计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
48(8): 1315-1331, 2011

僵尸网络综述

方滨兴[1,2] 崔 翔[1,3] 王 威[4]
[1](中国科学院计算技术研究所信息安全研究中心 北京 100190)
[2](北京邮电大学 北京 100876)
[3](中国科学院研究生院 北京 100049)
[4](中国人民公安大学 北京 100038)
(fangbx@cae.cn)

Survey of Botnets

Fang Binxing[1,2], Cui Xiang[1,3], and Wang Wei[4]
[1](*Research Center of Information Security, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
[2](*Beijing University of Posts and Telecommunications, Beijing* 100876)
[3](*Graduate University of Chinese Academy of Sciences, Beijing* 100049)
[4](*Chinese People's Public Security University, Beijing* 100038)

Abstract In recent years, botnets, evolving from traditional worms and Trojans, have become one of the most effective platforms for many Internet attacks. Botnets have even become a powerful weapon for cyberwarfare. Therefore, as defenders, we should pay more attention to botnets—both current research findings and their evolution trends. In this paper, we divide the evolution of botnets into five phases and analyze their characteristics and corresponding representative botnets in each phase. To describe botnets unambiguously, we define botnets formally and classify botnets into four classes based on topology structures. In order to have an overall perspective of current research works, we divide them into five fields: detection, tracking, measurement, prediction, countermeasures, and analyze each field in detail. Based on the comprehensive study of the development law of botnet attacks and defense, we exact several inescapable weaknesses inside botnets, which could be exploited to defend against botnets. To conclude the paper, we suggest possible countermeasures against botnets and predict possible evolution trends of botnets.

Key words botnet; C&C; countermeasure; value-added network attack; survey

摘 要 近年来,从传统蠕虫和木马发展形成的僵尸网络逐渐成为攻击者手中最有效的攻击平台,甚至可以成为网络战的武器,因此,关注僵尸网络已有研究成果与发展趋势都十分必要.将僵尸网络的发展历程概括为5个阶段,分析各阶段特点和代表性僵尸网络.对僵尸网络进行形式化定义并依据命令控制信道拓扑结构将其划分为4类.同时,将当前僵尸网络研究热点归纳为检测、追踪、测量、预测和对抗5个环节,分别介绍各环节的研究状况,并对每个环节的研究进展进行归纳和分析.通过研究僵尸网络在攻防对抗中的演进规律,提取僵尸网络存在的不可绕过的脆弱性.最后,综合分析当前僵尸网络研究现状,并展望僵尸网络发展趋势.

关键词 僵尸网络;命令控制信道;网络对抗;增值网络攻击;综述

中图法分类号 TP393

收稿日期:2011-04-07;**修回日期**:2011-06-15

基金项目:国家"九七三"重点基础研究计划基金项目(2007CB311100);国家自然科学基金项目(61070186,61070026)

僵尸网络(botnet)是通过入侵网络空间内若干非合作用户终端构建的、可被攻击者远程控制的通用计算平台.其中,"非合作"是指被入侵的用户终端没有感知;"攻击者"指的是对所形成的僵尸网络具有操控权力的控制者(botmaster);"远程控制"指的是攻击者可以通过命令与控制(command and control, C&C)信道一对多地控制非合作用户终端.一个被控制的受害用户终端成为僵尸网络的一个节点,可称之为"僵尸主机",俗称"肉鸡".一个僵尸网络可以控制大量的用户终端,可以获得强大的分布式计算能力和丰富的信息资源储备.利用僵尸网络,攻击者更易于发起分布式拒绝服务攻击(DDoS)、在线身份窃取(online Identity theft)、垃圾邮件(spam)、木马和间谍软件批量分发等网络攻击,我们可以把这种以僵尸网络为平台的攻击称之为"增值网络攻击".僵尸网络作为攻击者手中最有效的通用攻击平台,已成为当今互联网最大安全威胁之一和网络安全领域研究热点.

僵尸网络是一种控制命令驱动的信息系统,它的行为取决于控制者的命令输入.因此,一个具体的僵尸网络可能造成的危害通常难以预测.从已有的僵尸网络来看,其危害已影响到政治、经济和国家安全等多个重要领域,其对军事领域的影响也会很快显现出来.围绕着僵尸网络的危害性,国内外媒体出现过大量的报道.例如,2008 年爆发的飞客(conficker)[1]僵尸网络具有极强的扩散能力,短期内即感染了 1 000 多万台计算机,就如同某个军队在进行网络战实验一样,效果非凡.据报道,英国议会电脑网络和法国、德国等国家部分军事系统也被飞客所感染,甚至因网络阻塞而造成了法国海军战机停飞事故.国家计算机网络应急技术处理协调中心(CNCERT/CC)在 2010 上半年通过抽样监测发现中国大陆有 23.3 万个 IP 地址感染僵尸程序,而参与控制这些僵尸主机包括位于境外的 4 584 个控制服务器[2].赛门铁克(Symantec)公司 2010 年发布的互联网安全威胁报告显示:中国是僵尸网络的最大受害国之一,仅列美国之后,甚至在一段期间内亚太地区中的 71%的僵尸主机位于中国大陆境内[3].这些数字说明僵尸网络对我国造成的安全威胁是严重的.

僵尸网络带来的严重安全威胁引起了国际上的广泛关注.在学术界,USENIX 协会从 2007 年开始举办僵尸网络专题研讨会 HotBots(Workshop on Hot Topics in Understanding Botnets),2008 年,WORM 与 HotBots 合并为 LEET(Workshop on Lager-scale Exploits and Emergent Threats),专门探讨僵尸网络、间谍软件和蠕虫.同时,近年来在 USENIX Security Symposium, NDSS, CCS, RAID 等著名会议上发表的僵尸网络研究成果呈明显增多趋势.在工业界,微软公司在 2004 年发起了国际反僵尸网络工作组.在政府部门,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 3 个部门联合在 GA Tech 举办了名为"ARO-DARPA-DHS Special Workshop on Botnet"的僵尸网络专门研讨会,对这一严重安全威胁进行了深入探讨.

僵尸网络之所以形成如此严重的威胁,从技术角度来看,主要有以下原因:

1) 僵尸网络是从传统蠕虫和木马发展而来的一种新的攻击形式,蠕虫具有利用既有的安全漏洞而快速传染扩散的优势,但存在感染大量计算机后不被控制者所控制的缺点,即攻击者无法利用已感染的计算机形成增值网络攻击,甚至因其不可控而无法获知蠕虫扩散速度、感染规模和地理分布等基本信息;木马具有对受害者远程控制的能力,但存在感染速度慢、管理规模小和控制方式简单的缺点.僵尸网络是结合了两者优势、弥补了两者不足而形成的产物.

2) 僵尸网络实现了控制逻辑与攻击任务的分离.位于僵尸主机上的僵尸程序负责控制逻辑,真正的攻击任务由控制者按需动态分发.这种方法的核心要点是将完整的威胁实体分割为多个部分,从而既可以为任务分发提供良好的灵活性,又可以提高僵尸网络的可生存性.

3) 信息安全是一种伴随技术,相应安全措施的出现通常要明显滞后于所对应的信息新技术的应用,这种时间差为网络攻击提供了生存空间.就僵尸网络而言,寻找恰当的控制服务器是纯中心结构僵尸网络的生存基础,因此利用各种新技术的漏洞来构建僵尸网络的控制器是僵尸网络构建的核心要素之一.例如,公共 IRC 聊天室常被用于僵尸网络控制服务器;无需认证的 Web 2.0 服务可被用作僵尸网络的命令控制信道;亚马逊 EC2 云被 ZeuS 穿透进而被用作僵尸网络控制服务器;一些缺乏信息安全立法的国家所提供的服务器托管服务也经常被用作僵尸网络控制服务器.

4) 僵尸网络充分利用了密码学的成果,从理论上确保了其控制权不可被对抗僵尸网络的人所接管.

例如,由于在 P2P 僵尸网络中没有中心控制服务器,使得任何人都可以将控制命令作为所发布的资源而注入到 P2P 网络中,因此僵尸程序必须依赖公钥密码体系对控制命令认证,而控制者必须保证私钥的安全性,并防止控制信息重放攻击.

对僵尸网络的研究可归纳为检测(detection)、追踪(tracking)、测量(measurement)、预测(prediction)和对抗(countermeasure)5 个部分. 其中:检测的目的是发现新的僵尸网络;追踪的目的是获知僵尸网络的内部活动;测量的目的是掌握僵尸网络的拓扑结构、活跃规模、完全规模和变化轨迹;预测的目的是考虑未来可能出现的攻击技术并预先研究防御方法;对抗的目的是接管僵尸网络控制权或降低其可用性.

1 僵尸网络的定义及分类

僵尸网络的发展历程可以概括为 5 个阶段:以 IRC 协议为代表的开创阶段;以 HTTP 协议、简单 P2P 协议和 Fast-flux[4] 技术为代表的发展阶段;以 Domain Flux[5],URL Flux[6],Hybrid P2P[7] 协议为代表的对抗阶段;以智能手机僵尸网络为代表的融合阶段;以攻击物理隔离内网的僵尸网络为代表的广泛攻击阶段.

僵尸程序(bot)的起源可追溯到 1993 年的出现的"Eggdrop Bot",Eggdrop 像智能机器人(这就是 bot 一词的来源,即 robot 的简写)一样帮助 IRC 网络管理员更高效地管理网络[8-9]. 可见,早期的僵尸程序技术并不以破坏为目的. 1998 年出现的 GTBot 是第 1 个知名的恶意僵尸网络,它使用 IRC 协议构建命令控制频道,GTBot 在其程序中内嵌了一个流行的 IRC 客户端 mIRC.exe. 从 GTBot 广泛流行以后,人们开始意识到 IRC 协议是进行终端控制的一种有效途径,此后,基于 IRC 协议的僵尸网络层出不穷,如 PrettyPark,Sdbot,Spybot,Rbot,Agobot 等,这使得 IRC 协议成为构建僵尸网络命令控制信道的主流协议. 在僵尸网络的发展历程中,Sdbot 是第 1 个在程序内部实现了必要的 IRC 客户端协议的僵尸网络;Agobot 的高度模块化设计和广泛传播,使得僵尸网络的危害开始引起广泛重视. 在此时期,僵尸网络在社会上还是一个新名词,2003 年爆发的著名的口令蠕虫(DvlDr)其实已经是利用 IRC 协议进行命令控制的僵尸网络,但当时人们将注意力集中在其暴力破解口令的蠕虫特色上. 2004 年底,国家计算机网络应急技术处理协调中心处置了第 1 起控制者是国内黑客的僵尸网络事件,从此僵尸网络这一概念开始在国内迅速盛行.

随着攻防对抗的演进,攻击者意识到 IRC 协议已经不能应对易于被检测和被关闭的局面,为了让僵尸网络具备更好的隐蔽性和健壮性,控制者开始转向 P2P 协议和 HTTP 协议. 第 1 个 P2P 僵尸网络是 2002 年出现的 Slapper,以后相继出现了 Sinit,Phatbot,Nugache 和 Storm[10-11] 等. 早期的 P2P 僵尸网络仍然具有许多弱点,如 Slapper 并没有实现认证机制,这使其可被劫持;Sinit 采用非结构化的随机扫描方法发现 Peer,效率很低且造成流量异常;Phatbot 基于 WASTE 协议构建命令控制信道,易于被关闭;Nugache 的 Bootstrap 过程依赖于硬编码的 22 个 IP 的地址,这使其具有单点失效问题;Storm 采用的基于 DHT 思想的 Overnet 协议,可受到索引污染和 Sybil 攻击;等等. 尽管如此,早期的 P2P 僵尸网络还是相对安全的. Bobax[12],Rustock 和 Clickbot 是典型的基于 HTTP 协议的僵尸网络. 僵尸程序利用 HTTP 协议周期性轮询控制服务器获取控制命令,具有隐秘性强和易于穿透防火墙等优点.

以 Storm 和 Bobax 为代表的僵尸网络的成功,使得国际社会对僵尸网络高度重视,攻防对抗变得日益激烈,攻击者意识到简单的 P2P 和 HTTP 协议依然存在致命的弱点,控制信道容易被阻断,必须设计更高级的命令控制信道. 2008 年出现的 Conficker 僵尸网络同时使用了 Domain Flux 和 Random P2P 两种寻址方法,使得现有防御方法至今无法阻断其命令控制信道;同年出现的 Waledac[13] 僵尸网络,综合使用了 HTTP,Fast-flux 和 Hybrid P2P 协议,原理上具有极好的健壮性和隐秘性. 虽然因为 C&C 协议漏洞而导致 Waledac 最终被摧毁,但这代表僵尸网络开始向更高级的命令控制结构迈进.

随着智能手机的普及和 3G 通信的发展,智能手机逐渐具有更强大的计算能力和更方便的网络接入途径,这使得手机僵尸网络生存环境趋于成熟. 2009 年,首个针对 Symbian 平台的手机僵尸网络 Symbian.Yxes[14] 出现. 随后,出现了首个针对越狱 iPhone 的 iKee.B[15] 手机僵尸网络. 2010 年,首个针对 Android 系统的手机僵尸网络 Geinimi[16] 出现. 因为缺乏相应的防御系统,这些手机僵尸网络简单地采用基本的 HTTP 协议就足以满足控制需求.

2010年出现的具有部分僵尸网络功能的Stuxnet(震网)[17]是第1个可攻击工业控制系统的恶意代码,其首次实现了PLC(可编程逻辑控制器)rootkit,这标志着攻击物理隔离内网的僵尸网络开始出现萌芽.图1列出了僵尸网络发展历程中的里程碑事件.

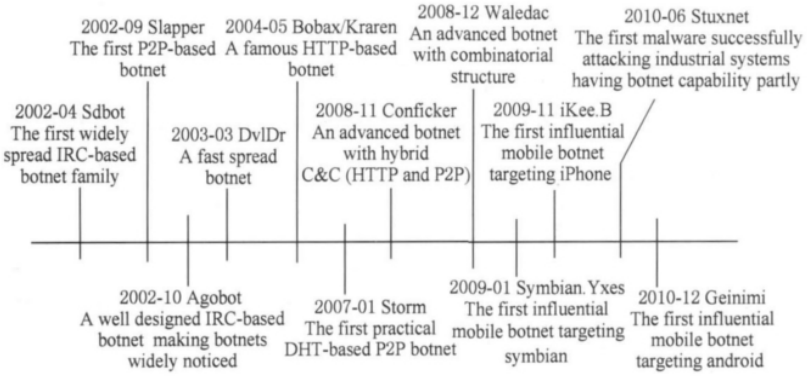

Fig. 1 Timeline of botnets evolution.

图1 僵尸网络发展历程

1.1 僵尸网络定义

国家工业和信息化部发布的《木马和僵尸网络监测与处置机制》和Xie等人[18]定义僵尸网络为:由攻击者通过控制服务器控制的受害计算机群.然而,控制服务器不是僵尸网络的必要组成部分.例如,P2P僵尸网络中没有控制服务器.Gu等人[19]定义为:僵尸网络是可被攻击者通过命令控制信道远程控制的可协同的计算机群,该定义覆盖了僵尸网络2个本质属性"可控性"和"协同性",因此说该定义应该是比较准确的.为准确描述僵尸网络,本文给出僵尸网络形式化定义.

定义1. 僵尸网络.僵尸网络由四元组所构成,反映的是可被攻击者通过命令控制信道进行远程控制使之进入特定状态的计算机群.记为 $Botnet=(ZOMBIE,CMD,\delta,CCC)$.

$ZOMBIE$:通过入侵手段获取的僵尸主机集合.记为 $ZOMBIE=(BOT,S,ACTIVITY)$.

BOT:运行在受控僵尸主机上的僵尸程序集合.记为 $BOT=\{bot_1,bot_2,\cdots,bot_n\}$,其中,$bot_i$ 表示运行在 $Zombie_i$ 上的僵尸程序,n 反映了僵尸网络的规模.

S:僵尸程序状态集合.记为 $S=\{s_1,s_2,\cdots,s_n\}$.s_i 反映了僵尸程序所处状态.

$ACTIVITY$:僵尸程序动作集合.记为 $ACTIVITY=\{activity_1,activity_2,\cdots,activity_n\}$.

CMD:僵尸程序可执行的控制命令集合.记为 $CMD=\{cmd_1,cmd_2,\cdots,cmd_n\}$.

δ:转换函数,反映了僵尸程序收到控制命令后产生的相应动作及状态变迁.记为 $\delta:BOT\times S\times CMD\rightarrow BOT\times S\times ACTIVITY$,并满足 $\delta(bot\times s_i\times cmd)=(bot\times s_j\times activity)$,$i>0$,$j>0$.

CCC:命令控制信道(C&C channel).控制者和僵尸程序通过命令控制信道传输控制命令及数据.记为 $CCC=(TOPOLOGY,METHOD,RESOURCE)$.

$TOPOLOGY$:拓扑结构集合,即僵尸网络命令控制信道的网络拓扑结构.记为 $TOPOLOGY=\{$pure C/S, pure P2P, combinatorial structure, hybrid structure$\}$

$METHOD$:控制者和僵尸程序共同使用的协议和算法.记为 $METHOD=\{PROTOCAL=\{$IRC, HTTP, Domain Flux, Fast-flux, URL Flux, Kademilia, Random, $\cdots\}$, $ALGRITHM=\{$RSA, MD5, RC4, DES, AES, BASE64, DGA, UGA, $\cdots\}\}$.

$RESOURCE$:控制者使用的软件资源集合,包括域名、验证码和密钥等;控制者和僵尸程序利用的中间节点集合,包括公共IRC/HTTP/Web 2.0服务器、P2P网络节点、跳板机等.记为 $RESOURCE=\{$Domain Name, Authenticode, Key, IRC Server, HTTP Server, P2P Peer, StepStone, $\cdots\}$

1.2 分类方法

本文关注的僵尸网络检测、追踪、测量、预测和对抗研究都紧密围绕着僵尸网络拓扑结构展开,因此有必要基于拓扑结构对僵尸网络进行分类,这也是目前僵尸网络主流分类法的立足点.例如,Cooke等人[20]将其分类为中心结构(Centralized)、P2P结构和随机结构(Random).其实,Random是非结构化P2P,属于P2P结构的一个子集.Leder等人[21]

分类为中心结构、非中心结构(Decentralized)和动态结构(Locomotive).但是,Domain Flux 这种新型的命令控制方法虽然控制服务器不断变换,但本质上依然是中心结构.而且,P2P 僵尸网络中的节点同样存在动态性.因此,动态结构缺少唯一性描述.本文将给出新的面向拓扑结构的分类法,以弥补上述缺陷.本文按照拓扑结构将僵尸分为 4 类:纯中心结构、纯 P2P 结构、组合结构和混合结构.

纯中心结构只采用客户端-服务器(C/S)模式,即所有僵尸程序连接到中心控制服务器以获取控制命令.按照僵尸程序定位中心控制服务器的方法,又可将纯中心结构分为静态中心结构和动态中心结构."静态"指的是中心控制服务器的域名或 IP 地址是固定的,硬编码在僵尸程序体内(如图 2(a));"动态"指的是控制服务器地址需要根据特定算法动态生成.寻址算法是固定的,硬编码在僵尸程序体内(如图 2(b)).当前,典型的具有静态中心结构的僵尸网络基于 IRC,HTTP,Fast-flux 和 URL Flux,动态中心结构僵尸网络基于 Domain Flux.纯中心结构具有通信速度快的优点,但需要具有公网可达服务器(具有静态的公网 IP 地址,且可接受远程客户端主动发起的连接并与之通信)的支持.

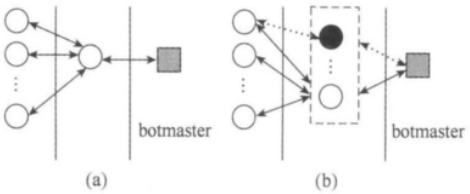

Fig. 2 Pure centralized structure.

图 2 纯中心结构

纯 P2P 结构只采用 Peer-to-Peer 模式,即每个僵尸程序既充当客户端又充当服务器,不存在专用的服务器,其拓扑结构如图 3(a)的中间部分所示.纯 P2P 结构不需要公网可达服务器支持,从而消除了单点失效问题.典型的具有纯 P2P 结构的僵尸网络基于分布式 Hash 表(distributed Hash table, DHT)和 Random 结构(即通过随机访问来寻找 Peer).基于 DHT 思想的结构具有良好的分布性,但容易受到索引污染(index poisoning)和 Sybil 攻击[22].Random 结构是最健壮的纯 P2P 模式,但会造成网络流量明显异常,且寻址和通信速度很慢.只有当僵尸网络规模极大时(如千万级或更多),Random 结构才会体现出优势.

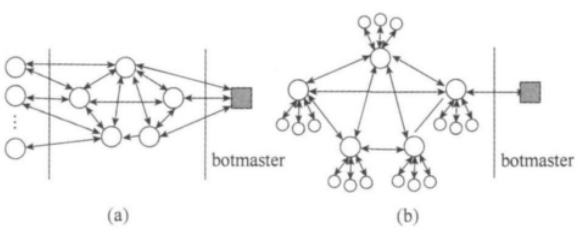

Fig. 3 Combinatorial Structure.

图 3 组合结构

组合结构包括 2 种情况:1)以 C/S 为主的组合结构.逻辑上是 C/S 结构,但服务器部分在物理实现上又是纯 P2P 结构(图 3(a)).在这种组合结构中,一部分公网可达僵尸主机同时作为客户端和服务器,其他僵尸主机仅作为客户端.这样,公网可达僵尸主机群构成一个开放的 P2P 网络,形成了一个动态控制服务器池.所谓动态服务器池是指由僵尸主机构成的动态变化的服务器群.动态指的是随着新的公网可达僵尸主机的加入而膨胀、随着公网可达僵尸主机的消亡而缩小、随着公网可达僵尸程序之间 Peer List 的交换而产生拓扑结构的变化.动态控制服务器池逻辑上相当于纯中心结构中的中心控制服务器群,两者的区别在于后者的控制服务器是静态的、可枚举的.2)以 P2P 为主的组合结构.逻辑上是 P2P 结构,但每个 Peer 在物理实现上又是 C/S(图 3(b))结构.在这种组合结构中,一部分公网可达僵尸主机作为服务器,管理一定数量的非公网可达僵尸主机,公网可达僵尸主机又通过 P2P 协议连接起来形成 P2P 网络.这样,就可以将一个大规模僵尸网络分裂为多个隐秘的僵尸子网,僵尸子网之间又通过健壮的 P2P 协议连接起来.由上述分析可见,组合结构兼具纯中心结构和纯 P2P 结构的优势,又消除了两者的缺点,是一种由控制者自定义的高级 C&C 协议,也正因如此,组合结构缺乏实践考验,可能存在设计缺陷从而被防御者反制,如 Waledac 被关闭事件即是反制成功的案例.

混合结构同时采用了一种以上上述结构,从而实现了命令控制信道的容灾及可生存性.例如,Conficker 同时使用了 Domain Flux 和 Random 协议.当 Domain Flux 失败时,则启动 Random 协议实现命令控制,从而增加了健壮性.基于 DHT 思想的纯 P2P 结构僵尸网络在 Bootstrap 过程中通常依赖于硬编码在僵尸程序内的地址列表,使之具有单点失效问题,如果此类僵尸网络使用 Domain Flux 协议获取 Bootstrap 所需资源,则可以消除这类脆弱性.

2 检测技术

检测的目的是发现新出现的僵尸网络."发现僵尸网络"包括3方面的含义:1)在单机上发现了僵尸程序样本并监测到C&C通信;2)通过网络边界流量监测或单点探测发现控制服务器的存在;3)在网络安全事件日志或网络应用数据中发现若干主机或账号的行为疑似由僵尸网络产生.可见,检测环节通常可以判定僵尸网络的存在,发现部分僵尸节点,但不能掌握大部分僵尸主机的地址,也不能掌握僵尸网络的拓扑结构、规模和行为等特征.

当前,检测僵尸网络的主要方法可归纳为4类:1)终端检测.首先利用蜜罐获取恶意代码样本,然后对捕获的恶意代码进行主机层面的分析进而筛选出僵尸程序.2)网络流量分析检测.利用僵尸网络C&C通信具有时空相似性的特性进行检测.所谓时空相似性,指的是大量僵尸程序在维持连接、收发控制命令和执行攻击任务时经常表现出协同性,使得多个僵尸程序会在同一时间窗内进行内容相似的通信.网络流量分析一般与网络安全事件检测联动来筛选可疑流并对流进行聚类分析.3)协议特征检测.有些僵尸网络具有个性化的协议特征,可以利用这些特性检测此类僵尸网络.4)基于增值网络攻击检测.某些特定安全事件有很大概率是由僵尸网络发起的,定位到这些安全事件的源头,就可能发现僵尸程序.

2.1 终端检测

定义2. 可控环境.可控环境包括安全运行环境(如沙箱、虚拟机、带有还原功能的真实计算机)和安全通信环境(如利用防火墙拦截攻击行为、重写攻击数据、利用入侵检测系统发现安全事件).

定义3. 沙箱恶意代码.在可控环境中运行的真实恶意代码称为沙箱恶意代码(SandMalware).沙箱恶意代码是真实恶意代码的完整拷贝,由可控环境过滤和篡改其发出的攻击流量和数据.

定义4. 沙箱僵尸程序.在可控环境中运行的真实僵尸程序称为沙箱僵尸程序(Sandbot).沙箱僵尸程序是真实僵尸程序的完整拷贝,由可控环境过滤和篡改其发出的攻击流量和数据.

利用终端检测僵尸网络分为2个环节:1)利用蜜罐捕获恶意代码.蜜罐是最早用于捕获僵尸程序的方法,早期的工作源于德国的蜜网项目组[23]、国内的国家计算机网络应急技术处理协调中心以及隶属于北京大学的狩猎女神项目组[24].按照蜜罐获取恶意代码的方式,可将其分为被动蜜罐和主动蜜罐.被动蜜罐需对外暴露漏洞,被动等待被恶意代码发现和攻击,典型代表是Nepenthes和Argos;主动蜜罐内部构造漏洞,主动联系恶意代码宿主(如恶意网页),促使宿主上的恶意代码成功下载执行,典型代表是Capture-HPC.Nepenthes可以有效发现攻击已知漏洞的恶意代码,Argos利用了污点传播技术从而可以发现利用0-Day漏洞的未知恶意代码;Capture-HPC可有效发现利用"网页挂马"(包括利用已知和部分未知漏洞的恶意网页)传播的恶意代码.2)主机层判定僵尸程序.利用蜜罐获取的网络攻击可能是黑客入侵、蠕虫自动传播和间谍软件窃密等,因此,需要在可控环境中运行SandMalware以判定其是否是僵尸程序.对SandMalware的进程、文件、注册表和网络行为进行自动化分析,可以判定其是否为Sandbot[25].

终端检测需要解决恶意代码样本获取和可控环境安全2个问题.利用被动蜜罐获取恶意代码,需要有畅通的网络环境并尽量多点部署蜜罐;利用主动蜜罐获取恶意代码,需要具备持续获取可疑恶意网页URL作为输入的能力.主被动蜜罐虽然可以检测到利用扫描漏洞和网页挂马来传播的恶意代码,但是不能检测到利用钓鱼邮件、P2P文件共享、移动介质、感染文件、PPC(pay-per-click)等方法传播的恶意代码,从而造成漏报.此外,国内外ISP对敏感端口(如TCP 135/445)的过滤,也让被动蜜罐丢失捕获部分僵尸网络的机会.搭建安全的可控环境也是一个困难的工作,对于未经完全逆向分析的恶意代码,一旦可控环境未能判定攻击行为,有可能放行SandMalware发起的网络攻击,从而涉及法律问题.此外,精心设计的僵尸程序可以有效识别可控环境,然后停止一切C&C通信以躲避检测.

2.2 网络流量分析检测

僵尸程序之间以及僵尸程序与控制服务器之间的通信具有时空相似性,这与正常用户的网络通信模式具有较大差异.例如,基于IRC协议的僵尸程序与IRC服务器之间会周期性传送PING/PONG命令;基于HTTP协议的僵尸程序会周期性连接控制服务器轮询是否存在新的控制命令;基于Kademlia协议的僵尸程序会周期性搜索控制命令;大量僵尸程序收到DDoS控制命令后,会在指定时间内同时向目标地址发起攻击.以上这些通信流量

都具有时空相似性,因此,通过网络流量分析能够发现僵尸网络.

网络流量分析系统可部署在网络边界上实时收集网络数据流(类似 NetFlow 格式),也可汇总不同网络的多个路由器发送来的 NetFlow 流.对相似的流(如具有相近的 bytes-per-packet,bytes-per-second,packets-per-flow)、汇聚的流(如多个客户端访问同一个目的 IP 和端口)、相同攻击的流(如发送垃圾邮件、端口扫描)作聚类分析,并将其标记为可疑的僵尸网络.然后,重点监测可疑僵尸网络的客户端和服务端通信,当加权可疑度超过阈值时,将其判定为僵尸网络.Gu 等人[19,26-27]利用该思想实现了 BotHunter,BotSniffer 和 BotMiner 3 个渐进的僵尸网络发现系统和方法,特别地,BotMiner 可发现拓扑结构无关和协议无关的僵尸网络.类似地,Karasaridis 等人[28]也利用时空相似性及其与攻击事件的关联在骨干网上监测传输层流量来检测 IRC 僵尸网络.

利用 C&C 流量时空相似性分析的方法可适应多种结构和协议的僵尸网络,具有很好的通用性和实时性.不同于终端检测,网络流量分析检测系统可部署在小型网络边界甚至骨干网上,从而既可以细粒度地监测特定用户网络,又可以从更高的视角在广域网层面并发检测更多的僵尸网络.僵尸网络需要改造 C&C 协议消除时空相似性才有可能绕过该方法的检测,而绕过检测是以降低僵尸网络攻击能力为代价的[29].

2.3 基于协议特征检测

针对具有明显协议特征的僵尸网络,可以利用协议特征检测以降低检测系统部署成本并提高准确性.1)在 IRC 僵尸网络中,僵尸程序通过硬编码的算法按照一定规律生成昵称(nickname),使得多个 IRC 客户端(僵尸程序)表现出昵称相似性,成为可检测点[30-31];2)在 Fast-flux 僵尸网络中,中心控制服务器域名对应 IP 不断跨 AS 域变化并且包含大量 NS 记录,这与正常的域名解析结果(包括 Round-robin DNS 和 Content Distribution Networks)差别很大,因此可通过探测特定域名方法判断其是否被僵尸网络所用[4,32].在 Fast-flux 僵尸网络中,僵尸程序与中心控制服务器之间的通信必须经过 Fluxbot,造成可判定的网络连接延迟和下载文件延迟,这些偏离了正常网络行为的协议特征使之成为可检测点[33].

基于 IRC 协议昵称相似性的僵尸网络检测系统需要检查报文 payload,所以不能使用 NetFlow 流,而是需要部署在网络边界上捕获包含 payload 的流量,这涉及到用户隐私问题.Fast-flux 控制服务器检测系统的输入是可疑域名,因此仅用单点探测即可,成本很低.基于协议特征的检测面向的协议很有限,只能检测少数特定类型的僵尸网络,不能广泛应用.

2.4 基于增值网络攻击检测

僵尸网络的主要用途就在于发起增值网络攻击.Symantec 的年度安全报告显示,2009 年度世界上 85%左右的垃圾邮件源自僵尸网络[3].因此,可以通过检测垃圾邮件间接发现僵尸网络,通过定位垃圾邮件源头来发现僵尸程序是目前常用的方法.由僵尸网络发出的垃圾邮件,具有以下特性,使之成为可检测点:1)大量邮件具有内容相似性[34];2)邮件账号注册异常,即同一 IP 地址在短时间内注册大量邮件账号[35];3)邮件发送地点变化,即同一账户在短期内跨 AS 域发送邮件[35];4)同一 IP 除发送垃圾邮件外,一般还会表现出其他攻击行为[34];5)垃圾邮件和合法邮件的流量模型存在差异[34].显然,定位 DDoS 攻击的真实源地址,也可检测僵尸网络[36].检测 DDoS 攻击方面已有大量文献,本文不再赘述.利用增值网络攻击检测僵尸网络具有很好的适应性,僵尸网络若要绕过该检测方法,只能以减少增值网络攻击为代价.

3 追踪技术

追踪的目的是发现控制者的攻击意图和僵尸网络的内部活动,即掌握控制者发布的控制命令及其触发的僵尸程序动作.追踪僵尸网络需要解决若干问题:1)C&C 协议提取的正确性、高效性和全面性.目前实用化的方法是人工分析,但这种方法对人员的技术水平要求很高,而且既耗时又易错.2)加密 C&C 协议的适应性.很多僵尸网络采用加密 C&C 信道——控制命令经过签名,敏感信息经过加密.此时,追踪的准备工作不仅包括掌握 C&C 协议的语法和语义,还需要掌握加解密所需的密钥和算法.3)追踪过程的隐秘性与匿名性.僵尸程序对每个控制命令都有确定的响应规则,僵尸程序维持 C&C 也具有一定的通信特征,防御者在追踪过程中稍有不慎,就可能被控制者察觉.

追踪的前提是掌握 C&C 协议,基于所掌握的 C&C 协议,可以采用的追踪方法可归纳为 2 类:1)以渗透的方式加入到僵尸网络中以求掌握僵尸网络内部活动情况,这种渗透的行为主体称之为僵尸网络

渗透者;2)在可控环境中运行 Sandbot. 上述 2 种追踪方法都需要先掌握 C&C 协议,且渗透方式需掌握更全面细致的 C&C 协议.

3.1 基于未知协议自动逆向的 C&C 协议提取

为提高 C&C 协议分析的正确性、高效性与全面性,Caballero 等人[37]提出了一种自动化的未知协议逆向分析技术,并实现了相应的工具 Dispatcher. 在可控环境中运行 Sandbot,利用污点传播(taint propagation)技术监视僵尸程序出入网络的数据,根据污点数据被用于关键系统函数的参数情况判断该污点数据的语法和语义. 例如,一个 4 B 长的网络数据被作为第 2 个参数传递给 socket connect 函数,则该数据很可能代表 IP 地址. 实验结果表明,利用 Dispatcher 可成功分析出 MegaD[37-38]的 C&C 协议. Cho 等人[39]将高延迟网络完整协议推测技术应用到僵尸网络的 C&C 协议提取. 不同于 Dispatcher 需要先运行 Sandbot 并监测学习网络流量,此方法需要编写一个"Bot Emulator"来构造大量语法测试请求发送给控制服务器,然后分析服务器响应. 因此,该方法需要事先了解控制服务器信息,并需要大量的网络访问. 作者优化了 L* 算法以便降低访问量和缩短推理时间. 实验表明,使用传统的 L* 算法需要 4 天多的时间来推理 MegaD 命令控制协议,改进的算法所需要的时间减为原来的 1/9.

3.2 基于 Infiltrator 思想的追踪

定义 5. 僵尸网络渗透者. 通过模仿真正僵尸程序的 C&C 协议来渗入僵尸网络,从而可以收到控制者和僵尸程序发来的控制信息,并获取信息源的地址. 将具有这样功能的程序称为僵尸网络渗透者(Infiltrator). Infiltrator 相当于被追踪僵尸程序的局部拷贝,即删除了攻击模块的僵尸程序.

Infiltrator 是一种确定的、可控的追踪方法,只要 Infiltrator 正确模拟了真实僵尸程序的 C&C 协议,就可以与其他僵尸程序获得相同的来自控制者的控制信息. 与真实僵尸程序不同的是:Infiltrator 收到控制命令后,不会产生相应的攻击动作,仅在必要时反馈伪造的执行结果. Infiltrator 需严格遵守原有 C&C 协议,如果 Infiltrator 的行为偏离了僵尸程序应有的轨道,则会引起控制者察觉进而招致攻击. 因此,为了自我保护,Infiltrator 通常利用诸如 Tor 的匿名通信系统间接联网.

针对 IRC 僵尸网络,Rajab[40]和 Freiling[41]等人都在掌握 IRC 僵尸网络 C&C 协议基础上,通过 Infiltrator 渗入僵尸网络记录其内部活动. Cho 等人[38]在自动获取 MegaD C&C 协议的基础上,通过 Infiltrator 对 MegaD 进行长达 4 个月的追踪. 通过追踪,不仅及时掌握了发送垃圾邮件相关指令和邮件模板,还结合 Google Hacking 获得了 MegaD 完整的、演进中拓扑结构. 更进一步,通过分析不同控制服务器的垃圾邮件策略,可以发现 MegaD 是被 2 组不同控制者管理的.

3.3 基于 Sandbot 思想的追踪

在可控环境中运行 Sandbot,并对其通信内容进行审计,从而可获知僵尸网络的活动. Sandbot 的优点是可以在不掌握 C&C 协议时即可迅速开始追踪,但掌握了 C&C 协议有助于提高追踪有效性. 例如,对于中心结构僵尸网络,可以预先提取所有的控制服务器域名,然后并发运行多个 Sandbot 实例,并通过网关设置使得每个 Sandbot 连接不同的服务器,以增加追踪结果的全面性. 因为不同的控制服务器参与的攻击行为未必相同,同一服务器也可能根据僵尸程序地理位置分布不同而分发不同命令. Sandbot 方法也存在明显的弊端:1)任何可控环境都无法完全避免 Sandbot 的危害. 尽管 Sandbot 运行在可控环境中,但可控环境不能判定 Sandbot 通信内容是攻击性的还是维持 C&C 所必须的,可控环境可能放行了恶意攻击流量、封锁了必要的 C&C 流量,也可能错误地篡改了关键内容,导致被控制者察觉. 2)难以处理加密协议. 常规防火墙不能识别加密流量,审计系统也只能记录控制命令密文. 3)对于严格检查运行环境的僵尸程序以及严格检查僵尸程序来源的控制服务器,Sandbot 方法可能会失效. 例如,有的僵尸程序会检查虚拟机,对于这种情况,可以在真实计算机上运行 Sandbot. 但僵尸程序还可能检查运行环境中的鼠标键盘活动、用户上网记录、安装的软件数量和路径等,从而易于判定该计算机并非真实的用户机. Sandbot 与僵尸程序之间的关系如图 4 所示.

Caballero 等人[37]利用 Dispatcher 成功分析出 MegaD 的 C&C 协议,然后利用 Sandbot 对 MegaD 实施追踪. 追踪过程中通过监控系统重写 Sandbot 发出的信息以欺骗控制服务器,使之认为 Sandbot 具备发送垃圾邮件的能力,从而使得 Sandbot 可以持续获得控制服务器新的控制信息. John 等人[42]系统化地阐述了名为 Botlab 的可控环境的设计与建设方法,并在 Botlab 可控环境中运行大量发送垃圾邮件的僵尸程序,研究其行为与特征. Grizzard[43]和 Kanich[44]等人在 VMware 虚拟机中运行 Storm

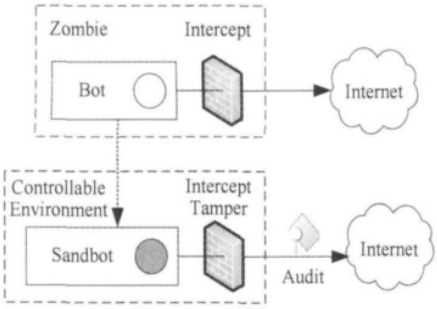

Fig. 4 Relationship between Bot and Sandbot.

图4 Bot与Sandbot之间的关系

Sandbot,并重写邮件相关信息,研究垃圾邮件成功率.类似地,Baltazar等人[45]在VMware虚拟机中运行Waledac Sandbot,通过Sniffer捕获其通信内容.

上述研究均着眼于追踪僵尸网络活动,而追踪溯源控制者方面的研究还处于萌芽状态.Ramsbrock等人[46]以IRC僵尸网络为实验对象,首次尝试了将IP溯源领域的研究成果应用到定位控制者物理位置上.监控系统在可控环境中监控Sandbot与控制者之间的通信,一旦Sandbot收到控制命令后,监控系统可以重写Sandbot对控制命令的响应报文,并在响应报文中加入4B的水印.假设控制者经过若干跳板(step-stone)连接到IRC服务器发布控制命令,如果在控制者和所用的第1个跳板之间存在上述监控系统,那么就可以定位到控制者真实IP.如果在所有跳板之间都存在监控系统,还可以还原出完整的控制路径.显然,为基本实现上述功能,需要在全球尽可能多的网络链路上部署监控系统实时监测水印.当然,这种方法理论上可行,但却难以实施.

4 测量技术

测量的目的是为了刻画僵尸网络拓扑结构、活跃规模(active size,处于在线状态的僵尸主机数量)、完全规模(total size,全部被控制了的主机,无论在线与否)等可度量属性及其动态变化轨迹,展现出僵尸网络的轮廓和特征.测量需要解决若干问题:1)当仅采用正当技术手段时,IRC和HTTP僵尸网络原理上不可测量,因为不存在测量接入点.虽然已有研究工作曾经成功测量IRC僵尸网络,但针对的是存在配置缺陷的情况.对HTTP僵尸网络的测量需要借助协调手段(如域名重定向、物理服务器控制权接管)才有可能.2)测量P2P僵尸网络时可能会受到合法P2P节点、研究者部署的节点(如Crawler,Sybil,Poisoner)以及DHCP,NAT和防火墙等因素的干扰.其中,合法P2P节点、Crawler节点、Sybil节点和DHCP会使得测量结果偏大,而NAT和防火墙又会使得测量结果偏小,Poisoner节点对测量结果的影响不可确定,取决于污染策略.3)僵尸网络会受到所处时区、开关机、新感染与被查杀、复制与迁移、防御者遏制等因素的影响,导致僵尸网络的活跃规模的剧烈抖动.实践证明,针对同一僵尸网络采用不同的测量方法,产生的结果可能相差一个数量级[13,47].所以,测量结果必须辅之以特定环境、特定策略、特定方法等的上下文说明才有参考意义.

当前的测量方法可归纳为5类:1)基于Crawler思想的测量;2)基于Sybil思想的测量;3)基于PeerlistPoisoner的测量;4)基于Sinkhole思想的测量;5)基于协议特性的局部测量.当然,无论哪种测量方法,都需要先获得和分析僵尸程序样本,掌握其C&C协议.

4.1 基于Crawler思想的测量

Crawler是以Infiltrator为基础发展而来的,可用于主动测量P2P僵尸网络.Crawler模仿真实僵尸程序的C&C协议主动联系Peer,记录该Peer返回的其他Peer地址后,重复执行上述操作,从而可遍历整个P2P网络.为了提高效率,Crawler并发访问大量目标Peer并增加通信频率,从而可以在短时间内定位大量Peer以实现测量.Crawler与P2P Infiltrator的侧重点不同:渗入P2P僵尸网络的Infiltrator按照预定寻址算法联系Peer,直到获得控制命令为止;而Crawler并不关注获得控制命令,而是持续运行直到不能发现新Peer.Crawler的弊端是不能遍历位于防火墙和NAT设备内的僵尸主机,从而造成漏报.

Kanich等人[11]编写了名为Stormdrain的Crawler测量Storm僵尸网络,分析了DHCP、NAT、防火墙和来自其他研究者的探测节点等因素造成的干扰,并论述了利用IP地址记数和利用Node ID记数的优缺点.

4.2 基于Sybil思想的测量

基于Sybil攻击的测量也是以Infiltrator为基础发展而来的,用于被动测量基于DHT思想的P2P僵尸网络.Sybil节点在具有公网可达IP的计算机上模仿大量僵尸主机被动等待来自其他Peer的通信请求,从而可以记录Peer的地址.发起Sybil

攻击需要作3方面准备:1)首先需利用Crawler获取大量Peer地址,然后由Sybil节点主动联系这些Peer,从而使得Sybil节点出现在目标Peer的邻居节点表中.可见Crawler是Sybil攻击的必要组成部分.2)Sybil节点需长期在线,以获得较高的信誉值,从而更易于被其他Peer选中为通信目标.3)Sybil节点需根据待攻击的Key空间生成具有相应Node ID(接近Key的Hash值)的P2P节点,使得查询请求被路由到Sybil节点.

Holz[10]和Kang[48]等人在分析了Storm僵尸网络C&C协议基础上,采用Crawler和Sybil相结合的方法对其进行测量.Kang等人[48]还发现在联系Sybil节点的僵尸程序中,有超过40%位于防火墙或NAT设备内.而且,Crawler可以探测到的僵尸主机几乎全部可通过Sybil节点发现.此外,Kang等人还利用"不均匀球罐模型"估算了利用Sybil攻击获取的完全规模(total size)与真实情况的比例.

4.3 基于PeerlistPoisoner的测量

针对自定义的、需交换peer-list的P2P僵尸网络,可以伪装为具有公网可达IP的僵尸主机向其他Peer推送虚假peer-list,从而污染其peer-list,当僵尸程序联系虚假Peer时,将会被记录,将这种具有污染peer-list功能的行为主体称为PeerlistPoisoner.需要说明,PeerlistPoisoner不仅可用于测量,也可以用于对抗(第6节).

Stock等人[13]编写了名为Walowdac的PeerlistPoisoner用来模拟Waledac僵尸网络中的Repeater(具有公网可达IP的僵尸主机)向大量Peer推送虚假peer-list,从而可被Spammer(防火墙或NAT设备内的僵尸主机)和Repeater主动联系,从而可记录其地址.

4.4 基于Sinkhole思想的测量

中心结构僵尸网络一般通过硬编码在僵尸程序内部的域名(domain name)或域名生成算法(domain generation algorithm, DGA)来寻址中心控制服务器.因此,可利用域名重定向或抢注域名方法将控制服务器域名解析到防御者部署的服务器(称为Sinkhole)上来,进而在Sinkhole上记录与之联系的僵尸主机地址.虽然中心结构的僵尸网络可以利用数字证书来鉴别控制服务器的真实性,但在僵尸程序试图从Sinkhole处获取数字证书时,其IP已经暴露给Sinkhole,从而使得测量成为可能.

Stone-Gross[5]预先注册了Torpig僵尸网络将会使用的域名,搭建Sinkhole伪装为控制服务器,不仅发现了18万个Torpig僵尸主机地址,还收集了70 GB的敏感信息.Dagon等人[49]利用Sinkhole对大量僵尸网络进行了测量.

4.5 基于协议特性的局部测量

基于Fast-flux思想的僵尸网络,僵尸程序硬编码的控制服务器(mothership)的域名是固定的,但其指向的是不断变化的、具有公网可达IP的僵尸主机(Fluxbot).因此,可以轮询控制服务器域名枚举Fluxbot.基于IRC和HTTP协议的僵尸网络,僵尸程序在解析控制服务器域名时,在其查询的域名服务器的缓存中会留下查询记录(在TTL时间范围内).因此,防御者可以探测大量域名服务器缓存,非递归地查询特定僵尸网络控制服务器域名,如果命中则说明至少有一个僵尸程序曾经请求该域名服务器解析该域名,将这种技术称为"DNS snooping".显然,DNS snooping只能估算中心结构僵尸网络的规模下限.

4.6 小 结

Crawler和Sybil的测量方法建立在Infiltrator的基础之上.僵尸程序、Infiltrator、Crawler和Sybil节点四者关系如图5所示:

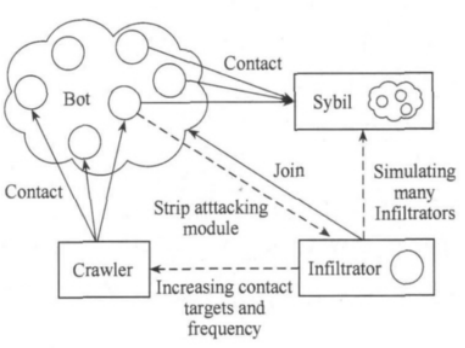

Fig. 5 Relationship among Bot, Infiltrator, Crawler and Sybil.

图5 Bot,Infiltrator,Crawler和Sybil之间的关系

实际上,僵尸网络测量技术大多是早已存在的网络测量技术的应用.对于中心结构僵尸网络,可以采用Sinkhole测量,准备工作包括域名重定向或域名抢注,分别适用于僵尸程序硬编码域名和动态生成域名的情况;对于基于DHT思想的P2P僵尸网络,可以采用Crawler与Sybil相结合的测量方法;对于需要交换peer-list信息的P2P僵尸网络,可以利用PeerlistPoisoner进行测量.Sinkhole,Sybil和PeerlistPoisoner都可以被动监听,这种能力可以有效解决防火墙和NAT设备内僵尸主机漏报问题.综合应用上述方法,可以取得更好的测量效果.

5 僵尸网络预测

僵尸网络区分于其他网络攻击形式的独特属性是其具有命令控制信道,一旦命令控制信道失效,僵尸网络将会瘫痪,从而降级为离散的、孤立的感染节点集合.因此,命令控制信道是攻防双方争夺主控权的关键点.命令控制信道面临的潜在威胁来自执法部门、CERT 组织、域名注册商、网络提供商、研究机构和竞争者等.据媒体报道,Waledac,MegaD,Mariposa 和 Rustock 等声名显赫的僵尸网络都曾受到过来自上述部门的反制,致使控制者失去控制权.因此可以预期,控制者必须持续设计更优的命令控制信道以对抗各种潜在攻击,确保即使在跨国、跨部门协作时,依然可以将控制命令在可容忍时间内分发给绝大部分僵尸程序.因此,作为防御者,必须预先考虑未来僵尸网络可能的构建方法并提前考虑应对方法.

僵尸网络预测的对象就是命令控制信道可能的设计方法.当前,预测研究包括 2 个方向:1)以个人计算机为攻击目标的高级僵尸网络构建技术;2)以智能手机为主要攻击目标的移动僵尸网络(mobile botnet)构建技术.前者旨在消除已存在的僵尸网络具有的脆弱性,设计具有表 1 中尽可能多的属性的高级僵尸网络.后者缺乏大量实际移动僵尸网络作为参照物,所以偏重于针对智能手机和计算机的差别提出移动僵尸网络可能的设计方法.

Table 1 The Design Objective of Botnets

表 1 僵尸网络设计目标

Object	Property	Capability
Controllability	Robustness	The capability to counter (crucial) nodes disabled
	Recoverability	The capability to recover C&C channel in case of temporal "shutdown"
Availability	Monitoring	The capability to monitor the status of bots and active size
	Efficiency	The speed to transfer commands to majority of bots
	Scalability	The capability to manage large-scale botnets
	Selectivity	The capability to manage specific sub-botnets
Confidentiality	Stealth	The capability to counter botnet detection, tracking, and measurement system
	Confidentiality	The capability to counter information analysis
	Cost-Effective	The capability to minimize the cost of resources
Authenticity	Anti-Hijacking	The capability to counter "Fake-Command Injection Attack"
	Intellisense	The capability to identify honeypots, infiltrators, sybils, and crawlers etc

5.1 基于 PC 的高级僵尸网络预测

Wang 等人[7]提出了一种组合结构僵尸网络的构建方法.该方法的主要思想是僵尸网络在扩散过程中,自组织为如图 3(a)所示的组合结构.该设计的优势在于:1)健壮性.整个僵尸网络不依赖特定服务器和域名的支持,Bootstrap 过程中依赖的初始节点更新速度也很快,从而消除了单点失效.2)扩展性.可管理极大规模的僵尸网络而不会出现性能瓶颈.该方法的局限在于命令控制信道的构建过程要求传播源必须掌握潜在僵尸主机的 IP,因此仅对溢出漏洞、猜口令等少数传播方法有效,对钓鱼邮件、P2P 文件共享、U 盘等传播方法无效,因为这些传播方法无法获知潜在僵尸主机的 IP 地址.

Vogt 等人[50]提出了一种组合结构的僵尸网络设计方法,该方法的主要思想是自动把一个大规模僵尸网络划分成多个僵尸子网,僵尸子网之间通过 P2P 协议连接起来,如图 3(b)所示.该方法的优势在于:1)隐秘性.多个僵尸子网对外表现独立、又可统一受控.2)扩展性.可管理极大规模的僵尸网络而不会出现性能瓶颈.但是,作者假设多数僵尸主机具有公网可达 IP 并且很多计算机会被重复感染,而这样的假设是缺乏理论与实验根据的.

为对抗防御者部署的蜜罐,Wang 等人[51]提出了一种蜜罐识别方法.该方法假设防御者部署的蜜罐一定不允许对外发起攻击,这样,控制者可以从僵尸网络中筛选出少量公网可达主机作为"Sensor",用于接收新感染主机发来的攻击,如果成功收到攻击,则认为攻击源不是蜜罐,并通知控制服务器允许该节点加入僵尸网络.类似地,Hund 等人[52]提出了一种僵尸主机信誉评估机制,通过验证一个僵尸主机是否可以成功对外发起攻击决定是否增加其信誉分,信誉分越高,说明其可信度越高,即不是蜜罐的

可能性越大.上述2种设计具有一定的智能感知能力.

常规的基于DHT的P2P僵尸网络易受索引污染攻击,为解决该问题,Starnberger等人[53]设计了Overbot协议.每个僵尸程序都会产生一个私有的并且随着时间而变化的Index Key,这样,防御者很难预测,也不可能为每个僵尸程序实时计算其Index Key,使得索引污染难以实施.但正因如此,用来发布控制命令的Sensors必须高负载地、高流量异常地为每个僵尸程序发布不同的〈Hash Key,Command〉.但是,如果具有这样高负载、高异常的中心节点(sensors)都具有生存能力,采用P2P的出发点便不再成立,而完全可以直接采用简单高效的纯中心结构.

此外,Singh等人[54]提出和评估了Email用于僵尸网络命令控制信道的可行性.王威等人[55]提出了利用UserID(用户身份)实现命令控制的思路并以Email为例评估了其可行性.Hund等人[52]首次提出了僵尸网络租赁机制,该机制允许控制者把僵尸网络的一部分主机、一部分功能、在一定的时间范围内向外租赁.其核心思想将承租方的公钥、允许承租方使用的控制命令集合(如{DDoS,Spam})、出租时间用控制者私钥签名,将证书交给承租方使用.僵尸程序收到带有数字证书和承租方私钥签名的带参控制命令(如DDoS#IP)后,验证其来源、有效期无误后执行.

5.2 基于智能手机的移动僵尸网络预测

智能手机与PC硬件结构和应用场景的差异,使得在智能手机上构建僵尸网络存在若干新的挑战:1)智能手机的电量是有限的,用户对电量消耗速度也有经验性预期,如果电池耗电速度超过了用户期望值,将构成电池消耗攻击,引起明显异常[56];2)如果C&C过程会造成明显的资费消耗(如发送大量短信、大量使用GRPS/3G流量),将会引起明显异常;3)智能手机联网状态和IP地址都不断变化,且公网可达IP稀缺,难以建立P2P网络.目前,智能手机僵尸网络上的研究还处于初级阶段,主要集中在如何设计一个符合移动智能终端特点的、未来可能被移动僵尸网络采用的命令控制信道.

Mulliner等人[57]提出了一种SMS-HTTP混合结构的命令控制信道.主要思路是将控制命令分发分为2个步骤:第1步将加密签名过的控制命令发布到网站上;第2步通过SMS将控制命令URL推送给僵尸程序.Zeng等人[58]提出一种类似的基于SMS构建起来的P2P结构的命令控制信道.SMS作为命令控制信道具有时效性强、健壮性好、离线用户(如关机、不在服务区等)可延迟接收等优点,但利用SMS建立和维持一个可连通的僵尸网络拓扑关系,需要发送和接受极大数量的SMS,造成用户端的明显异常,例如资费消耗过快和接收了大量的垃圾短信.

Cui等人[6]提出一种基于URL Flux动态中心结构的、低资源消耗的智能手机僵尸网络的设计方法,并以Android平台为攻击目标开发了Andbot,Andbot具有可控性高、隐秘性好和资源低耗等优点.

Singh等人[59]提出并评估了使用蓝牙作为命令控制信道的可行性.蓝牙作为命令控制信道的优点在于健壮性好、又不会造成额外的资费消耗;缺点是时效性很差,蓝牙只能用于近距离通信导致该方法的通信效率很低,实用化不如其他方式强.

此外,Traynor等人[60]评估了移动僵尸网络可能对蜂窝网络造成的影响.结果表明,一个小规模的移动僵尸网络可以对蜂窝网络造成严重的拒绝服务攻击.

5.3 小 结

虽然已经出现了大量高级PC僵尸网络,其传播速度之快、生存能力之强和命令控制信道之健壮,都已达到很高的技术水平.但是,就"选择性"、"资源低耗"和"智能感知"而言,僵尸网络的研究尚处于起步状态;就"健壮性"和"可监测"这种僵尸网络必备能力而言,很多高级僵尸网络依然还通过非技术手段获得(如bullet-proof);就"可恢复"能力而言,很多僵尸网络还在通过Pay-Per-Install实现.这些现象存在的原因是实战与研究有着不同的需求和目标,因此,从学术研究的角度,僵尸网络设计上仍然存在很多可改进的属性,有很大的研究空间.

6 僵尸网络对抗技术

利用技术手段对抗僵尸网络将其危害降至最低是僵尸网络研究的最终目标.本文所讨论的对抗是指通过技术手段开展的针对僵尸网络命令控制信道的对抗.域名重定向、域名注销、IP封锁、物理机关闭、终端杀毒、补丁分发、安全策略等协调手段不在本文讨论范围之内.

根据对抗效果可以把当前对抗研究归纳为4类:1)劫持(hijacking)僵尸网络,接管其全部控制权;2)挖掘并利用僵尸程序存在的溢出漏洞获得僵尸主机控制权,从而进行主机层面的僵尸程序清除;

3)污染(poisoning)僵尸网络,改变其拓扑结构和节点关系,从而遏制控制信息分发;4)关键节点拒绝服务攻击,降低关键节点的处理能力,从而降低僵尸网络的可用性.

6.1 僵尸网络劫持

劫持僵尸网络是最有效的对抗手段,可以伪装控制者向僵尸网络注入良性控制命令(如自删除命令、下载运行专杀工具命令),从而不仅可以清除整个僵尸网络,还可以接管其控制权.在IRC僵尸网络时期,僵尸程序通过频道密码、认证口令、管理权限等认证控制者身份,使其可以被劫持.HTTP和P2P僵尸网络,通常使用私钥对控制命令签名,这使得劫持从原理上不可能,除非C&C协议存在认证漏洞.例如:1)防御者伪装控制者身份发布控制命令,但僵尸程序不能鉴别控制命令真伪;2)防御者替换可被僵尸网络下载执行的文件,并重放真实的控制命令,但僵尸程序不能鉴别可执行文件真伪或者不能验证控制命令是否过期;3)防御者搭建Sinkhole吸引僵尸程序访问,但僵尸程序未对服务器身份进行认证.

TorPig和Kraken僵尸网络使用Domain Flux技术定位控制服务器,但未验证控制服务器身份.Stone-Gross[5]和Amini[61]等人利用该脆弱性,通过搭建Sinkhole分别成功劫持了Torpig和Kraken僵尸网络.事实上,目前国内存在大量利用网页挂马传播的Downloader,下载可执行文件后不进行认证就直接执行,从而可被利用于以文件替换的方式来实现劫持.

6.2 僵尸程序漏洞攻击

僵尸程序与其他程序一样,也会存在各种漏洞.如果僵尸程序实现的C&C协议存在缓冲区溢出漏洞,就可以直接向僵尸程序发送Shellcode或者构造特殊的控制命令等待其主动读取,从而造成僵尸程序缓冲区溢出而执行任意代码,从而达到接管僵尸主机控制权的目的.

Cho等人[62]通过Infiltrator渗入MegaD僵尸网络,利用类似于Fuzzing的技术对MegaD C&C协议进行漏洞挖掘,发现其存在内存越界读取漏洞和拒绝服务漏洞.Conficker.B使用的MD6算法存在缓冲区溢出漏洞,尽管该漏洞难以成功利用,但这说明,即使是十分成功的僵尸网络,依然存在被溢出的可能.2004年出现的震荡波蠕虫(sasser)的FTP服务模块就存在缓冲区溢出漏洞导致其广泛被攻击.

6.3 僵尸网络污染

僵尸网络拓扑结构决定了控制信息的流动路径,所以,改变拓扑结构就可以影响控制信息的流动.1)针对纯中心结构僵尸网络,利用Sinkhole攻击方法可将真正的控制服务器节点移除,Sinkhole取而代之,从而改变所有僵尸程序的通信路径,使得僵尸程序延迟获取甚至无法获取控制命令;2)针对基于DHT的P2P僵尸网络,可以用Sybil节点吸引僵尸程序与之通信,也可以利用索引污染攻击导致僵尸程序错误寻址,这样,就改变了控制信息的流动路径,降低其时效性;3)针对需要peer-list交换的P2P僵尸网络,当Peer之间交换信息时如果未对数据来源作认证,则可以向目标Peer注入伪造的Peer地址,从而对其污染.

国内外的研究者通过搭建Sinkhole吸引Conficker访问,虽然主要目的是对其测量,但也延缓了Conficker发现授权控制服务器的速度.Davis等人[63]利用Sybil攻击对Storm僵尸网络实施拒绝服务攻击.Waledac僵尸网络自定义的P2P协议中,Peer之间交换peer-list没有认证机制,微软公司协同学术研究机构对其Repeater节点的peer-list进行污染,导致其命令控制信道失效.

6.4 关键节点拒绝服务攻击

当上述攻击难以实施时,可以对僵尸网络关键节点实施拒绝服务攻击(DoS):1)可以对纯中心结构僵尸网络的控制服务器实施DDoS攻击,降低其可用性[63];2)对于需要Bootstrap的纯P2P僵尸网络,可以重点针对预先编码在僵尸程序中的IP列表进行DoS攻击,使得新感染的僵尸主机难以成功加入P2P网络;3)对于Fast-flux僵尸网络,除攻击Mothership以外,还可以重点攻击权威域名服务器,权威域名服务器是Fast-flux僵尸网络的脆弱点.

6.5 小 结

C&C协议的脆弱性分析可以看作漏洞挖掘技术在僵尸网络对抗上的应用.利用所发现的认证环节漏洞,可以劫持僵尸网络;利用所发现的远程溢出漏洞,可获得僵尸主机控制权.僵尸网络污染与关键节点拒绝服务攻击都可以看作针对僵尸网络的拒绝服务攻击,以降低其可用性.前者侧重技巧攻击,后者则属于流量攻击.各种对抗手段对不同拓扑结构的僵尸网络有效性不同,如表2所示.需要说明:1)Sinkhole攻击与中心服务器DoS攻击虽然原理上对URL Flux有效,但不具有现实可操作性,因为URL Flux中的控制服务器一般选择大型Web 2.0网站.2)peer-list污染攻击对基于DHT的P2P网络有一定效果,而对于自定义的、需要交换peer-list的P2P网络尤为有效.

Table 2 The Comparison of C&C Protocols

表 2 命令控制协议对抗能力对比

Countermeasures	Topology						
	IRC	HTTP	URL Flux	Fast-flux	Domain Flux	DHT	Random
Fake-Command Injection	√	√	√	√	√	√	√
Vulnerability Overflow	√	√	√	√	√	√	√
Index Poisoning	×	×	×	×	×	√	×
Sinkhole	√	√	×	√	√	×	×
Sybil	×	×	×	×	×	√	√
peer-list Pollution	×	×	×	×	×	√	×
C&C Servers DoS	√	√	×	√	√	×	×
Bootstrap Attack	×	×	×	×	×	√	×

Note: "√" represents the countermeasure is effective; "×" means noneffective.

7 结论与研究空间

僵尸网络作为一种网络攻击平台,长期以来成为攻击者手中最有效的武器,而随着网络融合时代的到来,僵尸网络的威胁又将延伸到更广阔的网络空间,僵尸网络正吸引着世界各国攻防人员投入更多的研究. 本文回顾了僵尸网络的发展历程,剖析了僵尸网络产生和发展的根源与内在驱动力,形式化定义了僵尸网络并按照拓扑结构将僵尸网络分为纯中心结构、纯 P2P 结构、组合结构和混合结构. 本文将当前僵尸网络研究热点归纳为检测、追踪、测量、预测和对抗 5 个环节,分别阐述了各方向的研究进展. 作为防御者,需要系统化地掌握僵尸网络研究进展,研究僵尸网络在攻防对抗中的演进方向,提取僵尸网络存在的不可绕过的脆弱性,进而有针对性地对抗僵尸网络. 同时,还需研究未来僵尸网络可能采用的攻击技术并预先构筑防御体系.

通过本文研究,可以得到如下结论:1)僵尸网络必须利用命令控制信道传递控制信息,而命令控制协议的设计可能存在漏洞,可以挖掘并利用命令控制协议漏洞清除用户终端上的僵尸程序甚至获得僵尸网络控制权;2)僵尸网络必须允许新节点加入,这导致任何僵尸网络都不能避免被追踪,利用 Infiltrator 追踪僵尸网络具有普适性;3)基于 Domain Flux,Fast-flux 的纯中心结构和基于 DHT 思想的纯 P2P 结构僵尸网络都存在固有脆弱点,可以对这类僵尸网络测量;4)域名系统是很多僵尸网络依赖的重要资源,掌握域名系统的控制权可以有效打击僵尸网络.

伴随信息技术的不断发展,僵尸网络攻击技术一定也会随之进化. 预先考虑未来可能出现的攻击技术,有利于提前设计优化的防御方法. 本文对僵尸网络攻击技术未来发展方向简要展望如下:1)僵尸网络局部化管理. 将大规模僵尸网络按照特定条件划分为多个子网,从而可以选择性地管理特定子网并增加隐秘性. 2)僵尸网络时空异构. 使僵尸程序之间及僵尸程序与控制服务器之间的通信不具备时空相似性,从而使得已有的利用该属性的研究成果失效. 3)智能感知. 可以发现 Honeypot,Infiltrator 和 Sybil 等渗入僵尸网络的"间谍"节点;可以自动识别重要用户并按预定策略发起增值网络攻击. 4)手机僵尸网络. 针对主流手机操作系统,以牟取经济利益和窃取个人信息的手机僵尸网络将迅速发展,重现 PC 僵尸网络发展历程. 5)攻击位于物理隔离专网的特定系统. 面向网络战,通过移动介质渗入物理隔离专网内的计算机,并攻击与之相连的特定系统(如工业系统、金融系统和军事系统). 可以预见,未来的高级僵尸网络完全可以做到在跨国和多方协作对抗下的生存,其攻击目标将延伸到更广阔的网络空间,并对更多用户、社会经济乃至国家安全造成更直接和更严重的威胁,僵尸网络攻防对抗是一个不断演进和长期相伴的过程.

参 考 文 献

[1] Porras P, Saidi H, Yegneswaran V. A foray into Conficker's logic and rendezvous points [R/OL]. Berkeley, CA: USENIX, 2009. [2011-06-10]. http://www.usenix.org/events/leet09/tech/full_papers/porras/porras_html/

[2] CNCERT. 中国互联网网络安全报告[EB/OL]. 2011. [2011-06-10]. http://www.cert.org.cn/UserFiles/File/2010%20first%20half.pdf. 2010

[3] Symantec Inc. Symantec global Internet security threat report trends for 2009 volume XV [EB/OL]. 2010. [2011-06-10]. http://eval. symantec. com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010. en-us. pdf

[4] Holz T, Gorecki C, Rieck C, et al. Detection and mitigation of fast-flux service networks [C] //Proc of the 15th Annual Network and Distributed System Security Symposium. Berkeley, CA: USENIX, 2008

[5] Stone-Gross B, Cova M, Cavallaro L, et al. Your botnet is my botnet: Analysis of a botnet takeover [C] //Proc of the 16th ACM Conf on Computer and Communications Security. New York: ACM, 2009: 635-647

[6] Cui Xiang, Fang Binxing, Yin Lihua, et al. Andbot: Towards advanced mobile botnets [C] //Proc of the 4th Usenix Workshop on Large-scale Exploits and Emergent Threats. Berkeley, CA: USENIX, 2011: No 11

[7] Wang P, Sparks S, Zou C C. An advanced hybrid peer-to-peer botnet [C] //Proc of the 1st Conf on 1st Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2007: No 2

[8] Wang Wei. Research on countermeasure techniques for the botnet [D]. Harbin: Harbin Institute of Technology, 2010 (in Chinese)
(**王威. 僵尸网络对抗技术研究**[D]. **哈尔滨: 哈尔滨工业大学**, 2010)

[9] Zhuge Jianwei, Han Xinhui, Zhou Yonglin, et al. Research and development of botnets [J]. Journal of Software, 2008, 19(3): 702-715 (in Chinese)
(**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)

[10] Holz T, Steiner M, Dahl F, et al. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm [C] //Proc of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats. Berkeley, CA: USENIX, 2008: No 9

[11] Kanich C, Levchenko K, Enright B, et al. The Heisenbot uncertainty problem: Challenges in separating bots from chaff [C] //Proc of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats. Berkeley, CA: USENIX, 2008: 1-9

[12] Ramachandran A, Feamster N. Understanding the network-level behavior of spammers [C] //Proc of the 2006 Conf on Applications, Technologies, Architectures and Protocols for Computer Communications. New York: ACM, 2006: 291-302

[13] Stock B, Engelberth M, Freiling F C, et al. Walowdac analysis of a peer-to-peer botnet [C] //Proc of the 2009 European Conf on Computer Network Defense. Washington, DC: IEEE Computer Society, 2009: 13-20

[14] Axelle A. Symbian worm Yxes towards mobile botnets [EB/OL]. 2010. [2011-06-10]. http://www. fortiguard. com/papers/EICAR2010 _Symbian-Yxes _Towards-Mobile-Botnets. pdf

[15] Porras P A, Saidi H, Yegneswaran V. An analysis of the iKee.B iPhone botnet [C] //Proc of the 2nd Int ICST Conf on Security and Privacy on Mobile Information and Communications Systems. Berlin: Springer, 2010: 141-152

[16] Tim Wyatt. Security alert: Geinimi, sophisticated new android trojan found in wild [EB/OL]. 2010. [2011-06-10]. http://blog. mylookout. com/2010/12/ge inimi_trojan/

[17] Nicolas F, Liam O, Eric C. W32. Stuxnet dossier [EB/OL]. 2011. [2011-06-10]. http://www. symantec. com/content/en/us/enterprise/media/security _response/whitepapers/w32 _stuxnet_dossier. pdf

[18] Xie Y, Yu F, Achan K, et al. Spamming botnets: Signatures and characteristics [C] //Proc of ACM SIGCOMM'08. New York: ACM, 2008: 171-182

[19] Gu G, Perdisct R, Zhang J, et al. BotMiner: Clustering analysis of network traffic for protocol - and structure-independent botnet detection [C] //Proc of the 17th USENIX Security Symp. Berkeley, CA: USENIX, 2008: 269-286

[20] Cook E, Jahanian F. The zombie roundup: Understanding, detecting, and disrupting botnets [C] //Proc of the 1st USENIX Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2005: 39-44

[21] Leder F, Werner T, Martini P. Proactive botnet countermeasures—An offensive approach [R/OL]. Tallinn, Estonia: Cooperative Cyber Defence Centre of Excellence (CCDCOE), 2009. [2011-01-14]. http://www. ccdcoe. org/publications/virtualbattlefield/15_LEDER_Proactive_Coutner-measures. pdf

[22] Wang P, Wu L, Aslam B, et al. A systematic study on peer-to-peer botnets [C] //Proc of Int Conf on Computer Communications and Networks (ICCCN). Washington, DC: IEEE Computer Society, 2009: 1-8

[23] Baecher P, Koetter M, Holz T, et al. The nepenthes platform: An efficientapproach to collect malware [C] //Proc of Int Symp on Recent Advances in Intrusion Detection. Berlin: Springer, 2006: 165-184

[24] Zhuge Jianwei, Han Xinhui, Zhou Yonglin, et al. HoneyBow: An automated malware collection tool based on the high-interaction honeypot principle [J]. Journal on Communications, 2007, 28(12): 8-13 (in Chinese)
(**诸葛建伟, 韩心慧, 周勇林, 等**. HoneyBow: **一个基于高交互式蜜罐技术的恶意代码自动捕获器**[J]. **通信学报**, 2007, 28(12): 8-13)

[25] Stinson E, Mitchell J C. Characterizing bots' remote control behavior [C] //Proc of the 4th Int Conf on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin: Springer, 2007: 89-108

[26] Gu G, Porras P, Yegneswaran V, et al. BotHunter: Detecting malware infection through ids-driven dialog correlation [C] //Proc of the 16th USENIX Security Symp. Berkeley, CA: USENIX, 2007: 167-182

[27] Gu G, Zhang J, Lee W. BotSniffer: Detecting botnet command and control channels in network traffic [C] //Proc of the 15th Annual Network and Distributed System Security Symp. Berkeley, CA: USENIX, 2008: 269-286

[28] Karasaridis A, Rexroad B, et al. Wide-scale botnet detection and characterization [C] //Proc of the 1st USENIX Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2007: No 7

[29] Stinson E, Mitchell J C. Towards systematic evaluation of the evadability of bot/botnet detection methods [C] //Proc of the 2nd Conf on USENIX Workshop on Offensive Technologies. Berkeley, CA: USENIX, 2008: 1-9

[30] Wang Wei, Fang Binxing, Cui Xiang. IRC botnet detection based on host behavior [J]. Chinese Journal of Computers, 2008, 32(10): 1980-1988 (in Chinese)
(王威, 方滨兴, 崔翔. 基于终端行为特征的IRC僵尸网络检测[J]. 计算机学报, 2009, 32(10): 1980-1988)

[31] Goebel J. Rishi: Identify bot contaminated hosts by IRC nickname evaluation [C] //Proc of the HotBots'07, 1st Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2007: No 8

[32] Nazario J, Holz T. As the net churns: Fast-flux botnet observations [C] //Proc of the Int Conf on Malicious and Unwanted Software. Washington, DC: IEEE Computer Society, 2008: 24-31

[33] Ching-Hsiang H, Huang C Y, Chen K T. Fast-flux bot detection in real time [C] //Proc of the 13th Int Conf on Recent Advances in Intrusion Detection. Berlin: Springer, 2011: 464-483

[34] Willa K Ehrlich, Anestis K, Liu D, et al. Detection of spam host and spam bots using network flow traffic modeling [C] //Proc of the 3th USENIX Workshop on Large-Scale Exploits and Emergent Threats. Berkeley, CA: USENIX, 2010: No 7

[35] Zhao Y, Xie Y, Yu F, et al. Gillum: Large scale spamming botnet detection [C] //Proc of the 6th USENIX Symp on Networked Systems Design and Implementation. Berkeley, CA: USENIX, 2009: 321-334

[36] Wang Hailong, Gong Zhenghu, Hou Jie. Overview of botnet detection [J]. Journal of Computer Research and Development, 2010, 47(12): 2037-2048 (in Chinese)
(王海龙, 龚正虎, 侯婕. 僵尸网络检测技术研究进展[J]. 计算机研究与发展, 2010, 47(12): 2037-2048)

[37] Juan C, Pongsin P, Christian K, et al. Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering [C] //Proc of the 16th ACM Conf on Computer and Communications Security. New York: ACM, 2009: 621-634

[38] Cho C Y, Caballero J, Grier C, et al. Insights from the inside: A view of botnet management from infiltration [C] //Proc of the 3rd USENIX Conf on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms and More. Berkeley, CA: USENIX, 2010: No 2

[39] Chia Y C, Domagoj B, Richard S, et al. Inference and analysis of formal models of botnet command and control protocols [C] //Proc of the 17th ACM Conf on Computer and Communication Security. New York: ACM, 2010

[40] Rajab M, Zarfoss J, Monrose F, et al. A multifaceted approach to understanding the botnet phenomenon [C] //Proc of the 6th ACM SIGCOMM Conf on Internet Measurement. New York: ACM, 2006: 41-52

[41] Freiling F, Holz T, Wicherski G. Botnet tracking: Exploring a root-cause methodology to prevent denial of service attacks [C] //Proc of the 10th European Symp on Research in Computer Security. Berlin: Springer, 2005: 319-335

[42] John J P, Moshchuk A, Gribble S D, et al. Studying spamming botnets using botlab [C] //Proc of the 6th USENIX Symp on Network Systems Design and Implementation. Berkeley, CA: USENIX, 2009: 291-306

[43] Grizzard B, Sharma V, Nunnery C, et al. Peer-to-peer botnets: Overview and case study [C] //Proc of the 1st Conf on First Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2007: No 1

[44] Kanich C, Kreibich C, Levchenko K, et al. Spamalytics: An empirical analysis of spam marketing conversion [C] //Proc of the 15th ACM Conf on Computer and Communications Security. New York: ACM, 2008: 3-14

[45] Jonell B, Joey C, Ryan F. Infiltrating waledac botnet's convert operation [EB/OL]. Trend Micro, 2009. [2011-06-10]. http://us. trendmicro. com/imperia/md/content/us/pdf/threats/securitylibrary/infiltrating_the_waledac_botnet_v2.pdf

[46] Daniel Ramsbrock, Xinyuan Wang, Xuxian Jiang. A first step towards live botmaster traceback [C] //Proc of the 11th Int Symp on Recent Advances in Intrusion Detection. Berlin: Springer, 2008: 59-77

[47] Rajab M, Zarfoss J, Monrose F, et al. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging [C] //Proc of the 1st Conf on 1st Workshop on Hot Topics in Understanding Botnets. Berkeley, CA: USENIX, 2007: No 5

[48] Kang B B, Eric C T, et al. Towards complete node enumeration in a peer-to-peer botnet [C] //Proc of the 4th Int Symp on Information, Computer and Communications Security. New York: ACM, 2009: 23-34

[49] Dagon D, Zou C, Lee W. Modeling botnet propagation using time zones [C] //Proc of 13th Annual Network and Distributed System Security Symp. Berkeley, CA: USENIX, 2006: 235-249

[50] Vogt R, Aycock J, Jacobson M. Army of botnets [C] //Proc of the 2007 Network and Distributed System Security Symp. Berkeley, CA: USENIX, 2007: 111-123

[51] Wang P, Wu L, Cunningham R, et al. Honeypot detection in advanced botnet attacks [J]. International Journal of Information and Computer Security, 2010, 4(1): 30-51

[52] Hund R, Hamann M, Holz T. Towards next-generation botnets [C] //Proc of the 2008 European Conf on Computer Network Defense. Washington, DC: IEEE Computer Society, 2008: 33-40

[53] Starnberger G, Kruegel C, Kirda E. Overbot—A botnet protocol based on Kademlia [C] //Proc of the 4th Int Conf on Security and Privacy in Communication Networks. New York: ACM, 2008: 1-9

[54] Kapil S, Abhinav S, et al. Evaluating email's feasibility for botnet command and control [C] //Proc of the 38th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks. Washington, DC: IEEE Computer Society, 2008: 376-385

[55] Wang W, Fang B X, Cui X, et al. A userID-centralized recoverable botnet: Structure research and defense [J]. International Journal of Innovative Computing, Information and Control, 2010, 6(4): 4307-4317

[56] Hahnsang K, Smith J, Kang G Shin. Detecting energygreedy anomalies and mobile malware variants [C] //Proc of the 6th Int Conf on Mobile Systems, Applications, and Services. New York: ACM, 2008: 239-252

[57] Mulliner C, SeifertIn J P. Rise of the iBots: Owning a telco network [C] //Proc of the 5th IEEE Int Conf on Malicious and Unwanted Software (Malware) Nancy. Washington, DC: IEEE Computer Society, 2010: 71-80

[58] Zeng Y, Hu X, Shin, K G. Design of SMS commanded-and-controlled and P2P-structured mobile botnet [R/OL]. Michigan: University of Michigan. 2010. [2011-01-20]. http://www. eecs. umich. edu/techreports/cse/2010/CSE-TR-562-10. pdf

[59] Singh K, Sangal S, Jain N, et al. Evaluating bluetooth as a medium for botnet command and control [C] //Proc of the Int Conf on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin: Springer, 2010: 61-80

[60] Traynor P, Lin M, Ongtang M, et al. On cellular botnets: Measuring the impact of malicious devices on a cellular network core [C] //Proc of the 16th ACM Conf on Computer and Communications security. New York: ACM, 2009: 223-234

[61] Amini P, Pierce C. Kraken Botnet infiltration [EB/OL]. Blog on DVLabs. 2008. [2011-06-10]. http://dvlabs. tippingpoint. com

[62] Chia Y C, Juan C. Botnet infiltration: Finding bugs in botnet command and control [EB/OL]. 2009. [2011-06-10]. http://www. eecs. berkeley. edu/~chiayuan/cs261

[63] Davis C R, Fernandez J, Neville S, et al. Sybil attacks as a mitigation strategy against the storm botnet [C] //Proc of the 3rd Int Conf on Malicious and Unwanted Software. Washington, DC: IEEE Computer Society, 2008: 32-40

Fang Binxing, born in 1960. Professor and PhD supervisor. Academician of Chinese Academy of Engineering. His current research interests include computer architecture, computer network and information security.

Cui Xiang, born in 1978. PhD candidate. His research interest is network security.

Wang Wei, born in 1981. PhD. Her research interest is network security.