

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи кібербезпеки антивірусного
захисту мобільних пристроїв”**

Виконав здобувач вищої освіти
IV курсу, групи КБ-21-ЗСК
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Галкіна Д.О.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 "Інформаційні технології"
Спеціальність 125 "Кібербезпека"
Освітньо-професійна (освітньо-наукова) програма "Кібербезпека"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Галкіній Дар'ї Олександрівні

(прізвище, ім'я, по батькові)

1. Тема роботи

*Програмне забезпечення системи кібербезпеки
антивірусного захисту мобільних пристроїв*

2. Керівник роботи

Смірнов Сергій Анатолійович, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 136-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту

23.05.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв*

5. 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 кібербезпеки в промислову експлуатацію.

6. Висновки

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

Структурна схема системи кібербезпеки

1 аркуш

Функціональна схема системи кібербезпеки

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

7. Дата видачі завдання « 17 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника

Смірнов С.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Галкіна Д.О.
(прізвище та ініціали)

АНОТАЦІЯ

Галкіна Д.О. Програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки антивірусного захисту мобільних пристроїв.

Метою розробки є програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв.

Результат роботи – програмна реалізація системи кібербезпеки антивірусного захисту мобільних пристроїв.

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 мобільних пристроях з ОС Android.

Програму розроблено в середовищі RAD Studio Delphi.

Ключові слова: кібербезпека, антивірус, мобільні пристрої

ABSTRACT

Halkina D.O. Software of the cyber security system of antivirus protection of mobile devices. 125 Cyber security. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the first (bachelor) level of higher education, software is developed, which is intended for the cyber security system of antivirus protection of mobile devices.

The purpose of the development is the software of the cyber security system of antivirus protection of mobile devices.

The result of the work is the software implementation of the cyber security system of antivirus protection of mobile devices.

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

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on mobile devices with Android OS.

The program was developed in the RAD Studio Delphi environment.

Keywords: cyber security, antivirus, mobile devices

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	18
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	37
3.4 Розробка діаграми процесів.....	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	41
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	41
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	62
6 ОСНОВНІ ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66

						ВКРБ-125.24.0039.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Галкіна Д.О.				Програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв	Літ.	Аркуш	Аркушів
Перев.	Смірнов С.А.					Б	1	72
Н.контр.	Коваленко А.С.				ЦНТУ КБ-21-3СК			
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- ООП – об'єктно-орієнтоване програмування;
- ОС – операційна система;
- КС – комп'ютерна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп'ютер;
- ЦО – цивільна оборона;
- ОПР – охорона праці;
- USB – Universal Serial Bus – універсальна послідовна шина;
- БС – базові станції;
- AMPS – Advanced Mobile Phone Service – поліпшена послуга мобільного телефонного зв'язку;
- SMS – Short message service – служба коротких повідомлень.
- КС – комп'ютерна система;
- URL – universal resource locator – локатор ресурсів Інтернет.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Зазвичай жоден смартфон не постачений передвстановленим антивірусом, за замовчуванням вважається, що користувач свого смартфона буде завантажувати тільки офіційні додатки з магазину Google Play. Магазин Google Play сам по собі є безпечним середовищем (у цьому магазині додатки проходять перевірки, у ході яких відбувається фільтрація небажаних додатків). Але крім офіційного магазину додатків користувач може завантажувати додатки з інших сайтів-джерел, тим самим не гарантовано, що додатки будуть безпечними, не всі люди переслідують благі цілі.

Безпека мобільного пристрою залишається дуже важливим питанням сучасності. На мобільних пристроях можуть зберігатися особисті дані людини, які можуть бути цікаві зловмисникам. В умовах сучасності актуальне використання онлайн-сервісів банків, тому при влученні вірусу в телефон є ризик, що ваші дані будуть читати чужі люди. Установка антивірусу на мобільний пристрій допоможе позбутися від проблем з вірусами.

При акуратному й грамотному користуванні й дотриманні мір обережності проблем з вірусами звичайно не виникає. Але як, не намагайся ховатися від вірусів, якщо є вихід в інтернет, то ризик завжди буде, отут вирішальну роль грає ступінь ризику.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв.

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

- Огляд існуючих систем антивірусного захисту мобільних пристроїв.
- Дослідження системи кібербезпеки антивірусного захисту мобільних пристроїв.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– Програмна реалізація системи кібербезпеки антивірусного захисту мобільних пристроїв.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі антивірусного захисту мобільних пристроїв.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ_2024

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для антивірусного захисту мобільних пристроїв. Вхід в систему тепер здійснюється не лише з комп'ютерів, але і з мобільних пристроїв, на яких, як правило, або взагалі немає антивірусу, а якщо і є – то безкоштовна, дуже обмежена в функціоналі.

Інформація з мобільних пристроїв (разом з паролями і логінами) може потрапити не завжди в дружелюбні руки.

Кількість загроз для ОС Android зростає в катастрофічному темпі разом з ростом кількості пристроїв

Можливості антивірусного захисту мобільних пристроїв:

- безперервний захист файлової системи пристрою в режимі реального часу (перевірка файлів, що зберігаються, встановлюваних програм і т.д.);
- сканування всіх файлів системи або окремих файлів і папок за запитом користувача;
- сканування архівів;
- сканування карти пам'яті;
- виявлення файлів автозапуску Windows;
- виявлення загроз у файлах. Lnk-ярлики для виклику файлів (що визначаються Dr.Web як Exploit.Cpllnk);
- видалення виявлених загроз безпеки або переміщення їх в карантин;
- фільтрації вхідних дзвінків та SMS-повідомлень на основі попередньо визначених та створених користувачем чорних і білих списків;
- оновлення вірусних баз Dr.Web через Інтернет-з'єднання;
- ведення статистики виявлених загроз і дій програми, а також журналу реєстрації подій;

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- пошук та екстренне блокування телефону при його втраті або крадіжці;
- захист від небажаних інтернет-ресурсів при використанні стандартного браузера Android, а також браузера Google Chrome.

1.2 Область застосування

Областю застосування системи є мобільні пристрої під керівництвом ОС Android. Android – операційна система для смартфонів, планшетів і нетбуків. Компанія Google придбала розробника програмного забезпечення Android inc. в 2005 році. Операційна система Android заснована на модифікованому ядрі Linux. Згодом, Google та інші учасники Open Headset Alliance співпрацювали для спільної розробки цієї нової операційної системи. Далі Android Open Source Project (AOSP) доручено підтримання та подальший розвиток платформи. У Android є велика спільнота розробників, які розширюють функціональність пристроїв.

OS Android має свій офіційний магазин з продажу додатків – Android Market. Включає він в себе як платні програми, так і безкоштовні. В даний момент, для України доступні для завантаження тільки безкоштовні програми та ігри. Так як OS Android є відкритою, користувачеві надається можливість завантажувати додатки та ігри з інших ресурсів.

Під Android розробники, в основному, пишуть програми на мові Java, керуючі пристроєм через розроблені Google бібліотеки.

Офіційно про OS Android стало відомо 5 листопада 2007 року, коли було оголошено про Open Headset Alliance – консорціум з 80 компаній. Більшу частину коду Android була випущена під ліцензією Apache.

Android програми включають в себе java-додатки та бібліотеки, які запускаються віртуальною машиною Dalvik з JIT компілятором. Бібліотеки включають в себе систему управління, графіку OpenGL ES 2.0, движок WebKit, графічний движок SGL, SSL і бібліотеки Bionic. OS Android складається з 12

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

мільйонів рядків коду, в тому числі 3-х мільйонів рядків XML, 2.8 мільйонів рядків на C, 2.1 мільйона рядків на Java і 1.75 мільйона рядків на C ++.

Android™ – надійна та гнучка програмна платформа. Завдяки найширшому спектру можливостей налаштування телефону, програм і віджетів, на екрані телефону зручно розміщується від 3 до 7 віртуальних екранів, які можна розбити по темах і категоріями (новини, ігри, соціальні мережі і т.д.). Головне меню програм ділиться на максимум 10 категорій з метою спростити і прискорити навігацію. Qwerty-клавіатура, вбудована в телефон, прискорить процес набору тексту.

Важливий фактор – швидкодія додатків, інтернету та решти функціоналу, яке забезпечує смартфон операційна система Android.

Медіаплеєр підтримує формати Xvidi DivX, що дозволяє дивитися їх без форматування відеофайлів. Програма відеоредагування містить функцію розкадровки – в ній навіть новачок освоїть відеомонтаж і зможе додавати до свого відео текст і музику прямо в телефоні.

Android забезпечує величезний вибір розваг, а також зручний і простий доступ до популярних соціальних мереж (Facebook, Вконтакте та ін), так що де б ви не були, ви завжди будете он-лайн. Різноманітність ігор обов'язково вас вразить, і ви чудово проведете час, граючи по бездротовому зв'язку зі своїми друзями або на самоті.

Android – відкрита платформа, що відрізняє її від інших, а також відкриває нові колосальні можливості і гарантує інноваційний напрямок розвитку. Унікальні програми для Android створюють незалежні розробники, у яких доступ до платформи практично необмежений. Тобто, кількість додатків, які не мають аналогів, все більш якісних, корисних і цікавих, в Android Market безперервно зростає. Операційна система оновлюється і піддається удосконаленням.

Android – це відмінний вибір, який підтверджується сотнями і сотнями покупок смартфонів на базі цієї ОС. Оптимальні опції, вражаючий набір

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

можливостей, прекрасний функціонал – все це робить операційну систему Android воістину однією з кращих на ринку.

Основні переваги Android:

– Більше 70 тисяч безкоштовних програм на Android Market (на даний момент).

– Зручний інтерфейс, що настроюється.

– Документи і пошта завжди доступні.

– Висока швидкість роботи: завантаження фото і відео за секунди, перегляд фільмів без затримок.

– Швидкий і легкий доступ в інтернет через браузер Google.

– Карти Google.

– Qwerty-екранна клавіатура.

– Доступ до популярних соціальних мереж.

– Величезна кількість ігор. Ігри по бездротовому з'єднанню.

– Фото, відео і музичні редактори.

– Сервіси Google: Gmail, Gtalk, Maps, Youtube.

– Надійна підтримка.

Серед основних нововведень – оптимізація зі слабенькими смартфонами.

Тепер Android нормально працюватиме з апаратами, в яких лише 512 Мб оперативної пам'яті. Також розробники покращили голосове управління і "дзвонилку". Останню навчили автоматично сортувати контакти залежно від того, як часто ви їм телефонуєте чи пишете. Варто відзначити і можливість приховати кнопки у повноекранному режимі, а додаток Hangouts тепер об'єднує SMS, MMS, миттєві повідомлення Google Talk та відеодзвінки через інтернет.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

У цьому розділі розглянемо які бувають антивіруси для Android, приведемо короткий огляд і класифікацію.

avast! Mobile Security

Як тільки я довідався зі спливаючої підказки avast! про появу нового антивірусу для Android, я відразу ж без роздумів поставила собі avast! на мобільний пристрій.

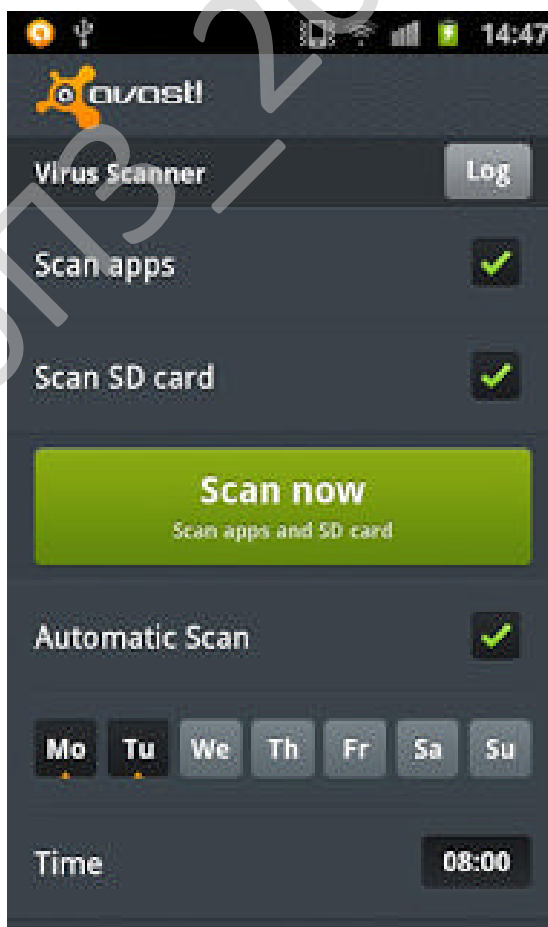


Рисунок 2.1 – Інтерфейс користувача avast! Mobile Security

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Набрано більше 275 тисяч завантажувачів, це чимало (в 2 рази більше, ніж скачали Dr.Web).

По функціях мобільний avast! випереджає всі інші антивіруси, такі як: AVG, Lookout, Norton, Kaspersky, McAfee. Наприклад, не всі антивіруси мають таку функцію, як звіт про конфіденційність, повідомлення про зміну сім-карти, а також взагалі немає брандмауера.

Dr.Web Mobile

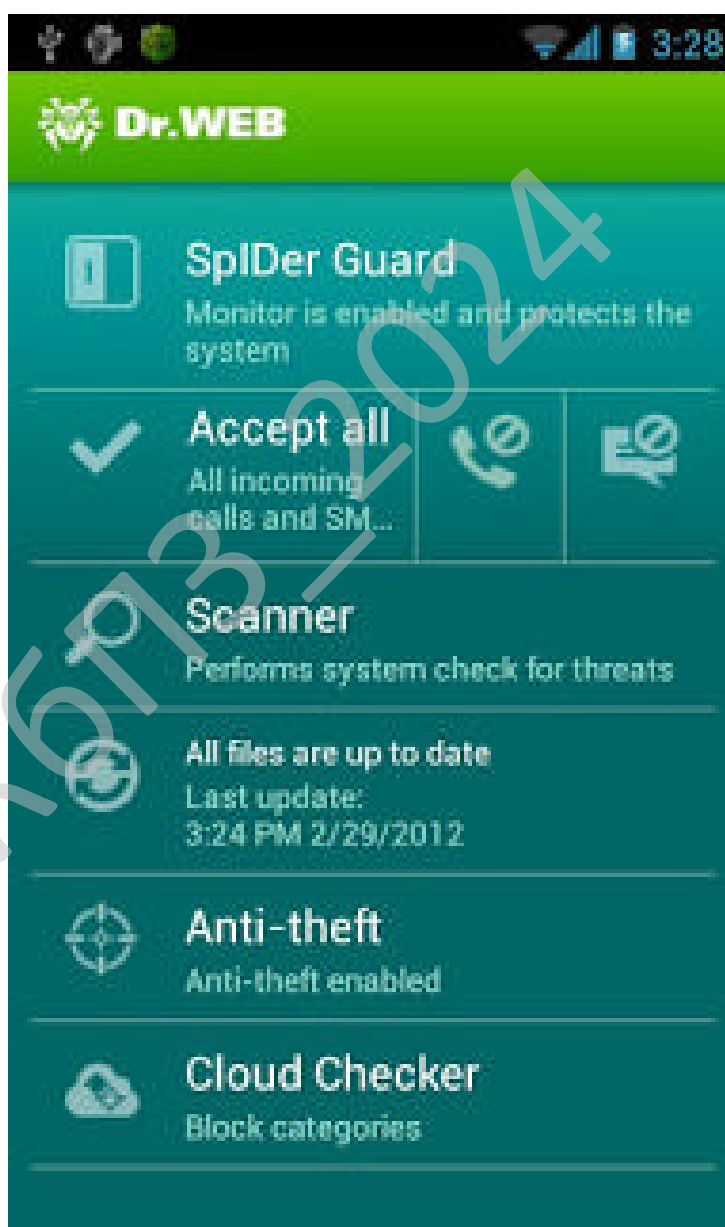


Рисунок 2.2 – Інтерфейс користувача Dr.Web Mobile

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Виділяють 4 версії Dr.Web Mobile:

- Dr.Web для Android (має розширений набір характеристик).
- Dr.Web для Symbian OS (схожі характеристики з версією для Windows Mobile).
- Dr.Web для Windows Mobile/
- Також є версія антивірусу Dr.Web Light. У загальному-те теж саме, тільки деякі можливості урізані. Наприклад, немає антиспаму, антизлодія, хмарного URL-фільтра за назвою Cloud Checker (деякі функції можуть бути змінені).

Lookout Антивірус

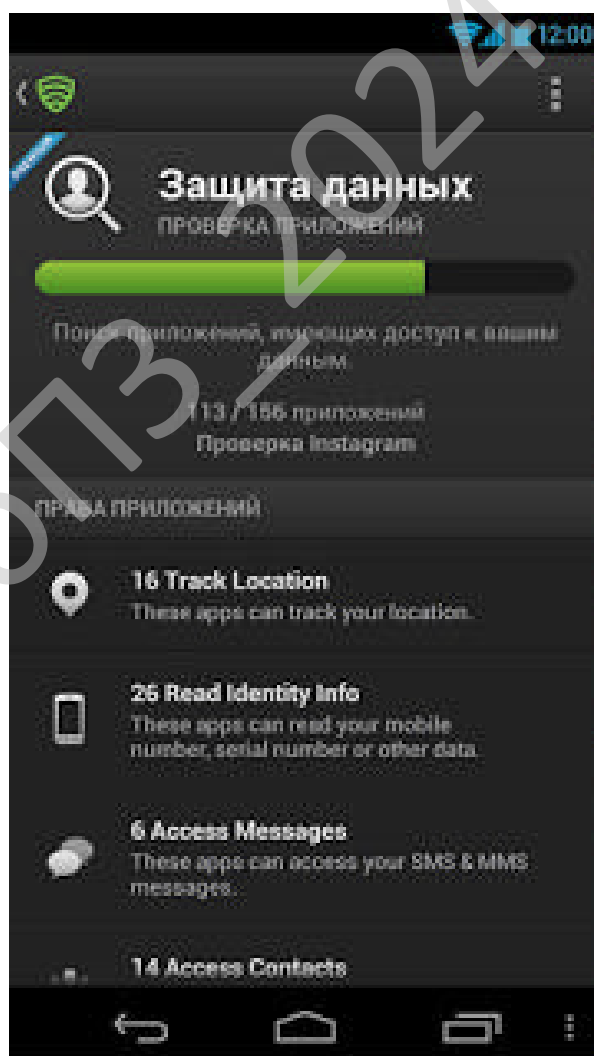


Рисунок 2.3 – Интерфейс користувача Lookout

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Lookout має безліч шанувальників, адже це саме мобільний антивірус, розроблювачі приділяють розробці підвищену увагу. Завантажень більше 400000.

Головною фішкою Lookout є наявність засобів для пошуку телефону, навіть якщо його втрати.

Android Antivirus

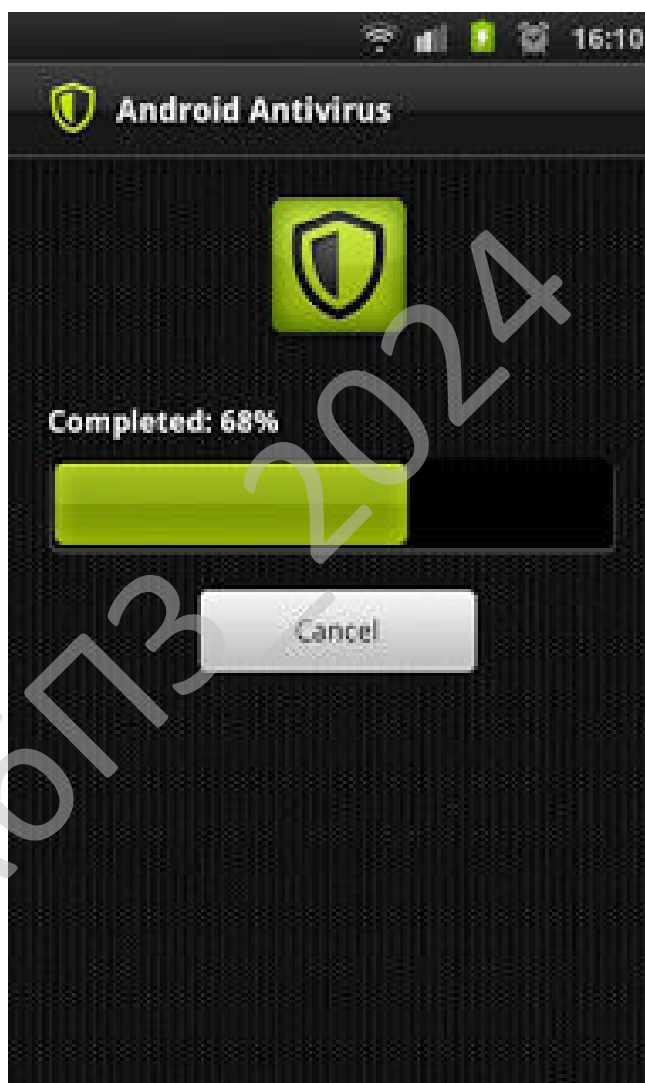


Рисунок 2.4 – Інтерфейс користувача Android Antivirus

Навіть таке буває, сам був здивований. Дуже оригінальна назва: Android Antivirus. Судячи з відзвів докладно не будемо зупинятися, не хвалять.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

До речі дуже схожий антивірус, тільки назва небагато інша – Antivirus for Android.

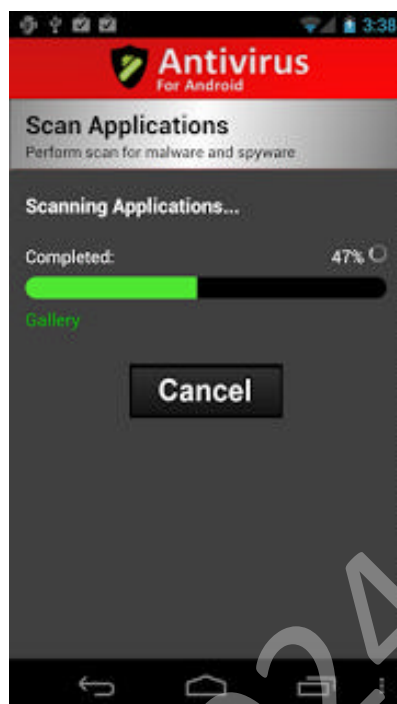


Рисунок 2.5 – Інтерфейс користувача Antivirus for Android

TrustGo Antivirus & Mobile Security

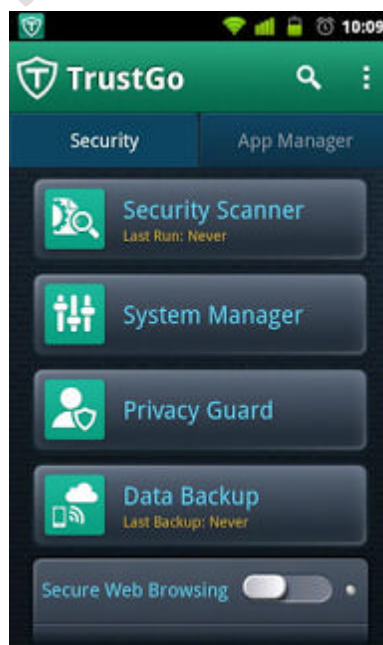


Рисунок 2.6 – Інтерфейс користувача TrustGo Antivirus & Mobile Security

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Таки пристойний антивірус із непоганим функціоналом, правда англійською мовою (для багатьох і це не перешкода).

VG

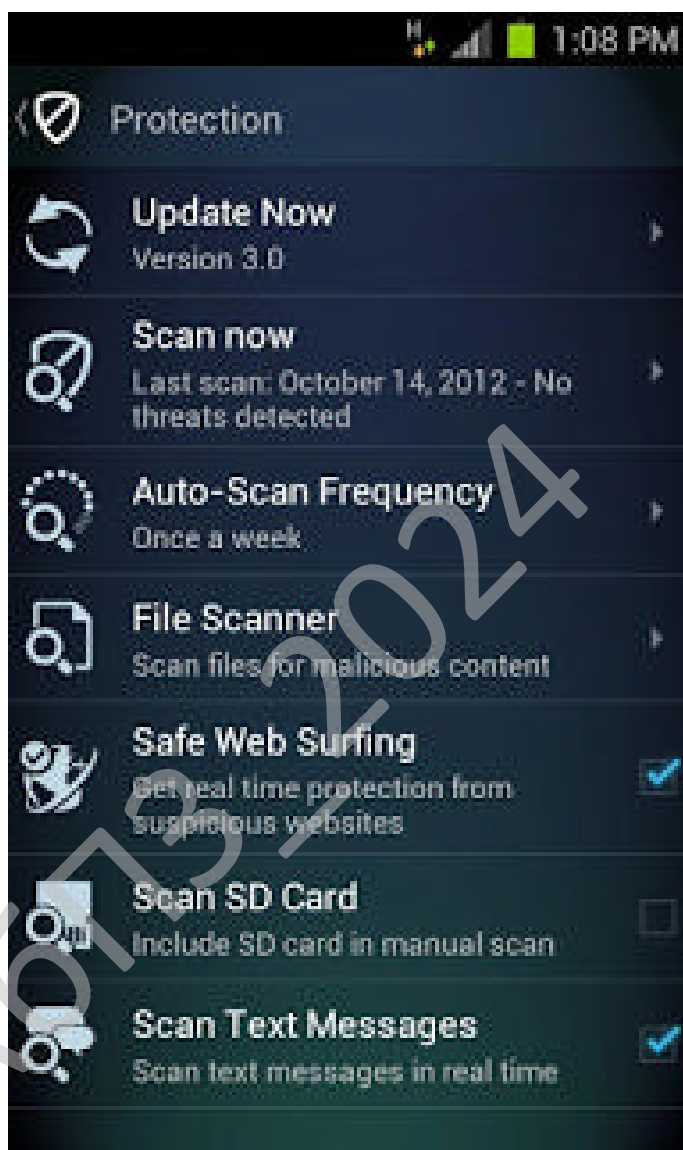


Рисунок 2.7 – Інтерфейс користувача VG

Просто рекордсмен по кількості завантажуваль – уже 500000. Один з найкращих мобільних антивірусів, для когось найкращий. Є професійна версія, а також версія для планшетів.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Zoner AntiVirus Free

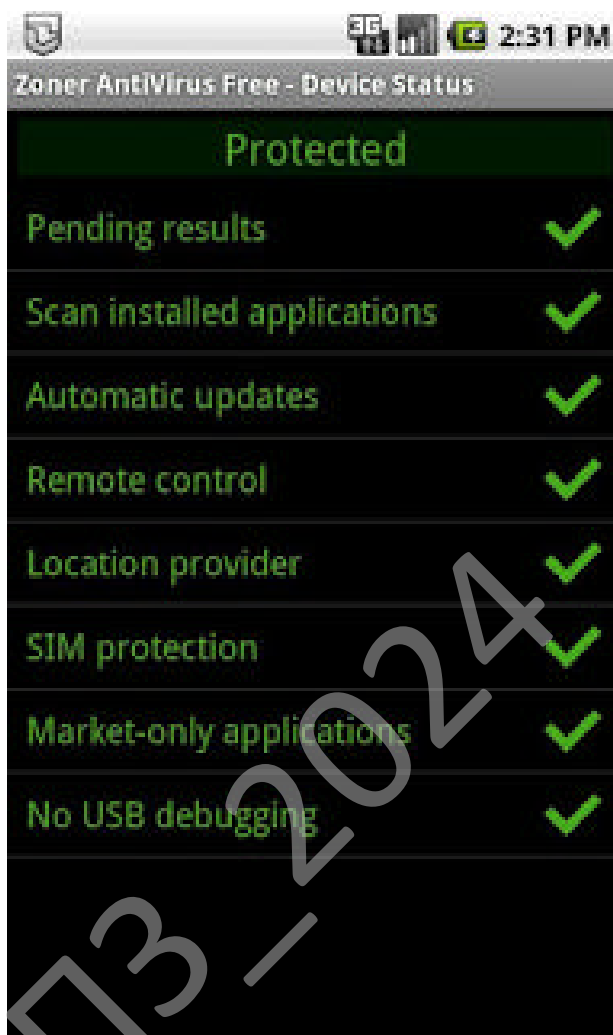


Рисунок 2.8 – Інтерфейс користувача Zoner

Ще один безкоштовний антивірус для телефону – Zoner. Захищає від вірусів, троянів, хробаків і інших вірусів. Гідний конкурент багатьом антивірусам.

Norton Mobile Security

Гідний конкурент на ринку мобільних антивірусів.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15



Рисунок 2.9 – Інтерфейс користувача Norton Mobile Security
Bitdefender Antivirus Free

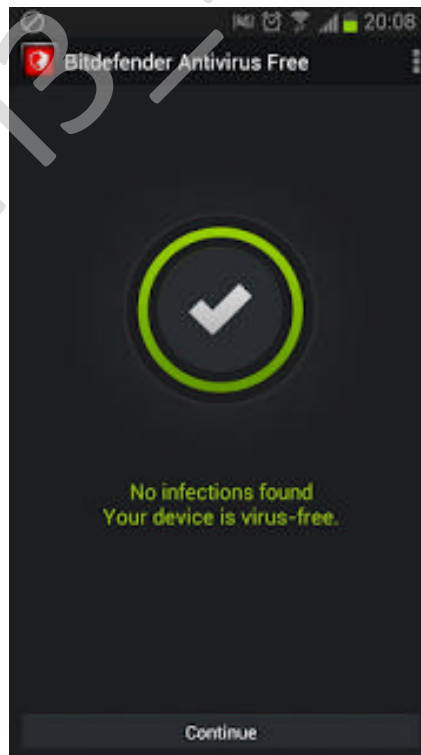


Рисунок 2.10 – Інтерфейс користувача Bitdefender Antivirus Free

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Гарний антивірус, віруси знаходить, що вже багато чого виходить, а головне – безкоштовний.

Окремо виділимо наступні програми:

- Antivirus Free (англійський, для відновлення баз просить грошей).
- SecureBrain Antivirus (не знаходить віруси).
- LabMSF Antivirus beta (теж не знаходить).
- BluePoint Antivirus Free (погано шукає).
- Armor for Android™ Antivirus (дорого коштує).
- Fastscan Anti-Virus (японські розроблювачі, відгуки нейтральні).
- MyAntiVirus Pro – AntiVirus (нічого привабливого).
- BlackBelt AntiVirus – Trial (середненько).
- Privateer Antivirus & Security (при тестах нижче середнього).
- Sophos Security & Antivirus (немає української мови, а в цілому, непоганий).
- Max Mobile Security – Lite (особливо нічого ні поганого, ні добре не можна виділити).
- Virus Seeker Mobile Security (маловідомий, середнього рівня).
- Mobile Security & Antivirus (є тільки демо-версія).
- Scan from syncNscan (нічого особливого).
- Netlux Android Total Security (аналогічно).
- AegisLab Antivirus Premium (у цілому, непоганий, кілька мов).
- White-Gate (тільки як доповнення до основного антивірусу).
- Mobile Security Lite (багато функцій, непогано).
- NQ Mobile Security & Antivirus (немає української мови).
- G Data AntiVirus Free (гарний, безкоштовний).

Всі представлені додатки доступні на play google com.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки 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 – використовувачи 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.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, 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.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– 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.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Це розширює потужність конструкцій records в Delphi, які використовуються 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 – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим 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.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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 Snake.
- 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 масштабуються 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 компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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 TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- 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.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки антивірусного захисту мобільних пристроїв.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти 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 кібербезпеки контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, 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 кібербезпеки 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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

За цей час антивірусні програми пройшли шлях від аматорських до комерційних пакетів, що приносять своїм авторам мільйони доларів. Комп'ютерні віруси заважають користувачам по усьому світу, і більш 20 років антивірусні експерти борються з ними.

Однак практично всі основні антивірусні технології були придумані ще до середини 90х років, і останні 10 років спостерігається лише екстенсивний розвиток. Необхідно чітко розуміти, що антивірусна програма (антивірус) це програма для виявлення комп'ютерних вірусів, а також небажаних (що вважаються шкідливими) програм і відновлення заражених (модифікованих) такими програмами файлів, а також для профілактики – запобігання зараження (модифікації) файлів або операційної системи шкідливим кодом.

Антивірусне програмне забезпечення складається з комп'ютерних програм, які намагаються знайти, запобігти розмноженню і видалити комп'ютерні віруси та інші шкідливі програми.

Антивіруси нерозривним образом зв'язані зі своїм антиподом – вірусами. Зникни, раптом, з якої-небудь причини віруси, необхідність в антивірусах зникла би. Але, на щастя для розроблювачів антивірусів, така ситуація з ряду гіпотетичних і фантастичних. Тому, щоб краще зрозуміти, етапи еволюційного розвитку антивірусних програм, необхідно розглядати їхнє вдосконалення невідривно від удосконалювання самих вірусів.

Усе більше розроблювачів пропонують антивірусні програми для боротьби з вірусами й захисту мобільних телефонів. У мобільних пристроях є наступні види боротьби з вірусами: сигнатурний; захист від спама по SMS; шифрування даних.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Зараз будь-який мобільний пристрій, що використовує будь-яку ОС без антивірусного захисту, вважається непрацездатним і ненадійним інструментом на якому втрата даних неминуча. На жаль, конкуренція між антивірусними компаніями призвела до того, що розвиток йде в бік збільшення кількості вірусів, а не в бік покращення їх детектування і алгоритмів лікування файлів, що заражені. На даний момент немає надійного антивірусу для захисту мобільних пристроїв.

Антивірусний захист для пристрою Android захищає від шкідливих програм. Розглянемо типові функції такого антивірусу.

Фільтрація вхідних дзвінків і SMS-повідомлень

Блокування вказаних номерів, користувачі яких не зможуть подзвонити або надіслати SMS.

Відслідковування місцеположення втраченого або викраденого пристрою

Компонент надає можливості дистанційного пошуку та повернення телефону.

Контроль стану захисту

Віджет для головного екрану дозволяє легко бачити стан захисту.

Попередження про заражені веб-сайти

Веб-екран попереджає про спроби переходу за будь-якими URL-адресами, зараженими шкідливим ПЗ.

Брандмауер

Якщо на телефоні виконаний рутінг, можете додати брандмауер та захистити свій пристрій навіть від найталановитіших кіберзломників.

Контроль мережевого трафіка та використання Wi-Fi

Якщо у встановлені місячні ліміти на обсяг трафіка, ця функція допоможе не вийти за рамки бюджету.

Автоматичне виправлення помилково введених URL-адрес

Якщо активували функцію SiteCorrect, вона допоможе уникнути проблем, пов'язаних із помилково введеними адресами.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Віджет

Відображення рядка стану на екрані телефону (для швидкого доступу до антивірусної програми для Android).

Веб-керування

Керуйте функцією s у телефоні із веб-порталу облікових записів avast!.

SiteCorrect

Автоматичне виправлення помилково введених URL-адрес з опціями «переходити автоматично» або «з підтвердженням».

Лічильник трафіка

Можливість дізнатися, скільки даних передається на телефон та з нього через підключення 3G та/або Wi-Fi. Перегляд статистики за день, місяць або рік, з прямим доступом до диспетчера програм на той випадок, якщо бажаєте внести зміни.

Інтерфейс користувача

Завдяки оптимізації інтерфейсу під відображення на планшеті тепер зручно використовувати також на планшетах з операційною системою Android.

Антивірус

Виконання сканування за вимогою встановлених програм і вмісту картки пам'яті, а також сканування програм при першому доступі до них (виконанні). Можливість планування сканувань, оновлення визначень вірусів, видалення програм і файлів, повідомлення про фальшиву тривогу в нашу антивірусну лабораторію.

Звіт про конфіденційність

Визначення та відображення (у вигляді таблиці) прав доступу та намірів установлених програм, визначення потенційних факторів небезпеки для конфіденційності – завдяки цьому знатимете, скільки інформації фактично надаєте кожній програмі.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Фільтрація дзвінків/SMS

Можливість фільтрувати дзвінки та (або) SMS-повідомлення зі списку контактів з використанням таких параметрів, як дні тижня, час початку та закінчення фільтрації. Перенаправлення блокованих дзвінків на голосову пошту, збереження блокованих повідомлень у журналі фільтра. Також можливе блокування вихідних дзвінків.

Менеджер програм

Ця функція аналогічна диспетчеру завдань Windows і відображає список програм, що виконуються, та їхній розмір (у Мб), навантаження на процесор та кількість потоків і служб, з можливістю зупинити або видалити програму/процес.

Веб-екран

Веб-екран – для Android, що входить до складу хмарної служби – WebRep, сканує кожен URL-адресу, що завантажується браузером, та попереджає про завантаження адрес, заражених шкідливими програмами.

Маскування програми

Після завантаження – користувач може вибрати будь-яке ім'я для маскування програми (наприклад, «гра Pinocchio»), щоб злодіям було ще важче знайти та видалити її.

Режим невидимки

Після увімкнення захисту від викладення значок програми приховується у панелі програм, не залишаючи будь-яких слідів на телефоні – тобто програма стає «невидимою», ускладнюючи для злодіїв завдання знайти або видалити її.

Список довірених SIM-карток

Визначення «білого списку» надійних SIM-карток, які можна використовувати в телефоні без подачі сигналу про викрадення. також можете легко очистити список довірених SIM-карток, залишивши в якості довіреної лише ту картку, що знаходиться в телефоні.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Економія заряду акумулятора

запускається та працює лише тоді, коли програмі потрібно виконати певні завдання. Це дозволяє економити заряд акумулятора та ускладнює для злодіїв завдання закрити програму.

Сповіднення про зміну SIM-картки

Якщо ваш телефон викрадено і використовується інша (неавторизована) SIM-картка, телефон може заблокувати себе, увімкнути сирену та надіслати сповіщення (на іншій пристрій) про новий номер та географічне положення.

Самозахист

Компонент є надзвичайно складним для видалення (особливо на телефонах з виконаним рутінгом). Він захищає себе від видалення, маскуючись із використанням різноманітних методів. На телефонах з рутінгом також здатен витримувати жорстке скидання до заводських установок і навіть відключати USB-порт телефону.

Дистанційна зміна конфігурації

Майстер налаштування допомагає користувачу пройти процес інсталяції на телефонах з рутінгом, чи то виконуючи запис безпосередньо, чи то створюючи файл оновлення. Для установки немає потреби у знанні інтерфейсу командної строки. Також підтримується можливість оновлення версії програми.

Дистанційні функції

Віддалене увімкнення сирени

Можливість увімкнути гучну сирену, яка відновлює максимальну гучність кожного разу, коли злодії намагаються зменшити її. Існує можливість вибору звуку сирени (наприклад, можете записати власний звук сирени або вибрати звуковий файл).

Віддалене блокування

Якщо бажаєте захистити ваші дані, можна блокувати телефон дистанційно. У такому випадку доступ до телефону стане можливим тільки після введення пароля, вказаного користувачем.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Блокування доступу до налаштувань телефону

Увімкнення віддаленого блокування диспетчера програм у телефоні та (або) налаштувань телефону.

Віддалене відображення

Дистанційне надсилання бажаного повідомлення на заблокований чи розблокований екран телефону (наприклад, з інформацією про винагороду за його повернення).

Віддалений пошук

Віддалений пошук телефону через GPS, WiFi або мобільну мережу (точність залежить від параметрів пошуку). можете визначати місцеположення телефону як одноразово, так і постійно (на телефонах з рутінгом, а також на телефонах без рутінгу, але з версією ОС Android від 1.6 до 2.2 можливе автоматичне увімкнення GPS).

Віддалене стирання даних

Дистанційний запуск безповоротного стирання всіх даних у телефоні (наприклад, контактів, журналу дзвінків, SMS/MMS, історії браузера, програм, облікових записів електронної пошти), яке включає надійне фізичне стирання даних з усіх карток пам'яті на пристроях з Android версій від 2.2 (на попередніх версіях доступність цієї функції є обмеженою).

Віддалені дзвінки

Дистанційна передача на телефон команди подзвонити з вимкненим екраном, щоб злодії не бачили, що відбувається такий дзвінок. У такий спосіб зможете почути, що відбувається поруч із телефоном.

Віддалена переадресація

Можливість дистанційно вмикати/вимикати переадресацію дзвінків або копіювання даних про дзвінки та (або) повідомлення SMS на іншій пристрій. У такому випадку на віддалений пристрій передаються дані про дзвінок, але не сам виклик. Утім, SMS-повідомлення переадресовуються цілком (також копія залишається на самому пристрої).

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Віддалене сповіщення про те, що телефон «Втрачений»

Дистанційне надсилання команди «Втрачений», яка викликає такі самі дії, як і заміна SIM-картки (напр., блокування телефону, сирену або блокування USB). Команда «Знайдений» дозволяє деактивувати команду «Втрачений».

Віддалене отримання історії

Відправлення запиту на отримання журналу дзвінків, контактів та SMS-повідомлень (кожне повідомлення відправляється окремо, але можливе застосування фільтру).

Віддалене відправлення SMS

Дистанційна команда телефону надіслати SMS-повідомлення на інший номер. У такий спосіб, навіть забувши десь телефон, зможете відповідати на вхідні дзвінки або SMS-повідомлення.

Віддалений перезапуск

Перезавантаження телефону SMS-командою, після чого буде запропоновано ввести PIN-код SIM-картки. (Працює лише на телефонах з виконаним рутінгом).

Сповіщення про низький заряд акумулятора

Телефон здатен відправляти на інший пристрій сповіщення про низький заряд акумулятора.

Дистанційна зміна конфігурації

Усі налаштування можна змінювати дистанційно.

3.2 Розробка структурної схеми

Антивірус не обмежуються захистом тільки від вірусів, а детектує так само троянські програми, Adware, Spyware і деякі інші.

Використання більш ніж одного антивірусного продукту для захисту в режимі реального часу може спричинити конфлікти та надмірне використання системних ресурсів, тому строго рекомендується використовувати лише один

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

антивірус для захисту в режимі реального часу.

Антивіруси надають захист від вірусів, троянів і руткітів. Але зі збільшенням кількості шкідливих програм, паралельно з антивірусом потрібно мати і інші програми для забезпечення безпеки комп'ютера – в першу чергу фаєрволи і антишпигуни.

Серед найнебезпечніших чинників, що можуть вплинути на функціонування організації це: втрата інформації; потрапляння до злочинців; порушення бізнес-процесів внаслідок технічних збоїв; неефективне використання працівниками робочого часу.

Антивірус містить модуль для захисту засобів обміну миттєвими повідомленнями і модуль для захисту програм P2P (peer-to-peer). Список підтримуваних програм досить широкий, включає більше 30 найменувань. Тоді як чати самі по собі не несуть серйозної загрози відносно проникнення вірусів, сучасні застосування для обміну повідомленнями є не тільки інструментом передачі інформації: більшість з них підтримує різні способи використання загального доступу до файлів, які можуть легко привести до вірусних заражень, якщо їх належним чином не відстежувати. Модуль захисту P2P не потребує подальшого опису – сучасні мережі P2P (наприклад, Kazaa) містять тисячі потенційно заражених файлів, тому питання захисту надзвичайно важливе.

Як наслідок це приводить до значних фінансових втрат. Обов'язкова властивість будь-якого якісного захисту – це здатність не заважати виконанню основних функцій системи, що захищається, предмета або людини.

Такий захист повинна забезпечувати безпека, не втручаючись у діяльність об'єкта, що захищається, по можливості бути невидимою, її вплив – зведене до мінімуму.

Однак від цього в жодному разі не повинна страждати надійність. У випадку антивірусного захисту такий баланс звичайно досягається шляхом введення двох різних режимів роботи антивірусних програм з можливістю самостійного докладного налаштування користувачем параметрів кожного з них,

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

а також розробки окремих програмних комплексів для комп'ютерів, що виконують різні функції.

При використанні антивірусних засобів, крім їхніх загальних характеристик (потужності, коректності перевірок, зручності роботи та ін.) важлива також їх актуальність, тобто відповідність вірусної бази даних теперішньому моменту її використання в антивірусних засобах.

У деяких випадках швидкого поширення вірусів важливо, щоб поточна антивірусна база на комп'ютері, що перевіряється, відрізнялася б від надаваної постачальником антивірусних засобів не більше ніж на добу. Відомі випадки, коли цей строк вдавалося скоротити до декількох годин, і це рятувало.

Список різних вірусів на сьогоднішній день дуже великий і постійно зростає. За деяким даними, щотижня до нього додається порядку 100 нових (включаючи модифікації раніше відомих). Звичайно, сучасні антивірусні продукти у своєму комплекті мають засобу відновлення антивірусних баз.

Зрозуміло, час від часу міняються й самі антивірусні засоби. Природнім було б передбачити той самий апарат для відновлення як антивірусних баз, так і модулів комплекту.

Однак не все так просто: виникають питання ліцензійності (користувач заплатив за однократну поставку, навіщо ж йому надавати наступну версію) і інші.

Розглянемо розроблену бакалаврську структурну схему системи яка зображена на рисунку 3.1.

Після завантаження розробленого ПЗ та початку роботи ПЗ у прихованому режимі в мобільному пристрої програма виконує дві основні функції.

1. Чекає запиту від користувача на перевірку файлової системи та при отриманні запиту проводить сканування. У свою чергу сканування розподіляється на два типи. Сканування у режимі антивірус коли проводиться порівняння заданих файлів з сигнатурами які знаходяться у антивірусної бази сигнатур для виявлення вірусів, троянів та інше.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

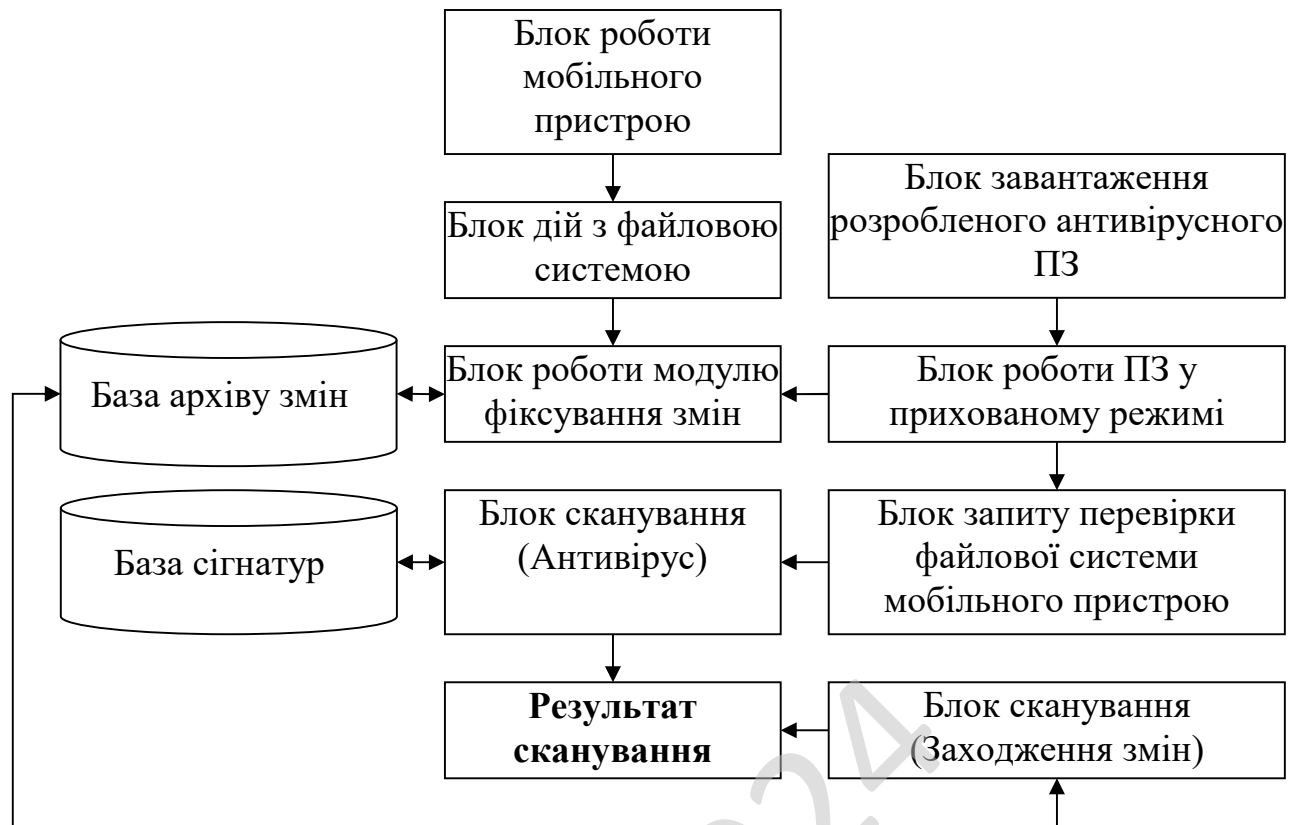


Рисунок 3.1 – Структурна схема системи

Та сканування для знаходження змін у файлах які вже раніше було про скановано на наявність вірусів, це сканування виявляє зміни файлах та при знаходженні цих змін дозволяє проводити їх відновлення. Всі зміни файлів знаходяться в архіві змін. Після сканувань результат виводиться на екран.

2. У автоматичному режимі перевіряє зміни у файловій підсистемі у модулі фіксування змін. При виявленні змін при роботі мобільного пристрою (дії користувача та програм) через архів змін дозволяє відновити файли.

Нормальній роботі мобільного пристрою заважають наступні загрози: Помилкова зміна, видалення, перейменування файлів користувачем; Віруси; Трояни; Хробаки (вірусні); Пошкоджене ПЗ стороннього розробника яке видозмінює файли користувача. Всі ці загрози діють на мобільний пристрій через файлову підсистему.

Розроблене бакалаврське ПЗ дозволяє частково, а в деяких випадках

повністю захистити мобільну систему від цих загроз за допомогою модуля перевірки файлів на віруси (антивірус) з антивірусною базою та модулю фіксування змін в системі з обліком всіх змін у архіві.

Антивірусну базу та інтерфейс ПЗ можна оновити через мережу Інтернет. Файли оновлення шукаються у каталогі data безкоштовного файл серверу у мережі Інтернет.

Основні функції та переваги антивіруса:

– Постійний антивірусний захист і нейтралізація загроз. Перевірка файлової системи здійснюється файловим монітором SpIDer Guard. У режимі реального часу він перевіряє всі файли при спробі їх збереження в пам'яті пристрою.

– Перевірка за запитом користувача. Вона здійснюється за допомогою спеціального компонента – сканера. Він дозволяє робити швидке або повне сканування файлової системи, а також перевіряти окремі файли і папки.

– Наявність карантину. Виявлені загрози можуть бути відразу видалені або при можливості поміщені в карантин, в якому відображається детальна інформація про потенційно шкідливі програми.

– Захист картки SD від інфікування файлами автозапуску і Exploit. Cpllnk, що представляють небезпеку для Windows.

– Малі розміри оновлень вірусних баз дозволяють заощадити трафік і зберегти кошти на рахунку при використанні лімітних мобільних тарифів.

– Докладна статистика про роботу антивіруса.

– Два зручних віджета для управління антивірусом на вибір користувача (розміри 1x1 і 4x1).

Антиспам:

– Вибір режимів фільтрації дзвінків і повідомлень.

– Можливість створення власних профілів фільтрації.

– Редагування чорного списку. У чорний список можуть бути додані номери, з яких ви хочете заблокувати вхідні дзвінки та повідомлення.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– Перегляд заблокованих дзвінків і повідомлень.

Антизлодій:

Антизлодій допоможе знайти мобільний пристрій у разі його втрати або крадіжки і при необхідності віддалено стерти з нього конфіденційну інформацію.

Управління Антизлодій виробляється надсиланням СМС на пристрій.

Антизлодій дозволяє:

– Включити опцію блокування телефону після перезавантаження.

– Заблокувати телефон з вимогою ввести пароль розблокування.

– Ввести обмеження на кількість помилок при введенні пароля розблокування.

– Розблокувати телефон за допомогою СМС.

– Отримати GPS-координати пристрою у вигляді посилання на Google Maps.

– Віддалено стерти дані у телефоні та на SD-карті.

– Включити на пристрої гучний звуковий сигнал і заблокувати екран.

– Задати власний текст, який буде відображатися на екрані заблокованого пристрою.

– Створити список з номерів близьких Вам людей, які будуть отримувати повідомлення про зміну SIM-карти на загубленому пристрої. З цих номерів можна буде управляти Антизлодієм і в тому числі розблокувати телефон в тому випадку, якщо ви забули пароль розблокування.

URL-фільтр:

Фільтр убезпечить користувача мобільного пристрою з вбудованим браузером Android від небажаних інтернет-ресурсів. З його допомогою можна заблокувати доступ відразу до всього списку рекомендованих і потенційно небезпечних сайтів або вибрати певні категорії:

– Наркотики.

– Відомі джерела вірусів.

– Нецензурна лексика.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

- Тероризм.
- Насильство.
- Зброя.
- Сайти для дорослих та інше.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Вона складається з наступних функціональних блоків:

- Проактивний захист. Завдяки використанню методів розширеної евристики дозволяє захистити дані як від відомих, так і раніше невідомих загроз.
- Персональний брандмауер. Здійснює перевірку усіх вхідних та вихідних повідомлень згідно заданим правилам.
- Централізоване управління. Програма дозволяє управляти та змінювати параметри безпеки з єдиної консолі.
- Автоматичні оновлення. Забезпечує ефективний захист мобільних пристроїв від усіх видів загроз.
- Захист від загроз. Усі програми, файли, папки та карти пам'яті SD, а також приховані файли і процеси постійно перевіряються на наявність інфікованих файлів.
- Антиспам модуль. Дозволяє визначити надійні контакти за допомогою чорних та білих списків, які можна настроїти, або просто заблокувати невідомі номери.
- Контроль пересилання даних у роумінгу та вартість дзвінків. Здійснює контроль вартості вихідних дзвінків та блокує завантаження даних у роумінгу.
- Аудит безпеки. Проводить діагностику всіх важливих функцій телефону, включаючи стан батареї, вільний простір на диску та запущені процеси.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

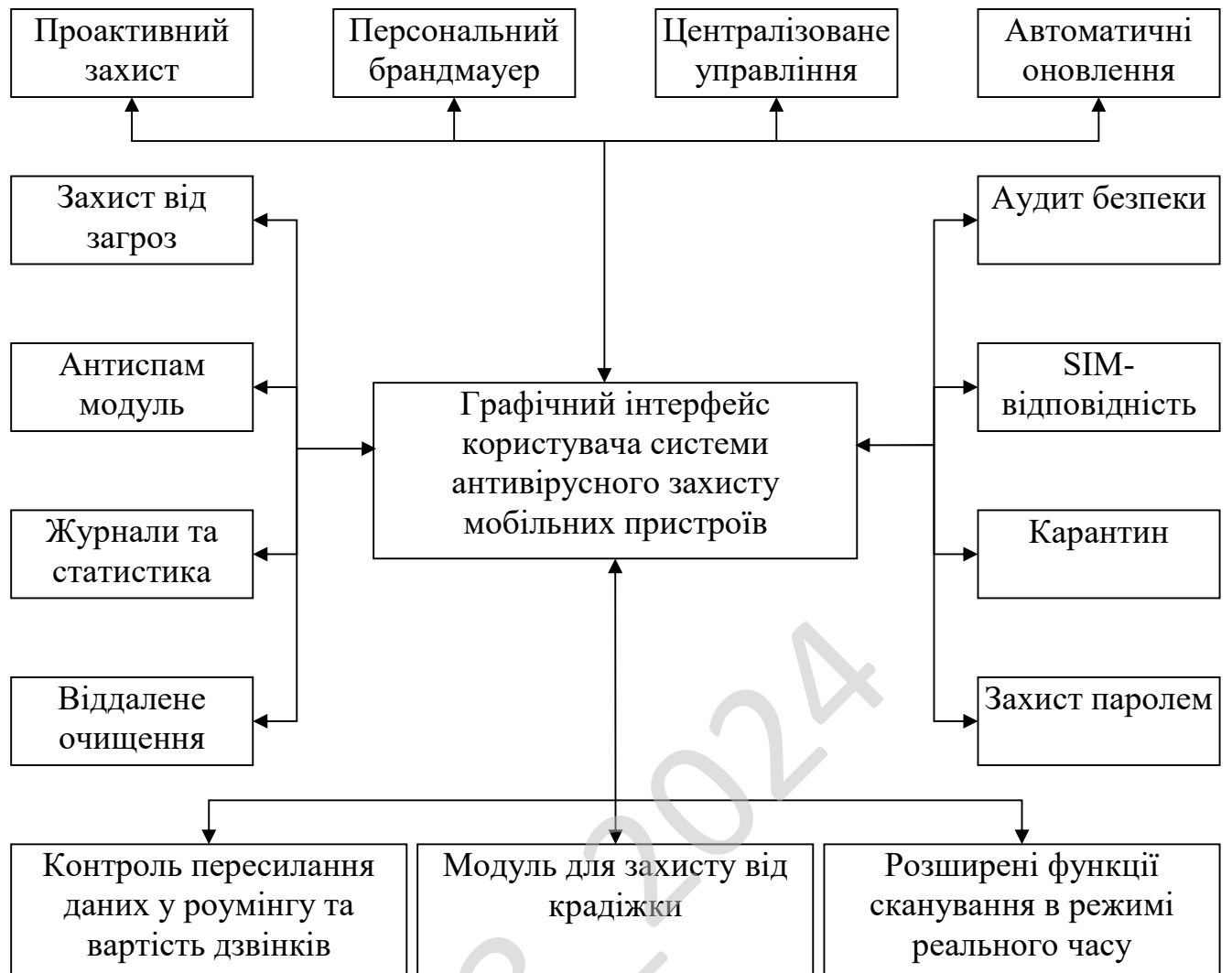


Рисунок 3.2 – Функціональна схема системи

– SIM-відповідність. Дозволяє визначити список довірених SIM-карти та отримувати повідомлення в разі несанкціонованої заміни SIM, у тому числі номер телефонної карти, IMEI та IMSI.

– Журнали та статистика. Надають інформацію про виконані завдання сканування, результати аудиту безпеки, роботу антиспам модуля, сканування в режимі реального часу та статус захисту.

– Модуль для захисту від крадіжки. Багаторівнева система безпеки дозволяє користувачам зберігати повний контроль над даними, що зберігаються у телефоні, у разі втрати або крадіжки мобільного пристрою.

– Карантин. Функція, що дозволяє видалити або ізолювати інфіковані файли в захищеному безпечному контейнері з можливістю подальшого їх відновлення.

– Віддалене очищення. У випадку надзвичайної ситуації видаляє усі дані, включаючи контакти, повідомлення та інформацію зі змінних носіїв пам'яті за допомогою простих команд SMS.

– Захист паролем. Забезпечує парольний захист мобільних пристроїв від несанкціонованої деінсталяції антивірусного програмного забезпечення або втручань у систему захисту.

– Розширені функції сканування в режимі реального часу. Можливість сканування і очищення інтегрованих пристроїв пам'яті та змінних носіїв інформації. Усі файли, вхідні або вихідні зі змінних носіїв інформації та через різні види підключень (Bluetooth, Wi-Fi) перевіряються на наявність загроз.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку.

Після початку роботи розробленого ПЗ ми потрапляємо до головного вікна ПЗ звідки можемо потрапити до налаштування ПЗ, журналу роботи ПЗ, довідкової інформації та перейти до вікна сканування ПЗ.

Далі можемо провести оновлення антивірусної бази сигнатур через антивірусну база та переглянути список наявних сигнатур з поясненнями. Також можна перейти до сканування (антивірус) з подальшою перевіркою файлів, блокування роботи вірусу, лікування файлів (видалення коду вірусу) та переглядом звіту роботи антивіруса.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків:

- Ініціалізація ПЗ.
- Інтеграція до ядра ОС.
- Запит – ПЗ інтегровано?
- Підключення модулів файлової підсистеми та файлу налаштувань ПЗ.
- Спроба доступу до функцій файлової підсистеми.
- Запит – є доступ?
- Очікування запиту користувача початку сканування.
- Запит WM_CLOSE?
- Запит – початок сканування?
- Підпрограма сканування ОС.
- Запит – оновлення ПЗ?
- Виведення вікна оновлення сигнатур ПЗ.
- Завантаження сигнатур ПЗ.
- Запит WM_CLOSE?
- Є помилки (запит).
- Виведення списку помилок.
- Звільнення ресурсів ПЗ.

На рисунку 4.2 наведено блок-схему підпрограми сканування ОС. Її робота складається з виконання наступних кроків:

- Запит моніторингу змін?

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

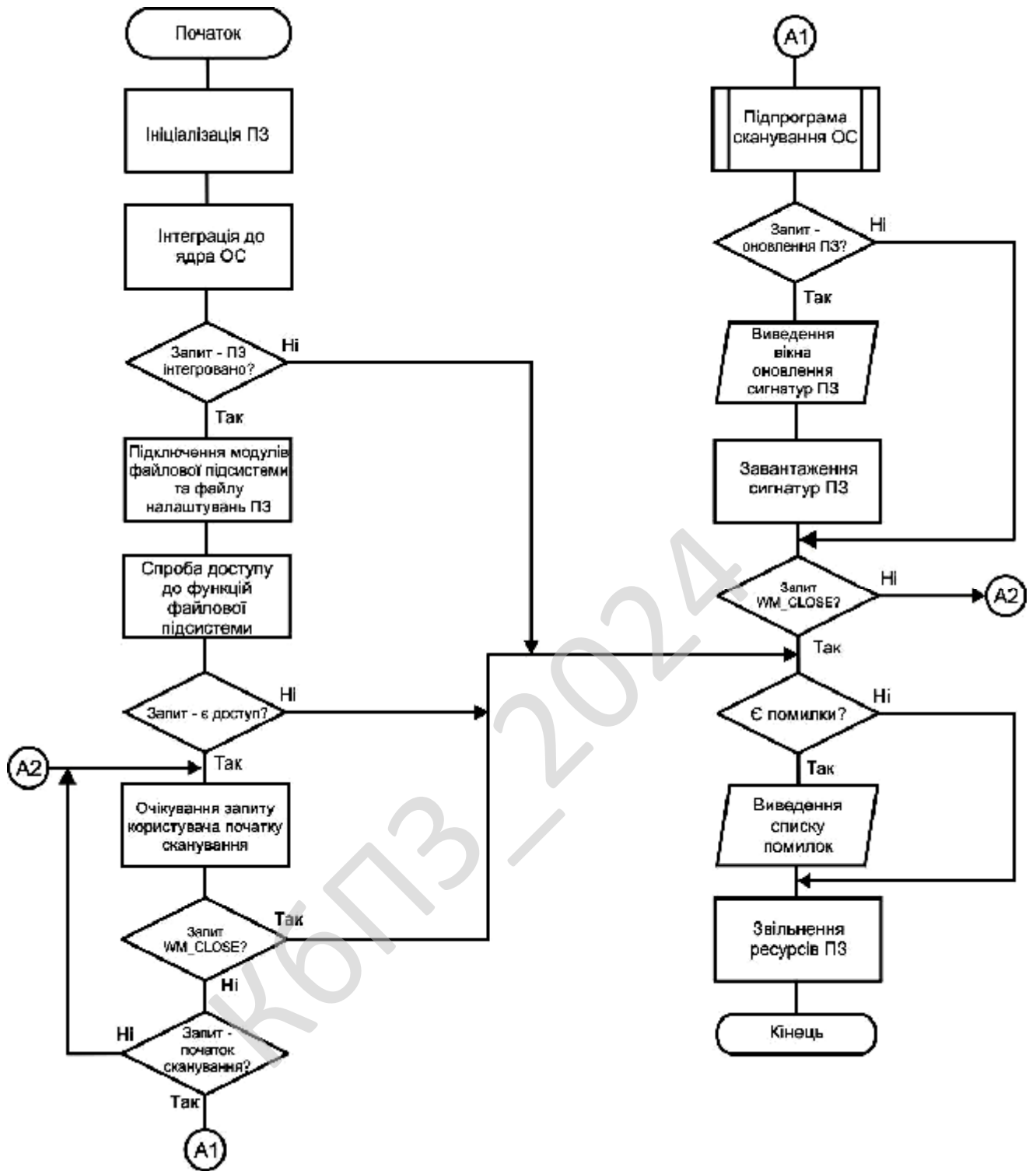


Рисунок 4.1 – Блок-схема основної програми

- Завантаження списку файлів файлової підсистеми.
- Формування списку файлів файлової підсистеми.
- Порівняння файлів.

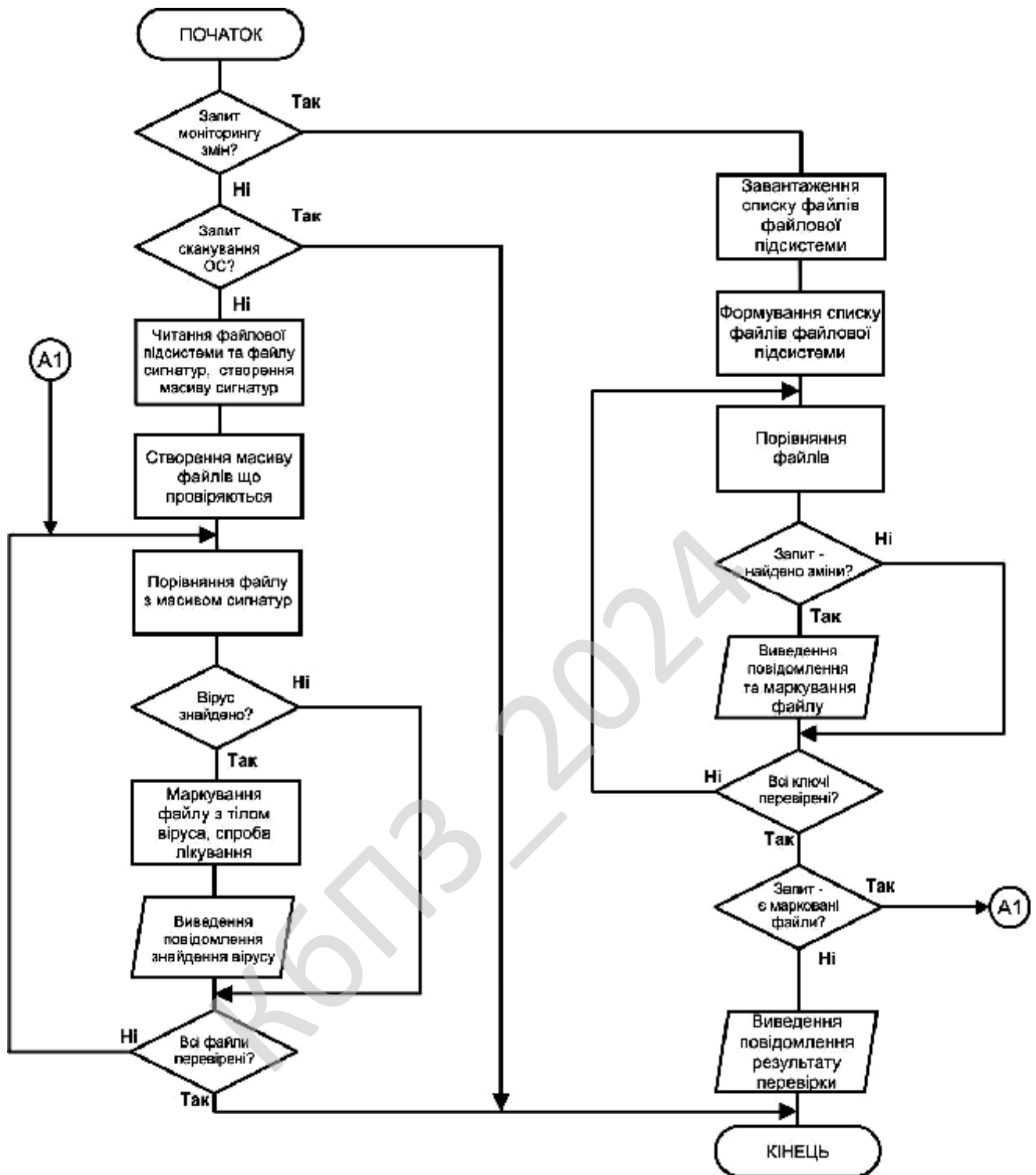


Рисунок 4.2 – Блок-схема підпрограми сканування ОС

- Запит – знайдено зміни?
- Виведення повідомлення та маркування файлу.
- Всі ключі перевірені (запит).

- Запит – є марковані файли?
- Запит сканування ОС?
- Читання файлової підсистеми та файлу сигнатур, створення масиву сигнатур.
- Створення масиву файлів що перевіряються.
- Порівняння файлу з масивом сигнатур.
- Вірус знайдено (запит).
- Маркування файлу з тілом вірусу, спроба лікування.
- Виведення повідомлення знайдення вірусу.
- Всі файли перевірені (запит).
- Виведення повідомлення результату перевірки.

Опис алгоритмів функціонування системи

Розглянемо розроблений вихідний код порівняння двох файлів по вмісту використовуємо тільки модуль Windows – головний алгоритм порівняння сигнатури з файлом який перевіряється. Вихідний код дозволяє зрівняти два файли по змісту.

Якщо є відмінності – видає false, якщо файли однакові – true. Ціль написання даної функції – уникнути використання додаткових модулів delphi, таких як sysutils, classes, тому що в Windows Mobile вони працюють украй нестабільно.

Додаткові функції: fileexists(перевірка наявності файлу), Tempdir(довідаємося тимчасову папку windows mobile), Createtemporaryfile (створення тимчасового файлу на диску, якщо це потрібно).

```
function Tempdir: string;
{функція повертає шлях до папки тимчасових файлів}
var Dir: array[0..MAX_PATH - 1] of char;
begin
  GettempPath(Sizeof(Dir), Dir);
  Result := Dir;
end;
function Fileexists(const Filename: pchar): Bool;
{імпортуємо функції перевірки наявності файлу з модуля sysutils}
```

```

function Fileage(const Filename: pchar): Integer;
var
  Handle: Thandle;
  Finddata: TWin32Finddata;
  Localfiletime: Tfiletime;
type
  Longrec = packed record
    case Integer of
      0: (Lo, Hi: Word);
      1: (Words: array[0..1] of Word);
      2: (Bytes: array[0..3] of Byte);
    end;
begin
  Handle := Findfirstfile(Pchar(Filename), Finddata);
  if Handle <> INVALID_HANDLE_VALUE then
    begin
      Windows.Findclose(Handle);
      if (Finddata.dwfileattributes and FILE_ATTRIBUTE_DIRECTORY) = 0 then
        begin
          Filetimetolocalfiletime(Finddata.ftlastwritetime, Localfiletime);
          if Filetimetodosdatettime(Localfiletime, Longrec(Result).Hi,
            Longrec(Result).Lo) then Exit;
        end;
      end;
      Result := -1;
    end;
begin
  result := false;
  if filename = '' then exit;
  Result := Fileage(Filename) <> -1;
end;

function Createtemporaryfile(Filename: string): string;
{створення тимчасової копії порівнюваних файлів,
 повертає ім'я свіжоспеченого файлу.}
const S: string = '_1we23yuiopasdfghj45zxcvbnmqw2452uiopasdfghjk
  1252vbnm1234567890';

var i, N: integer;
X: string;
label A;
begin
  Randomize;
  A:

```

					ВКРБ-125.24.0039.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		45

```

X := '';
for i := 0 to 7 do {генеруємо ім'я файлу довжиною 8 знаків}
begin
  N := Random(Length(S)+1);
  if N = 0 then Goto A;
  X := X + S[N];
end;
X := Tempdir + X + '.tmp';
if Fileexists(pchar(X)) = true then goto a else
begin
  if Copyfile(pchar(Filename), pchar(X), true) = true then
    Result :=X else Result := '';
end;
end;
{ функція порівняння }
function Comparefiles(File1, File2: string; Createtempfile: Boolean): Boolean;
{Createtempfile - параметр задає, чи необхідно копіювати порівнювані
файли чи ні. Зроблене для запобігання помилок, якщо файл,
приміром використовується запущений exe файл}
var F1, F2: file;
  B1, B2: array[0..1023] of Char;
  i1,i2:Integer;
begin
Result:=false;
  if (Fileexists(pchar(File1)) = false) or (Fileexists(pchar(File2)) = false) then
    Exit; {якщо один з файлів відсутній, то виходимо}
  if Createtempfile = true then
{якщо треба - створюємо тимчасові копії в папці Temp}
  begin
    File1 := Createtemporaryfile(File1);
    File2 := Createtemporaryfile(File2);
  end;
  Assign(F1, File1);
  Assign(F2, File2);
  Reset(f1, 1);
  Reset(f2, 1);
  if Filesize(f1) <> Filesize(f2) then
{якщо розміри файлів не збігаються, то вони(файли) у кожному разі не ідентичні}
  begin
    Closefile(F1);
    Closefile(F2);
    if Createtempfile = true then

```

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

begin
  Deletefile(pchar(File1));{забираємо сміття за собою}
  Deletefile(pchar(File2));
end;
Exit;
end;
repeat
{повторюємо операції поки файл не закінчиться}
  Blockread(F1, B1, Sizeof(B1),i1);
  Blockread(F2, B2, Sizeof(B2),i2);
{блочно читаємо й порівнюємо блоки}
{як тільки попадуться два, що різняться блоку, відразу виходимо, result:=false}
  if B1 <> B2 then
    begin
      Result := false;
      Closefile(F1);
      Closefile(F2);
      if Createtempfile = true then
        begin
          Deletefile(pchar(File1));
          Deletefile(pchar(File2));
        end;
      Exit;
    end else Result := true;
until Eof(F2);{кінець файлу}
Closefile(F1);
Closefile(F2);
  if Createtempfile = true then
{якщо ми створювали тимчасові копії, то їх потрібно вилучити}
  begin
    Deletefile(pchar(File1));
    Deletefile(pchar(File2));
  end;
end;

```

При початковій стадії проектування було з'ясовано, що написання програми буде проходити винятково за допомогою Remote API (RAPI).

RAPI у делфі реалізується використанням бібліотеки RAPI.DLL і інтерфейсу бібліотеки RAPI.pas.

Розглянемо виклик і роботу бакалаврської програми. Для виклику ПЗ необхідно виконати рядок коду:

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

```
device=Createobject ("AB_WRAPI.Device").
```

Необов'язкові параметри: `Errorcode` (число) – код помилки; `Errorstring` (рядок) – опис помилки. У ці попередньо створені параметри програма поверне код і опис помилки.

Якщо, повертається значення (тип `Boolean`) неправда або 0 – програма працює, якщо істина або будь-яке інше значення – програма не працює.

Розглянемо основний розроблений (`Device`) кореневий клас для керування КПК, його властивості й методи:

- `Connect()` As `Boolean` – підключення інтерфейсу RAPI;
- `Disconnect()` As `Boolean` – відключення;
- `Isconnected([Statusstring As String])` As `Boolean` – істина, якщо підключений;
- `Runcmd(cmd_line As String, parameters As String, [Errorcode As Long], [Errorstring As String])` As `Boolean` – запускає зазначену в `cmd_line` програму в КПК із параметрами з `parameters`;
- `Terminate()` – завершення роботи програми, зручно застосовувати перед обнулінням змінної `device=""`.

Розроблений `Remoteregistry` клас для керування реєстрами КПК, властивості й методи:

- `Createkey(Regkey As String, Valuename As String, Value, [dwtype As Long = -1], [Errorcode As Long], [Errorstring As String])` As `Boolean` створити ключ `Regkey`, якщо `Valuename""` те буде створене значення `Value`;
- `Deletekey(Regparentkeys As String, Subkeytodelete As String, [Errorcode As Long], [Errorstring As String])` As `Boolean` вилучити ключ `Regkey` з вітки `Regparentkeys`;
- `Readkey(Regkey As String, Valuename As String, [Errorcode As Long], [Errorstring As String])` читати значення `Valuename` з `Regkey`;
- `Setvalue(Regkey As String, Valuename As String, Value, [dwtype As Long = -1], [Errorcode As Long], [Errorstring As String])` As `Boolean` установити значення `Value` з іменем `Valuename` у вітці `Regkey`;

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

– Deletevalue(Regparentkeys As String, Valuenametodelete As String, [Errorcode As Long], [Errorstring As String]) As Boolean видалити значення з іменем Valuenametodelete у вітці Regkey.

При створенні ключа/значення створюються по всій ієрархії рядка Regkey, якщо відсутні, а якщо ні, то помилки не відбувається Regkey повинен починатися з "HKEY_CLASSES_ROOT", "HKEY_CURRENT_USER", "HKEY_LOCAL_MACHINE" – інших віток реєстру КПК ні, інакше відбудеться помилка.

– Fileexistondevice(Filename As String, [lhandle As Long], [Errorcode As Long = 0], [Errorstring As String]) As Boolean перевіряє існування файлу Filename у КПК;

– Copyfile(Existingfilename As String, Newfilename As String, [Replace As Boolean False], [Errorcode As Long], [Errorstring As String]) As Boolean копіює Existingfilename в Newfilename у КПК, якщо у Replace = 0 то заміняє, інакше – помилка;

– Movefile(Existingfilename As String, Newfilename As String, [Replace As Boolean = False], [Errorcode As Long], [Errorstring As String]) As Boolean переміщає Existingfilename в Newfilename у КПК, якщо Replace=0 то заміняє, інакше – помилка;

– Deletefile(Filetodevice As String, [Errorcode As Long], [Errorstring As String]) As Boolean видаляє Filetodevice у КПК;

– Createdirectory(Directoryname As String, [Errorcode As Long], [Errorstring As String]) As Boolean створює Directoryname у КПК, батьківські папки кінцевої в Directoryname повинні існувати;

– Removedirectory(Directoryname As String, [Errorcode As Long], [Errorstring As String]) As Boolean видаляє Directoryname у КПК;

– Copytodevice(Filetopc As String, Filetodevice As String, [Replace As Boolean = True], [Errorcode As Long], [Errorstring As String]) As Boolean копіює Filetopc у КПК із іменем Filetodevice, якщо Replace=0 то заміняє, інакше – помилка;

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

– Copytopс(Filetodevice As String, Filetopс As String, [Replace As Boolean = True], [Errorcode As Long], [Errorstring As String]) As Boolean копіює Filetodevice у РС із іменем Filetopс, якщо Replace=0 те заміняє, інакше – помилка;

– Getfilesize(Filename As String, [Errorcode As Long], [Errorstring As String]) As Long повертає розмір файлу Filename у КПК;

– Addfiletoexchange(Filetodevice As String, Filetopс As String, [Direction As Integer = 1], [Replace As Boolean = False], [Errorcode As Long], [Errorstring As String]) As Boolean додати файл Filetodevice/Filetopс до колекції підготовлених до обміну Direction – напрямок обміну, 1 – копіювати Filetopс із РС в Filetodevice у КПК 2 – копіювати Filetodevice із КПК в Filetopс у РС;

– Replace: якщо Replace=0 то заміняє, інакше – помилка;

– Filestoexchange Direction As Integer – напрямок обміну, Filetodevice As String – ім'я файлу в КПК;

– Filetopс As String – ім'я файлу в РС Replace As Boolean якщо Replace=0 то заміняє, інакше – помилка;

– Size As Long після обміну в Size буде записаний розмір фінального файлу Successfully As Boolean якщо Successfully=0 то обмін не зроблений;

– Exchange([Direction As Integer], [Errorcode As Long], [Errorstring As String]) As Boolean запускає обмін файлами з колекції Filestoexchange Clearfilestoexchange() очищає колекцію Filestoexchange;

– Lasterrorcode() As Long повертає код останньої помилки;

– Lasterrorstring() As String повертає опис останньої помилки.

Розглянемо копіювання файлів у бакалаврській програмі. Для копіювання файлів застосовується функція Copyfile. Формат її такий:

Copyfile(Вихідний_файл,файл_запису,Прапор_перезапису);

де: Вихідний_файл – повний шлях і назва файлу, який копіюється КПК;

Файл_запису – повний шлях і назва файлу, куди копіюється вихідний файл;


```

Indir:= 'c:\AB\output\dat.rar';
// каталог вихідних файлів
Outdir:= c:\AB'; // каталог файлів запису
Copyfile(Pchar(Indir+'1.txt'),Pchar(Outdir+'1.txt'),false);
Copyfile(Pchar(Indir+'2.txt'),Pchar(Outdir+'2.txt'),false);
end;

```

Розглянемо розроблений код (з використанням бібліотеки RAPI.DLL).

```

procedure TForm1.Button1Click(Sender: TObject);
var
// оголошення змінних
stemplate:Tstringlist;
Hdes:THandle;
o1:POverlapped;
Byteswritten:DWORD;
begin
stemplate:=Tstringlist.Create;
stemplate.Loadfromfile('test.xml');
cerapiinit;
// створення й копіювання файлу
Hdes := Ccreatefile ('\ipaq file store\test.xml', Generic_Write,
                    File_Share_Write,nil, Create_Always,0,0);
Cwritefile(Hdes,Pchar(stemplate.Text),Length(stemplate.Text),
           Byteswritten,@o1);
Cclosehandle(Hdes);
Cerapiuninit;
    showmessage('Скопійоване: '+inttostr(Byteswritten) + ' байт');
//виведення результату копіювання
end;

```

На основі цього коду й враховуючи особливості копіювання файлів на КПК я написав функцію для копіювання файлу з параметрами.

```

Function COPYFILE (Dest,Source:Pchar;
                  Nomessage: boolean=false) :integer;
var Filehdl,Cehandle,Handle: THandle;
Cefilename:Pwidechar; Res:integer;
Boolres:boolean; Buffercount,Readcount:cardinal;
buffer:array [1..1024] of char;
writesbyte:cardinal; Msgstr:String;
begin
    Res:=0;
    Handle:=Loadlibrary('rapi.dll');
// Підключення бібліотеки

```

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

    if Handle<>0 then
        begin
// Створення файлу
            Cewritefile := GetProcAddress(Handle, 'Cewritefile');
            Ccreatefile := GetProcAddress(Handle, 'Ccreatefile');
// Створення директорії
            Ccreatedirectory := GetProcAddress(Handle, 'Ccreatedirectory');
// Одержання адресного простору
            Cclosehandle := GetProcAddress(Handle, 'Cclosehandle');
            Cerapiiinit := GetProcAddress(Handle, 'Cerapiiinit');
            Cerapiuninit := GetProcAddress(Handle, 'Cerapiuninit');
// Перевірка стану
            Cegetlasterror := GetProcAddress(Handle, 'Cegetlasterror');
            if Assigned(Cerapiiinit) then Cerapiiinit;
// Виділення пам'яті
            Getmem(Cefilename,512);
            Msgstr:=Extractfiledir(Dest);
            Cefilename:=Stringtowidechar(Msgstr,Cefilename,length(Dest)+1);
            Boolres:=Ccreatedirectory(Cefilename,nil);
//Перетворення типу даних
            Cefilename:=Stringtowidechar(Dest,Cefilename,length(Dest)+1);
// Створення файлу з параметрами
            Cehandle:=Ccreatefile(Cefilename,GENERIC_READ or GENERIC_WRITE,
                                0,nil, CREATE_ALWAYS,
                                FILE_ATTRIBUTE_NORMAL,0);
            if (Cehandle<>INVALID_HANDLE_VALUE) then
                begin
                    Buffercount:=1024;Readcount:=1024;
                    Filehdl:=Createfile(Pchar(Source),GENERIC_READ,0,nil,OPEN_EXISTING,
                                        FILE_ATTRIBUTE_NORMAL,0);
                    if Filehdl<>INVALID_HANDLE_VALUE then
                        begin
                            while Readfile(Filehdl,Buffer,Buffercount,Readcount,nil)
                                and (Readcount<>0) do
                                Cewritefile(Cehandle,buffer,Readcount,writesbyte,nil);
                            Closehandle(Filehdl);
                        end
                    else
                        begin
// Виникли помилки під час виконання
                            Res:=Getlasterror;
                            Msgstr:='Не вдалося відкрити файл'+Source;

```

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

    // Якщо є помилки вивести її на екран
if Nomessage=false then Showmessage(Msgstr+'. Код помилки:'+Inttostr(Res));
end; //if Filehdl...
// Закриття файлу звільнення пам'яті
  Cefclosehandle(Cehandle);
  Freememory(Cefilename);
  end
else
  begin
// Якщо є помилки вивести її на екран
    Res:=Cegetlasterror;
    Msgstr:='Не вдалося створити файл '+Cefilename+' на КПК';
    if Nomessage=false then Showmessage(Msgstr+'. Код помилки:'+Inttostr(Res));
    end;//if Cehandle...
    if Assigned(Cerapiuninit) then Cerapiuninit;
// Звільнення ресурсів виділеної бібліотеки
    Freelibrary(Handle);
    end//if Handle<>0...
else
  begin
    Res:=1;
// Якщо бібліотека RAPI не знайдено вивести повідомлення
if Nomessage=false then Showmessage('Не вдалося відкрити бібліотеку RAPI.');
```

Для видалення тимчасових файлів використовувалася функція Deletefile у вигляді коду: if not Deletefile('c:\AB\input\tmpdat.rar') then Showmessage ('Помилка видалення');

Також при роботі з файлами було необхідне додавати рядок тексту у файл роботи програми (LOG файл).

Розглянемо реалізацію:

```

proceduretform1.Button1Click(Sender: TObject);
Varf:Textfile;
// оголошення файлової змінної для LOG файлу
st:String;
// строкова змінна
begin
// прив'язка назви файлу до файлової змінної
```

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

Assignfile(f, 'c:\AB\log\log.txt');
{ Команда Assignfile здійснює прив'язку рядка шляху файлу до файлової змінної. Усі
подальші операції з файлової змінної автоматично здійснюються із зазначеним
файлом.}
{$I-}
// відключення контролю помилок введення / виведення
Reset(f);
// відкриття файлу для читання
{$I+}
// включення контролю введення / виведення
ifioresult<>0 then
// якщо є помилка відкриття, то ...
begin
Showmessage('Помилка відкриття файлу log.txt');
Exit;
// вихід із процедури при помилці відкриття файлу
end;
While not EOF(f) do
// поки не кінець файлу робити цикл:
begin
    Readln(f, st);
// читати з файлу рядок
    Showmessage(st);
// виведення рядка
end;
Closefile(f);
// закрити файл
end;

```

{\$I-} і {\$I+} є директивами компіляторів, що в цьому місці відповідно слід відключити й включити контроль помилок введення / виведення. У цьому випадку при невдалій спробі відкриття файлу log.txt(файл відсутній або відкритий для запису іншою програмою) програма не видасть аварійної помилки й продовжить виконання даної процедури. Ця властивість корисна для обробки всіх можливих випадків у роботі програми.

Ioresult – змінна, яка зберігає в собі код помилки останньої операції введення / виведення. Якщо вона дорівнює нулю, то остання операція була успішно виконана.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

EOF(Файл) – функція, що повертає ознаку кінця файлу, показує, досягнувся чи ні кінець відкритого файлу.

Readln (Файл, Змінна) – процедура зчитування змінної з файлу. На відміну від команди Read робить зчитування рядка із символом, що завершується, перекладу рядка під кодами 13 і 10 (клавіша Enter).

Closefile(Файл) – процедура закриття раніше відкритого файлу.

Клас Tmemorystream забезпечує збереження даних в адресному просторі. При цьому методи доступу до цих даних залишаються тими ж, що й при роботі з файловими потоками. Це дозволяє використовувати адресний простір для зберігання проміжних результатів роботи ПЗ, а також за допомогою стандартних методів здійснювати обмін даними між пам'яттю й іншими фізичними носіями.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм PRESENT – окремий випадок SP-мережі й складається з 31 раунду. Довжина блоку становить 64 біта, а ключі підтримуються в 2 варіантах, 80- і 128-бітні. Такого рівня захисту повинно цілком вистачати для низькозахищених додатків, звичайно використовуваних для розгортання на основі тегів, а крім того, що важливіше, PRESENT багато в чому збігається своїми конструктивними особливостями з потоковими шифрами проекту estream, заточеними на ефективну реалізацію в залозі, що дозволяє нам адекватно порівнювати їх. Кожний з 31 раундів складається з операції XOR, щоб увести ключ K_i для $1 \leq i \leq 32$, де K_{32} використовується для «відбілювання» ключа, лінійної побітової перестановки й нелінійного шару заміщення (або, попросту говорячи, збільшення стійкості шифрування. Нелінійний шар використовує роздільні 4-бітні S-блоки, які застосовуються паралельно 16 раз на кожному раунді. Шифр, описаний псевдо-кодом представлено на рисунку 4.3.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

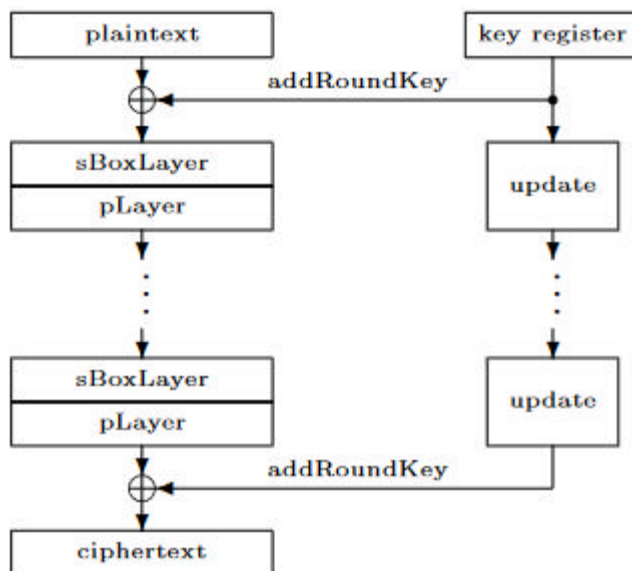


Рисунок 4.3 – Шифр PRESENT

Тепер кожна стадія визначається по черзі. Обґрунтування конструкції наведені нижче, а біти всюди нумеруються з нуля, починаючи із правого в блоці або слові.

Додавання раундового ключа (addRoundKey)

Заданий раундовий ключ $K_i = k_{63}^i \dots k_0^i$, де $1 \leq i \leq 32$, а так само поточний стан $b_{63} \dots b_0$. Додавання раундового ключа до поточного стану відбувається за модулем 2 ($b_j = b_j \oplus k_j^i$, де $0 \leq j \leq 63$).

Шар S-блоків (sBoxlayer)

Використовувані в PRESENT S-блоки відображають 4-бітні блоки в 4-бітні блоки. Дія цього блоку в шістнадцятковій системі числення наведене в наступній таблиці.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Рисунок 4.4 – Дія блоку в шістнадцятковій системі числення

Отже, регістр ключа зрушується на 61 позицію вліво, 4 крайніх лівих біта, що пройшли через S-блок і round_counter значення і складається за модулем 2 з бітами $k_{19}k_{18}k_{17}k_{16}k_{15}$ з К з найменшим значущим бітом з round_counter праворуч.

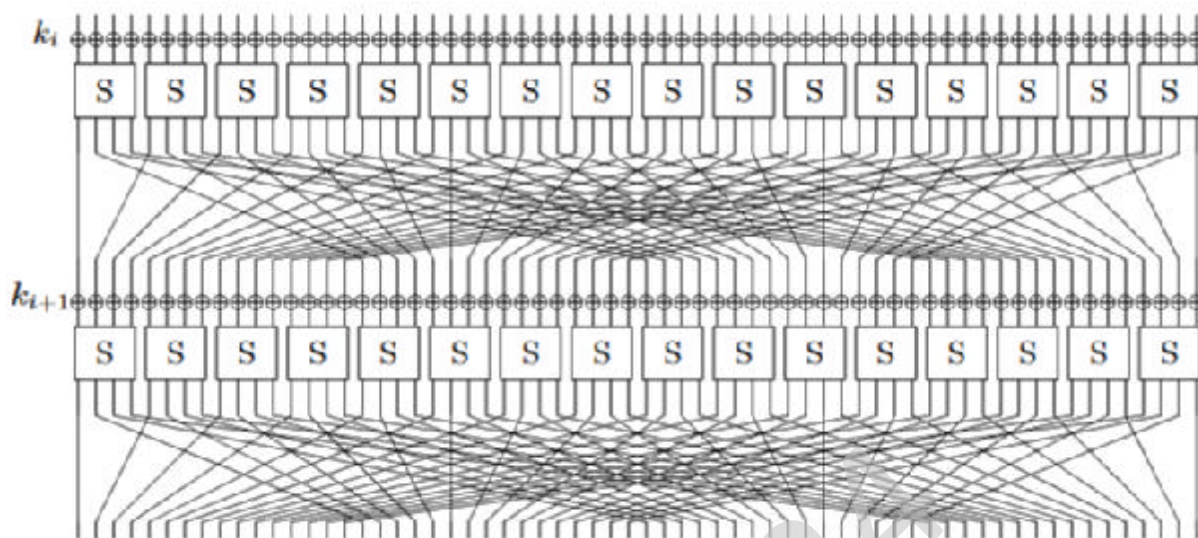


Рисунок 4.6 – Перетворення ключа (The key schedule)

Перетворення ключа для 128-бітного алгоритму можна знайти в додатку до оригінальної статті.

Конструктивні особливості PRESENT

Крім безпеки й ефективної реалізації, основне досягнення PRESENT – його простота. по цьому не дивно, що схожі проекти були прийняті в інших обставинах, і навіть були використані як навчальний посібник для студентів. У даній секції ми обґрунтуємо рішення, прийняті нами при проектуванні PRESENT. Однак, у першу чергу, опишемо очікувані прикладні вимоги.

Цілі й середовище застосування

При проектуванні блокового шифру, застосовного в жорстко обмежених оточеннях, важливо зрозуміти, що ми не створюємо блоковий шифр, який, неодмінно, буде застосовний у багатьох ситуаціях – для цього існує AES. Навпаки, ми націлені на досить специфічне застосування, для якого AES не підходить. Вищесказане визначає нам наступні характеристики:

– Шифр буде реалізований «у залізі»

– Додатки будуть вимагатися лише для регулювання рівня безпеки. Отже, 80-бітний ключ буде здоровим розв'язком. Відзначимо, що такої ж позиції дотримуються розроблювачі потокових шифрів проекту eSTREAM.

– Додатки не припускають шифровки великої кількості даних. Таким чином, реалізація може бути оптимізована для продуктивності або простору без внесення занадто великих змін.

– У деяких застосуваннях можлива ситуація, що ключ буде зафіксований при виробництві. У такому випадку, не треба буде змінювати ключ пристрою (що може вилитися в атаки з маніпуляцією ключем).

– Фізичний обсяг пристрою буде першим пріоритетом, після безпеки, що спричинить обмеження на пікові й середні споживання енергії, і, отже, зрушить швидкодія в область низькопріоритетних параметрів.

– У пристроях, що вимагають найбільш ефективного використання фізичного простору, блоковий шифр найчастіше зможе лише шифрувати дані (encryption-only mode). Таким чином, він зможе бути використаний у запит-відповідь (challenge-response) протоколах авторизації, і, при дотриманні контролю стану, може бути використаний для шифровки й дешифрування переговорів із пристроєм, використовуючи режим лічильника.

Виходячи з таких міркувань, розв'язали створити PRESENT як 64-бітний блоковий шифр із 80-бітним ключем. Шифровка й дешифрування, у цьому випадку, мають приблизно схожі фізичні вимоги. Маючи можливість підтримувати як шифрацію, так і дешифрацію, PRESENT буде компактніше, чим підтримуючий лише шифрацію AES. А у випадку encryption-only виконання, наш шифр виявиться й зовсім понад-легко. Суб-ключі що шифрують будуть обчислюватися на ходу.

У літературі є безліч прикладів атак компромісу між часом, датою й пам'яттю, або атак з використанням парадокса днів народження при шифровці великих обсягів даних. Однак, дані атаки залежать тільки від параметрів шифру й

не використовують внутрішню структуру. Наша мета полягає в тому, щоб ці атаки були кращим, що можуть застосувати проти нас. Атаки стороннього каналу й атаки з безпосереднім зломом чипа загрожують PRESENT тією самою мірою, як і іншим криптографічним примітивам. Однак для ймовірних застосувань, помірні вимоги безпеки роблять вигоду, одержувану зловмисником на практиці, досить обмеженою. В оцінці ризиків, подібні погрози не сприймаються як істотний фактор.

Перестановочний шар

При виборі шару змішування ключа, наша увага до апаратної ефективності вимагає наявності лінійного шару, який може бути реалізований з мінімальною кількістю керуючих елементів (наприклад, транзисторів. це приводить до побітової перестановки. Приділяючи увагу простоті, ми вибрали регулярну бітову перестановку, що допомагає провести прозорий аналіз безпеки.

КБПЗ - 2024

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з інтуїтивно зрозумілого інтерфейсу. Робота розробленого ПЗ проходить у автоматичному режимі. Користувачу необхідно завантажити розроблене ПЗ на телефон та запустити програму на виконання. У розділі сканування натиснути клавішу «QUICK SCAN».

Після цих дій ПЗ автоматично почне перевіряти файлоу підсистему. Якщо програма знаходить вірус вона спочатку проводить маркування файлу потім проведе спробу його вилікувати шляхом видалення частини файлу з тілом вірусу. Якщо файл вилікувати неможливо він маркується та переноситься до інфікованої папки.



Рисунок 5.1 – Головне вікно програми

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

На рисунку 5.2 зображено форму авторського права, де відображені дані розробника.

Було обрано Shareware умову розповсюдження. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (неповнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

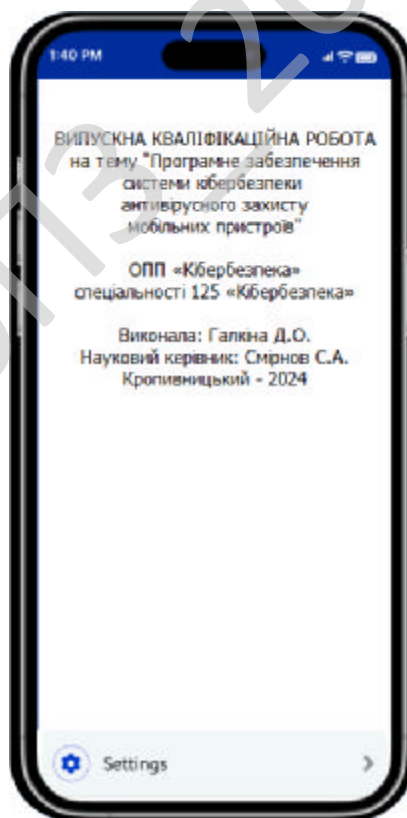


Рисунок 5.2 – Довідка

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки антивірусного захисту мобільних пристроїв.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем антивірусного захисту мобільних пристроїв.

– Досліджена система антивірусного захисту мобільних пристроїв.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки антивірусного захисту мобільних пристроїв.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання антивірусного захисту мобільних пристроїв.

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

Програма реалізована на мові високого рівня RAD Studio Delphi. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки антивірусного захисту мобільних пристроїв. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи кібербезпеки й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм PRESENT.

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.

КБПЗ-2024

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
2. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
3. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
4. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
5. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
6. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
7. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
8. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.
9. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
11. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.
12. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) *Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

15. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, Cracow, Poland, 22-25 September 2021. P. 414-418

16. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

17. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages 44-58.

19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-

feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

21. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

22. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

23. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

24. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

25. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

26. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

27. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-

quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

28. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

29. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 125-136.

30. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

31. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings Volume 2608*, 2020, Pages 646-660.

32. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

33. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

34. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during

Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

35. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

36. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

37. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

38. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P. 707-712.

39. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.701-706.

40. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.399-405.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

41. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT-2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.*

42. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019, P. 129-134.*

43. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.*

44. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.*

45. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.*

46. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884.*

47. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.*

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

48. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

49. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ПШПІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

50. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

51. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

52. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп’ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв’язку*, 2023, вип. 2(72), С. 170-178.

					ВКРБ-125.24.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					ВКРБ-125.24.0039.00.00.ТЗ			
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>	<i>Галкіна Д.О.</i>				<i>Програмне забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	<i>Смірнов С.А.</i>					<i>Б</i>	<i>1</i>	<i>6</i>
<i>Н. Контр.</i>	<i>Коваленко А.С</i>				<i>ЦНТУ КБ-21-3СК</i>			
<i>Затв.</i>	<i>Смірнов О.А.</i>							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки антивірусного захисту мобільних пристроїв.

2 Підстава для розробки

Підставою для розробки служить завдання на випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 136-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи кібербезпеки антивірусного захисту мобільних пристроїв.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-125.24.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи кібербезпеки з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи кібербезпеки антивірусного захисту мобільних пристроїв;
- 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.

					ВКРБ-125.24.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

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 мобільних пристроях з ОС Android і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

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 Android.

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 RAD Studio Delphi.

					ВКРБ-125.24.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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 – 72 аркуші.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					ВКРБ-125.24.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

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.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 8.06.2024 р.

					ВКРБ-125.24.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти
_____ Смірнов С.А.

*Програмне забезпечення системи кібербезпеки антивірусного захисту
мобільних пристроїв*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 39

Літера: РП

Кропивницький – 2024 року

ФАЙЛ ПРОЕКТУ ПЗ - FILE_PROTECTION.DPR

```
program File_Protection; // Початок файлу проекту ПЗ
// Програмне забезпечення системи кібербезпеки антивірусного захисту мобільних
// пристроїв
// Розробив студент Галкіна Дар'я Олександрівна
// гр. КБ-21-ЗСК, Кропивницький 2024
Uses // Підключення бібліотек
  Forms,
  Messages, Graphics,
  Windows,
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Report.pas' {Form2},
  Unit3 in 'fset.pas' {Form3},
  Protection_splash in 'U7.pas' {Form_Protection_Splash}.
{$R *.res} // Підключення ресурсів
begin
  Application.Initialize; //Ініціалізація ПЗ
try
  // Створення Splash форми
  Form_Protection_Splash:=TForm_Protection_Splash.Create(Application);
  Form_Protection_Splash.Show; // виведення Splash
  Form_Protection_Splash.Update; // оновлення Splash
  // відправлення системного повідомлення
  SendMessage(Form_Protection_Splash.Handle,WM_MY,0,'Start');
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  // відправлення системного повідомлення
  SendMessage(Form_Protection_Splash.Handle,WM_MY,0,'End');
  Application.CreateForm(TForm5, Form4);
  Finally Form_Protection_Splash.free; // звільнення Splash
end;
  Application.Run; // виведення
end.
```

ФАЙЛ ФОРМИ ПЗ - FORM1.PAS

```

unit Form1; // Початок файлу форми

// Програмне забезпечення системи кібербезпеки антивірусного захисту мобільних
// пристроїв

// Розробив студент Галкіна Дар'я Олександрівна
// гр. КВ-21-ЗСК, Кропивницький 2024

interface // Секція інтерфейсу (зовнішніх визначень модуля)

Uses // Підключення бібліотек
SysUtils, IniFiles, Forms, registry, Windows, Dialogs, Menus,
// Всі бібліотеки які починаються зі Id**, нестандартні
IdAntiFreezeBase, IdAntiFreeze, IdCustomTCPServer, IdCustomHTTPServer,
IdHTTPServer, IdBaseComponent, IdComponent, IdTCPConnection, IdTCPClient,
IdHTTP, StdCtrls, ExtCtrls, Controls, ComCtrls, Classes, idContext,
IdAuthentication, srchunit, FileCtrl, TLHelp32, ShellApi, psapi, Graphics,
StrUtils, shlobj, idstack, mapi, Messages;

const //введення констант
  WM_ICONTRAY = WM_USER + 1; // системне повідомлення

Type // Типи даних які визначаються
  pfilestream = ^TFileStream;

Type // Типи даних які визначаються
  TProtection = class (TForm)
    N8:      TMenuItem; // Елемент меню
    N9:      TMenuItem;
    N10:     TMenuItem;
    N11:     TMenuItem;
    N12:     TMenuItem;
    N18:     TMenuItem;
    N19:     TMenuItem;
    N20:     TMenuItem;
    N21:     TMenuItem;
    TabSheet19: TTabSheet;
    avpath:  TEdit; // компонент текстового введення даних
    Label28: TLabel; // текстові дані
    avscanbrowse: TButton; // компонент кнопки
    servname: TEdit; // компонент текстового введення даних
    Label5:  TLabel; // текстові дані
    artyp:   TRadioGroup;
    Button4: TButton; // компонент кнопки
    Button5: TButton; // компонент кнопки
    aredit:  TEdit; // компонент текстового введення даних
    GroupBox1: TGroupBox; // Група компонентів
    agshow:  TRadioGroup;
    firstz:  TCheckBox; // компонент введення (так/ні)
    pc:      TPageControl;
    TabSheet2: TTabSheet;
    Label11: TLabel; // текстові дані
    Label2:  TLabel; // текстові дані
    Label6:  TLabel; // текстові дані
    always: TComboBox;
    addb:    TButton; // компонент кнопки
    Button8: TButton; // компонент кнопки
    _http:  TCheckBox; // компонент введення (так/ні)
    Status: TStatusBar;
    maxcon: TEdit; // компонент текстового введення даних
    Label4: TLabel; // текстові дані
    TabSheet7: TTabSheet;
    GroupBox4: TGroupBox; // Група компонентів
    Label10: TLabel; // текстові дані
    Label11: TLabel; // текстові дані
    Label7:  TLabel; // текстові дані
    netbase: TEdit; // компонент текстового введення даних
  end;

```

```

locbase: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
Button2: TButton; // КОМПОНЕНТ КНОПКИ
Button1: TButton; // КОМПОНЕНТ КНОПКИ
TabSheet3: TTabSheet;
Label3: TLabel; // текстові дані
msk: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
TabSheet4: TTabSheet;
N13: TMenuItem;
N14: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
TabSheet5: TTabSheet;
id: TIdHTTP;
TabSheet1: TTabSheet;
Button3: TButton; // КОМПОНЕНТ КНОПКИ
logwindow: TRichEdit;
hstest: TButton; // КОМПОНЕНТ КНОПКИ
hstop: TButton; // КОМПОНЕНТ КНОПКИ
Label8: TLabel; // текстові дані
over: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
_ver: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
notdel: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
http: TIdHTTPServer;
TabSheet6: TTabSheet;
GroupBox3: TGroupBox; // Група компонентів
_uselochttp: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
Label19: TLabel; // текстові дані
_lochttpport: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
deli: TButton; // КОМПОНЕНТ КНОПКИ
ignor: TComboBox;
OpenDialog: TOpenDialog;
PopupMenu2: TPopupMenu; // Контексне меню ПЗ
MenuItem1: TMenuItem; // Елемент меню
MenuItem2: TMenuItem; // Елемент меню
txt: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
rport: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
GroupBox5: TGroupBox; // Група компонентів
hauth: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
hlogin: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
addit: TButton; // КОМПОНЕНТ КНОПКИ
Label9: TLabel; // текстові дані
proxy: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
rauth: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
GroupBox6: TGroupBox; // Група компонентів
TabSheet9: TTabSheet;
Memo2: TMemo;
Label25: TLabel; // текстові дані
Label26: TLabel;
Label27: TLabel;
detect: TButton; // КОМПОНЕНТ КНОПКИ
findweb: TButton; // КОМПОНЕНТ КНОПКИ
fullsearch: TCheckBox; // КОМПОНЕНТ ВВЕДЕННЯ (ТАК/НІ)
sresults: TListBox;
sstop: TButton; // КОМПОНЕНТ КНОПКИ
where: TComboBox;
PopupMenu1: TPopupMenu; // Контексне меню ПЗ
N1: TMenuItem; // Елемент меню
N2: TMenuItem; // Елемент меню
Label20: TLabel; // текстові дані
hpass: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
Label22: TLabel; // текстові дані
huser: TEdit; // КОМПОНЕНТ ТЕКСТОВОГО ВВЕДЕННЯ ДАНИХ
Label23: TLabel; // текстові дані
IdAntiFreeze1: TIdAntiFreeze;
TabSheet8: TTabSheet;
imp: TButton; // КОМПОНЕНТ КНОПКИ
exp: TButton; // КОМПОНЕНТ КНОПКИ
Button10: TButton; // КОМПОНЕНТ КНОПКИ

```

```

SaveDialog: TSaveDialog;
TabSheet10: TTabSheet;
GroupBox11: TGroupBox; // Група компонентів
run:      TEdit; // компонент текстового введення даних
multin:   TCheckBox; // компонент введення (так/ні)
mdir:     TComboBox;
chkpath:  TButton; // компонент кнопки
zapinet:  TCheckBox; // компонент введення (так/ні)
GroupBox9: TGroupBox; // Група компонентів
Label21: TLabel; // текстові дані
fnoerr:   TCheckBox; // компонент введення (так/ні)
fevent:   TCheckBox; // компонент введення (так/ні)
fnoerrlocb: TCheckBox; // компонент введення (так/ні)
ferr:     TCheckBox; // компонент введення (так/ні)
GroupBox7: TGroupBox; // Група компонентів
flnoerr:  TCheckBox; // компонент введення (так/ні)
flevent:  TCheckBox; // компонент введення (так/ні)
flnoerrlocb: TCheckBox; // компонент введення (так/ні)
flerr:    TCheckBox; // компонент введення (так/ні)
GroupBox10: TGroupBox; // Група компонентів
newhttp:  TCheckBox; // компонент введення (так/ні)
err:      TCheckBox; // компонент введення (так/ні)
errd:     TCheckBox; // компонент введення (так/ні)
prpass:   TEdit; // компонент текстового введення даних
plogin:   TEdit; // компонент текстового введення даних
zapnet:   TCheckBox; // компонент введення (так/ні)
getrec:   TButton; // компонент кнопки
traymenu: TPopupMenu; // Контексне меню ПЗ
end;

Var // Визначення змінних
started: tdatetime = -1;
marked: boolean = False;
updateok: boolean = True;
Colors: array[False..True] of TColor;
Windowsdir: string = '';
Tempdir: string = '';
sessiontotal: integer = 0;
LogFile: TextFile;
uselogs: boolean = False;
maxlog: longint = 2097152;
shutdown: boolean = False;
trayst: boolean = False;
nt: boolean = False;
PROTECTION: TProtection;
logfilepath: string = '';
isauto: boolean = False;
st: TSearchThread;
logonlyerrors: boolean = False;
rewr: boolean = False;
mypath: string = '';

Var // Визначення змінних
nowintray: boolean = False;
buildstamp: string = '100';
NotifyIconData: TNotifyIconData;

//-----
implementation // Секція реалізації
//-----
Uses // Підключення бібліотек
sntpsex, blksock;

{$R *.dfm} // Підключення ресурсів модуля

Var // Визначення змінних
stop: boolean = False;

function TProtection.Usersselectedfolder(title: string): string;

```

```

Var // Визначення змінних
Bi: _browseinfoa;
idl: Pitemidlist;
Str: array[0..260 - 1] of char;
begin
  Fillchar(Bi, Sizeof(Bi), 0);
  Bi.hwndowner:= PROTECTION.Handle;
  Bi.lpsztitle:= Pchar(title);
  Bi.ulflags:= 0;
  idl:= Shbrowseforfolder(Bi);
  Shgetpathfromidlist(idl, @Str[0]);
  Result:= Str;
end;

procedure TProtection._uselogclick(Sender: Tobject);
begin
  _maxlog.Enabled:= _uselog.Checked;
  _rewr.Enabled:= _uselog.Checked;
  _maxlog.Color:= colors[_uselog.Checked];
  logpath.Enabled:= _uselog.Checked;
  logpath.Color:= colors[_uselog.Checked];
  logp.Enabled:= _uselog.Checked;
  report.Enabled:= _uselog.Checked;
  ;
end;

procedure TProtection.overclick(Sender: Tobject);
begin
  _ver.Enabled:= over.Checked;
  _ver.Color:= colors[over.Checked];
end;

procedure TProtection.makehttpdir(ver: string);
Var // Визначення змінних
s: Tsearchrec;
list, index: Textfile;
begin
  createdir(mypath + 'http');
  assignfile(index, mypath + 'http\index.html');
  rewrite(list);
  rewrite(index);
  writeln(index, '<html>');
  writeln(index, '<head>');
  writeln(index, '</head>');
  writeln(index, '<body>');
  writeln(index, 'Current version: ', ver, '<br>');
  writeln(index, '<table border="1">');
  writeln(index, '<tr>');
  writeln(index, '<th>File</th>');
  writeln(index, '<th>Size</th>');
  writeln(index, '<th>Date</th>');
  writeln(index, '</tr>');
  writeln(list, '[' + ver + ']');
  if findfirst(locbase.Text + msk.Text, faanyfile + fahidden +
    fasysfile + faarchive + fareadonly, s) = 0 then
    repeat
      if ignor.Items.IndexOf(s.Name) <> -1 then
        continue;
      writeln(list, '+' + s.Name, ', 0');
      Log_PROTECTION('Locb: Копіюємо бази: ' + s.Name, 2, 0);
      if copyfileadv(locbase.Text + s.Name, mypath + 'http\' + s.Name) then
        begin
          writeln(index, Format('<tr><td><a
href="\"%s\"">%s</a><br></td><td>%d</td><td>%s</td></tr>',
            [s.Name, s.Name, S.Size, atetostr(Filedatetodatettime(S.Time))]);
          Log_PROTECTION('Locb: Успішно скопійована: ' + s.Name, 2, 1);
          end
        else
          Log_PROTECTION('Locb: Помилка копіювання: ' + s.Name, 255, 3);

```

```

    until findnext(s) <> 0;
    Sysutils.Findclose(s);
    closefile(list);
    writeln(index, '</table>');
    writeln(index, '</body>');
    writeln(index, '</html>');
    closefile(index);
end;

procedure TProtection.deliclick(Sender: TObject);
begin
    if ignor.Itemindex <> -1 then
        ignor.Items.Delete(ignor.Itemindex);
end;

procedure TProtection.addiclick(Sender: TObject);
Var // Визначення змінних
    s: string;
begin
    s:= trim(inputbox('Список виключень', 'Ім'я', ''));
    if s <> '' then
        if ignor.Items.IndexOf(s) = -1 then
            ignor.Items.Append(s);
end;

function TProtection.pupdate(fs, fd, mask: string): boolean;
Var // Визначення змінних
    S: Tsearchrec;
    all, new, bad, ignor: integer;
begin
    if (trim(netbase.Text) = '') or (trim(locbase.Text) = '') then
        begin
            Log_PROTECTION('Укажіть шляху для відновлення!', 255, 3);
            Result:= False;
            status.Panels[1].Text:= 'Помилка.';
            exit;
        end;
    if (not directoryexists(Sysutils.Excludetrailingpathdelimiter(fd))) then
        begin
            Log_PROTECTION(Format('Каталог локальних баз "%s" не існує. Перевірте
настроювання.',
                [Sysutils.Excludetrailingpathdelimiter(fd)]), 255, 3);
            Result:= False;
            status.Panels[1].Text:= 'Помилка.';
            exit;
        end;
    status.Panels[0].Text:= '';
    status.Panels[1].Text:= '';
    Result:= True;
    all:= 0;
    new:= 0;
    ignor:= 0;
    bad:= 0;
    fs:= Pchar(includetrailingbackslash(fs));
    fd:= Pchar(includetrailingbackslash(fd));
    Log_PROTECTION('Використовуємо сервер: ' + fs, 0, 1);
    if not directoryexists(fs) then
        begin
            Result:= False;
            Log_PROTECTION('Помилка підключення (каталог не існує): ' + fs, 255, 3);
            status.Panels[1].Text:= 'Помилка.';
            exit;
        end
    else
        Log_PROTECTION('Підключений: ' + fs, 0, 0);

        if Findfirst(fs + string(mask), fahidden + fasysfile + fareadonly +

```

```

faarchive, S) = 0 then
repeat
  Inc(all);
Log_PROTECTION('Знайдений: ' + s.Name, 0, 0);
  if (not fileexists(Pchar(fd + s.Name))) or
    (always.Items.IndexOf(s.Name) <> -1) then

    begin
      if copyfileadv(fs + s.Name, fd + s.Name) then
        begin
          Log_PROTECTION('Скопійований: ' + s.Name, 0, 2);
          Inc(new);
        end
      else
        begin
          Log_PROTECTION('Збій копіювання: ' + fs + s.Name + ' в ' + fd + s.Name,
            255, 3);
          Inc(bad);
        end;
        status.Panels[0].Text:= Format('Скопійоване: %d Помилки: %d', [new,
          bad]);

        end
      else
        begin
          Log_PROTECTION('Уже встановлений: ' + s.Name, 1, 0);
          Inc(ignor);
        end;
        until Findnext(s) <> 0;
        Sysutils.Findclose(s);
        Log_PROTECTION('Нових баз: ' + Inttostr(new), 0, 2);
        Log_PROTECTION('Усього баз: ' + Inttostr(all), 0, 0);
        status.Panels[1].Text:= Format('Усього: %d Нових: %d Пропущене: %d',
          [all, new, ignor]);
        if all = 0 then
          begin
            Log_PROTECTION('Бази не знайдені!', 255, 3);
            Result:= False;
          end;

          if multin.Checked then
            if not multicopy then
              Result:= False;
          if bad > 0 then
            Result:= False;

          if Result = True then
            begin
              Getupdateflag;
              if zapnet.Checked then
                Getvirrec;
            end;
          end;

          procedure TProtection.alwupclick(Sender: TObject);
          Var // Визначення змінних
            s: string;
          begin
            s:= trim(inputbox('Завжди оновляти', 'Ім'я', ''));
            if s <> '' then
              if always.Items.IndexOf(s) = -1 then
                always.Items.Append(s);

          end;

          procedure TProtection.Button8Click(Sender: TObject);
          begin
            if always.Itemindex <> -1 then
              always.Items.Delete(always.Itemindex);
          end;

```

```

procedure TProtection.httpcommandget(Acontext: Tidcontext;
  Arequestinfo: Tidhttprequestinfo; Aresponseinfo:
  Tidhttpresponseinfo);
Var // Визначення змінних
  Localdoc: string;
begin
  Localdoc:= arequestinfo.Document;
  localdoc:= Ansireplacetext(localdoc, '/', '\');
  if localdoc = '\' then
    localdoc:= 'index.html';
  localdoc:= copy(localdoc, pos('\', localdoc) + 1, 255);
  localdoc:= mypath + 'http\' + localdoc;
  if fileexists(localdoc) then
    begin
      Aresponseinfo.Responseno:= 200;
      Aresponseinfo.Contentlength:= filesizebyname(localdoc);
      if ansilowercase(extractfileext(localdoc)) = '.html' then
        Aresponseinfo.ContentType:= 'text/html'
      else
        Aresponseinfo.ContentType:= 'application/untyped-data';
      Aresponseinfo.Servefile(Acontext, localdoc);
    end
  else
    begin
      Aresponseinfo.Responseno:= 404; // Помилка 404
      Aresponseinfo.Contenttext:=
'<html><head><title>Error</title></head><body><h1>Не знайдено!</h1><br> Запит:
<b>'
        + Arequestinfo.Document +
        Aresponseinfo.Contentlength:= Length(Aresponseinfo.Contenttext);
        Aresponseinfo.ContentType:= 'text/html';
      end;
    end;
end;

procedure TProtection.pauthclick(Sender: Tobject);
begin
  plogin.Enabled:= pauth.Checked;
  ppass.Enabled:= pauth.Checked;
  plogin.Color:= colors[pauth.Checked];
  ppass.Color:= colors[pauth.Checked];
end;

procedure TProtection._httpclick(Sender: Tobject);
begin
  addit.Enabled:= _http.Checked;
  deli.Enabled:= _http.Checked;
  ignor.Enabled:= _http.Checked;
  ignor.color:= colors[_http.Checked];
end;

procedure TProtection._uselochttpclick(Sender: Tobject);
begin
  _lochttpport.Enabled:= _uselochttp.Checked;
  testserv.Enabled:= _uselochttp.Checked;
  maxcon.Enabled:= _uselochttp.Checked;
  _lochttpport.Color:= colors[_uselochttp.Checked];
  maxcon.color:= colors[_uselochttp.Checked];
  if _uselochttp.Checked then
    begin
      _http.Checked:= True;
      _http.Enabled:= False;
    end
  else
    _http.Enabled:= True;
  end;
end;

```

```

procedure TProtection.hauthclick(Sender: TObject);
begin
  hlogin.Enabled:= hauth.Checked;
  hpass.Enabled:= hauth.Checked;
  hlogin.Color:= colors[hauth.Checked];
  hpass.Color:= colors[hauth.Checked];
end;

procedure TProtection.Label25Mouseenter(Sender: TObject);
begin
  (Sender as TLabel).Color:= clred;
  (Sender as TLabel).Font.Color:= clwhite;
end;

procedure TProtection.Label25Mouseleave(Sender: TObject);
begin
  (Sender as TLabel).Color:= clbtnface;
  (Sender as TLabel).Font.Color:= clblack;
end;

procedure TProtection.Formkeydown(Sender: TObject; var Key: word; Shift:
Tshiftstate);
begin
  if key = vk_f1 then
    pc.Activepage:= tabsheet9
  else
    if key = vk_f2 then
      button3.Click
    else
      if key = vk_f3 then
        htest.Click;
end;

function TProtection.getver: string;
Var // Визначення змінних
  reg: Tregistry;
  s: Tstringlist;
begin
  Result:= '433';
  try
    reg:= Tregistry.Create;
    reg.Rootkey:= hkey_local_machine;
    if reg.Keyexists('\Software\Doctor Web, Ltd.\Dr.Web') then
      begin
        reg.Openkey('\Software\Doctor Web, Ltd.\Dr.Web', True);
        s:= Tstringlist.Create;
        reg.Getkeynames(s);
        s.Sort;
        if s.Count > 0 then
          Result:= s[s.Count - 1];
        Result:= copy(ansireplacetext(Result, '.', ''), 1, 3);
        s.Free;
      end;

    finally
      reg.Free;
    end;
end;

function TProtection.findnewver(l: Tstringlist): integer;
Var // Визначення змінних
  s: Tstringlist;
  i: integer;

function isnumber(d: string): boolean;
Var // Визначення змінних
  n: integer;
begin

```

```

    Result:= False;
    if trim(d) = '' then
        exit;
    for n:= 1 to length(d) do
        if not (d[n] in ['0'..'9']) then
            exit;
    Result:= True;
end;

begin
    Result:= -1;
    if l.Count = 0 then
        exit;
    s:= Tstringlist.Create;
    for i:= 0 to l.Count - 1 do
        if Isnumber(l[i]) then
            s.Append(l[i]);
    for i:= 0 to s.Count - 1 do
        if strtointdef(s[i], -1) > Result then
            Result:= Strtoint(s[i]);
    s.Free;
end;

procedure TProtection.Formcreate(Sender: TObject);
begin
    colors[False]:= clbtnface;
    colors[True]:= clwindow;
    memo2.Lines.Append('-----');
    memo2.Lines.Append('Складання від: ' + stamp);
end;

procedure TProtection.findwebclick(Sender: TObject);
begin
    enablesearchcontrols(False);
    sresults.Items.Clear;
    st:= Tsearchthread.Create(where.Text + ':', fullsearch.Checked, advs.Checked);
end;

procedure TProtection.sstopclick(Sender: TObject);
begin
    if (st <> nil) then
        st.Terminate;
end;

procedure TProtection.addbclick(Sender: TObject);
begin
    popupmenu1.Popup(PROTECTION.Left + addb.Left, PROTECTION.Top + addb.Top +
    addb.Height);
end;

procedure TProtection.NlClick(Sender: TObject);
begin
    opendirlog.Filter:= '*.vdb|*.vdb|*.*|*.*';
    opendirlog.FileName:= '';
    if directoryexists(extractfiledir(locbase.Text)) then
        opendirlog.Initialdir:= locbase.Text
    else
        opendirlog.Initialdir:= 'C:\';
    if opendirlog.Execute then
        if always.Items.IndexOf(Extractfilename(opendirlog.FileName))=-1 then
            always.Items.Append(Extractfilename(opendirlog.FileName));
end;

procedure TProtection.Menuitem1Click(Sender: TObject);
begin
    opendirlog.Filter:= '*.vdb|*.vdb|*.*|*.*';
    opendirlog.FileName:= '';
    if directoryexists(extractfiledir(locbase.Text)) then
        opendirlog.Initialdir:= locbase.Text

```

```

else
  opendialog.Initialdir:= 'C:\';

if opendialog.Execute then
if ignor.Items.IndexOf(Extractfilename(opendialog.FileName))=-1 then
ignor.Items.Append(Extractfilename(opendialog.FileName));

end;

procedure TProtection.additclick(Sender: TObject);
begin
popupmenu2.Popup(PROTECTION.Left + addit.Left, PROTECTION.Top +
addit.Top+addit.Height);
end;

procedure TProtection.htestclick(Sender: TObject);
Var // Визначення змінних
  r: boolean;
begin
  logwindow.Clear;
  Log_PROTECTION('Інтернет відновлення...', 0, 0);
  Syntax;
  if locbase.Text = '' then
    begin
Log_PROTECTION('Укажіть шлях до локальних баз!', 255, 3);
  Showmessage('Укажіть шлях до локальних баз!');
Log_PROTECTION('Відновлення пройшло з помилками!', 255, 3);
  if errd.Checked then
    begin
Showmessage('Відновлення пройшло з помилками! Подивитися звіт.');
```

```

  if agshow.Itemindex <> 2 then
    gofromtray;
  end
  else
  if nowintray then
    begin
  Showmessage('Відновлення пройшло з помилками!');
  if agshow.Itemindex <> 2 then
    gofromtray;
  end;
  pc.Activepage:= tabsheet2;
  exit;
  end;

  if not inetupd then
    begin
Log_PROTECTION('Відновлення пройшло з помилками!', 255, 3);
  if errd.Checked then
    begin
Showmessage('Відновлення пройшло з помилками! Подивитися звіт.');
```

```

  if agshow.Itemindex <> 2 then
    gofromtray;
  end
  else
  if nowintray then
    begin
  Showmessage('Відновлення пройшло з помилками!');
  if agshow.Itemindex <> 2 then
    gofromtray;
  end;
  end
  else
  begin
  if nowintray then
    begin
  Showmessage('Відновлення успішне завершене!');
  if agshow.Itemindex = 1 then
    gofromtray;
  end;
  end;
  end;
end;

```

```

    if arun.Checked then
    begin
        if not ask.Checked then
            r:= True
        else
            r:= messagedlg('Відновлення успішне заверрене. Запустити програму?',
                mtconfirmation, [mbok, mbcancel], 0) = idok;
        if r then
            if wait.Checked then
                runandwait(run.Text)
            else
                winexec(Pchar(run.Text), sw_show);
        end;
    if autoexit.Checked then
    begin
        if not askex.Checked then
            r:= True
        else
            r:= messagedlg('Вийти із програми?', mtconfirmation,
                [mbok, mbcancel], 0) = idok;
        if r then
            begin
                application.Processmessages;
                Close;
            end;
        end;
    end;
end;

end;

procedure TProtection.Label31Click(Sender: TObject);
begin
shellapi.Shellexecute(handle, 'open', '', '', sw_show);

end;

procedure TProtection.Button10Click(Sender: TObject);
begin
    dir.Items.Savetofile(tempdir + 'servers');
if messagedlg('Прийняти зміни в списку серверів?', mtconfirmation, [mbok,
mbcancel], 0) = idok then

    dir.Items.Loadfromfile(tempdir + 'servers');
    Sysutils.deletefile(tempdir + 'servers');
end;

procedure TProtection.impclick(Sender: TObject);
begin
    opendialog.Filter:= '*..*';
    opendialog.FileName:= '';
    opendialog.Initialdir:= mypath;
    if opendialog.Execute then
        dir.Items.Loadfromfile(opendialog.FileName);
end;

procedure TProtection.expclick(Sender: TObject);
begin
    savedialog.Filter:= '*..*';
    if savedialog.Execute then
        dir.Items.Savetofile(savedialog.FileName);
end;

procedure TProtection.browseclick(Sender: TObject);
begin
    opendialog.Filter:= '*.exe|*.exe';
    opendialog.FileName:= '';
    if directoryexists(extractfiledir(lochase.Text)) then
        begin

```

```

opendialog.Initialdir:= loibase.Text;
  end
  else
    opendialog.Initialdir:= 'C:\';

    if opendialog.Execute then
      run.Text:= opendialog.FileName;
    end;

procedure TProtection.arunclick(Sender: TObject);
begin
  ask.Enabled:= arun.Checked;
  browse.Enabled:= arun.Checked;
  run.Enabled:= arun.Checked;
  run.Color:= colors[arun.Checked];
  wait.Enabled:= (autoexit.Checked) and (arun.Checked);
end;

procedure TProtection.autoexitclick(Sender: TObject);
begin
  askex.Enabled:= autoexit.Checked;
  wait.Enabled:= (autoexit.Checked) and (arun.Checked);
end;

function iswin9x: Bool; //перевірка версії ОС
asm
  xor eax, eax
  mov ecx, cs
  xor cl, cl
  jecxz @@quit
  inc eax
  @@quit:
end;

function iswin9xsafe: boolean;
Var // Визначення змінних
  r: Tregistry;
begin
  r:= Tregistry.Create;
  r.Rootkey:= HKEY_LOCAL_MACHINE;
  r.Openkey('SYSTEM\Currentcontrolset\Control\Session Manager
    \Environment', True);
  if R.Valueexists('OS') then
    Result:= r.Readstring('OS') <> 'Windows_NT'
  else
    Result:= iswin9x;
  r.Closekey;
  r.Free;
end;

procedure TProtection.detectclick(Sender: TObject);
begin
  detectproxy(True);
end;

procedure TProtection.Getupdateflag;
Var // Визначення змінних
  Sr: Tsearchrec;
  st: _systemtime;
begin
  if findfirst(loibase.Text + 'drwtoday.vdb', fahidden + fasysfile +
fareadonly + faarchive, sr) = 0 then
    begin
      filetimetosystemtime(sr.Finddata.ftcreationtime, st);
      Log_PROTECTION(format('Останнє відновлення: %d - %d -%d',
[st.wday, st.wmonth, st.wyear]), 0, 2);
    end
  else

```

```

    Sysutils.Findclose(Sr);
end;

procedure TProtection.addoclick(Sender: TObject);
Var // Визначення змінних
    s: string;
begin
    s:= trim(inputbox('Обновляти', 'Адреса', ''));
    if s <> '' then
        begin
            if pos('http://', s) = 0 then
                s:= 'http://' + s;
            other.Items.Append(Ansireplacetext(s, '\', '/'));
        end;
end;

end;

procedure TProtection.deloclick(Sender: TObject);
begin
    if other.Itemindex <> -1 then
        other.Items.Delete(other.Itemindex);
end;

procedure TProtection.downaclick(Sender: TObject);
begin
    other.Enabled:= downa.Checked;
    other.Color:= colors[downa.Checked];
    addo.Enabled:= downa.Checked;
    delo.Enabled:= downa.Checked;
end;

procedure TProtection.Memo2Mousedown(Sender: TObject; Button: Tmousebutton;
    Shift: Tshiftstate; X, Y: integer);
begin
    hidecaret(Memo2.Handle);
end;

procedure TProtection.N3Click(Sender: TObject);
begin
    logwindow.Copytoclipboard;
end;

procedure TProtection.N4Click(Sender: TObject);
begin
    logwindow.Clear;
end;

procedure TProtection.N7Click(Sender: TObject);
begin
    logwindow.Selectall;
    logwindow.Copytoclipboard;
    logwindow.Selstart:= 0;
    logwindow.Selength:= 0;
end;

procedure TProtection.Popupmenu3Popup(Sender: TObject);
begin
    n3.Enabled:= logwindow.Lines.Count > 0;
    n4.Enabled:= logwindow.Lines.Count > 0;
    n7.Enabled:= logwindow.Lines.Count > 0;
end;

procedure TProtection.pcchange(Sender: TObject);
begin
    if pc.Activepage = tabsheet9 then
        hidecaret(memo2.Handle)
end;

```

```

else
if pc.Activepage = tabsheet15 then
  hidecaret(help.Handle)
else
if pc.Activepage = tabsheet19 then
  artypclick(nil);

end;

procedure TProtection.mdelclick(Sender: TObject);
begin
  if mdirs.Itemindex <> -1 then
    mdirs.Items.Delete(mdirs.Itemindex);

end;

function TProtection.multicopy: boolean;
Var // Визначення змінних
  s: Tsearchrec;
  i: integer;
  dl: Tstringlist;
begin
  Result:= True;
  if mdirs.Items.Count = 0 then
    begin
      Result:= False;
      exit;
    end;
  dl:= Tstringlist.Create;
  for i:= 0 to mdirs.Items.Count - 1 do
    if ((directoryexists(Sysutils.Excludetrailingpathdelimiter(mdirs.items[i]))
or
  Forcdirectories(Sysutils.Excludetrailingpathdelimiter(mdirs.items[i])))
and (not samefilename(Sysutils.Excludetrailingbackslash
(locbase.Text), Sysutils.Excludetrailingbackslash
(mdirs.Items[i]))) then
dl.Append(Sysutils.Excludetrailingpathdelimiter(mdirs.items[i]))
  else
    begin
      Log_PROTECTION('Неможливо ініціалізувати каталог: ' +
Sysutils.Excludetrailingpathdelimiter(mdirs.items[i]), 255, 3);
      Result:= False;
      end;

    if findfirst(locbase.Text + msk.Text, faanyfile + fahidden +
fasysfile + faarchive + fareadonly, s) = 0 then
      repeat
        Log_PROTECTION('Копіюємо бази: ' + s.Name, 3, 0);
        for i:= 0 to dl.Count - 1 do
          begin
            if copyfileadv(locbase.Text + s.Name, dl[i] + '\' + s.Name) then
              begin
                Log_PROTECTION('Успішно скопійована в: ' + dl[i] + '\' + s.Name, 3, 1);
              end
            else
              begin
                Log_PROTECTION('Помилка копіювання: ' + dl[i] + '\' + s.Name, 255, 3);
                Result:= False;
              end;
            end;
          until findnext(s) <> 0;
          Sysutils.findclose(s);
          if Result then
            Log_PROTECTION(Format('Успішно створені %d резервних копій баз.',
[dl.Count]),3,2)
          else
            Log_PROTECTION('Помилка при створенні резервних копій.', 255, 3);
            dl.Free;
          end;
end;

```

```

procedure TProtection.mtestclick(Sender: TObject);
begin
  if Multicopy then
    Showmessage('Резервні копії успішно створені.')
  else
    Showmessage('Помилка при створенні резервних копій!');
end;

procedure TProtection.multiiclick(Sender: TObject);
begin
  mdirs.Enabled:= (multii.Checked or multin.Checked);
  mdirs.Color:= colors[(multii.Checked or multin.Checked)];
  madd.Enabled:= (multii.Checked or multin.Checked);
  mdel.Enabled:= (multii.Checked or multin.Checked);
  mtest.Enabled:= (multii.Checked or multin.Checked);
  chkpath.Enabled:= (multii.Checked or multin.Checked);
end;

procedure TProtection.N5Click(Sender: TObject);
Var // Визначення змінних
  s: string;
begin
  s:= Userselectedfolder('Укажіть потрібний шлях');
  if canwriteto(s) then
    begin
      if s <> '' then
        if mdirs.Items.IndexOf(Sysutils.Includetrailingbackslash(s)) = -1
          then
            begin
              if (not samefilename(Sysutils.Excludetrailingbackslash (locbase.Text),
                Sysutils.Excludetrailingbackslash(s))) then
                mdirs.Items.Append(Sysutils.Includetrailingbackslash(s))
              else
                Showmessage(' Не можна вказувати в якості каталогу, каталог локальних баз. ');
            end
          end;
    end;
end;

procedure TProtection.N6Click(Sender: TObject);
Var // Визначення змінних
  s: string;
begin
  s:= trim(inputbox('Каталоги копіювання', 'Шлях', ''));
  if s <> '' then
    if mdirs.Items.IndexOf(s) = -1 then
      begin
        if Canwriteto(s) and (not
          samefilename(Sysutils.Excludetrailingbackslash(locbase.Text),
            Sysutils.Excludetrailingbackslash(s))) then
          mdirs.Items.Append(Sysutils.Includetrailingbackslash(s))
        else
          if messagedlg('Неможливо ініціалізувати каталог. Однаково додати в список?',
            mtconfirmation, [mbok, mbcancel], 0) = idok then
            mdirs.Items.Append(Sysutils.Includetrailingbackslash(s));
          end;
      end;
    end;
end;

procedure TProtection.maddclick(Sender: TObject);
begin
  popupmenu4.Popup(PROTECTION.Left + madd.Left, PROTECTION.Top + madd.Top +
    madd.Height);
end;

procedure TProtection.logpclick(Sender: TObject);
begin
  opendialog.Filter:= '*.log|*.log|*.*|*.*';
  if uselog then

```

```

begin
  opendialog.Initialdir:= extractfiledir(logfilepath);
  opendialog.FileName:= logfilepath;
end
else
begin
  opendialog.Initialdir:= windowsdir;
  opendialog.FileName:= windowsdir + 'drupdate.log';
end;
if opendialog.Execute then
logpath.Text:= opendialog.FileName;

end;

procedure TProtection.reportclick(Sender: Tobject);
begin
  if uselog then
  begin
    system.Flush(logfile);
    copyfile(Pchar(logfilepath), Pchar(changefileext(logfilepath, '.tmp')),
False);
winexec(Pchar('notepad.exe ' + Pchar(changefileext(logfilepath, '.tmp'))),
sw_show);
  end
  else
    Showmessage('Звіт не включений.');
```

РБГЗ-2024

```

end;

function InternetGetConnectedState(lpdwFlags: LPDWORD; dwReserved:DWORD):BOOL;
stdcall; external 'wininet.dll' Name 'InternetGetConnectedState';

function FileSizeByName(const AFilename: string): int64;
begin
  with TFileStream.Create(AFilename, fmOpenRead or fmShareDenyWrite) do
  try
    Result:= Size;
  finally
    Free;
  end;
end;

function InternetConnected: boolean;
Var // Визначення змінних
  dwConnectionTypes: DWORD;
begin
  dwConnectionTypes:= 7;
  if PROTECTION.nonet.Checked then
    Result:= True
  else
    Result:= InternetGetConnectedState(@dwConnectionTypes, 0);
end;

function SyncSysTime;
Var // Визначення змінних
  sn: TSntpSend;
begin
  if not InternetConnected then
  begin
    Log_PROTECTION(' Не виявлене підключення!', 255, 3);
    status.Panels[1].Text:= 'Помилка!';
    Result:= False;
    exit;
  end;
  sn:= TSntpSend.Create;
  sn.TargetHost:= times.Text;
  sn.SyncTime:= not test;
  case socksproxy.ItemIndex
  of
    0:
```

```

begin
  sn.Sock.SocksType:= ST_Socks5;
end;
1:
begin
  sn.Sock.SocksType:= ST_Socks4;
end;
2:
begin
  sn.Sock.SocksType:= ST_Socks4;
  sn.Sock.SocksResolver:= True;
end;
end;
sn.Sock.SocksIP:= proxy.Text;
sn.Sock.SocksPort:= IntToStr(strtointdef(socksport.Text, 1080));
if socksneedpass.Checked then
begin
  sn.Sock.SocksUsername:= socksuser.Text;
  sn.Sock.SocksPassword:= sockspass.Text;
end;
case prot.ItemIndex of
  0:
    Result:= sn.GetSNTP;
  1:
    Result:= sn.GetNTP;
end;
if Result then
begin
  if not Test then
Log_PROTECTION('Успішно синхронізований час із сервера ' + sn.Targethost, 0, 2);
Log_PROTECTION(Formatdatetime('Час, отриман із сервера: dddd, d mmmmm yyyy,
hh:nn:ss:z', sn.NtpTime), 0, 1);
  end
  else
  begin
Log_PROTECTION('Помилка синхронізації часу із сервера ' + sn.Targethost, 255,
3);
Log_PROTECTION('Опис помилки: ' + sn.Sock.Lasterrordesc, 255, 3);
  status.Panels[1].Text:= 'Помилка.';
  end;
  sn.Free;
end;

function copyfileadv(from, tof: string): boolean;
begin
  if from = tof then
  begin
    Result:= False;
    exit;
  end;
  if fileexists(tof) then
  begin
    Result:= Filesetreadonly(tof, False);
    if Result then
Log_PROTECTION(Format('Підготовлений до видалення файл %s', [tof]), 0, 0)
    else
Log_PROTECTION(Format('Немає доступу до файлу %s', [tof]), 255, 3);
    if Result then
      Result:= Windows.Deletefile(Pchar(tof));
    if Result then
Log_PROTECTION(Format('Вилучений раніше існуючий файл %s', [tof]), 0, 0)
    else
Log_PROTECTION(Format('Немає доступу до файлу %s', [tof]), 255, 3);
    end
  else
    Result:= True;
  if Result then
    Result:= Windows.Copyfile(Pchar(from), Pchar(tof), False);
  if Result then

```

```

Log_PROTECTION(Format('Скопійований %s в %s', [from, tof]), 0, 0)
else
Log_PROTECTION(Format('Помилка копіювання %s в %s', [from, tof]), 255, 3);
end;

function httpgetfilename(s: string): string;
Var // Визначення змінних
  i: integer;
begin
  Result:= '';
  for i:= length(s)
  downto 1 do
    if s[i] <> '/' then
      Result:= s[i] + Result
    else
      break;
  end;
end;

function cutfirst(s: string): string;
begin
  Result:= s;
  Delete(Result, 1, 1);
end;

function getprocessfilename(Proc: TPROCESSENTRY32): string;
Var // Визначення змінних
  ph: Thandle;
  md: array[0..1023] of Thandle;
  sz: array[0..255] of char;
  n: dword;
  i: longint;
begin
  Result:= '';
  ph:= Openprocess(PROCESS_QUERY_INFORMATION or PROCESS_VM_READ,
    False, proc.th32Processid);
  if (ph <> INVALID_HANDLE_VALUE) then
    try
      if Enumprocessmodules(ph, (@md), Sizeof(md), n) then
        begin
          for i:= 0 to n div Sizeof(Thandle) - 1 do
            begin
              if Getmodulefilenameex(ph, md[i], Pchar(@sz), MAX_PATH) <> 0 then
                begin
                  if i = 0 then
                    Result:= sz;
                end;
            end;
          end;
        end;
      finally
        Closehandle(ph);
      end;
    end;
end;

procedure .Handleerror(e, l: integer; fn: string; Ex: Exception);

  function oneline(s: string): string;
  begin
    Result:= strutils.Ansireplacetext(s, #13#10, ' ');
  end;

begin
  if ex is Eidsocketerror then
    if Syserrormessage((ex as Eidsocketerror).Lasterror) <> '' then
      Log_PROTECTION(Format('Помилка #%d ''%s''', [(ex as Eidsocketerror).Lasterror,
        Syserrormessage((ex as Eidsocketerror).Lasterror)]), e, l)
    else
      Log_PROTECTION(Format('Помилка #%d ''%s''', [(ex as Eidsocketerror).Lasterror,
        (oneline((ex as Eidsocketerror).Message))]), e, l)
  else

```

```

    if ex is Eidhttpprotocolexception then
Log_PROTECTION(Format('Помилка #d '%s'' при одержанні файлу '%s''',
    [(ex as Eidhttpprotocolexception).Errorcode, oneline(
    (ex as Eidhttpprotocolexception).Message), fn]), e, l)
    else
Log_PROTECTION(Format('Помилка '%s''', [oneline(ex.Message)]), e, l);
end;

function Spider: string;
Var // Визначення змінних
    cp: cardinal;
    pe: TProcessEntry32;
    n: string;
begin
    Result:= '';
    pe.dwSize:= sizeof(TProcessEntry32);
    cp:= CreateToolHelp32Snapshot(TH32CS_SnapProcess, 0);
    try
        if Process32First(cp, pe) then
            repeat
                if NT then
                    n:= getprocessfilename(pe)
                else
                    n:= pe.szExeFile;
                // пошук фалів антивірусу ДрВеб КПК
                if (AnsiUpperCase(ExtractFileName(n)) = 'SPIDER.EXE') or
                    (AnsiUpperCase(ExtractFileName(n)) = 'SPIDERNT.EXE') then
                    if fileexists(ExtractFilePath(n)) then
                        begin
                            Result:= ExtractFilePath(n);
                            break;
                        end;
                    until not Process32Next(cp, pe);
            finally
                CloseHandle(cp);
            end;
            if Result = '' then
                begin
                    if FileExists(ExtractFileDrive(WindowsDir)) then
                        Result:= ExtractFileDrive(WindowsDir);
                    end;
                end;
            end;

procedure .RunAndWait(cmdline: string);
Var // Визначення змінних
    si: STARTUPINFO;
    pi: PROCESS_INFORMATION;
begin
    ZeroMemory(@si, sizeof(si));
    si.cb:= SizeOf(si);
    if not CreateProcess(nil, // створення потоку обробки ActiveSync
        PChar(cmdline), nil, nil, False, 0, nil, nil, si, pi) then
        begin
            ShowMessage('Помилка');
            Exit;
        end;
    WaitForSingleObject(pi.hProcess, INFINITE);
    CloseHandle(pi.hProcess);
    CloseHandle(pi.hThread);
end;

procedure .Syntax;
begin
    socksport.Text:= IntToStr(strtointdef(socksport.Text, 1080));
    _maxlog.Text:= IntToStr(strtointdef(_maxlog.Text, 2097152));
    netbase.Text:= Trim(netbase.Text);
    if netbase.Text <> '' then
        netbase.Text:= Sysutils.Includetrailingbackslash(netbase.Text);
    locbase.Text:= Trim(locbase.Text);

```

```

if locbase.Text <> '' then
  locbase.Text:= Sysutils.Includetrailingbackslash(locbase.Text);
  _lochtppport.Text:= Inttostr(strtointdef(_lochtppport.Text, 16555));
  dir.Text:= Trim(Dir.Text);
  dir.Text:= Ansireplacetext(dir.Text, '\', '/');
  if dir.Text <> '' then
    if dir.Text[length(dir.Text)] <> '/' then
      dir.Text:= dir.Text + '/';
end;

procedure .install(Id, Run: string);
Var // Визначення змінних
  reg: Tregistry;
begin
  try
  reg:= Tregistry.Create;
  reg.Rootkey:= hkey_current_user;
  reg.Openkey('\SOFTWARE\Microsoft\Windows\Currentversion\Run', True);
  reg.WriteString(Id, run);
  Log_PROTECTION('Інсталяція: ' + reg.Currentpath + ', Ідентифікатор: ' +
    id + ', Рядок запуску: ' + run, 0, 2);
  finally
    reg.Free;
  end;
end;

procedure TProtection.deinstall(id: string);
Var // Визначення змінних
  reg: Tregistry;
begin
  try
  reg:= Tregistry.Create;
  reg.Rootkey:= hkey_current_user;
  reg.Openkey('\SOFTWARE\Microsoft\Windows\Currentversion\Run', True);
  if reg.Valueexists(id) then
    begin
      if reg.Deletevalue(id) then
        Log_PROTECTION('Ключ успішно вилучений.', 0, 2)
      else
        Log_PROTECTION('Не вдається вилучити ключ.', 255, 3);
      end
    else
      Log_PROTECTION('Ключ не існує.', 1, 3)
    finally
      reg.Free;
    end;
end;

procedure TProtection.Log_PROTECTION(s: string; ok, lev: byte);
procedure Addline(l: string);
begin
  PROTECTION.logwindow.Lines.Append(l);
  if updex.Checked then
    PROTECTION.logwindow.Update;
end;

procedure markdate;
begin
  if marked then
    exit;
writeln(logfile, '--- Новий сеанс ---');
writeln(logfile, Formatdatetime('Початок звіту: hh:nn:ss, DDDDD', now));
  marked:= True;
end;

begin
  if (ok = 0) and (not fnoerr.Checked) and (lev >= loglevel.Itemindex) then
    begin
      PROTECTION.logwindow.Selattributes.Color:= clblack;

```

```

    PROTECTION.logwindow.Selattributes.Style:= [fsbold];
    Addline(s);
end
else
if (ok = 255) and (not ferr.Checked) and (lev >= loglevel.Itemindex) then
begin
    PROTECTION.logwindow.Selattributes.Color:= clred;
    PROTECTION.logwindow.Selattributes.Style:= [fsbold];
    Addline(s);
end
else
if (ok = 1) and (not fevent.Checked) and (lev >= loglevel.Itemindex) then
begin
    PROTECTION.logwindow.Selattributes.Color:= clgreen;
    PROTECTION.logwindow.Selattributes.Style:= [fsbold];
    Addline(s);
end
else
if (ok = 2) and (not fnoerrlocb.Checked) and (lev >= loglevel.Itemindex) then
begin
    PROTECTION.logwindow.Selattributes.Color:= clblue;
    PROTECTION.logwindow.Selattributes.Style:= [fsbold];
    Addline(s);
end
else
if (ok = 3) and (not fnoerrmult.Checked) and (lev >= loglevel.Itemindex) then
begin
    PROTECTION.logwindow.Selattributes.Color:= clpurple;
    PROTECTION.logwindow.Selattributes.Style:= [fsbold];
    Addline(s);
end;

if (uselog) and (lev >= filelevel.Itemindex) then
begin
    case ok of
        0:
            if not flnoerr.Checked then
                begin
                    markdate;
                    writeln(logfile, s);
                end;
        1:
            if not flevent.Checked then
                begin
                    markdate;
                    writeln(logfile, s);
                end;

        2:
            if not flnoerrlocb.Checked then
                begin
                    markdate;
                    writeln(logfile, s);
                end;

        3:
            if not flnoerrmult.Checked then
                begin
                    markdate;
                    writeln(logfile, s);
                end;

        255:
            if not flerr.Checked then
                begin
                    markdate;
                    writeln(logfile, s);
                end;
    end;
end;

```



```

        begin
            application.Processmessages;
            Close;
        end;
    end;

end;

end
else
begin
Log_PROTECTION('Вкажіть шлях для відновлення!', 255, 3);
    Showmessage('Вкажіть шлях для відновлення!');
Log_PROTECTION('Відновлення пройшло з помилками!', 255, 3);
    if errd.Checked then
        begin
Showmessage('Відновлення пройшло з помилками! Подивитися звіт. ');
            if agshow.Itemindex <> 2 then
                gofromtray;
            end
        else
            if nowintray then
                begin
                    Showmessage('Відновлення пройшло з помилками!');
                    if agshow.Itemindex <> 2 then
                        gofromtray;
                    end;
                pc.Activepage:= Tabsheet2;
                end;
                button3.Enabled:= True;
            end;
        procedure TProtection.Formclose(Sender: TObject; var Action: Tcloseaction);
        Var // Визначення змінних
            ini: Tinifile;
        begin

            if (tray.Itemindex > 1) and (not shutdown) and (not isauto) then
                begin
                    gototray;
                    Action:= canone;
                    exit;
                end;
            syntax;

// Обробка даних з Ini файлу
            ini.Writestring('main', 'avpath', avpath.Text);
            ini.Writeinteger('main', 'traymode33', tray.Itemindex);
            ini.Writebool('main', 'cheatgetvirrecfornetupdate', zapnet.Checked);
            ini.Writeinteger('main', 'loglevel', loglevel.Itemindex);
            ini.Writeinteger('main', 'logfilelevel33', filelevel.Itemindex);
            ini.Writebool('main', 'agentrestoretofirsttab', firstz.Checked);
            ini.Writeinteger('main', 'raisewindowinagentmode', agshow.Itemindex);
            ini.Free;
            if uselog then
                closefile(logfile);
        end;

        procedure TProtection.Formactivate(Sender: TObject);
        Var // Визначення змінних
            ini: Tinifile;
            dr: dword;
            i, dt: integer;
            r: Tregistry;
        begin
            Onactivate:= nil;
            started:= now;
            dr:= getlogicaldrives;
            for i:= 0 to 31 do
                if (dr shr i) and 1 = 1 then

```

```

begin
dt:= getdrivetype(Pchar(char(byte('A') + i) + ':'));
if (dt <> DRIVE_CDROM) and (dt <> DRIVE_REMOVABLE) and (dt <> DRIVE_RAMDISK)
then
    where.Items.Append(char(byte('A') + i));
end;
where.Itemindex:= 0;

mypath:= extractfilepath(application.Exename);
pc.Activepageindex:= 0;
ini:= Tinifile.Create(windowmdir + 'drupdate.ini');
firstz.Checked:= ini.Readbool('main', 'agentrestoretotfirsttab', True);
agshow.Itemindex:= ini.Readinteger('main', 'raisewindowinagentmode', 0);
tray.Itemindex:= ini.Readinteger('main', 'traymode33', 0);
zapnet.Checked:= ini.Readbool('main', 'cheatgetvirrecforinetupdate', False);
zapinet.Checked:= ini.Readbool('main', 'cheatgetvirrecforinetupdate', False);
multii.Checked:= ini.Readbool('main', 'enablemulticopyforinet', False);
multin.Checked:= ini.Readbool('main', 'enablemulticopyforinet', False);
logpath.Text:= ini.Readstring('main', 'logfilepath', windowmdir +
'drupdate.log');
webgo.Itemindex:= ini.Readinteger('main', 'webgo', 0);
socksproxy.Itemindex:= ini.Readinteger('main', 'sockstype', 0);
socksport.Text:= ini.Readstring('main', 'socksport', '1080');
socksneedpass.Checked:= ini.Readbool('main', 'socksneedpassword', False);
socksuser.Text:= ini.Readstring('main', 'socksuser', '');
sockspass.Text:= ini.Readstring('main', 'sockspassword', '');
synctime.Checked:= ini.Readbool('main', 'syncsystem', synctime.Checked);
times.Text:= ini.Readstring('main', 'timeserver', 'clock.via.net');
prot.Itemindex:= ini.Readinteger('main', 'timeprotocol', 0);
times.Items.Commatext:= ini.Readstring('main', 'timeservers',
times.Items.Commatext);
timeerr.Checked:= ini.Readbool('main', 'timecritical', False);
comp.Checked:= ini.Readbool('main', 'safeautoupdate', False);
updex.Checked:= ini.Readbool('main', 'advancedwindowupdate', False);
downa.Checked:= ini.Readbool('main', 'downloadother', False);
other.Items.Commatext:= ini.Readstring('main', 'otherfiles',
other.Items.Commatext);
dir.Items.Commatext:= ini.Readstring('main', 'inetservers33',
dir.Items.Commatext);
ignor.Items.Commatext:= ini.Readstring('main', 'localhttpignorelist',
'drwebase.vdb');
always.Items.Commatext:= ini.Readstring('main', 'alwaysupdate33',
'drwrisky.vdb, drwrisky.txt, drwnasty.vdb, drwnasty.txt, drwtoday.vdb, drwtoday.txt, d
wrtoday.vdb, dwrtoday.txt, dwntoday.vdb, dwntoday.txt');
mdirs.Items.Commatext:= ini.Readstring('main', 'multicopydirs',
mdirs.Items.Commatext);
if mdirs.Items.Count > 0 then
    mdirs.Items.Delete(mdirs.Items.Count - 1);
locbase.Text:= ini.Readstring('main', 'localpath', '');
if locbase.Text = '' then
    locbase.Text:= spider;
avpath.Text:= ini.Readstring('main', 'avpath', locbase.Text);
run.Text:= ini.Readstring('main', 'autoruncmdline', run.Text);
if run.Text = '' then
    if directoryexists(Sysutils.Excludetrailingbackslash(avpath.Text)) then
        run.Text:= Sysutils.Includetrailingbackslash(avpath.Text);
arun.Checked:= ini.Readbool('main', 'autorun', False);
ask.Checked:= ini.Readbool('main', 'autorunask', True);
askex.Checked:= ini.Readbool('main', 'autorunexitask', askex.Checked);
autoexit.Checked:= ini.Readbool('main', 'autoexit', False);
wait.Checked:= ini.Readbool('main', 'autorunwait', False);
netbase.Text:= ini.Readstring('main', 'remotepath', netbase.Text);
msk.Text:= ini.Readstring('main', 'vdbmask', '*.vdb');
servname.Text:= ini.Readstring('main', 'servname', 'drupdate');
dir.Text:= ini.Readstring('main',
proxy.Text:= ini.Readstring('main', 'proxy', '*');
if proxy.Text = '*' then
    detectproxy(False)

```

```

else
  begin
    pport.Text:= ini.Readstring('main', 'proxyport', '80');
  end;
newhttp.Checked:= ini.Readbool('main', 'httpnewprotocol33', False);
hauth.Checked:= ini.Readbool('main', 'httpneedsverifying', False);
hlogin.Text:= ini.Readstring('main', 'httplogin', '');
fnoerr.Checked:= ini.Readbool('main', 'filternoerror', False);
fevent.Checked:= ini.Readbool('main', 'filterevent', False);
ferr.Checked:= ini.Readbool('main', 'filtererror', False);
fnoerrlocb.Checked:= ini.Readbool('main', 'filternoerrorlocb', False);
fnoerrmult.Checked:= ini.Readbool('main', 'filternoerrorrmcop', False);
flnoerr.Checked:= ini.Readbool('main', 'logfilternoerror', False);
flevent.Checked:= ini.Readbool('main', 'logfilterevent', False);
flerr.Checked:= ini.Readbool('main', 'logfiltererror', False);
flnoerrlocb.Checked:= ini.Readbool('main', 'logfilternoerrorlocb', False);
hpass.Text:= ini.Readstring('main', 'httppassword', '');
huser.Text:= ini.Readstring('main', 'httpuseragent', '');
plogin.Text:= ini.Readstring('main', 'proxyuser', '');
ppass.Text:= ini.Readstring('main', 'proxypassword', '');
flnoerrmult.Checked:= ini.Readbool('main', 'logfilternoerrorrmcop', False);
txt.Checked:= ini.Readbool('main', 'inetgettxt', False);
nonet.Checked:= ini.Readbool('main', 'disablenetcheck', False);
pauth.Checked:= ini.Readbool('main', 'proxynneedsverification', False);
if huser.Text = '' then
  begin
    r:= Registry.Create;
    r.Rootkey:= HKEY_CURRENT_USER;
    r.Openkey('\Software\Microsoft\Windows\Currentversion\Internet Settings',
      True);
    huser.Text:= r.Readstring('User Agent');
    r.Closekey;
    r.Free;
  end;
uselog:= ini.Readbool('main', 'log', False);
rewr:= ini.Readbool('main', 'alwaysrewritelog', False);
_rewr.Checked:= rewr;
notdel.Checked:= ini.Readbool('main', 'keepinstalledfilescopies', False);
logonlyerrors:= ini.Readbool('main', 'logonlyerrors', False);
_uselog.Checked:= uselog;
maxlog:= ini.Readinteger('main', 'maxlog', 2097152);
_maxlog.Text:= Inttostr(maxlog);
_maxlog.Enabled:= uselog;
over.Checked:= ini.Readbool('main', 'overrideversion', False);
_ver.Text:= ini.Readstring('main', 'useversion33', getver);
http.Defaultport:= ini.Readinteger('main', 'localhttpport', 16555);
http.Maxconnections:= ini.Readinteger('main', 'maxserverconnections', 0);
loglevel.Itemindex:= ini.Readinteger('main', 'loglevel', 2);
filelevel.Itemindex:= ini.Readinteger('main', 'logfilelevel33', 0);
maxcon.Text:= Inttostr(http.Maxconnections);
_lochttpport.Text:= Inttostr(http.Defaultport);
_uselochttp.Checked:= ini.Readbool('main', 'uselocalhttpserver', False);
_http.Checked:= ini.Readbool('main', 'localhttpdir', False);
err.Checked:= ini.Readbool('main', 'debugautoerrors', True);
errrd.Checked:= ini.Readbool('main', 'debugerrors', True);
if _uselochttp.Checked then
  begin
    _http.Checked:= True;
    _http.Enabled:= False;
  end;
if uselog then
  begin
logfilepath:= logpath.Text;
    if not fileexists(logfilepath) then
      begin
        assignfile(logfile, logfilepath);
        rewrite(logfile);
        uselog:= ioresult = 0;
      end
  end

```

```

else
begin
  assignfile(logfile, logfilepath);

  if (filesizebyname(logfilepath) > maxlog) or (rewr) then
  begin
    rewrite(logfile);
    uselog:= ioreresult = 0;

    end
  else
  begin
    append(logfile);
    uselog:= ioreresult = 0;
  end;
end;
ini.Free;
if (ansilowercase(Paramstr(1)) = 'час') then
begin
  isauto:= True;
  if not comp.Checked then
    application.Minimize;
  updateok:= Syncsystemtime(False);
  if not err.Checked then
  begin
    application.Processmessages;
    Close;
  end
  else
  begin
    if updateok then
    begin
      application.Processmessages;
      Close;
    end
    else
    begin
      application.Restore;
if messagedlg('Помилка синхронізації часу! Однаково вийти із програми?',
  mtconfirmation, [mbok, mbcancel], 0) = idok then
    begin
      application.Processmessages;
      Close;
    end;
  end;

  end;

  end;

end;

if (ansilowercase(Paramstr(1)) = '-auto') or
((ansilowercase(Paramstr(1)) = '/go') and (webgo.Itemindex = 1)) then
begin
  isauto:= True;
  if not comp.Checked then
    application.Minimize;
  updateok:= pupdate(Pchar(netbase.Text), Pchar(loibase.Text),
Pchar(msk.Text));
  if not err.Checked then
  begin
    application.Processmessages;
    Close;
  end
  else
  begin
    if updateok then
    begin
      application.Processmessages;

```

```

        Close;
    end
else
    begin
        application.Restore;
if messagedlg('Відновлення пройшло з помилками! Однаково вийти із програми?',
            mtconfirmation, [mbok, mbcancel], 0) = idok then
        begin
            application.Processmessages;
            Close;
        end;
    end;

end;

end;

end;

if (ansilowercase(Paramstr(1)) = '-autohttp') or
((ansilowercase(Paramstr(1)) = '/go') and (webgo.Itemindex = 2)) then
begin
    isauto:= True;
    if not comp.Checked then
        application.Minimize;
    updateok:= inetupd;
    if not err.Checked then
        begin
            application.Processmessages;
            Close;
        end
    else
        begin
            if updateok then
                begin
                    application.Processmessages;
                    Close;
                end
            else
                begin
                    application.Restore;
if messagedlg('Відновлення пройшло з помилками! Однаково вийти із програми?',
            mtconfirmation, [mbok, mbcancel], 0) = idok then
                begin
                    application.Processmessages;
                    Close;
                end;
            end;
        end;
    end;

end;

if _uselochttp.Checked then
    http.Active:= True;

end;

function TProtection.inetupd: boolean;
Var // Визначення змінних
    str: Tfilestream;
    ini: Tmeminifile;
    sec, upd, new: Tstringlist;
    last, ext: string;
    i, n: integer;
    Count: longint;
begin
    if loibase.Text = '' then
        begin
            Log_PROTECTION('Вкажіть шлях до локальних баз!', 255, 3);
            Result:= False;
            status.Panels[1].Text:= 'Помилка.';
            exit;
        end;
    end;
end;

```

```

end;
if (not directoryexists(Sysutils.Excludetrailingpathdelimiter(locbase.Text)))
then
begin
Log_PROTECTION(Format('Каталог локальних баз "%s" не існує. Перевірте
настроювання.',
[Sysutils.Excludetrailingpathdelimiter(locbase.Text)]), 255, 3);
Result:= False;
status.Panels[1].Text:= 'Помилка.';
exit;
end;
if not Internetconnected then
begin
Log_PROTECTION('Не виявлене підключення до мережі!', 255, 3);
Result:= False;
status.Panels[1].Text:= 'Помилка.';
exit;
end;
Result:= True;
status.Panels[0].Text:= '';
status.Panels[1].Text:= '';
Count:= 0;
sessiontotal:= 0;
hstest.Enabled:= False;
hstop.Enabled:= True;
stop:= False;
logwindow.Lines.Clear;
if PROTECTION.synctime.Checked then
begin
Log_PROTECTION('Синхронізація часу...', 0, 0);
if timeerr.Checked then
begin
if not syncsystemtime(False) then
Result:= False;
end
else
syncsystemtime(False);
end;
id.Proxyparams.Proxyusername:= plogin.Text;
id.Proxyparams.Basicauthentication:= pauth.Checked;
id.Request.Useragent:= huser.Text;
id.Proxyparams.Proxyport:= strtointdef(pport.Text, 80);
id.Proxyparams.Proxyserver:= proxy.Text;
id.Proxyparams.Proxypassword:= ppass.Text;
id.Request.Basicauthentication:= hauth.Checked;
id.Request.Username:= hlogin.Text;
id.Request.Password:= hpass.Text;
id.Request.Cachecontrol:= "no-cache";
if newhttp.Checked then
id.Protocolversion:= pv1_1
else
id.Protocolversion:= pv1_0;

Log_PROTECTION('Адреса відновлення: ' + dir.Text, 0, 1);
if id.Protocolversion = pv1_0 then
Log_PROTECTION('Версія протоколу: HTTP 1.0', 0, 1)
else
Log_PROTECTION('Версія протоколу: HTTP 1.1', 0, 1);
if trim(proxy.Text) <> '' then
Log_PROTECTION(Format('Прокси: %s:%d', [id.Proxyparams.Proxyserver,
id.Proxyparams.Proxyport]),
0, 0);
Log_PROTECTION('User-Agent: ' + id.Request.Useragent, 0, 0);
Log_PROTECTION('Тимчасовий каталог: ' + tempdir, 0, 0);
str:= Tfilestream.Create(tempdir + 'http_update.tmp', fmcreate);
try
Getfile(dir.Text + 'drweb32.lst', @str);
except
on E: Exception do

```

```

begin
  Indyreport;
  PROTECTION.Handleerror(255, 3, id.Request.URL, E);
  htest.Enabled:= True;
  hstop.Enabled:= False;
  str.Free;
  Result:= False;
  status.Panels[1].Text:= 'Помилка.';
  exit;
end;
end;
Log_PROTECTION('Отриманий drweb32.lst', 0, 0);
Inc(sessiontotal, str.Size);
str.Free;
Log_PROTECTION('Відповідь: ' + id.Response.Responsetext, 0, 0);
ini:= Tmeminifile.Create(tempdir + 'http_update.tmp');
upd:= Tstringlist.Create;
new:= Tstringlist.Create;
sec:= Tstringlist.Create;
ini.Readsections(sec);
if sec.Count > 0 then
  begin
    sec.Sort;
    Log_PROTECTION('Не знайдені секції: ', 0, 0);
    for i:= 0 to sec.Count - 1 do
      Log_PROTECTION(format('%d: [%s]', [i, sec[i]]), 0, 0);
      if sec.Count = 1 then
        last:= sec[0]
      else
        last:= Inttostr(findnewver(sec));
      end
    end
  else
    begin
      Log_PROTECTION('Помилка у форматі drweb32.lst!', 255, 3);
      Result:= False;
    end;

    if over.Checked then
      begin
        last:= _ver.Text;
        Log_PROTECTION('Версія встановлена користувачем: ' + last, 0, 1);
      end
    else
      begin
        Log_PROTECTION('Знайдена остання версія: ' + last, 0, 1);
      end;

      ini.Readsectionvalues(last, sec);
      ini.Free;
      if notdel.Checked = False then
        deletefile(Pchar(tempdir + 'http_update.tmp'));

      if sec.Count > 0 then
        begin
          for i:= 0 to sec.Count - 1 do
            if sec[i][1] = '+' then
              begin
                sec[i]:= copy(sec[i], 2, pos(',', sec[i]) - 2);
                ext:= ansilowercase(extractfileext(sec[i]));
                if (ext = '.vdb') or ((ext = '.txt') and (txt.Checked)) then
                  if not fileexists(locbase.Text + sec[i]) or
                     (always.Items.IndexOf(sec[i]) <> -1) then
                    upd.Append(sec[i])
                  else
                    Log_PROTECTION('Уже встановлена:' + sec[i], 1, 1)
                  end
                else
                  Log_PROTECTION('Ігноруємо небазу:' + sec[i], 1, 0);
                end
              end
            end
          end
        end
      end
    end
  end
end;

```

```

        end;
        upd.Sort; //сортування
    end
else
    begin
Log_PROTECTION('Даних за версією ' + last + ' не знайдене!', 255, 3);
        Result:= False;
    end;

    sec.Free;
    Log_PROTECTION('Отриманий список файлів:', 0, 1);
    if downa.Checked then
        begin
            for i:= 0 to other.Items.Count - 1 do
                if upd.IndexOf(httpgetfilename(other.Items[i])) = -1 then
                    upd.Append('+ ' + other.Items[i])
                else
                    Log_PROTECTION('Зігнорований: ' + other.Items[i] + '. Уже в черзі.', 0, 3);
            end;
            for i:= 0 to upd.Count - 1 do
                if upd[i][1] <> '+' then
                    Log_PROTECTION(upd[i], 0, 1)
                else
                    Log_PROTECTION('Зовнішній: ' + cutfirst(upd[i]), 0, 1);
                    if not (directoryexists(tempdir + 'drup_tmp') or Createdir(tempdir +
'drup_tmp')) then
                        begin
                            stop:= True;
                            Log_PROTECTION('Помилка ініціалізації тимчасового каталогу ' + tempdir +
'drup_tmp', 255, 3);
                            end;
                            for i:= 0 to
                                upd.Count - 1 do
                                    begin
                                        if stop then
                                            begin
                                                Log_PROTECTION('Зупинка', 255, 3);
                                                Result:= False;
                                                break;
                                            end;
                                            if upd[i][1] = '+' then
                                                Log_PROTECTION('Одержуємо файл: ' + cutfirst(upd[i]), 0, 1)
                                            else
                                                Log_PROTECTION('Одержуємо файл: ' + upd[i], 0, 1);
                                                status.Panels[0].Text:= Format('Одержуємо %d з %d', [i + 1, upd.Count]);
                                                if upd[i][1] = '+' then
                                                    str:= Tfilestream.Create(tempdir + 'drup_tmp\' +
httpgetfilename(upd[i]), fmcreate)
                                                else
                                                    str:= Tfilestream.Create(tempdir + 'drup_tmp\' + upd[i], fmcreate);
                                                application.Processmessages;
                                                try
                                                    if upd[i][1] = '+' then
                                                        Getfile(cutfirst(upd[i]), @str)
                                                    else
                                                        Getfile(dir.Text + upd[i], @str);
                                                except
                                                    on E: Exception do
                                                        begin
                                                            Indyreport;
                                                            PROTECTION.Handleerror(255, 3, upd[i], E);
                                                            str.Free;
                                                            Result:= False;
                                                            stop:= True;
                                                            break;
                                                        end;
                                                    end;
                                                end;
            end;
        end;
    end;
end;

```

```

    Inc(sessiontotal, str.Size);
    str.Free;
    if upd[i][1] = '+' then
        begin
            new.Append(cutfirst(upd[i]));
            Log_PROTECTION('Отриманий файл: ' + cutfirst(upd[i]), 0, 2);
        end
    else
        begin
            new.Append(upd[i]);
            Log_PROTECTION('Отриманий файл: ' + upd[i], 0, 2);
        end;

    end;

    for n:= 0 to new.Count - 1 do
        if new[n][1] = '+' then
            new[n]:= httpgetfilename(cutfirst(new[n]));

    if stop then
        begin
            Log_PROTECTION(Format('Отримані не всі файли (%d з %d)', [new.Count,
            upd.Count]), 255, 3);
            Result:= False;
        end;

    for i:= 0 to new.Count - 1 do
        begin
            status.Panels[0].Text:= Format('Встановлення %d з %d', [i + 1, new.Count]);
            if copyfileadv(tempdir + 'drup_tmp\' + new[i], locbase.Text + new[i]) then
                begin
                    Log_PROTECTION('Встановлено: ' + new[i], 1, 2);
                    Inc(Count);
                end
            else
                begin
                    Log_PROTECTION('Помилка копіювання: ' + new[i], 255, 3);
                    Result:= False;
                end;

            if notdel.Checked = False then
                if deletefile(Pchar(tempdir + 'drup_tmp\' + new[i])) then
                    Log_PROTECTION('Вилучений тимчасовий файл: ' + new[i], 0, 0)
                else
                    begin
                        Log_PROTECTION('Помилка видалення: ' + new[i], 255, 3);
                        Result:= False;
                    end;

            end;

            status.Panels[1].Text:= Format('Результат %d з %d', [Count, new.Count]);
            if upd.Count = Count then
                Log_PROTECTION('Нових баз усього: ' + Inttostr(upd.Count) +
                ', усі бази успішно встановлені.', 0, 2)
            else
                begin
                    Log_PROTECTION(Format('Нових баз на сервері: %d Отримане із сервера: %d
                    Установлене: %d',
                    [upd.Count, new.Count, Count]), 255, 3);
                    Result:= False;
                end;
            upd.Free;
            new.Free;
            if (_http.Checked) and (Result) then
                begin
                    Log_PROTECTION('Створюємо каталог для локального http відновлення.', 0, 1);
                    makehttpdir(last);
                end;

```

```

if (multii.Checked) and (Result) then
  if not multicopy then
    Result:= False;
if Result = True then
  begin
    Getupdateflag;
    if zapinet.Checked then
      Getvirrec;
    end;
  Log_PROTECTION(format('Отримане даних: %d байт (%f Кб)', [sessiontotal,
    sessiontotal / 1024]), 0, 2);
  htest.Enabled:= True;
  hstop.Enabled:= False;
end;

procedure TProtection.stopinetupdate(Sender: TObject);
begin
  if id.Connected then
    try
      id.Disconnect;
    except
      end;
  stop:= True;
  Log_PROTECTION('Перериваємо процес...', 255, 3);
end;

procedure TProtection.chkpathclick(Sender: TObject);
Var // Визначення змінних
  i, st: integer;
  s: string;
  nl: TStringlist;
begin
  st:= mdirs.Items.Count;
  if st > 0 then
    begin
      nl:= TStringlist.Create;
      for i:= 0 to mdirs.Items.Count - 1 do
        begin
          s:= Sysutils.Excludetrailingbackslash(mdirs.Items[i]);
          if Canwriteto(s) and (not samefilename (Sysutils.Excludetrailing(
            locbase.Text), Sysutils.Excludetrailingbackslash(s))) then
            nl.Append(Sysutils.Includetrailingbackslash(mdirs.Items[i]))
          else
            begin
              if messagedlg('Шлях ' + mdirs.Items[i] + ' не пройшов перевірку. Вилучити його зі
                списку?', mtconfirmation,
                [mbok, mbcancel], 0) <> idok then
                begin
                  nl.Append(Sysutils.Includetrailingbackslash(mdirs.Items[i]));
                end;
            end;
          end;
        end;
      mdirs.Items.Clear;
      mdirs.Items.Addstrings(nl);
      nl.Free;
    end;
  if st = 0 then
    begin
      Log_PROTECTION(Format('Немає шляхів для перевірки. Список порожній.', []), 0,
        2);
      Showmessage('Немає шляхів для перевірки. ');
    end
  else
    if st = mdirs.Items.Count then
      begin
        Log_PROTECTION(Format('Усі %d шляхи в списку успішно пройшли перевірку й
          подані', [st]), 0, 2);
        Showmessage('Перевірка завершена без помилок. ');
      end
    end
  end;

```

```

else
begin
Log_PROTECTION(Format(' З %d шляхів у списку успішно пройшли перевірку й додані
%d', [st, mdirs.Items.Count]), 0, 2);
Showmessage(Format(' З %d шляхів у списку успішно пройшли перевірку й додані
%d', [st, mdirs.Items.Count]));
end;
end;

function TProtection.Getvirrec: integer;
Var // Визначення змінних
sl: Tstringlist;
i: integer;
begin
Result:= -1;
sl:= Tstringlist.Create;
if Fileexists(Sysutils.extractfilepath(application.Exename) + 'tmpweb.$$$')
then
begin
sl.Loadfromfile(Sysutils.extractfilepath(application.Exename) + 'tmpweb.$$$');
for i:= 0 to sl.Count - 1 do
if pos('Total virus records:', sl[i]) > 0 then
begin
Result:= strtointdef(trim(copy(sl[i], 21, 255)), -1);
break;
end;
end;
Sysutils.deletefile(Sysutils.extractfilepath(application.Exename) +
'tmpweb.$$$');
sl.Free;
end
else
Result:= -1;
if Result = -1 then
Log_PROTECTION('Інформація про бази не доступна.', 255, 3)
Else
Log_PROTECTION(Format(' записів у базах рівно %d', [Result]), 0, 2);
end;

procedure TProtection.getreclick(Sender: Tobject);
Var // Визначення змінних
i: integer;
begin
i:= Getvirrec;
if i = -1 then
begin
Showmessage('Не можливо визначити кількість записів у базах.');
```

```

end;

procedure TProtection.Icontray(var Msg: Tmessage);
Var // Визначення змінних
  Cursorpos: Tpoint;
begin
  if Msg.lparam = WM_RBUTTONDOWN then
  begin
    Getcursorpos(Cursorpos);
    Traymenu.Popup(Cursorpos.x, Cursorpos.y);
  end
  else
  if Msg.lparam = WM_LBUTTONDOWN then
  begin
    if not Visible then
    begin
      gofromtray;
    end;
  end
  else
    inherited;
  end;

procedure TProtection.N8Click(Sender: Tobject);
begin
  gofromtray;
end;

procedure TProtection.N10Click(Sender: Tobject);
begin
  gofromtray;
  shutdown:= True;
  Close;
end;

procedure TProtection.Label32Mouseenter(Sender: Tobject);
begin
  (Sender as TLabel).Font.Color:= clred;
end;

procedure TProtection.Label32Mouseleave(Sender: Tobject);
begin
  (Sender as TLabel).Font.Color:= clblack;
end;

function Windir: string;
Var // Визначення змінних
  Pres: string;
  L: integer;
begin
  Pres:= Stringofchar(' ', 256);
  l:= Getwindirectory(Pchar(Pres), 255);
  Setlength(Pres, l);
  Result:= Pres;
  if Result <> '' then
    if Result[length(Result)] <> '\' then
      Result:= Result + '\';
end;

function Temp: string;
Var // Визначення змінних
  Buffer: array[0..MAX_PATH + 1] of char;
begin
  GettempPath(MAX_PATH + 1, Buffer);
  Result:= Sysutils.Includetrailingbackslash(Buffer);
end;

procedure TProtection.addtsclick(Sender: Tobject);
Var // Визначення змінних

```

```

    s: string;
begin
    s:= trim(inputbox('Сервер часу', 'Адреса', ''));
    if s <> '' then
        times.Items.Append(s);

end;

procedure TProtection.deltsclick(Sender: TObject);
begin
    if times.Itemindex <> -1 then
        times.Items.Delete(times.Itemindex);

end;

procedure TProtection.synctimeclick(Sender: TObject);
begin
    times.Enabled:= synctime.Checked;
    timeerr.Enabled:= synctime.Checked;
    addts.Enabled:= synctime.Checked;
    delts.Enabled:= synctime.Checked;
    prot.Enabled:= synctime.Checked;
    testts.Enabled:= synctime.Checked;
    syncts.Enabled:= synctime.Checked;
end;

procedure TProtection.testtsclick(Sender: TObject);
begin
    if Syncsystem(True) then
        Showmessage('Сервер часу працює')
    else
        Showmessage('Помилка доступу до сервера часу!');
end;

procedure TProtection.synctsclick(Sender: TObject);
begin
    if Syncsystem(False) then
        Showmessage('Час успішний синхронізоване')
    else
        Showmessage('Помилка доступу до сервера часу!');

end;

procedure TProtection.socksneedpassclick(Sender: TObject);
begin
    socksuser.Enabled:= socksneedpass.Checked;
    sockspass.Enabled:= socksneedpass.Checked;
    socksuser.Color:= colors[socksneedpass.Checked];
    sockspass.Color:= colors[socksneedpass.Checked];

end;

procedure TProtection.integclick(Sender: TObject);
Var // Визначення змінних
    s: string;
begin
    s:= Sysutils.Includetrailingbackslash(lochase.Text);
    if Fileexists(changefileext(s, '.drupdate')) then
        begin
            exit;
        end;
    if Fileexists(s) then
        begin
            copyfile(Pchar(s), Pchar(changefileext(s, '.drupdate')), False);
            copyfile(Pchar(application.exename), Pchar(s), False);
        end
    else
        begin
            Log_PROTECTION('Неможливо замінити стандартний файл.

```

```

        Файл не знайдений!', 255, 3);
    Showmessage('Неможливо замінити файл. Файл не знайдений!');
end;
end;

procedure TProtection.deintegclick(Sender: TObject);
Var // Визначення змінних
    s: string;
begin
    s:= Sysutils.Includetrailingbackslash(loctbase.Text);
    if Fileexists(s) then
        begin
            copyfile(Pchar(s), Pchar(changefileext(s, '.exe')), False);
            deletefile(Pchar(s));
        end
    Log_PROTECTION('Восстановленно стандартне відновлення!', 0, 3);
    Showmessage('Відновлення відновлене. ');
    end
    else
        begin
            Log_PROTECTION('Неможливо відновити. Файл не знайдений!', 255, 3);
            Showmessage('Неможливо відновити файл. Файл не знайдений!');
        end;
    end;

procedure TProtection.crhttpclick(Sender: TObject);
begin
    makehttpdir(_ver.Text);
end;

procedure TProtection.testservclick(Sender: TObject);
begin
    shellapi.Shellexecute(handle, 'open', Pchar('http://localhost:'
        +_lochttpport.Text), '', '', sw_show);
end;

procedure TProtection.Label33Click(Sender: TObject);
begin
    shellapi.Shellexecute(handle, 'open',
        '', '', sw_show);
end;

procedure TProtection.dshomeclick(Sender: TObject);
begin
    shellapi.Shellexecute(handle, 'open', '', '', sw_show);
end;

procedure TProtection.helpmousedown(Sender: TObject; Button: Tmousebutton;
    Shift: Tshiftstate; X, Y: integer);
begin
    hidecaret(help.Handle);
end;

procedure TProtection.Formpaint(Sender: TObject);
begin
    if ((tray.Itemindex > 3) or trayst) and (not shutdown) then
        begin
            gototray;
        end;
    onpaint:= nil;
end;

procedure TProtection.Controlwindow(var Msg: Tmessage);
begin
    if (Msg.Wparam = SC_MINIMIZE) and (tray.Itemindex in [1, 3, 5]) and
        (not shutdown) and (not isauto) then

```

```
begin
  gototray;
end
else
  inherited;
end;

initialization // ініціалізація модуля
nt:= not iswin9xsafe;
windowsdir:= windir;
Tempdir:= Temp;
end. // Початок файлу форми
```

КБПЗ_2024