ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП

ЗБІРНИК МАТЕРІАЛІВ

Міжнародної науково-практичної Інтернет-конференції

20-21 листопада 2023 р.

Міністерство освіти і науки України

Вінницький національний технічний університет
Національна академія Державної прикордонної служби України
ім. Богдана Хмельницького
Вінницький національний медичний університет ім. М.І. Пирогова
КЗВО «Вінницька академія безперервної освіти»
КЗ «Сумський обласний інститут післядипломної педагогічної освіти»
Інститут комп'ютерних систем і технологій "Індустрія 4.0"
ім. П. Н. Платонова
Люблінська політехніка (Польща)
Університет Бельсько-Бяльський (Польща)

«ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП»

ЗБІРНИК МАТЕРІАЛІВ

Міжнародної науково-практичної Інтернет-конференції
20-21 листопада 2023 р.

Суми/Віннця
НІКО/КЗВО «Вінницька академія безперервної освіти»
2023

УДК 004
ББК 32.97
Е50

Рекомендовано до видання Вченою радою КЗВО «Вінницька академія безперервної освіти» (протокол № 8 від 20.11.2023 р.)

Електронні інформаційні ресурси: створення, використання, доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2023. – 336 с.

ISBN 978-617-7422-23-4

Збірник містить матеріали Міжнародної науково-практичної Інтернет конференції «Електронні інформаційні ресурси: створення, використання, доступ. Матеріали збірника подано у авторській редакції. Автори опублікованих матеріалів несуть повну відповідальність за підбір, точність наведених фактів, цитат, статистичних даних, власних імен та інших відомостей, Матеріали відтворюються зі збереженням змісту, орфографії та синтаксису текстів, наданих авторами.

УДК 004
ISBN 978-617-7422-23-4

Для того, щоб скористатися даним ресурсом перейдіть на bookcreator.com і натисніть увійти. Увійдіть як студент або викладач. Після входу створіть власну бібліотеку та почніть створювати нову книгу, натиснувши «+ Нова книга» у верхньому лівому куті.

Використовуйте «+» у верхньому правому куті, щоб імпортувати або створити будь-який вміст. Використовуйте знак «і» у верхньому правому куті, щоб налаштувати параметри сторінки.

Скористайтеся кнопкою відтворення, щоб перейти на сторінку попереднього перегляду, щоб поділитися чи завантажити книгу, або натисніть Read To Me, щоб прослухати свою книгу.

Найкращий спосіб поділитися своєю книгою Book Creator — опублікувати її в Інтернеті. Публікуючи свою книгу в Інтернеті, ви отримаєте посилання на версію книги, доступну лише для читання. Можна гортати сторінки, відтворювати аудіо та відео та натискати будь-які гіперпосилання.

При публікуванні книги, можна обрати, чи залишати її приватною, що означає, що ви отримуєте безпечне посилання на книгу, яким можете поділитися з ким завгодно. Або можна вибрати «Для всіх», що означає, що ваша книга з'явиться в результатах пошуку Google (та в інших пошукових системах), а значить більше людей зможуть знайти її та поділитися нею.

Цікавою особливістю опублікованих книг є те, що будь-які зміни, які ви вносите в книгу в Book Creator, автоматично зберігаються в онлайн-версії. Тому немає потреби перевидавати книгу щоразу, коли були внесені зміни. Достатньо оновити браузер, і всі зміни набудуть чинності.

В будь-який момент можна скасувати публікацію своєї книги, просто натисніть значок глобуса у верхньому правому куті вашої книги. Натисніть посилання «Зупинити публікацію», і книга більше не буде доступна в Інтернеті.

Book Creator і публікація в Інтернеті сертифіковані iKeepSafe.org як безпечні для шкіл на відповідність COPPA, FERPA, California Ed Code 49073.1 і SOPIPA.

Creator Book - це потужний інструмент, який може використовуватися для підвищення ефективності навчання і задоволення потреб сучасних учнів. Він є доступним і простим у використанні, що дозволяє викладачам та учням створювати інтерактивні навчальні матеріали без спеціальних навичок програмування.

Creator Book є універсальним інструментом, який може використовуватися для створення інтерактивних книг для різних предметів і цільових аудиторій.

Перелік використаних джерел

1. Бондарчук Ж.А. Інтерактивний навчальний посібник "Програмування мовою Python", створений за допомогою сервісу H5P, Луцьк – 2021, 52 с.
2. Гусак Л.В. BooK Creator: конструктор мультимедійної книги. URL: https://vseosvita.ua/library/book-creator-konstruktor-multymediinoi-knyhy-644084.html (дата доступу 05.11.2023)
3. Онлайн-інструменти для викладання та навчання. URL: https://edtechbooks.org/onlinetools/book-creator (дата доступу 05.11.2023)
4. Book Creator. URL: https://bookcreator.com (дата доступу 05.11.2023)

УДК 004.415.5

ТУРЕНКО В.Р., РОМАНЮК О.В.
Вінницький національний технічний університет

РЕВЕРС-ІНЖЕНЕРІЯ ДЛЯ ОЦІНКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

Анотація. Розглянуто поняття реверс-інженерії та процес її застосування для оцінювання якості програмного продукту. Розглянуто переваги, які надає метод реверс-інженерії при оцінюванні якості архітектури, дизайну та безпеки.

Ключові слова: реверс-інженерія, зворотна розробка, якість ПЗ, безпека, проектування.

Якість є важливою складовою будь-якого програмного забезпечення. Відсутність будь-яких процесів забезпечення якості може призводити до значних наслідків. В історії достатньо випадків, коли ненадійне, невдало протестоване програмне забезпечення призводило до аварій, значних фінансових збитків та, навіть, до загибелі людей. Наприклад, втрата 440 мільйонів доларів за 45 хвилин компанією Knight Capital у 2012 [1], опромінення великими дозами радіації пацієнтів апаратом для променевої терапії Therac-25 у період з 1985-1987 рр. [2] та інші.

Існує ряд помилок, які можуть призводити до несправності: логічні, безпеки, інтерфейсу користувача та інші. Для запобігання найбільш серйозних проблем уже існує багато практик та методів. Але більшість з них розглядає програму до процесу компіляції/інтерпретації. При її безпосередньому запуску в машинному коді можуть виникати додаткові проблеми дизайну та безпеки. Реверс-інженерія призначена їх виявити, при цьому оперує мінімальною кількістю даних про вихідний код та архітектуру проекту.

Таким чином, аналіз застосування реверс-інженерії для оцінки якості програмного забезпечення є актуальною задачею.

Поняття, основні методи та застосування реверс-інженерії

Реверс-інженерія (також зворотна розробка, зворотне проектування) – це процес дослідження, отримання інформації про те, як працює певний об’єкт. Реверс-інженерія подібна до наукових досліджень, однак застосовується не до природних об’єктів, а до штучних, створених людиною [3].

Реверс-інженерія в програмуванні – це метод розробки, що полягає в дослідженні програми з метою розуміння принципів її роботи. Для дослідження зазвичай використовуються такі методи, як [4]:

1. *Моніторинг активності.* Дозволяє провести дослідження специфікацій та протоколів обміну інформацією. Цей метод не дає уявлення про логіку роботи програми.
2. *Дизасемблювання.* Полягає в перетворенні машинного коду в код асемблера, який придатний для читання та дослідження людиною. З використанням цього методу можна проводити дослідження будь-яких програм. Але використання певних технік при розробці програми може значно ускладнити дизасемблювання. Також цей метод вимагає високої кваліфікації людини, що проводить зворотну розробку, та великих затрат часу.
3. *Декомпіляція.* Полягає в перетворенні машинного коду у код на мові високого рівня, придатний для аналізу людиною. Метод важко реалізувати, що пов’язано зі складністю розробки інструментів декомпіляції.

Реверс-інженерія може використовуватися як при розробці продукту (за наявністю вихідного коду), так і для дослідження готового продукту (без вихідного коду). Серед найчастіших застосувань реверс-інженерії виділяють [5]:

1. *Зворотна розробка і програмні системи.* У широкому розумінні зворотна розробка – це дослідження програми ззовні, людиною, що не брала участі у розробці продукту. У такому разі вона дозволяє зрозуміти, як працює програма або система. Зворотну розробку можна використовувати для розв’язання цілого ряду задач кібербезпеки: пошук дефектів системи, дослідження вірусів та іншого шкідливого ПЗ, а також визначення складності відновлення критично важливих алгоритмів програмного забезпечення, які можуть допомогти у запобіганні шахрайству.
2. *Зворотна розробка і аналіз продукту*. Зворотне проектування корисне в аналізі продукту, оскільки воно допомагає ідентифікувати складові елементи, зрозуміти досвід користувача продукту та оцінити конкуренцію.
3. *Зворотна розробка та дослідницькі цілі.* Зворотне проектування є корисним у дослідницьких цілях для забезпечення якості програмного продукту. Дизасемблювання всіх компонентів показує, наскільки якісно був виготовлений програмний продукт. Також метод дозволяє вимірювати певні метрики, гарантуючи відповідність продукту вимогам. Для уникнення порушень патентних прав може знадобитися застосування зворотного проектування для аналогічного продукту конкурента. Окрім того, зворотне

проектування є цінним джерелом збору інформації, створення точних моделей, побудови власних моделей та розробки власних методів зворотного проектування, що задовольняють вимоги проекту.

Серед прикладних застосувань реверс-інженерії виділяють: модифікацію комп'ютерних ігор, втручання у протокол обміну даних онлайн-ігор та сервісів, відкриття платних функцій умовно-безкоштовних програм, розблокування програм, що вимагають уведення реєстраційного або ліцензійного коду тощо.

Реверс-інженерія для оцінки якості архітектури та дизайну програмного продукту

Реверс-інженерія та тестування. Реверс-інженерія має багато спільного з тестуванням методом «чорної скриньки» в інженерії програмного забезпечення. Тестувальник зазвичай використовує програмний інтерфейс додатку (API), але його ціль – знайти баги та незадокументовані функції маючи лише сам продукт, без вихідного коду [6].

Реверс-інженерія коду. Якщо реверс-інженерія застосовується до додатку, з метою відновлення його вихідного коду, то цей процес називають реверс-інженерія коду (reverse code engineering, RCE).

Для прикладу, декомпіляція скомпільованих файлів для Java платформи може бути виконано з використанням Jad. Одним з відомих випадків реверс-інженерії була перша реалізація PC BIOS, що була створена сторонніми розробниками. Це поклало початок індустрії IBM-сумісності для платформи комп'ютерів IBM, що була домінантною протягом багатьох років.

Класичним прикладом реверс-інженерії є програмне забезпечення Samba, що дозволяло системам, що не працюють на Microsoft Windows, передавати файли на системи, що працюють на ній. Розробники Samba використали реверс-інженерію для дослідження роботи обміну файлами Windows, щоб можна було емулювати її роботу на комп'ютерах інших операційних систем. Подібних прикладів багато, і серед найбільш відомих можна навести ще: Wine (емулює Windows API), OpenOffice.org (працює з форматами файлів Microsoft Office, специфікація яких не публікувалась Microsoft) та інші.

Реверс-інженерія коду доступна як опція у деяких IDE (наприклад, IntelliJ IDEA) та дозволяє будувати UML-діаграми (у тому числі діаграми класів) за наявним вихідним кодом. Це дозволяє оцінити реалізовану архітектуру додатку, порівняти її зі спроектованою.

Реверсна-інженерія для дослідження протоколу. Протокол – це набір правил, що описує формати повідомлень та способи їх обміну. Іншими словами, він визначає автомат станів протоколу. Таким чином, задача застосування зворотної розробки до протоколів поділяється на дві підзадачі: дослідження формату повідомлень та реверс-інженерія автомату станів.

Формати повідомлень традиційно піддавалися зворотній інженерії шляхом виснажливого ручного процесу, який включав аналіз того, як протоколи обробляють повідомлення, але останні дослідження запропонували ряд автоматичних рішень. Як правило, автоматичні підходи групують повідомлення в кластери за допомогою різних аналізів кластеризації або вони емулюють реалізацію протоколу, що відстежує обробку повідомлень.

Зі зворотним проектуванням кінцевих автоматів протоколів усе простіше. Загалом, автомати стану протоколу можна вивчати або через процес офлайн-навчання, який пасивно спостерігає за комунікацією та намагається побудувати найзагальніший автомат стану, який приймає всі спостережувані послідовності повідомлень, і онлайн-навчання, яке дозволяє інтерактивне генерування зондування послідовності повідомлень і прослуховування відповідей на ці пробні послідовності. Загалом відомо, що офлайн-навчання малих кінцевих автоматів є NP-повним [7], але онлайн-навчання можна здійснити за поліноміальний час [8].

Інші компоненти типових протоколів, як-от шифрування та хеш-функції, також можуть бути оброблені автоматично. Як правило, автоматичні підходи відстежують виконання реалізацій протоколу та намагаються виявити буфери в пам'яті, що містять незашифровані пакети [9].

Реверс-інженерія для оцінки безпеки програмного продукту

Реверс-інженерія є важливим інструментом для оцінки безпеки програмного продукту, дозволяючи спеціалістам виявляти та виправляти потенційні вразливості, забезпечуючи високий рівень захисту. Найвідомішим методом застосування реверс-інженерії для оцінки безпеки програмного продукту є тестування на проникнення. Розглянемо, як саме цей метод може сприяти підвищенню рівня безпеки програм.

Тестування на проникнення (англ. penetration test, pen test) передбачає проведення планованих атак етичними хакерами на інфраструктуру безпеки компанії для виявлення вразливостей, які потребують усунення. Це є частиною всебічної стратегії забезпечення безпеки веб-додатків [10].

Існує декілька типів тестування на проникнення [10]:

1. *Тестування білої скриньки (open-box pen test).* При тестуванні білої скриньки хакеру буде заздалегідь надано деяку інформацію щодо інформації про безпеку цільової компанії.
2. *Тестування чорної скриньки (closed-box pen test).* Також відомий як «одинарний сліпий» тест, у якому хакеру не надається жодної довідкової інформації, окрім назви цільової компанії.
3. *Тест на приховане проникнення (covert pen test).* Також відомий як «подвійний сліпий» тест. Це ситуація, коли майже ніхто в компанії не знає, що відбувається тестування, включно з IT-спеціалістами та спеціалістами з безпеки, які реагуватимуть на атаку. Для прихованих тестів особливо важливо, щоб хакер заздалегідь мав у письмовій формі обсяг та інші деталі тесту, щоб уникнути проблем із правоохоронними органами.
4. *Зовнішнє тестування на проникнення (external pen test).* У зовнішньому тесті «білий» хакер протистоїть зовнішнім технологіям компанії, таким як веб-сайт і зовнішні мережеві сервери. У деяких випадках хакеру можуть навіть не дозволити увійти в будівлю компанії. Це може означати проведення атаки з віддаленого місця або проведення тесту з вантажівки чи фургона, припаркованого неподалік.
5. *Внутрішнє тестування на проникнення (internal pen test).* Під час внутрішнього тестування етичний хакер виконує тест із внутрішньої мережі компанії. Цей вид тесту корисний для визначення того, скільки шкоди може завдати незадоволений працівник в обхід брандмауера компанії.

Після завершення перевірки етичний хакер надсилає результати своєї роботи команді безпеки цільової компанії. Потім цю інформацію можна використати для впровадження оновлень безпеки, щоб усунути будь-які вразливості, виявлені під час тестування, таким чином покращуючи безпеку продукту – одну зі складових якості.

Реверс-інженерія є ключовим інструментом для виявлення та усунення потенційних безпекових проблем у програмних продуктах. Цей підхід відкриває шлях до створення високоякісних та безпечних програм, що відповідають вимогам сучасних стандартів безпеки.

Висновки

Отже, реверс-інженерія може бути корисною для оцінювання якості програмного забезпечення. Завдяки цій техніці можна оцінити та покращити рівень безпеки розроблюваного продукту, протестувати на вразливість протокол обміну даними, оцінити якість архітектури та дизайну програмного забезпечення.

Список використаних джерел

1. Knight Capital Says Trading Glitch Cost It $440 Million - The New York Times. URL: https://archive.nytimes.com/dealbook.nytimes.com/2012/08/02/knight-capital-says-trading-mishap-cost-it-440-million/ (дата звернення: 12.11.2023).
2. Leveson, N. G., Turner, C. S. (1992). An investigation of the therac-25 accidents. Dep. of Information and Computer Science, Univ. of Calif.
3. Eilam, E., Chikofsky, E. J. (2005). Reversing secrets of reverse engineering. Wiley.
4. Зворотна розробка – Вікіпедія. URL: https://uk.wikipedia.org/wiki/ Зворотна_розробка (дата звернення: 12.11.2023).
5. A., Abigail. (2021). Reverse Engineering Research.

6. Shahbaz, Muzammil (2012). Reverse Engineering and Testing of Black-Box Software Components: by Grammatical Inference techniques. LAP LAMBERT Academic Publishing.
7. Gold, E (1978). "Complexity of automaton identification from given data". Information and Control.
8. D. Angluin (1987). "Learning regular sets from queries and counterexamples". Information and Computation.
9. Polyglot: automatic extraction of protocol message format using dynamic binary analysis. J. Caballero, H. Yin, Z. Liang, and D. Song. Proceedings of the 14th ACM conference on Computer and communications security, pp. 317–329.
10. What is penetration testing? | What is pen testing? | Cloudflare. URL: https://www.cloudflare.com/learning/security/glossary/what-is-penetration-testing/ (дата звернення: 12.11.2023).

ТУШИНСЬКИЙ В.Е.
Вінницький національний технічний університет

ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В НАУКОВИХ ДОСЛІДЖЕННЯХ: АНАЛІЗ СУЧАСНОГО СТАНУ

Анотація: Стаття присвячена детальному аналізу ефективності використання штучного інтелекту (ШІ) у сфері наукових досліджень. Вона висвітлює сучасний стан використання ШІ в науковому середовищі та визначає ключові фактори, що впливають на успішність застосування цих технологій. Розглядаються переваги та виклики, які виникають при використанні ШІ в різних наукових областях. Також аналізуються тенденції розвитку та перспективи використання ШІ для покращення якості наукових досліджень.

Abstract: The article is devoted to a detailed analysis of the effectiveness of artificial intelligence (AI) in the field of scientific research. It highlights the current state of AI use in the scientific environment and identifies the key factors that influence the success of the application of these technologies. The advantages and challenges that arise when using AI in various scientific fields are considered. It also analyses development trends and prospects for using AI to improve the quality of scientific research.

Ключові слова: штучний інтелект, наукові дослідження, ефективність, технології, інновації.

В сучасному науковому світі роль штучного інтелекту (ШІ) надзвичайно важлива, оскільки ця технологія відкриває нові можливості для виконання складних обчислень, аналізу великих обсягів даних та автоматизації дослідницьких процесів. Застосування ШІ в наукових дослідженнях дозволяє значно підвищити продуктивність та точність результатів.

Однією з ключових переваг використання ШІ в науці є здатність штучного інтелекту до аналізу великих обсягів інформації за короткий період часу. Це сприяє прискоренню наукових досліджень і дозволяє вченим зосередитися на творчому аспекті своєї роботи, залишаючи рутинні завдання на плечах алгоритмів ШІ.

Наприклад, у медицині штучний інтелект використовується для розробки нових ліків, діагностики захворювань та персоналізації лікування. У фізиці штучний інтелект використовується для моделювання складних фізичних явищ, таких як квантові взаємодії та космологія. У хімії штучний інтелект використовується для розробки нових матеріалів та синтезу хімічних речовин.

Крім того, штучний інтелект може допомогти вченим у таких завданнях, як:

- Отримання доступу до та обробка інформації з різних джерел, включаючи наукові статті, патентні бази даних, соціальні мережі та інші.
- Визначення тенденцій і закономірностей в даних.
- Формування гіпотез і теорій.
- Проектування експериментів і дослідницьких методів.

Разом із перевагами при використанні штучного інтелекту в науці виникають і виклики. Наприклад, проблеми етичного характеру пов'язані з використанням алгоритмів при прийнятті

ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП:

Збірник матеріалів
Міжнародної науково-практичної Інтернет-конференції
20-21 листопада 2023 р.

Редактор С.А.Пойда, М.С. Ніколаєнко
Комп'ютерне верстання С.А.Пойда, М.С. Ніколаєнко

Підписано до друку 15.11.2023 Гарнітура Times New Roman
Формат 60х84/16 Папір офсетний
Друк цифровий Ум. друк. арк. 19,4
Тираж 300 пр. Зам. № 2/23

Видавництво НІКО
м.Суми, вул.Харківська, 54
Свідоцтво про внесення до Державного реєстру
суб'єктів видавничої справи України
серія СМв № 044
від 15.10.2012
E-mail: ms.niko@i.ua
Телефон для замовлень: +38(066) 270-64-68