SECTRIO

MALWARE REPORT

Android File-Infector

Date: 27/04/2021

Sanjuktasree Chatterjee

Cybercriminals are now actively targeting the Android platform and their applications to conduct attacks with various malware families.

On 26th March 2021, we found a zero-day Android package file in our honeypot which is highly malicious. This file is a variant of a Chinese social networking application 'qq', which is found to be a malicious file infector that causes root-level access to the user's device and downloads multiple malicious files to infect other applications.

File Hash: 832c8563fc33fffcbcfb7411875f698b

Technical Analysis:

To analyse the apk sample, first, it needs to be decompiled. In Fig:1, The source code reveals that the file contains downloading URL which itself is malicious.

```
-->destFilePath:
-->finish by timeout
-->generateDownloadUrl, url: http://qzs.qq.com/open/mobile/login/qzsjump.html?
-->login activity:
-->login channelId is null
-->login channelId:
-->login get channel id exception.
```

Fig: 1

The file is a variant of an application 'qq' which is a Chinese social-networking application. The package name shows that the file is 'android.qqdownloader' (Fig:2) which is different from the main package name. It shows that the attacker uses the source code of 'qq' to lure the victim.

```
package="com.tencent.android.qqdownloader"
```

Fig: 2

In Fig:3, it is showing the file is doing 'RootUninstall' that can cause OS damage and data loss. The file itself changes 'SuperUser' permission to access the root-level.

```
<activity android:theme="@style/bb" android:name="com.tencent.pangu.link.LinkImpl
<activity android:name="com.tencent.assistant.activity.InstalledAppManagerActivit
<activity android:name="com.tencent.assistant.activity.RootUtilInstallActivity" a
<activity android:name="com.tencent.pangu.update.UpdateListActivity" android:laun
<activity android:name="com.tencent.pangu.update.photonui.UpdatePhotonListActivit
<activity android:name="com.tencent.pangu.update.UpdateIgnoreListActivity" androi
<activity android:theme="@style/d0" android:name="com.tencent.assistant.activity.
```

Fig: 3

It is a kind of malware that downloads malicious adware to the device so it can infect another valuable file and device memory. There are multiple URLs embedded in the source code of the file, out of thesesome URLs are used to update the file.

```
http://106.38.181.148/dd.myapp.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lq
http://android.bugly.qq.com/rqd/async
http://android.myapp.com/myapp/category.htm?orgame=1
http://android.rqd.qq.com/analytics/async
http://appimg1.3g.qq.com/android/50801/16629897/icon_72.png
http://appsupport.qq.com/cgi-bin/qzapps/mapp_addapp.cgi
http://bjuser.jpush.cn/v1/appawake/status
http://dd.myapp.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lqs.xy_1.0.400_10
http://dd.myapp.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lqs.xy_1.0.400_10
http://fusion.qq.com/cgi-bin/qzapps/unified_jump?appid=%1$s&from=%2$s&isOpenAppID=1
http://maweb.3g.qq.com/welcome.html
http://monitor.uu.qq.com/analytics/rqdsync
http://openmobile.qq.com/oauth2.0/m_jump_by_version?
http://pin.qq.com/yyb/index.html
http://pingma.qq.com:80
http://pingma.qq.com:80/mstat/report/
http://pingmid.qq.com:80/
http://pp.myapp.com/ma_icon/0/icon_11314354_21088111_1431569418/96
http://qzs.qq.com/open/mobile/login/qzsjump.html?
http://rqd.uu.qq.com/rqd/sync
http://schemas.android.com/apk/res/android
http://shnk.fcloud.store.qq.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lqs.x
http://shp.qpic.cn/ma_icon/0/icon_5848_17266235_1390032639/72
http://unipay.sdk.android/?
http://update.sdk.jiguang.cn/v1/push/sdk/postlist
http://www.qq.com
```

Fig: 4

There are so many permissions for the android file that has been defined in Androidmanifest.xml.We can easily understand that the file is malicious because it has few access permissions that are not given to any social networking applications. In the below Fig:5, we can see the suspicious permission for this app like CACHE_READ & WRITE, DEVICE_POWER, MOUNT_UNMOUNT_FILESYSTEMS, INSTALL_PACKAGES, these permissions leads to install other malicious applications.

```
<uses-permission android:name="com.qq.AppService.permission.out.IPC_SERVICE"/>
<uses-permission android:name="com.qq.superuser.READ_PERMISSION"/>
<uses-permission android:name="com.qq.AppService.permission.out.CACHE_READ_PERMISSION"/>
<uses-permission android:name="com.qq.AppService.permission.out.CACHE_WRITE_PERMISSION"/>
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.android.launcher.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.DEVICE_POWER"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.WRITE_APN_SETTINGS"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.DELETE_PACKAGES"/>
<uses-permission android:name="android.permission.MANAGE_USERS"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Fig: 5

The file has permission to READ_LOGS that is suspicious. It also has the access to read phone state and it can change the settings of the internet to connect with the malicious domain. The Write_Settings permission can disable the system antivirus.

```
<uses-permission android:name="android.permission.BROADCAST_STICKY"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.BATTERY_STATS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.SET_WALLPAPER"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
```

Fig: 6

The API call (Fig:7) shows that it will gather some valuable info like root access information so that it can access it by changing the settings.

```
JceCmd GetRestaurantList = new JceCmd(TbsListener.ErrorCode.INFO_FORCE_SYSTEM_
JceCmd GetRestaurantQueue = new JceCmd(TbsListener.ErrorCode.INFO_MISS_SDKEXTI
JceCmd GetRootAccessInfo = new JceCmd(213, 3014, "GetRootAccessInfo");
JceCmd GetRubbishInfo = new JceCmd(TbsListener.ErrorCode.DEXOAT_EXCEPTION, _Ge
JceCmd GetScanHeadData = new JceCmd(251, _GetScanHeadData, "GetScanHeadData");
```

Fig: 7

The login information saved by another application like mobile banking app, mailing app, and other social app, that can also be gathered by the sample. After collecting the information it will either delete the information or pass it to another end.

```
}

public List GetAllLoginInfo() {
    return this.mG.k();
}
```

Fig: 8

The sample collects the device locking information that is shown in the source code of the file.

```
    }

    public DevlockInfo GetDevLockInfo(String str) {
        return GetDevLockInfo(str, 0);
    }
```

Fig: 9

All the keys can be opened without the password so sample will do that by this API call.

```
    public int GetOpenKeyWithoutPasswd(String str, long j, long j2, int i,
        byte[][] bArr = null;
        return GetStWithoutPasswd(str, j, this.mOpenAppid, -1, i, j2, null,
    }
```

Fig: 10

'ExternalInstall' permission installs multiple malicious adware from the above URLs which leads to excessive battery consumption of the device and tries to stop working. There is the request proxy permission that helps the sample to change the network or Wi-Fi settings to proxy state and it connects with those URLs using the proxy network.

```
' android:name="com.tencent.assistant.manager.permission.PermissionRequestProxyActiv
' android:name="com.tencent.pangu.manager.ExternalInstallPermissionRequestActivity"
' android:name="com.tencent.pangu.link.LinkImplActivity" android:enabled="true" andr
:.assistant.activity.InstalledAppManagerActivity" android:launchMode="singleTask" an
:.assistant.activity.RootUtilInstallActivity" android:launchMode="singleTask" androi
```

Fig: 11

IOCS:

Malicious URLs:

http://106.38.181.148/dd.myapp.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lqs.
http://qzs.qq.com/open/mobile/login/qzsjump.html?
http://shnk.fcloud.store.qq.com/16891/58D327D09437660FC1BBD10298EA5427.apk?fsname=com.tencent.tmgp.lqs.xy_1.0.400_1000400.apk&__k1__=y
http://www.qq.com
http://imgcache.qq.com/wtlogin
http://youxi.vip.qq.com/m/act/

MITRE Techniques:

Install Insecure or Malicious Configuration(T1478)
Masquerade as Legitimate Application(T1444)
Access Sensitive Data in Device Logs(T1413)
Deliver Malicious App via Other Means (T1476)
Generate Fraudulent Advertising Revenue(T1472)

Sectrio Protection
Sectrio detects the Spyware-Trojan malware as 'SS_Gen_Android_Infector_A**'.**

Our Honeypot Network
This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 cities across the world. These cities have at least one of the following attributes:

- **Are landing centers for submarine cables**
- **Are internet traffic hotspots**
- **House multiple IoT projects with a high number of connected endpoints**
- **House multiple connected critical infrastructure projects**
- **Have academic and research centers focusing on IoT**
- **Have the potential to host multiple IoT projects across domains in the future**

Over 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.