# Motivation

*Mobile Security 2022*

Florian Draschbacher
florian.draschbacher@iaik.tugraz.at

Slides based on those by **Johannes Feichtner**

# Smartphones – History

## Once upon a time...

- PDA combined with a phone (starting in the late 90ies)
- IBM Simon (1994)
  - Touch Screen, Phone, Fax, E-Mail
- Nokia Communicator (1996)
  - Internet, Calendar, E-Mail, Business Apps
- Windows Mobile (2000)

Image: GR8Ausar / CC-BY-SA

# Early Smartphones

- Niche products for business use
  - Expensive
  - Impractical
  - Limited set of 3rd-party applications

- Very limited security
  - Hardware and OS often lacked basic security functionality
  - IBM Simon: No virtual memory
  - Windows Mobile: No file permissions, no real process isolation

# The Rising…

2007 / 2008:

- Apple iPhone, Android appeared
- New focus on user interface (multi touch screens)
- Business features quite limited
- Huge success in consumer area
- Many new ideas, concepts and applications

→ More recent: Also targeting the business area

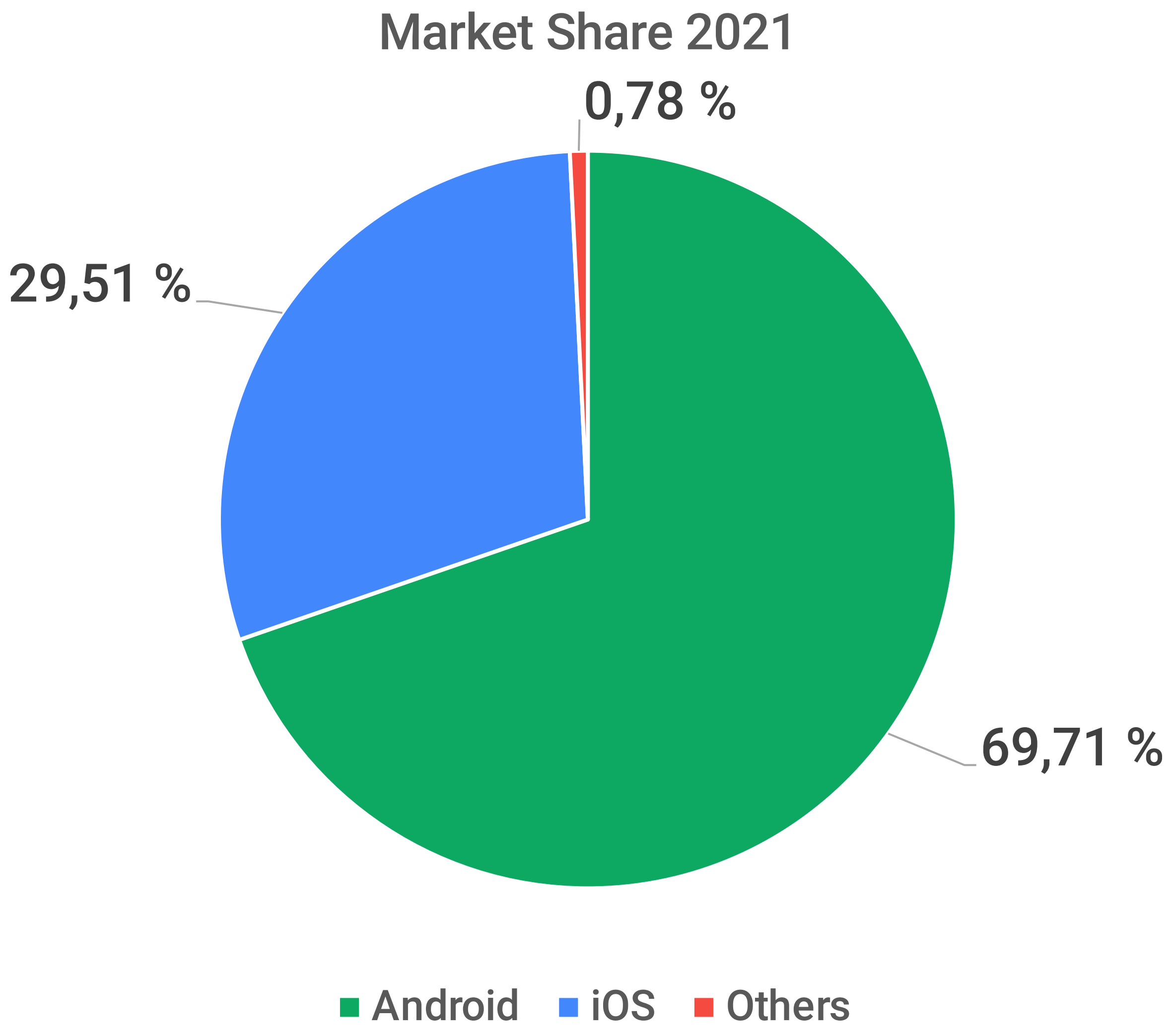
- – Container apps (Samsung Knox, Google for Work)
- – MDM Policies

Image: Google / CC-BY

# Today

- Android:
  - Developed by Google
  - Open-source
  - Linux kernel
  - Different devices, vendors

- iOS / iPadOS:
  - Developed by Apple
  - Closed-source (mostly)
  - Open-source XNU kernel
  - Closely related to watchOS, macOS



Source: statcounter.com

# Applications

- Social networks: Twitter, Facebook, Instagram, Snapchat, …
  - Contact data, Internet, Camera, Location (Network + GPS)

- Games: Online, multi-player, huge market
  - Internet, advertisements (Internet, Location, IDs), accelerometers, gyroscope

- Navigation: Hiking, biking, cities, maritime, aviation
  - Your location, „where are my friends?“

# Applications

- Business: e-mail, calendar, container apps
  - Access to critical data, e-mails (!), company infrastructure
- Augmented reality: Navigation, games, peaks, …
  - Camera, Compass, Orientation, Internet
- Banking: Online Banking, Mobile Payment
  - PIN / TAN entry, access to Secure Elements
  - Two-factor authentication tends to happen on one device…

# Applications

- Security software: Virus scanners, remote wipe / access
  - Access everything, sometimes rooted (Android) or with jail-break (iOS)
- Shopping: Amazon, Willhaben, AliExpress
  - Account information, credit card data, purchase history
- Personal data manager: Google Keep, Photos → Cloud, Password Managers
  - Handling sensitive data
  - User does not know / understand what happens behind the scenes

# Applications – Outlook

- Tablet & smartphone market share still growing
- More sophisticated apps
- Digital wallets
  - Covid Green Pass
  - Driving License [1]
- Mobile stolen → Identity stolen?!

Image: The Oxygen Team / GNU GPL

[1]: Source: oesterreich.gv.at

# Threats?

Now you know the possibilities but…

…what are the threats?

# Smartphone - Threats

- Companies know much about PC security
  → *Can we apply this mobile devices / smartphones?*

**Only in a very limited way!**

→ ***Smartphones have unique properties which raise new threats!***

> “Typical security defenses fail in mobile settings because they protect boundaries rather than information. Mobile users don’t respect traditional boundaries. The information itself must be protected.”

Source: Gartner

# Smartphone - Threats

- New technologies in combination with old ones
  - E.g. Linux as basis + key storage in hardware

- Mixed private / business use cases
  - How to separate these two spheres?
  - Limited administrative access to devices

- Legacy security strategies are ineffective
  - Innovation outpaces security practices

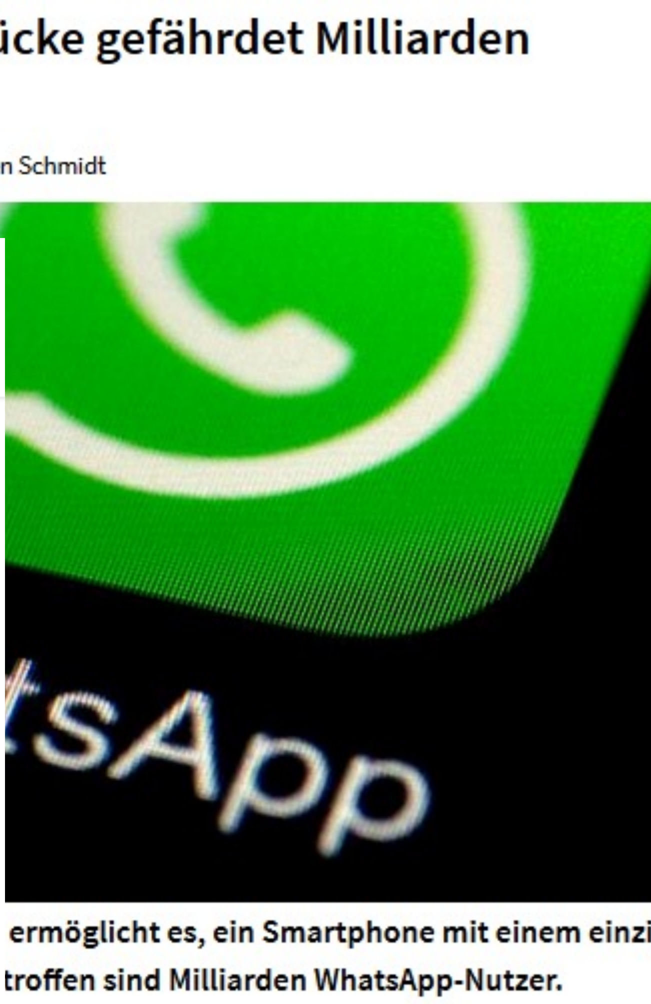
- Smartphones are every-day companions
  - Mobility poses risks

# New Mix of Technologies

- Ubiquituous Internet connection
  - UMTS / LTE, WiFi
- Telephone
  - SMS / MMS
  - Bluetooth
- Sensors
  - Microphone,
  - A-GPS,
  - Light Sensor,
  - Gyroscope,
  - …

## Kritische Sicherheitslücke gefährdet Milliarden WhatsApp-Nutzer

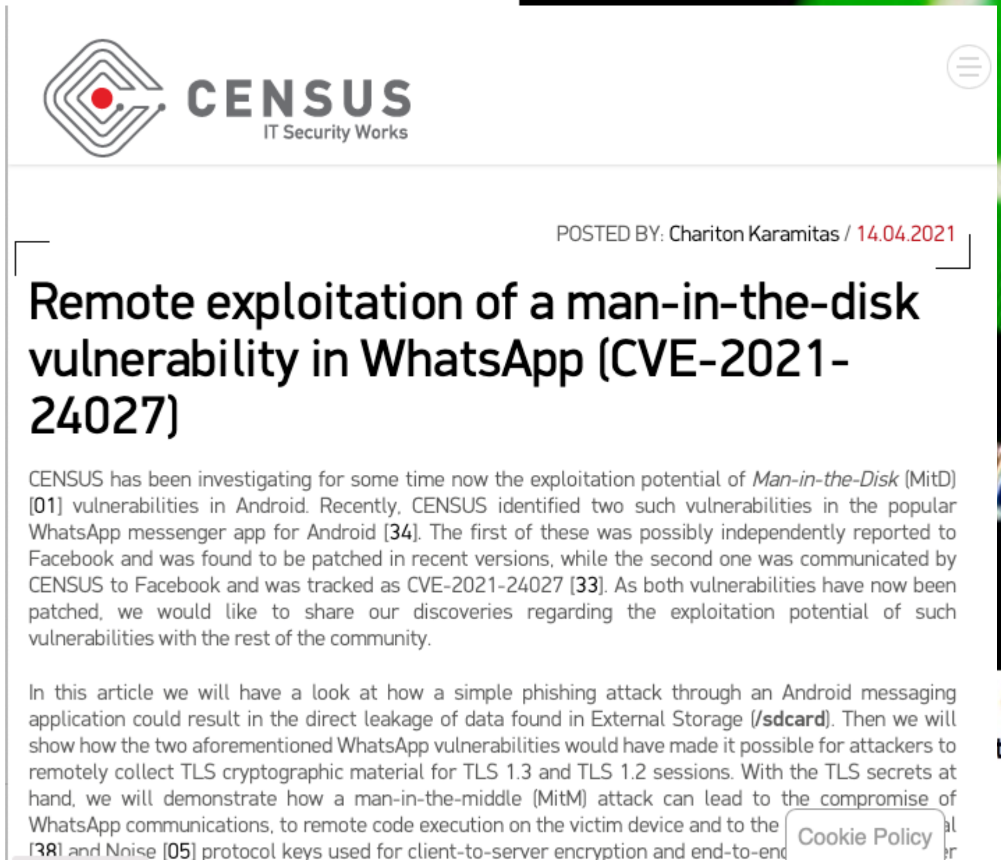
Alert! 10.10.2018 10:43 Uhr – Jürgen Schmidt



ermöglicht es, ein Smartphone mit einem einzi

troffen sind Milliarden WhatsApp-Nutzer.

Source: https://goo.gl/3mEYGf



POSTED BY: Chariton Karamitas / 14.04.2021

## Remote exploitation of a man-in-the-disk vulnerability in WhatsApp (CVE-2021-24027)

CENSUS has been investigating for some time now the exploitation potential of *Man-in-the-Disk* (MitD) [01] vulnerabilities in Android. Recently, CENSUS identified two such vulnerabilities in the popular WhatsApp messenger app for Android [34]. The first of these was possibly independently reported to Facebook and was found to be patched in recent versions, while the second one was communicated by CENSUS to Facebook and was tracked as CVE-2021-24027 [33]. As both vulnerabilities have now been patched, we would like to share our discoveries regarding the exploitation potential of such vulnerabilities with the rest of the community.

In this article we will have a look at how a simple phishing attack through an Android messaging application could result in the direct leakage of data found in External Storage (**/sdcard**). Then we will show how the two aforementioned WhatsApp vulnerabilities would have made it possible for attackers to remotely collect TLS cryptographic material for TLS 1.3 and TLS 1.2 sessions. With the TLS secrets at hand, we will demonstrate how a man-in-the-middle (MitM) attack can lead to the compromise of WhatsApp communications, to remote code execution on the victim device and to the al [38] and Noise [05] protocol keys used for client-to-server encryption and end-to-end r

Cookie Policy

Source: census-labs.com

# New Mix of Technologies

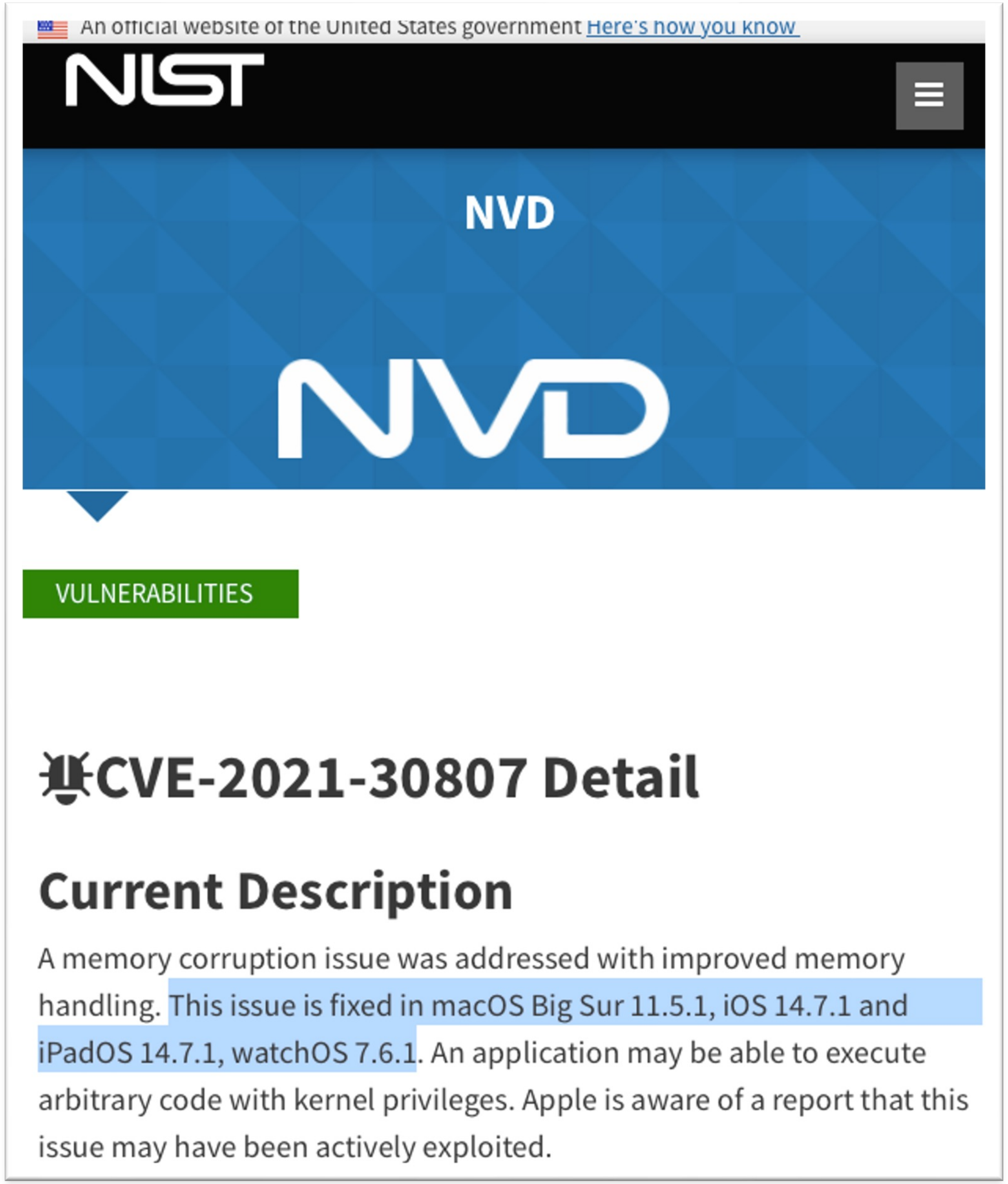
Shared OS & parts of it → shared security aspects!

- Often same attacks on the foundations
- Key Reinstallation Attack (KRACK) on WPA2
- OpenSSL

- iOS (XNU)

  → watchOS, tvOS

- Android (Linux)
  - ARM TrustZone
  - Vendor additions
  - ASLR bypass



An official website of the United States government Here's how you know

## CVE-2021-30807 Detail

### Current Description

A memory corruption issue was addressed with improved memory handling. This issue is fixed in macOS Big Sur 11.5.1, iOS 14.7.1 and iPadOS 14.7.1, watchOS 7.6.1. An application may be able to execute arbitrary code with kernel privileges. Apple is aware of a report that this issue may have been actively exploited.

Source: nvd.nist.gov

# Data & Sensors

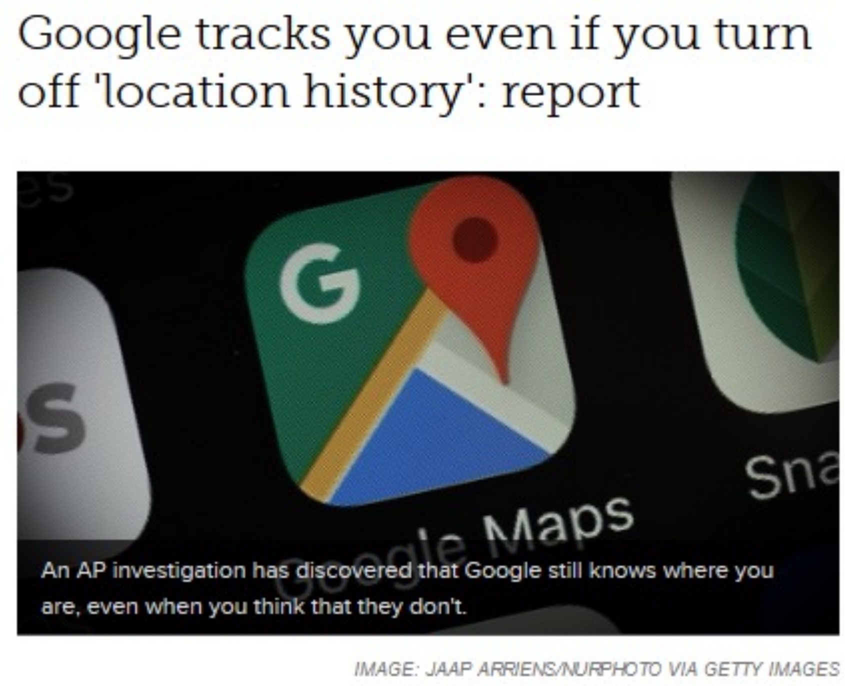
- „Data assets“
  - Private & business social network (mixed?)
  - Business data
    - E-mails, app data, access to infrastructure, e.g. VPN
  - Audio recordings, photos, videos
  - Passwords & Keys
    - WiFi passwords, Bank logins, …

***Stored in the cloud?***

# Data & Sensors

- Smartphone is taken everywhere
  - Collecting data even while not actively used
- Location
  - Network Cell ID (coarse)
  - GPS (fine)
    - Usually used with A-GPS for faster 3D fix
- Microphone, Motion Data, …
  - Ads may collect sensor data that leaks credit card info

    Source: Diamantaris et al., 2021



Google tracks you even if you turn off 'location history': report

An AP investigation has discovered that Google still knows where you are, even when you think that they don't.

IMAGE: JAAP ARRIENS/NURPHOTO VIA GETTY IMAGES

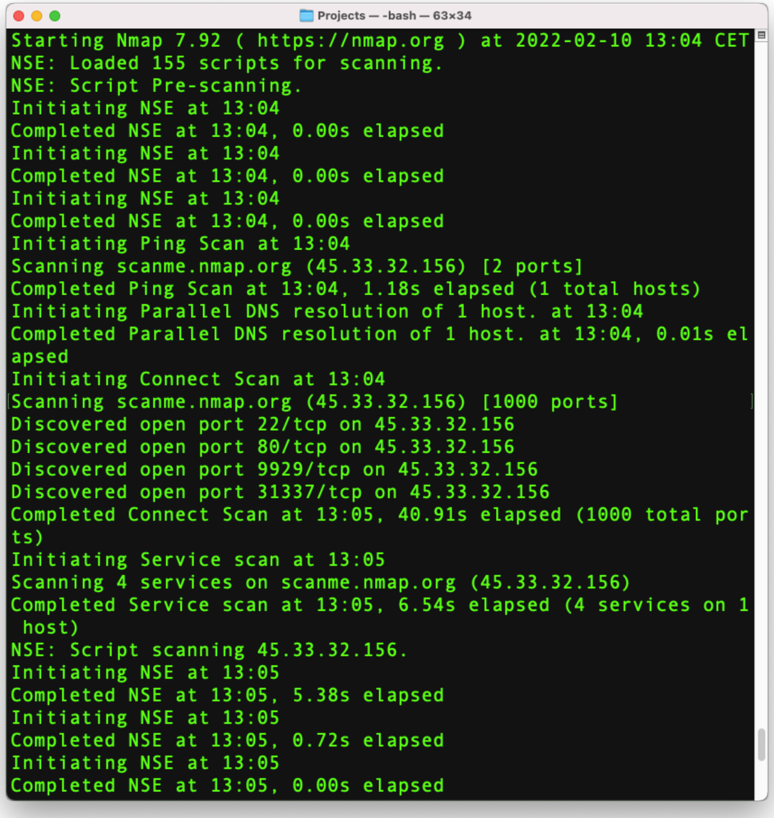
Source: mashable.com

# Mobility

- Install malware on smartphone on-the-fly
  - Steal it from a jacket, take it from a table, …

- WiFi Hotspots (old problems re-emerge)

- Use it for attacks
  - Spy with its microphone, camera
  - Do ARP Spoofing / MITM in WiFis
  - Scan networks
  - Open a rogue access point



```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-10 13:04 CET
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 13:04
Completed NSE at 13:04, 0.00s elapsed
Initiating NSE at 13:04
Completed NSE at 13:04, 0.00s elapsed
Initiating NSE at 13:04
Completed NSE at 13:04, 0.00s elapsed
Initiating Ping Scan at 13:04
Scanning scanme.nmap.org (45.33.32.156) [2 ports]
Completed Ping Scan at 13:04, 1.18s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:04
Completed Parallel DNS resolution of 1 host. at 13:04, 0.01s el
apsed
Initiating Connect Scan at 13:04
Scanning scanme.nmap.org (45.33.32.156) [1000 ports]
Discovered open port 22/tcp on 45.33.32.156
Discovered open port 80/tcp on 45.33.32.156
Discovered open port 9929/tcp on 45.33.32.156
Discovered open port 31337/tcp on 45.33.32.156
Completed Connect Scan at 13:05, 40.91s elapsed (1000 total por
ts)
Initiating Service scan at 13:05
Scanning 4 services on scanme.nmap.org (45.33.32.156)
Completed Service scan at 13:05, 6.54s elapsed (4 services on 1
 host)
NSE: Script scanning 45.33.32.156.
Initiating NSE at 13:05
Completed NSE at 13:05, 5.38s elapsed
Initiating NSE at 13:05
Completed NSE at 13:05, 0.72s elapsed
Initiating NSE at 13:05
Completed NSE at 13:05, 0.00s elapsed
```

# Business vs. Private Use

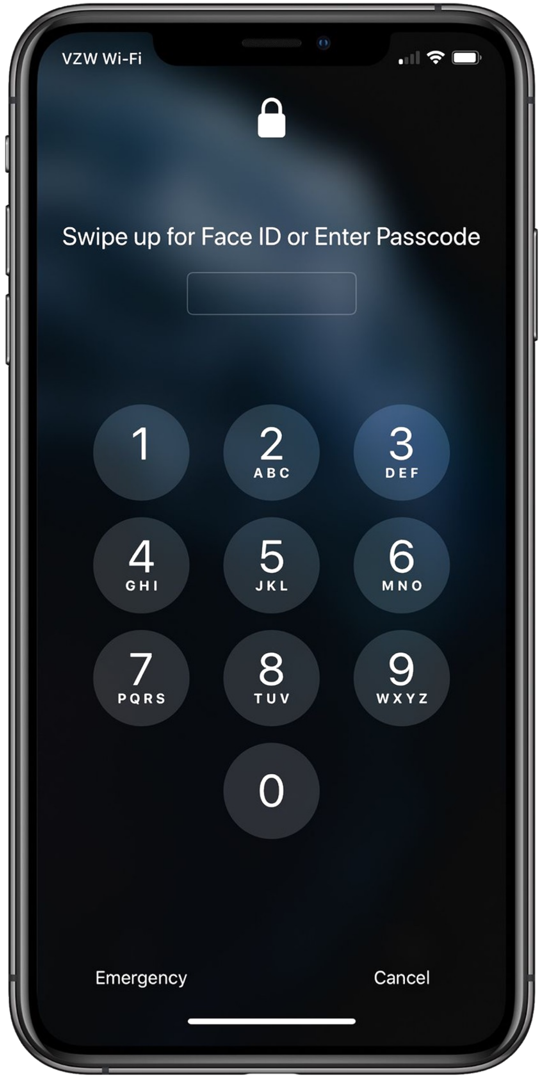
- Complete mixture of two areas
- Usually strict security policy for corporate apps
- No security policy for private apps on same device
  - Still effects on device‘s security
- BYOD – Bring your own device
  - Corporate apps on potentially insecure system

Pictures: Google / Apache 2.0

# Security vs. Usability

*Smart phones need to be easily approachable!*

- PIN codes, short passwords, screen unlock patterns
- Two-Factor-Authentication on one device
- Take pictures without unlocking the device



# Way out of the Dilemma: Risk Analysis

- Many threats & huge number of potential security issues

- Platform-specifics: encryption, PINs, cloud, permissions, applications, …

→ Can we fight everything in advance? What about new attacks / threats?

Define your assets: *What needs to be protected, what is important, …*

Define your threats: *Theft? Simple attacks? Sophisticated attacks?*

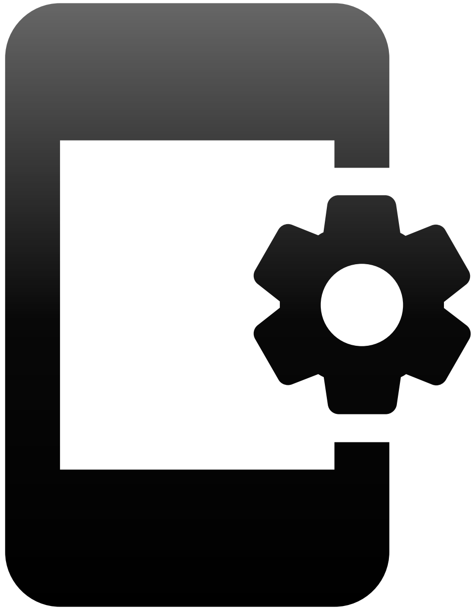
→ Analyze only the relevant security functions

→ Focus on important things (not sophisticated attacks)

# Analysis of Security Functions

# Applications – OS Integration

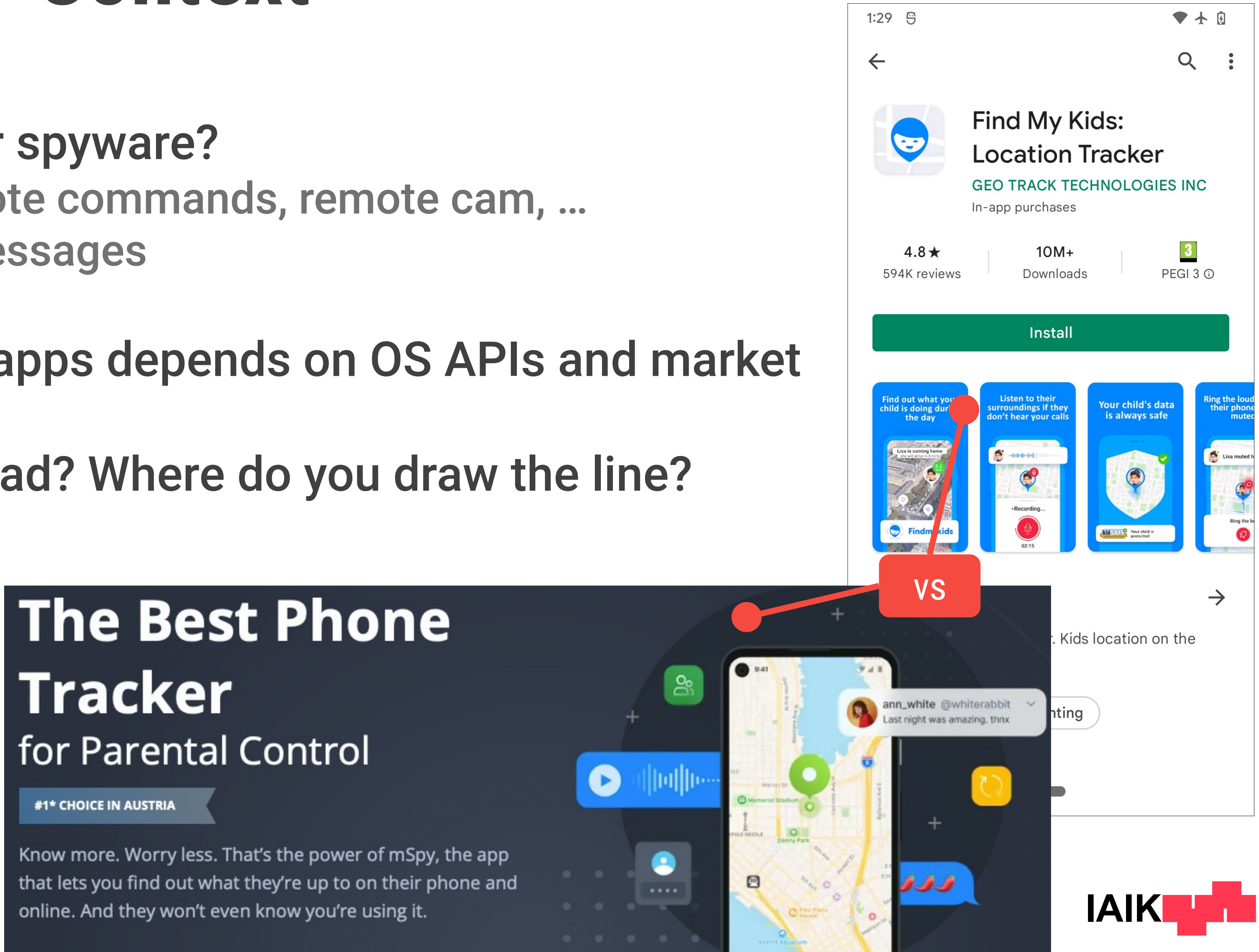
- Access to APIs, Sensors, other Apps
  - Inter-Process Communication (IPC)
  - Android Permissions
  - How does the user know what a permission serves for?
- Protection of application data?
  - Disk encryption vs. App-specific storage
- How deep can apps integrate with the system?
- Rooted / jailbroken vs. normal use cases



Picture: Google / Apache 2.0

# Applications – Context

- Security software or spyware?
  - Remote wipe, remote commands, remote cam, ...
  - Catch and relay messages
- Availability of such apps depends on OS APIs and market
- What makes them bad? Where do you draw the line?



1:29

Find My Kids: Location Tracker

GEO TRACK TECHNOLOGIES INC

In-app purchases

4.8★ 594K reviews | 10M+ Downloads | PEGI 3

Install

Find out what your child is doing during the day
Listen to their surroundings if they don't hear your calls
Your child's data is always safe
Ring the loud their phone muted
VS
. Kids location on the
nting

The Best Phone Tracker

for Parental Control

#1* CHOICE IN AUSTRIA

Know more. Worry less. That's the power of mSpy, the app that lets you find out what they're up to on their phone and online. And they won't even know you're using it.

ann_white @whiterabbit
Last night was amazing, thnx

Source: mspy.com

# Applications – Actors

## Users

- Plenitude of apps available → safe to use?
- What happens to my password?
- Are the developer’s promises met?

„military grade encryption“ ?

## Developers

- Security-critical functions correctly used?
- Adequate parameters chosen?

```
new PBEKeySpec(password, salt, iterationCount, keyLength);
```

Secret? Random?

100? 1000? 10000?

# Applications – Roles

## Analyst

- Traditional approach:
  Apps are either benign or malign

Too narrow-minded?!

- Fight against rising complexity and size
- Obfuscation makes manual analysis tedious
- Many tools available but
  - Often very focused on single aspects or
  - Powerful but not targeted

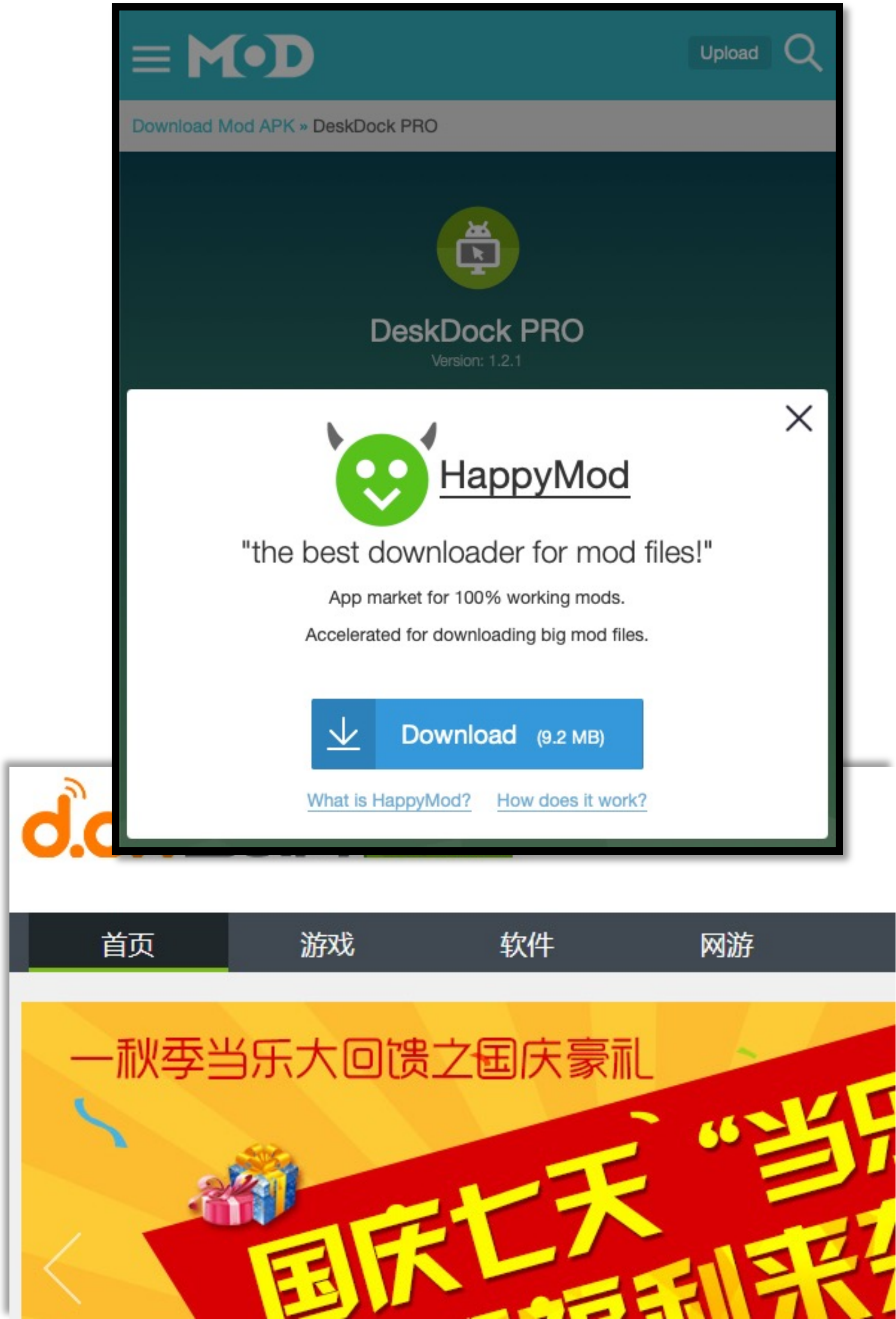
Trade off!

# Applications – Sources

- Depending on platform
  - Google Play
  - Apple App Store
  - F-Droid
  - …

- App Stores: Either walled garden or open
  - Especially critical: 3rd party app stores!

- Other sources: Direct URLs, e-mails, storage, …
  - Malware potential?



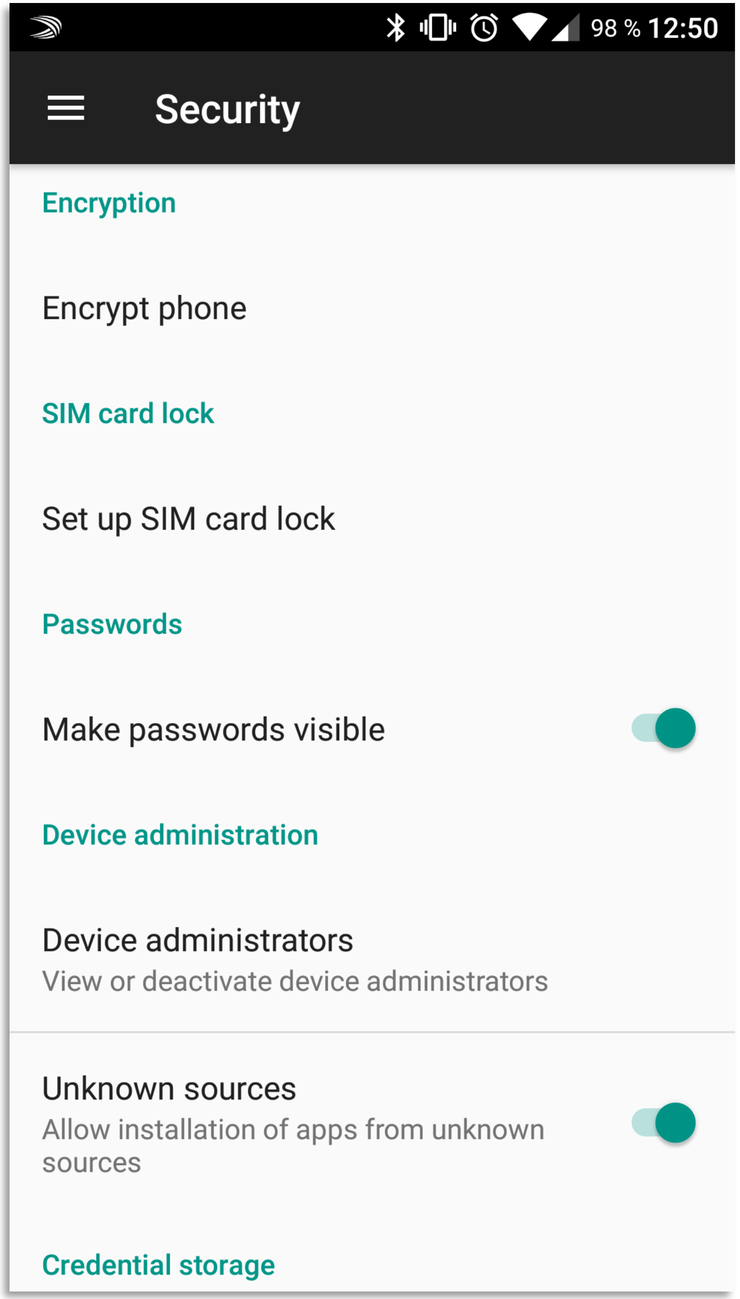
"the best downloader for mod files!"

App market for 100% working mods.

Accelerated for downloading big mod files.

# Applications – Sources

- App installation only from defined sources?
- Can the app be installed from a URL, e-mail, local storage, or USB?
- Does the smartphone warn you?



98 % 12:50

Security

Encryption

Encrypt phone

SIM card lock

Set up SIM card lock

Passwords

Make passwords visible ●

Device administration

Device administrators

View or deactivate device administrators

Unknown sources ●

Allow installation of apps from unknown sources

Credential storage

# Access Protection

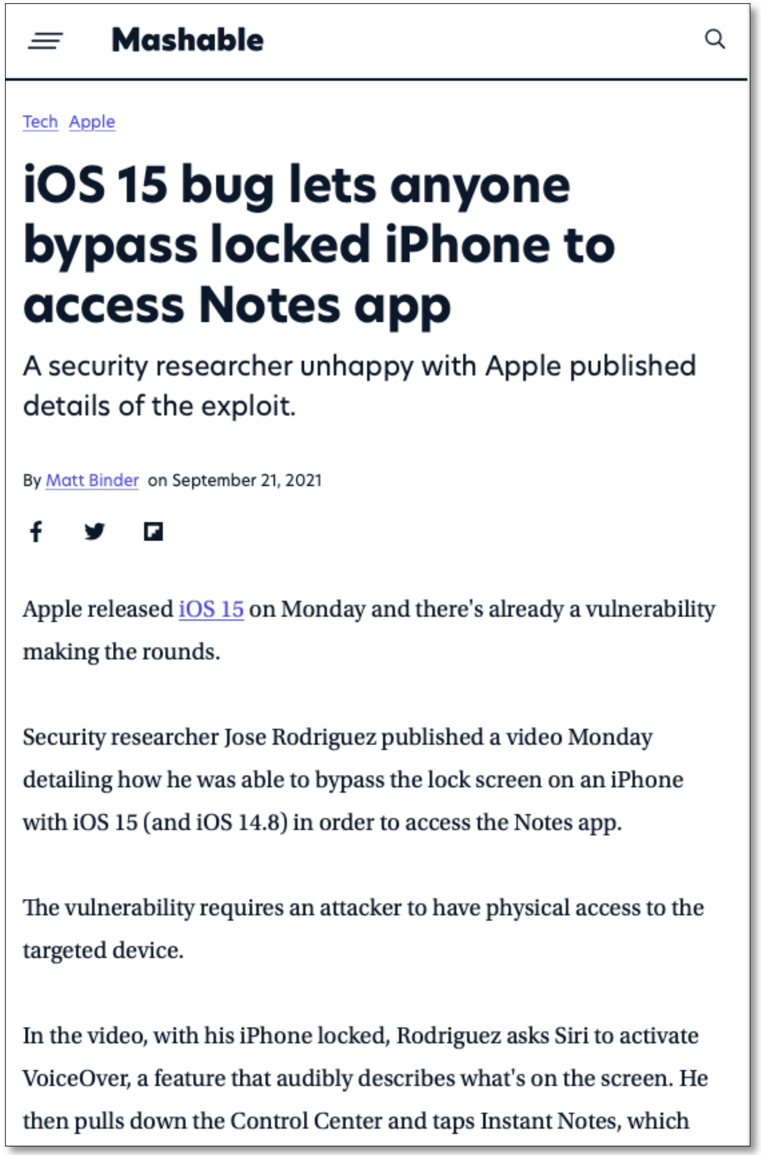
Scenario: *You want to protect your private / business data*

- How is this data protected on a mobile device?
- Basics
  - Smartphone locks
    - PINs, Passwords, Patterns, Biometric Fingerprints!
  - Encryption
    - Obvious, but important differences
- Remote Wipe

# Access Protection – PINs / Passwords

- PIN: Typically 4 digits, quite low entropy
- Passwords: No limits but usability?
- Patterns (Android):
  Nice but entropy? Looking over shoulder…
- Face ID / Unlock: Circumvent with photo?
- Fingerprints: TouchID with iOS 8, Android 6.0

# Access Protection – PINs / Passwords



Tech Apple

## iOS 15 bug lets anyone bypass locked iPhone to access Notes app

A security researcher unhappy with Apple published details of the exploit.

By Matt Binder on September 21, 2021

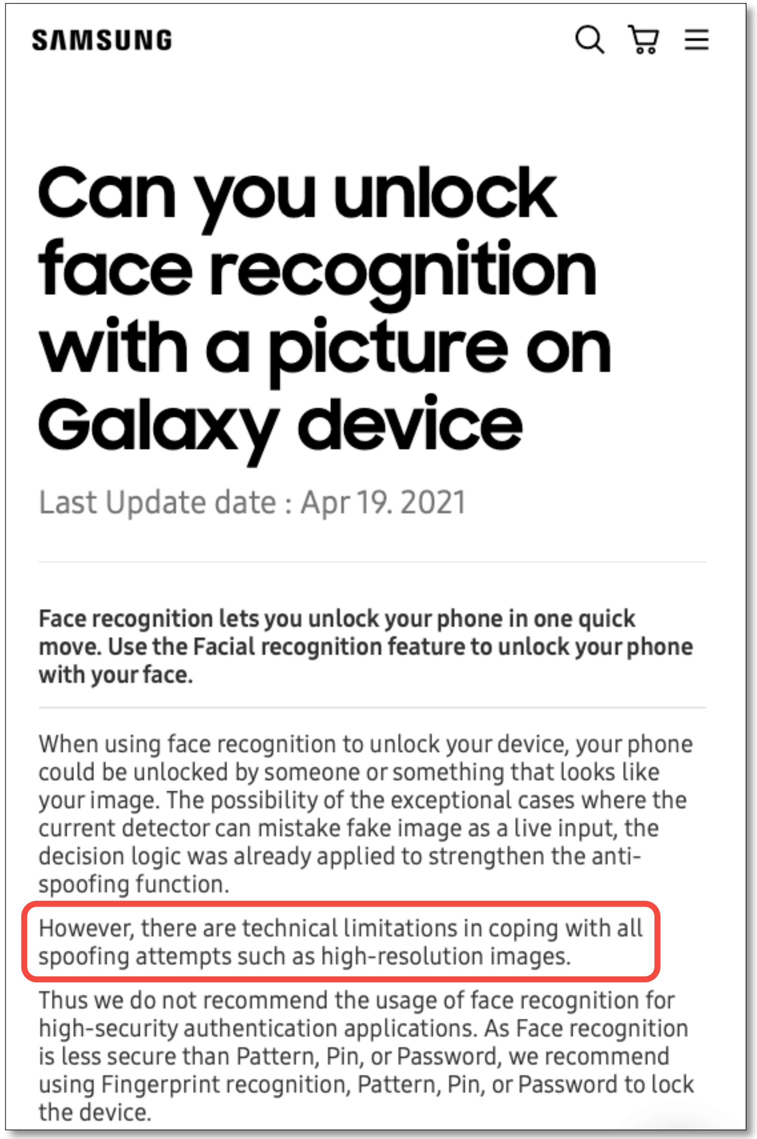
Apple released iOS 15 on Monday and there's already a vulnerability making the rounds.

Security researcher Jose Rodriguez published a video Monday detailing how he was able to bypass the lock screen on an iPhone with iOS 15 (and iOS 14.8) in order to access the Notes app.

The vulnerability requires an attacker to have physical access to the targeted device.

In the video, with his iPhone locked, Rodriguez asks Siri to activate VoiceOver, a feature that audibly describes what's on the screen. He then pulls down the Control Center and taps Instant Notes, which

Source: mashable.com



## Can you unlock face recognition with a picture on Galaxy device

Last Update date : Apr 19. 2021

**Face recognition lets you unlock your phone in one quick move. Use the Facial recognition feature to unlock your phone with your face.**

When using face recognition to unlock your device, your phone could be unlocked by someone or something that looks like your image. The possibility of the exceptional cases where the current detector can mistake fake image as a live input, the decision logic was already applied to strengthen the anti-spoofing function.

However, there are technical limitations in coping with all spoofing attempts such as high-resolution images.

Thus we do not recommend the usage of face recognition for high-security authentication applications. As Face recognition is less secure than Pattern, Pin, or Password, we recommend using Fingerprint recognition, Pattern, Pin, or Password to lock the device.

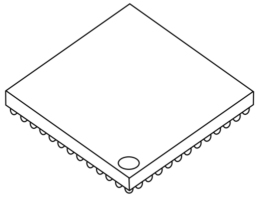
Source: samsung.com

# Access Protection – Encryption

**Protecting data using encryption**

→ Which scope? Whole storage or just certain data?

- Performance issue
  - Symmetric keys, often protected with asymmetric ones
- Where to store the keys?
  - **Nowhere!** → Derived from PIN / password!
  - Isolated Area → Device storage or Secure Element



Picture: ZyMOS / CC BY-SA

# Access Protection – Remote Wipe

- Encryption
  - Huge advantage for remote wipe

- From the Apple Platform Security Guide (Q1 / 2021)

> The metadata of all files in the data volume file system are encrypted with a **random volume key**, which is created when the operating system is first installed or when the device is wiped by a user ... When stored, the encrypted file system key is additionally wrapped by an “effaceable key” ... This key doesn’t provide additional confidentiality of data. Instead, it’s **designed to be quickly erased** on demand (by the user with the “Erase All Content and Settings” option, or by a user or administrator issuing a **remote wipe command** from a mobile device management (MDM) solution, Microsoft Exchange ActiveSync, or iCloud). Erasing the key in this manner renders all files cryptographically inaccessible.

Source: [apple.com](apple.com)

# Access Protection – User Credentials

- How are credentials stored?
  - Hardware / Software?
- Complex passwords will be stored…
  - VPN to infrastructure
- WiFi, VPN, website passwords, etc.
- Are they encrypted, protected via PIN / password?
- How can they be accessed?

# Mobile Device Management (MDM)

- **Deploy security policies that the user cannot change**
  - Password strength, encryption, applications, proxy, VPN, etc.
  - Forbid installation / removal of apps, limit bluetooth functionality, …
- **Get information from device**
  - Location, Call logs, SMS, Backups, …
- **Remote Actions**
  - OS Updates, Device Wipe, enforce device encryption, …

Challenge: Bring-your-own-device!

# Steady Improvements to Platform Security

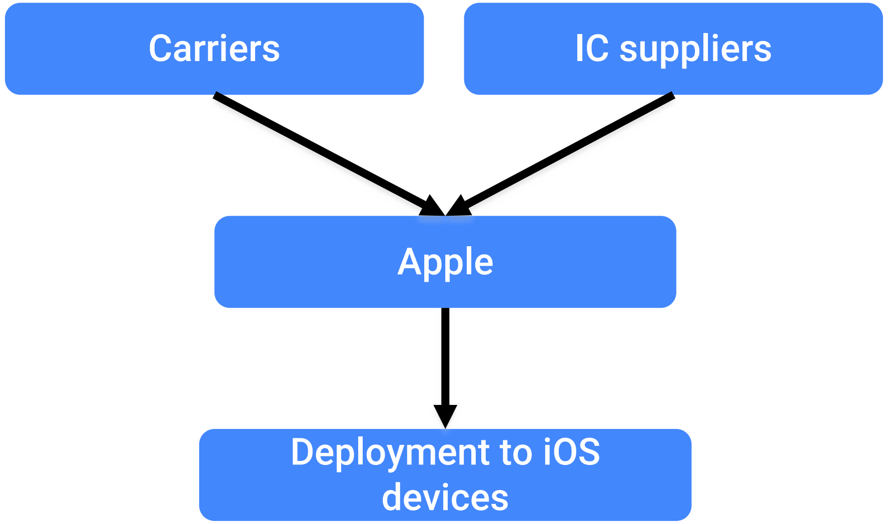
*Reduce attack surface of the system by implementing low-level safeguards*

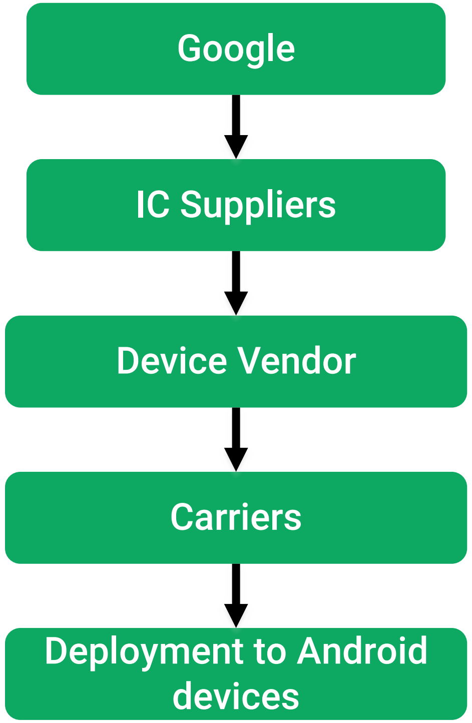
- iOS: Pointer Authentication (PAC)
  - Much more difficult to exploit e.g. buffer overflows to code execution
- Android: SELinux hardening
  - Increasingly restricted permissiveness of policy
- Both:
  - More fine-grained permissions

# Updates

- Security updates are vital, especially in business environments

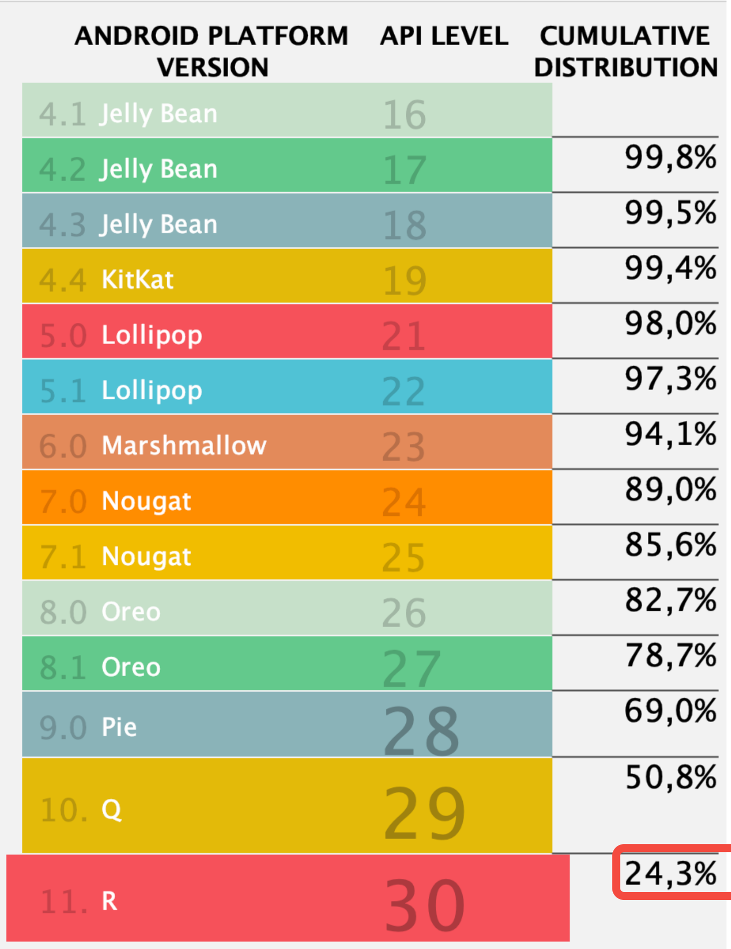
- Android: Slow update adoption
  - Improvements: Project Treble





Source: googleblog.com

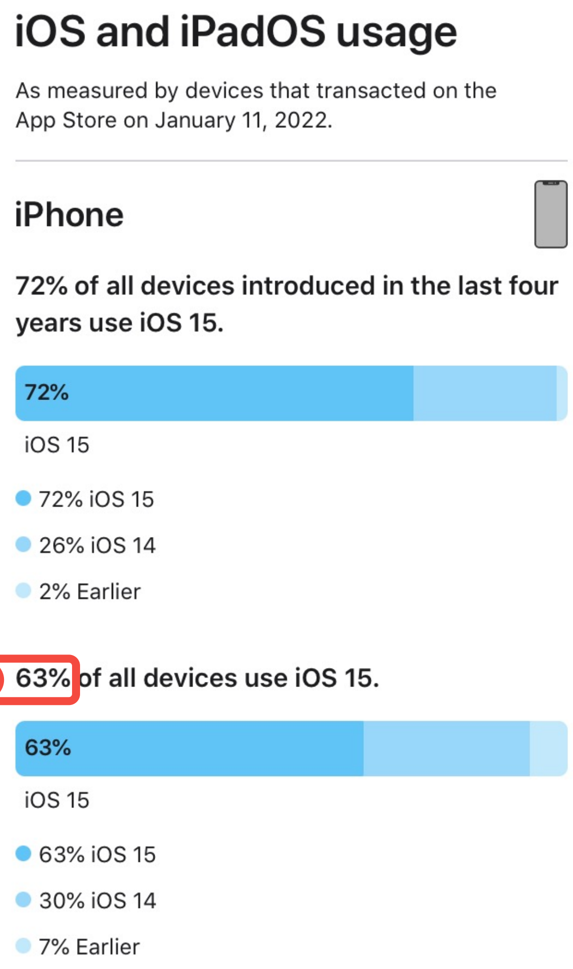
# Version Distributions (Q1/2022)



| ANDROID PLATFORM VERSION | | API LEVEL | CUMULATIVE DISTRIBUTION |
|---|---|---|---|
| 4.1 | Jelly Bean | 16 | |
| 4.2 | Jelly Bean | 17 | 99,8% |
| 4.3 | Jelly Bean | 18 | 99,5% |
| 4.4 | KitKat | 19 | 99,4% |
| 5.0 | Lollipop | 21 | 98,0% |
| 5.1 | Lollipop | 22 | 97,3% |
| 6.0 | Marshmallow | 23 | 94,1% |
| 7.0 | Nougat | 24 | 89,0% |
| 7.1 | Nougat | 25 | 85,6% |
| 8.0 | Oreo | 26 | 82,7% |
| 8.1 | Oreo | 27 | 78,7% |
| 9.0 | Pie | 28 | 69,0% |
| 10. | Q | 29 | 50,8% |
| 11. | R | 30 | 24,3% |

Source: Android Studio

VS



## iOS and iPadOS usage

As measured by devices that transacted on the App Store on January 11, 2022.

### iPhone

72% of all devices introduced in the last four years use iOS 15.

- ● 72% iOS 15
- ● 26% iOS 14
- ● 2% Earlier

63% of all devices use iOS 15.

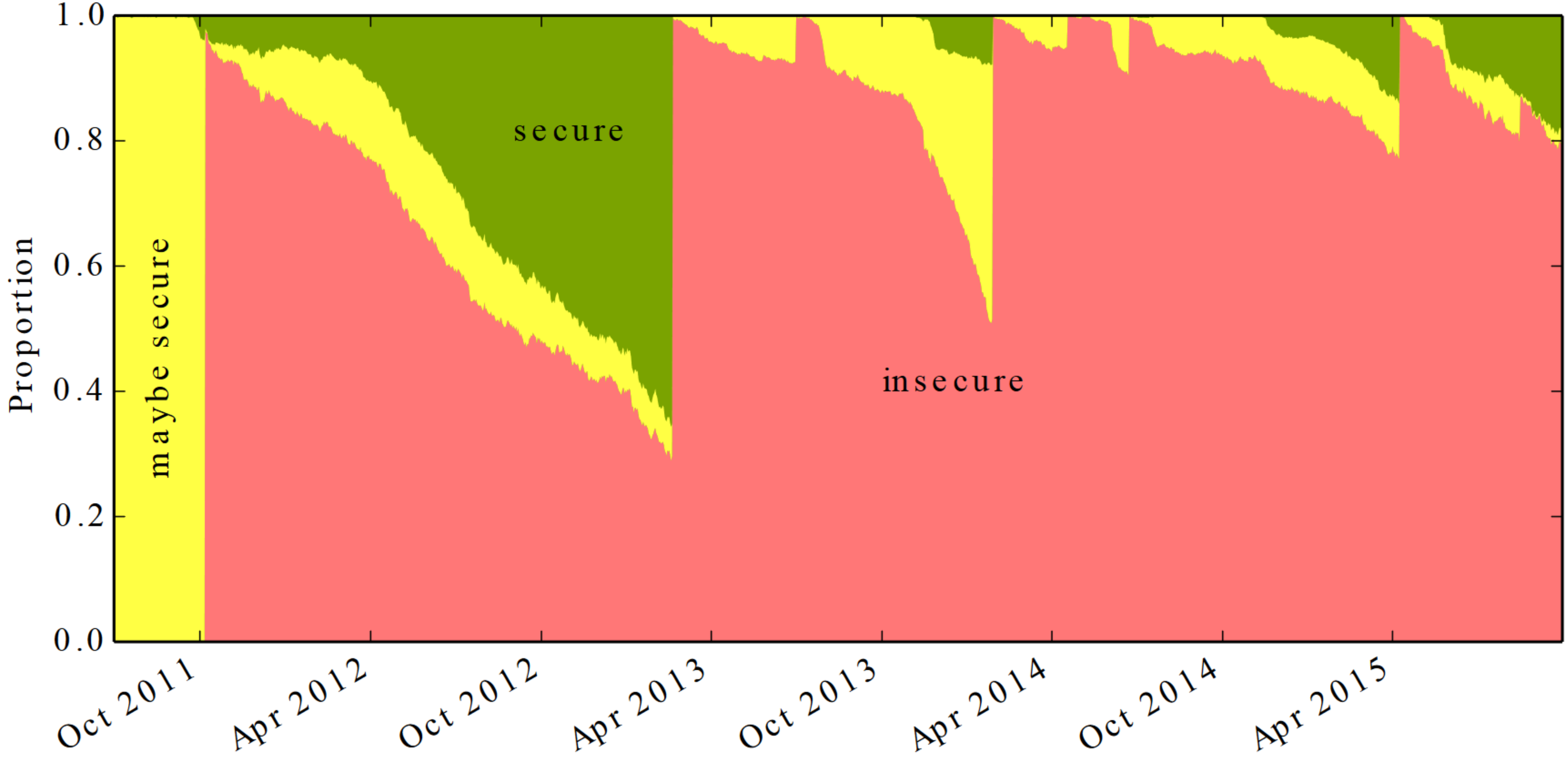
- ● 63% iOS 15
- ● 30% iOS 14
- ● 7% Earlier

Source: apple.com

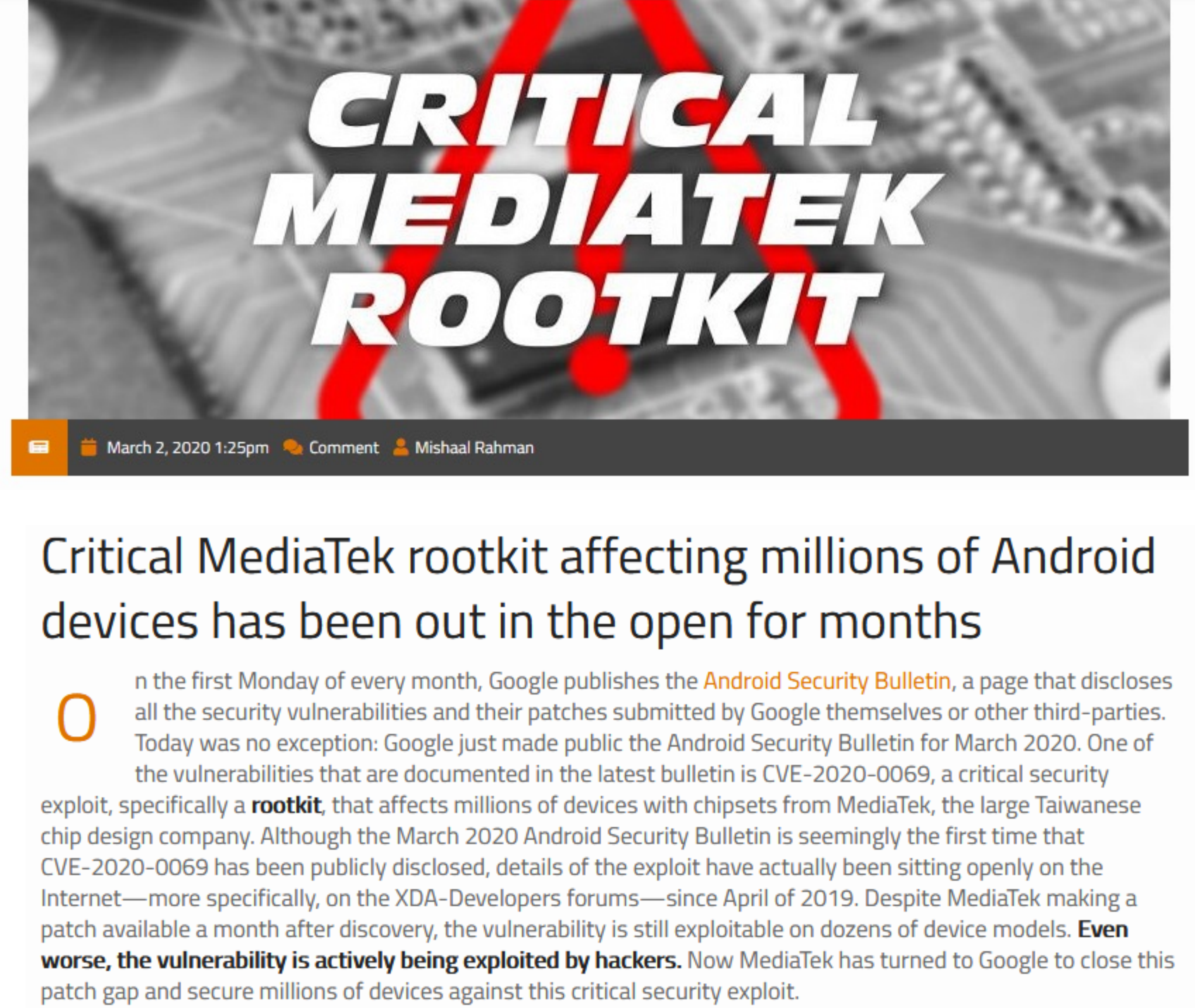
Android 11: Released in September 202**0**

iOS 15: Released in September 202**1**

# Vulnerable Android Devices



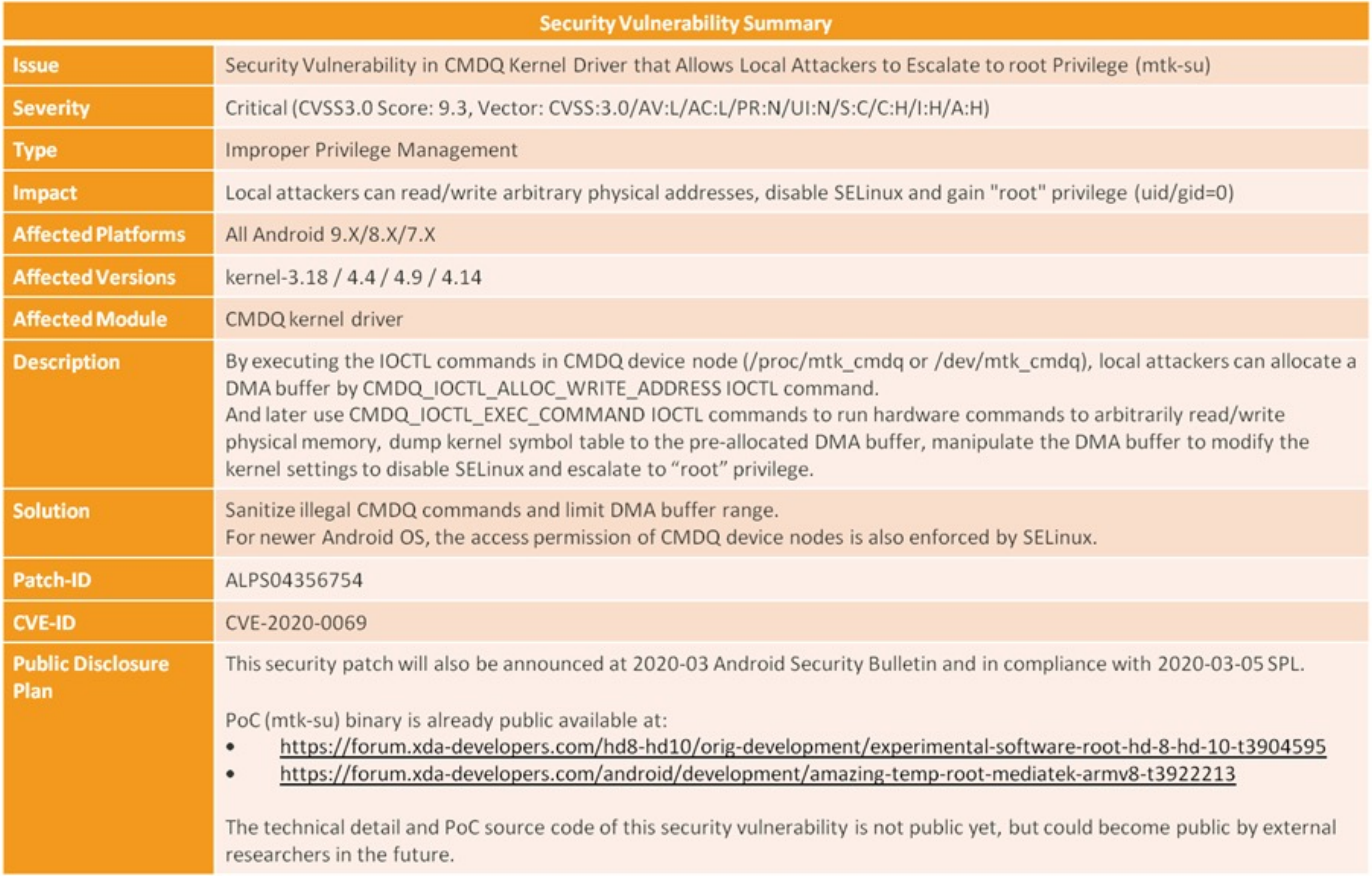
Source: androidvulnerabilities.org



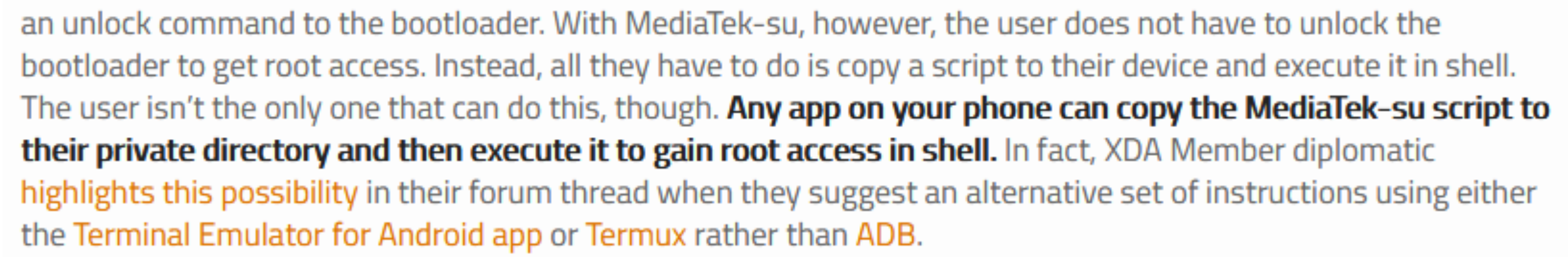
March 2, 2020 1:25pm Comment Mishaal Rahman

# Critical MediaTek rootkit affecting millions of Android devices has been out in the open for months

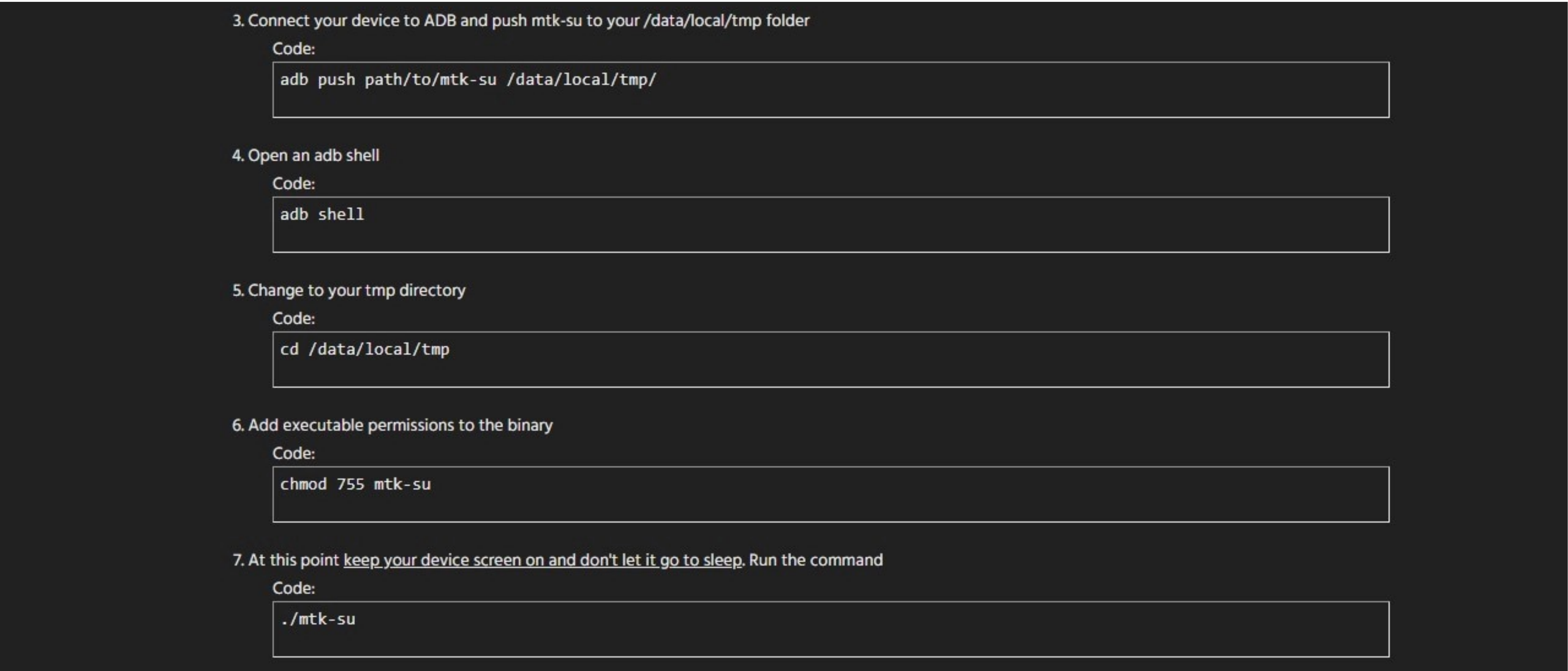
On the first Monday of every month, Google publishes the Android Security Bulletin, a page that discloses all the security vulnerabilities and their patches submitted by Google themselves or other third-parties. Today was no exception: Google just made public the Android Security Bulletin for March 2020. One of the vulnerabilities that are documented in the latest bulletin is CVE-2020-0069, a critical security exploit, specifically a **rootkit**, that affects millions of devices with chipsets from MediaTek, the large Taiwanese chip design company. Although the March 2020 Android Security Bulletin is seemingly the first time that CVE-2020-0069 has been publicly disclosed, details of the exploit have actually been sitting openly on the Internet—more specifically, on the XDA-Developers forums—since April of 2019. Despite MediaTek making a patch available a month after discovery, the vulnerability is still exploitable on dozens of device models. **Even worse, the vulnerability is actively being exploited by hackers.** Now MediaTek has turned to Google to close this patch gap and secure millions of devices against this critical security exploit.



| Security Vulnerability Summary | |
|---|---|
| **Issue** | Security Vulnerability in CMDQ Kernel Driver that Allows Local Attackers to Escalate to root Privilege (mtk-su) |
| **Severity** | Critical (CVSS3.0 Score: 9.3, Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H) |
| **Type** | Improper Privilege Management |
| **Impact** | Local attackers can read/write arbitrary physical addresses, disable SELinux and gain "root" privilege (uid/gid=0) |
| **Affected Platforms** | All Android 9.X/8.X/7.X |
| **Affected Versions** | kernel-3.18 / 4.4 / 4.9 / 4.14 |
| **Affected Module** | CMDQ kernel driver |
| **Description** | By executing the IOCTL commands in CMDQ device node (/proc/mtk_cmdq or /dev/mtk_cmdq), local attackers can allocate a DMA buffer by CMDQ_IOCTL_ALLOC_WRITE_ADDRESS IOCTL command.<br>And later use CMDQ_IOCTL_EXEC_COMMAND IOCTL commands to run hardware commands to arbitrarily read/write physical memory, dump kernel symbol table to the pre-allocated DMA buffer, manipulate the DMA buffer to modify the kernel settings to disable SELinux and escalate to "root" privilege. |
| **Solution** | Sanitize illegal CMDQ commands and limit DMA buffer range.<br>For newer Android OS, the access permission of CMDQ device nodes is also enforced by SELinux. |
| **Patch-ID** | ALPS04356754 |
| **CVE-ID** | CVE-2020-0069 |
| **Public Disclosure Plan** | This security patch will also be announced at 2020-03 Android Security Bulletin and in compliance with 2020-03-05 SPL.<br>PoC (mtk-su) binary is already public available at:<br>• https://forum.xda-developers.com/hd8-hd10/orig-development/experimental-software-root-hd-8-hd-10-t3904595<br>• https://forum.xda-developers.com/android/development/amazing-temp-root-mediatek-armv8-t3922213<br>The technical detail and PoC source code of this security vulnerability is not public yet, but could become public by external researchers in the future. |



an unlock command to the bootloader. With MediaTek-su, however, the user does not have to unlock the bootloader to get root access. Instead, all they have to do is copy a script to their device and execute it in shell. The user isn't the only one that can do this, though. **Any app on your phone can copy the MediaTek-su script to their private directory and then execute it to gain root access in shell.** In fact, XDA Member diplomatic highlights this possibility in their forum thread when they suggest an alternative set of instructions using either the Terminal Emulator for Android app or Termux rather than ADB.



3. Connect your device to ADB and push mtk-su to your /data/local/tmp folder

Code:

```
adb push path/to/mtk-su /data/local/tmp/
```

4. Open an adb shell

Code:

```
adb shell
```

5. Change to your tmp directory

Code:

```
cd /data/local/tmp
```

6. Add executable permissions to the binary

Code:

```
chmod 755 mtk-su
```

7. At this point keep your device screen on and don't let it go to sleep. Run the command

Code:

```
./mtk-su
```

# iOS – Latest CVEs with score >= 5.0

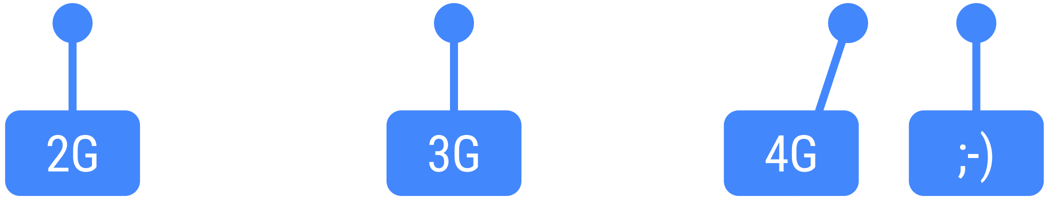
| CVE ID | Update date | Score | Access | Complexity | Patched? |
|---|---|---|---|---|---|
| CVE-2021-30996 | 2021-12-29 | 7.6 | Remote | High | ✓ |
| Race Condition: A malicious application may be able to execute arbitrary code with kernel privileges. | | | | | |
| CVE-2021-30995 | 2022-01-03 | 5.1 | Remote | High | ✓ |
| Buffer Overflow: An attacker in a privileged network position may be able to execute arbitrary code. | | | | | |
| CVE-2021-30993 | 2022-01-03 | 6.8 | Remote | Medium | ✓ |
| Race Condition: A malicious application may be able to elevate privileges. | | | | | |
| CVE-2021-30991 | 2021-12-29 | 9.3 | Remote | Medium | ✓ |
| Out-Of-Bounds Read: An attacker in a privileged network position may be able to execute arbitrary code. | | | | | |
| CVE-2021-30985 | 2021-12-29 | 9.3 | Remote | Medium | ✓ |
| Out-Of-Bounds Write: A malicious application may be able to execute arbitrary code with kernel privileges. | | | | | |
| CVE-2021-30984 | 2022-02-06 | 5.1 | Remote | High | ✓ |
| Race Condition: Processing maliciously crafted web content may lead to arbitrary code execution. | | | | | |

Source: cvedetails.com

# Communication

*Key aspect of smartphone: broadband always-on Internet connection*

- Mobile Network: GRPS, EDGE, UMTS, HSPA+, LTE, 5G



2G 3G 4G ;-)

- WiFi: Infrastructure, Ad-Hoc, Direct Mode
- Bluetooth: Low Energy?
- NFC, USB (+ Host), …

# Communication – Mobile Networks

- Many standards: GPRS/GSM has many security problems
  - A5/0: broken (and partly banned)
  - A5/1: broken using rainbow tables in 2009
  - A5/2: export version, broken in 1999
  - A5/3: Backport of Kasumi UMTS cipher

https://gsmmap.org

- Security is deployed on higher levels (VPNs, HTTPS, etc)

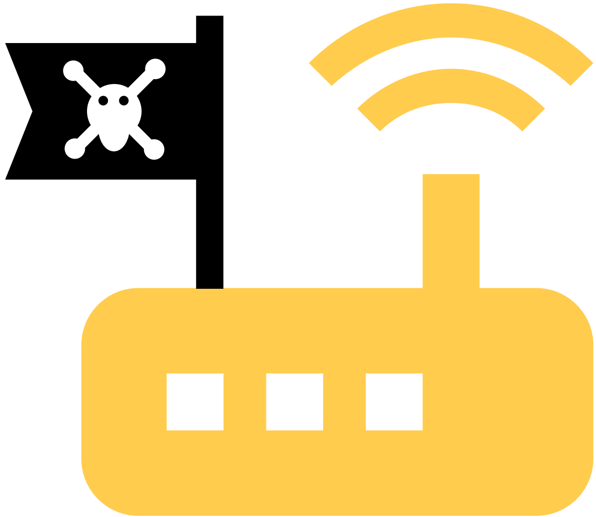
- However:
  - 2G still widely available, particularly in Europe
  - Telephone, SMS, MMS services integrated as apps into phone
  - MMS with Malware, e.g. „Stagefright“ on Android

# Communication – WiFi

- Huge problem: Open WiFi access points

- Old problems re-emerge:
  - ARP Poisoning
  - Sniffing unencrypted traffic
  - Phishing
  - Faking DNS entries
  - Faking TLS certificates (MITM → HTTPS)

**Tools:**
- Intercepter-NG
- Aircrack-ng
- …



Picture: Google / Apache 2.0

IAIK TU Graz

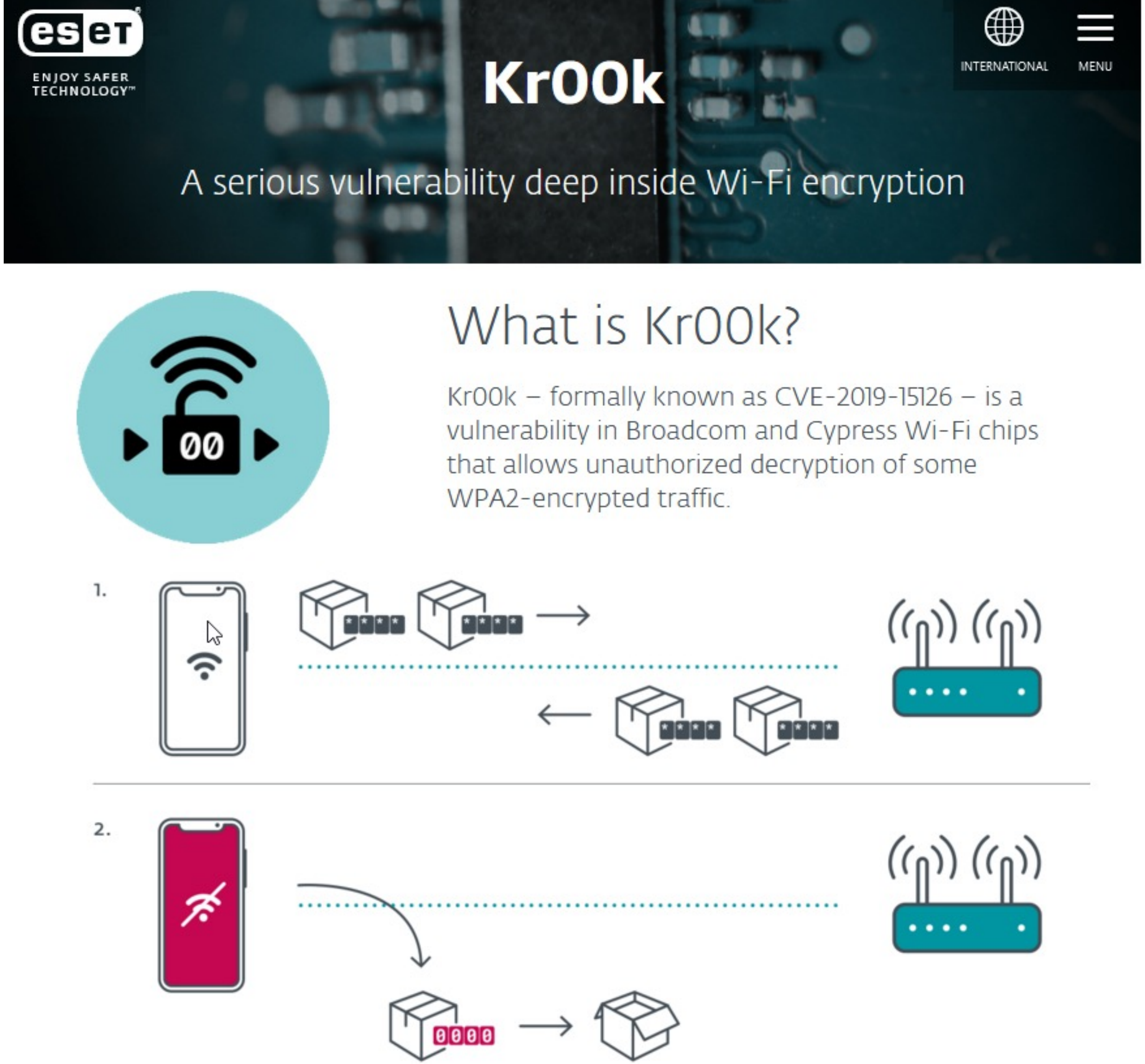
# Communication – WiFi

**Assuming that OS does certificate validation correctly...**

→ MDM: Force rejection of invalid HTTPS certificates?

- What about apps?
  - Encrypted traffic?
    - Changes in recent Android / iOS → push developers to use HTTPS whenever possible
  - Do they verify the certificate (correctly)?
- WiFi location (& user) tracking
  - Android: „Location service may scan for WiFis although WiFi disabled“
  - MAC address randomization since iOS 8 and Android 10

Source: [eset.com](eset.com)



# Kr00k

A serious vulnerability deep inside Wi-Fi encryption

## What is Kr00k?

Kr00k – formally known as CVE-2019-15126 – is a vulnerability in Broadcom and Cypress Wi-Fi chips that allows unauthorized decryption of some WPA2-encrypted traffic.

# Communication – VPN

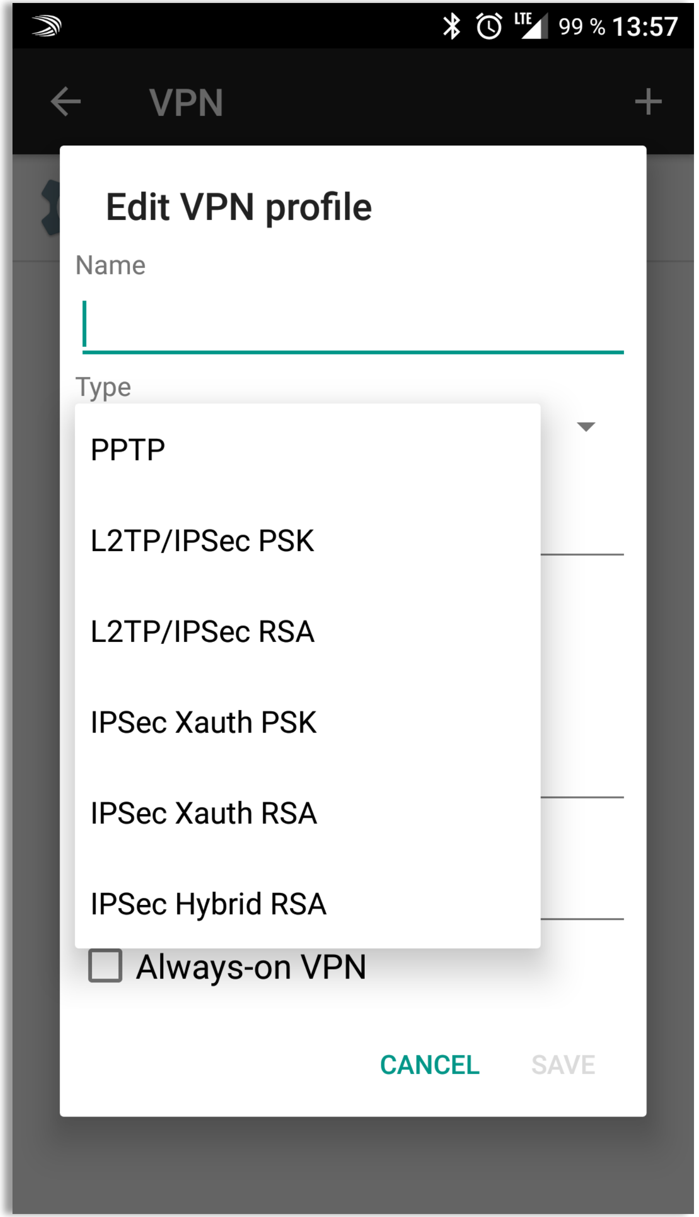
- Virtual Private Networks
  - Provide secure tunnel to company network
  - Many protocols: PPTP, IPSec, L2TP, TLS

- Which one to use?
  - PPTP → security holes with MS-CHAPv2 auth

- Shared keys vs. Certificates

- Supported encryption algorithms? Hash algorithms?

- Storage of VPN Client credentials?



# Communication – VPN

**Force all traffic over VPN...**

→ Avoid problems with open WiFis

→ Use security functions of company, e.g. proxies, virus scanners, etc

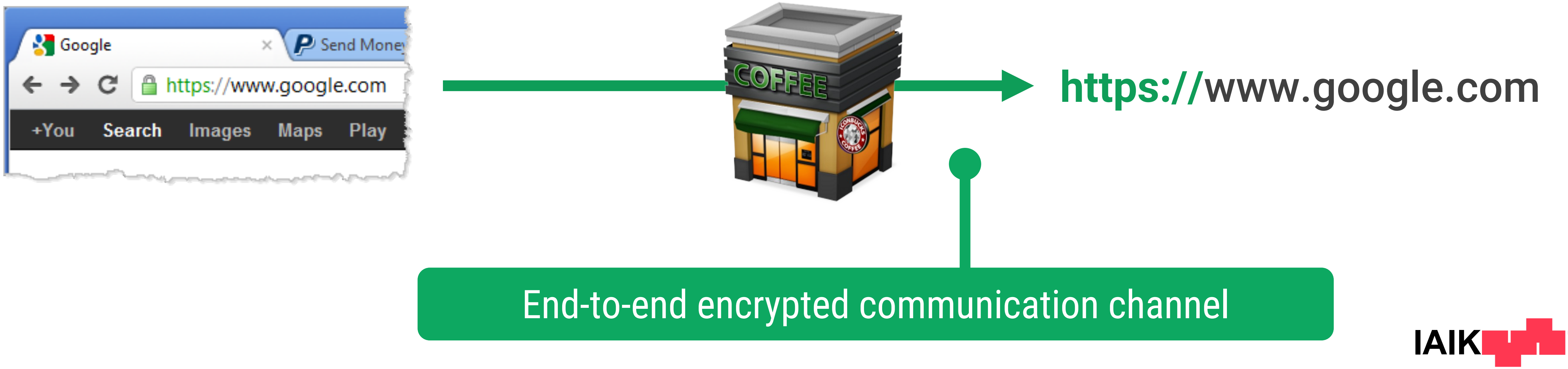
→ Only connect to trusted VPNs!

**Attackers cannot...**

*1. ... read the transfer*

*2. ... tamper the data transferred*

*3. ... impersonate the destination*



Picture: blaugrana-tez / CC BY-NC-ND

# Communication - Bluetooth

## Problems by design

- Visibility
- Pairing

## Problems by implementation

- BrakTooth (2021): DoS or code execution on 1400 chipsets Source: asset-group.github.io
  - Family of vulnerabilities in Bluetooth Classic Controllers
  - All running the same vulnerable firmware
- SweynTooth (2020): DoS, code execution or security bypass Source: asset-group.github.io
  - Family of vulnerabilities in Bluetooth LE SDKs of multiple SoC vendors
- Attackers just need to be in radio range
- Highlight flaws in the Bluetooth Stack Certification Process

# Communication - Location

Finding a GPS fix can take a long time…

→ ***Solution: Assisted GPS (A-GPS)***

- Send coarse location + IMSI to SUPL server
  - *„Secure User Plane Location Protocol“*
- SUPL server depends on device

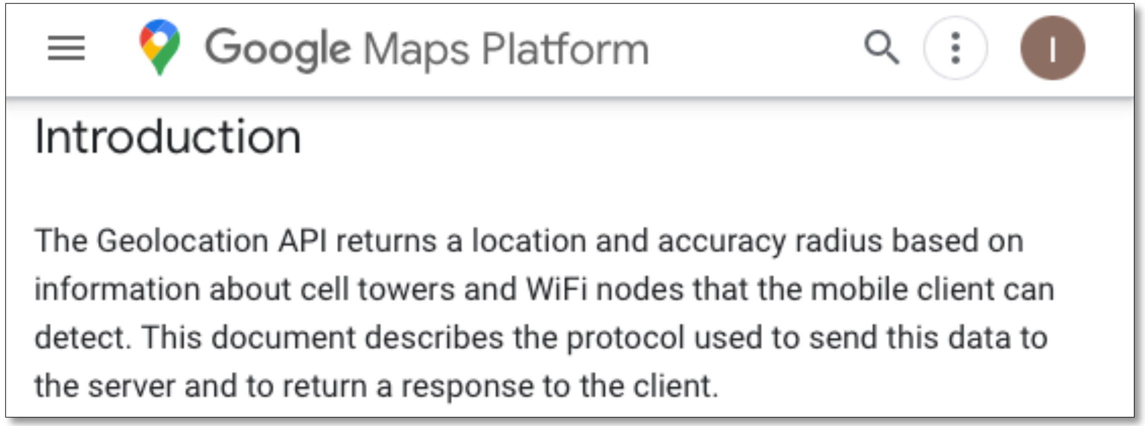
```
cat /etc/system/gps.conf | grep SUPL_HOST (or /vendor/etc/gps.conf)
SUPL_HOST=supl.google.com # Google
SUPL_HOST=supl.sonyericsson.com # Sony
SUPL_HOST=supl.qxwz.com # China(?)
...
```

***Good**: TLS is used to protect transfer*

***Bad**: The certificate's validity is not checked on some devices!* Source: wirelessmoves.com

# Communication - Location

*Google and others can locate you from connected WiFi nodes and cell towers*

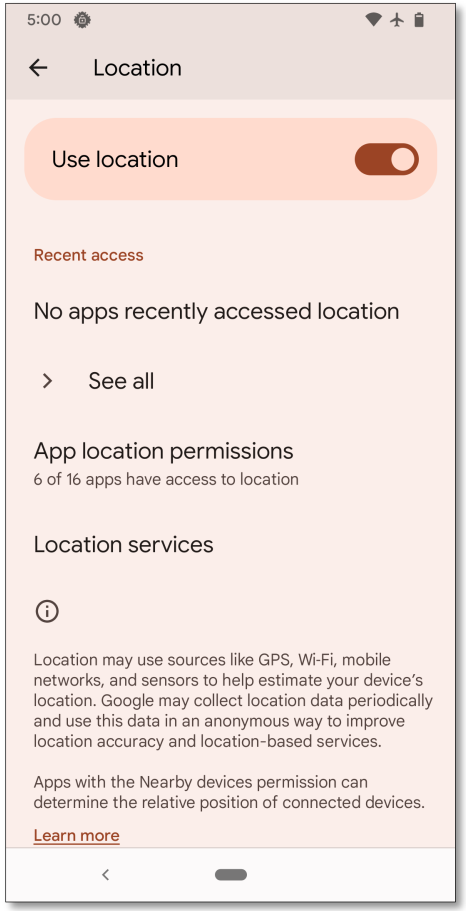


## Introduction

The Geolocation API returns a location and accuracy radius based on information about cell towers and WiFi nodes that the mobile client can detect. This document describes the protocol used to send this data to the server and to return a response to the client.

Source: developers.google.com

**How do they learn this mapping?**



Use location

Recent access

No apps recently accessed location

See all

App location permissions

6 of 16 apps have access to location

Location services

Location may use sources like GPS, Wi-Fi, mobile networks, and sensors to help estimate your device's location. Google may collect location data periodically and use this data in an anonymous way to improve location accuracy and location-based services.

Apps with the Nearby devices permission can determine the relative position of connected devices.

Learn more

> "Google may collect location data periodically and use this data in an anonymous way to improve location accuracy and location-based services"

# Communication – NFC

- Near Field Communication (NFC)
  - Short range (freq. 13.56 MHz) → some kind of security
  - Payments, Social Networking, Access tokens, …
- Devices can act as both reader and tag
- 2022: MitM attack against Apple Pay Source: practical_emv.gitlab.io
  - Payments without user authorization
- 2019: Flaw in Android Beam Source: trendmicro.com
  - Allows installing apps through NFC (install dialog has to be confirmed though)

Picture: mirrorsnake / CC BY-SA

# Communication: USB

- Most modern smartphones can act as USB host and client / accessory

- iOS
  - Proprietary protocols for Network, Audio, Screen Sharing via USB (Largely undocumented)
  - iOS Accessory Protocol (Licensable)
  - Debugging and management via ***usbmuxd*** and ***lockdownd*** (Reverse-Engineered by libimobiledevice)

- Android
  - Class-compliant Network, Audio implementations
  - Open Accessory Protocols for Audio and custom functionality
  - Debugging and socket muxing via ***Android Debug Bridge*** (ADB)

# Communication: USB

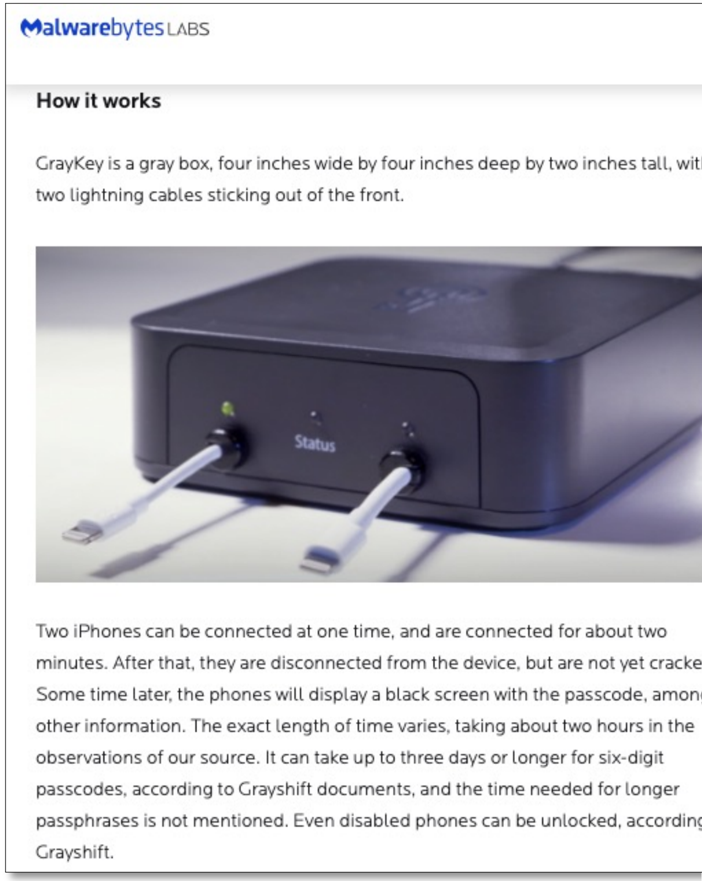
- USB Debugging Interfaces pose Security Risk: *“JuiceJacking”*

- **2012/2013: Android 4.2.2 / iOS 7 add user consent for debug connection** Sources: [cs.android.com](cs.android.com) / [theta44.org](theta44.org)

- **2017: GrayKey Box**
  - Brute-force pin and extract data from locked iOS device

- **2018: iOS 12 locks USB 1 hour after screen lock**

- **Today: O.MG Cable**
  - Computer hidden in charging cable
  - Keystroke injection via WiFi connection



Malwarebytes LABS

**How it works**

GrayKey is a gray box, four inches wide by four inches deep by two inches tall, wit two lightning cables sticking out of the front.

Two iPhones can be connected at one time, and are connected for about two minutes. After that, they are disconnected from the device, but are not yet cracke Some time later, the phones will display a black screen with the passcode, amon other information. The exact length of time varies, taking about two hours in the observations of our source. It can take up to three days or longer for six-digit passcodes, according to Grayshift documents, and the time needed for longer passphrases is not mentioned. Even disabled phones can be unlocked, according Grayshift.

Source: [malwarebytes.com](malwarebytes.com)

# Communication: USB

- Multiple iOS Jailbreaks were made possible by exploits of USB vulnerabilities

- Checkrain jailbreak / Checkm8 exploit (2019):
  - Use-after-free in USB code Source: [habr.com](habr.com)
  - Same code in iOS and BootROM

- evasi0n jailbreak (2013):
  - Insufficient pointer validation in `IOUSBDeviceFamily` driver Source: [azimuthsecurity.com](azimuthsecurity.com)

# Outlook

- 25.03.2022
  - Key & Data Storage on Mobile Devices

- 01.04.2022
  - iOS Platform Security