11. 32. Kargas, A. Interpretation of engineering drawings as solid models / A. Kargas, P. Cooley, T.H.E. Richards // Computer-Aided Engineering Journal. – april 1988. – P. 67 – 78.
12. Етапи розробки 3D персонажа: роз'яснення [Електронний ресурс] – Режим доступу: *https://nachasi.com/tech/2019/01/31/yak-pratsyuye-machine-learning/*
13. Лазерне 3D сканування [Електронний ресурс] – Режим доступу: https://galychgeobud.com.ua/uk/lazerne-3d-skanuvannya
14. Лазерне сканування як інструмент формування 3D кадастру та управління земельними ресурсами [Електронний ресурс] – Режим доступу: https://nubip.edu.ua/ node/75470
15. Лазерне 3D сканування об'єктів в Україні. [Електронний ресурс] – Режим доступу: *https://3dway.com.ua/blog/3d-scanning*

Робота виконана під науковим керівництвом канд. техн. наук, доцента
РЗАЄВОЇ С.Л.

ДОСЛІДЖЕННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ У КЛІЄНТ-СЕРВЕРНИХ ЗАСТОСУНКАХ

**МОЛЧАНОВА А.М., 1 курс 2мз група ФІТ ДТЕУ,
освітня програма «Інженерія програмного забезпечення»**

У статті розглянуто основні види потенційних загроз у клієнт-серверних системах, відповідно до них виділено та проаналізовано релевантні методи захисту даних у клієнт-серверних застосунках. Надано рекомендації щодо підходу до проєктування надійно захищених клієнт-серверних застосунків.

In the article, the main types of potential threats in client-server systems are considered, according to them relevant methods of data protection in client-server applications are selected and analyzed. Recommendations on the approach to the design of securely protected client-server applications are provided.

Актуальність. На сьогодні Інтернет є невід'ємним аспектом життя кожного з нас і, як наслідок, більшість програмних продуктів, котрими ми користуємося щодня, засновані на клієнт-серверній архітектурі. До таких, наприклад, відносяться сервіси електронної пошти, соціальні мережі, фінансові додатки, месенджери, онлайн-ігри тощо.

В свою чергу, використання клієнт-серверних додатків неможливе без передачі та отримання даних, що можуть стати об'єктом кіберзагроз, особливо в умовах воєнного стану. Саме тому критичним є захист даних на усіх етапах передачі мережею для забезпечення надійності клієнт-серверних систем у сучасному динамічному середовищі.

Метою статті є пошук слабких місць та на основі цього дослідження ефективності методів захисту даних у клієнт-серверних застосунках.

Об'єктом дослідження є процес обміну даними між клієнтом та сервером при використанні клієнт-серверних застосунків.

Предметом дослідження є методи шифрування, аутентифікації, авторизації, контролю доступу та моніторингу даних у контексті їх застосування у клієнт-серверних застосунках.

Аналіз попередніх досліджень. Вітчизняні та іноземні наукові джерела налічують широке коло напрацювань в галузі інформаційної безпеки загалом, і деякі з них висвітлюють окремі методи захисту інформації у кіберпросторі, таких вчених як О. Білокуров [1],

В. Богом'я [2], Т. Каткова [3], С. Лаптєв [4], О.С. Малець [5] та ін. Однак, питання забезпечення безпеки даних у клієнт-серверних застосунках потребує висвітлення та дослідження подальших перспектив розвитку методів захисту у таких системах.

Виклад основного матеріалу. Побудова будь-якої системи захисту має починатися з аналізу потенційних загроз, на які повинна реагувати дана система та ефективно їх долати або попереджувати їх негативний вплив у разі виникнення.

Коли мова йде про загрози інформації, що передається від клієнта до сервера і навпаки, загальноприйнятою є їх класифікація згідно стандарту ISO/IEC 15408 на три групи:

- загрози конфіденційності;
- загрози цілісності;
- загрози доступності.

Розглянемо нижче сутність та підвиди кожної з груп загроз.

Загроза порушення *конфіденційності* полягає в тому, що інформація стає відомою тому, хто не володіє повноваженнями доступу до неї.

Найпоширенішими з таких загроз є:

1) Перехоплення даних (Data Interception) – це процес перехоплення чутливої інформації під час її передачі через мережу. Наприклад, зловмисники можуть використовувати програмне забезпечення для перехоплення паролів, банківських реквізитів або інших конфіденційних даних.

2) Атака «Man-in-the-Middle» (MITM) – це атака, при якій зловмисник вставляється між комунікуючими сторонами в мережі і перехоплює та змінює передані дані без їхнього відома. Це дозволяє зловмиснику отримати доступ до конфіденційної інформації або навіть модифікувати дані.

3) Втрата або крадіжка пристроїв (Device Loss or Theft) – якщо пристрої, що містять конфіденційні дані, втрачаються або потрапляють у руки зловмисників, це може призвести до небажаного доступу до цих даних.

4) Атака на бездротові мережі (Wireless Network Attack) – це атака, яка використовується для зламування захищених бездротових мереж, таких як Wi-Fi, з метою отримання доступу до конфіденційних даних, що передаються по мережі.

5) Фішинг (Phishing) – це соціально-інженерна атака, під час якої зловмисники намагаються обманом отримати конфіденційні дані, такі як паролі чи номери кредитних карток, від користувачів шляхом відправлення підроблених повідомлень електронною поштою чи повідомлень у соціальних мережах.

Загрози порушення *цілісності* – це загрози, пов'язані з імовірністю модифікації тієї чи іншої інформації, що зберігається в інформаційній системі.

До таких загроз найчастіше відносять:

1) Маніпуляція даними (Data Manipulation) – це атака, внаслідок якої зловмисники намагаються змінити чи порушити цілісність даних, що передаються через мережу. Наприклад, зміна вмісту повідомлень або файлів під час їх передачі.

2) Введення шкідливих кодів (Malware Injection)- це атака, під час якої зловмисники впроваджують шкідливий код у систему або мережу з метою порушення їхньої цілісності. Це може включати введення вірусів, червів, троянців або іншого шкідливого програмного забезпечення.

3) Атаки на бази даних (Database Attacks) – це атаки, які спрямовані на порушення цілісності баз даних, наприклад, внесення змін у вміст бази даних, видалення чи втрата даних, чи отримання несанкціонованого доступу до конфіденційної інформації.

4) Маніпуляція параметрів запитів (Parameter Tampering) – це атака, при якій зловмисники намагаються змінити параметри запитів, що передаються через мережу, з метою зміни результатів або виклику некоректної поведінки системи.

5) Внесення змін у сесійні дані (Session Hijacking) – це атака, під час якої зловмисники намагаються взяти під контроль або викрасти сесійні дані аутентифікації користувача, щоб отримати доступ до захищених ресурсів або виконати недозволені дії в системі.

Загрози *доступності* представляють собою здійснення дій, які унеможливлюють чи ускладнюють доступ до ресурсів інформаційної системи.

Такими загрозами є:

1) Атаки з перевантаженням серверів (Server Overload) – це атаки, під час яких зловмисники намагаються перевантажити сервери, використовуючи велику кількість запитів чи ресурсів, що може призвести до відмови в обслуговуванні або зниження продуктивності.

2) Атаки на аутентифікацію (Authentication Attacks) – це атаки, які спрямовані на порушення доступності системи шляхом атак на механізми аутентифікації. Наприклад, спроби перебору паролів, використання слабких або скомпрометованих облікових записів.

3) Атаки на системи керування доступом (Access Control Attacks) – це атаки, які спрямовані на обхід або порушення механізмів контролю доступу до системи чи ресурсів. Наприклад, намагання отримати несанкціонований доступ до захищених даних чи ресурсів.

4) Видалення або знищення даних (Data Deletion or Destruction) – це атаки, які спрямовані на фізичне видалення або знищення даних чи ресурсів, що може призвести до втрати доступу до інформації або неможливості використання ресурсів.

5) Відмова в обслуговуванні через програмне забезпечення (Software Failure) – це ситуації, коли відмови в обслуговуванні виникають через програмні помилки, баги або недоліки, що призводять до недоступності системи чи даних.

Аналіз механізму дії загроз даним, що передаються та приймаються у процесі використання клієнт-серверних застосунків, дозволяє виділити нову ознаку класифікації загроз – за вразливим об’єктом: мережа; клієнт; сервер; код застосунка.

Таким чином, наведені вище найпоширеніші загрози безпеці даних, можна класифікувати, як представлено у табл.1.

Таблиця 1

Класифікація загроз даним клієнт-серверного застосунку за вразливим до атаки об’єктом

Тип загрози	Вразливий об’єкт			
	Мережа	Клієнт	Сервер	Код застосунка
Data Interception	+			
«Man-in-the-Middle» (MITM)	+			
Device Loss or Theft		+	+	
Wireless Network Attack	+			
Phishing		+		
Data Manipulation	+			+
Malware Injection				+
Database Attacks			+	+
Parameter Tampering		+		+
Session Hijacking				+
Server Overload			+	
Authentication Attacks				+
Access Control Attacks				+
Data Deletion or Destruction		+	+	
Software Failure				+

Як бачимо, найбільш схильним до вразливостей об’єктом є саме код клієнт-серверного застосунка.

Для порівняння, відповідно рейтингу OWASP Top 10 за 2023 рік[6] найбільшими вразливостями систем, що здійснюють передачу даних мережею Інтернет, є такі:

1. Порушена авторизація на рівні об’єкта
2. Порушена автентифікація
3. Порушена авторизація на рівні властивостей об’єкта

4. Необмежене використання ресурсів
5. Порушена авторизація функціонального рівня
6. Необмежений доступ до конфіденційних бізнес-потоків
7. Підробка запитів на стороні сервера
8. Неправильна конфігурація безпеки
9. Неналежне управління інвентарем
10. Небезпечне використання API

Наведений рейтинг підтверджує висновок про необхідність приділяти більше уваги кодовій базі клієнт-серверних застосунків, акцентуючись на забезпеченні їх надійного захисту від різних типів загроз.

Сучасна система захисту даних будується з підсистем, кожна з яких виконує свою важливу роль, і забезпечення їх якості роботи на високому рівні гарантуватиме безпеку роботи з даними у системі в цілому: шифрування; аутентифікації та авторизації; контролю доступу; моніторингу та аудиту безпеки.

Підсистема *шифрування* дозволяє захистити дані під час їх створення, зберігання і транспортування мережею. Так, наприклад, існують різні підходи до шифрування:

1) Шифрування для комунікації або наскрізне шифрування (End-to-End Encryption, E2EE). Даний метод передбачає шифрування даних на рівні додатків або клієнтської сторони перед їх відправленням через мережу. Зашифровані дані залишаються зашифрованими протягом усього шляху передачі, аж до моменту отримання їх призначеним отримувачем. Розшифрування відбувається лише на пристрої отримувача, тому що ключі для розшифрування зберігаються тільки на цьому пристрої.

Найпоширеніші алгоритми шифрування, які використовуються в E2EE, включають AES, RSA та Signal Protocol. Ці алгоритми вважаються безпечними та широко застосовуються через їх стійкість до атак.

Керування ключем шифрування також має вирішальне значення для забезпечення безпеки E2EE. Ефективне керування ключами передбачає створення, розповсюдження, зберігання та періодичне оновлення ключів шифрування. Погане керування ключами може призвести до вразливостей у процесі шифрування, що може поставити під загрозу загальну безпеку каналу зв'язку [7].

2) Шифрування каналу передачі даних (Channel Encryption). У цьому випадку шифрування відбувається на рівні транспортного протоколу, тобто передача даних зашифровується на мережевому рівні. Дані розшифровуються на призначеному сервері чи вузлі мережі, а не на кінцевому пристрої отримувача.

Найкраще зарекомендували себе на сьогодні такі типи захищених протоколів як TLS, SSL, IPsec, SSH, тож вони є найбільш рекомендованими до застосування.

На відміну від захищеного каналу, незахищений канал не зашифрований і може піддаватися прослуховуванню та підробці пакетів. Захищений зв'язок можливий через незахищений канал, якщо переданий вміст зашифровано перед передачею.

3) Шифрування пам'яті (Memory Encryption). Цей метод передбачає шифрування даних, які зберігаються в оперативній пам'яті пристрою або в оперативній пам'яті операційної системи. Дані автоматично шифруються, коли вони зберігаються в оперативній пам'яті, і розшифровуються тільки при необхідності доступу до них.

Це можна зробити за допомогою різних методів, таких як повне шифрування пам'яті (Full Memory Encryption), спотворення пам'яті (Memory Scrambling) та ізоляція пам'яті (Memory Isolation).

Загалом, незважаючи на те, що шифрування даних RAM може забезпечити підвищену безпеку, компроміси в продуктивності, вартості та складності сприяли його обмеженому застосуванню на практиці [8].

4) Шифрування баз даних (Database Encryption). Цей метод передбачає шифрування цілісної бази даних або окремих таблиць та колонок в базі даних. Шифрування баз даних

може бути використано як додатковий шар захисту, щоб зберегти конфіденційність даних навіть у випадку компрометації самого сервера або бази даних.

Поширеними методами шифрування баз даних є метод API (шифрування на рівні програми, яке підходить для будь-якого продукту бази даних), метод плагіна (підключаємий модуль шифрування або «пакет» до СУБД) та метод TDE (шифрування та дешифрування в самій системі бази даних), що є найпопулярнішим [9].

Підсистема *аутентифікації та авторизації* є ключовим аспектом безпеки у клієнт-серверних додатках, оскільки дозволяє перевіряти і контролювати доступ користувачів до ресурсів.

Підсистема *контролю доступу* в клієнт-серверних додатках визначає, як користувачі можуть взаємодіяти з ресурсами на основі їхньої ідентифікації та авторизації.

Станом на сьогодні виділяють 5 основних типів моделей контролю доступу [10]:

- Керування політикою на основі атрибутів (Attribution-Based Policy Control) – цей метод додає динамічний елемент, доступ до якого визначається набором контекстних атрибутів. Наприклад, запит на файл, надісланий поза звичайним робочим часом із невідомого місця, може викликати підтвердження або відмову в доступі до політики.

- Контроль доступу на власний розсуд (Discretionary Access Control, DAC) – власник даних контролює дані та відповідні системи, необхідні для доступу. Вони можуть делегувати дозволи суб'єктів локально, що робить DAC ідеальним гнучким підходом для окремих команд, які можуть визначати власні правила доступу. Однак гнучкість також може призвести до непослідовності, що робить цей децентралізований метод менш безпечним.

- Контроль доступу на основі ролей (Role-Based Access Control, RBAC) – дозволи попередньо призначені для організаційних ролей, які суб'єкти вже мають у системі. Ось чому ця форма контролю доступу працює найкраще, коли вона чітко дотримується організаційних структур та ієрархій. Варто зазначити, що цей метод є відносно жорстким і його важко масштабувати.

- Обов'язковий контроль доступу (Mandatory Access Control, MAC) – обов'язковий контроль доступу в кібербезпеці є найсуворішою формою контролю доступу до даних. Його зазвичай використовують уряди та військові. Адміністратори встановлюють мітки та дозволи безпеки для суб'єктів і об'єктів.

- Контроль доступу на основі політики (Policy-Based Access Control, PBAC) – PBAC додає ще більше динамізму та масштабованості для примусового доступу та контролю політики. Доступ настільки тонкий, наскільки ваша політика потребує, аж до рівня стовпців, рядків і клітинок. Фільтрація, маскування та анонімізація відбуваються в режимі реального часу, що забезпечує безпечне та сумісне самообслуговування.

Останньою для успішного впровадження надійної системи захисту даних клієнт-серверного застосунка є підсистема *моніторингу та аудиту безпеки*.

Спеціалізовані системи моніторингу призначені автоматизувати процес збору та аналізу інформації, яка надходить від різних засобів захисту. В західній термінології такі системи моніторингу позначаються абревіатурою SIEM (Security Information and Event Management). Технологія функціонування сучасних систем SIEM передбачає розподіл процесу обробки подій безпеки на шість основних етапів: фільтрація, агрегація, нормалізація, збір, кореляція та візуалізація.

Аудит безпеки дозволяє оцінити ступінь захищеності застосунка на основі даних моніторингу, а також виявити потенційні загрози та вразливі місця.

Отже, розробка системи безпеки даних клієнт-серверних застосунків може бути реалізована з використанням усіх 4 підсистем або за допомогою окремих з них, але при цьому якість захисту даних буде відповідно нижчою.

Розробка захищених клієнт-серверних застосунків вимагає врахування специфіки додатку, що впливає на вибір методів захисту. Залежно від виду застосунку, наприклад, фінансові послуги, соціальні мережі, електронна комерція, та типу даних, що обробляються, рекомендується використовувати наступні методи захисту:

- Наскрізне шифрування (E2EE) ідеально підходить для застосунків, які передають чутливу інформацію між клієнтом та сервером, забезпечуючи, шифрування даних від відправника до одержувача без можливості дешифрації на сервері.

- Мультифакторна аутентифікація (MFA) критично важлива для застосунків, що вимагають високого рівня захисту ідентифікаційних даних, зокрема в банківських та фінансових сервісах.

- Контроль доступу на основі політики (RBAC) забезпечує ефективне управління дозволами користувачів відповідно до їх ролей в організації, знижуючи ризик несанкціонованого доступу.

- Використання безпечних програмувальних інтерфейсів (API) особливо важливо для застосунків, які інтегрують сторонні сервіси, що потребує забезпечення безпеки на кожному етапі взаємодії з API.

Ці рекомендації допоможуть фокусуватись на важливих аспектах безпеки, що відповідають специфіці використання застосунків.

Висновки. Розвиток технологій розробки програмних продуктів супроводжується також розвитком видів інформаційних загроз, внаслідок чого виникає необхідність в проєктуванні та впровадженні ефективної системи захисту продукту з метою мінімізації потенційних вразливостей.

Клієнт-серверні застосунки щороку піддаються загрозам порушення конфіденційності, цілісності та доступності інформації, що можливі через різноманітні вразливості як мережі, якою передаються дані застосунка, так і пристрою клієнта, серверного обладнання, й самого коду застосунка. При цьому, аналіз причин атак на дані доводить, що вразливості систем захисту самих клієнт-серверних додатків посідають перше місце.

Так, реалізація сучасної системи забезпечення безпеки даних клієнт-серверного додатка має здійснюватися з використанням ефективних методів та алгоритмів шифрування, аутентифікації та авторизації, контролю доступу, моніторингу та аудиту безпеки, а вибір конкретних методів і технологій повинен враховувати вимоги до безпеки, зручності використання та специфіку додатка.

Список використаних джерел

1. Аналіз методів захисту інформації, що знаходять використання у сучасних месенджерах / О. О. Білокуров, М. А. Майба, О. К. Шлома, М. О. Дробяз // Проблеми електромагнітної сумісності перспективних безпроводових мереж зв'язку (ЕМС-2022) : матеріали восьмої Міжнародної науково-технічної конференції, 24–25 листопада 2022 р. – Харків, ХНУРЕ, 2022. – С. 71-73.

2. Богом'я В.І., Кочегаров В.С. Кібербезпека в хмарних сервісах за допомогою застосування криптографічних методів. Водний Транспорт: Збірник наукових праць. № 1(37), 2023. С. 239-246.

3. Каткова Т. І. Забезпечення криптографічного захисту державних інформаційних ресурсів. Наукові нотатки. Луцьк. 2022. № 73. С. 54–58.

4. Лаптєв С. О. Удосконалений метод захисту персональних даних від атак за допомогою алгорітмів соціальної інженерії. Кібербезпека: освіта, наука, техніка. № 4 (16), 2022. С. 45–622.

5. Остап-Святослав Малець. Розвиток й застосування криптографічних та стенографічних засобів захисту інформації в сучасному світі / О.-С. Малець, О.Смотр // збірник тез доповідей VI Всеукраїнської науково-практичної конференції молодих учених, студентів і курсантів «Захист інформації в інформаційно-комунікаційних системах», м. Львів, 30 листопада 2023 року. Львів, ЛДУ БЖД, 2023.

6. OWASP Top 10 API Security Risks – 2023. *OWASP API Security Top 10* : веб-сайт. URL: https://owasp.org/API-Security/editions/2023/en/0x11-t10/ (дата звернення: 18.03.2024).

7. Exploring E2EE: Real-world Examples of End-to-End Encryption. *Kiteworks* : веб-сайт. URL: https://www.kiteworks.com/secure-file-sharing/real-world-examples-of-end-to-end-encryption/ (дата звернення: 26.03.2024).

8. What are RAM data encryption methods, why is it not popular? *Quora* : веб-сайт. URL: https://www.quora.com/What-are-RAM-data-encryption-methods-why-is-it-not-popular (дата звернення: 30.03.2024).

9. Types of Database Encryption Methods. *N-able* : веб-сайт. URL: https://www.n-able.com/blog/types-database-encryption-methods (дата звернення: 01.04.2024).

10. Access control methods: What they are, how they work, and how to choose the right approach. *Velotix* : веб-сайт. URL: https://www.velotix.ai/resources/blog/what-are-the-best-access-control-methods/ (дата звернення: 03.04.2024).

Робота виконана під науковим керівництвом канд. екон. наук, доцента
ПАЛАГУТИ К.О.

ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ У АНДРОЇД-ДОДАТКАХ

МОЛЯВІН А.І., 1 курс 3м група ФІТ ДТЕУ,
освітня програма «Інженерія програмного забезпечення»

У статті розглянуто ключові аспекти та методики оптимізації продуктивності Android-додатків, що є важливим компонентом для розробників мобільного програмного забезпечення. Акцент робиться на ефективному використанні ресурсів пристрою, мінімізації споживання енергії та підвищення швидкодії додатків.

The article discusses the key aspects and methods of optimizing the performance of Android applications, which is an important component for mobile software developers. The emphasis is on the efficient use of device resources, minimizing energy consumption, and improving application performance.

Актуальність. У світі зростають очікування користувачів щодо мобільних додатків, продуктивність стає ключовим фактором, що впливає на успішність додатка на ринку. Високий рівень продуктивності та мінімальне споживання ресурсів є вирішальним для забезпечення задоволення користувачів та утримання їх уваги. Зі збільшенням кількості додатків в магазинах додатків, користувачі стають все більш вимогливими до швидкодії та енергоефективності. Користувачі не лише прагнуть швидкості завантаження та реакції на свої дії, але й очікують, що додатки не будуть чинити надмірного навантаження на батарею чи оперативну пам’ять пристрою.

Водночас, зростання кількості функцій у додатках та поява нових технологій, таких як штучний інтелект та машинне навчання, висувають нові виклики перед розробниками. Впровадження технологій вимагає глибокої оптимізації продуктивності для підтримання гладкого та приємного користувацького досвіду. Таким чином, оптимізація продуктивності додатків перетворюється на пріоритетний напрямок у розробці мобільного програмного забезпечення, що вимагає постійного вдосконалення знань та навичок розробника.

Метою статті є дослідження ефективних методів та практик оптимізації продуктивності додатків, з метою підвищення їх ефективності та конкурентоспроможності.

Об’єктом дослідження є процес оптимізації продуктивності мобільних додатків, реалізованих на платформі Android.