Разработка модели угроз Android приложений, свойственных ошибкам разработчика

Коромыслов Кирилл Евгеньевич
студент Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича, г. Санкт-Петербург, Россия, kirill1org@gmail.com

Красов Андрей Владимирович
к.т.н., доцент, зав. кафедрой Защищенных Систем Связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича, г. Санкт-Петербург, Россия, krasov@inbox.ru

Ушаков Игорь Александрович
к.т.н., преподаватель Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича, г. Санкт-Петербург, Россия, ushakovia@gmail.com

АННОТАЦИЯ

Введение: Операционная система (далее – ОС) Android является одной из самых популярных ОС для мобильных устройств в мире и представляет из себя множество сложных процессов, например, взаимодействие приложений с ОС обновление системы, установка и обновление приложений и т.п. Любой из этих процессов может содержать в себе уязвимости, что впоследствии приведет к определенным угрозам информационной безопасности. Разработчики ОС Android регулярно работают над улучшением безопасности ОС, исправляя различные критические ошибки и изменяя системный программный интерфейс приложения (далее – API), но слабым местом в этой системе все еще остается возможность разработчиков, ввиду их некомпетентности, допускать ошибки в написании программного кода, что впоследствии приводит к появлению угроз для информационной системы. Данный тип ошибок зачастую связан с гибкостью мобильных приложений, что создает дополнительные трудности в ограничении возможностей разработчиков приложений. **Постановка задачи:** основываясь на описании работы приложений под ОС Android, классифицировать угрозы и разработать модель угроз свойственных ошибкам разработчика**. Методы:** методы анализа компонентов приложений на ОС Android, классификация уязвимостей основывающаяся на моделировании процесса взаимодействия приложения с ОС, синтез модели угроз Android приложений, свойственных ошибкам разработчика. **Результаты:** была сформирована модель угроз, свойственных ошибкам разработчика, которая описывает основные угрозы информационной безопасности для данных, используемых в мобильном приложении на ОС Android. Также угроза имеет описание сценария реализации, что позволяет разработчику легче и быстрее идентифицировать существующую угрозу, либо же не допустить ее появление при дальнейшей разработке. **Практическая значимость:** использование данной модели угроз позволит разработчикам приложений повысить качество и безопасность кода, за счет устранения потенциальных угроз и, впоследствии, сократить экономические издержки компаний, повысить репутацию ОС Android. **Обсуждение:** новизна предложенной модели состоит в том, что она имеет дополненное представление, в отличие от существующих, и описана в контексте разработки безопасных мобильных приложений.

КЛЮЧЕВЫЕ СЛОВА: Android; мобильные приложения; модель угроз; ошибки разработчиков; мобильные устройства.

Введение

В настоящее время ОС Android является самой распространенной операционной системой, поскольку насчитывает более чем 2,5 миллиарда активных устройств [1] и играет основную роль в предоставлении интернет-услуг различного мобильного форм-фактора (включая, например, телефоны, планшеты, телевизоры, устройства интернета вещей, автомобили и т.п.). Использования мобильных устройств, в наши дни, затрагивает практически все сферы жизнедеятельности человека, например, общение, потребление медиаконтента, развлечения, финансы, здравоохранения и т.д. Многие из этих приложений становятся все более требовательными к безопасности и конфиденциальности данных [2], и Android как ОС должна обеспечивать соответствующие гарантии безопасности, причем как для пользователей, так и для разработчиков. Устройство может содержать различные уязвимости, которые способны впоследствии нанести удар по персональным данным и, как следствие, привести к определенным экономическим издержкам компании, взаимодействующие с рынком при помощи мобильных приложений на ОС Android.

В первую очередь, необходимо изучить основные компоненты создания приложений, с целью дальнейшего анализа и описания информационной системы.

Android-приложение состоит из четырех видов компонентов (рис. 1). Компоненты приложений взаимодействуют друг с другом с помощью специальных механизмов, называемых Intent.

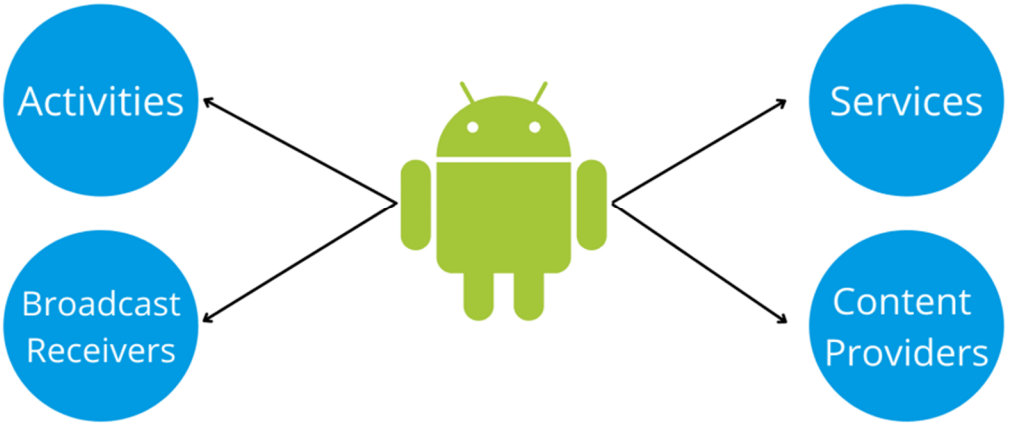

Рис. 1. Основные компоненты приложения Android

1) Активность содержит экран приложения и реализует пользовательский интерфейс. Чаще всего используется одна активность для описания одного экрана приложения, либо один компонент активности со множеством внутренних фрагментов (fragment’s) каждый из которых также реализует экран взаимодействия. Активность может открыть другую активность и передать туда данные, при помощи намерения. Во время работы пользователь может наблюдать и взаимодействовать только с одной активностью, остальные в это время находятся в стеке или уничтожаются.

2) Сервис (Service) используется для выполнения действий в фоне. В случае, когда необходимо выполнить какую-либо длительную операцию, например, загрузку файла или

отслеживание местоположения, с возможностью перейти в другое приложение на помощь приходит сервис.

3) Поставщик содержимого (Content provider) записывает данные и предоставляет доступ к данным, используя интерфейс реляционных баз данных. Каждый поставщик содержимого имеет уникальный URI для взаимодействия с данными и обрабатывает запросы на языке SQL. Поставщик содержимого применяется лишь в тех случаях, когда необходимо использовать общедоступные данные между несколькими приложениями.

4) Широковещательные приемники (Broadcast receiver) используются в роли почтовых ящиков сообщений от других приложений. Приложения Android могут отправлять или получать широковещательные сообщения из системы Android или других приложений Android, аналогично шаблону проектирования публикации и подписки и реализуется это при помощи широковещательных приемников. Эти отправляются, когда происходит интересующее пользователя событие. Например, система Android отправляет широковещательные сообщения при возникновении различных системных событий, таких как загрузка системы или начало зарядки устройства.

Схему взаимодействия приложения с устройством на ОС Android можно представить в виде схемы, изображенной на рисунке 1. Как можно заметить, это взаимодействие представляет из себя множество различных операций, таких как: обновление операционной системы, установка и обновление приложений из Play Market и сторонних источников, взаимодействия приложения с сервером, взаимодействия приложения с Android API и т.п.

Каждый из этих уровней взаимодействия может потенциально содержать в себе уязвимости, особенно это касается приложений, которые могли быть разработаны некомпетентными разработчиками.

Категории уязвимостей ОС Android

Исходя из схемы взаимодействия с устройством (рис. 2) можно сформулировать основные категории уязвимостей ОС Android.

1) Уязвимости ядра Linux и его модулей

Основная концепция уязвимостей данной категории определяется тем, что вредоносный код, использующий уязвимости в ядре Linux, может обеспечить доступ к данным пользователя или повысить права до уровня администратора системы (т.е получение root-прав).

Поскольку данная категория уязвимостей может привести к значительным последствиям, разработчики системы уделяют достаточно большое внимание усилению безопасности ядра Linux в Android (SELinux), но список уязвимостей этой категории продолжает пополняться с каждым годом.

Также к этой категории можно отнести уязвимости, которые возникают исключительно из-за использования Linux ядра в основе ОС Android.

Необходимо учитывать, что различные вендоры мобильных устройств добавляют в ядро модули для создания своих собственных оболочек ОС Android, в которых также могут содержаться уязвимости. Примеры уязвимостей повышения привилегий описаны в [3] [4].

2) Уязвимости модификаций и компонентов производителей устройств

В настоящее время ОС Android активно развивается и, будучи открытой платформой, производители смартфонов могут свободно расширять и модифицировать ее, что позволяет им отличаться от своих конкурентов и удивлять новыми преимуществами своих покупателей. Однако кастомизация производителей неизбежно влияет на общую безопасность, Поскольку эти модификации напрямую касаются ОС Android, они также

могут содержать приложения и сервисы, имеющие уязвимости, которые устанавливаются при инсталляции ОС и не могут быть удалены в дальнейшем.

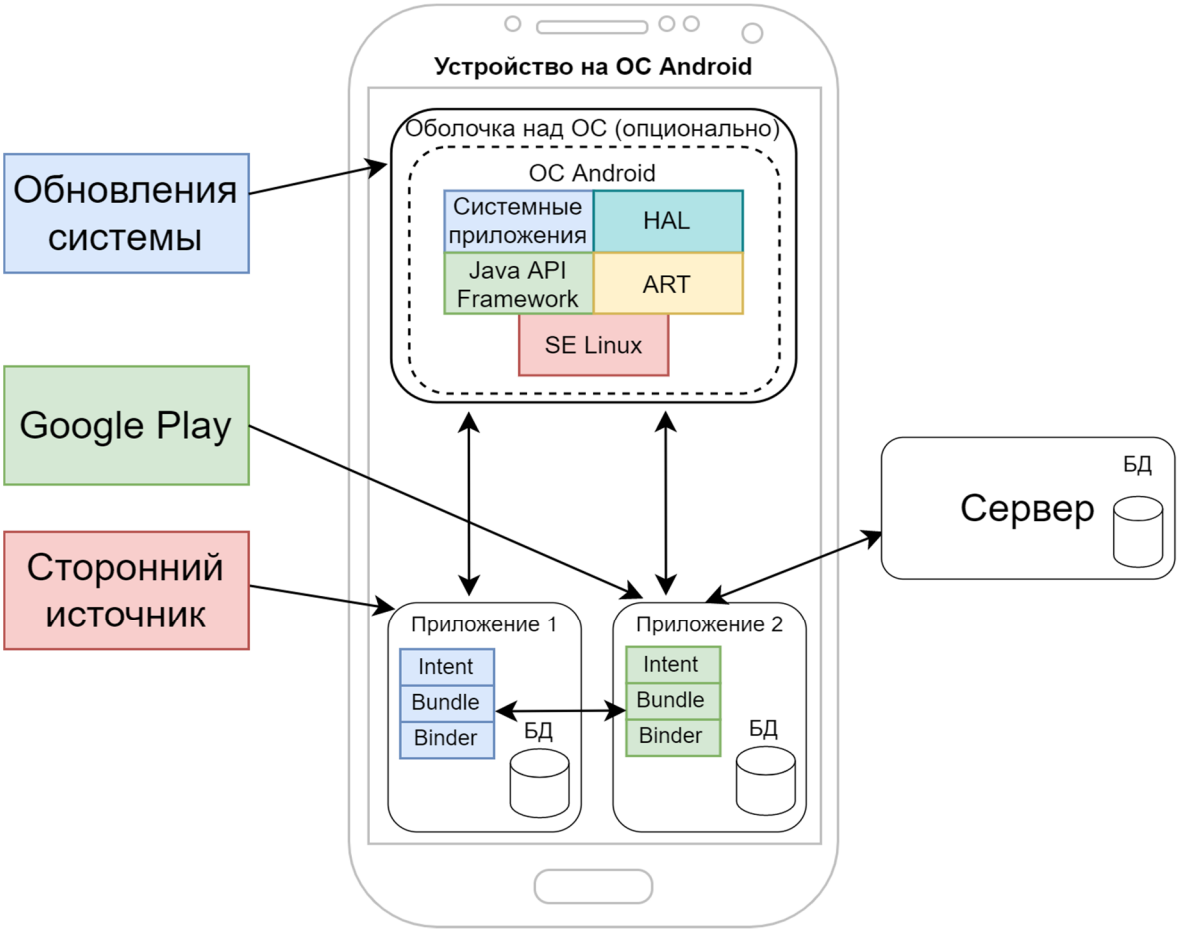

Рис. 2. Схема взаимодействия приложения с устройством на ОС Android

Анализ такого рода уязвимостей был приведенный в статье [5][6][7], где содержится информация о том, что в таких приложениях и сервисах содержится от 60 до 85% уязвимостей, обнаруженных во всей системе.

3) Уязвимости модулей в нативном коде

Приложения Android в основном написаны на языке Java и Kotlin, работающие на JVM. Как и программы на Java для настольных компьютеров, Android поддерживает Java Native Interface (JNI) и позволяет приложениям использовать нативные библиотеки. Известный факт, что в обычных условиях нативный код нарушает безопасность Java, поскольку он не соответствует правилам безопасности Java и имеет доступ ко всему адресному пространству.

Исследование [8] показывает, что в популярных приложениях Android нативные библиотеки используются достаточно часто. В настоящее время такие приложения, как правило, обеспечивают разнообразные дополнительные функциональные возможности, чтобы в полной мере использовать преимущества современных смартфонов на Android. Такие задачи, как шифрование, обработка аудио/видео потоков, обработка больших объемов данных, стали достаточно распространенными в популярных приложениях, где вероятность появления сторонних нативных библиотек значительно выше. Необходимо осознавать, что

эти сторонние нативные библиотеки пользуются всеми разрешениями, которые пользователь предоставляет приложению, в чем зачастую нет необходимости и это также может привести к появлению уязвимостей.

4) Уязвимости механизмов межпроцессного взаимодействия

В системе Android существует множество каналов связи, таких как намерение, поставщик контента или внешнее хранилище. Все эти каналы могут быть использованы злоумышленниками для осуществления атак, таких как кража широковещательных сообщений, перехват активностью, перехват сервисом, подмена намерений, повышение привилегий и атака договоренности приложений [9].

5) Уязвимости в приложениях

Любое приложение так или иначе работает с данными. Для достижения конфиденциальности и безопасности необходимо обеспечивать правильную обработку, передачу и хранения этих данных. Данная категория уязвимостей, как правило определяется ошибками разработчиков приложений, за счет их некомпетентности. Программисты, при написании приложений, могут использовать собственные или сторонние криптографические библиотеки с ошибками, не шифровать данные вовсе, использовать слабые методы защиты трафика, либо не реализовывать их вовсе, не выполнять необходимые требования для защиты авторизации и т.п. Наиболее часто встречающиеся уязвимости можно найти в рейтинге Mobile OWASP-10 [10], а также на сайте документации по написанию Android приложений [11].

Важной особенностью является то, что приложения, написанные на языке Java и Kotlin поддаются реверсинжинерингу, что впоследствии глубокого анализа приложения может привести к обнаружению уязвимостей.

6) Уязвимости в сервисах и библиотеках Android

К данному типу относятся уязвимости, которые могут содержаться в стандартных библиотеках, сервисах и службах на ОС Android. Так, например, ZimperLich, Mulliner и Plankton используют уязвимости в Dalvik с целью повышения привилегий до root или динамической подгрузки кода [12]. Еще одним примером может послужить уязвимость в механизме анализа видео из MMS-сообщений при помощи Stagefright, что может привести к доступу к файлам или запуску вредоносного кода [13].

7) Уязвимости в источниках загрузки приложений

ОС Android позволяет загружать и устанавливать приложения из различных источников, что позволяет увеличить вектор атак в этом направлении, поскольку приложения не проходят никаких дополнительных проверок по сравнению с процессом публикации на официальном источнике - интернет-магазине приложений Google Play. Поскольку, чаще всего Android-приложения можно с легкостью декомпилировать, появляется множество копий популярных приложений, реализующих вредоносный код и, тем самым, представляющих серьезную угрозу для конфиденциальных данных пользователей [14].

Поскольку Google Play для проверки использует песочницы, реализующие динамическую проверку кода, исследователями были выявлены некоторые способы сокрытия следов вредоносного кода в приложениях [15]. Также, необходимо учитывать, что Google Play поддерживает автоматическую установку приложений, находящихся в Google-аккаунте пользователя, таким образом, в случае его взлома, вредоносные приложения могут быть установлены автоматически. Получается, что нельзя назвать Google Play истинно безопасным источником приложений, учитывая вышеперечисленные ограничения, но компания Google продолжает работать над безопасностью своей платформы, и загрузка вредоносного приложения становится все менее и менее вероятным сценарием.

8) Уязвимости в аппаратных модулях и протоколах их взаимодействия с системой

Современные мобильные устройства невозможно представить без регулярного взаимодействия с другими окружающими его устройствами, будь то Bluetooth-наушники,

Wi-Fi роутер, компьютер или ноутбук и т.п. Благодаря этому мобильные устройства на ОС Android поддерживают большое количество протоколов, для взаимодействия со сторонними устройствами. Данные протоколы, несовершенны и также имеют определенные уязвимости, которые ограничены требованиями той или иной технологии.

В качестве примера, можно привести уязвимость в механизме NFC, которая позволяет выполнить несанкционированный платеж, либо получить данные банковской карты [16]; уязвимость в механизме Bluethooth, позволяющее удаленно выполнить зловредный код [17]

Модель угроз ошибок разработчика

Уязвимости свойственные межпроцессному взаимодействию (пункт 4) и уязвимости содержащиеся в самих приложениях (пункт 5), являются наиболее популярными угрозами в мобильных приложениях и требуют к себе повышенного внимания. Они появляются ввиду большого количества некомпетентных разработчиков, а также сложности ограничений их возможностей, поскольку это достигается ценой снижения гибкости и вариативности разработки.

Необходимость создания модели угроз также предписывается несколькими значимыми документами [18][19][20][21][22], в которых выдвигается рекомендации и требования ее определения с целью дальнейшего обеспечения безопасности и конфиденциальности данных.

Далее, в таблице 1, будут описаны основные угрозы ошибок разработчика, а также сценарии их реализации. Угрозы, их описание и сценарии реализации сформулированы на основе описания особенностей и принципов действия элементов приложения, а также пользовательском опыте и некоторых существующих уязвимостях.

Таблица 1

Модель угроз ошибок разработчика

№	Угроза	Описание угрозы	Сценарии реализации
1. Угрозы межкомпонентного взаимодействия			
1	Обработка потенциально конфиденциальных данных неявным или нежелательным приложением. (УБИ.102)	Данная угроза возникает в случаях, когда реализуется передача потенциально конфиденциальных данных между приложениями или компонентами, что может привести к утечке этих данных, поскольку принимающая сторона может обрабатывать такие данные определенным, нежелательным способом.	1) Обработка данных, вложенных в намерение (intent) осуществляется при помощи выбранного ранее или неизвестного приложения. 2) Использование флага “android:exported = true” по-умолчанию для всех компонентов приложения.
2	Использование привилегий приложения сторонним приложением,	Данная угроза возникает в случаях предоставления привилегий сторонним приложениям, через подключаемый компонент,	1) Использование компонентов, разрешающих межкомпонентное взаимодействие (android:exported = true), которое, в тоже время, может предоставлять доступ к

	посредством механизма IPC. (УБИ.007, УБИ.102, УБИ.184)	например, при использовании механизма IPC.	конфиденциальной информации.
2. Угрозы динамической загрузки кода			
1	Запуск зловредного кода при помощи динамически подгружаемого кода. (УБИ.006, УБИ.023)	Данная угроза возникает в случаях, когда разработчик не контролирует источник загрузки динамически подгружаемого кода, что впоследствии может привести к подмене легального кода зловредным, путем его загрузки и запуска вместо ожидаемого.	1) Динамическая загрузка кода из файла. (Расположенного в хранилище, либо ресурсах приложения).
			2) Динамическая загрузка кода из сети Интернет (например, по конкретному URL).
3. Сетевые угрозы			
1	Утечка потенциально конфиденциальных данных при передаче их в общедоступном формате. (УБИ.034, УБИ.069, УБИ.116)	Данная угроза возникает в случае передачи конфиденциальных данных в стандартном их представлении, либо в незашифрованном трафике в сети Интернет или локальных сетях.	1) Использование протокола HTTP вместо HTTPS, по причине невнимательности. Данный сценарий возможен при указании неверного URL, в момент конфигурации Retrofit, либо при открытии HttpURLConnection, при условии, что серверная часть поддерживает оба протокола.
			2) Передача данных в стандартном их представлении, без применения шифрования и хеширования.
2	Утечка авторизационных данных пользователей. (УБИ.034)	Данная угроза возможна в случае компрометации авторизационных данных пользователей, в момент их передачи.	1) Регулярная отправка авторизационных данных (например, логина и пароля) напрямую в сетевых запросах, с целью авторизации.
4. Угрозы компонента WebView			
1	Реализация зловредных сценариев при помощи посещения нежелательных URL. (УБИ.145, УБИ.167, УБИ.175)	Данная угроза возникает в случае, когда контент, получение доступа к которому осуществляется посредством компонента WebView, приводит к реализации зловредных, либо нежелательных сценариев.	1) Предоставление доступа к зловредным данным, например, фишинговому сайту.
2	Реализация	Данная угроза возникает в	1) Включение параметра WebView

	межсайтового скриптинга (XSS) (УБИ.041)	случае использования компонента WebView при определенной его конфигурации, что впоследствии может приводить к реализации межсайтового скриптинга.	- “setJavaScriptEnabled()” по умолчанию.
3	Загрузка и запуск вредоносного кода из web файлов. (УБИ.006, УБИ.012, УБИ.023)	Данная угроза возникает в случае использования компонента WebView при определенной его конфигурации, что впоследствии позволит выполнить потенциально зловредный код, находящийся в открываемом при помощи WebView файле.	1) Включение параметров: webViewSettings.setAllowFileAccess(true); webViewSettings.setAllowContentAccess(true); (true) при конфигурации WebView.
4	Получения доступа к методам системы Android посредством компонента WebView. УБИ.006, УБИ.012, УБИ.023)	Данная угроза возникает в случае доступа к нежелательному источнику при использовании небезопасной настройки компонента WebView, что позволяет вызывать методы в системе Android.	1) Включение параметра: addJavaScriptInterface() при конфигурации WebView.
5	Утечка конфиденциальных данных пользователя при работе с WebView. (УБИ.197)	Данная угроза возникает в случае компрометации конфиденциальных данных после работы с компонентом WebView.	1) Получение доступа или отправка пользователем его конфиденциальных данных с возможностью их кэширования
5. Угрозы превышения привилегий			
1	Получение доступа к конфиденциальным данным, которые не используются приложением. (УБИ.007, УБИ.184)	Данная угроза возникает в случае предоставления доступа к конфиденциальным данным пользователей посредством механизма запроса разрешений, которые впоследствии не используются приложением, но могут выступать источником их утечки.	1) Добавление различных разрешений с целью обеспечения масштабируемости приложения.
			2) Избегание удаления ненужных более разрешений (в файле Manifest.xml)
2	Злоупотребление привилегиями приложения.	Данная угроза возникает в случае, если выполняется запрос разрешений, цель	1) Выдача разрешений, которые потенциально можно заменить на разрешения на основе подписи

	(УБИ.007, УБИ.184)	выдачи которых можно заменить другим функционалом. Вместе с получением привилегий, предоставляемых разрешениями, появляется дополнительная вероятность их утечки.	(signature permission), в случае разработки подконтрольных приложений.
			2) Выдача разрешений, которые потенциально можно заменить на обращения к приложениям, предоставляющим необходимые данные при помощи намерений (Intent'ов), например, доступ к контактам пользователя.
			3) Предоставление сразу всех привилегий поставщику содержимого (ContentProvider).
6. Угрозы хранения данных			
1	Утечка потенциально конфиденциальных данных при хранении их в общедоступном формате.	Данная угроза возникает в случае, когда разработчики используют общедоступные форматы для хранения потенциально конфиденциальных данных, что впоследствии может привести к их утечке.	1) Хранение потенциально конфиденциальных данных в общедоступном, незашифрованном файле.
			2) Хранение потенциально конфиденциальных данных в SharedPreferences используя общедоступный режим.
			3) Хранение потенциально конфиденциальных данных в стандартном их представлении, без применения шифрования и хеширования.
2	Реализация SQL-инъекций при помощи поставщика содержимого (ContentProvider).	Данная угроза возникает в случаях, когда в момент доступа к данным, при помощи поставщика содержимого, появляется возможность реализовать SQL-инъекцию.	1) Использование не параметризированных методов для доступа к данным поставщика содержимого (query(), update (), delete ())
7. Угрозы пользовательского интерфейса			
1	Утечка конфиденциальных данных пользователей при разблокированном устройстве.	Данная угроза возникает в случаях, когда злоумышленник может получить доступ к потенциально конфиденциальным данным в приложении, при доступе к разблокированному устройству.	1) Предоставление доступа к конфиденциальным данным в приложении без дополнительной аутентификации (использование PIN-code или биометрические данные)

2	Утечка конфиденциальных данных пользователей при использовании некорректной настройки компонентов представления. (УБИ.067)	Данная угроза возникает в случаях некорректной настройки компонентов представления (view's), которые способствуют утечке конфиденциальных данных пользователя при их использовании.	1) Отображение конфиденциальных данных при использовании некорректного типа поля ввода (EditText). Например, использование android:inputType="textPersonName " для ввода пароля.
3	Утечка конфиденциальных данных пользователей при использовании функций записи экрана. (УБИ.067, УБИ.086, УБИ.115)	Данные угрозы возникают в случае использования функции записи экрана в моментах, когда пользователь взаимодействует с конфиденциальными данными.	1) Выполнение записи экрана в момент ввода пользователем пароля в поле ввода.
			2) Выполнение записи экрана в момент отображения конфиденциальной информации, например, данных банковского счета.
8. Угрозы связанные с инструментами разработки и библиотеками			
1	Реализация атак, характерных для устаревших версий инструментов разработки и библиотек. (УБИ.192)	Данные угрозы возникают в случае отказа от поддержки новых версий инструментов разработки и библиотек, с помощью которых разрабатывается проект, что приводит к возможности реализации уязвимостей, которые были исправлены в новых версиях.	1) Игнорирование запросов обновления зависимостей для поддержки актуальных версий библиотек и инструментов разработки.
2	Реализация атак, характерных для непопулярных или самописных инструментов разработки и библиотек. (УБИ.003, УБИ.008, УБИ.012)	Данные угрозы возникают в случае использования непопулярных или самописных инструментов разработки и библиотек, которые могут содержать в себе уязвимости.	1) Использование непопулярных или самописных библиотек в проекте.
9. Угрозы обратной разработки			
1	Использование исходного кода приложения с целью создания копии с	Данные угрозы возникают в случае получения исходного кода приложения в его исходном состоянии злоумышленниками с целью	1) Получение доступа к исходному коду приложения при помощи инструментов обратной разработки.

	добавлением злонамеренного кода. (УБИ.036)	дальнейшего изменения и добавления злонамеренного кода, позволяющего достичь их цели.	

Большинство вышеперечисленных угроз может привести к утечке конфиденциальных данных пользователей, что впоследствии создаст недоверие пользователей данному приложению и приведет к снижению экономической прибыли компании.

Заключение

ОС Android реализует сложнейшие архитектурные принципы и механизмы, которые впоследствии могут приводить к уязвимостям на разных уровнях.

Уязвимости в Android приложениях, ввиду сложности и гибкости ОС достаточно частые явления. Для того, чтобы избежать нарушения целостности, конфиденциальности и доступности данных разработчикам ОС Android и разработчикам приложений под ОС Android необходимо прикладывать значительные усилия.

Разработчики приложений под ОС Android могут пользоваться БДУ ФСТЭК [23], сборником популярных уязвимостей OWASP-Mobile 10 [10], либо документацией [11]. С целью объединения информации и предоставления ее в более наглядном и доступном виде в данной статье была разработана модель угроз ошибок разработчика.

Использование данной модели позволит разработчикам приложений снизить порог вхождения в использование модели угроз, а также сократить время исправления существующих в приложении уязвимостей, что впоследствии приведет к сокращению нарушений безопасности информации, улучшить качество разработки приложений под ОС Android, сократить экономические издержки компаний, повысить репутацию ОС Android в целом.

Поскольку смартфоны на ОС Android регулярно совершенствуются как на программном, так и на аппаратном уровне, необходимо также поддерживать в актуальном состоянии документы, регламентирующие эти угрозы, с целью анализа и устранения этих уязвимостей разработчиками.

Литература

1. Android Security 2017 Year in Review, March 2018.[Электронный ресурс] // URL: : https://source.android.com/security/reports/Google_Android_Security_2017_Report_Final.pdf (дата обращения: 21.05.2022)

2. Андрианов В. И. Инновационное управление рисками информационной безопасности : учебное пособие / *В. И. Андрианов, А. В. Красов, В. А. Липатников ; В. И. Андрианов, А. В. Красов, В. А. Липатников* ; Федеральное агентство связи, Федеральное гос. образовательное бюджетное учреждение высш. проф. образования "Санкт-Петербургский гос. ун-т телекоммуникаций им. проф. М. А. Бонч-Бруевича". – Санкт-Петербург : СПбГУТ, 2012. – 396 с. – ISBN 978-5-91891-092-4. – EDN QSMDNH.

3. *Hei X., Du X., Lin S.* Two vulnerabilities in Android OS kernel // Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013. P. 6123—6127.

4. *Zhou X. et al.* Identity, location, disease and more: Inferring your secrets from android public resources // Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013. P. 1017—1028.

5. *Wu L. et al.* The impact of vendor customizations on android security //Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. – 2013. – С. 623-634.

6. *Zhou X. et al.* The peril of fragmentation: Security hazards in android device driver customizations //2014 IEEE Symposium on Security and Privacy. – IEEE, 2014. – С. 409-423.

7. Stagefright (bug). [Электронный ресурс] // URL: https://en.wikipedia.org/wiki/Stagefright_(bug) (Дата обращения: 17.04.2022)

8. *Sun M., Tan G.* Nativeguard: Protecting android applications from third-party native libraries //Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks. – 2014. – С. 165-176.

9. *Wang J., Wu H.* Android Inter-App Communication Threats, Solutions, and Challenges //arXiv preprint arXiv:1803.05039. – 2018.

10. OWASP Mobile Top 10. [Электронный ресурс] // URL: https://owasp.org/www-project-mobile-top-10/ (Дата обращения: 18.04.2022)

11. Developer Guides. [Электронный ресурс] // URL: https://developer.android.com/guide (Дата обращения: 21.04.2022)

12. *Shabtai A., Mimran D., Elovici Y.* Evaluation of security solutions for Android systems //arXiv preprint arXiv:1502.04870. – 2015.

13. Android Stagefright contains multiple vulnerabilities. [Электронный ресурс] // URL:https://kb.cert.org/vuls/id/924951 (Дата обращения: 25.04.2022)

14. *Zhou Y. et al.* Hey, you, get off of my market: detecting malicious apps in official and alternative android markets //NDSS. – 2012. – Т. 25. – №. 4. – С. 50-52.

15. *Petsas T. et al.* Rage against the virtual machine: hindering dynamic analysis of android malware //Proceedings of the seventh european workshop on system security. – 2014. – С. 1-6.

16. NFC attack can steal your credit card information [Электронный ресурс] // URL: http://securityaffairs.co/wordpress/37667/hacking/nfc-attack-credit-card.html (Дата обращения: 27.04.2022)

17. (Mobile Pwn2Own) Google Android Bluetooth Forced Pairing Vulnerability [Электронный ресурс] // URL: http://www.aperturelabs.com/pdfs/1%20Mobile%20Pwn2Own%202-5-15.pdf (Дата обращения: 27.04.2022)

18. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ. [Электронный ресурс] // URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (Дата обращения: 20.05.2022)

19. Приказ ФСТЭК России от 18 февраля 2013 г. N 21. [Электронный ресурс] // URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/691-prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21 (Дата обращения: 23.05.2022)

20. Приказ ФСТЭК России от 11 февраля 2013 г. N 17. [Электронный ресурс] // URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/702-prikaz-fstek-rossii-ot-11-fevralya-2013-g-n-17 (Дата обращения: 25.05.2022)

21. Приказ ФСТЭК России от 14 марта 2014 г. N 31. [Электронный ресурс] // URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/868-prikaz-fstek-rossii-ot-14-marta-2014-g-n-3 (Дата обращения: 27.05.2022)

22. Приказ ФСТЭК России от 25 декабря 2017 г. N 239. [Электронный ресурс] // URL: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kii/288-prikazy/1592-prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239 (Дата обращения: 28.05.2022)

23. Банк данных угроз безопасности информации [Электронный ресурс] // URL: https://bdu.fstec.ru/threat (Дата обращения: 6.06.2022)

DEVELOPING A THREAT MODEL FOR ANDROID APPS INHERENT TO DEVELOPER BUGS

KIRILL E. KOROMYSLOV
Student
St. Petersburg, Russia, kirill1org@gmail.com

ANDREY V. KRASOV,
Candidate of Engineering Sciences, Associate Professor
St-Petersburg, Russia, krasov@inbox.ru

IGOR A. USHAKOV
Candidate of Engineering Sciences, Lecturer
St-Petersburg, Russia, ushakovia@gmail.com

ABSTRACT

Introduction: the Android operating system (hereinafter referred to as OS) is one of the most popular operating systems for mobile devices in the world and involves many complex processes, such as the interaction of applications with the OS system updates, installation and update of applications, etc. Each of these processes can contain vulnerabilities, which subsequently lead to information security risks. Android OS developers are regularly working on improving the OS security, fixing various critical bugs and changing the application system programming interface (hereinafter - API), but the weak point of this system is still the possibility for developers, due to their incompetence, to make mistakes in writing program code, which subsequently leads to the appearance of threats to the information system. This type of error is often related to the flexibility of mobile applications, which creates additional difficulties in limiting the capabilities of application developers. **Problem statement:** based on the description of the Android applications, classify threats and develop a model of threats peculiar to developer's mistakes. **Methods:** methods for analyzing the components of applications on the Android OS, vulnerability classification based on modeling the process of interaction between the application and the OS, the synthesis of the threat model of Android applications, inherent in the developer's mistakes. **Results:** a model of threats inherent in the developer's mistakes was formed, which describes the main threats to information security for the data used in the mobile application on the Android OS. The threat also has a description of the implementation scenario, which allows the developer to more easily and quickly identify the existing threat, or prevent its occurrence during further development.. **Practical significance:** the use of this threat model will allow application developers to improve code quality and security by eliminating potential threats and, subsequently, reduce the economic costs of companies, increasing the reputation of the Android OS. **Discussion:** the novelty of the proposed model is that it has an augmented representation, unlike existing ones, and is described in the context of secure Android mobile application development..

Keywords: Android; mobile applications; threat model; developer errors; mobile devices.

REFERENCES

1. Android Security 2017 Year in Review, March 2018. // URL: : https://source.android.com/security /reports/Google_Android_Security_2017_Report_Final.pdf (date of access: 21.05.2022)

2. Andrianov V.I. Innovative information security risk management : tutorial / V.I.Andrianov, A.V. Krasov, V.A. Lipatnikov; Federal Agency of Communications, Federal State Educational Budgetary Institution of Higher Professional Education "Saint-Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruevich". - Saint-Petersburg : Saint-Petersburg State University of Telecommunications. 2012. – 396 pp. – ISBN 978-5-91891-092-4. – EDN QSMDNH.

3. Hei X., Du X., Lin S. Two vulnerabilities in Android OS kernel // Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013. P. 6123—6127.

4. Zhou X. et al. Identity, location, disease and more: Inferring your secrets from android public resources // Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013. P. 1017—1028.

5. Wu L. et al. The impact of vendor customizations on android security //Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. – 2013. – С. 623-634.

6. Zhou X. et al. The peril of fragmentation: Security hazards in android device driver customizations //2014 IEEE Symposium on Security and Privacy. – IEEE, 2014. – С. 409-423.

7. Stagefright (bug). // URL: https://en.wikipedia.org/wiki/Stagefright_(bug) (date of access: 17.04.2022)

8. Sun M., Tan G. Nativeguard: Protecting android applications from third-party native libraries //Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks. – 2014. – С. 165-176.

9. Wang J., Wu H. Android Inter-App Communication Threats, Solutions, and Challenges //arXiv preprint arXiv:1803.05039. – 2018.

10. OWASP Mobile Top 10. // URL: https://owasp.org/www-project-mobile-top-10/ (date of access: 18.04.2022)

11. Developer Guides.// URL: https://developer.android.com/guide (date of access: 21.04.2022)

12. Shabtai A., Mimran D., Elovici Y. Evaluation of security solutions for Android systems //arXiv preprint arXiv:1502.04870. – 2015.

13. Android Stagefright contains multiple vulnerabilities. // URL:https://kb.cert.org/vuls/id/924951 (date of access: 25.04.2022)

14. Zhou Y. et al. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets //NDSS. – 2012. – Т. 25. – №. 4. – С. 50-52.

15. Petsas T. et al. Rage against the virtual machine: hindering dynamic analysis of android malware //Proceedings of the seventh european workshop on system security. – 2014. – С. 1-6.

16. NFC attack can steal your credit card information // URL: http://securityaffairs.co/wordpress/37667/hacking/nfc-attack-credit-card.html (date of access: 27.04.2022)

17. (Mobile Pwn2Own) Google Android Bluetooth Forced Pairing Vulnerability // URL: http://www.aperturelabs.com/pdfs/1%20Mobile%20Pwn2Own%202-5-15.pdf (date of access: 27.04.2022)

18. Federal Law "On Personal Data" of 27.07.2006 N 152-FZ. // URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (date of access: 20.05.2022)

19. FSTEC Order No. 21 of February 18, 2013.// URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/691-prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21 (date of access: 23.05.2022)

20. Order of FSTEC of Russia dated February 11, 2013 N 17.// URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/702-prikaz-fstek-rossii-ot-11-fevralya-2013-g-n-17 (date of access: 25.05.2022)

21. Order of FSTEC of Russia dated March 14, 2014 N 31. // URL: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/868-prikaz-fstek-rossii-ot-14-marta-2014-g-n-3 (date of access: 27.05.2022)

22. Order of the FSTEC of Russia dated December 25, 2017. N 239.// URL: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kii/288-prikazy/1592-prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239 (date of access: 28.05.2022)

23. Data bank of information security threats // URL: https://bdu.fstec.ru/threat (date of access: 6.06.2022)