УДК 681.12

ВАСИЛИНА А. В.

АНАЛІЗ СУЧАСНИХ МОВ ПРОГРАМУВАННЯ ДЛЯ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

У даній роботі проведено аналіз сучасних мов програмування, які можуть бути використані для захисту програмного забезпечення від дій зловмисників. Кожна з розглянутих мов має свої особливості та набори засобів, які кращим чином реалізують той або інший спосіб захисту програмного забезпечення.

Ключові слова: захист програмного забезпечення, мова програмування, Rust, Swift, Java, JavaScript, Python, Kotlin, C++, C#, Ruby, Go.

Abstract

This article analyzes modern programming languages that can be used to protect software from the actions of intruders. Each of the considered languages has its own capabilities and sets of tools that best implement one or another method of software protection.

Keywords: software protection, programming language, Rust, Swift, Java, JavaScript, Python, Kotlin, C++, C#, Ruby, Go.

Вступ

В сучасному світі, де інформаційні технології є необхідністю, захист програмного забезпечення є однією з найважливіших задач. Інформаційна безпека вимагає постійного підвищення рівня захисту, а для цього потрібні відповідні інструменти, зокрема, мови програмування, які дозволять розробляти безпечне програмне забезпечення, а також зробити злом системи безпеки складнішим. Кожен принцип захисту програмного забезпечення передбачає певний набір інструментарію, який найкращим чином справиться з поставленим завданням щодо захисту програмних додатків.

Результати дослідження

Rust - системна мова програмування, яка була розроблена компанією Mozilla з метою підвищення безпеки та швидкості розробки програм. Rust має вбудовані механізми безпеки, такі як контроль використання пам'яті та безпечні типи даних, що дозволяє запобігти багатьом типам програмних помилок. Цю мову програмування можна використовувати для розробки системних програм, браузерних розширень, мультиплатформних додатків, веб-сайтів. Також вона може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані в мову, наприклад, контролювання використання пам'яті, забезпечення безпечного доступу до даних та багатьох інших. Rust дозволяє розробникам створювати безпечні програми, які мають високий рівень захисту від кібератак. Rust має механізми захисту від атак на веб-додатки, такі як SQL Injection та Cross-Site Scripting (XSS), надає можливість використовувати фреймворки, такі як Iron та Rocket.

Swift - це мова програмування, розроблена компанією Apple для створення програм для iOS та macOS. Swift має вбудовані механізми безпеки, такі як безпечні типи даних та обробка помилок, що дозволяє запобігти багатьом типам програмних помилок. Дану мову програмування можна використовувати для розробки мобільних додатків, десктопних програм, серверних додатків, веб-сайтів та багатьох інших. Swift може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані в мову, наприклад, забезпечення безпечного доступу до даних, пам’яті, обробка помилок, механізм автоматичного підрахунку посилань (ARC) та Sandbox, що дозволяє обмежувати доступ до ресурсів комп'ютера. Крім того, Swift має зручний інтерфейс для роботи з криптографічними функціями. Одним з недоліків Swift є те, що вона працює тільки на платформі Apple, що обмежує її застосування в деяких випадках.

Java - це мова програмування, яка використовується для розробки великих проектів. Її можна використовувати для розробки серверних, мобільних, веб-додатків, десктопних програм та багатьох інших. Дана мова пропонує безліч бібліотек для роботи з мережевими протоколами та аутентифікацією. Java може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані в мову, наприклад, контроль доступу до об'єктів, запобігання

вразливостей, що стосуються витоку пам'яті та віддаленого виконання коду та багатьох інших. Одним з головних її переваг є те, що вона працює на віртуальній машині, що дозволяє запобігти багатьом типам атак. Крім того, Java підтримує пакети криптографічних інструментів, такі як пакет JCE (Java Cryptography Extension), що дозволяє шифрувати дані та забезпечувати безпеку мережевих з'єднань.

Також Java має вбудовані механізми захисту, такі як Security Manager та Access Controller, що дозволяють контролювати доступ до ресурсів комп'ютера. Крім того, вона має фреймворки, такі як Spring Security та Apache Shiro, які забезпечують захист від атак на веб-додатки. В ній також є механізми захисту від атак на веб-додатки, такі як SQL Injection та Cross-Site Scripting (XSS).

JavaScript - це мова програмування, яка використовується для створення динамічних веб-сайтів. У контексті захисту програмного забезпечення, JavaScript може бути використана для валідації введених даних та перевірки на вразливості. Крім того, JavaScript може бути використана для розробки додатків, що працюють на клієнтській стороні та використовують API для взаємодії з сервером. Вона має декілька вбудованих механізмів безпеки, таких як Same-Origin Policy та Content Security Policy. Крім того, більшість браузерів мають вбудовані механізми захисту від XSS-атак. JavaScript також має розширену бібліотеку для роботи з криптографією, що дозволяє забезпечити безпеку програмного забезпечення.

Python - це мова програмування, яка використовується для розробки веб-додатків та штучного інтелекту. Python має безліч бібліотек для роботи з мережевими протоколами, шифруванням та розшифруванням даних. Дана мова програмування може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані, наприклад, перевірка типів даних, обробка помилок, забезпечення безпечного доступу до даних, обмеження доступу до файлової системи та мережевих ресурсів.

Крім того, Python має різні фреймворки, такі як Django та Flask, які забезпечують захист від атак на веб-додатки. Також має механізми захисту від атак на веб-додатки, такі як SQL Injection та Cross-Site Scripting (XSS). Однією з переваг цієї мови програмування є її простота та легкість використання, що дозволяє швидко розробляти програмне забезпечення.

Kotlin є мовою програмування, розробленою компанією JetBrains. Вона також має вбудовану підтримку мультиплатформеності, що дозволяє розробляти програмне забезпечення для різних платформ. Kotlin можна використовувати для розробки мобільних додатків, серверних додатків, десктопних програм, веб-додатків та багатьох інших. Kotlin може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані в мову, наприклад, перевірка на null, забезпечення безпечного доступу до даних, захист від небезпечних операцій (safe call operator), можливість визначення змінних, які не можуть бути змінені після їх ініціалізації.

C++ - це мова програмування, яка використовується для розробки системних програм, графічних додатків та ігор. Вона може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, які вбудовані в мову, наприклад, перевірка меж масиву, оброблення виключних ситуацій, RAII (Resource Acquisition Is Initialization – механізм, який допомагає уникнути витоку ресурсів та інших проблем, пов'язаних з управлінням пам'яттю), забезпечення безпечного доступу до даних, контроль типів та багатьох інших.

Для додаткового захисту в C++ існують деякі фреймворки, такі як Microsoft's SDL (Security Development Lifecycle).

C# - це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft для розробки додатків для платформи .NET. Вона може бути застосована для захисту програмного забезпечення, за допомогою механізмів безпеки, наприклад, перевірка типів даних, підтримка безпеки відносно коду, механізм підпису цифрових підписів та криптографічних функцій.

Окрім вбудованих механізмів захисту, таких як Code Access Security, ASLR (Address Space Layout Randomization) і гнучкий механізм ролей Role-Based Security, C# має велику кількість сторонніх бібліотек для захисту програмного забезпечення, такі як бібліотека Security Guard та бібліотека Dotfuscator. Одним з недоліків C# є те, що вона працює тільки на платформі Windows, що обмежує її застосування в деяких випадках.

Ruby - це мова програмування, яка була розроблена у Японії. Вона має вбудовані механізми захисту від SQL-ін'єкцій та Cross-site scripting. Також Ruby має різні фреймворки, такі як Ruby on Rails, які забезпечують захист від атак на веб-додатки. Крім того, вона має розширені бібліотеки для роботи з криптографією (Ruby Encrypt), бібліотеки Ruby Obfuscator та RubyEncoder . Одним з недоліків Ruby є те, що вона не є дуже швидкою мовою програмування.

Go є мовою програмування, яка була розроблена Google для розробки захищених програм для веб-серверів та інших додатків. Вона має вбудовані механізми захисту від помилок з пам'яттю інструменти для роботи з мережевими протоколами та шифруванням даних. Крім того, Go надає можливість використовувати фреймворки, такі як Gin та Revel, які забезпечують додатковий захист від атак на веб-додатки. Go також має механізми захисту від атак на веб-додатки, такі як SQL Injection та Cross-Site Scripting (XSS).

Конкретний вибір мови програмування для захисту програмного забезпечення залежить від того, які засоби (пакети, бібліотеки, розширення тощо) має певна мова у своєму арсеналі. Так, за результатами аналізу інформаційних джерел сформову рекомендований перелік мов програмнування для різних пригнципів захисту(табл. 1).

Таблиця 1 – Застосування мов програмування для певних принципів захисту

Мови програмування	Застосування у захисті програмного забезпечення
C++, Swift, Java, Kotlin. Python, C#,	Захист від несанкціонованого копіювання
C++, Swift, Python, C#, Go	Захист від несанкціонованого дослідження
Rust, C++, Swift, Java, Python, Kotlin, C#, Ruby, Go	Захист від несанкціонованого використання
Rust, C++, Swift, Python, Kotlin, C#,	Захист від дампінгу
Rust, Swift, JavaScript, Python, Kotlin, Ruby, Go	Захист web-ресурсів

Кожна мова була придумана і створена для вирішення певного типу завдань. Велика частина мов перетинається у функціоналі, тому одну і ту саму задачу можна вирішувати різними інструментами. Вибір мови, якою реалізовано те чи інше завдання, багато в чому залежить від сфери її застосування. Питання в тому, чи буде вона працювати ефективно і без збоїв саме для обраного класу задач. Ось чому для вирішення різних завдань слід вибирати найбільш відповідні мови програмування.

Висновки

В кожній мові програмування є свої переваги і недоліки в плані безпеки програмного забезпечення. Вибір мови залежить від конкретних потреб проекту. Будь-яка мова може бути використана для розробки безпечного програмного забезпечення, якщо її використовувати правильно. Для того, щоб забезпечити безпеку програмного забезпечення, необхідно використовувати кращі практики програмування, використовувати безпечні функції та бібліотеки, оновлювати програмне забезпечення та використовувати найкращі способи захисту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Robert C. Seacord. Secure Coding in C and C++, 2nd Edition. СПб.: Williams, 2019. 496 с.
2. Oracle. Security in Java SE. Режим доступу: https://docs.oracle.com/en/java/javase/16/security/index.html. (дата звернення: 31.03.2023).
3. Microsoft. C# programming guide. Режим доступу: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/ (дата звернення: 30.03.2023).
4. Python Software Foundation. (2021). Cryptographic Services. Режим доступу: https://docs.python.org/3/library/crypto.html.
5. 7 Steps to Secure JavaScript. Режим доступу: https://blog.bitsrc.io/8-steps-to-secure-javascript-in-2021-6d54d5415264. (дата звернення: 21.03.2023).
6. Prateek Joshi. Artificial Intelligence with Python: Your Complete Guide to Building Intelligent Apps Using Python 3.x, 2nd Edition. Packt Publishing , 2020. 618 с.
7. Programming Rust: Fast, Safe Systems Development 2nd Edition, Автор: Jim Blandy, Jason Orendorff, Leonora Tindall. O'Reilly Media, 2021. 453 с.

Василина Анастасія Василівна – студентка групи 2БС-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: nstvsln@gmail.com.

Науковий керівник: ***Каплун Валентина Аполінаріївна*** - ст. викл. кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, e-mail: valuka8379@gmail.com.

Vasylyna Anastasia Vasylivna- is a student of group 2BS-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.

Supervisor: ***Kaplun Valentyna Apolinariivna*** - Lecturer of the Chair of Safety of Information and Communication Systems, NTU, Vinnytsia.