

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

Корчинський Дмитро Сергійович

Допускається до захисту:
Завідувач кафедри
інформаційних технологій,
д.т.н., доцент,
_____ Нескородєва Т. В.
«__» _____ 20__ р.

ПРОТИДІЯ ВІРУСАМ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
СИСТЕМАХ ТА ІОТ

Спеціальність 125 Кібербезпека
Кваліфікаційна (бакалаврська) робота

Науковий керівник:
Крижановський В.Г.,
д. т. н., професор, професор
кафедри інформаційних технологій

(підпис)

Оцінка : _____ / _____ / _____
(бали/за шкалою ЕКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця 2022

АНОТАЦІЯ

Корчинський Д.С. Протидія вірусам в інформаційно-комунікаційних системах та IoT. Спеціальність 125 Кібербезпека. Донецький національний університет імені Василя Стуса. Вінниця. 2022 рік.

У бакалаврській роботі проведено аналітичний огляд комп'ютерних вірусів, визначено актуальність протидії вірусам у інформаційно-комунікаційних системах та IoT та обґрунтовано та обрано середовище розробки ПЗ. Розроблено та протестовано систему протидії вірусам в ІКС та IoT. Знешкодження загроз відбувається автоматично за бажанням адміністратора мережі чи пристрою IoT. Тестування системи проводилось на тестовому вірусі *EICAR*.

Ключові слова: протидія вірусам, інформаційно-комунікаційні системи, інтернет речей, IoT.

Рис.: 14, Бібліограф.: 45 найм.

Korchynskyi D.S. Anti-virus in information and communication systems and IoT. Specialty 125 Cybersecurity. Vasyl Stus Donetsk National University. Vinnitsa. 2022.

The bachelor's thesis offers an analytical review of computer viruses, the relevance of anti-virus in information and communication systems and IoT was determined, and the software development environment was substantiated and selected. A system for combating viruses in ICS and IoT has been developed and tested. Threat removal is done automatically at the request of the network administrator or IoT device. The system was tested on the *EICAR* test virus.

Keywords: anti-virus, information and communication systems, IoT.

Fig.: 14, Bibliographer: 45 items

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД КОМП'ЮТЕРНИХ ВІРУСІВ.....	8
1.1. Визначення комп'ютерного вірусу.....	8
1.2. Боротьба із вірусами.....	9
1.2.1. Антивірусна програма Microsoft Security Essentials.....	11
1.2.2. Захист Microsoft 365.....	12
1.3. Класифікація комп'ютерних вірусів.....	12
1.3.1. Комп'ютерні хробаки.....	12
1.3.2. Програмне забезпечення-вимагач та фішинг.....	13
1.3.3. Боротьба із захистом електронної пошти та хмарними загрозами.....	14
1.3.4. Вірус веб-скриптів.....	15
1.3.5. Вірус-троян.....	16
1.4. Оцінка загроз, що представляють комп'ютерні віруси.....	16
РОЗДІЛ 2. АКТУАЛЬНІСТЬ ПРОТИДІЇ ВІРУСАМ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ ТА ІоТ.....	18
2.1. Поширення вірусів у інформаційно-комунікаційних системах та ІоТ.....	18
2.1.1. Структура систем ІоТ.....	19
2.1.2. Додатки для особистого чи ділового спілкування та їх уразливості.....	19
2.2. Використання Інтернету речей у різних галузях економіки.....	20
2.2.1. Інтелектуальні системи у енергетичному секторі.....	23
2.2.2. Інтелектуальні системи на транспорті.....	24
2.3. Системи та засоби протидії вірусам у ІоТ.....	25
2.3.1. Вразливість систем через використання слабких паролів.....	26
2.3.2. Комплексний аудит безпеки.....	26
2.3.3. Особливості керування пристроями ІоТ.....	27

2.3.4. Додаткові методи безпеки.....	29
РОЗДІЛ 3. ТЕОРІЯ РОЗРОБКИ ПЗ ДЛЯ ІОТ У EMBARCADERO DELPHI..	31
3.1. Історія та особливості використання мови програмування Delphi.....	31
3.2. Архітектура візуального контролю.....	33
3.3. Розробка для Інтернету речей.....	32
3.3.1. Використання BeaconFence для створення ПЗ ІоТ.....	35
РОЗДІЛ 4. РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ПРОТИДІЇ	
ВІРУСАМ В ІКС ТА ІОТ.....	38
4.1 Особливості, структура та застосування файлу вірусу EICAR.....	38
4.1.1. Виявлення тестового вірусу різними антивірусними програмами	39
4.1.2. Основне призначення тестового вірусу EICAR.....	39
4.2. Написання коду у середовищі Embarcadero.....	41
4.3. Тестування системи протидії вірусам в ІКС та ІоТ.....	43
4.3.1. Робота системи на виявлення вірусів.....	44
4.3.2. Робота системи на знешкодження вірусів.....	45
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А. Програмний лістинг.....	55

ВСТУП

Інтернет - дивовижний ресурс для пошуку. За один раз можна знайти чудове місце для літніх канікул або з'ясувати, хто був акторським складом фільму, який запам'ятався.

Повсякденні об'єкти, такі як годинники, побутова техніка та автомобілі, розумні дома тощо, підключаються до комунікаційних мереж - Інтернету речей (IoT) - для надання цілого ряду послуг та додатків, таких як особиста медична допомога, розумні електромережі, спостереження, домашня автоматизація та розумний транспорт тощо. Очікується, що кількість підключених пристроїв IoT зросте з 8,4 мільярдів у 2017 році до понад 25 мільярдів до 2023 році.

Хакери та розробники шкідливих програм уважно стежать за популярними пошуковими запитами та намагаються вмістити небезпечні сторінки у результати пошуку. Натиснувши шкідливе посилання комп'ютер може стати солдатом-зомбі в армії ботів. Користувач зненацька може виявити, що його банківський або криптовалютний рахунок вичерпано. Програми-вимагачі можуть зіпсувати ваші фотографії із відпустки або роботу за останні роки. Визнавши це, потрібен антивірусний захист, що є дуже **актуальним** сьогодні для Інтернету речей.

Метою роботи є комплексний аналіз поширення комп'ютерних вірусів у інформаційно-комунікаційних системах та IoT та розробка системи захисту від них.

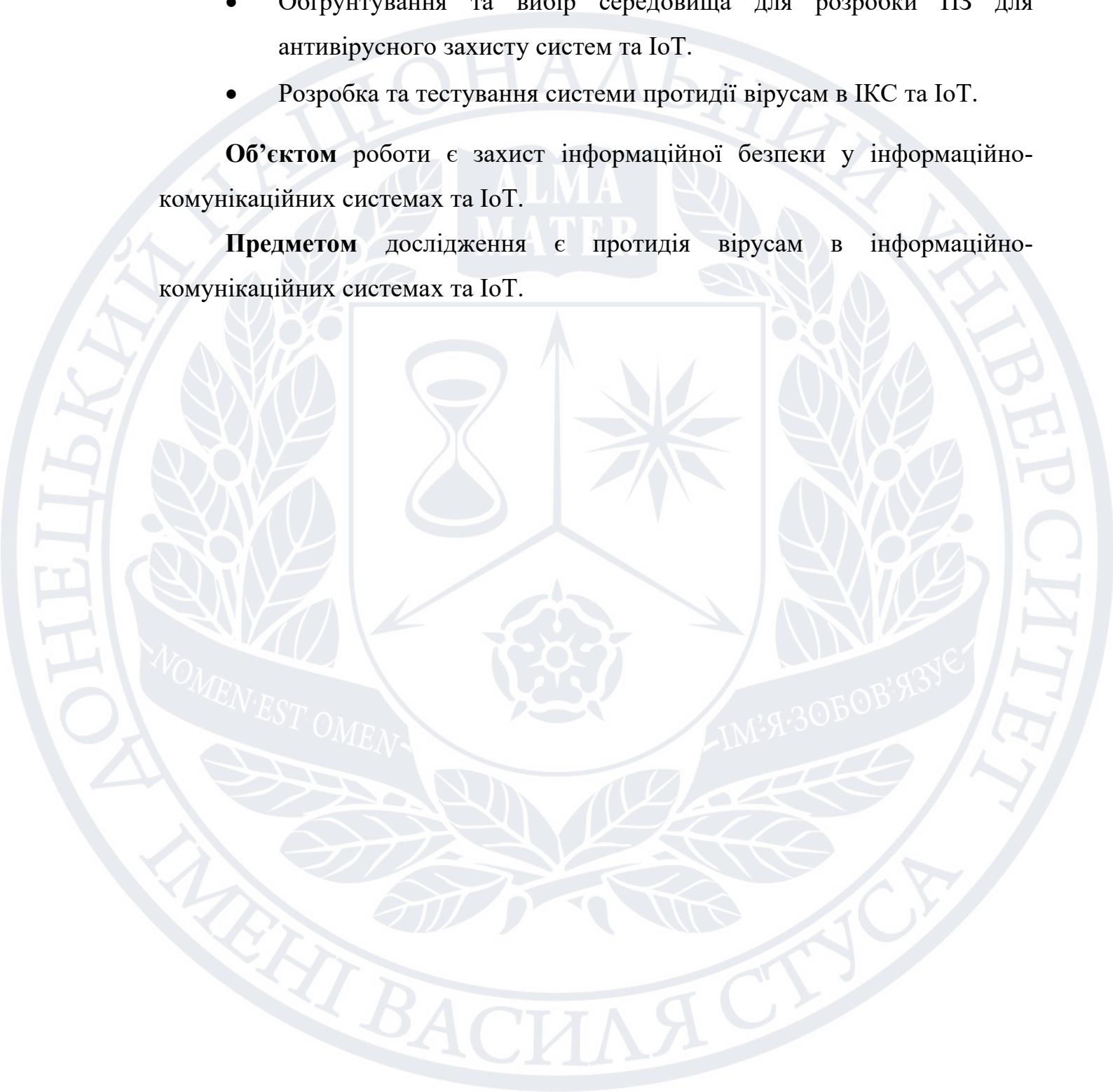
Завданням роботи є:

- Аналітичний огляд комп'ютерних вірусів як тип комп'ютерної програми, яка при виконанні реплікується, змінюючи інші програми та вставляючи власний код, аналіз основних методів боротьби з вірусами, їх класифікацію.

- Аналіз існуючих інструментів боротьби з вірусами, сфери їх поширення у інформаційно-комунікаційних системах та IoT.
- Обґрунтування та вибір середовища для розробки ПЗ для антивірусного захисту систем та IoT.
- Розробка та тестування системи протидії вірусам в ІКС та IoT.

Об'єктом роботи є захист інформаційної безпеки у інформаційно-комунікаційних системах та IoT.

Предметом дослідження є протидія вірусам в інформаційно-комунікаційних системах та IoT.



РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД КОМП'ЮТЕРНИХ ВІРУСІВ

1.1. Визначення комп'ютерного вірусу

Комп'ютерним вірусом більшість засобів масової інформації (ЗМІ) та звичайні кінцеві користувачі називають кожен шкідливий програму, про яку повідомляють у новинах. На щастя, більшість шкідливих програм не є вірусами. Комп'ютерний вірус змінює файли хоста (або вказівники на них) таким чином, що під час виконання файлу вірусу також запускається вірус.

Комп'ютерний вірус - це тип комп'ютерної програми, яка при виконанні реплікується, змінюючи інші програми та вставляючи власний код [1]. Якщо ця реплікація вдається, уражені ділянки називаються «зараженими» комп'ютерним вірусом, метафора, що походить від біологічних вірусів [2].

Комп'ютерний вірус – це шкідлива програма або авторський код, що використовується для виконання руйнівної діяльності на пристрої чи локальній мережі. Шкідлива діяльність коду може пошкодити локальну файлову систему, викрасти дані, переривати служби, завантажити додаткове шкідливе програмне забезпечення або будь-які інші дії, заcodedовані в програмі автором шкідливого програмного забезпечення. Багато вірусів видають себе за законні програми, щоб обманом змусити користувачів виконати їх на своєму пристрої, доставляючи корисне навантаження комп'ютерного вірусу [1, 3].

Вірус прямої дії отримує доступ до основної пам'яті комп'ютера та заражає всі програми, файли та директорії, розташовані на шляху `autoexec.bat`, перш ніж видалити себе. Цей вірус зазвичай змінює продуктивність системи, але здатний знищити всі дані на жорсткому диску комп'ютера та будь-якому USB-пристрої, підключеному до нього. Вірусів

прямої дії можна уникнути за допомогою антивірусних сканерів. Їх легко виявити, як і відновлення заражених файлів (рис. 1.1):

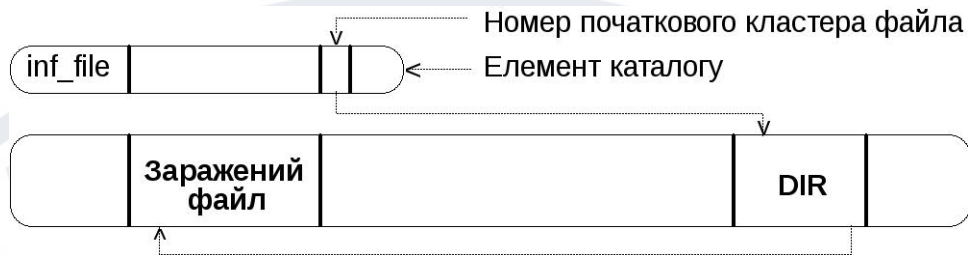


Рис. 1.1. Схема дії вірусу DIR

Чисті комп'ютерні віруси сьогодні є рідкістю, складаючи менше 10% усіх шкідливих програм. Це добре: віруси є єдиним типом шкідливих програм, які «заражають» інші файли. Через це їх особливо важко очистити, оскільки зловмисне програмне забезпечення має запускатися з законної програми. Це завжди було нетривіально, а сьогодні це майже неможливо. Найкращим антивірусним програмам важко зробити це правильно, і в багатьох (якщо не в більшості) випадках замість цього просто поміщають на карантин або видаляють заражений файл [2, 3-5].

1.2. Боротьба із вірусами

Сьогодні комплексна кібербезпека є обов'язковою для всіх пристроїв - настільних комп'ютерів, ноутбуків, планшетів, смартфонів, мереж та IoT-речей. Щоб бути ефективними, рішення кібербезпеки повинні забезпечувати захист у реальному часі для всіх дій, від електронної пошти до перегляду Інтернету, а не лише періодичного сканування жорсткого диска. Окрім того, найкращі сучасні програмні продукти безпеки не є статичними одноразовими інсталяціями із періодичними оновленнями. Якісний продукт кібербезпеки надається як послуга, відома як SaaS (Software-as-a-Service). Це означає, що на додаток до моніторингу пристроїв у режимі реального часу, саме

програмне забезпечення оновлюється у режимі реального часу із найновішою інформацією про існуючі та нові загрози, як їх запобігти та як усунути пошкодження [4, 6-8].

Ключем до вирішення проблеми компромісу є точна оцінка швидкості та масштабів вірусних інфекцій. Динаміка розповсюдження комп'ютерних вірусів потребує моделювання та аналізу процесу їх поширення. Після фундаментальних досліджень було запропоновано велику кількість моделей розповсюдження комп'ютерних вірусів, починаючи від моделей поширення на популяційному рівні та моделей поширення на рівні мережі та до моделей розповсюдження на індивідуальному рівні. Зокрема, особливий тип моделей розповсюдження, відомий як моделі Ssceptible-Infected (SI), особливо підходять для фіксації процесу поширення нового цифрового вірусу до випуску відповідного антивірусу.

Єдиний надійний спосіб уникнути «невидимих» вірусів - це завантаження із середовища, яке, як відомо, чисте. Після цього можна використовувати програмне забезпечення безпеки для того, щоб перевірити неактивні файли операційної системи. Більшість програмного забезпечення безпеки базується на сигнатурі вірусів, або вони використовують евристику [1, 10]. Програмне забезпечення безпеки може також використовувати базу даних хешів файлів для файлів ОС Windows, тому програмне забезпечення безпеки може ідентифікувати змінені файли та запитувати інсталяційний носій Windows, щоб замінити їх справжніми версіями. У старих версіях Windows використовуються функції криптографічного хешування файлів ОС Windows, які зберігаються в бібліотеці Windows, та дозволяють перевірку цілісності та автентичності файлів, можуть перезаписуватись так, щоб засіб перевірки системних файлів повідомляв, які змінені системні файли є автентичними. Сканування на наявність змінених файлів не завжди гарантує виявлення вірусів (рис. 1.2) [11].



Рис. 1.2. Схема дії файлового вірусу

Атаки програм-вимагачів, як-от вірус CryptoLocker 2013 року, досяг піку у 2017 році. Це зловмисне програмне забезпечення атакувало понад 250000 комп'ютерів, зашифрувавши всі файли. Як результат, на комп'ютері користувача відображалася червона записка про викуп, яка повідомляла, що важливі файли зашифровані на цьому комп'ютері і відкривалось платіжне вікно [12].

1.2.1. Антивірусна програма Microsoft Security Essentials

Приклади антивірусного програмного забезпечення Microsoft Windows включають додаткову програму Microsoft Security Essentials [13] (для Windows XP, Vista і Windows 7). Окрім того, кілька ефективних антивірусних програм доступні для безкоштовного завантаження з Інтернету (зазвичай обмежені для некомерційного використання) [14]. Деякі такі безкоштовні програми майже так само ефективні, як і комерційні конкуренти. Поширеним уразливостям безпеки присвоюються ідентифікатори CVE і пересилаються у Національну базу даних уразливостей. Secunia PSI [13], яка є прикладом

безкоштовного програмного забезпечення для особистого використання і яка перевірить ПК на наявність уразливого застарілого програмного забезпечення та спробує його оновити.

1.2.2. Захист Microsoft 365

Рішення Microsoft 365 за допомогою доступу Cloud EdgeSecure до корпоративних ресурсів захищають віддалених працівників.

Файли Microsoft Office можуть запускати макроси які використовують для завантаження додаткових шкідливих програм або запуску шкідливого коду. Макровіруси передають корисне навантаження, коли файл відкривається і макрос запускається [15].

1.3. Класифікація комп'ютерних вірусів

Основні принципи класифікації вірусів приведено на рис. 1.3 [19]:



Рис. 1.3. Принципи класифікації вірусів

1.3.1. Комп'ютерні хробаки

Комп'ютерний хробак – це шкідливе програмне забезпечення, як і вірус, але хробак отримує свою копію та поширює її іншим користувачам.

Хробаки також можуть доставляти корисне навантаження та вичерпувати ресурси оперативної пам'яті. Наприклад, поштовий хробак надсилає свою копію кожному зі списку контактів електронної пошти інфікованого користувача. Коли він потрапляє у папку вхідних одержувачів, кожен, хто запускає хробак, надсилає його до свого списку контактів. Поштові хробаки використовують пам'ять комп'ютера та дуже швидко поширюються Інтернетом, тому створюють проблеми інакші, ніж вірус [20].

1.3.2. Програмне забезпечення-вимагач та фішинг

Попередження про програмне забезпечення-вимагач і фішинг з'являються як прес-релізи на дошці оголошень Центру скарг на злочини у Інтернеті. Програма-вимагач - це вірус, який публікує на екрані користувача повідомлення про те, що екран або система залишається заблокованими або непридатними для використання, доки не буде здійснено платіж (рис. 1.4):

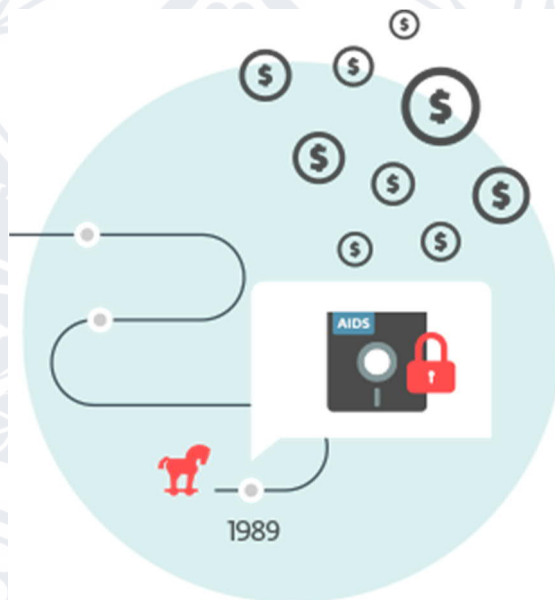


Рис. 1.4. Програмне забезпечення-вимагач

Фішинг – це обман, у якому зловмисник видає себе за друга, експерта із комп'ютерної безпеки чи іншої доброзичливої особи із метою переконати цільову особу розкрити паролі чи іншу особисту інформацію [21].

Ключем до захисту від комп'ютерних вірусів є наявність кількох рівнів захисту кібербезпеки. Наявність брандмауера або антивіруса – це чудовий початок, але це не всеосяжне рішення. У плані кібербезпеки слід враховувати кілька рівнів, як-от навчання користувачів, виявлення кінцевої точки та відповідь, фільтрація DNS тощо.

Програми-вимагачі часто шкодять компаніям, лікарням, поліцейським відділам та навіть цілим містам.

1.3.3. Боротьба із захистом електронної пошти та хмарними загрозами

Сьогодні є активним завдання захисту користувачів від загроз у електронній пошті та хмарних сховищ за допомогою розумного та цілісного підходу. Системним адміністраторам необхідно виявляти, протистояти атакам та повідомляти про них до того, як буде завдано шкоди. Також необхідно запобігти втраті даних через недбалі, скомпрометовані та зловмисні інсайдери, корелюючи вміст, поведінку та загрози [22].

Ці гнучкі віруси, які іноді називають «багатокомпонентними вірусами», стають найбільш поширеними. Хоча більшість вірусів атакують центральний процесор ПК або файли користувачів, багатокомпонентний може робити і те, й інше. Цьому універсальному вірусу важко запобігти, що пояснює швидкі темпи його зростання у останні роки. Зазвичай вони поширюються через файли .exe -та такі програми, як Word і Excel. Як дізнатися, чи комп'ютер заражений багатокомпонентним вірусом? Ці віруси «з'їдають» віртуальну пам'ять, як ніщо інше, тому необхідно очікувати повідомлення, що комп'ютер має низький об'єм віртуальної пам'яті та раптового сповільнення роботи комп'ютера [23].

Вірус також може надсилати посилання на веб-адресу як миттєве повідомлення усім контактам (наприклад, адресам електронної пошти друзів та колег), які зберігаються на зараженій машині. Якщо одержувач, думаючи, що посилання надійшло від друга (довіреного джерела), переходить за посиланням на веб-сайт, вірус, розміщений на сайті, може заразити цей новий комп'ютер та продовжити поширення [6]. Віруси, які поширюються за допомогою міжсайтових сценаріїв, вперше були виявлені у 2002 році [7] та були науково обґрунтовані у 2005 році. Було кілька випадків міжсайтових скриптових вірусів таких веб-сайтів, як MySpace із хробаком Samy та Yahoo! [24].

1.3.4. Вірус веб-скриптів

Вірус веб-скриптів атакує безпеку веб-браузера, дозволяючи хакеру вводити на веб-сторінки шкідливий код або сценарії на стороні клієнта. Це дозволяє кіберзлочинцям атакувати основні веб-сайти, такі як сайти соціальних мереж, постачальників послуг електронної пошти та будь-який сайт, на якому користувачі можуть вводити або переглядати будь-яку інформацію. Зловмисники можуть використовувати вірус, щоб розсилати спам, здійснювати шахрайські дії та пошкоджувати файли сервера. Захист від веб-скриптів залежить від розгортання програмного забезпечення для захисту веб-браузера у режимі реального часу, використання безпеки файлів cookie, вимкнення сценаріїв та використання інструментів для видалення шкідливого програмного забезпечення.

Зловмисне програмне забезпечення, яке шифрує дані та тримає їх як заручників, очікуючи на окупність криптовалюти, становили величезний відсоток шкідливого ПЗ за останні кілька років, і цей відсоток все ще зростає [22].

1.3.5. Вірус-троян

Від троянських програм важко захиститися із двох причин: їх легко писати (кіберзлочинці регулярно розробляють та розповсюджують набори для створення троянів) та поширюються шляхом обману кінцевих користувачів, які не можуть зупинити поширення вірусу. Щомісяця автори шкідливих програм викачують троянів мільйонами. Постачальники антишкідливого програмного забезпечення намагаються боротися із троянами.

Вірус зазвичай приєднується до програми, файлу або завантажувального сектора жорсткого диска. Як тільки вірус прикріплюється до цього файлу або програми, вони заражаються [23].

Трояни зазвичай надходять електронною поштою або надсилаються користувачам, коли вони відвідують заражені сайти. Найпопулярнішим типом троянів є підроблена антивірусна програма, яка спливає та стверджує, що пристрій заражений, а потім інструктує запуск програми для очищення ПК. Користувачі ковтають приманку та троян запускається.

1.4. Оцінка загроз, що представляють комп'ютерні віруси

Розмір загрози, яку представляють комп'ютерні віруси, можна визначити лише шляхом оцінки багатьох різних факторів [9]. Ці фактори включають відносну легкість, із якою можна записати комп'ютерний вірус, мотивацію написання комп'ютерного вірусу, шкоду та накладні витрати, завдані зараженими системами, а також правові наслідки комп'ютерних вірусів. Згідно із дослідженнями [10], розробка комп'ютерного вірусу вимагає більше наполегливості, ніж технічного досвіду. Це лякаюча заява для комп'ютерної спільноти.

Навчання комп'ютерних спеціалістів щодо небезпеки, яку представляють віруси для ефективної роботи комп'ютерної індустрії у цілому, підкреслюється як засіб гальмування поточного поширення комп'ютерних вірусних програм. Розроблено рекомендації, які допомагають користувачам комп'ютерів запобігти зараженню комп'ютерними вірусами. Ці рекомендації підтримують надійні загальні методи комп'ютерної безпеки як засобу боротьби із комп'ютерними вірусами [24].



РОЗДІЛ 2. АКТУАЛЬНІСТЬ ПРОТИДІЇ ВІРУСАМ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ ТА ІoT

2.1. Поширення вірусів у інформаційно-комунікаційних системах та ІoT

Більшість пристроїв і служб Інтернету речей піддаються ряду поширених загроз, які обговорювалися у розділі 1, як-от віруси та атаки відмови у обслуговуванні. Вжиття простих кроків для уникнення таких загроз та боротьби із вразливими місцями системи недостатньо. Отже, необхідне забезпечення безперебійного процесу реалізації політики, що підтримується жорсткими процедурами у реальному часі [25].

Основні елементи архітектури систем активного аудиту представлено на рис. 2.1:

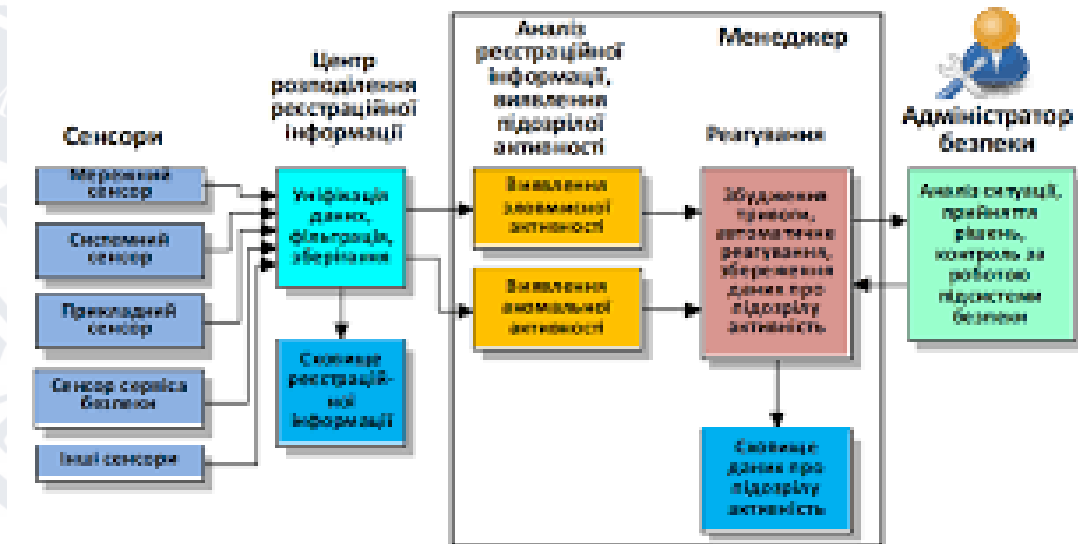


Рис. 2.1. Основні елементи архітектури систем активного аудиту

2.1.1. Структура систем IoT

Системи IoT базуються на двох основних компонентах: системне обладнання та системне програмне забезпечення. Обидва мають досить часто недоліки у їх розробці. Уразливість обладнання дуже важко визначити, а також важко виправити. Уразливості програмного забезпечення можна знайти в операційних системах, прикладному програмному забезпеченні та керуючому програмному забезпеченні, наприклад, у протоколах зв'язку.

Існує ряд факторів, які призводять до недоліків у розробці програмного забезпечення, включаючи людський фактор і складність програмного забезпечення. Технічні вразливості зазвичай виникають через людські слабкості та необізнаність. Результати нерозуміння вимог призводять до запуску проекту без плану, погану комунікацію між розробниками та користувачами, брак ресурсів, навичок та знань, а також неможливість керувати системою та контролювати її [26].

2.1.2. Додатки для особистого чи ділового спілкування та їх уразливості

Оскільки пристрої IoT тісно пов'язані між собою, хакерам потрібно лише використати одну вразливість, щоб маніпулювати всіма даними, роблячи їх непридатними для використання. Виробники, які не оновлюють свої пристрої регулярно залишають їх уразливими для кіберзлочинців.

Додатки для особистого чи ділового спілкування, побудовані на основі концепції виявлення присутності у Інтернеті, виявляють, коли суб'єкт може спілкуватися. Ці програми дозволяють співпрацювати за допомогою текстового чату, аудіо, відео або передачі файлів. Також були виявлені віруси, які дозволяють динамікам прослуховувати та згодом записувати розмови, щоб передати їх підслуховувачу. Отже, може бути гарною ідеєю

тримати систему безпеки у окремій мережі, до якої голосовий помічник не має доступу [27].

2.2. Використання Інтернету речей у різних галузях економіки

У табл. 2.1 наведено дані Beecham Research, що дають уявлення про область охоплення IoT [28].

Таблиця 2.1 - Дані використання IoT у різних галузях

Сектори послуг	Прикладні групи	Розташування	Приклади пристроїв
ІТ та мережі	Публічні	Послуги, е-комерція, центри даних, мобільний зв'язок, дротовий зв'язок, ISP	Сервери, сховища, РС, маршрутизатори, комутатори, PBX
	Корпоративні		
Безпека, охорона	Корпоративні	ІТ/центри даних, офіси, приватні мережі	
	Устаткування стеження, контроль	Радари/супутники, військова безпека, безпілотники, зброя, транспорт, кораблі, літаки, спорядження	Танки, винищувачі, бойові комплекти зв'язку, джипи
	Громадська інфраструктура	Люди, тварини, пошта, їжа/здоров'я, упаковка, багаж, підготовка води, екологія будівель, загальна екологія	Автомобілі, дорожні робітники, служби безпеки, пожежні, екологічний моніторинг
	Аварійні служби	Устаткування і персонал, поліція, пожежники, регулятори	Машини швидкої допомоги, машини аварійних служб
Роздрібна торгівля	Спеціалізовані	АЗС, ігрові клуби, боулінг, кіно, дискотеки, спецзаходи	Касові термінали, інтернетбанкінг, бірки, знаки, торгові автомати

	Туризм і громадське харчування	Готелі, ресторани, бари, кафе, клуби	
	Магазини	Супермаркети, торгові центри, поодинокі магазини, центри дистрибуції	
Транспорт	Неавтомобільний	Повітряний, залізничний, морський	Машини, освітлення, кораблі, літаки, знаки, митниця
	Автомобільний	Легкові, вантажні, будівельна техніка, позашляховики	
	Транспортні системи	Система оплати, управління трафіком, навігація	
Промисловість	Розподіл	Трубопроводи, конвеєри, обробка матеріалів	Насоси, клапани, чани, конвеєри, двигуни, приводи, перетворення, виробництво, складання/упаковка, ємності, танки
	Перетворення, дискретне	Метал, папір, гума, пластик, металовироби, електронні плати, тестування	
	Процеси	Нафтохімія, вуглеводні, їжа, напої	
	Автоматизація ресурсів	Гірнична справа, іригація, сільське господарство, лісове господарство	
Охорона здоров'я та науки про життя	Охорона здоров'я	Лікарні реанімації, мобільні станції, клініки, лабораторії, кабінети лікарів	MRI, КПК, імпланти, хірургічне обладнання, насоси, монітори, телемедицина
	Домашні системи	Імпланти, домашні системи моніторингу	

	Дослідження	Розробка ліків, діагностика, лабораторії	
Споживчий сектор і будинок	Інфраструктура	Проводка, мережевий доступ, управління енергоспоживанням	Цифрові фотоапарати, енергосистеми, посудомийки, електронні книги, настільні комп'ютери, пральні машини, датчики, лампочки, телевізори, MP3, ігрові приставки, освітлення, сигналізація
	Безпека	Охоронні системи/сигналізації, пожежна безпека, екобезпека, для людей похилого віку, для дітей, захист енергопостачання	
	Комфорт та розваги	Кондиціонери, освітлення, приставки, розважальні системи	
Енергетика	Попит/пропозиція	Виробництво енергії, передача і розподіл, низьковольтні мережі, якість енергії, управління енергією	Турбіни, вітряки, UPS, батареї, генератори, датчики, акумулятори
	Альтернативні джерела	Сонячна, вітрова, когенерація, електрохімічна	
	Нафта та газ	Платформи, бурові, гірлове обладнання, насоси, трубопроводи	
Споруди	Комерційні, організацій	Офіси, освіта, торгівля, громадське харчування, охорона здоров'я, аеропорти, стадіони	ОВКВ, транспорт, пожежна безпека, освітлення, охорона, доступ
	Промисловість	Виробничі, чисті, кампуси	

2.2.1. Інтелектуальні системи у енергетичному секторі

Енергетичний сектор був одним з перших, хто прийняв цифрові технології. У 1970-х роках енергетичні компанії були піонерами цифрових технологій, використовуючи новітні технології для полегшення управління мережею та її експлуатації. Нафтогазові компанії вже давно використовують цифрові технології для покращення прийняття рішень щодо геологорозвідувальних і видобувних активів. Промисловий сектор десятиліттями використовує засоби керування процесами та автоматизацію, особливо у промисловості, щоб максимізувати якість при мінімізації споживання енергії [29].

Огляд ключових інформаційних систем у енергетичному секторі демонструє різноманітні способи, якими цифрові технології можуть впливати на робочі місця. Загалом, цифровізація, ймовірно, призведе до подальшої ефективності у ланцюжку поставок, але із меншою ймовірністю замінить все ще значні потреби у робочій силі для основних інженерних та будівельних робіт. Роботи, які складаються із великої частки автоматизованих завдань - наприклад, пов'язані з передбачуваною, рутинною та повторюваною фізичною діяльністю, а також зі збором та обробкою даних - можуть бути піддані більшому ризику автоматизації, ніж ті, що з менш рутинною діяльністю. Працівникам, які підтримують цифрову інфраструктуру, знадобляться спеціалізовані навички ІКТ, такі як кодування та кібербезпека, в той час як у енергетичному секторі всім працівникам знадобляться загальні навички ІКТ для роботи із цифровими технологіями. Додаткові навички, такі як навички лідерства, комунікації та роботи у команді, стануть усе більш важливими для зростаючої кількості можливостей для спільної роботи за допомогою ІКТ. Темпи та масштаби цифровізації та її вплив на робочі місця у енергетичній системі залишаються дуже невизначеними й залежатимуть від ряду факторів, які будуть відрізнятися у залежності від регіонального та галузевого контекстів [30].

2.2.2. Інтелектуальні системи на транспорті

Інтелектуальні транспортні системи використовують цифрові технології у всіх видах транспорту для підвищення безпеки, надійності та ефективності. Темпи цифровізації у галузі збільшуються. За останні кілька років інвестиції транспортних компаній у цифрові технології різко зросли. Наприклад, глобальні інвестиції в інфраструктуру та програмне забезпечення із 2014 року зросли на понад 20% щорічно, досягнувши 47 мільярдів доларів США у 2016 році.

На даний момент транспорт забезпечує 28% світового кінцевого попиту на енергію та 23% глобальних викидів CO₂ від спалювання палива. Відповідно до Центрального сценарію МЕА, кінцеве споживання енергії для транспорту зросте майже вдвічі до 165 кДжоулів у 2060 році, причому більшість попиту надходить із вантажних автомобільних транспортних засобів (36%) та пасажирських малотоннажних транспортних засобів (28%). У всіх видах транспорту, цифрові технології допомагають підвищити енергоефективність та зменшити витрати на технічне обслуговування. У авіації новітні комерційні літаки оснащені тисячами датчиків, які генерують майже терабайт даних під час середнього польоту [31].

Аналітика оптимізує планування маршрутів та може допомогти пілотам приймати рішення під час польоту та зменшити споживання палива. Кораблі також оснащуються більшою кількістю датчиків, які допомагають екіпажу вжити заходів для оптимізації маршрутів, а прогрес у супутниковому зв'язку дає змогу покращити зв'язок. Найбільш революційні зміни від цифровізації можуть відбутися у автомобільному транспорті, де повсюдне підключення та технології автоматизації можуть кардинально змінити те, як переміщуються люди та товари.

Автоматизовані технології керування транспортними засобами можуть покращити їх безпеку завдяки різноманітним датчик та можливостям автоматизованого прийняття рішень, які можуть замінити людський контроль. Наслідки мобільності ACES для енергії та викидів дуже невизначені. Вони залежатимуть від сукупного ефекту змін у поведінці

споживачів, втручання технологічного прогресу та технології транспортних засобів. Останні дослідження оцінюють широкий спектр можливих результатів. Наприклад, у довгостроковій перспективі, за найкращого сценарію підвищення ефективності за рахунок автоматизації та розподілу поїздок, споживання енергії може зменшитися вдвічі порівняно із поточним рівнем. І навпаки, якщо підвищення ефективності не буде матеріалізовано, а ефекти від автоматизації призведуть до значно більше подорожей, споживання енергії може збільшитися більш ніж вдвічі [32].

2.3. Системи та засоби протидії вірусам у IoT

Вірусні атаки на системи IoT часто є цілеспрямованими атаками, які використовують шлях входу до IoT з метою закріплення всередині системи. Серед найбільш резонансних випадків - хробак Stuxnet, який використовувався для маніпулювання центрифугами всередині ядерних об'єктів у Ірані та BlackEnergy [33], який вплинув на потужності виробництва електроенергії в Україні. Незважаючи на те, що більшість атак зосереджені на крадіжці даних або промислового шпигунстві, обидва вищезгадані випадки продемонстрували кінетичний ефект шкідливого ПЗ. У офіційному документі Trend Micro під назвою Cyber Threats to the Mining Industry досліджується, як гірничодобувна промисловість все частіше стає об'єктом кампаній кібершпигунства [34]. Ці кампанії кібершпигунства розроблені для отримання найновіших технічних знань та розвідки, які допоможуть деяким групам інтересів процвітати та підтримувати конкурентні переваги.

Зазвичай брандмауер Windows вимикають через дратівливі спливаючі вікна та сповіщення, а потім просто забувають про нього. Якщо перезапустити його, необхідно перейти до панелі керування, далі до «Система та безпека» та обрати «Брандмауер Windows». Якщо

використовується Мас, то необхідно перейти до «Системні налаштування», потім «Безпека та конфіденційність», потім вкладку «Брандмауер» та увімкнути брандмауер на Мас.

2.3.1. Вразливість систем через використання слабких паролів

Поширена вразливість у системах Інтернету речей сьогодні пов'язана зі слабкими або незмінними паролями за замовчуванням. Погане керування обліковими даними пристрою ставить пристрої IoT під більший ризик стати об'єктами вірусної атаки. Непослідовні методи управління дозволяють здійснювати атаки, орієнтовані на паролі. Наприклад, паролі співробітників можуть не відповідати розширеним політикам керування паролями.

У 2018 році прийнято Каліфорнійський закон SB-327 IoT про заборону на використання сертифікатів за замовчуванням. Цей закон, нарешті, має на меті вирішити проблему використання слабких паролів. Поки виробники Інтернету речей повністю не усвідомлять потребу у цих змінах, обладнання безпеки IoT залишається за користувачами, постачальниками послуг IoT та IT-сервісами.

Безпосередній крок до захисту цих систем полягає у тому, щоб IT-адміністратори налаштували нові політики входу, які вимагають від користувачів та адміністраторів змінювати паролі пристроїв за замовчуванням. Ця політика означає додавання шарів особливих та складних комбінацій символів перед перерозподілом їх у живі середовища.

2.3.2. Комплексний аудит безпеки

Аудит безпеки - це систематична оцінка безпеки пристрою або послуги шляхом вимірювання того, наскільки вони відповідають набору встановлених критеріїв. Через багато помилок та вразливостей у більшості систем аудит безпеки відіграє важливу роль у визначенні будь-яких

недоліків, які можна використати, які ставлять дані під загрозу. В IoT потреба у аудиті системи залежить від програми та її цінності.

Відсутність керування пристроями та відсутність підтримки безпеки на пристроях, запущених у виробництві, включно з керуванням активами, оновленнями, безпечним виведенням із експлуатації, моніторингом систем та можливостями реагування є основним завданням безпеки IoT. Одним з найбільш значущих ризиків і проблем безпеки IoT є керування всіма нашими пристроями та закриття периметра.

Однак шахрайські пристрої або підроблені шкідливі пристрої Інтернету речей встановлюються у захищених мережах без авторизації. Зловмисний пристрій замінює або інтегрує оригінальний пристрій як член групи для збору або зміни конфіденційної інформації. Ці пристрої розривають периметр мережі.

2.3.3. Особливості керування пристроями IoT

Управління пристроями подібне до інших систем управління IT-активами: першочергові завдання - це надання, експлуатація та оновлення пристроїв. Ці проблеми стосуються всіх пристроїв, включаючи шлюзи. Виявлення та ідентифікація пристроїв IoT є необхідним першим кроком у моніторингу та захисту цих пристроїв. Великі мережі IoT, що містять багато майже ідентичних пристроїв, є привабливою мішенню для кіберзловмисників.



Рис. 2.2. Еталонна модель IoT по Рекомендації Y.2060

На рис. 2.2 зображена еталонна модель IoT від МСЕ-Т, що складається з чотирьох рівнів плюс можливості управління і безпеки, що діють між рівнями. До цього часу говорилось про рівень пристрою. У термінах функціональності зв'язку рівень пристрою включає в себе, грубо кажучи, фізичний та канальний рівні OSI.

Однак відновлюватися після вірусних атак звичайними засобами дорого і повільно, насамперед, якщо ці пристрої розташовані на великій географічній території, де адміністраторам мережі або операторам доводиться під'їжджати до пристроїв, щоб відновити їх вручну. Застаріла статична інвентаризація активів IoT, що контролює систему, далека від ефективного управління безпекою. Ідентифікація пристроїв за допомогою традиційних функцій IT-пристроїв, таких як IP-адреси та базові операційні системи, не працює для IoT. Лише шляхом визначення конкретного пристрою можна точно спланувати вимоги до доступу до мережі, тактику розгортання, оптимізацію стратегії безпеки та оперативні плани. Після визначення ідентифікаційних даних пристроїв системи безпеки можуть

відстежувати поведінку пристрою у контексті робочого процесу організації, а не розглядати його як динамічні IP-адреси невідомого типу пристрою.

Рішення безпеки IoT дозволяють організаціям виявляти та ідентифікувати пристрої IoT у своїх мережах. Незважаючи на значне зростання кількості IoT-активів, більшість організацій не знають про вразливості пристроїв і не керують питаннями безпеки чи профілями ризиків. Інтелектуальне сканування та профілювання пристроїв дають змогу командам із IT-безпеки бачити мережеві пристрої IoT, їхні профілі ризиків та поведінку у мережі під час взаємодії із іншими пристроями у мережі. Сучасні рішення безпеки IoT використовують, як правило, машинне навчання для визначення пристроїв IoT та для виявлення шкідливих шаблонів мережевого зв'язку [33].

2.3.4. Додаткові методи безпеки

Окрім використання цих методів безпеки, користувачі також повинні знати про нові розробки та технології. Останнім часом безпеці Інтернету речей приділяють більше уваги. Постійно проводяться дослідження щодо того, як захистити певні галузі, відстежувати загрози, пов'язані із Інтернетом речей й підготуватися до майбутніх змін, таких як 5G. Користувачі повинні розуміти, що IoT є активною та розвиваючою сферою, тому її безпеку завжди доведеться трансформувати та пристосовуватися до змін [34].

Хоча цифровізація може принести багато позитивних переваг, вона також може зробити IoT системи більш вразливими до вірусних атак. На сьогоднішній день порушення роботи IoT систем через повідомлення про вірусні атаки були відносно невеликими. Однак організовувати вірусні атаки стає легше та дешевше.

Конфіденційність та право власності на дані також є серйозними проблемами для користувачів, особливо тому, що більш детальні дані збираються зі все більшої кількості підключених пристроїв та приладів.

Наприклад, дані про споживання енергії у домогосподарствах, зібрані за допомогою розумних лічильників, можна використовувати, щоб визначити, коли хтось перебуває вдома, коли приймає душ чи робить чай. У той же час зведені та анонімізовані індивідуальні дані про споживання енергії можуть покращити розуміння IoT систем, такі як профілі навантаження та допомагають знизити витрати окремих споживачів..

Також важливою є перевірка налаштувань безпеки та конфіденційності за замовчуванням. Якщо є небажані функції, можна вимкнути їх. Варто вирішити, чи потрібно вимкнути мікрофони на деяких пристроях, наприклад, якщо не потрібно використовувати на них керування голосом. Це завадить злоумисникам слухати розмови.

Відсутність можливості безпечного оновлення пристрою включає відсутність перевірки мікропрограмного забезпечення на пристрої, відсутність безпечної доставки (незашифрована під час доставки), відсутність механізмів запобігання відкату та відсутність повідомлень про зміни безпеки через оновлення.

РОЗДІЛ 3. ТЕОРІЯ РОЗРОБКИ ПЗ ДЛЯ ІОТ У EMBARCADERO DELPHI

3.1. Історія та особливості використання мови програмування Delphi

Delphi 10 Seattle має найбільший потенціал з часів Delphi 7. Завдяки розширеній пам'яті для IDE та підтримці Windows 10 можна ефективно використовувати Delphi. За словами провідних фахівців, система дозволяє розробникам легко розробляти додатки, які розуміють точне місцезнаходження та переміщення користувача у межах визначених розробником полігональних зон. Також дійсно дає можливості для нових інтерактивних додатків для найрізноманітніших галузей [35].

Раніше часто використовувалися інструменти компанії Borland. Delphi був досить стабільним, C++ Builder був набагато більш помилковим. Розробники ПЗ оновлюють старі проекти Delphi до новішої IDE Delphi із деякими встановленими пакетами оновлень.

Відгуки, які отримувала компанія Embarcadero, були надзвичайно позитивними. Спеціалісти компанії розглядають можливість впровадження деяких змін у інфраструктуру за рекомендаціями користувачів [36].

Система розробки ПЗ Embarcadero - це вдосконалена IDE для сучасної мови програмування із високопродуктивними бібліотеками, які допомагають створювати надзвичайно швидкі нативні програми для IoT. Незалежні розробники та корпоративні команди виграють від використання системи для досягнення послідовних поставок проектів принаймні в 5 разів швидше, ніж рішення-конкуренти. Постійна безпека інфраструктури та даних була основним пріоритетом компанії на початку проекту [37].

У останній версії продукту є особливості та обмеження, але з IDE користувачам найлегше працювати та створювати продуктивні програмні продукти. Як правило, розробники віддають перевагу цій системі перед будь-

яким іншим. Також система дозволяє міжплатформну розробку FireMonkey та розробку для IoT.

3.2. Архітектура візуального контролю

Основна архітектура візуального контролю FireMonkey була значно перероблена, щоб уможливити кілька реалізацій презентації для кожного елемента керування, який називається ControlTypes - зокрема, можна використовувати власні презентації керування ОС. Нова базова архітектура базується на MVC та має зворотну сумісність, що дозволяє розробникам під час розробки обирати між типами елементів керування стилями та платформою для кожного елемента керування на підтримуваних елементах керування. Це дозволяє вибрати, чи буде контроль реалізований за допомогою графічного процесора FireMonkey або відтворений базовою операційною системою. У будь-якому випадку можна використати той самий API керування [38].

У Builder можна створювати програми VCL Forms, FireMonkey Desktop та FireMonkey Metropolis за допомогою майстра проекту. Звичайно, можна створити три різні програми «Hello World» приблизно за три хвилини і всі вони викликають вбудовану глобальну функцію швидкого доступу під назвою ShowMessage("вставити повідомлення тут"). Зберігання файлів зі значущими іменами займає більше часу, ніж сам код.

Впровадження додаткових елементів керування VCL надає широкі можливості інтерфейсу Windows для розробників Delphi та C++Builder для оновлення своїх програм Windows [39].

3.3. Розробка для Інтернету речей

Функціональність існуючих бізнес-додатків можна легко розширити для інтегрування мобільних пристроїв та нові гаджети IoT, які надають нові

рішення. Корисні IoT та бізнес-додатки більше не є окремими (прив'язаними до одного мобільного додатка), а розподілені на кількох рівнях, гаджетах та пристроях, таких як платформи операційних систем, включаючи Windows, Mac, iOS, Android, а також проміжне програмне забезпечення, хмари, сервери та корпоративні послуги [40]. Розширення існуючих додатків є великою перемогою для розробників бізнес-додатків, оскільки вони можуть включати рішення IoT, зберігаючи наявну інфраструктуру та використовуючи наявні та великі кодові бази. Практично всі галузі можуть відразу отримати вигоду від створення підключених додатків, таких як роздрібна торгівля, харчові послуги, охорона здоров'я, виробництво та промислова автоматизація.

Для пристроїв Інтернету речей середовище програмування надає можливість програмному забезпеченню безпечної підтримки модема. Розробник може створити програму для невеликого пристрою, який підключається за допомогою Bluetooth або WiFi. За допомогою цих функцій можна створити програму для годинника, наприклад фітнес-трекер, яка синхронізує операції та дані із іншою програмою на телефоні чи ПК. Модем додатків також можна використовувати для промислових пристроїв [41].

Embarcadero Technologies, постачальник програмних рішень для розробки додатків і баз даних, оголосив про низку нових технологій, включаючи новий інтерфейс Windows та продукти для журналів обміну. Embarcadero нещодавно придбала Raize Components, виробника CodeSite та Konopka Signature VCL Controls, які є двома новими додатковими рішеннями Embarcadero для розробників.

Embarcadero CodeSite - це рішення для розробників під Delphi, C++Builder і Visual Studio, яке допомагає розробникам налагоджувати розгорнуті програми Windows. Продукт Konopka Signature VCL Controls - це компонентне рішення Windows UI для розробників в середовищах Delphi та C++Builder [42].

Програмні компоненти Raize довгий час вважалися найякіснішими елементами керування інтерфейсом Windows, доступними для розробників.

Придбання продуктів і технологій Raize посилює постійну прихильність Embarcadero розробникам Windows Delphi і C++Builder, а також популярному фреймворку VCL, який сьогодні підтримує мільйони додатків Windows. Елементи керування Konopka Signature VCL дозволяють швидко розробляти сучасні інтерфейси користувача для додатків Windows. Продукт має понад 200 елементів керування інтерфейсом Windows.

Завдяки повній підтримці стилів VCL розробники можуть розробляти користувацькі та сучасні програми Windows 10. Продукт також містить понад 100 користувацьких редакторів для спрощення дизайну та розробки інтерфейсу Windows. Система ведення журналу CodeSite дає розробникам Visual Component Library (VCL), FireMonkey і NET більш глибоке уявлення про те, як виконується їхній живий код. Це допомагає розробникам ефективніше знаходити проблеми і забезпечувати правильну роботу програми. Класи журналу CodeSite дозволяють розробникам фіксувати інформацію під час виконання їх коду, а потім надсилати цю інформацію на дисплей або у файл журналу. Обидва стилі ведення журналів - ведення журналу у реальному часі та журналювання файлів - можна виконувати локально або віддалено. CodeSite підтримує популярні рамки розробки та інтегровані середовища розробки (IDE), включаючи RAD Studio Embarcadero та Visual Studio від Microsoft [43].

З випуском Windows 11 настав ідеальний час для розробників подумати про нову та розширену розробку Windows, оскільки Microsoft надає Windows більше форм-факторів та розширює її за допомогою Bluetooth LE, IoT та підтримки мобільних пристроїв, - сказав старший директор із розробки програм Embarcadero. Розробники можуть скористатися можливістю створити UI/UX для підключених додатків, які автоматично стилізовані для Windows 11 за допомогою елементів керування Konopka Signature VCL. Embarcadero також анонсувала BeaconFence, рішення для розробників на основі геолокації користувачів і керування зонами для внутрішніх і зовнішніх приміщень. Ця технологія дозволяє розробникам і незалежним

постачальникам програмного забезпечення легко додавати інтерактивну підтримку користувачів і пристроїв до програм iOS, Android, Windows і OS X.

3.3.1. Використання BeaconFence для створення ПЗ IoT

Embarcadero стверджує, що, хоча BeaconFence можна використовувати в широкому діапазоні застосувань, він особливо добре підходить для роздрібної торгівлі, охорони здоров'я, розваг та виробничих прикладних рішень. Близькі компоненти BeaconFence відстежують й повідомляють про місцезнаходження користувача або пристрою, присутність, вхід чи вихід з зони та переміщення як у приміщенні, так і на вулиці. Це дає змогу розробляти інтерактивні програми, які використовують кінцевого користувача, клієнта або пристрій поблизу або всередині будь-якого фізичного місця, будівлі чи об'єкта. Використовуючи стандартні iBeacons і AltBeacons, BeaconFence підтримуються прості радіальні зони, а також полігональні зони практично будь-якої форми або розташування, повідомляє компанія. Розробники просто імпортують креслення, фотографію із висоти пташиного польоту або креслення макета фізичного розташування, візуально розміщують маяки та малюють зони. Потім BeaconFence забезпечує відстеження місцезнаходження користувача або пристрою, включаючи події входу, виходу із зони.

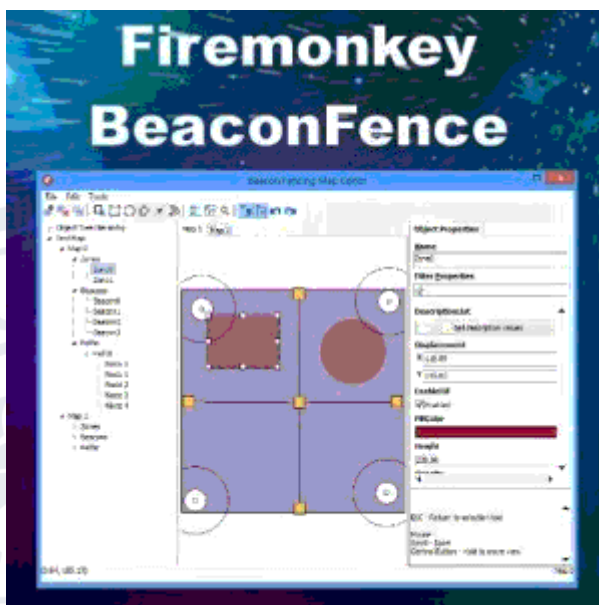


Рис. 3.1. Новий набір компонентів під назвою BeaconFence

Підтримка інтерактивної близькості історично була віднесена до науково-фантастичних фільмів, але останні кроки вперед, як правило, обмежувалися сценаріями на відкритому повітрі, а підтримка всередині приміщень, як правило, вимагала дорогих власних приймачів і міток RFID. За допомогою BeaconFence розробники тепер можуть забезпечити високоточне відстеження користувачів як у приміщенні, так і на вулиці за допомогою стандартних телефонів і планшетів із підтримкою Bluetooth, які користувачі вже мають, а також готових маяків, які можна легко встановити в будь-якому місці. Крім того, BeaconFence дозволяє розробникам легко розробляти додатки, які використовують точне місцезнаходження та переміщення користувача в межах визначених розробником полігональних зон, уможливлуючи рішення, які виходять далеко за межі традиційного маяка та підтримки RFID близькості [44].

BeaconFence дійсно відкриває нові можливості для нових інтерактивних додатків для найрізноманітніших галузей. BeaconFence - це один із кількох інструментів Embarcadero, які допомагають розробникам створювати програми Інтернету речей IoT. Керування місцезнаходженням користувача та часовим поясом є критичним при розробці наскрізних рішень

IoT. Під час спонсованого Embarcadero опитування 1040 розробників, проведеного Dimensional Research на початку 2021 року, 77 відсотків опитаних команд розробників заявили, що у 2022 році вони будуть активно розробляти рішення IoT та майже половина розробників Інтернету речей очікують, що їхні рішення будуть мати вплив на бізнес проекти [45].



РОЗДІЛ 4. РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ПРОТИДІЇ ВІРУСАМ В ІКС ТА ІoT

Розробку системи протидії почнемо зі завантаження стандартного файлу вірусу EICAR, який служить для перевірки роботоздатності антивірусу [46].

4.1. Особливості, структура та застосування файлу вірусу EICAR

EICAR (або EICAR-Test-File - від European Institute for Computer Antivirus Research) - стандартний файл, який використовується для перевірки працездатності антивірусу. Насправді вірусом не є. Будучи запущеним як COM-файл DOS, лише виводить текстове повідомлення та повертає управління DOS. Програма працює у середовищах, що підтримують виконання 16-бітного програмного забезпечення для DOS, таких як MS-DOS, OS/2, Windows 9x і 32-бітові Windows NT.

Хоча COM-файли у загальному випадку є бінарними, EICAR містить тільки символи ASCII. Тому будь-який користувач може переконатися у працездатності свого антивірусу, набравши у текстовому редакторі (наприклад, у Блокноті) тестовий рядок завдовжки 68-128 байт та зберігши його із розширенням .EXE або .COM. Символи CR/LF, які редактор може додати до кінця файлу, не впливають на роботу EICAR. Зазвичай, якщо резидентний монітор антивірусу увімкнено, після натискання кнопки «Зберегти» виводиться попередження.

4.1.1. Виявлення тестового вірусу різними антивірусними програмами

Антивірус, який виявив цей рядок, повинен вчинити точно так, як і при виявленні реального вірусу. Тому про те, що тривога навчальна, антивірус зазвичай повідомляє у назві вірусу:

EICAR Test-NOT virus! (avast!),
EICAR-Test-File (Антивірус Касперського),
EICAR Test File (Not a Virus!) (Doctor Web),
EICAR-AV-Test (Sophos),
EICAR_Test_File (RAV),
Eicar_test_file (Trend Micro),
Eicar-Test-Signature (Avira AntiVir),
EICAR_Test_File (FRISK),
EICAR_Test (+356) (Grisoft),
Eicar-Test-Signature (ClamAV),
Eicar.Mod (Panda Cloud Antivirus),
VIRUS:DOS/EICAR_Test_File (Microsoft Security Essentials).
Eicar тест файл (NOD32)
Teststring.Eicar (Comodo Internet Security, Comodo AntiVirus)
EICAR_test_file (Virus) (Outpost Security Suite)
Вкрай рідко зустрічаються антивіруси, які не реагують на цей тест.

4.1.2. Основне призначення тестового вірусу EICAR

Зрозуміло, EICAR не перевіряє, наскільки оперативно розробники реагують на віруси та наскільки якісно виліковуються заражені файли - для цього потрібна база даних свіжих вірусів. Його завдання інше: продемонструвати працездатність антивірусної системи та вказати, які об'єкти перевіряються антивірусом, а які – ні. Наприклад: є підозра, що

комп'ютер заражений. Чи діє резидентний монітор, чи вірус зумів його відключити?

Звичайний поштовий черв'як на зразок VBS.LoveLetter повинен для зараження пройти кілька стадій: прийти на поштовий сервер за протоколом SMTP; завантажитись на комп'ютер за протоколом POP3; записатися до бази поштового клієнта; за командою користувача розпакуватися у тимчасовий файл та запуститися. На якій стадії його буде помічено?

Існує багато способів «протягнути» шкідливу програму повз «очі» антивірусу: закодувати в Base64, вкласти в OLE-об'єкт Microsoft Word, в RAR, JPEG, стиснути пакувальником на зразок UPX. Що із цього антивірус розпакує?

Окрім того, антивіруси бувають не тільки локальні, а й мережеві - перевіряючи мережевий трафік; при помилці конфігурування вони або завантажуватимуть сервер зайвою роботою, або, навпаки, пропускатимуть шкідливі файли.

Просто щоб побачити реакцію антивірусу: так, у старих версіях антивірусу Касперського при виявленні вірусу був гучний свинячий вереск [47].

Для того, щоб перевірити, якою буде реакція антивірусу, звичайно, можна застосувати і "живий" вірус - але це "як запалювання урни для перевірки пожежної сигналізації". Для цього і був запропонований стандартизований файл, що не несе шкідливого навантаження.

Структура даного COM-файлу представлена нижче:

*X50!P%#@AP[4\PZX54(P^)7CC)7}\$EICAR-STANDARD-ANTIVIRUS-
TEST-FILE!\$H+H**

4.2. Написання коду у середовищі Embarcadero

Для створення системи протидії вірусам використаємо систему розробки ПЗ Embarcadero. Лістинг програми представлено у Додатку А. Процес написання коду подано на рис. 4.1:

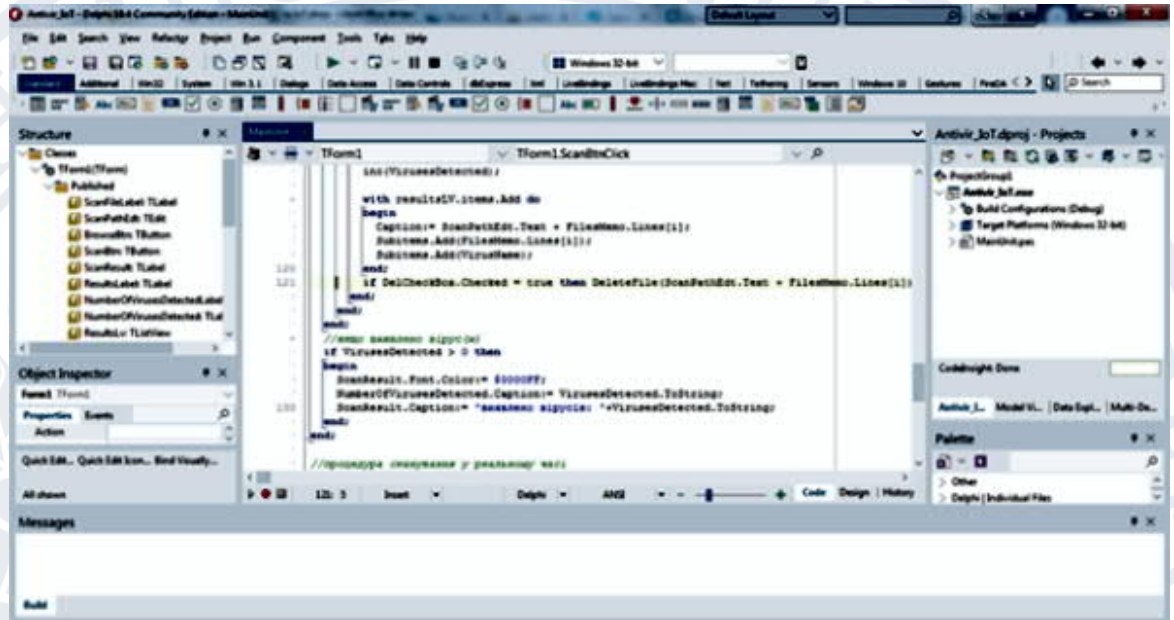


Рис. 4.1. Процес написання коду

Процес розробки графічного інтерфейсу системи представлено на рис. 4.2:

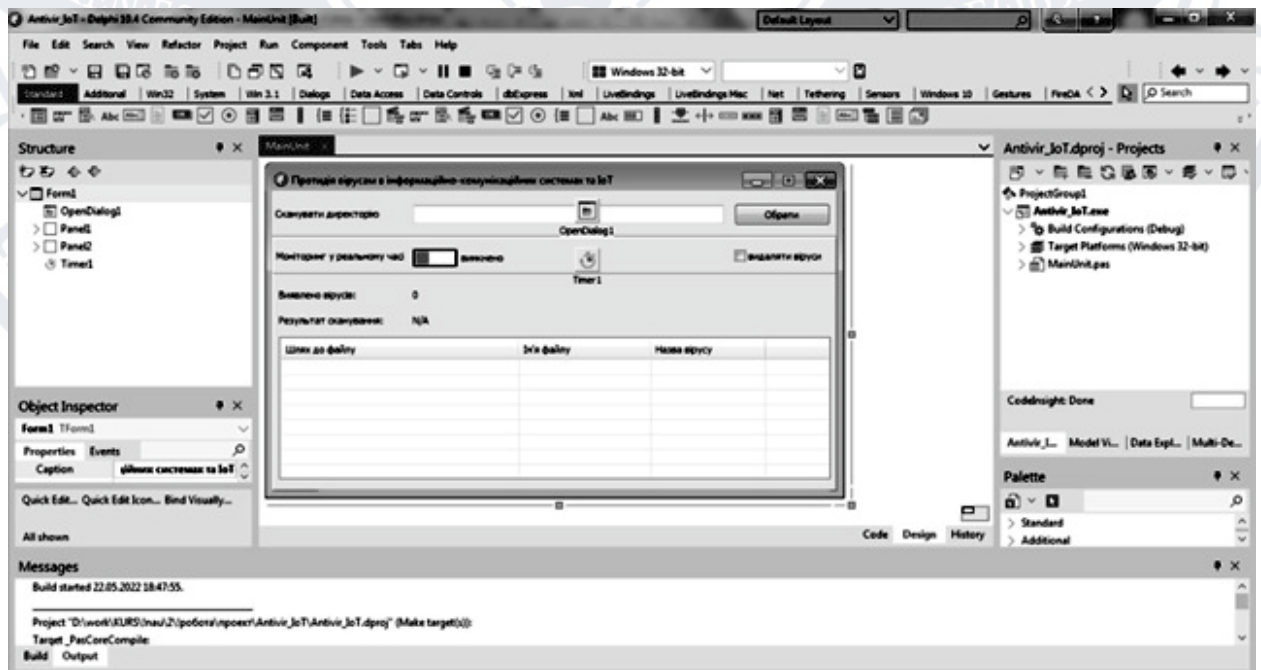


Рис. 4.2. Процес розробки графічного інтерфейсу

Після написання коду виконаємо компіляцію для отримання загрузочного файлу .exe файлу системи протидії вірусам (рис. 4.3):

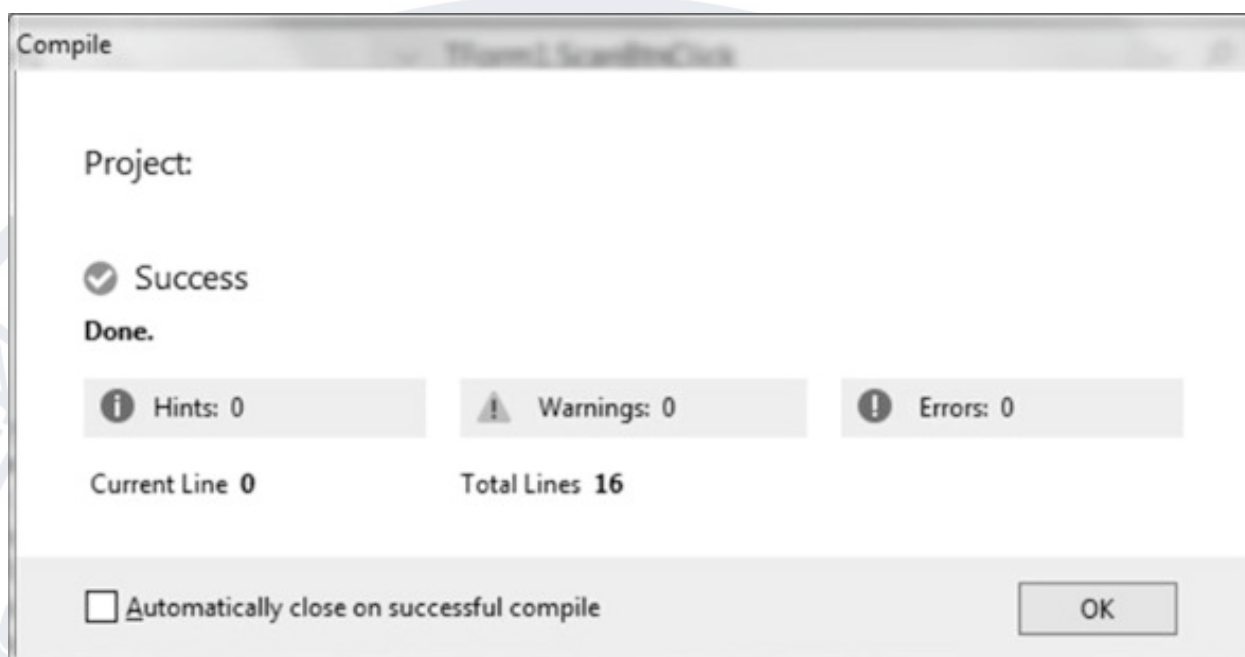


Рис. 4.3. Компіляція вихідного коду

4.3. Тестування системи протидії вірусам в ІКС та IoT

Розпочнемо тестування розробленої системи протидії вірусам. Для цього створимо на диску D директорію Antivir_IoT, у яку помістимо раніше скомпільований файл Antivir_IoT.exe. Далі запустимо його від імені адміністратора. Зовнішній вигляд розробленої системи представлено на рис. 4.4:

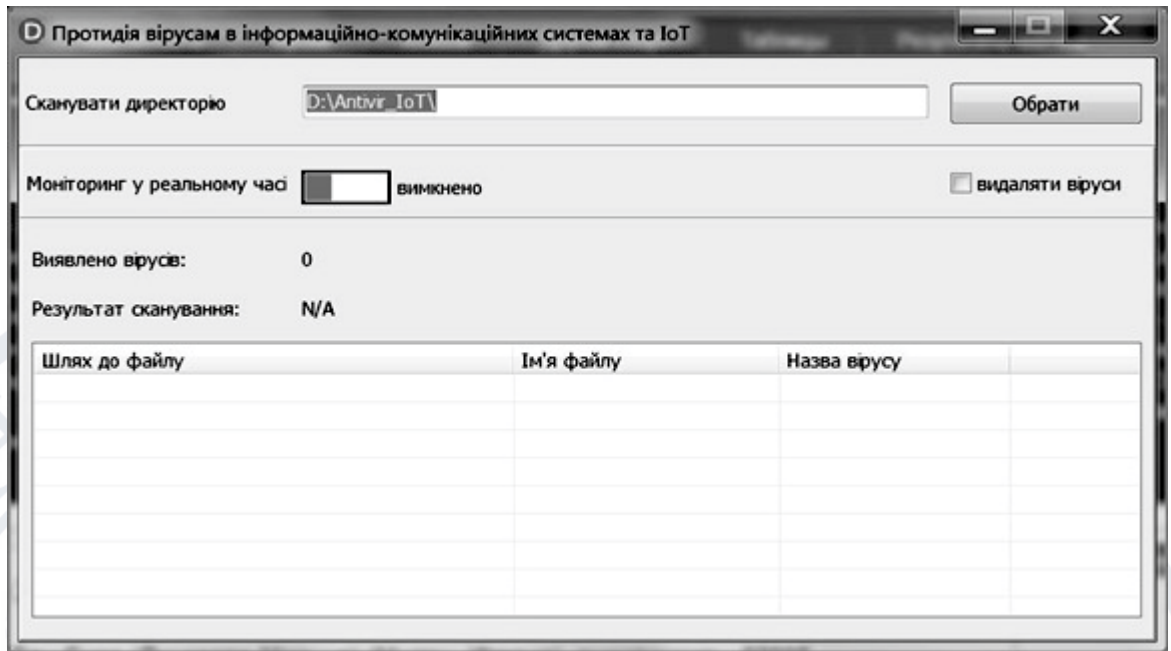


Рис. 4.4. Зовнішній вигляд розробленої системи

Перемістимо перемикач на панелі “Моніторинг у реальному часі” у положення “Ввімкнено”. Це дозволить сканувати, виявляти та, за необхідністю, видаляти шкідливі файли (рис. 4.5):

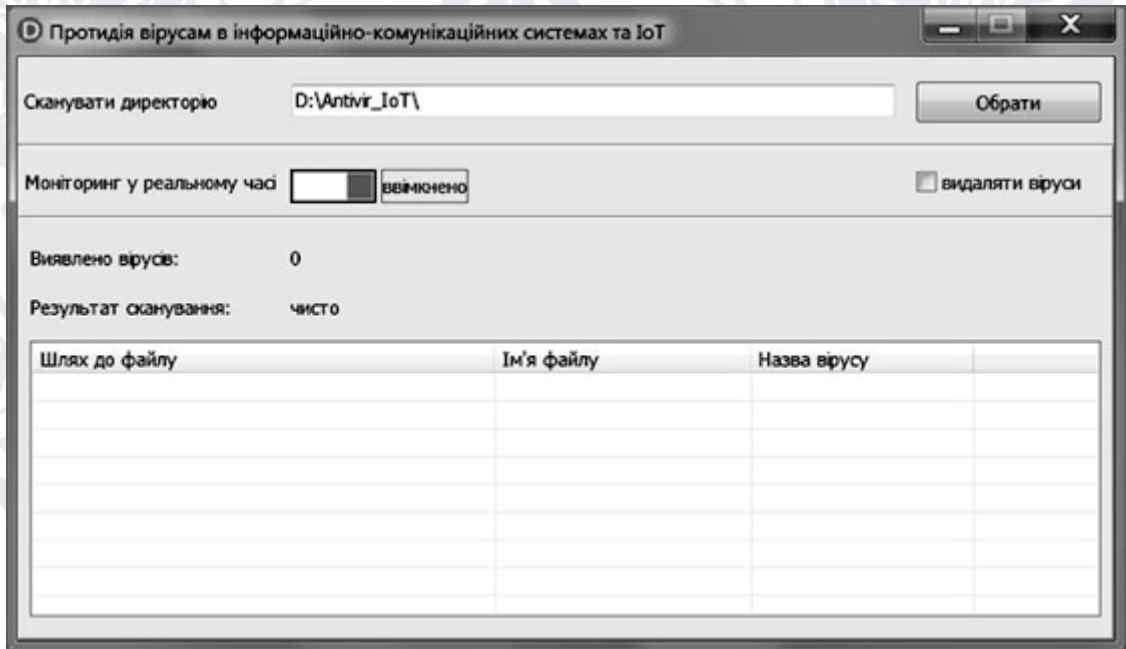


Рис. 4.5. Ввімкнення моніторингу вірусів у реальному часі

4.3.1. Робота системи на виявлення вірусів

Перевіримо роботу системи на виявлення вірусів. Для цього скопіюємо у директорию Antivir_IoT раніше завантажений файл EICAR. Перейменуємо його на virus.txt для наочності. Через проміжок часу побачимо реакцію системи (рис. 4.6):

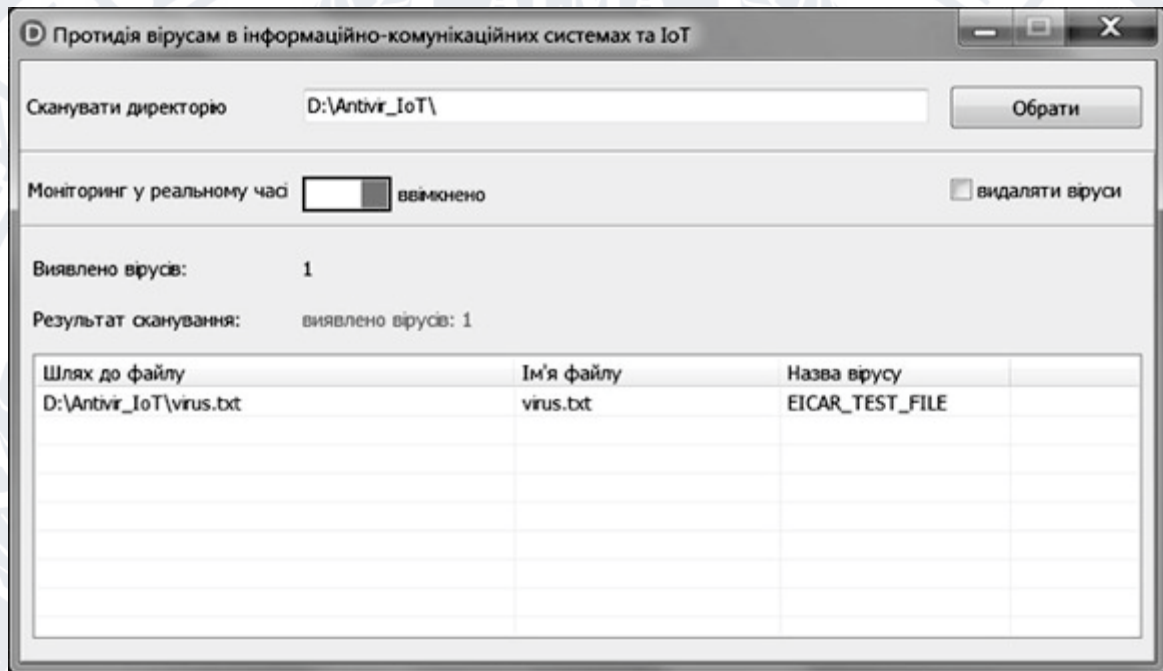


Рис. 4.6. Результат роботи системи на виявлення вірусів

На панелі можна побачити результат виявлення вірусу. У сітці вказано:

- шлях до файлу;
- ім'я файлу;
- назва вірусу.

4.3.2. Робота системи на знешкодження вірусів

Перевіримо роботу системи на знешкодження вірусів. Для цього необхідно встановити прапорець “Видаляти віруси”. Через проміжок часу

побачимо реакцію системи на знешкодження (рис. 4.7), про що свідчить надпис у Результаті сканування – чисто.

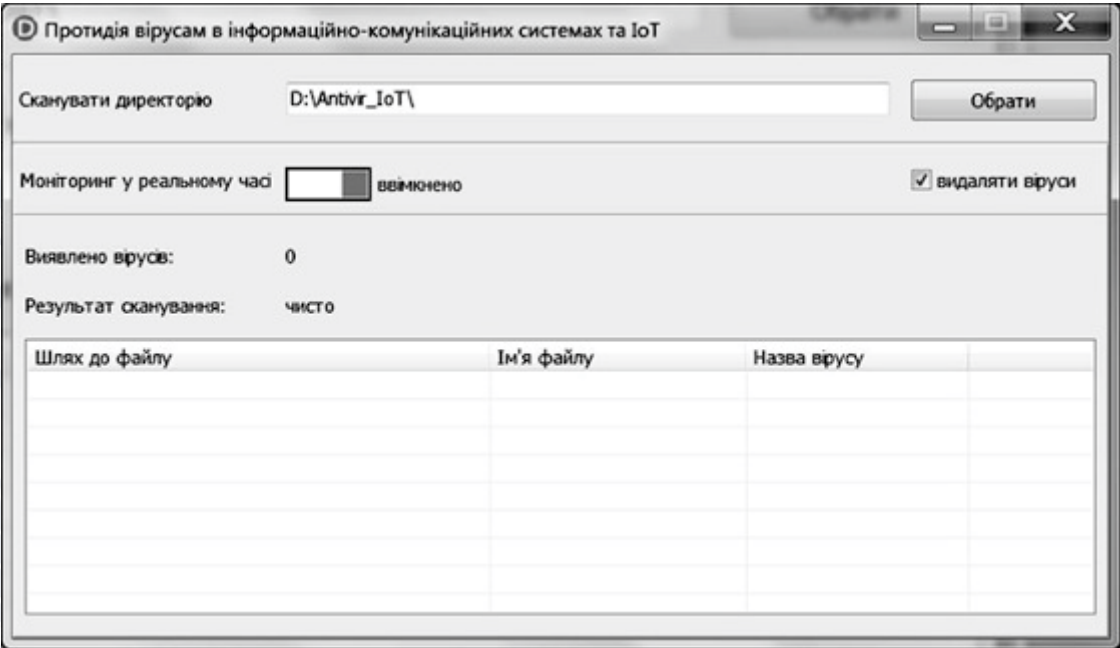


Рис. 4.7. Результат роботи системи на знешкодження вірусів

ВИСНОВКИ

За результатами роботи можна сформулювати наступні висновки:

1. Розглянуто методи та середовища розробки захисного ПЗ.
2. Обґрунтовано вибір системи розробки ПЗ Embarcadero.
3. Розроблено та протестовано систему протидії вірусам. Для тестування використано EICAR (або EICAR-Test-File – від European Institute for Computer Antivirus Research).
4. За результатом тесту, програма самостійно знешкодила загрозу(«virus.txt»),про що свідчить повторна діагностика системи без виявлення зараженого файлу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Review of THz Technologies for Rapid Sensing and Detection of Viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://pubmed.ncbi.nlm.nih.gov/34677305/>.
2. A tradeoff between the losses caused by computer viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191101>.
3. Computer Viruses Are "Rampant" on Medical Devices in Hospitals [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.technologyreview.com/2012/10/17/183245/computer-viruses-are-rampant-on-medical-devices-in-hospitals/>.
4. Detection of Plant Viruses and Disease Management [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.frontiersin.org/articles/10.3389/fpls.2020.01092/full>.
5. Email networks and the spread of computer viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://link.aps.org/doi/10.1103/PhysRevE.66.035101>.
6. The Best Antivirus Protection for 2022 | PCMag. (1970). Retrieved on May 21, 2022. - Access mode: <https://uk.pcmag.com/antivirus/8141/the-best-antivirus-protection>.
7. Типи та класифікація вірусів [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://sites.google.com/site/romancukolga/virusi/tipi-virusiv>.
8. The Top 10 Worst Computer Viruses in History | HP® Tech Takes [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.hp.com/us-en/shop/tech-takes/top-ten-worst-computer-viruses-in-history>.

9. Viruses | An Open Access Journal from MDPI[Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.mdpi.com/journal/viruses>.
- 10.9 Common Types of Computer Viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://hightouchtechnologies.com/9-common-types-of-computer-viruses/>.
11. An introduction to Computer Viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.osti.gov/servlets/purl/5608409>.
12. Computer Viruses vs Network Worms [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.kaspersky.com/resource-center/threats/computer-viruses-vs-worms>.
13. Computer virus [Electronic resource] - Retrieved on May 21, 2022. - Access mode: https://en.wikipedia.org/wiki/Computer_virus.
14. Malware Types and Classifications | Lastline [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.lastline.com/blog/malware-types-and-classifications/>.
15. SDT: A Virus Classification Tool Based on Pairwise Sequence [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0108277>.
16. The 10 Main Types of Computer Virus and How to Avoid Them [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.thebuddycompany.com/post/types-of-computer-virus>.
17. The Difference between a Computer Virus and Computer Worm [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.varonis.com/blog/what-is-a-computer-virus-and-computer-worm>.
18. The Three Major Types of Computer Viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://austinmobilecomputerrepair.com/the-three-major-types-of-computer-viruses/>.

19. The classification tree | Kaspersky IT Encyclopedia [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://encyclopedia.kaspersky.com/knowledge/the-classification-tree/>.
20. What Is a Computer Virus? [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.proofpoint.com/us/threat-reference/computer-virus>.
21. What are Computer Viruses? | Definition & Types of Computer Viruses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.fortinet.com/resources/cyberglossary/computer-virus>.
22. Roger A. Grimes. 9 types of malware and how to recognize them | CSO Online [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.csoonline.com/article/2615925/security-your-quick-guide-to-malware-types.html>.
23. 5 Tips for Staying Safe on Public Wi [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.globalsign.com/en/blog/staying-safe-using-public-wifi>.
24. Behavioural Computer Science: an agenda for combining modelling. [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0130-0>.
25. Convergence model of AI and IoT for virus disease control system [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8183316/>.
26. Cyber Security and the Internet of Things: Vulnerabilities, Threats [Electronic resource] - Retrieved on May 21, 2022. - Access mode: https://www.riverpublishers.com/journal_read_html_article.php?j=JCSM/4/1/4.
27. Digitalization and Energy – Analysis [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.iea.org/reports/digitalisation-and-energy>.
28. Industrial Control System [Electronic resource] - Retrieved on May 21, 2022. - Access mode:

<https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system>.

29. Information technology solutions, challenges, and suggestions [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7724285/>.
30. IoT Security Issues, Threats, and Defenses [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/iot-security-101-threats-issues-and-defenses>.
31. Top 10 IoT Vulnerabilities in Your Devices [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.keyfactor.com/blog/top-10-iot-vulnerabilities-in-your-devices/>.
32. What Is the Difference: Viruses, Worms, Trojans, and Bots? [Electronic resource] - Retrieved on May 21, 2022. - Access mode: https://tools.cisco.com/security/center/resources/virus_differences.
33. What is IoT (Internet of Things) and How Does it Work? [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>.
34. Why IoT security is important for your home network [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.kaspersky.co.uk/resource-center/threats/secure-iot-devices-on-your-home-network>.
35. Blog | Learn Delphi [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://learndelphi.org/blog/>.
36. Delphi 10 Seattle [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <http://www.embarcadero.com.pl/produkty/delphi/d10seattle/>.
37. Delphi: Internet of Things (IoT) Solutions [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://ftp.embarcadero.com/products/delphi/features/internet-of-things-iot>.

- 38.Embarcadero Launches Windows 10, IoT Dev Tools [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.eweek.com/development/embarcadero-launches-windows-10-iot-dev-tools/>.
- 39.Embarcadero Technologies | Delphi and C++ developer tools | Grey [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://greymatter.com/vendors/embarcadero/>.
- 40.IoT Mobile Application Development Solutions [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.embarcadero.com/solutions/internet-of-things>.
- 41.RAD in Action: Building Connected Apps with Bluetooth and App Dev [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.slideshare.net/embarcaderotechnet/rad-in-action-building-connected-apps-with-bluetooth-and-app-tethering>.
- 42.c++builder [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://stackoverflow.com/questions/7230518/is-embarcadero-c-builder-a-good-choice-as-an-ide>.
- 43.Joab Jackson. Embarcadero moves RAD Studio beyond Windows | Computerworld [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://www.computerworld.com/article/2907272/embarcadero-moves-rad-studio-beyond-windows.html>.
- 44.Security of Delphi remoting frameworks [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <http://codeverge.com/embarcadero.delphi.non-tech/security-of-delphi-remoting-frame/2011820>.
- 45.The Issue with Delphi Runtime Packages and Windows 10 Creators [Electronic resource] - Retrieved on May 21, 2022. - Access mode: <https://blog.marcocantu.com/blog/2017-june-delphi-packages-creators-update.html>.

ДОДАТОК А

Програмний лістинг

Головний модуль

```
unit MainUnit;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.DateUtils, System.Variants,  
System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,  
Vcl.ComCtrls, ScanUnit, Vcl.WinXCtrls, Vcl.ExtCtrls, Vcl.Menus, IOUtils;
```

```
const
```

```
VirusSign = '44D88612FEA8A8F36DE82E1278ABB02F:68';
```

```
VirusName = 'EICAR_TEST_FILE';
```

```
type
```

```
TForm1 = class(TForm)
```

```
ScanFileLabel: TLabel;
```

```
ScanPathEdt: TEdit;
```

```
BrowseBtn: TButton;
```

```
ScanBtn: TButton;
```

```
ScanResult: TLabel;
```

```
ResultsLabel: TLabel;
```

```
NumberOfVirusesDetectedLabel: TLabel;
```

```
NumberOfVirusesDetected: TLabel;
```

```
ResultsLv: TListView;
```

```
Timer1: TTimer;
```

```
ToggleSwitch1: TToggleSwitch;
```

```
OpenDialog1: TOpenDialog;
```

```
DelCheckBox: TCheckBox;
```

```
FilesMemo: TMemo;
```

```
Panel1: TPanel;
```

```

Panel2: TPanel;
Panel3: TPanel;
Label1: TLabel;
Panel4: TPanel;

procedure BrowseBtnClick(Sender: TObject);
procedure ScanBtnClick(Sender: TObject);
procedure ToggleSwitch1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.dfm}

//процедура вибору файлів директорії
procedure ListFileDir(Path: string; FileList: TMemo);
var
    Files: TArray<string>;
    i: integer;
begin
    FileList.Lines.Clear;
    Files:= TDirectory.GetFiles(Path);
    for i := 0 to Length(Files)-1 do FileList.Lines.Add(TPath.GetFileName(Files[i]));
end;

//процедура вибору директорії
procedure TForm1.BrowseBtnClick(Sender: TObject);

```

```

var
  path: string;
begin
  if OpenFileDialog1.Execute then
    begin
      path:= ExtractFileDir(OpenFileDialog1.FileName) + '\';
      ScanPathEdt.Text:= path;
      ListFileDir(path, FilesMemo);
      ScanBtn.Click;
    end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
  path: string;
begin
  //встановлюємо за замовчанням директорію, де розміщена програма
  path:= ExtractFileDir(application.ExeName) + '\';
  ScanPathEdt.Text:= path;
  ListFileDir(path, FilesMemo);
end;

//процедура сканування на віруси
procedure TForm1.ScanBtnClick(Sender: TObject);
var
  i, VirusesDetected: integer;
  isVirus: boolean;
begin
  VirusesDetected:= 0;

  scanresult.Caption:= '-';
  ScanResult.Font.Color:= $FFFFFF;
  NumberOfVirusesDetected.Caption:= VirusesDetected.ToString;
  resultslv.Items.Clear;
  resultslv.Clear;

```



```

ScanResult.Caption:= 'чисто';
ListFileDir(ScanPathEdt.Text, FilesMemo);

//перебір та сканування файлів директорії
for i:= 0 to FilesMemo.Lines.Count -1 do
begin
  if not (FilesMemo.Lines[i] = "") then
  begin
    isVirus:= ScanFile(ScanPathEdt.Text + FilesMemo.Lines[i]);
    //якщо знайдений вірус
    if isVirus then
    begin
      inc(VirusesDetected);

      with resultsLV.items.Add do
      begin
        Caption:= ScanPathEdt.Text + FilesMemo.Lines[i];
        Subitems.Add(FilesMemo.Lines[i]);
        Subitems.Add(VirusName);
      end;
      if DelCheckBox.Checked = true then DeleteFile(ScanPathEdt.Text + FilesMemo.Lines[i]);
    end;
  end;
end;
//якщо виявлено вірус(u)
if VirusesDetected > 0 then
begin
  ScanResult.Font.Color:= $0000FF;
  NumberOfVirusesDetected.Caption:= VirusesDetected.ToString;
  ScanResult.Caption:= 'виявлено вірусів: '+VirusesDetected.ToString;
end;
end;

//процедура сканування у реальному часі
procedure TForm1.Timer1Timer(Sender: TObject);

```

```
begin
```

```
    ScanBtn.Click;
```

```
end;
```

```
//ввіменення сканування у реальному часі
```

```
procedure TForm1.ToggleSwitch1Click(Sender: TObject);
```

```
begin
```

```
    Timer1.Enabled:= ToggleSwitch1.Enabled;
```

```
end;
```

```
end.
```

Модуль сканування

```
unit ScanUnit;
```

```
interface
```

```
uses windows, sysutils, classes, IdhashmessageDigest, idhash;
```

```
const
```

```
    VirusSign = '44D88612FEA8A8F36DE82E1278ABB02F:68';
```

```
    VirusName = 'EICAR_TEST_FILE';
```

```
function GetFileSize(FilePath: string): string;
```

```
function Md5File(const FileName: string): string;
```

```
function ScanFile(FilePath: string): boolean;
```

```
implementation
```

```
//функція сканування на віруси
```

```
function ScanFile(FilePath: string): boolean;
```

```
var
```

```

    Size,Hash,FileSign: string;
begin
    Size:= getFileSize(FilePath);
    Hash:= Md5File(FilePath);
    //функція підпису файлу
    FileSign:= Hash + ':' + Size;
    Result:= false;

    if FileSign = " then exit;
    //перевірка на наявність вірусу
    if FileSign = VirusSign then Result:= true;
end;

//функція отримання Md5 кодування файлу
function Md5File(const FileName: string): string;
var
    IdMd5: TidHashMessageDigest5;
    Fs: TFileStream;
begin
    iDMd5:= TIdhashmessageDigest5.Create;
    Fs:= TFileStream.Create(FileName, fmOpenRead or fmShareDenyWrite);
    try
        result:= IdMD5.HashStreamAsHex(Fs);
    finally
        fs.Free;
        IdMD5.Free;
    end;
end;

//функція отримання розміру файлу
function GetFileSize(FilePath: string): string;
var
    fs: TFileStream;
begin

```



```
try  
  fs:= TFileStream.Create(FilePath,Of_Read);  
  Result:= InttoStr(fs.Size);  
  fs.Free;  
except  
  fs.Free;  
  Result:= '0';  
end;  
end;
```

