南京郵電大學
Nanjing University of Posts and Telecommunications

手机取证——获取IMEI

汇报人：胡东博

CONTENT

背景介绍

IMEI

国际移动设备识别码（International Mobile Equipment Identity，IMEI），即通常所说的手机序列号、手机“串号”，用于在移动电话网络中识别每一部独立的手机等移动通信设备，相当于移动电话的身份证。

序列号共有15~17位数字：
前8位（TAC）是型号核准号码（早期为6位），是区分手机品牌和型号的编码。
接着2位（FAC）是最后装配号（仅在早期机型中存在），代表最终装配的代码。
后6位（SNR）是串号，代表生产顺序号。

■ Android系统版本

Android系统版本从最老的1、2、3到目前最新的Android 14，各个系统版本有一个对应的API号。

表1 几个Android开发常用的系统版本及API

Android版本	名称	API
Android 10.0	Q	29
Android 9.0	P	28
Android 8.0	O	26
Android 6.0	M	23
Android 4.4	K	19

初次尝试

IMEI获取

最开始使用的函数：TelephonyManager.getDeviceId()

但是后来发现，这个函数对Android 8.0以上的手机并不适用。

原因：Android 8.0后API函数改成了TelephonyManager.getImei()

```
private void displayIdentifiers() {
    TelephonyManager telephonyManager = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
    TextView textView = findViewById(R.id.imeiTextView);

    if (telephonyManager != null) {
        StringBuilder identifierBuilder = new StringBuilder();
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
Android 8 to 9
            String imei = telephonyManager.getImei();
            identifierBuilder.append("IMEI: ").append(imei);
        } else {
Below Android 8
            String imei = telephonyManager.getDeviceId();
            identifierBuilder.append("IMEI: ").append(imei);
        }
        textView.setText(identifierBuilder.toString());
    }
}
```

图1 初次尝试获取IMEI

IMEI获取

问题：

1、如何获取两个IMEI值?

2、如何获取Android 10.0及以上版本的IMEI?

源码分析

美亚手机取证大师分析

使用美亚手机取证大师，发现他可以获取任意手机型号和系统的多个IMEI值。

图2 LG手机检材

图4 华为mate10pro手机检材

图5 华为p30pro手机检材

■ 美亚手机取证大师分析

发现电脑版美亚手机取证大师取证的时候会在手机上安装一个APP，所有的操作都是在这个APP上进行的。

可以分析这个APP，看看美亚是怎么实现的。

图6 美亚APP

■ 美亚手机取证大师分析

使用jadx打开该APP，直接搜索IMEI，发现有24处代码涉及到。

标出的这个函数引用可以发现有getImei和getDeviceId这两个函数，大概率是获取IMEI的方法。

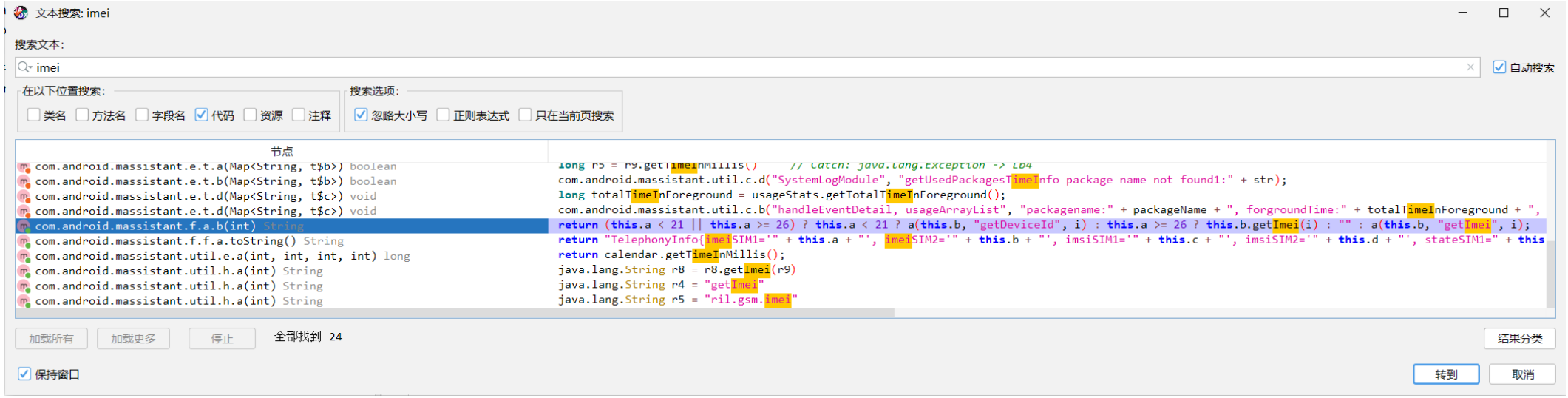

图7 imei代码文本搜索

■ 美亚手机取证大师分析

定位到上述函数，可以发现美亚APP也会对系统版本进行匹配，以使用正确的API函数。和前面分析的不同版本函数使用不同是一致的。

```
@SuppressLint({"NewApi"})
public String b(int i) {
    try {
        try {
            return (this.a < 21 || this.a >= 26) ? this.a < 21 ? a(this.b, "getDeviceId", i) : this.a >= 26 ? this.b.getImei(i) : "" : a(this.b, "getImei", i);
        } catch (C0031a unused) {
            return a(this.b, "getDeviceIdGemini", i);
        }
    } catch (C0031a unused2) {
        return i == 0 ? this.b.getDeviceId() : "";
    }
}
```

图8 美亚imei获取函数

美亚手机取证大师分析

查找该函数的引用情况。

查找用例： com.android.massistant.f.a.b(int) String

节点	
com.android.massistant.f.b.a(Context) a	this.c.b(b(0));
com.android.massistant.f.b.a(Context) a	this.c.c(b(1));
com.android.massistant.f.b.b(int) String	return TextUtils.isEmpty(str) ? super.b(i) : str;
com.android.massistant.f.b.b(int) String	return super.b(i);
com.android.massistant.f.b.b(int) String	return super.b(i);
com.android.massistant.f.c.a(Context) a	this.c.b(b(0));
com.android.massistant.f.c.a(Context) a	this.c.c(b(1));
com.android.massistant.f.d.a(Context) a	this.c.b(b(0));
com.android.massistant.f.d.a(Context) a	this.c.c(b(1));
com.android.massistant.f.d.b(int) String	return TextUtils.isEmpty(str) ? super.b(i) : str;
com.android.massistant.f.d.b(int) String	return super.b(i);
com.android.massistant.f.d.b(int) String	return super.b(i);
com.android.massistant.f.e.a(Context) a	this.c.b(b(0));
com.android.massistant.f.e.a(Context) a	this.c.c(b(1));
com.android.massistant.f.e.b(int) String	return TextUtils.isEmpty(str) ? super.b(i) : str;
com.android.massistant.f.e.b(int) String	return super.b(i);
com.android.massistant.f.e.b(int) String	return super.b(i);

图9 美亚imei获取函数引用

■ 美亚手机取证大师分析

逐个点进这些函数，发现了猫腻。

这里以其中一段代码为例。

这是一个三星型号的代码段，红色框里的是前面找到的imei获取函数，这里赋值了0和1。

美亚APP通过给getImei()赋值0和1，来区分两个IMEI号。

```
@Override // com.android.massistant.f.a
public a a(Context context) {
    this.c = new f.a();
    this.c.a("Samsung");
    this.c.a(a(0));
    this.c.b(a(1));
    this.c.c(b(context));
    this.c.b(b(0));
    this.c.c(b(1));
    return this;
}

@Override // com.android.massistant.f.a
public String b(int i) {
    if (this.a < 21) {
        try {
            String str = (String) a(c(i), "getDeviceId", null, null);
            return TextUtils.isEmpty(str) ? super.b(i) : str;
        } catch (Exception e) {
            e.printStackTrace();
            return super.b(i);
        }
    }
    return super.b(i);
}
```

图10 三星获取IMEI

再次尝试

IMEI获取

仿照美亚APP的写法，在处理getImei函数的时候，给他赋值0或1，以此获取两个IMEI号。

```
private void displayIdentifiers() {
    TelephonyManager telephonyManager = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
    TextView textView = findViewById(R.id.imeiTextView);

    if (telephonyManager != null) {
        StringBuilder identifierBuilder = new StringBuilder();
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
Android 8 to 9
            int slotCount = telephonyManager.getPhoneCount();
            for (int i = 0; i < slotCount; i++) {
                String imei = telephonyManager.getImei(i);
                if (imei != null) {
                    identifierBuilder.append("IMEI ").append(i + 1).append(": ").append(imei).append("\n");
                }
            }
        } else {
Below Android 8
            String imei = telephonyManager.getDeviceId();
            identifierBuilder.append("IMEI: ").append(imei).append("\n");
        }
        textView.setText(identifierBuilder.toString());
    }
}
```

图11 再次尝试获取IMEI

IMEI获取

仿照美亚APP的写法，在处理getImei函数的时候，给他赋值0或1，以此获取两个IMEI号。

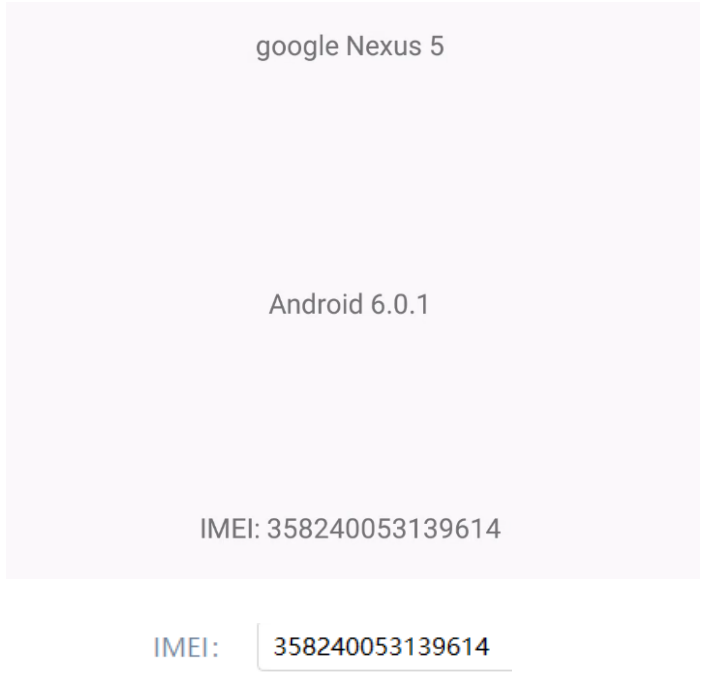

图12 GL手机IMEI

图13 mate10pro IMEI

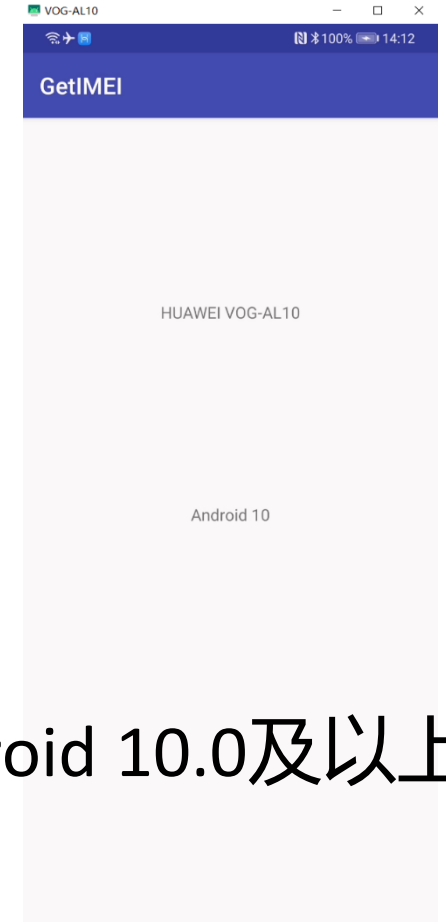

图14 p30pro无数据

Android 10.0及以上怎么获取呢?

■ 美亚手机取证大师分析

继续回到前面IMEI获取函数的引用部分，深入分析美亚APP，发现了更多的SDK和API函数。

```
@Override // com.android.massistant.f.a
public a a(Context context) {
    this.c = new f.a();
    this.c.a("Samsung");
    this.c.a(a(0));
    this.c.b(a(1));
    this.c.c(b(context));
    this.c.b(b(0));
    this.c.c(b(1));
    return this;
}

@Override // com.android.massistant.f.a
public String b(int i) {
    if (this.a < 21) {
        try {
            String str = (String) a(c(i), "getDeviceId", null, null);
            return TextUtils.isEmpty(str) ? super.b(i) : str;
        } catch (Exception e) {
            e.printStackTrace();
            return super.b(i);
        }
    }
    return super.b(i);
}
```

图15 三星获取IMEI

```
@Override // com.android.massistant.f.a
public a a(Context context) {
    this.c = new f.a();
    this.c.a("MTK");
    this.c.a(a(0));
    this.c.b(a(1));
    this.c.c(b(context));
    this.c.b(b(0));
    this.c.c(b(1));
    return this;
}
```

图16 MTK（联发科）获取IMEI

```
@Override // com.android.massistant.f.a
public a a(Context context) {
    this.c = new f.a();
    this.c.a("Qualcomm");
    this.c.a(a(0));
    this.c.b(a(1));
    this.c.c(b(context));
    this.c.b(b(0));
    this.c.c(b(1));
    return this;
}
```

图17 高通获取IMEI

结论1：美亚APP会根据芯片型号获取IMEI值

再次尝试

- **美亚手机取证大师分析**

看看上述这些[illegible]

```
private Object c(int i) {
    try {
        if (this.a < 21) {
            if (this.d == null)
                this.d = Class.
            }
            return a(this.d, nu
        } else if (this.h == nu
            Object newInstance
            this.h = newInstanc
            return newInstance;
        } else {
protected Object c() {
    try {
        return a(Class.forName("com.mediatek.telephony.TelephonyManagerEx"), null, "getDefault", null, null);
    } catch (Exception e) {
        Log.d("MTKDualSim", "isMTKDoubleSim-error:" + e.getMessage());
        return null;
    }
}
```

```
private boolean e(Context context) {
    if ("CMDC".equals(Build.MANUFACTURER)) {
        Log.d("QualcommDualSim", "!cmdc");
        return false;
    }
    if (this.a >= 21) {
        try {
            return ((Boolean) a(this.b, "isMultiSimEnabled
        } catch (a.C0031a e) {
int indexOf;
if (!"nubia".equals(Build.MANUFACTURER.toLowerCase(Locale.ENGLISH))) {
    Log.d("QualcommDualSim", "!nubia");
    return false;
}
try {
} catch (Exception e) {
    e.printStackTrace();
}
if (this.a >= 21)
    return ((Boole
}
String a = a("ro.p
Log.d("QualcommDua
if (!TextUtils.isE
    Log.d("Qualcom
    return true;
if ("huawei".equals(Build.MANUFACTURER.toLowerCase(Locale.E
    try {
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (this.a >= 21) {
        return ((Boolean) a(this.b, "isMultiSimEnabled", nu
    }
    String a = a("ro.board.platform");
    Log.d("QualcommDualSim", "huawei-execResult:" + a);
    return !TextUtils.isEmpty(a) && a.equals("hi3630");
}
```

图19 MTK判断多个IMEI

```
private boolean e(Context context) {
    if (!"CMDC".equals(Build.MANUFACTURER)) {
        Log.d("QualcommDualSim", "!cmdc");
        return false;
    }
    if (this.a >= 21) {
        try {
            return ((Boolean) a(this.b, "isMultiSimEnabled", null, null)).booleanValue();
        } catch (a.C0031a e) {
            e.printStackTrace();
        }
    } else {
        try {
            String a = a("persist.loc.nlp_name");
            if (!TextUtils.isEmpty(a)) {
                if (a.equals("com.qualcomm.location")) {
                    return true;
                }
            }
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
    return false;
}

private boolean f(Context context) {
    int indexOf;
    if (!"nubia".equals(Build.MANUFACTURER.toLowerCase(Locale.ENGLISH))) {
        Log.d("QualcommDualSim", "!nubia");
        return false;
    }
    try {
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (this.a >= 21) {
        return ((Boolean) a(this.b, "isMultiSimEnabled", null, null)).booleanValue();
    }
    String a = a("ro.product.board");
    Log.d("QualcommDualSim", "nubia execResult:" + a);
    if (!TextUtils.isEmpty(a) && (indexOf = a.toLowerCase().indexOf("msm")) >= 0) {
        Log.d("QualcommDualSim", "nubia index:" + indexOf);
        return true;
    }
    return false;
}

private boolean g(Context context) {
    if ("huawei".equals(Build.MANUFACTURER.toLowerCase(Locale.ENGLISH))) {
        try {
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (this.a >= 21) {
            return ((Boolean) a(this.b, "isMultiSimEnabled", null, null)).booleanValue();
        }
        String a = a("ro.board.platform");
        Log.d("QualcommDualSim", "huawei-execResult:" + a);
        return !TextUtils.isEmpty(a) && a.equals("hi3630");
    }
```

图20 高通判断多个IMEI

结论2：美亚APP会根据芯片型号以及对应的手机厂家API来判断是否存在多个IMEI

总结

IMEI获取

为什么Android 10.0及以上获取IMEI的流程要那么麻烦了？

正如背景介绍，IMEI等同于一个手机的身份证，手机号换了，IMEI号是不会变的，IMEI号和使用手机的人深度绑定。

轻易获取IMEI号很容易泄露个人隐私数据，因此获取高系统版本手机的IMEI号都需要芯片厂商或手机厂家的授权，即获得他们提供的SDK。

至于如何不使用IMEI确定设备唯一性，Google和中国国内的一些厂商推出了AndroidID、OAID、UUID等指纹。同时还可以借助获取手机的IP、MAC等信息辅助判断。

感谢指导