МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УДК 004.4

Тези доповідей

V Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

19–20 травня 2022 р.

Кропивницький 2022

УДК 004.4

Матеріали V Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 19–20 травня 2022 р. – Кропивницький: ЦНТУ, 2022. – 72 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

УДК 004.453, 004.33

В.С. Катаєв, І.С. Каплун, І.О. Бондаренко
kataev@vntu.net, kaplun.irka@gmail.com, fm.ub15b.bondarenko@gmail.com
Вінницький національний технічний університет, м. Вінниця

ПІДВИЩЕННЯ ЗАХИЩЕНОСТІ ПРОГРАМ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ З ВИКОРИСТАННЯМ АПАРАТНОГО ЗАСОБУ

Вступ. Будь – який продукт, котрий використовує кожна людина, є товаром, який має автора, ціну, гарантії якості, а отже може бути об'єктом крадіжки та незаконного використання. Необхідність внесення до програмного забезпечення захисних функцій протягом його життєвого циклу від етапу з'ясування задуму на створення програм і їх розробки до етапів випробувань, експлуатації, модернізації і супроводу програм є досить поширеною [1].На сьогодні актуальним завданням є підвищення захищеності програм від НСД, зокрема шляхом використання протоколу активного захисту.

Розроблення алгоритмів роботи. Пропонується застосовувати для електронного ключа, що дає доступ до програмного забезпечення, покращений протокол активного захисту (при кожному запуску програми перевіряється наявність ключа у портах ПК). Специфіка апаратної розробки полягає у самознищенні ключа у разі його відсутності, або невірному використанні. Така функція є новою серед його аналогів.

Переваги запропонованої модифікації електронного ключа: наявність алгоритму активного захисту (якщо ключ отримує невірний запит, відбувається очищення пам'яті мікроконтролера)

Це може бути реалізовано з допомогою власного бутлоадера і зроблено для унеможливлення використання ключа після спроби використання його на іншому комп'ютері, або спроби його вивчення методом чорної скриньки (коли зловмисники вивчають реакцію приладу на різні запити).

Для перевірки валідності запиту, ключ перевіряє:

- базову швидкість порту, з яким він працює;
- коректність протоколу роботи з комп'ютером та коректність запиту, що надходить до пристрою.

Також система захищена від підключення ключа в інші ПК, відправки йому запитів і дослідження, що ключ відповість на той чи інший запит. У разі некоректного запиту, flash - пам'ять очищується.

Принцип роботи розробки полягає у тому, що після первинної активації (яка проводиться довіреною особою), ключ виявляється в ПК і програма його розпізнає, відбувається прив'язка. Програма при запуску перевіряє наявність ключа в портах. Ключ, в свою чергу, під час надходження до нього запиту, перевіряє: швидкість передачі даних, на якій йому прийшло повідомлення; коректність протоколу (реалізовано т.зв. "рукостискання", воно ж handshake); коректність запиту, який він отримав від комп'ютера. Запит є динамічним, тобто кожен раз відрізняється.

Структура запиту може бути описана наступним чином:

- комп'ютер послідовно відкриває всі доступні послідовні порти;
- при відкритті порту, він перевіряє, чи приходять в порт певні дані (ключ при цьому в свій порт відправляє хеш - код свого серійного номера, щоб його могли впізнати);
- якщо виявлено певний (не обов'язково правильний) ключ, відбувається його верифікація;
- ПК здійснює відправку ключеві дати активації в хешованому вигляді;
- ключ здійснює хешування своєї дати активації та зіставляє хеш-коди;
- у випадку, якщо хеш-коди збігаються, то відбувається робота з програмою далі;
- у випадку, якщо хеш – коди не збігаються, відбувається очищення flash - пам'яті.

Після запуску програми, активації, перевірки ключа, і запуску захищуваного програмного забезпечення, відбувається періодична перевірка присутності ключа.

Для виконання вказаної функції відбуваються такі дії:

- послідовний порт залишається відкритим при активації;
- у порт, в якому вставлений ключ, відправляється хешована кількість секунд з моменту активації;
- ключ здійснює перевірку правильності отриманої інформації;
- у випадку, якщо фіксуються некоректні дії – відбувається очищення флеш-пам'яті ключа;
- у випадку, якщо вся робота коректна, то відбувається трансформація рядка, отриманого від ПК та його відправка (таким чином, якщо витягти ключ, захищуваний модуль перестане працювати).

Отже, запропонована модифікація є новою серед аналогів електронних ключів для захисту програмного забезпечення, проте може бути достатньо дієвою, що є підставою для проведення подальшої розробки. Алгоритм роботи системи контролю доступу до програмного забезпечення базується на двоетапній авторизації будь – якого користувача, що дає можливість забезпечити її коректну роботу та надійний захист.

Засоби для реалізації розробки. Система контролю доступу до програмного забезпечення складається з програмної частини (що знаходиться на ПК разом з захищуваним ПЗ та прив'язується до апаратної частини ПК) та електронного ключа, виконаного на базі мікроконтролера Atmega 328.

При розробці апаратної частини пристрою, увага приділяється практичній реалізації електронного ключа, що в результаті надає користувачу можливість отримати доступ до захищуваного програмного

забезпечення. Апаратна реалізація ключа виконана на мікроконтролері AtMega328 сімейства AVR виробництва компанії Atmel. Плата розробки, на базі якого зроблений ключ – Arduino Nano [2].

Користувацька частина системи контролю доступу до програмного забезпечення з використанням USB-ключів розроблена в середовищі програмування Visual Studio на мові об'єктно - орієнтованого програмування C# [3].

Для виконання поставленої задачі, розроблені модулі додатку: модуль користувацької частини; модуль апаратної частини реалізації. Для програмної реалізації модуля користувацької частини проекту, слід віднести такі блоки: блок 1 - робота з послідовними портами, блок 2 - індикація даних, блок 3 - перевірка повідомлень, блок 4 - авторизація, блок 5 - контроль портів, блок 6 - розробка захищуваного додатку та його інтерфейсу. Модульне проектування апаратної частини практичної реалізації містить такі блоки: блок 1 - передача даних, блок 2 – завантажувач, блок 3 - процес здійснення перевірки коректності здійснюваних запитів.

Тестування розробки. Наступним етапом роботи є практична перевірка надійності електронного ключа для захисту програмного забезпечення.

Перша перевірка буде здійснюватися за допомогою сніфера. Отже, було встановлено команду, якою обмінюються ключ і комп'ютер при спільний автентифікації, дана команда записується. Злам ключа реалізовано за допомогою термінальної програми, адже розроблюваний ключ працює з перетворювачем usb – com. Якщо електронний ключ від'єднати від ПК, він відповідно зникне з вікна диспетчера пристроїв. Такі дані свідчать про те, що ключ «спілкується» з ПК за допомогою USB – UART перетворювача.

Наступний метод – спробувати проаналізувати ключ та дізнатись, що він складається з Atmega328, який являється мікроконтролером сімейства AVR, а отже в більшості випадків прошивається через ISP. Виходи SCK, MISO, MOSI, Reset використовуються для підключення програматора. Для того, щоб здійснити подальше підключення необхідно спеціальне програмне забезпечення для роботи з програматором. Спробуємо зчитати прошивку ключа з флеш – пам'яті мікроконтролера. Для цього скористаємось додатком extreme burner. Спробуємо здійснити підключення. З отриманих даних можна побачити, що підключення не відбулось. Такий результат пояснюється тим, що підключення не відбулось з тієї причини, що в мікроконтролері відключений ISP для неможливості виведення, а також перезавантаження як логічного виходу. Такі налаштування суттєво ускладнюють процес зчитування програматора для зловмисника.

Отже, можна припустити, що теоретично ключ може піддатись зламу у випадку майже неможливої випадковості, методом підбору і т.д. Проте слід зауважити, що часу на здійснення спроб для несанкціонованого доступу до ключа, потрібно чи мало, проте, ключ – лише один. І запрограмований він таким чином, що в жодному разі не допускає невірної активації чи найменшої спроби зламу. Такі ключі орієнтовно можна використовувати для особливо важливого програмного забезпечення, використання якого несанкціонованими користувачами взагалі не допускається.

Висновки. В даній роботі було описано основні етапи розробки системи контролю доступу з використанням форми авторизації та електронного USB – ключа. В ході розробки, головним структурним елементом підсистеми було обрано пристрій Arduino Nano на основі мікроконтролера ATmega328, оскільки в ньому міститься усе необхідне для зручної роботи з контролером.

Тестування програми проводилось на ПК з операційною системою Windows XP, 7, 8, 8.1 та 10. На вказаних вище операційних системах розроблювальна система захисту ПЗ працює коректно за умови встановленого драйверу для мікросхеми CH-340g (USB – UART конвертер).

На практиці протестовано результативність та стійкість розроблюваного додатку. Тестування показало, що розроблювальна система захисту ПЗ працює коректно, а дана розробка потребує незначну кількість ресурсів для своєї коректної роботи.

Список літератури

1. Огляд сучасних методів захисту програмного забезпечення. StudFiles. URL: https://studfile.net/preview/3905114/ (дата звернення: 01.05.2022).

2. Arduino Nano: все, що потрібно знати про плату розробки. Hardware libre. URL: https://www.hwlibre.com/uk/arduino-nano/ (дата звернення: 01.05.2022).

3. Вступ в C# URL: https://programm.top/uk/c-sharp/tutorial/introduction/ (дата звернення: