КОМПЬЮТЕРНЫЕ НАУКИ

УДК 004.056.53

БАГАТОРІВНЕВИЙ ПІДХІД ДО ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ВИКОРИСТАННЯ ДОДАТКІВ В ОПЕРАЦІЙНІЙ СИСТЕМІ ANDROID

КУПЕРШТЕЙН Л.М., ВОЙТОВИЧ О.П., ОСТАПЕНКО-БОЖЕНОВА А.В., ПРОКОПЧУК С.А.

Аналізуються можливі загрози комерційним додаткам на мобільних пристроях з операційною системою Android, існуючі методи та засоби захисту додатків від несанкціонованого доступу. Розробляється багаторівнева модель захисту та програмний засіб для мобільних пристроїв під керуванням операційної системи Android, яка включає в себе модуль захисту коду, модуль віддаленого контролю, модуль захисту бази даних і модуль багатофакторної автентифікації.

Ключові слова: несанкціонований доступ та використання, ОС Android, загрози мобільному додатку, багаторівневий захист.

Key words: unauthorized access and use, Android OS, to mobile application threats, multilevel protection.

Вступ

З моменту виходу ОС Android на споживчий ринок в 2008 році увага до даної операційної системи збільшується. З кожним роком кількість додатків, що випускаються для цієї операційної системи, зростає в геометричній прогресії. Додатки використовуються для полегшення різних дій та завантажуються мільярдами людей по всьому світу [1]. За останній час додатки поширилися настільки, що практично кожен аспект людської діяльності тепер можна здійснити за допомогою програми, написаної для ОС Android [2].

Нові програми продовжують з'являтися, надаючи послуги, починаючи від новин, погоди та розваг, до таких серйозних компаній, як банківська справа, медична допомога, фінанси та навіть безпека у домі. Додатки для цих підприємств містять конфіденційну та особисту інформацію (наприклад, дані банківського рахунку, хвороби та ліки, інвестиційна таємниця тощо). Комерційні додатки, зазвичай, не мають належного захисту як коду додатку, так і даних, що зберігаються [1-3]. Тому актуальною задачею є вдосконалення захисту мобільних додатків.

Отже, метою дослідження є підвищення захищеності додатків в операційній системі Android від несанкціонованого використання як коду додатку, так і даних.

Постановка задачі

Викрадення даних, що зберігаються у додатках, завжди вважається однією з найбільш критичних загроз безпеці Android. Дослідження показують [4, 5], що це може статися навіть для додатків, які в принципі не мають вразливостей, але вразливості можуть бути у самій операційній системі, наприклад, спільне використання мережевих даних. Під час атаки шкідливий додаток може функціонувати у фоновому режимі і збирати дані цільового додатку.

Багато додатків використовують інформацію, що зберігається в базах даних. Крім того, додаток може взаємодіяти з базами даних іншого додатку, що надає такі функціональні можливості. Вразливий додаток може дозволити шкідливій програмі порушувати цілісність та конфіденційність даних, що зберігаються в базах даних [5, 6]. Така легкість в отриманні несанкціонованого доступу до вмісту файлів спровокувала появу великої кількості спеціальних інструментів, розрахованих на усунення проблеми доступу до коду додатків.

Крім того, важливою проблемою залишається можливість аналізу коду додатку за допомогою реверс - інженірингу. А саме, дослідження коду додатку, а також документації на нього з метою розуміння принципів його роботи, захисних механізмів, зберігання даних тощо для виконання несанціонованої зміни або копіювання, використання додатку чи іншого об'єкта з аналогічними функціями [7, 8].

Загалом можна виділити такі загрози безпеки для Android-додатків:

– несанкціонований доступ до даних у додатку;
– перехоплення даних в каналах передачі;
– несанкціонований доступ до даних у базах даних;
– аналіз коду додатку;
– несанкціоноване використання додатку.

Існуючі засоби захисту додатків реалізують окремі механізми (наприклад, тільки обфускацію коду або автентифікацію користувачів при доступі до даних), не враховуючи весь комплекс загроз, що може призвести до зменшення рівня безпеки, замість очікуваного покращення. А багаторівневий підхід, який враховує різноманітні особливості функціонування додатків у операційній системі Android, дозволить реалізувати перекриття вказаних загроз і тим самим покращити безпеку додатків.

Багаторівнева модель захисту додатків в ОС Android

Для захисту мобільних Android-додатків пропонується комплексний системний підхід на основі багаторівневої моделі. На рис. 1 наведена узагальнена графічна модель такого підходу.

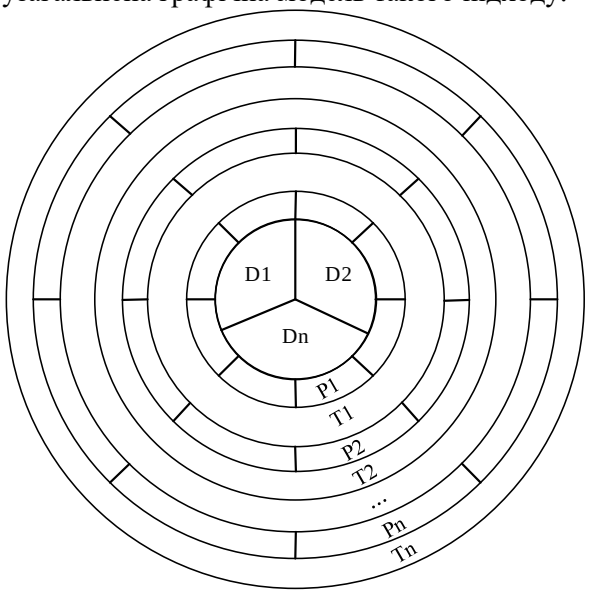

Рис. 1. Загальна графічна модель багаторівневого захисту

Рівень захищеності на мобільних пристроях з ОС Android можна описати такою моделлю:

$$Z = \{D, T, P\},$$

$$D = \{D1, D2, ..., Dn\},$$

$$T = \{T1, T2, ... Tn\},$$

$$P = \{P1, P2, ..., Pn\},$$

де **D** – об'єкт захисту; **T** – методи захисту; **P** – загрози.

В центрі моделі знаходяться об'єкти захисту додатку **D**, в кільцях навколо об'єктів − загрози **P** і контрзаходи **T**, що протидіють цим загрозам.

Для підвищення захищеності мобільного додатку, а саме зменшення ймовірності виникнення та реалізації атак шодо його несанкціонованого використання зловмисником пропонується структура системи захисту з певною надлишковістю, а саме перекриттям загроз низкою контрзаходів. На рис. 2 наведена графічна модель багаторівневого захисту мобільного Android-додатку, який містить чутливу інформацію з урахуванням типових загроз несанкціонованого використання.

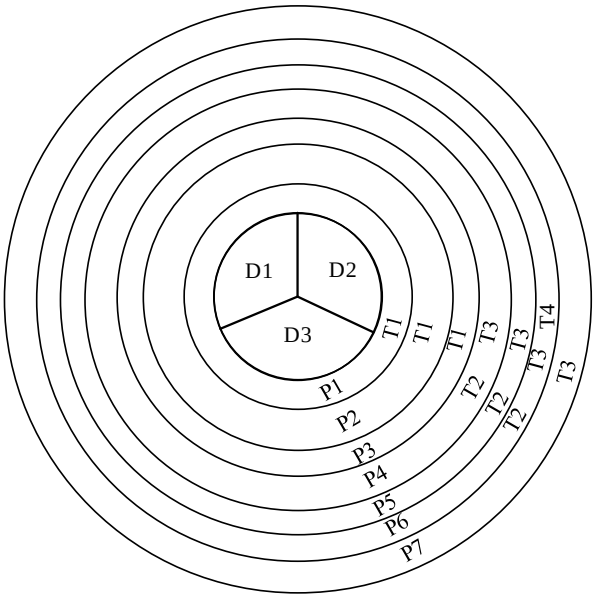

Рис. 2. Графічна модель багаторівневого захисту мобільних додатків з ОС Android від несанкціонованого використання

Об'єктами захисту є:

– код додатку D1,

– дані додатку D2,

– доступ до інфтерфейсу додатку D3.

Методи та засоби захистуAndroid-додатку:

– обфускація коду Т1,

– автентифікація Т2,

– віддалений контроль та управління додатком Т3,

– захист бази даних (БД) Т4.

Зарози несанкціонованого використання додатку та відповідні контрзаходи:

– копіювання коду Р1 захищається Т1;

– аналіз коду Р2 захищається Т1;

– аналіз структури коду Р3 захищається Т1;

– заміна даних Р4 захищається Т2 і Т3;

– крадіжка даних Р5 захищається Т2 і Т3;

– перегляд даних Р6 захищається Т2, Т3 та Т4;

– перехоплення в каналах передачі Р7 захищається Т3.

Застосування такого підходу дозволить значно підвищити надійність системи захисту, а також забезпечити гнучкість у її комплектуванні залежно від пріоритетів користувача.

Дана модель має надлишковість методів захисту, але її перевагою є перекриття деяких загроз кількома методами захисту, що значно зменшує ймовірність виникнення цієї загрози.

Архітектура системи захисту

На основі моделі багаторівневого захисту розроблено та реалізовано архітектуру системи захисту мобільного додатку (рис. 3).

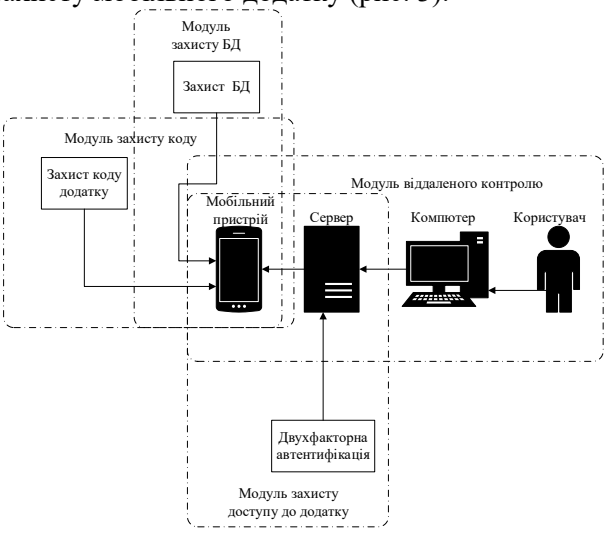

Рис. 3. Архітектура системи захисту

Архітектура системи (рис. 3.) включає модуль обфускації коду додатку для захисту від реверс-інженірингу, модуль багатофакторної автентифікації для контролю доступу до додатку, модуль віддаленого контролю додатку та управління даними, модуль захисту БД для шифрування даних додатку.

Для спрощення впровадження та використання системи захисту пропонується її розробка, програмна реалізація у вигляді бібліотеки.

Модуль захисту коду додатку від реверс-інженірингу використовує обфускації коду під час збору додатку в інсталяційний apk-файл.

Модуль віддаленого контролю має серверну і клієнтську частину. Користувач за допомогою веб-інтерфейсу може відправляти на мобільний пристрій команди керування додатком, а саме:

– очищення даних додатку;

– блокування/розблокування доступу до додатку;

– резервне копіювання даних додатку;

– відновлення даних додатку.

Модуль багатофакторної автентифікації також має клієнт-серверну архітектуру. Спочатку користувач відправляє логін та пароль на сервер. Якщо вони підтверджуються, то додаток за допомогою Bluetooth починає шукати апаратний токен. Для апаратного токену використовується браслет з вбудованим Bluetooth-адаптером. Модуль захисту БД орієнтований на шифрування інформації, у тому числі даних автентифікації. На рис. 4 наведено загальну модель функціонування системи захисту.

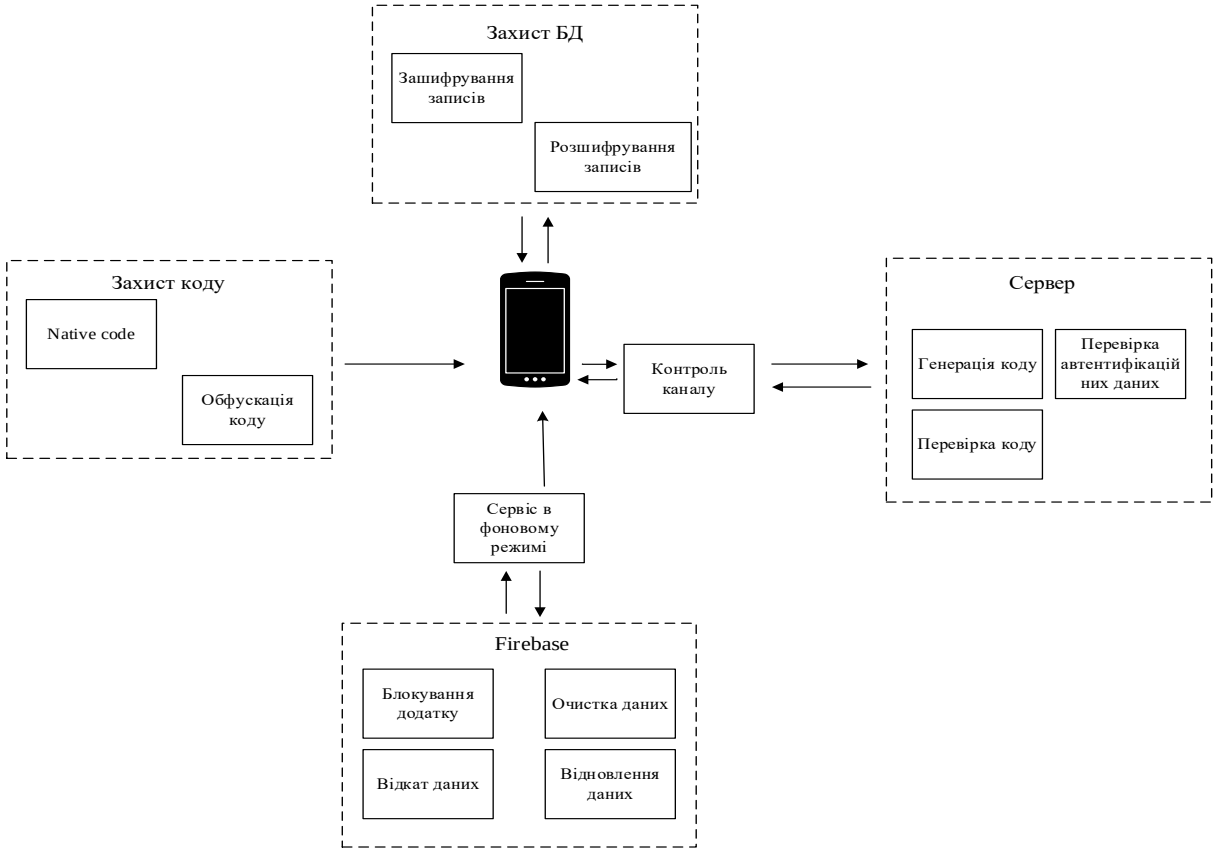

Рис. 4. Загальна модель функціонування системи захисту

Модуль багатофакторної автентифікації використовує сервер для перевірки автентифікаційних даних, генерації коду підтвердження та перевірки цього коду. Модуль віддаленого контролю включає в себе серверну частину, з якої відправляються команди на мобільний пристрій, команди приймаються сервісом, який завжди працює в фоновому режимі. Модуль захисту БД зашифровує всі записи в базу даних протоколом AES і розшифровує їх при читанні з бази. Для захисту роботи з сервером в додатку виконується примусовий контроль каналу, тобто перевірка на те, щоб дані не відправлялись/приймались з посторонніх серверів.

Модуль захисту коду додатку

На основі аналізу досліджених варіантів для захисту коду від реверс-інженірингу було обрано засіб ProGuard [9], код додатку та всіх бібліотек, код яких стискається, зачищається від "мертвого" коду і невикористовуваних змінних і обфускується. Для захисту коду Java винесено частину функцій в native-код. Обфускація коду виконується в момент збору apk-файлу. Ресурси додатку, aidl-файли та код додатку відправляються на Java-компілятор, де код додатку обфускується і отриманий код перетворюється в байт-код. Далі береться код бібліотек, доданих в проект, і разом з основним кодом додатку вони записуються в файл classes.dex. Далі завантажуються інші ресурси, що розміщені в підключених бібліотеках або jar-файлах, і додаток збирається в apk-файл. Потім в файл додається інформація про його підпис debug- або release-ключем.

При цьому виконується лексична обфускація коду додатку. В загальному модель роботи обфускації можна представити так:

$$\mathbf{R} = \{\mathbf{K}, \mathbf{Cl}\}$$

де **R** – результат обфускації; **K** – значення з таблиці заміни; **Cl** – назва класу, методу або змінної.

Під час збору додатку в apk-файл збираються всі файли додатку і викликається функція обфускації коду. Значення, якими заміняються назви класів, методів і змінних, знаходиться в таблиці. Таблиця складається з послідовного списку літер англійського алфавіту a…z. У разі, якщо клас достатньо великий і літер не вистачає, послідовність продовжується, включаючи по 2 літери алфавіту aa…zz. Обирається перший параметр з класу, який буде замінений, з таблиці заміни, по порядку обираються значення для заміни. Якщо значення з таблиці не збігається з назвою класу, методу або змінної, виконується заміна, якщо збігається, береться наступне значення з таблиці. Після завершення обфускації процес збору apk-файлу продовжується.

Модуль віддаленого контролю додатком

Для віддаленого контролю та управління додатком використані такі функції: повне очищення даних додатку; блокування входу в додаток; створення бекапу даних додатку на сервері; відновлення даних додатку з сервера.

При розробці запропоновано безкоштовний постачальник хмарних сервісів Firebase [10] від компанії Google. Він дозволяє налаштувати базу даних для збереження користувачів та відправляти push-повідомлення з командами в додаток, щоб контролювати його віддалено. В Firebase також є вбудована база даних, в якій зберігаються токени для відправки команд.

При першому вході в мобільний додаток генерується токен розміром в 225 символів для ідентифікації мобільного пристрою. Загальну модель генерації токен можна предстаивти так:

$$\mathbf{To} = \{\mathbf{Pa}, \mathbf{I}, \mathbf{M}\},$$

де **To** – токен; **Pa** – назва пакету додатку; **I** – id-додатку; **M** – період повторної генерації ключа.

Після того, як токен був згенерований, на мобільний пристрій можна відправляти команди для віддаленого контролю.

На віддаленому сервері додаток шифрує повідомлення для додатку і відправляє push notification на мобільний пристрій. Загальна модель процесу шифрування команд наведена нижче:

$$\mathbf{H} = \{\mathbf{Te}, \mathbf{Key}\},$$

де **H** – зашифрований текст; **Te** – текст для шифрування; **Key** – ключ шифрування розмірністю 128 біт.

Додаток за допомогою сервіса, що працює в фоновому режимі, отримує команду. Перед виконанням команди додаток перевіряє достовірність сервера; якщо сервер пройшов перевірку, команда виконується, якщо ні - ігнорується. Після цього виконується розшифрування отриманого повідомлення.

Додаток з серверу може приймати такі команди:

clear_app – повне очищення даних додатку;

block_app – блокування входу в додаток;

backup_data – створення бекапу даних додатку на сервері;

recovery_data – відновлення даних додатку з сервера.

Отримавши одну з наведених вище команд, додаток в фоновому режимі викликає модуль, який виконує дії залежно від команди. Команда clear_app виконує очищення даних додатку, видалення всіх таблиць бази даних, очищення

кешу додатку, вихід з облікового захисту. Команда block_app виконує блокування входу в додаток, тобто в момент запуску головного екрану в методі onCreate, який викликається першим в життєвому циклі Activity, викликається метод для перевірки, чи не заблокований додаток. Якщо вхід заблоковано, метод завершує роботу додатку. Команда backup_data призначена для створення бекапу даних додатку та відправки його на сервер у вигляді zip архіву, якщо це файли, і у файлі розширення json, якщо це дані з бази. Для цього користувачу потрібно викликати метод з бібліотеки та вказати, які дані та у якому вигляді він буде зберігати. Команда recovery_data призначена для відновлення даних з сервера в додаток.

Модуль автентифікації

Парольна схема автентифікації має як значні переваги (простота реалізації, відомість тощо), так і значні недоліки (підглядування, підбір слабких паролів, повторне введення тощо). Використання лише парольної схеми не дозволяє реалізувати активну автентифікацію (постійну перевірку особистості користувача на основі аспектів його взаємодії з обчислювальним пристроєм) для контролю за географічним місцезнаходженням або взаємним положенням пари користувач/мобільний пристрій.

На основі аналізу варіантів активної автентифікації [1, 2] було обрано апаратні токени, які дозволяють виконувати багатофакторну автентифікацію за парольною схемою та на основі апаратного бездротового токену, що додатково влючає в себе такі можливості:

– перевірку MAC-адреси бездротового пристрою;

– налаштування радіусу прийняття сигналу;

– налаштування часу повторної перевірки наявності токена.

Користувач відправляє автентифікаційні дані на сервер, сервер перевіряє їх і дає відповідь мобільному додатку. Як база даних в додатку використовується Firebase, в ній знаходиться таблиця з логінами та паролями, з якими зрівнюються дані, що приходять з мобільного пристрою. Якщо дані збігаються, то мобільний додаток запускає сервіс, який починає перевіряти, чи є поруч апаратний токен, наприклад, підключений через вбудований Buetooth, за яким мобільний пристрій підключається до нього і зчитує токен. Після підтвердження токена можна працювати з додатком.

В додатку при кожному вході незалежно від того, коли останній раз використовувався додаток, викликається вікно автентифікації. Автентифікаційні дані (наприклад, логін та пароль) відправляються на сервер для перевірки. У разі, якщо користувача з такими даними не існує, сервер повертає в додаток повідомлення про помилку. Після підтвердження даних користувача на мобільний пристрій відправляється повідомлення про вдалу автентифікацію, в цей момент на мобільному пристрої вмикається Bluetooth і сервіс пошуку. Він знаходить всі BLE-пристрої і отримує їх MAC-адреси. Загальна модель роботи автентифікації

$$\mathbf{R} = \{\mathbf{L}, \mathbf{Pw}, \mathbf{RT}\},$$

де **L** – логін; **Pw** – пароль; **RT** – результат, повернутий на основі перевірки апаратного токена:

$$\mathbf{RT} = \{\mathbf{MA}, \mathbf{Ti}, \mathbf{Ra}\},$$

тут **MA** – MAC-адреса апаратного токена; **Ti** – часовий інтервал, через який здійснюється опитуванння; **Ra** –відстань, на якій сигнал вважається автентичним.

Кожна MAC-адреса (**MA**) порівнюються зі збереженою при реєстрації на мобільному пристрої. При відповідності у базі даних надається доступ до роботи з додатком. Через заданий інтервал часу (**Ti**) відбувається перевірка, чи апаратний токен знаходиться на відстані, що не перевищує задану (**Ra**). Якщо користувач з апаратним токеном вийде з радіусу дії Bluetooth-зони мобільного пристрою, між ними розірветься зв’язок, і додаток завершить роботу.

Модуль захисту бази даних

Для вирішення проблеми несанкціонованого копіювання інформації з бази даних, наприклад, у випадку отримання доступу до мобільного пристрою зловмисником або шкідливим процесом, було реалізовано модуль криптографічного захисту бази даних. Шифрування здійснюється для окремих записів, а не БД в цілому. Шифрування даних здійснюється на основі блокового симетричного шифру AES з ключем 128 біт. Ключ шифрування генерується на основі пакету додатку і випадкового числа, збереженого в ресурсах додатку. При цьому кожному запису БД відповідає свій ключ. Ключі шифрування можуть зберігатися як локально, так і віддалено, і при цьому можуть бути також додатково зашифровані. Модуль реалізовано на основі системи керування базами даних Realm, яка є однією із найбільш швидких мобільних систем [13].

Висновки

Набула подальшого розвитку модель захисту Android-додатку від несанкціонованого використання, яка відрізняється своєю багаторівневою структурою з перекриттям загроз, що дозволяє організувати ефективний захист доступу до додатку на основі багатофакторної автентифікації, захисту програмного коду на основі обфускації та захисту даних додатку на основі віддаленого контролю та управління ним, а також збереження даних в захищеній базі даних.

На основі запропонованої моделі розроблено архітектуру системи захисту Android-додатку. Система містить модуль обфускації коду додатку для захисту від реверс-інженірингу, модуль автентифікації для контролю доступу до додатку, модуль віддаленого контролю додатку та управління даними, модуль захисту БД для шифрування даних додатку.

На основі запропонованої архітектури системи розроблено програмний модуль для захисту від несанкціонованого використання Android-додатку у вигляді бібліотеки, який дозволяє забезпечити захист від загроз конфіденційності, цілісності та доступності. Розроблена бібліотека дозволяє зменшити тривалість розробки Android-додатків на етапі робочого проектування при реалізації вимог щодо безпеки за рахунок зниження трудозатрат. При цьому система захисту є гнучкою, тобто при розробці додатку можна до нього інтегрувати той чи інший модуль захисту залежно від встановлених вимог.

Литература:

1.*Voitovych O.P., Hurskyi M.V., Snigovyy D.S., Kupershtein L.M.* “Monitoring tool for Android operating system”, in Scientific journal Herald of Khmelnytskyi national university 2017. Issue 3, Volume 249. 236-241 p.

2. *Tabassum, Gulista, Shikha Pandit, and Nupur Ghosh.* "Android Application Security", in Journal of Emerging Technologies and Innovative Research. Vol. 1. No. 7. 2014.

3. Kupershtein L.M., Voitovych O.P., Kaplun V. A., Prokopchuk S.O. “The database-oriented approach to data protection in Android operation system”, in Scientific journal Herald of Khmelnytskyi national university 2018, Issue 1, 18-22 p.

4. *Zhang, N., Yuan, K., Naveed, M., Zhou, X., & Wang, X.* “Leave me alone: App-level protection against runtime information gathering on android” In Security and Privacy (SP), 2015 IEEE Symposium on (pp. 915-930). IEEE.

5. *Hassanshahi, B., & Yap, R.H.* “Android Database Attacks Revisited”. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. P. 625-639.

6. *Baryshev, Y., Kaplun, V. and Neiyumina, K.* “Discretional model and method of distributed information resources access control”. In Scientific Works of Vinnytsia National Technical University. 2 (Jun. 2017).

7. *Kim, N. Y., Shim, J., Cho, S. J., Park, M., & Han, S.* “Android Application Protection against Static Reverse Engineering based on Multidexing”. J. Internet Serv. Inf. Secur., 6(4), 54-64 (2016).

8. *Dong, S., Li, M., Diao, W., Liu, X., Liu, J., Li, Z., & Zhang, K.* “Understanding Android Obfuscation Techniques: A Large-Scale Investigation in the Wild”. arXiv preprint arXiv:1801.01633 (2018).

9. http://proguard.sourceforge.net/.

10. Firebase is Google’s mobile platform that helps you quickly develop high-quality apps. URL: https://firebase.google.com/.

11. *Baryshev Yu., Kaplun V.* “Remote user authentication method for network services” in Information Technology and Computer Engineering. 2014 Vol. 2, no. 30, 1.

12. *Fridman, L., Weber, S., Greenstadt, R., & Kam, M.* “Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location” in IEEE Systems Journal, 11(2), 513-521. (2017).

13. https://realm.io.

Надійшла до редколлегії 02.06.2018

Рецензент: д-р техн. наук, проф. Бараннік В.В.

Куперштейн Леонід, канд. техн. наук, доцент кафедри захисту інформації Вінницького національного технічного університету. Наукові інтереси: застосування інтелектуальних технологій в кібербезпеці. Адреса: Україна, 21021, Вінниця, вул. Хмельницьке шосе, 95, кімн. 2424, e-mail: kupershtein@vntu.edu.ua

Войтович Олеся, канд. техн. наук, доцент кафедри захисту інформації Вінницького національного технічного університету. Наукові інтереси: кібербезпека. Адреса: Україна, 21021, Вінниця, вул. Хмельницьке шосе, 95, кімн. 2424, e-mail: voytovych.olesya@vntu.edu.ua

Остапенко-Боженова Аліна, асистент кафедри захисту інформації Вінницького національного технічного університету. Наукові інтереси: застосування криптографічного захисту інформації. Адреса: Україна, 21021, Вінниця, вул. Хмельницьке шосе, 95, кімн. 2424, e-mail: asja87@gmail.com

Прокопчук Сергій, магістр з кібербезпеки. Наукові інтереси: безпека ОС Android. Адреса: Україна, 21021, Вінниця, вул. Хмельницьке шосе, 95, кімн. 2424, e-mail: prokopchukserhii@gmail.com

Kupershtein Leonid, PhD, associated professor of the information protection department, Vinnytsya National Technical University. Scientific interests: intellectual technology applications in cyber security. Address: 95 Khmelnytske shose, Vinnytsya, 21021, Ukraine, e-mail: kupershtein@vntu.edu.ua

Voitovych Olesia, PhD, associated professor of the information protection department, Vinnytsya National Technical University. Scientific interests: cyber security. Address 95 Khmelnytske shose, Vinnytsya, 21021, Ukraine, e-mail: voytovych.olesya@vntu.edu.ua

Ostapenko-Bozhenova Alina, assistant of the information protection department, Vinnytsya National Technical University. Scientific interests – cryptography. Address 95 Khmelnytske shose, Vinnytsya, 21021, Ukraine, e-mail: asja87@gmail.com

Prokopchuk Serhii, master of cybersecurity, Vinnytsya National Technical University. Scientific interests: Android OS protection. Address 95 Khmelnytske shose, Vinnytsya, 21021, Ukraine, e-mail: prokopchukserhii@gmail.com

Prokopchuk Serhii, master of cybersecurity, Vinnytsya National Technical University. Scientific interests – Android OS protection. Address 95 Khmelnytske shose, Vinnytsya, 21021, Ukraine, e-mail: prokopchukserhii@gmail.com