому етапі, починаючи з аутентифікації користувача та закінчуючи збереженням даних. Кількість інформації, яка зберігається та обробляється у хмарних сервісах є привабливою

для злочинців, але з розвитком загроз на хмарні сервіси розвиваються і методи забезпечення безпеки даних, які з кожним днем вдосконалюються, оновлюються та створюються.

В процесі виконання дипломної роботи буде проаналізовано хмарні сервіси та детально розглянуто процес синхронізації та збереження та обробка даних. Синхронізація та збереження є важливими компонентами хмарних сервісів. У цій роботі було зосереджено увагу на методах синхронізації, такі як синхронізація за прапором стану, синхронізація за мітками часу, математична синхронізація та інші, а також на інструментах синхронізації.

На цей момент у хмарних сервісах зберігається велика кількість даних, яку важко уявити звичайній людині. Саме тому процес зберігання має бути безпечним та надійним. При аналізі та досліджені зберігання даних будуть також розглянуті їх моделі та подальша обробка даних, адже ці процеси є зв’язаними.

Для високо рівня безпеки даних потрібен ретельний аналіз загроз для постачальників хмарних сервісів та їх користувачів, а також розуміння поведінки порушників, які можуть завдати шкоди даним.

Метою даної роботи є аналіз та дослідження хмарних сервісів для зберігання та синхронізації даних. В результаті роботи буде розглянуто моделі порушників, загроз та безпеки хмарних сервісів.

1 ОГЛЯД, АНАЛІЗ ТА ДОСВІД ВИКОРИСТАННЯ ХМАРНИХ СЕРВІСІВ

1.1 Загальні відомості

Треба почати з того, що хмарні обчислення – це широкий термін, який просто означає надання обчислювальних послуг, які можуть включати сервери, бази даних, сховище, мережі, програмне забезпечення, аналітику даних, рішення безпеки, організаційні системи, віртуальні комп'ютери та багато іншого – через Інтернет.

Звернемось до визначення хмарних обчислень яке надає NIST: «Хмарні обчислення – це модель забезпечення повсюдного, зручного мережевого доступу на вимогу до спільного пулу конфігурованих обчислювальних ресурсів (наприклад, мереж, серверів, сховищ, додатків і сервісів), які можна швидко надавати та звільняти з мінімальними зусиллями з управління або взаємодії з постачальниками послуг» [1].

Мета хмарних обчислень – дозволити користувачам скористатися перевагами всіх цих технологій. Багато організацій переїжджають у хмару, оскільки це дозволяє користувачам зберігати свої дані в хмарі та мати доступ до них у будь-який час з будь-якого місця.

Також NIST визначає такі характеристики хмарних обчислень [1]:

1) Самообслуговування на вимогу. Споживач може в односторонньому порядку надавати обчислювальні можливості, такі як серверний час і мережеве сховище, за потреби автоматично, без необхідності взаємодії людини з кожним постачальником послуг.

2) Широкий доступ до мережі. Можливості доступні через мережу і доступні через стандартні механізми, які сприяють використанню гетерогенних тонких або товстих клієнтських платформ (наприклад, мобільні телефони, планшети, ноутбуки та робочі станції).

3) Об'єднання ресурсів. Обчислювальні ресурси провайдера об'єднуються для обслуговування декількох споживачів за моделлю мультиорендаря, при цьому різні фізичні та віртуальні ресурси динамічно призначаються і перепризначаються відповідно до попиту споживачів. Існує відчуття незалежності від місця розташування, оскільки клієнт, як правило, не контролює і не знає точного місця розташування наданих ресурсів, але може вказати місце розташування на більш високому рівні абстракції (наприклад, країна, штат або центр обробки даних). Приклади ресурсів включають зберігання, обробку, пам'ять і пропускну здатність мережі.

4) Швидка еластичність. Ресурси можуть гнучко надаватися і вивільнятися, в деяких випадках автоматично, для швидкого масштабування ззовні та всередину відповідно до попиту. Споживачеві часто здається, що можливості, доступні для надання, необмежені та можуть бути використані в будь-якій кількості в будь-який час.

5) Вимірюваний сервіс. Хмарні системи автоматично контролюють і оптимізують використання ресурсів, використовуючи можливості обліку на певному рівні абстракції, що відповідає типу послуги (наприклад, зберігання, обробка, пропускна здатність і активні облікові записи користувачів). Використання ресурсів можна відстежувати, контролювати та звітувати, забезпечуючи прозорість як для постачальника, так і для споживача послуги, що використовується.

1. 2 Огляд та аналіз архітектури хмарних обчислень

Систему хмарних обчислень можна поділити на дві частини, які зв’язані між собою через Інтернет: front-end, те що бачить користувач, і back-end – хмара системи [4]. До front-end відноситься комп’ютер клієнта та додаток, необхідний для доступу до хмари, під back-end мається на увазі хмарні обчислювальні сервери, такі як комп’ютери, сервери та сховища даних [5]. Задоволення запитів клієнтів, адміністрування системи та моніторинг трафіку забезпечує

центральний сервер, який дотримується спеціальних протоколів та використовує спеціальне програмне забезпечення.

Архітектуру хмарних обчислень можна поділити на такі рівні як клієнт, послуги та сервер.

Клієнт хмари – це комп'ютерне обладнання клієнта та/або програмне забезпечення, яке покладається на послуги хмарного обчислення, а сервер – комп'ютерне обладнання та/або програмне забезпечення, необхідне для надання послуг хмарних обчислень. Послуги, які надають хмарні обчислення діляться на безліч інших, такі як резервне копіювання як послуга, сховище як послуга та інші, але NIST виділяє три головні послуги: програмне забезпечення як послуга, платформа як послуга та інфраструктура як послуга [1].

1.2.1 Моделі обслуговування

Відрізняють такі основні моделі обслуговування як:

- Програмне забезпечення як послуга

У підході SaaS постачальник програмного забезпечення розміщує програму у своєму центрі обробки даних, і в більшості випадків клієнт отримує доступ до неї через веб-браузер або спеціальний інтерфейс, який було завантажено на комп'ютер користувача. Споживач не керує базовою хмарною інфраструктурою, включаючи мережу, сервери, операційні системи, сховище або навіть окремі можливості додатків, за винятком обмежених налаштувань конфігурації додатків для конкретного користувача [6].

Користувачі можуть отримати доступ до додатків і сервісів SaaS з будь-якого місця за допомогою комп'ютера або мобільного пристрою з доступом до Інтернету. У моделі SaaS користувачі отримують доступ до прикладного програмного забезпечення та баз даних.

До моделі SaaS охоплювати такі характеристики:

- Користувачі мають змогу користуватися функціональними можливостями програмного забезпечення без самостійного розгортання

програмного забезпечення. Доступ до програм здійснюється через веб-інтерфейс, який запускається з Інтернету, забезпечує високий рівень масштабованості.

- Модель SaaS дозволяє кожному споживачеві отримати вигоду від останніх технологічних функцій постачальника.
- Доступність, оскільки кінцеві користувачі можуть отримати доступ до програмного забезпечення з будь-якого місця, де є підключення до Інтернету.

Прикладами моделі SaaS є SalesForce.com, Google Mail, Google Docs та інші.

- Платформа як послуга

PaaS належить до додатків, створених мовою розробки, яка розміщується постачальником хмарних послуг у хмарній інфраструктурі [7]. Вона надає інфраструктуру та повне операційне середовище та середовище розробки для розгортання додатків користувачів, але на відміну від інфраструктури як послуги, користувач не контролює екземпляр віртуалізації або мережеву конфігурацію хмарного сховища.

PaaS дозволяє безпечно зберігати та обробляти конфіденційні дані користувачів, використовуючи можливості криптографічних співпроцесорів для захисту від несанкціонованого доступу [3].

До характеристик PaaS відноситься:

- Економія часу користувачів, адже вони можуть зосередиться на розробці програмного забезпечення, а не на підтримці, оновленні та роботі над рутинними завданнями, пов’язаними з середовищем тестування та розгортання.
- Процес зворотного зв'язку щодо конфіденційності, який інформує користувачів про різні операції, що застосовуються до їхніх даних, та інформує їх про будь-які потенційні ризики, які можуть поставити під загрозу конфіденційність їхньої інформації [8].

До поширених продуктів PaaS належать Lightning Platform від Salesforce, AWS Elastic Beanstalk та Google App Engine.

- Інфраструктура як послуга

Споживач не керує і не контролює базову хмарну інфраструктуру, але має контроль над операційними системами, сховищами, розгорнутими додатками та, можливо, обмежений контроль над окремими мережевими компонентами (наприклад, брандмауерами хостів) [1].

IaaS прагне бути повністю сторонньою послугою, яка замінює центр обробки даних та пропонує попередньо налаштоване обладнання (або програмне забезпечення) через інтерфейс. Користувачі мають виділений обсяг пам'яті та можуть запускати, зупиняти, отримувати доступ і налаштовувати віртуальну машину і сховище за власним бажанням.

Переваги рішень IaaS [9]:

- Знижує вартість капітальних витрат.
- Користувачі платять за послуги, які їм потрібні.
- Доступ до IT-ресурсів та інфраструктури корпоративного рівня.
- Користувачі можуть у будь-який час збільшувати та зменшувати масштаб ресурсів відповідно до своїх вимог.

Прикладами використання IaaS є Rackspace Cloud Servers, Google, Amazon EC2, IBM та Verizon.

1.2.2 Моделі розгортання

Моделі розгортання відрізняються між собою за рівнем відносин між споживачем послуг та постачальником хмарних послуг. Так, наприклад, один користувач може бути зацікавлений в особистій хмарі, а інший – в хмарі з доступом до багатьох суб’єктів. Також, модель можна обрати за рівнем потреб споживача, наприклад, низька вартість експлуатації, чи високий рівень забезпечення надійності та безпеки. NIST визначає чотири моделі розгортання хмар: відкрита, закрита, суспільна та гібридна [1].

- Публічна хмара

Дана модель хмари перебуває в юрисдикції постачальника хмарних послуг, який надає хмарні послуги для широкого кола громадськості, а керування

інфраструктурою належить до постачальника публічної хмари. Сутність цієї хмари полягає в тому, що будь-який користувач з доступом до Інтернету може скористатися послугами хмари.

До переваг публічної хмари зі сторони постачальника послуг відноситься відсутність початкового капіталовкладення в інфраструктуру та перенесення ризиків на споживачів даної моделі хмари [2]. Користувачі, своєю чергою, отримують просте налаштування, безперервний час безперебійної роботи, доступність до даних та масштабованість.

Треба зазначити, що публічна хмара має вагомі недоліки, такі як втрата конфіденційності та безпеки даних. Споживачам хмари не надається інформація щодо місця зберігання даних, метода резервного копіювання та доступу до неавторизованих користувачів.

Прикладами публічних хмар є Google Apps, Amazon AWS, Microsoft Office 365 та інші, які широко розповсюджені та маються велику аудиторію споживачів.

- Приватна хмара

Ця хмарна інфраструктура надається виключно єдиній організації та може належати та управлятися організацією, третьою стороною або деякою їх комбінацією, і хмара може існувати на території організації або за її межами [1].

Приватна хмара розміщується в центрі обробки даних компанії та надає свої послуги лише користувачам всередині цієї компанії або її партнерам, тому доступ до цієї хмари мають тільки члени організації та вповноважені треті особи.

Головною перевагою приватної хмари є забезпечення високого рівня безпеки даних та конфіденційності, а також економія коштів, якщо вона використовує вільні потужності вже наявного центру обробки даних. Надання таких невикористаних потужностей через хмарні інтерфейси дозволяє використовувати ті ж інструменти, що і при роботі з публічними хмарами, а також скористатися можливостями, притаманними програмному забезпеченню для управління хмарами, такими як інтерфейс самообслуговування, автоматизоване

управління обчислювальними ресурсами та можливість продавати наявні надлишкові потужності компаніям-партнерам [3].

З недоліків приватної хмари виділяють високу вартість, оскільки придбання обладнання, програмного забезпечення та персоналу призводить до вищих затрат для організації.

- Суспільна хмара

Ця хмара схожа на комбінацію приватної та публічних хмар, адже інфраструктура надається для ексклюзивного використання двом або більше організаціям, які мають спільні інтереси або спільні проблеми щодо безпеки та конфіденційності. Суспільна хмара може належати та управлятися однією чи декількома організаціями в спільноті, третьою стороною чи деякою їх комбінацією, і може існувати на території організації або організацій, або за їх межами [1].

До переваг цієї хмари можна віднести вартість, оскільки всі затрати розподіляються між учасниками спільноти, і можливість передати управління хмарою на аутсортінг хмарному постачальнику, тобто обрати третю неупереджену сторону, яка не буде зв’язана з жодним членом суспільної хмари та пов’язана контрактом.

З недоліків можна виділити фіксований обсяг пропускної здатності та сховища даних, оскільки відбувається розподілення між всіма організаціями спільноти.

- Гібридна хмара

Ця хмара уявляє собою сукупність двох хмар або більше, які залишаються унікальними об'єктами, але пов'язані між собою стандартизованою або власною технологією, що дозволяє переносити дані та додатки (наприклад, "хмарний вибух" для балансування навантаження між хмарами) [1]. В ідеалі гібридна хмара дозволяє бізнесу скористатися перевагами масштабованості та економічної ефективності, які пропонує публічне хмарне обчислювальне середовище, не наражаючи критично важливі програми та дані на сторонні вразливості [3].

До переваг гібридної хмари можна віднести загальну гнучкість організацій завдяки можливості використати публічні хмари, що допомагає оптимізувати витрати на інфраструктуру на різних етапах життєвого циклу додатків та можливість міграції локальних на масштабовані хмарні рішення без втрати систем, які працюють сьогодні.

Вагомим недоліком є складність виявлення проблем в середині через складну архітектуру хмари, а також можливі ризики у політиці безпеки, що охоплюють гібридне хмарне середовище.

1.3 Досвід використання хмарних сервісів

У [14] хмарні сервіси визначаються як послуги, які надають користувачеві мережевий доступ до масштабованого та гнучко організованого пулу розподілених фізичних або віртуальних ресурсів, що надаються в режимі самообслуговування та адміністрування на вимогу (наприклад, програмне забезпечення, сховище, обчислення та обчислювальні можливості).

Як засвідчує дослідження аналітичної агенції Gartner – хмарні сервіси почали набирати свою популярність у 2009 році [10]. З кожним роком все більше компаній вдосконалювалось в перевагах хмарних обчислень та впроваджували їх в свою роботу. Як показує статистика на рис 1.1, у 2020 році обсяг ринку хмарних сервісів у світі складав 266 млрд. доларів, а у 2022 році – 309 млрд. [11] зазначає, що український ринок хмарних обчислень теж набрав помітної популярності, і у 2019 році ринок зріс майже вдвічі.

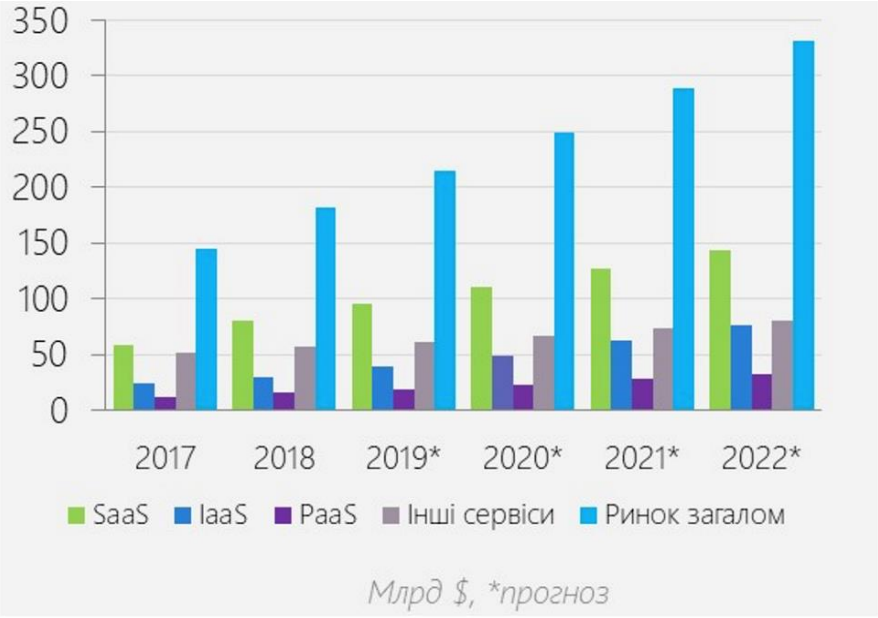

Рисунок 1.1 – Найпопулярніші хмарні сервіси у світі

Така шалена статистика пояснюється появленням вірусу COVID, через який весь світ був вимушений продовжувати набувати освіту та працювати вдома. Наразі під час військового положення в Україні через вторгнення російської федерації на територію України, школярі, студенти та працівники багатьох галузей продовжують свою освіту та життя за допомогою хмарних сервісів.

1.3.1 Використання хмарних сервісів для бізнесу

Треба почати з того, що кожна компанія в світі використовує хмарні обчислення, оскільки це надає великий потенціал для розвитку та забезпечує ефективні бізнес-можливості для компаній.

Використовуючи хмарні обчислення, компанії отримують певні переваги [15]: зменшення інвестицій у створення IT-інфраструктури та адміністративних витрат для IT-персоналу; можливості використання сучасних ІКТ, які дозволяють ефективно управляти бізнес-процесами; масштабність бізнесу та адаптація та пристосування до мінливих умов ринку.

Розглянемо найпоширеніші хмарні сервіси, такі як Google Apps, Microsoft Office 365 та Zoho Docs, які широко використовуються в сьогоденні та

забезпечують компанії доступністю, легким адмініструванням, комфортним обміном документами та співпрацею зі сторонніми компаніями.

- Google Apps for Work

Google Apps for Work – це набір програм на основі хмарних технологій, якій надає професійні послуги для компаній, такі як електрону пошту, захищені відеоконференції, спільні календарі, спільний доступ до документів для подальшого редагування, Workspace Storage для зберігання файлів та створювання резервних копій для кожного користувача.

На офіційній сторінці [16] можна побачити місячну вартість підписки на Google Workspace, яка варіюється від $7.20 до $21.60 в залежності від типу бізнесу (Бізнес Початковий, Бізнес Стандартний, Бізнес Плюс) та типу плану (гнучкий план, річний або строковий план).

Google надає адміністративний контроль для компаній, за яким вони можуть [16]:

- керувати обліковими записами користувачів і налаштуваннями безпеки з центральної консолі адміністратора;
- контролювати доступ користувачів до функцій і послуг;
- дистанційно керувати особистим мобільним парком;
- відстежувати тенденції використання за допомогою перевірок і звітів;
- отримувати цілодобову підтримку від експертів Google Workspace.

- Microsoft Office 365

Microsoft Office 365 надає можливість обмінюватися даними в режимі реального часу та інструменти для комфортної співпраці, що достатньо схожі на інструменти вищерозглянутого сервісу. Всі документи, які створюються та змінюються в хмарі зберігаються та синхронізуються на всіх пристроях, таким чином забезпечуючи можливістю мати доступ до попередніх версій користувачам.

Microsoft Office 365 надає три бізнес-плани (Бізнес Базовий, Бізнес Стандартний та Бізнес Преміум), місячна вартість яких залежить від обраного плану та вимог до плану [17].

Користувачі отримують [17]:

- настільні версії програм Microsoft 365: Outlook, Word, Excel, PowerPoint, OneNote (а також лише Access і Publisher для ПК);
- можливість зберігати та діліться файлами за допомогою хмарного сховища OneDrive;
- щомісячне автоматичне оновлення особистої програми новими функціями та можливостями;
- отримання допомоги в будь-який час завдяки цілодобовій телефонній та веб-підтримці від Microsoft.

- Zoho Docs

Zoho Docs – це комплексна онлайнова система керування документами, яка використовується для створення, зберігання, спільного використання та спільної роботи над документами майже будь-якого доступного формату [18].

Даний хмарний сервіс буде корисним будь-якої компанії, для керування документами, співпраці, адміністративним керуванням, безпеки та аналітики.

Сервіс має Zoho Office Suite, який містить текстовий процесор, програмне забезпечення для створення слайд-шоу, а також програми для роботи з таблицями. Zoho Docs дозволяє синхронізувати зміни в документах на всіх підключених пристроях користувача.

Система оплати за послуги мало чим відрізняється від попередніх представників хмарних сервісів, та вимагає щомісячної оплати вартість якої залежить від типу, які можуть бути безкоштовним, стандартним та преміум [19].

1.3.2 Використання хмарних сервісів для навчання

Хмарні сервіси набули своєї популярності не тільки в сфері бізнесу, а ще є в сфері освіти, тим самим спрощуючи організацію освітнього процесу. За

допомогою різних сервісів студенти мають інструменти для групових проєктів, такі як спільне редагування документів з подальшою синхронізацією, можливість підтримувати зв’язок через чати та відеоконференції, підтримка зв’язку з викладачем та інше. Поява таких високотехнологічних платформ, як Google Apps, Microsoft Windows Azure, IBM Blue Cloud та інші розширили доступність та якість освіти, зокрема індивідуальної.

Сьогодні університети можуть відігравати ключову роль у формуванні регіональних стратегій та визначенні пріоритетів місцевої спеціалізації, враховуючи інтелектуальні ресурси, навички та компетенції науково-виробничого персоналу, що існує як в університетських структурах, так і в місцевому бізнесі [12].

Використання хмарних сервісів в освітньому процесі надає можливість покращити такі аспекти, як:

- зменшення витрат на надання матеріалів, методом їх завантаження на електронні хмари;
- прозорий процес перевірки робіт за допомогою інформації про час відправлення роботи, відправника та отримувача роботи;
- збереження інформації щодо відвідування та доступу до матеріалів в одному місці;
- можливість адаптувати хмарний сервіс під особисті запити навчального закладу та зробити його найбільш корисним для навчального процесу;
- доступність до хмарного сервісу через Інтернет, уникаючи необхідності використання спеціального обладнання.

2 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХМАРНИХ СЕРВІСІВ ДЛЯ ЗБЕРІГАННЯ ДАНИХ

2.1 Аналіз даних

З розвитком технологічного прогресу та збільшенням кількості користувачів в мережі відповідно експоненційно зростає обсяг даних, які потребує вилучення та завантаження. Ці дані генеруються в процесі створення технологій допомоги водієві та автономних транспортних засобів, пристроїв Інтернету речей, включаючи сенсори в будинках, фабриках і містах, створення контенту з високою роздільною здатністю для відео 360 і доповненої реальності та 5G-зв'язку [13].

На рис. 2.1 показано прогноз від [13], що зазначає, що обсяг створених цифрових даних до 2025 року зросте до 175 зетабайт або 175 трильйонів гігабайт.

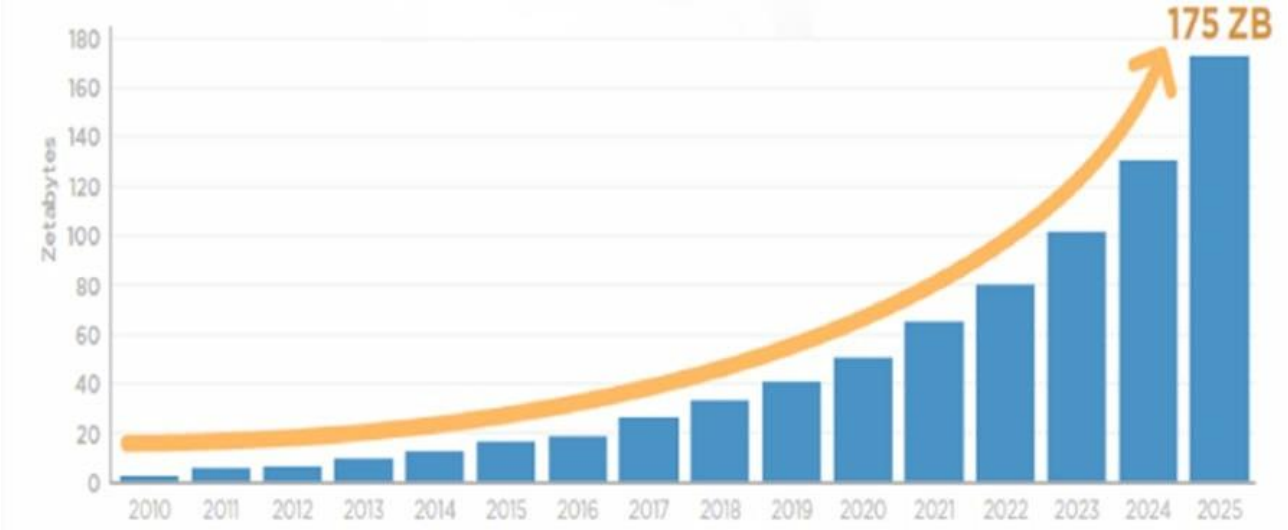

Рисунок 2.1 – Обсяг створених цифрових даних

Така кількість даних вимагає їх ретельного аналізу та надійного зберігання за допомогою методу керування даними. Це спонукає дослідників розробляти нові типи баз даних для їх обробки та забезпечувати високий рівень їх захисту.

2.2 Збір та зберігання даних

2.2.1 Збір даних

Відрізняють два види даних, такі як статистичні пакетні дані, які зберігаються у статичній формі, та динамічні потокові дані, які надаються безперервною послідовністю екземплярів. Потокові дані не зберігаються повністю, і багато елементів відкидаються відразу після обробки [31].

Збір потокових даних через свою нестабільну передачу вимагає особливого інструменту збору, який буде забезпечувати надійність, відмовостійкість та стабільність. Такою системою є Flume, надійна та відмовостійка розподілена потокова система обробки даних, яка збирає, агрегує та передає велику кількість каротажних даних з різних джерел до централізованого сховища [32]. Flume виступає проміжним програмним забезпеченням між приймачами даних та їх джерелами.

Треба зазначити ще одну актуальну для потових даних систему, а саме систему Kafka – універсальна система обміну повідомленнями з відкритим вихідним кодом, яка в основному використовується для побудови конвеєрів даних у реальному часі та потокових додатків [33]. Kafka уникає асинхронізації між обробкою даних та їх генерацією за допомогою використання черг для обробки даних.

В свою чергу, статичні пакетні дані потребують надійної передачі та збір різних типів даних. Головним інструментом для виконання цих задач є ETL, який отримує дані з джерел, потім їх перетворює та завантажує до сховищ [34]. Рисунок 2.2 детально показує, як у процесі роботи ETL використовує різні операції обробки, такі як з’єднання та перетворення, для очищення даних від пошкоджених.

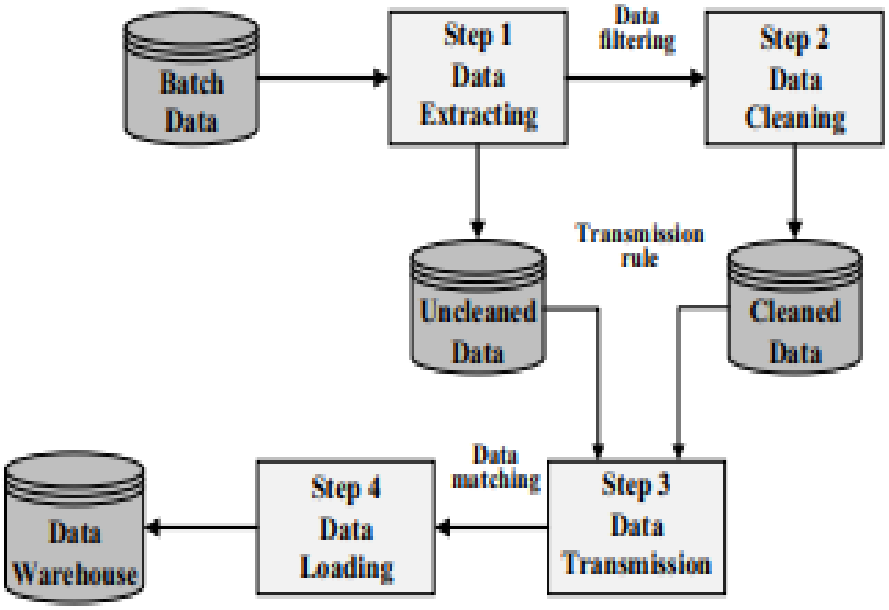

Рисунок 2.2 – Процес роботи ETL

2.2.2 Зберігання даних

В останні десятиліття широкого застосування набули реляційні бази даних та методи управління структурованими даними [35]. Широкого використання набули такі бази даних як NoSQL, NewSQL, а також розподілені файлові системи та інші системи управління даними.

У 2003 році компанія Google розробила масштабовану файлову систему для великомасштабного розподіленого зберігання даних під назвою GFS [36]. HDFS є частиною базового проєкту Apache Hadoop, а її ідея проєктування належить до архітектури сервісів великих даних до системи GFS [37]. HDFS на сьогодні вважається найпоширенішим інструментом для роботи з великими даними, що підтримує надмірність, надійність, масштабованість для систем з паралельною розподіленою архітектурою [38].

Розглянемо базу даних NoSQL, яка має нереляційний режим управліннями даних. Вона містить пари ключ-значення, колонкові, графіки або документи. У таблиці 2.1 зазначено декілька типів БД NoSQL, надається їх коротка характеристика та недоліки та переваги.

Таблиця 2.1 – Типи баз даних NoSQL

Бази даних NoSQL	Типи	Переваги	Недоліки
Redis	Ключ-значення	Відмінна продуктивність читання і запису; постійність даних; розділення читання та запису	Без функції автоматичної відмовостійкості та відновлення; не підтримується для розширення онлайн
Memcached		Обробка даних на основі пам'яті; розподілений і масштабований режим; висока швидкість обробки	Невелика ємність одного кеша; не підтримується збереження даних
MongoDB	Документноорієнтовані	Висока швидкість доступу; кілька типів даних; висока ефективність запитів	Низька ефективність читання-запису; велика займаність простору
CouchDB		Доступність і одночасність; висока гнучкість	Низька ефективність запитів; висока складність обслуговування
Hbase	Колонкові	Висока швидкість завантаження; підходить для зберігання великих даних; ефективна швидкість стиснення	Низька ефективність мультиумовного запиту; безпосередня підтримка запиту SQL
Cassandra		Еластична масштабованість; гнучка схема; багатоспискові структури даних	Не підтримує транзакції ACID і атомарні операції

Продовження таблиці 2.1 – Типи баз даних NoSQL

Бази даних NoSQL	Типи	Переваги	Недоліки
Neo4j	Графоорієнтовані	Ефективне виконання запитів; Високопродуктивний графічний алгоритм	Не підтримується великий розділ графа
GraphDB		Транзакції ACID та атомарні операції; багатомодельні об'єкти	Висока складність

Бази даних, які належать до типу ключ-значення, застосовуються для зберігання інформації користувача, пов’язаної з ідентифікатором, конфігураційні файли та параметри. Redis та Memcached мають просту структуру даних, високу масштабованість, просту структуру та низьку ефективність оновлення запитів.

Документоорієнтовані бази даних мають блоки обробки даних на основі документів та кілька форматів даних. Вони орієнтовані для зберігання файлів журналу та аналізу даних у реальному часі.

Колонкові бази даних мають високу кореляцію даних в одному сімействі стовпців та мають модель зберігання на її основі. Вони дозволяють мати доступ для читання в реальному часі та підходять для роботи над великими даними.

Графоорієнтовані бази даних використовують формат зберігання на основі графів та має ефективне виконання запитів. Вони підходять для реляційного зберігання даних, механізмівтощо.

Що стосовно NewSQL, то на відміну від NoSQL, вона реляційною та має не тільки таку ж масштабованість, як NoSQL, але й надає ACID та SQL-сервіси для транзакцій [40].

2.3 Обробка даних

Обробка даних – це важливий процес після, оскільки саме він відділяє та очищає необроблені дані для того, щоб надати високопродуктивні послуги з обробки даних клієнтам. На сьогоднішній день існують такі популярні моделі обробки даних як обробка пакетних даних, обробка потокових даних, гібридна обробка даних та обробка даних графа.

- Обробка пакетних даних

Як було зазначено вище, пакетні дані є статичними та містять великий обсяг різних типів даних, тож для обробки таких даних використовується розподільний метод офлайн-обчислення, який також може виконувати паралельне обчислювання.

Обробка пакетних даних зберігає дані за допомогою HDFS та використовує такі механізми комунікації як RPC/HTTP. За швидкістю обробки дана модель повільна за наступні моделі та має хвилинний рівень обробки. Сценаріями, у яких застосовується дана ця модель, є автономний аналіз та обробка даних, а також масштабний пошук в Інтернеті.

Прикладом такої обробки даних є розподілена модель програмування MapReduce, яка була розроблена компанією Google у 2004 році. MapReduce дозволяє користувачам будувати складні обчислення на великих масивах даних, не турбуючись про синхронізацію, відмовостійкість, надійність чи доступність [34].

- Обробка потокових даних

Ця обробка застосовується для таких типів даних, які вимагають відповіді у реальному часі, тому швидкість обробки цієї моделі має мілісекундний рівень. Обробка потокових даних використовує чергу повідомлень як механізм комунікацій та направлена на аналіз та планування в реальному часі та безперервний розрахунок.

Найпопулярнішим фреймворком обробки є розподілена система Storm та Apache Samza. Storm – це розподілена система обробки потокових даних у реальному часі з відкритим кодом [41], яка надає високу відмовостійкість, надійність та масштабованість. Apache Samza — це структура потокової обробки, яка здатна ефективно обробляти великі обсяги даних [42], яка може обробляти мільйони повідомлень за секунду.

- Гібридна обробка даних

Дана модель об’єднує в собі як пакетну, так і потокову обробку даних, тим самим виконуючи баланс між часом обробки даних та їх обсягом. Гібридна обробка даних має таку саму швидкість обробки як і потокова — мілісекундний рівень. На відміну від інших моделей, гібридна використовує спільну пам’ять або чергу повідомлень для комунікації та спрямована на інтерактивне машинне навчання та поступове обчислення. Для зберігання даних модель використовує пам’ять чи диск.

Apache Spark – це нова платформа пакетної обробки даних із можливостями потокової обробки даних [43]. Spark використовує метод під назвою Resilient Distributed Dataset, тим самим прискорюючи пакетну обробку даних, помістивши весь процес у пам’ять [44]. Компонент Spark Steaming гарно справляється з навантаженням, але не має такої продуктивності як потокова обробка.

Іншим прикладом є Flink – це фреймворк аналізу даних з відкритим кодом, яким керує Apache для пакетної та потокової обробки даних [45]. В порівняні з Apache Spark, цей фреймворк швидше обробляє дані та використовує постійну чергу повідомлень у різних точках потоків даних.

- Графова обробка даних

Модель використовується для графових даних з великою кількістю вершин та ребер. Зазвичай така модель використовується для карти соціальної мережі та аналізу дорожньої карти руху. Швидкість обробки даних має секундний рівень,

комунікація виконується за допомогою черг повідомлень та використовує розподілену файлову систему та бази даних для зберігання даних.

Pregel – це вершинно-центрична обчислювальна модель, запущена компанією Google у 2010 році, для визначення власних алгоритмів за допомогою визначеної користувачем обчислювальної функції.

Головною перевагою цього фреймворку є висока ефективність у роботі з ітеративним обчисленням графових даних, обчислення великих графічних даних та забезпечення високої продуктивності.

2.4 Аналіз і візуалізація даних

Аналіз даних потрібен для прогнозування та аналізу майбутніх тенденцій та моделей, і це передостанній крок перед тим, як інформація буде представлена споживачам послуг передачі даних.

Для поглибленого аналізу великих даних застосовують машинне навчання – це область досліджень, що зосереджується на теоріях, системах навчання та атрибутах алгоритмів, що включає штучний інтелект, теорію інформації, оптимальне керування, когнітивну науку, математику та інженерію, інтелектуальний аналіз даних, системи керування, системи ідентифікації, інформатику тощо [46].

Як вже було зазначено, після аналізу даних слідує їх віртуалізація, яка представляє оброблені дані користувачам за допомогою таблиць, діаграм, зображень тощо. Через те, що люди часто привертають увагу на яскраві речі, у віртуалізації також застосовуються яскраві ефекти, які зацікавлять користувача та підвищать переконливість та зрозумілість даних.

За допомогою візуалізації даних аналітики даних використовують тенденції даних, шаблони даних, зв’язки та іншу інформацію, щоб глибше вивчати дані з різних вимірів, щоб ще більше покращити аналіз даних. Інструменти візуалізації даних широко використовувалися, в основному включаючи інструменти для створення діаграм (RawGraphs, Google Charts тощо);

засоби формування класів карт (Modest Maps, Openheatmap, ColorBrewer тощо);
засоби формування шкали часу (Cube, Timeflow, Dipity тощо).

3 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХМАРНИХ СЕРВІСІВ ДЛЯ СИНХРОНІЗАЦІЇ ДАНИХ

3.1 Загальні відомості та принцип синхронізації даних

Користувачі все більше залежать від електронної інформації, яка обробляється та зберігається на безлічі комп'ютерів і пристроїв зберігання даних, з'єднаних дротовими та бездротовими мережами. До найпоширеніших сервісів хмарного зберігання даних можна віднести Dropbox, Google Drive і Microsoft OneDrive, які надають користувачам зручний і надійний спосіб зберігати та ділитися даними не залежно від розташування, типу пристрою та часу.

Всі дані користувачів, які зберігаються на хмарному сховищі (документи, зображення тощо), автоматично синхронізуються на всіх пристроях, такі як ПК, смартфон та інші, як одночасно підключені до хмари (наприклад, ПК, планшетах і смартфонах), підключених до хмари вчасно. Ключовою операцією хмарних сервісів є синхронізація даних, яка автоматично відображає зміни в локальних файлових системах користувачів у хмарі за допомогою низки мережевих з'єднань.

Синхронізація даних – це техніка підтримки узгодженості між даними від джерела, розташованого в одному місці, до цілі, розташованої в іншому місці, і навпаки, узгодження даних з часом [20] та збереження кількох копій набору даних в узгодженості одна з одною. Зазвичай у хмарних сервісах забезпечення синхронізації відносять до підтримки цілісності даних.

Звернемось до рис. 3.1, який демонструє принцип синхронізації даних. Дана архітектура має три основні компоненти [21]: клієнт, сервер керування та сервер зберігання даних.

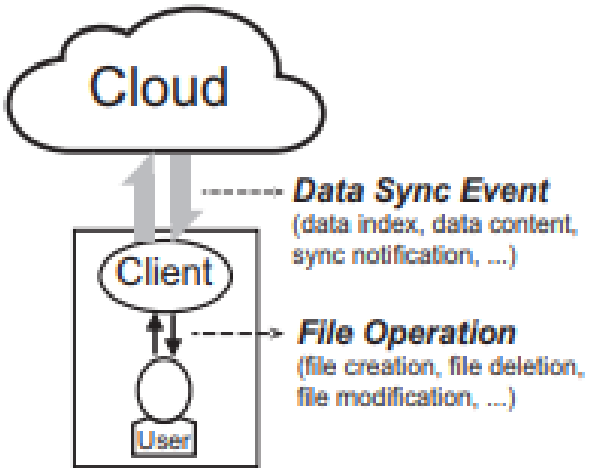

Рисунок 3.1 – Принцип синхронізації даних

Як правило, у хмарній службі зберігання користувач має локальну папку, яка також має назву «папка синхронізації», у якій кожна операція з файлами, яку виконує користувач, синхронізується з хмарою, розробленою постачальником послуг. Синхронізація файлу передбачає, що клієнт розбиває зміст файлу на фрагменти та індексує їх для створення метаданих. Слід зазначити, що кожна подія синхронізації даних тягне за собою мережевий трафік.

Абстракція файлової системи на стороні сервера відрізняється від клієнтської, оскільки на стороні сервера метадані та зміст файлів користувача розділяються та зберігаються на відповідних серверах. Під час процесу синхронізації метадані обмінюються з сервером керування через потік інформації метаданих, а вміст передається через потік зберігання даних.

Не менш важливим є потік сповіщень, що надає користувачу інформацію щодо стану оновлення змін з інших пристроїв у хмарі.

Слід також зазначити такі основні можливості синхронізації даних, як фрагментація, що може розділити вміст на блоки даних; групування, яке поєднує фрагменти та надає можливість їх передавати; дедуплікація, яка забезпечує уникнення повторень передачі вмісту та блоків, які вже існують у хмарі; дельта-кодування, тобто передача не всього файлу, а лише зміненої частини; стиснення.

3.2 Методи синхронізації

Механізми синхронізації даних розрізняють як завантаження та звантаження. Завантаження – це процес передачі даних з мобільні програми до віддаленої системи, а звантаження – це зворотний процес, тобто передача даних з віддаленого сервера до мобільної програми.

Зазвичай синхронізацію даних поділяють на синхронізацію впорядкових та невпорядкових, і якщо з першим типом даних все зрозуміло, то другий – викликає проблему узгодження множин. В залежності від обраної розподіленої моделі було розроблено декілька рішень, що розв'язують цю проблему [22].

- Комплексна синхронізація

Даний метод синхронізації є достатньо застарілим, але має легку реалізацію, а також гарантує передачу змін даних з одного пристрою на інший [22]. Під час синхронізації один з пристроїв передає всі дані, які він має, на інший пристрій для локального порівняння. Це, як правило, дуже неефективно, оскільки в більшості випадків до даних вносяться лише деякі зміни, а всі дані надсилаються через мережу для порівняння.

- Синхронізація за прапором стану

При цій синхронізації клієнт зберігає інформацію про дані у вигляді прапорців стану, які зазначають, чи був елемент створений, змінений або видалений. Цей метод є більш ефективним ніж комплексна синхронізація, оскільки під час синхронізації користувач може надіслати елементи, на яких встановлено прапорці. Недоліком є те, що при синхронізації з декількома користувачами виникає проблема, що окрім запису про змінені дані, треба зберігати інформацію про інших користувачів, з якими відбувалась синхронізація [23].

- Синхронізація за мітками часу

При використанні синхронізації на основі часових міток, кожен клієнт зберігає інформацію про час останньої зміни даних, а також часову мітку для

останньої синхронізації, що містить позначку часу та інформацію про те, коли дані були вставлені, змінені або видалені. Під час процесу синхронізації надсилаються лише ті елементи, які було змінено з моменту останньої синхронізації, що є більш зручним методом, ніж синхронізація за прапором стану, оскільки не потребується зберігати інформацію про те, які саме зміни були синхронізовні з тим чи іншим користувачем.

Проблема в методі синхронізації за мітками часу виникає в обставинах, коли два клієнти повністю синхронізовані з іншими користувачами, але синхронізуються один з одним вперше. Це викликає ситуацію, коли обидва користувачі надсилатимуть усі дані для синхронізації, хоча між ними немає жодних змін [19].

- Математична синхронізація

Сутність методу полягає у використовування математичних властивостей даних, які потрібно синхронізувати. Мінскі та Трахтенберг [26] використовують характеристичні поліноми для представлення своїх наборів даних, як запропоновано в [24], а потім надсилають достатньо даних для реконструкції полінома та знаходження змін. Оскільки математична синхронізація з використанням характеристичних поліномів вимагає знання кількості змін заздалегідь, Мінскі та Трахтенберг запропонували використовувати ймовірнісну схему для визначення верхньої межі кількості змін, яка вимагає надсилання декількох запитів, щоб визначити всі зміни з дуже високою ймовірністю [23, 25, 26].

- Синхронізація на основі дайджесту повідомлень

Цей метод є ще однією формою математичної синхронізації, рішення якої не залежить від особливостей конкретної бази даних постачальника і використовує лише стандартні операції SQL.

Дайджест повідомлення – це числове представлення вмісту повідомлення фіксованого розміру, обчислене за допомогою геш-функції, який може бути зашифрований, формуючи електронний підпис [27].

Слід зазначити, що рішення синхронізації на основі дайджесту повідомлень вимагає декількох таблиць на сервері та сильно залежить від реляційної моделі бази даних, використовуючи зовнішні ключі та об'єднання JOINs. Крім того, вони мають по одній таблиці для кожного клієнта на сервері. [25]

- Синхронізація журналів

Синхронізація працює за схемою виконання кожної зміни як транзакції та зберігання її в журналі, які потім синхронізуються з іншими користувачами. Зазвичай, цей механізм синхронізації використовується в базах даних та в рішеннях для контролю вихідного коду. Кожна операція з даними передається іншому клієнту після завершення процесу синхронізації журналу. Журнали, що містять зміни, можуть значно зростати, оскільки вони зберігають всі операції на додаток до звичайних даних.

3.3 Інструменти сінхронізації

В даному підрозділі розглядаються такі основні інструменти синхронізації як Remote sync, Unison, Syncany та Dropbox. Вони зазнали найбільшої популярності серед інших інструментів та активно використовуються у сьогоденні. Таблиця 3.1 показує основні можливості та недоліки інструментів синхронізації даних, а також зазначає їх техніки та модифікації.

Таблиця 3.1 – Порівняння інструментів синхронізації

Інструмент синхронізації даних	Техніка синхронізації даних	Виявлення модифікації	Особливості	Недоліки
Rsync	Зіставлення контрольної суми, однорангова, постійна контрольна сума та контрольна сума MD5	На основі часу модифікації файлу	Стиснення файлу перед передачею	Синхронізація файлів лише між двома хостами

Продовження таблиці 3.1 – Порівняння інструментів синхронізації

Інструмент синхронізації даних	Техніка синхронізації даних	Виявлення модифікації	Особливості	Недоліки
Syncany	Зіставлення контрольної суми, метод фрагментації та дедуплікації	Техніка дедуплікації з кількома фрагментами	Підтримуйте версію, оптимізуйте пропускну здатність, мінімізуйте розмір сховища за допомогою дедуплікації	Синхронізація файлів лише між двома хостами
Unison	Використовує двонаправлений алгоритм синхронізації	Виявлення оновлення та друга звірка	Двонаправлена синхронізація	Потужність обробки потрібна з обох сторін
Dropbox	Подрібнювальний алгоритм	Порівняння двійкових файлів відмінностей	Підтримування версій	Небезпечний обмін файлами

3.3.1 Remote sync

Remote sync (rsync) – це програмний інструмент з відкритим кодом, який спочатку був написаний для систем UNIX, але в сьогоденні також працює на платформах Windows, Mac і Linux.

Спираючісь на дані про розмір файлу та дату зміни, програма визначає стан змін файлу. Синхронізація визнає вставлені або додані дані, видалені дані, а

також дані переміщення з мінімальними витратами на передачу, що приводить до висновку, що Rsync надає краще рішення, ніж інші методи синхронізації файлів.

Rsync створює списки файлів та блоків локальної та віддаленої файлової системи та порівнює їх, а потім надсилає змінені блоки даних одним пакетом у кожному напрямку [29].

Деякі функції rsync включають [28]:

- оновлення цілих дерев каталогів і файлових систем;
- додаткове зберігання символічних та жорстких посилань, право власності на файли, дозволи, пристрої та час;
- не вимагає спеціальних привілеїв для встановлення;
- внутрішній конвеєр зменшує затримку для кількох файлів;
- підтримує анонімний rsync, який ідеально підходить для дзеркального відображення.

3.3.2 Unison

Unison – це ще один файл-синхронізатор для Unix і Windows, який використовує двонаправлений алгоритм синхронізації, який дуже схожий на Rsync, але використовує двонаправлений алгоритм синхронізації.

Зазвичай для синхронізації файлів використовуються два методи; один – для оновлення виявлення, а другий – узгодження. Першочергово Unison звернутися до часу модифікації файлу та номерів коренів файлів, щоб потім зазначити оновлення. Він аналізує попередній запис синхронізації та при виявлені будь-якої зміни властивості файлу позначає цей файл як брудний.

Unison пропонує кілька переваг перед різними методами синхронізації, такими як Coda, rsync, Intellisync тощо. Unison може працювати та синхронізуватись між Windows і багатьма платформами UNIX. Unison не потребує привілеїв адміністратора, доступу до системи чи змін ядра для роботи. Unison може синхронізувати зміни файлів і каталогів в обох напрямках, на одній машині або в мережі за допомогою ssh або прямого з’єднання через сокет.

3.3.3 Syncany

Syncany також є програмою для хмарного зберігання та обміном файлами з відкритим кодом, яка дозволяє користувачам створювати резервні копії та ділитися певними папками своїх робочих станцій, використовуючи будь-які типи сховищ.

За допомогою техніки усунення дублікатів Syncany намагається зменшити зберігання та загальний час синхронізації, а при використанні техніки кількох блоків – намагається зменшити кількість запитів, що надходять на сервер, та вхідний розмір файлу перед завантаженням на сервер [30]. Таким чином через уникнення дублікатів зменшується використання диска на віддалених пристроях.

Недоліком цього методу є те, що він дуже інтенсивно використовує ЦП. Основна проблема, виявлена аналізом синхронізації, полягає в пошуку відповідного алгоритму багатокомпонентного [30] та дедуплікації [30]. У цій техніці він намагається знайти повторювані послідовності байтів (фрагментів) і зберігає лише одну копію цієї послідовності. Якщо фрагмент даних знову з’являється в тому самому файлі (або в іншому файлі того самого набору файлів даних), тоді дедуплікація зберігає лише посилання на цей фрагмент замість його точного вмісту.

3.3.4 Dropbox

Dropbox є поширеним хмарним сервісом для зберігання та синхронізації даних і має широкий вибір функцій. Він надає користувачам зручний спосіб зберігання, доступу і спільної роботи з файлами та даними через Інтернет.

Технологія синхронізації, використовна в Dropbox, дозволяє автоматично оновлювати файли та дані на всіх підключених пристроях, що підтримують Dropbox. При зміні або додаванні файлів на одному пристрої, ці зміни автоматично відображаються на всіх інших підключених пристроях. Це забезпечує єдність даних і зручність в роботі з файлами.

Dropbox надає клієнтське програмне забезпечення для Apple MacOS, Microsoft Windows X, Linux, для Google AndroidOS, Apple iOS і BlackberryOS та інших веб-браузерів [39]. Цей сервіс підтримує різноманітні типи даних, такі як фотографія, документи, відео та інші.

Одна із переваг цього сервісу це простота його використовування, надійність та безпека. Усі дані, які зберігаються та синхронізуються у Dropbox зберігаються у вигляді резервних копій, захищаються шифруванням та механізмами безпеки, що забезпечують конфіденційність і цілісність даних.

4 МОДЕЛІ ЗАГРОЗ, ПОРУШНИКА ТА БЕЗПЕКИ ХМАРНИХ СЕРВІСІВ

4.1 Моделі загроз

Хмарні сервіси зберігають великий обсяг даних, який з кожним днем тільки зростає, а клієнти все більше довіряють особисту інформацію сервісам та очікують отримати відповідний рівень безпеки даних. На жаль, кількість загроз та атак на хмарні сервіси так само зростає, як і рівень безпеки, тому користувачам та постачальникам хмарних послуг варто розуміти які загрози можуть порушити цілісність, конфіденційність та доступність даних, що зберігаються у сервісах даних.

В деяких випадках загроза може не мати серйозних наслідків, але більшість часу загрози мають серйозний характер, наприклад, крадіжка даних, розміщення даних на сторонніх джерелах, зміна особистих даних та інше. Також слід зазначити загрози, які з'являються і наслідок людської помилки, такі як слабкий рівень аутентифікації, невідповідальне ставлення до особистого телефона або іншої техніки тощо.

Розуміння різних видів загроз та вразливостей становить головним механізмом безпеки в хмарних сервісах. Моделювання загроз – це структурований спосіб, за допомогою якого можна ідентифікувати всі можливі ризики та загрози, пов’язані з конкретною мережею; а також може запропонувати деякі контрзаходи для їх запобігання [47]. Моделювання загроз допомагає розробляти нові та вдосконалювати існуючі безпечні системи для забезпечення безпеки даних у хмарних сервісах.

4.1.1 Процес моделювання загроз

Моделювання загроз є довгим процесом, оскільки для ретельного дослідження загроз не достатньо однієї спроби. Також треба враховувати, що всі

програми та сервіси часто оновлюються та змінюється, що також вимагає довгого процесу моделювання загроз.

Весь процес загроз можна поділити на наступні шість етапів:

1) Оцінка даних. Цей етап передбачає аналіз системи та визначення найцінніших даних, якій треба надати захист від пошкодження.

2) Створення огляду архітектури. Кожний сервіс має особисту архітектуру та систему, що додає складностей для моделювання загроз, тому аналіз сервісу, процес його роботи, потік даних і мережі є важливою складовою моделювання.

3) Декомпозиція програми. Етап спрямований на створення політику безпеки для системи. Декомпозиція містить фрагмент системи, для якої проводиться моделювання загроз, з інфраструктурою, базовим обладнанням та мережею. Політика безпеки має визначати вразливості, які можуть з’явитися при розробці системи.

4) Визначення загроз. Спираючись на попередній аналіз системи, її архітектури та політики безпеки починається визначення загроз.

5) Документація загроз. Всі виявлені загрози мають бути проаналізовані за задокументовані для подальшого їх використання. Це є обов’язковим етапом, який допомагає у подальшому етапі моделювання.

6) Оцінка загроз. На цьому етапі усі виявлені та задокументовані загрози розподіляються за шкалою від найвищого до найнижчого впливу на систему.

4.1.2 Загрози для користувачів хмарного сервісу

- Неоднозначність відповідальності

Ресурси, які використовуються користувачами хмарних послуг, надаються через моделі обслуговування, такі як програмне забезпечення як послуга, платформа як послуга, інфраструктура як послуга та інші. Між користувачами та провайдерами послуг має бути чітко визначена відповідальність, а її відсутність може призвести до концептуальних конфліктів або інцидентів як з боку користувача, так і збоку постачальника послуг.

- Ризик прив’язки до постачальника послуг

Ця загроза може виникнути при переході від власної системи до хмарних сервісів, а також між двома різними хмарними сервісами. Під час міграції частина даних може бути втрачена через різні моделі обслуговування. Таким чином, коли компанія переходить від однієї моделі до іншої, вона надає контроль постачальникам хмар. В ідеальній моделі міграція додатків від одного хмарного провайдера до іншого повинна бути простою, що є ще одним викликом для додатків хмарних обчислень, але оскільки кожен хмарний провайдер використовує окрему стандартну мову для своїх систем, це наразі неможливо [48].

- Втрата довіри

Для користувачів хмарними сервісами існує загроза безпеці через відсутність прозорого доступу до інформації щодо рівня безпеки сервісу. Відсутність способів отримати та поділитися рівнем безпеки постачальника у формалізований спосіб спричиняє появу загрози через втрату довіри [49].

- Незахищений доступ користувача хмарної служби

Дуже поширена загроза для користувачів, при якій зловмисник має доступ до особистих даних та може змінювати, видаляти, додавати дані користувача. Користувачі часто легковажно ставляться до важливості безпеки особистих даних і такі дії, як повторне використання пароля, ненадійний рівень аутентифікації, призводять до втрати доступу до даних. Зловмисники використовують різні методи атак для того, щоб отримати несанкціонований доступ до даних іншої людини. Особливо популярними є фішінгові атаки, використання вразливостей програмного забезпечення та шахрайство.

- Відсутність інформації/управління активами

Підписуючись на послуги хмарних обчислень, користувачі хмарних служб дуже стурбовані відсутністю інформації від хмарних провайдерів щодо керування даними/активами, наприклад розташування конфіденційних активів/даних, відсутність управління фізичним зберіганням даних, надійність резервного копіювання тощо.

- Втрата та витік даних

Втрата криптографічних ключів або кодів доступу створює серйозні проблеми для користувачів хмарних сервісів. Тому недостатня кількість криптографічних елементів, таких як ключі шифрування, коди авторизації та права доступу, може призвести до значних витрат через втрату або розкриття даних [49].

4.1.3 Загрози для постачальників хмарних послуг

- Невідповідність захисту

Через децентралізовану архітектуру хмарної інфраструктури її механізми захисту, ймовірно, будуть неузгодженими між розподіленими модулями безпеки [49]. Таким чином, зловмисник може скористуватися ситуацією та провести атаку порушити конфіденційність та цілісність.

- Еволюційні ризики

При переході з фази проєктування у фазу виконання рівень безпеки відрізняється. Саме тому, коли під час виконання системи буде проводиться реалізація залежного компонента системи може бути виявлено вразливості, які не були виявлені про проєктування.

- Переривання діяльності

Переривання надання послуг впиває на доступність та може призвести до її втрати, оскільки від набір послуг (апаратне забезпечення, додатки, платформа) залежать усі робочі процеси та інфраструктура.

- Ліцензійні ризики

В залежності від кількості встановлень та кількості користувачів постачальник має придбати певну кількість ліцензій для програмного забезпечення. Управління кількостями ліцензій не таке гнучке, як управлянні ресурсами хмарних сервісів, тому постачальнику важко передбачити яку кількість ліцензій йому потрібна, а це може викликати конфлікти використання програмного забезпечення.

- Погана інтеграція

Ця загроза стосується не тільки користувачів, а й постачальників хмарних послуг, оскільки перехід між хмарами завдає шкоди даним та змін конфігурації. При поганій інтеграції можуть уникнути багато функціональних та нефункціональних наслідків, що може ускладнити роботу організації.

- API незахищеного адміністрування

Сутність загрози полягає в тому, що при використанні API користувачі не завжди дотримуються правил введення особистих даних та аутентифікації. Через це API з недостатнім рівнем безпеки більш чутливими для дій зловмисників.

- Помилка ізоляції гіпервізора

Гіпервізор – це функція, яка абстрагує ОС і програми від основного апаратного забезпечення комп’ютера [50]. Кілька віртуальних машин, розміщених спільно на одному фізичному сервері, спільно використовують ресурси процесора та пам’яті, віртуалізовані гіпервізором [49]. Такі умови є сприятливими для загрози, яка може призвести до незаконного доступу до пам’яті віртуальних машин.

- Недоступність послуги

DDoS-атаки націлені на веб-сайти та сервери, порушуючи роботу мережевих сервісів з метою виснаження ресурсів програми [48]. Зловмисники, які стоять за цими атаками, наводнюють сайт несанкціонованим трафіком, що призводить до погіршення функціональності сайту або взагалі виводить його з ладу [51].

- Зловживання правом постачальника хмарних послуг

Організації, які надають послуги, не застраховані від внутрішньої загрози від злочинця-постачальника. При використанні хмарних сервісів користувач довіряє свої дані у постачальнику хмарних сервісів, таким чином з’являється ризик, що провайдер, який має зловмисні наміри, може використати чутливі дані користувача у своїх корисливих цілях.

4.2 Модель порушника

Важко надати точне визначення такому поняттю, як модель порушника, але якщо розглядати його абстрактно, можна сказати, що це опис дій порушника, який надає інформацію щодо його можливостей. Мета побудови моделі порушника – допомогти забезпечити захист інформації та ресурсів, що знаходяться в хмарних сервісах, шляхом аналізу зловмисників.

Існує безліч факторів, які впливають на опис дій порушника, що завдає складнощів для точної побудови його моделі. В залежності від моделі обслуговування, моделі послуг, рівню контролю інформації та власника інформації буде сформовано багато різних моделей порушника. В даному розділі буде розглянуто основні моделі, за якими можна класифікувати дії порушника.

У [52] надається шість категорій для визначення профілю порушника: категорія осів, характер дій, рівень доступу та можливостей, методи та засоби, мета дій порушника. Кожна з цих категорій має свої рівні та особливості, які буде розглянуто нижче.

За категорією осіб порушники розділяються на внутрішніх та зовнішніх. До внутрішніх порушників відносяться особи, які мають відношення до організації, наприклад працівники провайдера хмарних послуг, працівники користувача та сторонні особи. Інші порушники, які не мають прямого відношення до організації відносяться до зовнішніх.

За характером дій порушників поділяють на випадкових, пасивних та активних [53]. Під випадковим порушником мається на увазі користувач, якій без злого замислу випадково отримав доступ до об'єкта захисту та зробив над ним дію, тим самим порушуючи безпеку даних. Пасивний порушник спеціально порушив політику безпеки даних, але, на відміну від активних порушників, не зробив дії над об’єктом. Активним порушником називають того, хто показово порушає безпеку даних з ціллю несанкціонованого управління доступом.

За рівнем доступу та можливостей порушників поділяють на 4 рівні [54]. Перший рівень має найменшу кількість можливостей в порівнянні з іншими, а саме запуск фіксованого набору програм, що реалізують відомі функції обробки інформації. Другий рівень доповнюється можливістю створення нових програм та нових функцій обробки інформації. Третій рівень має вплив за базове програмне забезпечення. Четвертий рівень має найбільше кількість доступу та можливостей – проєктування та реалізація апаратних компонентів ІТС, додавання нових функцій обробки даних.

За рівнем ознайомленості порушники можуть бути користувачем, спеціалістом, адміністратором системи, спеціалістом з найвищим рівнем знань або адміністратором безпеки. В залежності від рівня порушник має відповідні особисті можливості та знання.

За методом та засобом, що використовує порушник, відрізняють агентурні (отримання повідомлень), штатні, пасивні (технічні засоби) та активні (засоби, що мають вплив на АС) методи та засоби.

За метою і дією порушники поділяються на такі основні типи:

- зловмисні дії над конфіденційною інформацією шляхом отримання доступу до ресурсів;

- заволодіння доступу особи, яка є персоналом або користувачем, для подальшого користування їхніми правами доступу до ресурсів;

- установка технічних, програмних та інших засобів для викрадення, змінення, зниження, додавання інформації.

Цей список можна продовжувати довго, оскільки кожен порушник має свої особисті цілі, які важко передбачити. Даний критерій є найскладнішим для визначення, бо мислення порушників непередбачене і навіть з всім описом дій порушників не можна гарантувати, що не з’являть нові порушники з новими ідеями.

4.3 Модель безпеки

Модель безпеки у хмарних сервісах – це набір політик, процедур, технологій та інструментів, що застосовуються для захисту даних, ресурсів та інфраструктури, що знаходяться в хмарних середовищах. Вона охоплює різні аспекти безпеки, такі як конфіденційність, цілісність, доступність, автентифікацію та авторизацію, та невідмовність [11], які спрямовані на захист інформації від загроз та атак.

Конфіденційність забезпечує запобігання розголошення інформації третім, неавторизованим, користувачам, організаціям чи процесам. Звертаючись до першого розділу, де описувались моделі хмар, зазначимо, що саме приватна хмара забезпечує конфіденційність, в той момент, коли у публічній хмарі є проблеми з відстежуванням правил автентифікації, що призводить до порушення конфіденційності.

Найкращім шляхом реалізацій конфіденційності є шифрування. Використання сильних шифрувальних алгоритмів забезпечує, що дані перебувають у зашифрованому вигляді під час передачі та зберігання. Тільки авторизовані особи, які мають правильний ключ розшифрування, можуть отримати доступ до розшифрованих даних.

Цілісність визначає точність, надійність та узгодженість даних, гарантуючи, що дані не були змінені третьою особою. [56] розглядають методи забезпечення цілісності даних у хмарному сховищі, які охоплюють контроль цілісності даних за допомогою шифрування, захист від несанкціонованої зміни ресурсів та інше.

Доступність забезпечує, що користувач має доступ та можливість використовувати обширні можливості сервісу. Вдосконалити доступність можна за допомогою надання резервних копій користувачу. Таким чином користувач може використовувати дані, збережені з попереднього запиту не зв’язуючись з конкретним сервером. Другим способом забезпечення доступності є передача

даних на сервер, який має зміст клонованого сервера, таким чином користувач не втрачає інформацію.

Невідмовність – це метод безпеки, що забезпечує доказову неможливість відмови від здійснених дій або транзакцій з боку відправника чи одержувача. Відрізняють два типи відмов: відмова від призначення повідомлення та відмова від джерела повідомлення. Існують такі основні методи забезпечення невідмовності, як цифровий підпис, дозволяючи підтвердити автентичність документа або повідомлення, а також ідентифікувати особу, яка його створила; захист від викривання приватного ключа за допомогою шифрування та безпечних протоколів, оскільки втрата або компрометація особистого ключа може позбавити гарантій невідмовності.

Автентифікація забезпечує коректність доступу до даних користувача, за допомогою різних методів аутентифікації. Основна мета автентифікації – запобігти несанкціонованому доступу до даних та ресурсів, а також забезпечити взаємодію з відповідними користувачами.

Відрізняють п’ять головних методів автентифікації [57]:

1) Автентифікація паролем. Цей метод є одним із найпоширеніших через свою простоту у використанні, але й найбільш ненадійним. Для створення пароля користувачі використовують цифри, літери, спеціальні символи та складають з них комбінацію. На жаль, найчастіше користувачі створюють пароль спираючись на факти особистого життя (дата народження, ім’я домашніх тварин тощо), оскільки такий пароль легко запам’ятати. Іншою проблемою стає те, що користувачі рідко оновлюють паролі та часто використовують одні й ті самі комбінації для різних сервісів.

2) Багатофакторна автентифікація. Цей метод передбачає використання двох або більше факторів для підтвердження ідентичності користувача, наприклад, використання згенерованих кодів. Багатофакторна автентифікація є більш надійною, ніж автентифікація паролем, але залишається вразливою.

3) Автентифікація на основі сертифіката. Використання цифрових сертифікатів та ключів доступу для ідентифікації та автентифікації користувачів. Користувачі мають унікальні сертифікати або ключі, які підтверджують їх ідентичність під час взаємодії з хмарним сервісом.

4) Біометрична автентифікація. Цей метод базується на унікальних біологічних характеристиках люди, такі як обличчя, відбиток пальця, сітківка ока та голос. Цей метод є одним з найнадійніших, оскільки біометричні дані можна порівнювати вже з авторизованими користувачами, а також цей метод можна поєднувати з іншими методами автентифікації для покращення безпеки даних.

5) Автентифікація на основі маркерів. Метод використовується для підтвердження ідентичності користувача при доступі до ресурсів або послуг в хмарних сервісах.

Автентифікація на основі маркерів (також відома як токен-базова автентифікація) є методом безпеки, який використовується для підтвердження ідентичності користувача при доступі до ресурсів або послуг в хмарних сервісах. Користувач отримує маркер або токен, який представляє його ідентичність після успішної авторизації.

Моделі безпеки, що базуються на ЦРУ є найбільш використовуваними для забезпечення інформаційної безпеки. Як і будь-яка система безпеки вона має свої переваги та вразливості, але через її універсальність саме вона була розглянута у цьому підрозділі.

ВИСНОВКИ

У результаті роботи було досягнуто мети аналізу та дослідження синхронізації та зберігання даних ху хмарних сервісах. В першу чергу ми детально ознайомились з поняттям хмарних обчислень та ознайомились з їх моделями, які поділяються на моделі розгортання та моделі обслуговування.

Моделі розгортання бувають приватними, публічними, гібридними та загальними та відрізняються людиною чи організацією, яка може належати, управлятись та експлуатувати їх. Моделі розгортання, в свою чергу, бувають інфраструктура як послуга, платформа як послуга та програмне забезпечення як послуга і відрізняються вони за рівнем надання послуг.

При дослідженні досвіду використання хмарних сервісів було виявлено, що найбільш популярними є модель розгортання сервіс як послуга. Наразі обсяг ринку хмарних сервісів у світі вже вкладає понад 309 млрд. Доларів і український ринок також набирає помітної популярності.

Проаналізувавши різні сфери використання хмарних сервісів, такі як бізнес та освіта, було виявлено найпопулярніші сервіси, такі як Google Apps for Work, Microsoft Office 365 та Zoho Docs.

За аналізом даних, які зберігаються у хмарних сервісах, обсяг створених даних складає 175 зетабайт. У результаті дослідження були розглянуті системи для обробки потокових та пакетних даних. Оскільки ці дані відрізняються за статичністю та динамічністю, до них не можна використовувати однакові системи обробки. Для динамічних потокових даних використовуються такі системи як Flume та Kafka, а для статичних пакетних даних — ETL.

Для аналізу зберігання даних було розглянуто нереляційну базу даних NoSQL та реляційну NewSQL та наведено основі приклади NoSQL баз даних, такі як Redis, MongoDB, Hbase та інші та їх переваги та недоліки.

У третьому розділі, який присвячений аналізу та дослідженню синхронізації даних, було зазначено принцип синхронізації та її основні можливості. Було розглянуто такі методи синхронізації: комплексна синхронізація, синхронізація за прапором стану, синхронізація за мітками часу, математична синхронізація, синхронізація на основі дайджест повідомлення, синхронізація журналів.

Важливою частиною огляду синхронізації є огляд інструментів для синхронізації в хмарних сервісах. Було розглянуто такі головні інструменти Rsync, Unison, Syncany та Dropbox, проведено їх порівняльну характеристику. Деякі переваги та недоліки та особливості роботи.

В останньому розділі дипломної роботи було проведено аналіз моделей загроз, порушника та безпеки хмарних сервісів. Процес моделювання загроз містить такі пункти, як оцінка даних, створення огляду архітектури, декомпозиція програм, визначення загроз, документація загроз та оцінка загроз. Також загрози можна поділити на загрози для користувачів та загрози для постачальників.

Розглянута модель порушника охоплює багато категорій: категорія осіб, характер дій, рівень доступу та можливостей, методи та засоби, мета дій порушника. Кожна з цих категорій була проаналізована та досліджена.

У моделі безпеки звертається увага на моделі ЦРУ та забезпечення безпеки за допомогою автентифікації. Конфіденційність, доступність та цілісність допомагають класифікувати загрози для ефективного протистояння їм. Модель тріади ЦРУ є найбільш універсальної та відомою у всьому світі, саме тому ці моделі були розглянути у роботі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mell P. and Grance T. The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*. Vol.53. No.6, 2009. P. 50
2. Nelson L. S. da Fonseca, Raouf. Cloud Services, Networking, and Management, 2015. P. 7-10
3. Goyal S. Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review. *International Journal of Computer Network and Information Security. Vol. 6, no. 3,* 2014. P. 20–29
4. Cloud Computing Architecture, URL: https://computer.howstuffworks.com/cloud-computing/cloud-computing1.htm (дата звернення: 150.03.2023)
5. Yashpalsinh Jadeja, Kirit Modi. Cloud Computing - Concepts, Architecture and Challenges. *Intenational Conference on Computing, Electronics and Electrical Technologies [ICCEET]*, 2012
6. Mell P. and Grance T. Effectively and securely using the cloud computing paradigm. *National Institute of Standards and Technology*, 2009
7. Comparison of Several Cloud Computing Platforms / Peng J et al. *2nd IEEE International Symposium on Information Science and Engineering,* 2009. p. 23-27
8. Enabling public verifiability and data dynamics for storage security in cloud computing / Wang Q. et al. *Computer Security–ESORICS*, 2009, p. 355-370
9. Rashid A., Chaturvedi A. Cloud Computing Characteristics and Services A Brief Review. *International Journal of Computer Sciences and Engineering*. 2019. Vol. 7, no. 2. P. 421–426.
10. Hype Cycle for Cloud Computing, URL: https://www.gartner.com/en/documents/1078112 (дата звернення: 19.03.2023)

11. ПЕРЕЇЖДЖАЄМО У ХМАРИ: ЯКІ ПЕРЕВАГИ ДЛЯ БІЗНЕСУ ВІД ХМАРНИХ СЕРВІСІВ, URL: https://hub.kyivstar.ua/news/pereyizhdzhayemo-u-hmary-yaki-perevagy-dlya-biznesu-vid-hmarnyh-servisiv/ (дата звернення: 19.03.2023)
12. Rashid A., Chaturvedi A. Cloud Computing Characteristics and Services A Brief Review. *International Journal of Computer Sciences and Engineering*. 2019. Vol. 7, no. 2. P. 421–426.
13. Coughlin T. 175 Zettabytes By 2025. *Forbes*. URL: https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/ (дата звернення: 19.03.2023)
14. *ISO/IEC 17788:2014*. (2015, January 6). ISO. URL: https://www.iso.org/standard/60544.html (дата звернення: 28.03.2023)
15. Berman, S., Kesterson-Townes L., Marshall A., Srivathsa R. The power of cloud. Driving business model innovation. *Executive Report - IBM Global Business Services*, 2012. P. 4-6
16. Business editions - Google Workspace Admin Help, URL: https://support.google.com/a/answer/13062337?hl=en, (дата звернення: 19.03.2023)
17. *Microsoft 365 Apps for Business - Microsoft 365*, URL: Ihttps://www.microsoft.com/en-us/Imicrosoft-365/business/microsoft-365-apps-for-business?activetab=pivot:overviewtab, (дата звернення: 19.03.2023)
18. The G2 on Zoho Docs, URL: https://www.g2.com/products/zoho-docs/reviews, (дата звернення: 10.05.2023)
19. Zoho | Cloud Software Suite for Businesses – Zoho, URL: https://www.zoho.com/, (Дата звернення: 10.05.2023)
20. Mohammad Faiz, Udai Shanker. Data Synchronization in Distributed Client-Server Applications. *2nd IEEE International Conference on Engineering and Technology (ICETECH)*, 2016.

21. Cui Y., Lai Z., Dai N. A first look at mobile cloud storage services: architecture, experimentation, and challenges. *IEEE Network*. 2016. Vol. 30, no. 4. P. 16–21.
22. Agarwal S., Starobinski D., Trachtenberg A. On the scalability of data synchronization protocols for PDAs and mobile devices. *IEEE Network*. 2002. Vol. 16, no. 4. P. 22–28.
23. Starobinski D., Trachtenberg A., Agarwal S. Efficient PDA synchronization. *IEEE Transactions on Mobile Computing*. 2003. Vol. 2, no. 1. P. 40–51.
24. RichardJ. Lipton. Efficient checking of computations. *STACS 90, volume 415 of Lecture Notes in Computer Science*, 1990.
25. A database synchronization algorithm for mobile devices / M.-Y. Choi et al. *IEEE Transactions on Consumer Electronics*. 2010. Vol. 56, no. 2. P. 392–398.
26. Minsky Y., Trachtenberg A., Zippel R. Set reconciliation with nearly optimal communication complexity. *IEEE Transactions on Information Theory*. 2003. Vol. 49, no. 9. P. 2213–2218.
27. IBM Documentation. URL: https://www.ibm.com/docs/en/ibm-mq/7.5?topic=concepts-message-digests, (дата звернення: 21.04.2023)
28. rsync features. URL: https://rsync.samba.org/features.html, (дата звернення: 25.04.2023)
29. Mahdizadeh, E.H., Yong Lee Kee, Li Kok Wei, Ghods, M.H. Pre-processing directory structure for improved RSYNC transfer performance. *Advanced Communication Technology (ICACT), 13th International Conference on , vol., no.,* 2011. P.1043-1048
30. Philipp C. Heckel. Minimizing remote storage usage and synchronization time using deduplication and multichunking: Syncany as an example. *Technical Report TR-CS-96-05, University of Mannheim,* 2012

31. Karunaratne P., Karunasekera S., Harwood A. Distributed stream clustering using micro-clusters on Apache Storm. *Journal of Parallel and Distributed Computing*. 2017. Vol. 108. P. 74–84.
32. Apache Flume. URL: http://flume.apache.org/ , (дата звернення: 21.04.2023)
33. Apache Kafka. URL: http://kafka.apache.org/, (дата звернення: 21.04.2023)
34. Big Data ETL Implementation Approaches: A Systematic Literature Review (P) / J. Nwokeji et al. *The 30th International Conference on Software Engineering and Knowledge Engineering*. 2018.
35. Relational databases as a massive information source for defeasible argumentation / C. A. D. Deagustini et al. *Knowledge-Based Systems*. 2013. Vol. 51. P. 93–109.
36. Ghemawat S., Gobioff H., Leung S. T. The Google File System. *Proceedings of the* Ghemawat S., Gobioff H., Leung S.-T. The Google file system. *the nineteenth ACM symposium*, Bolton Landing, 2003. P. 29-43
37. Yeh T., Chien T. Building a version control system in the Hadoop HDFS. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, 23–27 April 2018. 2018.
38. An efficient distributed caching for accessing small files in HDFS / K. Bok et al. *Cluster Computing*. 2017. Vol. 20, no. 4. P. 3579–3592.
39. *Sync files and folders - Dropbox*. URL: https://www.dropbox.com/en_GB/features/sync, (дата звернення: 21.04.2023)
40. Pavlo A., Aslett M. What's Really New with NewSQL?. *ACM SIGMOD Record*. 2016. Vol. 45, no. 2. P. 45–55.
41. Apache storm. URL: http://storm.apache.org/, (дата звернення: 02.05.2023)
42. Apache Samza. URL: http://samza.apache.org/, (дата звернення: 02.05.2023)
43. Apache Spark. URL: http://spark.apache.org/, (дата звернення: 02.05.2023)

44. Backhoff O., Ntoutsi E. Scalable Online-Offline Stream Clustering in Apache Spark. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, Spain, 12–15 December 2016.
45. Apache Flink. URL: https://flink.apache.org/, (дата звернення: 02.05.2023)
46. Blockchain-Based Systems and Applications: A Survey / Zhang J. et al. *Journal of Internet Technology, Vol. 21, No. 1*, 2020. P. 1-14
47. Kamatchi R., Ambekar K. Analyzing Impacts of Cloud Computing Threats in Attack based Classification Models. *Indian Journal of Science and Technology*. 2016. Vol. 9, no. 21. P. 3-4
48. Єсіна М, Кравченко А., Кравченко С. Огляд загроз безпеці та цілісності даних у хмарних обчисленнях. Подано до журналу *Радотехніка*. 2023.
49. VimalRosy J., Britto Ramesh Kumar S. Security Threats in Cloud Computing. *Asian Journal of Computer Science and Technology*. 2019. Vol. 8, S1. P. 80–83.
50. Posey B., Gillis A. S. Hypervisor. URL: https://www.techtarget.com/searchitoperations/definition/hypervisor, (дата звернення: 30.04.2023)
51. What is a DDoS attack? URL: https://www.microsoft.com/en-us/security/business/security-101/what-is-a-ddos-attack, (дата звернення: 14.04.2023)
52. Аулов І. Ф. Дослідження моделі загроз ключових аспектів хмари та пропозиції захисту від них. *Східно-Європейський журнал провідних технологій*. 2015. № 77.
53. Jansen W. Guidelines on Security and Privacy in Public Cloud Computing. *NIST Pub Series: Special Publication (NIST SP), 2011*
54. Jansen W. A. Cloud Hooks: Security and Privacy Issues in Cloud Computing. *2011 44th Hawaii International Conference on System Sciences (HICSS 2011)*, Kauai, HI, 4–7 January 2011. 2011.

55. Relational databases as a massive information source for defeasible argumentation / C. A. D. Deagustini et al. *Knowledge-Based Systems*. 2013. Vol. 51. P. 93–109.
56. Кравченко А., Кравченко С., Бобух В. Загрози цілісності даних та пропоновані рішення у хмарних обчисленнях. *IX-th International Scientific and Technical Conference,* May 25 – 26, 2023

ДОДАТОК А

УДК 004.056.5

М. В. ЄСІНА, канд. техн. наук, А.А. КРАВЧЕНКО, С.О. КРАВЧЕНКО

ОГЛЯД ЗАГРОЗ БЕЗПЕЦІ ТА ЦІЛІСНОСТІ ДАНИХ У ХМАРНИХ ОБЧИСЛЕННЯХ

Вступ

Хмарні обчислення – це швидко набираюча популярності та розвитку технологія, яка поєднує у собі декілька підходів та моделей з надання та управління IT сервісами. Згідно з визначенням Національного інституту стандартів і технологій (NIST) США, хмарні обчислення – це модель забезпечення повсюдного та зручного доступу на вимогу, через мережу до спільного пулу обчислювальних ресурсів, що підлягають налаштуванню (наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), і, які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера [1]. Хмарні обчислення розглядаються як одна з найуспішніших обчислювальних технологій, здатних вирішити цілу низку проблем, що стоять перед людством.

Хмарні обчислення мають декілька ключових особливостей, як то надійність, широкий мережевий доступ, масштабованість інфраструктури, гнучкість, незалежність від місця розташування, економія на масштабах і економічна ефективність та стійкість [2, 3].

Через зростаючу популярність і широку експлуатацію послуг хмарних обчислень виникає необхідність високого рівня безпеки. У сьогоднішніх реаліях люди використовують технології хмарних обчислень у великих обсягах, наприклад, на роботі, в особистих цілях та інше, так як вони мають велику довіру до цих технологій. Щоб запобігти втраті довіреної інформації провайдери послуг мають забезпечити її цілісність.

Стаття присвячена огляду загроз безпеці та цілісності технологій хмарних обчислень, тому що це є важливим аспектом даної технології. Зазначимо, що безпека хмарних обчислень означає захист даних, тоді як цілісність – їх надійність. Безпека та цілісність даних є основною проблемою користувачів, пов'язаною із хмарними обчисленнями.

Основна частина

[4] визначає безпеку хмарних обчислень як «Піддомен комп'ютерної безпеки, мережевої безпеки та, ширше, інформаційної безпеки. Це стосується широкого набору політик, технологій і елементів керування, які застосовуються для захисту даних, додатків і відповідної інфраструктури хмарних обчислень».

Коли організація вирішує зберігати дані або розміщувати додатки в публічній хмарі, вона втрачає можливість мати фізичний доступ до серверів, на яких зберігається її інформація [8]. Як наслідок, потенційно конфіденційні дані піддаються ризику інсайдерських атак. Згідно зі звітом Альянсу хмарної безпеки за 2010 рік [5], внутрішні атаки є однією з семи найбільших загроз у хмарних обчисленнях. Тому постачальники хмарних послуг повинні забезпечити проведення ретельних перевірок співробітників, які мають фізичний доступ до серверів у дата-центрі. Крім того, центри обробки даних рекомендується часто моніторити на предмет підозрілої активності. Існує чотири основні аспекти безпеки в хмарі, за які відповідають як постачальники, так і клієнти [6]:

- Обмеження доступу. Оскільки в хмарі всі ресурси доступні через Інтернет, дуже важливо переконатися, що лише належні користувачі матимуть доступ до потрібних їм інструментів протягом визначеного періоду часу.
- Захист даних. Організації повинні розуміти, де розташовано їхні ресурси, і застосувати відповідні елементи керування для захисту даних та інфраструктури, де вони розміщені.
- Відновлення даних. У разі порушення безпеки надзвичайно важливо мати надійне рішення для резервного копіювання даних і план їхнього відновлення.
- План реагування. У разі атак організацям потрібен спеціальний план, який дасть їм змогу зменшити наслідки та запобігти ураженню інших систем.

Загрози цілісності

Як і у будь-якій іншій системі, загрози безпеки для технологій хмарних обчислень можна поділити на загрози конфіденційності, цілісності та доступності. Загалом цілісність даних означає захист даних від несанкціонованого видалення, модифікації чи фальсифікації [10].

Предметом дослідження цієї статті є саме загрози цілісності, які розглянуто нижче.

1) Несанкціонований доступ

Ця атака направлена на безконтрольні зміни даних, на які не зможе впливати авторизований користувач. Зловмисник, який проводить несанкціонований доступ з послідовною зміною даних, може провести атаку зовні або зсередини організації-власника хмари. Це найсерйозніша атака, якщо це станеться, то витік даних відбувається шляхом використання старого обладнання та повторного використання драйверів [11].

2) Блокування даних

Ця загроза утворюється при переході від одного постачальника послуг до іншого. Так як різні постачальники надають різні послуги, тому при переході може трапитись втрата даних користувачів або їх блокування. У хмарі немає правил або умов щодо того, як зберігати дані, це залежить від постачальника хмарних послуг (CSP) [12]. Зазвичай, дані будуть розкидані по всьому серверу та системам. В ідеальній моделі міграція додатків від одного хмарного провайдера до іншого повинна бути простою, що є ще одним викликом для додатків хмарних обчислень, але оскільки кожен хмарний провайдер використовує окрему стандартну мову для своїх систем, це наразі неможливо.

3) SQL-ін'єкції

SQL-ін'єкції націлені на SQL-сервери, які запускають уразливі програми баз даних. Хакери використовують вразливі місця веб-серверів і вводять шкідливий код, щоб обійти вхід і отримати несанкціонований доступ до серверних баз даних. У разі успіху хакери можуть маніпулювати вмістом баз даних, отримати конфіденційні дані, віддалено виконувати системні команди або навіть взяти під контроль веб-сервер для подальшої злочинної діяльності [13].

4) Атака "людина посередині" (MiMA)

MiMA зазвичай виникає, коли різні користувачі хмари спілкуються один з одним або спільно використовують ресурси з хмарного середовища [14]. Недостатнє шифрування може зробити користувачів вразливими до атаки "людина посередині", яка є неприємною атакою [15]. TLS – криптографічний протокол, який дозволяє клієнт-серверному додатку [16] запобігти підслуховуванню будь-якої конфіденційної інформації, що відбувається на HTTPS, який використовує TLS. Якщо людина отримує доступ до невідомої мережі і виконує свою роботу в HTTP, зловмисник, який виступає в ролі посередника, потім скористається цим, перехопивши всі конфіденційні дані через HTTPS-пакети.

5) DDoS-атака

Мабуть,Що у сучасних технологіях це є найбільш серйозною проблемою, оскільки не може бути усуненою повністю. На сьогоднішній момент є лише деякі методи пом'якшення наслідків, які допомагають зменшити ризики та послідовності таких атак. DDoS-атаки націлені на веб-сайти та сервери, порушуючи роботу мережевих сервісів з метою виснаження ресурсів програми. Зловмисники, які стоять за цими атаками, наводнюють сайт несанкціонованим трафіком, що призводить до погіршення функціональності сайту або взагалі виводить його з ладу [17].

6) Атаки на автентифікацію

Атаки на автентифікацію складно класифікувати як саме атаки на цілісність, але вони можуть спровокувати такі загрози, тому слід їх зазначити. Нижче наведено декілька відомих атак на автентифікацію.

- Атака на відтворення

Ця атака відбувається, коли невідома особа переглядає трафік даних, а потім надсилає комунікаційні дані на своє місце, як оригінального відправника. Щоб запобігти цій атаці, зазвичай впроваджують мітки часу та порядкові номери [18].

- Атака грубої сили або атака за словником

Це базова атака, при якій зловмисник перебирає всі можливі комбінації для пароля, щоб отримати доступ до даних користувачів. Чим більше довжина пароля, тим більше часу знадобиться зловмиснику, щоб вгадати правильний пароль [19].

- Фішингова атака

Йдеться про те, як зловмисник перебирає всі можливі способи атаки на жертву, підбираючи всі комбінації коду та паролів. Знов-таки, чим складніший код, тим більше часу знадобиться зловмиснику для його підбору [20], при цьому витрачений час буде зростати не лінійно.

7) Атака відкату

Такі атаки можуть виникати під час оновлення системи у випадку, якщо постачальник у цей момент надає старе програмне забезпечення для користування. Це може спровокувати втрату даних, що зберігаються у цій системі. Відкат також відбувається без належного видалення старих даних користувача та оновлення системи до нової версії [21].

8) Атака підробки тегів

Ця атака відбувається, якщо нечемний продавець обманює своїх клієнтів, показуючи неправильний штрих-код чи даючи неправильне посилання. Якщо користувач сканує його на своєму пристрої, то зловмисник отримує доступ до всіх конфіденційних даних, що призводить до можливих ризиків шахрайства та витоку приватної інформації [21].

9) Візантійська атака

Ця атака відбувається на різні частини хмарних обчислень шляхом зупинки або виходу з ладу систем. Це станеться, коли запит буде некоректно проходить через систему [21].

10) Атака на систему доменних імен (DNS)

Ця атака відбувається, якщо систему атакує якесь шкідливе програмне забезпечення. DNS перетворює доменні імена на IP-адреси, і користувач не може бачити, наскільки правильно відбувається перетворення. Кожного разу, коли відкривається невідома веб-сторінка, зловмисник може легко отримати доступ до персональної інформації, що використовується на серверах [22].

11) Сніфферні атаки

Атака відбувається коли користувач натискає на деякі SOAP (Simple Object Access Protocol) – повідомлення або шкідливе посилання. Після того, як натиснуте посилання буде активоване, програма перехоплює потік пакетів в мережі і отримує доступ до персональних даних користувачів, таких як паролі, реквізити банківських рахунків тощо, які не є зашифрованими [23]. Залежно від обсягу даних та їх характеру, внаслідок цієї атаки втрати даних можуть бути непомітними або стати причиною великих проблем для цілої організації.

Методи забезпечення цілісності даних у хмарному середовищі

Проаналізувавши наведені вище загрози, перейдемо до методів забезпечення цілісності та запобігання атак на цілісність у хмарних сховищах. Як було зазначено вище, не завжди можна повністю усунути загрози, але майже завжди можна прийняти міри для їх запобігання та зменшення ймовірних втрат. Існує декілька механізмів та схем, які було запропоновано для захисту володіння даними та їх цілісності в середовищі хмарних обчислень. Нижче наведено декілька механізмів та схем, які ми розглянуті в попередніх дослідженнях [21].

1. Пом'якшення наслідків підробки тегів та атак витоку даних

Щоб запобігти цій атаці, існує схема, запропонована Yun Zhu та ін. [24], відома як Cooperative Provable Data Possession (CPCP), яка використовується в поєднанні з двома іншими (Homomorphic Verifiable Response та Hash Index Hierarchy), що забезпечує прозору перевірку даних та надійний захист.

Сутність даного методу виглядає так: перед тим, як клієнт надсилає інформацію до CSP, він створює тег виклику, а потім надсилає його постачальнику хмарних послуг пізніше. Клієнт кидає виклик провайдеру хмарних послуг, перевіряючи цілісність даних за допомогою довіреної третьої сторони (TTP) [21].

2. Послаблення атаки відкату

У запропонованій схемі захист від атаки відкату в хмарному середовищі здійснюється шляхом застосування методу геш-дерева Меркла [25, 26]. У цьому методі тег блоку даних та значення його лічильника оновлюються шоразу, коли оновлюються нові дані. Якщо зловмисник хоче змінити дані, значення лічильника також зміниться.

3. Пом'якшення наслідків візантійського збою та зловмисних атак на дані

Рішенням для даної проблеми є запропонована Browers та ін. [27] криптосистема HAIL (High Availability and Integrity Layer) Protocol. Цей протокол гарантує, що дані користувача зберігаються неушкодженими і можуть бути безпечно отримані з серверів. Для забезпечення гарантії доступності даних використовується коригувальний код Erasure [21].

4. Захист цілісності даних за допомогою шифрування

Даний метод є найкращим для загального захисту даних у хмарному середовищі та використовується у будь-якій системі. Оскільки технології продовжують розвиватися, а старі технології стають вразливіше, з'являтимуться нові методи злому шифрів, а також фатальні недоліки в старих методах шифрування. Хмарні провайдери повинні постійно оновлювати своє шифрування, оскільки дані, які вони зазвичай містять, є особливо цінними [7]. Перед тим як зберігати дані на сервері, слід зашифрувати їх та обчислити геш-значення даних. Це гарантуватиме, що дані не були змінені [28].

5. Техніка доказового володіння даними (PDP)

Техніка PDP використовує протокол відповіді на запит для перевірки цілісності даних, що зберігаються на хмарному сервері. Цей метод використовує симетричне шифрування, наприклад, MAC або будь-яке інше. Файл заповнюється метаданими перед зберіганням або надсиланням його на хмарний сервер. Після надсилання файлу до постачальника хмарних послуг користувач все одно зберігає метадані файлу, щоб перевірити його цілісність. Після цього користувач видаляє локальну копію файлу та перевіряє докази володіння файлом сервером за допомогою протоколу відповіді на виклик [29].

6. Техніка доведення можливості витучення (POR)

Метод Proof of Retrievability (POR) використовується для віддаленої перевірки даних, які зберігаються у постачальника хмарних послуг, за допомогою ключа автентифікації. У цьому методі дані не потрібно отримувати з CSP, і користувач також не зберігає оригінальну копію файлу локально. Користувач зберігає свій файл у CSP разом із ключем автентифікації. Потім користувач може перевірити цілісність даних за допомогою ключа автентифікації, не отримуючи файл з CSP [29, 9].

Висновки

У цій статті було розглянуто поняття безпеки даних у хмарних обчисленнях та коротко оглянуто декілька з тих поширених загроз, що заважають забезпечити цілісність інформації, що там зберігається. Зараз технології хмарних обчислень – це те, чим люди користуються, майже не акцентуючи на це уваги, бо у сьогоденні це явище, що зустрічається майже усюди та у кожній компанії. Не дивлячись на те, що хмарні технології зазвичай впроваджують найсучасніші технології безпеки, на жаль, абсолютно позбутись усіх ризиків неможливо. Окрім огляду поширених загроз ця стаття оглядає можливі методи вирішення проблем, мінімізування можливих вразливостей від атак та забезпечення цілісності у хмарних обчисленнях.

Список літератури: 1. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology,Vol.53,no.6, p. 50, 2009. 2. Reese, G. (2009) Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, California: O'Reilly Media. 3. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., and Brandic, I. (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for

delivering computing as the 5th utility. Future Generation Computer Systems, 25 (6), pp. 599-616. 4. Cloud computing security, Режим доступу – http://en.wikipedia.org/wiki/Cloud_ computing_security. 5. "Top Threats to Cloud Computing v1.0" Cloud Security Alliance. 6. Що таке безпека в хмарі?, Microsoft. Режим доступу – https://www.microsoft.com/uk-ua/security/business/security-101/what-is-cloud-security. 7. Rukavitsyn, Andrey N.; Borisenko, Konstantin A.; Holod, Ivan I.; Shorov, Andrey V. (2017). "The method of ensuring confidentiality and integrity data in cloud computing". *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*. pp. 272–274. 8. Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu – Data Security and Privacy in Cloud Computing. 9. M. S. Giri, B. Gaur, and D. Tomar, A Survey on Data Integrity Techniques in Cloud Computing. 10. Yunchuan Sun, Junsheng Zhang zhangjs, Yongping Xiong, and Guangyu Zhu (2014) "Data Security and Privacy in Cloud Computing". 11. Dissanayaka, Akalanka Mailewa, Susan Mengel, Lisa Gittner, and Hafiz Khan. "Vulnerability prioritization, root cause analysis, and mitigation of secure data analytic framework implemented with mongodb on singularity linux containers." In Proceedings of the 2020 the 4th International Conference on Compute and Data Analysis, pp. 58-66. 2020. 12. A. Jyoti, M. Shrimali, S. Tiwari, and H. P. Singh, "Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 11, pp. 4785–4814, Nov. 2020, doi: 10.1007/s12652-020-01747-z. 13. Te-Shun Chou "Security threats on cloud computing vulnerabilities", International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 3, June 2013, pp. 84-85. 14. Ramandeep Kaur, Pushpendra Kumar Pateriya, "A Study on Security Requirements in Different Cloud Frameworks", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013, pp.134-135. 15. Y. Chen, L. Li, and Z. Chen, "An Approach to Verifying Data Integrity for Cloud Storage," in 2017 13th International Conference on Computational Intelligence and Security (CIS), Dec. 2017, pp. 582–585, doi: 10.1109/CIS.2017.00135. 16. H. Mohapatra, "Handling of Man-In-The-Middle Attack in WSN Through Intrusion Detection System," Int. J. Emerg. Trends Eng. Res., vol. 8, no. 5, pp. 1503–1510, May 2020, doi: 10.30534/ijeter/2020/05852020. 17. What is a DDoS attack? Режим доступу – https://www.microsoft.com/en-us/security/business/security-101/what-is-a-ddos-attack. 18. Lai, Cheng-I., Alberto Abad, Korin Richmond, Junichi Yamagishi, NajimDehak, and Simon King. "Attentive filtering networks for audio replay attack detection." In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6316-6320. IEEE, 2019. 19. Shetty, Roshan Ramprasad, Akalanka Mailewa Dissanayaka, Susan Mengel, Lisa Gittner, Ravi Vadapalli, and Hafiz Khan. "Secure NoSQL based medical data processing and retrieval: the exposome project." In Companion Proceedings of the10th International Conference on Utility and Cloud Computing, pp. 99-105. 2017. 20. Mailewa Dissanayaka, Akalanka, Roshan Ramprasad Shetty, Samip Kothari, Susan Mengel, Lisa Gittner, and Ravi Vadapalli. "A review of MongoDB and singularity container security in regards to hipaa regulations." In Companion Proceedings of the10th International Conference on Utility and Cloud Computing, pp. 91-97. 2017. 21. "Survey on various data integrity attacks in cloud environment and the solutions - IEEE Conference Publication." Режим доступу – https://ieeexplore.ieee.org/abstract/document/6528889. 22. Thapa, Suman, and Akalanka Mailewa. "The Role of Intrusion Detection/Prevention Systems in Modern Computer Networks: A Review." In Conference: Midwest Instruction and Computing Symposium (MICS), vol. 53, pp. 1-14. 2020. 23. S. Sadalai and S. S., A Survey on Cloud Security Issues and Challenges with Possible MeasuresA Survey on Cloud Security Issues and Challenges with Possible Measures. 2016. 24. Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec. 2012, doi: 10.1109/TPDS.2012.66. 25. J. Feng, Y. Chen, D. H. Summerville, and K. Hwang, "Fair Non-repudiation Framework for Cloud Storage: Part II," in Cloud Computing for Enterprise Architectures, Z. Mahmood and R. Hill, Eds. London: Springer, 2011, pp. 283–300. 26. J. Feng, Y. Chen, D. Summerville, W. Ku, and Z. Su, "Enhancing cloud storage security against roll-back attacks with a new fair multi-party nonrepudiation protocol," in 2011 IEEE Consumer Communications and Networking Conference (CCNC), Jan. 2011, pp. 521–522, doi: 10.1109/CCNC.2011.5766528. 27. H. Lin and W. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 6, pp. 995–1003, Jun. 2012, doi: 10.1109/TPDS.2011.252. 28. R. V. Rao and K. Selvamani, "Data Security Challenges and Its Solutions in Cloud Computing," Procedia Comput. Sci., vol. 48, pp. 204–209, Jan.2015, doi: 10.1016/j.procs.2015.04.171. 29. K. N. Sevis and E. Seker, "Survey on Data Integrity in Cloud," in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Jun. 2016, pp. 167–171, doi: 10.1109/CSCloud.2016.35.

ДОДАТОК Б

ЗАГРОЗИ ЦІЛІСНОСТІ ДАНИХ ТА ПРОПОНОВАНІ РІШЕННЯ У ХМАРНИХ ОБЧИСЛЕННЯХ

Анастасія КРАВЧЕНКО[a], Софія КРАВЧЕНКО[a], Всеволод БОБУХ[b]

[a] *Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків, Україна*
[b] *АТ «Інститут інформаційних технологій», вул. Колегіальна 15, Харків, Україна*

Анотація. *Зараз технології хмарних обчислень – це те, чим люди користуються майже усюди. Хмарні технології зазвичай впроваджують найсучасніші технології безпеки, але абсолютно позбутись усіх ризиків неможливо, а враховуючи той факт, що люди довіряють важливу інформацію, треба максимально забезпечити її цілісність.*

Ключові слова. *Хмарні обчислення, цілісність, загрози цілісності.*

1. Вступ

Хмарні обчислення розглядаються як одна з найуспішніших обчислювальних технологій, здатних вирішити цілу низку проблем, що стоять перед людством. Згідно з визначенням NIST США, хмарні обчислення – це модель забезпечення повсюдного та зручного доступу на вимогу, через мережу до спільного пулу обчислювальних ресурсів, що піддаються налаштуванню, і, які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера [1]. Хмарні обчислення мають декілька ключових особливостей, як надійність, широкий мережевий доступ, масштабованість інфраструктури, гнучкість, незалежність від місця розташування, економія на масштабах і економічна ефективність та стійкість [2].

2. Забезпечення цілісності в хмарних обчисленнях

Безпека хмарних обчислень визначається як «Піддомен комп'ютерної безпеки, мережевої безпеки та, ширше, інформаційної безпеки. Це стосується широкого набору політик, технологій і елементів керування, які застосовуються для захисту даних, додатків і відповідної інфраструктури хмарних обчислень».

Коли організація вирішує зберігати дані або розміщувати додатки в публічній хмарі, вона втрачає можливість мати фізичний доступ до серверів, на яких зберігається її інформація [4] .Як наслідок, потенційно конфіденційні дані підлаються ризику інсайдерських атак. Крім того, центри обробки даних рекомендується часто перевіряти на предмет підозрілої активності.

2.1. Поширені атаки на цілісність

Цілісність даних означає захист даних від несанкціонованого видалення, модифікації чи фальсифікації [5].

1. Несанкціонований доступ. Атака направлена на безконтрольні зміни даних, на які не зможе впливати авторизований користувач.

2. Блокування даних. Загроза утворюється при переході від одного постачальника послуг до іншого і так як різні постачальники надають різні послуги, тому при переході може трапитись втрата даних користувачів або їх блокування.

3. SQL-ін'єкції. Атака націлена на SQL-сервери, які запускають уразливі програми баз даних. Хакери використовують вразливі місця веб-серверів і вводять шкідливий код, щоб обійти вхід і отримати несанкціонований доступ до серверних баз даних.

4. Атака "людина посередині" (MiMA). MiMA зазвичай виникає, коли різні користувачі хмари спілкуються один з одним або спільно використовують ресурси з хмарного середовища [5]. Недостатнє шифрування може зробити користувачів вразливими до атаки MiMA, яка є непрямою атакою.

5. DdoS-атака. DDoS-атаки націлені на веб-сайти та сервери, порушуючи роботу мережевих сервісів з метою виснаження ресурсів програми.

6. Атака підробки тегів. Атака відбувається, якщо продавець обманює своїх клієнтів, показуючи неправильний штрих-код, за яким переходить користувач та втрачає контроль над безпекою особистих даних.

7. Візантійська атака. Ця атака відбувається на різні частини хмарних обчислень шляхом зупинки або виходу з ладу систем.

2.2. Методи забезпечення цілісності даних

Проаналізувавши наведені вище загрози, перейдемо до методів забезпечення цілісності та запобігання атак на цілісність у хмарних сховищах. Нижче наведено декілька механізмів, що використовують для захисту володіння даними та їх цілісності в середовищі хмарних обчислень.

1. Пом'якшення наслідків підробки тегів та атак витоку даних. Щоб запобігти цій атаці, існує схема, запропонована Yun Zhu та ін. [4]. Сутність методу: перед тим, як клієнт надсилає інформацію до CSP, він створює тег виклику, надсилає його постачальнику хмарних послуг. Клієнт кидає виклик провайдеру, перевіряючи цілісність даних за допомогою TTP.

2. Послаблення атаки відкату. У цьому методі тег блоку даних та значення його лічильника оновлюються шоразу, коли оновлюються нові дані.

3. Пом'якшення наслідків візантійського збою та зловмисних атак на дані. Рішенням для даної проблеми є криптосистема HAIL Protocol, який гарантує, що дані користувача зберігаються неушкодженими і можуть бути безпечно отримані з серверів.

4. Захист цілісності даних за допомогою шифрування. Даний метод є найбільш поширеним. Хмарні провайдери повинні постійно оновлювють своє шифрування, оскільки дані, які вони зазвичай містять, є особливо цінними. Перед тим як зберігати дані на сервері, слід зашифрувати їх та обчислити геш-значення даних.

5. Техніка доказового володіння даними (PDP). Техніка PDP використовує протокол відповіді на запит для перевірки цілісності даних, що зберігаються на хмарному сервері. Файл заповнюється метаданими перед зберіганням або надсиланням його на хмарний сервер. Після надсилання файлу до постачальника хмарних послуг користувач все одно зберігає метадані файлу, щоб перевірити його цілісність. Після цього користувач видаляє локальну копію файлу та перевіряє докази володіння файлом сервером за допомогою протоколу відповіді на виклик [4].

6. Техніка доведення можливості вилучення (POR). Метод POR використовується для віддаленої перевірки даних, які зберігаються у постачальника хмарних послуг, за допомогою ключа автентифікації. Користувач зберігає свій файл у CSP разом із ключем автентифікації. Потім користувач може перевірити цілісність даних за допомогою ключа автентифікації, не отримуючи файл з CSP [3].

3. Висновки

У цій роботі було коротко оглянуто декілька з тих поширених загроз, що заважають забезпечити цілісність інформації, що там зберігається та методи їх запобігання. На жаль, повністю забезпечити це не завжди можливо, але можна мінімізувати ризики, і саме на такі методи було звернено увагу.

4. Список літератури

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology,Vol.53,no.6, p. 50, 2009.

[2] Reese, G. (2009) Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, California: O'Reilly Media.

[3] K. N. Sevis and E. Seker, "Survey on Data Integrity in Cloud," in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Jun. 2016, pp. 167–171

[4] Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu – Data Security and Privacy in Cloud Computing.

[5] Ramandeep Kaur, Pushpendra Kumar Pateriya, "A Study on Security Requirements in Different Cloud Frameworks", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013, pp.134-135.