第 43 卷 第 9 期 计 算 机 工 程 2017 年 9 月
Vol. 43 No. 9 Computer Engineering September 2017

·安全技术· 文章编号：1000-3428(2017)09-0185-09 文献标志码：A 中图分类号：TP393.08

基于域名系统流量的 Fast-Flux 僵尸网络检测方法

左晓军[1]，董立勉[1]，曲 武[2]

（1. 国网河北省电力公司电力科学研究院，石家庄 050021；
2. 北京启明星辰信息安全技术有限公司核心研究院，北京 100193）

摘 要： 在僵尸网络中，为保持服务器的可用性和隐蔽性，与域名关联的 Flux-Agent 的 IP 地址需要不停地变动，而黑名单策略对于阻止 Fast-Flux 僵尸网络攻击已经失效。为解决该问题，基于域名系统流量的分析和识别技术，提出一种新的 Fast-Flux 僵尸网络检测方法，用于检测互联网中使用 Fast-Flux 技术的僵尸网络，且对域名的分析不局限于来自垃圾邮件、点击欺诈或黑名单列表的可疑域名。实验结果表明，该方法能够以较高的准确率检测 Fast-Flux 僵尸网络，并且有利于完善黑名单列表。

关键词： 僵尸网络；Fast-Flux 域名；域名系统流量；层次聚类；机器学习

中文引用格式： 左晓军，董立勉，曲 武. 基于域名系统流量的 Fast-Flux 僵尸网络检测方法[J]. 计算机工程，2017，43(9)：185-193.

英文引用格式： ZUO Xiaojun，DONG Limian，QU Wu. Fast-Flux Botnet Detection Method Based on Domain Name System Traffic[J]. Computer Engineering，2017，43(9)：185-193.

Fast-Flux Botnet Detection Method Based on Domain Name System Traffic

ZUO Xiaojun[1]，DONG Limian[1]，QU Wu[2]

（1. Electric Power Research Institute，State Grid Hebei Electric Power Company，Shijiazhuang 050021，China；
2. Core Research Institute，Beijing Venustech Cybervision Co.，Ltd.，Beijing 100193，China）

【Abstract】 In a botnet，to maintain availability and invisibility of servers，the IP address of Flux-Agent associated with the domain name is changing constantly，and the blacklist policy is no longer effective in preventing Fast-Flux botnet attacks. In order to solve this problem，based on the analysis and recognition technologies of domain name system traffic，a new Fast-Flux botnet detection method is proposed. The method can detect the botnet using Fast-Flux technology in the Internet，which is not confined to the analysis of suspicious domain names from spame-mails，click fraud，or blacklists. Experimental results show that，this method can detect Fast-Flux botnets with higher accuracy，and help to give a more perfect blacklist.

【Key words】 botnet；Fast-Flux domain name；Domain Name System（DNS）traffic；hierarchical clustering；machine learning

DOI： 10.3969/j.issn.1000-3428.2017.09.033

0 概述

僵尸网络（Botnet）是攻击者出于恶意目的，传播受控僵尸程序控制大量主机，并通过一对多的命令控制信道所组成的网络，例如 IRC 协议、P2P 协议、HTTP 协议。基于 HTTP 协议构建僵尸网络的 C&C 策略主要包含 2 个方面的优点：1）IRC 协议是僵尸网络主流协议，安全研究领域对于 IRC 协议关注已久，并进行了深入的研究。基于 HTTP 协议构建的僵尸网络，其 C&C 通信可以隐藏在大量的互联网 Web 应用中，从而使得基于 HTTP 协议的僵尸网络更难以被检测。2）防火墙对于 IRC 协议的阻断。由于 IRC 协议已经不再是主流的协议，且被大量僵尸网络利用，许多企业、机构都在防火墙上过滤了该协议，而使用 HTTP 协议可以通过防火墙。

C&C 服务器在僵尸网络中扮演关键角色，不但包含受控主机的信息，而且可以控制发动攻击。一旦僵尸网络的 C&C 服务器被研究人员跟踪并攻破，该僵尸网络将会被接管。为了避免 C&C 服务器被检测到，僵尸主控者采用 Fast-Flux 技术提高僵尸网络的健壮性，该技术采用一个不断变化的受控代理主机的网络进行通信。本质上是一种域名系统

作者简介： 左晓军（1973—），男，高级工程师、硕士，主研方向为信息安全、网络管理、软件工程；董立勉，高级工程师；曲 武，博士后、CCF 会员。

收稿日期： 2016-05-19 **修回日期：** 2016-07-26 **E-mail：** quwu_ustb@163.com

(Domain Name System,DNS)技术的新应用,是于DNS RR(Resource Record)上设定非常短的生存时间(Time To Live,TTL),并以循序的方式在一个 IP 集合中做频繁替换,这些 IP 集合表示受控主机集合,在 Fast-Flux 技术中作为 proxy 角色。因此,僵尸网络导致的 DNS 流量与合法用户的 DNS 流量存在本质的区别,可以通过监测旁路 DNS 流量,提取相应特征对采用 Fast-Flux 技术的僵尸网络进行检测。本文提出一种基于 DNS 旁路流量检测 Fast-Flux 僵尸网络的算法,设计并实现原型系统 BotFired 的检测。

1 背景和相关工作

1.1 Fast-Flux 僵尸网络

文献[1]综述了僵尸网络的相关技术,并说明了 Single-Flux 与 Double-Flux 2 种类型的 Fast-Flux 网络。Fast-Flux 于 DNS RR 上设定非常短的 TTL,并以循序的方式在一个 IP 集合中做频繁替换,这些 IP 集合表示受控主机集合,在 Fast-Flux 技术中作为 Proxy 角色。图 1 显示了正常网络与 Fast-Flux 网络之间的技术差异,通常合法网站的域名查询返回的 IP 是固定不变的、规模有限的集合,而 Fast-Flux 网络返回的 IP 是频繁变化的,而且用户查询的内容并不存在这些 IP 对应的主机上,这些主机仅作为 Proxy 将需求转发给实际的内容提供主机 Botmaster,并将结果返回给用户。对于使用浏览器点击链接的用户而言,这个过程是相同的,但是背后使用的技术和影响却是差异很大。对于攻击者,利用 Fast-Flux 技术使得僵尸网络不易被追查,延长了僵尸网络的寿命。

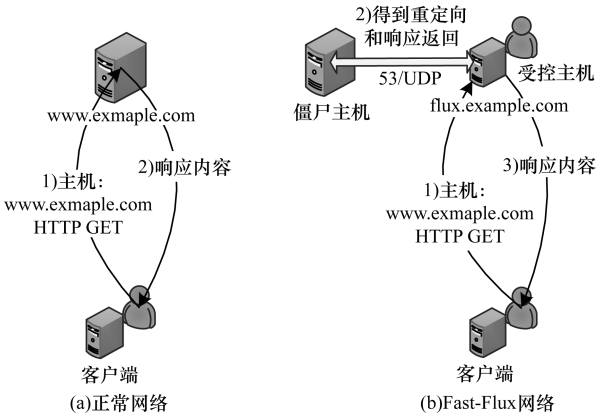


图 1 正常网络与 Fast-Flux 网络的差异

Single-Flux 通常在一个网络中包含多个受控主机,进行登记和网络域名地址的注销。该技术通常与 DNS A 地址记录相关,并且为单一网络域名生成一个可变的目标地址列表。而且,Single-Flux 技术的 DNS 记录 TTL 设置通常较短,以保证记录不会被缓冲,并保证网络地址能够快速变更而避免被记录。Double-Flux 与 Single-Flux 技术相似,但更为复杂,除了不断变化 DNS A 记录以外,对于 DNS NS 记录也不断变化。如图 2 所示,Single-Flux 与 Double-Flux 具有一定的技术差异。在不考虑 DNS 缓存的情况下,虽然用户查询 DNS 所得到的结果是一样的,但 Single-Flux 技术查询名称服务器的 IP 是固定的,而 Double-Flux 技术查询名称服务器(受控主机)的 IP 是不断变化的。因此,Double-Flux 技术使得名称服务器仅表现 Proxy 功能,用于流量导向,这些受控主机并不具备名称服务器功能,而实际的 DNS 响应记录则在攻击者控制的 Botmaster 主机上。如果 Double-Flux 的一个受控主机被发现,以上实现能够确保僵尸网络具备一个保护层和生存能力。在使用 Double-Flux 技术的架构中,由于受控主机在 DNS 网络中仅作为代理,将这些受控主机隐藏在一个代理网络中,有助于保护控制节点。通过添加多个代理,提高僵尸网络的生存率。

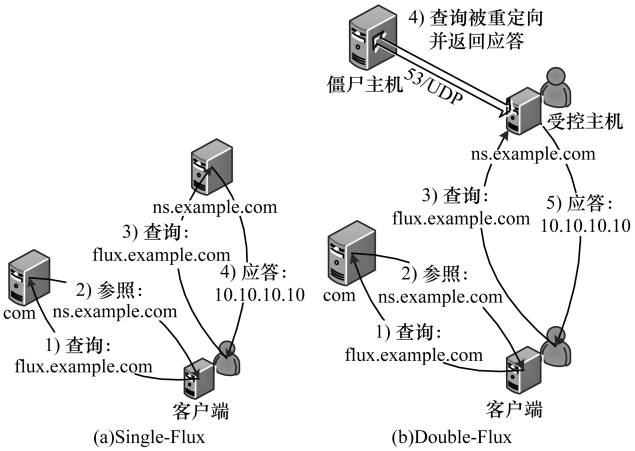


图 2 Single-Flux 与 Double-Flux 技术的差异

1.2 僵尸网络检测

在网络安全领域,僵尸网络已经成为最大的威胁之一。之前的研究成果大部分聚焦在分析和理解僵尸网络的操作和威胁。由于僵尸程序的模块化及其开发工具的广泛传播,具有一般技能的黑客就能建立起自己的僵尸网络。而且,能够在基于特征的杀毒程序和 IDS 更新特征之前制造出大量僵尸程序变种。此外,沙箱、蜜罐和密网都能够捕获和分析僵尸恶意程序,但此类方法需要花费太长的时间导致不能够从实际上减轻僵尸网络的危害。为了克服这些限制,很多僵尸网络检测方法都被提出,试图从受控节点的主机行为和网络行为进行僵尸网络检测。文献[2]对于 Fast-Flux 类的僵尸网络做了详细综述,包括产生背景、分类、特点、检测方法和减轻或移除方法。由于 IRC 协议是被僵尸网络使用最广泛的 C&C 通信协议,大多数检测方法都是聚焦于 IRC 协议特征进行检测,例如流量监控、使用 IRC 协议的 C&C 服务器识别等。文献[3]采用被动监控 IRC 流

量方式检测 IRC 昵称、IRC 服务器,以及使用不常用的服务器端口来检测僵尸主机。文献[4]提出使用 IRC 消息统计和 TCP 流量异常来检测僵尸网络。基于 IRC 的 C&C 通信和僵尸网络中其他常见行为,免费的僵尸网络检测工具 BotHunter[5] 使用 IDS 驱动的对话关联进行检测。文献[6]提出基于典型的 Netflow 数据进行检测,即分析受控主机和 C&C 服务器之间的 Netflow 数据查找异常。由于采用了 Netflow 数据,该类技术能够部署到 ISP 的核心交换机位置进行大规模僵尸网络检测。2008 年,Georgia 理工学院公开了 BotSniffer[7],该系统是一个用于检测和阻断僵尸网络的原型系统。利用 HTTP 和 IRC 的 C&C 流量分析技术,BotSniffer 试图通过深入分析 C&C 通道的通信模式来确定受控僵尸主机。但由于 BotSniffer 过于依赖基于 IRC 和 HTTP 协议的 C&C 流量进行检测,对于其他自定义的协议,例如 P2P 协议、FTP 协议等,以及加密的 C&C 流量无能为力。

文献[8]研究了僵尸网络的结构,并且强调了基于 P2P 协议的僵尸网络潜在的威胁。同时,认为仅通过 C&C 通信检测僵尸网络并不是完全有效的。文献[9-10]建立一个分布式评估架构对使用 IRC 协议的僵尸网络的行为进行评测,表示僵尸网络贡献了互联网中绝大多数的恶意流量。文献[11]研究了僵尸网络的每个特征,并且对其传播进行建模。近年来,基于 P2P 协议的僵尸网络出现在互联网环境中,此类僵尸网络使用 P2P 架构作为 C&C 通道,对于节点失效具有很强的免疫力,故很难摧毁此类僵尸网络。文献[12]分析了当前主流的 P2P 僵尸网络 Storm 的架构和通信协议。而文献[13]提出一种更为高级的复合 P2P 僵尸网络。为改善 P2P 僵尸网络在节点失效和 C&C 通道被关闭情况下的存活能力,提出使用更为健壮的连接、控制流量多样性,加密 C&C 通道,灵活的系统恢复等机制进行僵尸网络优化。

2007 年,德国密网工作组首次在互联网上捕获到使用 Fast-Flux 技术的僵尸网络[14],并详细说明 Fast-Flux 技术,定义了 Single-Flux 与 Double-Flux 2 种类型的 Fast-Flux 网络。研究人员已经在 Fast-Flux 域名检测领域做了大量工作[15-17]。根据调研的结果,这些工作多数的区别在于描述 Fast-Flux 域名的行为特征个数和分类算法细节,而且研究的数据源大部分都局限于垃圾邮件中的 URL 链接。这些垃圾邮件主要来自垃圾邮寄蜜罐或者垃圾邮件过滤器,而 URL 链接则是从垃圾邮件的内容提取的。然后,使用获取的 URL 链接库,采用主动探测的方式重复发起 DNS 请求,收集 DNS 返回的 IP 地址集合,分类 URL 及其映射的 IP 地址集为 Fast-Flux 服务和非 Fast-Flux 服务。文献[18]提出的方法与前文的方法不同,并不局限于使用来自垃圾邮件的 URL 数据。使用来自边界路由器的 NetFlow 信息,能够识别出使用重定向 Flux 技术的僵尸网络。但是,提取的 Netflow 信息并不能检测使用透明转发代理的 Flux 客户端,仅是针对重定向的主机。而且,同样使用 DNS 主动探测技术,这与文献[15,17]类似,但更多的是为了分类来自垃圾邮件的可疑域名和关联 Netflow 信息。

本文提出一种新的恶意 Fast-Flux 服务网络的旁路检测和跟踪方法,通过旁路分析 DNS 服务器流量来实现检测。在检测原型系统 BotFire 中,通过在 DNS 服务器前面使用端口镜像技术获得 DNS 的旁路流量,从而实现对用户的 DNS 请求和 DNS 返回的监测。由于电力企业网络 DNS 服务器的流量较大,DNS 数据采集器在将数据发送到存储平台之前,可选用预置的过滤器识别合法的域名请求,仅保留可疑的 Fast-Flux 域名请求。为了避免过滤掉疑似的 Fast-Flux 域名请求,过滤器通常采用保守策略。但是,该过程依然能够削减大量的合法 DNS 流量,保留疑似 Fast-Flux 域名请求。对于潜在的恶意 Fast-Flux 域名将被采集器送往存储平台,然后可以根据时间粒度进行更为细粒度的分析。利用聚类算法将相互关联的域名进行聚合。例如,对于指向相同 Internet 服务或相同的 CDN 或相同的恶意 Fast-Flux 网络的域名分别聚合。域名聚合之后,使用分类算法对聚合后的域名分类为恶意的 Fast-Flux 服务网络和合法服务。以上过程与传统的方法不同[15-17],传统的方法更关注于对单个域名的判别,即恶意的 Fast-Flux 服务和合法的 Fast-Flux 服务。此外,本文实现了检测恶意 Fast-Flux 服务的原型系统 BotFire,并且使用来自电力企业的 DNS 旁路流量进行检测。测试中,旁路采集器每天收到来自不同地区 IP 的 0.5亿个 DNS 请求,存储平台共存储 45 d 的此类数据。

2　检测系统

2.1　BotFire 架构

通过对 Fast-Flux 域名的研究,Fast-Flux 服务网络通常具有以下特征:较短的 TTL,快速变化的 IP 集合,IP 集合分布在不同的网络中。然而,一些合法的服务,例如合法的 CDN、NTP 服务器池、IRC 服务器池等,通常使用一个与 Fast-Flux 服务类似的域名集合提供服务。例如,与合法 CDN 相关的域名也经常使用一个较低的 TTL,并且解析为一个跨多个网络的 IP 地址集合。与 NTP 服务器池相关的域名也会使用一个非常大的 IP 地址集合,并且使用类 Round-Robin 算法周期变化。仅分析单一的特征并不能准确地将恶意 Flux 域名与合法域名区分开。因

此,BotFire 系统将综合使用多个特征进行分类。

BotFire 系统的总体架构如图 3 所示。每个 DNS 旁路采集器将监控来自用户的 DNS 请求和 DNS 服务器的解析响应。由于 BotFire 系统的存储模块采用的是 HDFS 分布式文件系统,可以满足电力企业内部对 DNS 大流量的需求。此外,BotFire 系统的采集子系统也提供了预过滤模块,用户可以通过勾选过滤规则降低存储和分析的数据量。由于 BotFire 系统仅对 Flux 域名和解析的 IP 集合进行分析,流量过滤器 F1 负责识别与 Flux 域名最为相关的 DNS 查询,过滤非 Flux 域名。在 BotFire 系统中,候选的 Flux 域名列表 L 被保存在内存数据库中,并周期更新。L 列表包含以下特征:候选 Flux 域名,指定时间窗 ΔT 内域名最大的 TTL,指定时间窗 ΔT 内域名查询 DNS 返回的 IP 集合等。在指定的时间窗 ΔT 结束时,过滤器 F2 将对候选列表进行处理,即通过预定义的规则过滤非 Flux 域名,这些预定义的规则来自历史 Flux 域名的统计分析结果。例如,在 ΔT 范围内,对于指定的域名,F2 将会检测由 DNS 返回的 IP 集合规模是否变大。实际上,在 ΔT 范围内,如果一个域名被查询若干次,没有新解析的 IP 出现,则该域名基本不可能与恶意 Flux 服务相关。反之,对于一个给定的域名,若每隔 TTL 时间,由 DNS 解析的 IP 集合都会发生变化,该域名将会作为候选的 Flux 域名。对于由 F2 确认不是 Flux 服务相关的域名,将会从 L 列表中删除。

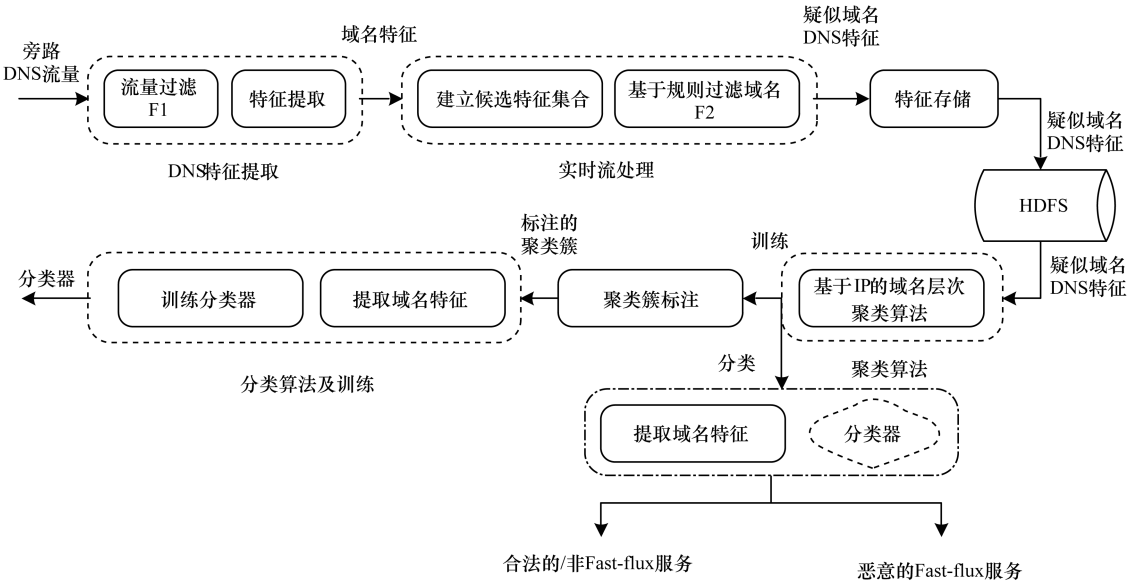


图 3 BotFire 系统的总体架构

当数据规模累积到某个固定的阈值或者经过了 $\Delta T'$ 的时间,候选的 Flux 域名列表 L 将被传输到大数据平台进行存储和进一步检测。在 BotFire 系统实现中,特征列表 L 将被存储到 HDFS 中,然后以指定的时间单位进行 Offline 分析,通常可以设定该单位为天、周、月。在分析过程中,首先根据域名所包含的 IP 集合进行归类,即使用聚类算法将使用相同服务的域名聚成一个簇。也就是说,在 $\Delta T'$ 范围内,对于 2 个候选的 Flux 域名,若它们所包含的解析 IP 集合交集大于某个设定的阈值,例如 1/3,则可以判断这 2 个域名使用的服务是相似的。基于 IP 集合相似性概念,BotFire 系统应用层次聚类算法将 IP 集合交集大于某个阈值的域名归为一个聚类簇。实验中,每个聚类簇都分别表示一个独立的候选 Fast-Flux 服务网络。毫无疑问,如果过滤器 F1 和 F2 都是设置的偏向保守,除了保留了大量的与恶意 Fast-Flux 服务相关的域名,也保留了部分与合法 CDN、NTP 服务器池和其他合法服务相关的域名,这些域名与恶意的 Fast-Flux 域名共享了若干特征上的相似性。因此,在聚类结果中,聚类簇既可能是恶意的 Flux 服务,也可能是合法的 CDN 服务、NTP 服务器池等。聚类结束后,每个候选的 Flux 服务网络,即域名和相关 IP 集合的聚类簇,都将进入服务分类器。该分类器将被训练对聚类簇为恶意 Flux 服务和合法/non-Flux 服务进行分类。

2.2 流量过滤器

为了描述流量裁剪过滤器 F1 的工作流程,首先需要形式化 DNS 查询和响应过程,即 $q^{(d)}=(t_i,T^{(d)},P^{(d)})$。其中,$q^{(d)}$ 表示在时间 t_i,某个用户查询 DNS,并获取了 DNS 返回的 IP 地址集合;$T^{(d)}$ 表示 DNS 响应的 TTL,即该 DNS 记录在 DNS 服务器上缓存时间;$P^{(d)}$ 表示由 DNS 服务器返回的解析 IP 地址集合;$prefix(P^{(d)},16)$ 表示从 $P^{(d)}$ 集合中获取的不同的 B 类 IP 地址的集合。为了过滤 DNS 的合法流量,并且避免误过滤疑似恶意 Flux 服务相关的域名,BotFire 系统的 F1 过滤器选择以下规则进行过滤:

F1-0:$d \in S_{\text{alexa}}$

F1-1:$T^{(d)} \geqslant 21.6 \times 103$ s

F1-2:$|P^{(d)}| \leqslant 2$ or $T^{(d)} \geqslant 30$ s

F1-3:$p = \dfrac{|prefix(P^{(d)},16)|}{|P^{(d)}|} \leqslant \dfrac{1}{3}$

接下来,主要描述以上规则的设置原理。如前文所述,Flux 域名通常随机生成、访问量不大、低 TTL,时间量级是分钟级别的,且解析后的 IP 集合随着时间快速变化。规则 F1-0 通过抽取 www. Alexa. com 站点前 1 000 000 条合法域名作为白名单,符合 $d \in S_{\text{alexa}}$的域名将会过滤掉。规则 F1-1 排除了域名 TTL 超过 2 h 的查询,这样的域名几乎不能是 Flux 域名。规则 F1-2 主要关注 Flux 域名的 DNS 查询,该类通常返回一个相对大的 IP 集合。Flux 域名使用该策略的原因是每台 Flux 终端的在线时间是不可预测的,因此使用大量的解析 IP 为 Flux 服务提供一个容错的机制。而且,也可以使用一个较小的 IP 集合,但设置非常低的 TTL 来实现该机制。使用该策略的情况下,若一个 Flux 客户端离线,使用另一个 DNS 查询将会马上发现新的 Flux 客户端,前面的响应将会从 DNS 缓存中迅速清除。规则 F1-2 的设置包括了以上这 2 个场景。规则 F1-3 主要关注 Flux 客户端通常分布在多个不同的网络和组织中。但是,大部分合法域名解析的 IP 地址都处于一个或几个不同的网络。本文使用函数 $prefix(P^{(d)},16)$计算解析后 IP 处于不同网络的数目,比率 p 允许系统能够排除非合法 Flux 服务的域名。

2.3　规则过滤器 F2

当监控 DNS 流量时,内存数据库维护一个候选 Flux 域名列表 L,该列表存储候选 Flux 域名的历史信息,而且随着 DNS 查询送往过滤器 F1 后进行更新。为了阐述 L 的更新机制,形式化定义如下:在时间 t,一个候选的 Flux 域名 d 可以使用如下元组表示,$d = (t_i, Q_i^{(d)}, T_{\max i}^{(d)}, R_i^{(d)}, G_i^{(d)})$。其中,$t_i$是对于域名 d 最后一次 DNS 查询的时间戳;$Q_i^{(d)}$ 是直到时间点t_i域名 d 的总 DNS 查询数;$T_{\max i}^{(d)}$是直到时间点t_i域名 d 的最大 TTL;$R_i^{(d)}$是直到时间点t_i域名 d 解析的不重复 IP 地址累积集合;$G_i^{(d)}$ 是一个序列对 $\{(t_j, r_j^{(d)})\}_{j=1,2,\cdots,i}$, $r_j^{(d)} = |R_j^{(d)}| - |R_{j-1}^{(d)}|$。也就是在时间点$t_j$观测到的 IP 集合规模与$t_{j-1}$观测的到集合的差值。在序列 $G_i^{(d)}$ 中,系统仅保留满足 $r_j^{(d)} > 0$ 条件的集合。因此,$G_i^{(d)}$ 可以描述为直到时间点t_i,IP 集合的增长程度。当一个与域名 d 相关的 DNS 查询 $q^{(d)} = (t_i, T^{(d)}, P^{(d)})$通过 F1 过滤器,若满足 $d \in L$,则根据 $q^{(d)}$ 更新内存数据库中候选域名 d 的元组信息。为了能够压缩候选域名的规模,同时要保留疑似恶意 Flux 服务的域名。在每个固定的时间窗 ΔT,列表 L 将被过滤,每个 ΔT 都要检测候选的 Flux 域名是否满足预置条件。定义t_j为过滤操作发生时间,令$p = \dfrac{prefix(R_j^{(d)},16)}{|R_j^{(d)}|}$为直到时间$t_j$域名 d 解析后 IP 集合的网络前缀率。BotFire 系统使用 F2 过滤器从列表 L 中移除满足以下条件域名。F2-1:$Q_j > 100$ and $|G_j^{(d)}| < 3$ and ($|R_j^{(d)}| \leqslant 5$ or $p \leqslant 0.42$)。也就是说,对于 F2-1 过滤器,若查询查过 100 次,解析 IP 累积值增长不超过 2 次,IP 累积数较低(低于 5)或网络前缀率较低(低于 0.42),这样的域名将被过滤掉。过滤器 F2 主要是基于 Flux 域名特征进行判别,F2 过滤掉的域名很大概率上是与 Flux 服务无关的。

2.4　域名存储

本节将引入存储周期的概念,即将内存数据库的域名及其特征信息存储到 HDFS 的时间间隔,定义为 T_S,即每隔T_S时间,BotFire 系统将对内存数据库中候选的 Flux 域名列表 L 进行存储。通常,T_S是根据各采集器的速率之和与内存大小决定的,即根据数据规模和内存大小来决定,例如每小时数据规模较小,T_S将设置以天为单位,规模较大将设置以小时为单位。

2.5　域名聚类

与存储周期类似,引入聚类周期的概念,即启动聚类算法的时间间隔,定义为T_C,即每隔T_C时间,BotFire 系统将对 HDFS 中的候选 Flux 域名文件进行聚类,将 IP 集合相近的域名归为聚类簇。通常,T_C是根据各采集器的速率之和决定的,即数据规模来决定,例如,每天数据规模较小,T_C将设置以周为单位,规模较大将设置以天为单位。引入聚类过程主要基于以下考虑,为了规避黑名单列表,僵尸网络通常使用大量的 Fast-Flux 域名控制恶意 Flux 服务,这些域名所有都指向与同一个 Flux 服务相关的 Flux 客户端。为了聚类 Flux 域名,使用 CURE 层次聚类算法。CURE 算法先把每个数据点看成一类,然后合并距离最近的类,直至类个数达到要求为止。为了应用聚类算法处理域名集合 $D = \{d_1, d_2, \cdots, d_n\}$,首先需要对距离相似性进行定义:

$$sim(\alpha,\beta) = \frac{|R^{(\alpha)} \cap R^{(\beta)}|}{|R^{(\alpha)} \cup R^{(\beta)}|} \cdot \frac{1}{1 + e^{\gamma - \min(|R^{(\alpha)}|, |R^{(\beta)}|)}}$$

其中,α 和 β 表示 2 个域名;$R^{(\alpha)}$和 $R^{(\beta)}$分别表示这 2 个域名解析的 IP 地址集合,相似性距离范围定义为 $sim(\alpha,\beta) \in [0,1]$。因子$\dfrac{|R^{(\alpha)} \cap R^{(\beta)}|}{|R^{(\alpha)} \cup R^{(\beta)}|}$表示集合 $R^{(\alpha)}$和 $R^{(\beta)}$之间的 Jaccard 距离,用来评估这 2 个 IP 地址集合之间的相似性。$\dfrac{1}{1 + e^{\gamma - \min(|R^{(\alpha)}| - |R^{(\beta)}|)}}$为置信因子,可以描述为$|R^{(\alpha)}|$和$|R^{(\beta)}|$之间的差值越大,置信因子越大。为了说明置信因子的选择,见以下实例:对于 $|R^{(\alpha)} \cap R^{(\beta)}| = 1$ 以及 $|R^{(\alpha)} \cup R^{(\beta)}| = 4$ 的

情况,和 $|R^{(\alpha)} \cap R^{(\beta)}| = 10$ 以及 $|R^{(\alpha)} \cup R^{(\beta)}| = 40$ 的情况,Jaccard 距离值是相同的,都为 0.25。但是,显然第 2 种情况的相似性高于第 1 种情况的相似性,因此认为第 2 种情况更为可信。γ 为先验参数,仅被用于调整置信因子。在实验中,设置 $\gamma = 3$,若 $|R^{(\alpha)} \cap R^{(\beta)}| = 3$,置信因子为 0.5。随着 $|R^{(\alpha)}|$ 和 $|R^{(\beta)}|$ 之间的差值变大,置信因子将趋向 1。

聚类过程中,相似性矩阵 $\boldsymbol{P} = \{s_{ij}\}_{i,j=1,2,\cdots,n}$ 包含了域名对 (d_i, d_j) 之间的相似性 $s_{ij} = sim(d_i, d_j)$。层次聚类算法将 $\boldsymbol{P}$ 作为输入,输出一个系统树图,即一个类树形结构,叶子表示集合 D 中的域名,边长度表示聚类簇之间的距离。聚类过程获得的系统树图并没有定义将域名分成聚类簇的分区,而是定义域名之间的关系。若要将域名集合 D 分成聚类簇,需要依据某个高度 h 对系统树图进行剪切。为了选择最好的树图剪切值,即获得最好的聚类簇,本文使用了一种基于多尺度辅助重抽样的层次聚类算法。实验表明,聚类簇的个数与剪切高度之间的关系,处于 0.2~0.6 的剪切高度值,聚类簇并没有发生明显的变化,故提供了一种合理的域名聚类簇结果。并且通过人工分析这些域名聚类簇,证明其结果是正确的。

2.6 域名分类

每个候选 Flux 域名的聚类簇 C_i 都可以被看成一种候选 Flux 服务,这种服务由 C_i 中所有域名集合和这些域名对应的 IP 地址集合联合定义。由于过滤器 F1 和 F2 采用的是保守过滤策略,不可能完全过滤掉与合法 CDN 和其他合法 Internet 服务(例如 NTP 池服务,此类服务类似于 Flux 服务)相关的域名。因此,在采集和聚类候选 Flux 域名后,采用监督学习方法建立分类器,将恶意 Flux 服务域名与合法/非 Flux 服务域名区分开。

为了区分恶意 Flux 服务与合法/非 Flux 服务域名,BotFire 系统引入了统计特征集合,这些统计特征主要是描述 Flux 服务的行为特征。在文献[17]中,为监督学习方法提供了较为全面的 Fast-Flux 域名特征,通过查询单一域名返回的 IP 地址集获得 9 个统计特征。本文采用文献[17]中的部分特征描述与恶意 Flux 服务相关的域名簇。此外,添加了一些额外的新特征。将 BotFire 系统使用的特征分为 2 组,分别为主动特征和被动特征。主动特征定义为需要计算一些额外的外部信息才能获得的特征,例如 DNS 命令 nslookup,whois,IP 地址的地理信息,BGP 协议数据等。被动特征定义为从采集器的 DNS 信息获取的特征。对于每个域名聚类簇,BotFire 系统将计算以下特征。

被动特征描述如下:

1)IP 总数(Num_IP):在聚类簇中,所有域名对应的不重复 IP 总数;

2)域名总数(Num_Domain):在聚类簇中,所有不同域名的总数;

3)域名的平均 TTL(Avg_TTL):在聚类簇中,域名的平均 TTL;

4)网络前缀多样性(Div_NetPre):该特征是一个比例,即 B 类网络数目与 IP 集合规模的比值。该特征用来评估 IP 地址集合在不同网络中的发散程度,与特征 7)和特征 8)类似;

5)每个网络域名个数(Num_Perdom):观测聚类周期 $T_{c,m-1}, T_{c,m-2}, \cdots, T_{c,m-n}$,对于指定的域名聚类簇内,解析了至少一个 IP 地址不重复的域名个数,其中 n 为自然数,默认值为 7。尽管 Flux 域名的高可变性,Flux 网络本质上还是相当稳定的[16]。因此,同一个 Flux 客户端可以被许多不同的域名使用。在聚类簇中,与 IP 关联的域名个数可以使用此特征进行评价;

6)IP 增长率(GR_IP):在一个聚类簇中,每个 DNS 查询过程,返回与指定域名相关的新 IP 地址的平均数目,定义为 $\frac{1}{|C_i|} \cdot \sum_{d \in C_i} \frac{R^{(d)}}{Q^{(d)}}$。

主动特征描述如下:

1)BGP 前缀多样性(Div_BGP):定义为聚类簇中 IP 集合对应的不同自治系统(Autonomous System,AS)与 IP 集合规模的比值。自治系统多样性与 BGP 前缀多样性,组织多样性定义相类似;

2)国家码多样性(Div_CC):对于聚类簇中的每个 IP,可以映射为地理位置。因此,国家码多样性可以定义为 IP 地址对应的不同国家数目与 IP 集合规模的比值;

3)地区码多样性(Div_AC):对于聚类簇中的每个 IP,可以映射为地理位置。因此,地区码多样性可以定义为 IP 地址对应的不同地区数目与 IP 集合规模的比值;

4)动态 IP 率(Rate_DIR):恶意 Flux 服务网络的大多数受控主机主要是来自家庭用户机器。为了评估一个 IP 是否是一个企业或家用机,对每个 IP 执行一个逆向 DNP 查询命令,获取查询结果进行关键字匹配,例如 DHCP,DSL,DIAL-UP 等,在 DNS 中判断主机是否使用了动态 IP 地址。因此,动态 IP 率定义为使用动态 IP 地址的 IP 与 IP 集合规模的比值;

5)平均在线比率(Avg_Online):该特征通过主动探针进行获取,即指定的时间间隔连接聚类簇中的 IP 地址,例如一天连接 6 次,连接类型选择 TCP 连接,端口号分别为 80/53/443,即 HTTP/DNS/HTTPS 服务。如果受控主机接受建立 TCP 连接,表示该主机在线,否则判为离线。每个主机在线时间评估可以采用指定时间间隔内探测连接次数的成功

率。因此,平均在线比率定义为聚类簇 IP 探测连接成功次数和与探测总次数的比值。

在描述统计特征之后,应用随机森林分类器(Random Forest Classifier,RFC)将聚类簇C_i分成恶意 Flux 服务和合法/非 Flux 服务。使用随机森林分类器基于以下考虑:

1)RFC 在通用数据集上表现良好,相对其他算法有很大优势;

2)RFC 能够处理较高维度的数据,无需进行特征选择。并且,在训练完成后,RFC 能够给出哪些特征对于分类贡献更大;

3)在创建 RFC 时,对范化误差使用的是无偏估计。所以,在 RFC 算法中不需要再进行交叉验证或者单独的测试集来获取测试集误差的无偏估计;

4)RFC 训练速度快,且在训练过程中,能够检测到特征之间的相互影响;

5)实现相对简单,容易扩展成并行化方法。

在 RFC 训练过程中,首先需要对训练数据集进行标注,即对恶意 Flux 服务聚类簇和合法/非 Flux 服务聚类簇进行标注。标注后,RFC 基于训练数据集进行训练。然后,分类器被用来分类聚类簇。

3　系统训练和实验评估

在实验过程中,由于硬件环境运行其他服务,本次实验与其他程序共享计算资源,因此,在实验结果上可能会出现小幅波动。硬件环境构成包括 1 台 Master 管理节点服务器,7 台计算节点。

3.1　DNS 流量采集

在电力企业 DNS 服务器前,BotFire 检测系统的采集器采用旁路镜像流量,流量采集器设定为 Collector。在 28 d 内,采集器监控到 30 万个用户请求,时间为 2015 年 5 月 11 日到 6 月 8 日。在这期间,采集器获得 A 类型和 CNAME 类型的查询近 0.3 亿次,即 1.16×10eps,峰值为 4×102eps。在采集器上,可以根据需求对 DNS 流量进行预处理,即开启过滤器 F1 和 F2 的规则。BotFire 检测系统的运行周期可以设置为天、周或月,在实验中,采用以天为周期。如果将过滤器 F1 和 F2 的规则全部开启,DNS 流量的 90% 都能被过滤掉,即到达聚类组件的数据量可以被约减到 160 万,252 MB 的数据量。DNS 流量经过采集器过滤之后,候选 Flux 域名数据将会被转发到 HDFS 中。在实际检测过程中,由于 F1 和 F2 通常采用保守策略,候选 Flux 域名列表仍然包括所有的可疑恶意 Flux 服务域名,以及部分合法的 CDN、NTP 服务器池和其他合法服务。

3.2　候选 Flux 域名的聚类

在每个采集周期结束时,HDFS 中候选的 Flux 域名数据将会进行聚类。本文应用 CURE 层次聚类算法将同属于一个网络域名归为一个聚类簇,该算法采用的是一种从上而下逐步分解的策略。当采集周期设置为 1 d 时,HDFS 中每天数据规模为 252 MB。聚类模块的输出为系统树图,需要通过剪枝获得聚类簇,剪枝高度 $h=0.61$。该值的选择通过实验获取系统树图的最大平稳区域,该平稳区域能够从聚类簇图谱获得。$h=0.61$,位于最大平稳域的末端,能够提供高质量的聚类簇,获得 3 250 个域名聚类簇。聚类过程是一个完全的非监督过程,自动评估聚类结果质量是非常困难的。BotFire 检测系统通过将聚类结果映射到图谱上,然后人工分析图谱获得域名聚类簇的质量。在多数情况下,人工分析是最为直接的方法。人工分析表明,本文使用的 CURE 层次聚类算法能够正确识别属于恶意 Flux 服务的域名聚类簇。

3.3　聚类簇的分类

将域名聚类簇分成恶意 Flux 服务和合法/非 Flux 服务 2 类,该过程使用监督学习方法建立分类器。为了能够使用一个监督学习方法,需要对基准数据集进行标注,该数据集主要用来训练 BotFire 检测系统的分类器和评估分类器的精度。接下来描述如何对聚类簇进行标注,然后对分类器使用的统计特征向量进行分析。为了构建实验的基准标注数据集,需要人工对大量的聚类簇进行检查和标注。标注结束后,使用 RFC 分类器自动分类。

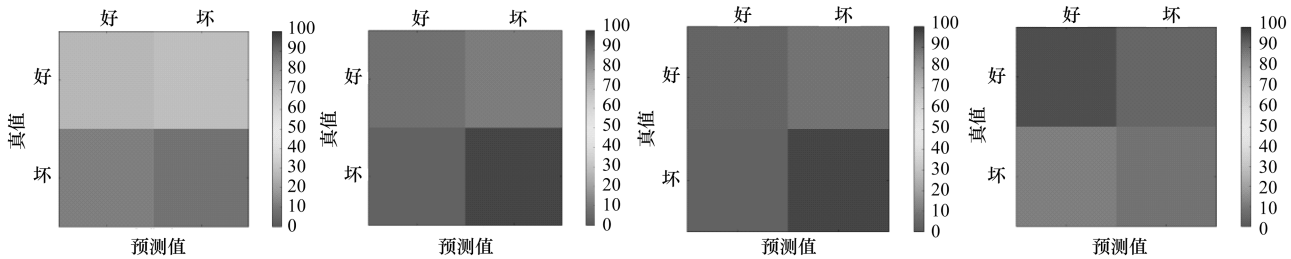
BotFire 检测系统的分类组件是使用 RFC 分类器最重要的原因之一,即训练后获取的 RFC 分类器是相当容易解释的,其输出的类别由个别树输出类别的众数决定。特别是,当使用统计特征进行训练时,IP 增长率是最具有区分能力的特征,是 RFC 分类器的多个决策树决定的。这也验证了快速变化的 Flux 域名 IP 是恶意 Flux 服务网络中最显著的特征。

由于本文仅聚焦于分类恶意 Flux 服务,而且对于每个 Flux 服务网络的 Flux 客户端数量相对较大。因此,在分类过程中,可以对聚类簇进行过滤,仅考虑解析的 IP 地址数大于指定阈值的聚类簇,例如$T_{\mathrm{IP}}\geqslant10$。在人工分析和交互性展示的协助下,在2015 年1 月,能够标注 432 个与恶意 Flux 服务相关的多类聚类簇,例如为恶意行为服务的 Flux 网络、成人内容、钓鱼网站等。同时,5 310 个与非 Flux 或合法 Flux 聚类簇,包括与不同 CDN 相关的聚类簇、NTP 池、IRC 池和其他合法的服务。使用预标注的数据集和 10 交叉验证方法,评估了分类器的精度,实验结果如表 1 和图 4 所示。显然,使用 RFC 分类器可以获得较高的 ROC 曲线下

面积。同时,分类器能够获得较高的检测率和较低的误报率。本文还分别测试了不同特征集合对分类结果的影响,结果如图 5 所示。分别使用所有特征、静态特征和动态特征(Fe-3,Fe-5,Fe-6)进行训练和分类,实验结果如表 1所示。RFC 分类器训练结束后,使用 PCA 算法选择出最重要的特征为 Fe-6,次要的为 Fe-3 和Fe-5。其中,Fe-6,Fe-3,Fe-5 分别表示 IP 增长率、域名的平均 TTL 和每个网络域名个数。这表明,这 3 个特征是区分恶意 Flux 网络和合法网络最重要的特征。为了简化比较过程,选择这 3 个特征来评估分类器的性能。正如表 1 所示,括号内的数据表示每次评估的标准方差。仅使用这 3 个特征即可获得较好的分类精度,与使用全部特征进行训练和分类获得的精度差异不大,但误报率相对较高。而且,评估过程处于可控的环境中。也就是说,使用上文提到的整个标注数据集训练RFC分类器。然后,使用训练好的分类器去分类新的域名聚类簇,这些域名聚类簇来自 6 月初到中旬的 DNS 流量。在分类过程中,平均每天获得了518 个域名聚类簇,其中,$T_{IP} \geq 10$。在分类结果中,33 个域名聚类簇被标识为恶意 Flux 网络服务,通过人工分析得出 RFC 分类器能够以较低的误报率获得很高的预测精度,确认了表 1描述的结果。总而言之,从 4 月初到 6 月底的 60 d 时间里,平均每天监测到 17 个恶意 Flux 服务网络,总计 18 124 个 Flux 域名和 7 235 个不同 Flux 客户端的 IP 地址。

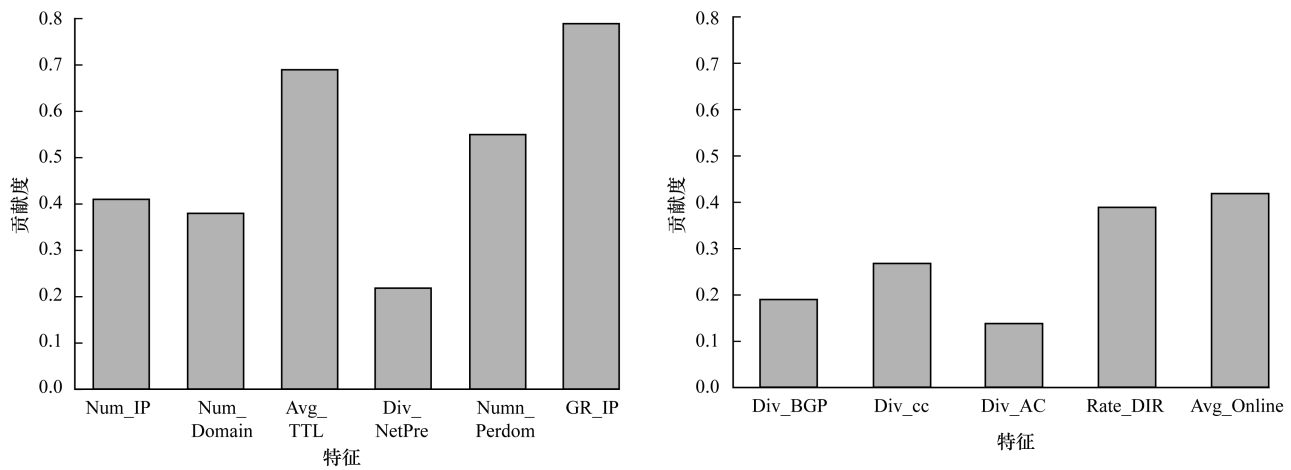
表 1 使用 10 交叉验证测试的分类性能

特征类别	ROC 曲线下面积	准确率/%	误报率/%
主动和被动特征	0.992(0.003)	98.9(0.36)	0.20(0.36)
被动特征	0.993(0.005)	99.4(0.53)	0.30(0.53)
Fe-6,Fe-3,Fe-5	0.989(0.006)	97.3(0.49)	0.32(0.49)
主动特征	0.620(0.080)	69.5(0.42)	2.80(0.46)



(a)主动特征的混淆矩阵 (b)Fe-6,Fe-3和Fe-5的混淆矩阵 (c)被动特征的混淆矩阵 (d)主动和被动特征的混淆矩阵

图 4 不同特征集合对应的混淆矩阵



(a)主动特征贡献 (b)被动特征贡献

图 5 特征的贡献指数

4 结束语

本文基于 DNS 流量分析,提出一种新的 Fast-Flux 僵尸网络检测方法。与传统的方法不同,本文提出的方法并不局限于分析垃圾邮件和预置黑名单列表中的恶意域名。通过用户的访问行为检测 Fast-Flux 僵尸网络,即用户访问各类垃圾内容,与此同时 DNS 记录了整个访问解析过程。实验过程中使用了电力企业网络 DNS 服务器的旁路流量,结果表明,本文实现的 BotFire 检测系统能够精确地检测恶意 Flux 服务网络。

BotFire 系统可以看作是一个僵尸网络检测的框架原型,可以在多个层面进行改进。当采集周期设置为 1 d 时,HDFS 中每天数据规模为 252 MB,若采用普通单机层次聚类算法,例如 CURE 层次聚类算法,处理 GB 级别的数据需要3.2 h。若采集周期设置为一周,聚类时间达到了22.4 h,不能满足检测系统的需求。因此,未来 BotFire 检测系统采用 Mahout 提供的分布式层次聚类算法,可根据数据规模对主机进行横向扩展,提高聚类性能。对于分类模块,当前使用 RFC 算法,为了满足大规模数据的训练和检测需求,BotFire 检测系统未来可引入 Mahout 提供的分布式 RFC 算法,根据数据规模对主机进行横向扩展,提高训练和分类性能。仅适用 DNS 流量数据进行检测是本文的一大特点,但由于 DNS 数据所含信息量有限,为进一步提高检测精度、降低误报率,可尝试使用灵活高效的多源数据采集与融合机制,例如引入 Netflow 数据、包解析或关联其他设备的检测结果。

参考文献

[1] 诸葛建伟,韩心慧,周勇林,等. 僵尸网络研究[J]. 软件学报,2008,19(3):702-715.

[2] ZHOU Shijie. A Survey on Fast-Flux Attacks [J]. Information Security Journal: A Global Perspective, 2015,24(4-6):79-97.

[3] GOEBEL J,HOLZ T. Rishi: Identify Bot Contaminated Hosts by Irc Nickname Evaluation[C]//Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets. New York,USA:ACM Press,2007:8.

[4] BINKLEY J R,SINGH S. An Algorithm for Anomaly-based Botnet Detection[C]//Proceedings of the 2nd Con-ference on Steps to Reducing Unwanted Traffic on the Internet. New York,USA:ACM Press,2006:43-48.

[5] GU Guofei,PORRAS P A,YEGNESWARAN V,et al. Bot Hunter: Detecting Malware Infection Through IDS-driven Dialog Correlation[C]//Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium. New York,USA:ACM Press,2007:1-16.

[6] KARASARIDIS A,REXROAD B,HOEFLIN D. Wide-scale Botnet Detection and Characterization [C]// Proceedings of the 1st Conference on Hot Topics in Understanding Botnets. New York, USA: ACM Press, 2007:1-7.

[7] GU G,ZHANG J,LEE W. Bot Sniffer: Detecting Botnet Command and Control Channels in Network Traffic[C]// Proceedings of the 15th Annual Network and Distributed System Security Symposium. New York,USA:ACM Press, 2008:2-19.

[8] COOKE E, JAHANIAN F, MCPHERSON D. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets [C]//Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop. New York,USA:ACM Press,2005:1-6.

[9] ABU R M, ZARFOSS J, MONROSE F, et al. A Multifaceted Approach to Understanding the Botnet Phenomenon [C]//Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. New York,USA:ACM Press,2006:41-52.

[10] FABIAN M, TERZIS M A. My Botnet is Bigger than Yours (Maybe, Better than Yours): Why Size Estimates Remain Challenging[C]//Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets. New York,USA:ACM Press,2007:1-5.

[11] DAGON D, ZOU C C, LEE W. Modeling Botnet Propaga-tion Using Time Zones [C]//Proceedings of Network & Distributed System Security Symposium. New York,USA:ACM Press,2006:2-13.

[12] GRIZZARD J B, SHARMA V, NUNNERY C, et al. Peer-to-Peer Botnets: Overview and Case Study[C]// Proceedings of the 1st Conference on Hot Topics in Understanding Botnets. New York, USA: ACM Press, 2007:1.

[13] WANG Ping, SPARKS S, ZOU C C. An Advanced Hybrid Peer-to-Peer Botnet [J]. IEEE Transactions on Dependable and Secure Computing,2010,7(2):113-127.

[14] HOLZ T, GORECKI C, FREILING F, et al. Detection and Mitigation of Fast-flux Service Networks[EB/OL]. (2009-03-03). http://www. isoc. org/isoc/conferences/ndss/08/papers/16_measuring_and_detecting. pdf

[15] 康　乐.基于 DNS 数据流的僵尸网络检测技术研究[D].哈尔滨:哈尔滨工业大学,2011.

[16] KONTE M,FEAMSTER N,JUNG J. Fast Flux Service Networks: Dynamics and Roles in Online Scam Hosting Infrastructure[M]. [S. l.]: Georgia Institute of Technology,2008.

[17] PASSERINI E, PALEARI R, MARTIGNONI L, et al. FluxOR: Detecting and Monitoring Fast-flux Service Networks [M]//Flegel U, Markatos E, Robertson W. Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin,Germany:Springer,2008.

[18] HU Xin, KNYSZ M, SHIN K G. Measurement and Analysis of Global IP-usage Patterns of Fast-flux Botnets[C]// Proceedings of INFOCOM ' 11. Washington D. C. ,USA:IEEE Press,2011:2633-2641.

编辑　顾逸斐