УДК 004.056.53

ВОПРОСЫ БЕЗОПАСНОСТИ ПРИ УРОВНЕ ДОСТУПА СУПЕРПОЛЬЗОВАТЕЛЬ В ОПЕРАЦИОННОЙ СИСТЕМЕ ANDROID

Калинин С.Б., Рейнюк А.В.

Аннотация: В настоящее время большинство мобильных устройств работают на операционной системе Android, которая позволяет пользователю гибко настраивать различные параметры системы. Некоторые более продвинутые настройки заблокированы с помощью разделения прав между обычными пользователями устройства и суперпользователем. Как разработчики устройств и программного обеспечения, так и пользователи заинтересованы в сохранности чувствительных данных, особенно в области здоровья и финансов. Получения прав суперпользователя несет за собой риски появления уязвимостей и утечки данных в системе безопасности. Происходит постоянная модернизация как методов получения доступа суперпользователя, так и механизмов обнаружения и борьбы с ними.

Ключевые слова: Android, суперпользователь, безопасность, смартфон.

Kalinin S.B., Reiniuk A.V.

Abstract: Nowadays, most mobile devices run on the Android operating system, which allows the user to flexibly customize various system settings. Some of the more advanced settings are locked down by separating the rights between regular users of the device and the superuser. Device and software developers and users alike are interested in keeping sensitive data safe, especially in the areas of health and finances. Gaining superuser rights brings with it the risks of security vulnerabilities and data leaks. Both the methods for gaining superuser access and the mechanisms for detecting and combating them are constantly evolving.

Key words: Android, superuser, security, smartphone.

ВВЕДЕНИЕ

Приватность и целостность данных играют важную роль в цифровых услугах, которыми ежедневно пользуются множество людей. При использовании подобных услуг, обе стороны – пользователь и организация, предоставляющая услугу – заинтересованы в том, чтобы использовать защищенные каналы связи и протоколы обмена данными, а также иметь возможность проверить целостность этих данных и нескомпрометированность приложений. С развитием мобильных устройств, смартфоны стали одним из самых популярных способов взаимодействия с цифровыми услугами. Приложения, устанавливаемые на смартфоны, нацелены на предоставление полноценного опыта для пользователя и глубоко интегрируются в операционную систему. Соответственно, упомянутая обеспокоенность приватностью и целостностью данных также применяется к этим приложениям.

Широко используемая операционная система Android позволяет большому кругу производителей представлять огромное разнообразие устройств каждое из которых использует одинаковую операционную систему. Эта открытость привела к глобальной популярности смартфонов и других устройств, базирующихся на операционной системе Android [1]. Она также предоставляет широкий спектр возможностей настройки устройства под персональные нужды каждого пользователя в сравнении с конкурентными системами.

Организации, предоставляющие цифровые услуги, разрабатывают приложения для их использования на устройствах Android. В случае некоторых приложений, расчет на предоставляемую пользователем безопасность устройства бывает достаточным, т.к. при использовании личная информация пользователя не будет затронута и каким-либо образом передана через каналы связи, например Интернет (приложения просмотра изображений, текстовые записки и прочее). Другие же приложения в свою

очередь требуют использования доверенной исполнительной платформы (например, банковские приложения). Если данные, обрабатываемые этими приложениями, попадут в руки злоумышленников то это может привести к серьезным последствиям, таким как кража идентификации личности, денежных средств и прочим. Эти приложения, как и пользователи, владельцы интеллектуальной собственности, имеют общий интерес в обеспечении защиты своих данных, и должны каким-то образом убедиться в целостности устройства, на котором запускается приложение, а также конфиденциальность используемых каналов связи.

Такой контроль может напрямую противоречить с возможностями пользователя настраивать и управлять параметрами своего устройства. Одним из популярных методов получения более широкого контроля над собственным устройством и повышенных прав доступа является процесс получения прав суперпользователя смартфона.

Какие преимущества могут предоставить права суперпользователя:

1. Удаление предустановленных производителем приложений. Чаще всего такие приложения не могут быть удалены при использовании прав стандартного пользователя

2. Установка неофициальных версий приложений. Получение прав суперпользователя открывает возможность установки неофициальных версий операционной системы Android, которые могут как предоставить дополнительный функционал, так и увеличить количество угроз для устройства.

3. Блокировка рекламы. Суперпользователь также может заблокировать встроенную рекламу на уровне системы, без необходимости установки дополнительных приложений.

4. Настройка внешнего вида. Расширенные права позволят установить частные наборы визуальных тем и иконок для устройства.

5. Приложения, работающие только с правами суперпользователя. Некоторые приложения требуют наличия прав суперпользователя для корректной работы, и получение таких прав позволяет пользователю установить такие приложения (например, для глубокой очистки телефона).

6. Сохранение и восстановление состояния устройства. Права суперпользователя позволяют сохранить и восстановить файлы приложений на своих устройствах в случае его замены или сброса до заводских настроек.

7. Настройка производительности. Доступ суперпользователя может открыть возможность настройки энергопотребления и работы устройства для увеличения срока работы и жизни батареи.

8. Настройка аппаратной части. Расширенный доступ предоставляет возможность увеличить аппаратные возможности устройства посредством разгона центрального процессора и других компонентов.

Проблемы безопасности при получении прав суперпользователя:

1. Вредоносное программное обеспечение. Устройства с правами суперпользователя уязвимы и более подвержены к атакам вредоносных приложений, которые могут быть запрограммированы на сбор информации со всего устройства и данных, вводимых пользователем, для дальнейшей передачи третьим лицам.

2. Отсутствие обновлений системы безопасности. Такие устройства могут перестать получать актуальные обновления и будут открыты к новым угрозам.

3. Ненамеренные изменения в системе. Суперпользователь может ненамеренно модифицировать систему таким образом что это приведет к нестабильностям, сбоям в работе и потере данных.

4. Системные уязвимости. Расширенные права могут вскрыть ранее защищенные каналы и открыть больше возможностей для взлома устройства.

Далее будут изложены доступные приложениям возможности для проверки и обеспечения того, чтобы исполнительная среда, в которой они

работают, позволяла безопасную обработку конфиденциальных данных. Будут представлены методы получения прав суперпользователя для смартфонов и актуальные средства распознавания и противодействия этому.

1. ПРИМЕРЫ В ЛИТЕРАТУРЕ

Процесс получения дополнительных привилегий и уровня доступа суперпользователя на Linux и UNIX устройствах существовал задолго до появления операционной системы Android. Методы получения прав суперпользователя соответственно были адаптированы для альтернативной среды Android. Далее рассмотрим работы, сконцентрированные на получении таких прав и подготовке к этому Android устройств.

Общую картину на существующие методы получения прав суперпользователя представляют несколько публикаций:

- Ян и др. [2] подробно описывают метод получения прав суперпользователя и выделяют три основных этапа: разблокировка загрузчика, установка неофициальной программы восстановления, получение прав для приложений и инструментов. Также приводится пример получения внесистемного доступа суперпользователя, при котором в память устройства устанавливается Linux система с расширенными правами с помощью которой затем запускается сама операционная система Android.
- Сан и др. [3] в своей работе выделяют несколько подходов к получению прав суперпользователя устройства:

А) режим загрузчика - позволяет загружать сторонние образы в память устройства;

Б) восстановление из стороннего источника - возможность подменить файлы системы, не меняя сам образ системы;

В) загрузочная SD-карта - некоторые производители поддерживают режим загрузки с внешнего носителя, на котором может быть установлена система с доступом суперпользователя;

Г) использование уязвимостей приложений – позволяет получить расширенный доступ используя бреши в правах других приложений;

Д) привилегированный отладочный мост Android – инструмент отладки систем Android, изначально имеет повышенные права доступа.

- Видас и др. [4] в своей работе делают акцент на атаках, которые происходят уже при наличии уровня доступа суперпользователя и оценивают их воздействие на целостность устройств. Авторы рассматривают пять классов атак:

А) без привилегий суперпользователя – пользователь устанавливает приложения из неизвестного источника несмотря на предупреждения системы;

Б) удаленный доступ – получение приложения из подтвержденного источника, которое маскируется под не вредоносное, но в определенный момент получает инструкции для использования уязвимостей;

В) прямой доступ с включенным отладочным мостом – получение доступа к правам суперпользователя с помощью отладочного моста Android;

Г) прямой доступ с выключенным отладочным мостом – взаимодействие с режимом восстановления устройства для получения расширенных прав;

Д) прямой доступ к незащищенному устройству – установка вредоносного программного обеспечения напрямую при контакте с устройством.

Заключением этой работы стало подтверждение того, что использование методов получения прав суперпользователя на устройстве приводит к увеличению рисков вредоносного вмешательства сторонних приложений и нарушения стабильности работы устройства с точки зрения безопасности.

- С разнообразием способов получения прав суперпользователя, разработчики приложений стремятся компенсировать уязвимости наличия

этих прав, внедряя в свои приложения большое количество контрмер. Эта гонка вооружений между новыми методами получения расширенных прав и смягчению последствий детально описана в работе Нгуен и др. [5].

Этот обзор предоставляет сводку о наиболее распространенных подходов к распознаванию наличия прав доступа суперпользователя и методах получения прав суперпользователя на обнаружение которых они направлены, а также способы, предоставляемые ядром Linux.

2. МЕТОДЫ ПОЛУЧЕНИЯ ПРАВ СУПЕРПОЛЬЗОВАТЕЛЯ

Получение прав суперпользователя на Android устройстве может быть выполнено с помощью разнообразных подходов. Некоторые из них опираются на нежелательное поведение приложений или операционной системы, другие же следуют установленным путям получения повышенных привилегий через инструменты, предоставляемые производителем устройства или разработчика операционной системы.

Далее будут выделены две основных категории подходов к получению расширенных прав и описаны различия в том, как выполняется повышение прав доступа, а также способы их обнаружения.

2.1 Типы подходов к повышению прав

В первую очередь необходимо определить в каком состоянии устройство Android можно определить как имеющее доступ к правам суперпользователя. Операционная система Android была создана на основе ядра Linux, которое базируется на принципе существования в системе множества пользователей с различными конкретными функциональными ролями и разделенными правами. Пользователя, который владеет максимальным набором прав и разрешений называют суперпользователем (а также «root»). Наличие широкого доступа к системе позволяет изменять параметры и права, установленные любым другим пользователем, и управлять внутренними системными процессами и функциями. В это же

время, обычные приложения запускаются в своем собственном ограниченном пространстве, используя пользователя со стандартными правами тем самым, не получая доступа к различным важным функциям и файлам операционной системы.

Устройство можно назвать имеющим права доступа суперпользователя если приложение, изначально работающее под непривилегированным пользователем, повышает свои права до уровня суперпользователя.

1. «Мягкий доступ» – существует только в рамках текущей сессии работы устройства. Такой метод получения прав суперпользователя может быть прикрыт с помощью обновлений операционной системы. Доступ суперпользователя, полученный таким способом, сложно обнаружить, так как он не вносит изменений в операционную систему. С помощью исследования поведения и разрешений исполняемых в текущий момент процессов можно обнаружить поведение присущее этому типу доступа. Из-за того, что этот способ детектирования опирается на просмотр других процессов, то сам он должен обладать расширенным набором привилегий. Методы получения прав, которые относятся к мягким, чаще всего опираются на уязвимость в системе безопасности, которая может быть использована для повышений своих прав доступа.

2. «Жесткий доступ» – поддерживается на постоянной основе с помощью внесения изменений в файловую систему или специфический раздел. Обычно выполняется с помощью записи самодельного программного обеспечения в память устройства, чаще всего снимая заводскую гарантию. Один из наиболее распространенных примеров – приложение «Magick» [6], которое модифицирует только загрузочный сектор при этом не трогая саму операционную систему Android.

2.2 Эффект

Как только устройство было подвержено операциям по получению прав суперпользователя, следует рассматривать исполнительную среду Android

как скомпрометированную. Файлы приложений становятся уязвимыми для манипуляций, а системные функции – для подмены. Становится возможным изменять значения и поведение функций в системе. Безопасность каналов данных может быть нарушена, могут иметь место утечки. В таком состоянии проблема распознавания наличия прав суперпользователя становится еще более сложной [7].

Конечный пользователь может не смочь определить имеются ли права суперпользователя на устройстве, так как его поведение может не отличаться от того, что было до получения прав, но при этом вредоносные приложения могут работать в фоне, никак себя не показывая, и не требовать от пользователя каких-либо разрешений.

3. ПРОТИВОДЕЙСТВИЕ ПОЛУЧЕНИЮ ПРАВ СУПЕРПОЛЬЗОВАТЕЛЯ

Большинство производителей смартфонов адаптируют операционную систему с открытым кодом Android под свои нужды. Это может быть необходимо для обеспечения специфичного для устройства функционала или для брэндирования устройств. Иногда разработчики включают сложные функции, обеспечивающие безопасность, но чаще всего происходит обратный процесс как указано в [8].

Исследователи из команды Project Zero из Google предлагают альтернативный подход к интеграции мер противодействия и безопасности в операционную систему Android. Вместо того, чтобы использовать жестко установленные решения безопасности, включенные в ядро Linux, более эффективным может оказаться выделение этих мер в отдельные приложения, что позволит более быстро их обновлять и модифицировать без необходимости обновления всей системы. В среде Linux уже существуют примеры такого подхода к обеспечению безопасности, SELinux [9] и AppArmor [10], и они хорошо себя зарекомендовали. Эти решения могут быть применены к базирующимся на Android операционным системам без больших вложений сил. Google продолжает внедрение безопасных стандартных конфигураций и включает современные и экспериментальные

системы усиления безопасности и инструменты в новых выпусках Android [11].

Существует широкий спектр систем противодействия, присущих каждому отдельному производителю. В то время как методы, основывающиеся на аппаратном исполнении, тяжело или невозможно обойти, программные решения могут быть не такого высокого качества как возможности, предоставляемые ядром Linux. Это происходит из-за ограниченных возможностей производителей тестировать, фиксировать и исправлять ошибки в архитектуре программного обеспечения для поддержания действенных мер безопасности.

Один из примеров специфичных для производителя методов усиления защиты как со стороны аппаратной части, так и программной это Samsung Knox [12] – платформа, включающая в себя комплекс решений безопасности, базирующийся на технологии ARM TrustZone [13]. Эта система также постоянно совершенствуется, так, например, в старых версиях Samsung Knox, существовали некоторые уязвимости (как упомянутая в [14]), которые были тщательно задокументированы и могли предоставить повышенный доступ к функциям смартфона.

Подводя итог, меры противодействия не устраняют наличие уязвимостей системы безопасности как таковые, но могут ограничить возможности, которые могут предоставить такие уязвимости.

4. РАСПОЗНАВАНИЕ НАЛИЧИЯ ПРАВ ДОСТУПА СУПЕРПОЛЬЗОВАТЕЛЯ

С развитием методов получения прав суперпользователя, распознавание механизмов также прогрессировало. Приложения, работающие с чувствительными данными, такие как банковские приложения или те, которые занимаются отслеживанием о состоянии здоровья пользователя, серьезно нуждаются в подтверждении целостности среды для избежания утечек персональных данных другим лицам.

В то время как методы получения прав либо работают, либо нет, методы распознавания такого доступа не могут дать четкий ответ. Если доказательства наличия доступа суперпользователя получится обнаружить, то механизм распознавания может точно утверждать, что исследуемая среда была подвержена влиянию программ для получения прав. Если же, с другой стороны, доказательства не будут обнаружены, то единственным выводом будет только то, что они не обнаружены – среда все так же может иметь права суперпользователя, но при этом не оставлять никаких следов, которые мог бы обнаружить механизм распознавания.

Следует считать, что это утверждение правдиво для всех механизмов распознавания прав, так как суперпользователь и принадлежащие ему привилегии являются частью того, как ядро управляет системой и будут существовать всегда. В то же время, механизм обнаружения расширенного доступа может искать только индикаторы доступа этих привилегий для обычного пользователя. Такие проверки, проводимые в потенциально нарушенной среде, могут быть обмануты с помощью других механизмов программного обеспечения для получения прав. Пример таких скрывающих приложений это Magick [6] и RootCloack [15]. Эти приложения скрывают файлы, папки и процессы чаще всего проверяемые механизмами распознавания. Так как скрывающие приложения также имеют предопределенный список действий, то механизмы обнаружения прав суперпользователя могут работать на опережение в своих алгоритмах обнаружения, включая новые индикаторы, которые еще не были покрыты алгоритмами скрывающих приложений. Такая ситуация приводит к гонке вооружений между скрывающими и обнаруживающими приложениями.

4.1 Распространенные методы распознавания

Наиболее распространенные методы распознавания, представленные ниже, используются как в приложениях с открытым исходным кодом, так и в различных закрытых приложениях от сторонних разработчиков. Если приложения с открытым кодом можно прямо исследовать с целью

обнаружения подходов к защите устройств от получения прав суперпользователя, то в сторонних приложениях такие методы могут быть модифицированы или расширять перечисленные здесь из-за сложности процессов или их закрытости.

Большинство методов распознавания доступа уровня суперпользователя работают только для устройств с «мягким доступом», так как они опираются на изменения в правах доступа и данных приложений непосредственно во время работы операционной системы Android как на показатели для подтверждения наличия прав суперпользователя. При «жестком доступе» расширенные права делегируются на ранних этапах запуска системы тем самым открывая возможность использования, в ином случае недопустимых, функций смартфона и при этом не оставляет следов в среде Android.

Одним из распространенных источников приложений и библиотек для обнаружения доступа суперпользователя с открытым кодом является RootBeer [16].

Другой комплекс мер, широко используемый на Android устройствах это Google’s SafetyNet [17]. Этот комплекс непосредственно интегрирован в операционную систему и служит для самодиагностики устройства на наличие прав суперпользователя в локальном режиме. Собираемая информация о поведении приложений и функций в обобщенном виде обрабатывается в фоновом потоке, не покидая пространства устройства. Информация о методах обнаружения расширенных прав, используемых в этом комплексе, не распространяется, но можно предположить, что он включает в себя некоторые из общедоступных которые будут перечислены далее:

1. Установленные пакеты. Большинство приложений для получения прав суперпользователя устанавливаются как обычные Android приложения. Простейшим подходом проверить имеются ли на устройстве такие права это проверить установлено ли какое-либо из таких приложений.

2. Проверка файлов. Исполнительные файлы для получения расширенных прав могут быть занесены в память устройства напрямую, без установки. Можно проверить их наличие в файловой системе.

3. Соответствие версий. Стандартные образы операционной системы Android помечаются в коде меткой «release-tag», что отображает их как версии для общего пользования. Если же есть другая метка (например «test-tag»), то скорее всего на устройстве установления версия операционной системы, предназначенная для разработчиков или другая неофициальная версия, которая может иметь права суперпользователя.

4. Команда «su» и другие установленные приложения. Приложения «su» чаще всего используется в среде Linux и позволяет использовать команды и права другого пользователя. Если такое приложение удается найти, то пользователь может использовать его для повышения своих прав и получения уровня доступа суперпользователя.

Также другие приложения, обычно не используемые на Android системах, могут быть установлены приложением с расширенными правами. Если такое приложение удастся найти и у пользователя есть возможность доступа к нему, то это тоже показывает наличие таких прав.

5. Проверка системных параметров. Существует набор параметров и расширенных возможностей функций, доступ к которым может быть получен только при наличии прав суперпользователя. Можно проверить, какие значения выставлены у таких параметров и под каким пользователем выполняются команды.

6. Доступ к файловой системе. Большинство приложений для получения прав суперпользователя модифицируют разрешения системных папок, предоставляя к ним доступ другим пользователям. В обычной Android среде, приложения не должны иметь доступ к просмотру файлов других приложений и процессов, но вместо этого взаимодействовать через интерфейс со средой. Если обнаружен доступ к другим приложениям и данным принадлежащим другим процессам, для которых пользователь

предоставляет доступ, то это можно считать еще одним индикатором доступа уровня суперпользователя.

7. Проверка доступных процессов. Этот метод сильно отличается от предыдущих, так как он может быть применен только в момент, когда приложение для получения прав суперпользователя активно. Менеджер активности получает список процессов, которые работают в данный момент, который в дальнейшем можно проверить на наличие процессов с повышенными правами.

Несмотря на большое количество способов проверки наличия расширенных прав и их постоянную актуализацию, методы обхода таких способов также постоянно эволюционируют.

4.2 Дополнительные методы обнаружения

В дополнение к методам обнаружения из предыдущего раздела, далее будут представлены альтернативные методы распознавания, основанные на существующем функционале ядра Linux и дополнительных программах, способных работать на операционной системе Android.

1. Обнаружение сторонних связанных библиотек. Используя предзагрузку библиотек [18], можно внедрить стороннюю библиотеку в любой запускающийся процесс. Например, можно заменить функцию, используемую для открывания файлов из системы. При таком механизме обнаружения прав суперпользователя, при открытии файла с именем, о котором известно, что он связан с приложениями для получения расширенных прав, функция может возвращать ошибку или заданное значение, а для всех остальных файлов будет задано стандартное поведение.

2. Обнаружение отслеживания и вмешательства в процесс. Используя команду «ptrace» (системную команду Linux для отслеживания процессов [19]) приложения сокрытия прав суперпользователя может перехватить каждую команду отслеживаемого приложения и поменять их или извлечь какие-либо значения в процессе работы этой программы. Такой подход

позволяет использовать более широкий спектр методов обнаружения, поскольку можно перехватить не только заранее определенный список функций и вызовов, но и возвращаемые значения функций, которые можно проверять и изменять, чтобы увеличить количество успешно скрытых индикаторов прав суперпользователя.

ЗАКЛЮЧЕНИЕ

По нашему мнению, в настоящее время заинтересованность к уровню доступа суперпользователя может быть в основном у исследователей и разработчиков. Обычному пользователю нет явной необходимости получать такой доступ на устройстве, а разработчики предоставляют все больше продвинутого функционала в виде дополнительных приложений и встроенных в операционную систему функций, которые ранее были доступны только с расширенным доступом. В современных версиях операционной системы Android уже доступна достаточно гибкая настройка внешнего вида интерфейса, а широкий набор управляемых параметров энергопотребления позволяет продлить время жизни батарей таких устройств. Тенденции развития в этом направлении ведут к полному отключению возможности получения прав суперпользователя.

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации №075-01024-21-02 от 29.09.2021 (проект FSEE-2021-0015).

СПИСОК ЛИТЕРАТУРЫ

1. Глобальная статистика StatCounter. URL: https://gs.statcounter.com/vendor-market-share/mobile/worldwide (дата обращения: 27.11.2023)

2. Yan H. Methods for avoiding rooting in Android System. Bachelor Thesis Project. Linnaeus University. Faculty of Technology. Department of Computer Science, 2017.

3. Sun S.-T., Cuadros A., Beznosov K. Android rooting: Methods, detection, and evasion. 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. New York, NY, USA, 2015, pp. 3–14. URL: https://doi.org/10.1145/2808117.2808126 (дата обращения: 28.11.2023)

4. Vidas T., Votipka D., Christin N. All your droid are belong to us: A survey of current android attacks. WOOT'11, August 2011, pp. 81–90.

5. Android rooting: An arms race between evasion and detection / L. Nguyen-Vu, N.-T. Chau, S. Kang, S. Jung // Security and Communication Networks. vol. 2017. 2017.

6. Magisk: The Magic Mask for Android. URL: https://github.com/topjohnwu/Magisk (дата обращения: 29.11.2023)

7. Casati L., Visconti A. The dangers of rooting: data leakage detection in android applications. Mobile Information Systems. vol. 2018. 2018.

8. Project Zero: Mitigations are attack surface, too. URL: https://googleprojectzero.blogspot.com/2020/02/mitigations-are-attack-surface-too.html (дата обращения: 01.12.2023)

9. Security-Enhanced Linux in Android. Android Open Source Project. URL: https://source.android.com/security/selinux/ (дата обращения: 01.12.2023)

10. AppArmor Security Project. https://gitlab.com/apparmor/apparmor/-/wikis/home (дата обращения: 01.12.2023)

11. Hardening Firmware Across the Android Ecosystem. URL: https://security.googleblog.com/2023/02/hardening-firmware-across-android.html (дата обращения: 02.12.2023)

12. Samsung Knox Security Solution. Whitepaper. URL: https://images.samsung.com/is/content/samsung/p5/global/business/mobile/SamsungKnoxSecuritySolution.pdf (дата обращения 04.12.2023)

13. ARM Security Technology. Building a Secure System using TrustZone Technology. URL: https://documentation-service.arm.com/static/5f212796500e883ab8e74531 (дата обращения: 04.12.2023)

14. Shen D. Defeating samsung knox with zero privilege. BlackHat. USA. 2017.

15. Open source module for Xposed Framework that hides root from specific apps. URL: https://github.com/devadvance/rootcloak (дата обращения: 05.12.2023)

16. Simple to use root checking Android library and sample app. URL: https://github.com/scottyab/rootbeer (дата обращения: 05.12.2023)

17. Protect against security threats with SafetyNet. URL: https://developer.android.com/privacy-and-security/safetynet (дата обращения 06.12.2023)

18. Dynamic linker. URL: https://linux.die.net/man/8/ld.so (дата обращения: 06.12.2023)

19. Process trace command. URL: https://linux.die.net/man/2/ptrace (дата обращения: 06.12.2023)