УДК 004.8:004.056.53
DOI: 10.31673/2409-7292.2024.040008

Гавриляк В. Р., Опірський І. Р.

АНАЛІЗ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В DEVSECOPS ТА КІБЕРБЕЗПЕЦІ

Зі стрімким глобальним розвитком цифрових технологій зростає й кількість кіберзагроз, які стають дедалі складнішими та витонченішими. Організації змушені постійно адаптувати свої підходи до кібербезпеки, впроваджуючи сучасні методи захисту для мінімізації ризиків і запобігання втратам критично важливих даних. Одним із перспективних варіантів є DevSecOps, який зосереджується на інтеграції безпеки на кожному етапі життєвого циклу розробки програмного забезпечення. Це дозволяє не лише підвищити захищеність продукту й знизити ризики завдяки автоматизації процесів, але й забезпечити постійний моніторинг та швидке реагування на загрози. Відповідно до сучасних тенденцій, використання штучного інтелекту стало новим етапом у розвитку кібербезпеки та DevSecOps. ШІ надає організаціям можливість виявляти загрози на ранніх стадіях шляхом аналізу великих обсягів даних та виявлення аномалій у режимі реального часу. Однак використання ШІ повинно бути контрольованим і відповідати етичним стандартам, оскільки обробка великих обсягів даних викликає питання щодо рівня конфіденційності. Тому важливо застосовувати ШІ з належним контролем та дотриманням нормативних вимог. У цьому дослідженні проаналізовано методологію DevSecOps та роль штучного інтелекту в кібербезпеці. Розглянуто міжнародні напрацювання в сфері застосування ШІ в DevSecOps і встановлено, що хоч кількість досліджень у цій галузі постійно зростає, але у науковому просторі України таких робіт небагато. Таким чином, поточна робота може стати теоретичною основою для українських організацій та науковців у подальших дослідженнях використання ШІ в кібербезпеці та циклі розробки програмного забезпечення.

Ключові слова: DevOps, DevSecOps, штучний інтелект (ШІ/AI), цикл розробки програмного забезпечення (SDLC), кібербезпека, ризики.

Вступ

Попри впровадження гнучких методологій розробки програмного забезпечення, на початку 21 століття команди розробників (Dev) та команди операційної підтримки (Ops) ще рідко взаємодіяли між собою. Для вирішення цієї проблеми була представлена методологія DevOps, яка мала на меті налагодити комунікацію між цими командами для досягнення кращих результатів у розробці.

Протягом наступного десятиліття практики DevOps стали невід'ємною частиною процесу створення програмного забезпечення, вдало інтегруючись у роботу більшості сучасних команд. Проте, незважаючи на успіхи та широке впровадження, DevOps виявився не без недоліків. Однією з основних проблем було часте протиріччя між швидкими циклами розробки та надійними заходами безпеки, які часто впроваджувалися на завершальних етапах, що створювало ситуацію "security bottleneck". Перебої, людські помилки, кібератаки, витоки даних, програми-вимагачі, вразливості в безпеці і, як наслідок, втрата даних — це реальність, з якою стикнувся DevOps. Ця проблема підкреслила потребу в еволюції методології, що призвело до появи DevSecOps. Проте розвиток методології DevOps не зупинився на простому додаванні аспекту кібербезпеки. Стрімкий прогрес інформаційних технологій, значною мірою трансформував сучасний світ завдяки застосуванню штучного інтелекту. Це сприяло виникненню нового підходу до забезпечення безпеки в життєвому циклі розробки програмного забезпечення: AI in DevSecOps.

Аналіз літературних джерел та формулювання проблеми

На сьогоднішній день організації стикаються з складним завданням: впроваджувати програмне забезпечення у найкоротші терміни, не поступаючись при цьому надійністю заходів безпеки. Щоб відповідати таким вимогам, розробникам та компаніям важливо впроваджувати DevSecOps. Дана методологія забезпечує автоматизацію впровадження безпеки на кожному етапі життєвого циклу DevOps, починаючи з початкового проектування і до інтеграції, тестування, розгортання та фінальної доставки. Однак впровадження DevSecOps стикається з низкою труднощів, серед яких складність інтеграції безпекових рішень у швидкі та гнучкі процеси розробки, а також значні витрати ресурсів на постійний моніторинг і адаптацію до нових загроз.

Одним з імовірних методів вирішення подібних проблем може стати штучний інтелект, який за останні роки почав стрімко розвиватися та інтегруватися у велику кількість сфер, зокрема в розробку програмного забезпечення та кібербезпеку. Спираючись на "Global DevSecOps Report 2024" від Gitlab все більша частина опитуваних компаній та розробників вже використовує або планує почати використовувати ШІ в найближчі 2 роки для створення програмного забезпечення [1]. Згідно з дослідженнями BlackDuck "Global State of DevSecOps 2024": "Революція штучного інтелекту вже завершилася — і ШІ переміг, принаймні в контексті інтеграції ШІ в процеси розробки програмного забезпечення". Впровадження ШІ у сферу розробки програмного забезпечення вже пройшло критичний момент, адже понад 90% респондентів їхнього опитування використовують допомогу ШІ в тій чи іншій мірі [2]. В свою чергу, дослідження IBM "Cost of a Data Breach Report 2024" показало, що використання штучного інтелекту в системи безпеки та автоматизації бізнесу окуповується, знижуючи суму збитків на втрачені дані в районі 2.2 мільйонів доларів США [3]. Інтеграція штучного інтелекту у DevSecOps може стати вирішальним фактором, оскільки обіцяє низку переваг, таких як підвищення ефективності та покращення виявлення загроз. Однак це також вводить значні ризики та виклики, які не можна ігнорувати: багато організацій не готові до управління кіберзагрозами, які ШІ привносить.

Проблеми впровадження DevSecOps та штучного інтелекту досліджуються багатьма міжнародними науковцями та спеціалістами відповідних галузей. Так згідно з статистичними дослідженнями проведеними в цій роботі, кількість наукових напрацювань в галузі "AI in DevSecOps" зростає кожного року, наприклад відсоток робіт за 2024 рік складає близько 40% від усіх досліджень за період 2021-2024 року. Однак тільки незначна частина цих досліджень фокусується на комплексному використанні ШІ для всіх етапів DevSecOps[4]. Також варто відзначити, що не зважаючи на значну кількість міжнародних досліджень, відсоток українських робіт в цьому напряму дуже малий. Це створює прогалини у взаємодії на різних етапах імплементації штучного інтелекту в процеси DevSecOps.

Мета роботи та цілі дослідження

Метою роботи є дослідження підходів використання штучного інтелекту в аспекті кібербезпеки та DevSecOps.

Для вирішення поставленої мети розглянуто такі завдання:

1. Проаналізувати методологію DevSecOps, як подальший етап розвитку DevOps.
2. Оглянуто сучасні тенденції використання ШІ в кібербезпеці та DevSecOps.
3. Проведено статистичний аналіз сучасних міжнародних та українських досліджень в галузі "AI in DevSecOps".

Результати досліджень

У 2023 році розмір ринку DevSecOps становив 6,3 мільярда доларів США, і очікується, що до 2032 року він досягне 45,93 мільярда доларів[5]. Так згідно з минулорічним дослідженням від Gitlab, завдяки введенню практик DevSecOps, організаціям вдається виявляти більше вразливостей на ранніх етапах розробки. 71 % спеціалістів з безпеки зазначили, що щонайменше чверть усіх вразливостей у системі безпеки виявлялись розробниками[6]. Ці показники змушують організації будь-якого розміру впроваджувати практики DevSecOps у свої робочі процеси.

Як більшості з нас вже відомо DevSecOps — це методологія створення програмного забезпечення, яка поєднує принципи розробки (Dev), безпеки (Sec) та операційних процесів (Ops). Це досягається шляхом встановлення культури співпраці, автоматизації та постійного вдосконалення усіх команд. Кожен, хто залучений до розробки програмного забезпечення, відіграє роль у впровадженні безпеки в процеси безперервної інтеграції та безперервної доставки (CI/CD). Іншими словами: "Культура DevSecOps полягає в тому, що безпека повинна бути відповідальністю кожного".

Ключові принципи DevSecOps. Уявімо типовий сценарій розробки програмного забезпечення. Припустимо, продукт уже створений, протестований і готовий до випуску.

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Практики DevOps успішно застосовані, однак, перед остаточним релізом необхідно провести безпекові перевірки. Команда безпеки виявляє десятки вразливостей, багато з яких тісно інтегровані в функціонал і логіку додатку. Проект зупиняється, поки ці проблеми не будуть усунені. Що ще гірше, через те, що вразливості не були виявлені на ранніх етапах, їх усунення потребує суттєвих затрат часу і ресурсів для переробки значної частини коду.

Щоб уникнути несподіванок в останній момент, уповільнень проекту, модель DevSecOps визначила такі основні принципи [7]:

Автоматизація безпеки: Автоматизація процесів та тестування безпеки є основним принципом DevSecOps. Автоматизуючи завдання, такі як сканування вразливостей, аналіз коду та перевірки на відповідність, організації можуть забезпечити послідовні та надійні заходи безпеки протягом усього процесу розробки.

"Shift left": Інтеграцію безпеки починають розглядати якомога раніше в циклі створення програмного забезпечення. Такий підхід дозволяє організаціям досягти балансу між швидкістю, гнучкістю та надійними заходами безпеки, що призводить до створення більш безпечних і надійних програмних додатків.

Безперервний моніторинг: Постійний моніторинг та оцінка безпеки є ще одним ключовим принципом DevSecOps. Це включає моніторинг у реальному часі додатків у виробничих середовищах для виявлення будь-яких аномалій або потенційних загроз. Безперервний моніторинг дозволяє негайно реагувати на інциденти безпеки.

Співпраця: Співпраця між командами є важливою для того, щоб безпека була пріоритетом на всіх етапах розробки. Розробники, операційні команди та фахівці з безпеки повинні тісно співпрацювати для виявлення потенційних ризиків, впровадження відповідних заходів контролю та усунення будь-яких вразливостей, що можуть виникнути.

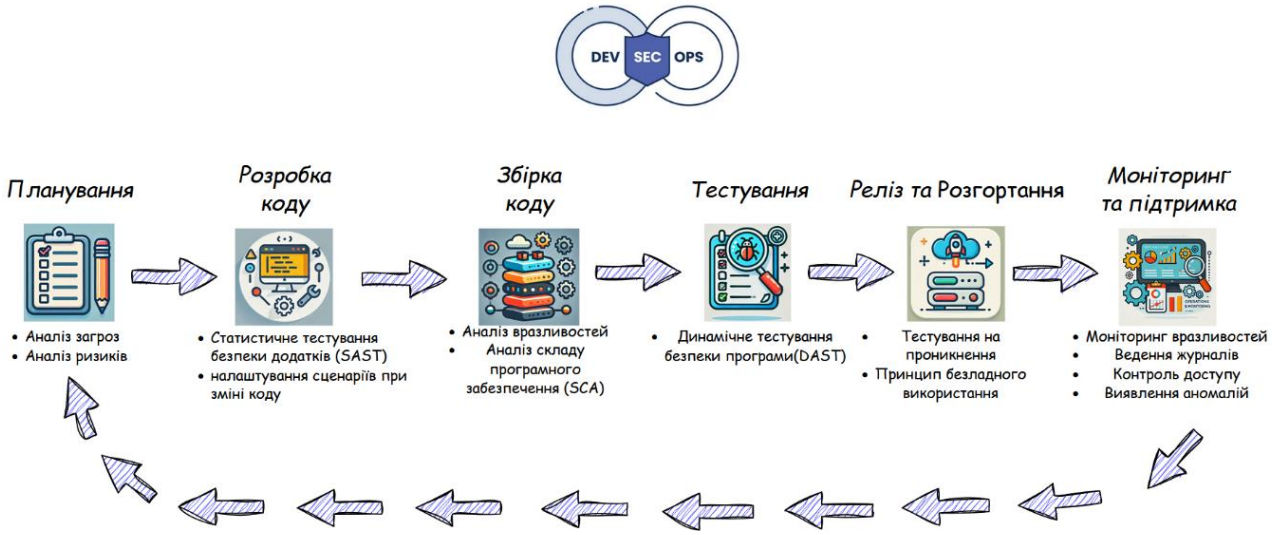

Рис. 1. Цикл розробки ПЗ з використанням підходу DevSecOps

Цикл розробки ПЗ. Спираючись на посібник з основ DevSecOps, виданий Департаментом Безпеки США, розробка програмного забезпечення *з* використанням методології DevSecOps включає [8]:

Планування. Є найменш автоматизованою фазою DevSecOps, що базується на співпраці, обговорені, огляді і розробці стратегії аналізу безпеки. Планування розробки програмного забезпечення включає планування управління конфігураціями, планування управління змінами, проектування системи, планування тестування та планування безпеки. Деякі інструменти планування будуть використовуватися протягом усього життєвого циклу програмного забезпечення: інструменти командної співпраці, система відстеження проблем і система управління проектами.

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Розробка. Дана фаза характеризується діяльністю, яка перетворює вимоги зазначені на етапі планування у вихідний код. До вихідного коду відносять: код застосунків, тестові сценарії, інфраструктуру як код (IaC), безпеку як код (SaC), сценарії автоматизації робочого процесу. Одним з найпростіших засобів безпеки на цьому етапі може бути уніфіковане середовище розробки (IDE). Функції IDE допомагають розробникам з автозавершенням коду, семантичним підкресленням та управлінням бібліотеками. Серед іншого варто виділити практики рецензування коду колегами або парне програмування, що також дозволяє покращити контроль якості коду. Також варто пам'ятати, що весь код, створений під час розробки, повинен бути закомічений у репозиторій вихідного коду та контролюватися версіями. Хоча це не вважається явним, важливо, щоб команди DevSecOps встановили чітку стратегію для проектування та створення компонентних артефактів програмного забезпечення, які містять нові або оновлені можливості, випущені через конвеєр CI/CD. Лише шляхом декомпозиції застосунку на окремий набір керованих сервісів можна правильно уникнути високого ризику монолітної розробки.

Збірка коду. Виконується побудова та пакетування додатків, сервісів і мікросервісів у артефакти. Найпопулярнішими інструментами на цьому етапі є інструменти статичного аналізу безпеки додатків (SAST), які сканують вихідний код на наявність помилок, вразливостей та недоліків; а також інструменти аналізу складу програмного забезпечення (SCA), що перевіряють пакетні менеджери, вихідний код та його залежності, бінарні файли й образи контейнерів.

Тестування. Дана фаза запускається після створення артефакту збірки та його успішного розгортання в середовищі тестування. Тестові заходи можуть включати, але не обмежуються, модульними тестами, функціональними тестами, інтеграційними тестами, системними тестами, регресійними тестами, приймальними тестами, тестами продуктивності та різними тестами безпеки. Виконання комплексного набору таких тестів займає значну кількість часу, тому необхідно прагнути до повної автоматизації даного процесу. Фаза тестування використовує інструменти динамічного тестування безпеки додатків (DAST) для виявлення реальної поведінки додатків, на такі події як автентифікація користувача, авторизація, SQL-ін'єкції та доступ через API. Також варто зазначити використання інструментів інтерактивного тестування безпеки додатків (IAST), що аналізує код на наявність вразливостей під час роботи додатку, коли його запускає автоматичний тест або будь-яка діяльність, що взаємодіє з функціональністю додатку.

Реліз та розгортання. Питання безпеки, які потрібно вирішити під час цієї фази, стосуються кінцевого середовища використання. Наприклад, будь-які відмінності в конфігурації між реальним середовищем розгортання і попередніми (середовище розробки та тестування) повинні бути ретельно перевірені. Сертифікати TLS і DRM повинні бути підтверджені та перевірені щодо їх валідності та часу життя. Фаза розгортання є хорошим часом для використання інструментів, які витягують інформацію з працюючої системи, щоб визначити, чи працює вона відповідно до очікувань. Організації можуть використовувати принципи хаос-інженерії, проводячи експерименти на системі, щоб підвищити впевненість у здатності системи витримувати складні умови. Крім того, фаза зосереджена на забезпеченні безпеки інфраструктури шляхом перевірки конфігураційних значень середовища, таких як контроль доступу користувачів, доступ до мережевого брандмауера та управління секретними даними. Для цього, часто використовують принцип найменших привілеїв (PoLP). Такий підхід означає, що будь-який користувач, програма або процес має мінімально необхідні права для виконання своїх функцій. Це включає аудит API ключів і токенів доступу, щоб їхні власники мали обмежений доступ.

Моніторинг та підтримка продукту. Компанії повинні відстежувати та підтримувати роботою додатка, щоб виявляти будь-які атаки, відхилення системи або витоки. Інструменти підтримки використовуються для масштабування системи, балансування навантаження та резервного копіювання. Розгортання застосунків повинно мати налаштовані належні політики

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

регулювання навантаження та масштабування. Якщо заданий поріг досягається або перевищується (наприклад, якщо використання пам'яті або центрального процесора перевищує попередньо встановлений поріг), система автоматично запускає дії оптимізації навантаження або масштабування. Щодо моніторингу, то використовуються інструменти для збору та оцінки ключової інформації про використання застосунку з метою виявлення тенденцій та визначення проблемних областей. Моніторинг охоплює базові апаратні ресурси, мережеву передачу даних, мікросервіси, контейнери, інтерфейси, нормальну та аномальну поведінку кінцевих точок, а також аналіз журналів подій безпеки.

Виклики для DevSecOps. Впровадження DevSecOps може бути складним для організацій, особливо на початкових стадіях [9]. Так в сучасних дослідженнях, серед проблем виділяють найбільш популярні:

Впровадження необхідних стандартів безпеки. Розробляючи програмне забезпечення, як нам впевнитись, що практики безпеки, які ввели, є достатніми і відповідають бізнес потребам? Наприклад, існують окремі стандарти для охорони здоров'я та фінансового сектору. Також, можуть існувати напрямки, в котрих немає потрібних напрацювань для створення достатнього рівня захисту.

Недостатній рівень знань та кооперації. Відсутність спільного розуміння та необхідних навичок серед різних учасників процесу розробки програмного забезпечення є серйозною перешкодою для успішного впровадження DevSecOps. Наприклад, розробники можуть допускати значно більшу кількість помилок в своєму коді, не маючи базових розумінь безпеки. В свою чергу, команди безпеки, без відповідної комунікації, втрачають велику кількість часу на аналіз та тестування усієї різноманітності технологій, які використовуються при розробці програмного забезпечення.

Нестача кадрів. Дослідження показують, що одними з найбільш затребуваних навичок у 2023 році є DevOps (35%), кібербезпека (16%), а також ШІ та машинне навчання (24%). Також підкреслюється, що лише 11% команд добре володіють методами безперервної інтеграції та безпеки [10].

Використання ШІ в кібербезпеці. Швидкий розвиток штучного інтелекту, особливо генеративного ШІ, змінює ландшафт кібербезпеки, пропонуючи як можливості, так і виклики. З одного боку, ШІ може значно зменшити навантаження і вирішити проблему нестачі фахівців у сфері кібербезпеки, автоматизуючи рутинні завдання, прискорюючи час реагування та покращуючи видимість на всій поверхні атаки. Однак, з іншого боку, широке впровадження ШІ у бізнес-процеси створює нові вразливості. Важливо розуміти, що імплементація системи штучного інтелекту мусить супроводжуватись відповідними регуляційними процесами.

Застосування штучного інтелекту в кібербезпеці не є новинкою. За даними дослідження EY Global Cybersecurity Leadership Insights Study 2023, з 2015 року спостерігається значне зростання досліджень, патентів та інвестицій, пов'язаних із ШІ у кіберзахисті. Наразі ШІ застосовується в 59% усіх кіберпатентів і з 2017 року є провідною технологією, яку досліджують у сфері кібербезпеки. Так, ключовим елементом ефективного та гнучкого підходу до кібербезпеки в провідних компаніях стає інтеграція ШІ – 62% з них вже використовують або перебувають на завершальній стадії впровадження штучного інтелекту чи машинного навчання (ML) [11].

Виходячи з досліджень можна виділити такі основні переваги:

Постійний розвиток. Можливості штучного інтелекту постійно зростають, оскільки він навчається на нових даних. Техніки, такі як глибоке навчання та машинне навчання, дозволяють AI розпізнавати нові шаблони, виявляти незвичні або підозрілі дії, що відхиляються від встановлених норм.

Покращення точності визначення загроз. Аналізатор ризиків, що працює на основі штучного інтелекту, може значно швидше й точніше обробляти дані та передбачати потенційні загрози майже в реальному часі. Так згідно з дослідженням *"Performance*

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity" – ШІ в середньому надає 90 % точність виявлення загроз [12].

Обробка великих обсягів даних. Згідно з *"Cloud Security Alert Fatigue Report"* від Orca Security: 59% організацій отримують понад 500 сповіщень безпеки на день, а 38% – понад 1000. 43% IT-керівників у цих організаціях сказали, що понад 40% сповіщень є помилковими, а 49% сказали, що понад 40% мають низький пріоритет. Незважаючи на те, що 56% респондентів витрачають більше 20% свого дня на перегляд сповіщень і вирішення, які з них слід розглянути в першу чергу, більше половини (55%) сказали, що їх команда пропускала критичні сповіщення, 41% зазначили, що такі сповіщення пропускаються щотижня, а 22% – щодня [13]. В даній ситуації, штучний інтелект може сканувати величезні обсяги даних, щоб ідентифікувати потенційні загрози та відфільтрувати незагрозливі дії.

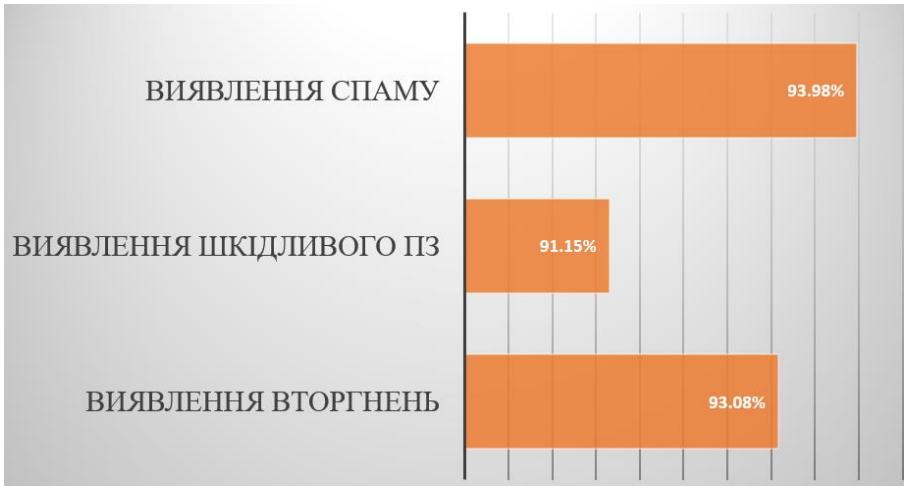

Рис. 2. Середня точність виявлення загроз використовуючи техніки машинного навчання

Таблиця 1

Проблеми безпеки, котрі вирішує ШІ в різних галузях

Галузь	**Опис**
Науки про життя	Для контролю нормальної роботи програмного забезпечення та виявлення аномального втручання в систему з боку зловмисників, ШІ та машинне навчання використовуються в імплантованих медичних пристроях (IMDs), таких як кардіостимулятори та інсулінові помпи.
Охорона здоров'я	Захист даних пацієнтів від зловмисників за допомогою ML і ШІ. Генеративний ШІ використовується для створення реалістичних, але синтетичних даних пацієнтів, щоб замінити реальні дані.
Енергетика	Тестування та покращення захищеності електросистем та електростанцій за допомогою ШІ, через апаратне моделювання в циклі (Hardware-in-the-loop (HIL)) та тести-симуляції.
Телекомунікації	Алгоритми виявлення проникнень на основі машинного навчання можуть краще захистити мобільні пристрої від зловмисників, які використовують техніки обфускації для проникнення в мобільні пристрої.
Фінансові послуги	Аналіз поведінки користувачів(UEBA) створює профілі та поведінкові моделі об'єктів, допомагаючи виявляти аномалії, які можуть бути оцінені за допомогою машинного навчання для виявлення загроз та потенційних інцидентів, спричинених людськими помилками.
Транспорт	Моніторинг вторгнень і аномальної поведінки на основі ШІ впроваджується в автономних транспортних засобах для захисту їхніх систем керування та компонентів управління.

Згідно з опитуванням проведеним в рамках дослідження IBM "Cost of a Data Breach Report 2024" [3]:

Використання AI в усіх аспектах кібербезепки значно знизило середні витрати на витік порівняно з організаціями, які не використовували штучний інтелект. Наприклад, коли організації широко використовували ШІ та автоматизацію, їхні середні витрати на витік даних становили 3,76 мільйона доларів США. Тим часом організації, які не використовували ці інструменти для профілактики, отримали 5,98 мільйонів доларів США витрат, що становить різницю на 45,6%.

Скрізь, де застосовувався штучний інтелект та автоматизація, прискорювалась робота з виявлення та локалізації порушень, в середньому цей час скоротився на 100 днів.

Широке використання зменшило середній MTTI (Mean time to investigate) та MTTC (Mean time to contain) для порушень даних на 33% для реагування та на 43% для запобігання.

Застереження щодо використання ШІ. Згідно з дослідженнями Gartner щодо цифрового ринку – "Майже всі організації прагнуть інвестувати в штучний інтелект — 92% опитаних компаній планують вкладати кошти в AI-інструменти у 2024 році, і 71% зазначають, що вони впевнені або дуже ймовірно зроблять це". За прогнозами Gartner, ринок програмного забезпечення на основі AI досягне вартості $297,9 мільярда до 2027 року, порівняно зі $124 мільярдами у 2022 році, зі щорічним середньорічним темпом зростання 19,1%. [14].

Оскільки організації поспішають використовувати ШІ, багато хто нівелює безпеку самих інструментів ШІ. Спираючись на дослідження Інституту бізнес цінностей при IBM, 70 % організацій вважають, що іноваційність рішення переважає над його правильним і безпечним імплементуванням [15]. Це призводить до того, що зловмисники вже націлюються на вразливості в системах штучного інтелекту, наприклад використовуючи ін'єкційні запити для атак на розмовні боти.

Таблиця 2

Топ-10 ризиків щодо Машинного навчання згідно з OWASP [16]

№	Вид	Опис
1	Маніпуляція вводом	Зловмисники навмисно змінюють вхідні дані, щоб обманути модель
2	Модифікація даних	Маніпуляція даними для навчання, щоб змусити модель діяти небажаним і непередбачуваним чином.
3	Інверсія моделі	Зловмисники здійснюють зворотне проектування моделі, щоб отримати з неї інформацію
4	Виявлення приналежності	Маніпуляція даними для навчання моделі з метою спричинення поведінки, яка розкриває конфіденційну інформацію
5	Крадіжка моделі	Отримання доступу до параметрів моделі
6	Атака на ланцюг поставок	Зміна або підміна компонентів, таких як платформи управління даними, ПЗ для управління моделями, хаби моделей, тощо.
7	Модифікація навчання	Навчання моделі для виконання певного завдання перед її додатковим налаштуванням для іншого завдання з метою викликати небажану поведінку.
8	Викривлення моделі	Зміна розподілу навчальних даних, щоб змусити модель поводитися небажаним чином.
9	Маніпуляція виводом	Зміна вихідних даних моделі для викликання небажаної або шкідливої поведінки в системі, яка її використовує.
10	Маніпуляція параметрами моделі	Маніпуляція параметрами моделі, щоб змусити її поводитися небажаним чином.

Крім атак зі сторони зловмисників, необхідно звернути увагу на обізнаність організацій щодо вимог безпечної імплементації штучного інтелекту. Згідно з дослідженням від Orca Security, статистика безпеки розгорнутих моделей штучного інтелекту в хмарних сервісах та середовищах показує, що [17]: 27% організацій котрі використовують не налаштували приватний доступ кінцевих точок Azure OpenAI. Це підвищує ризик того, що зловмисники можуть отримати доступ, перехопити або змінити дані, які передаються між хмарними ресурсами та AI-сервісами; 20% організацій, котрі використовують OpenAI, мають принаймні один ключ доступу, збережений у незахищеному місці; 45% кошиків для Amazon SageMaker

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

використовують стандартну схему іменування; 98 % організацій, котрі використовують Amazon SageMaker, мають віддалену машину з увімкненим root-доступом; 62% організацій, розгортали AI-пакет, який містив принаймні одну вразливість CVE; 98% організацій, які використовують Google Vertex AI, не увімкнули шифрування для своїх власних ключів, що зберігаються; 81% викритих ключів доступу до OpenAI зберігалися в історії комітів репозиторію.

Серед основних проблем з котрими стикаються при налаштуванні безпеки для штучного інтелекту визначають наступні:

Швидкість прогресу: Швидкий розвиток AI зосереджений на полегшенні використання, часто нехтуючи питаннями безпеки.

Тіньовий ШІ: Команди безпеки не мають повної видимості щодо використання ШІ в організаціях.

Контроль ресурсів: Неправильні конфігурації ресурсів при запуску нових сервісів, включаючи налаштування ролей, бакетів, користувачів та інших активів, створюють значні ризики для середовища.

Дотримання нормативних вимог: Швидкозмінність вимог регуляторних стандартів вимагає балансу між інноваціями, безпекою та відповідністю новим правовим нормам.

Поради щодо захисту ШІ. Завдяки ретельному управлінню інтеграцією AI, організації можуть використовувати його можливості для досягнення вищих результатів у безпеці, знижуючи при цьому властиві ризики використання систем на основі ШІ. Спираючись на дослідження від IBM [15] та ORCA Security [17], можна вивести наступні поради:

Оновлення стратегій управління, ризиків та відповідності стандартам (GRC). Підхід, орієнтований на управління та дизайн, особливо важливий для генеративного AI з огляду на нові регуляторні рекомендації: EU AI Act, National AI Initiative Act (США), AI and Data Act (Канада), AI Governance in Japan, Ethical Norms for New Generation AI (Китай).

Захист інфраструктури та процесів для розробки ШІ.

Остерігайтеся налаштувань за замовчуванням. Хмарні AI-сервіси, орієнтовані на потреби розробників, пропонують функції та налаштування для підвищення ефективності. Це часто означає наявність налаштувань за замовчуванням, які можуть створити ризики для безпеки в робочому середовищі.

Контроль вразливостей. Хоча сфера безпеки AI є відносно новою, більшість вразливостей такими не є. AI-сервіси часто використовують існуючі рішення з відомими вразливостями.

Обмеження привілеїв. Надмірні привілеї надають зловмисникам свободу переміщення та можливість здійснення багатофазних атак у разі отримання початкового доступу.

Захист даних. Захист даних вимагає поєднання кількох найкращих практик. Це включає використання власних ключів шифрування та їх захист при збереженні.

Моніторинг. Проводьте регулярні аудити безпеки, тестування на проникнення та вправи з використанням червоних команд для виявлення та усунення потенційних вразливостей у середовищі AI та пов’язаних додатках.

Навчання персоналу. Перегляньте практики кібергігієни та основи безпеки (усвідомлення, поведінка та культура) у вашій організації. Проводьте навчання та активності з підвищення обізнаності про кібербезпеку, орієнтовані на конкретні ролі, особливо пов’язані з ШІ як новою поверхнею загроз. Охопіть усіх зацікавлених осіб, залучених до розробки, впровадження та використання AI-моделей, включаючи працівників, які використовують інструменти на основі штучного інтелекту.

Кожна порада включає різні рівні інвестицій, зобов’язань і відповідальності. Хоча деякі організації вже взяли стратегію впровадження, деякі застосовують декілька, а деякі все ще знаходять свій шлях і формалізують свою стратегію. З точки зору безпеки, кожен варіант залежить від того, хто за що відповідає та як ця відповідальність може бути розподілена.

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Використання ШІ для DevSecOps. DevSecOps із підтримкою штучного інтелекту – це вдосконалене втілення підходу DevSecOps, який інтегрує штучний інтелект у розробку, безпеку та операційну систему. Ця інтеграція надає процесу розробки можливості, які можуть обробляти величезні обсяги інформації, вивчати шаблони даних і автоматизувати процеси прийняття рішень. Результатом є динамічна система, у якій інструменти штучного інтелекту допомагають виявляти потенційні проблеми, оптимізувати операції розробки та забезпечувати надійні заходи безпеки протягом усього життєвого циклу розробки програмного забезпечення.

Об'єднання штучного інтелекту в DevSecOps пропонує безліч переваг, які переводять процес розробки програмного забезпечення в нову сферу ефективності та безпеки:

Покращення безпеки. Алгоритми штучного інтелекту чудово виявляють вразливості та потенційні загрози шляхом постійного аналізу змін коду та потоків даних. Це дозволяє виявляти та зменшувати ризики, гарантуючи, що безпека є проактивним аспектом SDLC.

Підвищення ефективності виконання дій. Автоматизація на основі штучного інтелекту може виконувати повторювані та трудомісткі завдання, від перевірки коду до налаштування середовища, звільняючи людські ресурси для зосередження на більш складних і критичних завданнях.

Прогнозована аналітика. За допомогою машинного навчання ШІ може передбачати майбутні результати на основі історичних даних. У контексті DevSecOps це означає передбачення системних збоїв, стрибків робочого навантаження та навіть поведінки користувачів.

Динамічна адаптація. Системи штучного інтелекту за своєю суттю створені для адаптації та розвитку. У контексті DevSecOps це означає безперервне вдосконалення процесів, оскільки штучний інтелект вивчає нові дані та взаємодії, тим самим постійно вдосконалюючи робочий процес.

Забезпечення якості. Завдяки здатності ШІ аналізувати та тестувати програмне забезпечення на кожному етапі розробки якість кінцевого продукту значно покращується. AI може швидко виявляти дефекти або області, які потрібно вдосконалити, забезпечуючи відповідність кінцевого продукту найвищим стандартам якості.

Оптимізація ресурсів. Штучний інтелект може надавати інформацію про ефективніше використання ресурсів, оптимізацію хмарної інфраструктури для економії коштів чи розподіл людських ресурсів там, де вони можуть бути найбільш ефективними.

Аналіз сучасних досліджень, щодо ШІ в DevSecOps. На сьогоднішній день існує безліч наукових напрацювань, щодо використання можливостей штучного інтелекту для досягнення цілей DevSecOps. За даними бази Google Scholar, кількість таких робіт становить близько 2000.

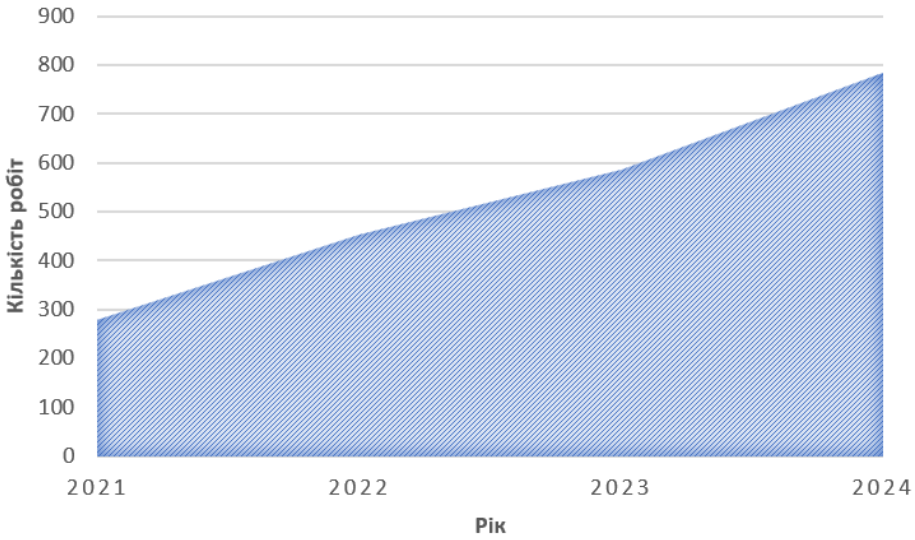

Рис. 3. Статистика кількості міжнародних робіт в напрямку "AI in DevSecOps"

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Згідно з Рис. 3 та статистичними даними міжнародних компаній [1, 2, 3], інтерес до досліджень у цьому напрямку продовжує зростати. Водночас слід зазначити, що значна частина цих досліджень спрямована лише на покращення безпеки на окремих етапах DevSecOps, тоді як комплексний підхід із використанням ШІ набув популярності недавно [4].

Проаналізувавши доступні дослідження можна виділити наступні:

"Log-based anomaly detection without log parsing". NeuralLog – це новий підхід до виявлення аномалій, який забезпечує ефективне та результативне виявлення аномалій на реальних наборах даних. На відміну від існуючих методів, NeuralLog не потребує парсингу журналів, запобігаючи таким чином втраті інформації через помилки аналізу. Кожне повідомлення журналу безпосередньо перетворюється у семантичний вектор, який здатен захоплювати як семантичну інформацію, вбудовану у повідомлення журналів, так і зв'язок між ними. Далі, приймаючи послідовність семантичних векторів як вхідні дані, для виявлення аномалій застосовується модель класифікації на основі трансформерів [18].

"Dataflow Analysis-Inspired Deep Learning for Efficient Vulnerability Detection". Досліджується ідея поєднання алгоритмів аналізу потоків даних (DFA) з глибинним навчанням для створення невеликих, але ефективних моделей для виявлення вразливостей. DFA обчислює шаблони використання даних та зв'язки у графі управління потоком і визначає вразливість на основі її першопричини [19].

"Within-project defect prediction of infrastructure-as-code using product and process metrics". Робота спрямована на аналіз дефектів в сценаріях Інфраструктури як коду (IaC) завдяки прогнозуванню помилок в коді та дослідженні метрик кінцевого продукту і процесу його створення. Даний процес здійснюється за допомогою підходу RADON для прогнозування дефектів IaC [20].

"KGSecConfig: A Knowledge Graph Based Approach for Secured Container Orchestrator Configuration". Дане дослідження фокусується на вирішенні проблеми захищеного конфігурування контейнерезованих застосунків. Запропоновано підхід KGSecConfig, який базується на машинному навчанні та використовує графи знань (KG) для систематичного збору, зв'язування і кореляції даних з метою автоматизації процесу налаштувань безпеки контейнерних оркестраторів [21].

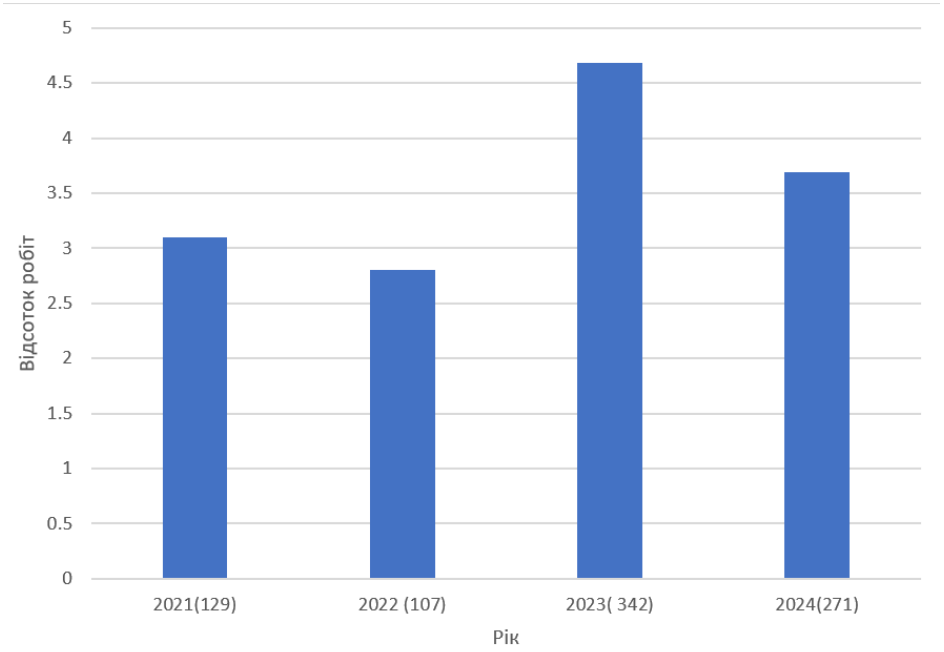

Рис. 4. Кількість українських досліджень пов'язаних з SDLC/DevOps/DevSecOps серед досліджень на тематику ШІ та захисту інформації згідно з Google Scholar

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

Не зважаючи на значну кількість міжнародних досліджень, база українських наукових робіт в цьому напряму є досить незначною. Згідно з аналізом праць на платформі *Google Scholar*, розвиток ШІ та DevSecOps серед українських наукових робіт займає близько 4 %, а в базі Національної бібліотеки України імені В.І. Вернадського, було знайдено тільки роботи, які б пов'язували штучний інтелект та захист інформації.

Таблиця 3

Результати пошуку досліджень в базі Національної бібліотеки України імені В.І. Вернадського

Запит	Кількість публікацій
ключове слово "DevOps"	5
ключове слово "DevSecOps"	2
Ключове слово "Штучний інтернет"	116 (з них пов`язаних з захистом інформації: 7)

Висновки

Штучний інтелект щоденно змінює підходи до кібербезпеки, особливо у DevSecOps, де безпека відіграє важливу роль у розробці програмного забезпечення та операціях. Здатність ШІ автоматизувати тестування, покращувати аналіз коду, прискорювати оновлення та забезпечувати швидке реагування на інциденти значно посилює загальний рівень безпеки.

У статті проаналізовано використання штучного інтелекту в кібербезпеці та методології DevSecOps. Підкреслюється важлива роль у об'єднанні безпеки з розробкою та операційною діяльністю, акцентуючи на підході, де безпека є спільною відповідальністю на всіх етапах життєвого циклу програмного забезпечення.

Проаналізовано інтерес щодо досліджень у цьому напрямку. Згідно з статистикою, кількість праць за 2024 складає більше третини робіт цієї тематики за період 2021-2024 року. Водночас значна частина цих досліджень зосереджується на покращенні безпеки окремих етапів DevSecOps, тоді як комплексний підхід із використанням ШІ став популярним лише нещодавно. Варто зазначити, що, попри велику кількість міжнародних досліджень, українські наукові роботи в цій галузі обмежені та становлять близько 4 % від усіх вітчизняних досліджень у сфері ШІ та захисту інформації.

Слід також вказати, що штучний інтелект у контексті кібербезпеки та DevSecOps є новою технологією, яка має багато невідомих і обмежень. Тому організаціям рекомендується впроваджувати рішення на основі ШІ обережно, з чітким розумінням можливостей і обмежень. Оптимальним підходом є поступове навчання ШІ чистому коду, новітнім практикам тестування та методам виявлення кіберзагроз, а також інтеграція політик (EU AI Act) та структурних рішень команд у процес навчання ШІ.

Перелік посилань

1. 2024 global devsecops report. Gitlab. URL: https://about.gitlab.com/developer-survey/ (дата звернення 01.10.2024).
2. Global state of devsecops 2024. Black Duck. URL: https://www.blackduck.com/resources/analyst-reports/state-of-devsecops.html (дата звернення 26.10.2024).
3. Cost of a data breach report 2024. IBM. URL: https://www.ibm.com/reports/data-breach. (дата звернення 26.10.2024).
4. Fu M., Pasuksmit J., Tantithamthavorn C. AI for DevSecOps: A Landscape and Future Opportunities. 2024. URL: https://doi.org/10.48550/arXiv.2404.04839 (дата звернення 01.10.2024).
5. DevSecOps Market Report Scope & Overview. SNS Insider. URL: https://www.snsinsider.com/reports/devsecops-market-2416. (дата звернення 27.10.2024).
6. 2023 Global DevSecOps Report. Gitlab. URL: https://about.gitlab.com/developer-survey/previous/2023/. (дата звернення 26.10.2024).

© Гавриляк, В. Р., & Опірський, І. Р. (2024). Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці. Сучасний захист інформації, 4(60), 73–84. https://doi.org/10.31673/2409-7292.2024.040008.

7. Development, Security, and Operations: A Brief Guide on DevSecOps. USCSI. URL: https://www.uscsinstitute.org/cybersecurity-insights/blog/development-security-and-operations-a-brief-guide-on-devsecops. (дата звернення 20.10.2024).

8. DevSecOps Fundamentals Guidebook: DevSecOps Tools & Activities. 2-ге вид. Defense Office of Prepublication and Security Review, 2021. URL: https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsTools-ActivitiesGuidebook.pdf (дата звернення 20.10.2024).

9. Yankel J., Yasar H. 5 Challenges to Implementing DevSecOps and How to Overcome Them. 2023. URL: https://doi.org/10.58012/fywc-yq50 (дата звернення 20.10.2024).

10. Nastenko V. Integrating Security in DevOps: Best Practices, Tools, and Challenges. URL: https://tech-stack.com/blog/integrating-security-in-devops-best-practices-tools-and-challenges/. (дата звернення 28.10.2024).

11. How can cybersecurity transform to accelerate value from AI? / R. Watson та ін. EY. URL: https://www.ey.com/en_gl/insights/consulting/transform-cybersecurity-to-accelerate-value-from-ai. (дата звернення 28.10.2024).

12. Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity / K. Shaukat et al. Energies. 2020. Т. 13, № 10. 2509. URL: https://doi.org/10.3390/en13102509 (дата звернення: 20.11.2024).

13. 2022 Cloud Security Alert Fatigue Report. Orca Security. URL: https://orca.security/wp-content/uploads/2022/03/Orca-2022-Cloud-Security-Alert-Fatigue-Report.pdf (дата звернення: 20.11.2024).

14. Jain A. 4 AI Software Trends Shifting Buying Behavior. Gartner Digital Markets. URL: https://www.gartner.com/en/digital-markets/insights/ai-software-trends. (дата звернення: 20.11.2024).

15. Securing generative AI: What matters now. IBM. URL: https://www.ibm.com/thought-leadership/institute-business-value/en-us/report/securing-generative-ai (дата звернення: 20.11.2024).

16. OWASP Machine Learning Security Top Ten | OWASP Foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/www-project-machine-learning-security-top-10/ (дата звернення: 20.11.2024).

17. 2024 State of AI Security Report. Orca Security. URL: https://orca.security/lp/2024-state-of-ai-security-report/#the-report. (дата звернення: 22.11.2024).

18. Le V.-H., Zhang H. Log-based Anomaly Detection Without Log Parsing. 2021. URL: https://doi.org/10.48550/arXiv.2108.01955. (дата звернення: 29.11.2024).

19. Steenhoek B., Gao H., Le W. Dataflow Analysis-Inspired Deep Learning for Efficient Vulnerability Detection. ICSE '24: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. 2024. С. 1–13. URL: https://doi.org/10.1145/3597503.3623345 (дата звернення: 02.11.2024).

20. Within-Project Defect Prediction of Infrastructure-as-Code Using Product and Process Metrics / S. Dalla Palma та ін. IEEE Transactions on Software Engineering. 2022. Т. 48, № 6. С. 2086–2104. URL: https://doi.org/10.1109/TSE.2021.3051492. (дата звернення: 02.11.2024).

21. Mubin Ul Haque, Kholoosi M. M., Babar M. A. KGSecConfig: A Knowledge Graph Based Approach for Secured Container Orchestrator Configuration. 2021. URL: https://doi.org/10.48550/arXiv.2112.12595 (дата звернення: 02.11.2024).

22. OWASP DevSecOps Guideline | OWASP Foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/www-project-devsecops-guideline/ (дата звернення: 19.10.2024).

23. Javed A. What is DevSecOps and why is it so important?. IBM. URL: https://developer.ibm.com/articles/devsecops-what-and-why/ (дата звернення: 19.10.2024).

Надійшла 25.11.2024