Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідуюча кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«14» червня 2022 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

дипломної роботи
***бакалавра*.**
(назва освітнього ступеня)

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітня програма Кібербезпека
(назва освітньої програми)

на тему: «Засоби захисту інформації на мобільних пристроях»

Виконавець: студентка IV курсу, групи КБ-41

_____ **Діана КОНОНЕЦЬ**
(підпис) (ім'я прізвище)

	Прізвище, ініціали	**Підпис**
Керівник	Іван ПАРХОМЕНКО	

Нормоконтроль	Сергій ДАКОВ	

Київ 2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідуюча кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«01» листопада 2021 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої програми)

Студенту **КБ-41** **КОНОНЕЦЬ Діана Ігорівна**
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи Засоби захисту інформації на мобільних пристроях

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ
Тема дипломної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №5 від 29.10.2021 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ
Архітектури, компоненти мобільних додатків, шифрування даних та алгоритми їх захисту.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ
Архітекрута ОС мобільних пристроїв, основні компоненти додатків, нормативно-правова база, вразливості та загрози доступу даних, механізми безпеки, захист в ОС Android/iOS, рекомендації розробникам додатків.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ
Практична цінність Поєднання засобів захисту мобільних пристроїв та формування рекомендацій щодо їх впровадження.

5. ДАТА ВИДАЧІ ЗАВДАННЯ
Дата видачі завдання: 29.10.2021 року

Завдання видав _____ (підпис) Іван ПАРХОМЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Діана КОНОНЕЦЬ (ініціали, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.10.2021 – 27.01.2022	*виконано*
2	Аналіз літератури	28.01.2022 – 11.02.2022	*виконано*
3	Дослідження ОС мобільних пристроїв та архітектури додатків	12.02.2022 – 24.02.2022	*виконано*
4	Дослідження нормативно-правової бази	25.02.2022 – 24.03.2022	*виконано*
5	Дослідження проблеми розгортання атак, вразливостей та загроз даних	25.03.2022 – 07.04.2022	*виконано*
6	Дослідження політики безпеки та механізмів захисту додатків	08.04.2022 – 20.04.2022	*виконано*
7	Аналіз архітектури антивірусних ПЗ	21.04.2022 – 05.05.2022	*виконано*
8	Проведення дослідження антивірусного ПЗ, формування реуомендацій	05.05.2022 – 01.06.2022	*виконано*
9	Оформлення пояснювальної записки	02.06.2022 – 06.06.2022	*виконано*
10	Підготовка до захисту	07.06.2022 – 10.06.2022	*виконано*

Завдання видав _____ (підпис) Іван ПАРХОМЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Діана КОНОНЕЦЬ (ініціали, прізвище)

Термін подання дипломної роботи до ЕК 06 червня 2022 року

РЕФЕРАТ

Пояснювальна записка: 56 с., 25 рис., 1 табл., 30 джерел.

Об'єктом дослідження є процес виявлення та протидії загрозам, що спрямовані на мобільні пристрої.

Метою роботи є дослідження засобів захисту інформації на мобільних пристроях.

Для досягнення поставленої мети необхідно вирішити такі *завдання***:**

- дослідити архітектуру операційних систем мобільних пристроїв, компоненти мобільних додатків та нормативно правову базу;
- здійснити опис проблем розгортання атак, особливості вразливостей та загроз для мобільних пристроях;
- дослідити інструменти, що використовуються для захисту мобільних пристроїв, а також встановити рекомендації користувачам для безпечної роботи з мобільними пристроями;
- провести тестування антивірусного ПЗ та визначити найбільш надійні інструменти захисту.

У роботі проаналізована література з захисту мобільних пристроїв, проведено дослідження щодо захисту персональних даних при використані мобільних пристроїв.

Предметом дослідження є набір інструментів, що реалізують методи захисту мобільних пристроїв.

Практичною цінністю отриманих результатів є поєднання теоретичного та програмого застосування засобів та механізмів захисту мобільних пристроїв.

Отримані результати допоможуть забезпечити безпеку персональних даних користувача в мобільних пристроях.

Ключові слова: операційна система, архітектура мобільних додатків, безпека в мобільних додатках, політика безпеки, вразливості даних, механізми безпеки даних, антивірусне програмне забезпечення.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AAB	–	Android App Bundle
API	–	Application Programming Interface
FLE	–	Filesystem-Level Encryption
IPS	–	Intrusion Prevention System
HTTP	–	HyperText Transfer Protocol
HTTPS	–	HyperText Transfer Protocol Secure
IT	–	Information Technology
IP	–	Internet Protocol
IPC	–	Inter-Process Communication
IPS	–	Intrusion Prevention System
mRAT	–	Mobile Remote Access
PBKDF	–	Password-Based Key Derivation Function
SDK	–	Software Development Toolkit
ОС	–	Операційна система

ЗМІСТ

ВСТУП

Актуальність даної роботи обумовлена повсякденним використання мобільних пристроїв для різних потреб, що встановлені користувачем. Особисті мобільні пристрої завжди більш вразливі оскільки містять багато конфіденційної інформації про користувача.

Інформація є основою діяльності людини. Саме цей аспект є найбільш привабливим для діяльності зловмисників. Атаки можуть порушити робочі процеси та завдати шкоди репутації, а також мати відчутні витрати. Тому дослідження засобів захисту інформації, що забезпечить безпеку та конфіденційність критично важливих даних є актуальним.

Метою роботи є дослідження засобів захисту інформації на мобільних пристроях.

Для досягнення поставленої мети необхідно вирішити такі *завдання*:

- дослідити архітектуру операційних систем мобільних пристроїв, компоненти мобільних додатків та нормативно правову базу;
- здійснити опис проблем розгортання атак, особливості вразливостей та загроз для мобільних пристроях;
- дослідити інструменти, що використовуються для захисту мобільних пристроїв, а також встановити рекомендації користувачам для безпечної роботи з мобільними пристроями;
- провести тестування антивірусного ПЗ та визначити найбільш надійні інструменти захисту.

Об'єктом дослідження є процес виявлення та протидії загрозам, що спрямовані на мобільні пристрої.

Предметом дослідження є набір інструментів, що реалізують методи захисту мобільних пристроїв.

Практичною цінністю отриманих результатів є поєднання теоретичного та програмого застосування засобів та механізмів захисту мобільних пристроїв.

РОЗДІЛ 1
ІНСТРУМЕНТИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1 Архітектура ОС Android

Операційна система Android — це операційна система для мобільних пристроїв, розроблена компанією Google (GOOGL) для використання, в основному, для сенсорних пристроїв (мобільних телефонів і планшетів). Інтуїтивне керування пристроями відбувається завдяки його структурі – за допомогою рухів пальців, які відображають звичайні рухи. Google також використовує програмне забезпечення Android у телевізорах, автомобілях та наручних годинниках, кожен з яких оснащений унікальним інтерфейсом користувача.

Операційна система Android була вперше розроблена Android Inc., компанією, що займається програмним забезпеченням, перш ніж Google придбала її в 2005 році. Інвестори та аналітики електронної індустрії поставили під сумнів справжні наміри Google вийти на ринок мобільних пристроїв після цього придбання. Google оголосила про майбутній випуск свого першого комерційно пристрою на базі Android у 2007 році.

Архітектура Android містить різну кількість компонентів для підтримки будь-яких потреб пристрою Android. Програмне забезпечення Android містить ядро Linux з відкритим вихідним кодом, яке містить набір бібліотек C/C++, які доступні через служби фреймворку додатків. [1]

Серед усіх компонентів ядро Linux забезпечує основну функціональність функцій операційної системи для смартфонів, а забезпечення запуску програми Android забезпечує віртуальна машина Dalvik (DVM).

Основні компоненти архітектури Android (рис. 1.1):

- Додатки
- Фреймворк додатків
- Час виконання Android

- Бібліотеки платформи
- Ядро Linux

Рисунок 1.1 – Архітектура системи Android [3].

Фреймворк додатків. Фреймворк додатків — це набір програмних інструментів, який дає змогу розробникам додатків зібрати готовий продукт, який відповідає вимогам його власника. Фреймворк забезпечує основу програми, що доповнена графікою, анімацією, спеціальними функціями та функціональністю. Фреймворки додатків розроблені для спрощення процесу розробки додатків і полегшують керування, змінення та виправлення помилок у майбутньому.

Міжпроцесний зв’язок Binder. Механізм зв’язку між процесами Binder дозволяє програмній платформі перетинати межі процесів і викликати код

системних служб Android. Це дає змогу API високого рівня взаємодіяти із системними службами Android.

Системні служби Android. Системні служби є модульними компонентами, який дозволяє виконувати тривалі операції, не взаємодіючи з користувачем. Функціональні можливості, надані API фреймворку додатків, взаємодіють із системними службами для доступу до базового обладнання. Android включає дві групи сервісів: системні, такі як: диспетчер вікон і диспетчер сповіщень; і медіа – сервіси, пов'язані з відтворенням і записом медіа.

Рівень апаратної абстракції (HAL). Рівень апаратної абстракції – це логічний поділ коду, який служить рівнем абстракції між фізичним обладнанням комп'ютера та його програмним забезпеченням. Основна мета HAL полягає в тому, щоб приховати різні апаратні архітектури від ОС, забезпечуючи єдиний інтерфейс до системної периферії.

Ядро Linux. Розробка драйверів пристроїв схожа на розробку типового драйвера пристрою для Linux. Android використовує версію ядра Linux з кількома спеціальними доповненнями, такими як Low Memory Killer (система керування пам'яттю), блокування пробудження (системна служба PowerManager), драйвер Binder IPC та інші важливі функції. [2, 3]

1.2 Архітектура ОС iOS

iOS – це операційна система розроблена компанією Apple. iOS встановлена на усіх пристроях Apple. Це друга найпоширеніша ОС у світі після Android.

У 2005 році, коли Стів Джобс почав працювати над новим смартфоном, рішенням команди було ухвалено за основу взяти вже існуючи ОС Macintosh та iPod.

iOS повністю відрізняється від своїх конкурентів. Усі програми знаходяться всередині окремої захисної оболонки пристрою. Це створено для уникнення можливих конфлітів між програмами. iOS розроблено таким чином, що якщо у

пристрій випадково потрапляє вірус, він не може зашкодити іншим програмам. Хоча в інших операційних системах такої функції немає. [7]

Архітектура iOS - це багатошарова архітектура. На найвищому рівні iOS працює як посередник між основним обладнанням і програмами. Програми не зв'язуються безпосередньо з апаратним забезпеченням.

Програми взаємодіють з апаратним забезпеченням через набір чітко визначених системних інтерфейсів. Ці інтерфейси дозволяють легко писати програми, які постійно працюють на пристроях з різними апаратними можливостями.

Нижчі рівні надають основні послуги, на які покладаються всі програми, а рівень вищого рівня надає складну графіку та послуги, пов'язані з інтерфейсом.

Apple надає більшість своїх системних інтерфейсів у спеціальних пакетах, які називаються фреймворками. Фреймворк — це каталог, який містить динамічну спільну бібліотеку, яка є файлами .a, пов'язаними ресурсами, такими як файли заголовків, зображення та допоміжні програми, необхідні для підтримки цієї бібліотеки. Кожен шар має набір фреймворків, який розробник використовує для створення додатків (рис. 1.2).

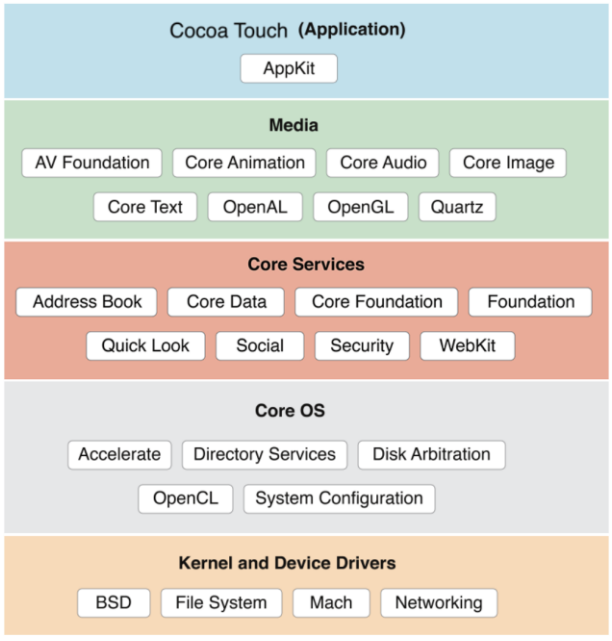

Рисунок 1.2 – Архітектура системи iOS [8].

1. Основний рівень ОС (Core OS):

Рівень Core OS містить функції низького рівня, на яких побудовані більшість інших технологій.

- Основний модуль Bluetooth;
- Accelerate Framework;
- Зовнішня рамка аксесуарів;
- Структура служб безпеки;
- Структура локальної аутентифікації.

2. Рівень основних служб (Core Services).

Деякі з важливих фреймворків, доступних на основних рівнях служб, докладно описані:

- Структура адресної книги – надає програмний доступ до бази даних контактів користувача.
- Cloud Kit Framework – дає середовище для переміщення даних між вашим додатком та iCloud.
- Основна структура Foundation – інтерфейси, які надають основні функції керування даними та сервісні функції для програм iOS.
- Основна структура розташування – надає додаткам інформацію про місцезнаходження та заголовок.
- Healthkit Framework – нова структура для обробки інформації, пов’язаної зі здоров’ям користувача.
- Платформа Homekit – нова платформа для спілкування з підключеними пристроями в домі користувача та керування ними.
- Соціальна платформа – простий інтерфейс для доступу до акаунтів користувачів у соціальних мережах.

3. Media Layer: Технологія графіки, аудіо та відео увімкнена за допомогою Media Layer.

4. Cocoa Touch Layer. Шар Cocoa Touch розташований у верхній частині стеку ОС і містить фреймворки, які найчастіше використовуються розробниками додатків.

Cocoa Touch написаний на Objective-C, заснований на стандартному API Cocoa для Mac OS X і був розширений і змінений для задоволення потреб iPhone. [8]

1.3 Основні компоненти мобільних додатків

Перш ніж говорити про інструменти розробки мобільних додатків, нам потрібно зрозуміти, що таке мобільний додаток насправді.

Мобільний додаток, який зазвичай називають просто додатком, — це тип програмного забезпечення, призначеного для запуску на мобільному пристрої.

Мобільні програми надають користувачам послуги, подібні до тих, які доступні на ПК, наприклад, онлайн-банкінг, доступ до електронної пошти або комуніукація. Додаток – це програмне забезпечення, розроблене для конкретної мети.

Компоненти програми є основними будівельними блоками програми для Android. Кожен компонент є точкою входу, через яку система або користувач можуть увійти у вашу програму.

Програми для Android можна писати за допомогою мов програмування – Kotlin, Java та C++. Інструменти Android SDK компілюють ваш код разом із будь-якими даними та файлами ресурсів у файл .apk або пакет Android App Bundle (AAB) [4].

Пакет Android, який є архівним файлом із суфіксом .apk, містить вміст програми для Android, необхідний під час виконання, і це файл, який пристрої на базі Android використовують для встановлення програми.

Android App Bundle, який є архівним файлом із суфіксом .aab, містить вміст проекту програми Android. AAB – це формат публікації, який не можна встановити на пристрій Android, він відкладає створення файлів .apk на пізній етап. Формат публікації AAB включає скомпільований код і ресурси програми, а також відкладає створення файлів .apk і підписання в Google Play.

Існує чотири типи компонентів програми, а саме:

Діяльність (Activity). Це інтерфейс користувача, який служить точкою входу в додаток. Іншими словами: візуальна інформація, що відображає додаток користувачу, міститься в дії програми. Наприклад, додаток онлайн-банкінгу має дію, що демонструє список транзакцій, наступна дія – створення платежу та іншу дію –перегляд власних коштів з картки. Незважаючи на те, що ці дії можуть функціонувати разом, утворюючи узагальнений досвід користувача в додатку, кожна з них є незалежною. Таким чином, користувач може розпочати будь-яку з цих дій, якщо програма дозволяє це. Наприклад, камера, що вбудована у мобільний пристрій, може розпочати сканування картки чи QR-коду в програмі онлайн-банкінгу, що створить новий платіж, аби дозволити користувачеві надіслати ввласні кошти чи сплатити послугу.

Діяльність сприяє таким ключовим взаємодіям між системою та додатком:

1. Відстеження дії користувача, аби система виконувала процес, на якому розміщена активність.

2. Збереженяя процесів раніше використаних дій користувача.

3. Припинення процесу з відновленням діяльності попереднього.

Служби (Services). Це компонент програми, який працює у фоновому режимі та використовується для виконання довготривалих завдань або завдань для віддаленого процесу. Наприклад, здійснення процесу транзакції не потребує прямої присутності користувача у додатку після згоди на операцію – процес не переривається використанням іншої програми, коли програма знаходиться у фоновому режимі.

Трансляційний приймач (Broadcast receivers). Це компонент програми, який дозволяє програмі відповідати до системних або програмних повідомлень про події. Оскільки широкомовний приймач можна використовувати для відповіді на загальносистемні події, він дозволяє програмі визначати операційні потоки, відмінні від потоків користувачів. Тобто програма може виконувати послідовність операцій без втручання користувача. Наприклад, програму онлайн-банкінгу можна спланувати для запуску служби, яка починає синхронізацію транзакцій щоразу, коли отримує повідомлення про те, що пристрій під’єднано до Інтернету. Аналогічно,

програми також можуть ініціювати трансляції; наприклад, контактна програма може повідомляти іншим програмам через трансляцію, коли додається новий контакт. Оскільки приймачі мовлення — це «ще один чітко визначений запис у програмі, система може доставляти трансляцію навіть у програми, які наразі не запущені».

Постачальник контенту (Content providers). Це компонент програми, який використовується для керування даними, які є приватними для програми або загальними з іншими програмами. Як правило, дані знаходяться в постійному місці зберігання, такому як файлова система, база даних SQL. Кілька програм можуть запитувати або змінювати дані про постачальника вмісту, якщо він це дозволяє і додаток має на це дозвіл [4].

Додатки iOS. Існує три типи програм:

- Нативні програми використовують Objective C/Swift для створення програми.
- Гібридні програми використовують такі фреймворки, як Xamarin, Cordova тощо, а також Objective C/Swift.
- Веб-додатки – це адаптивні версії веб-сайтів, створені для роботи на мобільних пристроях.

Нативний додаток iOS може використовувати код Objective C або Swift разом із будь-якою з їх рідних бібліотек або фреймворків, доступних для використання в додатках iOS. З чого складається файл IPA (рис. 1.3).

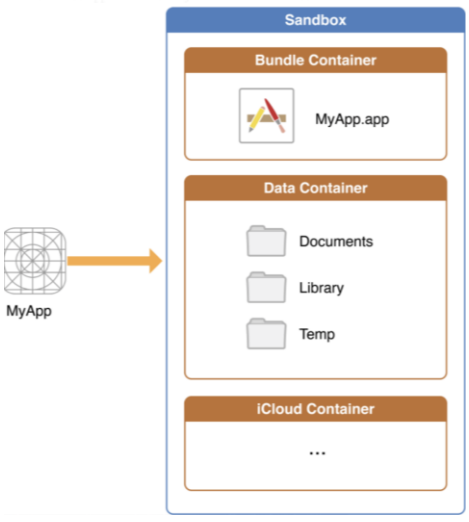

Рисунок 1.3 – Додаток для iOS, що працює у власному каталозі пісочниці [5].

Каталог Bundle або контейнер IPA складається з усіх файлів, які постачаються разом із програмою після встановлення з Apple App Store або будь-якого іншого джерела. Таким чином, файли в цьому каталозі залишаться незмінними протягом усієї певної версії програми.

Каталог «Дані» або контейнер локального сховища даних складається з файлів, які розробник бажає зберегти для програми протягом часу, коли програма встановлена на пристрої.

Файли можуть використовуватися для кешування інформації для швидкого доступу або зберігання інформації в автономному режимі як резервної копії для відновлення програми з того місця, яке було призначено розробником. Таким чином, файли в цьому каталозі, а також інформація у файлах продовжуватимуть змінюватися, поки програма використовується.

Каталог iCloud містить дані, які використовують програми iOS із підтримкою iCloud. Файли в цьому каталозі призначені для зберігання та оновлення джерелами, з яких користувач вирішив оновити файл. Зазвичай він складається з двох частин:

1. Документи. Файли в цьому каталозі призначені для читання та оновлення безпосередньо користувачем. Ці файли регулярно створюються в iCloud для синхронізації.

2. Дані – ці файли не призначені для редагування чи додавання безпосередньо користувачем. За бажанням розробника дані можуть зберігатися в різних каталогах [5].

1.4 Шифрування даних

Шифрування – це процес кодування всіх даних користувача на пристрої Android, що відбувається за допомогою симетричних ключів шифрування. Після того, як пристрій зашифровано, всі дані, створені користувачем, автоматично шифруються перед записом на диск, а всі зчитування автоматично розшифровують дані, перш ніж повернути їх у процес виклику. Шифрування гарантує, що навіть

якщо неавторизована сторона спробує отримати доступ до даних, вони не зможуть їх прочитати.

Починаючи з версії Android 3.0, користувач може шифрувати розділ даних. Шифрування виконується шаром dm-crypt (система шифрування дисків) в ядрі. При активації шифрування пристрою користувачеві необхідно мати пароль або PIN-код. Пароль/PIN користувача та сіль (рядок даних, який передає геш-функції разом з паролем), отримані з /dev/urandom, використовуються як вхідні дані у функції виведення ключа на основі пароля 2 (PBKDF2).

Починаючи з версії Android 7.0, ОС підтримує шифрування на основі файлів – шифрування на рівні файлової системи (FLE). FLE дозволяє шифрувати різні файли різними ключами, які можна розблокувати незалежно один від одного. Доступ до зашифрованих даних можна отримати тільки після успішної аутентифікації.

Версії від Android 5.0 до Android 9 підтримують повне шифрування диска.

Повне шифрування диску – це процес кодування всіх даних користувача на пристрої Android за допомогою зашифрованого ключа. Після того, як пристрій зашифровано, всі дані, створені користувачем, автоматично шифруються перед записом на диск, а всі зчитування автоматично розшифровують дані, перш ніж повернути їх у процес виклику. [5, 6]

Apple надає бібліотеки, які включають реалізації найбільш поширених криптографічних алгоритмів. Посібник із криптографічних послуг Apple містить узагальнену документацію про те, як використовувати стандартні бібліотеки для ініціалізації.

Apple CryptoKit був випущений разом із iOS 13 і побудований на основі рідної криптографічної бібліотеки Apple “corecrypto”. CryptoKit містить безпечні алгоритми хешування, криптографії з симетричним ключем і криптографії з відкритим ключем [8].

Apple CryptoKit містить наступні алгоритми:

- Хеші - MD5 - SHA1 - SHA-2 256-бітовий дайджест (результат перетворення вхідного повідомлення довільної довжини у вихідний бітовий рядок фіксованої довжини) - SHA-2 384-бітний дайджест - SHA-2 512-бітний дайджест.

Хеші є результатом такого алгоритму хешування, як MD5 або SHA (алгоритм безпечного хешування). Ці алгоритми по суті мають на меті створити унікальний рядок фіксованої довжини – хеш-значення – для будь-якої частини даних. Оскільки кожен файл на комп'ютері, в кінцевому рахунку, є лише даними, які можна представити у двійковому форматі, алгоритм хешування може взяти перетворити дані у рядок фіксованої довжини як результат обчислення. Результатом є хеш-значення файлу [9].

- Symmetric-Key – щоб розшифрувати інформацію, потрібно мати той самий ключ, який використовувався для її шифрування. На практиці ключі представляють собою спільний секретний ключ між двома або більше сторонами, який можна використовувати для підтримки приватного інформаційного посилання. Ця вимога, щоб обидві сторони мали доступ до секретного ключа, є одним з основних недоліків шифрування симетричним ключем у порівнянні з шифруванням із відкритим ключем [10].
- Відкритий ключ – це метод шифрування даних двома різними ключами та надання одного з ключів, відкритого, доступного для використання будь-кому. Інший ключ відомий як закритий ключ. Дані, зашифровані відкритим ключем, можна розшифрувати лише за допомогою приватного ключа, а дані, зашифровані за допомогою приватного ключа, можна розшифрувати лише відкритим ключем. Шифрування з відкритим ключем також відоме як асиметричне шифрування [11].

1.5 Нормативно-правова база

В Україні розроблено і впроваджено наступні законодавчі та нормативні документи щодо захисту інформації, технічного захисту інформації, захисту персональних даних, електронного цифрового підпису, технічного захисту інформації:

- Закон України «Про захист інформації в інформаційно-телекомунікаціних системах»;
- Закон України «Про інформацію»;

- Закон України «Про захист персональних даних»;
- Закон України «Про доступ до публічної інформації»;
- Закон України «Про авторське право і суміжні права»;
- «Конвенція про захист осіб у зв’язку з автоматизованою обробкою персональних даних»;
- Директива 97/66/ЄС Європейського Парламенту і Ради «Стосовно обробки персональних даних і захисту права на невтручання в особисте життя в телекомунікаційному секторі»;
- Постанова Пленуму Вищого адміністративного суду України 29 вересня 2016 року № 10 «Про практику застосування адміністративними судами законодавства про доступ до публічної інформації»;
- Закон України « Закон України від 27.03.2014 № 1170-VII «Про внесення змін до деяких законодавчих актів України у зв’язку з прийняттям Закону України «Про інформацію» та Закону України «Про доступ до публічної інформації»»;
- Національний стандарт ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння»;
- Національний стандарт ДСТУ 7624:2014 «Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення»;
- Національний стандарт ДСТУ 7564:2014 «Інформаційні технології. Криптографічний захист інформації. Функція гешування»;
- Національний стандарт ДСТУ ГОСТ 28147:2009 «Системи обробки інформації. Захист криптографічний. Алгоритми криптографічного перетворення»;
- Концепція технічного захисту інформації в Україні. Затверджено постановою Кабінету Міністрів України від 08.10.97 № 1126;
- Постанова Кабінету Міністрів України від 25 вересня 2011 р. № 616 «Про затвердження Положення про Державний реєстр баз персональних даних та порядок його ведення»;

- Положення про технічний захист інформації в Україні. Затверджено Указом Президента України від 27 вересня 1999 р. № 1229;
- Постанова Кабінету Міністрів України від 13 березня 2002 р. № 281 «Про деякі питання захисту інформації, охорона якої забезпечується державою»;
- Положення про державну експертизу в сфері технічного захисту інформації, затверджене наказом Адміністрації Державної служби спеціального зв'язку та захисту інформації України 16.05.07 № 93, зареєстровано в Міністерстві юстиції України 16.07.07 за № 820/14087.

Висновки за розділом 1

Проаналізовано архітектуру операційних систем на базі Android та iOS. Android містить ядро Linux з відкритим вихідним кодом, яке містить набір бібліотек C/C++, у той час коли iOS - це багатошарова архітектура, що на найвищому рівні працює як посередник між основним обладнанням і програмами. Також розглянуто основи архітектури додатків цих ОС.

Apple та Andoid використовують вбудовані методи шифрування даних для забезпечення конфіденційності користувача.

Визначена нормативно-правова база, яка регулює зберігання, передачу та доступ до різних рівнів інформації на законодавчому рівні.

РОЗДІЛ 2
ЗАГРОЗИ ПЕРСОНАЛЬНИХ ДАНИХ

2.1 Проблема розгортання атак

Мобільні додатки знаходяться в епіцентрі сучасних тенденцій розвитку. Більшість із цих програм мають архітектуру клієнт-сервер. Клієнт працює на операційній системі, якою найчастіше є Android або iOS. Цей клієнт завантажується на пристрій із платформ додатків, де розробники публікують свої додатки. З точки зору користувача, клієнтом, встановленим на смартфоні, є мобільний додаток. Це те, з чим користувач взаємодіє. Але насправді є ще один компонент: сервер, який розміщується розробником. Серверним компонентом є веб-додаток, який взаємодіє з клієнтом через Інтернет за допомогою спеціального інтерфейсу API. Тож насправді ми можемо розглядати сервер як більш важливий компонент. Тут зберігається та обробляється інформація. Сервер також відповідає за синхронізацію даних користувача між пристроями.

Сучасні мобільні ОС мають різні механізми безпеки. За замовчуванням інстальована програма може отримати доступ лише до файлів у власних каталогах пісочниці, а права користувача не дозволяють редагувати системні файли. Тим не менш, помилки, допущені розробниками при розробці та написанні коду для мобільних додатків, викликають вразливості [12].

Комплексні перевірки безпеки мобільного додатка включають пошук уразливостей у клієнті та сервері, а також передачу даних між ними. Додатки Android, як правило, містять критичні вразливості дещо частіше, ніж ті, що написані для iOS. Але ця різниця не суттєва, а загальний рівень безпеки клієнтів мобільних додатків для Android і iOS приблизно однаковий. Близько третини всіх уразливостей на стороні клієнта для обох платформ є високоризикованими (рис. 2.1).

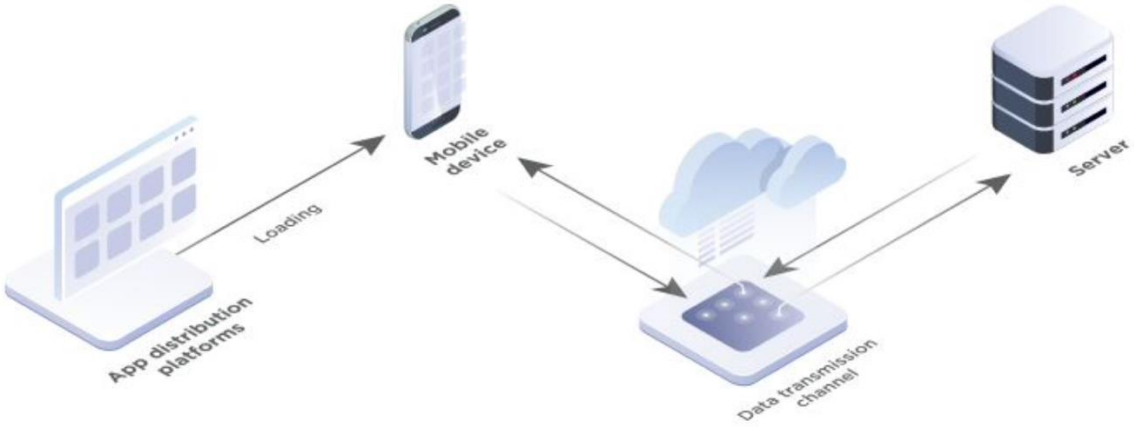

Рисунок 2.1 – Взаємодія клієнт-сервер у мобільному додатку [12].

Багато критики зосереджено на Apple у цьому відношенні, оскільки компанія історично пропонувала надійніший захист для своїх користувачів, ніж фрагментована екосистема Android. Більшість атак на мобільні пристрої пов’язані з хакерами, які намагаються вкрасти конфіденційну інформацію, таку як списки контактів, намагаються надіслати текстові повідомлення або запустити атаку на відмову в обслуговуванні [12].

За винятком сервісів Google Play та кількох інших, вихідний код Android здебільшого є відкритим. Оновлений вихідний код Android знаходиться в Android Open-Source Project (AOSP), тому виробники смартфонів і розробники додатків можуть отримати та змінити його відповідно до потреб своїх користувачів [13].

Це робить платформу прозорою, що дозволяє забезпечувати безпеку та виправлення помилок. Тим не менш, це також може бути шлюзом для зловмисників або погано захищених програм, щоб запровадити шкідливе програмне забезпечення у смартфон Android.

З огляду на гнучкість платформи Android, розробники та виробники додатків несуть відповідальність за підтримку безпеки та виправлення лазівок у програмному забезпеченні. Такий підхід більше фокусується на безпеці додатків, а не на самій безпеці платформи.

2.2 Вразливості даних

Магазин Google Play для Android. Android стала однією з найпопулярніших операційних систем для різних мобільних платформ з провідною часткою ринку. Його популярність значною мірою пояснюється політикою відкритого розповсюдження додатків, яка привернула сторонніх розробників і привела до того, що в Google Play було опубліковано понад мільйон додатків. Google вимагає від розробників Android виконувати одноразову реєстрацію за плату в розмірі 25 доларів США і не вимагає від них розкривати свою справжню особу. Додатки, надіслані в Google Play, публікуються без перевірки коду, що пришвидшує процедуру публікації додатків, однак відсутність перевірки дає змогу зловмисникам поширювати шкідливе програмне забезпечення через офіційні ринки. Майже будь-яка програма завжди потрапляє в Google Play Store, якщо є Manifest файл – містить опис пакету встановлення програм для операційної системи [14].

Хоча Google використовує Play Protect для перегляду програм, перш ніж користувачі завантажують їх, ці програми вже є в Play Store. Тому користувачі, швидше за все, ігнорують попередження безпеки, навіть якщо це є шкідливе програмне забезпечення [15].

Незважаючи на те, що Android має найбільшу базу мобільних користувачів у світі, безпека Android має сумнівну репутацію безпеки, головним чином тому, що нею ніхто не володіє, ніхто не регулює. Насправді, наявність великої кількості користувачів робить ОС більшою метою для кібератак.

Ось деякі з найбільш значущих недоліків безпеки Android:

- Код з відкритим вихідним кодом: зловмисники можуть легко шукати та використовувати вразливі місця.
- Обмежені оновлення: пристрої гарантують лише два роки оновлень.
- Кілька маркетплейсів додатків: кожен ринок перевіряється по-різному, що полегшує публікацію шкідливих програм.

Хоча Android пропонує регулярні оновлення, проблема полягає в тому, що не всі телефони працюють на найновішій версії ОС.

Apple App Store. Apple уважно перевіряє додаток, використовуючи свою програму огляду програм, перш ніж дозволити його розмістити в App Store. Це дозволяє тестувати програми самостійно, використовуючи рішення персоналу (тобто людину) та симулятор варіантів використання для виявлення вразливостей, лазівок у безпеці, дотримання стандартів тощо.

На відміну від Android, розробники iOS повинні адаптувати свої програми до стандартів безпеки платформи iOS. Нерідко Apple відхиляє програму, яка не відповідає її стандартам.

Ця перевірка безпеки робить App Store більш безпечним, ніж Play Store. Натрапити скоріше на зловмисне програмне забезпечення можна у Play Store, ніж у App Store.

iOS розроблена так, щоб бути надійною та безпечною з моменту включення пристрою. Вбудовані функції безпеки захищають від шкідливих програм і вірусів і допомагають захистити доступ до особистої інформації та корпоративних даних. Несанкціоновані модифікації iOS обходять функції безпеки і можуть викликати численні проблеми зі взломом iPhone, iPad або iPod touch, зокрема:

- Уразливості безпеки: джейлбрейк (видалення програмних обмежень, які навмисно встановлені виробником пристрою) пристрою усуває рівні безпеки, призначені для захисту особистої інформації та пристрою iOS. Якщо цей захист видалено з пристрою iOS, хакери можуть викрасти особисту інформацію, пошкодити пристрій, атакувати мережу або запровадити зловмисне, шпигунське програмне забезпечення чи віруси.
- Нестабільність: часті та несподівані збої в роботі пристрою, збої та зависання вбудованих програм і програм сторонніх розробників, а також втрата даних.
- Скорочений термін дії акумулятора: зламане програмне забезпечення спричиняє прискорений розряд батареї, що скорочує роботу iPhone, iPad або iPod touch на одному заряді акумулятора.
- Ненадійний голос і дані: перервані дзвінки, повільне або ненадійне з’єднання даних, а також затримка чи неточні дані про місцезнаходження.

- Неможливість застосувати майбутні оновлення програмного забезпечення: деякі несанкціоновані зміни спричинені пошкодженню iOS – не підлягають ремонту. Це може призвести до того, що зламаний iPhone, iPad або iPod touch стане остаточно непрацездатним після встановлення майбутнього оновлення iOS від Apple.

Apple постійно застерігає від встановлення будь-якого програмного забезпечення, яке зламує iOS. Важливо також зазначити, що несанкціонована зміна iOS є порушенням ліцензійної угоди кінцевого користувача на програмне забезпечення iOS, і через це Apple може відмовити в обслуговуванні для iPhone, iPad або iPod touch, на яких встановлено будь-яке несанкціоноване програмне забезпечення [16].

2.3 Загрози доступу даних

Фішинг. Фішинг є одним із найпоширеніших векторів атак. Більшість кібератак починаються з фішингового листа, який містить шкідливе посилання або вкладення, що містить шкідливе програмне забезпечення. На мобільних пристроях фішингові атаки мають різноманітні засоби доставки посилань і шкідливих програм, зокрема електронну пошту, SMS-повідомлення, платформи соціальних мереж та інші програми [17].

Рутування відноситься до отримання root-доступу до ядра системи. Це найвищий користувач у ієрархії системи, якому дозволено виконувати дії з вищими привілеями. З одного боку, це може надати користувачеві системи додаткову функціональність, але з іншого боку це також може завдати йому чи його компанії багато шкоди.

Рутування досягається шляхом підвищення привілеїв через вразливість безпеки, яка в основному характерна для обладнання та операційної системи. Приклад такої вразливості був знайдений у процесорах Exynos 4210 і 4412, яким оснащені Samsung Galaxy SII, SIII та інші популярні смартфони і планшети. Експлойт, що використовує цю вразливість, може «обійти системні дозволи на рівні

ядра, використовуючи переваги дозволів на читання/запис у ядрі» – отримання користувачеві root-доступу.

З іншого боку, рутування може завдати серйозної шкоди, оскільки воно обходить функції безпеки, запроваджені ОС. Процес рутингу може заблокувати телефон, а це означає, що зміна операційної системи призвела до того, що телефон більше не працюватиме. Крім того, безпека пристрою може бути порушена. Оскільки кожна програма може запитувати права root, кожна програма потенційно має доступ до всіх конфіденційних даних на пристрої [18, 19].

Троянські програми iOS Surveillance та Mobile Remote Access (mRAT). Ці атаки здійснюють джейлбрейк (операція, за допомогою якої можна відкрити повний доступ до файлової системи апарату) на пристрої, який видаляє всі вбудовані механізми безпеки iOS, а також встановлює програмне забезпечення для спостереження та mRAT, яке дає зловмиснику можливість віддалено отримати доступ до всього, що зберігається і проходить через пристрій.

Підроблені сертифікати iOS Enterprise або Developer. Ці атаки використовують сертифікати розповсюдження для «бічного завантаження» програми (зі зловмисним ПЗ), що означає, що її не потрібно проходити через процес перевірки Apple App Store і її можна завантажити прямо на пристрій.

WiFi Людина посередині (MitM). Атака MitM відбувається, коли пристрій підключається до шахрайської точки доступу Wi-Fi. Оскільки всі комунікації проходять через мережевий пристрій, контрольований зловмисником, вони можуть підслуховувати і навіть змінювати зв’язок мережі.

Атаки MitM завжди викликали занепокоєння для бездротових пристроїв, проте поширеність смартфонів в особистому та діловому житті людини зробила мобільні пристрої набагато привабливішими для цієї атаки.

На жаль, типові тривожні та попереджувальні знаки, які люди звикли бачити на комп’ютерах і ноутбуках, набагато більш тонкі в їхніх мобільних аналогах. Наприклад, обмежений екран мобільних пристроїв часто приховує URL-адреси від користувача, тому вони не підтверджують, що URL-адреса, на яку вказує браузер, насправді є призначеною [17].

Уразливості системи нульового дня. Уразливість нульового дня – це вразливість у системі чи пристрої, яка була розкрита, але ще не виправлена. Експлойт, який атакує вразливість нульового дня, називається експлойтом нульового дня.

Оскільки вони були виявлені до того, як фахівцям безпеки та розробникам програмного забезпечення стало відомо про них — і до того, як буде випущене виправлення — уразливості нульового дня становлять більший ризик для користувачів [20].

2.4 Політика безпеки

Сьогодні програми є одними з найважливіших елементів архітектури безпеки. Незважаючи на те, що додатки забезпечують користувачам дивовижні переваги в продуктивності, вони також можуть негативно вплинути на безпеку, стабільність і дані користувача, якщо з ними не працювати належним чином.

При побудові додатків для Android потрібно слідувати загальновідомим правилам, аби забезпечити довіру між клієнтом та програмою.

Забезпечення безпечного спілкування. При захисті даних, якими відбувається обмін між додатком та іншими програмами або між додатком і веб-сайтом, це покращує стабільність додатка та захищаєте дані при надсиланні й отриманні.

Застосування дозволів на основі підпису. Під час обміну даними між двома програмами, використання дозволів на основі підпису перевіряють, що програми, які мають доступ до даних, підписані одним і тим же ключем підпису. Таким чином, ці дозволи пропонують більш спрощену та безпечну роботу користувача.

Заборона доступу до постачальників вмісту додатка. Дані, що не потребують надсиланню до іншого додатка, повинні містити заборону доступу до об’єктів ContentProvider, які містить основна програма.

Запит облікових даних, перш ніж показувати конфіденційну інформацію. Запит паролю (PIN-код/пароль/біометричний пароль) для верифікації доступу до даних.

Безпека мережі. У наступних розділах описано, як можна підвищити безпеку мережі вашої програми [21].

Спираючись на унікальні можливості апаратного забезпечення Apple, політика безпеки системи відповідає за контроль доступу до системних ресурсів на пристроях Apple без шкоди при використанні. Безпека системи охоплює процес завантаження, оновлення програмного забезпечення та захист системних ресурсів комп'ютера, таких як центральний процесор, пам'ять, диск, програмні програми та збережені дані.

Apple надає рівні захисту для гарантування, що додатки не містять відомих шкідливих програм і не були підроблені. Додаткові засоби захисту забезпечують ретельний доступ із програм до даних користувача. Ці засоби керування безпекою забезпечують стабільну безпечну платформу для програм, що дає змогу тисячам розробників надавати сотні тисяч програм для iOS, iPadOS та macOS — і все це без впливу на цілісність системи. Користувачі можуть отримати доступ до цих програм на своїх пристроях Apple, не боячись вірусів, шкідливих програм або несанкціонованих атак.

На iPhone, iPad та iPod touch усі програми отримані з App Store — і всі програми знаходяться в пісочниці – механізм для безпечного виконання програм — щоб забезпечити найточніший контроль.

Як додатковий контроль на різних платформах, пісочниця допомагає захистити дані користувачів від несанкціонованого доступу додатків. А в macOS дані в критичних областях найзахищені, що допомагає користувачам залишатися контролювати доступ до файлів на робочому столі, документах, завантаженнях та інших областях з усіх додатків, незалежно від того, чи програми, які намагаються отримати доступ, самі є ізольованими чи ні.

Додатки мають відповідати всім юридичним вимогам – відповідальність місцевим законам, а не лише політиці компанії.

Захист конфіденційності користувачів має першорядне значення в екосистемі Apple, під час роботи з особистими даними потрібно переконатися, що є дотримання

методів конфіденційності, чинного законодавства та умов Ліцензійної угоди програми Apple для розробників.

1. Збір та зберігання даних

(i) Політика конфіденційності: усі програми повинні містити посилання на свою політику конфіденційності в полі метаданих App Store Connect і в межах програми у легкодоступному вигляді. Політика конфіденційності повинна чітко:

- Визначите, які дані (якщо такі є) збирає програма/сервіс, як збирають ці дані та всі види використання цих даних.
- Підтвердити, що будь-яка третя сторона, з якою програма ділиться даними користувача (відповідно до цих Інструкцій), наприклад інструменти аналітики, рекламні мережі чи пов’язані організації, які матимуть доступ до даних користувача — забезпечує захист даних користувача, як зазначено в політиці конфіденційності програми та вимагається цими Інструкціями.
- Дозвіл. Додатки, які збирають дані користувачів або використання, повинні отримати згоду користувача на збір, навіть якщо такі дані вважаються анонімними на момент збору або відразу після нього.
- Мінімізація даних: програми повинні запитувати доступ лише до даних, що стосуються основної функціональності програми, і повинні збирати та використовувати лише дані, необхідні для виконання відповідного завдання.
- Доступ: додатки повинні поважати налаштування дозволів користувача та не намагатися маніпулювати, обманювати чи змусити людей погодитися на непотрібний доступ до даних.
- Вхід в обліковий запис: якщо додаток не містить значних функцій на основі облікового запису – отже, виеористаннч можливе без входу.
- Розробники, які використовують програми для таємного виявлення паролів або інших приватних даних, будуть видалені з Програми розробників Apple.

2.Використання та обмін даними

- Якщо інше не передбачено законом, ви не маєте права використовувати, передавати чи ділитися чиїмись особистими даними без попереднього дозволу.

- Дані, зібрані для конкретної мети, не можуть бути перероблені без подальшої згоди, якщо інше прямо не передбачено законом.
- Обмежити використання інформації з Контактів, Фотографій або інших API, що мають доступ до даних користувачів, для створення бази даних контактів для власного використання або для продажу/розповсюдження третім сторонам.
- Додатки, які використовують Apple Pay, можуть передавати дані користувача, отримані через Apple Pay, лише третім сторонам для полегшення або покращення доставки товарів і послуг [22, 23].

2.5 Механізм безпеки додатків

Системні розділи Android та iOS недоступні для записів, що запобігає випадковій або навмисній зміні файлів. Крім того, обидві операційні системи використовують принцип «пісочниці». Відповідно до цього принципу кожна програма працює окремо і не може отримати доступ до системних файлів або даних інших програм. У системі iOS майже всі програми працюють під непривілейованим режимом. У системі Android кожна програма має свого користувача, який розмежовує права запуску програм у ядрі операційної системи.

Основні відмінності між механізмами безпеки для Android і iOS зводяться до:

- Обмежений доступ до ядра;
- Перевірка завантаженої ОС;
- Право керування доступом [24].

Перш ніж з’явитися в App Store, додатки iOS перевіряються, тестуються та перевіряються відповідно до вимог. Кожна програма, встановлена на iOS, повинна мати унікальний сертифікат — «iOS Developer Program» — отриманий після процесу верифікації. Ці заходи забезпечують захист від шкідливого програмного забезпечення в App Store [25].

Google не перевіряє програми перед завантаженням їх у Google Play, але регулярно виконує сканування магазину для виявлення шкідливих програм. Цей підхід може бути небезпечним.

Google Play безперечно містить шкідливе програмне забезпечення, але, володіючи деякими базовими навичками користувача, ви можете захистити свій пристрій і ОС. Наприклад, завантажуючи програми на пристрій Android, користувач може побачити повний список дозволів доступу, які потрібні додатку. Якщо щось на кшталт програми-ліхтарика запитує доступ до списку контактів або доступу до Інтернету, це, безперечно, шкідливе програмне забезпечення [26].

В iOS ситуація з правами доступу дещо інша: кожен запит на доступ має бути прийнятий або скасований користувачем.

З точки зору розробника, основним ризиком є втрата клієнта в результаті хакерської атаки. Android і iOS подібні в протистоянні локальним і веб-атакам. Однак, якщо розробники під час розробки дотримуються критеріїв безпеки, вони можуть розробити добре захищений додаток для Android та iOS.

Програми для Android все ще легко декомпілювати та замінювати первинний код шкідливим, тому розробникам слід застосовувати методи обфускації коду. Обфускація коду — це процес модифікації виконуваного файлу для утруднення або неможливості декомпіляції або розбирання програм. Здебільшого це робиться за допомогою автоматизованих інструментів перед створенням програми.

Хоча програми iOS уразливі до переповнення буфера, розробники iOS використовують механізми, які можуть запобігти експлуатації цих уразливостей. Серед цих механізмів використовуються такі параметри компіляції, як PIE (Незалежний від позиції виконуваний файл), SSP (Захист від розбиття стека) і ARC (автоматичний підрахунок посилань). Ці параметри ефективно керують пам’яттю та запобігають помилкам, які можуть призвести до переповнення буфера [24].

Тому програми для Android і iOS досить безпечні, якщо розробники дотримуються вимог безпеки.

Висновки за розділом 2

У даному розділі розглягуті проблеми розгортання атак. Мобільні пристрої стають основними цілями для нападу зловмисників. Завантажені додатки з

неперевірених сервісів можуть стати точкою витоку конфіденційних даних. У екосистемі Android користувачі можуть нехтувати цим правилом. ОС iOS має жорстке обмеження щодо встановлення стороніх додатків, на відміне від Android.

Політика безпеки додатків є важливим елементом у розробці, аби забезпечити довіру між клієнтом та програмою. Захист конфіденційності користувачів має бути на першому місці значимості. Проте, користувач також повинен керувати своїми даними. Кожен додаток при встановленні надсилає запит на читання, модифікацію даних. Перед натисканням на погодження якось дії, потрібно уважно читати повідомлення запиту.

Важливою перевагою офіційних магазинів додатків є проходження сертифікації та тестування на зловмисне ПЗ перед публікацією на сервіси App Store чи Google Play.

РОЗДІЛ 3
ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЇ НА МОБІЛЬНИХ ПРИСТРОЯХ

3.1 Архітектура антивірусного програмного забезпечення

Антивірусне програмне забезпечення – це програма або набір програм, призначених для запобігання, пошуку, виявлення та видалення програмних вірусів та іншого шкідливого програмного забезпечення, такого як хробаки, трояни, рекламне програмне забезпечення тощо [25].

Антивірусне програмне забезпечення виявляє віруси за допомогою двох поширених методів – виявлення на основі сигнатур і виявлення на основі евристики для виявлення шкідливого коду. Виявлення на основі сигнатур є ефективним способом виявлення шкідливого програмного забезпечення. Сигнатура зазвичай базується на частині коду, вилученого з самого вірусу. Антивірусне програмне забезпечення буде сканувати файли, повідомлення електронної пошти та інші дані за допомогою певних протоколів, а потім порівнюватиме ці дані з підписами, що зберігаються в його базі даних. Коли дані збігаються з сигнатурою, антивірусне програмне забезпечення виконає ряд дій (залежно від конфігурації та правил), таких як поміщення файлу в карантин, видалення файлу з вірусом та попередження про те, що сталася подія. Основна проблема з виявленням на основі сигнатур полягає в тому, що антивірус залежить від актуальності бази даних, що містить віруси, і може бути затримка часу відповіді між новими загрозами, які оновлюються у вірусній базі даних. Крім того, евристичний аналіз аналізує збірку шкідливого коду, досліджуючи код і логіку, щоб визначити, чи призначене програмне забезпечення для виконання шкідливих дій. Евристичне виявлення забезпечує переваги в його здатності ідентифікувати невідоме шкідливе програмне забезпечення, вирішуючи проблеми з базами даних виявлення сигнатур, які вимагають регулярних оновлень [26, 27].

Загалом антивірусне програмне забезпечення складається з чотирьох компонентів (рис. 3.1). Менеджер антивіруса керує загальними функціями. Це

користувальницький інтерфейс, який може виконувати сканування зловмисного програмного забезпечення, переглядати результати відновлення зловмисного програмного забезпечення, посилати сигнали тривоги. У корпоративному середовищі антивірусний менеджер є непрямим інтерфейсом для інтегрованого керування. Застосовується політика безпеки, встановлена адміністратором безпеки, і передається інформація про роботу антивірусу, яку запитує адміністратор безпеки. Менеджер антивіруса також надає функцію оновлення БД сигнатур зловмисного програмного забезпечення та антивірусної системи, яка відповідає за перевірку та виправлення шкідливих програм [28].

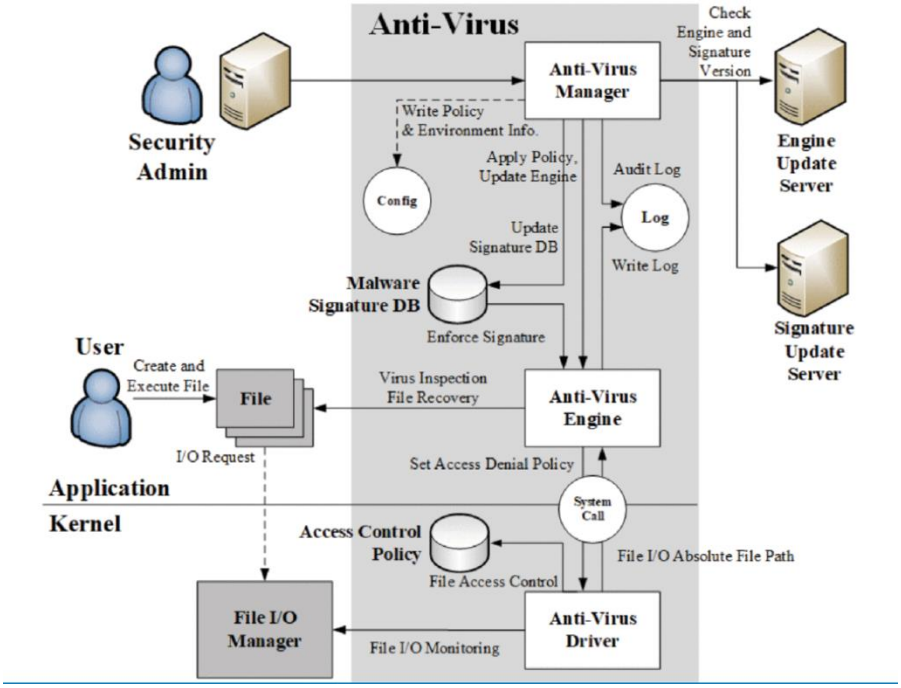

Рисунок 3.1 – Загальна структура антивірусного ПЗ [28].

Антивірусний двигун — це набір технічних механізмів для роботи зі шкідливим програмним забезпеченням в системі. Він витягує інформацію про файл і порівнює її з базою даних сигнатур шкідливих програм, щоб визначити, чи є файл шкідливим. Якщо файл є шкідливим, антивірусна система відновлює його. Якщо

файл не вдається відновити, антивірусна система створює політику відмови в доступі та надсилає її антивірусному драйверу для застосування.

У базі даних Malware Signature реєструються сигнатури зловмисного програмного забезпечення для визначення того, чи містить сканований файл зловмисне програмне забезпечення, і правила відновлення файлів, заражених шкідливим програмним забезпеченням [29, 30].

Антивірусний драйвер — це функція моніторингу входу зловмисного програмного забезпечення в режимі реального часу, яка обмежує доступ до файлів користувача, заражених шкідливим програмним забезпеченням. Він відстежує ввід-вивід файлової системи і отримує абсолютний шлях до файлу для антивірусної системи для перевірки на наявність шкідливих програм. Коли антивірусна система визначає, що файл є зловмисним програмним забезпеченням, але не може відновити файл, антивірусний драйвер отримує політику відмови у доступі, створену та надану антивірусною платформою; він реєструє та застосовує політику заборони доступу користувача [31].

3.2 Параметри тестування

Антивірус - це один із процесів, який може використовувати багато апаратних ресурсів (пам’ять і центральний процесор).

Для тестування було обрано наступні антивірусні програмні забезпечення:

1. Bitdefender
2. McAfee AntiVirus Plus
3. Avira
4. Sophos Intercept X for Mobile
5. ESET Antivirus
6. Dr.Web

Під час тестування фіксуватися будуть наступні значення:

1. швидкість та час сканування;
2. автоматичне сканування;

3. вплив на час автономної роботи;

4. Wi-Fi перевірка;

5. VPN;

6. Web Protection;

7. Управління додатками.

3.3 Процес тестування.

Для проведення дослідження було встановлено емулятор Android, віртуальна машина – Parallels Desktop. Версію образу я обрала – Android 9.0 (рис.3.2).

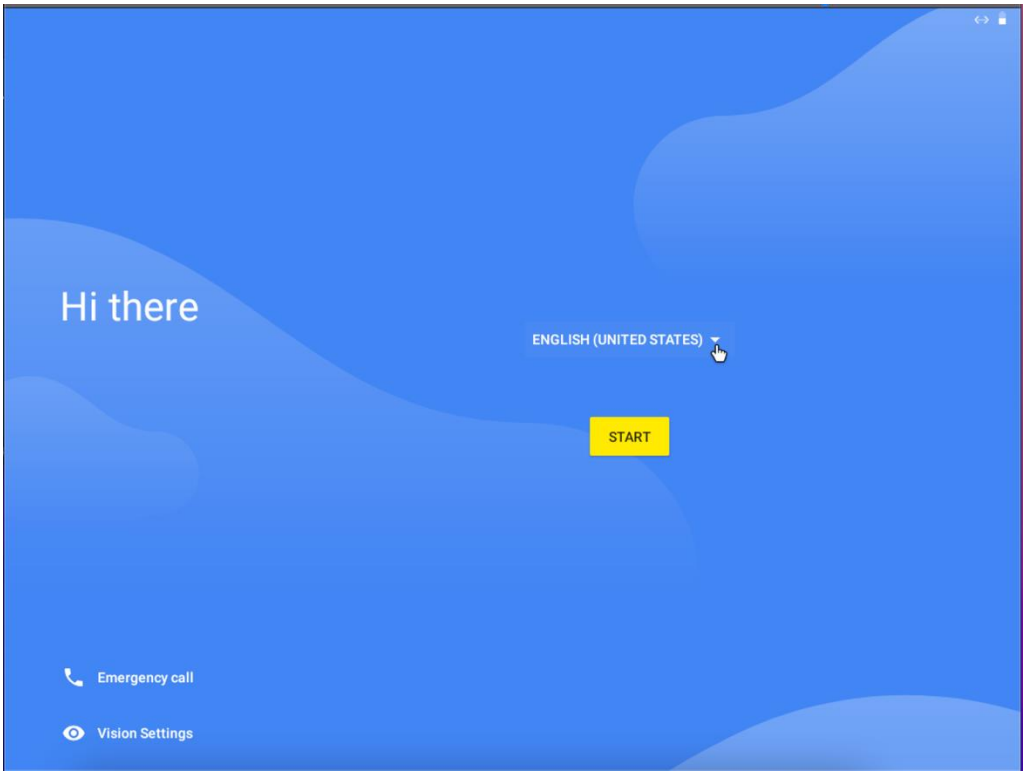

Рисунок 3.2 – Встановлений образ Android 9.0

Першим кроком потрібно створити акаунт у Google, що дасть можливість завантажувати додатки з офійної платформи Google Play. Тому введемо усі необхідні дані, що від нас вимагаються (рис.3.3-3.4).

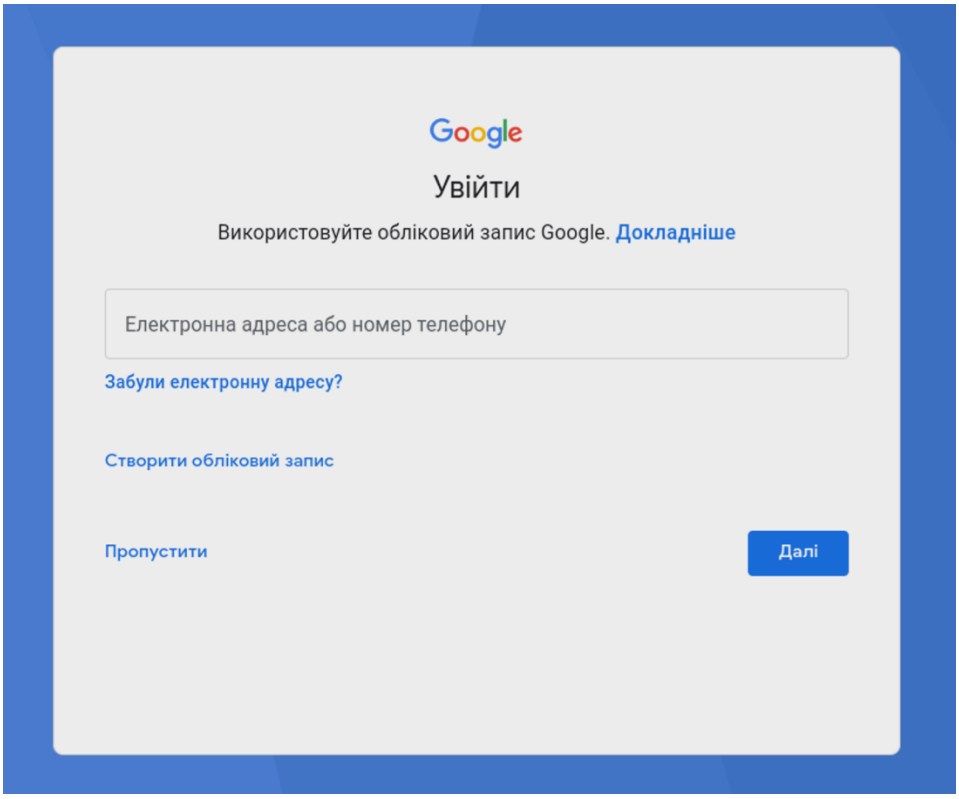

Рисунок 3.3 – Створення аккаунту у Google

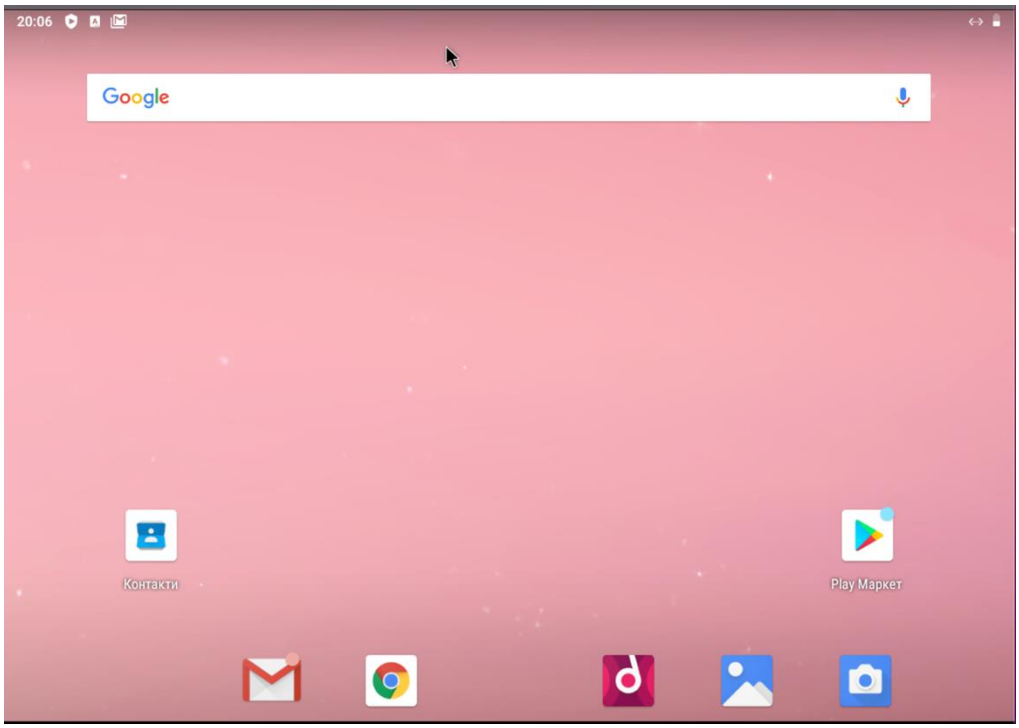

Рисунок 3.4 – Робочий стіл ОС Android

Для тестування на емулятор Android були встановлені різні типи файлів: картинки, ігри, файли із шкідливим вмістом та інші (рис.3.5-3.6).

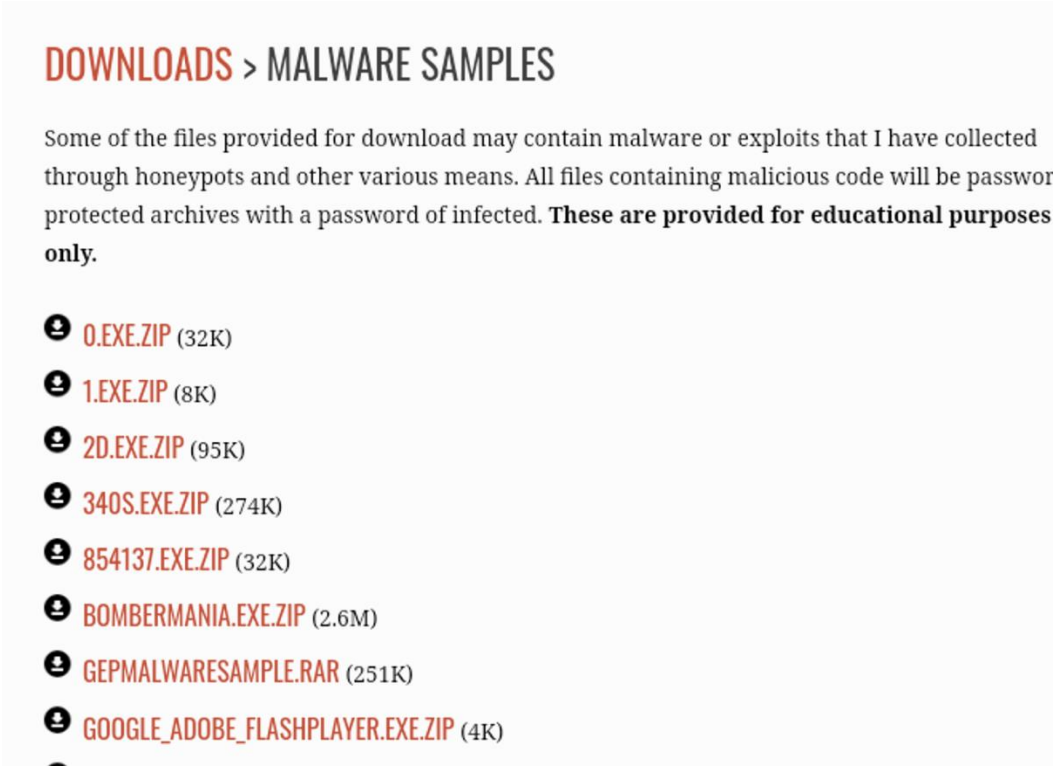

Рисунок 3.5 – Зловмисні файли

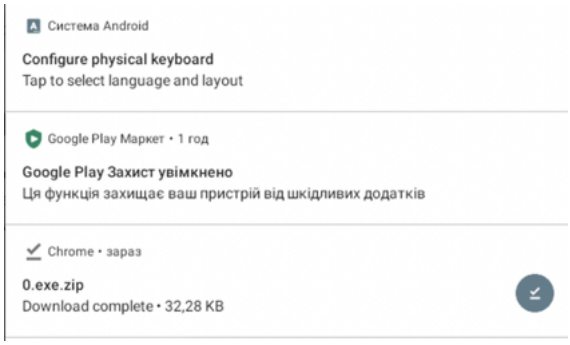

Рисунок 3.6 – Завантаження шкідливого ПЗ

Наступним кроком було встановлення антивірусні ПЗ, за допомогою яких будуть фіксуватися підозрілі дії. Відразу починаємо сканування нашої ОС (рис.3.7).

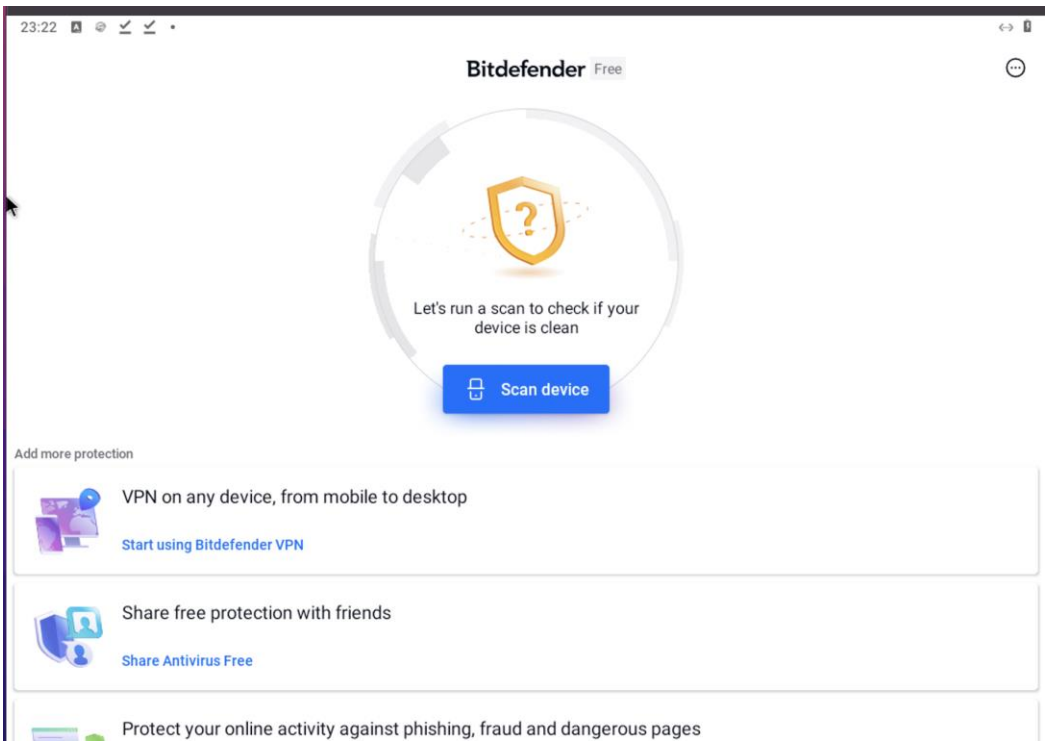

Рисунок 3.7 – Запуск сканування

Bitdefender відразу розпізнав шкідливі файли та пропонує їх усунути (рис.3.8).

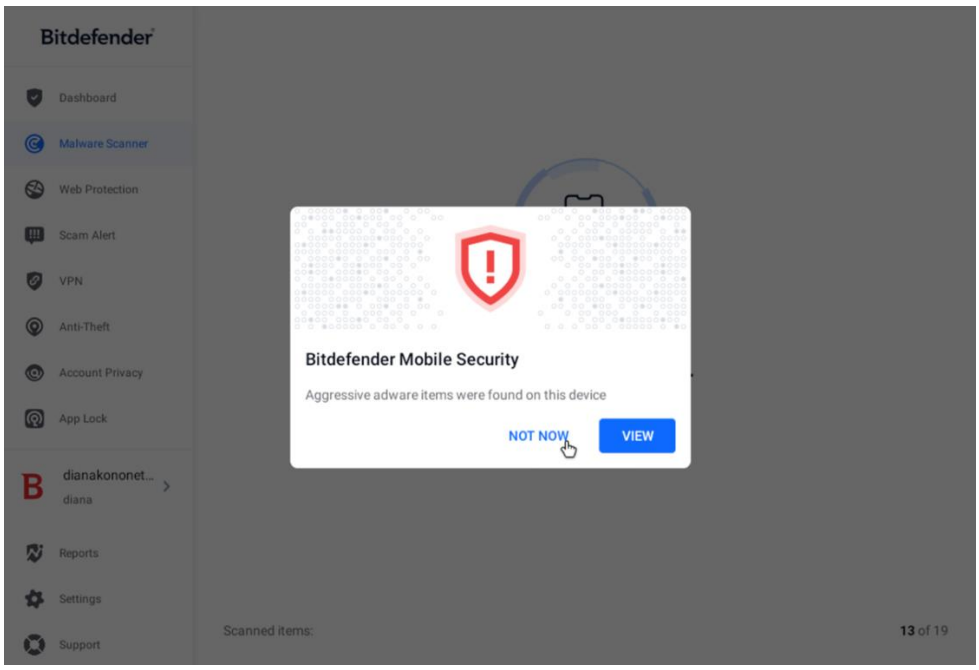

Рисунок 3.8 – Результат сканування

Для порівняння результатів оберемо наступне антивірусне ПЗ: Dr.Web. Дане ПЗ пропонує провести повне та швидке сканування, для більш надійніших результатів ми оберемо повне (рис.3.9).

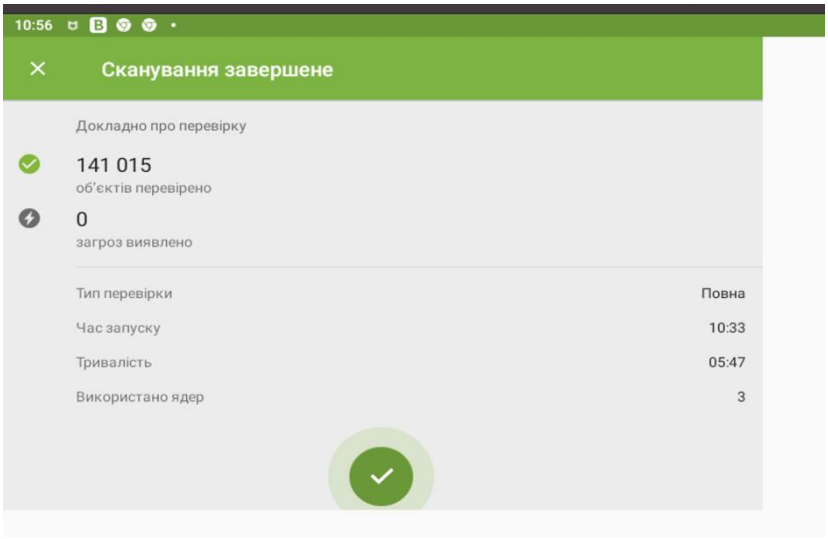

Рисунок 3.9 – Результати сканування

Результати дослідження: Dr.Web провів сканування за фіксований час – 05:47, що на відміну від попередньої програми перевищує тривалість на 5 хвилин. Окрім цього, даний антивірус не виявив жодних загроз, у той час коли Bitdefender – 3.

Після цього було встановлено гру з ненадійного ресурсу. Bitdefender відразу показує результат та попереджає про ненадійне походження файлу (рис.3.10).

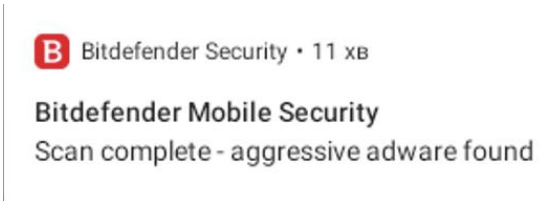

Рисунок 3.10 – Попередження про загрозу

3.4 Результати тестування

1. Було провереденно тестування за швидкість сканування. На графіку нижче зображено порівняння усіх об’єктів сканування, що були зазначені у попередньому пункті (рис.3.11).

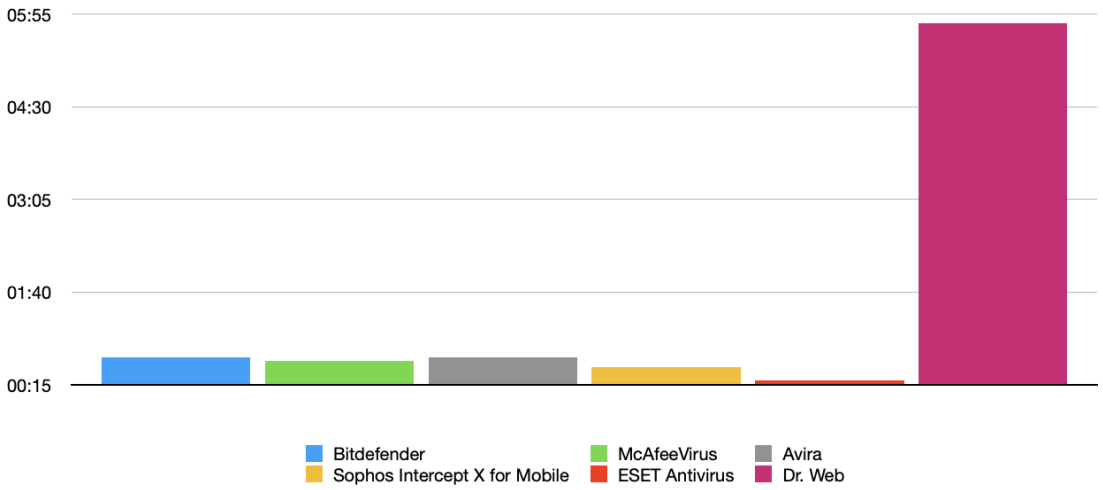

Рисунок 3.11 – Швидкість сканування

2. Параметр № 2 – автоматичне сканування. Фіксувалися швидкість реагувань на загрози та підозрілі дії. Оцінювалося кожне ПЗ від 0 до 5 (рис.3.12).

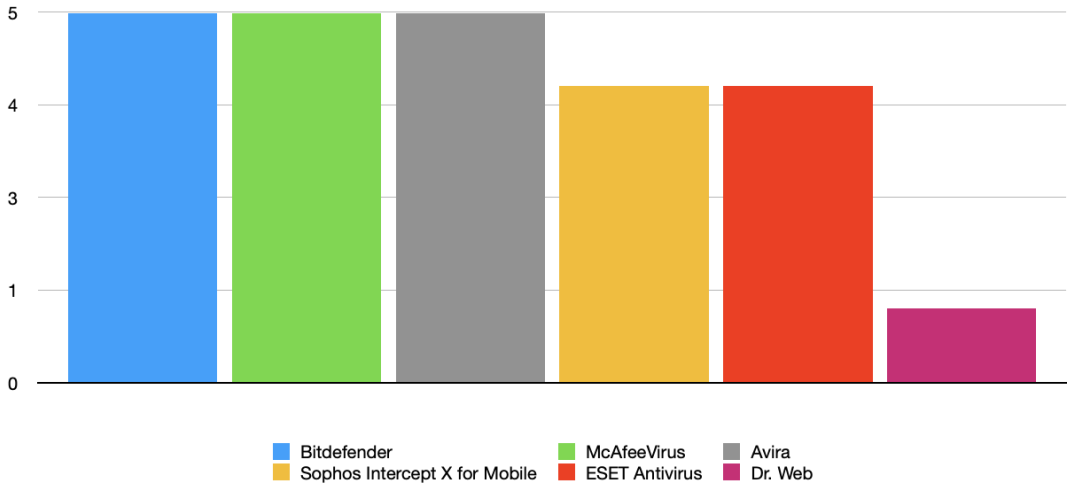

Рисунок 3.12 – Автоматичне сканування

3. Параметр № 3 – вплив на час автономної роботи. Фіксувалися швидкість реагувань на загрози та підозрілі дії. Оцінювалося кожне ПЗ у рівному співвідношенні у відсотках (рис.3.13).

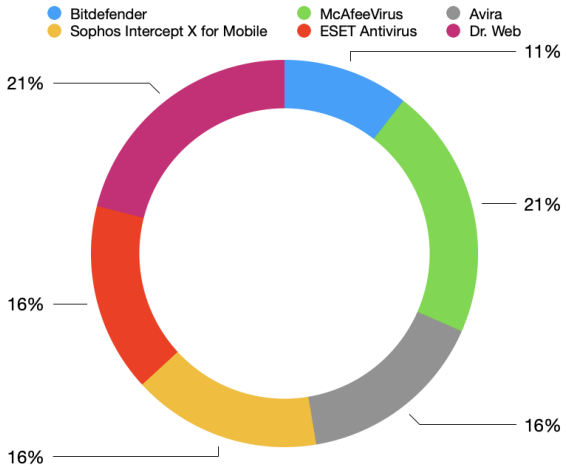

Рисунок 3.13 – Вплив на час автономної роботи

4. Параметр № 4 – Wi-Fi перевірка. Дана функція була у наступних антивірусних додатках: Avira, Bitdefender та McAfeeVirus. Кожен додаток виконав свою задачу, тому результати розподілені рівномірно (рис.3.14).

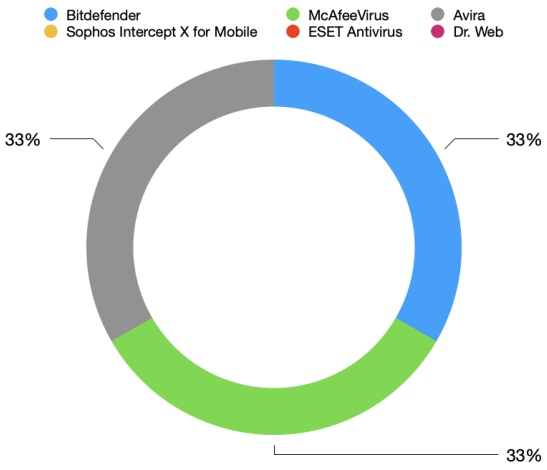

Рисунок 3.14 – Wi-Fi перевірка

5. Параметр № 5 – VPN. Вбудований VPN, що задає додатковий захист даних, трафік є необмеженим та не сповільнює роботу веб-перегляду. Дана опція доступна у: Avira, Bitdefender, ESET Antivirus та McAfeeVirus – результати розподілені рівномірно (рис.3.15).

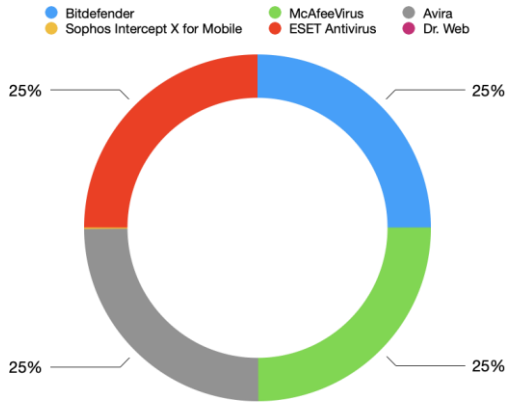

Рисунок 3.15 – VPN

6. Параметр № 6 – Web Protection. Більшість основних браузерів мають плагіни, доповнення, панелі керування, додаткові помічники, які можуть загрожувати безпеці. Контроль цих компонентів може допомогти. Внесення до чорного списку шкідливих доменів та посилань є основною функцією. Коли Web Protection визначає потенційно небезпечний веб-сайт, який користувач має намір відвідати, виводиться відповідне попередження. Усі додатки мали успіх під час дослідження тому мають наступні результати (рис. 3.16).

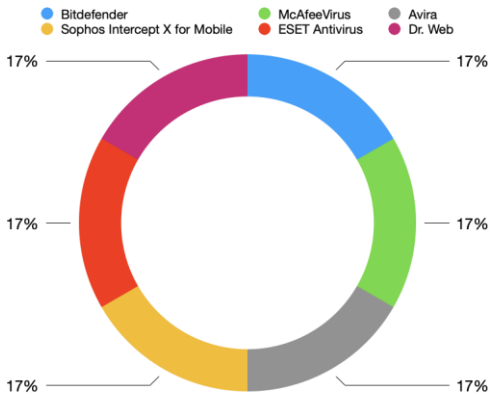

Рисунок 3.16 – Web Protection

7. Параметр № 7 – управління додатками. Управління додатками постійно відстежує процеси, файли, програми та ключі реєстру для запобігання несанкціонованій поведінці. Контроль програм, що запускаються на пристроях та як їм дозволено виконуватись. Дана функція працює на усіх ПЗ, окрім Dr.Web (рис.3.17).

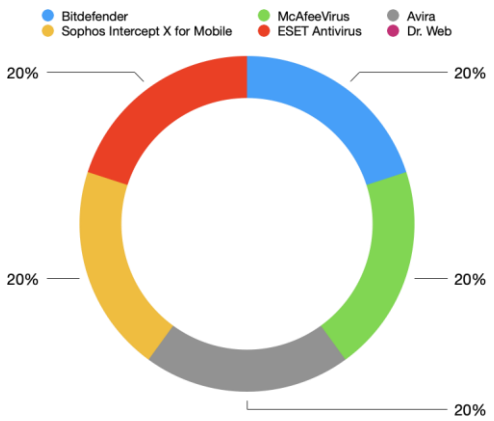

Рисунок 3.17 – Управління додатками

8. У таблиці зазначені основні характеристики антивірусного ПЗ (табл. 3.1).

Таблиця 3.1

Характеристика антивірусного ПЗ

Параметр	**Bitdefender**	**McAfeeVirus**	**Avira**	**Sophos Intercept X for Mobile**	**ESET Antivirus**
Швидкість сканування	00:40	00:37	00:40	00:31	00:19
Автоматична сканування	так	так	так	так	так
Вплив на час автономної роботи	ні	ні	ні	ні	ні
Wi-Fi перевірка	так	так	ні	так	так
VPN	так	так	так	ні	ні
Web Protection	так	так	так	так	так
Управління додатками	так	так	так	так	так

Розроблені рекомендації користувачам:

1. Регулярне оновлення ОС та додатків. Ігнорування сповіщень може викликати загрозу даних, у той час коли регулярні оновлення гарантують останні конфігурації безпеки. Щоб переконатися, що всі додатки оновлені, користувачі iOS можуть зайти в App Store, щоб перевірити наявність доступних оновлень. Користувачі Android можуть зробити те саме, зайшовши в Play Store.

2. Уникання загальнодоступних точок Wi-Fi. Щоб залишатися в безпеці під час використання загальнодоступного Wi-Fi, обов'язково підключіться за допомогою VPN. Зміна вашої віртуальної мережі захистить місцезнаходження та інформацію від зловмисників.

3. Резервне копіювання даних. Це гарантує доступ до даних у разі втрати пристрою. Платформи хмарних сховищ вбудовані у пристрої iOS, проте є багато альтернатив, якщо ОС пристрою – Android.

4. Управління пристроями та мобільними додатками. Це налаштування, моніторинг та керування персональними пристроями, що використовуються.

5. Завантаження додатків із офіційних джерел. Кіберзлочинці створюють шахрайські мобільні програми, які імітують офіційні додатки, щоб отримати конфіденційну інформацію користувачів.

6. Встановлення антивірусного ПЗ.

3.5 Двофакторна автентифікація

Двофакторна автентифікація (2FA) — це специфічний тип багатофакторної аутентифікації, який посилює безпеку доступу, вимагаючи двох методів для підтвердження особи. Ці фактори можуть включати те, що відомо користувачу – логін та пароль, а також додаткові методи для схвалення запитів на автентифікацію, наприклад додаток для смартфона.

2FA захищає від фішингу, соціальної інженерії та атак з використанням паролів, а також захищає ваші логіни від зловмисників, які використовують слабкі або вкрадені облікові дані.

Для впровадження двофакторної автентифікації у додатки ми будемо використовувати бібліотеку Speakeasy, що має вбудовану функцію TOTP (Одноразовий пароль на основі часу). Використовуючи метод на основі TOTP, ми створюємо одноразовий пароль на стороні користувача через програму для смартфона. Це означає, що користувач завжди має доступ до свого одноразового пароля. А також запобігає надсиланню текстового повідомлення сервером при кожній спробі увійти до системи. Крім того, згенерований пароль змінюється через певний проміжок часу, що робить його по суті одноразовим.

Алгоритм дій (рис. 3.18):

1. Внутрішній сервер створює секретний ключ для конкретного користувача (рис. 3.19).

2. Сервер передає цей секретний ключ до програми користувача (рис. 3.20).

3. Додаток ініціалізує лічильник.

4. Телефонна програма генерує одноразовий пароль, використовуючи цей секретний ключ та лічильник.

5. Програма змінює лічильник через певний інтервал і відновлює одноразовий пароль, роблячи його динамічним.

6. Успішне або не успішне проходження авторизації (рис. 3.21).

```
app.post("/totp-secret", (request, response, next) => {
  var secret = Speakeasy.generateSecret({ length: 20 });
  response.send({ "secret": secret.base32 });
});

app.post("/totp-generate", (request, response, next) => {
  response.send({
      "token": Speakeasy.totp({
          secret: request.body.secret,
          encoding: "base32"
      }),
      "remaining": (30 - Math.floor((new Date()).getTime() / 1000.0 % 30))
  });
});

app.post("/totp-validate", (request, response, next) => {
  response.send({
      "valid": Speakeasy.totp.verify({
          secret: request.body.secret,
          encoding: "base32",
          token: request.body.token,
          window: 0
      })
  });
});
```

Рисунок 3.18 – Реалізація алгоритму двофакторної автентифікації.

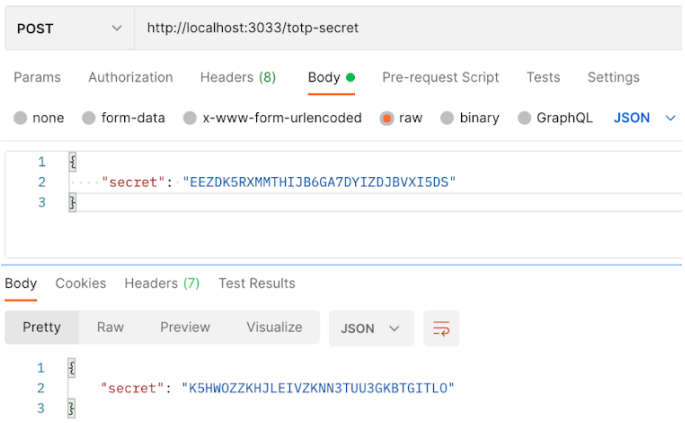

Рисунок 3.19 – Генерація секрету.

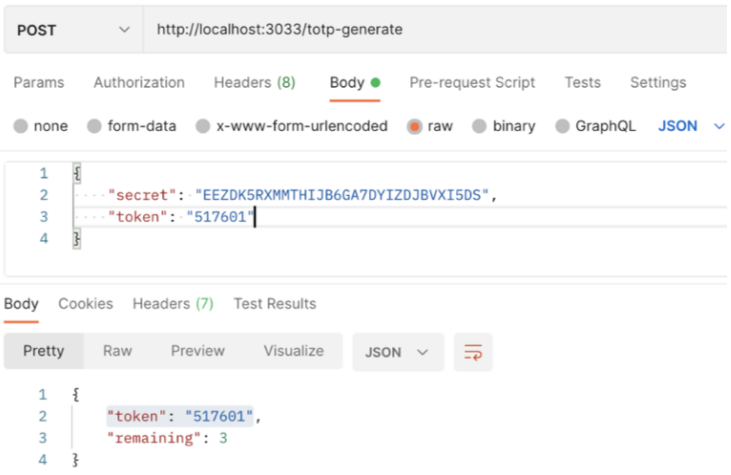

Рисунок 3.20 – Генерація одноразового паролю.

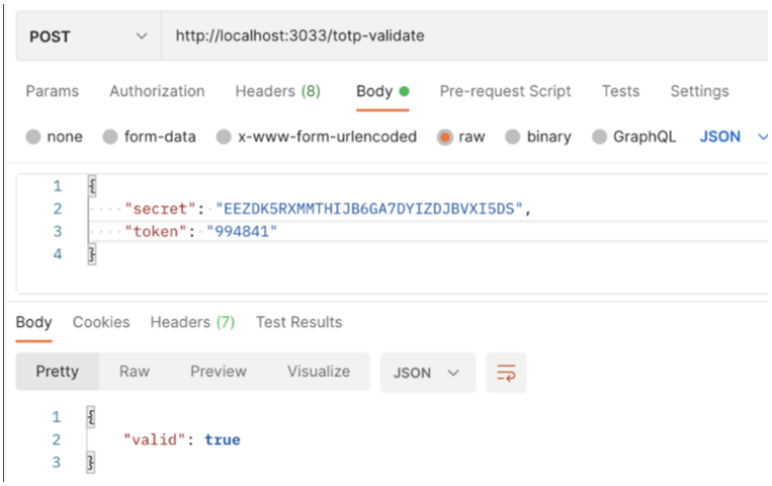

Рисунок 3.21 – Успішне проходження авторизації.

Висновки за розділом 3

Було проведенно тестування антивірусного ПЗ та надано рекомендації користувачам щодо вибору програмного забезпечення.

Найкращими застосунками для потреб загального виявлення загроз та швидкого реагування на загрози виявилися:

1. Avira
2. Sophos Intercept X for Mobile
3. ESET Antivirus

Застосунками, що включають у себе обширний спектр послуг, таких як: сканування надійності Wi-Fi, вбудований VPN, та Web Protection:

1. Bitdefender

2. McAfee AntiVirus Plus

Інші антивірусні ПЗ мають низький рівень реагування, надійності та практичного використання.

Окрім цього, були розроблені рекомендації для запобігання витоку даних та максимального забезпечення конфіденційності зі сторони користувача.

Розроблено додаткову функцію захисту мобільних додатків, що заснована на двофакторній автентифікації. Її впровадження допоможе захистити користувачів від зловмисних дій хакерів, що можуть бути націлені на отримання доступу до конфіденційних даних.

ВИСНОВКИ

У першій частині були розглянуті операційні системи Android та iOS, архітектура додатків для цих систем та механізми шифрування даних. ОС Android шифрують свої дані на рівні файлової системи або на повне шифрування диску. Компанія Apple використовує набір інструментів CryptoKit, що містить безпечні алгоритми хешування, криптографії з симетричним ключем і криптографії з відкритим ключем.

У другому розділі роботи було досліджено проблеми розгортання атак, особливості вразливостей та загроз. Основною проблемою ОС Android є публікування на сервісі Google Play додатків без перевірки коду, що дає змогу зловмисникам поширювати шкідливе програмне забезпечення через офіційні сервіси. Apple приділяє більшк увагу до перевірки додатків перш ніж дозволити розміщення в App Store.

Третя частина включає дослідження інструментів захисту, тестування та встановлення рекомендацій користувачам щодо безпечної роботи з мобільними пристроями. Було проведено дослідження антивірусних ПЗ, які мають декілька параметрів для потреб користувача: швидкість сканування, швидкість реагування на події, вбудовані функції: сканування надійності Wi-Fi, вбудований VPN, та Web Protection. Також розроблено додатковий інструмент захисту мобільних додатків – двофакторна автентифікація.

Отже, поставлені задачі та мети диплої роботи виконано, а саме:

- досліджено архітектуру операційних систем мобільних пристроїв, компоненти мобільних додатків та нормативно правову базу;
- здійснено опис проблем розгортання атак, особливості вразливостей та загроз для мобільних пристроях;
- досліджено інструменти, що використовуються для захисту мобільних пристроїв, а також встановлено рекомендації користувачам для безпечної роботи з мобільними пристроями;

- проведено тестування антивірусного ПЗ та визначено найбільш надійні інструменти захисту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Android Operating System [Електронний ресурс] – Режим доступу до документа: https://www.investopedia.com/terms/a/android-operating-system.asp

2. Android Architecture [Електронний ресурс] – Режим доступу до документа: https://source.android.com/devices/architecture

3. Android Architecture [Електронний ресурс] – Режим доступу до документа: https://www.geeksforgeeks.org/android-architecture/

4. Application Fundamentals [Електронний ресурс] – Режим доступу до документа: https://developer.android.com/guide/components/fundamentals

5. Encryption [Електронний ресурс] – Режим доступу до документа: https://source.android.com/security/encryption

6. Full-Disk Encryption [Електронний ресурс] – Режим доступу до документа: https://source.android.com/security/encryption/full-disk

7. IOS Operation System [Електронний ресурс] – Режим доступу до документа: https://digitaltechakshay.medium.com/what-is-the-ios-operating-system-b19c5d19f5bc

8. Apple CryptoKit [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/documentation/cryptokit/

9. HashFunction [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/documentation/cryptokit/hashfunction

10. SymmetricKey [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/documentation/cryptokit/symmetrickey

11. PublicKey [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/documentation/cryptokit/p256/signing/publickey

12. OS Security [Електронний ресурс] – Режим доступу до документа: https://www.makeuseof.com/apple-vs-android-which-is-more-secure/

13. Android Open Source Project [Електронний ресурс] – Режим доступу до документа: https://emteria.com/learn/android-open-source-project

14. Google Play Developer Account [Електронний ресурс] – Режим доступу до документа: https://www.goodbarber.com/blog/how-to-open-a-google-play-developer-account-a297/

15. Google Play Protect [Електронний ресурс] – Режим доступу до документа: https://developers.google.com/android/play-protect

16. Unauthorized modification of iOS can cause security vulnerabilities, instability, shortened battery life, and other issues [Електронний ресурс] – Режим доступу до документа: https://support.apple.com/en-mt/HT201954

17. Threats to iOS Mobile Devices [Електронний ресурс] – Режим доступу до документа: https://idency.com/wp-content/uploads/2014/08/Lacoon-White-Paper-iOS-Threats.pdf

18. Rooting [Електронний ресурс] – Режим доступу до документа: https://www.techopedia.com/definition/31284/rooting-smartphones

19. Major vulnerability found in Exynos [Електронний ресурс] – Режим доступу до документа: https://www.sammobile.com/2012/12/16/major-vulnerability-found-on-exynos-4-devices/

20. Zero Day Vulnerability [Електронний ресурс] – Режим доступу до документа: https://www.trendmicro.com/vinfo/us/security/definition/zero-day-vulnerability

21. Security Tips [Електронний ресурс] – Режим доступу до документа: https://developer.android.com/training/articles/security-tips

22. Apple Privacy Policy [Електронний ресурс] – Режим доступу до документа: https://www.apple.com/legal/privacy/en-ww/

23. App privacy details on the App Store [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/app-store/app-privacy-details/

24. Android and IOS security mechanisms. OS vulnerabilities [Електронний ресурс] – Режим доступу до документа: https://www.a1qa.com/blog/android-and-ios-security-mechanisms-os-vulnerabilities/

25. Apple Developer Program [Електронний ресурс] – Режим доступу до документа: https://developer.apple.com/programs/

26. How Malware Keeps Sneaking Past Google Play’s Defenses [Електронний ресурс] – Режим доступу до документа: https://www.wired.com/story/google-play-store-malware/

27. Malware and Anti-Virus Architecture [Електронний ресурс] – Режим доступу до документа: https://eforensicsmag.com/malware-and-anti-virus-architecture/

28. N. K. Dien, T. T. Hieu and T. N. Thinh, "Memory-based multi-pattern signature scanning for ClamAV Antivirus", Proc. Int. Conf. Future Data Secur. Eng., pp. 58-70, Nov. 2014.

29. P. Szewczyk and M. Brand, "Malware detection and removal: An examination of personal anti-virus software", Proc. Austral. Digit. Forensics Conf., pp. 56, Mar. 2008.

30. O. E. Osaghae, F. A. Egbokhare and S. C. Chiemeke, "Design of generic antivirus system", Can. J. Pure Appl. Sci., vol. 2775, pp. 2775, Feb. 2014.