ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.3, March 2008, pp.702−715
DOI: 10.3724/SP.J.1001.2008.00702

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

僵尸网络研究*

诸葛建伟[1], 韩心慧[1], 周勇林[2], 叶志远[1], 邹 维[1+]

[1](北京大学 计算机科学技术研究所,北京 100871)

[2](国家计算机网络应急技术处理协调中心,北京 100029)

Research and Development of Botnets

ZHUGE Jian-Wei[1], HAN Xin-Hui[1], ZHOU Yong-Lin[2], YE Zhi-Yuan[1], ZOU Wei[1+]

[1](Institute of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

+ Corresponding author: Phn: +86-10-82529688, Fax: +86-10-82529207, E-mail: zouwei@icst.pku.edu.cn

Zhuge JW, Han XH, Zhou YL, Ye ZY, Zou W. Research and development of botnets. *Journal of Software*, 2008,19(3):702−715. http://www.jos.org.cn/1000-9825/19/702.htm

Abstract: Botnet is a novel attack strategy evolved from traditional malware forms; it provides the attackers stealthy, flexible and efficient one-to-many Command and Control mechanisms, which can be used to order an army of zombies to achieve the goals including information theft, launching distributed denial of service, and sending spam. Botnet has stepped into the expanding phase, and has been a serious threat to Internet security, especially in China mainland. In this paper, the evolution process, concept, functional structure and execution mechanism of botnet are presented, the Command and Control mechanisms and propagation model are discussed, and the latest techniques on botnet tracking, detection and prevention are reviewed. The developing trends of botnet and further topics in this area are also analyzed.

Key words: network security; botnet; malware; bot; propagation model

摘 要: 僵尸网络是一种从传统恶意代码形态进化而来的新型攻击方式,为攻击者提供了隐匿、灵活且高效的一对多命令与控制机制,可以控制大量僵尸主机实现信息窃取、分布式拒绝服务攻击和垃圾邮件发送等攻击目的.僵尸网络正步入快速发展期,对因特网安全已造成严重威胁,对中国大陆造成的危害尤为严重.介绍了僵尸网络的演化过程和基本定义,深入剖析了僵尸网络的功能结构与工作机制,讨论了僵尸网络的命令与控制机制和传播模型,并归纳总结了目前跟踪、检测和防御僵尸网络的最新研究成果,最后探讨了僵尸网络的发展趋势和进一步的研究方向.

关键词: 网络安全;僵尸网络;恶意代码;僵尸程序;传播模型

中图法分类号: TP393 **文献标识码**: A

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z445, 2006AA01Z410 (国家高技术研究发展计划(863)); the National Information Security Research Plan of China under Grant No.2006A30 (国家 242 信息安全计划); the Electronic Development Fund of the Ministry of Information Industry of China under Grant No.[2006]634 (信息产业部电子发展基金); the IBM Ph.D. Fellowship Plan (IBM 全球博士生英才计划)

Received 2007-06-21; Accepted 2007-09-04

僵尸网络(botnet)是攻击者出于恶意目的,传播僵尸程序控制大量主机,并通过一对多的命令与控制信道所组成的网络.僵尸网络是从传统恶意代码形态包括计算机病毒、网络蠕虫、特洛伊木马和后门工具的基础上进化,并通过相互融合发展而成的目前最为复杂的攻击方式之一.

由于为攻击者提供了隐匿、灵活且高效的一对多控制机制,僵尸网络得到了攻击者的青睐和进一步的发展,从而已成为因特网最为严重的威胁之一.利用僵尸网络,攻击者可以轻易地控制成千上万台主机对因特网任意站点发起分布式拒绝服务攻击,并发送大量垃圾邮件,从受控主机上窃取敏感信息或进行点击欺诈以牟取经济利益[1].

近年来,僵尸网络的活跃已经引起国外安全业界的充分重视:僵尸网络已成为安全领域的学术研究和讨论的热点问题,ACM 协会从 2003 年开始举办的 WORM 会议(Workshop on rapid malcode)和 USENIX 协会从 2005 年开始举办的 SRUTI 会议(Workshop on steps to reducing unwanted traffic in the Internet)均以僵尸网络为重要议题.此外,USENIX 协会从 2007 年开始举办僵尸网络专题探讨会 HotBots(Workshop on hot topics in understanding botnets).工业界和政府部门也同样关注僵尸网络对因特网所带来的严重安全威胁,微软公司在 2004 年发起了国际反僵尸网络工作组,2006 年 6 月,美国陆军研究办公室 ARO、国防高级研究计划署 DARPA 和国土安全部 DHS 等 3 个部门联合在 GA Tech 举办了僵尸网络专门研讨会,汇集学术界、政府部门和工业界的研究人员对这一新兴安全威胁进行了深入探讨,并汇总出版了《Botnet Detection: Countering the Largest Security Threat》[2].

Symantec 公司 2006 年监测数据表明[3,4],中国大陆被僵尸网络控制的主机数占全世界总数的比例从上半年的 20%增长到下半年的 26%,已超过美国,成为最大的僵尸网络受害国.但与此极不相称的是,国内对僵尸网络的关注和研究工作还较少.国家计算机网络应急技术处理协调中心在 2004 年底破获了国内第一起大规模的僵尸网络案件;北京大学计算机研究所在僵尸网络跟踪方面进行了长期而持续的研究[5−8];哈尔滨工业大学的孙彦东等人对僵尸网络安全威胁现状和研究进展进行了简要综述[9].

作为一种日趋严重的因特网安全威胁,僵尸网络已成为安全领域研究者所共同关注的热点,但目前国内外还尚未有详细而全面介绍僵尸网络机理和研究成果的综述论文.鉴于僵尸网络对国内因特网用户已造成的严重威胁,为深入理解僵尸网络机理和发展趋势,对僵尸网络的研究进展有一个总体把握,并促进国内在该方向上的研究,综述僵尸网络研究进展工作十分有意义.

本文阐述了僵尸网络的定义、功能结构与工作机制,并重点分析了僵尸网络的核心——命令与控制机制的不同实现方法,然后对僵尸网络的传播模型、僵尸网络跟踪、检测与防御技术的各个方面的主要研究工作进行了总体介绍,并对僵尸网络研究的发展趋势进行了展望.

1 僵尸网络的定义、功能结构与工作机制

1.1 僵尸网络的定义

僵尸网络是在网络蠕虫、特洛伊木马、后门工具等传统恶意代码形态的基础上发展、融合而产生的一种新型攻击方式.从 1999 年第一个具有僵尸网络特性的恶意代码 PrettyPark 现身因特网,到 2002 年因 SDbot 和 Agobot 源码的发布和广泛流传,僵尸网络快速地成为了因特网的严重安全威胁.第一线的反病毒厂商一直没有给出僵尸程序(bot)和僵尸网络的准确定义,而仍将其归入网络蠕虫或后门工具的范畴.从 2003 年前后,学术界开始关注这一新兴的安全威胁,为区分僵尸程序、僵尸网络与传统恶意代码形态,Puri 在文献[10]中及 McCarty 在文献[11]中均定义“僵尸程序为连接攻击者所控制 IRC 信道的客户端程序,而僵尸网络是由这些受控僵尸程序通过 IRC 协议所组成的网络”.为适应之后出现的使用 HTTP 或 P2P 协议构建命令与控制信道的僵尸网络,Bacher 等人[12]给出了一个更具通用性的定义:僵尸网络是可被攻击者远程控制的被攻陷主机所组成的网络.为了能够更加明确地区分僵尸网络和其他安全威胁,我们在文献[5]中强调了僵尸网络与其他攻击方式最大的区别特性在于攻击者和僵尸程序之间存在一对多的控制关系.Rajab 等人在文献[13]中也指出,虽然僵尸网络使用了其他形态恶意代码所利用的方法进行传播,如远程攻击软件漏洞、社会工程学方法等,但其定义特性在于对

控制与命令通道的使用.

综合上述分析,本文定义僵尸网络是攻击者(称为 botmaster)出于恶意目的,传播僵尸程序控制大量主机,并通过一对多的命令与控制信道所组成的网络.僵尸网络区别于其他攻击方式的基本特性是使用一对多的命令与控制机制.另外,我们的定义也强调了僵尸网络的恶意性以及具备的网络传播特性.

1.2 僵尸网络的演化过程

虽然僵尸网络这种新兴安全威胁在近些年才被学术界所关注,但它之前已经过了 10 余年的演化过程,表 1 给出了僵尸网络演化过程时间线.

Table 1 Timeline of botnet evolution

表 1 僵尸网络的演化过程

Date	Name	Author name/Nick	Description
12/1993	Eggdrop	Robey Pointer, Jeff Fisher, *et al*.	First non-malicious IRC bot
06/1999	PrettyPark	Anonymous	First malicious bot using IRC as C&C protocol
2000	GT-Bot	Sony, mSg and DeadKode	First widely spreading IRC bot based on mIRC executables and scripts
02/2002	SDbot	SD	First stand-alone IRC bot code base
09/2002	Slapper	Anonymous	First worm with P2P communications protocol
10/2002	Agobot	Ago	Incredibly robust, flexible, and modular design
09/2003	Sinit	Anonymous	Peer-to-Peer bot using random scanning to fund peers
03/2004	Phatbot	Ago	Peer-to-Peer bot based on WASTE
2004	Rbot/rxbot	Nils, RacerX90, *et al*.	Descendant of SDbot, most wildly distributed IRC bot code base
2004	Gaobot	Anonymous	Type I bot spreads through many approaches
05/2004	Bobax	Anonymous	Bot using HTTP based command and control mechanism

僵尸网络的历史渊源可以追溯到 1993 年因特网初期在 IRC 聊天网络中出现的 Bot 工具——Eggdrop,它实现为 IRC 聊天网络中的智能程序,能够自动地执行如防止频道被滥用、管理权限、记录频道事件等一系列功能,从而帮助 IRC 网络管理员更方便地管理这些聊天网络.

而之后,黑客受到良性 Bot 工具的启发,开始编写恶意僵尸程序对大量的受害主机进行控制,以利用这些主机资源达到恶意目的.1999 年 6 月,在因特网上出现的 PrettyPark 首次使用了 IRC 协议构建命令与控制信道,从而成为第一个恶意僵尸程序.之后,IRC 僵尸程序层出不穷,如在 mIRC 客户端程序上通过脚本实现的 GT-Bot、开源发布并广泛流传的 Sdbot、具有高度模块化设计的 Agobot 等,这使得 IRC 成为构建僵尸网络命令与控制信道的主流协议.为了让僵尸网络更具隐蔽性和韧性,黑客不断地对僵尸网络组织形式进行创新和发展,出现了基于 P2P 协议及 HTTP 协议构建命令与控制信道的僵尸程序,著名的案例包括传播后通过构建 P2P 网络支持 DDoS 攻击的 Slapper[14]、使用随机扫描策略寻找邻居节点的 Sinit、基于 WASTE 协议构建控制信道的 Phatbot 以及 2004 年 5 月出现的基于 HTTP 协议构建控制信道的 Bobax 等.

随着僵尸网络这种高效可控的攻击平台得到广泛的认同和使用,黑客也开始将传统的各类恶意代码技术融合到新型僵尸程序中,包括蠕虫主动传播技术、邮件病毒传播技术、Rootkit 隐藏技术、多态变形及对抗分析技术等,如 2004 年爆发的 Gaobot 和 Rbot,这种技术融合趋势使得僵尸网络的功能更加强大,传播渠道更加多样和隐蔽,也增加了防御者对僵尸网络进行发现、跟踪和防御的难度.

1.3 僵尸网络的功能结构

最早出现的 IRC 僵尸网络由僵尸网络控制器(botnet controller)和僵尸程序两部分组成.

由于 IRC 僵尸网络基于标准 IRC 协议构建其命令与控制信道,因此,其控制器可构建在公用 IRC 聊天服务器上,但攻击者为保证对僵尸网络控制器的绝对控制权,一般会利用其完全控制的主机架设专门的僵尸网络命令与控制服务器,最为常用的控制服务器架设软件是开源的 Unreal,其他的还包括 ConferenceRoom,ircu, bahamut,hybrid 等[8,12].

Barford 等人在分析了 GT-Bot,Sdbot,Agobot 和 Spybot 这 4 个主流 IRC 僵尸程序源码的基础上,提出了一种僵尸程序功能结构的分类方法[15],他们从僵尸网络体系结构(botnet architecture)、僵尸网络控制机制(botnet control mechanism)、僵尸主机控制机制(host control mechanism)、传播机制(propagation mechanisms)、破解和

攻击机制(exploits and attack mechanisms)、恶意代码样本分发机制(malware delivery mechanisms)、混淆机制(obfuscation mechanisms)和欺骗机制(deception mechanisms)这 7 个方面来描述和刻画每个僵尸程序所具有的功能特性.但是该分类方法并没有体现出对僵尸程序各功能模块的清晰划分,如将网络传播过程中的远程主机漏洞破解攻击以及利用受控主机发起的分布式拒绝服务攻击都归入了破解和攻击机制,而这两者显然具有不同的功能和实现.

我们在文献[15−17]的基础上,参考文伟平等人对网络蠕虫的功能结构分析[18],进一步通过对目前主流僵尸程序的总结,提出了如图 1 所示的僵尸程序功能结构.僵尸程序的功能模块可以分为主体功能模块和辅助功能模块,主体功能模块包括了实现僵尸网络定义特性的命令与控制模块和实现网络传播特性的传播模块,而包含辅助功能模块的僵尸程序则具有更强大的攻击功能和更好的生存能力.

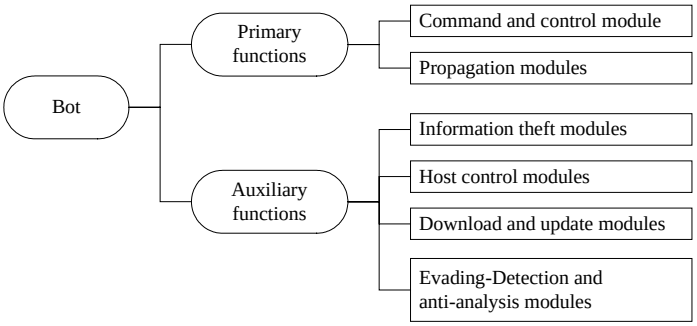

Fig.1 Functional structure of bots

图 1 僵尸程序的功能结构

主体功能模块中的命令与控制模块作为整个僵尸程序的核心,实现与僵尸网络控制器的交互,接受攻击者的控制命令,进行解析和执行,并将执行结果反馈给僵尸网络控制器.传播模块通过各种不同的方式将僵尸程序传播到新的主机,使其加入僵尸网络接受攻击者的控制,从而扩展僵尸网络的规模.僵尸程序可以按照传播策略分为自动传播型僵尸程序和受控传播型僵尸程序两大类[13],而僵尸程序的传播方式包括通过远程攻击软件漏洞传播、扫描 NetBIOS 弱密码传播、扫描恶意代码留下的后门进行传播、通过发送邮件病毒传播、通过文件系统共享传播等.此外,最新的僵尸程序也已经开始结合即时通信软件和 P2P 文件共享软件进行传播.

辅助功能模块是对僵尸程序除主体功能外其他功能的归纳,主要包括信息窃取、僵尸主机控制、下载与更新、躲避检测与对抗分析等功能模块:

① 信息窃取模块用于获取受控主机信息(包括系统资源情况、进程列表、开启时间、网络带宽和速度情况等),以及搜索并窃取受控主机上有价值的敏感信息(如软件注册码、电子邮件列表、帐号口令等);

② 僵尸主机控制模块是攻击者利用受控的大量僵尸主机完成各种不同攻击目标的模块集合,目前,主流僵尸程序中实现的僵尸主机控制模块包括 DDoS 攻击模块、架设服务模块、发送垃圾邮件模块以及点击欺诈模块等;

③ 下载与更新模块为攻击者提供向受控主机注入二次感染代码以及更新僵尸程序的功能,使其能够随时在僵尸网络控制的大量主机上更新和添加僵尸程序以及其他恶意代码,以实现不同攻击目的;

④ 躲避检测与对抗分析模块,包括对僵尸程序的多态、变形、加密、通过 Rootkit 方式进行实体隐藏,以及检查 debugger 的存在、识别虚拟机环境、杀死反病毒进程、阻止反病毒软件升级等功能,其目标是使得僵尸程序能够躲避受控主机的使用者和反病毒软件的检测,并对抗病毒分析师的分析,从而提高僵尸网络的生存能力.

HTTP 僵尸网络与 IRC 僵尸网络的功能结构相似,所不同的仅仅是 HTTP 僵尸网络控制器是以 Web 网站方式构建.而相应地,僵尸程序中的命令与控制模块通过 HTTP 协议向控制器注册并获取控制命令.

由于 P2P 网络本身具有的对等节点特性,在 P2P 僵尸网络中也不存在只充当服务器角色的僵尸网络控制

器,而是由 P2P 僵尸程序同时承担客户端和服务器的双重角色.P2P 僵尸程序与传统僵尸程序的差异在于其核心模块——命令与控制模块的实现机制不同,如 Phatbot 僵尸程序是在基于 IRC 协议构建命令与控制信道的 Agobot 基础上,通过采用 AOL 的开源 P2P 协议 WASTE 重新实现其命令与控制模块,从而可以构建更难跟踪和反制的 P2P 僵尸网络.

一些流行和最新出现僵尸程序的功能模块统计情况见表 2,其中包括了经典的 IRC 僵尸程序如 Sdbot, Agobot,GT-Bot 和 Rbot 等,近年来流行的 HTTP 僵尸程序如 Bobax,Rustock[19]和 Clickbot[20]等,以及 P2P 僵尸网络 Phatbot 等.

Table 2 Funcation modules of some popular and latest bots

表 2 一些流行和最新出现僵尸程序的功能模块统计情况

Bot	Version	Command & control module	Propagation modules	Information theft modules	Host control modules	Download and update modules	Evading detection and anti-analysis modules
SDbot	v0.5b	Lightweight version of IRC	N/A*	Sysinfo; cdkeys	Udp/Icmp flood; deploy servers; execute command	Download update	N/A
Agobot	v4.0	Derivative of IRC	DCOM/ Dameware/ Radmin Bagle/Mydoom/ NetBIOS/ MS-SQL	Sysinfo; network bandwith & speed; host uptime; software keys; email list;	Generic DdoS module; PC control; autostart control; send spam	ftp.download http.visit http.update http.download	Polymorphism encoding strategies; test for debuggers and vmware; killing AV processes and disabling AV auto-updating
GT-Bot	with-dcom	IRC	RPC-DCOM	Sysinfo	Udp/Syn flood; execute command	N/A	N/A
Rbot	Rbot.A	IRC	NetBIOS/LSASS/ WebDav/Dcom/ MS-SQL/uPnP/ Dameware/WKS/ WINS/Beagle/ Mydoom...	Sysinfo; software keys; sniff passwords;	Deploy servers; send spam; generic DDoS module	Dowload	Encrypted with packers; killing AV processes and disabling AV auto-updating
Bobax	Bobax.A	HTTP	MS04-011 LSASS	Network speed	Execute command; send spam;	Update	N/A
Phatbot	Phatbot.A	WASTE	DCOM/DCOM2 Mydoom/Beagle Dameware/ NetBIOS/ MS-SQL WebDav/CPanel WKS/UPnP	Sysinfo; software keys; email list; sniff passwords	Generic DDoS module; deploy servers; PC control; autostart control; send spam;	ftp.download http.visit http.update http.download	Polymorphism encoding strategies; kill other malware (MSBlast, Welchia, Sobig.F); killing AV processes and disabling AV auto-updating
Rustock	Rustock.B	Encrypted HTTP	MS06-042 exploit	N/A	Send spam; opens a covert proxy	Download	Rootkit; multiple levels of obfuscation; use of RC4 encrypted C&C
Clickbot	Clickbot.A	HTTP	Trojan horse; distribute using existing botnets	N/A	Click fraud execute command	Get_Update	Implemented as a BHO

1.4 僵尸网络的工作机制

IRC 僵尸网络的工作机制如图 2 所示[14]:① 攻击者通过各种传播方式使得目标主机感染僵尸程序;② 僵尸程序以特定格式随机产生的用户名和昵称尝试加入指定的 IRC 命令与控制服务器;③ 攻击者普遍使用动态域名服务将僵尸程序连接的域名映射到其所控制的多台 IRC 服务器上,从而避免由于单一服务器被摧毁后导致整个僵尸网络瘫痪的情况;④ 僵尸程序加入到攻击者私有的 IRC 命令与控制信道中;⑤ 加入信道的大量僵尸程序监听控制指令;⑥ 攻击者登陆并加入到 IRC 命令与控制信道中,通过认证后,向僵尸网络发出信息窃取、僵尸主机控制和攻击指令;⑦ 僵尸程序接受指令,并调用对应模块执行指令,从而完成攻击者的攻击目标.

* SDbot 源码作者为防止 SDbot 被滥用,其开源发布源码包中没有包含网络传播模块,因此不能用于构建真正的僵尸网络,但 SDbot 源码的广泛发布仍对僵尸网络的日益泛滥起到了非常大的促进作用.

其他新型僵尸网络的工作机制与 IRC 僵尸网络类似,主要差异在于命令与控制机制的不同.

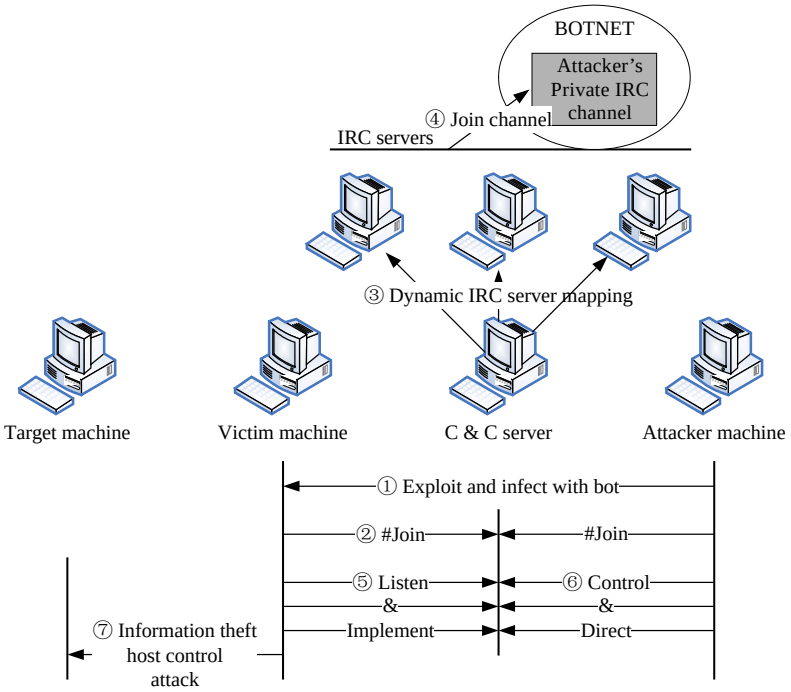

Fig.2 Execute mechanisms of IRC botnets[14]

图 2 IRC 僵尸网络的工作机制[14]

2 僵尸网络的命令与控制机制

僵尸网络的基本特性是使用一对多的命令与控制机制,因此,理解命令与控制机制的实现是深入了解僵尸网络机理的必要前提.当前主要使用的僵尸网络命令与控制机制包括:基于 IRC 协议的命令与控制机制,基于 HTTP 协议的命令与控制机制,以及基于 P2P 协议的命令与控制机制这 3 大类.

2.1 基于IRC协议的命令与控制机制

IRC 协议是因特网早期就广泛使用的实时网络聊天协议,它使得世界各地的因特网使用者能够加入到聊天频道中进行基于文本的实时讨论,根据 IRC 协议规范 RFC 2810[21]:"IRC 协议基于客户端-服务器模型,用户运行 IRC 客户端软件连接到 IRC 服务器上,IRC 服务器可以通过互相连接构成庞大的 IRC 聊天网络,并将用户的消息通过聊天网络发送到目标用户或用户群".IRC 网络中最为普遍使用的一种通信方式是群聊方式,即多个 IRC 客户端连接到 IRC 网络并创建一个聊天信道,每个客户端发送到 IRC 服务器的消息将被转发给连接这个信道的全部客户端.此外,IRC 协议也支持两个客户端之间的私聊方式.

由于 IRC 协议提供了一种简单、低延迟、匿名的实时通信方式,而且,它也被黑客普遍使用于相互间的远程交流,因此在僵尸网络发展初期,IRC 协议自然成为了构建一对多命令与控制信道的主流协议.

基于 IRC 协议,攻击者向受控僵尸程序发布命令的方法有 3 种:设置频道主题(TOPIC)命令,当僵尸程序登录到频道后立即接收并执行这条频道主题命令;使用频道或单个僵尸程序发送 PRIVMSG 消息,这种方法最为常用,即通过 IRC 协议的群聊和私聊方式向频道内所有僵尸程序或指定僵尸程序发布命令;通过 NOTICE 消息发送命令,这种方法在效果上等同于发送 PRIVMSG 消息,但在实际情况中并不常见.

IRC 僵尸网络中发送的命令可以按照僵尸程序对应实现的功能模块分为僵尸网络控制命令、扩散传播命令、信息窃取命令、主机控制命令和下载与更新命令.其中,主机控制命令还可以细分为发动 DDoS 攻击、架设

服务、发送垃圾邮件、点击欺诈等.一条典型的扩散传播命令[8],如“.advscan asn1smb 200 5 0 -r -a -s”,其中,最先出现的点号称为命令前缀,advscan 则为命令字,扩散传播命令的参数一般包括远程攻击的漏洞名、使用的线程数量、攻击持续时间、是否报告结果等.

2.2 基于HTTP协议的命令与控制机制

HTTP 协议则是近年来除 IRC 协议外的另一种流行的僵尸网络命令与控制协议,与 IRC 协议相比,使用 HTTP 协议构建僵尸网络命令与控制机制的优势包括两方面:首先,由于 IRC 协议已经是僵尸网络主流控制协议,安全业界更加关注监测 IRC 通信以检测其中隐藏的僵尸网络活动,使用 HTTP 协议构建控制信道则可以让僵尸网络控制流量淹没在大量的因特网 Web 通信中,从而使得基于 HTTP 协议的僵尸网络活动更难以被检测;另外,大多数组织机构在网关上部署了防火墙,在很多情况下,防火墙过滤掉了非期望端口上的网络通信,IRC 协议使用的端口通常也会被过滤,而使用 HTTP 协议构建控制信道一般都可以绕过防火墙.

目前,已知的采用 HTTP 协议构建命令与控制机制的僵尸程序有 Bobax,Rustock[19],Clickbot[20]等.例如,Bobax 僵尸程序,它首先会访问类似“http://hostname/reg?u=ABCDEF01&v=114”的一个 URL,向僵尸网络控制器发送注册请求,如果连接成功,则僵尸网络控制器将反馈这一请求,并在返回内容中包含当前攻击者对僵尸网络发出的控制命令,Bobax 僵尸程序则从返回内容中解析出命令并进行执行,Bobax 僵尸程序接受的命令包括:upd(下载并执行更新程序)、exe(执行指定的程序)、scn(使用 MS04-011 破解程序扫描并感染主机)、scs(停止扩散扫描)、prj(发送垃圾邮件)、spd(报告网络连接速度)等.

2.3 基于P2P协议的命令与控制机制

基于 IRC 协议和 HTTP 协议的命令与控制机制均具有集中控制点,这使得这种基于客户端-服务器架构的僵尸网络容易被跟踪、检测和反制,一旦防御者获得僵尸程序,他们就能很容易地发现僵尸网络控制器的位置,并使用监测和跟踪手段掌握僵尸网络的全局信息,通过关闭这些集中的僵尸网络控制器也能够比较容易地消除僵尸网络所带来的威胁.为了让僵尸网络更具韧性和隐蔽性,一些新出现的僵尸程序开始使用 P2P 协议构建其命令与控制机制.

Grizzard 等人在文献[22]中对 P2P 僵尸网络的发展历程进行了综述,Slapper,Sinit,Phatbot,SpamThru, Nugache 和 Peacomm 等出现的 P2P 僵尸网络实现了各种不同的 P2P 控制机制,并体现出一些先进的设计思想.为了消除容易被防御者用于摧毁僵尸网络的 bootstrap 过程,第一个构建 P2P 控制信道的恶意代码 Slapper 在网络传播过程中对每个受感染主机都建立了一个完整的已感染节点列表[14];Sinit 同样也消除了这一过程并使用了公钥加密进行更新过程的验证;Nugache 则试图通过实现一个加密混淆的控制信道来躲避检测.但这些已有 P2P 僵尸程序的控制协议的设计并不成熟[23]:Sinit 僵尸程序使用了随机扫描的方法寻找可交互的其他 Sinit 僵尸程序,这导致构造的 P2P 僵尸网络连接度非常弱,并且由于大量的扫描流量而容易被检测;Phatbot 在其 bootstrap 过程中利用了 Gnutella 的缓冲服务器,这也使得构建的僵尸网络容易被关闭.此外,Phatbot 所基于的 WASTE 协议在大规模网络中的扩展性并不好;Nugache 的弱点在于其 bootstrap 过程中对一个包含 22 个 IP 地址的种子主机列表的依赖;Slapper 并没有实现加密和通信认证机制,使僵尸网络很容易被他人所劫持.另外,Slapper 的已感染节点列表中包含了组成僵尸网络所有僵尸程序的信息,这使得防御者从一个捕获的程序中即可获得僵尸网络的全部信息.最后 Slapper 复杂的通信机制产生了大量网络流量,使其很容易引起网络流分析工具的警觉[14].

Wang 等人在文献[23]中提出了一种更加先进的混合型 P2P 僵尸网络命令与控制机制的设计框架,在此框架中,僵尸程序被分为两类:拥有静态 IP 地址并从因特网可以访问的僵尸程序称为 *servent bots*,这类僵尸程序承担客户端和服务器的双重角色;其他由于 IP 地址动态分配、私有 IP 或防火墙过滤等原因无法从因特网访问的僵尸程序称为 *client bots*,每个节点的邻居节点列表中只包含 *servent bots*.僵尸网络控制者通过认证机制后,可从网络中的任意节点注入其控制命令,当一个节点获取新的控制命令后,通过向其邻居节点转发,从而快速传递到每个 *servent bot*,*client bot* 则从其邻居节点列表中的 *servent bots* 获取控制命令.在此设计框架基础上,Wang

等人还进一步提出了通过命令认证、节点对加密机制、个性化服务端口等机制保证僵尸网络的健壮性和韧性.

Vogt 等人[24]则提出了一种层叠化的“super-botnets”僵尸网络群构建方式,即在僵尸网络的传播过程中不断分解以保证对僵尸网络规模的限制,并通过小型僵尸网络间邻居节点关系和基于公钥加密的通信机制构造僵尸网络群.

2.4 各种命令与控制机制的效率和韧性评价

评价僵尸网络命令与控制机制的两个重要指标是效率(efficiency)和韧性(resiliency)[25],僵尸网络命令与控制机制的效率关注僵尸网络控制者能以多快的速度把攻击命令传递到所有受控僵尸程序,从而有效完成其攻击目的.由于命令分发的速度与两个恶意节点间的距离直接相关,因此,僵尸网络的直径可以作为评价其控制机制效率的分析参数.僵尸网络的韧性则关注随机清除若干恶意节点后僵尸网络能够保持的最大连接度.

对于由 IRC 协议和 HTTP 协议构建的集中式命令与控制机制,僵尸网络控制点与每个受控僵尸主机均直接连接,因此其最大直径为 2,注入控制命令的僵尸网络控制者连接节点与其他主机的距离均为 2,因此,集中式僵尸网络的效率非常高.但如果僵尸网络控制点被防御者清除,就会导致僵尸网络被完全分解,因此,集中式僵尸网络的韧性很弱.

Li 等人在文献[25]中对随机网络方式构建的僵尸网络(如 Sinit 等)、小世界模型僵尸网络和类 Gnutella 僵尸网络(如 Phatbot 等)的效率和韧性进行了仿真分析.3 种不同 P2P 类型构建僵尸网络中的最大直径和平均直径与网络包含节点数之间的关系严格符合对数函数趋势.在平均节点度为 4 的情况下,随机网络方式构建的 100 万节点僵尸网络的最大直径为 17,平均直径为 11,考虑到因特网上两台主机间的平均延迟和拥塞等因素,僵尸网络控制者可以在少于 6 分钟的时间内将一个 1M 字节的攻击代码扩散到全部 100 万节点上;小世界模型僵尸网络中的最大直径则为 90,平均直径为 39;类 Gnutella 僵尸网络中 ultrapeer 节点间的最大直径为 4.8,叶子节点间的最大直径为 6.8,而平均直径则为 4.这 3 种不同 P2P 类型僵尸网络韧性仿真分析结果显示:对于随机网络方式构建的僵尸网络,当节点连接度为 2 时,随机清除 10%的节点后,网络连接度就下降为大约 43.3%;一旦节点连接度增大到 4,网络连接度就仅下降到 99.8%;当节点连接度为 10,即使 80%的节点被清除,剩余节点的网络连接度还能够达到 70%.这说明以随机网络方式构建的僵尸网络的韧性很好;而小世界模型僵尸网络中一旦 10%的节点被清除,其网络连接度就开始显著下降;而清除 70%~80%节点将完全分割僵尸网络.仿真分析结果显示,类 Gnutella 僵尸网络的韧性比两者更好,即使清除 75%节点后剩余节点还保持完全连接;而清除 87.5%节点后网络连接度仍保持大约 97%.

从 3 种不同 P2P 类型僵尸网络的效率和韧性仿真分析结果的比较可以看出,小世界模型僵尸网络虽然很容易构建,但其效率和韧性都比不上随机网络和类 Gnutella 类型,因此并不适合作为僵尸网络的命令与控制机制的实现方式;虽然随机网络方式构建的僵尸网络的效率和韧性都处于良好的水平,但类 Gnutella 方式构建的僵尸网络能够达到更高的效率和更好的韧性.虽然这种方式对于黑客而言较难构建,但已存在的类 Gnutella 方式的 P2P 网络为他们构建高效率且高鲁棒性的僵尸网络奠定了基础.

3 僵尸网络的传播模型研究

僵尸网络的传播特性是从网络蠕虫继承而来,但又与传统网络蠕虫的自动传播不同,僵尸网络的传播扩散一般是受控的,同时,在传播扩散目标的选择上遵循同子网地址优先的策略.网络蠕虫的传播模型适合采用传染病模型,已提出的网络蠕虫传播模型包括最基本的 SI 模型[26],SIR 模型[27],以及考虑蠕虫反制机制的双因素模型[28]等.由于僵尸网络传播所具有的受控性和区域性,网络蠕虫的传播模型并不适合用于对僵尸网络发展趋势的刻画和预测,因此,研究者开始研究适应僵尸网络传播特性的模型.

考虑到计算机在夜间关机下线后进入非易感状态的因素以及僵尸网络感染过程中存在的区域性偏好因素,Dagon 等人在文献[29]中提出了一个基于时区的僵尸网络传播模型.

首先,考虑在同一时区中的封闭网络,传播模型的微分方程为

$$\begin{cases} \mathrm{d}I(t)/\mathrm{d}t = \beta I'(t)S'(t) - \mathrm{d}R(t)/\mathrm{d}t \\ S(t) = N(t) - I(t) - R(t) \\ I'(t) = \alpha(t)I(t) \\ S'(t) = \alpha(t)S(t) \\ \mathrm{d}R(t)/\mathrm{d}t = \gamma I'(t) \end{cases} \tag{1}$$

式(1)中,$N(t)$表示 t 时刻该时区内原始易感主机的总数;$I(t)$为 t 时刻被感染主机数,$I'(t)$为 t 时刻在线的被感染主机数;$\alpha(t)$定义为 *diurnal shaping function*,即 t 时刻该时区计算机在线的比率,该比率根据观察统计确定,一般在白天达到峰值,而在深夜中由于大部分计算机的下线达到低谷;$S(t)$为 t 时刻易感主机数,而 $S'(t)$为在线易感主机数;$R(t)$为被免疫的主机数;感染率$\beta=\eta/\Omega$[30],其中,η为恶意代码的扫描率,而Ω为恶意代码扫描的 IP 地址空间的大小;γ定义为免疫比率.根据式(1)可以导出僵尸网络每日传播模型微分方程为

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \beta\alpha^2(t)I(t)[N(t) - I(t) - R(t)] - \gamma\alpha(t)I(t) \tag{2}$$

进一步扩展到多时区模型,最终得到的基于时区的僵尸网络传播模型方程为

$$\frac{\mathrm{d}I_i(t)}{\mathrm{d}t} = \alpha_i(t)(N_i(t) - I_i(t) - R_i(t)) \cdot \sum_{j=1}^{K} \beta_{ji}\alpha_j(t)I_j(t) - \gamma_i\alpha_i(t)I_i(t) \tag{3}$$

其中,$N_i(t)$,$I_i(t)$,$R_i(t)$分别为第 i 个时区中易感主机总数、已感主机数和免疫主机数;$\alpha_i(t)$为该时区中的上线计算机比率函数;β_{ji}为从时区 j 到时区 i 的感染率;γ_i为时区 i 的免疫率.

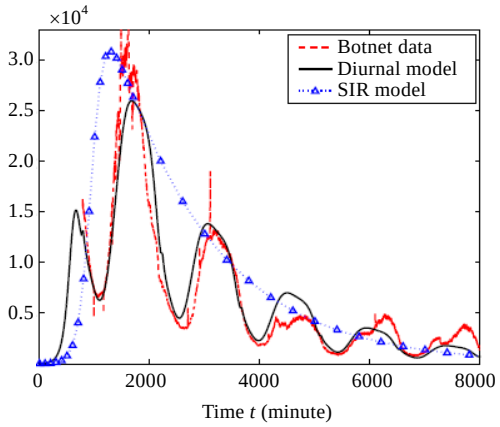

Fig.3 Comparison of diurnal and SIR model with botnet traffic[29]

图 3 基于时区的传播模型与 SIR 模型对实际僵尸网络传播数据的符合度比较[29]

通过对实际僵尸网络规模发展数据的观察和统计,如图 3 所示,基于时区的传播模型较传统 SIR 传播模型能够更好地符合因特网上僵尸网络的实际传播规律.但该模型只关注了传统远程攻击漏洞的传播方式,而当前的僵尸网络也融合了邮件病毒、即时通信和 P2P 文件共享软件等其他传播方式,对这些新型传播方式进行准确刻画的传播模型还有待进一步研究.此外,基于不同 P2P 协议的僵尸网络结构对其传播趋势的影响也是需要深入探讨的问题.

为了能够在实际环境中测试和验证僵尸网络的工作机制和传播模型,隔离的实验测试环境是必不可少的.Barford 等人提出了僵尸网络实际测试环境(botnet evaluation environment,简称 BEE)的设计方案[31],通过创建僵尸程序镜像文件库、部署 DHCP,DynDNS,IRC 等支撑服务,实现测试环境安全控制策略,从而构建一个可用于大规模僵尸网络实际测试的隔离的自包含网络环境.

4 僵尸网络的跟踪、检测与防御研究

僵尸网络已成为目前因特网最为严重的安全威胁之一,同时,僵尸网络本身具有的特性也使其成为黑客们用于 DDoS 攻击、发送垃圾邮件、窃取敏感信息等各种攻击行为的高效平台.为了应对僵尸网络的安全威胁,研究者已在僵尸网络跟踪、检测与防御等多方面开展了深入的研究工作,下面我们将讨论近年来开展的相关研究工作.

4.1 僵尸网络跟踪研究

充分了解僵尸网络的内部工作机理,是防御者应对僵尸网络安全威胁的前提条件.僵尸网络跟踪(botnet

tracking)为防御者提供了一套可行的方法,其基本思想是,首先通过各种途径获取因特网上实际存在的僵尸网络命令与控制信道的相关信息,然后模拟成受控的僵尸程序加入僵尸网络中,从而对僵尸网络的内部活动进行观察和跟踪.

最早开展僵尸网络跟踪研究工作的团队是德国蜜网项目组[12,32],Bacher 和 Holz 等人通过部署包含有 Windows 蜜罐主机的第二代蜜网捕获了因特网上实际传播的大量僵尸程序,然后使用 snort_inline 分析出僵尸程序所连接的 IRC 命令与控制信道信息,包括 IRC 服务器的域名/IP 和端口号、连接 IRC 服务器的密码(可选)、僵尸程序用户标识和昵称的结构、加入的频道名和可选的频道密码,然后使用 IRC 客户端追踪工具 drone 根据控制信道信息加入到僵尸网络进行跟踪.他们在大约 4 个月的时间内对超过 100 个僵尸网络进行了持续的跟踪观察,向安全业界第一次系统地展示了僵尸网络的内部工作机制.德国蜜网项目组还进一步研究并开发了基于低交互式蜜罐技术的恶意代码捕获器 Nepenthes[33],从而支持大规模的僵尸程序样本采集和进一步的僵尸网络跟踪.

Johns Hopkins 大学的 Rajab 等人进一步提出了一个多角度同时跟踪大量实际僵尸网络的方法[13],包括旨在捕获僵尸程序的分布式恶意代码采集体系、对实际僵尸网络行为获取内部观察的 IRC 跟踪工具以及评估僵尸网络全局传播足迹的 DNS 缓冲探测技术.在此方法基础上,他们在 3 个月期间跟踪了 192 个 IRC 僵尸网络,并通过对多角度获取数据的关联分析展示了僵尸网络的一些行为和结构特性.

Rajab 等人在进一步工作[34]中对“僵尸网络的规模监测和估计”这一重要问题进行了细致的探讨.他们认为,目前研究领域并没有对“僵尸网络规模”给出严格定义,因此导致了一些误解和不一致,而监测并估计“僵尸网络规模”必须阐明所使用的计数方法,给出明确的解释并说明监测过程相关的上下文信息.他们将僵尸网络生命周期内任意时间点上感染的全部僵尸主机数量定义为僵尸网络的全局足迹(footprint),而将特定时间点接受僵尸网络命令与控制信道控制的在线僵尸主机数量定义为僵尸网络的实时规模(live population).根据所使用信息的不同,对僵尸网络的规模估计方法分为通过内部视图信息计数和通过外部信息估计两大类,通过内部视图信息对僵尸程序计数,又包含有内部渗透和控制服务器 DNS 劫持两种技术手段;通过外部信息只能粗略地估计僵尸网络的全局足迹,可使用的技术手段包括 DNS 缓冲探测[13]和 DNS 黑名单探测[35]等.

根据公开资料,国内在僵尸网络跟踪方面的研究工作并不多,主要的工作集中在北京大学计算机研究所和国家计算机网络应急技术处理协调中心(CNCERT/CC).CNCERT/CC 在 2004 年底通过跟踪技术手段破获了国内第一起大规模的僵尸网络案件.诸葛建伟等人在文献[5]中介绍了发现和跟踪大量僵尸网络的方法,并通过大量的僵尸网络发现和跟踪经验给出了控制服务器所属国、僵尸网络规模的分布统计;进一步研究并开发了基于高交互式蜜罐技术的恶意代码自动捕获器 HoneyBow,并结合 Nepenthes,HoneyBow 以及第三代蜜网技术构建了更加全面的恶意代码自动捕获体系以支持大规模的僵尸网络跟踪[7];并对 1 961 个实际僵尸网络的活动行为记录进行了深入调查和分析,给出了僵尸网络捕获趋势、控制服务器分布、僵尸网络规模与被控主机分布、僵尸网络各种攻击行为的分析结果[8].

僵尸网络跟踪方法的优势在于能够全方位地了解僵尸网络的控制服务器位置、行为特性和结构特性,为防御者进一步检测与处置僵尸网络提供了充分的信息支持.存在的不足包括:① 基于蜜罐技术的采集和跟踪方法无法有效地检测出全部活跃的僵尸网络,无法为因特网用户提供直接保护;② 僵尸网络控制者在充分认识跟踪方法后,可以采取信息裁减机制、更强的认证机制等方法加大僵尸网络跟踪方法的难度,并减少跟踪所能够获取的信息.此外,各种基于 HTTP 协议和基于 P2P 协议的僵尸网络命令与控制机制的使用为僵尸网络跟踪带来了较大困难;③ 防御者对僵尸网络实施跟踪一旦被发现,就很可能被僵尸网络控制者实施 DDoS 攻击.

4.2 僵尸网络的检测方法研究

在利用跟踪方法了解僵尸网络内部工作机制的基础上,近两年来,研究者开始探索在业务网络中识别僵尸网络安全威胁的检测方法.

Binkley 等人提出了一个基于 TCP 扫描权重(TCP work weight)的启发式异常检测算法以检测 IRC 僵尸网络控制通信[36,37].该算法基于 IRC 僵尸网络中大量僵尸主机连接到同一 IRC 频道,并接受网络传播命令进行大

量的 TCP SYN 扫描这一观察,按照式(4)定义 TCP 扫描权重这一评价指标,并通过识别 TCP 扫描权重超出正常阈值的被感染 IP 地址及其连接的 IRC 频道对僵尸网络进行检测.

$$w=(S_s+F_s+R_r)/T_{sr} \quad (4)$$

其中,S_s 为发送的 SYN 包和 SYN|ACK 包数量,F_s 为发送的 FIN 包数量,R_r 为接收的 RESET 包数量,T_{sr} 为全部 TCP 数据包数量,TCP 扫描权重 w 为 TCP 控制报文数与总 TCP 报文数的比重.Binkley 等人提出的这种方法只适用于明文方式传播控制信道命令的 IRC 僵尸网络.

Strayer 等人[38]也提出了通过检查带宽使用、持续时间和数据包时序等网络流属性来识别 IRC 僵尸网络命令与控制通信的方法.Livadas 等人则应用机器学习方法来对 IRC 僵尸网络通信流量进行检测[39],他们将任务分解为两个步骤:首先使用机器学习领域中经典的原始贝叶斯、贝叶斯网络、J48 决策树等分类器对 IRC 流量和非 IRC 流量进行区分,实验结果显示,原始贝叶斯分类器取得了最好的效果,误报率为 2.49%、漏报率为 15.04%均达到了较低水平;然后再从 IRC 流量中区分正常 IRC 通信和僵尸网络控制流量,在这个步骤中,所有的 3 种分类器均没有达到理想的效果,最好的贝叶斯网络分类器也仅达到了误报率为 10%~20%、漏报率 30%~40%间的平衡.这一研究工作表明,简单地将机器学习方法应用到僵尸网络检测并不能取得良好的效果,必须充分考虑僵尸网络控制机制的内在特性.

Goebel 等人在文献[40]中描述了一种简单而高效的 IRC 僵尸网络检测方法 Rishi,其基本思想是被动监听网络流量,通过开源的 ngrep 工具获取其中包含的 IRC 协议连接信息,然后用 *n*-gram 分析方法实现评分函数,通过对 IRC 昵称的异常评定,检测出内部网络中被 IRC 僵尸网络所感染的僵尸主机.利用这种方法,Goebel 等人在德国 RWTH Aachen university 校园网的 10G 网关上两周内检测出了 82 台被感染的僵尸主机.该方法虽然对现有的 IRC 僵尸网络检测比较有效,但还存在如下的不足之处:① Rishi 方法依赖于正则表达式来检测和评价一个僵尸程序昵称,但目前存在一些僵尸程序使用与 IRC 用户类似的昵称命名结构,从而导致 Rishi 无法有效检测,僵尸网络控制者也很容易修改僵尸程序昵称命名结构以绕过 Rishi 所定义的正则表达式;② 该方法只能用于对基于标准 IRC 协议僵尸网络的检测,无法应对基于 HTTP,P2P 协议和其他自定义协议的僵尸网络.

AT&T 实验室的 Karasaridis 等人在文献[41]中描述了一种在 ISP 骨干网层面上检测和刻画僵尸网络行为的方法,由如下步骤组成:① 对 AT&T Internet Protect 底层传感器触发的事件进行聚合,识别出具有可疑行为的主机;② 基于缺省 IRC 服务端口、识别到集中服务器的连接以及 IRC 流量模型特征这 3 个启发式规则,识别出可能的僵尸网络命令与控制连接;③ 分析可能的命令与控制连接,计算出连接同一服务器的可疑僵尸主机数量,计算出可疑连接与 IRC 流量模型的相似性距离,并结合两者计算该可疑连接为僵尸网络命令与控制信道的得分;④ 通过与其他数据源(如基于蜜罐技术发现的僵尸网络数据)的关联、DNS 域名验证和人工验证确认检测到的僵尸网络.与之前的工作相比,Karasaridis 等人的方法具有如下优势:① 分析方法完全在传输层以下进行,没有涉及应用层信息,因此,检测效率将更高,可在骨干网上实施;② 基于网络流数据被动监听与分析,不涉及隐私问题,并可以检测加密通信的 IRC 僵尸网络;③ 误报率低,在实验中达到了 2%的较好水平,同时可以量化僵尸网络的规模等信息.

Gu 等人采用 IDS 驱动的会话关联方法实现了能够检测僵尸程序感染的 BotHunter 系统[42].该系统基于证据链(evidence trail)关联思想,将僵尸程序感染过程视为一台内网主机与外网一台或多台主机间的信息交互序列,包括目标扫描、破解攻击、二进制代码注入与执行、命令与控制信道连接和对外扫描等步骤.BotHunter 系统底层采用 Snort 入侵检测系统的特征检测方法以及两个关注僵尸程序的异常检测插件 SLADE 和 SCADE,以对僵尸程序感染的各个步骤进行检测:SLADE 插件实现了对流入连接的有损性 *n*-gram 负载分析方法,通过对执行协议负载的字节分布异常检测出恶意代码攻击;SCADE 插件进行针对恶意代码的平行及垂直端口扫描分析,可以检测出流入连接和流出连接中的扫描事件.然后,BotHunter 关联分析器将底层 IDS 报告的流入扫描报警、破解攻击报警和外出控制信道报警、对外扫描报警等事件联系在一起,从而给出一个详细的包含所有相关事件的僵尸程序感染会话场景.BotHunter 系统的优点在于首次提出了一个关联和刻画僵尸程序整个感染过程的实时分析系统,并通过实际测试 35 个最近的僵尸程序验证了其有效性.

由于目前 IRC 协议仍是僵尸网络的主流控制协议,所以,几乎所有的相关研究工作都是关注 IRC 僵尸网络控制信道的检测和刻画.基于 HTTP 协议和基于 P2P 协议的僵尸网络由于具有较强的个性化差异,目前还无法给出通用化的检测方法,但随着这两类僵尸网络近年来的不断发展,构建对这两类僵尸网络的有效检测方法将是一个重要的研究课题.

4.3 僵尸网络的防御与反制

僵尸网络的防御与反制存在两种不同的方法:由于构建僵尸网络的僵尸程序仍是恶意代码的一种,因此,传统的防御方法是通过加强因特网主机的安全防御等级以防止被僵尸程序感染,并通过及时更新反病毒软件特征库清除主机中的僵尸程序.Overton 等人给出了防御僵尸程序感染的方法[43],包括遵循基本的安全策略以及使用防火墙、DNS 阻断、补丁管理等技术手段.另一种防御方法是针对僵尸网络具有命令与控制信道这一基本特性,通过摧毁或无效化僵尸网络命令与控制机制,使其无法对因特网造成危害.由于命令与控制信道是僵尸网络得以生存和发挥攻击能力的基础,因此,第 2 种防御方法比第 1 种更加有效.

对于集中式僵尸网络而言,在发现僵尸网络控制点的基础上,最直接的反制方法是通过 CERT 部门协调处理关闭控制点,然而,僵尸网络控制者可以在另外一台主机上重新构建控制服务器,并通过改变动态域名所绑定的控制服务器重建僵尸网络控制信道,所以,防御者还需通过联系域名服务提供商移除僵尸程序所使用的动态域名,从而彻底移除僵尸网络控制服务器.此外,在获取域名服务提供商的许可条件下,防御者还可以使用 DNS 劫持技术[29]来获取被僵尸网络感染的僵尸主机 IP 列表,从而及时通知被感染主机用户进行僵尸程序的移除.通过控制点或者僵尸程序代码追溯僵尸网络控制者是反制的一个重要目标,但更具挑战性.Ianelli 等人分析了这一问题的难度[44],并指出需要通过有效的国际协作、恶意代码起源的深入追溯以及对涉及的犯罪资金流进行跟踪等方法尝试完成这一目标.

由于 P2P 僵尸网络不存在集中的控制点,因此,对 P2P 僵尸网络的反制将更为困难.如何有效地检测和反制 P2P 僵尸网络还有待进一步研究.

5 总 结

作为一种从传统恶意代码形态进化而来的高级攻击方式,僵尸网络提供了隐匿、灵活且高效的一对多命令与控制机制,从而被攻击者所广泛接受并用于实现窃取敏感信息、发送分布式拒绝服务攻击和发送垃圾邮件等攻击目的.僵尸网络正在步入快速发展期,并已对因特网安全造成了严重威胁,对中国大陆造成的危害尤为严重.

此外,僵尸网络还呈现出如下发展趋势:命令与控制机制从基于 IRC 协议逐渐转移到基于 HTTP 协议和各种不同类型的 P2P 协议,以增强僵尸网络的隐蔽性和鲁棒性;在网络传播方面借鉴并融合了各类传统恶意代码的传播方式,包括最新的通过即时通信软件和 P2P 文件共享软件进行传播;通过增强认证和信道加密机制,对僵尸程序进行多态化和变形混淆,引入 Rootkit 隐藏机制使得对僵尸网络的检测、跟踪和分析更加困难.

鉴于僵尸网络所呈现的技术特点及其发展趋势,安全领域研究者必须加强僵尸网络的研究,并协调反病毒业界和应急响应部门进行有效反制,才能有效遏制其快速发展的势头.我们预期,僵尸网络领域在未来一段时间内研究的重点方向包括:① 新型的僵尸网络命令与控制机制及其应对策略;② 僵尸网络传播模型的进一步研究及在实验测试环境中的验证;③ 更具准确性和高效性的僵尸网络检测机制,特别是针对新型的僵尸网络命令与控制机制;④ 具有一定自动化程度的僵尸网络反制辅助平台的研究和实现.

References:

[1] Geer D. Malicious bots threaten network security. IEEE Computer, 2005,38(1):18−20.

[2] Lee WK, Wang C, Dagon D. Botnet Detection: Countering the Largest Security Threat. New York: Springer-Verlag, 2007.

[3] Symantec Inc. Symantec Internet security threat report: Trends for January 06~June 06. Volume X. 2006. http://eval.symantec.com/mktginfo/enterprise/white_papers/ent-whitepaper_symantec_internet_security_threat_report_ix.pdf

[4] Symantec Inc. Symantec Internet security threat report: Trends for July 06~December 06. Volume XI. 2007. http://eval.symantec.com/mktginfo/enterprise/white_papers/ent-whitepaper_symantec_internet_security_threat_report_x_09_2006.en-us.pdf
[5] Zhuge JW, Han XH, Ye ZY, Zou W. Discover and track botnets. In: Proc. of the Chinese Symp. on Network and Information Security (NetSec 2005). 2005. 183−189. (in Chinese with English abstract). http://www.honeynet.org.cn/reports/僵尸网络的发现与跟踪_NetSec2005_.pdf
[6] Zhuge JW, Han XH, Chen Y, Ye ZY, Zou W. Towards high level attack scenario graph through honeynet data correlation analysis. In: Proc. of the 7th IEEE Workshop on Information Assurance (IAW 2006). 2006. Piscataway: IEEE Computer Society Press. 2006. 215−222.
[7] Zhuge JW, Han XH, Zhou YL, Song CY, Guo JP, Zou W. HoneyBow: An automated malware collection tool based on the high-interaction honeypot principle. Journal on Communications, 2007,28(12):8−13 (in Chinese with English abstract).
[8] Han XH, Guo JP, Zhou YL, Zhuge JW, Cao DZ, Zou W. An investigation on the botnets activities. Journal on Communications, 2007,28(12):167−172. (in Chinese with English abstract).
[9] Sun YD, Li D. Overview of botnet. Computer Applications, 2006,26(7):1628−1630 (in Chinese with English abstract).
[10] Puri R. Bots & botnet: An overview. SANS White Paper. 2003. http://www.sans.org/reading_room/whitepapers/malicious/1299.php
[11] McCarty B. Botnets: Big and bigger. IEEE Security & Privacy, 2003,1(4):87−90.
[12] Bacher P, Holz T, Kotter M, Wicherski G. Know your enemy: Tracking botnets. 2005. http://www.honeynet.org/papers/bots
[13] Rajab MA, Zarfoss J, Monrose F, Terzis A. A multifaceted approach to understanding the botnet phenomenon. In: Almeida JM, Almeida VAF, Barford P, eds. Proc. of the 6th ACM Internet Measurement Conf. (IMC 2006). Rio de Janeriro: ACM Press, 2006. 41−52.
[14] Arce I, Levy E. An analysis of the slapper worm. IEEE Security & Privacy, 2003,1(1):82−87.
[15] Barford P, Yegneswaran V. An inside look at botnets. In: Christodorescu M, Jha S, Maughan D, Song D, Wang C, eds. Advances in Information Security, Malware Detection, Vol.27. Springer-Verlag, 2007. http://www.springerlink.com/content/w4576m3186524245/
[16] Holz T. A short visit to the bot zoo. IEEE Security & Privacy, 2005,3(3):76−79.
[17] Canavan J. The evolution of malicious IRC bots. In: Proc. of the 2005 Virus Bulletin Conf. (VB 2005). 2005. http://www.symantec.com/avcenter/reference/the.evolution.of.malicious.irc.bots.pdf
[18] Wen WP, Qing SH, Jiang JC, Wang YJ. Research and development of Internet worms. Journal of Software, 2004,15(8):1208−1219 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1208.htm
[19] Chiang K, Lloyd L. A case study of the rustock rootkit and spam bot. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323138&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820
[20] Daswani N, Stoppelman M, the Google Click Quality and Security Teams. The anatomy of Clickbot.A. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323139&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820
[21] Kalt C. RFC 2810: Internet relay chat: Architecture. RFC 2810, IETF, 2000.
[22] Grizzard JB, Sharma V, Nunnery C. Peer-to-Peer botnets: Overview and case study. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323129&coll=GUIDE&dl= GUIDE&CFID=16751383&CFTOKEN=82837820
[23] Wang P, Sparks S, Zou CC. An advanced hybrid peer-to-peer botnet. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323130&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820
[24] Vogt R, Aycock J, Jacobson MJ. Army of botnets. In: Proc. of the 14th Annual Network & Distributed System Security Conf. (NDSS). 2007. http://www.isoc.org/isoc/conferences/ndss/07/abstracts/54.shtml
[25] Li J, Ehrenkranz T, Kuenning G, Reiher P. Simulation and analysis on the resiliency and efficiency of malnets. In: Proc. of the IEEE Symp. on Measurement, Modeling, and Simulation of Malware (MMSM 2005). Monterey: IEEE Computer Society Press, 2005. 262−269.
[26] Zou CC, Gong W, Towsley D. Code red worm propagation modeling and analysis. In: Atluri V, ed. Proc. of the 9th ACM Conf. on Computer and Communications Security (CCS 2002). New York: ACM Press, 2002. 138−147.
[27] Kim J, Radhakrishnan S, Dhall SK. Measurement and analysis of worm propagation on Internet network topology. In: Proc. of the IEEE Int'l Conf. on Computer Communications and Networks (ICCCN 2004). 2004. 495−500. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1401716
[28] Zou CC, Gong W, Towsley D. Worm propagation modeling and analysis under dynamic quarantine defense. In: Staniford S, ed. Proc. of the ACM CCS Workshop on Rapid Malcode (WORM 2003). New York: ACM Press, 2003. 51−60.
[29] Dagon D, Zou CC, Lee W. Modeling botnet propagation using time zones. In: Proc. of the 13th Annual Network and Distributed System Security Symp. (NDSS 2006). 2006. http://www.isoc.org/isoc/conferences/ndss/06/proceedings/papers/modeling_botnet_propagation.pdf
[30] Zou CC, Towsley D, Gong W. On the performance of Internet worm scanning strategies. Elsevier Journal of Performance Evaluation, 2005,63(7):700−723.
[31] Barford P, Blodgett M. Toward botnet mesocosms. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323134&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820
[32] Freiling F, Holz T, Wicherski G. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. In: Proc. of the 10th European Symp. on Research in Computer Security (ESORICS 2005). LNCS 3679, Milan: Springer-Verlag, 2005. 319−335.
[33] Baecher P, Koetter M, Holz T, Dornseif M, Freiling FC. The nepenthes platform: An efficient approach to collect malware. In: Vimercati SD, Syverson P, eds. Proc. of the 9th Int'l Symp. on Recent Advances in Intrusion Detection (RAID). LNCS 4219, Springer-Verlag, 2006. 165−184.

[34] Rajab MA, Zarfoss J, Monrose F, Terzis A. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323133&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820

[35] Ramachandran A, Feamster N, Dagon D. Revealing botnet membership using DNSBL counter-intelligence. In: Proc. of the USENIX Workshop on Steps to Reducing Unwanted Traffic in the Internet (SRUTI 2006), Vol.2. Berkeley: USENIX Association, 2006. 8. http://portal.acm.org/citation.cfm?id=1251304&dl=&coll=

[36] Binkley JR, Singh S. An algorithm for anomaly-based botnet detection. In: Proc. of the USENIX 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2006). 2006. 43−48. http://portal.acm.org/citation.cfm?id=1251296.1251303&coll=&dl=

[37] Binkley JR. Anomaly-Based botnet server detection. In: Proc. of the FloCon 2006 Analysis Workshop. 2006. http://www.cert.org/flocon/2006/presentations/botnet0606.pdf

[38] Strayer T, Walsh R, Livadas C, Lapsley D. Detecting botnets with tight command and control. In: Proc. of the 31st IEEE Conf. on Local Computer Networks (LCN'06). Tampa: IEEE Computer Society Press, 2006. 195−202.

[39] Livadas C, Walsh B, Lapsley D, Strayer T. Using machine learning techniques to identify botnet traffic. In: Proc. of the 2nd IEEE LCN Workshop on Network Security. 2006. 967−974.

[40] Goebel J, Holz T. Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323136&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820

[41] Karasaridis A, Rexroad B, Hoeflin D. Wide-Scale botnet detection and characterization. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007. http://portal.acm.org/citation.cfm?id=1323128.1323135&coll=GUIDE&dl=GUIDE&CFID=16751383&CFTOKEN=82837820

[42] Gu G, Porras P, Yegneswaran V, Fong M, Lee W. BotHunter: Detecting malware infection through IDS-driven dialog correlation. In: Proc. of the 16th USENIX Security Symp. (Security 2007). 2007. http://www.usenix.org/events/sec07/tech/gu.html

[43] Overton M. Bots and botnets: Risks, issues and prevention. In: Proc. of the 2005 Virus Bulletin Conf. 2005. http://momusings.com/papers/VB2005-Bots_and_Botnets-1.0.2.pdf

[44] Ianelli N, Hackworth A. Botnets as a vehicle for online crime. In: Proc. of the 18th Annual FIRST Conf. 2006. http://www.cert.org/archive/pdf/Botnets.pdf

附中文参考文献:

[5] 诸葛建伟,韩心慧,叶志远,邹维.僵尸网络的发现与跟踪.见:中国网络与信息安全技术研讨会论文集.2005.183−189.

[7] 诸葛建伟,韩心慧,周勇林,宋程昱,郭晋鹏,邹维.HoneyBow:一个基于高交互式蜜罐技术的恶意代码自动捕获器.通信学报,2007,28(12):8−13.

[8] 韩心慧,郭晋鹏,周勇林,诸葛建伟,曹东志,邹维.僵尸网络活动调查分析. 通信学报, 2007,28(12):167−172.

[9] 孙彦东,李东.僵尸网络综述.计算机应用,2006,26(7):1628−1630.

[18] 文伟平,卿斯汉,蒋建春,王业君.网络蠕虫研究与进展.软件学报,2004,15(8):1208−1219. http://www.jos.org.cn/1000-9825/15/1208.htm

诸葛建伟(1980－),男,浙江瑞安人,博士,助理研究员,主要研究领域为入侵检测与关联,蜜罐与蜜网技术,网络攻防,恶意代码分析.

叶志远(1963－),男,高级工程师,主要研究领域为网络与信息安全.

韩心慧(1969－),男,博士生,助理研究员,主要研究领域为网络与信息安全.

邹维(1964－),男,研究员,主要研究领域为网络与信息安全.

周勇林(1974－),男,博士生,高级工程师,主要研究领域为互联网安全监测,应急响应处理.