Mobile Apps Security

ITU PITA Workshop on
Mobile network planning and security

Sami TABBANE

21-22 October 2019

Objectives

Provide an overview of Apps security issues, classification, importance and targeted points of attacks

Agenda

I. Introduction

II. Mobile applications threats

III. Mobile Architecture

IV. Attacks Points

Agenda

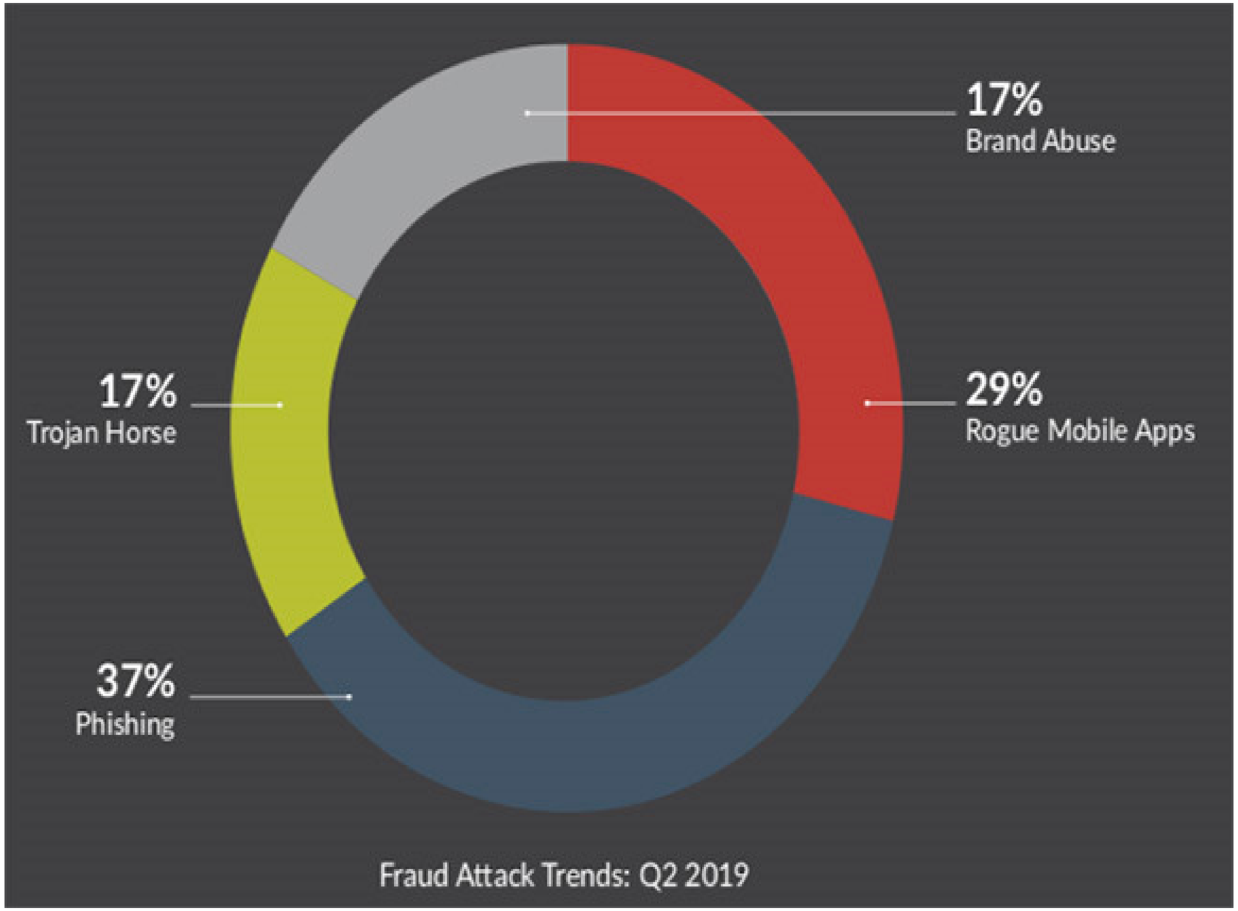
I. Introduction

Mobile Management and Security Challenges

- **1 in 20** Mobile devices stolen in 2010
- **194 billion** apps downloaded in 2018: Google Play downloads of **76 billion, 30 billion** downloads on iOS App Store
- **70%** of Mobile device spam is fraudulent financial services
- **40%** of mobile malware rises in 2018

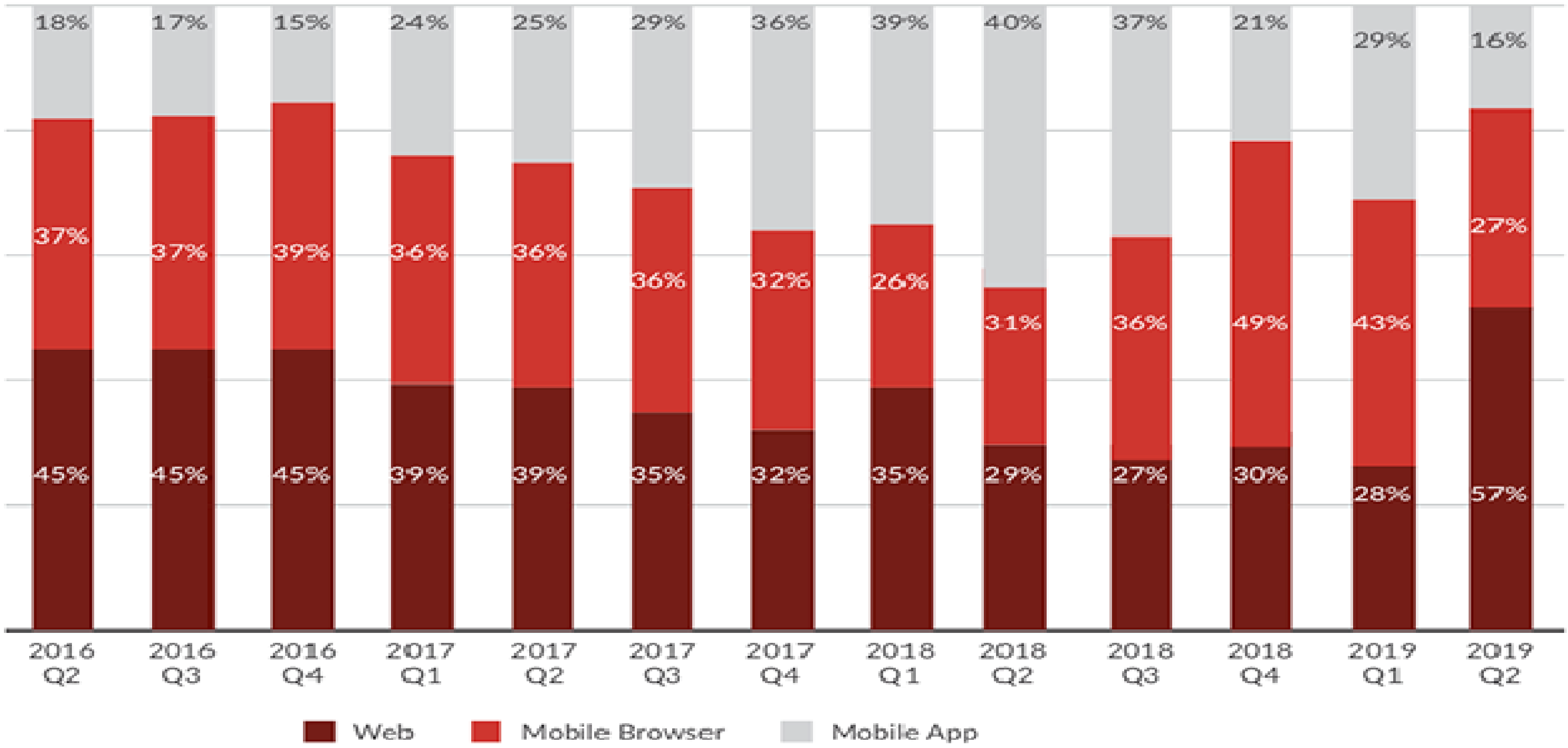
Sources:
Evans Data Mobile Developer Survey Mobile Development Report 2012
Business Insider (September 2012)
G DATA Analysts 2018
BusinessOfApps

Fake mobile app fraud tripled in first half of 2019



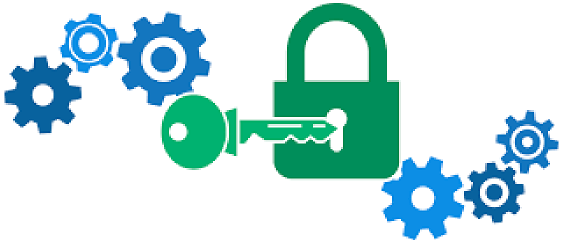
Attacks via financial malware and rogue mobile apps have increased significantly (80 and 191 percent, respectively).

Fraud transaction distribution by channel



What do attackers want?

- **Credentials**
  - To your device
  - To external services (email, banking, etc.)

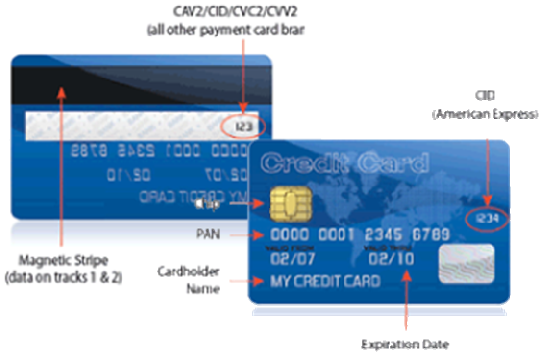


- **Personal Data**
  - Full Name,
  - ID card number, Social Security Number
  - Address book data
  - Location data

Mobile applications threats

What do attackers want?

- ➢ **Cardholder Data**
  - Card Numbers
  - Expiration
  - CVV, etc.

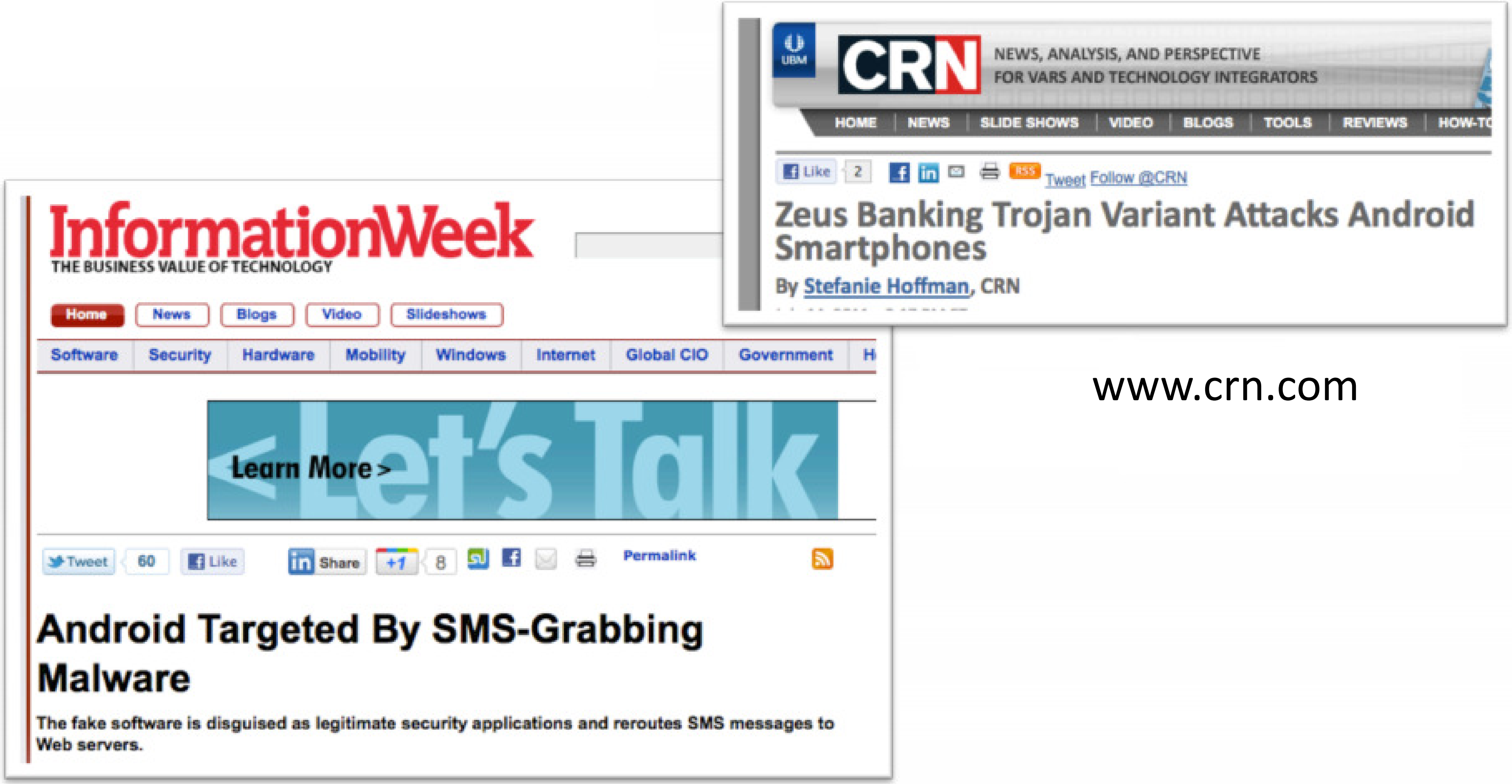


- ➢ **Access to your device**
  - Sniff your connections
  - Use your device (botnets, spamming)
  - Steal trade secrets or other sensitive data

What are the hackers objectives?

- **Stealing Money** and **information**
- **Embarrassing** people
- **Getting famous**
- **Breaking out of restrictive application licensing** and functionality
- **Breaking out of restrictive platforms**
- **Hacking** for the Anonynous …

Apps In the Press



Zeus Banking Trojan Variant Attacks Android Smartphones

By Stefanie Hoffman, CRN

Android Targeted By SMS-Grabbing Malware

The fake software is disguised as legitimate security applications and reroutes SMS messages to Web servers.

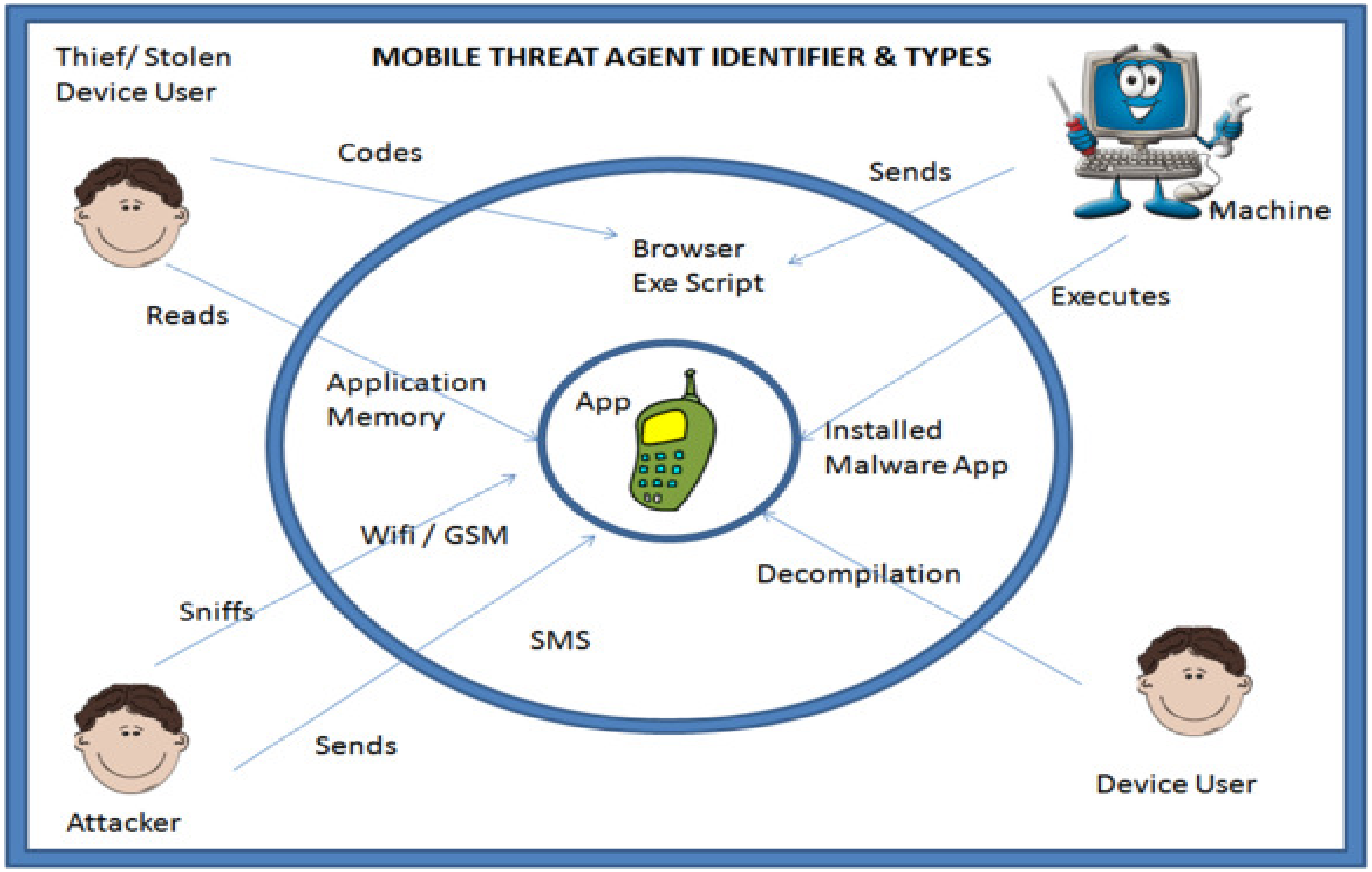
www.crn.com

www.informationweek.com

Agenda

II. Mobile applications threats

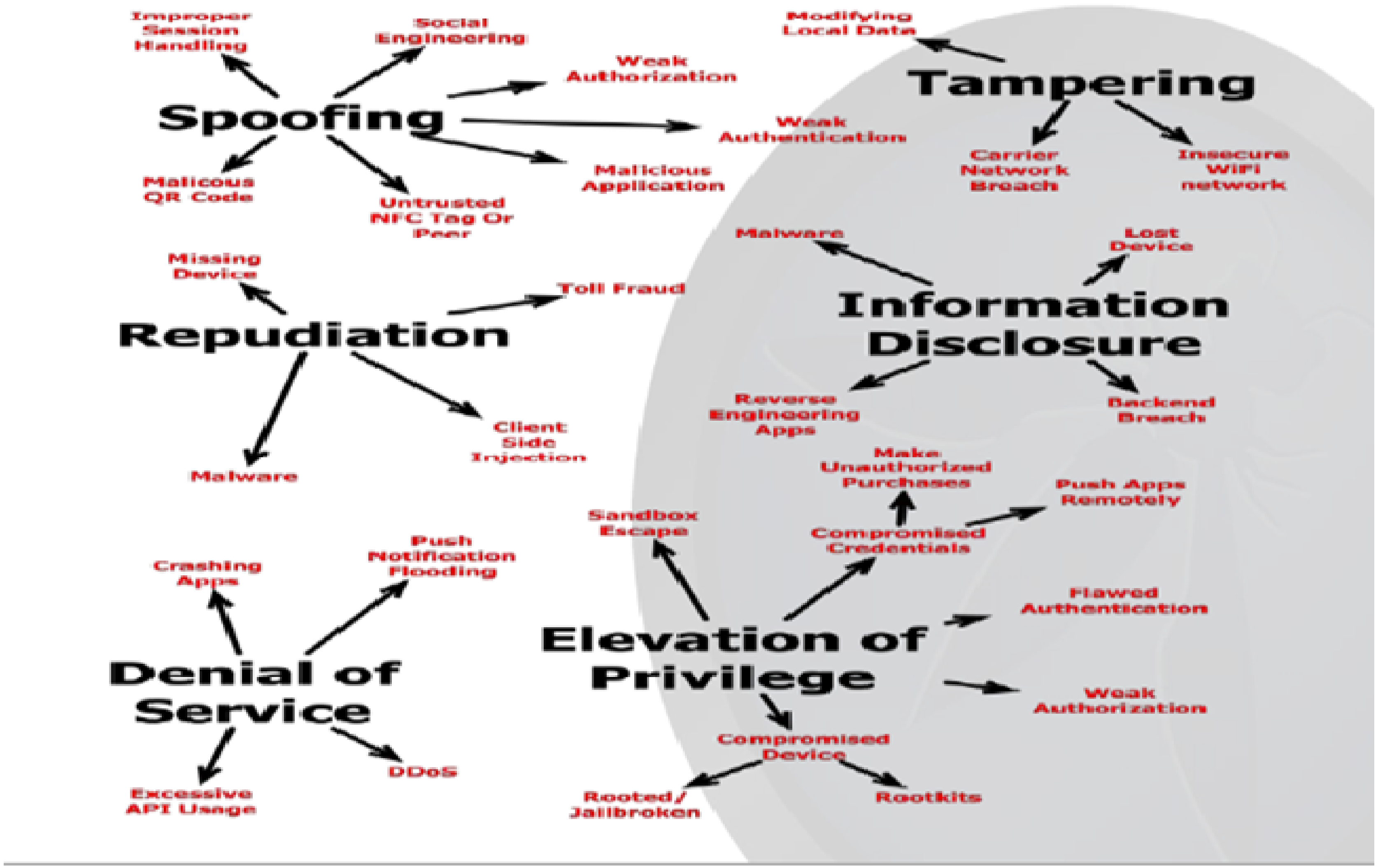
Mobile applications threats



Source: OWASP

The **Open Web Application Security Project** (**OWASP**) is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security

Mobile applications threats



Spoofing
- Improper Session Handling
- Social Engineering
- Weak Authorization
- Weak Authentication
- Malicious Application
- Untrusted NFC Tag Or Peer
- Malicous QR Code

Tampering
- Modifying Local Data
- Carrier Network Breach
- Insecure WiFi network

Repudiation
- Missing Device
- Toll Fraud
- Client Side Injection
- Malware

Information Disclosure
- Malware
- Lost Device
- Reverse Engineering Apps
- Backend Breach

Denial of Service
- Crashing Apps
- Push Notification Flooding
- Excessive API Usage
- DDoS

Elevation of Privilege
- Sandbox Escape
- Compromised Credentials
  - Make Unauthorized Purchases
  - Push Apps Remotely
- Flawed Authentication
- Weak Authorization
- Compromised Device
  - Rooted/ Jailbroken
  - Rootkits

Trends in mobile attacks

Data leakage

One of the most worrisome threats to enterprise security in 2019

Solution: Scan apps for "leaky behavior," and automate the blocking of problematic processes. Mobile threat defense (MTD) solutions: Symantec's Endpoint Protection Mobile, CheckPoint's SandBlast Mobile, and Zimperium's zIPS Protection.

Social engineering

91% of cybercrime starts with email by phishing (users are three times more likely to respond to a phishing attack on a mobile device than a desktop, *IBM*)

83% of phishing attacks over the past year took place outside the inbox — in text messages or in apps like FB Messenger and WhatsApp as well as games and social media services.

Wi-Fi interference

A quarter of devices open potentially insecure Wi-Fi networks. 4% of devices have encountered a MITM attack. Network spoofing (*disguising a communication from an unknown source as being from a known, trusted source*) has increased, and yet less than half of people bother to secure their connection while traveling and relying on public networks.

Out-of-date devices

Smartphones, tablets and small connected devices generally don't come with guarantees of timely and ongoing SW updates

Cryptojacking attacks

Someone uses a device to mine for cryptocurrency without the owner's knowledge. A third of all attacks in the first half of 2018

Some security issues sources

- *35% of communications* sent by mobile devices are unencrypted
- The average device connects to over *160 unique IP addresses* daily.
- *43% of mobile device users* do not use a passcode, PIN or pattern lock on their devices
- *25% mobile apps* include at least one high-risk security flaw

OWASP Top 10 Mobile Risks

M1: Improper Platform Usage

M2: Insecure Data Storage

M3: Insecure Communication

M4: Insecure Authentication

M5: Insufficient Cryptography

M6: Insecure Authorization

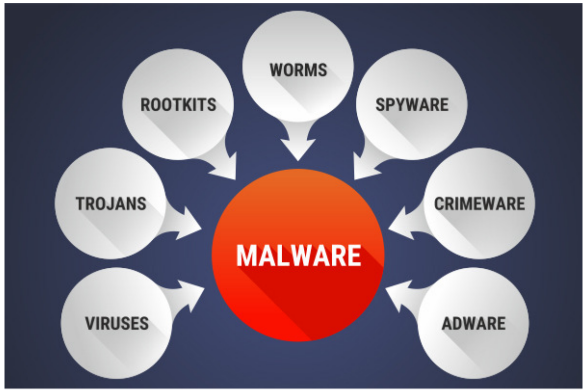
M7: Client Code Quality

M8: Code Tampering

M9: Reverse Engineering

M10: Extraneous Functionality

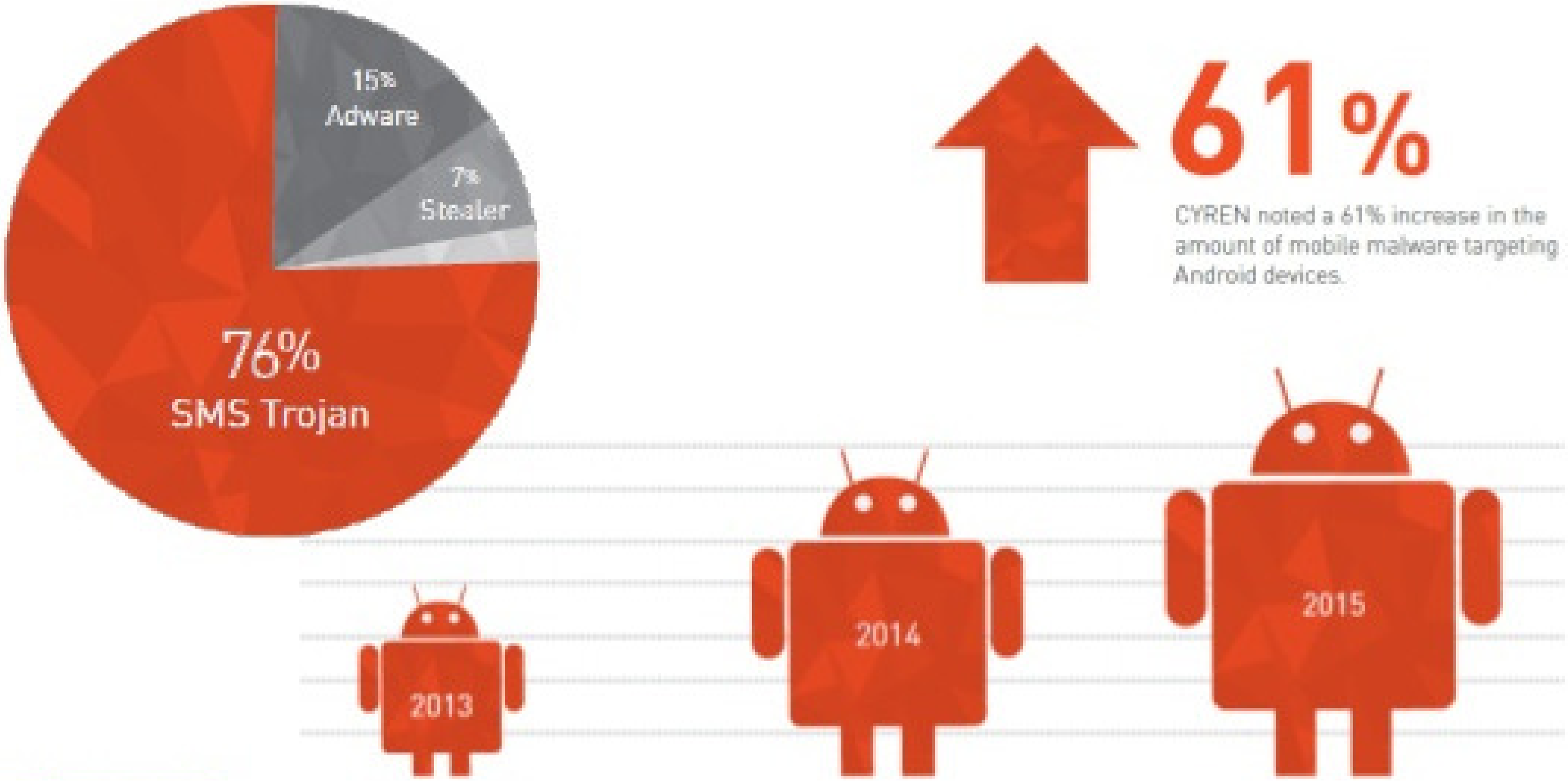
Malware attacks



Definition: **Malicious** software, commonly known as **malware**, is any software that brings harm to a computer system. **Malware** can be in the form of worms, viruses, Trojans, spyware, adware and rootkits, etc., which steal protected data, delete documents or add software not approved by a user.

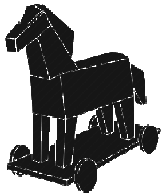
- **Explosion of unknown malwares**
- Known malwares are maintaining the same ratio outlining the **mismatch of antivirus solutions** to protect against such threats.
- For both Android and iOS applications, **data leakage** and **corruption** represent the **main source of threat** despite store policies and permissions.
- 1 out of 4 Applications embeds vulnerabilities for which 75% are ranked in the OWASP TOP 10.
- The **most popular applications are not the safest ones**.
- **Games**, **Entertainment** and **Tools** have **higher threat ratios**.

Android malware 2015



CYREN noted a 61% increase in the amount of mobile malware targeting Android devices.

Some current Android malwares

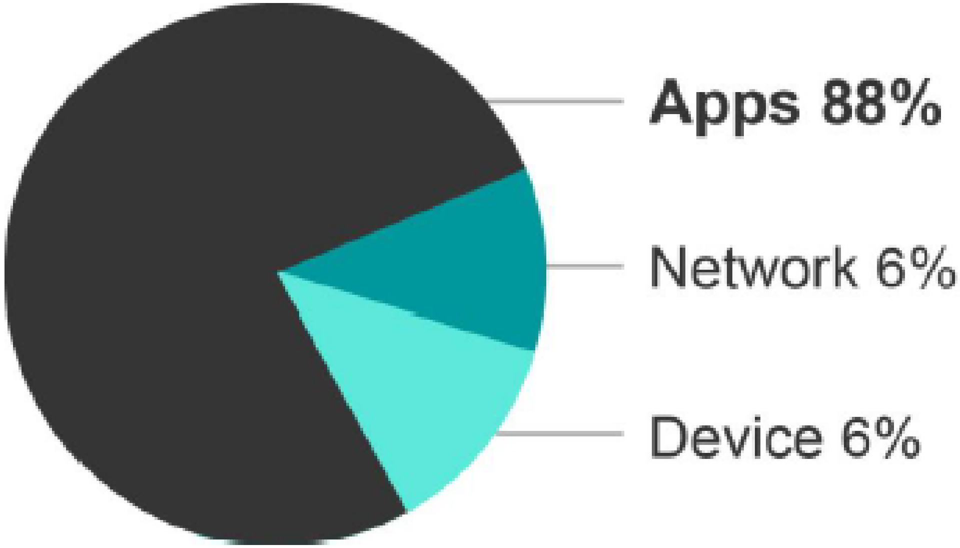


Description

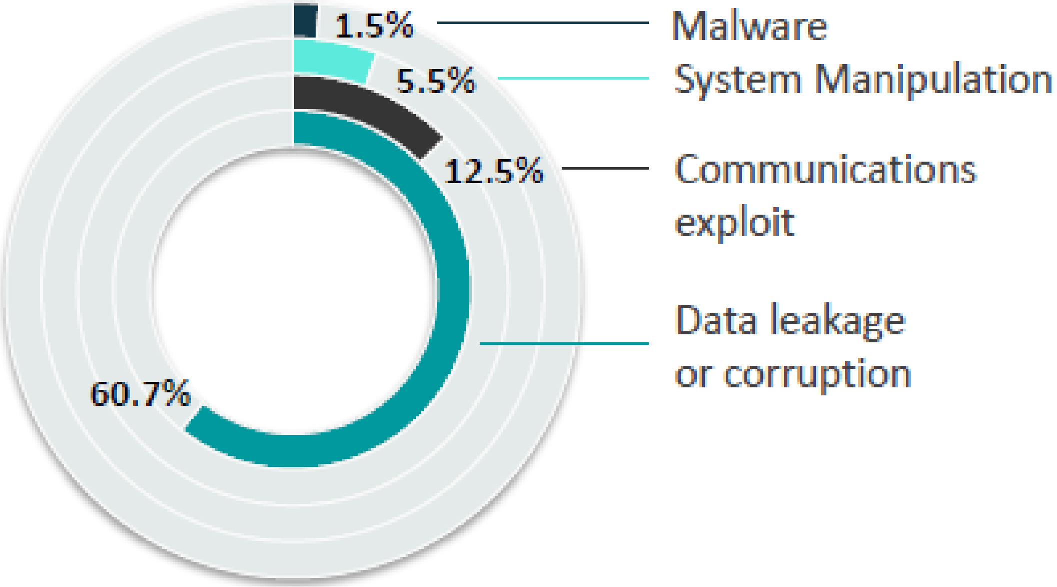
- **AccuTrack**
  This application turns an Android smartphone into a **GPS tracker**.
- **Ackposts**
  This **Trojan steals contact information** from the compromised device and uploads them to a remote server.
- **Acnetdoor**
  This **Trojan** opens a **backdoor on the infected device and sends the IP address** to a remote server.
- **Adsms**
  This is a **Trojan** which is allowed to **send SMS messages**. The distribution channel is through a SMS message containing the download link.
- **BankBot**
  This malware tries to **steal users' confidential information** and money from bank and mobile accounts associated with infected devices.

http://forensics.spreitzenbarth.de/android-malware/

Repartition of mobile threats



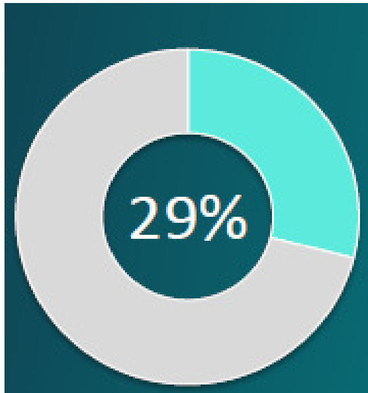
The combined set of Applications across devices highlights more than 60% of applications to feature **data leakage or corruption.**



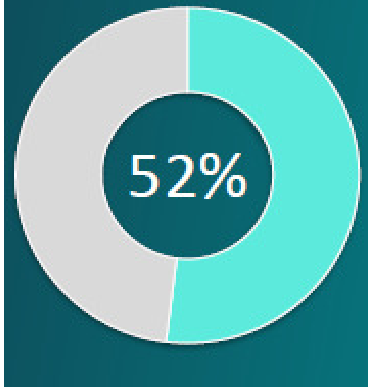
On average, a smartphone has **51 applications** with:

- 1 out of 2 **being intrusive**.
- **25 Safe** applications
- **25.4 Intrusive** applications
- **0.6 Malicious** application

Repartition of mobile threats



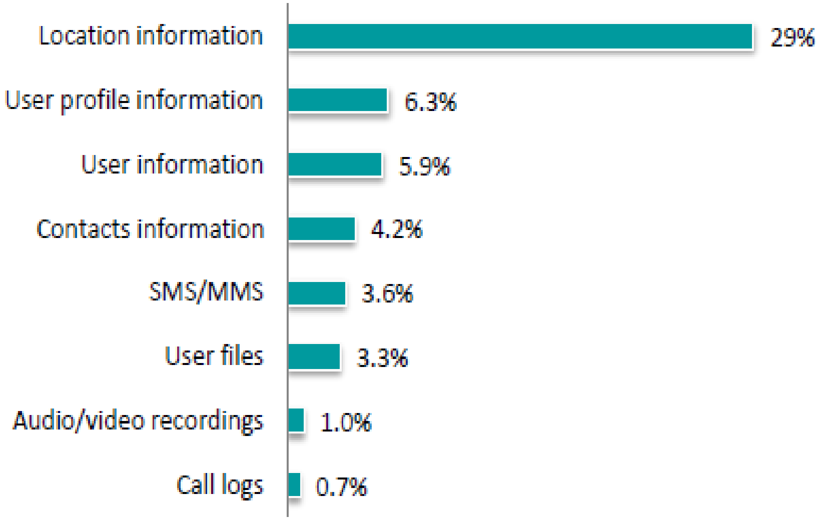
of Applications exploit user's location



of Applications manipulate hardware information

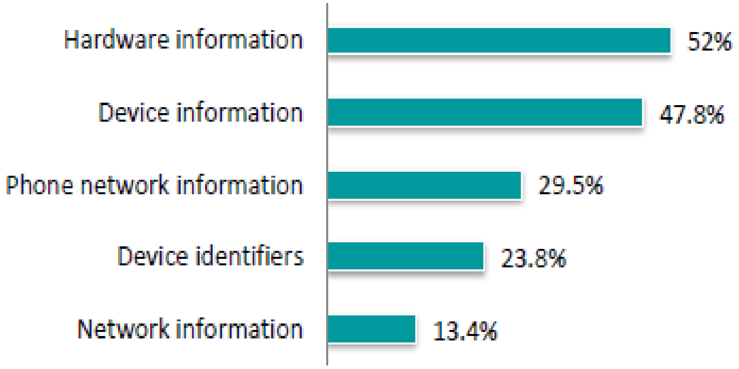
User's data

% of Applications manipulating user's data



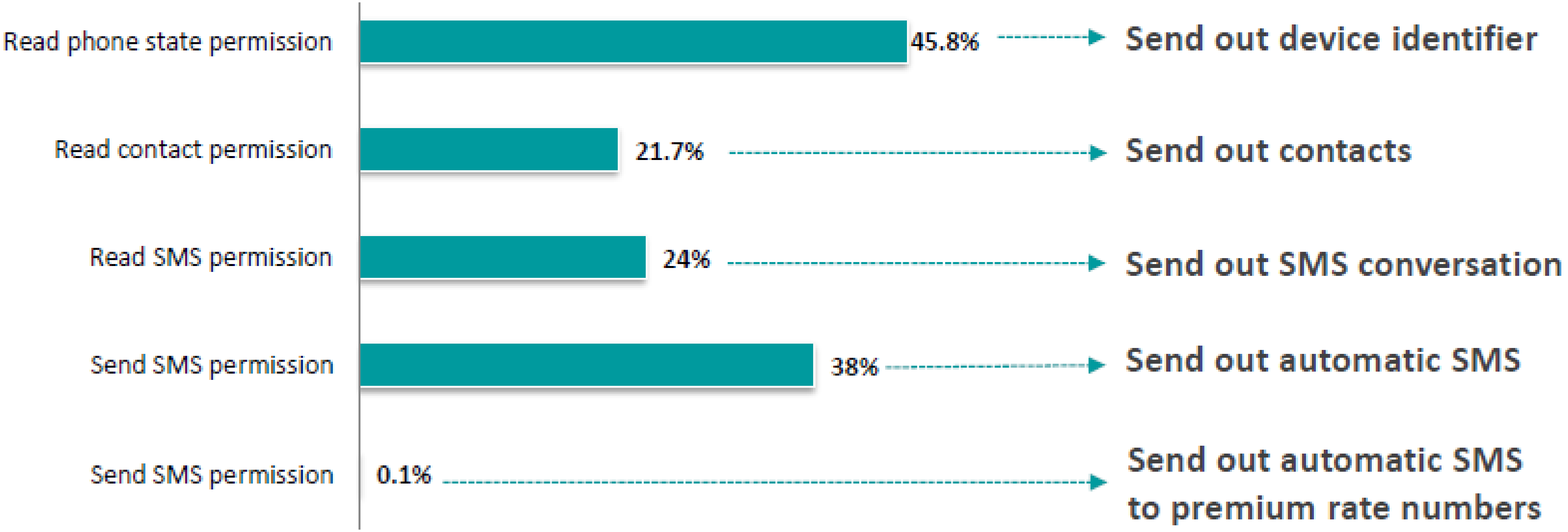
Device's data

% of Applications manipulating device's data

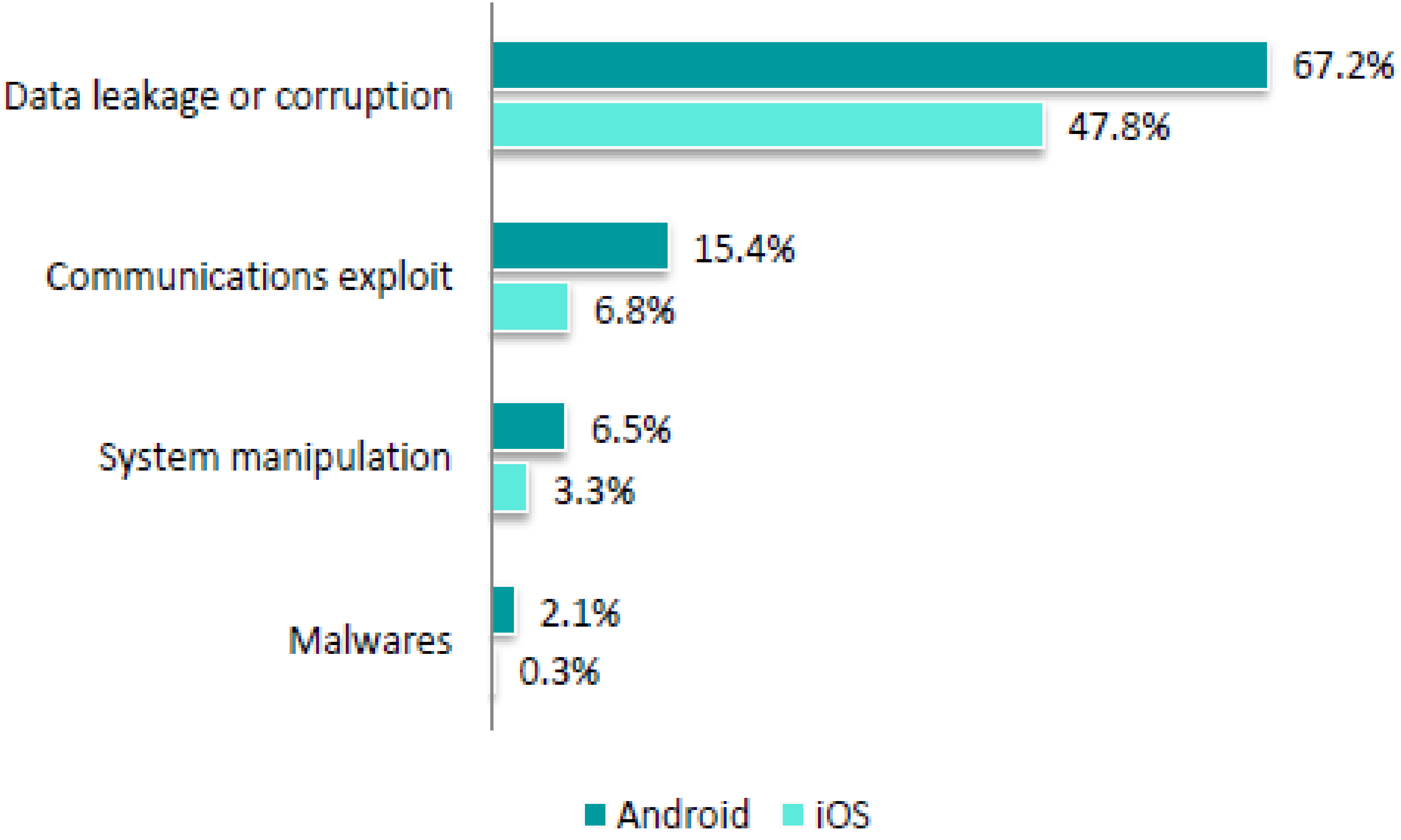


% of Apps bypassing permissions and performing data leakage

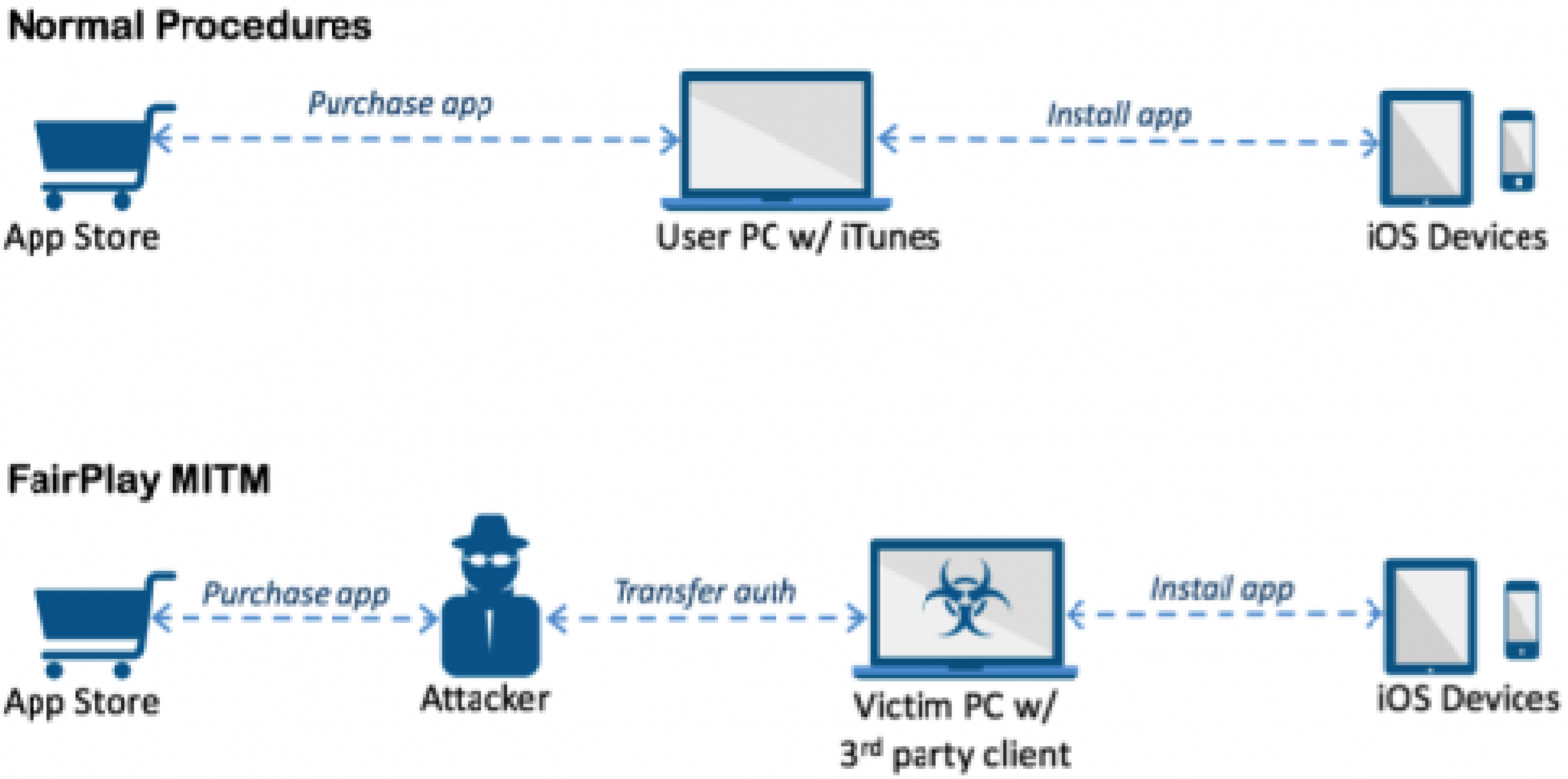
Permissions cannot be considered as safeguards for data protection nor reflect any risk level to evaluate Applications



% Apps featuring a malicious or an intrusive behaviors



FairPlay vulnerability Based



- Requires malware on user PC, installation of malicious app in App Store
- Continues to work after app removed from store
- Silently installs app on phone

Tests results about banking apps weaknesses

1. **World-Writable Files**—World writable files allow other apps to have write access to files. Approximately 33% of banking apps running on Android created or modified a file such that the file could be modified by other apps.

2. **Broken SSL Check / Sensitive Data in Transit**—Approximately 13% of banking apps running on Android are not performing proper certificate validation or hostname verification. Sensitive data could be intercepted via a MITM attack.

3. **Writable Executables**—A writable executable file in combination with another issue could lead to additional app vulnerabilities and make the app susceptible to remote code execution. Approximately 7% of tested banking apps running on Android OS had writable executable files.

Tests results about banking apps weaknesses

4. **Obfuscation**—The source code is not obfuscated for approximately 60% of banking apps running on Android. These apps can easily be reverse-engineered. *Code Obfuscation is the process of modifying an executable so that it is no longer useful to a hacker but remains fully functional*

5. **Secure Random**—Apps using the Oracle Java Cryptography Architecture (JCA) for key generation, signing, or random number generation may not receive cryptographically strong values on Android devices due to improper initialization of the pseudo-random number generator (PRNG). Approximately 73% of banking apps running on Android are vulnerable because of issues related to the Secure Random implementation.

6. **Dynamic Code Loading**—Around only 33% of banking apps running on Android use dynamic code loading within the APK (Android Application Package). It allows specifying which components of the app should not be loaded by default when the app is started. Typically, core components and additional dependencies are loaded natively at runtime, however, dynamically loaded components are only loaded as requested.

Tests results about banking apps weaknesses

7. **Cookie "HttpOnly"**—40% of banking apps running on iOS do not have the "HttpOnly" flag appropriately set. When a cookie is set with the HTTPOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies, and can help prevent attacks like XSS (cross-site scripting), as the cookie cannot be accessed via client side (for example, using a JavaScript™ snippet code).

8. **Cookie "Secure"** —54% of banking apps running on iOS do not have the "secure“ flag appropriately set. When set to true, the "secure" flag tells the browser to only send the cookie if the request is sent using a secure channel. This prevents the cookie from being transmitted over unencrypted requests.

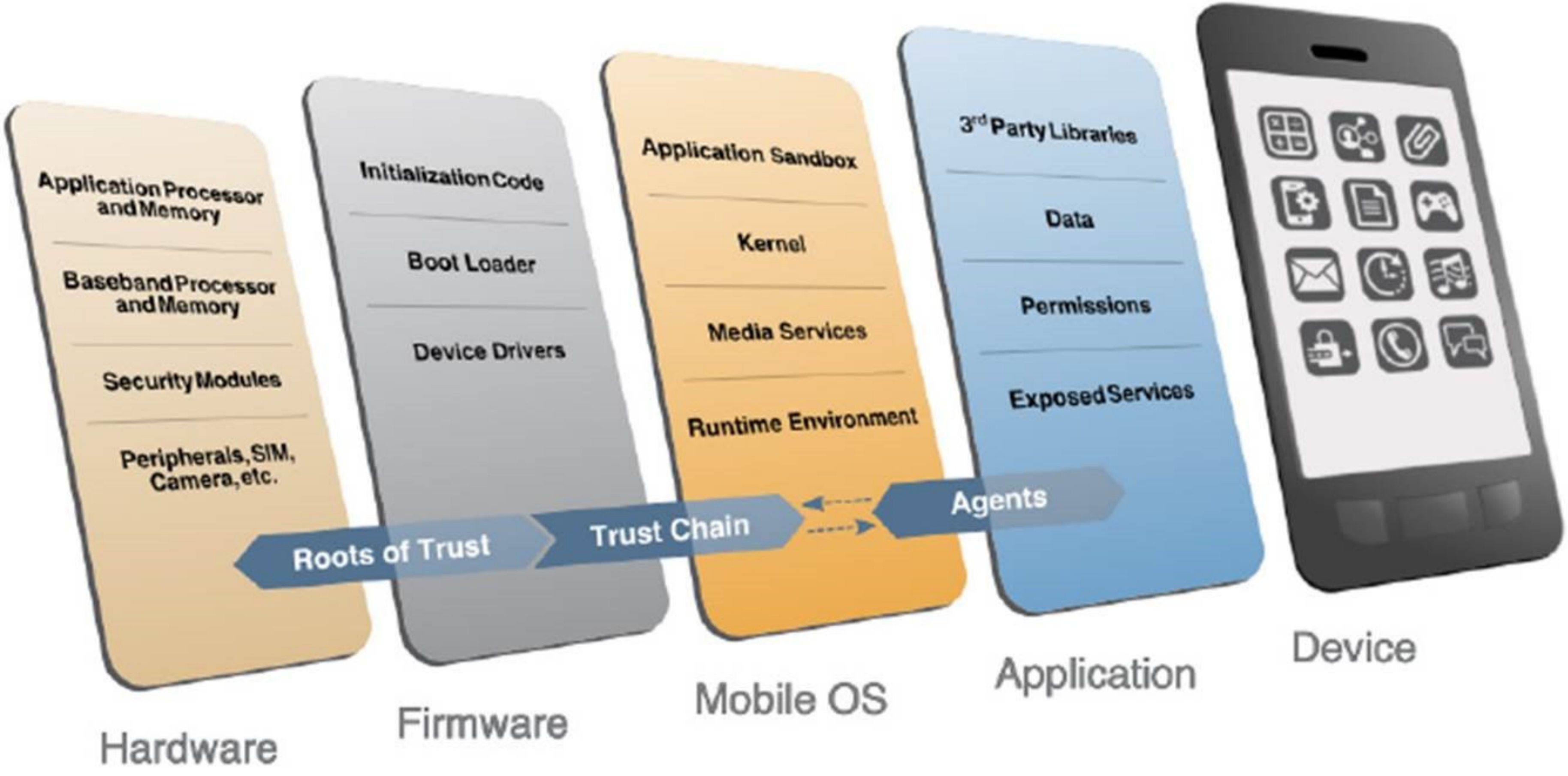
9. **Transport Layer Security Traffic with Sensitive Data**—80% of banking apps running on iOS has sensitive values intercepted while proxying SSL and Transport Layer Security (TLS) app communications, such as Username, Password, GPS coordinates, Wi-Fi Mac (Media Access Control) Address, IMEI (International Mobile Equipment Identity) Serial Number, and Phone Number. Sending sensitive data without certificate pinning creates higher risk as an attacker with network privileges, or who has compromised TLS, is better positioned to intercept data.

10. **App Transport Security (ATS)**—ATS provides secure connections between an app and the back-end server(s). It is on by default when an app is linked to iOS SDK. With ATS-enabled, HTTP connections are forced to use HTTPS (TLS v1.2), and any attempts to connect using insecure HTTP will fail. 60% of tested banking apps running on iOS had ATS globally disabled, which allows a connection regardless of HTTP or HTTPS configuration, connection to servers with lower TLS versions and a connection using cipher suites that do not support forward secrecy.

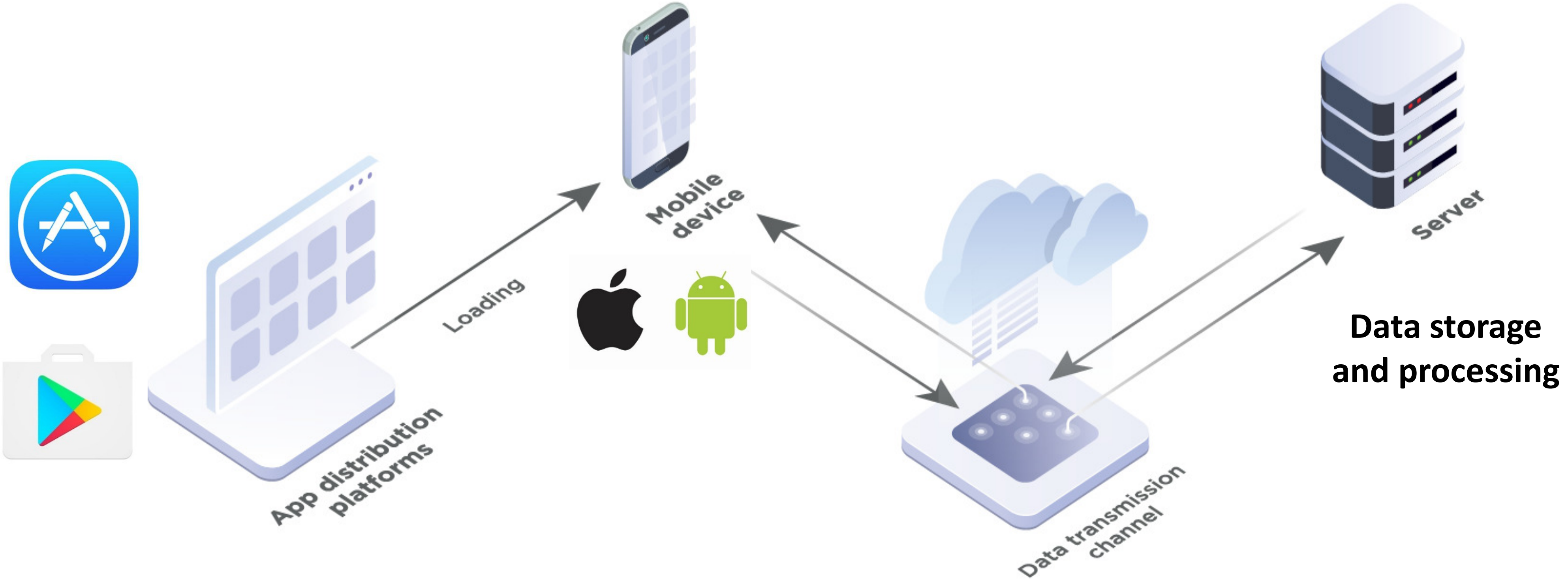
Agenda

III. Mobile Architecture

General Mobile Architecture



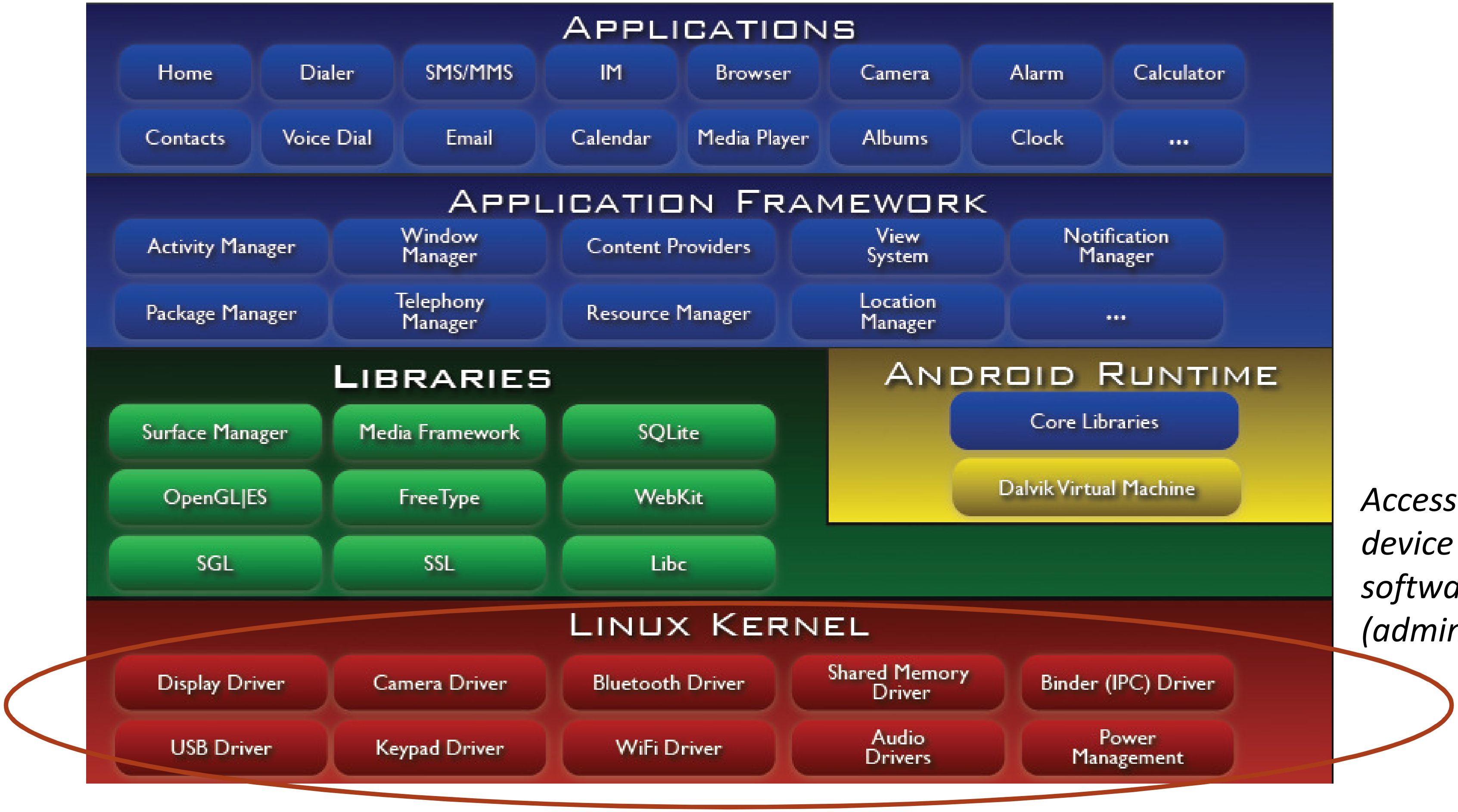
How mobile applications work



Data storage and processing

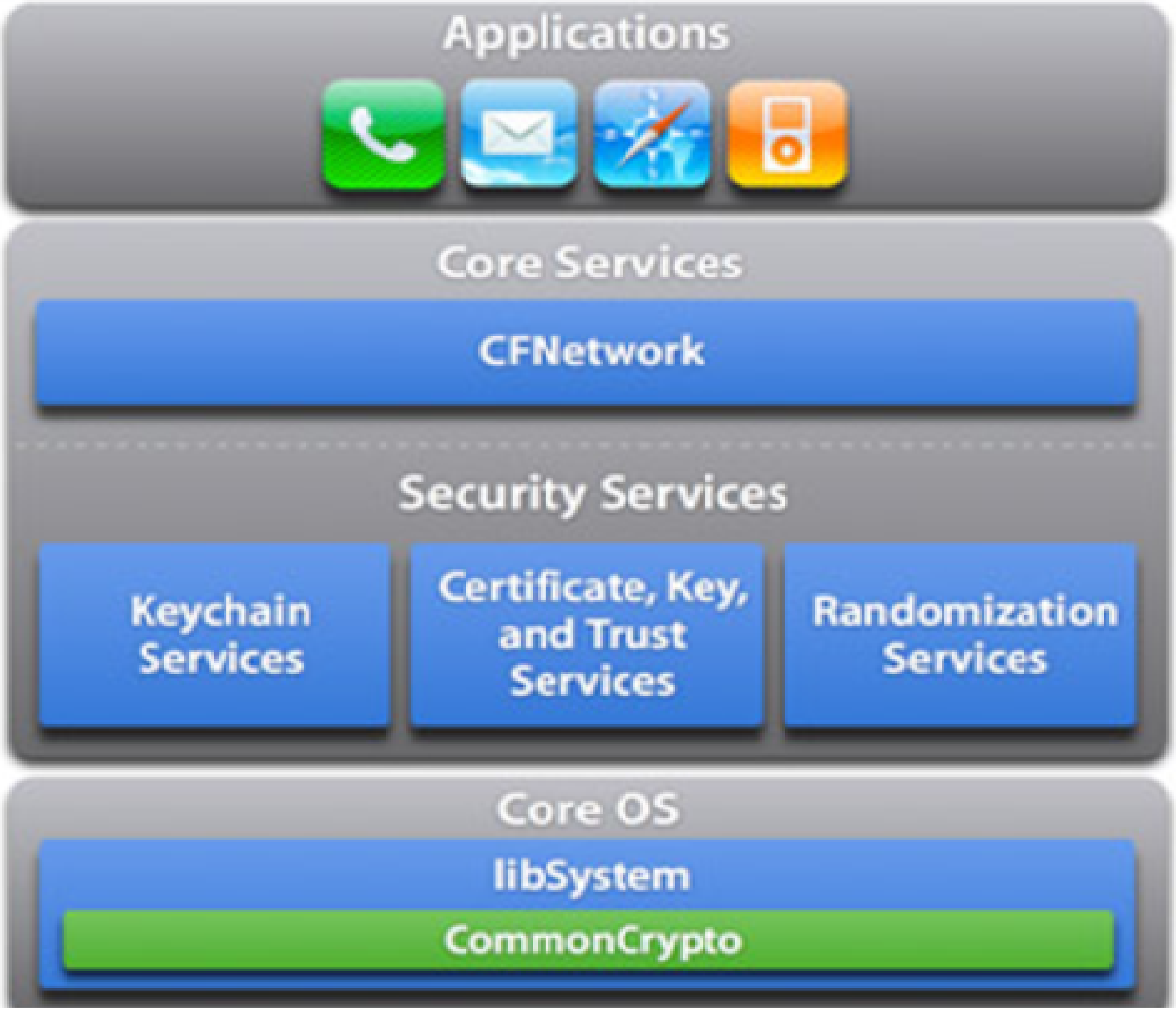
Most of applications have a **client–server architecture**

Android Architecture



Access to these features requires device rooting and allows run softwares ad super user (administrator)

iOS Architecture



Applications

Core Services

CFNetwork

Security Services

Keychain Services

Certificate, Key, and Trust Services

Randomization Services

Core OS

libSystem

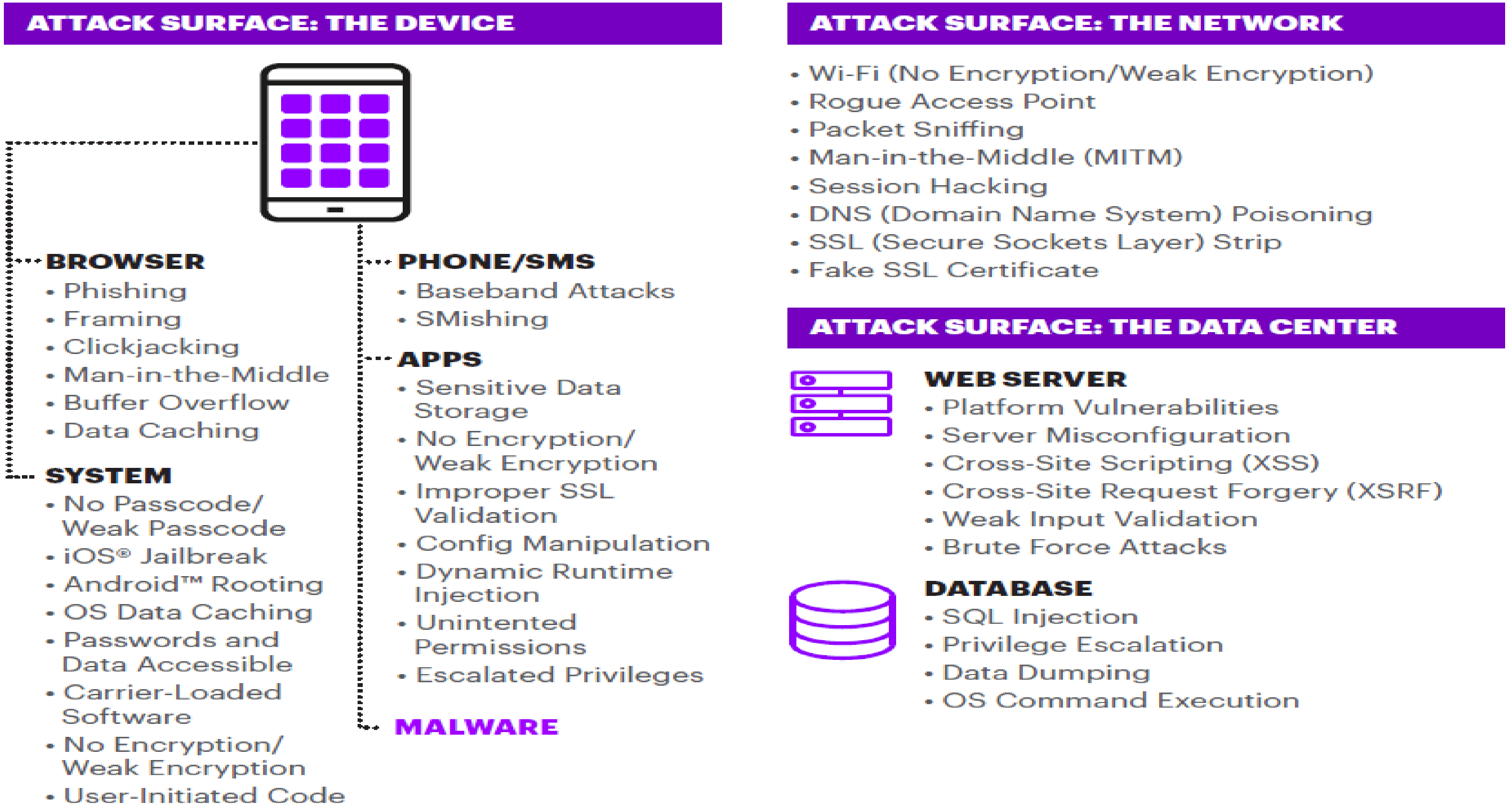
CommonCrypto

Agenda

IV. Attacks Points

The 3 points of possible attacks

3 points in the mobile technology chain where parties may exploit vulnerabilities to launch malicious attacks—the device, network and data center.



ATTACK SURFACE: THE DEVICE

BROWSER
- Phishing
- Framing
- Clickjacking
- Man-in-the-Middle
- Buffer Overflow
- Data Caching

SYSTEM
- No Passcode/ Weak Passcode
- iOS® Jailbreak
- Android™ Rooting
- OS Data Caching
- Passwords and Data Accessible
- Carrier-Loaded Software
- No Encryption/ Weak Encryption
- User-Initiated Code

PHONE/SMS
- Baseband Attacks
- SMishing

APPS
- Sensitive Data Storage
- No Encryption/ Weak Encryption
- Improper SSL Validation
- Config Manipulation
- Dynamic Runtime Injection
- Unintented Permissions
- Escalated Privileges

MALWARE

ATTACK SURFACE: THE NETWORK
- Wi-Fi (No Encryption/Weak Encryption)
- Rogue Access Point
- Packet Sniffing
- Man-in-the-Middle (MITM)
- Session Hacking
- DNS (Domain Name System) Poisoning
- SSL (Secure Sockets Layer) Strip
- Fake SSL Certificate

ATTACK SURFACE: THE DATA CENTER

WEB SERVER
- Platform Vulnerabilities
- Server Misconfiguration
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (XSRF)
- Weak Input Validation
- Brute Force Attacks

DATABASE
- SQL Injection
- Privilege Escalation
- Data Dumping
- OS Command Execution

Attacks Points

Native Operating Systems: Android

- Android is a Linux based platform developed by Google and the open handset alliance.

- Application programming for it is done exclusively in java

- Like Java apps, they can be **easily reversed** with the right tools

Reversing allows to:

  - **Get Hardcoded credentials**
  - Find out **what crypto** is used
  - How does the app **handle input or output**

- With Android bytecode can even be altered and apps repackaged

Attacks Points

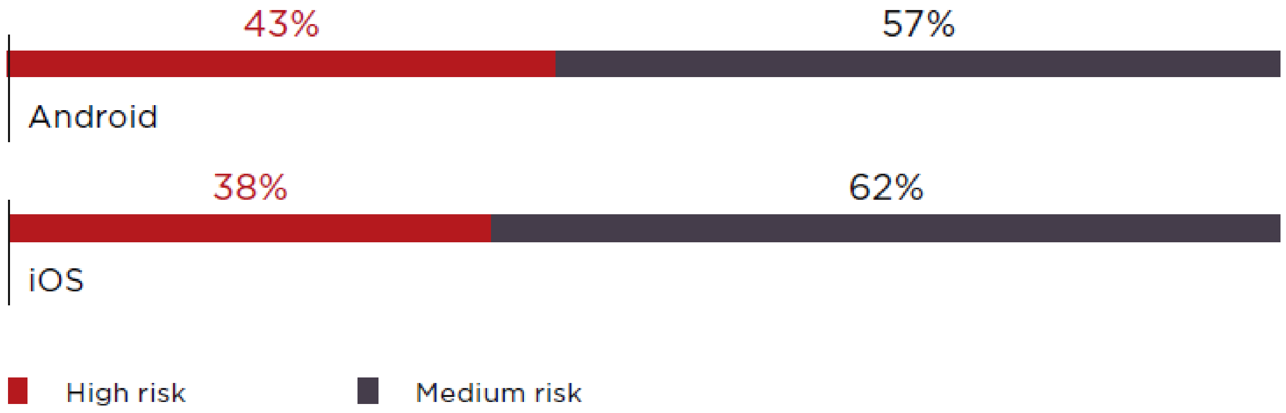
Native Operating Systems: iOS

- Based on an ARM version of XNU kernel from OSX
- Application programming for it is done in native Objective C, Swift or even C
- iOS strictly enforces application boundaries and sandboxing (definition: **Network sandboxes** monitor **network** traffic for suspicious objects and automatically submit them to the sandbox environment, where they are analyzed and assigned malware **probability scores and severity rating**)
- If Android is the Wild West, iOS is a Frontier Fort, But Once you breach the walls of the fort, you own the place….
- Reversing is also possible (Harder than Android) and the first step is **Jail-breaking**

Attacks Points

Native Operating Systems

Android applications tend to contain critical vulnerabilities slightly more often than those written for iOS (**43%** vs. **38%**)

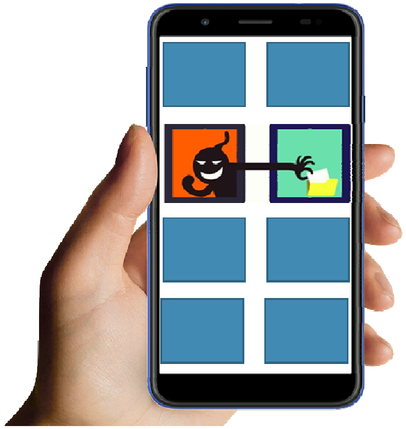


Source: Pt Security_Vulnerabilities
and threats in mobile Applications_2019

Attacks Points

Mobile environment

- Communication with other apps, or access the application directories of other app



- Sniffing Not encrypted Sensitive Data

Sensitive data
Attacker's PC

- Binary:
  - Reverse engineering to understand the binary
  - Find vulnerabilities that may be exploitable

Attacks Points

Mobile environment

- Platform:
  - o Function hooking
  - o Malware installation
  - o Mobile botnets
  - o Application architecture decisions based on platform

Attacks Points

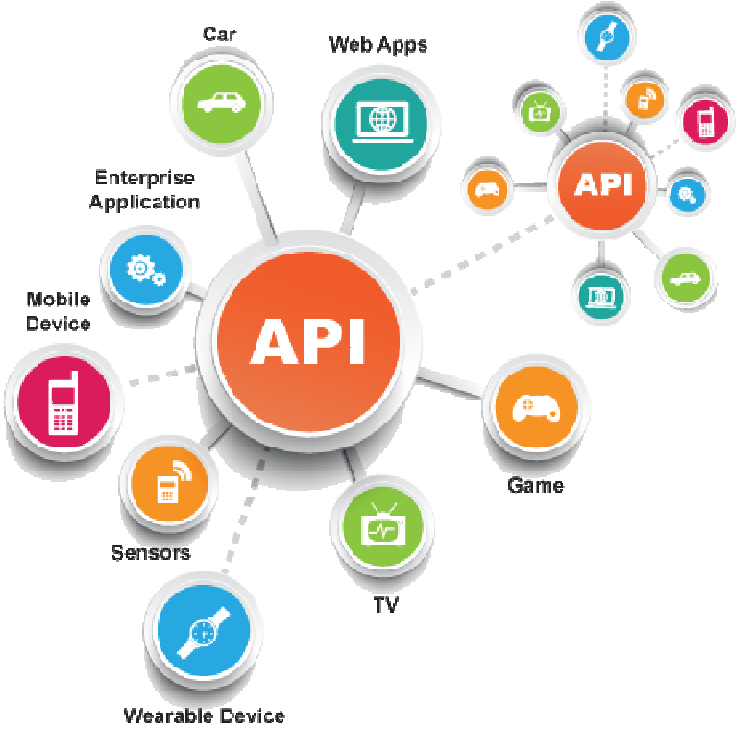
Server Side

- Data Storage:
  - Key stores
  - Application file system
  - Application database
  - Caches
  - Configuration files

Attacks Points

Server Side

- Application Programming Interface (API):
  - Broken authentication
  - Excessive data exposure
  - Broken function level authorization
  - Security misconfiguration
  - Injection
  - Insufficient logging and monitoring



Agenda

Recommendations

Recommendations

1. **Keep your system updated** An update is available? Install it right away; it probably holds corrections of your current version’s detected breaches.
2. **Don’t download applications outside official stores** The App Store and the Google Play Store provide both a first level security evaluation, prior to publishing the App on their stores. In opposition to third party stores, on which a great part of their available applications are malwares.
3. **Set a screen lock model or password** to avoid an physical access to the device by an attacker
4. **Use biometric authentication** (fingerprint, voice, or face) if your device supports it
5. **Stay vigilant when going through your inbox**. Carefully check links before opening them, if the linked address contains any misspellings, the email is not genuine.

Recommendations

6. **Verify applications' permissions** Asking a permission should be related with the app's main purpose. For instance, an App that is meant to edit pictures shouldn't have access to your microphone.
7. **Don't do sensitive operations through open networks** You have to transfer money to an important supplier's bank account? Doing so during your lunch break while connected to the restaurant's Wi-Fi might expose you to an attack such as a man-in-the-middle. You should wait to be connected to a secured network.
8. **Don't root / jailbreak your device** A "cracked" device gives an easier access to the information it contains and increases its vulnerability.
9. **Get a security solution** Make sure that you are using a solution that will help you assure apps legitimacy and protect from "zero-day" attacks.

Conclusion

You can not be 100% safe, but you can make it hard –

Defense in Depth

Trustwave SpiderLabs

Thank You