Міністерство освіти і науки України
Національний університет «Чернігівська політехніка»

Карпачев Ігор Ігорович

УДК 004.942; 004.021

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ БЕЗПЕКИ МОБІЛЬНИХ ПРИСТРОЇВ

05.13.06 – інформаційні технології

Автореферат
дисертації на здобуття наукового ступеня
кандидата технічних наук

Чернігів – 2021

Дисертацією є рукопис.

Роботу виконано на кафедрі інформаційних та комп'ютерних систем Національного університету «Чернігівська політехніка» Міністерства освіти і науки України.

Науковий керівник: доктор технічних наук, професор
Казимир Володимир Вікторович,
Національний університет «Чернігівська політехніка», професор кафедри інформаційних та комп'ютерних систем.

Офіційні опоненти: доктор технічних наук, професор
Стеценко Інна Вячеславівна,
Національний технічний університет «Київський політехнічний інститут імені Ігоря Сікорського», професор кафедри автоматизованих систем обробки інформації та управління;

кандидат технічних наук, доцент,
Пєвнєв Володимир Яковлевич,
Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», доцент кафедри комп'ютерних систем, мереж і кібербезпеки.

Захист відбудеться «23» вересня 2021 року о 13^{00} годині на засіданні спеціалізованої вченої ради К 79.051.03 в Національному університеті «Чернігівська політехніка» за адресою: 14035, м. Чернігів, вул. Шевченка, 95.

З дисертацією можна ознайомитись у бібліотеці Національного університету «Чернігівська політехніка» за адресою: 14035, м. Чернігів, вул. Шевченка, 95.

Автореферат розіслано «20» серпня 2021 р.

Учений секретар
спеціалізованої вченої ради В.П. Войтенко

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. У наш час мобільні пристрої дедалі більше стають характерною рисою і необхідним елементом як технологічного розвитку, так і життєзабезпечення. Поступово в умовах всеосяжної діджиталізації вони проникають у всі сфери життя людини. Завдяки мобільним пристроям здійснюється управління складними кіберфізичними системами, у тому числі через мережі Інтернету речей, стає доступним широкий спектр фінансових та соціальних послуг, організується обмін інформацією та моніторинг життєво важливих функцій. Фактично, такого роду пристрої переходять із розряду іграшкових приладів у категорію критичних застосунків.

Питання забезпечення функціональної безпеки критичних застосунків достатньо глибоко висвітлено в наукових працях відомих закордонних та вітчизняних науковців, таких як A. Avizienis, G. Johnson, J.C. Laprie, B. Randell, Є. В. Брежнєв, О. М. Одарущенко, В. В. Скляр, В. С. Харченко. Виходячи із сформованої вказаними вченими загальної таксономії понять загальна проблема забезпечення функціональної безпеки належить до систем моніторингу та управління. На сучасному етапі функціональна безпека повинна спиратися на інформаційну безпеку та кібербезпеку, що доповнюють одна одну. У той час як мета інформаційної безпеки полягає в забезпеченні доступності, цілісності й конфіденційності даних системи, функціональна безпека забезпечує здатність системи виконувати передбачені функції при існуванні ризику виникнення небезпечних подій, в тому числі під впливом зовнішніх загроз.

Що стосується безпосередньо мобільних пристроїв, то тут особливо гостро питання функціональної безпеки постає перед найбільш поширеними реалізаціями на базі ОС Android, дослідженню функціональної безпеки яких присвячена значна кількість публікацій сучасних науковців та фахівців наукових лабораторій IT-корпорацій: J. Warton, P. Roche, S. Felker, A. Саммерс, K. Gopal, A. Davies, M. Zhan, R. Mayer, A. Dao, D. Smith, T. Norby, M. Alison, С. В. Зайцев, М. С. Дорош, І. В. Стеценко та ін. Отримані в межах даних досліджень результати вказують на те, що найбільший рівень загрози слід пов’язувати не з апаратними ресурсами та операційною системою (ОС), а саме з мобільними додатками, що містять у собі такі вразливості, як недосконалості у визначенні привілеїв користувачів, витік інформації, низький рівень захисту функціональних компонентів, уразливості міжкомпонентних комунікацій та інші. Більшість існуючих заходів безпеки, що надаються ОС Android, базуються на попереджувальних діях та системних обмеженнях для забезпечення безпеки платформи загалом.

Слід зазначити, що підходи на основі машинного навчання, статичного аналізу та штучних нейромережевих алгоритмів, що дозволяють визначити рівень загрози на основі базових параметрів програмного коду, наприклад, при аналізі викликів API (Application Programming Interface), також не забезпечують належний рівень функціональної безпеки. Враховуючи значну

затримку у вирішенні питань безпеки та потенційний ризик втрати приватних даних або навіть порушення функціональної безпеки системи, існує потреба в додатковому блоці безпеки, який допоможе повідомити користувача про потенційно шкідливе програмне забезпечення під час виконання. Тому актуальним є наукове завдання із подальшого розвитку інформаційної технології забезпечення функціональної безпеки мобільних пристроїв за рахунок удосконалення існуючої моделі безпеки ОС Android шляхом впровадження методу динамічного виявлення потенційно небезпечних додатків з метою подальшого їх блокування для запобігання небажаних впливів

Зв'язок роботи з науковими програмами, планами, темами. Результати дисертаційної роботи використані при розробці мобільного додатку в межах європейського проєкту за програмою Tempus-IV «Інноваційна гібридна стратегія IT-аутсорсингового партнерства (IHSITOP)», у комерційних науково-дослідних роботах Національного університету «Чернігівська політехніка» при створенні захищеної системи електронного голосування «Mobile-RADA», мобільному додатку «Datatouch (Datascope Ltd.)» для будівельних компаній та розповсюдженій системі доставок «DMS (Datascope Ltd.)».

Мета і завдання дослідження. Мета дисертаційної роботи полягає в покращенні функціональної безпеки мобільних пристроїв за рахунок удосконалення моделі безпеки ОС Android з можливістю врахування ризику використання шкідливих програмних додатків.

Досягнення поставленої мети передбачає вирішення таких завдань:

1. Визначення потенційних загроз функціонуванню мобільних пристроїв.
2. Аналіз існуючої системи безпеки ОС Android щодо загроз безпечній роботі програмних додатків.
3. Розробка моделі прав доступу при взаємодії ОС Android із програмними додатками.
4. Розробка методів динамічного виявлення та блокування потенційно небезпечних додатків.
5. Розробка програмних засобів реалізації запропонованих методів забезпечення функціональної безпеки.
6. Оцінка ефективності розроблених моделей, методів та інформаційної технології шляхом проведення експериментів із реальними пристроями та додатками.

Об'єкт дослідження – система функціональної безпеки ОС Android та процес її функціонування в умовах зовнішніх загроз.

Предмет дослідження – моделі та методи забезпечення функціональної безпеки на рівні операційної системи мобільних пристроїв.

Методи дослідження. При розв'язанні поставлених завдань були використані: методи системного аналізу та методи теорії множин – при аналізі системи безпеки ОС Android та розробці моделі прав доступу,

методи біоінформатики – у процесі розробці методу динамічного виявлення потенційно небезпечних додатків, теорії ймовірностей та математичної статистики – для планування та статистичного аналізу результатів експериментів із реальними додатками, об'єктно-орієнтованого аналізу та графічні нотації UML – при проєктуванні та розробці програмних засобів, які реалізують запропоновану інформаційну технологію забезпечення функціональної безпеки мобільних пристроїв.

Наукова новизна одержаних результатів.

1) **Вперше** запропонована класифікація типів небезпечних додатків, яка, на відміну від існуючих, базується на групуванні дозволів на використання API функцій за ступенем потенційних впливів, що дає можливість оцінити додатки за рівнем небезпеки для користувача при прийнятті рішень на їх використання.

2) **Вперше р**озроблена модель прав доступу при взаємодії ОС Android з програмними додатками, яка, на відміну від існуючих, встановлює відношення між групами, дозволами та API функціями, що дає можливість використовувати псевдосимволи функцій для прискорення ідентифікації додатків.

3) **Удосконалено** метод динамічного виявлення потенційно небезпечних додатків за рахунок використання сигнатур функціональних ланцюжків додатків, що будуються та порівнюються під час виконання додатків, що дозволяє оцінити ризик при їх ідентифікації.

4) **Набула подальшого розвитку** інформаційна технологія забезпечення функціональної безпеки мобільних пристроїв, яка, на відміну від існуючих, дозволяє здійснити динамічне управління процесами використання додатків ОС Android за рахунок аналізу виявленого вектору атаки.

Практичне значення одержаних результатів.

Розроблена інформаційна технологія забезпечення функціональної безпеки мобільних додатків має практичне втілення у вигляді програмного комплексу, до складу якого входять два розроблені програмні засоби:

- програмний засіб AMalDetector, який призначений для моніторингу процесу виконання мобільних додатків та підтримки прийняття рішення щодо шкідливості мобільного програмного забезпечення шляхом комунікації з серверним аналізованим ядром CMMD;

- програмний засіб CMMD, який призначений для виявлення схожості та ідентичності аналізуємої та шаблонної сигнатур функціональних ланцюжків додатку (СФЛД) шляхом сиквенційної агрегації фрагментованих викликів API функцій із використанням методів глобального та локального вирівнювання послідовностей.

Розроблена інформаційна технологія та програмний комплекс перевірені в умовах практичного проведення дослідження на реальних мобільних додатках. Проведені тести підтвердили здатність запропонованої інформаційної технології підвищувати ефективність системи функціональної безпеки мобільних пристроїв на базі ОС Android за істинним позитивним показником (TPR) на рівні 99% при низькому хибно позитивному показнику

(FPR) на рівні 0,02% з одночасним обчислюванням коефіцієнтів ідентичності та подібності для порівнюваних ланцюжків викликів API функцій.

Результати дисертаційних досліджень впроваджені:

- при виконанні міжнародного проєкту Tempus 530319-TEMPUS-1-2012-1-DE-TEMPUS-JPHES «Інноваційна гібридна стратегія IT-аутсорсингового партнерства з підприємствами (IHSITOP)»;
- при виконанні НДР «Розробка програмно-апаратного комплексу захищеної системи електронного голосування «Mobile-RADA» за договорами НУ «Чернігівська політехніка» з Чернігівською обласною радою № 480/18, 492/19, 508/20 та з Корюківською міською радою № 79/482/19;
- при створенні мобільного додатка «Datatouch (Datascope Ltd.)» для будівельних компаній та системи управління доставками «DMS (Datascope Ltd.);
- у навчальному процесі на кафедрі інформаційних та комп'ютерних систем НУ «Чернігівська політехніка» при проведенні лекцій та лабораторних робіт із дисципліни «Програмування мобільних пристроїв» у процесі навчання бакалаврів спеціальності 123 – комп'ютерна інженерія та з дисципліни «Статистичні методи обробки інформації» в процесі навчання аспірантів спеціальності 122 – комп'ютерні науки.

Особистий внесок здобувача. Усі результати, які виносяться до захисту, отримані автором особисто. У роботах [1; 2] здобувачу належать усі теоретичні та практичні результати; також проведено аналіз та обґрунтовано недосконаліть існуючої система привілеїв ОС Android у площині функціональної безпеки; запропоновано методи покращення захисту ОС від шкідливого програмного забезпечення. У роботах [3; 4] здобувачем обґрунтовано використання Java Native Interface (JNI) у критичних з погляду продуктивності частинах програмної системи для підвищення швидкості виконання мобільного програмного додатка, описані розроблені здобувачем моделі та практичні результати експериментів з ними. У роботах [5; 6; 7] здобувачеві належить розробка та реалізація інформаційної технології забезпечення функціональної безпеки ресурсами ОС Android, методів побудови СФЛД на базі динамічного функціонального трасування API викликів, способів обробки послідовностей за допомогою локального та глобального методів порівняння СФЛД, математична модель прав доступу.

Апробація результатів роботи. Основні наукові та практичні результати дисертаційної роботи доповідались та обговорювались на наукових конференціях:

- X Міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем», MODS-2015 (Чернігів-Київ, 22-26 червня, 2015);
- XI Міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем», MODS-2016 (Чернігів, 27 червня – 1 липня, 2016);

- XIII Міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем», MODS-2018 (Київ – Чернігів – Жукин, 25-29 червня, 2018).

Публікації. За матеріалами дисертаційної роботи опубліковано 10 наукових праць, серед яких 1 стаття у періодичному науковому виданні Європейського Союзу з наукового напряму дисертації, 6 статей у наукових фахових виданнях України та 3 публікації у збірниках матеріалів міжнародних наукових конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та 5 додатків. Повний обсяг дисертації становить 141 сторінку, у тому числі: 110 сторінок основного тексту, 52 рисунків, 9 таблиць, список використаних джерел із 142 найменувань та 5 додатків.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** обґрунтовано актуальність теми дисертаційної роботи, сформульовано мету роботи й завдання дослідження, які потрібно вирішити для її досягнення, наведено інформацію про зв’язок роботи з науковими програмами. Подано стислу характеристику результатів дослідження, їхню наукову новизну та практичне значення отриманих результатів. Наведено апробацію та публікацію основних результатів дисертаційної роботи.

У **першому** розділі проведено аналіз вимог до систем функціональної безпеки мобільних пристроїв, у тому числі при захисті від зовнішніх кіберзагроз. Визначено поняття і показники ефективності функціональної безпеки мобільних приладів, показано сучасні методи захисту операційних систем та програмних додатків. Проведено аналіз ресурсів системи функціональної безпеки ОС Android, вказано критичні місця роботи програмних додатків під управлінням цієї системи. Визначено основні типи загроз, що можуть бути пов’язані з використанням програмних додатків ОС Android і виділені потенційні місця загроз функціональній безпеці, зокрема: наявність зловмисного програмного коду в середовищі виконання програмного застосунку або в самому програмному додатку, наявність вразливостей у програмних застосунках та ОС, перехоплення зловмисниками конфіденційних даних, некоректний опис програмного застосунку, блокування апаратних модулів фізичного пристрою. Показано, що забезпечення функціональної безпеки базується на відповідності функціональних властивостей компонентів додатка та системи, у межах якої він виконується. Таким чином, стандарти функціональної безпеки лежать в основі моніторингу електронних та програмних елементів комплексу системного та прикладного програмного забезпечення, що вимагає аналізу кожної окремої функції безпеки на відповідному рівні. Множина API викликів, за допомогою яких додаток взаємодіє з ресурсами ОС, визначає, як повинні діяти деякі програмні компоненти (підпрограми, протоколи та інструменти), якщо їх викликають

програмні додатки. Відстежуючи та аналізуючи API виклики, можна з'ясувати, як програми поводяться, обробляють дані та взаємодіють між собою.

За результатами проведеного аналізу обґрунтовано задачі дослідження та сформульовано основне наукове завдання роботи.

У **другому** розділі описана розроблена модель прав доступу при взаємодії із додатками, яка лежить в основі розробленого методу забезпечення функціональної безпеки ОС «Android». Детально розглянуто процес побудови бази даних шаблонних СФЛД шкідливих додатків на базі проєкту «Android Malgenome Project», процес виділення метаданих для аналізу СФЛД та категоризація API функцій. В основу такої бази даних було покладено дослідницький набір послідовностей API функцій, який представлено у вигляді «.csv» файлу, де кожен стовпчик у таблиці являє собою послідовність API функцій та запитів на дозволи від операційної системи. СФЛД являє собою набір функцій із набору Software Development Kit (SDK) Android, які послідовно використовує програмний додаток під час свого виконання. Приклад фрагментів послідовностей API функцій шкідливих додатків наведено на рис. 1. Рядки відносяться до певних додатків, а стовпчики описують наявність відповідної API функції в бінарному форматі «0» або «1».

1	READ_PHONE_STATE	getBinder	ClassLoader	Landroid.content.Context.registerReceiver	Ljava.lang.Class.getField	Landroid.content.Context.unregisterReceiver
2	1	0	0	1	0	1

Рисунок 1 – Фрагмент сигнатури функціонального ланцюжка з проєкту Malgenome

Фактично, БД шаблонних СФЛД являє собою набір множин функцій Software Development Kit (SDK) Android, що використовуються програмними додатками під час свого виконання.

Оскільки дослідницький проєкт «Android Malgenome Project» не дає ніякої інформації щодо назви або опису впливу, необхідно сформувати додаткові метадані, які визначають вектор атаки шкідливого додатка. До цих метаданих (атрибутів) слід віднести: напрям атаки (група дозволів), характер атаки (дії через певні дозволи) і рівень небезпеки. Базуючись на даних офіційного сайту Google як основного розробника ОС Android, можна виділити множину небезпечних для функціональної безпеки дозволів, що впливають на поточний стан ОС, блокуючи для користувача прямий доступ до критичних ресурсів мобільного пристрою. Така множина може бути представлена у вигляді списку дозволів разом із виділеними рівнями небезпеки для користувача:

- WRITE_CALENDAR (середній);
- CAMERA (помірно високий);
- WRITE_CONTACTS (помірно високий);
- ACCESS_FINE_LOCATION (помірно високий);
- ACCESS_COARSE_LOCATION (помірно високий);
- RECORD_AUDIO (помірно високий);

- CALL_PHONE (високий);
- RECEIVE_BOOT_COMPLETED (помірно високий);
- WRITE_CALL_LOG (дуже високий);
- ADD_VOICEMAIL (середній);
- USE_SIP (середній);
- PROCESS_OUTGOING_CALLS (дуже високий);
- BODY_SENSORS (середній);
- SEND_SMS (високий)
- RECEIVE_SMS (дуже високий);
- READ_SMS (дуже високий);
- RECEIVEW_APP_PUSH (дуже високий);
- RECEIVE_MMS (високий);
- WRITE_EXTERNAL_STORAGE (дуже високий).

API виклики, які користувач може ініціювати, маючи привілеї з наведеного вище списку, можуть бути поділені на дві основні категорії залежно від вектора атаки небезпечного додатка: первинні (функціональна безпека), та вторинні (інформаційна безпека). Така категоризація дозволів та відповідних ним API функцій дозволяє ідентифікувати напрямок атаки шкідливого додатка: функціональна або інформаційна безпека.

Оскільки база даних містить перелік назв API викликів у строковому представленні (API функцій), порівняння послідовностей СФЛД є доволі ресурсномістке завдання, яке можна значно спростити, якщо замінити API функцію на псевдосимвол. При формуванні такого псевдосимволу необхідно враховувати, що для того, щоб програмний додаток міг використовувати будь-яку функцію для доступу до ресурсів операційної системи, йому необхідні дозволи (android permissions), які користувач надає під час інсталяції або виконання додатку. У свою чергу, кожен дозвіл належить до своєї групи дозволів ОС Android. Отже, використовуючи створену базу даних СФЛД, кожну функцію можна закодувати псевдосимволом, який складається з трьох атрибутів: номера SDK-групи, номера дозволу, що належить цій групі, та номера API функції (рис. 2).

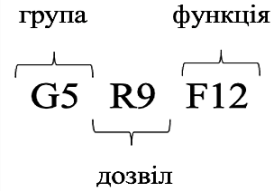

Рисунок 2 – Приклад кодування API функції псевдосимволом

Математична модель псевдосимволу має вигляд трійки:

$$s = (g, r, f),$$

де $s \in S$ – множині псевдо символів, $g \in G$ – множині груп дозволів, r $\in R$ – множині дозволів, f $\in F$ – множині API функцій, причому між множинами G та R, як і між множинами R та F, встановлюється відповідність $1 \to \infty$ (1 до

багатьох), а множини S та F знаходяться у взаємно однонозначній відповідності – множина псевдосимволів має потужність, однакову з потужністю множини API функцій. Побудова псевдосимволу починається у зворотному порядку, тобто спочатку знаходиться індекс функції в таблиці функцій SDK, далі знаходиться позиція дозволу зі списку android.permissions, в який входить ця функція, та група, до якої входить знайдений дозвіл. Такий підхід, за рахунок зменшення довжини аналізованої послідовності символів у порівнянні зі строковим представленням API викликів, дозволяє пришвидшити пошук збігів СФЛД та, одночасно, покращити інформативність при нотифікації користувача.

Відповідно до моделі прав доступу на основі запропонованих псевдосимвольних конструкцій, було побудовано математичну модель послідовності API функцій, за допомогою якої виконується порівняння аналізованої та шаблонної СФЛД. Позначимо всю множину шаблонних СФЛД, що відповідають шкідливим додаткам, як

$$P = \{P_k\},\ k = \overline{1, K},$$

де K – кількість шаблонних СФЛД у базі даних. У свою чергу, кожний шаблон СФЛД є ланцюжок API функцій

$$P_k = \left(p_j^k\right),\ k = \overline{1, K}, j = \overline{1, n},$$

де n – довжина k-*ї* шаблонної послідовності, j – порядковий номер API функції у k-му ланцюжку.

Під час запуску додатку ми отримаємо ланцюжок викликів API функцій, який позначимо як вектор

$$\mathrm{C} = (\mathrm{c_i}),\ \mathrm{i} = \overline{1, \mathrm{m}},$$

де m – довжина зафіксованого фрагмента послідовності API функції.

Завдання полягає в тому, щоб знайти шаблон, який відповідає ланцюжку C^q, побудованому в межах певного кванту часу q. Для його вирішення пропонується метод динамічного виявлення шкідливого додатка, який базується на застосуванні двох відомих алгоритмів з біоінформатики: алгоритму глобального вирівнювання Нідлмана – Вунша та його модифікації – алгоритму локального вирівнювання Сміта – Уотермена. У запропонованому методі селекція підмножини СФЛД, яка має локальні області збігів із фрагментованим ланцюжком API-функцій, спочатку відбувається за допомогою методу Сміта-Ватермана. Цей алгоритм зручний при порівнянні двох послідовностей, довжина яких значно відрізняється. У нашому випадку він буде застосований для пошуку шаблонів із набору P, подібних ланцюжку C^q. У цьому випадку послідовності $P_k = \left(p_j^k\right)$ та $C^q = \left(c_i^q\right)$ є вхідними послідовностями в циклі по k. Враховуючи, що на початку квантування довжини ланцюжків функцій API можуть бути незначними, ця обставина є вирішальною.

Послідовність API функцій у шаблонній СФЛД, яка являє собою множину функцій, може не відповідати послідовності функцій отриманого ланцюжка $C = (c_i)$. Тому у шаблонній $P_k = (p_j^k)$ необхідно упорядкувати перші m символів відповідно до отриманого ланцюжка $C = (c_i)$. Якщо для якогось елемента виконуються $c_i \notin P_k$, то p_i^k присвоюється значення неіснуючого символу □. У результаті буде отримана упорядкована СФЛД P_k^0, з якою і буде порівнюватися ланцюжок C.

На кожному кроці циклу по k вирівнювання послідовностей виконується за допомогою матриці подібності $M_{m\times n}$ з елементами $M_{i,j}$, які обчислюються за правилом:

$$M_{i,j} = max\{M_{i-1,j-1} + w;\ M_{i-1,j} + g;\ M_{i,j-1} + g; 0\},\ i,j > 1, \qquad (1)$$

де w – оцінка за збіг (+1) або невідповідність (–1) символів у рядку i та стовпці j, g – штраф за розрив (g = –1), коли перехід відбувається уздовж рядка або стовпця. Якщо варіанти дають від'ємні значення, результат дорівнює 0.

Матриця заповнюється шляхом обчислення сусідніх значень (діагоналі, верхньої та лівої поточної комірки), починаючи з нульової комірки. Для пошуку оптимального вирівнювання використовується процедура зворотного відстеження шляху, починаючи з комірки з найбільшим значенням і рухаючись до попередніх позицій, поки буде досягнута комірка з оцінкою 0. Серед усіх шаблонів $P = \{P_k\}$ вибираються ті, які мають максимальний бал (без пропусків). Така процедура повторюється щоразу, коли довжина фрагментованого СФЛД збільшується шляхом додавання знову виявлених функцій API, поки довжина ланцюжка C не стане рівною середній довжині безперервної послідовності API функцій у шаблонних СФЛД. Наприклад, для двох шаблонних послідовностей $P_1 = (D, W, A, B, S, K, C, K, V, B)$, $P_2 = (W, A, B, K, C, W, B)$ та зафіксованої фрагментованої послідовності $C_1 = (A, B, K)$ результати локального вирівнювання можуть мати вигляд, представлений на рис. 3.

A□KCKVB
I I
ABK

Рисунок 3 – Приклад локального вирівнювання СФЛД

У цьому прикладі, для спрощення, застосовані позначення псевдосимволів СФЛД буквами англійського алфавіту.

Локальний алгоритм вирівнювання послідовностей вирішує завдання пошуку множини потенційно небезпечних СФЛД, які буде використано для подальшого аналізу. Це зменшує кількість СФЛД, які необхідно аналізувати. У результаті буде обрано один або кілька шаблонів $P' \subset P$.

На другому етапі виявлення шкідливого ПЗ використовується алгоритм глобального вирівнювання послідовностей Нідлмана – Вунша. Алгоритм виконує порівняння послідовностей по всій довжині, що необхідно для

оцінки збігу аналізованого СФЛД з послідовностями множини P'. На цьому кроці розмірність матриці подібності встановлюється таким чином, щоб забезпечити максимальну узгодженість аналізованих СФЛД по довжині. Правило для обчислення значень комірок матриці $M_{m\times n'}$ має таку форму:

$$M_{i,j} = max\{M_{i-1,j-1} + w;\ M_{i-1,j} + g;\ M_{i,j-1} + g;\}, i,j > 1 \quad (2)$$

Значення нижньої правої комірки матриці буде найкращим показником вирівнювання з двох послідовностей.

На рис. 4 наведено приклад глобального вирівнювання для послідовності $C_2 = (A, B, K, C, W)$, отриманої в процесі виконання попереднього етапу:

ABKCWB
I I I I I
ABKCWH

Рисунок 4 – Приклад глобального вирівнювання СФЛД

Для оцінки ризику виявлення шкідливого ПЗ пропонується використовувати коефіцієнт ідентичності, який обчислюється за формулою:

$$I = \frac{N}{max(L_1,\ L_2)} \times 100, \quad (3)$$

де N – кількість псевдосимволів, які повністю збігаються в обох послідовностях, $L_1,\ L_2$ – довжини порівнюваних СФЛД.

Тоді ризик виявлення шкідливого додатка буде дорівнювати

$$R = 1 - I/100. \quad (4)$$

Для наведеного на рис. 4 прикладу коефіцієнт ідентичності дорівнює 83 %, а ризик – 0.17.

Для врахування можливого маскування викликів небезпечних API функцій шляхом мутації певних псевдосимволів, що належать до однієї групи, пропонується додатково використовувати коефіцієнт подібності СФЛД, який розраховується за формулою:

$$S = \frac{N+E}{max(L_1,\ L_2)} \times 100, \quad (5)$$

де E – число псевдосимволів, які не збігаються і належать до однієї групи.

Для прикладу з рис. 4, якщо символи B та H належать до однієї групи, коефіцієнт подібності буде дорівнювати 1.

У **третьому розділі** детально розглянута архітектура запропонованого програмного комплексу забезпечення функціональної безпеки мобільних пристроїв. Проведено аналіз сучасного програмного забезпечення функціонального трасування API викликів шкідливих додатків для операційної системи Android. Як базове середовище дослідження був вибраний емулятор Genymotion. Для здійснення динамічного функціонального трасування використовувався Frida Framework, який дозволяє вбудовувати фрагменти коду, написані мовою JavaScript, у будь-які програмні додатки, в тому числі для ОС

Android. Запропонована функціональна схема перехоплення та декорування API виклику, яка представлена на рис. 4.

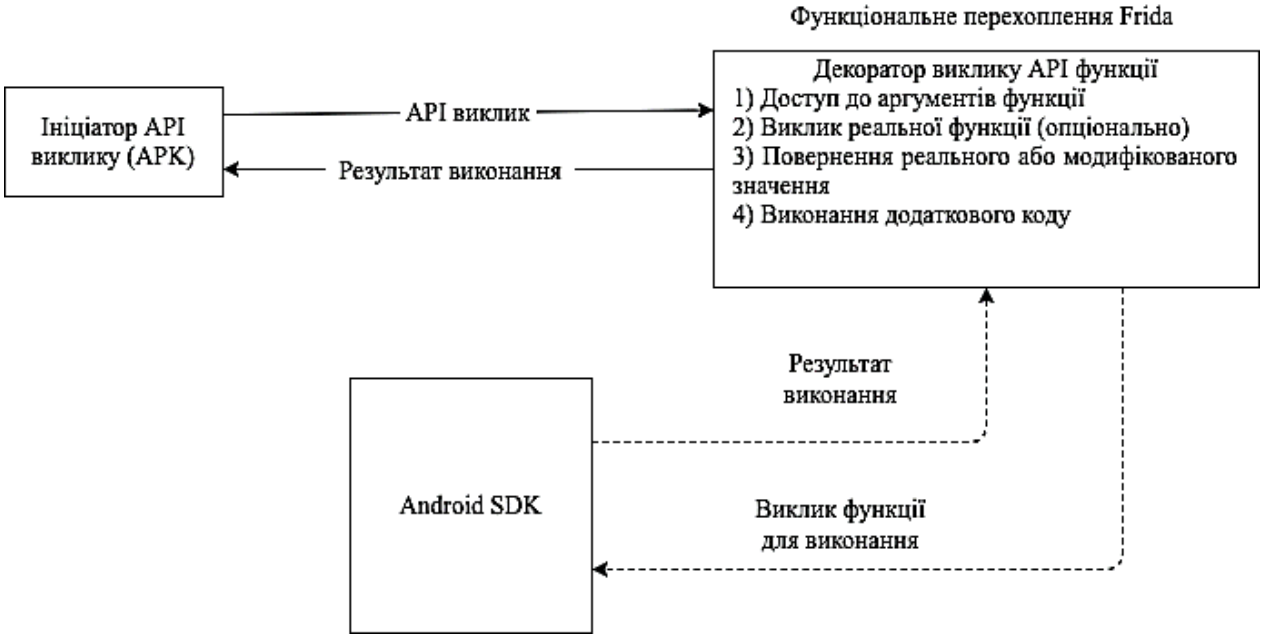

Рисунок 4 – Схема перехоплення API виклику

Розроблений програмний комплекс для виявлення шкідливих додатків складається з двох програмних засобів: AMalDetector (Android Malware Detector) та CMMD (Cloud Mobile Malware Detector). Схема використання програмного комплексу забезпечення функціональної безпеки наведена на рис. 5.

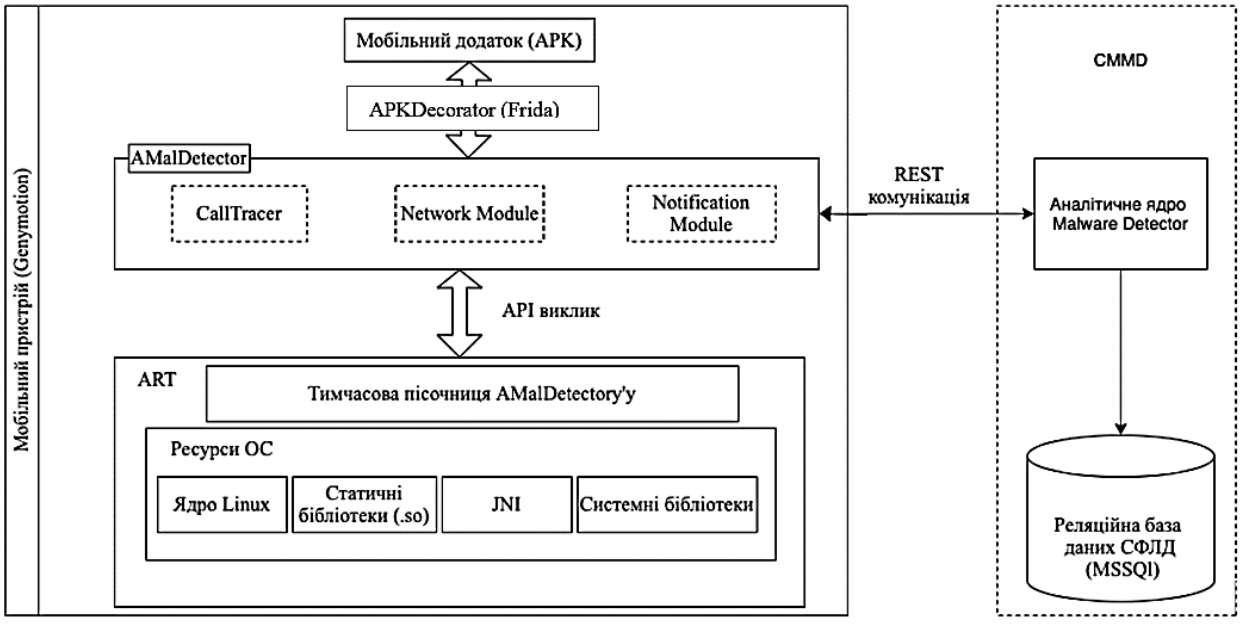

Рисунок 5 – Схема використання програмного комплексу

Програмний засіб AMalDetector являє собою клієнтську частину архітектури комплексу функціональної безпеки, яка виконується на

мобільному пристрої та базується на декоруванні функціоналу програмного додатка Android за допомогою Frida. Архітектура програмного засобу AMalDetector складається з трьох модулів: CallTracer, NetworkModule, NotificationModule. Аналізований APK (Application Package) додатка ініціює API виклики з Android SDK, які перехоплюються Frida і передаються на AMalDetector. APKDecorator є скрипт, який вбудовується в Frida. Він написаний на Python та JavaScript та виступає функцією зворотного зв'язку між потенційно небезпечним додатком та модулем CallTracer. Модуль CallTracer використовує тимчасову пісочницю в середовищі виконання ART (Android RunTime) для серіалізації СФЛД додатка у файл. У поточній реалізації цей модуль також використовується для видалення тимчасових файлів, якщо аналізований додаток не є шкідливим.

NetworkModule використовує мережеве програмне забезпечення операційної системи (Network API Android SDK) для побудови та надсилання запиту на сервер CMMD, використовуючи REST сервіси CMMD. NotificationModule формує текстову нотифікацію користувача про потенційну небезпеку шляхом візуалізації значень атрибутів вектора атаки та отриманих результатів аналізу СФЛД.

Програмний засіб CMMD реалізований у вигляді веб-додатку як Azure-сервіс і призначений для визначення ідентичності аналізованої та шаблонної СФЛД за допомогою розробленого методу динамічного виявлення потенційно небезпечних додатків, включаючи конкатенацію фрагментованих наборів API функцій, упорядкування шаблонних СФЛД та послідовне використання методів локального та глобального вирівнювання. За результатами порівняння СФЛД сервер CMMD надсилає об'єкт-відповідь із значеннями атрибутів вектора атаки для нотифікації користувача. Приклад нотифікації користувача про ідентифікацію шкідливого додатка наведено на рис. 6.

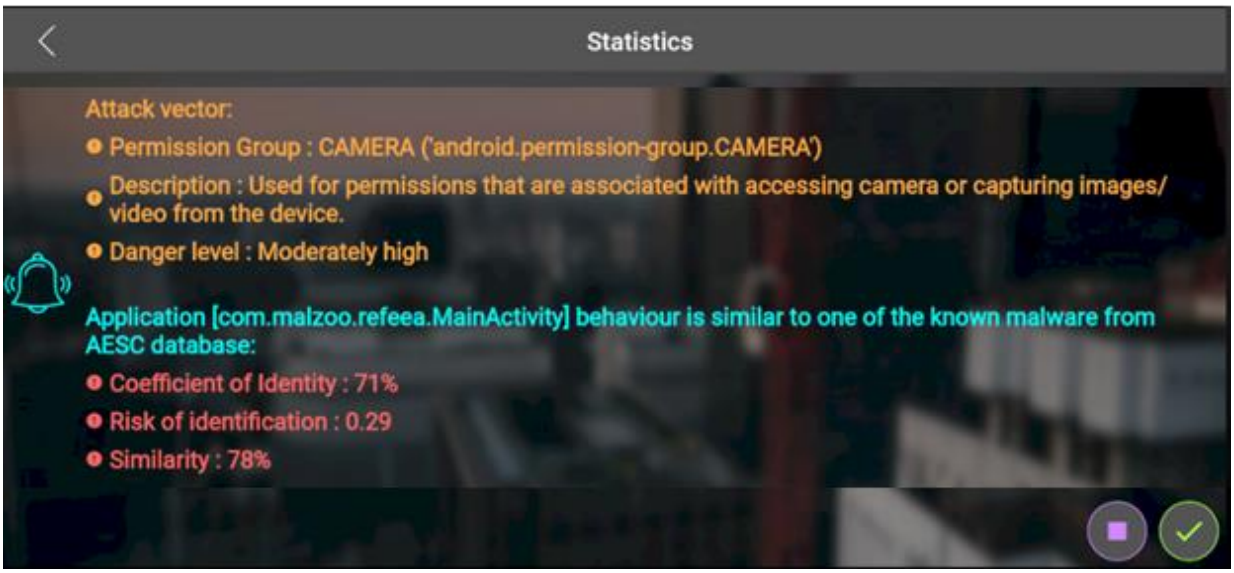

Рисунок 6 – Приклад нотифікації користувача

У **четвертому розділі** описана розроблена інформаційна технологія забезпечення функціональної безпеки та розглянуто результати експериментів з обчисленням відповідних метрик, проведено порівняння з відомими результатами, отриманими іншими методами динамічного пошуку шкідливого програмного забезпечення.

Схема розробленої інформаційної технології зображена на рис. 7.

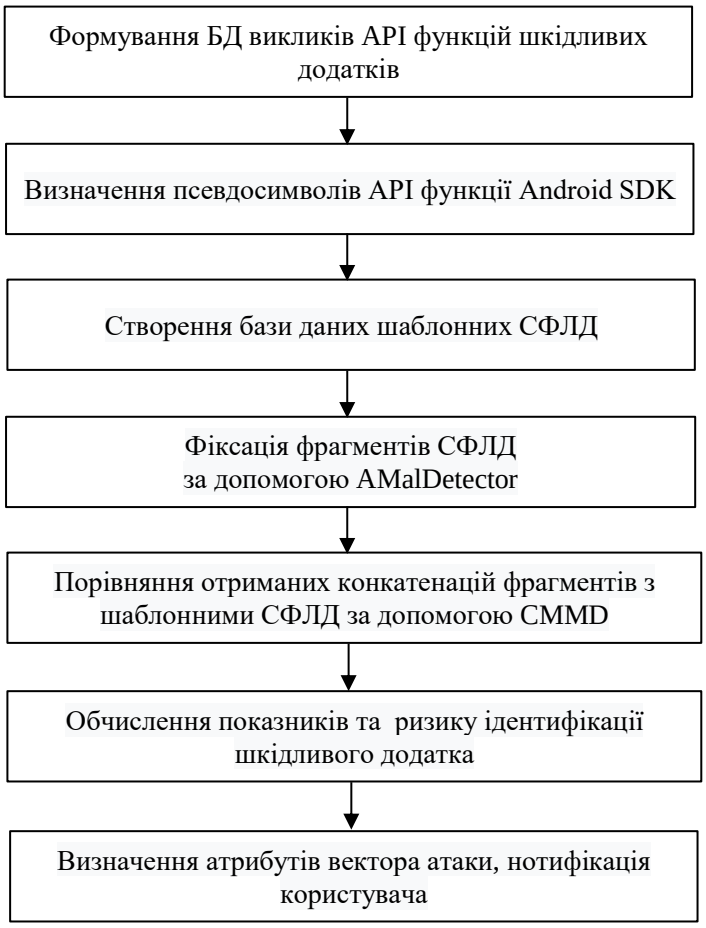

Рисунок 7 – Схема розробленої інформаційної технології

Формування БД викликів API функцій шкідливих додатків відбувається на основі даних проєктів, що аналізують шкідливі додатки ОС Android, наприклад «Malgenome». Далі API функції кодуються псевдосимволами, з використанням яких будуються шаблонні СФЛД. Клієнтський модуль AMalDetector фіксує фрагменти сигнатури викликів протягом часового інтервалу q. При отриманні чергового фрагмента аналізованої СФЛД відбувається конкатенація фрагментованих послідовностей. Серверний модуль CMMD обробляє вхідні послідовності API функцій та за допомогою запропонованого методу порівнює отримані фрагменти API функцій, або їх конкатенації, з шаблонними СФЛД з БД. Фінальним результатом роботи програмного комплексу є обчислення ризику ідентифікації шкідливого додатка на основі коефіцієнта ідентичності та нотифікація користувача шляхом візуалізації атрибутів вектора атаки.

Для оцінки ефективності розробленої інформаційної технології використовувалися матриця невідповідностей задачі класифікації в комбінації з відомими метриками:

$$TPR = \frac{\text{TP}}{\text{TP+FN}} = Recall \quad (5)$$

$$FPR = \frac{\text{FP}}{\text{FP+TN}} \quad (6)$$

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

$$\text{F} - \text{score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall+Precision}} \quad (8)$$

Для обчислення істино позитивної (*True Positive Rate – TPR*, або *Recall - повнота*) та істино негативної (*False Positive Rate - FPR*) метрик класифікації використовувалися елементи матриці невідповідності, а саме: *TP (true positive)/TN (true negative)* – число правильно ідентифікованих шкідливих/нешкідливих додатків, та *FP(false positive)/FN (false negative)* – число неправильно ідентифікованих шкідливих/нешкідливих додатків. Щодо інших метрик, то *Precision (влучність)* – це відношення правильно ідентифікованих шкідливих додатків до загального числа шкідливих додатків, а *F-score* є середнім гармонійним влучності та повноти, дає їх агреговану оцінку.

При тестуванні інформаційної технології було проведено 3 експерименти для 523 програмних додатків у вигляді APK-файлів, отриманих із відомих репозиторіїв шкідливих додатків та з Google Play Store. Кожен експеримент характеризувався двома основними параметрами: сумарна кількість додатків та кількість шкідливого програмного забезпечення в межах цього експерименту. Результати експериментів представлені в табл. 1.

Таблиця 1 – Результати експериментів з виявлення шкідливих додатків

№ Експ.	Кількість APK	Malware (%)	TPR (%)	FPR (%)	Влучність	Повнота	F-score
1	100	94	99	67	0.96	0.99	0.97
2	250	85	99	5.4	0.99	0.99	0.99
3	500	70	99	1.3	0.99	0.99	0.99

Отримані результати показують, що збільшення числа нешкідливих додатків може призвести до значного зменшенням значення FPR. Цей факт підтверджує правильність гіпотези про те, що ланцюжки СФЛД шкідливих і нешкідливих додатків суттєво відрізняються. Таким чином, запропонований метод динамічного виявлення шкідливих додатків шляхом порівняння СФЛД здатен ефективно вирішувати задачу ідентифікації. При використанні додатків із перевірених офіційних джерел цей метод буде коректно їх класифікувати. Більшість сучасних онлайн-магазинів мобільного

програмного забезпечення, включаючи Google Play Store, являєть собою середовище, в якому значною мірою переважають нешкідливі програмні додатки, що збігається з вхідними даними експерименту номер 3 із найнижчим показником FPR.

Результати експериментів були порівняні з відомими результатами, отриманими за допомогою алгоритмів машинного навчання (MLA – Machine Learning Algorithm), таких як K-найближчий сусід (KNN – K-nearest neighbours) та багатошаровий персептрон (MLP – multilayer perceptron). У табл. 2 наведено дані порівняння методів виявлення шкідливих додатків для вибірки об'ємом в 1100 АРК, з яких 100 – шкідливі додатки, відібрані випадковим чином з БД Android Malgenome Project, а 1000 АРК нешкідливих додатків отримані з Google Play Store. У табл. 2 запропонований метод позначений як СФЛД. Загальні характеристики генеральної сукупності шкідливих додатків відносно СФЛД такі: загальне число шкідливих додатків 3702, максимальна довжина шаблонної СФЛД дорівнює 112 псевдосимволів, а середня довжина – 32 псевдосимволи.

Таблиця 2 – Результати порівняння методів виявлення шкідливих додатків

Метод	TPR (%)	FPR (%)	Влучність	Повнота	F-score
MLP	93.03	6.97	0.93	0.972	0.944
KNN	98.63	1.37	0.98	0.986	0.986
СФЛД	**99.10**	**0.20**	**0.99**	**0.991**	**0.990**

У табл. 2 наведені усереднені дані щодо TPR=0,991 для запропонованого методу з використанням СФЛД за результатами 10 експериментів із випадковою вибіркою по 100 базових СФЛД шкідливих додатків (точність результату становить 0,007 для довірчої ймовірності 0,95).

MLA забезпечує достатньо хороше значення TPR, але з досить високим FPR, який погіршує загальний показник F-score, що робить використання цього методу обмеженим у змішаних середовищах. Водночас, зважаючи на порівняння F-score, можна зробити висновок, що гармонійне середнє демонструє збалансовану оцінку результатів експериментів. Попри те, що значення метрики F-score в експерименті з СФЛД та експерименті з використанням KNN майже збігаються, останній характеризується значенням FPR=1,37 %, в той час як запропонований метод показує значно нижче значення FPR=0,2 %.

Отже, можна зробити висновок, що спроможність розробленого програмного комплексу хибно ідентифікувати нормальний програмний додаток як шкідливий набагато менше.

З погляду тривалості виявлення шкідливого програмного забезпечення, яка є значною для клієнтських реалізацій методів MLA безпосередньо на

мобільних пристроях, запропонований метод виявлення шкідливого ПЗ не має цього недоліку, оскільки його алгоритми та аналіз СФЛД виконуються на серверній стороні у хмарному середовищі, де обчислювальний потенціал набагато вищий. Тож час відгуку буде визначатися передусім швидкістю мережі. Також перевагою віддаленого сервера є логічна інкапсуляція реалізації разом із неможливою процедурою зворотного інжинірингу в порівнянні з рішенням на базі клієнта Android, де навіть обфускація коду не є надійним способом інкапсуляції реалізації. Але, враховуючи необхідність виконання порівняння СФЛД при виконанні додатків у реальному часі, для запобігання можливостей впливу на функціональну безпеку мобільного пристрою використання запропонованого методу, як і всіх інших, слід вважати більш доречним на етапі попередньої перевірки додатків.

ВИСНОВКИ

У дисертаційній роботі сформульовано та вирішене актуальне наукове завдання з подальшого розвитку інформаційної технології забезпечення функціональної безпеки мобільних пристроїв за рахунок удосконалення існуючої моделі безпеки ОС Android шляхом впровадження методу динамічного виявлення потенційно небезпечних додатків.

Для досягнення поставленої мети, яка полягає в підвищенні ефективності системи функціональної безпеки мобільних пристроїв за рахунок удосконалення моделі безпеки ОС Android з можливістю врахування ризику використання шкідливих програмних додатків, були отримані такі результати:

1. Визначено поняття функціональної безпеки щодо програмних додатків апаратно-програмної платформи Android. Показано, що аналіз ефективності функціональної безпеки має проводитись на рівні категорій показників, через визначення набору властивостей програмного коду додатку, які характеризують процес його функціонування, а також встановлення рівня шкідливості додатка.

2. Проведено аналіз існуючих систем забезпечення функціональної безпеки мобільних додатків ОС Android як комплексу систем моніторингу та управління. Побудовано узагальнену схему роботи мобільного пристрою з погляду забезпечення функціональної безпеки.

3. Виділено основні типи загроз, пов’язаних із використанням програмних додатків, такі як наявність зловмисного програмного коду, наявність вразливостей у програмних застосунках, перехоплення зловмисниками конфіденційних даних, некоректний опис програмного застосунку.

4. Запропоновано класифікацію типів шкідливих додатків, яка, на відміну від існуючих, базується на групуванні API функцій додатків та дає можливість оцінити їх за ступенем потенційних впливів при прийнятті

рішень на використання додатків за запропонованими атрибутами вектора атаки.

5. Розроблена модель прав доступу при взаємодії ОС Android із програмними додатками, яка встановлює відношення між групами дозволів, дозволами та функціями API та дає можливість ввести кодування функцій для їх ідентифікації.

6. Визначено базову модель псевдосимволу СФЛД, яка складається з етапів послідовного знаходження збігів та індексації групи привілеїв, самого привілею та відповідної API функції. Псевдосимволи, створені на основі цієї моделі, дозволяють унікально ідентифікувати API-функції, що використовує програмний додаток, та прискорити пошук збігів СФЛД.

7. Розроблено метод динамічного аналізу програмних додатків, що базується на побудові сигнатури функціонального ланцюжка додатка API функції та порівняння його з шаблонною СФЛД. Визначено основні етапи процесу аналізу додатків ОС Android та розроблено відповідний математичний апарат на основі відомих алгоритмів вирівнювання послідовностей з біоінформатики. Проведено аналіз ефективності роботи запропонованого методу на основі статистичних даних та зразків зловмисного програмного коду з Android Malgenome Project.

8. Проведено аналіз сучасного програмного забезпечення функціонального трасування API викликів для операційної системи Android. Запропонована функціональна схема перехоплення та декорування API викликів за допомогою розроблених скриптів для Frida Framework.

9. Розроблено програмний комплекс, який працює у хмарному середовищі та забезпечує перевірку програмних додатків, що використовуються мобільними пристроями, на предмет їх збігу зі шкідливими додатками за послідовністю викликів API функцій з одночасним інформуванням користувача про можливі наслідки використання шкідливого програмного забезпечення з визначенням потенційного ризику.

10. Визначено особливості експериментального дослідження запропонованих методів забезпечення функціональної безпеки ОС «Android» та способи можливого ухилення шкідливого додатку від ідентифікації у випадку їх виконання у віртуальному середовищі або емуляторі.

11. Здійснена перевірка достовірності отриманих результатів шляхом їх порівняння з відомими результатами для алгоритмів машинного навчання, які підтвердили високий рівень класифікації шкідливих додатків при використанні запропонованого методу за показником F-score, який навіть перевищує відомі аналоги, при точності обчислення TPR в 0,007 та значному зменшенні помилок 2-го роду більше ніж у 5 разів.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Казимир В., Карпачев І. Функціональна безпека архітектури мобільної операційної системи Android. *Технічні науки та технології*: науковий журнал. 2015. № 1 (77). С. 127-134.

2. Kazymyr V., Karpachev I. Improving of existing permission system in Android OS. *International Journal "Information Models and Analyses*. 2015. Vol. 4. P. 336-344.

3. Kazymyr V., Karpachev I. Improving time-critical code performance with JNI. *Mathematical machines and systems*. 2016. Vol. 2. P. 72-77.

4. Казимир В., Карпачев І. Забезпечення контролю доступу додатків до ресурсів ОС Android методом системної безпеки. *Технічні науки та технології*: науковий журнал. 2016. № 4 (6). С. 131-138.

5. Казимир В., Карпачев І., Усик А. Моделі систем безпеки ОС Android. *Технічні науки та технології*: науковий журнал. 2018. № 2 (12). С. 116-126.

6. Казимир В., Карпачев І., Сіпаков В. Динамічний аналіз послідовностей API-викликів OS Android. *Технічні науки та технології*: науковий журнал. 2019. № 4 (18). С. 85-91.

7. Карпачев І., Казимир В. Виявлення шкідливих додатків ОС Android по сигнатурі функціонального ланцюжка додатку. *Технічні науки та технології*: науковий журнал. 2021. № 1(23). С. 109-117.

8. Карпачев И.И. Системо-центрическая безопасность в Android // *Математичне та імітаційне моделювання систем «МОДС 2015» :* X Міжнародна науково-практична конференція (Чернігів, 22-26 Червня 2015 р.). Чернігів, 2015. С. 422-424.

9. Karpachev I. Bounded context in strategic domain-driven design // *Математичне та імітаційне моделювання систем «МОДС 2016»*: XI Міжнародна науково-практична конференція (м. Чернігів-с. Жукін, 27 червня – 1 липня 2016 р.). Чернігів, 2016. С. 398-400.

10. Казимир В.В., Карпачев І.І., Литвин С.В., Усік А.М. Архітектура моделей системи безпеки мережі інтернету речей на базі ОС Android // *Математичне та імітаційне моделювання систем «МОДС 2018»*: XIII Міжнародна науково-практична конференція (Київ-Чернігів-Жукін, 25-29 червня 2018 р.). Чернігів, 2018. С. 335-336.

АНОТАЦІЯ

Карпачев І. І. Інформаційна технологія забезпечення функціональної безпеки мобільних пристроїв. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Національний університет «Чернігівська політехніка», Чернігів, 2021.

Дисертація присвячена виявленню потенційно небезпечних програмних додатків задля забезпечення функціональної безпеки мобільних пристроїв,

що функціонують на базі операційних систем Android. У межах роботи вперше запропонована класифікація типів небезпечних додатків, яка базується на групуванні дозволів на використання API функцій за ступенем потенційних впливів, що дає можливість оцінити додатки за рівнем небезпеки для користувача при прийнятті рішень на їх використання. Розроблена модель прав доступу при взаємодії ОС Android з програмними додатками, яка встановлює відношення між групами, дозволами та API функціями, що дає можливість використовувати псевдосимволи функцій для прискорення ідентифікації додатків. Удосконалено метод динамічного виявлення потенційно небезпечних додатків за рахунок використання сигнатур функціональних ланцюжків додатків, що будуються та порівнюються під час виконання додатків, що дозволяє оцінити ризик при їх ідентифікації. Набула подальшого розвитку інформаційна технологія забезпечення функціональної безпеки мобільних пристроїв, яка дозволяє здійснити динамічне управління процесами використання додатків ОС Android за рахунок аналізу виявленого вектору атаки. Розроблена інформаційна технологія має практичне втілення у вигляді програмного комплексу, до складу якого входять два програмних засоби, які працюють з використанням хмарного середовища та реалізовані на мовах C#, Kotlin і JavaScript.

Ключові слова: мобільний пристрій, функціональна безпека, ОС Android, модель прав доступу, API функція, сигнатура функціонального ланцюжка додатку.

АННОТАЦИЯ

Карпачев И.И. Информационная технология обеспечения функциональной безопасности мобильных устройств. – На правах рукописи.

Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.06 - информационные технологии. - Национальный университет «Черниговская политехника», Чернигов, 2021.

Диссертация посвящена выявлению потенциально опасных приложений для обеспечения функциональной безопасности мобильных устройств, функционирующих на базе операционных систем Android. В рамках работы впервые предложена классификация типов опасных приложений, основанная на группировке разрешений на использование API функций по степени потенциальных воздействий, что позволяет оценить приложения по уровню опасности для пользователя при принятии решений на их использование. Разработана модель прав доступа при взаимодействии ОС Android с программными приложениями, которая устанавливает отношение между группами, разрешениями и API функциями, что позволяет использовать псевдосимволы функций для ускорения идентификации приложений. Усовершенствован метод динамического обнаружения потенциально опасных приложений за счет использования сигнатур функциональных

цепочек приложений, которые строятся и сравниваются при выполнении приложений, что позволяет оценить риск при их идентификации. Получила дальнейшее развитие информационная технология обеспечения функциональной безопасности мобильных устройств, которая позволяет осуществить динамическое управление процессами использования приложений ОС Android за счет анализа обнаруженного вектора атаки. Разработанная информационная технология имеет практическое воплощение в виде программного комплекса, в состав которого входят два программных средства, которые работают с использованием облачных технологий и реализованы на языках C #, Kotlin и JavaScript.

Ключевые слова: мобильное устройство, функциональная безопасность, ОС Android, модель прав доступа, API функция, сигнатура функциональной цепочки приложения.

ABSTRACT

Karpachev I. I. Information technology to ensure functional safety of mobile devices. – Manuscript copyright.

The dissertation for the scientific degree of the candidate of technical sciences (PhD) in specialty 05.13.06 – information technologies. – National University “Chernihiv Polytechnica”, Chernihiv, 2021.

The dissertation is devoted to the identification of potentially dangerous applications to ensure the functional safety of mobile devices running on Android operation systems. For the first time, the classification of types of malicious applications is proposed. It is based on the grouping of permissions to use API functions by the degree of potential impact, which allows to assess the applications by the level of danger to the user for making decision to use it. The model of access permissions for interaction OS Android with applications is developed, which establishes the relationship between groups, permissions and API functions. It allows to use function aliases to speed up the identification of applications. The method of dynamic detection of potentially dangerous applications is improved due to the use of application functional chain signatures, which are built and compared during the execution of applications that allows to assess the risk of identification. The information technology for ensuring the functional safety of mobile devices has been further developed, that allows to control the processes of using Android applications by dynamic analysis the detected attack vector. The developed information technology has a practical embodiment in the form of a software package, which includes two software tools that work in cloud environment and implemented by using C #, Kotlin and JavaScript.

Key words: mobile device, functional safety, API function, application functional chain signature.