**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Центральноукраїнський національний технічний університет**

**Кафедра кібербезпеки та програмного забезпечення**

На правах рукопису

Мінаков Ігор Сергійович

**Програмне забезпечення системи кібербезпеки для проведення пентесту IT інфраструктури**

Спеціальність: 125 «Кібербезпека»

Освітній ступінь: бакалавр

Науковий керівник:

**Смірнов Сергій Анатолійович** _____

(підпис) (дата)

кандидат технічних наук

**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

*Мінакову Ігорю Сергійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для проведення пентесту IT інфраструктури*

керівник роботи *Смірнов Сергій Анатолійович, канд. техн. наук*
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 185-02 від 28.12.2020 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 для проведення пентесту IT 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

**Мінаков І.С. Програмне забезпечення системи кібербезпеки для проведення пентесту IT інфраструктури. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2021.**

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи кібербезпеки для проведення пентесту IT інфраструктури.

Метою розробки є програмне забезпечення системи кібербезпеки для проведення пентесту IT інфраструктури.

Результат роботи – програмна реалізація системи кібербезпеки для проведення пентесту IT 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.

**Ключові слова**: кібербезпека, пентест, IT інфраструктура

# ABSTRACT

**Minakov I.S. Cybersecurity software for pentest IT infrastructure. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2021**

This undergraduate qualification software is developed, please designed for cybersecurity system to conduct pentest IT infrastructure.

The purpose of the development is cybersecurity system software for conducting an IT infrastructure test.

The result is a software implementation of a cybersecurity system to test the IT infrastructure.

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

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

The program can use on a PC architecture IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of Delphi 10.

**Keywords:** cybersecurity, pentest, IT infrastructure

**ЗМІСТ**

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

| | | |
|---|---|---|
| КМ | – | комп’ютерна мережа |
| КСАЗ | – | комплексна система антивірусного захисту |
| МЕ | – | міжмережевий екран |
| ПЗ | – | програмне забезпечення |
| ПК | – | персональний комп’ютер |
| ACL | – | Access Control List |
| FTP | – | File Transfer Protocol |
| http | – | HyperText Transfer Protocol |
| POP3 | – | Post Office Protocol Version 3 |
| SMTP | – | Simple Mail Transfer Protocol |
| VLAN | – | Virtual Local Area Network |

# ВСТУП

**Актуальність теми**. Пентест (penetration test, пенетрейшн тест) – тестування на проникнення й безпеку, інакше аналіз системи на наявність уразливостей. Це метод оцінки безпеки інформаційної системи шляхом моделювання атаки зловмисників. Пентестинг ведеться з позиції потенційного атакуючого й може містити в собі активне використання уразливостей системи.

Ціль тестування – виявити можливі уразливості й недоліки, здатні привести до порушення конфіденційності, цілісності й доступності інформації, спровокувати некоректну роботу системи або привести до відмови від обслуговування, а так само спрогнозувати можливі фінансові втрати й економічні ризики. Тестування торкається як віртуального рівня, так і фізичного.

За результатами тестування на проникнення дається оцінка можливостей поточного рівня захищеності витримати спробу вторгнення потенційного зловмисника, дані про кількість часу й ресурсів, необхідних для успішної атаки на замовника. У випадку виявлення уразливостей в обов'язковому порядку складається список рекомендацій з усунення вищевказаних уразливостей.

Суть робіт полягає в моделюванні дій зловмисника, навмисного одержати доступ до інформаційних систем замовника й порушити цілісність, конфіденційність або доступність приналежної замовникові інформації. Найчастішими об'єктами досліджень є:

- Системи керування базами даних.
- Мережеве устаткування.
- Мережеві служби й сервіси (наприклад, електронна пошта).
- Засобу захисту інформації.
- Прикладне програмне забезпечення.
- Серверні й користувацькі операційні системи.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи кібербезпеки для проведення пентесту ІТ інфраструктури.

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

– Огляд існуючих систем кібербезпеки для проведення пентесту ІТ інфраструктури.

– Дослідження системи кібербезпеки для проведення пентесту ІТ інфраструктури.

– Програмна реалізація системи кібербезпеки для проведення пентесту ІТ інфраструктури.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для проведення пентесту ІТ інфраструктури.

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

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Результатом проведеного тесту на проникнення є звіт фахівця. Форма й зміст звіту не регулюється на законодавчому рівні, що вказує на те, що формат звіту визначається експертом, його складовим. Звичайно звіт містить наступні дані:

– Дані про експерта (експертів), що склав звіт.

– Дата початку й завершення провадження робіт.

– Підстава для виробництва дослідження.

– Надані замовником ресурси.

– Використані матеріали й довідкова література.

– Використані програмні й апаратні засоби.

– Обставини проведення робіт.

– Хід проведення робіт (процес пентестингу).

– Виявлені критичні уразливості.

– Рекомендації з усунення критичних уразливостей.

– Додаткова інформація й застосунку до звіту (посилання, розшифрування)

Таким чином, за результатами проведення penetration test замовник одержує повну інформацію про уразливості власної інформаційної системи, а так само конкретні вказівки й рекомендації до усунення цих уразливостей.

## 1.2 Область застосування

Тестування на проникнення проводиться із застосуванням широкого списку спеціалізованих програм і застосунків (добір паролів, пошук уразливостей портів IP-мереж, виявлення шкідливих програм) і охоплює велику кількість пунктів перевірки. Найпоширеніші з них:

– Збір інформації (пошук даних про замовника у відкритих джерелах, збір даних про допуски співробітників).

– Пошук технічної бази (визначення й збір даних про існуючих ресурси, операційних системи, програмному забезпечення й застосунках).

– Аналіз уразливостей і погроз (виявлення уразливостей у системах безпеки, застосунках і програмному забезпеченні із застосуванням спеціалізованих програм і утиліт).

– Експлуатація й обробка даних (на цьому етапі відбувається імітація реальної атаки зловмисників для одержання відомостей про наявні уразливості з метою наступного аналізу, а так само збір даних про можливі строки злому системи й розрахунків економічних ризиків).

– Формування звіту (етап оформлення отриманої інформації, складання рекомендацій і інструкцій до усунення існуючих уразливостей)

Для чого ж потрібний pen test і як часто необхідно проводити тестування? Як ми вже згадували вище по тексту, тест на проникнення дає найбільш повну картину про стан інформаційної безпеки на підприємстві, дозволяє виявити слабкі й незахищені місця й вчасно вжити заходів по поліпшенню безпеки, дати розуміння про поточну роботу відділів, пов'язаних з інформаційною безпекою, дає план дій по усуненню уразливостей. Багато фахівців по інформаційній безпеці рекомендують проводити penetration test на регулярній основі, найкраще рішення – щорічно. Технології інформаційної безпеки дуже швидко застарівають, рішення, оптимальне для підприємства замовника на даний момент, не буде таким через якийсь час. Слід зазначити, що фахівця для проведення пін тесту

краще вибирати з бо ку, це повинен бути компетентна людина, незацікавлений і безсторонній. Співробітники служби безпеки організації – замовника на цю роль не підходять, тому що прямо зацікавлені в результаті й можуть просто не мати потрібний рівень знань. Експерт із боку, що володіє мінімальними знаннями про архітектуру системи безпеки замовника, з більшою ймовірністю виявить її уразливості. Тест на проникнення в мережу – необхідний елемент забезпечення інформаційної безпеки для будь-якої організації.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для проведення пентесту ІТ 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

**Програми для проведення пентесту**

В penetration test використовуються певні програми для роботи з уразливостями систем, наприклад:

**Metasploit** – програма для надання інформації про уразливості, допомогу в створенні характерних ознак вірусних програм для систем виявлення вторгнень (наприклад, антивірусів), створення й тестування атак на обчислювальні системи.

**Nmap** – утиліта, призначена для сканування, що налаштовується, IP-мереж з будь-якою кількістю об'єктів, визначення стану об'єктів скануємої мережі (портів і відповідних їм служб). Програма доступна в різних версіях для множини операційних систем.

**Nessus** – інструмент для автоматизації перевірки й виявлення уразливостей і проломів у захисті інформаційних систем. Програма поширюється по General Public License, тобто, програма має відкритий вихідний код.

**Kali Linux** – дистрибутив з певними налаштовуваннями, застосунками й інструментами, призначений для етичного хакингу й тестування на проникнення. Дана програма так само працює на декількох платформах.

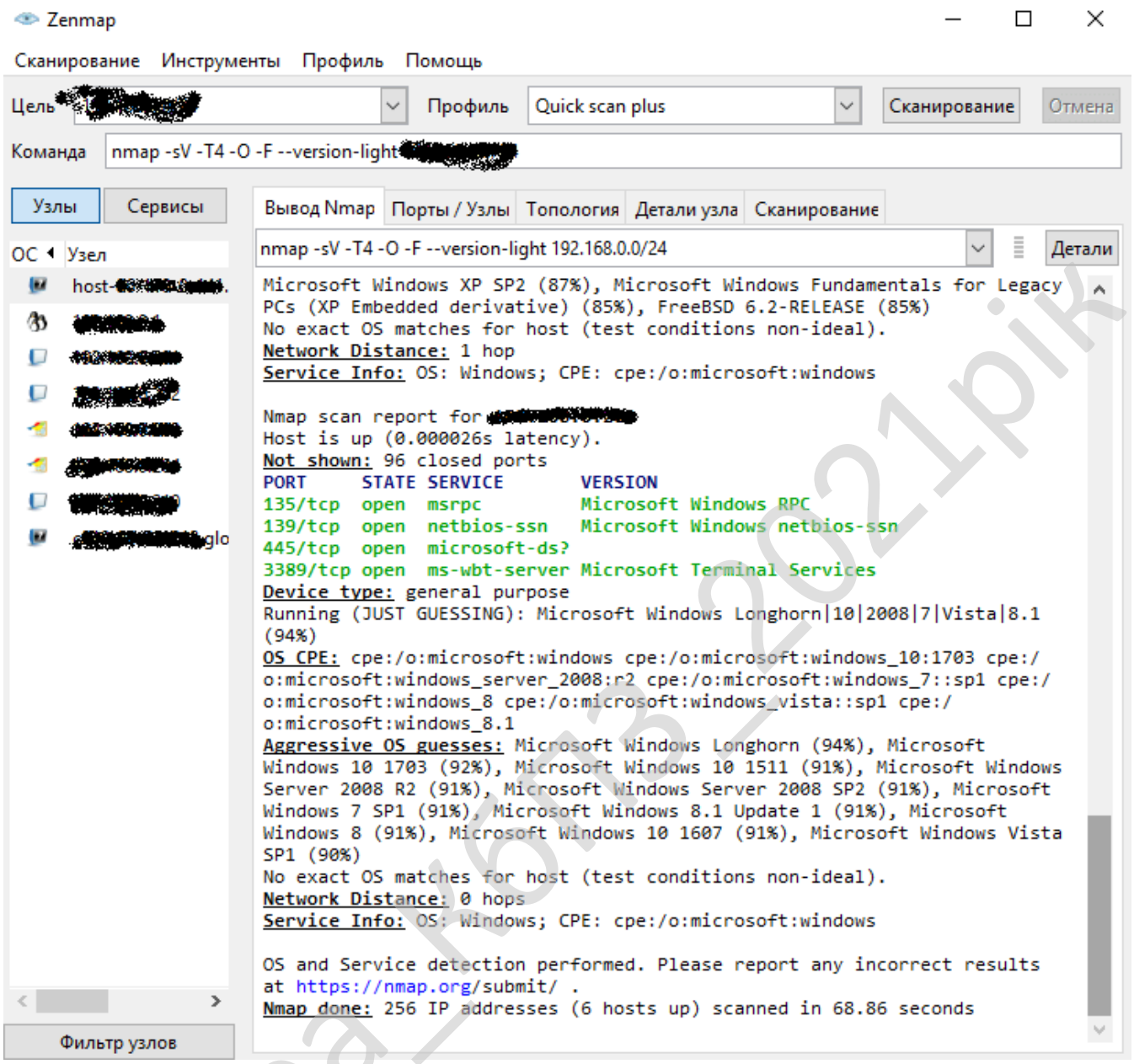


Рисунок 2.1 – Інтерфейс користувача Nmap

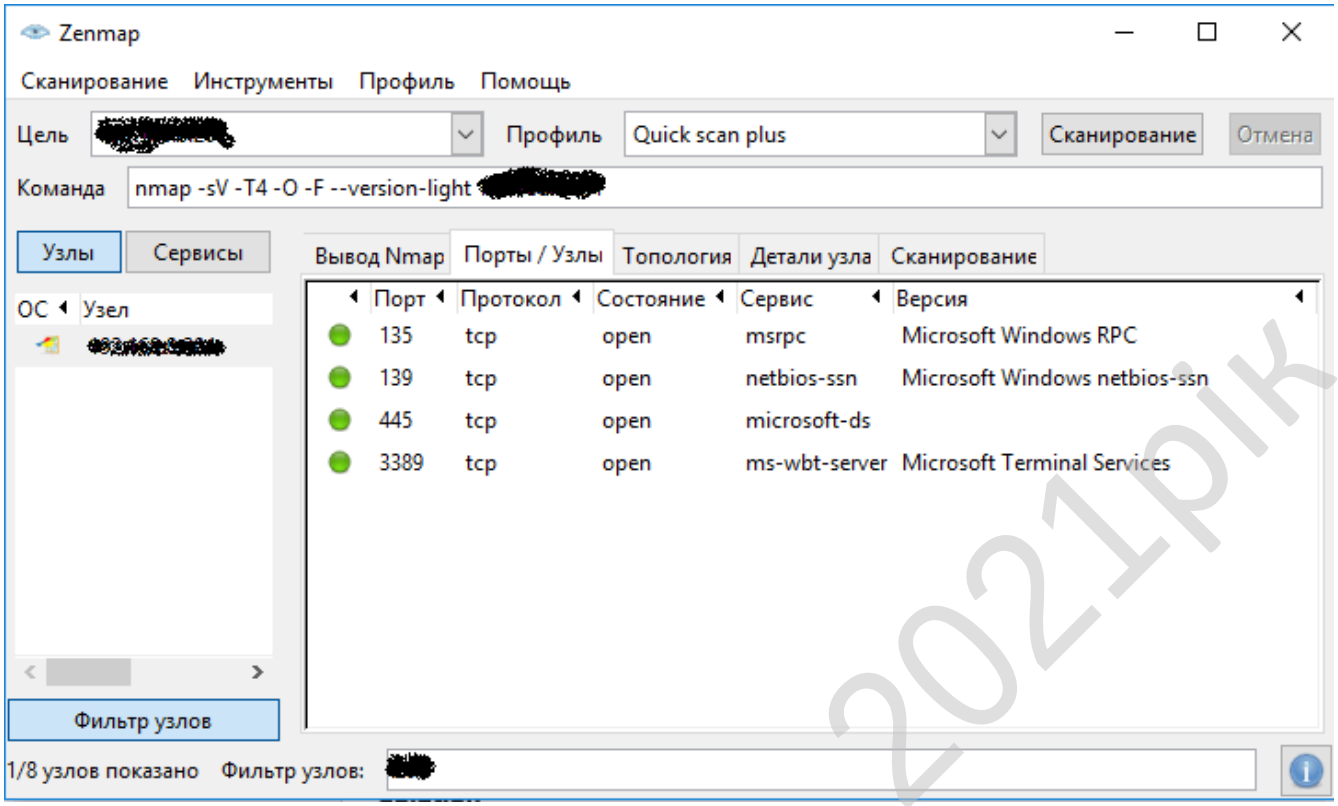


Рисунок 2.2 – Інтерфейс користувача Nmap

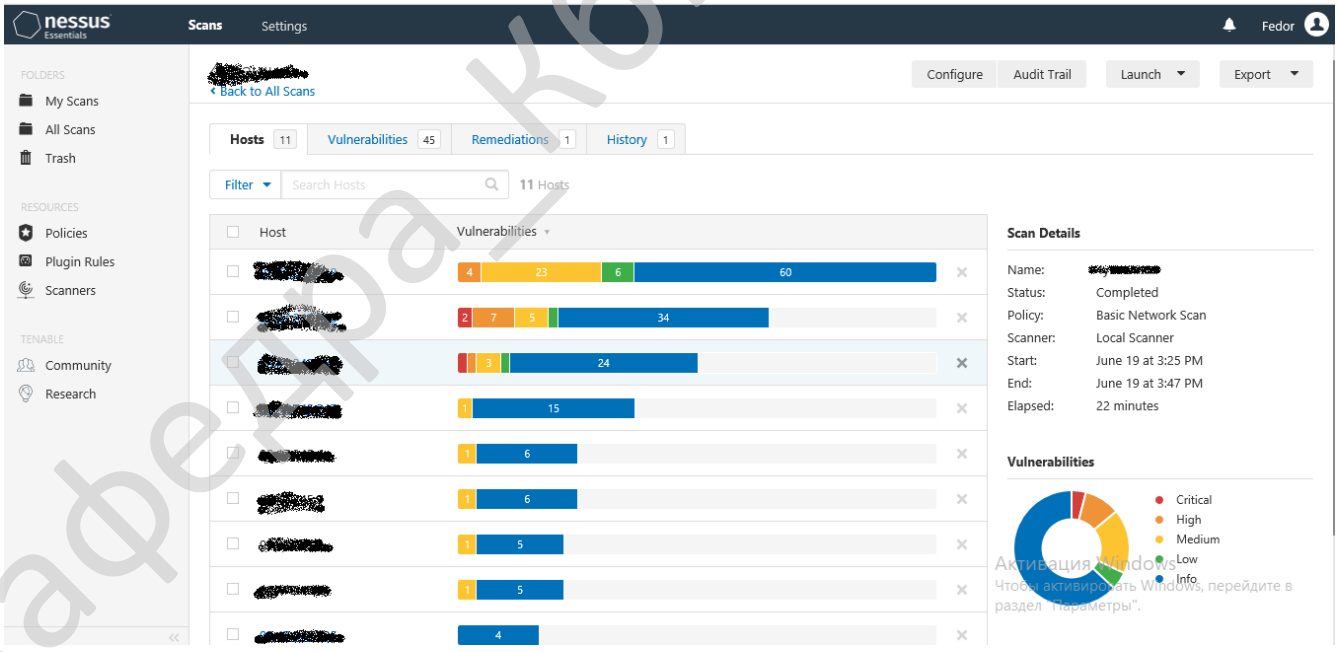


Рисунок 2.3 – Інтерфейс користувача Nessus

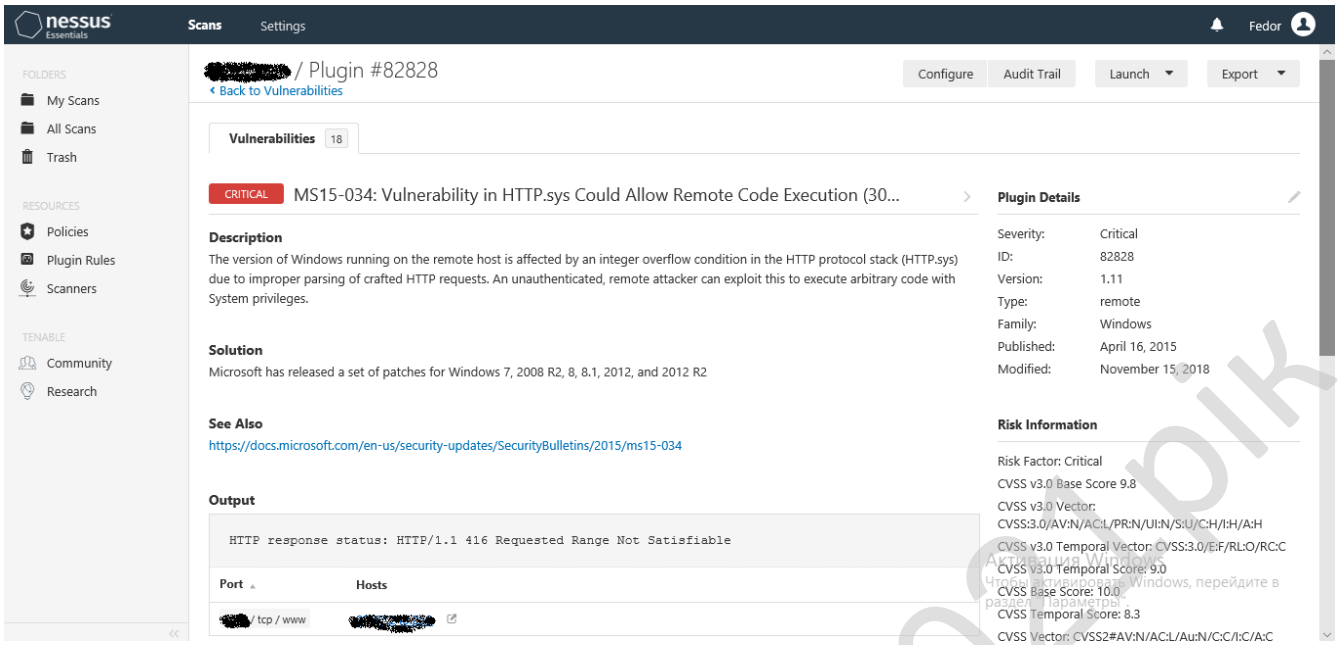


Рисунок 2.4 – Інтерфейс користувача Nessus

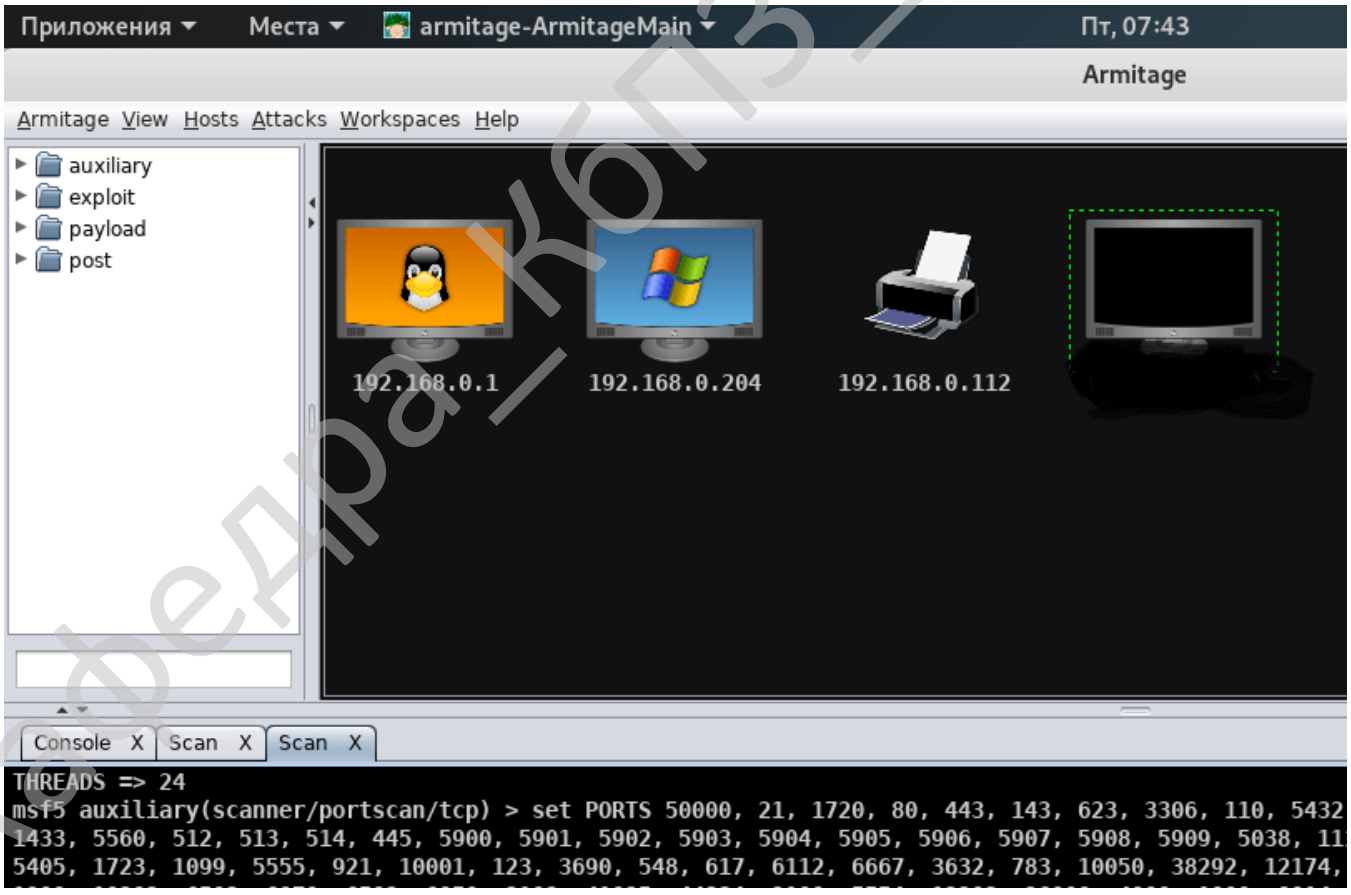


Рисунок 2.5 – Інтерфейс користувача Kali Linux

**Приклади пентестингу**

Які питання ставляться перед фахівцями, методика тестування на проникнення, як виглядає процес тестування? Розповімо про це на прикладі тестувань, проведених експертами:

– На замовлення приватної організації був укладений договір на проведення пентесту інфраструктури вищезгаданої організації. Дослідження проводилося з офісу експертної установи, використовувався зовнішня IP-адреса. У ході тестування на першому етапі виявлялися використовувані типи устаткування за допомогою програмного забезпечення Router Scan і Nmap. У процесі тестування периметра компанії були виявлені сервери й програмне забезпечення з уразливими компонентами, які дозволяють зловмисникові реалізувати атаки на одержання несанкціонованого доступу до серверів банку, інформації клієнтів і здійснити несанкціоноване проникнення у внутрішню мережу банку. Установлене, що вразлива версія Windows містить певний уразливий компонент, який використовується веб сервером, завдяки чому віддалений зловмисник може виконати віддалений код на сервері або викликати відмова в обслуговуванні вищевказаного сервера. Дана рекомендація з установки відновлень безпеки Windows.

– На замовлення приватної організації був укладений договір на проведення пентесту інфраструктури вищезгаданої організації. Дослідження проводилося з офісу експертної установи, використаний зовнішній IP-адреса. У ході тестування на першому етапі виявлялися використовувані типи устаткування за допомогою програмного забезпечення Router Scan і Nmap. Виходячи з отриманої за допомогою сервісу Whois інформації, резервний IP-адреса належить пулу адрес провайдера, що надає послуги організації – замовникові. Послуга для даного пулу поставляється по стандарту ADSL. Даний стандарт зв'язку є застарілим і несе погрозу відмови в обслуговуванні. Однак, застосування даного стандарту зв'язки для резервного каналу припустиме. На наступному етапі проведене сканування засобами Zmap по черзі всіх наданих адрес на наявність відкритих портів для можливостей подальшого пошуку уразливостей. У

результаті аналізу виявлений порт 9091, відкритий по протоколу TCP. Інших відкритих ресурсів не ідентифіковане. Інші порти мають статус closed, filtered або open|filtered, що робить їхнє подальше застосування в тестуванні на проникнення недоцільним. Ідентифікація використовуваних сервісів на наданих замовником IP-адресах не виконана, тому що тестуєма інфраструктура носить закритий характер. Для забезпечення повноти тестування на проникнення й аналізу уразливостей, був застосований засіб Armitage і база даних уразливостей Metasploit. У ході перевірки уразливостей на ресурсах Замовника не виявлене.

## 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 додатковими 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 формі 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 трохи 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 додаткової 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 для проведення пентесту IT 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

У даному розділі мова йтиме про перший крок у зборі інформації про компанію. А точніше про оцінку безпеки IT інфраструктури компанії на підставі даних з її сайтів і DNS записів за допомогою тестування на проникнення, називаному «пентест».

Починаємо процес пентесту зі збору загальної інформації про об'єкт: які сервіси використовуються, скільки серверів і які в них адреси і т.д.

Вихідною точкою найчастіше є домен 2-го рівня замовника, і він же є сайтом компанії. Спробуємо із цієї інформації вичавити максимум. Причому зробимо це в пасивному режимі, щоб системи моніторингу не зафіксували підозрілу активність і т.д.

Почати випливає з аналізу сайту й інформації на ньому. Якщо почнемо сканувати сайт, можемо спровокувати спрацьовування алертів системи захисту, тому скористаємося сервісом, який уже раніше просканувала цей сайт і має його зліпок. Так, такий сервіс існує й називається Google!

**Збір інформації в Google**

Як же Google може показати інформацію, що цікавить? Звичайний серфінг по сайтах перетворюється в професійний пасивний сканер, коли починаєш використовувати спеціальні команди в пошукових запитах – «дорки».

**Пошук ключових слів на сайті**

«ключове слово» site:domain.com

Застосування:

Корисно зібрати email адреси, вони ж логіни, які використовуються в компанії.

Найчастіше основний домен – це не тільки адреса сайту компанії, але й поштовий домен. Відповідно вводимо в пошуку Google

```
@domain.com site:domain.com
```

**Пошук файлів на сайті**

«ключове слово» filetype:pdf site:domain.com

Застосування:

Наприклад, знайти договори, там можливо є інформація про партнерів.

Договір filetype:pdf site:domain.com

Так само можемо з легкістю знайти robot.txt, у тому числі тестові версії, які лежать не в кореневій директорії

Договір filetype:txt site:domain.com

**Пошук в URL**

inurl:«що шукаємо»

Застосування:

Допомагає при пошуку sql ін'єкцій. Широко відомий пошуковий запит, з якого починається «полювання» за sql ін'єкціями

```
inurl:.php?id=
```

Велика кількість «дорок» 2017 року для пентесту можна знайти по посиланню, доступному наприкінці статті для зареєстрованих відвідувачів сайту.

**Аналіз DNS записів**

Ще один сервіс, який може чимало повідати про об'єкт пентесту – публічний DNS. Щоб зовнішній мир міг коректно працювати із сервісами компанії, необхідна публікація деякої інформації.

У терміналі Linux уводимо команду whois «домен».

Наприклад, whois aaa.com

```
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object
refer: whois.verisign-grs.com
 domain: COM
organisation: Verisign Global Registry Services
address: 12061 Bluemont Way
```

```
address: Reston Virginia 20190
       address: United States
contact: administrative
name: Registry Customer Service
organisation: Verisign Global Registry Services
address: 12061 Bluemont Way
address: Reston Virginia 20190
address: United States
phone: +1 703 925-6999
fax-no: +1 703 948 3978
       e-mail: info@verisign-grs.com
contact: technical
name: Registry Customer Service
organisation: Verisign Global Registry Services
address: 12061 Bluemont Way
address: Reston Virginia 20190
address: United States
phone: +1 703 925-6999
fax-no: +1 703 948 3978
       e-mail: info@verisign-grs.com
nserver: A.GTLD-SERVERS.NET 192.5.6.30 2001:503:a83e:0:0:0:2:30
nserver: B.GTLD-SERVERS.NET 192.33.14.30 2001:503:231d:0:0:0:2:30
nserver: C.GTLD-SERVERS.NET 192.26.92.30 2001:503:83eb:0:0:0:0:30
```

і так далі.

Представляється безліч корисної інформації. Найчастіше, у контактах вказується реальна поштова адреса адміністратора. Якщо вийде провести вдалу brute force атаку на цей логін, можна одержати повний контроль над усієї IT інфраструктурою.

Так само буває більша удача зустріти заповнене поле inetnum, у якому вказується діапазон IP адрес, що належать компанії

```
inetnum 95.200.118.0 – 95.200.119.255
```

Не забуваємо про таку важливу інформацію, як список піддоменів. Якщо не вдається пройти основний домен ( 2-го рівня), сервіси на які ведуть піддомени, як правило, виявляються менш захищені.

Ще одним джерелом пасивного збору інформації є веб сервіси, що надають whois інформацію. Такі як 2ip. Крім того, що одержали раніше, також

можна визначити навіть CMS, на якій працює сайт. Ця інформація зазначена в розділі Система Керування сайтом (CMS).

**Пошук «живих» хостів**

Тепер перейдемо до активного збору інформації.

На попередньому етапі, якщо повезло, одержали пул адрес, які належать об'єкту. Знайдемо які використовуються в цей момент

```
fping -Asg 95.200.118.0/24 -r 2 >> adress.list
cat address.list | grep alive
```

Одержуємо список хостів, які відповіли на ping.

Далі можна почати вивчення хостів із цього списку командою nmap.

**Скануємо мережу через nmap**

Швидка перевірка стандартних портів

```
nmap -ss «IP адреса або підмережа»
```

ss означає “тиху” перевірку. Є ймовірність, що системи моніторингу не помітять сканування.

Перевірка всіх портів

```
nmap -ss «IP адреса або підмережа» -p1-65536
```

Визначення версії ОС

```
nmap -O «IP адреса або підмережа»
```

**Автоматизуємо процес аналізу**

Наостанку, хотілося б освітити зручний інструмент, який дозволить автоматизувати всі вищеописані методи розвідки й навіть більше.

Застосунок називається recon-ng.

Посилання для завантаження https://bitbucket.org/Lanmaster53/recon-ng.git

Передвстановлен в Kali Linux.

Являє собою модульний фреймворк, схожий на Metasploit Framework (MF), але призначений по більшій частині для пасивного аналізу веб ресурсів. Механізм керування аналогічний MF.

Основні модулі, список яких доступний для зареєстрованих відвідувачів, дозволить вам наочно оцінити придасться вам цей фреймворк чи ні.

### 3.2 Розробка структурної схеми

Тестування на проникнення може бути зовнішнім і внутрішнім. У якості одного з елементів зовнішнього пентесту виділяється соціальна інженерія.

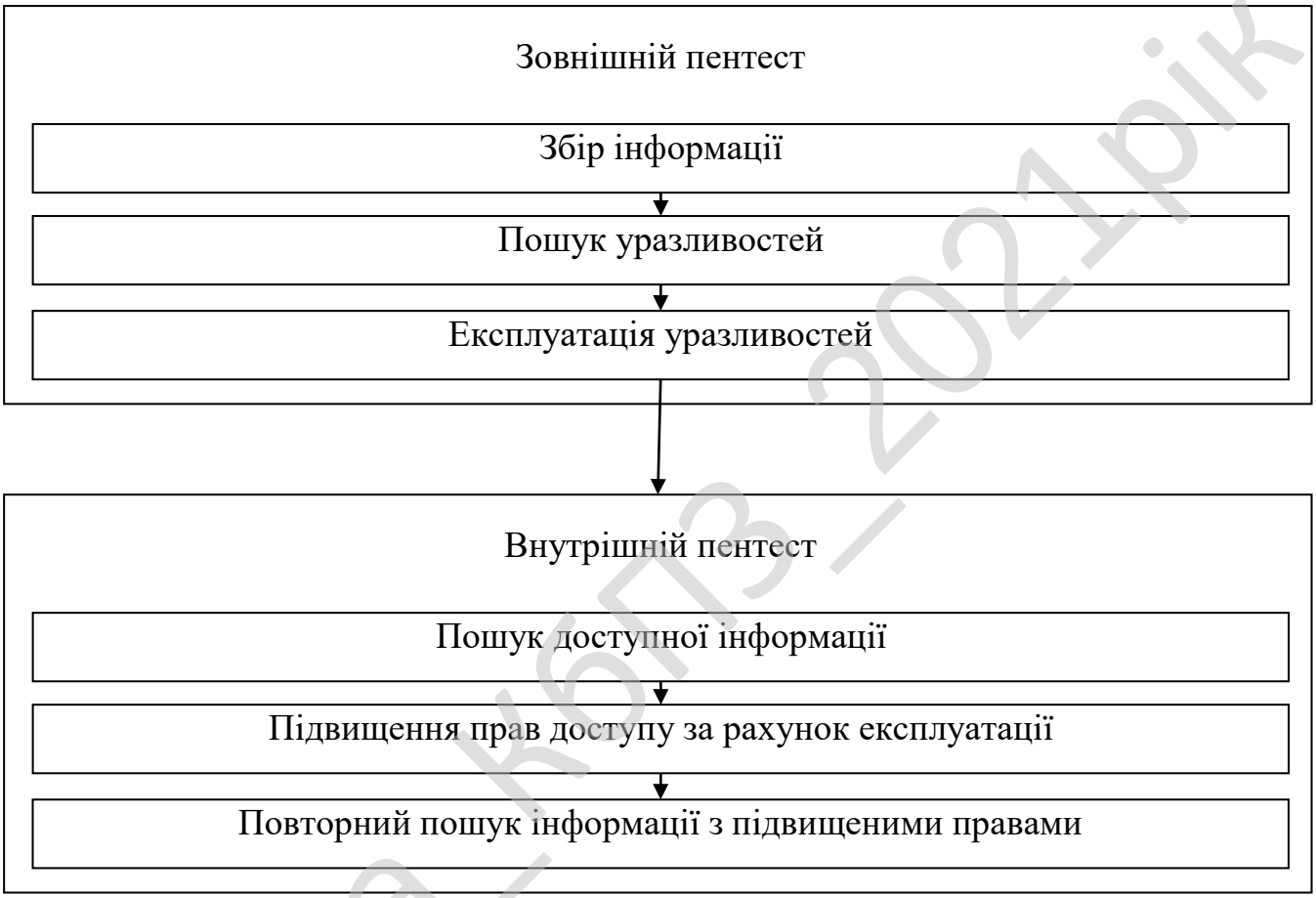


Рисунок 3.1 – Структурна схема системи

**Зовнішній пентест**

Зовнішній пентест моделює дії порушника, і припускає дії хакера зовні. Даний варіант пентесту складається з наступних етапів:

1. **Збір інформації**. Пентестера цікавлять, як правило, адреси зовнішнього периметра Замовника – web-сайти, сервери. При цьому аналізується як загальнодоступна інформація, інформація передана Замовником, так і, у ряді випадків, конфіденційна інформація, отримана від третіх осіб. Частина

відомостей або вся необхідна інформація може бути передана за договором Виконавцеві безпосередньо Замовником.

2. **Пошук уразливостей**. На даному етапі застосовуються різні сканери, які, у загальному випадку, можуть доповнювати друга-друга. З найбільш відомих – Nessus, Xspider, Openvas, Maxpatrol.

3. **Експлуатація уразливостей**. На даному етапі від пентестерів потрібен творчий підхід – адже етичному хакерові не можна допустити збиток інфраструктурі Замовника, тому даний етап у ряді випадків пропускається. Експлуатація уразливостей відбувається тільки за згодою Замовника за умови не допустити порушення функціонування систем.

Окремим напрямком зовнішнього пентесту є **соціальна інженерія**. Найбільше масово використовуваним її видом інструментом є розсилання заражених електронних листів. За результатами розсилання аналізується не тільки робота антивірусних засобів і інструментів моніторингу трафіка, але й головний фактор – відповідальність співробітників при реагуванні на одержання подібних листів. Сам факт відкриття листа вже може бути предметом внутрішнього розгляду або підставою для навчання співробітників базовим вимогам ІБ.

**Внутрішній пентест**

Внутрішній пентест проводиться з метою перевірки інфраструктури замовника на наявність уразливостей зсередини. При цьому пентестер одержує доступ усередину із правами рядового користувача (або взагалі без прав), але з фізичним доступом до мережі або робочого місця. При цьому тестування може проводитися як віддалено, за допомогою VPN-підключення, або із присутністю пентестера на території Замовника.

Оскільки внутрішні порушники становлять основну масу в числі інцидентів ІБ, саме внутрішній пентест становить особливий інтерес. При цьому немає необхідності шукати крапки входу – адже моделюється ситуація, коли хакер уже проникнув усередину. Тому його дії складаються з наступних етапів:

– Пошук доступної інформації (найчастіше рядовому користувачеві доступна конфіденційна інформація, що перебуває на мережевому ресурсі, просто ярлик від ресурсу явно не виведений на робочий стіл)

– Підвищення прав доступу за рахунок експлуатації відомих уразливостей

– Повторний пошук інформації з підвищеними правами – залежно від погодженого із замовником ТЗ.

**Методика проведення пентесту**

Оскільки методика проведення пентестів регуляторами не визначена, опишемо основні нюанси, які слід урахувати при виконанні даного вимоги:

1. Пентест (pentest) може привести до відмов в обслуговуванні, у випадку, якщо виявлена й використана істотна уразливість. Зрозуміло, настання негативних наслідків залежить від кваліфікації виконавця, однак, найчастіше, це неминуче. У зв'язку із цим краще прямо обмовити в технічному завданні на пентест або застосовуваній методиці пентесту вимогу не експлуатувати виявлені уразливості.

2. Тестування на проникнення – досить широке поняття. Воно може містити в собі зовнішнє дослідження ( по методу чорного ящика), відкритий аналіз існуючої інфраструктури ( по методу білого ящика), змішані методи (сірий ящик), соціальну інженерію й інші розділи. У зв'язку із цим необхідно визначитися з обсягом тестування. Це прямо вплине на повноту виявлених уразливостей, але, у той же час, відіб'ється на строках і вартості.

3. У якості виконавця робіт може залучатися організація, що має ліцензію на технічний захист конфіденційної інформації.

– Вимога проводити пентест є нормативноюм і критерій достатності не встановлений, тому замовникові робіт необхідно визначитися з обсягом проведених робіт.

Очевидно, що найбільш повним тестуванням буде змішане – відкритий аналіз і пентест по чорному ящикові. При цьому слід домовитися про незалежність експертів, що реалізують дані підходи.

Однак, як було відзначено вище, подібне дослідження може бути надзвичайно дорогим і тривалим. Тому тестування можна проводити не на всій інфраструктурі, а на вибірці – наприклад, затвердивши перелік серверів, сервісів, робочих станцій або інформаційних систем, використовуваних банком.

При виборі напрямку для тестування (зовнішнє або внутрішнє) по чорному ящикові – слід пам'ятати, що переважна більшість порушень доступу до інформації – наслідок внутрішніх порушників.

Методики пошуку уразливостей суттєво різняться по інструментарію й алгоритмам, і залежать від особливостей інфраструктури. Відомості про кроки пошуку приводяться у звіті про пентесті.

**Які результати пентесту можна одержати?**

У результаті дій пентестера, виявляються недоліки інформаційної безпеки, які можуть полягати в наступному:

– Некоректне розмежування прав доступу.
– Слабкі паролі.
– Неактуальні версії програмного забезпечення.
– Погана сегментація обчислювальної мережі.
– Та ін.

### 3.3 Розробка функціональної схеми

Остання пара років була багата на події, які різко підвищили інтерес суспільства до теми хакерських атак. Скандал зі зломом систем демократичної партії США, виведення з ладу енергетичних систем інфраструктури Міністерства фінансів і казначейства України, віруси-здирники, що вже не тільки шифрують файли, але й блокувальні роботу промислового й медичного устаткування, MIRAL, гігантський ботнет з побутових пристроїв, що залишив без зв'язку половину США й Ліберію, зловмисники, що масово потрошать банки. Під ударом

навіть SWIFT. Хакери з кіношних гіків стали частиною реальності мільярдів людей.

Цілком природно, що бізнес сьогодні в першу чергу вкладає ресурси в практичну безпеку на противагу формальному виконанню вимог регуляторів мінімальними засобами. І також природно його бажання перевірити, наскільки ефективно побудована система безпеки захищає від мережевих акул.

Цього разу ми розв'язали зосередитися винятково на практичних моментах інформаційної безпеки (ІБ), пов'язаних з комп'ютерними атаками й безпосереднім захистом від них. Для злому у виконанні «білих капелюхів», тобто фахівців, що легально імітують дії зловмисників, використовується термін «тестування на проникнення» (penetration test, pentest). За цим терміном ховаються відразу кілька напрямків дослідження захищеності, і в кожному з них працюють свої вузькі фахівці. У статті ми розберемося, що таке пентест, навіщо він потрібний і де проходить границя між атакою хакера й тестуванням на проникнення.

Пентест по суті своєї – це один з видів аудит а ІБ. І в цьому його головну відмінність від реального злому. Хакер шукає саму коротку дорогу до контролю над системами жертви. Якщо на периметрі найшлася діра, зловмисник зосереджує на закріпленні й розвитку атаки всередину. А пентестер, якому замовили зовнішнє тестування мережі, повинен скрупульозно обстежити хост за хостом, навіть якщо вже знайдена ціла купа дір. Якщо хости однотипні (наприклад, 1000 однакових робочих станцій), дослідник, звичайно, може зробити контрольну вибірку, але пропускати принципово відмінні системи неприпустимо. Напевно, для замовника це найпростіший спосіб визначити неякісний пентест.

Пентест не заміняє повноцінний аудит ІБ. Для нього характерний вузьконаправлений погляд на досліджувані системи. Пентест по суті своєї має справа з наслідками, а не із причинами недоліків ІБ. Навіщо ж його взагалі проводити? Коли промисловість випускає новий зразок військової техніки, інженери ретельно прораховують властивості броні, характеристики озброєння,

але на військовому прийманні техніку однаково викочують на полігон, обстрілюють, підривають і т.д. Експеримент – критерій істини. Пентест дозволяє зрозуміти, чи так добре, як ми думаємо, вибудувані в нас процеси ІБ, чи надійні системи захисту, чи вірна конфігурація на серверах, чи розуміємо ми шлях, по якому піде реальний хакер. Таким чином, може зложитися враження, що пентест необхідний компаніям, які вже ґрунтовно вклалися в ІБ. У теорії це так, але на практиці найчастіше зовсім інакше.

Пентест складається з наступних елементів:

Пентест = Дослідження + Звіт і рекомендації + Шоу

**Дослідження** – це сама очевидна частина пентесту. Як у кіно: дивні хлопці в худі вночі громлять IT-оборону. На ділі найчастіше все трохи прозаїчніше, зате даний образ дозволяє пентестерам не дотримувати корпоративного дресс-код.

**Звіт** – звичайно не сама улюблена частина роботи для пентестерів, але вона критично важлива. Замовник робіт повинен одержати детальний опис усіх успішних і неуспішних спроб проникнення, зрозумілий опис уразливостей і, що дуже важливо, рекомендації з їхнього усунення. До останньої частини раціонально залучати профільних фахівців з ІБ, тому що знати, як зламати, зовсім не означає знати, як правильно й безпечно це поправити в реальності корпоративної IT-інфраструктури.

І останній компонент, заради якого найчастіше й організує весь пентест, – це **шоу**. Такий аудит на порядок перевершує будь-який іншою по наочності, особливо для непрофесіоналів. Це кращий спосіб продемонструвати недоліки ІБ керівництву компанії в доступній для неспеціалістів формі. Коротке (на парі сторінок) Executive Summary зі сканом паспорта CEO, титульного аркуша конфіденційного звіту й бази клієнтів може принести для ІБ у компанії більше користі, чому весь 200 сторінковий звіт, що йде далі. Саме тому найчастіше пентест замовляють компанії, де ІБ до цього доладно не займалися, і бізнес, а найчастіше й IT, не розуміють серйозності існуючих ризиків.

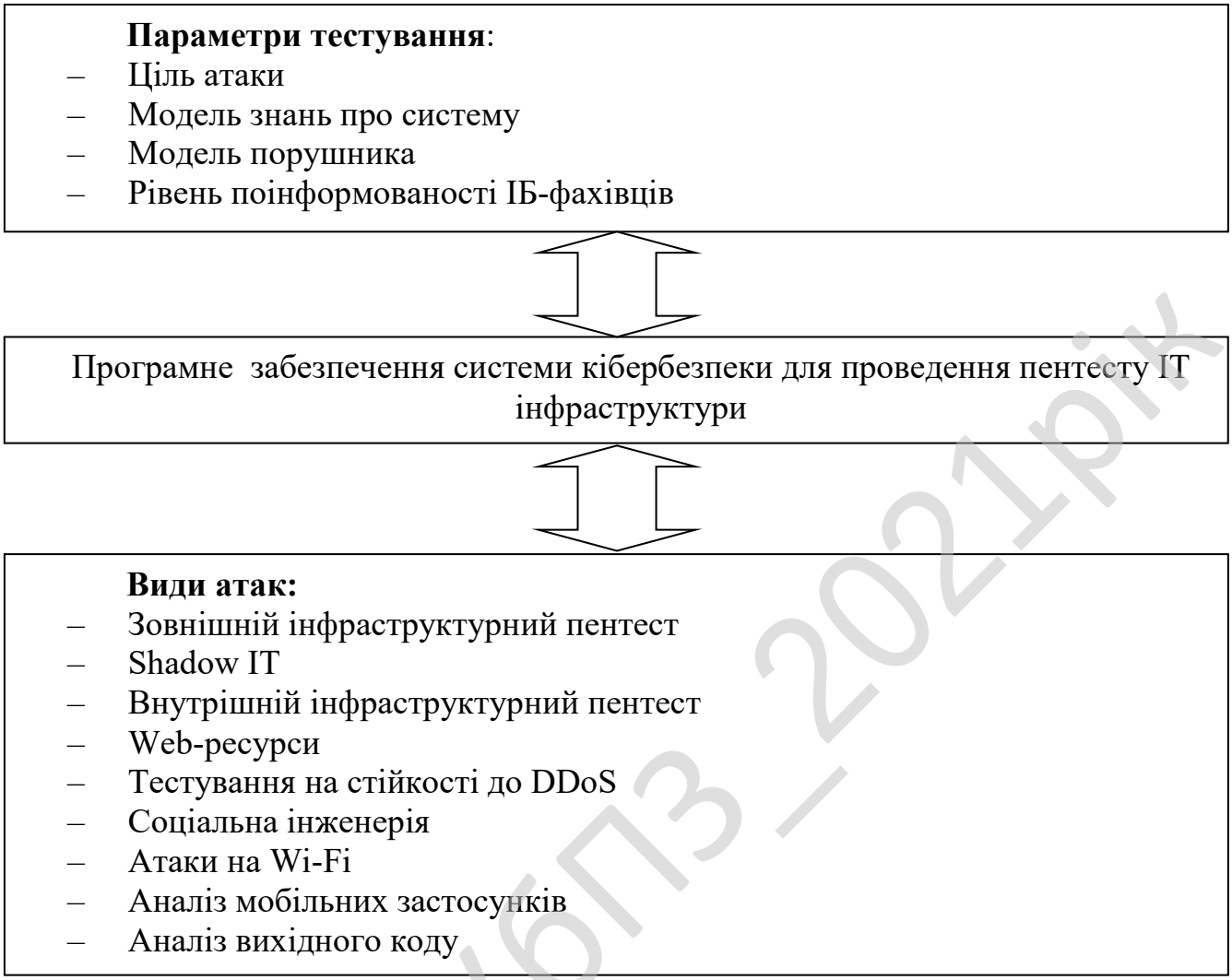


Рисунок 3.2 – Функціональна схема системи

**Параметри тестування**

Пентести можна класифікувати всілякими способами. Зупинимося тільки на тих, які мають практичну цінність при конфігуруванні пентесту під себе.

Поставлена замовником робіт **ціль атаки** може сильно відрізнятися від пентесту до пентесту. Під «просто зламайте нас» звичайно мається на увазі захват контролю над IT-інфраструктурою (права адміністратора домену, мережевого устаткування), компрометація бізнес-систем і конфіденційної інформації. А бувають вузьконаправлені пентести. Наприклад, у рамках сертифікації по вимогах безпеки карткових даних PCI DSS метою щорічного обов'язкового пентесту є компрометація саме карткових даних. Тут у перший же день робіт

мережа банку може бути повністю захоплена, але, якщо останній бастіон із секретними даними не впаде, організація успішно пройде перевірку.

**Модель знань про систему** визначає стартову позицію пентестера. Від повної інформації про систему (White box) до повної її відсутності (Black box). Найчастіше виділяють і середній варіант (Grey box), коли, наприклад, пентестер імітує дії непривілейованого користувача, що має деякі дані про систему. Це може бути рядовий клерк, компанія-партнер, клієнт із доступом в особистий кабінет і т.п. White box – це скоріше аудит, а не класичний пентест. Застосовується в тому випадку, коли потрібно детально вивчити захищеність на вузькій ділянці. Наприклад, перевіряється новий клієнтський портал. Дослідникові надається вся інформація із системи, найчастіше вихідний код. Це допомагає детально вивчити систему, але чи ледь імітує реальні атаки. Замовники Black box пентесту прагнуть одержати повну імітацію атаки хакера, який не має інсайдерської інформацією про систему.

Модель знань сильно перетинається з поняттям **модель порушника**. Хто нас атакує: зовнішній хакер, інсайдер, адміністратор? Розподіл це дуже умовно. Компрометація робочої станції рядового користувача або підрядника з технічної точки зору моментально перетворює зовнішнього хакера у внутрішнього порушника.

**Рівень поінформованості ІБ-фахівців** визначає, хто знає про проведення робіт і наскільки докладно. Найчастіше, крім техніки, тестується й персонал, тому роботи координує директор по ІБ або ІТ, а адміністратори вважаються, що борються з реальними хакерами, якщо, звичайно, взагалі зауважують атаку. Такі кібернавчання дозволяють оцінити не тільки наявність уразливостей у системах, але й зрілість процесів ІБ, рівень взаємодії між підрозділами й т.п. Повною протилежністю є імітація дій зловмисника з метою навчання систем захисту. У цьому випадку пентестер працює на невеликій ділянці, а адміністратори фіксують реакцію засобів захисту й ІТ-систем, коректують налаштовування, готовлять правила для SIEM і т.п. Наприклад, імітується ситуація, коли хакер уже

проникнув у закритий сегмент. Як він буде підвищувати свої привілеї в системах? Пентестер по черзі відпрацьовує всі відомі йому вектори атак для максимально повного навчання систем безпеки.

**Види атак**

Скільки пентестерів, стільки класифікацій типів атак. Нижче я приведу класифікацію базових атак, яку використовуємо ми. Звичайно, самий повний пентест – це атака по всіх можливих напрямках. Але обмеження бюджету, часу, скоупа й завдань пентесту змушують вибирати.

**Зовнішній інфраструктурний пентест** – аналіз мережевого периметра з Інтернету. Пентестер намагається скомпрометувати доступні мережеві сервіси й по можливості розвити атаку всередину мережі. Багато уважають, що це і є імітація реальної атаки, спрямованої на проникнення в мережу компанії ззовні. На ділі зловмисники сьогодні в 80-90% випадків долають мережевий периметр із використанням методів соціальної інженерії. Не потрібно ломитися в кріпосні стіни, якщо під ними є чудовий підкоп. Однак часто дірки бувають і отут. Наприклад, недавно ми проводили роботи для великого авіаційного заводу, у рамках яких ще на етапі автоматичного аналізу сканер підібрав пароль до системи віддаленого керування АСУ ТП. Недбалість підрядника, що забув відключити дистанційний доступ, дозволяла хакерові підняти тиск у трубопроводах з технічними рідинами на порядок. З усіма, що випливають у прямому й переносному значенні.

Такий пентест як огляд у дантиста: краще проводити його регулярно, щоб попереджати проблеми на ранніх стадіях.

**Shadow IT**

Часте проникнення проходить із використанням систем, які випадають із поля зору IT. Усі сервери на периметрі оновлені, а про IP-Телефонію або систему відеоспостереження забули. І хакер уже усередині. Для такої інфраструктури, що випала з поля зору адміністраторів, є спеціальний термін – Shadow IT. По оцінці

Gartner, до 2022 року до третини всіх зломів будуть проходити з використанням Shadow IT. На наш погляд, це цілком реалістична оцінка.

Наприклад, одного разу пентестер знайшов на ідеально захищеному периметрі банку неоновлені системи колл-центру, через які за 2 дня вдалося повністю скомпрометувати всі основні банківські АС. Виявилося, що за них відповідав не департамент IT, а телефоністи. В іншому випадку крапкою входу для пентесту була мережа секретарів на ресепшн, повністю ізольована від корпоративної. Яке ж був подив замовника робіт, коли через пару днів пентестер відзвітувався про повний захват мережі. Йому вдалося зламати неоновлений принтер, залити на нього шелл і одержати доступ в VLAN керування принтерами. Скомпрометувавши їх усі, пентестер одержав доступ в усі офісні сегменти компанії.

**Внутрішній інфраструктурний пентест** імітує дії інсайдера або зараженого вузла усередині мережі. Мережа повинна будуватися так, щоб компрометація окремих робочих станцій або серверів не приводила до повного падіння оборони. На ділі більш ніж у половині випадків з нашої практики від прав «доступ до мережевої розетки» до «адміністратор домену» проходить не більш одного робочого дня.

Мережа компанії може бути дуже великий, тому в ряді випадків замовникові слід чітко визначити для пентестера мети атак. Наприклад, доступ до SAP і фінансовим документам із грифом «Конфіденційно». Це дозволить рациональнее витратити час пентестера й імітувати реальну замовлену хакерську атаку.

**Web-ресурси** представляють окремий мир з погляду пентесту з величезним набором різних технологій і специфічними атаками. Ясно, що під вебом можна розуміти що завгодно, що має доступ у мережу. Тут ми маємо на увазі різні web-сайти, портали й специфічні Api-інтерфейси, доступні з мережі. Практика показує, що в середньому для компанії аналіз усього її мережевого периметра займає менше часу, чому одного web-сайту, особливо якщо там є якісь

інтерактивні елементи, особистий кабінет і т.п. Цей напрямок переживає справжній бум, у першу чергу через розвиток електронного бізнесу банками й масового виходу ритейла в Інтернет.

Основними результатами атаки на web-ресурс звичайно є компрометація даних із СУБД і можливість атаки на клієнтів (наприклад, різні види XSS перебувають на сайтах кожного другого банку). Ледве рідше компрометація web-сервера дозволяє проникнути в саму мережу компанії, але найчастіше, якщо шукані дані вже скомпрометовані, це може й не знадобитися зловмисникові.

При аналізі web важливо перевірити не тільки технічну частину, але й саму логіку роботи й реалізації бізнес-функцій. Дотепер іноді можна одержати знижку в 99% в інтернет-магазині або скористатися чужими бонусними балами, злегка модифікувавши рядок запиту до сервера в адресному рядку.

Атаки на web можуть бути здійснені й усередині мережі, адже про безпеку внутрішніх ресурсів звичайно не замислюються, але на ділі більшість хакерів спочатку атакує інфраструктуру, тому що це самий короткий шлях до адміністратора домену. За web беруться, коли ніщо інше не допомогло або коли потрібно пробратися в ізольовані мережеві сегменти.

Ріст інтересу до **тестування на стійкості до DDoS** особливо помітний в останні парі років. Інформація про великі атаки постійно з'являється в пресі, але ними справа не обмежується. У сегменті роздрібного інтернет-ритейла, наприклад, у піки продажів (перед святами) атаки йдуть практично безупинно. Що робити із примітивними атаками, спрямованими на вичерпання каналу зв'язки або ресурсів серверів шляхом відправлення величезних обсягів трафіка, у цілому ясно. Цікавіше вивчити стійкість ресурсу до атак рівня застосунку. Навіть один клієнт, що генерує порівняно невелике число специфічних запитів до web-сайту, може вивести його з ладу. Наприклад, специфічні запити в поле пошуку по сайту можуть повністю покласти back-end.

**Соціальна інженерія**, тобто використання для злому людської неуважності, безтурботності або ненавченості, сьогодні стала самим популярним способом проникнення в мережу компанії.

Більше того, існує думка, що від цього лома немає приймання. Цей термін поєднує величезне число технік облудних повідомлень, що включають розсилання, поштою, телефонне й особисте спілкування для одержання доступу на об'єкт або до систем, розкидання в офісу компанії-жертви флешок зі шкідливим вкладенням і багато чого іншого.

**Атаки на Wi-Fi** помилково відносять до внутрішнього пентесту. Якщо ваш смартфон не ловить корпоративний Wi-Fi за межами прохідний, це не дає гарантій того, що зловмисники не зможуть уводити, увести до ладу нього дотягтися.

Спрямована антена з ebay вартістю 100$ дозволяла нам проводити роботи з відстані більш кілометра від крапки доступу. У рамках пентесту Wi-Fi не завжди розглядається як крапка проникнення в мережу. Частіше він використовується для атаки на користувачів. Наприклад, пентестер паркується в прохідній підприємства до почала робочого дня й розвертає мережу з тим же іменем (SSID), що в корпоративного Wi-Fi.

Пристрою в сумках і кишенях співробітників намагаються приєднатися до знайомої мережі й передають для автентифікації в ній… доменні логін і пароль. Потім пентестер використовує ці витоки для доступу до пошти користувачів, VPN-серверам і т.д.

**Аналіз мобільних застосунків** для зловмисника спрощується тим, що їх легко скачати з магазину й детально досліджувати в «пісочниці», відновивши вихідний код.

Для звичайних web-ресурсів про таку розкіш доводиться тільки мріяти. Тому даний вектор атаки так популярний сьогодні. Мобільні клієнти зараз дуже поширені не тільки в банків і ритейла.

Їх випускають усі підряд, а про безпеку думають у саму останню чергу.

Умовне дослідження мобільного застосунку можна розділити на 3 компоненти: аналіз відновленого вихідного коду на наявність дір у безпеці, дослідження застосунку в «пісочниці» і аналіз методів взаємодії застосунку із сервером (зміст пакетів, API, уразливості самого сервера).

У нас недавно був кейс, коли API серверної частини мобільного банківського застосунку працювало так, що можна було сформувати пакет, що викликає переклад довільної суми грошей з будь-якого рахунку в банку на будь-який інший рахунок. І це було не дослідження перед стартом застосунку – воно вже давно було в продуктиве.

Багато шахрайських схем сьогодні також реалізуються за допомогою мобільних застосунків, тому що про боротьбу із фродом забувають ще частіше, чим про ІБ.

**Аналіз вихідного коду** не цілком коректно вважати пентестом, особливо якщо замовник передає вихідні коди на дослідження у відкритому виді. Це скоріше аудит безпеки застосунку по моделі «білого ящика».

Однак ці роботи найчастіше проводяться разом з пентестом для забезпечення більш високого рівня виявлення уразливостей, тому про них варто згадати тут. Пентест дозволяє підтвердити або спростувати недоліки, знайдені в рамках аналізу коду (адже в конкретній інфраструктурі далеко не всі проблеми безпеки реально можуть бути проексплуатовани).

Це суттєво знижує число ложнопозитивних знахідок, якими грішить аналіз коду, особливо автоматизований. У той же час у результаті аналізу коду найчастіше перебувають діри, про яких пентестер не догадався.

По нашому досвіду, найчастіше замовляють аналіз коду мобільних застосунків і web-сервісів, як найбільш підданих атакам.

**Обмеження пентесту**

Основними обмеженнями, які відрізняють пентест від реальної атаки, ускладнюючи роботу «білим капелюхам», є кримінальний кодекс і етика. Наприклад, пентестер найчастіше не може атакувати системи партнерів

замовника, домашні комп'ютери співробітників, інфраструктуру операторів зв'язки, він не використовує в соціальній інженерії залякування, погрози, шантаж, підкуп і інші дуже ефективні приймання злочинців.

Тим більше переконливі результати успішного проникнення в рамках «чистого» пентесту. Якщо ваш пентестер переступає закон у рамках робіт, десять раз подумайте, чи коштує підпускати такої людину до ваших ключових систем.

Пентест, як і медогляд, більшість стандартів рекомендує проходити не рідше разу в рік.

При цьому непогано періодично міняти фахівців, які проводять роботи, щоб уникнути замилювання погляду й оцінити захищеність із різних ракурсів. Адже в будь-якого фахівця або команди тією чи іншою мірою формується деяка спеціалізація.

Пентест – це й час, і витрати, і стрес для безпечників, але складно знайти більш наочний і наближений до реальності спосіб оцінки захищеності IT-інфраструктури. У кожному разі краще, щоб діру знайшов найнятий за договором фахівець, чому хакер. Адже перше часто закінчується для служби ІБ виділенням додаткових засобів на безпеку, а друге – пошуком нової роботи.

### 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. 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.

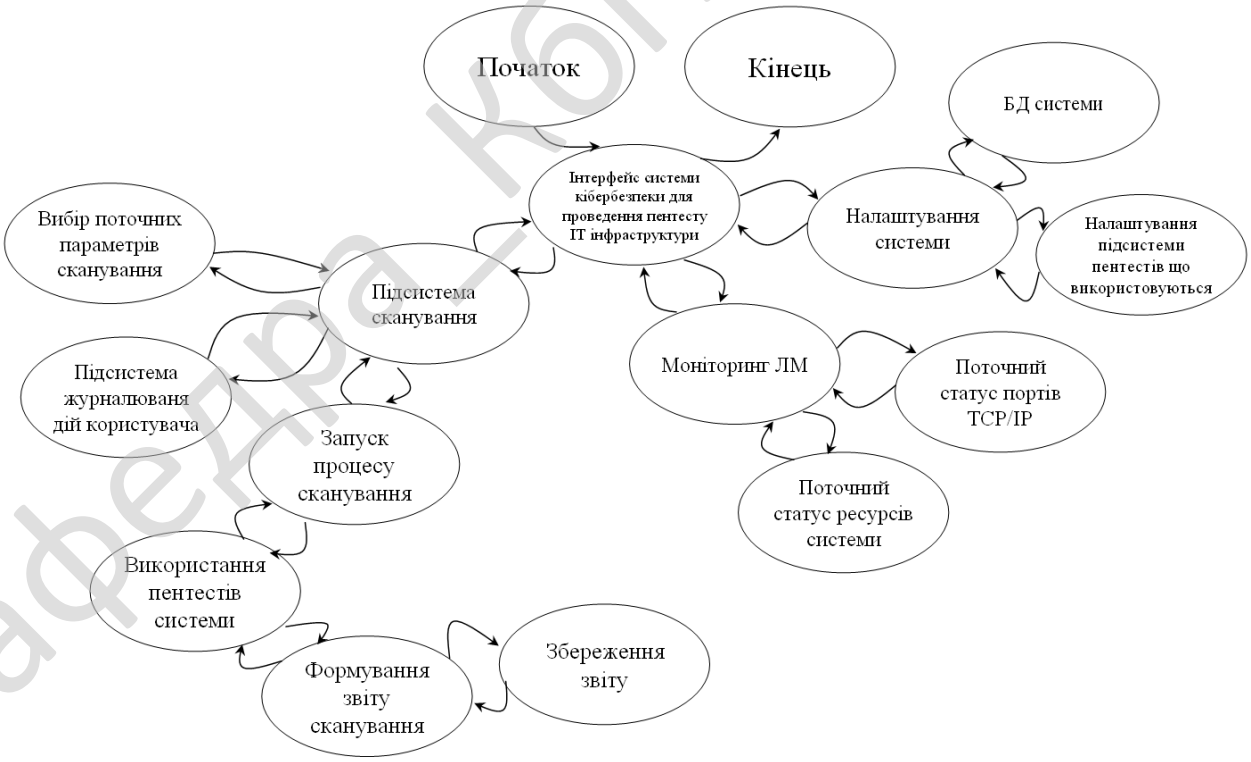


Рисунок 3.3 – Діаграма взаємодії процесів

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

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1. Основний алгоритм дій наступний:

– Виведення інтерфейсу системи кібербезпеки.

– Сканування поточного статусу системи.

– Формування пакету сканування.

– Виділення параметрів поточного сканування.

– Підпрограма крокового сканування.

– Об’єкт пентесту сформовано.

– Виведення інформації про знайдення об’єкту пентесту.

– Тип атаки у зеленій зоні.

– Сформувати звіт об’єкту.

– Формування звіту.

– Виведення повного звіту сканування.

– Запит формування поточного звіту ресурсів системи.

– Запуск контролю ресурсів системи.

– Підпрограма сканування ресурсів системи.

– Виведення звіту ресурсів системи.

На рисунку 4.2 зображено роботу підпрограми в якій можна побачити що алгоритм послідовних дій виконується наступним чином:

– Сканування з використанням пентесту системи.

– Запит прав адміністратора.

– Встановлення прав локального адміністратора.

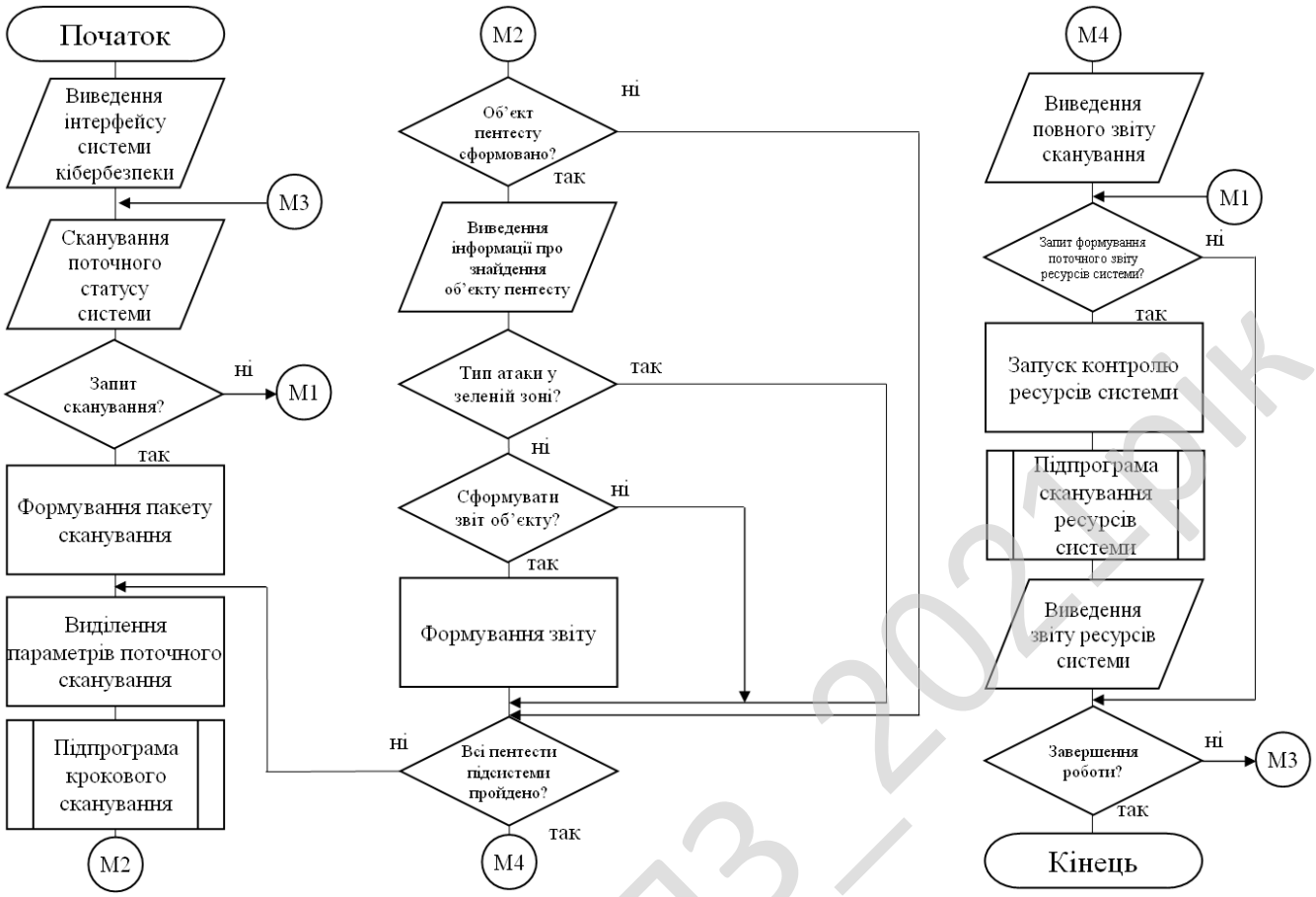


Рисунок 4.1 – Блок схема основної програми

– Пошук активних об’єктів пентесту.

– Застосування  евристичного аналізу підсистеми.

– Застосування параметрів користувача.

– Формування пакету сканування.

– Активний об’єкт знайдено.

– Сканування та визначення типу.

Розглянемо вихідний код функції роботи з підрахуванням наявних пентестів у локальній базі.

```
function CountPENTEST():Integer;
var
  CB: Integer;
begin
CB := 0;
if not FileExists(ExtractFilePath(ParamStr(0)) + 'PentestBase.dat') then
```

```
begin CountBase := 0; end
else begin
AssignFile(BaseFile, ExtractFilePath(ParamStr(0)) +  ' PentestBase.dat ');
reset(BaseFile);
readln(BaseFile, FileBase);
readln(BaseFile, FileBase);
While FileBase <> '}' do
begin
CB := CB + 1;
readln(BaseFile, FileBase);
end;
CloseFile(BaseFile);
CountBase := CB;
end;
end;
```

Розглянемо вихідний код процедури пошуку файлів пентестів на локальній системі.

```
procedure FindPENTESTfile(Dir:String);
var
  SR:TSearchRec; FindRes: Integer; EX : String;
begin
FindRes:=FindFirst(Dir+'*.*',faAnyFile,SR);
While FindRes=0 do
begin
if ((SR.Attr and faDirectory)=faDirectory) and ((SR.Name='.')or(SR.Name='..'))
then
begin
FindRes:=FindNext(SR);
Continue;
end;
if ((SR.Attr and faDirectory)=faDirectory) then
begin
Scan(Dir+SR.Name+'\');
FindRes:=FindNext(SR);
Continue;
end;
Ex := ExtractFileExt(Dir+SR.Name);
if     (LowerCase(Ex)   =   LowerCase('.exe'))   or   (LowerCase(Ex)   =   '.com')   or
(LowerCase(Ex) = LowerCase('.dll')) then
begin
      FindAndKill(Dir + SR.Name);
```

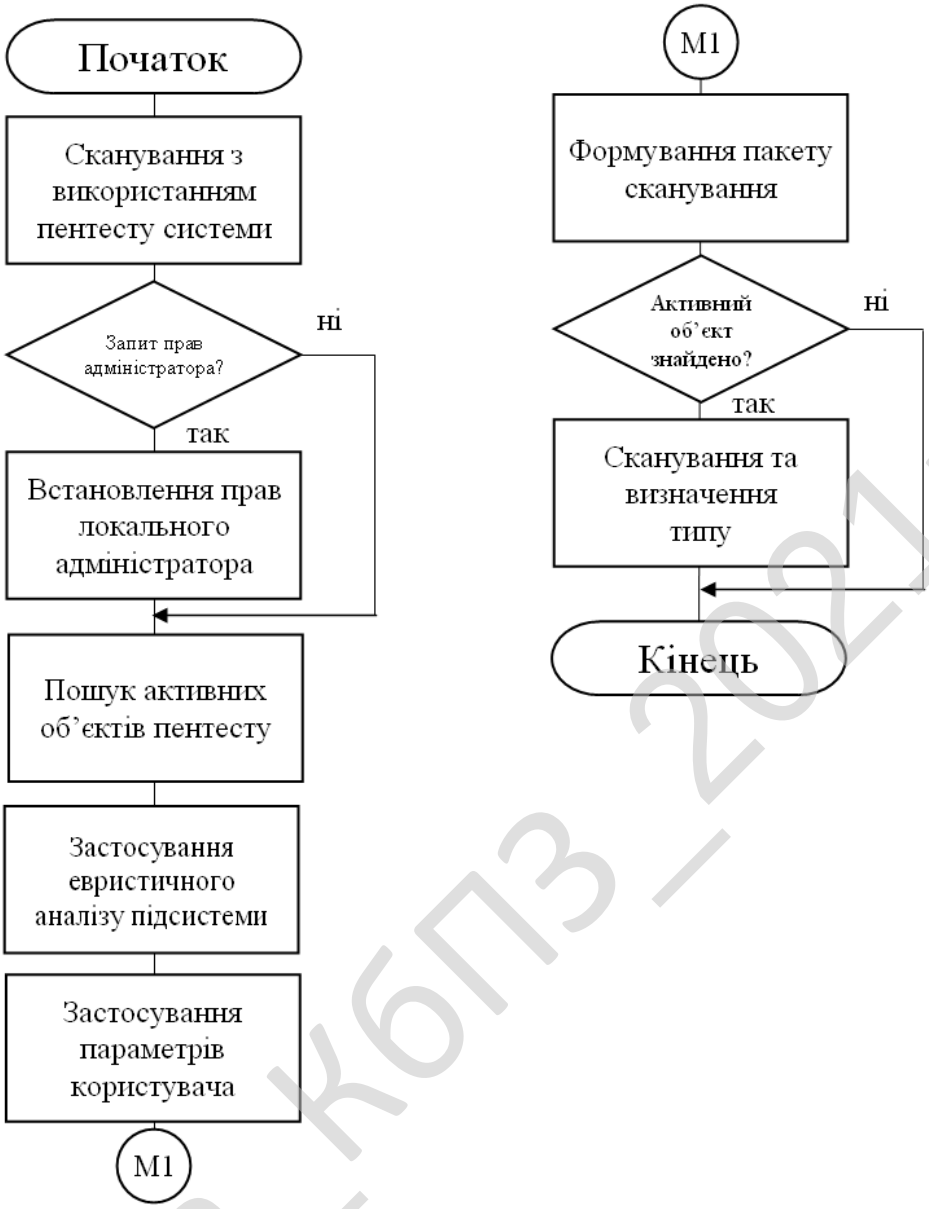


Рисунок 4.2 – Блок схема підпрограми

```
end;
FindRes:=FindNext(SR);
end;
FindClose(SR);
end;
```

Розглянемо вихідний код процедури локальних змін для подальшого формування поточного статусу системи.

```
{$R *.dfm} // ресурси реалізації

Procedure TStanSystemNow.StartMonitor;
```

```
begin
  StartProcessControl;
  PausePC.Enabled := True;
  StopPC.Enabled := True;
  StartPC.Enabled := False;
end;
procedure TStanSystemNow.StartPCClick(Sender: TObject);
begin
  StartMonitor;
end;
procedure TStanSystemNow.PausePCClick(Sender: TObject);
begin
  PauseProcessControl;
  PausePC.Enabled := False;
  StopPC.Enabled := True;
  StartPC.Enabled := True;
end;
procedure TStanSystemNow.ClosePCClick(Sender: TObject);
begin
  Close;
end;

// дії по таймеру
procedure TStanSystemNow.Timer1Timer(Sender: TObject);
  var
  ss,mm,hh:String;
begin
  if isMonRun then
    if Not MonPaused then
      Label1.Caption := MF.PCActive
    else
      Label1.Caption := MF.PCPaused;
  if not isMonRun then
    Label1.Caption := MF.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 TStanSystemNow.StopPCClick(Sender: TObject);
begin
  ExitProcessControl;
  PausePC.Enabled := False;
  StopPC.Enabled := False;
  StartPC.Enabled := True;
end;
procedure TStanSystemNow.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
      MF.MonFileCN := MF.MonFileCN + 1;
      MF.Label4.Caption := inttostr(MF.MonFileCN);
      MF.Edit3.Text := ProcList[ProcList.count-1];
      ID := _ScanFileEx(ProcList[ProcList.count-1]);
      if ID <> -1 then
      begin
        MF.ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+'
'+MF.ProcControlSt+ ' ' + '['+MF.INFECTED+' - '+GetName(ID)+']
'+ProcList[ProcList.count-1]);
        MF.MonFileInfected := MF.MonFileInfected + 1;
```

```
        MF.Label5.Caption := inttostr(MF.MonFileInfected);
        MF.Edit2.Text := GetName(ID);
        MF.Edit1.Text := ProcList[ProcList.count-1];
        MF.BalloonTrayIcon(MF.Handle ,1,10, ProcList[ProcList.count-1]
,'['+MF.INFECTED+' - '+GetName(ID)+' ]',bitError);
        if OptionsForm.PCAutoKill.Checked then
        if Not KillProcess(ExtractFileName(ProcList[ProcList.count-1])) then
Showmessage(MF.ErrorKillProc);
        ShowAlarmForm(ProcList[ProcList.count-1],'['+MF.INFECTED+' -
'+GetName(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;
```

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротака Такеучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос–функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеҐрейс та Шталь у книжці Злі проблеми, справедливі рішення послалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990–х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом щоб обробити та описати весь їхній досвід та найкращі

практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників.

Протягом кожного спринту, 15–30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту.

Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

Product backlog – це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називається «історіями» (user story) або елементами backlog–у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum–процесу.

Обов'язкові поля:

1. ID – унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування.

2. Назва (Name) – стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти, про що йдеться і відрізнити одну історію від іншої.

3. Важливість (Importance) – ступінь важливості даної історії на погляд product owner 'а. Зазвичай являє собою натуральне число, іноді для цієї цілі використовуються числа Фібоначчі. Чим більше значення, тим більше пріоритет.

4. Попередня оцінка (initial estimate) – початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point'ах. Приблизно відповідає числу «ідеальних людино–днів».

5. Як продемонструвати (how to demo) – стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

Додаткові поля. Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner'у визначитися з його пріоритетами.

Категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет.

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з

групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

Sprint backlog – містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі.

Burndown chart – показує, скільки вже виконано і скільки ще залишається зробити.

**Планування спринта (Sprint Planning Meeting)**

Проходить на початку нової ітерації Спринта:

– Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда;

– На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино–годинах;

– Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі;

– Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт;

– Тривалість наради обмежена зверху 4–8 годинами в залежності від тривалості ітерації, досвіду команди тощо;

– (перша частина наради) Беруть участь Product Owner + Команда: обирають задачі із Product Backlog;

– (друга частина наради) Бере участь лише команда: обговорюють технічні деталі реалізації, наповнюють Sprint Backlog.

**Щоденна нарада (Daily Scrum Meeting)**

Відбувається кожен день протягом спринта. Є «пульсом» ходу спринта. Нараді властиві наступні обмеження:

– починається точно вчасно;

– всі можуть спостерігати, але говорять тільки обрані;

– триває не більш ніж 15 хвилин;

– проводиться в одному і тому ж місці протягом одного спринта.

Протягом наради кожен член команди відповідає на 3 запитання:

– Що зроблено з моменту попередньої щоденної наради?;

– Що буде зроблено з моменту поточної наради до наступної?;

– Які проблеми заважають досягненню цілей спринта? (Над рішенням цих проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.)

Демонстрація (Sprint Review Meeting):

– Проходить у кінці ітерації (спринта).

– Команда демонструє внесок функціональності до продукту всім зацікавленим особам.

– Залучається максимальна кількість глядачів.

– Усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт).

– Обмежена 4–ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective):

– Члени команди висловлюють свою думку про минулий спринт.

– Відповідають на два основних запитання: Що було зроблено добре у минулому спринті?; Що потрібно покращити в наступному?.

– Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).

– Обмежена 1–3ма годинами.

Було використано MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій.

У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу.

Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому

горизонтальне масштабування стає простішим. Використовувана модель документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого безсхемного стилю (schemaless style), а внутрішнє угруповання релевантних даних забезпечує додатковий виграш в швидкодії.

Нереляційний підхід досить зручний для створення баз даних, у яких горизонтальне масштабування означає розгортання на множині машин. Можливість забезпечувати найкращу продуктивність повинна існувати паралельно з підтримкою більшої функціональності, ніж це дозволяє використання пар «ключ-значення» (у чистому вигляді).

Технологія баз даних має працювати скрізь, починаючи з серверів користувача та віртуальних машин і закінчуючи хмарними технологіями.

MongoDB, на думку розробників, має заповнити розрив між простими сховищами даних типу «ключ-значення» (швидкими і легко масштабованими) і великими СКБД (зі структурними схемами і потужними запитами).

Основні можливості MongoDB:

1. Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних).

2. Досить гнучка мова для формування запитів.

3. Динамічні запити.

4. Повна підтримка індексів.

5. Профілювання запитів.

7. Швидкі оновлення «на місці».

8. Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео.

9. Журналювання операцій, що модифікують дані в БД.

10. Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг.

11. Може працювати відповідно до парадигми MapReduce.

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS.

Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Є докладна і якісна документація, велике число прикладів і драйверів для популярних мов Java, C++, C#, PHP, Python, Perl, Ruby.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave.

Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave.

Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB – продукт досить молодий, і відтак у ньому зустрічаються помилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості). Компанія-розробник надає платні підтримку, хостинг, консультації.

### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм MMB, в основі якого лежить змішування операцій різних алгебраїчних груп. MMB – ітеративний алгоритм, що складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох великих оборотних нелінійних підстановок.

Ці підстановки визначаються за допомогою множення по модулю $2^{32}$-1 з постійними множниками. У підсумку з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм MMB оперує 32-бітовими підблоками тексту ($x_0$, $x_1$, $x_2$, $x_3$) і 32-бітовими підблоками ключу ($k_0$, $k_1$, $k_2$, $k_3$).

Це спрощує реалізацію алгоритму на сучасних 64-бітових процесорах. Чергуючись із операцією XOR, шість разів використовується нелінійна функція *f*.

Запишемо операції алгоритму (всі операції з індексами виконуються по модулю 4):

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

Функція *f* виконується в три кроки:

1. $x_i = c_i * x_i$ для $i = 0..3$ (Якщо на вході множення одні одиниці, то на виході – теж одні одиниці).

2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.

3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0..3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}$-1.

Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}$-1, результат теж дорівнює $2^{32}$-1.

В алгоритмі використовуються наступні константи:

$C$ = 2aaaaaaa, $c_0$ = 025f1cdb, $c_1 = 2*c_0, c_2=2^3 *c_0, c_3=2^7 *c_0.$

Константа $C$ – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом.

У константи $c_0$ є інші особливі характеристики. Константи $c_1$, $c_2$ і $c_3$ – зрушені версії $c_0$, і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість $c_i$ використовується $c_i^{-1}$. Значення $c_0^{-1}$ = 0dad4694.

# 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:

– Верхнього меню: Файл; Сканування; Контроль процесів; Параметри; Довідка.

– Функціональних кнопок головного вікна ПЗ: Сканування; Параметри.

– Розділу обрання типу пентесту (Test type and destination).

– Розділу обрання поточних параметрів (General parameters).

– Розділу додаткових параметрів що формуються у відповідності з обраним пентестом системи.

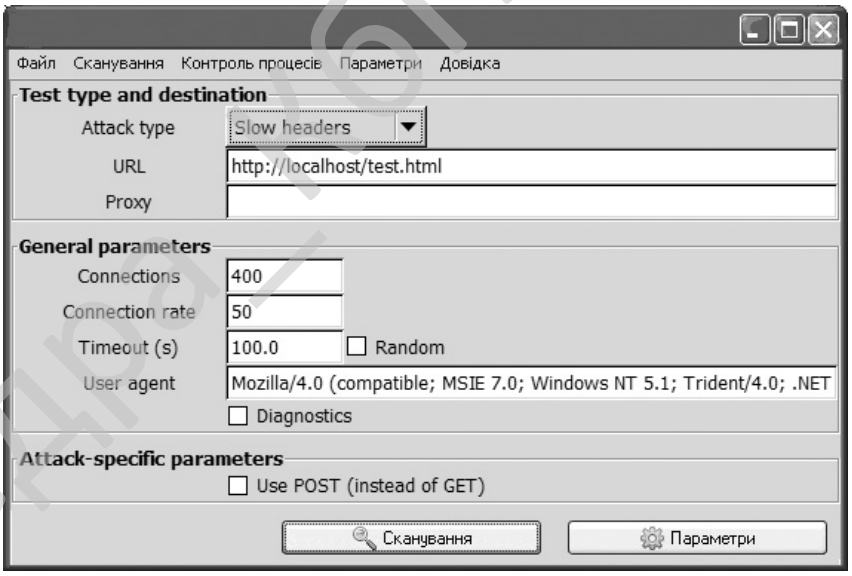


Рисунок 5.1 – Головне вікно ПЗ

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.

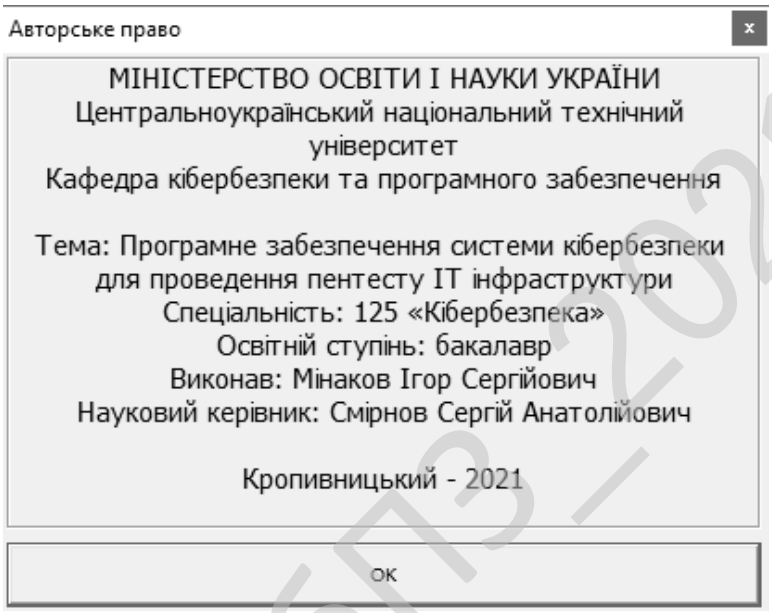


Рисунок 5.2 – Авторське право

Розглянемо 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.

# 6 ОСНОВНІ ВИСНОВКИ

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

– Досліджена система кібербезпеки для проведення пентесту ІТ інфраструктури.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для проведення пентесту ІТ інфраструктури.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання для проведення пентесту ІТ 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. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для проведення пентесту ІТ 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 XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

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. Моделирование защищенного канала связи с использованием экспоненциальной GERT-сети / С.Г. Семенов, А.А. Можаев // Информатика, математическое моделирование, экономика. – Смоленськ.: Смоленский филиал АНО ВПО ЦС РФ "Российский университет кооперации". – 2012. – Том.1. – С. 152-160.

2. Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов // Вісник Національного технічного університету «Харківський політехнічний інститут». – Х.:НТУ «ХПІ». – 2012. –№62 (968). – С 173-181.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

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

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

46. 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.

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

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

49. 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.

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

51. 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.

Додаток А
(обов'язковий)

**Технічне завдання**

**Зміст**

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки для проведення пентесту IT інфраструктури.

## 2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 185-02 від 28.12.2020 року).

## 3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи кібербезпеки для проведення пентесту IT 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 для проведення пентесту IT 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.

### 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 – 68 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 2.06.2021 р.

Додаток Б

(обов’язковий)

**Міністерство освіти і науки України**

**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Смірнов С.А.

***Програмне забезпечення системи кібербезпеки для проведення пентесту IT інфраструктури***

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 41

Літера: РП

Кропивницький – 2021 року

**Файл Pentest_IT_InfectedAction.pas – вибір дії над ІТ інфраструктурою**

```
unit Pentest_IT_InfectedAction;

interface

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

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 Pentest_IT_Main, Pentest_IT_Options;

{$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 Pentest_IT_Options.pas – параметри пентесту IT інфраструктури**

```
unit Pentest_IT_Options;

interface

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

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;
    optAntiVirus_ScanLabel: 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 Pentest_IT_Main, uPluginInfo, Pentest_IT_AddPath, 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_Pentest_IT_Scanner',False);
  end else
  begin
    ChangeReg('Virus_Pentest_IT_Scanner',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_ANTIVIRUS_SCAN_SUBDIR         := SCNSUBDIR.Checked;
        OPT_USE_HEX_MODE        := SCNHEX.Checked;
        OPT_USE_CRC_MODE        := SCNCRC.Checked;
        OPT_USE_HEX_INPOS       := SCNHEXINPOS.Checked;
        OPT_SEND_ANTIVIRUS_SCAN_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('*','Pentest_IT.Pentest_IT_Scan',MainForm.SysMenu,ParamStr(0)+'
%1');

FileTAddAction('Directory','Pentest_IT.Pentest_IT_Scan',MainForm.SysMenu,ParamSt
r(0)+' %1');

FileTAddAction('Drive','Pentest_IT.Pentest_IT_Scan',MainForm.SysMenu,ParamStr(0)
+' %1');
```

```
  end else
  begin
    FileTDelAction('Drive','Pentest_IT.Pentest_IT_Scan');
    FileTDelAction('Directory','Pentest_IT.Pentest_IT_Scan');
    FileTDelAction('*','Pentest_IT.Pentest_IT_Scan');
  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 Pentest_IT_Monitor.pas – монітор (контроль процесів)**

```
unit Pentest_IT_Monitor;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls, Pentest_IT_Kernel, Pentest_IT_Types,
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;
    PCAntiVirus_Scaned: 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 Pentest_IT_Main, Pentest_IT_InfectedAction, Pentest_IT_Options;
//***Функція створення параметрів проведення пентесту***//

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,'SeAntiVirus_ProjectPrivilege',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 := _Pentest_IT_ScanFileEx(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 := _Pentest_IT_ScanFileEx(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 Pentest_IT_Project.dpr – головний файл проекту**

```
program Pentest_IT_Project;
// Список підключаємих модулів
uses
  Forms,
  SysUtils,
  Pentest_IT_Kernel in '..\Pentest_IT Virus_Pentest_IT_Scanner
Modues\Pentest_IT_Kernel.pas',
  Pentest_IT_Types in '..\Pentest_IT Virus_Pentest_IT_Scanner
Modues\Pentest_IT_Types.pas',
  avMonitor in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avMonitor.pas',
  avVirus_Pentest_IT_Scanner in '..\Pentest_IT Virus_Pentest_IT_Scanner
Modues\avVirus_Pentest_IT_Scanner.pas',
  avHex in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avHex.pas',
  avDataBase in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avDataBase.pas',
  avHash in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avHash.pas',
  avExt in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avExt.pas',
  avAPI in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avAPI.pas',
  avConfig in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avConfig.pas',
  avShield in '..\Pentest_IT Virus_Pentest_IT_Scanner Modues\avShield.pas',
  langs in 'langs.pas',
  Pentest_IT_Main in 'Pentest_IT_Main.pas' {MainForm},
  uSelInfo in 'uSelInfo.pas' {InformationForm},
  Pentest_IT_Options in 'Pentest_IT_Options.pas' {OptionsForm},
  uPluginInfo in 'uPluginInfo.pas' {PluginAPIForm},
  Pentest_IT_AddPath in 'Pentest_IT_AddPath.pas' {AddUserPathForm},
  Pentest_IT_About in 'Pentest_IT_About.pas' {AboutForm},
  uSelDir in 'uSelDir.pas' {SelDirFrm},
  uMessage in 'uMessage.pas' {MessageFrm},
  uHideForm in 'uHideForm.pas' {HideForm},
  Pentest_IT_Monitor in 'Pentest_IT_Monitor.pas' {MonitorForm},
  Pentest_IT_InfectedAction in 'Pentest_IT_InfectedAction.pas' {ActionForm},
  uSplash in 'uSplash.pas' {SplashForm};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Virus_Pentest_IT_Scanner';
  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_Pentest_IT_ScannerKernel;
  {Hide Splash Form}
  SplashForm.Hide;
  Sleep(200);
  {Create Tray Icon}
  MainForm.CreateTray;
  {}
  if OptionsForm.AUTORUN.Checked then begin
```

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

**Файл Pentest_IT_Main.pas – основна програма**

```
unit Pentest_IT_Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, ExtCtrls, Menus, ImgList, XPMan,
Pentest_IT_Kernel, Pentest_IT_Types, 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;
    Pentest_IT_ScanPathesTab: TTabSheet;
    Pentest_IT_ScanningTab: TTabSheet;
    ReportTab: TTabSheet;
    BottomPanel: TPanel;
    Pentest_IT_ScanBTN: TButton;
    SaveBTN: TButton;
    PathList: TListView;
    Bevel1: TBevel;
    Pentest_IT_ScanList: 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;
mnuShowAntiVirusVirus_Pentest_IT_Scanner: TMenuItem;
mnuHideAntiVirusVirus_Pentest_IT_Scanner: TMenuItem;
N2: TMenuItem;
mnAntiVirus_Options: 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;
Pentest_IT_ScanTopBtn: TLabel;
Pentest_IT_ScanMenu: TPopupMenu;
mnuSelAntiVirus_ScanPath: TMenuItem;
mnuShowReport: TMenuItem;
N12: TMenuItem;
OptionTopBtn: TLabel;
PCTopBtn: TLabel;
mnuAntiVirusProcessControl: TMenuItem;
N19: TMenuItem;
mnuPCShow: TMenuItem;
N21: TMenuItem;
mnuPCRun: TMenuItem;
mnuPCPause: TMenuItem;
mnuPCStop: TMenuItem;
mnuAntiVirus_ScanStart: TMenuItem;
mnuStopAntiVirus_Scan: TMenuItem;
N13: TMenuItem;
mnuSaveReport: TMenuItem;
N26: TMenuItem;
mnuGoToTray: TMenuItem;
SOURCESTRING: TListBox;
LabelPanel: TPanel;
Pentest_IT_ScanFile: TLabel;
procedure DelAllClick(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure ExitBTNClick(Sender: TObject);
procedure Pentest_IT_ScanListDblClick(Sender: TObject);
procedure Pentest_IT_ScanBTNClick(Sender: TObject);
procedure InitVirus_Pentest_IT_ScannerKernel;
Procedure StartAntiVirus_Scan(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 mnuHideAntiVirusVirus_Pentest_IT_ScannerClick(Sender: TObject);
    procedure mnuShowAntiVirusVirus_Pentest_IT_ScannerClick(Sender: TObject);
    procedure mnuExitClick(Sender: TObject);
    procedure mnAntiVirus_OptionsClick(Sender: TObject);
    procedure mnuHelpClick(Sender: TObject);
    procedure mnuAboutClick(Sender: TObject);
    procedure ApplicationEventsMinimize(Sender: TObject);
    procedure AppMinimize(Sender: TObject);
    procedure FormPaint(Sender: TObject);
    procedure Pentest_IT_ScanListCustomDrawItem(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 Pentest_IT_ScanTopBtnClick(Sender: TObject);
    procedure mnuShowReportClick(Sender: TObject);
    procedure mnuSelAntiVirus_ScanPathClick(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 Pentest_IT_ScanMenuPopup(Sender: TObject);
    procedure mnuAntiVirus_ScanStartClick(Sender: TObject);
    procedure mnuStopAntiVirus_ScanClick(Sender: TObject);
    procedure mnuSaveReportClick(Sender: TObject);
    procedure CopyRightLabelClick(Sender: TObject);
    Procedure CreateTray;
  protected
    procedure MineRestore(var Msg: TMessage); message WM_MINERESTORE;
    procedure SendAntiVirus_Scanning(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;

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

        AntiVirusMonitor      : String;
        AntiVirusInit         : String;
        LoadAPI          : String;
        LoadDB           : String;
        CreateDrvList    : String;
        OptFileNotFnd    : String;
        LoadOptFile      : String;
        InitProcedures   : String;
        initShield       : String;
        ErrorInit        : String;
        LogBevel         : String;
        DBKnowledge      : String;
```

```
    SCNOBJ          : String;
    Pentest_IT_ScanExecute     : String;
    Pentest_IT_ScanEnd         : String;
    PrepareToAntiVirus_Scan   : String;
    FileIgnor       : String;
    FileIfect       : String;
    FileAntiVirus_Scanned     : String;
    DataAntiVirus_Scanned     : String;
    IGNORED         : String;
    SKIPBYSIZE      : String;
    INFECTED        : String;
    STOPB           : String;
    RETURNB         : String;
    ANTIVIRUS_SCANB          : String;
    SCNFILE         : String;
    FileDel         : String;
    FileNotDel      : String;
    PATHNOSEL       : String;
    SysMenu         : String;
    NfoAntiVirusVirus_Pentest_IT_Scanner   : String;
    NfoAntiVirusKernel    : String;
    NfoAntiVirusBuild     : String;
    DelDialog       : String;
    DelAllDialog    : String;
    DelError        : String;
    HelpNOFound     : String;
    avShieldMes     : String;
    avError         : String;
    DelResult       : String;
    AllInfected     : String;
    DeleteInfected  : String;
    SkippedInfected : String;
    AntiVirusCloseDlg     : String;
    AllreadyInAntiVirus_Scan  : 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;
  AntiVirusVirus_Pentest_IT_ScannerCapt  = 'Антивірусний захист операційної
системи від шкідливих програм';
  AntiVirusVirus_Pentest_IT_ScannerVS    = '';
var
  MainForm      : TMainForm;
  inAntiVirus_Scan        : Boolean = False;
  NeedToReturn  : Boolean = False;
  FirstRun      : Boolean = True;
  P             : TPoint;
  MayClose      : boolean=false;
```

```
implementation

uses uSelInfo, Pentest_IT_Options, Pentest_IT_AddPath, Pentest_IT_About, Math,
uMessage, uHideForm,
  Pentest_IT_Monitor, Pentest_IT_InfectedAction, uPluginInfo;
{$R *.dfm}

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

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

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

procedure TMainForm.SendAntiVirus_Scanning;
var
  pcd: PCopyDataStruct;
begin
  pcd := PCopyDataStruct(Msg.LParam);
  if not inAntiVirus_Scan then
  begin
    StartAntiVirus_Scan(PChar(pcd.lpData));
  end
  else begin
    MessageDlg(AllreadyInAntiVirus_Scan,mtError,[mbOK],0);
  end;
end;

procedure TMainForm.MineRestore(var Msg: TMessage);
begin
  if (Msg.Msg = WM_MINERESTORE) then
  begin
    mnuShowAntiVirusVirus_Pentest_IT_Scanner.Click;
  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 := 'Pentest_IT Virus_Pentest_IT_Scanner';
  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_ANTIVIRUS_SCAN_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_ANTIVIRUS_SCAN_FILE;

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

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

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

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

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

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

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

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

      if GETParamName(AntiVirusConfig[i]) = 'PATH' then begin
      with OptionsForm.PathList.Items.Add do begin
        Caption := GetParam(AntiVirusConfig[i]);
        if DirectoryExists(Caption) then ImageIndex := 4 else ImageIndex := 5;
      end;
      end;

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

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

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

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

      if GETParamName(AntiVirusConfig[i]) = 'AUTOSAVEREPORTTO' then
OptionsForm.ReportSavePath.Text := GETParam(AntiVirusConfig[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.Pentest_IT_ScanList.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.Pentest_IT_ScanList.Handle, WM_VSCROLL, SB_BOTTOM, 0);
end;

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

```
end;

//***Функція вибору параметрів проведення пентесту на віруси***//

procedure OnAntiVirus_ScanComplete;
  var
  Pentest_IT_ScanEndBalloonText: String;
  i: integer;
begin
  MainForm.ProgressBar.Max := 1;
  MainForm.ProgressBar.Position := MainForm.ProgressBar.Max;
  MainForm.Pentest_IT_ScanBTN.Caption := MainForm.RETURNB;
  NeedToReturn := True;
  inAntiVirus_Scan := 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.Pentest_IT_ScanFile.caption  := MainForm.Pentest_IT_ScanEnd;
  OnAddToLogStr('',-1);
  OnAddToLogStr(MainForm.Pentest_IT_ScanEnd,0);
  OnAddToLogStr('',-1);
  OnAddToLogStr(MainForm.FileAntiVirus_Scanned+inttostr(MainForm.FileCN),0);
  OnAddToLogStr(MainForm.FileIgnor+inttostr(MainForm.FileIgnored),0);
  OnAddToLogStr(MainForm.FileIfect+inttostr(MainForm.FileInfected),0);

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

  Pentest_IT_ScanEndBalloonText := MainForm.Pentest_IT_ScanEnd + ':' + Return +
Return
                        +' >>
'+MainForm.FileAntiVirus_Scanned+inttostr(MainForm.FileCN) + Return
                        +' >> '+MainForm.FileIgnor+inttostr(MainForm.FileIgnored)
+ Return
                        +' >>
'+MainForm.FileIfect+inttostr(MainForm.FileInfected) + Return
                        +' >>
'+MainForm.DataAntiVirus_Scanned+Format('%.2f',[Pentest_IT_ScannedDataSize /
1024 / 1024])+' Mb';

  MainForm.BalloonTrayIcon(MainForm.Handle
,1,10,Pentest_IT_ScanEndBalloonText,'Pentest_IT
Virus_Pentest_IT_Scanner',bitInfo);
end;

//***Функція початку проведення пентесту***//

Procedure OnAntiVirus_ScanStart;
  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.Pentest_IT_ScanBTN.Caption := MainForm.STOPB;
```

```
  MainForm.SaveBTN.Enabled := False;
  MainForm.Pentest_IT_ScanList.Clear;
  MainForm.Pentest_IT_ScanningTab.Show;
  MainForm.FileCN       := 0;
  MainForm.FileInfected := 0;
  MainForm.FileIgnored  := 0;
  inAntiVirus_Scan := True;
  NeedToReturn := False;
  OnAddToLogStr(MainForm.Pentest_IT_ScanExecute,0);
  if AntiVirusVirus_Pentest_IT_Scanner.AvAction = TAntiVirus_ScanDir then
  else
  OnAddToLogStr(MainForm.SCNOBJ+AntiVirusVirus_Pentest_IT_Scanner.FileName,0);
  OnAddToLogStr('',-1);
  MainForm.BalloonTrayIcon(MainForm.Handle
,1,10,MainForm.Pentest_IT_ScanExecute,'Pentest_IT
Virus_Pentest_IT_Scanner',bitInfo);
  AntiVirusVirus_Pentest_IT_Scanner.Resume;
end;

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

Procedure AntiVirusKernelMessageAPI(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.Pentest_IT_ScanBTN.Enabled := False;
  end;

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

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

  if MES = MES_PREPARINGTOANTIVIRUS_SCAN then
MainForm.Pentest_IT_ScanFile.Caption := MainForm.PrepareToAntiVirus_Scan;

  if mes = MES_INITKERNEL then OnAddToLogStr(MainForm.AntiVirusInit,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_ONANTIVIRUS_SCANEXECUTE then
    OnAntiVirus_ScanStart;

  if MES = MES_ONANTIVIRUS_SCANCOMPLETE then
    OnAntiVirus_ScanComplete;

  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.Pentest_IT_ScanFile.caption  := '['+inttostr(MainForm.FileCN)+']
'+ExtractFileName(Pr_1);
    end
    else
      MainForm.Pentest_IT_ScanFile.caption  := ExtractFileName(Pr_1);
    if OPT_SEND_ANTIVIRUS_SCAN_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 антивірусa***//

procedure TMainForm.InitVirus_Pentest_IT_ScannerKernel;
  var
  i:integer;
begin

//******//
        AntiVirusMonitor      := SOURCESTRING.Items[0];
        AntiVirusInit         := 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];
        Pentest_IT_ScanExecute     := SOURCESTRING.Items[13];
        Pentest_IT_ScanEnd         := SOURCESTRING.Items[14];
        PrepareToAntiVirus_Scan   := SOURCESTRING.Items[15];
        FileIgnor       := SOURCESTRING.Items[16];
        FileIfect       := SOURCESTRING.Items[17];
        FileAntiVirus_Scanned     := SOURCESTRING.Items[18];
        DataAntiVirus_Scanned     := SOURCESTRING.Items[19];
        IGNORED         := SOURCESTRING.Items[20];
        SKIPBYSIZE      := SOURCESTRING.Items[21];
        INFECTED        := SOURCESTRING.Items[22];
        STOPB           := SOURCESTRING.Items[23];
        RETURNB         := SOURCESTRING.Items[24];
        ANTIVIRUS_SCANB           := SOURCESTRING.Items[25];
        SCNFILE         := SOURCESTRING.Items[26];
        FileDel         := SOURCESTRING.Items[27];
        FileNotDel      := SOURCESTRING.Items[28];
        PATHNOSEL       := SOURCESTRING.Items[29];
        SysMenu         := SOURCESTRING.Items[30];
        NfoAntiVirusVirus_Pentest_IT_Scanner   := SOURCESTRING.Items[31];
        NfoAntiVirusKernel    := SOURCESTRING.Items[32];
        NfoAntiVirusBuild     := 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];
        AntiVirusCloseDlg     := SOURCESTRING.Items[44];
        AllreadyInAntiVirus_Scan  := 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(AntiVirusKernelMessageAPI);
        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)+'
'+NfoAntiVirusVirus_Pentest_IT_Scanner +AntiVirusVirus_Pentest_IT_ScannerVS);
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+' '+NfoAntiVirusKernel
+GetKernelVersion);
  ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+' '+NfoAntiVirusBuild
+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('*','Pentest_IT.Pentest_IT_Scan',SysMenu,ParamStr(0)+
' %1');

OptionsForm.FileTAddAction('Directory','Pentest_IT.Pentest_IT_Scan',SysMenu,Para
mStr(0)+' %1');

OptionsForm.FileTAddAction('Drive','Pentest_IT.Pentest_IT_Scan',SysMenu,ParamStr
(0)+' %1');
  end else
  begin
    OptionsForm.FileTDelAction('Drive','Pentest_IT.Pentest_IT_Scan');
    OptionsForm.FileTDelAction('Directory','Pentest_IT.Pentest_IT_Scan');
    OptionsForm.FileTDelAction('*','Pentest_IT.Pentest_IT_Scan');
  end;
end;

//***Функція початку проведення пентесту***//

Procedure TMainForm.StartAntiVirus_Scan(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
    AntiVirusVirus_Pentest_IT_Scanner :=
TAvVirus_Pentest_IT_Scanner.Create(true);
    AntiVirusVirus_Pentest_IT_Scanner.NeedForAPI := TRUE;
    AntiVirusVirus_Pentest_IT_Scanner.AvAction   := TAntiVirus_ScanDir;
    Path.Add(ExtractFileDrive(Paramstr(0))+'\');
    AntiVirusVirus_Pentest_IT_Scanner.Dirs       := Path;
    OnAntiVirus_ScanStart;
    exit;
  end;
```

```
  if DirectoryExists(Parametr+'\') then
  begin
    AntiVirusVirus_Pentest_IT_Scanner :=
TAvVirus_Pentest_IT_Scanner.Create(true);
    AntiVirusVirus_Pentest_IT_Scanner.NeedForAPI := TRUE;
    AntiVirusVirus_Pentest_IT_Scanner.AvAction   := TAntiVirus_ScanDir;
    Path.Add(Parametr+'\');
    AntiVirusVirus_Pentest_IT_Scanner.Dirs       := Path;
    OnAntiVirus_ScanStart;
    exit;
  end;

  if FileExists(Parametr) then
  begin
    AntiVirusVirus_Pentest_IT_Scanner :=
TAvVirus_Pentest_IT_Scanner.Create(true);
    AntiVirusVirus_Pentest_IT_Scanner.NeedForAPI := false;
    AntiVirusVirus_Pentest_IT_Scanner.AvAction := TAntiVirus_ScanFile;
    AntiVirusVirus_Pentest_IT_Scanner.FileName := Parametr;
    OnAntiVirus_ScanStart;
    exit;
  end;

end;

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

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

procedure TMainForm.Pentest_IT_ScanBTNClick(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 inAntiVirus_Scan = False then
    begin
      if PATH.Count-1 <> -1 then
      begin
        AntiVirusVirus_Pentest_IT_Scanner :=
TAvVirus_Pentest_IT_Scanner.Create(true);
        AntiVirusVirus_Pentest_IT_Scanner.FreeOnTerminate := True;
        AntiVirusVirus_Pentest_IT_Scanner.NeedForAPI      := true;
        AntiVirusVirus_Pentest_IT_Scanner.AvAction        := TAntiVirus_ScanDir;
        AntiVirusVirus_Pentest_IT_Scanner.Dirs            := MainForm.Path;
        OnAntiVirus_ScanStart;
      end
      else begin
        MessageDlg(PATHNOSEL,mtError,[mbOk],0);
      end;
    end
    else begin
      CloseAntiVirus_ScanThread;
    end;
  end else
  begin
    Pentest_IT_ScanBTN.Caption := Pentest_IT_ScanB;
    MainForm.SaveBTN.Enabled := False;
    NeedToReturn := False;
    Pentest_IT_ScanPathesTab.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 Pentest_IT_ScanList.Items.Count-1 do
      Report.Add(Pentest_IT_ScanList.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 := AntiVirusVirus_Pentest_IT_ScannerVS;
end;

procedure TMainForm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  if MessageDlg(AntiVirusCloseDlg,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 (Pentest_IT_ScanList.ItemIndex <> -1) and
(Pentest_IT_ScanList.Selected.ImageIndex = 2) and (inAntiVirus_Scan = False)
then
  begin
    Del.Visible := true;
  end
  else
    Del.Visible := False;

  if (Pentest_IT_ScanList.ItemIndex <> -1) and (inAntiVirus_Scan = 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 Pentest_IT_ScanList.Items.Count - 1 do
        if Pentest_IT_ScanList.Items.Item[i].ImageIndex = 2 then
      begin
            c:=c+1;
                    try
                        if
DeleteFileBC(Pentest_IT_ScanList.Items.Item[i].SubItems[0]) then
          begin
            d:=d+1;
                          Pentest_IT_ScanList.Items.Item[i].ImageIndex := 4;

      ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileDel+Pentest_IT_S
canList.Items.Item[i].SubItems[0]);
```

```
          end
                        else begin

      ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileNotDel+Pentest_I
T_ScanList.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(Pentest_IT_ScanList.Selected.SubItems[0]) then
    begin
      Pentest_IT_ScanList.Selected.ImageIndex := 4;

ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileDel+Pentest_IT_ScanLis
t.Selected.SubItems[0]);
    end
    else begin

ReportMemo.Lines.Add(FormatDateTime('[hh:mm:ss]',now)+FileNotDel+Pentest_IT_Scan
List.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                       := AntiVirusVirus_Pentest_IT_ScannerCapt;
  TopPn.DoubleBuffered                  := true;
  TopRightPanel.DoubleBuffered      := true;
  PathList.DoubleBuffered             := true;
  Pentest_IT_ScanList.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;
  Pentest_IT_ScanList.Columns.Items[0].Width := Pentest_IT_ScanList.Width - 25;
end;

procedure TMainForm.mnuHideAntiVirusVirus_Pentest_IT_ScannerClick(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.mnuShowAntiVirusVirus_Pentest_IT_ScannerClick(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.mnAntiVirus_OptionsClick(Sender: TObject);
begin
  if not inAntiVirus_Scan 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
  mnuHideAntiVirusVirus_Pentest_IT_Scanner.Click;
end;

procedure TMainForm.FormPaint(Sender: TObject);
begin
  if FirstRun then
    if OptionsForm.AUTOHIDE.Checked then
    begin
      mnuHideAntiVirusVirus_Pentest_IT_Scanner.Click;
    end;
  FirstRun := false;
end;

procedure TMainForm.Pentest_IT_ScanListCustomDrawItem(Sender: TCustomListView;
  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);
begin
  with Pentest_IT_ScanList.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.Pentest_IT_ScanTopBtnClick(Sender: TObject);
begin

Pentest_IT_ScanMenu.Popup(MainForm.Left+Pentest_IT_ScanTopBtn.Left+3,MainForm.To
p+Pentest_IT_ScanTopBtn.Top+38);
end;

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

procedure TMainForm.mnuSelAntiVirus_ScanPathClick(Sender: TObject);
begin
  if not inAntiVirus_Scan then
    Pentest_IT_ScanPathesTab.Show;
end;

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

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

procedure TMainForm.mnuGoToTrayClick(Sender: TObject);
begin
  mnuHideAntiVirusVirus_Pentest_IT_Scanner.Click;
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 inAntiVirus_Scan then mnAntiVirus_Options.Enabled := False else
mnAntiVirus_Options.Enabled := True;
end;

procedure TMainForm.Pentest_IT_ScanMenuPopup(Sender: TObject);
begin
  mnuPCRun.Enabled := MonitorForm.StartPC.Enabled;
  mnuPCPause.Enabled := MonitorForm.PausePC.Enabled;
  mnuPCStop.Enabled := MonitorForm.StopPC.Enabled;
  mnuSaveReport.Enabled := SaveBTN.Enabled;
  if inAntiVirus_Scan then mnuAntiVirus_ScanStart.Enabled := False else
mnuAntiVirus_ScanStart.Enabled := True;
  if inAntiVirus_Scan then mnuStopAntiVirus_Scan.Enabled := True else
mnuStopAntiVirus_Scan.Enabled := False;
end;

procedure TMainForm.mnuAntiVirus_ScanStartClick(Sender: TObject);
begin
  Pentest_IT_ScanBTN.Click;
end;

procedure TMainForm.mnuStopAntiVirus_ScanClick(Sender: TObject);
begin
  Pentest_IT_ScanBTN.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 Pentest_IT_AddPath.pas – додавання шляхів проведення пентесту**

```
unit Pentest_IT_AddPath;

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 Pentest_IT_Main, Pentest_IT_Options, 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 Pentest_IT_About.pas – довідка**

```
unit Pentest_IT_About;

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 Pentest_IT_Main;

{$R *.dfm}

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

end.
```