ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

Дисциплина:

«Обеспечение информационной безопасности мобильных устройств»

ОТЧЕТ ПО ЛАБАРАТОРНОЙ РАБОТЕ №1

«Проблемы безопасности и угрозы платформ Android и IOS»

Выполнил:

Гаврилова В. В., студент группы N33471

(подпись)

Проверил:

Федоров Иван Романович

24.02.2022

(отметка о выполнении)

(подпись)

Санкт-Петербург

2022г.

Содержание

1. Цель работы

Ознакомиться с основными проблемами безопасности и угрозами таких мобильных операционных систем как Android и IOS.

2. Основная часть

Приложения для Android с критически опасными уязвимостями встречаются несколько чаще, чем программы для iOS (43% против 38%). Однако эта разница несущественна, и общий уровень защищенности клиентских частей мобильных приложений для Android и iOS примерно одинаков. Около трети всех уязвимостей в клиентских частях мобильных приложений для обеих платформ имеют высокий уровень риска.

2.1 Проблемы безопасности Android [1]

А) Уязвимости ядра Linux и его модулей

К данной категории проблем относятся уязвимости, которые присущи всем ОС, основанным на той же версии ядра Linux, что и ОС Android. Эксплойты, использующие уязвимости в ядре, могут получить данные пользователя или права администратора системы. Повысив привилегии процесса до прав администратора системы, вредоносная программа может отключить систему безопасности Android, вставить в существующие программы вредоносный код и установить руткит. К тому же производители различных устройств добавляют в ядро модули для своих устройств; в этих модулях также могут быть уязвимости.

Б) Уязвимости модулей в машинных кодах

Android-приложения поддерживают запуск машинного кода через интерфейс JNI. Это порождает еще одну категорию уязвимостей, связанную с широко известными уязвимостями утечек памяти в низкоуровневых языках (например, в C и C++). Поскольку на уровне процессов ОС Android нет никаких ограничений, кроме накладываемых ядром Linux, сторонние библиотеки в машинных кодах могут использовать разрешения, выданные всему приложению, для совершения вредоносной активности. Также модули приложения в машинных кодах используются авторами вредоносных приложений, чтобы обойти инструменты анализа и мониторинга уровня Android. Эти модули также могут использовать техники противодействия анализу, используемые в обычных программах.

В) Уязвимости механизмов межкомпонентного взаимодействия

К данной категории относятся уязвимости, связанные с взаимодействием между различными компонентами приложений. Так как на уровне операционной системы приложение ограничено песочницей процесса, ему необходим механизм доступа к

компонентам ОС Android для взаимодействия с ними. Это происходит через устройство /dev/Binder и различные другие сервисы ОС Android. Как уже говорилось выше, параметры этого доступа задаются в файле манифеста, в виде XML-файла с разрешениями. Так, например, приложение может воспользоваться правами доступа другого приложения и получить с помощью него данные через ICC. Также могут быть уязвимости, связанные со сторонними библиотеками. Сторонние библиотеки, используемые в приложении, получают тот же набор ограничений, что и само приложение. Поэтому сторонние библиотеки могут использовать разрешения, выданные всему приложению, для совершения вредоносной активности. Приложения к тому же могут перехватывать системные события, пересылаемые через широковещательный запрос, и сохранять информацию о входящих звонках и СМС.

Г) Уязвимости модификаций и компонентов производителей устройств

В последнее время производители различных мобильных устройств стали выпускать свои модифицированные прошивки Android. Эти прошивки могут содержать различные приложения и сервисы, разработанные производителем устройства, которые чаще всего нельзя удалить. Например, в октябре 2016 года был обнаружен скрытый бэкдор в прошивках Foxconn. Анализ этих сервисов, показывает, что в них содержится от 65 до 85% уязвимостей, обнаруженных во всей системе. К тому же стоит отметить, что уязвимости, которые были обнаружены и исправлены в основной ветке Android, могут долгое время оставаться в ветках производителей устройств.

Д) Уязвимости в самих приложениях

Каждое приложение сохраняет какие-то данные о пользователе. Эти данные должны быть защищены должным образом, чтобы к ним не могли получить доступ другие приложения, — но такая защита предусмотрена не всегда. Например, Skype в одной из версий приложения сохранял базу данных контактов в открытом виде на диске. Таким образом, контакты можно было прочитать любым другим приложением, у которого есть доступ к диску. Также приложения могут использовать криптографические библиотеки с ошибками или же какие-то собственные реализации. К тому же не все приложения производят хорошую аутентификацию и авторизацию пользователя. Кроме этого, приложения могут позволять SQL-инъекции и подвержены атакам XSS. Также стоит отметить, что большинство разрабатываемых приложений написаны на Java без использования какой-либо защиты для бинарного кода, а байт-код Java, как известно, легко поддается дизассемблированию и анализу. К этой категории уязвимостей относится также известный список Mobile OWASP-10.

Е) Уязвимости во встроенных сервисах и библиотеках

Стандартный набор библиотек и сервисов, работающих в Android, также содержит уязвимости. Например, недавно была обнаружена уязвимость Stagefright в библиотеке для отображения видео в MMS-сообщениях, которой были подвержены все версии Android, начиная с 2.2. Позже была обнаружена уязвимость в компоненте MediaServer, которой подвержены все версии Android с 2.3 до 5.1.

Ж) Интернет-источники

Android-приложения распространяются через широкое количество источников помимо официального магазина приложений. Поскольку Android-приложения написаны в основном на Java, то они легко поддаются обратной разработке и переупаковке с использованием вредоносного код. Кроме того, песочницу анализа приложений Bouncer, используемую в официальном каталоге, легко обойти. Поэтому и в самом официальном магазине содержится большое количество вредоносных программ. Кроме этого, Android поддерживает удаленную установку приложений через GooglePlay на устройства, связанные с Google-аккаунтом. Таким образом, если взломать учетную запись Google для устройства, можно установить из GooglePlay вредоносное приложение, которое туда предварительно загрузил злоумышленник. При этом на экране мобильного телефона не требуется каких-либо подтверждений этих действий, поскольку они запрашиваются в окне браузера и приложение устанавливается на телефон в фоновом режиме при получении доступа к Интернету. Также к этой категории уязвимостей относится использование социальной инженерии, когда для продолжения работы предлагают установить приложение из неавторизованного источника.

З) Уязвимости аппаратуры и связанных с ней модулей и протоколов

Мобильные устройства, работающие под управлением ОС Android, имеют широкий набор аппаратуры для взаимодействия с внешним миром. Соответствующие уязвимости можно эксплуатировать при непосредственной близости к устройству или при наличии физического доступа к устройству. Примерами таких атак служат атака типа «отказ в обслуживании» на технологию Wi-Fi Direct, кража данных кредитных карт с помощью NFC , исполнение удаленного кода через Bluetooth, установка вредоносного приложения без ведома пользователя через adb с помощью механизма бэкапов.

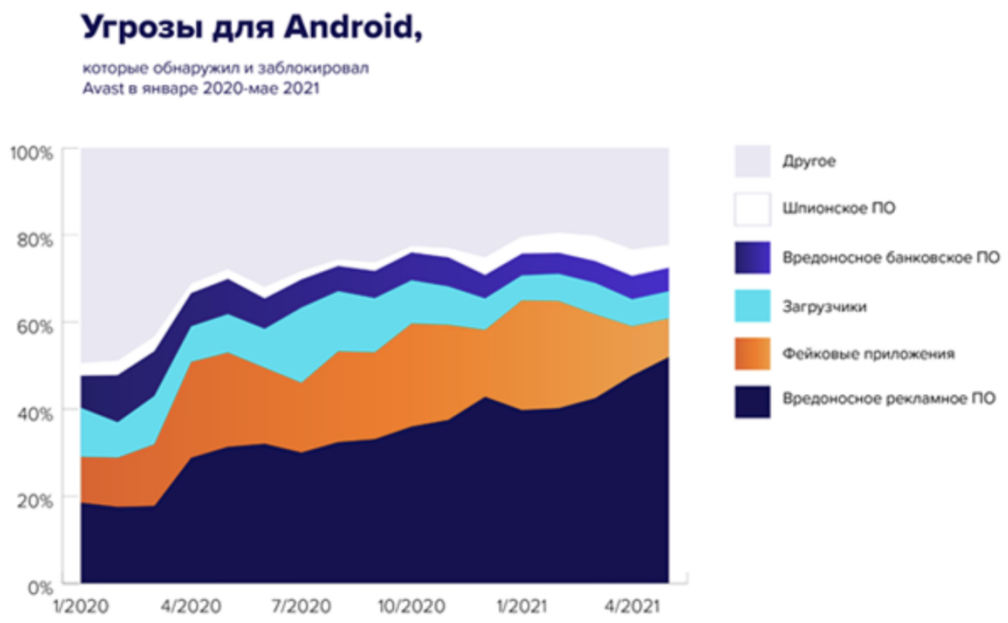

Рисунок 1 – Угрозы Android [2]

2.2 Проблемы безопасности IOS

А) Межпроцессное взаимодействие в iOS-приложениях [3]

В iOS версии 8 компания Apple представила новую технологию под названием App Extensions, с помощью которой приложения могут делиться своими функциональными возможностями с другими установленными на устройстве приложениями (например, мобильные приложения для социальных сетей позволяют быстро делиться контентом из браузера). Для организации взаимодействия между расширением (App Extension) и основным приложением (Containing App) разработчики нередко используют механизм Deep Linking. Вызов приложения при этом осуществляется посредством зарегистрированной в системе собственной схемы URL. В процессе установки основное приложение регистрирует себя в качестве обработчика схем, указанных в файле Info.plist. Подобного рода схемы не привязаны к приложению: если на устройстве присутствует вредоносное ПО, которое обрабатывает ту же схему URL, то невозможно предсказать, какое именно приложение будет запущено. Это открывает злоумышленнику возможности для проведения фишинговых атак и кражи учетных данных пользователей.

Б) Сторонние клавиатуры [3]

В 2018 году существовала такая проблема, как отсутствие ограничений на использование установленных пользователем клавиатурных расширений. Компания Apple позволила использовать клавиатуры сторонних производителей начиная с iOS версии 8, в это время такая возможность уже существовала в Android. Стоит отметить, что iOS накладывает более строгие ограничения на использование клавиатуры, чем Android; однако

Apple не может контролировать, что делают разработчики клавиатур с данными нажатия клавиш, если пользователь разрешает этим приложениям сетевое взаимодействие.

В) Проблемы в самих приложениях

Серверные части мобильных приложений в равной степени содержат уязвимости как в коде самого приложения, так и в механизмах его защиты. В числе последних стоит отметить недостатки реализации двухфакторной аутентификации. Например, если послать сразу друг за другом, с минимальным интервалом, два одинаковых запроса к серверу, то одноразовые пароли отправляются пользователю приложения на устройство и через push-уведомления, и в SMS на привязанный номер телефона. В результате злоумышленник, имея возможность перехватывать SMS-сообщения, может совершать операции от имени законного пользователя, например переводить деньги с его счета на свой.

Г) Автонабор номера телефона [4]

Изучив документацию Apple по ссылке tel можно отметить, что когда пользователь переходит по ссылке на номер телефона на странице, iOS показывает алерт, который спрашивает действительно ли пользователь хочет набрать номер телефона и инициализирует набор, если пользователь жмет по «Согласен». Когда пользователь открывает URL со ссылкой tel через установленное приложение, iOS не показывает алерт и инициализирует звонок без последующего подтверждения пользователем.

Д) Просмотр последних открытых приложений [3]

В мобильных устройствах есть возможность просмотра недавно использованных программ и быстрого переключения между ними. Для этого, когда пользователь сворачивает приложение, операционная система делает снимок состояния экрана. Прямой доступ к снимкам есть только на устройствах с административными привилегиями. Важно предусмотреть вариант, при котором на скриншотах экрана окажутся чувствительные данные; например, в случае с мобильным банком на изображение могут попасть данные платежной карты. Эти изображения могут быть похищены, например если устройство заражено вредоносным ПО.

Рисунок 2 – Доли уязвимостей разных типов [3]

2.3 Основные угрозы Android

Большинство вирусных исследовательских компаний выделяют следующие виды угроз:

1) **Adware и кликеры**. Иногда для данного вида угроз используется термин «Madware» (Mobile Adware). Основная цель этого класса ВПО – показ пользователю нерелевантной рекламы и генерирование искусственных переходов на сайты рекламодателей. С помощью «Madware» злоумышленники зарабатывают «клики» и демонстрируют оплачивающим их компаниям иллюзию интереса пользователей.

2) **Spyware** – ПО, осуществляющее кражу персональных данных или слежку за своим носителем. Фактически, мобильное устройство может превратиться в полноценный «жучок», передавая злоумышленникам данные о сетевой активности, геолокации, истории перемещений, а также фото и видеоинформацию, данные о покупках, кредитных картах и др.

3) **Дроппер** – ВПО, целью которого является скачивание другого вредоносного ПО.

4) **Вирус** – ПО, которое наносит явный вред, например, выводит из строя конкретное приложение или одну из функций устройства.

5) **Бот – агент бот-сетей**, ВПО, которое по команде C&C-сервера осуществляет требуемую злоумышленнику сетевую активность.

6) **Вредоносное банковское ПО**: как отмечает Dark Reading, количество вредоносных мобильные программ, нацеленных на сервисы онлайн-банкинга, растет: хакеры стремятся скомпрометировать пользователей, которые предпочитают вести свой бизнес, в том числе совершать денежные переводы и платежи, с мобильных устройств. В третьем квартале 2015 года было обнаружено более 1,6 миллиона инсталляционных пакетов вредоносных программ, многие из которых были предназначены для проникновения на устройства пользователей, развертывания и сбора логинов и паролей для входа в банковские системы. Затем эти данные отправлялись обратно на командный сервер злоумышленников. В третьем квартале 2015 года мобильные банковские троянцы стали самой быстрорастущей угрозой в дикой природе.

7) **Мобильные программы-вымогатели**: изначально созданные для ПК, программы-вымогатели «блокируют» важные данные пользователя, такие как документы, фотографии и видео, зашифровывали эту информацию, а затем требуют выкуп за ее расшифровку. Если выкуп не выплачивается вовремя (обычно в биткойнах) все файлы удаляются или просто блокируются и навсегда становятся недоступными для пользователя. По данным International Data Group (IDG), 74% компаний сообщили о случаях нарушения безопасности в 2015 году, причем программы-вымогатели были одной из наиболее часто встречавшихся угроз. Создатели этого типа вредоносного ПО использовали улучшенную производительность смартфонов и анонимную сеть Tor для заражения устройств и шифрования хранящиеся на нем данных.

8) **Мобильное шпионское ПО** загружается на ваше устройство как программа, отслеживает вашу активность, регистрирует ваше местоположение и изучает важную информацию, такую как имена пользователей и пароли к аккаунтам электронной почты или сайтам онлайн магазинов. Во многих случаях шпионское ПО поставляется вместе с другими считающимися безопасными программами и спокойно собирает данные в фоновом режиме. Вы даже можете не замечать его присутствия до тех пор, пока не снизится производительность устройства, или вы не запускаете на планшете или смартфоне антивирусную проверку. Как отмечает Krebs on Security, шпионское ПО теперь — это крупный бизнес: например, компания mSpy создает «легитимные» приложения для родителей или супругов, чтобы они могли «отслеживать» своих детей или партнеров. По иронии судьбы, mSpy был взломан в мае 2015 года, развенчав само понятие «безопасных» шпионских программ.

9) **Вредоносное ПО, передающееся через MMS**: производители вредоносных программ ищут способы использования текстовой коммуникации как способа доставки вредоносного ПО. Как отмечает CSO Online, уязвимость Stagefright в медиа-библиотеке

Android позволила злоумышленникам отправлять текстовое сообщение с вложенным вредоносным ПО на любой мобильный номер. Даже если пользователи не открывали вложение или не читали текст, вредоносное ПО все равно разворачивалось на устройстве и давало хакерам доступ к вашему смартфону. Проблема была быстро исправлена, но была доказана возможность текстовых сообщений как способа заражения мобильных устройств.

10) **Мобильное рекламное ПО**: рекламное ПО в своем развитии шагнуло далеко вперед от надоедливых всплывающих окон и простого сбора данных. Доход многих создателей рекламы зависит от количества кликов и загрузок. Согласно ZDNet, некоторые из них разработали специальный код «malvertising», который может заражать и запускать ваше устройство, заставляя его скачивать определенные типы рекламного ПО, которое позволяет злоумышленникам похищать личную информацию.

11) **SMS-троянцы**: киберпреступники заражают мобильные устройства, охотясь за тем, что пользователи больше всего любят в своих телефонах – текстовыми сообщениями. SMS-троянцы устраивают настоящий финансовый хаос, отправляя SMS-сообщения на премиум-номера по всему миру, в разы увеличивая телефонные счета пользователей. В 2015 году Android-устройства пользователей подверглись заражению банковским троянцем, который мог перехватывать текстовые сообщения, содержавшие финансовую информацию, а затем отправлять киберпреступникам по электронной почте копию этих сообщений, тем самым предоставляя им всю необходимые данные для проникновения в банковские аккаунты пользователей. [7]

Мобильные устройства также подвержены и традиционным атакам (например, DNS Hijacking, E-mail Phishing), так как используют те же базовые пользовательские сервисы, что и персональные ПК.

2.3.1 Примеры вредоносного мобильного ПО для Android

Рассмотрим примеры наиболее известного мобильного ВПО: «Агент Смит», Culprit, SockPuppet или Unc0ver (далее, в разделе 2.4), Ztorg, Monokle, Android/Locker.B.

Заподозрить заражение **«Агентом Смитом»** можно по заметному увеличению показа нерелевантной рекламы. Пока это единственное зафиксированное вредоносное действие этого ВПО, хотя, технически, оно имеет огромный вредоносный потенциал. Масштаб заражения Агентом Смитом – 25 млн. устройств, преимущественно, в Азии. Поведение ВПО частично напоминает работу таких вирусов, как Gooligan, Hummingbad, CopyCat. «Агент Смит» действует следующим образом:

1) Пользователь скачивает дроппер в составе зараженного приложения (бесплатной игры или приложения с возрастным цензом).

2) Дроппер проверяет наличие на мобильном устройстве популярных приложений, таких как WhatsApp, MXplayer, ShareIt.

3) Дроппер скачивает и распаковывает архив, который превращается в APK-файл, при необходимости, обновляет и заменяет легитимное популярное приложение на зараженный вариант.

Culprit – ВПО под ОС Android, представляющее собой встроенный в видеофайл код, эксплуатирующий уязвимость CVE-2019-2107 в ОС Android 7.0 до 9.0 (Nougat, Oreo, Pie). Достаточно открыть видеофайл, полученный в фишинговом MMS или сообщении из мессенджера, и ВПО получает полные права в системе.

Старый троян **Ztorg** под ОС Android после установки собирает сведения о системе и устройстве, отправляет их на командный сервер, откуда приходят файлы, позволяющие получить на устройстве права суперпользователя (Jailbreak). ВПО распространяется через зараженные приложения и рекламные баннеры.

Monokle под ОС Android и iOS – троян, позволяющий вести полноценный шпионаж за жертвой: записывать нажатия клавиатуры, фотографии и видео, получать историю интернет-перемещений, приложений социальных сетей и мессенджеров, вплоть до записи экрана в момент ввода пароля. Троян снабжен рядом эксплойтов для реализации необходимых прав в системе, распространяется, предположительно, с помощью фишинга и зараженных приложений. Первые версии ВПО появились под ОС Android, но уже появились версии для устройств Apple. [5]

Android/Locker.B – представитель семейства вредоносных программ, блокирующих доступ к операционной системе зараженного устройства. Вымогатель распространяется через форумы, специально созданные злоумышленниками, и файлообменные сервисы. Он маскируется под программу для работы с камерой в WhatsApp, антивирус для Android, мобильное приложение Dropbox или Flash Player. После установки на смартфон или планшет вредоносное приложение запрашивает права администратора устройства. Получив необходимые разрешения, малварь блокирует доступ к операционной системе, меняя PIN-код экрана блокировки. Далее Locker.B выводит на экран требование выкупа, оформленное в классическом стиле «полицейских вымогателей». Сумма выкупа варьируется – 25 или 50 долларов или евро. Интересно, что злоумышленники принимают выкуп подарочными картами iTunes и предоставляют жертвам подробную инструкцию по их покупке и использованию. Данная версия вымогателя наиболее активна в странах Латинской Америки. Тем не менее, злоумышленникам не составит труда переориентировать угрозу на другие регионы. [6]

2.4 Основные угрозы IOS [8]

Для того, чтобы выделить основные угрозы операционной IOS, обратимся к исследованию проекта по обеспечению безопасности мобильных приложений (OWASP). Впервые такое исследование было проведено в 2011 году. После этого аналогичные исследования проводились в 2014 и 2016 годах. В списках 2014 и 2016 года полностью совпадает только один пункт, в остальном же структура категорий сильно изменилась. Например, некоторые пункты были разделены, а некоторые, наоборот, объединены в один, что упростило их рассмотрение.

1. **Неправильное использование платформы.**

Данный вид уязвимостей стоит во главе списка. Независимо от того Android это или iOS, при разработке для любой из этих платформ необходимо придерживаться конкретных требований с целью обеспечения должного уровня безопасности итогового продукта. Но бывает, что при создании приложений некоторые из предписанных правил и рекомендаций непреднамеренно нарушаются или просто реализуются с ошибками. В результате возникает угроза, вызванная неверным использованием какой-либо возможности платформы или отсутствием реализации протоколов безопасности. Здесь можно упомянуть следующие случаи:

1) Ошибочное использование функции iOS Touch ID может привести к неавторизованному доступу к устройству.

2) Неверное применение iOS Keychain, вследствие чего чувствительные данные, например ключи сессии или пароли, сохраняются в локальном хранилище приложения, а не в защищенном.

3) Запрос чрезмерных или неверных разрешений платформы.

2. Небезопасное хранилище данных.

Мобильные устройства нередко теряются или крадутся, оказываясь в руках злоумышленников. Помимо этого, утечка личной информации пользователя может произойти из-за вредоносного ПО, которое позволяет атакующему задействовать уязвимости устройства. С помощью взлома или рутинга устройства можно легко обойти защиту шифрованием, поэтому разработчики ПО должны исходить из предположения, что злоумышленники могут получить доступ к файловой системе. И поскольку практически все приложения так или иначе сохраняют информацию, очень важно реализовать ее сохранение в такой области, где она не будет доступна другому приложению или пользователю.

3. Небезопасная коммуникация

Если данные передаются незашифрованными в виде чистого текста, любой, кто отслеживает данную сеть, может перехватить и прочесть их. Мобильные приложения, как правило, обмениваются данными по модели клиент-сервер. При этом процесс передачи через сеть оператора или интернет должен быть реализован безопасно. Траффик может перехватываться прокси-серверами, базовыми станциями, а также с помощью взлома WiFi или путем установки на устройство вредоносного ПО.

4. Небезопасная аутентификация

Прежде чем предоставить доступ, приложение должно проверить подлинность пользователя. Обход аутентификации обычно реализуется через существующие уязвимости, такие как неправильная проверка сервисных запросов сервером. Мобильные приложения должны проверять и удерживать подлинность пользователя, особенно в процессе передачи конфиденциальных данных, например, финансовой информации.

5. Недостаточная криптографическая стойкость

Существует два случая, в которых криптография системы может быть скомпрометирована для раскрытия чувствительных данных:

1) Слабый внутренний алгоритм шифрования/дешифрования.

2) Пробелы в реализации самого процесса криптографии.

Успешный взлом в таких случаях может быть следствием ряда факторов, включая:

- Обход встроенных алгоритмов шифрования кода.
- Неправильное управление цифровыми ключами.
- Использование пользовательских или устаревших протоколов шифрования.

6. Небезопасная авторизация

Для разных пользователей предусматриваются разные права, в результате чего одни получают стандартный доступ, в то время как другие, например администраторы, могут иметь дополнительные разрешения и привилегии. Слабые схемы авторизации, несмотря на успешную проверку подлинности пользователя, могут не справляться с проверкой его прав на доступ к запрашиваемым ресурсам. Подобный недочет позволяет хакерам авторизовываться и выполнять атаки с целью повышения привилегий.

7. Качество кода клиента

Эта категория в некотором смысле является общей причиной проблем мобильного клиента, связанных с неправильной реализацией кода. Атакующий может передавать особые входные данные в вызовы функций, провоцируя их выполнение и наблюдая за поведением приложения. В связи с этим падает производительность, повышается потребление памяти и т.п. В этом случае стоит иметь в виду, что ошибки в коде нужно

исправлять локальным способом, поскольку они возникают в мобильном клиенте и отличаются от ошибок серверной стороны.

Привести же они могут к:

- уязвимостям форматирующих строк;
- переполнению буфера;
- внедрению небезопасных сторонних библиотек;
- удаленному выполнению кода.

При разработке приложений зачастую используются сторонние библиотеки, которые сами по себе могут содержать баги и быть недостаточно протестированы. Эти нюансы находятся вне контроля разработчика, поскольку исходный код ему недоступен. В остальных же случаях чаще всего ошибки кода исправляются переписыванием его соответствующих частей.

8. Подделка кода

Иногда в магазинах встречаются поддельные версии приложений. От оригинала их отличает встроенное в исполняемый файл вредоносное содержимое, например закладка, позволяющая получать несанкционированный доступ к системе. Злоумышленники могут повторно подписывать эти поддельные приложения и размещать их в сторонних магазинах или даже напрямую доставлять жертве через фишинговые атаки.

9. Реверс-инжиниринг

Злоумышленники могут разобрать и декомпилировать приложение для анализа кода. Этот способ взлома особенно опасен, так как позволяет инспектировать, понять и изменить код, включив в него вредоносную функциональность или транслирование нежелательной рекламы. Разобравшись в принципе работы приложения, хакеры могут изменить его с помощью таких инструментов, как IDA Pro, Hopper и прочих. После реализации нужного им поведения, они могут повторно скомпилировать приложение и использовать в своих умыслах.

10. Лишняя функциональность

Иногда разработчики могут ненамеренно оставлять закладки или дополнительную функциональность, которые не очевидны для конечного пользователя. В результате продукт выпускаются в продакшен с функцией, которая не должна быть доступной, что создает дополнительные риски для безопасности приложения. Хакеры эксплуатируют подобные слабые места программ непосредственно из своих систем, не нуждаясь в содействии со стороны регулярных пользователей. Они изучают файлы конфигурации, двоичные файлы и прочие компоненты, раскрывая функционал бэкенд-части, который затем используют для совершения атак.

2.4.1 Примеры вредоносного мобильного ПО для IOS

SockPuppet или **Unc0ver** – ВПО, позволяющее получить злоумышленнику права суперпользователя для систем iOS и MacOS (Jailbreak). ВПО скачивается в составе зараженного приложения, которое определенный промежуток времени было доступно даже в официальном магазине Apple. ВПО регулярно обновляется и эксплуатирует уязвимость CVE-2019-8605, которая наследуется новыми версиями iOS. В версиях iOS 12.2 и 12.3 уязвимость была закрыта, после чего вновь появилась в версии 12.4 и была пропатчена в версии 12.4.1. [5]

3. Вывод

Я изучила основные проблемы безопасности и угрозы Android и IOS. Мной было отмечено, что в целом, как проблемы безопасности, так и угрозы у двух мобильных операционных систем, в сущности, схожи (мой вывод так же подтверждает исследование OWASP, которое не разделено на проблемы Android и проблемы IOS). Так же, мной было отмечено, что перечень актуальных уязвимостей можно найти на сайте:

- https://www.cvedetails.com/vulnerability-list/vendor_id-49/product_id-15556/Apple-Iphone-Os.html (Для IOS)
- https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/year-2022/Google-Android.html (Для Android)

Так же, при необходимости, можно изучить уязвимости конкретных мобильных приложений, отыскав их на сайтах с репортами о проведенных эксплоитах:

- https://www.exploit-db.com (Для Android)
- https://www.vulnerability-lab.com/show.php?cat=mobile (IOS)

4. Источники

[1] https://habr.com/ru/company/pt/blog/332904/

[2] https://www.tadviser.ru/index.php/Статья:Безопасность_Android

[3] https://www.ptsecurity.com/ru-ru/research/analytics/mobile-application-security-threats-and-vulnerabilities-2019/

[4] https://habr.com/ru/post/235311/

[5] https://www.securitylab.ru/analytics/501302.php

[6] https://www.anti-malware.ru/analytics/Threats_Analysis/keeping-mobile-devices-safe-from-cyber-threats

[7] https://www.kaspersky.ru/resource-center/threats/mobile

[8] https://habr.com/ru/company/ruvds/blog/537456/