基于恶意代码传播日志的网络安全态势分析

王琴琴[1,2], 周 昊[3], 严寒冰[1,3], 梅 瑞[1,2], 韩志辉[3]

[1] 中国科学院信息工程研究所第二研究室 北京 中国 100093
[2] 中国科学院大学网络空间安全学院 北京 中国 100049
[3] 国家计算机网络应急技术处理协调中心 北京 中国 100029

摘要 网络安全态势一直是网络安全从业人员的关注点。本文基于 2018 年 10 月至 2019 年 3 月的我国恶意代码的传播日志, 利用恶意代码的静态特征、动态特征及其传播特征对网络态势进行分析。然后基于社区发现算法, 对其中传播最广泛的 Mirai 家族程序构成的网络进行团伙发现, 结果表明, 社区发现算法能够将 Mirai 网络识别为多个社区, 社区间的域名资源具有明显的差异性, 社区内域名资源具有相似性。

关键词 网络安全态势; 恶意代码传播; Mirai; 社区发现算法
中图法分类号 TP309.5 DOI 号 10.19363/J.cnki.cn10-1380/tn.2019.09.02

Cyber Security Posture Analysis based on Spread Logs of Malware

WANG Qinqin[1,2], ZHOU Hao[3], YAN Hanbing[1,3], MEI Rui[1,2], HAN Zhihui[3]

[1] The 2nd Laboratory, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3]National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), Beijing 100029, China

Abstract The cyber security posture has always been the focus of network security practitioners. This paper collects spread logs of malware in China from October 2018 to March 2019, and then analyzes cyber security posture from the static and dynamic characteristics of malicious files, as well as the propagation characteristics. Moreover, based on the community discovery algorithm, the paper makes a gang discovery on the network composed of the most widely spread Mirai family programs. The results show that the community discovery algorithm can identify the Mirai network as multiple communities. Domain names between communities have significant differences, and domain names within the same community have similarities.

Key words cyber security posture; spread logs of malware; Mirai; community discovery algorithm

1 引言

2018 年, CNCERT 全年捕获计算机恶意程序样本数量超过 1 亿个, 涉及计算机恶意程序家族 51 万余个, 较 2017 年增加 8132 个。全年计算机恶意程序传播次数日均达 500 万余次[1]。

上文 CNCERT 的报告附录中, 从恶意程序、安全漏洞、拒绝服务攻击、网络安全、工业互联网安全、互联网金融安全等方面描述了 2018 年的网络安全概况, 这与本文所做的工作有相似之处。报告是对网络安全的全局概览, 本文从传播的恶意代码着手, 去探索网络安全状况的具体细节, 并提出了基于社区发现算法的团伙发现方法。我们基于 CNCERT 的恶意代码数据, 对数据中的程序做恶意性检测, 从恶意代码传播来了解网络状况, 以多个维度特征来描述网络态势, 利用社区发现算法来探索 Mirai 网络的团伙发现。

随着网络和通信技术的发展, 人们越来越关注网络安全。网络安全通常由网络系统生成、发起或者提取的相关数据反映出来[2]。安全相关数据具有 5V(大量(Volume)、多样(Variety)、低价值密度(Value)、高速(Velocity)、真实(Veracity))特征, 这在数据采

通讯作者: 严寒冰, 博士, 正高级工程师, Email: yhb@cert.org.cn。
本课题得到国家自然科学基金重点项目（No. U1736218）和科技部重大专项（No. 2018YFB0804704）资助。
收稿日期: 2019-05-21; 修改日期: 2019-08-13; 定稿日期: 2019-08-20

集[5-10,25]方面造成巨大挑战[4]。网络安全数据[11-17]多样体现在用来反映网络安全的数据范围是多样的, 包括完整内容数据、会话数据、元数据、日志数据、告警数据等。

本文是从传播的恶意代码来对网络安全态势进行分析, 与本文具有相同意义的文章很多, 例如网络测量工作、网络态势感知等相关工作, 这些文章都是通过数据来对网络安全进行刻画和描述。Cozzi等人[18]对Linux系统下的恶意程序进行测量和分析, 去探索 Linux 恶意程序的偏好和特点。本文受这篇文章启发实现了对传播恶意代码的多维特征测量和分析。

由于网络数据规模很庞大, 如何高效地挖掘复杂网络中有意义的信息, 是本文研究的另一个重点。复杂网络社区发现方法的研究对分析复杂网络的拓扑结构和层次结构, 理解社区的形成过程, 预测复杂网络的动态变化, 发现复杂网络中蕴含的规律特征具有十分重要的意义。已有一些文章将社区发现应用于互联网网络分析与聚类, 以及社会组织结构网络分析[19]中。周瑞等人[20]将社区发现应用到 Web 网络, 探索 Web 网络结构、功能, 发现规律并预测行为。朱天等人[3]发明一种基于网络攻击伴随行为的DDoS 攻击群体分析方法。该方法从互联网攻击资源出发, 来发现 DDoS 攻击群体。因此本文将互联网攻击资源作为团伙发现的主要依据。我们将社区发现算法应用于团伙发现, 恶意代码的传播可视作一个复杂网络, 将传播属性看作网络结点, 传播记录之间的资源共享关系看作网络的边, 一个团伙是指资源共享的一组结点, 这和社区的意义是类似的。

本文使用社区发现算法来探索恶意代码的社区特点。社区发现算法是将网络划分为多个社区的算法, 根据网络划分依据, 大概分为四类, 基于标签传播的社区发现, 基于信息论的社区发现, 基于谱分析的社区发现, 以及基于模块度优化的社区发现。基于标签传播的社区发现算法[21]目标是使得每个结点与它的大多数邻居在同一社区中, 算法简单, 计算时间为线性时间, 但是算法结果不稳定。基于信息论的社区发现算法[22]是从信息论的角度出发, 例如将社区发现问题转换为信息论中的一个基础问题, 寻找拓扑结构的有效压缩方式, 或者将问题转换为寻找描述网络上随机游走的有效编码方式。基于谱分析的社区发现算法[23]建立在谱图理论的基础上, 根据特定图矩阵的特征向量导出对象的特征, 利用导出特征来推断对象之间的结构关系, 该算法开销很大。基于模块度优化的社区发现算法[24]是研究最多的算法, 将社区发现问题定义为优化问题, 然后搜索目标值最优的社区结构, 算法结果稳定。本文选用了模块度优化算法中的 Fast Newman 算法, 也称作 Fast Greedy 算法来探索网络结构。

本文研究目的是探索网络安全态势, 网络安全态势需要考虑各方面的安全因素, 从整体上反应安全状况。我们通过对网络上传播的恶意代码进行研究, 以探索网络安全态势。本文主要的贡献包括: 1、通过对传播恶意代码的多个维度分析, 将网络安全状况进行了全局的展示, 揭露出网络面临的主要威胁、恶意代码的流行趋势、恶意代码的传播特点等等; 2、针对影响网络最广泛的 Mirai 恶意程序, 我们进行了 Mirai 网络结构的探索, 在这个过程中应用社区发现算法进行 Mirai 攻击团伙的发现, 这是对网络主流威胁的理解和分析。

首先记录恶意代码的传播日志作为后续研究数据集。网络安全态势从恶意代码的多维特征来描述, 大体上分为恶意代码的静态特征和动态特征, 以及恶意代码的传播特征。其中恶意代码的静态特征和动态特征具体从恶意代码的最早出现时间、文件类型、加壳信息、恶意代码家族、API、漏洞利用等方面进行统计分析。恶意代码传播特征从(1)恶意代码传播量统计; (2)恶意代码感染源地区分布, 以及面临感染风险的地区分布; (3)域名和恶意代码的承载关系, 三个方面进行统计分析。数据集中传播最广泛的恶意代码为 Mirai 家族代码, 利用社区发现算法应用于 Mirai 网络的传播中, 去研究攻击团伙的特点。

文章的结构如下, 第 1 章为引言; 第 2 章是数据获取和处理, 描述了本文的数据集, 即恶意代码传播日志的获取和处理工作; 第 3 章是恶意代码多维特征分析, 从恶意代码的静态特征和动态行为, 以及传播特征来对网络态势做出分析; 第 4 章是基于社区发现的攻击团伙发现, 将社区发现算法应用于 Mirai 网络, 发现 Mirai 网络中攻击者的偏好; 第 5 章是结论, 对此次研究做出了总结。

2 数据获取与处理

本文数据集是 2018 年 10 月至 2019 年 3 月的恶意代码传播日志。恶意代码是指 EXE 文件、DLL 文件、PDF 文件、Microsoft Office 文件、URL 和 HTML 文件、CPL 文件、Visual Basic (VB) 脚本文件、ZIP 压缩文件、Java JAR、Python 文件、APK 文件、ELF 文件等文件格式, 通过网络或其他设备散播的, 故意对个人计算机、服务器、智能设备、计算机网络等造成隐私或机密数据外泄、系统损害、数据丢失

等非使用预期故障及信息安全问题, 并试图以各种方式阻挡用户移除它们。传播日志是指代码通过 HTTP 协议进行网络传播的记录。

数据集的准备工作由四个模块组成, 数据获取, 数据存储, 数据清洗和融合, 数据选择。

数据获取模块, 本文所使用数据集来源于国家计算机网络应急技术处理协调中心(CNCERT/CC)提供的测试数据集。

数据存储模块, 将获取的大规模样本数据利用 Hadoop[26]和 Spark[27]平台进行存储。

数据清洗和融合模块, 我们将数据进行格式化, 存储传播数据, 包含传播记录(下载源 IP, 受攻击 IP, 域名, md5, 时间), 以及对应的传播载荷文件。其中域名清洗规则是提取主域名, 例如 HTTP 请求域名为 http://xxx/yyy/zzz, 提取主域名 xxx。md5 通过 HTTP 载荷数据的 MD5 获得。下载源 IP、受攻击 IP、时间均从 HTTP 请求中获取。其次, 我们对每天的传播记录进行融合处理, 将每天的传播记录进行融合, 例如某传播记录组下载源 IP、受攻击 IP、域名、md5 在某一天多次出现, 我们将其进行融合为一条记录, 并添加字段 count 来记录次数, 其中时间字段精确到天。

数据选择模块, 我们使用恶意代码判定系统对传播载荷进行恶意性检测, 并进行标记, 恶意代码是本文研究主要关注的对象。恶意代码判定系统提供病毒、蠕虫、木马和各种恶意软件分析服务, 可以对传播代码进行快速检测。

最后我们将恶意代码的传播记录添加下载源 IP 地区, 受攻击 IP 地区两个字段, 分别对下载源 IP 地址和受攻击 IP 地址进行地区查询, 并更新传播记录。

最终获取数据集为恶意代码的传播记录, 其内容包含恶意代码的 md5、下载源 IP、受攻击 IP、下载源 IP 地区、受攻击 IP 地区、HTTP 请求的域名、时间、次数等。数据集中的数据是数据融合之后的结果, 共 6819561 条记录, 其中包含数据融合前的恶意代码传播次数为 921940104 次, 传播的恶意代码约 40000 个。

3 恶意代码多维特征分析

本节将对数据集中的恶意代码进行多维特征分析。这项工作对 2018 年 10 月至 2019 年 3 月传播的恶意代码进行概览, 使得我们对这半年时间内网络安全状况有全局的了解, 也有利于后续的研究。分析工作主要从两方面进行: (1)恶意代码的静态特征和动态特征分析; (2)恶意代码的传播特征分析。

3.1 静态特征和动态特征

本节将传播数据集中约 40000 个恶意代码进行静态特征和动态特征分析。我们使用 python 编写程序, 利用恶意代码判定系统来获取恶意代码的检测报告, 检测报告包含恶意代码的静态信息和动态行为信息。本节根据返回的报告内容, 选取恶意代码的静态特征, 包括最早出现时间、文件类型、加壳信息, 选取动态特征, 包括恶意代码家族、API、漏洞利用, 从静态和动态的多个特征维度进行分析。

最早出现时间。为了更好地了解网络中传播的恶意代码, 首先我们对这些代码的最早出现时间进行统计, 这个时间为代码最早出现在网络中的时间。如图 1 所示, 纵轴表示代码最早出现的年份, 横轴表示代码数目的比例。可见代码出现范围在 2006 年至 2019 年, 其中 58.6%恶意代码最早出现在 2018 年和 2019 年, 97.55%恶意代码最早出现在 2012 年至 2019 年。

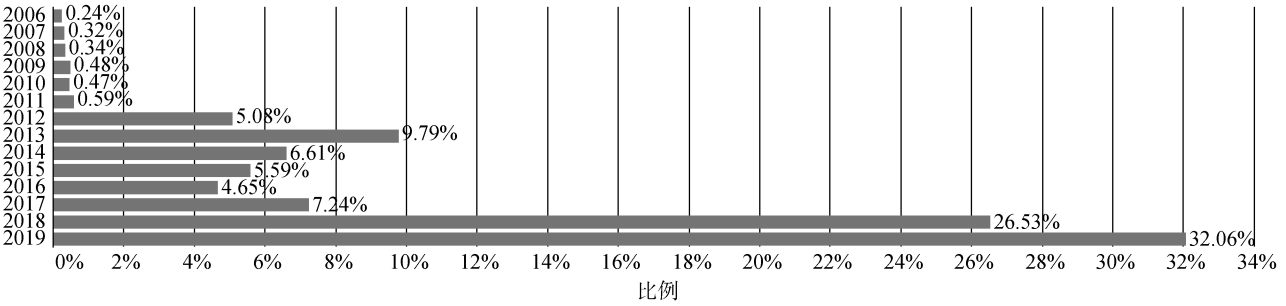


图 1 网络中最早出现时间统计

Figure 1 First appearance statistics

文件类型。将恶意代码的文件类型进行统计。如表 1 所示, 传播的恶意代码文件类型占比最大的是 ELF, 其次是 ZIP、Win32 EXE 、HTML, 这四种类型共占比约 90%。除此之外, Android 恶意代码也是最值得关注的。

表 1 文件类型统计

Table 1 File type statistics

文件类型	所占比例
ELF	26.4%
ZIP	25.1%
Win32 EXE	21.7%
HTML	16.5%
Android	4.5%
MS Word Document	2.3%
Win32 DLL	1.3%
MS Excel Spreadsheet	0.7%
JAR	0.4%
Text	0.3%
Rich Text Format	0.2%
Flash	0.2%

加壳信息。为了绕过静态和动态分析, 恶意代码的编写者经常为恶意代码采取加密或者加壳的措施。本节我们将恶意代码的加壳信息进行统计[28]。如表 2 所示, 在此次研究的恶意代码中加壳的代码占比 19.5%。加壳工具多样化, 且易于使用, 也是编写者们采用加壳的原因之一。接下来我们对代码的加壳工具进行统计, 如图 2 所示, 纵轴表示加壳工具, 横轴表示工具使用比例。其中 UPX[29]以绝对的优势位列第一, 是最受欢迎的加壳工具, 其免费开源, 且支持许多不同操作系统下的可执行文件格式, 是典型的使用压缩算法的加壳工具。其他的压缩型加壳工具还有 ASPack[30]、NsPack[31]、PeCompact[32]等。除了通过压缩躲避检查的方法外, 有些加壳工具使用加密手段来保护代码, 例如 Armadillo、Themida、Molebox 等。

表 2 加壳统计

Table 2 Packing statistics

是否加壳	所占比例
加壳	19.5%
未加壳	80.5%

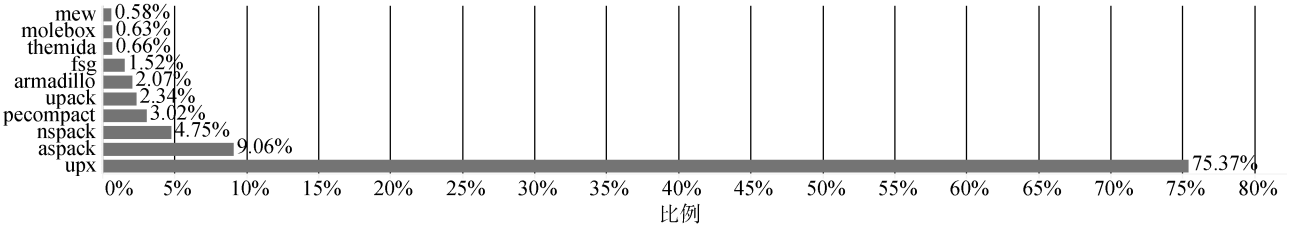


图 2 加壳工具统计

Figure 2 Packers statistics

恶意代码家族。本节对恶意代码的家族进行统计。我们的恶意代码判定系统使用多引擎的方式对恶意代码进行综合判定, 每个引擎给出的家族名命名规则不统一, 无法进行统计分析, 因此我们使用 AVClass[33]进行家族名的统一化。AVClass 实现了一种最先进的技术来规范化, 删除通用令牌, 并检测恶意软件样本的一组 AV 标签之间的别名。AVClass 只要能够输出名称, 就意味着不同的防病毒软件对恶意软件所属的类(系列)存在普遍共识。如表 3 所示, 恶意代码家族多为木马、蠕虫, 用于构建僵尸网络, 窃取信息等, 其次是宏病毒、广告软件。

在恶意样本家族统计中, Mirai[34]变种占 20.0%, 是占比最高的恶意家族。它是一款能自我传播的蠕虫, 通过发现、攻击并感染存在漏洞的 IoT(Internet-of-Things)设备实现自我复制, 也是一种僵尸网络, 通过一组中央命令控制服务器来控制被感染的设备, 以达到通过僵尸网络进行大规模网络攻击的目的。关于 Mirai 最早的公开报告出现在 2016 年 8 月, 曾在 2016 年 9 月份大规模 DDoS 攻击 Krebs on Security 以及 OVH 时风靡一时[35], 利用存在漏洞的家用路由器、空气质量检测仪以及个人监控摄像头等小型物

表 3 恶意家族统计

Table 3 Malware family statistics

家族名称	所占比例
Mirai	20.0%
Ramnit	14.1%
Gafgyt	8.4%
Flystudio	3.6%
Kuaiba	1.4%
Delf	1.3%
Virut	1.3%
Onlinegames	1.2%
Mailcab	0.7%
Hiddad	0.7%
Qqpass	0.6%
Razy	0.6%

联网设备进行 Mirai 样本的感染, 该攻击由这些被控制的设备所发起。

Ramnit[36]是一种通过可移动驱动器传播的蠕虫。该蠕虫还可用作后门, 允许远程攻击者访问受感染的计算机。Delf[37]多用于数据窃取。Mailcab[38]是一种群发邮件宏病毒, 它通过将自身插入受感染计算机上的任何打开的 Microsoft Excel 文档进行传播。然后它将自己发送到 Microsoft Outlook 通讯簿中的所有联系人。

API。将恶意代码调用的 API 进行统计。如表 4 所示, 我们列出了部分敏感 API, 其中次数表示每个程序调用API的平均次数, 是API调用次数除以恶意代码数量得到的。例如每个程序平均调用 RegSetValueExW 2.43 次。

表 4 API 统计

Table 4 API statistics

API	次数
RegSetValueExW	2.43
RegSetValueExA	2.00
DeviceIoControl	1.61
CreateRemoteThread	1.48
CreateProcessInternalW	0.42
ShellExecuteExW	0.07
SetWindowsHookExA	0.045
SetWindowsHookExW	0.033
IsDebuggerPresent	0.011
InternetOpenUrlW	0.009
URLDownloadToFileW	0.005

从 API 统计结果来看, 恶意代码设置注册表键值非常频繁, 倾向于创建本地或者远程的进程或线程, 设置消息钩子来进行监视, 从网络下载文件等。部分代码还采用反检测技术, 使用 IsDebuggerPresent 来反调试。

漏洞利用。一些恶意代码利用漏洞, 来得到计算机或者设备的控制权。本节对数据集中采用漏洞利用的代码进行分析。

数据集中检测到漏洞利用的代码共 227 个, 其中利用的漏洞共 36 个, CVE-2017-11882、CVE-2017-17215、CVE-2017-0199 是被利用次数最多的漏洞。CVE-2017-11882[39]是 Microsoft Office 内存损坏漏洞, 可导致远程执行。CVE-2017-17215[40]华为家用路由器 HG532 发现远程代码执行漏洞, 攻击者利用该漏洞进行蠕虫传播, 构建僵尸网络。CVE-2017-0199[41]是 Microsoft Office 文档漏洞, 可导致远程执行。除此之外, Android 漏洞 CVE-2011-1823 [42]允许本地用户绕过检查执行任意代码, 以及通过漏洞 CVE-2012-6422[43]获得部分安卓手机的权限。

3.2 传播特征

本节对恶意代码的传播特征进行分析, 主要从以下三方面进行: (1)恶意代码传播量统计; (2)恶意代码感染源地区分布, 以及面临感染风险的地区分布; (3)域名和恶意代码的承载关系。

恶意代码传播量。本节对恶意代码家族传播量进行了统计分析, 以家族为单位的恶意代码传播量的比例如表 5 所示。

表 5 恶意代码传播量统计

Table 5 Malicious program traffic statistics

家族名称	所占比例
Agentb	10.22%
Floodad	10.17%
Mirai	10.17%
Gaofenquming	6.27%
Skeeyah	4.84%
Qjwmonkey	4.26%
Funshion	3.90%
Microfake	2.74%
Hafen	2.66%
Zegost	2.36%

据统计, Agentb 是传播量最大的家族, 占比 10.22%。Agentb 是用于破坏、阻塞、修改或者复制数据, 以及破坏计算机或网络性能的恶意代码系列。这系列程序的流行意味着我们的计算机和网络时刻面临着威胁, 我们必须采取安全的措施保护网络和计算机。

其次是 Floodad 占比 10.17%, 这是一种广告软件, 在程序运行时自动显示广告的软件包。除此之外, Qjwmonkey 和 Funshion 也是广告软件。广告软件通常采用网络浏览器弹出广告的形式, 在大多数情况下, 用户不知道本地计算机上安装的广告软件组件。从某种程度上来说, 广告软件不是恶意软件, 但是在这里我们将其和恶意软件一起统计, 来观察广告软件的状况, 事实确是如此, 广告软件非常流行, 很大程度上影响用户体验。

Mirai 是在上文恶意代码家族统计中提到过的, 它是恶意代码中占比最高的家族。在这部分中, Mirai 在恶意家族传播量中占比 10.17%, 也是最流行三个家族之一。除此之外, Gaofenquming 是文件传播器,

具有通过将代码附加到其他程序或文件来传播的能力。Skeeyah 能够窃取个人资料, 下载更多恶意软件或让黑客控制受感染电脑, 其被微软称为 2016 年亚洲常见三大恶意软件之一。Microfake 和 Zegost 是特洛伊木马, 具有远程访问连接处理, 执行拒绝服务, 或者分布式拒绝服务, 捕获键盘输入, 删除文件或对象或终止进程的功能。除了明确的恶意软件, 广告软件之外, 还有一部分风险软件 Hafen, 因其以不合理的方式利用资料, 可能带来安全风险。

恶意代码感染源地区。本节我们对传播记录中除中国大陆以外的感染源地区进行了统计, 在此次研究的数据集中下载源覆盖了 132 个国家和地区。在如图 3 所示感染源地区中, 美国是主要感染源地区, 美国占比 65.21%, 其次是英国、印度、中国香港、俄罗斯、法国等地区。总体来看, 感染源呈集中化。

面临感染风险地区。本节我们将传播记录中我国受攻击的地区进行统计, 即面临感染风险地区的分布统计。如图 4 所示, 面临感染风险地区共 32 个。广东省以 21.80%的比例居于首位, 其次是四川省占比 7.95%, 上海市占比 7.13%。

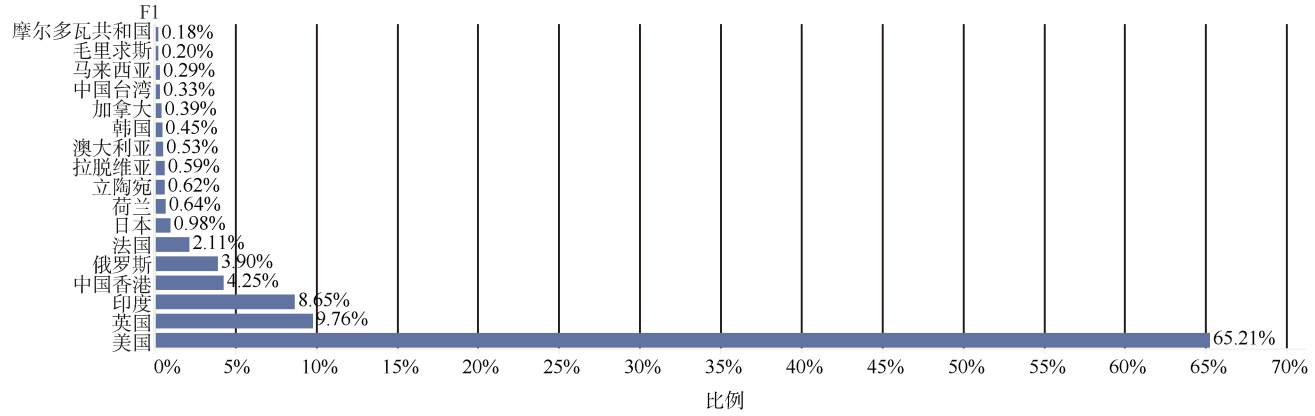


图 3 感染源地区

Figure 3 Source of infection

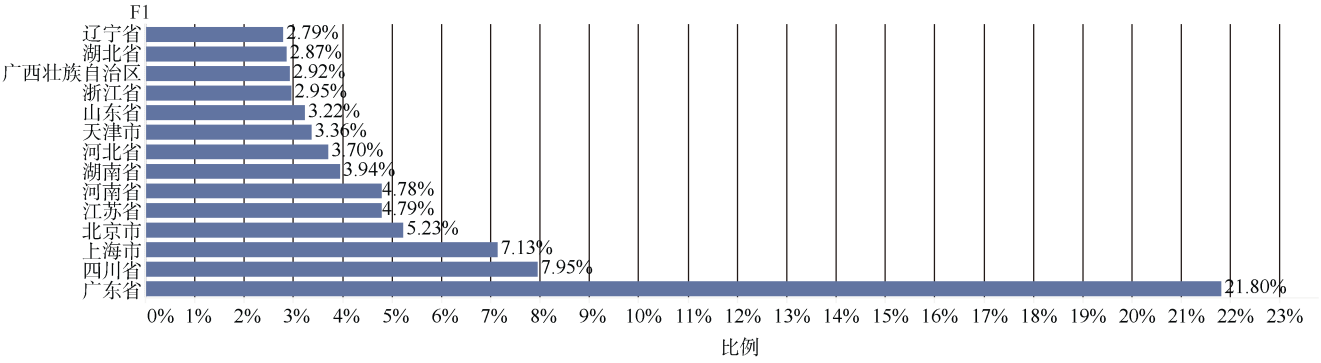


图 4 面临感染风险地区

Figure 4 Destination of infection

域名和恶意代码的承载关系。很多恶意代码存在于多个域名上, 同时, 一个域名也可以承载多个恶意代码。在这一节, 我们对传播记录中域名和恶意代码的承载关系进行探索。

根据统计结果, 一个域名平均承载 3.5 个恶意代码。其中约2%的域名承载67.6%的恶意代码, 在这2%的域名中, 平均每个域名承载 118 个恶意代码。另外90.7%的域名只承载一个恶意代码。可见, 大部分域名承载一个恶意软件, 少部分域名承载多个域名, 域名和恶意代码的关系呈现明显的长尾分布特征。

4 基于社区发现的攻击团伙发现

社区发现是发现社区的方法, 社区反应的是网络中个体行为的局部特性和相互之间的关联关系, 利用社区发现可以帮助我们理解整个网络的结构, 以及分析、预测整个网络中各元素之间的关联或者交互关系。攻击团伙往往在攻击资源上表现为社区行为, 存在同一团伙内的关联性, 即, 某一团伙内攻击资源可能共享。因为攻击团伙和社区的相似性, 我们使用社区发现算法对攻击团伙进行研究。

在前面章节中提到, Mirai 是此次恶意代码传播日志研究中, 恶意代码家族统计中变种占比最大, 且具有最大传播量之一的恶意家族。因此, 本章节我们对 Mirai 网络进一步研究。首先, 我们对 Mirai 的机理和特点进行简要阐述, 以及对数据集中的 Mirai 代码特点和传播特征进行测量。然后, 基于社区发现算法对 Mirai 僵尸网络进行探索。

4.1 Mirai

Mirai 是恶意软件的一种, 使得运行 Linux 的计算机系统成为远程操控的僵尸, 以达到通过僵尸网络进行大规模网络攻击的目的。Mirai 的主要感染对象是物联网设备。Mirai 构建的僵尸网络已经参与了几次影响广泛的大型分布式拒绝服务攻击(DDoS 攻击)。

Mirai 由两个功能组成, 传播和攻击。传播功能是指能够自我传播的蠕虫。攻击功能是指它会通过一组中央命令控制(Command and Control, C&C)服务器来控制被感染的设备。此次研究的传播记录中, 记录着大量的 Mirai 传播记录, 我们通过记录来了解 Mirai 僵尸网络的状况, 以及探索攻击者的偏好。

对数据集中 Mirai 代码进行统计, 代码基本上都是 ELF 文件类型, 其中 26.8%使用 UPX 工具进行加壳, 1.1%程序被检测出是漏洞利用(exploit)程序, 主要利用了两种漏洞, CVE-2017-17215 和 CVE-2018-10088。华为 HG532 系列路由器是一款为家庭和小型办公用户打造的高速无线路由器产品, CVE-2017-17215 是华为家用路由器 HG532 发现远程代码执行漏洞, 攻击者利用该漏洞进行蠕虫传播, 构建僵尸网络。该漏洞[44]被用来作为病毒 Mirai 的升级版变种 OKIRU/SATORI, 漏洞利用的是 upnp 服务存在的注入漏洞, 从而实现任意命令执行。CVE-2018-10088[45]是 XiongMai uc-httpd 缓冲区错误漏洞, XiongMai uc-httpd 是中国雄迈(XiongMai)公司的一款应用于摄像机等产品中的 HTTP 保护程序。XiongMai uc-httpd 1.0.0 版本中存在缓冲区溢出漏洞。攻击者可借助 Web 摄像机阅读器界面利用该漏洞造成拒绝服务。

对 Mirai 程序的传播记录进行统计, 感染源地区如表 6 所示, 主要来源地区是印度, 占比 51.9%, 其次是美国 33.9%。

4.2 社区发现算法应用

2017 年 9 月 30 日, 作为 Mirai 的作者, Anna-senpai 通过某个黑客论坛公布了 Mirai 源代码。自此多个团体或者团伙开始利用 Mirai 来发起攻击。为了更加了解这些团体, 我们使用社区发现算法来发现他们, 并了解他们。

表 6 Mirai 的感染源地区

Table 6 Source area infected by mirai

地区	所占比例
印度	51.90%
美国	33.90%
法国	8.95%
中国	1.39%
拉脱维亚	0.96%
荷兰	0.86%
英国	0.74%
加拿大	0.40%
俄罗斯	0.24%
丹麦	0.24%

构建社区网络是社区发现的第一步。我们将团伙之间的交互关系用社区网络进行表示, 团伙的交互主要体现在攻击资源的共享。数据集内容包括恶意代码的 md5、下载源 IP、受攻击 IP、下载源 IP 地区、受攻击 IP 地区、HTTP 请求的域名、时间、次数。下面我们对这八个属性进行讨论选取。

下载源 IP 地区、受攻击 IP 地区、时间和次数对网络构建意义不大, 我们不作考虑。

恶意代码的 md5、下载源 IP、HTTP 请求的域名三个属性均属于团伙的攻击资源范围内, 也就是说, 同一个团伙可能使用相同的 md5, 因此将 md5 作为社区网络的交互关系, 下载源 IP 和域名也是类似。其中 HTTP 请求的域名需要进行过滤, 除去无意义字符串, 即直接使用 IP 字符串做域名的数据, 得到 102 个不同域名。

受攻击 IP 我们倾向于不作为关联关系的参考, 因为 Mirai 程序具有自我传播功能和攻击功能, 受攻击 IP 包括被感染和被控制的用户 IP。由于该恶意代码传播具有蠕虫特点, 其传播路径并不完全受攻击团伙控制, 若将其传播对象纳入社群发现中的特征将引入较多噪声。我们为了证明此观点也进行了对比实验, 将受攻击 IP 作为关联关系与受攻击 IP 不作为关联关系两种情况都进行了社区发现。

4.2.1 实验设计

实验一将 md5、下载源 IP、受攻击 IP、域名作为关联关系, 实验二将 md5、下载源 IP、域名作为关联关系。

首先, 我们提取数据集中 Mirai 程序的传播记录, 实验一使用四元组<md5, ip1(下载源 IP), ip2(受攻击 IP), domain(域名)>表示, 共 6700 条无重复记录, 实验二使用三元组<md5, ip1(下载源 IP), domain(域

名)>表示, 共 541 条无重复记录。其次, 我们将域名作为社区网络结点, 将域名之间的关联关系作为社区网络的边, 这里的关联关系是指, 共享攻击资源的域名具有关联关系, 以实验一为例, 传播记录中存在<md5_1, ip1_1, ip2_1, domain_1>和<md5_2, ip1_2, ip2_2, domain_2>两条记录, 其中 md5_1=md5_2, 或者 ip1_1=ip1_2, 或者 ip2_1=ip2_2, 那么 domain_1 与 domain_2 存在关联关系, 增加社区网络的边<domain_1, domain_2>。需要强调的是网络是无向网络, 且两个结点之间最多存在一条边, 排除指向自己的边。

根据上文描述的利用域名之间的关联关系, 实验一网络具有 756 条边, 实验二网络具有 231 条边。

在图论中, 集聚系数是用来描述一个图中的结点之间结集成团的系数。具体来说, 是一个结点的邻接点之间相互连接的程度。局部集聚系数是指图中具体的某个结点与它相连的点抱成团的程度, 该结点的局部集聚系数等于所有与它相连的结点之间所连的边的数量, 除以这些结点之间可以连出的最大边数。一个结点的局部集聚系数总是在 0 到 1 之间, 越接近 1, 说明它的邻居们越是抱成一团。平均集聚系数是图中所有结点的局部集聚系数的平均值。

实验一中网络图的平均集聚系数为 0.67, 实验二中为 0.88, 说明实验二比实验一中网络结点更有形成团的倾向。

在社区网络构建完成之后, 我们使用社区发现算法 Fast Greedy 来发现社区结构。Fast Greedy 算法[46]是基于贪婪算法思想的一种凝聚算法, 也是基于模块度[47]优化的社区发现算法。算法首先将网络中每个结点自定义成一个社区; 其次计算出任意两个社区合并的模块度, 找到模块度增加最大的合并方式进行社区合并; 最后直到合并成一个社区停止, 找到这个过程中模块度最大的社区划分, 模块度 Q 定义如下:

$$Q=\sum_{i}\left(e_{ij}-a_i^2\right)$$

其中, i 和 j 分别表示两个社区, 分别称为社区 i 和社区 j, e_{ij} 表示社区 i 和社区 j 内部边数目的和与总边数的比值, a_i 表示社区 i 内部的点所关联的所有边的数目与总边数的比例。

模块度 Q 值用来衡量一个社区的划分是不是相对比较好的结果, 范围在 0~1 之间, Q 值越大说明网络划分的社区结构准确度越高, 在实际的网络分析中, Q 值的最高点一般出现在 0.3~0.7。

4.2.2 实验结果

社区发现算法实验结果如图 5 所示, 左图为实验一结果, 右图为实验二结果。图中结点为 102 个域名, 不同颜色代表不同的社区, 左图中点过于集中域名无法显示。实验一发现 7 个社区, 社区的模块度为 0.36, 实验二发现 27 个社区, 社区的模块度为 0.79。

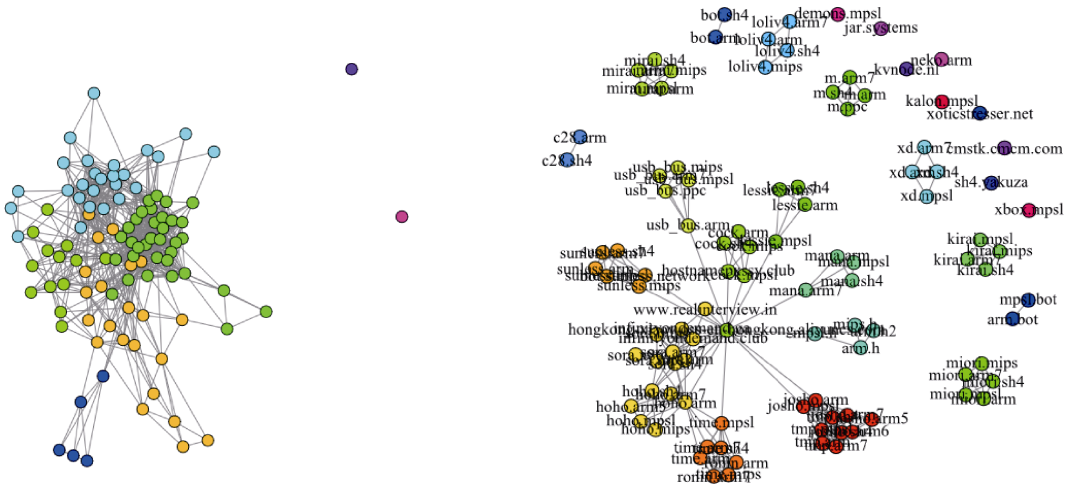


图 5 社区

Figure 5 Community

从模块度的角度来看, 实验二的社区结果更令人满意。从图形化的结果来看, 实验一中社区错综复杂, 没有社区的明显边界, 实验二中每个社区相对独立, 界限清晰。从域名分类结果来看, 实验二中同一社区内的域名有明显的命名规则, 实验一没有此特征。由上述三点可得到, 实验二的实验设置更加合理, 受攻击 IP 确实存在大量混淆。

实验二中的 27 个社区中, 最大的社区有 15 个域名, 最小的 1 个域名, 其中 18 个社区包含的域名数量大于 1。如表 7 所示, 最大的社区 1, 拥有 15 个域

名, 其命名规则为 sora.*、hoho.*、infinityondmand.*。其次是拥有 10 个域名的社区 2, 其命名偏好为 josho.*、tmp.*。拥有 7 个域名的社区 3 命名规则为 time.*、ronin.*。拥有 6 个域名的社区 4、5 命名规则分别为 sunless.*、cock.*。拥有 5 个域名的社区 6、7、8 命名规则分别为 usb_bus.*、mirai.*、miori.*。

表 7　社区命名规则

Table 7　Community names rules

社区编号	命名规则
1	sora.*、hoho.*、infinityondmand.*
2	josho.*、tmp.*
3	time.*、ronin.*
4	sunless.*
5	cock.*
6	usb_bus.*
7	mirai.*
8	miori.*

4.2.3　实验评估

我们使用 t-SNE(t-distributed stochastic neighbor embedding)对实验的结果进行评估。本文使用的社区发现算法本质上属于聚类, 因此我们使用 t-SNE 进行算法评估。t-SNE 是一种用于挖掘高维数据的非线性降维算法。它将多维数据映射到适合于人类观察的两个或多个维度, 通过视觉直观验证算法有效性。

将实验一的四元组< md5, ip1, ip2, domain >共 6700 条记录作为输入矩阵, 设置每条记录的标签文本为该条记录中域名所属的社区编号 1 到 7。标签文本仅用于结果的显示。同样将实验二的三元组< md5, ip1, domain >共 541 条记录作为输入矩阵, 设置每条记录的标签文本为该条记录中域名所属的社区编号 1 到 27。实验参数均设置 dims = 2, 表示降维之后的维度为 2; max_iter = 500, 表示最大迭代次数为 500。

实验结果如图 6 所示, 左图为实验一结果, 右图为实验二结果。图中每个点为一条记录, 用标签文本(社区编号)来标识, 且用不同颜色来区分。

左图中大部分团都很分散, 例如编号为 1、2、3、5 的点分布, 部分集中, 部分散落到各处, 与社区发现结果不一致, 表明聚类效果差。右图中相同颜色或者相同编号的点比较接近, 大致聚成团, 而不同的点之间比较远, 利用 t-SNE 可视化的结果和社区发现结果基本一致, 因此可以认为实验二结果有效。

无论是对实验结果的分析和实验评估都表明利用恶意代码的 md5、下载源 IP、域名来作为团伙的关联关系更加合适, 且社区发现算法能够发现团伙的社区结构。观察发现每个社区都体现出明确的命名偏好, 社区间表现出清晰地差异。因此基本可以推测攻击者为多个团体, 采用不同的基础设施传播 Mirai 程序。

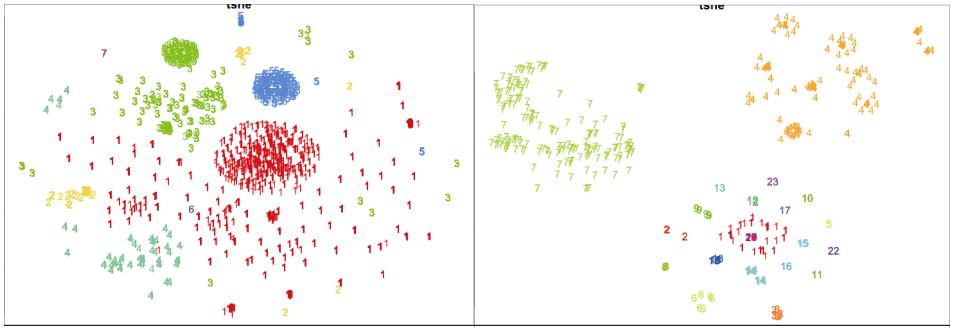


图 6　T-SNE

Figure 6　T-SNE

5　结论

本文基于 2018 年 10 月至 2019 年 3 月 CNCERT 互联网测试样本数据集, 从中判断恶意性代码, 对恶意代码的传播记录进行提取, 通过对恶意代码的多维特征进行分析, 对网络安全状况进行全面的剖析。除此之外, 传播最广泛的为 Mirai 恶意代码家族, 攻击者试图控制具有漏洞的物联网设备, 形成僵尸网络, 发起攻击。然后将社区发现应用于 Mirai 网络攻击的团伙发现中, 发现多个攻击团伙, 表现出明显的团伙差异。

下一步工作将应用社区发现于其他的网络研究中, 发现团伙特征, 对团伙进行刻画描述。

致　谢　该研究内容由国家自然科学基金重点项目(No. U1736218)和科技部重大专项(No. 2018YFB0804704)提供支持。

参考文献

[1] “2018 年我国互联网网络安全态势报告,” 国家互联网应急中心

CNCERT, https://mp.weixin.qq.com/s/-p7Uf9vdoJPTgKVAEPpbYA, Apr. 2019.

[2] X.Y. Jing, Z. Yan, and W. Pedrycz, “Security Data Collection and Data Analytics in the Internet: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 586-618, 2018.

[3] 朱天, 严寒冰, 朱丽, "基于网络攻击伴随行为的DDoS攻击群体分析方法", CN108173884A[P]. 2018.

[4] P. Porras, and V. Shmatikov, “Large-scale collection and sanitization of network security data: risks and challenges,” *New security paradigms Workshop* (NSPW’06), pp. 57-64, 2006.

[5] A. Papadogiannakis, M. Polychronakis, and E. P. Markatos, "Streamoriented network traffic capture and analysis for high-speed networks," *IEEE J. Sel. Areas Commun*, vol. 32, no. 10, pp. 1849-1863, Oct. 2014.

[6] L. Deri, “nCap: Wire-speed packet capture and transmission,” *Workshop on End-to-End Monitoring Techniques and Services* (E2EMON’05), pp. 47-55, 2005.

[7] L. Braun, A. Didebulidze, N. Kammenhuber, and G. Carle, “Comparing and improving current packet capturing solutions based on commodity hardware,” *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (IMC’10), pp. 206-217, 2010.

[8] J. L. García-Dorado, J. Aracil, J. A. Hernandez, and J. E. L. de Vergara, “A queueing equivalent thresholding method for thinning traffic captures,” *NOMS 2008-2008 IEEE Network Operations and Management Symposium* (NOMS’08), pp. 176-183, Apr. 2008.

[9] W. Y. Li, D. Wu, and B. Zhou, “A study of traffic collection techniques for network management and accounting systems,” *Proc. 8th Int. Conf. Comput. Supported Cooper. Work Design* (CSCWD’04), pp. 578-581, 2004.

[10] C. Wheelus, T. M. Khoshgoftaar, R. Zuech, and M. M. Najafabadi, “A session based approach for aggregating network traffic data—The SANTA dataset,” *2014 IEEE International Conference on Bioinformatics and Bioengineering* (BIBE’14), pp. 369-378, 2014.

[11] K. H. Ramah, H. Ayari, and F. Kamoun, “Traffic anomaly detection and characterization in the Tunisian National University network,” *International Conference on Research in Networking* (NETWORKING’06), pp. 136-147, 2006.

[12] S. B. Alias, S. Manickam, and M. M. Kadhum, “A study on packet capture mechanisms in real time network traffic,” *2013 International Conference on Advanced Computer Science Applications and Technologies* (ACSAT’13), pp. 456-460, 2013.

[13] R. Gad, M. Kappes, and I. Medina-Bulo, “Monitoring traffic in computer networks with dynamic distributed remote packet capturing,” *2015 IEEE International Conference on Communications* (ICC’15), pp. 5759-5764, 2015.

[14] L. Zabala, A. Ferro, and A. Pineda, “Modelling packet capturing in a traffic monitoring system based on Linux,” *2012 International Symposium on Performance Evaluation of Computer & Telecommunication Systems* (SPECTS’12), pp. 1-6, 2012.

[15] A. N. Singh, and R. C. Joshi, “A honeypot system for efficient capture and analysis of network attack traffic,” *2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies* (ICSCCN’11), pp. 514-519, 2011.

[16] K. Gao, J. Liu, J. Guo, and R. An, “Study on data acquisition solution of network security monitoring system,” *2010 IEEE International Conference on Information Theory and Information Security* (ICITIS’10), pp. 674-677, 2010.

[17] N. Bonelli, A. DiPietro, S. Giordano, G. Procissi, F. Vitucci, L. Salgarelli, G. Bianchi, and N. Blefari-Melazzi, “Towards smarter probes: In-network traffic capturing and processing,” *Trustworthy Internet*, pp. 289-301, 2011.

[18] E. Cozzi, M. Graziano, and Y. Fratantonio, “Understanding linux malware,” *2018 IEEE Symposium on Security and Privacy* (SP’18), pp. 161-175, 2018.

[19] X. Zhang, B.Q. Liu, and X.L. Wang, “Research on community detection methods in complex network,” *Computer Engineering and Applications*, vol. 51, no. 24, pp. 1-7, 2015.
(张鑫, 刘秉权, 王晓龙, “复杂网络中社区发现方法的研究”, *计算机工程与应用*, 2015, 51(24): 1-7.)

[20] R. Zhou, H.L. Yan, and H.X. Pan, “Research on Network Community Discovery Method for Complex Networks,” *Modern Economic Information*, no. 24, pp. 9-10(in Chinese), 2017.
(周瑞, 阎海玲, 潘华贤, “面向复杂网络的Web社区发现方法研究”, *现代经济信息*, 2017(24): 9-10.)

[21] S.C. Liu, F.X. Zhu, and L. Gan, “A La-bel-Propagation- Probability-Based Algorithm for Overlapping Community Detection,” *Chinese Journal of Computers*, vol. 39, no.4, pp. 717-729, 2016.
(刘世超, 朱福喜, 甘琳, “基于标签传播概率的重叠社区发现算法”, *计算机学报*, 2016, 39(4): 717-729.)

[22] M. Rosvall, and C.T. Bergstrom. “An information-theoretic framework for resolving community structure in complex networks,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7327-7331, 2018.

[23] S.F. Gong, W.L. Chen, and P.T. Jia, “Survey on algorithms of community detection,” *Application Research of Computers*, vol. 30, no. 11, pp.3216-3220+3227, 2013.
(龚尚福, 陈婉璐, 贾澎涛, “层次聚类社区发现算法的研究”, *计算机应用研究*, 2013, 30(11): 3216-3220+3227.)

[24] X. Liu, and D.Y. Yi, “Complex Network Community Detection by Local Similarity”, *Acta Automatica Sinica*, vol. 37, no. 12, pp. 1520-1529, 2011.
(刘旭, 易东云, “基于局部相似性的复杂网络社区发现方法”, *自动化学报*, 2011, 37(12): 1520-1529.)

[25] J. Parry, D. Hunter, K. Radke, and C. Fidge, “A network forensics

tool for precise data packet capture and replay in cyber-physical systems," *Proceedings of the Australasian Computer Science Week Multiconference* (ACSW'16), pp. 1-10, 2016.

[26] K. Shvachko, H. Konstantin, S. Radia, and R Chansler, "The hadoop distributed file system," *MSST*, vol. 10, pp. 1-10, 2010.

[27] M. Zaharia, M. Chowdhury, and M.J. Franklin, "Spark: Cluster computing with working sets," *HotCloud* (HotCloud'10), 2010.

[28] V. Laxmi, M.S. Gaur, and P. Faruki, "PEAL—packed executable analysis," *International Conference on Advanced Computing, Networking and Security* (ADCONS'11), pp. 237-243, 2011.

[29] "UPX: Ultimate Executable Packer", UPX, http://upx.sourceforge.net, Apr. 2019.

[30] "Aspack: Executable Packer", Aspack, http://www.aspack.com, Apr. 2019.

[31] "NSPack: Executable Packer", NSPack, http://nspack.lastdownload.com, Apr. 2019.

[32] "PECompact: Executable Packer", PECompact, http://www. pecompact.com, Apr. 2019.

[33] M. Sebastián, and R. Rivera, "Avclass: A tool for massive malware labeling," *International Symposium on Research in Attacks, Intrusions, and Defenses* (RAID'16), pp. 230-253, 2016.

[34] C. Kolias, G. Kambourakis, and A. Stavrou, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80-84, 2017.

[35] M. Antonakakis, T. April, and M. Bailey, "Understanding the mirai botnet," *26th {USENIX} Security Symposium* (USENIX'17), pp.1093-1110, 2017.

[36] "W32.Ramnit", Symantec, https://www.symantec.com/security-center/writeup/2010-011922-2056-99, Mar. 2015.

[37] "Backdoor.Delf.Family", Symantec, https://www.symantec.com/security-center/writeup/2003-050207-0707-99, Feb. 2007.

[38] "XM.Mailcab@mm", Symantec, https://www.symantec.com/security-center/writeup/2012-030512-4322-99, Apr. 2012.

[39] "CVE-2017-11882", NVD, https://nvd.nist.gov/vuln/detail/CVE-2017-11882, Nov. 2017.

[40] "CVE-2017-17215", NVD, https://nvd.nist.gov/vuln/detail/CVE-2017-17215, Mar. 2018.

[41] "CVE-2017-0199", NVD, https://nvd.nist.gov/vuln/detail/CVE-2017-0199, Apr. 2017.

[42] "CVE-2011-1823", NVD, https://nvd.nist.gov/vuln/detail/CVE-2011-1823, Jun. 2011.

[43] "CVE-2012-6422", NVD, https://nvd.nist.gov/vuln/detail/CVE-2012-6422, Dec. 2012.

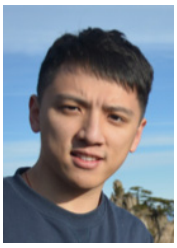
[44] "CVE-2017-17215-HG532 命令注入漏洞分析", 先知社区, https://xz.aliyun.com/t/4819, Apr. 2019.

[45] "CVE-2018-10088", NVD, https://nvd.nist.gov/vuln/detail/CVE-2018-10088, Aug. 2018.

[46] M.E.J. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, pp. 066-133, 2004.

[47] A. Clauset, M.E.J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, pp. 066-111, 2004.

王琴琴 于 2017 年在大连理工大学网络安全专业获得学士学位。现在中国科学院信息工程研究所网络空间安全专业攻读博士学位。研究领域为网络空间安全。研究兴趣包括: 恶意代码分析, 异常流量分析。Email: wangqinqin@iie.ac.cn



周 昊 于 2017 年在北京邮电大学信息与通信工程专业获得硕士学位。现任国家计算机网络应急技术处理协调中心初级工程师。研究领域为反网络诈骗、网络安全监测等。研究兴趣包括：机器学习、复杂网络。Email: zhh@cert.org.cn

韩志辉 于 2015 年在中国科学院软件研究所获得博士学位。现任国家计算机网络应急技术处理协调中心工程师。研究领域为系统安全与网络安全。Email: hzh@cert.org.cn

严寒冰 于 2006 年在清华大学计算机系获得博士学位。现在国家计算机网络应急技术处理协调中心任运行部主任，中国科学院信息工程研究所客座博士生导师，北京航空航天大学客座博士生导师。研究领域为网络安全，网络攻防、图形图像处理与分析、海量数据检索。Email: yhb@cert.org.cn

梅 瑞 于 2014 年在北京大学软件工程专业获得硕士学位，现在中国科学院大学网络空间安全专业攻读博士学位，研究领域为软件安全分析。Email: meirui@iie.ac.cn