ISEC TU Graz

Motivation

Mobile Security 2025

Florian Draschbacher
florian.draschbacher@tugraz.at

Some slides based on material 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

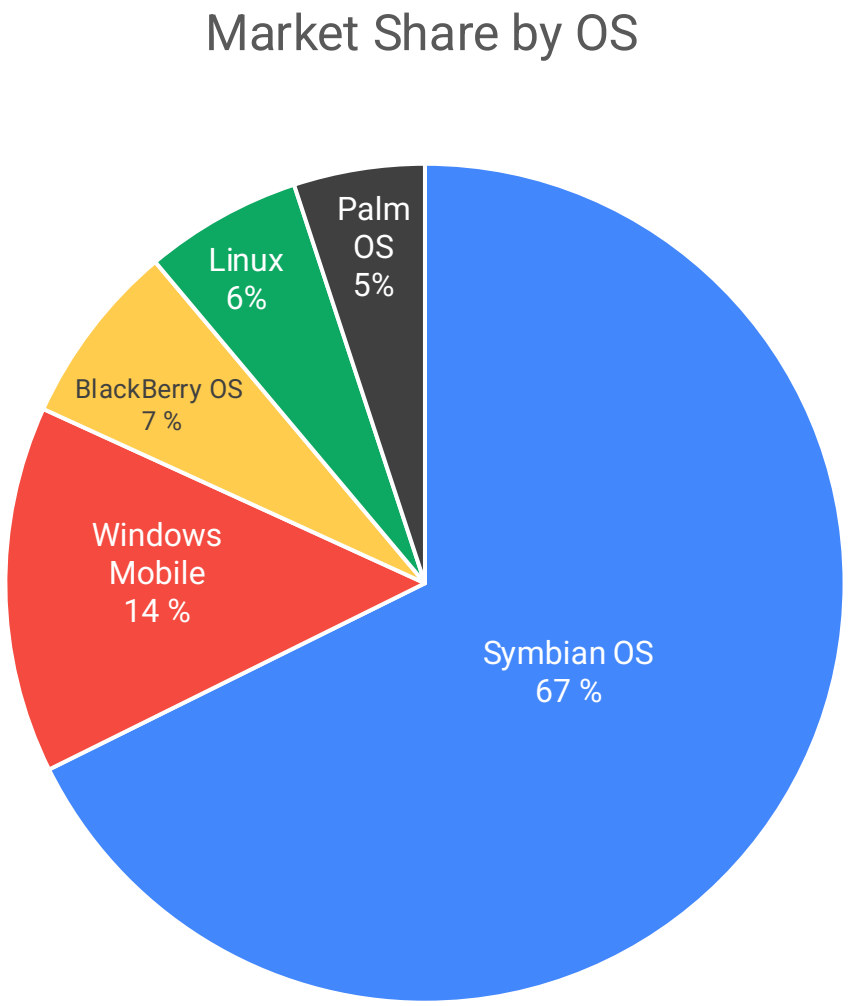
Early Smartphones

- 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

Smartphone Trends By 2006

- Code Signing and User-grantable Permissions
  - Symbian OS, BlackBerry OS
- Linux kernel and custom Java VM
  - Nokia Maemo platform
  - Motorola EZX platform
- Smartphones try to enter consumer market



Source: canalys.com

2007: The iPhone

- First smartphone fully targeted at consumer market
- Novel capacitive touchscreen UI
  - Pencil-free on-screen keyboard
- ”Full-featured” web browser
- Key to emergence of app industry
  - Only web apps in iPhone 1.0
  - Only code-signed native apps later

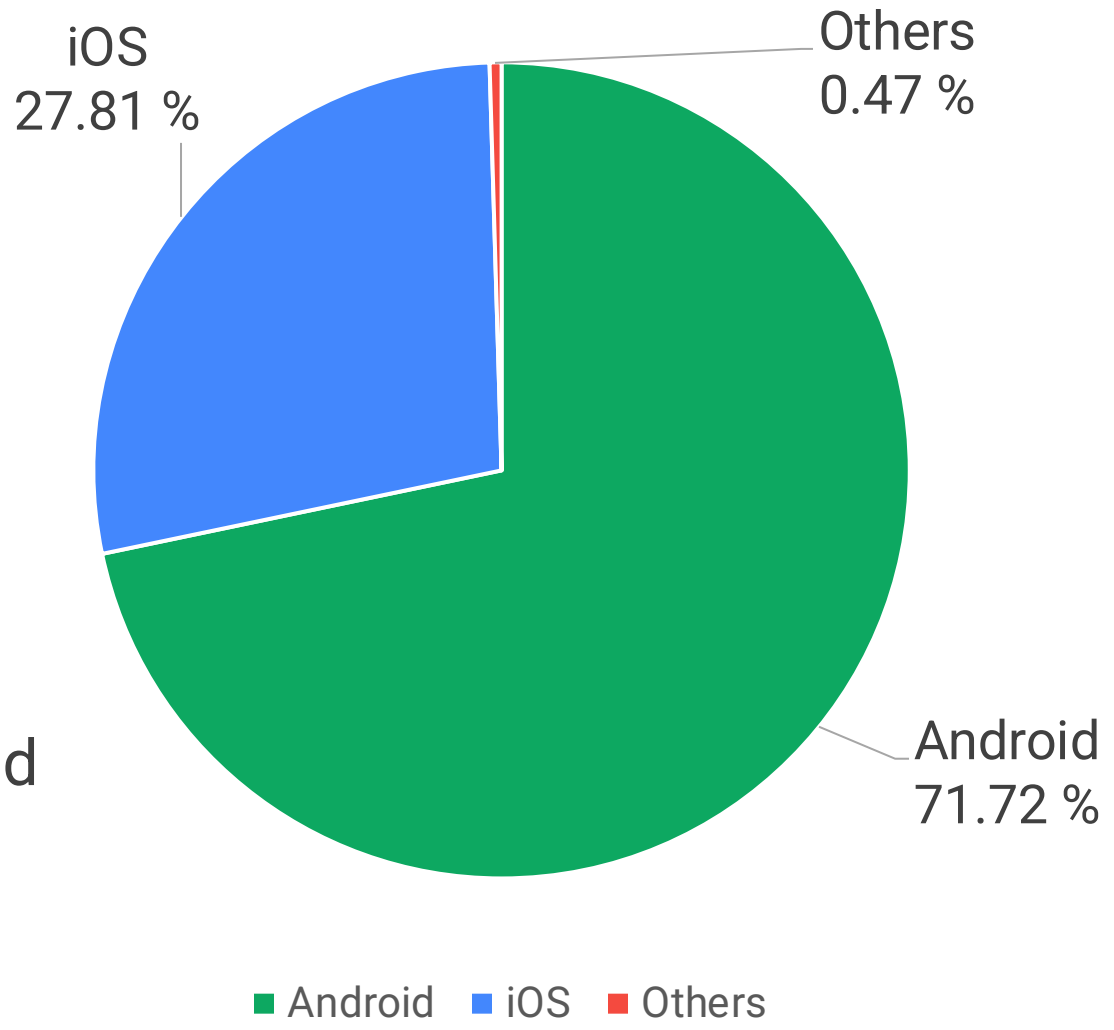
2008: First Android-based smartphone

- Startup founded in 2003
  - Goal: Develop OS for smarter mobile devices
  - Competitor to Symbian and Windows Mobile
- 2005: Sold to Google
  - Shipped on devices of Open Handset Alliance
- 2007: Radical shift after introduction of iPhone
  - Focus on touchscreen devices

Image: Flickr: Michael Oryl/ CC-BY-SA

Today

- Android most popular OS
  - Even when compared to desktops
- 6.7B smartphone subscriptions
  - Close to 70% of world's population!
- Account for ~62% of Internet traffic
- More than 3 million apps for iOS & Android
  - Industry of hundreds of billions of $



Sources: statcounter.com; statista.com; Data from late 2024 / early 2025,

Applications

- Social networks: X (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?“

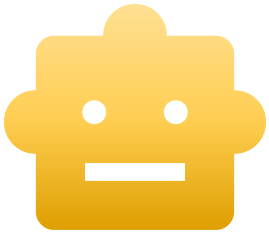
Applications

- 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

Everything turns smart



- iOS was the first in a family of related mobile OSs
  - watchOS, tvOS, audioOS

Image: Screenshot, apple.com

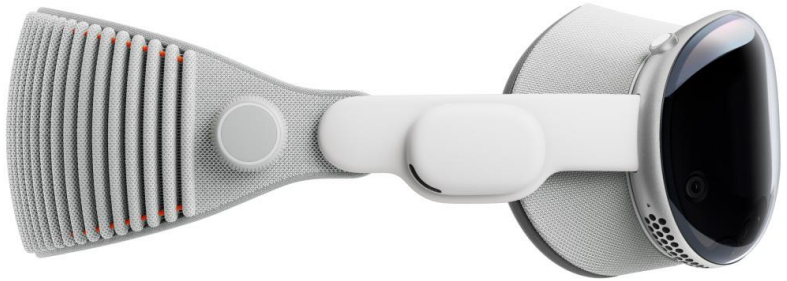
- Android is everywhere
  - Android TV, Wear OS, Home appliances, …

- Emerging market of embedded, connected, smart devices
  - Similarities to smartphones
  - Internet of Things

Image: lametric.com

Entirely new smart device categories

- Voice Assistants
- Smart Watches
- Spatial Computers
- Smart Mirrors

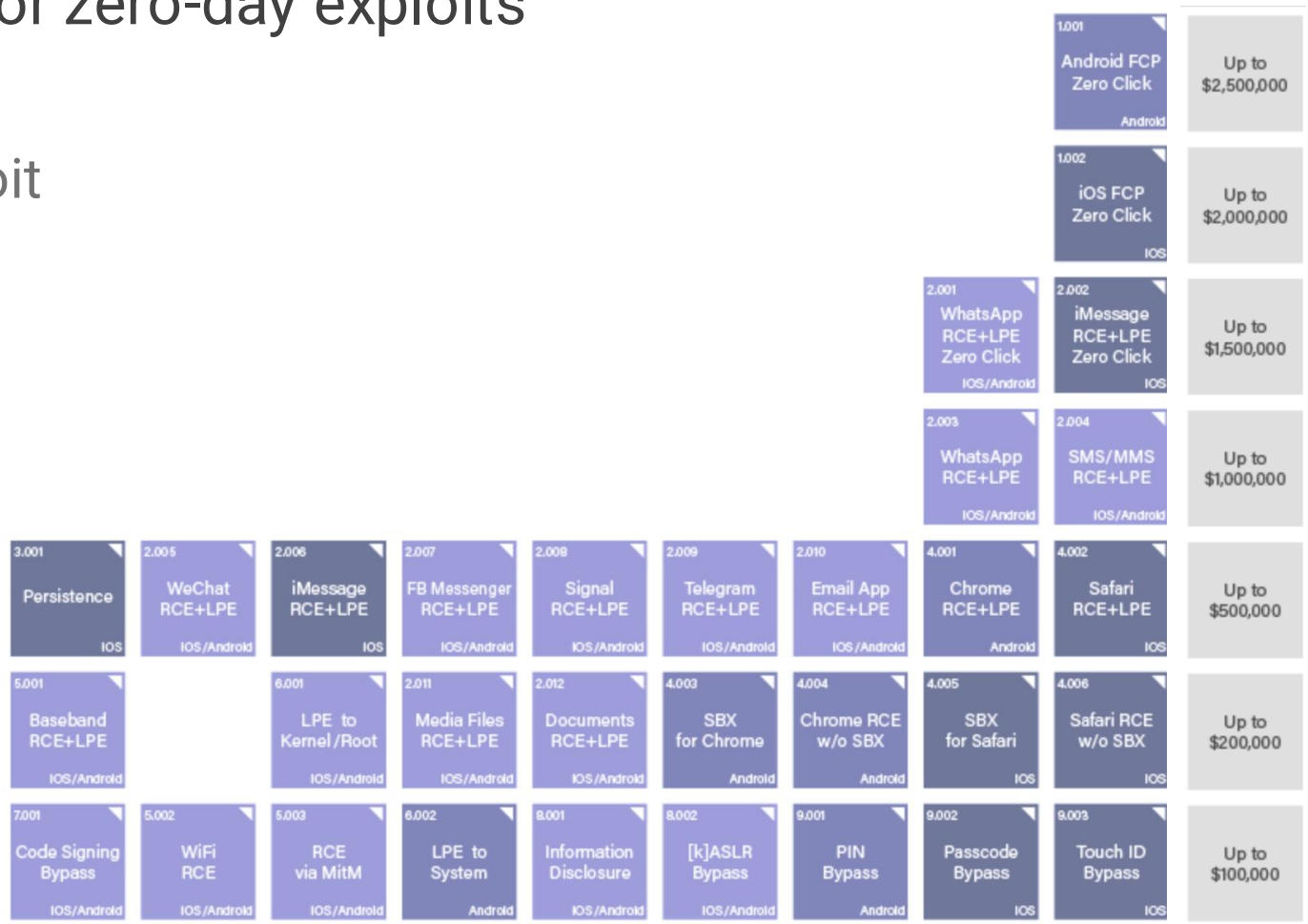


Threats

Mobile Devices Attract Attackers

- There is an industry and market for zero-day exploits
- Zerodium (defunct now):
  - Up to $2,500,000 for mobile exploit

- Apple:
  - Up to $2,000,000 Source: apple.com
- Google:
  - Up to $1,000,000 Source: google.com
- Samsung:
  - Up to $1,000,000 Source: samsungmobile.com
- Microsoft:
  - Up to $250,000 Source: microsoft.com

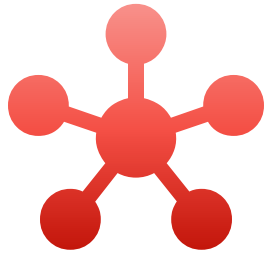


Source: zerodium.com

ISEC TU Graz

Why Are Mobile Devices Interesting to Attackers?

- High density of sensitive data
  - “Many valuable assets”
- High degree of connectedness
  - “Large attack surface”
- Unique challenges
  - Security vs. Usability
  - Security vs. Innovation
  - Security vs. Customization



Pictures: Google / Apache 2.0

Assets on Mobile Devices

Attackers are aiming at…

- Data (Confidentiality)
  - Personal Data: Pictures, Messages, Files, Browsing History, …
  - Sensor Data: GPS, Microphone, Camera, Accelerometer, …
  - Authentication Data: Passwords, Credentials, Bank Accounts, Car Keys, …
- Availability
  - What if you cannot call emergency when you need it
  - What if you can no longer access your data?
- Device Resources
  - CPU power, Display space, Network access, …

Attack Surface: Cellular

- 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

Cellular Attack: Stagefright Vulnerabilities

- A very dangerous vulnerability haunted Android in 2015
- Media files from many sources were parsed in *libStagefright*
- The code ran as very powerful user media_server
- There was an integer overflow in the library
  - Could be exploited to gain code execution

- Attack: Send MMS with maliciously crafted image attachment
  - Get remote code execution just by knowing victim's phone number

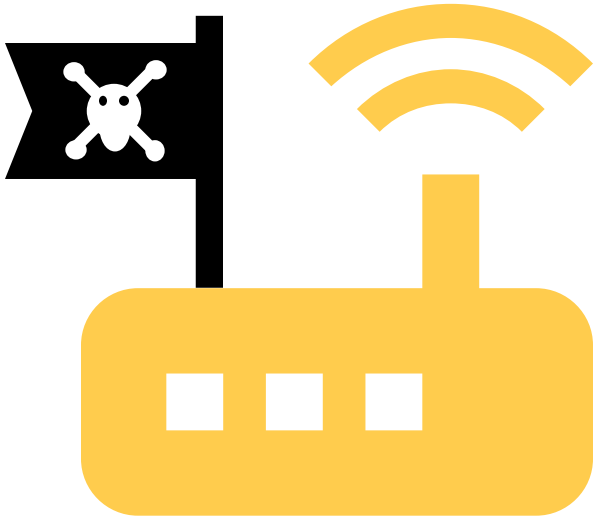
Cellular Attack: Stagefright Exploit

Attack Surface: 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

Attack Surface: WiFi

- Problems also in the protocol itself
  - Design and Implementation
- 2017: KRACK
  - Key Reinstallation Attack effectively allowed bypassing WPA2 encryption
- 2019: KR00K
  - Newer variation of KRACK
- 2021: FragAttacks
  - Inject WiFi frames into WPA3 protected network
  - Allows e.g. to enforce malicious DNS server

Attack Surface: WiFi

- Interesting demo by Ian Beer of Google's Project Zero
  - AWDL Proximity exploit
- AWDL: Apple Wireless Direct Link
  - Ad-hoc WiFi protocol underlying AirDrop, AirPlay, CarPlay, Handoff, Quickstart, …
- iOS kernel driver contained double-free in frame parsing
  - Can be exploited over the air!
- Enables kernel read and write, which allows infiltrating any app process

Attack Surface: WiFi

AWDL Proximity exploit

Source:
https://googleprojectzero.blogspot.com/2020/12/an-ios-zero-click-radio-proximity.html

Attack Surface: 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

Attack Surface: Bluetooth

- Both iOS and Android added device finding capabilities
  - Based on Bluetooth technology, even if Bluetooth is seemingly disabled
- Devices constantly scan for tags in surrounding: Report lost tags to server
- Attack allows exploiting this network
  - Track any device‘s location
  - Just using its Bluetooth address


Macworld

Home / Software / News

NEWS

Researchers hack Bluetooth devices so they can be trackable in Apple’s Find My

The nRootTag hack can be performed remotely and doesn't require administrator privilege, which is a little scary.

By Roman Loyola

Senior Editor, Macworld | MAR 1, 2025 2:46 PM PST

Source: macworld.com

Attack Surface: 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

Attack Surface: 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)

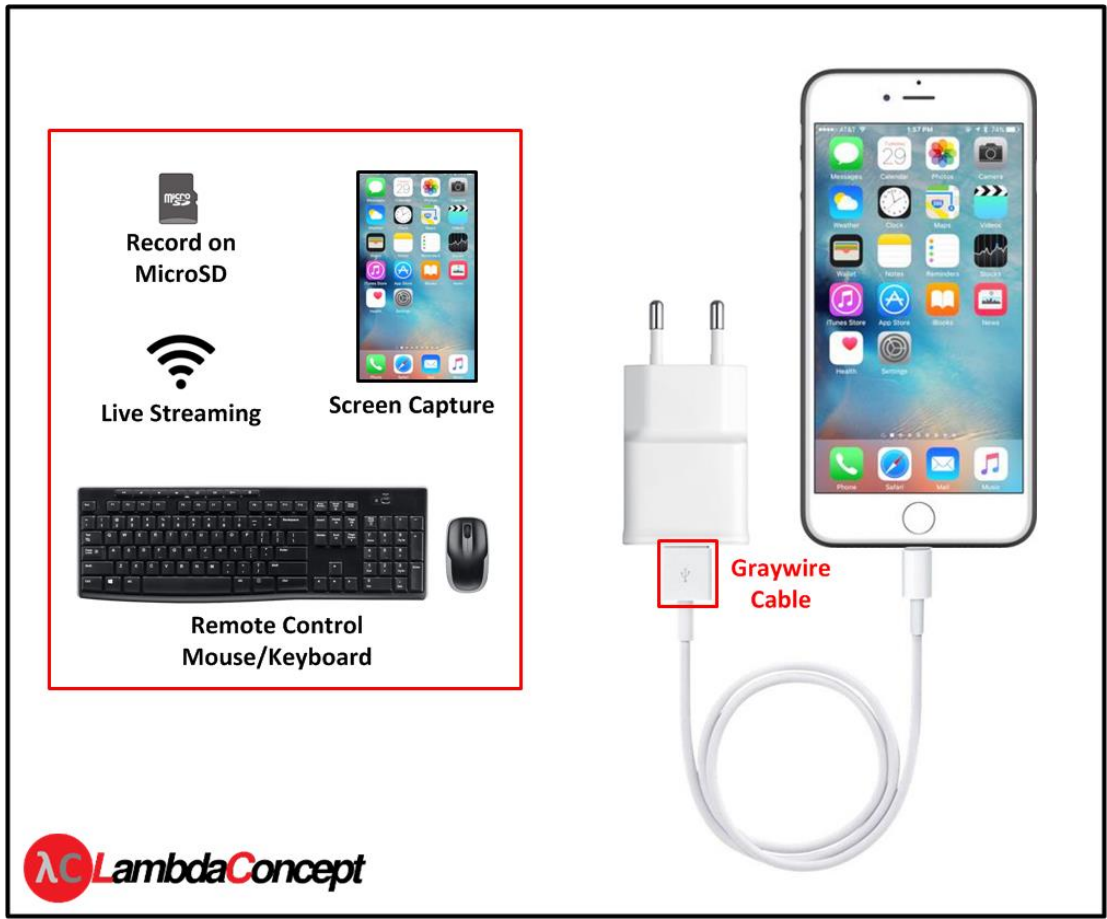
- Android
  - Class-compliant Network, Audio implementations
  - Open Accessory Protocols for Audio and custom functionality
  - Debugging and socket muxing via *Android Debug Bridge* (ADB)

Attack Surface: USB (High-Level)

- Juice-jacking attacks
  - Seemingly harmless charger or charging cable
  - Actually acts as an accessory or computer to the smartphone

- Early attacks (malicious USB host):
  - Extract files, Install apps
  - Fixed by user confirmation prompts

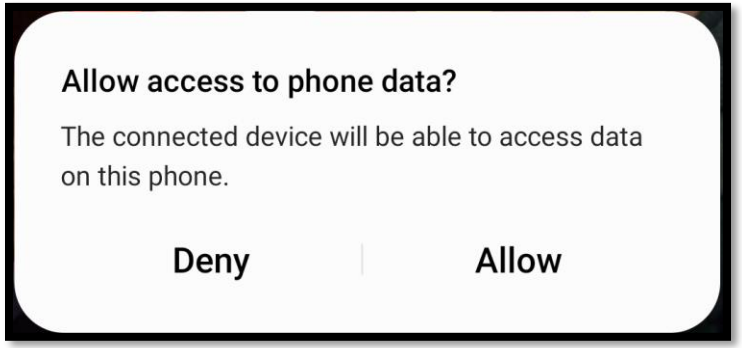
- Recent attacks (malicious USB device):
  - Record screen contents
  - Trigger voice commands
  - Inject user input

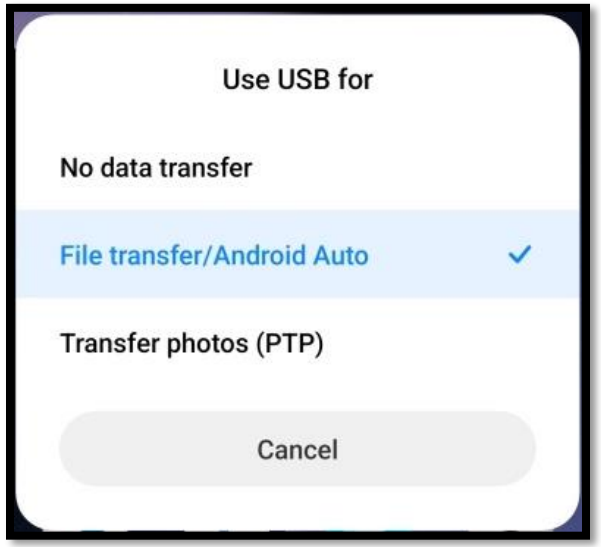


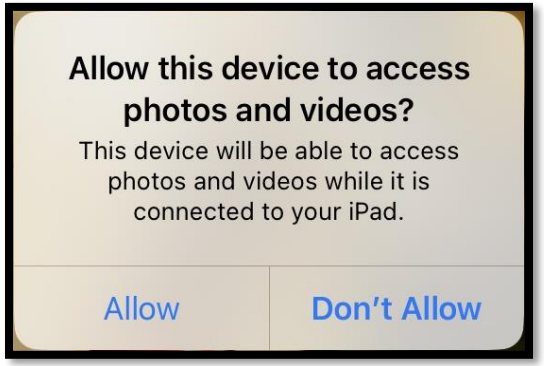
Source: http://blog.lambdaconcept.com/post/2019-09/graywire/

Attack Surface: USB (High-Level)

- Juice-Jacking fixed with user prompts?
- Recent study by ISEC: Attacks are still possible!

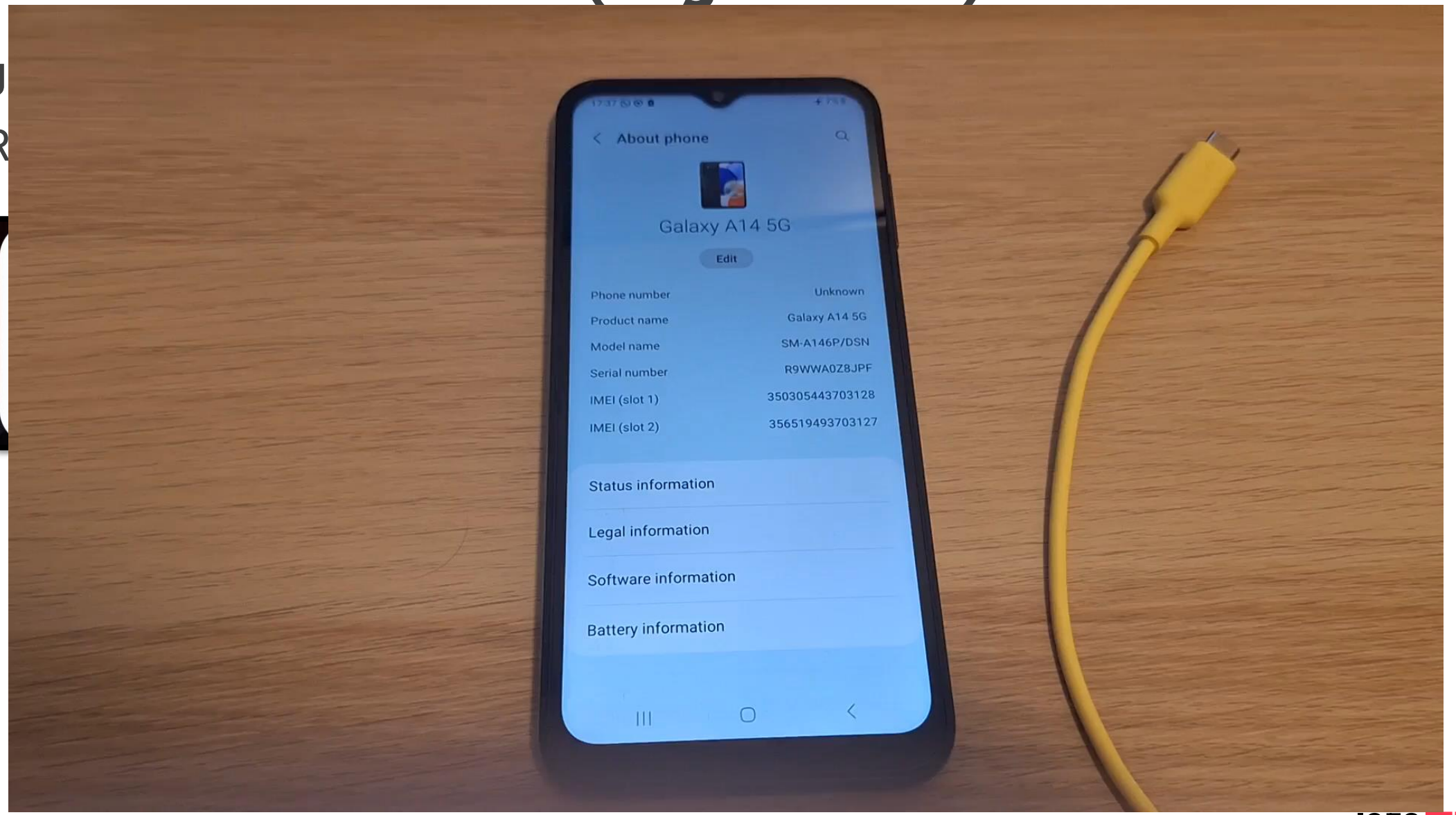






Attack Surface: USB (High-Level)

- J
- R



Attack Surface: USB (Low-Level)

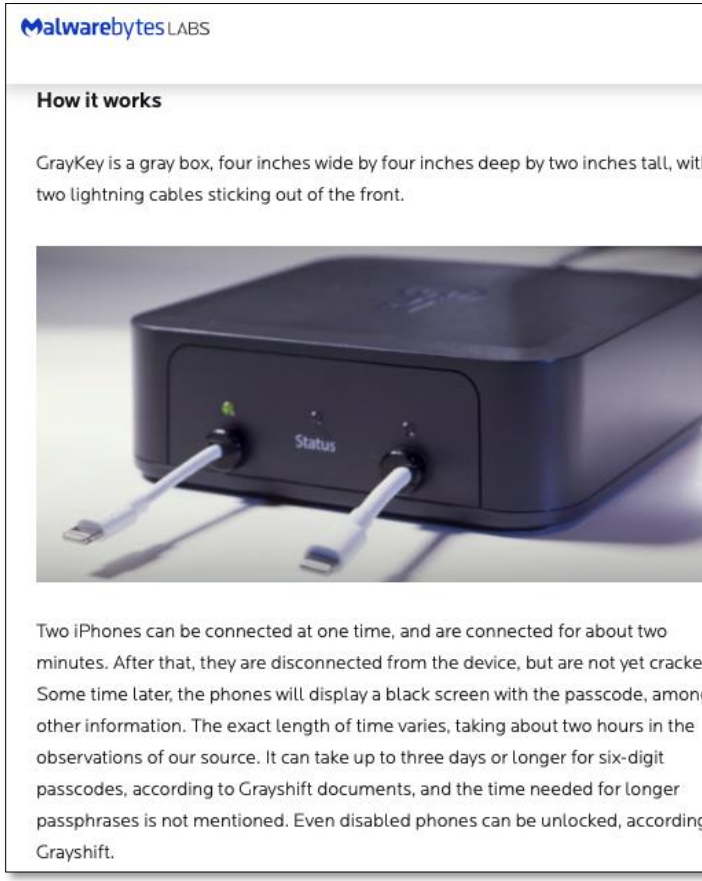
Commercial solutions allow file extraction from locked devices

- Exploiting a collection of low-level vulnerabilities

- Magnet GrayKey Box
  - Brute-force pin and extract data from locked iOS device
  - iOS 9.x-18.x, Android 10-13 magnetforensics.com

- Cellebrite Inseyets / UFED cellebrite.com
  - Supports iOS and Android, exact versions unknown

- MSAB XRY
  - Few details available msab.com



Malwarebytes LABS

How it works

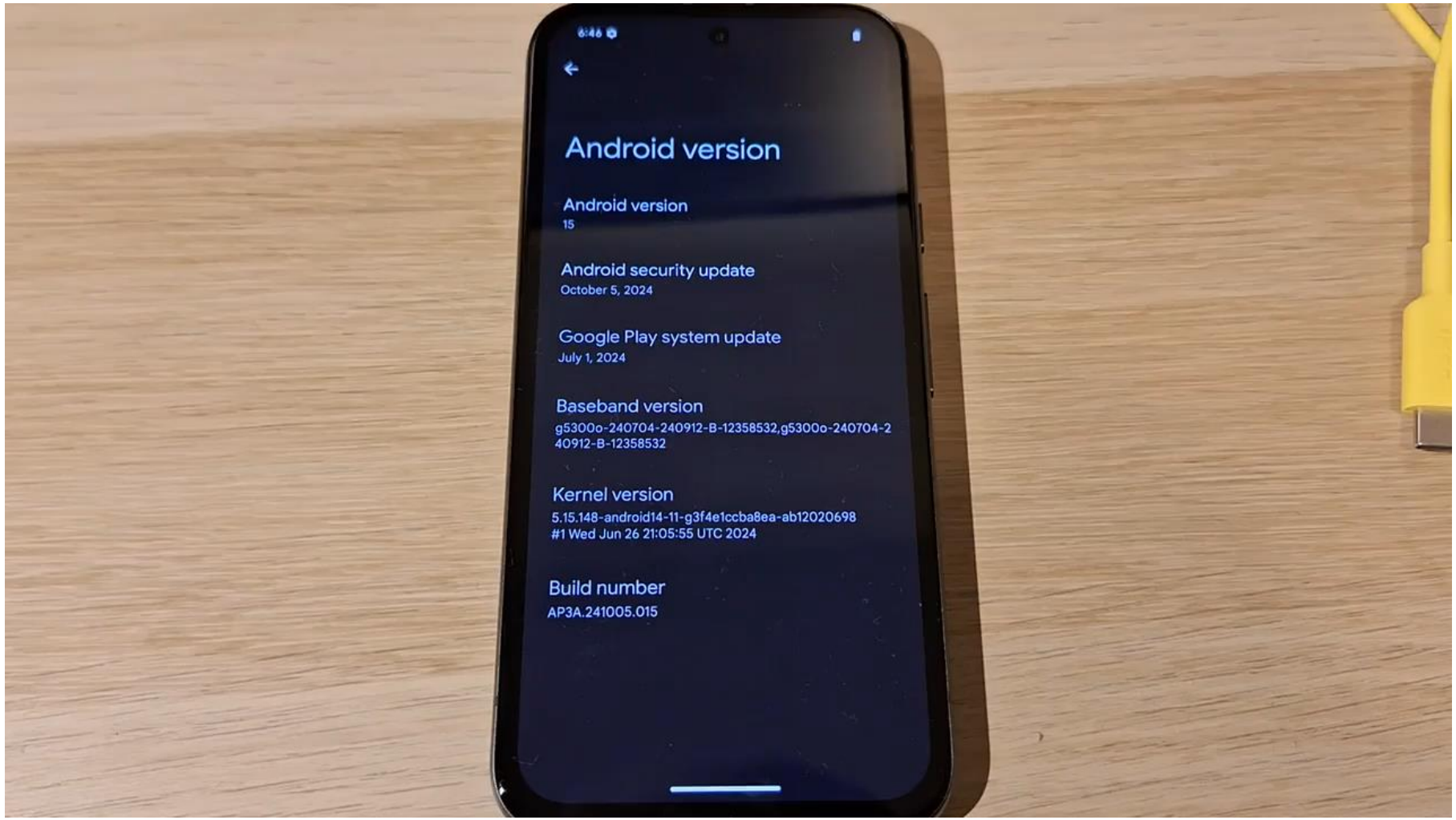
GrayKey is a gray box, four inches wide by four inches deep by two inches tall, wit
two lightning cables sticking out of the front.

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, accordin
Grayshift.

Source: malwarebytes.com

Attack Surface: USB (Low-Level)

- Demo: USB-based file access on locked devices



Attack Surface: USB (Low-Level)

- Cellebrite UFED has been investigated recently
  - Was used for spying on activists in Serbia
  - Cellebrite allowed installing persistent spyware

- Google Project Zero analysed the attack:
  1. Cellebrite UFED used to unlock phone
  2. Manually installed third-party app
  3. App exploited vulnerability in kernel driver
     - Some drivers are accessible to apps
  4. Installed privileged persistent spyware

Source: amnesty.com

Attack Surface: USB (Low-Level)

- 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
  - Same code in iOS and BootROM

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

Attack Surface: Internet Connection

- A considerable portion of apps misconfigure TLS
  - Trust any server certificate
  - Don’t check certificate subject

- Consequence:
  - MITM attacks may extract e.g. credentials, credit card data, …

- Additionally: Most apps do not use certificate pinning
  - State-scale actors may still intercept traffic!

Attack Surface: 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

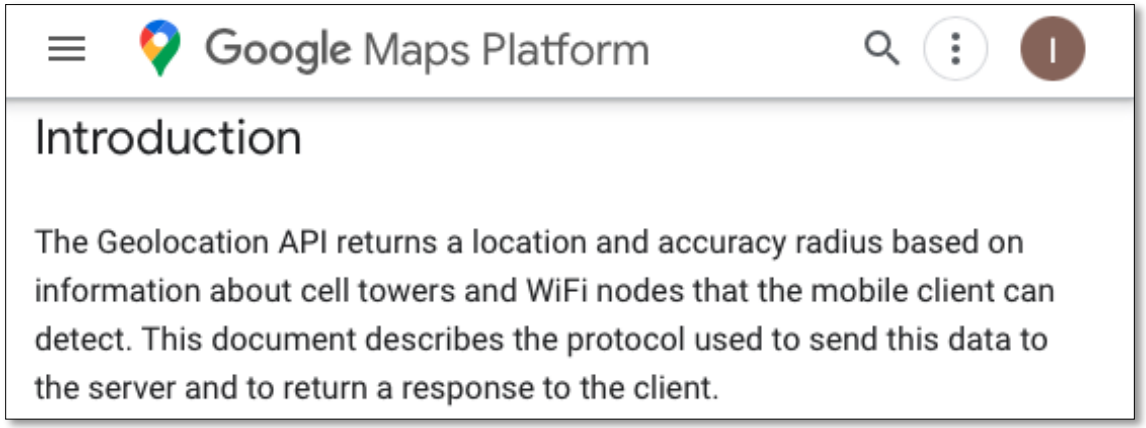
```
cat /etc/system/gps.conf | grep SUPL_HOST (or /vendor/etc/gps.conf)
SUPL_HOST=supl.google.com # Google
SUPL_HOST=supl.sonyericsson.com # Sony
SUPL_HOST=supl.qxwz.com # China(?)
...
```

Good: TLS is used to protect transfer

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

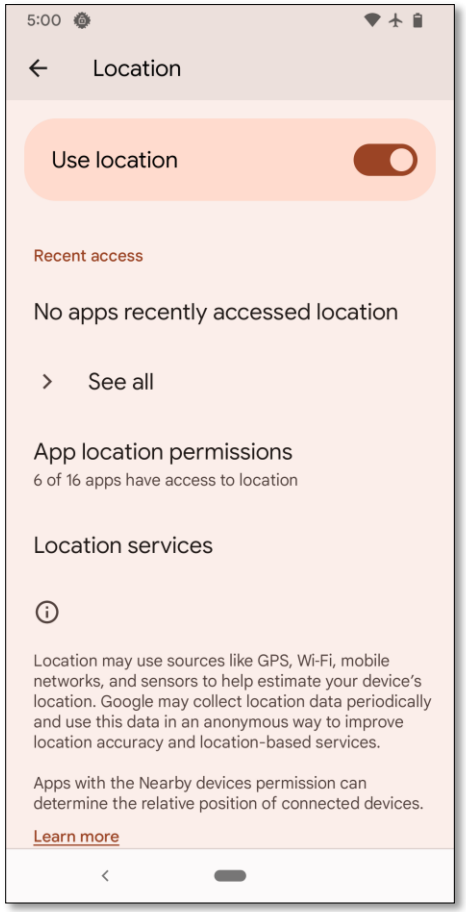
Attack Surface: Location

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



Source: developers.google.com

How do they learn this mapping?



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

Attack Surface: Apps

Potentially malicious developers can get code execution and escalate from there

- Psychic Papers
  - iOS apps could get arbitrary entitlements due to XML parser bugs
- DirtyCOW
  - Linux race condition in COW that allows to gain temporary root access
- macDirtyCOW
  - Similar more recent vulnerability allows temporary system modification on iOS
- Vulnerabilities in Kernel Drivers
  - These still come from vendors, and sometimes are accessible to untrusted apps

Additional Challenges

Smartphone - Threats

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

Only in a very limited way!

- Many parts of Android and iOS were implemented specifically for them

- Only a handful of security experts on teams
  - No chance to *review* every single code line!
  - Help *design* features with security in mind

Smartphone - Challenges

- New technologies in combination with old ones
  - E.g. Linux as basis + key storage in hardware
- Mixed private / business use cases
  - How to properly 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

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

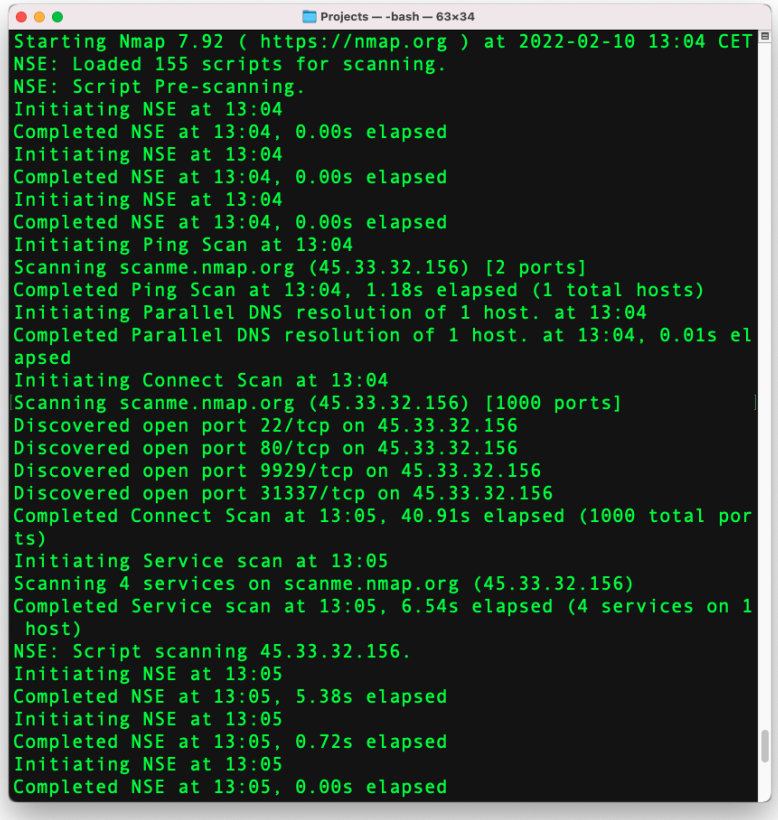
Google tracks you even if you turn off 'location history': report

IMAGE: JAAP ARRIENS/NURPHOTO VIA GETTY IMAGES

Source: mashable.com

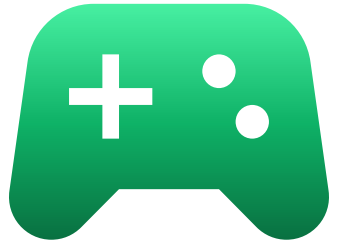
Mobility

- Install malware on smartphone on-the-fly
  - Steal it from a jacket, take it from a table, …
- Use it for attacks
  - Spy with its microphone, camera
  - Do ARP Spoofing / MITM in WiFis
  - Scan networks
  - Open a rogue access point



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

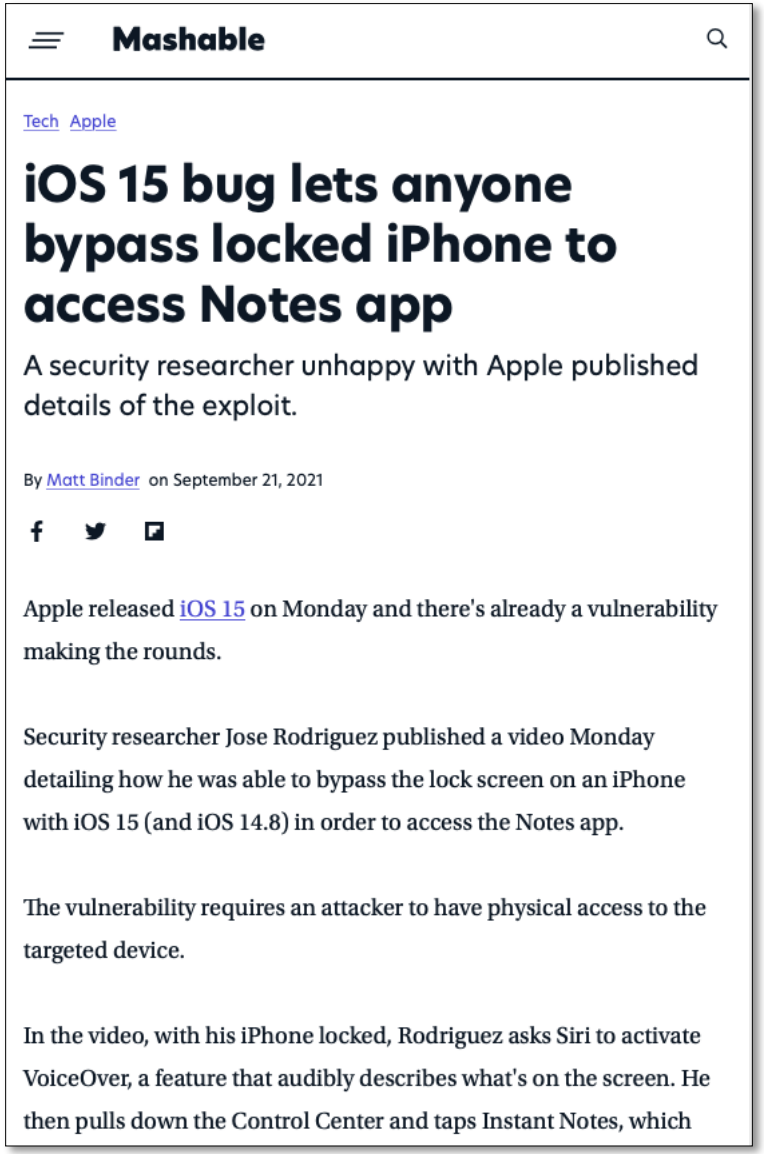
Smart phones need to be easily approachable!

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

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 – Screen Locks



Mashable

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

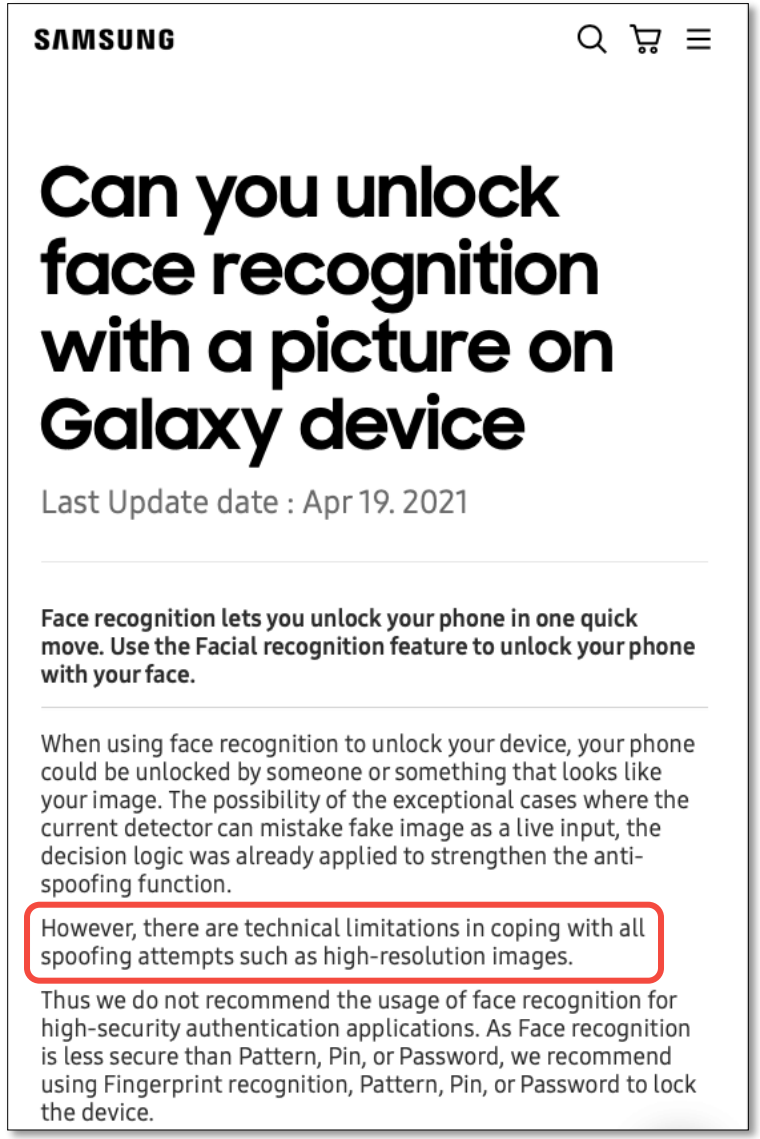
Apple released iOS 15 on Monday and there's already a vulnerability making the rounds.

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



SAMSUNG

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

Security vs. Efficiency

CENSUS
IT Security Works

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
[38] and Noise [05] protocol keys used for client-to-server encryption and end-to-end

Cookie Policy

Source: census-labs.com

Kritische Sicherheitslücke gefährdet Milliarden WhatsApp-Nutzer

Alert! 10.10.2018 10:43 Uhr – Jürgen Schmidt

ermöglicht es, ein Smartphone mit einem einzigen

troffen sind Milliarden WhatsApp-Nutzer.

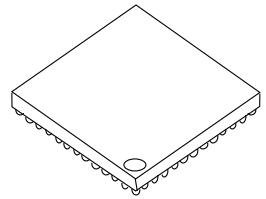
Source: https://goo.gl/3mEYGf

Security vs. Performance

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



Mobile Device Management (MDM)

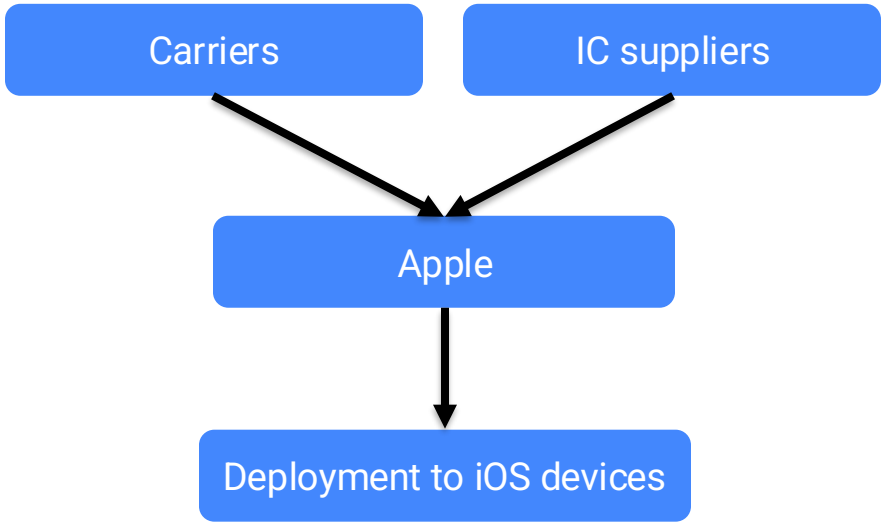
- **Deploy** security policies that the user cannot change
  - Password strength, encryption, applications, proxy, VPN, etc.
  - Forbid installation / removal of apps, limit bluetooth functionality, …
- **Get** information from device
  - Location, Call logs, SMS, Backups, …
- **Remote Actions**
  -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

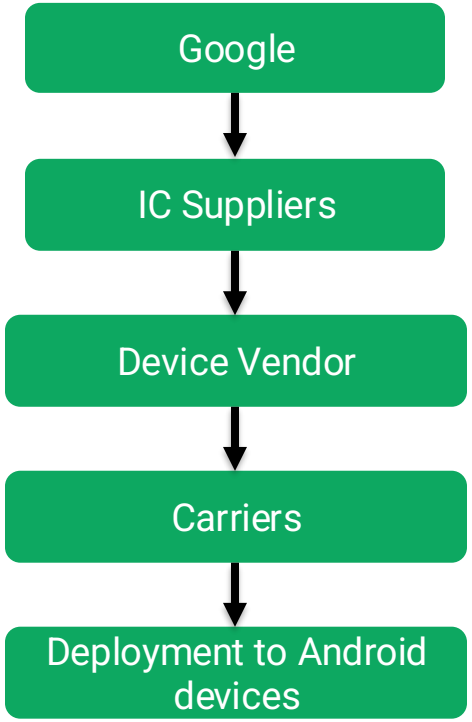
Challenge: Bring-your-own-device!

Updates

- Security updates are vital, especially in business environments

- Android: Slow update adoption
  - Improvements: Project Treble, Generic Kernel Image





Source: googleblog.com

Version Distributions (Q1/2025)

Source: Android Studio

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.4	KitKat	19	
5	Lollipop	21	99,7%
5.1	Lollipop	22	99,6%
6	Marshmallow	23	98,8%
7	Nougat	24	97,4%
7.1	Nougat	25	96,4%
8	Oreo	26	95,4%
8.1	Oreo	27	93,9%
9	Pie	28	89,6%
10	Q	29	81,2%
11	R (2020)	30	67,6%
12	S	31	48,6%
13	T	33	33,9%
14	U	34	13,0%

Last updated: May 1, 2024

Android 14: Released in September 2023

VS

iOS and iPadOS usage

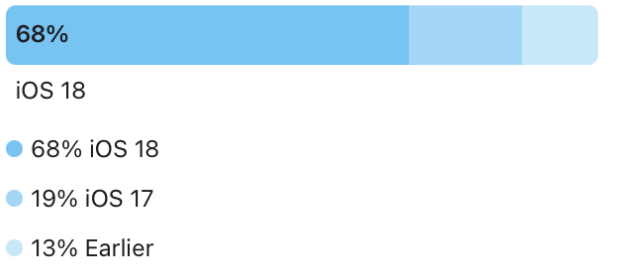
As measured by devices that transacted on the App Store on January 21, 2025.

iPhone

76% of all devices introduced in the last four years use iOS 18.

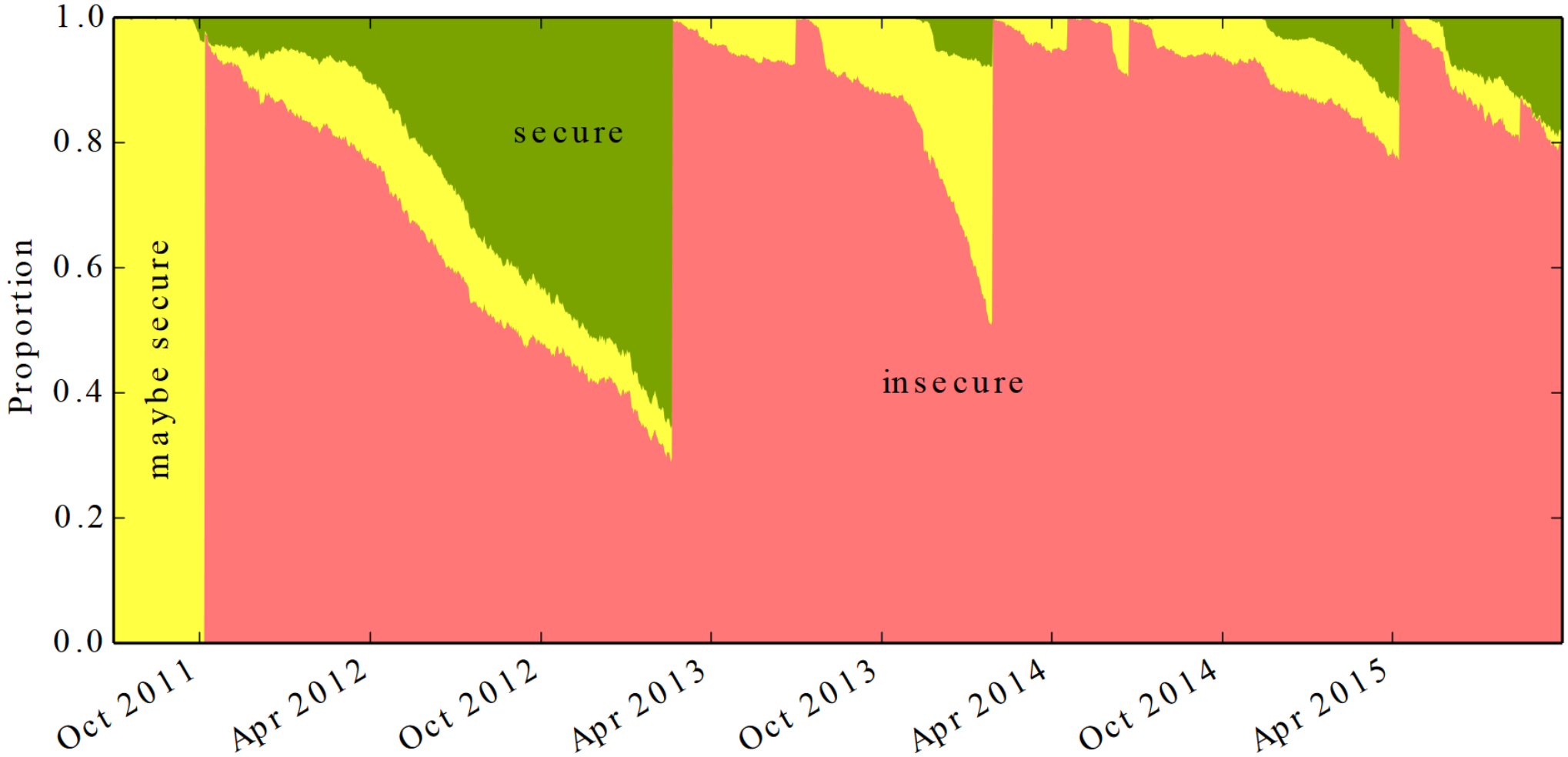
Source: apple.com

68% of all devices use iOS 18.



iOS 18: Released in September 2024

Vulnerable Android Devices



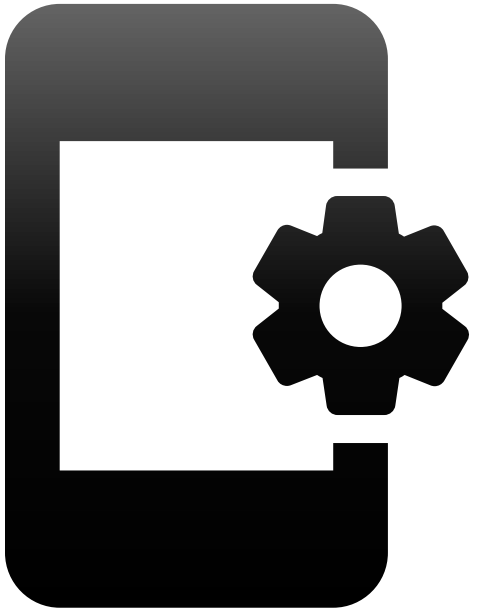
Source: androidvulnerabilities.org

Proprietary Technology

- iOS and Android split almost the entire industry among them
  - A single vulnerability may affect an incredibly large number of users
- Much of the technology is proprietary
  - iOS is closed source
  - All major Android vendors ship closed-source proprietary additions
- No choice but use device vendor's OS
  - Otherwise no full functionality (e.g. device attestation)
  - Forced to accept preinstalled adware and malware

Applications – OS Integration

- Access to APIs, Sensors, other Apps
  - Inter-Process Communication (IPC)
  - Android Permissions
  - How does the user know what purpose a permission serves?
- 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

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?

Outlook

Topics Mobile Security 2025

- iOS Platform Security
  - iBoot, SEP, Data Protection, Jailbreak
- iOS Application Security
  - Components, Permissions, Crypto, …
  - Sandbox, Signing, Malware, …
- Android Platform Security
  - Verified Boot, FBE, SELinux, Root, …
- Android Application Security I & II
  - Components, Permissions, Crypto, …
  - Sandbox, Signing, SafetyNet, …
- Mobile Hardware Security
  - IoT, Embedded, Interfaces, FW
- Mobile Network Security
  - GMS, 3G, 4G, Attacks
- Mobile Security Research
  - Approach, Bounties, Laws, News

Outlook

- 14.03.2025
  - Android Platform Security
  - Assignment 1 Details

- 21.03.2025
  - Android Application Security I