КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№4 (8), 2020
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2020.8.4960
УДК 004.056.53

Драгунцов Роман Ігорович
студент
Державний Університет Телекомунікацій, Київ, Україна
ORCID: 0000-0002-1781-7530
draguntsow@yahoo.com

Рабчун Дмитро Ігорович
к.т.н., доцент кафедри управління інформаційною безпекою
Державний Університет Телекомунікацій, Київ, Україна
ORCID: 0000-0002-5555-0910
rabchundima92@gmail.com

Бржевська Зореслава Михайлівна
аспірант кафедри інформаційної та кібернетичної безпеки
Державний Університет Телекомунікацій, Київ, Україна
ORCID: 0000-0002-7029-9525
zoreska.puzniak@gmail.com

ПРИНЦИПИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА БАЗІ КЛІЄНТСЬКИХ ДОДАТКІВ ДЛЯ ОС ANDROID

Анотація. У статті розглядаються, порівнюються та аналізуються основні вектори атак на інформаційні системи, що використовують додатки ОС Android в якості клієнтських інтерфейсів. Даний аналіз проводиться з метою отримання базового матеріалу для розробки практичних принципів забезпечення безпеки на рівні архітектури таких систем. Виконано категоріювання можливих атак та вразливостей, що полягають в їхній основі в контексті безпеки Android додатків та з урахуванням моделі безпеки самої операційної системи і середовища. Для виконання поставлених дослідницьких завдань було проведено аналіз компонентів Android-додатку та типової інформаційної інфраструктури досліджуваних систем. що так чи інакше впливають на їхню захищеність. Проведено аналіз наявної інформації щодо розповсюджених вразливостей цих компонентів та атак, що передбачають експлуатацію даних проблем. Досліджено декілька можливих моделей порушника, що можуть виконувати атаки на інформаційну систему. В результаті проведеного дослідження отримано аналітичні дані щодо векторів порушення цілісності та конфіденційності інформації з обмеженим доступом в інформаційних системах, що надають доступ до неї через мобільні додатки. В рамках порівняльної характеристики надається аналіз можливого впливу порушника на інформаційну систему зважаючи на його технічні можливості та поверхні атаки на кожному з визначених напрямків. Отримані теоретичні висновки щодо модифікації архітектури інформаційних систем, побудованих на базі мобільних додатків з метою підвищення їх захищеності від розповсюджених загроз інформації. Результати можуть бути використані для формування моделі загроз та порушника для додатку, що надає доступ до інформації з обмеженим доступом, розробки рекомендацій щодо реалізації тих чи інших етапів життєвого циклу інформаційної системи з метою зменшення ризиків компрометації даних, розробки технічних вимог до етапів тестування та розробки тощо.

Ключові слова: Android; безпека застосунків; архітектура інформаційних систем; мобільні додатки; безпека мобільних додатків; поверхня атаки; модель безпеки.

ISSN 2663 - 4023

1. ВСТУП

Постановка проблеми. В умовах зростання частоти використання додатків операційної системи Android в якості клієнтських інтерфейсів для роботи з критичними даними, такими як банківська, фінансова, державна інформація, актуальною постає проблема дослідження та протидії потенційним атакам, націленим на порушення характеристик даної інформації. Оскільки значна кількість вразливостей інформаційних систем зумовлена помилками на етапі побудови архітектури інформаційної системи, розробка архітектурних принципів щодо забезпечення безпеки таких систем є актуальною задачею.

Аналіз останніх досліджень і публікацій. Незважаючи на високу розповсюдженість пристроїв під управлінням операційної системи Android та, відповідно, інформаційних систем, що будуються на них, як на клієнтських, тематика захищеності архітектури таких систем не є достатньо висвітленою у наукових роботах [1]. Архітектура інформаційних систем підприємств, що базується на класичних стаціонарних системах (від Microsoft, Oracle, тощо) є широко відомою, її проблеми та особливості достатньо досліджені, а принципи та методології побудови (такі як CASE) наявні у відкритому доступі [2]. Слід звернути увагу на той факт, що у сучасних дослідженнях стосовно архітектури комерційних інформаційних систем акцент зроблено на неоднорідність та складну структуру таких систем [2, 3], що, в свою чергу, призводить до появи проблем захищеності. Зокрема, саме в питанні впровадження мобільних додатків у інфраструктуру, постають проблеми гетерогенності середовища та необхідності в кросплатформеності [4]

Значну роль у розробці інформаційних систем, побудованих на мобільних клієнтських додатках відіграє безпосередньо розробка таких додатків [5], оскільки для кожної з систем власник розробляє спеціальний клієнтський додаток для реалізації свої потреб [6]. Це зумовлює фрагментацію та, відповідно, виникнення значної кількості загроз.

В роботі [1] окрім теоретичного дослідження вразливостей застосунків Android наводиться аналіз конкретних прикладів таких проблем безпеки знайдених у додатках в Google Play Market. Зокрема, такі вразливості, що спричинені недостатнім захистом технічної конфіденційної інформації та некоректним використанням Android API були визначені як достатньо розповсюджені та з технічної точки зору висвітлюються у роботах [7,8]. На ризики пов’язані з перепакуванням додатків та недовіреністю середовища їхнього функціонування звертається увага у роботах [1,7] – зокрема, в статті [7] також висвітлюється проблема шкідливого ПЗ для ОС Android, що також впливає на ландшафт загроз клієнтського додатку. Вразливості, пов’язані з архітектурою та кодом самої ОС Android – такі як QuadRooter, Certifi-gate – наводяться в роботі [9]. Повноцінний перелік дій з захисту усіх елементів інформаційної системи на базі Android додатку з перспективи AppSec-підходу викладено у джерелі [10]. У ньому ж надається вичерпне розділення даних операцій на домени по відношенню до додатку – архітектура ОС, дозволи додатків, безпека компонентів, безпека сховища даних та безпека каналу комунікації.

Важливість використання моделей порушника та визначення поверхні атаки досліджуваної системи викладено у роботах [11,12].

Мета статті. В даній статті за мету береться визначення принципів забезпечення безпеки для інформаційних систем, в яких клієнтська сторона представляється у вигляді мобільних додатків операційної системи Android. Реалізація даних принципів

при побудові нової системи або модифікації існуючої має зменшити ризики реалізації атак досліджених типів.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

2.1 Вектори атак на інформаційні системи на базі клієнтських додатків для ОС Android

В рамках даного дослідження за модельну приймається така інформаційна система, що складається з додатку для ОС Android в якості клієнтського інтерфейсу, WEB-серверу, що представляє собою інтерфейс Backend, певної інформаційної інфраструктури в рамках backend та комунікаційного каналу за протоколом HTTP.

Перш за все, слід окреслити основні напрями атак, що можуть бути здійснені на такі системи - їх перелік надається нижче [13]:

1) На процеси серверної частини;
2) На клієнтський додаток;
3) На канал комунікації.

Далі надається дослідження складу кожної з цих категорій шляхом розгляду функціональних компонентів атакованого об’єкту, що обумовлюють ті чи інші вразливості.

2.1.1 Вектори атак на серверну частину ІС

Інформаційна система, що передбачає використання мобільного додатку в якості клієнтського інтерфейсу, містить серверну частину, яка, найчастіше, реалізує основну частину функціональності [14]. Така частина створюється за допомогою розповсюджених технологій проектування та, здебільшого, містить можливості для взаємодії з певним накопичувачем даних – в даному дослідженні архітектура і тип сховища даних не є принциповими, тому, з метою спрощення, для позначення серверного сховища даних інформаційної системи буде використовуватись термін «база даних».

В даному випадку під процесами серверної частини додатку маються на увазі будь-які процеси в інформаційній системі, що забезпечуються функціоналом, реалізованим серверним ПЗ. До таких, зокрема мають належати [15]:

- Аутентифікація, авторизація та розмежування доступу;
- Контроль сеансів;
- Бізнес-логіка;
- Валідація даних;
- Обробка помилок;

Дані елементи є характерними для побудови додатків, що оперують інформацією з обмеженим доступом. Реалізація кожного з цих елементів може містити вразливості, експлуатація яких може призвести до порушення нормального функціонування системи та порушення параметрів інформації з обмеженим доступом.

В широкому розумінні, експлуатація вразливостей механізмів авторизації та аутентифікації може призвести до горизонтального або вертикального підвищення привілеїв. Вертикальне підвищення привілеїв обумовлює отримання користувачем більших прав, аніж ті, що надаються йому системою за нормального її функціонування. Горизонтальне підвищення привілеїв призводить до отримання прав, аналогічних тим, якими володіє атакуючий, але таких, що дозволяють персоніфікувати себе як іншого

користувача системи. Обидві атаки обумовлюють високий ризик порушення параметрів інформації з обмеженим доступом – наприклад, заволодіння конфіденційною інформацією, що належить іншому користувачу, заволодіння інформацією, що потребує більшого рівня допуску, тощо. Прикладами вразливостей, що обумовлюють такі проблеми, можуть бути можливість типової SQL ін'єкції у авторизаційному механізмі у випадку звернення до бази даних, відсутність обмежень на неуспішні спроби аутентифікації (захист від brute-force атак) тощо. Вразливості ж розмежування доступу ґрунтуються на проблемах у реалізації відповідних процесів перевірки ідентифікатора користувача (його сеансу та ролі) з наданими йому правами. Такі вразливості є критичними по відношенню до систем, що оброблюють інформацію з обмеженим доступом. До таких проблем належать, зокрема, IDOR (Insecure Direct Object Reference – небезпечний прямий доступ до об'єкту), відсутність або некоректна перевірка токену доступу тощо.

Вразливості контролю сеансів тісно пов'язані з процесами авторизації, оскільки оперують переважно одною множиною ресурсів системи. Порушення в роботі механізмів контролю сеансів можуть призводити до захоплення сеансу іншого користувача, створення умов для такого захоплення, а в деяких випадках до створення ризику відтермінованої компрометації. До таких вразливостей належать, у першу чергу, відсутність перевірки ідентифікатора сесії при аутентифікації (можливість реалізації атаки Session Fixation), використання ідентифікатору сесій низької складності, тобто таких, що можуть бути підібраними за відносно короткий час, відсутність прив'язки сесії до пристрою та відсутність (або недостатність) обмеженості часу дійсності сеансу.

Вразливості бізнес-логіки здебільшого не мають чіткої класифікації, оскільки представляють собою різнорідні проблеми в реалізації тих чи інших алгоритмів у роботі з даними в додатку, а отже залежать від конкретної реалізації. Вразливості бізнес-логіки, зазвичай, дозволяють використовувати незапланований при проектуванні функціонал, не порушуючи коректність роботи програми. До таких проблем, зокрема, належать недостатня валідація порядку дій у процесі, порушення в обробках паралельних процесів (Race condition), тощо [16].

Валідація даних має застосовуватися до усіх даних, що отримуються з недовіреного середовища, у випадку відсутності у користувача запланованої можливості для модифікації таких даних. Порушення даного процесу, його відсутність або недоліки можуть призвести до маніпулювання даними порушником, в робочому циклі системи. Це, в свою чергу, може призвести до незапланованих змін у такому процесі. Вразливості валідації можуть мати необмежений вплив на систему, що є критичним при обробці даних з обмеженим доступом. До найбільш поширених типів експлуатації помилок валідації належать усі типи атак ін'єкцій (SQL, OS-command, XML, JavaScript тощо).

Недостатня або некоректна обробка помилок може дозволити порушнику змінити коректний хід роботи програми та, таким чином, отримати доступ до розголошенної технічної інформації або отримати певні можливості впливу на систему. Головною метою системи обробки помилок є стабілізація роботи системи у випадку нештатної ситуації, яка, зокрема, може бути створена штучно [17].

2.1.2. Вектори атак на клієнтський додаток

Атаки, що здійснюються на клієнтський мобільний додаток в рамках досліджуваної архітектури, можуть мати вплив у першу чергу на ті дані, що оброблюються локально. Підвищення привілеїв та доступ до даних інших користувачів

мають значно меншу ймовірність. До основних функціональних елементів, специфічних для клієнтського додатку, належать [7]:

- Інтерфейс взаємодії з користувачем;
- Міжпроцесна взаємодія;
- Локальне накопичення даних;

Окрім цих функціональних елементів на клієнтський додаток покладається відповідальність за встановлення захищеного каналу зв'язку із сервером [10], однак дана проблема винесена в окрему категорію – канал зв'язку.

Інтерфейс взаємодії з користувачем в системі Android забезпечується за допомогою елементів додатку Activity та WebView. Activity є абстракцією робочого екрану додатку з елементами інтерфейсу, призначеними для взаємодії з користувачем. Інакше кажучи, кожен елемент Activity надає користувачу інтерфейс для взаємодії з даними системи. Компоненти WebView призначені для відображення веб-сторінок і в цілому аналогічні іншим типам веб-переглядачів, а отже й наслідують більшість їхніх вразливостей. До вразливостей, що можуть виникати через інтерфейс взаємодії з користувачем належать такі, що обумовлені закладеними в код додатку елементами конфіденційної інформації – це створює ризик їх виявлення шляхом аналізу за допомогою методів зворотної розробки (reverse engineering). Іншим типом є можливості програмного доступу до елементу інтерфейсу з боку інших додатків, що функціонують в операційній системі. [8]

Міжпроцесна взаємодія в ОС Android реалізується переважно за допомогою функціоналу системного віртуального пристрою Binder. Обмін даними виконується за допомогою механізмів BroadcastReceiver та ContentProvider, що, відповідно, отримують та надають дані іншим процесам. Дані, що отримуються таким чином, є недовіреними, а отже можуть створювати ризик зловживання функціоналом та внесення неправомірних змін в робочий цикл додатку. В залежності від алгоритмів, що здійснюють обробку або надання даних, можуть виникати ті чи інші вразливості, пов`язані, здебільшого, з валідацією або логікою додатку. Так, наприклад, вразливість в реалізації ContentProvider може створювати витоки інформації до інших додатків, а в BroadcastReceiver – непередбачені можливості для функціоналу [8].

Локальне накопичення даних в додатках ОС Android може здійснюватися за допомогою файлів у локальній файловій системі (зокрема спеціальних – Shared Preferences), за допомогою локальних баз даних, таких як SQLite, або в системних сховищах, таких як KeyChain. За нормальних умов доступ до даних конкретного додатку забороняється для інших процесів, що не мають явного на це дозволу, однак існує ряд вразливостей, що можуть обумовлювати витоки інформації через локальне сховище. Найпростішим прикладом такої проблеми є збереження даних додатком на SD карті – архітектура ОС Android надає менші гарантії безпеки даних збережених на зовнішньому носії, а отже доступ до них може бути отриманий з інших процесів. Більші можливості для експлуатації можуть мати шкідливі процеси з підвищеними привілеями, що можуть так, чи інакше отримати доступ до захищених файлів додатку [10].

ISSN 2663 - 4023

Окрім окреслених функціональних компонентів, вразливості клієнтського додатку можуть бути обумовлені й іншими факторами, такими як вразливості використаних бібліотек чи класичні дефекти нативного коду - в разі наявності такого. Однак, переважно, вектор експлуатації для порушника в такому разі не змінюється і залишається в межах окреслених категорій. Всі інші вразливості клієнтського додатку, що мають відмінний вектор експлуатації, можуть бути віднесені до категорії вразливостей, що зумовлені середовищем.

2.1.3. Вектори атак на канал комунікації

Переважна більшість сучасних клієнт-серверних додатків для ОС Android і, зокрема, таких, що призначені для обробки інформації з обмеженим доступом, використовують для комунікації протокол HTTP [13]. Використання даного протоколу без додаткового захисту саме по собі може вважатися серйозною вразливістю системи, оскільки при цьому відсутні будь-які можливості для забезпечення конфіденційності та цілісності інформації при передачі. Для забезпечення цих характеристик використовується захищений протокол з'єднання – HTTPS. Однак, зважаючи на той факт, що мобільні пристрої працюють, здебільшого, у недовіреному середовищі, яке може бути контрольоване потенційним порушником [1], можливості для використання звичайного протоколу HTTPS є обмеженими – порушник може контролювати DNS сервер мережі та перехоплювати клієнтський трафік, таким чином підриваючи можливості для перевірки криптографічного сертифікату серверу. Для подолання цієї проблеми використовується технологія SSL Pinning, що передбачає закладення певної множини довірених сертифікатів серверу безпосередньо у код додатку, таким чином, вилучаючи етап перевірки серверного сертифікату через третю сторону з процесу встановлення захищеного з'єднання. Однак даний механізм може бути імплементований некоректно, шо нівелює його захисні можливості.

З наведених фактів можна виділити чотири можливі типи атак на канал комунікації в системі досліджуваної архітектури [18]:

1) MitM атака на незахищений канал комунікації (HTTP);
2) MitM атака з підміною DNS серверу в мережі на захищений канал комунікації (HTTPS);
3) Експлуатація вразливостей імплементації технології SSL Pinning для здійснення атаки на захищений канал;
4) Криптографічні атаки на захищений канал комунікації.

2.2. Концепція безпеки досліджуваних ІС

В рамках дослідження розглядається архітектура інформаційної системи, що відображена на рис.1.

ISSN 2663 - 4023

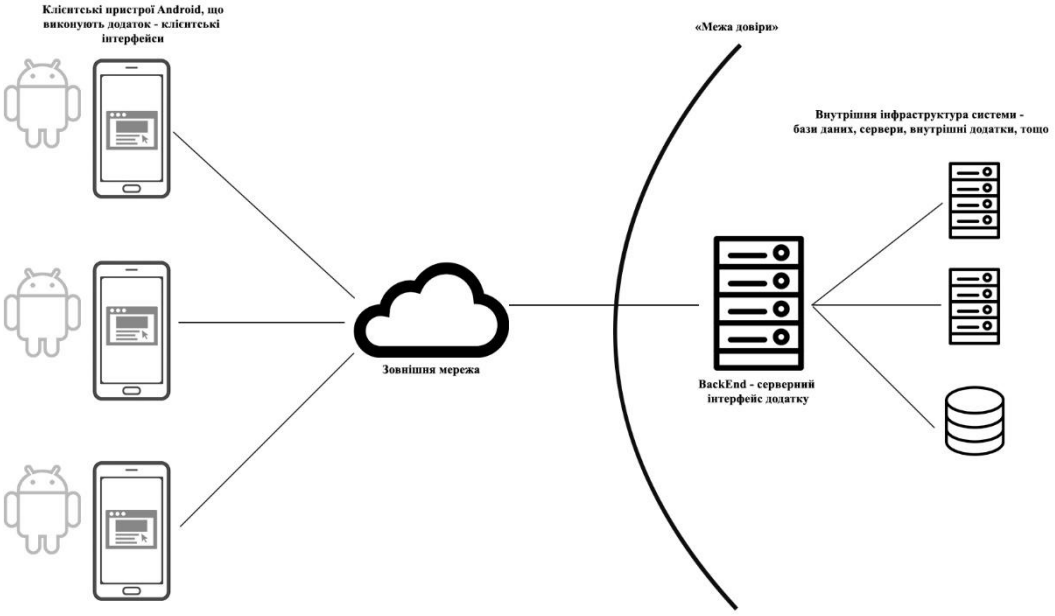

Рис. 4 – Схема, що відображає типову структуру інформаційної системи клієнт-серверної архітектури з Android-додатками в якості клієнтів [4]

Така архітектура передбачає збереження всієї конфіденційної інформації системи в сховищі даних на серверній частині. Надання доступу до даних та логіка обробки реалізовані у серверному додатку, що реалізує API, яке через канал комунікації (HTTPS) пов'язане з клієнтським додатком. Слід відмітити, що у реальному середовищі кількість клієнтських додатків буде більшою за 1, а архітектура та середовище системи, в якій додаток функціонує, є невідомими, а отже недовіреними. Крім того, слід звернути увагу на те, що цілісність додатку не є гарантованою, оскільки його виконуваний код може бути проаналізований та змінений на стороні користувача. Це, в свою чергу, означає, що сам додаток може також не мати відомої архітектури, а отже також переводить його у межу недовіреного середовища [1].

Таким чином, межу довіри в досліджуваній інформаційній системі можна визначити перед вхідними точками серверного API – всі інші елементи належать до недовіреного середовища. Отже, клієнтський інтерфейс може бути будь-яким, сумісним з серверним API та каналом комунікації. Оскільки на меті є визначення основних проблем безпеки такої архітектури, слід визначити відповідні моделі порушника. Такі моделі є абстрагованими від мотивації та інструментарію порушника, оскільки ці параметри не мають впливу на архітектуру – за мету порушника приймається неавторизований доступ до конфіденційних даних.

A. Порушник є стороннім атакуючим, що не має доступу до конкретного авторизованого клієнтського додатку, доступу до клієнтського пристрою та доступу до серверної частини системи. При цьому порушник має змогу аналізувати алгоритми роботи типового клієнтського додатку та його взаємодію з сервером. В рамках даної моделі також розглядається можливість отримання певного контролю над мережевим середовищем. Очевидно, що дана модель може бути найбільш розповсюдженою, оскільки не передбачає жодних специфічних умов для реалізації;
B. Порушник має певний рівень доступу до пристрою клієнта, що є прихованим від самого клієнта, доступ до серверної частини та безпосередньо до додатку відсутній.

Дана модель передбачає завчасне отримання порушником доступу до клієнтського пристрою на програмному рівні – наприклад, за допомогою троянського додатку, що обмежує вірогідність реалізації таких сценаріїв;

C. Порушник представляє собою клієнта в системі та може змінювати клієнтський додаток довільним чином. Доступ до серверної частини системи – на рівні звичайного клієнта. В залежності від призначення додатку, дана модель може бути як повністю вірогідною (відкрита реєстрація в системі), такі і достатньо складною для реалізації, що переводить її до категорії інсайдерських (закрита система з реєстраційною верифікацією). Розглядаючи системи, призначені для використання широким колом користувачів, передбачається перший варіант;

D. Порушник має повний фізичний доступ до пристрою клієнта. Для реалізації цієї моделі необхідне безпосереднє заволодіння пристроєм, що потребує від порушника здійснення певних дій поза інформаційним простором (викрадення, конфіскація, тощо)

В цілях дослідження розглядається можливість компрометації додатку як для користувача з середнім рівнем привілеїв, так і для користувача з підвищеним рівнем доступу до системи, а отже й до даних інших користувачів. Очевидно, що принципової різниці у векторах експлуатації для порушників обраних типів, у даній ситуації немає.

Очевидно, що порушник А має вектори експлуатації, що зводять поверхню атаки до серверного Backend інформаційної системи та каналу комунікації між ним та клієнтським додатком. Завдяки наявності знань про алгоритми функціонування клієнтського додатку, такий порушник має уявлення про алгоритми обробки клієнтських запитів на серверній частині, а отже може проводити дослідження на предмет вразливостей. У випадку відкритості або некоректного захисту каналу комунікації можлива реалізація атаки Man-in-the-Middle.

Поверхня атаки для порушника В представляє собою, у першу чергу, інтерфейс міжпроцесної взаємодії додатків, механізм доступу до носіїв інформації (внутрішнього та зовнішнього накопичувачів) та системні виклики ОС Android. Доступ до таких внутрішніх частин ОС може бути наданий, наприклад, за допомогою троянського додатку, встановленого на пристрої користувача. Також, можлива реалізація атак типу Tapjcking, Man-in-the-Disk, логування клієнтських дій (Keylogging).

Порушник моделі С має привілейований доступ до системи, тобто отримує певний рівень доступу до своїх даних у інформаційній системі. Це означає, що поверхнею атаки для нього є простір функціоналу, доступний для авторизованого користувача, тобто такий, що знаходиться поза першорівневими захисними бар'єрами. В поєднанні із уявленням про алгоритми взаємодії клієнтського додатку із серверною стороною, цей доступ дозволяє порушнику проводити аналіз захищеності серверної сторони системи, оминаючи захисні механізми першої лінії.

Модель D передбачає захоплення порушником пристрою користувача, за умови відсутності системного рівня блокування пристрою. Таким чином, в залежності від стану клієнтського додатку, можливі два сценарії розвитку подій:

1) Отримання доступу до авторизованого сеансу користувача;
2) Отримання доступу лише до системи й додатку без авторизованого сеансу.

В першому випадку порушник здійснює неавторизований доступ до конфіденційних даних клієнта та переходить до моделі С. В другому сценарії порушник наближається до моделі В з розширеними привілеями.

Аналізуючи дані моделі, очевидно, що поверхня атаки для них не є однорідною і включає або не включає в себе т, чи інші вектори з визначених у розділі 2.1 – це відображено в таблиці 1:

Таблиця 1

Порівняння поверхні атаки для різних моделей порушника

Модель	Backend	Додаток	Канал комунікації
A	Частковий	Відсутній	Повний/частковий
B	Частковий	Частковий/Повний	Відсутній
C	Частковий	Повний	Н/в
D	Частковий	Повний	Н/в

В таблиці вказуються наступні можливі типи впливу:

- Н/в – вплив можливий, але не має сенсу для даної моделі порушника;
- Відсутній – вплив неможливий;
- Частковий – порушник має обмежені можливості щодо атаки на об'єкт;
- Повний – порушник не має технічних обмежень середовища щодо атаки на об'єкт.

Вплив на інформацію в системі для кожного з окреслених векторів експлуатації визначається в залежності від конкретної архітектури та типу оброблюваних даних, тому в даному аналізі наведені лише узагальнені твердження про поверхню атаки для кожного з типів порушників.

3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеного дослідження проблем забезпечення безпеки архітектури інформаційних систем на основі Android додатків зроблено ряд висновків:

1. Найбільшу поверхню атаки надає безпосередньо клієнтський додаток, оскільки одразу за декількох сценаріїв експлуатації порушник отримуватиме повний доступ до атакованого об'єкта;
2. За будь-яких умов атаки на серверну частину додатку можливі, однак можливості порушника завжди будуть обмежені;
3. Канал комунікації додатку та серверної частини надає найменшу поверхню атаки порушнику, що забезпечує мінімальну кількість необхідних дій, щодо забезпечення безпеки інформації

Наведені висновки про значення поверхні атаки по різним векторам експлуатації дають підстави запропонувати принципи забезпечення безпеки архітектури інформаційної системи на базі клієнтських додатків Android:

1. Мінімум клієнтських даних мають зберігатися та оброблятися в додатку на клієнтському пристрої. Уся функціональність системи та логіка мають бути реалізовані виключно на серверній частині, в межах зони довіри;
2. Має бути імплементований жорсткий контроль користувацьких сеансів. Жодні персоналізовані та конфіденційні дані, пов'язані з роботою додатку, не мають залишатися на пристрої після завершення сеансу;
3. Найбільший акцент має бути зроблений в підвищенні захищеності серверної частини інформаційної системи – найбільш прискіпливий аудит вихідного

коду та опису архітектури, тестування, впровадження захисних механізмів та інших контролів безпеки;

4. Захист каналу зв'язку має бути імплементований на рівні підтримки цілісності та конфіденційності інформації – це виконується за допомогою криптографічного захисту каналу, за допомогою криптографічних протоколів SSL/TLS або інших. Таким чином, даний вектор експлуатації може бути повністю нівельований.

Перспективним напрямком досліджень відносно до отриманих результатів вважаємо аналіз можливостей практичного впровадження розроблених принципів та актуалізації по відношенню до існуючого ландшафту загроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, "A Study of Android Application Security", 2011.
[2] И. Лещаков, "Архитектура информационной системы предприятий", *Молодой ученый*, vol. 1, no. 155, pp. 13-15, 2017. [Accessed 10 March 2020].
[3] Z. Joerg, *Architecture of Interoperable Information Systems - An enterprise Model-based Approach for Describing and Enacting Collaborative Business Processes*. 2012, pp. 1-3.
[4] В. Копытов, А. Шульгин and С. Федоров, "РАЗРАБОТКА АРХИТЕКТУРЫ ИНТЕГРАЦИОННОЙ СРЕДЫ КРОССПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМОЙ", *Международный научно-исследовательский журнал*, vol. 1, no. 38, 2015. [Accessed 10 March 2020].
[5] R. Al-Sayyed, S. Manaseer and O. Rababeh, "Mobile Information System, How to Build with Case Study", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 4, no. 4, 2010. Available: 10.3991/ijim.v4i4.1357 [Accessed 10 March 2020].
[6] A. Jørgensen, *The Future of the Mobile Application Market*. Trondheim: Norwegian University of Science and Technology, 2014, pp. 29-34.
[7] S. Chatterjee, K. Paul, R. Roy and A. Nath, "A Comprehensive Study on Security issues in Android Mobile Phone — Scope and Challenges", *International Journal of Innovative Research in Advanced Engineering*, vol. 3, no. 3, 2016. [Accessed 10 March 2020].
[8] B. Schmerl et al., "Architecture Modeling and Analysis of Security in Android Systems", *Software Architecture*, pp. 274-290, 2016. Available: 10.1007/978-3-319-48992-6_21 [Accessed 10 March 2020].
[9] S. Khan and I. Firdous, "Review on Android App Security", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 4, pp. 225-228, 2017. Available: 10.23956/ijarcsse/v7i4/0195 [Accessed 10 March 2020].
[10] J. Six, *Application security for the Android platform*. Beijing: O'Reilly, 2012.
[11] P. Manadhata, K. Tan, R. Maxion and J. Wing, "An Approach to Measuring a System's Attack Surface", 2007. Available: 10.21236/ada476977 [Accessed 10 March 2020].
[12] Q. Do, B. Martini and K. Choo, "The role of the adversary model in applied security research", *Computers & Security*, vol. 81, pp. 156-181, 2019. Available: 10.1016/j.cose.2018.12.002 [Accessed 10 March 2020].
[13] European Maritime Safety Agency, "System and Application Technical Landscape", 2014.
[14] P. Gadient, M. Ghafari and O. Nierstrasz, *Web APIs in Android through the Lens of Security*. 2020.
[15] NIST, "Guide to Secure Web Services", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, 2007.
[16] F. Sun, L. Xu and Z. Su, "Detecting Logic Vulnerabilities in E-commerce Applications", *Proceedings 2014 Network and Distributed System Security Symposium*, 2014. Available: 10.14722/ndss.2014.23351 [Accessed 10 March 2020].
[17] "OWASP Top Ten", *Owasp.org*, 2017. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed: 10- Mar- 2020].
[18] S. Fahl, M. Harbach, T. Muders, M. Smith, L. Baumgärtner and B. Freisleben, "Why eve and mallory love android", *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*, 2012. Available: 10.1145/2382196.2382205 [Accessed 10 March 2020].

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (8), 2020
ISSN 2663 - 4023

Drahuntsov Roman
student
State University of Telecommunications, Kyiv, Ukraine
ORCID: 0000-0002-1781-7530
draguntsow@yahoo.com

Rabchun Dmytro
PhD, Associate Professor at the Department of Information and Cyber Security
State University of Telecommunications, Kyiv, Ukraine
ORCID: 0000-0002-5555-0910
rabchundima92@gmail.com

Brzhevska Zoreslava
postgraduate student at the Department of information and cybersecurity
State University of Telecommunications, Kyiv, Ukraine
ORCID: 0000-0002-7029-9525
zoreska.puzniak@gmail.com

ARCHITECTURE SECURITY PRINCIPLES OF THE ANDROID APPLICATIONS-BASED INFORMATION SYSTEM

Abstract. In this article common attack vectors on the information systems, which are based on the Android client applications, are observed, analyzed and compared. The purpose of this analysis consists in creating the theoretical base for development the practical principles of securing the architecture level of such systems. To accomplish the aims set, there was conducted the categorization of attacks and vulnerabilities specific to the Android information infrastructure and environment. There were also conducted analysis of Android application functional components and typical underlying infrastructure which have possible impact on a system security. Available data about the widespread vulnerabilities of the described elements was analyzed in context of possible exploitation. Based on the Android application usage model there were figured out several adversary models and attack vectors related to the researched information system type. Developed adversary models were formed with a focus on technical possibilities and threat abstraction. Mentioned vectors can be used by an attacker to violate the confidentiality and integrity of critical information in the system. The carried out research was used to form the characteristic comparison of the mentioned vectors and adversary models to evaluate the attack surface on the different parts of information system represented as attack vectors. As a result, we have developed the theoretical principles for securing the architecture of Android applications-driven information systems. Achieved results can be used to form the threat and adversary model, create practical recommendations for the information risk reducing practices in Android-applications driven information systems and to develop the technical requirements for security testing and development.

Keywords: Android; application security; information system architecture; mobile application; mobile security; attack surface; security model.

REFERENCES (TRANSLATED AND TRANSLITERATED)

[1] W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, "A Study of Android Application Security", 2011. (in English)
[2] I. Leshhakov, "Corporative information system architecture", *Molodoj uchenyj*, vol. 1, no. 155, pp. 13-15, 2017. [Accessed 10 March 2020]. (in Russian)
[3] Z. Joerg, *Architecture of Interoperable Information Systems - An enterprise Model-based Approach for Describing and Enacting Collaborative Business Processes*. 2012, pp. 1-3. (in English)
[4] V. Kopytov, A. Shulgin and S. Fedorov, " DEVELOPMENT OF THE ARCHITECTURE INTEGRATION ENVIRONMENT CROSS-PLATFORM MOBILE APPLICATIONS WITH CORPORATE INFORMATION SYSTEMS ", *Mezhdunarodnyj nauchno-issledovatel'skij zhurnal*, vol. 1, no. 38, 2015. [Accessed 10 March 2020]. (in Russian)

[5] R. Al-Sayyed, S. Manaseer and O. Rababeh, "Mobile Information System, How to Build with Case Study", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 4, no. 4, 2010. Available: 10.3991/ijim.v4i4.1357 [Accessed 10 March 2020]. (in English)
[6] A. Jørgensen, *The Future of the Mobile Application Market*. Trondheim: Norwegian University of Science and Technology, 2014, pp. 29-34. (in English)
[7] S. Chatterjee, K. Paul, R. Roy and A. Nath, "A Comprehensive Study on Security issues in Android Mobile Phone — Scope and Challenges", *International Journal of Innovative Research in Advanced Engineering*, vol. 3, no. 3, 2016. [Accessed 10 March 2020]. (in English)
[8] B. Schmerl et al., "Architecture Modeling and Analysis of Security in Android Systems", *Software Architecture*, pp. 274-290, 2016. Available: 10.1007/978-3-319-48992-6_21 [Accessed 10 March 2020]. (in English)
[9] S. Khan and I. Firdous, "Review on Android App Security", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 4, pp. 225-228, 2017. Available: 10.23956/ijarcsse/v7i4/0195 [Accessed 10 March 2020]. (in English)
[10] J. Six, *Application security for the Android platform*. Beijing: O'Reilly, 2012. (in English)
[11] P. Manadhata, K. Tan, R. Maxion and J. Wing, "An Approach to Measuring a System's Attack Surface", 2007. Available: 10.21236/ada476977 [Accessed 10 March 2020]. (in English)
[12] Q. Do, B. Martini and K. Choo, "The role of the adversary model in applied security research", *Computers & Security*, vol. 81, pp. 156-181, 2019. Available: 10.1016/j.cose.2018.12.002 [Accessed 10 March 2020]. (in English)
[13] European Maritime Safety Agency, "System and Application Technical Landscape", 2014.
[14] P. Gadient, M. Ghafari and O. Nierstrasz, *Web APIs in Android through the Lens of Security*. 2020. (in English)
[15] NIST, "Guide to Secure Web Services", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, 2007. (in English)
[16] F. Sun, L. Xu and Z. Su, "Detecting Logic Vulnerabilities in E-commerce Applications", *Proceedings 2014 Network and Distributed System Security Symposium*, 2014. Available: 10.14722/ndss.2014.23351 [Accessed 10 March 2020]. (in English)
[17] "OWASP Top Ten", *Owasp.org*, 2017. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed: 10- Mar- 2020]. (in English)
[18] S. Fahl, M. Harbach, T. Muders, M. Smith, L. Baumgärtner and B. Freisleben, "Why eve and mallory love android", *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*, 2012. Available: 10.1145/2382196.2382205 [Accessed 10 March 2020]. (in English)