Comparative analysis of Android and iOS from security viewpoint

The Seminar is produced by
Jiaxiang Li(s4632882) | Chaohui Lin(s4670461) | Zian Huang(s4679385)

Computer Science Review 40 (2021) 100372

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review Article

Comparative analysis of Android and iOS from security viewpoint

Shivi Garg [a,b], Niyati Baliyan [b,*]

[a] Faculty of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, India
[b] Information Technology Department, Indira Gandhi Delhi Technical University for Women, Delhi, India

ARTICLE INFO

Article history:
Received 5 May 2020
Accepted 23 January 2021
Available online 13 February 2021

Keywords:
Android
iOS
Malware
Security
Smartphones
Vulnerabilities

ABSTRACT

Smartphone usage has increased exponentially in the recent years. Android and iOS are the most popular smartphone platforms, while the ease of use along with the computational power to handle a wide array of applications attracts millions of users worldwide, also raises the security concerns on these platforms. This paper presents a comparative analysis between Android and iOS on a wide range of security aspects. It analyzes data for the period 2015-2019 and gives a detailed snapshot of not only the quantum of vulnerabilities, but also their impact. In addition, the paper leverages the well-established security triad i.e. CIA (Confidentiality, Integrity, Availability) to compare both the operating systems. The comprehensive and pragmatic approach taken in the paper makes it easier to infer that Android is more susceptible to security breaches and malware attacks as compared to iOS. Hence, researchers should divert their efforts and focus on finding solutions to problems pertaining to Android. The paper concludes by laying down future research directions and scope of work, which can be leveraged not only by application developers, but also by researchers. This will help make Android safer for users and will further increase its demand as a mobile operating system.

Contents

1. Introduction

Technological advancements in smartphones are at par with personal computers. With increased computing power, smartphones are becoming ubiquitous part of daily life. Hence, number of smartphone users has exponentially risen in the last five years. The fact is established with the help of Statista [1], where the

[*] Corresponding author.
E-mail addresses: shivi002phd16@igdtuw.ac.in (S. Garg), niyatibaliyan@igdtuw.ac.in (N. Baliyan).

https://doi.org/10.1016/j.cosrev.2021.100372
1574-0137/© 2021 Elsevier Inc. All rights reserved.

Shivi Garg, Niyati Baliyan

Comparative analysis of Android and iOS from security viewpoint

Computer Science Review

Available here

The hook

- Do you often use your mobile phone or how often do you download a software?
- Do Android and iOS have an impact on our life?

Content

- Introduction
- Related work and research
- Comparison between Android and iOS
- Software vulnerabilities common in Android and iOS
- Data collection
- Vulnerability trends in Android vs. iOS
- Malware attacks in Android and iOS
- Research direction and future scope

Chapter 1: Introduction of Android and iOS

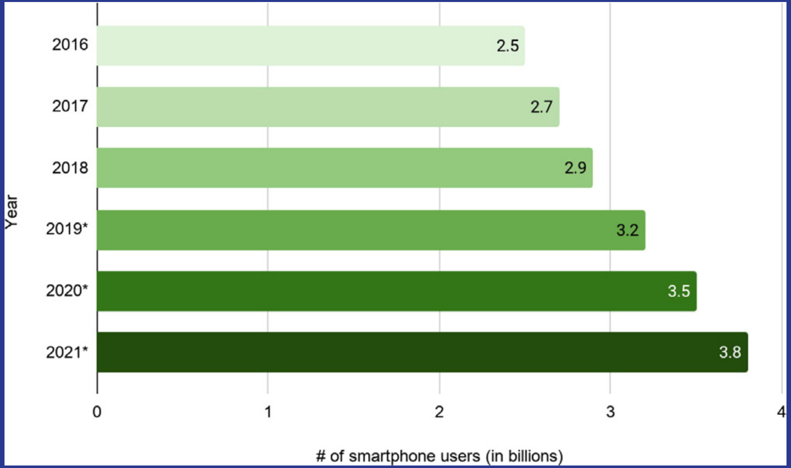


Fig. 1. Number of Smartphone users by year.

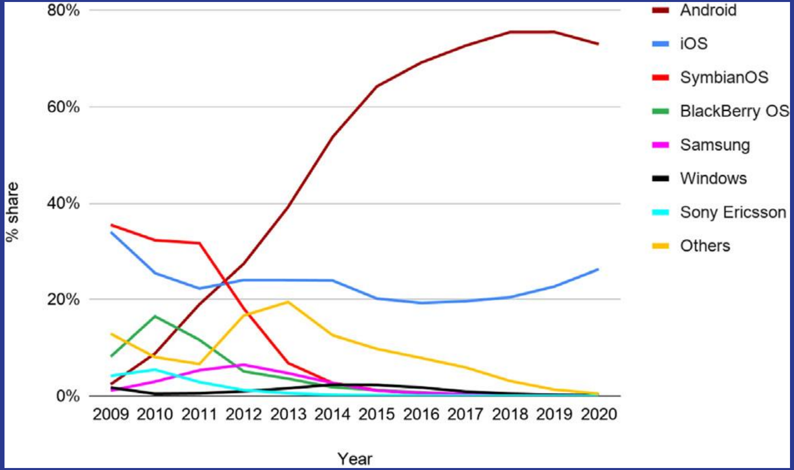


Fig. 2. Market share of different Mobile OS from 2009–2020.

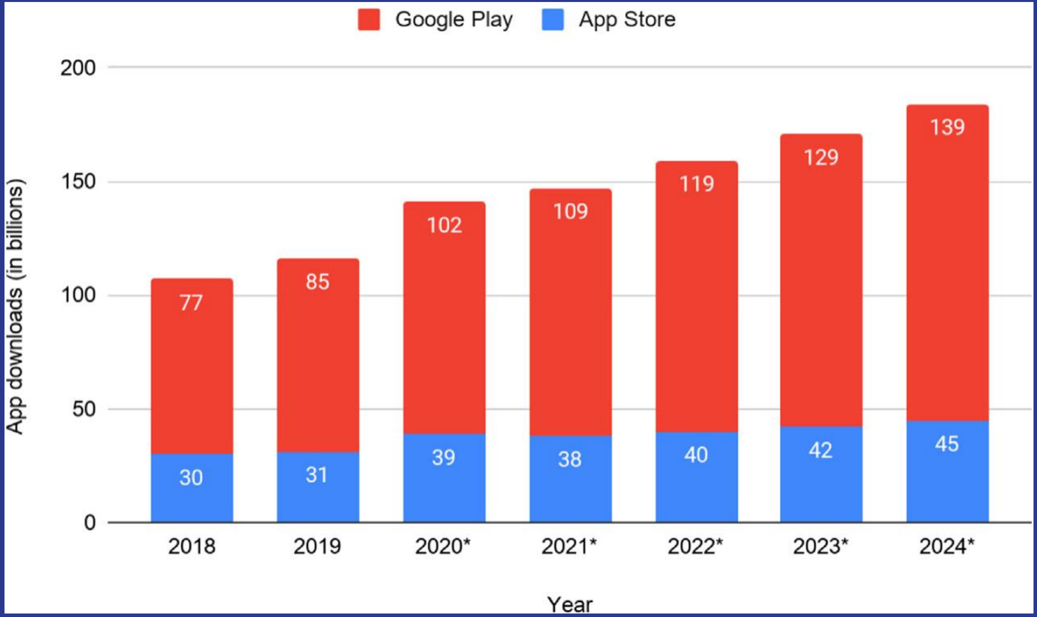


Fig. 3. Mobile application downloads from 2018 to 2024 by store worldwide.

Chapter 2: Related research and work

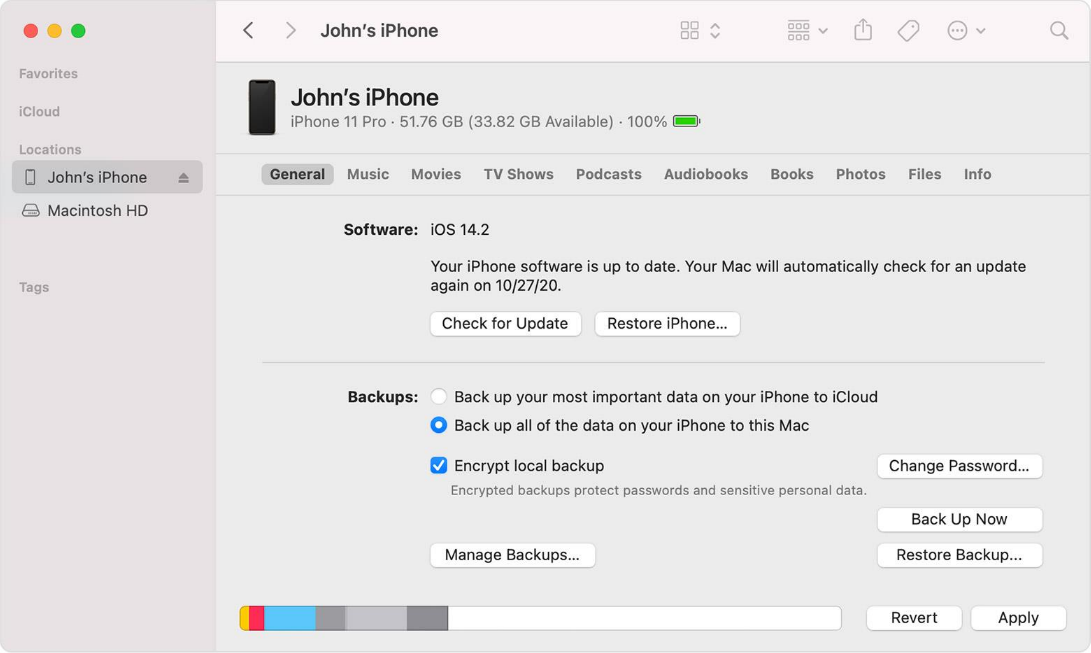
- **Web view vulnerabilities**
  by Hidhaya and Geetha
  Result: The team achieved a detection accuracy of 85%, but run time/dynamic behaviour of the apps wasn’t analyzed.
- **Smartphone security vulnerabilities based on the apps**
  by L Zhang’s team
  Result: They are able to demonstrate SSL validation, HTML5-based application attacks, but only use function to analyze SSL vulnerabilities.
- **Survey of smartphone security and the attacks on mobile application**
  by Ahvanooey‘s group.
  Result: Survey is not very comprehensive and doesn’t provide clear viewpoint of the mobile platform vulnerabilities.
- … …

Obtaining root privileges of Android

STOP
WARNING
STOP
Make Just 1 Tiny Mistake While Rooting
Your Expensive Android Device And You
Could Risk Turning It Into A Brick, $#^@!

- The customized Linux kernel of Android can make it vulnerable when the root privileges are obtained.

Attacks on backup that store on the iTunes



John's iPhone

iPhone 11 Pro · 51.76 GB (33.82 GB Available) · 100%

Software: iOS 14.2

Your iPhone software is up to date. Your Mac will automatically check for an update again on 10/27/20.

Backups:
○ Back up your most important data on your iPhone to iCloud
● Back up all of the data on your iPhone to this Mac

☑ Encrypt local backup

Encrypted backups protect passwords and sensitive personal data.

backup stored on the iTunes can be easily attacked by using different techniques.
e.g. Brute-force attack

Research by Peter Teufl, Thomas Zefferer & Christof Stromberger

Chapter 3: Comparison between Android and iOS

3.1 System architecture
3.2 Security
3.3 Isolation mechanism
3.4 Encryption mechanism
3.5 App permissions
3.6 Auto erase mechanism
3.7 Application provenance

3.1 System architecture

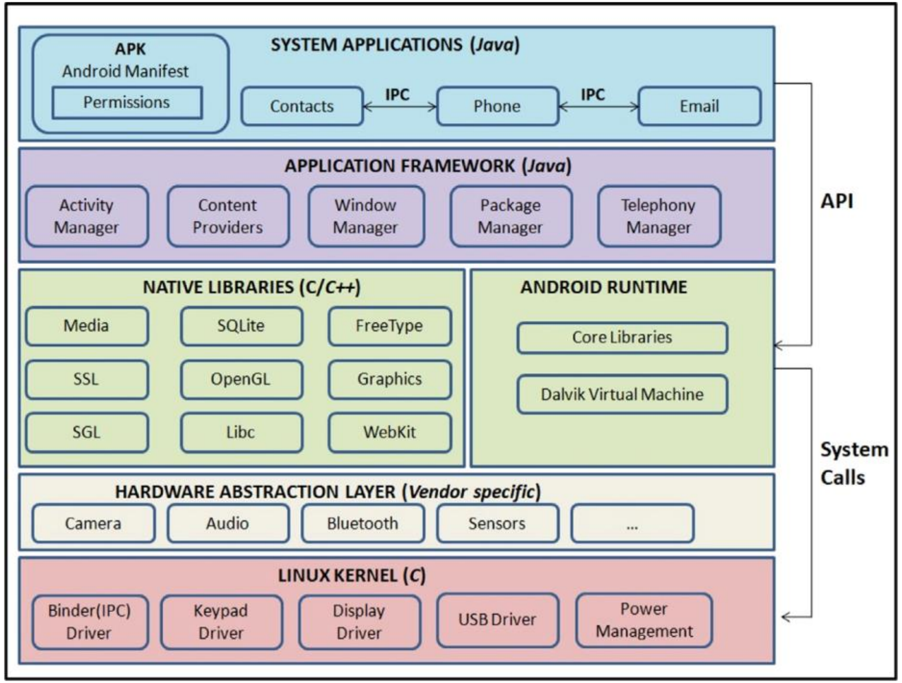


Fig. 4. Android architecture.

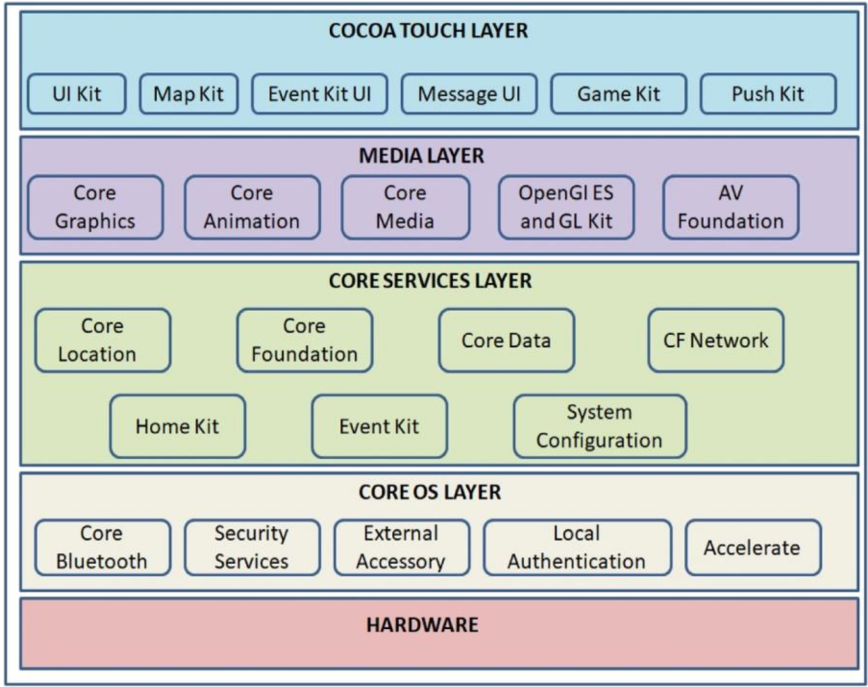


Fig. 5. iOS architecture.

3.2 Security

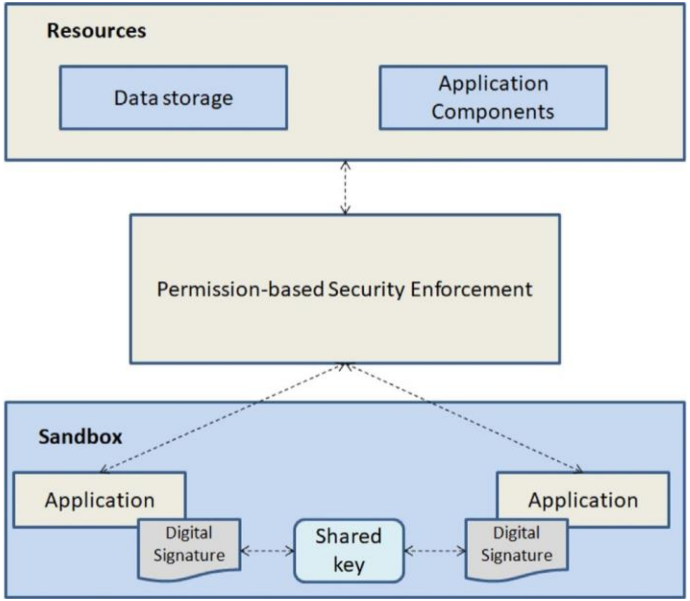


Fig. 6. Android security model.

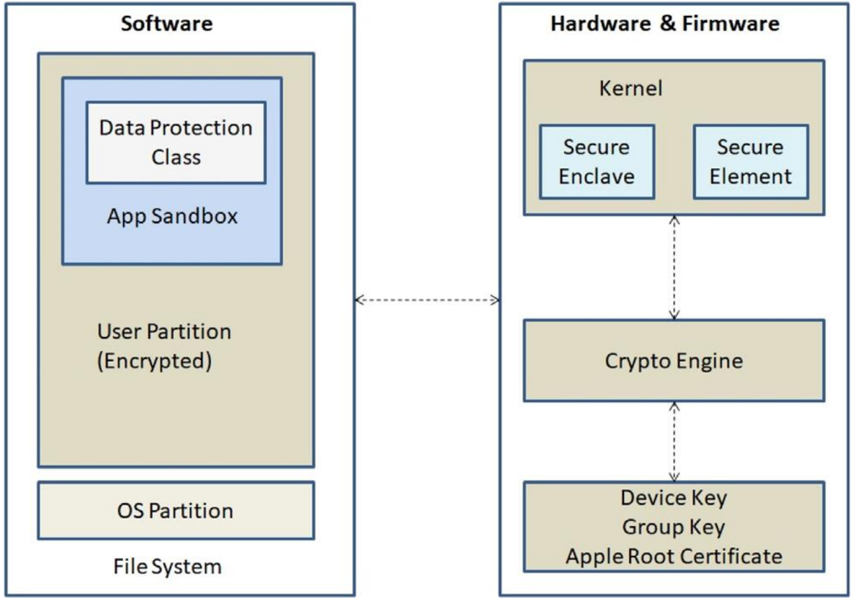
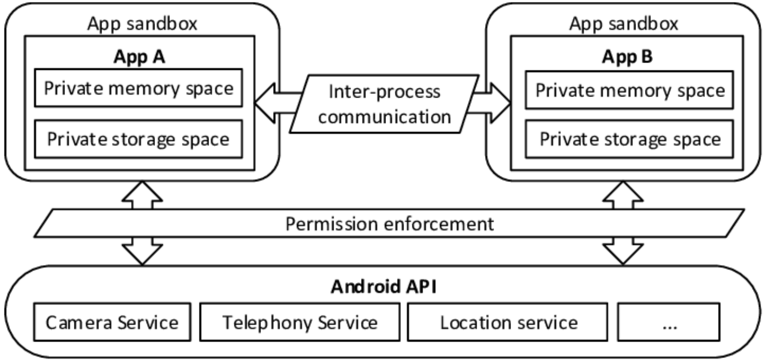
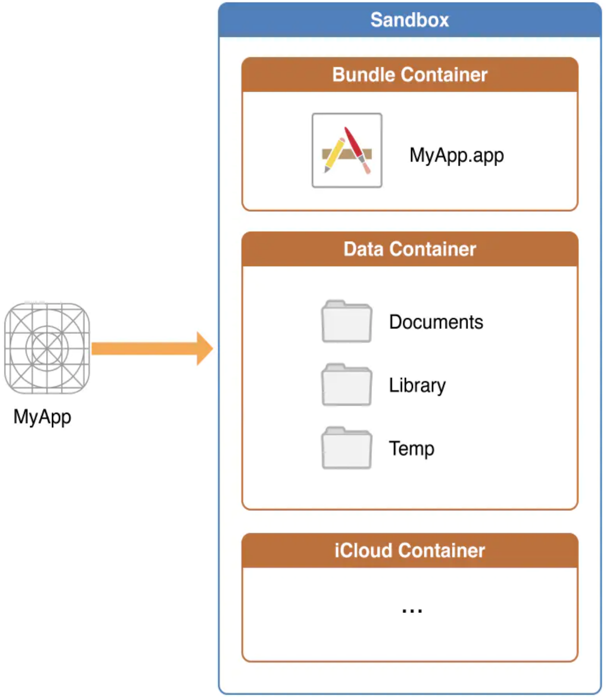


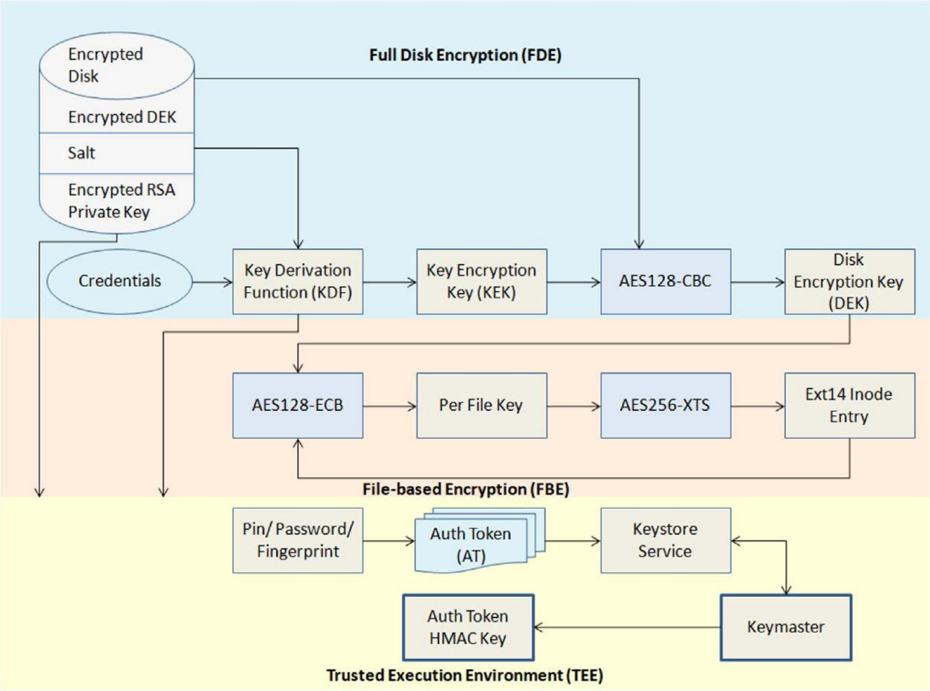
Fig. 7. iOS security model.

3.3 Isolation mechanism





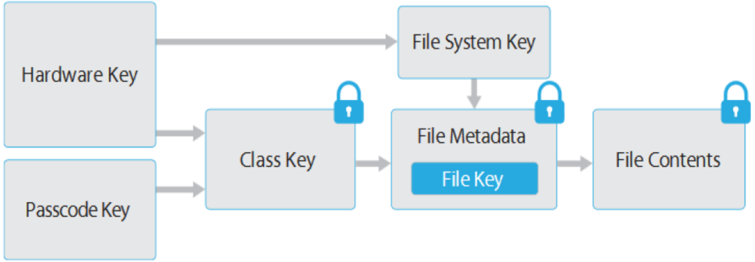
3.4 Encryption mechanism



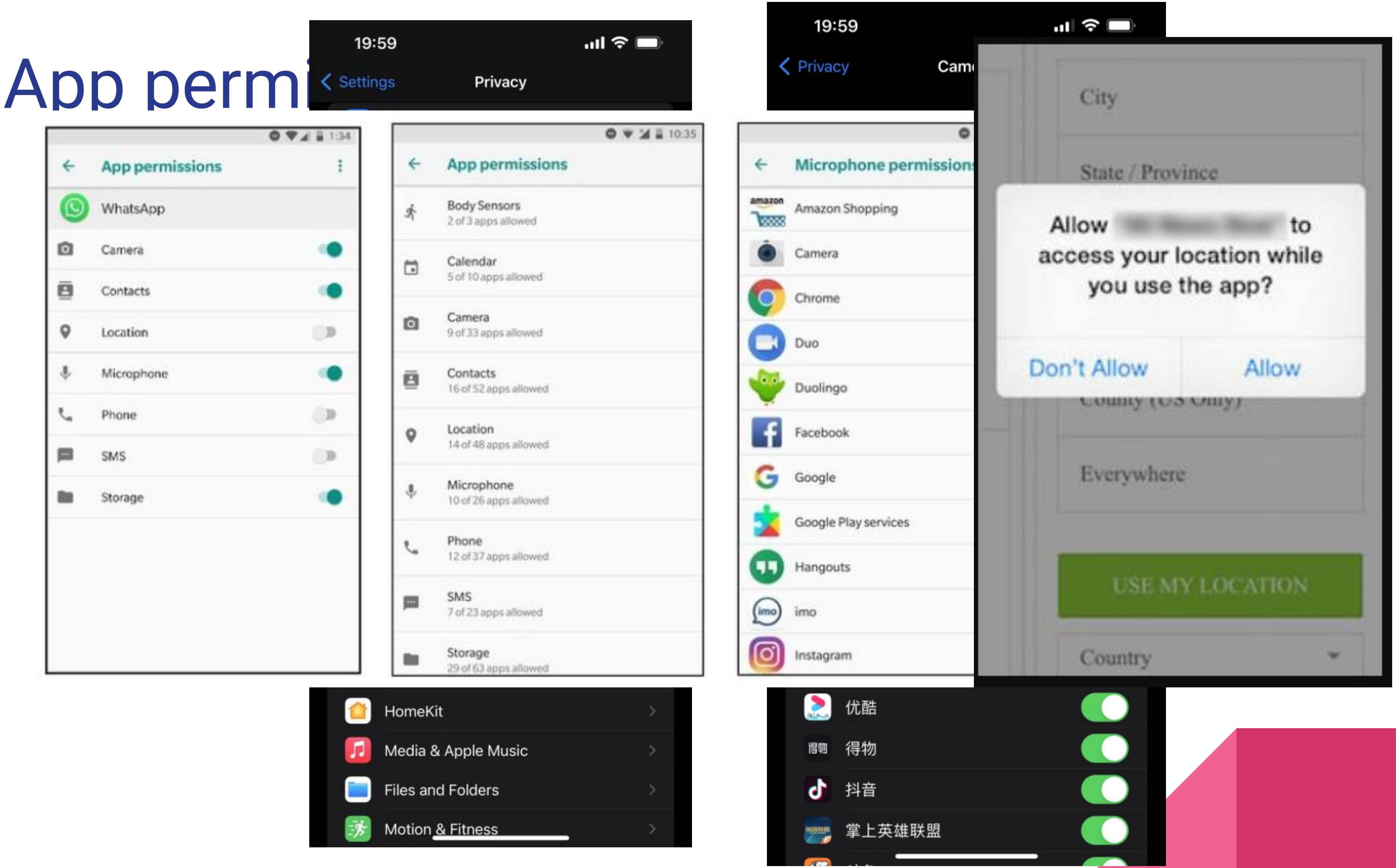
Full Disk Encryption (FDE)
Encrypted Disk
Encrypted DEK
Salt
Encrypted RSA Private Key
Credentials
Key Derivation Function (KDF)
Key Encryption Key (KEK)
AES128-CBC
Disk Encryption Key (DEK)
AES128-ECB
Per File Key
AES256-XTS
Ext14 Inode Entry
File-based Encryption (FBE)
Pin/ Password/ Fingerprint
Auth Token (AT)
Keystore Service
Auth Token HMAC Key
Keymaster
Trusted Execution Environment (TEE)

Fig. 8. Encryption model used in Android.

Fig. 9. Encryption model used in iOS.



3.5. App permi



19:59

Settings Privacy

App permissions

WhatsApp

Camera	●
Contacts	●
Location	○
Microphone	●
Phone	○
SMS	○
Storage	●

App permissions

- Body Sensors
  2 of 3 apps allowed
- Calendar
  5 of 10 apps allowed
- Camera
  9 of 33 apps allowed
- Contacts
  16 of 52 apps allowed
- Location
  14 of 48 apps allowed
- Microphone
  10 of 26 apps allowed
- Phone
  12 of 37 apps allowed
- SMS
  7 of 23 apps allowed
- Storage
  29 of 63 apps allowed

- HomeKit
- Media & Apple Music
- Files and Folders
- Motion & Fitness

19:59

Privacy Cam

Microphone permissions

- Amazon Shopping
- Camera
- Chrome
- Duo
- Duolingo
- Facebook
- Google
- Google Play services
- Hangouts
- imo
- Instagram

优酷	●
得物	●
抖音	●
掌上英雄联盟	●

City

State / Province

Allow [redacted] to access your location while you use the app?

Don't Allow | Allow

County (US Only)

Everywhere

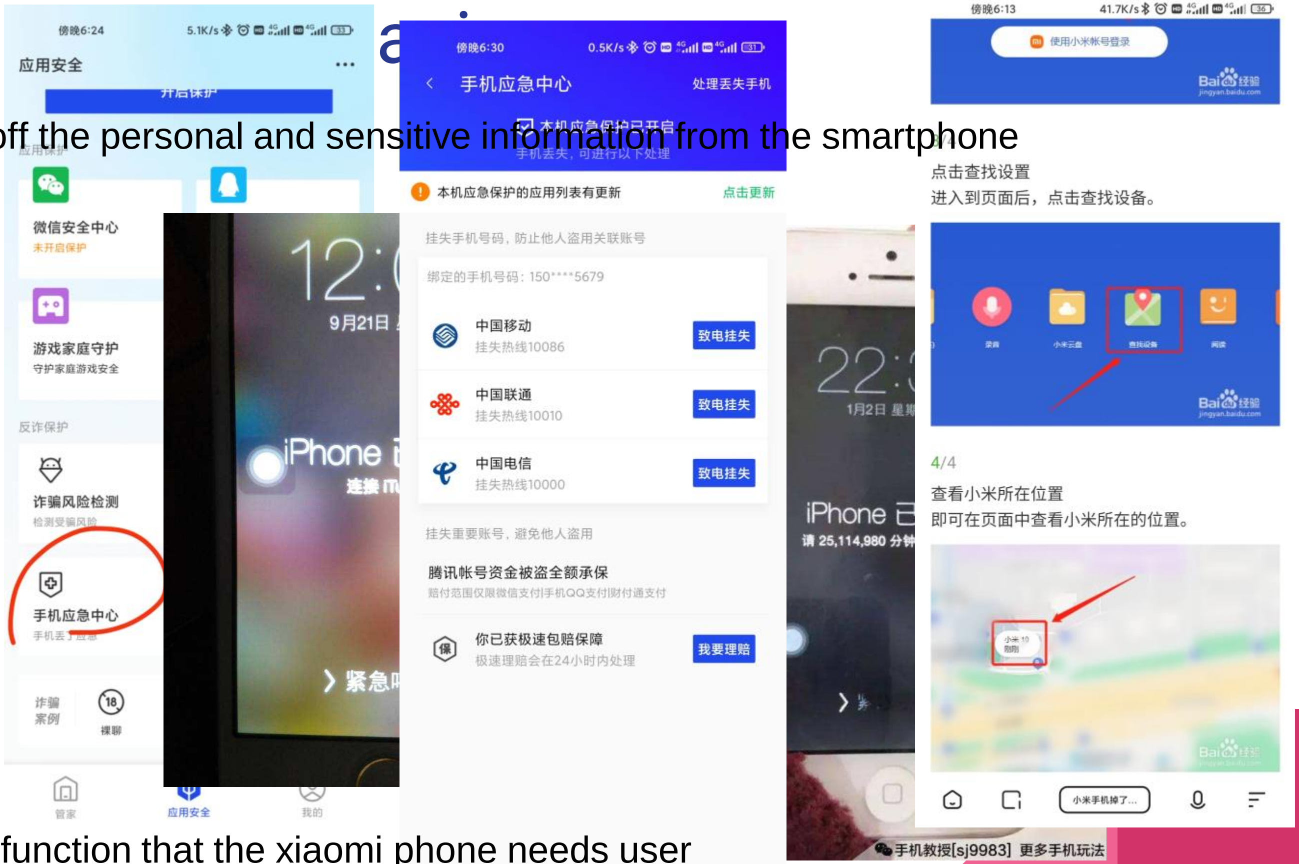
USE MY LOCATION

Country

3.6. Auto a

it can wipe off the personal and sensitive information from the smartphone

First and second pictures are about an app which is called tencent housekeeper. There is no function which can auto initialise the phone.



傍晚6:24

应用安全

开启保护

应用保护

微信安全中心
未开启保护

游戏家庭守护
守护家庭游戏安全

反诈保护

诈骗风险检测
检测受骗风险

手机应急中心
手机丢了应急

诈骗案例

裸聊

12:

9月21日

iPhone 已

紧急

傍晚6:30

手机应急中心

处理丢失手机

☑ 本机应急保护已开启
手机丢失，可进行以下处理

本机应急保护的应用列表有更新 点击更新

挂失手机号码，防止他人盗用关联账号

绑定的手机号码：150****5679

中国移动 挂失热线10086	致电挂失
中国联通 挂失热线10010	致电挂失
中国电信 挂失热线10000	致电挂失

挂失重要账号，避免他人盗用

腾讯帐号资金被盗全额承保
赔付范围仅限微信支付|手机QQ支付|财付通支付

你已获极速包赔保障
极速理赔会在24小时内处理
我要理赔

22:

1月2日 星期

iPhone 已
请 25,114,980 分钟

傍晚6:13

使用小米帐号登录

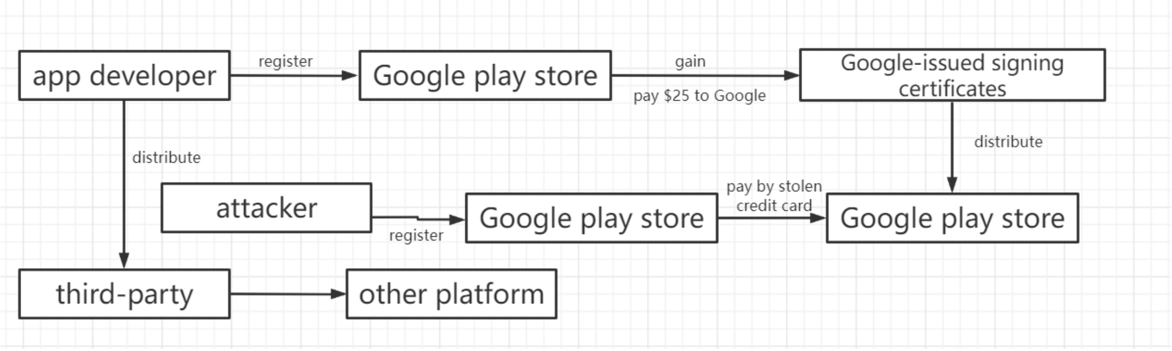
点击查找设置
进入到页面后，点击查找设备。

4/4
查看小米所在位置
即可在页面中查看小米所在的位置。

手机教授[sj9983] 更多手机玩法

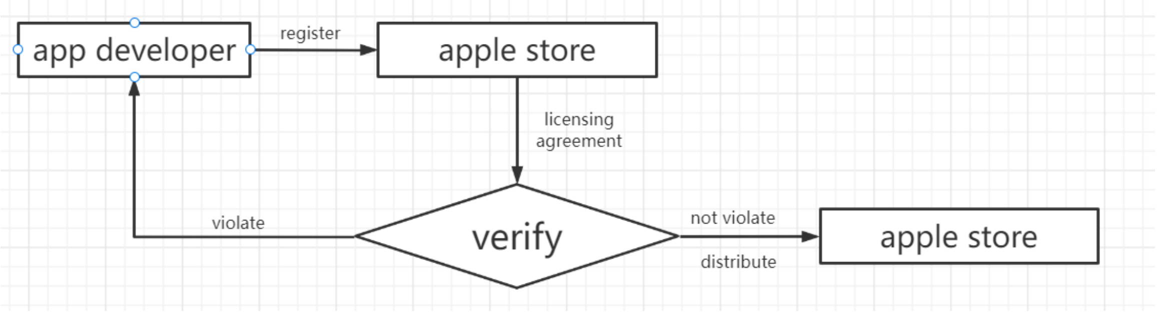
The third picture is a function that the xiaomi phone needs user to login the official website and try to locate phone.

Android

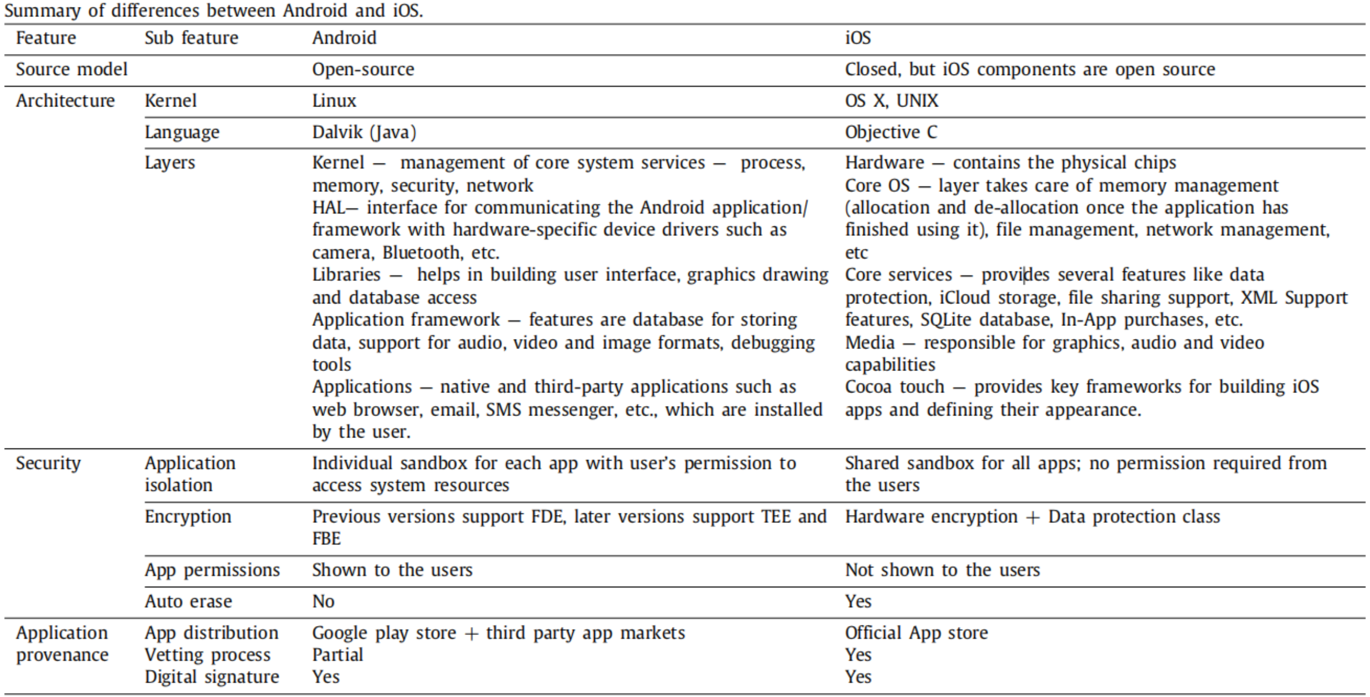


app developer
register
Google play store
gain
pay $25 to Google
Google-issued signing certificates
distribute
distribute
attacker
register
Google play store
pay by stolen credit card
Google play store
third-party
other platform

IOS



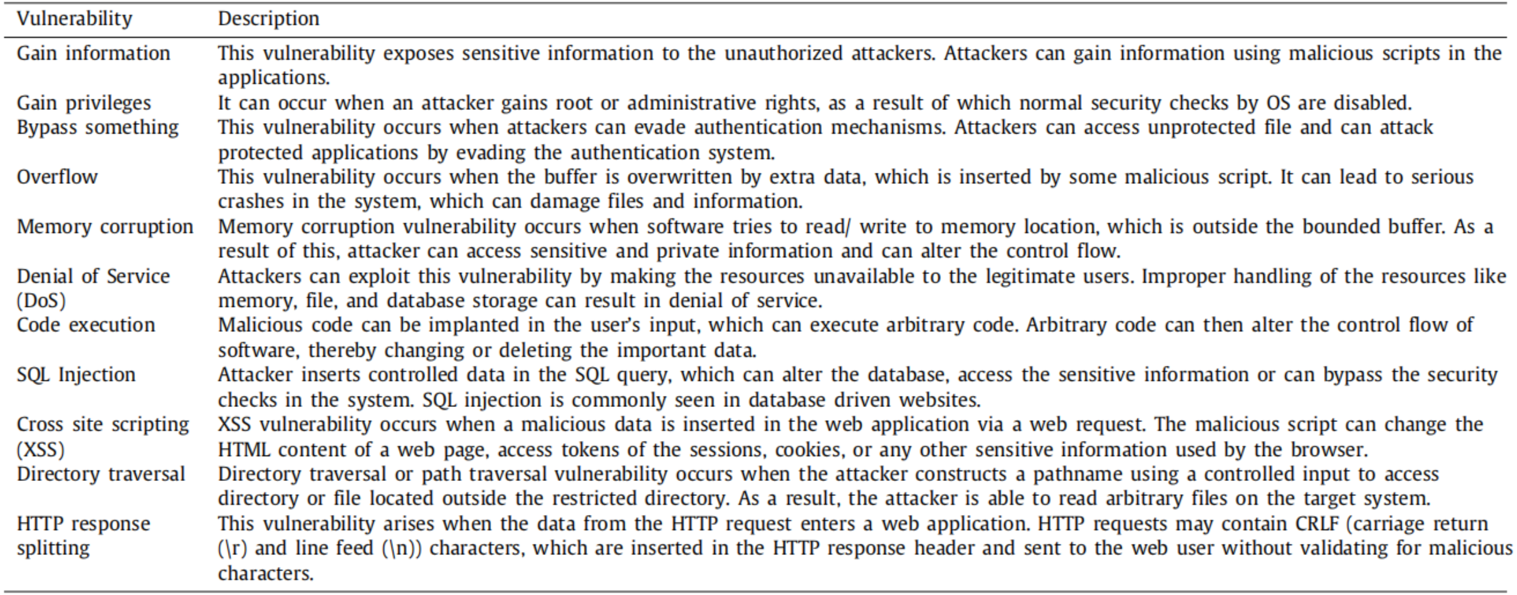
app developer
register
apple store
licensing agreement
violate
verify
not violate
distribute
apple store



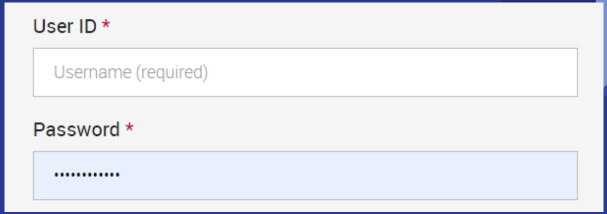
Summary of differences between Android and iOS.

Feature	Sub feature	Android	iOS
Source model		Open-source	Closed, but iOS components are open source
Architecture	Kernel	Linux	OS X, UNIX
	Language	Dalvik (Java)	Objective C
	Layers	Kernel — management of core system services — process, memory, security, network HAL— interface for communicating the Android application/framework with hardware-specific device drivers such as camera, Bluetooth, etc. Libraries — helps in building user interface, graphics drawing and database access Application framework — features are database for storing data, support for audio, video and image formats, debugging tools Applications — native and third-party applications such as web browser, email, SMS messenger, etc., which are installed by the user.	Hardware — contains the physical chips Core OS — layer takes care of memory management (allocation and de-allocation once the application has finished using it), file management, network management, etc Core services — provides several features like data protection, iCloud storage, file sharing support, XML Support features, SQLite database, In-App purchases, etc. Media — responsible for graphics, audio and video capabilities Cocoa touch — provides key frameworks for building iOS apps and defining their appearance.
Security	Application isolation	Individual sandbox for each app with user's permission to access system resources	Shared sandbox for all apps; no permission required from the users
	Encryption	Previous versions support FDE, later versions support TEE and FBE	Hardware encryption + Data protection class
	App permissions	Shown to the users	Not shown to the users
	Auto erase	No	Yes
Application provenance	App distribution	Google play store + third party app markets	Official App store
	Vetting process	Partial	Yes
	Digital signature	Yes	Yes

Common vulnerabilities in Android and iOS.



Vulnerability	Description
Gain information	This vulnerability exposes sensitive information to the unauthorized attackers. Attackers can gain information using malicious scripts in the applications.
Gain privileges	It can occur when an attacker gains root or administrative rights, as a result of which normal security checks by OS are disabled.
Bypass something	This vulnerability occurs when attackers can evade authentication mechanisms. Attackers can access unprotected file and can attack protected applications by evading the authentication system.
Overflow	This vulnerability occurs when the buffer is overwritten by extra data, which is inserted by some malicious script. It can lead to serious crashes in the system, which can damage files and information.
Memory corruption	Memory corruption vulnerability occurs when software tries to read/ write to memory location, which is outside the bounded buffer. As a result of this, attacker can access sensitive and private information and can alter the control flow.
Denial of Service (DoS)	Attackers can exploit this vulnerability by making the resources unavailable to the legitimate users. Improper handling of the resources like memory, file, and database storage can result in denial of service.
Code execution	Malicious code can be implanted in the user's input, which can execute arbitrary code. Arbitrary code can then alter the control flow of software, thereby changing or deleting the important data.
SQL Injection	Attacker inserts controlled data in the SQL query, which can alter the database, access the sensitive information or can bypass the security checks in the system. SQL injection is commonly seen in database driven websites.
Cross site scripting (XSS)	XSS vulnerability occurs when a malicious data is inserted in the web application via a web request. The malicious script can change the HTML content of a web page, access tokens of the sessions, cookies, or any other sensitive information used by the browser.
Directory traversal	Directory traversal or path traversal vulnerability occurs when the attacker constructs a pathname using a controlled input to access directory or file located outside the restricted directory. As a result, the attacker is able to read arbitrary files on the target system.
HTTP response splitting	This vulnerability arises when the data from the HTTP request enters a web application. HTTP requests may contain CRLF (carriage return (\r) and line feed (\n)) characters, which are inserted in the HTTP response header and sent to the web user without validating for malicious characters.



User ID *

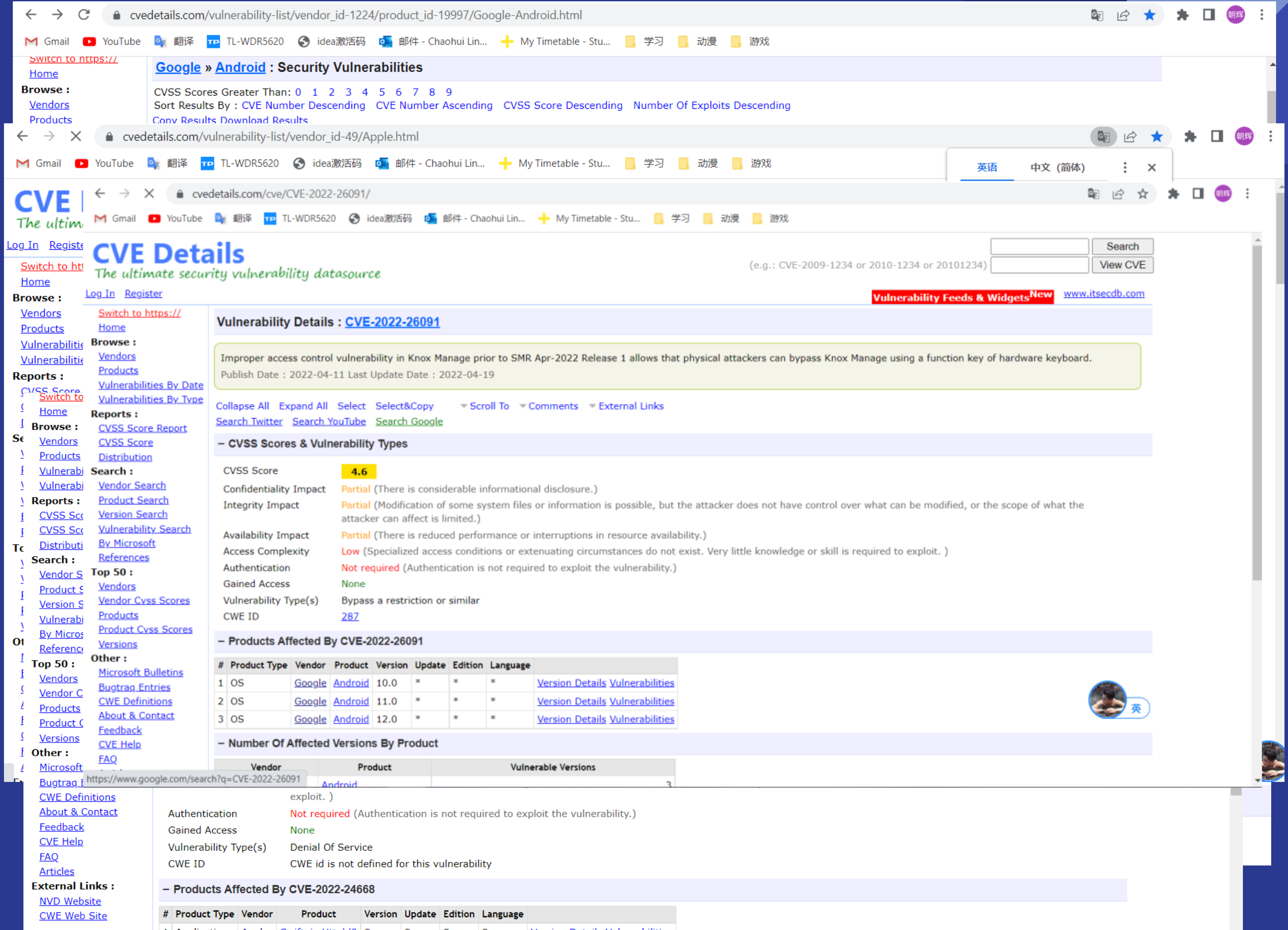
Username (required)

Password *

....

```
SELECT *
FROM table
WHERE username =  ‘Jerry’  and password = 111
```

```
SELECT *
FROM table
WHERE username =  ‘Jerry’  and password = 222 or 1=1
```



cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html

Google » Android : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

cvedetails.com/vulnerability-list/vendor_id-49/Apple.html

cvedetails.com/cve/CVE-2022-26091/

CVE Details

The ultimate security vulnerability datasource

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)

Vulnerability Details : CVE-2022-26091

Improper access control vulnerability in Knox Manage prior to SMR Apr-2022 Release 1 allows that physical attackers can bypass Knox Manage using a function key of hardware keyboard.

Publish Date : 2022-04-11 Last Update Date : 2022-04-19

– CVSS Scores & Vulnerability Types

CVSS Score	4.6
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar
CWE ID	287

– Products Affected By CVE-2022-26091

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	OS	Google	Android	10.0	*	*	*	Version Details Vulnerabilities
2	OS	Google	Android	11.0	*	*	*	Version Details Vulnerabilities
3	OS	Google	Android	12.0	*	*	*	Version Details Vulnerabilities

– Number Of Affected Versions By Product

Vendor	Product	Vulnerable Versions
	Android	3

exploit.)

Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service
CWE ID	CWE id is not defined for this vulnerability

– Products Affected By CVE-2022-24668

#	Product Type	Vendor	Product	Version	Update	Edition	Language

Vulnerability trends in Android vs. iOS

CIA

The confidentiality, integrity and availability (CIA) is the core foundation of information security.

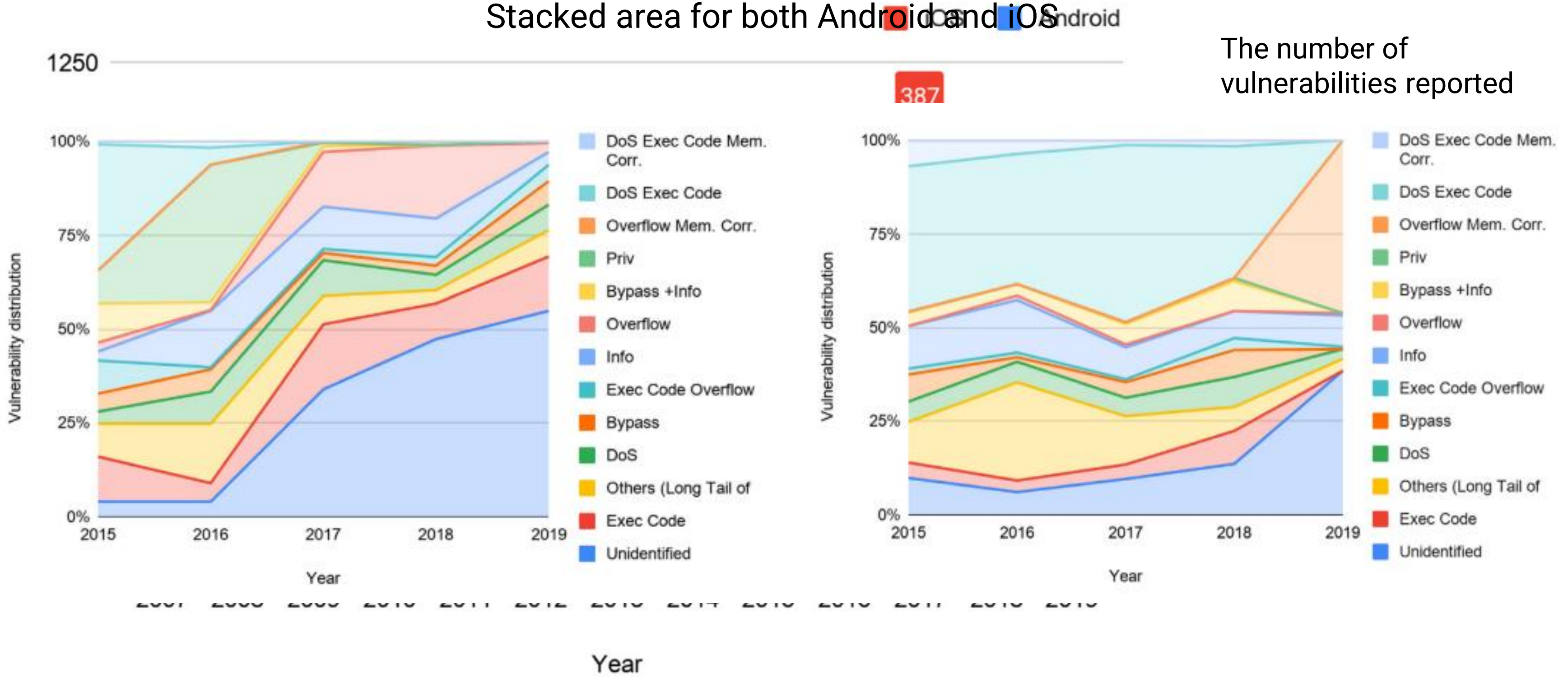
- Confidentiality
- Integrity
- Availability

CIA triad is the backbone of information security system; therefore, it is important to understand the impact of vulnerabilities on CIA.

Vulnerability trends in Android vs. iOS

Stacked area for both Android and iOS

The number of vulnerabilities reported

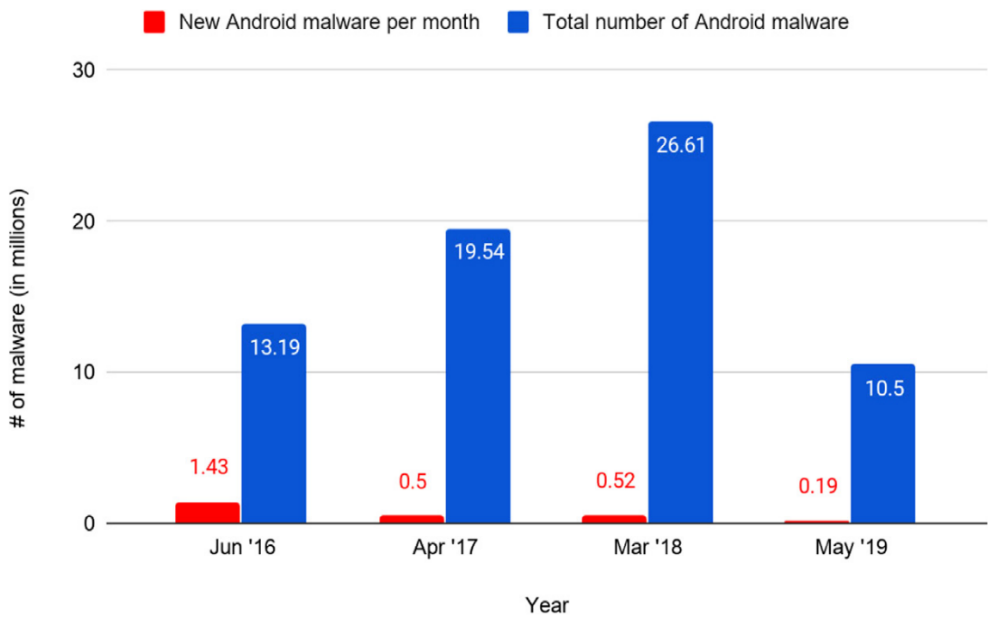


Malware attacks in Android and iOS

types of mobile malware

- **Trojan**
- **Ransomware**
- **Backdoor**
- **Spyware**
- **Adware**

the growth of Android malware worldwide



There is a steep decline in the growth of malware from 2018 to 2019.

WHY?

Research directions and future scope

- Focus on Andriod

why?

Future scope

- Proposed a novel parallel classifier scheme for malware detection in Android , which achieved an accuracy of 98.27%.
- A need of standard, structured and updated comprehensive malware dataset
- A shift from ML to Deep Learning (DL) to handle issues like large data volume and high false positive rate.

Question time

Reference

- https://source.android.com/security
- https://www.apple.com/privacy/docs/Building_a_Trusted_Ecosystem_for_Millions_of_Apps_A_Threat_Analysis_of_Sideloading.pdf
- https://www.diffen.com/difference/Android_vs_iOS
- https://www.geeksforgeeks.org/difference-between-ios-and-android/

Thanks for watching!