Motivation

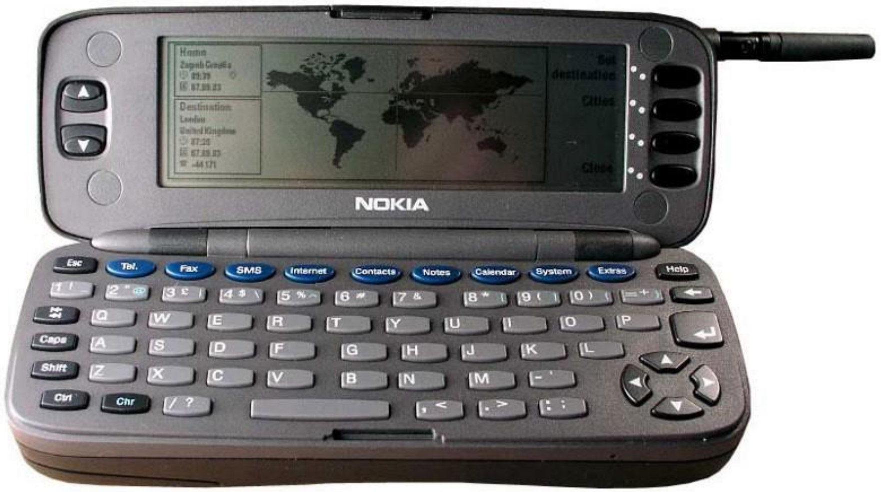
ACN / Mobile Security 2020

Johannes Feichtner
johannes.feichtner@iaik.tugraz.at

Smartphones – History

Once upon a time...

- PDA combined with a phone (starting in the late 90ies)
- Nokia Communicator (1996)
  - Combining Internet, Calendar, E-Mail, simple apps with phone
- Windows Mobile (2000)
- Proprietary lightweight OS
  - Usability?



Smartphones – Prototype

- BlackBerry starting in 2002
  - Research in Motion (RIM)
  - Push e-mail
  - Many security functions
  - Focus on business use → consumer area neglected for a long time
- Still sophisticated security functions but market share < 0.1% in Q1 / 2020
- App development easier on other platforms
- BlackBerry OS 10
  - Support for Android apps

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

Smartphone – History

- Apple iOS: One version now
- Android: 2.x smartphones, 3.x tablets → merge in 4.x
- BlackBerry: PlayBook

Focus of this lecture: Android (87% share) & iOS (13% share)

– Similar considerations regarding security

– Some differences: area of use, functionality

IDC
Q1 / 2020

Various OS platforms

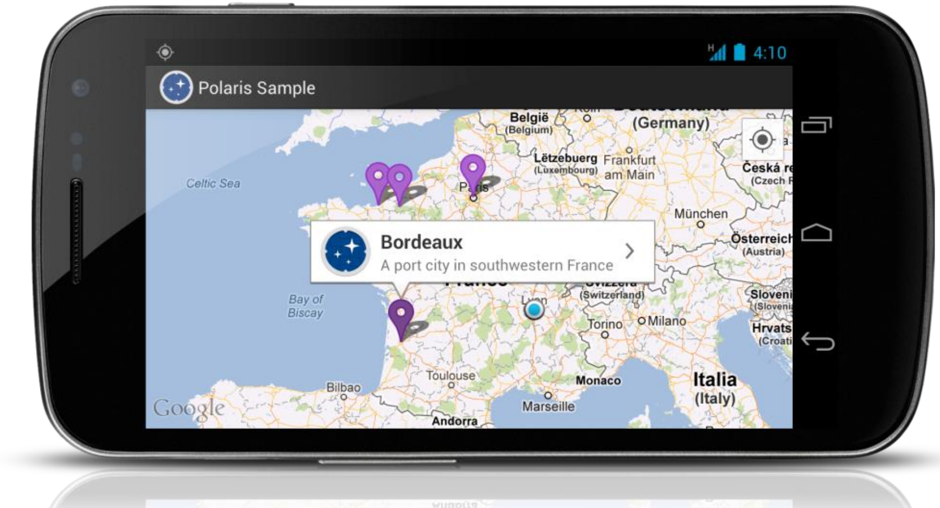
	iOS	Android	~~Firefox OS~~	~~Windows 10 Mobile~~	~~BlackBerry~~	Tizen	Sailfish OS	~~Ubuntu Touch~~
Company	Apple	Google	~~Mozilla~~	~~Microsoft~~	~~BlackBerry Ltd.~~	Samsung	Sailfish Alliance	~~Canonical Ltd.~~
Version	13.3	10.0	~~2.6~~	~~10 (1709)~~	~~10.3.3~~	4.0	3.2.1	~~16.04~~
Status	Proprietary	Free	~~Free~~	~~Proprietary~~	~~Proprietary~~	Free	Free	~~Free~~
OS Family	Darwin	Linux	~~Linux~~	~~Windows~~	~~QNX~~	Linux	Linux	~~Linux~~

Applications

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

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

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



Applications

- Business: e-mail, container apps, SAP, health-related 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)
- Spyware: „spy on your wife / husband / children“
  - Record audio, take photos, send location remote commands

- 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 (+ thanks to sensors)
- Malware evolution (compare to PCs)
  - Same protection mechanisms adequate?!
- Mobile stolen → Identity stolen?!
- Increased business use (shifting from BlackBerry)

Threats?

Now you know the possibilities but…

…what are the threats?

Smartphone - Threats

- Companies do 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, https://goo.gl/GjGuAY

Smartphone - Threats

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

- Distributed eco-system

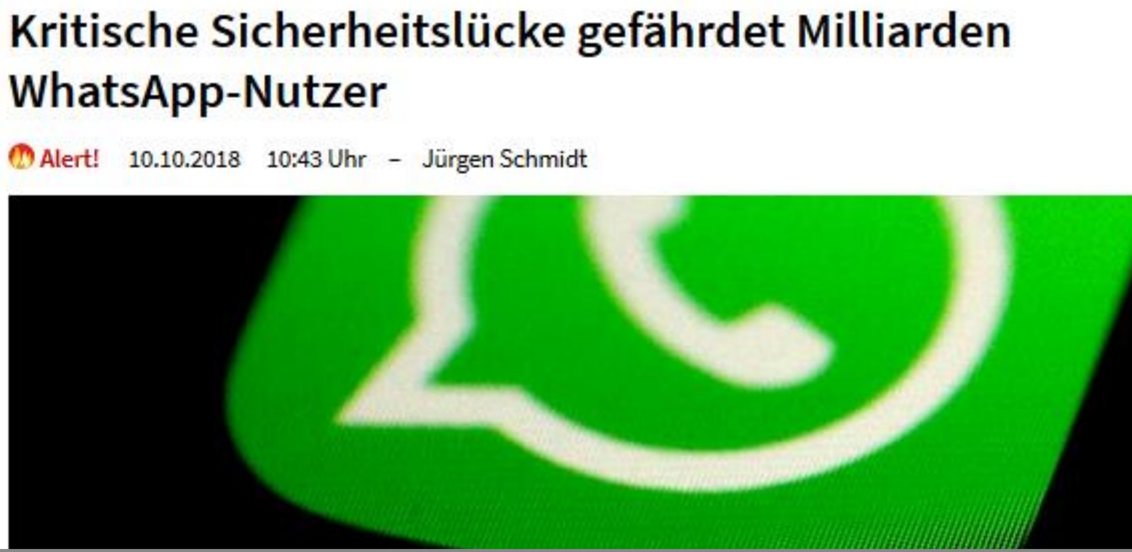
- Mixed private / business use cases
  - How to separate these two spheres?
  - Limited administrative access to devices

- Legacy security strategies are ineffective
  - Innovation outpaces security practices

New Mix of Technologies

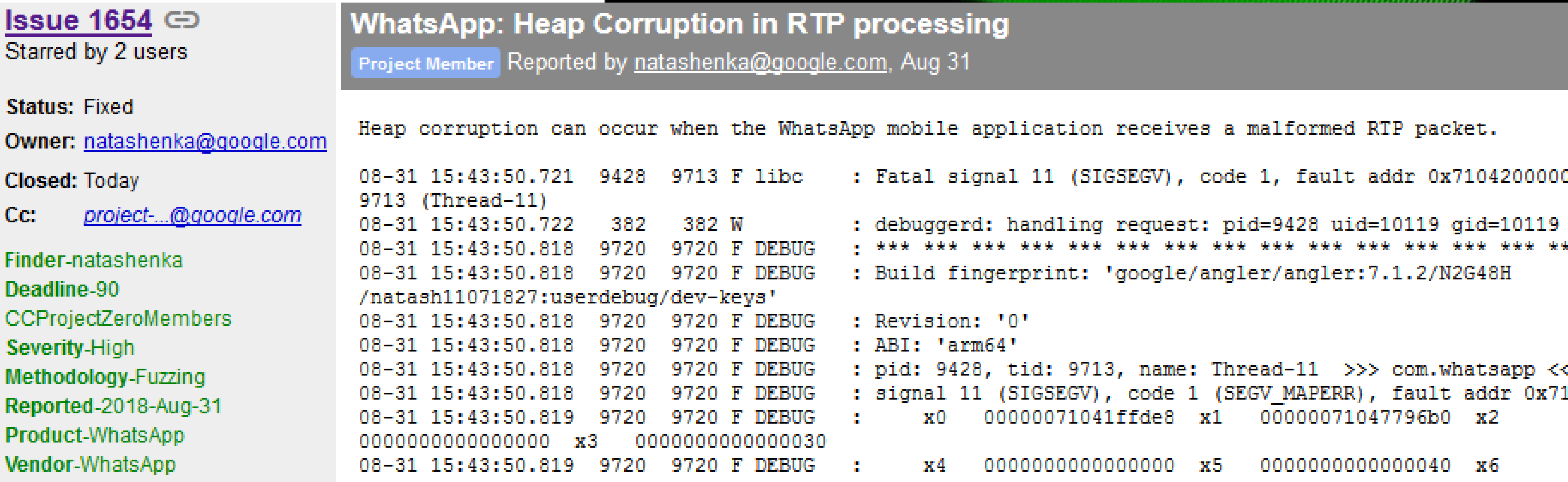
- Permanent Internet connection
  - UMTS / LTE, WiFi
- Telephone
  - SMS / MMS
  - Bluetooth

- Sensors
  - Microphone,
  - A-GPS,
  - Light Sensor,
  - Gyroscope,
  - ...



Kritische Sicherheitslücke gefährdet Milliarden WhatsApp-Nutzer

Alert! 10.10.2018 10:43 Uhr – Jürgen Schmidt



Issue 1654
Starred by 2 users

Status: Fixed
Owner: natashenka@google.com
Closed: Today
Cc: project-...@google.com

Finder-natashenka
Deadline-90
CCProjectZeroMembers
Severity-High
Methodology-Fuzzing
Reported-2018-Aug-31
Product-WhatsApp
Vendor-WhatsApp

WhatsApp: Heap Corruption in RTP processing

Project Member Reported by natashenka@google.com, Aug 31

```
Heap corruption can occur when the WhatsApp mobile application receives a malformed RTP packet.

08-31 15:43:50.721  9428  9713 F libc    : Fatal signal 11 (SIGSEGV), code 1, fault addr 0x710420000
9713 (Thread-11)
08-31 15:43:50.722   382   382 W         : debuggerd: handling request: pid=9428 uid=10119 gid=10119
08-31 15:43:50.818  9720  9720 F DEBUG   : *** *** *** *** *** *** *** *** *** *** *** *** *** *** **
08-31 15:43:50.818  9720  9720 F DEBUG   : Build fingerprint: 'google/angler/angler:7.1.2/N2G48H
/natash11071827:userdebug/dev-keys'
08-31 15:43:50.818  9720  9720 F DEBUG   : Revision: '0'
08-31 15:43:50.818  9720  9720 F DEBUG   : ABI: 'arm64'
08-31 15:43:50.818  9720  9720 F DEBUG   : pid: 9428, tid: 9713, name: Thread-11  >>> com.whatsapp <<
08-31 15:43:50.818  9720  9720 F DEBUG   : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x71
08-31 15:43:50.819  9720  9720 F DEBUG   :     x0   00000071041ffde8  x1   00000071047796b0  x2
0000000000000000  x3   0000000000000030
08-31 15:43:50.819  9720  9720 F DEBUG   :     x4   0000000000000000  x5   0000000000000040  x6
```

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

New Mix of Technologies

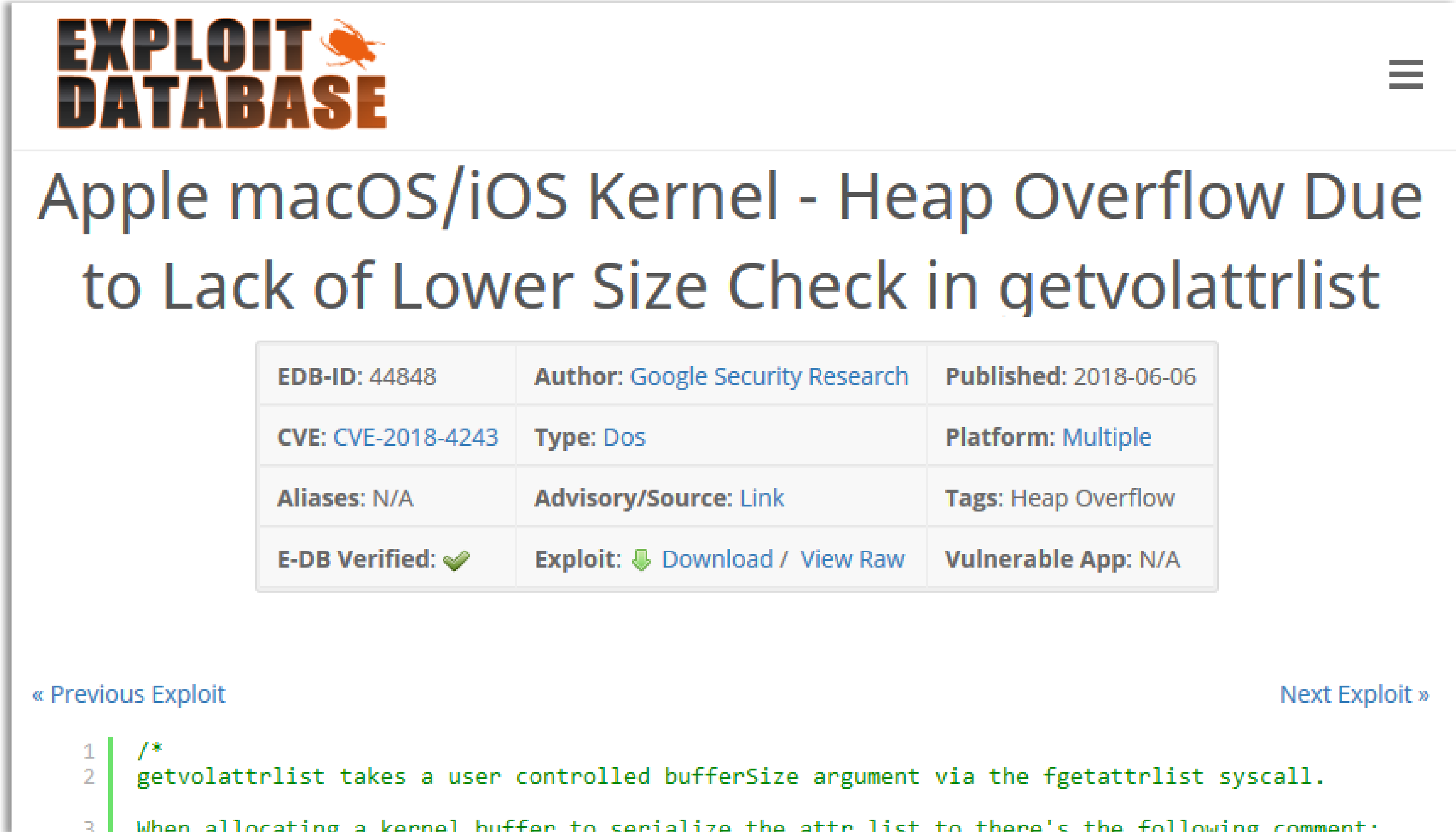
Shared OS & parts of it → shared security aspects!

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

- iOS (Darwin OS X)
  → watchOS, tvOS

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

See: https://goo.gl/yk8DWW



EXPLOIT DATABASE

Apple macOS/iOS Kernel - Heap Overflow Due to Lack of Lower Size Check in getvolattrlist

EDB-ID: 44848	**Author:** Google Security Research	**Published:** 2018-06-06
CVE: CVE-2018-4243	**Type:** Dos	**Platform:** Multiple
Aliases: N/A	**Advisory/Source:** Link	**Tags:** Heap Overflow
E-DB Verified: ✓	**Exploit:** Download / View Raw	**Vulnerable App:** N/A

« Previous Exploit

Next Exploit »

```
/*
getvolattrlist takes a user controlled bufferSize argument via the fgetattrlist syscall.

When allocating a kernel buffer to serialize the attr list to there's the following comment:
```

Data & Sensors

- Location
  - Network Cell ID (coarse)
  - GPS (fine)
    - Usually used with A-GPS for faster 3D fix

- „Articles of value“
  - 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, …

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

Stored in the cloud?

…

Mobility

- Smartphone is taken everywhere
  – Collecting data even if not actively used

- Always online
  – Google‘s location history with location turned off

- There is so much information in e-mails…

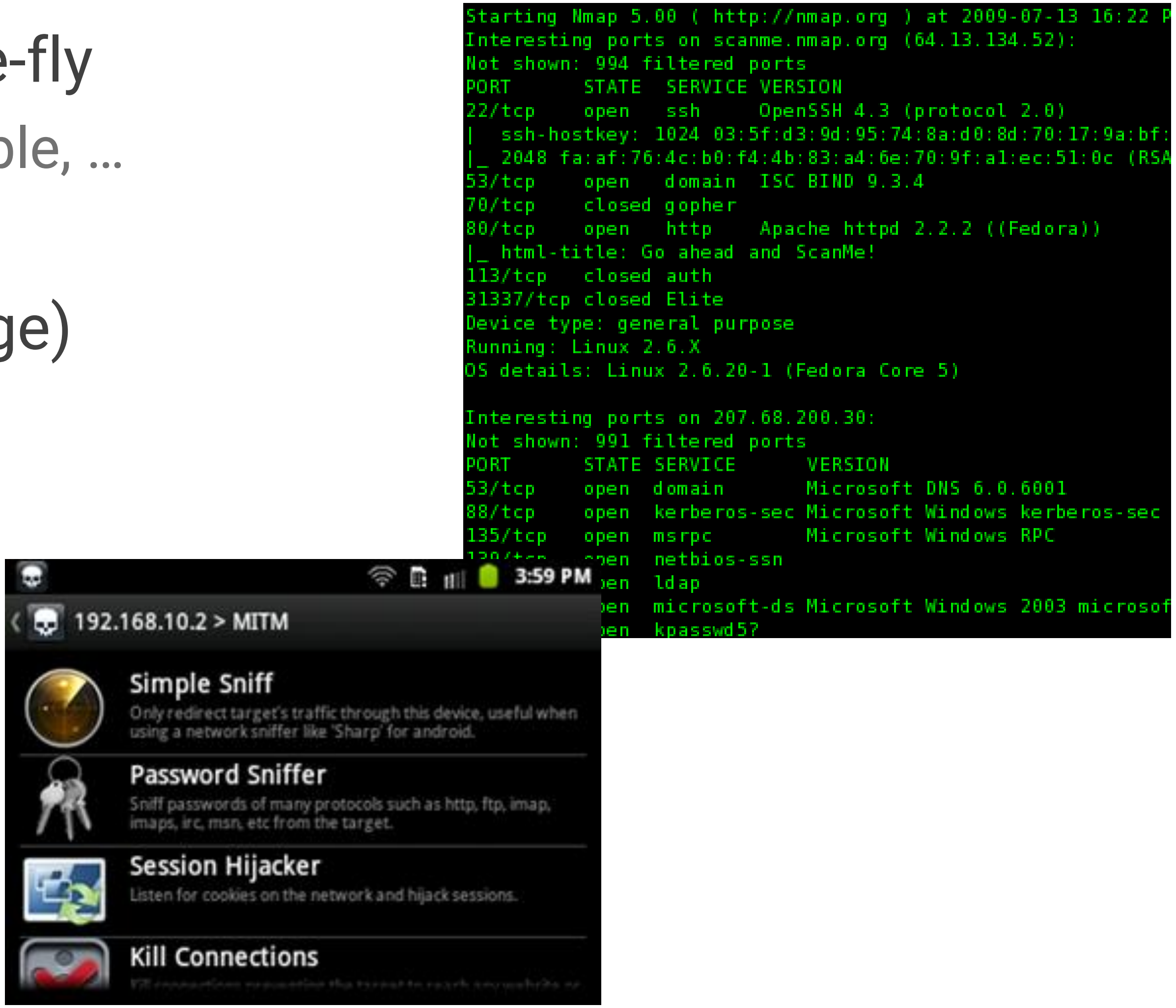
- Business, vacation, beer after work, sports, …

Header: Timestamp, Server chain, AV Checking, DKIM, Internal & External IP addresses
Body: Your personal message

How to protect critical data in such environments?!

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
  - Capture WPA handshakes



```
Starting Nmap 5.00 ( http://nmap.org ) at 2009-07-13 16:22 P
Interesting ports on scanme.nmap.org (64.13.134.52):
Not shown: 994 filtered ports
PORT      STATE  SERVICE VERSION
22/tcp    open   ssh     OpenSSH 4.3 (protocol 2.0)
|  ssh-hostkey: 1024 03:5f:d3:9d:95:74:8a:d0:8d:70:17:9a:bf:
|_ 2048 fa:af:76:4c:b0:f4:4b:83:a4:6e:70:9f:a1:ec:51:0c (RSA
53/tcp    open   domain  ISC BIND 9.3.4
70/tcp    closed gopher
80/tcp    open   http    Apache httpd 2.2.2 ((Fedora))
|_ html-title: Go ahead and ScanMe!
113/tcp   closed auth
31337/tcp closed Elite
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.20-1 (Fedora Core 5)

Interesting ports on 207.68.200.30:
Not shown: 991 filtered ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Microsoft DNS 6.0.6001
88/tcp    open  kerberos-sec Microsoft Windows kerberos-sec
135/tcp   open  msrpc        Microsoft Windows RPC
          open  netbios-ssn
           en   ldap
           en   microsoft-ds Microsoft Windows 2003 microsof
           en   kpasswd5?
```

Simple Sniff
Only redirect target's traffic through this device, useful when using a network sniffer like 'Sharp' for android.

Password Sniffer
Sniff passwords of many protocols such as http, ftp, imap, imaps, irc, msn, etc from the target.

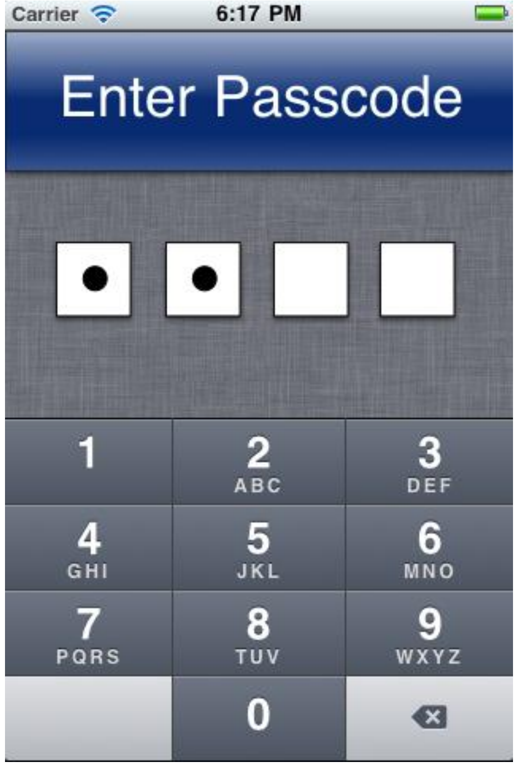
Session Hijacker
Listen for cookies on the network and hijack sessions.

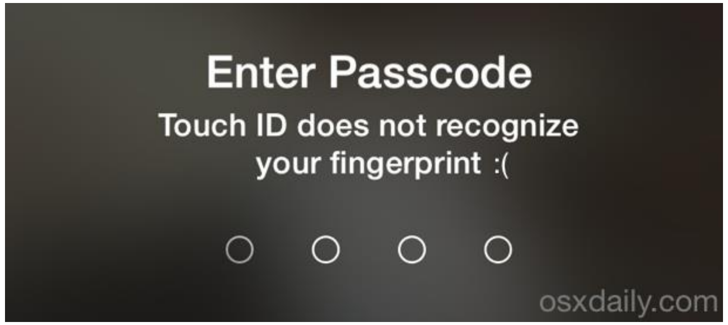
Kill Connections

Security Functions

No one wants to enter a passphrase to access the phone!

- PIN codes, short passwords, screen unlock patterns
- Security updates take their time
  - Often entire OS images are transmitted
- Device administration, management, …
- Traditional security policies do not cover smartphones





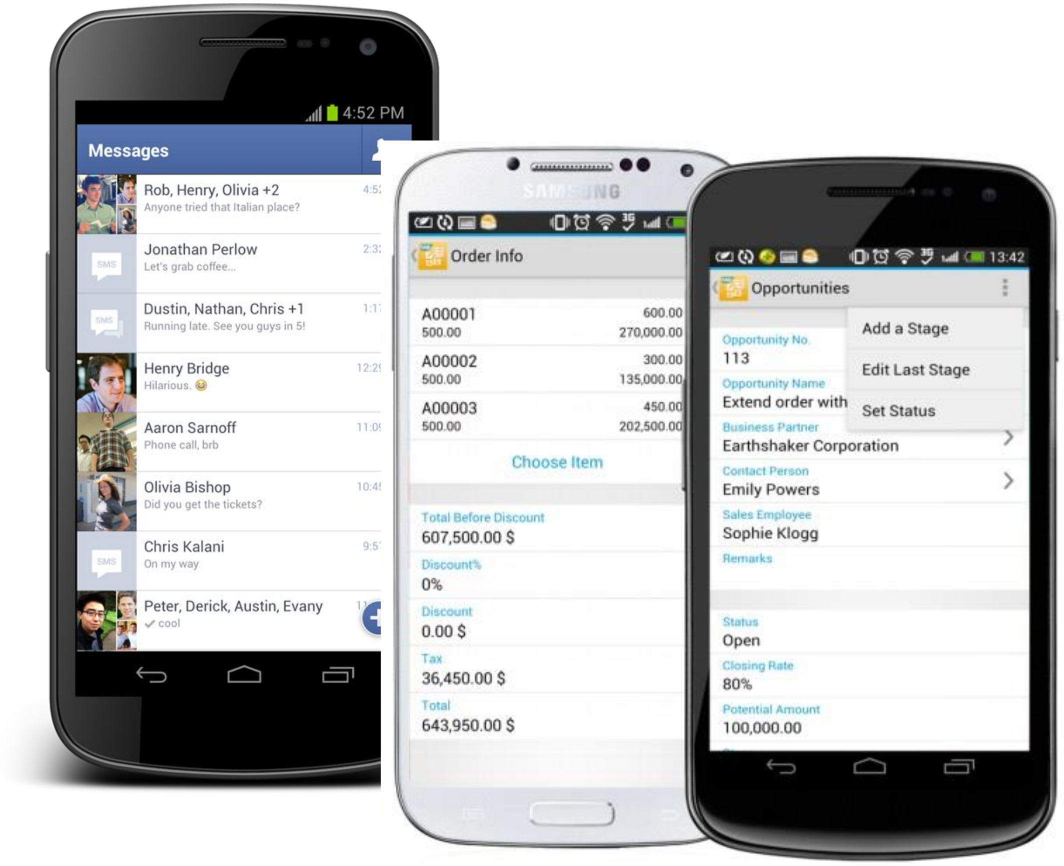
Enter Passcode

Touch ID does not recognize your fingerprint :(

osxdaily.com

Business vs. Private Use

- Complete mixture of two areas
- Used in same environments
- Applications
    - Facebook vs. SAP
- Private area (Spyware?)
- BYOD – Bring your own device



4:52 PM

Messages

Rob, Henry, Olivia +2 — Anyone tried that Italian place? — 4:5

Jonathan Perlow — Let's grab coffee... — 2:3

Dustin, Nathan, Chris +1 — Running late. See you guys in 5! — 1:1

Henry Bridge — Hilarious. 😀 — 12:2

Aaron Sarnoff — Phone call, brb — 11:0

Olivia Bishop — Did you get the tickets? — 10:4

Chris Kalani — On my way — 9:5

Peter, Derick, Austin, Evany — ✓ cool

Order Info

A00001 500.00	600.00 270,000.00
A00002 500.00	300.00 135,000.00
A00003 500.00	450.00 202,500.00

Choose Item

Total Before Discount: 607,500.00 $

Discount%: 0%

Discount: 0.00 $

Tax: 36,450.00 $

Total: 643,950.00 $

13:42

Opportunities

Add a Stage

Edit Last Stage

Set Status

Opportunity No.: 113

Opportunity Name: Extend order with

Business Partner: Earthshaker Corporation

Contact Person: Emily Powers

Sales Employee: Sophie Klogg

Remarks

Status: Open

Closing Rate: 80%

Potential Amount: 100,000.00

CHALLENGE 1: Broader mobile attack surface

THE DEVICE

BROWSER

- Phishing
- Framing
- Clickjacking
- Man-in-the-Middle
- Buffer overflow
- Data caching

SYSTEM

- No/Weak passcode
- Android rooting/iOS jailbreak
- OS data caching
- Passwords & data accessible
- Carrier-loaded software
- No/Weak encryption
- User-initiated code
- Confused deputy attack
- TEE/Secure Enclave Processor
- Side channel leak
- Multimedia/file format parsers
- Kernel driver vulnerabilities
- Resource DoS
- GPS spoofing
- Device lockout

PHONE/SMS

- Baseband attacks
- SMS phishing

APPS

- Sensitive data storage
- No/Weak encryption
- Improper SSL validation
- Configuration manipulation
- Dynamic runtime injection
- Unintended permissions
- Escalated privileges
- UI overlay/pin stealing
- Third-party code
- Intent hijacking
- Zip directory traversal
- Clipboard data
- URL schemes
- GPS spoofing
- Weak/No Local authentication
- Integrity/tampering/repacking
- Side channel attacks
- App signing key unprotected
- App transport security
- XML serialization
- JSON-RPC
- SQLite database

⚠ MALWARE

THE NETWORK

- Wi-Fi (no/weak encryption)
- Rogue access point
- Packet sniffing
- Man-in-the-middle
- Session hijacking
- DNS poisoning
- SSL Strip
- Fake SSL certificate
- Baseband
- Wifi (chip/firmware attack)
- BGP hijacking
- IMSI-catcher
- LTE
- HTTP Proxies
- VPNs

CLOUD / DATA CENTER

WEB SERVER

- Platform vulnerabilities
- Server misconfiguration
- Cross-site scripting
- Cross-site request forgery
- Weak input validation
- Cross origin resource sharing
- Brute force attacks
- Side channel attacks
- Hypervisor attack
- VPN

DATABASE

- SQL injection
- Privilege escalation
- Data dumping
- OS command execution

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, …*

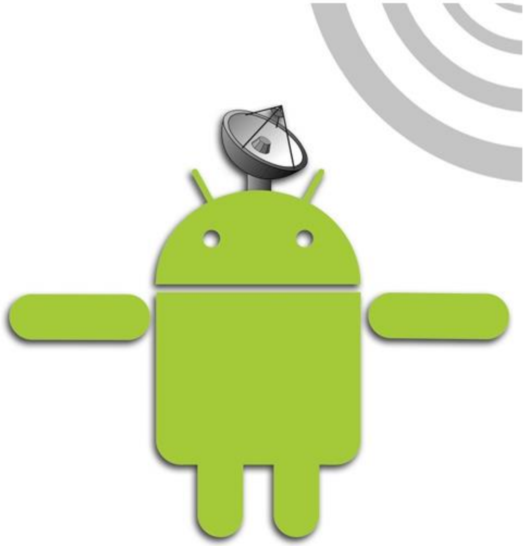
Define your threats: *Theft? Simple attacks? Sophisticated attacks?*

→ Analyze only the relevant security functions

→ Focus on important things (not sophisticated attacks)

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

Applications – Context

- Security software or spyware?
  - Remote wipe, remote commands, remote cam, …
  - Catch and relay SMS messages
    - WhatsApp, Signal, Telegram, …

- Availability of such apps depends on OS APIs and market

- What makes them bad? Where do you draw the line?

Applications – Roles

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

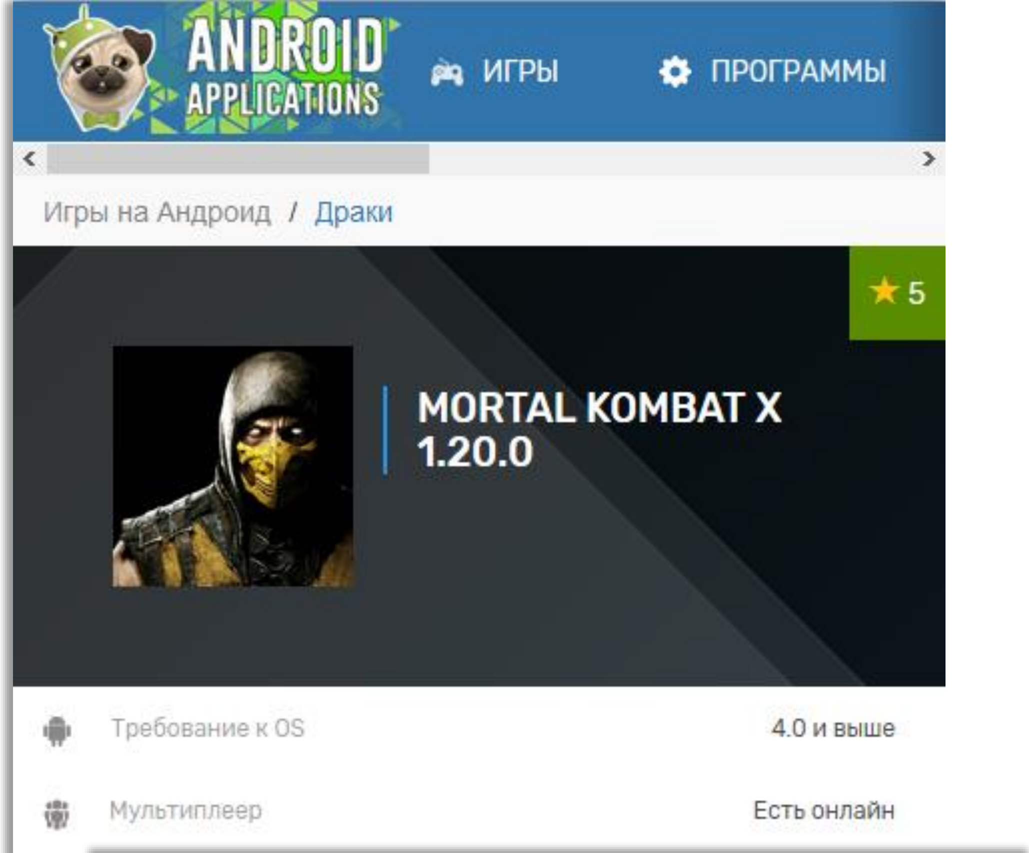
Too narrow-minded?!

- 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 iTunes
  - BlackBerry World
  - …
- App Stores: Either walled garden or open
  - Especially critical: 3rd party app stores!
- Other sources: Direct URLs, e-mails, storage, …
  - Malware potential?

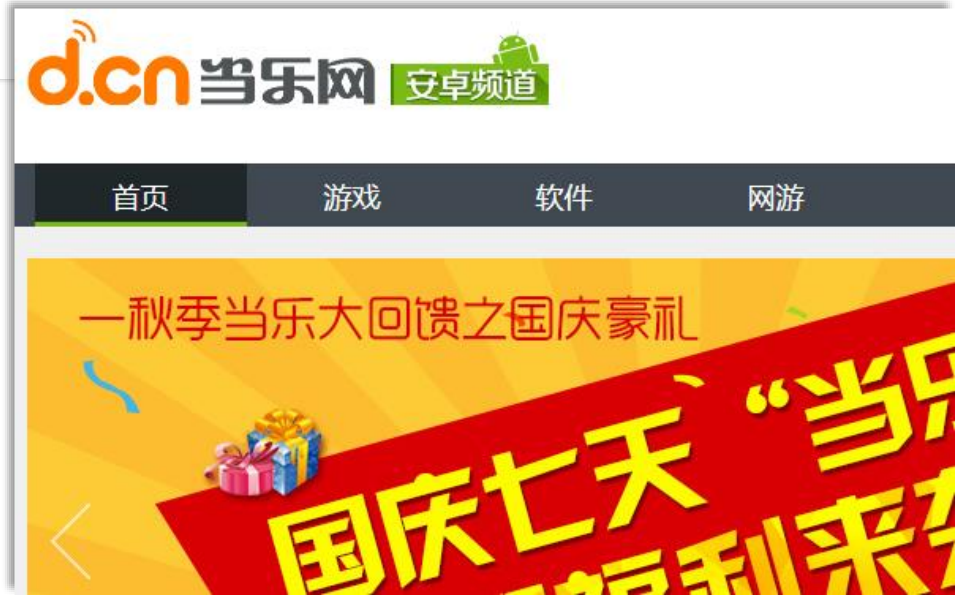


Игры на Андроид / Драки

MORTAL KOMBAT X 1.20.0

Требование к OS 4.0 и выше

Мультиплеер Есть онлайн



Applications – Sources

Walled Garden

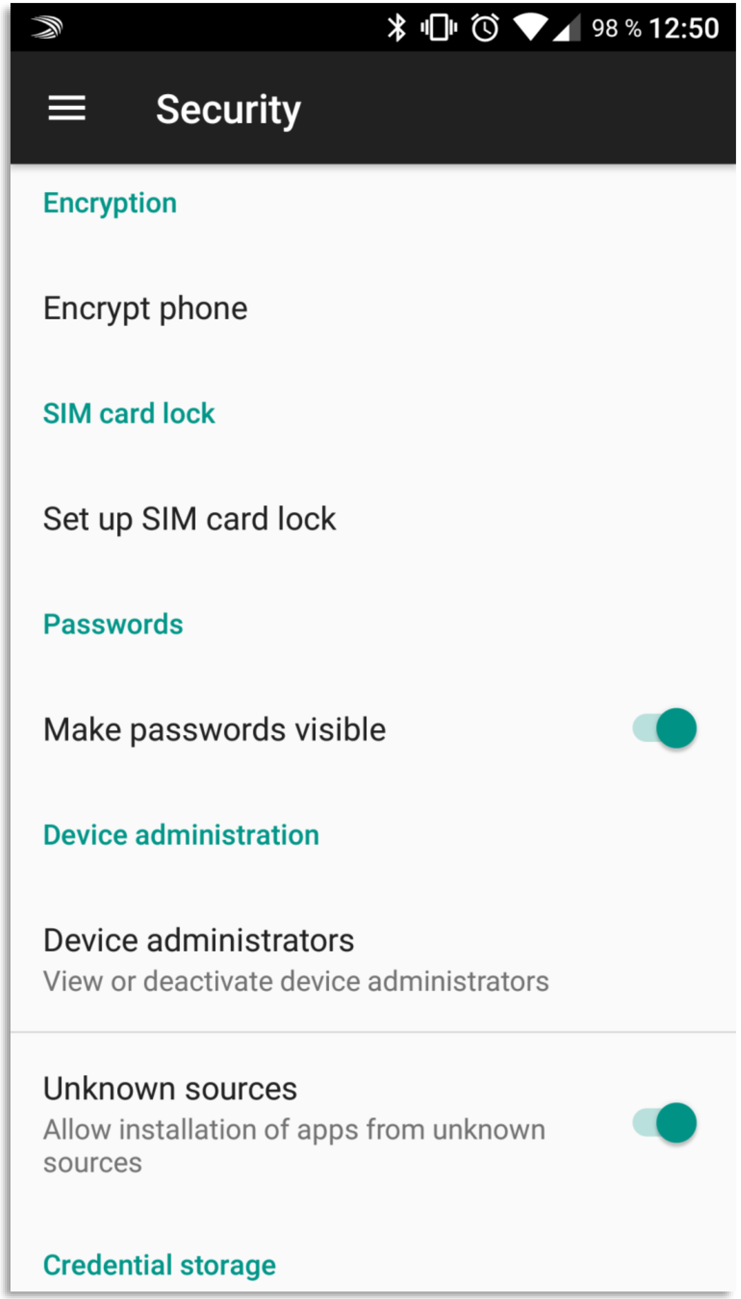
- Apple App Store
  - Cydia 3rd party repositories
- BlackBerry World
  - Also other sources allowed
- Windows Phone Market
  - Exclusive but 3rd party repos

Open approach

- Google Play
  - 3rd party markets

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*

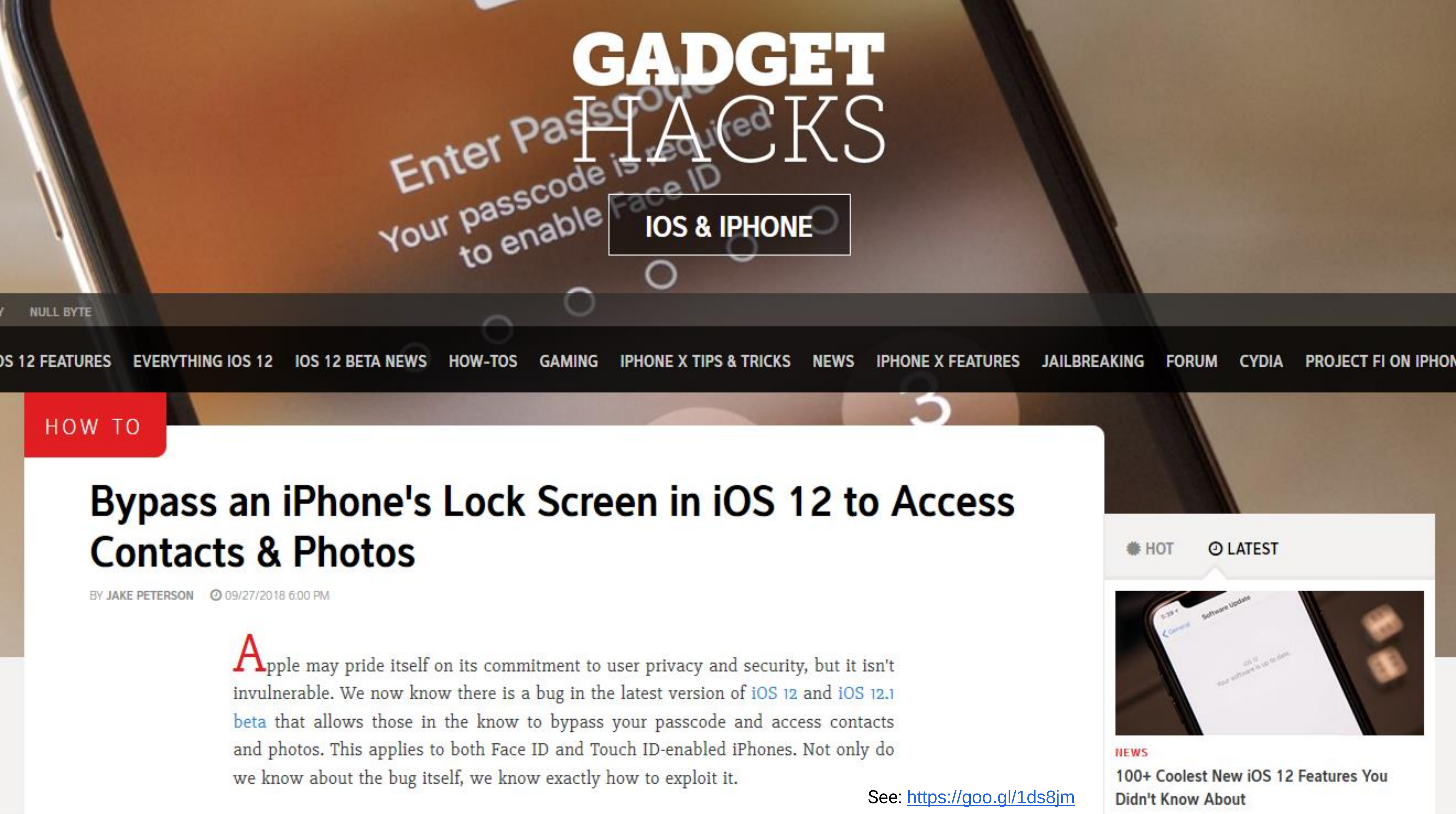
- How is this data protected?

- 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 unlock: Introduced with Android 4.x, revamped with Android 5.x
- Fingerprints: TouchID with iOS 8, Android 6.0



Bypass an iPhone's Lock Screen in iOS 12 to Access Contacts & Photos

BY JAKE PETERSON 09/27/2018 6:00 PM

Apple may pride itself on its commitment to user privacy and security, but it isn't invulnerable. We now know there is a bug in the latest version of iOS 12 and iOS 12.1 beta that allows those in the know to bypass your passcode and access contacts and photos. This applies to both Face ID and Touch ID-enabled iPhones. Not only do we know about the bug itself, we know exactly how to exploit it.

NEWS

100+ Coolest New iOS 12 Features You Didn't Know About

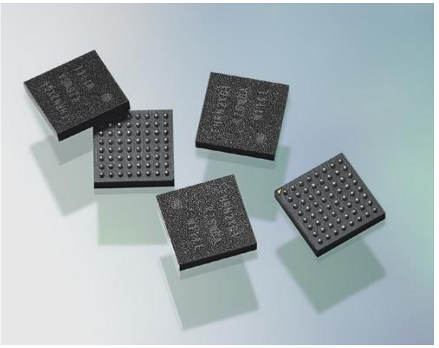
See: https://goo.gl/1ds8jm

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!
  - Device storage or Secure Element



Access Protection – Remote Wipe

- Encryption
  - Huge advantage for remote wipe

- From the iOS Security Guide (Q4 / 2019)

> The metadata of all files in the file system is encrypted with a **random key**, which is created when iOS is first installed or when the device is wiped by a user. The file system key is stored in Effaceable Storage. Since it’s **stored on the device**, this key is not used to maintain the 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) server, Exchange ActiveSync, or iCloud). Erasing the key in this manner renders all files cryptographically inaccessible.

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)

Challenge: Bring-your-own-device!

- **Deploy** security policies that the user cannot change
  - Password, encryption, applications, proxy, VPN, etc.
  - *Settings:* Minimum password requirements, …
  - 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, …

Updates

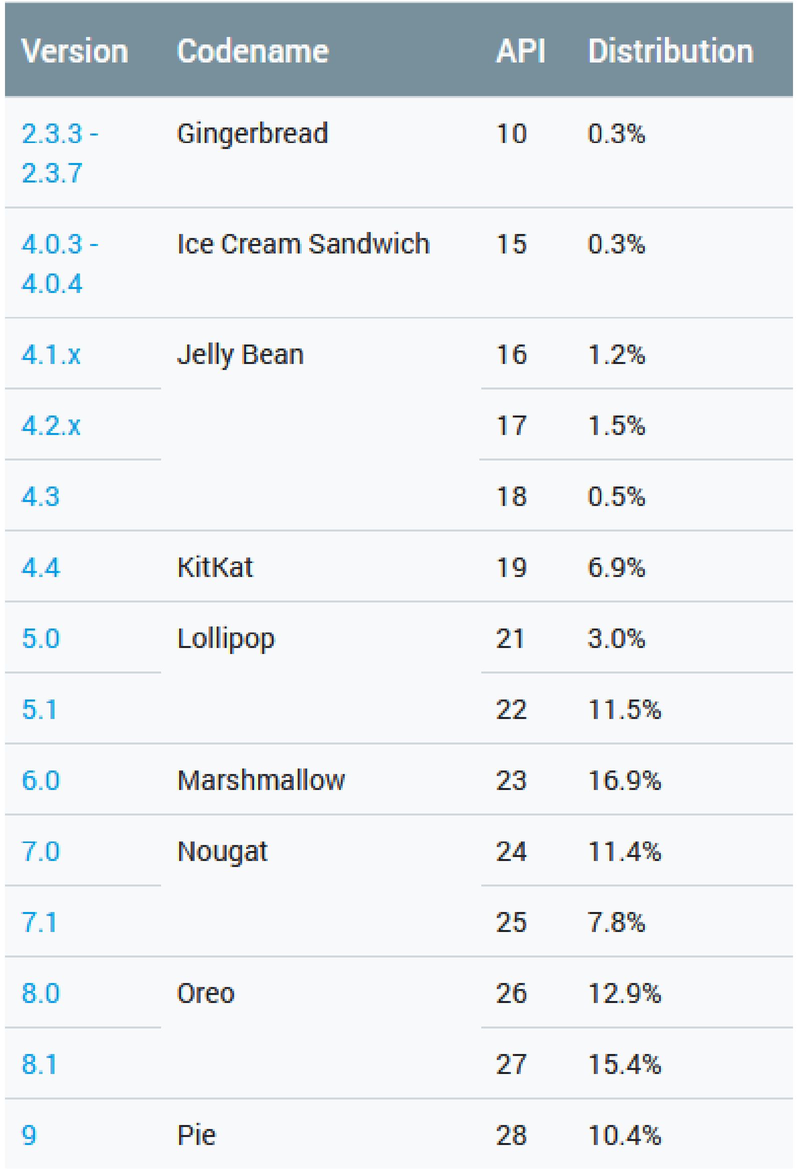
- Security updates are vital, especially in business environments

- Delta updates? Complete firmware?
  - iOS 13 for iPhone 6s+ (1.97 GB)

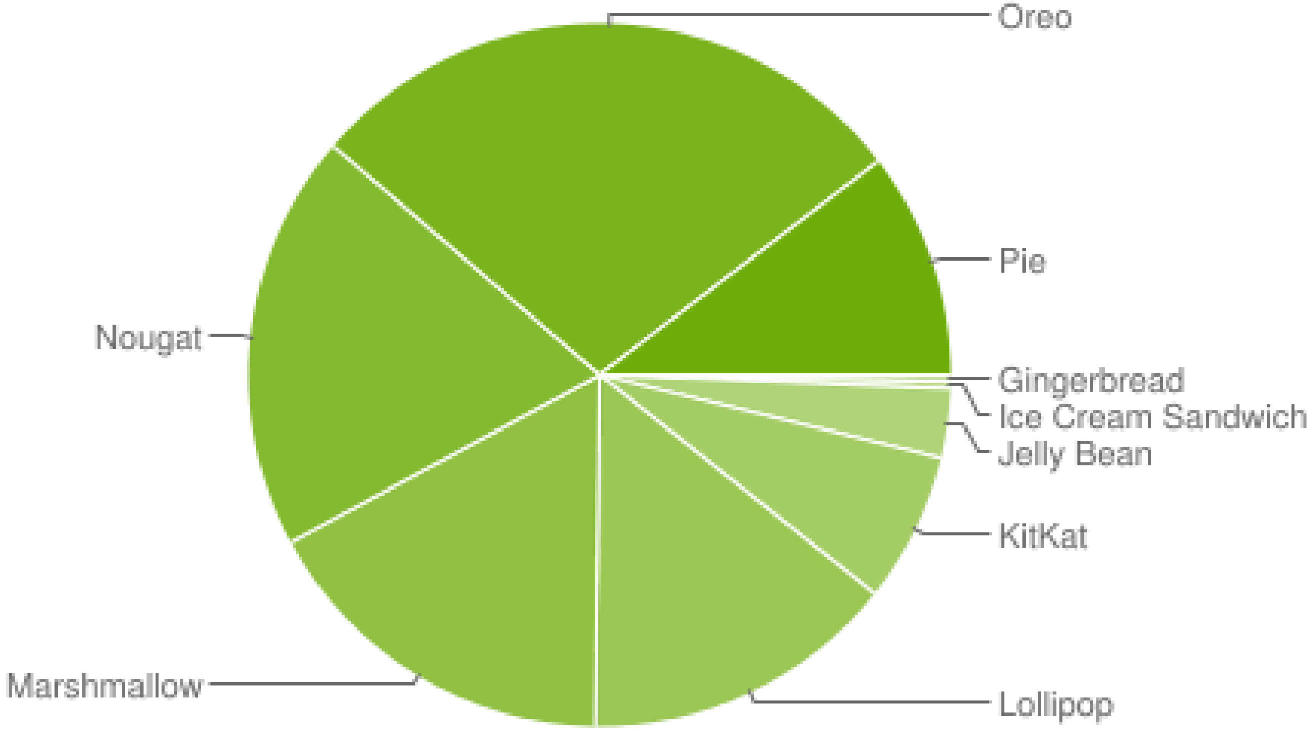
- How long are updates available for a platform?
  - New approach for unification: Android One

- How fast are they distributed?
  - Over MDM?
  - Over The Air (OTA) or wired?
  - Who is involved in this process?

Updates – Android

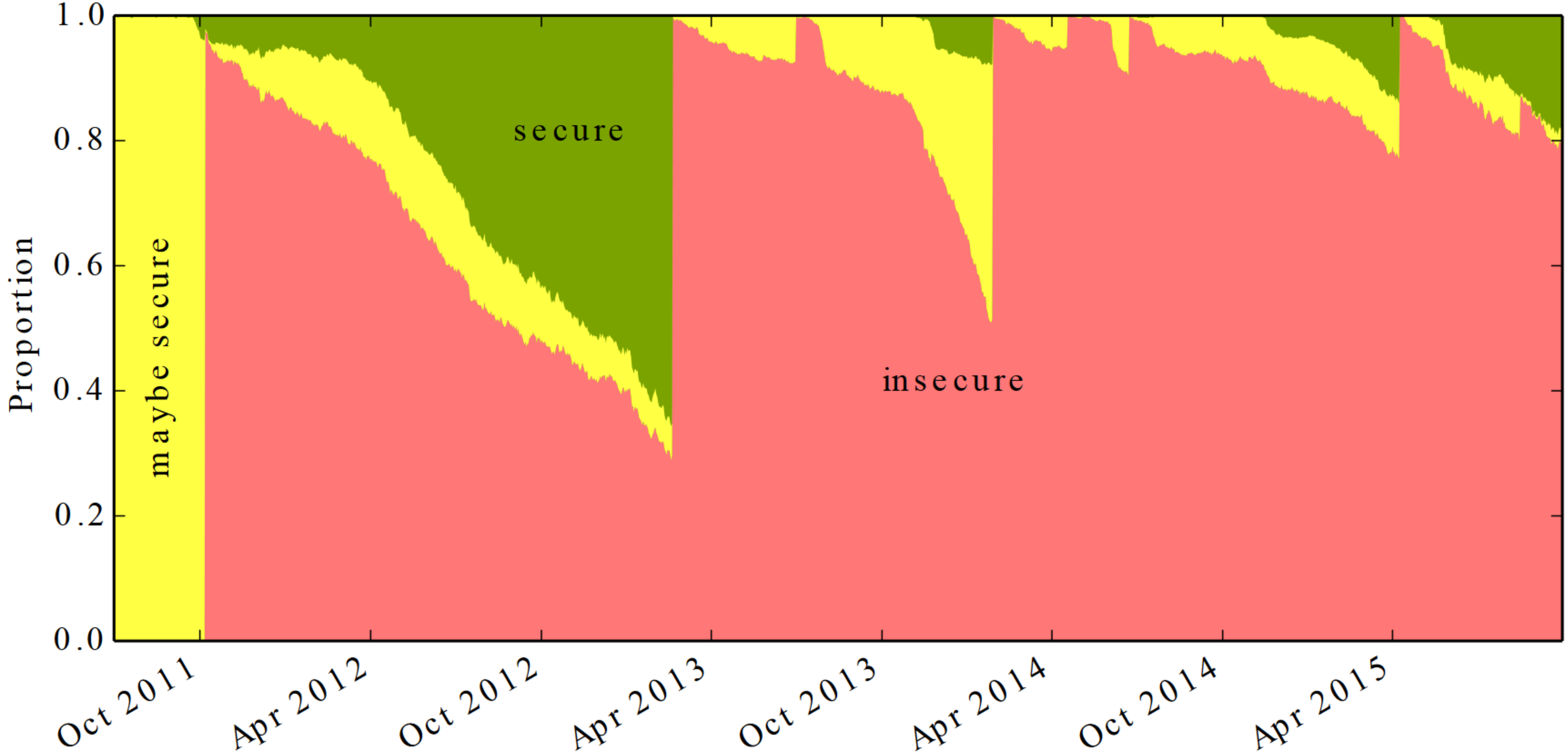


Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

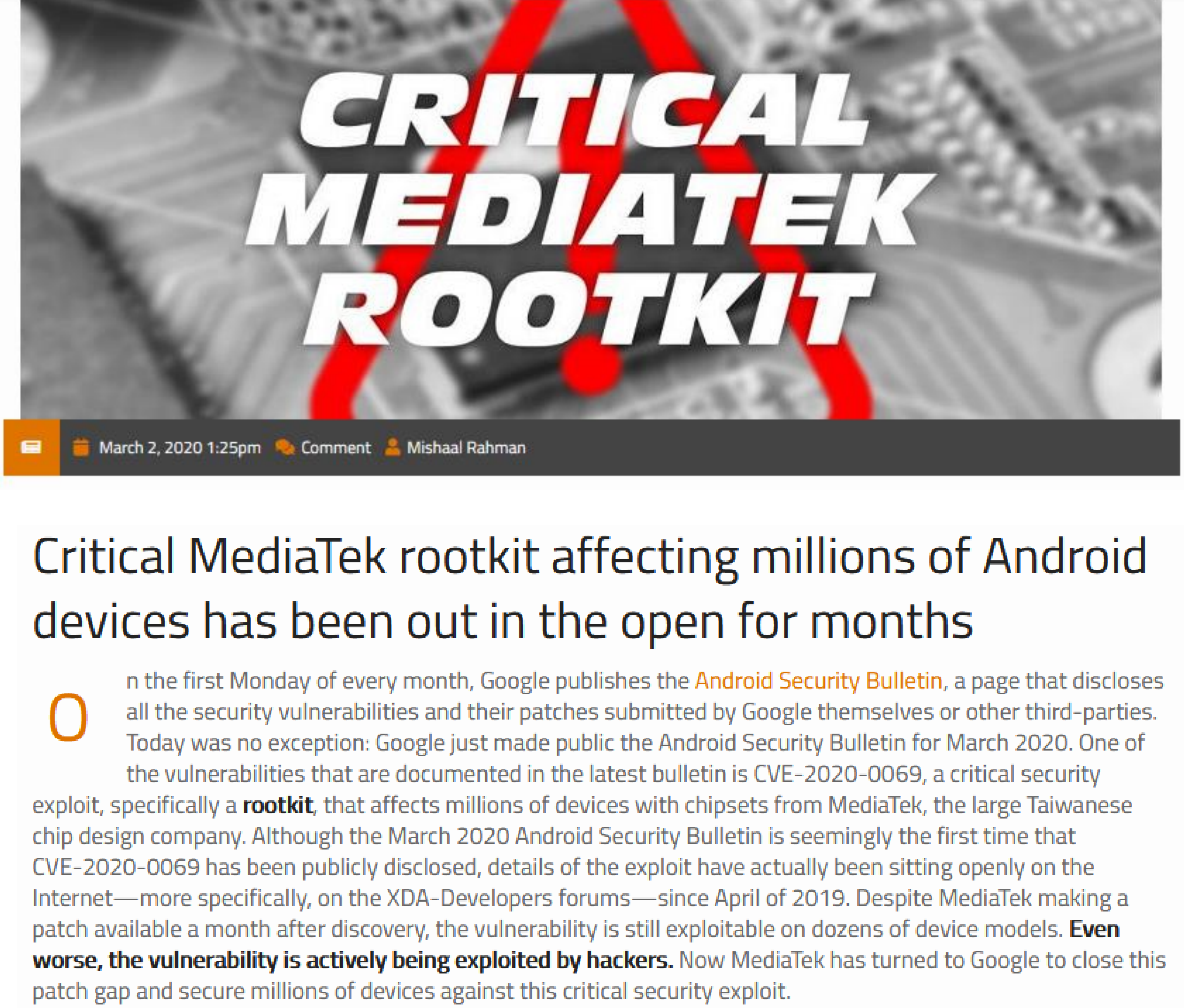


http://developer.android.com/about/dashboards/index.html

Vulnerable Android devices



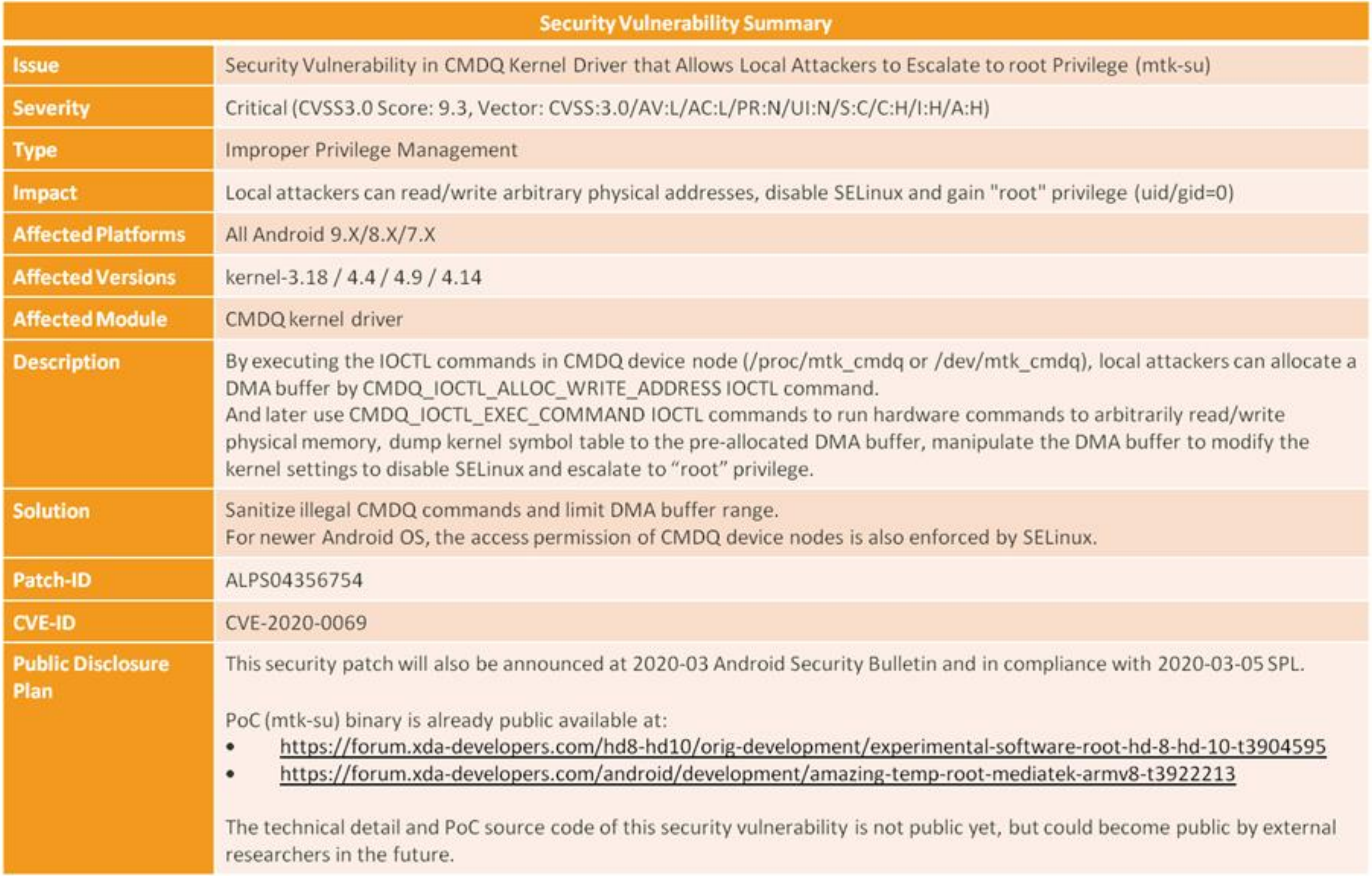
http://androidvulnerabilities.org



March 2, 2020 1:25pm Comment Mishaal Rahman

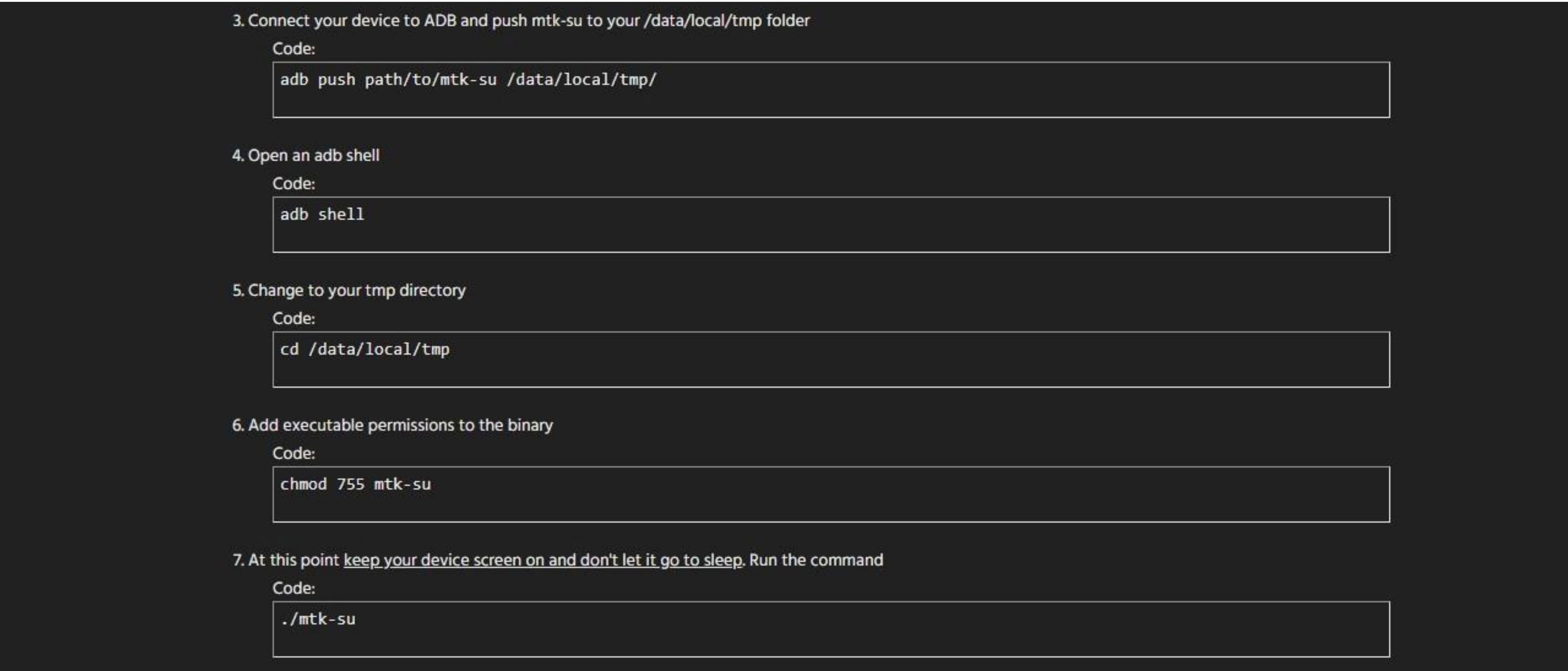
Critical MediaTek rootkit affecting millions of Android devices has been out in the open for months

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. 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. 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. PoC (mtk-su) binary is already public available at: • https://forum.xda-developers.com/hd8-hd10/orig-development/experimental-software-root-hd-8-hd-10-t3904595 • https://forum.xda-developers.com/android/development/amazing-temp-root-mediatek-armv8-t3922213 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-2019-8906	2019-04-16	6.8	Remote	Medium	✓
Do_core_note in readelf.c in libmagic.a in file 5.35 has an out-of-bounds read because memcpy is misused					
CVE-2018-20506	2019-06-19	6.8	Remote	Medium	✓
SQLite 3.25.3 encounters an integer overflow for FTS3 queries in a “merge” operation					
CVE-2018-20505	2019-06-19	5.0	Remote	Low	✓
SQLite 3.25.2 allows remote attackers to cause a denial of service (application crash)					
CVE-2018-4464	2019-04-05	9.3	Remote	Medium	✓
iOS before 12.1.1, macOS before 10.14.2, memory corruption issue					
CVE-2018-4461	2019-04-05	6.8	Remote	Medium	✓
iOS before 12.1.1, Safari < 12.0.2, iTunes 12.9.2 for Windows, multiple memory corruption issues					
CVE-2018-4447	2019-04-05	9.3	Remote	Medium	✓
iOS before 12.1.1, macOS before 10.14.2, memory corruption issue					
CVE-2018-4443	2019-04-05	9.3	Remote	Medium	✓

https://goo.gl/URpyX7

Communication

Key aspect of smartphone: broadband always-on Internet connection

- Mobile network: GRPS, EDGE, UMTS, HSPA+, LTE, (5G)
- WiFi
- Bluetooth
- NFC
- Cloud!

Communication – Mobile Networks

- Many standards: 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 (current standard)

http://gsmmap.org

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

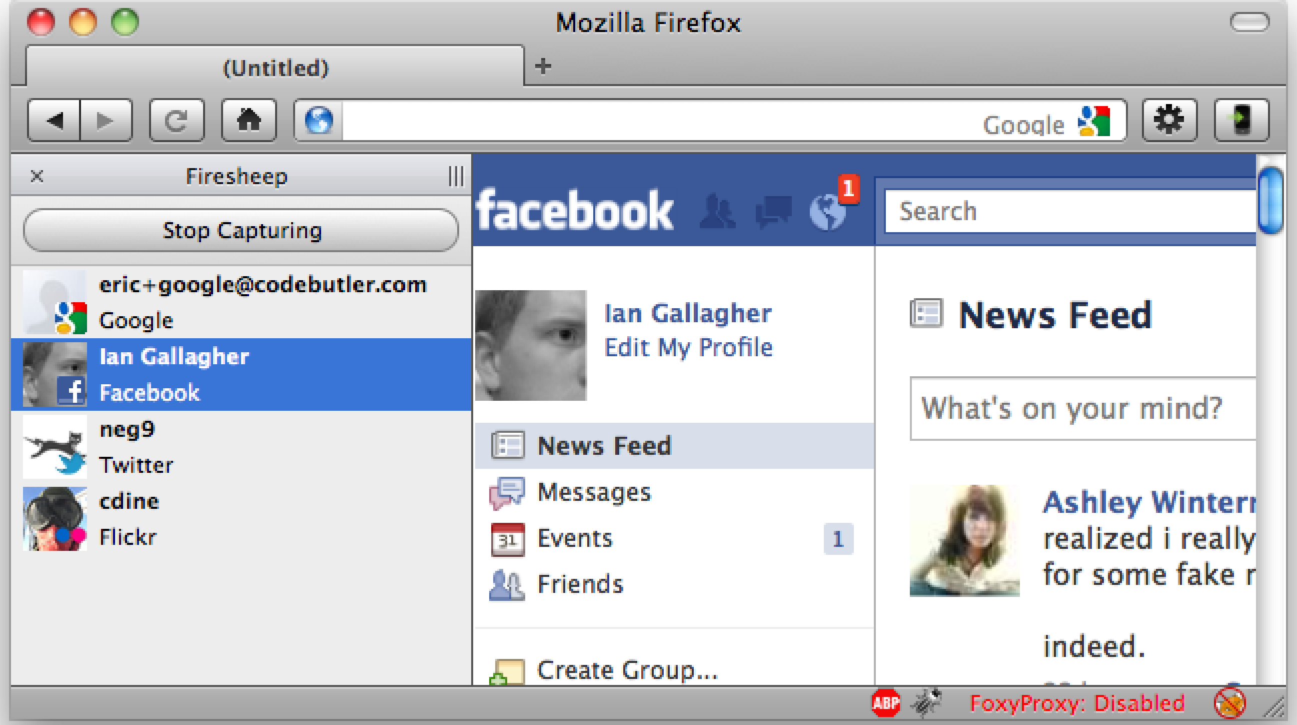
- However:
  - Telephone, SMS, MMS services integrated as apps into phone
  - Attack over SMS, e.g. Denial of Service
  - MMS with Malware, e.g. „Stagefright“ on Android

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
- Droidsheep
- Firesheep
- Dsploit
- …



eric+google@codebutler.com
Google
Ian Gallagher
Facebook
neg9
Twitter
cdine
Flickr

Ian Gallagher
Edit My Profile

Ashley Winterr
realized i really
for some fake r
indeed.

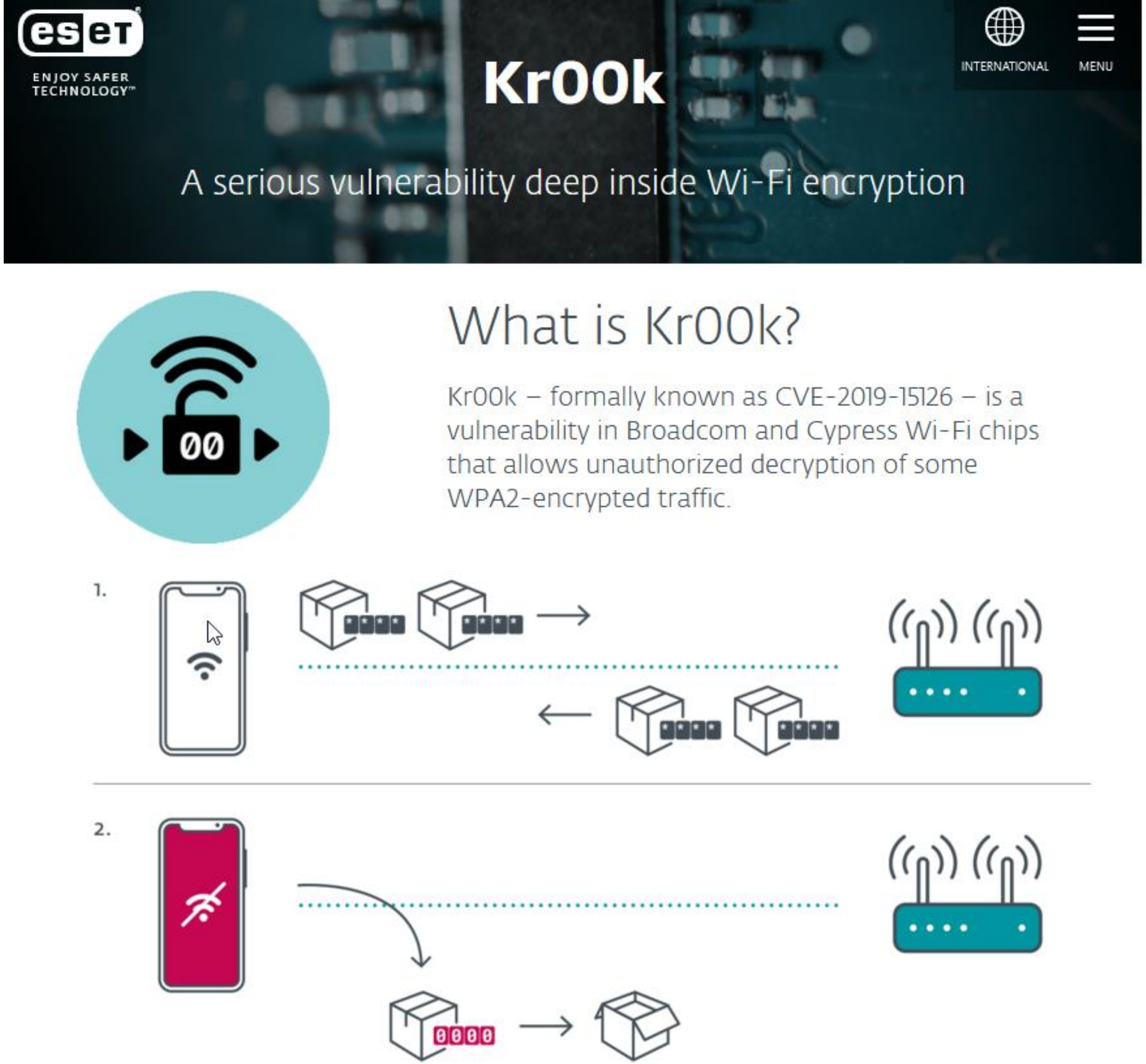
Communication – WiFi

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)?
  - Send out your location, unique IDs for advertisements

- WiFi location (& user) tracking
  - Android: „Location service may scan for WiFis although WiFi disabled“
  - Apple: MAC address randomization since iOS 8



eset ENJOY SAFER TECHNOLOGY™

INTERNATIONAL MENU

Kr00k

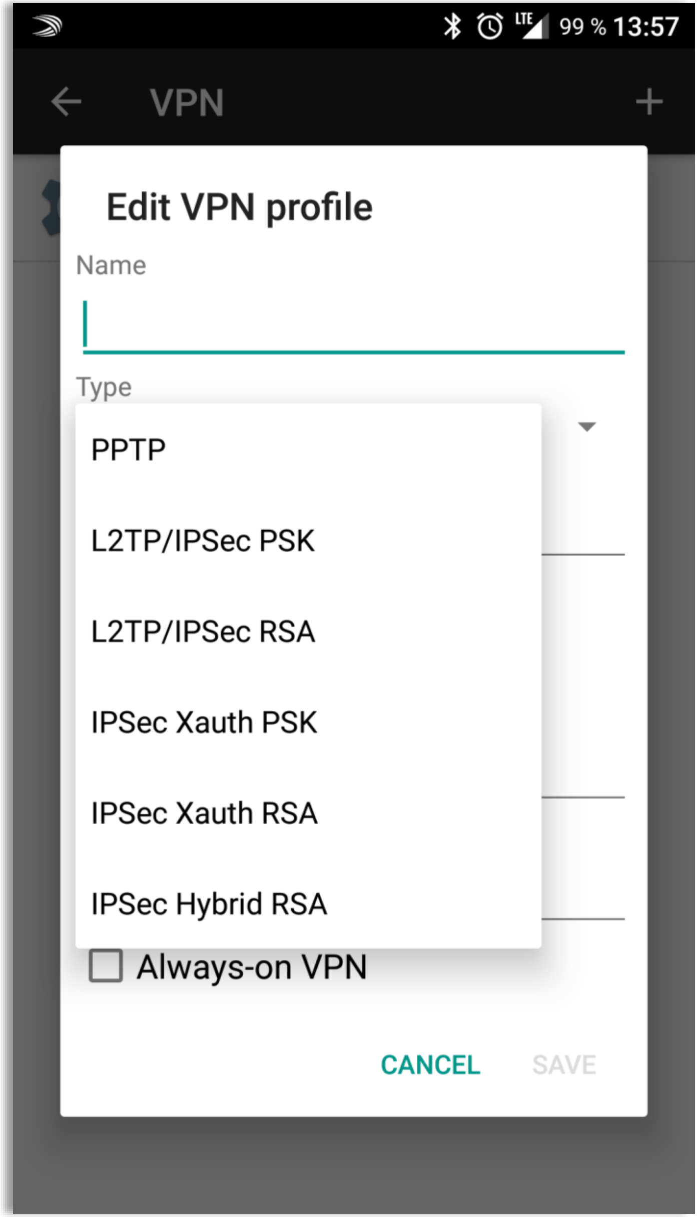
A serious vulnerability deep inside Wi-Fi encryption

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

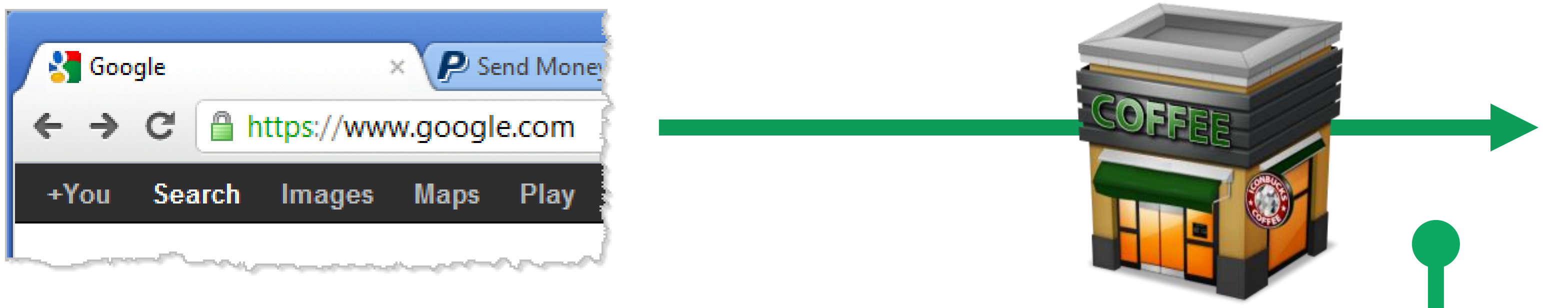
Force all traffic over VPN...

→ Avoid problems with open WiFis

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

Attackers cannot...

1. *... read the transfer*
2. *... tamper the data transferred*
3. *... impersonate the destination*



https://www.google.com

End-to-end encrypted communication channel

Communication - Bluetooth

Problems by design

- Visibility
- Secure data transfer?
- Tethering?

Problems by platform

- *„Heap overflow allows Remote Code Execution“* (CVE-2017-0781, CVE-2017-0782)
  - Same implementation flaw in Android and iOS! - „BlueBorne“

See: https://goo.gl/Gti1Tj

- Critical vulnerability in iOS AirDrop, fixed with iOS 9
  - Be within Bluetooth range
  - Install any app remotely on vulnerable iPhone
    → implicitly trusted, no warning dialog!

See: http://goo.gl/vsK7yC

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(?)
...
```

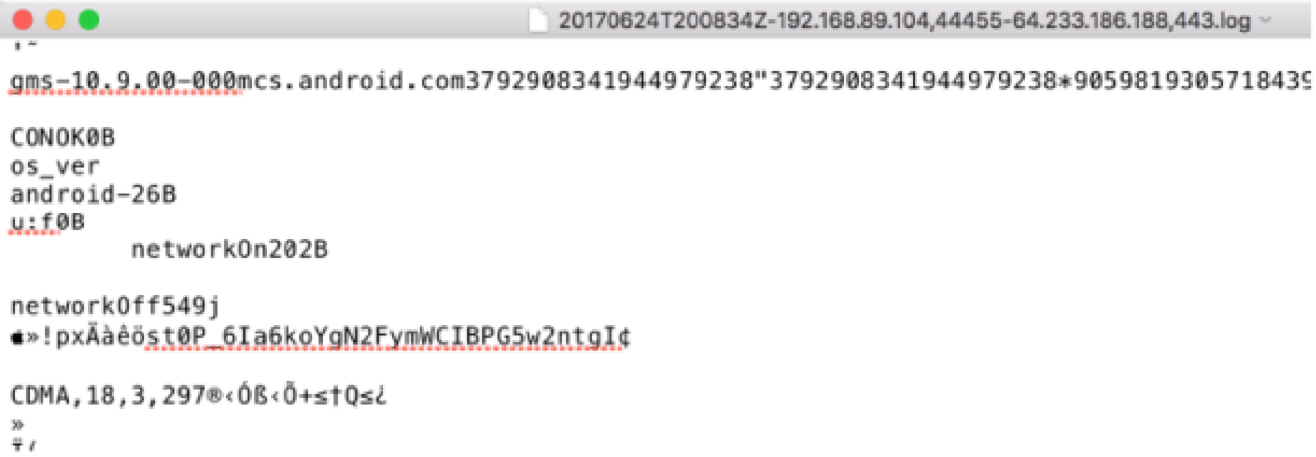
Good: TLS is used to protect transfer

Bad: The certificate‘s validity is not checked!

See: https://goo.gl/LE7uBf

Communication - Location

Cell tower location sent to Google (sniffed):



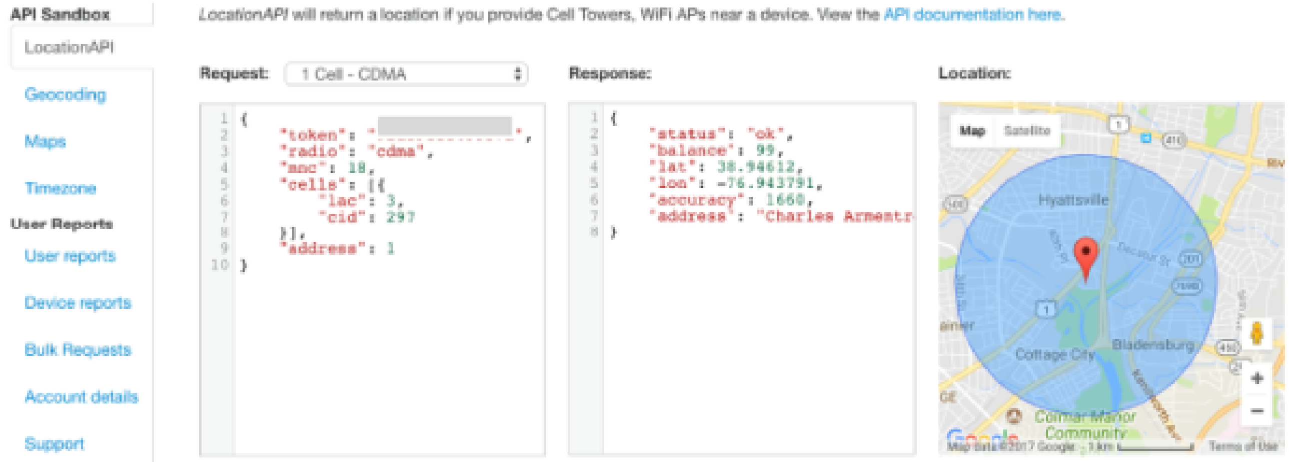
20170624T200834Z-192.168.89.104,44455-64.233.186.188,443.log

```
gms-10.9.00-000mcs.android.com3792908341944979238"3792908341944979238*905981930571843

CONOK0B
os_ver
android-26B
u:f0B
        networkOn202B

networkOff549j
»!pxÅàêöst0P_6Ia6koYgN2FymWCIBPG5w2ntgI¢

CDMA,18,3,297®‹ÓB‹Õ+≤†Q≤¿
»
```



API Sandbox

LocationAPI

Geocoding

Maps

Timezone

User Reports

User reports

Device reports

Bulk Requests

Account details

Support

LocationAPI will return a location if you provide Cell Towers, WiFi APs near a device. View the API documentation here.

Request: 1 Cell - CDMA

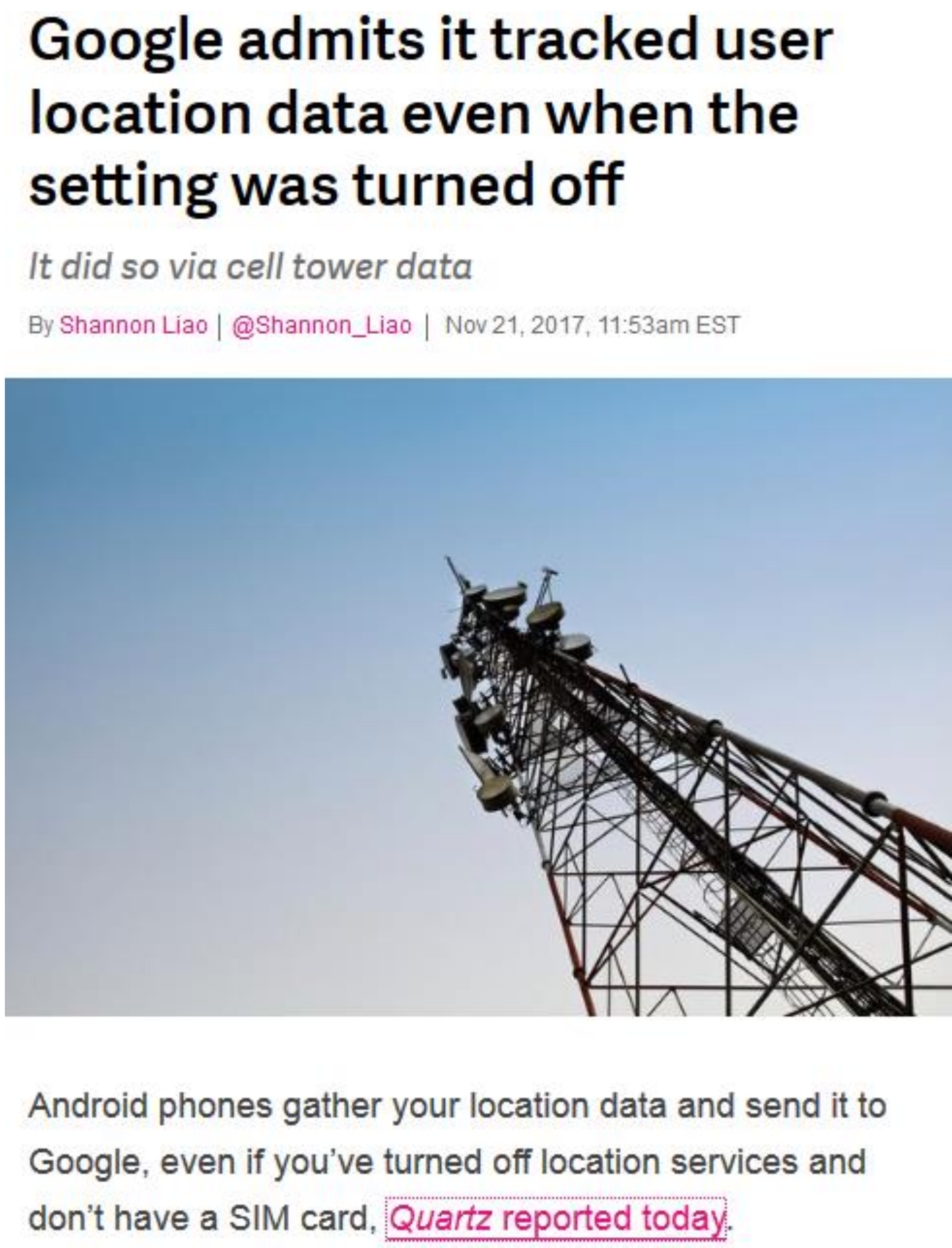
```
{
    "token": "[redacted]",
    "radio": "cdma",
    "mnc": 18,
    "cells": [{
        "lac": 3,
        "cid": 297
    }],
    "address": 1
}
```

Response:

```
{
    "status": "ok",
    "balance": 99,
    "lat": 38.94612,
    "lon": -76.943791,
    "accuracy": 1660,
    "address": "Charles Armentr
}
```

Location:

See: https://goo.gl/LAHPah



Google admits it tracked user location data even when the setting was turned off

It did so via cell tower data

By Shannon Liao | @Shannon_Liao | Nov 21, 2017, 11:53am EST

Android phones gather your location data and send it to Google, even if you've turned off location services and don't have a SIM card, *Quartz reported today*.

See: https://goo.gl/wD7pcX

Communication – NFC

- Near Field Communication (NFC)
  - Short range (freq. 13.56 MHz) → some kind of security
  - Commerce, Social Networking, Access tokens, …

- Devices can act as both reader and tag

- Rising popularity with payment: Android Pay & Apple Pay
  - Token instead of card → generated and stored on phone's Secure Element

Communication – Cloud

- More and more cloud services
  - **Apps:** Dropbox, Google Drive, OneDrive, …
  - **Backup:** iCloud, Google Backup, Google Photos
  - **Sync:** iCloud, Google frameworks
  - **Messaging:** Apple Push Notifications, Google Cloud Messaging

→ *Security? Company vs. private data? MDM controls? Business model?*

- Legal aspects
  - Safe Harbour pact declared invalid on Oct. 6, 2015
  - Successor: EU-US Privacy Shield in effect since July 12, 2016

Do you know what data of you is in the cloud and where it is?!

Outlook

- 12.03.2020
  - Key & Data Storage on Mobile Devices
  - (Off-topic: Licenses)

- 02.04.2020
  - iOS Platform Security