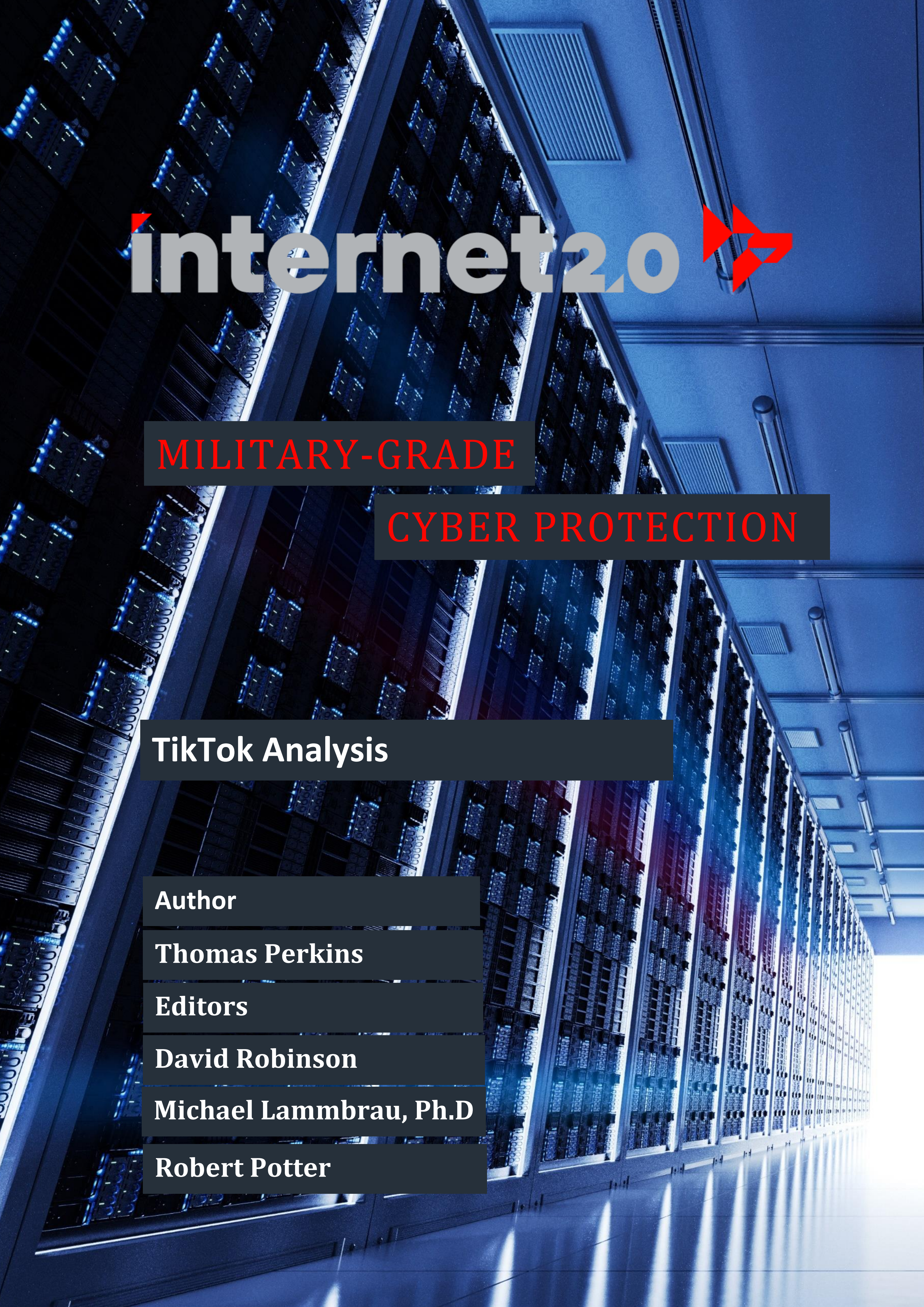


internet2.0

MILITARY-GRADE

CYBER PROTECTION

TikTok Analysis

Author

Thomas Perkins

Editors

David Robinson

Michael Lammbrau, Ph.D

Robert Potter

Table of Contents

Executive Summary

This report is a technical analysis of the source code of TikTok mobile applications Android 25.1.3 as well as IOS 25.1.1. Analysis of the Android application was performed using a Galaxy S9 cell phone. Internet 2.0 conducted static and dynamic analysis of the source code between 01-12 July 2022. This report aims to analyse TikTok device and user (customer) data collection. Prepared by Internet 2.0, this report is for policy makers and legislators to make evidence-based decisions. TikTok is a dominant social media application and is the 6th most used application globally with forecasted advertising revenues in 2022 expected to be USD12 billion. In our analysis the TikTok mobile application does not prioritise privacy. Permissions and device information collection are overly intrusive and not necessary for the application to function. The following are examples of excessive data harvesting.

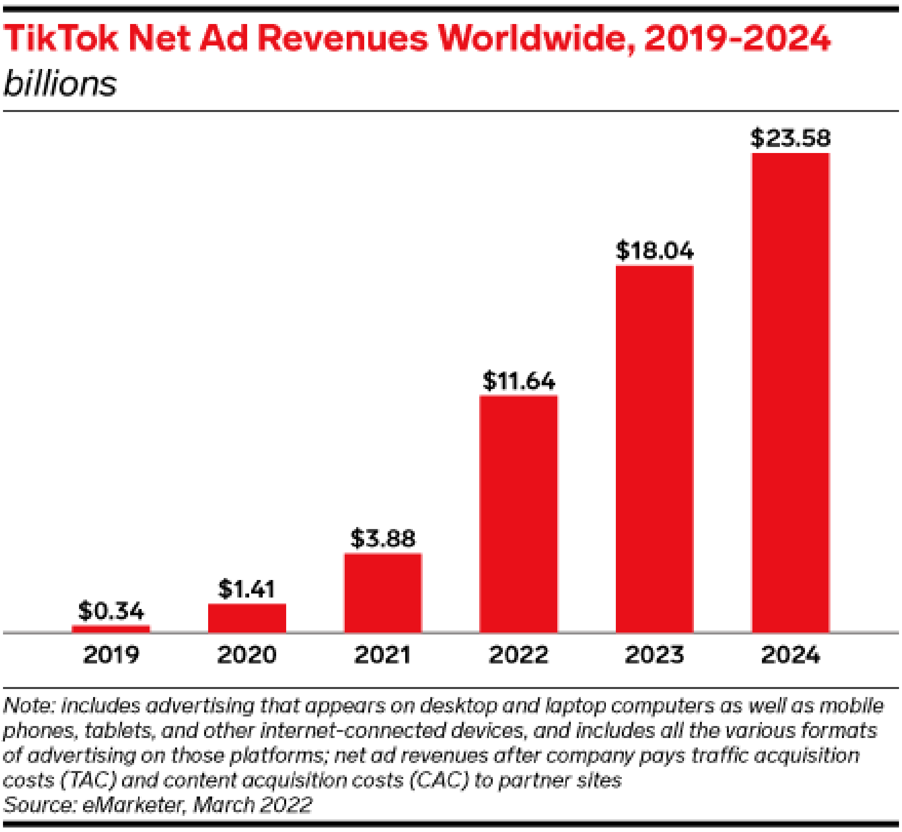
- **Device Mapping.** The application retrieves all other running applications on the phone. TikTok also gathers all applications that are installed on the phone. In theory this information can provide a realistic diagram of your phone.
- **Location.** TikTok checks the device location at least once per hour.
- **Calendar.** TikTok has persistent access to the calendar.
- **Contacts.** TikTok has access to contacts and if the user denies access, it continuously requests for access until the user gives access.
- **Device information**. TikTok has code that collects the following device detailed information on Android.
    - Wi-Fi SSID
    - Device build serial number
    - SIM serial number
    - Integrated Circuit Card Identification Number (this is global unique serial number that is specifically tailored to your SIM card)
    - Device IMEI
    - Device MAC address
    - Device line number

- Device voicemail number
- GPS status information (updates on the GPS location)
- Active subscription information
- All accounts on the device
- Complete access to read the clipboard (dangerous as password managers use clipboards)

Also of note is that TikTok IOS 25.1.1 has a server connection to mainland China which is run by a top 100 Chinese cyber security and data company Guizhou Baishan Cloud Technology Co., Ltd.

Introduction

TikTok is currently one of the dominant social media application in the market. It is the 6th most used application. As at September 2021 TikTok has over 1 billion active users globally with 142.2 million users in North America.[1] It has been downloaded over 3.5 billion times as of January 2021, with 43.7% of users 18-24 years old and 31.9% 25 to 34 years old. TikTok's projected advertising annual revenue in 2022 will hit USD12 billion, up from USD1.41 billion in 2020.[2]


TikTok Net Ad Revenues Worldwide, 2019-2024
billions

Note: includes advertising that appears on desktop and laptop computers as well as mobile phones, tablets, and other internet-connected devices, and includes all the various formats of advertising on those platforms; net ad revenues after company pays traffic acquisition costs (TAC) and content acquisition costs (CAC) to partner sites
Source: eMarketer, March 2022

Figure 1. Projected TikTok advertising revenue (see footnote 2)

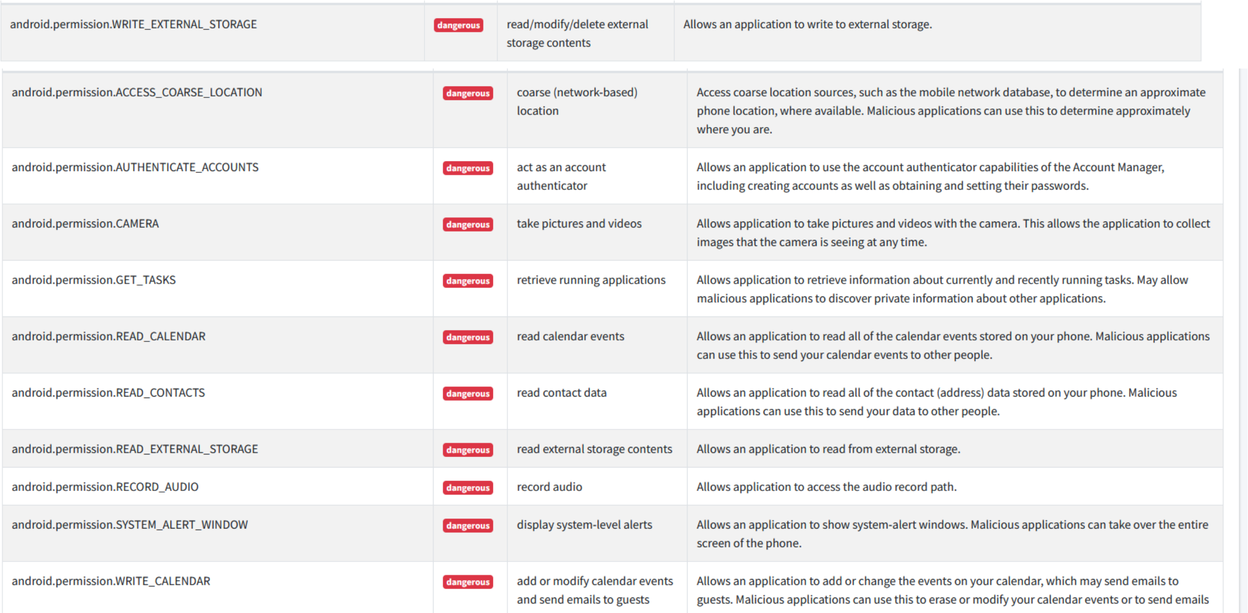
[1] https://www.shopify.com/blog/tiktok-statistics

[2] https://www.insiderintelligence.com/content/tiktok-douyin-digital-ad-spend

Internet 2.0 conducted static and dynamic analysis of the TikTok mobile application Android 25.1.3 as well as static analysis of the TikTok mobile application IOS 25.1.1 to understand user and device data collection.[3] The analysis also seeks to confirm the existence of any malicious code or features of the application. We decompiled the source code of the application available on the app stores and analysed it through multiple systems (including multiple sandbox services) and manual source code reviews. This is divided into the following sections: user permissions and third-party data access; device and user data harvesting; and conclusion.

User Permissions and Third-Party Data Access

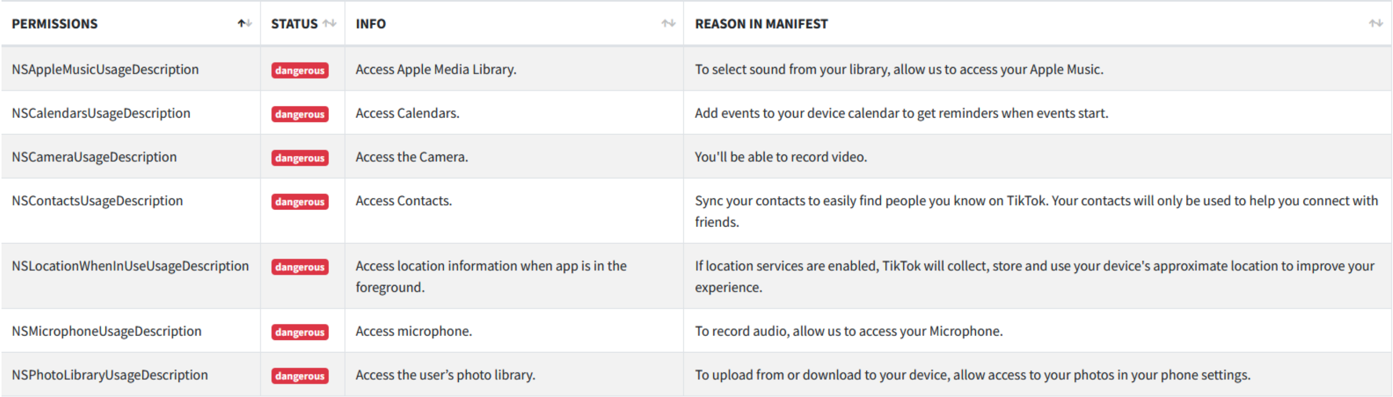
There are certain permissions that the Android documentation considers to be "dangerous". They are considered dangerous due to the permission providing additional access to restricted data. For example, the ability to read all SMS messages could be considered dangerous because an application could send all your texts to a server and save the information for future use (such as a malware). Unfortunately, TikTok makes use of a lot of these dangerous permissions. We noted the Android version had many more than the IOS version. IOS has a justification system where to gain a permission the developer must justify why this permission is required before it is granted. We believe the justification system IOS implements systematically limits a culture of "grab what you can" in data harvesting. The fact that TikTok had far more permissions for Android over IOS is a good demonstration of their culture when it comes to privacy.



android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.AUTHENTICATE_ACCOUNTS	dangerous	act as an account authenticator	Allows an application to use the account authenticator capabilities of the Account Manager, including creating accounts as well as obtaining and setting their passwords.
android.permission.CAMERA	dangerous	take pictures and videos	Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.
android.permission.GET_TASKS	dangerous	retrieve running applications	Allows application to retrieve information about currently and recently running tasks. May allow malicious applications to discover private information about other applications.
android.permission.READ_CALENDAR	dangerous	read calendar events	Allows an application to read all of the calendar events stored on your phone. Malicious applications can use this to send your calendar events to other people.
android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people.
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows an application to read from external storage.
android.permission.RECORD_AUDIO	dangerous	record audio	Allows application to access the audio record path.
android.permission.SYSTEM_ALERT_WINDOW	dangerous	display system-level alerts	Allows an application to show system-alert windows. Malicious applications can take over the entire screen of the phone.
android.permission.WRITE_CALENDAR	dangerous	add or modify calendar events and send emails to guests	Allows an application to add or change the events on your calendar, which may send emails to guests. Malicious applications can use this to erase or modify your calendar events or to send emails to guests.

Figure 2a. TikTok Android access permissions rated as dangerous.

[3] This analysis provides impartial advice for users to evaluate the extent to which their data is collected for privacy reasons. It allows policy advisors and legislators to make evidence-based decisions when discussing privacy concerns with vendors. This report was written for a global audience and does not include any legal or jurisdiction based regional assessments.

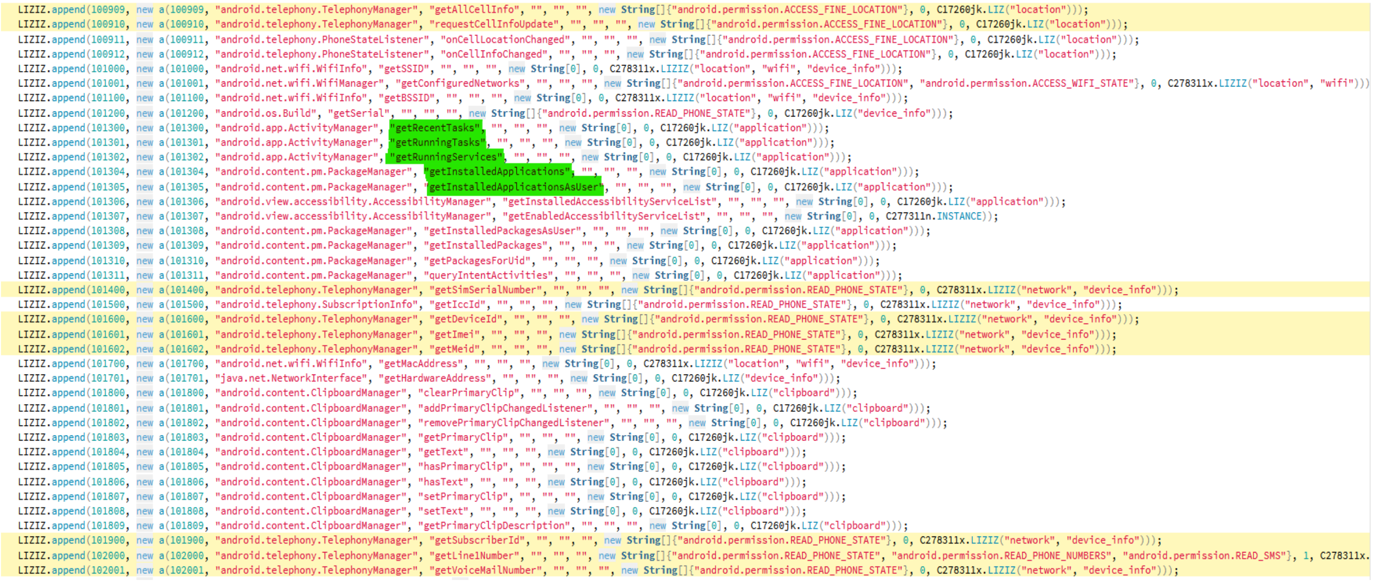


PERMISSIONS	STATUS	INFO	REASON IN MANIFEST
NSAppleMusicUsageDescription	dangerous	Access Apple Media Library.	To select sound from your library, allow us to access your Apple Music.
NSCalendarsUsageDescription	dangerous	Access Calendars.	Add events to your device calendar to get reminders when events start.
NSCameraUsageDescription	dangerous	Access the Camera.	You'll be able to record video.
NSContactsUsageDescription	dangerous	Access Contacts.	Sync your contacts to easily find people you know on TikTok. Your contacts will only be used to help you connect with friends.
NSLocationWhenInUseUsageDescription	dangerous	Access location information when app is in the foreground.	If location services are enabled, TikTok will collect, store and use your device's approximate location to improve your experience.
NSMicrophoneUsageDescription	dangerous	Access microphone.	To record audio, allow us to access your Microphone.
NSPhotoLibraryUsageDescription	dangerous	Access the user's photo library.	To upload from or download to your device, allow access to your photos in your phone settings.

Figure 2b. TikTok IOS access permissions rated as dangerous.

Device mapping

The Android application collects all other running and installed applications on the phone (this is an unnecessary function), see figure 3. Theoretically, this information can provide a realistic diagram of your phone.

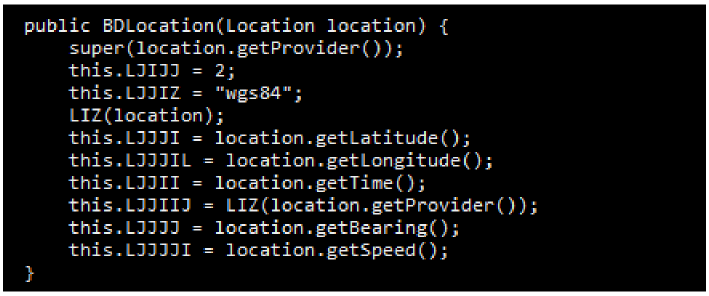


```
LIZIZ.append(100909, new a(100909, "android.telephony.TelephonyManager", "getAllCellInfo", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100910, new a(100910, "android.telephony.TelephonyManager", "requestCellInfoUpdate", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100911, new a(100911, "android.telephony.PhoneStateListener", "onCellLocationChanged", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100912, new a(100912, "android.telephony.PhoneStateListener", "onCellInfoChanged", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(101000, new a(101000, "android.net.wifi.WifiInfo", "getSSID", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101001, new a(101001, "android.net.wifi.WifiManager", "getConfiguredNetworks", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_WIFI_STATE"}, 0, C278311x.LIZIZ("location", "wifi")))
LIZIZ.append(101100, new a(101100, "android.net.wifi.WifiInfo", "getBSSID", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101200, new a(101200, "android.os.Build", "getSerial", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("device_info")));
LIZIZ.append(101300, new a(101300, "android.app.ActivityManager", "getRecentTasks", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101301, new a(101301, "android.app.ActivityManager", "getRunningTasks", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101302, new a(101302, "android.app.ActivityManager", "getRunningServices", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101304, new a(101304, "android.content.pm.PackageManager", "getInstalledApplications", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101305, new a(101305, "android.content.pm.PackageManager", "getInstalledApplicationsAsUser", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101306, new a(101306, "android.view.accessibility.AccessibilityManager", "getInstalledAccessibilityServiceList", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101307, new a(101307, "android.view.accessibility.AccessibilityManager", "getEnabledAccessibilityServiceList", "", "", "", new String[0], 0, C277311n.INSTANCE));
LIZIZ.append(101308, new a(101308, "android.content.pm.PackageManager", "getInstalledPackagesAsUser", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101309, new a(101309, "android.content.pm.PackageManager", "getInstalledPackages", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101310, new a(101310, "android.content.pm.PackageManager", "getPackagesForUid", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101311, new a(101311, "android.content.pm.PackageManager", "queryIntentActivities", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101400, new a(101400, "android.telephony.TelephonyManager", "getSimSerialNumber", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101500, new a(101500, "android.telephony.SubscriptionInfo", "getIccId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101600, new a(101600, "android.telephony.TelephonyManager", "getDeviceId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101601, new a(101601, "android.telephony.TelephonyManager", "getImei", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101602, new a(101602, "android.telephony.TelephonyManager", "getMeid", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101700, new a(101700, "android.net.wifi.WifiInfo", "getMacAddress", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101701, new a(101701, "java.net.NetworkInterface", "getHardwareAddress", "", "", "", new String[0], 0, C17260jk.LIZ("device_info")));
LIZIZ.append(101800, new a(101800, "android.content.ClipboardManager", "clearPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101801, new a(101801, "android.content.ClipboardManager", "addPrimaryClipChangedListener", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101802, new a(101802, "android.content.ClipboardManager", "removePrimaryClipChangedListener", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101803, new a(101803, "android.content.ClipboardManager", "getPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101804, new a(101804, "android.content.ClipboardManager", "getText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101805, new a(101805, "android.content.ClipboardManager", "hasPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101806, new a(101806, "android.content.ClipboardManager", "hasText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101807, new a(101807, "android.content.ClipboardManager", "setPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101808, new a(101808, "android.content.ClipboardManager", "setText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101809, new a(101809, "android.content.ClipboardManager", "getPrimaryClipDescription", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101900, new a(101900, "android.telephony.TelephonyManager", "getSubscriberId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(102000, new a(102000, "android.telephony.TelephonyManager", "getLine1Number", "", "", "", new String[]{"android.permission.READ_PHONE_STATE", "android.permission.READ_PHONE_NUMBERS", "android.permission.READ_SMS"}, 1, C278311x.
LIZIZ.append(102001, new a(102001, "android.telephony.TelephonyManager", "getVoiceMailNumber", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
```

Figure 3: Get all applications and running tasks on the device (green highlight).

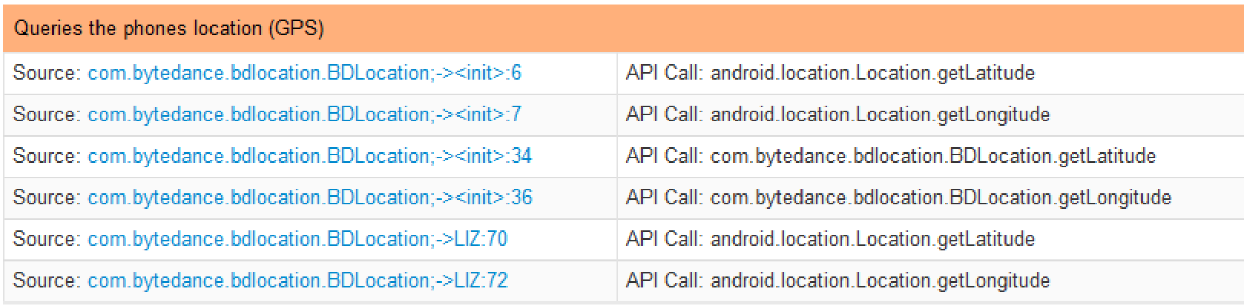
GPS and Locations requests

The Android application queries the device GPS location at least once per hour while running. This command is seen in figures 4 and 5.



```
public BDLocation(Location location) {
    super(location.getProvider());
    this.LJIJJ = 2;
    this.LJJIZ = "wgs84";
    LIZ(location);
    this.LJJJI = location.getLatitude();
    this.LJJJIL = location.getLongitude();
    this.LJJII = location.getTime();
    this.LJJIIJ = LIZ(location.getProvider());
    this.LJJJJ = location.getBearing();
    this.LJJJJI = location.getSpeed();
}
```

Figure 4: Get location code.

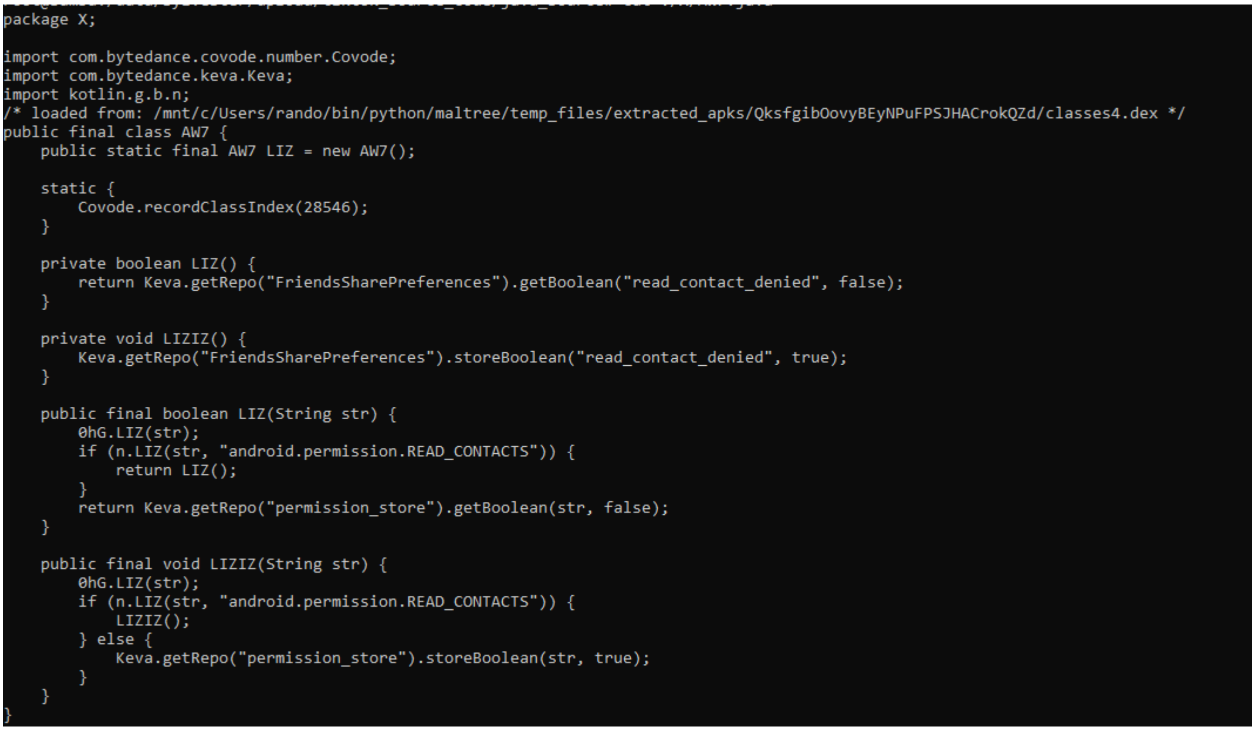


Queries the phones location (GPS)	
Source: com.bytedance.bdlocation.BDLocation;-><init>:6	API Call: android.location.Location.getLatitude
Source: com.bytedance.bdlocation.BDLocation;-><init>:7	API Call: android.location.Location.getLongitude
Source: com.bytedance.bdlocation.BDLocation;-><init>:34	API Call: com.bytedance.bdlocation.BDLocation.getLatitude
Source: com.bytedance.bdlocation.BDLocation;-><init>:36	API Call: com.bytedance.bdlocation.BDLocation.getLongitude
Source: com.bytedance.bdlocation.BDLocation;->LIZ:70	API Call: android.location.Location.getLatitude
Source: com.bytedance.bdlocation.BDLocation;->LIZ:72	API Call: android.location.Location.getLongitude

Figure 5: TikTok get longitude and latitude data requests.

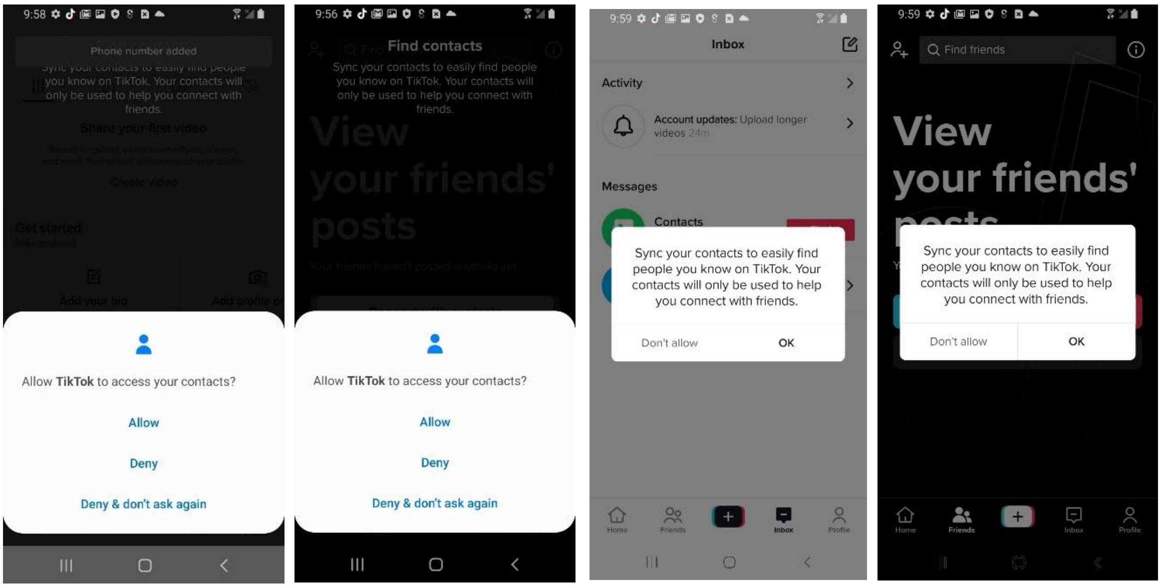
Contacts

The Android application requests access to user contacts. If the user denies access the application will continuously ask for access. TikTok does this as it runs its code in a loop that if a Boolean (true or false) is stored as false, it will keep prompting until given a true value (see figure 6). It is normal for an application to initially request access to contacts but TikTok's persistent, endless harassment for user contacts access is abnormal. It reflects a culture that does not prioritize privacy or a user's preferences for privacy.



```
package X;

import com.bytedance.covode.number.Covode;
import com.bytedance.keva.Keva;
import kotlin.g.b.n;
/* loaded from: /mnt/c/Users/rando/bin/python/maltree/temp_files/extracted_apks/QksfgibOovyBEyNPuFPSJHACrokQZd/classes4.dex */
public final class AW7 {
    public static final AW7 LIZ = new AW7();

    static {
        Covode.recordClassIndex(28546);
    }

    private boolean LIZ() {
        return Keva.getRepo("FriendsSharePreferences").getBoolean("read_contact_denied", false);
    }

    private void LIZIZ() {
        Keva.getRepo("FriendsSharePreferences").storeBoolean("read_contact_denied", true);
    }

    public final boolean LIZ(String str) {
        0hG.LIZ(str);
        if (n.LIZ(str, "android.permission.READ_CONTACTS")) {
            return LIZ();
        }
        return Keva.getRepo("permission_store").getBoolean(str, false);
    }

    public final void LIZIZ(String str) {
        0hG.LIZ(str);
        if (n.LIZ(str, "android.permission.READ_CONTACTS")) {
            LIZIZ();
        } else {
            Keva.getRepo("permission_store").storeBoolean(str, true);
        }
    }
}
```

Figure 6: The source code for Contacts information.



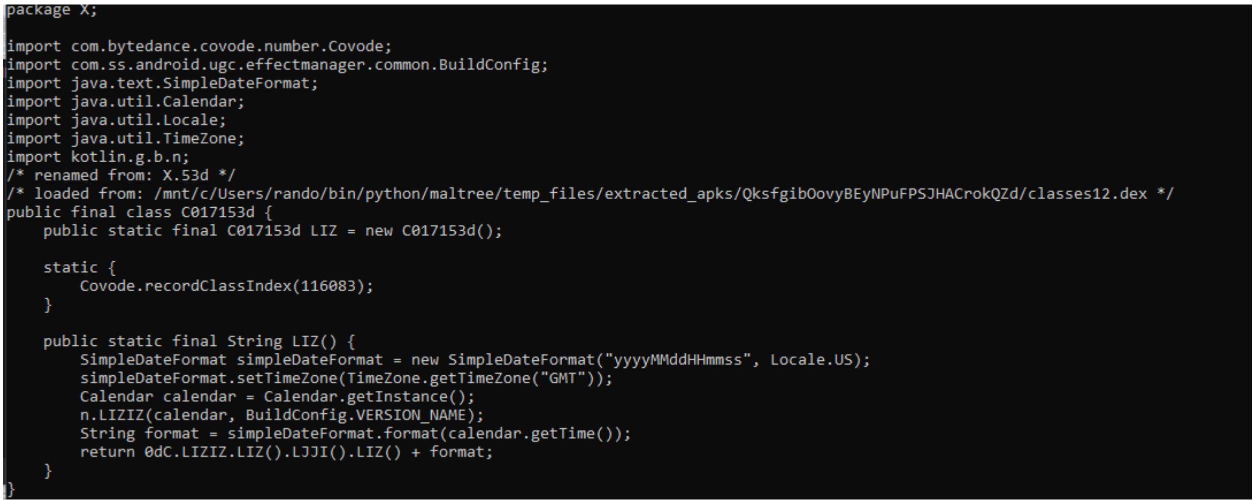
Allow **TikTok** to access your contacts?

Sync your contacts to easily find people you know on TikTok. Your contacts will only be used to help you connect with friends.

Figure 7: TikTok Contacts access request prompts while in application.

Calendar

The Android application has persistent access to read and modify calendar, see figure 8. TikTok only uses the calendar for special circumstances, for example when there is a TikTok LIVE event, based on our analysis. The persistency of access to the calendar is excessive in our opinion.

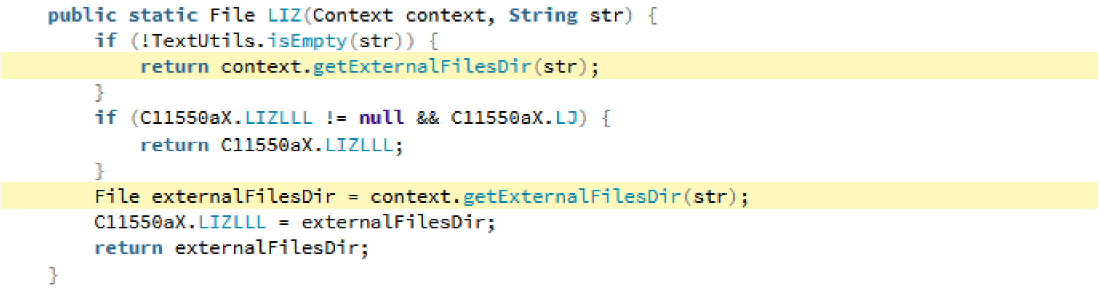


```java
package X;

import com.bytedance.covode.number.Covode;
import com.ss.android.ugc.effectmanager.common.BuildConfig;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;
import java.util.TimeZone;
import kotlin.g.b.n;
/* renamed from: X.53d */
/* loaded from: /mnt/c/Users/rando/bin/python/maltree/temp_files/extracted_apks/QksfgibOovyBEyNPuFPSJHACrokQZd/classes12.dex */
public final class C017153d {
    public static final C017153d LIZ = new C017153d();

    static {
        Covode.recordClassIndex(116083);
    }

    public static final String LIZ() {
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyyMMddHHmmss", Locale.US);
        simpleDateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        Calendar calendar = Calendar.getInstance();
        n.LIZIZ(calendar, BuildConfig.VERSION_NAME);
        String format = simpleDateFormat.format(calendar.getTime());
        return 0dC.LIZIZ.LIZ().LJJI().LIZ() + format;
    }
}
```

Figure 8: Persistent calendar access.

External storage

TikTok Android application requests access to external storage. This is a standard command for a social media application to store video and images. The aspect we list as excessive is TikTok doesn't just retrieve the ability to see folders it retrieves a list of everything available in the external storage folder where the application has the access to place files, see figure 9.



```java
public static File LIZ(Context context, String str) {
    if (!TextUtils.isEmpty(str)) {
        return context.getExternalFilesDir(str);
    }
    if (C11550aX.LIZLLL != null && C11550aX.LJ) {
        return C11550aX.LIZLLL;
    }
    File externalFilesDir = context.getExternalFilesDir(str);
    C11550aX.LIZLLL = externalFilesDir;
    return externalFilesDir;
}
```

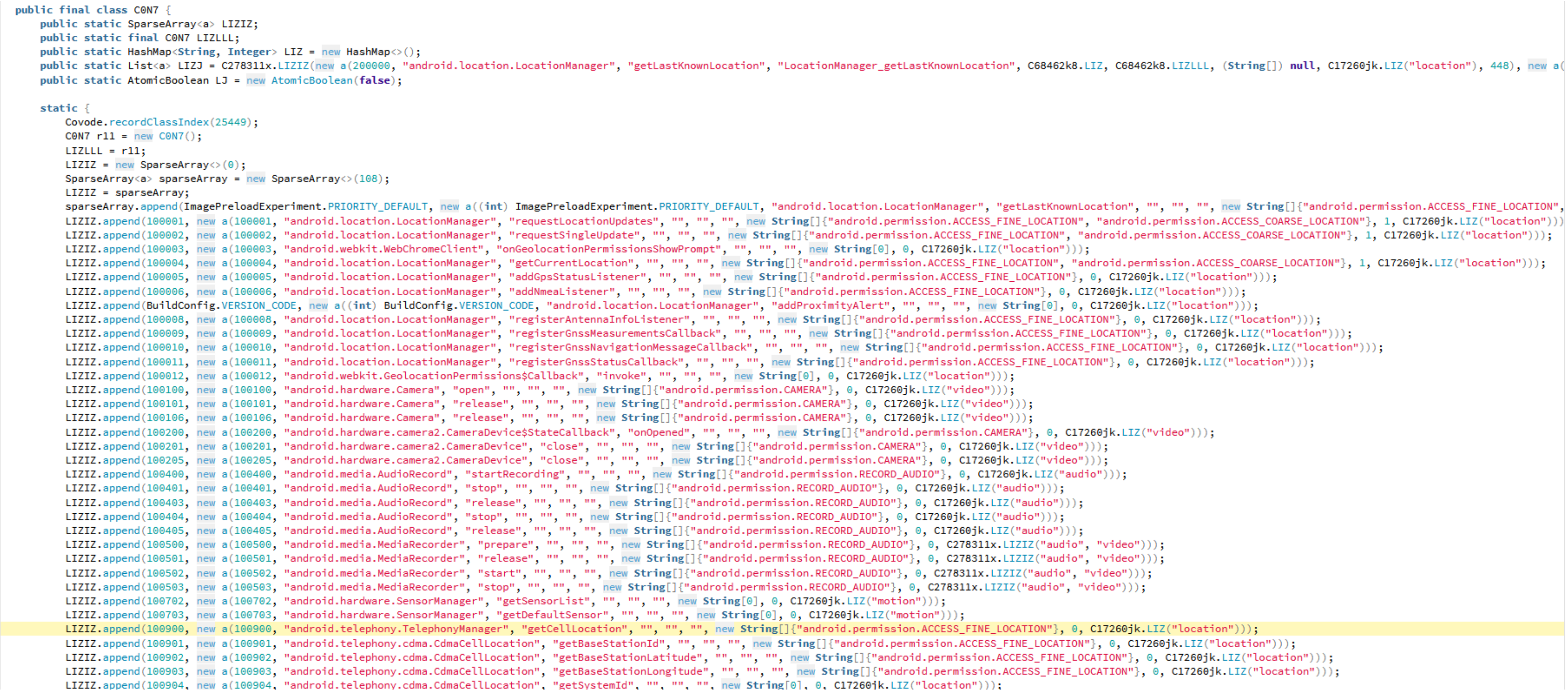
Figure 9: List everything in external storage.

Device and user data harvesting

Device Data

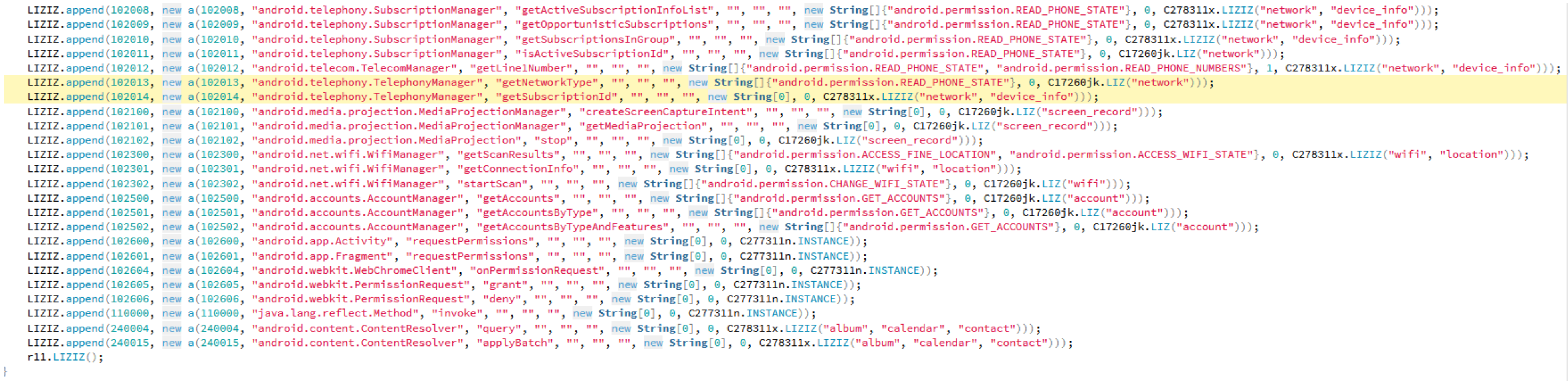
TikTok also has potential to harvest an excessive amount of data about the device, it is important to note that due to limitations with dynamic analysis it is not currently possible to determine if any of this data is ever taken from the device, however, the Android application has code that can gather the following additional device details. See figures 10-12

- Wi-Fi SSID
- Device build serial number
- SIM serial number
- Integrated Circuit Card Identification Number (this is global unique serial number that is specifically tailored to your SIM card)
- Device IMEI
- Device MAC address
- Device line number
- Device voicemail number
- GPS status information (updates on the GPS location)
- Active subscription information
- All accounts on the device
- Complete access to read the clipboard (dangerous as password managers use clipboards)



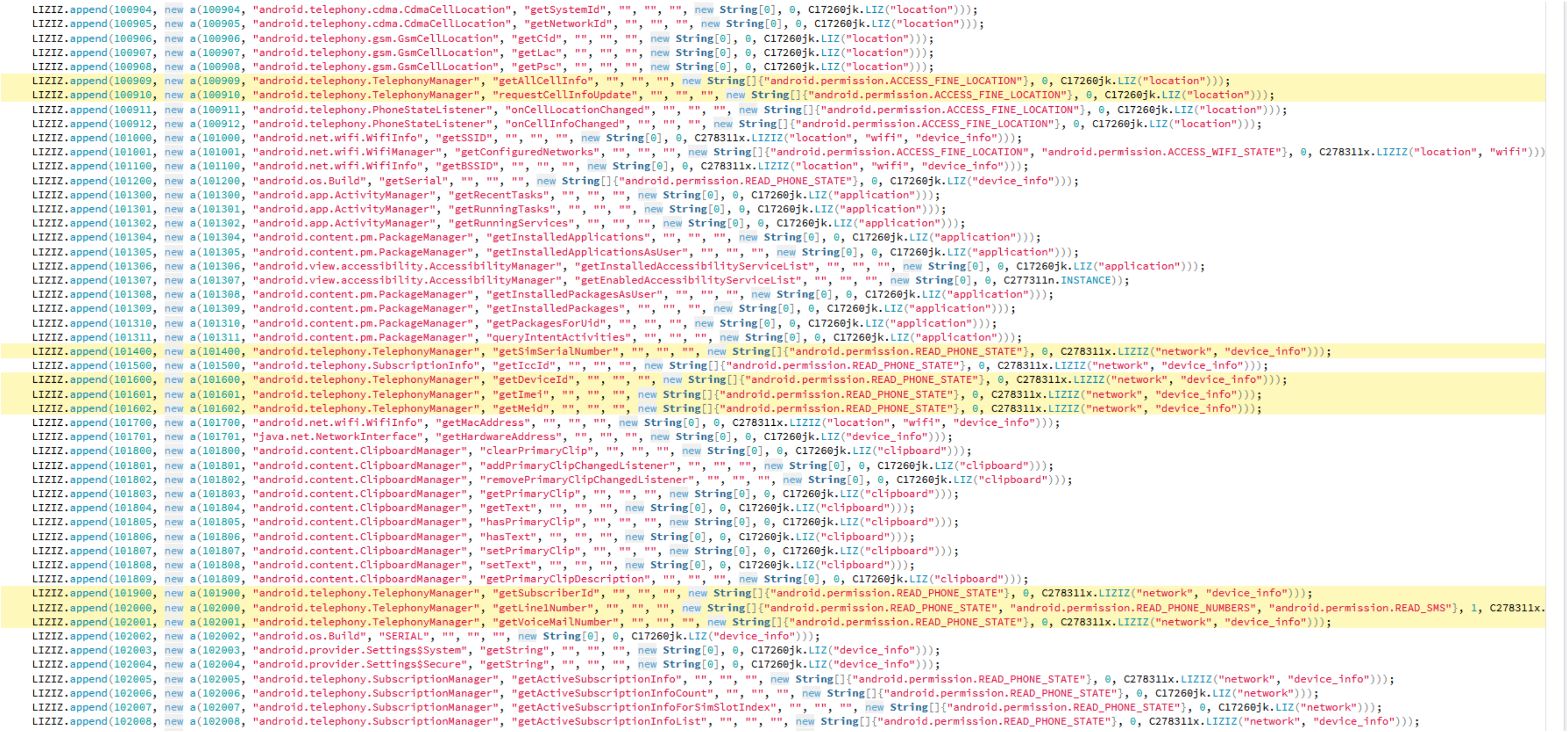
```
public final class C0N7 {
    public static SparseArray<a> LIZIZ;
    public static final C0N7 LIZLLL;
    public static HashMap<String, Integer> LIZ = new HashMap<>();
    public static List<a> LIZJ = C278311x.LIZIZ(new a(200000, "android.location.LocationManager", "getLastKnownLocation", "LocationManager_getLastKnownLocation", C68462k8.LIZ, C68462k8.LIZLLL, (String[]) null, C17260jk.LIZ("location"), 448), new a(
    public static AtomicBoolean LJ = new AtomicBoolean(false);

    static {
        Covode.recordClassIndex(25449);
        C0N7 r11 = new C0N7();
        LIZLLL = r11;
        LIZIZ = new SparseArray<>(0);
        SparseArray<a> sparseArray = new SparseArray<>(108);
        LIZIZ = sparseArray;
        sparseArray.append(ImagePreloadExperiment.PRIORITY_DEFAULT, new a((int) ImagePreloadExperiment.PRIORITY_DEFAULT, "android.location.LocationManager", "getLastKnownLocation", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION",
        LIZIZ.append(100001, new a(100001, "android.location.LocationManager", "requestLocationUpdates", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_COARSE_LOCATION"}, 1, C17260jk.LIZ("location")))
        LIZIZ.append(100002, new a(100002, "android.location.LocationManager", "requestSingleUpdate", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_COARSE_LOCATION"}, 1, C17260jk.LIZ("location")));
        LIZIZ.append(100003, new a(100003, "android.webkit.WebChromeClient", "onGeolocationPermissionsShowPrompt", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
        LIZIZ.append(100004, new a(100004, "android.location.LocationManager", "getCurrentLocation", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_COARSE_LOCATION"}, 1, C17260jk.LIZ("location")));
        LIZIZ.append(100005, new a(100005, "android.location.LocationManager", "addGpsStatusListener", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100006, new a(100006, "android.location.LocationManager", "addNmeaListener", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(BuildConfig.VERSION_CODE, new a((int) BuildConfig.VERSION_CODE, "android.location.LocationManager", "addProximityAlert", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
        LIZIZ.append(100008, new a(100008, "android.location.LocationManager", "registerAntennaInfoListener", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100009, new a(100009, "android.location.LocationManager", "registerGnssMeasurementsCallback", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100010, new a(100010, "android.location.LocationManager", "registerGnssNavigationMessageCallback", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100011, new a(100011, "android.location.LocationManager", "registerGnssStatusCallback", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100012, new a(100012, "android.webkit.GeolocationPermissions$Callback", "invoke", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
        LIZIZ.append(100100, new a(100100, "android.hardware.Camera", "open", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100101, new a(100101, "android.hardware.Camera", "release", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100106, new a(100106, "android.hardware.Camera", "release", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100200, new a(100200, "android.hardware.camera2.CameraDevice$StateCallback", "onOpened", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100201, new a(100201, "android.hardware.camera2.CameraDevice", "close", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100205, new a(100205, "android.hardware.camera2.CameraDevice", "close", "", "", "", new String[]{"android.permission.CAMERA"}, 0, C17260jk.LIZ("video")));
        LIZIZ.append(100400, new a(100400, "android.media.AudioRecord", "startRecording", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C17260jk.LIZ("audio")));
        LIZIZ.append(100401, new a(100401, "android.media.AudioRecord", "stop", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C17260jk.LIZ("audio")));
        LIZIZ.append(100403, new a(100403, "android.media.AudioRecord", "release", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C17260jk.LIZ("audio")));
        LIZIZ.append(100404, new a(100404, "android.media.AudioRecord", "stop", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C17260jk.LIZ("audio")));
        LIZIZ.append(100405, new a(100405, "android.media.AudioRecord", "release", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C17260jk.LIZ("audio")));
        LIZIZ.append(100500, new a(100500, "android.media.MediaRecorder", "prepare", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C278311x.LIZIZ("audio", "video")));
        LIZIZ.append(100501, new a(100501, "android.media.MediaRecorder", "release", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C278311x.LIZIZ("audio", "video")));
        LIZIZ.append(100502, new a(100502, "android.media.MediaRecorder", "start", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C278311x.LIZIZ("audio", "video")));
        LIZIZ.append(100503, new a(100503, "android.media.MediaRecorder", "stop", "", "", "", new String[]{"android.permission.RECORD_AUDIO"}, 0, C278311x.LIZIZ("audio", "video")));
        LIZIZ.append(100702, new a(100702, "android.hardware.SensorManager", "getSensorList", "", "", "", new String[0], 0, C17260jk.LIZ("motion")));
        LIZIZ.append(100703, new a(100703, "android.hardware.SensorManager", "getDefaultSensor", "", "", "", new String[0], 0, C17260jk.LIZ("motion")));
        LIZIZ.append(100900, new a(100900, "android.telephony.TelephonyManager", "getCellLocation", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100901, new a(100901, "android.telephony.cdma.CdmaCellLocation", "getBaseStationId", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100902, new a(100902, "android.telephony.cdma.CdmaCellLocation", "getBaseStationLatitude", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100903, new a(100903, "android.telephony.cdma.CdmaCellLocation", "getBaseStationLongitude", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
        LIZIZ.append(100904, new a(100904, "android.telephony.cdma.CdmaCellLocation", "getSystemId", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
```

Figure 10: TikTok Data harvest image.



```
    LIZIZ.append(102008, new a(102008, "android.telephony.SubscriptionManager", "getActiveSubscriptionInfoList", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
    LIZIZ.append(102009, new a(102009, "android.telephony.SubscriptionManager", "getOpportunisticSubscriptions", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
    LIZIZ.append(102010, new a(102010, "android.telephony.SubscriptionManager", "getSubscriptionsInGroup", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
    LIZIZ.append(102011, new a(102011, "android.telephony.SubscriptionManager", "isActiveSubscriptionId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("network")));
    LIZIZ.append(102012, new a(102012, "android.telecom.TelecomManager", "getLine1Number", "", "", "", new String[]{"android.permission.READ_PHONE_STATE", "android.permission.READ_PHONE_NUMBERS"}, 1, C278311x.LIZIZ("network", "device_info")));
    LIZIZ.append(102013, new a(102013, "android.telephony.TelephonyManager", "getNetworkType", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("network")));
    LIZIZ.append(102014, new a(102014, "android.telephony.TelephonyManager", "getSubscriptionId", "", "", "", new String[0], 0, C278311x.LIZIZ("network", "device_info")));
    LIZIZ.append(102100, new a(102100, "android.media.projection.MediaProjectionManager", "createScreenCaptureIntent", "", "", "", new String[0], 0, C17260jk.LIZ("screen_record")));
    LIZIZ.append(102101, new a(102101, "android.media.projection.MediaProjectionManager", "getMediaProjection", "", "", "", new String[0], 0, C17260jk.LIZ("screen_record")));
    LIZIZ.append(102102, new a(102102, "android.media.projection.MediaProjection", "stop", "", "", "", new String[0], 0, C17260jk.LIZ("screen_record")));
    LIZIZ.append(102300, new a(102300, "android.net.wifi.WifiManager", "getScanResults", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_WIFI_STATE"}, 0, C278311x.LIZIZ("wifi", "location")));
    LIZIZ.append(102301, new a(102301, "android.net.wifi.WifiManager", "getConnectionInfo", "", "", "", new String[0], 0, C278311x.LIZIZ("wifi", "location")));
    LIZIZ.append(102302, new a(102302, "android.net.wifi.WifiManager", "startScan", "", "", "", new String[]{"android.permission.CHANGE_WIFI_STATE"}, 0, C17260jk.LIZ("wifi")));
    LIZIZ.append(102500, new a(102500, "android.accounts.AccountManager", "getAccounts", "", "", "", new String[]{"android.permission.GET_ACCOUNTS"}, 0, C17260jk.LIZ("account")));
    LIZIZ.append(102501, new a(102501, "android.accounts.AccountManager", "getAccountsByType", "", "", "", new String[]{"android.permission.GET_ACCOUNTS"}, 0, C17260jk.LIZ("account")));
    LIZIZ.append(102502, new a(102502, "android.accounts.AccountManager", "getAccountsByTypeAndFeatures", "", "", "", new String[]{"android.permission.GET_ACCOUNTS"}, 0, C17260jk.LIZ("account")));
    LIZIZ.append(102600, new a(102600, "android.app.Activity", "requestPermissions", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(102601, new a(102601, "android.app.Fragment", "requestPermissions", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(102604, new a(102604, "android.webkit.WebChromeClient", "onPermissionRequest", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(102605, new a(102605, "android.webkit.PermissionRequest", "grant", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(102606, new a(102606, "android.webkit.PermissionRequest", "deny", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(110000, new a(110000, "java.lang.reflect.Method", "invoke", "", "", "", new String[0], 0, C277311n.INSTANCE));
    LIZIZ.append(240004, new a(240004, "android.content.ContentResolver", "query", "", "", "", new String[0], 0, C278311x.LIZIZ("album", "calendar", "contact")));
    LIZIZ.append(240015, new a(240015, "android.content.ContentResolver", "applyBatch", "", "", "", new String[0], 0, C278311x.LIZIZ("album", "calendar", "contact")));
    r1l.LIZIZ();
}
```

Figure 11: TikTok Data harvest image.



```
LIZIZ.append(100904, new a(100904, "android.telephony.cdma.CdmaCellLocation", "getSystemId", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
LIZIZ.append(100905, new a(100905, "android.telephony.cdma.CdmaCellLocation", "getNetworkId", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
LIZIZ.append(100906, new a(100906, "android.telephony.gsm.GsmCellLocation", "getCid", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
LIZIZ.append(100907, new a(100907, "android.telephony.gsm.GsmCellLocation", "getLac", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
LIZIZ.append(100908, new a(100908, "android.telephony.gsm.GsmCellLocation", "getPsc", "", "", "", new String[0], 0, C17260jk.LIZ("location")));
LIZIZ.append(100909, new a(100909, "android.telephony.TelephonyManager", "getAllCellInfo", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100910, new a(100910, "android.telephony.TelephonyManager", "requestCellInfoUpdate", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100911, new a(100911, "android.telephony.PhoneStateListener", "onCellLocationChanged", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(100912, new a(100912, "android.telephony.PhoneStateListener", "onCellInfoChanged", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION"}, 0, C17260jk.LIZ("location")));
LIZIZ.append(101000, new a(101000, "android.net.wifi.WifiInfo", "getSSID", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101001, new a(101001, "android.net.wifi.WifiManager", "getConfiguredNetworks", "", "", "", new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_WIFI_STATE"}, 0, C278311x.LIZIZ("location", "wifi")))
LIZIZ.append(101100, new a(101100, "android.net.wifi.WifiInfo", "getBSSID", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101200, new a(101200, "android.os.Build", "getSerial", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("device_info")));
LIZIZ.append(101300, new a(101300, "android.app.ActivityManager", "getRecentTasks", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101301, new a(101301, "android.app.ActivityManager", "getRunningTasks", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101302, new a(101302, "android.app.ActivityManager", "getRunningServices", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101304, new a(101304, "android.content.pm.PackageManager", "getInstalledApplications", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101305, new a(101305, "android.content.pm.PackageManager", "getInstalledApplicationsAsUser", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101306, new a(101306, "android.view.accessibility.AccessibilityManager", "getInstalledAccessibilityServiceList", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101307, new a(101307, "android.view.accessibility.AccessibilityManager", "getEnabledAccessibilityServiceList", "", "", "", new String[0], 0, C277311n.INSTANCE));
LIZIZ.append(101308, new a(101308, "android.content.pm.PackageManager", "getInstalledPackagesAsUser", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101309, new a(101309, "android.content.pm.PackageManager", "getInstalledPackages", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101310, new a(101310, "android.content.pm.PackageManager", "getPackagesForUid", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101311, new a(101311, "android.content.pm.PackageManager", "queryIntentActivities", "", "", "", new String[0], 0, C17260jk.LIZ("application")));
LIZIZ.append(101400, new a(101400, "android.telephony.TelephonyManager", "getSimSerialNumber", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101500, new a(101500, "android.telephony.SubscriptionInfo", "getIccId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101600, new a(101600, "android.telephony.TelephonyManager", "getDeviceId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101601, new a(101601, "android.telephony.TelephonyManager", "getImei", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101602, new a(101602, "android.telephony.TelephonyManager", "getMeid", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(101700, new a(101700, "android.net.wifi.WifiInfo", "getMacAddress", "", "", "", new String[0], 0, C278311x.LIZIZ("location", "wifi", "device_info")));
LIZIZ.append(101701, new a(101701, "java.net.NetworkInterface", "getHardwareAddress", "", "", "", new String[0], 0, C17260jk.LIZ("device_info")));
LIZIZ.append(101800, new a(101800, "android.content.ClipboardManager", "clearPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101801, new a(101801, "android.content.ClipboardManager", "addPrimaryClipChangedListener", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101802, new a(101802, "android.content.ClipboardManager", "removePrimaryClipChangedListener", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101803, new a(101803, "android.content.ClipboardManager", "getPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101804, new a(101804, "android.content.ClipboardManager", "getText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101805, new a(101805, "android.content.ClipboardManager", "hasPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101806, new a(101806, "android.content.ClipboardManager", "hasText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101807, new a(101807, "android.content.ClipboardManager", "setPrimaryClip", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101808, new a(101808, "android.content.ClipboardManager", "setText", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101809, new a(101809, "android.content.ClipboardManager", "getPrimaryClipDescription", "", "", "", new String[0], 0, C17260jk.LIZ("clipboard")));
LIZIZ.append(101900, new a(101900, "android.telephony.TelephonyManager", "getSubscriberId", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(102000, new a(102000, "android.telephony.TelephonyManager", "getLine1Number", "", "", "", new String[]{"android.permission.READ_PHONE_STATE", "android.permission.READ_PHONE_NUMBERS", "android.permission.READ_SMS"}, 1, C278311x.
LIZIZ.append(102001, new a(102001, "android.telephony.TelephonyManager", "getVoiceMailNumber", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(102002, new a(102002, "android.os.Build", "SERIAL", "", "", "", new String[0], 0, C17260jk.LIZ("device_info")));
LIZIZ.append(102003, new a(102003, "android.provider.Settings$System", "getString", "", "", "", new String[0], 0, C17260jk.LIZ("device_info")));
LIZIZ.append(102004, new a(102004, "android.provider.Settings$Secure", "getString", "", "", "", new String[0], 0, C17260jk.LIZ("device_info")));
LIZIZ.append(102005, new a(102005, "android.telephony.SubscriptionManager", "getActiveSubscriptionInfo", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
LIZIZ.append(102006, new a(102006, "android.telephony.SubscriptionManager", "getActiveSubscriptionInfoCount", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("network")));
LIZIZ.append(102007, new a(102007, "android.telephony.SubscriptionManager", "getActiveSubscriptionInfoForSimSlotIndex", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C17260jk.LIZ("network")));
LIZIZ.append(102008, new a(102008, "android.telephony.SubscriptionManager", "getActiveSubscriptionInfoList", "", "", "", new String[]{"android.permission.READ_PHONE_STATE"}, 0, C278311x.LIZIZ("network", "device_info")));
```

Figure 12: TikTok Data harvest image.

Of note: Joe's Sandbox rated the Android application as malicious for Spyware and Evader categories as seen in Figure 13 because of device and user data collection by the application and evasive techniques the application uses to block any type of analysis. Many applications have anti-sandbox run commands now to inhibit automatic analysis, the sandbox identifies these and categories it in the evader category.

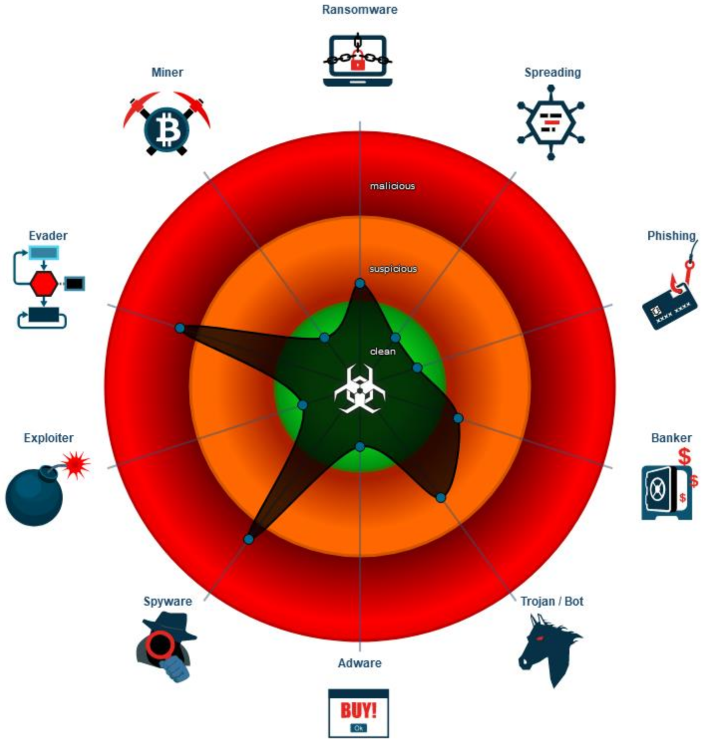


Figure 13. TikTok rating as per Joe's sandbox (https://www.joesandbox.com/).

IOS connects to mainland China

TikTok are specific in their statement that TikTok user data is stored in Singapore and the US. However, we found many subdomains in the IOS application resolving all around the world including: Sydney, Adelaide and Melbourne (Australia); New York City, Las Vegas, San Francisco, San Jose, Monrovia, Cambridge, Kansas City, Dallas, Mountain View (USA); Utama and Jakarta (Indonesia), Kuala Lumpur (Malaysia), Paris (France), Singapore (Singapore) and Baishan (China). During analysis we could not determine with high confidence the purpose for the China Server connection or where user data is stored. The China server connection is run by 贵州白山云科技股份有限公司 Guizhou Baishan Cloud Technology Co., Ltd a cloud and cyber security company. The subdomain connected to the "China server connection" resolved in multiple locations around the world including in China. The IP address resolving to China regularly changed locations, however, connectivity to Baishan Guangxi China was visible across a number of different IP addresses over time. This was confirmed through the use of a number of security products and methods, including virus total, Metasploit, security trails and sandboxing. Interestingly, this company has been rated a top 100 Chinese cyber security company and in 2021 established a joint big data laboratory with Guizhou University.[4] Of note only the IOS version had this mainland direct server connection. We could not find any direct server connections with mainland China in the Android version of the application.

Conclusion

For the TikTok application to function properly most of the access and device data collection is not required. The application can and will run successfully without any of this data being gathered. This leads us to believe that the only reason this information has been gathered is for data harvesting. It is also notable that the device only needs to ask the user for permission to perform each of these actions once and then follow the user's preferences. The application however has a culture of persistent access or continuously asking for a decision reversal by the user. The hourly checking of location is also unnecessary. Finally, device mapping, external storage access, contacts and third-party applications data collection allows TikTok the ability to reimage the phone in the likeness of the original device.

[4] https://baike.baidu.com/reference/23443686/44e44NXRi0exRZo-8rbRsVSmZl-hjxLfaZVO4j748emXOcfv_uNtLc1yLac09EyZEBSnmwlHmEjKgrSKyJqfjRJXffvnMrZx3fjyd7KgfZXHQTJqcQiSTTzNcYs12v7vcNN_
https://baike.baidu.com/reference/23443686/cc63DG_6ZWBsyHhiqR45OVCvsMnuyzIROgdcmvvuXilWB48sb7YhfKhpeWv0ZpsePYpHl2EMcS8LdZe2yWIZPp3rLCUtoQfy96e5-_uuvbQ

internet2.0

MILITARY-GRADE

CYBER PROTECTION

Australia

Level 1
18 National Circuit
Barton ACT 2600
ABN: 17 632 726

United States

Suite 100
211 N Union St
Alexandria 22314
EIN: 86-1567068

contact@internet2-0.com