Something to Remember Us By: Device Confiscated by Russian Authorities Returned with Monokle-Type Spyware Installed - The Citizen Lab

Archived: 2026-04-05 21:51:09 UTC

Key Findings

- This joint investigation with First Department, a legal assistance organization, found spyware covertly implanted on a phone returned to a Russian programmer accused of sending money to Ukraine after he was released from custody.
- He describes being subjected to beatings and an intense effort to recruit him as an informant for the Russian Federal Security Service (FSB).
- Our analysis finds that the spyware placed on his device allows the operator to track a target device's location, record phone calls, keystrokes, and read messages from encrypted messaging apps, among other capabilities.
- The spyware bears many similarities to the Monokle family of spyware, previously reported on by Lookout Mobile Security, which they attribute to the "Special Technology Center," a contractor to the Russian government.
- Our analysis also finds certain differences from previously-reported samples of Monokle spyware, suggesting that it is either an updated version of Monokle or new software created by reusing much of the same code.

Read the full report by The First Department here and watch the video.

Introduction

The First Department is a legal assistance organization founded by exiled Russian human rights lawyer Ivan Pavlov that specializes in defending those accused of treason and espionage in Russia. Pavlov left Russia in September 2021 after facing persecution for his legal work. The First Department plays an essential role in supporting individuals targeted for repression by the Russian government. The organization has been headed by Dmitry Zair-Bek since May 2022.

In June 2024, The First Department received a report from Kirill Parubets, a Russian programmer who was released from a 15-day period in administrative detention by Russian authorities. Parubets, who consented to being named in this report, was accused of engaging in money transfers to Ukraine. His Android device had been confiscated at the time his apartment was searched, during which he was subjected to beatings, among other things, to compel him to disclose his device password.

Both Parubets and his spouse were taken into custody and detained. During his detention, Parubets describes being subjected to an intense effort at recruitment as an informant by Russia's Federal Security Service (the FSB).

He was threatened with life imprisonment if he failed to cooperate. The recruitment effort suggests a focused and ongoing interest by FSB in his work and contacts, including in Ukraine.

Following Parubets' release from detention, his device was returned to him at the Lubyanka building, the FSB's headquarters. Parubets quickly began observing unusual behavior, including a suspicious notification "*Arm cortex vx3 synchronization*" on the device, which was an Oukitel WP7 running Android 10. This notification is not a standard notification on this device.

Working with Parubets and his spouse, The First Department examined the device and identified a likely-malicious app that he had not installed, and that appeared to have been introduced onto the phone during his detention. The First Department subsequently contacted the Citizen Lab for assistance with technical analysis.

Technical Analysis

Our analysis confirms that the application identified by The First Department is malicious, and that it appears to be a trojanized version of the *genuine* Cube Call Recorder application. The *genuine* (non-malicious) Cube Call Recorder is an app listed in the Google Play Store that is designed to allow an individual to automatically record incoming phone calls, as well as calls within messaging apps.

The SHA sum of the malicious version of the app:

```
 Malicious App SHA-256
737f60749c1919ad22102be27d52ba199ec4b707a985c42011b22ce0a4512c90
```

Spyware Functionality

First Stage

There are some hints about the functionality of the spyware in the permissions requested by the trojanized app. The spyware requests many permissions that the legitimate version of the application does not, including:

- Access to location information when the application is not in use
- Read and send SMS messages
- Install additional packages
- Read calendar entries
- Record screen captures
- List other applications on the device
- Answer phone calls
- Get account details
- Record video with the camera

The spyware also shares several permissions with the legitimate application (which are also common to spyware) such as:

- Accessing precise location
- Recording phone calls
- Getting information about the target's contacts

Permission	**Trojanized App**	**Legitimate App**
Accessing fine location	✅	✅
Recording phone calls	✅	✅
Getting information about the target's contacts	✅	✅
Access to location information when the application is not in use	✅	❌
Read and send SMS messages	✅	❌
Install additional packages	✅	❌
Read calendar entries	✅	❌
Record screen captures	✅	❌
List other applications on the device	✅	❌
Answer phone calls	✅	❌
Get account details	✅	❌
Record video with the camera	✅	❌

Table 1
Differences in permissions between the spyware and the legitimate application it is disguised as.

Most of the malicious functionality of the application is contained in the class `com.android.twe1ve`, a class that is unique to this sample of spyware and not present in the Cube Call Recorder app available in the Google Play Store.

Most of the malicious functionality of the application is hidden in an encrypted second stage of the spyware. Once the spyware is loaded onto the phone and executed, the second stage is decrypted and loaded into memory. This type of obfuscation can help hide malicious activity from some antivirus software.

The second stage is a dex file encrypted using simple XOR encryption with a static repeating key. The second stage is stored in a data file called `license` located in the `assets` directory of the unpacked apk file. The java class `com.catalinagroup.callrecorder.App` loads `lib/arm64-v8a/library.so`, which provides functionality for `com.system.info.Info` to unpack the second stage.

Java code to load the native ARM library which is responsible for unpacking the second stage of the spyware:

```
static {       System.loadLibrary("rary");    }
```

The app then calls into the loaded library to extend the app by attaching assets/library as a base context.

Java code to load the decrypted license file into memory in the context of the trojanized application:

```
    public void attachBaseContext(Context context) {       Info.get(context, "license");        super.attachBas
```

Second Stage

The second stage of the spyware contains additional core Android application libraries in the `com.android.twe1ve` class, as well as importing other common cryptography and Android libraries. It also includes several open source software libraries: an RTMP for real time audio/video streaming, and an SMB library presumably for uploading files taken from the device.

The second stage contains many common spyware capabilities, including:

- Location Tracking
- Screen capture
- Keylogging
- Recording calls
- Extracting files from the device
- Extracting stored passwords
- Reading messages from other messaging apps
- Adding a new device administrator
- Injection of Javascript
- Executing shell commands

- Extracting the device unlock password

It also contains functionality for decrypting settings and data files which are also stored in the assets directory in seemingly randomly named files.

Interestingly, we find several references to iOS in the code, suggesting the possibility of an iPhone version of this spyware.

Reference to iOS permissions in the settings code:

```
MwBi.MwLBLiL = new MwIN.MwKuK.MwKuK.MwIN.Mwuk("settingsName", 11, 2); MwBi.MwiB = new MwIN.MwKuK.MwKuK.MwIN.Mwu
```

There are also commands from the command and control infrastructure referencing iOS: “ `ShowiCloudLogin` ”, and “ `GetHealthKit` .” These are the same references to iOS which were originally reported by Lookout in 2019.

Technical experts at The First Department suspected that this spyware might be related to the Monokle family of spyware, originally reported on by Lookout in 2019. Lookout described Monokle as advanced mobile spyware with connections to Russian threat actors. At that time, Lookout linked Monokle to Special Technology Center, Ltd., a company based in St. Petersburg, Russia.

Throughout the analysis of the sample provided by The First Department we found key similarities to the original Monokle spyware sample, but also some differences, leading us to assess that this is either an updated version of Monokle, or that it has been created by reusing much of the original Monokle code.

Command & Control Similarities

The most compelling evidence that the app installed on the individual’s device is related to the Monokle sample from the 2019 Lookout report is the overlap in the commands issued by the command and control server, including many of the same exact strings. This sample and the 2019 sample both also use the string `BaseSystemCommand` as the prefix for all command strings, which appears to be unique to these two samples.

Our Sample	Lookout Monokle
`BaseSystemResponse_ExecuteShellCommand`	`baseSystem.executeShellCommand.`
`BaseSystemResponse_GetApplicationsList`	`baseSystem.getApplicationsList`
`BaseSystemResponse_GetCallsList`	`baseSystem.getCallsList`
`BaseSystemResponse_GetLocation`	`baseSystem.getLocation`
`BaseSystemResponse_GetScreenPassword`	`baseSystem.getScreenPassword`
`BaseSystemResponse_GetSmsList`	`baseSystem.getSmsList`
`BaseSystemResponse_InstallCertificate`	`baseSystem.installCertificate`
`BaseSystemResponse_GetKeyLogging`	`baseSystem.getKeyLogging`
`BaseSystemCommand_InstallApplication`	`baseSystem.installApplication`
`BaseSystemCommand_SetAudioRecordMode`	`baseSystem.setAudioRecordMode`

Table 2
Selected similarities between command and control commands.

Additionally, the same iOS-related commands present in this sample were also observed by Lookout in their 2019 report.

Additional Similarities

There are additional similarities between the sample identified by The First Department and the 2019 Monokle spyware sample. However, these additional similarities include several common tactics of spyware and would not be as significant on their own without the unique Command & Control overlaps.

Use of Similar Folders for Malware Staging

The sample identified by The First Department uses the `assets` folder for storing other stages of spyware and settings, and decrypting that data with a static repeating XOR key. This is the same TTP used by Monokle according to the report from Lookout.

Use of Accessibility Settings and Other Similarities

The sample also makes use of accessibility settings, a feature noted in the Lookout report. Many of the other capabilities present in this sample such as geofencing, streaming audio, gathering health kit data, and recording the unlock screen password are all present in Lookout's reporting on Monokle as well.

Trojanization/Hijacking of Legitimate Applications

This spyware was packaged as a backdoored version of a legitimate application, which is a common technique. Monokle was also typically packaged as a trojanized version of a legitimate application.

Differences with the Lookout Monokle Sample

Although the analysis found numerous similarities between this sample and the original reporting on Monokle, there are also some differences that are important to mention. The names of the specific files stored in the assets folder have changed and the encryption of the configuration file is more sophisticated than the 2019 sample. The new sample uses a different key than is used for the second stage, making it much more difficult to decrypt and extract additional Command and Control information.

Some of the permissions have changed as well. The app now requests new permissions such as "`ACCESS_BACKGROUND_LOCATION`", "`INSTALL_PACKAGES`", and "`LOCAL_MAC_ADDRESS`". Many third-party application-specific permissions, such as "`org.thoughtcrime.securesms.ACCESS_SECRETS`", and "`com.android.browser.permission.READ_HISTORY_BOOKMARKS`" have been removed. Some Android permissions such as "`USE_FINGERPRINT`", and "`SET_WALLPAPER`" have also been removed.

However, even with these changes, the many significant similarities in operations, functionality, and geopolitical motivations lead us to assess that this is either an updated version of the Monokle spyware or new software created by reusing much of the same code.

Implications of Device Tampering

It is common for the FSB to engage in targeted digital surveillance against individuals they perceive as threats, such as the use of sophisticated social engineering to steal credentials as described in the Rivers of Phish campaign the Citizen Lab uncovered in partnership with Access Now and multiple regional civil society organizations. Malicious activities that target individuals across the globe often rely on tricking a user into engaging with the attackers. However, the tactics often change when an individual is within physical proximity of the attackers.

Detention and device confiscation can provide an unique opportunity for an adversary to install spyware without the same technical challenges presented by remote attacks. This opportunity is especially pronounced if the adversary has user-level access to the device and is able to compel the individual to provide credentials and/or device passcodes, as they were in this case.

This case illustrates that the loss of physical custody of a device to a hostile security service like the FSB can be a severe risk for compromise that will extend beyond the period where the security services have custody of the device. In this case, the target noticed several odd behaviors on their device after he was released from detention, such as an unfamiliar and suspicious notification and the presence of an app that he had not installed. However, not every attempt to infiltrate and monitor a device is likely to result in such visible alerts.

We encourage members of civil society that have lost physical custody of their device to a security service, especially a technically competent service in an authoritarian state like Russia, to seek expert assistance when the device is returned to them. Any person whose device was confiscated and later returned by such services should assume that the device can no longer be trusted without detailed, expert analysis.

Acknowledgements

We first wish to acknowledge the bravery of Kirill Parubets for coming forward and sharing the details and samples with The First Department and The Citizen Lab. Thanks to Dmitry Zair-Bek and The First Department for their assistance in this investigation. Thanks to Lookout for their original reporting on Monokle and for sharing additional findings to support this research.

We thank our colleagues at The Citizen Lab for assistance with preparing, editing, and reviewing this report, with special thanks to Bahr Abdul Razzak, Adam Senft, Siena Anstis & Alyson Bruce. Professor Ron Deibert is the principal investigator of the Citizen Lab and this project was undertaken under an approved University of Toronto research ethics protocol # 37346, “Comparative Analysis of Information Security Threats Experienced by Civil Society.”

Appendix – Indicators of Compromise

SHA-256 Sum

```
737f60749c1919ad22102be27d52ba199ec4b707a985c42011b22ce0a4512c90
```

Commands sent by the C2 Server

`BaseSystemCommand_ApplyAgentUpdate`

`BaseSystemCommand_ClearResults`

`BaseSystemCommand_DeleteFile`

`BaseSystemCommand_DeviceControl`

`BaseSystemCommand_DeviceReset`

`BaseSystemCommand_ExecuteShellCommand`

`BaseSystemCommand_GetAccessibility`

`BaseSystemCommand_GetAgentInfo`

`BaseSystemCommand_GetAppUsageStatsList`

`BaseSystemCommand_GetApplicationsList`

`BaseSystemCommand_GetCallsList`

`BaseSystemCommand_GetContactsList`

`BaseSystemCommand_GetDeviceInfo`

`BaseSystemCommand_GetEmailsList`

`BaseSystemCommand_GetFile`

`BaseSystemCommand_GetFilesList`

`BaseSystemCommand_GetHealthKit`

`BaseSystemCommand_GetInstantChatsList`

`BaseSystemCommand_GetKeyLogging`

`BaseSystemCommand_GetLocalSettingsList`

`BaseSystemCommand_GetMeetingsList`

`BaseSystemCommand_GetMmsList`

`BaseSystemCommand_GetNotesList`

`BaseSystemCommand_GetPreparedTaskResultsData`

`BaseSystemCommand_GetRegistryKeysList`

`BaseSystemCommand_GetSmsList`

`BaseSystemCommand_InjectJS`

`BaseSystemCommand_InstallApplication`

`BaseSystemCommand_InstallCertificate`

`BaseSystemCommand_MakeCall`

`BaseSystemCommand_PrepareFileArchive`

`BaseSystemCommand_ScheduleConnection`

`BaseSystemCommand_SendSms`

`BaseSystemCommand_SetAccessibility`

`BaseSystemCommand_SetAgentSettings`

`BaseSystemCommand_SetAgentUid_deprecated`

`BaseSystemCommand_SetApplicationRestriction`

`BaseSystemCommand_SetAudioListenMode`

`BaseSystemCommand_SetAudioRecordMode`

`BaseSystemCommand_SetAudioStreamingMode`

`BaseSystemCommand_SetCallDropMode`

`BaseSystemCommand_SetCallRecordMode`

`BaseSystemCommand_SetCallbackMode`

`BaseSystemCommand_SetCatchFiles`

`BaseSystemCommand_SetCommunicationMode_deprecated`

`BaseSystemCommand_SetConnectPeriod_deprecated`

`BaseSystemCommand_SetControlPhones_deprecated`

`BaseSystemCommand_SetEventActions`

`BaseSystemCommand_SetFileCrypto_deprecated`

`BaseSystemCommand_SetGeofencesList`

`BaseSystemCommand_SetInstantChatAccumMode`

`BaseSystemCommand_SetKeyLogging`

`BaseSystemCommand_SetKeychain`

`BaseSystemCommand_SetLocationTracking`

`BaseSystemCommand_SetPhotoShotMode`

`BaseSystemCommand_SetScreenCastRecordMode`

`BaseSystemCommand_SetScreenPasswordMode`

`BaseSystemCommand_SetScreenRecordMode`

`BaseSystemCommand_SetScreenShotMode`

`BaseSystemCommand_SetServerAddress_deprecated`

`BaseSystemCommand_SetTransportCrypto_deprecated`

`BaseSystemCommand_SetUsbTunnelPort_deprecated`

`BaseSystemCommand_SetVideoRecordMode`

`BaseSystemCommand_SetVideoStreamingMode`

`BaseSystemCommand_SetWatchFolders`

`BaseSystemCommand_ShowMessage`

`BaseSystemCommand_ShowiCloudLogin`

`BaseSystemCommand_SqlQuery`

`BaseSystemCommand_StopScheduledTasks`

`BaseSystemCommand_ToggleBluetooth`

`BaseSystemCommand_ToggleGPS`

`BaseSystemCommand_ToggleWifi`

`BaseSystemCommand_UninstallApplication`

`BaseSystemCommand_UploadFileToAgent`

`BaseSystemResponse_CancelAllCommands`

`BaseSystemResponse_Error`

`BaseSystemResponse_ExecuteShellCommand`

`BaseSystemResponse_GetAccessibility`

`BaseSystemResponse_GetAccountsList`

`BaseSystemResponse_GetAgentInfo`

`BaseSystemResponse_GetAppUsageStatsList`

`BaseSystemResponse_GetApplicationsList`

`BaseSystemResponse_GetBrowserBookmarks`

`BaseSystemResponse_GetBrowserHistory`

`BaseSystemResponse_GetBrowserTracking`

`BaseSystemResponse_GetCallsList`

`BaseSystemResponse_GetCapabilities`

`BaseSystemResponse_GetContactsList`

`BaseSystemResponse_GetDeviceInfo`

`BaseSystemResponse_GetEmailsList`

`BaseSystemResponse_GetEventTracking`

`BaseSystemResponse_GetFile`

`BaseSystemResponse_GetFilesList`

`BaseSystemResponse_GetGeofencesList`

`BaseSystemResponse_GetHealthKit`

`BaseSystemResponse_GetInstantChatsList`

`BaseSystemResponse_GetInterfacesStates_deprecated`

`BaseSystemResponse_GetJSOutput`

`BaseSystemResponse_GetKeyLogging`

`BaseSystemResponse_GetKeychain`

`BaseSystemResponse_GetLocalSettingsList`

`BaseSystemResponse_GetLocation`

`BaseSystemResponse_GetLocationTracking`

`BaseSystemResponse_GetMMSList`

```
BaseSystemResponse_GetMeetingsList
```

```
BaseSystemResponse_GetNetworkingData_deprecated
```

```
BaseSystemResponse_GetNotesList
```

```
BaseSystemResponse_GetNotificationsList_deprecated
```

```
BaseSystemResponse_GetPreparedTaskResultsList
```

```
BaseSystemResponse_GetRegistryKeysList
```

```
BaseSystemResponse_GetSMSList
```

```
BaseSystemResponse_GetScheduledTasksList
```

```
BaseSystemResponse_GetScreenPassword
```

```
BaseSystemResponse_GetUserDictList
```

```
BaseSystemResponse_SetAudioRecordMode
```

```
BaseSystemResponse_SetScreenRecordMode
```

```
BaseSystemResponse_SetVideoRecordMode
```

```
BaseSystemResponse_SqlQuery
```

```
BaseSystemResponse_UploadFileToAgent
```

Fields in Data and Settings Files

```
      AGENT_SETTINGS(1, "agentSettings"),
```

```
       SERVICE_KILLED(2, "serviceKilled"),
```

```
       RADIO_INFO(3, "radioInfo"),
```

```
       TURN_GPS_ON(4, "turnGpsOn"),
```

```
       LOCATION_TRACKING_ON(5, "locationTrackingOn"),
```

```
       LOCATION_TRACKING_PERIOD(6, "locationTrackingPeriod"),
```

```
       HAVE_SCREEN_CAP_PERMISSION(7, "haveScreenCapPermission"),
```

```
       KEY_LOGGING_MODE(8, "keyLoggingMode"),
```

```
       ACCESSIBILITY_MODE(9, "accessibilityMode"),
```

```
       ACCESSIBILITY_MASKS(10, "accessibilityMasks"),
```

```
    SCREEN_UNLOCK_HOOK(11, "screenUnlockHook"),
    SCREEN_CAST_RECORD_PARAMS(12, "screenCastRecordParams"),
    SCREEN_SHOTS_SETTINGS(13, "screenShotsSettings"),
    PHOTO_SHOT_SETTINGS(14, "photoShotSettings"),
    PHOTO_SHOTS_CURRENT_QUANTITY(15, "photoShotsCurrentQuantity"),
    CURRENT_AUDIO_TASK(16, "currentAudioTask"),
    CURRENT_VIDEO_TASK(17, "currentVideoTask"),
    CURRENT_AUDIO_LISTEN_TASK(18, "currentAudioListenTask"),
    LEVEL_SETTINGS(19, "levelSettings"),
    GEOFENCES(20, "geofences"),
    SCHEDULED_COMMANDS(22, "scheduledCommands"),
    COMMANDS(23, "commands"),
    SCHEDULED_COMMANDS_ID_TIME(24, "scheduledCommandsIdTime"),
    LAST_COMMAND_ID(25, "lastCommandId"),
    EVENT_ACTION_LIST(26, "eventActionList"),
    INSTANT_CHAT_ACCUMULATE_MODE(27, "instantChatAccumulateMode"),
    CALL_RECORD_MODE(28, "callRecordMode"),
    CALL_RECORD_SOURCE_PHONE(29, "callRecordSourcePhone"),
    CALL_RECORD_SOURCE_IM(30, "callRecordSourceIM"),
    RECORD_CALL_MASKS(31, "recordCallMasks"),
    DROP_CALL_MASKS(32, "dropCallMasks"),
    APPLICATION_RESTRICTION_LIST(33, "applicationRestrictionList"),
    NEED_IMMEDIATELY_CONNECTION_TIME(34, "needImmediatelyConnectionTime"),
    WATCH_FOLDERS(35, "watchFolders"),
    CATCH_FILES(36, "catchFiles"),
    LAST_DEVICE_ON_TIME(37, "lastDeviceOnTime"),
```

```
        TASK_ID_CALL_RECORD(50, "taskIdCallRecord"),
        TASK_ID_CALL_DROP(51, "taskIdCallDrop"),
        TASK_ID_SCREEN_PASSWORD(52, "taskIdScreenPassword"),
        TASK_ID_KEYLOGGING(53, "taskIdKeylogging"),
        TASK_ID_LOCATION_TRACKING(54, "taskIdLocationTracking"),
        TASK_ID_ACCESSIBILITY(55, "taskIdAccessibility"),
        RECS__AUDIO(100, "RECS_AUDIO"),
        RECS__PHOTO(101, "RECS_PHOTO"),
        RECS__VIDEO(102, "RECS_VIDEO"),
        RECS__SCREEN__SHOT(103, "RECS_SCREEN_SHOT"),
        RECS__RESERVED(104, "RECS_RESERVED"),
        RECS__ACCESSIBILITY(105, "RECS_ACCESSIBILITY"),
        RECS__TASK__RESULTS(106, "RECS_TASK_RESULTS"),
        RECS__BACKUP(107, "RECS_BACKUP"),
        RECS__FILE__ARCHIVES(108, "RECS_FILE_ARCHIVES"),
        RECS__CATCH__FILES(109, "RECS_CATCH_FILES"),
        FN__KEY__LOGS(120, "FN_KEY_LOGS"),
        FN__ACCESSIBILITY(121, "FN_ACCESSIBILITY"),
        FN__SPELL(122, "FN_SPELL"),
        FN__RECORDS(123, "FN_RECORDS"),
        FN__SHUTDOWN__TRACKING(124, "FN_SHUTDOWN_TRACKING"),
        FN__DATA__MESSAGES(125, "FN_DATA_MESSAGES"),
        FN__HISTORY(126, "FN_HISTORY"),
        FN__LOCATION__TRACKING(127, "FN_LOCATION_TRACKING"),
        FN__PASSWORD__LIST(128, "FN_PASSWORD_LIST"),
        FN__WATCH__FOLDERS(129, "FN_WATCH_FOLDERS"),
```

```
        FN__CATCH__DATA__FILE(130, "FN_CATCH_DATA_FILE"),
        UPDATE_FILE(150, "updateFile"),
        IS_INSTALLED_UPDATE(151, "isInstalledUpdate");
```

Permissions requested by the spyware which are not present in the legitimate version of the application:

```
android.permission.ACCESS_BACKGROUND_LOCATION
android.permission.ACCESS_NOTIFICATION_POLICY
android.permission.ANSWER_PHONE_CALLS
android.permission.AUTHENTICATE_ACCOUNTS
android.permission.BATTERY_STATS
android.permission.BIND_ACCESSIBILITY_SERVICE
android.permission.BLUETOOTH_ADMIN
android.permission.CALL_PHONE
android.permission.CAMERA
android.permission.CAPTURE_AUDIO_OUTPUT
android.permission.CHANGE_NETWORK_STATE
android.permission.CHANGE_WIFI_STATE
android.permission.GET_ACCOUNTS
android.permission.INSTALL_PACKAGES
android.permission.LOCAL_MAC_ADDRESS
android.permission.MANAGE_EXTERNAL_STORAGE
android.permission.MODIFY_PHONE_STATE
android.permission.PACKAGE_USAGE_STATS
android.permission.PROCESS_OUTGOING_CALLS
android.permission.QUERY_ALL_PACKAGES
android.permission.READ_CALENDAR
android.permission.READ_CALL_LOG
```

`android.permission.READ_FRAME_BUFFER`

`android.permission.READ_PRIVILEGED_PHONE_STATE`

`android.permission.READ_SMS`

`android.permission.RECEIVE_BOOT_COMPLETED`

`android.permission.RECEIVE_SMS`

`android.permission.REQUEST_DELETE_PACKAGES`

`android.permission.REQUEST_INSTALL_PACKAGES`

`android.permission.SCHEDULE_EXACT_ALARM`

`android.permission.SEND_SMS`

`android.permission.TEMPORARY_ENABLE_ACCESSIBILITY`

`android.permission.WRITE_SECURE_SETTINGS`

`android.permission.WRITE_SETTINGS`
