МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Черненко Євген Віталійович

Програмне забезпечення системи кібербезпеки віртуальної інфраструктури

Спеціальність: 125 «Кібербезпека»

Освітній ступінь: бакалавр

Науковий керівник:

Босько Віктор Васильович _____

(підпис) (дата)

кандидат технічних наук, доцент

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

_____ О.А. Смірнов

(підпис) ПІБ

«____»_____2021 р.

Кропивницький – 2021

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
Спеціальність *125 Кібербезпека*

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____О.А.Смірнов
« 11 » січня 2021 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Черненку Євгену Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки віртуальної інфраструктури*

керівник роботи *Босько Віктор Васильович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 129-02 від 14.12.2020 року

2. Строк подання студентом роботи до захисту *22.05.2021 р.*

3. Мета та завдання кваліфікаційної бакалаврської роботи*: Метою розробки є програмне забезпечення системи кібербезпеки віртуальної інфраструктури*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

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

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

6. Дата видачі завдання « 11 » січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	14.05.2021 р.	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Черненко Є.В. Програмне забезпечення системи кібербезпеки віртуальної інфраструктури. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи кібербезпеки віртуальної інфраструктури.

Метою розробки є програмне забезпечення системи кібербезпеки віртуальної інфраструктури.

Результат роботи – програмна реалізація системи кібербезпеки віртуальної інфраструктури.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: кібербезпека, віртуальна інфраструктура

ABSTRACT

Chernenko Ye.V. Virtual infrastructure cybersecurity system software. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification the software which is intended for system of cybersecurity of virtual infrastructure is developed.

The purpose of the development is the software of the cybersecurity system of the virtual infrastructure.

The result is the software implementation of the cybersecurity system of the virtual infrastructure.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of Delphi 10.4 Sydney.

Keywords: cybersecurity, virtual infrastructure

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВМ	–	віртуальні машини
КМ	–	комп’ютерна мережа
КСАЗ	–	комплексна система антивірусного захисту
МЕ	–	міжмережний екран
ПЗ	–	програмне забезпечення
ПК	–	персональний комп’ютер
ACL	–	Access Control List
DRS	–	постійне балансування навантаження між хостами
FTP	–	File Transfer Protocol
http	–	HyperText Transfer Protocol
IaaS	–	Infrastructure as a Service, інфраструктура як сервіс
MITM		людина посередині, man-in-the-middle
POP3		Post Office Protocol Version 3
SMTP		Simple Mail Transfer Protocol
VDI		віртуалізація робочих місць співробітників
VLAN		Virtual Local Area Network

ВСТУП

Актуальність теми. Ні для кого не секрет, наскільки швидко росте популярність віртуалізації. Усе більше й більше компаній відмовляються від своїх серверів і воліють віддати ці турботи хостинг провайдерам. Безумовно, це правильне рішення для більшості компаній. У першу чергу це заощаджує не маленькі суми з бюджету, які витрачаються на обслуговування цих серверів, такі як оплата електроенергії, заміна комплектуючих, зарплата персоналу по обслуговуванню і т.д.

Так само не варто забувати, що віртуальна інфраструктура» дозволяє заощаджувати на потужностях. Для тих самих завдань, віртуалізація вимагає менше ресурсів, у порівнянні із класичними «залізними» серверами. Досягається це постійним автоматичним перерозподілом ресурсів і виділенням однієї й тієї ж частини пам'яті під однотипні, але роздільні елементи структури.

На даний момент, крім віртуалізації серверів, починає рости інтерес також до віртуалізації робочих місць співробітників (VDI).

Це є більш серйозним кроком у майбутнє пари IT/бізнес і дозволяє не тільки заощаджувати чималі засоби, але й значно підвищити продуктивність і результативність співробітників, підвищуючи до максимуму їх мобільність. VDI дає можливість працювати зі своїм особистим робочим простором з будь-якої точки миру й з будь-якого пристрій (комп'ютер, ноутбук, планшет, телефон). Для підключення необхідний тільки інтернет, причому не обов'язково «широкий», досить 1 Мбіт/с.

Після усвідомлення всіх цих переваг керівники компаній починають замислюватися, а що можна поставити на противагу?

Практично в усіх, першим і самим хвилюючим є питання безпеки. Ми віддаємо всю нашу інфраструктуру й усі наші дані кудись, де з ними можуть зробити що завгодно без нашого відому.

Багатьох це питання зупиняє, і вони відкладають його в далекий ящик. Але це відбувається не через складність питання, а через його нестандартність. Для стаціонарних застосунків уже багато років як розроблені й затверджені стандарти по захисту даних. Для віртуалізації поки далеко не всі аспекти описані, але це не виходить, що ми беззахисні. Методи захисту є. Здебільшого вони засновані на методах захисту стаціонарних систем.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки віртуальної інфраструктури.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем кібербезпеки віртуальної інфраструктури.

– Дослідження системи кібербезпеки віртуальної інфраструктури.

– Програмна реалізація системи кібербезпеки віртуальної інфраструктури.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віртуальної інфраструктури.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки віртуальної інфраструктури, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації хмарного антивірусу. Комп'ютери вже давно ввійшли в повсякденну реальність. Давно минули часи, коли самодостатній комп'ютер сам по собі був приводом для замилування. Але радість від можливості користуватися настільки чудовою річчю супроводжується турботами. Найважливіша із цих турбот полягає в тому, щоб захистити комп'ютер від вірусів. Про антивіруси сьогодні знає практично кожна людина. Серед користувачів особливу популярність придбали суперечки про те, який з застосунків подібного роду краще справляється зі своїми завданнями й меншою мірою «напружує» комп'ютер. Із часом усе більше сервісів переміщається з локального жорсткого диска на віддалені сервера. Тому сьогодні ми поговоримо про «антивірусні хмари». Ці сервіси оберігають комп'ютер, перебуваючи часом за тисячі кілометрів від нього.

Принципи роботи хмарних антивірусів

Завдяки старанням Стефани Кроуфорд (Stephanie Crawford) на сторінках ресурсу Howstuffworks з'явився досить докладний огляд хмарних антивірусних застосунків. Цей огляд повною мірою заслуговує на увагу й детальнішого розгляду, оскільки в ньому відбиті всі основні особливості цієї порівняно нової категорії сервісів.

Для початку небагато термінології. Вірусом називається програма, яка самостійно встановлюється на комп'ютер і вносить у розміщені на ньому дані небажані зміни. Основним джерелом вірусів у сучасній реальності є Інтернет.

Щоб відгородити свої дані від шкідливого коду, люди застосовують антивірусне програмне забезпечення. Сьогодні можна скористатися не тільки встановлюваними на комп'ютер антивірусними застосункуми, але й хмарними

антивірусними засобами. Процеси такого програмного забезпечення здебільшого відбуваються на віддаленому інтернет-сервері, а не на жорсткому диску комп'ютера.

Хмарні антивіруси є комплексами із клієнтського застосунку й веб-сервісу. Обидві частини антивірусу працюють спільно. Клієнт – це невелика програмка, яка працює на комп'ютері користувача й сканує систему, перевіряючи, не чи заражена вона шкідливим кодом. Відомо, що традиційні антивірусні програми, установлювані на комп'ютер, є «пожирателями ресурсів», але клієнтські програми хмарних антивірусів вимагають набагато менше обчислювальної потужності.

Веб-сервіс хмарного антивірусу розташовується в Інтернеті, на одному або декількох серверах. Більшу частину завдань по обробці даних виконує саме він, тому комп'ютеру користувача не доводиться не обробляти, не зберігати значні обсяги інформації. Через певні проміжки часу програма-клієнт сканує комп'ютер. Суть сканування полягає в пошуку шкідливого коду, інформація про який є в базі даних веб-сервісу.

Тепер перелічимо ті переваги, якими має хмарний антивірус у порівнянні із традиційним:

– Програма-Клієнт має доступ до самих свіжих даних про шкідливий код через усього кілька хвилин після того, як про нього «довідався» веб-сервіс. Немає необхідності постійно обновляти антивірусне програмне забезпечення.

– Програма-Клієнт дуже мала й задовольняє невеликою обчислювальною потужністю. Отже, вона не відволікає комп'ютер від інших виконуваних їм завдань.

– Хмарні антивіруси безкоштовні. Втім, відновлення, додаткові утиліти й підтримка пропонуються за гроші.

Тепер, коли ми довідалися про того, що представляє собою хмарне антивірусне програмне забезпечення, розглянемо ті функції, які виконує типовий хмарний антивірус.

Функції хмарного антивірусу

Інтерфейс користувача хмарного антивірусу не викличе серйозних питань ні в кого з тих, хто має досвід використання традиційних антивірусних програм. І роботу він виконує ту ж: сканує комп'ютер, виявляє шкідливий код і чистить від нього систему.

Перелічимо основні функції, доступні в користувацькому інтерфейсі хмарного антивірусу:

- Сканування всього комп'ютера або окремих папок.
- Можливість налаштування автоматичного режиму сканування із вказівкою тих файлів, які слід включити в область сканування.
- Перегляд докладного звіту про те, який шкідливий код був виявлений у процесі сканування.
- Дії по видаленню або відновленні файлів, поміщених у карантин або файлів, які були знешкоджені тем або іншим способом.

У цих основних функціях відмінностей від традиційного антивірусу не спостерігається. Але є ті можливості, які властиві винятково хмарним антивірусним сервісам. Як ми вже говорили, хмарний антивірус розподіляє виконання своїх завдань між комп'ютером користувача (програма-клієнт) і віддаленим веб-сервером (або декількома серверами), доступ до якого здійснюється через Інтернет.

Таким чином, частина ресурсів є «загальної» для всіх користувачів. Це не тільки обчислювальні потужності серверів, але й центральна база даних, що містить дані про шкідливий код.

Ця база даних складається різними способами. Для кожного продукту характерні свої методи її поповнення. Наприклад, Panda Cloud Antivirus одержує дані із сукупності джерел, яку самі розроблювачі називають «Колективним розумом»: з IT і програмних ресурсів, від хостів-пасток (комп'ютерів, спеціально залишених у якості принади для вірусів), а також від самих користувачів.

Immunet Free Antivirus одержує свої дані від співтовариства користувачів (Immunet Cloud) і саме в такий спосіб його розроблювачі довідаються про потенційні погрози. Сервери, на яких розміщені хмарні антивіруси, працюють із алгоритмами, здатними класифікувати віруси по різних ознаках, у тому числі й по ступеню небезпеки.

Хмарні бази даних відрізняються не тільки методиками збору інформації. Реальною перевагою хмарних антивірусів є та швидкість, з якої вони здатні забезпечити захист від нових погроз.

У хмарних антивірусах передбачена також можливість кешування бази даних на комп'ютері для подальшого використання в офлайн-режимі. Зрозуміло, у цьому випадку база буде містити дані за станом на момент її збереження. Цей кеш може обновлятися під час виходу комп'ютера в Інтернет. Але він не містить повний перелік інформації про шкідливий код, тільки про найпоширеніші погрози.

1.2 Область застосування

Областю застосування є віртуальна інфраструктура. Віртуальна інфраструктура – аналог традиційної (фізичної) IT-інфраструктури, яка обслуговує бізнес-процеси в сучасній компанії. Це набір серверів і встановленого на них спеціального ПЗ, які розташовані на устаткуванню хостинг-провайдера.

Що таке IaaS

В англійській версії термін звучить як IaaS – Infrastructure as a Service, тобто інфраструктура як сервіс. Клієнт одержує пул віртуальних ресурсів, куди входять:

- процесорні потужності;
- оперативна пам'ять;
- місце на диску;

– можливість підключення інших ресурсів (наприклад, графічних потужностей).

Кожному клієнтові доступний інтернет-канал для звертань до сервера й передачі даних, зовнішня IP-адреса (можливість підключення декількох IP-адрес), віртуальні машини (на базі Windows Server, Centos, Ubuntu Server і т.д.) і ліцензійне ПЗ. Використовуючи різні комбінації ресурсів, установлюючи будь-які системні й прикладні застосунки, можна створити робітниче середовище, яке відповідає потребам конкретного підприємства.

Віртуальна інфраструктура VMWare базується на таких принципах:

– поділ: кілька операційних систем працюють на одній фізичній машині, а ресурси розподілені між декількома віртуальними машинами;

– ізоляція: несправності, які виникають на одній віртуальній машині, не впливають на роботу інших віртуальних машин, навіть якщо вони перебувають на одному фізичному сервері;

– мобільність: віртуальну машину, яка зберігається у вигляді набору файлів, можна переміщати й копіювати.

Можливості віртуалізації для корпоративних користувачів

– Універсальна конфігурація. Обсяг оперативної пам'яті або вільне місце на накопичувачах – усі параметри підбираються індивідуально під потреби інфраструктури клієнта.

– Зниження витрат на зміст IT-інфраструктури. Обчислювальні й інші ресурси доступні як послуга й не вимагають покупки фізичного устаткування, його налаштування, адміністрування, усунення неполадок. Немає необхідності наймати в штат технічних фахівців, у майбутньому будуть не потрібні інвестиції в модернізацію. ресурси, що звільнилися, можна використовувати для розвитку пріоритетних напрямків бізнесу.

– Підтримка масштабування. Якщо змінилися вимоги застосунків і сервісів, у будь-який момент можна змінити обсяг послуги – вибрати більш-менш

потужні процесори, інші обсяги накопичувачів і ОЗП. Вартість оренди буде перелічена.

– Гнучкі мережні налаштування. Хмарна віртуальна інфраструктура підтримує керування через інтуїтивно зрозумілий веб-інтерфейс. Можливе налаштування firewall, VPN, NAT і інших сервісів і служб.

– Керованість і контроль. Клієнт може створювати віртуальні машини, перерозподіляти доступні ресурси оптимальним образом, здійснювати моніторинг бізнес-процесів.

– Безперервність бізнесу. Технології віртуалізації захищають критичні застосунки. Для проведення технічного обслуговування або аварійного відновлення не потрібна зупинка застосунків.

– Підвищення відказостійкості інфраструктури. Збільшується швидкість реагування на технічні неполадки. У випадку виходу з ладу застосунку Клієнта будуть автоматично перенесені на робочі обчислювальні потужності.

– Тестування застосунків перед впровадженням. Хмарне середовище стане оптимальною платформою для перевірки роботи застосунків перед розгортанням.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки віртуальної інфраструктури, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи

Хмарні антивірусні сканери є корисними інструментами для додаткового аналізу підозрілих файлів, причому для проведення перевірки користувачі не потрібно встановлювати програму.

Були протестовані онлайн сканери

Дослідники перевіряли, пропонує чи вендор онлайн інструмент сканування. На жаль не всі компанії пропонують подібні можливості, і не всі інструменти відрізняються високою якістю.

У результаті в тестуванні взяли участь наступні продукти й сервіси: Bitdefender Quickscan, Comodo Cloud Antivirus, ESET Online Scanner, F-Secure Online Scanner, Mcafee Security Scan Plus, Norton Security Scan, Panda Cloud Cleaner і Trend Micro Housecall.

Дуже швидко стало ясно, що не всі продукти є класичними онлайн сканерами, які працюють у браузері. Багато антивірусних вендорів зараз пропонують автономні утиліти, які потрібно завантажувати й установлювати на комп'ютері. Вони можуть працювати разом з основним антивірусним захистом без конфліктів і проблем із продуктивністю. Вони можуть використовуватися тільки при необхідності додатково перевірити статус безпеки системи.

Важливе зауваження: даний тест не включає повноцінні антивірусні продукти. У випробуванні беруть участь безкоштовні інструменти, які можуть використовуватися для сканування й оцінки безпеки системи. Дані програми не призначені для заміни основних застосунків безпеки, зокрема антивірусів, файєрволів, інструментів захисту від шпигунського ПЗ. Вони повинні бути використані для доповнення антивірусного захисту й допомоги при ідентифікації

потенційних проблем у системі, які можуть бути викликані дією шкідливих програм.

Як проходила процедура тестування?

По-перше, з'ясовувалася концепція й архітектура продукту – чи є він класичним онлайн сканером, автономним застосунком або гібридною версією двох підходів. Перед тем, як установити або запустити тестуємий продукт, тестова система заражалася шкідливими об'єктами. Потім відвідувалися небезпечні веб-сайти з небажаними куками, пропозиціями установки тулбарів і шкідливих програм.

Після того, як дослідники переконувалися, що система інфікована, а на диску були збережені небезпечні файли, сканери послідовно запускалися. Фіксувався час сканування системи, кількість виявлених зразків, надавана інформація про погрози й пропоновані застосунки. Сканування виконувалися на розділі розміром в 20 гігабайт із 17 гігабайтами зайнятого простору. Тестова машина управлялася Windows 10 з відключеним Захисником Windows, у системі було примусово запущено 17 шкідливих програм.

Bitdefender Quickscan

Bitdefender Quickscan – один зі справжніх онлайн сканерів. Проте, запустити сканер можна не у всіх браузерах. Bitdefender Quickscan підтримує роботу в Internet Explorer, Mozilla Firefox і Google Chrome. Відмітною рисою сканера є висока швидкість роботи – перевірка тестової системи зайняла менш однієї хвилини.

На жаль, Bitdefender Quickscan просто повідомляє, заражена система чи ні й не повідомляє назву активної погрози. Ніяку додаткову інформацію користувач не одержує, йому пропонується встановити 3-місячну пробну версію Bitdefender Internet Security для дезінфекції системи.

Даний інструмент працює швидко й безпроблемно, але являє сумнівну цінність для користувачів. Інші вендори пропонують більш цікаві альтернативи.

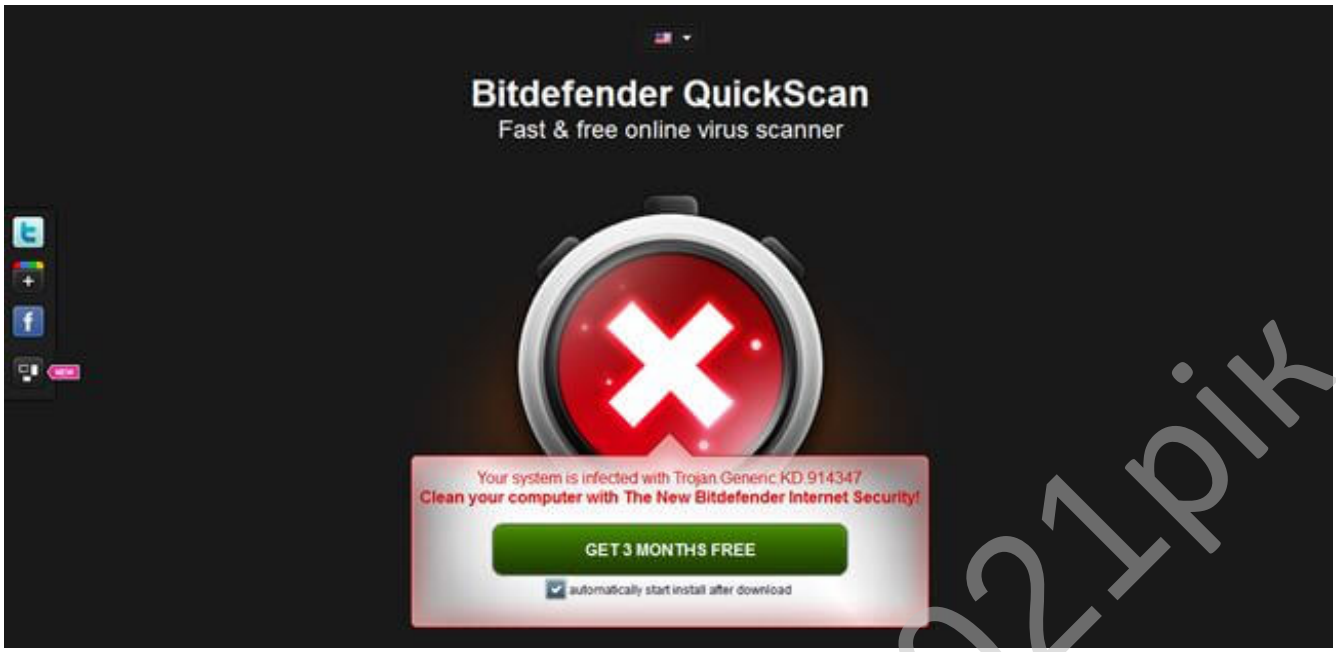

Рисунок 2.1 – Bitdefender Quickscan

Comodo Cloud Antivirus

Comodo Cloud Antivirus – хмарний сканер, який представлений у вигляді автономної програми, доступної для завантаження й установки на комп'ютери Windows. Ви відразу заметете, що сканер виглядає скоріше, як повноцінної антивірусний застосунок. Користувацький інтерфейс пропонує велика кількість опцій і налаштувань, а також кілька видів аналізу системи: швидке, повне або вибіркове сканування.

Швидке сканування тестової системи за допомогою Comodo Cloud Antivirus завершилося за 2 хвилини, у результаті був виявлений усього один шкідливий зразок. Це слабкий результат, тому додатково було проведено повне сканування системи. Цього разу перевірка виконувалася 59 хвилин, і Comodo зміг виявити 13 зловредів з 17.

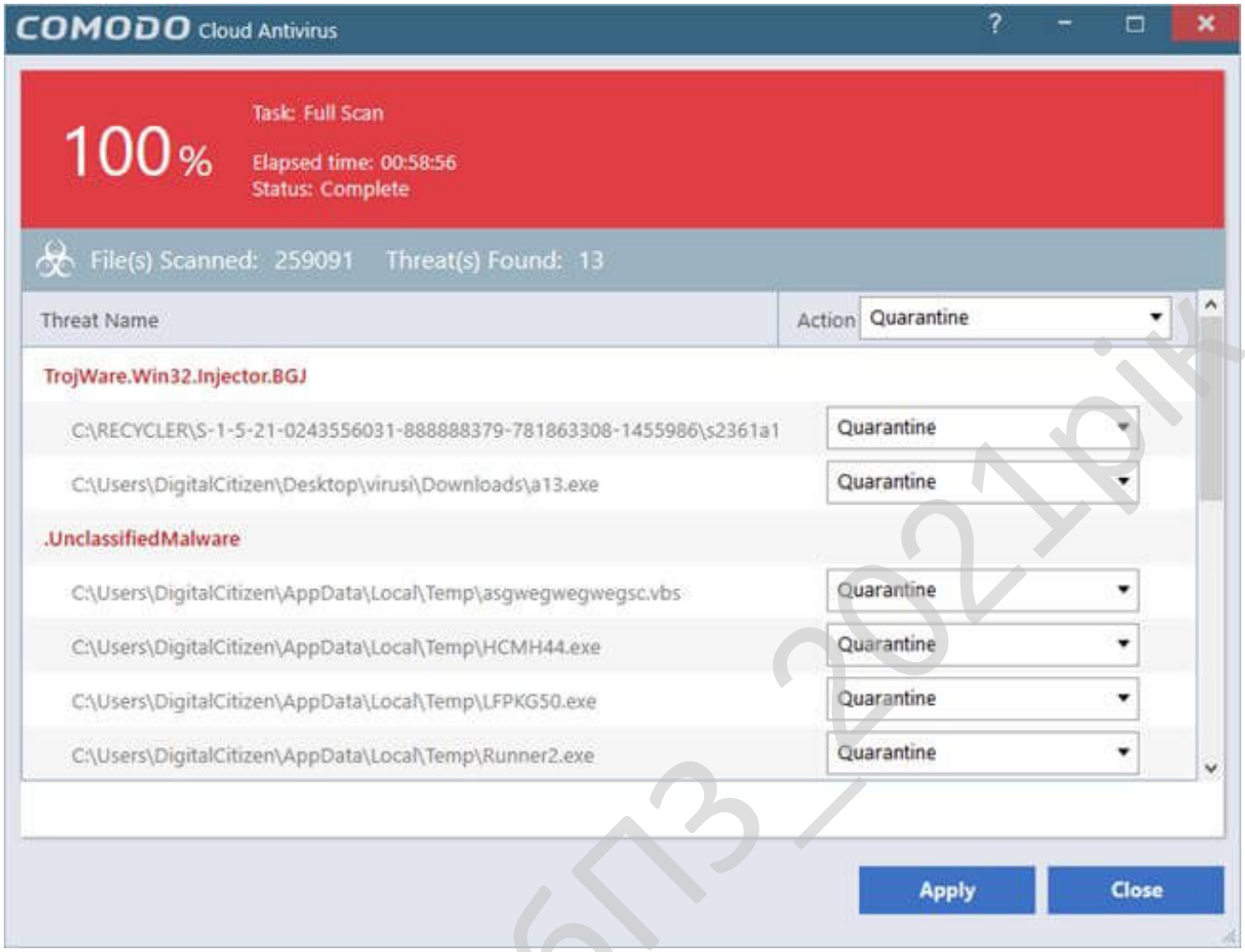

Рисунок 2.2 – Comodo Cloud Antivirus

Comodo Cloud Antivirus – один з деяких антивірусних сканерів, який здатний вилучити виявлені погрози. Цю особливість цінують користувачі.

ESET Online Scanner

ESET Online Scanner – один з найкращих онлайн сканерів. Продукт доступний як окремий застосунку для Windows, які можна завантажити й запустити на комп'ютері. Перед виконанням сканування ESET автоматично завантажує останні сигнатури на комп'ютер і дозволяє набудовувати параметри перевірки. За замовчуванням ESET Online Scanner настроєний на повне сканування комп'ютера: оперативна пам'ять, об'єкти автозавантаження й локальні

диски. Користувач може задати автоматичне очищення виявлених погроз, сканування архівів і перевірку на предмет потенційно небажаних застосунків.

Повне сканування системи зайняло більш 29 хвилин, і ESET зміг розпізнати всі 17 тестових зразків. Для кожного окремого зловреда ESET вивів назву й шлях розташування. Примітно, що після закінчення перевірки, користувач може налаштувати видалення файлів сканера із системи.

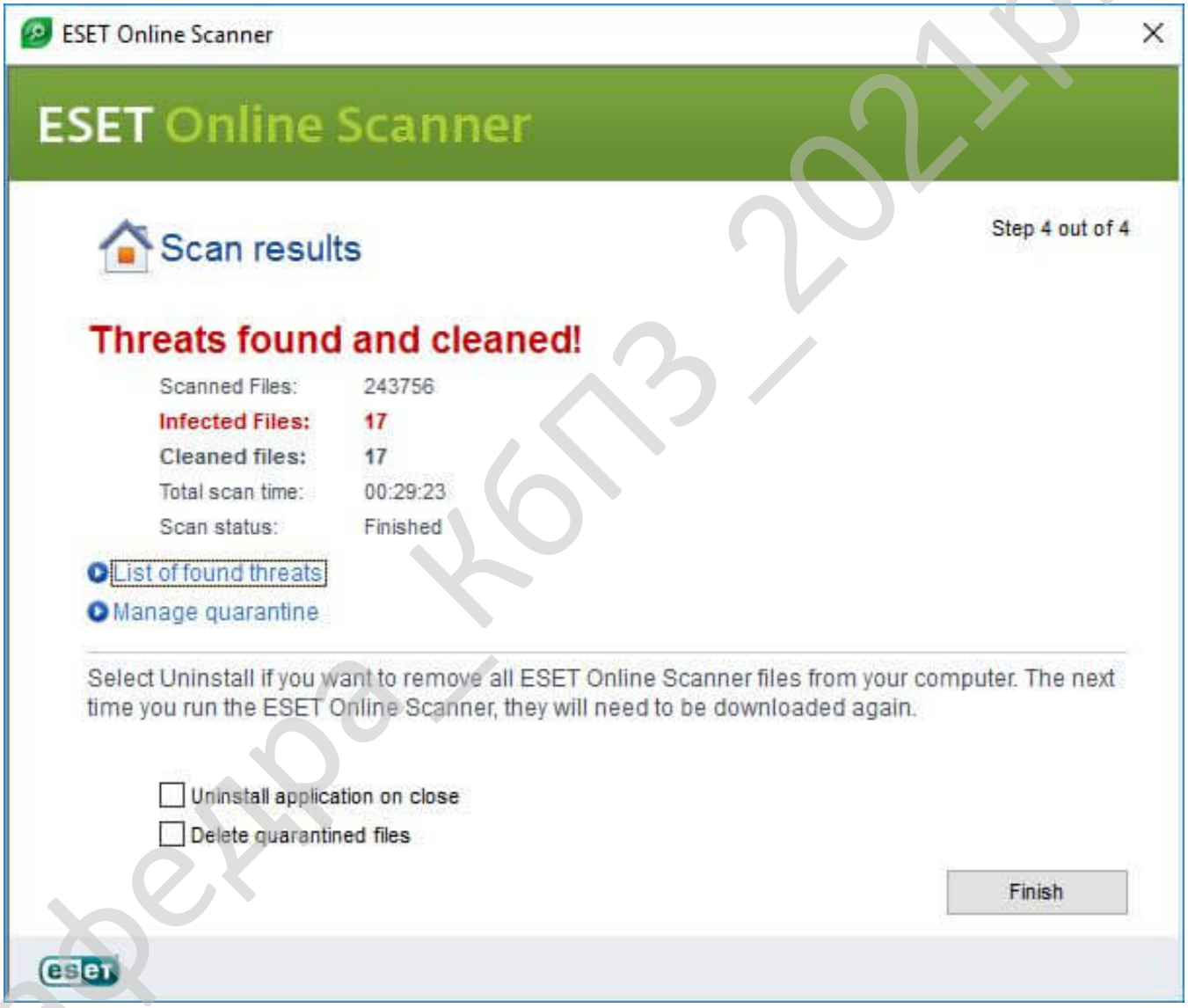

Рисунок 2.3 – ESET Online Scanner

Продукти ESET просуваються стримано й ненав'язливо. ESET Online Scanner прекрасно справляється із завданнями додаткового сканера й продемонстрував високу ефективність у випробуванні.

F-Secure Online Scanner

F-Secure Online Scanner – інша автономна програма, яку потрібно попередньо завантажувати й запускати. Перед виконанням перевірки, продукт завантажує новітні сигнатурні визначення.

Потім F-Secure Online Scanner приступає до сканування системи в пошуках шкідливих програм. Користувач не може нічого налаштувати й не може вибрати тип перевірки – увесь процес виконується автоматично.

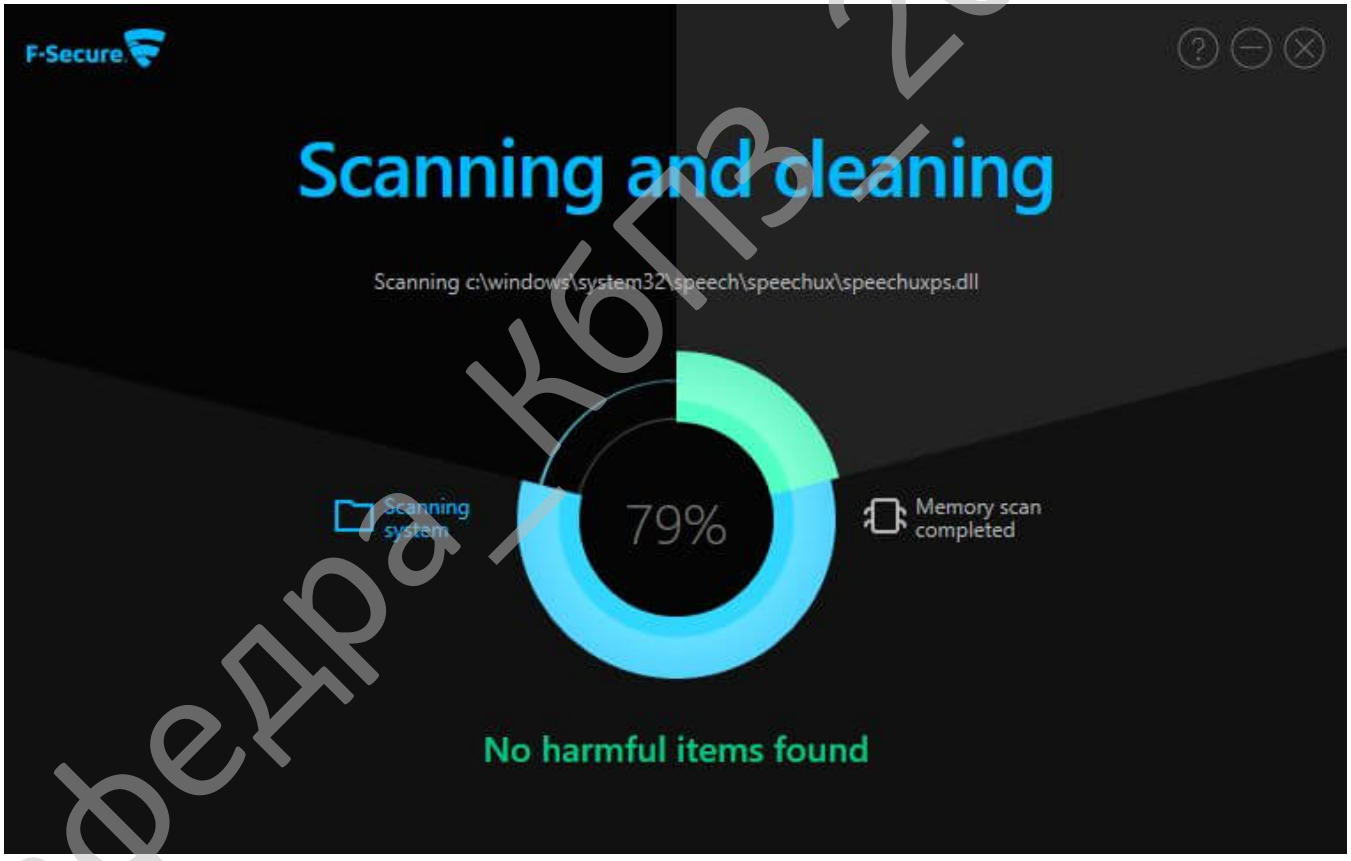

Рисунок 2.4 – F-Secure Online Scanner

Якщо F-Secure Online Scanner виявляє інфіковані файли, продукт просить виконувати перезавантаження системи для того, щоб провести очищення погроз.

При бажанні можна подивитися інформацію про розташування погроз, загальній кількості заражених файлів – якщо цього недостатньо, можна перейти на сайт для одержання додаткових даних про знайдені шкідливі програми.

F-Secure Online Scanner – один з найшвидших сканерів, перевірка тестової системи з 20-гігабайтним розділом виконувалася менш хвилини. F-Secure у підсумку виявив 12 вірусів з 17.

Mcafee Security Scan Plus

Mcafee Security Scan Plus має довге й складне ім'я, але в дійсно є дуже простим застосунком для Windows, яке перевіряє комп'ютер на предмет погроз і проблем безпеки. Mcafee Security Scan Plus автоматично обновляє свої файли при запуску, а сканування тестової системи продукт виконав менш, чим за хвилину. На жаль, користувач не може задавати параметри перевірки. У тестовій системі Mcafee виявив тільки один шкідливий файл із 17.

Складно знайти причини для регулярної перевірки комп'ютера за допомогою Mcafee, але Mcafee Security Scan Plus передбачає таку можливість. Користувач може запланувати перевірки, як йому зручно.

Mcafee Security Scan Plus не пропонує функцію очищення, тому єдиний виявлений програмою зразок, залишився активним у системі. При Виборі опції захист інструмент перенаправляє Вас на сайт Mcafee для придбання підписки Mcafee Total Protection для 3 комп'ютерів. Вам не здається, що це занадто багато для одного зараженого пристрою?

Рисунок 2.5 – Mcafee Security Scan Plus

Norton Security Scan

Norton Security Scan – десктопний застосунок, який потрібно завантажувати і встановлювати на комп'ютері. При запуску програма автоматично завантажує новітні антивірусні визначення й відновлення движка сканування. Norton Security Scan перевіряє системи не тільки на предмет

шкідливих програм, але й на предмет підозрілих куків й потенційно небезпечних застосунків.

Рисунок 2.6 – Norton Security Scan

При наведенні курсору миші на виявлену погрозу, Norton відображає докладну інформацію про неї. Після установки Norton Security Scan автоматично запускає сканування системи, на тестовій машині продукт виявив 12 зловредів за 2 хвилини. На жаль, користувач не може набудовувати перевірку, видалення виявлених погроз теж недоступно.

При натисканні кнопки “Fix now” Norton Security Scan відправить Вам на сайт і рекомендаціями із завантаження продуктів Norton.

Panda Cloud Cleaner

Panda Cloud Cleaner довів високу точність і ефективність. Продукт поставляється у вигляді окремого застосунку, який потрібно завантажити й установити в системі Windows. Перед стартом сканування Panda завантажує останні сигнатури в автоматичному режимі. Користувач може запустити швидку перевірку або вибіркову перевірку.

Коли сканування завершене, Panda Cloud Cleaner виводить звіт про виявлені шкідливі програми, невідомі файли і підозрілі політики і об'єкти, які можуть бути видалені.

Рисунок 2.7 – Panda Cloud Cleaner

При виборі окремого розділу можна подивитися детальну інформацію про виявлені проблеми.

Швидке сканування виконувалося 8 хвилин і виявило 13 заражених файлів, у той час як повний перевірка зайняла 32 хвилини й змогла виявити 17

шкідливих зразків. Високий рівень виявлення робить Panda Cloud Cleaner фаворитом тестування.

Trend Micro Housecall

Trend Micro Housecall – автономний застосунок, який потрібно скачати й запустити на комп'ютері. При запуску продукт автоматично завантажує новітні сигнатури. інтерфейс програми гранично простий. За допомогою кнопки налаштувань користувач може вибирати швидке, повне або вибіркове сканування.

Швидке сканування завершилося за 7 хвилин і виявило 3 інфікованих файлу. Повне сканування системи виконувалося 462 хвилини (і це не помилка!) і дозволило розпізнати 11 файлів. Тривалий час сканування випробовував терпіння дослідників.

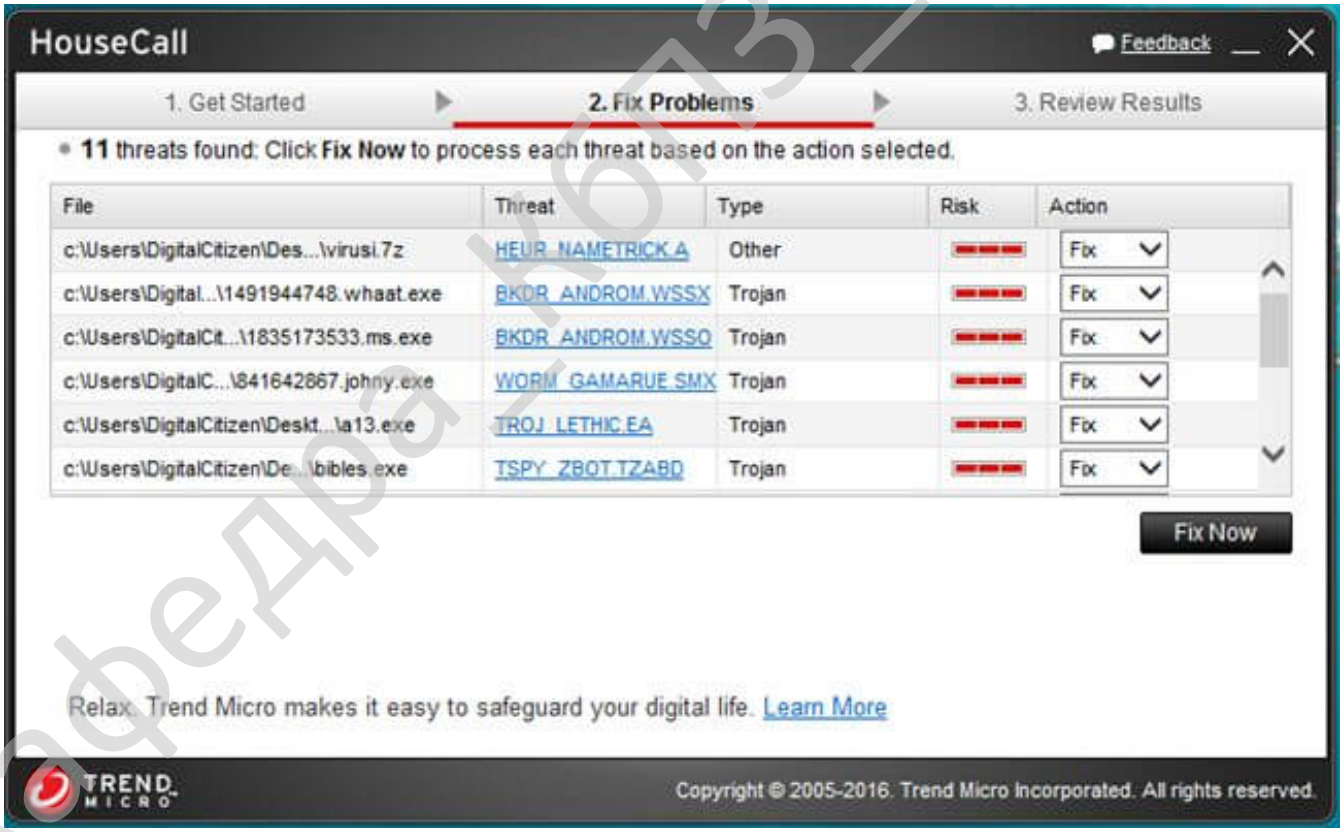

Рисунок 2.8 – Trend Micro Housecall

Для кожного шкідливого об'єкта відображається ім'я, точне розташування, тип і рівень небезпеки. З позитивної сторони, Trend Micro Housecall дозволяє видаляти шкідливі елементи, що цінують користувачі.

Trend Micro Housecall є інноваційним продуктом, який має можливості, які відсутні в більшості безкоштовних сканерів. Низька швидкість сканування й середній рівень виявлення не дозволяє Trend Micro наблизиться до лідерів.

Якщо Ви намагаєтеся вибрати гарний хмарний сканер, то Panda Cloud Cleaner імовірно буде кращим варіантом. Також рекомендуємо ESET Online Scanner і F-Secure Online Scanner, що розташувалися на 2 і 3 місці рейтингу відповідно. Усі перераховані програми є надійними інструментами безпеки, які змогли виявити більшість погроз. Усі три продукти дозволяють виконувати очищення системи й не примушують установлювати додаткові програми захисту. Коли Ви зіштовхнулися із зараженням, це буде більшим плюсом.

Майже всі інші продукти показали низьку ефективність виявлення зловредів і носять винятково інформаційний характер, не дозволяючи видаляти небезпечні об'єкти. Вони в основному являють собою маркетингові інструменти, призначені для продажу комерційних застосунків.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

− Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WEBView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

− Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

− Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

− Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

− Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

− Відладник Win 64 (на LLDB) і збирач для C++.

− Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

− Підтримка Metal Driver GPU для macOS і iOS.

− Вбудований Fmxlinux.

− Компонент TWEBbrowser для iOS тепер реалізований на WkWEBview API. Реалізація компонента Media Player для macOS тепер використовує

Avfoundation. Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

− Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

− Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

− Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

− У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуться, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними WEB-технологіями від Microsoft – новим WEBView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom

managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуться щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовучи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існучим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це

означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на ф ормі VCL автоматично масштабуться під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати декілька VCL стилів на різних формах в одному застосунку або в різних компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовуть стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент TWEBbrowser для iOS тепер реалізований на WkWEBview API.

– Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки віртуальної інфраструктури.

В процесі розробки кваліфікаційної бакалаврської роботи необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи кібербезпеки, що дозволить

реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Основні погрози для віртуальної інфраструктури

Розглянемо основні вектори погроз для віртуальної інфраструктури.

1. Фізичні погрози

Давайте розберемо основні відмінності стаціонарних систем від віртуальних з погляду відказостійкості.

Приміром, раніше в нас було 5 серверів для реалізації наших програмних потреб. Елементів відмови в нас було – кількість комплектуючих помножити на 5. При перехід до віртуалізації в нас один залізний сервер (хост Машина) і 5 віртуальних серверів. Елементів відмови в нас в 5 раз менше. Усі елементи серверів – програмні. У частині фізичного захисту перше, що нас повинне цікавити, це те, як живуть наші хост машини. Багато провайдерів з радістю проводять екскурсії по серверних залах. Тому рекомендується в обов'язковому порядку переконатися в якості умов хостингу.

2. Помилки при розробці ПЗ

Як відомо, поки, код програмного забезпечення пишуть люди. Незалежно від того, комерційний проект чи ні, недоліків у розробці не уникнути. Різниця тільки в кількості цих помилок.

У зв'язку із цим, із завидною частотою ці помилки виявляються або самими розроблювачами, або ентузіастами «баг-хантерами», або багатостраждальними користувачами, а іноді непорядними шукачами уразливостей, які у свою чергу використовують цю помилку (уразливість) у своїх корисливих цілях. Називають таку уразливість – 0day (0 днів пройшло з моменту виявлення й виправлення, уразливість розроблювачам не відома й не виправлена). На жаль, від таких погроз не захиститися. Зате 1day уразливість ми

можемо «прикрити», установивши свіже відновлення з відповідним виправленням. Тому так важливо мати завжди актуальну, пропатченну версію корпоративного ПЗ.

На жаль, на сьогоднішній день практично в кожній компанії можна знайти й 1day і 2day … Nday уразливості. Тому бажане вести централізований облік версій використовуваного ПЗ й мати можливість його централізовано обновляти.

3. Доступ до керування інфраструктурою

Зараз основний тренд в адмініструванні систем спрямований на «централізацію» керування. Керуванням усім і вся походить із однієї точки. Якщо одержати дані УЗ із доступом у центр керування, можна одержати повний контроль над усією інфраструктурою в цілому. Один з основних способів убезпечити себе від цього «апокаліпсиса» – розмежування прав адміністраторів.

Невеликий приклад.

Співробітник servicedesk повинен створювати нові користувацькі віртуальні машини (ВМ). Виходить, йому видаються права тільки на створення машин на основі еталонного образа, заздалегідь підготовленого адміністраторами. Без доступу до даних на самих ВМ, без можливості зміни налаштувань ВМ і хост машин. Тобто заборонене всі, крім створення машин.

Адміністраторам потрібно дати права на зміну налаштувань ВМ і хост машин, але заборонити доступ до користувацьких даних. У випадку компрометації УЗ адміністратора максимум, що буде зроблено – це видалення ВМ і даних. Що у свою чергу можна відновити з бекапів. Зате не буде «злита» важлива інформація.

Співробітникам ІБ треба надати доступ до даних співробітників, але заборонити доступ до редагування налаштувань.

Тверде розмежування прав доступу методом «заборонити все, крім необхідного» дуже важливий крок, на який багато закривають очі.

4. Перехоплення віртуальної пам'яті машин при живій міграції

Продукція компанії VMWare уже не перший рік радує нас чудовим інструментом vmotion. Він, що дозволяються в рамках HA кластера здійснювати переїзд ВМ між хостами без вимикання системи. Така процедура проводиться як у ручному режимі, наприклад, для звільнення хоста нужденного в обслуговуванні, так і в автоматичному режимі засобами DRS (постійне балансування навантаження між хостами). При включенні максимальної чутливості DRS до змін міграції відбуваються досить часто. У кластері з 100 ВМ і 10 хост машин, кількість міграцій за день досягає 70-100 раз.

З урахуванням такої частоти з'явився один з популярних видів атак – перехоплення трафіку оперативної пам'яті при міграції від одного хоста до іншого. Таким способом можна одержати важливу інформацію, наприклад, пари «логін/пароль», якщо вони зберігалися у відкритому виді. А так само просто «убити» машину, якщо пам'ять передається на хост-приймач не повністю або невірна. Атака проводиться методом «MITM» (людина посередині, man-in-the-middle).

Основна рекомендація в такій ситуації – поділ мереж. Мережа керування хостами й мережа vmotion повинні бути ізольовані друг від друга. В ідеалі, мережа міграції краще зробити локальної.

Далі ми розглянемо:

- атака «відмова в обслуговуванні»;
- перехід від менш привілейованих віртуальних машин (ВМ) до керуючих;
- вірусні атаки;
- брутфорс і експлойти по незакритих портах;
- централізоване логування;
- керування паролями.

Атака «відмова в обслуговуванні»

Крім крадіжки коштовної інформації з метою перепродажу або вимагання також не варто забувати про атаки «відмова в обслуговуванні». До них ставляться DDoS атаки й будь-які інші, які приводять до часткової або повній непрацездатності сервісу.

Щоб запобігти таким атакам існує безліч інструментів:

- апаратні пристрої, які беруть на себе навантаження від атаки;
- софтверні застосунки, які детектують небезпечний трафік і переривають спілкування;
- веб-сервіси, які також можуть захистити від частини типів DDoS атак (наприклад, для сайтів DNS DDoS).

У даній ситуації кращим рішенням буде обмеження виділених потужностей на ВМ. У продуктів VMWare за замовчуванням усі машини створюються з безлімітними ресурсами ЦП, що може привести до серйозних проблем з хост машиною, на якій у свою чергу можуть розташовуватися інші важливі ВМ. У випадку атаки на конкретну ВМ, вона забере всі ресурси з хоста й, у найкращому разі, ми одержимо «гальма» на інших ВМ цього хоста, а в найгіршому разі, падіння всієї хост машини. Тому обов'язково вносите обмеження по виділенню ресурсів для ВМ.

Перехід від менш привілейованих ВМ до керуючих

Одне з головних правил мережної безпеки – це сегментування мережі. Це ж ставиться й до забезпечення безпеки віртуальної інфраструктури.

Якщо ви використовуєте у своїй компанії VDI (є ВМ, які використовують прості користувачі) або у вас є тестові машини з низьким рівнем привілеїв, то ризик компрометації цих машин набагато вище, чим серверів, які не «ходять» в інтернет і т.д. Тому такого типу машини необхідно ізолювати від критично важливих ВМ. Це робиться, наприклад, такими маніпуляціями:

- Рознесенням ВМ по різним хостам (нагадую, що одержавши доступ до ВМ, можна одержати доступ до хост машині).

– Поділом мереж на VLAN. Якщо ви використовуєте послуги хостингу, і у вас немає доступу до залізного мережного устаткування, ви з легкістю можете налаштувати це на віртуальному устаткуванні від VMWare (distribution switch і більш просунутий NSX).

І т.д.

Вірусні атаки

Віруси, мабуть, найчастіші погрози, які зустрічалися кожному. Колись це просто баннери, настирлива реклама, «глюки» системи, але бувають і більш неприємні прояви:

– Шифрувальники. Шифрують геть усе, аж до системних файлів, але так, щоб ціль могла все-таки зайти в систему й прочитати їхня пропозиція до оплати для розшифрування.

– Бекдори. Створюють із ваших улюблених серверів «дедики» – сервера, на які можна заходити в будь-який зручний час для здійснення своїх «капосних» справ, щоб замести сліди.

– Ботнет. Підключають ваші сервера до ботнет мережі, для спільних атак (DDoS і т.д.).

і т.д.

Для захисту віртуальної інфраструктури є три типи антивірусного захисту:

1. **Агентська**

Класичний антивірус, який установлюється на кожну ВМ і працює автономно, тільки для цієї ВМ.

2. **Гібридна**

На ВМ установлюють тонкий агент, який аналізує, що відбувається на ВМ, але самі бази зберігаються й основні маніпуляції відбуваються на окремому загальному сервері.

3. **Безагентська**

На самі ВМ не встановлюється нічого. Захист здійснюється при використанні технології vshield. На хост машинах створюється додатковий адаптер, а активності на ВМ перевіряє безпосередньо сам хост.

Безумовно, самим пріоритетним уважається метод безагентського захисту. Він показує найвищі показники продуктивності. Особливо це яскраво помітно при одночасному включенні (або login) на ВМ, наприклад, ранком, коли всі ваші співробітники одночасно заходять на свої машини. При агентському захисті під час завантаження відбувається перевірка файлів кожної ВМ одночасно, що створює чимале навантаження на дискову підсистему й інші ресурси.

Одним з важливих мінусів безагентського захисту є сильно урізаний функціонал керування окремими машинами. Усі правила, більшим ступенем, поширюються на хост машини. Це зручно, коли у вас однотипні ВМ на тому самому хосте, але якщо це сервера, які вимагають індивідуальних налаштувань, той цей застосунок не саме зручне. Коли ми набудовували подібний застосунок в одній з телекому компаній середнього рівня, нам довелося пожертвувати продуктивністю й використовувати класичний агентський варіант.

Гібридний варіант у цей момент розбудовується дуже активно, але поки ми гідного продукту не виділили.

Брутфорс і експлойти по незакритих портах

Помніть, кожний незакритий порт – це мільйон можливостей для зловмисника по злому вашої системи. Будь-яка атака починається з команди nmap -ss «ваш IP адреса». Після аналізу портів, будується вектор атаки, і починаються самі дії.

Тому необхідно закривати всі, чому ви не користуєтеся, а тим, що використовуєте, обмежити по remote ip, благо в продуктах VMWare досить широкі можливості файєрвола.

Якщо порти вашого сервісу використовують із різних IP? і це обмежити не виходить, то не забувайте використовувати системи протидії брутфорсу, такі як: fail2ban (для Linux), ipban (для Windows) і т.д. Вони дозволяють блокувати IP, з якого приходять запити на авторизацію після n-ї кількості невдалих спроб. В vsphere, починаючи з 6.0, даний інструмент уже вбудований у систему.

PS. І не забувайте додати у виключення свої адреси.

Централізоване логування

Якщо, незважаючи на всі ваші вжиті заходи по забезпеченню безпеки з вами приключилася неприємна подія й вас зламали, потрібно й у цьому випадку винести для себе користь. А саме – знайти причину, по якій вийшло скомпрометувати ваш ресурс і внести виправлення в інші ваші налаштування, щоб історія не повторилася.

Як же знайти сліди входу?

Перше, що приходить на розум – логи сервера. Уже там-те точно повинне бути зафіксоване: хто заходив, звідки й що робив. Але заглянувши в журнал, ми бачимо, що він порожній. Навіть роботи, які ходять по просканованим підмережам зачищають свої дії. Чого вуж говорити про цілеспрямовану атаку.

Щоб не виявитися в такій ситуації, необхідно використовувати централізований збирач балок. Якщо локально логи будуть очищені, у нас завжди буде копія на сервері логування. Також, збирач балок дуже корисний у повсякденній роботі. Наприклад, для пошуку IP адреси по MAC, якщо у вас декілька DHCP і ви не знаєте який саме його видав і т.д.

Керування паролями

Так, менеджер паролів потрібний не тільки користувачам ПК. Потрібний він, тому що кожна система повинна мати свій унікальний пароль. Для кожного повинна бути знайома ситуація, коли їсти завдання розгорнути новий сервіс і для цього створюється демо-зона (тестова машина). Потім міняються пріоритети й проект відкладається. Машина звивається запущеної з нашим стандартним адмінським паролем. Виходять нові 0-day уразливості, але нашу тестову машину ніхто не обновляє, про неї забули. Згодом вона стає відкритою дверима для проникнення й одержання нашого стандартного адмін-пароля (який один для всіх систем!).

Тому потрібно побрати за правило: яка б це система не була (тестова або продакшн), розвертається на 5 хв, на 10 хв або на роки – завжди, якщо машина має мережний адаптер, потрібно генерувати їй 15+ символьний пароль із

максимальним рівнем складності (більші маленькі букви, цифри, спецсимволи і т.д.) і затягати цей унікальний пароль у менеджер паролів. Це простої й немаловажна дія жоден раз урятує вас від серйозних наслідків.

3.2 Розробка структурної схеми

Основним засобом боротьби з вірусами були й залишаються антивірусні програми. Можна використовувати антивірусні програми (антивіруси), не маючи представлення про те, як вони влаштовані. Однак без розуміння принципів пристрою антивірусів, знання типів вірусів, а також способів їх поширення, не можна організувати надійний захист комп'ютера. Як результат, комп'ютер може бути заражений, навіть якщо на ньому встановлені антивіруси.

Сьогодні використовується кілька основних методик виявлення й захисту від вірусів:

- сканування;
- евристичний аналіз;
- використання антивірусних моніторів;
- виявлення змін;
- використання антивірусів, вбудованих в BIOS комп'ютера.

Крім того, практично всі антивірусні програми забезпечують автоматичне відновлення заражених програм і завантажувальних секторів. Звичайно, якщо це можливо.

Сканування

Найпростіша методика пошуку вірусів полягає в тому, що антивірусна програма послідовно переглядає, що перевіряються файли в пошуку сигнатур відомих вірусів. Під сигнатурою розуміється унікальна послідовність байт, що належить вірусу, що й не зустрічається в інших програмах.

Антивірусні програми-сканери здатні знайти тільки вже відомі й вивчені віруси, для яких була визначена сигнатура. Застосування простих програм-сканерів не захищає Ваш комп'ютер від проникнення нових вірусів.

Для, що шифруються й поліморфних вірусів, здатних повністю змінювати свій код при зараженні нової програми або завантажувального сектору, неможливо виділити сигнатуру. Тому прості антивірусні програми-сканери не можуть виявити поліморфні віруси.

Евристичний аналіз

Евристичний аналіз дозволяє виявляти раніше невідомі віруси, причому для цього не треба попередньо збирати дані про файлову систему, як цього вимагає, наприклад, розглянутий нижче метод виявлення змін.

Антивірусні програми, що реалізують метод евристичного аналізу, перевіряють програми й завантажувальні сектори дисків і флешек, намагаючись виявити в них код, характерний для вірусів. Евристичний аналізатор може виявити, наприклад, що програма, що перевіряється, установлює резидентний модуль у пам'яті або записує дані в здійсненний файл програми.

Практично всі сучасні антивірусні програми реалізують власні методи евристичного аналізу.

Коли антивірус виявляє заражений файл, він звичайно виводить повідомлення на екрані монітора й робить запис у власному або системному журналі. Залежно від налаштувань, антивірус може також направляти повідомлення про виявлений вірус адміністраторові мережі.

Якщо це можливо, антивірус виліковує файл, відновлюючи його вміст. А якщо ні, то пропонується тільки одна можливість – вилучити заражений файл і потім відновити його з резервної копії (якщо, звичайно, вона у Вас є).

Антивірусні монітори

Існує ще цілий клас антивірусних програм, які постійно перебувають у пам'яті комп'ютера, і відслідковують усі підозрілі дії, виконувані іншими програмами. Такі програми звуться антивірусних моніторів або сторожів.

Монітор автоматично перевіряє всі програми, що запускаються, створювані, відкриваються документи, що й зберігаються, файли програм і документів, отримані через Інтернет або скопійовані на жорсткий диск із флешки й компакт-диска. Антивірусний монітор повідомить користувача, якщо яка-небудь програма спробує виконати потенційно небезпечна дія.

Виявлення змін

Коли вірус заражає комп'ютер, він змінює вміст жорсткого диска, наприклад, дописує свій код у файл програми або документа, додає виклик програми-вірусу у файл AUTOEXEC.BAT, змінює завантажувальний сектор, створює файл-супутник. Таких змін, однак, не роблять «безтілесні» віруси, що живуть не на диску, а в пам'яті процесів ОС.

Антивірусні програми, називані ревізорами диска, не виконують пошук вірусів по сигнатурах. Вони запам'ятовують попередньо характеристики всіх областей диска, які зазнають нападу вірусу, а потім періодично перевіряють їх (звідси відбувається назва програми-ревізори). Ревізор може знайти зміни, зроблені відомим або невідомим вірусом.

Захист, вбудований в BIOS комп'ютера

У системні плати комп'ютерів теж вбудовують найпростіші засоби захисту від вірусів. Ці засоби дозволяють контролювати всі звертання до головного завантажувального запису жорстких дисків, а також до завантажувальних секторів дисків і флешек. Якщо яка-небудь програма спробує змінити вміст завантажувальних секторів, спрацьовує захист і користувач одержує відповідне попередження.

Однак цей захист не дуже надійний. Існують віруси (наприклад, Tchechen.1912 і 1914), які намагаються відключити антивірусний контроль BIOS, змінюючи деякі гнізда в енергонезалежній пам'яті (CMOS-пам'яті) комп'ютера.

Особливості захисту корпоративної інтрамережі

Корпоративна інтрамережа може нараховувати сотні й тисячі комп'ютерів, що відіграють роль робочих станцій і серверів. Ця мережа звичайно підключена

до Інтернету й у ній є поштові сервери, сервери систем автоматизації документообігу, такі як Microsoft Exchange і Lotus Notes, а також нестандартні інформаційні системи.

Для надійного захисту корпоративної інтрамережі необхідно встановити антивіруси на всі робочі станції й сервери. При цьому на файл-серверах, серверах електронної пошти й серверах систем документообігу слід використовувати спеціальне серверне антивірусне програмне забезпечення. Що ж стосується робочих станцій, їх можна захистити звичайними антивірусними сканерами й моніторами.

Розроблені спеціальні антивірусні проксі-сервери й брандмауери, скануючі минаючий через них трафік програмні компоненти, що й видаляють із нього шкідливі. Ці антивіруси часто застосовуються для захисту поштових серверів і серверів систем документообігу.

Захист файлових серверів

Захист файлових серверів повинна здійснюватися з використанням антивірусних моніторів, здатних автоматично перевіряти всі файли сервера, до яких іде обіг по мережі. Антивіруси, призначені для захисту файлових серверів, випускають усі антивірусні компанії, тому у Вас є багатий вибір.

Захист поштових серверів

Антивірусні монітори неефективні для виявлення вірусів у поштових повідомленнях. Для цього необхідні спеціальні антивіруси, здатні фільтрувати трафік SMTP, POP3 і IMAP, крім влучення заражених повідомлень на робочі станції користувачів.

Для захисту поштових серверів можна придбати антивіруси, спеціально призначені для перевірки поштового трафіку, або підключити до поштового сервера звичайні антивіруси, що допускають роботу в режимі командного рядка.

Захист серверів систем документообігу

Сервери систем документообігу, такі як Microsoft Exchange і Lotus Notes, зберігають документи в базах даних власного формату. Тому використання

звичайних файлових сканерів для антивірусної перевірки документів не дасть ніяких результатів.

Існує ряд антивірусних програм, спеціально призначених для антивірусного захисту подібних систем. Ці програми сканують пошту й файли вкладень, видаляючі в реальному часі всі шкідливі програми, виявляють макрокомандні віруси й троянські програми у формах і макросах, у файлах сценаріїв і в об'єктах OLE. Перевірка виконується в режимі реального часу, а також на вимогу.

Захист нестандартних інформаційних систем

Для антивірусного захисту нестандартних інформаційних систем, що зберігають дані у власних форматах, необхідно або вбудовувати антивірусне ядру в систему, або підключати зовнішній сканер, що працює в режимі командного рядка.

Мережний центр керування антивірусами

Якщо інтрамережа нараховує сотні й тисячі комп'ютерів, то необхідно централізоване віддалене керування антивірусними програмами й контроль їх роботи. Виконання в «ручному» режимі таких операцій, як відстеження відновлень антивірусної бази даних і завантажувальних модулів антивірусних програм, контроль ефективності виявлення вірусів на робочих станціях і серверах і т.п., малоефективно, якщо в мережі є велика кількість користувачів або якщо мережа складається з територіально віддалених друг від друга сегментів.

Якщо ж не забезпечити своєчасне й ефективне виконання перерахованих вище операцій, технологія антивірусного захисту корпоративної мережі обов'язково буде порушена, що рано або пізно приведе до вірусного зараження. Наприклад, користувачі можуть неправильно налаштувати автоматичне відновлення антивірусної бази даних або просто виключати свої комп'ютери в той час, коли таке відновлення виконується. У результаті автоматичне відновлення не буде виконано й виникне потенційна погроза зараження новими вірусами.

У сучасних антивірусних системах реалізовані наступні функції віддаленого керування й контролю:

– установка й відновлення антивірусних програм, а також антивірусних баз даних;

– централізована дистанційна установка й налаштування антивірусів;

– автоматичне виявлення нових робочих станцій, підключених до корпоративної мережі, з наступної автоматичною установкою на ці станції антивірусних програм;

– планування завдань для негайного або відкладеного запуску (таких як відновлення програм, антивірусної бази даних, сканування файлів і т.п.) на будь-яких комп'ютерах мережі;

– відображення в реальному часі процесу роботи антивірусів на робочих станціях і серверах мережі.

Мережні центри керування дозволяють управляти антивірусним захистом усієї мережі з однієї робочої станції системного адміністратора. При цьому для прискорення процесу установки антивірусів у віддалених мережах, підключених до основної мережі повільними каналами зв'язки, у цих мережах створюються власні локальні дистрибутивні каталоги.

При використанні клієнт-серверної архітектури основою мережного центру керування є антивірусний сервер, установлений на одному із серверів корпоративної мережі. З ним взаємодіють, з одного боку, програми-агенти, установлені разом з антивірусами на робочих станціях мережі, а з іншого сторони – керуюча консоль адміністратора антивірусного захисту.

Антивірусний сервер виконує керуючі й координуючі дії. Він зберігає загальний журнал подій, що мають відношення до антивірусного захисту й виникаючих на всіх комп'ютерах мережі, список і розклад виконання завдань. Антивірусний сервер відповідає за приймання від агентів і передачу адміністраторові антивірусного захисту повідомлень про виникнення тих або інших подій у мережі, виконує періодичну перевірку конфігурації мережі з метою

виявлення нових робочих станцій або робочих станцій з, що змінився конфігурацією антивірусних засобів і т.д.

Крім агентів, на кожній робочій станції й сервері корпоративної мережі встановлюється антивірус, що виконує сканування файлів і перевірку файлів при їхнім відкритті (функції сканера й антивірусного монітора). Результати роботи антивірусу передаються через агентів антивірусному серверу, яких їх аналізує й протоколює в журналі подій.

Керуюча консоль може являти собою стандартний застосунок Microsoft Windows з віконним інтерфейсом або аплет (snap-in) керуючої консолі Control Panel операційної системи Microsoft Windows.

Користувацький інтерфейс керуючої консолі дозволяє переглядати деревоподібну структуру корпоративної мережі, одержуючи при необхідності доступ до окремих комп'ютерів тих або інших груп користувачів або доменів.

Багаторівневі системи з WEB-інтерфейсом

Архітектура багаторівневих систем з WEB-інтерфейсом припускає використання WEB-сервера як ядра системи. Завданням цього ядра є, з одного боку, організація діалогової інтерактивної взаємодії з користувачем, а з іншого – із програмними модулями тієї або іншої системи.

Переваги такого підходу полягають в уніфікації способів керування різними системами мережі, а також у відсутності необхідності встановлювати на робочу станцію адміністратора які-небудь керуючі програми або консолі. Адміністрування може виконуватися з будь-якого комп'ютера мережі, а якщо мережа підключена до Інтернету, то з будь-якого місця земної кулі, де є Інтернет і комп'ютер із браузером.

Для захисту керуючої інформації при її передачі по Інтернеті або корпоративній інтрамережі застосовуються протоколи SSH або інші аналогічні засоби (наприклад, власні захищені модифікації протоколу HTTP).

Ця система дозволяє повністю управляти й контролювати роботу корпоративної системи антивірусного захисту з однієї робочої станції через

браузер, навіть якщо окремі фрагменти мережі перебувають у різних країнах або на різних континентах.

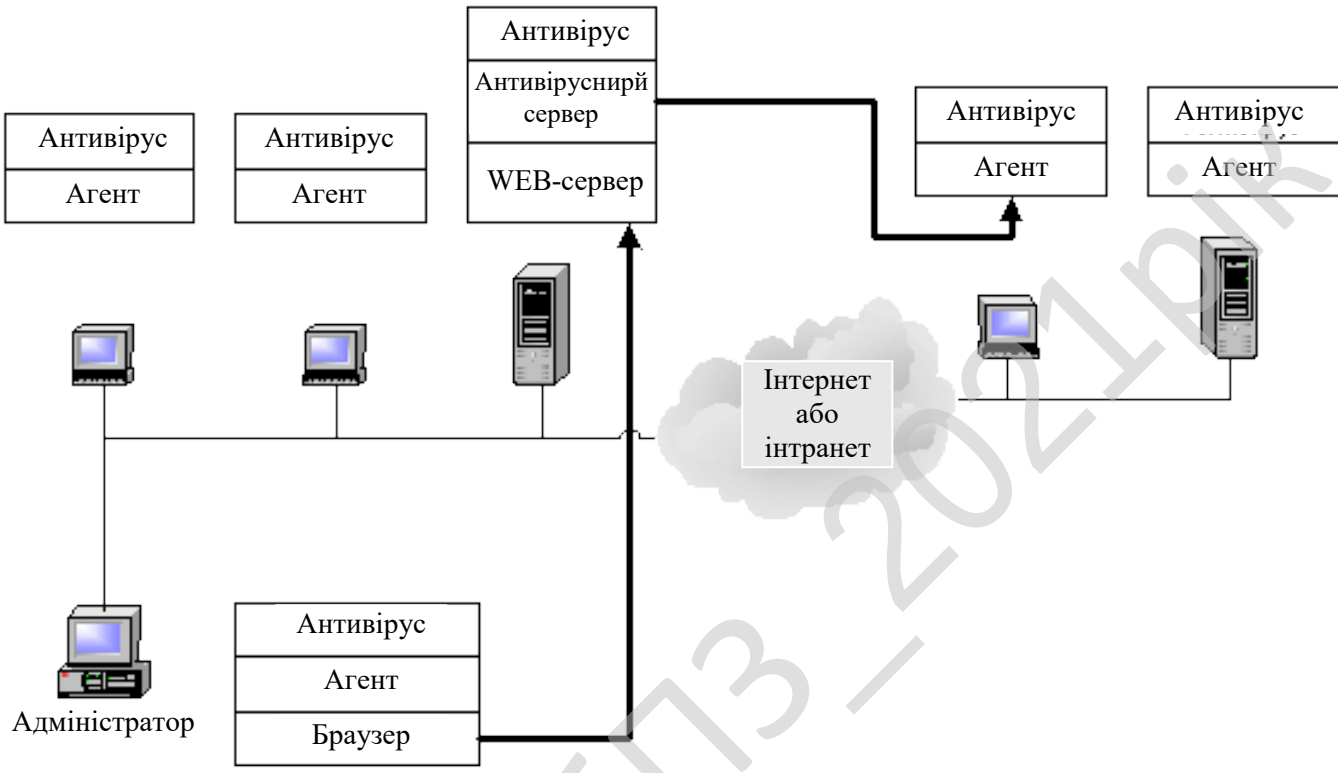

Рисунок 3.1 – Структурна схема системи

Адміністратор антивірусного захисту управляє її роботою через браузер, а не через консольний застосунок.

На робочих станціях установлюється антивірус. Цей антивірус управляється антивірусним сервером через агента.

На комп'ютері, що відіграє роль антивірусного сервера, установлюється WEB-сервер Microsoft IIS. Спеціальне WEB-застосунок, що працює на цьому сервері, управляє антивірусним сервером. Воно також надає адміністраторові користувацький інтерфейс для керування системою антивірусного захисту.

Що ж стосується повідомлень про виникнення подій у корпоративній системі антивірусного захисту, то вони передаються програмами-агентами

серверу й розсилаються по електронній пошті, по пейджинговим мережам, через системи SMS і т.п.

Адміністративно-технологічні методи захисту

Для того щоб антивірусні програми ефективно виконували свої функції, необхідно строго дотримувати рекомендацій з їхнього застосування, описані в документації. Особлива увага слід звернути на необхідність регулярного відновлення вірусних баз даних і програмних компонентів антивірусів. Сучасні антивіруси вміють завантажувати файли відновлень через Інтернет або по локальній мережі. Однак для цього їх необхідно налаштувати відповідним чином.

Однак навіть без застосування антивірусних програм можна постаратися запобігти проникненню вірусів у комп'ютер і зменшити шкода, яку вони нанесуть у випадку зараження. От що випливає для цього зробити в першу чергу:

– блокуйте можливі канали проникнення вірусів: не підключайте комп'ютер до Інтернету й локальної мережі компанії, якщо в цьому немає необхідності, відключите пристрою зовнішньої пам'яті, такі як пристрою CD-ROM і USB-порти;

– настройте параметри BIOS таким чином, щоб завантаження ОС виконувалася тільки з жорсткого диска, але не із флешок;

– забороните програмну зміну вмісту енергонезалежної пам'яті BIOS;

– виготовте системну завантажувальну флешку, записавши на неї антивіруси й інші системні утиліти для роботи з диском, а також диск аварійного відновлення Microsoft Windows;

– перевіряйте всі програми й файли документів, записувані на комп'ютер, а також флешки за допомогою антивірусних програм новітніх версій;

– установлюйте програмне забезпечення тільки з ліцензійних компакт-дисків;

– установите на всіх флешках захист від запису й знімайте її тільки якщо буде потреба;

– обмежте обмін програмами й флешками;

– регулярно виконуйте резервне копіювання даних;

– установлюйте мінімально необхідні права доступу до каталогів файлового сервера, захищайте від запису каталоги дистрибутивів і програмних файлів;

– складіть інструкцію для користувачів по антивірусному захисту, описавши в ній правила використання антивірусів, правила роботи з файлами й електронною поштою, а також опишіть дії, які слід почати при виявленні вірусів.

Проблема домашніх комп'ютерів

Часто співробітники компаній працюють не тільки в офісі, але й будинку, обмінюючись файлами між домашнім комп'ютером і офісною робочою станцією. Системний адміністратор компанії не в змозі захистити від вірусів усі домашні комп'ютери співробітників. Віруси можуть потрапити на домашній комп'ютер з Інтернету, а також у результаті обміну ігровими програмами. Найчастіше це відбувається, якщо до домашнього комп'ютера мають доступ інші члени родини й діти.

Усі файли, які співробітники приносять із будинку на роботу, слід розглядати як потенційно небезпечні. У відповідальних випадках такий обмін слід повністю заборонити, або сильно обмежити. Потенційно небезпечні «домашні» файли необхідно перевіряти перед відкриттям антивірусними програмами.

Установка персональних брандмауерів

Корпоративна мережа, підключена до Інтернету, повинна бути захищена від атак хакерів за допомогою брандмауера. Однак крім цього можна додатково захистити робочі станції й сервери мережі, установивши на них персональні брандмауери.

Крім фільтрації небажаного трафіку, деякі персональні брандмауери здатні захистити комп'ютер від троянських аплетів Java і елементів керування ActiveX. Такі компоненти можуть бути вбудовані в поштові повідомлення формату HTML і в сторінки троянських WEB-сайтів.

Персональні брандмауери, що перебувають у так званому режимі навчання, можуть надати допомогу у виявленні трафіку від троянських програм, логічних бомб і інших небажаних шкідливих компонентів. Коли такий компонент спробує встановити зв'язок з комп'ютером хакера, брандмауер відобразить на екрані попереджуюче повідомлення.

Слід помітити, що в налаштуваннях браузера Ви також можете відключити можливість використання активних компонентів, таких як аплети Java і елементів керування ActiveX. Однак персональні брандмауери більш універсальні, і дозволяють блокувати використання таких компонентів будь-якими програмами, наприклад, поштовими клієнтами

3.3 Розробка функціональної схеми

Розглянемо функціональні можливості розробленого в даній роботі програмного забезпечення системи кібербезпеки віртуальної інфраструктури, а саме хмарного антивірусу – VirtualInfrastructureCybersecurity. Користувацький інтерфейс відображає екран статусу, що вказує на те, чи є які-небудь помилки в системі або з нею все в порядку. Якщо в цей момент йде сканування, то користувач бачить зелені смужки, що проходять «хвилями» по рядкові процесів. Якщо сканування не проводиться, то користувач може його запустити за допомогою значка «Сканування».

Користувач може налаштувати деякі опції сканування:

- Оптимізоване сканування: сканування певних самої програмою папок і файлів у системі Windows. Тих папок і файлів, які найбільш уразливі для існуючого шкідливого коду.
- Повне сканування: сканування системи Windows цілком. У тому випадку, якщо в користувача на жорсткому диску багато даних, це може зайняти багато часу.
- Вибіркове сканування: користувач сам вибирає ті папки, які він праг би перевірити на присутність шкідливого коду.

Коли сканування завершується, VirtualInfrastructureCybersecurity повідомляє користувача про ті проблеми, які були виявлені на його комп'ютері. Програма не виводить список усіх цих проблем відразу. Вона показує користувачеві: все добре або не вс е добре. Якщо система не в порядку, користувач може клацнути за повідомленням, у якому зазначено скільки проблем було виявлено в процесі сканування.

Або клацнути на значку звіту. У такий спосіб можна одержати інформацію про ті типи злобливого коду, які були виявлені на комп'ютері. У цьому звіті присутня посилання на повний «звіт про події», що містить інформацію про все, що проробила програма під час сканування, усіх виявлених шкідливих файлах і про те, що з ними зробив антивірус.

Як і інші антивірусні застосунки, VirtualInfrastructureCybersecurity помістить у карантин деякі підозрілі файли, про яких програма не знає точно, є вони шкідливими чи ні. В VirtualInfrastructureCybersecurity карантин називається «кошиком». Це значить, що дані файли знешкоджені шляхом їхнього перейменування таким чином, щоб їх було непросто знайти. Так антивірусний сервіс дає користувачеві можливість відновити підозрілий файл.

Кожний хмарний антивірус має як налаштування за замовчуванням, так і дає користувачеві можливість їх зміни. Якщо нажати на відповідний значок у нижньому правому куті програми, з'являється вікно налаштувань. Налаштувати можна наступні речі:

– Налаштувати параметри з'єднання через проксі-сервер. Домашнім користувачам це потрібно рідко, а в офісах зустрічається частенько.

– Включити або відключити автоматичне сканування.

– Включити або відключити попереднє сканування USB-накопичувачів. Це свого роду «вакцинація», що утрудняє вірусам можливість «стрибнути» із флешки на жорсткий диск комп'ютера. Але ця можливість є тільки у версії Pro.

– Здійснити розширені налаштування, що ставляться до процесу сканування, кошику й звіту.

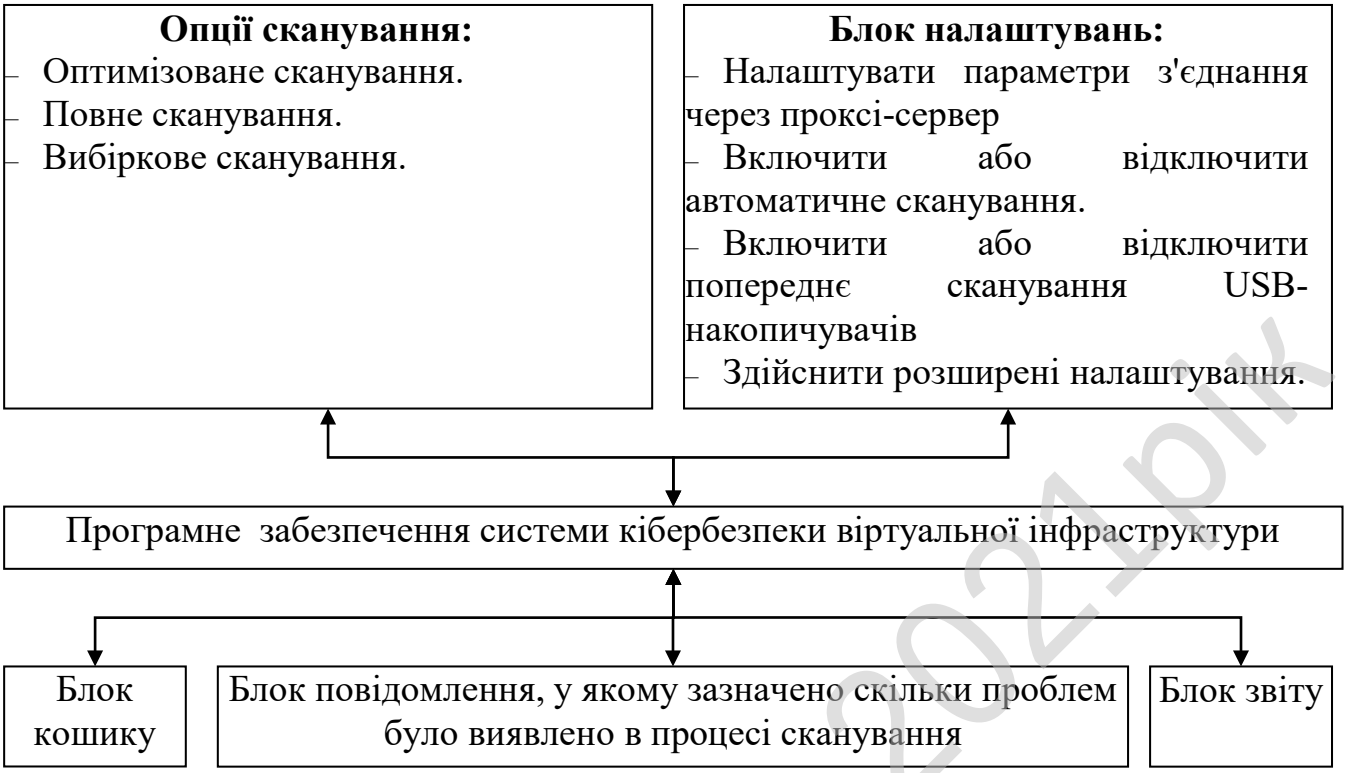

Рисунок 3.2 – Функціональна схема системи

Кому потрібний антивірус?

Історично слава головного об'єкта для вірусів дісталася операційним системам Windows. Прийнято вважати, що це пов'язане з уразливостями, що відкривають щілини в системі для шкідливого коду. Тому розроблювачі антивірусного ПЗ наголошують на своїх застосунках для Windows.

При цьому Mac і комп'ютери під керуванням Linux, всупереч розхожій думці, не є повністю невразливими для шкідливого коду. Для цих платформ теж створюються антивірусні програми, за допомогою яких користувачі можуть захистити свої комп'ютери.

За що критикують хмарні антивіруси?

Деякі люди не довіряють хмарним антивірусом, уважаючись, що їм не вистачає ряду важливих компонентів, відповідальних за безпеку й продуктивність. Викликає побоювання й той факт, що ці сервіси збирають інформацію з комп'ютерів користувачів. Виключення робиться для деяких

здійсненних файлів. Таким чином, зі списку переданих в «хмари» виключаються ті файли, які потенційно можуть містити персональну інформацію. Експерти рекомендують: перш, ніж скористатися хмарним антивірусом, слід з'ясувати, які файли можуть стати частиною бази даних сервісу.

Не все випробовують ентузіазм і відносно продуктивності антивірусів, які перебувають на комп'ютері користувача лише частково. Хмарні антивіруси можуть збільшувати час, необхідне для завантаження й вимикання комп'ютера. Крім того, вони можуть сповільнити роботу інших застосунків.

Затримка роботи комп'ютера може бути незначним, але при цьому відчутним. Але при цьому традиційні антивіруси теж знижують продуктивність системи. Так що ця якість не є особливістю хмарного антивірусу, а в цілому притаманно продуктам, що забезпечують захист комп'ютера від шкідливого коду.

Залишається відкритим питання про те, чи дійсно хмарні антивіруси уступають традиційним або ж користувачі просто вірні звичці? Звичка часом буває сильніше будь-яких логічних доводів.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3.

Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

Основні складові елементи діаграми взаємодії процесів це потоки даних:

– Репозиторії, потік сховища даних.

– Потоки зовнішні по відношенню до системи сутності.

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

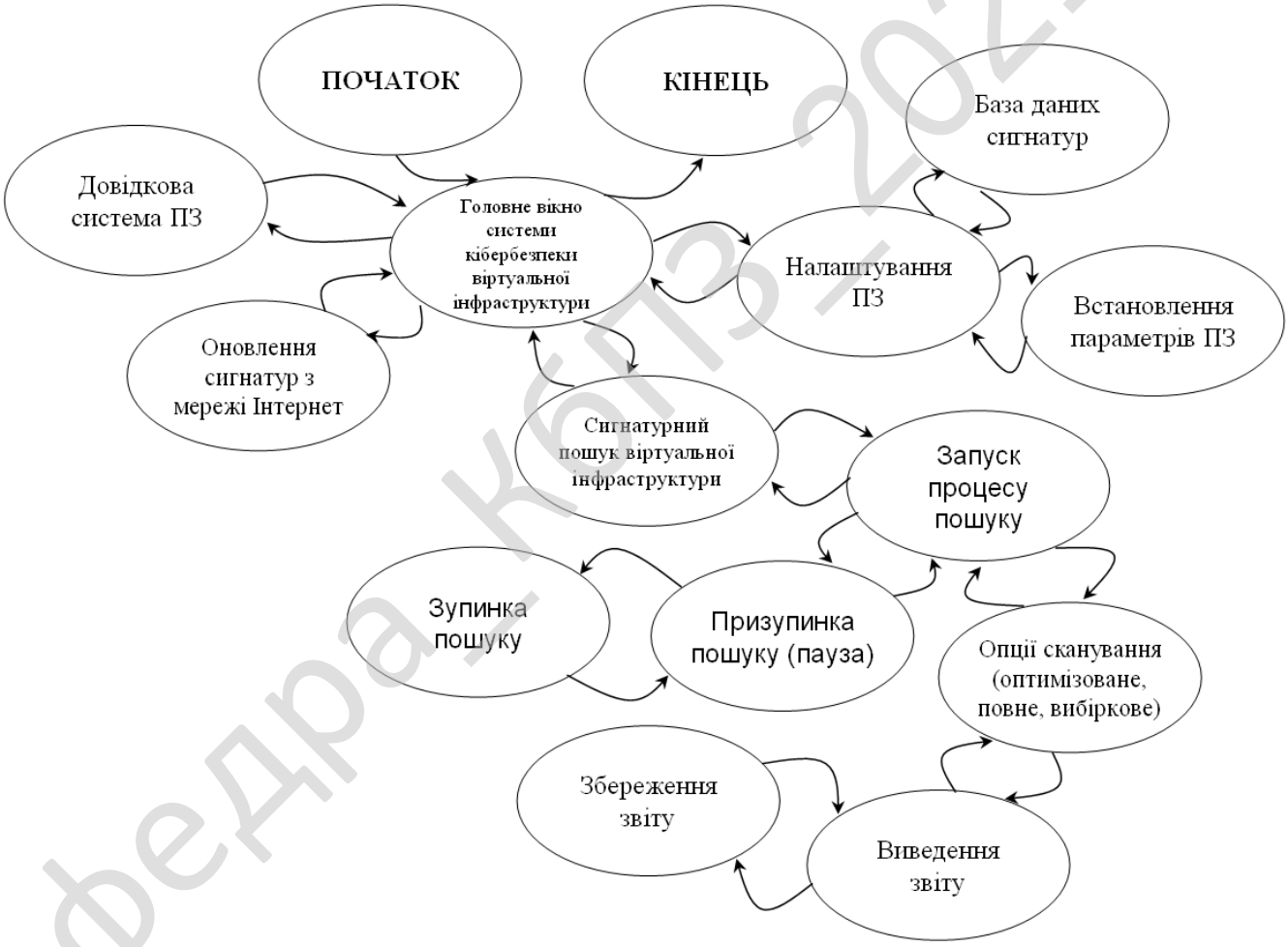

Рисунок 3.3 – Діаграма взаємодії процесів

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки

чому буде показано взаємодію системи в цілому у подальшому.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Послідовність дій та викликів підпрограм в загальному алгоритмі роботи основної програми що зображено на рисунку 4.1. у вигляді блок-схеми, розглянемо її детально:

– Виведення вікна системи кібербезпеки віртуальної інфраструктури.

– Формування та виведення списку ресурсів.

– Запит сканування ВІ.

– Формування списку ресурсів для сканування.

– Обрання користувачем об’єктів перевірки.

– Підпрограма пошуку загроз ВІ.

– Запит знайдено загрози.

– Виведення інформації пошуку.

– Запит загрозу видалено.

– Переміщення файлу загрози до папки карантину ПЗ.

– Формування файлу локального звіту.

– Запит – всі файли перевірено.

– Виведення звіту загроз пошуку.

– Запит моніторингу.

– Запуск моніторингу підсистем

– Підпрограма моніторингу підсистем.

– Виведення звіту роботи.

– Завершення роботи (кінець циклу запитів).

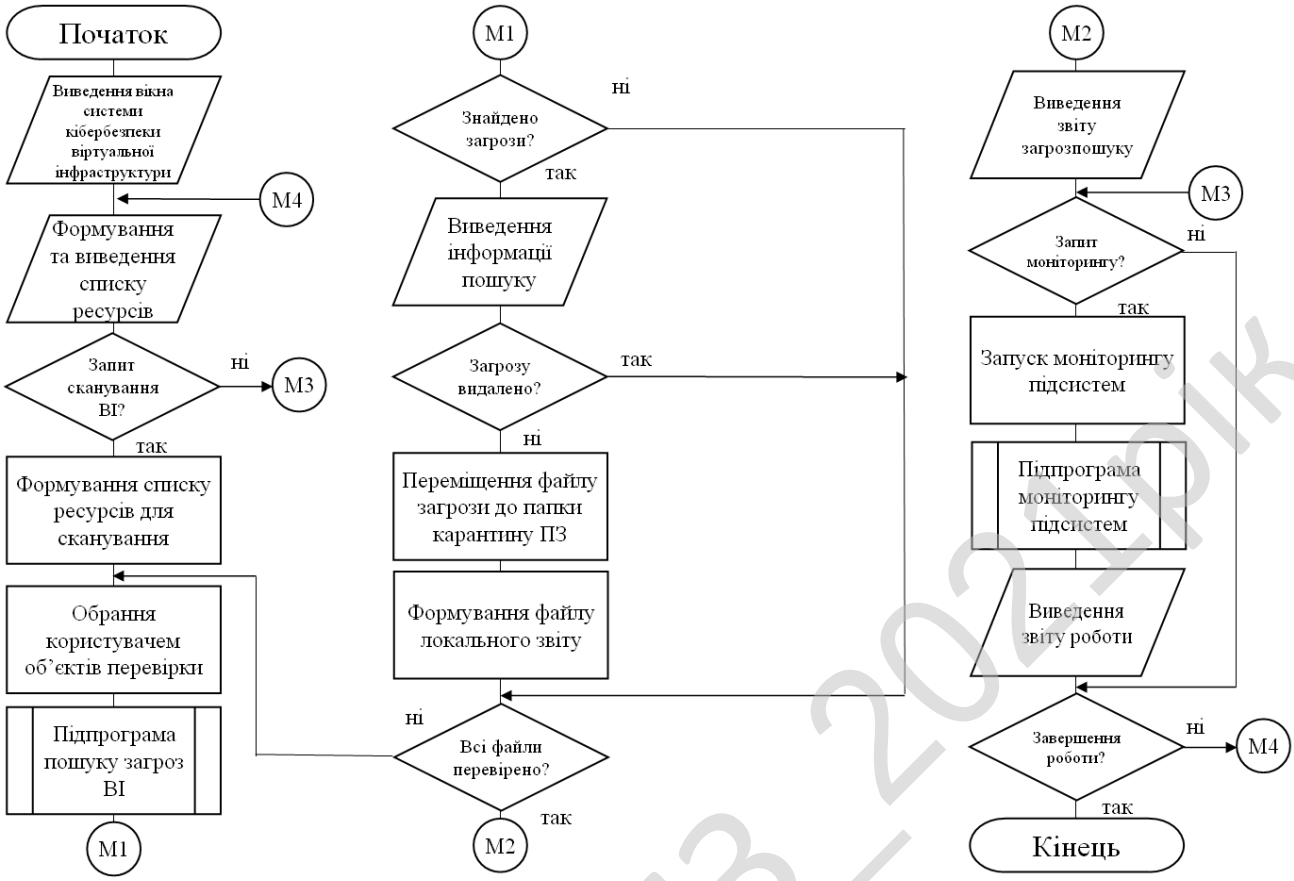

Рисунок 4.1 – Блок-схема основної програми

На рисунку 4.2 зображено роботу підпрограми з реалізацією наступних дій:

– Завантаження файлу перевірки.

– Запит файл архівовано.

– Спроба розпакування архіву.

– Перевірка CRC/MD5.

– Сигнатурний пошук.

– Отримання та формування даних до формату аналізу.

– Аналіз ВІ системою кібербезпеки.

– Запит загрозу знайдено.

– Визначення типу загрози.

– Запит – сигнатура коду відома.

– Запит можна вилікувати загрозу.

– Видалення коду загрози.

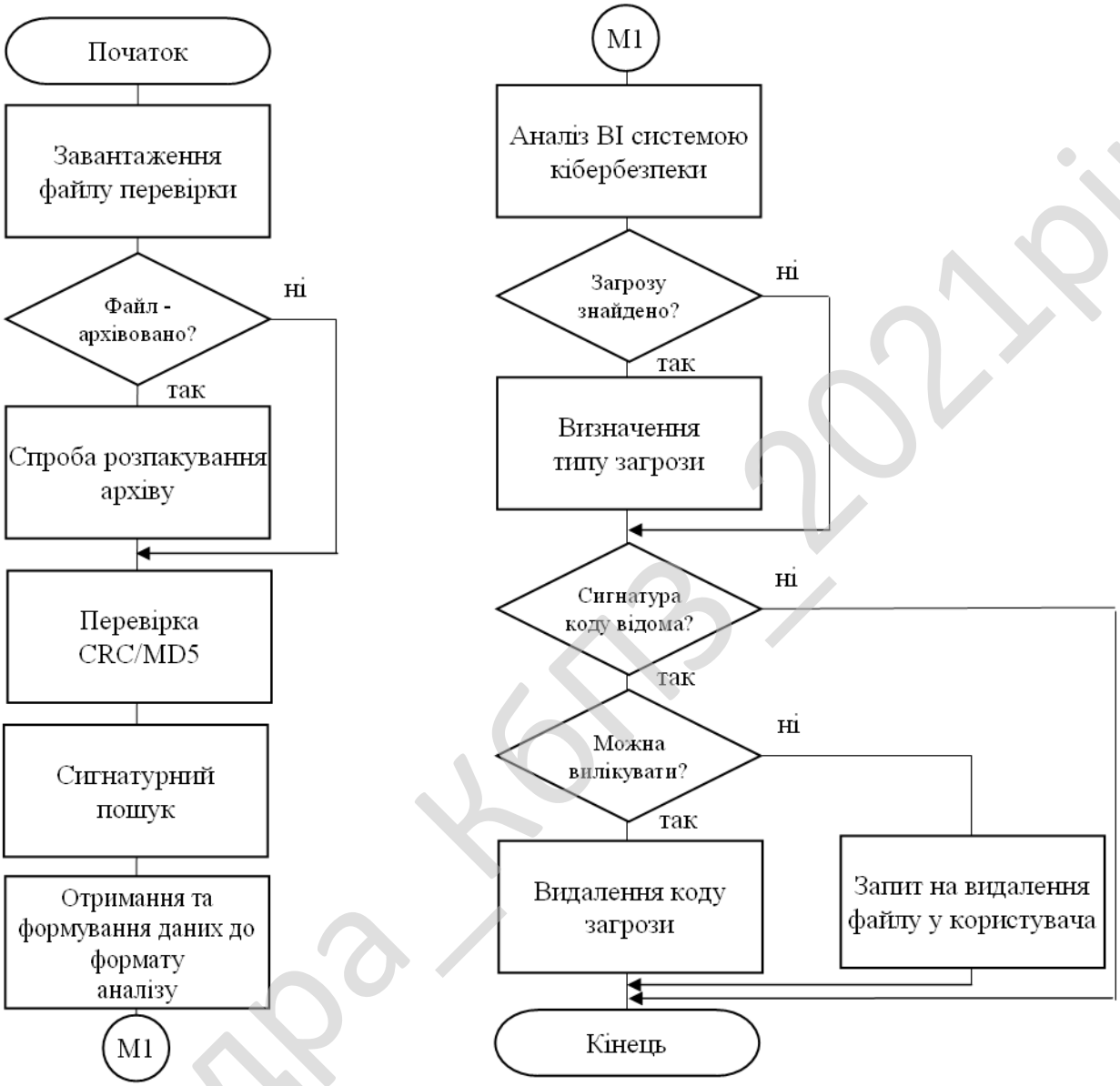

Рисунок 4.2 – Блок-схема роботи підпрограми

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так

званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики XP зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

Технологія екстремального програмування була розроблена Кентом Беком, Уардом Каннінґхемом та Роном Джеффріесом під час роботи над Chrysler Comprehensive Compensation System (C3). У 1996 Кент Бек став лідером проекту і почав вдосконалювати методи розробки, що застосовувалися в роботі над проектом. Свій метод він виклав у книзі «Extreme Programming Explained», котру було видано у жовтні 1999. Після купівлі Крайслера компанією Даймлер–Бенц проект C3 було скасовано у лютому 2000.

Хоча саме екстремальне програмування є відносно новим, багато її практик вже існували і використовувались протягом певного часу; однак, методологія підносить "найкращі практики" до екстремального рівня. Для прикладу, практика по плануванню і написанню тестів перед написанням кожної маленької частини коду було використано раніше в проекті НАСА "Меркурій".

Для зменшення часу на розробку ПЗ деякі формальні документи тестування (такі як приймальне тестування) писались паралельно (або й раніше) з написанням самого ПЗ. Незалежна група тестування НАСА може писати процедури тестування базуючись на формальних вимогах до продукту до того як програмне забезпечення розроблене та інтегроване в систему. В XP ця концепція піднесена до "екстремального рівня" завдяки написанню автоматичних тестів які перевіряють поведінку навіть малих частинок коду, а не тільки значних функціональних частин ПЗ.

Посібник Extreme Programming Explained: Embrace Change описує Екстремальне Програмування, як:

– Спроба примирити гуманність і продуктивність.

– Механізм для соціальної зміни.

– Шлях до удосконалення.

– Стиль розвитку.

Дисципліна розробки програмного забезпечення.

Головною метою Екстремального Програмування є скорочення вартості неочікуваних змін. У традиційних методах розробки (на кшталт SSADM) вимоги до розвитку системи визначаються на початку роботи над проектом, і часто виправляються пізніше. Це означає, що вартість проекту через зміни буде більшою за заплановану (традиційна особливість для програмного забезпечення, що проектується).

XP використовується для скорочення вартості змін, завдяки представленню простих значень, принципів і методів. При використанні екстремального програмування, проект повинен стати гнучкішим щодо змін.

Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше.

XP намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає

зміни до вимог як звичайні, неминучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваним. Основна ідея полягає в тому що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах – неминучі.

Екстремальне програмування також вводить набір практик та принципів на основі методології гнучкої розробки програмного забезпечення.

Екстремальне програмування описує чотири базові активності що виконуються при розробці програмного забезпечення: написання коду, тестування, слухання та дизайн. Написання коду. Прихильники XP заявляють що єдиним дійсно важливим результатом розробки ПЗ є код: без готового коду нема продукту.

Тестування. Методологія екстремального програмування заявляє, що якщо дрібне тестування може перевірити незначну частину функціональності, то багато дрібних тестів можуть перевірити набагато більше частинок і продукт в цілому.

Основні прийоми XP. Дванадцять основних прийомів екстремального програмування (за першим виданням книги Extreme programming explained) можуть бути об'єднані в чотири групи:

1. Короткий цикл зворотного зв'язку (Fine scale feedback).

1.1. Розробка через тестування (Test driven development).

1.2 Гра в планування (Planning game).

1.3. Замовник завжди поруч (Whole team, Onsite customer).

1.4 Парне програмування (Pair programming).

2. Безперервний, а не пакетний процес.

2.1 Безперервна інтеграція (Continuous Integration).

2.2 Рефакторинг (Design Improvement, Refactor).

2.3 Часті невеликі релізи (Small Releases).

3. Розуміння, що поділяється всіма учасниками.

3.1 Простота (Simple design).

3.2 Метафора системи (System metaphor).

3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).

3.4 Стандарт кодування (Coding standard or Coding conventions).

4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

– Ведення декількох проектів.

– Гнучка система доступу з використанням ролей.

– Система відстеження помилок.

– Діаграми Ганта та календар.

– Ведення новин проекту, документів та управління файлами.

– Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.

– Власна Wiki для кожного проекту.

– Форуми для кожного проекту.

– Облік часових витрат.

– Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

– Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).

– Створення записів про помилки на основі отриманих листів

– Підтримка LDAP автентифікації.

– Можливість самореєстрації нових користувачів.

– Багатомовний інтерфейс (у тому числі українська мова).

– Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання. показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

– номер (ідентифікатор) дефекту;

– хто повідомив про дефект;

– дата і час виявлення дефекту;

– версія продукту, в якій виявлено дефект;

– серйозність (критичність) дефекту та пріоритет рішення;

– опис кроків для відтворення дефекту (неправильної поведінки програми);

– відповідальний за усунення дефекту;

– обговорення можливих рішень та їх наслідків;

– поточний стан виправлення дефекту;

– версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.

2. Призначений – призначений відповідальний за виправлення дефекту.

3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

– Дубль (повторює дефект, що вже знаходиться в роботі).

– Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

Управління вимогами це процес запису, аналізу, трасування, пріоритезації і узгодження вимог та контролю змін і доведення до їх зацікавлених

сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покриють також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

Завдання управління вимогами

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзання меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, експортуючи дані вимог.

Аналіз здійсненності

На стадії аналізу здійсненності визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використовуваним інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?»

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягаря користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному

математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі ». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив, що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроектувати автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

Розробка та тестування. На стадії розробки і тестування, основна діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення

прототипу і ітераційне тестування. Для програмного додатка користувацький інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

Пошук файлів при повному скануванні

```
unit scandata;
// назва модулю
interface
// інтерфейс
Uses
//бібліотеки
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, FileCtrl, ComCtrls, ExtCtrls;
Type
// типи та клас форми
TForm1 = class(TForm)
    Button1: TButton;
    ListBox1: TListBox;
    Button2: TButton;
    DateTimePicker1: TDateTimePicker;
    Label2: TLabel;
    Bevel1: TBevel;
    ListBox2: TListBox;
    Edit2: TEdit;
    Label3: TLabel;
    Button3: TButton;
    Edit3: TEdit;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    Bevel2: TBevel;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
```

```
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure scanFile(Dir:String);
  end;

var
// екземпляр класу
  Form1: TForm1;

implementation
// реалізаційна частина
{$R *.dfm}
// підключення ресурсного файлу
procedure TForm1.Button1Click(Sender: TObject);
var
 Dir1,Dir2:string;
 SR:TSearchRec;
 scanRes:Integer;
// змінна для записи результату пошуку
begin
 SelectDirectory('Перегляд каталогу',Dir1,Dir2);
 ListBox1.Clear;
 scanRes:=scanFirst(Dir2+'\*.*',faAnyFile,SR);
 while scanRes=0 do
// поки ми знаходимо файли (каталоги),
// то виконувати цикл
// якщо знайдений елемент каталог і
  begin
   if ((SR.Attr and faDirectory)=faDirectory) and
// він має назву "." або "..", тоді:
   ((SR.Name='.')or(SR.Name='..')) then
    begin
     scanRes:=scanNext(SR);
// продовжити пошук
     Continue;
// продолжить цикл
    end;
// якщо у файлу (каталогу) дата створення менше,
// ніж встановлено в DateTimePicker1, то
```

```
   if FileDateToDateTime(SR.Time)<DateTimePicker1.Date then
    begin
     scanRes:=scanNext(SR);
// продовжити пошук
     Continue;
// продовжити цикл
    end;   }
   ListBox1.Items.Add(SR.Name+'      '+Dir2+'     '
     +IntToStr(SR.Size)+'  Bite');
// додавання в список знайденого элементу
   if Application.Terminated then Break;
   Application.ProcessMessages;
   scanRes:=scanNext(SR);
// продовження пошуку за заданими умовами
  end;
 scanClose(SR);
// закриваємо пошук
 Label2.Caption:='Items: '+IntToStr(ListBox1.Items.Count);
end;

procedure TForm1.scanFile(Dir:String);
var
 i:integer;
 SR:TSearchRec;
 scanRes:Integer;
 Dir1,Dir2:string;
begin
 Label4.Caption:='Path: - '+Dir2+'*.*';
 i:=0;
 scanRes:=scanFirst(Dir+'*.*',faAnyFile,SR);
 while scanRes=0 do
  begin
   if ((SR.Attr and faDirectory)=faDirectory) and
   ((SR.Name='.')or(SR.Name='..')) then
     begin
      scanRes:=scanNext(SR);
      Continue;
     end;
// якщо знайдений каталог, то
   if ((SR.Attr and faDirectory)=faDirectory) then
    begin
// входимо в процедуру пошуку з параметрами поточного каталогу
```

```
      scanFile(Dir+SR.Name+'\');
      scanRes:=scanNext(SR);
// після огляду вкладеного каталогу ми продовжуємо пошук
// в цьому каталозі
      Continue;
// продовжити цикл
     end;
    ListBox2.Items.Add(SR.Name+'       -       '+Dir+'       -       '
    +IntToStr(SR.Size)+' - Bite');
    if Application.Terminated then Break;
    Application.ProcessMessages;
    scanRes:=scanNext(SR);
   end;
 scanClose(SR);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
 ListBox2.Clear;
// очищення списку файлів
 scanFile(Edit2.Text);
// пошук файлів з початковими умовами
 Label3.Caption:='Items: '+IntToStr(ListBox2.Items.Count);
end;

procedure TForm1.Button3Click(Sender: TObject);
var
 s:string;
begin
 s:=Edit3.Text;
 with ListBox1 do
 ItemIndex:=Perform(LB_SELECTSTRING, ItemIndex, LongInt(S));
 with ListBox2 do
 ItemIndex:=Perform(LB_SELECTSTRING, ItemIndex, LongInt(S));
 // якщо Multiselect = true, то при пошуку файлу знайдена рядок колір не
// виділяється
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
 SendMessage(ListBox1.Handle,LB_SetHorizontalExtent,1000,0);
 SendMessage(ListBox2.Handle,LB_SetHorizontalExtent,1000,0);
```

```
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
 Listbox2.Sorted:=true;
 ListBox2.Items.BeginUpdate;
 SendMessage(ListBox2.Handle,LB_SetHorizontalExtent,1000,0);
 ListBox2.Items.EndUpdate;
end;
// сортування
procedure TForm1.CheckBox2Click(Sender: TObject);
begin
 Listbox1.Sorted:=true;
 SendMessage(ListBox1.Handle,LB_SetHorizontalExtent,1000,0);
end;
// очистка
procedure TForm1.Button4Click(Sender: TObject);
var
 i:integer;
begin
 for i:=ListBox1.Items.Count-1 downto 0 do
  if ListBox1.Selected[i] then
   begin
    ListBox1.Items.Delete(i);
    ListBox1.Selected[i]:=true;
   end;
 for i:=ListBox2.Items.Count-1 downto 0 do
  if ListBox2.Selected[i] then
   begin
    ListBox2.Items.Delete(i);
    ListBox2.Selected[i]:=true;
   end;
end;
end.
```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI_91. Механізм алгоритму LOKI_91 подібний DES (рисунок 4.3). Блок даних розщеплюється на ліву й праву половини й проходить

16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

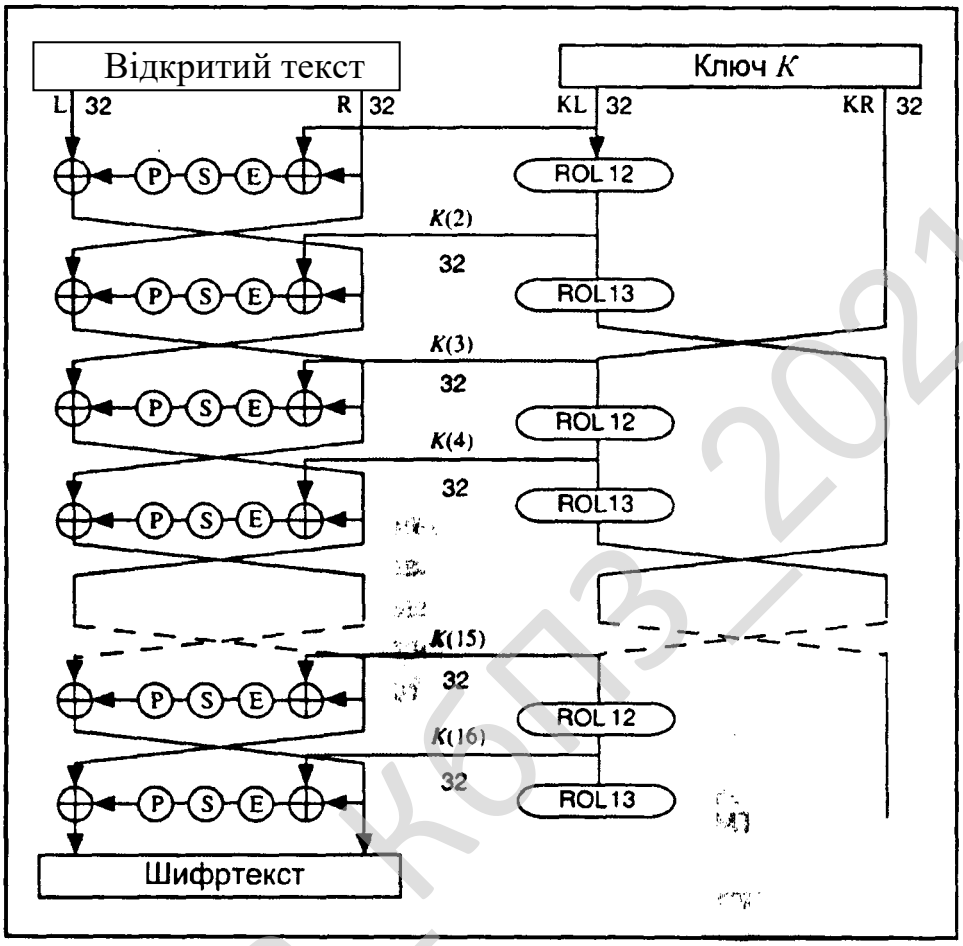

Рисунок 4.3 – Алгоритм LOKI91

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r, а вісім внутрішніх біт утворять номер c. Вихід S-блоку, O, має наступне значення: $O(r,c) = (c + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$.

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 32-бітове число, що піддається операції перестановки, описаній в таблиці 3.

Нарешті, для одержання нової лівої половини виконується операція XOR правої половини з колишньою лівою половиною, а ліва половина стає новою правою половиною.

Після 16 раундів для одержання остаточного шифртексту знову виконується операція XOR над блоком і ключем.

Таблиця 4.3 – Перестановка за допомогою P-блоку

32	24	16	8	31	23	15	7	30	22	14	6	29	21	13	5
2	20	12	4	27	19	11	3	26	18	10	2	25	17	9	1

Підключи генеруються із ключа досить прямолінійно. 64-бітовий ключ розбивається на ліву й праву половини. На кожному раунді підключем служить ліва половина. Далі вона циклічно зрушується вліво на 12 або 13 біт, потім після кожних двох раундів ліва й права половини міняються місцями. Як і в DES, для зашифрування й розшифрування використовується один й той самий алгоритм із деякими змінами у використанні підключів.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку головного вікна можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Функціональних кнопок ПЗ: Сканування; імпортування; Модифікація; Видалення; Додавання; Налаштування; Авторство.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші: Налаштування; Довідка; Журнал роботи.

– Верхнього меню: Файл; Дані; Додатки; Налаштування; Довідка.

– Розділу обрання групи.

– Розділу виведення результату роботи системи.

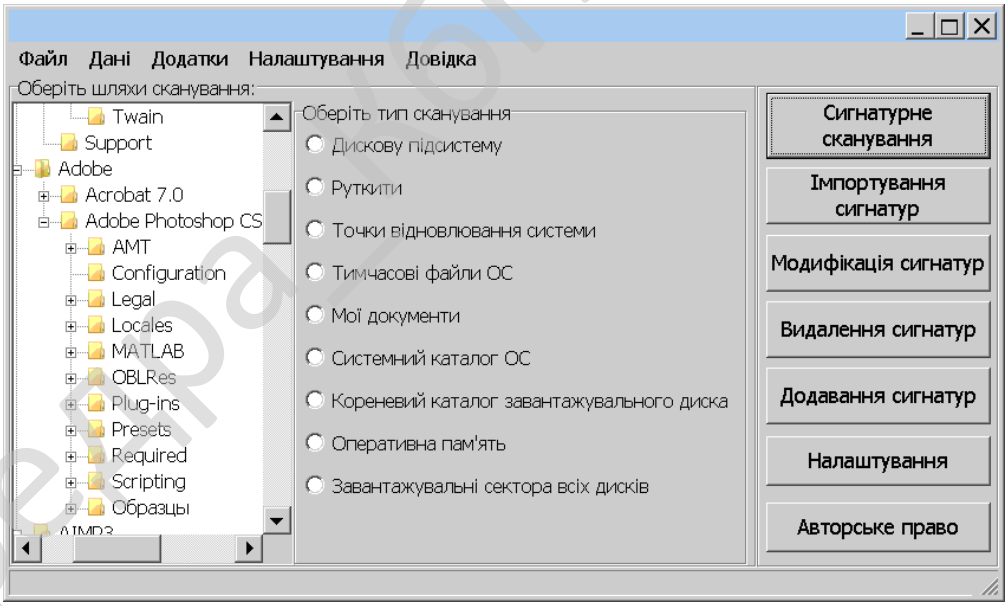

Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Розроблена програма має дуже простий і зрозумілий інтерфейс з

користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

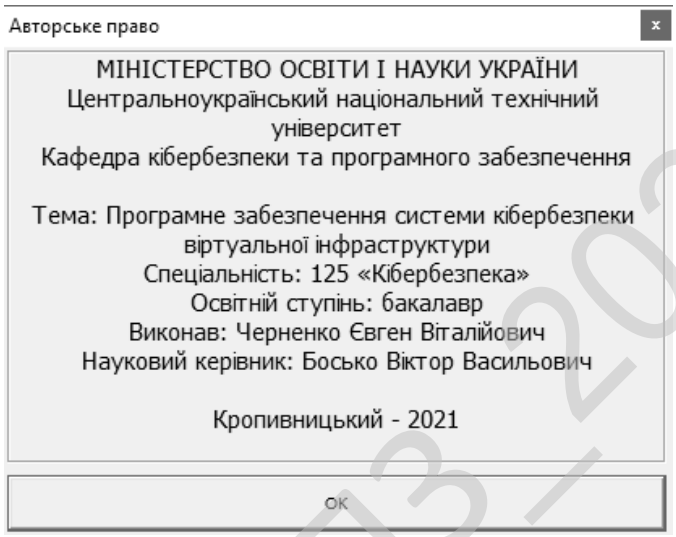

Рисунок 5.2 – Авторське право

Процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Таким чином у результаті вищерозглянутого можна стверджувати що розроблено інтерфейс системи у відповідності з вибраною метою роботи. Система містить максимальний необхідний набір функцій придатних для виконання будь-яких дій для забезпечення повноцінної роботи програми. Далі розглянемо висновки та використані літературні джерела.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи кібербезпеки віртуальної інфраструктури.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем кібербезпеки віртуальної інфраструктури.

– Досліджена система кібербезпеки віртуальної інфраструктури.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки віртуальної інфраструктури.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання віртуальної інфраструктури.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки віртуальної інфраструктури. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне

програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI_91.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Семенов С.Г. Защита данных в компьютеризированных управляющих системах / С.Г. Семенов, В.В. Давыдов, С.Ю. Гавриленко. – LAP Lambert Academic Publishing GmbH & Co. KG (Саарбрюккен, Германия), 2014. – 236 с.

2. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

3. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

4. Смирнов С.А. метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

5. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

6. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов //

Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

7. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. – Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

8. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

9. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

10. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

11. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

12. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

13. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

14. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

15. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

16. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

17. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

18. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных виру сов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

19. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція

«Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

20. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

21. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

22. Смирнов С. А. технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

23. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

24. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

25. Смирнов С. А. Метод управления доступом к облачным телекоммуни-кационным ресурсам для обеспечения защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

26. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

27. Смирнов С. А. Разработка комплекса gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії» (IT & I): зб. тез II міжнар. наук.-практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

28. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

29. Смирнов С. А. gert-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

30. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення

кібернетичної безпеки та захисту інформації: зб. наук. праць II Міжнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

31. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

32. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформатика та системні науки (ІСН-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

33. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. тез наук.-практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

34. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук.-практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

35. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового

забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня – 1 квітня 2016 р. – Х.: НТУ «ХПІ», 2016. – С. 14.

36. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). –Кіровоград: КНТУ, 2016. – С. 182-186.

37. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

38. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна та економічна безпека (INFECO-2016): зб. тез III міжнар. наук. -практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

39. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Сборник тезисов XII международной конференции «Стратегия качества в промышленности и образовании» (г. Варна, Болгария, 30 мая – 02 июня 2016 г.). – Варна: ТУВ, 2016. – С. 581-585.

40. Смирнов С. А. Оценка эффективности метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. С. Коваленко // РадіоЕлектроніка та ІнфоКомунікації: зб. тез першої наук. – техн. конф., м. Київ, 11-16 вересня 2016 р. – К.: НТУУ «КПІ», 2016. – С. 17.

41. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

42. Столлингс В. Современные компьютерные сети / Вильям Столлингс. –СПб.: Питер, 2003. – 778 с.

43. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.

44. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.

45. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.:Вильямс, 2006. – 1103 с.

46. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.

47. Elwalid, D. Mitra, I. Saniee, and I. Widjaja. Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs // Journal of lightwave technology. – 2003. – №21(11), P. 2828-28-38.

48. A.B. Bagula, M. Botha, and A.E Krzesinski. Online Traffic Engineering: The Least Interference Optimization Algorithm // IEEE Communications Society – 2004, P. 1232-1236.

49. Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Evaluating the Impact of Stale Link State on Quality-of-Service Routing // IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.

50. Basabi Chakraborty. Simultaneous Search for Multiple Routes using Genetic Algorithm / IEEE lntemational Conference on Computational oltelligencs for Measurement System and Applications Boston. MA, USA, 14-16, July 2004, P. 77-80/

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки віртуальної інфраструктури.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 129-02 від 14.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи кібербезпеки віртуальної інфраструктури.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи кібербезпеки віртуальної інфраструктури;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпечення.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

– Структурна схема системи – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема алгоритму роботи програми – 2 аркуша.
– Пояснювальна записка – 85 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 7.06.2021 р.

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Босько В.В.

Програмне забезпечення системи кібербезпеки віртуальної інфраструктури

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 43

Літера: РП

Кропивницький – 2021 року

Файл VirtualInfrastructureCybersecurityMain.pas – основна програма

```
unit VirtualInfrastructureCybersecurityMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, ExtCtrls, Menus, ImgList, XPMan,
VirtualInfrastructureCybersecurityKernel,
VirtualInfrastructureCybersecurityTypes, ShellAPI, ShlObj,
  AppEvnts, OneHist, langs, jpeg;

  const
  WM_NOTIFYTRAYICON = WM_USER + 1;
  WM_MINERESTORE = WM_USER    + $877;

  type
  TIconType = (itSmall, itLarge);

  type
  NotifyIconData_50 = record
   cbSize: DWORD;
   Wnd: HWND;
   uID: UINT;
   uFlags: UINT;
   uCallbackMessage: UINT;
   hIcon: HICON;
   szTip: array[0..MAXCHAR] of AnsiChar;
   dwState: DWORD;
   dwStateMask: DWORD;
   szInfo: array[0..MAXBYTE] of AnsiChar;
   uTimeout: UINT; // union with uVersion: UINT;
   szInfoTitle: array[0..63] of AnsiChar;
   dwInfoFlags: DWORD;
  end;

  const
  NIF_INFO      =        $00000010;
  NIIF_NONE     =        $00000000;
  NIIF_INFO     =        $00000001;
  NIIF_WARNING  =        $00000002;
  NIIF_ERROR    =        $00000003;

  type
  TBalloonTimeout = 10..30;
  TBalloonIconType = (bitNone,
                      bitInfo,
                      bitWarning,
                      bitError);

type
  TMainForm = class(TForm)
    MainPages: TPageControl;
    VirtualInfrastructureCybersecurityScanPathesTab: TTabSheet;
    VirtualInfrastructureCybersecurityScanningTab: TTabSheet;
    ReportTab: TTabSheet;
    BottomPanel: TPanel;
    VirtualInfrastructureCybersecurityScanBTN: TButton;
    SaveBTN: TButton;
    PathList: TListView;
    Bevel1: TBevel;
    VirtualInfrastructureCybersecurityScanList: TListView;
    ReportMemo: TMemo;
    ImageList: TImageList;
    DrivesImg: TImageList;
    PathMenu: TPopupMenu;
```

```
    AddFolder: TMenuItem;
    DeletePath: TMenuItem;
    N1: TMenuItem;
    Reftesh: TMenuItem;
    SaveDialog: TSaveDialog;
    XPManifest: TXPManifest;
    Bevel4: TBevel;
    DelMenu: TPopupMenu;
    Del: TMenuItem;
    TrayMenu: TPopupMenu;

mnuShowAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner:
TMenuItem;

mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner:
TMenuItem;
    N2: TMenuItem;
    mnVirtualInfrastructureCybersecurityOptions: TMenuItem;
    N4: TMenuItem;
    mnuHelp: TMenuItem;
    mnuAbout: TMenuItem;
    N7: TMenuItem;
    mnuExit: TMenuItem;
    Image1: TImage;
    TopPn: TPanel;
    Bevel3: TBevel;
    Image2: TImage;
    RightPanel: TPanel;
    ExitBTN: TButton;
    TopRightPanel: TPanel;
    Image3: TImage;
    VersionLabel: TLabel;
    AboutBTN: TLabel;
    DelAll: TMenuItem;
    ApplicationEvents: TApplicationEvents;
    ProgressBar: TProgressBar;
    VirtualInfrastructureCybersecurityScanTopBtn: TLabel;
    VirtualInfrastructureCybersecurityScanMenu: TPopupMenu;
    mnuSelVirtualInfrastructureCybersecurityScanPath: TMenuItem;
    mnuShowReport: TMenuItem;
    N12: TMenuItem;
    OptionTopBtn: TLabel;
    PCTopBtn: TLabel;
    mnuAntiVirus_with_signatureProcessControl: TMenuItem;
    N19: TMenuItem;
    mnuPCShow: TMenuItem;
    N21: TMenuItem;
    mnuPCRun: TMenuItem;
    mnuPCPause: TMenuItem;
    mnuPCStop: TMenuItem;
    mnuVirtualInfrastructureCybersecurityScanStart: TMenuItem;
    mnuStopVirtualInfrastructureCybersecurityScan: TMenuItem;
    N13: TMenuItem;
    mnuSaveReport: TMenuItem;
    N26: TMenuItem;
    mnuGoToTray: TMenuItem;
    SOURCESTRING: TListBox;
    LabelPanel: TPanel;
    VirtualInfrastructureCybersecurityScanFile: TLabel;
    procedure DelAllClick(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure ExitBTNClick(Sender: TObject);
    procedure VirtualInfrastructureCybersecurityScanListDblClick(Sender:
TObject);
    procedure VirtualInfrastructureCybersecurityScanBTNClick(Sender: TObject);
    procedure InitVirus_VirtualInfrastructureCybersecurityScannerKernel;
    Procedure StartVirtualInfrastructureCybersecurityScan(Parametr: String);
    procedure SaveBTNClick(Sender: TObject);
    procedure DeletePathClick(Sender: TObject);
```

```
    procedure RefteshClick(Sender: TObject);
    procedure AddFolderClick(Sender: TObject);
    function CreateDrivesList(ListView: TListView): boolean;
    procedure AboutBTNClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure HelpBTNClick(Sender: TObject);
    procedure DelMenuPopup(Sender: TObject);
    procedure DelClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormHide(Sender: TObject);
    procedure
mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerCl
ick(Sender: TObject);
    procedure
mnuShowAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerCl
ick(Sender: TObject);
    procedure mnuExitClick(Sender: TObject);
    procedure mnVirtualInfrastructureCybersecurityOptionsClick(Sender: TObject);
    procedure mnuHelpClick(Sender: TObject);
    procedure mnuAboutClick(Sender: TObject);
    procedure ApplicationEventsMinimize(Sender: TObject);
    procedure AppMinimize(Sender: TObject);
    procedure FormPaint(Sender: TObject);
    procedure VirtualInfrastructureCybersecurityScanListCustomDrawItem(Sender:
TCustomListView;
      Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);
    function BalloonTrayIcon(const Window: HWND; const IconID: Byte; const
Timeout: TBalloonTimeout; const BalloonText, BalloonTitle: String; const
BalloonIconType: TBalloonIconType): Boolean;
    procedure VirtualInfrastructureCybersecurityScanTopBtnClick(Sender:
TObject);
    procedure mnuShowReportClick(Sender: TObject);
    procedure mnuSelVirtualInfrastructureCybersecurityScanPathClick(Sender:
TObject);
    procedure PCTopBtnClick(Sender: TObject);
    procedure OptionTopBtnClick(Sender: TObject);
    procedure mnuGoToTrayClick(Sender: TObject);
    procedure mnuPCShowClick(Sender: TObject);
    procedure mnuPCRunClick(Sender: TObject);
    procedure mnuPCPauseClick(Sender: TObject);
    procedure mnuPCStopClick(Sender: TObject);
    procedure TrayMenuPopup(Sender: TObject);
    procedure VirtualInfrastructureCybersecurityScanMenuPopup(Sender: TObject);
    procedure mnuVirtualInfrastructureCybersecurityScanStartClick(Sender:
TObject);
    procedure mnuStopVirtualInfrastructureCybersecurityScanClick(Sender:
TObject);
    procedure mnuSaveReportClick(Sender: TObject);
    procedure CopyRightLabelClick(Sender: TObject);
    Procedure CreateTray;
  protected
    procedure MineRestore(var Msg: TMessage); message WM_MINERESTORE;
    procedure SendVirtualInfrastructureCybersecurityScanning(var Msg: TMessage);
message WM_COPYDATA;
  private
    Procedure WMSysCommand(var message: TWMSysCommand); message WM_SysCommand;
    procedure WMTRAYICONNOTIFY(var Msg: TMessage); message WM_NOTIFYTRAYICON;
    { Private declarations }
  public
    FileCN        : Integer;
    FileInfected  : Integer;
    FileIgnored   : Integer;
    FileDVC       : integer;

    MonFileCN        : Integer;
    MonFileInfected  : Integer;
```

```
    Path          : TStringList;
    DeActiveTray  : Boolean;

//*****************************************************************************//

        AntiVirus_with_signatureMonitor      : String;
        AntiVirus_with_signatureInit         : String;
        LoadAPI          : String;
        LoadDB           : String;
        CreateDrvList    : String;
        OptFileNotFnd    : String;
        LoadOptFile      : String;
        InitProcedures   : String;
        initShield       : String;
        ErrorInit        : String;
        LogBevel         : String;
        DBKnowledge      : String;
        SCNOBJ           : String;
        VirtualInfrastructureCybersecurityScanExecute     : String;
        VirtualInfrastructureCybersecurityScanEnd         : String;
        PrepareToVirtualInfrastructureCybersecurityScan   : String;
        FileIgnor        : String;
        FileIfect        : String;
        FileVirtualInfrastructureCybersecurityScanned     : String;
        DataVirtualInfrastructureCybersecurityScanned     : String;
        IGNORED          : String;
        SKIPBYSIZE       : String;
        INFECTED         : String;
        STOPB            : String;
        RETURNB          : String;
        VIRTUALINFRASTRUCTURECYBERSECURITYSCANB          : String;
        SCNFILE          : String;
        FileDel          : String;
        FileNotDel       : String;
        PATHNOSEL        : String;
        SysMenu          : String;

NfoAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner   :
String;
        NfoAntiVirus_with_signatureKernel    : String;
        NfoAntiVirus_with_signatureBuild     : String;
        DelDialog        : String;
        DelAllDialog     : String;
        DelError         : String;
        HelpNOFound      : String;
        avShieldMes      : String;
        avError          : String;
        DelResult        : String;
        AllInfected      : String;
        DeleteInfected   : String;
        SkippedInfected  : String;
        AntiVirus_with_signatureCloseDlg     : String;
        AllreadyInVirtualInfrastructureCybersecurityScan  : String;
        ProcControlSt    : String;
        ErrorKillProc    : String;
        PCActive         : String;
        PCPaused         : String;
        PCStoped         : String;
        PCInit           : String;
        PCPause          : String;
        PCStop           : String;
        PCRestore        : String;
        LASTDBDATA       : String;
        DATABASEdate     : String;
        BASELOADED       : String;
        DBerrorI1        : String;
        DBerrorI2        : String;
        DBerrorI3        : String;
```

```
        MLoad           : String;
        MunLoad         : String;

  end;

//****************************************************************************//
// Створення головної форми
resourcestring
  Return          = #13#10;
  AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerCapt  =
'Антивірусний захист операційної системи від шкідливих програм';
  AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerVS    =
'';
var
  MainForm      : TMainForm;
  inVirtualInfrastructureCybersecurityScan        : Boolean = False;
  NeedToReturn  : Boolean = False;
  FirstRun      : Boolean = True;
  P             : TPoint;
  MayClose      : boolean=false;
implementation

uses uSelInfo, VirtualInfrastructureCybersecurityOptions,
VirtualInfrastructureCybersecurityAddPath,
VirtualInfrastructureCybersecurityAbout, Math, uMessage, uHideForm,
  VirtualInfrastructureCybersecurityMonitor,
VirtualInfrastructureCybersecurityInfectedAction, uPluginInfo;
{$R *.dfm}

//****************************************************************************//

Procedure TMainForm.WMSysCommand(var message: TWMSysCommand);
begin
  If message.CmdType = SC_MINIMIZE then
mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.C
lick
  Else Inherited;
End;

//****************************************************************************//

procedure TMainForm.SendVirtualInfrastructureCybersecurityScanning;
var
  pcd: PCopyDataStruct;
begin
  pcd := PCopyDataStruct(Msg.LParam);
  if not inVirtualInfrastructureCybersecurityScan then
  begin
    StartVirtualInfrastructureCybersecurityScan(PChar(pcd.lpData));
  end
  else begin

MessageDlg(AllreadyInVirtualInfrastructureCybersecurityScan,mtError,[mbOK],0);
  end;
end;

procedure TMainForm.MineRestore(var Msg: TMessage);
begin
  if (Msg.Msg = WM_MINERESTORE) then
  begin

mnuShowAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.C
lick;
  end;
end;

//****************************************************************************//
```

```
function TMainForm.BalloonTrayIcon(const Window: HWND; const IconID: Byte; const
Timeout: TBalloonTimeout; const BalloonText, BalloonTitle: String; const
BalloonIconType: TBalloonIconType): Boolean;
  const
  aBalloonIconTypes : array[TBalloonIconType] of
                      Byte = (NIIF_NONE, NIIF_INFO, NIIF_WARNING, NIIF_ERROR);
  var
  NID_50 : NotifyIconData_50;
begin
  if Not OptionsForm.SHOWBALOONHINT.Checked then Exit;
  FillChar(NID_50, SizeOf(NotifyIconData_50), 0);
  with NID_50 do begin
    cbSize := SizeOf(NotifyIconData_50);
    Wnd := Window;
    uID := IconID;
    uFlags := NIF_INFO;
    StrPCopy(szInfo, BalloonText);
    uTimeout := Timeout * 1000;
    StrPCopy(szInfoTitle, BalloonTitle);
    dwInfoFlags := aBalloonIconTypes[BalloonIconType];
  end;
  Result := Shell_NotifyIcon(NIM_MODIFY, @NID_50);
end;

procedure TMainForm.WMTRAYICONNOTIFY(var Msg: TMessage);
begin
  case Msg.LParam of
    WM_LBUTTONUP:
      begin
      if Not DeActiveTray then
        begin
          MayClose := False;
          GetCursorPos(p);
          MayClose:= false;
          DeActiveTray := False;
          showwindow(Application.handle, SW_SHOW);
          showwindow(MainForm.handle, SW_SHOW);
          Application.Restore;
        end
        else
        begin
          SetForegroundWindow(HideForm.Handle);
        end;
      end;
    WM_RBUTTONUP:
      begin
        if Not DeActiveTray then
        begin
          GetCursorPos(p);
          TrayMenu.Popup(P.X, P.Y);
        end;
      end;
  end;
end;

Procedure TMainForm.CreateTray;
  var
  tray: TNotifyIconData;
begin
  with tray do
  begin
    cbSize := SizeOf(TNotifyIconData);
    Wnd := MainForm.Handle;
    uID := 1;
    uFlags := NIF_ICON or NIF_MESSAGE or NIF_TIP;
    uCallBackMessage := WM_NOTIFYTRAYICON;
    hIcon :=  Application.Icon.Handle;
    szTip := 'AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner';
```

```
  end;
  Shell_NotifyIcon(NIM_ADD, Addr(tray));
end;

Procedure DestroyTray;
  var
  tray: TNotifyIconData;
begin
  with tray do
  begin
    cbSize := SizeOf(TNotifyIconData);
    Wnd := MainForm.Handle;
    uID := 1;
  end;
  Shell_NotifyIcon(NIM_DELETE, Addr(tray));
end;

//*Функція визначення шляху***********************************//

Function GetShortPathBC(lPath:string): string;
  var
  D,F,P: String;
  i    : integer;
begin
  D := lPath[1]+':\';
  F := ExtractFileName(lPath);
  ShowMessage(D+'..'+F);
end;

Function GETParam(Str: String): String;
  var
  TMP,Str1,Str2 : String;
  PS: integer;
begin
  Result := '';
  TMP := STR;
  if TMP <> '' then
  if pos('=',TMP) <> 0 then
  begin
    ps := pos('=',TMP);
    Str1 := Copy(TMP,0,ps-1);
    Str2 := Copy(TMP,ps+1,length(Tmp));
    Result := Str2;
  end;
end;

Function GETParamName(Str: String): String;
  var
  TMP,Str1,Str2 : String;
  PS: integer;
begin
  Result := '';
  TMP := STR;
  if TMP <> '' then
  if pos('=',TMP) <> 0 then
  begin
    ps := pos('=',TMP);
    Str1 := Copy(TMP,0,ps-1);
    Str2 := Copy(TMP,ps+1,length(Tmp));
    Result := Str1;
  end;
end;

//***Функція завантаження опцій ***//

Procedure LoadOptions;
  var
  i: integer;
begin
```

```
  LoadConfig_;
  OptionsForm.ModulesLOAD.Checked     := OPT_MODULES_LOAD;
  OptionsForm.DBPATH.Text             := OPT_DB_DIR;
  OptionsForm.MODULESPATH.Text        := OPT_MODULE_DIR;
  OptionsForm.USESHIELD.Checked       := OPT_USE_SHIELD;
  OptionsForm.SHIELDSILENT.Checked    := OPT_SILENT_SHIELD_MODE;
  OptionsForm.SCNSUBDIR.Checked       :=
OPT_VIRTUALINFRASTRUCTURECYBERSECURITYSCAN_SUBDIR;
  OptionsForm.SCNHEX.Checked          := OPT_USE_HEX_MODE;
  OptionsForm.SCNCRC.Checked          := OPT_USE_CRC_MODE;
  OptionsForm.SCNBIT.Checked          := OPT_USE_BYTE_MODE;

  OptionsForm.SCNHEXINPOS.Checked     := OPT_USE_HEX_INPOS;
  OptionsForm.DisplayScnFiles.Checked :=
OPT_SEND_VIRTUALINFRASTRUCTURECYBERSECURITYSCAN_FILE;

  OptionsForm.PathList.Clear;
  OptionsForm.ExtList.Clear;
  for i := 0 to AntiVirus_with_signatureConfig.Count-1 do begin

    if GETParamName(AntiVirus_with_signatureConfig[i]) = 'EXT' then
      with OptionsForm.ExtList.Items.Add do begin
        Caption := GetParam(AntiVirus_with_signatureConfig[i]);
        ImageIndex := 3;
      end;
      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'SHOWBALOONHINT' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.SHOWBALOONHINT.Checked := False else
      OptionsForm.SHOWBALOONHINT.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'PROCCONTROLAUTOMODE'
then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.PCAutoLoad.Checked := False else
      OptionsForm.PCAutoLoad.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'PROCCONTROLAUTOKILL'
then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.PCAutoKill.Checked := False else
      OptionsForm.PCAutoKill.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) =
'PROCCONTROLAUTOACTION' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.PCAutoAction.Checked := False else
      OptionsForm.PCAutoAction.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) =
'PROCCONTROLDELINFECT' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.PCDelInfect.Checked := False else
      OptionsForm.PCDelInfect.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) =
'PROCCONTROLSKIPINFECT' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
OptionsForm.PCSkipInfect.Checked := False else
      OptionsForm.PCSkipInfect.Checked := True;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'HIDETIP' then begin
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'OFF' then
HideForm.ShowHideTip.Checked := False else
      HideForm.ShowHideTip.Checked := True;
      end;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'PATH' then begin
      with OptionsForm.PathList.Items.Add do begin
        Caption := GetParam(AntiVirus_with_signatureConfig[i]);
```

```
        if DirectoryExists(Caption) then ImageIndex := 4 else ImageIndex := 5;
      end;
      end;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'AUTOSAVEREPORT' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'ON' then
OptionsForm.AutoSaveReport.Checked := true else
      OptionsForm.AutoSaveReport.Checked := False;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'REGISTERSYSMENU'
then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'ON' then
OptionsForm.RegisterSysMenu.Checked := true else
      OptionsForm.RegisterSysMenu.Checked := False;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'AUTORUN' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'ON' then
OptionsForm.AUTORUN.Checked := true else
      OptionsForm.AUTORUN.Checked := False;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'AUTOHIDE' then
      if GetParam(AntiVirus_with_signatureConfig[i]) = 'ON' then
OptionsForm.AUTOHIDE.Checked := true else
      OptionsForm.AUTOHIDE.Checked := False;

      if GETParamName(AntiVirus_with_signatureConfig[i]) = 'AUTOSAVEREPORTTO'
then OptionsForm.ReportSavePath.Text :=
GETParam(AntiVirus_with_signatureConfig[i]);
  end;
end;

function GetHDDSerial(ADisk : char): dword;
  var
  SerialNum : dword;
  a, b : dword;
  VolumeName : array [0..255] of char;
begin
  Result := 0;
  if GetVolumeInformation(PChar(ADisk + ':\'), VolumeName, SizeOf(VolumeName),
  @SerialNum, a, b, nil, 0) then
    Result := SerialNum;
end;

function TMainForm.CreateDrivesList(ListView: TListView): boolean;
  var
  Bufer : array[0..1024] of char;
  RealLen, i : integer;
  S : string;
begin
  ListView.Clear;
  RealLen := GetLogicalDriveStrings(SizeOf(Bufer),Bufer);
  i := 0; S := '';
  while i < RealLen do begin
    if Bufer[i] <> #0 then begin
      S := S + Bufer[i];
      inc(i);
    end else begin
      inc(i);
      with ListView.Items.Add do begin
        Caption := S;
        if GetDriveType(PChar(S)) = DRIVE_RAMDISK then ImageIndex := 3;
        if GetDriveType(PChar(S)) = DRIVE_FIXED then ImageIndex := 3;
        if GetDriveType(PChar(S)) = DRIVE_REMOTE then ImageIndex := 0;
        if GetDriveType(PChar(S)) = DRIVE_CDROM then ImageIndex := 1;
        if GetDriveType(PChar(S)) = DRIVE_REMOVABLE then ImageIndex := 2;
      end;
      S := '';
    end;
  end;
```

```
  For i := 0 to OptionsForm.PathList.Items.Count-1  do begin
    with ListView.Items.Add do begin
      Caption := OptionsForm.PathList.Items[i].Caption;
      ImageIndex := OptionsForm.PathList.Items.Item[i].ImageIndex;
    end;
  end;
  Result := ListView.items.Count > 0;
end;

procedure OnAddToLogStr(LogString: String; ID: integer);
  var
  TMP : String;
begin
  with MainForm.VirtualInfrastructureCybersecurityScanList.Items.Add do begin
    if ID = -1 then
      Caption := LogString
    else begin
      Caption := FormatDateTime('[hh:mm:ss]',now) + ' ' + LogString;
      MainForm.ReportMemo.Lines.Add(Caption);
      if ID = 2 then begin
        TMP := LogString;
        system.Delete(Tmp,1,pos(']',Tmp)+1);
        SubItems.Add(TMP);
      end;
      ImageIndex := ID;
    end;
    ImageIndex := ID;
  end;
  SendMessage(MainForm.VirtualInfrastructureCybersecurityScanList.Handle,
WM_VSCROLL, SB_BOTTOM, 0);
end;

procedure AddToMonLogStr(LogString: String; ID: integer);
  var
  TMP : String;
begin
{ }
end;

//***Функція вибору параметрів сканування на віруси***//

procedure OnVirtualInfrastructureCybersecurityScanComplete;
  var
  VirtualInfrastructureCybersecurityScanEndBalloonText: String;
  i: integer;
begin
  MainForm.ProgressBar.Max := 1;
  MainForm.ProgressBar.Position := MainForm.ProgressBar.Max;
  MainForm.VirtualInfrastructureCybersecurityScanBTN.Caption :=
MainForm.RETURNB;
  NeedToReturn := True;
  inVirtualInfrastructureCybersecurityScan := False;
  MainForm.Path.Clear;

  for i := 0 to MainForm.PathList.Items.Count-1 do
    MainForm.PathList.Items.Item[i].Checked := false;

  MessageBeep(MB_ICONASTERISK);
  MainForm.SaveBTN.Enabled := true;
  MainForm.VirtualInfrastructureCybersecurityScanFile.caption  :=
MainForm.VirtualInfrastructureCybersecurityScanEnd;
  OnAddToLogStr('',-1);
  OnAddToLogStr(MainForm.VirtualInfrastructureCybersecurityScanEnd,0);
  OnAddToLogStr('',-1);

OnAddToLogStr(MainForm.FileVirtualInfrastructureCybersecurityScanned+inttostr(Ma
inForm.FileCN),0);
  OnAddToLogStr(MainForm.FileIgnor+inttostr(MainForm.FileIgnored),0);
```

```
  OnAddToLogStr(MainForm.FileIfect+inttostr(MainForm.FileInfected),0);

OnAddToLogStr(MainForm.DataVirtualInfrastructureCybersecurityScanned+Format('%.2
f',[VirtualInfrastructureCybersecurityScannedDataSize / 1024 / 1024])+' Mb',0);
  MainForm.ReportMemo.Lines.Add(MainForm.LogBevel);
  if OptionsForm.AutoSaveReport.Checked then begin
    MainForm.ReportMemo.Lines.SaveToFile(OptionsForm.ReportSavePath.Text);
  end;

  VirtualInfrastructureCybersecurityScanEndBalloonText :=
MainForm.VirtualInfrastructureCybersecurityScanEnd + ':' + Return + Return
                        +' >>
'+MainForm.FileVirtualInfrastructureCybersecurityScanned+inttostr(MainForm.FileC
N) + Return
                        +' >> '+MainForm.FileIgnor+inttostr(MainForm.FileIgnored)
+ Return
                        +' >>
'+MainForm.FileIfect+inttostr(MainForm.FileInfected) + Return
                        +' >>
'+MainForm.DataVirtualInfrastructureCybersecurityScanned+Format('%.2f',[VirtualI
nfrastructureCybersecurityScannedDataSize / 1024 / 1024])+' Mb';

  MainForm.BalloonTrayIcon(MainForm.Handle
,1,10,VirtualInfrastructureCybersecurityScanEndBalloonText,'AntiVirus_with_signa
ture Virus_VirtualInfrastructureCybersecurityScanner',bitInfo);
end;

//***Функція початку сканування***//

Procedure OnVirtualInfrastructureCybersecurityScanStart;
  var
  i: integer;
begin
  MainForm.FileDVC := 0;
  MainForm.ProgressBar.Position := 0;
  MainForm.ProgressBar.Max := 0;

  ClearExtList;
  for i := 0 to OptionsForm.ExtList.Items.Count-1 do begin
    AddToExtList(ExtractFileExt(OptionsForm.ExtList.Items.Item[i].Caption));
  end;

  MainForm.VirtualInfrastructureCybersecurityScanBTN.Caption := MainForm.STOPB;
  MainForm.SaveBTN.Enabled := False;
  MainForm.VirtualInfrastructureCybersecurityScanList.Clear;
  MainForm.VirtualInfrastructureCybersecurityScanningTab.Show;
  MainForm.FileCN       := 0;
  MainForm.FileInfected := 0;
  MainForm.FileIgnored  := 0;
  inVirtualInfrastructureCybersecurityScan := True;
  NeedToReturn := False;
  OnAddToLogStr(MainForm.VirtualInfrastructureCybersecurityScanExecute,0);
  if
AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.AvAction
= TVirtualInfrastructureCybersecurityScanDir then
  else

OnAddToLogStr(MainForm.SCNOBJ+AntiVirus_with_signatureVirus_VirtualInfrastructur
eCybersecurityScanner.FileName,0);
  OnAddToLogStr('',-1);
  MainForm.BalloonTrayIcon(MainForm.Handle
,1,10,MainForm.VirtualInfrastructureCybersecurityScanExecute,'AntiVirus_with_sig
nature Virus_VirtualInfrastructureCybersecurityScanner',bitInfo);

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.Resume;
end;

//***Функція підключення ядра антивіруса***//
```

```
Procedure AntiVirus_with_signatureKernelMessageAPI(MES: Integer; const Pr_0:
Integer = 0; Pr_1: String = ''; Pr_2: String = '');
begin

  if MES = MES_NONE then Exit;

  if mes = MES_LOCKINPUT then
  begin
    MainForm.ProgressBar.Enabled := False;
    MainForm.VirtualInfrastructureCybersecurityScanBTN.Enabled := False;
  end;

  if mes = MES_UNLOCKINPUT then
  begin
    MainForm.ProgressBar.Position := 0;
    MainForm.ProgressBar.Enabled := True;
    MainForm.VirtualInfrastructureCybersecurityScanBTN.Enabled := True;
  end;

  if MES = MES_VIRTUALINFRASTRUCTURECYBERSECURITYSCANMAXPROGRESS then begin
    MainForm.FileDVC := mainForm.FileCN;
    MainForm.ProgressBar.Max := Pr_0-MainForm.FileDVC;
  end;

  if MES = MES_PREPARINGTOVIRTUALINFRASTRUCTURECYBERSECURITYSCAN then
MainForm.VirtualInfrastructureCybersecurityScanFile.Caption :=
MainForm.PrepareToVirtualInfrastructureCybersecurityScan;

  if mes = MES_INITKERNEL then
OnAddToLogStr(MainForm.AntiVirus_with_signatureInit,0);

  if mes = MES_INITAPI then OnAddToLogStr(MainForm.LoadAPI,0);

  if mes = MES_LOADBASES then OnAddToLogStr(MainForm.LoadDB,0);

  if mes = MES_LOADCONFIG then OnAddToLogStr(MainForm.LoadOptFile,0);

  if mes = MES_INITSHIELD then OnAddToLogStr(MainForm.initShield,0);

  if mes = MES_ERRORONINIT then OnAddToLogStr(MainForm.ErrorInit,2);

  if MES = MES_LOADDBDATE then begin
    MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.BASELOADED+ ExtractFileName(Pr_1)+'
('+MainForm.DATABASEdate+_ConvertDate(Pr_2)+')');
  end;

  if MES = MES_ERROR then begin
    MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.avError);
  end;

  if MES = MES_ONVIRTUALINFRASTRUCTURECYBERSECURITYSCANEXECUTE then
    OnVirtualInfrastructureCybersecurityScanStart;

  if MES = MES_ONVIRTUALINFRASTRUCTURECYBERSECURITYSCANCOMPLETE then
    OnVirtualInfrastructureCybersecurityScanComplete;

  if MES = MES_ONPROGRESS then begin
    if MainForm.ProgressBar.Enabled then begin
      MainForm.FileCN := MainForm.FileCN + 1;
      if MainForm.ProgressBar.Max > 0 then
        MainForm.ProgressBar.Position := MainForm.FileCN-MainForm.FileDVC;
      MainForm.VirtualInfrastructureCybersecurityScanFile.caption  :=
'['+inttostr(MainForm.FileCN)+'] '+ExtractFileName(Pr_1);
    end
    else
      MainForm.VirtualInfrastructureCybersecurityScanFile.caption  :=
ExtractFileName(Pr_1);
```

```
    if OPT_SEND_VIRTUALINFRASTRUCTURECYBERSECURITYSCAN_FILE then
MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+MainForm.SCNFILE
+ Pr_1);
  end;

  if MES = MES_ONVIRFOUND then begin
    OnAddToLogStr('['+MainForm.INFECTED+' - '+Pr_2+'] '+Pr_1,2);
    MainForm.FileInfected := MainForm.FileInfected + 1;
    MainForm.BalloonTrayIcon(MainForm.Handle ,1,10,Pr_1 ,'['+MainForm.INFECTED+'
- '+Pr_2+'] ',bitError);
  end;

  if MES = MES_ONREADERROR then begin
    OnAddToLogStr('['+MainForm.IGNORED+'] '+Pr_1,1);
    MainForm.FileIgnored  := MainForm.FileIgnored + 1;
  end;

  if MES = MES_SKIPBYSIZE then begin
    OnAddToLogStr('['+MainForm.SKIPBYSIZE+'] '+Pr_1,1);
    MainForm.FileIgnored  := MainForm.FileIgnored + 1;
  end;

  if MES = MES_ADDTOLOG then begin
    OnAddToLogStr(Pr_1,Pr_0);
  end;

  if MES = MES_SHIELD_INFECT then begin
    MessageFrm.Caption := 'avShield Messsage';
    MessageFrm.InformationLabel.Caption := 'avShield Messsage';
    MessageFrm.InfoLabel.Caption := 'Warning!';
    MessageFrm.Memo1.Text := MainForm.avShieldMes;
  end;

end;

//***Функція ініціалізація ядра антивіруса***//

procedure TMainForm.InitVirus_VirtualInfrastructureCybersecurityScannerKernel;
  var
  i:integer;
begin

//******//
        AntiVirus_with_signatureMonitor      := SOURCESTRING.Items[0];
        AntiVirus_with_signatureInit         := SOURCESTRING.Items[1];
        LoadAPI          := SOURCESTRING.Items[2];
        LoadDB           := SOURCESTRING.Items[3];
        CreateDrvList    := SOURCESTRING.Items[4];
        OptFileNotFnd    := SOURCESTRING.Items[5];
        LoadOptFile      := SOURCESTRING.Items[6];
        InitProcedures   := SOURCESTRING.Items[7];
        initShield       := SOURCESTRING.Items[8];
        ErrorInit        := SOURCESTRING.Items[9];
        LogBevel         := SOURCESTRING.Items[10];
        DBKnowledge      := SOURCESTRING.Items[11];
        SCNOBJ           := SOURCESTRING.Items[12];
        VirtualInfrastructureCybersecurityScanExecute     :=
SOURCESTRING.Items[13];
        VirtualInfrastructureCybersecurityScanEnd         :=
SOURCESTRING.Items[14];
        PrepareToVirtualInfrastructureCybersecurityScan   :=
SOURCESTRING.Items[15];
        FileIgnor        := SOURCESTRING.Items[16];
        FileIfect        := SOURCESTRING.Items[17];
        FileVirtualInfrastructureCybersecurityScanned     :=
SOURCESTRING.Items[18];
        DataVirtualInfrastructureCybersecurityScanned     :=
SOURCESTRING.Items[19];
        IGNORED          := SOURCESTRING.Items[20];
```

```
        SKIPBYSIZE      := SOURCESTRING.Items[21];
        INFECTED        := SOURCESTRING.Items[22];
        STOPB           := SOURCESTRING.Items[23];
        RETURNB         := SOURCESTRING.Items[24];
        VIRTUALINFRASTRUCTURECYBERSECURITYSCANB          :=
SOURCESTRING.Items[25];
        SCNFILE         := SOURCESTRING.Items[26];
        FileDel         := SOURCESTRING.Items[27];
        FileNotDel      := SOURCESTRING.Items[28];
        PATHNOSEL       := SOURCESTRING.Items[29];
        SysMenu         := SOURCESTRING.Items[30];

NfoAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner   :=
SOURCESTRING.Items[31];
        NfoAntiVirus_with_signatureKernel    := SOURCESTRING.Items[32];
        NfoAntiVirus_with_signatureBuild     := SOURCESTRING.Items[33];
        DelDialog       := SOURCESTRING.Items[34];
        DelAllDialog    := SOURCESTRING.Items[35];
        DelError        := SOURCESTRING.Items[36];
        HelpNOFound     := SOURCESTRING.Items[37];
        avShieldMes     := SOURCESTRING.Items[38];
        avError         := SOURCESTRING.Items[39];
        DelResult               := SOURCESTRING.Items[40];
        AllInfected     := SOURCESTRING.Items[41];
        DeleteInfected  := SOURCESTRING.Items[42];
        SkippedInfected := SOURCESTRING.Items[43];
        AntiVirus_with_signatureCloseDlg     := SOURCESTRING.Items[44];
        AllreadyInVirtualInfrastructureCybersecurityScan  :=
SOURCESTRING.Items[45];
        ProcControlSt   := SOURCESTRING.Items[46];
        ErrorKillProc   := SOURCESTRING.Items[47];
        PCActive        := SOURCESTRING.Items[48];
        PCPaused        := SOURCESTRING.Items[49];
        PCStoped        := SOURCESTRING.Items[50];
        PCInit          := SOURCESTRING.Items[51];
        PCPause         := SOURCESTRING.Items[52];
        PCStop          := SOURCESTRING.Items[53];
        PCRestore       := SOURCESTRING.Items[54];
        LASTDBDATA      := SOURCESTRING.Items[55];
        DATABASEdate    := SOURCESTRING.Items[56];
        BASELOADED      := SOURCESTRING.Items[57];
        DBerrorI1       := SOURCESTRING.Items[58];
        DBerrorI2       := SOURCESTRING.Items[59];
        DBerrorI3       := SOURCESTRING.Items[60];

        MLoad           := SOURCESTRING.Items[61];
        MunLoad         := SOURCESTRING.Items[62];

        InitKernel(AntiVirus_with_signatureKernelMessageAPI);
        LoadOptions;

//***Функція створення списку дисків***//

  CreateDrivesList(PathList);

  for i := 0 to GetPluginAPICount do
    with OptionsForm.APIList.Items.Add do
      begin
        Caption := GetPluginAPIName(i) + '
('+ExtractFileName(GetPluginAPIPath(i))+')';
        SubItems.Add(GetPluginAPIAutor(i));
        SubItems.Add(GetPluginAPIInfo(i));
        SubItems.Add(GetPluginAPIPath(i));
      end;

  ReportMemo.Lines.Add('');
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+NfoAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner
+AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerVS);
```

```
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+NfoAntiVirus_with_signatureKernel +GetKernelVersion);
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+NfoAntiVirus_with_signatureBuild +GetKernelBuild);
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+DBKnowledge+IntToStr(GetDBRecCount));

  if GetDBVersionDate = '01.01.1880' then
    ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+' '+LASTDBDATA+'0')
  else
    ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+LASTDBDATA+GetDBVersionDate);

  ReportMemo.Lines.Add(LogBevel);
  ReportMemo.Lines.Add('');

  if OptionsForm.RegisterSysMenu.Checked then begin

OptionsForm.FileTAddAction('*','AntiVirus_with_signature.VirtualInfrastructureCy
bersecurityScan',SysMenu,ParamStr(0)+' %1');

OptionsForm.FileTAddAction('Directory','AntiVirus_with_signature.VirtualInfrastr
uctureCybersecurityScan',SysMenu,ParamStr(0)+' %1');

OptionsForm.FileTAddAction('Drive','AntiVirus_with_signature.VirtualInfrastructu
reCybersecurityScan',SysMenu,ParamStr(0)+' %1');
  end else
  begin

OptionsForm.FileTDelAction('Drive','AntiVirus_with_signature.VirtualInfrastructu
reCybersecurityScan');

OptionsForm.FileTDelAction('Directory','AntiVirus_with_signature.VirtualInfrastr
uctureCybersecurityScan');

OptionsForm.FileTDelAction('*','AntiVirus_with_signature.VirtualInfrastructureCy
bersecurityScan');
  end;
end;

//***Функція початку сканування***//

Procedure TMainForm.StartVirtualInfrastructureCybersecurityScan(Parametr:
String);
  var
  T : String;
begin
  if GetDBRecCount = 0 then
  begin
    MessageFrm.Caption := DBerrorI1;
    MessageFrm.InformationLabel.Caption := DBerrorI1;
    MessageFrm.InfoLabel.Caption := DBerrorI2;
    MessageFrm.Memo1.Text := DBerrorI3;
    MessageFrm.ShowModal;
    Exit;
  end;

  if Parametr = 'DRV' then
  begin
    AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner :=
TAvVirus_VirtualInfrastructureCybersecurityScanner.Create(true);

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.NeedForA
PI := TRUE;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.AvAction
:= TVirtualInfrastructureCybersecurityScanDir;
    Path.Add(ExtractFileDrive(Paramstr(0))+'\');
```

```
    AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.Dirs
:= Path;
    OnVirtualInfrastructureCybersecurityScanStart;
    exit;
  end;

  if DirectoryExists(Parametr+'\') then
  begin
    AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner :=
TAvVirus_VirtualInfrastructureCybersecurityScanner.Create(true);

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.NeedForA
PI := TRUE;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.AvAction
:= TVirtualInfrastructureCybersecurityScanDir;
    Path.Add(Parametr+'\');
    AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.Dirs
:= Path;
    OnVirtualInfrastructureCybersecurityScanStart;
    exit;
  end;

  if FileExists(Parametr) then
  begin
    AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner :=
TAvVirus_VirtualInfrastructureCybersecurityScanner.Create(true);

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.NeedForA
PI := false;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.AvAction
:= TVirtualInfrastructureCybersecurityScanFile;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.FileName
:= Parametr;
    OnVirtualInfrastructureCybersecurityScanStart;
    exit;
  end;

end;

procedure TMainForm.ExitBTNClick(Sender: TObject);
begin
  Close;
end;

procedure TMainForm.VirtualInfrastructureCybersecurityScanListDblClick(Sender:
TObject);
begin
  if VirtualInfrastructureCybersecurityScanList.ItemIndex <> -1 then
  begin
    InformationForm.InfoMemo.Text :=
VirtualInfrastructureCybersecurityScanList.Selected.Caption;
    InformationForm.ShowModal;
  end;
end;

procedure TMainForm.VirtualInfrastructureCybersecurityScanBTNClick(Sender:
TObject);
  var
  i: integer;
  err: boolean;
begin
  err:= false;

  for i := 0 to PathList.Items.Count-1 do
  begin
    if PathList.Items.Item[i].Checked then
```

```
    begin
      Path.Add(PathList.Items.Item[i].Caption);
      if not DirectoryExists(PathList.Items.Item[i].Caption+'\') then
        begin
          MessageDlg(PATHNOSEL,mtError,[mbOk],0);
          Exit;
        end;
    end;
  end;

 { if GetDBRecCount = 0 then
  begin
    MessageFrm.Caption := DBerrorI1;
    MessageFrm.InformationLabel.Caption := DBerrorI1;
    MessageFrm.InfoLabel.Caption        := DBerrorI2;
    MessageFrm.Memo1.Text               := DBerrorI3;
    MessageFrm.ShowModal;
    Exit;
  end;     }

  if NeedToReturn = false then
  begin
    if inVirtualInfrastructureCybersecurityScan = False then
    begin
      if PATH.Count-1 <> -1 then
      begin
        AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner
:= TAvVirus_VirtualInfrastructureCybersecurityScanner.Create(true);

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.FreeOnTe
rminate := True;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.NeedForA
PI      := true;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.AvAction
:= TVirtualInfrastructureCybersecurityScanDir;

AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.Dirs
:= MainForm.Path;
        OnVirtualInfrastructureCybersecurityScanStart;
      end
      else begin
        MessageDlg(PATHNOSEL,mtError,[mbOk],0);
      end;
    end
    else begin
      CloseVirtualInfrastructureCybersecurityScanThread;
    end;
  end else
  begin
    VirtualInfrastructureCybersecurityScanBTN.Caption :=
VirtualInfrastructureCybersecurityScanB;
    MainForm.SaveBTN.Enabled := False;
    NeedToReturn := False;
    VirtualInfrastructureCybersecurityScanPathesTab.Show;
  end;
end;

procedure TMainForm.SaveBTNClick(Sender: TObject);
  var
  Report: TStringList;
  i: integer;
begin
  if SaveDialog.Execute then
  begin
    Report:= TStringList.Create;
    For i := 0 to VirtualInfrastructureCybersecurityScanList.Items.Count-1 do
```

```
Report.Add(VirtualInfrastructureCybersecurityScanList.Items.Item[i].Caption);
    Report.SaveToFile(SaveDialog.FileName);
    Report.Free;
  end;
end;

procedure TMainForm.DeletePathClick(Sender: TObject);
begin
  try
    if PathList.ItemIndex <> -1 then
      if PathList.Selected.ImageIndex > 3 then
      begin
        OptionsForm.PathList.Items.Delete(PathList.Selected.Index-
((PathList.Items.Count-1) - (OptionsForm.PathList.items.count-1)));
        PathList.Items.Delete(PathList.Selected.Index);
      end;
    OptionsForm.SaveOptions;
  except
  end;
end;

procedure TMainForm.RefteshClick(Sender: TObject);
begin
  CreateDrivesList(PathList);
end;

procedure TMainForm.AddFolderClick(Sender: TObject);
begin
  AddUserPathForm.ShowModal;
end;

procedure TMainForm.AboutBTNClick(Sender: TObject);
begin
 DBKnowledge+IntToStr(GetDBRecCount);
 AboutForm.ShowModal;
end;

procedure TMainForm.FormShow(Sender: TObject);
begin
  VersionLabel.Caption :=
AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerVS;
end;

procedure TMainForm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  if MessageDlg(AntiVirus_with_signatureCloseDlg,mtInformation,[mbYes]+[mbNo],0)
= 6 then begin
    if OptionsForm.AutoSaveReport.Checked then begin
      MainForm.ReportMemo.Lines.SaveToFile(OptionsForm.ReportSavePath.Text);
    end;
  end else Action := caNone;
end;

procedure TMainForm.HelpBTNClick(Sender: TObject);
begin
  if FileExists(ExtractFilePath(paramstr(0))+'\Help.chm') then
    ShellExecute(0,'',PChar(ExtractFilePath(paramstr(0))+'\Help.chm'),nil,nil,1)
  else
    MessageDlg(HelpNOFound,mtError,[mbOk],0);
end;

procedure TMainForm.DelMenuPopup(Sender: TObject);
begin
  if (VirtualInfrastructureCybersecurityScanList.ItemIndex <> -1) and
(VirtualInfrastructureCybersecurityScanList.Selected.ImageIndex = 2) and
(inVirtualInfrastructureCybersecurityScan = False) then
  begin
    Del.Visible := true;
```

```
  end
  else
    Del.Visible := False;

  if (VirtualInfrastructureCybersecurityScanList.ItemIndex <> -1) and
(inVirtualInfrastructureCybersecurityScan = False) then
    DelAll.Visible := true
  else
    DelAll.Visible := false;
end;

procedure TMainForm.DelAllClick(Sender: TObject);
  var
     i,d,e,c: integer;
begin
  d:=0;
  e:=0;
  c:=0;
  if MessageDlg(DelAllDialog,mtInformation,[mbCancel]+[mbYes],0) = 6 then
  begin
        for i := 0 to VirtualInfrastructureCybersecurityScanList.Items.Count - 1
do
        if VirtualInfrastructureCybersecurityScanList.Items.Item[i].ImageIndex =
2 then
      begin
            c:=c+1;
                    try
                        if
DeleteFileBC(VirtualInfrastructureCybersecurityScanList.Items.Item[i].SubItems[0
]) then
          begin
            d:=d+1;

      VirtualInfrastructureCybersecurityScanList.Items.Item[i].ImageIndex := 4;

      ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileDel+VirtualInfra
structureCybersecurityScanList.Items.Item[i].SubItems[0]);
          end
                        else begin

      ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileNotDel+VirtualIn
frastructureCybersecurityScanList.Items.Item[i].SubItems[0]);
            e:=e+1;
                        end;
                    except
                    end;
   end;

  MessageDlg(DelResult + Return
                                  + Return
                              + AllInfected + IntToStr(c) + Return
                        + DeleteInfected + IntToStr(d) + Return
                        + SkippedInfected + IntToStr(e),mtInformation,[mbOK],0);
  end;
end;

procedure TMainForm.DelClick(Sender: TObject);
begin
  if MessageDlg(DelDialog,mtInformation,[mbCancel]+[mbYes],0) = 6 then
  begin
  try
    if
DeleteFileBC(VirtualInfrastructureCybersecurityScanList.Selected.SubItems[0])
then
    begin
      VirtualInfrastructureCybersecurityScanList.Selected.ImageIndex := 4;

ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileDel+VirtualInfrastruct
ureCybersecurityScanList.Selected.SubItems[0]);
```

```
    end
    else begin

ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileNotDel+VirtualInfrastr
uctureCybersecurityScanList.Selected.SubItems[0]);
      MessageDlg(DelError,mtWarning,[mbOk],0);
    end;
  except
  end;
  end;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
  Path := TStringList.Create;
  TopPn.ControlStyle            := ControlStyle + [csOpaque];
  TopRightPanel.ControlStyle    := ControlStyle + [csOpaque];
  Caption                       :=
AntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScannerCapt;
  TopPn.DoubleBuffered                  := true;
  TopRightPanel.DoubleBuffered     := true;
  PathList.DoubleBuffered            := true;
  VirtualInfrastructureCybersecurityScanList.DoubleBuffered             :=
true;
  BottomPanel.DoubleBuffered    := true;
  MonFileCN                     := 0;
  MonFileInfected               := 0;
end;

procedure TMainForm.AppMinimize(Sender: TObject);
begin
  ShowWindow(Application.Handle, SW_HIDE);
end;


procedure TMainForm.FormDestroy(Sender: TObject);
begin
  DestroyTray;
end;

procedure TMainForm.FormHide(Sender: TObject);
begin
  showwindow(Application.handle, SW_HIDE);
  showwindow(MainForm.handle, SW_HIDE);
end;

procedure TMainForm.FormResize(Sender: TObject);
begin
  PathList.Columns.Items[0].Width := PathList.Width - 25;
  VirtualInfrastructureCybersecurityScanList.Columns.Items[0].Width :=
VirtualInfrastructureCybersecurityScanList.Width - 25;
end;

procedure
TMainForm.mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurit
yScannerClick(Sender: TObject);
begin
  DeActiveTray := True;
  MayClose := True;
  showwindow(Application.handle, SW_HIDE);
  showwindow(MainForm.handle, SW_HIDE);
  if not HideForm.ShowHideTip.Checked then
  begin
    HideForm.Show;
    SetForegroundWindow(HideForm.Handle);
    Application.BringToFront;
  end else DeActiveTray := False;
end;
```

```
procedure
TMainForm.mnuShowAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurit
yScannerClick(Sender: TObject);
begin
  DeActiveTray := False;
  showwindow(Application.handle, SW_SHOW);
  showwindow(MainForm.handle, SW_SHOW);
  Application.Restore;
  MayClose := False;
end;

procedure TMainForm.mnuExitClick(Sender: TObject);
begin
  Close;
end;

procedure TMainForm.mnVirtualInfrastructureCybersecurityOptionsClick(Sender:
TObject);
begin
  if not inVirtualInfrastructureCybersecurityScan then begin
    LoadOptions;
    OptionsForm.Show;
  end;
end;

procedure TMainForm.mnuHelpClick(Sender: TObject);
begin
  if FileExists(ExtractFilePath(paramstr(0))+'\Help.chm') then
    ShellExecute(0,'',PChar(ExtractFilePath(paramstr(0))+'\Help.chm'),nil,nil,1)
  else
    MessageDlg(HelpNOFound,mtError,[mbOk],0);
end;

procedure TMainForm.mnuAboutClick(Sender: TObject);
begin
DBKnowledge+IntToStr(GetDBRecCount);
  if GetDBVersionDate = '01.01.1880' then

  try
    AboutForm.ShowModal;
  except
  end;
end;

procedure TMainForm.ApplicationEventsMinimize(Sender: TObject);
begin

mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.C
lick;
end;

procedure TMainForm.FormPaint(Sender: TObject);
begin
  if FirstRun then
    if OptionsForm.AUTOHIDE.Checked then
    begin

mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.C
lick;
    end;
  FirstRun := false;
end;

procedure
TMainForm.VirtualInfrastructureCybersecurityScanListCustomDrawItem(Sender:
TCustomListView;
  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);
begin
  with VirtualInfrastructureCybersecurityScanList.Canvas.Brush do
```

```
  begin
    case Item.ImageIndex of
      0: Color := $00FFF1EC;
      2: Color := $00ECECFF;
      1: Color := $00ECFBFF;
      4: Color := $00EDFFEC;
    end;
  end;
end;

procedure TMainForm.VirtualInfrastructureCybersecurityScanTopBtnClick(Sender:
TObject);
begin

VirtualInfrastructureCybersecurityScanMenu.Popup(MainForm.Left+VirtualInfrastruc
tureCybersecurityScanTopBtn.Left+3,MainForm.Top+VirtualInfrastructureCybersecuri
tyScanTopBtn.Top+38);
end;

procedure TMainForm.mnuShowReportClick(Sender: TObject);
begin
  if not inVirtualInfrastructureCybersecurityScan then
    ReportTab.Show;
end;

procedure
TMainForm.mnuSelVirtualInfrastructureCybersecurityScanPathClick(Sender:
TObject);
begin
  if not inVirtualInfrastructureCybersecurityScan then
    VirtualInfrastructureCybersecurityScanPathesTab.Show;
end;

procedure TMainForm.PCTopBtnClick(Sender: TObject);
begin
  MonitorForm.Show;
end;

procedure TMainForm.OptionTopBtnClick(Sender: TObject);
begin
  if not inVirtualInfrastructureCybersecurityScan then begin
    LoadOptions;
    OptionsForm.ShowModal;
  end;
end;

procedure TMainForm.mnuGoToTrayClick(Sender: TObject);
begin

mnuHideAntiVirus_with_signatureVirus_VirtualInfrastructureCybersecurityScanner.C
lick;
end;

procedure TMainForm.mnuPCShowClick(Sender: TObject);
begin
  MonitorForm.Show;
end;

procedure TMainForm.mnuPCRunClick(Sender: TObject);
begin
  MonitorForm.StartPC.Click;
end;

procedure TMainForm.mnuPCPauseClick(Sender: TObject);
begin
  MonitorForm.PausePC.Click;
end;

procedure TMainForm.mnuPCStopClick(Sender: TObject);
```

```
begin
  MonitorForm.StopPC.Click;
end;

procedure TMainForm.TrayMenuPopup(Sender: TObject);
begin
  mnuPCRun.Enabled := MonitorForm.StartPC.Enabled;
  mnuPCPause.Enabled := MonitorForm.PausePC.Enabled;
  mnuPCStop.Enabled := MonitorForm.StopPC.Enabled;
  if inVirtualInfrastructureCybersecurityScan then
mnVirtualInfrastructureCybersecurityOptions.Enabled := False else
mnVirtualInfrastructureCybersecurityOptions.Enabled := True;
end;

procedure TMainForm.VirtualInfrastructureCybersecurityScanMenuPopup(Sender:
TObject);
begin
  mnuPCRun.Enabled := MonitorForm.StartPC.Enabled;
  mnuPCPause.Enabled := MonitorForm.PausePC.Enabled;
  mnuPCStop.Enabled := MonitorForm.StopPC.Enabled;
  mnuSaveReport.Enabled := SaveBTN.Enabled;
  if inVirtualInfrastructureCybersecurityScan then
mnuVirtualInfrastructureCybersecurityScanStart.Enabled := False else
mnuVirtualInfrastructureCybersecurityScanStart.Enabled := True;
  if inVirtualInfrastructureCybersecurityScan then
mnuStopVirtualInfrastructureCybersecurityScan.Enabled := True else
mnuStopVirtualInfrastructureCybersecurityScan.Enabled := False;
end;

procedure TMainForm.mnuVirtualInfrastructureCybersecurityScanStartClick(Sender:
TObject);
begin
  VirtualInfrastructureCybersecurityScanBTN.Click;
end;

procedure TMainForm.mnuStopVirtualInfrastructureCybersecurityScanClick(Sender:
TObject);
begin
  VirtualInfrastructureCybersecurityScanBTN.Click;
end;

procedure TMainForm.mnuSaveReportClick(Sender: TObject);
begin
  SaveBTN.Click;
end;

procedure TMainForm.CopyRightLabelClick(Sender: TObject);
  Const
  begin
  ShellExecute(0,'',pChar(''+URL),NIL,NIL,SW_SHOWNORMAL);
end;

end.
```

Файл VirtualInfrastructureCybersecurityMonitor.pas – монітор (контроль процесів)

```
unit VirtualInfrastructureCybersecurityMonitor;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls,
VirtualInfrastructureCybersecurityKernel,
VirtualInfrastructureCybersecurityTypes, TLHelp32, Psapi;

type
  TMonitorForm = class(TForm)
    TopPanel: TPanel;
    BackImage: TImage;
    InformationLabel: TLabel;
    Image1: TImage;
    InfoLabel: TLabel;
    Bevel: TBevel;
    StartPC: TButton;
    PausePC: TButton;
    ClosePC: TButton;
    LastInfectBox: TGroupBox;
    Edit1: TEdit;
    Edit2: TEdit;
    LastFileBox: TGroupBox;
    Edit3: TEdit;
    InfoPCLabel: TGroupBox;
    PCVirtualInfrastructureCybersecurityScaned: TLabel;
    PCInfected: TLabel;
    PCStat: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    PCTime: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Timer1: TTimer;
    StopPC: TButton;
    Timer2: TTimer;
    procedure StartPCClick(Sender: TObject);
    procedure PausePCClick(Sender: TObject);
    procedure ClosePCClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure StopPCClick(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure CreateParams(var Params: TCreateParams); override;
    Procedure StartMonitor;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  MonitorForm : TMonitorForm;
  H,M,S       : integer;
  MonPaused   : Boolean = False;
  isMonRun    : Boolean = False;
  ProcList    : TStrings;
  FileLast    : String;
  FileLastID  : integer;
implementation

uses VirtualInfrastructureCybersecurityMain,
VirtualInfrastructureCybersecurityInfectedAction,
VirtualInfrastructureCybersecurityOptions;
//***Функція створення параметрів сканування***//
```

```
procedure TMonitorForm.CreateParams(var Params: TCreateParams);
begin
  inherited CreateParams(Params);
  with params do
    ExStyle := ExStyle or WS_EX_APPWINDOW;
end;

//***Функція відображення вікна попередження про віруси***//

Procedure ShowAlarmForm(FileName,VirName: String);
  var
  ActFrm : TActionForm;
begin
  if OptionsForm.PCAutoAction.Checked then
  begin
    if OptionsForm.PCDelInfect.Checked then
      if Not DeleteFileBC(FileName) then ShowMessage(MainForm.DelError);
    Exit;
  end;
  ActFrm := TActionForm.Create(nil);
  with ActFrm do begin
    Edit1.Text := FileName;
    Edit2.Text := VirName;
  end;
  ActFrm.Show;
  SetForegroundWindow(ActFrm.Handle);
  ActFrm.SetFocus;
end;

//***Функція створення журналу перевірки***//

procedure CreateWinProcessList(List: Tstrings);
  var
  hSnapShot: THandle;
  ProcInfo: TProcessEntry32;
begin
  if List = nil then Exit;
  hSnapShot := CreateToolHelp32Snapshot(TH32CS_SNAPPROCESS, 0);
  if (hSnapShot <> THandle(-1)) then
  begin
    ProcInfo.dwSize := SizeOf(ProcInfo);
    if (Process32First(hSnapshot, ProcInfo)) then
    begin
      List.Add(ProcInfo.szExeFile);
      while (Process32Next(hSnapShot, ProcInfo)) do begin
        List.Add(ProcInfo.szExeFile);
        end;
    end;
    CloseHandle(hSnapShot);
  end;
end;

procedure CreateWinNTProcessList(List: TStrings);
  var
  PIDArray: array [0..1023] of DWORD;
  cb: DWORD;
  I: Integer;
  ProcCount: Integer;
  hMod: HMODULE;
  hProcess: THandle;
  ModuleName: array [0..300] of Char;
begin
  if List = nil then Exit;
  EnumProcesses(@PIDArray, SizeOf(PIDArray), cb);
  ProcCount := cb div SizeOf(DWORD);
  for I := 0 to ProcCount - 1 do
  begin
    hProcess := OpenProcess(PROCESS_QUERY_INFORMATION or
```

```
      PROCESS_VM_READ,
      False,
      PIDArray[I]);
    if (hProcess <> 0) then
    begin
      EnumProcessModules(hProcess, @hMod, SizeOf(hMod), cb);
      GetModuleFilenameEx(hProcess, hMod, ModuleName, SizeOf(ModuleName));
      if FileExists(ModuleName) then
        List.Add(ModuleName);
      CloseHandle(hProcess);
    end;
  end;
end;

procedure GetProcessList(List: Tstrings);
  var
  ovi: TOSVersionInfo;
begin
  if List = nil then Exit;
  ovi.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  GetVersionEx(ovi);
  case ovi.dwPlatformId of
    VER_PLATFORM_WIN32_WINDOWS: CreateWinProcessList(List);
    VER_PLATFORM_WIN32_NT: CreateWinNTProcessList(List);
  end
end;

//***Функція знищення процесу вірусу***//

function KillProcess(ProcCapt: String): boolean;
  var
  ProgCap       : string;
  hSnapShot     : THandle;
  uProcess      : PROCESSENTRY32;
  r             : longbool;
  KillProc      : DWORD;
  hProcess      : THandle;
  cbPriv        : DWORD;
  Priv,PrivOld  : TOKEN_PRIVILEGES;
  hToken        : THandle;
  dwError       : DWORD;
begin
  ProgCap:= ProcCapt;
  hSnapShot:=CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);
  uProcess.dwSize := Sizeof(uProcess);

  try
    if(hSnapShot<>0)then
    begin
      r:=Process32First(hSnapShot, uProcess);
      while r <> false do
      begin
        if  ProgCap = uProcess.szExeFile then
          KillProc:= uProcess.th32ProcessID;
        r:=Process32Next(hSnapShot, uProcess);
      end;
    CloseHandle(hProcess);
    CloseHandle(hSnapShot);
    end;
  except
  end;

  hProcess:=OpenProcess(PROCESS_TERMINATE,false,KillProc);
  if hProcess = 0 then
  begin
    cbPriv:=SizeOf(PrivOld);
    OpenThreadToken(GetCurrentThread,TOKEN_QUERY or
TOKEN_ADJUST_PRIVILEGES,false,hToken);
```

```
    OpenProcessToken(GetCurrentProcess,TOKEN_QUERY or
TOKEN_ADJUST_PRIVILEGES,hToken);
    Priv.PrivilegeCount:=1;
    Priv.Privileges[0].Attributes:=SE_PRIVILEGE_ENABLED;

LookupPrivilegeValue(nil,'SeVirtualInfrastructureCybersecurityProjectPrivilege',
Priv.Privileges[0].Luid);
    AdjustTokenPrivileges(hToken,false,Priv,SizeOf(Priv),PrivOld,cbPriv);
    hProcess:=OpenProcess(PROCESS_TERMINATE,false,KillProc);
    dwError:=GetLastError;
    cbPriv:=0;
    AdjustTokenPrivileges(hToken,false,PrivOld,SizeOf(PrivOld),nil,cbPriv);
    CloseHandle(hToken);
  end;

  if TerminateProcess(hProcess,$FFFFFFFF) then
  begin
    Result := True;
  end
  else
  begin
    Result := False;
  end;
end;

//***Функція перехвату управління процесами***//

Procedure ExecuteProcessControl;
  var
  i, ID: integer;
begin
  ProcList := TStringList.Create;
  GetProcessList(ProcList);
  For i := 0 to ProcList.Count-1 do
  begin
    Application.ProcessMessages;
    MainForm.MonFileCN := MainForm.MonFileCN + 1;
    MonitorForm.Label4.Caption := inttostr(MainForm.MonFileCN);
    MonitorForm.Edit3.Text := ProcList[i];
    ID := _VirtualInfrastructureCybersecurityScanFileEx(ProcList[i]);
    if ID <> -1 then begin
      MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.ProcControlSt+' ' + '['+MainForm.INFECTED+' - '+GetVirusName(ID)+']
'+ProcList[i]);
      MainForm.MonFileInfected := MainForm.MonFileInfected + 1;
      MonitorForm.Label5.Caption := inttostr(MainForm.MonFileInfected);
      MonitorForm.Edit2.Text := GetVirusName(id);
      MonitorForm.Edit1.Text := ProcList[i];
      MainForm.BalloonTrayIcon(MainForm.Handle
,1,10,ProcList[i],'['+MainForm.INFECTED+' - '+GetVirusName(id)+' ]',bitError);
      if OptionsForm.PCAutoKill.Checked then
        if Not KillProcess(ExtractFileName(ProcList[i])) then
Showmessage(MainForm.ErrorKillProc);
      ShowAlarmForm(ProcList[i],'['+MainForm.INFECTED+' - '+GetVirusName(id)+'
]');
    end;
  end;
  FileLast   := ProcList[ProcList.count-1];
  FileLastID := ProcList.count-1;
end;

//***Функція управління процесами***//

Procedure StartProcessControl;
begin
  if isMonRun = False then begin
    ExecuteProcessControl;
    MonitorForm.Timer2.Enabled := true;
    isMonRun := true;
```

```
    MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.PCInit);
  end else
    if MonPaused then begin
      MonPaused := False;
      MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.PCRestore);
    end;
end;

Procedure PauseProcessControl;
begin
  MonPaused := True;
  MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.PCPause);
end;

Procedure ResumeProcessControl;
begin
  MonPaused := False;
  MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.PCRestore);
end;

Procedure ExitProcessControl;
begin
  isMonRun := False;
  ProcList.Free;
  MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.PCStop);
end;

//***Функція старту моніторингу змін у системі***//

{$R *.dfm}
Procedure TMonitorForm.StartMonitor;
begin
  StartProcessControl;
  PausePC.Enabled := True;
  StopPC.Enabled := True;
  StartPC.Enabled := False;
end;

procedure TMonitorForm.StartPCClick(Sender: TObject);
begin
  StartMonitor;
end;

procedure TMonitorForm.PausePCClick(Sender: TObject);
begin
  PauseProcessControl;
  PausePC.Enabled := False;
  StopPC.Enabled := True;
  StartPC.Enabled := True;
end;

procedure TMonitorForm.ClosePCClick(Sender: TObject);
begin
  Close;
end;

procedure TMonitorForm.Timer1Timer(Sender: TObject);
  var
  ss,mm,hh:String;
begin
  if isMonRun then
    if Not MonPaused then
      Label7.Caption := MainForm.PCActive
    else
```

```
      Label7.Caption := MainForm.PCPaused;

  if not isMonRun then
    Label7.Caption := MainForm.PCStoped;

  if isMonRun then
  if Not MonPaused then
  begin
    s:=s+1;
    if s = 59 then
    begin
      s:=0;
      m:=m+1;
    end;
    if m = 59 then
    begin
      m:=0;
      h:=h+1;
    end;
    ss:=inttostr(s);
    mm:=inttostr(m);
    hh:=inttostr(h);
    if length(ss) = 1 then ss:='0'+ss;
    if length(mm) = 1 then mm:='0'+mm;
    if length(hh) = 1 then hh:='0'+hh;
    Label8.Caption := hh+':'+mm+':'+ss;
  end;
end;

procedure TMonitorForm.StopPCClick(Sender: TObject);
begin
  ExitProcessControl;
  PausePC.Enabled := False;
  StopPC.Enabled := False;
  StartPC.Enabled := True;
end;

procedure TMonitorForm.Timer2Timer(Sender: TObject);
  var
  ID: integer;
begin
  if isMonRun = False then Exit;
  if MonPaused = False then
  begin
    ProcList.Clear;
    GetProcessList(ProcList);
    if ProcList.Count-1 <> FileLastID then
    if ProcList[ProcList.count-1] <> FileLast then
    Begin
      MainForm.MonFileCN := MainForm.MonFileCN + 1;
      MonitorForm.Label4.Caption := inttostr(MainForm.MonFileCN);
      MonitorForm.Edit3.Text := ProcList[ProcList.count-1];
      ID :=
_VirtualInfrastructureCybersecurityScanFileEx(ProcList[ProcList.count-1]);
      if ID <> -1 then
      begin
        MainForm.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MainForm.ProcControlSt+ ' ' + '['+MainForm.INFECTED+' - '+GetVirusName(ID)+']
'+ProcList[ProcList.count-1]);
        MainForm.MonFileInfected := MainForm.MonFileInfected + 1;
        MonitorForm.Label5.Caption := inttostr(MainForm.MonFileInfected);
        MonitorForm.Edit2.Text := GetVirusName(ID);
        MonitorForm.Edit1.Text := ProcList[ProcList.count-1];
        MainForm.BalloonTrayIcon(MainForm.Handle ,1,10, ProcList[ProcList.count-
1] ,'['+MainForm.INFECTED+' - '+GetVirusName(ID)+' ]',bitError);
        if OptionsForm.PCAutoKill.Checked then
        if Not KillProcess(ExtractFileName(ProcList[ProcList.count-1])) then
Showmessage(MainForm.ErrorKillProc);
```

```
        ShowAlarmForm(ProcList[ProcList.count-1],'['+MainForm.INFECTED+' -
'+GetVirusName(ID)+' ]');
      end;
      FileLast   := ProcList[ProcList.count-2];
      FileLastID := ProcList.count-1;
    end else begin
      FileLast   := ProcList[ProcList.count-1];
      FileLastID := ProcList.count-2;
    end;
  end;
end;

end.
```

Файл VirtualInfrastructureCybersecurityProject.dpr – головний файл проекту

```
program VirtualInfrastructureCybersecurityProject;
// Список підключаємих модулів
uses
  Forms,
  SysUtils,
  VirtualInfrastructureCybersecurityKernel in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner
Modues\VirtualInfrastructureCybersecurityKernel.pas',
  VirtualInfrastructureCybersecurityTypes in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner
Modues\VirtualInfrastructureCybersecurityTypes.pas',
  avMonitor in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avMonitor.pas',
  avVirus_VirtualInfrastructureCybersecurityScanner in
'..\AntiVirus_with_signature Virus_VirtualInfrastructureCybersecurityScanner
Modues\avVirus_VirtualInfrastructureCybersecurityScanner.pas',
  avHex in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avHex.pas',
  avDataBase in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avDataBase.pas',
  avHash in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avHash.pas',
  avExt in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avExt.pas',
  avAPI in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avAPI.pas',
  avConfig in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avConfig.pas',
  avShield in '..\AntiVirus_with_signature
Virus_VirtualInfrastructureCybersecurityScanner Modues\avShield.pas',
  langs in 'langs.pas',
  VirtualInfrastructureCybersecurityMain in
'VirtualInfrastructureCybersecurityMain.pas' {MainForm},
  uSelInfo in 'uSelInfo.pas' {InformationForm},
  VirtualInfrastructureCybersecurityOptions in
'VirtualInfrastructureCybersecurityOptions.pas' {OptionsForm},
  uPluginInfo in 'uPluginInfo.pas' {PluginAPIForm},
  VirtualInfrastructureCybersecurityAddPath in
'VirtualInfrastructureCybersecurityAddPath.pas' {AddUserPathForm},
  VirtualInfrastructureCybersecurityAbout in
'VirtualInfrastructureCybersecurityAbout.pas' {AboutForm},
  uSelDir in 'uSelDir.pas' {SelDirFrm},
  uMessage in 'uMessage.pas' {MessageFrm},
  uHideForm in 'uHideForm.pas' {HideForm},
  VirtualInfrastructureCybersecurityMonitor in
'VirtualInfrastructureCybersecurityMonitor.pas' {MonitorForm},
  VirtualInfrastructureCybersecurityInfectedAction in
'VirtualInfrastructureCybersecurityInfectedAction.pas' {ActionForm},
  uSplash in 'uSplash.pas' {SplashForm};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Virus_VirtualInfrastructureCybersecurityScanner';
  Application.CreateForm(TMainForm, MainForm);
  Application.CreateForm(TInformationForm, InformationForm);
  Application.CreateForm(TOptionsForm, OptionsForm);
  Application.CreateForm(TPluginAPIForm, PluginAPIForm);
  Application.CreateForm(TAddUserPathForm, AddUserPathForm);
  Application.CreateForm(TAboutForm, AboutForm);
  Application.CreateForm(TSelDirFrm, SelDirFrm);
  Application.CreateForm(TMessageFrm, MessageFrm);
  Application.CreateForm(THideForm, HideForm);
  Application.CreateForm(TMonitorForm, MonitorForm);
  Application.CreateForm(TActionForm, ActionForm);
```

```
  Application.CreateForm(TSplashForm, SplashForm);
  {Show Splash form}
  SplashForm.CRLabel.Caption   := 'Kernel '+GetKernelVersion;
  SplashForm.CRLabel00.Caption := 'Build ' +GetKernelBuild;
  SplashForm.Show;
  {}
  Init;
  langs.SwitchAllFormsToLng(01,01,ExtractFilePath(Paramstr(0))+'default.lng');
  {init kernel}
  MainForm.InitVirus_VirtualInfrastructureCybersecurityScannerKernel;
  {Hide Splash Form}
  SplashForm.Hide;
  Sleep(200);
  {Create Tray Icon}
  MainForm.CreateTray;
  {}
  if OptionsForm.AUTORUN.Checked then begin

OptionsForm.ChangeReg('Virus_VirtualInfrastructureCybersecurityScanner',False);
  end else begin

OptionsForm.ChangeReg('Virus_VirtualInfrastructureCybersecurityScanner',True);
  end;
  {}
  if ParamStr(1) <> '' then
  MainForm.StartVirtualInfrastructureCybersecurityScan(ParamStr(1));
  {}
  if OptionsForm.PCAutoLoad.Checked then begin
    MonitorForm.StartMonitor;
  end;
  {}
  Application.Run;
end.
```

Файл VirtualInfrastructureCybersecurityAddPath.pas – додавання шляхів сканування

```
unit VirtualInfrastructureCybersecurityAddPath;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, ShellCtrls;

type
  TAddUserPathForm = class(TForm)
    Bevel: TBevel;
    TopPanel: TPanel;
    Image13: TImage;
    InformationLabel: TLabel;
    InfoLabel: TLabel;
    ApplyBTN: TButton;
    CanselBTN: TButton;
    ShellTreeView: TShellTreeView;
    Image1: TImage;
    procedure CanselBTNClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure ShellTreeViewClick(Sender: TObject);
    procedure ApplyBTNClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AddUserPathForm: TAddUserPathForm;

implementation

uses VirtualInfrastructureCybersecurityMain,
VirtualInfrastructureCybersecurityOptions, uSelInfo;

{$R *.dfm}

procedure TAddUserPathForm.CanselBTNClick(Sender: TObject);
begin
  Close;
end;
procedure TAddUserPathForm.FormShow(Sender: TObject);
begin
  ApplyBTN.Enabled := false;
end;
procedure TAddUserPathForm.ShellTreeViewClick(Sender: TObject);
begin
  if DirectoryExists(ShellTreeView.Path+'\') then
    ApplyBTN.Enabled := True else
    ApplyBTN.Enabled := False;
end;

procedure TAddUserPathForm.ApplyBTNClick(Sender: TObject);
begin
  with OptionsForm.PathList.Items.Add do
  begin
    Caption := ShellTreeView.Path+'\';
    if DirectoryExists(Caption) then ImageIndex := 4 else ImageIndex := 5;
  end;
  OptionsForm.SaveOptions;
  MainForm.CreateDrivesList(MainForm.PathList);
  Close;
end;

end.
```

Файл VirtualInfrastructureCybersecurityInfectedAction.pas – вибір дії над інфікованим об'єктом

```
unit VirtualInfrastructureCybersecurityInfectedAction;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, VirtualInfrastructureCybersecurityKernel;

type
  TActionForm = class(TForm)
    DeleteVir: TButton;
    SkipVir: TButton;
    ApplyToAll_Check: TCheckBox;
    Bevel1: TBevel;
    InfoInfectedBox: TGroupBox;
    InfoVirusInfo: TGroupBox;
    Edit1: TEdit;
    VirInfo_2: TLabel;
    VirInfo_0: TLabel;
    VirInfo_1: TLabel;
    TopPanel: TPanel;
    BackImage: TImage;
    InformationLabel: TLabel;
    InfoLabel: TLabel;
    Image2: TImage;
    Bevel: TBevel;
    Edit2: TEdit;
    procedure SkipVirClick(Sender: TObject);
    procedure DeleteVirClick(Sender: TObject);
    procedure CreateParams(var Params: TCreateParams); override;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ActionForm: TActionForm;

implementation

uses VirtualInfrastructureCybersecurityMain,
VirtualInfrastructureCybersecurityOptions;

{$R *.dfm}
procedure TActionForm.CreateParams(var Params: TCreateParams);
begin
  inherited CreateParams(Params);
  with params do
    ExStyle := ExStyle or WS_EX_APPWINDOW;
end;

procedure TActionForm.SkipVirClick(Sender: TObject);
begin
  if ApplyToAll_Check.Checked then
  begin
    OptionsForm.PCAutoAction.Checked := True;
    OptionsForm.PCSkipInfect.Checked := true;
    OptionsForm.SaveOptions;
  end;
  Close;
end;
//*********функція знищення вірусу**********
procedure TActionForm.DeleteVirClick(Sender: TObject);
begin
```

```
  if ApplyToAll_Check.Checked then
  begin
    OptionsForm.PCAutoAction.Checked := True;
    OptionsForm.PCDelInfect.Checked := true;
    OptionsForm.SaveOptions;
  end;
  if Not DeleteFileBC(Edit1.Text) then ShowMessage(MainForm.DelError)
  else Close;
end;

end.
```

Файл VirtualInfrastructureCybersecurityOptions.pas – параметри антивірусу

```
unit VirtualInfrastructureCybersecurityOptions;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, ComCtrls, registry,
VirtualInfrastructureCybersecurityKernel,
VirtualInfrastructureCybersecurityTypes;

type
  TOptionsForm = class(TForm)
    Bevel: TBevel;
    TopPanel: TPanel;
    BackImage: TImage;
    InformationLabel: TLabel;
    InfoLabel: TLabel;
    ApplyBTN: TButton;
    CanselBTN: TButton;
    OptionsPages: TPageControl;
    optTabOther: TTabSheet;
    optTabPathes: TTabSheet;
    optTabModules: TTabSheet;
    AutoSaveReport: TCheckBox;
    ReportSavePath: TEdit;
    EditSaveReportBTN: TSpeedButton;
    optTabFilter: TTabSheet;
    ExtList: TListView;
    PathList: TListView;
    APIList: TListView;
    AddBTN: TSpeedButton;
    DelBTN: TSpeedButton;
    EditBTN: TSpeedButton;
    SaveDialog: TSaveDialog;
    DisplayScnFiles: TCheckBox;
    optReportLabel: TLabel;
    optSysLabel: TLabel;
    RegisterSysMenu: TCheckBox;
    OPTModulePanel: TPanel;
    ModulesLOAD: TCheckBox;
    optModInfLabel: TLabel;
    optModListLabel: TLabel;
    optShieldLabel: TLabel;
    USESHIELD: TCheckBox;
    SHIELDSILENT: TCheckBox;
    optTabMain: TTabSheet;
    DBDirLabel: TLabel;
    DBPATH: TEdit;
    Bevel6: TBevel;
    optPathesLabel: TLabel;
    SpeedButton1: TSpeedButton;
    ModDirLabel: TLabel;
    MODULESPATH: TEdit;
    SpeedButton2: TSpeedButton;
    Bevel7: TBevel;
    optVirtualInfrastructureCybersecurityScanLabel: TLabel;
    SCNSUBDIR: TCheckBox;
    SCNHEX: TCheckBox;
    SCNCRC: TCheckBox;
    SCNHEXINPOS: TCheckBox;
    SCNBIT: TCheckBox;
    AUTORUN: TCheckBox;
    AUTOHIDE: TCheckBox;
    Image1: TImage;
    Bevel1: TBevel;
    Bevel2: TBevel;
```

```
    Bevel5: TBevel;
    Bevel3: TBevel;
    Bevel4: TBevel;
    optTabPC: TTabSheet;
    optPCLabel: TLabel;
    Bevel8: TBevel;
    PCAutoLoad: TCheckBox;
    PCAutoKill: TCheckBox;
    PCAutoAction: TCheckBox;
    PCDelInfect: TRadioButton;
    PCSkipInfect: TRadioButton;
    optPCInfoLabel: TLabel;
    SHOWBALOONHINT: TCheckBox;
    procedure ApplyBTNClick(Sender: TObject);
    procedure optTabOtherShow(Sender: TObject);
    procedure optTabFilterShow(Sender: TObject);
    procedure optTabPathesShow(Sender: TObject);
    procedure optTabModulesShow(Sender: TObject);
    Procedure SaveOptions;
    procedure CanselBTNClick(Sender: TObject);
    procedure APIListDblClick(Sender: TObject);
    procedure AddBTNClick(Sender: TObject);
    procedure DelBTNClick(Sender: TObject);
    procedure EditBTNClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure EditSaveReportBTNClick(Sender: TObject);
    procedure FileTAddAction(key, name, display, action: String);
    procedure FileTDelAction(key, name: String);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure optTabMainShow(Sender: TObject);
    procedure ChangeReg(StrName: ShortString; delete: boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  OptionsForm: TOptionsForm;

implementation

uses VirtualInfrastructureCybersecurityMain, uPluginInfo,
VirtualInfrastructureCybersecurityAddPath, uSelDir, uHideForm;

{$R *.dfm}
//**********Запис у реєстр системи***********
procedure TOptionsForm.ChangeReg(StrName: ShortString; delete: boolean);
  var
  reg: TRegistry;
begin
  Reg := nil;
  try
    reg := TRegistry.Create;
    reg.RootKey := HKEY_LOCAL_MACHINE;
    reg.LazyWrite := false;
    reg.OpenKey('Software\Microsoft\Windows\CurrentVersion\Run',false);
  if not delete then reg.WriteString(StrName, ParamStr(0)+' -M')
  else reg.DeleteValue(StrName);
    reg.CloseKey;
    reg.free;
  except
    if Assigned(Reg) then Reg.Free;
  end;
end;

procedure TOptionsForm.FileTDelAction(key, name: String);
  var
```

```
  myReg: TRegistry;
begin
  try
    myReg:=TRegistry.Create;
    myReg.RootKey:=HKEY_CLASSES_ROOT;
    if key[1] = '.' then
      key := copy(key,2,maxint)+'_auto_file';
    if key[Length(key)-1] <> '\' then
      key:=key+'\';
    myReg.OpenKey('\'+key+'shell\', true);
    if myReg.KeyExists(name) then
      myReg.DeleteKey(name);
    myReg.CloseKey;
    myReg.Free;
  except
  end;
end;

procedure TOptionsForm.FileTAddAction(key, name, display, action: String);
  var
  myReg:TRegistry;
begin
  try
    myReg:=Tregistry.Create;
    myReg.RootKey:=HKEY_CLASSES_ROOT;
    if name='' then name:=display;

    if key[1] = '.' then
      key:= copy(key,2,maxint)+'_auto_file';

    if key[Length(key)-1] <> '\' then
      key:=key+'\';
    if name[Length(name)-1] <> '\' then
      name:=name+'\';
    myReg.OpenKey(key+'Shell\'+name, true);
    myReg.WriteString('', display);
    MyReg.CloseKey;
    MyReg.OpenKey(key+'Shell\'+name+'Command\', true);
    MyReg.WriteString('', action);
    myReg.Free;
  except
  end;
end;

Procedure TOptionsForm.SaveOptions;
  var
  i:integer;
begin
  if AUTORUN.Checked then
  begin
    ChangeReg('Virus_VirtualInfrastructureCybersecurityScanner',False);
  end else
  begin
    ChangeReg('Virus_VirtualInfrastructureCybersecurityScanner',True);
  end;

//******//

        OPT_MODULES_LOAD        := ModulesLOAD.Checked;
        OPT_DB_DIR              := DBPATH.Text;
        OPT_MODULE_DIR          := MODULESPATH.Text;
        OPT_USE_SHIELD          := USESHIELD.Checked;
        OPT_SILENT_SHIELD_MODE  := SHIELDSILENT.Checked;
        OPT_VIRTUALINFRASTRUCTURECYBERSECURITYSCAN_SUBDIR         :=
SCNSUBDIR.Checked;
        OPT_USE_HEX_MODE        := SCNHEX.Checked;
        OPT_USE_CRC_MODE        := SCNCRC.Checked;
        OPT_USE_HEX_INPOS       := SCNHEXINPOS.Checked;
```

```
        OPT_SEND_VIRTUALINFRASTRUCTURECYBERSECURITYSCAN_FILE      :=
DisplayScnFiles.Checked;
        OPT_USE_BYTE_MODE       := SCNBIT.Checked;
//******//
        ClearOtherParamList;
//******//
        if SHOWBALOONHINT.Checked then AddOtherParamString('SHOWBALOONHINT=ON')
        else AddOtherParamString('SHOWBALOONHINT=OFF');

        if PCAutoLoad.Checked then AddOtherParamString('PROCCONTROLAUTOMODE=ON')
        else AddOtherParamString('PROCCONTROLAUTOMODE=OFF');

        if PCAutoKill.Checked then AddOtherParamString('PROCCONTROLAUTOKILL=ON')
        else AddOtherParamString('PROCCONTROLAUTOKILL=OFF');

        if PCAutoAction.Checked then
AddOtherParamString('PROCCONTROLAUTOACTION=ON')
        else AddOtherParamString('PROCCONTROLAUTOACTION=OFF');

        if PCDelInfect.Checked then
AddOtherParamString('PROCCONTROLDELINFECT=ON')
        else AddOtherParamString('PROCCONTROLDELINFECT=OFF');

        if PCSkipInfect.Checked then
AddOtherParamString('PROCCONTROLSKIPINFECT=ON')
        else AddOtherParamString('PROCCONTROLSKIPINFECT=OFF');

        if AutoSaveReport.Checked then AddOtherParamString('AUTOSAVEREPORT=ON')
        else
        AddOtherParamString('AUTOSAVEREPORT=OFF');
        AddOtherParamString('AUTOSAVEREPORTTO='+ReportSavePath.Text);

        if RegisterSysMenu.Checked then
AddOtherParamString('REGISTERSYSMENU=ON')
        else AddOtherParamString('REGISTERSYSMENU=OFF');

        if AutoRun.Checked then AddOtherParamString('AUTORUN=ON')
        else
        AddOtherParamString('AUTORUN=OFF');

        if AutoHide.Checked then AddOtherParamString('AUTOHIDE=ON')
        else
        AddOtherParamString('AUTOHIDE=OFF');

        if HideForm.ShowHideTip.Checked then AddOtherParamString('HIDETIP=ON')
        else
        AddOtherParamString('HIDETIP=OFF');

        ClearExtList;
        for i := 0 to ExtList.Items.Count-1 do
        AddToExtList(ExtList.Items.Item[i].Caption);

        for i := 0 to PathList.Items.Count-1 do
        AddOtherParamString('PATH='+PathList.Items.Item[i].Caption);
//******//
        SaveConfig_;
//******//
end;

procedure TOptionsForm.ApplyBTNClick(Sender: TObject);
begin
  SaveOptions;
  MainForm.CreateDrivesList(MainForm.PathList);
  if RegisterSysMenu.Checked then
  begin

FileTAddAction('*','AntiVirus_with_signature.VirtualInfrastructureCybersecurityS
can',MainForm.SysMenu,ParamStr(0)+' %1');
```

```
FileTAddAction('Directory','AntiVirus_with_signature.VirtualInfrastructureCybers
ecurityScan',MainForm.SysMenu,ParamStr(0)+' %1');

FileTAddAction('Drive','AntiVirus_with_signature.VirtualInfrastructureCybersecur
ityScan',MainForm.SysMenu,ParamStr(0)+' %1');
  end else
  begin

FileTDelAction('Drive','AntiVirus_with_signature.VirtualInfrastructureCybersecur
ityScan');

FileTDelAction('Directory','AntiVirus_with_signature.VirtualInfrastructureCybers
ecurityScan');

FileTDelAction('*','AntiVirus_with_signature.VirtualInfrastructureCybersecurityS
can');
  end;
  Close;
end;

procedure TOptionsForm.optTabOtherShow(Sender: TObject);
begin
  AddBTN.Enabled := False;
  DelBTN.Enabled := False;
  EditBTN.Enabled := False;
end;

procedure TOptionsForm.optTabFilterShow(Sender: TObject);
begin
  AddBTN.Enabled := true;
  DelBTN.Enabled := true;
  EditBTN.Enabled := true;
end;

procedure TOptionsForm.optTabPathesShow(Sender: TObject);
begin
  AddBTN.Enabled := True;
  DelBTN.Enabled := True;
  EditBTN.Enabled := False;
end;

procedure TOptionsForm.optTabModulesShow(Sender: TObject);
begin
  AddBTN.Enabled := False;
  DelBTN.Enabled := False;
  EditBTN.Enabled := False;
end;

procedure TOptionsForm.CanselBTNClick(Sender: TObject);
begin
  Close;
end;

procedure TOptionsForm.APIListDblClick(Sender: TObject);
begin
  if APIList.ItemIndex <> -1 then
  begin
    PluginAPIForm.NameEdit.Text := APIList.Selected.Caption;
    PluginAPIForm.AutorEdit.Text := APIList.Selected.SubItems[0];
    PluginAPIForm.OtherMemo.Text := APIList.Selected.SubItems[1];
    PluginAPIForm.PathEdit.Text := APIList.Selected.SubItems[2];
    PluginAPIForm.ShowModal;
  end;
end;

procedure TOptionsForm.AddBTNClick(Sender: TObject);
begin
  if optTabFilter.Showing then
```

```
  begin
    with ExtList.Items.Add do begin
      Caption := '';
      ImageIndex := 3;
      EditCaption;
    end;
  end;
  if optTabPathes.Showing then AddUserPathForm.Showmodal;
end;

procedure TOptionsForm.DelBTNClick(Sender: TObject);
begin
  try
    if optTabFilter.Showing then ExtList.Items.Delete(ExtList.Selected.Index);
    if optTabPathes.Showing then PathList.Items.Delete(PathList.Selected.Index);
  except
  end;
end;

procedure TOptionsForm.EditBTNClick(Sender: TObject);
begin
  if optTabFilter.Showing then
    if ExtList.ItemIndex <> -1 then
      ExtList.Selected.EditCaption;
end;

procedure TOptionsForm.FormShow(Sender: TObject);
begin
  optTabMain.Show;
end;

procedure TOptionsForm.EditSaveReportBTNClick(Sender: TObject);
begin
  if SaveDialog.Execute then ReportSavePath.Text := SaveDialog.FileName;
end;

procedure TOptionsForm.SpeedButton1Click(Sender: TObject);
begin
  SelDirFrm.ShowModal;
  if SelDirFrm.ModalResult = mrOk then
  begin
    DBPATH.Text := SelDirFrm.ShellTreeView.Path + '\';
  end;
end;

procedure TOptionsForm.SpeedButton2Click(Sender: TObject);
begin
  SelDirFrm.ShowModal;
  if SelDirFrm.ModalResult = mrOk then
  begin
    MODULESPATH.Text := SelDirFrm.ShellTreeView.Path + '\';
  end;
end;

procedure TOptionsForm.optTabMainShow(Sender: TObject);
begin
  AddBTN.Enabled := False;
  DelBTN.Enabled := False;
  EditBTN.Enabled := False;
end;

end.
```

Файл VirtualInfrastructureCybersecurityAbout.pas – довідка

```
unit VirtualInfrastructureCybersecurityAbout;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons, ShellAPI, ComCtrls, jpeg;

type
  TAboutForm = class(TForm)
    Bevel2: TBevel;
    Panel1: TPanel;
    OkBTN: TBitBtn;
    Bevel1: TBevel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Image1: TImage;
    procedure OkBTNClick(Sender: TObject);
    procedure LinkLabelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AboutForm: TAboutForm;

implementation

uses VirtualInfrastructureCybersecurityMain;

{$R *.dfm}

procedure TAboutForm.OkBTNClick(Sender: TObject);
begin
  Close;
end;

end.
```