2017 年

Android 恶意软件专题报告

2018 年 3 月 01 日

摘　要

- 2017年全年，360互联网安全中心累计截获Android平台新增恶意软件样本757.3万个，平均每天新增2.1万。全年相比2016年（1403.3万）下降46.0%，从2015年来看，新增恶意软件呈现总体下降趋势。
- 2017全年，从手机用户感染恶意软件情况看，360互联网安全中心累计监测到Android用户感染恶意软件2.14亿，相比2016年2.53亿人次下降15.4%，平均每天恶意软件感染量约为58.5万人次。
- 2017年Android平台新增恶意软件主要是资费消耗，占比高达80.2%；相比2016年增加了6个百分点。
- 2017 年从地域分布来看，感染手机恶意软件最多的地区为广东省，感染数量占全国感染数量的10.4%；其次为河南（6.8%）、山东（6.5%）、河北（5.9%）、浙江（5.9%）。
- 2017 年 Android 平台恶意软件感染量最多的十大城市。北京（4.9%）、广州（2.1%）、重庆（1.8%）、成都（1.7%）、东莞（1.5%）。位居Top10的城市还有石家庄、深圳、郑州、南京、杭州。
- 2017年恶意软件使用了多种新技术，分别是针对系统运行库的攻击，利用Telegram软件协议远控，手机挖矿，手机僵尸网络发起DDOS攻击，使用SOCKS代理和SSH协议穿透内网防火墙，恶意软件多级化，滥用应用多开技术以及高级定向攻击持续化。
- 2017年度CVE Details报告显示，Android系统以842个漏洞位居产品漏洞数量榜首，与2016年523个相比，增长61.0%。
- 2017 年不断曝出 Android 漏洞在恶意样本上利用，利用的漏洞主要有屏幕录制漏洞（CVE-2015-3878）、脏牛漏洞（CVE-2016-5195）、TYPE_TOAST（CVE-2017-0752）和Janus安卓签名漏洞（CVE-2017-13156）。
- Google通过引入最新的机器学习模块和技术，显著提升了Google Play的安全检测能力，并能够有效发现假冒的软件、包含了违规内容的软件、以及恶意软件。除此之外，Google还从系统和研发两个方面提升了整体安全环境。
- 在协同打击网络犯罪方面，截至2017年底，猎网平台已与全国300个地区的公安机关建立联系。全年协助各地区公安机关协查案件219起，破案23起，抓获嫌疑人共计137人。同时，360烽火实验室对外开展移动平台电子取证培训10余次，涵盖河南、长春、浙江等全国多地。在涉及移动平台的网络犯罪案件侦办中，实验室通过溯源等分析手段，协助公安机关找到恶意软件作者QQ号，手机号及邮箱线索31万余条，案件涉及的恶意软件分析报告18篇。
- 从移动威胁趋势上看，具备自动化和对抗能力的恶意软件工厂不断涌现，恶意挖矿木马愈演愈烈，公共基础服务成为恶意软件利用的新平台，脚本语言成为恶意软件新的技术热点，这4个方面将成为今后的主要趋势。

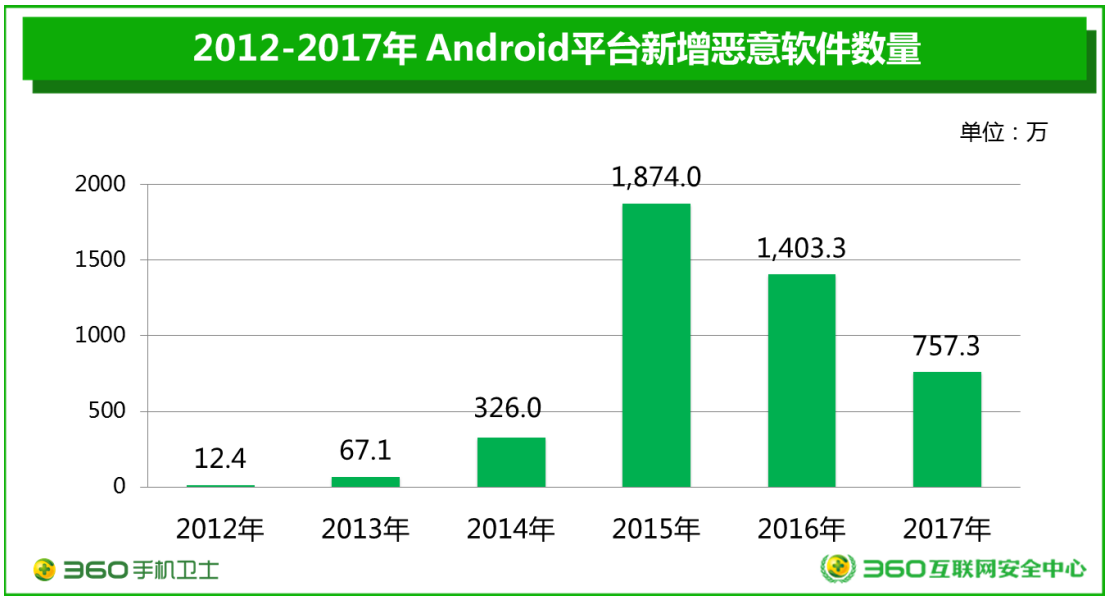
关键词：移动安全、恶意软件、漏洞利用、网络犯罪、威胁趋势

目　　录

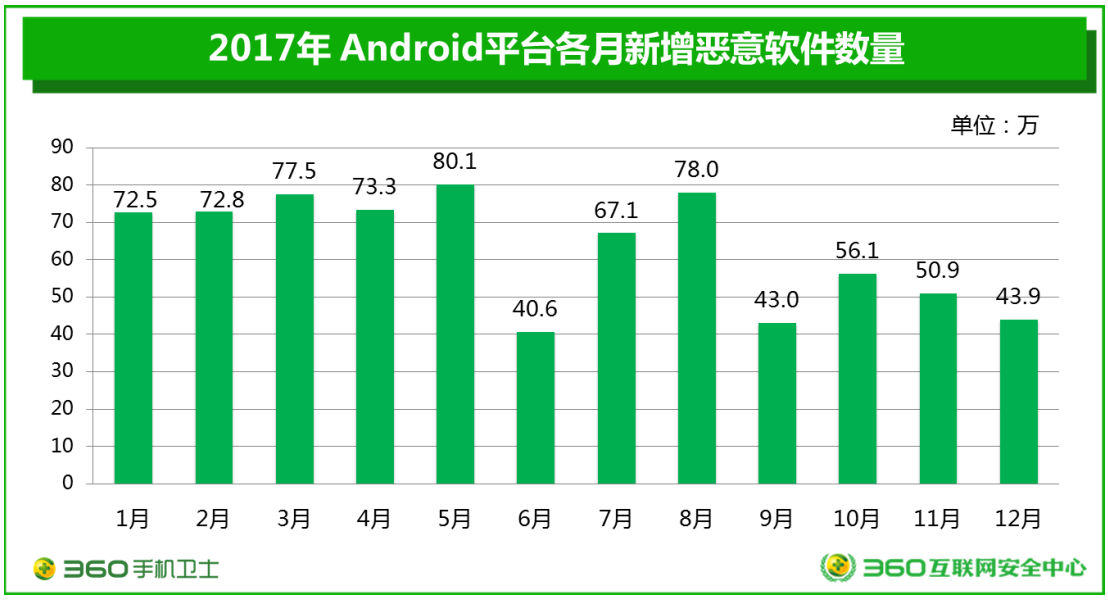
第一章 总体态势

一、 恶意软件新增量与感染量

2017 年全年，360 互联网安全中心累计截获 Android 平台新增恶意软件样本 757.3 万个，平均每天新增 2.1 万。全年相比 2016 年（1403.3 万）下降 46.0%，从 2015 年来看，新增恶意软件呈现总体下降趋势，且今年下降幅度较大，显示了移动恶意软件总体进入平稳高发期。

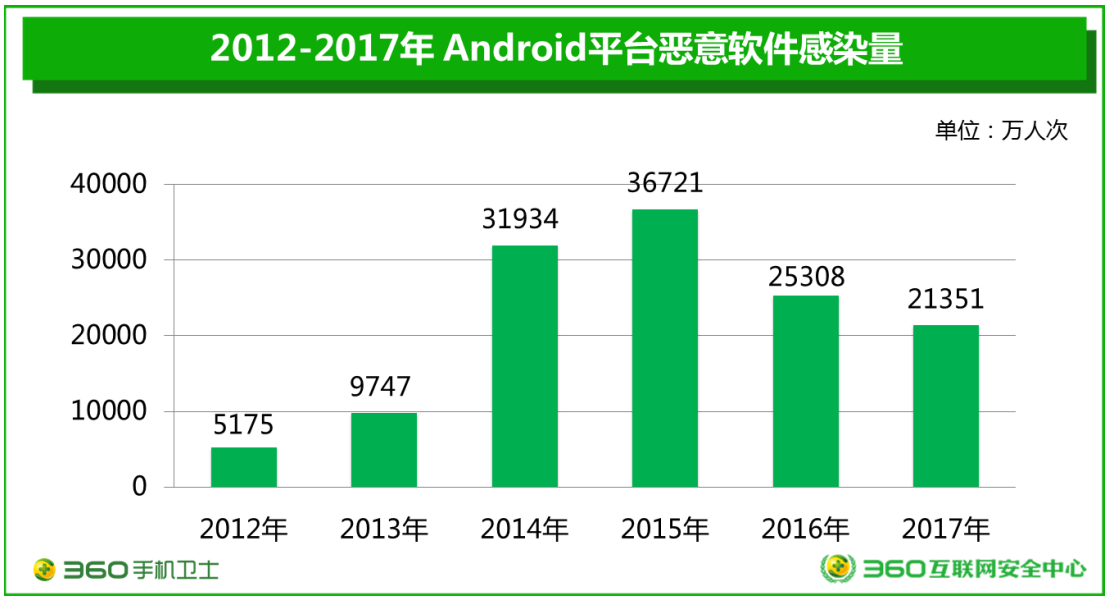


下图是 2017 年各月 Android 平台新增恶意软件样本量的分布图。由图可见，新增恶意软件整体呈现上半年高、下半年低的态势，即 1-5 月新增恶意软件量整体呈现曲线上升，在 5 月达到最高峰。下半年除 8 月为 78.0 万个新增样本外，其余月份均较低。



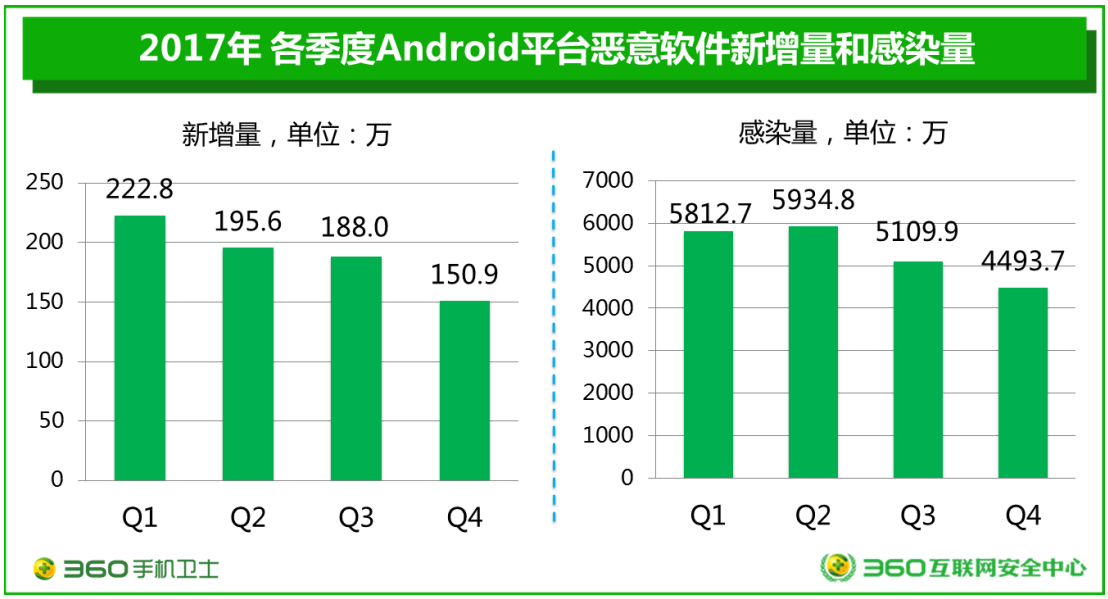
2017 全年，从手机用户感染恶意软件情况看，360 互联网安全中心累计监测到 Android 用户感染恶意软件 2.14 亿，相比 2016 年 2.53 亿人次下降 15.4%，平均每天恶意软件感染量约为 58.5 万人次。

从近六年的移动恶意软件感染人次看，经过 2012-2015 年的高速增长期，2016 和 2017 年呈现下降趋势，说明手机恶意软件进入平稳期。



下图是 2017 年 Android 平台新增恶意软件感染量的按季度对比情况，每季度的新增恶意样本均在下降。

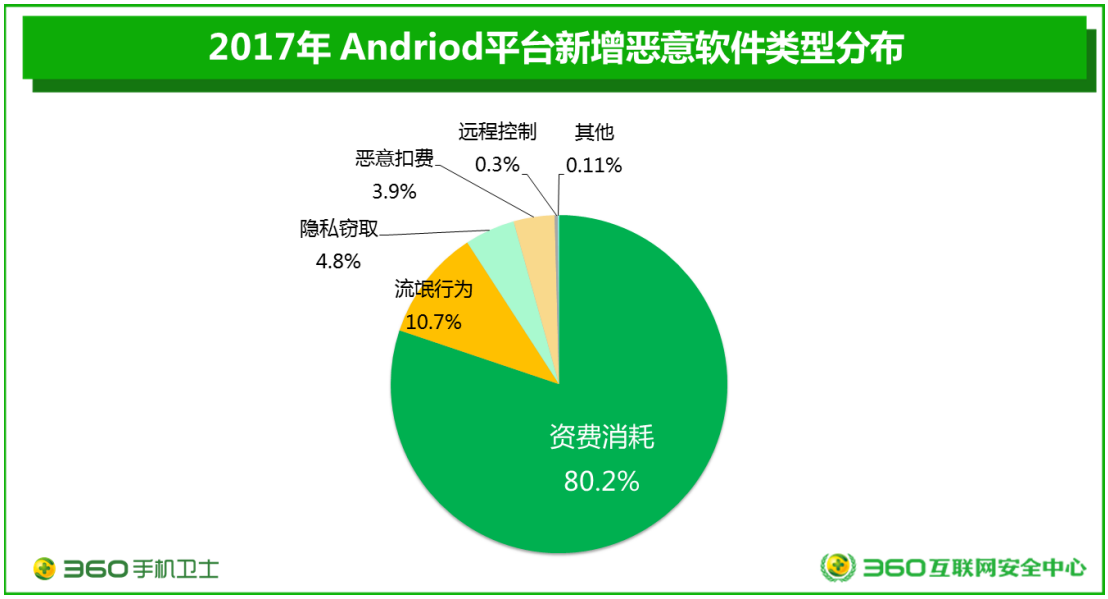
全年来看，2017 年四个季度的感染量呈现下降趋势。其中二季度最高约为 5934.8 万人次，四季度的感染量则最少，仅为 4493.7 万人次。



二、 恶意软件危害分析

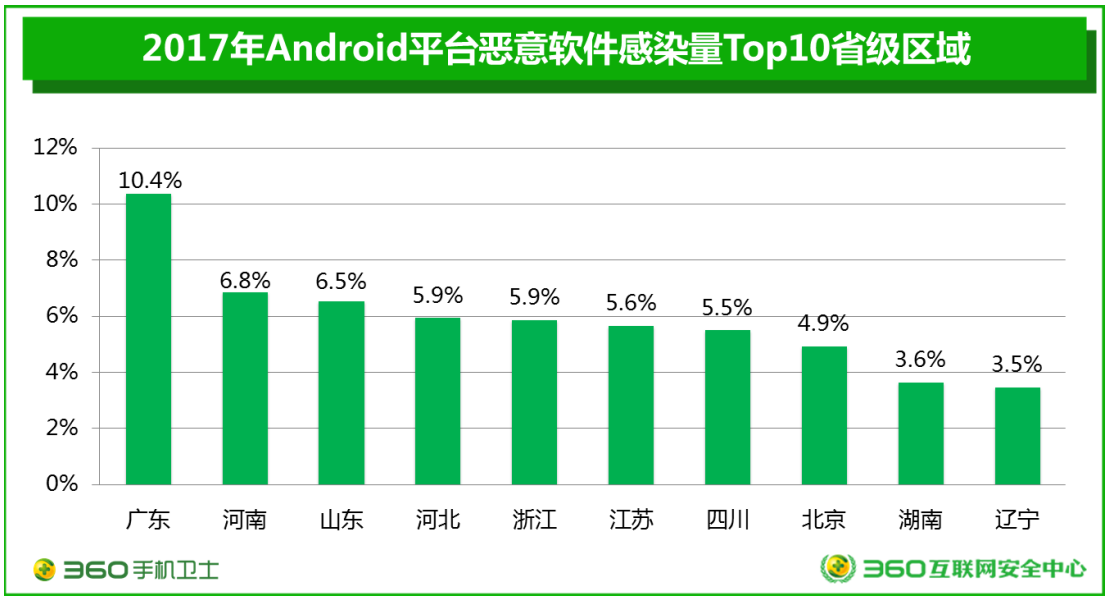
根据中国反网络病毒联盟的分类标准，360 互联网安全中心在 2017 全年监测的 Android 平台恶意软件的分类统计如下图。从图中可见，2017 年 Android 平台新增恶意软件主要是资费消耗，占比高达 80.2%；相比 2016 年增加了 6 个百分点。

资费消耗类型的恶意样本占比已达到 3/4，说明移动端恶意软件依然是以推销广告、消耗流量等手段，增加手机用户的流量资费等谋取不法商家的经济利益。当前主流运营商的资费模式重心已经转向流量，而不再单纯倚重语音通话。资费消耗类恶意软件对用户资费造成的影响还是比较明显。

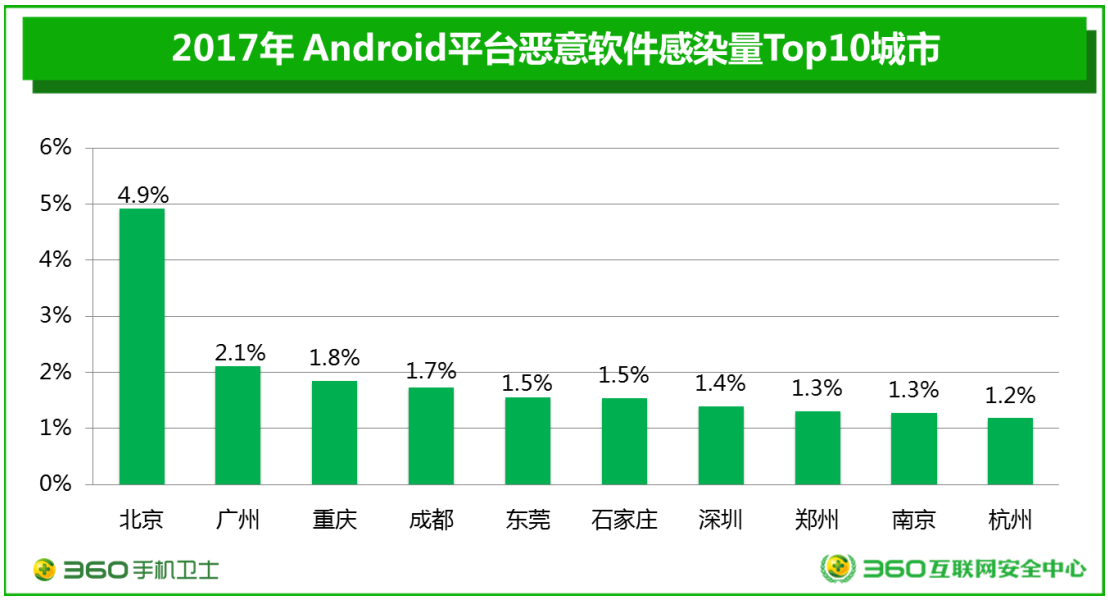


三、 地域分析

2017 年从地域分布来看，感染手机恶意软件最多的地区为广东省，感染数量占全国感染数量的 10.4%；其次为河南（6.8%）、山东（6.5%）、河北（5.9%）、浙江（5.9%）。



下图给出了 2017 年 Android 平台恶意软件感染量最多的十大城市。毫无疑问，北京用户感染 Android 平台恶意软件最多，占全国城市的 4.9%；其次是广州（2.1%）、重庆（1.8%）、成都（1.7%）、东莞（1.5%）。位居 Top10 的城市还有石家庄、深圳、郑州、南京、杭州。



第二章 盘点恶意软件的新技术

一、 针对系统运行库的攻击出现

今年 4 月在 Google Play 应用商店上发现了新的系统级恶意软件 Dvmap[1]。它根据 Android 系统的版本将恶意代码注入到系统库 libdmv.so 或 libandroid_runtime.so 中，这两个库都是与 Dalvik 和 ART 运行时环境相关的运行时库。注入之后会以 Root 权限替换系统正常文件，同时部署恶意模块。恶意模块能够关闭谷歌对于应用的安全检查（Verify Apps）功能，并且更改系统设置去操作安装任何来自第三方应用市场的应用程序。

```
if ( a1
  && (v2 = sub_12BC("/system/lib/libdvm.so", "_Z30dvmHeapSourceStartupBeforeForkv"),
      v3 = fopen("/system/lib/libdvm.so", "rb"),
      (v4 = v3) != 0) )
{
```

```
v1 = a1;
if ( a1
  && (v2 = sub_14F0("/system/lib/libandroid_runtime.so", "nativeForkAndSpecialize"), v2 != -1)
  && (v3 = fopen("/system/lib/libandroid_runtime.so", "rb"), (v4 = v3) != 0) )
{
```

```
sub_221C("settings put secure package_verifier_user_consent -1");
sub_221C("settings put global package_verifier_enable 0");
sub_221C("settings put global upload_apk_enable 0");
sub_221C("settings put secure install_non_market_apps 1");
```

图 2.1：Dvmap 家族代码片段

二、 利用 Telegram 软件协议的木马首次出现

今年 6 月国外安全厂商首次发现利用 Telegram 软件协议控制的木马 Android.Spy.377.origin[2]。它不但能够盗取用户手机中的隐私信息，并且还可以远程控制手机拨打电话、发送短信，删除手机中指定文件等。

Telegram 是以云端为基础的轻量级即时通讯软件，Telegram Bot 是基于 Telegram 客户端的第三方程序。用户可以通过向 Bot 发送信息、照片、指令、在线请求等一系列的方式于 Bot 互动。Bot 的所有者通过 Bot 的 API 访问并请求 Telegram Server 的信息。

由于电报信息交换协议提供了简单的通信方式，攻击者无需在受害者的设备上启用端口转发。然而攻击者首先需要开发自己的 Bot，这个 Bot 生成的令牌嵌入到木马的配置文件中。一旦受害者设备被感染，攻击者就能够通过自动创建的通道来控制它。

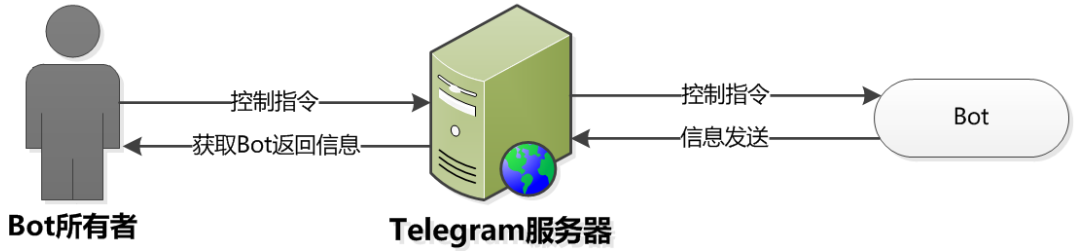


图 2.2：使用 Telegram protocol 控制原理

三、 挖矿木马再现身

挖矿木马最早是 2013 年在 PC 平台上被发现，而首个手机挖矿木马 CoinKrypt[3]最早被国外安全厂商在 2014 年 3 月曝光。手机挖矿木马经过一阵沉寂后，随着电子加密货币价格的一路走高，恶意软件作者又重新将目标转向了挖矿。

在代码层上的表现形式为，早期恶意软件使用了开源的矿池代码库进行挖矿，今年曝光的恶意软件使用矿池提供的浏览器 JavaScript 脚本进行挖矿。由于浏览器 JavaScript 挖矿脚本配置灵活简单，具有全平台化等特点，受到越来越多的恶意挖矿木马的青睐，同时也导致了利用 JavaScript 脚本挖矿的安全事件愈发频繁。

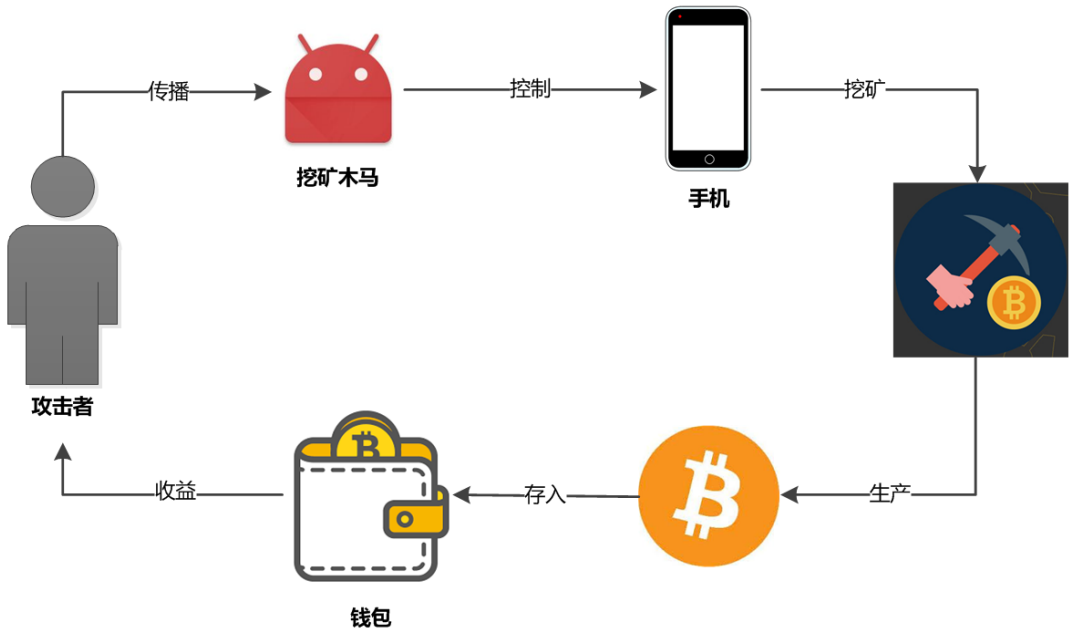


图 2.3：攻击者通过挖矿木马赚取收益的攻击流程

四、 僵尸网络发起 DDOS 攻击

2017 年 8 月多个内容分发网络（CDN）和内容提供商受到来自被称为 WireX 的僵尸网络的严重攻击。

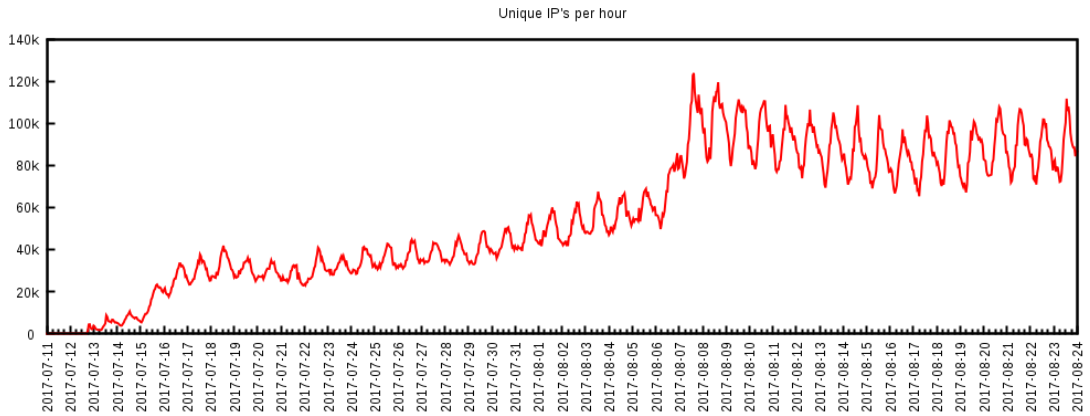


图 2.4：WireX 僵尸网络每小时增长量[4]

WireX 僵尸网络主要由运行恶意应用程序的 Android 设备组成，它使用了三种攻击方式：

1. UDP Flood：WireX 会创建 50 个线程，每个线程中都会连接该主机和端口，开启 Socket 之后，使用 UDP 协议发送随机数据，每次会发送 512 个字节的数据，一个线程中一共会发送一千万次，也就是 10000000*512=5120000000 字节的数据，因为一共实现了创建了 50 个线程，所以，理论上会发送 10000000*512*50=256000000000（2560 亿）字节。
2. Deceptive Access Attack：WireX 会创建 20 个 WebView，然后使用每个 WebView 访问要攻击的网站。
3. Deceptive Click Attack：WireX 会模拟鼠标事件进行点击，点击要攻击的网站页面中所有的 URL 链接。

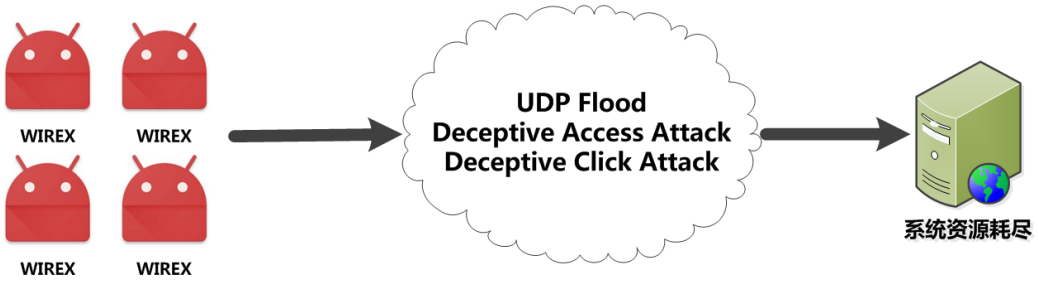


图 2.5：WireX 僵尸网络三种攻击方式

五、 企业攻击进阶

针对企业内网安全的攻击，继去年 6 月份首次出现的 DressCode[5]恶意家族后，今年又出现了利用移动设备攻击企业内网的新的恶意家族 MilkyDoor[6]。

然而，与 DressCode 不同的是，MilkyDoor 不仅利用 SOCKS 代理实现从攻击者主机到目标内网服务器之间的数据转发，而且利用 SSH(Secure Shell)协议穿透防火墙，加密传输数据，进而实现数据更隐蔽的传输。

MilkyDoor 木马采用远程端口转发实现数据加密传输，整个过程步骤如下：

1. 木马主动与攻击者主机建立一个 SSH 安全连接。
2. 攻击者主机将数据发送到它的 R 端口上。
3. 位于攻击者主机端的 SSH 服务器接收到 R 端口上的数据后，将其加密并转发到位于木马端的 SSH 客户端上。
4. SSH 客户端解密收到的数据并将其转发到木马监听的 L 端口上。

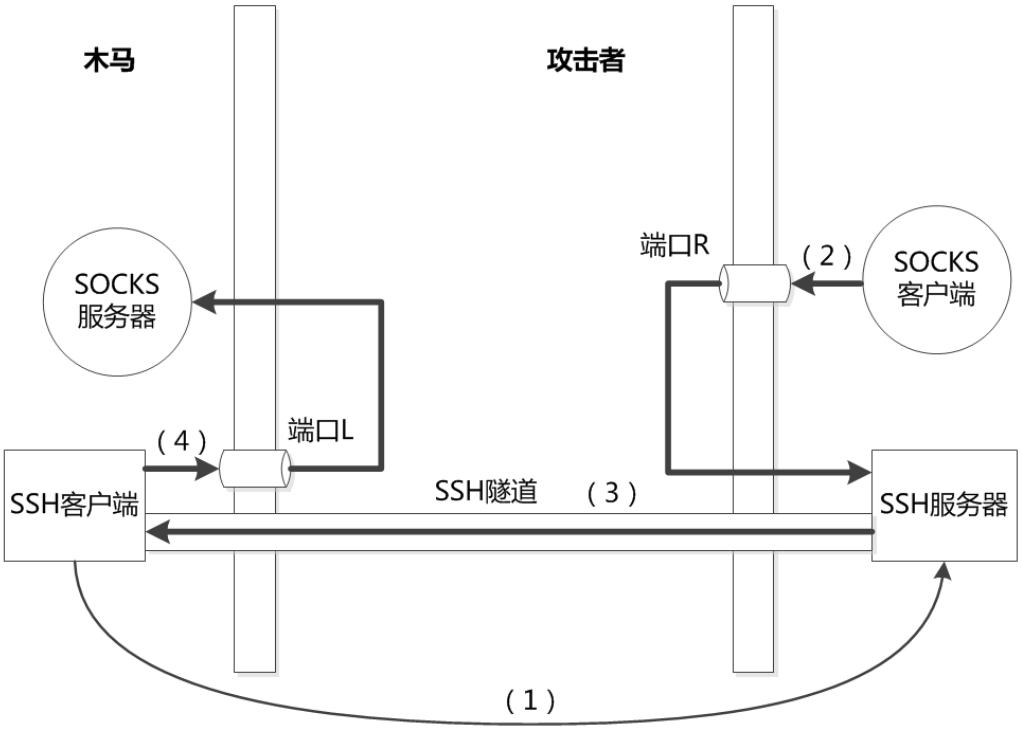


图 2.6：攻击者利用远程端口转发传输数据的过程

六、 恶意软件出现多级化

在恶意软件对抗方面，今年曝光的 Chamois 恶意家族，拥有多种分发渠道。它是 Google 认为迄今为止在 Android 平台上看到的最大的 PHA（Potentially Harmful Application）家族之一。

早在 2013 年在 Google Play 上就发现有通过联网指令控制延迟下载恶意软件，从而绕过 Google 扫描器检测阶段的恶意家族 Badnews[8]。相比 Badnews，Chamois 采用了多级化技术手段，它的代码使用不同的文件格式在 4 个不同的阶段执行。这个多阶段的过程使得这个家族在检测过程中变得更加复杂，因为必须执行第一阶段才能达到第二阶段，执行第阶段才能达到第三阶段，依次执行才能到达恶意的部分。

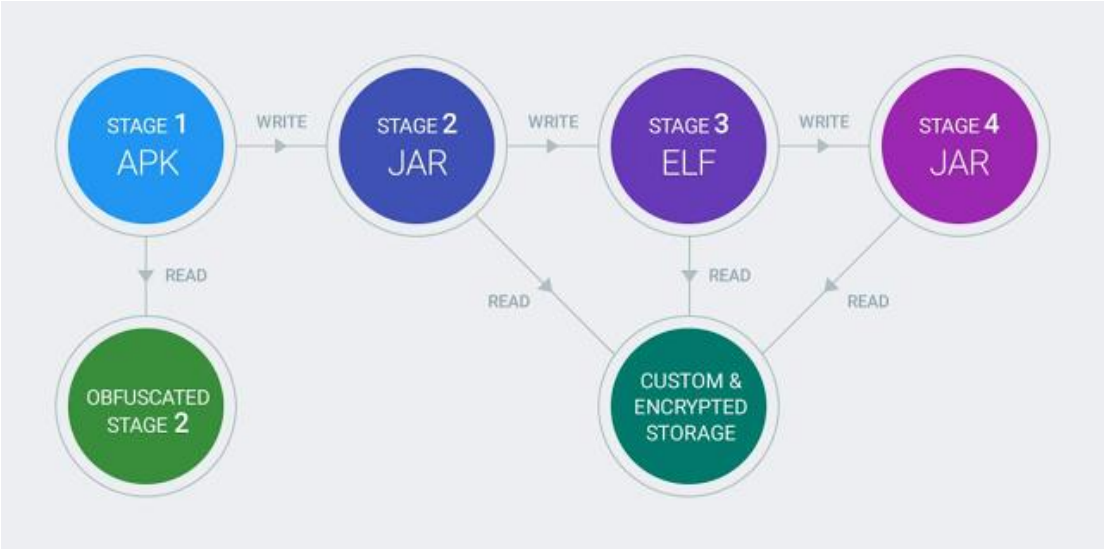


图 2.7：Chamois 恶意家族使用不同的文件格式在 4 个不同的阶段中执行流程[7]

七、 应用多开技术被滥用

VirtualApp[9]（简称 VA）是一个 App 虚拟化引擎。它能够创建一个虚拟空间，可以在虚拟空间内任意的安装、启动和卸载 APK，这一切都与外部隔离，如同一个沙盒。运行在 VA 中的 APK 无需在外部安装，即 VA 支持免安装运行 APK。VA 目前被广泛应用于双开/多开、应用市场、模拟定位、一键改机、隐私保护、游戏修改、自动化测试、无感知热更新等技术领域。

正是由于 VA 的应用广泛，也被恶意软件滥用。滥用实例主要为免杀、木马及广告。

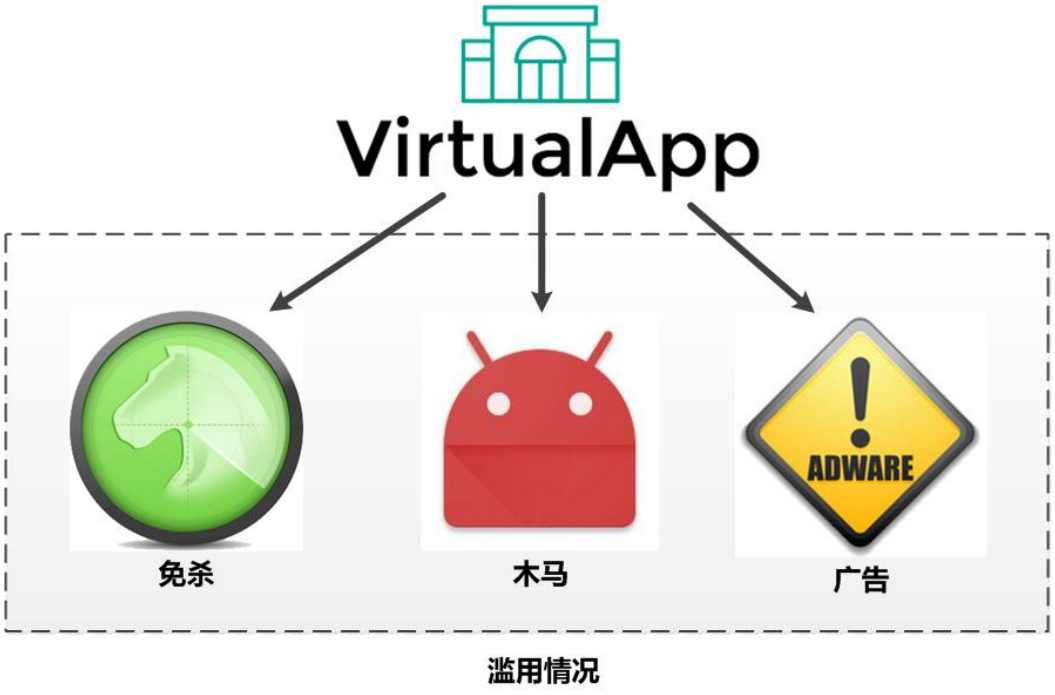


图 2.8：VirtualApp 滥用情况

1. 免杀：由于 VA 特性，恶意软件常以子包形式加密存储在 VA 内，在引擎扫描时使用 VA 的样本，主包代码特征表现一致，而子包由于加密导致引擎无法识别，从而绕过杀软静态检测。

2. 木马：以 Trojan-Spy.AndroidOS.Twittre[10]家族为例，通过 VA 启动 Twitter，Twitter 启动后，修改后的 VirtualCore 模块 Hook 了 EditText 类的 getText 函数，从而在 Twitter 登录窗口劫持用户的输入。用户的登录凭据被捕获后，恶意软件将其上传到远程服务器上。

3. 广告：今年使用 VA 技术的广告开始出现，通常使用 VA 的主包会在手机桌面创建了多个快捷方式，当用户点击快捷方式时，它会启动主包内相应的应用程序。对于这种启动方式与默认的启动方式相比很难进行区分，并且无需安装减少了用户操作过程，很大程度上既推广了应用，又避免了被用户发现。

八、高级定向威胁持续进行

APT 攻击（Advanced Persistent Threat，高级持续性威胁）堪称是在网络空间里进行的军事对抗，攻击者会长期持续的对特定目标进行精准的打击。

2017 年，曝光了一系列涉及移动平台的 APT 组织行动。与 2016 年相比，从披露的组织活跃度看，双尾蝎（APT-C-23）[11]组织在 2017 年中较为活跃。其在攻击中使用的恶意软件不断改进，所使用的新变种 VAMP[12]、FrozenCell[13]和 GnatSpy[14]相继曝光。

对比攻击目标和攻击国家，从攻击目标上，2017 年移动平台的 APT 攻击目标增加医疗、教育和金融方向；从攻击国家上，包括中国在内全球多个国家均是 APT 攻击的受害国。

从影响的移动平台看，去年曝光的 NSO Group[15]组织制作的恶意软件 Android 版本被发现，与 iOS 版本最大的不同是 Android 版本使用的是 Framaroot 方案提升软件权限，而没有使用零日漏洞。另外，像双尾蝎、Operation Manul[16]组织在 PC 端和移动端出现了相同控制 C&C 信息，这也表明 APT 攻击向着平台组合化方向发展。

组织/行动	恶意家族	影响的移动平台	被攻击国家	攻击目标
疑似Hamas	ViperRAT	Android	以色列	军事
Operation Manul Dark Caracal	DarkCaracal	Android	中国及其他20多个国家	军事、企业 医疗、金融、制造
Lazarus_Group Operation Blockbuster	Backdoor	Android	韩国	媒体、金融 宗教、关键基础设施
人面狮行动（APT-C-15）	AnubisSpy	Android	中东国家	政治、医疗
双尾蝎（APT-C-23）	GnatSpy VAMP FrozenCell	Android	巴勒斯坦	军事、教育

图 2.9：2017 年涉及移动平台的 APT 攻击行动

第三章 移动威胁持续进化

一、严峻的系统环境

Android 系统开源就意味着在安全问题上显得更加透明，运用工具审查安全漏洞变得更容易。根据汇总 CVE 数据的网站出具的 2017 年度 CVE Details 报告显示，Android 系统以 842 个漏洞位居产品漏洞数量榜首，与 2016 年 523 个相比，增长 61.0%，继续蝉联漏洞之王。

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Android	Google	OS	842
2	Linux Kernel	Linux	OS	453
3	Iphone Os	Apple	OS	387
4	Imagemagick	Imagemagick	Application	357
5	Mac Os X	Apple	OS	299
6	Windows 10	Microsoft	OS	268
7	Windows Server 2016	Microsoft	OS	252
8	Windows Server 2008	Microsoft	OS	243
9	Windows Server 2012	Microsoft	OS	235
10	Debian Linux	Debian	OS	230

图 3.1：2017 年 CVE 网站产品漏洞数量 TOP 排名情况[17]

Google 每次发布 Android 新版本，对系统安全性都有所增强，但是由于 Android 系统碎片化严重，系统版本更新速度慢，系统安全环境整体提升受到影响。

截止 2018 年 1 月，Google 发布的 Android 系统版本分布统计，Android Marshmallow（Android 6.0）总占比已达 28.6%，占比第二的是 Android Nougat（Android 7.0/7.1）达到 26.2%，而最新系统版本 Android Oreo（Android 8.0/8.1）仅占 0.7%。

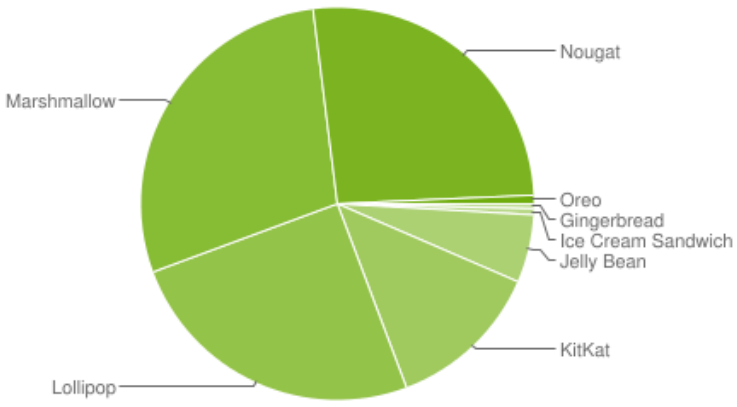
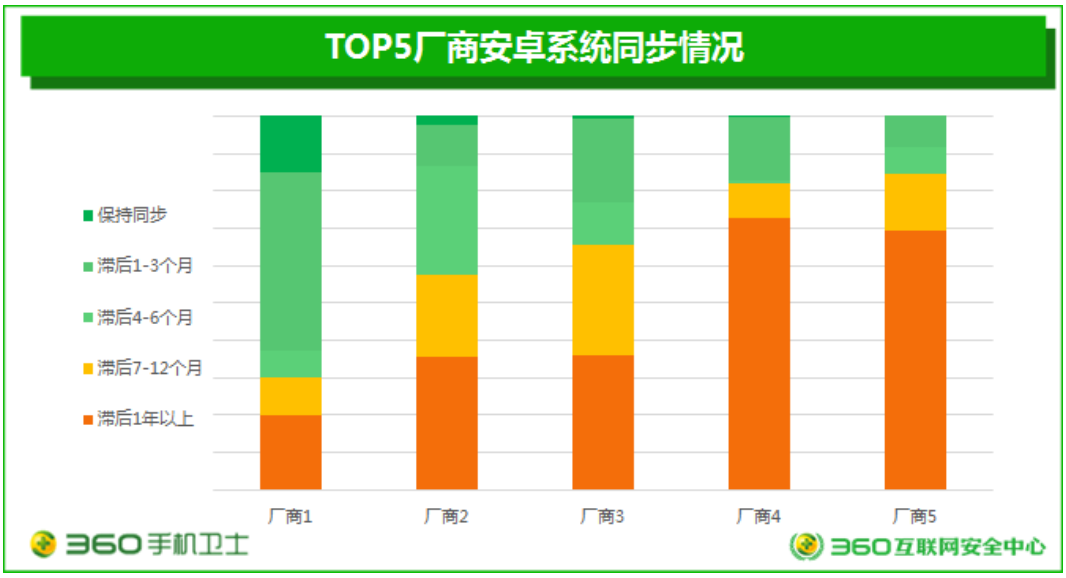


图 3.2：截止 2018 年 1 月 Android 系统版本分布占比情况[18]

二、厂商漏洞修复情况

Android 操作系统目前仍未有非常完善的补丁机制为其修补系统漏洞，再加上 Android 系统碎片化严重，各手机厂商若要为采用 Android 系统的各种设备修复安全问题则需投入大量人力物力。受到 Android 系统的诸多特性的影响，系统版本的碎片化问题日益突出。就每一款手机而言，厂商在其维护周期内，通常会隔一段时间向用户推送一次升级版本，而用户在大多数情况下可以自主选择升级或不升级。综合这些特性，在 Android 系统的安全漏洞方面，也产生了严重的碎片化问题。

根据《2017 年度安卓系统安全性生态环境研究》报告数据，下图为各厂商手机中实际存在的安全补丁级别情况，该情况是将各厂商现存手机中实际补丁日期与谷歌官方最新版本（2017 年 12 月）版本对比，综合安全补丁级别最高、最新的手机品牌前 5 名。图中绿色方块面积越大，说明该厂商的手机补丁级别相对越高，漏洞修复相对越及时；相反，如果黄色和橙色面积越大，则说明补丁级别越低，漏洞修复越滞后。



图中我们可以看出，在及时推送安全补丁级别方面，TOP5 的厂商在本季度的检测结果显示较好，而且在本季度的调研中这五个厂商均有保持与谷歌最新安全补丁同步的更新提供，这也显示了厂商对于用户手机中安全补丁等级的逐步重视。

三、漏洞利用情况

综合上述对 Android 系统环境的介绍，我们可以看出仍然存在大量未升级至新版本系统和未打补丁的设备正在被使用，这些与安全更新脱节的现象直接导致用户手机暴露于各种漏洞的威胁之下，可造成用户的隐私、财产安全。

2017 年开始不断曝出 Android 漏洞在恶意样本上利用，下面我们以恶意样本漏洞利用的实例，来分析漏洞对 Android 用户的实际威胁：

(一) 屏幕录制漏洞

屏幕录制漏洞（CVE-2015-3878）[19]是我们在 2015 年发现并提交给 Google 安全团队，

Google 在 Android 5.0 中引入了 MediaProjection 服务，MediaProjection 服务可以让应用开发者获取屏幕内容和记录系统音频。在 Android 5.0 之前，应用开发者需要应用在 Root 权限下运行或者用设备的 Release Key 对应用进行签名，只有这样才可以使用系统保护的权限来获取屏幕内容。而且，使用 MediaProjection 服务时，不需要在 AndroidManifest.xml 中声明请求的权限。

为了使用 MediaProjection 服务，应用只需要通过 intent 请求系统服务的访问权限。对系统服务的访问是通过 System UI 的弹窗提示用户请求的应用要获取屏幕内容来授权的。攻击者可以用任意消息来覆盖 System UI 的弹窗提示，诱使用户点击并授权攻击者的应用获取屏幕内容。

2017 年 12 月被 ANVA 反病毒联盟曝出，有恶意样本使用该漏洞针对 Android 5.0-6.0 系统的手机进行屏幕截图，使用进程保护技术，窃取用户隐私。

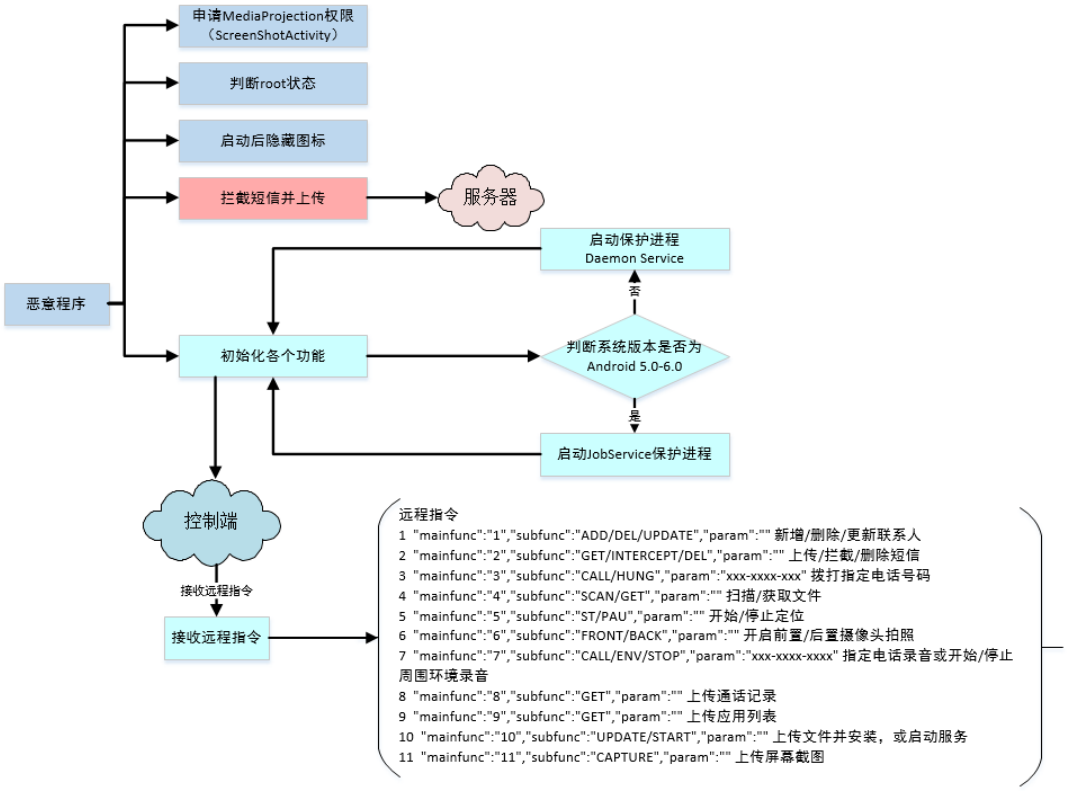


图 3.3：恶意软件执行流程[20]

(二) 脏牛漏洞

脏牛漏洞（CVE-2016-5195）[21]首次公开于 2016 年，是基于 Linux kernel 的一个严重的提权漏洞，允许攻击者获得目标系统访问的 Root 权限。

2017 年 9 月，国外安全厂商披露了第一个利用脏牛漏洞的恶意样本。而实际上，我们早在 4 月份就已经捕获到了利用该漏洞的恶意软件，脏牛漏洞最早被运用在 Dvmap 家族中，主要作用是利用脏牛漏洞提升权限并且替换系统文件。

```
v19 = sub_1A34(v3, "dst_libandroid_runtime.so", "/system/lib/libandroid_runtime.so");
v20 = v19 | sub_1A34(v3, "myip", "/system/bin/ip");
if ( !(sub_1A34(v3, "dst_build.prop", "/system/build.prop") | v20) )
{
```

```
{
  arg = v11;
  v17 = 0;
  v18 = 0;
  pthread_create(&newthread, 0, (void *(*)(void *))sub_2C28, &arg);
  pthread_create(&th, 0, (void *(*)(void *))sub_2CD0, &arg);// madviseThread
  pthread_create(&v20, 0, (void *(*)(void *))sub_2D00, &arg);// procselfmemThread
  pthread_join(newthread, 0);
  pthread_join(th, 0);
  pthread_join(v20, 0);
  close(v5);
```

图 3.4：在 Dvmap 中利用脏牛漏洞替换系统文件代码片段

(三) Toast Overlay 攻击

Overlay 攻击需要在其他运行的应用、窗口或进程上绘制和叠加 Android 视图(例如图像或者按钮)。Toast Overlay 攻击的典型场景是，欺骗用户点击攻击者指定的非法的窗口或按钮。“Toast”窗口(TYPE_TOAST)是 Android 上支持的 Overlay 类型之一，用于显示其他应用程序的通知。然而，TYPE_TOAST 类型的窗口未进行权限检查，所以不需要请求 SYSTEM_ALERT_WINDOW 权限。

TYPE_TOAST（CVE-2017-0752）[22]漏洞影响范围广，除了 Android 最新版本(8.0/Oreo)外，所有低版本的用户均受到该漏洞的影响。

```
// frameworks/base/services/core/java/com/android/server/policy/PhoneWindowManager.java
2068        String permission = null;
2069        switch (type) {
2070            case TYPE_TOAST:
2071                // XXX right now the app process has complete control over
2072                // this...  should introduce a token to let the system
2073                // monitor/control what they are doing.
2074                outAppOp[0] = AppOpsManager.OP_TOAST_WINDOW;
2075                break;
2076            case TYPE_DREAM:
...             ......
2089            case TYPE_SYSTEM_OVERLAY:
2090                permission = android.Manifest.permission.SYSTEM_ALERT_WINDOW;
2091                outAppOp[0] = AppOpsManager.OP_SYSTEM_ALERT_WINDOW;
2092                break;
2093            default:
2094                permission = android.Manifest.permission.INTERNAL_SYSTEM_WINDOW;
2095        }
2096        if (permission != null) {
...             ......
2143        }
2144        return WindowManagerGlobal.ADD_OKAY;
2145    }
```

图 3.5：TYPE_TOAST 类型的窗口未进行权限检查[23]

2017 年 11 月，国外安全厂商发现第一例使用 Toast Overlay 攻击的恶意家族 TOASTAMIGO，它利用 Android 的 Accessibility 辅助功能，使其具有广告点击、应用程序安装、自我保护/持久性功能。TOASTAMIGO 在授权后，会启动一个窗口，表明可以“分析”应用程序。而在这个窗口背后，应用程序会执行操作或指令，包括安装第二个恶意应用。

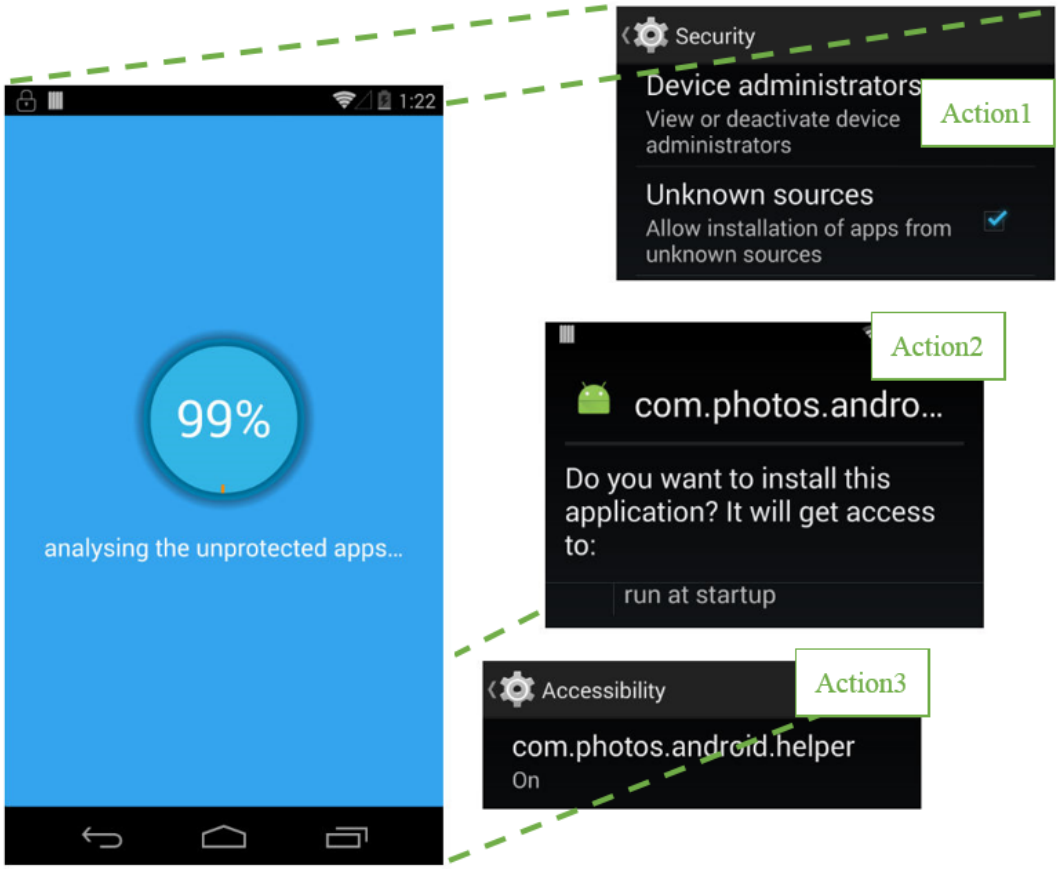


图 3.6：利用 Toast Overlay 攻击的恶意家族 TOASTAMIG 攻击原理[24]

(四) Janus 安卓签名漏洞

Android 12 月安全公告中披露了一个名为“Janus”的高危漏洞（CVE-2017-13156）[23]，攻击者可以利用该漏洞绕过 Android 系统的 Signature Scheme V1 签名机制，直接对 App 进行篡改。由于签名和验证机制是 Android 系统整体安全机制建立的最基础部分，利用该漏洞可以绕过整个 Android 系统的安全机制。基于 Signature Scheme V1 签名机制的 App 在 Android 5.1 到 8.0 系统均受“Janus”漏洞影响。

一般来说，恶意软件有两种方式来利用这个漏洞。一种是可以用来隐藏 Payload。恶意软件会把自己伪装成一个干净的 DEX 文件，而恶意 Payload 文件存储在 APK 文件中之后加载，同期，国外安全厂商捕获到利用该漏洞的恶意软件 ANDROIDOS_JANUS.A[24]就使用了这种方式；另一种是在原始开发者不知情的情况下更新已经安装的应用。攻击者可以用这种方式来访问原来的应用中受保护的数据，比如用户身份证书信息和隐私信息。冒充合法应用的身份还可以绕过杀软等安全解决方案。

第四章 技术创新夯实安全的堤防

在对抗不良应用和恶意开发者方面，根据 Google 官方数据[25]显示，2017 年 Google Play 应用商店下架了超过 70 万款违反了 Google Play 政策的应用程序，而这个数字跟 2016 年相比增长了 70%。这不仅说明 Google 移除了更多的恶意软件，而且还表明 Google 能够在恶意软件攻击阶段的早期更加准确且快速地发现它们。实际上，其中有 99%的恶意软件在用户真正安装它们之前就已经被成功识别并删除了。

Google 通过引入最新的机器学习模块和技术，显著提升了 Google Play 的安全检测能力，并能够有效发现假冒的软件、包含了违规内容的软件、以及恶意软件。

另外，Google 还研发出了新的检测模块以及检测技术来发现那些恶意开发者。在 2017 年，Google 对 10 万名恶意开发者创建新账号并发布其他软件增加了难度。

除此之外，Google 还从系统和研发两个方面提升了整体安全环境。

一、 系统更新遏制手机勒索

Google 对 Android 系统安全十分重视。从 2015 年开始提出了月度公共安全更新计划，主要是为了及时提供漏洞信息和补丁，带动各个手机厂商及时更新系统修复漏洞。同时，Android 系统版本在不断更新，每一个版本在系统安全方面都有明显的改进。

纵观 Google 在 Android 系统安全方面的更新，每一版都在遏制恶意软件方面做出了积极应对，这其中就包括勒索软件。

在早期的 Android L 版本中，获取当前运行栈顶程序 getRunningTasks 方法被废弃，阻止了了劫持 Activity 类的勒索软件。

Android M 版本中，悬浮窗权限 SYSTEM_ALERT_WINDOW 开始被列为一种危险程度较高的权限，使用时需要用户动态授权，防止用户手机在毫无防备的情况下被锁定，通过用户授权干预，能起到一定程度的缓解作用。

Android N 版本中明确规定，第三方应用开发者只能使用 resetPassword API 为无密码设备设置初始密码，而不能重置或清除已有的设备密码。Android N 中对于 resetPassword API 所添加的限制能阻止手机勒索软件对已有锁屏密码的重置，从而使得部分使用该手段的勒索软件失效。

2017 年 Google 发布了最新 Android 版本（Android 8.0/Oreo），在新版本中 Android 禁用了 5 种窗口类型。其中，3 种是勒索软件常用的系统窗口类型，从而进一步遏制手机勒索软件。

图 4.1：Android 在各个版本中遏制恶意软件的措施

二、开发规范约束进一步增强

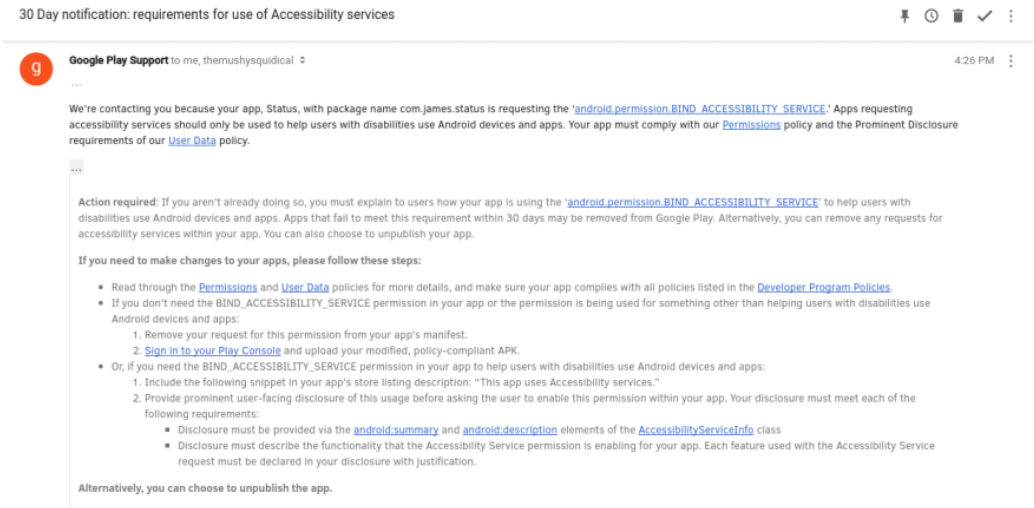
开发阶段是一个应用生命周期的启点，开发规范既能够约束开发者养成了良好的编码习惯，同时又能保障应用的代码安全。为了避免权限被滥用，Android 8.0/Oreo 加强了权限控制[26]。在 Android 8.0/Oreo 之前，如果应用在运行时请求权限并且被授予该权限，系统会错误地将属于同一权限组并且在清单中注册的其他权限也一起授予应用。

对于针对 Android 8.0 的应用，此行为已被纠正。系统只会授予应用明确请求的权限。然而，一旦用户为应用授予某个权限，则所有后续对该权限组中权限的请求都将被自动批准。例如，假设某个应用在其清单中列出 READ_EXTERNAL_STORAGE 和 WRITE_EXTERNAL_STORAGE。应用请求 READ_EXTERNAL_STORAGE，并且用户授予了该权限。如果该应用针对的是 API 级别 24 或更低级别，系统还会同时授予 WRITE_EXTERNAL_STORAGE，因为该权限也属于同一 STORAGE 权限组并且也在清单中注册过。如果该应用针对的是 Android 8.0/Oreo，则系统此时仅会授予 READ_EXTERNAL_STORAGE；不过，如果该应用后来又请求 WRITE_EXTERNAL_STORAGE，则系统会立即授予该权限，而不会提示用户。

在软件功能使用方面，Google 开始正确引导开发者使用 Android 提供的 Accessibility 服务，该服务意在帮助残疾人士更好地使用 Android 设备。我们在 2016 年发布的《ANDROID ACCESSIBILITY 安全性研究报告》报告，报告对 Accessibility 服务进行了全面的分析，对 Accessibility 的滥用情况进行了举例分析，本应该用在辅助残疾人的应用上，却被用在红包外挂、免 Root 安装、广告应用甚至是恶意应用上。

2017 年，有开发者收到了来自 Google 的邮件[27]，邮件中指出 Google 计划从 Google Play 应用商店中删除所有利用 Accessibility 服务的应用，除非使用该服务的应用目的确实是为了给有缺陷的用户提供便利功能。

虽然许多用户和开发者对 Google 的决定表示担忧，指出一些合法应用经常使用 Accessibility 服务作为功能的解决方法，否则可能难以或不可能实现。但是，从这个邮件能够看出 Google 在解决安全问题上的态度和决心。


30 Day notification: requirements for use of Accessibility services

Google Play Support to me, themushysquidical 4:26 PM

We're contacting you because your app, Status, with package name com.james.status is requesting the 'android.permission.BIND_ACCESSIBILITY_SERVICE.' Apps requesting accessibility services should only be used to help users with disabilities use Android devices and apps. Your app must comply with our Permissions policy and the Prominent Disclosure requirements of our User Data policy.

...

Action required: If you aren't already doing so, you must explain to users how your app is using the 'android.permission.BIND_ACCESSIBILITY_SERVICE' to help users with disabilities use Android devices and apps. Apps that fail to meet this requirement within 30 days may be removed from Google Play. Alternatively, you can remove any requests for accessibility services within your app. You can also choose to unpublish your app.

If you need to make changes to your apps, please follow these steps:

- Read through the Permissions and User Data policies for more details, and make sure your app complies with all policies listed in the Developer Program Policies.
- If you don't need the BIND_ACCESSIBILITY_SERVICE permission in your app or the permission is being used for something other than helping users with disabilities use Android devices and apps:
  1. Remove your request for this permission from your app's manifest.
  2. Sign in to your Play Console and upload your modified, policy-compliant APK.
- Or, if you need the BIND_ACCESSIBILITY_SERVICE permission in your app to help users with disabilities use Android devices and apps:
  1. Include the following snippet in your app's store listing description: "This app uses Accessibility services."
  2. Provide prominent user-facing disclosure of this usage before asking the user to enable this permission within your app. Your disclosure must meet each of the following requirements:
     - Disclosure must be provided via the android:summary and android:description elements of the AccessibilityServiceInfo class
     - Disclosure must describe the functionality that the Accessibility Service permission is enabling for your app. Each feature used with the Accessibility Service request must be declared in your disclosure with justification.

Alternatively, you can choose to unpublish the app.

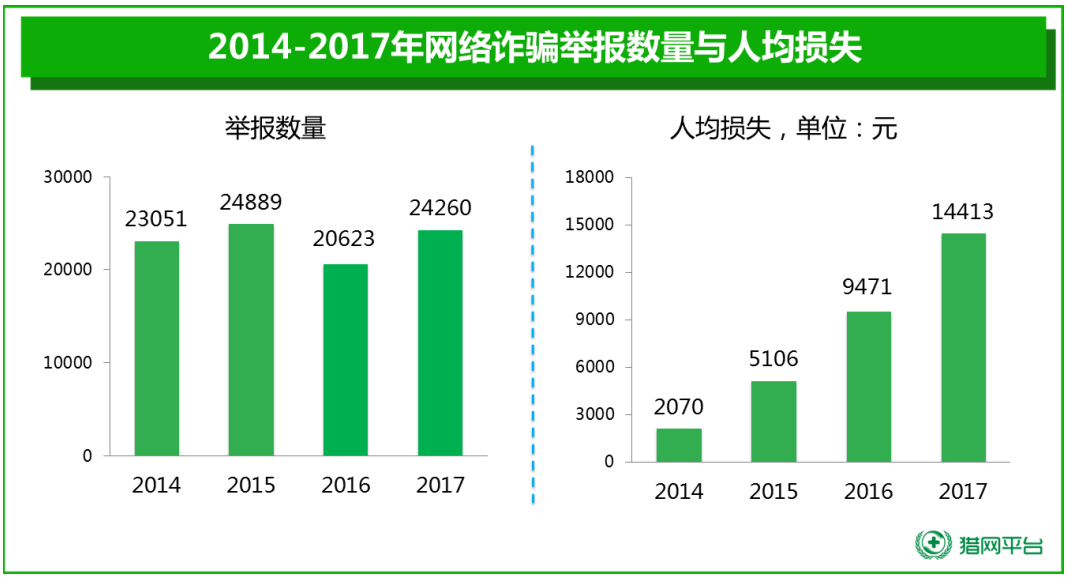
图 4.2：Google 通知开发者关于约束 Accessibility 服务功能的邮件

第五章 万物皆变人是安全的尺度

今年中国互联网安全大会（ISC）主旨是万物皆变人是安全的尺度，谈到关于人的重要性，即人是网络安全中的软肋，也是安全中的关键一环。值得注意的是，在注重培养安全人才的同时也要注重整个社会安全意识的提高。以当前泛滥的网络诈骗为例，大多数网络诈骗的发生都是骗子利用受害者安全意识低，防范能力不足而实施的。数据显示，绝大多数网络诈骗并不是通过高超的技术盗取受害者钱财，而依旧是靠“忽悠”。

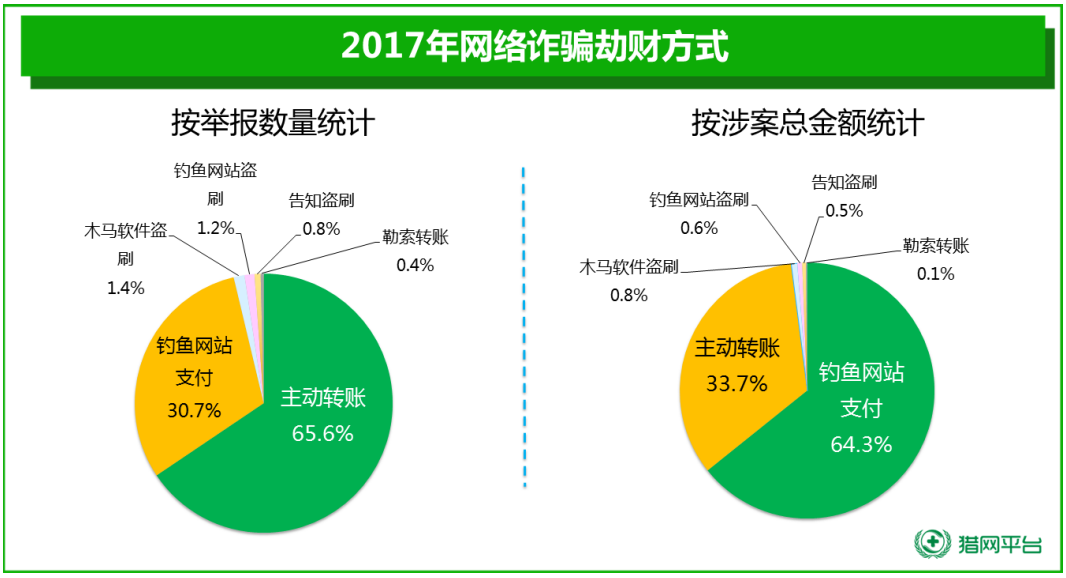
一、 网络诈骗综述

2017 年，由北京市公安局联合 360 共同推出的，通过互联网平台和互联网技术打击网络诈骗犯罪的创新平台——猎网平台共收到全国用户提交的有效网络诈骗举报 24260 例，举报总金额 3.50 亿余元，人均损失 14413.4 元。与 2016 年相比，网络诈骗的举报数量增长了 17.6%，人均损失却增长了 52.2%。2014-2017 年统计以来，每年的人均损失均出现大幅增长，可见网络诈骗正在严重的威胁着网民的财产安全。



在猎网平台 2017 接到的用户举报中，有 15911 人是通过银行转账、第三方支付、扫二维码支付等方式主动给不法分子转账，占比 65.6%，其次有 7442 人在虚假的钓鱼网站上支付，占比 30.7%；安装木马软件从而被盗刷的用户有 328 人，占比 1.4%；在钓鱼网站上填写用户的账号、密码等隐私信息后，被盗刷的用户有 302 人，占比 1.2%；主动告知账号密码/二维码/验证码/付款码盗刷从而被盗刷的有 185 人，占比 0.8%，还有 88 人遭到勒索软件、恐吓电话等被勒索盗刷，占比 0.4%。

如果从涉案总金额来看，钓鱼网站支付，占比 64.3%，累计 2.2 亿元；其次受害者主动转账占比 33.7%，累计 1.2 亿元；木马软件导致盗刷占比 0.8%，累计 279.7 万元；钓鱼网站导致盗刷占比 0.6%，累计 215.6 万元；主动告知账号密码/二维码/验证码/付款码从而被盗刷占比 0.5%，累计 167.8 万元。



二、 警企协同打击网络犯罪

截至 2017 年底，猎网平台已与全国 300 个地区的公安机关建立联系。全年协助各地区公安机关协查案件 219 起，破案 23 起，抓获嫌疑人共计 137 人。

同时，360 烽火实验室对外开展移动平台电子取证培训 10 余次，涵盖河南、长春、浙江等全国多地。在涉及移动平台的网络犯罪案件侦办中，实验室通过溯源等分析手段，协助公安机关找到恶意软件作者 QQ 号，手机号及邮箱线索 31 万余条，案件涉及的恶意软件分析报告 18 篇。

下面列举我们协助警方破获的重点案件：

案例一

今年 2 月 3 日，营口市开发区公安局接到孟女士报警称，其农业银行卡内 50 余万元人民币被盗。警方发现，1 月 20 日，孟女士收到含有木马链接的短信，不慎点击后手机中招，短信被拦截，11 天内其银行卡内的 50 余万被犯罪分子分 800 余笔交易到其他消费网站。

经初步调查，犯罪分子事先在网上购买大量身份证、银行卡等信息，通过群发短信的方式将木马病毒植入受害人银行卡绑定的手机。该木马病毒可以拦截手机短信、提取手机通讯录、获取网上银行的手机验证码，骗子借此盗刷受害人银行卡内资金。通过资金流追查，警方从第三方支付平台为孟女士追回被骗资金 2700 余元。

借助猎网平台，通过溯源分析，找到木马劫持受害人短信所使用的邮箱，帮助警方成功锁定了嫌疑人的线索。

经过两个多月缜密细致的调查取证，营口警方发现确认这是一伙分工明确、链条完整的诈骗团伙，分布在广西、北京、福建三地。其中，广西宾阳嫌疑人负责操作木马，套取被害人银行卡消费的验证码；北京窝点负责被害人的钱以话费形式转移至手机充值卡，然后再将话费转卖，回款后与宾阳嫌疑人三七分成；福建窝点将被害人的钱充值到苹果云账户，购买苹果商品消费，再低价转卖。

4 月 13 日和 20 日，警方分赴北京、福建、广西，抓获刘某、梁某、黄某、黄某某等四

名嫌疑人，缴获作案电脑 5 台、机箱 3 台、平板电脑 2 台、手机 21 部、银行卡 23 张。

案例二

今年年初，淮北市民刘先生突然收到短信，写道“海峰，你看这个女的面熟吗”，文字后带有一个链接。看到是熟人发来的信息，刘先生没多想就点击了链接，进入后发现是英文看不懂，刘先生就随手删除了短信。

次日一早，刘先生突然连续收到短信提示，银行卡先后两次被消费 1960 元。密码从未告诉过他人，怎么会被消费？担心卡内 100 多万资金的安全，刘先生赶紧到银行求助，将卡内余额转存，随后向警方报案。

淮北公安通过检测刘先生的手机发现，他收到的实为一条诈骗短信，其点击短信链接后手机自动安装了木马程序，通讯录、短信等信息都被犯罪分子拦截盗取。根据盗取到的短信验证码等信息，犯罪分子盗刷了刘先生手机绑定的银行卡内的钱。同时，刘先生点击链接后，类似短信还被自动发送给了他的多名通讯录好友。

在 360 烽火实验室技术支持协助下，关联出同源代码木马 199 个，这 199 个木马共绑定了用于接收受害人短信的邮箱 2 个、手机号码 7 个。通过淮北警方对木马设置的邮箱和手机号进一步调查，成功摧毁了木马制作、传播、盗刷、销赃整个犯罪团伙，跨越四省抓获 9 名犯罪嫌疑人。据初步调查,短短半年时间，全国近 10 万部手机被木马感染，近 800 人遭遇经济损失，团伙诈骗金额超过 155 万。

第六章 威胁趋势预测

一、具备自动化和对抗能力的恶意软件工厂不断涌现

2017 年，BlackHat 的议题《AVPASS: Automatically Bypassing Android Malware Detection System》[28]引发业内关注，这是首次在国际黑客会议上公开的，关于自动化探测安全软件规则工具的集合介绍。

AVPASS 是一个可以探测 Android 杀毒软件的检测模型，并结合探测到的信息和混淆技术构造特定 APK 来绕过杀毒软件检测的工具。AVPASS 不仅可以推测出杀毒软件使用的特征，而且可以推导出其检测规则，因此，（理论上）它可以自动的变形 APK，使得任何杀软将一个恶意 APP 误认为一个正常 APP，即免杀。

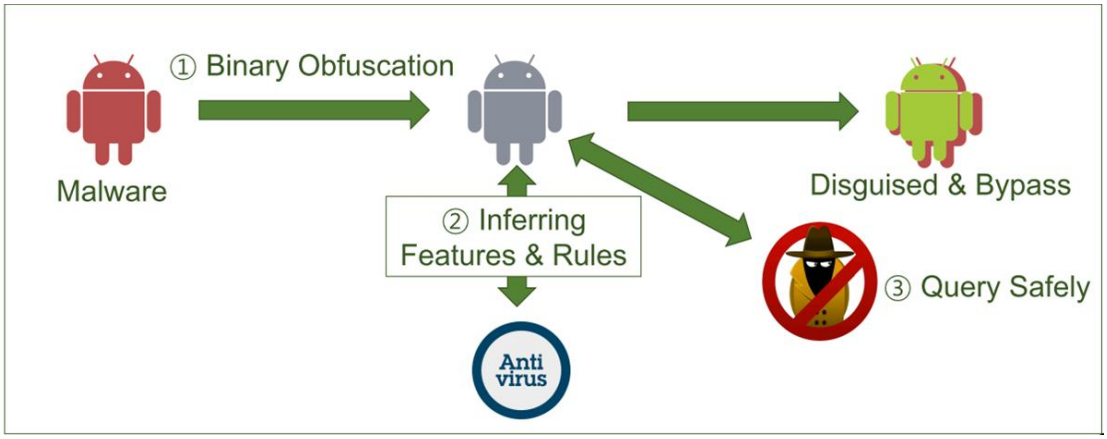


图 6.1：AVPASS 检测模型

AVPASS 实现了自动化免杀，这种技术已经衍生出多种类型软件的一键生成器，比如广告软件和勒索软件。

图 6.2：恶意广告（左）和勒索软件（右）一键生成器

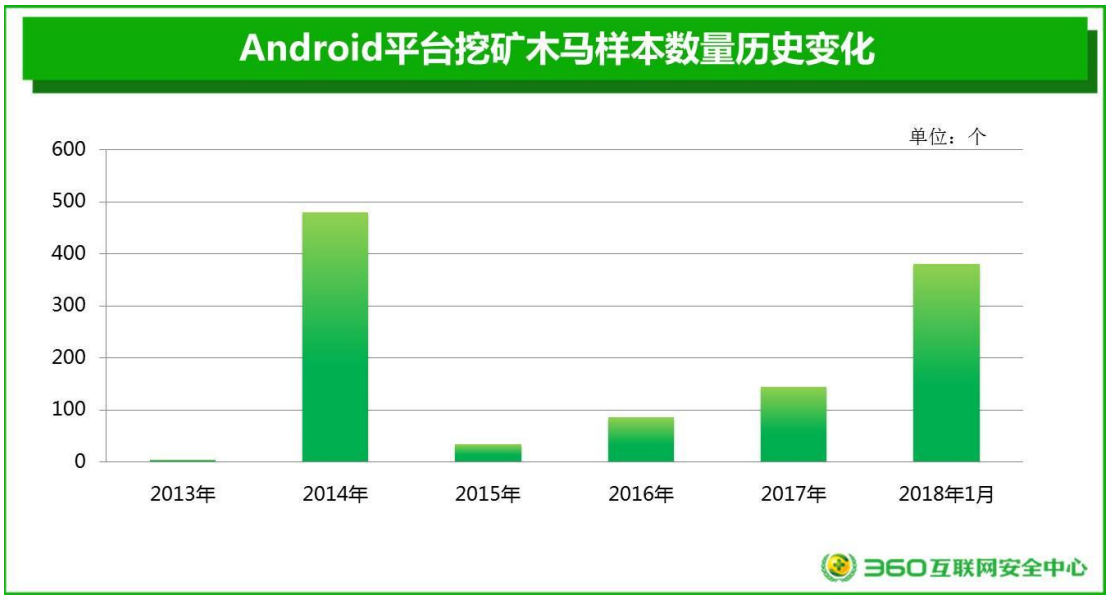
在 2016 年我们发布的《ANDROID 逃逸技术汇编》报告，报告汇总了 60 多种恶意软件对抗检测的手段。由此可见，恶意软件与检测手段的对抗从未停止，恶意软件工厂取代了人工繁琐的代码编写、编译过程，进一步降低了制作门槛，不仅生成速度变快，并且能够根据需要进行定制，这种恶意软件生产模式必将成为未来的发展趋势之一。

二、 恶意挖矿木马愈演愈烈

2017 年最疯狂的莫属电子货币，以比特币为代表的电子货币，年内单价突破了 2 万美元，随着电子货币价格暴涨，针对电子货币相关的攻击事件也越来越频繁。

今年 5 月，以比特币为勒索目标的 WannaCry[29]勒索病毒全球大爆发，至少 150 个国家、30 万名用户中招，造成损失达 80 亿美元，已经影响到金融，能源，医疗等众多行业，造成严重的危机管理问题。中国部分 Windows 操作系统用户遭受感染，校园网用户首当其冲，受害严重，大量实验室数据和毕业设计被锁定加密。部分大型企业的应用系统和数据库文件被加密后，无法正常工作，影响巨大。

而在移动平台上，从 2013 年开始至 2018 年 1 月，360 烽火实验室共捕获 Android 平台挖矿木马 1200 余个，其中仅 2018 年 1 月 Android 平台挖矿木马接近 400 个，占全部 Android 平台挖矿类木马近三分之一。



相比 PC 平台，移动终端设备普及率高，携带方便，更替性强，因而安全问题的影响速度更快，传播更广。然而，移动平台在挖矿能力上受限于电池容量和处理器能力，并且在挖矿过程中会导致设备卡顿、发热、电池寿命骤降，甚至出现手机物理损坏问题，就目前来看移动平台还不是一个可持续性生产电子货币的平台。

今年曝光的 Android 挖矿木马事件，初步显示应用盈利模式由广告转向挖矿，门罗币成为挖矿币种首选以及攻击目标向电子货币钱包转移成为 Android 平台挖矿木马的演变的三大趋势。挖矿和勒索成为 2017 年两大全球性的安全话题，2018 年将会继续延续。

三、 公共基础服务成为恶意软件利用的新平台

今年恶意软件使用的新技术中，出现了利用 Telegram 软件协议和 VA 应用多开技术，在早期还出现过恶意软件还使用了 Google 的 GCM 推送服务以及第三方消息推送服务，这些公共基础服务、开源代码被恶意利用，成为恶意软件的新平台。

以使用 Telegram 软件协议服务和 Google 的 GCM 服务为例，需要恶意作者申请自己唯一的 ID，进行指令控制时仅通过 ID 来区分，访问的都是服务提供商的服务器，并不是恶意软件作者自己的服务器，从而避免了通过 C&C 追踪溯源。

而在使用 VA 应用多开技术时，也可以实现对公共基础服务的恶意利用。恶意软件不但在 Manifest 中声明的权限和组件结构、数量上基本相似，又常以子包形式加密存储在 VA 内，主包代码特征也表现一致，在检测引擎扫描时使用 VA 的样本引擎无法识别，从而绕过杀软静态检测。

综合来看，公共基础服务被恶意利用，一方面，是由于这些基础服务提供了简便易用的接入方法，使得恶意软件方便快速的接入；另一方面，恶意软件利用这些基础服务，能够实现绕过检测、躲避跟踪溯源。

四、 脚本语言成为恶意软件新的技术热点

脚本语言具有跨平台、配置灵活简单特性，方便开发者增强应用体验效果。但同时，也为恶意软件躲避查杀提供了便利。2017 年我们发现了三种利用 JavaScript 脚本语言实施恶意行为的隐蔽方式。

第一种是利用 JavaScript 模拟点击刷网页、广告流量。我们在《移动平台流量黑产研究——流量作弊与流量泡沫》报告中，以某个渠道近一周的数据为例进行抽样分析，恶意软件每 30 秒完成一次刷量行为，据此每个受害用户的手机每天能产生 2880 次虚假访问，最近一周产生的虚假访问总数为 1.8 亿次。按照每千次 10 元到 20 元的收费标准，最近一周能为该渠道创造 185 万元的收入。

```
function exeSearchAndClick(type,rate){
function exeSearch(type,content,rate){
function getRandomClickPosition(fromx,tox,fromy,toy){

function clickJsAd(){
function jsAdCallBack(){
function searchBaidu(){
function searchSogou(){
```

图 6.3：某恶意软件使用的 JavaScript 模拟点击脚本

第二种是利用 JavaScript 私自发送短信，我们在《Expensivewall 家族变种再现 Google Play》报告中发现 Google Play 上存在 Expensivewall 恶意家族变种。该家族运行原理是先根

据 getSimCountryIso 获取到的国别码获取不同的广告 URL；其次，通过 webview 方法加载的含有 JavaScript 脚本的广告页面；最后，在打开的广告展示页面上，点击关闭按钮×时触发发短信操作。

```
    <script>
        function toContinue() {
            var operatorCode = pagejs.getOperatorCode()+"";
            if(operatorCode.trim() == "52018" || operatorCode.trim() == "52005" ) { //DTAC  52018|52005
                pagejs.sendSMS('B10', '4739998');
            }else{
                pagejs.sendSMS('B3','4139807');
            }
        }
    </script>
</head>
<body>
<header><img class="shut" src="http://wap.yasune360.com/lp_au/image/shut.png" alt="" onclick="toContinue()"></header>
```

图 6.4：Expensivewall 使用 JavaScript 脚本发送短信

第三种是利用 JavaScript 进行挖矿，Coinhive 提供的灵活方便的 JavaScript API，在网页中调用 Coinhive 官网中的 JavaScript 文件 coinhive.min.js 并指定一个唯一的标识符即可。

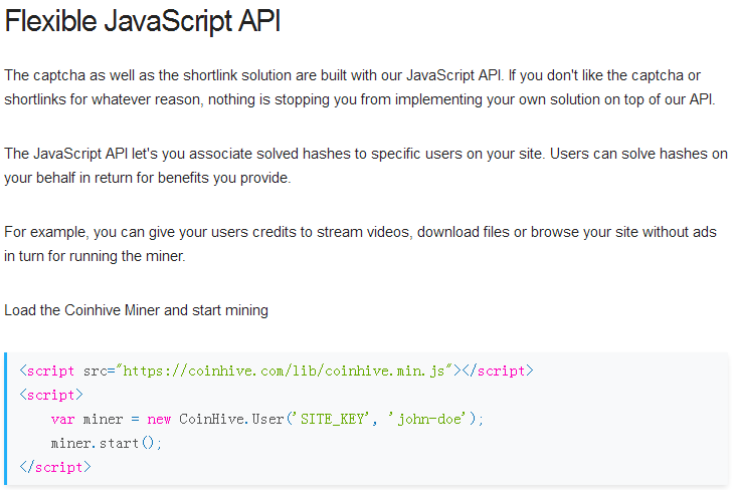


图 6.5：Coinhive 提供的 JavaScript API[30]

以上这三种隐蔽方式，展现了一种新的恶意软件发展的新趋势。由于这种方式不依赖与核心代码，在软件动态运行时才会触发调用，给安全软件检测提出了更高的要求和挑战。未来在脚本语言的检测防御上，各个安全厂商应该提高重视程度，提供更加全面的安全防护模式。

附录一：参考资料

[1] Dvmap: the first Android malware with code injection：
https://securelist.com/dvmap-the-first-android-malware-with-code-injection/78648/

[2] Doctor Web: Android Trojan controlled via Telegram spies on Iranian users：
https://news.drweb.com/show/?i=11331&lng=en

[3] CoinKrypt: How criminals use your phone to mine digital currency：
https://blog.lookout.com/coinkrypt

[4] The WireX Botnet: How Industry Collaboration Disrupted a DDoS Attack：
https://blog.cloudflare.com/the-wirex-botnet/

[5] 内网穿透——ANDROID 木马进入高级攻击阶段：
http://blogs.360.cn/360mobile/2016/12/01/analysis_of_dresscode/

[6] 内网穿透——ANDROID 木马进入高级攻击阶段（二）：
http://blogs.360.cn/360mobile/2017/05/25/analysis_of_milkydoor/

[7] Detecting and eliminating Chamois, a fraud botnet on Android：
https://security.googleblog.com/2017/03/detecting-and-eliminating-chamois-fraud.html

[8] The Bearer of BadNews：https://blog.lookout.com/the-bearer-of-badnews

[9] VirtualApp：https://github.com/asLody/VirtualApp

[10] Malware posing as dual instance app steals users' Twitter credentials：
https://blog.avast.com/malware-posing-as-dual-instance-app-steals-users-twitter-credentials

[11] 双尾蝎组织（APT-C-23）伸向巴以两国的毒针：
http://zt.360.cn/1101061855.php?dtid=1101062514&did=490322462

[12] Targeted Attacks in the Middle East Using KASPERAGENT and MICROPSIA：
https://researchcenter.paloaltonetworks.com/2017/04/unit42-targeted-attacks-middle-east-using-kasperagent-micropsia/

[13] FrozenCell: Multi-platform surveillance campaign against Palestinians：
https://blog.lookout.com/frozencell-mobile-threat

[14] New GnatSpy Mobile Malware Family Discovered：
http://blog.trendmicro.com/trendlabs-security-intelligence/new-gnatspy-mobile-malware-family-discovered/

[15] Pegasus for Android: the other side of the story emerges：
https://blog.lookout.com/pegasus-android

[16] Mobile Advanced Persistent Threat actor conducting global espionage campaign from Lebanon：https://blog.lookout.com/dark-caracal-mobile-apt

[17] Top 50 Products By Total Number Of "Distinct" Vulnerabilities in 2017：
https://www.cvedetails.com/top-50-products.php?year=2017

[18]Android 平台版本：https://developer.android.com/about/dashboards/index.html

[19] CVE-2015-3878：https://source.android.com/security/bulletin/2015-10-01

[20] 利用 CVE-2015-3878 的安卓间谍软件信息共享：
https://mp.weixin.qq.com/s?__biz=MzAxNjMzOTY1OA==&mid=2655229765&idx=1&sn=4413d2d58573c8274c79ac7c884a0e9a&chksm=804170e0b736f9f6d9f3639047016ddc5b83ff06f7723f55257a87800332061bbe545b3fa68a&mpshare=1&scene=1&srcid=1207GsaNCJsAU7r9o5zmbLy3&pass_ticket=7cAS2PAVyXh3tk6OiVp8TbOmX3VRCO9qfULmiQDSaJ0%3D#rd

[21] CVE-2016-5195：https://source.android.google.cn/security/bulletin/2016-11-01

[22] CVE-2017-0752：https://source.android.com/security/bulletin/2017-09-01

[23] Android Toast Overlay Attack: “Cloak and Dagger” with No Permissions：
https://researchcenter.paloaltonetworks.com/2017/09/unit42-android-toast-overlay-attack-cloak-and-dagger-with-no-permissions/

[24] Toast Overlay Weaponized to Install Several Android Malware：
https://blog.trendmicro.com/trendlabs-security-intelligence/toast-overlay-weaponized-install-android-malware-single-attack-chain/

[23] CVE-2017-13156：https://source.android.google.cn/security/bulletin/2017-12-01

[24] Janus Android App Signature Bypass Allows Attackers to Modify Legitimate Apps：
http://blog.trendmicro.com/trendlabs-security-intelligence/janus-android-app-signature-bypass-allows-attackers-modify-legitimate-apps/

[25] How we fought bad apps and malicious developers in 2017：
https://android-developers.googleblog.com/2018/01/how-we-fought-bad-apps-and-malicious.html

[26] Android 8.0 Behavior Changes：
https://developer.android.com/about/versions/oreo/android-8.0-changes.html

[27] Google to Ban Android Apps Misusing Accessibility Service：
https://www.securityweek.com/google-ban-android-apps-misusing-accessibility-service

[28] AVPASS: LEAKING AND BYPASSING ANTIVIRUS DETECTION MODEL AUTOMATICALLY：
https://www.blackhat.com/us-17/briefings/schedule/#avpass-leaking-and-bypassing-antivirus-detection-model-automatically-7354

[29] 勒索病毒 WannaCry：https://baike.so.com/doc/1727832-27028923.html

[30] Coinhive – Monero JavaScript Mining: https://coinhive.com/

360 烽火实验室

360 烽火实验室，致力于 Android 病毒分析、移动黑产研究、移动威胁预警以及 Android 漏洞挖掘等移动安全领域及 Android 安全生态的深度研究。作为全球顶级移动安全生态研究实验室，360 烽火实验室在全球范围内首发了多篇具备国际影响力的 Android 木马分析报告和 Android 木马黑色产业链研究报告。实验室在为 360 手机卫士、360 手机急救箱、360 手机助手等提供核心安全数据和顽固木马清除解决方案的同时，也为上百家国内外厂商、应用商店等合作伙伴提供了移动应用安全检测服务，全方位守护移动安全。