Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему
Метод та система підтримки перевірки веб-додатків на вразливості

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

КРМКБ. 2201142.22.01.24 ПЗ

Виконав: студент 2 курсу, група КБм-22-1 Підпис Чвалов А.А.

Керівник доц., к.т.н, доцент Підпис Кльоц Ю.П.

Нормоконтролер старший викладач Підпис Мостовий С.В.

До захисту допускаю:
Зав. кафедри кібербезпеки, к.т.н., доц Підпис Кльоц Ю.П.

20 грудня 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Освітній рівень Магістр

Галузь знань 12 Інформаційних технологій

Спеціальність 125 Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"30" 08 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Чвалова Андрія Анатолійовича

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод та система підтримки перевірки веб-додатків на вразливості

Керівник роботи Кльоц Юрій Павлович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

кандидат технічних наук, доцент

Затверджена наказом № 30 ректора університету, додаток №25 від 15.08.2023

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2023

3. Вихідні дані до проекту (роботи) Розробка методів виявлення та захисту веб-додатків від кібератак, аналіз вразливостей та створення моделі захисту, оцінка ефективності захисних механізмів та розробка інтегрованої системи ідентифікації загроз

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Технології виявлення вразливостей у веб-додатках. Методи побудови систем підтримки прийняття рішень. Методика підтримки перевірки веб-додатків на вразливості. Система підтримки перевірки веб-додатків на вразливості. Висновки та пропозиції.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри кібербезпеки		

7. Дата видачі завдання «01» вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження і узгодження тематики КРМ з керівником	01.06.2023	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	04.09.2023	
3	Робота над розділом 1 – види вразливостей у веб-додатках; підходи та методи виявлення вразливостей; системи підтримки користувачів; постановка задачі	18.09.2023	
4	Робота над розділом 2 – розробка алгоритмів та методів побудови систем підтримки та прийняття рішень	02.10.2023	
5	Робота над розділом 3 – розробка методики підтримки та перевірки веб-додатків на вразливості	16.10.2023	
6	Робота над розділом 4 – аналіз ефективності системи перевірки веб-додатків на вразливості	06.11.2023	
7	Робота над науковою публікацією	10.11.2023	
8	Узгодження отриманих результатів, оформлення пояснювальної записки згідно вимог	15.11.2023	
9	Попередній захист роботи	17.11.2023	
10	Захист роботи на засіданні ЕК	06.12.2023	

Студент _____ А.А. Чвалов
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ Ю.П. Кльоц
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод та система підтримки перевірки веб-додатків на вразливості

Автор роботи: Чвалов Андрій Анатолійович

Керівник роботи: зав. кафедри, Кльоц Юрій Павлович

Загальний обсяг роботи: 97 сторінок, 13 рисунків, 29 таблиць, 1 додаток, 50 посилань.

Ключові слова: виявлення вразливостей, веб-додатки, система підтримки.

Системи підтримки перевірки веб-додатків на вразливості зосереджені на виявленні та захисті веб-додатків від вразливостей. В умовах постійної еволюції кіберзагроз і методів атак, ефективний захист веб-додатків вимагає глибокого розуміння загроз та передових методів захисту.

Дана робота зосереджена на вивченні існуючих методик і моделей для захисту веб-додатків від кібератак та аналізу їх ефективності. Основна увага приділена виявленню та нейтралізації SQL-ін'єкцій, які можуть призвести до серйозних загроз. Основною частиною дослідження є розробка моделі захисту, яка враховує різноманітні механізми атак та пропонує ефективні методи їх нейтралізації.

19.12.2023

ANNOTATION

Theme of qualification work: Method and system to support web application vulnerability checking

Author of the work: Chvalov Andrii Anatoliiovych

Mentor: Ph.D., Assoc. Klots Yurii Pavlovich

Total volume of work: 97 pages, 13 figures, 29 tables, 1 appendix, 50 references.

Keywords: vulnerability detection, web applications, support system.

Support systems for web application vulnerability checking focus on detecting and protecting of vulnerabilities in web applications. Given the constant evolution of cyber threats and attack methods, effective web application security requires a deep understanding of threats and best practices.

This paper focuses on the study of existing techniques and models for protecting web applications from cyber attacks and analysing their effectiveness. The focus is on detecting and neutralising SQL injections that can lead to serious threats. The main part of the research is the development of a protection model that takes into account various attack mechanisms and offers effective methods of neutralising them.

19.12.2023

ЗМІСТ

ВСТУП

Актуальність теми. Технічний розвиток сьогодення надає широкий дистанційний доступ до інформаційних ресурсів, окрім полегшення шляхів отримання інформації та зменшення часу на її обробку, зростає рівень ризику несанкціонованого доступу до інформації обмеженого доступу. Зловмисники можуть використовувати для цього цілий ряд засобів, одним з яких є ін’єкційна атака на визначений інформаційний ресурс, тобто застосування шкідливого коду спрямованого на веб-додаток чи сервер із використанням зовнішніх запитів з метою виведення з ладу інформаційної системи чи бази даних, що розташовані в ній і для стороннього використання не призначені.

Враховуючи це, важливим стає вивчення методів проведення різноманитних атак та методів їх захисту. Розробка ефективних моделей для діагностики вразливостей цих ресурсів дозволить не тільки знизити ризики від зловмисних втручань, а й визначити оптимальні стратегії захисту, залежно від потенційної небезпеки атак та важливості зберіганої інформації.

Мета роботи полягає в розробці системи підтримки перевірки веб-додатків на вразливості.

Об’єктом дослідження є методи боротьби з вразливостями веб-додатків.

Предметом дослідження є розробка системи оцінки ступеню захищеності інформаційних ресурсів у веб-просторі.

Для досягнення цієї мети в роботі необхідно виконати наступні завдання:

- розглянути технології виявлення вразливостей у веб-додатках;
- оцінити методи побудови систем підтримки прийняття рішень;
- розробити методику підтримки перевірки веб-додатків на вразливості;
- виконати розробку вимог до програмної реалізації системи підтримки перевірки веб-додатків на вразливості;
- надати аналіз ефективності модулю прийняття рішень системи підтримки перевірки веб-додатків на вразливості;

Методологічна основа та методи дослідження: в основі роботи послужили загальнонаукові та спеціальні методи дослідження, включаючи історичний, аналітичний, методики аналізу та синтезу, порівняння аналіз тощо. Аналітичний та історичний метод було застосовано для аналізу наукової бази стосовно ін’єкційних атак та існуючих методик аналізу ризиків в наслідок їх виникнення. Теоретико-методологічною основу роботи становлять наукові концепції, представлені вітчизняними та закордонними науковцями, законодавчі нормативно-правові акти України у галузі інформатизації та захисту інформації. Методи аналізу та синтезу використано для аналізу існуючих моделей та розроблених засобів захисту інформації інтернет-ресурсів. Метод порівняння використано під час визначення рівня ефективності методик захисту від впливу ін’єкційних атак.

Інформаційну базу дослідження становлять наукові розробки, праці вітчизняних учених, офіційні документи, звітні дані. У цьому контексті застосовано передові наукові методики у сфері моделювання складних систем, автоматизованих процесів, загальної теорії управління, автоматизованих систем управління, інформаційної безпеки, захисту інформації, теорії прийняття рішень та інших аспектів.

Структура роботи. Структурно робота складається зі вступу, трьох основних розділів із підрозділами, висновків і списку використаних джерел, який містить 50 джерел.

1 ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ У ВЕБ-ДОДАТКАХ

1.1 Види вразливостей веб-додатків та їх наслідки

Серед безлічі вразливостей, які можуть поставити під загрозу безпеку веб-додатків, дві найбільш відомі – це ін’єкція SQL і міжсайтовий скрипт XSS [44, 50].

Міжсайтовий скриптинг є дуже поширеною формою атаки, спрямованої на веб-додатки. У цьому типі атаки, відомому як XSS, зловмисник вставляє потенційно шкідливий код JavaScript у сценарії на стороні клієнта. Після введення коду він стає активним коли відкривається веб-сторінка, що дозволяє зловмиснику реалізувати свої мерзенні плани. Змінюючи код, зловмисник отримує неавторизований доступ до особистої інформації користувача, коли він натискає певну URL-адресу. Крім того, ця атака XSS також може маніпулювати веб-сторінкою програми веб-сайту, перенаправляючи законних користувачів на оманливі або шахрайські веб-сайти.

Другим заходом є перевірка вхідних даних, яка спрямована на те, щоб веб-додаток доставляв лише надійні дані. Це важливо для запобігання заподіянню шкоди базі даних, веб-програмі або компрометації особистих даних кінцевого користувача ненадійними або шкідливими даними. Хоча перевірка вхідних даних зазвичай асоціюється із запобіганням атакам SQL-ін’єкцій, вона також може ефективно перешкоджати атакам XSS. Багато авторитетних веб-сайтів використовують методи перевірки введених даних, які обмежують введення спеціальних символів у текстові поля, тим самим зміцнюючи їхній захист від атак XSS. Третій захід — очищення введених користувачем даних. Очищення даних передбачає зміну вхідних даних для забезпечення їх дійсності. Одним із підходів до досягнення цього є укладення отриманих даних у подвійні лапки, що особливо корисно для веб-додатків, які використовують розмітку HTML. Завдяки перетворенню недійсних даних у дійсний формат веб-додаток і базу даних можна захистити від потенційної шкоди [14, 37].

Уразливості XSS можна пом’якшити за допомогою трьох найважливіших заходів. Першим заходом є запобігання витоку даних, що передбачає захист даних, отриманих веб-програмою, перш ніж вони стануть доступними для кінцевого користувача. Це досягається шляхом цензури отриманих даних і заборони виводу певних символів, таких як '<' і '>', які можуть бути використані у шкідливі способи. Переконавшись, що користувачі не зможуть додавати власний код на веб-сторінку, можна легко уникнути ризику сценаріїв JavaScript і HTML. Таким чином, щоб пом’якшити вразливості XSS, надзвичайно важливо впровадити такі заходи, як запобігання витоку даних, перевірка введених даних і санітарна обробка даних. Використовуючи ці методи, веб-програми можуть значно підвищити свою безпеку та захистити від зловмисних атак.

Для дослідження методів реалізації ін’єкційних атак вважаємо за потрібне спочатку визначити сутність даного поняття. Отже ін’єкційна атака – це процес, в наслідок якого зловмисник може отримати доступ до операційної системи або доставити шкідливий код у веб-додаток, щоб мати доступ до конференційної інформацію чи системи. Своїми діями зловмисник змушує операційну систему сприймати команди, наче вони були ініційовані законим користувачем, так операційна система починає обробляти несанкціоновану команду. Під час такого зловмисник здатний отримати будь-яку необхідну йому інформацію з інформаційних ресурсів шляхом отримання доступу через ін’єкцію.

Визначають такі методи реалізації ін’єкційних атак [48]:

– Кодова ін’єкція (Code Injection)
– SQL ін’єкція (SQL Injection)
– Командна ін’єкція (Command Injection)
– Міжсайтовий сценарій XSS (Cross-Site Scripting)
– XPath ін’єкція (XPath Injection)
– Ін’єкція через пошту (Mail Command Injection)
– Ін’єкція CRLF (CRLF Injection)
– Ін’єкція заголовка хоста (Host Header Injection)

– Ін’єкція ЛДАП (LDAP Injection)

Найбільш розповсюдженою вразливістю є SQL ін’єкції, ож доцільно розглянути саме їх механізми більш детально. Успішна атака з застосуванням SQL-ін'єкцій може призвести до несанкціонованого доступу до бази даних (наприклад, дані кредитної картки, паролі або інші конфіденційні дані користувач). Останнім часом багато гучних витоків інформації відбулися в наслідок атак з використанням SQL-ін'єкцій, що завдало збитків репутації та застосування штрафів з боку контролюючих органів [6, 27].

Найчастіші приклади впровадження SQL:

– UNION атаки, в наслідок якої отримують дані з різних таблиць інформаційної бази даних веб-ресурсу;

– отримання прихованих конфіденційних даних, де ін’єкція SQL змінюється задля отримання додаткових результатів;

– сліпе впровадження SQL, при якому результати запиту не відстежуються у відповідях веб-ресурсу;

– підрив логіки веб-додатку, при цьому змінюється запит, щоб заважати правильній логіці застосунку;

– дослідження бази даних, де відкривається інформація про версії і структури бази даних [25].

Цей тип атаки вважається одною із найнебезпечніших загроз у сфері кібербезпеки.

Часто веб-додатки піддаються ризику через SQL атаки, що дозволяють зловмисникам легко отримати контроль над базою даних. При атаках із використанням SQL ін'єкцій зловмисник вводить спеціальний код у поля вводу, що призводить до некоректної роботи веб-програми, відхиляючись від оригінального наміру розробника. Таким чином, цей тип кодової ін'єкції доцільно детально розглянути.

Встановлено, що атаки SQL Injection (SQLIA) вважаються найпоширенішою із загроз безпеки для інформаційних баз даних. SQLIA – це категорія атак інєкції

коду, при яких користувач не здійснює перевірку. Фактично, зловмисники здатні отримувати сторонній несанкціонований доступ при дописуванні частки до рядка запиту, що виконується серверною частиною у БД веб-додатку. Веб-додатки у сфері фінансів та системи безпеки банківських даних також піддаються ризику через цю уразливість. Хакери мають змогу порушити приватність, безпеку та недоторканість цих систем. Розробники впроваджують деякі методи захисного шифрування, для створення захисту та закриття цієї вразливості, але цього часто недостатньо.

Основні типи атак, що використовують техніку SQL-ін'єкції.

а) Класична атака (внутрішньо смугова SQL-ін’єкція (класична атака). Цей тип є найпоширенішим, ін'єкція відбувається в основному в той час, коли зловмисник використовує один і той самий канал зв’язку для запуску атаки та збору результатів.

За даним типом внутрішньо смугові SQL-ін'єкції існують у таких різновидах:

1) SQL-ін’єкція на основі помилок користувачів, грунтується на повідомленні про помилку, що формується сервером баз даних, при запиті про отримання інформації стосовно структури;

2) використання SQL-ін’єкцій з урахуванням об’єднання. Цей тип заснований на використанні SQL UNION для поєднання результатів двох або більше операторів SELECT у один, який потім повертається та формується як HTTP-відповідь [15].

б) Сліпа або Blind SQLi (інференційна SQL-ін'єкція). У випадку сліпої SQL-ін’кції зловмисник не здатний побачити результат власної атаки, оскільки дані та інформація не передаються через веб-додаток. Така ін’єкція має назву Blind SQLi або сліпа. Інференціальні SQL-ін'єкції визначають двох типів:

1) бульова сліпа SQLi, вона заснована на створенні SQL-запиту до інформаційної бази даних, який вимагає від програми формувати інший результат залежно запиту: результат «FALSE» або «TRUE»;

2) сліпа SQLi, вона грунтується на часі надсилання відповідного SQL-запиту до бази данних, що вимагає базу даних чекати певний час (у секундах), перш

ніж відповісти. Отже, час відповіді вказує зловмиснику, яким є результат запиту «TRUE» або «FALSE». Атака через застосування SQL стає можливою через некоректну обробку даних, що виявляються під час SQL-запитів.

в) Позасмугова SQL-ін'єкція – виникають, коли атакуючий не здатний використовувати однаковий шлях для проведення нападу та отримання даних з результатів. Цей підхід дозволяє зловмисникам застосовувати нестандартні методи SQL-втручань, які базуються на альтернативних логічних підходах, особливо в ситуаціях, де відповіді сервера є нестабільні і захист, що базується на швидкості відповідей, не є достатньо надійним [15].

Проаналізовані методи залежать від здатності сервера бази даних формувати DNS або HTTP запити для отримання хакером необхідних йому даних.

Зазвичай SQL-ін'єкції виконується в три етапи [16]:

– зловмисник формує шкідливий HTTP-запит до веб-додатку;
– веб-застосунок формує SQL запит включаючи шкідливий код;
– серверна частина виконує SQL-запит та повертає результат виконання;

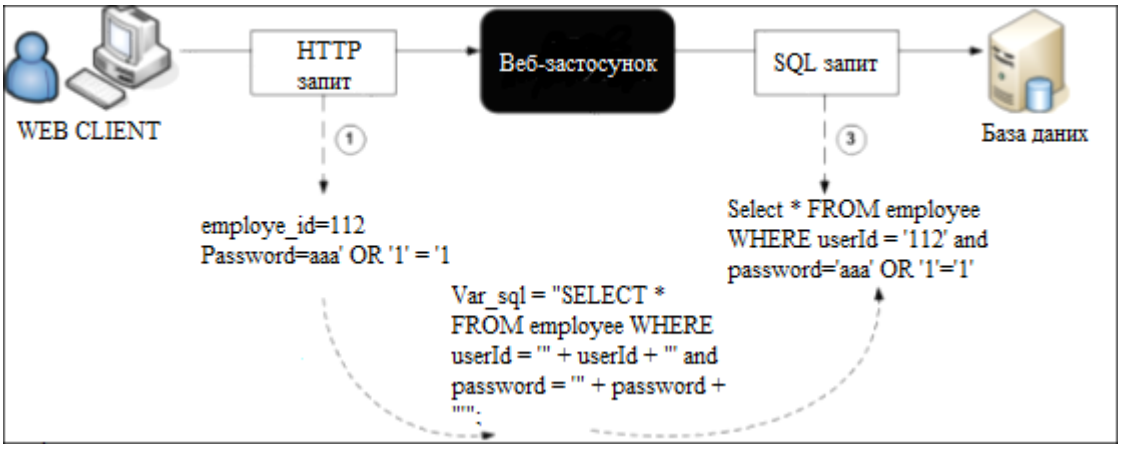

Рисунок 1.1 - Приклад атаки SQL-ін’єкції [23]

Розмір змінних, які використовуються у веб-додатках, також може бути вразливістю. Якщо змінні дозволяють зберігати більше даних, ніж необхідно, це створює можливість для зловмисників вводити модифіковані або підроблені запити SQL. Динамічні SQL-запити, які створюються сценаріями або програмами в рядку

запиту, також уразливі до ін’єкційних атак. Перехоплюючи введені користувачем дані та створюючи пропозиції WHERE в інструкції запиту, зловмисники можуть вводити шкідливий код і маніпулювати базою даних.

Таким чином, дуже важливо знати про ці вразливості, щоб ефективно захистити веб-додатки від атак SQL-ін’єкцій. Застосування належних заходів безпеки та регулярне оновлення та виправлення програми можуть допомогти зменшити ці ризики. Підтримка кількох виразів, таких як функція UNION, у базі даних може надати зловмисникам більше можливостей для успішних атак SQL-ін’єкцій. Наявність під-запитів у базі даних також створює вразливість, оскільки зловмисники можуть вставляти шкідливий код у ці запити. Ще однією вразливістю є права доступу до бази даних. Якщо ці привілеї доступу не визначено належним чином, це може призвести до несанкціонованого доступу та маніпулювання базою даних. Процедури SQL, які зберігаються в базі даних на основі мови SQL, також можуть становити вразливість. Зловмисники можуть виконати ці процедури та потенційно пошкодити базу даних, операційну систему та інші компоненти мережі. Контроль клієнтів – ще одна вразливість, яку слід враховувати. Якщо перевірка введених даних виконується лише для сценаріїв на стороні клієнта, зловмисники можуть перевизначити функції безпеки цих сценаріїв, використовуючи міжплатформні сценарії на проміжних етапах, ставлячи під загрозу безпеку програми. Тип даних змінних – ще одна вразливість, якою можна скористатися. Зловмисники можуть використовувати цю лазівку для зберігання зловмисних даних, що призведе до потенційних порушень безпеки [33].

Розглядаючи атаки SQL-ін’єкцій, важливо враховувати різноманітні вразливості, які природним чином можуть існувати у веб-додатках. Однією з таких вразливостей є недійсний логін, який зазвичай використовується в атаках SQL-ін’єкцій. Це відбувається, коли веб-програма має параметри, які зазвичай використовуються в запитах SQL, і якщо не позначити ці параметри, зловмисники можуть маніпулювати ними. Повідомлення про помилки, які генеруються серверними або серверними програмами, також можуть становити вразливість. Хоча ці повідомлення про помилки корисні для налагодження під час розробки,

вони також можуть надати цінну інформацію зловмисникам, підвищуючи ризики для програми.

Деякі типи введення коду формують з помилками інтерпретації, які лише видають особливого значення для користувача. Такі помилки інтерпретації можуть існувати і за межами інформаційних технологій та веб-ресурсів, наприклад, у повсякденному житті власні імена іноді складно відрізнити від звичайних слів. Так і з деякими варіантами введення коду – неможливо відрізнити код введення користувачем від системних регламентованих команд. Але для розробників програмного забезпечення це не викликає складнощів і обійти цю заборону фахівці здатні досить легко.

Технології ін'єкції коду мають популярність у системах злому при отриманні інформації, розширенні можливостей або стороннього несанкціонованого доступу. Існують різні цілі, для яких ін’єкція коду може бути використана зловмисно. Одним з прикладів їх використання є зміна інформації в базах даних через SQL-ін'єкції, що може призвести до неправильної роботи веб-додатків або витоку конфіденційної інформації. Крім того, ін’єкція коду може бути використана для інсталяції шкідливого програмного забезпечення чи доставки шкідливого коду на сервер, для подальшого виконання скриптового сценарію на боці сервера, наприклад ASP або PHP. Іншим потенційним зловмисним використанням впровадження коду є привілейована ескалація до root. Цього можна досягти, скориставшись уразливістю Shell у двійкових файлах rootuid UNIX або локальних системах, на яких працює служба Windows. Крім того, ін’єкція коду може бути використана як засіб атаки на користувачів Інтернету за допомогою ін’єкції HTML/Script. Однак важливо зазначити, що ін’єкційний код також можна використовувати з благими намірами. Наприклад, ін’єкція коду може бути використана для зміни чи коригування поведінки програми чи системи без будь-яких зловмисних намірів. Це може служити способом «змусити» систему виконувати певне завдання, пропонуючи гнучкість і налаштування [38].

Крім того, ін’єкція коду може бути використана як засіб атаки на користувачів Інтернету за допомогою ін’єкції HTML/Script. Під час введення коду

він може ініціювати різноманітні дії. Наприклад, він може створити презентацію додаткового стовпця, який спочатку не був видимий на сторінці результатів пошуку. Крім того, він може запропонувати інноваційний метод фільтрації, організації або групування даних за допомогою поля, яке не входить до стандартних функцій оригінального проекту. Крім того, код дозволяє включати певні компоненти, які забезпечують підключення до онлайн-ресурсів в рамках офлайн-програми.

Іншим уразливим використанням коду може бути з'ясування найбільш помилкових дефектів під час усунення цих недоліків. Науковці визначають цей метод як тест «білого капелюха», або тест на проникнення. Даний тест вважається робочою процедурою, коли програміст-експерт перевіряє програмне забезпечення на вразливості, він здійснює санкціоновані спроби зламу. При цьому, якщо підозра на вторгнення не пов'язана із робочими санкціонованими процедурами, варто вжити заходи щодо протидії ін'єкцій.

Автоматизована загроза відноситься до типу загрози безпеці комп'ютера, яка передбачає використання програмного забезпечення, спеціально розробленого для виконання численних повторюваних завдань. Ці завдання виконуються за допомогою засобів автоматизації, таких як інтернет-боти. Однак впровадження технології виявлення ботів у режимі реального часу може значно допомогти у запобіганні та зменшенні автоматизованих загроз. Приклади автоматизованих загроз включають накопичення рахунків, проблеми, пов'язані з картками, і атаки на відмову в обслуговуванні (DoS).

1.2 Підходи до виявлення вразливостей та методи

Ін'єкційні атаки найкраще попереджати, відслідковувати та блокувати раніше, ніж зловмисник встигне повністю захопити систему. Отже розглянемо окремі моделі, які допоможуть ідентифікувати ін'єкційні атаки.

Максимально ефективніший спосіб визначення вразливостей перед використанням ін'єкцій – це впроваджувати у локальній мережі компанії

автоматизований веб-сканер вразливостей. Наявність загрози можна також виявити вручну за допомогою тесту на проникнення, однак це може займати більше часу та ресурсів.

Автоматичний сканер швидше реагує на повідомлення про загрозу і допомагає створити захисну відповідь для протидії злочинним атакам.

Ідентифікувати кодову атаку можна за допомогою аналізу змін у моделях трафіку та відхилень від нормального використання. Для реалізації цього рішення до мережі додається два додаткових пристрої. Один пристрій відповідає за моніторинг вхідного трафіку та виявлення будь-яких ознак атаки, а інший пристрій фільтрує зовнішній трафік. Однак у разі атаки пристрій «скруббер» перехоплює будь-який трафік, який вважається зловмисним, запобігаючи його проникненню в обмежені канали та ресурси клієнта. Це гарантує, що основні послуги продовжуватимуть надаватися клієнту, одночасно пом’якшуючи вплив атаки. Пристрій моніторингу на сервері виконує чотири дії для виявлення самої атаки та джерела атаки [21]:

– реєстрація ознак вторгнення шляхом аналізу аномальних змін у трафіку на певних вузлах системи;

– визначення початкових точок атак (веб-додатки, сервер, електронна пошта та інші);

– зупинення потоків даних, що приходять від виявлених джерел агресії, особливо якщо це відбувається з боку зовнішніх атакуючих, а не зсередини локальної мережі;

– оцінка ефективності зупинення атаки та перевірка на відновлення нормального рівня мережевого трафіку;

Використання ентропії мережевого трафіку для ідентифікації потенційних атак базуючись на аналізі коливань ентропійних показників. Цей метод передбачає порівняння короткотермінових показників ентропії (які відображають миттєву невизначеність) з довготривалими значеннями цієї ж характеристики (репрезентують загальну невизначеність у відсутності атак). Такий підхід дозволяє

виявити значні відхилення у локальних показниках від загальноприйнятих норм, що часто свідчить про зростання ризику мережевих атак [22].

Відомі системи виявлення загроз можна розділити на системи виявлення ознак і системи виявлення аномалій. Головним недоліком систем виявлення функцій є те, що частіше вони призначені для встановлення певних типів атак (часто найнебезпечніших на момент формування системи). Коли виявляються нові атаки або трансформуються параметри пересування, задачу виявлення загрози потрібно вирішувати знову. Системи встановлення аномалій (у зв'язку із складністю моделювання нормального інтернет-трафіку) впроваджують різні припущення стосовно функціонування системи, наприклад, відносно статистичну однорідність трафіку. Але групи комп'ютерних систем, відповідно яких застосовуються такі припущення не оцінювалися. Тож незначні зміни у структурі наданих послуг або трафіку можуть викликати необхідність перенавчання алгоритму виявлення загроз. Можливим рішенням цієї ситуації є впровадження комплексного підходу до розробки системи боротьби із зовнішніми атаками, що включає ведення історій транзакцій, моніторингу системи, ведення спеціального репозиторію для інтелектуального аналізу зловмисників та їх дій, а також визначення стратегії. Пропонується побудувати систему захисту на основі наступних елементів [23]:

– заходи попередньої обробки та зберігання;

– агенти відстеження;

– сховище для зберігання інформації стосовно операції, що описують роботу системи;

– сховища з аналітичними компонентами для встановлення загроз та ознак діяльності зловмисників;

– заходи проти нападів.

Вкрай необхідним елементом утворення такої системи є встановлення відповідного математичного забезпечення стосовно кожного етапу роботи [23]:

– Відстеження трафіку. Розробка заходів для оцінки складу трафіку, його навантаження, активності користувачів. З метою забезпечення цього завдання

треба розробити алгоритми відстеження частоти та кількості захоплення пакетів відповідно до навантаження каналу та інших параметрів мережі. Якщо пакети захоплюються дуже часто, трафік сповільнюється або взагалі може утворити «сліпі зони», про які буде відсутня інформація.

– Попередня обробка перехоплених пакетів, аналіз найнебезпечніших загроз, забезпечення зберігання інформації у сховищах. На цьому етапі вкрай важлива швидка оцінка загроз з мінімальними витратами ресурсів, рекомендовані прості та адаптивні варіанти ідентифікації та вимкнення деяких елементів мережі, де зафіксовано ін'єкцію.

– Аналіз даних у момент завантаження в пам'ять, оцінка загрози, виявлення атак. Після збереження та фіксації інформації в репозиторії рекомендують провести комплексну оцінку та встановити можливі ризики. При цьому доцільно застосовувати алгоритми багатоканального моніторингу мережі та розрахунок ковзного середнього щодо трафіку за окремими вузлами.

– Оцінка фонових даних для встановлення спроб сканування, імпульсних атак та загроз погіршення якості. Ця оцінка має проводитися за графіком або на регулярній основі. Ці атаки складають меншу загрозу, однак важливо застосування їх глибокого аналізу. При цьому впроваджуються методи Data Mining, нейронні мережі, інтелектуальні системи правил тощо [34].

– Запровадження процедури виявлення атак. Якщо буде виявлено перевищення зазначених нормативних показників на будь-якому з попередніх етапів, або при встановленні нестандартної активності існує можливість загрози нападу. Відповідно зазначають налаштування експертної системи для оцінки рівня загрози та запровадження рішення про наявність атаки.

– Оцінка ризику, вибір оптимальної моделі, верифікація, пошук стратегії. Виявлення атаки відразу ж ставить питання про розробку протидії. Відповідно типу і окремих характеристик конкретної атаки система заходів протидії може суттєво відрізнятися. Таким чином можна говорити про розробку стратегії протидії атаки. Залежно від якості заходів, наприклад, якість обслуговування зареєстрованих

користувачів, визначення стратегії має змінюватися. Розробка ефективних методів протидії має опиратися на глибоке аналітичне розуміння взаємодії між процесами впровадження нових рішень та механізмами їх захисту [23].

Щоб захиститися від SQL-ін’єкції, часто пропонують впровадження захисного кодування, але це дуже складна система захисту. Розробники намагаються вбудувати окремі елементи керування у код виходу, однак атаки все рівно надають нові способи обійти ці елементи керування. Сучасні новітні методи оборонного кодування стикаються із новими формами ін’єкцій, за якими важко слідкувати розробникам. З іншого боку, застосування найкращих методик захисного кодування виявляється дуже складним для «рядового» кодера і потребує спеціальних навичок. Все це вказує на необхідність вирішення проблеми ін'єкції SQL новими методами.

SQLIA, або SQL-ін'єкції, є методом кібернападу, при якому зловмисники вставляють SQL-команди через інтерфейси користувача веб-застосунків або через параметри, що дають доступ до неочікуваних системних ресурсів. Цей метод стає надзвичайно ефективним через недоліки у перевірці даних, що вводяться користувачами в ці застосунки.

Рекомендовано використовувати для ідентифікації дані, які мають бути включені у SQL-запит для забезпечення захисту від ін'єкції SQL;

– HttpOnly - це сигнал для файлів cookie HTTP, у разі його встановлення виявляється взаємодія клієнтського сценарію з файлами cookie, попереджаючи таким чином окремим атакам XSS;

– вихідне кодування, це перешкоджання атакам при введенні HTML-коду (XSS) на користувачів сайту;

– модульне виокремлення корпусу від ядра;

SQL-ін’єкції іноді застосовуються як параметризовані запити, введення в білий список, збережені процедури тощо, щоб допомогти зменшити проблеми з ін’єкцією коду. Модель ідентифікації повинна базуватися на контролі внесених змін до «білого списку» та стороннього несанкціонованого доступу до окремих реєстрів.

У випадку атак означених програм, які не мають прав доступу до баз даних, абсолютно виключається можливість у зловмисників отримати незаконний доступ до локальних даних або іншої інформації. Коротко розглянемо кращі відомі варіанти захисту від атак SQL-ін’єкції, окреслимо рекомендовані заходи безпеки під час розробки компонентів бази даних та загальні рекомендації щодо конструювання веб-систем із застосуванням серверів баз даних [15].

Перший спосіб поєднують із необхідністю фільтрації інформації, що надходить на сервер: спеціальні символи мають бути перевірені, а числову інформацію звіряють з введеним типом. Ще рекомендовано обмежити введення (наприклад, запити, що перевищують окреслену кількість, інформація, введена після перевірки на сервері, відхиляються) [16].

Другий спосіб – застосування параметричних запитів серверами баз даних. Взагалі параметричні запити – це метод передачі інформації, при якому зовнішні параметри подаються на сервер незалежно від SQL запитів. Для основних мов програмування реалізація цих функцій вже розроблена [16]:

– Delphi - властивість TQuery.Params;
– Java - клас PraparedStatement;
– PHP - властивість MySQLi.
– C # - властивість SqlCommand.Parameters;

Третій спосіб – максимальне обмеження відображення повідомлень стосовно помилок користувача (відображаються лише загальні повідомлення про збої, які для всіх можливі). При цьому на стороні сервера всі невдалі або помилкові запити треба відстежувати, щоб у випадку атаки їх можна було розглянути та проаналізувати.

Періодичне тестування та моніторинг фахівці також вважають досить ефективними методами захисту від SQL ін'єкцій. При цьому найкращий спосіб перевірки – спроба ввести свій код у SQL.

Окремі науковці рекомендують математичний спосіб ідентифікації SQL-ін’єкцій за допомогою використання обмеженої знизу функції, що напряму

залежить від вхідного рядка. Цей метод виявлення атак з використанням набору числових символів надає можливість більш точно встановити вразливість виду SQL-ін’єкції. На початку створюється набір символів, що поєднуються як з атакою, так і з запитами користувачів, відповідно раніше відомому порогу, застосовуючи приблизні дані атак та нормальних запитів. Експериментами зі штучними даними показали, що окреслений набір містить пробіл, крапку з комою і праву дужку, що максимально підходить для ідентифікації атаки чи нормального запиту [17].

Крім цього можна впроваджувати різноманітні додатки, які розпізнають несанкціоновані введення чи якусь іншу підозрілу активність. Ось деякі з них:

WAVES — це технологічна система Blackbox для запровадження тестування веб-додатків на можливість появи SQL. Інструмент аналізує всі точки веб-додатків, які можна використати для впровадження SQL-ін’єкції.

JDBC-Checker – може бути застосований для протидії атакам на невідповідність типів у динамічно утвореному ланцюжку запитів.

Перевірка запитів SQL Guard і SQL Check відбувається в момент виконання, де відбувається перевірка на попередньо визначену граматичну модель, яка допускає лише дійсні запити. Цей процес перевірки передбачає аналіз структури запиту як до, так і після введення даних користувача, все на основі встановленої моделі.

AMNESIA – поєднує статичний аналіз і моніторинг продуктивності. Під час статичної фази він моделює різні типи запитів, при цьому програма може легально утворювати їх в кожній точці доступу до бази даних. Запити перехоплюються перед відправкою в базу даних і перевіряються зі статично побудованими моделями на динамічній фазі.

WebSSARI – використовує статичний аналіз для перевірки потоків перешкод від передумов для чутливих функцій. Даний інструмент працює на основі очищеного впускного отвору, який пройшов через визначений набір фільтрів. Недоліками цього підходу є встановлена передумова, оскільки чутлива функція не може бути точно виражена, тому деякі фільтри можна пропустити.

SecuriFly – ще один інструмент, який був реалізований для Java. SecurityFly

намагається вилікувати рядки запиту, які були згенеровані за допомогою введених користувачем даних, але, на жаль, ін’єкція у числові поля проникає дуже швидко і її не можна зупинити за таким підходом. Основним обмеженням цього підходу є складність визначення всіх джерел користувацьких даних.

IDS використовує систему виявлення вторгнень (IDS) для виявлення SQL-ін’єкції на базі методів навчання комп’ютера. Технологічна система формує моделі для поширених запитів, далі під час здійснення запиту, який не відповідає моделі, його буде ідентифіковано як атаку. Цей інструмент показав суттєву ефективність для виявлення атак, але дуже залежить від навчання. Іншими словами, буде багато помилкових і помилково негативних результатів.

Іншим методом у цій групі є SQL-IDS, який створений на принципі написання специфікацій для веб-додатків, що окреслюють передбачувану структуру SQL-запитів, утворених програмою, а також на результатах автоматичного моніторингу здійснення цих запитів SQL на предмет порушень даних специфікацій.

SQLPrevent складається з перехоплювача запитів HTTP. Початковий потік даних змінюється після розгортання на веб-сервері SQLPrevent. HTTP-запити зберігаються у поточному локальному сховищі [19].

Отже, розроблено ряд методів ідентифікації загрози несанкціонованого проникнення, однак вони не систематизовані та розрізнені, тому дана проблема потребує подальших більш детальних досліджень.

1.3 Системи підтримки користувачів

Існує шість ключових етапів, які критично важливі для ефективної стратегії захисту від наслідків ін'єкційних кібератак.

– Оцінювання ризиків ін'єкційних кібератак.

На першому етапі ми зосереджуємося на розробці стандартів, які бізнес повинен впровадити для керування ризиками у майбутньому. Це означає, що кожен сегмент та рівень організації має використовувати єдиний підхід до оцінки ризиків.

Тут важливо вирішити, чи буде оцінка ризиків виконуватися якісно чи кількісно, які масштаби використовувати для відстеження і який рівень ризику ін'єкційних атак є прийнятним.

– Реалізація оцінки ризиків ін'єкційних кібератак.

На другому етапі, з огляду на те, що аналіз ризику інформаційної безпеки здійснюється для активів, пов'язаних з обробкою даних, ідентифікуємо активи, що можуть бути вразливі до ін'єкційних атак та визначаємо потенційні загрози, аналізуємо вплив та ймовірність кожної з можливих комбінацій "актив - загроза - вразливість" і розраховуємо загальний ризик виникнення ін'єкційних атак.

– Впровадження стратегії управління ризиками ін'єкційних кібератак.

На третьому етапі звертаємо увагу на те, що не всі виявлені ризики мають однакову важливість. Наша мета - визначити ті ризики, які можуть завдати найбільшої шкоди системі, та зосередити увагу на цих критичних ризиках. Потрібно пам'ятати, що найбільшу загрозу становить не сама ін'єкційна атака, а значні втрати, які вона може спричинити.

– Звіт про аналіз ризиків системи захисту від ін’єкційних атак.

На цьому етапі формується звіт, який документує всі попередні кроки. Його часто використовують як для аудиту, так і для контролю власних результатів.

– Положення щодо впровадження системи захисту від ін’єкційних атак.

Цей документ реально показує профіль безпеки окресленої системи – за результатами системи управління ризиками рекомендують перевіряти всі впроваджені інструменти управління безпекою, встановити їх доцільність та проаналізувати процес впровадження. Це положення є дуже важливим, оскільки застосовується як основний документ у процесі перевірки працездатності методу захисту.

– План управління ризиками впливу ін’єкційних атак.

Розробка даного плану ґрунтується на чіткому визначенні того, хто саме буде впроваджувати інструменти управління безпекою, за яких умов, у який період тощо. Отже, залежно від визначеного рівня ризику на даний момент часу

виконується конкретний алгоритм реагування на атаку.

Основними загрозами інформаційної безпеки під час ін’єкційних атак стосовно системи є:

– Неконтрольований доступ. У сучасному світі, більшість даних створюється та обробляється через комп'ютеризовані технології. Ця інформація зазвичай циркулює всередині організацій або між організаціями та їхніми клієнтами через загальнодоступні телекомунікаційні мережі (простіші шляхи доступу). Великі обсяги цих даних мають конфіденційний характер. Порівнюючи ризик неволоділого викриття конфіденційної інформації в електронних системах обробки даних із традиційними методами, стає ясно, що характерною особливістю цифрових систем є можливість перенесення великої кількості даних у більш зручний формат, що може відбуватися непомітно в рамках ін'єкційної атаки. Серед вразливих елементів тут виступають сервери, термінали, робочі станції та автоматизовані робочі місця.

– Відмова у обслуговуванні внаслідок ін’єкційної атаки. Ін`єкція сприяє навантаженню на систему, формуючи додаткові запити, збільшуючи можливу кількість запитів, тим самим перевищуючи пропускну спроможність системи. Найбільш руйнівними такі атаки можуть бути у сфері електронних фінансових операцій, особливо в системах платежів, які забезпечують негайні розрахунки. У таких випадках, отримувачі коштів залежать від надходжень для виконання своїх зобов'язань. Збитки від серйозних збоїв в системі через ін'єкційні атаки можуть значно перевищити витрати на відновлення системи, баз даних, або програмного забезпечення. Особливо уразливими є робочі станції та сервери.

– Повномасштабна кібератака. Ін’єкційні вливання можуть завдати наслідку більш серйозних проблем для системи, ніж пригнічення її працездатності чи відсутності стабільного доступу. Можливими наслідками є: викривлення, викрадення, знищення інформації внаслідок неконтрольованої модифікації запиту, пошкодження інформації під час несанкціонованого логічного доступу до бази даних зовнішніми зловмисниками. Тут можливі як репутаційні втрати, так і фінансові, якщо було викрадено/скопійовано персональні дані або ж переведено

кошти у невідомому напрямку. При цьому уразливі місця: робочі станції, термінали, банкомати, сервери.

Розглядаючи систематичний підхід до захисту інформації, необхідно дотримуватися певних критеріїв. Ці вимоги диктують, що захист інформації повинен охоплювати кілька важливих факторів [32]:

– Безперервність. Ця вимога випливає з того факту, що зловмисники шукають способи обійти захист необхідної їм інформації, здійснюючи ін'єкційну атаку на конкретний ресурс;

– Графік. Планування відбувається шляхом розробки для кожної служби детальних стратегій захисту даних від сторонніх ін’єкційних атак у сфері її компетенції з врахуванням загальних цілей управління;

– Цілеспрямований. Захищаються лише ті інформаційні ресурси, які потребують захисту для певної мети, а всі інші ресурси видаляються з публічного доступу через HTTP-запити;

– Фундамент. Захисту підлягають лише інформаційні ресурси, які об'єктивно можуть потрапити під вплив ін'єкційної атаки, втрата яких може призвести до пошкодження об'єкта управління;

– Надійність. Методи та форми захисту від ін’єкційних атак повинні надійно гарантувати невразливість до будь-яких методів нелегітимного вторгнення. Це стосується захисту даних у різних форматах, мовах та на різних типах фізичних носіїв;

– Масштабність. Незалежно від типу та характеру атаки, система безпеки має автоматично блокувати доступ до інформації, використовуючи ефективні засоби контролю, що забезпечують універсальний захист від різних видів загроз;

– Складність. Захист інформаційних ресурсів має бути всебічним, включаючи різноманітні аспекти і методики протидії ін’єкційним атакам. Не можна обмежуватися використанням лише окремих рішень або технологій. Такий всебічний підхід до захисту є важливим, оскільки він розглядає захист як комплексний процес, в якому різні аспекти і компоненти взаємодіють і

підтримують одне одного, створюючи ефективну систему захисту.

Система захисту інформації вимагає виконання наступних вимог [25, с. 121]:

Щоб забезпечити належне функціонування інформаційної системи, важливо мати чітке визначення повноважень і прав користувачів, коли йдеться про доступ до різних типів інформаційних ресурсів. Це включає в себе надання користувачам необхідних повноважень для виконання призначених завдань, а також мінімізує потребу для кількох користувачів спільно використовувати засоби захисту. Вкрай важливо розглядати випадки та спроби несанкціонованого доступу до конфіденційної інформації, а також оцінювати рівень конфіденційності такої інформації. Крім того, важливо забезпечити контроль за справністю засобів захисту та своєчасно реагувати на можливі несправності.

Організацію захисту інформації в автоматизованих системах можна розділити на три рівні залежно від середовища, в якому інформація знаходиться. Цей поділ визначається за критерієм навколишнього середовища і включає соціальне середовище, інженерно-технологічне середовище та середовище людини/машини. Соціальне середовище стосується окремих осіб, спільнот і держави, тоді як інженерно-технологічне середовище включає машини, апаратне забезпечення, програмне забезпечення та автоматизацію. Нарешті, середовище людини/машини передбачає взаємодію між людьми та автоматизованими системами [12].

Важливим аспектом організації інформаційної безпеки є забезпечення захисту інформації, що передбачає класифікацію заходів протидії на різні групи. Як у теоретичному, так і в практичному контексті зазвичай без особливих двозначностей виділяють три різні групи [17, с. 81]:

Існують різні способи забезпечення захисту, як активного, так і пасивного. До активних засобів захисту належать такі тактики, як збір розвідувальних даних, поширення дезінформації, створення шуму для відволікання потенційних загроз тощо. З іншого боку, пасивні засоби захисту включають фізичне встановлення екранів або бар’єрів для запобігання несанкціонованому витоку інформації. Проте найефективнішим підходом до захисту є комплексне поєднання як активних, так і

пасивних заходів, що створює комплексну систему захисту, що забезпечує максимальну безпеку [36].

Для реалізації проаналізованих методів захисту комп'ютерних систем застосовуються універсальні механізми захисту інформації [25; с. 122].

До числа таких механізмів відносяться:

– ідентифікація (іменування й упізнавання), аутентифікація (підтвердження дійсності) і авторизація (присвоєння повноважень) суб'єктів;

– розмежування та контроль доступу до ресурсів системи;

– аналіз й реєстрація подій, що відбуваються у системі;

– контролювання цілісності ресурсів системи.

Системи ідентифікації, аутентифікації й авторизації необхідні для підтвердження суб'єкта, дозволу його роботи у системі, окреслення законності прав суб'єкта на даний об'єкт або на визначенні дії з ним.

Недоліки означених методик виявлені або в їх громіздкості, надмірній кількості дій по відстеженню ймовірності ризиків, або у недостатності критеріїв оцінювання, якщо методика охоплює тільки певну категорію ризиків атак, а не комплекс усіх можливих небезпек та загроз.

Дослідження рівня безпеки інформаційних ресурсів визначається одним з найважливіших етапів розробки системи захисту веб-ресурсу від ін'єкційних атак, оскільки вона надає можливість оцінити реальний рівень захищеності інформації і можливі фінансові витрати на проектування системи захисту. Базовими параметрами такої системи можуть бути:

– ймовірний час захищеного функціонування інформаційного ресурсу за умов заданої довірчої вірогідності;

– величина ймовірності виникнення порушення безпеки інформації за окреслений фіксований час;

– характеристика ймовірності ін'єкційної атаки (інтегральні показники; щільність розподілу ймовірності, функції характеристик тощо);

– функція відновлення втраченої інформації в часі, що оцінюється

відповідним чином;

– поява ін’єкційних атак, що розраховується математичним очікуванням інтервалу між щільністю розподілу і сусідніми порушеннями [34].

Отже, оскільки параметрів може бути багато, обираємо три основні, які можуть бути враховані при створенні моделі захисту.

1.4 Постановка задачі

Виходячи з вищевикладеного, можемо сформувати наступну задачу:

– оглянути методи побудови систем підтримки прийняття рішень, включаючи наступні механізми: призначення систем підтримки прийняття рішень та особливості їх використання; аналіз структури та засобів формування системи підтримки прийняття рішень; розробка алгоритмів прийняття рішень на основі чек-листів.

– проаналізувати методику підтримки перевірки веб-додатків на вразливості, розкривши наступні питання: опис алгоритму синтезу чек-листа для перевірки на вразливості; методи оцінки повноти чек-листа; аналіз обчислювальної складності методів.

– розробити критерії оцінки ефективності системи підтримки перевірки веб-додатків на вразливості, в тому числі: обґрунтування та вибір середовища реалізації; алгоритм роботи системи підтримки перевірки веб-додатків на вразливості; програмні характеристики моделювання СППР; аналіз ефективності модулю прийняття рішень системи підтримки перевірки веб-додатків на вразливості.

2 МЕТОДИ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

2.1 Призначення систем підтримки прийняття рішень та особливості їх використання

Процес прийняття рішення, який розуміється як процес обрання однієї з кількох можливих альтернатив дії, пронизує все життя людини. Більшість рішень ми приймаємо не замислюючись, тому що існує автоматична поведінка, створена багаторічною практикою. Деяким рішенням надається невелика вага, тому ми не надто думаємо, роблячи вибір. І, нарешті, є проблеми вибору, над вирішенням яких людина думає давно. Як правило, ці проблеми унікальні, не повторюються і вимагають розгляду багатьох альтернатив. У таких завданнях новим є або предмет вибору, або обставини, в яких здійснюється вибір.

Проблема прийняття управлінських рішень є найважливішою в менеджменті. Вона займає центральне місце у соціології організацій. Окреслюючи організацію як інструмент управління, численні соціологи та експерти в області теорії управління, починаючи з М. Вебера, безпосередньо акцентують увагу на важливості формування та виконання управлінських рішень. Ефективність керівництва безпосередньо залежить від якості цих рішенью. Зацікавленість соціологів у цьому питанні зумовлена тим, що рішення впливають на всю сукупність взаємодій, що виникають в процесі керування робочою діяльністю та структурою організації. Вони задають напрям цілям, інтересам, взаєминам та стандартам. Аналізуючи повний управлінський цикл, що включає цілепокладання, планування, організацію, координацію, контроль і цілепокладання, легко помітити, що в кінцевому підсумку він виступає у вигляді елементів управління: підготовки та реалізації управлінських рішень. Отже, рішення є основою управління та організації..

У наукових працях існують різноманітні погляди на те, які рішення, ухвалені

в межах організації, можна вважати управлінськими. Деякі експерти включають до цього переліку, наприклад, рішення про наймання або звільнення працівників і т. ін. Це обґрунтовано, оскільки управлінськими можна вважати ті рішення, які спричиняють зміни у внутрішніх відносинах організації.

Таким чином, управлінські рішення завжди пов'язані зі змінами в структурі організації і, як правило, ініціюється керівником чи уповноваженим органом, відповідальним за наслідки їх втілення чи реалізації. Правомочності для прийняття таких рішень чітко визначені у структурних вимогах організації. Проте, кількість людей, які беруть участь у процесі формування рішень, перевищує кількість тих, хто має право їх приймати.

У сучасних організаціях підготовка управлінських рішень часто є окремою функцією, відокремленою від їх затвердження, і може вимагати співпраці цілої групи фахівців. Це особливо характерно для класичного менеджменту, де це задача кадрових служб.

Процес прийняття рішення включає розробку спеціального плану, метою якого є досягнення конкретних цілей. Розробка такого плану здійснюється відповідними службами управління, але з часом у цей процес все частіше залучаються безпосередні підрядники.

Відповідно до загальних принципів менеджменту управлінське рішення повинно відповідати певним вимогам, а саме:

– мати наукове обґрунтування, тобто враховувати об'єктивні закономірності: технологічні, економічні та організаційні особливості об'єкта, на діяльність якого вплине рішення;

– цілеспрямованість, тобто чітке досягнення цілей, поставлених перед суб'єктом господарювання або його відокремленим підрозділом;

– носити кількісний та якісний характер, тобто містити конкретні кількісні показники та математичні розрахунки наслідків реалізації рішення, а також детальний якісний опис тих аспектів, які не піддаються кількісній оцінці;

– відповідати законодавству, тобто не суперечити чинному

законодавству, наказам Міністерства, стандартам, нормам, інструкціям та іншим нормативно-правовим документам;

– бути оптимальним, забезпечувати той варіант рішення, який відповідає економічному критерію ефективності – отримання максимального результату при найменших витратах при збереженні всіх інших аспектів процесу управління;

– пунктуальність, тобто інвестування в заздалегідь визначені терміни підготовки, донесення рішень до конкретних підрядників і контроль за їх виконанням;

– бути комплексним, тобто аналізувати та враховувати всі аспекти управління;

– бути гнучким, тобто вчасно реагувати на зміни в економічному середовищ;

– бути повністю формалізованим, тобто містити вичерпну конкретизацію конкретних способів виконання завдання, необхідних ресурсів, умов виконання, складу виконавців, порядку їх взаємодії, законності документів.

Дотримання всіх вимог до управлінських рішень має важливе значення для забезпечення їх конкретності, адекватної інформації та чіткого розподілу відповідальності за їх виконання. Якщо рішення для управління відповідає вимогам, його можна розгорнути та досягти цілей управління об’єктом.

Загальні підходи до проектування інформаційно-керуючих систем включають 5 етапів:

– аналіз системи управління та прийняття рішень, починається з дослідження всіх типів рішень, які потребують інформації. При цьому необхідно враховувати потреби кожного рівня управління та функціональної сфери;

– аналіз вимог до інформації, що дозволяє визначити, яка саме інформація потрібна для прийняття кожного рішення;

– агрегування рішень - групування їх за завданнями управління та відповідна координація інформаційно-управлінських систем;

– проектування та планування процесу обробки інформації - розробка

реальної системи модифікації, збору, передачі та зберігання нформації;

– проектування та контроль системи управління - полягає у створенні та впровадженні системи оцінки інформації, що видається системою управління інформацією, і дозволяє розпізнавати та виправляти спостережувані помилки. Ефективність інформаційних систем управління залежить від того, чи враховані потреби конкретних менеджерів на етапі проектування, чи навчені користувачі роботі з інформаційною системою управління, від рівня ефективності самої системи і т.д. В останні десятиліття підвищенню ефективності інформаційних систем управління сприяють досягнення в області технологій обробки інформації і впровадження комп'ютерів в процес управління.

Розглянемо набір інструментів нечіткої логіки для прийняття рішень у деяких типових ситуаціях.

Прийняття рішень у незрозумілих умовах. Одним із найпоширеніших застосувань нечіткої логіки є прийняття рішень за умов невизначеності, коли цілі та обмеження встановлюються нечітким набором. Прийняття рішень – це вибір альтернативи, яка одночасно відповідає як невизначеним цілям, так і невизначеним обмеженням. У цьому сенсі цілі та обмеження є симетричними до рішення, що стирає різницю між ними та дає змогу представити рішення як злиття нечітких цілей та обмежень.

У процесі прийняття рішень за методикою Беллмана-Заде не акцентується різниця між метою і обмеженнями. Будь-який поділ на цілі та обмеження є умовним. Можна змінити місця заборонених воріт, але суть рішення залишиться незмінним. В класичній доктрині прийняття рішень така заміна обмежень функцією переваг неприпустима. Однак тут також є деякі приховані подібності між цілями та обмеженнями. Це стає очевидним при застосуванні методу невизначених множників Лагранжа та штрафних функцій, коли мета та обмеження об’єднані в єдиний функціонал.

Розгляд критеріїв оцінювання платоспроможності фізичних осіб. На сьогоднішній день проблема ефективного повернення кредитів, наданих фізичним особам, стоїть на порядку денному більшості фінансових інституцій. Вирішальну

роль у цьому процесі відіграє глибина і точність аналізу платоспроможності потенційних клієнтів. Цей термін у фінансовому контексті передбачає здібність та бажання суб'єкта дотримуватися умов погашення кредиту. Основним інструментом оцінки є аналіз кредитної історії особи, що включає в себе дані про отримані раніше кредити та способи їх виплат. У кредитній системі великої кількості банків оцінка кредитної історії проводиться експертом, який покладається переважно на свій досвід та інтуїцію, що може призвести до внесення суб’єктивних міркувань, які не мають достатньої основи.

Зважаючи на суб'єктивність та інтуїтивний характер експертних оцінок, важливо зменшити вплив особистого досвіду експертів, зосередившись на більш об'єктивних показниках. Це можливо завдяки стандартизації процесу прогнозування поведінки позичальників та ухвалення рішень про кредитування. Відтак, актуальним є розробка уніфікованого методу оцінки платоспроможності індивідів. Тут можуть бути використані кредитні історії, аналіз яких часто має суб'єктивний характер, що знижує точність оцінок. Застосування теорії нечітких множин Заде дозволяє кількісно оцінити вербальні характеристики..

Математичний аналіз діяльності експертів можна представити через механізми категоризації кредитних історій, які зазвичай використовуються у банківській практиці. Проблему можна розв'язати за допомогою алгоритму виведення нечіткої логіки Мамдані, що добре адаптований для роботи з системами, заснованими на людських поглядах та припущеннях.

Подальше використання даних про "надійні" і "ризиковані" кредити, а також інформації про клієнтів (результати опитувань, довідки про доходи, право власності тощо) дозволить сформувати відношення між якістю кредитної історії та характеристиками позичальника. Нечітка модель на етапі розгляду кредитної заяви допоможе визначити ймовірний рівень фінансової дисципліни потенційного клієнта.

Оцінка інвестиційних проектів також зустрічається з труднощами через недостатність, неоднозначність або невизначеність вхідних даних. Рішення цієї проблеми можна знайти через застосування різноманітних методів, в тому числі

статистичних або мінімаксних, але часто вони не є повністю ефективними. У цьому контексті, використання концепцій на основі теорії нечітких множин виявляється найбільш придатним. Вони забезпечують більшу гнучкість при оцінці інвестиційних показників, таких як чистий поточний дохід (NPV) і внутрішня норма прибутку (IRR), які поєднують в собі невизначеність даних та умов реалізації проекту.

Один із прикладів, що демонструє ефективність застосування «нечіткого підходу», є завдання оцінки ключових параметрів інвестиційного плану, такихпоказників як чиста поточна вартість (NPV) та внутрішня норма доходності (IRR). Вони поєднують неоднозначність і невизначеність даних і неоднозначність і неоднозначність середовища проекту.

Застосування нечітких множин у бізнесі. Наразі починають з’являтися інформаційні ресурси, які збирають інформацію про діяльність сотень українських корпорацій. Вся ця накопичена кількість даних ніким серйозно не вивчалася, і не було уявлення про те, як все це різноманіття кількісної та якісної інформації можна проаналізувати в одному ключі. Сьогодні остаточно сформувався такий підхід до аналізу економічних даних – підхід Fuzzy Economics. Йдеться лише про створення системи інтелектуального аналізу на основі розробленої методології, проведення аналітичної обробки даних і розробки оптимальних економічних рішень.

Однією з таких економічних проблем, яку можна ефективно вирішити за допомогою нечітких методів, є якісний аналіз, заснований на агрегуванні ієрархії факторів. Нехай певні властивості економічного об’єкта (фінансова стійкість підприємства, інвестиційна привабливість застави, рівень менеджменту керуючої компанії, ринкова привабливість ділянки забудови тощо) представимо у вигляді дерева ієрархії факторів і:

- в межах ієрархії визначаються системи співвідношення переваги одних підвластивостей над іншими для одного рівня ієрархії;
- підвластивості, які є нижчими ланками ієрархії, можна виміряти як кількісно, так і якісно (в тому числі вербально).
– У цьому випадку комплексна оцінка міцності основної властивості

можлива, якщо:

- проводити всі вимірювання якісно, роблячи нечітку класифікацію кількісних факторів;
- Для моделювання системи краще використовувати системи зважування Saati або Fishburn;
- створити набір якісних рівнів факторів у двовимірній згортці, де факторні ваги є однією з вагових систем, а вузлові точки класифікатора є другою ваговою системою.

Отримавши рівень міцності комплексної властивості, нормований на деяке стандартне середовище (наприклад, 01-інтервал), можна розпізнати якісний рівень цієї комплексної властивості на основі відповідного нечіткого класифікатора. Крім того, на основі отриманої оцінки якості можна провести відповідний аналіз якості товару та деяких його додаткових властивостей (наприклад, максимальна якість досягається за фіксованої ціни або, навпаки, мінімальна ціна за постійний рівень якості Вибрані об'єкти утворюють множину Еджворта - Парето.

2.2 Аналіз структури та засобів формування системи підтримки прийняття рішень

Розглянемо окремі моделі прийняття рішень в умовах невизначеності, які допускають використання в системі підтримки прийняття рішень.

Інформаційно-аналітичні технології в сфері управління – це сукупність методів перегляду та аналізу інформації про процеси дослідження, конкретних процесів діагностики, синтезу та аналізу, а також оцінки наслідків впливу прийняття різних варіантів рішень.

Оптимальний підхід до вирішення задач передбачає декілька етапів: спочатку ідентифікація та аналіз проблеми, далі - встановлення обмежень та визначення критеріїв для вибору рішень. Потім слід провести пошук різноманітних варіантів, їх ретельну оцінку та вибір оптимального з них. Такий підхід є корисним при виборі між кількома варіантами, оцінюючи їх згідно з встановленими

критеріями. Наведемо приклад управлінського рішення щодо вибору технічних засобів для реалізації завдань, що стоять перед підприємством.

Підприємства-учасники тендерів пропонують чотири варіанти забезпечення з дотриманням прописаних в тендері технічних умов Оцінка основних виробничих характеристик перерахованих позицій наведена в табл. 2.1.

Таблиця 2.1 - Вихідні дані для прийняття управлінського рішення

Параметри тендеру			Оцінка пропозицій				Вагові коефіцієнти, %
Критерій	Сутність критерію	Одиниця виміру	I	II	III	IV	
A	Безпечність	бали	9	4	4	3	10
B	Ресурсомісткість	бали	0,8	0,6	0,2	0,8	10
C	Технічні можливості	бали	2	3	6	3	20
D	Можливість модифікації	бали	7	4	9	2	10
E	Ціна	ум.гр.од.	50	35	60	60	35
F	Кількість рекламацій	шт./рік	6	4	7	6	15

Змістом управлінського рішення щодо вибору найбільш вигідної пропозиції є аналіз цих параметрів шляхом багатокритеріальної оцінки, тобто визначення такої пропозиції, за якої всі критерії досягали б оптимального, мінімального або максимального значення, як можливо за даних умов. Основними методами багатокритеріальної оптимізації є:

- Метод справедливого компромісу.
- Метод рівномірної оптимальності.
- Метод згортання критеріїв.
- Метод головного критерію.
- Метод ідеальної точки (за принцип Севіджа).

Кожен із цих підходів має унікальний порядок виконання розрахунків, різний рівень достовірності в оцінках (на результати одного можна покладатися більше або менше, залежно від ризиків та інших чинників), та свої специфічні сильні та слабкі сторони.

Результати розрахунків за вищевказаними критеріями представлені в таблиці 2.2.

Таблиця 2.2 - Результати пошуку оптимальної пропозиції за різними методами багатокритеріальної оптимізації

Асортиментні позиції	Метод				
	рівномірної оптимальності	справедливого компромісу	згортання критеріїв	головного критерію	ідеал. точки
I	-0,6	30,4	-0,006	-0,6	1,0
II	0,1	60,3	0,007	0,0	0,8
III	0,2	39,6	0,004	-1,0	1,0
IV	-2,4	12,0	-0,010	-1,0	1,0
Оптимальне значення	0,2	60,3	0,007	0,0	0,8
Оптимальний варіант	III	II	II	II	II

Як бачимо, за результатами розрахунків чотирьох із п’яти методів останній виявився найбільш оптимальною пропозицією.

Якщо повернутися до характеристик цієї пропозиції, то вони в цілому відповідають умові логіки вибору, тому що її ціна мінімальна (35 ум. гр. од.), а безпека хоч і не на найвищому, але на високому рівні (4 бали). Ресурсомісткість середня (0,6 бала), кількість скарг мінімальна (4 шт./рік). Крім того, можливість модифікації під завдання підприємства має немінімальний рівень (4 бали). Тому природним управлінським рішенням буде зосередити ресурси на підготовці до

реалізації такого рішення.

Модель обмеженої раціональності подібна до раціональної моделі, але вона не спирається на аналіз усіх альтернатив за всіма критеріями, а зупиняється на виборі, заснованому на першому найкращому критерії, і не порівнює альтернативи далі. Досить типовою для виборчих процесів є модель обмеженої раціональності, коли виборець не аналізує повністю всі передвиборчі обіцянки, попередню діяльність кандидата, його досягнення та інші чинники, а обирає один із критеріїв відбору, наприклад, відсутність досвіду роботи у владі. споруди як фактор «чистоти»». Так приймається управлінське рішення, коли акцент робиться не на особистих якостях і досягненнях виконавців, а на тому, яку роль вони повинні відіграти в спецоперації.

Модель, яка ґрунтується на ірраціональності, виходить з припущення, що вибір рішень відбувається до детального аналізу можливих варіантів. Така модель використовується досить часто для:

– вирішення принципово нових, незвичайні рішення, які важко розв’язати;

– вирішення проблем в умовах дефіциту часу;

– ситуації, коли керівник або група керівників мають достатні повноваження, щоб нав'язати своє рішення.

На багатьох підприємствах нераціональні рішення неприпустимі, оскільки від функціонування системи залежить ризик банкрутства чи інших негативних наслідків.

В умовах загрози та невизначеності для прийняття управлінських рішень необхідні відповідні алгоритми. Під час створення цих методик, особливу увагу слід приділяти розробці алгоритмів, які забезпечують виконання окремих задач або вирішення певних питань у у контексті функціонування СПД, а саме:

– Створення проектної групи, основним завданням якої буде визначення цілей, масштабів та етапів проекту та визначення кінцевих користувачів.

– Визначення цілей моніторингу подій та початкової зони реалізації

проекту.

– Визначте початкові сценарії використання, охоплені системою швидкого реагування.

– Визначити вимоги до збору даних, зберігання даних, звітності та моніторингу подій.

– Оцініть, скільки та які джерела даних знадобляться для вибраних сценаріїв (щодо кількості подій за секунду, внутрішньої пам’яті чи потужності обробки), а потім перевірте, чи вдається отримати доступ до цих джерел даних.

Після цього зібрані дані застосовуються для:

– Екологічна та ресурсна оцінка.

– Оцінювання необхідних параметрів архітектурного підходу та технік та методів збору даних, щоб відповісти на запитання: які зусилля знадобляться для інтеграції джерел даних і чи зможуть ці джерела підтримувати безперебійне функціонування системи журналювання без зниження її продуктивності.

– Визначення потенціалу джерел даних генерувати передбачувані події. Деякі джерела можуть бути обмежені через їх низьку продуктивність тощо.

– Розробка процесу спостереження за подіями та адекватного реагування на несподівані інциденти. Варто звернути особливу увагу на конкретизацію сценаріїв реагування на різноманітні інциденти, для цього також доцільно проводити симуляційні та навчальні заходи для розвитку навичок реагування на інциденти.

– Встановлення відповідних процесів і політики для визначення потрібних ресурсів для системи оперативного реагування. Після збору інформації підбирається необхідне технічне обладнання та програмне забезпечення.

Розглянемо загальну концепцію дій та алгоритм прийняття управлінських рішень у нашій системі прийняття рішень. Для створення системи прийняття та реалізації рішень необхідно спочатку визначити рівень завдань і відповідних заходів в умовах необхідності їх виконання. Нижче наведено алгоритм, за яким ми побудуємо систему реагування на конкретні події в умовах невизначеності (рис.

2.1.).

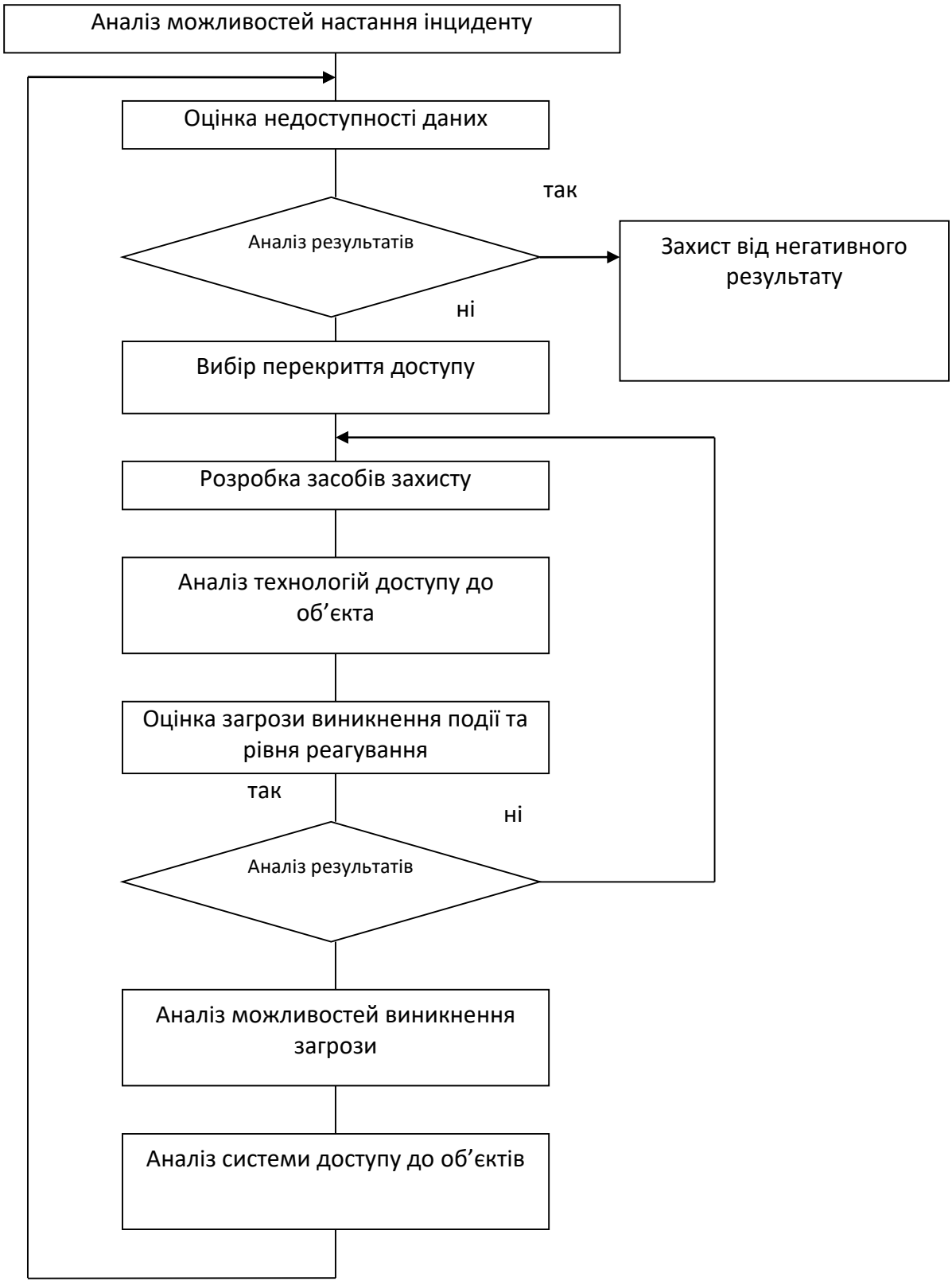

Рисунок 2.1 - Алгоритм прийняття управлінських рішень задля забезпечення поставлених задач

На рис. 2.2 зображена графічна схема виявлення SQL-ін’єкцій у веб-застосунках.

В рамках виконання роботи буде спроектовано програмний продукт, який виділено червоним кольором, що містить [23]:

– Секція збору та анілізу інцидентів;
– Секція візуалізації даних про проникнення;
– Секція для архівування даних в базу даних.

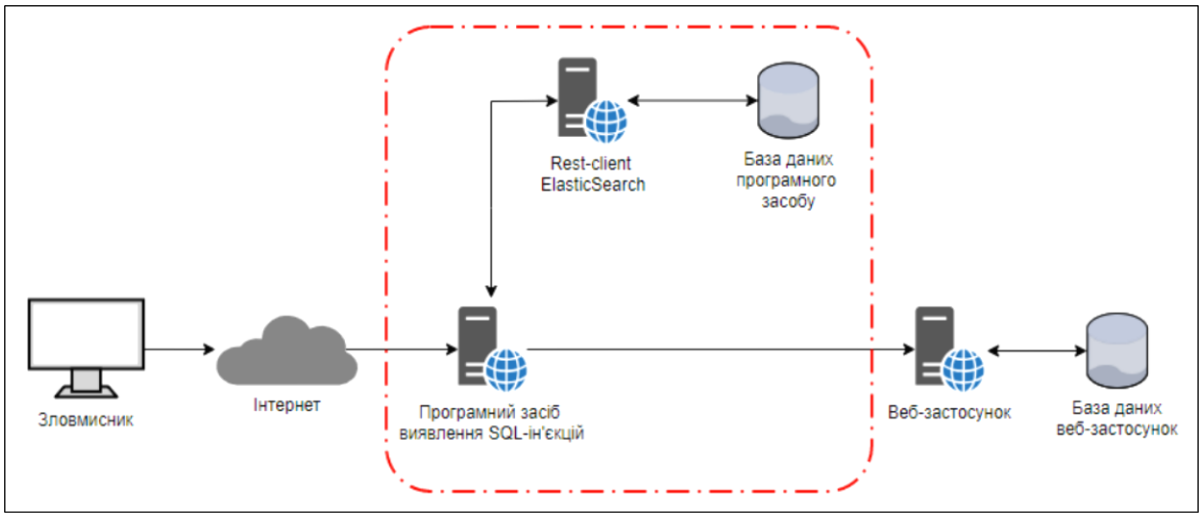

Рисунок 2.2 - Графічна схема програмного забезпечення для виявлення вразливостей у веб-застосунках

Загальні функціональні вимоги, що ставляться до клієнтської та серверної частин системи, представлено у табл. 2.3 та 2.4.

Для дослідження коректності вимог загально для системи потрібно перевірити чи не конфліктують вони між собою та чи не суперечать одна одній. Матриці залежності цих вимог до програмного забезпечення з ідетифікації вразливостей у роботі веб-застосунку наведені в табл. 2.5 та 2.6. В наведених таблицях позначення X в комірці означає, що протиріч між відповідними вимогами не має бути. Наприклад, В1 «Можливість виявляти наявність SQL-ін’єкції у веб-застосунку» не має заважати В5 «Безперервна робота програмного засобу».

Таблиця 2.3 - Функціональні вимоги серверної частини

Кодовий номер	Вимога
1	2
В1	Здатність ідентифікувати SQL-ін’єкції у веб-застосунку
В2	Реєстрація даних про SQL-ін’єкцію у файл журналу
В3	Сповіщення про SQL-ін’єкцію адміністратору веб-додатку
В4	Занесення інформації про SQL-ін’єкцію в базу даних
В5	Неперервна робота програмного рішення
В6	Здатність виявлети SQL-ін’єкцій

Таблиця 2.4 - Функціональні вимоги клієнтської частини

Кодовий номер	Вимога
В7	Здатність перевірки поточного стану системи
В8	Здатність перегляду конфігурації системи
В9	Здатність експорту даних про стану системи в форматі Excel
В10	Здатність авторизації у системі
В11	Здатність змінювати конфігурації програмного забезпечення для ідентифікації SQL-ін’єкції у веб-додатку

Таблиця 2.5 - Матриця взаємозв'язків між вимогами функціональної складової серверної частини програмного забезпечення

Код вимоги	В1	В2	В3	В4	В5	В6
В1	X	X	X	X	X	X
В2		X	X	X	X	X
В3			X	X	X	X
В4				X	X	X
В5					X	X
В6						X

Таблиця 2.6 - Матриця взаємозв'язків між вимогами функціональної складової клієнтської частини програмного забезпечення

Код вимоги	В7	В8	В9	В10	В11
В7	X	X	X	X	X
В8		X	X	X	X
В9			X	X	X
В10				X	X
В11					X

Розроблені нами матриці залежності, які наведені у таблицях 2.5 та 2.6 жодних протиріч вимог або накладань вимог не виявили.

2.3 Розробка алгоритмів прийняття рішень на основі чек-листів

В контексті моделювання захисту від ін'єкційних атак, ключове значення має симуляція цих атак, що дозволяє отримати необхідні дані для нашої моделі. Таким чином, для оцінювання рівня захищеності системи від потенційних ін’єкційних атак вибрано три основні параметри: частота появи ін’єкційної атаки протягом визначеного часового періоду, що визначається через симуляційну модель, швидкість реагування на неавторизовану ін’єкцію в запиті, і ступінь значущості інформаційного ресурсу для системи. Всі ці показники безпосередньо впливають на ймовірність ефективного захисту системи від ін’єкційних атак, що ми оцінюємо як кінцевий результат. Відповідно, потрібно досліджувати взаємозв'язок між наступними параметрами:

Незалежні змінні:

X1 – вірогідність ін’єкційної атаки в HTTP-запиті, у відсотках

X2 – час необхідний для відновлення роботи вузла після відкидання запиту, у секундах

X3 – рівень важливості працездатності вузла, на який здійснювалась атака, у відсотках

Залежна змінна:

Y – ймовірність захищеності вузла від ін’єкційної атаки, у відсотках.

Для побудови моделі захисту та оцінки її параметрів ми використаємо табличний процесор Excel.

За результатами імітаційного моделювання були отримані результати, які наведені у таблиці 2.7.

В результаті розраховані коефіцієнти кореляції між показниками ймовірності виникнення атаки та наслідків її впливу і показником якості системи захисту від атак (табл. 2.8).

Оцінка отриманих даних показала (див. табл. 2.8), що показник значення коефіцієнта кореляції Пірсона становить 0,52 між частотою виникнення атак і рівнем системи захисту, це вказує на відносно слабкий кореляційний зв'язок цих параметрів, тобто система захисту повинна спрацьовувати при кожній атаці, незалежно від її ймовірності.

Таблиця 2.7 – Вхідні умови дослідження рівня захисту від ін’єкційних атак

номер модельованої атаки	Значення рівня захисту			Ступінь рівня захисту від ін’єкційних атак
	X1	X2	X3	
1	28	50	56	52
2	36	52	54	56
3	30	54	40	50
4	39	58	56	64
5	38	54	44	56

Таблиця 2.8 – Результати обчислення коефіцієнтів кореляції Пірсона rxy між факторними ознаками x та y

Зв'язок факторних ознак	Коефіцієнт кореляції Пірсона rxy
x1-y: ймовірність ін’єкційної атаки – захист від атаки	0,52
x2-y: час відновлення роботи після ін’єкції – захист від атаки	0,74
x3-y: рівень важливості вузла – захист від атаки	0,83

При цьому рівень важливості виявляє найбільший коефіцієнт кореляції (див. табл. 2.8), бо у випадку, якщо атака відбувалася на менш важливий технічно вузол, то її наслідки виявлялися незначними, наприклад, знижувалася швидкість обміну даними між внутрішніми ресурсами, однак у випадку, якщо атака відбувалася на важливий вузол, від якого залежить діяльність всього інформаційного ресурсу, наслідки атаки будуть критичними і настільки рівень захисту має бути вищий..

Термін відновлення роботи ресурсу також має важливе значення, адже стороння ін’єкційна атака може завдати негативних наслідків та мати на меті й уповільнення роботи сервера із запитами, отже швидкість реагування на атаку та вчасне її знешкодження мають важливого значення.

Виявлення кореляційних зв’язків між кількох ознак (комплексним впливом показників виникнення ін’єкційних атак на результати успішності системи захисту), представлені у таблиці 2.9, показали тісноту зв’язків між ними.

Таблиця 2.9 – Розрахунки сукупних коефіцієнтів множинної кореляції між факторними характеристиками x1, x2, x3 та y

Факторна характеристика	Коефіцієнт множинної кореляції
x1-x2-y: ймовірність атаки та час відновлення після неї – рівень захисту	0,89
x1-x3-y: ймовірність атаки на важливий вузол – рівень захисту	0,93
x1-x2-x3-y: якість реагування на атаки – рівень захисту	0,97

Із наведених у таблиці 2.9 даних можна побачити, що коефіцієнт множинної кореляції рівня системи захисту із показниками факторних ознак реагування на атаки (до уваги береться саме одночасний вплив комплексу усіх перерахованих факторів) дещо зростає.

Повернемося до багатофакторної регресійної моделі, наведеної на початку, та підставимо до формули показники факторних ознак та обчислені коефіцієнти

кореляції, в результаті ми встановлюємо, що якість системи захисту від сторонніх ін’єкційних атак усіх видів безпосередньо впливає на кінцевий результат і дорівнює 0,92, що відповідає 92% успіхів відбиття атак і складає якість системи вчасного реагування та виявлення шкідливих ін’єкції.

Тож, враховуючи значення коефіцієнтів кореляції Пірсона, можемо зробити висновок про високу залежність окреслених понять. Значить, ігнорування вибраних параметрів у цій залежності є неприпустимим.

Таким чином, ми здатні створити багатофакторну модель для оцінювання ефективності системи захисту від ін’єкційних атак на базі трьох незалежних параметрів.

Множинна регресія являє собою метод статичного аналізу зв’язку з одного боку залежної змінної Y, з іншого – множиними змінними (X1, X2,…Xn), це метод служить для визначення пріоритетності незалежних змінних. Він забезпечує простий варіант структурної ідентифікації, вибираючи з набору можливих регресивних зв'язків єдиний для прогнозування результатів.

Для створення багатофакторної регресійної моделі необхідно:

– Визначити змінні моделі

Загальний вигляд математичної моделі:

$$Y = f(X_1, X_2, X_3) \tag{2.1}$$

де Y – значення захищеності (залежна змінна), X_1 – ймовірність виникнення ін’єкційної атаки у певний період часу (незалежна змінна), X_2 – час відновлення роботи після атаки (незалежна пояснювальна змінна), X_3 – відображає значимість ресурсу, на який відбувалася ін’єкція (незалежна змінна)

– Окреслення параметрів модель

Окреслення параметрів моделі полягає в створенні аналітичної форми математичної багатофакторної моделі на базі досліджуваних параметрів. Вона складається з визначеного виду функції чи функцій, що застосовуються для

побудови моделей, має ймовірнісні параметри, які притаманні стохастичним залишкам математичної моделі.

Ми будуємо лінійну модель залежності значення Y:

$$Y = a_0 + a_1X_1 + a_2X_2 + a_3X_3 \quad (2.2)$$

Оцінка параметрів моделі здійснюється із використанням:

– методу найменших квадратів;

– стандартну функцію “Лінейн”

Оцінювання залежності параметрів моделі має вигляд:

$$A = (X'X)^{-1}X'Y \quad (2.3)$$

Звідси

$a_0 = 0.967$

$a_1 = 0.640$

$a_2 = 0.343$

$a_3 = 0.287$

$\hat{Y} = 0.967 + 0.640X_1 + 0.343X_2 + 0.287X_3$ – вигляд утвореної моделі.

Можемо дійти висновку, що при однакових умовах у випадку якщо незалежна змінна X_1 (ймовірність виникнення атаки) збільшується на один відсоток, при цьому залежна змінна Y (рівень системи захисту від атаки) також повинна збільшитись на 0,64%. Відповідно, за таких однакових умов, якщо незалежна змінна X_2 (час відновлення системи після атаки) збільшується на 1 с, то успішність рівня системи захисту також має збільшуватися на 0,343%. Якщо X_3 (важливість вузла/ресурсу) збільшується на один відсоток, то рівень системи його захисту відповідно збільшується на 0,287%.

Кореляційна матриця будується між результуючою та факторними ознаками, тобто між рівнем захисту Y та чинниками X_1 – ймовірність виникнення ін’єкційної

атаки в HTTP-запиті, %, X_2 – час відновлення роботи після відхилення запиту, с, X_3 – рівень важливості працездатності вузла, на який здійснювалась атака, %. Вигляд кореляційної матриці наведений у таблиці 2.4

Таблиця 2.10 – Кореляційна матриця

Показник	Y	X_1	X_2	X_3
Y	1			
X_1	0,837121	1		
X_2	0,741215	0,692106	1	
X_3	0,522913	0,122239	-0,09009	1

Отже, найбільш значимий зв’язок виявлено між Y та X_1 і X_2. В той час зв'язок з X_3 менш вагомий.

Таким чином, було сформовано багатофакторну модель залежності необхідного рівня системи захисту від сторонніх атак із конкретними характеристиками кожного ресурсу, на який існує потенційна загроза здійснення атаки.

Встановлюємо коваріаційну матрицю, оцінюємо стандартні похибки та надаємо інтервальну оцінку параметрам моделі.

Знаходимо $\hat{Y}$ за формулою (2.4)

$$\hat{Y} = 0.967 + 0.640X_1 + 0.343X_2 + 0.287X_3 \quad (2.4)$$

Знаходимо залишки $u = Y - \hat{Y}$ та квадрати залишків $\hat{u}^2$

Залишки u

$$\sum_{i=l}^{n} (y_i - \hat{y})^2 = úu \quad (2.5)$$

$$n - m \tag{2.6}$$

$$\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{n - m} = \frac{u\acute{u}u}{n - m} = o_u^2 \tag{2.7}$$

Визначаємо дисперсію похибки враховуючи числа ступенів свободи

$$Du = sum\frac{\hat{u}^2}{(n - m)} \tag{2.8}$$

Дисперсія без урахування ступенів свободи

$$Du1 = sum\frac{\hat{u}^2}{n} \tag{2.9}$$

Стандартне відхилення похибки від дисперсії з урахуванням ступенів свободи Su – корінь квадратний з Du

$Su = 2.5435$, $Du = 6.4691$, $Du1 = 1.2938$

Будуємо коваріаційну матрицю, наведену в таблиці 2.11.

Таблиця 2.11 – Коваріаційна матриця

Показник	a^0	a^1	a^2	a^3
a^0	23,04	17,68	9,44	16,8
a^1	17,68	19,36	8,08	16,8
a^2	9,44	8,08	7,04	-1,6
a^3	16,8	3,6	-1,6	44,8

a0, a1, a2, a3 – оцінки параметрів моделі.

У головній діагоналі матриці коваріації знаходяться дисперсії оцінок параметрів моделі.

Знайдемо стандартну похибку Sa^j:

a0=1,80

a1=0,40

a2=0,195

a3=0,097

a0 – нестійка оцінка параметрів, тобто статистично незначуща.

Оцінимо достовірність моделей, використовуючи наступні значення:

– коефіцієнти детермінації і кореляції;

– критерій Фішера (F- критерій);

– критерій Стьюдента (t- критерій).

Будуємо статистичний аналіз у Excel та отримуємо оцінку даних:

$$M(u) = 0, \tag{2.10}$$

де $R^2 = \left(Dy - \frac{Du}{Dy}\right)$ - коефіцієнт детермінації.

Наступним кроком знайдемо дисперсії по y:

$$Dy = \frac{sum(yi - yser)^2}{n} - 1 = 6.063 \tag{2.11}$$

де Dy – дисперсія з урахуванням ступенів свободи.

$$Dy1 = \frac{sum(yi - yser)^2}{n} = 5.760 \tag{2.12}$$

де $Dy1$ – дисперсія без урахування ступенів свободи.

$$Dregr = \frac{sum\left(y^i - yser\right)^2}{m} - 1 = 28.8 \tag{2.13}$$

де $Dregr$ – дисперсія регресії для F-критерію.

$R^2 = 0.95$ – коефіцієнт детермінації

Це вказує на високий рівень достовірності. Розрахунок коефіцієнту детермінації з урахуванням ступенів свободи показує, що на 95% варіація y визначається варіацією x. А лише 5 %- інші фактори.

Таким чином, залежність між x та y дуже висока.

F-критерій визначається на основі оцінки статистичної важливості математичної моделі в цілому.

$$F = \frac{\delta_{x^2}}{\delta_{y^2}} \tag{2.14}$$

Також висуваємо 2 гіпотези: $H0$ – де модель статистично недостовірна та $H1$ – де модель статистично достовірна

F факт. $> F$ табл. $-$гіпотеза $H1$

F факт. $< F$ табл. $-$гіпотеза $H0$

F факт. $= \frac{Dregr}{Du} = 4.75$

Ступінь свободи $1 = m - 1 = 3$, Ступінь свободи $2 = n - m = 16$

F табл. $= 3.2388$

Отже, F табл. менше, ніж F факт., що означає, що математична модель статистично достовірна в цілому.

Отже, ми приймаємо гіпотезу $H1$.

Таким чином, розроблена нами багатофакторна модель може використовуватись для аналізу рівня захищеності від потенційних вразливостей.

2.4 Висновки до розділу 2

Таким чином, ми розробили модель оцінки потреби у рівні захисту для кожного вузла/ресурса в залежності від трьох параметрів – ймовірності атаки на

даний вузол, часу відновлення нормальної роботи вузла після атаки і встановлення ступеню важливості окресленого ресурсу для функціонування системи загалом, на яку здійснюються атаки. Розроблена модель є лінійною і підтверджує потребу у системі захисту від несанкціонованих атак для кожного об’єкта системи, а налаштовуваний параметр виявлення появи сторонньої ін’єкції змінюється відповідно до кожного окремого об’єкту залежно від наслідків шкідливого впливу, до яких здатна призвести атака на цей об’єкт. Розроблена модель дозволяє вчасно реагувати на можливі атаки і сприяти зменшенню навантаження на систему захисту загалом, розмежовуючи рівень необхідності у захисті окремо для кожного компонента системи. Практична реалізації розробленої моделі захисту від сторонніх ін’єкційних атак може бути здійснена як у статичному форматі (налаштування здійснюють за результатами разової оцінки), так і у динамічно-автоматичному форматі, залежно від кількості об’єктів, які потребують захисту, а також від рівню загроз для кожного з цих об’єктів.

3 МЕТОДИКА ПІДТРИМКИ ПЕРЕВІРКИ ВЕБ-ДОДАТКІВ НА ВРАЗЛИВОСТІ

3.1 Опис алгоритму синтезу чек-ліста для перевірки на вразливості

Чекліст – це список, який включає ряд важливих перевірок під час тестування програмного забезпечення системи. Відмічаючи пункти списку, тестувальник може дізнатися про стан процесу виконаної роботи та якість програмного продукту. Працюючи за чеклістом, підвищується відповідальність та виключається ймовірність необхідності повторної перевірки продукту за тими ж кейсами, а також зростає якість тестування, тому що ймовірність пропустити якийсь функціонал значно знижується. Отже, вкрай важливо знати позиції чекліста та вміти ефективно ним користуватися. Чеклісти як правило створюють у Google-таблицях для надання загального доступу необхідним QA-фахівцям.

Пункти чекліста можуть формуватися як лінійна структура, так і деревоподібна, з розділами/підрозділами або без них. Його пункти мають бути короткими та зрозумілими тестувальникам, які зазвичай ще не знайомі з веб-додатком. Пункти формують однозначними, щоб не було інших трактувань. Всі пункти оформлюють однією мовою: українською або англійською.

Як правило, у складі кожного чекліста має бути кілька стовпців, кожен з яких призначений для тестування окремої платформи. Завжди треба вказувати назву пристрою, браузера та його версію (рис. 3.1).

Google Chrome	Mozilla Firefox	Opera	Safari
91.0.4472.80	89.0.1	77.0.4054.172	14.1

Рисунок 3.1 – Схема заповнення інформації про браузер та його версію

Здійснювати тестування сайту або додатку можуть декілька тестувальників одночасно. При цьому кожен з них зосереджується на одній чи двох платформах на

якій і веде роботу. У чеклисті для кожної з платформ вказують особу яка відповідальна за визначений сегмент тестування.

Після завершення перевірок, тестувальник вказує статус на впроти кожного з протестовано пункту. Варіанти статусів можливі такі:

«Passed» – задача виконана без виявлення помилок;

«Failed» – виявлено один чи декілька багів;

«Blocked» – тестування неможливе через блокування однією з помилок;

«In Progress» – завдання в процесі виконання;

«Not run» – пункт ще не пройшов перевірку;

«Skipped» – пункт вирішено не тестувати з певних причин, наприклад, через відсутність реалізованого функціоналу.

Для більшої наочності зазвичай кожному статусу присвоюють відповідний колір.

3.2 Методи оцінки повноти чек-ліста

Для ефективного проектування варіантів – необхідно окреслити дійових осіб – акторів. Зазвичай програма виявлення вторгнень надає можливість користувачеві виявляти та відслідковувати SQL-ін’єкції у веб-застосунку, при цьому єдиним користувачем системи має бути адміністратор (див. рис. 3.1) [8, 28].

Детальний опис варіантів використання представлено у табл. 3.1 – 3.8.

Таблиця 3.1 - Варіант використання «Авторизація на сторінці перевірки стану веб-додатку»

Найменування	Авторизація на сторінці перевірки станувеб-додатку
1	2
Права доступу	Адміністратор
Інші права	Немає

Кінець таблиці 3.1

1	2
Опис	Можливість увійти у систему для перегляду поточного стану веб-додатку
Попередні умови	Відкрита веб-сторінка перегляду стану веб-додатку
Вихідні умови	Відкрита веб-сторінка перегляду поточного стану веб-додатку

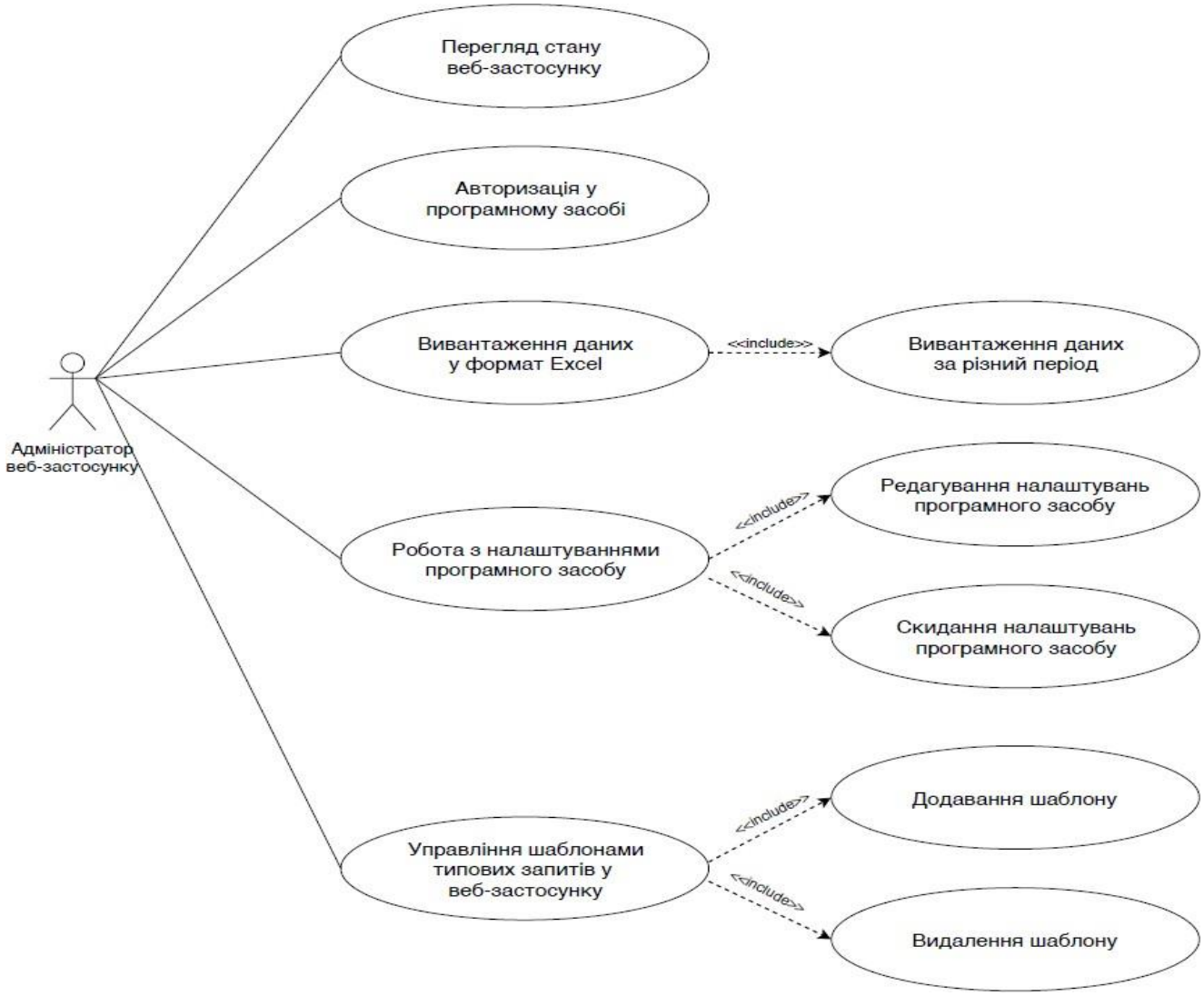

Рисунок 3.1 - Структурна схема варіантів використання програмного забезпечення для виявлення вразливостей

Таблиця 3.2 - Варіант використання «Створення налаштувань програмного засобу виявлення вразливостей»

Найменування	Створення налаштувань програмного засобу виявлення вразливостей
Права доступу	Адміністратор
Опис	Здатність змінювати конфігурацію програмного забезпеченнявиявлення з виявлення вразливостей
Попередні умови	Відкрита сторінка конфігурації програмного засобу виявлення вразливостей де адміністратор є авторизованим
Вихідні умови	Застосована конфігурація програмного засобуз виявлення вразливостей

Таблиця 3.3 - Варіант використання «Експорт даних про вразливості у Excel»

Найменування	Вивантаження даних про вразливості у форматіExcel
Права доступу	Адміністратор
Опис	Здатність експорту даних про виявлені вразливості у веб-додатку в Excel
Попередні умови	Відкрита сторінка виявленихвразливостей
Вихідні умови	Сформований вивід у Excel форматі

Таблиця 3.4 - Варіант використання «Видалити користувацькі конфігурації програмного засобу з виявлення вразливостей»

Найменування	Видалити користувацькі конфігурації програмного засобу з виявлення вразливостей
1	2
Права доступу	Адміністратор

Кінець таблиці 3.4

1	2
Опис	Можливість видалити конфігурації
Попередні умови	Відкрита сторінка конфігурації з виявлення вразливостей де адміністратор є авторизований
Вихідні умови	Видалено користувацькі конфігурації системи виявлення вразливостей

Таблиця 3.5 - Варіант використання «Редагувати користувацькі конфігурації системи виявлення вторгнень»

Найменування	Редагувати користувацькі конфігурації системи
Права доступу	Користувач
Опис	Можливість здійснювати редагування конфігурації системи
Попередні умови	Відкрита сторінка конфігурації системи
Вихідні умови	Відредаговані конфігурації системи виявлення вторгнень

Таблиця 3.6 - Варіант використання «Перегляд стану веб-застосунку»

Найменування	Перегляд стану веб-застосунку
1	2
Права доступу	Адміністратор
Опис	Можливість переглядати поточний стану веб-додатку
Попередні умови	Відкрита сторінка перегляду стану веб-додатку де адміністратор є авторизованим
Вихідні умови	Перегляд стану веб-додатку

3.3 Аналіз обчислювальної складності методів

Для визначення ефективності інформаційно-аналітичної системи слід провести функціонально-вартісний аналіз (ФВА). Для аналізу системи розглянемо спочатку функції, які на неї покладаються, і виходячи з цього, спроектуємо структуру системи.

Основні функції, які повинен реалізовувати програмний продукт:

– Перевірка прав доступу до веб-застосунку.

– Надання доступу для передачі або прийому інформації в системі.

– Забезпечення отримання інформаційної та розрахункової інформації щодо вразливостей застосунку з системи, згідно з пріоритетом та правом доступу користувача.

Головна функція F0 реалізує ціль створення ПП.

Враховуючи специфічні цілі, можна виокремити такі основні завдання ПП:

F1 – перевірка прав доступу при спробі входу до бази даних (БД);

F2 – надання доступу при введенні пароля для перегляду чи редагування БД;

F3 – забезпечення надання запрошеної інформації.

Варіанти реалізації основних функцій записуються в морфологічній карті системи. На основі цієї карти побудовано позитивно-негативну матрицю (таблиця 3.7).

Таблиця 3.7 - Позитивно-негативна матриця

Основні функції	Варіанти реалізації функцій	Переваги	Недоліки
1	2	3	4
F_1	а)	Доступ до БД всіх користувачів	Можливість зміни інформації (помилок в БД)
	б)	Доступ до БД після введення пароля	Можливість несанкціонованого доступу

Кінець таблиці 3.7

1	2	3	4
F_1	в)	Встановлення рівнів пріоритету	Ускладнення програми
F_2	а)	Перевірка пріоритету пароля	Збільшення об’єму програми
	б)	Перевірка режиму доступу	Збільшення об’єму програми
	в)	Надання доступу відповідно прав	Збільшення об’єму програми
F_3	а)	Надання інформації про вразливості	Обмеженість інформації
	б)	Комплекс інформаційно-розрахункових даних	Складнощі у випадку помилки у алгоритмі
	в)	Проведення розрахунків за запитом	Неповнота інформації

На основі даних про зміст основних функцій, які повинна реалізовувати система, визначимо основні параметри, які впливають на обчислювальну складність.

До головних параметрів, які характеризують ПП, можна віднести об'єм пам'яті, яку займають дані, потреби в об'ємі оперативної пам'яті, потенціальний об'єм програми. Так як ПП необхідний для роботи всієї ради, важливе значення має зміст інформації, що відображається. У зв’язку з необхідністю швидко реагувати на спробу несанкціонованого доступу до локальної мережі, велике значення має такий параметр, як швидкість обробки запитів. Коефіцієнт використання ПП може вказати доцільність створення даного ПП. Для характеристики розробляємого ПП використовуються наступні параметри:

x1 - об'єм пам'яті, яку займають дані; x2 - об'єм пам'яті, яку займає додаткова інформація;x3 - потреби в об'ємі оперативної пам'яті; x4 - потенціальний об'єм програми; x5 - коефіцієнт використання ПП; x6 – швидкість обробки запитів;

Система параметрів, прийнята до розрахунків, достатньо повно характеризує

споживчі властивості ПП.

Результати ранжування параметрів приведені в таблиці 3.8.

Таблиця 3.8 - Результати ранжування параметрів

Назва	Умовні позначення	Ранг параметра за оцінкою експерта							Сума рангів,	Відхилення,	Δ_i^2
параметра		1	2	3	4	5	6	7	R_i	Δ_i	
Об'єм пам'яті, яку займають дані	x_1	1	2	1	1	2	1	1	9	-26	676
Об'єм пам'яті, яку займає додаткова інформація	x_2	5	5	7	5	5	5	5	37	2	4
Потреби в об'ємі оперативної пам'яті	x_3	1	3	1	2	1	1	1	10	-25	625
Потенціальний об'єм програми	X_4	7	8	6	7	7	7	7	49	14	196
Коефіцієнт використання ПП	x_5	8	5	7	6	7	7	8	48	13	169
Швидкість обробки запитів	x_6	3	2	3	4	3	4	3	22	-13	169
Результат		25	25	25	25	25	25	25	175		1839

Визначимо суму рангів кожного показника (по рядках):

$$R_i = \sum_{l=1}^{N} r_{il} \tag{3.1}$$

де r_{il} – ранг i-го параметра, визначений l-м експертом; N – число експертів.

Обчислимо середню суму рангів (T) за формулою

$$T = \frac{1}{n} R_{ij} = \frac{175}{6} = 30 \tag{3.2}$$

Визначимо відхилення суми рангів кожного параметру (R_i) від середньої суми рангів (T):

$$\triangle_i = R_i - T \tag{3.3}$$

Сума відхилень за всіма параметрами дорівнює 0.

Обчислимо квадрат відхилень за кожним параметром ($\triangle_i^2$) та загальну суму квадратів відхилень

$$S = \sum_{i=1}^{n} \triangle_i^2 \tag{3.4}$$

Отримані результати занесемо в таблицю 3.9.

Визначимо коефіцієнт узгодженості (конкордації) за формулою

$$W = \frac{12 \times S}{N^2(n^3 - n)} = \frac{12 \times 1839}{7^2 \times (7^3 - 7)} = \frac{22068}{26464} = 0.833888 \approx 0.834 \tag{3.5}$$

Коефіцієнт узгодженості має значення в інтервалі $0 \leq W \leq 1$.

Порівнюючи розрахункову величину W з нормативною WH (для програмного продукту $WH = 0.67$), бачимо, що $W > WH$.

Використовуючи отримані від кожного експерта результати ранжування параметрів (таблиця 3.8), проводимо попарне порівняння всіх параметрів і результати занесемо в таблицю 3.9.

Будемо використовувати наступні значення коефіцієнтів переваги (a_{ij}):

$$a_{ij} = \begin{cases} 1.5 \text{ при } x_i > x_j \\ 1.0 \text{ при } x_i = x_j \\ 0.5 \text{ при } x_i < x_j \end{cases} \tag{3.6}$$

де x_i і x_j – параметри, які порівнюються між собою.

Таблиця 3.9 - Попарне порівняння параметрів

Параметри	Експерти							Підсумкова оцінка	Числове значення коефіцієнтів переваги (a_{ij})
	1	2	3	4	5	6	7		
1	2	3	4	5	6	7	8	9	10
x_1 і x_2	>	>	>	>	>	>	>	>	1,5
x_1 і x_3	>	>	<	>	<	>	>	>	1,5
x_1 і x_4	>	>	>	>	>	>	>	>	1,5
x_1 і x_5	>	>	>	>	>	>	>	>	1,5
x_1 і x_6	>	>	>	>	>	>	>	>	1,5
x_1 і x_7	>	>	>	>	>	>	>	>	1,5
x_2 і x_3	<	<	<	<	<	<	<	<	0,5
x_2 і x_4	<	<	<	<	<	<	<	<	0,5
x_2 і x_5	>	>	<	>	>	>	>	>	1,5
x_2 і x_6	>	>	>	>	>	>	>	>	1,5
x_2 і x_7	<	<	<	<	<	<	<	<	0,5
x_3 і x_4	>	>	>	>	>	>	>	>	1,5
x_3 і x_5	>	>	>	>	>	>	>	>	1,5
x_3 і x_6	>	>	>	>	>	>	>	>	1,5
x_3 і x_7	>	>	>	>	>	>	>	>	1,5
x_4 і x_5	>	>	>	>	>	>	>	>	1,5
x_4 і x_6	>	>	>	>	>	>	>	>	1,5
x_4 і x_7	<	<	<	>	>	<	<	<	0,5
x_5 і x_6	>	<	<	>	>	<	>	>	1,5
x_5 і x_7	<	<	<	<	<	<	<	<	0,5
x_5 і x_7	<	<	>	<	<	<	<	<	0,5
x_6 і x_7	<	<	<	<	<	<	<	<	0,5

На основі числових даних a_{ij} таблиці 3.9 складемо квадратну матрицю $A = \| a_{ij} \|$ (таблиця 3.10).

Розрахунок вагомості (пріоритетності) кожного параметра φ_iпроводимо за наступними формулами:

$$\varphi_i = \frac{b_i}{\sum_{i=1}^{n} b_i} \tag{3.7}$$

$$b_i = \sum_{j=1}^{n} a_{ij} \text{ (6.7)} \tag{3.8}$$

де b_i – вагомість і-го параметра згідно результатів оцінок всіх експертів; він зазначається як сума числових значень коефіцієнтів переваги (a_{ij}) даних усіх експертів по і-му параметру.

Розрахунок відносних оцінок вагомості (φ_i) здійснюємо декілька разів, поки наступне значення не буде відхилятися від попереднього менше ніж на 5%. На другій і наступних варіантах значення коефіцієнта вагомості (φ_i) розраховуємо таким чином:

$$\varphi_i' = \frac{b_i'}{\sum_{i=1}^{n} b_i'} \tag{3.9}$$

де b_i' визначається:

$$b_i' = a_{i1}b_1 + a_{i2}b_2 + \ldots + a_{in}b_n \tag{3.10}$$

В нашому випадку

$b_1' = 1.5 \times 13 \times 8 + 1 \times 13 = 156 + 13 = 169$

$b_2' = 1.5 \times 9 \times 4 + 0.5 \times 9 \times 4 + 1 \times 9 = 54 + 18 + 9 = 81$

$b_3' = 1.5 \times 12 \times 7 + 0.6 \times 12 \times 1 + 1 \times 12 = 126 + 6 + 12 = 144$

$b_4' = 1.5 \times 10 \times 5 + 0.5 \times 10 \times 3 + 1 * 10 = 75 + 15 + 10 = 100$

$b_5' = 1.5 \times 7 \times 2 + 0.5 \times 7 \times 6 + 1 \times 7 = 21 + 21 + 7 = 49$

$b_6' = 1.5 \times 6 \times 1 + 0.5 \times 6 \times 7 + 1 \times 6 = 9 + 21 + 6 = 36$

$b_7' = 1.5 \times 11 \times 6 + 0.5 \times 11 \times 2 + 1 \times 11 = 99 + 11 + 11 = 121$

Результати розрахунків заносимо в таблицю 3.10.

Таблиця 3.10 - Розрахунок вагомості параметрів

X_i	X_j							Перша ітерація		Друга ітерація	
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	B_i	φ_i	b_i'	φ_i'
X_1	1,0	1,5	1,5	1,5	1,5	1,5	1,5	10	0,16	169	0,2
X_2	0,5	1,0	0,5	0,5	1,5	1,5	0,5	6	0,1	81	0,1
X_3	0,5	1,5	1,0	1,5	1,5	1,5	1,5	9	0,15	144	0,18
X_4	0,5	1,5	0,5	1,0	1,5	1,5	0,5	7	0,12	100	0,13
X_5	0,5	0,5	0,5	0,5	1,0	1,5	0,5	5	0,09	49	0,06
X_6	0,5	0,5	0,5	0,5	0,5	1,0	0,5	4	0,07	36	0,06
X_7	0,5	1,5	0,5	1,5	1,5	1,5	1,0	8	0,14	121	0,15
Всього								49	0.84	789	0.88

Отже, відносна оцінка, яку отримано на останній стадії розрахунків, окреслюється як коефіцієнт вагомості (φ_i) і-го параметру.

Порівнюючи варіанти реалізації функцій за їх недоліками та перевагами, а також коефіцієнтів вагомості означених параметрів обираємо такі варіанти реалізації функцій:

1) $F_{1а} + F_{2б} + F_{3б}$

2) $F_{1а} + F_{2б} + F_{3б}$

3) $F_{1а} + F_{2б} + F_{3в}$

4) $F_{1б} + F_{2а} + F_{3а}$

5) $F_{1б} + F_{2а} + F_{3б}$

6) $F_{1б} + F_{2а} + F_{3в}$

7) $F_{1в} + F_{2в} + F_{3а}$

8) $F_{1в} + F_{2в} + F_{3б}$

9) $F_{1в} + F_{2в} + F_{3в}$

Виконаємо обчислення узагальнюючих показників якості ПП за формулою 3.11 для кожного варіанту окремо по кожній функції.

$$К_{трк} = К_{тр}[F_{1k}] + К_{тр}[F_{2k}] + К_{тр}[F_{3k}] \quad (3.11)$$

де $К_{тр}[F_{1k}]$ – показник технічного рівня першої функції К-го варіанту реалізації основних функцій виробу.

Результати розрахунків, проведених по формулі (3.10) зведемо в таблицю 3.11.

Таблиця 3.11 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри, що приймають участь в реалізації функцій	Абсолютне значення параметра	Оцінка параметра, Ві	Коефіцієнт вагомості параметра, φ_i	Коефіцієнт рівня якості, $К_{тр}[F_{1k}]$
1	2	3	4	5	6	7
F1(x1)	а	x1	3	2	0,2	0,4
	б	x1	2,7	5,8	0,2	0,6
	в	x1	2,5	3,2	0,2	0,5
F2(x2)	а	x2	0,5	5,3	0,1	0,55
	б	x2	0,7	5,5	0,1	0,5
	в	x2	1	5,2	0,1	0,55
F3(x3)	а	x3	256	1,5	0,18	0,27
	б	x3	128	5,5	0,18	0,99
	в	x3	64	4,3	0,18	0,22

За даними таблиці 3.11 та розрахунків отримано показники рівня якості для кожного з варіантів програмного продукту

$$К_{трк} = К_{тр}[F_{1k}] + К_{тр}[F_{2k}] + К_{тр}[F_{3k}] \quad (3.12)$$

$К_{тр1} = 0.4 + 0.5 + 0.27 = 1.17$

$К_{тр2} = 0.4 + 0.5 + 0.99 = 1.89$

$К_{тр3} = 0.4 + 0.5 + 0.22 = 1.12$

$К_{тр4} = 0.6 + 0.55 + 0.27 = 1.42$

$К_{тр5} = 0.6 + 0.55 + 0.99 = 2.14$

$К_{тр6} = 0.6 + 0.55 + 0.22 = 1.37$

$К_{тр7} = 0.5 + 0.55 + 0.27 = 1.32$

$К_{тр8} = 0.5 + 0.55 + 0.99 = 2.04$

$К_{тр9} = 0.5 + 0.55 + 0.22 = 1.27$

Як бачимо з розрахунків, кращим є 5 варіант, у якого коефіцієнт технічного рівня ($К_{тр}$) має максимальне значення.

Логічна структура включає набір функціонально-логічних модулів, що мають власні процедури і об’єкти, вони представляють стандартні прототипи додатків інформаційних баз даних: форми, звіти запитів, вікна для перегляду таблиць даних і т.д., а також програмні одиниці, що реалізують деякі автоматизовані функції або завдання предметної досліджуваної області [9].

На рис. 3.2 – 3.3 представлено структурні схеми класів програмного засобу, вони наглядно оформлені за допомогою відповідної UML-діаграми [10, 28].

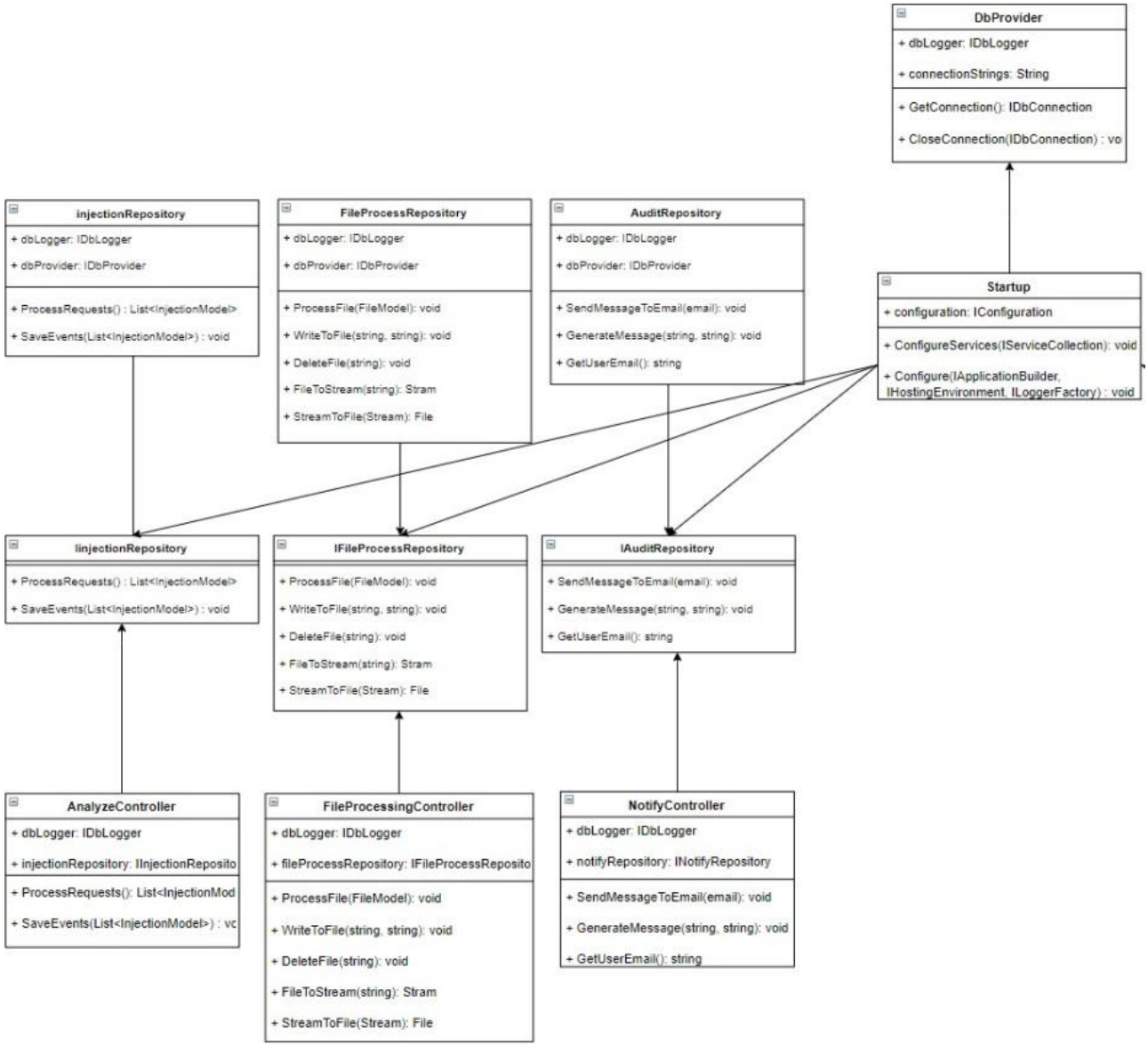

Рисунок 3.2 - Структурна схема класів програмного засобу

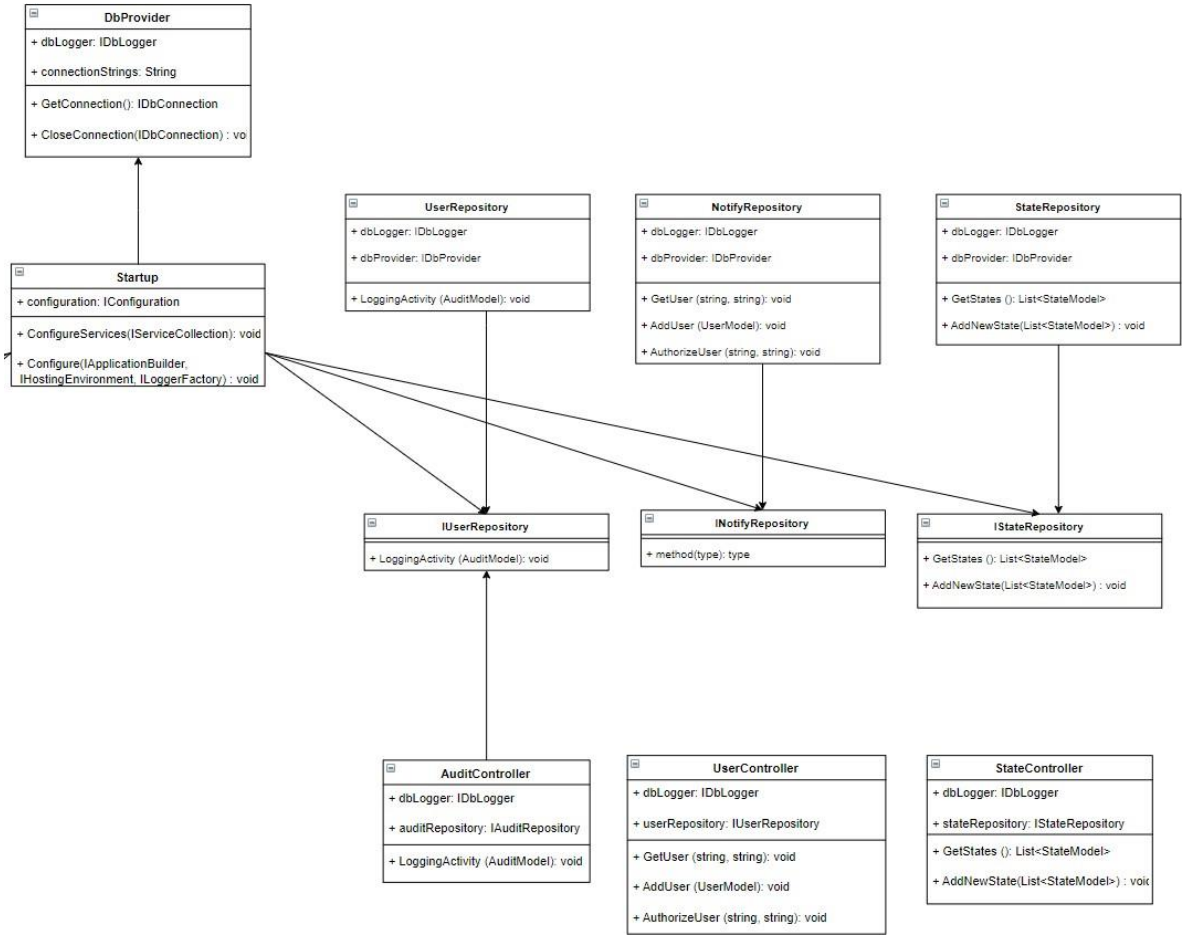

Рисунок 3.3 - Структурна схема класів програмного засобу

3.4 Висновки до розділу 3

Метою даного дослідження є розробка всеохоплюючої моделі оцінювання вразливостей різних видів, яка повинна враховувати існуючі підходи ін’єкції та розробити ефективну стратегію щодо «закриття дір», тобто усунення вразливостей в існуючій системі, а також виявити спроби проникнення в систему через ін’єкцію, та реагування відповідно до загроз. Таким чином, ми оглянили ключові види ін`єкційних атак, та потенційну шкоду, а також підходи та моделі виявлення подібних атак в залежності від їхньої специфіки. З урахуванням проведеного аналізу стає зрозумілим, що більшість рекомендацій, від науковців у цій галузі є розрізненими і не забезпечують дієвого захисту ін’єкційним атакам, тому це спонукало сформулювати та конкретизувати завдання дослідження з метою розробки універсальної моделей протидії ін’єкційним атакам.

4 СИСТЕМА ПІДТРИМКИ ПЕРЕВІРКИ ВЕБ-ДОДАТКІВ НА ВРАЗЛИВОСТІ

4.1 Обґрунтування та вибір середовища реалізації

Для реалізації програмного засобу перевірки веб-додатків було обрано мову програмування C#. Вона дозволяє провести реалізацію як серверної так і клієнтської частин. Ця мова є широко вживаною, вона зручна у використанні, швидко розвивається та може бути використана для будь-яких цілей.

Для кращої організації коду та спрощення розробки було використано платформу ASP.NET Core 2.0. ASP.NET Core - це досить популярний редизайн ASP.NET 4.x, архітектурні зміни якої призвели до меншої, більш модульної структурної системи [31].

Реалізація серверної частини відбувається за допомогою C#. Цей технологічний набір є популярним та показує швидкий розвиток. В першу чергу було розроблено структуру API, забезпечувати яку повинен був сервер, крім цього продумано було всю маршрутизацію та потік даних на вході і на виході [32].

Серверна частина розгортається окремо від клієнтської частини, тому структурно частини незалежні і кожна має свій простір для більшого вдосконалення та масштабованості.

З основних переваг ASP.NET Core можна відзначити [35]:

– Уніфікований підхід до створення інтерфейсів веб-додатків і WebApi;

– Гнучка архітектура, зручна для реалізації модульного тестування;

– Razor Pages спрощує та підвищує ефективність кодувальних процесів;

– Кросплатформеність: підтримка Windows, MacOS і Linux у розробці та запуску.

– Відкрите ПЗ з орієнтацією на потреби розробників;

– Інтеграція з сучасними клієнтськими фреймворками та підходами у

розробці;

- Передова конфігураційна система;
- Вбудована функціональність ін'єкції залежностей;
- Ефективна та гнучка система обробки HTTP-запитів;
- Можливості хостингу в IIS, Nginx, Apache, Docker, тощо;
- Варіативність у версіонуванні додатків при роботі з .NET Core;
- Зручний інструмент для сучасної веб-розробки.

З метою візуалізації інформації про наявність вторгнення було використано мову AngularMaterail та мову розмітки HTML, які дають можливість динамічно змінювати види та стани об’єктів і елементів [39].

Реалізація клієнтської частини відбувалась на мові програмування JavaScript із застосуванням допоміжних фреймфорків, наприклад Angular – платформу розробки програмного засобу з відкритим кодом, що використовують для побудови інтерфейсів користувачів (зовнішній інтерфейс) [40].

Інші компоненти можна розглядати як невеликі частини інтерфейсу, які не залежать один від одного. Існує можливість формувати прості додатки зі списком компонентів та відповідною системою пошуку, щоб отримати інформацію за словом. Вікно пошуку, блоки з визначеними іменами та основний екран, де розташовані дві інші коробки, окреслюються як окремі компоненти в Angular [41].

4.2 Алгоритм роботи системи підтримки перевірки веб-додатків на вразливості

Усі заходи поділяють на пасивні (що запускаються один раз та далі працюють самостійно) та активні (нормальне функціонування яких відбувається тільки у присутності або під контролем людини), а також превентивні (використовуються у випадках попередження виникнення SQL-атаки) і реакційні (використовуються у випадках виникнення загрози або усунення шкідливого впливу SQL-атаки).

На рис. 4.1 наведена блок схема виявлення та зупинення SQL -атаки.

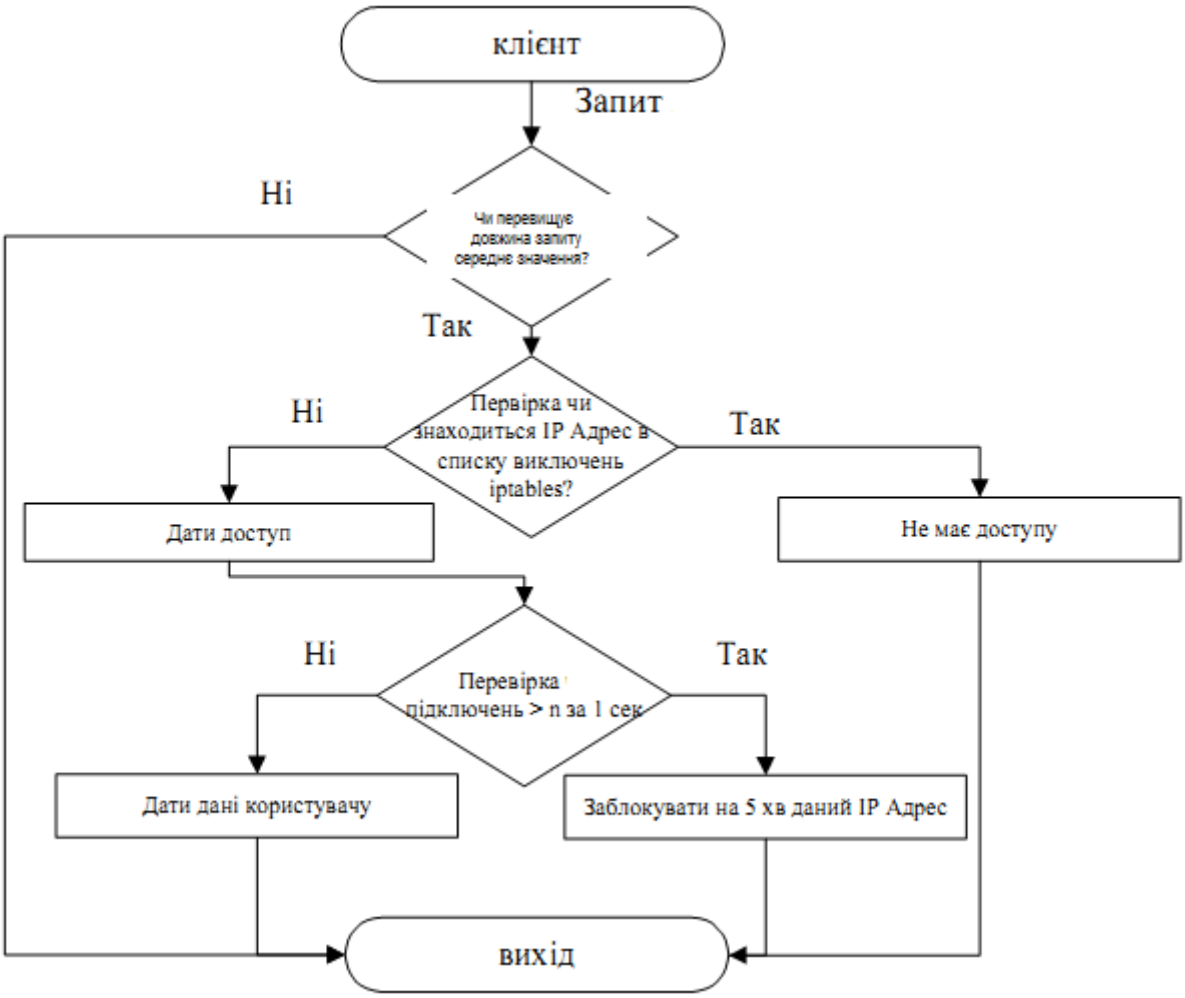

Рисунок 4.1 - Алгоритм виявлення та протидії SQL -атакам [35; с. 11]

Отже, на початку виконується перевірка чи відбувається збільшення часу підключення до сервера чи тривалості HTTP-запитів порівняно зі звичайними показниками. У разі виявлення відхилень, це вважається підставою для здійснення перевірки чи відбувається атака на сервер і чи вміщують HTTP-запити небезпечну ін’єкцію. Спочатку аналізуються довжина запитів та перевіряється чи належать IP-адреси, з яких надходять запити, до списку небезпечних. Після цього розраховується кількість запитів від кожної IP-адреси. Коли отримане число більше, ніж x, то це є підставою вважати, що відбувається ін’єкційна атака. Для захисту і припинення надходження запитів із шкідливим кодом, необхідно заблокувати підозрілі IP-адреси на кілька хвилин. Це надасть можливість перевірити будову запитів і зупинити ін’єкційну атаку шляхом блокування HTTP-запитів з підозрілих адресів. Ця схема вважається дієвою для більшості типів атак.

Єдина різниця, що для кожного типу атаки буде відрізнятися число x, від якого починається відлік перевищення довжини запиту чи його часу проходження [37].

Аналогічно можна перевіряти час тривалості запиту, який відрізняється від стандартного, оскільки вже має додатковий код.

Коли з’являється підозра, що відбувається SQL-атака, усю інформацію про з’єднання варто записувати в окремий лог. Ключовими полями в файлі, котрі нас цікавлять є час і вхідний IP-адрес з’єднання, щоб надалі перевіряти дані адреси. Потім аналізуємо дані з метою виявлення всіх адрес, з яких число та тривалість проходження запитів перевищує заданий показник. Потім встановлені адреси блокуються в iptables.

У ситуаціях, коли існує повторювана загроза ін'єкційних атак на певний інтернет-ресурс, ефективним рішенням є використання засобів технічного і інженерного захисту. Цей вид захисту, відомий як ІТЗ, включає в себе набір технічних інструментів, заходів безпеки та спеціалізованих організацій, які діють з метою забезпечення надійного захисту конфіденційної інформації.

Різноманітність цілей, задач, об’єктів захисту і заходів, що проводяться припускає розгляд деякої системи класифікації за видом, орієнтацією та іншими характеристиками.

Наприклад, інструменти ІТЗ можна розглядати виходячи з об’єктів їх впливу. У цьому сенсі вони можуть впроваджуватися для захисту людей, інформації, матеріальних засобів, фінансів тощо.

Різноманіття класифікаційних характеристик дає змогу розглядати ІТЗ за об’єктами впливу, способами реалізації, масштабом охоплення, характером заходів, класом засобів зловмисників, протидія яким чиниться зі сторони служби безпеки.

Безсумнівно, такий поділ засобів захисту інформації досить умовний, бо на практиці частіше вони реалізуються в комплексі.

Від атак середнього рівня небезпеки для захисту можно використовувати алгоритм проходження запитів лише визначеної довжини. Для цього застосовують спеціальні програми з урахуванням топології мережі та шляхів здійснення запитів.

Формування спеціалізованих програмних інструментів для аналізу та створення структури окремих мережевих елементів передбачає, перш за все, створення додаткових математичних моделей для цих структур. Ці моделі повинні не лише точно відображати конфігурації систем, але й бути ефективними для їх подальшої інтеграції. Крім того, важливим є розробка відповідних обчислювальних методик для ідентифікації та нейтралізації потенційно шкідливих запитів. Для цього можна використати графи як математичні моделі топологій. Сьогодні розроблено досить широке коло методів визначення різних шляхів між початковою та кінцевими вершинами, контурами чи циклами, остовами дерев, множинами вершин чи дуг, що задовольняють окресленим умовам тощо. Стає нагальною потреба у розробці не лише нових методів обробки графів топологій з врахуванням можливостей ін’єкції, але й спеціальних засобів представлення графів для обробки та проходження запитів, наприклад, у множині чи списку вершин та дуг, що зменшує ефективність відповідних програмних продуктів топологічного аналізу та синтезу. Отже, найбільш придатними вважаються матричні моделі топологій систем, оскільки вони є більш структурованими і простіше обраховуються.

Виникає потреба у комплексному підході до розробки матричних моделей аналізу та синтезу топологій HTTP-запитів. Стає актуальною науково-технічна проблема розробки обчислювальних методів аналізу матричних моделей топологій для формування ефективних програмних засобів для архітектури проектування таких систем.

Розглянемо, яким саме чином може бути представлена зазначена нами двоелементна система. Формально наведену таблицю можна надати у вигляді дводольного графу, один сегмент грифу – користувачі, інший – запити, ними надіслані. Отже, відбувається формування матриці цих зв’язків, яку можна зберігати у форматі csv на комп’ютерному носії.

Аналіз графів можна здійснювати як в загальному вимірі, так і специфічно для індивідуального ресурсу, з орієнтацією на виявлення найбільш ефективного способу обробки запитів протягом дозволеного часового проміжку. Розглянемо ілюстративний приклад дослідження графу, який спрямований на визначення

найвдалішого маршруту для передачі інформації. Уявімо ситуацію, де потрібно обробити запити до ресурсу за участю кількох користувачів, іноді вони надсилають запити до ресурсу, що перевищують середню довжину. Цифри, розмішені над стрілками демонструють різноманітність шляхів надсилання запитів (рис. 4.2).

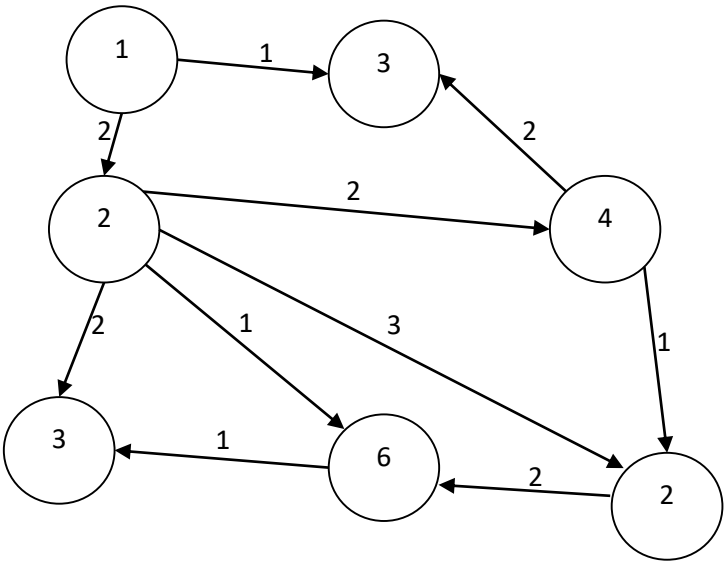

Рисунок 4.2 - Граф надходження запитів від різних користувачів

Виконаємо обчислення складності для даного графу, зобразивши граф у вигляді ієрархічної структури (рис. 4.3):

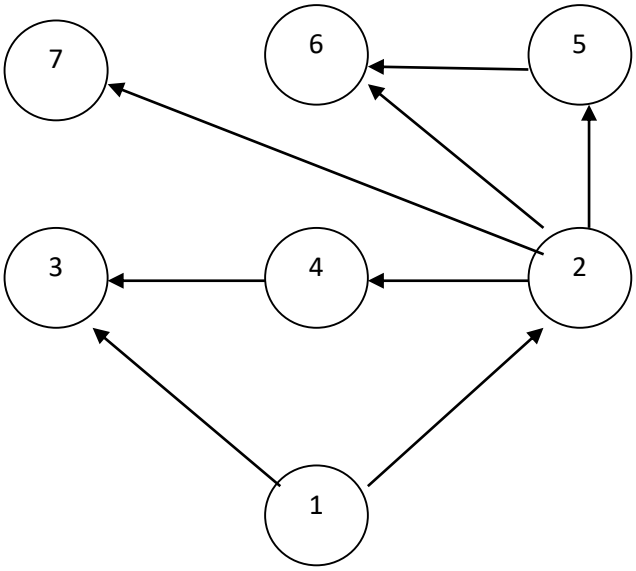

Рисунок 4.3 - Ієрархічний граф комунікаційної мережі

Перетворимо отриманий трирівневий граф на еквівалентний, у якому відсутні суміжні вершини на однакових рівнях (рис. 4.4).

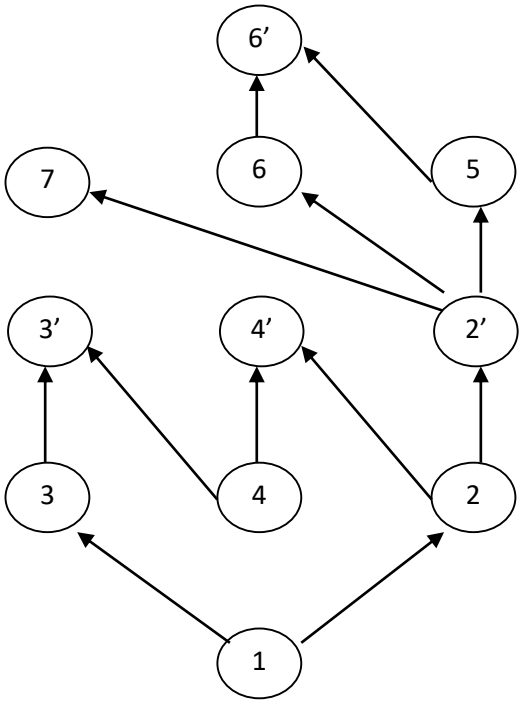

Рисунок 4.4 - Еквівалентна схема обробки запитів у мережевій системі

Одержаний граф (що зображений на рисунку 4.4) описується наступними матрицями інциденцій:

$$W1 = \begin{array}{c|ccc} & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 0 \end{array} \qquad W2 = \begin{array}{c|ccc} & 2` & 3` & 4` \\ \hline 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 \end{array} \qquad W3 = \begin{array}{c|ccc} & 5 & 6 & 7 \\ \hline 2` & 1 & 1 & 1 \\ 3` & 0 & 0 & 0 \\ 4` & 0 & 0 & 0 \end{array}$$

$$W4 = \begin{array}{c|c} & 6` \\ \hline 5 & 1 \\ 6 & 1 \\ 7 & 0 \end{array} \qquad W = W1W2W3W4 = \begin{array}{c|ccc} & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 0 \end{array} * \begin{array}{c|ccc} & 2` & 3` & 4` \\ \hline 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 \end{array}$$

$$W3W4 = \begin{array}{c|ccc} & 2` & 3` & 4` \\ \hline 1 & 1 & 1 & 1 \end{array} * \begin{array}{c|ccc} & 5 & 6 & 7 \\ \hline 2` & 1 & 1 & 1 \\ 3` & 0 & 0 & 0 \\ 4` & 0 & 0 & 0 \end{array} \qquad W4 = \begin{array}{c|ccc} & 5 & 6 & 7 \\ \hline 1 & 1 & 1 & 1 \end{array} * \begin{array}{c|c} & 6` \\ \hline 5 & 1 \\ 6 & 1 \\ 7 & 0 \end{array} = 2$$

Для графа на рис. 4.4 знайдемо максимальний шлях з вершини a_1 у вершину

a_7

Для вершини a_1 приймаємо $q_s^{\text{макс}}(a_1 a_1) = 0$. Для вершин

$a_2, a_3 \div q_s^{\text{макс}}(a_1 a_1) = 2, q_s^{\text{макс}}(a_1 a_3) = \text{макс}(1{,}2 + 2 + 2) = 6$

Для вершини $a_4 \div q_s^{\text{макс}}(a_1 a_4) = 2 + 2 = 4.$

Для вершини $a_5 \div q_s^{\text{макс}}(a_1 a_5) = \text{макс}(2 + 3{,}2 + 2 + 1) = 5$

Для вершини $a_6 \div q_s^{\text{макс}}(a_1 a_6) = \text{макс}(2 + 1{,}5 + 2) = 7$

Для вершини $a_7 \div q_s^{\text{макс}}(a_1 a_7) = \text{макс}(2 + 2{,}7 + 1) = 8$

Значення функції на максимальному шляху становить вісім одиниць, а сам маршрут складається з точок 1-2-5-6-7. Це означає, що найкращий варіант для пересилання запиту запита від користувача 1 до ресурсу 7 за визначеними параметрами проходить через вузли 2, 5 та 6. Тим часом, інші вузли відкидають запити, що не відповідають обмеженням за довжиною.

Інформація про присутність або відсутність підозрілих запитів може бути відображена через створення біографу (графу з двома компонентами). Це дозволяє нам не просто кількісно оцінювати наявні проблемні запити, але й аналізувати їхню якість, а саме: розрізняти ті запити, які саме були вирішені автоматично, а які – ні.

Використання цієї моделі допомагає відстежувати розмір запитів та виявляти конфлікти, коли окремі запити від різних користувачів перевищують норму. Це сприяє запобіганню втрати інформаційних даних, оскільки зберігається історія всіх запитів, у тому числі дата виникнення підозрілого запиту та дані про користувача, який цей запит ініціював, що дозволяє в разі необхідності оцінити ризики використання інформаційного ресурсу певними користувачами.

4.3 Програмні характеристики моделювання СППР

Для аналізу фізичного будови системи виявлення вразливостей в програмному забезпеченні, були побудовано діаграми компонентів, які представлені на рис. 4.5 – 4.6 [28].

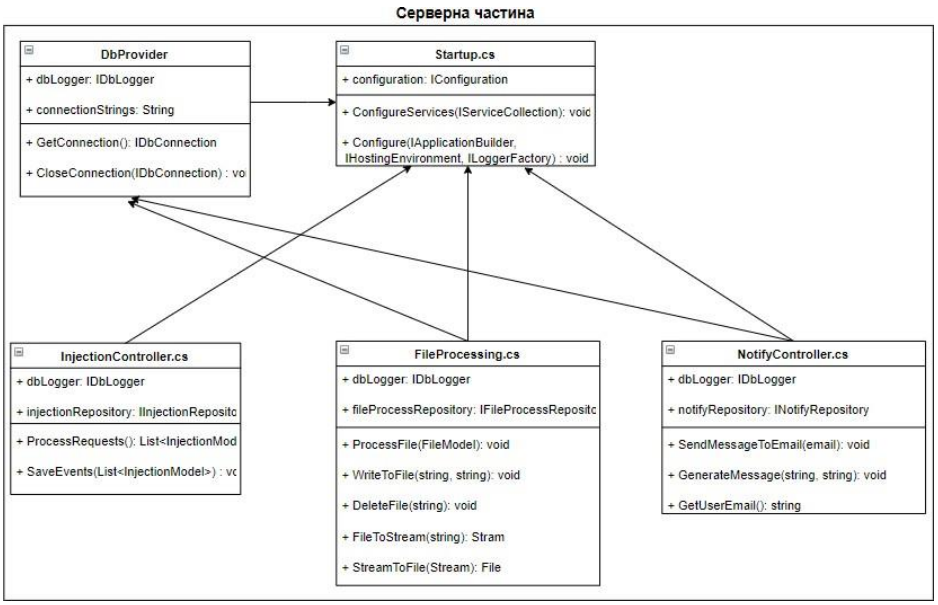

Рисунок 4.5 - Діаграма компонентів серверної частини програмного засобу виявлення вразливостей

На рис. 4.5 можна побачити структуру файлів програмного засобу виявлення вразливостей, а саме серверної частини, в яких реалізовано класи, що було описано у п. 3.2 [13]:

Файл Startup.cs відіграє ключову роль у функціонуванні інструменту для виявлення недоліків безпеки. InjectionController.cs включає класи для пошуку, збору і аналізу вразливостей. DbProvider.cs вміщує клас для доступу до даних у базі даних. FileProcessing.cs включає клас для операцій з файлами, таких як запис та читання даних. DbConnecction.cs містить клас для з'єднання з базою даних. NotifyController.cs включає клас для надсилання повідомлень адміністратору про SQL-ін'єкції.

На рис. 4.6 представлена структура файлів інструменту для виявлення вразливостей, особливо клієнтської частини, яка містить реалізацію описаних у пункті 3.2 класів:

stateCtrl.js – JavaScript-файл для управління клієнтською частиною інструменту, включаючи управління даними про стан веб-застосунку;

index.html – головна HTML-сторінка застосунку з виявлення вразливостей;

settings.html – HTML-сторінка для налаштувань інструменту;

settingsCtrl.js – JavaScript-файл для управління налаштуваннями клієнтської частини, включаючи їх зміну, видалення або додавання адміністратором;

authorization.html – HTML-сторінка для роботи з обліковими записами користувачів;

usersCtrl.js – JavaScript-файл для управління авторизацією, реєстрацією та видаленням адміністраторів у клієнтській частині інструменту.

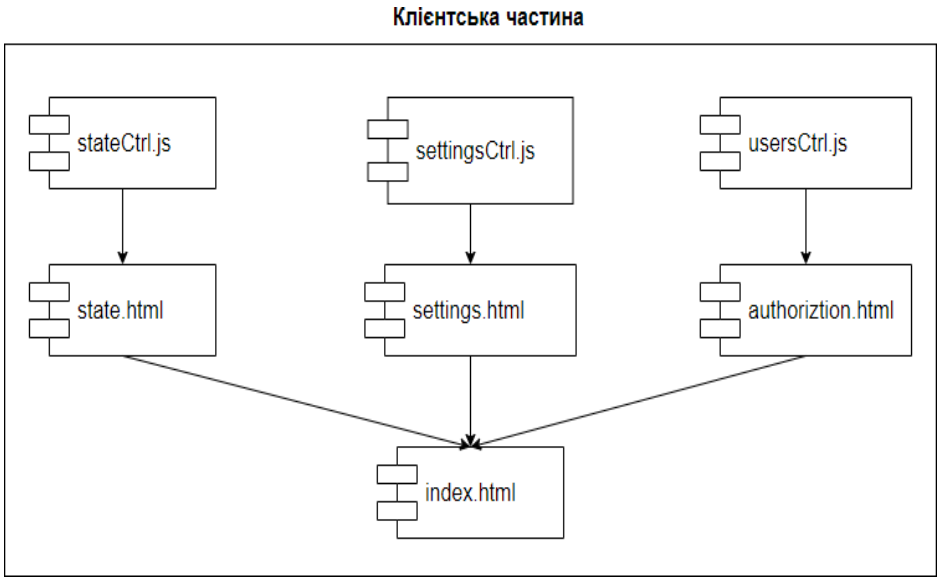

Рисунок 4.6 - Діаграма компонентів клієнтської частини програмного засобу виявлення вразливостей

4.4 Аналіз ефективності модулю прийняття рішень системи підтримки перевірки веб-додатків на вразливості

Експериментальне дослідження здійснювалось шляхом моделювання атаки ін’єкційного типу та впровадження засобів для її виявлення та нейтралізації. Спершу визначаємо необхідний рівень захисту за розробленою моделлю. У випадках, коли цей рівень не перевищує 50%, активний захист не є обов'язковим. При показниках між 50 і 60% варто слідкувати за окремими характеристиками системи, що можуть сигналізувати про ризик ін’єкцій. За показником понад 60% рекомендується запустити моніторинг вхідних запитів на визначених ресурсах. І у

ситуаціях, коли рівень безпеки перевищує 80%, доцільно встановити більш вдосконалене програмне забезпечення для детального аналізу кожного запиту.

Розглянемо процес відстеження запитів, коли рівень небезпеки вище 60%.

Експериментальне моделювання SQL-атак виконують на базі спеціально створеної розподіленої мережі з використанням програм netmap, nping та WPEPro. В процесі аналізу мережеві сервіси часто зазнають впливу SQL-атак різної інтенсивності які вбудовуються ін’єкційні елементи в HTTP-запити. Система, використовуючи розроблену модель, повинна оцінити потенційну небезпеку кожного запиту і відповідно реагувати: блокувати запити з високим ризиком, перенаправляти їх на детальний розгляд для виявлення небезпечних елементів або визначати їх як безпечні і пропускати. Для збору даних використовується спеціальний аналітичний інструментарій, що базується на відкритих бібліотеках WinPcap та jpcap. Приклад базової інформації, що використовується для проведення досліджень, наведено в табл. 4.1.

При цьому з метою максимального наближення до реальних умов справжнього середовища використання мережевих служб під час атак SQL-ін'єкцій, було створено тестовий потік даних. Цей потік був унікальною комбінацією реального мережевого трафіку та штучно створених даних, які включали запити, що містили шкідливі коди.

Після цього проводять розрахунок ентропії для кожної послідовності та з отриманих значень визначають, чи відбувається атаки на певний момент (табл. 4.2).

Таблиця 4.1 - Параметри моделювання SQL-атак

№ послідовності	Кількість пакетів	Час передачі пакетів у хвилинах	
		Звичайний режим	Режим атаки
1	50000	15	20
2	100000	30	35
3	150000	60	65
4	200000	75	86
5	250000	84	98

Таблиця 4.2 - Значення ентропії для послідовностей пакетів

№ послідовності	Значення ентропії H без атаки, біт	Значення ентропії H* при загрозі ін’єкційної атаки, біт
1	2,1	3,2
1	3,32	1,28
2	3,28	1,51
1	2,0	3,1
3	3,21	1,93
4	3,25	1,71
5	3,39	1,88

Таким чином, обчисливши ентропію HTTP-запитів можна виявляти ініціювання ін'єкційних атак та ефективно протидіяти їх впливу. Іншими словами, аналізування мережевого трафіку в реальному часі служить як детектор незвичайних подій або вхідних запитів із підозрілим змістом. Зокрема, показники ентропії чутливі до змін під час SQL-атак, де її рівень знижується приблизно на 70% через подовження запитів додатковими SQL-елементами. Важливо, що розмір пакету не впливає на ці обчислення. У даному контексті ентропія змінюється відповідно до характеристик приходящих пакетів, наприклад, зі збільшеним обсягом або спеціальними ключовими словами у тексті. Таким чином, використання ентропії для ідентифікації початку ін'єкційної атаки може бути ефективним у таких системах, як IDS (системи виявлення вторгнень), системи управління інформаційною безпекою корпорацій, а також у системах, що підтримують процес прийняття рішень.

Як уже зазначалося, існуючі системи виявлення загроз можна поділити на дві категорії: виявлення аномалій та виявлення характеристик. Певною проблемою систем, орієнтованих на виявлення характеристик, є їх обмеженість у виявленні специфічних видів атак, особливо тих, які вважались найбільш загрозливими на момент створення системи. Це призводить до необхідності адаптації системи при

появі нових атак чи зміні характеристик відомих атак. Що стосується систем виявлення аномалій, то вони базуються на складніших припущеннях щодо поведінки мережевого трафіку, наприклад, виходячи з гіпотези про статистичну однорідність. Тим не менш, обмеження цих припущень та умови їх застосування часто залишаються поза обговоренням. Відповідно, навіть незначні варіації в структурі трафіку або зміни в наданих послугах можуть потребувати переосмислення та доопрацювання алгоритму виявлення. Однією з можливих стратегій розв'язання цієї проблематики є застосування комплексного підходу до створення системи захисту, що охоплює такі аспекти, як моніторинг системи, збереження історії транзакцій, створення спеціалізованого репозиторію для аналізу діяльності та стратегій протидії зловмисникам. Пропонується побудувати систему захисту на основі наступних компонентів:

– агенти нагляду;

– засоби первинної обробки даних та їх збереження;

– репозиторій для реєстрації та зберігання даних про транзакції, який описує діяльність системи;

– репозиторій з аналітичними інструментами для ідентифікації потенційних загроз та ознак несанкціонованої активності;

– агенти для проведення контрзаходів [13; с. 572].

Важливим елементом такої системи є визначення відповідного математичного забезпечення для кожного аспекту роботи:

Моніторинг мережного трафіку. Перехват пакетів для оцінювання обсягу та характеристик трафіку, а також активності користувачів. Реалізація цього процесу передбачає створення алгоритмів, що регулюють частоту і обсяг перехоплення пакетів з урахуванням навантаження на мережу та інших критеріїв. Надмірне перехоплення пакетів може уповільнити трафік, тоді як рідкісне перехоплення може призвести до виникнення неконтрольованих зон, де інформація буде втрачена [13; с. 572].

Попередня обробка захоплених пакетів, визначення потенційно

найнебезпечніших загроз та їх зберігання в базі даних. Важливість цього етапу полягає в швидкій аналітиці з мінімальним використанням ресурсів, тому ефективним рішенням є застосування простих та гнучких методів встановлення порогів або, при потребі, використання послідовного методу CUSUM.

Аналіз даних при завантаженні у сховище, виявлення атак, оцінка загроз. Завдяки збереженню даних у базі, можна здійснити детальну оцінку і виявити можливі вразливості. Оптимальним рішенням для цього є використання описаних раніше алгоритмів, зокрема CUSUM та методу рухомого середнього.

Фоновий аналіз інформації для ідентифікації спроб виявлення, атак зниження ефективності та ритмічних нальотів. Цей процес може бути неперервним або відбуватися за попередньо встановленим графіком. Враховуючи, що такі атаки становлять менший ризик, з'являється можливість для глибшого аналізу цих інцидентів. Для цього застосовуються методи Data Mining, системи інтелектуальних правил, нейронні мережі тощо.

Прийняття рішення про виявлення атаки. При перевищенні певних порогових значень на одному з попередніх етапів формується признак про можливу загрозу атаки. У такій ситуації активізується розроблена система експертної оцінки, завданням якої є аналізувати потенційну небезпеку та ухвалювати обґрунтовані рішення стосовно запуску контратак.

Оцінюючи ризики, необхідно обрати підходящу модель, підтвердити її достовірність та розробити методику реагування. При виявленні атаки негайно виникає необхідність у визначенні відповідних контрзаходів. Відповідь на атаку залежить від її характеру та унікальних характеристик, тому стратегії відповіді варіюються. Це означає, що потрібно розробити «стратегію» реагування, яка буде адаптуватися в залежності від умов ефективності контрзаходів, наприклад, якості обслуговування користувачів. Розробка потенційних стратегій має базуватися на аналізі взаємодії між атакуючими та захисними агентами. Вивчення аналітичних моделей допомагає зрозуміти ефективність контрзаходів та їх можливі наслідки. Ігровий підхід тут виникає через саму суть конфліктної взаємодії між нападником та системою захисту, де ключовим аспектом є вплив на завантаженість системи. Це

може стосуватися як загальної завантаженості, так і завантаженості критично важливих елементів системи, таких як процесор, оперативна пам'ять чи мережеві канали [3; с. 573].

Для формування стратегії протистояння важливо спершу проаналізувати ключові характеристики моделі конфлікту, в межах якої відбувається суперництво. Цей процес включає в себе кілька етапів:

– Визначення типу динаміки.

– Аналіз чисельності та сили атакуючих.

– Оцінювання рівня загрози.

– Визначення можливості опору та передбачення можливих наслідків.

– Реалізація заходів протидії та аналіз результатів, порівнюючи їх із прогнозами.

Після впровадження захисних стратегій система повина оцінити їхею ефективність, відстежуючи рівень загрози. У разі продовження атаки, стратегію необхідно скоригувати [3; с. 573].

Для ефективного використання даної моделі, можливо внести певні доповнення у процес виявлення загрози ін'єкційних атак. Це передбачає аналіз додаткових параметрів та оцінку їх впливу на стратегії захисту від атак.

Серед параметрів, що враховувались при виборі середовища проведення експерименту, були визначені наступні параметри:

Таблиця 4.3 - Умови проведення дослідницького експерименту

Значення	Показники рівня захисту			Результат ступеню рівня захисту від ін’єкційних атак
	X1	X2	X3	
Контрольне	37	59	66	64
Експериментальне	38	57	67	64

Отже, за допомогою цієї моделі ми здатні проаналізувати ступінь захисту від атаки під час експерименту, враховуючи заздалегідь відомі параметри:

$\hat{Y}КЗ = 0,967 + 0,640 \cdot 37 + 0,343 \cdot 59 + 0,287 \cdot 66 = 63,9\%$

$\hat{Y}ЕЗ = 0,967 + 0,640 \cdot 38 + 0,343 \cdot 57 + 0,287 \cdot 67 = 64,1\%$

Отже, оцінка за моделлю ідентична фактичному значенню з точністю до 0,1%. Це явно підтверджує відмінну ефективність даної моделі і підтримує можливість її використання в майбутньому для передбачення рівня захисту від потенційних атак.

Отже, у цій моделі проводиться оцінка ймовірність того, що інформаційний ресурс не зможе протистояти діям агенту загрози. Ця оцінка базується на основі показників потенційної загрози та ефективності захисних заходів. Результатом виконання даного етапу є визначення уразливості, яке обчислюється за допомогою матриці визначення уразливості, поданої в табл. 3.4 [46].

Таблиця 4.4 - Матриця визначення ймовірності вразливості

Вразливість						
Можливість загрози	ДВ	ДВ	ДВ	ДВ	В	ДВ
	В	ДВ	ДВ	В	С	Н
	С	ДВ	В	С	Н	ДН
	Н	В	С	Н	ДН	ДН
	ДН	С	Н	ДН	ДН	ДН
		ДН	Н	С	В	ДВ
	Ефективність захисних засобів					

Також враховується можливість імплементації ін'єкційних атак протягом певного часового інтервалу. Обчислення виконується на основі частоти подій ризику та вразливості, отриманих під час аудиту інформаційного ресурсу. Результатом цього кроку є значення частоти подій, які призводять до втрат, та визначається за допомогою матриці частоти подій втрат, наведеної у табл. 3.5 [46].

Таблиця 4.5 - Матриця визначення частоти подій, що призводять до негативних наслідків

Частота подій, що приводить до негативних наслідків						
Частота подій реалізації загрози	ДВ	С	В	ДВ	ДВ	ДВ
	В	Н	С	В	В	В
	С	ДН	Н	С	С	С
	Н	ДН	ДН	Н	Н	Н
	ДН	ДН	ДН	ДН	ДН	ДН
		ДН	Н	С	В	ДВ
	Вразливість					

Після цього проводиться оцінка ймовірного обсягу збитків внаслідок атаки.

Для визначення можливих наслідків та їхньої імовірності можна скористатися наступними трьома пунктами:

– визначити можливий сценарій подій, що найімовірніше призведе до максимальних наслідків;

– визначити розмір збитків, пов'язаних із кожним з цих сценаріїв;

– підсумувати загальну величину збитків для кожної форми втрат.

Оцінити ймовірну величину збитків можна наступним чином:

– визначити найбільш ймовірну загрозу чи подію;

– оцінити розмір збитків для кожної форми втрат;

– підсумувати суму збитків.

На даному етапі відбувається порівняння якісних і кількісних показників.

Після цього здійснюється визначення джерела ризику ін'єкційних атак, чітке формулювання критеріїв моніторингу та оцінка можливих збитків у разі успішної атаки.

На цій стадії процесу відбувається комплексне узагальнення інформації, зібраної на попередніх етапах. Виходячи з цієї зібраної інформації, визначаються потенційна частота випадків ін'єкційних атак та можливий обсяг майбутніх збитків.

Для коректного визначення рівня ризику, пов'язаного із ін'єкційними

атаками, необхідно мати на увазі два ключові аспекти:

– підрахована кількість випадків, які можуть призвести до втрат;

– підрахована ймовірна величина збитків.

Дану інформацію можна представити у різноманітних форматах. У більшості випадків слід вказувати оцінений ризик втрат на найвищому рівні, щоб забезпечити інформованість особи, яка приймає рішення, стосовно можливого найнегативнішого розвитку подій.

Таблиця 4.6 - Матриця визначення величини ризику

Ризик						
Вірогідна величина втрати	Критична	В	В	К	К	К
	Висока	С	В	В	К	К
	Значна	С	С	В	В	К
	Середня	Н	С	С	В	В
	Низька	Н	Н	С	С	С
	Дуже низька	Н	Н	С	С	С
		ДН	Н	С	В	ДВ
Частота подій, що приводить до втрат						

На цьому етапі необхідно чітко визначити наступні величини: частота подій, що спричиняють збитки; імовірність збитки; максимальні збитки в найгіршому випадку.

За допомогою табл. 4.6 встановлюється величина ризику, а її значення розшифровується за допомогою таблиці 4.7 [3].

Таблиця 4.7 - Значення рівня ризику втрати інформаційних ресурсів

Значення	Рівень ризику
1	2
К	Критичний (більше 80;)

Кінець таблиці 4.7

1	2
B	Високий (більше 60%)
C	Середній (більше 50%)
H	Низький (менше 50%)

У деяких ситуаціях, з метою більш точної оцінки ризиків, пов'язаних із інформацією, можливо використовувати метод, що передбачає перехід від якісної до кількісної оцінки інформаційних ризиків.

4.5 Висновки до розділу 4

Таким чином, ми сформували модель перевірки додатку на вразливості на основі аналізу існуючих ризиків за видами, а також розробили методологію прогнозування окремих ризиків, за допомогою якої можна визначити ймовірність настання того чи іншого ризику ін’єкційної атаки і провести заходи по запобіганню втраті інформації.

Також було проведено аналіз можливих заходів по збереженості інформації в умовах ризику ін’єкційних атак і запропоновано методику зберігання інформації на розподілених ресурсах з контролем модифікації.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Роботу присвячено дослідженню методикам та моделям захисту і виявлення вразливостей веб-додатків. В цій роботі поставлено та виконано наступні завдання:

– розглянуто поняття вразливостей та атак, а також методи захисту від них. Особливу увагу приділено проблемі SQL-ін’єкцій. Ці ін'єкційні атаки можуть спричинити викрадення, знищення або спотворення інформації, а також можуть призвести до захоплення ОС користувача для подальшого контролю або просування по внутрішній мережі веб-додатку;

– проведено огляд існуючих стратегій протидії кібератакам, що ґрунтуються на ідентифікації атаки та виборі методів захисту буть то програмі, апаратні чи інші засоби. Ідентифікація ін'єкційних атак виконується за допомогою аналізу вхідних запитів на основі довжини, вмісту певних елементів, ключових слів, символів та інших оціночних критеріїв, що дозволяють ідентифікувати підозрілі дії та проводити глибокий аналіз за допомогою спеціалізованого програмного забезпечення;

– було визначено основну мету наукового дослідження. Головною метою дослідження виступає створення всебічної моделі захисту проти різноманітних ін’єкційних атак. Ця модель повинна включати в себе знання про існуючі способи ін’єкції, здатна ефективно виявляти та нейтралізувати слабкі місця в системі, а також виявляти нелегітимні спроби доступу через ін’єкції та запобігати їм. Отже, було проаналізовано ключові види ін’єкційних атак, небезпеки, які вони несуть, і підходи до розпізнавання таких атак. Результати дослідження показали, що більшість існуючих рекомендацій вчених щодо цього питання є розрізнені та не забезпечують достатньої протидії ін’єкційним атакам, що спонукало до формування та уточнення цілей дослідження у контексті розробки більш універсальних моделей захисту проти ін’єкційних атак;

– вибрано та обґрунтовано критерії розроблюваної моделі. В процесі цього вибору було враховано три ключові змінні, які використовувалися при

подальшому створенні системи захисту від ін'єкційних атак;

– здійснено розробку моделі захисту від ін’єкційних атак. Розроблена тривимірна модель і проведено аналіз її кореляційних властивостей, що демонструє відсутність взаємозалежності між вибраними параметрами. В результаті, ми розробили комплексну модель, яка забезпечує належний рівень захисту від ін’єкційних атак;

– проведено аналіз адекватності моделі. В ході цього дослідження було створено модель для оцінки потреби в рівні захисту для кожного вузла, залежно від трьох основних параметрів: ймовірності атаки, часу відновлення та значущості ресурсу, що піддавався атакам. Наша модель є лінійною та визначає необхідний рівень захисту для кожного об'єкта в системі, при цьому кожен параметр виявлення ін'єкцій може змінюватися окремо, враховуючи можливі наслідки атаки на кожний окремий об'єкт. Наша модель дозволяє ефективно виявляти можливі атаки та оптимізувати обсяг захисту, сприяючи зниженню загрози для всієї системи. Практична реалізація захисної моделі від ін'єкційних атак може бути здійснена як у статичному форматі (за результатами ручної настройки), так і у динамічному автоматичному режимі, в залежності від кількості об'єктів, що потребують захисту, і рівня загрози для кожного з них;

– розроблено методику проведення експериментального дослідження. Вона включала в себе формування пакетів запитів, що містять ін’єкцію та засобів реагування на загрозу. Під час проведення експерименту було виявлено ще один фактор, як ентропія, що також може бути включений у модель;

– виконано дослідження характеристик моделі оцінки вразливостей веб-додатків щодо ін’єкційних атак за допомогою методу чек-листа. В результаті, була створена модель на основі аналізу існуючих видів ризиків, і була розроблена методологія передбачення конкретних ризиків. Ця методологія дозволяє оцінити ймовірність виникнення певного виду ін'єкційної атаки і вжити необхідні заходи для запобігання втраті інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bastian Ballmann. Understanding Network Hacks: Attack and Defense with Python 3. 2021. С. 37-45.

2. Джейсон Андресс, Захист даних. Від авторизації до аудиту, 2021. 231-233 с.

3. Антонюк П. Є. Класифікація ймовірних способів вчинення атак на інформацію як напрям протидії комп'ютерній злочинності. URL: http://www.nbuv.gov.ua/portal/Soc_Gum/bozk/19text/g1927.htm. (дата звернення 22.10.2023)

4. Бабенко Т. В. Дослідження ентропії мережевого трафіка як індикатора DDoS-атак. Науковий вісник НГУ. 2013. Вип. 2. С. 86-89.

5. Багнюк Н. В. Види DDoS-атак та алгоритм виявлення DDoS-атак типу Flood-атак. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2015. Вип. 18. С. 6-12.

6. ВПРОВАДЖЕННЯ ТАКИХ СИМВОЛІВ SQL. URL: https://cqr.company/ua/web-vulnerabilities/sql-wildcard-injection/ (дата звернення 03.04.2023).

7. Гарасимчук О.І., Костів Ю.М. Оцінка ефективності систем захисту інформації. Вісник КНУ імені Михайла Остроградського. 2016. Вип. 1. С. 16–20.

8. Гнатюк С. Є. Математичні моделі оцінки та прогнозування надійності програмно-керованих засобів захисту інформації в системі урядового зв'язку. Ukrainian Information Security Research Journal. 2016. Вип. 2. С. 150–156.

9. Грищук Р. В. Атаки на інформацію в інформаційно-комунікаційних системах. Сучасна спеціальна техніка. 2011. Вип. 1(24). С. 61-66.

10. Державна служба спеціального зв'язку та захисту інформації України. Офіційний сайт. URL: https://cip.gov.ua/ (дата звернення 06.12.2023).

11. Діордіца І. Система забезпечення кібербезпеки: сутність та призначення. Інформаційне право. 2017. Вип. 7. С. 109–116.

12. Делембовський М.М., Шабала Є.Є., Терентьєв О.О. аналіз методів та

шляхів вирішення захисту інформації в інформаційно-телекомунікаційних системах // ГРААЛЬ НАУКИ, 2021. С. 249-254.

13. Єрмошин В.В., Хорошко В.О., Капустян М.В. Методика оцінки інформаційних ризиків системи управління інформаційною безпекою. Сучасний захист інформації. 2010. Вип. 3. С. 95–104.

14. Защита от XSS [Електронний ресурс] – Режим доступу: URL: http://www.spysoft.net/zashhita-ot-xss/ (дата звернення 08.09.2023)

15. Ettore Galluccio, Edoardo Caselli, Gabriele Lombari, SQL Injection Strategies: Practical techniques to secure old vulnerabilities against modern attacks: Packt Publishing, 2020. 93-318 с.

16. Інформаційна безпека (соціально-правові аспекти) : підруч. / за заг. ред. Є. Д. Скулиша. К. : КНТ, 2010. 776 с.

17. Інформаційна безпека. Підручник / В. В. Остроухов, М. М. Присяжнюк, О. І. Фармагей, М. М. Чеховська та ін.; під ред. В. В. Остроухова – К.: Видавництво Ліра-К, 2021. – 412 с.

18. Інформаційні дані. Збиток може бути не тільки матеріальним, а й, наприклад, моральним. https://datami.ua/informatsijna-bezpeka-vidi-zagroz-i-metodi-yih-usunennya/ (дата звернення 10.09.2023)

19. Як запобігти ін'єкції SQL? (Повний посібник). Кібербезпека. https://zephyrnet.com/uk/how-to-prevent-sql-injections-complete-guide/(дата звернення 11.09.2023)

20. Кошель А.О. Поняття ризику та його види при використанні земельних ресурсів у ринкових умовах. URL: http://www.nbuv.gov.ua/portal/Chem_Biol/Vldau/APK/2010_1/files/ 10kalimc.pdf. (дата звернення 05.11.2023)

21. Липінська Є.І. Зарубіжний досвід захисту інформації в сфері підприємництва та його використання в Україні. Порівняльно-аналітичне право. 2017. Вип. 5. С. 148–150.

22. Лук’янова В.В., Головач Т.В. Економічний ризик: навч. посіб. К.: Академвидав, 2007. 464 с.

23. Ляшенко О.М., Криль Я.М. Кадрова безпека у системі економічної безпеки підприємства. Економіка. Менеджмент. Підприємництво : зб. наук. праць. 2013. Вип. 25 (2). 220 с.

24. Маковецький О. М. Підходи до удосконалення методики оцінки ефективності комплексної системи захисту інформації / О. М. Маковецький, І. Р. Мальцева, Н. А. Паламарчук, Ю. О. Черниш, О. В. Шемендюк. Сучасні інформаційні технології у сфері безпеки та оборони. 2016. № 2 (26). С. 54–58.

25. Менеджмент інформаційної безпеки: підруч.: у 2 ч. / А. К. Гринь, О. Д. Довгань, В. І. Журавель та ін.; за заг. ред. Є. Д. Скулиша. К. : Наук.–вид. Центр НА СБ України, 2013. Ч.1. 456 с.; Ч.2. 604 с.

26. Міжсайтовий скриптинг. https://ukeywaf.com/baza/shho-take-mizhsajtovyj-skryptyng/ (дата звернення 15.10.2023)

27. Найвідоміші вразливості веб застосунків. XSS та SQL ін’єкції, вразливості автентифікації, 27 жовтня 2022. URL: https://dou.ua/forums/topic/40613/ (дата звернення 18.10.2023)

28. Невойт Я.В. Метод оцінювання стану захищеності інформаційних ресурсів на основі дослідження джерел загроз інформаційній безпеці : дис. канд. техн. наук. спец. 21.05.01. К. ДУТ, 2016. 110 с.

29. Нестеренко М.М., Романов А.О. Аналіз методів захисту серверів від розподілених TCP SYN-FLOOD атак. URL: conferenc.its.kpi.ua/proc/article/download/73108/68432 (дата звернення 18.09.2023)

30. Огляд фреймворків JavaScript. Що, для чого і коли використовувати. 14 вересня 2021. https://dou.ua/forums/topic/34739/ (дата звернення 19.09.2023)

31. Перевалова Л. В., Кваша С.В. Захист конфіденційної інформації: проблеми та шляхи вирішення. Вісник Національного технічного університету «Харківський політехнічний інститут». Збірник наукових праць. Тематичний випуск: Актуальні проблеми розвитку українського суспільства. 2011. № 30. С.109-112.

32. Проблеми впровадження сучасних стандартів інформаційної безпеки в умовах становлення національної системи кібербезпеки України. URL:

http://www.niss.gov.ua/content/articles/files/1_cPPP-standarts_27-04_Gn_var_FIN-732b6.pdf (дата звернення 05.12.2023)

33. Процедура міграції для MS SQL Server. https://help.eset.com/protect_install/90/uk-UA/db_migration_sql.html (дата звернення 19.09.2023)

34. Разбираємось, в чому різница між Data Mining и Data Extraction. 21 жовтня 2020. https://habr.com/ru/companies/skillfactory/articles/524336/ (дата звернення 20.09.2023)

35. Сенейко Ю. В. Сучасні підходи до трактування категорії «ризик». Регіональна економіка. 2006. № 1. С . 206-211.

36. Створення комплексних систем захисту інформації. https://iit.com.ua/index.php?page=itemdetails&gtype=2&type=1&id=81 (дата звернення 21.09.2023)

37. ТЕСТУВАННЯ БЕЗПЕКИ: SQL-ІН'ЄКЦІЇ // Онлайн-курси від компанії ATestLab https://training.qatestlab.com/blog/technical-articles/security-testing-sql-injection/ (дата звернення 21.09.2023)

38. ТЕСТУВАННЯ БЕЗПЕКИ: XSS-ІН'ЄКЦІЇ. 16.06.2020. https://training.qatestlab.com/blog/technical-articles/security-testing-xss-injection

39. Ткачук Т. Ю. Забезпечення інформаційної безпеки: досвід окремих країн східної Європи. Інформація і право. 2017. № 4 (23). С. 62–72.

40. Ткачук Т. Ю. Сучасні загрози інформаційній безпеці держави: теоретико-правовий аналіз. Інформаційне право. 2017. № 10. С. 182–186.

41. Чунарьова А., Чунарьов А. Система управління інформаційною безпекою на базі міжнародних стандартів серії ISO. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2012. № 2. С. 48–52.

42. Шатун В.Т., Гладун О.В. Інформаційна безпека – невід'ємна складова національної безпеки України. Наукові праці. Державне управління. 2016. Вип. 255, Т. 267. С. 174–180.

43. Шпінталь М.Я., Орловський Н. М. Методи захисту робочих станцій від

DDoS-атак. ACIT’2014. Тернопіль, 2014. С. 230–231.

44. Що таке SQL та Бази даних? / aCode. https://acode.com.ua/sql-intro/ (дата звернення 07.10.2023)

45. Factor Analysis of Information Risk (FAIR). URL: http://www.riskmanagementinsight.com/.(дата звернення 05.11.2023)

46. Feinstein L., Schnackenberg D., Balupari R., Kindred D. Statistical Approaches to DDoS Attack Detection and Response. DARPA Information Survivability Conference and Exposition. 2003. № 1. С. 303-314.

47. Injections – The Many Faced Threat. URL: https://www.code-intelligence.com/blog/types-of-injection-vulnerabilities (дата звернення 11.11.2023)

48. ISACA RAM. URL: https://www.isaca.org/Pages/default. aspx?cid=1002083&Appeal=SEM&gclid=CMD70-aXs8oCFcLVcgoduzI Amg/ (дата звернення 14.11.2023)

49. NIST SpecialPublication 800-30 Risk Management Guide for Information Technology Systems. URL: http://www.nist.gov. (дата звернення 15.11.2023)

50. SQL-ін’єкції / aCode. https://acode.com.ua/sql-injection/ (дата звернення 16.11.2023)

ДОДАТОК А ПЕРЕЛІК НАУКОВИХ ПРАЦЬ

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2023»

17-18 листопада 2023

Хмельницький 2023

УДК 004:37:001:62

Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький. 2023. 345с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: *apkt.khnu@gmail.com*

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2023

XV Всеукраїнська науково-практична конференція

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

Робочі мови конференції: українська, англійська

ОРГКОМІТЕТ:

Олег СИНЮК – голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор

Олег САВЕНКО – заступник голови оргкомітету, декан факультету Інформаційних технологій ХНУ, доктор технічних наук, професор

Олександр БАРМАК – заступник голови оргкомітету, завідувач кафедри Комп'ютерних наук ХНУ, доктор технічних наук, професор

Тетяна ГОВОРУЩЕНКО – завідувач кафедри Комп'ютерної інженерії та інформаційних систем ХНУ, доктор технічних наук, професор

Олена ВИСОЦЬКА – доктор технічних наук, завідувач кафедри Радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», професор

Євгеній ЛАВРОВ – доктор технічних наук, професор (Сумський державний університет)

Людмила ТІМОФЄЄВА – відповідальна за студентську науково-дослідну роботу ХНУ

Олександр МАЗУРЕЦЬ – секретар конференції, к.т.н., доцент кафедри Комп'ютерних наук ХНУ

Марина МОЛЧАНОВА – секретар конференції, викладач кафедри Комп'ютерних наук ХНУ

КОНТАКТНА ІНФОРМАЦІЯ:

e-mail для листування: *apkt.khnu@gmail.com*

УДК 004.056

Кльоц Ю.П., Петляк Н.С., Чвалов А.А.

Хмельницький національний університет

ТЕХНОЛОГІЇ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБРЕСУРСІВ

Кожен веб-ресурс має вразливості. Наявність вразливостей пояснюється здатністю людей допускати помилки, оскільки великі веб-ресурси пише не одна людина, а ціла група. І досить часто вразливості виникають через недостатню перевірку коду, некоректне налаштування серверу та використання неперевірних бібліотек і фреймворків від сторонніх розробників. Однак, для запобігання вразливостям та забезпечення безпеки веб-ресурсів, необхідно впроваджувати технології тестування. В даній роботі проведено порівняння найпоширеніших технологій виявлення вразливостей.

Almost any web resource has vulnerabilities. The presence of vulnerabilities is easily explained by the ability of people to make mistakes, because large web resources are written not by one person, but by a whole group. And quite often vulnerabilities occur due to insufficient code verification, incorrect configuration of the server and the use of untested libraries and frameworks from third-party developers. However, to prevent vulnerabilities and ensure the security of web resources, it is necessary to implement testing techniques. This paper compares the most common vulnerability detection techniques.

Захист веб-ресурсів надзвичайно важливий, особливо в контексті стрімкого розвитку і збільшення залежності сучасного суспільства від технологій та інтернету.

Додатково, важливо враховувати, що високий рівень доступності інтернету і швидкий розвиток технологій призводять до збільшення кількості зловмисників із спеціалізованими навичками, які шукають слабкі місця у веб-ресурсах для незаконного доступу і зловживання. Тому, для забезпечення безпеки веб-ресурсів, необхідно активно застосовувати передові практики та технології, які дозволяють вчасно виявляти і усувати вразливості.

Ринок безпеки також стрімко набирає оберти і ця динаміка особливо помітна з ростом вразливостей і ризиком їх експлуатування зловмисниками. Цей тренд особливо помітний з результатів дослідження Grand View Research [1], що зображений на рисунку 1.

Метою роботи є розгляд найпоширеніших технологій використовуваних на практиці, а саме DAST (Динамічне тестування безпеки додатків)[2] що дозволяє виявити вразливості в реальних умовах експлуатації та SAST (Статичне тестування безпеки додатків)[3] яке спрямоване на виявлення вразливостей на етапі розробки, шляхом аналізу вихідного коду.

DAST – технологія тестування чорної скриньки, яка динамічно перевіряє роботу веб-ресурсу без знання внутрішньої будови та схеми роботи. Даний підхід

використовується для виявлення вразливостей на пізніх стадіях розробки, якими може скористатися зловмисник. Це досягається за допомогою методів введення помилок, а саме передачі шкідливих даних та інструкцій, з метою виявлення поширених вразливостей безпеки, таких як SQL-ін'єкції, міжсайтові сценарії (XSS) та інші, що перераховані в OWASP Top 10[4].

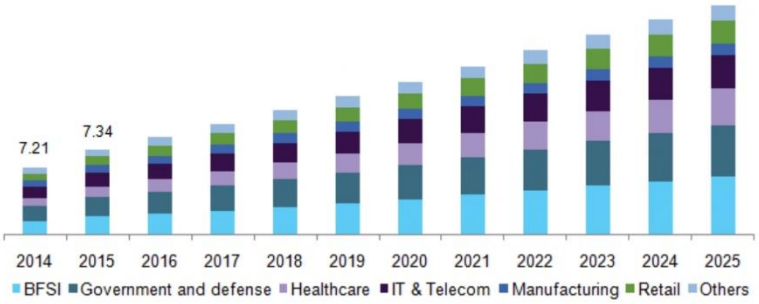

Рисунок 1 – Ринок безпеки додатків у США

SAST –технологія тестування білої скриньки яка існує вже більше десяти років. SAST дозволяє розробникам виявляти вразливості на ранніх стадіях розробки, оскільки в цьому підході аналізується внутрішня структура, будова та вихідний код. Вагомою перевагою є й те що статичний аналізатор точно вказує на вразливий фрагмент в коді та надає рекомендації щодо його усунення.

Обидві технології тестування для виявлення вразливостей та недоліків, використовуються по-різному, але найбільш ефективною практикою є їх використовувати в тандемі. Такий комплексний підхід дозволяє покрити усі аспекти безпеки веб-ресурсів та забезпечити більш повний захист. В таблиці 1 наведені ключові відмінності між DAST і SAST технологіями [5].

Таблиця 1 – Ключові відмінності між DAST і SAST

DAST	SAST
Потрібен працюючий веб-ресурс	Потрібен вихідний код
Підтримує будь яку мову програмування для створення веб-ресурсів	Залежить від використовуваної мови програмування та технологій
Немає можливості точно визначити джерело проблеми	Точно визначає вразливий фрагмент у вихідному коді
Може виявляти проблеми, пов'язані з часом виконання, потоками та середовищем	Не виявляє проблеми що пов'язані з часом виконання, потоками та середовищем
Вразливості можна виявити після завершення циклу розробки	Сканування можна виконати, як тільки код буде вважатися завершеним
Дорогий спосіб усунення недоліків, так як застосовується в кінцевому циклу розробки	Дешевший спосіб усунення недоліків, оскільки виявляються на ранніх стадіях
Не вимагає активної участі розробників у процесі тестування, по закінченню надається звіт з рекомендації для усунення вразливостей	Вимагає участі розробників, оскільки вони повинні аналізувати знайдені вразливості та усувати їх безпосередньо в вихідному коді

Використання обох технологій протягом усього процесу розробки веб-ресурсу, та продовження моніторингу після його випуску – це те, що в кінцевому підсумку забезпечує безпеку веб-ресурсу та бізнесу.

У сучасному цифровому світі, де безпека веб-ресурсів стає дедалі важливішою, використання двох ключових технологій SAST та DAST є надзвичайно важливими. Обидві технології мають свої унікальні переваги і доповнюють одна одну, створюючи комплексну техногоію тестування безпеки, та допомагають виявляти і виправляти вразливості на усіх етапах розробки.

Крім того, важливо пам'ятати про постійний моніторинг безпеки веб-ресурсу, навіть після його випуску. Така практика забезпечить тривалий та ефективний захист, що допоможе уникнути потенційний загроз для бізнесу та користувачів.

Перелік посилань

1. Application Security Market Size, Share | Industry Trends Report, 2025. URL: https://www.grandviewresearch.com/industry-analysis/application-security-market.
2. Static application security testing. URL: https://en.wikipedia.org/wiki/Static_application_security_testing.
3. Dynamic application security testing. URL: https://en.wikipedia.org/wiki/Dynamic_application_security_testing.
4. OWASP Top 10:2021. URL: https://owasp.org/Top10/.
5. SAST vs. DAST | Lightrun. URL: https://lightrun.com/sast-vs-dast/.

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК 2023

ЗБІРНИК НАУКОВИХ ПРАЦЬ

Комп'ютерна верстка: **Олександр МАЗУРЕЦЬ,**
Марина МОЛЧАНОВА

Підписано до друку 16.11.2023.
Версія друку «APKN2023_CorpusPaper v7mod1 Finita».

E-mail: *apkt.khnu@gmail.com*
ХНУ. м. Хмельницький, вул. Інститутська,11.

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Чвалова Андрія Анатолійовича
ПІБ здобувача вищої освіти
студента ФІТ, 2 курсу, групи КБм-22-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5.12.2023
дата


підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 7%**

	Документ		Сумарний збіг по Базі Даних	
ID: 123035 Назва: Метод та система підтримки перевірки веб-додатків на вразливості Додано в БД: 2023-12-13 Автора: Чвалов А.А. Керівники: Кльоц Ю.П. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	109638	913	6441 (6%)	75 (8%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016018388

Дата перевірки:
18.12.2023 16:49:31 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
18.12.2023 16:53:08 EET

ID користувача:
100008300

Назва документа: **Чвалов_18.12.2023_00.58**

Кількість сторінок: **87** Кількість слів: **16934** Кількість символів: **128230** Розмір файлу: **981.71 KB** ID файлу: **1015705861**

7.7%
Схожість

Найбільша схожість: **3.15%** з Інтернет-джерелом (https://ir.library.knu.ua/server/api/core/bitstreams/02d5a3b4-6078-4b..

7.45% Джерела з Інтернету	788	Сторінка 89
0.69% Джерела з Бібліотеки	72	Сторінка 95

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

34.8%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за **8 слів** та **0%**)

Немає вилучених Інтернет-джерел

34.8% Вилученого тексту з Бібліотеки	1	Сторінка 95

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	21

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:
Назва: Метод та система підтримки перевірки веб-додатків на вразливості

Автор: Чвалов Андрій Анатолійович
Спеціальність: 125 – Кібербезпека
Освітня програма: освітньо-професійна
Науковий керівник: Кльоц Юрій Павлович, к.т.н, доцент
Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 7,7%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 3%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи _____ Ю.П.Кльоц

Гарант ОП _____ В.Ю. Тітова

Завідувач кафедри кібербезпеки _____ Ю.П. Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

освітньо-кваліфікаційного рівня «магістр»

Магістр Чвалов Андрій Анатолійович
Тема: Метод та система підтримки перевірки веб-додатків на вразливості
Галузь знань 12 Інформаційні технології Спеціальність 125 Кібербезпека денної форми навчання

Обсяг дипломної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень ___; кількість сторінок записки 92;

1. Короткий зміст ДР та прийнятих рішень В рамках роботи проведено дослідження з розробки системи підтримки перевірки веб-додатків на вразливості. В роботі поставлено і вирішено задачі: дослідити технології виявлення вразливостей у веб-додатках; оглянути методи побудови систем підтримки прийняття рішень; розробити методику підтримки перевірки веб-додатків на вразливості; виконати розробку вимог до програмної реалізації системи підтримки перевірки веб-додатків на вразливості

2. Висновок про відповідність МР завданню Магістерська робота у достатній мірі відповідає поставленому завданню як у теоретичній і практичній частині роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовується актуальність теми дослідження; її зв'язок із науковими програмами, планами, темами та формулюється мета і основні завдання дослідження. У першому розділі було проведено аналіз видів вразливостей веб-додатків та їх наслідки, підходи до виявлення вразливостей та методи, системи підтримки користувачів та постановка задач дослідження. У другому розділі описано призначення систем підтримки прийняття рішень та особливості їх використання, аналіз структури та засобів формування системи підтримки прийняття рішень, розробку алгоритмів прийняття рішень на основі чек-листів. У третьому розділі проведено опис алгоритму синтезу чек-ліста для перевірки на вразливості, методи оцінки повноти чек-ліста, аналіз обчислювальної складності методів. У четвертому розділі роботи практичне застосування методів перевірки веб-додатків на вразливість, обґрунтування та вибір середовища реалізації, алгоритм роботи системи підтримки перевірки веб-додатків на вразливості, програмні характеристики моделювання сппр, аналіз ефективності модулю прийняття рішень системи підтримки перевірки веб-додатків на вразливості

4. Позитивні сторони проекту полягають в розширенні знань про використання системи підтримки перевірки веб-додатків на вразливості

5. Негативні сторони проекту : У роботі недостатньо приділено увагу формулюванню визначень понятійного апарату дослідницької роботи, математичної та експериментальних моделей. Робота носить більше аналітичний характер і не має чітко визначених практичних результатів

6. Оцінка графічного оформлення та пояснювальної записки роботи. Графічне оформлення виконане відповідно до теми дипломної роботи із дотриманням усіх стандартів. У загальному графічне оформлення виконане на достатньому технічному рівні. Пояснювальна записка відповідає нормам для її оформлення та вимогам .

7. Відгук про роботу в цілому В загальному дипломна робота заслуговує позитивної оцінки.

8. Інші зауваження

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що дипломна робота заслуговує позитивної оцінки

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, доктор технічних наук, професор Мартинюк Валерій Володимирович.

« 20 » грудня 2023 . (підпис)