black hat ASIA 2020

OCTOBER 1-2, 2020
BRIEFINGS

Hey Google, Activate Spyware!

When Google Assistant Uses a Vulnerability as a Feature

Erez Yalon
Director Security Research, Checkmarx
@ErezYalon

Introduction

CVE-2019-2234

FORBES

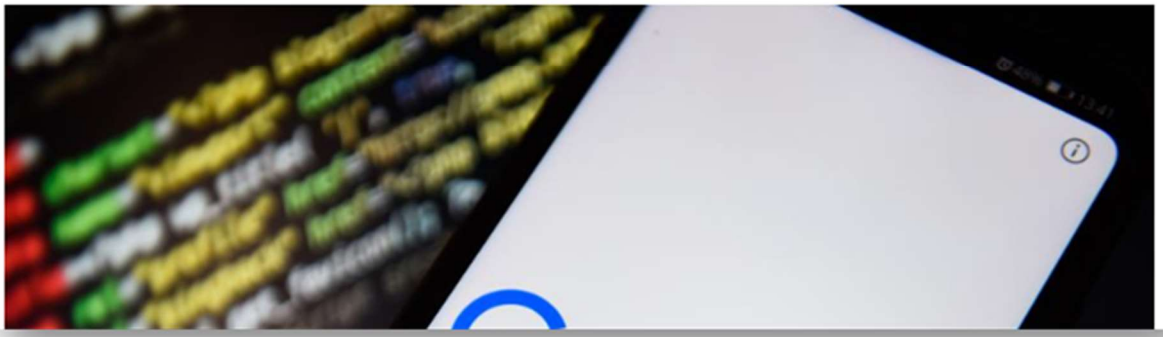
EDITORS' PICK | 1,951,872 views | Nov 19, 2019, 07:30am

Google Confirms Android Camera Security Threat: 'Hundreds Of Millions' Of Users Affected

Davey Winder Senior Contributor

Cybersecurity

I report and analyse breaking cybersecurity and privacy stories



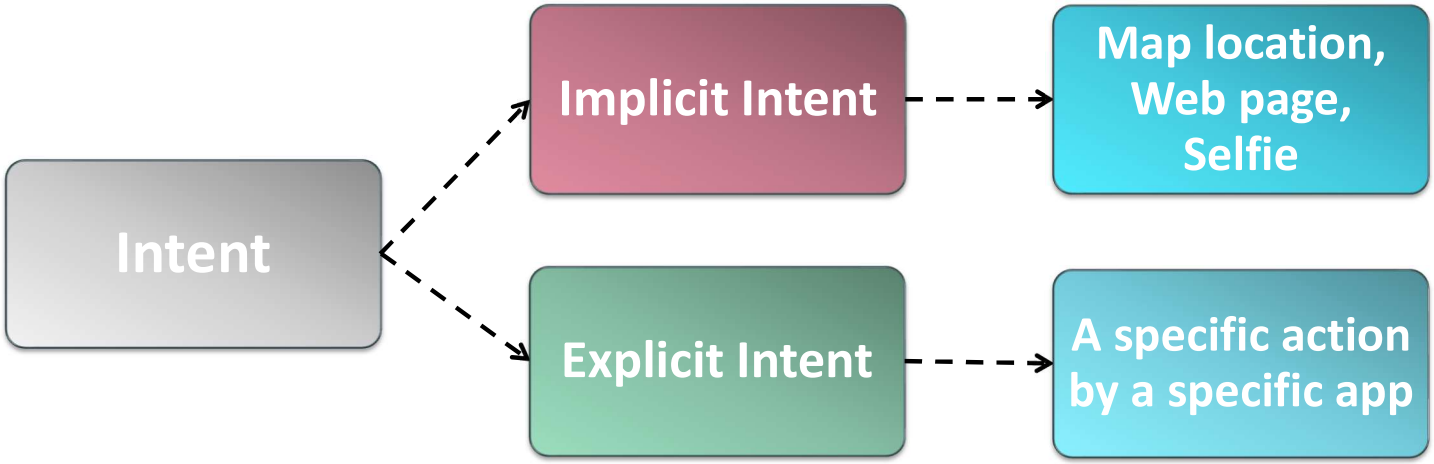
Agenda

- Android Terminology
- Step I – Let's Get Hacking
- Step II – Persistence
- Step III – Stealth
- Step IV – Always Want More
- Step V – Disclosure
- What We Learned

Let’s Take a Selfie…
By Using Google’s Personal Assistant

Terms – Intents 101

- Intents are a call for action



Terms – Permissions 101

Permissions overview (https://developer.android.com/guide/topics/permissions/overview)

"The purpose of a permission is to protect the privacy of an Android user. Android apps must request permission to access sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet). Depending on the feature, the system might grant the permission automatically or might prompt the user to approve the request."

"A central design point of the Android security architecture is that no app, by default, has permission to perform any operations that would adversely impact other apps, the operating system, or the user. This includes reading or writing the user's private data (such as contacts or emails), reading or writing another app's files, performing network access, keeping the device awake, and so on."

Terms – Permissions 101 (Cont.)

- Declaring the need for permission

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.example.snazzyapp">

    <uses-permission android:name="android.permission.CAMERA"/>
    <!-- other permissions go here -->

    <application ...>
        ...
    </application>
</manifest>
```

Terms – Permissions 101 (Cont.)

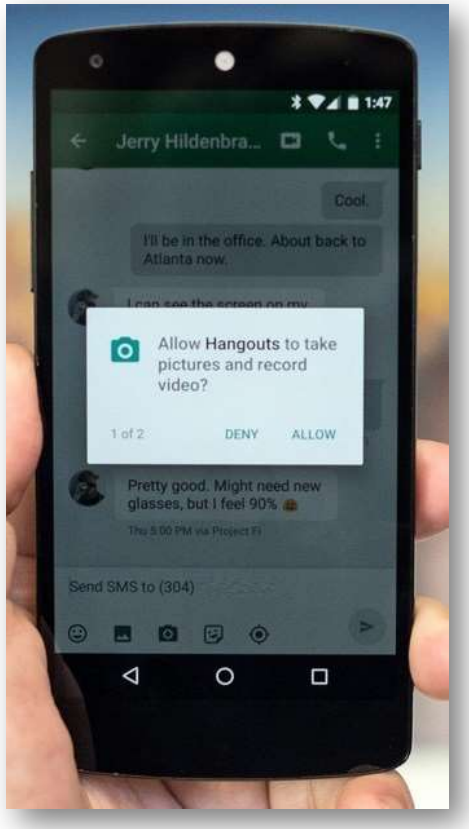
- Declaring the need for permission
- Check for permission

```
if (ContextCompat.checkSelfPermission(
        thisActivity, Manifest.permission.CAMERA)
        != PackageManager.PERMISSION_GRANTED) {
    // Permission is not granted
}
```

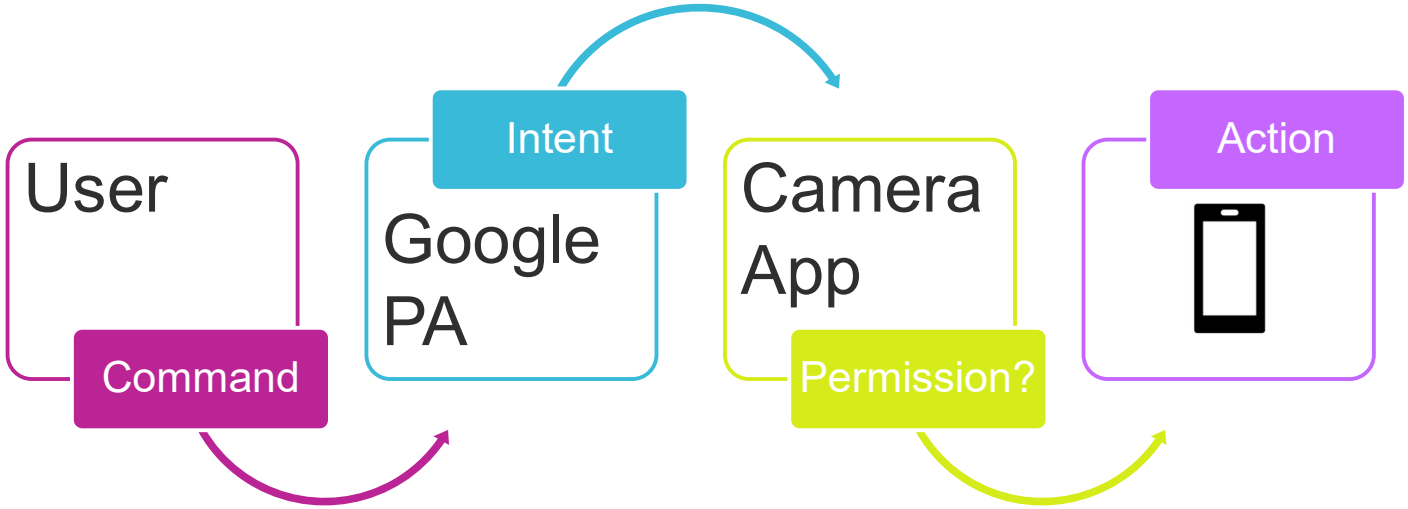
Terms – Permissions 101 (Cont.)

- Declaring the need for permission
- Check for permission
- Ask for permission

```
ActivityCompat.requestPermissions(thisActivity,
        new String[]{Manifest.permission.READ_CONTACTS},
        MY_PERMISSIONS_REQUEST_READ_CONTACTS);
```



The Selfie


User
Command
Intent
Google PA
Camera App
Permission?
Action

Step I – Let’s Get Hacking

The First Step of Finding a Hole in the System

Step I

- Finding a hole started by analyzing exported activities

```
<activity android:allowEmbedded=["true" | "false"]
          android:allowTaskReparenting=["true" | "false"]
          android:alwaysRetainTaskState=["true" | "false"]
          android:autoRemoveFromRecents=["true" | "false"]
          android:banner="drawable resource"
          android:clearTaskOnLaunch=["true" | "false"]
          android:colorMode=[ "hdr" | "wideColorGamut"]
          android:configChanges=["mcc", "mnc", "locale",
                                 "touchscreen", "keyboard", "keyboardHidden",
                                 "navigation", "screenLayout", "fontScale",
                                 "uiMode", "orientation", "density",
                                 "screenSize", "smallestScreenSize"]
          android:directBootAware=["true" | "false"]
          android:documentLaunchMode=["intoExisting" | "always" |
                                      "none" | "never"]
          android:enabled=["true" | "false"]
          android:excludeFromRecents=["true" | "false"]
          android:exported=["true" | "false"]
          android:finishOnTaskLaunch=["true" | "false"]
          android:hardwareAccelerated=["true" | "false"]
          android:icon="drawable resource"
          android:immersive=["true" | "false"]
          android:label="string resource"
          android:launchMode=["standard" | "singleTop" |
                              "singleTask" | "singleInstance"]
          android:lockTaskMode=["normal" | "never" |
```

Step I (Cont.)

- Google's Camera application provided many of those

```
com.google.android.apps.camera.legacy.app.activity.main.CameraActivity
com.android.camera.CameraLauncher
com.android.camera.CameraActivity
com.android.camera.activity.CaptureActivity
com.android.camera.VideoCamera
com.android.camera.CameraImageActivity
com.android.camera.CameraVideoShortcutActivity
com.android.camera.CameraDeepLinkActivity
com.android.camera.SecureCameraActivity
com.google.android.apps.camera.legacy.app.settings.CameraSettingsActivity
com.google.android.apps.camera.legacy.app.refocus.ViewerActivity
com.google.android.apps.camera.photobooth.activity.PhotoboothActivity
com.google.android.libraries.social.licenses.LicenseMenuActivity
```

Step I (Cont.)

- Google's Camera application provided many of those
- Which mapped into different classes

```
com.google.android.apps.camera.legacy.app.activity.main.CameraActivity
com.google.android.apps.camera.legacy.app.activity.CaptureActivity
com.google.android.apps.camera.legacy.app.activity.CameraImageActivity
com.google.android.apps.camera.legacy.app.activity.CameraDeepLinkActivity
com.google.android.apps.camera.legacy.app.activity.SecureCameraActivity
com.google.android.apps.camera.legacy.app.settings.CameraSettingsActivity
com.google.android.apps.camera.legacy.app.refocus.ViewerActivity
com.google.android.apps.camera.photobooth.activity.PhotoboothActivity
com.google.android.libraries.social.licenses.LicenseMenuActivity
```

Step I (Cont.)

- Google's Camera application provided many of those
- Which mapped into different classes
- And different actions

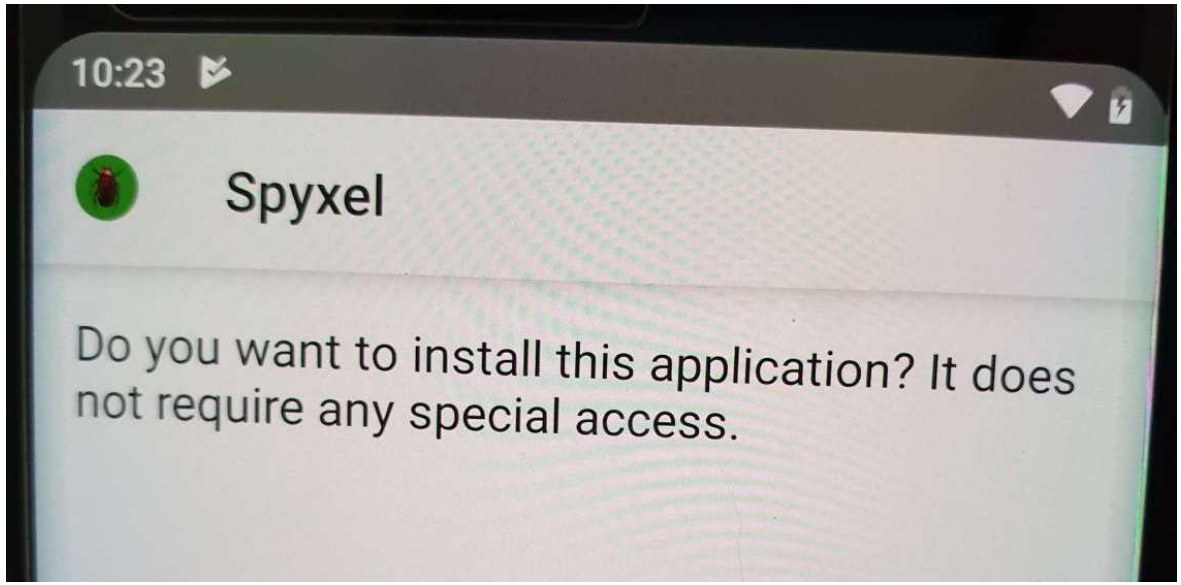
```
android.media.action.IMAGE_CAPTURE
android.media.action.IMAGE_CAPTURE_SECURE
android.media.action.STILL_IMAGE_CAMERA
android.media.action.STILL_IMAGE_CAMERA_SECURE
ndroid.media.action.VIDEO_CAPTURE
android.media.action.VIDEO_CAMERA
```

Step I (Cont.)

- The camera does not care who calls it into action?
- Invoking the **`android.media.action.VIDEO_CAMERA`** action starts the Google Camera and it immediately starts to record
- **`android.intent.extra.USE_FRONT_CAMERA`** allows the user to select the front camera (or back camera if absent)
- **`android.intent.extra.TIMER_DURATION_SECONDS`** allows the camera to have a timer before taking a photo (3 seconds minimum, hardcoded)

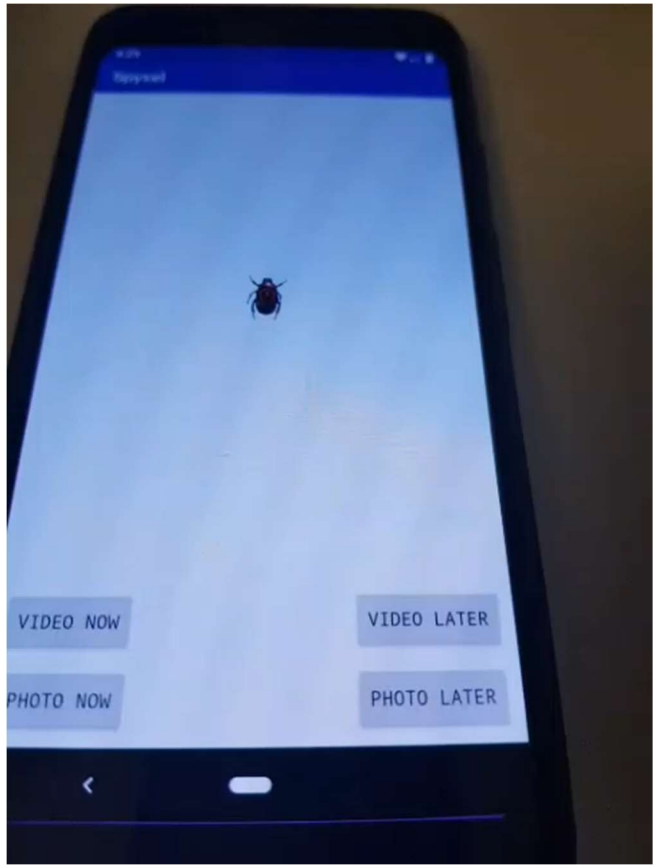
End of Step I – What We Have So Far

- A rogue application that requires no permissions



End of Step I – What We Have So Far

- A rogue application that requires no permissions
- **But can take pictures and videos at will**



What Do Hackers Really Want?

- Find an entrance ☑
- Establish persistence ☐
- Be secretive and stealthy ☐
- More! ☐

Step II – Persistence

Persistence

- The Camera App
  - Already persistent by design
  - Waiting for the right intent
- The Rogue App

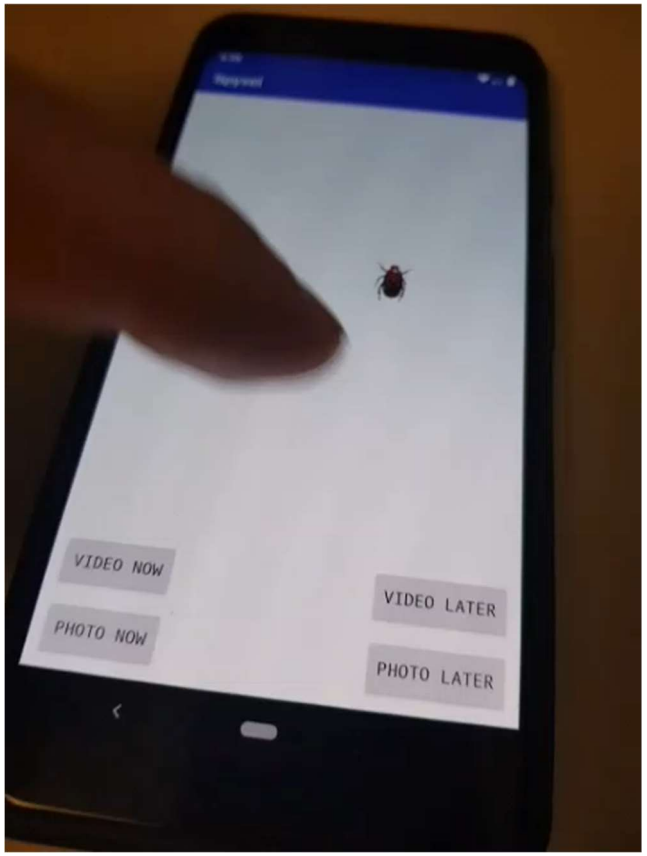
Android Developers > Docs > Guides

Guide to background processing

Every Android app has a main thread which is in charge of handling UI (including measuring and drawing views), coordinating user interactions, and receiving lifecycle events. If there is too much work happening on this thread, the app appears to hang or slow down, leading to an undesirable user experience. Any long-running computations and operations such as decoding a bitmap, accessing the disk, or performing network requests should be done on a separate background thread. In general, anything that takes more than a few milliseconds should be delegated to a background

Persistence (Cont.)

- When the app closes,
  the screen is off,
  and the phone is locked

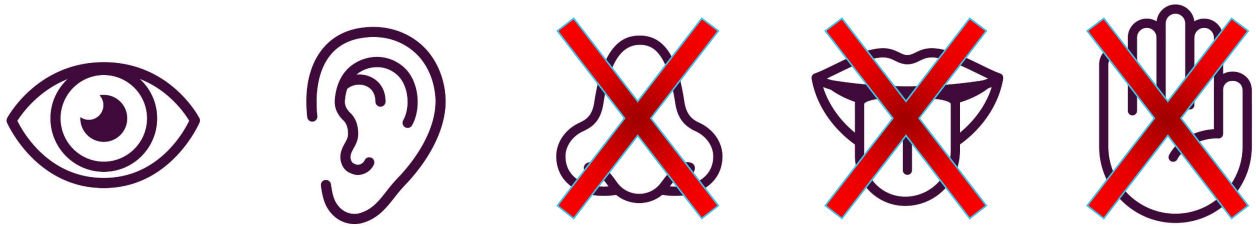


What Do Hackers Really Want?

- Find an entrance ☑
- Establish persistence ☑
- Be secretive and stealthy ☐
- More! ☐

Step III – Stealth

Stealth – Fooling the Senses



- Screen
- Media on device
- Shutter sound

Stealth – Shutter, Be Quiet!

- The phone cannot be muted without the right permission
- Using the function ***adjustStreamVolume*** with the flag **ADJUST_LOWER**, it is possible to lower the volume until it reaches silence.

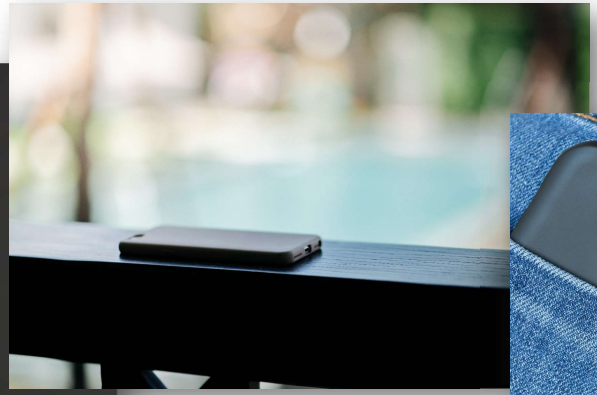
```
AudioManager am = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
for (int i=0;i < 30; i++) {
    am.adjustStreamVolume(AudioManager.STREAM_ALARM, AudioManager.ADJUST_LOWER, 0);
    am.adjustStreamVolume(AudioManager.STREAM_NOTIFICATION, AudioManager.ADJUST_LOWER, 0);
    am.adjustStreamVolume(AudioManager.STREAM_SYSTEM, AudioManager.ADJUST_LOWER, 0);
}
```

- This does not seem to be expected behavior.

CVE-2020-0089

Stealth – What About the Screen?

- Proximity sensor to the rescue!



- The rogue app will wait until the screen is covered

Stealth – What About the Media Files?

- The **storage** permission can help with that

Top App Permissions That Could Access User Information

Permission	What the Permission Does *"Allows the app to ..."*	# of Apps	% of Apps
Modify or delete the contents of your USB storage	... write to the USB storage. Allows the app to write to the SD card.	559,941	54%
Read phone status and identity	... access the phone features of the device. This permission allows the app to determine the phone number and device IDs, whether a call is active, and the remote number connected by a call.	361,616	35%
Precise location (GPS and network-based)	... get your precise location using the Global Positioning System (GPS) or network location sources such as cell towers and Wi-Fi. These location services must be turned on and available to your device for the app to use them. Apps may use this to determine where you are and may consume additional battery power.	246,750	24%
View Wi-Fi connections	... view information about Wi-Fi networking, such as whether Wi-Fi is enabled and name of connected Wi-Fi devices.	235,093	23%
Approximate location (network-based)	... get your approximate location. This location is derived by location services using network location sources such as cell towers and Wi-Fi. These location services must be turned on and available to your device for the app to use them. Apps may use this to determine approximately where you are.	216,770	21%

54%

Michelle Atkinson
Pew Research Center
November 10, 2015

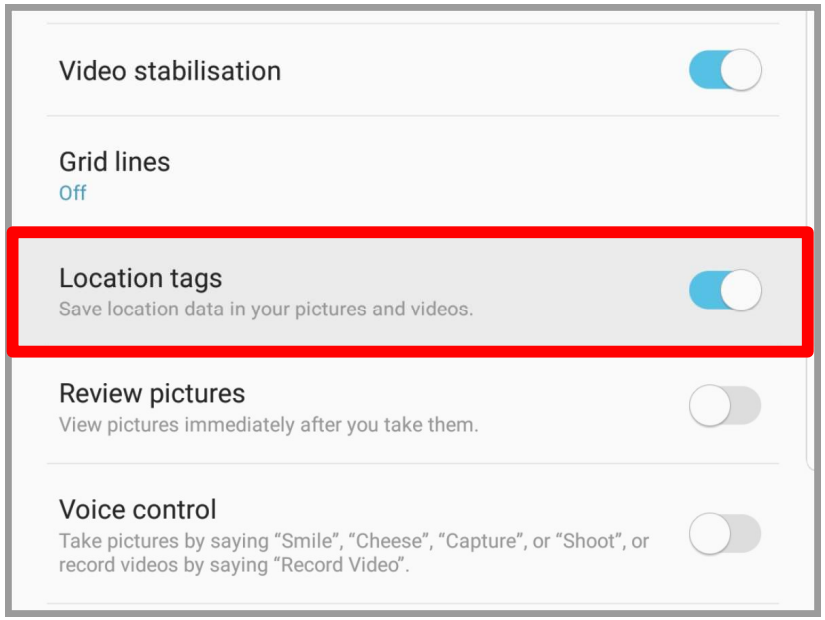
What Do Hackers Really Want?

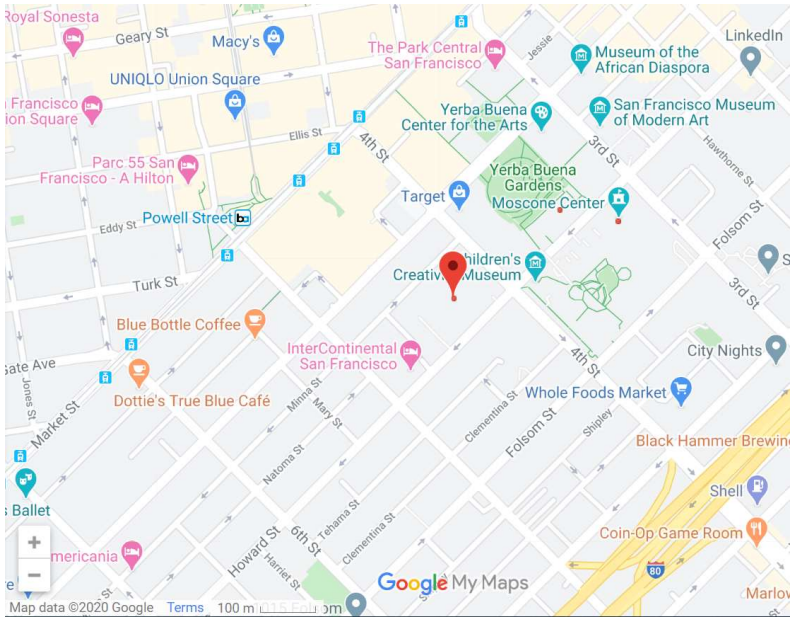
- Find an entrance ☑
- Establish persistence ☑
- Be secretive and stealthy ☑
- More! ☐

Step IV – We Always Want More

We Always Want More

- I can see you, but can I find you?

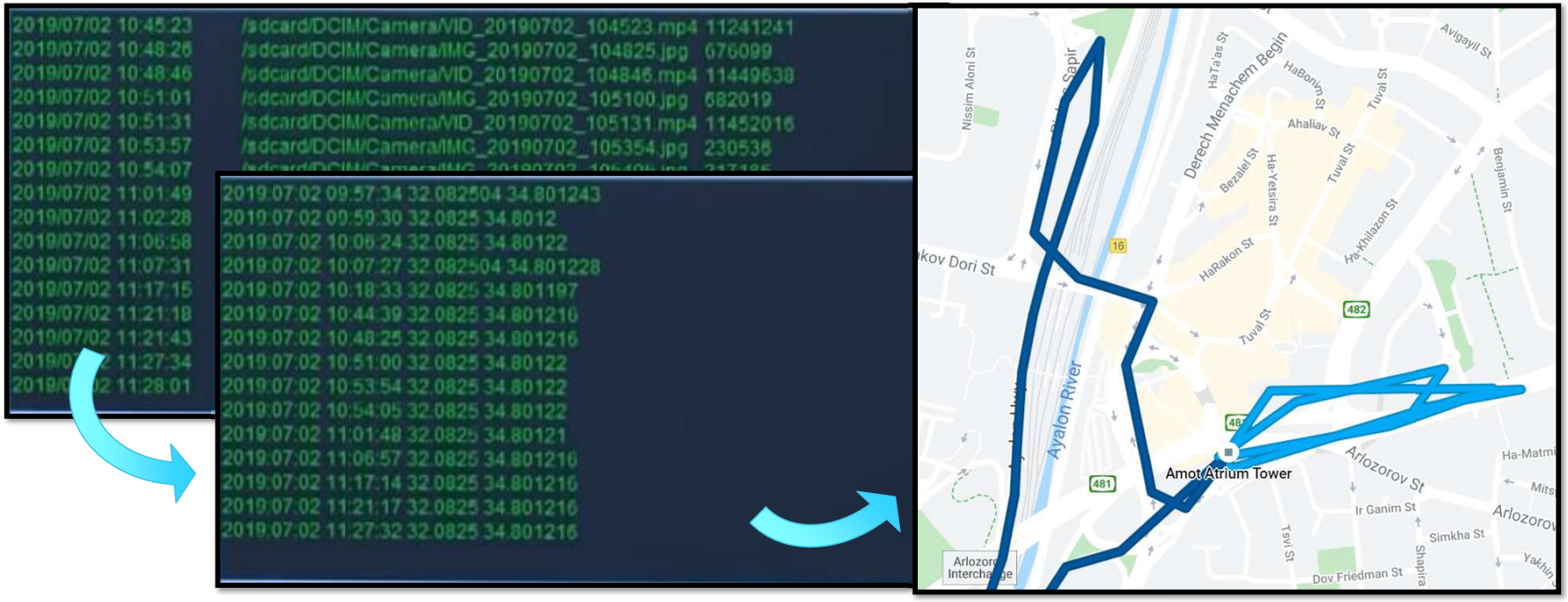


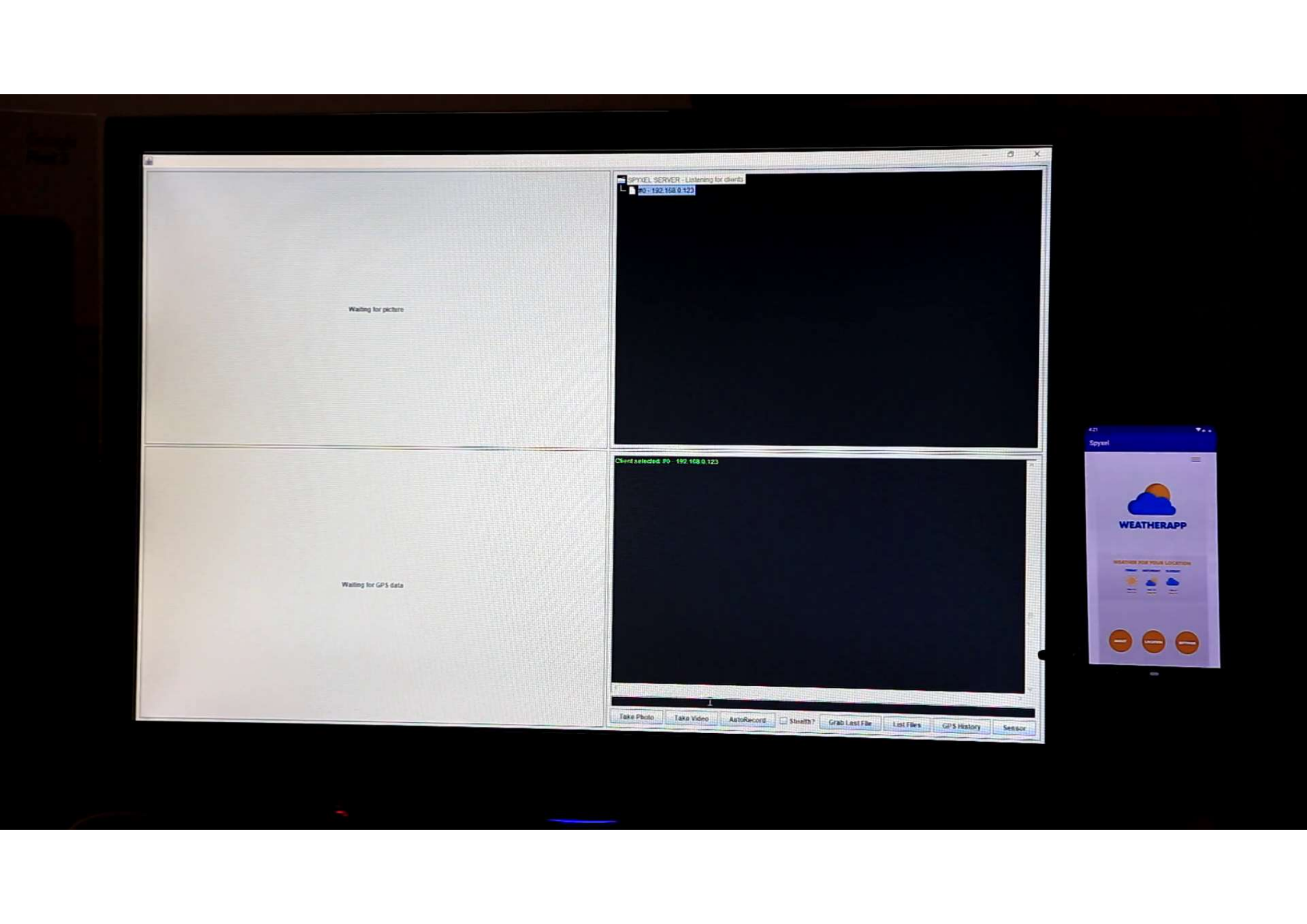


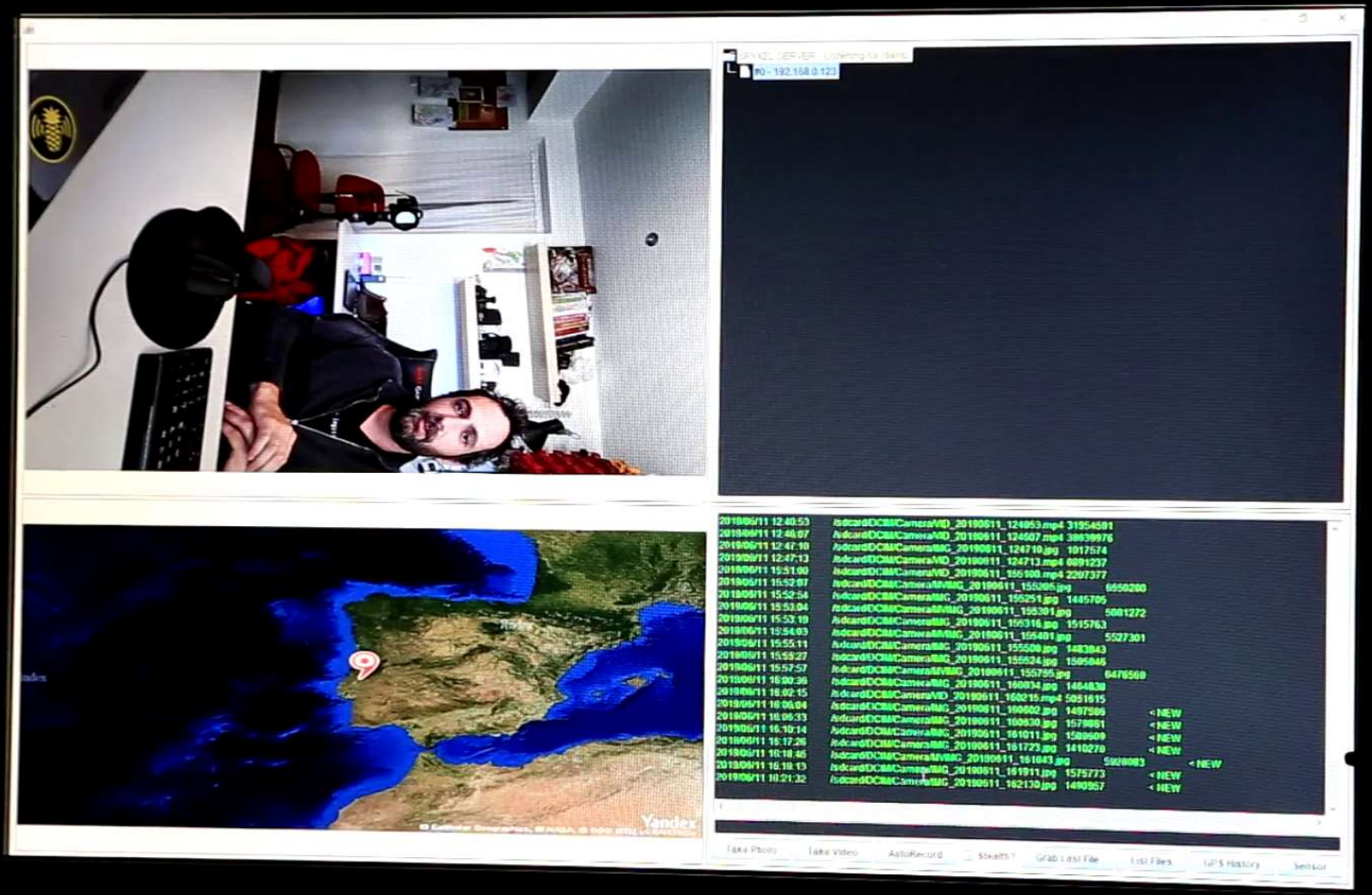
- Real-time geolocation

We Always Want More (Cont.)

- The targeted phone evolves into a tracking device



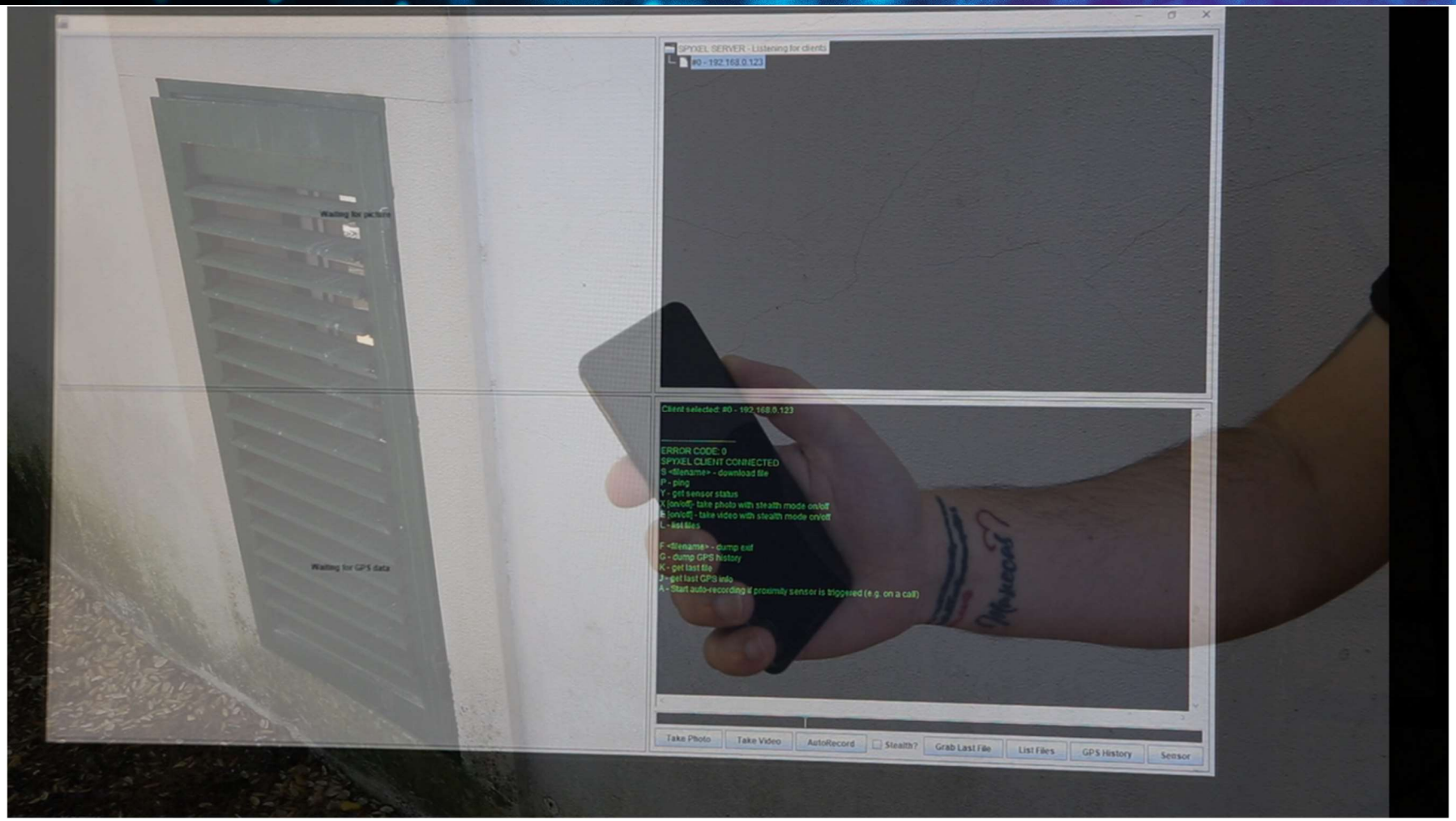

Waiting for picture
SPYXEL SERVER - Listening for clients
#0 - 192.168.0.123
Waiting for GPS data
Client selected: #0 - 192.168.0.123
Take Photo
Take Video
AutoRecord
Stealth?
Grab Last File
List Files
GPS History
Sensor
Spyxel
WEATHERAPP


#0 - 192.168.0.123
2019/06/11 12:40:53 /sdcard/DCIM/Camera/VID_20190611_124053.mp4 31954501
2019/06/11 12:46:07 /sdcard/DCIM/Camera/VID_20190611_124607.mp4 38639976
2019/06/11 12:47:10 /sdcard/DCIM/Camera/IMG_20190611_124710.jpg 1017574
2019/06/11 12:47:13 /sdcard/DCIM/Camera/VID_20190611_124713.mp4 6891237
2019/06/11 15:51:00 /sdcard/DCIM/Camera/VID_20190611_155100.mp4 2207377
2019/06/11 15:52:07 /sdcard/DCIM/Camera/IMG_20190611_155206.jpg 6550200
2019/06/11 15:52:54 /sdcard/DCIM/Camera/IMG_20190611_155251.jpg 1445705
2019/06/11 15:53:04 /sdcard/DCIM/Camera/IMG_20190611_155301.jpg 5081272
2019/06/11 15:53:19 /sdcard/DCIM/Camera/IMG_20190611_155316.jpg 1515763
2019/06/11 15:54:03 /sdcard/DCIM/Camera/IMG_20190611_155401.jpg 5527301
2019/06/11 15:55:11 /sdcard/DCIM/Camera/IMG_20190611_155500.jpg 1483843
2019/06/11 15:55:27 /sdcard/DCIM/Camera/IMG_20190611_155524.jpg 1505846
2019/06/11 15:57:57 /sdcard/DCIM/Camera/IMG_20190611_155755.jpg 6476580
2019/06/11 16:00:36 /sdcard/DCIM/Camera/IMG_20190611_160034.jpg 1464830
2019/06/11 16:02:15 /sdcard/DCIM/Camera/VID_20190611_160215.mp4 5051615
2019/06/11 16:06:04 /sdcard/DCIM/Camera/IMG_20190611_160602.jpg 1497586 < NEW
2019/06/11 16:06:33 /sdcard/DCIM/Camera/IMG_20190611_160630.jpg 1578851 < NEW
2019/06/11 16:10:14 /sdcard/DCIM/Camera/IMG_20190611_161011.jpg 1589609 < NEW
2019/06/11 16:17:26 /sdcard/DCIM/Camera/IMG_20190611_161723.jpg 1410270 < NEW
2019/06/11 16:18:46 /sdcard/DCIM/Camera/IMG_20190611_161843.jpg 5998083 < NEW
2019/06/11 16:19:13 /sdcard/DCIM/Camera/IMG_20190611_161911.jpg 1575773 < NEW
2019/06/11 16:21:32 /sdcard/DCIM/Camera/IMG_20190611_162130.jpg 1490957 < NEW
Take Photo
Take Video
AutoRecord
Stealth?
Grab Last File
List Files
GPS History
Sensor
Yandex

4:22

But Wait, There's More

- One last proximity sensor scenario
- Your private calls are not so private anymore


SPYXEL SERVER - Listening for clients
#0 - 192.168.0.123
Waiting for pictures
Waiting for GPS data
Client selected: #0 - 192.168.0.123
ERROR CODE: 0
SPYXEL CLIENT CONNECTED
S <filename> - download file
P - ping
Y - get sensor status
X [on/off] - take photo with stealth mode on/off
E [on/off] - take video with stealth mode on/off
L - list files
F <filename> - dump exif
G - dump GPS history
K - get last file
J - get last GPS info
A - Start auto-recording if proximity sensor is triggered (e.g. on a call)
Take Photo
Take Video
AutoRecord
Stealth?
Grab Last File
List Files
GPS History
Sensor

What Do Hackers Really Want?

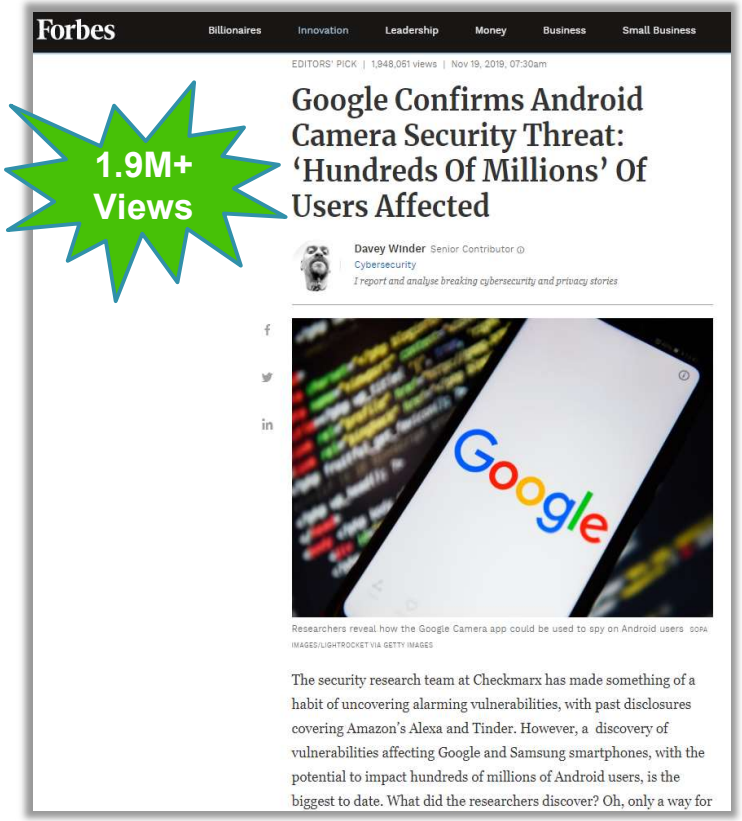
- Find an entrance ☑
- Establish persistence ☑
- Be secretive and stealthy ☑
- More! ☑

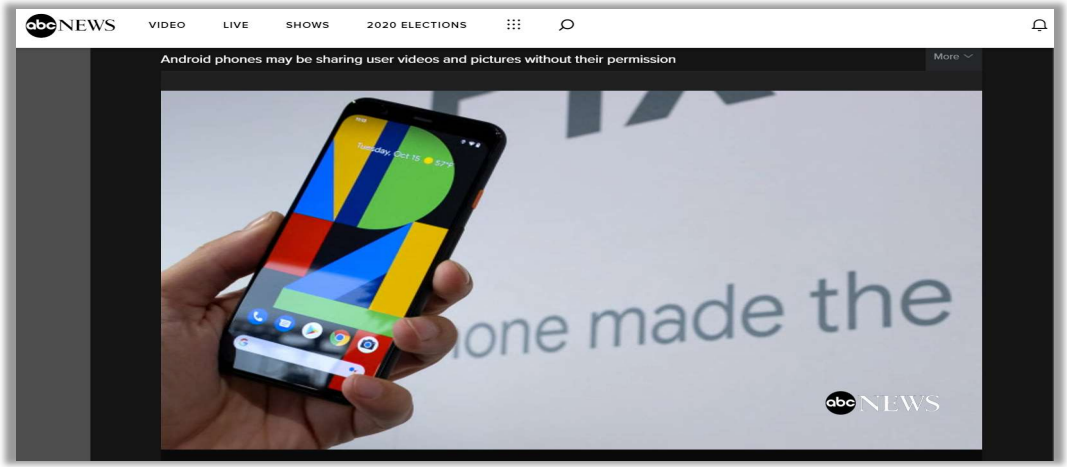
Step V – The Disclosure Process

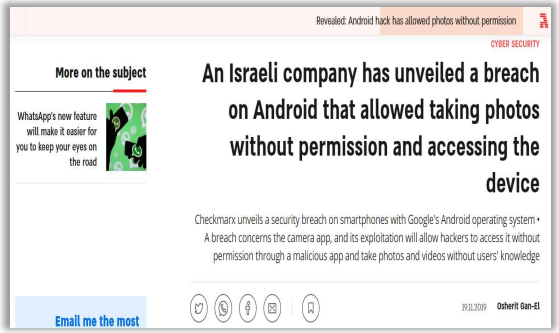
Disclosure Timeline (2019)

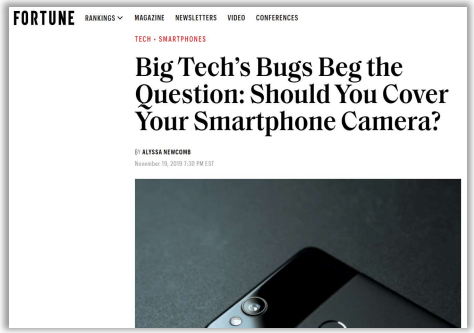
Jul 4	Submitted a full report with PoC malicious app and video to Android's security team at Google
Jul 13	Google set the severity of the finding as "moderate"
Jul 18	Sent further feedback to Google
Jul 23	Google raised the severity of the finding to "High"
Aug 1	Google issues CVE-2019-2234
Aug 1	Google confirms vulnerabilities may affect other Android smartphone vendors
Aug 18	Multiple vendors were contacted regarding the vulnerabilities
Aug 29	Samsung confirmed they are affected as well
Nov	Both Google and Samsung approved the publication

Media Traction









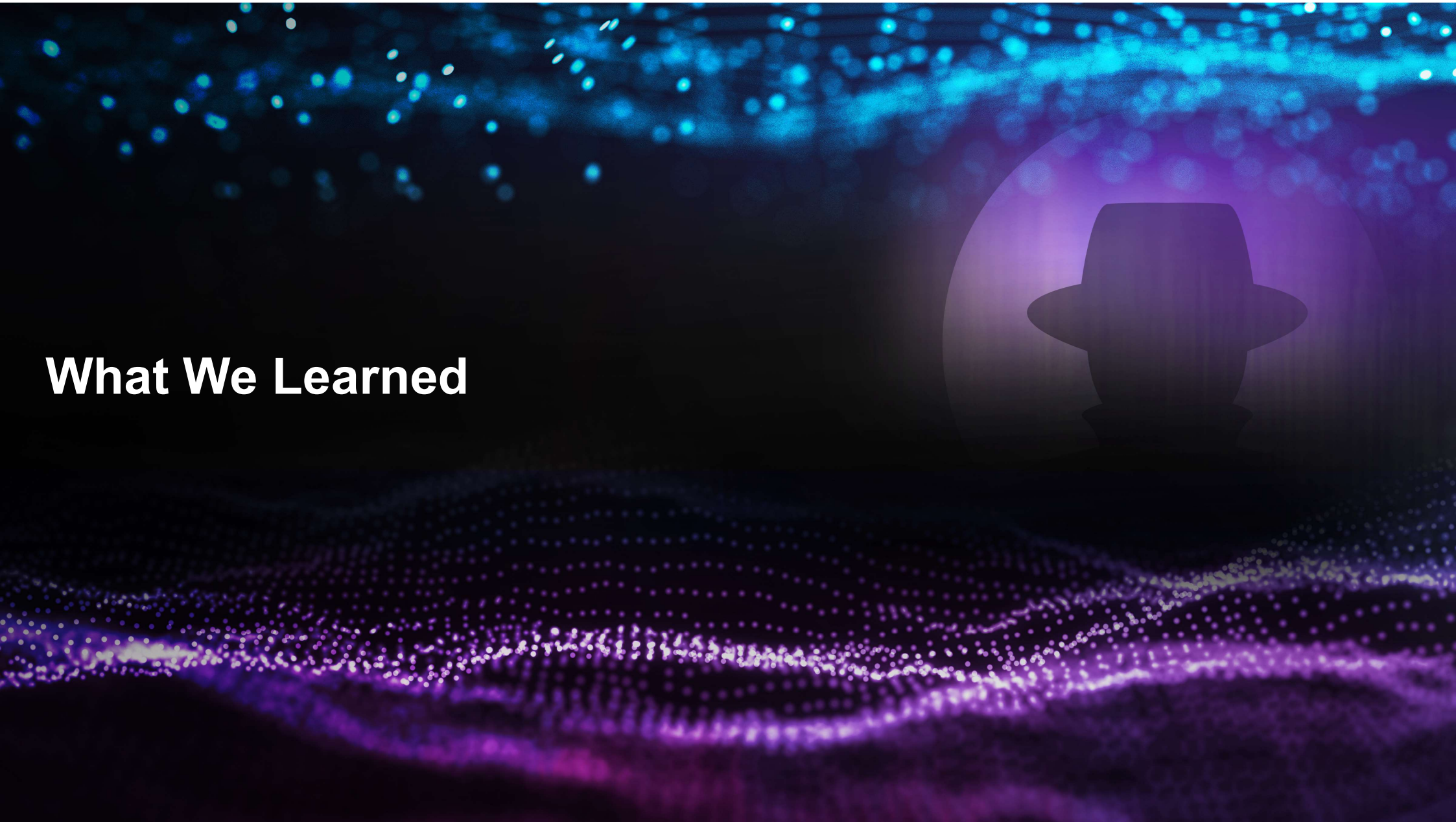
Disclosure

- Google's Response

"We appreciate Checkmarx bringing this to our attention and working with Google and Android partners to coordinate disclosure.

The issue was addressed on impacted Google devices via a Play Store update to the Google Camera Application in July 2019. A patch has also been made available to all partners."

What We Learned



What We Learned

Developers

- Keep an eye on your exported activities
- The Android permission-system is there for you

Researchers/Pentesters/Hackers(?)

- Finding a hole in security is only the first step
- Other vulnerabilities are your friends - chain them together

Bounty Hunters

- Sometimes it is worth haggling :)

Thank You!