Назаров Є.М.
здобувач вищої освіти
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Науковий керівник: к.т.н., ст. викл. Мірошниченко М.Ю.

ЗАХИСТ НА РІВНІ ДОДАТКІВ: СУЧАСНІ ВИКЛИКИ ТА ТЕХНОЛОГІЇ

Анотація. Захист на рівні додатків є важливим аспектом в сучасній інформаційній безпеці. Ця стаття присвячена дослідженню сучасних викликів та передових технологій у цій області. Розглядаються технологій та методи захисту додатків, включаючи шифрування, аутентифікацію та контроль доступу. Описано особливості DevSecOps-підходу для забезпечення вбудованої безпеки в розробку додатків. Використання мультифакторної автентифікації, шифрування, захисту від атак та витоків даних дозволяє забезпечити безпеку користувачів та конфіденційність їх даних.

Ключові слова: програмне забезпечення, захист додатків, кібербезпека, DevSecOps, методи шифрування.

Nazarov Ye.M. Application-level security: modern challenges and technologies. *Application-level protection is an important aspect of modern information security. This article explores current challenges and advanced technologies in this area. Application security technologies and methods are reviewed, including encryption, authentication, and access control. Features of the DevSecOps approach to ensure built-in security in application development are described. The use of multi-factor authentication, encryption, protection against attacks and data leaks allows you to ensure the safety of users and the confidentiality of their data.*

Key words: *software, application security, cyber security, DevSecOps, encryption techniques.*

Актуальність дослідження. Сучасний цифровий простір надає нам безліч можливостей для роботи та відпочинку завдяки мережі Internet, мобільним додаткам, імерсивним технологіям [5, с. 178] тощо.

Разом із цим, з'являються загрози для безпеки даних та інформації, в які потрапляють звичайні користувачі, підприємства та інші організації. Спроби несанкціонованого доступу, кібератаки та витоки даних стають все більшими викликами для сучасних розробників і користувачів додатків. Тому програмне забезпечення, зокрема інформаційні системи, повинні мати різні механізми захисту до внутрішніх та зовнішніх загроз, забезпечувати безпечну обробку та передачу інформації [7, с. 120]. Розвиток технологій дозволяє створювати більш потужні та ефективні засоби захисту інформації та програмних продуктів.

Метою статті є аналіз сучасних підходів до захисту інформації та працездатності програмного забезпечення від зловмисного втручання.

Виклад основного матеріалу. До важливих підходів до захисту інформації можна віднести технологію MFA (мультифакторну автентифікацію). MFA є важливим елементом кібербезпеки, що широко використовується в банківській сфері, електронній комерції, корпоративних мережах та інших сферах, де безпека даних має важливе значення. Цей метод дозволяє значно підвищити безпеку, оскільки навіть у випадку втрати логіну та пароля, зловмисники не матимуть доступу до додатку без додаткових підтверджень. Дана технологія полягає в тому, що користувачі додатків повинні підтверджувати свою ідентичність не лише через стандартний спосіб (введення логіну та паролю), але й шляхом застосування додаткових автентифікаційних факторів [8, с. 13].

До основних компонентів МФА можна віднести наступні фактори:

1. «Що вам відомо?» – фактор, що включає в себе знання конфіденційної інформації, такої як пароль, ПІН-код, відповіді на секретні питання;
2. «Що вам належить?» – фактор, що охоплює фізичні предмети або пристрої, які користувач повинен мати при собі, такі як смарт-карти, токени, мобільні пристрої або USB-ключі;
3. «Хто ви?» – фактор, що базується на біометричних даних, таких як відбитки пальців, сканування обличчя, розпізнавання голосу або інші фізичні характеристики користувача.

Процес MFA може включати в себе різні комбінації цих факторів. Наприклад, 2FA (двофакторна аутентифікація) вимагає застосування двох факторів, зазвичай комбінації пароля і одного з інших факторів, таких як одноразовий код, що відправляється на мобільний пристрій користувача. При застосуванні методу трьохфакторної аутентифікації

(3FA) від користувача вимагається підтвердити особу за допомогою трьох різних факторів.

Іншою важливою складовою захисту конфіденційних даних на рівні додатків є шифрування інформації за допомогою складних алгоритмів. Як відомо шифрування – це процес перетворення звичайного тексту, вхідних даних, у криптований (тобто шифрований) вигляд за допомогою використання спеціального ключа [1, с. 23]. Для застосування шифрування на рівні додатків необхідно враховувати ряд важливих принципів:

1. Симетричне та асиметричне шифрування. Для симетричного шифрування один і той же ключ використовується для шифрування та розшифрування даних. У випадку асиметричного шифрування використовуються два ключі, один для шифрування, інший – для розшифрування [4, с. 23].
2. Безпека ключів передбачає, що ключі мають бути належним чином збережені і захищені від несанкціонованого доступу.
3. Протоколи обміну ключами використовуються для безпечного обміну ключами між додатками.

Сучасні методи шифрування можна поділити на:

1. Протоколи TLS (Transport Layer Security) і SSL (Secure Sockets Layer) використовуються для забезпечення безпечного з’єднання між клієнтом і сервером. Вони використовують симетричне та асиметричне шифрування для захисту даних під час передачі [1, с. 25].

2. Шифрування даних в базах даних надає можливість шифрувати дані безпосередньо в самій базі. Це допомагає захистити дані від несанкціонованого доступу навіть після їх збереження.

3. Шифрування повного диску полягає в тому, що всі дані на диску або в пристрої шифруються автоматично. Це допомагає захистити дані в разі фізичного доступу до пристрою.

4. Шифрування певного застосунку передбачає шифрування даних, які обробляються конкретним додатком. Шифровані дані можуть бути використані тільки в межах додатку і не будуть доступними для інших процесів.

Важливо вибрати належний метод шифрування відповідно до конкретних потреб і потенційних загроз та ефективно управляти ключами. Звісно, це потребує ретельного планування та застосування

найкращих практик. Слід додати, що застосування методів шифрування може призвести до збільшення обчислювального навантаження.

Окрім шифрування даних та MFA, також потрібно продумати алгоритми для захисту від SQL-ін'єкції [6, с. 32] та XSS – вразливості крос-сайтового скриптування. Захист від SQL-ін'єкцій та атак XSS є важливою частиною розробки безпечних додатків та веб-сайтів [3, с. 137].

SQL-ін'єкція – це атака, під час якої зловмисник використовує некоректно оброблені вхідні дані для виконання зловмисних SQL-запитів до бази даних. Наслідки SQL-ін'єкцій можуть призвести до втрати конфіденційної інформації, порушення цілісності даних, а також до заборонених дій з базою даних, наприклад видалення таблиць або записів [11]. Слід додати, що мова структурованих запитів SQL призначена для створення, модифікації та управління даними в реляційних базах даних та синтаксично представлена у вигляді текстового рядку, який містить оператори та параметри [2, с. 20]. Як наслідок, вставити зловмисний код в SQL-запит не є такою важкою проблемою. Найпоширенішою формою SQL-ін'єкції є вставка SQL-коду у веб-форми, URL-адреси та інші вхідні дані.

Для захисту від SQL-ін'єкцій слід порекомендувати наступні засоби:

- використовувати параметризовані запити, тобто замість вставки змінних безпосередньо в SQL-запити потрібно використовувати параметризовані запити, для вставки даних безпечним способом;
- екранувати вхідні дані, а саме всі вхідні дані, що надходять від користувачів, повинні бути екрановані для видалення спеціальних символів та заборонених послідовностей;
- валідувати дані, а саме встановлювати правила валідації для вхідних даних, аби відсіяти некоректні дані перед їх обробкою.

Як було зазначено вище, атаки крос-сайтового скриптування полягають у вставці у веб-сторінки або додатки шкідливого JavaScript-коду, який виконується в браузері користувача Це дає зловмиснику можливість здійснювати дії від імені користувача та отримувати доступ до його сесійних файлів та інших конфіденційних даних. Наслідки XSS атак включають в себе втрату конфіденційних даних [12, с. 102], можливість виконання небезпечних дій на веб-сайті або додатку, а також порушення приватності користувачів.

Для захисту від атак XSS потрібно:

- екранувати виходи – всі дані, які виводяться на веб-сторінці, повинні бути екрановані для видалення спеціальних символів та JavaScript-кодів;
- використовувати заголовки безпеки; саме налаштування заголовків безпеки в HTTP-відповідях допомагає запобігти виконанню зловмисного JavaScript-коду в браузері;
- перевіряти та фільтрувати вхідні дані, зокрема всі вхідні дані від користувачів повинні бути оброблені та очищені перед виведенням на сторінку.

Крім відсутності достатньої автентифікації, слабкого шифрування або різноманітних зловмисних ін'єкції слід звернути увагу на ще один важливий аспект безпеки. Це витік даних, що являє собою незаконне або несанкціоноване розголошення конфіденційної інформації третім особам. Наслідки витоків даних можуть бути катастрофічними (фінансові збитки, втрата репутації, правові наслідки тощо).

Для попередження витоків даних важливо належним чином здійснювати моніторинг інформаційних потоків. Наприклад, компанії повинні визначити для себе, яка інформація є найбільш цінною та конфіденційною та слідкувати за каналами її передачі/використання. Також важливо обмежити доступ до конфіденційної інформації тільки співробітникам з відповідним доступом. Наступним кроком є встановлення систем моніторингу, які відстежують рух даних в мережі та сповіщають про будь-які підозрілі дії. Наприклад, дієвим захистом від Internat-втручань є використання сніферів, призначених для аналізу вихідних TCP з'єднань [10, с. 138].

Також потрібно забезпечити можливість виявлення аномальних паттернів в поведінці користувачів системи, що може свідчити про витік даних. Швидке виявлення витоків даних є важливим для мінімізації їхніх наслідків. Ретельний аналіз журналів подій може розкрити підозрілі активності. Для виявлення витоків даних використовуються такі методи та інструменти:

- IDS (системи виявлення вторгнень) виявляють незвичайну або підозрілу активність в мережі та сповіщають адміністраторів про можливі витоки даних;
- ADS (системи виявлення аномалій) аналізують великі обсяги даних та виявляють аномальні паттерни, які можуть свідчити про витік;

– AAMS (автоматизовані системи моніторингу доступу) дозволяють виявляти незвичайні спроби доступу до систем та даних.

Важливо створити культуру безпеки. Для цього варто використовувати DevOps методологію (від англ. Development і Operations) – це методологія розробки програмного забезпечення та управління IT-інфраструктурою, спрямована на автоматизацію безперервної доставки нових оновлень програмного забезпечення, гарантуючи їх правильність та надійність [13, с. 2] через покращення співпраці між розробниками програмного забезпечення (Development) і операторами (Operations).

Основні принципи та характеристики DevOps включають:

1. Автоматизація використання автоматизованих інструментів для розробки, тестування, розгортання та моніторингу програмного забезпечення. Це допомагає позбутися ручних процесів і зменшує ризик помилок;

2. CI (безперервна інтеграція) – систематичне об’єднання програмного коду розробки в спільний репозиторій, де він автоматично тестується. Це дозволяє розробникам вчасно виявляти і виправляти помилки;

3. CD (безперервне розгортання) – постійне автоматичне розгортання коду в продуктивному середовищі після успішного проходження тестів. Це допомагає зменшити час між внесенням змін та їх впровадженням.

4. Систематичний моніторинг програмного забезпечення в реальному часі та використання зворотного зв’язку для виявлення та виправлення проблем.

5. Культура якості – DevOps сприяє впровадженню культури співпраці, довіри та відкритості між розробниками та операторами. Це важливо для вирішення конфліктів, забезпечення спільної мети, що полягає у надійній та швидкій розробці програмного забезпечення [9, с. 30].

Висновки. Отже, захист на рівні додатків – це критичний аспект в розробці та використанні сучасних програмних додатків. Розробники та організації повинні приділяти велику увагу заходам безпеки на рівні додатків, щоб захистити себе та своїх користувачів від кіберзагроз. На

шляху до розробки та використання додатків безпека повинна бути завжди на першому плані.

До переваг MFA можна віднести підвищену безпеку за рахунок використання декількох факторів підтвердження особистості користувача. Це може бути введення біометричних даних, таких як відбитки пальців, розпізнавання обличчя, власний голос, а також одноразові коди, які генеруються на мобільних пристроях. Такий метод суттєво ускладнює дії зловмисників, які намагаються незаконно отримати доступ до системи або даних.

Використання сучасних алгоритмів шифрування забезпечує конфіденційність даних під час зберігання та передачі даних, запобігаючи їх перехопленню зловмисниками.

Науковці радять слідкувати за правильною обробкою та валідацією вхідних даних, які надходять від користувачів. Використання параметризованих запитів до бази даних та обмеження виконання JavaScript-коду на сторінках додатків допомагає запобігти можливим атакам. Потрібно розуміти сутності цих загроз та вживати відповідні заходи захисту. Це допомагає запобігти можливим атакам і забезпечити безпеку користувачів та конфіденційність даних.

DevOps допомагає організаціям створювати та впроваджувати новий функціонал швидше, зменшувати ризики помилок під час впровадження змін та поліпшувати якість програмного забезпечення. Ця методологія стала популярною в організаціях, що прагнуть досягнути більшої гнучкості та ефективності у сфері розробки та управління IT-проектами.

Література

1. Думанська Н. О. Шифрування даних в інформаційних системах. *Тези доповідей V студентської вузівської наукової конференції «Математичні методи, моделі та інформаційні технології в управлінні підприємством»* (9 листопада 2020 р., м. Вінниця). 2020. С. 23–25.
2. Йолкіна А. С., Шаров С. В. Особливості використання мови SQL для обробки даних. *Інформаційні технології проектування:* зб. наук. пр. магістрантів та студентів. 2013. С. 18–24.

3. Коваленко О. В. Оцінка ефективності технології тестування безпеки. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. 2018. № 29 (68). С. 137–141.
4. Колесников В. А. Порівняльна характеристика симетричного та асиметричного шифрування. *Збірник тез доповідей Міжнародної наукової інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення»* (12 червня 2018 р., м. Тернопіль). Вип. 29. 2018. С. 23–24.
5. Колмакова В. О. Імерсивні технології як сучасна освітня стратегія підготовки майбутніх фахівців. *Українські студії в європейському контексті*: зб. наук. пр. 2022. № 5. С. 177–182.
6. Луговський Б. М., Кузнєцов П. В. Підхід для виявлення SQL-ін'єкцій. *Теоретичні та практичні дослідження молодих вчених*: зб. тез доп. 14-ї Міжнар. наук.-практ. конф. магістрантів та аспірантів (1-4 грудня 2020 р., м. Харків). 2020. С. 31–32.
7. Мірошниченко М. Ю. До проблеми захисту інформаційних систем. *Сучасний стан та перспективи розвитку електротехнічних систем: матеріали IV Всеукр. наук.-практ. інтернет-конференції пам'яті В.В. Овчарова* (04-18 листопада 2021 р., м. Мелітополь). 2021. С. 120–122.
8. Славінський В. О. Методи багатофакторної автентифікації до веб додатків за допомогою штучного інтелекту та технології блокчейн : магістерська дис. : 122 Комп'ютерні науки / Славінський Всеволод Олександрович. Київ, 2022. 124 с.
9. Сусукайло В. А. Використання підходу devsecops для аналізу сучасних загроз інформаційної безпеки. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*. 2021. № 2(14). С. 26–35.
10. Шаров С. В., Лубко Д. В. Розробка та використання сніферу як засобу забезпечення безпеки TCP з'єднань. *Системи обробки інформації*. 2017. № 5. С. 138–144.
11. Clarke J. et al. SQL Injection Attacks and Defense. Syngress Publishing, Inc. Elsevier, Inc., 2009. 473 p.
12. Grossman J. et al. XSS Attacks: Cross Site Scripting Exploits and Defense. MA: Syngress, Elsevier, 2007. 463 p.
13. Leite L. et al. A survey of DevOps concepts and challenges. *ACM Computing Surveys (CSUR)*. 2019. Т. 52. № 6. Pp. 1–35.