ResidentBat: A new spyware family used by Belarusian KGB

Janik Besendorf, Viktor Schlüter, Maximilian Paß,
RESIDENT.NGO Team

Content

1 Key Findings

- This report introduces the previously unknown Spyware ResidentBat used by the Belarussian KGB (secret service) - It targets Android phones and is installed through physical device access
- Broad application permissions and an accessibility service allow the app access to a wide range of data, spanning phone calls, SMS, encrypted messenger chats and files on the phone.

2 Introduction

There seems to be a growing trend in surveillance of civil society where no remote attacks against the phone are used to install spyware, but rather physical access. Surveillance of journalists, lawyers and other members of civil society is on the rise, with the use of spyware such as NoviSpy, Massistant and Monokle. This kind of tactical spyware requires physical rather than remote access to install spyware on a target's phone.

What we document in this report follows this trend. In the third quarter of 2025(Q3 of 2025), RESIDENT.NGO identified a malware sample found on a journalist's phone shortly after the journalist had met with the Belarusian KGB (secret service). Following an initial analysis, RESIDENT.NGO escalated the case to the Digital Security Lab (DSL) at Reporters Without Borders. The results presented in this report stem from our joint research. This previously unknown spyware, which we detected, is used by the Belarusian KGB to track and surveil targets. As the malware contains the strings "bat" and "resident", we call this spyware ResidentBat.

Reporters Without Borders is grateful to Amnesty International's Security Lab for forensic and technical support during this investigation, and for peer-reviewing an earlier draft of this research.

3 Techical Analysis of ResidentBat

ResidentBat is bundled as a regular Android app in APK format. We identified two APKs, but have not published their names or their package names, in order to protect the identity of the targeted journalist. Henceforth, we will refer to the APKs as APK1 and APK2.

3.1 Capabilities

This spyware implements various techniques to collect data. Rather than using exploits, it remains within the boundaries that Android ordinarily permits for apps. In short, ResidentBat accesses the following types of data:

- All SMS
- Incoming calls
- Outgoing calls
- Files on the device
- Camera
- Android browser bookmarks (which was removed in Android 6)
- Clipboard which is only available until Android 10
- Internal microphone
- Various apps and messengers through screen monitoring capabilities of the included accessibility service

- Device administration capabilities

These are similar to the capabilities of stalkerware apps. The amount of accessible data is limited compared to spyware that uses exploits to break out of the application sandbox and gain root access. Nevertheless, it still leads to severe privacy implications. The advantage for spyware developers is that it is far cheaper than developing spyware with exploits.

3.2 Permissions in the Android Manifest

ResidentBat requests 38 permissions in the `AndroidManifest`. 13 of these are classified as runtime permissions. They grant access to a wide range of resources, including access to SMS, audio recording and phone call tracking. Additionally, the `AndroidManifest` declares components of the apps. The components of this spyware contain activities, receivers and services. ResidentBat's manifest includes receivers that listen for intents that contain sensitive user data:

```
android.intent.action.BOOT_COMPLETED
android.intent.action.QUICKBOOT_POWERON
com.htc.intent.action.QUICKBOOT_POWERON
android.intent.action.NEW_OUTGOING_CALL
android.provider.Telephony.SMS_RECEIVED
android.intent.action.DATA_SMS_RECEIVED
android.intent.action.PACKAGE_ADDED
android.intent.action.PACKAGE_REMOVED
android.app.action.DEVICE_ADMIN_ENABLED
android.app.action.ACTION_DEVICE_ADMIN_DISABLED
android.intent.action.USER_PRESENT
android.intent.action.TIMEZONE_CHANGED
android.intent.action.TIME_SET
android.intent.action.DATE_CHANGED
```

One service, called `ResidentService`, is declared as a foreground service and an accessibility service. As a foreground service, it can be active in the background while other apps are active (however it needs to show a notification while doing so). As an accessibility service, it can access screen content screen content. With the option `canRetrieveWindowContent` activated within its accessibility service, ResidentBat can iterate over the objects in the window of all apps and, for instance, collect the content of all objects containing text. The option `canTakeScreenshot` is also activated, allowing access to screenshots. Additionally, it uses the media projection APIs to capture a screen video stream.

```
<service
    android:name="com.google.bat.resident.ResidentService"
    android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE"
    android:enabled="true"
    android:exported="false"
    android:stopWithTask="false"
    android:foregroundServiceType="microphone|camera|mediaProjection|
        connectedDevice|location|phoneCall|mediaPlayback|dataSync">
    <meta-data
        android:name="android.accessibilityservice"
        android:resource="@xml/serviceconfig"/>
    <intent-filter>
        <action android:name="android.accessibilityservice.
            AccessibilityService"/>
    </intent-filter>
</service>
```

However, just declaring permissions in the AndroidManifest is not enough for the app to be able to access the data. The user also needs to grant the permissions to the app in the setting or when the app shows a pop-up. In spyware attacks like these this is usually done manually by the attackers after install. Additionally just because the permissions are defined in the AndroidManifest and granted by the user, that does not mean that the data is actually accessed. Each of the spyware samples comes with a configuration file which allows to granularly enable or disable specific data sources for surveillance. The configuration parameters are listed in the following table.

Short parameter	Verbose name	Short parameter	Verbose name
sars	serverAddress	asi	isNeedToTrackUserPhotos
spd	uploadPeriod	asi	isNeedToTrackUserPhotos
dvw	isOnlyWifiAllowedToUse	tff	isNeedToTrackAppsTraffic
asp	isNeedToUploadAsap	rua	isNeedToTrackInstalledApps
ric	isNeedToTrackIncomingCalls	xca	isNeedToTrackAccLog
roc	isNeedToTrackOutgoingCalls	wbaa	isAccLogBlacklistEnabled
rcs	isNeedToRecordCalls	swa	isNeedToTakeScreenshots
rcl	callRecordingChannel	tnf	isNeedToTrackNotifications
dpy	isDictaphoneHasPriorityOverCalls	was	accLogWatchedApps
afc	isNeedToRequestAudioFocus	sha	screenshotWatchedApp
soo2	stealthRecordingMode	wbsa	isScreenshotsBlacklistEnabled
tfc	isNeedToCheckMicAvailability	scp	screenshotPeriod
ris	isNeedToTrackIncomingSms	sts	screenshotPeriodFromInclusive
ros	isNeedToTrackOutgoingSms	ste	screenshotPeriodToExclusive
ugd	isNeedToTrackLocationPrecisely	nfa	notificationWatchedApps
rgl	isNeedToTrackLocation	wbna	isNotificationsBlacklistEnabled
lrp	locationTrackingPeriod	ipm	isProxyModeEnabled
ppb	isNeedToTrackContacts	pxn	proxyApp (String)
tms	isNeedToTrackMms	csn	isNeedToCloseSystemNotifications
rba	isNeedToTrackBrowserHistory	cas	isNeedToCheckAccServiceState

This includes various options to enable or disable data targets such as SMS, location, app traffic, browser history or contacts. Other parameters control collection frequencies, for example, how often screenshots are taken. The option `isOnlyWifiAllowedToUse` is likely intended to hide the spyware's traffic from mobile data usage statistics.

The device administration API, which is used by ResidentBat allows for setting device policies such as minimum lengths of passwords, and performing the following administrative tasks (as described in the Android documentation).

- Prompt user to set a new password.
- Lock device immediately.
- Wipe the device's data (that is, restore the device to its factory defaults).

It is noteworthy that the Device Admin feature was launched in 2010 as part of Android 2.2. It was replaced by the "Device Owner" feature in 2014 with Android 5. Android 10 (2019) was the first Android version to stop supporting "Device Admin".

This feature also lets the app run in the background without being suspended by the package manager. The following line was included in the Android logs from the infected device that we analyzed:

```
Cannot suspend package ""APK1"": has an active device admin
```

The Device Admin functionality is used for three app functions:

1. To report to the server if device admin status is active.
2. To use the `DevicePolicyManager.wipeData(int)` to wipe the device when the corresponding command is triggered by the C2 Server.
3. To remove itself from the device admin apps `removeActiveAdmin(str)`. This is also required before the app can be uninstalled.

Using dynamic analysis, we verified that the the spyware records phone calls, app content, and executes remotely invoked commands like reading files. Using a mocked C2 we also verified that the data is in fact sent to the C2 server.

4 ResidentBat Caught in The Wild

In Q3 of 2025, a journalist whose identity we verified but cannot make public for safety reasons, was interrogated by the Belarusian KGB. Prior to the interview, they were asked to put their phone in a locker, which was locked with a key. Later, the journalist was asked to show the contents of their phone to a KGB officer, so they went to the locker and retrieved it.

The journalist told us: "In the room [the] KGB officer asked all the time to unlock the screen, thus I think he just saw the password [Ed. PIN] I entered." We assume that after the KGB officer discovered the password [Ed. PIN], they took the decvice from the journalist and installed the spyware.

Later, the journalist was notified by their phone's internal anti-virus component, that a suspicious app had been installed. The anti-virus component recommended uninstalling the app, which the journalist did. Unsettled by these events, the journalist contacted RESIDENT.NGO to analyze the device who then escalated the case to the Digital Security Lab at Reporters without Borders.

After a deeper analysis of the phone, a version of the ResidentBat spyware (with APK1) was identified. Android log data shows it was installed by the KGB during the interrogation. As can be seen in the timeline below, the first usage times of the ResidentBat spyware were during the interrogation period.

As mentioned above two apps were installed by the KGB (with package names APK1 and APK2). The journalist uninstalled APK2, however, APK1 remained active and undiscovered until our forensic analysis.

To protect the identity of the journalist, we have set out the following timeline with relative timestamps. The starting time is undisclosed and labelled as t0 which lies within Q3 of 2025. All times included in this report only specify how long after t0 an event occurred. Relative timestamps are shown as t0 (the undisclosed starting time) plus + HH:MM:SS, specifying how many hours, minutes and seconds have passed since t0.

4.1 Relative Timeline of the Events

relative Timestamp after t0	Event
0 days 00:00:00	(approx.) The device was seized by KGB
0 days 00:54:17	APK1 app was visible on screen for first time
0 days 00:55:27	APK1 app was visible on screen for the last time
0 days 00:57:40	APK2 was visible for the last time
0 days 01:07:34	APK1's certificate starts being valid
0 days 03:00:00	(approx.) The device was returned by KGB, USB Debugging is disabled.
1 day 17:50:00	APK2 was uninstalled
2 day 07:17:36	traces of APK1 accessing contact list
2 day 21:28:31	indicator found that APK1 has been registered as device administration app

Interestingly, APK1's certificate was only valid after `t0 + 01:07:34` but the app was already installed before `t0 + 00:54:17`. To test whether Android usually allows the installation of apps whose certificates are only valid in the future, we tried to install an app on a test device where the certificate validity also only started in the future. This worked without issues.

We therefore assume that Android does not limit the installation of apps to the validity period of the application files. A likely explanation why the certificate of APK1/2 was only valid from a time in the future is that the system, which creates ResidentBat samples, has signed the app with a `ValidFrom` field that is a couple of minutes later than the APK creation time, possibly to shadow the exact time of the malware sample creation.

4.2 Traces left on device

Apart from the Android apps themselves, the installation process added a new adb key to the device, which specifies the user and host name. The adb key, user and hostname are also not mentioned in this report for safety reasons.

5 Other samples/incidents

By pivoting from the sample we found in the wild to other versions of ResidentBat on Virus Total, we were able to identify eight additional samples, with the following hashes:

```
02dc81ea172e45f0a6fd7241fffd1042f6925c52d2f91dee36085634207be4f1
07d39205f9ba159236477a02cdb3350fac4f158e0dbf26576bb50604339b1f42
0ed73428c7729806be57989f340a09a323af914f197cc0cbb5509316ca5baf7b
48e87bfcaa665bfbfcb027227384905878f090bbc19d02f74c41ade3cafb0950
77126e749a9c1144ae3cebb8deb0b72fc90d4eb73d1072a69a1248b4f518bb47
820c394b22b950335eb5cf21bc7df5c7a33081169f41440c74d67e7a8f196960
c3b92d05b105465881c0f68f5cf6c3edb24d2e5317ffd1256cb68c7921fe0721
fe05ba40f2d4b15db83524c169d030d097abc6713139ce6068969d97a24aa195
```

We identified these samples through a collection of shared attributes:

- the distinguished name of their APK certificate (`C:c, CN:cn, L:l, O:o, ST:st, OU:ou`)
- the same base name for the Receivers and Activities (`com.google.bat.*`)

The samples used the following package names (used to identify Android Apps within the system) and app names (used to identify themselves to the user interface).

Package Name	App Name	Hash	First seen from
com.oneplussync.bat	OnePlus Sync Services	77126e	United States
com.google.bat	Google System Service	48e87b	Switzerland
com.google.android.service	Google System Service	c3b92d	Belarus
com.linkedln.service	Linkedln	0ed734	Belarus
com.huaweisettingsapp.mkz	HUAWEI Settings	02dc81	Belarus
com.hihonor.core.service	HONOR Core	fe05ba	Belarus
com.android.framework.safety	System Framework	820c39	Belarus
cm.google.android.apps.assistant	Android Services	07d392	Belarus

Manually analyzing the samples confirmed that they are indeed the same spyware with some minor variations. These variations include:

- Package Names
- Config and login credentials
- Certificates used to sign the applications
- Certificates used to create a secure communication channel to the command and control (C2) servers
- Slightly different structures of the code, forming two subgroups of the samples

Table 1 in the appendix lists an overview over the identified malware samples and how some of their attributes varied. What they have in common is the receivers prefix `com.google.bat` and the name of the accessibility service `com.google.bat.resident.ResidentService`. From this we derived the name "ResidentBat".

The apps that are targeted to be monitored by grabbing their screen content also varied. The following table shows the spyware configuration in each sample which determines which apps are subject to monitoring and data collection. The full table can be found in the Appendix (Table 2).

Property	02dc81	c3b92d	fe05ba	0ed734	07d392	820c39
com.android.chrome	✅	✅	✅	✅	✅	✅
com.android.email	❌	❌	❌	❌	✅	❌
com.android.settings	✅	✅	✅	✅	❌	✅
com.azure.authenticator	❌	❌	✅	❌	❌	✅
com.hihonor.photos	❌	❌	✅	❌	❌	✅
com.huawei.photos	❌	❌	✅	❌	❌	✅
com.iMe.android	❌	❌	✅	❌	❌	✅
com.icq.mobile.client	❌	❌	✅	❌	❌	✅
com.instagram.android	✅	✅	✅	✅	✅	✅
com.microsoft.teams	✅	✅	✅	✅	❌	✅
com.samsung.android.app.notes	❌	❌	✅	❌	❌	✅
com.sec.android.app.sbrowser	✅	✅	✅	✅	❌	✅
com.skype.raider	✅	✅	✅	✅	✅	✅
com.tencent.mm	✅	✅	✅	✅	✅	✅
com.viber.voip	✅	✅	✅	✅	✅	✅
com.vk.im	✅	✅	✅	✅	❌	✅
com.vkontakte.android	✅	✅	✅	✅	✅	✅
com.whatsapp	✅	✅	✅	✅	✅	✅
com.yandex.browser	❌	❌	✅	❌	❌	✅
im.thebot.messenger	❌	❌	✅	❌	❌	✅
im.vector.app	✅	✅	✅	✅	❌	✅
org.mozilla.firefox	✅	✅	✅	✅	❌	✅
org.telegram.messenger	✅	✅	✅	✅	✅	✅
org.thoughtcrime.securesms	✅	✅	✅	✅	❌	✅
ru.yandex.disk	❌	❌	✅	❌	❌	✅
us.zoom.videomeetings	✅	✅	✅	✅	❌	✅

The list of surveillance targets includes various messengers such as Telegram, Viber, Skype, VKontakte and Signal. The high overlap in target apps between many of the samples makes it likely that these surveillance operations were conducted by the same spyware operator(s).

The following timeline gives a contextual overview of when the found ResidentBat samples were likely used. Each sample includes a certificate which is used to communicate with the C2 server, and a certificate which is used to sign the application file itself. By analyzing these periods, they show a sequence and strong evidence that this operation has been ongoing since at least April 2021. Because the APK certificate "Valid From" time was very close to the time of the infection, we assume that the "Valid From" -times of other samples also correspond to their installation times. With this we can see when these samples were likely generated and installed on the target devices:

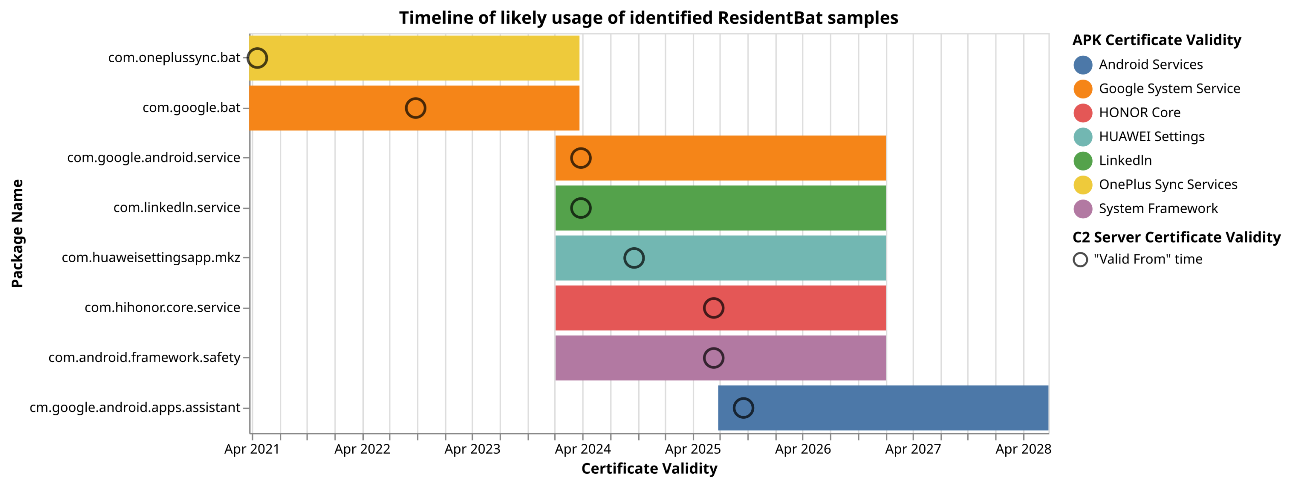


Figure 1: Timeline of likely usage of identified ResidentBat samples

Note that the C2 Server validity times are the times in which the certificate of the used C2 server was valid. As many of the certificates share their validity periods, it can be assumed that these C2 servers were set up in chunks at the same time.

6 C2 Communication and C2 Servers

ResidentBat specifies the command and control (C2) server within the "sars" configuration key of the configuration file.

Based on the C2 server in the sample we obtained, we developed a fingerprint FP1 that also identifies the other C2 servers:

```
services.tls.certificates.leaf_data.subject.common_name="server" and
services.banner_hashes="sha256:6
    f6676d369e99d61ce152e1e2b2eb6f5e26a4331f4008b5d6fe567edefdbeaca"
```

We identified 25 live hosts matching this fingerprint on censys.io, the IP addresses for these are listed in the IoCs section. We attribute these hosts with medium to high confidence to the ResidentBat spying operation, however, we don't have clear evidence for all of them. To verify the fingerprint, we extracted the list of C2 servers found in the configuration files of the ResidentBat samples identified through VirusTotal:

```
https://188.120.230[.]46:7003
https://45.155.7[.]166:7035
https://79.132.136[.]191:7007
https://79.132.136[.]191:7017
https://mtcat[.]info:7007
https://mtcat[.]info:7017
```

We identified the following overlaps from the found samples to the alleged ResidentBat hosts identified on Censys:

- The host at `79.132.136[.]191` matches the c2 endpoint from the sample 02dc81 fe05ba and 820c39
- `mtcat[.]info` resolves to `176.10.124.158`, which is also a host matched by the fingerprint FP1
- The host at `188.120.230[.]46` is also a host matched by the fingerprint; with this information we treated FP1 as verified.

7 Timeline of ResidentBat operation

We used the following criteria to identify ResidentBat activity:

- Eight samples from the VirusTotal matching the three criteria mentioned above (APK certificate, receiver prefix, configuration file structure)
- The fingerprint FP1 identifying C2 servers

The earliest clear evidence of ResidentBat activity is the C2 server certificates from samples 48e87b and 77126e. Their validity starts at March 2021, which is one month earlier than the validity of the APK discussed in the section before. Therefore, we can conclude that the ResidentBat operation by the KGB has been running since at least March 2021. To look even further back in time, we used the SecurityTrails.com API to identify domains that previously pointed to the identified C2 IP addresses. Some of these domains, for example msim[.]info displayed activity going as far back as 2016. However, the historical internet scanning data for that time is limited, which makes it hard to verify if servers closely connected to the ResidentBat operation were running under that domain at that time. At this time, we don't have concrete evidence of who developed the ResidentBat spyware. The two main scenarios are that it has been:

- Developed in-house by the KGB; or
- Purchased as a white label solution by the KGB or developed by a commercial third party for the KGB.

In one of the older samples, 77126e, we found what looks like English user interface strings for the operator panel, controlling the spyware. It contains lines such as:

```
    <string name="turn_net_roaming_desc">Turn data on in roaming. WiFi is
       used by default.</string>
    <string name="up_time">Upload frequency</string>
    <string name="upload_asap">Upload data ASAP</string>
    <string name="upload_asap_desc">Upload right after new data collected</
       string>
    <string name="upload_history">Upload internal browser history</string>
    <string name="upload_mms">Upload MMS</string>
```

and

```
    <string name="upload_runned_desc">Upload application running list at the
        sync time</string>
```

This could be seen as an indication that this product is not developed only for Belarusian users by Belarusian developers. However, who developed ResidentBat remains an open question.

8 How users can protect themselves

Users have two main ways of protecting themselves against such attacks:

- Preventing their device from being seized by authorities
- Preventing attackers from installing spyware when the phone is seized. This can be done by:
  - Keeping their PIN code from the authorities (however they might use forensic tools to extract your PIN)
  - Using a device that does not support side loading of apps.

The first point can be implemented by using secondary devices with less relevant data on them: These could be used in scenarios where device confiscation is more likely, such as crossing borders and attending police interrogations or protests. Of course, leaving the phone in a secure location during those situations can also prevent these types of spyware attacks.

The second point can be acted on by:

- Only entering your PIN when no person or camera is watching
- Using alphanumerical password rathan than a numerical PIN, as these are harder to shoulder surf
- Using an iPhone, which has limited support for installation of apps outside of the official app store, and fewer permissions like the accessibility service used by ResidentBat.
- Using secure operating systems like GrapheneOS, which provide features to protect against shoulder surfing:
  - A PIN-scrambling feature which changes the layout of the keyboard on the lockscreen every time. This makes it harder to observe the pin on camera or in person.
  - Support for 2-factor unlocking, which requires a fingerprint and a PIN to unlock the device. Surveillance by the KGB in this case would not have been possible, because the camera could only have filmed the PIN. Without the fingerprint they could not have unlocked the phone without the journalist. However, noticing this, the authorities could make the user unlock the device by force. This would endanger the user further, but at least the attack would have been t to the journalist.

Additionally, GrapheneOS has a duress password feature, where a user can create a different password to the one that unlocks the phone. Instead of unlocking the phone, the password deletes all data from it.

However, GrapheneOS is not the most accessible operating system as it currently only supports Google Pixel smartphones. If switching to GrapheneOS is too much of an obstacle for users, we advise them to regularly check for signs of infection described in the next section.

Generally speaking, we advise users to use smartphones that run a recent version of Android or iOS and still receive security updates from the manufacturer. Android security features have significantly improved across versions and will continue to do so. Using an outdated Android phone means that there are unfixed security vulnerabilities and that these new Android security features are missing.

We also highly encourage users who might be targeted by spyware to enable Android's Advanced Protection Mode (AAPM). Available since Android 16, it provides important security features for users who might be infected with spyware. For example, it blocks the installation of apps from unknown sources and prevents disabling Google Play Protect. To enable this mode, go to Settings -> Security & Privacy and under "Other settings," tap Advanced Protection. For more, see the documentation

9 How users can check if they are affected

Together with this report, we have released public indicators of compromise for use with the Mobile Verification Toolkit (MVT) a forensic analysis tool released by Amnesty International with further development by the forensic community including RSF's Digital Security Lab. These indicators allow MVT to detect all versions of the spyware we identified, as well as unknown ResidentBat versions in the scope of the already observed variations in app package names. If you detect that this app was active on your phone, we invite you to contact RESIDENT.NGO, the Digital Security Lab at Reporters without Borders or the Security Lab at Amnesty International. We all work in close collaboration and are very interested in finding out more about where the spyware was used and who was affected by its operation.

Because the spyware can remotely wipe the phone, we suggest that any users who wish to find out whether they are affected, first put their phone into Airplane mode and only then use tools like MVT, or have their device checked by civil society forensic experts.

The spyware displayed a custom update notification when recording calls and Android displayed a media projection status bar icon when the spyware was screenshotting other apps. If users see a notification from an app that they do not know or a media projection indicator despite not sharing their screen, that could indicate a spyware infection.

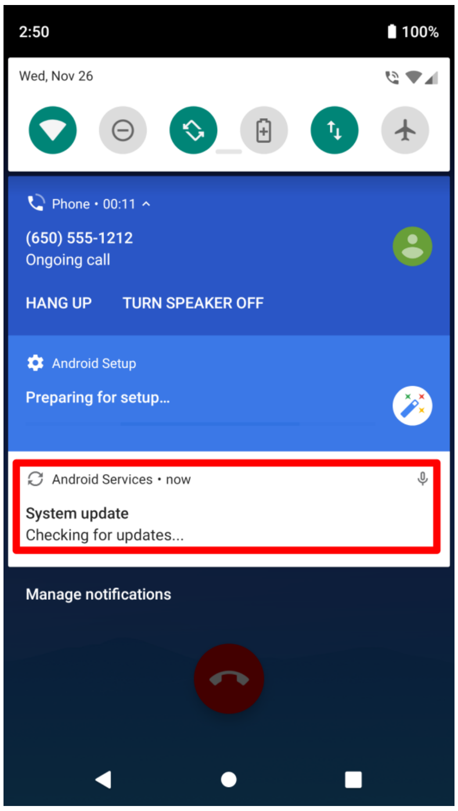


Figure 2: The decoy message shown while recording a phone call

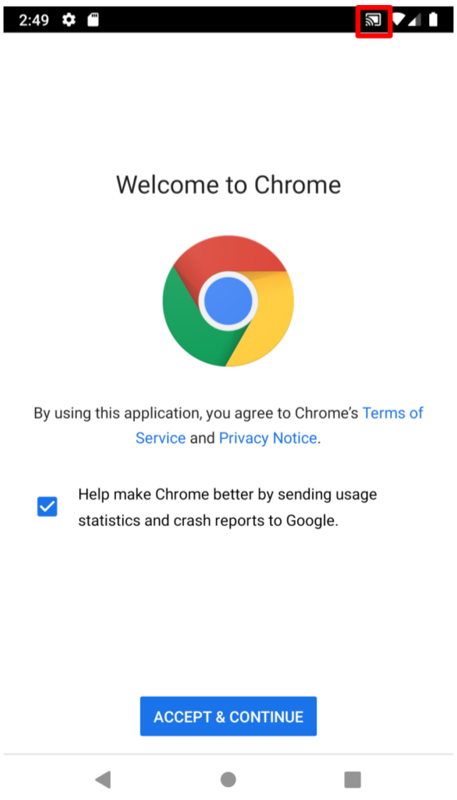

2:49
Welcome to Chrome
By using this application, you agree to Chrome's Terms of Service and Privacy Notice.
Help make Chrome better by sending usage statistics and crash reports to Google.
ACCEPT & CONTINUE

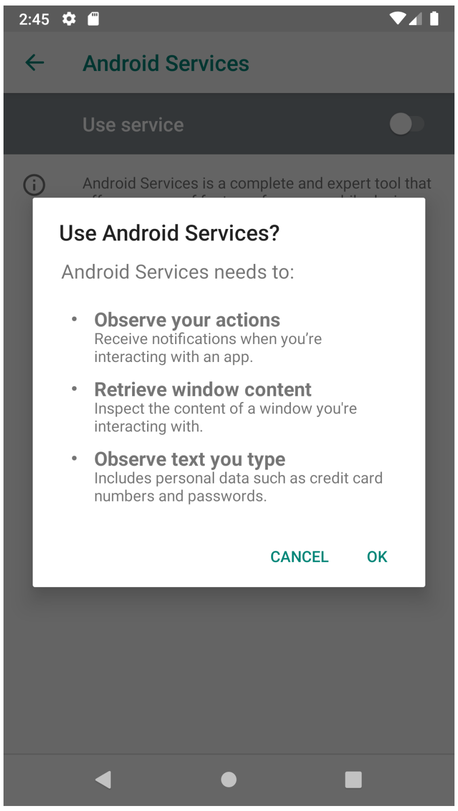
Figure 3: The media projection icon being displayed

Before the spyware was installed on the journalist's phone the attackers disabled Google Play Protect. Potential targets of spyware should always keep Google Play Protect enabled. If it is disabled and you did not do so yourself, this could indicate that someone installed malware on your device.

You can check if it is enabled if you go to Play Store -> Profile icon in the top right -> Play Protect. Users can also search for unknown Device Admin apps and Accessibility Services. To look for unknown device admin apps go to Settings -> Security & Privacy -> More Security & Privacy -> Device Admin and see if any unknown device admin apps are installed and/or enabled. If in doubt, disable and uninstall them.

To look for unknown Accessibility Services go to Settings -> Accessibility and look under the heading Downloaded apps and see if any unknown Accessibility Services are installed and/or enabled. If in doubt, disable and uninstall them. Users can also check for unknown apps in the Android settings. For this, go to Settings -> Apps ->See all apps. Note that malicious apps can give themselves legitimate names and app icons; if in doubt contact one of the above-mentioned civil society forensics teams.

Please note that the location in settings might vary based on Android version and smartphone manufacturer


Use Android Services?

Android Services needs to:

- Observe your actions
  Receive notifications when you're interacting with an app.
- Retrieve window content
  Inspect the content of a window you're interacting with.
- Observe text you type
  Includes personal data such as credit card numbers and passwords.

Figure 4: Screenshot of the Accessibility Service activation screen from sample 07d392

10 Possible improvements for enhancing Android security

Protecting users from attackers with physical access and knowledge of passwords is difficult. However, we would like to make a few suggestions that aim to mitigate similar attacks.

While (AAPM) prevents the installation of apps from untrusted sources, it is possible to turn AAPM off using the same authentication method (PIN or password) that unlocks the phone. In the case described here, it would therefore not necessarily have been able to protect the journalist as the PIN was known to the attacker. Additionally, if AAPM is turned off, this is not immediately obvious to the user unless they actively check for it. Our suggestion is to require two-factor authentication (PIN/password + biometrics) to turn off AAPM. Furthermore, it would be beneficial to remind the user a few days after AAPM has been disabled. This reminder could also be sent via email. The same approach should be applied when Google Play Protect is disabled.

As the Accessibility Service was crucial for the most privacy-invasive features of ResidentBat, it would be helpful to regularly remind the user when an Accessibility service is enabled.

Google announced that in a future release of Android it will restrict the installation of apps that have not been signed by a developer who has verified their identity with Google. After backlash from Android users, Google announced that it will introduce a flow for advanced users to enable the installation of apps that have not been signed by a developer who has verified their identity with Google. We suggest regularly reminding users when this feature is enabled.

We previously mentioned that GrapheneOS supports two-factor authentication for unlocking. We suggest that Google include such a feature in AOSP (Android Open Source Project) as well.

11 Acknowledgements

We thank the affected journalist for trusting us with their data. We also thank Censys.io and VirusTotal for granting us access to their services, which helped with significant parts of this investigation.

Reporters Without Borders' Digital Security Lab maintains a forensic partnership with Amnesty International's Security Lab. Tools and services shared by the Security Lab, including the Mobile Verification Toolkit (MVT) have supported RSF/DSL to carry out this research and play an important role in civil society forensic investigations against advanced digital threats. RSF/DSL has significantly contributed to those tools and encourage others to use and contribute to them as well.

12 IoCs

12.1 IP addresses

```
62.109.26.144
91.107.122.180
5.129.230.104
82.146.35.54
62.109.12.75
79.132.136.191
83.220.169.120
5.129.213.114
5.253.63.176
62.109.11.98
62.109.19.123
185.248.103.85
5.129.231.158
185.18.54.246
91.240.87.211
185.248.103.128
185.248.103.247
188.120.230.46
37.46.133.87
5.253.61.156
79.132.141.31
37.46.128.62
91.228.152.4
91.192.102.69
```

12.2 Package Names

```
com.google.android.service
com.google.bat
com.huaweisettingsapp.mkz
com.linkedln.service
com.oneplussync.bat
cm.google.android.apps.assistant
com.android.framework.safety
com.hihonor.core.service
```

12.3 Receiver, Intent and Accessibility Service Prefix

```
com.google.bat.*
```

12.4 APK Hashes

```
02dc81ea172e45f0a6fd7241fffd1042f6925c52d2f91dee36085634207be4f1
07d39205f9ba159236477a02cdb3350fac4f158e0dbf26576bb50604339b1f42
0ed73428c7729806be57989f340a09a323af914f197cc0cbb5509316ca5baf7b
48e87bfcaa665bfbfcb027227384905878f090bbc19d02f74c41ade3cafb0950
77126e749a9c1144ae3cebb8deb0b72fc90d4eb73d1072a69a1248b4f518bb47
820c394b22b950335eb5cf21bc7df5c7a33081169f41440c74d67e7a8f196960
c3b92d05b105465881c0f68f5cf6c3edb24d2e5317ffd1256cb68c7921fe0721
```

```
fe05ba40f2d4b15db83524c169d030d097abc6713139ce6068969d97a24aa195
```

12.5 APK Certficates

```
18afc5c6bfaee504a26291f6bf3e6f823dbedd54bba0c4acac2e7c2414b3e24d
c1884e617348ebbdfe7cfe5fc99945b37296d6ebc6059bb74fbaeea277d32941
e5016f3cfb937d502dabedc32ca3bdef3bbcce032fb3b1bff3b9c6482895f4fd
d12616542268d32329f1c4357b5d5a57e954e13d2338d27bb8439794291b8c6d
3e9f1192e33cb851b48479629c93d29770a4f76af00f1e42a3c6e7f97db62c79
6782039a81a85264acdc6af0973b225ada6009f76faae7f948a1de040bb32f0c
a6a067b0d899fb514b7b4597d4fe16fcd4d7e5c361f6c84b3d45ed7e394036c7
6d6278ffc80ad9dd1b1c6b445847ce108f3ea5ce349f232689e9b8c1fd10801e
```

13 Appendix

13.1 Table 1: Overview of metadata of identified samples

hash	Package Name	App Name	Receivers prefix	Accessibility service	C2 Certificate valid from	C2 Certificate valid until	APK Certificate valid from	APK Certificate valid until
77126e	com.oneplussync.bat	OnePlus Sync Services	bat[1]	res[2]	2021-03-23 14:18:47+03:00	2024-03-22 14:18:47+03:00	2021-04-19 13:16:08+03:00	2048-09-04 14:16:08+03:00
48e87b	com.google.bat	Google System Service	bat[1]	res[2]	2021-03-23 14:18:47+03:00	2024-03-22 14:18:47+03:00	2022-09-26 13:03:09+03:00	2050-02-11 13:03:09+03:00
c3b92d	com.google. android.service	Google System Service	bat[1]	res[2]	2024-01-03 13:50:17+03:00	2027-01-02 13:50:17+03:00	2024-03-27 12:27:14+03:00	2051-08-13 13:27:14+03:00
0ed734	com.linkedln.service	Linkedln	bat[1]	res[2]	2024-01-03 13:50:17+03:00	2027-01-02 13:50:17+03:00	2024-03-27 12:44:20+03:00	2051-08-13 13:44:20+03:00
02dc81	com.huawei settingsapp.mkz	HUAWEI Settings	bat[1]	res[2]	2024-01-03 13:57:00+03:00	2027-01-02 13:57:00+03:00	2024-09-19 20:13:27+03:00	2052-02-05 20:13:27+03:00
fe05ba	com.hihonor. core.service	HONOR Core	bat[1]	res[2]	2024-01-03 13:57:00+03:00	2027-01-02 13:57:00+03:00	2025-06-10 18:13:52+03:00	2052-10-26 19:13:52+03:00
820c39	com.android. framework.safety	System Framework	bat[1]	res[2]	2024-01-03 13:57:00+03:00	2027-01-02 13:57:00+03:00	2025-06-10 18:20:53+03:00	2052-10-26 19:20:53+03:00
07d392	cm.google.android. apps.assistant	Android Services	bat[1]	res[2]	2025-06-25 09:52:29+03:00	2028-06-24 09:52:29+03:00	2025-09-17 13:09:06+03:00	2053-02-02 13:09:06+03:00

[1] (com.google.bat.*)

[2] (com.google.bat.resident.ResidentService)

13.2 Table 2: Full Overview of ResidentBat configurations

Property	02dc81	c3b92d	fe05ba	0ed734	07d392	820c39
app.cryptocourse.wallet	❌	❌	✅	❌	❌	✅
app.nicegram	❌	❌	✅	❌	❌	✅
by.belbet.android	❌	❌	✅	❌	❌	✅
by.fonbet	❌	❌	✅	❌	❌	✅
ch.protonmail.android	✅	✅	✅	✅	❌	✅
cn.wps.moffice_i18n	❌	❌	✅	❌	❌	✅
com.android.chrome	✅	✅	✅	✅	✅	✅
com.android.email	❌	❌	❌	❌	✅	❌
com.android.settings	✅	✅	✅	✅	❌	✅
com.azure.authenticator	❌	❌	✅	❌	❌	✅
com.betera.beterizaciamobileapp	❌	❌	✅	❌	❌	✅
com.betera.sport	❌	❌	✅	❌	❌	✅
com.codespaceapps.aichat	❌	❌	✅	❌	❌	✅
com.currency.exchange.prod2	❌	❌	✅	❌	❌	✅
com.dddev.gallery.album.photo.editor	❌	❌	✅	❌	❌	✅
com.discord	✅	✅	✅	✅	❌	✅
com.facebook.appmanager	✅	✅	✅	✅	✅	✅
com.facebook.katana	✅	✅	✅	✅	✅	✅
com.facebook.orca	✅	✅	✅	✅	❌	✅
com.google.android.apps.docs	❌	❌	✅	❌	❌	✅
com.google.android.apps.photos	❌	❌	✅	❌	❌	✅
com.google.android.apps.tachyon	❌	❌	✅	❌	❌	✅
com.google.android.gm	✅	✅	✅	✅	❌	✅
com.google.android.gms	✅	✅	✅	✅	❌	✅
com.google.android.talk	❌	❌	✅	❌	❌	✅
com.hihonor.notepad	❌	❌	✅	❌	❌	✅
com.hihonor.photos	❌	❌	✅	❌	❌	✅
com.huawei.notepad	❌	❌	✅	❌	❌	✅
com.huawei.ohos.photos	❌	❌	✅	❌	❌	✅
com.huawei.photos	❌	❌	✅	❌	❌	✅
com.iMe.android	❌	❌	✅	❌	❌	✅

Property	02dc81	c3b92d	fe05ba	0ed734	07d392	820c39
com.icq.mobile.client	❌	❌	✅	❌	❌	✅
com.imo.android.imoim	✅	✅	✅	✅	❌	✅
com.imo.android.imoimhd	✅	✅	✅	✅	❌	✅
com.instagram.android	✅	✅	✅	✅	✅	✅
com.instagram.lite	❌	❌	✅	❌	❌	✅
com.loudtalks	✅	✅	✅	✅	❌	✅
com.mi.globalbrowser	✅	✅	✅	✅	❌	✅
com.microsoft.office.outlook	❌	❌	✅	❌	❌	✅
com.microsoft.teams	✅	✅	✅	✅	❌	✅
com.miui.gallery	❌	❌	✅	❌	❌	✅
com.miui.notes	❌	❌	✅	❌	❌	✅
com.opera.browser	✅	✅	✅	✅	✅	✅
com.payeer	❌	❌	✅	❌	❌	✅
com.perm.kate_new_6	✅	✅	✅	✅	❌	✅
com.radolyn.ayugram	❌	❌	✅	❌	❌	✅
com.samsung.android.app.notes	❌	❌	✅	❌	❌	✅
com.sec.android.app.sbrowser	✅	✅	✅	✅	❌	✅
com.sec.android.gallery3d	❌	❌	✅	❌	❌	✅
com.skype.raider	✅	✅	✅	✅	✅	✅
com.tencent.mm	✅	✅	✅	✅	✅	✅
com.viber.voip	✅	✅	✅	✅	✅	✅
com.vk.im	✅	✅	✅	✅	❌	✅
com.vkontakte.android	✅	✅	✅	✅	✅	✅
com.whatsapp	✅	✅	✅	✅	✅	✅
com.whatsapp.w4b	❌	❌	✅	❌	❌	✅
com.yandex.browser	❌	❌	✅	❌	❌	✅
im.thebot.messenger	❌	❌	✅	❌	❌	✅
im.vector.app	✅	✅	✅	✅	❌	✅
org.linkmessenger.me	❌	❌	✅	❌	❌	✅
org.mozilla.firefox	✅	✅	✅	✅	❌	✅
org.telegram.BifToGram	✅	✅	✅	✅	❌	✅
org.telegram.messenger	✅	✅	✅	✅	✅	✅
org.telegram.messenger.web	✅	✅	✅	✅	❌	✅

Property	02dc81	c3b92d	fe05ba	0ed734	07d392	820c39
org.thoughtcrime.securesms	✅	✅	✅	✅	❌	✅
org.thunderdog.challegram	✅	✅	✅	✅	✅	✅
ru.mw	❌	❌	✅	❌	❌	✅
ru.yandex.disk	❌	❌	✅	❌	❌	✅
ru.yandex.searchplugin	❌	❌	✅	❌	❌	✅
ru.yandex.yandexmaps	❌	❌	✅	❌	❌	✅
ru.yandex.yandexnavi	❌	❌	✅	❌	❌	✅
us.zoom.videomeetings	✅	✅	✅	✅	❌	✅