УДК 004.45(043.2)

МОВА ПРОГРАМУВАННЯ RUST ЯК ІНСТРУМЕНТ ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА НАДІЙНОСТІ ВБУДОВАНИХ СИСТЕМ

Національний авіаційний університет, Київ

Ключові слова: Мова Rust, безпека, вбудовані системи, мова програмування

Вступ

У сучасному авіаційному секторі, де безпека та надійність вбудованих систем є критичною, використання відповідних інструментів для розробки програмного забезпечення набуває все більшого значення. У цьому контексті мова програмування Rust може виступати як ключовий інструмент, оскільки вона пропонує безпеку пам'яті та надійність на рівні системи, що є критичною характеристикою для сучасних інформаційних та комунікаційних технологій в авіації.

Матеріали і методи

У даному дослідженні об'єктом є мова програмування Rust та її можливості щодо забезпечення безпеки та надійності вбудованих систем. Для досягнення поставленої мети було використано аналітичний підхід, що передбачав огляд наукової літератури, а також аналіз і порівняння практичних застосувань мови Rust у вбудованих системах авіації. Були застосовані методи аналізу властивостей мови програмування, порівняльний аналіз з іншими мовами програмування для вбудованих систем, а також вивчення та аналіз практичних прикладів використання Rust.

Результати

Реактивний підхід до кібербезпеки, характерний для багатьох компаній, неефективний у прогнозуванні та захисті від інцидентів, що може призвести до недостатнього захисту вбудованих систем. Вразливості безпеки пам'яті, зокрема ті, що впливають на доступ, запис та розподіл пам'яті у ненавмисний спосіб, є критичними проблемами кібербезпеки. За загальною статистикою встановлено, що не менше 70% всіх вразливостей програм припадають на проблеми керування пам’яттю, що в традиційних мовах програмування C та C++ керуються вручну програмістом, і при компіляції програми не виконується відстеження того, чи всі частини програми правильно керують пам’яттю.

Rust має вбудовану систему власностей та позик, що дозволяє уникнути багатьох типових помилок, пов'язаних з управлінням пам'яттю, таких як переповнення буфера або дереференсіювання нульових вказівників. Для критичних програм, які можуть

використовуватись в комп’ютерах медичних, автомобільних та авіаційних систем, гарним вибором мови написання може бути використання саме мови Rust. Її спільнота хоч і нова, але активно розвивається і поширює по мережі Інтернет відкритий та безпечний код. Rust має вбудовану систему керування паралельністю, яка дозволяє розробникам створювати безпечні паралельні програми, уникнувши багатьох проблем, пов'язаних з багатопотоковими програмами, таких як гонки за даними.

Ці можливості доводять, що Rust може допомогти у вирішенні поширених проблем ПЗ, які є актуальними вже понад 35 років. У лютому місяці 2024 року президент США підписує документ, що визначає стратегію розвитку кібербезпеки країни, розроблений SISA, NSA, FBI та міжнародними агентствами з кібербезпеки. Цей звіт містить рекомендації для виробників технологій щодо усунення вразливостей кібербезпеки в їхніх продуктах, зокрема вказівки на важливість безпеки пам'яті для усунення різних уразливостей програмного забезпечення. Аналізовано використання мов програмування C і C++ у космічних системах, а також привернута увага до Rust як безпечної для пам'яті мови, яка має потенціал, але ще не була достатньо випробувана в космічних системах.

Висновки

Розробники критичного програмного забезпечення використовують інструмент, який було винайдено вже близько 54 роки тому. Це дійсно гарний показник - мова С пройшла довгий шлях, і стала виходом бізнесу для написання програм, що тісно працюють з ядром. Але головна проблема цієї мови - керування пам’яттю, яка вирішувалась лише досвідченістю програміста, залишаючи можливість утворення вразливого коду. США помітили це і намагаються приділити увагу вирішенню цієї проблеми. Українським компаніям розробки ПЗ, в особливості ПЗ для військових цілей, варто звернути увагу на покращення безпеки власних продуктів, змінивши робочий потік на безпечніші інструменти.

Список використаних джерел

1. BACK TO THE BUILDING BLOCKS: A PATH TOWARD SECURE AND MEASURABLE SOFTWARE [Електронний ресурс] // The White House. – 2024. – Режим доступу до ресурсу: https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf

2. Troutwine B. Hands-On Concurrency with Rust / Brian Troutwine., 2018. – 462 с.