DOI: https://doi.org/10.36910/6775-2524-0560-2025-58-08
УДК 004.94:518.5
Добришин Юрій Євгенович, к.т.н., доцент
https://orcid.org/0000-0003-2473-9507
Національна академія Служби безпеки України, м. Київ, Україна

КЛАСИФІКАЦІЯ ТА КОДУВАННЯ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВНАСЛІДОК ДІЇ КІБЕРАТАК

Добришин Ю.Є. Класифікація та кодування дефектів програмного забезпечення внаслідок дії кібератак. В роботі на підставі проведеного аналізу впливу кібератак на стан працездатності автоматизованих систем та комплексів, пропонується методика класифікації та кодування дефектів пошкодженого програмного забезпечення. Методика класифікації та кодування дефектів програмного забезпечення розроблена на підставі певних характеристик та класифікаційних ознак найбільш розповсюджених кібератак та дефектів програмного забезпечення, що виникають внаслідок їх дії. В основу класифікації покладені існуючі зв'язки між пошкодженим програмним забезпеченням та можливими дефектами, що виникають внаслідок дії кібератак, а також між дефектами програмного забезпечення та способами їх виявлення та відновлення. На підставі аналізу класифікаційних ознак розроблена структура технологічного коду дефекту пошкодженого програмного забезпечення, елементами якого є певні класифікаційні ознаки. На прикладі одного з виду кібератак SQL-ін'єкція приведений практичний приклад класифікації та призначення кодів для дефектів пошкодженого програмного забезпечення, а також варіанти можливих комбінацій кодів дефектів, що можуть виникати під час дії кібератаки виду SQL-ін'єкція. З метою подальшої формалізації відновлення дефектів пошкодженого програмного забезпечення, приведений ієрархічний взаємозв'язок рівній теоретичного дереву стану атаки виду SQL-ін'єкція, що відповідає порядку класифікації та кодування. Дослідження, що були проведені, дозволяють застосовувати методику класифікації та кодування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак для автоматизованого проектування технологічних процесів діагностики дефектів пошкодженого програмного забезпечення, а також визначення доцільності відновлення програмних модулів та рішення інших технологічних питань щодо захисту інформації.

Ключові слова: дефект пошкодженого програмного забезпечення, класифікаційні ознаки, діагностування, технологічний код, кібератака, класифікатор дефектів.

Dobryshyn Yu. Classification and coding of software defects resulting from cyberattacks. Based on the analysis of the impact of cyberattacks on the state of operability of automated systems and complexes, the article proposes a methodology for classifying and coding defects of damaged software. The methodology for classifying and coding software defects was developed based on certain characteristics and classification features of the most common cyberattacks and software defects resulting from their effects. The classification is based on existing relationships between damaged software and possible defects resulting from cyberattacks, as well as between software defects and methods for their detection and recovery. Based on the analysis of classification features, the structure of the technological code of a damaged software defect was developed, the elements of which are certain classification features. Using the example of one of the types of cyberattacks, SQL-injection, a practical example of classification and assignment of codes for defects of damaged software is given, as well as options for possible combinations of defect codes that may arise during the action of a cyberattack of the type SQL-injection. In order to further formalize the repair of damaged software defects, a hierarchical relationship equal to the theoretical state tree of the SQL injection attack type is given, which corresponds to the order of classification and coding. The studies that have been conducted allow applying the methodology for classifying and coding damaged software defects due to cyberattacks for automated design of technological processes for diagnosing damaged software defects, as well as determining the feasibility of restoring software modules and solving other technological issues related to information protection**.**

Keywords: damaged software defect, classification features, diagnostics, technological code, cyberattack, defect classifier.

Постановка наукової проблеми. Аналіз пошкодженого програмного забезпечення в наслідок дії різного роду кібератак свідчить про те, що процес відновлення працездатності автоматизованих систем та програмних комплексів складно формалізувати, враховуючи особливість, неповноту та суперечливості інформації, що з'являється після навмисних дій, які здійснюються за допомогою засобів електронних комунікацій. Основу забезпечення працездатності програмного забезпечення складають процеси діагностування, на базі яких у подальшому будуються технологічні операції з відновлення дефектів програмного забезпечення.

Технологія діагностування передбачає виявлення дефектів програмного забезпечення, які з'являються після дії різного роду кібератак, унаслідок наявних вразливостей у складі програмних модулів та компонентів автоматизованих систем. Тому задачі щодо виявлення, оцінки та класифікації дефектів програмного забезпечення, а також їх подальшого групування в технологічно - подібні групи відрізняється значною трудомісткістю. Окрім того складно забезпечити автоматизацію процесу відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, тому що на теперішній час відсутні формалізовані методики опису внутрішнього змісту та взаємозв'язків окремих технологічних операцій з відновлення дефектів програмного забезпечення.

Під час виконання технологічних операцій з відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, крім виявлення, діагностування та аналізу дефектів, важливе місце займають операції класифікації, кодування та групування дефектів програмного забезпечення в технологічно - подібні групи.

Тобто задача передбачає виконання операцій розпізнавання конкретного дефекту, а саме, виявлення набору ознак для подальшої їх класифікації та групування з метою накопичування технологічно-подібних дефектів за ознакою оптимального способу відновлення. Розв'язання вище описаної задачі потребує проведення технологічних операцій щодо аналізу дефектів програмного забезпечення, яке відновлюється внаслідок дії кібератак, а також здійснення класифікації дефектів, виконання кодування ознак дефектів та розробки так званого комплексного дефекту.

Дефект, для якого необхідно розробити технологічний процес, повинен бути представлений у виді набору ознак, які мають відповідний код. За результатами чого можливо автоматизувати процес відновлення, який передбачає віднесення дефекту до певного образу, для якого вже існує блок - схема відновлення.

Технологічний процес відновлення, який складається з блок-схеми, являє собою сукупність образів та можливих рішень. Під час автоматизації, опис дефекту відноситься до певного образу та підставі можливих варіантів для нього призначаються певні рішення стосовно відновлення. Причому для кожного варіанту необхідно придати ступінь важливості, для того щоб з множини рішень обрати оптимальний. Такий підхід передбачає проведення робіт з класифікації, кодування ознак дефектів пошкодженого програмного забезпечення та розробку, на базі вказаного, комплексної моделі дефекту з метою її використання для вибору технологічного процесу відновлення програмного забезпечення внаслідок дії кібератак.

Аналіз досліджень. Задачею технологічного процесу відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, є визначення відповідного дефекту, тому рішення вказаної задачі базується на встановленні відповідності між різними ознаками, що характеризують в цілому дефект, та технологічними методами їх виявлення та усунення.

Різноманіттю дефектів пошкодженого програмного забезпечення, особистому досвіду та інтуїції співробітника, що виконує оцінку стану працездатності програмного забезпечення за допомогою обчислювальних засобів, протиставляється кінцеве число типів дефектів, ознак та технологічних рішень. Необхідно визначити основні типи дефектів, здійснити їх аналіз, та підставі чого розробити класифікатори, що інтерпретують відомості про дефекти з метою автоматизованого проектування процесів відновлення пошкодженого програмного забезпечення. Необхідною вимогою також є виконання досліджень щодо формалізації ознак та понять, що відносяться до опису дефектів пошкодженого програмного забезпечення. Для вирішення зазначених завдань, необхідно здійснити класифікацію дефектів пошкодженого програмного забезпечення та на її основі розробити масиви технологічних кодів дефектів.

Існуючі на теперішній час системи класифікації дефектів пошкодженого програмного забезпечення недостатньо задовольняють можливість їх використання для автоматизації, тому потребують доопрацювання.

Аналіз наукових робіт свідчить, що до цього часу ще не розроблена універсальна система класифікації дефектів пошкодженого програмного забезпечення, державні установи та компанії під час роботи застосовують свої власні методи класифікації шляхом розбиття дефектів пошкодженого програмного забезпечення на різні класи з подальшим групуванням.

Одними з найперших робіт, де були розроблені та описані наукові основи класифікації дефектів та помилок програмного забезпечення, були роботи закордонних та вітчизняних наукових авторів Тайера Т, Липоу М, Марека Л, Девейна Е., Шингера Н.

В зазначених роботах класифікація дефектів програмного забезпечення базувалися на підставі основних видів діяльності, пов'язаних з життєвим циклом розробки автоматизованих систем та програмних модулів. Класифікація, кодування та прогнозування дефектів програмного забезпечення залишається актуальною і на теперішній час.

Так, декілька вітчизняних авторів [1] вважають, що дефекти програмного забезпечення необхідно класифікувати на підставі їх функціональності, наприклад, функціональні та нефункціональні. Враховуючи таку класифікацію, автори пропонують, що під час проведення діагностування дефектів програмного забезпечення, необхідно у якості основної ознаки класифікації обирати дефекти, які негативно впливають на атрибути якості програмного

забезпечення. Науковці вважають, що нефункціональний дефект може прогресувати, тому здійснення заходів з виявленням ознак нефункціональних дефектів на фазі проектування та супроводу автоматизованих систем та комплексів, є головним підходом щодо класифікації дефектів програмного забезпечення**.**

Дослідження класифікації дефектів програмного забезпечення приведено також у наукові роботі [2]. Авторами запропонований метод автоматичної класифікації дефектів з метою їх швидкого знаходження та відновлення. Наукові співробітники стверджують, що на теперішній час не існує автоматичної класифікації дефектів програмного забезпечення, тому розглядають та пропонують відповідні концепції щодо аналізу дефектів та їх управлінню, а також класифікацію дефектів програмного забезпечення, яка значно підвищить ефективність тестування програмного забезпечення та знизить вартість його обслуговування.

Декілька робіт, присвячених питанням класифікації дефектів програмного забезпечення, передбачають застосування окремих математичних моделей та методів машинного навчання. Так у наукової роботі [3] наведена методика побудови регресійної моделі щодо прогнозування появи дефектів програмного забезпечення, що відображає математичний зв'язок між щільністю дефектів та їх кількістю. Підходи, запропоновані авторами, дозволяють покращити час прогнозування дефектів та якість їх класифікації.

Інтерес представляє робота вітчизняного автора [4]. У науковій роботі приведені математичні моделі вразливостей програмного забезпечення, а також критерії класифікації дефектів програмного забезпечення, які дозволяють сформулювати правила їх класифікації. Автором також запропонований метод та механізми прогнозування появи дефектів, які можуть бути застосовані для прогнозування збоїв і вразливостей як прикладного, так і системного програмного забезпечення.

Продовженням робіт з питань аналізу дефектів програмного забезпечення та їх класифікації, є наукова робота фахівців [5]. У роботі дефекти програмного забезпечення класифікуються в залежності від ступеня їх впливу на працездатність програмного забезпечення. Дефекти програмного забезпечення класифікуються та обираються з використанням певного алгоритму, який обирається на підставі аналізу показників якості програмного забезпечення.

Авторами наукової роботи [6] здійснюється оцінка узгодженості між ознаками дефектів програмного забезпечення на підставі дослідження певних класифікаторів. У роботі ознаки дефектів розподіляються на специфічні, які належать до одного класифікатора та агностичні для іншого класифікатора. Набори даних про дефекти програмного забезпечення описуються функціями, які впливають на процес їх віднесення до певних класифікаторів.

У науковій роботі [7] приведені дослідження класифікації дефектів програмного забезпечення з використання штучного інтелекту. Під час діагностування, авторами спочатку виконується класифікація дефектів програмного забезпечення із застосуванням компонентів експертної системи, яка має певну структуру, що побудована на базі програмного забезпечення MS SQL Server та масиви дефектів програмного забезпечення. Класифікація дефектів програмного забезпечення здійснюється в автоматичному режимі шляхом визначення категорії дефектів на підставі типових ознак та логічного висновку виходячи з характеристик певного типу дефекту програмного забезпечення. За результатами діагностики дефектів програмного забезпечення робиться певний висновок щодо їх подальшої класифікації.

За висновками авторів, експериментальні результати показують, що запропонована система діагностування та класифікації дефектів програмного забезпечення з використанням штучного інтелекту суттєво перевершує точність традиційного механізму діагностування, що здійснюється без використання засобів обчислювальної техніці. Метод класифікації, який використовується у науковій роботі, на першому етапі визначає категорію дефектів програмного забезпечення за допомогою типових ознак, а далі за рахунок зменшення бази знань експертної системи, вибирає назву дефекту з числа характеристик, які належать до певного виду дефекту. Інтерес в роботі представляє база знань експертної системи. Окрім допоміжних таблиць, існує база дефектів, яка включає дві таблиці: таблицю дефектів типу керування та потоку даних та структуровану та нефункціональну таблицю дефектів.

Метод класифікації та групування дефектів програмного забезпечення, який використовує концепцію нечіткої логіки представлений у роботі [8]. Цей підхід дозволяє отримати складність, неповноту або невизначеність, що часто відображається в процесі аналізу дефектів. Метод наведений у науковій праці може бути корисним під час експлуатації складних автоматизованих

систем та комплексів. На думку авторів дефекти програмного забезпечення можуть представлятися в різних контекстах та у відмінності від класичних методів класифікації дефектів, де кожен дефект чітко потрапляє в одну з категорій, наприклад, важливий, неважливий або критичний, дефект програмного забезпечення може частково відноситися до кількох категорій (наприклад, може бути одночасно важливим і критичним, залежно від контексту). Такий методичний прийом класифікації дозволяє врахувати різні характеристики дефектів програмного забезпечення та забезпечує ітераційне їх групування з подальшим вибором оптимального способу відновлення.

У науковій роботі [9] наведені окремі методичні підходи щодо класифікації дефектів, за допомогою застосування технологічних операцій з прогнозування програмного забезпечення. Автори стверджують, що для дефектів програмного забезпечення найбільш застосовуваними є методи класифікації, які використовують елементи логістичної регресії та випадкового лісу. Підходи щодо прогнозування дефектів програмного забезпечення базуються на підставі бінарної класифікації, яка класифікує модулі програмного забезпечення на дефектний та недефектний. На думку авторів, класифікація дефектів програмного забезпечення є важливим елементом виявлення дефектів, тому в роботі визначені основні напрямки досліджень щодо методів вибору та удосконалення класифікаторів, як важливого елементу моделі виявлення дефектів.

У науковій статті [10] автори стверджують, що класифікація дефектів програмного забезпечення є головним фактором у забезпеченні його якості. Окрім цього процес класифікації дефектів, спрямований на прогнозування появи дефектів відіграє вирішальну роль під час призначення способів їх відновлення. Для прогнозування та класифікації дефектів автори пропонують застосування методів машинного навчання, на базі якого здійснюється аналіз виявлення закономірностей і взаємозв'язків між різними факторами та виникненням дефектів програмного забезпечення. Після проведення діагностування класифікація дефектів програмного забезпечення здійснюється на основі виявлених характеристик, характеру виникнення та критичності.

Колектив авторів [11] зазначає, що дефекти програмного забезпечення необхідно класифікувати відповідно до вимог та фаз розробки життєвого циклу програмного забезпечення. Цей підхід зосереджується на дефектах, що допущені на етапі розробки модулів та компонентів програм. За результатами робіт, автори представляють методику класифікації дефектів програмного забезпечення із урахуванням помилок, що призводять до збоїв його працездатності. Ознаками класифікації дефектів виступають подібності та походження помилок за результатами розробки програмного забезпечення.

Необхідно зазначити, що аналіз літературних джерел, приведених наукових підходів, свідчить про актуальність проблеми класифікації та кодування дефектів під час розробки та супроводу програмного забезпечення. Але існуючі методики не можуть бути у повному обсязі застосовані до класифікації дефектів пошкодженого програмного забезпечення в умовах неповноти інформації, яка існує під час здійснення кібератак. Крім того технологія діагностування та відновлення пошкодженого програмного забезпечення суттєво відрізняється від традиційної технології, яка застосовується під час розробки та оновлення програмного забезпечення, тому що має різний склад та послідовність технологічних операцій. Проблема класифікації дефектів пошкодженого програмного забезпечення залишається актуальною.

Мета роботи. Метою статті є вирішення проблеми щодо класифікації та кодування дефектів пошкодженого програмного забезпечення з метою розробки комплексної моделі дефекту та використання пропонованої моделі для вибору технологічного процесу відновлення пошкодженого програмного забезпечення внаслідок дії кібератак.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Працездатність пошкодженого програмного забезпечення, що потребує відновлення, характеризується кортежем дефектів, які необхідно усунути.

Припустимо, що існує пошкоджене програмне забезпечення $\{P_r\}$, технічний стан якого визначається сукупністю дефектів $\{d_{1\ldots}d_n\}$, і кожний дефект пошкодженого програмного забезпечення характеризується набором відповідних ознак $\{W_i\}$. Виходячи з цього можна записати:

$$d_1(W_{11}), d_1(W_{21}), \ldots d_1(W_{k1}),$$
$$d_2(W_{12}), d_1(W_{22}), \ldots d_1(W_{k2}), \quad (1)$$

$$d_n(W_{1n}), d_1(W_{2n})_{,} \dots d_1(W_{kn})$$

Пошкоджене програмне забезпечення, що потребує відновленню, має один або декілька дефектів; одному та тому же дефекту відповідає один або декілька ознак.

Сукупність всіх можливих дефектів, які доповнюються певними ознаками можуть сформувати комплексний образ пошкодженого програмного забезпечення, яке описується за допомогою наступного виразу:

$$d_1(W_{11}, W_{21}, W_{k1}) \dots \dots d_n\left(W_{1n,}W_{2n,}W_{kn}\right) \quad (2)$$

Задача складається в тому, щоб здійснити розпізнавання певного пошкодженого програмного забезпечення серед теоретичні можливих комбінацій кодів комплексного образу пошкодженого програмного забезпечення з подальшим віднесенням його до певної групи, для якої вже існує відповідний технологічний процес відновлення.

Таким чином для реалізації методики класифікації дефектів під час відновлення пошкодженого програмного забезпечення, необхідно виконати:

- визначити номенклатуру пошкодженого програмного забезпечення;
- здійснити аналіз дефектів програмного забезпечення та описати їх відповідні ознаки щодо номенклатури пошкодженого програмного забезпечення;
- розробити та виконати кодування ознак дефектів пошкодженого програмного забезпечення;
- розробити комплексний образ пошкодженого програмного забезпечення, який можна описати за допомогою комбінацій кодів дефектів та їх ознак.

У даної статті пропонується класифікатор дефектів пошкодженого програмного забезпечення, який розроблений на підставі певних класифікаційних ознак. В основу класифікації покладені існуючі зв'язки між класифікаційними характеристиками кібератак, пошкодженим програмним забезпеченням та можливими дефектами, що виникають під час дії кібератак, а також між дефектами програмного забезпечення та способами їх виявлення та відновлення. Для кожного об'єкту, що приймає участь у класифікації, призначається відповідний код.

Виявлення дефектів пошкодженого програмного забезпечення під час виконання технологічних операцій діагностування потребує знання можливих класів кібератак, які можуть здійснюватися під час експлуатації програмного забезпечення. Тому першим напрямком класифікації є визначення джерела виникнення дефекту в залежності від класу кібератак $\{K_{ka}\}$, що застосовуються зловмисниками для своїх корисних дій (табл. 1).

Таблиця 1. Класи кібератак

№ з.п	Клас кібератаки	Код
1	Зовнішні	11
2	Внутрішні	12
3	Технічні	13

Але клас кібератаки не в повної мірі відображає повноту інформації, що необхідна для проведення діагностування пошкодженого програмного забезпечення та подальшого вибору технологічного маршруту відновлення дефектів, тому іншим напрямком класифікації є зазначення виду кібератаки, яка пов'язана з основними компонентами інфраструктури автоматизованих систем $\{I_s\}$ (табл.2).

Таблиця 2. Види кібератак

№ з.п	Вид кібератаки	Код
1	Фішинг (Phishing)	21
2	Рансомвар (Ransomware)	22
3	DDoS-атака (Distributed Denial of Service)	23
4	SQL-ін'єкція (SQL Injection)	24
5	Людина посередині (Man-in-the-Middle)	25
6	Крадіжка сесії (Session Hijacking)	26
7	Зловмисне програмне забезпечення (Malware)	27
8	Атаки на вразливості (Zero-Day Attacks)	28
9	Програми-шпигуни (Spyware)	29

10	Атака дистанційного доступу до ресурсів (Remote Access Trojans, RATs)	210
11	Кросс-сайт скриптинг (Cross-Site Scripting, XSS)	211
12	Соціальна інженерія (Social Engineering)	212
13	Скіммінг (Skimming)	213

Кібератаки можуть мати різноманітні причини або мотиви, що лежать в основі їхнього проведення, тому іншим напрямком класифікації буде визначення причини $\{Res_{ka}\}$, чому вказаний вид кібератаки застосовується зловмисником (табл.3)

Таблиця 3. Причини появи видів кібератак

№ з.п	Причина появи кібератаки	Код
1	Фінансова вигода	31
2	Політичні та ідеологічні мотиви	32
3	Шпигунство	33
4	Конкурентні переваги	34
5	Помста або особисті мотиви	35
6	Військові та терористичні мотиви	36
7	Тестування	37
8	Технічні помилки	38
9	Атаки, пов'язані з соціальними мережами	39
10	Хакерство заради розваги	310
11	Вразливість системи	311

Важливим напрямком класифікації є розподіл дефектів у залежності від цілей, що використовують зловмисники $\{M_{ka}\}$. До таких дефектів необхідно віднести злам системи безпеки, виток конфіденційної інформації, неавторизований доступ та зміна налаштування системи, відмова в обслуговуванні, системні збої або аварії, порушення системи моніторингу**,** порушення роботи з базами даних. Наприклад, кібератака може здійснюватися на апаратні компоненти серверів, пристроїв введення/виведення, систем зберігання даних, мережеві пристрої тощо (табл.4)

Таблиця 4. Ціль кібератак

№ з.п	Ціль кібератаки	Код
1	Викрадення особистих даних	41
2	Доступу до облікових записів	42
3	Вимагання грошей за повернення доступу до даних	43
4	Перевантаження системи з метою її виведення з ладу або тимчасового припинення роботи	44
5	Модифікація даних у базах даних	45
6	Викрадення даних у базах даних	46
7	Викрадення конфіденційної інформації	47
8	Шкідливе втручання в роботу системи	48
9	Використання вразливостей для отримання доступу до системи або даних	49
10	Шпигунство за користувачем	410
11	Віддалене керування комп'ютером жертви для крадіжки даних або інших зловмисних цілей	411
12	Викрадення платіжної інформації	412
13	Маніпуляція даними	413

Наступним напрямком класифікації є розподіл дефектів у залежності до типу програмного забезпечення $\{P_r\}$, наприклад, належність до системного програмного забезпечення, прикладного програмного забезпечення, а також до інструментального програмного забезпечення (табл.5).

Таблиця 5. Типи програмного забезпечення

№ з.п	Тип програмного забезпечення	Код
1	Системне програмне забезпечення	51
2	Прикладне програмне забезпечення	52

3	Інструментальне програмне забезпечення	53

На підставі аналізу пошкодженого програмного забезпечення були виявлені та проаналізовані основні типи дефектів $\{T_d\}$, що виникають під час дії кібератак. Перелік основних таких типів дефектів наведений у таблиці 6.

Таблиця 6. Основні типи дефекти пошкодженого програмного забезпечення внаслідок дії кібератак

№ з.п	Тип дефекту	Код
1	Неавторизований доступ	61
2	Зміна налаштування системи	62
3	Відмова в обслуговуванні	63
4	Нестабільність роботи системи	64
5	Збій у роботі системи	65
6	Проблеми з оновленнями системи	66
7	Порушення системи моніторингу	67
8	Порушення роботи компонентів комп'ютерної мережі	68
9	Збій системи безпеки	69
10	Некоректна обробка вхідних даних, що використовують запити	610

Для більш детальної характеристики типів дефектів та подальшого опису комплексного коду дефекту пошкодженого програмного забезпечення, здійснимо класифікацію підтипів $\{T_{pd}\}$ дефектів пошкодженого програмного забезпечення (табл. 7).

Таблиця 7. Основні підтипи дефектів пошкодженого програмного забезпечення внаслідок дії кібератак

№ з.п	Підтип дефекту	Код
1	Втрата контролю над системою	71
2	Вилучення критичних файлів	72
3	Зміна критичних файлів	73
4	Відсутність відповіді системи на запити	74
5	Втрата доступу до функцій системи	75
6	Системні збої у роботі програмного забезпечення	76
7	Системні аварії у роботі програмного забезпечення	77
8	Невідповідність логіки роботи програми	78
9	Витік оперативної пам'яті системи	79
10	Невірне використання процесорного часу	710
11	Порушення роботи з базами даних	711
12	Відсутність патчів безпеки	712
13	Некоректне оновлення програмного забезпечення	713
14	Зміна інформації у системних журналах	714
15	Видалення системних журналів	715
16	Порушення роботи мережевих протоколів	716
17	Порушення конфіденційності	717
18	Пошкоджена цільність даних	718
19	Порушення цільності програмного коду	719
20	Небезпека витоку даних	720
21	Виток даних	721
22	Пошкодження даних	722
23	Шифрування файлів для вимагання	723

Класифікація передбачає розподіл дефектів пошкодженого програмного забезпечення за критерієм важливості щодо їх подальшого відновлення $\{K_v\}$. Таки дефекти пропонується включати в групи: основні та неосновні. Під основними розуміють таки дефекти, які потребують обов'язкової

перевірки під час виконання технологічних операцій з діагностування та подальшого відновлення. Неосновними є такі дефекти, при яких пошкоджене програмне забезпечення може тимчасове бути експлуатуватися (табл.8).

Таблиця 8. Критерій важливості дефекту пошкодженого програмного забезпечення

№ з.п.	Критерій важливості дефекту	Код
1	Основний	81
2	Неосновний	82

Дефекти, що виникають в наслідок дії кібератак, можна класифікувати за значимістю $\{Z_{ka}\}$. Наприклад, є дефекти, які суттєво впливають на надійність роботи програмного забезпечення (пошкоджена цільність даних, порушення конфіденційності інформації, порушення цільності програмного коду), а є такі, які потребують усунення в другу чергу. Така класифікаційна ознака повинна враховуватися під час формування технологічного процесу, що визначає послідовність відновлення дефектів пошкодженого програмного забезпечення (табл.9).

Таблиця 9 Критерій значимості дефекту пошкодженого програмного забезпечення

№ з.п.	Критерій значимості дефекту	Код
1	Програмне забезпечення можна використовувати	91
2	Програмне забезпечення можна використовувати після переналагодження	92
3	Програмне забезпечення можна використовувати обмежено	93
4	Програмне забезпечення можна використовувати після відновлення	94
5	Програмне забезпечення потребує заміни	95

Під час проведення діагностування дефектів пошкодженого програмного забезпечення однієї з важливих ознак класифікації є розподіл дефектів у залежності від способу їх виявлення $\{S_{vka}\}$. Таки ознаки забезпечують оптимальне використання спеціалізованого програмного та технічного забезпечення, сприяють оптимальному застосуванню операцій щодо налаштування системи моніторингу та виявлення кібератак (табл.10).

Таблиця 10. Способи виявлення дефектів пошкодженого програмного забезпечення

№ з.п.	Спосіб виявлення дефектів	Код
1	Аналіз журналів безпеки	101
2	Перевірка цілісності файлів	102
3	Аналіз трафіку мережі	103
4	Аналіз вразливостей програмного забезпечення	104
5	Тестування на проникнення	105
6	Аналіз наявності шкідливих програм	106
7	Моніторинг поведінки користувачів	107

Таким чином на підставі аналізу класифікаційних ознак розроблена структура технологічного коду дефекту пошкодженого програмного забезпечення, елементами якого є вказані класифікаційні ознаки

$$K_d = \{K_{ka}, I_s, Res_{ka}, M_{ka}, P_r, T_d, T_{pd}, K_v, Z_{ka}, S_{vka}\} \quad (3)$$

Дослідження, що були проведені за методикою, що приведена, дозволяють виконувати кодування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак. Розроблений технологічний код необхідний для машинного проектування технологічних процесів діагностики дефектів пошкодженого програмного забезпечення, а також визначення доцільності відновлення програмних модулів та рішення інших технологічних питань щодо захисту інформації.

Фрагмент можливих комбінацій кодування дефектів, що можуть виникати під час дії кібератаки виду SQL-ін'єкція представлений на рис 1 та таблиці 11. Ієрархічний взаємозв'язок рівній теоретичного дереву стану атаки виду SQL-ін'єкція відповідає порядку кодування, який описаний раніше.

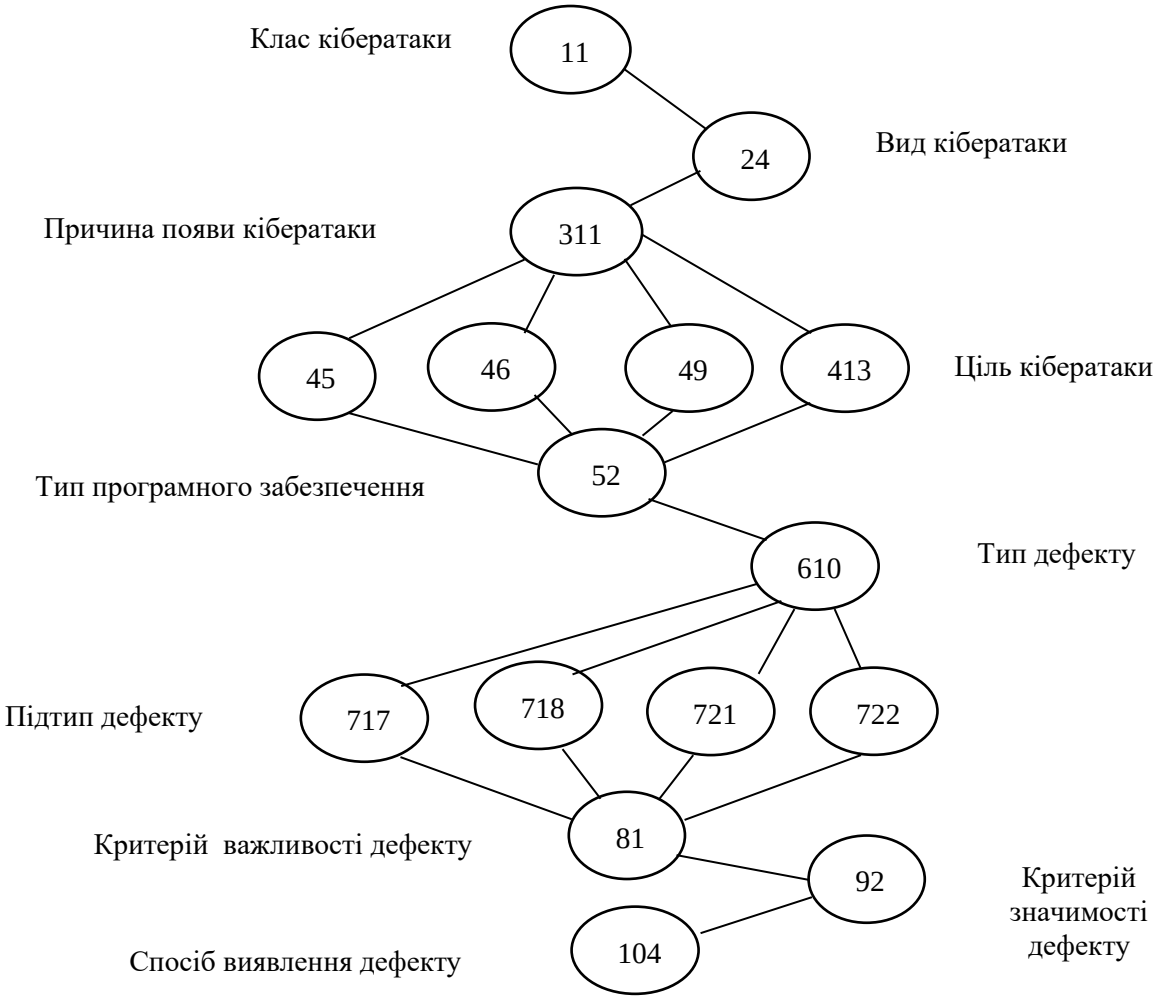

Рис.1. Теоретичне дерево стану атаки виду SQL-ін'єкція

Таблиця 11. Варіанти кодування дефектів атаки виду SQL-ін'єкція

Вид кібератаки SQL-ін'єкція			
Варіант 1	Варіант 2	Варіант 3	Варіант 4
11243114552610717 8192104	11243114652610718 8192104	1124311495261072181 92104	1124311413526107228192104

Суттєвою обставиною, яка спрощує технологічні операції класифікації дефектів є те, що для класифікації необхідно мати повний комплект зразків дефектів пошкодженого програмного забезпечення. Кожний комплект зразків характеризується набором ознак $\{PR_d\}$ та номером програми $\{NR_d\}$.

Таким чином класифікація зводиться до порівняння відповідних ознак дефекту, що діагностується, з нормативними ознаками програми, яка належить певному класу програмного забезпечення. Якщо ознаки співпадають, то зразок дефекту буде найдений, йому буде присвоєний номенклатурний номер, який відноситься до набору ознак пошкодженого програмного забезпечення.

Наведений метод класифікації має певні недоліки:

- складність вибору стандартного набору ознак дефектів програмного забезпечення;
- постійна потреба у збільшені кількості ознак;
- збільшення трудомісткості класифікації.

Зниження трудомісткості класифікації може бути забезпечено за рахунок збільшення кількості кроків порівняння та зменшення ознак дефекту програмного забезпечення. Тобто,

вказаний спосіб, зводиться до однієї операції порівняння та потребує {n} ознак дефекту, необхідних та достатніх для ідентифікації пошкодженої програми за допомогою одного порівняння. Якщо використовувати для кожного порівняння одинку ознаку, то кількість порівнянь збільшиться. Задача зводиться до вибору такого способу класифікації дефекту, який потребує мінімальних затрат під час його діагностування.

Тривалість класифікації дефекту пошкодженого програмного забезпечення можна визначити виходячи з наступного виразу:

$$T_d = T_{diag} + T_{ac} \quad (4)$$

де: T_{diag} – витрати, що присутні під час виконання операцій діагностування;

T_{ac} – час роботи автоматизованої системи, призначеної для класифікації дефектів пошкодженого програмного забезпечення.

Витрати T_{diag} , що присутні під час виконання операцій діагностування, визначаються часом, що витрачає співробітник, який здійснює попереднє діагностування пошкодженого програмного забезпечення.

Такі витрати можливо визначити, виходячи з наступного виразу:

$$T_{diag} = \sum_{i=1}^{n} T_{pdk} \quad (5)$$

де: T_{pd} - тривалість попереднього діагностування пошкодженого програмного забезпечення;

i – номер мікрооперації;

k ознака, що використовується під час класифікації.

Таким чином загальна тривалість здійснення класифікації дефектів пошкодженого програмного забезпечення визначається:

$$T_d = \sum_{i=1}^{n} T_{pdk} + T_{ac} \quad (6)$$

Висновки. Таким чином, дослідження, що були проведені, дозволяють застосувати методику класифікації та кодування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак, яка передбачає порівняння відповідних ознак дефекту, що діагностується, з нормативними ознаками програми, яка належить певному класу програмного забезпечення. Суттєвою обставиною, яка спрощує технологічні операції класифікації та кодування дефектів є те, що для класифікації необхідно мати повний комплект зразків дефектів пошкодженого програмного забезпечення, який характеризується набором ознак та номером програми. На підставі запропонованої класифікації представлена структура технологічного коду дефекту пошкодженого програмного забезпечення, який може бути використаний під час автоматизованого проектування різних технологічних операцій, а також для формалізації задачі діагностики програмного забезпечення, розробки послідовності проектування технологічного процесу діагностування пошкодженого програмного забезпечення внаслідок дії кібератак.

Список бібліографічного опису

1. Шингера Н.Я., Андрійчук П.Р. (2017). Особливості дефектів програмного забезпечення, VI Міжнародна науково-технічна конференція молодих учених та студентів. Актуальні задачі сучасних технологій. Тернопіль, Україна.

2. Junting Gao., Liping Zhang., Fengrong Zhao., Ye Zhai. (2019). Research on Software Defect Classification, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chengdu, China.

3. Felix, E. A., & Lee, S. P. (2020). Predicting the number of defects in a new software version. PLoS ONE, 15(3). Retrieved from https://doi.org/10.1371/journal.pone.0229131.

4. Hovorushchenkoa T. (2021). Criteria and Rules for Classification of Software Failures and Vulnerabilities, The 1st International Workshop on Information Technologies: Theoretical and Applied Problems. Ternopil, Ukraine

5. Akif, H. M., Reddy, R. V., Nagella, K., & Vidya, S. (2021). Software Defect Estimation Using Machine Learning Algorithms. International Journal of Recent Technology and Engineering, 10(1), 204.

6. Gopi Krishnan Rajbahadur, Shaowei Wang, Gustavo Ansaldi Oliva, Ahmed E. Hassan The Impact of Feature Importance Methods on the Interpretation of Defect Classifiers IEEE Transactions on Software Engineering PP(99), 7, January 2021 URL: http://dx.doi.org/10.1109/TSE.2021.3056941 (date of access: 06.12.2024)

7. Wang H., Yuan L., (2022). Software engineering defect detection and classification system based on artificial intelligence. Nonlinear Engineering, 11 (1), 380-386.

8. Yuxiang Gao,Yi Zhu,Yu Zhao (2022). Dealing with imbalanced data for interpretable defect prediction. Information and Software Technology, 151(2). Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0950584922001410789-823.
9. Хіль, О. С., & Яковина, В. С. (2023). Аналіз проблеми застосування методів машинного навчання для оцінювання та прогнозування дефектів програмного забезпечення. Scientific Bulletin of UNFU, 33(3), С. 110-116. URL: https://doi.org/10.36930/40330316
10. Charalampos M. , Karanikola A., Сотиріс К. (2024). Data-Efficient Software Defect Prediction: A Comparative Analysis of Active Learning-enhanced Models and Voting Ensembles. Information Sciences, 676. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S002002552400700X?via%3Dihub.
11. Tushar Agrawal, Gursimran Singh Walia., Vaibhav K. Anu. (2024). Development of a Software Design Error Taxonomy: A Systematic Literature Review. SN Computer Science, 5. Retrieved from https://link.springer.com/article/10.1007/s42979-024-02797-2.
12. Li, Z., Niu, J. & Jing, XY. (2024). Software defect prediction: future directions and challenges. Autom Softw Eng 31, 19 URL: https://doi.org/10.1007/s10515-024-00424-1.

References

Shingera N.Ya., Andriychuk P.R. (2017). Peculiarities of software defects, VI International Scientific and Technical Conference of Young Scientists and Students. Current Problems of Modern Technologies. Ternopil, Ukraine.
2. Junting Gao., Liping Zhang., Fengrong Zhao., Ye Zhai. (2019). Research on Software Defect Classification, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chengdu, China.
3. Felix, E. A., & Lee, S. P. (2020). Predicting the number of defects in a new software version. PLoS ONE, 15(3). Retrieved from https://doi.org/10.1371/journal.pone.0229131.
4. Hovorushchenkoa T. (2021). Criteria and Rules for Classification of Software Failures and Vulnerabilities, The 1st International Workshop on Information Technologies: Theoretical and Applied Problems. Ternopil, Ukraine
5. Akif, H. M., Reddy, R. V., Nagella, K., & Vidya, S. (2021). Software Defect Estimation Using Machine Learning Algorithms. International Journal of Recent Technology and Engineering, 10(1), 204.
6. Gopi Krishnan Rajbahadur, Shaowei Wang, Gustavo Ansaldi Oliva, Ahmed E. Hassan The Impact of Feature Importance Methods on the Interpretation of Defect Classifiers IEEE Transactions on Software Engineering PP(99), 7, January 2021 URL: http://dx.doi.org/10.1109/TSE.2021.3056941 (date of access: 06.12.2024)
7. Wang H., Yuan L., (2022). Software engineering defect detection and classification system based on artificial intelligence. Nonlinear Engineering, 11 (1), 380-386.
8. Yuxiang Gao,Yi Zhu,Yu Zhao (2022). Dealing with imbalanced data for interpretable defect prediction. Information and Software Technology, 151(2). Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0950584922001410789-823.
9. Khil, O. S., & Yakovyna, V. S. (2023). Analysis of the problem of applying machine learning methods for evaluating and predicting software defects. Scientific Bulletin of UNFU, 33(3), P. 110-116. URL: https://doi.org/10.36930/40330316
10. Charalampos M. , Karanikola A., Сотиріс К. (2024). Data-Efficient Software Defect Prediction: A Comparative Analysis of Active Learning-enhanced Models and Voting Ensembles. Information Sciences, 676. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S002002552400700X?via%3Dihub.
11. Tushar Agrawal, Gursimran Singh Walia., Vaibhav K. Anu. (2024). Development of a Software Design Error Taxonomy: A Systematic Literature Review. SN Computer Science, 5. Retrieved from https://link.springer.com/article/10.1007/s42979-024-02797-2.
12. Li, Z., Niu, J. & Jing, XY. (2024). Software defect prediction: future directions and challenges. Autom Softw Eng 31, 19 URL: https://doi.org/10.1007/s10515-024-00424-1.