P2P 僵尸网络研究与进展

杜 勤 吕光丽 苏 忠 林 繁 张少华

(空军指挥学院网络中心 北京 100097)

摘 要 由于基于IRC协议的僵尸网络存在单点失效的天然缺陷,越来越多的僵尸网络转而使用非集中式命令与控制信道。基于P2P协议的僵尸网络就是其中最重要的一种。P2P僵尸网络经过10多年的发展,技术已经完全成熟,它们具有更强的弹性和鲁棒性,更难以被清除,被认为是新一代的僵尸网络。阐述了P2P僵尸网络的发展历程,详细分析了功能结构、分类方法和工作过程,介绍了P2P僵尸网络传播模型和跟踪、检测、防御方法的研究进展。

关键词 网络安全,P2P,僵尸网络,恶意代码,传播模型

中图法分类号 TP309.5 **文献标识码** A

Research and Development of Peer-to-Peer Botnets

DU Qin LU Guang-li SU Zhong LIN Fan ZHANG Shao-hua

(Network Center, Air Force Command College, Beijing 100097, China)

Abstract Because of IRC-based botnets' central point of failure, more and more botnets turn to decentralized Command & Control Channel. Peer-to-Peer protocol based botnet (P2P botnet) is the most important one. After more than 10 years' development, the technology of P2P botnets has been fully mature. They are more resilient and robust, and more difficult to be eliminated. They are considered to be the new generation of botnets. In this paper, the P2P botnet's evolution process was introduced, the functional structure, taxonomy and execution process of P2P botnet were discussed in detail, the propagation model and research development on technology of tracking, detecting and defending methods were proposed.

Keywords Network security, P2P, Botnet, Malware, Propagation model

1 引言

"僵尸程序"(bot)一词来源于"robot",是早期因特网中出现在IRC[1]聊天网络中方便管理员管理网络的智能程序,后来被黑客利用发展成为恶意程序,用来感染并远程控制主机。感染了僵尸程序的主机被称为僵尸主机(Zombie),由多个僵尸主机联成的网络就成为僵尸网络[2](botnet)。利用僵尸网络,攻击者(botmaster)能够发起包括发送垃圾邮件、分布式拒绝服务攻击、窃取敏感信息、点击欺诈、网络仿冒、传播恶意代码和保存非法文件等恶意行为[3]。正是由于为攻击者提供了简单的能够发起多种攻击方式的平台,僵尸网络得到迅速发展,很快成为网络攻击的主要手段之一。根据美国国会的一份研究报告显示,Shadowserver Foundation在2006年11月至2007年5月间发现1400台活跃的命令与控制服务器,而由这些服务器控制的僵尸主机从2007年3月到7月间由500,000台猛增到3000,000台[4]。早期的僵尸网络多是基于IRC聊天协议搭建集中式的命令与控制信道(Command & Control Channel, C&C),尽管它具有搭建简单、可扩展性强、传递命令效率高等优点,但是却存在单点失效的天然缺陷,即一旦命令与控制服务器被关闭,将会使整个僵尸网络瘫痪[5],于是基于P2P协议构造C&C的僵尸网络(简称P2P僵尸网络)就应运而生了。

与传统的客户机/服务器通信模式不同,P2P网络采用分布式结构,利用网络边缘节点(非中央服务器)传递控制命令、管理信息和资源文件,网络中每个节点既是客户机又是服务器。由于不存在中央节点,P2P僵尸网络具有更强的韧性,例如,Li等人的研究表明,类Gnutella僵尸网络在清除75%节点后仍能保持完全连接,在清除87.5%节点后97%的剩余节点仍保持连接[6]。从2002年6月28日发现首例P2P蠕虫Apache Scalper[7]以来,P2P僵尸网络功能和结构不断翻新,技术日渐成熟,对互联网安全的威胁也越来越大。2007年1月出现的Storm[8,9](也被称为Peacomm)基于Overnet网络协议,融合了社会工程学、Rootkit隐藏技术[10]、Fast-Flux技术[10]和对抗分析技术,被认为是第一个完全分布式的P2P僵尸网络[8]。据相关资料[11]显示,到2007年11月,Storm每24小时内有230,000台在线主机,而在2007年8月22日当

本文受国家自然科学基金项目(61071065)资助。

杜 勤(1982—),男,硕士生,助理工程师,主要研究方向为军事信息处理与军事信息网格、信息系统安全,E-mail:duqin2000@sina.com;吕光丽(1980—),男,硕士生,研究实习员,主要研究方向为军事网络系统安全;苏 忠(1969—),男,博士,副教授,主要研究方向为信息网络安全、系统性能评估;林 繁(1980—),男,硕士,讲师,主要研究方向为Web信息挖掘;张少华(1970—),男,博士,副教授,主要研究方向为军事信息处理与计算机网络战。

天，Storm产生的网络通信流量占互联网上所有被病毒感染信息的99%，成为世界上规模最大、危害最大的僵尸网络。

目前，P2P僵尸网络已经引起人们的高度重视，国内外各研究机构和反病毒厂商纷纷展开了对它的跟踪和研究，有些研究者认为它是下一代的僵尸网络[12]。本文主要从P2P僵尸网络发展历程、功能结构、工作过程等角度进行全面分析，并介绍当前P2P僵尸网络传播模型和检测、防御方法的研究进展。

2 P2P僵尸网络的发展历程

P2P僵尸网络的发展与P2P技术的发展和广泛应用密切相关。表1列举了近10年来互联网中出现的典型P2P技术应用和基于P2P协议的恶意程序。

表1 P2P僵尸网络发展历程

时间	名称	类型	描述
1999.05	Napster	P2P协议	首个得到广泛应用的拥有中央节点的P2P协议。
2000.03	Gnutella	P2P协议	首个采用完全分布式结构的P2P协议
2001.03	FastTrack	P2P协议	采取混合P2P结构的P2P协议
2002.06	Apche Scalper	P2P蠕虫	首个利用简单P2P协议进行通信的蠕虫病毒
2002.09	Slapper	P2P蠕虫	采用自制P2P协议进行通信的蠕虫病毒
2003.05	WASTE	P2P协议	使用RSA加密算法的P2P协议，用于组建小型P2P网络
2003.09	Sinit	P2P僵尸程序	使用随即扫描发现对等点加入网络的P2P僵尸程序
2003.11	Kademlia	P2P协议	使用分布式哈希表查找对等网中信息的P2P协议
2004.03	Phatbot	P2P僵尸程序	基于WASTE的P2P僵尸程序
2006.03	SpamThru	P2P僵尸程序	使用自制协议发送垃圾邮件且拥有中央节点的P2P僵尸程序
2006.04	Nugache	P2P僵尸程序	使用连接预定义节点加入网络的P2P僵尸程序
2007.01	Storm	P2P僵尸程序	基于Kademlia协议的P2P僵尸程序
2008.02	MayDay	P2P僵尸程序	使用80端口进行通信的P2P僵尸程序
2008.11	Waledac	P2P僵尸程序	使用自制的简单P2P协议的层次化P2P僵尸程序

P2P技术的发展大致可分为3代[13]。第一代以1999年出现的用于MP3音乐共享的Napster[14]为代表，采用集中式目录结构。它把多个用户主机联成一个对等网络，而使用一台目录服务器集中保存各用户主机提供的共享资源索引。用户首先登录目录服务器获取文件索引，然后连接实际保存文件的用户主机下载文件。第二代以2000年出现的Gnutella[14]为代表，采用纯分布式P2P网络结构。它不再使用中央服务器，各用户主机随机接入网络，与相邻一组节点对等连接。查询共享资源时，采取类似递归查询的方法，把文件查询消息广播到相邻节点，直到有主机反馈查询命中消息，或者直到所有主机都没有找到文件为止。这种网络具有良好的健壮性和可扩展性，但是却会明显降低速度和查询能力。纯分布式的P2P网络的另一种方案是使用分布式哈希表将节点组织成"有结构的"完全分布式P2P网络。这种方案以2003年出现的Kademlia为代表。第三代以2001年出现的FastTrack[15]为代表，采取混合式P2P结构，在纯P2P结构的基础上，引入超级节点的概念，将网络中的节点划分成多个"自治簇"，簇中采取集中式目录结构，簇间采取纯分布式P2P结构通过超级节点连接起来。由于只有在查询结果不充分的情况下，查询才会泛洪到别的"自治簇"，因此采取超级节点方式能够有效抑制网络广播风暴，同时能够提高查询速度。

受P2P协议及软件发展的影响和启发，P2P僵尸网络的发展到目前为止也大致经历了四个阶段。第一阶段为萌芽阶段，以Apache Scalper、Slapper和Sinit为标志。Apache Scalper是最早使用了P2P协议进行通信的恶意程序，但可能是由于其运行平台FreeBSD数量有限或是因为使用的P2P过于简单，没有在互联网上造成较大影响，从而没有引起人们的注意[7]；Slapper[7]对Apache Scalper进行了改进，使用自制的P2P协议进行消息的传递和路由，新感染的主机通过连接父节点IP加入网络，父节点向新感染主机传送自己所知的节点列表，并向全网广播新节点的加入信息，以保持全网的同步；Sinit[16]向随机IP地址发送探测包的方式以发现已感染的网络节点，尽管实现起来非常简单但效率低下，而且非常容易暴露。第二阶段为发展阶段，以Phatbot[17]、SpamThru[18]和Nugache[19]为典型代表。Phatbot采用了WASTE协议构建P2P网络，尽管WASTE协议本身没有中央节点，但是Phatbot却选择使用Gnutella网络的缓冲服务器发现僵尸节点，不但扩展性有限，而且依赖性强；SpamThru是一个发送垃圾邮件的P2P僵尸网络，它依靠一台中央服务器维持对网络的控制，在中央服务器被关闭时可以及时将所有节点都导向一台新的控制服务器；Nugache采取连接预定义的22个节点的方式加入网络，并能够通过美国在线及时通信工具(AIM)进行传播，在预定义节点被全部关闭时，网络将不能增长，但是已有节点仍能够保持连接。第三阶段为成熟阶段，以Storm为代表。Storm基于Kademlia协议构建完全分布式的P2P网络，它的出现标志着P2P僵尸网络技术已经成熟。第四阶段为新发展阶段，MayDay[20]和Waledac[21]是这个阶段的典型代表，它们都采用了多种协议进行通信以增强可靠性。MayDay在僵尸主机和控制服务器之间采用HTTP协议进行通信，当HTTP通信失败时，可以使用TCP和加密ICMP两种协议中的一种进行P2P通信；Waledac拥有基本的HTTP通信代理功能，处理命令与控制信息和其他节点之间的一般通信，Waledac节点担任Spammer、Repeater和Protector等角色，不同角色的节点之间利用简单自制P2P协议维持并更新节点列表。

3 P2P僵尸网络的功能结构

3.1 P2P僵尸网络组成

一个典型的P2P僵尸网络主要由攻击者主机、跳板主机、僵尸主机和P2P协议等4部分组成，如图1所示。攻击者主机主要功能是发布攻击、同步、更新等命令，掌握僵尸主机数量、状态、命令执行情况等信息。跳板主机主要用于采用集中式命令与控制信道的僵尸网络之中，它的功能主要是隐藏攻击者真实IP地址，实施反跟踪。由于P2P的天然特性(对等网络，一台主机既是服务器又是客户机)，攻击者可以从任意一点接入网络，从而非常有效地隐藏自己的IP，于是跳板主机是可选的。僵尸主机实施具体的攻击行为，包括感染节点、发送垃圾邮件、发起DDoS攻击等等，并且返回执行结果。P2P协议是僵尸网络的软件成分，实现了僵尸主机的对等互

连，构成了 P2P 僵尸网络的命令与控制信道。P2P 协议可以是现有的 P2P 网络应用的协议，也可以是僵尸程序编写者自制的协议。一款优良的 P2P 协议应该实现互联互通、容错冗余、加密认证、负载均衡等功能，保证僵尸网络能够保持较好的连通度、可扩展性、较强的韧性和较高的效率。除此之外，对于某些采用混合 P2P 结构的僵尸网络而言，网络中还存在一些具有特殊用途的中央服务器，这点将在 3.3.2 节中详细讨论。

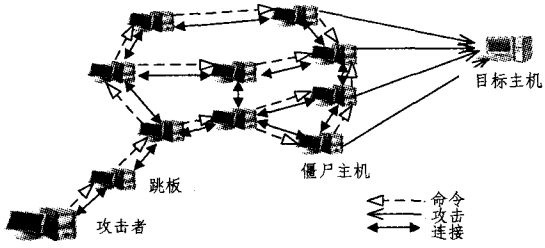

图 1　P2P 僵尸网络组成

3.2　P2P 僵尸程序功能结构分析

僵尸程序是僵尸网络各种功能的主要承载者，对僵尸程序进行系统分析将有利于深入了解僵尸网络的构造方法，从而研究评估、跟踪、检测和防御的有效方法。在参考诸葛建伟[2]等人所做的僵尸程序功能结构分析的基础上，深入分析了表 1 所列的各种 P2P 恶意程序，认为针对 P2P 僵尸程序，可以从以下两方面对文献[2]中僵尸程序功能结构进行改进：(1)增加通信和组网模块以适应 P2P 僵尸程序的需要。在基于 IRC 协议和 HTTP 协议的僵尸网络中，僵尸主机之间不存在信息交换，也不需要编写程序发现节点以加入网络。攻击者只需编写能够分析基于文本的命令字符串功能，即可实现对僵尸主机的控制。而在 P2P 僵尸网络中，情况完全不同，僵尸主机之间需要定期交换自己掌握的节点列表以保证同步，命令和控制信息也必须经过中间节点转发才能路由到所有僵尸主机。在我们考察的几种 P2P 僵尸网络中，新感染主机必须通过某种特定机制才能加入网络，这需要攻击者自己编程实现，例如 Sinit 向一些随机 IP 地址的 53 号端口发送发现(discovery)数据包来探测僵尸主机，只有发现僵尸主机才能够连入网络。(2)将信息窃取模块和僵尸主机控制模块合并，统称为恶意行为模块，执行攻击者的命令，完成恶意攻击。基于这两方面改进，我们提出如图 2 所示的 P2P 僵尸程序功能结构。

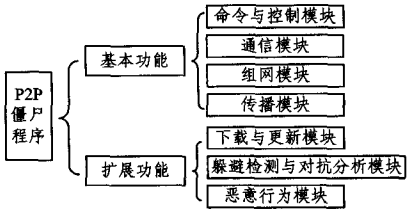

图 2　P2P 僵尸程序功能结构

P2P 僵尸程序功能结构仍分为基本功能模块和扩展功能模块两大部分。基本功能模块实现 P2P 网络的构建与控制，包括命令与控制模块、通信模块、组网模块与传播模块四部分。命令与控制模块依然是整个僵尸网络的核心，其实现命令的发布，认证与解析攻击者的命令，调用相应恶意行为模块执行该命令，并将执行结果反馈给攻击者。通信模块实现僵尸主机之间的通信，包括对消息的加密、对命令的路由和节点列表的交换等等。组网模块实现新感染节点加入网络的方式(随机扫描，连接父节点，连接预定义节点等等)和对新感染节点类别的判断(通过判断新感染节点 IP 是公网 IP 还是私有 IP，是静态还是动态，是否在防火墙之后，来决定新感染节点是作为服务者主机(Servent bots)还是作为客户主机(Client bots))[22]等功能。传播模块通过社会工程学、垃圾邮件、漏洞扫描以及传统的病毒或蠕虫等技术和方法将自身传播到尽可能多的主机上。

扩展功能模块主要包括下载与更新模块、躲避检测与对抗分析模块以及恶意行为模块等三部分。恶意行为模块是包括窃取主机敏感信息、记录主机按键、发动 DDoS 攻击、发送垃圾邮件等等恶意行为在内的功能模块，执行具体的命令，完成恶意攻击。下载与更新模块、躲避检测与对抗分析模块与文献[2]相同。

3.3　P2P 僵尸网络的分类

在文献[18]中，作者按照感染范围的不同，将 P2P 僵尸网络分为“parasite”，“leeching”和“bot-only” 3 类。在“parasite”类型中，P2P 僵尸网络利用现有的 P2P 网络传递命令和消息，且只感染 P2P 网络内部主机；在“leeching”类型中，P2P 僵尸网络也是利用现有网络和协议通信，但是可以感染网络外主机；在“bot-only”类型中，P2P 僵尸程序利用现有的或自制的协议组建新的 P2P 网络，可以感染互联网上所有主机。

这里，在参考 P2P 网络的拓扑结构分类[23]的基础上，结合 P2P 僵尸网络特点和当前的研究成果，将 P2P 僵尸网络分为纯 P2P 僵尸网络、集中式 P2P 僵尸网络、混合 P2P 僵尸网络和 P2P 僵尸网络群等四类。

3.3.1　纯 P2P 僵尸网络

在纯 P2P 僵尸网络中，不存在中央节点，所有的僵尸主机之间都是对等连接，僵尸主机既是服务器也是客户机。网络结构如图 3 所示。命令的传递、节点同步、二次感染代码的下载等等都通过这一对等网络实现。与 P2P 网络结构相对应，纯分布式 P2P 僵尸网络也可分为“无结构”和“有结构”两类。典型的“无结构”P2P 僵尸网络有 Sinit、Nugache 等，典型的“有结构”P2P 僵尸网络有 Peacomm 等。

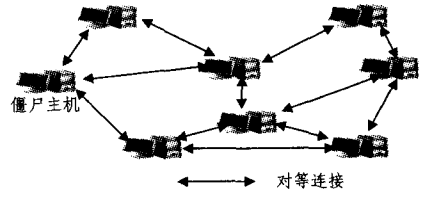

图 3　纯 P2P 僵尸网络拓扑结构

3.3.2　集中式 P2P 僵尸网络

集中式 P2P 僵尸网络是在纯 P2P 僵尸网络的基础上引入中央节点而构成的网络结构。如图 4 所示，网络中设置至少一台服务器来控制所有僵尸主机，协调全网主机的行动。各僵尸主机之间采取对等连接，服务器和僵尸主机之间采取不对等连接。服务器在不同的僵尸网络中发挥不同的功能，例如在 SpamThru 中，有一台控制服务器实现对网络的控制，另有多台邮件模板服务器供僵尸程序下载垃圾邮件模板；又如在 Phatbot 中，新感染主机需要连接到 Gnutella 网络的缓冲服务器获取僵尸网络节点列表。对于采取集中式 P2P 结

构的僵尸网络而言，仍然存在单点失效的问题，但是可以通过P2P网络加以弥补。如在SpamThru中，只要攻击者还掌握至少一个节点，他就能够将所有僵尸主机重新导向新的控制服务器；又如在Phatbot中，失去缓冲服务器并不会影响到僵尸网络的通信，只不过僵尸网络的规模不能再扩大了。典型的集中式P2P僵尸网络有Phatbot和SpamThru等。

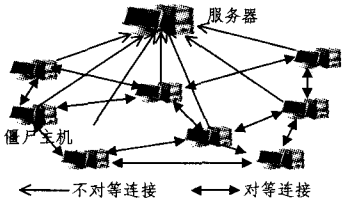

图4 集中式P2P僵尸网络拓扑结构

3.3.3 混合式P2P僵尸网络

混合式P2P僵尸网络是纯P2P僵尸网络和集中式P2P僵尸网络的结合体。目前Internet上尚未出现采取此种结构的P2P僵尸网络，不过Wang等人在文献[22]中介绍了一种被称为高级混合P2P僵尸网络的构造方法，如图5所示。在这种结构中，僵尸主机根据IP地址的类型被分为两类：servent主机和client主机。servent主机拥有静态IP地址，相互之间采取对等连接，client主机或因为IP地址动态分配，或因为拥有私有IP，或因为防火墙等因素不能够在Internet上被其他主机访问，servent主机和client主机之间采取不对等连接，一台servent主机可以连接多台client主机，但不能连接所有client，反过来，一台client主机可以连接多台servent主机，但同样不能连接所有servent主机。

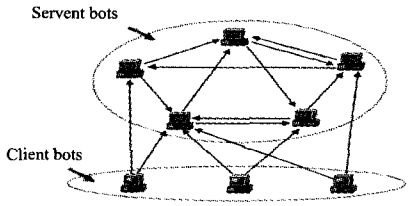

图5 混合P2P僵尸网络拓扑结构[22]

3.3.4 P2P僵尸网络群

Vogt等人提出了一种被称为“Super-Botnet”的僵尸网络结构[24]，如图6所示。在这种僵尸程序的传播过程中，形成了大批彼此相互隔离的小的僵尸网络。这些小僵尸网络采取P2P结构(也可以采取集中式C&C结构，图6所示为此类结构)，僵尸网络之间采取再感染的方式交换路由信息，将攻击者的命令与控制信息传递到所有僵尸网络，从而这个P2P僵尸网络群就构成了一个超级僵尸网络(Super-Botnet)。P2P僵尸网络群结构有网络流量小、不容易被发现的优点，我们认为僵尸网络群结构是未来僵尸网络发展的重要方向。

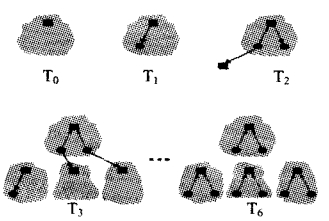

图6 Super-Botnet构造过程图[24]

4 P2P僵尸网络的工作过程

大体上来讲，P2P僵尸网络工作过程可以分为三个阶段[25]：感染阶段、组网与二次感染阶段和接受与执行命令阶段。

4.1 感染阶段

P2P僵尸网络必须拥有一定数量的僵尸主机才能有效地达到攻击目的，而使互联网上的主机变成僵尸主机的过程就是感染。图7为P2P僵尸网络感染阶段的示意图。感染的方式主要可分为两类：①手动感染方式，即攻击者利用系统漏洞(弱密码、远程溢出等)连接到主机A，将僵尸程序拷贝到主机上，完成感染。②自动感染方式，即将僵尸程序与传统病毒传播技术、蠕虫主动传播技术和恶意代码等技术相互融合，完成僵尸程序的自动传播。方式①主要存在于早期僵尸网络之中，在P2P僵尸网络中感染方式②更为常见。

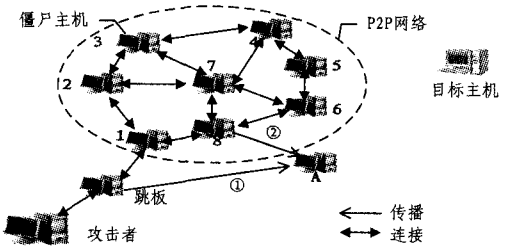

图7 P2P僵尸网络的感染阶段

4.2 组网与二次感染阶段

在这一阶段，新感染主机工作的大致步骤为：①通过各种方式(随机扫描、连接父节点、连接预定义节点等)发现并加入P2P网络，并向网络其他节点通知它的加入；②从相邻节点得到P2P网络全部或部分节点信息；③在网络中寻找并下载二次感染代码，完成自身的更新，进入等待命令状态。这时新感染主机才成为了真正意义上的P2P僵尸主机。例如，对于Peacomm[10]来说，僵尸程序在新感染节点上得以运行后，首先连接保存在程序当中的初始网络节点，一旦连接成功，Peacomm就从其他节点得到更多关于僵尸网络的信息。然后利用这些信息搜索并下载二次感染代码。Peacomm的二次感染代码主要包括程序更新、主机控制和执行恶意行为等功能。

4.3 接受与执行命令阶段

在完成组网与二次感染之后，僵尸主机进入接受与执行命令阶段。接受攻击者命令是此阶段首要任务，一般来讲分为“拉”方法(pull)和“推”方法(push)[25]。“拉”方法是指僵尸主机定期主动地从某一固定地址读取命令。而“推”方法是指僵尸主机被动地等待命令的到来，并且将命令传递到相邻节点。命令有唯一的标识符，以便僵尸主机确定是否执行过该命令。命令一般都需经过加密与认证，从而僵尸主机能够正确地判断命令的发出者是否是攻击者本人。一旦僵尸主机确定命令是新命令，并且确实是由攻击者发出，就调用恶意行为模块解析执行该命令，并将执行结果反馈给攻击者主机。如图8所示，攻击者发出攻击命令，命令通过跳板、主机1、主机8，传递到主机A，主机A判断是否为新命令以及命令发出者是否合法后，调用恶意行为模块开始执行命令。这时，主机A又从主机6接受到该命令，经判断，已经执行该命令，将命令丢弃。命令执行完毕后，主机A将执行结果经原路由反馈给攻击者。

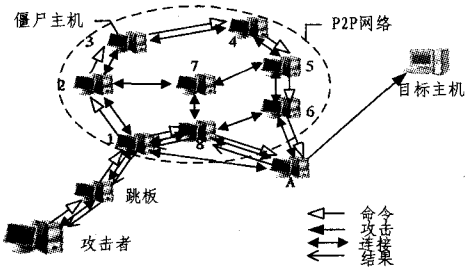

图 8　接受与执行命令阶段

5　P2P 僵尸网络的传播模型研究

P2P 僵尸网络的分布式结构降低甚至消除了传统 IRC 僵尸网络存在的单点失效的风险，因此对 P2P 僵尸网络的防御变得更加困难。研究 P2P 僵尸网络的传播规律，有助于加深对其的了解，从而探索出行之有效的防御方法。

Ruitenbeek 等人在深入研究 Storm 僵尸网络传播规律的基础上，利用 Möbius 软件构造了一个随机活动网络模型[11]。模型假设 P2P 僵尸主机在具有初始感染、联网主机、非活动传播主机、活动传播主机、非活动工作主机和活动工作主机等六种状态。随着时间的变化，僵尸主机能够以一定的概率从一种状态转化到另一种状态。仿真结果表明，当感染过程概率降低时，传播速度变慢了。另外，降低两个不同概率得到的两条曲线非常接近，表明反病毒手段可以在僵尸程序传播过程的任意阶段起作用。

另一方面，在保持概率不变的情况下，应逐步提高僵尸主机的免疫率。仿真结果表明，当免疫率较高时，可有效抑制僵尸网络增长，甚至缩小其规模。同时也可以看出，当非活动主机免疫率较低时，单纯提高活动主机免疫率也可以有效抑制僵尸程序传播，反过来也是一样。

尽管随机活动网络模型是从研究 Storm 僵尸网络出发，但是它却适用于一般的 P2P 僵尸网络。随机活动网络模型的不足是：没有考虑易感染主机的数量上限，因此不能准确描述 P2P 僵尸网络后期的传播情况。

6　P2P 僵尸网络的检测与防御研究

诸葛建伟等人在文献[2]中介绍了多种跟踪、检测和防御僵尸网络的方法，有些方法可以移植到对抗 P2P 僵尸网络上来。例如文献[26]中就介绍了如何使用蜜罐和蜜网技术[27]跟踪 P2P 僵尸网络的方法。同时 P2P 僵尸网络也有自身特殊性，因此也必然有其检测和防御的特殊方法。

6.1　P2P 僵尸网络的检测

Schoof 等人中提出了根据 P2P 僵尸网络独有特征进行检测的 4 种方法[19]。①基于开放的端口。P2P 僵尸主机之间的通信要求僵尸主机必须开放并时刻监听某些特定的端口，通过检测经过这些端口的数据流量或扫描这些端口就可能发现僵尸主机。这种方法的不足在于如果僵尸程序使用其它程序也在使用的端口时就会出现误报。②基于失败连接。由于防火墙的阻挡、网络地址转换或主机关机等因素，P2P 僵尸主机连接其他节点的失败率很高，网络上会出现大量的“目的地址不可达”ICMP 包和 TCP 重置包。这使得进出 P2P 僵尸主机的网络流量要远比正常主机大，通过流量分析就能够确定僵尸主机的位置。③基于节点发现。在某些 P2P 僵尸网络中，如 Nugache、Storm 等，僵尸程序保存着一张节点列表。我们可以对这些列表中的主机进行监视，检测到与它们连接的其它主机，进而检测到 P2P 僵尸网络的全部节点。④基于通信包检测(packet inspection)。有些 P2P 僵尸程序利用现有的 P2P 网络进行通信，与正常用户很难区别。深入分析可以发现，僵尸网络传递命令与控制信息呈现出通信次数多、占用带宽窄的特征，这与文件共享时的网络流量特征不同。在网络未使用加密的情况下，可以采用深入分析通信包的方法加以检测。

Kang 等人提出了一种检测 P2P 僵尸网络的 Multi-chat CUSUM 模型[28]。其主要思想是：P2P 僵尸网络在传播和工作过程中呈现出 3 种异常状态：①ICMP“目标不可达”包数量增多；②UDP 包数量增多；③SMTP 包数量增多。探测网络中的这 3 种通信包数量，作为输入，经过非参数 CUSUM 算法计算之后，结果与一个定值(由不同网络状态决定的常量)相比较，就可以确定网络中是否有僵尸主机存在。

Noh 等人提出了一种基于多相通信流模型(Multi-Phased Flow Model)的 P2P 僵尸网络检测方法[29]。该方法大致可分为 3 个步骤：首先将网络连接时间定义为 60 秒，在网络流中去除重传和失效的数据包，将剩下的数据流按照相似度合并；其次将合并后的数据流按照协议、端口和流量计算成为一个状态值；最后使用马尔可夫链计算数据流在各状态值之间的变化情况，并将结果与正常网络数据流进行比较，以此来判断是否存在 P2P 僵尸主机。经实验表明，该模型能够较好地检测出 SpamThru、Storm 和 Nugache 这三种僵尸网络，概率分别是 96.15%、100%和 95%，但是存在一定的误报率，概率为 2.88%。

6.2　P2P 僵尸网络的防御

Wang 等人提出了两种 P2P 僵尸网络对抗技术[25]。一种是索引投毒(index poisoning)技术。这种技术主要适用于基于索引的 P2P 僵尸网络(Index-based P2P botnet)，该僵尸网络利用索引来实现命令的发布与读取。索引投毒技术的主要思想就是在索引中插入假命令，当假命令的比例较大时，僵尸主机将以很大的可能性获得假命令，从而达到干扰僵尸主机对正确命令或文件的获取，阻扰僵尸网络命令与控制信道正常工作的目的。这种对抗技术的不足之处在于：①适用范围有限，对于不采用索引构造命令与控制信道的 P2P 僵尸网络就不起作用；②反对抗措施比较简单，例如采取动态变换索引的方法、对索引进行加密认证的方法等等。

另一种是 Sybil 攻击技术。其主要思想是：在 P2P 僵尸网络中插入 Sybil 节点，将命令路由到错误节点或监视与命令相关的通信等以破坏 P2P 僵尸网络的正常运行。不同的网络结构插入 Sybil 节点的方法也不同。在无结构 P2P 网络，例如 Gnutella 网络中，应该选择插入端节点作为 Sybil 节点；而在有结构 P2P 网络，如 Kad[30] 中，Sybil 节点的 ID 应该与已知的传递命令的索引哈希值接近。Sybil 攻击方法的缺点是插入 Sybil 节点需要物理的或虚拟的机器，花费比索引投毒要高。

Holz 等人针对 Storm 提出干扰节点攻击(eclipse attack)和污染攻击(Polluting)两种对抗措施[30]。干扰节点攻击是 Sybil 攻击的一种特殊形式，它们的实施方法是非常类似的，就是在被干扰节点附近部署一定数量的拥有与被干扰节点相

近哈希值的伪造节点。二者的主要区别在于干扰节点攻击只针对分布式哈希表(DHT)ID空间的很小一部分实施干扰。但是由于在Overnet中,命令是通过整个哈希值空间传播,而不是在某个关键字临近区域传播,干扰节点攻击效果并不理想。污染攻击的主要思想是发布大量的与攻击者命令哈希值相同的内容,以干扰P2P僵尸主机对命令的正常获取。实验表明,该方法对于常规查询算法(只在与命令哈希值接近的节点上查询)十分有效,但对于全节点查询算法(在P2P网络所有节点上查询)来说,僵尸主机依然能够正确获取攻击者命令。

结束语 以上主要回顾了P2P僵尸网络从出现到成熟的过程,提出了一种与以往研究成果不同的P2P僵尸网络功能结构,对P2P僵尸网络的分类方法进行了有益的尝试,又从一般意义上阐述了P2P僵尸网络的工作过程,并且介绍了当前学术界对P2P僵尸网络传播模型和检测、防御方法的研究进展。

可以预测,在未来一段时间内,为增强自身的隐蔽性和鲁棒性,P2P僵尸网络将可能采取如下的措施:①控制网络的规模,减少网络流量,以降低被发现的可能性;②使用强壮的认证及加密方法;③僵尸主机之间采用多种通信方法或通信协议。

加强对P2P僵尸网络的监测和研究,重点研究方向可从以下几方面着眼:①小型P2P僵尸网络的检测、鉴别与跟踪方法;②P2P僵尸网络传播模型的深入研究,如P2P僵尸网络规模受攻击者控制时的传播模型研究、存在对抗因素(人为因素、杀毒软件等)时的传播模型研究等等;③P2P僵尸网络防御与对抗方法研究。

参 考 文 献

[1] Kalt C. Internet relay chat: architecture[Z]. Request for Comments: RFC 2810. 2000

[2] 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

[3] Banday M, Qadri J, Shah N. Study of Botnets and their threats to Internet Security[J]. Sprouts: Working Papers on Information Systems, 9(24)

[4] Botnets W C. Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress[R]. CRS Report for Congress. 2008

[5] Grizzard J, Sharma V, Nunnery C. Peer-to-Peer Botnets: Overview and Case Study[C]// Proc. of the 1st Workshop on Hot Topics in Understanding Botnet(HotBots 2007). 2007

[6] Li J, Ehrenkranz T, Kuenning G, et al. Simulation and Analysis on the Resiliency and Efficiency of Malnets[C]// Proc. of the IEEE Symp. on Measurement, Modeling and Simulation of Malware(MMSM 2005). Monterey: IEEE Computer Society Press, 2005:262-269

[7] Arce I, Levy E. An Analysis of the Slapper Worm[J]. The IEEE Security & Privacy, 2003, 1(1):82-87

[8] Mukamurenzi N. Storm Worm: A P2P Botnet[J]. Master of Science in Communication Technology, February 2008

[9] Porras P, Saidi H, Yegneswaran V. A Multi-perspective Analysis of the Storm(Peacomm) Worm[R]. Computer Science Laboratory, SRI International, 2007

[10] Steggink M, Idziejczak I. Detection of peer-to-peer botnets[R]. Research report for System and Network Engineering, University of Amsterdam, Netherlands, 2007

[11] Ruitenbeek E, Sanders W. Modeling Peer-to-Peer Botnets[C]// Proc. of 5th International Conference on Quantitative Evaluation of Systems(QEST '08). September 2008:307-316

[12] Wang P, Aslam B, Zou C C. Peer-to-Peer Botnets: The Next Generation of Botnet Attacks[OL]. http://www.eecs.ucf.edu/~czou/ research/P2PBotnets-bookChapter.pdf

[13] 邢小良. P2P技术及其应用[M]. 北京:人民邮电出版社,2008

[14] Chien E. Malicious Threats of Peer-to-Peer Networking[R]. Symantec Whitepaper. 2003

[15] Liang J, Kumar R, Rose K. The FastTrack overlay: A measurement study[J]. Computer Networks, 2006, 50:842-858

[16] Stewart J. Sinit P2P Trojan Analysis [R/OL]. http://www.secureworks.com/research/threats/sinit

[17] Stewart J. Phatbot Trojan Analysis [R/OL]. http://www.secureworks.com/research/threats/phatbot

[18] Stewart J. SpamThru Trojan Analysis[R/OL]. http://www..secureworks.com/research/threats/spamthru

[19] Schoof R, Koning R. Detecting peer-to-peer botnets[R]. System and Network Engineering. University of Amsterdam, 2007

[20] Pinzon S. Understanding and Blocking the New Botnet [R]. Whatchguard whitepaper, April 2008

[21] Calvet J, Davis C, Bureau P. Malware Authors Don't Learn, and That's Good! [OL]. http://www.cs.mcgill.ca/~carlton/papers/waledac-article_ final.pdf

[22] Wang P, Sparks S, Zou CC. An Advanced Hybrid Peer-to-Peer Botnet[C]// Proc. of the 1st Workshop on Hot Topics in Understanding Botnets(HotBots 2007). 2007

[23] Yang B, Molina H G. Designing a Super-Peer Network[C]// International Conference on Data Engineering, Mar. 2003

[24] Vogt R, Aycock J, Jacobson M. Army of Botnets[C]// Proc. 14th Network and Distributed System Security Symp(NDSS '07). San Diego, CA, 2007:111-123

[25] Wang P, Wu L, Aslam B, et al. A Systematic Study on Peer-to-Peer Botnets[C]// Proc. of International Conference on Computer Communications and Networks(ICCCN). Aug. 2009

[26] Holz T. Tracking and Mitigation of Malicious Remote Control Networks[M]

[27] Spitzner L. Honeypots [OL]. http://www.tracking-hackers.com/papers/honeypots.html, 2003

[28] Kang J, Zhang J Y, Li Q, et al. Detecting New P2P Botnet with Multi-chart CUSUM[C]// International Conference on Networks Security, Wireless Communications and Trusted Computing. 2009

[29] Noh S, Oh J, Lee J, et al. Detecting P2P Botnets using a Multi-Phased Flow Model[C]// Digital Society, 2009, '09 Third International Conference on Digital Society. Feb. 2009:247-253

[30] Steiner M, En-Najjary T, Biersack E. A global view of kad[C]// Proc. of the ACM Internet Measurement Conf(IMC '07). 2007

[31] Holz T, Steiner M, Dahl F, et al. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on StormWorm[C]// Proc. of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats(LEET '08). April 2008