УДК 004.056.53:007.77
DOI: 10.31673/2409-7292.2024.040012

Фединишин Т. О., Партика О. О.

АНАЛІЗ ЗАКОДОВАНИХ ОБЛІКОВИХ ДАНИХ В ANDROID ДОДАТКАХ

У даній роботі представлено результати масштабного дослідження поширеності закодованих секретів, таких як API-ключі та облікові дані, у 6165 Android-додатках, отриманих із Google Play Store. Використовуючи інструменти MobSF і Trufflehog, було виявлено, що значна кількість додатків містить в коді чутливі дані, які створюють серйозні ризики для безпеки. Зокрема, виявлено закодовані облікові дані хмарних провайдерів, таких як Amazon Web Services і Google Cloud Platform, що можуть призводити до несанкціонованого доступу, компрометації конфіденційної інформації, зловживання ресурсами та фінансових втрат. Аналіз показав, що категорія "Здоров'я та фітнес" має найвищу частоту вбудованих секретів, за нею йдуть "Новини та журнали", "Музика та аудіо", "Фотографія" та "Соціальні мережі". У додатках з категорії "Комунікації", які обробляють чутливу інформацію, такі як особисті повідомлення та мультимедіа, закодовані облікові дані створюють додаткові ризики, зокрема перехоплення даних, компрометацію цілісності комунікацій та атаки на відмову в обслуговуванні. В результаті дослідження також виявлено проблеми в управлінні й ротації секретів, що перешкоджає впровадженню розробниками найкращих практик безпеки. Це свідчить про необхідність посилення автоматизації процесів виявлення та оновлення секретів у мобільних додатках. Отримані результати також підкреслюють потребу у впровадженні централізованих рішень для управління конфіденційною інформацією, таких як системи управління секретами. Для зменшення ризиків пропонується інтеграція рішень типу DevSecOps, що забезпечить безпеку на всіх етапах розробки програмного забезпечення. Крім того, дослідження акцентує увагу на важливості дотримання стандартів безпеки, таких як OWASP Mobile Top 10, для мінімізації вразливостей у мобільних додатках.

Ключові слова: безпека мобільних додатків, android security, data privacy, static analysis, improper credentials usage, OWASP Mobile, MobSF, Trufflehog.

Вступ

Мобільні пристрої, особливо смартфони, постійно розвиваються і зараз є найпоширенішим засобом зв'язку між людьми через телефонні дзвінки або мережу Інтернет. Крім комунікації, такі дії, як обробка документів, перегляд потокового відео, електронна пошта та ігри, також можуть легко виконуватись на смартфонах, що робить їх більш універсальними та необхідними, ніж будь-коли. За даними [1], очікується, що смартфони залишатимуться домінуючими пристроями для роботи в мережі Інтернет, особливо із впровадженням стандартів зв'язку 5G та 6G у майбутньому.

Смартфони та численні додатки, які підтримують різні функції, стали невід'ємною частиною сучасного життя. Люди дедалі більше покладаються на мобільні додатки для широкого спектра щоденних завдань, використовуючи їх кілька разів на день. Apple App Store [2] та Google Play [3] разом пропонують понад вісім мільйонів додатків. Проте, походження та безпека цих додатків не завжди можуть бути гарантовані. Незважаючи на процедури перевірки, що застосовуються Apple та Google перед розміщенням додатків у відповідних магазинах, багато мобільних додатків все ж мають вразливості та становлять значні ризики для безпеки. Мобільні операційні системи не мають достатніх інструментів для виявлення шкідливого програмного забезпечення, яке може скомпрометувати персональні дані. Як наслідок, мобільні додатки становлять потенційні загрози для безпеки, оскільки їхні вразливості можуть бути використані зловмисниками для несанкціонованого доступу до ресурсів пристрою, включно з чутливою інформацією користувача [4].

Додатки є важливим елементом мобільної екосистеми, що потребує подальших досліджень для розробки ефективних методів та інструментів, спрямованих на зниження ризиків, пов'язаних з їх використанням.

Мета та завдання дослідження

Метою цього дослідження є проведення масштабного статичного аналізу понад 6000 Android-додатків з Google Play для виявлення закодованої чутливої інформації, такої як API-ключі та облікові дані, за допомогою MobSF та Trufflehog. Виявлення та аналіз секретів є підґрунтям для знаходження потенційних вразливостей та надання рекомендацій для покращення управління конфіденційними даними в додатках для ОС Android [5].

Огляд останніх досліджень

OWASP Mobile Top 10. Відкритий проект з безпеки веб застосунків (OWASP) є некомерційною організацією, що присвячена підвищенню безпеки програмного забезпечення через глобальну співпрацю та участь зацікавлених сторін. OWASP надає платформу для лідерів у сфері промисловості, академічних установ і урядів, де обговорюються та просуваються найкращі практики у сфері комп'ютерних обчислень. Серед ініціатив проекту OWASP є створення та щорічне оновлення списку, що висвітлює 10 основних ризиків для мобільних додатків. Цей список ідентифікує ключові загрози безпеки, включаючи ризики для даних, внутрішніх і зовнішніх комунікацій пристроїв, а також інші вразливості в мобільних додатках.

Список поширених кіберзагроз для мобільних додатків та їхні описи, згідно з OWASP Mobile Top 10 за 2024 рік:

1. Неправильне використання облікових даних - загрози, пов'язані із використанням вбудованих облікових даних та неправильне використання облікових даних у мобільних додатках, на яке можуть здійснюватися автоматизовані атаки за допомогою загальнодоступних або спеціально створених інструментів.
2. Недостатня безпека ланцюга постачання - стосується незабезпечення захисту сторонніх компонентів, сервісів або бібліотек, інтегрованих у мобільні додатки, що може призвести до виникнення вразливостей і підвищення ризику компрометації в усьому ланцюгу постачання програмного забезпечення.
3. Незахищена аутентифікація/авторизація - загрози, пов'язані із експлуатацією вразливостей аутентифікації та авторизації.
4. Недостатня валідація вхідних/вихідних даних - недостатня перевірка та очищення даних із зовнішніх джерел, таких як введення користувача або мережеві дані, у мобільному додатку.
5. Незахищена комунікація - незабезпечення належного захисту передачі конфіденційних даних між мобільним додатком та зовнішніми об'єктами, такими як сервери або інші пристрої, що може призвести до потенційного перехоплення, підробки або розкриття інформації.
6. Недостатні засоби контролю конфіденційності - стосується недостатнього захисту даних користувачів у мобільному додатку, що може призвести до несанкціонованого доступу, розкриття або неправильного використання конфіденційної інформації, такої як місцезнаходження, контакти чи інші приватні дані.
7. Недостатній захист бінарного коду - відсутність належного захисту від реверс-інжинірингу або втручання в бінарний код мобільного додатку, що може дозволити зловмисникам модифікувати, експлуатувати або шкідливо поширювати додаток.
8. Неправильна конфігурація безпеки - стосується неналежного налаштування параметрів безпеки, дозволів і контролів, що може призвести до виникнення вразливостей і несанкціонованого доступу.
9. Небезпечне зберігання даних - стосується недостатнього захисту конфіденційних даних, збережених на мобільному пристрої, що може призвести до несанкціонованого доступу, витоку даних або розкриття інформації, якщо механізми зберігання не забезпечені належним захистом.
10. Недостатня криптографія – використання слабких алгоритмів шифрування, що може підірвати конфіденційність, цілісність і автентичність конфіденційної інформації [6].

Аналіз літератури

Вразливості мобільних додатків, такі як помилки аутентифікації та авторизації, витік даних і пов'язані з ними ризики безпеки - від вразливостей API, слабкої авторизації та аутентифікації, ін'єкцій на стороні клієнта, поганої безпеки на стороні сервера, небезпечного зберігання та передачі даних, неналежного управління сеансами до використання недосконалих або небезпечних алгоритмів шифрування - становлять значні загрози [7]. У

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

сучасному цифровому середовищі користувачі часто довіряють своїм пристроям конфіденційну інформацію, включаючи фінансові та медичні дані, що є серйозним викликом для мобільних розробників і провайдерів. Кіберзлочинці часто націлюються на дані, оброблювані мобільними додатками та пристроями [8]. Крім того, збільшення використання мобільних додатків для Інтернету речей (IoT) посилило загрозу атак типу “wormhole” [9–13].

Тестування за еталоном NowSecure [14] показало, що 85% з перевірених додатків містили один або більше ризиків безпеки. У понад 50% проаналізованих додатків виявлено вразливості, які компрометували захист даних під час передачі. Крім того, приблизно одна третина протестованих додатків мала проблеми, пов'язані з їхнім вихідним кодом. Особливо вразливими до помилок у коді виявилися додатки для Android, що може піддати їх реверс-інжинірингу та іншим потенційним загрозам.

Матеріали та методи

Для цього дослідження було зібрано комплексний набір даних, що складається з 6165 файлів APK, отриманих з Google Play Store. Цей набір даних був ретельно підібраний для представлення різноманітного спектра додатків з різних категорій та рівнів популярності, що забезпечує широкий і репрезентативний зразок екосистеми мобільних додатків. Зібрані файли APK були піддані статичному аналізу за допомогою Mobile Security Framework (MobSF) [15], визнаного інструменту для оцінки безпеки мобільних додатків. MobSF було використано для проведення статичного аналізу кожного APK файлу, зосереджуючи увагу на виявленні потенційно конфіденційної інформації, вбудованої у код додатка. Процес аналізу включав обробку розділу “secrets” для кожного APK, де перераховані можливі вбудовані секрети, такі як ключі API, токени аутентифікації та інші облікові дані. Після статичного аналізу виявлені у розділі “secrets” рядки піддавались подальшому аналізу за допомогою Trufflehog [16], інструменту, який спеціалізується на виявленні секретів у кодових базах. Trufflehog використовувався для перевірки достовірності виявлених секретів і для розрізнення справжніх секретів від хибних спрацьовувань. Цей вторинний аналіз мав на меті надати більш точну оцінку потенційних ризиків безпеки, пов’язаних із вбудованими обліковими даними в APK. Ця методологічна структура сприяла суворому вивченню практик управління обліковими даними у мобільних додатках і допомогла отримати висновки щодо наслідків для безпеки у разі розкриття секретної інформації у додатках для Android.

Вибірка зразків

Станом на 2024 рік у Google Play Store розміщено понад 3,5 мільйона додатків [17]. Проведення комплексної оцінки всіх цих додатків вимагало б значних серверних ресурсів і значного часу. Відтак, автори даного дослідження зосередились на аналізі підмножини найпопулярніших додатків. Початковий етап включав оцінку популярності мобільних додатків. Дані про завантаження додатків, сегментовані за країнами та категоріями, були отримані з SimilarWeb [18]. На момент дослідження було ідентифіковано 59,108 унікальних додатків у 57 категоріях і 96 країнах. Згодом для аналізу були завантажені файли APK цих додатків. Через відсутність прямого методу завантаження файлів APK з Google Play , використовувались сторонні сервіси, такі як APKCombo. Через обмеження в ресурсах зберігання та обчислювальних потужностях, а також наявність файлів APK на сторонніх сервісах, вдалося завантажити та проаналізувати 6,165 файлів APK.

Кількість завантажених додатків для кожної категорії Google Play наведена в таблиці 1, яка включає лише 22 категорії, де було завантажено щонайменше 15 файлів APK.

Статичний аналіз

Статичний аналіз за допомогою MobSF є важливим методом для оцінки безпеки мобільних додатків. MobSF - це універсальний, інструмент із відкритим вихідним кодом, розроблений для аналізу як Android, так і iOS додатків, з метою виявлення потенційних вразливостей безпеки. Процес починається з подачі файлу APK (Android Package Kit) до MobSF. Через особливості розробки мобільних додатків APK-файли необхідно розібрати і декомпілювати для ретельного аналізу. MobSF використовує такі інструменти, як APKTool та

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

jadx, для декомпіляції APK, перетворюючи скомпільований байткод у більш доступний для читання людиною формат. Цей етап є ключовим, оскільки він розбиває додаток на його складові частини, включаючи файл маніфесту, ресурси та код. Після декомпіляції APK MobSF здійснює аналіз коду додатку. Аналіз зосереджується на кількох ключових аспектах: виявлення витоку конфіденційних даних, ідентифікація небезпечних практик кодування та виявлення відомих вразливостей. MobSF сканує код на наявність вбудованих секретів, таких як ключі API, облікові дані та токени, що можуть становити значні ризики безпеки, якщо будуть розкриті. Крім того, інструмент оцінює використання криптографічних алгоритмів та інших заходів безпеки, щоб переконатися, що вони реалізовані правильно.

Таблиця 1

Розподіл досліджуваних додатків по категоріях

Ідентифікатор категорії додатку	Кількість завантажених додатків
SPORTS	580
PARENTING	566
PHOTOGRAPHY	452
NEWS_AND_MAGAZINES	443
SOCIAL	438
TOOLS	437
ENTERTAINMENT	417
PRODUCTIVITY	352
COMMUNICATION	312
AUTO_AND_VEHICLES	280
PERSONALIZATION	274
BOOKS_AND_REFERENCE	246
DATING	202
MUSIC_AND_AUDIO	187
MAPS_AND_NAVIGATION	173
ART_AND_DESIGN	139
BUSINESS	123
BEAUTY	102
EDUCATION	69
MEDICAL	68
HEALTH_AND_FITNESS	28
LIFESTYLE	17

Виявлення секретів за допомогою Trufflehog

Trufflehog - це спеціалізований інструмент, призначений для виявлення конфіденційної інформації, такої як API ключі, облікові дані та токени, у кодових базах. Спочатку розроблений для Git-репозиторіїв, Trufflehog показав свою ефективність [16] у різних контекстах безпеки, включаючи статичний аналіз мобільних додатків. Основна функціональність Trufflehog ґрунтується на двох основних техніках: шаблонному співставленні та аналізі на основі ентропії. Інструмент використовує набір попередньо визначених регулярних виразів та евристик для виявлення шаблонів, що зазвичай асоціюються із секретами. Ці шаблони включають різноманітні облікові дані та токени, які часто вбудовані безпосередньо в код додатку. Використовуючи ці шаблони, Trufflehog здатний виявляти широкий спектр конфіденційної інформації, яка могла б залишитися непоміченою. Окрім шаблонного співставлення, Trufflehog застосовує аналіз на основі ентропії для оцінки випадковості деяких рядків у коді. Рядки з високими значеннями ентропії вказують на потенційні секрети, оскільки вони менш імовірні для випадкового виникнення у несекретних даних. Цей метод покращує здатність Trufflehog виявляти секрети, які можуть не відповідати загальноприйнятим шаблонам, але все ж становлять ризик розголошення. Для кожного

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

виявленого секрету Trufflehog надає детальну інформацію про його розташування в коді, що полегшує цільові заходи для усунення загрози.

Результати

У цьому дослідженні було проведено комплексний аналіз безпеки 6165 Android-додатків у 22 категоріях. Аналіз здійснювався за допомогою комбінації інструментів, зокрема MobSF і Trufflehog, для виявлення неправильного використання облікових даних відповідно до структури OWASP Mobile Top 10. Результати аналізу вразливостей надають інформацію про стан безпеки вибраних додатків.

Вбудовані секрети для роботи зі сторонніми сервісами

Під час дослідження виявлено облікові дані для різних сторонніх сервісів та частота їх наявності для кожного типу облікових даних. На рис. 1 показано кількість виявлених секретів для кожного сервісу. Як видно, API ключ для сервісу Twitter є найбільш поширеними вбудованими даними в коді додатків.

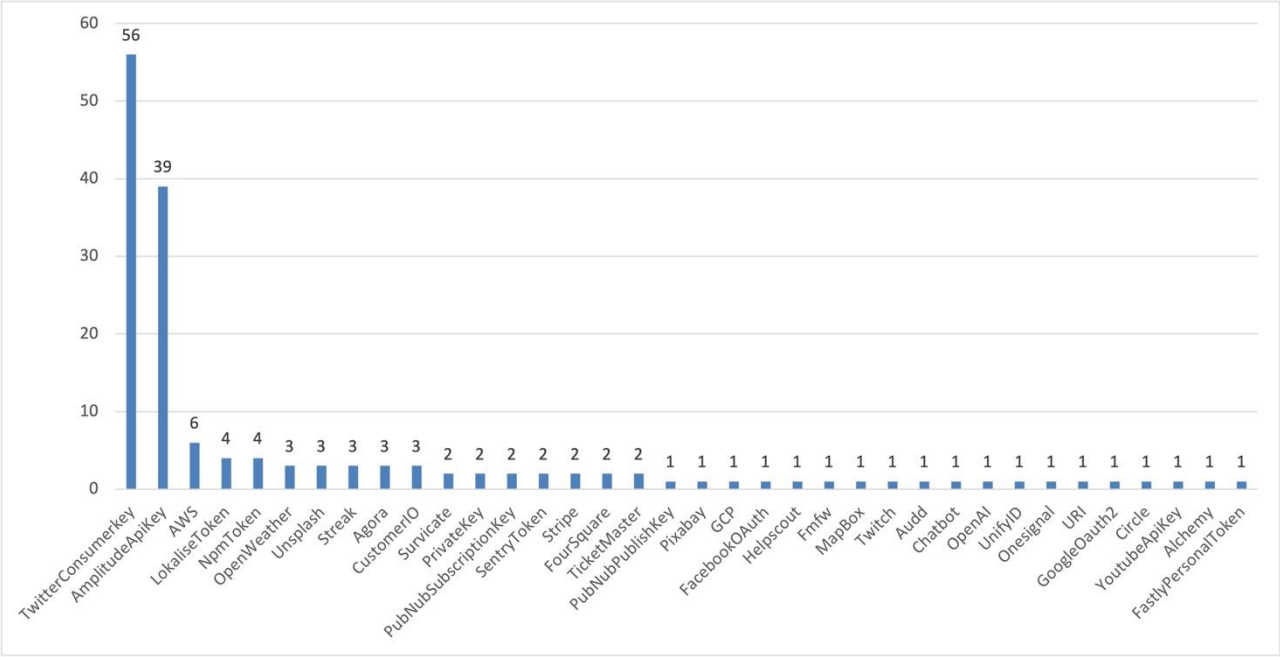

Рис. 1. Кількість виявлених секретів для кожного сервісу

Результати дослідження показують, що деякі додатки містять вбудовані секрети постачальників хмарних сервісів. Наявність секретів AWS (Amazon Web Services) та GCP (Google Cloud Platform) у коді мобільних додатків створює значні ризики безпеки, які можуть мати серйозні наслідки як для самого додатку, так і для його користувачів:

1. Несанкціонований доступ і витоки даних - вбудовані секрети, такі як ключі API та токени аутентифікації, надають прямий доступ до хмарних сервісів і ресурсів. Якщо ці секрети будуть розкриті через код додатку, зловмисники можуть скористатися ними для отримання несанкціонованого доступу до хмарних ресурсів. Це може призвести до несанкціонованого доступу до даних, витоків даних та потенційного компрометації конфіденційної інформації користувачів, що зберігається в хмарі.

2. Збільшена поверхня атаки - вбудовування секретів безпосередньо в код додатку збільшує поверхню атаки, полегшуючи зловмисникам ідентифікацію та експлуатацію вразливостей. Інструменти та техніки реверс-інжинірингу можуть розкрити ці вбудовані секрети, що дозволяє зловмисникам отримати доступ до хмарних сервісів.

3. Зловживання хмарними ресурсами - якщо зловмисник отримує вбудовані облікові дані для доступу до хмарних сервісів, він може використовувати хмарні ресурси в злочинних цілях. Це може включати запуск несанкціонованих серверів, виконання витратних операцій

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

або проведення дій, що можуть призвести до значних фінансових витрат для хмарного облікового запису. Це може спричинити непередбачені витрати та виснаження ресурсів, що вплине як на роботу додатку, так і на його фінансову життєздатність.

4. Компрометація цілісності додатку - вбудовані секрети можуть також призвести до компрометації цілісності додатку. Якщо зловмисники можуть використати ці облікові дані для модифікації або втручання в роботу хмарних сервісів, вони можуть змінити функціональність додатку, впровадити шкідливий код або порушити нормальну роботу додатку. Це може підірвати довіру користувачів і завдати шкоди репутації додатку.

5. Складність у ротації та управлінні - вбудовані секрети ускладнюють управління та ротацію облікових даних. Хорошою практикою є регулярне оновлення секретів для зниження ризику довготривалого розголошення. Однак вбудовані секрети вимагають ручного втручання для оновлення, що може призвести до потенційних прогалин у безпеці та тривалого розкриття, якщо облікові дані скомпрометовані.

6. Відповідність стандартам та юридичні наслідки – вбудовування конфіденційної інформації в код додатку може також порушувати вимоги щодо відповідності регуляторним нормам і юридичним вимогам, пов'язаним із захистом даних та конфіденційністю. Регуляції, такі як GDPR, HIPAA та інші, передбачають суворий контроль за обробкою та захистом конфіденційної інформації. Розкриття облікових даних для хмарних сервісів може призвести до порушення вимог щодо відповідності, юридичних наслідків і штрафів.

Вбудовані секрети в розрізі категорій додатків

Дослідження показує, що деякі категорії Google Play мають значно різний відсоток додатків з вбудованими секретами. На рис. 2 представлено кількість проаналізованих додатків, кількість додатків, де були виявлені секрети, та відсоток таких додатків у кожній категорії.

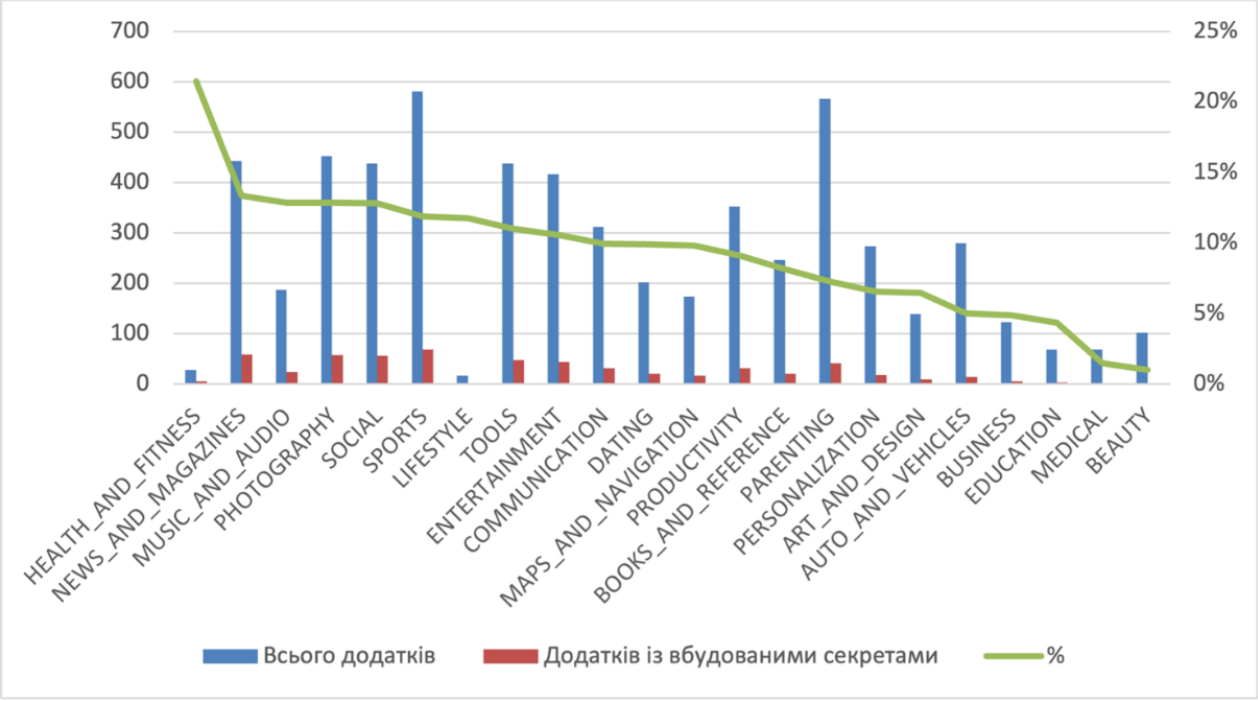

Рис. 2. Відсоток додатків з вбудованими секретами за категоріями

Як видно на Рис. 2, категорія з найбільшою кількістю вбудованих секретів – "Здоров'я та фітнес" – 21% додатків у цій категорії містять вбудовані секрети. У наступних чотирьох категоріях – "Новини та журнали", "Музика та аудіо", "Фотографія" та "Соціальні мережі" – 12% додатків мають вбудовані облікові дані.

Важливим висновком є те, що 10% додатків у категорії "Комунікація" мають вбудовані секрети. Вбудовування секретів, таких як облікові дані та токени API, у комунікаційних

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

додатках несе значні ризики безпеки, що може призвести до серйозних наслідків як для користувачів, так і для постачальників послуг. Комунікаційні додатки, які обробляють особисті повідомлення, дзвінки та медіа файли, є особливо вразливими до атак, коли секрети вбудовані в код додатку. Нижче описані деякі основні ризики, пов'язані з вбудованими секретами в таких додатках:

1. Несанкціонований доступ до даних користувачів - вбудовані облікові дані можуть бути легко отримані зловмисниками за допомогою методів реверс-інжинірингу. Такий несанкціонований доступ до токенів API або ключів аутентифікації може дозволити зловмисникам перехоплювати конфіденційні дані користувачів, включаючи особисті повідомлення, журнали дзвінків і медіа файли. Такі порушення представляють значні ризики для конфіденційності, оскільки скомпрометовані дані можуть бути використані для крадіжки особистих даних, стеження або експлуатації.
2. Компрометація цілісності комунікацій – цілісність комунікаційних сервісів залежить від захищених каналів передачі. Розкриті вбудовані секрети підривають цю цілісність, дозволяючи зловмисникам видавати себе за легітимних користувачів або сервіси. Це створює можливості для атак типу "людина посередині" (MITM), під час яких комунікації можуть бути перехоплені, змінені або інфіковані шкідливим вмістом без відома користувача, що ставить під загрозу автентичність і конфіденційність переданої інформації.
3. Переривання сервісу та атаки на відмову в обслуговуванні (DoS) - зловмисники, які мають доступ до вбудованих секретів, можуть використовувати їх для зловживання комунікаційними сервісами, надсилаючи надмірну кількість запитів або неправильно використовуючи API. Такі дії можуть призвести до атак на відмову в обслуговуванні (DoS), порушуючи роботу сервісу для легітимних користувачів. Цей тип атаки не лише негативно впливає на досвід користувачів, але й може завдати шкоди репутації постачальника послуг.
4. Захоплення облікових записів та крадіжка особистих даних - вбудовані токени API або облікові дані дозволяють зловмисникам отримати контроль над обліковими записами користувачів. Це призводить до несанкціонованого доступу, при якому зловмисники можуть блокувати доступ користувачів до їхніх облікових записів, надсилати шахрайські повідомлення або виконувати неавторизовані дії. Захоплення облікових записів може призвести до крадіжки особистих даних, атак із застосуванням соціальної інженерії або розповсюдження шкідливого контенту через скомпрометовані облікові записи.

Висновки

У цьому дослідженні представлено аналіз поширеності закодованих облікових даних у додатках та пов'язані із цим безпекові ризики. Аналіз 6,165 Android-додатків із різних категорій із використанням інструментів MobSF та Trufflehog виявив, що значна кількість додатків містить закодовані секрети, які становлять серйозну загрозу для конфіденційності даних користувачів та цілісності додатків. Результати дослідження показують, що закодовані облікові дані хмарних провайдерів, таких як AWS та GCP, є поширеним явищем, яке створює серйозну вразливість, що може призвести до несанкціонованого доступу, зловживання ресурсами та потенційних витоків даних. Зокрема, було виявлено, що можливими наслідками є несанкціонований доступ до конфіденційних даних, компрометація цілісності додатків та підвищена вразливість до атак типу "відмова в обслуговуванні" (DoS). Крім того, закодовані секрети ускладнюють ротацію та управління обліковими даними, що робить дотримання розробниками найкращих практик у забезпеченні безпеки додатків проблематичним.

Особливо схильними до наявності закодованих секретів виявилися додатки в категоріях "Здоров'я та фітнес", "Новини та журнали", "Музика та аудіо", "Фотографія" та "Соціальні мережі", причому найвища частота спостерігалася у додатках категорії "Здоров'я та фітнес". Важливо зазначити, що додатки для комунікації також продемонстрували високу поширеність закодованих секретів, що створює унікальні ризики через обробку чутливої особистої інформації, зокрема повідомлень, дзвінків та мультимедійних даних.

© Фединишин, Т. О., & Партика, О. О. (2024). Аналіз закодованих облікових даних в Android додатках. Сучасний захист інформації, 4(60), 117–124. https://doi.org/10.31673/2409-7292.2024.040012.

Рекомендації

Для вирішення виявлених проблем автори рекомендують впроваджувати системи управління секретами, такі як HashiCorp Vault, AWS Secrets Manager або Azure Key Vault, які забезпечують централізоване зберігання облікових даних, API-ключів і токенів аутентифікації, виключаючи їх вбудовування у код додатків. Також автори рекомендують інтеграцію принципів DevSecOps у процес розробки, що дозволить автоматизувати перевірку коду на наявність конфіденційної інформації на етапах CI/CD, що сприятиме своєчасному усуненню вразливостей до релізу. Автори наголошують на важливості регулярної ротації облікових даних, підтримувої за допомогою автоматизованих рішень, що зменшить ризики компрометації секретів у випадку їхнього витоку. У випадках, коли уникнути зберігання секретів у коді неможливо, автори рекомендують використовувати сучасні алгоритми шифрування, наприклад AES-256, для забезпечення належного захисту. Ці заходи знизять ризики і підвищать загальний рівня безпеки мобільних додатків.

Перелік посилань

1. C. Liu, et al., MobiPCR: Efficient, accurate, and strict ML-based mobile malware detection, Future Generation Comput. Syst. 144 (2023) 140–150. doi: 10.1016/j.future.2023.02.014.
2. Apple Appstore. URL: https://www.apple.com/appstore/
3. Google Play. URL: https://play.google.com/store
4. O. Mykhaylova, et al., Mobile Application as a Critical Infrastructure Cyberattack Surface, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3550 (2023) 29–43.
5. Y. Dreis, et al., Model to Formation Data Base of Internal Parameters for Assessing the Status of the State Secret Protection, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 277–289.
6. A. Horpenyuk, I. Opirskyy, P. Vorobets, Analysis of Problems and Prospects of Implementation of postQuantum Cryptographic Algorithms, in: Classic, Quantum, and Post-Quantum Cryptography, vol. 3504 (2023) 39–49.
7. E. Zaitseva, et al., Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Application Insecurity, Systems, 11(5) (2023). doi: 10.3390/systems11050242.
8. P. Zhu, et al., Using Blockchain Technology to Enhance the Traceability of Original Achievements, IEEE Trans. Eng. Manag. 70 (2023) 1693–1707.
9. S.-Y. Kuo, F.-H. Tseng, Y.-H. Chou, Metaverse Intrusion Detection of Wormhole Attacks based on a Novel Statistical Mechanism, Future Gener. Comput. Syst. 143 (2023) 179–190.
10. B. Zhurakovskyi, et al., Secured Remote Update Protocol in IoT Data Exchange System, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 67–76.
11. V. Sokolov, et al., Method for Increasing the Various Sources Data Consistency for IoT Sensors, in: IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PICST) (2023) 522–526. doi: 10.1109/PICST57299. 2022.10238518.
12. O. Shevchenko, et al., Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 2923 (2021) 277– 282.
13. V. Dudykevych, et al., Platform for the Security of Cyber-Physical Systems and the IoT in the Intellectualization of Society, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 449–457.
14. A Decade in, How Safe Are Your iOS and Android Apps? URL: https://www.nowsecure.com/blog/ 2018/07/11/adecade-in-how-safe-are-your-ios-andandroid-apps
15. Mobile Security Framework (MobSF). URL: https://mobsf.github.io/docs
16. Trufflehog. URL: https://github.com/trufflesecurity/ trufflehog
17. How Many Apps in Google Play Store? (2024). URL: https://www.bankmycell.com/blog/number-ofgoogle-play-store-apps/
18. Similarweb Digital Intelligence: Unlock Your Digital Growth. URL: https://www.similarweb.com/

Надійшла 23.11.2024