DOI: https://doi.org/10.36910/6775-2524-0560-2023-50-07
УДК 004.94:518.5
Добришин Юрій Євгенович, к.т.н., доцент
https://orcid.org/0000-0003-2473-9507
Бондаренко Іван Дмитрович, к.ю.н, доцент
https://orcid.org/0000-0001-9164-0721
Сидоренко Сергій Миколайович, старший викладач
https://orcid.org/0009-0003-1185-1505
Національна академія Служби безпеки України, м. Київ, Україна

ФОРМАЛІЗАЦІЯ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ДІАГНОСТИКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПІСЛЯ ВПЛИВУ КІБЕРАТАК

Добришин Ю.Є., Бондаренко І.Д., Сидоренко С.М. Формалізація технологічного процесу діагностики програмного забезпечення після впливу кібератак. В роботі розглядаються питання застосування логіко-математичного апарату при формалізації процесів діагностування програмного забезпечення після його пошкодження (внаслідок впливу кібератак), під час експлуатації автоматизованих інформаційно-телекомунікаційних систем та комплексів. Запропоновано математичний апарат автоматизації процесу діагностування програмного забезпечення, який використовує логіку предикатів та дозволяє застосовувати математичні вирази, що описують властивості дефектів програмного забезпечення, правила їх виводу, відповідно до положень, прийнятих в математичній логіці. Приклади описів проектних рішень дозволяють автоматизувати проектування технологічних операцій діагностування пошкодженого програмного забезпечення після впливу кібератак, а також, забезпечують створення логічних проектних процедур щодо операцій відновлення пошкодженого програмного забезпечення.
Ключові слова: автоматизована інформаційно-телекомунікаційна система, діагностування, структурно-технологічні взаємозв'язки, предикат, дефекти програмного забезпечення.

Dobryshyn Yu., Bondarenko I., Sydorenko S. Cyberattack impacted software diagnostics technological process formalization. In the article, the application of logic-mathematical apparatus for cyberattack damaged software diagnostics processes is studied. Software diagnostics automation mathematical apparatus that uses predicate logic and allows you to use mathematical expressions describing the properties of software defects, the rules for their derivation, in accordance with the provisions adopted in mathematical logic is proposed. Described examples provide both cyberattack affected software diagnostics automation and damaged software restore procedures.
Key words: automated information and telecommunication system, diagnostics, structural-technological relationships, predicate, software defects.

Постановка наукової проблеми.

В комплексі дій, направлених на забезпечення відновлення пошкодженого програмного забезпечення внаслідок впливу кібератак, важливе місце займає його діагностування.

Технологічний процес діагностування відноситься до складних процесів, що відбуваються у багаторівневих автоматизованих системах, і ставить за мету виявлення пошкоджень та несправностей програмного забезпечення на момент їх діагностування, а також відрізняється значною трудомісткістю. Процес діагностування також є складним щодо автоматизації, через відсутність відповідних математичних моделей.

Суттєвою проблемою під час діагностування програмного забезпечення є неповнота, невизначеність і суперечливість інформації стосовно властивостей дефектів та технологічних операцій, які необхідно призначати з метою подальшого відновлення порушення конфіденційності, цілісності, доступності електронних інформаційних ресурсів, що обробляються в автоматизованих інформаційно-телекомунікаційних системах та комплексах.

Технологічний процес діагностування пошкодженого програмного забезпечення складно формалізувати за відсутності теорії і методології автоматизації процесу створення діагностичних моделей, що описують функціональні залежності між об'єктами, які приймають участь у процесі діагностування.

Практика аналізу виходу з ладу програмного забезпечення після навмисних дії, які здійснюються за допомогою засобів електронних комунікацій, складає основу методології діагностування, на базі якої можливо розробити формалізовані методики рішення окремих задач щодо призначення технологічних операцій з відновлення дефектів програмного забезпечення автоматизованих інформаційно-телекомунікаційних систем, а також визначити внутрішній зміст операцій та взаємозв'язки між ними.

Розв'язання вище описаної задачі потребує застосування спеціального математичного апарату та побудови математичної моделі, у вигляді систем множин та відношень.

Така математична модель дозволить класифікувати та проаналізувати властивості та відношення між технологічними об'єктами, що приймають участь у процесі діагностування програмного забезпечення, зможе виявити якісні зв'язки між операціями щодо властивостей дефектів та їх відновлення, формалізувати технологію призначення послідовності операцій з відновлення працездатності програмного забезпечення.

Математичний апарат повинен включати розробку математичних виразів, які описують дефекти програмного забезпечення, правила їх виводу відповідно з положеннями, прийнятими в математичній логіці.

Аналіз останніх досліджень і публікацій.

Вітчизняні та закордонні науковці, під час розгляду проблем діагностування програмного забезпечення зупиняються, в основному, на питаннях удосконалення супроводження програмного забезпечення.

Так у наукових працях [1,2] надається опис моделі супроводження інформаційних систем на етапах життєвого циклу.

На думку авторів, інформаційна система виступає пасивною категорією, як в процесі досліджень, так і в процесі проектування. Функціонування інформаційної системи описується моделями розробки, функціонування та розвитку. Автори зазначають, що модель супроводження програмного забезпечення, як частина життєвого циклу інформаційної системи, складається з послідовності виконання взаємопов'язаних процесів, у тому числі, діагностування, опис яких можливо надати у виді систем множин та відношень.

Інтерес представляють наукові роботи [3,4], автори яких стверджують, що для автоматизації програмного забезпечення потрібні методи і засоби ідентифікації дефектів проектування, а також прогнозування кількості помилок на етапі експлуатації інформаційних систем, також розглядають питання аналізу дефектів програмного забезпечення, їх класифікація, закономірності появи та шляхи усунення.

Теоретичні дослідження щодо моделей та методів діагностики розглянуті у роботах [5-9]. У своїх роботах науковці представляють теоретичні та практичні підходи щодо розробки логіко-математичних методів діагностування складних систем, а також методів оцінювання точності інформаційно-вимірювальних систем діагностики.

Аспекти автоматичної діагностики систем доповнюють наукові роботи [10-12]. В цих працях авторами успішно вирішенні завдання, що належать до методів та засобів контролю та діагностики технічних систем та програмного забезпечення.

Незважаючи на проведені дослідження, проблема автоматизації технологічного процесу діагностики програмного забезпечення після впливу кібератак, залишається актуальною і потребує подальшого вивчення та дослідження.

Реальною базою, призначеною вирішувати автоматизацію процесу діагностування програмного забезпечення, пошкодженого кібератакою, а також, визначення складу та послідовності технологічних операцій з відновлення роботи програмного забезпечення, повинна виступати відповідна формалізована теорія, яка передбачає опис предметної області шляхом визначення множини технологічних об'єктів, що приймають участь у процесі діагностування та знаходження зв'язків між ними, з подальшим формуванням певного математичного апарату.

В свою чергу, формалізація процесів діагностування програмного забезпечення, яке пошкоджено у наслідок впливу кібератак, надає можливість забезпечувати автоматизацію технології відновлення дефектів програмного забезпечення, а також є базою для створення багаторівневих програмно-апаратних систем щодо автоматизованого збору, оцінки та управління процесами діагностування та відновлення програмного забезпечення різних автоматизованих систем та комплексів.

Мета дослідження.

Метою даної статті є вирішення актуальної проблеми щодо створення формалізованих основ інформаційної технології діагностування пошкодженого програмного забезпечення, яка дозволяє визначати властивості дефектів, способи та послідовність їх відновлення.

Виклад основного матеріалу дослідження

Операції діагностики програмного забезпечення після його пошкодження та впливу кібератак, розміщуються першими у технологічному процесі відновлення працездатності програмних компонентів автоматизованої інформаційної системи та ставлять за мету визначення повної і всебічної інформації щодо стану програмного забезпечення зазначеної системи.

Проведення діагностики пошкодженого програмного забезпечення потребує суттєвих знань для вибору методів та послідовності його відновлення, у тому числі таких, які не зазначені у технічній документації у явному вигляді, а набуті співробітниками під час аналізу даних про дефекти програмного забезпечення та способів їх усунення.

Таким чином, процес діагностики відрізняється значною трудомісткістю і є складним з точки зору його автоматизації за відсутності відповідних методичних рекомендацій.

Аналіз автоматизованих інформаційних систем, що вийшли з ладу після впливу кібератак, свідчить про те, що будь-яка система $s_i \in \{S\}$, яка потребує перевірки на наявність дефектів може бути формально задана кортежом:

$$s_i = \{ I, G, C\} \quad (1)$$

де:

I- загальні відомості про автоматизовану інформаційну систему;

G – множина дефектів програмного забезпечення, де $G \in \{Q\}$;

C- структура автоматизованої інформаційної системи.

Загальні відомості $\{I\}$ про автоматизовану інформаційну систему можливо представити у вигляді багатьох параметрів, що характеризують властивості її програмного забезпечення

$$I = \{ B_{0,} B_{1,} \dots B_{i,} \dots B_{n,}\} \quad (2)$$

де:

$i = \overline{o,n}$

Склад кожного з дефекту можливо записати за допомогою його властивостей.

$$G = \{ Y_{0,} Y_{1,} \dots Y_{i,} \dots Y_{m,}\} \quad (3)$$

де:

$j = \overline{o,m}$

Структуру автоматизованої інформаційної системи $\{C\}$ опишемо графом, у якому вершинам відповідають дефекти $\{G\}$, а ребрам $\{E\}$ – множина взаємозв'язків та відношень між дефектами.

Умова, що всі елементи графу задані в одній і тій же множині, дає можливість визначити ряд відношень, які мають певні властивості та дозволяють виявити взаємозв'язки, доступні математичній інтерпретації з метою формалізації технологічного процесу діагностики програмного забезпечення після впливу кібератак, а також прийняття рішення щодо стану дефектів та послідовності проведення операцій щодо їх усунення.

Серед таких відношень необхідно виділити відношення сумісності передвизначення, слідування, домінування, еквівалентності.

Умова 1.

Сумісність «↔» програмного забезпечення P_f, де $f \in \{N\}$ що діагностується та дефекту g_j, $j \in \{G\}$де досягається за умови виконання умови:

$$\forall_N P_f \ \forall_G g_j \ \exists S_p \ \exists S_g \ \ni P\{(P_f[S_p^1] = g_f[S_g^1]) \wedge (P_f[S_p^2] = g_j[S_g^1]) \wedge (P_f[S_p^3] = g_j[S_g^3])\} \Rightarrow (P_f \leftrightarrow g_j) \quad (4)$$

де:

P_f – програмне забезпечення;

g_j – дефект програмного забезпечення;

S_p, S_g – властивості програмного забезпечення та дефекту.

Умова 2.

Відношення передвизначення «→» можливо визначити на підставі раніше сформованої умови сумісності:

$$\forall_G g_f^1 \ \forall_G g_f^2 \ \exists S_g \ \ni P\{(g_f^1[S_g] > g_f^2[S_g])\} \Rightarrow (g_f^1 \rightarrow g_f^2) \quad (5)$$

Крім того, відношення передвизначення забезпечує транзитивність дефектів програмного забезпечення, яку можливо записати у математичному вигляді:

$$\{(g_f^1 \rightarrow g_f^2) \wedge (g_f^2 \rightarrow g_f^3)\} \Rightarrow (g_f^1 \rightarrow g_f^3) \quad (6)$$

Умова 3.

Послідовність упорядкування дефектів під час проведення діагностики, описується за допомогою умови слідування. У математичному вигляді зазначене відношення доцільно записати наступним чином:

$$\forall_N P_f\ \exists g_f^1\ \exists g_f^2\ \exists S_p\ \exists S_g\ \ \exists\ P\{(P_f[S_p^1] \leftrightarrow g_f^1[S_g^1]) \wedge (P_f[S_p^2] \leftrightarrow g_f^2[S_g^2])\ \wedge\ (g_f^1 \neq g_f^2)\} \Rightarrow (g_f^1 \sim g_f^2) \quad (7)$$

Умова 4.

Дефект g_f^1 домінує над дефектом g_f^2 тільки тоді, коли буде виконуватися наступна умова:

$$\forall_G\ g_f^1\ \forall_G\ g_f^2\ \ \exists S_g\ \exists\ P\left\{\left(P_f(g_f^1[S_g^1] = 1) \wedge \left(\left(P_f(g_f^2[S_g^2] = 0)\right) \Rightarrow \left(g_f^1\ \gg g_f^2\right)\right)\right\} \quad (8)$$

Умова 5.

Два дефекти вважаються еквівалентні один одному, якщо виконується наступна умова:

$$\forall_G\ g_f^1\ \forall_G\ g_f^2\ \ \exists S_g\ \exists\ P\left\{\left((g_f^1[S_g^1] = g_f^2[S_g^2]) \wedge \left(\left(g_f^1\ \neq g_f^2\right)\right) \Rightarrow \left(g_f^1\ \approx g_f^2\right)\right\} \quad (9)$$

Сформовані відношення та умови їх виконання, дозволяють виокремити задачу побудови технологічного процесу діагностики програмного забезпечення після впливу кібератак, як послідовність логічних перетворень над формальним описом зазначеного процесу.

Також, реалізація третьої технологічної умови значно зменшує кількість визначених дефектів за рахунок того, що кожний дефект класифікований по ступені впливу на працездатність програмного забезпечення, тобто важливості.

Це дозволяє визначати для кожного програмного забезпечення основні модулі, обов'язкові для перевірки під час діагностики.

Інші дефекти, які можна вважати неосновними і наявність яких тимчасово дозволяється, можуть розглядатися як умовно допустимі і пов'язані з основними дефектами певними закономірностями. Тобто має місце залежність:

$$g_{fні} = F(g_{foi}) \quad (10)$$

де, $g_{fні}$ величина неосновного дефекту, g_{foi} величина основного дефекту.

Необхідно підкреслити, що після діагностики та вибору способів усунення основних дефектів програмного забезпечення, для неосновних дефектів можливе автоматичне призначення способів їх відновлення.

В результаті, з усіх дефектів програмного забезпечення, що були виявлені, розглядатимуться тільки властивості основних дефектів, що суттєво зменшить час на проведення технологічних операцій з діагностики програмного забезпечення після його пошкодження та впливу кібератак.

Висновки та перспективи подальшого дослідження.

Таким чином, наведені формалізовані умови дають можливість розробити алгоритм послідовності проектування технологічного процесу діагностики програмного забезпечення після впливу кібератак. Також, запропонований метод дозволяє здійснити автоматизацію процесу, як одну із задач щодо створення сучасних систем автоматичного проектування технологічних процесів відновлення програмного забезпечення після пошкодження та впливу кібератак.

Аналіз діагностування пошкодженого програмного забезпечення внаслідок впливу кібератак, дає можливість визначити ряд відношень, які мають певні властивості та дозволяють виявити взаємозв'язки, доступні математичної інтерпретації з використанням логіко-математичного апарату щодо формалізації технологічного процесу діагностики програмного забезпечення. Відношення сумісності, передвизначення, слідування, домінування, еквівалентності дозволяють створити умови автоматизованого проектування різних проектних рішень під час розробки операцій з діагностування пошкодженого програмного забезпечення та у подальшому визначитися з послідовністю призначення технологічних операцій з відновлення його дефектів.

Список бібліографічного опису

1. Lisetcky Yu. (2018). Models of support for enterprise information system by stages of the life cycle software. *Software & Systems*, 3, 455-460.
2. Руденська Г. (2020). Моделі та процеси життєвого циклу інформаційної системи управління оборонними ресурсами. *Інформація та управління проєктами інформації Збройних Сил*. 1, 59 - 65.
3. Нечай О. (2009). Метод діагностики об'єктно–орієнтованого програмного забезпечення. *Вісник НАУ*. 5, 100 - 111.
4. Щербаков О. Луценко Є. (2011). Оцінка ефективності тестування програмного забезпечення на основі аналізу кількості та критичності знайдених дефектів. *Системи обробки інформації*. 3, 88 - 92.

5. Литвиненко О. Є. Нечипорук О.П. (2016). *Логіко-математичні методи діагностування складних систем*. Київ: Артмедіа прінт,
6. Марченко Н.Б., Нечипорук В.В. Нечипорук О.П., Пепа Ю.В. (2014). *Методи оцінювання точності інформаційно-вимірювальних систем діагностики.* Київ: Задруга.
7. Lisetcky Yu., Snytyuk V. (2015). *Formal presentation of a corporate integrated system in the form of a set mathematical models,* 17-th Intern. Conf. System Analysisand Information Technologies SAIT 2015. Kyiv.
8. Syrotkina O. (2011) *Automatic Subsystem of Data Transmission Diagnostics is the Baseof Reliability and Stability of Modern SCADA Systems for Mission-critical Applications*. 6th International Forum for Students and Young Researches, April 14-15, 2011: Abstracts. – Dnipropetrovsk.
9. Syrotkina O., Alekseyev M. (2016). Software Diagnostics for Reliability of SCADA Structural Elements. Power Engineering and Information Technologies in Technical Objects Controls. *Taylor & Francis Group,* 259-265.
10. Сакович Л., Павлов В., Лівенцев С., Небесна Я. (2012). Порівняльний аналіз моделей надійності програмного забезпечення засобів спеціального зв'язку. *Information Technology and Security*. 2(2). 61 - 69.
11. Nechyporuk O. (2014) .Adjustment of the generalized logical model of compound systems diagnosting according to the situation. *The Advanced Science Journal*, 2, 20 - 23.
12. .Guchenko I. (2014) Usability management in the context of software architecture. *Інженерія програмного забезпечення*. 2(18), 20 - 25.

References

1. Lisetcky Yu. (2018). Models of support for enterprise information system by stages of the life cycle software. *Software & Systems*, 3, 455-460.
2. Rudenska G. V. (2020). Models and processes of the life cycle of the defense resource management information system. *Information and management of information projects of the Armed Forces*, 1, 59–65.
3. Nechay O.S. (20090 A method of object-oriented software diagnostics. *Bulletin of NAU*. 5, 100-111.
4. Shcherbakov O. V. Lutsenko. E.S. (2011). Evaluation of the effectiveness of software testing based on the analysis of the number and criticality of the defects found. *Information processing systems*, 3, 88–92.
5. Lytvynenko O.E. Nechiporuk O.P. (2016). *Logical-mathematical methods of diagnosing complex systems.* Kyiv: Artmedia Print.
6. Marchenko N.B., Nechiporuk V.V. Nechiporuk O.P., Pepa Y.V. (2014). *Methods of assessing the accuracy of information and measurement systems of diagnostics*. Kyiv: Zadruga.
7. Lisetcky Yu., Snytyuk V. (2015). Formal presentation of a corporate integrated system in the form of a set mathematical models, 17-th Intern. Conf. System Analysisand Information Technologies SAIT 2015. Kyiv.
8. Syrotkina O. (2011) Automatic Subsystem of Data Transmission Diagnostics is the Baseof Reliability and Stability of Modern SCADA Systems for Mission-critical Applications. 6th International Forum for Students and Young Researches, April 14-15, 2011: Abstracts. – Dnipropetrovsk.
9. Syrotkina O., Alekseyev M. (2016). Software Diagnostics for Reliability of SCADA Structural Elements. *Power Engineering and Information Technologies in Technical Objects Controls. Taylor & Francis Group*, 259-265.
10. Sakovich L.M., Pavlov V.P., Liventsev S.P., Nebesna Y.E. (2012). Comparative analysis of software reliability models of special communication tools. *Information Technology and Security,* 2(2), 61-69.
11. Nechyporuk O. (2014). Adjustment of the generalized logical model of compound systems diagnosting according to the situation. *The Advanced Science Journal*, 2, 20-23.
12. Guchenko I.V. (2014). Usability management in the context of software architecture. *Software engineering*, 2(18), 20 - 25.