

Взломщик секретов: F6 исследовала новую версию Android-трояна Falcon

f6.ru/blog/falcon-rat

March 11, 2026



Внешние источники

Разрешить установку из этого источника



Ваши личные данные и данные телефона более уязвимы для атак приложений из неизвестных источников. Устанавливая приложения из этого источника, вы соглашаетесь с тем, что несете полную



Pay

Установить приложение?

ОТМЕНА

УСТАНОВИТЬ

В ноябре 2025 года специалисты департамента противодействия финансовому мошенничеству ([Fraud Protection](#)) компании F6 зафиксировали новую волну атак на пользователей в России с использованием банковского Android-трояна Falcon. Впервые об этом специалисты F6 [сообщили](#) в начале февраля 2026-го. По данным F6 на конец февраля, в России насчитывалось более **10 тыс.** Android-устройств, скомпрометированных трояном Falcon.

***Falcon** — вредоносная программа для операционной системы Android, впервые обнаруженная в июле 2021 года. Основана на банковском трояне **Anubis**. Falcon похищает информацию с зараженного устройства с помощью команд, полученных с управляющего сервера. Летом 2022 года ВПО распространяли под видом российских банковских приложений.*

Образцы этой вредоносной программы аналитики департамента киберразведки ([Threat Intelligence](#)) и департамента Fraud Protection обнаружили в начале 2026 года. Исследование показало: по сравнению с версией трёхлетней давности новый Falcon серьёзно усовершенствовали. В приложение добавили модуль VNC для удалённого управления устройством пользователя, возможность отправлять скомпрометированные данные через Telegram – и это лишь малая часть изменений, которые превратили хорошо забытое ВПО в очень опасного монстра.

В отличие от трояна **Mamont** и вредоносных версий легитимного приложения **NFCGate**, которые злоумышленники изначально создавали для кражи денег с банковских счетов пользователей, **Falcon** предназначался в первую очередь для кражи данных. Однако версия Falcon образца 2026 года напоминает высокотехнологичный швейцарский нож, которым киберпреступники могут воспользоваться не только для кражи ключа от квартиры, где «деньги лежат», так и для хищения кошелька, и для шпионажа. Особая опасность новой версии трояна – в том, что для неопытного пользователя его действия, которые приводят к получению контроля над устройством и краже чувствительных данных, малозаметны и порой могут казаться «глюками» в работе Android-смартфона.

Само вредоносное приложение FalconRAT мимикрирует под приложения крупных банков, государственных структур, платёжные сервисы, а также VPN-сервисы, подменяя название приложения и его интерфейс. Falcon нацелен на кражу данных более чем **30** приложений ведущих российских банков и инвестиционных фондов, госсервисов, операторов сотовой связи, маркетплейсов, социальных сетей и мессенджеров, включая зарубежные, магазинов приложений, популярных сервисов бесплатных объявлений, приложения для бесконтактных платежей, заказа такси, покупки билетов и бронирования, приложения российских почтовых и «облачных» сервисов, YouTube, а также VPN.

Проще говоря, вредоносное приложение действует как универсальная отмычка, с помощью которой злоумышленники могут получить доступ к учётным записям пользователей для работы в самых разных сервисах. Для этого Falcon в момент запуска целевых

приложений подменяет изображение на экране, открывая веб-страницу, стилизованную под конкретный сервис. Как только пользователь введёт на такой фейковой странице логин, пароль и код двухфакторной аутентификации, эти данные сразу же отправляются к злоумышленникам.

Аналитики F6 отмечают крайне высокий уровень защиты ВПО от антивирусов. Falcon может удалять антивирусы с устройства сразу после собственной установки, а также препятствует пользователю в использовании встроенных механизмов для анализа устройства и удаления вредоносного приложения. Так что наличие антивируса на Android-устройстве пользователя не гарантирует его безопасности при атаке с использованием Falcon. Защита RAT от антифрод-систем куда слабее: для их обхода разработчики ВПО используют смену названий пакетов и высокий уровень обфускации трояна.

Специалисты F6 исследовали несколько обнаруженных образцов FalconRAT, изучили функционал и возможности ВПО. Результатами исследования делимся в этом блоге.

Процесс установки Falcon RAT

Процесс установки FalconRAT начинается с установки дроппера — приложения, которое инициирует установку непосредственно самого ВПО.

Злоумышленники производят распространяют такие APK-файлы через фишинговые веб-ресурсы, имитирующие под государственные или банковские сервисы, а также мессенджеры. При открытии APK-файла у пользователя запрашивается разрешение на установку приложения из неизвестного источника.



Внешние источники

Разрешить установку из этого источника



Ваши личные данные и данные телефона более уязвимы для атак приложений из неизвестных источников. Устанавливая приложения из этого источника, вы соглашаетесь с тем, что несете полную



Pay

Установить приложение?

ОТМЕНА

УСТАНОВИТЬ

Рисунок 1. Запрос разрешения на установку приложения-дроппера из стороннего источника

Сервис Google Play Protect детектирует ряд образцов дроппера как подозрительное приложение. Однако под воздействием социальной инженерии пользователь может обойти блокировку установки вредоносного приложения и продолжить её.

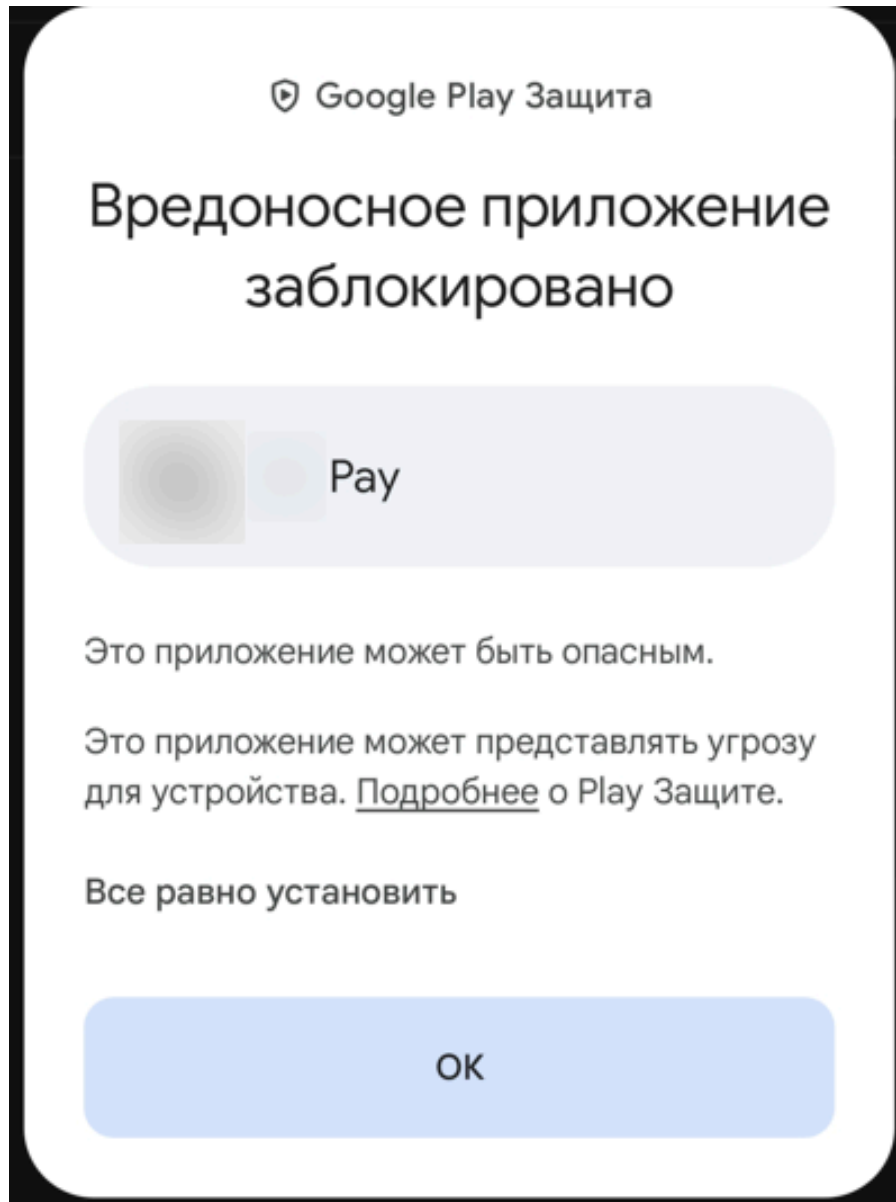
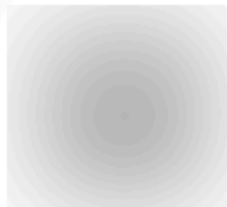


Рисунок 2 – Детектирование с помощью Google Play Protect

Однако далее, после установки дроппера, мимикрирующего под платёжный сервис, встроенные тесты безопасности Android никаких угроз в нём уже не обнаруживают и предлагают пользователю запустить это приложение.

Проверено тестами безопасности

Угрозы не обнаружены



Pay

Версия: 17.18.1.0

Размер: 14.9 МБ

Антивирус

Вирусы не найдены



Проверка фальсифицированных приложений

Приложение является легитимным



Другие уязвимости

Других угроз не обнаружено



Другие приложен...

Открыть

Рисунок 3 – Приложение-дроппер признается безопасным
встроенным антивирусом

Сразу после установки и запуска приложения-дроппера пользователю демонстрируется веб-страница (webview), информирующая о необходимости обновить приложение «***Pay», под видом которого будет загружен APK-файл непосредственно ВПО Falcon.

Обновление Приложения

1. Обновите приложение

Вы используете версию, которая больше не поддерживается. Пожалуйста, обновитесь до последней версии, чтобы продолжать использовать приложение



Загрузить сейчас

2. Разрешите браузеру устанавливать приложения

При установке появится сообщение о неизвестном источнике – это стандартное

Обновление Приложения

1. Обновите приложение

Вы используете версию, которая больше не поддерживается. Пожалуйста, обновитесь до последней версии, чтобы продолжать использовать приложение



Подождите, идет загрузка...

Загрузить сейчас

2. Разрешите браузеру устанавливать приложения

При установке появится сообщение о неизвестном источнике – это стандартное

Рисунки 4-5 – Установка FalconRAT из приложения-дроппера

После нажатия на кнопку загрузки в webview происходит скачивание со стороннего веб-ресурса APK-файла вредоносного приложения, которое сопровождается запросом разрешения на установку из сторонних источников для уже установленного приложения-дроппера. Для пользователя это выглядит как запрос на установку обновления уже установленного приложения платёжного сервиса.

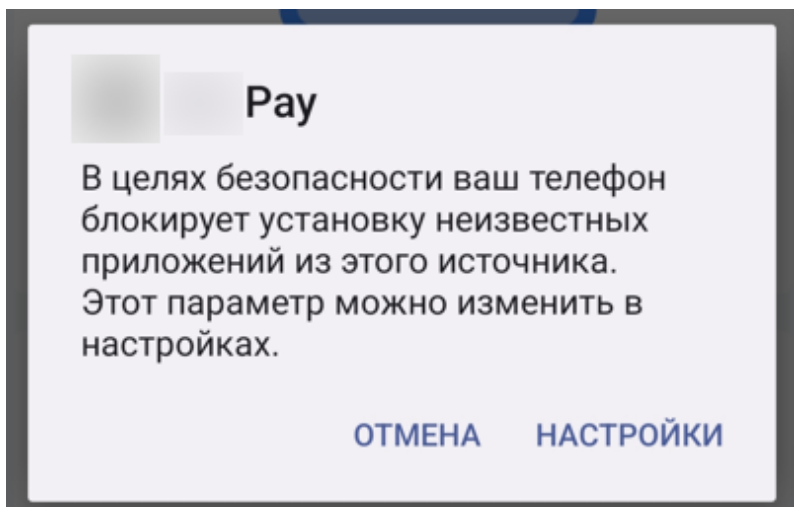


Рисунок 6 – Запрос разрешения на установку FalconRAT из приложения-дроппера

Предоставление разрешений ВПО

Сразу после установки фейкового «обновления» вредоносное приложение инициирует создание push-уведомления раз в несколько секунд, которое начинает появляться на экране и тем самым настойчиво вынуждает пользователя кликнуть на него под предлогом «включения доступа».

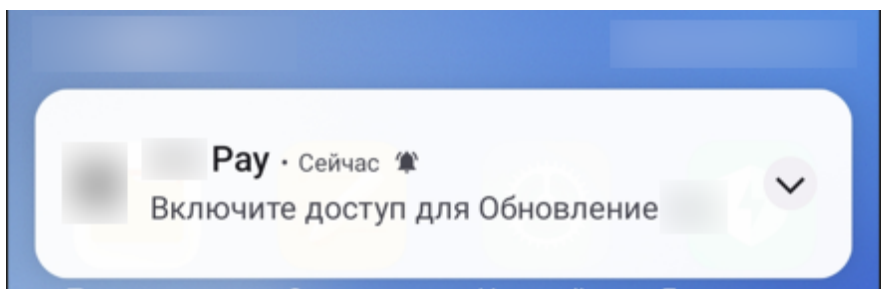


Рисунок 7 – Настойчивое push-уведомление с запросом о включении доступа

После того, как пользователь нажимает на это уведомление, ВПО запрашивает доступ к Accessibility-службам на устройстве.

Android Accessibility – стандартный сервис, встроенный в операционную систему Android набор функций и инструментов, предназначенных для помощи пользователям с ограниченными возможностями.

Перед запросом доступа к Accessibility-службам вредоносное приложение демонстрирует веб-страницу, которая демонстрирует процесс предоставления таких прав. Дизайн веб-страницы визуально схож с дизайном реальной страницы настроек, в которой осуществляется управление службой Accessibility.

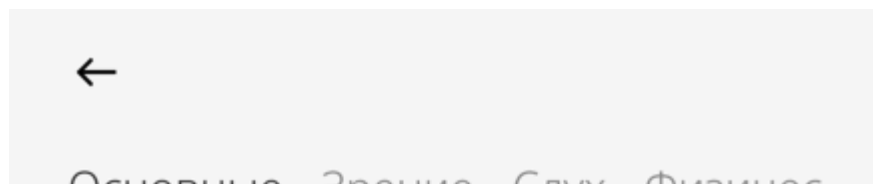


Рисунок 8 – Демонстрация подсказки о включении Accessibility-сервиса

Анализ программного кода ВПО показывает, что веб-страница загружается с управляющего сервера по адресу: [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NWQwZGVhZmMxMWU3MDIjMTFjYzNmZmY3MTA0YmEyMzM1ZTY3ZDVjOQ==](https://hot-mango[.]com/alone/api/api.php?get_manual=NWQwZGVhZmMxMWU3MDIjMTFjYzNmZmY3MTA0YmEyMzM1ZTY3ZDVjOQ==). Значение параметра `get_manual` является зашифрованным с помощью алгоритма RC4 значением `xiaomi_miu_i_9_13_ver`. Кроме данного URI в коде вредоносного мобильного приложения обнаружены следующие URI:

- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NTMwZGZkZmMyM2JiMDk5ZDQ2ZTIIMGNkNWl=](https://hot-mango[.]com/alone/api/api.php?get_manual=NTMwZGZkZmMyM2JiMDk5ZDQ2ZTIIMGNkNWl=) (зашифрованное значение — `vivo_5_13_ver`);
- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NGQxMWVhZTQxOWU3MDk5OTJhODdhNWY3NWY3MWUx](https://hot-mango[.]com/alone/api/api.php?get_manual=NGQxMWVhZTQxOWU3MDk5OTJhODdhNWY3NWY3MWUx) (зашифрованное значение — `huawei_5_13_ver`);
- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NTYwNWU2ZTAwOWUwMzMzMzQwZTlhNzliNzY2MmY2Nzl=](https://hot-mango[.]com/alone/api/api.php?get_manual=NTYwNWU2ZTAwOWUwMzMzMzQwZTlhNzliNzY2MmY2Nzl=) (зашифрованное значение — `samsung_5_13_ver`);
- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NTcwMWVhZmYxMWViMDIjMzA1YzZmOWY3MWM0YmEyMzM1ZTY3ZDVjOQ==](https://hot-mango[.]com/alone/api/api.php?get_manual=NTcwMWVhZmYxMWViMDIjMzA1YzZmOWY3MWM0YmEyMzM1ZTY3ZDVjOQ==) (зашифрованное значение — `realme_oppo_5_13_ver`);
- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NDEwMWVhZmYxOWUyMjMzMzE4ZDdmOGRkNDg3OA==](https://hot-mango[.]com/alone/api/api.php?get_manual=NDEwMWVhZmYxOWUyMjMzMzE4ZDdmOGRkNDg3OA==) (зашифрованное значение — `default_manual`);
- [https://hot-mango\[.\]com/alone/api/api.php?get_manual=NGlwYmY4ZmYxOWViMjMzMzA1ZGFIM2Nm](https://hot-mango[.]com/alone/api/api.php?get_manual=NGlwYmY4ZmYxOWViMjMzMzA1ZGFIM2Nm) (зашифрованное значение — `nosleep_plug`).

По каждому из вышеперечисленных URL находится веб-страница, которая демонстрирует пользователю смартфона конкретного производителя способ включения сервисов Accessibility. Webview с конкретной страницей демонстрируется в зависимости от производителя смартфона пользователя.

Рисунок 9 – Генерирование URL для различных моделей устройств

Рисунки 10-12 – Веб-страницы подсказок для разных моделей смартфонов

Рисунки 10-12 – Веб-страницы подсказок для разных моделей смартфонов

Рисунки 10-12 – Веб-страницы подсказок для разных моделей смартфонов

После просмотра веб-страницы с инструкцией по включению Accessibility-служб происходит перенаправление на настоящую страницу настроек сервиса Accessibility.

Рисунки 13-14 – Включение доступа к службам Accessibility

Перенаправление пользователя на страницу настроек производится путем вызова из JavaScript-кода открытой веб-страницы соответствующего метода Android-приложения, создающего intent (*объект, который действует как посредник между компонентами приложений*) для открытия нужного окна настроек.

Рисунок 15 – Интерфейс взаимодействия с JS на webview

Сразу после того, как пользователь предоставляет ВПО доступ к Accessibility-службам, автоматически происходит запрос и предоставление ряда разрешений, необходимых для работы вредоносного приложения.

Анализ программного кода Falcon RAT показал: механизм автоматического предоставления разрешений, необходимых для работы RAT, основан на поиске среди элементов интерфейса элемента со значением текстового поля «Разрешить» или «Всегда разрешать». При обнаружении такого элемента ВПО выполняет нажатие на него. Таким образом, процесс автоматического запроса и предоставления разрешений для пользователя будет выглядеть как быстрое открытие/закрытие окон в смартфоне в течение нескольких секунд. С высокой вероятностью неопытный пользователь не увидит в этом ничего подозрительного.

Рисунок 16-18 – Код автоматического нажатия на "Разрешить" при запросе разрешения

Рисунок 16-18 – Код автоматического нажатия на "Разрешить" при запросе разрешения

Рисунок 16-18 – Код автоматического нажатия на "Разрешить" при запросе разрешения

Среди предоставляемых автоматическим образом разрешений присутствуют следующие.

- Чтение и отправку SMS («*android.permission.SEND_SMS*», «*android.permission.RECEIVE_SMS*», «*android.permission.READ_SMS*»).
- Выключение оптимизации батареи (*android.settings.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS*).

Рисунок 19 – Запрос разрешений для их автоматического предоставления

Для Xiaomi-устройств реализована отдельная функция: она предоставляет разрешения через приложение «Безопасность», производя автоматический переход в раздел разрешений для установленного ВПО и установку всех разрешений для самого себя в состояние «Всегда разрешать». Для всех остальных приложений происходит перебор элементов разрешений с установкой каждого в состояние «Запретить» (вероятно, для того чтобы отключить необходимые права приложениям, предназначенным для защиты устройства).

Для устройств LG и Xiaomi реализована автоматическая процедура предоставления доступа к использованию батареи устройства без ограничений — тем самым ВПО предоставляет себе возможность по работе в фоновом режиме без риска завершения работы приложения операционной системой.

Рисунок 21 – Разрешение на использование батареи устройства без ограничений

Атака на пользователя. Этап инжекта

Веб-инъект (внедрение кода) – техника атаки, при которой вредоносное ПО внедряет код своей веб-страницы, отображаемый на устройстве жертвы, с целью перехвата данных или подмены информации в реальном времени.

Когда все необходимые ВПО разрешения автоматически предоставлены, приложение инициирует открытие webview-страницы. Запуск WebView осуществляется через FakeBankActivity путем обращения к веб-странице по адресу: `«trello[.]com/1/cards/68430d745c8dc32198d196dd/attachments/68b9899ac099abe8a425ffae/download/ru_***pay.html»`.

Рисунок 22-23 – Запуск webview с фишинговой веб-страницей

Рисунок 22-23 – Запуск webview с фишинговой веб-страницей

На открывшейся веб-странице пользователю предлагается ввести данные карты для прохождения авторизации / прикрепления карты (в зависимости от того, под какое приложение замаскировано ВПО). В рассматриваемом образце данные карты запрашиваются под предлогом ее прикрепления к сервису ***Pay, под который маскируется ВПО.

Рисунок 24-27 – Запрос ввода персональных данных пользователя на фишинговой странице

Рисунок 24-27 – Запрос ввода персональных данных пользователя
на фишинговой странице

Рисунок 24-27 – Запрос ввода персональных данных пользователя
на фишинговой странице

После того, как пользователь вводит данные банковской карты, из JavaScript на веб-странице происходит обращение к методу `saveInjData` мобильного приложения с передачей в качестве параметра строки с введенными пользователем данными.

Рисунок 28: Отправка введенных данных из веб-страницы в Android-приложение

Метод `savelnjData` производит отправку строки с перехваченными данными на управляющий сервер, а также в Telegram-канал злоумышленников, используя API Telegram-бота.

Рисунки 29-31 – Обработка и отправка перехваченных данных пользователя в Telegram-чат злоумышленников и сохранение в локальном хранилище на устройстве

Рисунки 29-31 – Обработка и отправка перехваченных данных пользователя в Telegram-чат злоумышленников и сохранение в локальном хранилище на устройстве

Рисунки 29-31 – Обработка и отправка перехваченных данных пользователя в Telegram-чат злоумышленников и сохранение в локальном хранилище на устройстве

Примечательно, что для маскировки работы RAT на устройстве после отправки украденных данных ВПО открывает новое окно со ссылкой на страницу приложения ***Pay в официальном магазине приложений. Цель злоумышленников при совершении такого действия – заставить пользователя думать, что установленное приложение легитимно. Что, в свою очередь, позволяет предположить: кража чувствительных данных – это ещё не конец атаки, а всего лишь первый её этап.

Рисунок 32 – Перенаправление пользователя на страницу загрузки настоящего приложения ***Pay

Функциональные возможности FalconRAT. Продолжение атаки.

Перехват SMS

Следующий логичный для злоумышленников шаг после кражи данных банковской карты – получение кодов двухфакторной аутентификации. Ранее мы уже видели среди запросов ВПО предоставление разрешений на чтение СМС на устройстве. И действительно, в коде ВПО можно найти перехватчик СМС, срабатывающий при поступлении нового сообщения на устройство.

Рисунок 33 – Перехватчик СМС

Обработка полученного СМС заключается в пересылке на сервер таких его параметров, как:

1. дата;
2. номер отправителя;
3. текст СМС;
4. SIM (на которую пришло сообщение).

Также данные СМС отправляются в ранее упомянутый Telegram-чат через Telegram-бота в виде набора данных:

1. Bot ID (уникальный идентификатор);
2. Number (номер отправителя);

- 3. Text (текст сообщения);
- 4. Stream (SMS MESSENGER)

Отдельно стоит отметить, что ВПО ищет в сообщениях сокращенные ссылки, содержащих домен крупного банка: вероятно, злоумышленники преследуют цель раскрытия персональных данных клиента этого банка.

Перехват телефонных звонков

FalconRAT также производит логирование телефонных звонков путем постоянного прослушивания intent в Android. При обнаружении intent с значением EXTRA_STATE_RINGING происходит запись лога о входящем звонке в формате:

«Дата: 01.01.2026

Номер: 8xxxxxxxxxxx»

Рисунок 34 – Код, отвечающий за логирование телефонных звонков на устройстве

Инжекты

FalconRAT предоставляет злоумышленникам большие возможности по управлению инъектами и, кроме его осуществления сразу после установки приложения, вредоносное приложение способно выполнять следующие функции.

— **Инъект при запуске конкретного приложения.** Когда пользователь запускает приложение, package_name которого был ранее получен, инициируется запуск инъекта.

Рисунок 35 – Запуск инъекта при старте определенного приложения на устройстве

Рисунок 36 – Проверка приложения на соответствие полученному с сервера списку пакетов

— **Инъект по заданной задержке и тегу**, когда тег конкретного образца сравнивается со значением, полученным с сервера. Это значит, что злоумышленники могут рассылать инъекты, заточенные под образцы приложений, мимикрирующих под конкретный сервис. Данный инъект также выполняется с задержкой, передаваемой вместе с настройками инъекта.

Рисунок 37 – Запуск инъекта при совпадении тега после задержки от времени установки FalconRAT

— **Инъект по заданной задержке** идентичен предыдущему, но он не обращает внимания на тег конкретного образца, а лишь выполняется с задержкой, открывая ранее переданный с сервера инъект.

Рисунок 38 – Запуск инъекта после задержки от времени установки FalconRAT

— **Инжект, иницируемый через отправку push-уведомления вредоносным приложением.**

Рисунок 39-40 – Отправка push-уведомления с инжектом

Рисунок 39-40 – Отправка push-уведомления с инжектом

В результате исследования найденных образцов FalconRAT обнаружены ссылки на инжекты, выполненные в виде веб-страниц, которые мимикрируют под государственные и банковские сервисы, а также сервисы VPN-провайдеров. Ссылки на веб-страницы образец ВПО получает с сервера.

Команды с сервера

Функциональные возможности ВПО реализованы через механизм отправки команд с управляющего сервера. Полный перечень обрабатываемых мобильным ВПО команд выглядит следующим образом.

- **sentSms**. Команда осуществляет отправку СМС с устройства жертвы. Она принимает два аргумента: номер адресата и текст отправляемого сообщения, передаваясь в формате: `«sentSms::8xxxxxxxxxx::MessageText»`.
- **spamSms**. Команда осуществляет рассылку СМС всем контактам устройства пользователя с помощью ранее описанной SpamSMSActivity. Формат принимаемой команды: `«spamSms::TEST_SMS»`.
- **forwardAdminPanel**. Команда осуществляет смену управляющего сервера на переданный в аргументе в формате: `«forwardAdminPanel::https://example.com»`
- **startApp**. Команда осуществляет запуск приложения на устройстве пользователя путем отправки intent с соответствующим package_name, получаемым с сервера. Формат команды: `«startApp::com.example.app»`
- **startAlert**. Команда показывает пользователю оповещение. С сервера передаются заголовок окна предупреждения, а также текст. Формат команды: `«startAlert::Заголовок::Предупредительный текст»`.
- **startUssd**. Команда осуществляет звонок или вызов USSD в скрытом для пользователя виде. Формат команды: `«startUssd::8xxxxxxxxxx»`.
- **startInj**. Команда осуществляет запуск инжекта, используя функционал ранее описанного Custom Inject – с сервера вместе с командой передается URL открываемого инжекта. Формат получаемой с управляющего сервера команды: `«startInj::https://trello.com/aaa/test.html»`. Формат intent для открытия Custom activity:

```
«which»:>command_inj»  
«inj»:https://trello.com/aaa/test.html  
«command»:>startInj:https://trello.com/aaa/test.html»
```

- **startPush.** Команда запускает push-уведомление для пользователя, при нажатии на которое осуществляется запуск встроенного инжекта (уже показанного пользователю на этапе установки). Этот функционал предусмотрен на случай, если пользователь преждевременно завершил процесс ввода данных своей карты на этапе установки приложения: в таком случае у злоумышленника есть возможность продемонстрировать инжект повторно.
- **removeApp.** Команда запускает удаление приложения с переданным `package_name`, а также запускает процесс мониторинга повторной установки удаленного приложения: таким образом RAT не позволит снова установить нежелательное для него приложение. Формат команды с сервера: `removeApp::com.example.test`.
- **killBot.** Команда осуществляет самоудаление RAT с зараженного устройства. Вероятно, этот механизм был реализован для сокрытия злоумышленниками следов совершения преступления. Механизм работы заключается в уже описанном ранее процессе удаления приложения: на этот раз он применяется ВПО по отношению к собственному `package_name`.
- **StartVnc.** Команда производит запуск VNC – удаленного доступа к экрану и удаленного управления смартфоном.
- **unlockScreen.** Команда производит разблокировку экрана устройства.
- **changeSms14.** Команда производит запрос на предоставление ВПО прав «SMS-приложения по умолчанию».
- **startForward.** Команда представляет собой реализацию отправки USSD-кода в формате `*21*телефон#`, который является USSD-кодом, реализующим переадресацию вызовов. Формат команды: `startForward::8xxxxxxxxxxx`.

Дальнейшие векторы атаки

Набор серверных команд ВПО позволяет выяснить дальнейшие векторы атаки, которые злоумышленники могут применить в отношении пользователя зараженного устройства. Среди них можно перечислить следующие.

1. Перенаправление СМС и звонков с помощью использования USSD-кодов с целью кражи OTP-кодов и прохождения двухфакторной верификации.
2. Демонстрация новых inject-страниц, мимикрирующих под банковские и государственные сервисы в момент, когда пользователь планирует совершить ввод данных карты на легитимной странице. Это позволит получить от жертвы данные карт других банков и иную чувствительную информацию незаметно для пользователя.

Безграничные возможности по развитию атаки открываются с функционалом удаленного управления VNC и возможностью управления (запуска и удаления) приложений с устройства — такой функционал может быть использован в любых иных целях, включая загрузку дополнительных иных видов ВПО.

FalconRAT предоставляет и возможность рассылки СМС с контролируемого телефона массово или же в индивидуальном порядке, что позволяет мошенникам распространять ВПО знакомым пользователя, находящимся в списке контактов, используя в том числе индивидуальный подход.

Команда `forwardAdminPanel` позволяет осуществлять непрерывный контроль зараженного устройства даже в случае блокировки серверного компонента — злоумышленники могут сменить панель администрирования и продолжить управление устройством уже с другой админ-панели. Одним из вариантов действий злоумышленников может быть продажа зараженных устройств владельцу другой панели.

Техническая реализация FalconRAT

Инициализация ВПО

При запуске вредоносного приложения происходит проверка наличия интернета и завершение приложения при его отсутствии (защита от отладки и проверки в sandbox).

Происходит проверка региона устройства, установленного в параметре «*bot_country_code*» — приложение завершается при значениях AU или US. Исходя из этого можно сделать вывод о том, что злоумышленники избегают атак на пользователей из Австралии и США.

Далее в процессе запуска происходит инициализация ряда параметров Shared Preferences, в которых ВПО хранит информацию необходимую для работы, в том числе:

«*bot_id_tmp*» — Android ID;

«*admin_panel_url*» — панель администрирования ВПО;

«*accessibility_status*» — статус службы Accessibility;

«*dropper_removed*» — статус приложения-установщика (в рассматриваемом образце не использован);

«*installed_apps*» — установленные на устройстве приложения;

«*is_all_settings*» — флаг предназначен для определения факта получения основных настроек с сервера;

«*prev_msg*» — текст последнего полученного СМС;

«*notif_perm_requested_count*» — было ли запрошено разрешение android.permission.POST_NOTIFICATIONS;

«*sms_perm_requested_count*» — были ли запрошены разрешения для перехвата СМС;

«*current_time_plus_time_tuk*» — время на устройстве;

«*bot_logs*» — содержит последний лог, сгенерированный ВПО;

«*bot_injects*» — содержит данные, введенные пользователем на страницах инъектов;

«*bot_sms*» — содержит перехваченного СМС;

«*bot_commands*» — содержит команды, полученные с сервера;

«*injs_open_app*» — содержит URL показываемого пользователю инъекта;

«*interval_tuk_tuk*» — временной интервал обращения к серверу;

«*installed_apps_was_set*» — был ли собран список приложений;

«*link_was_opened*» — была ли открыта ссылка из SMS, присланного банком;

«*packages_need_remove_always*» — список приложений, которые ВПО будет удалять с устройства.

Система логирования

Система логирования в ВПО основана на использовании поля *bot_logs* в SharedPreferences. В данное поле записываются логи в формате:

~—————-(*module_name*)

data

Рисунок 41 – Реализация логирования в FalconRAT

Пример использования системы логирования для:

1. Входящих СМС.

Рисунки 42-43 – Логирование входящих СМС

Рисунки 42-43 – Логирование входящих СМС

Формат лога:

~------(INBOX_SMS)

Дата: <дата получения СМС>

Номер: <Номер отправителя>

Текст: <Текст СМС>

SIM: <Да/нет>

2. Полученных push-уведомлений.

Рисунок 44 – Логирование входящих push-уведомлений

Формат лога:

~------(INTERCEPTED PUSH)

Name: <package_name приложения>

Package: <package_name ВПО>

Title: <Заголовок push>

Text: <Текст push>

Взаимодействие с управляющим сервером

Взаимодействие с управляющим сервером производится мобильным ВПО для отправки похищенных данных пользователя, а также для синхронизации информации об устройстве. Отправка похищенных данных пользователя осуществляется с помощью метода UpdateData().

Рисунок 45 – Функция UpdateData()

Метод UpdateData отправляет на сервер по адресу [https://hot-mango\[.\]com/api/api.php](https://hot-mango[.]com/api/api.php) POST-запрос, содержащий зашифрованную с помощью алгоритма RC4 структуру в формате JSON, которая выглядит следующим образом.

```
{ «bot_id»: <Уникальный идентификатор бота> «inj_codes_open_app»: <Перечень идентификаторов веб-инъектов> «bot_injects»: <Данные карт, введенные жертвой> «bot_logs»: <Логи вредоносного приложения> «bot_sms»: <Перехваченные СМС> «installed_apps»: «not_need_all_settings»: <Нужно ли получение от сервера структуры «all_bots_settings»> }
```

В ответ сервер отправляет JSON-структуру, содержащую обновленные настройки бота, а также новые команды.

```
{ «bot_commands»: <Команды для выполнения ботом> «bot_settings»: { «injects»: <Перечень приложений, при открытии которых будет запускаться инъект> «interepting_sms»: <Вкл./откл. запуска перехвата СМС> «interepting_push»: <Вкл./откл. запуска перехвата push_уведомлений> } «all_bots_settings»: { «inject_links»: <Ссылки на фишинговые страницы для инъектов> «interval_sec»: <Временной интервал для синхронизации с управляющим сервером> «injs_or_urls_open_app»: <Параметры настройки инъекта, запускаемого при открытии приложения> «auto_injs_after_install»: <Параметры настройки автоматического инъекта после установки приложения> «auto_injs»: <Параметры настройки автоматического инъекта по заданному времени> «auto_injs_tag»: <Параметры настройки автоматического инъекта по заданному времени и тегу> «push_injs»: <Параметры настройки push-инъекта> «auto_del_apps»: <Автоматически удаляемые с помощью ВПО приложения> «ask_apps_sms»: <Значение для отсрочки выполнения парсинга приложений и перехвата СМС на устройстве> «auto_sms»: <Номер и текст СМС, отправляемого при запуске ВПО> «track_apps»: <Приложения, для которых ВПО будет перехватывать push-уведомления> } }
```

Тем самым метод реализует механизм обновления настроек и выполнения команд ботом в реальном времени.

Метод Insert_Or_Update_Bot_Info() реализует отправку POST-запроса на [https://hot-mango\[.\]com/api/api.php](https://hot-mango[.]com/api/api.php). Передаваемый запрос содержит зашифрованный по алгоритму RC4 JSON, включающий в себя основную информацию о зараженном устройстве и выглядит следующим образом.

```
“bot_id”: <Уникальный идентификатор бота> “bot_id_tmp”: <Идентификатор android_id устройства> “extra_bot_status”: <Не используется в конечном итоге> “country”: <Страна, где находится устройство> “ip_address”: “android_version”: <Версия Android> “phone_manufacturer”: <Производитель устройства> “phone_model”: <Модель устройства> “mobile_operator”: <Название мобильного оператора> “battery_doze_status”: <Статус настройки оптимизации батареи> “play_protect_status”: <Статус Play Protect>
```

“accessibility_status”: <Статус служб доступности на устройстве> “sms_intercepting_status”: <Предоставлены ли СМС-разрешения> “push_intercepting_status”: <Разрешено ли чтение push других приложений> “tag_status”: «***_1.6.1» (тег образца ВПО) }

Рисунок 46 – Функция Insert_Or_Update_Bot_Info()

Обеспечение собственной безопасности ВПО

ВПО включает в себя также механизмы по обеспечению собственной безопасности.

Примером может служить защита от удаления и остановки работы приложения. Система защиты от удаления реализована таким образом, что на сервер отправляются данные о действиях пользователя, которые могут привести к удалению RAT с устройства. В функции SelfDefense() реализованы такие методы защиты, как:

- Обнаружение элемента с текстовым полем «***Pay». Данное условие работает для смартфонов VIVO и Huawei – на сервер отправляется лог, содержащий сообщение «Risk removing BOT—Зашли в свойства дроппера»
- Если открыты настройки устройства со списком установленных приложений, в таком случае выполняется автоматическое нажатие первой найденной кнопки интерфейса окна со списком приложений (единственная кнопка в нем — кнопка выхода из данного раздела настроек). Тем самым, ВПО не дает просмотреть список приложений на устройстве.
- Обнаружение элемента с текстовым полем «***Pay» в открытом приложении com.sec.android.app.launcher (системный лаунчер для смартфонов на базе **Samsung**). В таком случае на сервер отправляется лог с сообщением «Risk removing BOT—Попытка удаления дроппера».
- Открыты приложения с именами пакетов «com.miui.cleaner» или «com.miui.cleanmaster» (приложение Cleaner для **Xiaomi**-устройств). ВПО мешает их работе, производя выход на главный экран и отправляя лог с сообщением «Risk removing BOT—Зашли в Cleaner»
- Открыто окно свойств приложения на **Xiaomi**-устройстве («com.miui.securitycenter:id/app_manager_details_applabel»). ВПО реагирует на такое действие пользователя перенаправлением его на главный экран и логированием действий с помощью сообщения «Risk removing BOT—Зашли в свойства дроппера».
- Открыто приложение с именем пакета com.android.settings (приложение «Настройки») в стандартной ОС Android и обнаружена строка «***Pay». На основании этого условия ВПО делает вывод об открытии списка приложений в окне настроек и производит выход на главный экран с отправкой лога, содержащего сообщение «Risk removing BOT—Зашли в все приложения, где можно удалить».
- Открыто приложение «com.android.launcher3» и обнаружены элементы окна удаления приложения — в таком случае ВПО отправляет лог «»Risk removing BOT—Попытка удаления дроппера»».

- На экране найдены элементы анимации очистки в системном интерфейсе Android. В таком случае ВПО отправляет лог «Risk removing BOT—Зашли в свойства приложения или Cleaner», а затем происходит переход на главный экран.
- Для **Meizu**-устройств при обнаружении открытого приложения «Настройки», а также элементов, идентифицируемых как размер приложения и название приложения, также делается вывод об открытии полного списка приложений на устройстве и происходит выход на главный экран с отправкой лога «Risk removing BOT—Зашли в свойства приложения или Cleaner».
- Для устройств **Realme** и **Oppo** происходит поиск заголовка окна удаления приложения и кнопки «Нет». При обнаружении таких элементов ВПО производит нажатие кнопки «Нет», препятствуя удалению приложения, а затем отправляет сообщение «Risk removing BOT—Попытка удаления дроппера» в лог.

Рисунок 47 – Реализация самозащиты от удаления

ВПО не только препятствует удалению приложения, но и мешает вызвать меню глобальных действий, из которого можно произвести выключение или перезагрузку приложения. При обнаружении иконки такого меню на экране происходит возврат пользователя на главный экран устройства.

Рисунок 48 – Реализация самозащиты от перезапуска

Также ВПО использует механизм защиты от использования встроенной системы очистки, включающей антивирус – такие распространены в смартфонах китайских производителей. По аналогии с ранее описанными действиями по идентификации определенных элементов на экране устройства происходит процесс поиска текстовых элементов «Антивирус» и «Очистка»: при их обнаружении происходит переход на домашний экран с отправкой на сервер сообщения «Risk removing BOT—Зашли в Cleaner».

Рисунок 49 – Препятствование запуску встроенного антивируса

ВПО блокирует действия пользователя по переходу в раздел настроек сервиса Accessibility. Как и в ранее описанных методах самозащиты, ВПО принудительно закрывает окно настроек при обнаружении своего названия на экране при активном окне настроек сервисов доступности, а также ряда других специфичных для данного меню элементов экрана.

Рисунок 50 – Защита от отзыва прав на Accessibility-сервис

VNC

VNC-сервис ВПО реализован на основе WebSocket-взаимодействия с другим хостом — `wss://go.hot-labubu[.]com:8096`, и имеет собственный интерфейс взаимодействия с этим сервером.

Обработка данных, переданных с сервера, происходит в функции `handleClientCommand` на основе системы команд. VNC-сервис способен обрабатывать следующие команды.

- *tap_rel*. Данная команда является реализацией нажатия на элемент экрана, расположенный по переданным координатам.
- *swipe_rel*. Данная команда является реализацией свайпа на устройстве с переданными параметрами.
- *UNLOCK_SCREEN*. Команда осуществляет разблокировку экрана путем имитации выполнения свайпа с фиксированными параметрами.
- *SHOW_OVERLAY*. Команда осуществляет создание черного окна поверх всего экрана, скрывая от пользователя происходящее на нем.
- *HIDE_OVERLAY*. Команда противоположна предыдущей и позволяет убрать черное окно, перекрывающее экран.
- *IMAGE_ON*. Команда осуществляет включение способа перехвата экрана №1.
- *IMAGE_OFF*. Команда осуществляет выключение способа перехвата экрана №1.
- *IMAGE2_ON*. Команда осуществляет включение способа перехвата экрана №2.
- *IMAGE2_OFF*. Команда осуществляет выключение способа перехвата экрана №2.
- *CLOSE_CONNECTION*. Команда вызывает завершение соединения VNC с сервером.
- *REQUEST_VNC_PERMISSION*. Команда вызывает запрос разрешения у пользователя на захват экрана для работы VNC.
- *REQUEST_OVERLAY_PERMISSION*. Команда вызывает запрос у пользователя разрешения открывать приложение поверх других окон.
- *COPY_TO_ANDROID*. Команда осуществляет вставку переданного с сервера текста в активную форму ввода.

Рисунки 51-52 – Команды VNC-сервиса FalconRAT

Рисунки 51-52 – Команды VNC-сервиса FalconRAT

Функционал VNC-сервиса реализован тремя способами.

— 1. **Постоянное создание скриншотов экрана** (ВПО поддерживает функционал для Android 13 и выше).

Рисунок 53 – Удаленное управление экраном смартфона через постоянные снимки экрана устройства

— 2. **Непрерывная передача кадров захваченного экрана** (требуется предоставления разрешения пользователем).

Рисунки 54-55 – Функционал VNC, реализованный через классический перехват экрана

Рисунки 54-55 – Функционал VNC, реализованный через классический перехват экрана

Также данный способ для сохранения фоновой работы ВПО создает постоянное уведомление, сообщающее об обновлении.

— 3. Парсинг текстовых элементов на экране.

Рисунки 56-58 – Перехват текстовых элементов с экрана

Рисунки 56-58 – Перехват текстовых элементов с экрана

Рисунки 56-58 – Перехват текстовых элементов с экрана

Как можно заметить, функция `getJSONArrayOfUIElems()` осуществляет рекурсивный парсинг текстовых элементов — их содержимого, названия класса, расположения и размера. В дальнейшем собранная на основе этих данных JSON-структура отправляется на сервер с помощью WebSocket-соединения. На серверной стороне эти данные могут быть использованы для восстановления исходного состояния экрана.

Таким образом наличие функционала VNC, способного функционировать в трех режимах работы, становится одним из мощнейших инструментов злоумышленников, включенных в FalconRAT. Дополнительный функционал управления overlay – перекрывающим экран пользователя черным окном — позволяет злоумышленникам скрыть от пользователя свои действия и воспрепятствовать попыткам пользователя помешать использованию VNC. Функция вставки текста делает более удобным ввод данных в формы ввода, тем самым ускоряя процесс проведения мошеннических действий на устройстве.

Как защититься от FalconRAT

Рекомендации специалистов F6 для пользователей

- **Не общайтесь** в мессенджерах с неизвестными, кем бы они ни представлялись: сотрудниками банков, операторов почтовой и сотовой связи, госсервисов или коммунальных служб.
- **Не переходите** по ссылкам из СМС и сообщений в мессенджерах, даже если внешне они похожи на сообщения от банков и других официальных структур.
- **Не устанавливайте** приложения по рекомендациям незнакомцев, по ссылкам из СМС, сообщений в мессенджерах, писем и подозрительных сайтов. Устанавливайте приложения только из официальных магазинов приложений, таких как RuStore и GooglePlay. *Перед установкой обязательно проверяйте отзывы о приложении, обращайте особое внимание на негативные отзывы – это поможет определить фейковые и потенциально опасные программы.*
- Если вам предлагают установить или обновить приложение банка и присылают ссылку, позвоните на горячую линию, указанную на официальном сайте банка, и уточните, действительно ли полученное вами предложение исходит от банка.

- **Не сообщайте посторонним CVV и ПИН-коды** банковских карт, логины и пароли для входа в онлайн-банк. **Не вводите эти данные** на незнакомых, подозрительных сайтах и в приложениях, которые устанавливаете впервые.
- Если вы понимаете, что ваша банковская карта скомпрометирована, сразу же заблокируйте её, позвонив на горячую линию банка, или с использованием банковского приложения.
- **Не удаляйте** банковские приложения из телефона по запросу третьих лиц. Банковские приложения имеют встроенную защиту от мошеннических атак и могут защитить вас от действий преступников.
- Не предоставляйте чувствительных разрешений приложениям, если не понимаете, с какой целью приложение его запрашивает. Особое внимание рекомендуется обратить на разрешение на доступ к службе Accessibility («Специальные возможности») — FalconRAT демонстрирует, что одно такое разрешение способно дать вредоносному приложению возможность предоставить самому себе любые другие разрешения.
- Проверить устройство на наличие подозрительных приложений рекомендуется, если наблюдается странное поведение смартфона — автоматический запуск / завершение работы приложений, автоматическое сворачивание приложений и окон, перенаправление на домашний экран. Все вышеперечисленные признаки могут свидетельствовать об удаленном управлении или автоматизации работы вредоносного ПО с устройством.

Рекомендации специалистов F6 для подразделений информационной безопасности банков

- Учитывать данные геолокации пользователей.
- При использовании СМС для отправки OTP-кода реализовать механизмы проверки приложения, принимающего СМС на устройстве пользователя.
- Реализовать дополнительные меры защиты на устройстве пользователя по выявлению сторонних вредоносных приложений.
- Реализовать меры по выявлению приложений, имеющих доступ к службам Accessibility.

Защититься от подобных операций банкам помогают антифрод-решения с анализом сессионных и поведенческих данных, а также технологии выявления подозрительной активности, способные анализировать как отправителя, так и получателя платежа.

Например, для оценки риска транзакции и проверки получателя денег применяется комплексный подход с помощью линейки продуктов [Fraud Protection от F6](#).

Индикаторы компрометации

MD5: e2860a08f9e98dba438071a08c700f6a

SHA-1: 30623bce9970f2c9804093d8be65e7920618636a

SHA-256: 99e9c7df033f694e8174a772c8d296a24d108c02e090560627666d36491511a6

Filename: m.apk

MD5: fc9fc8e23b7fee15c94816953c91cb2f

SHA-1: 5409de35cc29575af2a24d976a1d59bde4048e0c

SHA-256: 6958b411b2b910452a2cd3e3a73c2534f1ccc6131952dddb041d5fed5f67d246

Filename: app333.apk

MD5: af320d7eb443fa6cf05295e48944d7c1

SHA-1: 0768b9d3f9dc2b6268f2526023ca7e5e4ffbac86

SHA-256: 113a970413e3927766036cbfb90b2d05e38b9440ac067a69a380cb7fcb067c38

MD5: fe81ae304147b3b7174d13168d18f9e7

SHA-1: d881144b20aa13945e83fdf87440f9a5b5b3267a

SHA-256: 5cd0e81c954fdeea04c03bcd89ecc009e0cbb372ac15aafc19b359440b8933dd

© 2003 – 2025 F6 — российский разработчик технологий для борьбы с киберпреступлениями.