International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Review on Android Application Security

P. Vignesh Pejathaya[1], Imran Khan[2], Vignesh Shetty[3], Sayeesh [4], Manish B. Shriyan [5]

[1]*Dept. of CSE, Alva's Institute of Engineering & Technology, Mijar, Moodbidri, India*
[2]*Dept. of CSE, Alva's Institute of Engineering & Technology, Mijar, Moodbidri, India*
[3]*Dept. of CSE, Alva's Institute of Engineering & Technology, Mijar, Moodbidri, India*
[4]*Dept. of CSE, Alva's Institute of Engineering & Technology, Mijar, Moodbidri, India*
[5]*Dept. of CSE, Alva's Institute of Engineering & Technology, Mijar, Moodbidri, India*

Abstract - *Smartphone's are used by billions of people all over the world which indicates the applications of the Smartphone is increasing and the main concern of any applications is the marketplaces to completely validate if an application is malicious or legitimate. Recent advances in hardware and telecommunications have enabled the development of low cost mobile devices equipped with a variety of sensors. It makes an easy target for the malware developers and other computer criminals. The anti-malware organizations and academic researchers have produced and proposed many security methods and mechanisms in order to recognize and classify the security threat of the Android operating system. This paper tells about the misuse of app permissions using Shared User ID, how two factor authentications fail due to inappropriate and improper usage of app permissions using spyware, data theft in Android applications, security breaches or attacks in Android and analysis of Android, and Windows operating system regarding its security.*

1. WORKING OF MOBILE APPLICATIONS

Mobile applications are at the epicenter of current development trends. Most of these applications have a client–server architecture. The client runs on the operating system, which is most frequently Android. This client is downloaded to the device from the app distribution platforms, where developers publish their wares. As perceived from the user's point of view, the client installed on the Smartphone is the mobile application. This is what the user interacts with to make purchases, pay bills, or read emails. But in fact, there is also another component: the server, which is hosted by the developer. Often this role is performed by the same software that is responsible for generating and processing content on the site. In other words, most often the server-side component is a web application that interacts with the mobile client over the Internet by means of a special application programming interface (API). Which gives us two types of vulnerabilities

1.1 Client-side vulnerability :

Android applications tend to contain critical vulnerabilities slightly more often than those written for iOS (43% vs. 38%). But this difference is not significant, and the overall security level of mobile application clients for Android and iOS is roughly the same. About a third of all vulnerabilities on the client side for both platforms are high-risk ones

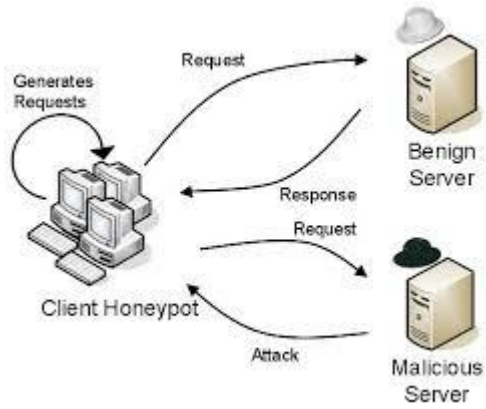

Fig -1

1.2 Server-side vulnerability :

Server-side components contain vulnerabilities both in application code and in the app protection mechanisms. The latter include flaws in the implementation of two-factor authentication. Let us consider one vulnerability our experts encountered in an application. If two identical requests are sent to the server one right after the other, with a minimal interval between them, one-time passwords are sent to the user's device both as push notifications and via SMS to the linked phone number. The attacker can intercept SMS messages and impersonate the legitimate user, for instance, by cleaning out the user's bank account.

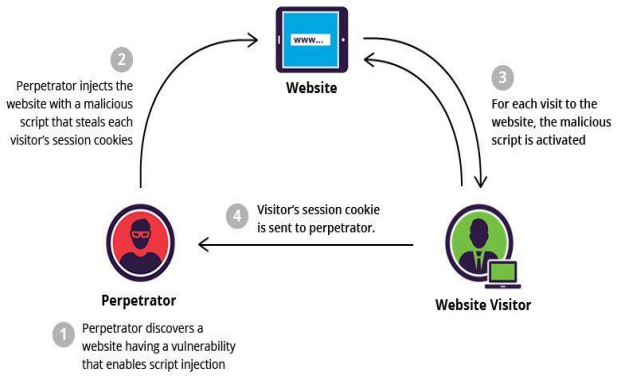

Fig -2

2. Security Analysis of Mobile Device-to-Device Network Applications

Smartphone have become the major part of human's life. Nowadays mobile applications are playing major role in many areas such as banking, social networking, financial apps, and entertainment and so on. For every desktop or web application an alternate mobile app is available. With just single click number of mobile apps is available from Google's play market. With this huge number of applications securities is an important issue. Many research articles discuss about the security of the applications and the malicious apps that may affect or leak sensitive data such as International Mobile equipment Identity Number (IMEI) of device, credit or debit card information, location information and so on. As the android market is growing, security risk has increased and thus focus should be given to the security

3. MOBILE DEVICE-TO-DEVICES (D2D):

Smartphone has now become a standardized feature in many mobile devices, by which mobile devices can communicate with each other even when commercial Internet access is not available. Because D2D network is expected to be an intrinsic part of the Internet of Things (IoT) and mobile device is the smartest and the most advanced commercial device in everyday usage, the D2D feature and related security protocols it adopts influences the design and implementation of many other IoT devices. While D2D network provides tangible benefits to users, it also raises the security risks of information leaking.

While D2D network provides tangible benefits to users, it also raises the security risks of information leaking. This paper presents an in-depth empirical security analysis on mobile D2D network among Android devices. Android apps could establish a mobile D2D network in various ways, including Wi-Fi hotspot, Wi-Fi Direct and Bluetooth. Those mobile D2D protocols normally take different protection mechanisms, which makes security investigation considerably challenging. So here main focus is on most popular apps in the Google Play Store, with aggregated downloads more than 500 million. Our analysis reveals some critical vulnerability. The key findings are bi-fold. First, the current mobile D2D network framework enabled by Android has significant flaw of over privilege issue. Second, identify that most data transfer over mobile D2D network is unencrypted. Furthermore, exploit the identified Android framework flaws to construct three proof-of-concept attacks and conclude with security lessons and suggestions of possible solutions against the identified security issues.

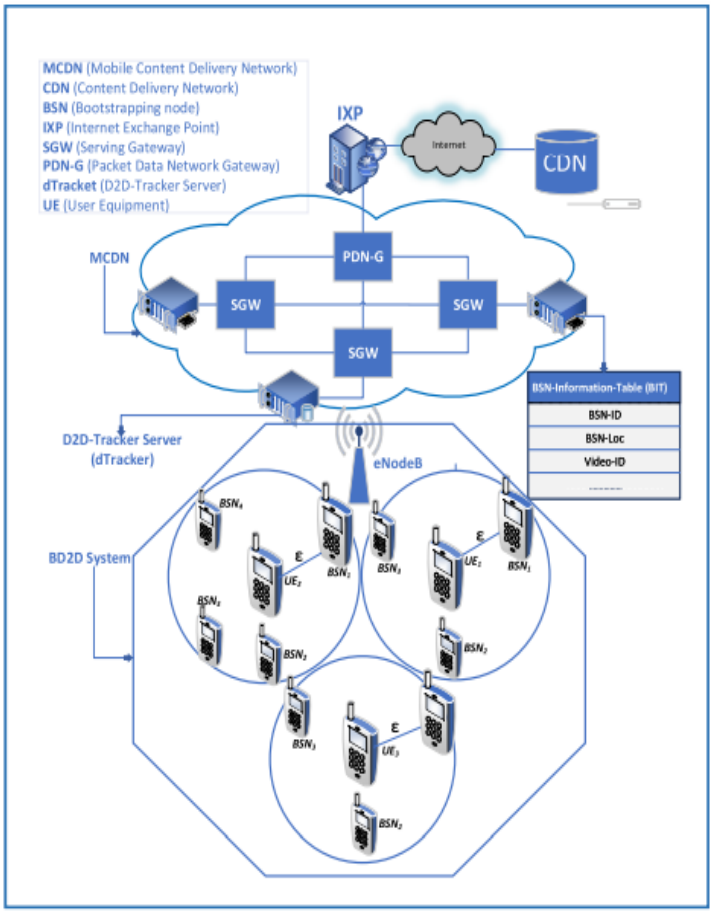

Fig -3

4. MOBILE D2D NETWORK ARCHITECTURE AND THREAT MODEL

- **Wi-Fi Hotspot Wi-Fi hotspot**

Also known as Wi-Fi Tethering, has been gaining popularity as a convenient, on-the-move, and cost-effective wireless Internet access technology. A Wi-Fi hotspot network is a star-like cluster based network with a central node providing 3G/4G data tethering and allowing plural surrounding nodes to join and to gain internet access. Henceforth refer to the central node and the surrounding node as the cluster head (CH) and the cluster member (CM).

- **WPA2 PSK is a system API:**

To create a WPA2 PSK secured hotspot network programmatically, the developer must set the Wi-Fi configuration parameter to support WPA2 PSK key management. However, the WPA2 PSK is a system API as well; developers cannot access this security setting directly.

- **Hotspot without password absolutely unprotected:**

The use of hotspot imposes the risk of people capturing Realtime traffic over the wireless connections. Attackers can easily capture, from the air, the packets of unsecured connections to hotspots. It is intuitive that open hotspot has

no security protection but many hundred million download apps still choose to use open hotspot, seeking to maximize the convenience. Those apps leverage the user experience over security but expose the information of the smart phone to the public.

- **Sensitive network information leakage**

As discussed in IV-A, the over privileged issue due to coarse-grained framework opens a back door to a third-party app to access key information about the network. Because the framework does not place any restrictions on outbound Internet communication either, such a design flaw allows malicious apps to abuse this ability to leak sensitive information from a victim's smart phone. However, information leakage via the Internet is not what we concentrate on for two reasons. First, in a D2D network scenario, Internet is assumed unavailable; second, D2D data is transmitted locally without the Internet access. We exploit an offline attack model based on the D2D network architecture described in III. Following attack logic, a third-party device can receive key information about the D2D network in offline mode.

- **Avoidable human-involved authentication process**

The human-involved two-way authentication process during mobile D2D network establishment. Human involvement is one of the best solutions to eliminate security risk in D2D network because a human can physically identify the appropriate device and approve the connection request. However, due to coarse-grained framework design, hackers can bypass the two-way authentication process. From IV-B, attackers can offline retrieve the password of the network. Now using the password, any device can join the network without undertaking the two-way authentication.

- **Insecure data transfer**

Developers have great freedom to choose how to transmit data on the D2D network. Android neither specifies nor limits the protocol to make data transfer on the application level. Because file transfer is the most common application using the mobile D2D network, Here the focus is on security analysis on the protocol choices related to file transfers.

- **Shareit**

Claims to be the number 1 file transfer tool in the world. It has over 500 million users in China, 300 million users in India and over 1 billion users in world wide. Despite being such a popular application, its security implementation is coarse-grained and we found multiple security issues. Before we can make an analysis on the network, we must identify which protocol it selects as the network basis. The default D2D network established by Shareit is via hotspot. Wi-Fi Direct feature is included in the setting and user can enable it. However, Shareit marks Wi-Fi Direct as a Beta feature.

Some key Algorithum used:

AES (Advanced Encryption Standards) Algorithm:

The basic AES-Algorithm works in four steps as follows: Sub-byte Transformation: Is a nonlinear byte Substitution, using a substation table (s-box), which is constructed by multiplicative inverse and Affine Transformation. Shift rows transformation: Is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from one to three bytes. Mix columns transformation: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers. Add round key transformation: Is a simple XOR between the working state and the round key. This transformation is its own inverse.

Step 1: Registration and Lock Screen: Upon the installation of the application the user has to register himself to the application. During registration, the user sets up e-mail and password. The same password will be used for further access to the application. On successful registration a mail is sent to the registered e-mail id containing registration details.

Step 2: Encrypt: If the user wishes to encrypt the data stored in the device he/she can manually select the files which needs to be encrypted and click on the encrypt button and the file will be encrypted. AES-256 bit algorithm is used for encryption of the data. Once the user encrypts the file, the original file gets deleted from that location and the encrypted file is then stored in a new folder. In case of brute force attack by an intruder, the application will automatically trigger the encryption algorithm and all the files marked by the user as confidential, will be encrypted. There will be limit of entering the password incorrectly and once the intruder crosses the limits the data will be automatically encrypted.

Step 3: Settings: Here, the user is provided with three functionalities: Change Password, Select Confidential Files and List Confidential Files. Change Password: The user can change his/her password. After which he will receive a mail on the registered email-id with new details. Add Confidential Files: User can select all the confidential files which will be kept hidden. View Confidential Files: The user will be provided with the list of all the confidential files. Here the user can un-hide them.

Step 4: Cloud Backup: Advantages of cloud backup is to ensure the safety, longevity, and high accessibility of the user data. Cloud backup is provided to the user in case any files are lost due to automatic encryption. There are many places you can back up your data files to. A Cloud backup is where a remote, on-line, or managed service provides users with a system for backing up, storing, and recovering data files.

➢ Encrypt

If the user wishes to encrypt the data stored in the device he/she can manually select the files which needs to be encrypted and click on the encrypt button and the file will be encrypted (Fig 4).

Fig -4

On encryption the image thumbnail will be visible as shown in Fig 8 and the encrypted files are moved to a new location created by the application shown in Fig 5.

Fig -5

In case of brute force attack by an intruder, the application will automatically trigger the encryption algorithm and all the files marked by the user as confidential, will be encrypted (Fig 6).

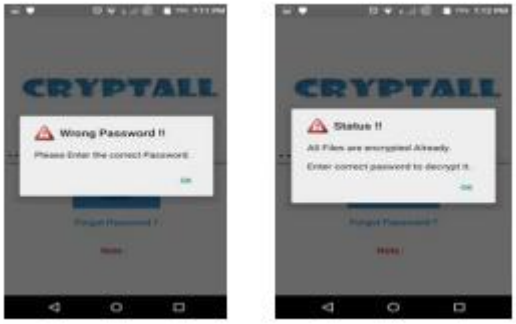

Fig -6

There will be limit of entering the password incorrectly and once the intruder crosses the limits the data will be automatically encrypted.

➢ Decrypt

The user will be provided with the list of all the encrypted files. The user can select any of the encrypted files and decrypt it to recover the data. After decryption the data will be stored in the Decrypted Folder (Fig 7)

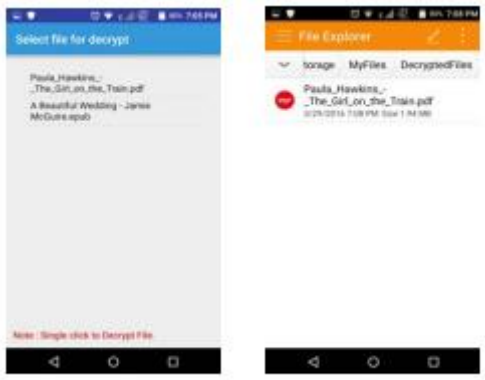

Fig -7

5. CONCLUSION

By performing an empirical security evaluation of the mobile D2D framework on Android operating system. Analyzing mobile D2D network is challenging because there were multiple mechanisms to establish a D2D network on Android and each of them uses distinct security protections. Additionally, the apps installed on Android were close-sourced individual modules so we did not know how file transfer service was implemented on the application level. Mobile applications and related security breaches receive a lot of media attention. New threats are being discovered every day so making an application to keep your data secure from such threats is important. You cannot be 100% safe, but you can make it hard. The proposed approach aims to provide user satisfaction of storing personal and sensitive image data including images and files securely in their mobile phones.

REFERENCES

[1] Fei Shao, Zinan Chang, Yi Zhang, "AES Encryption Algorithm Based on the High Performance Computing of GPU," Second International Conference on Communication Software and Networks, 2010.[Date of access: 20 August 2015].

[2] Abhishek Vichare and Tania Jose, Jagruti Tiwari, Uma Yadav , "Data Security using Authenticated Encryption and Decryption Algorithm for Android Phones", International Conference on Computing, Communication and Automation (ICCCA2017)

[3] Nasreen anjum and Zhaohui yang "Device-to-Device (D2D) Communication as a Bootstrapping System in a Wireless Cellular Network", International Journal of Computer Applications, n January 7, 2019, date of current version January 23, 2019

[4] SHAO Guo-hon1, "Application Development Research Based on Android Platform", 2014 7th International Conference on Intelligent Computation Technology and Automation

[5] Karthick S, "Android Security Issues and Solutions", International Conference on Innovative Mechanisms for Industry Applications
(ICIMIA 2017)

[6] Alireza Sadeghi, Hamid Bagheri and Sam Malek, "Analysis of Android Inter-App Security Vulnerabilities Using COVERT", 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering

[7] Peter Teufl, Andreas Fitzek, Daniel Hein, Alexander Marsalek, Alexander Oprisnik, Thomas Zefferer , "Android Encryption Systems", 2018 5th International Conference on Intelligent Computation Technology and Automation

[8] Ming-Yang Su, Sheng-Sheng Chen, Tsung-Ren Wu, Hao-Sen Chang, You-Liang Liu, "Permission Abusing by Ad Libraries of Smartphone Apps", ©2019 IEEE International Conference on Intelligent Computation Technology and Automation

[9] Positive technologies, Vulnerabilities and threats in mobile applications, 2019.