Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО "МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ"

(национальный исследовательский университет)

Кафедра 402

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КУРСОВАЯ РАБОТА**

" Управление информационной безопасностью в мобильных устройствах на платформе Google Android"

Выполнил: Студент группы М4О-213Б-19

\_\_\_\_\_\_ Чувикин П.Н.

Преподаватель:

\_\_\_\_\_\_ Бритвин Н.В.

Москва 2021 г.

Содержание:

*Оглавление*

[Введение 3](#_Toc72864714)

[Глава 1. Структура ОС Android. Особенности работы с безопасностью. 4](#_Toc72864715)

[Организация безопасной ОС 4](#_Toc72864716)

[Защита скачиваемых файлов 5](#_Toc72864717)

[Безопасность устройства 5](#_Toc72864718)

[Криптография 6](#_Toc72864719)

[Защита NFC 6](#_Toc72864720)

[Разграничение пользователей 6](#_Toc72864721)

[Разрешение доступа 6](#_Toc72864722)

[Безопасность, регулируемая пользователем 7](#_Toc72864723)

[Глава 2. Угрозы безопасности ОС Андроид 8](#_Toc72864724)

[Атака с получение физического доступа 8](#_Toc72864725)

[Вредоносное приложение 8](#_Toc72864726)

[Атакующий в канале связи 9](#_Toc72864727)

[Удаленные атаки 10](#_Toc72864728)

[Атаки на серверную часть 10](#_Toc72864729)

[Сценарии работы системы безопасности в андроид 10](#_Toc72864730)

[Глава 3. Организация работы и управления процессами в Android Binder 12](#_Toc72864731)

[Реализация системы управления безопасностью. Проверка работы ОС при определённых ситуациях. Улучшение работы системы при разработке. 14](#_Toc72864732)

[Последовательность работы системы при конкретном сценарии 14](#_Toc72864733)

[Настройка системы безопасности для приложения 16](#_Toc72864734)

[Заключение 19](#_Toc72864735)

[Список использованных источников 20](#_Toc72864736)

Введение

Компания Android Inc была основана в 2003 году. Первичная задумка операционной системы Android была об упрощении работы с камерами. Но, в дальнейшем, в связи с приоритетами рынка, командой разработчиков было решено попытаться составить конкуренцию в сфере смартфонов крупным компаниям как “Sumbian” и “windows mobile”.  
Первая версия Android как ОС была представлена 23 сентября 2008 года. В котором было представлено сразу несколько, уже привычных сегодня нововведений.[1] По сей день данная ОС регулярно обновляется и поддерживается. Конечно же, в связи с быстрым ростом популярности и спроса на рынке, на смартфоны с системой Android возлагались всё новые стандарты, запросы. Ведь, с развитием новых функций растёт и количество данных, проходящих, хранящихся, на одном устройстве. А уже эти данные нужно защищать совершенно по-разному, в связи с отличаем функциональности различных блоков, вшитых в современные смартфоны.

Глава 1. Структура ОС Android. Особенности работы с безопасностью.

Операционная система, рассчитанная на смартфоны, имеет в себе ряд уже обыденных вещей, находящихся в комплектации современного смартфона, но отсюда возникают и ограничения. В большинстве случаев, смартфоны подразумеваются для работы одного пользователя и являются индивидуальными. [2] [3]

Организация безопасной ОС

Один из основных видов построения защищённых операционных систем - комплексный. Конкретно о нем и стоит поговорить, если мы рассматриваем систему безопасности ОС Андроид. Так какие же плюсы у комплексного подхода и чем он отличается от другого подхода (фрагментарного).  
При комплексном подходе в ходе организации защиты защитные функции вносятся в операционную систему ещё на этапе проектирования ОС и являются ее неотъемлемой частью. Элементы подсистемы тесно взаимодействуют друг с другом при решении различных задач, связанных с ИБ. В основном, подсистема защиты создаётся так, чтобы можно было заменять отдельные ее элементы на другие для обеспечения взаимодействия с другими элементами подсистемы защиты.  
Когда как при фрагментарном подходе организуется поочерёдная защита от конкретных угроз. И самое трудное в этом подходе - совместить организацию различных программ, объединить все в одну систему. При данном подходе существует минус в том, что элементы системы не действуют связанно и при нарушении работы одного элемента безопасности, остальные будут работать в штатном режиме. [4]

*Конфигурация Android*

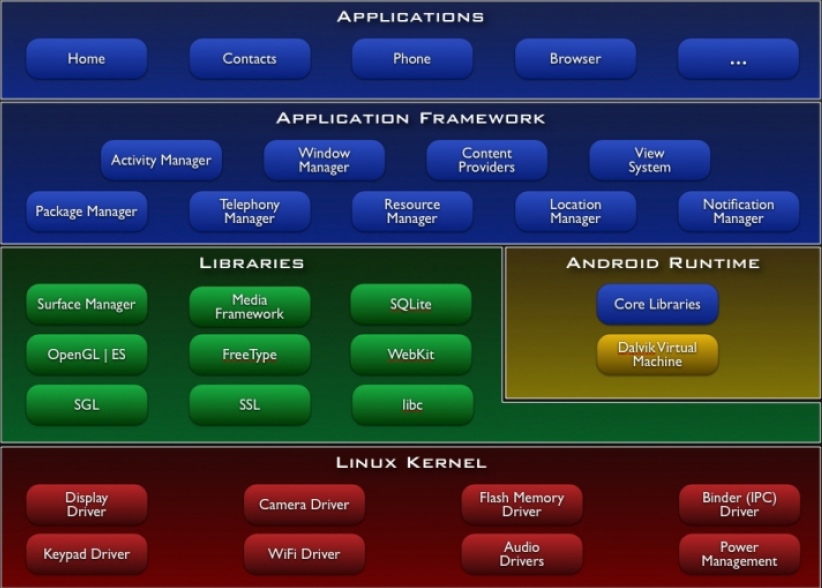
Все версии разработаны на ядре Linux, но по сравнению с ним, Android релизы урезаны по сравнению с версиями Linux 2.6. Основные действия, такие как управление памятью, процессами и пользователями обеспечивается ядром Linux. Приложения для Android разработаны на Java и выполняются на виртуальной машине под названием Dalvik VM. Они поддерживаются платформой приложения, которая предоставляет часто используемые функции через интерфейс. Различные библиотеки позволяют приложениям реализовывать графику, шифрование связи или баз данных. Стандартная библиотека («Bionic») - библиотека, созданная на основе BSD для страиваемых устройств. [2]  


Рисунок 1. Архитектура Андроид

Защита скачиваемых файлов

Обеспечение защиты ОС от возможных заражённых или вредоносных файлов, скачивающихся на смартфон, обеспечивается несколькими функциями.  
Во-первых, большинство пользователей Android используют официальное приложение "Google Play Store". Загруженные туда пользователями приложения проверяются на наличие вредоносных программ. Что обеспечивает дополнительную безопасность.[3]  
Во-вторых, существует встроенная проверка скаченного файла. Если проверка находит вероятную угрозу для дальнейшей работы устройства, то оно предупредит вас об этом.

Безопасность устройства

Различие методов обеспечения безопасности устройства но ОС Android заключается в разнообразии как технических способностей устройства, так и специфики конфигурации устройства.  
Существует как логинизация через биометрию, так и через множество различных комбинированных паролей.[3]  
К биометрии можно отнести: отпечаток пальца, сканирование лица, сканирование радужной оболочки глаза.  
К паролям: ПИН-код, пароль, графический ключ.  
Вход по данным биометрии является более быстрым и удобным(не нужно вспоминать пароль), но, при этом, он более дорогой в плане необходимого считывающего оборудования и кодировки считанной информации.

Криптография

Андроид включает различные методы криптографии: хеширование, симметричное и ассиметричное шифрование и ключи аутентификации. Данные методы могут комбинироваться и обеспечивать безопасную передачу данных.  
Но всё же, лучшим решением обеспечения безопасности передачи - использование протоколов, предназначенных для сохранения конфиденциальности и целостности данных. Наиболее широко использующийся - Secure Sockets Layer (SSL) и Transport Layer Security (TLS). Андроид поддерживает данные протоколы используя стандарт Java Secure Socket Extension (JSSE) [3]

Защита NFC

NFC - это технология которая позволяет устройству на близком расстоянии обмениваться информацией с другим устройством по средствам коротких радиоволн.  
Защита NFC может быть как физическая защита, так и защита самой системой, которая позволяет проверять доступ пользователя к использованию NFC. Так и система позволяет кодировать информацию, передаваемую от устройства к устройству. [3]

Разграничение пользователей

Помимо основного(первого) пользователя смартфона на ОС Android, можно создавать дополнительных пользователей.  
Все права и связи между пользователями могут настраиваться вручную. За исключением работы с блокировкой устройства и работой с SIM, у второго пользователя равные права. Для второго пользователя выделяется отдельная часть в памяти, где будет хранится уже его информация. Переход между пользователями защищён паролем. [2]

Разрешение доступа

Разрешения назначаются каждому приложению (определяемому уникальным именем пакета) во время установки службой диспетчера пакетов системы. Диспетчер пакетов поддерживает центральную базу данных установленных пакетов, как предустановленных, так и установленных пользователем, с информацией о пути установки, версии, сертификате подписи и назначенных разрешениях для каждого пакета, а также список всех разрешений, определенных на устройстве. [3]  
  
4 вида разрешений:  
-нормальное (Стандартное значение. Не несёт угроз другим приложениям или системе)  
  
-опасное (Разрешения с опасным уровнем защиты предоставляют доступ к пользовательским данным, или некоторой форме контроля над устройством.)  
  
-подписанное (Разрешение предоставляется только тем приложениям, которые зашифрованы тем же ключом, что и приложение, объявившее разрешение.)  
  
-системное (Компромисс для предустановленных приложений. Что-то среднее между нормальным и системным разрешением)

Безопасность, регулируемая пользователем

В приложениях, к которым нет доверия со стороны пользователя, можно лишний раз остеречься от сбора информации, о которое соглашения не было дано. Данная функция существует в ОС отдельно для каждого приложения. Вы можете отключить, например, передачу данных в этом приложении.

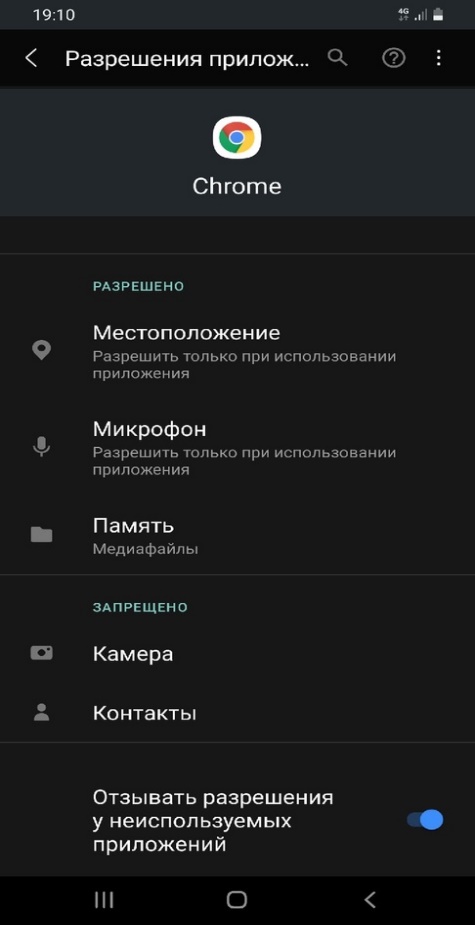


Рисунок 2 Разрешения приложения.

Глава 2. Угрозы безопасности ОС Андроид

Возможные векторы атак на ОС Андроид  
-Атака с получение физического доступа  
-Вредоносное приложение на устройстве  
-Атакующий в канале связи  
-Удаленные атаки  
-Атаки на серверную часть

Атака с получение физического доступа

Как правило, они подразумевают доступ человека напрямую к смартфону: это происходит в том случае, если устройство украли, владелец его потерял или отнес в сервис. Однако есть и достаточно необычный способ атаки, для которого используется вредоносная зарядная станция. [5]

Вредоносное приложение

Атаки с помощью вредоносных приложений  
Есть несколько источников таких приложений:  
-Официальный магазины ОС — Google Play. Редко, но встречаются вредоносные приложения  
-Неофициальные сайты и магазины приложений (third-party appstore). Для Android-устройств достаточно разрешить установку из недостоверных источников, а затем скачать apk-файл приложения с сайта.  
-Существует возможность загрузки приложения при переходе по ссылке — механизм Google Play Instant. [5]

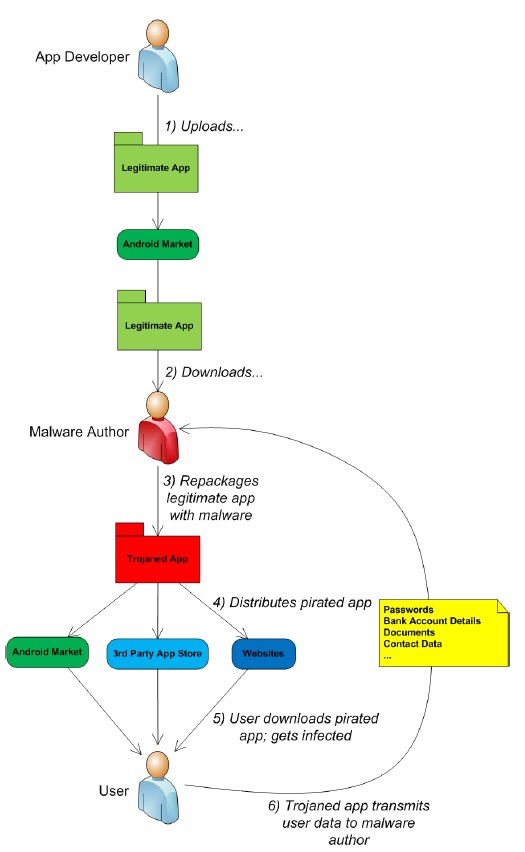


Рисунок 3. Процесс репакинга

Атакующий в канале связи

Для того чтобы злоумышленник смог действовать из канала связи, ему необходимо выполнить атаку «человек посередине», то есть чтобы весь трафик, передаваемый между клиентским мобильным приложением и серверной частью проходил через устройство злоумышленника. Иногда в приложениях встречаются уязвимости, позволяющие такие атаки.  
при установке защищенного соединения клиентское приложение проверяет подлинность сертификата сервера и соответствие его параметров параметрам сервера. Однако иногда разработчики для удобства при работе над приложением отключают такие проверки, забывая включить их обратно в полной версии. Как итог, приложение принимает любой сертификат сервера для установки защищенного соединения, в том числе и сертификат злоумышленника.[5]

Удаленные атаки

Некоторые уязвимости в мобильных приложениях можно проэксплуатировать удаленно, и для этого даже не требуется контролировать передачу данных между приложением и сервером. Многие приложения реализуют функциональность по обработке специальных ссылок, например myapp://. Такие ссылки называются deeplinks, и работают они как на Android, так и на iOS. Переход по такой ссылке в браузере, почтовом приложении или мессенджере может спровоцировать открытие того приложения, которое умеет такие ссылки обрабатывать. Вся ссылка целиком, включая параметры, будет передана приложению-обработчику. Если обработчик ссылки содержит уязвимости, то для их эксплуатации будет достаточно вынудить жертву перейти по вредоносной ссылке. [5]

Атаки на серверную часть

Для атаки на сервер мобильного приложения злоумышленнику, как правило, достаточно изучить, как происходит взаимодействие клиентского приложения с сервером, и уже исходя из собранной информации о точках входа попытаться видоизменять запросы с целью обнаружить и проэксплуатировать уязвимости.  
  
Зачастую устройство серверной части мобильного приложения ничем не отличается от веб-приложения. Как правило, устроены серверы мобильных приложений еще проще и часто представляют из себя json- или xml-api, редко работают с HTML-разметкой и JavaScript, как это часто делают веб-сайты.

Сценарии работы системы безопасности в андроид

При фрагментарном подходе, как мы уже успели заметить, работает принцип слаженной работы нескольких модулей, настраиваемых и функционирующих между собой. По сути, модульность похода должна обеспечивать наименьшее количество действий при определённых ситуациях. Каждый модуль, организующий систему защиты, должен отправить запрос на работу другого модуля, если это необходимо при организации информационной безопасности.  
Начало работы такой системы может быть спровоцировано началом работы пользователем по определённому сценарию, или совершению действия, которое может быть заведомо небезопасным.

Примером такого сценария может служить загрузка файла по сети интернет. Где, с самого начала, модули должны проверить сертификат сервера при подключении. Потом, должны вызвать UI запрос на подтверждение скачивания файла. Уже затем по защищённым каналам скачать файл. Проверить его на наличие вредоносного ПО.[4]  
  
Атака Cloak and Dagger  
Если вкратце, то атака сводится к тому, что, используя приложение из Google Play, не требующее от пользователя никаких специальных разрешений, атакующие получают возможность выводить интерфейс этого приложения поверх других и нажимать за пользователя кнопки — так, что он сам этого не видит.  
  
Это оказывается возможным потому, что при установке приложений из Google Play от пользователя не требуют в явном виде разрешать приложениям доступ к функциям SYSTEM\_ALERT\_WINDOW

Также доступ к SYSTEM\_ALERT\_WINDOW и ACCESSIBILITY\_SERVICE позволяет злоумышленникам проводить фишинговые атаки так, что пользователь о них никогда не догадается.  
  
Например, когда пользователь открывает приложение Facebook и пытается ввести туда логин и пароль, другое приложение с доступом к Accessibility может это отследить. А дальше, воспользовавшись SYSTEM\_ALERT\_WINDOW и возможностью выводить слои поверх остальных, оно может показать пользователю покрашенное в цвета «Фейсбука» фишинговое окно, в которое ничего не подозревающий пользователь и введет логин и пароль от своей учетной записи

DDOS атака, или смс флуд   
Один день многопоточного спама (в среднем, до 10 линий, ~1000 сообщений и ~200 звонков в час) нагружает сим-карту зачастую подразумевает 100% прекращение работы сим карты. (быстро заполняет память)  
Устройством становится невозможно пользоваться и часто спам блок не может противостоять этому

Глава 3. Организация работы и управления процессами в Android Binder

Binder - относительно новая кроссплатформенная система управления пакетами, разработанная под Андроид, на основании OpenBinder. Binder реализует распределенную по компонентам архитектуру, основанную на абстрактных интерфейсах. Он похож на Comon Object Model (COM) на Windows, но в отличие от него, Binder запускается на одном устройстве и не поддерживает удаленный запуск процедур. [3][6]  
Так как в UNIX-подобной системе процесс не может получить доступ к памяти другого процесса, поэтому, ядро контролирует все процессы и потому может предоставить работу с системой управления пакетами.

Драйвер Binder управляет частью адресного пространства каждого процесса. Драйвер Binder  
управляет фрагментом памяти, доступный процессу только для чтения; все записи выполняется модулем ядра. Когда процесс отправляет данные другому процессу, ядро ​​выделяет некоторое пространство в целевом процессе и копирует данные сообщения прямо из отправляющего процесса.

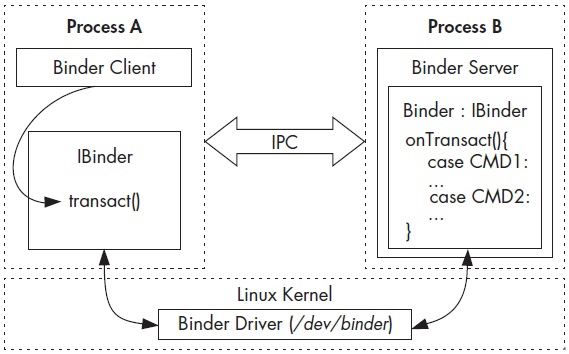


Рисунок 4. Binder обмен пакетами

На более высоком уровне каждый объект, к которому можно получить доступ через Binder framework реализует интерфейс IBinder и называется объектом Binder. Вызов объекта Binder выполняется внутри транзакции Binder, которая содержит ссылку на целевой объект, идентификатор выполняемого метода, и буфер данных.

Драйвер Binder автоматически добавляет идентификатор процесса (PID) и эффективный идентификатор пользователя (EUID) вызывающий процесс для транзакции. Вызываемый процесс может проверить PID и EUID и принять решение должен ли он выполнять запрошенный метод на основе его внутренней логики, или общесистемные метаданные о вызывающем приложении.[3]

Этот фреймворк позволяет синхронно и асинхронно вызывать методы удаленных объектов так, будто они локальные, обмениваться файловыми дескрипторами между процессами, link to death (автоматическое оповещение в случае, если Binder определенного процесса был прерван), и т.д. 

Технически каждому сервису Binder присваивает 32 битный токен, являющимся уникальным в рамках всех процессов системы (за чем следит драйвер). Этот токен используется как указатель на сервис. Если у клиента есть этот указатель, то он может взаимодействовать с сервисом

Реализация системы управления безопасностью. Проверка работы ОС при определённых ситуациях. Улучшение работы системы при разработке.

В данной части работы я опишу и покажу взаимодействие модулей, обеспечивающих системную ЗИ. Покажу, как можно изменять настройки отдельных модулей, для обеспечения большей практичности взаимодействия при реализации различных проектов.  
Для проверки используем сценарий со скачиванием файла из браузера Chrome. При этом, используя эмулятор ОС Android на ОС Windows в рабочем пространстве “Android Studio”, позволяющее нам работать с кодом приложений и ОС.

Последовательность работы системы при конкретном сценарии

При старте эмулятора, Binder работает с данными, получаемыми с различных модулей, установленными на устройстве. Каждому процессу соответствует свой ID, как уже было рассказано выше.

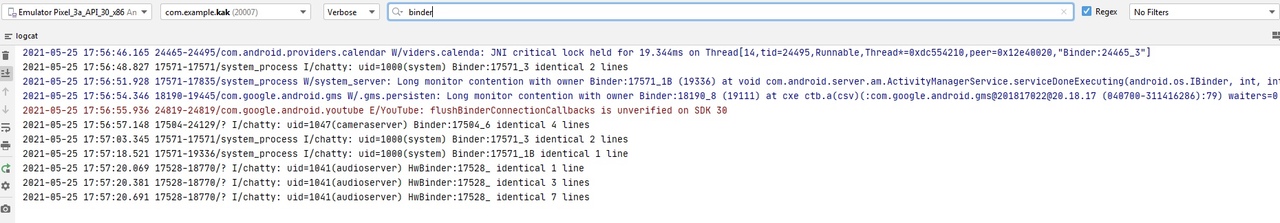


Рисунок 5. Работа Binder при запуске

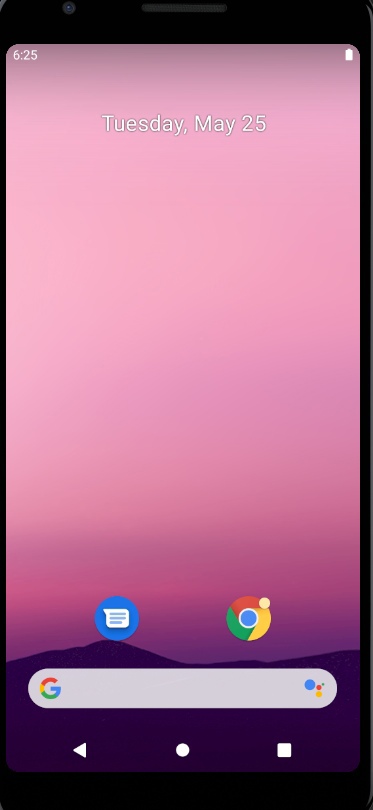


Рисунок 6. Интерфейс эмулируемого устройства

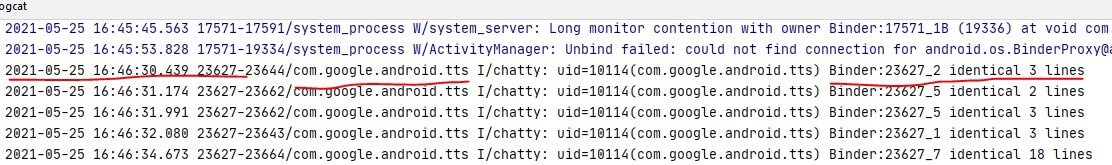
Запуск браузера приводит к началу взаимодействия в системе. В логах можно заметить начало работы уже знакомого Binder, обеспечивающего начало взаимодействия между различными модулями.  


Рисунок 7. Работа браузера и Binder

При этом, можем наблюдать работающий модуль с новой настройкой сетевой конфига  


Рисунок 8. Логи с работой модуля

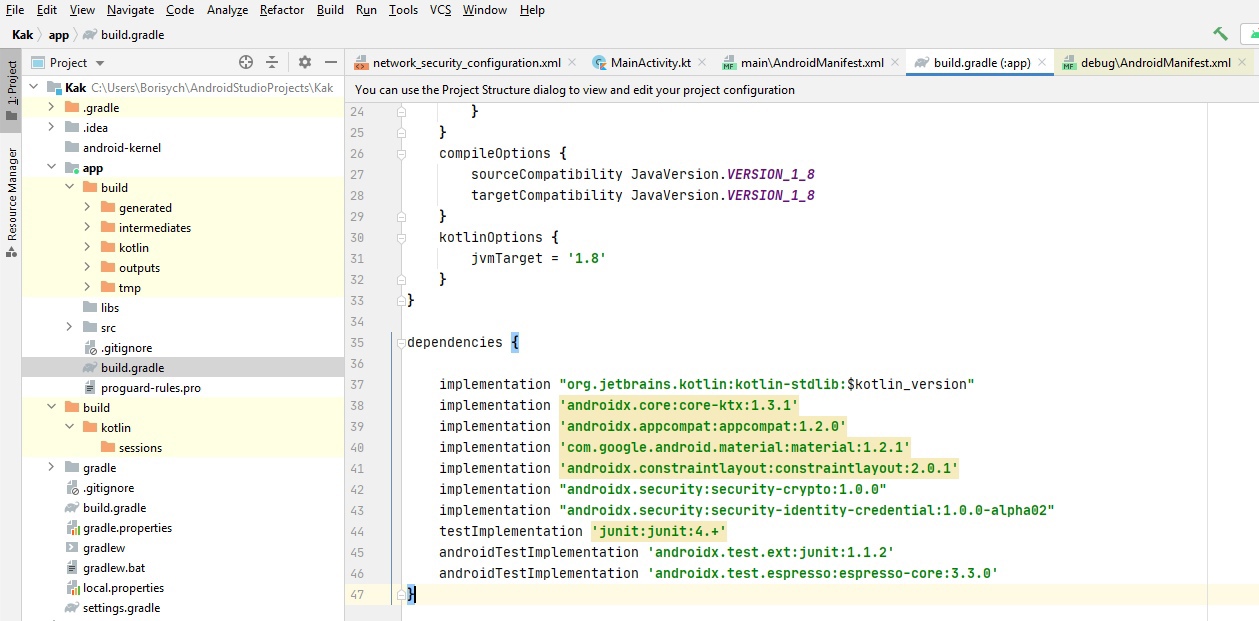
Конфиги мы добавляли скачиванием библиотек и связыванием их с системой. Работают они при отправке данных.  


Рисунок 9. Конфиги для сети

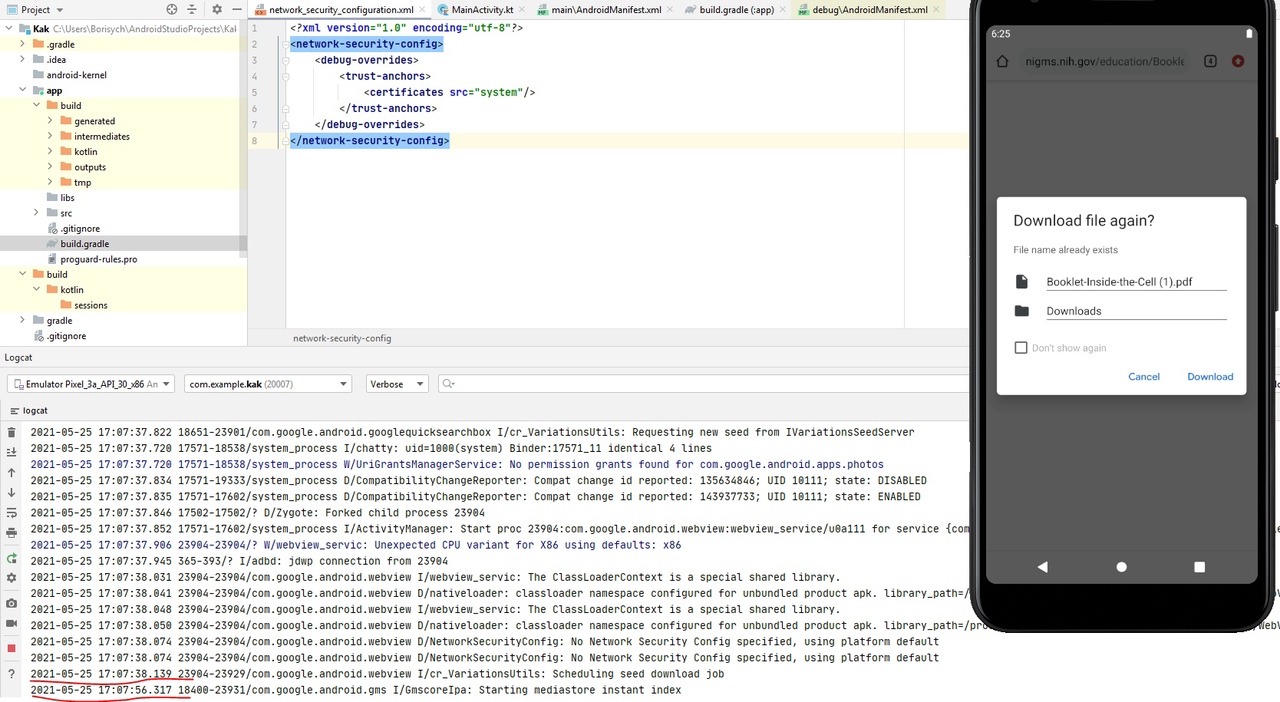
Далее, находим файл в сети интернет для скачивания. При предложении браузером места загрузки файла, уже выбирается директория, куда будет установлен файл. Система выбирает путь, определённый для данного приложения.   
  


Рисунок 10. Загрузка файла

Методы шифрования, определённые в приложении (Chrome), указываются в конце загрузки.  


Рисунок 11. Шифрование данных браузером

Настройка системы безопасности для приложения

Приложение является одним из основных элементов, составляющих ОС Android в целом. Каждое приложение, помимо функционала, требует собственных настроек безопасности. А уже сами настройки нужно соотнести так, чтобы эффективно использовать систему безопасности, уже представленную в ОС.

Начнём с того, что приложение нуждается в регистрации. А для связи этого приложения с сетью интернет нам нужен сетевой сертификат. Данную вещь мы можем создать в приложении. Хранится он будет как файл с расширением “.jks”

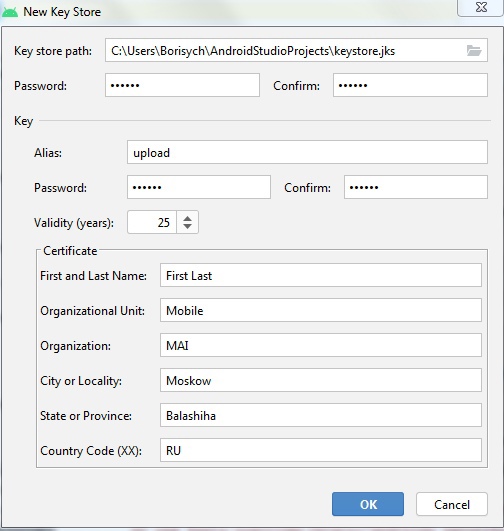


Рисунок 12. Сетевой сертификат

Для приложения, сертификат будет является идентификатором в сети. В нём зашифрована информация о приложении. Зашифрован он методом AES-256.  
Также, для приложения существуют уже названные ранее критерии шифрования трафика в сети интернет через сетевой конфиг. [10]  
В рабочем приложении можно настраивать проверку безопасности файла, или же его защищать.  
Можно настроить, конечно, срабатывание после установки файла. Но это уже вне тематики курсовой работы.  
  


Рисунок 13. Проверка файла

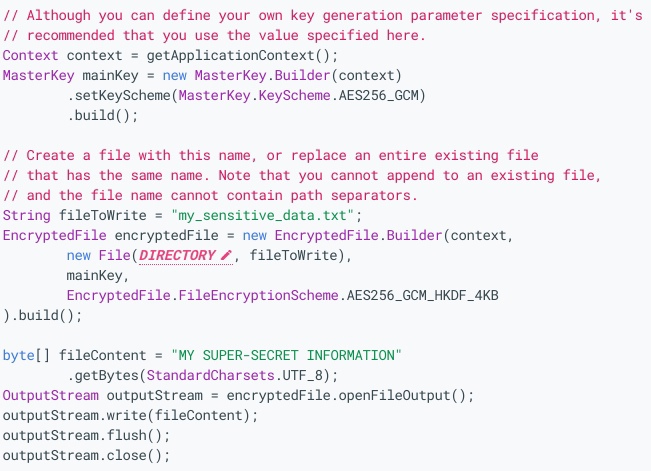
Возможность зашифровки файла, созданного в приложении. Продемонстрирован метод шифрования AES-256  


Рисунок 14. Шифрование файла

Можно сказать, что методов защиты приложения от возможных атак большое количество. И основываясь на уже реализованных в системе методах, можно значительно улучшить взаимосвязь в работе систем для любых ситуаций.

Заключение

По итогам проведённой работы можно сделать несколько выводов.  
Во-первых, более эффективной методикой построения безопасной ОС можно считать комплексный подход, так как элементы системы тесно взаимодействуют друг с другом.  
Во-вторых, организация безопасности Android является работой не только разработчиков ОС, но и сторонних разработчиков. Это обеспечивается как открытостью кода системы, так и открытым кодом ядра. Несомненно, эта открытость порождает разработчиков вирусного ПО. Но данная система имеет частые обновления, закрывающие недочёты безопасности.

В сегменте работы с ОС Android существуют ограничение в виде малого количества учебных пособий и статей на русском языке, связанных с работой систем безопасности.

Список использованных источников

1. Ресурс о технике ://gagadget.com URL: <https://gagadget.com/android/40285-samaya-podrobnaya-istoriya-android-ot-astro-boy-do-pie/> О истории Android (дата обращения 14.04.2021)
2. Учебное пособие "AISEC-TR-2012-001-Android-OS-Security" Главы 2, 3.
3. Учебное пособие "Android\_Security\_Internals-An\_In-Depth\_Guide\_to\_Android\_s\_Security\_Architecture" Главы 1,2,3,5,8
4. Учебное пособие "Защита в операционных системах 2014" Глава 1 пункт 2.
5. Безопасность мобильных устройств и приложений // [habr.com](https://vk.com/away.php?to=http%3A%2F%2Fhabr.com&cc_key=): сообщество IT-специалистов. URL: [https://habr.com/ru/company/pt/blog/509814/](https://vk.com/away.php?to=https%3A%2F%2Fhabr.com%2Fru%2Fcompany%2Fpt%2Fblog%2F509814%2F&cc_key=)
6. Статья об использовании Binder в архитектуре Android. // [habr.com](https://vk.com/away.php?to=http%3A%2F%2Fhabr.com&cc_key=): сообщество IT-специалистов. URL: [https://habr.com/ru/post/274635/](https://vk.com/away.php?to=https%3A%2F%2Fhabr.com%2Fru%2Fpost%2F274635%2F&cc_key=) (дата обращения 5.05.2021)
7. Документация Android разработки // [developer.android.com](https://vk.com/away.php?to=http%3A%2F%2Fdeveloper.android.com&cc_key=) URL: [https://developer.android.com/training/articles/secur..](https://vk.com/away.php?to=https%3A%2F%2Fdeveloper.android.com%2Ftraining%2Farticles%2Fsecurity-config&cc_key=) (дата обращения 5.05.2021)
8. Документация Android разработки // [developer.android.com](https://vk.com/away.php?to=http%3A%2F%2Fdeveloper.android.com&cc_key=) URL: [https://developer.android.com/topic/security/data-and..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fdeveloper.android.com%2Ftopic%2Fsecurity%2Fdata-android-versions) Чтение файлов (дата обращения 5.05.2021)
9. Документация Android разработки // [developer.android.com](https://vk.com/away.php?to=http%3A%2F%2Fdeveloper.android.com&cc_key=) URL: [https://developer.android.com/guide/topics/security/c..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fdeveloper.android.com%2Fguide%2Ftopics%2Fsecurity%2Fcryptography) Шифрование файлов (дата обращения 5.05.2021)
10. Документация Android разработки от пользователей // [android-doc.github.io](https://vk.com/away.php?to=http%3A%2F%2Fandroid-doc.github.io&cc_key=) URL: [https://android-doc.github.io/tools/publishing/app-si..](https://vk.com/away.php?to=https%3A%2F%2Fandroid-doc.github.io%2Ftools%2Fpublishing%2Fapp-signing.html&cc_key=) Создание сертификата приложения. (дата обращения 24.05.2021)