

面向可扩展僵尸网络的安全控制方法

刘强¹, 李鹏飞¹, 付章杰²

(1. 国防科技大学计算机学院, 湖南 长沙 410073; 2. 南京信息工程大学计算机与软件学院, 江苏 南京 210044)

摘要: 僵尸网络是互联网面临的主要威胁之一。当前, 网络服务类型多样、安全漏洞频出、以物联网设备为代表的海量联网设备部署更加有利于僵尸网络全球扩展。未来僵尸网络将更加具有跨平台特性和隐匿性, 这给网络空间带来了严重的安全隐患。因此, 针对僵尸网络自身开展深入研究, 可以为新的僵尸网络防御研究提供研究对象, 对于设计下一代网络安全防护体系具有重要意义。提出一种基于 HTTP 的可扩展僵尸网络框架来解决僵尸网络自身存在的兼容性、隐匿性与安全性问题, 该框架基于中心式控制模型, 采用 HTTP 作为僵尸网络通信协议, 并对通信内容进行基于对称密码学的块加密。进一步地, 提出了一种面向多平台架构的僵尸网络安全控制方法, 该方法利用源码级代码集成与交叉编译技术解决兼容性问题, 引入动态密钥加密通信机制克服传统僵尸网络流量存在规律性和易被分析的不足, 设计服务器迁移与重连机制解决中心式僵尸网络模型存在的单点失效问题, 以提高僵尸网络存活率。3 个不同控制性水平场景下的仿真实验结果表明, 僵尸网络的规模与其命令与控制 (C&C, command and control) 服务器服务负载之间存在线性关系; 此外, 在僵尸网络规模相同的条件下, 越高的控制性会带来越高的吞吐量和越大的系统负载, 从而验证了所提方法的有效性和现实可行性。

关键词: 僵尸网络; 安全控制; 多平台架构; 高级加密标准

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2023002

Secure controlling method for scalable botnets

LIU Qiang¹, LI Pengfei¹, FU Zhangjie²

1. College of Computer, National University of Defense Technology, Changsha 410073, China

2. School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

Abstract: Botnet is one of main threats towards the Internet. Currently, botnets can expand to the whole world due to various types of network services, pervasive security vulnerabilities and massive deployment of networked devices, e.g., internet of things (IoT) devices. Future botnets will become more cross-platform and stealthy, which introduces severe security risks to cyberspace. Therefore, in-depth research on botnets can offer study targets to corresponding defensive studies, which is very meaningful for designing an architecture to secure the

收稿日期: 2022-05-31; 修回日期: 2022-07-01

通信作者: 刘强, qiangliu06@nudt.edu.cn

基金项目: 湖南省自然科学基金 (2021JJ30779)

Foundation Item: The Natural Science Foundation of Hunan Province (2021JJ30779)

引用格式: 刘强, 李鹏飞, 付章杰. 面向可扩展僵尸网络的安全控制方法[J]. 网络与信息安全学报, 2023, 9(1): 42-55.

Citation Format: LIU Q, LI P F, FU Z J. Secure controlling method for scalable botnets[J]. Chinese Journal of Network and Information Security, 2023, 9(1): 42-55.

next-generation cyberspace. Hence, an HTTP-based scalable botnet framework was proposed to address the problems of compatibility, stealthiness and security. Specifically, the framework adopted a centralized controlling model. Moreover, it used the HTTP protocol as the designed botnet's communication protocol and block encryption mechanisms based on symmetric cryptography to protect the botnet's communication contents. Furthermore, a secure control mechanism for multi-platform botnets was designed. In particular, the proposed mechanism utilized source-level code integration and cross-compilation techniques to solve the compatibility challenge. It also introduced encrypted communication with dynamic secret keys to overcome the drawbacks of network traffic regularity and ease of analysis in traditional botnets. Moreover, it designed server migration and reconnection mechanisms to address the weakness of single-point-failure in centralized botnet models. Simulation results in three experimental scenarios with different levels of botnet controllability show that there is a linear relationship between the size of a botnet and the service overhead of the related C&C servers. In addition, under the condition of the same botnet scale, a higher level of controllability introduces a higher throughput and a greater system overhead. The above results demonstrate the effectiveness and the practical feasibility of the proposed method.

Keywords: botnet, secure control, multi-platform architecture, advanced encryption standard

0 引言

近年来,传统的社交活动逐渐转化为互联网上的线上活动,直播、游戏、短视频等不断融入人们生活,互联网行业成为近几年来持续高速增长的行业。然而,网络的高速发展带来了许多安全隐患,以分布式拒绝服务(DDoS, distributed denial of service)为代表的多样化攻击手段层出不穷,攻击活动呈增长趋势^[1]。腾讯安全联合绿盟科技发布的报告显示^[2],近两年的DDoS攻击次数迅速增长,攻击流量连续5年达到了TB级别,DDoS反射元分布与物联网(IoT)发展速度及其基数相关。欧盟网络与信息安全局(ENISA)在其发布的2021年态势报告中指出^[3],DDoS攻击变得更加有针对性与持久性,成为勒索软件用于威逼支付的一大新型手段,物联网设备加入5G网络更是进一步助长了DDoS的威力;恶意软件攻击虽然在近两年有所下降,但主流恶意软件的家族种类并未发生改变,僵尸网络仍然是其中的主要组成部分,其攻击与绕过手段较以前更为多样。根据以色列安全公司SAM Seamless Network报告^[4],2021年发生了超过9亿次与物联网相关的网络攻击,Mirai和Mozi僵尸网络还在持续增加重要的新功能,体现了物联网生态系统安全的紧迫性。针对僵尸网络自身开展深入研究,可以为新的僵尸网络防御研究提供研究对象,拓宽安全防御发展的新思路,对于设计下一代网络安全

防护体系具有重要意义。

从攻击者的角度看,实现可扩展的僵尸网络模型需要解决两个关键性问题:控制目标的选择问题和僵尸程序的存活性问题。控制目标的选择是指在实际环境中探测存在安全漏洞的攻击目标,并将僵尸程序运行在其上的过程;僵尸程序的存活性是指僵尸主机上线后的隐匿性和安全性。事实上,上述两个方面的问题直接影响了僵尸网络规模的扩展速度。另外,在物联网时代,多种类型的、驻留有不同安全漏洞的终端设备接入互联网中,为了尽可能扩大僵尸网络规模,必然需要解决僵尸程序的兼容性和通用性问题。因此,一种能够面向多种平台架构的僵尸程序设计方法是关键且必要的。当僵尸网络进入控守阶段时,由于现有的安全检测手段繁多,并且互联网用户的安全意识不断提升,传统的僵尸网络能够被有效防御。为了提高命令与控制(C&C, command and control)服务器指令的执行效率,提高攻击链整体攻击效果,需要研究隐匿且安全的僵尸网络控制方法。

基于上述分析,本文提出了一个基于HTTP的可扩展僵尸网络框架,该框架针对多种类型设备联网的现实情况,注重在代码开发层面考虑兼容性问题,从而对不同运行环境拥有较强的鲁棒性;此外,僵尸程序能够按需加载不同的功能模块,实现功能的灵活编排与相互配合,具有强大的可扩展性。进一步地,本文提出了一种面向多

平台架构的僵尸网络安全控制方法,以解决不同类型僵尸网络主机的兼容性问题 and 僵尸网络通信过程的隐匿性问题。具体来讲,该方法联合利用源码级代码集成与交叉编译技术,解决了兼容性问题;针对传统的僵尸网络流量具有规律性、易被分析的特点,该方法引入了使用动态密钥的高级加密标准(AES)加密通信机制;针对中心式僵尸网络模型存在单点失效的弱点,该方法还设计了服务器迁移与重连机制,以提高僵尸网络的存活率。本文将 C&C 服务器服务能力的压力测试抽象成为一个 Web 服务器提供用户访问服务的负载能力测试,通过不同控制性水平场景下的仿真实验检验 C&C 服务器对整个僵尸网络的群控能力,以验证本文方法的有效性与现实可行性。

1 研究现状

有关僵尸网络的研究,主要集中在对僵尸网络的行为分析及检测防御方法^[5]。僵尸网络的检测方法中,基于机器学习的检测发展逐渐占据主流,这类检测是目前僵尸网络需要直面进行对抗的防御手段。吴迪等^[6]提出 BotCatcher 检测系统,旨在解决传统机器学习需要人工选择特征的限制,从空间、时间两个维度学习僵尸网络的特征,相比单一模型具有更高的准确性;受多模型学习的思路启发,沈琦等^[7]提出了以集成学习的方式混合多种机器学习算法,通过不同学习器对不同特征学习能力的结合得到强学习期,总体提升了检测方法的检测性能;肖喜生等^[8]提出了基于流量摘要的检测方法,将原始数据流按照源主机地址聚合后,划分时间窗口生成流量摘要记录,通过构建随机森林等机器学习分类模型以达到更好的检测效果。检测方法的特点在于对僵尸网络通信流量的特征提取上,其检测效果很大程度上取决于特征集选择,目前还没有标准的指导原则来设计能够准确描述网络流量的特征集^[9]。对于僵尸网络的模拟仿真与设计开发,主要集中在根据已有的僵尸网络模型进行拓展,阐述僵尸网络未来变种的可能趋势。研究者对出现在互联网中的主流僵尸网络进行了行为及通信控制模型的分析,总结了僵尸网络模型的定义和分类,提出了

对应的检测手段与方法。对于提炼出的模型与成果,主要通过现存公开的僵尸网络源码或者公开的僵尸网络流量数据进行仿真模拟,以测试自身的防御策略效果,存在样本较为单一、缺乏变种的问题,在当前恶意程序变种繁杂的大背景下,实际效果相对减弱。通过分析僵尸网络的行为提出防御策略手段,从网络空间角色的角度而言,防守方是不可避免的。跳出单纯的防御角色,从攻击方角度进行僵尸网络行为的思考,以此开拓防御策略的思维,不妨为一种新的改进方式。自主设计仿真,预测未来恶意程序的行为趋势,在思路上有准备,有助于缓解“见招拆招”的被动防御局面。

僵尸网络的设计与仿真可以划分为基于通信协议的设计仿真和基于目标环境的设计仿真,也存在兼顾二者的相关研究,如 Wang 等^[10]提出的综合短信和 HTTP C&C 信道的移动僵尸网络模型,同时兼顾了手机移动端的环境及其通信手段,此类情况一般在通信协议与攻击环境适应性更好、配合更佳的情形下出现。严格来说,通信协议的支持依赖于目标环境,目标环境同时需要多种通信手段支撑服务,二者存在相辅相成的关系。

基于通信协议的设计仿真,主要指针对僵尸网络进行命令控制通信的信道协议模型进行开发设计,通信协议的选择主要由僵尸网络的群控模型决定。对于 HTTP 模型,相应的研究设计与仿真相对较少,但特点仍然鲜明。Knysz 等^[11]提出了“Wi-Fi Botnet”模型,其能够依赖 Wi-Fi 网络进行数据传输通信,分布广泛难以定位,拥有较强攻击能力,具有通信量小、隐匿性好的特点,主要受 Wi-Fi 接入点分布距离的限制。对于 P2P 模型,网络结构复杂多变,各类设计相对更多,其结构相对难以破坏。Wang 等^[12]提出了具有通信信道加密的高级混合 P2P 网络,对节点列表创新与更新进行了详细的介绍,但缺乏严格的认证;Han 等^[13]提出了一个高级的 P2P 僵尸网络模型并用 5 个性能指标进行评估,对僵尸网络设计进行优化;高见等^[14]提出了超级节点更新邻居节点的算法,基于该算法设计了两层架构的 P2P 僵尸网络,主要针对 P2P 网络结构进

行优化从而提升僵尸网络整体的运行效率。

基于目标环境的设计仿真,主要指针对僵尸网络期望攻击感染的目标而进行的专门设计。柳淑婷等^[15]基于 Android 设计了僵尸网络,将攻击命令加密后附加入图片,通过微博账户发布以进行命令的传达,具有较高的隐蔽性,符合移动端使用的行为特点与环境要求,但缺乏一定的通用性;张翥然^[16]基于 Mirai 病毒构建物联网僵尸网络病毒传播模型进行研究及仿真,通过对物联网节点分类来建立 Mirai 病毒的传播动力学模型,研究了僵尸网络感染期间的关键问题,对于病毒防治策略有相应启发,但缺乏多样本的实例支撑。

随着新型技术的发展,有学者运用前沿技术提出了全新的僵尸网络模型。区块链技术以自身具有弹性与可用性的特点吸引了更多攻击者的注意,让攻击者能够在现有公共区块链上搭建更坚固的僵尸网络^[17]。周安民等^[18]提出了基于比特币网络双向通信的 P2P 僵尸网络 D-BitBot 模型, D-BitBot 分为控制端、普通节点和感知节点,将通信数据嵌在不可用的比特币交易中以模拟真实的比特币节点流量,实现流量伪装,并且通过仿真测试证明了其拥有较好的鲁棒性与较大的危害性,为安全研究敲响了警钟。赵昊等^[19]提出了基于区块链智能合约的僵尸网络信道模型 SCBot,通过结合 P2P 结构与 C&C 服务器层将僵尸网络划分为独立子网,力图解决传统中心结构的单点失效问题与 P2P 模型的命令传递效率问题,并通过仿真测试证明了其具有更高的鲁棒性与效率,但存在维护成本的问题。基于新型技术的僵尸网络还在发展中,传统的僵尸网络仍是网络空间中的主要威胁。

综上所述,在仿真设计方面,基于 HTTP 的样本仿真设计相关研究较少,相关研究缺乏典型样例。仿真设计最终得出的防御策略结论需要实际样本的案例进行支撑,正向的攻击思路能够带来防御策略的扩展与发散。新型技术支撑的僵尸网络模型具有发展的新势头,但尚未成为主流,对于传统僵尸网络的研究仍然是必要且急需的。大多数进行仿真设计的学者,以更加深入地掌握恶意软件行为特征为切入点,在提炼共性规律的

基础上提出防御策略与方法,同时为防御策略的实测提供实例支撑。

2 基于 HTTP 的可扩展僵尸网络框架

2.1 设计思路

图1展示了一个基于 HTTP 的可扩展僵尸网络框架,该框架主要包括基于 HTTP 的通信与 AES 加密隐蔽、面向多平台的兼容性与鲁棒性设计以及功能可扩展性与可用性设计3个方面。

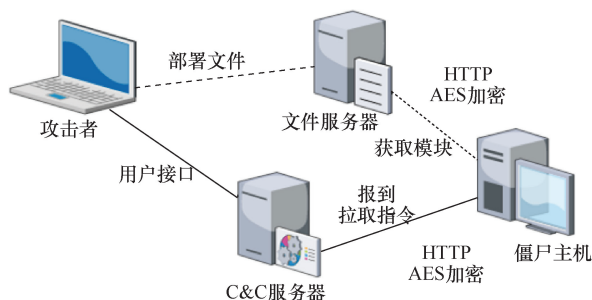


图1 基于 HTTP 的可扩展僵尸网络框架
Figure 1 The framework of an HTTP-based scalable botnet

2.1.1 基于 HTTP 的通信与 AES 加密隐蔽

中心式网络的通信模型主要采用 HTTP 模型。使用 HTTP 具有较多优点,HTTP 作为一个广泛应用的协议,僵尸网络可以将流量轻松隐藏在海量的流量中。即使被检测发现,也不能通过阻塞 HTTP 和端口来进行防御,否则会影响到大部分业务。然而,HTTP 采用明文方式传输数据,因此必须考虑其安全性与隐匿性。考虑到加解密代价及消息的时延性问题,对称密钥加密稍优于非对称密钥加密方法,并且实现上较为容易。基于上述分析,HTTP 通信使用 AES 对称密钥加密算法进行加密保护,并且根据实际情况进行方式的调整与改进。

随着使用加密通信的恶意软件不断增多,加密流量的网络检测技术逐步走向成熟。结合机器学习技术,基于加密流量的僵尸网络检测方法通过大量分析抓取的恶意流量样本,从中提取出相应的特征以将其与正常加密通信流量进行区分,其准确率取决于训练程度的深浅。针对基于机器学习的检测方法,其训练的程度取决于样本的数量及其特征的凸显程度,检测准确率与特征集的选择相关。在足够的时间下,能够抓取到足够的

恶意流量,从而达到足量样本的需求。在通信流量上做出削减,能够延缓被检测的速度,但最终仍会被彻底清除,因此对于僵尸网络的生存力提升是远远不够的。规避加密流量特征的检测,应当削弱僵尸网络通信时的规律性,从而尽可能地弱化通信行为特征。对于本文提出的僵尸网络框架,在上述加密性的基础上,需要对其僵尸网络内部通信协议进行隐蔽。对此,本文提出以下3种思路。

1) 使用 HTTP 进行通信的传输,伪装成正常的 HTTP 访问。

2) 使用隐写技术,将通信内容隐藏在伪装成 HTTP 访问反馈的网页中。

3) 使用较有灵活性的僵尸网络内部协议,通过主动的内部通信协议变化来弱化通信行为特征。

第1种思路在实际运用中的兼容性问题相对突出,需要更进一步的研究予以支撑;第2种思路的隐蔽性相对更高,但其带来的通信流量相对更大;第3种思路的扩展性较强,灵活性较好,但可能存在其扩展通信框架特征易被识别的问题,对使用者的能力要求较高。基于本文提出的僵尸网络模型,考虑使用第3种思路进行通信的设计,结合加密手段以达到一定的隐蔽效果,规避检测方法。在后续研究中,可以考虑后两种思路的结合,从而达到一种相对更好的平衡以规避检测。

2.1.2 面向多平台的兼容性与鲁棒性设计

物联网技术发展迅速,以路由器、摄像头等小型化操作系统为代表的设备接入网中,由于性能与需求的不同,这些设备采用的 CPU 架构、操作系统环境皆有不同,且系统功能一般相对简易,对僵尸程序的兼容性提出了较高要求。本文设计的僵尸网络以接入网中的嵌入式设备为主要目标,同时兼顾其他终端主机(如服务器、个人计算机等)。本文考虑并分析了各类目标的共通特点,针对嵌入式设备不同架构的问题,运用交叉编译技术将僵尸程序编译移植至每一种环境,使僵尸网络具有面向多平台的兼容性。

除了兼容性问题,僵尸程序需要稳定运行以

保证与攻击者的通信,从而能够执行各种指定操作。因此,僵尸程序的鲁棒性需要从设计上进行规划考虑,以保证其能够稳定适用于各类环境。针对 HTTP 模型的特性,本文设计的僵尸程序只提供报到拉取指令、一般指令执行与下载的功能,尽可能地将功能最小化,以达到其对运行环境依赖最小化的目的。通俗而言,本文设计的僵尸程序主要以“下载器”的形式存在,主进程自身仅集成一般的简单功能,复杂功能通过下载相应模块文件运行而达成,实现类似于“加载插件”的效果。

2.1.3 功能可扩展性与可用性设计

在多种攻击方式相配合、多种漏洞利用的环境下,可扩展性是僵尸网络扩展自身规模的一个重要特性。可扩展的僵尸网络模型要求僵尸程序满足功能灵活多变和模块化的特点,并能按需将功能模块加载使用。

针对中心式网络模型信息掌握集中、下发指令时延短的特点,该框架通过设计控制管理 UI (user interface) 界面来控守整个网络,把握每一台僵尸主机的基本信息,从而充分发挥中心式网络的优势,有利于提升攻击效率和僵尸网络的可用性。

2.2 可扩展僵尸网络框架总体设计

可扩展僵尸网络框架由上至下划分为3个结构: UI 接口、功能服务器、僵尸主机端。UI 接口以 Web 前端作为控制界面,能够查看到僵尸网络的基本情况,根据需要对僵尸进行指令下达,易于操作使用;功能服务器分为 C&C 服务器(同时为数据服务器)、文件服务器及终端服务器,各个服务器具有相对独立的功能,根据实际场景需要可以合并不同功能;僵尸主机端作为僵尸网络的末端,是各类指令的受体,为各种功能具体化实现的场所。总体而言,攻击者为僵尸网络中的“首脑”,对于整体的僵尸网络具有绝对的话语权;C&C 服务器、文件服务器与终端服务器相当于辅佐的“差役”,将攻击者下达的各项指令传达到位,对受控主机进行管理与功能支持;而受控主机则是名正言顺的“僵尸”,对传达的指令无条件地执行,实现攻击者想要的各类操作。

3 面向多平台架构的僵尸网络安全控制

3.1 面向多平台架构的僵尸程序制作机制

多平台迁移的问题涉及目标终端指令架构、目标环境支持等问题。除此之外，还需要考虑代码编写以及原理实现上的影响，如大小端存储方式不同所带来的行为差异等。

图 2 展示了面向多平台架构的僵尸程序制作机制。

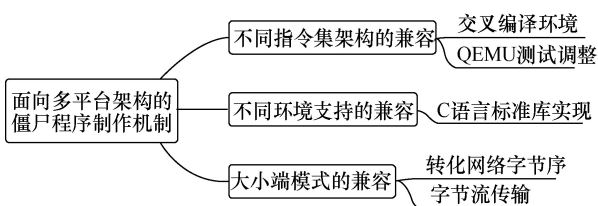


图 2 面向多平台架构的僵尸程序制作机制
Figure 2 Zombie programming mechanism for a multi-platform architecture

3.1.1 不同指令集架构的兼容

当前网络环境中，存在各类指令集架构的终端设备，如 x86_64、MIPS、ARM 系列等，还存在嵌入式设备（如 PowerPC 架构等）。如何让僵尸程序兼容多种类型的指令集架构，是其广泛植入运转亟须解决的问题。为了良好地兼容各类指令集架构，本文采用纯 C 语言进行僵尸程序的具体开发，C 语言作为较底层的程序设计语言，可尽可能地适配各类架构环境。在纯 C 语言程序设计的基础上，利用交叉编译技术，将源代码编译为相应指令集架构下的僵尸程序。

在僵尸程序调试与测试方面，本文利用虚拟化技术（如 QEMU 框架）搭建相应指令集架构的多种版本虚拟环境，开展针对僵尸程序的调试与测试，最终获得相应架构下较为通用和稳定的僵尸程序。

3.1.2 不同环境支持的兼容

各类设备由于功能不同，其支持的编程环境有所差异。例如，作为承担流量转发任务的路由设备，由于其受承担的功能、内存空间、磁盘容量等综合因素限制，不支持基于 OpenSSL 库的密码学加解密运算。另外，如果僵尸程序对运行环境有所依赖，即使通过交叉编译，也无法正常运行相关程序；如果僵尸程序以静态编译的方式解决，将导致程序过大，进而导致其无法上传或产

生的攻击流量过大。上述两种情况均不利于僵尸网络控制及其规模扩展。因此，具有高度兼容性的僵尸程序应当以最基本的库函数为支持且独立运行，以最大限度地降低对程序运行环境的依赖。

本文采用源码级代码集成的实现方式将 AES 加密功能和基于 HTTP 的网络通信结合起来，使得僵尸程序的依赖压缩到仅使用 C 语言标准库，从而实现对不同运行环境的兼容。

3.1.3 大小端模式的兼容

除了指令集与环境支持的问题，本文还考虑小端模式与大端模式的兼容问题。二者的主要区别在于数据存储在内存或磁盘中的顺序有所不同，这给僵尸程序的行为隐性带来影响。例如，假设 C&C 服务器使用小端模式，僵尸主机使用大端模式，字长为 4 byte，传输的指令为 int 类型的指令代码。如果没有考虑大小端模式的兼容，则可能出现服务器发送 0x1 而僵尸主机接收结果为 0x1000000 的情况，致使程序工作异常。面对大小端模式兼容的问题，本文在网络传输的层面上提供了一种解决方案：首先将源设备字节顺序转换为网络字节顺序，再将网络字节顺序转换为目标设备字节顺序，具体实现在 C 语言所提供的 htonl、ntohl 函数，从而有效地解决 int 类型数据的传输问题。考虑到加密流量通信，同时兼顾兼容大小端模式设备的问题，本文主要采用字节类型的数据传输方式，其中，字节类型传输相当于以字节流的方式传输数据至目标设备，目标设备读取数据后按照自身使用的大小端模式进行数据存储，以此解决通信双方所使用的大小端模式不一致造成的行为异常问题。

3.2 僵尸主机安全控制方法

僵尸网络的隐匿性及其安全性决定了其存活力大小，进而影响其网络规模的大小。然而，僵尸网络的隐匿性及安全性取决于其通信的过程，如果通信过程中的行为关键特征易于检测，则僵尸网络无法逃避当前网络安全检测手段。事实上，僵尸网络检测通常基于其通信模型，其防御的关键在于打击通信模型中固有的弱点，如中心式通信模型具有的单点失效弱点。因此，本文认为通过在设计上尽可能地减少通信模型弱点，有效地规避当前现有的常见检测手段，最大限度地保证

僵尸网络的可存活性。

3.2.1 通信维度上动态密钥 AES 加密

传统的僵尸网络一般以简单的数据加密手段（如异或操作）来保护通信数据，即使使用了加密密钥，也存在密钥重用或者密钥在同一消息中出现的现象，如 BlackEnergy2^[20]重用相同密钥加密通信。因此，这类僵尸网络信道加密机制主要通过混淆通信数据来逃避基于特征的安全检测方法，具有一定的隐匿性，然而无法抵抗基于人工

的通信流量分析，从而使僵尸网络的生存性受到影响，甚至有被接管的风险。为了保证僵尸网络自身的安全性，在增强通信隐匿性的基础上提升其抗流量分析能力，本文提出使用基于密文分组链接（CBC）工作模式的动态密钥 AES 加密来保护僵尸网络通信流量，其工作框架如图 3 所示。

以某一僵尸主机与 C&C 服务器的通信为例，结合图 3，可以得到如图 4 所示的动态密钥加密通信流程。

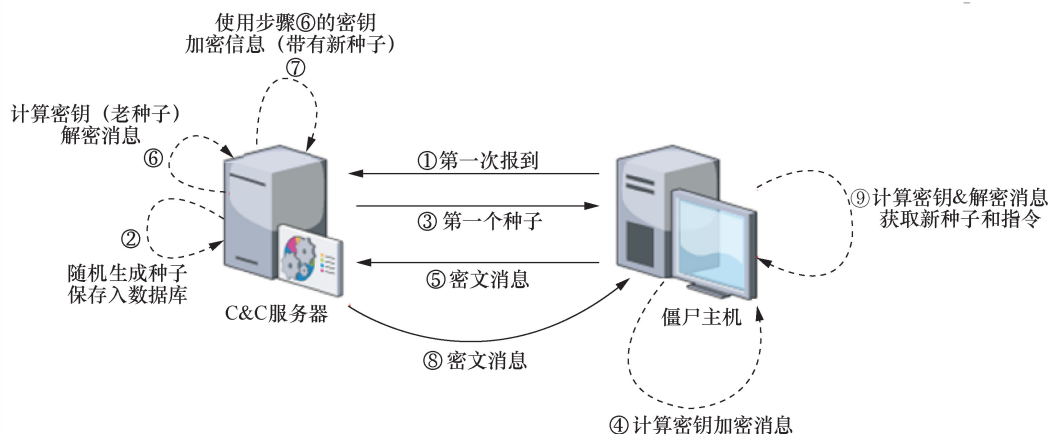


图 3 动态密钥加密通信工作框架

Figure 3 Framework of encrypted communication with dynamic secret keys

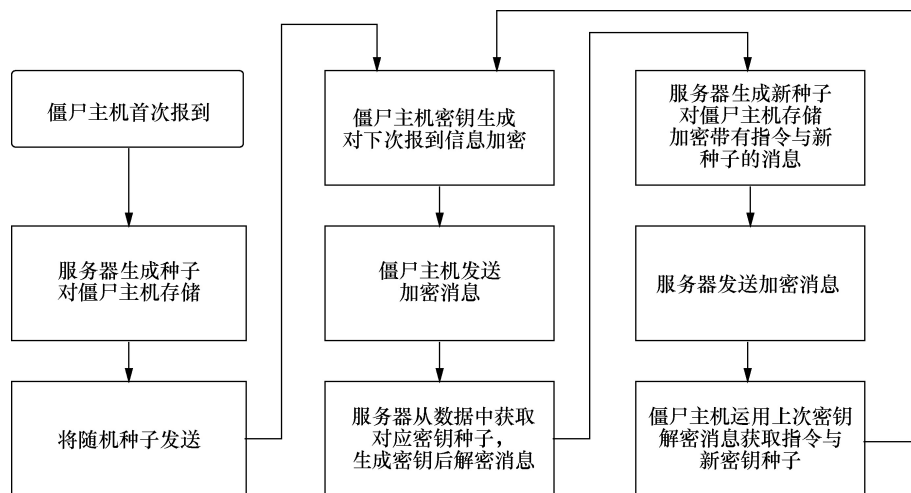


图 4 动态密钥加密通信流程

Figure 4 Process of encrypted communication with dynamic secret keys

- ① 僵尸主机第一次向 C&C 服务器通信报到。
- ② 生成随机的种子，在数据库中对对应报到的僵尸主机进行存储。
- ③ 将生成的随机种子发送给僵尸主机。
- ④ 僵尸主机运行密钥生成算法，并用此密钥结合 AES 加密算法，对下一次报到的信息进行加密。
- ⑤ 僵尸主机将加密的报到消息发送给 C&C

服务器。

⑥ C&C 服务器收到消息后,从数据库中找到对应僵尸主机的密钥种子,使用与僵尸主机相同的密钥生成算法生成密钥,并对消息进行解密。

⑦ 进行消息的处理后,重复步骤②的操作,运用步骤⑥中的密钥加密将要发送的消息,其中,发送的消息包含指令及下一次通信所用的密钥生成种子。

⑧ C&C 服务器将消息发送给僵尸主机。

⑨ 僵尸主机使用与步骤④相同的种子进行密钥计算,同时解密出相应消息,获取下一次新的种子与命令,重复步骤④至步骤⑨的过程。

通过上述方法,加密密钥、C&C 服务器指令信息和僵尸主机报到信息等受到了加密保护,并且引入动态密钥后使得人工分析解密的难度加大,从而使通过劫持消息来接管僵尸网络变得几乎不可能实现。值得一提的是,本文在密钥生成算法设计中采用了哈希加盐算法,通过在密钥的随机种子后加入盐值,从而随机变化密钥生成的结果,以保证每一次密钥的唯一性,从而使暴力破解手段变得不可行。综上所述,动态密钥 AES 加密可以有力地提升僵尸网络控制的安全性,提高通信流量的隐匿性和抗分析能力。

3.2.2 设备维度上服务器迁移与重连机制

通常,中心式僵尸网络中的僵尸主机被检测到针对某一个 IP 地址的异常访问而被发现,进而导致 IP 封禁从而掉线。为了解决这个问题,本文设计了一种 C&C 服务器快速迁移方法来提高僵尸网络的灵活性和隐匿性,进一步设计了僵尸主机重新连接机制以确保僵尸主机的在线率,提高僵尸网络的可存活性。服务器迁移与重连机制状态机如图 5 所示。

迁移机制与重连机制存在一定的相同之处,但其应用场景有所不同。迁移机制主要用于某 C&C 服务器控守较长时间后,为规避检测进行的操作,属于由 C&C 服务器发起的一种主动的调整行为;然而,重连机制则是 C&C 服务器在遭遇到网络性能波动或者僵尸主机的 IP 地址被封禁导致报到错误的场景下进行的操作,属于由僵尸主机发起的一种被动的保障存活性行为。

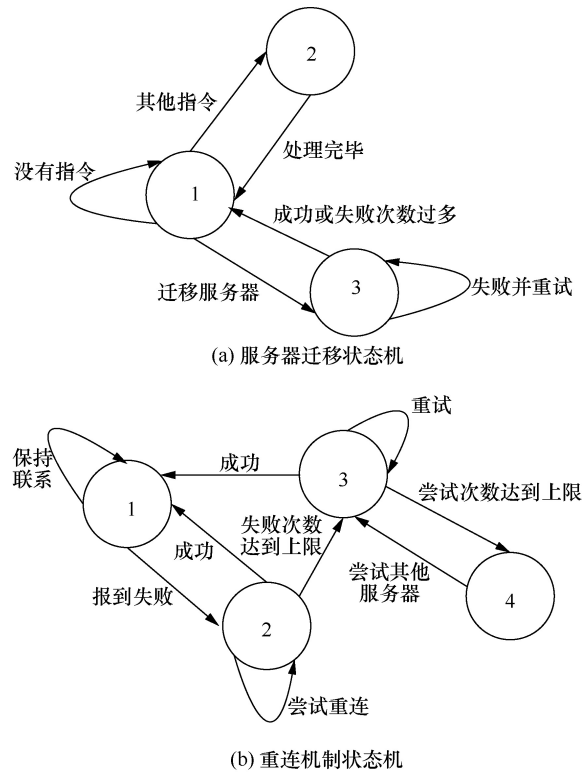


图5 服务器迁移与重连机制状态机
Figure 5 State machines of server migration and reconnection mechanism

关于迁移机制,正常通信时保持在状态 1 或状态 1、2 循环。C&C 服务器在接收到迁移指令后,在下发指令消息的同时将新的 C&C 服务器 IP 地址告知下辖的僵尸主机。如果服务器迁移成功,则会将原来的 C&C 服务器 IP 地址记录在历史连接记录中,以备未来可能进行的重新连接操作使用。迁移失败的次数限制被预先定义在僵尸程序中。关于重连机制,正常状态在状态 1 循环,当出现连接失败时,则转移至状态 2。状态 2 为初步重连状态,僵尸主机执行对当前 C&C 服务器的重连操作,成功则回归状态 1,若失败多次则认为该 C&C 服务器无法连接,从而转移到状态 3。状态 3 是僵尸主机尝试连接新的 C&C 服务器的状态,该状态与状态 2 的重连稍有不同,是按照僵尸主机第一次报到的流程开始通信,具体来讲,状态 2 是从第④步开始,而状态 3 则从第①步开始。如果连接新服务器成功,则转移至状态 1,失败多次则转移至状态 4。状态 3 和状态 4 的循环代表着在尚未连接成功时,对历史连接记录进行循环遍历直至成功连接。同样,重连失败的次数限制被预先定义在僵尸程序中。除此之外,

在编写僵尸程序源码的时候可以将多个 C&C 服务器的 IP 地址预先写入程序中,有助于提升重连成功率。

综上所述,通过迁移机制和重连机制,僵尸网络可以进一步增强其通信隐匿性与鲁棒性,在发挥中心式通信模型优势的同时尽可能地减少其单点失效带来的不利影响。

4 仿真测试及结果分析

僵尸网络通信流量处理是 C&C 服务器的主要负载来源,本文提出的通信加密给每一次通信带来了额外的工作负载,因此,服务器的负载能力直接影响着其控制的僵尸网络规模。鉴于此,开展针对 C&C 服务器服务能力的压力测试能够检验服务器对整个僵尸网络的群控能力,对本文方法进行模型抽象和性能测试是验证其正确性与实用性的必要方式。

4.1 测试模型与环境

由于本文研究的僵尸网络采用了基于 HTTP 的通信方式,因此,可以将每一个僵尸主机看作“用户”,而僵尸主机报到的行为相当于“用户”对“网页界面”的一次访问,将 C&C 服务器服务能力的压力测试抽象成为一个 Web 服务器提供用户访问服务的负载能力测试。针对中心式的网络通信模型,C&C 服务器提供“发送指令”服务可以类比成为一个 Web 服务器在收到来自不同地域地区的用户访问请求后返回相应的处理结果。僵尸网络测试抽象模型示意如图 6 所示。

将本文提出的僵尸主机控制过程抽象为 Web 网页服务访问通信模型具有良好的拟合性与合理性。具体而言,用户访问 Web 网页服务的过程具有一定的随机性,与僵尸主机随机间隔时间报到的行为相似;用户通过 HTTP 访问 Web 网页服务,服务器反馈的结果为网页内容或服务数据,与僵尸网络中 C&C 服务器反馈僵尸主机行动指令的过程十分类似。值得一提的是,针对僵尸主机控制过程的合理抽象有助于降低方法性能测试的难度,同时有助于开展大规模僵尸网络的性能测试工作。由于大规模僵尸网络涉及的僵尸主机数量众多,搭建实网环境和基于虚拟机的虚拟网络环

境均存在成本过高、测试周期长、环境搭建难等突出问题。因此,本文通过纯软件方式模拟 C&C 服务器与僵尸主机之间的通信过程,便于进一步开展方法性能测试,并反馈与实际情况相近的结果。此外,通过软件仿真,本文可以实现大规模僵尸主机报到的通信过程模拟,模拟出的僵尸网络规模能够达到成百上千甚至上万台僵尸主机的规模。

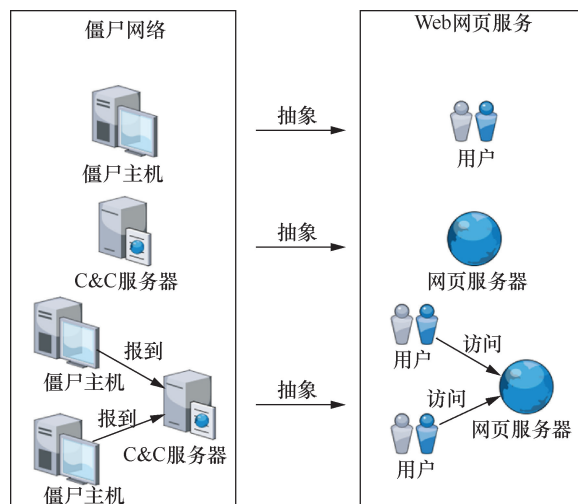


图6 僵尸网络测试抽象模型示意

Figure 6 The abstract model of botnet testing diagram

本文的仿真测试基于常用的开源软件 JMeter 实现,该软件支持自定义构造 HTTP 报文和多线程发送数据包。JMeter 使用 BeanShell 脚本对 POST 数据进行定制填充,可以充分地模拟出僵尸网络中的通信流量。仿真环境中的 Web 服务器端配置基本上与真实的 C&C 服务器配置一致,此外,仿真网络的拓扑结构相对简易,由物理机 (IP 地址: 192.168.8.100) 与虚拟机 (IP 地址: 192.168.8.177) 共同构成,其中物理机运行 JMeter 进行僵尸网络仿真规模的控制及流量发送,虚拟机负责 C&C 服务器的功能。

具体的仿真测试参数如表 1 所示。

4.2 仿真测试设计

由于本文提出的僵尸主机控制方法的主要负载点是加解密过程和数据库的交互过程,仿真测试在具体编程实现上可以进行一定的简化处理。随机的密钥生成种子来自 C&C 服务器, JMeter 是僵尸主机通信流量的主要发起者。JMeter 主要模拟僵尸主机端功能,但由于其功能限制,难以

表 1 仿真测试参数
Table 1 Simulation testing parameters

测试机	IP 地址	操作系统	磁盘空间	CPU	内存大小/GB
物理机	192.168.8.100	Windows 10	—	Intel Core i7-8700 CPU @ 3.20 GHz	64
虚拟机	192.168.8.177	Ubuntu 16.04	50 GB		2

完全模拟出获得相应的密钥种子以保护每一次通信的过程。此外, JMeter 软件的主要任务为模拟出合法的僵尸网络加密通信流量, 且仿真测试的目标是该通信方法针对 C&C 服务器服务能力的测试, 能够在测试时保持服务器解密与操作数据库的负载与实际使用时一致则达到测试效果。鉴于此, 综合考虑实现难度与实验合理性, 本文假设使用固定的密钥种子进行密钥生成并进行方法性能测试。

僵尸网络的僵尸主机大部分处于报到状态, 测试中仅对报到流量进行模拟测试即可覆盖绝大多数的网络通信情况, 同时对于负载情况具有一定的代表性。JMeter 通过多线程技术发送流量包模拟众多的用户访问, 其功能支持设置发包间隔以确定网络吞吐量。本次测试的自变量为僵尸网络的僵尸规模以及其报到频率, 在实际测试中的表现形式即吞吐量, 而吞吐量的控制由 JMeter 所设定的虚拟用户数及访问间隔实现。假设虚拟用户数量为 V , 测试时间为 T , 每名用户的访问间隔时间为 I , 每个用户每次通信发送的 HTTP 有效报文大小为 B , 则吞吐量 F 的计算如式(1)所示。

$$F = \frac{V \cdot T \cdot B}{I} \quad (1)$$

从式(1)可知, 如果将测试时间 T 、访问时间间隔 I 与每次通信的报文大小 B 固定, 则吞吐量 F 与虚拟用户数量 V 呈线性关系。为保持测试情况的一致, 测试时间 T 在测试时是相同的, 而测试中仅模拟常见的报到情况 (即 B 相对固定), 因此测试中需重点考虑访问时间间隔 I 与虚拟用

户数量 V 。通过改变时间间隔 I , 能够划分僵尸网络的具体使用场景, 而网络规模的增长是所有场景下僵尸网络所需要进行的行为。因此, 时间间隔 I 将在场景划分中被确定, 由此可以将自变量限定为虚拟用户数量 V 。在具体测试操作中, 时间间隔 I 是一个随机产生值, 其范围限定在一个时间区间内, 且取得区间内各值的概率相同。因此, 以区间平均值确定 I 的取值是合理可行的。综上所述, 测试设计上通过确定测试时间 T , 依据时间间隔 I 划分场景, 将虚拟用户 V 作为自变量是合理可行的, 能够代表在某一使用场景下的僵尸网络规模扩展的仿真模拟。

为了度量 C&C 服务器服务能力的负载, 最直观的量化指标有 CPU 占有率、内存占有率、响应时间、磁盘输入/输出 (I/O) 速度等。根据本文提出的加密通信原理, C&C 服务器需要进行数据加解密运算和数据库访问, 因此, 本文认为 CPU 占有率、内存占有率、磁盘 I/O 和响应时间等指标能够相对全面且合理地量化指示服务器的负载。

为模拟不同场景下的使用情况, 本文按照控制强度不同, 将仿真测试场景划分为高控制性、中控制性和低控制性 3 种场景, 其中, 控制程度的高低取决于僵尸主机的报到间隔时间, 如表 2 所示。

为了评价在某个测试场景中僵尸网络规模和 C&C 服务器负载之间的关系, 仿真实验中的僵尸网络规模被设定为从 100 个僵尸主机开始, 到 1 000 个僵尸主机时结束。其中, 多个波次的僵尸主机数量增长模拟了一个僵尸网络扩展的过程,

表 2 不同测试场景的参数设定
Table 2 Parameter settings of different testing scenarios

测试场景	报到间隔区间/s	僵尸主机规模/个	僵尸主机规模增长速度/(个·波次 ⁻¹)	每波次测试时间/s
低控制性	180~360	100~1 000	100	30
中控制性	60~120	100~1 000	100	30
高控制性	10~30	100~1 000	100	30

每个波次的仿真时间为 30 s，每一波僵尸主机增长在 5 s 内完成。图 7 展现了仿真测试中僵尸网络规模增长情况。

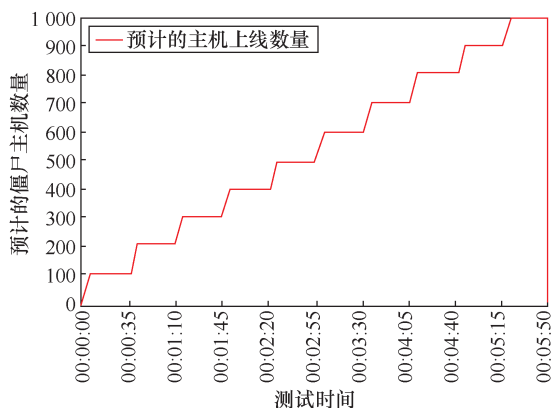


图7 仿真测试中僵尸网络规模增长情况
Figure 7 The increase of botnet size in simulation testing

4.3 测试结果与分析

本节将实现上述的测试设计，并对结果进行展示，其中，CPU 占有率计算为 MySQL 软件和 Apache 服务器 CPU 占有率的平均值；内存占有率的计算为服务器总体的内存占用率，从开启所有 C&C 服务器的必需服务后开始起算，测试开始后每波次进行记录，取每波次的结果平均值；磁盘 I/O 通过 sysstat 包的 iostat 命令进行读取，可以获取实时磁盘的读写情况，最后取该测试区间内结果的平均值。根据式(1)，结合僵尸网络规模对每波次测试的理论吞吐量 F 进行计算。经 Wireshark 流量抓包分析，单个用户每一次报到的报文大小 B 为 365 byte。测试时间 T 为 30 s，报到间隔 I 由不同控制性场景决定。

(1) 低控制性场景

在低控制性水平场景下，对应的结果如图 8~图 10 所示，其中，计算理论吞吐量 F 的报到间隔 I 取报到时间的理论平均值 270 s。

(2) 中控制性场景

在中控制性水平场景下，对应的结果如图 11~图 13 所示，其中，计算理论吞吐量 F 的报到间隔 I 取报到时间的理论平均值 90 s。

(3) 高控制性场景

在高控制性水平场景下，对应的结果如图 14~图 16 所示，其中，计算理论吞吐量 F 的报到间隔 I 取报到时间的理论平均值 20 s。

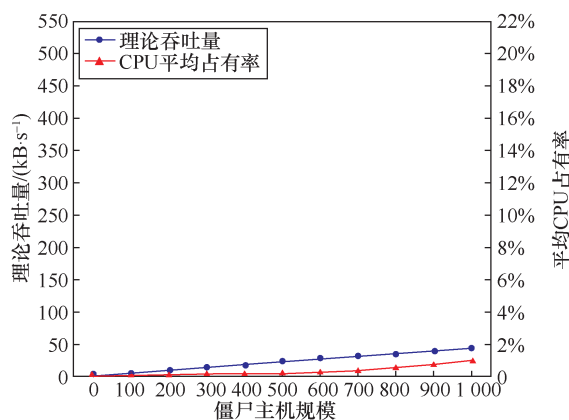


图8 低控制性水平场景下理论吞吐量与平均 CPU 占有率变化情况
Figure 8 Theoretical throughput and average CPU utilization versus at a low level of botnet controllability

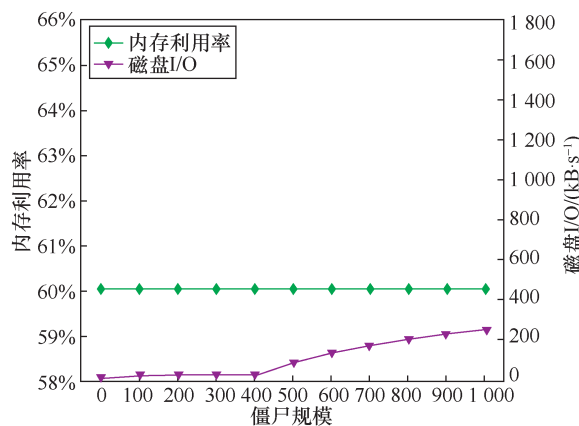


图9 低控制性水平场景下内存利用率与磁盘 I/O 变化情况
Figure 9 Memory utilization and disk I/O versus at a low level of botnet controllability

平均 CPU 占有率的主要来源为 mysqld 进程，即来自数据库的操作；模拟请求取得的响应结果均正确，为反馈的加密通信流量；理论吞吐量与现实吞吐量基本一致。以下对实验结果进行具体分析。

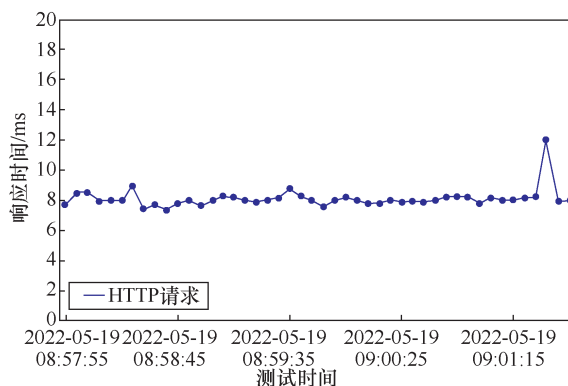


图10 低控制性水平场景下 C&C 服务器响应时间变化情况
Figure 10 The C&C server's response time at a low level of controllability

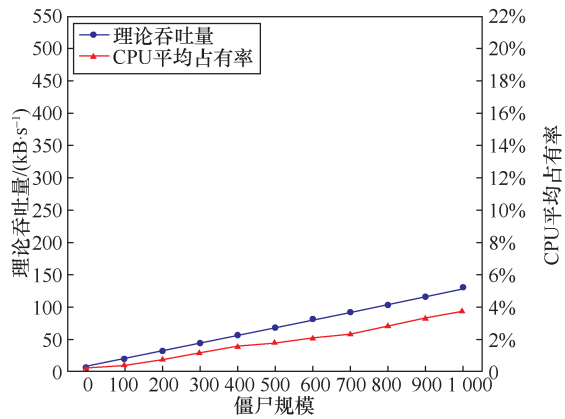


图 11 中控制性水平场景下理论吞吐量与平均 CPU 占有率变化情况
Figure 11 Theoretical throughput and average CPU utilization versus the botnet size at a middle level of controllability

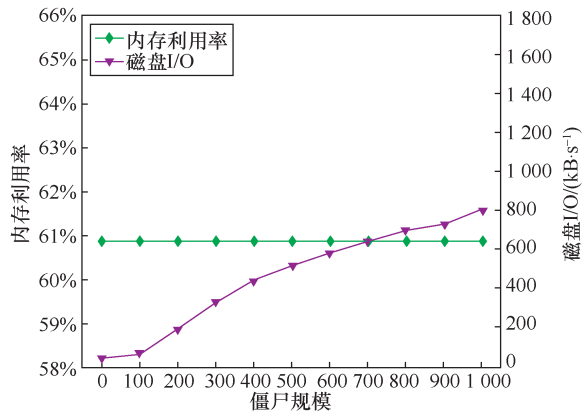


图 12 中控制性水平场景下内存利用率与磁盘 I/O 变化情况
Figure 12 Memory utilization and disk I/O versus the botnet size at a middle level of controllability

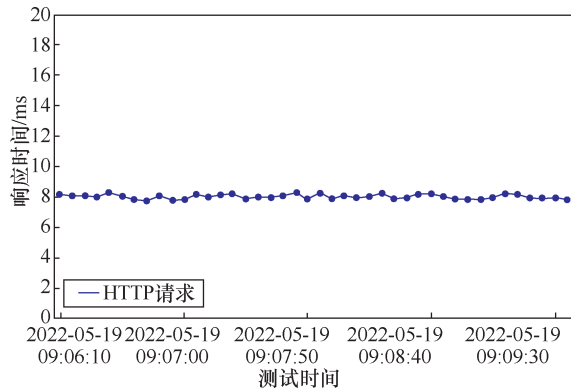


图 13 中控制性水平场景下响应时间变化情况
Figure 13 The C&C server's response time at a middle level of controllability

纵向对比低、中、高 3 种场景下的各项结果，不难发现三者共同呈现的特点。

1) 内存占有率普遍平稳，增长相对平缓。特别是控制性较低的场景下，内存占有率的变化能够近似整体操作系统的内存占用浮动。由此能够

推断，该控制方法下，内存占有率的变化受僵尸规模的增长影响不明显，基本的服务器内存配置足以支撑其运行。

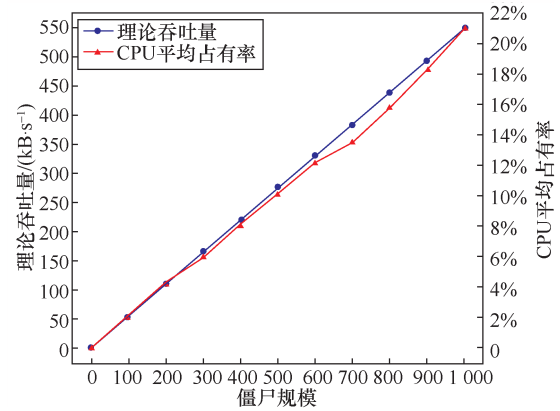


图 14 高控制性水平场景下理论吞吐量与平均 CPU 占有率变化情况
Figure 14 Theoretical throughput and average CPU utilization versus the botnet size at a high level of botnet controllability

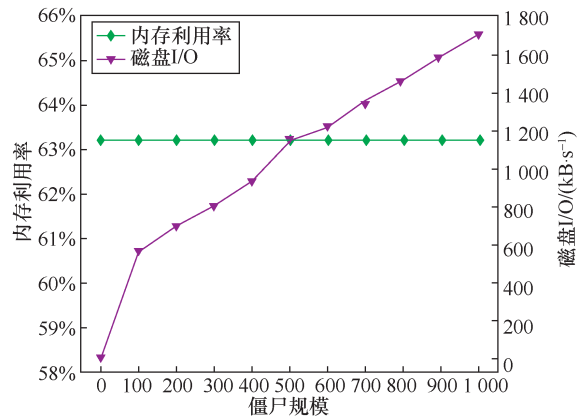


图 15 高控制性水平场景下内存利用率与磁盘 I/O 变化情况
Figure 15 Memory utilization and disk I/O versus the botnet size at a high level of botnet controllability

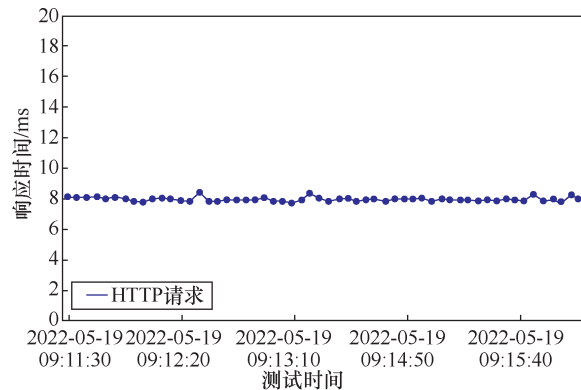


图 16 高控制性水平场景下响应时间变化情况
Figure 16 The C&C server's response time at a high level of controllability

2) 响应时间普遍在 8 ms 上下浮动，基本不受规模扩大与报到频率的影响。在保证响应时间

平稳的同时, 响应内容能够保证正确无误, 有力地证明了本文方法的实用性与可行性。

3) 平均 CPU 占有率与磁盘 I/O 均随僵尸规模增长而增大。增长的速度与其所在的控制性场景下密切相关, 这与测试的设计预期相吻合。当控制性较高时, 僵尸报到的频率越高, C&C 服务器需要处理的报文越多, 负载将会越大, 反之亦然。鉴于僵尸网络实际使用中的报到频率设置, 为保持隐匿, 其基本上将报到间隔设置为远高于仿真测试中所定下的间隔时间。参考低控制性下的负载指标, CPU 占有率基本保持在 1% 以下, 磁盘 I/O 远低于磁盘 MB 级的吞吐量。由此能够推断, 在实际使用中, 使用本文方法的 HTTP 僵尸网络通信模型具有较高的效率, 且对于配置要求较为宽松。

4) 本文定义的吞吐量指标与报到时间间隔呈明显的负相关关系。仿真结果表明, 在相同规模时, 越高的控制性会带来越高的吞吐量, 相应的负载指标越高, 能够从网络流量的角度解释说明服务器的负载程度。

横向对比每一场景下的结果, 其重点主要在于 CPU 占有率及磁盘 I/O 的变化上。不难发现, 平均 CPU 占有率与磁盘 I/O 与僵尸规模呈正相关性。由纵向对比可以推断, 该线性关系的增长斜率与该场景下的控制性有关, 即与报到时间间隔相关。在越低的控制性下线性关系呈现得越不明显, 在规模较大时负载指标出现一种快速增长的趋势。出现此现象可能与僵尸通信机制相关。在控制性较低条件下, 报到时间相对较长, 在测试区间内, 在该时间前上线的僵尸可能在不报到的情况, 从而造成前期负载指标较为平缓的现象。在僵尸网络规模增长后, 其在每一段时间内的报到次数随之平均, 从而呈现较为合理的相关关系。此类现象在低控制性下较为明显, 而高控制性报到频率较快, 造成在每一确定时间段中报到次数相对平均, 从而能够使得负载指标与僵尸规模呈现较好的线性关系。观察不同吞吐量下负载指标的变化, 结合 3 种情景, 可以看出经式(1)计算的吞吐量指标能够直观反映负载与网络规模的关系, 吞吐量与负载基本呈正相关关系。对于翻倍式的吞吐数值增长, 相应负载指标并未呈现相似的增长速

度, 而是相对平缓地增长, 说明了本文方法的可扩展性。总而言之, 测试结果说明本文方法具有较好的性能, 能够在实际中加以应用。

综上所述, 仿真测试的结果验证了僵尸主机安全控制方法的可行性与实用性, 展示出僵尸网络规模与 C&C 服务器服务能力负载之间存在线性关系, 有力地支撑了本文所提出僵尸网络框架的应用实践。

5 结束语

为了解决僵尸网络自身存在的兼容性、隐匿性与安全性问题, 本文提出了一种基于 HTTP 的可扩展僵尸网络框架, 进一步提出了一种面向多平台架构的僵尸网络安全控制方法。针对多架构的设备, 该方法联合应用源码级代码集成与交叉编译技术, 进一步提升僵尸程序的兼容性; 针对传统僵尸网络流量具有规律性、易被分析的特点, 该方法提出基于动态密钥 AES 加密的保密通信方法; 针对中心式僵尸网络模型存在单点失效的弱点, 该方法设计了服务器迁移与重连机制, 从功能设计上减少安全弱点的影响, 提高僵尸网络存活率。仿真实验结果验证了本文方法的有效性和现实可行性。

虽然本文提出的动态密钥 AES 加密方法能够实现足够的安全性, 但鉴于基于 HTTP 的数据本身是明文传输的客观现实, 流量被加密在行为上存在不符合常理的问题。为此, 后续工作可以考虑使用自带数据加密特性的 HTTPS 进行数据通信, 从而使通信流量合理地以加密形式出现。另一个思路是利用数字隐写技术, 将待加密的通信数据隐藏在 HTTP 报文传输的常规数据(如图片)中, 以实现行为更加合理且隐匿效果优良的僵尸网络通信。

参考文献:

- [1] HAN B, YANG X R, SUN Z G, et al. OverWatch: A cross-plane DDoS attack defense framework with collaborative intelligence in SDN[J]. Security and Communication Networks, 2018.
- [2] 腾讯云 T-Sec DDoS 防护团队, 绿盟科技威胁情报团队. 2021 年全球 DDoS 威胁报告[R]. 2022.
Tencent Cloud T-SEC DDoS Defense Team, NSFOCUS Threat Intelligence Team. 2021 annual report of world-wide DDoS threats[R]. 2022.

- [3] European Union Agency for Cybersecurity (ENISA). ENISA threat landscape 2021[EB]. 2022.
- [4] 王钰莹. SAM Seamless Network 报告显示 2021 年物联网攻击超过 10 亿次[J]. 互联网天地, 2022(5): 60.
WANG Y Y. SAM Seamless Network report shows more than 1 billion IoT attacks in 2021[J]. China Internet, 2022(5): 60.
- [5] HE J, YANG Y X, WANG X L, et al. Adaptive traffic sampling for P2P botnet detection[J]. International Journal of Network Management, 2017, 27(5): e1992.
- [6] 吴迪, 方滨兴, 崔翔, 等. BotCatcher: 基于深度学习的僵尸网络检测系统[J]. 通信学报, 2018, 39(8): 18-28.
WU D, FANG B X, CUI X, et al. BotCatcher: botnet detection system based on deep learning[J]. Journal on Communications, 2018, 39(8): 18-28.
- [7] 沈琦, 涂哲, 李坤, 等. 基于集成学习的僵尸网络在线检测方法[J]. 计算机应用研究, 2022, 39(6): 1845-1851.
SHEN Q, TU Z, LI K, et al. Online botnet detection method based on ensemble learning[J]. Application Research of Computers, 2022, 39(6): 1845-1851.
- [8] 肖喜生, 龙春, 杜冠瑶, 等. 基于流量摘要的僵尸网络检测[J]. 计算机系统应用, 2021, 30(8): 186-193.
XIAO X S, LONG C, DU G Y, et al. Botnet detection based on flow summary[J]. Computer Systems & Applications, 2021, 30(8): 186-193.
- [9] 罗扶华, 张爱新. 基于深度学习的僵尸网络检测技术研究[J]. 通信技术, 2020, 53(1): 174-179.
LUO F H, ZHANG A X. Botnet detection technology based on deep learning[J]. Communications Technology, 2020, 53(1): 174-179.
- [10] WANG X F, ZHANG D F, SU X. Design of mobile botnets Integrated SMS and HTTP protocol C&C channel[J]. Journal of Chinese Computer Systems, 2014, 35(7): 1458-1463.
- [11] KNYSZ M, HU X, ZENG Y, et al. Open Wi-Fi networks: Lethal weapons for botnets[C]//Proceedings of the 2012 IEEE INFOCOM. 2012: 2631-2635.
- [12] WANG P, SPARKS S, ZOU C C. An advanced hybrid peer-to-peer botnet[J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(2): 113-127.
- [13] HAN Q, YU W, ZHANG Y, et al. Modeling and evaluating of typical advanced peer-to-peer botnet[J]. Performance Evaluation, 2014, 72: 1-15.
- [14] 高见, 芦天亮, 王威. 基于 P2P 的僵尸网络 C&C 设计与仿真[J]. 中国人民公安大学学报(自然科学版), 2015, 21(3): 65-70.
GAO J, LU T L, WANG W. C&C design and simulation of P2P-based botnets[J]. Journal of Chinese People's Public Security University (Science and Technology), 2015, 21(3): 65-70.
- [15] 柳淑婷, 傅梓怡, 范亚芹. 基于 Android 的僵尸网络设计与实现[J]. 吉林大学学报(信息科学版), 2016, 34(2): 182-185.
LIU S T, FU Z Y, FAN Y Q. Design and implementation of mobile botnet based on android system[J]. Journal of Jilin University (Information Science Edition), 2016, 34(2): 182-185.
- [16] 张翕然. 物联网僵尸网络病毒传播模型研究及仿真[D]. 重庆: 重庆理工大学, 2021.
ZHANG X R. Research and Simulation of internet of things botnet virus propagation model[D]. Chongqing: Chongqing University of Technology, 2021.
- [17] BAUMGARTNER K, GARNAEVA M. BE2 custom plugins, router abuse, and target profiles[EB]. 2014.
- [18] 周安民, 钟毅, 左政, 等. D-BitBot: 比特币网络双向通信的 P2P 僵尸网络模型[J]. 哈尔滨工业大学学报, 2020, 52(5): 66-74.
ZHOU A M, ZHONG Y, ZUO Z, et al. D-BitBot: a P2P duplex botnet model in Bitcoin network[J]. Journal of Harbin Institute of Technology, 2020, 52(5): 66-74.
- [19] 赵昊, 舒辉, 康维, 等. 基于智能合约的高对抗性僵尸网络研究[J]. 网络与信息安全学报, 2021, 7(4): 30-41.
ZHAO H, SHU H, KANG F, et al. High resistance botnet based on smart contract[J]. Chinese Journal of Network and Information Security, 2021, 7(4): 30-41.
- [20] WANG W, MA X. Blockchain-based botnets for command-and-control resilience[J]. Botnets: Architectures, Countermeasures, and Challenges, 2019: 217.

[作者简介]



刘强 (1986-), 男, 江西临川人, 博士, 国防科技大学副教授, 主要研究方向为网络安全、机器学习和人工智能安全。



李鹏飞 (2000-), 男, 安徽无为, 国防科技大学硕士生, 主要研究方向为二进制漏洞挖掘和恶意程序分析。



付章杰 (1983-), 男, 河南南阳人, 博士, 南京信息工程大学教授、博士生导师, 主要研究方向为网络与信息安全。