*УДК 004.056.53*

*https://doi.org/10.52058/2786-6025-2023-2(16)-448-459*

**Шаров Сергій Володимирович** кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук, Таврійський державний агротехнологічний університет імені Дмитра Моторного, вул. Жуковського, 66, м. Запоріжжя, 69600, https://orcid.org/0000-0001-5732-9980

**Лубко Дмитро Вікторович** кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Таврійський державний агротехнологічний університет імені Дмитра Моторного, вул. Жуковського, 66, м. Запоріжжя, 69600, https://orcid.org/0000-0002-2506-4145

# РОЗРОБКА КОМП'ЮТЕРНОЇ ПРОГРАМИ ДЛЯ ЗАХИСТУ ВИКОНУВАНИХ ФАЙЛІВ WINDOWS

**Анотація.** У статті повідомляється про розробку програмного засобу для захисту виконуваних файлів ОС Windows. Зазначається, що стрімкий розвиток інформаційного суспільства вплинув на збільшення кількості програмного забезпечення, що виконує системні, інструментальні, прикладні функції тощо. З'ясовано, що поширеною проблемою в ІТ-індустрії є комп'ютерне піратство. Несанкціонована модифікація та розповсюдження ліцензійного програмного забезпечення завдає шкоди розробникам, підприємствам, звичайним користувачам. Часто зловмисні дії спрямовані на виконувані файли операційної системи Windows, які структурно складаються з декількох заголовків та секцій. З'ясовано, що зловмисники можуть здійснити декомпіляцію або дизасемблювання виконуваного файлу, отримати доступ до вихідного програмного коду, проаналізувати логіку роботи програми та зробити в ній певні модифікації. Виявлено, що захист від несанкціонованої модифікації комп'ютерних програм та мобільних додатків забезпечується розробкою та впровадженням законодавчих документів, використанням апаратних та програмних методів захисту. Для захисту часто використовуються обфускаційні методи, які дозволяють змінювати вихідний програмний код при збереженні функціональності програми. Висвітлено приклади використання методу заплутуючих перетворень, інтеграції цифрового водяного знаку в структуру виконуваного файлу тощо. Повідомляється про розробку програмного засобу, який дозволяє захистити виконуваний файл Windows без наявності вихідного коду. Описано основні функціональні можливості утиліти: анти налагодження, захист таблиці

імпорту exe-файлу, захист паролем, випробувальний режим та ін. Програмний засіб був розроблений за допомогою мови програмування C++, середовища Visual C++, бібліотеки Qt, JsonCpp та інших інструментальних засобів. У подальшій роботі передбачається вдосконалити можливості програмного засобу та перевірити його працездатність.

**Ключові слова:** програмне забезпечення, комп'ютерне піратство, захист, операційна система, Windows, обфускаційні методи.

**Sharov Sergii Volodymyrovych** PhD of Pedagogical Sciences, Associate professor, Associate professor of the Department of Computer Science, Dmytro Motornyi Tavria State Agrotechnological University, Zhukovsky St., 66, Zaporizhzhia, 69600, https://orcid.org/0000-0001-5732-9980

**Lubko Dmytro Viktorovych** PhD of Technical Sciences, Associate professor, Associate professor of the Department of Computer Science, Dmytro Motornyi Tavria State Agrotechnological University, Zhukovsky St., 66, Zaporizhzhia, 69600, https://orcid.org/0000-0002-2506-4145

## THE DEVELOPMENT OF A COMPUTER PROGRAM FOR THE PROTECTION OF WINDOWS EXECUTABLE FILES

**Abstract.** The article reports on the development of a software tool to protect Windows OS executable files. It is noted that the rapid development of the information society has influenced the increase in the number of software that performs system, tool, application functions, etc. It has been found that computer piracy is a common problem in the IT industry. Unauthorized modification and distribution of licensed software harms developers, enterprises, and ordinary users. Malicious actions often target executable files of the Windows operating system, which are structurally composed of several headers and sections. It has been defined that attackers can decompile or disassemble the executable file, gain access to the source program code, analyze the logic of the program and make certain modifications to it. It has been found that protection against unauthorized modification of computer programs and mobile applications is provided by the development and implementation of legislative documents, the use of hardware and software protection methods. For protection, obfuscation methods are often used, which allow changing the source program code while preserving the functionality of the program. Examples of using the method of confusing transformations, integration of a digital watermark into the structure of an executable file and others have been highlighted. The development of a software tool that allows you to protect a Windows executable file without the presence of the source code has been

reported. The main functionality of the utility has been described: anti-debugging, protection of the exe-file import table, password protection, trial mode, etc. The software was developed using the C++ programming language, the Visual C++ environment, the Qt library, JsonCpp and other tools. In the future work, it is planned to improve the capabilities of the software tool and check its performance.

**Keywords:** software, computer piracy, protection, operating system, Windows, obfuscation methods.

**Постановка проблеми**. Сьогодні ми спостерігаємо знаний прорив у технологічному розвитку інформаційного суспільства, повсюдне застосування мережі Internet, мобільних технологій, хмарних обчислень та систем штучного інтелекту у побуті, на виробництві, в бізнесі та інших сферах. В свою чергу, це впливає на розвиток різноманітного програмного забезпечення, що виконує системні, інструментальні, прикладні функції тощо. Це дозволяє комфортно відчувати себе у цифровому просторі та повною мірою користуватися перевагами інформаційного суспільства.

На жаль, широке використання IT-технологій поряд з численними перевагами супроводжується низкою проблем. У першу чергу це стосується нелегальної модифікації, копіювання та розповсюдження ліцензійного програмного забезпечення. Комп'ютерне піратство і незаконне використання програмного забезпечення завдає величезної шкоди не тільки розробниками, а й підприємствам та звичайним громадянам в залежності від виду модифікації.

Внаслідок технологічних особливостей виконувані файли операційної системи Windows представлена у вигляді машинного коду, або у викладі проміжного байт-коду, що виконується віртуальною машиною. Такі особливості відкривають широкі можливості для зловмисників. За допомогою спеціального програмного забезпечення вони можуть відновити вихідний код комп'ютерної програми, знайти слабкі місця та здійснити незаконну модифікацію програмного забезпечення.

Оскільки кількість програмного забезпечення постійно зростає, збільшується і кількість потенційних можливостей для зловмисників та комп'ютерного піратства. Крім вдосконалення правового захисту ліцензійного програмного забезпечення на рівні держави, існує потреба у розробці нових методів та способів захисту виконуваних файлів. Завдяки цьому програмний код стане більш незрозумілим для зворотнього інжинірингу при повному збереженні функціональності програмного забезпечення.

**Аналіз останніх досліджень і публікацій**. Внаслідок актуальності проблеми комп'ютерного піратства та незаконного використання ліцензійного програмного забезпечення існує значна кількість наукових праць, що висвітлюють різні напрямки захисту. Аналіз досвіду щодо боротьби з

комп'ютерним піратством здійснено у роботі А. Бакали. Дослідження С. Вітер та І. Світлишина присвячені напрямкам та засобам захисту облікової інформації на підприємствах.

Стосовно ОС Windows А. Ільєнко, С. Ільєнко, Т. Куліш розглянули деякі вразливості операційної системи та висвітлили основні підходи та методи до її захисту. У дослідженні J. E. Tevis, Jr J. Hamilton аналізується структура exe-файлу з точки зору вразливості, описано методологію аналізу виконуваного файлу з метою виявлення аномалій та вразливостей.

Для захисту програмного коду від несанкціонованого аналізу та використання використовуються різноманітні методи, зокрема метод заплуюучих перетворень (Я. Корнага, Ю. Базака, М. Базалій), впровадження цифрового водяного знаку в асемблерний код програми з подальшим компілюванням виконуваного файлу (А. Стороженко, А. Горпенюк, Н. Лужецька). Дослідники І. Степаненко, В. Кінзерявий, А. Наджі А. та І. Лозінський запропонували власну класифікацію обфускаційних методів та розглянули сутність окремих методів захисту програмного коду. У роботі А. Левчука також висвітлено способи обфускації вихідного коду.

Незважаючи на значну кількість існуючих програмних засобів, що використовуються для захисту виконуваних файлів, зловмисники розробляють нові способи зламу програмного забезпечення. Тому виникає потреба у створенні нових способів захисту програмного забезпечення, у тому числі виконуваних файлів Windows.

**Мета статті** полягає у повідомлені про розробку комп'ютерної програми для захисту виконуваних файлів ОС Windows, описі його функціональних можливостей.

**Виклад основного матеріалу.** В сучасній ІТ-індустрії комп'ютерне піратство та захист цифрових даних вважається однією з найпоширених проблем, яка потребує постійної уваги [1]. Розповсюдження піратського програмного забезпечення знижує рентабельність галузі, завдає фінансової шкоди розробникам [2]. Крім того, оскільки більшість підприємств використовує спеціалізоване програмне забезпечення, крадіжка економічної інформації та персональних даних негативно вплине на їх комерційну діяльність [3].

Боротьба проти несанкціонованого використання програмного забезпечення та його модифікації здійснюється у різних напрямках. По-перше, розробляються механізми легалізації програмного забезпечення та його офіційного використання на законодавчому рівні за рахунок узгодження окремих нормативно-правових актів з питань інтелектуальної власності, дотримання авторського права на програмні засоби, спеціальних академічних програм ліцензування [2] тощо. З іншого боку, для захисту програмного

забезпечення та інформації (персональної, економічної тощо) використовуються різноманітні апаратні, програмні засоби або їх поєднання. До апаратних методів можна віднести використання електронних ключів, оптичних дисків, інші фізичні пристрої або спеціальне обладнання [1], що підключається до комп'ютера та забезпечує ідентифікацію особи або надає доступ. Звісно, не слід забувати про фізичне обмеження доступу до інформації, яка зберігається на носіях або пристроях.

Досить різноманітною є лінійка програмних засобів для забезпечення інформаційної безпеки та захисту програмного забезпечення. Під ними розуміється сукупність алгоритмів та компонентів, що реалізуються та інтегруються у програмне забезпечення спеціального призначення, забезпечують функції контролю, обмеження доступу тощо [4]. Наприклад, у випадку контролю за Internet-трафіком добре себе зарекомендували сніфери (sniffers), які перехоплюють пакети та надають можливість системному адміністратору їх проаналізувати, виявити підозрілу активність користувачів або програмного забезпечення та здійснити відповідні корективи [5]. Якщо мати на увазі операційну систему Windows, то вона має вбудовані засоби захисту системних файлів та програмного забезпечення. Виявлені недоліки усуваються за допомогою оновлення та патчів Microsoft, які оперативно з'являються у відповідь на нові загрози. Також існують додаткові програмні модулі для захисту ОС Windows. Наприклад, у роботі [6] повідомляється про розробку програмного модулю, який за допомогою технології Blockchain перевіряє цифрові сертифікати. На думку Я. Корнаги, витрати, що пов'язані зі створенням захисного програмного забезпечення, є економічно вигідними для розробників, які постійно втрачають великі прибутки внаслідок нелегального використання та копіювання ліцензійного програмного забезпечення [4].

Функціональність ОС Windows забезпечується встановленим програмним забезпеченням, яке зазвичай представлено у вигляді скомпільованих exe-файлів, в яких відсутній вихідний код програми. Вони виконуються безпосередньо операційною системою, або віртуальною машиною. Назва Portable Executable (PE) пов'язана з тим, що даний формат не залежить від архітектури процесора, для якого побудований виконуваний файл. Кожний виконуваний файл складається з декількох заголовків і декількох секцій. Заголовки містять службову інформацію, що описує різні властивості файлу і його структуру. Секції містять дані, які розміщуються в адресному просторі під час завантаження файлу в пам'ять. Прикладом можуть слугувати секції коду, ресурсів, статичних та динамічних даних. Кожна секція PE-файлу має ім'я та набір атрибутів. Атрибути секції визначають, чи містить вона виконуваний код, чи доступна для читання та запису, чи повинна вона залишатися в пам'яті після завантаження файлу тощо. У дослідженні J. Tevis

and Jr J. Hamilton зазначається, що не всі структурні блоки виконуваного файлу Windows можуть містити інформації, яка дозволить виявити шкідливий код. Так, вразливість системи безпеки може бути виявлена через аналіз вмісту section table, import table, symbol table [7].

Іноді до виконуваних файлів застосовуються методи зворотнього інжинірингу, тобто перетворення машинного коду в програму на мові Assembler [1]. Такий підхід є досить поширеним при статичному аналізі виконуваного файлу на наявність шкідливого програмного коду. Над виконуваними файлами, які виконуються віртуальною машиною, можна здійснити декомпіляцію у програмний код на мові високого рівня. Для цього існують багато безкоштовних декомпіляторів: Boomerang, RecStudio, Hex rays [8]. Слід зазначити, що декомпіляція та дизассемблювання застосовуються цілком законно у випадку зворотнього проєктування програмного засобу, розуміння структури програми та алгоритму її роботи. В той же час, ті ж самі методи можна використовувати для зламу захисних функцій програмного забезпечення, модифікації виконуваного файлу з метою усунення перевірки ліцензії тощо. Все залежить від мети, яка була покладена в основу динамічного або статичного аналізу комп'ютерної програми.

Ефективними методами захисту програмного забезпечення є обфускаційні методи, які дозволяють змінювати вихідний програмний код при збереженні функціональності програми. Модифікований таким чином виконуваний файл ускладнить аналіз вихідного коду програми та алгоритму її роботи для подальшої модифікації. У роботі [9] зазначається, що обфускаційні методи можна класифікувати в залежності від вижду захисту: зміна пунктуації програмного коду; перетворення змінних; трансформація структури програмного коду. Я. Корнага звертає увагу на використання заплутуючих перетворень для захисту виконуваних файлів, після чого процес реверсивної інженерії буде ускладнений, а отриманий програмний код буде неефективним [4]. Інші вчені інтегрують у вихідний програмний код додаткові секції. Наприклад, у роботі [1] впровадження водяного знаку у вихідний код здійснюється за декілька кроків: дизасемюлювання виконуваного файлу; вставки водяного знаку та додаткових перетворень в асемблерний програмний код; повторна компіляція отриманого файлу.

Розроблена нами комп'ютерна програма використовує обфускаційні методи та забезпечує захист виконуваного файлу при його завантаженні в пам'ять комп'ютера. Ми використали інтеграцію додаткової секції з фрагментами захисного програмного коду в кінець виконуваного файлу. Точка входу коригується таким чином: при завантаженні exe-файлу управління передається додатковому захисному фрагменту, а в складі захисного фрагмента передбачена процедура повернення до оригінальної точці входу.

Перевагою такого підходу є відсутність будь-яких обмежень на структуру та розмір виконуваного коду і даних, вільне оперування атрибутами секції. Це дозволяє спростити реалізацію інтегрованого модуля та покращити стабільність його роботи. Процес додавання нової секції складається з декількох етапів:

1. Пошук вільної адреси у віртуальному адресному просторі.
2. Переміщення коду модуля за даною адресою за допомогою директорії елементів, що переміщуються.
3. Внесення змін в директорію імпорту. Так як всі зміщення в таблиці імпорту вказані відносно точки проектування файлу (ImageBase), таблиця імпорту після переміщення буде вказувати на null.
4. Додавання у виконуваний файл підготовленої секції модулю захисту, службових даних, необхідних для його роботи, модифікація оригінальних заголовків виконуваного файлу.

Спрощена структура виконуваного файлу до і після інтеграції додаткової секції представлена на рис. 1.


***Рис. 1**. Структура виконуваного файлу до і після захисту*

Алгоритм роботи захищеного виконуваного файлу полягає у наступному. Після запуску модифікованого виконуваного файлу управління передається на точку входу модуля захисту, який першим чином перевіряє цілісність власної конфігурації, заголовків та тіла виконуваного файлу. Перевіряється наявність налагоджувача та виконується авторизація (в залежності від конфігурації системи захисту). Розпаковується оригінальний

програмний код. Заповнюється таблиця адресів імпортованих функцій та передається виконання на оригінальну точку входу виконуваного файлу.

Робота програмного засобу не потребує наявності вихідного коду PE-файлу, який потрібно захистити. Після завантаження програми користувач може отримати доступ до функціональних можливостей за допомогою відповідних кнопок (рис. 2):

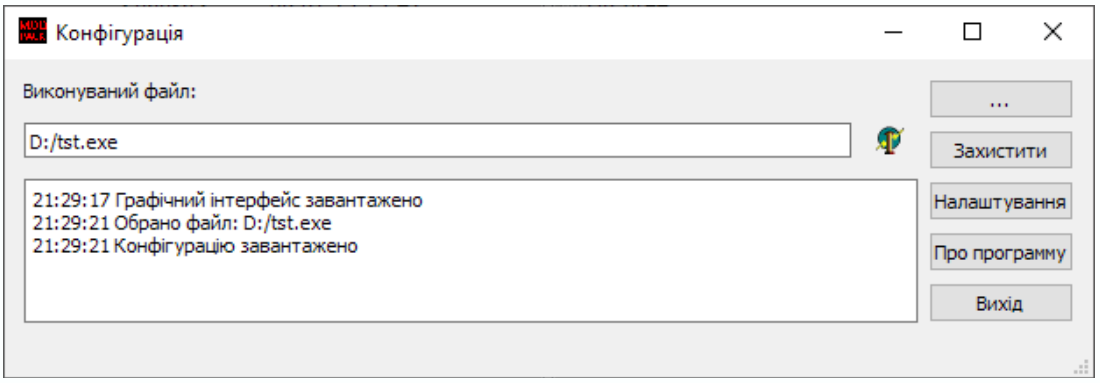


***Рис. 2**. Головне вікно програмного засобу*

Розроблений програмний засіб має такі можливості: вибрати виконуваний файл для захисту (кнопка «…»); накласти захист на вибраний файл (кнопка «Захистити»); налаштувати опції захисту (кнопка «Налаштування»); прочитати інформацію про програму (кнопка «Про програму»); вийти з програми (кнопка «Вихід»). Вікно налаштування конфігурації програмного засобу дозволяє довільно змінювати конфігурацію захисту PE-файлу (рис. 3).

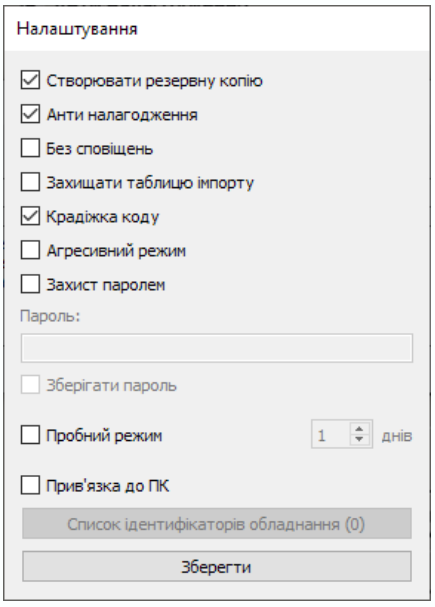


***Рис. 3**. Вікно налаштувань захисних дій програмного засобу*

Опція «Створювати резервну копію» дозволяє на всяк випадок зберегти оригінальний файл виконуваного файлу перед процедурою захисту. Ця корисна функція допоможе користувачу у разі потреби повернутися до незахищеної версії файлу, порівняти функціональність обох файлів (захищеного та незахищеного). Ім'я резервної копії співпадає з ім'ям захищуваного файлу, але має розширення .bak. Опція «Анти налагодження» дозволяє захистити виконуваний файл від налагодження. Опція «Без сповіщень» блокує усі сповіщення користувача, які генерує захист. Опція «Захищати таблицю імпорту» дозволяє захистити таблицю імпорту від її модифікацій, перехоплення та відновлення під час налагодження. Опція «Крадіжка коду» забороняє крадіжку коду з точки входу захищеного програмного засобу з метою унеможливлення створення робочого знімку пам'яті процесу. Опція «Агресивний режим» дозволяє при знаходженні debugger виконувати агресивні дії (блокування вводу, відключення зображення на дисплеї тощо), що спрямовані на ускладнення видалення захисту з програмного продукту. Опція «Захист паролем» дозволяє захистити програмний засіб статичним паролем, який потрібно вводити кожен раз при завантаженні виконуваного файлу. Якщо активована опція «Зберігати пароль», то пароль вводити потрібно тільки один раз. Якщо активовано пробний режим, то пароль буде запитаний тільки після його закінчення. Опція «Пробний режим» дозволяє обмежити термін використання захищеного файлу з моменту його першого завантаження. Опція «Прив'язка до ПК» дозволяє використовувати виконуваний файл тільки на тих комп'ютерах, ідентифікатори обладнання яких збережені у виконуваному файлі.

Під час тестування програмного засобу було виявлено декілька особливостей. Зокрема: опція прив'язки до ідентифікаторів обладнання персонального комп'ютера не буде коректно працювати на віртуальних машинах. Не завжди правильно спрацьовує опція агресивного захисту, тому її рекомендується не встановлювати без поважних причин. При ручній зміні системної дати може некоректно спрацювати функція пробного режиму. Ці недоліки планується усунути у подальшій роботі.

Основною мовою розробки програмного засобу була обрана мова програмування C++. Вона широко використовується для розробки програмного забезпечення, будучи однією з найбільш популярних мов програмування. Область її застосування включає створення операційних систем, драйверів, прикладних програм та ін. Мова програмування C++ підтримує різні парадигми програмування, кросплатформність, надає гнучкі та ефективні засоби визначення нових типів, має високу сумісність з мовою програмування C. Вона дозволяє працювати з мовою Accembler, пам'яттю, адресами на низькому рівні. У якості інтегрованого середовища розробки

додатків було використано середовище Visual C++. Для розробки графічного інтерфейсу користувача була використана кросплатформна бібліотека забезпечення Qt. Вона надає програмісту не тільки зручний набір бібліотек класів, а й певну модель розробки додатків, певний каркас їх структури. Важливою перевагою Qt є добре продуманий та логічний набір класів, який надає програмісту дуже високий рівень абстракції. Налаштування захисту виконуваного файлу зберігаються у структурованому вигляді у форматі Json. Даний формат дозволяє вільно читати та модифікувати конфігурацію у будь-якому текстовому редакторі. Для програмної обробки конфігурації захищеного виконуваного файлу була використана вільна бібліотека JsonCpp.

Слід зазначити, що не тільки програмне забезпечення персональних комп'ютерів потребує захисту. Не менш поширеними є смартфони, які є неодмінним атрибутом сучасної людини. У цьому контексті І. Миронець, В. Пономаренко повідомляють про створення додатку для ОС Android, призначеного для двохфакторної ідентифікації користувача смартфону. Додаток встановлюється на мобільний пристрій, не потребує під'єднання до мережі Інтернет або мережі мобільного оператора, пропонує користувачу для ідентифікації спеціально створену таблицю, порядок елементів якої динамічно змінюється [10].

Крім використання різноманітних утиліт для захисту виконуваних файлів потрібно пам'ятати про організаційні заходи та інформаційну культуру користувачів. Давно відомо, що інформаційна необізнаність користувача може призвести до втрати персональних даних, зламу комп'ютерної техніки, операційної системи [11] тощо. У випадку використання Інтернет без врахування можливих кіберзагроз користувач може стати жертвою Інтернет-шахрайства, крадіжок паролів і файлів, втрати грошових цінностей [5]. На підприємстві доречно створити механізм підзвітності для відслідковування інформаційних потоків, що зменшить ризик втрати економічної інформації [3]. Тільки комплексні заходи, спрямовані на інформаційну безпеку, захист та легальне використання ліцензійного програмного забезпечення, призведе до позитивних наслідків.

**Висновки**. Отже, актуальність захисту програмного забезпечення пояснюється значними втратами, які несуть розробники ліцензійного програмного забезпечення, можливими втратами економічної інформацій підприємствами, а також втратою персональних даних користувачами. Доволі часто зловмисні атаки спрямовані на PE-файли, які працюють під операційною системою Windows. Отримуючи доступ до вихідного коду виконуваного файлу за допомогою декомпіляції або дизасемблювання, зловмисники можуть зробити певні модифікації, знову скомпілювати та розповсюджувати вже змінене програмне забезпечення.

Боротьба проти несанкціонованого використання програмного забезпечення та його модифікації здійснюється через розробку законодавчих документів, відповідальність за використання піратського програмного забезпечення, використання апаратних та програмних методів захисту. Для захисту часто використовуються обфускаційні методи, які дозволяють змінювати вихідний програмний код при збереженні функціональності програми. Ефективними виявилися захисті утиліти, які використовували метод заплуюучих перетворень, інтеграцію цифрового водяного знаку в код програми та ін.

Мета розробленої комп'ютерної програми полягає у захисті виконуваних файлів Windows, що дозволить ускладнити їх реверс-інжиніринг та вносення несанкціонованих змін у вихідний код. Робота програмного засобу передбачає налаштування способів захисту, таких як «Анти налагодження», «Захист таблиці імпорту», «Захист паролем» та ін. Програмний засіб був розроблений за допомогою мови програмування C++, середовища Visual C++, бібліотеки Qt, JsonCpp та інших інструментальних засобів.

Програмний засіб має простий графічний інтерфейс, не потребує від користувача спеціальних знань з інформатики. У подальшій роботі передбачається вдосконалити можливості програмного засобу та перевірити його працездатність в реальних умовах.

***Література:***

1. Стороженко А. О., Горпенюк А. Я., Лужецька Н. М. Методика захисту програмного забезпечення шляхом впровадження ЦВЗ в асемблерний код програми. *Вісник Національного університету «Львівська політехніка»*. Сер.: Автоматика, вимірювання та керування. 2013. № 753. С. 80–84.

2. Бакала А. А. Комп'ютерне піратство: шляхи подолання та міжнародний досвід боротьби з ним. *Часопис цивілістики*. 2016. № 20. С. 207–210.

3. Вітер С. А., Світлишин І. І. Захист облікової інформації та кібербезпека підприємства. *Економіка і суспільство*. 2017. № 11. С. 497–502.

4. Корнага Я. І., Базака Ю. А., Базалій М. Ю. Захист програмного забезпечення за допомогою заплутуючих перетворень. *The scientific heritage*. 2020. № 54-1. С. 72–74.

5. Lubko D., Sharov S., Strokan O. Software development for the security of TCP-connections. *Modern Development Paths of Agricultural Production: Trends and Innovations*. 2019. P. 99–109.

6. Ільєнко А., Ільєнко С., Куліш Т. Перспективні методи захисту операційної системи Windows. *Кібербезпека: освіта, наука, техніка*. 2020. № 4(8). С. 124–134.

7. Tevis J. E., Hamilton Jr J. Static analysis of anomalies and security vulnerabilities in executable files. *Proceedings of the 44th annual Southeast regional conference*. 2006. 560–565.

8. Левчук А.С. Методи захисту від змін та дизассемблювання виконавчих файлів ОС Windows. *Ukrainian Journal of Educational Studies and Information Technology*. 2017. № 5(1). С. 166–169.

9. Степаненко І. В., Кінзерявий В., Наджі А., Лозінський І. Сучасні обфускаційні методи захисту програмного коду. *Безпека інформації*. 2016. № 22(1). С. 32–37.

10. Миронець І. В., Пономаренко В. М. Автоматизована система захисту програмного забезпечення для операційної системи Android. *Вісник Черкаського державного технологічного університету*. 2020. № 1. С. 43–49.

11. Янішевський Д. Актуальні питання захисту інформації в комп'ютерних системах і мережах. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*. 2004. № 8. С. 81–85.

***References:***

1. Storozhenko A. O., Horpeniuk A. Ya., Luzhetska N. M. (2013). Metodyka zakhystu prohramnoho zabezpechennia shliakhom vprovadzhennia TsVZ v asemblernyi kod prohramy [The method of software protection by implementing the TsVZ in the assembly code of the program]. *Visnyk Natsionalnoho universytetu "Lvivska politekhnika" – Bulletin of the Lviv Polytechnic National University, 753,* 80-84 [in Ukrainian].

2. Bakala A. (2016). Kompiuterne piratstvo: shliakhy podolannia ta mizhnarodnyi dosvid borotby z nym [Computer piracy: the way of negotiation and international experience the strife of IT]. *Chasopys tsyvilistyky – Journal of Civil Studies, 20,* 207-210 [in Ukrainian].

3. Viter S. A., Svitlyshyn I. I. (2017). Zakhyst oblikovoi informatsii ta kiberbezpeka pidpryiemstva. Ekonomika i suspilstvo [Protection of accounting information and cyber security of the enterprise]. *Ekonomika i suspilstvo – Economy and society, 11,* 497-502 [in Ukrainian].

4. Kornaha Ya. I., Bazaka Yu. A., Bazalii M. Yu. (2020). Zakhyst prohramnoho zabezpechennia za dopomohoiu zaplutuiuchykh peretvoren [Software protection using through confusing transformations]. *The scientific heritage, 54-1,* 72-74 [in Ukrainian].

5. Lubko D., Sharov S., Strokan O. (2019). Software development for the security of TCP-connections. *Modern Development Paths of Agricultural Production: Trends and Innovations*. P. 99–109.

6. Ilyenko A., Ilyenko S., Kulish T. (2020). Perspektyvni metody zakhystu operatsiinoi systemy Windows [Prospective protection methods of windows operation system]. *Kiberbezpeka: osvita, nauka, tekhnika – Cyber security: education, science, technology, 4(8),* 124-134 [in Ukrainian].

7. Tevis J. E., Hamilton Jr J. (2006). Static analysis of anomalies and security vulnerabilities in executable files. *Proceedings of the 44th annual Southeast regional conference,* 560–565.

8. Levchuk A.S. (2017). Metody zakhystu vid zmin ta dyzassembliuvannia vykonavchykh failiv OS Windows [Methods of protection against modification and disassembly of executable files of the Windows OS]. *Ukrainian Journal of Educational Studies and Information Technology, 5(1).* 166–169 [in Ukrainian].

9. Stepanenko I. et al. (2016). Suchasni obfuskatsiini metody zakhystu prohramnoho kodu [Modern obfuscation methods for secure coding]. *Bezpeka informatsii – Information security, 22(1),* 32-37 [in Ukrainian].

10. Myronets I., Ponomarenko V. (2020). Avtomatyzovana systema zakhystu prohramnoho zabezpechennia dlia operatsiinoi systemy Android [Automated system of software protection for android operation system]. *Visnyk Cherkaskoho derzhavnoho tekhnolohichnoho universytetu – Bulletin of the Cherkasy State Technological University, 1,* 43-49 [in Ukrainian].

11. Yanishevskyi D. (2004). Aktualni pytannia zakhystu informatsii v kompiuternykh systemakh i merezhakh [Current issues of information protection in computer systems and networks]. *Pravove, normatyvne ta metrolohichne zabezpechennia systemy zakhystu informatsii v Ukraini – Legal, normative and metrological support of the information protection system in Ukraine, 8,* 81–85 [in Ukrainian].