Analyzing the Growing Needs of Users to Employ Hardware and Software Restrictions in Smartphones for Increased Privacy and Data Security

Jithu Philip[1] *and Merin Raju*[2]

Abstract

The use of smartphones has become an inevitable part of the lives of common people, as these days almost every day to day activity requires the active use of smartphones for the real world task processing. As the demanding needs of smartphones with the bundled operating system code along with a huge variety of user applications increased, there also exists an evil part hidden within the wide variety of applications that the application developers make use of to gain profit by collecting details about users either as private personal documents or in the form of metadata. As normal users of the system, they may not be always informed or well known about the ways in which these applications work. There also exists a problem where user security is compromised in the form of malicious code. Many government organizations collect and record information about user activity through telecom operators in the name of some backdoor operations they were working without the user's consent. In this paper we try to focus on the perspectives that a common user needs to be aware of, to make their system more private and secure it from data breeches.

Keywords : Countermeasures, Data Breaches, Data Security, Defense, Malware Attacks, User Privacy

I. INTRODUCTION

The usage of mobile operating systems worldwide [1] shows their popularity and increasing acceptance in almost all productive areas. The portable nature of smartphones with the increasing number of user applications made them almost compete with devices like personal computers. Even though the user experience that a smartphone provides is not that similar to professional high end devices, they have proven their existence in this area and have beaten some of the high form factor machines, even with their smaller sized chips.

The ability of ARM based chips to achieve more computational performance with their higher energy efficient nature makes them able to do almost all of the basic operations that a user needs to achieve through the available applications [2], [3], [4], [5], [6]. The increasing growth and usage share also made it the meeting place of everything, that is, the user of a smartphone keeps all of private personal documents, sensitive passwords for banking related stuff etc. within the device itself. This also increases risk in cases where, if the security of the device is compromised in any form, the user's personal documents may be made available to the threat or utilized in some form to make profit.

There exist different types of data security issues that can be shared or leaked by user applications, which fetch user data in its pure form or as metadata. Most of the applications gain this by achieving permissions on the user's device either at install time or at runtime. Most common users who lack the knowledge of the underlying processing behavior of these applications agree and accept most of the permissions requested by the applications. There also exists another category of

Manuscript Received : June 10, 2021; Revised : July 17, 2021 ; Accepted : July 23, 2021.
Date of Publication : October 5, 2021.

[1] J. Philip, *Multimedia Specialist,* Philco Media, Kottayam, Kerala.
Email: jithuphiliph@gmail.com ; ORCID iD : https://orcid.org/0000-0003-4834-5772

[2] M. Raju, *Lecturer (Computer Science),* Department of Commerce, Bishop Kurialacherry College for Women, Amalgiri P.O., Kottayam - 686 561, Kerala. Email : merin.raju12s@gmail.com ; ORCID iD : https://orcid.org/0000-0002-8882-9651

DOI : https://dx.doi.org/10.17010/ijcs/2021/v6/i5/166514

security problem, that an attacker from outside can utilize a flaw in the device's source code, and gain access to the system in the form of a malware [7].

II. EXISTING LAYERS OF SECURITY IN MOBILE OPERATING SYSTEMS

A. Device Security

The typical user of a smartphone initiates its operation by switching the device on. As soon as the device turns on, the boot loader does the booting operation by initiating the operating system loading process. A boot process consists of different verification processes so as to check the integrity of the source code operations. These days attackers try to tamper the system during boot time, through the use of malicious code. Most widely used operating systems like iOS and Android use their own security checks while booting. iOS uses a process called "secure boot chain" [8] for the verification of the running code. Android uses a "verified boot" process to ensure that the device has not undergone any unauthorized modifications. In case a failure is found in these operations, the boot process may be halted and a notification is sent to the user.

Encryption of the device is done to further secure the system. If the device encryption is enabled, user needs to enter a password or Personal Identification Number (PIN) during the boot process [9]. If the device has successfully booted without any security problems, and the password for decryption has been entered, the user can access the User Interface (UI). The UI can be further secured with the help of a screen lock or password lock. Disk encryption can also be employed as it protects data on the device through the ongoing operations. Encryption to a higher level is achieved by high end devices using specific hardware like secure enclave [8], [10] chips in iPhones and Titan-M [11], [12], [13], [14] chips in Google Pixel devices. Trusted Execution Environment (TEE) [7], [15] is another approach where dedicated secure hardware is used in combination with its own operating system software which works apart from the real execution environment for achieving isolation from the real world processing.

B. Application Security

The security of applications is achieved in each operating system through its own way. Application sandbox is one such security mechanism where only the required data and resources are shared with an application. Application sandbox is an access control technology which is mostly enforced at kernel level. Sandboxing of applications is designed in such a way that users have the provision to choose what they share with an application. This allows users with the option that their critical data and access to the system itself is protected in its major share even if an application that is running in the system is compromised and is vulnerable to attack.

The representation of an application working within a sandboxed environment is shown in Fig.1. Here, users

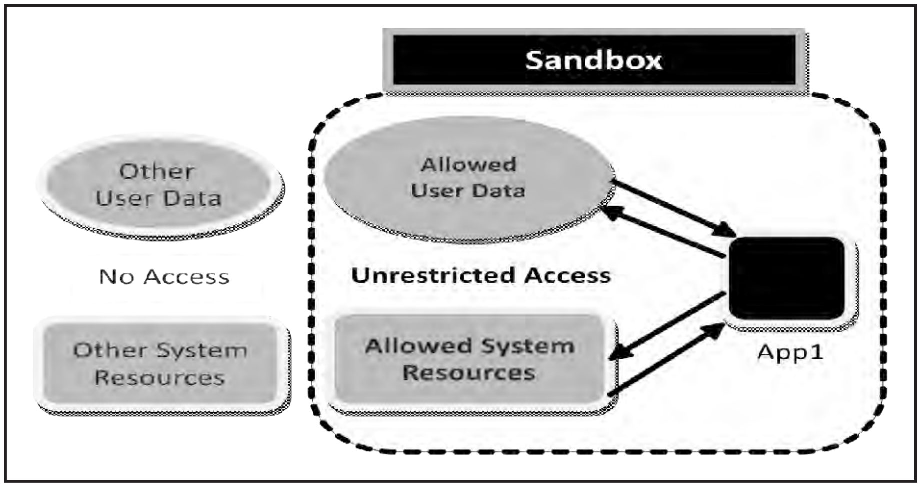


Fig. 1. The Design of an Application Working within a Sandboxed Environment

have the provision to choose what data and resources they share with the application. In this case, the application App1 has only restricted access to its own data and resources. Since all the access permissions are explicitly given by users based on their interactions with the application, data and resources other than what App1 is granted access is isolated from App1.

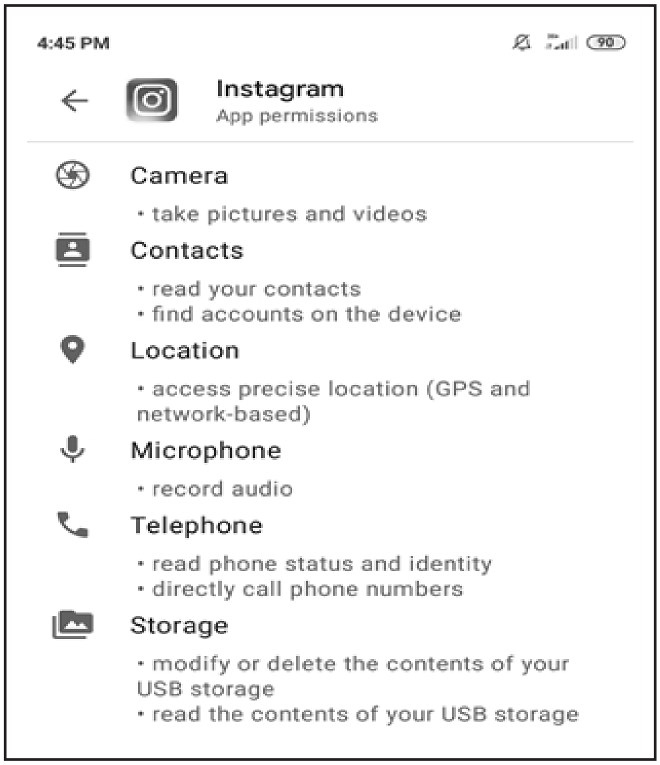
C. Network Security

The huge user base related to different applications and the communications through internet introduced higher needs to make the participating networks secure. There exist cases where data sent between nodes is intercepted by attackers using malicious networks using spoofed network names [9]. To prevent these attacks iOS and Android uses secure communication protocols such as Transport Layer Security (TLS) [9], [8]. Both of them also support Virtual Private Network (VPN) connections. Instant messaging apps became a popular way for users to communicate. Data interception attacks are a concern in these cases too. Most messaging apps these days use end-to-end encryption to prevent attacks of these kinds [9].

III. TYPES OF APPLICATION PERMISSIONS EMPLOYED IN MOBILE OPERATING SYSTEMS

A. Install-Time Permissions

The install time permission setting employs an all-or-nothing approach in which the users are required to accept a permissions list while installing an application. The install time permission dialog of Instagram application while installing from Google Playstore is shown in Fig. 2. In scenarios like this, users are mostly unaware of all the needs related to the permissions requested by the application [16]. This also introduced a way for outside attacks to the system as the default permission granting nature.



4:45 PM 90

Instagram
App permissions

Camera
- take pictures and videos

Contacts
- read your contacts
- find accounts on the device

Location
- access precise location (GPS and network-based)

Microphone
- record audio

Telephone
- read phone status and identity
- directly call phone numbers

Storage
- modify or delete the contents of your USB storage
- read the contents of your USB storage

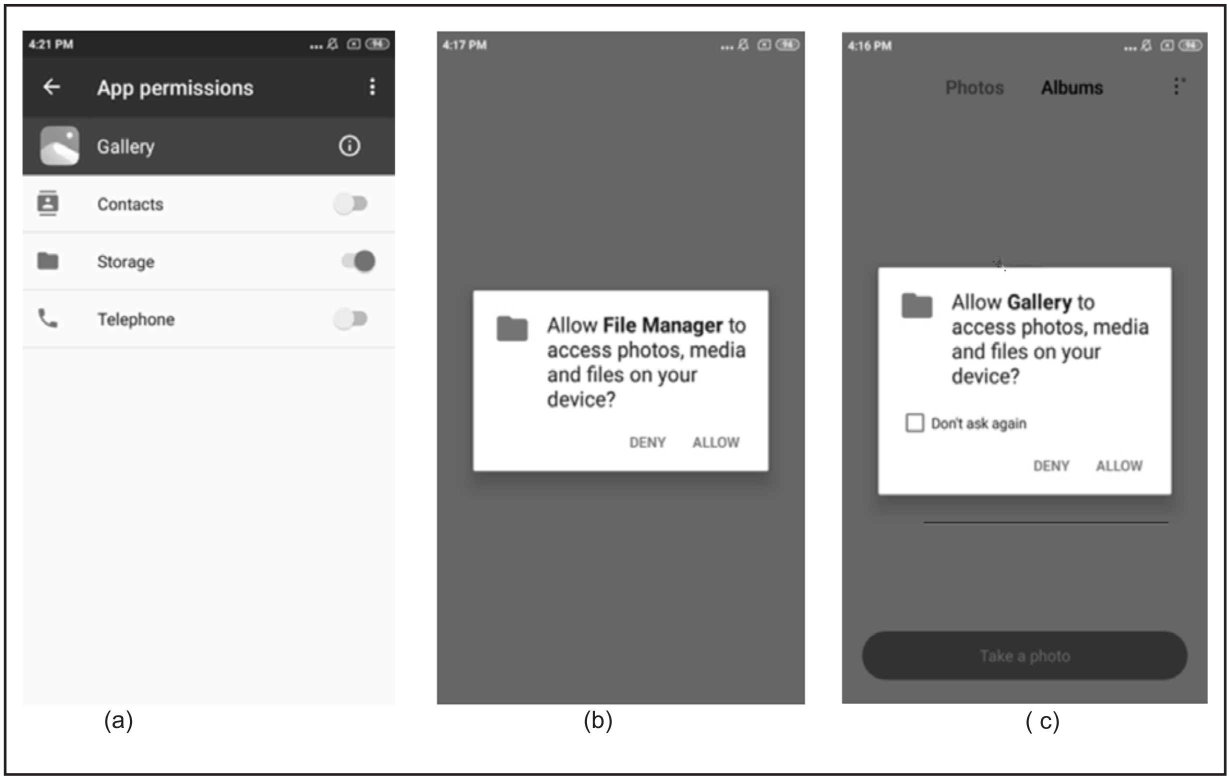
Fig. 2. The Install-Time Permission Dialog of Instagram Application While Installing from Android Playstore

B. Runtime Permissions

The all-or-nothing approach of install-time permissions has been criticized in several works [17], [18], [19]. Runtime permissions provides users with the choice of decision making where permission-wise requests are shown each time whenever required. Runtime permissions appear to be more effective as permission requests are prompted only during the usage time of an application. This helps users to notice and identify whether the requested permission is relevant or not. The general application permission settings of the gallery application is shown in Fig. 3(a).

The runtime permission dialog of file manager application is shown in Fig. 3(b). The runtime permission dialog of file manager application with *don't ask again option* is shown in Fig. 3(c). All of the shown permission settings in Fig. 3. are from the Android operating system.

The selective permission setting introduced in the iOS operating system which allows only the selected photos to be shared with an application is shown in Fig. 4. Here, it shows the settings window where the user can choose and select the photos to be shared. The demonstration in Fig. 4. shows the specified permission setting of the signal application working within the operating system iOS.



4:21 PM

App permissions

Gallery

Contacts

Storage

Telephone

4:17 PM

Allow **File Manager** to access photos, media and files on your device?

DENY ALLOW

4:16 PM

Photos Albums

Allow **Gallery** to access photos, media and files on your device?

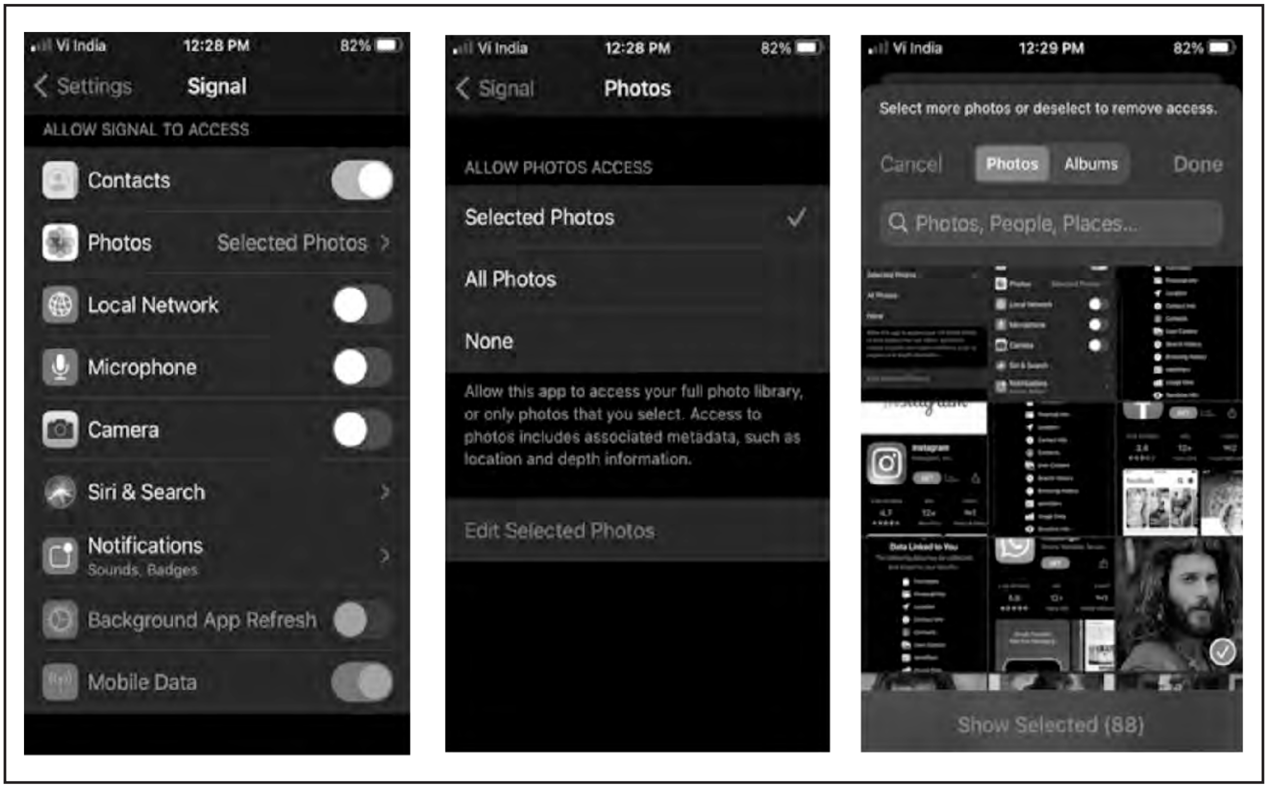
☐ Don't ask again

DENY ALLOW

Take a photo

(a) (b) (c)

Fig. 3. (a) General Application Permission Settings (b) Runtime Permission Dialog of File Manager Application, and (c) Runtime Permission Dialog of File Manager Application with *don't ask again option*



Vi India 12:28 PM 82%

< Settings Signal

ALLOW SIGNAL TO ACCESS

Contacts

Photos Selected Photos >

Local Network

Microphone

Camera

Siri & Search >

Notifications >
Sounds, Badges

Background App Refresh

Mobile Data

Vi India 12:28 PM 82%

< Signal Photos

ALLOW PHOTOS ACCESS

Selected Photos ✓

All Photos

None

Allow this app to access your full photo library, or only photos that you select. Access to photos includes associated metadata, such as location and depth information.

Edit Selected Photos

Vi India 12:29 PM 82%

Select more photos or deselect to remove access.

Cancel Photos Albums Done

Photos, People, Places...

Show Selected (88)

Fig. 4. Selective Permission Setting of Photos to the Signal Application in iOS

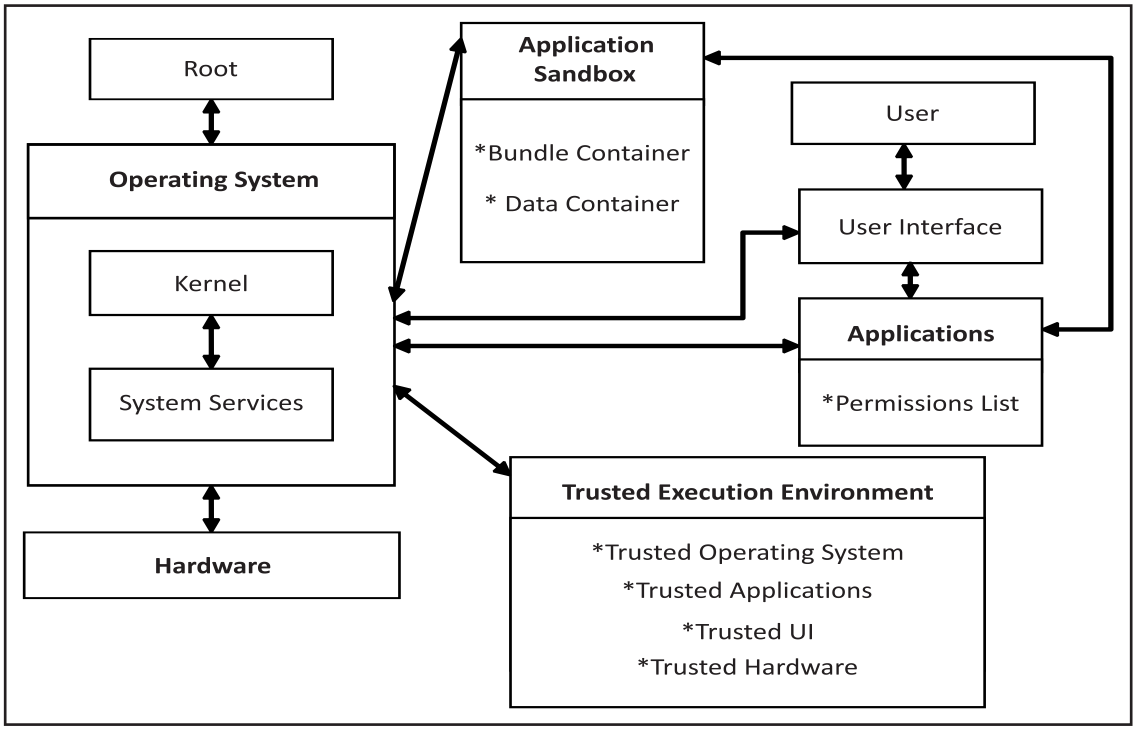


Fig. 5. How User Interacts With Different Operational Layers in a System

IV. EXISTING SECURITY IMPLEMENTATIONS

As described in section II of this document, there exist different layers of security in mobile operating systems for their flawless working. The normal work behavior of a secure system from a user's operational workflow perspective, and how all of the operational layers interconnect is shown in Fig. 5. Here, if the signing and verification stages are completed, the boot loader loads the operating system. Once the user completes the authentication procedures, they can interact with the system through different applications available. The application holds a list of permissions that it needs to get opted for its successful working. The security of the working applications is further achieved by application sandbox whose behavior is different for different operating systems. Like in case of Android it uses a UID based sandbox and in case of iOS it uses a per-app sandbox [20]. The applications which were secured by application sandbox interact with the system hardware through system services.

Hardware encryption or locks exist which ensure the authenticity of working user through different keys [8]. The secure enclave processor chips found on iPhone and the Titan-M chips on Google Pixel devices are specially designed encrypted hardware mechanisms that were made for the purpose of securing user authentication. User authentication processes like biometric authentication, storage and matching process of confidential passwords are also done with the help of these secure hardware. There also exists a successful approach called Trusted Execution Environment (TEE) where dedicated secure hardware is used in combination with its own operating system software which works apart from the real execution environment for achieving isolation from the real world processing of operations.

V. NEED FOR A SECURE AND TRUSTED EXECUTION ENVIRONMENT

The complex nature of hardware with controlling software makes smartphones a target of many attackers. The hardware with the sensors plays a major role while operating the device and through which most of the malicious operators fetch user data as described as followimg :

(1) Device Modem, IMEI, IMSI : Mobile broadband

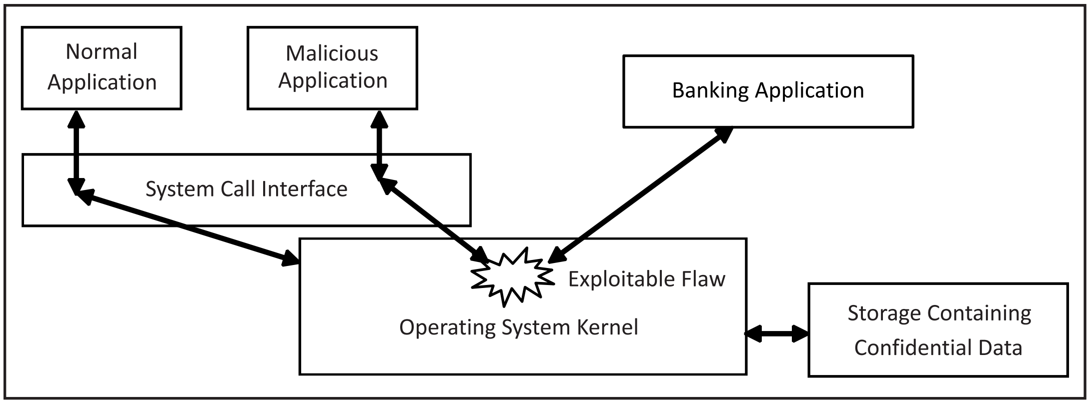


Fig. 6. Attacker Taking Control of the System Through Malicious Application

modem is the hardware which sends and receives the signal for telecommunication. There exist cases where the security agencies under government itself collects unnecessary details about the telecommunication without users' consent [21]. The International Mobile Equipment Identity (IMEI) is an identifier which is unique to a device, and International Mobile Subscriber Identity (IMSI) is a number which uniquely identifies the user of a network. Most applications request permission to access track of these identifiers [22], [23].

(2) Sensors, Camera, Microphones : Different types of sensors are activated inside a smartphone like accelerometer, proximity sensor, gyroscope etc., that most of the applications request and gains access to. The camera and microphone that the user works with can also be used as a medium for spying.

(3) Location Services, GPS : Location services are used by the operating system for functions like finding the lost device location. Location tracking user applications also use these service to guide through different locations with the help of a GPS tracker.

(4) Bluetooth, Wifi : Bluetooth and Wifi are hardware components used for data sharing within short distances. These also can be compromised to attack if the connected network or device is untrusted or malicious [24].

Despite the presence of all of the security mechanisms described in section II of this document, most users still lack knowledge about the internal behavior and working of the system and its applications. Lot of user awareness is required in this area as malware attacks and data breaches are growing day by day [25], [26], [27], [28], [29].

There exist surveys [20] that show the increasing number of malware attacks on mobile operating systems like Android and iOS. As the user base increased,

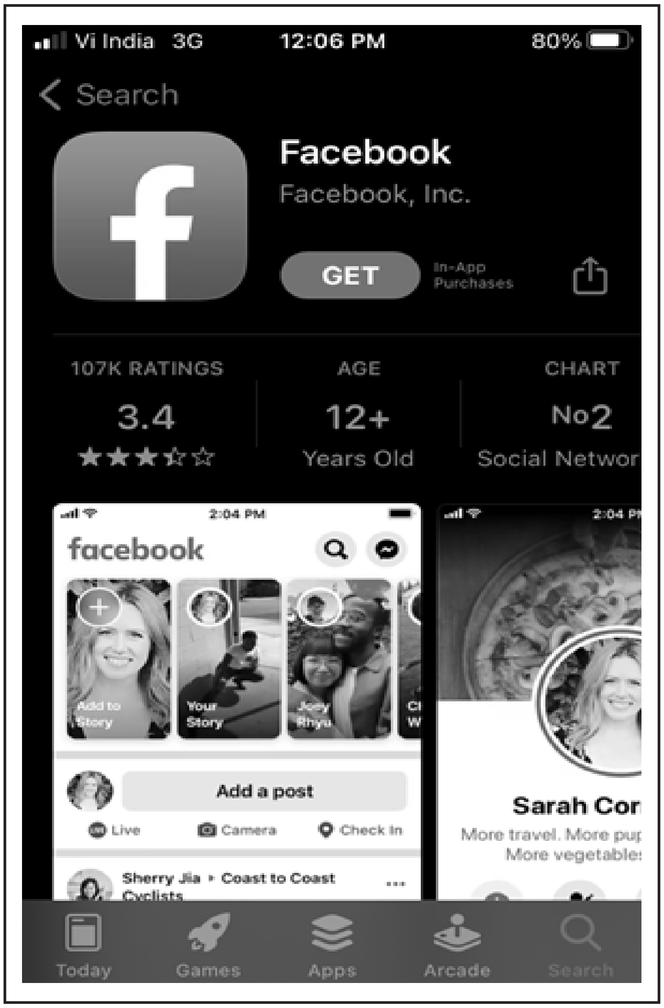
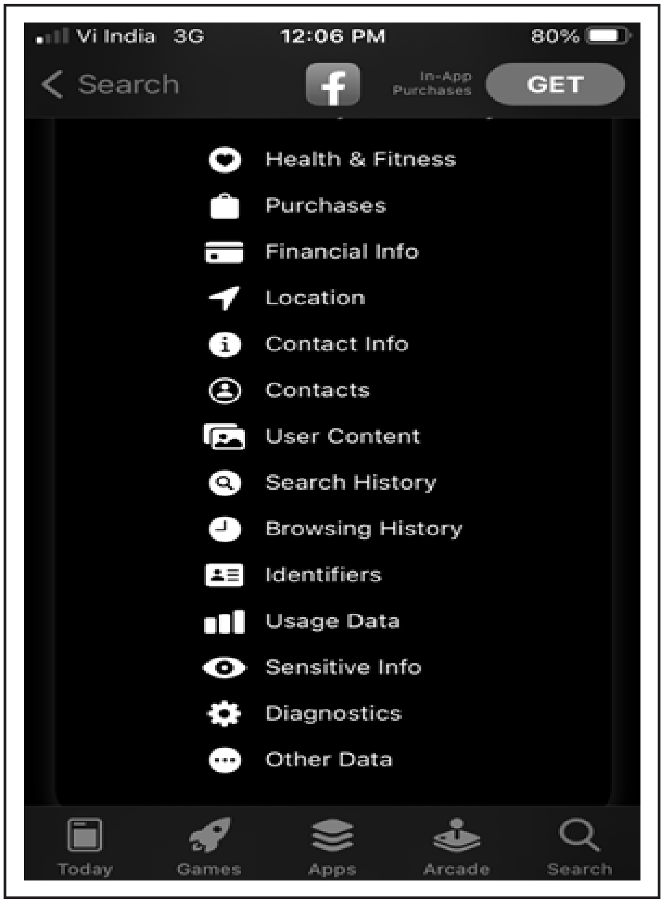


Fig. 7(a). Facebook Application while Installing from iOS App Store



Vi India 3G 12:06 PM 80%

< Search In-App Purchases GET

Health & Fitness

Purchases

Financial Info

Location

Contact Info

Contacts

User Content

Search History

Browsing History

Identifiers

Usage Data

Sensitive Info

Diagnostics

Other Data

Fig. 7(b). The Data that the Facebook App Collects from a User as a Link for Identity

attacks happening against the system also increased. The demonstration of an attacker taking control of the system through a malicious application is shown in Fig. 6. If the system is not secured while exploring the details, there is a possibility that an intruder taking control of an application can make use of flaws in the source code to take advantage of it, thereby compromising the security of the entire system.

There also exists a scenario in which the user application collects user data directly or as metadata. Applications make use of this information to gain profit by sharing these personal details with third parties. Third parties involved may be advertisers, e-commerce platforms etc.

The screenshot of Facebook application while installing from iOS App store is shown in Fig. 7(a). The data that the Facebook app collects from a user as a link for identity is shown in Fig. 7(b).

During the analysis stage we collected details about the permissions that the most widely used applications request. The data that these applications can access from a user's device is huge and can affect the security of the concerned devices. The details of the data that the major applications collects from users and share with third parties are given in Table I.

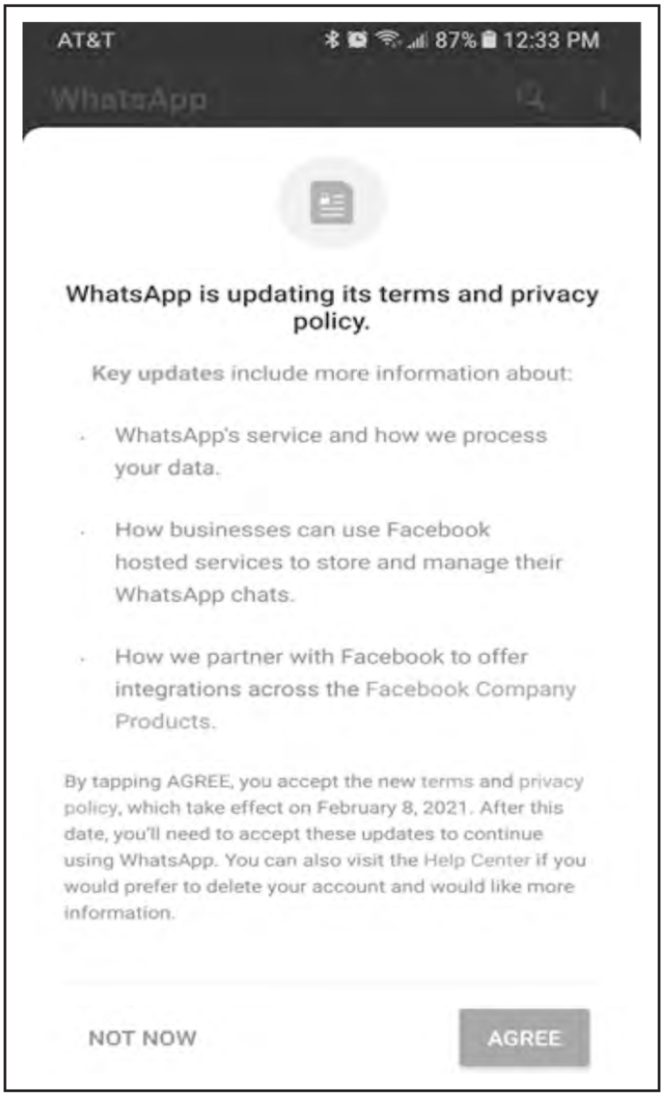
TABLE I.

USER DATA THAT THE APPLICATIONS COLLECTS AND SHARES WITH THIRD PARTIES

No	Applications	Purchase	Location	Contact Info	Contacts	User Content	Search History	Browsing History	Identifiers	Usage Data	Diagnostics	Sensitive Info	Financial Info	Other Data
1	Instagram	√	√	√	√	√	√	√	√	√	√		√	
2	Facebook	√	√	√	√	√					√		√	√
3	Whatsapp	√	√	√	√	√			√	√	√		√	
4	LinkedIn	√	√	√		√	√		√	√				
5	Uber Eats	√	√	√			√		√	√				
6	Youtube		√	√			√	√	√	√				
7	eBay	√					√	√	√	√				
8	TikTok		√	√				√	√	√				
9	Reddit		√			√			√	√				
10	Snapchat		√	√					√	√				
11	Spotify		√	√					√	√				
12	Amazon Prime Video								√	√				√
13	Twitter	√							√	√				
14	Walmart								√	√				

VI. RESTRICTIONS ON USER CUSTOMIZABLE APPLICATION PERMISSIONS AND PRIVACY POLICIES

As described in the previous sections there exists a lack of user friendliness for customizing the enabling and disabling of application permissions. Application developers are updating their privacy policies to adapt to the needs of data collection so as to share these data with third parties. A recently updated privacy policy agreement window of the messaging application Whatsapp, which introduced many controversies because of sharing user's data outside of the application is shown in Fig. 8.



WhatsApp is updating its terms and privacy policy.

Key updates include more information about:

- WhatsApp's service and how we process your data.
- How businesses can use Facebook hosted services to store and manage their WhatsApp chats.
- How we partner with Facebook to offer integrations across the Facebook Company Products.

By tapping AGREE, you accept the new terms and privacy policy, which take effect on February 8, 2021. After this date, you'll need to accept these updates to continue using WhatsApp. You can also visit the Help Center if you would prefer to delete your account and would like more information.

Fig. 8. The Recently Updated Privacy Policy Agreement Window of the Messaging Application Whatsapp

VII. CONCLUSION

During the analysis stage of this work we installed different applications on both the platforms, Android and iOS. The user permissions list requested by the applications at each stage were collected and studied. Based on the studies it is found out that even though there exist some security locks that the operating system applies on individual applications, it may not be sufficient in all cases. The end user of the system who is dealing with all of these applications needs to be aware of the fact that a fake or malicious application which has gained access to the system can cause harm to the entire workflow of the system. The non-transparent nature of newly introduced privacy policies also made a way for the increasing number of data breaches on the system. The runtime permissions list that an application is requesting access to is to be verified and re-checked by the user to increase data security in the system. Anyhow, it needs to be said that the increase in changing application permissions and privacy policies will be more and more over the years as personal user data collection in the form of metadata plays a huge role in advertising and making profit. So every end user needs to be aware of the fact that securing and monitoring the system from a hardware and software perspective is required for increased privacy and data security.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable suggestions regarding the details of this document.

AUTHORS' CONTRIBUTION

The authors participated and processed different phases during the making of this document. Merin Raju collected details about the behaviour of different applications, security flaws reported on different platforms, restrictions on user customizable policies etc. Jithu Philip conducted analysis of the existing security implementations regarding devices and their applications. He also analyzed the structure of the current application permissions employed in mobile applications, and also studied the level of security impact that a user customizable system can achieve in comparison with the existing working scenario. Jithu Philip and Merin Raju actively participated in the writing of this manuscript.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non-financial interest in the subject matter, or materials discussed in the manuscript.

FUNDING ACKNOWLEDGMENT

The authors received no financial support during the research, processing, and publication of the article.

REFERENCES

[1] *Mobile Operating System Market Share Worldwide*. Statcounter. April 2021. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[2] R. Triggs, “Arm vs x86: Instruction sets, architecture, and all key differences explained,” *Android Authority,* June 8, 2021. [Online]. Available: https://www.androidauthority.com/arm-vs-x86-key-differences-explained-568718/

[3] E. Engheim, “Why is Apple’s M1 chip so fast?,” *Debugger,* November 28, *2020.* [Online]. Available: https://debugger.medium.com/why-is-apples-m1-chip-so-fast-3262b158cba2

[4] J. Turley, “Apple M1 vs. Intel Core i7: The benchmark wars continue,” *Electron. Eng. J.,* February 15, 2021. [Online].Available: https://www.eejournal.com/article/apple-m1-vs-intel-core-i7-the-benchmark-wars-continue/

[5] P. Tracy, “Apple M1 vs. Intel CPU: This is the best processor for your laptop,” *Laptop.* [Online]. Available: https://www.laptopmag.com/amp/news/apple-m1-vs-intel-cpu-this-is-the-best-processor-for-your-laptop

[6] J. Hruska,“Current x86 vs. Apple M1 performance measurements are flawed,” *ExtremeTech,* December 7, 2020. [Online]. Available: https://www.extremetech.com/computing/318020-flaw-current-measurements-x86-versus-apple-m1-performance

[7] J. Philip and M. Raju, "Security impact of trusted execution environment in rich execution environment based syst," *Indian J. of Comput. Sci.,* vol. *5,* no. 4–5, pp. 26–37, 2020. [Online]. Available: http://dx.doi.org/10.17010/ijcs%2F2020%2Fv5%2Fi4-5%2F154785

[8] J. Philip and M. Raju, “An overview about the security architecture of the mobile operating system iOS,” *Indian J. of Comput. Sci.,* vol. 4*,* no. 1, pp. 13–18, 2019. [Online]. Available: http://dx.doi.org/10.17010/ijcs%2F2019%2Fv4%2Fi1%2F142412

[9] User Experience with Mobile Security and Privacy Mechanisms. July 4, 2017. [Online]. Available: https://www.qu.tu-berlin.de/fileadmin/fg41/kraus_lydia.pdf

[10] T. Mandt, M. Solnik, and D. Wang, “Demystifying the secure enclave processor,” *OffCell Res. and Azimuth Security*. [Online]. Available: https://www.blackhat.com/docs/us-16/materials/us-16-Mandt-Demystifying-The-Secure-Enclave-Processor.pdf

[11] B. Barrett, “The tiny chip that powers up pixel 3 security,” *Wired,* [Online]. Available: https://www.wired.com/story/google-titan-m-security-chip-pixel-3/

[12] R. Triggs, “Will Google’s Titan M make it harder for the ROMing scene?,” 2018. [Online]. Available: https://www.androidauthority.com/titan-m-security-chip-915888/

[13] C. Hoffman, “Your smartphone has a special security chip. here’s how it works,” 2018. [Online]. Available: https://www.howtogeek.com/387934/your-smartphone-has-a-special-security-chip.-heres-how-it-works/

[14] X. Xin, “Titan M makes Pixel 3 our most secure phone yet,” 2018. [Online]. Available: https://www.blog.google/products/pixel/titan-m-makes-pixel-3-our-most-secure-phone-yet/

[15] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *14th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Commun.,* August 2015, Helsinki, Finland. [Online]. Available: https://doi.org/10.1109/Trustcom.2015.357

[16] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D.Wagner. "Android permissions: User attention, comprehension, and behavior," in *Proc. of the Eighth Symp. on Usable Privacy and Security (SOUPS).* ACM. 2012, p. 3. [Online]. Available: https://cups.cs.cmu.edu/soups/2012/proceedings/a3_Felt.pdf

[17] S. Garfinkel and H. R. Lipford. "Usable security: History, themes, and challenges," in *Synthesis Lectures on Inform. Security, Privacy, and Trust,* 2014, pp. 1–124. Morgan & Claypool. [Online]. Available: https://doi.org/10.2200/S00594ED1V01Y201408SPT011

[18] J. Lin, B. Liu, N. Sadeh, and J. I. Hong. "Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings," in *Proc. of the Tenth Symp. On Usable Privacy and Security (SOUPS).* 2014, pp. 199–212. [Online]. Available: https://www.usenix.org/system/files/conference/soups2014/soups14-paper-lin.pdf

[19] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D.Wagner, and K. Beznosov, "Android permissions remystified: A field study on contextual integrity," in *USENIX Security Symp.* 2015, pp. 499–514.

[20] J. Philip and M. Raju, "A formal overview of application sandbox in Android and iOS with the need to secure sandbox against increasing number of malware attacks", *Indian J. of Comput. Sci.,* vol. *4,* no. 3, pp. 32–40, 2019. [Online].Available: http://dx.doi.org/10.17010/ijcs%2F2019%2Fv4%2Fi3%2F146164

[21] BBC News, "Edward Snowden: Leaks that exposed US spy programme," January 17, 2014. [Online]. Available: https://www.bbc.com/news/world-us-canada-23123964

[22] A. Alshehri, A. Hewins, M. McCulley, H. Alshahrani, H. Fu, and Y. Zhu, "Risks behind device information permissions in Android OS," *Commun. and Network,* vol. *9,* no. 4, pp. 219–234, 2017. [Online]. Available: https://doi.org/10.4236/cn.2017.94016

[23] S. Achleitner and C. Xu, "Android apps leaking sensitive data found on Google play with 6 million U.S. downloads," November 24, 2020. [Online]. Available: https://unit42.paloaltonetworks.com/android-apps-data-leakage/

[24] "Wireless connections and bluetooth security tips, "[Online]. Available: https://www.fcc.gov/consumers/guides/how-protect-yourself-online

[25] R. Sobers, "98 Must-Know Data Breach Statistics for 2021," *Varonis,* April 16, 2021. [Online]. Available: https://www.varonis.com/blog/data-breach-statistics/

[26] L. Constantin, "One in three organizations suffered data breaches due to mobile devices," CSP, March 6, 2019. [Online]. Available: https://www.csoonline.com/article/3353560/one-in-three-organizations-suffered-data-breaches-due-to-mobile-devices.html

[27] "Smartphones face high hacking risk in 2020: Report," *The Hindu,* January 1, 2020. [Online]. Available:https://www.thehindu.com/sci-tech/technology/gadgets/smartphones-face-high-hacking-risk-in-2020-report/article30450912.ece

[28] "Remote working linked to data breach in 66% Indian firms: Survey," CISO. in, August 20, 2020. [Online]. Available: https://ciso.economictimes.indiatimes.com/news/remote-working-linked-to-data-breach-in-66-indian-firms-survey/77653551

[29] The Guardian, "Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach," [Online]. Available: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

About the Authors

Jithu Philip received M.Sc. degree in Computer Science from School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala, India in 2014. He is currently working as Multimedia Specialist with Philco Media, Kottayam, Kerala, India. His research interests are in the areas of Operating Systems and Computer Security.

Merin Raju received M.Sc. degree in Computer Science from School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala, India in 2014. She is currently working as Lecturer with Computer Science, Department of Commerce, Bishop Kurialacherry College for Women, Amalagiri, Kottayam, Kerala, India. Her research interests are focused on Computer Security.