Смірнов Д.С., аспірант
Яцків І.В., здобувач
Західноукраїнський національний університет

ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВБУДОВАНИХ СИСТЕМ

Захист програмного забезпечення вбудованих систем, таких як Raspberry Pi, набуває актуальності через стрімке поширення таких пристроїв у сферах IoT, автоматизації, медицини та промисловості, що підвищує потребу в запобіганні копіюванню та несанкціонованому запуску, а також через їх вразливість до атак завдяки обмеженим ресурсам і фізичній доступності; це критично важливо для захисту інтелектуальної власності розробників від піратства, забезпечення безпеки даних у критичних системах (наприклад, розумних будинках чи транспорті), де порушення може загрожувати витоком інформації або навіть життю, а також для відповідності посиленим регуляторним стандартам безпеки, таким як GDPR чи NIST, що робить захист ПЗ ключовим елементом сучасної кібербезпеки та конкурентоспроможності [1, 2].

Для захисту ПЗ вбудованих систем використовують ряд підходів, зокрема:

1) обфускація коду. Обфускація коду змінює вихідний код програми, щоб зробити його складнішим для розуміння та зворотньої інженерії, при цьому зберігаючи його функціональність. Це може включати перейменування змінних та функцій на беззмістовні імена, додавання непотрібного коду, зміну структури коду тощо. Відносно простий у застосуванні, особливо для інтерпретованих мов, таких як Python, які часто використовуються на Raspberry Pi. Однак, обфускований код впливає на продуктивність програми, що часто є критично для вбудованих систем , а також ускладнює налагодження та підтримку коду;

2) апаратне блокування (Hardware Locking). Прив'язка програми до унікальних характеристик апаратного забезпечення, наприклад Raspberry Pi, щоб програма працювала лише на конкретному пристрої або набору пристроїв. Програма збирає інформацію про апаратне забезпечення пристрою (наприклад, серійний номер процесора, MAC-адресу мережевої карти) та використовує її для перевірки валідності запуску. Raspberry Pi має унікальний серійний номер, який можна отримати програмно (наприклад, через /proc/cpuinfo). MAC-адресу мережевої карти також можна використовувати, але вона менш унікальна, ніж серійний номер. Може бути ефективним, якщо програмне забезпечення призначене для використання на конкретному обладнанні. Проте можливий обхід через емуляцію апаратного забезпечення або модифікацію коду програми, щоб ігнорувати апаратну перевірку. Для підвищення надійності можна використовувати комбінацію декількох апаратних характеристик;

3) онлайн активація. Програма при запуску або періодично звертається до сервера для перевірки ліцензійного статусу. Якщо ліцензія валідна, програма продовжує роботу; інакше, вона може припинити роботу або обмежити функціональність. Ускладнює використання неліцензійних копій, оскільки для роботи програми потрібне постійне або періодичне підключення до інтернету. Серед недоліків, це залежність від інтернет-з'єднання, програма не працюватиме без доступу до інтернету. Сервери активації можуть стати ціллю атак, що вплине на працездатність програмного забезпечення;

4) використання апаратного модуля безпеки (TPM або HSM). Апаратний модуль безпеки (наприклад, Trusted Platform Module або спеціальний чип), який вбудована система може використовувати для аутентифікації. Програма запускається лише за наявності цього модуля. Серед переваг, це високий рівень захисту, важко обійти без фізичного доступу до модуля. Однак, необхідно враховувати додаткові витрати на обладнання та складність інтеграції з системами, які за замовчуванням не мають вбудованого TPM;

5) шифрування коду. Шифруванню підлягають основні частини програми і для їх запуску необхідно введення ключа, який може бути згенерований на основі унікальних даних пристрою (наприклад, серійного номера). Ключ можна перевіряти локально або через сервер. Такий підхід ускладнює копіювання, особливо якщо перевірка відбувається онлайн, однак, потребує надійного механізму генерації ключів і захисту від злому шифрування.

Захист програмного забезпечення вбудованих систем вимагає комплексного підходу, який включає як апаратні, так і програмні рішення. Використовуючи комбінацію цих методів, можна забезпечити високий рівень захисту комерційного програмного забезпечення від несанкціонованого копіювання та розповсюдження.

Список використаних джерел:

1. Rachit, Bhatt, S., Ragiri, P. R. Security trends in Internet of Things: A survey. SN Applied Sciences. 2021. Vol.3. P.1-14.

2. Mousavi, S. K., Ghaffari, A., Besharat, S., Afshari, H.. Security of internet of things based on cryptographic algorithms: a survey. Wireless Networks. 2021.Vol. *27*(2). P.1515-1555.