УДК 004.432
DOI: 10.31471/1993-9965-2023-2(55)-61-69

ОГЛЯД МОВ ПРОГРАМУВАННЯ У РАКУРСІ КІБЕРБЕЗПЕКИ

М. В. Крихівський, Т. О. Ваврик*, Л. М. Гобир

ІФНТУНГ; 76019, м. Івано-Франківськ, вул. Карпатська, 15; тел. (0342) 727139;
e-mail: t e t i a n a . v a v r y k @ n u n g . e d u . u a

У сучасному цифровому світі, де обмін даними та інформацією швидко розширюється, організація безпеки стає надзвичайно важливою задачею. Метою розроблення найпопулярніших мов програмування, які використовуються сьогодні, не був захист інформації. Програмісти, що їх використовують, повинні скласти власні підходи до запобігання недолікам безпеки та виправленню вразливостей. У цій роботі досліджується кібербезпека як властивість мов програмування. Розуміючи джерела загальних вразливостей інформаційних систем, варто побудувати мови, які можуть запобігти деяким з них. Зокрема, досліджено конструкції та властивості мов, які можуть запобігти або зменшити наслідки кібератак. Розглянуто популярні мови програмування, які використовуються для створення програмних продуктів у ракурсі кібербезпеки. Проаналізовано їх основні особливості, можливості та інструменти, що допомагають забезпечити надійність алгоритмів і захищеність даних. Зроблено огляд таких мов програмування, як Rust, Julia , Nim, Ruby, Roota, і проаналізовано, які інструменти та бібліотеки вони пропонують для захисту від загроз безпеки. Також запропоновано певні практики програмування, які можуть допомогти забезпечити безпеку даних, такі як валідація вхідних даних, шифрування та перевірка аутентичності. Дослідження включає огляд основних вразливостей, які можуть бути пов'язані з використанням кожної мови, а також прийоми та методи безпеки, що можуть бути використані для зменшення ризиків. Також описано заходи, які треба використовувати для запобігання вразливостям і зловмисним атакам. Розглянуто мови програмування для різних платформ та проаналізовано їх переваги та недоліки, з точки зору безпеки. Метою статті є дослідження можливостей розробників програмних продуктів для забезпечення від кібератак на етапі проєктування та вироблення рекомендацій для типових структур даних і алгоритмів. На основі проведеного аналізу робиться висновок про те, що вибір мови програмування має важливе значення для забезпечення безпеки програмного продукту. Розробники повинні уважно вибирати мову програмування та використовувати найкращі практики безпеки для запобігання вразливостям та забезпечення безпеки розроблюваного програмного продукту.

Ключові слова: мови програмування, кібербезпека, безпека інформаційних систем, проєктування програмних продуктів, інженерія програмного забезпечення.

In today's digital world, where the exchange of data and information is rapidly expanding, the organization of security becomes an extremely important task. The most popular programming languages used today were not designed to protect information. Programmers using them must develop their own approaches to preventing security flaws and fixing vulnerabilities. This work examines cyber security as a property of programming languages. By understanding the sources of common vulnerabilities in information systems, it is worth building languages that can prevent some of them. In particular, the constructions and properties of languages that can prevent or reduce the consequences of a cyberattack are investigated. Popular programming languages used to create software products from the perspective of cyber security are considered. Their main features, capabilities and tools that help ensure the reliability of algorithms and data security are analyzed. It reviews programming languages such as Java, Python, C++, and others, and analyzes what tools and libraries they offer to protect against security threats. Certain programming practices are also suggested that can help ensure data security, such as input validation, encryption, and authentication. The latest developments and trends in the field of programming security are described, as well as the measures that should be used to prevent vulnerabilities and malicious attacks. Programming languages for different platforms are considered and their advantages and disadvantages from the point of view of security are analyzed. The goal of this article is to investigate software product developers' ability to protect against cyberattacks during the design stage, as well as to develop recommendations for common data structures and algorithms.Based on the findings, it is determined that programming language selection is critical for assuring software security. Developers should carefully select their programming language and follow security best practices to avoid vulnerabilities and assure the security of the software they create.

Keywords: programming languages, cyber security, security of information systems, design of software products, software engineering.

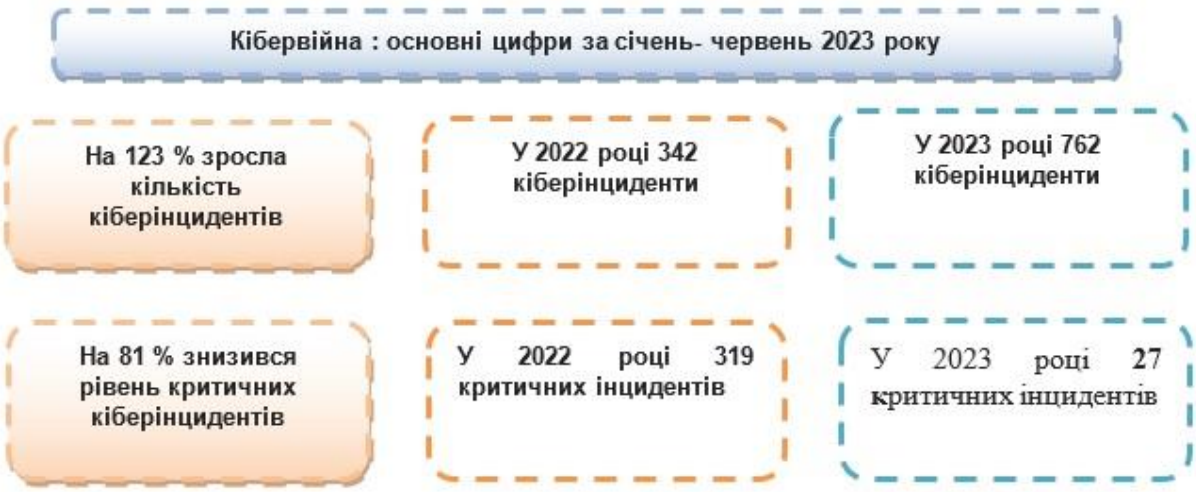

Рисунок 1 – Дані з аналітичного звіту Держспецзв'язку за підтримки USAID EU4PAR про кібератаки [1]

Вступ

У сучасному цифровому світі, де технології швидко розвиваються й інформація стає все більш доступною, особиста безпека стає надзвичайно важливою для кожного користувача Інтернету. Це особливо актуально для України. Від початку війни з боку рф втричі зросла кількість кібератак проти України (рис. 1).

Вороги намагаються отримати будь-яку інформацію, яка може дати перевагу у війні проти нашої країни. На думку міжнародних експертів, країна, яка здійснює безпрецедентні кібератаки, ще не застосувала найбільшою мірою всі свої можливості.

Тому без наявного плану дій, засобів моніторингу, сформованої системи захисту інформації підприємств, держструктури споживачі потрапляють під ризик інформаційних витоків, пошкодження інформації, простою, втрат доступу до серверів і послуг при ймовірних актуальних чи нових кіберзагрозах.

Необхідно формувати, реалізовувати та вдосконалювати підходи та принципи здійснення інформаційної безпеки, аналізувати інформацію про події в сучасному кіберпросторі, оцінювати актуальність захисної системи.

Аналіз сучасних закордонних і вітчизняних досліджень та публікацій

У контексті вивчення та вирішення наукових прикладних проблем кібербезпеки слід зазначити, що багато наукових установ світу й України присвячують свої теоретичні та прикладні дослідження розробленню цього наукового напрямку. Важливим кроком було прийняття Закону України «Про основні засади кібербезпеки України». Відповідно до нього, кібербезпека – це захист життєво важливих інтересів особистості та громадянина, суспільства та держави при використанні кіберпростору, який забезпечує сталий розвиток інформаційного суспільства та цифрового комунікаційного середовища, своєчасне виявлення, запобігання та нейтралізацію реальних та потенційних загроз для національної безпеки України в кіберпросторі [2].

Детальний аналіз та розуміння динаміки кіберзагроз у контексті інформаційної війни, в стані якої знаходиться Україна, представлено в статті [3]. У ній також висвітлюються ефективні стратегії та заходи для забезпечення кібербезпеки в такому складному середовищі. Результати цього дослідження розкрили основні особливості інформаційних війн та їх вплив на кібербезпеку в глобальному масштабі та в окремих організаціях і державах.

Серед науковців, які займаються кібербезпекою, хочемо виокремити грецьких вчених Павла Хеймонідіса та Константіноса Рантоса. У їхній статті [4] розглянуто традиційні методології та стандарти оцінки ризиків інформаційної безпеки (RA), запропоновано моделі динамічної оцінки ризику (DRA) для постійної та динамічної оцінки ризиків інформаційної безпеки в (майже) реальному часі. Проаналізовано області застосування моделей та інформацію, яку вони використовують для отримання результатів. Дано відповідь на важливі дослідницькі питання щодо розвитку динамічного ризику та методологій його оцінки. Проаналізовано вже розроблені методи оцінки, а також майбутні напрямки досліджень. У цьому напрямі важливо правильно вибирати систему програмування як для тестування існуючого захисту інформаційної безпеки, так і для модифікації з метою вдосконалення структури кіберзахисту.

ISSN 1993–9965 print
ISSN 2415–3524 online
Науковий вісник ІФНТУНГ
2023. № 2(55)

Із зростаючими проблемами кібербезпеки для організацій і окремих користувачів через зростання кількості та складності загроз, включаючи появу кіберзагроз на основі штучного інтелекту (ШІ), підвищується важливість оцінок кібербезпеки. Оцінка безпеки – це процес визначення поточного стану кібербезпеки інформаційної системи й оцінки виконання цілі безпеки. Якщо її виконувати методологічно, то це гарантує, що критичні кіберасети будуть захищені від загроз, та основна комунікаційна інфраструктура не призведе до збоїв та не сприятиме вторгненню. Відсутність оцінки кібербезпеки перешкоджає розгортанню нових технологій у цільових галузях. Крім того, регулярне проведення оцінок кібербезпеки стає все більш необхідним для урядів та галузевих регуляторів, що, зокрема, стосується критичних секторів [5]. Це спонукає до пошуку ефективних та застосовних методів оцінки кібербезпеки та вибору ефективних мов програмування для їх реалізації.

Процес оцінки й обробки ризиків є найважливішим для впровадження ефективної програми кібербезпеки та відіграє вирішальну роль для національних і міжнародних правил у сфері захисту даних. Проте визначення надійних моделей кіберризику залишається відкритою проблемою. Кількісні підходи, зазвичай базуються на системах оцінки, які пов'язують певну оцінку з технологічним організаційним контекстом. Проте, спосіб, у який вона зазвичай реалізується, не дає реалістичної оцінки кіберризику та пов'язаного з ним впливу [5]. Виходячи з цих передумов, чітко постає необхідність у покращенні кількісної оцінки кіберризиків організації шляхом динамічного моніторингу атак і вразливостей, яких зазнає організація, що зумовлює необхідність вибору адекватної (повинні бути доступні готові інструменти) мови програмування.

Протягом десятиліть у сфері кібербезпеки було запропоновано різні методи оцінки її стану в інформаційних системах. Особливу увагу слід приділяти питанню застосовності методів у реалістичних контекстах і середовищах. У зв'язку з цим у роботі [6] було зазначено проблеми та обмеження, пов'язані із застосуванням методів, а також можливі підходи до їх вирішення. Крім того, у цій статті систематизовано термінологію та вказано додаткові дослідження, які можуть бути корисними під час оцінювання.

Порівняння мови програмування Python з найпопулярнішими мовами програмування (C++, C#, Java та JS) для розв'язання задач кібербезпеки та захисту інформації виконано у статті [7]. Тут досліджено можливості для розробки додатків аналізу мережі та даних, інструментів захисту, інструментів для тестування, автоматизації задач та веб-розробки. Наведені авторами приклади коду на Python та JS для розробки шифраторів даних та результати аналізу мережі, а також описаний фреймворк для тестування безпеки OWASP ZAP на Python показують [7], що «Python є потужним інструментом для розв'язання завдань кібербезпеки та захисту інформації, здатним конкурувати з іншими мовами програмування загального призначення».

Мета та завдання досліджень

Надважливим ми вважаємо дослідження кібербезпекових можливостей алгоритмічних мов програмування, які закладають основи проєктування безпечних програмних продуктів. Головним завдання цього дослідження є виявлення слабких та сильних сторін мов програмування у ракурсі кібербезпеки.

Виклад основного матеріалу дослідження

Загрози інформаційній безпеці виникають у багатьох формах і загалом їх можна поділити на декілька видів:

- Стихійні лиха – це загрози, що викликані якимись особливими обставинами, природними або спричиненими людиною, а саме, аварії, пошкодження цілісності, стихійні та техногенні катастрофи, пожежі, повені, землетруси, урагани, епідемії тощо.
- Техногенна антропогенна загроза – це загроза, яка безпосередньо стосується діяльності людей. До них належать неправдива інформація, шпигунство, вербування, хабарництво, здирництво, крадіжка, розголошення або введення в оману, будь-які людські помилки та необережна поведінка. Організаторами техногенних загроз можуть бути різні групи: диверсанти, екстремісти, злочинні організації, шахраї, зловмисники, недобросовісні партнери, співробітники спецслужб тощо. Подібні загрози можуть бути ненавмисними, і в цьому випадку їх джерелом можуть бути: оператори, співробітники, технічний або обслуговуючий персонал тощо.
- Апаратна загроза – загроза, яка ініціюється за допомогою технічних пристроїв. До них відносяться: електромагнітне випромінювання, пошкодження, відключення, прослуховування, відеоспостереження, незаконні підключення тощо.

- Програмне забезпечення інколи є загрозою. Це, наприклад: злом, ін'єкція, атака, перехоплення, відмова в обслуговуванні (DoS/DDoS), зловмисне програмне забезпечення (Malware), помилки коду, збої системи, зависання, пошкодження та втрата даних, неправильні дозволи доступу, неправильна конфігурація тощо.

Основними напрямками кібербезпеки є:

- захист додатків і програмного забезпечення (Application Security);
- захист веб-сайтів і електронних ресурсів (Web Application Security);
- захист мобільних пристроїв, додатків і систем (Mobile Security);
- захист комп'ютерних мереж (Network Security);
- захист об'єктів критичної інфраструктури (SCADA Security);
- захист інтернету речей (IoT Security).

Зловмисники постійно шукають нові способи вторгнутися у комп'ютерні системи та отримати незаконний доступ до цінної інформації. Тому для забезпечення безпеки в Інтернеті розробники постійно створюють нові інструменти та технології.

Одним з інноваційних підходів до безпеки є «проактивний» підхід. Цей підхід передбачає вживання заходів і запобіжних дій, які надають захист ще до виникнення загрози або небезпеки. Одним з ключових аспектів проактивного підходу до безпеки є обізнаність. Важливо мати глибоке розуміння ризиків і загроз, з якими можна стикнутися в цифровому просторі. Це дозволяє розробникам ефективно працювати над запобіганням проблем із безпекою, а також планувати відповідні заходи безпеки. Другим важливим елементом є постійне оновлення й удосконалення. Проактивний підхід до безпеки передбачає постійний моніторинг нових загроз і вразливостей. Команда інженерів повинна бути готова реагувати швидко й ефективно на будь-які нові проблеми з безпекою, які можуть виникнути.

Проактивний підхід допомагає досліджувати та готуватися до найнебезпечніших кібератак, безпосередньо не стикаючись з ними. Він ще називається Shift to Left, адже ми зміщуємо фокус лівіше – від моменту атаки, до її планування.

Елементами проактивного підходу до безпеки можна вважати:

– вивчення нових кібератак;
– актуалізація зміни рішень безпеки;
– Threat Hunting.

Результатом проактивного підходу до безпеки є використання мов програмування, які мають вбудовані функції й інструменти для захисту даних. Розвиток кіберзахисту вимагає постійного вдосконалення технологій та інструментів. Однією з важливих складових такого підходу є розроблення нових мов програмування та інструментів для кіберзахисту. При створенні програм необхідно враховувати потенційні загрози та ризики, що можуть порушити конфіденційність, цілісність та доступність даних. Особливу увагу слід приділяти вибору мови програмування, оскільки вона може чинити суттєвий вплив на рівень безпеки створеного програмного продукту.

Розглянемо мови програмування, що вже широко використовуються та мають потенціал для використання в кіберзахисті.

Rust. Мова програмування Rust є популярною системною мовою, яка поєднує в собі високу продуктивність з гарантіями безпеки та надійності. Розроблена компанією Mozilla, Rust набуває все більшої популярності серед розробників через свої унікальні особливості. Однією з ключових особливостей Rust є система власності, яка дозволяє досягти високої безпеки пам'яті та уникнути багатьох типових помилок, таких як нульові вказівники та переповнення буферу. Завдяки контролю власності Rust забезпечує автоматичне управління пам'яттю без потреби в збиранні сміття, що дозволяє підтримувати високу продуктивність і ефективність коду. Ще однією перевагою Rust є її потужна система типів, яка дозволяє виявляти баги на етапі компіляції та забезпечує статичну безпеку типів. Валідатор типів Rust забезпечує виявлення багів та недоречностей ще до виконання програми, що робить розробку більш надійною та простішою. Rust також має екосистему розширень та бібліотек, які допомагають забезпечити гнучкість та повторне використання коду. Існує багато розроблених сторонніх бібліотек та інструментів, які спрощують розробку різних проєктів, від веб-розробки до системного програмування. Загалом, мова програмування Rust позиціонує себе як інструмент для розробників, які цінують якість і безпеку коду, безпеку пам'яті та високу продуктивність. Rust є чудовим вибором для проєктів, де надійність та ефективність є критичними факторами. Ця мова набуває все більшої популярності серед розробників.

Rust відома своєю безпекою, швидкодією та здатністю до очищення типів даних. Мова програмування Rust може допомогти уникнути багатьох типових помилок, які присутні в ін-

ISSN 1993–9965 print ISSN 2415–3524 online
Науковий вісник ІФНТУНГ 2023. № 2(55)

ших мовах, таких як переповнення буфера, використання пам'яті після звільнення та інші. Також Rust має багато інструментів для кіберзахисту, наприклад, бібліотеки для роботи з шифруванням, криптографією, мережевою безпекою та багато інших [8].

Julia. Мова програмування Julia є високорівневою, динамічною мовою програмування загального призначення, спеціально розробленою для наукових обчислень та чисельного аналізу. Julia пропонує простий та експресивний синтаксис, наближений до синтаксису математичних формул, що робить код простим та легким для розуміння. Однією з головних переваг Julia є його швидкодія. Julia володіє потужною системою компіляції Just-In-Time (JIT), яка перетворює код у високоефективний машинний код, забезпечуючи швидке виконання програм. Це особливо важливо для наукових обчислень, де час виконання може бути критичним фактором. Julia також має широку підтримку математичних операцій і бібліотек, таких як лінійна алгебра, інтерполяція, оптимізація та машинне навчання. Вона поєднує в собі деякі з найкращих функцій інших мов програмування, таких як Python, R і MATLAB, що робить Julia зручною для чисельного аналізу та статистики. Julia також має потужне співтовариство розробників, яке постійно розширює функціональність та покращує ядро мови. Це означає, що можна швидко знайти рішення на форумах, отримати підтримку від розробників та використовувати підручники та документацію, що допоможе швидко впоратися з розробкою коду. Мова програмування Julia є потужним інструментом для наукових обчислень та чисельного аналізу. Вона поєднує швидкодію, читабельний синтаксис та багатий набір бібліотек.

Мова програмування Julia є потужним і гнучким інструментом для кіберзахисту. Завдяки своєму високому рівню продуктивності та простоті використання, Julia широко використовується у сфері аналізу даних, обробки імейджів, створення алгоритмів машинного навчання та багато іншого. Її широкі функціональні можливості дозволяють розробникам кіберзахисту швидко та ефективно вирішувати складні завдання безпеки мережі, шифрування й аутентифікації, виявлення вразливостей і виконання аудитів безпеки. Використання Julia для кіберзахисту може значно полегшити розробку та розгортання надійних і ефективних захисних рішень. Вона може бути корисною в області кіберзахисту для розроблення алгоритмів аналізу даних та виявлення загроз.

Nim. Розроблення Nim розпочалася в 2008 році, і з тих пір набула значного розповсюдження в сфері кіберзахисту. Вона є сучасною мовою, яка поєднує в собі переваги різних мов, таких як Python, C++, гнучкість якої дозволяє створювати потужні алгоритми для виявлення та запобігання кібератакам.

Мова програмування Nim поєднує в собі різні парадигми програмування, такі як об'єктно-орієнтоване програмування, функціональне програмування та процедурне програмування. Nim має простий та повноцінний синтаксис, що дозволяє легко вивчати та швидко писати як прості, так і складні програми. Одна з головних переваг Nim полягає в його високій продуктивності. Код, написаний на Nim, може бути компільований у виконуваний файл, який працює на власному швидкому виконавчому рівні. Nim також підтримує генерування ефективного машинного коду для різних платформ, таких як Windows, macOS, Linux та інші. Це дозволяє розробникам виконувати свої програми на різних пристроях без значних зусиль. Ще один важливий аспект Nim – це його здатність взаємодіяти з іншими мовами програмування та бібліотеками. Nim має розширення, що дозволяють викликати код, написаний мовами C, C++, Java та іншими. Це дозволяє легко інтегрувати Nim у вже існуючі проєкти та використовувати існуючі бібліотеки. Мова програмування Nim надає багато можливостей для розроблення програмного забезпечення, включаючи веб-розробку, системне програмування, наукові обчислення та багато іншого. Його поєднання простоти, продуктивності та гнучкості робить Nim привабливим вибором для різних проєктів та завдань програмування.

Окрім того, Nim підтримує розроблення криптографічного програмного забезпечення. Завдяки своїм функціям, вона стає незамінним інструментом для реалізації шифрування, цифрових підписів та інших криптографічних алгоритмів, які є важливими складовими в сфері кіберзахисту.

Ruby. Ще одна мова програмування, яка часто використовується для розроблення вебзастосунків. Вона підтримує об'єктно-орієнтоване програмування та функціональні конструкції, що забезпечує багато можливостей для розроблення різноманітних інструментів.

Вона має велику кількість бібліотек, які полегшують розроблення прикладних програм для кіберзахисту. Наприклад, бібліотека OpenSSL надає механізми шифрування та підтримку протоколу SSL / TLS.

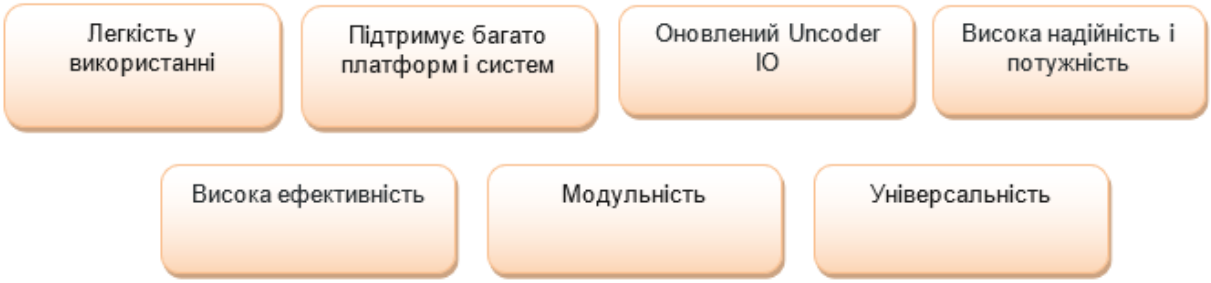

Рисунок 2 – Переваги мови програмування Roota

Мова програмування Ruby є динамічною, об'єктно-орієнтованою мовою з простим синтаксисом, що дозволяє розробникам писати елегантний та зрозумілий код. Вона була створена в Японії в 1995 році і з того часу набула значної популярності у розробників всього світу. Одним з головних принципів Ruby є «зручність для розробників» (principle of least astonishment), що означає, що мова прагне бути інтуїтивною та логічною для користувача. Ruby має ряд вбудованих властивостей, які значно спрощують роботу з даними, текстами, масивами і хеш-структурами. Ruby має багатий набір бібліотек, котрі забезпечують велику кількість функціональності. Багато з цих бібліотек є високоякісними та добре документованими, що дозволяє легко використовувати сторонні рішення для розширення функціональності вашого проєкту. Ruby також має потужне співтовариство розробників, що активно підтримує та вдосконалює ядро мови, розробляє нові бібліотеки та пропонує кращі практики. Це робить Ruby відкритою та живою мовою програмування з багатьма ресурсами для навчання. Ruby також широко використовується для веб-розробки з допомогою фреймворків, таких як Ruby on Rails. Rails, забезпечує структуру та шаблони для швидкого й ефективного розроблення вебдодатків, дозволяючи розробникам сконцентруватися на бізнес-логіці своїх додатків. Загалом, Ruby є потужною мовою програмування, яка поєднує в собі простоту та елегантність з великою кількістю ресурсів та підтримки розробників.

Ruby має потужні засоби для роботи з регулярними виразами, що стають незамінним інструментом у кіберзахисті. Вони дозволяють проводити пошук та оброблення текстів для виявлення шаблонів, таких як електронні адреси, IP-адреси, URL-адреси тощо.

Roota. Roota є однією з ініціатив у сфері кіберзахисту. Вона спеціально розроблена для захисту операційних систем та об'єктів інфраструктури, які відповідають за роботу критичних секторів економіки, таких як електроенергетика, транспорт, виробництво тощо.

Над створенням нової мови працювали топові кіберспеціалісти SOC Prime: Руслан Міхальов, Роман Ранський, Адам Свон, Андрій Безверхий та Олександр Бредіхін. Фідбек збирався командою понад 7 років, концепцію продумували з 2021 року, після залучення інвестицій. Завдяки підготовці та плануванню саме розроблення логіки мови тривало кілька місяців, реліз відбувся у листопаді 2023 року [9].

Мова Roota поєднує ряд принципів та практик, спрямованих на розумне управління загрозами та забезпечення стійкості системи. Вона враховує особливості оперативної технології, що використовується в критичних секторах, і надає рекомендації щодо застосування контролів безпеки, аналізу ризиків та виявлення небезпек.

Мова Roota включає такі аспекти, як моніторинг та виявлення вторгнень, управління доступом, шифрування даних, аналіз поведінки та інші технічні та організаційні заходи безпеки. Це допомагає забезпечити надійність та відновлюваність роботи інфраструктури, навіть в умовах кібератак або випадкових помилок. Можна виділити ряд переваг над іншими мовами програмування (рис. 2) для розробки програмних додатків кібербезпеки.

Щоб не стати жертвою кібератаки, варто бути проактивним та використовувати колективний кіберзахист: як тільки один учасник ринку стикається з новою загрозою, інші кіберфахівці також повинні отримувати інформацію про неї, разом створювати методи захисту та поширювати правила детектування. Щоб ця концепція працювала, індустрії потрібна уніфікована мова для ведення бази знань сценаріїв виявлення загроз. Саме для цього в компанії SOC Prime створили мову Roota – універсальну опенсорсну мову для колективного кіберзахисту, що дозволяє миттєво перекладати алгоритми детектування загроз у будь-яку нативну мову комплексних систем виявлення загроз, наприклад, SIEM, EDR, XDR чи Data Lake [9,12].

ISSN 1993–9965 print
ISSN 2415–3524 online
Науковий вісник ІФНТУНГ
2023. № 2(55)

Таблиця 1 – Переваги та недоліки мов програмування Rust, Julia, Nim та Ruby, з точки зору їх застосування в кіберзахисті

Мова програмування	Переваги	Недоліки
Rust	1) Висока швидкодія та безпека даних. 2) Наявність багатьох бібліотек для криптографії. 3) Вбудована підтримка конкурентності та паралельного програмування.	1) Складність вивчення, особливо для новачків. 2) Недостатня підтримка деяких стандартів криптографії.
Julia	1) Простий синтаксис. 2) Легка інтеграція з іншими мовами програмування. 3) Активний розвиток спільноти та екосистеми.	1) Помірна швидкодія відносно інших мов. 2) Обмежена кількість бібліотек для криптографії.
Nim	1) Висока швидкодія та ефективність. 2) Легкість вивчення. 3) Гнучкість та можливість інтеграції з мовами C/C++.	1) Невелика популярність, порівняно з іншими мовами. 2) Менша кількість бібліотек для криптографії.
Ruby	1) Зрозумілий синтаксис. 2) Велика кількість готових бібліотек та фреймворків. 3) Вбудована підтримка регулярних виразів. 4) Зручна для швидкої розробки прототипів та скриптів.	1) Нижча швидкодія, порівняно з іншими мовами. 2) Потребує інтерпретації, що може призводити до проблем з ефективністю. 3) Обмежена підтримка паралельного програмування. 4) Менша підтримка для криптографічних завдань.

Мова Roota продовжує розвиватися та вдосконалюватися, щоб надавати ефективний захист від кіберзагроз сучасним системам критичного значення. Вона є однією з багатьох ініціатив, спрямованих на підвищення рівня кібербезпеки інфраструктури та захисту від небезпек.

Висновки

Кібербезпека стає все важливішою частиною кіберпростору. Екосистема інформаційної та кібернетичної безпеки містить:

- захист даних;
- захист додатків;
- захист кінцевих точок;
- безпеку мереж;
- ідентифікацію та доступ;
- криптографію;
- аналітику загроз та вразливостей;
- розслідування інцидентів.

Це доволі складна структура зі своєю динамікою, що інтенсифікується в умовах інформаційної війни [10,11]. Для вирішення цих проблем доцільно використовувати комп'ютерні програми, що створюються в середовищах систем програмування. Вибір мови програмування забезпечує ефективність рішень проблем. Ефективність вибору залежить від переваг та недоліків мов програмування, з точки зору їх застосування в кіберзахисті (табл. 1), та інструментів програмування (табл. 2).

Отже, можна зробити висновок:

- Rust забезпечує високу швидкодію та безпеку, а також різноманітність бібліотек, що робить відмінним вибором для кіберзахисту.
- Julia є зручним вибором для наукових обчислень, але її популярність у кіберзахисті ще невелика.
- Nim відзначається високою швидкодією та ефективністю, але вимагає більшого зусилля для вивчення.
- Ruby забезпечує зручний синтаксис та широкий вибір готових бібліотек, але може бути повільною та вимагати більше технічних ресурсів.

Технологія спеціалізованих мов програмування ще не досконала та потребує подальшого розвитку.

З усіх мов програмування ми виділяємо мову Roota, в розробці якої брали участь українські фахівці. Застосування мови програмування Roota для кіберзахисту має кілька важливих переваг:

Таблиця 2 – Порівняльна таблиця бібліотек для кіберзахисту у мовах програмування Rust, Julia, Nim та Ruby

Мова програмування	Бібліотеки, які використовуються для кіберзахисту
Rust	1) RustCrypto (Криптографічні алгоритми) 2) tokio (Асинхронне програмування) 3) serde (Серіалізація та десеріалізація даних) 4) hyper (Високопродуктивний HTTP клієнт та сервер) 5) actix-web (Високопродуктивний веб-фреймворк
Julia	1) Crypto.jl (Криптографічні функції) 2) LibPQ.jl (PostgreSQL клієнт для Julia) 3) HTTP.jl (HTTP клієнт та сервер) 4) WebSockets.jl (WebSocket бібліотека)
Nim	1) NimCrypto (Криптографічні реалізації) 2) NimTLS (TLS/SSL реалізація) 3) asyncnet (Асинхронна мережева бібліотека)
Ruby	1) RbNaCl (Нативний Ruby байндінг) 2) OpenSSL(Ruby байндінг для OpenSSL) 3) net-ssh (SSH клієнт та сервер) 4) rack-protection (Захист від різних атак)

- Ефективність: Roota має малий розмір виконуваних файлів, що робить її ефективним у великих проєктах з кіберзахисту.

- Висока продуктивність: Roota може конкурувати з низькорівневими мовами програмування, що дозволяє швидко виконувати розрахунки та обробку даних.

- Мультипарадигмальність: Roota підтримує як процедурну, так і об'єктно-орієнтовану парадигми програмування, що дозволяє розробляти різноманітні компоненти кіберзахисту.

- Наявність багатофункціональних бібліотек: Спільнота розробників Roota активно створює багатофункціональні бібліотеки для кіберзахисту, що дозволяє розширити можливості мови.

Загалом, використання мови програмування Roota для кіберзахисту може значно полегшити і прискорити процес розробки програм та додатків, забезпечуючи надійний та безпечний кіберзахист.

Література

1. Від початку війни в Україні втричі зросла кількість кібератак – Мінфін США. *Аналітичний портал «Слово і діло»*. URL: https://www.slovoidilo.ua/2023/11/18/novyna/bezpeka/pochatku-vijny-ukrayini-vtrychi-zrosla-kilkist-kiberatak-minfin-ssha

2. Закон України «Про основні засади кібербезпеки України» від 5 жовтня 2017 року. URL: https://zakon.rada.gov.ua/laws/show/2163-19

3. Kravchenko O., Veklych V., Krykhivskyi M., Madryha T. Cybersecurity in the face of information warfare and cyberattacks. *Multidisciplinary Science Journal*. 2024. Vol. 6, 2024ss0219. URL: https://doi.org/10.31893/multiscience.2024ss0219

4. Pavlos Cheimonidis, Konstantinos Rantos. Risk Assessment in Cybersecurity. *A Systematic Literature Review*. September 2023. Future Internet 15(10):324. URL: https://doi.org/10.3390/fi15100324

5. Paolo Santini, Giuseppe Gottardi, Marco Baldi. A Data-Driven Approach to Cyber Risk Assessment. *Security and Communication Networks*. Sept 2019. Vol. 09 URL: https://doi.org/10.1155/2019/6716918

6. Rafał Leszczyna. Review of cybersecurity assessment methods: Applicability perspective. *Computers & Security*, September 2021, Vol. 108, 102376. URL: https://doi.org/10.1016/j.cose.2021.102376

7. Шуліпа Н. С., Мазурик А. В. Дослідження ефективності застосування мови Pyton для створення додатків кібербезпеки та захисту інформації. *Сучасний захист інформації*. 2023. Vol. 3(55), P. 32–37. https://doi.org/10.31673/2409-7292.2023.030004.

8. Мова програмування Rust. URL: https://lemon.school/blog/mova-programuvannya-rust

9. Зроблено в Україні: Roota – мова для колективного кіберзахисту. URL: https://newssky.com.ua/zrobleno-v-ukrayini-roota-mova-dlya-kolektyvnogo-kiberzahystu-z-vidkrytym-kodom/

10. Best Programming Languages for Cybersecurity. URL: https://serokell.io/blog/programming-languages-for-cybersecurity

11. Sakharkar S. Systematic Review: Analysis of Coding Vulnerabilities across Languages. *Journal of Information Security*, 2023, Vol. 14, P. 330-342. doi: 10.4236/jis.2023.144019.

12. Secure development and deployment guidance. URL: https://www.ncsc.gov.uk/collection/developers-collection/principles/produce-clean-maintainable-code

References

1. Vid pochatku viiny v Ukraini vtrychi zrosla kilkist kiberatak – Minfin SShA [Since the beginning of the war in Ukraine, the number of cyber attacks has tripled – the US Treasury Department]. *Analitychnyi portal «Slovo i dilo»*. URL: https://www.slovoidilo.ua/2023/11/18/novyna/bezpeka/pochatku-vijny-ukrayini-vtrychi-zrosla-kilkist-kiberatak-minfin-ssha [in Ukrainian]

2. Zakon Ukrainy «Pro osnovni zasady kiberbezpeky Ukrainy» vid 5 zhovtnia 2017 roku [Law of Ukraine "On the Basic Principles of Cyber Security of Ukraine" dated October 5, 2017]. URL: https://zakon.rada.gov.ua/laws/show/2163-19 [in Ukrainian]

3. Kravchenko O., Veklych V., Krykhivskyi M., Madryha T. Cybersecurity in the face of information warfare and cyberattacks. *Multidisciplinary Science Journal*. 2024. Vol. 6, 2024ss0219. URL: https://doi.org/10.31893/multiscience.2024ss0219

4. Pavlos Cheimonidis, Konstantinos Rantos. Risk Assessment in Cybersecurity. *A Systematic Literature Review*. September 2023. Future Internet 15(10):324. URL: https://doi.org/10.3390/fi15100324

5. Paolo Santini, Giuseppe Gottardi, Marco Baldi. A Data-Driven Approach to Cyber Risk Assessment. *Security and Communication Networks*. Sept 2019. Vol. 09 URL: https://doi.org/10.1155/2019/6716918

6. Rafał Leszczyna. Review of cybersecurity assessment methods: Applicability perspective. *Computers & Security*, September 2021, Vol. 108, 102376. URL: https://doi.org/10.1016/j.cose.2021.102376

7. Shulipa N. S., Mazuryk A. V. A study of the effectiveness of using the Python language to create cyber security and information protection applications. *Suchasnyi zakhyst informatsii.* 2023. No 3, P. 32-37. https://doi.org/10.31673/2409-7292.2023.030004 [in Ukrainian]

8. Rust programming language. URL: https://lemon.school/blog/mova-programuvannya-rust [in Ukrainian]

9. Made in Ukraine: Roota is a language for collective cyber defense. URL: https://lemon.school/blog/mova-programuvannya-rust [in Ukrainian]

10. Best Programming Languages for Cybersecurity. URL: https://serokell.io/blog/programming-languages-for-cybersecurity

11. Sakharkar S. Systematic Review: Analysis of Coding Vulnerabilities across Languages. *Journal of Information Security*, 2023, Vol. 14, P. 330-342. doi: 10.4236/jis.2023.144019.

12. Secure development and deployment guidance. URL: https://www.ncsc.gov.uk/collection/developers-collection/principles/produce-clean-maintainable-code