INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCHERS
Volume 1,issue 1,2023

Улучшить системы безопасности Android в средствах связи

Абдурахимов Озодбек Азимжанович

Тожидинов Азизбек Илхомжонович

Студенты Ферганского филиала ТУИТ имени Мухаммеда аль-Хорезмий

Аннотация: Появление технологий породило множество разработок, которые упростили работу людей. Из-за растущей потребности быть в авангарде коммуникации и информации люди прибегают к использованию мобильных телефонов, причем большой процент предпочитает системы на базе Android. Точно так же системы подвержены различным угрозам, исходящим от технологий, и связанные с этим обнаруженные недостатки безопасности и несанкционированный доступ к конфиденциальной информации могут представлять серьезную угрозу для общей производительности систем Android. Сосредоточено на том, как можно улучшить системы безопасности Android при использовании или установке приложений..

Ключевые слова: Boson NetSim, Cisco, GNS3 (Virtual машина), CLI.

За последние полтора десятилетия добавление программного обеспечения Android к мобильным телефонам захватило современный рынок. Общие цифры, полученные в результате многочисленных исследований, показывают, что не менее 133 миллионов пользователей смартфонов владеют устройством Android, и это число растет день ото дня. Это означает увеличение количества непреднамеренных атак и угроз для операционных систем. Прежде чем выделить различные проблемы, с которыми сталкиваются системы Android, необходимо понять конкретную структуру программного обеспечения, а также их уровни защиты и архитектуру системы.

Разработчики Android сосредоточились на том, чтобы архитектура безопасности систем зависела от ядер на базе Linux, что обеспечивает изоляцию приложений друг от друга для обеспечения конвергенции. Методы изоляции требуют подписей и различных разрешений для правильной работы со сложной моделью безопасности. Усложнения разработаны таким образом, что пользователь может просматривать и изменять настройки по своему усмотрению. В качестве примера можно привести просмотр установленных параметров безопасности, чтобы пользователь мог изменить предпочтения, которые ему подходят. Часто операционная система Android поставляется с инструкциями по изменению настроек, но задачи могут показаться сложными.

Он продолжается описанием модели безопасности безопасных систем Android и ролей пользователей в безопасности. Будучи открытой платформой, Android позволяет своим обычным и опытным пользователям развертывать несколько уровней функций безопасности на каждом устройстве, помогая улучшить их работу. Во внутренней системе, используемой операционной системой Android, каждый уровень работает вместе с другими уровнями для обеспечения безопасности системы. Все уровни объединены в основной уровень, который опирается на собственные идеи компонента безопасности Linux, чтобы изолировать их.

Первый уровень, представляющий собой ядро Linux, устанавливается на устройство и предоставляет множество аппаратных и программных приложений. Результаты

исследований показывают, что аппаратные драйверы, используемые ядром, включают данные дисплея, камеру и устройства Bluetooth. Это затрудняет тестирование и доработку, поскольку любая ошибка может привести к повреждению структуры устройства. Разработчики Android смягчили возникновение таких ситуаций, минимизировав возможность системных разрешений.

Уровень аппаратной абстракции, который создает связь между программным обеспечением устройства и аппаратным обеспечением. Пользователи не могут настраивать этот уровень самостоятельно, так как системы зависят от анализа данных и производительности системы. Поскольку устройство Android построено с использованием ядра Linux, для процесса разработки требуется большое количество библиотек C и C++. Функция среды выполнения Android содержит среду, которая позволяет системе работать эффективно. Функция среды выполнения, ранее известная как Dalvik, помогает компилировать системные коды при установке на мобильное устройство.

Этот компонент системы Android содержит необходимые API-интерфейсы, используемые системами Java, которые помогают создавать системы Android. Прикладной уровень важен для обеспечения требований к данным и графическому пользовательскому интерфейсу устройств. С другой стороны, прикладной уровень интегрирован с системными приложениями, отделенными от установленных пользователем приложений. Приложения сильно зависят от ресурсов, которые система предоставляет снизу вверх.

Песочница приложения на основе ядра дает каждому приложению уникальный UID (идентификатор пользователя) и GID (идентификатор группы). Это предотвращает доступ приложений к данным других приложений. Таким образом, загруженное приложение не может войти в систему.

В то время как некоторым приложениям требуется доступ к другим приложениям, безопасный IPC позволяет критически важным приложениям взаимодействовать друг с другом. Это делается с помощью собственных сокетов, соединителей и целей, при этом большинство приложений используют собственные сокеты для безопасного взаимодействия с другими приложениями. Привязки также важны при развертывании кода в среде Android и активируются только во время работы приложения. Исследования показывают, что намерения помогают собирать информацию об операциях, которые необходимо выполнить в дополнение к предыдущим событиям.

Как упоминалось выше, системы Android имеют много нативного кода внутри программы. Это приводит к тому, что разработчики Android вынуждены запускать большинство системных служб с меньшими привилегиями, чтобы предотвратить злонамеренное использование системных ресурсов. Локальным кодам назначаются разные подписи, аналогичные центрам сертификации, как показано в следующем разделе.

По сравнению с системами iOS, системы Android имеют другие кодовые символы, которые не требуют авторизации пользователя в процессе подписания. Однако стандартный процесс требует подписи кода приложений, которые затем используются в модели безопасности. Подписи также используют открытые ключи для добавления криптографической обработки к сертификатам X.509. Подписи кода различаются в зависимости от схемы подписи Java JAR. Подписи кода обеспечивают подлинность этого приложения с примером подписи кода, показанным ниже:

<разрешения>

...

<имя элемента =

"com.google.android.googleapps.permission.ACCESS_GOOGLE_PASSWORD"

package = «com.google.android.gsf.login» security = «2»/>

...

Есть несколько причин, по которым сигнатуры кода важны, в первую очередь для обеспечения того, чтобы обновления приложения исходили от одного и того же автора (одна и та же политика происхождения) плагина для установления доверительных отношений между приложениями. Обе упомянутые выше роли выполняются путем сравнения сертификатов, подписанных целевым приложением, установленным с помощью сертификата обновления. Беспроводные обновления (OTA), а также исправления программного обеспечения зависят от сигнатур кода для обеспечения подлинности и идентичности.

Наконец, были реализованы разрешения приложений Android, чтобы помочь пользователю предотвратить доступ определенных приложений к критически важным системным ресурсам. Система Android позволяет пользователю выбирать, устанавливать ее или нет. Ядро Linux сортирует процесс сопоставления привилегий и разрешает доступ к системным файлам и другим приложениям.

Существует несколько подходов к защите приложений и программного обеспечения Android с двух разных точек зрения. Как разработчик, вы можете предпринять определенные действия, чтобы лучше обеспечить безопасность своих пользователей. Одним из таких примеров может быть тестирование на проникновение, когда разработчик выявляет и снижает любые риски, которым подвержен пользователь. Процесс тестирования на проникновение учитывает ряд процессов, в том числе возникновение атак, анализ взаимодействия приложений с файлами, емкость системы хранения и связь между объектами. Вторая точка зрения исходит из безопасности приложения обычным пользователем с целью повышения его производительности.

Внешние связи могут быть еще одним фактором риска. Как правило, самые популярные приложения используют ту или иную форму криптографии для ограничения несанкционированного доступа. При разработке приложения крайне важно, чтобы пользователь никогда не жестко кодировал криптографический ключ, потому что приложения для Android имеют открытый исходный код и взлом неизбежен.

Тестирование программы, которая оценивает конфиденциальные данные, имеет решающее значение для исправления любых возникающих пробелов. Обеспечение надлежащих разрешений на чтение/запись с максимальными ограничениями, связанными с системными файлами, также может ограничить количество приложений, имеющих доступ к пользовательским данным. Чтобы убедиться, что ненужные данные не собираются, разработчик должен оценить, какие данные хранятся во время тестирования на проникновение данных.

Дополнительные состояния безопасности пользователя

Управление разрешениями. Как упоминалось ранее, пользователь должен убедиться, что устанавливаемые им приложения не являются бесполезными и что ему предоставлены необходимые разрешения. С прошивкой Android Marshmallow пользователь теперь может заблокировать приложение от использования разрешения. Это может показаться нелогичным, но с учетом этого программа не будет работать корректно и заставит пользователя явно запрашивать разрешения на запуск. Однако это может гарантировать, что некоторые сторонние приложения не будут работать с определенным оборудованием или файлами, если это вас беспокоит. Удаленное стирание и отслеживание — еще одна функция, которую предоставляет Android, — удаленное стирание и отслеживание на случай, если вор заберет ваш телефон. Резервное копирование файлов. Пользователи могут создавать резервные копии своих приложений и файлов на жестком диске или в облаке.

Хранение данных. По умолчанию большая часть данных хранится во внутренней памяти устройства Android для быстрого доступа. Вы также можете зашифровать файлы с помощью ключа с помощью KeyStore, который защищает файл паролем. Хранение приложений или данных извне — самый безопасный способ, например, SD-карта, установленная в телефоне, по умолчанию глобально доступна для чтения и записи; однако в более новых версиях Android внешние SD-карты могут быть зашифрованы. Используйте криптографические схемы для наиболее безопасного хранения, так как большинство приложений работают с Java Cryptographic Architect, который обычно использует 256-битный AES, 256-битные открытые ключи, режимы CBC, CTR или GCM, обеспечивая целостность с помощью HMAC-SHA1/256/. 512 или GCM. Полное или файловое шифрование диска — это примерный способ обеспечения конфиденциальности для устройства Android, использующего AES (128), CBC и SHA256 для шифрования данных.

Предотвращение вредоносного ПО. Хороший способ предотвратить вредоносное ПО — не нажимать на элементы, которые выглядят слишком хорошо, чтобы быть правдой. Другими словами, будьте осторожны с веб-сайтами, которые вы посещаете, всплывающими окнами с рекламой, на которые, как вам может показаться, нужно нажать, и заслуживают ли доверия загружаемые вами приложения.

Блокировка экрана—блокировка клавиатуры, сканер радужной оболочки глаза, сканер отпечатков пальцев Существуют тысячи приложений, которые можно использовать для предотвращения доступа. Однако лучше проверить отзывы и популярность этого приложения, чтобы обеспечить безопасность ваших данных. Все приложения в плеймаркете не являются подлинными.

Обновляйте свою операционную систему — само собой разумеется, но многие пользователи пренебрегают обновлением своих телефонов.

Список использованной литературы:

1.Dadakhon, T. (2022). Factors that Review Students' Imagination in the Educational Process. Spanish Journal of Innovation and Integrity, 5, 551-557.

2.Davi, L., Dmitrienko, A., Sadeghi, A.R. and Winandy, M. (2010) Privilege Escalation Attacks on Android. International Conference on Information Security, Chengdu, 17-19 December 2010, 346-360.

3.Sun, S.T., Cuadros, A. and Beznosov, K. (2015) Android Rooting: Methods, Detection, and Evasion. 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, Denver, 12-16 October 2015, 3-14.

4. Turgunova, N., Turgunov, B., & Umaraliyev, J. (2023). AUTOMATIC TEXT ANALYSIS. SYNTAX AND SEMANTIC ANALYSIS. Engineering problems and innovations.

5. Turg‘unov, B., Turg‘unova, N., & Umaraliyev, J. (2023). AVTOMOBILSOZLIKDA AVTOMATLASHTIRISHNING O ‘RNI. Engineering problems and innovations.

6. Nafisaxon, T. U., Jamshidbek To‘xtasin o‘g‘li, U., Arsenevna, D. E., & Azimjon o‘g‘li, A. O. (2022). AVTOMATLASHTIRILGAN AVTOTURARGOH IMKONIYATLARI VA QULAYLIKLARI. INNOVATION IN THE MODERN EDUCATION SYSTEM, 3(25), 45-48.

7. Abdurakhmonov, S. M., Kuldashov, O. K., Tozhiboev, I. T., & Turgunov, B. K. (2019). The Optoelectronic Two-Wave Method for Remote Monitoring of the Content of Methane in Atmosphere. Technical Physics Letters, 45, 132-133.

8. Kodirov, E., Turgunov, B., & Muxammadjonov, X. (2019). IN THE WORLD REFUSES TO USE FACE RECOGNITION TECHNOLOGY. Мировая наука, (9 (30)), 34-36.