РОЗДІЛ IV. ІНФОРМАЦІЙНО-КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 004.056.57
DOI: 10.25140/2411-5363-2018-2(12)-116-126

Володимир Казимир, Ігор Карпачев, Анна Усік

МОДЕЛІ СИСТЕМИ БЕЗПЕКИ ОС ANDROID

Актуальність теми дослідження. *На сьогодні набуло значного поширення використання користувачами мобільних пристроїв та додатків з метою зберігання приватної та конфіденційною інформації. Поряд з цим зростають методи та шляхи поширення шкідливого програмного забезпечення в ОС Android. Для ефективної боротьби з поширенням необхідний новий та покращений підхід до аналізу моделей прав доступу OS Android. У статті розглянуто розроблення нових моделей, які базуються на стандартному методі LSI, статистичному та категоріальному аналізі різних сервісів Google, що дозволить покращити існуючу систему безпеки.*

Постановка проблеми. *У галузі забезпечення безпечної та ефективної роботи мобільних застосувань функціональна та інформаційна безпека розглядаються як дві фундаментальні складові, що взаємодоповнюють одна одну. Одним із найефективніших способів отримання зловмисником доступ до конфіденційної інформації є обхід системи одноразової перевірки ОС Android. Одним із засобів підвищення надійності роботи є розробка моделей безперервного захисту.*

Аналіз останніх досліджень і публікацій. *Розглянуто останні публікації у відкритому доступі, включаючи статистичні дані Google Malware Project.*

Виділення недосліджених частин загальної проблеми. *Розробка та математичне обґрунтування моделей безперервного аналізу привілеїв в ОС Android.*

Постановка завдання. *Запропонувати базову модель захисту ОС Android, що базується на аналізі привілеїв.*

Виклад основного матеріалу. *Запропоновано модель безперервного аналізу програмних застосувань, що базується на аналізі дозволів методом латентно-семантичної індексації.*

Висновки відповідно до статті. *Проведено аналіз та показано вади базового підходу захисту ОС «Android Permission», за якого здійснюється одноразова перевірка роботи застосувань. Також наведено найбільш репрезентативні результати тестування наведених моделей для різних варіантів навчальної вибірки.*

Ключові слова: *безпека; система безпеки; функціональна безпека; алгоритм системи захисту; модель системи захисту ОС Android; латентно-семантична індексація; LSI.*

Рис.: 9. Табл.: 2. Бібл.: 9.

Актуальність теми дослідження. На сьогодні набуло значного поширення використання користувачами мобільних пристроїв та додатків з метою зберігання приватної та конфіденційною інформації. Поряд з цим зростають методи та шляхи поширення шкідливого програмного забезпечення в ОС Android. Для ефективної боротьби з поширенням необхідний новий та покращений підхід до аналізу моделей прав доступу OS Android. У статті розглянуто розроблення нових моделей, які базуються на стандартному методі LSI, статистичному та категоріальному аналізі різних сервісів Google, що дозволить покращити існуючу систему безпеки.

Постановка проблеми. Проблеми захисту ОС Android пов'язані з недосконалістю багатьох факторів самої системної платформи ОС, що є зручною й ефективною для використання та інтеграції нових програмних платформ, але при цьому залишає доступ зловмисникам до функціональних вузлів та конфіденційних даних. Ключовою проблемою є необхідність активної взаємодії застосувань між собою на рівні «застосування – ОС Android», що зумовлює потребу в побудові адекватних моделей прав доступу, та роботи застосувань для кожного додатку ОС Android.

Аналіз останніх джерел і публікацій. Системний аналіз роботи мобільних застосувань обов'язково включає в себе блок забезпечення функціональної безпеки, що здійснює моніторинг потенційно небезпечних умов та ідентифікує відповідні події, які можуть призвести до втрати даних, доступу до конфіденційних даних сторонніх осіб або блокування сторонніми особами належного доступу. На базовому етапі прикладного аналізу системи дозволів «Android Permission» [4] – є побудова терм-документної матриці, використання сингулярного розкладу [7] та подальший аналіз на основі латентно-семантичної індексації (LSI: Latent Semantic Indexing) [8].

Формулювання цілей статті (постановка завдання). Мета статті полягає в побудові моделей безперервного аналізу роботи програмних застосувань, що базуються на аналізі дозволів методом латентно-семантичної індексації.

Модель прав доступу. Прикладний програмний інтерфейс застосувань API є найбільш чутливим компонентом ОС Android, а його захист здійснюється через налаштування відповідної системи дозволів «Android Permission» [4]. Таким чином, критичні функції додатків у межах цієї системи мають бути включені через підтвердження запиту на доступ до AndroidManifest.xml. Але ефективність захисту, що базується на системі дозволів, має певні обмеження. При базовому підході, характерному для «Android Permission», здійснюється одноразова перевірка роботи застосування (single-point check), яку зловмисне програмне забезпечення (ПЗ) може обійти й надалі передавати конфіденційні дані за допомогою викликів API без будь-яких обмежень.

У цій статті для побудови системи безпеки ОС Android пропонуємо використовувати моделі захисту, що здійснюють безперервний аналіз ПЗ. Основою такого підходу має бути аналіз дозволів на основі латентно-семантичної індексації (LSI: Latent Semantic Indexing) [8].

Нині LSI можна вважати стандартною методикою пошуку інформаційних блоків, у якій для визначення найбільш релевантного набору файлів та текстових документів використовуються ключові елементи коду та слова. Методика спирається на обчисленні матриці, у якій рядки задаються елементами коду та словами, а стовпчики – файлами та документами. При пошуку релевантних файлів та документів матриця зменшується за допомогою методу сингулярного розкладу (SVD: Singular Value Decomposition) [7]. Так, наприклад, у межах задачі пошуку зловмисного ПЗ необхідно провести аналіз на відповідність відомим сигнатурам загроз списків дозволів у файлах XML. Запити формуються на основі списку небезпечних дозволів, після чого вектор запиту використовується для ранжування застосувань. Аналіз списків дозволів у файлах XML має певну специфіку щодо аналізу текстових файлів, тому процедура застосування методики LSI [6] має бути визначена згідно з особливостями поставленої задачі. На рис. 1 наведена схема аналізу дозволів ПЗ Android, що базується на LSI.

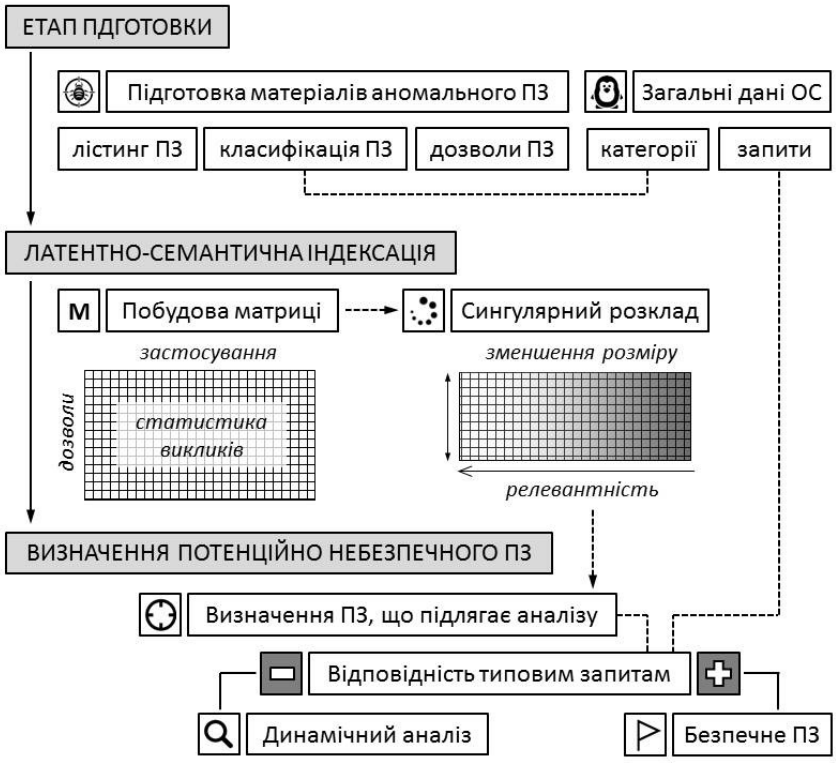

Рис. 1. Схема аналізу дозволів ПЗ Android, що базується на LSI

Алгоритм аналізу дозволів, що працює за даною схемою, складається з таких етапів:

– визначення переліку аномальних додатків для подальшого аналізу їх лістингу за методикою LSI;

– класифікація даних застосувань відповідно до переліку дозволів;

– включення дозволів у масив ключових слів LSI;

– побудова двовимірної матриці *M*, стовпчики якої складаються з набору аномальних додатків, а рядки – з переліку дозволів (таким чином, елементи матриці визначають статистику доступу ПЗ до ресурсів ОС);

– застосування щодо матриці *M* процедури SVD з метою зменшення її розміру та визначення найбільш релевантних до сигнатур зловмисного ПЗ застосувань та їх ранжування;

– визначення набору ПЗ, що підлягає подальшому аналізу та отримання для кожного з даних застосувань набору стандартних дозволів;

– якщо дозвіл на застосування не відповідає загальному набору типових дозволів із найбільш релевантних категорій, ПЗ позначається як потенційно зловмисне і підлягає динамічному аналізу.

Слід зауважити, що представлена схема є загальним алгоритмом забезпечення безпеки ОС Android. Вона не враховує те, що не всі потенційно небезпечні (ризиковані) дозволи викликаються застосуваннями. Тим більше, отримання ризикованого дозволу не характеризує застосування як однозначно зловмисний ПЗ. Але наявність ризикованого дозволу може призвести до хибного позитивного підтвердження. Щоб зменшити відсоток таких випадків, необхідно перевірити програму у віртуальному емуляторі (sandbox) і ретельно проаналізувати рівень його безпеки. Крім того, застосування методики LSI, включаючи формування терм-документної матриці та використання сингулярного розкладу (метод SVD), також не є тривіальною задачею, тому для кожною конкретної задачі необхідно визначити актуальні підходи та показати процес аналізу.

Першим етапом аналізу дозволів на основі латентно-семантичної індексації є побудова терм-документної матриці *M* та вектору запитів *q* на її основі:

$$\begin{cases} M = [k * n] \\ q = [k * l] \end{cases}, \tag{1}$$

де *q* – кількість дозволів, які відповідають термінам; *n* – кількість застосувань (що в межах цієї методики характеризуються файлами XML, які відповідають документам); *l* – кількість викликів. Отже, множина дозволів *p* та множина застосувань *a* можуть бути визначені як:

$$\begin{cases} p = [p_1; p_k] \\ a = [a_1; a_n] \end{cases}. \tag{2}$$

Сингулярний розклад зумовлює представлення терм-документної матриці *M* у наступній формі:

$$M = V * S * \left(V^{-1}\right)^T, \tag{3}$$

де V – власний вектор матриці *M*, а *S* – діагональна матриця.

Наступним етапом обирається значення *m*, *m*<*n,* щоб зменшити розмірність матриці $M(k*n)$ до матриці $M_m(k*m)$. Аналогічно S_i обирається через зменшення розмірності *S* і вектору $V_m(m*m)$. Таким чином, кожному застосуванню a_i відповідатиме вектор-рядок v_i .

Тепер отримати апроксимоване значення меншої розмірності вектору запиту q_m можна через добуток трьох матриць:

$$q_m(m*l) = q * V_m * S_m^{-1}. \tag{4}$$

Після цього можна визначити подібність елементів запиту та застосування через відповідну функцію $F(q_i, a_i)$. Відповідно до задачі нам необхідно застосувати міру подібності для дійснозначних векторів. Тому в межах роботи пропонуємо використати коефіцієнт Отіаі-Баркмана[9]:

$$F(q_i, a_i) = \frac{q_i \cdot a_i}{\| q_i \| \cdot \| a_i \|}, \quad (5)$$

де $q_i \cdot a_i$ – скалярній добуток q_i і a_i, а $\| q_i \|$ і $\| a_i \|$ – їхня потужності.

Для перевірки ефективності роботи алгоритму були використані статистичні дані сервісу Google Play [1] та зразки інформаційного ресурсу «Android Malware Genome Project» [2], ранжовані за категоріями «Розважальні сервіси» (відповідає категоріям «Games» і «Entertainment»), «Комунікаційні засоби» (відповідає категорії «Communication»), «Мультимедійні ресурси» (відповідає категоріям «Music & Audio» та «Media & Video») і «Десктоп-віджети» (відповідає категорії «Music and Video»). Базуючись на даних ресурсу [3], для вказаних категорій можна отримати результати, наведені у табл. 1.

Таблиця 1

Відсоток зловмисного ПЗ залежно від категорії застосувань

Категорія ПЗ	Відсоток ПЗ, %	Звичайне ПЗ, %	Зловмисне ПЗ, %
Розважальні сервіси	85	90	80
Комунікаційні засоби	10	6	13
Мультимедійні ресурси	3	2	4%
Десктоп-віджети	2	2	3

Зразки були поділені на зразки навчальної вибірки (80 % від повної вибірки) та зразки для тестування методики LSI (20 % від повної вибірки). Аналіз запитів на отримання дозволів дає змогу перевірити, якою мірою пов'язані категорії застосувань та категорії запитів та як це співвідноситься з навчальною вибіркою. У цьому прикладі, у зв'язку зі специфікою матеріалу, представленого для навчання системи захисту, співвіднесення відбувалося саме зі зразками зловмисного ПЗ. Тому до схеми, представленій на рис. 1, були внесені відповідні зміни.

Усі потенційно небезпечні запити на дозволи були віднесені, відповідно до стандартної класифікації, до однієї з трьох категорій (рис. 2): конфіденційність (privacy), трафік (billing), робота ОС та ПЗ (system operation).

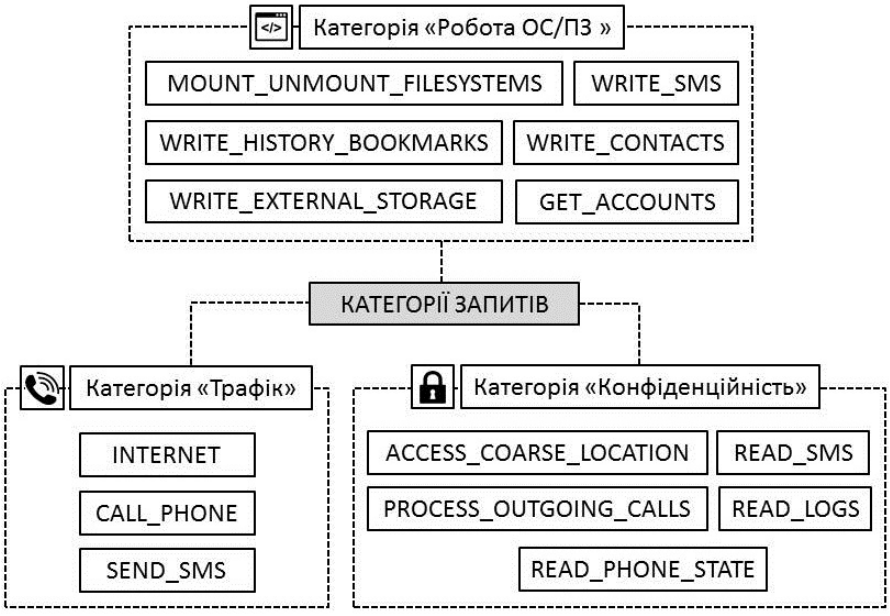

Рис. 2. Категорії запитів із високим ризиком ОС Android

Найбільш репрезентативні результати тестування запропонованої моделі для даних трьох категорій представлені у табл. 2.

Таблиця 2

Результати тестування LSI-моделі аналізу ПЗ ОС Android, %

Розважальні сервіси	Комунікаційні засоби	Мультимедійні ресурси	Десктоп-віджети	Відповідність результатів
88	6	3	3	84
92	5	2	1	93
94	3	2	1	98

Тестування даної моделі показує, що ефективність статичного аналізу залежить від процентного співвідношення найбільш актуальних категорій (у даному випадку категорії «Розважальні сервіси») у навчальній виборці. Але, слід крім того зауважити, що в будь якому випадку залишається певний процент похибок другого роду, для відслідковування яких необхідно використовувати динамічний аналіз.

Модель роботи застосувань. Як показує тестування алгоритму аналізу дозволів на основі латентно-семантичної індексації [5], на ефективність статичного аналізу значною мірою впливає адекватність моделювання роботи застосувань та повнота відповідних статистичних даних, пов'язаних з використанням зловмисним ПЗ окремих категорій запитів. Для розробки цілісного методу необхідно побудувати узагальнену модель роботи застосувань, залучити до розгляду статистичний аналіз використання запитів та побудувати класифікатор потенційних загроз.

У загальному виді результати аналізу застосувань на потенційно небезпечне ПЗ можна поділити на чотири види найбільш типових показників (рис. 3):

– істинно позитивні, кількість яких визначається як *TP* (true positive);

– істинно негативні, кількість яких визначається як *TN* (true negative);

– хибно позитивні, або помилки другого роду, кількість яких визначається як *FP* (false positive);

– хибно негативні, або помилки першого роду, кількість яких визначається як *FN* (false negative);

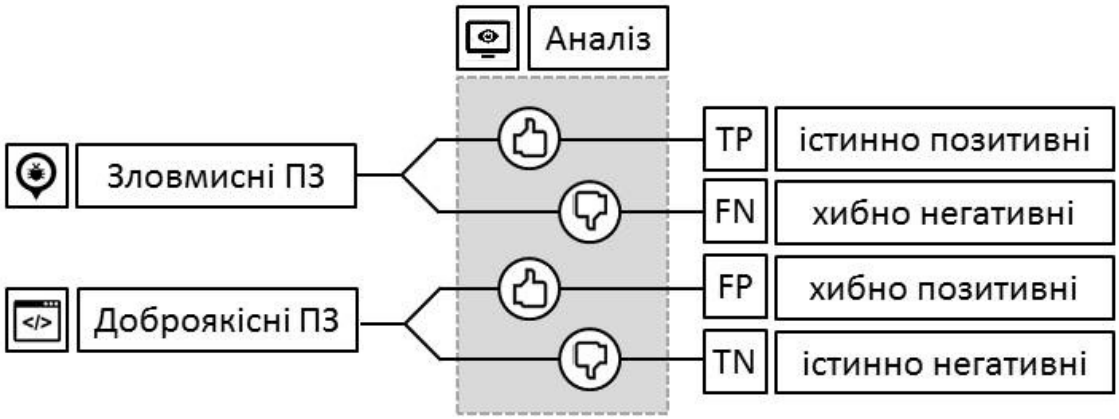

Рис. 3. Розподіл результатів аналізу за видами показників

На основі даних величин можна визначити такі показники, як кількість істинно позитивних рішень *TPR* (true positive rate), кількість хибно позитивних рішень *FPR* (false positive rate), а також точність передбачення позитивних значень *PPV* (positive predictive value) та точність передбачення негативних значень *FPV* (positive predictive value):

$$\begin{cases} TPR = \dfrac{TP}{TP + FN} \\ FPR = \dfrac{FP}{TN + FP} \\ TNR = \dfrac{TN}{TN + FP} \\ FNR = \dfrac{FN}{TP + FN} \end{cases}, \tag{6}$$

$$\begin{cases} PPV = \dfrac{TP}{TP + FP} \\ FPV = \dfrac{TN}{TN + FN} \end{cases}. \tag{7}$$

Одним із найбільш ефективних методів перевірки результатів роботи та вдосконалення розробленої моделі аналізу застосувань ОС Android є Баєсів класифікатор, який відноситься до ймовірнісних класифікаторів та характеризується простим і компактним алгоритмом, що використовує мінімум апаратних ресурсів ОС, але при цьому характеризується високою точністю. Робота з цим класифікатором включає у себе фази навчання та тестування. На етапі навчання модель класифікатора отримує на вхід навчальну вибірку зразків доброякісного та зловмисного ПЗ для ОС Android. Надалі, під час тестування або роботи, модель виявляє належність застосування до зловмисного ПЗ, використовуючи дані, отримані під час навчання.

Для проведення ефективної класифікації необхідно визначити певну статистику отримання запитів на дозволи та API-викликів, приклад якої наведено на рис. 4.

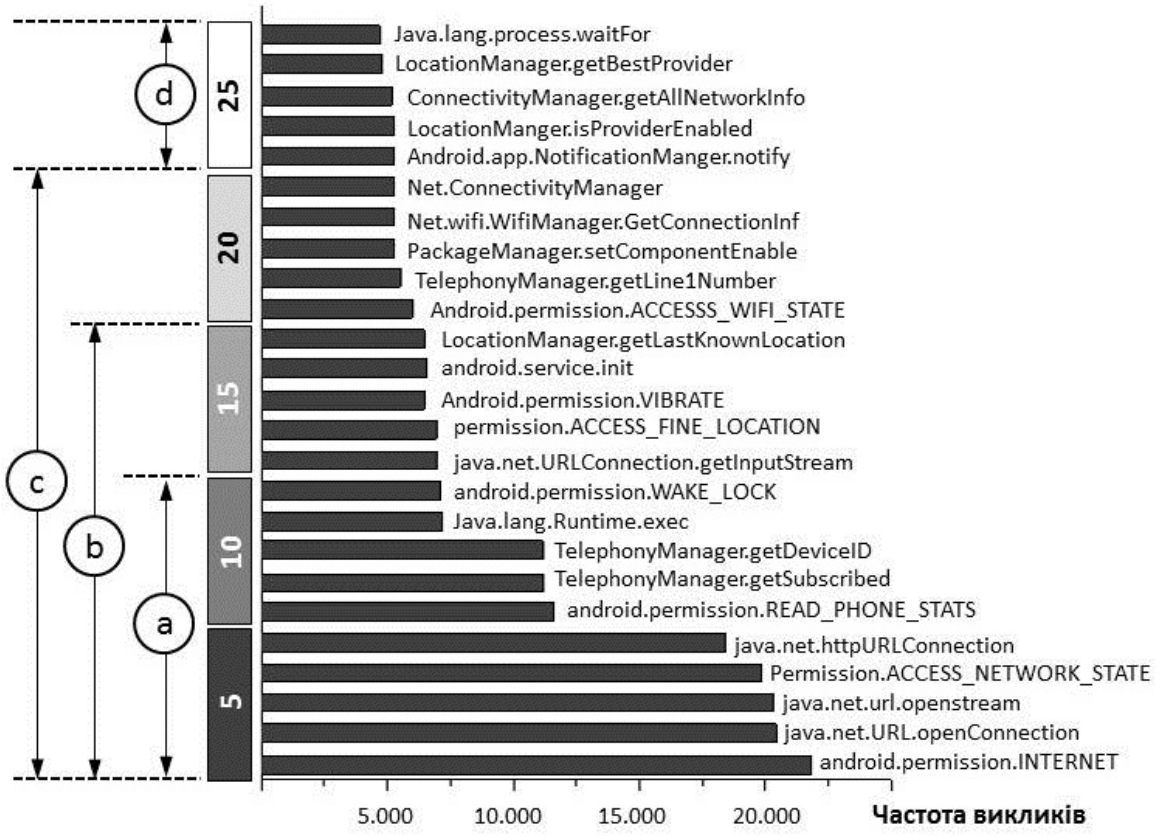

Рис. 4. Статистика отримання запитів на дозволи ОС Android

У цьому випадку запити було поділено на десять тих, що найбільш активно застосовуються (група «a»), п'ятнадцять тих, що найбільш активно застосовуються (група «b»), двадцять тих, що найбільш активно застосовуються (група «c»), п'ять тих, що найменш активно застосовуються (група «d»). Для цих груп запитів були визначені відповідні показники: частота похибок та точність (відповідно, похибок першого та другого роду), відсотковий склад TNR, TPR, FNR, FPR, а також показники точності FPV і PPV (рис. 5–9).

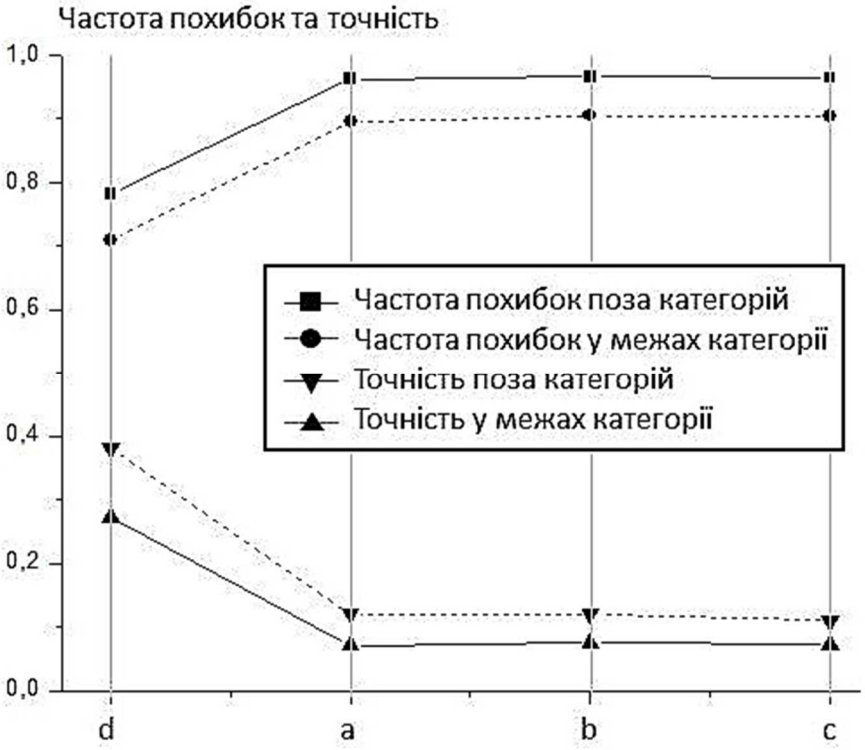

Рис. 5. Частота похибок та точність аналізу

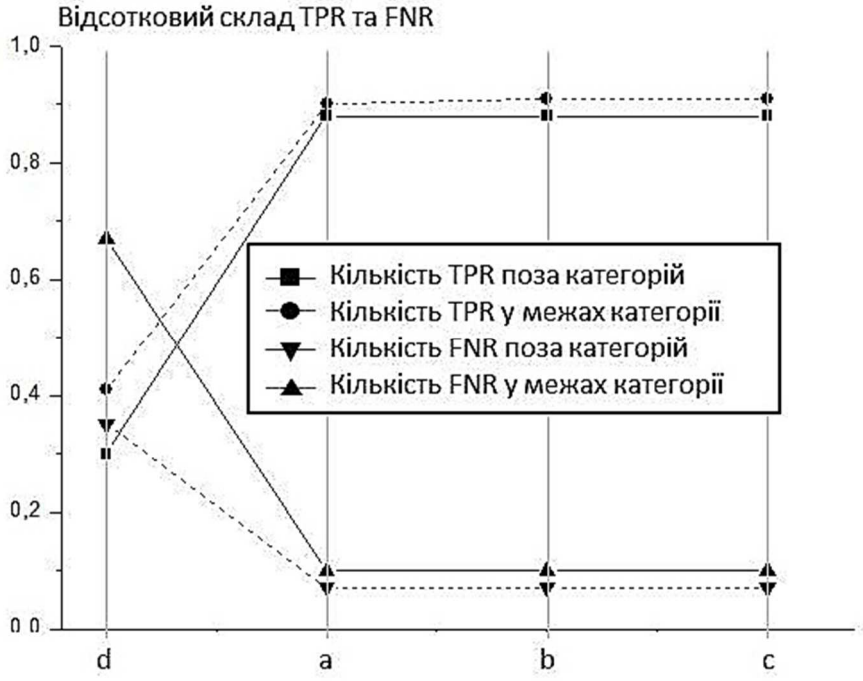

Рис. 6. Відсотковий склад TPR та FNR

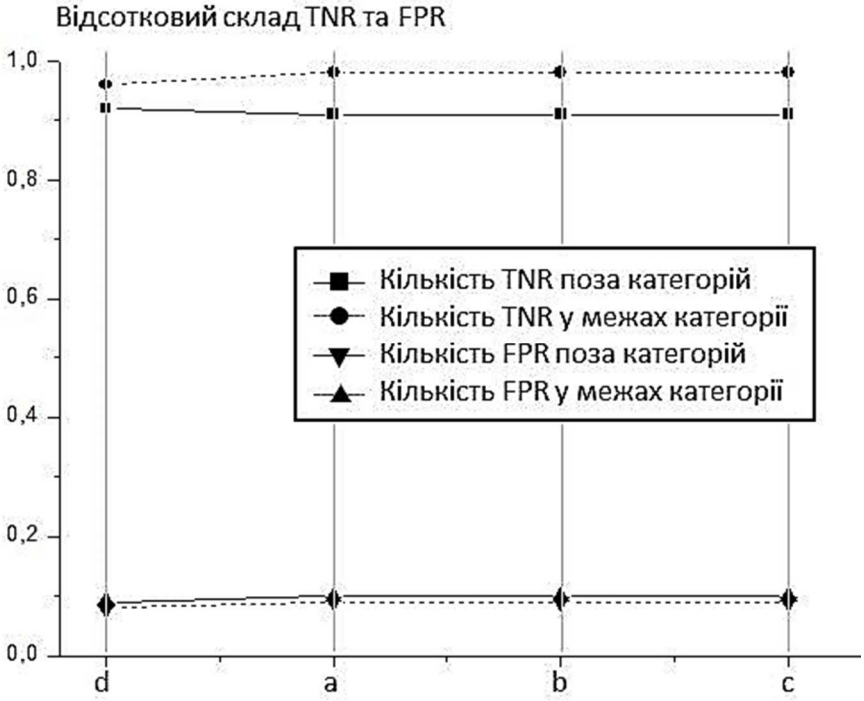

Рис. 7. Відсотковий склад TNR та FPR

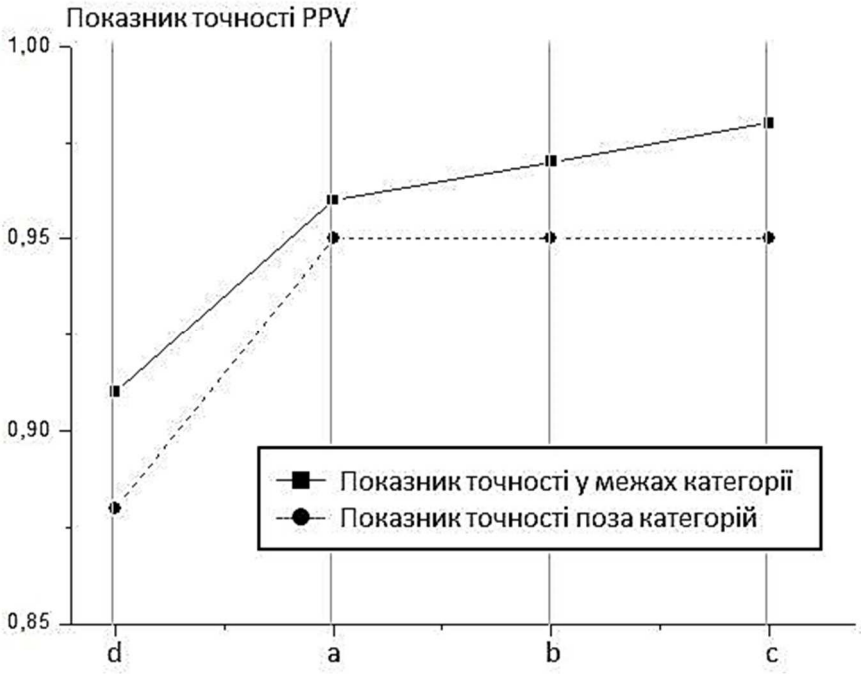

Рис. 8. Показник точності PPV

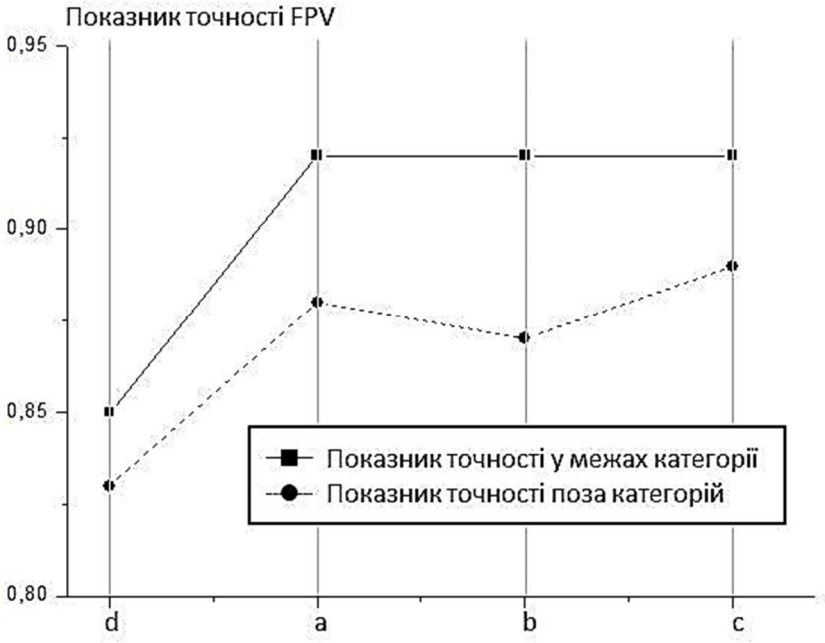

Рис. 9. Показник точності FPV

Аналіз представлених графіків показує, що зі збільшенням групи збільшується точність аналізу та, відповідно, зменшується кількість похибок. Особливо різниця очевидна при порівнянні груп «a» і «d». У цьому випадку різниця пов'язана не лише з розміром групи, а й з її актуальністю. Крім того, слід зауважити, що під час тестування розроблених моделей на зразках, що перебувають поза межами категорій, які використовувалися при навчанні, зменшується точність аналізу та, відповідно, збільшується кількість похибок.

Висновки відповідно до статті. Запропоновані моделі при постійній перевірці дозволів, які використовуються застосуванням, значно зменшують негативний потенціал зловмисника. Але для забезпечення ефективної безпеки ОС Android необхідно визначити всі способи, за допомогою яких зловмисне ПЗ може асоціювати себе з завданням чи іншим застосуванням, що надає надмірні привілеї і становитиме потенційну загрозу. Це можливо лише через експериментальне вивчення способів управління ОС Android завданнями, що реалізується через дослідження процесу роботи системи за умови встановлення всіх можливих комбінацій перемикачів налаштувань платформи, які можуть вплинути на статус завдання. При цьому має здійснюватися аналіз додаткових привілеїв, які можуть бути отримані застосуванням, коли воно приєднується до завдання.

Список використаних джерел

1. Google Play [Електронний ресурс]. – Режим доступу : https://play.google.com.

2. Android Malware Genome Project [Електронний ресурс]. – Режим доступу : http://www.malgenomeproject.org.

3. Malware mobile applications [Електронний ресурс]. – Режим доступу : https://www.gnu.org/proprietary/malware-mobiles.ru.html.

4. Manifest.permission [Електронний ресурс]. – Режим доступу : http://developer.-ndroid.com/reference-/android/Manifest.permission.html.

5. Permissions reference [Електронний ресурс]. – Режим доступу : https://developers.facebook.-com/docs/authentication/permissions.

6. Latent Semantic Indexing [Електронний ресурс]. – Режим доступу : https://nlp.stanford.edu/IR-book/html/htmledition/latent-semantic-indexing-1.html.

7. Singular Value Decomposition (SVD) [Електронний ресурс]. – Режим доступу : https://www.cs.cmu.edu/ ~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf.

8. Latent Semantic Indexing [Електронний ресурс]. – Режим доступу : http://www.cs.haifa.ac.il/~rita/uml_course/lectures/LSI.pdf.

9. Cosine Similarity [Електронний ресурс]. – Режим доступу : https://studbooks.net/2244345/matematika_himiya_fizika/kosinusnaya_mera.

References

1. Google Play. (2016). *play.google*. Retrieved from https://play.google.com [in English].

2. Android Malware Genome Project. (2012). *www.malgenomeproject.org*. Retrieved from http://www.malgenomeproject.org/ [in English].

3. Malware mobile applications. (2018). *www.gnu.org*. Retrieved from https://www.gnu.org/proprietary/malware-mobiles.ru.html [in English].

4. Manifest.permission. (2016). *developer.-ndroid.com*. Retrieved from http://developer.-ndroid.com/reference-/android/Manifest.permission.html [in English].

5. Permissions reference. (2016). *developers.facebook.-com*. Retrieved from https://developers.facebook.-com/docs/authentication/permissions/ [in English].

6. Latent Semantic Indexing. (2016). *nlp.stanford.edu*. Retrieved from https://nlp.stanford.edu/IR-book/html/htmledition/latent-semantic-indexing-1.html [in English].

7. Singular Value Decomposition. (2016). *www.cs.cmu.edu*. Retrieved from https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf [in English].

8. Latent Semantic Indexing. (2011). *www.cs.haifa*. Retrieved from http://www.cs.haifa.ac.il/~rita/uml_course/lectures/LSI.pdf [in English].

9. Cosine Similarity. (2018). *studbooks.net*. Retrieved from https://studbooks.net/2244345/ matematika_himiya_fizika/kosinusnaya_mera [in English].

UDC 004.056.57

Volodymyr Kazymyr, Igor Karpachev, Anna Usik

MODELS OF THE SYSTEM OF SECUTIRY OS ANDROID

Urgency of the research. *Storing user's private and confidential information have been widely used today by users of mobile applications. Methods and ways of spreading malware in the Android operating system are growing at the same time. A new and improved approach to Android OS access model analysis is needed for effective protection. The article deals with the development of new models based on the standard LSI method, statistical and categorical analysis of various Google services, which will lead to improvement in the existing security system.*

Target setting. *In the area of ensuring the safe and efficient operation of mobile applications, functional and information security are considered as two fundamental components complementing each other. One of the most effective ways for an intruder to access confidential information is bypassing the single-point OS check of the system. One of the tools to improve the reliability of work is the development of continuous protection models.*

Actual scientific researches and issues analysis. *Recent open publications were considered, including statistical data from Google Malware Project.*

Uninvestigated parts of general matters defining. *The development and mathematical justification of the models of continuous analysis of privileges in Android OS.*

The research objective. *Suggest a basic Android OS protection model based on the analysis of privileges.*

The statement of basic materials. *Proposed a model of continuous analysis of software applications, based on the analysis of permissions by latent semantic indexation.*

Conclusions. *The analysis and the defects of the basic approach of protection of OS "Android Permission" in which one-time verification of application work is carried out is shown. There are also provided the most representative results of the testing of the given models for various variants of the training sample.*

Keywords: *security; security system; functional security; security system algorithm; Android OC protection model; latent semantic indexing; LSI.*

Fig.: 9. Table: 2. References: 9.

УДК 004.056.57

Владимир Казимир, Игорь Карпачев, Анна Усик

МОДЕЛИ СИСТЕМЫ БЕЗОПАСНОСТИ ОС ANDROID

Актуальность темы исследования. *На сегодняшний день получило широкое распространение использование пользователями мобильных устройств и приложений с целью хранения частной и конфиденциальной информации. Наряду с этим растут методы и пути распространения вредоносного программного обеспечения в ОС Android. Для эффективной борьбы с распространением необходим новый и улучшенный подход к анализу моделей прав доступа OS Android. В статье*

рассмотрена разработка новых моделей которые базируются на стандартном методе LSI, статистическом и категориальном анализ различных сервисов Google, что позволит улучшить существующую систему безопасности.

Постановка проблемы. *В области обеспечения безопасной и эффективной работы мобильных приложений функциональная и информационная безопасность рассматриваются как две фундаментальные составляющие, взаимодополняют друг друга. Одним из самых эффективных способов получения злоумышленников доступ к конфиденциальной информации является обход системы однократной проверки ОС Android. Одним из средств повышения надежности работы является разработка моделей непрерывного защиты.*

Анализ последних исследований и публикаций. *Рассмотрены последние публикации в открытом доступе, включая статистические данные Google Malware Project.*

Выделение неисследованных частей общей проблемы. *Разработка и математическое обоснование моделей непрерывного анализа привилегий в ОС Android.*

Постановка задачи. *Предложить базовую модель защиты ОС Android основанный на анализе привилегий.*

Изложение основного материала. *Предлагается модель непрерывного анализа программных приложений, основанный на анализе разрешений методом латентно-семантической индексации.*

Выводы в соответствии со статьей. *Проведен анализ и показаны недостатки базового подхода защиты ОС «Android Permission» при котором осуществляется одноразовая проверка работы приложений. Также приведены наиболее репрезентативные результаты тестирования приведенных моделей для различных вариантов обучающей выборки.*

Ключевые слова: *безопасность; система безопасности; функциональная безопасность; алгоритм систем безопасности; модель системы защиты ОС Android; латентно-семантическая индексация; программное обеспечение; LSI.*

Рис.: 9. Табл.: 2. Библ.: 9.

Казимир Володимир Вікторович – доктор технічних наук, професор, проректор з наукової роботи, Чернігівський національний технологічний університет (вул. Шевченка 95, м. Чернігів, 14035, Україна).
Казимир Владимир Викторович – доктор технических наук, профессор, проректор по научной работе, Черниговский национальный технологический университет (ул. Шевченка 95, г. Чернигов, 14035, Украина).
Kazymyr Volodymyr – Doctor of Technical Sciences, Professor, Vice-rector for scientific work, Chernihiv National University of Technology (95 Shevchenka Str., 14035 Chernihiv, Ukraine).
E-mail: vvkazymyr@gmail.com
ORCID: https://orcid.org/0000-0001-8163-1119
Scopus Author ID: 56644727300

Карпачев Ігор Ігорович – аспірант кафедри інформаційних та комп'ютерних систем, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14035, Україна).
Карпачев Игорь Игоревич – аспирант кафедры информационных и компьютерных систем, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14035, Украина).
Karpachev Igor – Phd student, Department of Informational and Computer Systems, Chernihiv National University of Technology (95 Shevchenka Str., 14035 Chernihiv, Ukraine).
E-mail: benchakalaka@gmail.com
ORCID: https://orcid.org/0000-0003-1910-3264
ResearcherID: R-3626-2016

Усік Анна Миколаївна – аспірант кафедри інформаційних та комп'ютерних систем, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14035, Україна).
Усик Анна Николаевна – аспирант кафедры информационных и компьютерных систем, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14035, Украина).
Usik Anna – Phd student, Department of Informational and Computer Systems, Chernihiv National University of Technology (95 Shevchenka Str., 14035 Chernihiv, Ukraine).
E-mail: usik18@gmail.com
ORCID: https://orcid.org/0000-0003-4965-6863