Android App Usage and Cell Tower Location: Private. Sensitive. Available to Anyone?

Dr. Ryan Johnson

About Us

Quokka (formerly Kryptowire) was jump-started by the Defense Advanced Research Projects Agency (DARPA) in late 2011 and our R&D was supported by the Department of Homeland Science & Technology (DHS S&T) and the National Institute of Standards and Technology (NIST), providing mobile security and privacy solutions for various industries and public entities

Enterprise Mobile Security: Software Assurance, Developer Integration & Personal Device Management (PDM), Firmware Testing, Threat Feed, & Security Analytics



Agenda

Summary of Findings

Android Permissions

App Usage Data

Location Data

Cell Tower Location Data

Local and Remote Exposures of App Usage Data and Cell Tower Location Data

Conclusion

Summary of Findings

Exposure	Vendor(s) Impacted	Android Versions	Access Vector
Cell Tower Identity	Samsung	8 - 14	System properties
App Usage	Samsung	10 - 14	Virtual file in `/proc`
App Usage	Multiple vendors (Qualcomm app)	12 - 14	Implicit broadcast `Intent` message
App Usage	Nokia	12 - 13	Unprotected content provider component
Third-party app list transmitted via HTTP	TECNO & Infinix	11 - 12	Network sniffing
App Usage	TECNO, Infinix, & Itel	11 - 12	System settings

Android Permissions

Permissions regulate capabilities of apps and read/write access to data repositories

Each permission has a protection level (`normal`, `dangerous`, `signature`, `development`, etc.)

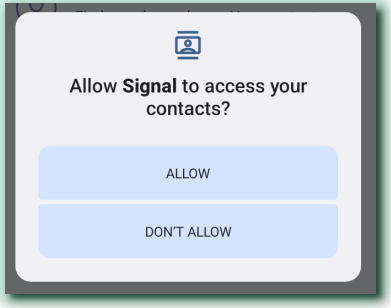
Declared by the Android Framework and apps themselves

Declaration

```
<permission android:name="android.permission.READ_CONTACTS"
    android:permissionGroup="android.permission-group.UNDEFINED"
    android:label="@string/permlab_readContacts"
    android:description="@string/permdesc_readContacts"
    android:protectionLevel="dangerous" />
```

Request

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
```



Permissions *do not always* circumscribe app behavior due to vulnerabilities, capability leaks, and data leaks

App Usage Data

Identifies the apps (by package name) that the user interacts with

Historically, the `GET_TASKS` permission guarded app usage data up until Android Lollipop (5.0)

```
<!-- @deprecated No longer enforced. -->
<permission android:name="android.permission.GET_TASKS"
android:label="@string/permlab_getTasks"
android:description="@string/permdesc_getTasks"
android:protectionLevel="normal" />
```

`REAL_GET_TASKS` permission is not obtainable by third-party apps

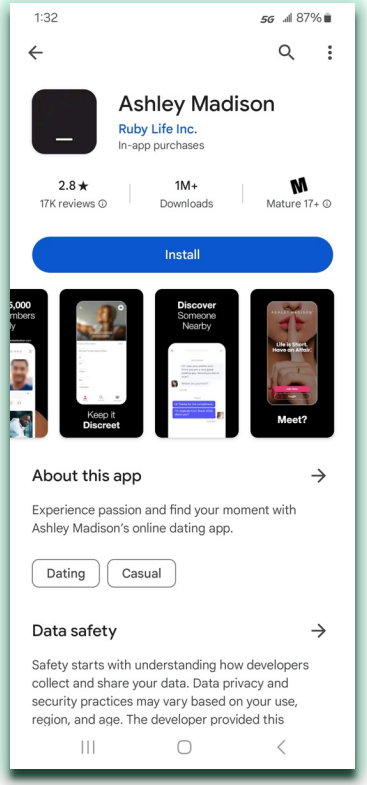
```
<permission android:name="android.permission.REAL_GET_TASKS"
android:protectionLevel="signature|privileged" />
```

Users can grant the `PACKAGE_USAGE_STATS` permission to third-party apps via the special access menu in the Settings app

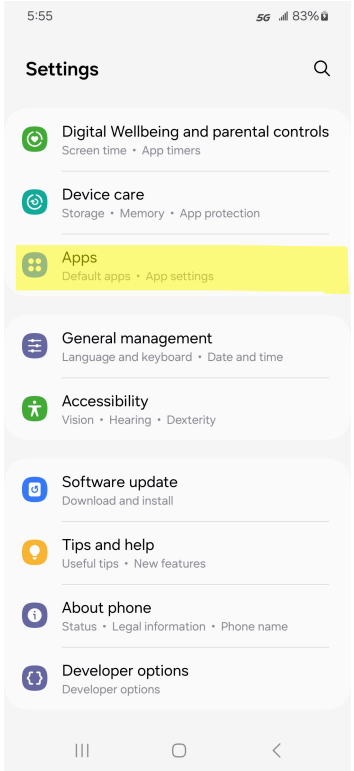
```
<permission android:name="android.permission.PACKAGE_USAGE_STATS"
android:protectionLevel="signature|privileged|development|appop|retailDemo" />
```

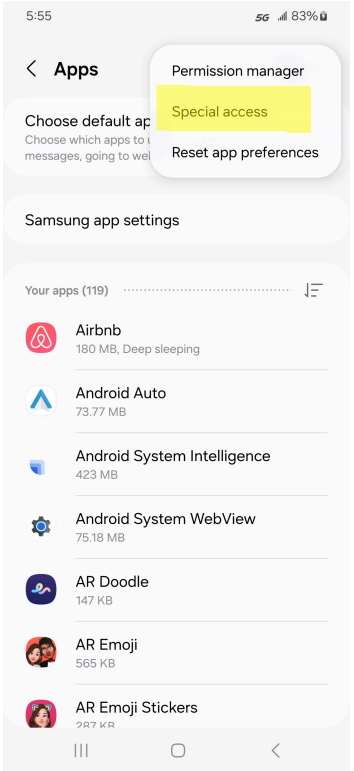
Controversial apps can be dependent on culture and context

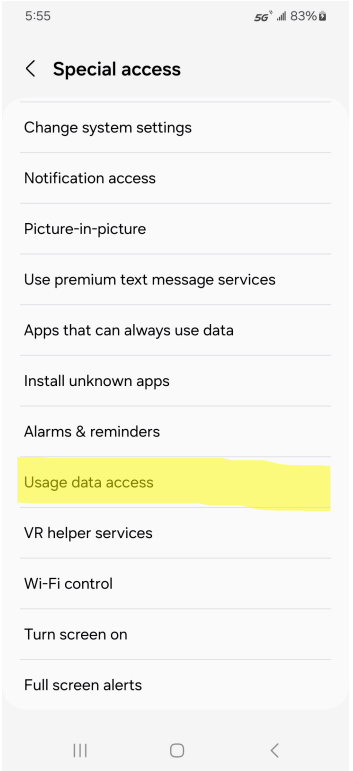
(ಠ_ಠ)

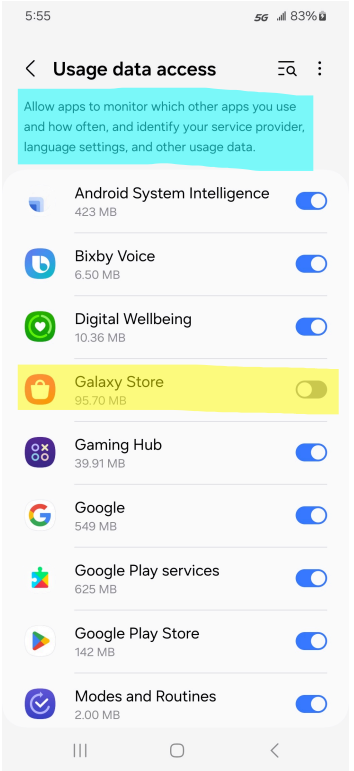


Workflow for Granting App Usage Data to Apps









Location Data

Differing levels of location accuracy (FINE or COARSE) and category (foreground or background)

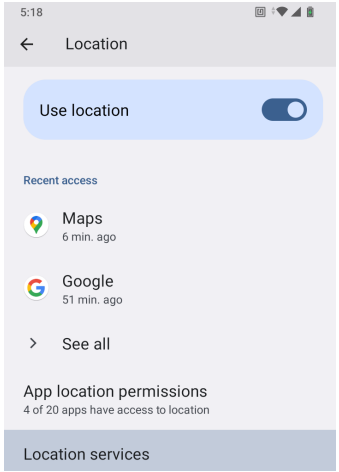
```
<permission android:name="android.permission.ACCESS_FINE_LOCATION" android:permissionGroup="android.permission-group.UNDEFINED"
android:label="@string/permlab_accessFineLocation" android:description="@string/permdesc_accessFineLocation"
android:backgroundPermission="android.permission.ACCESS_BACKGROUND_LOCATION" android:protectionLevel="dangerous|instant"/>
```

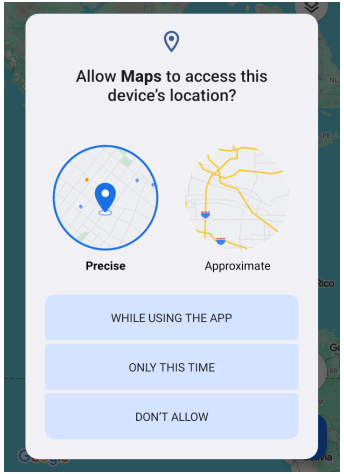
```
<permission android:name="android.permission.ACCESS_COARSE_LOCATION" android:permissionGroup="android.permission-group.UNDEFINED"
android:label="@string/permlab_accessCoarseLocation" android:description="@string/permdesc_accessCoarseLocation"
android:backgroundPermission="android.permission.ACCESS_BACKGROUND_LOCATION" android:protectionLevel="dangerous|instant"/>
```

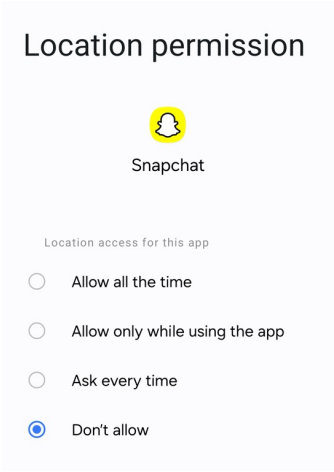
Locations restrictions based are on the app's target SDK level

Network-based location uses data from cell towers and Wi-Fi access points

Location services *can* be disabled by the user







Threat Model

User downloads and installs a third-party app targeting Android 13 (API level 33) that requests two permissions solely for persistent execution: `RECEIVE_BOOT_COMPLETED` and `FOREGROUND_SERVICE`

None of the local vulnerabilities require any access permissions or special privileges

App executes in the background gathering leaked data until the user uninstalls the app

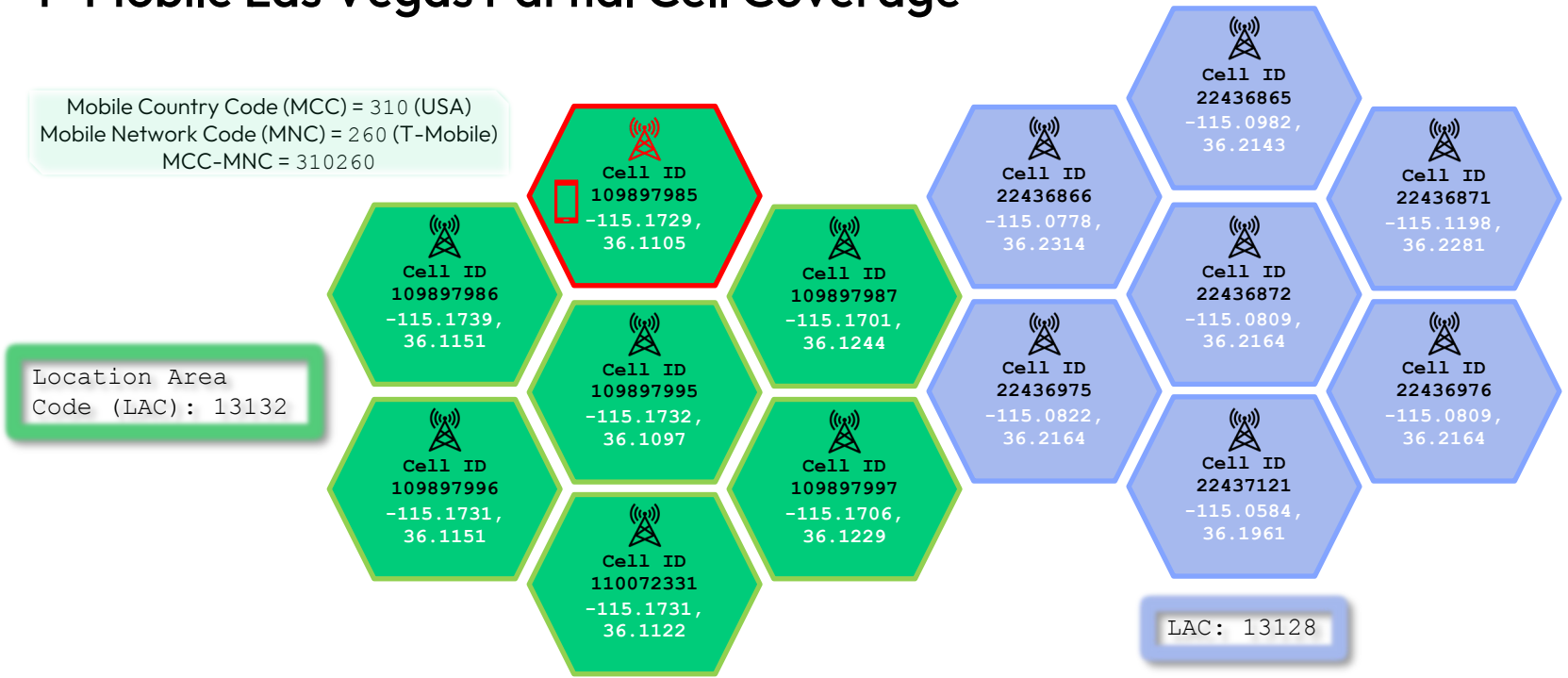
Optionally requests the `INTERNET` permission for transmission of harvested data

For remote exposures, attacker must be present on the local network or on-path



T-Mobile Las Vegas Partial Cell Coverage

Mobile Country Code (MCC) = 310 (USA)
Mobile Network Code (MNC) = 260 (T-Mobile)
MCC-MNC = 310260



Standard APIs to directly obtain cell tower data requires the caller to posses the `ACCESS_FINE_LOCATION` permission

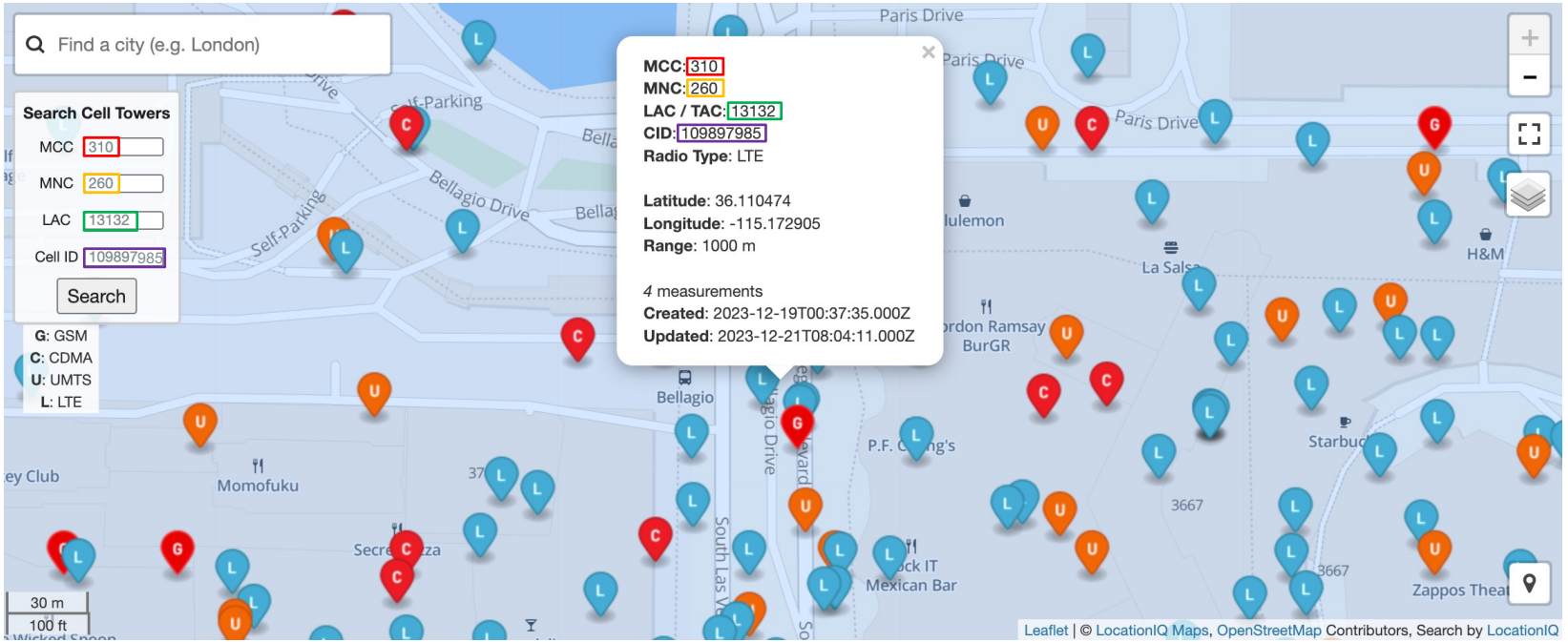
- `java.util.List<android.telephony.CellInfo> android.telephony.TelephonyManager.getAllCellInfo()`
- `android.telephony.CellLocation android.telephony.TelephonyManager.getCellLocation()` – Deprecated in Android 8.0

Mapping Cell Tower Identity to GPS Coordinates

MCC-MNC: 310260
LAC: 13132
Cell ID (CID): 109897985

Longitude = 36.110474
Latitude = -115.172905

https://opencellid.org/#zoom=18&lat=36.110474&lon=-115.172905



System Properties

A system-wide repository of key-value pairs

Can be accessed by apps via executing the `getprop` command

Have been known to contain non-resettable device IDs in the past

Access control is enforced by SELinux

Radio properties have an SELinux context of `u:object_r:radio_prop:s0` that is readable by third-party apps and generally have a prefix of `ril` (Radio Interface Layer)

Third-party apps cannot set system properties, but more privileged processes can set them using the `setprop` command or the `SystemProperties` APIs that are hidden from the public Android API

```
b0q:/ $ getprop
[DEVICE_PROVISIONED]: [1]
[aaudio.hw_burst_min_usec]: [2000]
[aaudio.mmap_exclusive_policy]: [2]
[aaudio.mmap_policy]: [2]
[apex.all.ready]: [true]
[arm64.memtag.process.system_server]: [off]
[audio.sys.mute.latency.factor]: [2]
[audio.sys.noisy.broadcast.delay]: [500]
[audio.sys.offload.pstimeout.secs]: [3]
[audio.sys.routing.latency]: [0]
...
```

```
b0q:/ $ getprop -Z
[DEVICE_PROVISIONED]: [u:object_r:default_prop:s0]
[aaudio.hw_burst_min_usec]: [u:object_r:aaudio_config_prop:s0]
[aaudio.mmap_exclusive_policy]: [u:object_r:aaudio_config_prop:s0]
[aaudio.mmap_policy]: [u:object_r:aaudio_config_prop:s0]
[apex.all.ready]: [u:object_r:apex_ready_prop:s0]
[arm64.memtag.process.system_server]:
[u:object_r:arm64_memtag_prop:s0]
[audio.sys.mute.latency.factor]: [u:object_r:default_prop:s0]
[audio.sys.noisy.broadcast.delay]: [u:object_r:default_prop:s0]
[audio.sys.offload.pstimeout.secs]: [u:object_r:default_prop:s0]
[audio.sys.routing.latency]: [u:object_r:default_prop:s0]
...
```

Samsung Cell Tower Identity Exposure

Vulnerability: Accessing three or four system properties on Samsung smartphones reveals the cell tower identity to which the device first connected when powered on; free online databases can map the cell tower identity to GPS coordinates

Versions Impacted: Android 8 – 14

Software Responsible: `rild` binary

Root cause: Lack of access control on the system properties containing the cell tower identity

GSM	CDMA
`ril.CHAR` – Cell ID (CID)	`ril.BRAVO` – Base Station ID (BID)
`ril.LIMA` – Location Area Code (LAC)	`ril.SIERRA` – System ID (SID)
---	`ril.NOVEMBER` – Network ID (NID)
`gsm.operator.numeric` – MCC-MNC	`ro.cdma.home.operator.numeric` – MCC-MNC `gsm.operator.numeric` (unreliable) – MCC-MNC

Check if your GSM device is vulnerable:

```
% adb shell getprop | grep -e ril.CHAR -e ril.LIMA
-e gsm.operator.numeric
[gsm.operator.numeric]: [310260,]
[ril.CHAR]: [109897985]
[ril.LIMA]: [13132]
```

MCC MNC LAC CID
310 260 13132 109897985

Public Land Mobile Network (PLMN) = MCC MNC

Location Area Identity (LAI) = MCC MNC LAC

Cell Global Identity (CGI) = MCC MNC LAC CID

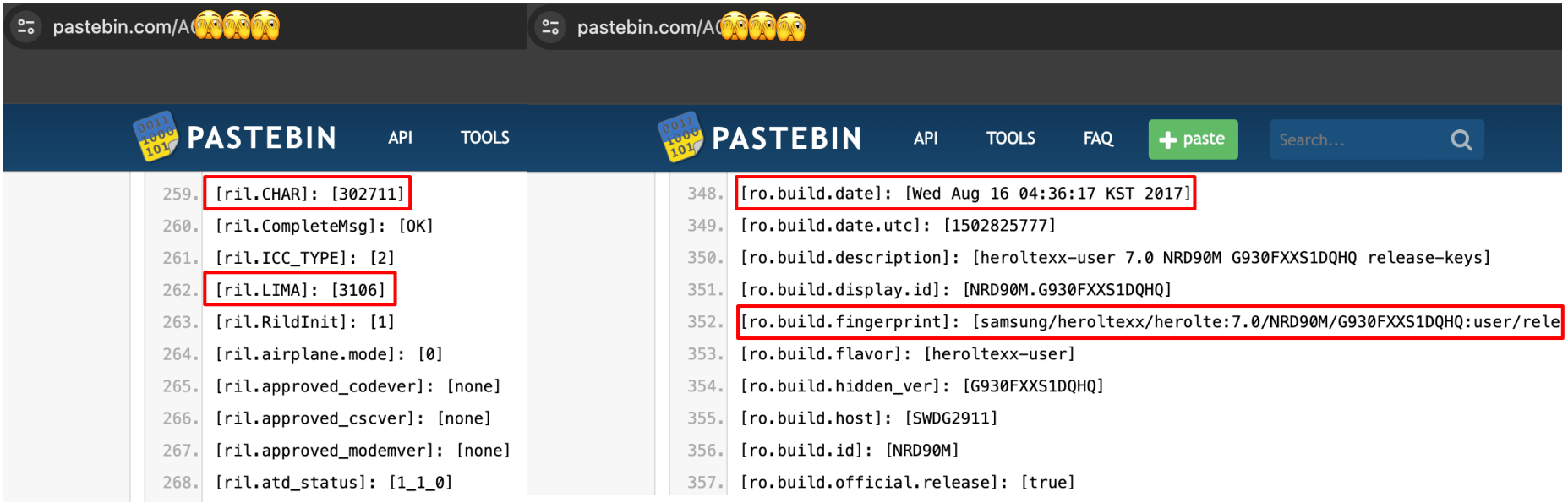
Samsung Cell Tower Identity Exposure Exploitation Demo



Sample of Samsung Smartphones Impacted – Cell Tower Identity Exposure

Model	Build Fingerprint	Build Date
Galaxy A25 5G	samsung/a25xdxx/a25x:14/UP1A.231005.007/A256EXXS2AXCC:user/release-keys	Wed Apr 10 2024
Galaxy S22 Ultra	samsung/b0quew/b0q:14/UP1A.231005.007/S908U1UES4DXD1:user/release-keys	Mon Apr 1 2024
Galaxy S21 Ultra 5G	samsung/p3quew/p3q:14/UP1A.231005.007/G998U1UESAFXD1:user/release-keys	Mon Apr 1 2024
Galaxy A13 5G	samsung/a13xtfn/a13x:13/TP1A.220624.014/S136DLUDS7DWH3:user/release-keys	Fri Aug 25 2023
Galaxy A03s	samsung/a03sutfn/a03su:13/TP1A.220624.014/S134DLUDU6CWB6:user/release-keys	Mon Feb 20 2023
Galaxy A14	samsung/a14mnnxx/a14m:13/TP1A.220624.014/A145PXXU1AWB1:user/release-keys	Fri Feb 10 2023
Galaxy A03 Core	samsung/a3coreub/a3core:11/RP1A.201005.001/A032MUBU1AUKC:user/release-keys	Wed Nov 17 2021
Galaxy S10+	samsung/beyond2ltexx/beyond2:10/QP1A.190711.020/G975FXXS9DTK9:user/release-keys	Fri Nov 13 2020
Galaxy A10e	samsung/a10etfn/a10e:9/PPR1.180610.011/S102DLUDS3ATA1:user/release-keys	Wed Jan 29 2020
Galaxy S8	samsung/dreamltexx/dreamlte:8.0.0/R16NW/G950FXXS4CRLB:user/release-keys	Wed Dec 26 2018

Online Resources Can Contain Cell Tower Identity Exposure



Online listings of system properties or `bugreport`/`dumpstate` files since at least 2018 (or earlier) from Samsung smartphones can contain information that reveal cell tower locations even when location services are disabled

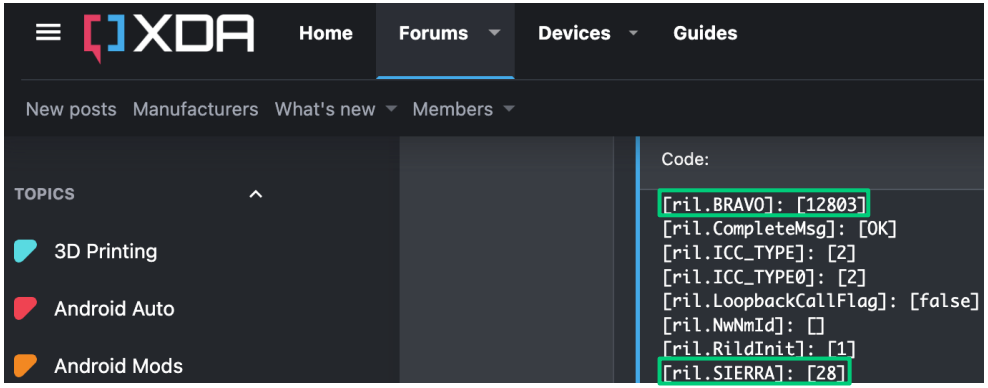
Does not require a SIM card to be inserted into the smartphone for the system properties to appear on smartphones in the United States, as they can call emergency services without a SIM

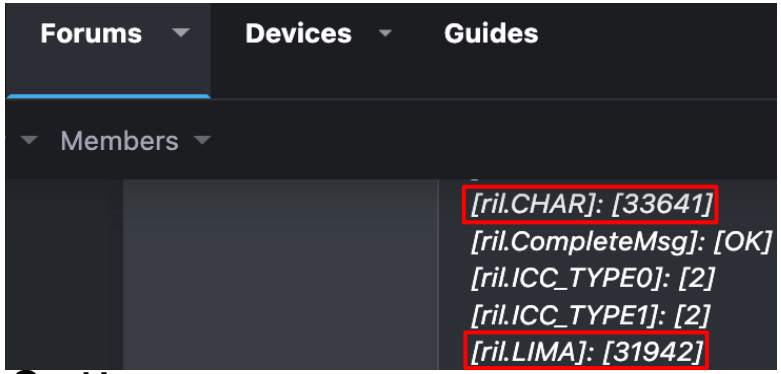
Online Resources Can Contain Cell Tower Identity Exposure

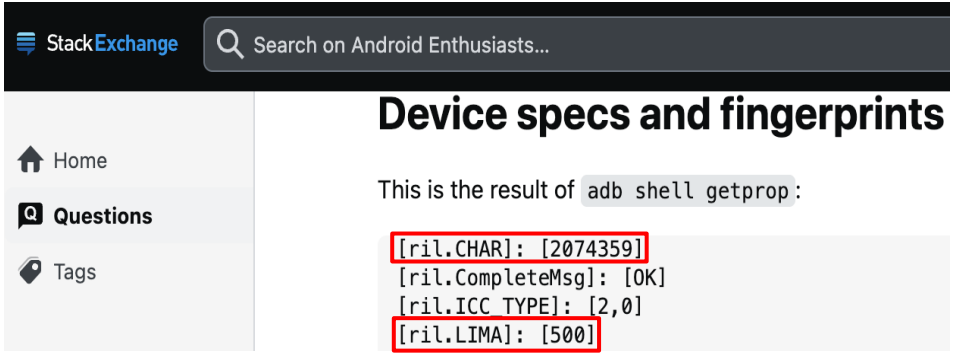
build.prop / 三星-SM-G6100.txt

Code | Blame | 1245 lines (1241 loc) · 42.4 KB

```
933     [radio.atfwd.start]: [false]
934     [ril.CHAR]: [19400434]
935     [ril.CompleteMsg]: [OK]
936     [ril.ICC_TYPE]: [0,0]
937     [ril.IsCSIM]: [0,0]
938     [ril.LIMA]: [41025]
939     [ril.RildInit]: [1,1]
```





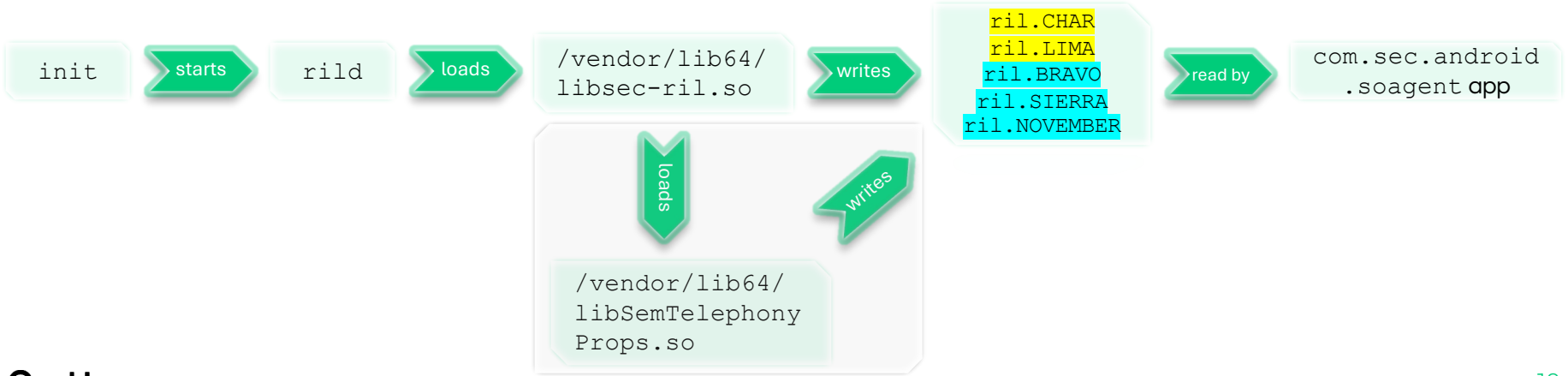


A Pinch of Indirection

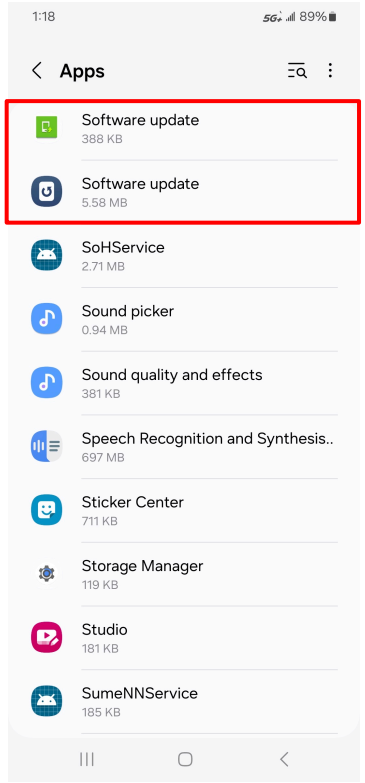
The `com.sec.android.soagent` app has adequate privileges (i.e., `system` UID) to access the cell tower information directly using the standard `TelephonyManager` APIs instead of reading from the system properties that contain the cell tower identity

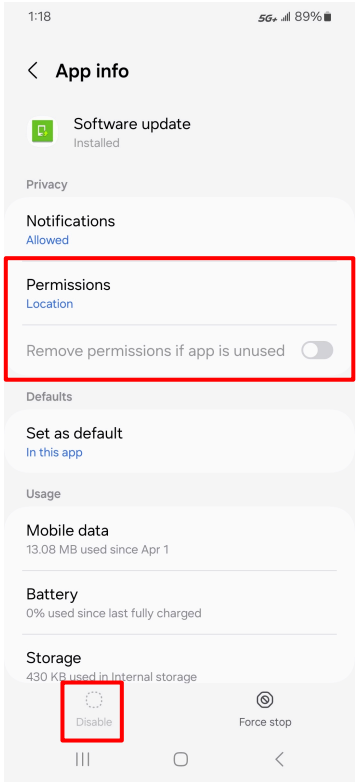
Indirect access (paired with no access control) leads to the local exposure of cell tower identity via these system properties

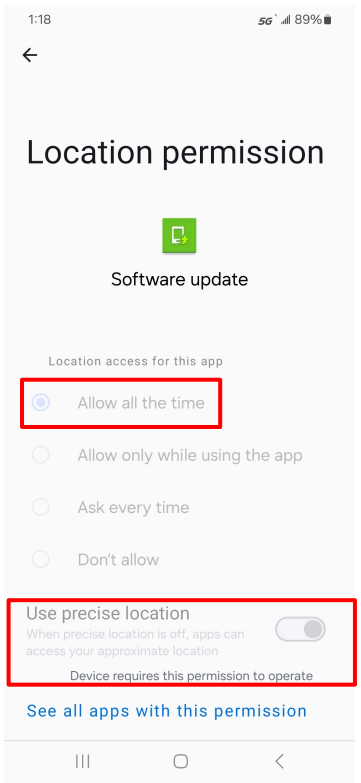
Perhaps there are multiple consumers of the system properties?



App That Accesses Cell Tower Identity System Properties







Samsung Galaxy A25 5G - (`samsung/a25xdxx/a25x:14/UP1A.231005.007/A256EXXS2AXCC:user/release-keys`)

Package name: `com.sec.android.soagent`

Label: `Software update`

Path: `/system/priv-app/SOAgent7/SOAgent7.apk`

Shared UID: `android.uid.system`

Version code: `730501000`

Version name: `7.3.05`

Platform build version code: `34`

Platform build version name: `14`

Identifier Renaming Obfuscation in the Software Update App

```
public abstract class lIIIIIIIIllllllllIIll extends llIIIIllllllIIIIIIIll {
    public static final String llllIIIIIIIIIIIIIIIII = "lIIIIIIIIIIIIIIIIIII";

    ...

    public void IllIIIIIIIIIIIIIIIl(JSONObject jSONObject) {
        try {
            String str = lllIIIIIIIIIIIIIIIII;
            lIIIIlllllIIIIIIIIII.llIIIIIIIIIIIIIIIIIII(str, "Collecting cellInfo");
            int lIIIIIIIIIIIIIIIIIII = lIIIIIIIIIIIIIIIIIl();
            if (1 == lIIIIIIIIIIIIIIIIIl) {
                jSONObject.put("deviceNetworkType", "CDMA");
                jSONObject.put("deviceNetworkCellInfo", SemSystemProperties.get("ril.BRAVO", ""));
                jSONObject.put("deviceNetworkLocationAreaInfo", SemSystemProperties.get("ril.SIERRA", ""));
                jSONObject.put("deviceNetworkNid", SemSystemProperties.get("ril.NOVEMBER", ""));
                lIIIIlllllIIIIIIIIII.lIIIIIIIIIIIIIlIIII(str, "SOAnid : " + SemSystemProperties.get("ril.NOVEMBER", ""));
            } else if (lIIIIIIIIIIIIIIIIIl == 0) {
                jSONObject.put("deviceNetworkType", "GSM");
                jSONObject.put("deviceNetworkCellInfo", SemSystemProperties.get("ril.CHAR", ""));
                jSONObject.put("deviceNetworkLocationAreaInfo", SemSystemProperties.get("ril.LIMA", ""));
            } else {
                jSONObject.put("deviceNetworkType", "NONE");
            }
        } catch (JSONException e) {
            String str2 = lllIIIIIIIIIIIIIIIII;
            lIIIIlllllIIIIIIIIII.llIIIIIIIIIIIIIIIIIl(str2, "JSONException: " + e);
        }
    }

...
```

Code snippet of the `void lIIIIIIIIllllllllIIll.IlllIIIIIIIIIIIIIIl(JSONObject)` method that accesses various system properties and then adds them to a `JSONObject` prior to transmitting them to either the `https://dir-apis.samsungdm.com/api/v1/device` URL (default) or an alternate URL which has specific requirements: `https://dir-apis.samsung.com.cn/api/v1/device`

Remote Transmission of the Cell Tower Identity

`https://dir-apis.samsungdm.com/api/v1/device`

Frequency: Sent only once after initial device setup (or until a factory reset)

Data Transmitted (All app versions): IMEI value(s), device serial number, unique number, MNC-MCC by network operator, MNC-MCC by SIM card, CID, and LAC (or CDMA equivalents)

Responsible Component: `com.sec.android.soagent/.service.AddJobService`

`https://dir-apis.samsungdm.com/api/v1/device/heartbeat`

Frequency: Every 14 days (assuming no reboot)

Data Transmitted (All app versions): IMEI values(s), device serial number, unique number, MNC-MCC by network operator, and MNC-MCC by SIM card

Data Transmitted (App versions 4.1.18 - 5.0.16): *Additionally* transmits the CID and LAC (or CDMA equivalents) but this depends on settings in the device's Country Specific Code (CSC) configuration file

Responsible Component: `com.sec.android.soagent/.service.HeartBeatJobService`

Roadblocks to Capturing HTTPS Network Traffic

The `com.sec.android.soagent` app uses HTTPS for all network connections

The app does not trust root Certificate Authorities (CAs) installed by the user

```
info: [16:15:50.082][192.168.2.49:59970] client connect
info: [16:15:50.183][192.168.2.49:59970] server connect dir-apis.samsungdm.com:443 (18.200.58.29:443)
warn: [16:15:50.434][192.168.2.49:59970] Client TLS handshake failed. The client does not trust the proxy's
certificate for dir-apis.samsungdm.com (OpenSSL
Error([('SSL routines', '', 'ssl/tls alert certificate unknown')]))
info: [16:15:50.435][192.168.2.49:59970] client disconnect
info: [16:15:50.436][192.168.2.49:59970] server disconnect dir-apis.samsungdm.com:443 (18.200.58.29:443)

E SOAgent : [HeartBeatJobService]java.security.cert.CertPathValidatorException: Trust
anchor for certification path not found.
```

Use a hooking framework to hook and modify the following APIs for the `com.sec.android.soagent` app:

`java.net.Url(java.lang.String)` – Change `https://` prefix to `http://` for the `String` parameter since the app uses `java.net.HttpURLConnection` instead of `java.net.HttpsURLConnection` therefore no `ClassCastException` is thrown

`boolean com.squareup.okhttp.HttpHandler$CleartextURLFilter.checkURLPermitted(java.lang.String)` – Always return `true` to allow HTTP network calls since the app does not explicitly allow them in its `AndroidManifest.xml` file as it lacks the `android:usesCleartextTraffic` and `android:networkSecurityConfig` attributes in its `application` element

Remote Transmission of Cell Tower Identity (Galaxy A25 5G)

Flow Details

2024-05-25 19:45:40 POST http://dir-apis.samsungdm.com/api/v1/device

← 200 OK text/html [no content] 4ms

Request | Response

Content-Type:	application/json
Accept:	application/json
Authorization:	consumer_id="CE11237B19E300CB347E",signature="uuEV3tlEEkrfsZ/1mcaWMzhBIsO6TTK9
User-Agent:	Dalvik/2.1.0 (Linux; U; Android 14; SM-A256E Build/UP1A.231005.007)
Host:	dir-apis.samsungdm.com
Connection:	Keep-Alive
Accept-Encoding:	gzip
Content-Length:	1136

JSON


```
{
    "deviceVO": {
        "bitInfo": "{\"WB\":1,\"TB\":1,\"ABS\":1,\"Reason\":\"F\",\"BinaryStatus\":{\"R\":2,\"B\"
:2,\"U\":2,\"H\":0,\"O\":2,\"DT\":2,\"DO\":2,\"ES\":0,\"ET\":\"0\",\"HDM\":\"FFFFFFFF\"}}",
        "clientVersion": "7.3.05",
        "countryIso": "GT",
        "customerCode": "GTO",
        "dataNetworkCellInfo": "[redacted]",
        "dataNetworkCellType": "LTE",
        "dataNetworkLocationAreaInfo": "[redacted]",
        "dataNetworkType": 13,
        "deviceID": "IMEI:350616259196813",
        "deviceModelName": "SM-A256E",
        "deviceNetworkCellInfo": "[redacted]",
        "deviceNetworkLocationAreaInfo": "[redacted]",
        "deviceNetworkType": "GSM",
        "eulaVersion": "E1.02.02|P1.04.08",
        "fingerPrint": "samsung/a25xdxx/a25x:14/UP1A.231005.007/A256EXXS2AXCC:user/release-keys",
        "fwVersion": "A256EXXS2AXCC/A256EOWO2AXCC/A256EXXS2AXCC",
        "mccByDevice": "712",
        "mccByNetwork": "310",
        "mccBySIM": "310",
        "mncByNetwork": "260",
        "mncBySIM": "240",
        "networkBearer": "1",
        "pcb2d": "G3B021329SP0R",
        "rooting": "",
        "secType": "N",
        "secondDeviceID": "IMEI:350960289196819",
        "securityPatchVersion": "2024-04-01",
        "sepVersion": "15.0",
        "serialNumber": "R5CX2055KGB",
        "sk": "AWJCcm6h3MoAd1716680725829BkIEIL",
        "uniqueNumber": "CE11237B19E300CB347E"
    }
}
```


`https://dir-apis.samsungdm.com/api/v1/device`

Cell Tower Info


```
"dataNetworkCellInfo": "[redacted]"
"dataNetworkLocationAreaInfo": "[redacted]",
```


Network and SIM Info


```
"mccByDevice": "712",
"mccByNetwork": "310",
"mccBySIM": "310",
"mncByNetwork": "260",
"mncBySIM": "240",
```


Device identifiers


```
"deviceID": "IMEI:350616259196813",
"secondDeviceID": "IMEI:350960289196819"
```


Serial Number and Unique Value


```
"serialNumber": "R5CX2055KGB",
"uniqueNumber": "CE11237B19E300CB347E"
```

Requirements for Persistent Cell Identity Transmission

Version name of the `com.sec.android.soagent` app is `4.1.18` to `5.0.16` (i.e., Android 8 and 9)
AND
Any of the following conditions are true with respect to the CSC configuration file (which has a path of `/system/omc/<CSC code>/cscfeature.xml` on a Galaxy S8 smartphone running Android 8):

(1) `CountryISO` element has a value of `CN` or `cn`

(2) `CscFeature_Common_EulaVersion` element has a value that is greater than or equal to `1`

(3) `CscFeature_SetupWizard_DisablePrivacyPolicyAgreement` element has a value that is equal to `1`

```
<?xml  version="1.0" encoding="UTF-8" ?>
<SamsungMobileFeature>
  <Version>ED00008</Version>
  <Country>UNITED ARAB EMIRAT</Country>
  <CountryISO>AE</CountryISO>
  <SalesCode>XSG</SalesCode>
  <FeatureSet>
    <CscFeature_Audio_ConfigDefCallSampleRate>SWB</CscFeature_Audio_ConfigDefCallSampleRate>
    <CscFeature_Calendar_EnableLocalHolidayDisplay>ARABIC</CscFeature_Calendar_EnableLocalHolidayDisplay>
    <CscFeature_Calendar_SetColorOfDays>XXXXBRX</CscFeature_Calendar_SetColorOfDays>
    <CscFeature_Clock_DisableIsraelCountry>TRUE</CscFeature_Clock_DisableIsraelCountry>
    <CscFeature_Common_AutoConfigurationType>NO_DFLT_CSC, SIMBASED_OMC</CscFeature_Common_AutoConfigurationType>
    <CscFeature_Common_ConfigAllowedPackagesDuringDataSaving>com.sec.android.daemonapp</CscFeature_Common_ConfigAllowedPackagesDuringDataSaving>
    <CscFeature_Common_ConfigSvcProviderForUnknownNumber>off,off,off</CscFeature_Common_ConfigSvcProviderForUnknownNumber>
    <CscFeature_Common_EulaVersion>2</CscFeature_Common_EulaVersion>
    ...
```

Periodic Transmission of Cell Tower Identity (Galaxy S8)

```
Flow Details
2024-05-25 19:02:51 PUT http://dir-apis.samsungdm.com/api/v1/device/heartbeat
                    ← 200 OK text/html [no content] 4ms
Request
Content-Type:       application/json
Accept:             application/xml
Authorization:      consumer_id="CE11182B884EA00302",access_token="NTg3MjAzMjcyRGlS",signature="D8I
User-Agent:         Dalvik/2.1.0 (Linux; U; Android 8.0.0; SM-G950F Build/R16NW)
Host:               dir-apis.samsungdm.com
Connection:         Keep-Alive
Accept-Encoding:    gzip
Content-Length:     643
JSON
{
    "deviceVO": {
        "clientVersion": "4.1.18",
        "customerCode": "XSG",
        "deviceID": "IMEI:355258098663920",
        "deviceModelName": "SM-G950F",
        "deviceNetworkCellInfo": "[illegible]",
        "deviceNetworkLocationAreaInfo": "[illegible]",
        "deviceNetworkType": "GSM",
        "eulaVersion": 2,
        "fingerPrint": "samsung/dreamltexx/dreamlte:8.0.0/R16NW/G950FXXS4CRLB:user/release-keys",
        "fwVersion": "G950FXXS4CRLB/G950FOXM4CRL1/G950FXXU4CRKB",
        "mccByDevice": "424",
        "mccByNetwork": "310",
        "mccBySIM": "310",
        "mncByNetwork": "260",
        "mncBySIM": "240",
        "networkBearer": "WIFI",
        "rooting": "Y",
        "secType": "N",
        "secondDeviceID": "IMEI:355259098663928",
        "serialNumber": "RF8M10XL2HZ",
        "uniqueNumber": "CE11182B884EA00302"
    }
}
```

`https://dir-apis.samsungdm.com/api/v1/device/heartbeat`

Cell Tower Info

```
"deviceNetworkCellInfo": "[illegible]",
"deviceNetworkLocationAreaInfo": "[illegible]",
```

Network and SIM Info

```
"mccByDevice": "424",
"mccByNetwork": "310",
"mccBySIM": "310",
"mncByNetwork": "260",
"mncBySIM": "240",
```

Device identifiers

```
"deviceID": "IMEI:355258098663920",
"secondDeviceID": "IMEI:355259098663928",
```

Serial Number and Unique Value

```
"serialNumber": "RF8M10XL2HZ",
"uniqueNumber": "CE11182B884EA00302"
```

Network Endpoint Selection in Recent System Update App Versions

```
public abstract class lllIllIIlIIllllIIlIl {
    public static final String llIIIIllllIIllIIllIl = "lllIllIIlIIllllIIlIl";
    public static final byte[] llIlIIIIlIIIIIlIlIII = {1, 15, 5, 11, 19, 28, 23, 47, 35, 44, 2, 14, 6, 10, 18, 13, 22, 26, 32, 47, 3, 13,
7, 9, 17, 30, 21, 25, 33, 45, 4, 12, 8, 63, 16, 31, 20, 24, 34, 46};
    public static String lllIlIlIIIllIIlIllIl = "https://api.samsungidp.com";
    public static String llllIIIllIlIIIIllllI = "https://dir-apis.samsungdm.com";


    public static void IllIlIIIIlIIlIIIllIl(int i) {
        if (i != 0) {
            llllIIIllIlIIIIllllI = "https://dir-apis.samsungdm.com";
        } else {
            llllIIIllIlIIIIllllI = "https://dir-apis.samsung.com.cn";
        }
        String str = llIIIIllllIIllIIllI;
        lIllIllllIIIllllIllI.llIlIIIIlIIIIIlIlIII(str, "ServerType is " + i);
    }
...
```

The `com.sec.android.soagent` app selects the non-default `https://dir-apis.samsung.com.cn` URL if any of the following conditions are true:

(1) Network operator has an MCC value of `460` (China)

(2) Network operator is `null` and the SIM operator has an MCC value of `460`

(3) Network operator and SIM operator are both `null` and device's CSC XML file has the first `MCCMNC` tag starting with 460 (while ignoring values of `001` and `999`)

(4) Network operator and SIM operator are both `null` and device's CSC XML file either does not exist or does not have a `MCCMNC` tag and the system property key named `ro.csc.countryiso_code` contains a value of `CN`

Samsung `kperfmon` Local App Usage Data Exposure

Vulnerability: `kperfmon` kernel module leaks app usage data via the `/proc/kperfmon` virtual file to which third-party apps have read access on Samsung smartphones

Versions Impacted: Android 10 – 14

```
a25x:/ $ cat /proc/modules | grep kperfmon
kperfmon 28672 0 - Live 0x0000000000000000
```

Software Responsible: `kperfmon` kernel module

Root Cause: File permissions make the `/proc/kperfmon` virtual file globally readable

```
a25x:/ $ ls -laZ  /proc/kperfmon
-rw-rw-r-- 1 root root u:object_r:proc_perf:s0  0 2024-07-10 17:23 /proc/kperfmon
```

Third-party apps can read files with an SELinux context of `u:object_r:proc_perf:s0`

Source code for the `kperfmon` kernel module can be found here: official and unofficial

`kperfmon` Kernel Module Official Source Code

`SM-S906B_14_Opensource.zip` (downloaded on May 4, 2024) for the Samsung Galaxy S22+ (`SM-S906B`) smartphone

`drivers/kperfmon/kperfmon.c`

```c
#include "kperfmon.h"

#define PROC_NAME                       "kperfmon"



static int __init kperfmon_init(void)
{
        struct proc_dir_entry *entry;
        // char kperfmon_version[KPERFMON_VERSION_LENGTH] = {0, };


        CreateBuffer(&buffer, BUFFER_SIZE);


        if (!buffer.data) {
                pr_info("%s() - Error buffer allocation is failed!!!\n", __func__);
                return -ENOMEM;
        }

        entry = proc_create(PROC_NAME, 0664, NULL, &kperfmon_fops);

        ...
```

```c
#if LINUX_VERSION_CODE >= KERNEL_VERSION(5, 10, 0)
static const struct proc_ops kperfmon_fops = {
        .proc_open = kperfmon_open,
        .proc_read = kperfmon_read,
        .proc_write = kperfmon_write,
};
#else
static const struct file_operations kperfmon_fops = {
        .read = kperfmon_read,
        .write = kperfmon_write,
};
#endif
```

```c
module_init(kperfmon_init);
module_exit(kperfmon_exit);
```

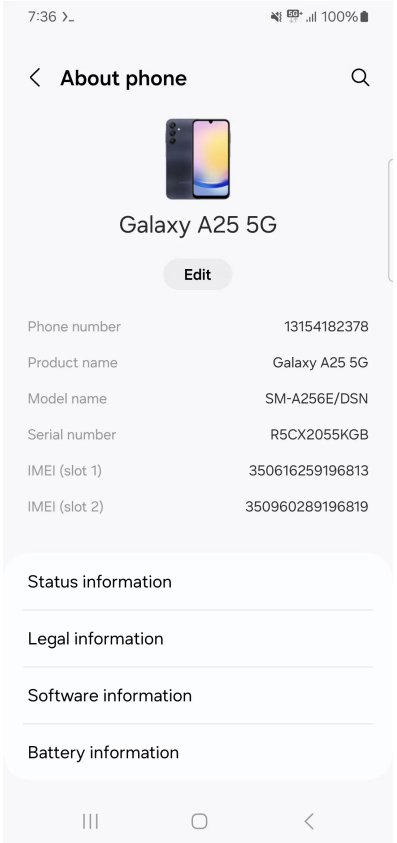
Sample of Impacted Samsung Devices – `kperfmon` App Usage Data Exposure

Model	Build Fingerprint	Build Date
Galaxy S22 Ultra	samsung/b0quew/b0q:14/UP1A.231005.007/S908U1UES4DXD1:user/release-keys	Mon Apr 1 17:33:33 KST 2024
Galaxy S21 Ultra 5G	samsung/p3quew/p3q:14/UP1A.231005.007/G998U1UESAFXD1:user/release-keys	Mon Apr 1 18:21:57 KST 2024
Galaxy A25 5G	samsung/a25xdxx/a25x:14/UP1A.231005.007/A256EXXS2AXCC:user/release-keys	Wed Apr 10 06:13:45 KST 2024
Galaxy Z Fold5	samsung/q5qsqw/q5q:13/TP1A.220624.014/F946USQU1AWG4:user/release-keys	Fri Jul 7 11:21:08 KST 2023
Galaxy S10+	samsung/beyond2ltexx/beyond2:10/QP1A.190711.020/G975FXXS9DTK9:user/release-keys	Fri Nov 13 12:14:56 KST 2020

Check if your device is vulnerable:

```
% adb shell 'grep -e APPLAUNCH -e CPUTOP /proc/kperfmon'
[07-12 22:23:34.715 1  6474     0 (120)][LOG][CPUTOP] [surfaceflinger=2.16%] [com.tinder=1.70%] [system_server=1.67%]
[com.android.systemui=0.50%] [com.android.chrome=0.36%]
[07-12 22:24:23.148 1  1786     0 (141)][LOG][APPLAUNCH]
[com.ashleymadison.mobile],com.almlabs.ashleymadison.xgen.ui.splash.SplashActivity,SPLASH_SCREEN(1),66,51,73,690,COLD(7),speed-profile(8) [S]
...
```

`kperfmon` App Usage Data Exposure Exploitation Demo



Implicit Broadcast `Intent` Messages

An `Intent` object is an abstraction for a message that is sent within/between apps and can contain various data

Implicit `Intent` messages lack a concrete destination package *and* lack a destination app component address

Implicit `Intent` messages contain only an `action` for loose-coupling (e.g., `android.intent.action.SEND`)

Implicit `Intent` messages can be received by any app that has dynamically registered for the `action` when the implicit `Intent` message is sent without a requisite access permission (which can be specified by the `Intent` sender)

```
Intent implicit_intent = new Intent("hey_everybody");
implicit_intent.putExtra("secret", saucy_secret);
sendBroadcast(implicit_intent);
```

Can be received by any app that dynamically registers for the `hey_everbody` action

```
Intent explicit_intent = new Intent("hey_you");
explicit_intent.putExtra("secret", saucy_secret);
explicit_intent.setPackage("s.o.m.e.a.p.p");
sendBroadcast(explicit_intent, "some.permission");
```

Can only be received by broadcast receiver components within an app with a package name of `s.o.m.e.a.p.p` app that register for the `hey_you` action

Qualcomm Workload Classifier App Usage Data Exposure

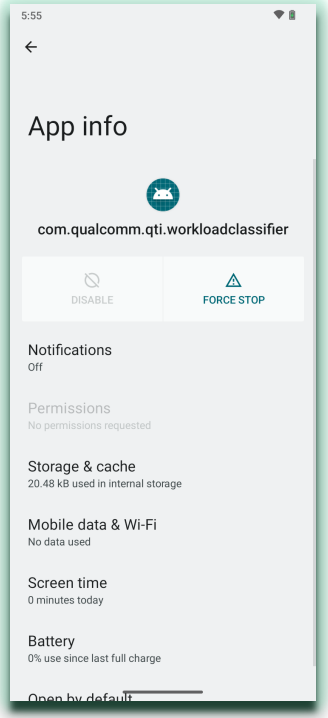
Vulnerability: A pre-installed Qualcomm app broadcasts the package name of each user-started app in an implicit broadcast `Intent` message with an action of `com.qualcomm.qti.workloadclassifier.APP_LAUNCH` when the app is not already executing

Versions Impacted: Android 12 - 14

Software Responsible: `com.qualcomm.qti.workloadclassifier` pre-installed app

Multiple Android vendor devices are impacted (OnePlus, Xiaomi, Nokia, etc.)

The implicit `Intent` messages can be changed to explicit `Intent` messages since the only consumer of the `com.qualcomm.qti.workloadclassifier.APP_LAUNCH` broadcast `Intent` messages is the `com.qualcomm.qti.workloadclassifier` pre-installed app itself



Receiving Implicit `Intent` Messages from the Qualcomm Workload Classifier App

```
registerReceiver(new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent == null) {
            Log.d("qworkload", "null intent received");
            return;
        }
        if ("com.qualcomm.qti.workloadclassifier.APP_LAUNCH".equals(intent.getAction()))
            Log.d("qworkload", "started_package_name=" + intent.getStringExtra("PKG_NAME"));
    }
}, new IntentFilter("com.qualcomm.qti.workloadclassifier.APP_LAUNCH"));
```

Check if your device is vulnerable:

```
% adb logcat qworkload:V -s
D qworkload: started_package_name=com.ashleymadison.mobile
D qworkload: started_package_name=cougar.dating.mature.sugar.older.women
D qworkload: started_package_name=com.instagram.android
D qworkload: started_package_name=jackpal.androidterm
D qworkload: started_package_name=com.facebook.katana
D qworkload: started_package_name=com.google.android.apps.safetyhub
D qworkload: started_package_name=good.on.you.for.reading.this
D qworkload: started_package_name=com.google.android.contacts
D qworkload: started_package_name=com.netflix.mediaclient
D qworkload: started_package_name=com.google.android.gm
```

Qualcomm Workload Classifier Impacted App Versions

Vulnerable Status	Version Code	Version Name	SHA-256 Message Digest
Yes	34	14	1073361ce49c363e2999b9797366f156c816aa56131eb8d56691cdb0bc84074d
Yes	33	13	3167904ef9311b6e7de051d6229bd96de071679fe0e912295eff658819fcb609
Yes (static verification)	32 Android 12L	12 Android 12L	f164d971efa2da5abdb289410592397c372bf94ae8a5a490d09a69382cc2b763
Yes	31	12	d19d395a457f9a9e8ed619fae5820c0012837d5dfba32d523fef96378b6519c3
No	30	11	c6b685f95f9827ed6a0d3f39d8d3f87111a2c18f32feeea9857304ce8338f2a0
No	29	10	01dd36a463420dc0287299013941e36d8aaaed8a2d41894bb32fff336aea4d6d

Nokia User Experience Program App Usage Data Exposure

Vulnerability: Exposes app usage data via an exported content provider when the user joins the User Experience Program on Nokia smartphones

Versions Impacted: Android 12 - 13

Software Responsible: Exported content provider app component in the `android` package (i.e., `system_server`)

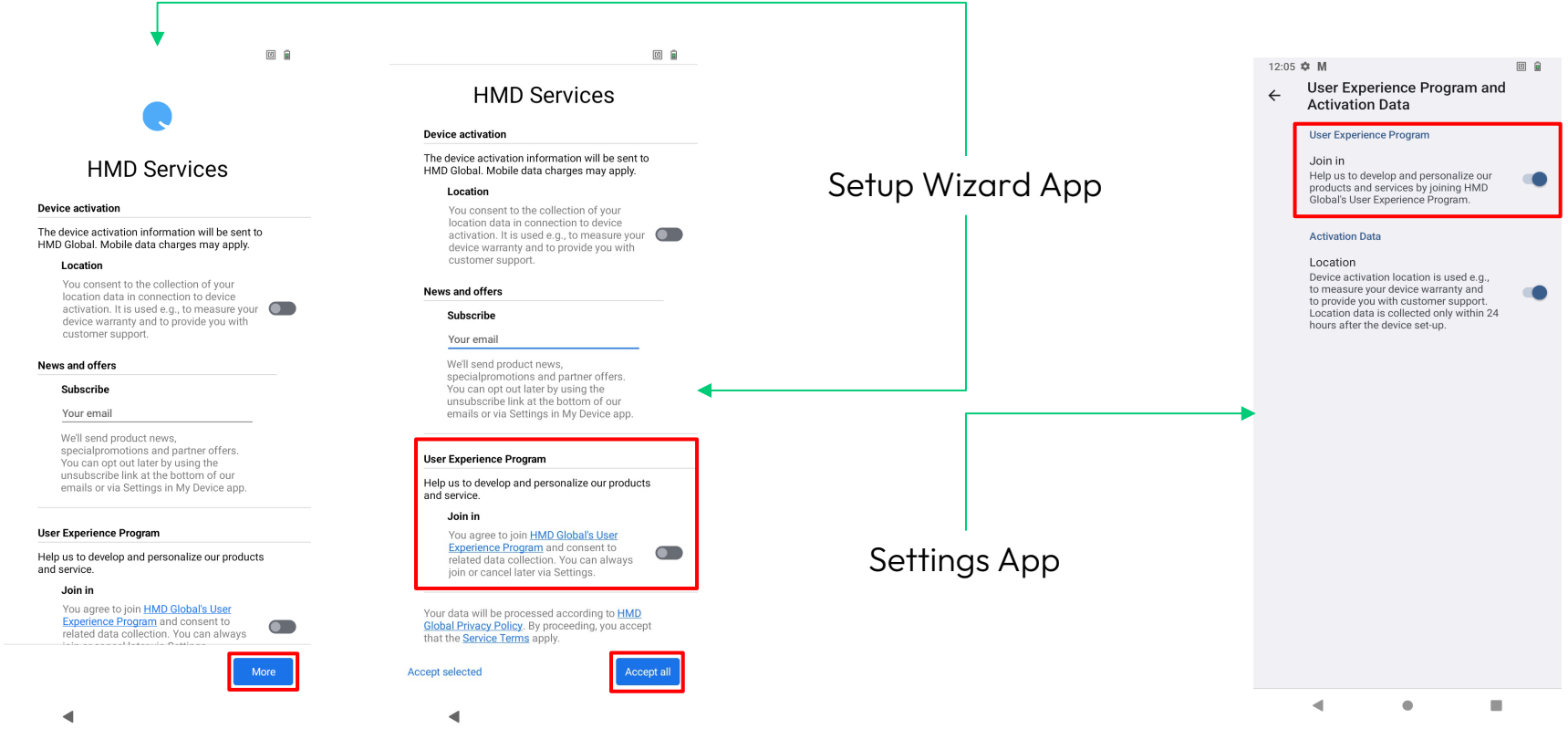
```
<provider android:authorities="com.hmddatacollect.com" android:enabled="true" android:exported="true"
android:name="com.android.server.hmddatacollect.HMDDataCollectContentProvider" android:singleUser="true"/>
```

Various system services within `system_server` have been modified to report events to the content provider

Requires the user opting-in during initial device setup or via the Settings apps which can be performed at anytime

The `persist.sys.setupwizard.join_user_experience_program` system property is set to a value of `true` when the User Experience Program is active

Nokia User Experience Program Activation



Nokia User Experience Program Table Contents

Exposed tables in the exported content provider component reveal app usage data and even contains the package names of certain background apps that the user may not be actively using

`AppStartTime` – Reveals package names of apps directly started by the user

```
Row: 11 id=44, packageName=jackpal.androidterm/.Term, StartUpTime=591, startType=1, timeStamp=H18
```

`RamConsump` – Reveals package names of apps and their RAM usage

```
Row: 37 id=2136, packageName=org.thoughtcrime.securesms, BgDuration=1313033, AvgBgMem=34288, MaxBgMem=68856,
AvgRunMem=44902, MaxRunMem=113288, RunWeight=121211635749.0, timeStamp=H20
```

`MobileData` – Reveals package names of apps and their mobile transmit/receive data

```
Row: 3 id=4, packageName=com.google.android.inputmethod.latin, RxData=22751700, TxData=279853, timeStamp=H22
```

`WifiData` – Reveals package names of apps and their Wi-Fi transmit/receive data

```
Row: 0 id=1, packageName=com.google.android.googlequicksearchbox, RxData=124961775, TxData=915809, timeStamp=H16
```

Sample of Impacted Nokia Devices - User Experience Program App Usage Data Exposure

Model	Build Fingerprint	Build Date
G50	Nokia/Punisher_00WW/PHR_sprout:13/TKQ1.220807.001/00WW_3_33E:user/release-keys	Tue Apr 2 14:13:50 UTC 2024
G310 5G	Nokia/Shadow_04US/SDT:13/TKQ1.221223.001/04US_1_13D:user/release-keys	Tue Sep 19 12:10:39 UTC 2023
C210	Nokia/Raven_00US/RVOA:13/TKQ1.230213.001/00US_1_160:user/release-keys	Thu Jan 25 15:50:00 UTC 2024
C12	Nokia/Nova_00M0/NVA:12/SP1A.210812.016/00WW_1_220:user/release-keys	Fri Mar 24 06:57:43 UTC 2023

Check if your device is vulnerable:

```
% adb shell content query --uri content://com.hmddatacollect.com/AppStartTime
```

Transsion

Not a trance DJ name but a Chinese smartphone manufacturer with three main brands: TECNO, Infinix, & Itel

It had the fourth largest (9.9%) global market share of smartphones in the first quarter of 2024 with 84.9% year-over-year growth according to IDC

Popular vendor in Africa, dubbed the "Smartphone King of Africa"

Has had security issues in the past: Triada, outdated software, and local arbitrary command injection

Currently being sued by Qualcomm for patent infringement



Remote Exposure of Installed Third-Party App List via HTTP

Vulnerability: Pre-installed system binary (`/system/bin/osi_bin`) transmits the user's third-party app list via HTTP to the `http://clog.geniex.com:9110/index` URL without user consent or awareness

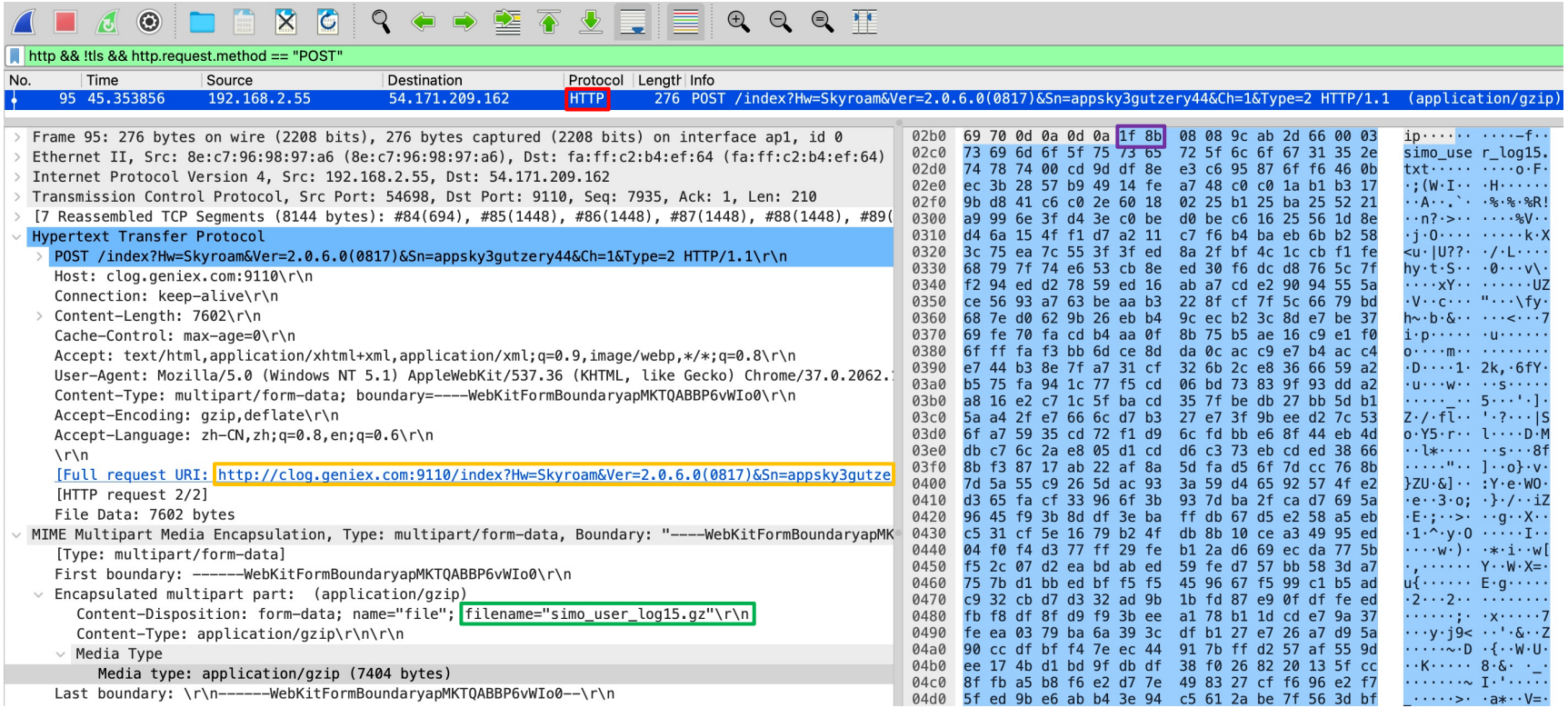
Impacted Devices: Infinix SMART 7 and TECNO Pova Neo 2 (purchased in Riyadh, Saudi Arabia)

Versions Impacted: Android 11 - 12

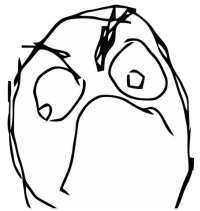
Snippet of a log file that was embedded as a GZIP in an HTTP POST request due to the GENIE-X software:

```
[default] 2024-04-29 15:31:51:492 file:val_os_linux.c function:val_system_block line:364 serial_number = 0xf04403d0, cmd = /system/bin/chmod 777
/mnt/traffic
[rsim] 2024-04-29 15:31:51:674 file:val_rsim_manage.c function:val_rsim_set_last_plmn line:3356 plmn:310260
[ui server] 2024-04-29 15:31:54:3216 file:stp_ui_socket_server.c function:stp_ui_socket_server_init line:310 bind ret = 0, NAME:@uisocket
[router] 2024-04-29 15:31:56:5393 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:230 app number:101
[router] 2024-04-29 15:31:56:5745 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10138]:[org.thoughtcrime.securesms]
[router] 2024-04-29 15:31:57:6859 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10187]:[com.topjohnwu.magisk]
[router] 2024-04-29 15:31:57:6911 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10128]:[com.samsung.android.spay]
[router] 2024-04-29 15:31:57:6928 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10121]:[com.coinbase.android]
[router] 2024-04-29 15:31:57:6930 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10168]:[com.transsion.soundrecorder]
[router] 2024-04-29 15:31:57:6933 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10146]:[com.transsion.trancare]
[router] 2024-04-29 15:31:57:7005 file:val_router_linux.c function:val_add_remove_get_appinfo_list_rsp line:242 [10123]:[com.spacegame.solitaire]
...
```

GZIP Embedded in HTTP POST Form Data (GENIE-X)



Vulnerabilities in Simo's virtual SIM (vSIM) Software (2021)



CVE-2021-41848 – Local `root` privilege escalation vulnerability that allows any local app that possesses the `android.permission.WRITE_EXTERNAL_STORAGE` permission to provide a forged software update that exposes arbitrary shell script execution with `root` privileges, and can optionally provide an arbitrary ARM binary that executes with `root` privileges on system startup due to relying on a hard-coded AES key (in the binary itself) for verification of the software update

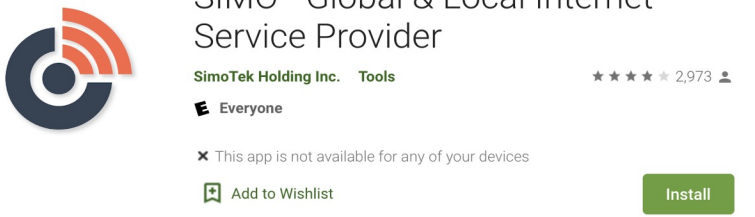
CVE-2021-41849 - The `com.skyroam.silverhelper` pre-installed app leaks the device's IMEI values to system properties which are accessible to third-party apps

CVE-2021-41850 – The `/system/bin/osi_bin` system binary transmits the list of all installed third-party apps and one of the IMEI values in an HTTP POST request to the `http://log.skyroam.com.cn:9110/index` URL

Time for a rebrand…?

	Simo	GENIE-X
System Binaries	`/system/bin/osi` `/system/bin/osi_bin`	`/system/bin/osi` `/system/bin/osi_bin`
Pre-Installed App	`com.skyroam.silverhelper`	`com.skyroam.silverhelper`
Logging URL	`http://log.skyroam.com.cn:9110/index`	`http://clog.geniex.com:9110/index`
User-Facing App	`com.skyroam.app`	`com.transtech.geniex`

Taken Down 💀



Source: https://play.google.com/store/apps/details?id=com.skyroam.app



Source: https://play.google.com/store/apps/details?id=com.transtech.geniex

GENIE-X Supported Devices and Supported Countries

Supported Devices

TECNO

CAMON 20/20 PRO/20 Premier 5G/30/30 5G/30 Pro 5G
SPARK 9/10/10 Pro/20C/20/20 Pro/20 Pro+
POP 6 Pro/6 Neo/7
POVA Neo3/5/5 Pro 5G

Infinix
The Future is Now!

NOTE 30/30 PRO/40/40 PRO/40S/40 5G/40Pro 5G/
HOT 20/20 PLAY/20S/30/30i/30 PLAY/40/40 PRO/40i
ZERO 20/30
SMART 8 PLUS/8 PRO/
GT 20 Pro

Supported Countries



Nigeria
Coveraged



Kenya
Coming soon



Ghana
Coveraged



Pakistan
Coming soon

Source: https://www.geniex.com/

Sample of Transsion Models Exposing Foreground App

Model	Settings Namespace	Key Name	Build Fingerprint
Tecno Pova Neo 2	global	top_resume_package	TECNO/LG6n-OP/TECNO-LG6n:12/SP1A.210812.016/230201V1103:user/release-keys
Itel Vision 3	system	current_focused_app	Itel/F6321/itel-S661LP:11/RP1A.201005.001/GL-V134-20231121:user/release-keys
Itel Vision 3	secure	ITEL_AMS_StartProcessLocked	Itel/F6321/itel-S661LP:11/RP1A.201005.001/GL-V134-20231121:user/release-keys
Infinix Smart 7	global	top_resume_package	Infinix/X6515-OP/Infinix-X6515:12/SP1A.210812.016/240308V1471:user/release-keys
Infinix Hot 30i	system	current_focused_app	Infinix/X669D-GL/Infinix-X669D:12/SP1A.210812.016/GL-20240408V414:user/release-keys

Check if your device is vulnerable:

```
% adb shell 'settings get global top_resume_package; settings get system current_focused_app; settings get secure ITEL_AMS_StartProcessLocked'
```

Conclusion

Vendors can unintentionally expose sensitive data in various ways that can be used to track and profile the user

These leaks bypass restrictions imposed on third-party apps by the Android Framework

Third-party apps you download might not be as constrained as you think

Privileged pre-loaded software deserves increased security vetting

Questions?

Thanks for attending!

