电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS

论文题目 基于相似性分析的僵尸网络检测研究与实现

学科专业 计算机应用技术

学 号 201121060450

作者姓名 席 瑞

指导教师 侯孟书 教授

分类号 _____ 密级 _____

UDC [注1] _____

学　位　论　文

基于相似性分析的僵尸网络检测研究与实现

（题名和副题名）

席瑞

（作者姓名）

指导教师　侯孟书　教 授

电子科技大学　成 都

（姓名、职称、单位名称）

申请学位级别　硕士　学科专业　计算机应用技术

提交论文日期　2014.03　论文答辩日期　2014.05

学位授予单位和日期　电子科技大学　2014 年 6 月 29 日

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

RESEARCH AND IMPLEMENTATION OF BOTNET DETECTION BASED ON SIMILARITY ANALYSIS

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Major:	**Computer Applied Technology**
Author:	**Xi Rui**
Advisor:	**Professor Hou Mengshu**
School:	**School of Computer Science & Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期： 年 月 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期： 年 月 日

摘　要

僵尸网络因其自身易控制、范围广、难检测等特点已发展成为目前互联网中最主要、最广泛的网络攻击平台，Botmaster 可以利用僵尸网络实施窃取信息、Spamming、DDOS 攻击等恶意行为。随着僵尸网络技术不断发展，僵尸种类和规模的不断增加，针对僵尸网络的检测技术也已成为网络安全领域的一个研究热点。

目前已有的僵尸网络检测技术主要有网络流量内容检测、异常检测、日志检测等，这些检测技术都会存在一些缺陷或限制，例如对不同类型僵尸网络检测的通用性不足、检测系统在实际环境中部署困难等。

为解决检测方法通用性不足的问题，本文通过分析僵尸网络的工作原理和通信机制发现僵尸主机表现出的网络流量特征、主机行为具有一定的群体相似性特点。以相似性为理论依据，本文结合已有的僵尸网络检测技术提出了基于相似性分析的僵尸网络检测方法，设计并实现了一种检测系统模型。该检测系统模型包括网络抓包、网络流量分析、主机行为分析和交叉关联聚类四个主要功能模块，通过抓取、分析网络中的 TCP 和 UDP 数据包来检测网络中是否存在僵尸网络。其中，网络抓包模块部署在被监测网络边界处监测和收集网络数据包，网络流量分析模块功能通过对网络数据包分析提取相似特征并找出具有相似特征的主机，主机行为分析模块功能则根据网络流量特征找出可疑的主机行为以及表现出相同行为的主机，交叉关联聚类模块针对网络流量分析模块和主机行为分析模块得到的结果进行过滤、相似性度量计算找出可疑的、属于同一僵尸网络的主机。

本文在现有实验资源条件下搭建了一个简单的局域网环境，并分别在不同的实验情景中对检测系统进行实验验证，通过对实验结果分析可以发现此检测方法能够有效的检测出局域网内的僵尸主机。在本文最后提出了一些预防僵尸网络感染的措施，并对本文的工作进行了总结、提出下一步的工作和目标。

关键字：僵尸网络，检测系统，相似性，聚类

ABSTRACT

With the characteristics of easily control, a wide range of influence and difficult to detect, Botnet has become the most important and widely employed cyber attacks platform in the Internet. Botmaster can be used to launch DDOS Attacks, Spamming, steal information and other malicious activities. The Evolving of technology and the increasing of the botnet make the detection become a hot research area in network security.

At present, the content of network traffic detection, anomaly detection, logging detection are the main detection techniques. But there are still some faults and limitations, for example, the method could not be used to detect more than one kinds of botnets, and also it is hard to be deployed in a real environment.

To resolve the problem of versatility, by analyzing network traffic and communication mechanism, we find the network traffic and host behavior of botnet master has a certain degree of group similarity characteristics. Based on similarity theory, this paper proposed a botnet detection method based on similarity analysis combined with the existing botnet detection technology, then designed and implemented a detection system model. The detection system model, which consists of network packet capture, network traffic analysis, host behavior analysis and cross-correlation clustering those four main function modules. The system detects whether there is a botnet or not in a certain network by analyzing the captured TCP and UDP data packets. Among them, the network packet capture module is deployed at the border of the network to monitor and collect network data packets. Network traffic analysis module is to find the hosts that have similar characteristic by analyzing the network packets. Based on the characteristics of network traffic, host behavior analysis module identifies suspicious behaviors and hosts that exhibit the same behaviors. With the result data of network traffic analysis module and host behavior analysis module, the cross-correlation clustering module finds out the suspicious hosts belonging to the same botnet by the way of filtering and calculating similarity measure.

In order to verify the detection system, we build a simple local area network environment with the limited experimental resources and propose some different experimental scenarios. According to the experimental results, it is couldc be found that

this detection method could detect botnet hosts in the LAN. Finally, in the last of the paper, we puts forward to a number of botnet infection prevention measures, summarize this work and propose some future work and goals.

Key word: botnet, detection system, similarity, clustering

目 录

第一章 绪 论

1.1 研究背景与意义

Interent 在世界范围内各个地区的广泛普及、不同行业的广泛使用以及技术的不断进步，由一些恶意网络活动（如常见的 Trojan、Malware、Spyware、DDOS 攻击等）引起的网络安全问题不断暴露出来，引起各国政府、安全机构和网络用户的广泛关注。虽然世界各国颁布了相关法律法规及相应的防范措施以应对 Internet 上存在的安全问题，但是网络攻击者们也在不断地利用具有自我保护、自我隐藏能力且能够高效执行网络攻击行为的网络技术、方法来规避安全系统的检测，使得网络安全问题无法得到有效控制或者彻底解决，例如 Botnet。

Botnet（亦称为僵尸网络）是由互联网中被攻击者感染且能被攻击者远程控制的主机组成的一个网络，其网络规模大小从几十台到成千上万台主机各不相同。通常，网络攻击者将僵尸网络作为一个攻击平台实施各种网络恶意行为（例如 DDOS 攻击、Spamming、信息窃取等）以达到其个人某种目的，僵尸网络的出现使得对网络恶意行为的防御和检测工作变得更加困难，给企业或者个人的信息安全带来很严重的威胁及经济损失。例如 2013 年 12 月 06 日，微软宣布瓦解了全球最大的犯罪网络—ZeroAccess[1]，该犯罪网络影响了全球约 200 万台电脑。犯罪团伙利用被感染主机组成网络，在用户不知情的情况下让电脑点击广告，在搜索引擎上（如微软的 Bing、谷歌的 Google 等）欺骗广告客户，让他们为不可能带来销售额的活动支付费用。微软表示，ZeroAccess 每个月让 Bing、Google 和 Yahoo 等搜索引擎损失约 270 万美元。

2012 年中国互联网网络安全报告对于境内、境外的僵尸网络控制服务器和受控主机进行了详细分析[2]。报告指出，在利用木马或者僵尸网络控制服务器对主机进行控制的事件中，控制服务器 IP 总数为 360263 个，较 2011 年增加 19.9%，受控主机 IP 总数为 52724097 个，较 2011 年大幅增加 93.3%；2012 年，境内木马或僵尸程序控制服务器 IP 数量为 286977 个，境外木马或僵尸程序控制服务器 IP 数量为 73286 个，较 2011 年分别有了 13.1%和 56.9%的上升幅度，具体如图 1-1 所示；

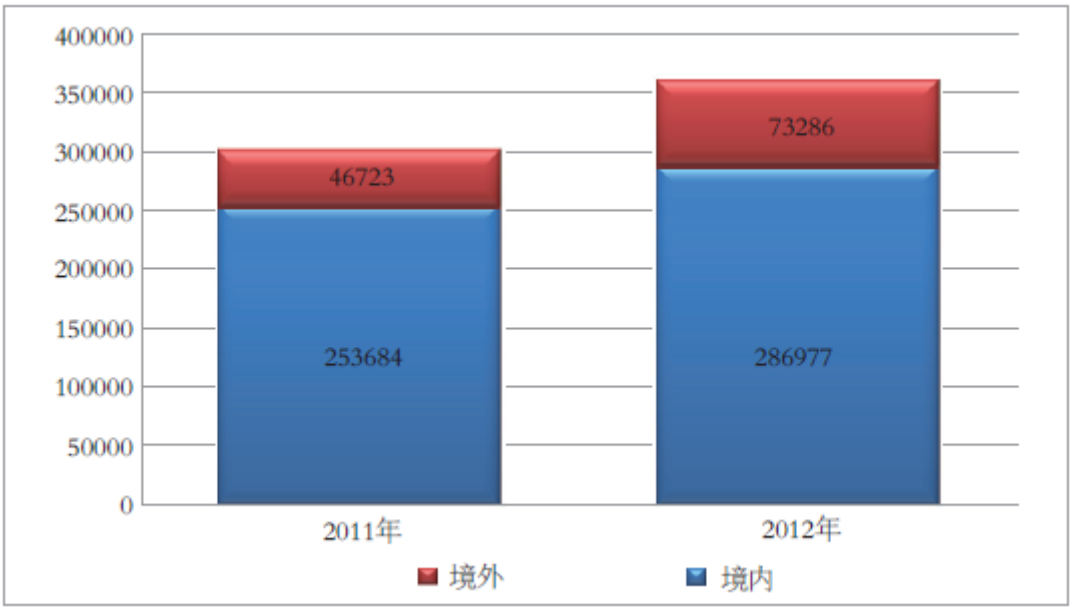

图 1-1 2012 年与 2011 年木马或僵尸程序控制服务器数据对比（来源：CNCERT/CC[2]）

2012 年，在发现的因感染木马或僵尸程序而形成的僵尸网络中，规模 100-1000 的占 79.2%。控制规模在 1000-5000、5000-20000、2 万-5 万的主机 IP 地址僵尸网络数量与 2011 年相比 cc 分别减少 338、46、11 个，控制规模在 5 万-10 万的僵尸网络数量与 2011 年持平，10 万以上则大幅增加 62 个，分布情况如图 1-2 所示。

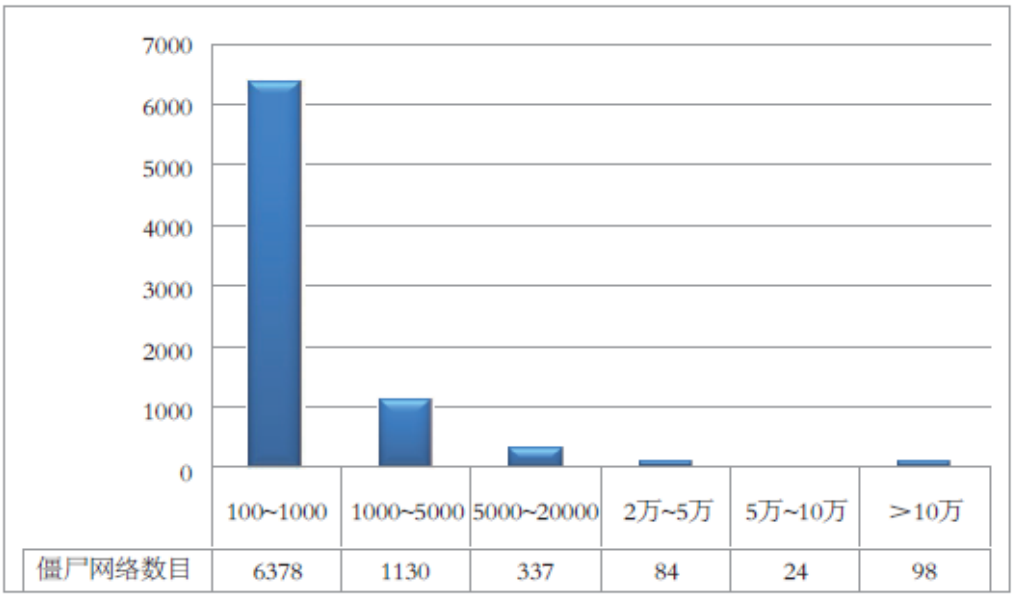

图 1-2 2012 年僵尸网络规模分布（来源：CNCERT/CC[2]）

境外木马或者僵尸程序控制服务器 IP 数量前 10 位按国家和地区分布如图 1-3

所示，其中美国位居第一，占境外控制服务器的 17.6%，日本和中国台湾分列第二和第三，占比分别为 9.6%和 7.6%。

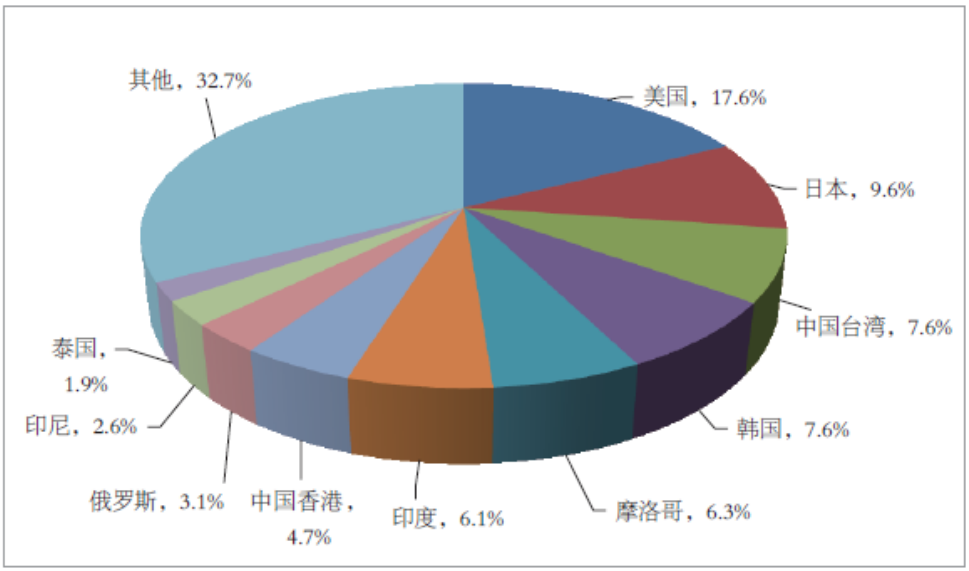

图 1-3 2012 年境外僵尸网络程序控制服务器 IP 地址按国家和地区分布（来源：CNCERT/CC[2]）

2012 年，境内共有 14646225 个 IP 地址的主机被植入木马或僵尸程序，境外共有 38077872 个 IP 地址的主机被植入木马或僵尸程序，数量较 2011 年均有了大幅增长，增幅分别达到了 64.7%和 107.2%，如图 1-4 所示。

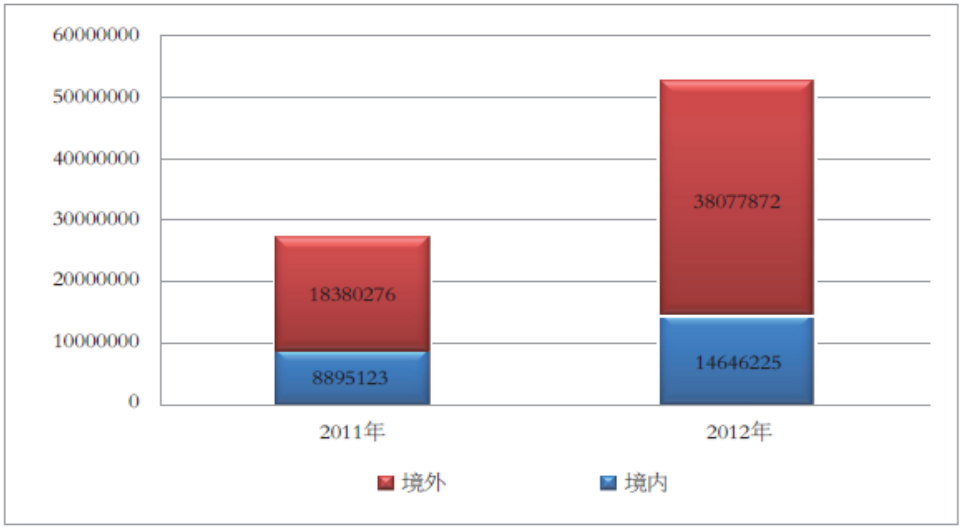

图 1-4 2012 年和 2011 年僵尸受控主机数量对比（来源：CNCERT/CC[2]）

另外，根据华为云安全中心统计[3]，2013 年中国和美国的僵尸网络主机数分别占全球僵尸网络主机整体数量的 30.3%和 28.2%，我国超越美国成为全球受僵尸

网络影响最严重的国家和地区。美国作为全球僵尸网络控制者最多的国家，占全球总量的 42.2%，德国、法国位列第二、三位，分别占到了 9.1%、7%，而中国作为僵尸网络受灾最严重的国家，其境内控制者只占到总量约 3.8%。

从上述统计报告中可以发现我国作为世界上僵尸网络威胁最严重的国家，其僵尸网络控制者的占有量确实很少的，无论在政治上、经济上等方面都潜伏着很严重的威胁，迫切需要对僵尸网络检测技术进行研究，提出一种高效、精准且通用的检测方法保证互联网络安全、为网络用户提供一个干净、安全的网络环境，对僵尸网络检测技术的研究具有很重要的意义。

1.2 研究现状

在僵尸网络发展初期，由于感染途径有限、网络威胁影响小并未得到安全研究人员的广泛关注。随着僵尸网络技术的不断进步和成熟、影响范围的逐渐扩大以及危害程度的急剧增强，网络安全研究机构或公司开始把 Botnet 作为网络安全领域的热点问题展开深入研究。

2004 年僵尸网络程序 Agobot/Gaobot、rBot/Spybot 以及变种程序在 Internt 上广泛传播，各安全公司开始成立专门研究部门、投入大量资源从攻击性和恶意性对僵尸网络进行研究、提取特征信息、更新病毒库文件。一些传统安全公司（例如 Symantec、Kaspersky 等）不仅对 Botnet 进行了长期、实时的跟踪和监测，而且将 Botnet 活动情况作为一个独立部分在会议上报告并写入安全报告。

最开始学术界对 Botnet 的研究工作主要是蜜网组织研究人员采用蜜网技术捕获、追踪和分析不同种类僵尸网络来了解其工作原理、主机行为、活动特征等信息，其中比较著名的蜜网组织有 Azusa Pacific 大学的 Bill McCartv、法国的 Richard Clarke、华盛顿大学的 Dave Dittrich 以及德国的蜜网项目组等。例如，在 2004 年末到 2005 年初的三个月时间里，来自德国的一个蜜网项目组研究人员利用蜜罐捕获机制追踪、监测了 100 多个活跃的僵尸网络，通过对追踪结果分析做出首份与僵尸网络追踪有关的技术分析研究报告[4,5]。目前，在国际上举办了许多关于僵尸网络方面研究的会议，其中影响力较大的有 USENIX 协会举办的 SRUTI（workshop on Steps to Reducing Unwanted Traffic in the Internet）、LEET(workshop on Large-scale Exploits and Emergent Threats)，ACM、IEEE 等举办的 ACSAC(Annual Computer Security Applications Conference)、RAID(international symposium on Recent Advances in Intrusion Detection)等等，通过这些会议的举办不仅在僵尸网络检测领域取得了显著地研究成果，而且也在不断增加其在僵尸网络相关研究领域的影响力。

目前，研究人员通过对不同类型僵尸网络基本特征的学习提出了许多针对不同类型 Botnet 的检测方法并取得了一定效果，其中基于主机或者基于网络的检测方法是当前网络安全研究人员研究僵尸网络检测技术的重点。

基于主机的检测技术通过分析某一台主机行为中是否存在可疑的僵尸网络行为来判断该主机是否存在僵尸程序，该检测技术的原理是僵尸程序在执行的过程中调用系统库文件的序列不同于系统合法进程调用系统库文件的序列[6]，通过识别系统中异常的调用系统库文件序列判断是否存在僵尸程序。例如，Masud 等人提出的僵尸网络检测方法[7]根据僵尸程序比人类响应迅速的特征，通过挖掘主机日志文件、关联日志文件来捕获该特征，MABDS（Model of Multi-Agent Bots Detection System）检测系统是利用主机入侵检测系统和操作系统事件日志分析器检测僵尸程序是否存在[8]，文献[9]提出了一种利用操作系统规则检测僵尸程序感染行为的、基于主机的安全工具 DeWare。

基于网络的检测技术是目前使用最多、最广泛的检测技术，同时也是僵尸网络检测领域的一个研究热点。BotProbe 利用僵尸程序采用某些固定命令应答模式进行会话通信的特征[10] 动态的向内部主机注入网络包，分析应答数据包信息来判断会话的另一端是否为僵尸程序，Rishi 提出的基于签名的 IRC 僵尸网络检测系统把已知IRC僵尸网络对僵尸程序的命名规范作为标签对IRC僵尸程序进行追踪[11]，Strayer 等人提出的基于网络的检测方法结合机器学习技术对网络流量进行分析检测僵尸网络[12]，Choi 等人提出了一种基于 DNS 查询相似性特征的僵尸网络检测方法，该方法利用 DNS 查询操作具有源 IP 地址固定性、瞬时高峰性和动态性的特点对 DNS 查询进行相似性匹配[13]，Ping Wang 等人针对 P2P 僵尸网络根据其通信行为相似性和周期性特点提出了一种检测 P2P 僵尸网络方法，该方法首先利用增量式分类技术从网络流量中分离出 P2P 流量，再利用 Dynamic clustering technique、Boolean auto-correlation technique 识别出可疑的僵尸主机[14]。

在 2005 年左右我国国内才开始对僵尸网络及其检测技术展开研究工作[15]，其中北京大学和国家计算机网络应急技术处理协调中心是国内最早从事僵尸网络相关工作研究的两家研究单位。北京大学计算机科学技术研究所的研究人员利用沙箱、蜜罐两种研究手段对病毒样本进行分析并采集僵尸程序在加入网络前 C&C 信道的属性信息，研究人员通过追踪已捕获的、从事非法网络活动的 Botnet，分析收集到的数据掌握该 Botnet 在全球的分布情况、规模数量等信息。随着僵尸网络威胁日益严重、活动日趋频繁，僵尸网络检测技术研究的重要也与日俱增，同时在许多高校中针对僵尸网络检测技术展开了相关研究工作，例如中山大学的王涛等人根据僵尸网络特征相似性提出一种检测方法[16]；华中科技大学的王彬彬等人分

别根据异常地址对应关系、网络通信流的簇分布相似率和网络连接的行为模式三个方面提出了检测网络中 Bot 主机的方法[17]；北京邮电大学的苏云琳等人通过检测异常的网络流量的存在来判断 Botnet 是否存在[18]；电子科技大学的严庆等人将数据挖掘技术与网络入侵检测技术相结合，相互协同工作检测僵尸网络[19]；哈尔滨工业大学的李超等人提出一种在 IRC 僵尸网络初期阶段分析 Bot 程序与 C&C 服务器的会话过程的特征，并结合多种检测技术进行检测的 IRC 僵尸网络检测方法[20]等等。

目前已有的僵尸网络检测技术均是在全面学习僵尸网络基本知识、仔细分析对不同类型僵尸网络的特征、深入研究不同种类僵尸网络的内在特征之后提出的检测方法。通过上述研究现状可以发现，僵尸网络检测技术研究在国内外已经引起社会各个领域广泛关注，无论安全公司、网络安全科研单位或者高校都在不断增加僵尸网络领域的研究力度。但是在互联网技术不断成熟、更新日新月异的同时，Botnet 技术也在不断提高和完善，给僵尸网络检测和防御工作带来极大地困难和挑战。

1.3 研究目标及内容

僵尸网络作为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计算机网络平台已经成为当今互联网上最主要的网络攻击平台，被攻击者们广泛用于进行 DDOS 攻击、Spamming、Click Fraud、信息窃取等恶意网络行为。僵尸网络技术的不断发展和创新势必会对僵尸网络检测技术提出很大的挑战，关于僵尸网络检测技术的研究也必然会成为研究热点。

当前僵尸网络检测技术主要有基于网络流量内容的检测技术、基于网络行为的检测技术、基于时间关联的检测技术和基于日志的检测技术等，但上述检测技术都会存在一些缺陷或限制，例如检测方法通用性不强，只能针对一种或者一类僵尸网络进行检测；检测方法在实际应用的难以部署等。

通过对僵尸网络基本原理、实现机制等知识的学习和分析发现 Botnet 主机在网络流量和主机行为上表现出群体相似性的特征，利用此特征提出一种基于相似性分析的检测技术，设计并实现检测模型。主要研究内容如下：

(1) 深入学习和分析僵尸网络基本原理和实现机制；

(2) 学习现有僵尸网络检测技术的基本原理；

(3) 设计并实现一种僵尸网络检测模型；

(4) 提出有效预防僵尸网络的措施。

1.4 论文组织结构

本文共分为六个章节。

第一章作为本文绪论部分，分别从研究背景及意义、国内外研究现状和论文组织结构三个方面做出了扼要阐述。

第二章对僵尸网络知识进行了总结性阐明。本章节可以分为僵尸网络概述、僵尸网络技术、僵尸网络检测技术和发展趋势四个部分对僵尸网络知识进行详细介绍和分析。首先简单介绍了僵尸网络的基本知识，包括僵尸网络定义、生命周期、感染途径和恶意活动。然后重点对僵尸网络技术和僵尸网络检测技术进行了消息说明。最后针对僵尸网络未来的发展趋势做出了简单介绍。

第三章在第二章的理论基础之上，提出了一种基于相似性分析的僵尸网络检测模型，并分别从检测模型的目标、检测模型的提出和检测模型设计三个方面对模型进行说明。

第四章对第三章提出的监测系统模型各个模块的具体实现过程和方法进行详细阐述。

第五章对监测模型的实验验证过程进行说明。首先介绍实验环境的部署情况，然后对于实验测试方案进行详细介绍说明，最后对于实验结果进行分析总结并给出一些预防僵尸网络感染的措施。

第六章对系统僵尸检测模型的局限性进行了说明，并展望了未来工作。

1.5 本章小结

本章作为论文的绪论部分对论文的研究背景与意义、研究现状和研究内容及目标进行了论述说明，同时简单介绍论文的组织结构。目前僵尸网络在 Internet 上活动日趋频繁、威胁日益严重，互联网安全现状让人堪忧，针对僵尸网络检测方法的研究迫在眉睫。

第二章 僵尸网络综述

本章我们主要从僵尸网络基本概念、僵尸网络技术和僵尸网络检测技术三个部分对僵尸网络展开介绍，同时针对目前僵尸网络最新技术及僵尸网络检测方法研究趋势做简要介绍。拟为提出一种新的僵尸网络检测方法提供相关理论和技术基础。

2.1 僵尸网络的概述

在本小节我们从僵尸网络的定义、生命周期、感染途径和恶意活动四个方面对僵尸网络的基本知识进行了详细的介绍，有利于对僵尸网络有一个比较全面、基础的人生和了解。

2.1.1 僵尸网络的定义

根据文献[21]、[22]中对僵尸网络的介绍，恶意僵尸网络是一群被攻占、运行着恶意程序的主机(一般称为“Bot 主机”) 在一个网络攻击者(一般称为“Botmaster”) 的远程控制下所组成的一个网络。其中，恶意程序一般称为 Bot 程序，其在网络主机合法用户未察觉的情况下自动运行事先定义好的功能，且通过该程序可以远程控制网络中主机执行一些指定的操作。文献[22]、[23]和[24]对僵尸网络程序与传统恶意软件之间的区别都作出了说明，指出僵尸网络与蠕虫、木马等传统恶意软件相比不仅具备传播、远程控制功能同时还具有独特的命令与控制信道机制。

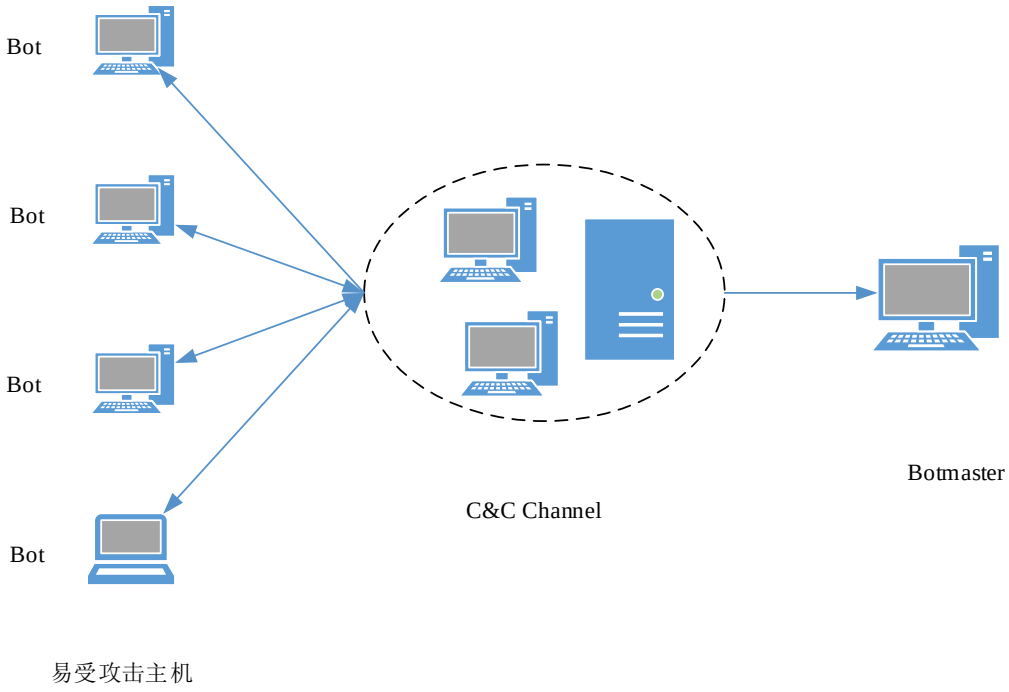

图 2-1 僵尸网络组成示意图

如图 2-1 所示，僵尸网络的组成主要包括 Botmaster、Bot 和命令控制信道(Command and Control Channel，简称 C&C Channel) 三个基本组成部分。由图 2-2 僵尸网络进行网络恶意攻击示意图可发现僵尸网络具有如下四个属性，

(1) 攻击性。攻击性指 Botmaster 采用非法手段扫描、入侵和感染网络中正常主机，控制僵尸网络中主机执行网络恶意行为，例如 DDOS 攻击、Spamming、Click Fraud 等。

(2) 非合作性。非合作性指 Botmaster 在网络主机合法用户完全不知情的状况下注入 Bot 程序到计算机中运行。

(3) 可控制性。可控制性指 Botmaster 能够利用 C&C Channel 远程控制僵尸网络中的 Bot 主机执行 Botmaster 发送的特定攻击行为命令，是僵尸网络与其他网络恶意行为的本质区别。

(4) 协作性。协作性指僵尸网络中 Bot 主机联合协同工作完成 Botmaster 发送的攻击命令，协作性使得基于僵尸网络发动的网络攻击行为影响范围更广、攻击力度更强、防御难度更高。

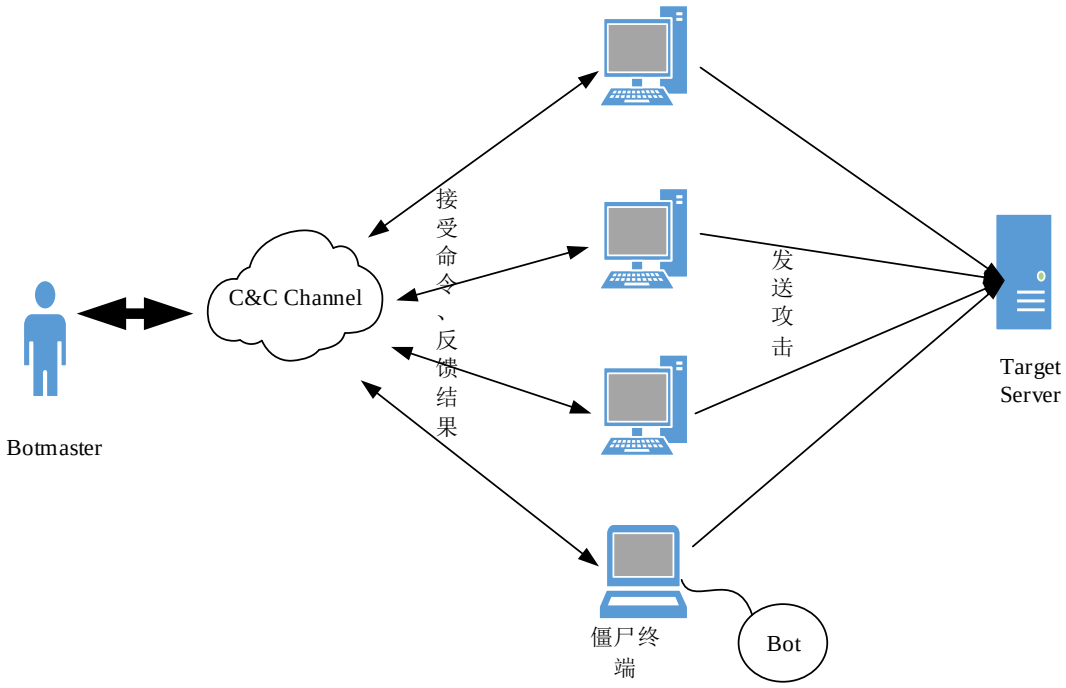

图 2-2 僵尸网络发送攻击示意图

2.1.2 僵尸网络的生命周期

僵尸网络整个生命周期可分为 4 个阶段：感染、连接、恶意命令与控制、更新与维护，如图 2-3 所示，

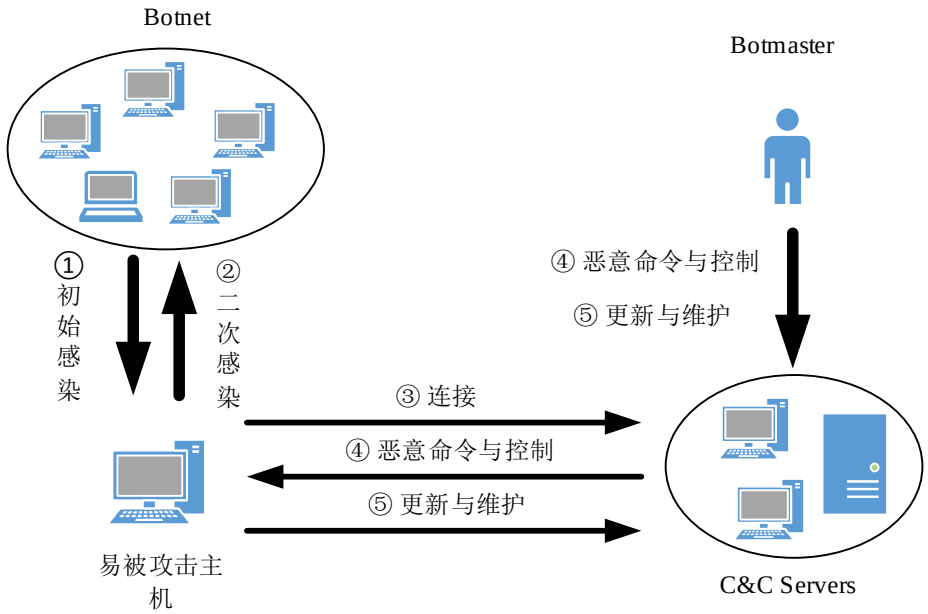

图 2-3 僵尸网络的生命周期示意图

感染阶段分成初始感染和二次感染两个阶段。在初始感染阶段攻击者利用不

同方法和技术扫描目标局域网，找出容易被攻击的主机并注入恶意脚本程序。在二次感染阶段被注入主机执行恶意脚本程序利用 FTP、HTTP、DNS 等协议从某一特定网络位置获得可执行的 Bot 程序，Bot 程序在目标主机安装成功之后目标主机就成为僵尸网络的一个僵尸终端，运行着 Bot 程序。Bot 程序具有自启动功能，每次僵尸终端启动时都会自动运行。

在连接阶段，Bot 程序创建一个 C&C 信道并向 C&C Server 发起连接加入僵尸网络。在恶意控制与指令阶段，已成功加入网络作为一个僵尸终端的 Bot 主机接收 Botmaster 通过 C&C 信道向僵尸网络终端分发的命令并执行，C&C 信道的存在使得 Botmaster 能够远程控制大量 Bot 主机进行不同的非法活动。

在维护与更新阶段，Bot 程序执行 Botmaster 发送的更新命令从某一特定网络位置下载并更新可执行 Bot 程序。一般情况下，Botmaster 会因为避免僵尸网络检测系统的检测、给 Bot 程序添加新的功能等原因对 Bot 程序进行更新，例如 Botmaster 为避免僵尸网络检测系统检测、保证僵尸网络隐蔽轻便、延长僵尸网络的活跃时间对 C&C 服务器进行迁移操作，此时就需要更新 Bot 程序中 C&C servers 地址来保证网络正常工作[25,26]。另外，Botmaster 也会采用 Dynamic DNS(DDNS)[27] 技术频繁的更新和改变 server 的 IP 地址，当僵尸网络 C&C server IP 地址被安全部门破坏之后，Botmaster 只需要将原先的 DNS 域名绑定到一个新的 IP 地址就能够建立一个新的 C&C server，新的 IP 地址在域名生存期内传播给网络中所有 Bot 程序，Bot 程序接收到该 IP 地址之后转移到新建立 C&C server 位置并存活下来[28]。

2.1.3 僵尸网络的感染途径

僵尸网络技术最初主要是被 IRC 聊天室管理员用于对 IRC 聊天室进行管理，但后来这种技术被黑客广泛用于控制网络中易于攻击的主机执行一些非法活动以达到其个人某种目的。当时由于 Bot 程序感染采用被动方式即需要网络主机用户进行下载才能够在主机上运行，其影响范围和威胁程度都比较小。随着网络蠕虫技术的不断发展和成熟，Botmaster 将 Bot 程序与蠕虫传播技术相结合使得 Bot 程序能够主动的在网络中进行传播，这种方式称为主动方式，类似的方法还有远程漏洞攻击、弱口令扫面入侵、邮件附件、恶意文档、文件共享等等。例如，僵尸程序 Agobot/Gaobot 和 rBot/Spybot 采用蠕虫传播技术在网络中传播[29]，这种方法具有感染率高、传播速度快、规模大的特点。后来为了提高 Bot 程序传播过程的隐蔽性，Botmaster 采用网页挂马(drive-by-download) 的方式进行传播[30]。

近年来，Botmaster 常常采用比主动方式或被动方式在传播过程中欺骗性更强的社会工程学(social engineering) 手段在 Internet 上传播僵尸程序，例如在对 Storm

僵尸程序的研究分析发现，将传播邮件的标题和内容替换成一些最近发生的新闻和热点事件来诱使网络用户点击邮件进行 Bot 程序传播具有很高的成功率[31]; 此外，一种称为 Koobface 的 Bot 程序经常给会盗用被感染主机用户的社交网站帐号，向该账号社交好友发送恶意 URL 诱使对方点击该连接，进而感染其计算机。目前网络上一些比较流行的社交网络(如 facebook 、Weibo、blog 等)、即时通信(Instant Message) 工具(如 QQ、MSN、 微信等) 都可能是 Botmaster 传播僵尸程序的重要途径和渠道[32]。

如今，为了增强僵尸程序在网络传播效率，Botmaster 现有的多种传播、感染方法结合起来传播僵尸程序，例如僵尸程序、木马、Spy 及搜索引擎等传播方法相结合[33]。同时为了躲避网络中僵尸网络检测，Botmaster 将加密技术和 Bot 主机相结合。

2.1.4 僵尸网络的恶意活动

最初僵尸网络的研究目的是希望可以使得网络上一些管理员对某一个网络机器的管理更加方便，例如利用僵尸程序可以使得基于 IRC 聊天室管理员对聊天室的管理工作更加便捷、简单。相反地，目前网络上存在的僵尸程序很大一部分并未被广泛用于从事合法活动，而是被一些不法分子利用在网络上从事各种恶意网络行为以满足自身利益、实现某种目的，这些恶意活动可以分为如下几种[14,34]，

(1) DDOS 攻击

DDOS 攻击是僵尸网络中最常用的一种攻击方式，DDOS 攻击是通过大量的消耗目标服务器端网络带宽、系统资源而使得该服务器无法正常提供网络服务的一种攻击策略。例如网络犯罪分子首先向僵尸网络中发布连接某一服务器的指令，网络中 Bot 程序在接收到此命令后开始向该服务器发送服务请求连接，此时服务器就会疲于处于大量的来自僵尸网络的服务请求连接而无法正常工作、陷入瘫痪状态。数据显示，目前网络中最常出现的 DDOS 攻击类型有 TCP SYN FLOOD、ICMP FLOOD 和 HTTP REQUEST FLOOD，如图 2-4 所示为 DDOS 攻击示意图。

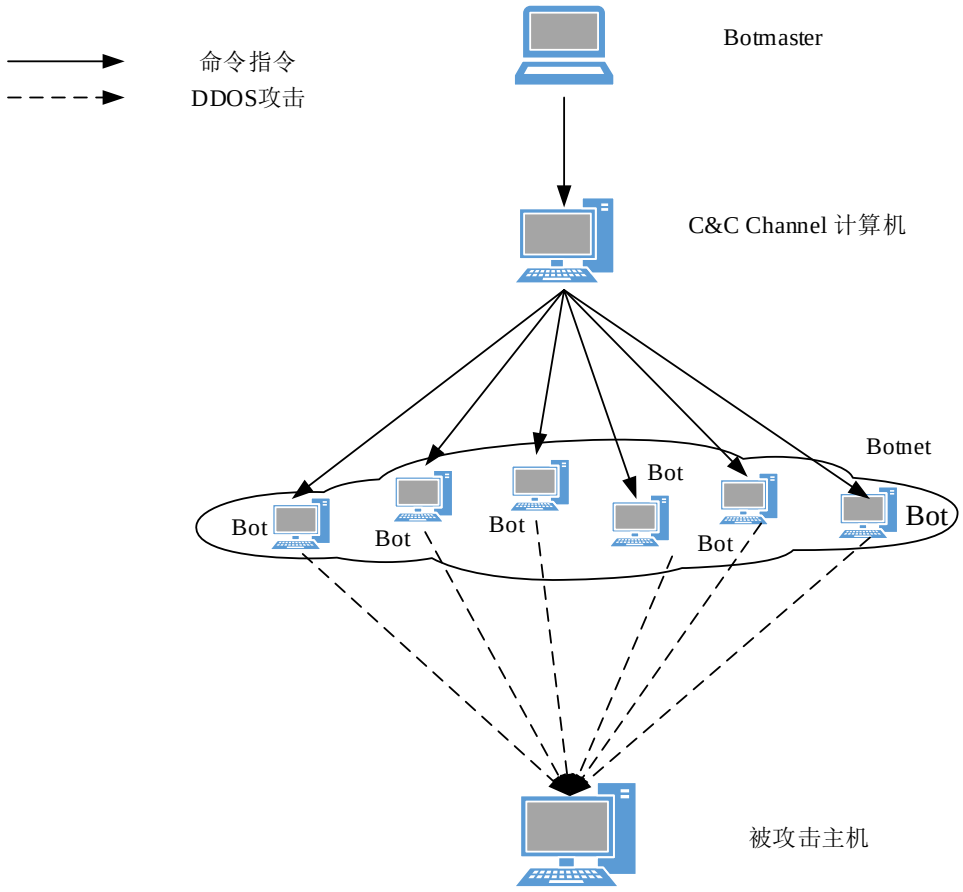

图 2-4 DDOS 攻击示意图

针对DDOS攻击的一般措施需要满足如下2个条件：a.控制者大量被攻占机器；b.关闭远程控制机制。无论如何，为避免此类攻击依然需要提出更加高效的方法，Michael Bailey 提出一种通过搜索蜜罐网中隐藏的僵尸程序来防止 DDOS 攻击。

(2) Spamming

由僵尸网络 Spamming 攻击发出的垃圾邮件数量占到了世界上垃圾邮件总量的 70%到 90%。研究报告标明，一旦被 Bot 程序感染且打开了 SOCKS v4/v5 代理的主机常会被用执行 Spamming 的恶意活动，Bot 程序利用某些特殊功能收集主机合法用户邮箱中联系人地址并向这些邮箱地址发送大量的垃圾邮件。研究人员假设 Bot 程序在执行 Spamming 攻击时会在很短的时间内向网络中发送大量的垃圾邮件，并依据此假设提出一种采用分布式架构的、独立于邮件内容的垃圾邮件分类系统 Trinity。

(3) 信息窃取

Bot 程序不仅能够探测到其寄存主机发送或接收的网络流量，同时能够对用

户输入的命令数据进行监测和返回，这种功能使得 Botmaster 可以很轻松的获取 Bot 主机合法用户的用户名、密码等敏感信息。由于 Bot 主机对正在运行的被感染机器的性能影响很小且处于安全软件监视区域之外，一般很难被检测到。窃取用户信息的 Bot 程序采用按键记录的方法监听主机键盘活动情况，过滤掉没有意义的按键信息发送给 Botmaster，按键记录方法使得网络攻击者能够在网络中窃取到大量用户隐私信息和凭证数据。证据显示，为快速搜索被感染主机中大量企业和金融数据，Bot 程序功能也变得更加复杂。

(4) 身份欺诈

身份欺诈是互联网中一种快速增长的网络犯罪行为，其中网络中广泛传播的钓鱼邮件就是一种典型的身份欺诈行为，钓鱼邮件诱使接收者点击邮件中包含的看似合法的 URL 并提交私人或者秘密信息，僵尸网络程序根据垃圾信息机制生成这些钓鱼邮件并在 Internet 上广泛传播。另外，僵尸网络也可能会通过建立许多虚假网站冒充官方商业网站来收集网络中被害用户信息，与此同时在僵尸主机中不断的弹出这些虚假网站直至主机用户关闭计算机。

(5) 网络感染

僵尸网络在构建初期其规模比较小，通过不断感染网络中主机加入僵尸网络使得网络规模更大、攻击效果更加明显。感染僵尸网络常用的方法有扫描有漏洞的主机、传播带有 Bot 程序附件的 E-mail。

由上述僵尸网络恶意活动可以发现，网络攻击者利用僵尸网络发动恶意活动或者网络攻击主要是利用了僵尸网络以下几个特点。

a) 网络规模大，例如 DDOS 攻击、Spamming；
b) 网络具有隐蔽性，隐蔽性包括通信隐蔽，Bot 程序的隐蔽性等，例如信息窃取；
c) 可作为跳板，网络攻击者在利用僵尸网络进行恶意活动或者网络攻击时，即使网络中一些节点被发现，安全人员也很难追踪到攻击者，能够很好的保护攻击者身份。

如下表 2-1 所示，我们从三个方面对上述五种恶意活动进行分析比较。

表 2-1 僵尸网络恶意活动对比

恶意活动	可检测性	设计难度	攻击效果
DDOS 攻击	高	低	高
Spamming	中	低	高
信息窃取	低	高	低

身份欺诈	中	高	中
网络感染	中	高	中

2.2 僵尸网络技术

在本节中，我们将重点对僵尸网络的 C&C 信道协议、C&C 体系架构及僵尸网络最新技术三个方面进行阐述说明。通过这些内容的学习有助于我们对僵尸网络具体实现、工作流程等核心知识有更加深入的理解，并为僵尸网络检测方法的提出提供了理论依据和技术支撑。

2.2.1 僵尸网络的 C&C 信道协议

为提高僵尸网络通信的隐蔽性，Botmaster 经常采用网络中最常用的、规范中已定义的标准协议作为 C&C 信道通信协议，用于 Bot 程序间的通信。学习网络协议有利于我们追踪僵尸网络攻击的起源和监听、破解 Bot 程序与 Botmaster 之间、Bot 程序和 Bot 程序之间的通信会话过程。目前，僵尸网络的 C&C 信道常采用的协议有以下几种[24,35]，

(1) IRC（Internet Relay Chat）协议

IRC 协议是基于 TCP/IP 协议、可用于在 Internet 上进行一对多通信聊天的一种协议，具有简单、低延时和匿名通信的特点。进行通信的 IRC 客户端都将连接到一个中心型 Server，Server 使用 nickname 区分不同的 IRC 客户端以及 IRC 客户端消息的转发。

由于 IRC 协议具有简单、低延迟、匿名和易控制的特点已成为目前僵尸网络 C&C 信道使用最广泛的一种协议，在僵尸网络中利用一个中心服务器将消息和命令转发给僵尸网络 Bot 程序或者 Botmaster。IRC 协议虽然很明显的有点，但是使用该协议通信的网络也存在着易检测、单点失效、易破坏等非常严重的缺陷，例如使用 IRC 协议进行通信时所有的信息都会通过中心服务器进行转发，中心服务器网络流量大、网络流量传输方向固定容易的特点使得其容易被 Internet 上安全系统检测、追踪。同时由于所有的信息都要通过中心服务器进行转发，一旦中心服务器陷入瘫痪或者被封堵将导致整个网络系统陷入瘫痪无法工作，因此对于使用 IRC 协议通信的僵尸网络网络只需要检测到其中心服务器并破坏就能瓦解整个僵尸网络。

(2) HTTP 协议

HTTP 协议是目前互联网中广泛存在并主要应用于 Web 服务的一种通信协议，相比较于 IRC 协议具有难以检测和网络限制少的两个优势。首先在 Internet

中存在着大量的 Web 服务和 HTTP 协议数据流量，从海量的 HTTP 协议数据中检测用于僵尸网络通信的 HTTP 协议数据是比较困难的且对检测系统性能要求比较高。其次网络管理员为保证局域网环境安全会在网络出口处安装防火墙、配置规则来限制某些网络协议（例如 IRC 协议等）数据包进入局域网，由于 HTTP 协议常用于提供 Web 服务，防火墙等安全软件一般都采取放行策略使得利用 HTTP 协议通信的僵尸网络能够很容易绕过防火墙等安全软件检测进行通信。正是基于这两个优势，越来越多的 Botmaster 开始采用 HTTP 协议作为 C&C 信道协议进行通信，但是由于 HTTP 协议采用了与 IRC 协议相同的 C/S 模型通信，需要 HTTP 服务器作为通信中心，因此也同样存在单点失效等缺陷。

(3) P2P 协议

无论僵尸网络的 C&C 信道协议是 IRC 协议还是 HTTP 协议，其通信模型始终是中心型的 C/S 模型，当单一的中心服务器瘫痪或者被除去时整个僵尸网络就会陷入瘫痪无法工作。为克服这个缺点，Botmaster 开始逐渐使用结构更加复杂的 P2P 协议作为僵尸网络 C&C 信道通信机制。

P2P 协议是一种无中心化形式通信模型，网络中每个节点都充当两种角色（即服务使用者和服务提供者），节点之间可直接通信、提供服务。由于网络中不存在中心服务器，因此网络不存在瓶颈、单点失效等问题且具有很好的鲁棒性，同时 Internet 上的机器能够通过 P2P 网络中任何一个节点加入网络，网络具有很强的扩展性。采用 P2P 协议作为 C&C 通信机制，网络中不存在单一的中心服务器、节点之间可以直接通信、网络中流量比较分散且无序流动等特点使得僵尸网络被检测到几率大大降低。由于网络中 Bot 主机是对等关系，部分节点的丢失或被抓获并不影响整个网络正常工作，因此也不存在单点失效或网络瓶颈的问题。综上可以发现，相对于采用 IRC 协议或 HTTP 协议 C/S 模型进行通信的僵尸网络，基于 P2P 协议的僵尸网络具有更好的健壮性、隐蔽性和扩展性。

2.2.2 僵尸网络的 C&C 体系架构

Botmaster 想要在互联网中控制僵尸网络从事一些非法恶意活动，不仅要保证 Bot 程序在本地自身安全以及通信会话过程的隐蔽，如何在 Bot 主机间通信、快速地分发指令也是僵尸网络中一个核心问题。在本小节中，根据僵尸网络的拓扑结构、工作模式的不同主要介绍中心型 C&C 、非中心型 C&C 和混合型 C&C 三种类型的体系架构。

(1) 中心型 C&C

中心型 C&C 是采用与传统的 C/S 网络模型相似的架构实现的，例如采用 IRC

协议作为C&C 信道通信协议的僵尸网络或者采用HTTP协议作为作为C&C 信道通信协议的僵尸网络都是中心型C&C架构。在中心型C&C 架构中，所有的Bot 主机与 C&C 服务器建立通信信道并利用 C&C 服务器向所有的 Bot 主机发送命令或者提供 Bot 程序的更新服务。中心型 C&C 具有快速响应、协作性好的特点，Botmaster 通过反馈信息可以很容易的检测僵尸网络的状态、掌握整个网络的基本信息，例如僵尸网络 Bot 主机在 Internet 中活动的数目、在整个 Internet 上的分布情况等等。然而中心型 C&C 架构的僵尸网络也存在一个十分致命的单点失效问题，整个网络可以很容易的被检测到和关闭。图 2-5 显示了中心型 C&C 架构的僵尸网络通信模型，在该模型中有两个中心 C&C 进行 Botmaster 和 Bot 主机之间命令分发、消息转发。

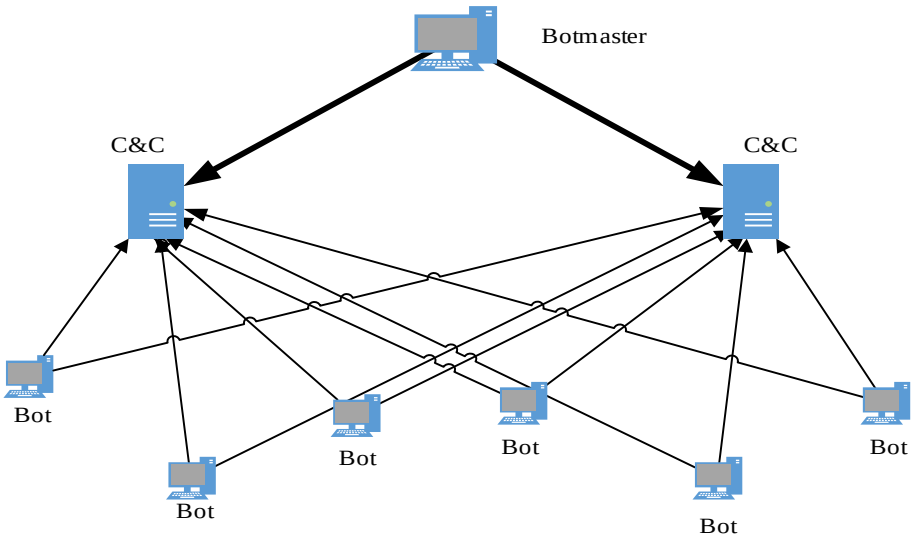

图 2-5 中心型僵尸网络的 C&C 架构

目前中心型 C&C 僵尸网络使用最多、最广泛的网络协议有 IRC 协议和 HTTP 协议。以 IRC 协议为例，网络中 Bot 主机通过 Botmaster 在 C&C 服务器上创建 IRC 信道与 C&C 服务器连接，等待接收命令、执行恶意操作。在 Symantec 2010 年网络安全威胁报告[36]中指出 2009 年发现的中心型僵尸网络中使用 IRC 协议作为主要通信协议僵尸网络占总数的 31%左右，这也说明基于 IRC 协议僵尸网络是目前 Botmaster 最常采用的一种中心型僵尸网络。图 2-6 显示了在基于 IRC 的僵尸网络中一个中心 IRC 服务器转发命令和数据。

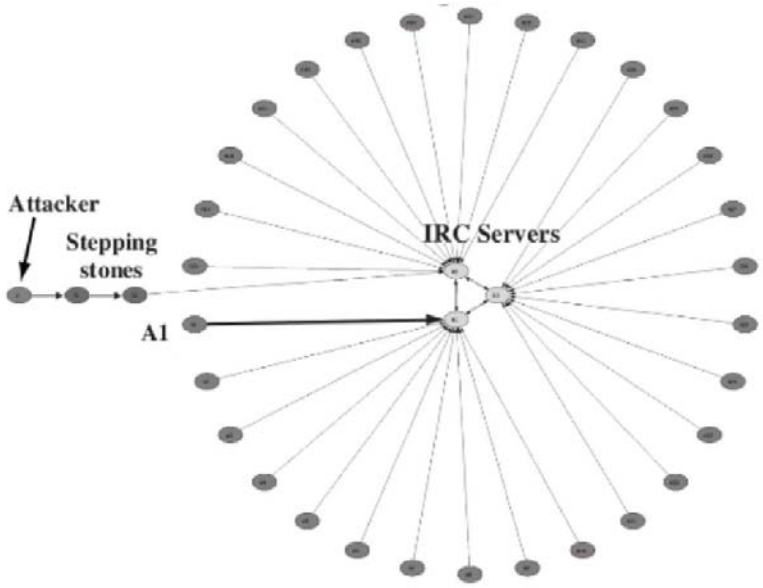

图 2-6 基于 IRC 的僵尸网络（来源：H. R. Zeidanloo etal.[37]）

随着越来越多的研究人员将注意力放在 IRC 流量监测，通过 IRC 流量异常来检测僵尸网络，以及防火墙软件封堵了一些不希望开放的端口（例如 IRC 端口等），基于 IRC 僵尸网络在 Internet 中的生存能力开始逐渐下降。网络攻击者发现 HTTP 协议具有流量大、容易绕过防火墙等安全软件的检测的特点，便将其作为 C&C 信道通信协议使得僵尸网络更加难以检测。如图 2-7 所示，基于 HTTP 协议僵尸网络采用“PULL” 的方式获取消息，而基于 IRC 协议僵尸网络则使用“PUSH” 方式。目前已知的使用 HTTP 协议的僵尸网络有 Bobax[38]、ClickBot[39]和 Rustock[40]等。

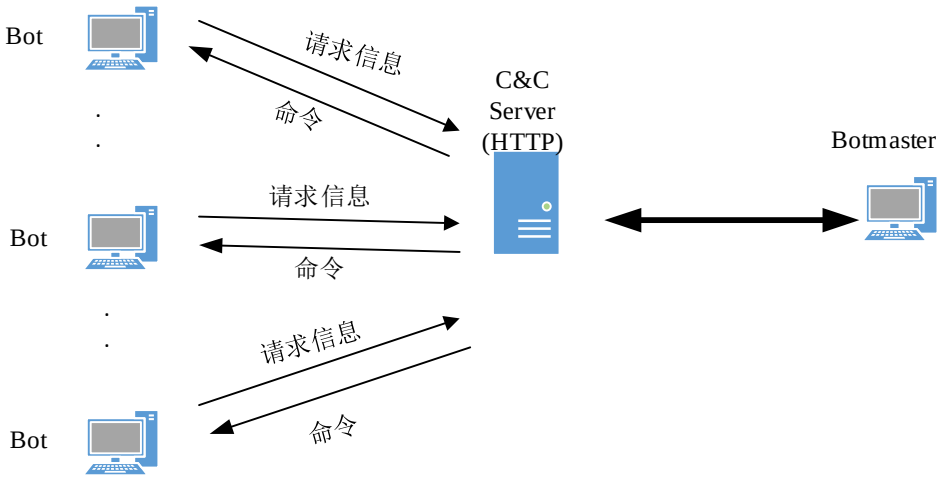

图 2-7 基于 HTTP 协议僵尸网络框架

(2) 非中心型 C&C

由于中心型 C&C 僵尸网络存在单点失效、易关闭等缺陷，攻击者们希望构建的僵尸网络具有很好隐蔽性、难以摧毁、对中心服务器的依赖性比较低且允许网络中一些 Bot 主机被发现和破坏。基于这几点要求 Botmaster 采用容错能力更好、结构更加复杂的 P2P 模型作为 C&C 结构模型，基于 P2P 模型的僵尸网络比中心型 C&C 僵尸网络具有更好的鲁棒性，随着基于 P2P 模型被 Botmaster 广泛使用创建僵尸网络将会给网络防御工作带来更大的困难和挑战。

图 2-8 所示在 P2P 模型中没有中心节点存在，每一个 Bot 主机与网络中其他某些 Bot 主机建立连接，网络中 Bot 既充当客户端角色又具有提供服务的功能，一个新的 Bot 主机只要知道僵尸网络中某一个 Bot 主机的地址就能够能够加入网络，而且网络在部分 Bot 处于非活动状态状态下依然能够正常运行。

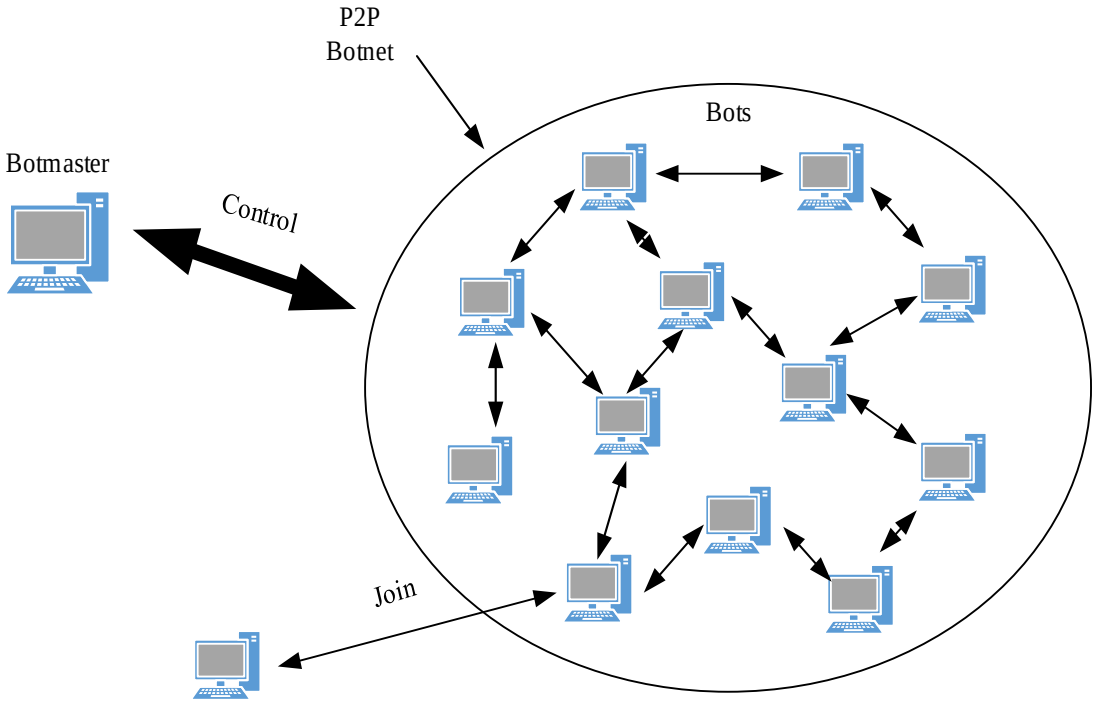

图 2-8 P2P 僵尸网络架构示意图

非中心型僵尸网络根据僵尸网络非中心化程度的不同可分为部分非中心化和完全非中心化两种。在完全非中心化僵尸网络中，Botmaster 可以注入一条命令给网络中任意一个 Bot 主机，也可以广播给某一类 Bot 主机。由于可以从网络中任意一个 Bot 主机注入命令，为避免向该网络注入错误命令，通常采用一些认证机制来保证命令的正确性和安全性。

第一个已知的 P2P 僵尸网络是 2003 年出现的 Slapper worm[41]，其他一些著名的 P2P 僵尸网络有 Sinit[42]，Phatbot[43]，Nugache[44]和 Storm worm[45,46]。P2P 僵尸网络虽然很好的消除或隐藏了中心型僵尸网络存在单点失效等问题，但由于网络每一个节点都保存其他一些节点的信息，容易遭到虚假信息攻击导致节点保存信息错误、整个网络处于紊乱状态。

(3) 混合型 C&C

混合型 C&C 结合了中心型 C&C 和非中心型 C&C 的特点将整个僵尸网络的 Bot 主机分成不同的级别和不同的角色，例如在 Ping Wang 等提出的一种混合型 P2P 僵尸网络[14]中，Bot 程序被分成两类：a) Servant Bot，该类型 Bot 程序在网络中充当两种网络角色（即服务器端和客户端），具有静态的、可路由的 IP 地址，且能够在 Internet 任何位置连接到；b) Client Bot，该类型 Bot 程序只能够主动连接其他 Bot 程序，该 Bot 程序的主机 IP 地址一般是动态分配的、私有的且无法路由到的，该节点主机通常隐藏在防火墙后面、无法从外部 Internet 链接到。

根据图 2-9 所示为 Ping Wang 等人提出的混合型 P2P 僵尸网络结构模型，其具有以下特点，

a) 在 Bot 节点只保存部分 Servant Bot 的 IP 地址在网络节点列表中；
b) Servant Bot 持续监听自定义端口、等待接收其他 Bot 程序发送的消息或者命令；
c) 所有 Bot 定期地与网络节点列表中的 Servant Bot 节点进行通信、获取 Botmaster 发送的命令；
d) 当 Servant Bot 接收到一个新的、从未接收到的命令时，它会快速的将这条命令转发给其节点列表中保存的所有 Servant Bot 节点。

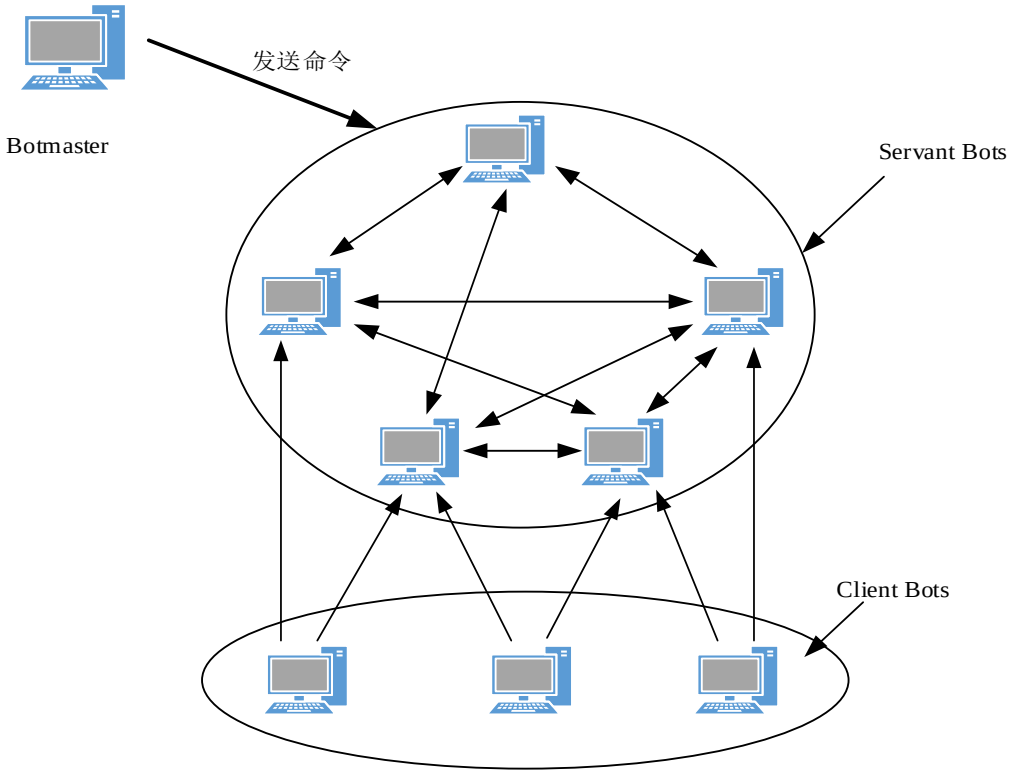

图 2-9 混合型 P2P 僵尸网络的 C&C 架构

2.2.3 僵尸网络最新技术

随着僵尸网络检测技术的不断提高、检测手段的不断完善，Botmaster 采用了更多的新技术来增强网络隐蔽性、延长僵尸网络生存时间、提高僵尸网络检测的难度。目前僵尸网络中采用的一些新技术主要有以下几种，

(1) Fast Flux

在最新出现的一些僵尸网络中，Botmaster 经常使用 FFSNs（Fast-Flux Service Networks）[47,48]技术作为 C&C 信道机制。Fast-Flux 是一种 DNS 技术，利用一个已被攻击的、网络地址经常变化的主机充当一个代理来隐藏钓鱼网站或散播恶意软件的网络站点。同时，一些网络恶意程序通常会将 Fast-Flux 技术与 P2P 网络、分布式 C&C、基于网络的负载平衡和代理重定向等技术结合使用，以此来对抗攻击发现和攻击对策。

Fast-Flux 技术提供匿名机制，使用一个具有公共 IP 地址的、妥协的主机作为代理，并将真正的 C&C 服务器域名与这些代理主机的 IP 地址绑定、隐藏在代理后面提供服务，通常为保证安全及长期可用会对服务器域名绑定的 IP 地址进行定期更新变换。Botmaster 将 Fast-Flux 技术应用到僵尸网络构建成一个动态代理网络，不仅保护了僵尸网络控制服务器，而且隐藏了僵尸网络真实拓扑结构及工作机制，

具有很好的隐蔽性，例如僵尸网络 Storm worm[45,46]的实现就使用了 Fast-Flux 技术。

(2) Domain Flux 技术

Domain Flux 技术是对基于 HTTP 协议 C&C 信道的一种变形，其利用 DNS（Domain Name System）中域名查询机制以及防火墙等安全软件对其放行的策略在僵尸网络中进行通信、命令分发等操作。Domain Flux 技术在僵尸网络中共享一个由域名生成算法产生的包含大量动态域名的域名池，Botmaster 从域名池中随机选择域名进行注册并充当控制端服务器域名，Bot 程序采用对整个域名池轮询的方式查找控制端服务器。

目前已知的采用 Domain Flux 技术构建 C&C 信道的僵尸网络包括 Srizbi[49]、Torpig[50]、Confickcer[51] 等，其中 Conficker 中使用的 Domain Flux 技术最为成熟。至今为止，Conficker 共包括了 A~E 六个版本且每个版本的动态域名池的规模不尽相同，这也是从 C&C 信道区别 Conficker 版本最主要的方法。动态域名池的规模不断扩大，给安全人员的防御工作创造了极大地困难和挑战，同时也是采用 Domain Flux 技术的僵尸网络至今仍然得以存活的重要原因。

(3) URL Flux 技术[52,53]

URL Flux 技术采用的原理与 Domain Flux 原理十分相似，都需要在僵尸网络中间共享一套生成算法和一个资源池，不同的是 URL Flux 技术通过一个目标地址池对僵尸网络进行控制而不是一个动态域名池，Domain Flux 的实现必须要有一个公网 IP 地址存在，而 URL Flux 技术仅仅需要 Web 2.0 服务器即可。

目前互联网中的网络用户不再只是传统的信息接收方，也可能是网络中制作、分享信息的一方，以网络信息分享为目的的 Web 2.0 技术迅速发展给僵尸网络 C&C 信道创建提供了一种新的方式和机遇，例如目前网络上非常流行的 BLOG 、Weibo 等都可能被 Botmaster 用于构建 C&C 信道。同时，Web 2.0 对用户提供域名个性化、自定义设置的特征也让 Bot 程序寻找控制服务器地址的操作变得更加方便、简捷。

2.3 僵尸网络检测技术

在上一节中，我们从僵尸网络的通信协议、C&C 架构以及僵尸网络最新技术三个方面进行简单介绍并有了一定的了解，在本节中将会对目前网络中出现的僵尸网络检测技术进行简要说明和分析，通过对检测知识的学习有利于我们提出一种新的检测方法。

目前比较流行的僵尸网络检测技术主要是通过监测网络流量或者主机行为是否异常进行判断，根据检测数据来源和异常模式的不同可以将网络中经常使用的

检测方法大致分为基于网络流量内容的检测技术、基于网络行为的检测技术、基于时间关联的检测技术和基于日志相关性的检测技术四类。

2.3.1 基于网络流量内容的检测技术

基于网络流量内容的检测技术作为入侵检测系统常采用的一种检测技术，其首先对网络流量的内容特征展开分析，然后与网络中已发现的僵尸网络的特征进行匹配，判断在网络中是否存在该种僵尸网络。该检测技术需要维护一个特征库，以便能够进行快速匹配和检测准确性，这种方法仅能够检测到那些已知的且具有显著特征的僵尸网络，例如使用 TCP 端口号为 8 的 Nugache、 采用 UDP 端口号 53 的 Sinit 以及最新发现的 Phatbot，Conficker 等都可以通过该技术检测出来。

由于基于网络流量内容的检测技术能够准确、快速的检测到僵尸网络，且容易部署，被安全人员广泛用于检测僵尸网络。但是该检测方法对于未知的僵尸网络、变形的僵尸网络、或者具有多种形态的僵尸网络是无法检测到的， 而且这种方法需要有人工操作，用于检测急剧增加且变化多端的僵尸网络的效果并不明显。

2.3.2 基于网络行为的检测技术

由于僵尸网络在通信过程中具有一定的时间关联性和群体相似性的特性，研究人员便利用这个特性对僵尸网络的网络流量行为进行分析，期望能够找到具有一般性的网络异常行为模式，并根据此模式来检测僵尸网络。

Nagaraja 等人提出的 BotGrep[54] 能够通过流量分析成功检测到结构化 P2P 僵尸网络的存在。首先，BotGrep 收集 ISP 骨干网上的流量并得到节点间的流量分布图；然后，在流量分布图中利用随机游走方法提取出结构化 P2P 网络的子图；最后将子图与其他检测系统如蜜罐等的检测结构结合起来判断该结构化 P2P 网络是否为僵尸网络。

虽然，基于网络行为的检测技术能够检测到未知僵尸网络的存在，且很难被逃避，但是由于网络流量多样性、复杂性和及其呈现出来的庞大数据量，这种方法很难做到实时检测和保证准确度，并未在实际应用中广泛使用。

2.3.3 基于时间关联的检测技术

由于 Botmaster 常常控制僵尸网络中 Bot 主机从事一些恶意主机行为，因此僵尸网络的网络流量和其从事的一些恶意事件，如 DDOS 攻击、恶意软件传播、二进制文件下载等具有一定的时间关联性。Gu 等利用此特性，提出了 BotSniffer[55] 和 BotMiner[56] 两种方法。在 BotSniffer 中假设处于同一个局域网中的 bot 主机活

动在时间和空间上都存在一定的关联性，根据网络中的流量发现可疑的僵尸网络C&C 网络流量，再与异常事件日志文件进行关联分析进行检测。而与 BotSniffer 对具体网络协议流量进行分析不同的是，BotMiner 首先根据目的地址和端口对所有流量进行聚合分组，再根据流量属性进行聚合分析，并将结果与异常事件日志进行关联分析，最终得到一组可疑的 bot。

这种方法把僵尸网络的网络行为和一些已知的恶意事件关联起来检测僵尸网络，不仅减小了检测的范围，而且提高了检测结果的精确度。但是其依然存在一些缺陷，例如，BotSniffer 依赖于具体的网络协议检测方法，只能检测特定类型通信协议的僵尸网络。

2.3.4 基于日志相关性的检测技术

诺丁汉大学 Al-Hammadi 等人提出的基于日志相关性的僵尸网络检测方法与 Malan 等人的方法类似，其出发点都是基于僵尸主机行为的相关性。这种方法在每台检测主机记录网络通信调用 Windows API(application programming interface) 函数的具体信息并生成日志， 由于僵尸主机行为的同步性，它们对 Windows API 的调用时间上也会非常一致。根据日志文件中各台主机调用 API 信息的时间相关性判断是否存在僵尸网络，该方法是一种离线检测方法，而且只针对 Windows 平台的主机。

2.3.5 僵尸网络检测技术分析

关于目前已有的僵尸网络检测技术本文主要介绍了上述四种检测方法，虽然这些检测方法比较成熟，但依然存在一些问题和不足的地方。

(1) 基于网络流量内容的检测方法需要建立一个僵尸网络特征库，根据特征库内容进行检测，具有一定的滞后性且只能检测那些已知的僵尸网络；

(2) 基于网络行为的检测方法在网络数据量比较大情况下，其效率会降低且无法检测未知的僵尸程序；

(3) 基于时间关联的检测技术利用僵尸网络在时间上的关联性进行检测虽然减小检测范围、提高检测精度，但只能检测某些协议类型的僵尸网络；

(4) 基于日志关联性的检测方法是一种离线方法，实时性较差。

除了上述介绍的四种检测方法，一些其他的检测方法也同样存在一些问题和劣势，例如基于蜜罐蜜网技术的检测方法，该方法利用通过蜜罐和蜜网获取僵尸程序的技术比较成熟，但由于在蜜网和蜜罐的部署非常复杂，且对僵尸网络进行反汇编、分析其行为特征也非常困难导致该方法也会异常的复杂和困难。一般使

用于非常有实力的科研组织，且需要将蜜罐蜜网部署在网络各个关键节点才能够取得较好检测效果。同时该技术通常只能对一种或一类僵尸网络进行检测研究，检测实时性比较低。基于 DNS 协议的检测方法只通过 DNS 信息进行检测，忽略带有加密的僵尸网络信息，以及探测僵尸网络的群体行为，有利于大规模检测。但是该方法需要把检测系统部署在骨干 ISP 节点，威胁互联网中隐私信息，且需要比较长时间检测。

通过对现有僵尸网络检测技术的分析总结为本文的僵尸网络检测方法的提出提供指导思路。

2.4 僵尸网络检测研究趋势

从当前僵尸网络检测的研究现状可以看出,相似性和协同成为未来检测技术发展的方向。由于僵尸网络本身是人为编写的恶意程序，其运行不可避免地具有规律性，而僵尸网络中大量僵尸程序的运行，它们的行为必然能表现出一定的相似性和规律性。

(1) 相似性检测

以行为或其它方面的相似性作为检测的突破口的思想在现有文献已经出现。突出的例子是 BotSniffer，它在异常检测的基础上，利用 bots 在响应行为上具有的强同步性和内容的相关性来进行检测。此外，李翔等，在基于主机的 P2P 流量 检测和恶意行为检测的基础上,提出一个通过对 P2PBot 主机行为进行融合分析的 P2P 僵尸网络的监测模型[58]；王涛等，提出一种基于网络群体行为特点分析的检测方法[59]；李晓桢等，从僵尸网络的基本要素着手，提出对通信聚类和行为聚类进行关联分析的检测方法[60]；梁其川等，提出了通过对网络中主机通信状况和网络活动状况进行协同分组，在对聚类结果进行关联分析的检测方法[61]。这些研究都与相似性有着非常密切的关系，而且随着僵尸网络的不断发展，僵尸网络的隐秘性也在不断加强，但本质上，大量僵尸程序的运行产生的行为的相似性是无法隐藏的，这是相似性观点在僵尸网络检测中的发展方向的重要依据。

(2) 协同检测

随着社会的不断发展，网络环境日趋复杂，由于互联网中网络之间相互独立，僵尸网络往往能够在松散的互联网体系中隐藏自己，这也是僵尸网络难以检测和防御的重要原因。现有僵尸网络检测方法往往通过分析单一主机或网络中的行为或流量特征进行检测，者无法全面获取僵尸网络各阶段的行为特点，成为检测短板。协同检测的核心即把网络中分散的信息进行集中分析从而发现其中隐藏的僵尸网络，从宏观上发现僵尸网络的特点，成为检测和反制的重要发展方向。在现

有文献中，协同思想更多的出现在入侵检测系统的相关研究中，伴随着分布式入侵检测系统的出现而出现，在僵尸网络方面的检测较少，其中，Hailong Wang 等，提出的基于协同的僵尸网络检测模型[62]；肖斌等提出分布式 Botnet 系统预警模型。僵尸网络检测从本质上可归为入侵检测的范畴，将协同检测的思想融入僵尸网络的检测将是未来僵尸网络发展的重要方向[63]。

协同检测可以在更大范围内寻找僵尸节点之间的相似性，这两个发展方向相互促进，是不可分离的两个重要发展方向。

2.5 本章小结

在本章中，首先对僵尸网络的定义、生命周期、感染途径和恶意行为几个方面进行介绍，具备了对僵尸网络概括性的认识。然后，重点介绍了僵尸网络 C&C 信道协议、体系结构以及最新技术，对僵尸网络工作过程和原理有了一定程度的了解。最后，对目前常用的僵尸网络检测方法进行介绍、总结和分析。经过本章梳理，对僵尸网络有了全面详细的认识，为本文僵尸网络检测方法的提出打下坚实基础。

第三章 基于相似性分析的检测模型设计

因为僵尸网络程序单独的攻击行为或者不具有恶意攻击行为的僵尸网络对整个网络安全并不能构成威胁，因此本文的研究工作主要是针对利用命令控制信道收发命令、执行网络恶意行为的僵尸网络展开的。

3.1 检测模型的目标

检测并找到被监测网络中属于同一个僵尸网络的 Bot 主机，且与僵尸网络采用何种 C&C 架构、何种通信协议无关是本文提出的检测模型的目标。本文利用主动分析的方法对被监测网络的网络流量进行分析，并根据同一僵尸网络主机群体相似性的特点进行一定的相似聚类分析实现检测模型的目标。

本文提出的僵尸网络检测模型要实现的目标主要包括如下几点，

(1) 独立于僵尸网络的C&C架构和C&C通信协议。无论Botmaster构建的C&C信道架构是中心型、非中心型或者混合型的，还是采用 IRC 协议、HTTP 协议或者 P2P 协议等作为 C&C 通信协议，检测模型都能够检测到 Bot 主机的存在。

(2) 独立于僵尸网络通信的具体传输内容和意义。在对网络数据流分析时只根据网络数据包 IP 层和传输层的信息提取网络数据流的相关特征信息，不对僵尸网络传输内容的意义进行分析。

3.2 检测模型的提出

在第二章中，我们对僵尸网络以及僵尸网络检测相关知识进行详细介绍，通过这些知识的学习可以发现目前所有的僵尸网络都具有如下两个特性：a）C&C 信道；b）攻击性。C&C 信道在僵尸网络中被用于 Botmaster 与 Bot 之间信息的传送，Botmaster 利用 C&C 信道控制着整个僵尸网络的运行以及 Bot 主机的行为。僵尸网络由于经常被 Botmaster 当作一个平台，控制着网络中的机器对互联网上的合法用户进行攻击，破坏了正常的网络秩序，具有一定的攻击性，这也是检测僵尸网络并破坏的原因所在。利用这两种特性我们可以将僵尸网络和网络中其他病毒、木马等恶意程序区分开来，例如木马虽然能够进行各种各样、不同程度的恶意行为，但是木马之间没有 C&C 信道的存在常常表现出“个人英雄主义”，其产生的破坏性也是无法与僵尸网络的群体攻击相比的。

由于 Bot 程序的运行对其主机合法用户是透明的，为使其能够自动与 C&C 服

务器进行联系，Botmaster 一般会将 Bot 加入网络的过程进行程序化，同时所有的 Bot 程序统一的接收 Botmaster 的命令。因此，无论 Botmaster 采用何种形态的 C&C 信道构建僵尸网络，网络中所有的 Bot 主机都会表现出相似的网络流量。同时，当 Botmaster 向网络中全部或者部分 Bot 主机发送攻击命令后，这些主机会执行相同的攻击动作，因此当僵尸网络发动攻击时也会表现出一定的相似性。

综上，对于任何一种类型僵尸网络，无论其 C&C 信道采用何种拓扑形态、何种通信协议，同一僵尸网络 Bot 主机在网络流量和恶意行为方面都会存在一定的相似度。基于论文的研究内容，在本文中提到的主机行为指的是僵尸主机在收到控制端的远程控制下表现出的攻击行为或者恶意行为，例如 HTTP REQUEST FLOOD 攻击、DDOS 攻击等，而与通常理解的主机 API 调用、内存使用、数据访问等主机行为不同。本文以同一僵尸网络中 Bot 主机表现的网络流量和主机行为具有群体相似性作为基于相似性分析的僵尸网络检测系统模型提出的理论根据和立足点，本系统模型的设计如图 3-1 所示，检测收集网络数据流量，对数据流分别进行网络流量和主机行为聚类和相似性分析，找出可能属于同一僵尸网络的一组网络主机。

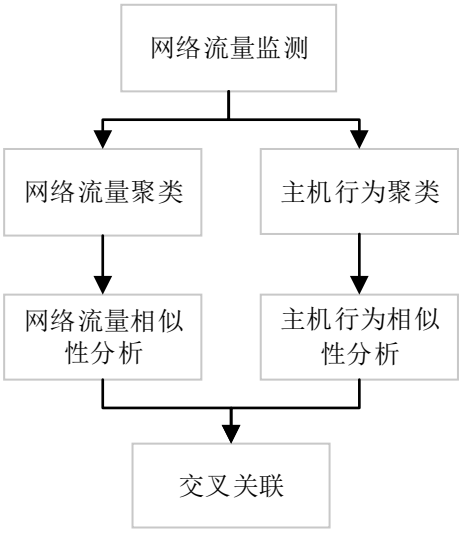

图 3-1 监测系统模型图

3.3 检测模型的设计

本小节将对在上一小节提出的检测系统模型的网络抓包、网络流量分析、主机行为分析和交叉关联四个模块进行详细介绍。如图 3-2 所示，我们提出的基于相似性分析的僵尸网络检测系统模型主要包括四个模块:

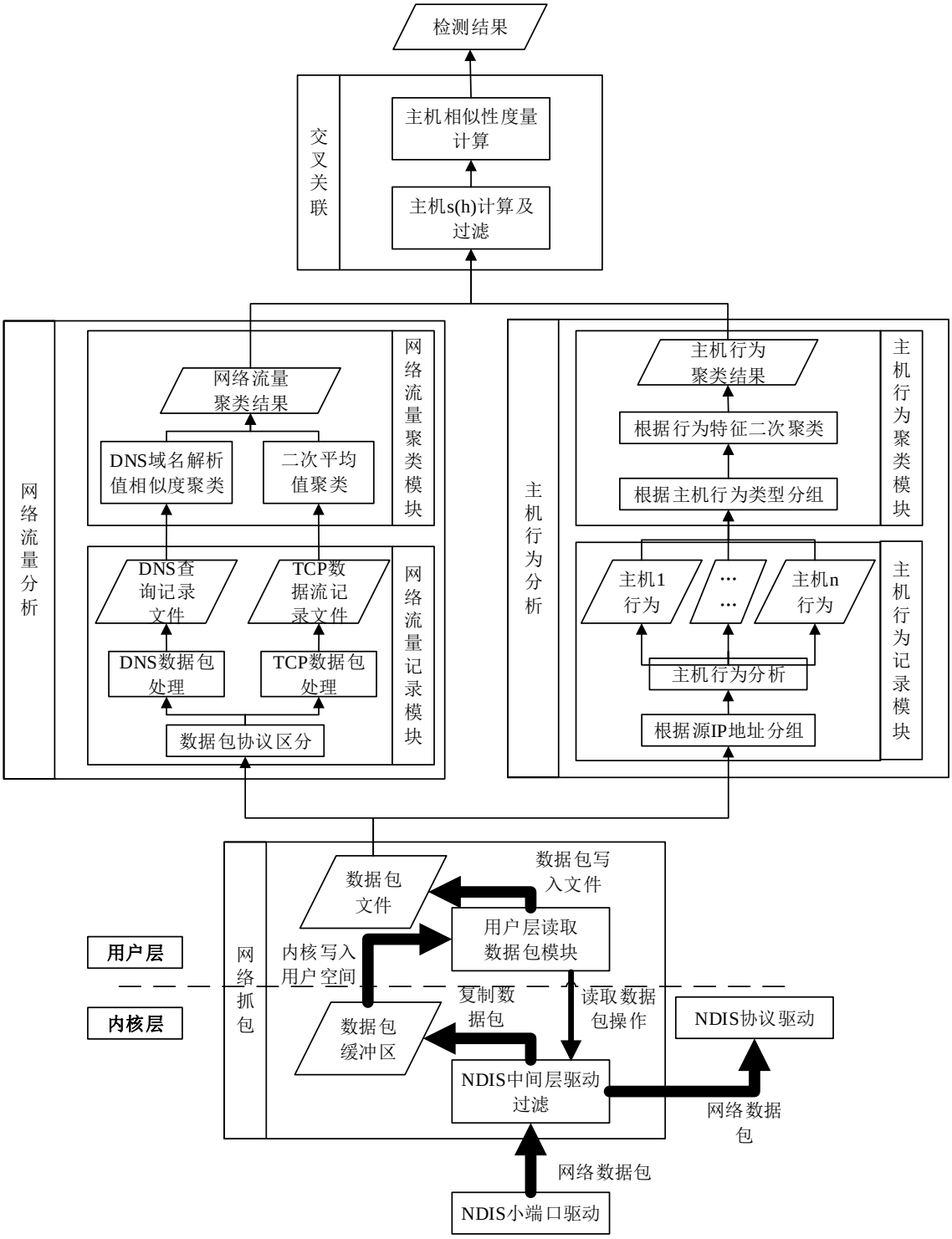

图 3-2 检测系统模型

(1) 网络抓包模块：收集被监测网络中主机与外部网络通信的网络数据包。

(2) 网络流量分析模块：分析网络数据流，提取网络流量特征信息，聚类得到具有形似特征的主机。该模块分成网络流量记录和网络流量聚类两个功能

模块，其中前者对抓取的网络数据流进行分析提取特征信息并以一定格式记录，后者对记录的数据信息进行聚类计算找出相似的主机。

(3) 主机行为分析模块：分析每个主机发送的网络数据流发现该主机存在的可疑的网络活动，聚类分析找出具有相同活动的主机。该模块分成主机行为记录模块和主机行为聚类模块，前者分析抓取的网络数据流匹配出每台主机可疑的网络活动并以一定格式保存，后者对保存的每台主机可疑网络活动进行聚类分析找出相似主机行为的主机。

(4) 交叉关联模块：该网络流量聚类模块和主机行为聚类模块的聚类结果结合在一起，并利用交叉关联的方法进行分析，最终生成一份检测报告。

3.3.1 网络抓包模块

网络抓包模块的主要功能是收集被监测网络内部与外部网络通信的数据流，为收集到所有的数据包，我们将该模块部署在被监测网络与其他网络的交界处，例如网关。如图 3-3 所示，我们只对 TCP 协议和 DNS 协议感兴趣，其中网络抓包模块对于 DNS 数据包会记录查询的域名和域名对应的 IP 地址信息，而对 TCP 数据包则只记录报头部分信息，因此减小了记录文件的大小和系统的负载，进一步提高了数据流检测的效率。

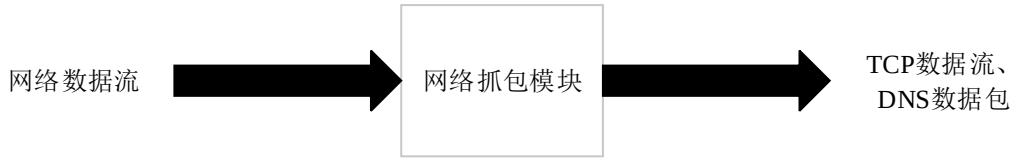

图 3-3 网络抓包模块示意图

3.3.2 网络流量分析模块

网络流量分析模块的主要功能是分析网络数据流，提取网络流量特征信息，聚类得到具有形似特征的主机，该模块分成网络流量记录和网络流量聚类两个功能模块，其中网络流量记录模块功能是对抓取的网络数据流进行分析提取特征信息并以一定格式记录，网络流量聚类功能是对前者记录的数据信息进行聚类计算找出相似的主机。

(1) 网络流量记录模块

我们在收集网络数据的过程中只对 TCP 协议和 DNS 协议的数据包感兴趣，但是这两种协议在网络中表现的形式是截然不同的，例如 TCP 协议在网络上以数据流形式存在，且通信过程的开始和结束要分别经历三次握手建立连接、四次握手

释放链接两个阶段，可以将三次握手和四次握手作为一次完整的通信过程的标识，并将此时间段内的数据流为单位对网络流量进行分析。DNS 协议在网络层使用的是 UDP 协议，通信过程不需要进行三次握手和四次握手，只能通过每一个 DNS 报文来分析网络流量。

其次，为避免那些我们不感兴趣的网络流量的影响，采取如下几步对网络流量进行过滤，a）对于非 DNS 网络流量，只记录从被监测网络内部向网络外部发送的网络流量，这样避免了网络内部主机之间通信流量和从外向内发送的网络流量的干扰。b）利用白名单机制进行流量过滤，对于 DNS 域名在白名单上的 DNS 流量采取不记录操作，同样对于发往白名单上 IP 地址的数据流也采取不记录操作。白名单中内容我们选择一些知名网站（如 Alexa）上提供的部分数据来构成，或者自定义白名单中数据。通过这两步过滤操作，不仅减少了正常网络流量的干扰，而且大大降低数据处理的工作量，在一定程度上减轻系统负载、提高系统效率。

在记录基于 TCP 协议网络流量的数据流过程中，为更好的描述每个数据流以及在下一步对数据流进行聚类操作，通过实验分析，我们采用数据流每个数据包的平均字节数（average bytes per packet，abpp）和数据流平均每秒字节数（average bytes per second，abps）作为特征值对数据流进行描述。为正确计算数据流的这些特征值需要准确的判断检测的数据包属于哪一个数据流，因此网络 TCP 数据流的记录过程如图 3-4 所示。

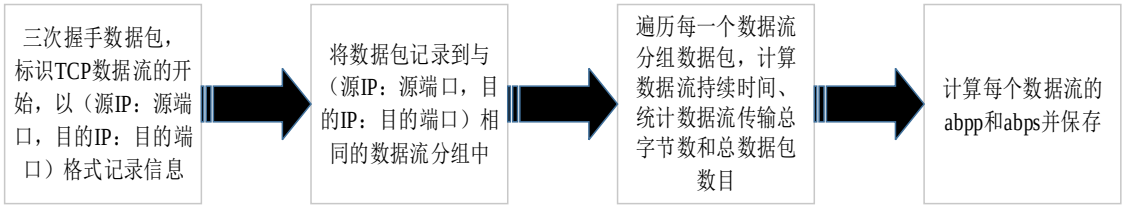

图 3-4 网络数据流记录过程示意图

相对于 TCP 协议需要将每个数据包正确的记录在相应的数据流中，DNS 协议数据包的记录则要显得简单、容易，通过解析 DNS 数据包，记录 DNS 查询域名、域名对应的 IP 地址、源 IP 地址、查询时间等等信息。

(2) 网络流量聚类模块

网络流量聚类模块的功能是通过对网络流量记录模块产生的数据分析，把在网络流量上具有一定相似性的主机归并成一组。与网络流量记录模块采用两种方法来记录 TCP 数据流和 DNS 数据包，我们在该模块也采用两种聚类方法对记录数据进行分析归类。

根据 TCP 数据流保存的数据具有如下两个特征：a)数据一维性；b)聚类数据

的规模无法预知性。而平均聚类法能够在事先不知道分组的个数，从一维的数据中筛选出相近的数据放在一组，且只需要给定一个聚类因子的值。因此我们采用此方法对 TCP 数据流进行聚类操作，方法流程如图 3-5 所示。

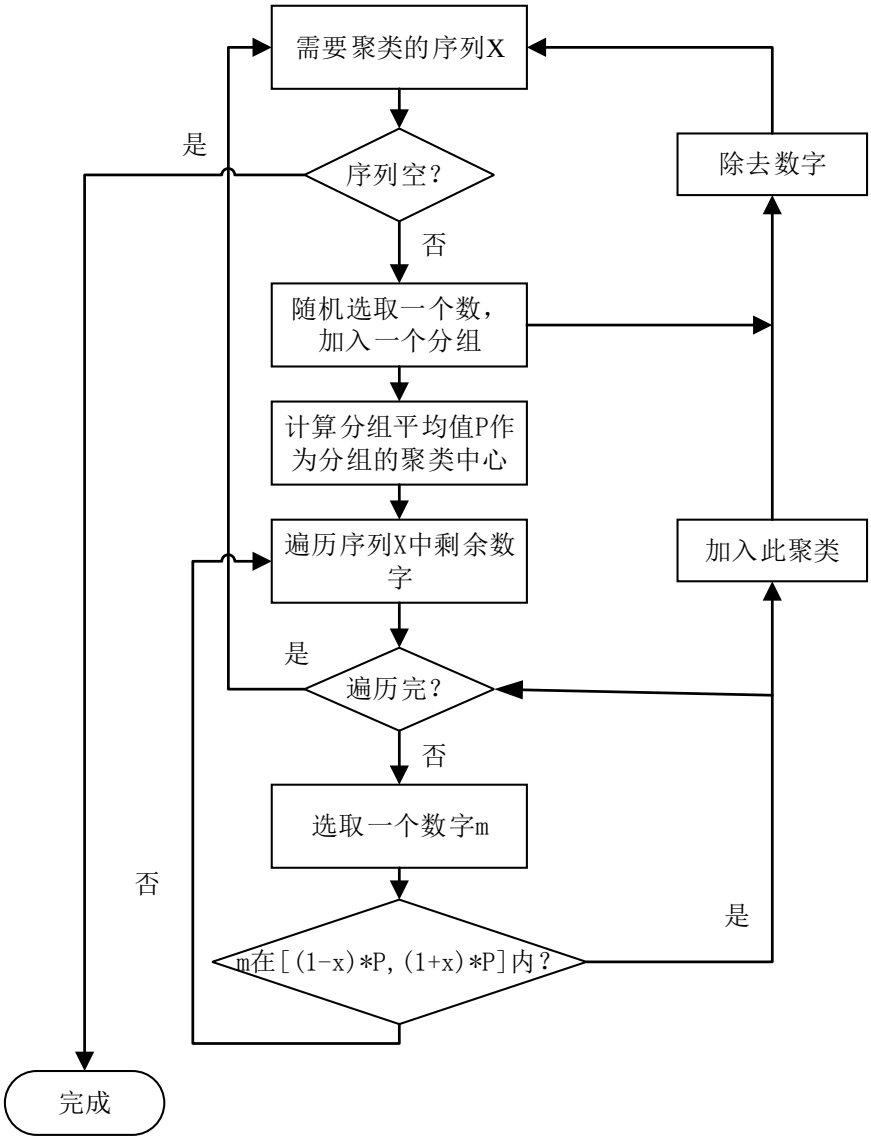

图 3-5 平均聚类法流程图

网络流量聚类模块分别利用数据流的 abpp 和 abps 形成向量，利用平均值聚类法进行二次聚类操作，然后从二次聚类得到的数据流提取源 IP 地址列表，最后除去重复的或者在其他分组中出现的分组。具体的聚类流程如图 3-6 所示。

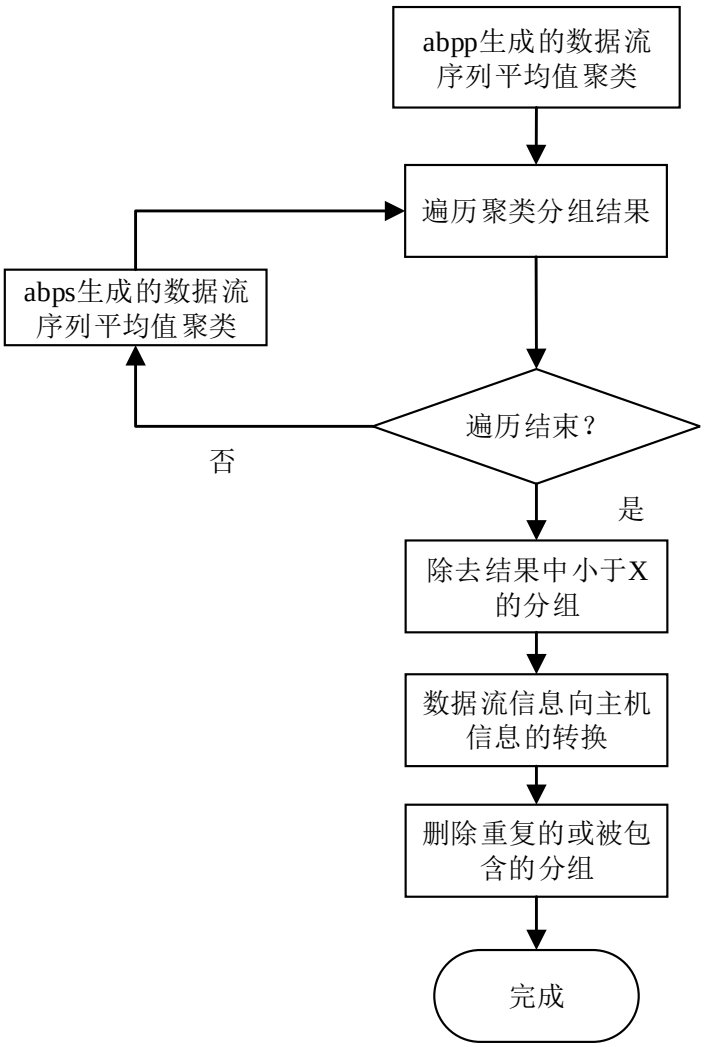

图 3-6 TCP 数据流二次聚类流程图

僵尸网络为实现迁移利用域名生成算法 DGA 随机生成一组域名，且这些域名在活跃期间都可以解析为属于同一个集合的 IP 地址，Bot 主机按照一定的时间周期对连接不同的域名，根据域名进行分组会得到一部分相同的 IP 地址。针对于 DNS 数据包聚类过程如下，

a) 首先根据 DNS 域名信息进行分类，获取该域名解析出的 IP 地址组；
b) 然后利用公式（3-1）计算两个不同域名之间解析 IP 地址组的相似度，当相似度达到某一个阈值的时候将这两个域名放在同一组；

$$\frac{(D_1 \cap D_2)}{(D_1 \cup D_2)}$$ ---公式（3-1）

其中，D1、D2 分别表示域名 Domain1 和 Domain2 所对应的解析 IP 地址的集

合。

c) 最后删除重复或者被包含的分组，将每一组域名信息转换成查询这些域名的主机 IP 地址。

通过网络流量聚类模块的聚类方法得到具有相似网络流量的主机列表，由于聚类操作的数据源和聚类算法都可能存在一定的误差，因此得到的聚类结果存有误测或漏测的可能性。为提高检测结果准确性，我们还需要将此结果同主机行为聚类结果进行交叉处理。

3.3.3 主机行为分析模块

主机行为分析模块的主要功能是分析每个主机发送的网络数据流发现该主机存在的可疑的网络活动，聚类分析找出具有相同活动的主机，该模块分成主机行为记录模块和主机行为聚类模块，主机行为记录模块的功能是分析抓取的网络数据流匹配出每台主机可疑的网络活动并以一定格式保存，主机行为聚类模块功能是对前者保存的每台主机可疑网络活动进行聚类分析找出相似主机行为的主机。

(1) 主机行为记录模块

主机行为记录模块主要目的是通过对被监测网络的通信数据进行检测，记录在此期间网络内部主机进行了何种操作，并找出可疑的行为，例如 DDOS Attack、二进制文件下载、Spamming、Scanning 等行为。

该记录模块首先将源 IP 地址相同的数据流分成一组，然后对该组的网络流量进行分析，判断此源 IP 对应的网络主机是否具有可疑的、恶意的行为，并且将可疑的行为记录下来。同时为了便于对本模块记录的数据进行聚类分析，将行为的类别以及行为定性说明以一个固定的格式保存。

本模块产生的结果是以网络内部每一台主机为单位进行流量检测得到的可疑行为记录，而在检测过程中可能会把正常主机的合法行为误认为可疑的、恶意的行为。如果仅仅根据这些记录数据来判断网络中的主机是不是 Bot 主机会产生很大的误差，为减小这种误差，还需要对本模块生成的可疑行为记录利用主机行为相似性特点进行分析。

(2) 主机行为聚类模块

僵尸网络在进行网络攻击时 Bot 主机会表现出相似的主机行为，主机行为聚类模块就是利用这个特点对主机行为记录模块产生的数据进行分析，得出具有相似性的主机。

对主机行为记录数据采用二次聚类分组操作，首先根据记录中行为类别（如 DDOS、Spamming、Scanning 等）进行分组，其次根据每一类行为的具体特征进

行二次聚类，如图 3-7 所示为主机行为二次聚类。

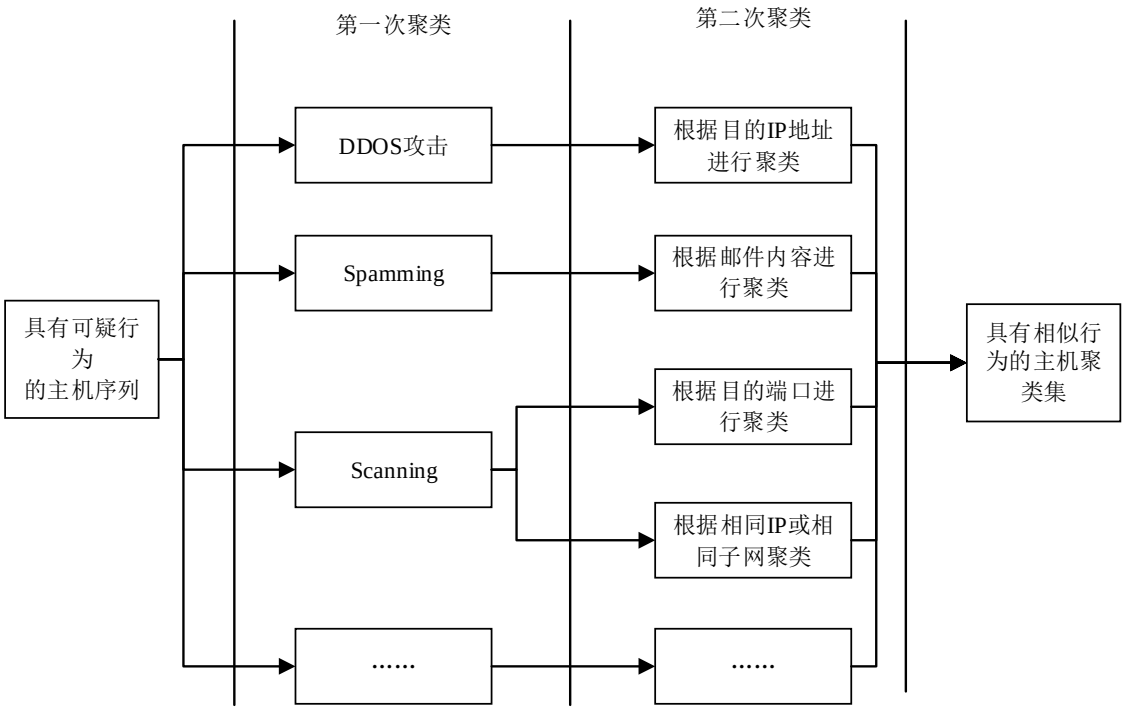

图 3-7 主机行为二次聚类示意图

为更好的理解聚类模块的工作原理，我们以对 HTTP REQUEST FLOOD 主机行为的聚类过程为例进行简要说明，首先通过对主机行为记录模块产生的主机行为数据进行分类，假设分类之后得到与 HTTP REQUEST FLOOD 主机行为有关的主机行为记录数据如下所示，

Web-1	192.168.2.2	8080	Web-1	192.168.2.4	8080
Web-1	192.168.2.12	8080	Web-1	192.168.2.15	8080
Web-2	192.168.2.3	80			
Web-3	192.168.2.6	80	Web-3	192.168.2.17	80
Web-3	192.168.2.11	80	Web-3	192.168.2.13	80
Web-3	192.168.2.19	80			
Web-4	192.168.2.11	8080	Web-4	192.168.2.6	8080
Web-4	192.168.2.17	8080	Web-4	192.168.2.19	8080

然后，我们根据攻击目标和端口对主机进行分组，共分成四组如下所示，

192.168.2.2 192.168.2.4 192.168.2.12 192.168.2.15

192.168.2.3

192.168.2.6 192.168.2.11 192.168.2.13 192.168.2.17 192.168.2.19

192.168.2.6 192.168.2.11 192.168.2.17 192.168.2.19

此时对分组结果进行过滤后的结果如下，

192.168.2.2 192.168.2.4 192.168.2.12 192.168.2.15

192.168.2.6 192.168.2.11 192.168.2.13 192.168.2.17 192.168.2.19

3.3.4 交叉关联模块

交叉关联模块的功能是根据网络流量聚类模块和主机行为聚类模块产生的两个结果集，对两个结果集进行交叉检测、计算，找到结果集中主机之间的关联，最终得到同一僵尸网络的主机。该模块的交叉计算过程分成两个部分：a)对主机行为聚类模块聚类结果集中具有至少一种可疑行为的主机计算出一个值 s(h)，过滤那些值 s(h)小于某一特定阈值 θ 的主机；b)对主机行为聚类结果集中剩余的主机与网络流量聚类结果集中的主机进行相似性度量的计算，并根据该计算结果对 a)中剩余的主机进行分组。整个模块的聚类过程如图 3-8 所示。

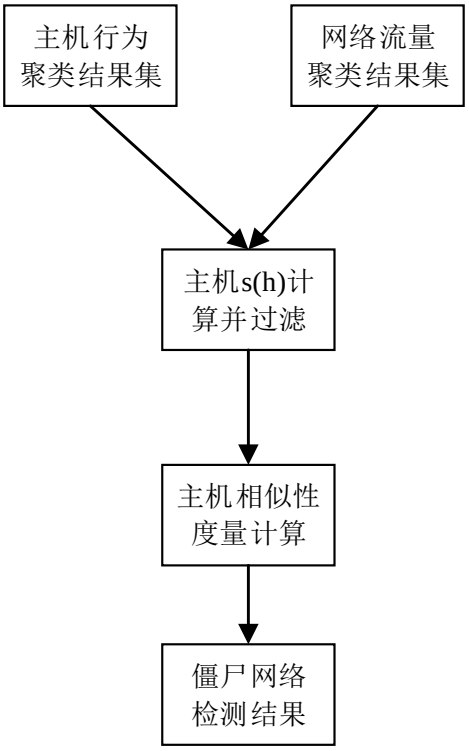

图 3-8 交叉关联流程图

(1) 主机 s(h)计算

假设集合 H 表示主机行为聚类模块生成结果中主机集合，h 表示只有表现一种可疑行为的主机，且 h∈H，$A^{(h)}=\{A_i\}_{i=1...m_h}$ 表示主机 h 表现出的 m_h 种可疑主机行为的集合，$C^{(h)}=\{C_i\}_{i=1,...,n_h}$ 表示网络流量记录结果集中与主机 h 有关的 n_h 个网络

流量的集合，主机 h 的值 s(h)计算公式如公式（3-2）所示。

$$s(\mathrm{h}) = \sum_{\substack{i,j \\ j>i \\ t(A_i)\neq t(A_j)}} \omega(\mathrm{A}_i)\omega(\mathrm{A}_j)\frac{\left|\mathrm{A}_i \cap \mathrm{A}_j\right|}{\left|\mathrm{A}_i \cup \mathrm{A}_j\right|} + \sum_{i,k}\omega(\mathrm{A}_i)\frac{\left|\mathrm{A}_i \cap C_K\right|}{\left|\mathrm{A}_i \cup C_K\right|} \qquad \text{---公式（3-2）}$$

其中 $\mathrm{A_i}$、$\mathrm{A_j} \in A^{(h)}$，$\mathrm{C_k} \in C^{(h)}$，$t(\mathrm{A_i})$表示可疑行为 $\mathrm{A_i}$ 的行为类型，例如 Scanning、Spamming、DDOS 攻击等，$\omega(\mathrm{A}_i) \geq 1$表示分配给可疑行为 $\mathrm{A_i}$ 的行为权值，且大小与行为的强弱有关。

如果主机 h 表现出多种类型的可疑主机行为且与主机 h 同一集合的其他主机也表现出相同的多种类型行为，那么主机 h 计算得到的值 s(h)就会比较高。例如，我们假设主机 h 表现出 Scanning 和 Exploit 两种类型可疑行为，令 $\mathrm{A_1}$ 表示具有 Scanning 可疑行为的主机集合，$\mathrm{A_2}$ 表示具有 Exploit 可疑行为的主机集合，集合 A1 和 A2 中重复的主机数量越大，值 s(h)越大。相同地，如果主机 h 所在的主机行为聚类集合与网络流量聚类集合有很大的重合，说明主机 h 所在集合中其他主机与它具有类似的主机行为和网络流量模式。

(2) 相似性度量计算

相似性度量计算是对（1）得到的结果计算主机之间网络流量和主机行为的相似度，当同一个主机行为聚类结果中的两台主机具有至少一种网络流量是一样的，那么就可以把它们聚类在一起。

假设集合 B 表示（1）得到的主机集合，首先过滤、去掉网络流量聚类结果 C 和主机行为聚类结果 A 中不包含集合 B 中任何一台主机的序列，分别得到 $C(\mathrm{B}) = \{\mathrm{C}_i^{(\mathrm{B})}\}_{i=1,\ldots,n_B}$ 和 $A(\mathrm{B}) = \{\mathrm{A}_i^{(\mathrm{B})}\}_{i=1\ldots m_B}$，取 $\mathrm{K^{(B)}} = \mathrm{C^{(B)}} \cup \mathrm{A^{(B)}} = \{\mathrm{K}_i^{(\mathrm{B})}\}_{i=1,\ldots,n_B+m_B}$；然后对于 B 中任一主机 h 利用二进制向量 b(h)描述，其中向量 b(h)第 i 个元素 $\mathrm{b_i}$ 取值为 0 或者 1，如果 $\mathrm{h} \in K_i^{(\mathrm{B})}$，则 $\mathrm{b_i}=1$；否则 $\mathrm{b_i}=0$；最后主机 $\mathrm{h_i}$ 与主机 $\mathrm{h_j}$ 相似值 $\mathrm{sim(h_i, h_j)}$利用公式(3-3)计算得到。

$$\mathrm{sim}(\mathrm{h}_i, \mathrm{h}_j) = \sum_{k=1}^{m_B} I(\mathrm{b}_k^{(\mathrm{i})} = \mathrm{b}_k^{(\mathrm{j})}) + I(\sum_{k=m_B+1}^{m_B+n_B} I(\mathrm{b}_k^{(\mathrm{i})} = \mathrm{b}_k^{(\mathrm{j})}) \geq 1) \qquad \text{---公式（3-3）}$$

其中，$\mathrm{b^{(i)}}$表示 $\mathrm{b(h_i)}$，函数 I(X)的取值与 X 有关，当 X 为真时，I(X)为 1；当 X 为假时，I(X)为 0。

3.4 本章小结

在本章中，首先针对僵尸网络在网络通信和网络攻击过程中所表现出来的一

些特征展开分析，发现僵尸网络在网络流量和主机行为上具有一定的相似性。根据相似性特征提出了基于相似性分析的僵尸网络检测模型，并对模型的各个模块的功能设计及相关算法进行了详细的说明。

第四章 检测模型的实现

在上一章中，我们对基于相似性分析的僵尸网络检测系统模型设计进行了介绍，整个检测系统模型分为 4 个部分：a)网络抓包模块 b)网络流量分析模块 c)主机行为分析模块 d)交叉关联模块。在本章中，我们将给出检测模型中各个模块的具体实现。

4.1 网络抓包模块的实现

在对网络流量检测时，我们通常会将网络流量监测模块部署在被监测局域网络的边界处，例如网关、路由器、防火墙等，能够监测、获取从内网发送出去或者流向内网的网络流量。

在本章我们采用 NDIS 中间层驱动技术实现网络数据包的获取，NDIS (Network Driver Interface Specification,网络驱动接口规范)是 Windows 提供的针对 NIC（Network Interface Cards,网络接口卡）制定的一种 API（Application Programming Interface）接口，定义了网卡驱动程序和协议驱动程序之间通信接口规范，使得底层物理网卡相对于协议驱动程序是透明的，协议驱动程序在不需改动的情况下就能够与不同类型的物理网卡进行通信。NDIS 中间层驱动位于 NDIS 协议驱动和 NDIS 小端口驱动之间，向上对协议驱动提供小端口驱动功能，向下对小端口驱动程序提供协议驱动功能，其具体框架如图 4-1 所示。

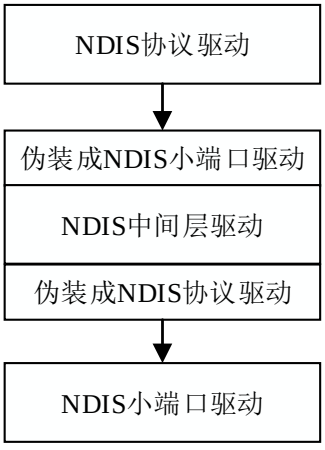

图 4-1 NDIS 中间层驱动框架图

在此模块的具体实现过程中，我们利用 NDIS 中间层驱动接收小端口驱动向上传递的网络数据包，对数据包进行分析，复制 DNS 数据包和 TCP 数据包并提交到

用户空间写入文件 Packets 中，整个模块的工作流程如图 4-2 所示。

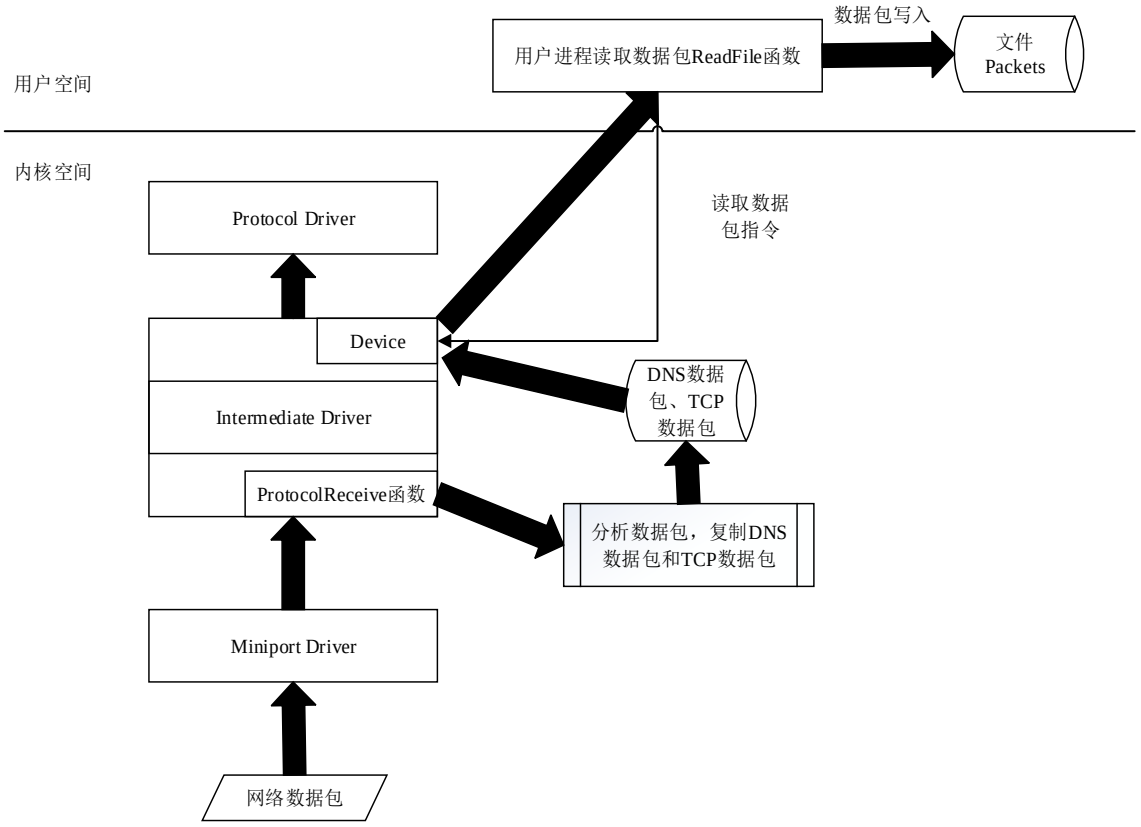

图 4-2 网络流量监测模块工作流程图

其中，对中间层驱动的 ProtocolReceive 函数修改，对小端口驱动传递上来的数据包进行分析，复制 DNS 数据包和 TCP 数据包并保存，同时将原数据包向上调用协议驱动程序接口函数。我们在内核 NDIS 中间层驱动中获取数据包过程的关键代码如表 4-1 所示，

表 4-1 NDIS 驱动抓包关键代码

```
1.  Packet =NdisGetReceivedPacket(pAdapt->BindingHandle,
                                  MacReceiveContext);
2.  if (Packet != NULL)
3.  {
4.     if(!Globals.DriverState)
5.        goto SkipCapturePacket;
6.     fStatus = AnalysisPacket(Packet, TRUE);
7.     if(fStatus == STATUS_ACCEPT)
8.     {
9.        DBGPRINT(("copy packet starting .....\n"));
10.       do{
11.            virtualAddress = SafeCopyNdisPacket(&newPacket, Packet,
    Globals.RcvPacketPool, Globals.RcvBufferPool, &size);
```

```
12.           if(virtualAddress)
13.           {
14.              DBGPRINT(("copy success, starting insert
                                  list....\n"));
15.              pEnt = MD_RCV_PKT_TO_LIST_ENTRY(newPacket);
16.              NdisAcquireSpinLock(&Globals.GlobalLock);
17.              InsertTailList(&Globals.RcvPackets,pEnt);
18.              DBGPRINT(("insert success....\n"));
19.              NdisReleaseSpinLock(&Globals.GlobalLock);
20.              DBGPRINT(("ending insert list....\n"));
21.              break;
22.            }
23.
24.       }while(TRUE);//end do
25.
26.       DBGPRINT(("copy packet end .....\n"));
27.       if(NULL ==Globals.notifyRecord)
28.       {
29.           DBGPRINT(("event not initialised!\n"));
30.       }else
31.       {
32.           if(!KeReadStateEvent(Globals.notifyRecord->Event))
33.       {
34.       KeSetEvent(Globals.notifyRecord->Event, 0, FALSE);
35.       DBGPRINT(("icmp event signaled!\n"));
36.     }
37.  }
38.  Status = NDIS_STATUS_SUCCESS;
```

当用户空间程序调用 Readfile 函数读取数据包时，中间层驱动的创建的设备将保存的数据包写入用户空间申请的内存地址中区。将数据包从内核空间传递到用户空间的关键代码如表 4-2 所示，

表 4-2 内核空间与用户空间通信关键代码

```
1. outDataLength = pIrpSp->Parameters.Read.Length;
2. DBGPRINT(("out buffer length is %ld:\n",outDataLength));
3. pDst= MmGetSystemAddressForMdlSafe(
            pIrp->MdlAddress, NormalPagePriority);
4. if(pDst==NULL)
5. {
6.    NtStatus = STATUS_INSUFFICIENT_RESOURCES;
7.    return NtStatus;
8. }

9. if(IsListEmpty(&Globals.RcvPackets))
10.{
11.   NtStatus = STATUS_SUCCESS;
12.   NdisMoveMemory(pDst,message,6);
13.   BytesReturned = 0;
14.   {
15.     NdisAcquireSpinLock(&Globals.GlobalLock);
16.     KeClearEvent(Globals.notifyRecord->Event);
17.     DBGPRINT(("clear event in kernel success"));
18.     NdisReleaseSpinLock(&Globals.GlobalLock);
19.   }
20.   goto CompleteTheIRP;
21.}
22.NdisAcquireSpinLock(&Globals.GlobalLock);
23.pRcvPacketEntry = Globals.RcvPackets.Flink;
24.RemoveEntryList(pRcvPacketEntry);
25.NdisReleaseSpinLock(&Globals.GlobalLock);
26.pRcvPacket = MD_LIST_ENTRY_TO_RCV_PKT(pRcvPacketEntry);
27.BytesRemaining = MmGetMdlByteCount(pIrp->MdlAddress);
28.pNdisBuffer = pRcvPacket->Private.Head;
29.while (BytesRemaining && (pNdisBuffer != NULL))
30.{
31.   #ifndef WIN9X
32.     NdisQueryBufferSafe(pNdisBuffer, &pSrc,
```

```
                  &BytesAvailable,   NormalPagePriority);
33.     if (pSrc == NULL)
34.     {
35.       break;
36.     }
37.   #else
38.     NdisQueryBuffer(pNdisBuffer, &pSrc, &BytesAvailable);
39.   #endif
40.   if (BytesAvailable)
41.   {
42.     ULONG  BytesToCopy = MIN(BytesAvailable, BytesRemaining);
43.

44.     NdisMoveMemory(pDst, pSrc, BytesToCopy);
45.     BytesRemaining -= BytesToCopy;
46.     pDst += BytesToCopy;
47.   }
48.   NdisGetNextBuffer(pNdisBuffer, &pNdisBuffer);
49. }
50. pIrp->IoStatus.Status = STATUS_SUCCESS;
51. pIrp->IoStatus.Information = MmGetMdlByteCount(pIrp->MdlAddress)
    - BytesRemaining;
52. IoCompleteRequest(pIrp, IO_NO_INCREMENT);
53. if ((pRcvPacket->Private.Pool) != Globals.RcvPacketPool)
54. {
55.   NdisReturnPackets(&pRcvPacket, 1);
56. }else
57. {
58.   ndisprotFreeReceivePacket(pRcvPacket);
59. }
```

4.2 网络流量分析模块的实现

4.2.1 网络流量记录模块的实现

网络流量记录模块针对网络流量监测模块中记录的网络数据包进行分析，首先区分出 DNS 数据包和 TCP 数据包，并采用不同的方法处理数据包得到聚类模块需要的数据信息。模块的具体实现方案流程图如图 4-3 所示。

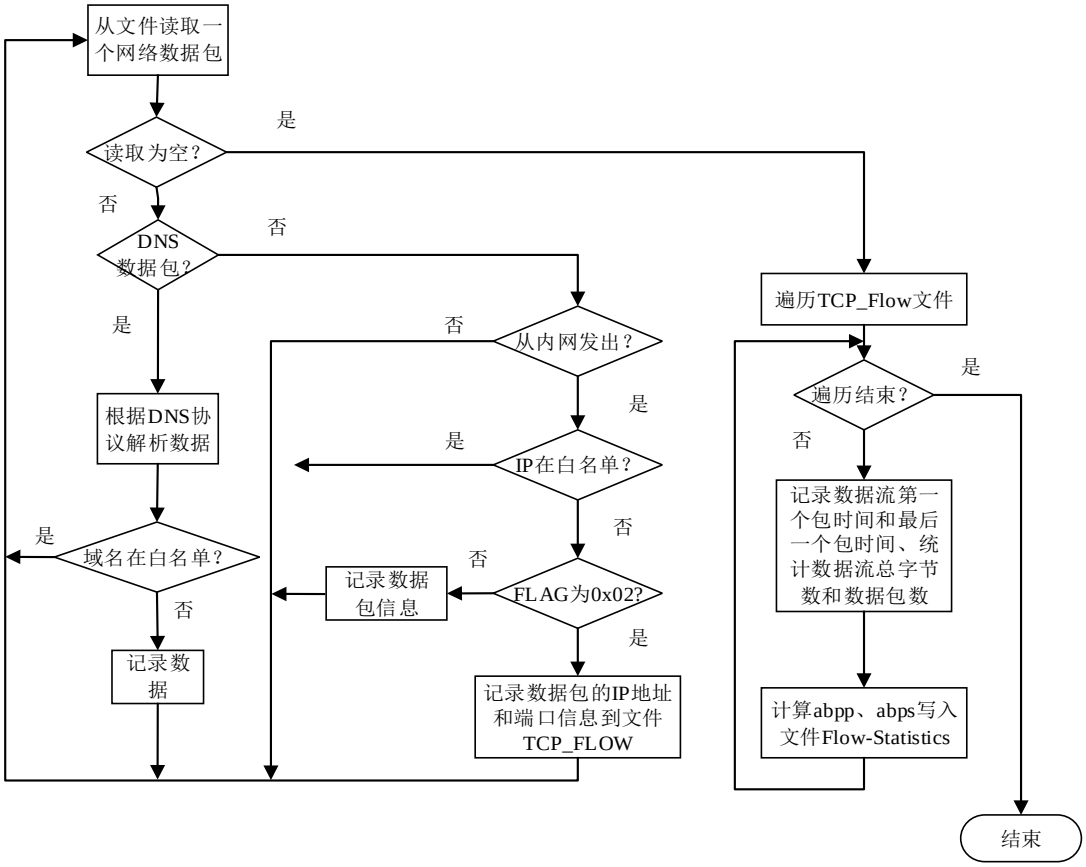

图 4-3 网络流量记录模块工作流程图

如图所示，网络流量记录模块首先是区分 DNS 数据包和 TCP 数据包，根据 DNS 数据包 IP 头部端口号为 53 来判断一个数据包是否为 DNS 数据包。解析 DNS 数据包提取出查询域名，然后将解析出的域名与白名单中合法域名进行匹配，如果在白名单中没有匹配到则将数据包信息保存在文件 DNS_Domain，其中 DNS_Domain 文件中信息保存格式是(源 IP 地址，查询域名，域名解析 IP 地址，域名访问时间)。DNS 域名解析具体代码如表 4-3 所示，

表 4-3 域名解析代码

```
1.  void ParseDNSName(UCHAR *chunk,UCHAR *ptr , char *name , int *len)
2.  {
3.     int n , alen , flag;
4.     char *position = name + (*len);
5.     for(;;)
6.     {
7.        flag = (int)ptr[0];
8.        if(flag == 0)
9.           break;

10.       if(IsPointer(flag))
11.       {
12.          n = (int)ptr[1];
13.          ptr = chunk + n;
14.          ParseDNSName (chunk , ptr , name , len);
15.          break;
16.       }else{
17.          ptr ++;
18.          memcpy(position , ptr , flag);
19.          position += flag;
20.          ptr += flag;
21.          *len += flag;
22.          if((int)ptr[0] != 0)
23.          {
24.             memcpy(positon , "." , 1);
25.             position += 1;
26.             (*len) += 1;
27.          }
28.       }
29.    }
30. }
```

由于我们在流量监测模块只获取 DNS 数据包和 TCP 数据包，如果一个数据包不是 DNS 数据包，那么它必是 TCP 数据包。针对 TCP 数据包我们需要找出不同的通信流，根据 TCP 协议在正常的数据流传输之前利用三次握手过程来建立连接的特性，我就以三次握手作为一个数据流的标识来提取出不同的数据流。TCP 数据流识别的步骤如下：

(1) 过滤网络内部通信的数据包和从外网发送到被监测网络内的数据包。通过判断数据包的源 IP 地址和目的 IP 地址是否为内网的 IP 地址来对数据报进行过滤，这里我们使用被监测网络的子网掩码分别对源 IP 和目的 IP 进行与操作，判断其结果是否等于子网号，仅仅保留源 IP 是内网地址同时目的 IP 不是被监测网络内网地址的数据包。

(2) 利用白名单对数据包进行二次过滤。白名单中保存合法的 IP 地址，利用正则表达式匹配的方法，将数据包的 IP 地址与白名单中合法 IP 地址进行匹配，过滤掉 IP 地址在白名单中的数据包

(3) 判断数据包 TCP 头部字段 FLAG 字段是否为 0x02。通过分析 TCP 头部各字段的意义可以知道，TCP 协议通过设置头部字段 FLAG 值为 0x02 来表示三次握手过程主动连接端发送的第一个 SYN 数据包，即一个 TCP 数据流的开始。

(4) 如果 FLAG 字段不是值 0x02，说明该数据包可能属于某一个数据流，将此数据包保存在文件 TCP_Packets，在具体实现过程中我们采用 16 进制的形式将数据包写入文件。否则，提出数据包 IP 头部的源 IP 地址、源端口、目的 IP 地址和目的端口保存在文件 TCP_Flow，文件中每一条数据流信息记录的格式为：源 IP 地址:源端口-目的 IP 地址:目的端口。

通过上述四步之后，我们不仅得到了 TCP 数据流信息，而且进一步过滤掉那些我们不感兴趣的数据包，如内网通信数据包、目的地址为被监测网络内网地址的数据包，大大降低数据流的规模，减轻了工作量。

在获取到网络数据流的信息之后，我们就需要计算每一个数据流的特征值 abpp 和 abps，特征值的计算我们通过遍历文件 TCP_Flow 和文件 TCP_Packets 实现的，具体过程是：

(1) 从文件 TCP_Flow 中读取一条数据流信息记录（格式为：源 IP 地址:源端口-目的 IP 地址:目的端口）；

(2) 遍历文件 TCP_Packets 与数据流信息（源 IP 地址:源端口-目的 IP 地址:目的端口）进行匹配，找到匹配的数据包。在遍历的过程中统计满足条件的数据包个数 packets_number、所有数据包总的大小 bytes_number，同时记录此数据流开始时间（即第一个数据包的时间）Flow_start_time 和数据流结束时间（即最后一个数据包的时间）Flow_end_time。

(3) 计算数据流的特征值 abpp、abps，计算方法如式(4-1)和式(4-2)所示。并将计算结果保存在文件 Flow_Statistics，其格式为：源 IP 地址:源端口 目的 IP 地址:目的端口 abpp abps。

$$abpp = bytes\_number \Big/ packets\_number \qquad \text{---公式(4-1)}$$

$$abps = bytes\_number \Big/ (Flow\_end\_time - Flow\_start\_time) \qquad \text{---公式(4-2)}$$

(4) 重复步骤(1)~(3)直到文件 TCP_Flow 遍历结束。

通过上述操作之后，我们就能得到 DNS 通信和 TCP 数据流的相关信息，接下来我们就可以利用这些信息进行聚类操作。

4.2.2 网络流量聚类模块的实现

在网络流量记录模块中，我们将获取的网络数据包通过分析分别将数据包相关信息记录在文件 DNS_Domain 和文件 Flow_Statistics 中，下面我们将分别对这两个文件保存的信息进行聚类操作得到 DNS 域名异常解析或网络流量相似的主机。

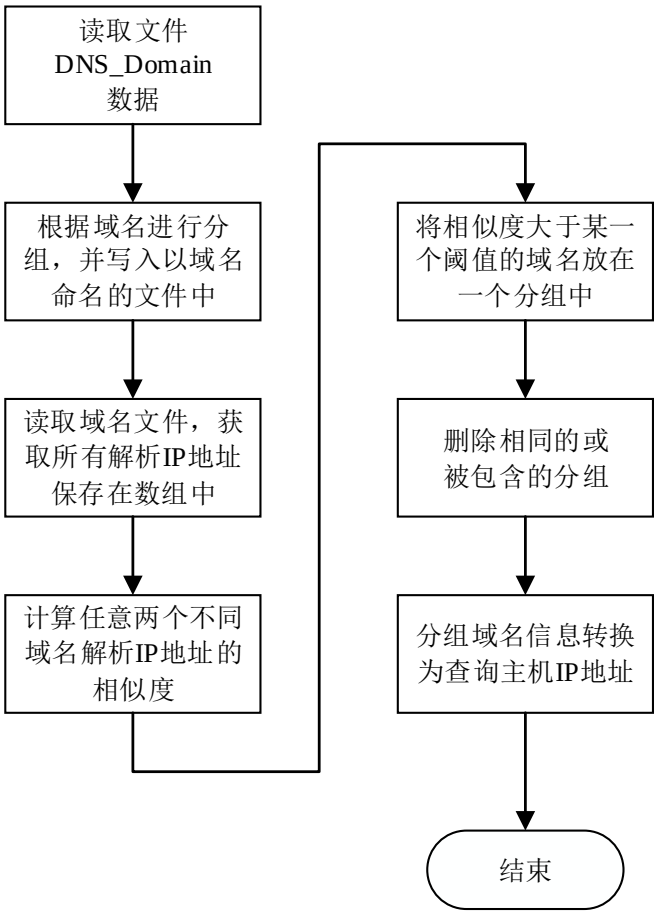

图 4-4 DNS 数据包聚类操作流程图

如图 4-4 所示为我们对 DNS 数据包进行聚类操作具体实现过程，我们首先遍历文件 DNS_Domain，根据解析域名将文件中数据进行分类，并写入以解析域名为文件名的文件中。然后读取每个域名文件信息，获取域名解析对应的 IP 地址保存在数组中，利用公式(3-1)计算任意两个不同域名的相似度，并将相似度大于某个阈值的两个域名放在同一个分组中，将分组结果写入文件。最后删除文件中相同的或者被包含的分组，再将过滤后的分组信息转换成发送相应域名查询的主机 IP 地址，保存在文件 Flow_Cluster。

然后，我们采用平均值聚类的方法对文件 Flow_Statistics 中的数据流信息分析，

同时根据僵尸网络网络流量的群体性，在聚类过程中过滤那些大小为 1 的结果集。在聚类因子的选择上我们也是在两次聚类过程中使用了不同的值，其中第一次聚类算法的聚类因子要大于第二次的聚类因子，这样在第一次聚类时减少聚类分组的个数、扩大聚类分组的范围，尽可能得使网络流量相似的主机在同一个聚类分组中。在第二次聚类采用值比较小的聚类因子对第一次聚类结果进行筛选，能够在一定程度上避免正常主机的网络流量的干扰，提高聚类结果的准确度，降低检测结果错误率。网络流量聚类模块具体实现流程如图 4-5 所示。

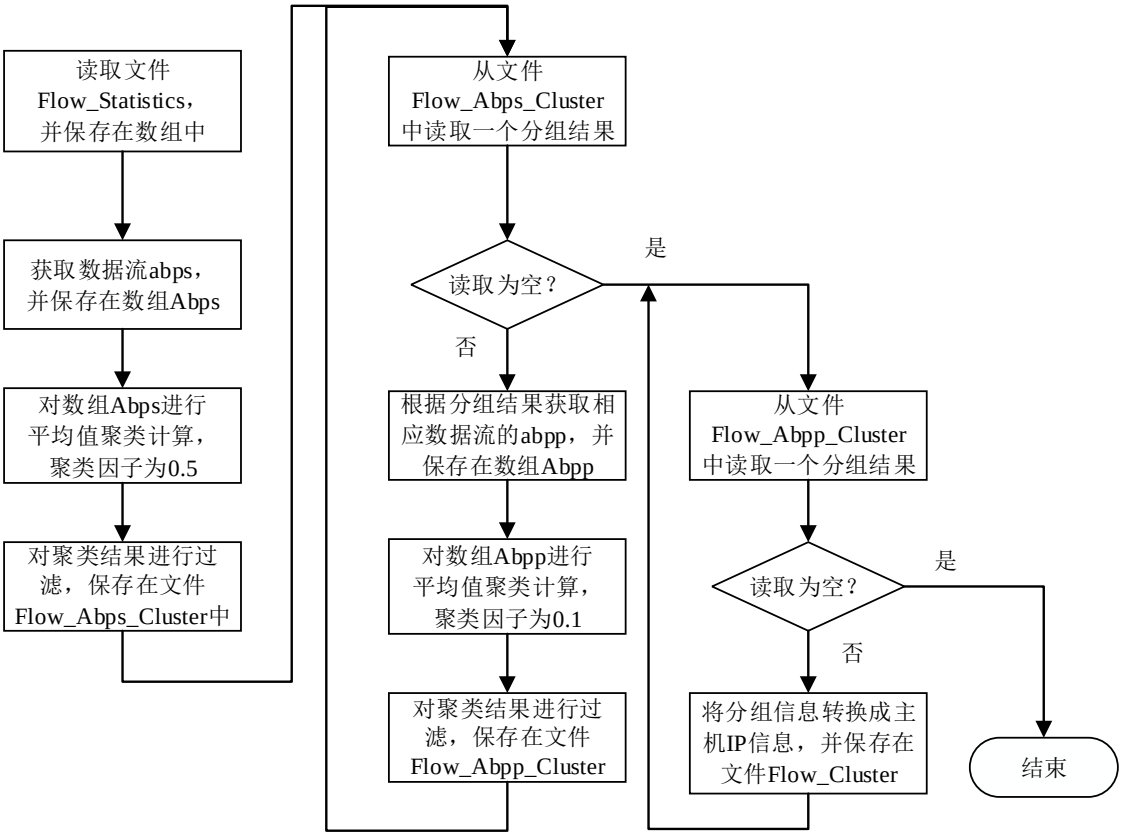

图 4-5 TCP 数据流聚类流程图

(1) 使用二维数组 Flow[][6]的形式保存从文件 Flow_Statistics 读取的数据。其中一位数组 Flow[]对应着文件 Flow_Statistics 中每一行数据流记录，Flow[][i],i=0,1,2,3,4,5.分别保存数据流记录中的源 IP 地址、源端口、目的 IP 地址、目的端口、abpp 和 abps。

(2) 选择聚类因子为 0.5，对数据流的 abps 进行平均值聚类计算。首先我们遍历二维数组 Flow 每一行，将值 Flow[i][5]保存在数组 Abps[i]中；然后在对数组 Abps 进行计算分组，将 abps 在数组 Abps 中的下标 i 作为标识在分组中记录；最后将分组中记录个数大于 1 的结果写入文件

Flow_Abps_Cluster，且每一行对应着一个分组结果。

(3) 从文件 Flow_Abps_Cluster 读取一行数据即一个分组结果，并根据读取的分组结果中下标 i 的值，将 Flow[i][5]值保存在数组 Abpp[i]中。选择聚类因子为 0.1，对数组 Abpp[i]进行平均值聚类计算，删除大小小于某个阈值的聚类分组，将其他分组写入文件 Flow_Abpp_Cluster。重复此操作直到读取文件 Flow_Abps_Cluster 结束。

(4) 进行 Flow_Abpp_Cluster 文件中分组结果的信息转换。将 Flow_Abpp_Cluster 文件中每一个分组结果，将分组中每一个下标 i 转换为对应的源 IP 地址信息，并将该分组信息保存在文件 Flow_Cluster，其中每一行代表一个分组主机 IP 信息。

4.3 主机行为分析模块的实现

4.3.1 主机行为记录模块的实现

主机行为记录模块是对获取的网络数据包展开分析找出具有恶意行为的主机，要从数据包中找出潜藏的恶意行为需要对恶意行为在网络流量中表现的特征有所了解，同时还需要对分析结果可能会产生干扰的网络技术有一定认识。

SYN FLOOD 是目前最流行、使用最广泛的一种 DDOS 攻击形式，通过发送大量虚假的 TCP 连接请求数据包给某一特定服务器，造成被攻击服务器资源耗尽（例如 CPU 满负荷运行、内存不足等）陷入瘫痪、无法继续提供服务。下面我们以利用 HTTP 协议进行 SYN Flood 攻击的 HTTP REQUEST FLOOD 为例来说明主机行为记录模块的工作原理和流程。

HTTP REQUEST FLOOD 通过模仿正常用户请求 Web 服务的过程，发送大量的 HTTP REQUEST 数据包给 Web 服务器，消耗服务器资源。因此我们可以总结出 HTTP REQUEST FLOOD 主机行为在网络流量中表现的特征有两点：a)发送 TCP 连接请求数据包请求与 Web 服务器端建立 TCP 连接； b)向 Web 服务器端发送 HTTP REQUEST 数据包且端口为 80 或 8080。在进行 HTTP REQUEST FLOOD 主机行为分析时，还需要考虑到目前访问量大的网站为保证网站的正常运行和访问所采用的 DNS 负载均衡技术。该技术在 DNS 服务器中给同一个域名绑定了多个 IP 地址（如图 4-6 所示为百度使用 DNS 负载均衡技术后的 DNS 解析结果），客户端可以通过不同的 IP 地址访问同一个域名的不同服务器，这也就导致仅仅通过 IP 地址来判断 HTTP REQUEST FLOOD 主机行为会产生很严重的遗漏。

```
C:\Users\Ryan.xi>nslookup www.baidu.com
Server:  dns.uestc.edu.cn
Address:  202.112.14.151

Non-authoritative answer:
Name:    www.a.shifen.com
Addresses:  119.75.217.56
          119.75.218.77
Aliases:  www.baidu.com
```

图 4-6 使用 DNS 负载均衡的解析结果

针对可能会出现这种情况的 HTTP REQUEST FLOOD 攻击，我们将源 IP 地址转换成其对应的注册域名，再进行分析判断。为提高效率，我们利用 IP 地址-域名映射文件机制（其中映射文件中保存常用的域名和其对应 IP 地址信息），将 IP 地址与映射文件记录中的 IP 地址项进行匹配查找 IP 值相同的记录，获取相应的域名值；当在映射文件中匹配不到时，我们利用 DNS 反向解析机制来查找绑定该 IP 地址的注册信息，将 IP 地址和域名作为一条记录写入映射文件，再利用 DNS 正向解析机制获取该域名绑定的所有的 IP 地址信息并写入映射文件。

DNS 反向解析是将 IP 地址映射到一个已注册域名地址，由于 DNS 域名服务器默认情况下是不提供域名反向解析服务的，在实验中我们通过网址 http://www.dnsstuff.com/tools 提供的 IPWHOIS LOOKUP 工具手动查询 IP 地址相对应的注册信息，如图 4-7 所示为查询 IP 地址 115.25.217.2 注册信息。

Contact Information

Registrant	Administrative Contact	Technical Contact
~{KQ:|9+K>~} Sohu Company CN Network Research Center, Beijing 100084, China	CER-AP CER-AP cernet-helpdesk-ip@net.edu.cn XL1-CN	CER-AP CER-AP SZ2-AP

图 4-7 IP 地址 115.25.217.2 反向解析结果图

主机行为记录模块对 HTTP REQUEST FLOOD 主机行为记录的具体实现过程如图 4-8 所示，

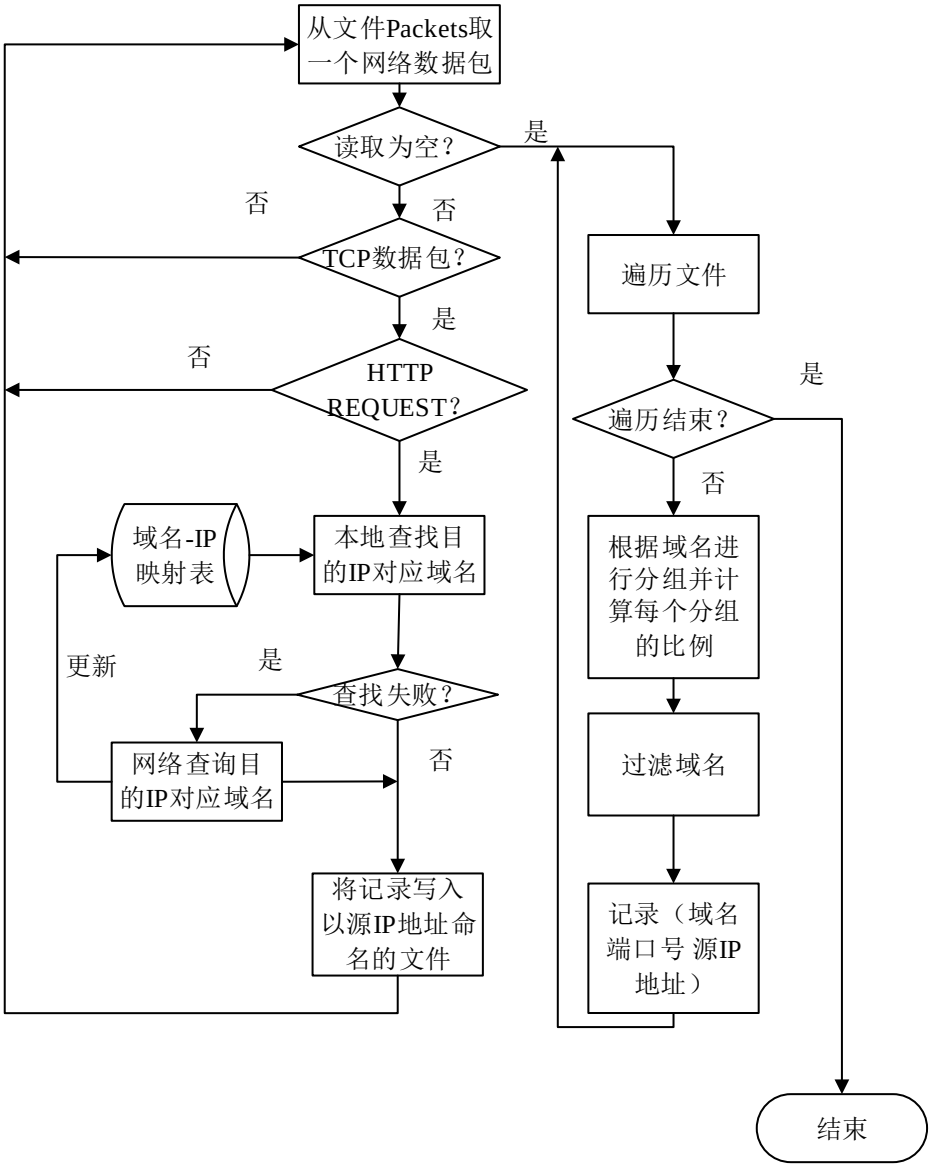

图 4-8 HTTP REQUEST FLOOD 主机行为记录流程图

(1) 从文件 Packets 中读取一个数据包信息，用字符指针指向该数据包内存地址；

(2) 若读取为空则执行(7)，反之执行(2)；

(3) 判断该数据包是否为 TCP 类型，不是则返回(1)；

(4) 判断数据包的 flag 字段是否为 0x02，并且目的端口值为 80 或者 8080，不是则返回(1)；

(5) 根据目的 IP 地址，从域名-IP 映射表中查找该 IP 对应的域名；

(6) 若查找成功则将（域名、目的端口）作为一条记录写入源 IP 地址命名的文件中。若查找失败，则从网络中查询目的 IP 地址对应的域名，将返回的

域名及原数据包端口写入文件，同时更新域名-IP 映射表。返回(1)

(7) 读取一个以 IP 命名的文件；

(8) 根据域名对读取的记录进行分组，然后计算每一个域名的比例；

(9) 删除比例小于某一个阈值的域名；

(10)把过滤后的每一个域名，以（域名 端口号 源 IP 地址）的形式写入文件 HTTP_REQUEST_FLOOD；

(11)若所有以 IP 命名文件遍历完则结束，否则返回(7)。

4.3.2 主机行为聚类模块的实现

在主机行为记录模块中我们已将可疑行为的数据包记录在文件中，接下来我们将对文件中记录的信息展开分析，找出具有相似行为的网络主机。主机行为聚类模块的具体实现过程如图 4-9 所示，

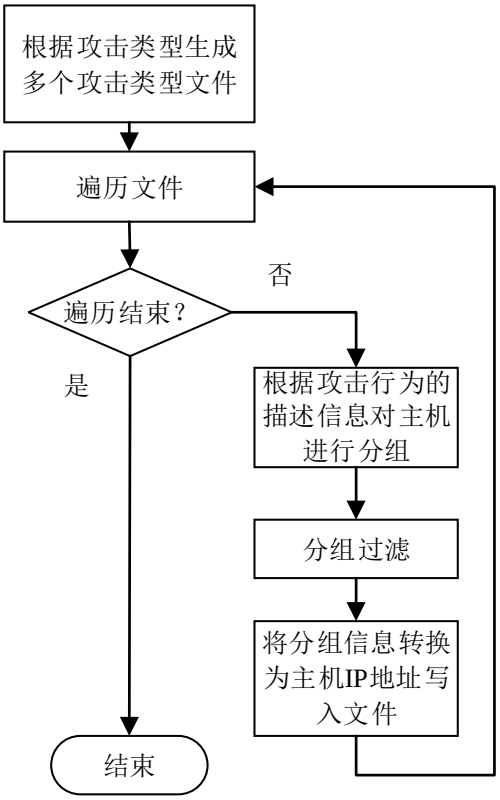

图 4-9 主机行为聚类模块流程图

如图 4-9 所示，首先主机行为聚类模块将记录的主机行为根据行为类型分组，并将每一种主机行为的信息保存在以该主机行为命名的文件中。其次，遍历所有以主机行为命名的文件，根据这种主机行为的信息进行二次分组，具有相同攻击信息的主机被分在一组。然后过滤掉该攻击类型分组中不满足条件的分组结果。

最后将过滤后的分组信息转换成相应的主机 IP 地址信息写入文件 Attack_Cluster，文件每一行表示一个分组的主机 IP 信息。由于 Botmaster 进行网络攻击时会向僵尸网络中许多 Bot 主机发送命令、表现出相似的主机行为，因此为避免网络内部个别主机的非正常行为对聚类结果的干扰，对二次聚类结果中具有相似行为主机数目小于一定阈值的结果集过滤掉。

4.4 交叉关联模块的实现

通过网络流量聚类模块和主机行为聚类模块的聚类分组操作之后将具有相似行为主机分在了一个集合，并将结果分别保存在文件 Flow_Cluster 和 Attack_Cluster。在交叉关联模块中我们将对这两个文件进行交叉处理，得到属于同一个僵尸网络的一组或几组主机列表。

首先，读取文件 Flow_Cluster 和 Attack_Cluster 中数据分别保存在二维数组 FlowCluster、AttackCluster，其中二维数组的每一行表示一个分组，如果一台主机不在该分组中，那么行号和列号对应的值为 0，反之为 1。

然后，遍历数组 AttackCluster 获取至少表现一种主机行为的主机保存在数组 AttackHost,针对数组 AttackHost 中任意 i 代表的主机 h，分别遍历二维数组 FlowCluster、AttackCluster 获取包含主机 h 的分组，并将该分组所在二维数组的行号 j 记录在数组 HostFlowCluster 和 HostAttackCluster，根据式 3-2 计算 s(h)。过滤值小于 0.5 的主机，最后将筛选后的主机保存在数组 HostFilter 中。

其次，根据数组 HostFilter 对数组 FlowCluster、AttackCluster 中不在 HostFilter 的相应位置进行置零。然后利用公式 3-3，遍历数组 HostFilter 中主机计算其与数组中其他主机的相似度，将满足条件的主机分在一个组中，并将分组的结果保存在文件 result 中，其中每一行代表一个分组结果。

最后，删除文件中相同的分组或者在其他分组中重复出现的分组结果，过滤之后的分组即为检测结果。

4.5 本章小结

本章针对第三章提出的检测系统模型的实现过程做出了详细说明，根据系统的工作流程，对各个功能模块的实现方法依次进行介绍，并给出了检测实例对系统的检测功能进行了更清晰地展示。

第五章 检测模型测试与分析

本章将对检测系统的检测功能进行实验验证和分析，检测将分别从实验环境部署、实验测试方案以及结果分析三个方面对实验过程进行介绍，最后提出了在互联网使用中，远离僵尸网络的预防措施。

5.1 实验环境部署

5.1.1 实验环境拓扑

为了避免实验测试过程中对正常网络造成不良影响，实验搭建了测试模拟环境，利用 7 台机器搭建了两个局域网（Local1 和 Local2），其中局域网 Local1 内使用虚拟机软件创建了六个虚拟实验主机，其中一个虚拟实验主机用作 IRC 服务器主机、一个虚拟实验主机用作僵尸网络控制主机、三个虚拟实验主机用于搭建三个 Web 服务器、一个虚拟实验主机配置 DNS 域名解析服务器；在两个局域网之间使用一台主机配置双网卡连接两个局域网，进行两个局域网之间网络通信数据包转发；局域网 Local2 作为被监测网络，内有 4 台机器且在每台机器上运行三个虚拟机系统，即有 12 个虚拟主机。实验拓扑结构如图 5-1 所示，

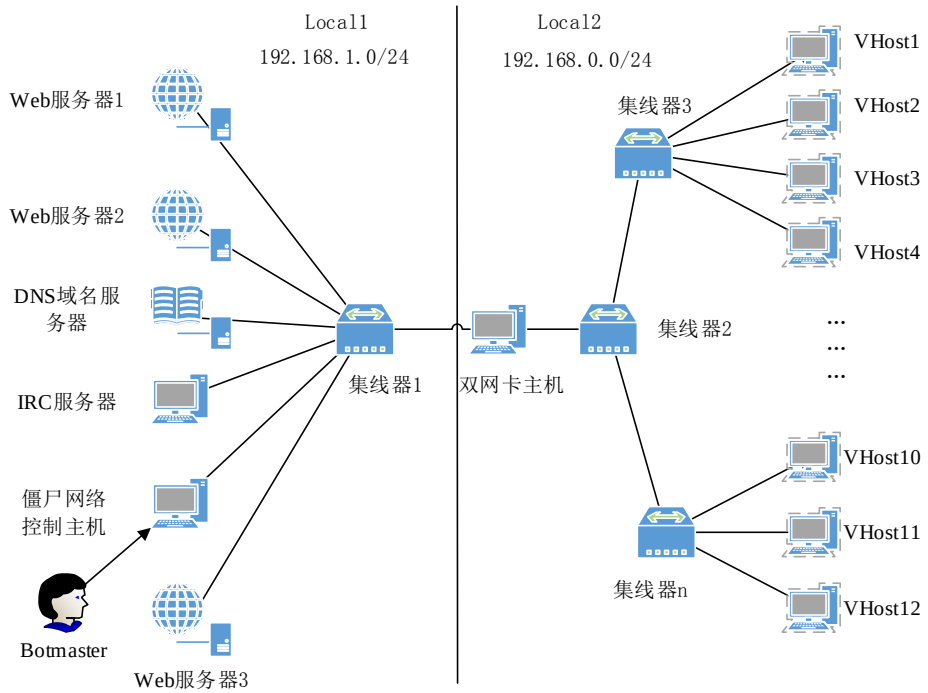

图 5-1 实验环境拓扑图

实验测试使用的主机配置参数如表 5-1 所示，

表 5-1 实验机器配置参数表

配置类别	具体参数
CPU	Intel Core(TM) i5-3210M 2.5GHz
主板	技嘉 B75M-D3V
内存	4G
硬盘	500G
显示器	LG 24EN33

实验主机的具体 IP 分配和主机说明等信息如表 5-3 所示，

表 5-3 实验主机详细信息表

局域网名称	主机名称	网络 IP 地址	默认网关	说明
Local1 (192.168.1.0/24)	Web 服务器 1	192．168.1.100	192.168.1.1	提供 Web 服务
	Web 服务器 2	192．168.1.101	192.168.1.1	提供 Web 服务
	Web 服务器 2	192．168.1.101	192.168.1.1	提供 Web 服务
	DNS 服务器	192．168.1.103	192.168.1.1	提供域名查询
	IRC 服务器	192．168.1.104	192.168.1.1	僵尸网络服务器
	控制主机	192．168.1.105	192.168.1.1	Botmaster
	Route_Host	192.168.1.1	192.168.1.1	网卡 1IP 地址
Local2 (192.168.0.0/24)		192.168.0.1		网卡 2IP 地址
	VH1	192.168.0.101	192.168.0.1	实验测试被监测网络主机
	VH2	192.168.0.102	192.168.0.1	
	VH3	192.168.0.103	192.168.0.1	
	VH4	192.168.0.104	192.168.0.1	
	VH5	192.168.0.105	192.168.0.1	
	VH6	192.168.0.106	192.168.0.1	
	VH7	192.168.0.107	192.168.0.1	
	VH8	192.168.0.108	192.168.0.1	
	VH9	192.168.0.109	192.168.0.1	
	VH10	192.168.0.110	192.168.0.1	
	VH11	192.168.0.111	192.168.0.1	
	VH12	192.168.0.112	192.168.0.1	

5.1.2 实验环境配置

搭建实验环境用到的软件类型及版本如表 5-2 所示，

表 5-2 实验软件型号列表

软件类型	具体参数
虚拟机软件	VMware Workstation 9.0
操作系统 1	Windows Server 2003(32 位)
操作系统 2	Windows XP sp3(32 位)
操作系统 3	Ubuntu 8.04
编译工具	Visual Studio 2008

(1) web 服务器搭建

WEB 服务器使用 Windows Server 2003 操作系统自带的 WEB 服务，在 Windows

Server 2003 上面创建一个网站的具体过程如下，

a) 首先打开系统中 Internet 信息服务（IIS），
b) 然后在 Internet 信息服务管理器中将“Web 服务扩展”中的 WebDAV 的“状况”属性值更改为“允许”，
c) 最后在 Internet 信息服务管理器中“网站”选项中选择“新建网站”，并在创建向导中设置网站的 IP 地址、网站 TCP 端口号、网站的主机头和网站主目录，这样一个网站即可创建完成。

在实验中创建网站的具体描述如表 5-3 所示，

表 5-3 实验新建网站信息表

机器名	网站 IP 地址	网站 TCP 端口	网站域名
Web 服务器 1	192.168.1.100	80	Web1. BotTest.com
Web 服务器 2	192.168.1.101	8080	Web2. BotTest.com
Web 服务器 3	192.168.1.102	80	Web1. BotTest.com

(2) DNS 域名服务器搭建

由于 Windows Server 2003 操作系统自带了 DNS 域名服务，因此只需要打开系统 DNS 服务并进行配置即可。为保证局域网中主机能够通过该域名查询到 Web 服务器的 IP 地址，我们在 DNS 域名服务器中对创建的网站域名进行注册，同时为模仿简单的负载均衡技术，我们在 DNS 域名服务器中为域名“Web1. BotTest.com”绑定两个 IP 地址（即 Web 服务器 1 和 Web 服务器 3 的 IP 地址）。整个 DNS 域名服务器搭建具体过程如下，

a) 打开“管理工具”中的“配置您的服务器向导”，进入“服务器角色”向导页，
b) 选择“DNS 服务器”进行安装 DNS 服务器并运行配置 DNS 服务器向导配置 DNS，并利用静态 IP 地址进行 DNS 服务器配置。
c) 给整个实验环境创建一个区域，区域名称为“BotTest.com”，并设置“这台服务器维护该区域”，
d) 在区域创建完成之后，即可为 Web 服务器创建域名。在 DNS 控制台窗口的“BotTest.com”选项执行“新建主机”命令，

e) 在“新建主机对话框”的“名称”选项栏中填写“Web1”，并在“IP 地址”栏中填写 Web 服务器 1 的主机 IP 地址，域名“Web1.BotTest.com”创建完成。

f) 重复执行步骤 d)和 f)，创建 Web 服务器 2 上的域名 “Web2.BotTest.com”，

g) 给域名“Web1.BotTest.com”添加绑定 Web 服务器 3 的 IP 地址（192.168.1.102）。

通过以上过程，我们就可以通过域名来访问 Web 服务器。

(3) 局域网 Local2 搭建

在实验中，为降低资源开销同时保证一定数量的主机数，我们利用 VMware WorkStation 软件在每台机器上搭建了两个 Windows XP 虚拟机，共 18 个虚拟系统作为实验主机。为保证所有实验主机的 IP 地址为同一个子网，我们在 VMware 中将实验主机的网卡模式设置为桥接模式，并给每个系统手动配置 IP 地址、网关 IP 地址和默认 DNS IP 地址，如图 5-2 所示。

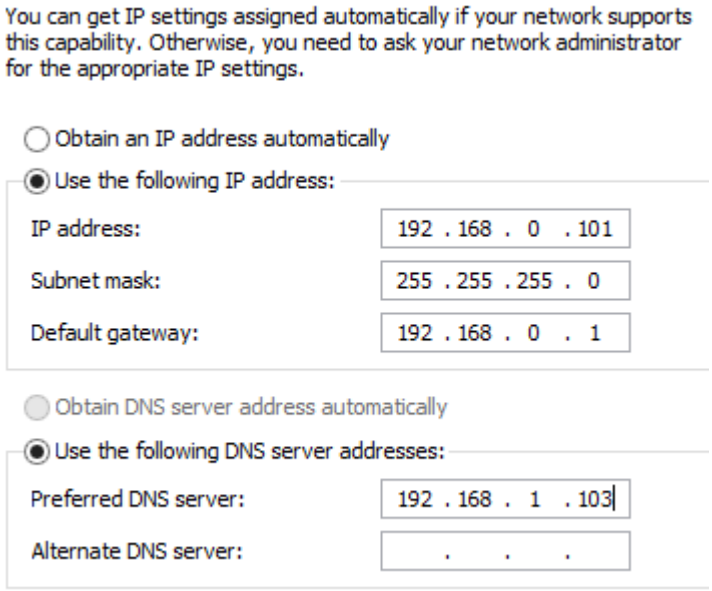

图 5-2 虚拟主机网络配置图

(4) 双网卡机器配置

在实验环境中利用一台双网卡机器充当路由功能部署在局域网 Local1 和 Local2 中间，通过配置路由表信息即可使两个局域网进行通信，在该主机上运行我们的检测系统就能够检测到所有进出局域网 Local1 的网络流量信息。在手动配置完两个网卡的网络地址后，我们利用 route 命令更改路由表信息，具体修改命令如下，

a) 删除默认设置：route delete 0.0.0.0

b) 设置默认网关为网卡 1 的 IP 地址：

route add 0.0.0.0 mask 0.0.0.0 192.168.1.1 –p

c) 添加局域网 Local2 路由信息：

route add 192.168.0.0 mask 255.255.255.0 192.168.0.1 -p

d) 添加局域网 Local2 路由信息：

route add 192.168.1.0 mask 255.255.255.0 192.168.1.1 –p

为使数据包能够在两个局域网之间传递，我们还需要打开本台主机的路由转发功能，操作方法是：打开注册表，找到注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters，选择 IPEnableRouter 修改 REG_DWORD 的值为 0x1。

5.1.3 僵尸网络搭建

在实验测试环境的网络地址信息配置完成并且 DNS 域名服务器和 Web 服务器架设成功之后，我们要在该环境中搭建一个僵尸网络来对检测系统进行测试。由于实验环境是在一个局域网中且与外部网络隔离，在搭建僵尸网络时需要对僵尸网络程序要能够对配置文件修改，僵尸程序 SDbot 不仅能够互联网上下载到其源码，而且可以修改其配置文件重新编译以适应实验环境。由于 SDbot 是一种 IRC 僵尸网络程序，所以我们还要在局域网中架设一台 IRC 服务器来完成整个僵尸网络的创建。

a) IRC 服务器端

在实验中，我们使用开源的 IRC 服务器程序 UnrealIRCd 在 Linux 操作系统 Ubuntu 12.04 版本上搭建 IRC 服务器，修改 UnreadIRCd 的配置文件 unrealircd.conf 可创建监听信道，运行命令./unreal start 启动服务器。

b) SDbot 客户端

由于互联网上提供了 SDbot 僵尸程序源代码，所以我们直接下载了源代码，修改配置信息并重新编译即可生成我们需要的僵尸程序。如图 5-3 所示为 SDbot 程序配置信息，

```
// bot configuration
 const char botid[] = "BotTest"; // bot id
 const char password[] = "123456"; // bot password
 const  int maxlogins = 20; // maximum number of simultaneous logins
 const char server[] = "192.168.1.104"; // server
 const  int port = 6667; // server port
 const char serverpass[] = "Bot123456"; // server password
 const char channel[] = "BotTest1"; // channel that the bot should join
 const char chanpass[] = "BotTest1"; // channel password
 const char server2[] = ""; // backup server (optional)
 const  int port2 = 6667; // backup server port
 const char channel2[] = ""; // backup channel (optional)
 const char chanpass2[] = ""; // backup channel password (optional)
 const BOOL topiccmd = FALSE; // set to TRUE to enable topic commands
 const BOOL rndfilename = FALSE; // use random file name
 const char filename[] = "BotTest.exe"; // destination file name
 const BOOL regrun = TRUE; // use the Run registry key for autostart
 const BOOL regrunservices = TRUE; // use the RunServices registry key for autostart
 const char valuename[] = "Configuration Loader"; // value name for autostart
 const char prefix = '.'; // command prefix (one character max.)
 const char version[] = "sdbot v0.5b by [sd]"; // bot's VERSION reply
 const  int cryptkey = 0; // encryption key (not used right now)
 const  int maxaliases = 16; // maximum number of aliases (must be greater than the number of predefined aliases).
```

图 5-3 SDbot 程序配置图

其中，password 表示获取僵尸网络控制权的密码，server 表明运行 IRC 服务器主机的 IP 地址，port 表示 IRC 服务器监听的端口，serverpass 表示僵尸程序连接服务器时需要提供的连接口令，channel 和 chanpass 表示僵尸网络创建的一个信道和该信息密码，filename 表示该源码编译连接之后生成的目标程序的名称。在被感染主机运行双击，程序就会根据配置信息自动加入僵尸网络信道，等待接收 Botmaster 发送的指令。

在实验过程中我们常用的 SDbot 命令如表 5-4 所示，

表 5-4 实验中用到的 Sdbot 命令列表

命令	参数	命令说明
.login	口令	申请获取僵尸网络控制权
.logout	无	退出僵尸网络
.repeat	<次数> <命令>	将命令重复执行一定次数

5.2 实验场景

为对僵尸网络检测系统功能、检测结果准确度等方面进行检测，本小节设计了不同的实验场景来测试。

5.2.1 实验场景一

在实验场景一中，主要对检测系统的基本功能模块进行测试，在实验主机 VH1~VH5 运行僵尸网络 Bot1（信道为 BotTest1），并对 Web 服务器 1 发动 HTTP REQUEST FLOOD 攻击。实验具体步骤如下，

(1) 首先，我们在检测系统的 IP 白名单中添加合法 IP 地址，例如 Web 服务器 1 的 IP 地址（192.168.1.100）、Web 服务器 3 的 IP 地址（192.168.1.102）、双网卡机器两个网卡的 IP 地址（192.168.1.1、192.168.0.1）以及 DNS 域名服务器 IP 地址（192.168.1.103）；在 DNS 白名单中添加域名“Web1.BotTest.com”信息；在域名-IP 映射表中添加信息“Web1.BotTest.com 192.168.1.100”、“Web1.BotTest.com 192.168.1.102”

(2) 其次，在控制主机运行一个 SDbot 僵尸网络程序并加入信道“BotTest1”，僵尸网络主机 VH1~VH5 运行程序加入僵尸网络，僵尸网络控制主机的 SDbot 程序执行.login 指令申请获取僵尸网络 Bot1 控制权。在主机 VH6 上安装正常 IRC 客户端程序，并加入僵尸网络 Bot1 的信道 BotTest1。

(3) 然后，在双网卡机器上进行连续两个小时的网络流量监测及抓包，在这段时间内，使用命令.repeat 控制僵尸网络 Bot1 中主机对网站“Web1.BotTest.com”发动 HTTP REQUEST FLOOD 攻击。

(4) 最后，对检测系统的检测结果进行分析、总结。

5.2.2 实验场景二

在实验场景一中，仅仅实现了对检测系统基本功能的简单测试，且僵尸网络工作比较简单，因此在方案二中使用比较广泛的技术---利用 DNS 查询机制漏洞与控制服务器进行连接，模拟目前的僵尸网络，例如 Fast Flux、Domain Flux 等实现机制，对检测系统的 DNS 异常网络流量检测功能进行测试。

在实验主机 VH1~VH5 和 VH8~VH12 分别运行僵尸网络 Bot1（信道为 BotTest1）和僵尸网络 Bot2（信道为 BotTest2），同时为增加对主机行为检测的干扰，我们利用实验主机 VH6、VH7 频繁的访问网站“Web1.BotTest.com”和“Web2.BotTest.com”。实验具体步骤如下，

(1) 配置检测系统多个白名单信息，例如 IP 地址白名单、DNS 域名白名单。在域名-IP 映射表中添加网站“Web1.BotTest.com”和“Web2.BotTest.com”对应的 IP 地址信息。

(2) 利用域名生成算法生成一组域名并保存在文件 Domain 中，并将文件 Domain 拷贝到实验主机 VH 在域名服务器中将该组域名绑定到 IRC 服务

器的 IP 地址上。

(3) 在控制主机运行两个僵尸网络程序并分别加入信道“BotTest1”和“BotTest2”，僵尸网络主机 VH1~VH5 和 VH8~VH12 分别运行程序加入僵尸网络，控制主机的两个僵尸程序分别执行命令.login 12345 获取各自僵尸网络控制权。

(4) 在双网卡机器上进行连续两个小时的网络流量监测及抓包，在此时间段内，实验主机分别进行如下几种操作：a) 僵尸网络 Bot1 和 Bot2 在不同时刻向网站“Web1.BotTest.com” 和“Web2.BotTest.com”发动 HTTP REQUESET FLOOD。b) 主机 VH7 和 VH9 分别频繁的访问网站“Web1.BotTest.com” 和“Web2.BotTest.com”。c) 僵尸网络 Bot1 的主机运行 DNS 查询程序随机查询文件 Domain 中的域名。

(5) 对检测系统的检测结果进行分析、总结。

5.3 实验结果及分析

在上一小节中，设计了两种不同的实验场景对检测系统进行测试，在本小节中主要对不同场景下的实验过程和结果进行分析。

5.3.1 实验一的结果分析

将网络抓包模块将从底层接收到数据包信息以字符串的形式保存在文件 Packets-1。由于整个过程中实验主机 VH1~VH5 查询的域名为“Web1.BotTest.com”，在进行解析 IP 地址相似性计算得到结果为 0，因此网络流量记录模块和网络流量聚类模块对 DNS 数据包分析得到的结果为空。网络流量记录模块对 TCP 数据包分析得出的数据流记录信息如图 5-4 所示。

```
#192.168.0.103:1053-192.168.1.103:6667#
#192.168.0.101:1058-192.168.1.103:6667#
#192.168.0.102:1049-192.168.1.103:6667#
#192.168.0.105:1050-192.168.1.103:6667#
#192.168.0.104:1042-192.168.1.103:6667#
```

图 5-4 实验一 TCP 数据流记录图

通过对 TCP 数据流 abps 和 abpp 特征计算，并通过二次平均值聚类得到一组主机序列得到分组结果如图 5-5 所示，

```
[1].%192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104-192.168.0.106%
```

图 5-5 实验一网络流量聚类结果

主机行为记录模块对文件 Packets 进行分析得到主机行为 HTTP REQUEST FLOOD 文件如图 5-6 所示：

```
#Web1.BotTest.com 80 192.168.1.103#
#Web1.BotTest.com 80 192.168.0.101#
#Web1.BotTest.com 80 192.168.0.102#
#Web1.BotTest.com 80 192.168.0.105#
#Web1.BotTest.com 80 192.168.0.104#
#Web2.BotTest.com 8080 192.168.0.110#
#Web2.BotTest.com 8080 192.168.0.108#
#Web2.BotTest.com 8080 192.168.0.111#
#Web2.BotTest.com 8080 192.168.0.109#
#Web2.BotTest.com 8080 192.168.0.112#
```

图 5-6 实验一主机行为 HTTP REQUEST FLOOD 记录文件

然后根据域名和端口号对记录文件聚类得到一个分组结果如图 5-7 所示：

```
[1].%192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104%
```

图 5-7 实验一主机行为聚类结果

对网络流量聚类将结果和主机行为聚类结果进行交叉关联分析得到一个分组结果，如图 5-8 所示：

```
[1].$192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104$
```

图 5-8 实验一检测结果

从图 5-8 可看出检测系统检测到主机主机 VH1~VH5 属于同一个僵尸网络，与实验预期结果一致。

在整个实验过程中只有 Bot 主机发出网络流量和进行 HTTP REQUEST FLOOD 攻击相关的网络数据，没有任何其他正常主机的干扰，最终得出的结果也是与实际情况相符合的。根据实验分析我们可以发现，通过网络流量分析得到 Bot 主机结果未必准确（主机 VH6 为正常主机，与 Bot1 主机具有相似的网络数据流特征），通过与主机行为分析结果进行交叉关联可以增加结果的准确性。

当然，实验一的环境是理想化的，还需要在更加复杂、干扰更多的环境中进行测试。

5.3.2 实验二的结果分析

将网络抓包模块将从底层接收到数据包信息以字符串的形式保存在文件 Packets-2。网络流量记录模块和网络流量聚类模块对 DNS 数据包分析得到的结果为一个分组如图 5-9 所示：

```
[1].%192.168.0.110-192.168.0.108-192.168.0.111-192.168.0.109-192.168.0.112%
```

图 5-9 实验二 DNS 数据包检测结果

对 TCP 数据流特征值 abps 和 abpp 计算、二次聚类得到两个分组，如图 5-10 所示：

```
#192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104#
#192.168.0.110-192.168.0.108-192.168.0.111-192.168.0.109-192.168.0.112#
```

图 5-10 实验二 TCP 数据流分析结果

对 DNS 数据包分析和 TCP 数据流聚类结果进行过滤结果如图 5-11 所示：

```
[1].%192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104%
[2].%192.168.0.110-192.168.0.108-192.168.0.111-192.168.0.109-192.168.0.112%
```

图 5-11 实验二网络流量聚类结果

主机行为记录模块对文件 Packets 进行分析得到主机行为 HTTP REQUEST FLOOD 文件，根据域名和端口号对记录文件聚类得到两个分组结果，如图 5-12 所示：

```
[1].%192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104-192.168.0.107%
[2].%192.168.0.110-192.168.0.108-192.168.0.111-192.168.0.109-192.168.0.112-192.168.0.106%
```

图 5-12 实验二主机行为聚类集过

对网络流量聚类将结果和主机行为聚类结果进行交叉关联分析得到两个分组结果，如图 5-13 所示：

```
[1].$192.168.0.103-192.168.0.101-192.168.0.102-192.168.0.105-192.168.0.104$
[2].$192.168.0.110-192.168.0.108-192.168.0.111-192.168.0.109-192.168.0.112$
```

图 5-13 实验二检测结果

从图 5-13 可看出检测系统检测到主机 VH1~VH5 属于同一个僵尸网络，主机 VH8~VH12 属于同一个僵尸网络，与实验预期结果一致。

在实验二中，主机 VH7、VH6 虽然分别与僵尸网络 Bot1 和 Bot2 表现出相似的主机行为，但网络流量不具有相似性，因此被过滤掉，这个实验结果也说明主机行为和网络流量具有相似性是检测判断一个网络中是否存在或者一台主机是否为 Bot 主机的必要条件。

在上述两个实验中我们简单地对检测系统的基本功能进行测试，证明了我们提出的检测系统是可行的。但是这两个实验模拟的网络环境比较简单，网络数据流量、数据包类型、TCP 通信连接时间、网络复杂情况等都无法跟实际互联网络环境相比较，因此仍然需要在复杂的环境中对检测系统进行更进一步的测试。

5.4 僵尸网络预防措施

打击互联网上的僵尸网络不能仅仅依靠网络安全人员的努力，互联网用户也应该在日常的上网过程中养成良好的上网习惯、培养自己的网络安全意识，避免被僵尸程序感染，成为 Bot 主机被网络不法分子控制利用。下面简单介绍几种预防僵尸网络的措施，

(1) 在使用计算机时，应尽量避免使用计算机管理员的身份登录，严格控制计算机用户的权限。

(2) 对于匿名网络邮件应该谨慎打开或点击下载其中的附件。

(3) 不上陌生非法的网站，禁用浏览器中一些高危险插件、脚本程序。

(4) 开启计算机中的病毒防火墙和网络防火墙，及时对病毒库进行更新，定期对计算机检查。

5.5 本章小结

在本章的工作中，首先对系统的实验环境进行介绍，分别从拓扑结构、环境搭建及系统配置三个方面展开说明，然后设计了不同层次的实验场景、实现了对系统由简单到复杂的环境检测测试验证。实验结果显示，该检测系统能够有效的检测出局域网内的僵尸主机。

第六章 总结与展望

僵尸网络因其覆盖面广、难检测、可控制、破坏力大等特点被网络犯罪分子广泛用于从事网络恶意行为、恶意攻击等犯罪行为的网络平台，其数量、规模也在不断地增长，并利用各种技术提高网络的隐蔽性以规避目前安全检测系统的检测。针对目前已有检测方法存在的一些缺陷，本文将僵尸网络具有的基本特征作为突破口，力求能够找到一种独立于僵尸网络种类的僵尸网络检测方案。

6.1 工作总结

目前僵尸网络已经成为网络安全领域的一个研究热点，许多专家、学者也已经在僵尸网络技术、僵尸网络检测技术方面进行了大量的研究分析，为僵尸网络检测技术的不断创新打下了坚实的理论基础。本文在已有的僵尸网络检测技术理论基础之上，根据僵尸网络网络流量和主机行为相似性的特点，提出了基于相似性分析的僵尸网络检测方法。

首先本文从僵尸网络概念、恶意行为类型、C&C 通信协议、C&C 体系架构等方面对僵尸网络进行全面介绍，根据这些知识得出僵尸网络相似性的特点，并依此为依据提出了基于相似性分析的僵尸网络检测方法。此方法的检测原理就是通过对被监测网络的网络数据进行分析，找出在此时段表现出相似网络流量和网络主机行为的主机。

在检测系统具体实现过程中，收集到网络数据包之后，并行执行如下两个过程：a)对数据包分析，获取通信记录并进行聚类操作；b)分析数据包，获取可以主机行为记录，聚类分组相似的主机。在通信聚类过程中，我们分别对 DNS 数据包和 TCP 数据流进行处理，根据在不同时间段内 DNS 域名解析与解析 IP 地址之间的关系，找出异常的 DNS 通信，并以解析 IP 地址为单位对主机分组。在处理 TCP 数据流时，首先对 TCP 数据包归类，计算数据流的特征值 abpp 和 abps，然后利用平局值聚类算法依次对 abps 和 abpp 进行聚类。在对主机进行主机行为相似聚类时，首先将主机根据主机行为类型进行分组，然后利用主机行为信息对前一步分组结果进行二次聚类分组。在上述两个过程执行完之后，对它们产生的结果进行交叉关联操作，找出具有相似网络流量和主机行为的一组或机组主机。

最后在完成检测系统模型具体实现之后，文章搭建了一个实验环境、设计不同的实验方案对检测系统的可行性进行验证，并给出一些预防僵尸网络感染的防护性措施。

6.2 下一步工作

虽然基于检测系统能够识别中被监测网络中不同类型的僵尸网络主机，但是由于受到时间、实验环境、僵尸网络种类等因素的限制，本次工作中仍然几点不足之处，主要表现在如下几点：

(1) 检测系统的验证过程是在我们自己搭建的局域网环境中进行的，其网络数据流量大小、数据包类型等方面与网吧、企业等复杂网络环境无法相比的，因此检测系统在性能上、准确度上等还有待进一步测试。

(2) 在网络流量检测全面性上有待补充完善，本检测系统只是针对 TCP 数据流和 DNS 数据包进行收集分析，对于其他网络层协议或者基于 UDP 的其他应用层协议可以进行深入分析。

(3) 本文实验只是针对检测系统的功能进行测试，关于检测系统性能也有待于进一步测试分析。

(4) 由于在该领域研究时间不长、钻研尚未透彻，依然存在一定的知识盲点、漏洞，在进行网络流量和主机行为相似性分析时，采用何种分析策略、聚类方法准确高效地区分正常主机表现出的与 Bot 主机相似的流量或行为等仍有待于进一步的深入研究、优化。

针对以上提出的不足之处，在将来的研究工作中将会从检测算法、检测准确度、兼容性、系统负载能力等方面深入研究不断完善该检测系统，并将其部署在各种复杂的网络环境中测试，通过不断地测试、发现问题、解决问题来增加其实用性，减小僵尸网络在互联网中的威胁。

致 谢

值此论文完成之际，向在这三年学习期间内给予我关心、支持和帮助的老师、同学以及多年来一直默默付出的家人表达我最真诚的、由衷的感谢。

首先我要感谢我的导师侯孟书教授这三年来对我的培养和教导，他用他严谨的治学方针、脚踏实地的科研态度、和一心为国家安全事业做贡献的伟大追求不断鞭策着我、影响着我。在学术上，他教会我在遇到问题时如何来看待它、正确的分析和结局问题的方法；在思想上，教导我脚踏实地、刻苦努力、树立正确的人身观、价值观和世界观；在生活上，能够给予我无私的关怀和帮助，解决生活中的困扰。感谢侯老师的辛勤教导，感谢你不仅授我以“鱼”，而且授我以“渔”，在此再次向我的导师表达我最诚挚谢意。

其次，非常感谢实验室的李玉军老师，李老师勤奋刻苦的工作态度、谨慎负责的科研作风都给我留下深刻影响。在工作中陪着我们一起“奋战”，始终对我们的工作提供极大地帮助和支持，不断地给予我们信心，在生活上也给予无私的关心和爱护。

再次，感谢实验室的同学们营造了一个轻松愉快的生活环境和专心刻苦的科研氛围，大家互相帮助、互相学习、共同提高。

最后，我要感谢我的父母及所有家人，感谢他们一直以来辛辛苦苦、默默无私的奉献。他们对我的关心、爱护和支持总是毫无保留的付出，是我追求学业过程中最坚强的后盾和精神支撑。

向在百忙之中抽出时间对本文进行评审并提出宝贵意见，以及前来参加答辩的各位专家、教授表示由衷的感谢。

参考文献

[1] 微软瓦解最大僵尸网络：百万台 PC 曾中招.[OL] http://news.zol.com.cn/418/4186952.html

[2] 国家计算机网络应急技术处理协调中心 著. 2012 年中国互联网网络安全报告[R]. 北京: 国家计算机网络应急技术处理协调中心，2013

[3] 解读 2013 华为僵尸网络与 DDOS 攻击报告[OL]. http://safe.zol.com.cn/411/4112749_all.html

[4] C. Kanich，K. Levchenko，B. Enright，et al. The Heisenbot uncertainty problem: Challenges in separaring bots from chaff[C]. Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats. Berkely, CA: USENIX, 2008:1-9

[5] A. Ramachandran，N. Feamster. Understanding the neW ork level behavior of spammers[C]. Proceedings of the 2006 Conf on Applications, Technologies, Architectures and Protocols for Computer Communication. New York: ACM,2006: 291-302

[6] T. Micro. Taxonomy of Botnet Threats[R]. Trend Micro White Paper, 2006.

[7] M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, K. Hamlen. Flow-based identification of botnet traffic by mining multiple log files[C]. First International Conference on Distributed Framework and Applications, DFmA 2008, 2008: 200–206.

[8] M. Szymczyk. Detecting botnets in computer networks using multiagent technology[C].Fourth International Conference on Dependability of Computer Systems, DepCos-RELCOMEX’09, 2009: 192–201.

[9] K. Xu，D. Yao，Q. Ma，et al. Detecting infection onset with behavior-based policies[C]. Network and System Security (NSS), 2011 5th International Conference on. IEEE, 2011: 57-64.

[10] G. Gu，V. Yegneswaran，P. Porras，et al. Active botnet probing to identify obscure command and control channels[C]. Computer Security Applications Conference, 2009. ACSAC'09. Annual. IEEE, 2009: 241-253.

[11] J. Goebel，T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation[C]. Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets. 2007: 8-8.

[12] W. T. Strayer，D. Lapsely，R. Walsh，et al. Botnet detection based on network behavior [M]. Botnet Detection. Springer US, 2008: 1-24.

[13] H. Choi，H. Lee，H. Kim. BotGAD: detecting botnets by capturing group activities in network

traffic[C]. Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE. ACM, 2009: 2.

[14] P. Wang，S. Sparks，C. C. Zou. An advanced hybrid peer-to-peer botnet [J]. Dependable and Secure Computing, IEEE Transactions on, 2010, 7(2): 113-127.

[15] 诸葛建伟，韩心慧，叶志远等.僵尸网络的发现与跟踪[C].全国网络与信息安全技术研讨会，北京，2005:1-7

[16] 王涛.僵尸网络检测与传播抑制[D].广州:中山大学，2010:1-104.

[17] 王斌斌.僵尸网络检测方法研究[D].武汉:华中科技大学，2010:1-110.

[18] 苏云琳.僵尸网络检测系统的分析与设计[D].北京:北京邮电大学，2010:1-76.

[19] 严庆.基于Bot会话关联的僵尸网络检测方法[D].成都:电子科技大学，2009:1-62.

[20] 李超.基于行为特征的IRC僵尸网络检测技术研究[D].哈尔滨:哈尔滨工业大学，2008:1-58.

[21] B. Saha，A. Gairola. Botnet: an overview [J]. CERT-In White Paper, CIWP-2005-05, 2005, 240.

[22] A. M. Rajab，J. Zarfoss，F. Monrose，et al. A multifaceted approach to understanding the botnet phenomenon[C]. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. ACM, 2006: 41-52.

[23] E. Cooke，F. Jahanian，D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets[C]. Proceedings of the USENIX SRUTI Workshop. 2005, 39: 44.

[24] D. Dagon，G. Gu，C. P. Lee，et al. A taxonomy of botnet structures[C]. Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual. IEEE, 2007: 325-339.

[25] Z. Zhu，G. Lu，Y. Chen，et al. Botnet research survey[C]. Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE, 2008: 967-972.

[26] H. Choi，H. Lee，et al. Botnet detection by monitoring group activities in DNS traffic[C]. Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on. IEEE, 2007: 715-720.

[27] Dynamic updates in the domain name system (dns update) [OL]. http://www.faqs.org/rfcs/rfc2136.html/.

[28] R. Villamarín-Salomón，J. C. Brustoloni. Identifying botnets using anomaly detection techniques applied to DNS traffic[C]. Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE. IEEE, 2008: 476-481.

[29] L. McLaughlin. Bot software spreads, causes new worries [J]. Distributed Systems Online, IEEE, 2004, 5(6): 1.

[30] N. Provos，P. Mavrommatis，M. A. Rajab，et al. All your iFRAMEs point to us[C].

Proceedings of the 17th Conf. on Security Symp. San Jose: USENIX Association, 2008. 1-15.

[31] C. Kanich，C. Kreibich，K. Levchenko，et al. Spamalytics: An empirical analysis of spam marketing conversion[C]. Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008: 3-14.

[32] D. Turner，M. Fossi，E. Johnson，et al. Symantec global internet security threat report–trends for july-december 07[J]. Symantec Enterprise Security, 2008, 13: 1-36.

[33] B. McCarty. Botnets: Big and bigger [J]. Security & Privacy, IEEE, 2003, 1(4): 87-90.

[34] M. Bailey，E. Cooke，F. Jahanian，et al. A survey of botnet technology and defenses[C]. Conference for Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology. IEEE, 2009: 299-304.

[35] Z. Zhu，G. Lu，Y. Chen，et al. Botnet research survey[C]. Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE, 2008: 967-972.

[36] M. Fossi，G. Egan，K. Haley，et al. Symantec internet security threat report trends for 2010[J]. Volume, 2011, 16: 20.

[37] H. R. Zeidanloo，A. A. Manaf. Botnet command and control mechanisms[C]. Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on. IEEE, 2009, 1: 564-568.

[38] J. Stewart. Bobax Trojan analysis [OL]. http://www.secureworks.com/research/threats/bobax/

[39] N. Daswani，M. Stoppelman. The anatomy of Clickbot. A[C]. Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets. USENIX Association, 2007: 11-11.

[40] K. Chiang，L. Lloyd. A case study of the rustock rootkit and spam bot[C]. The First Workshop in Understanding Botnets. 2007, 20.

[41] I. Arce，E. Levy. An analysis of the slapper worm [J]. Security & Privacy, IEEE, 2003, 1(1): 82-87.

[42] J. Stewart. Sinit P2P trojan analysis [OL]. http://www.secureworks.com/research/threats/sinit,2003.

[43] J. Stewart. Phatbot trojan analysis [OL]. http://www.secureworks.com/research/threats/phatbot,2004.

[44] S. Stover，D.Dittrich，J Hernandez， et al. Analysis of the Storm and Nugache Trojans: P2P is here [J]. USENIX; login, 2007, 32(6): 18-27.

[45] N. M. Mukamurenzi. Strom Worm: A P2P Botnet [D]. Norway:

Norwegian University of Science and Technology,2008:5-14

[46] T.M.S. Labs, Security Labs Report January – June 2011 Recap[R], Security Labs, 2011.

[47] A. Caglayan，M. Toothaker，D. Drapaeau，et al. Behavioral analysis of fast flux service networks[C]. Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies. ACM, 2009: 48.

[48] J. Nazario，T. Holz. As the net churns: Fast-flux botnet observations[C]//Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on. IEEE, 2008: 24-31.

[49] J. E. Dunn. Srizbi grows into world's largest botnet [OL]. http://www.csoonline.com/article/356219/srizbi-grows-into-world-s-largestbotnet.

[50] B Stone-Gross，M Cova，B Gilbert， et al. Analysis of a botnet takeover [J]. Security & Privacy, IEEE, 2011, 9(1): 64-72.

[51] S Shin，G Gu，N Reddy，et al. A large-scale empirical study of conficker [J]. Information Forensics and Security, IEEE Transactions on, 2012, 7(2): 676-690.

[52] E. J. Kartaltepe，J. A. Morales，S Xu，et al. Social network-based botnet command-and-control: emerging threats and countermeasures[C]. Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2010: 511-528.

[53] Zh.Q. Zhang, X. Cui and Ch. G. Liu. Web2Bot: Botnet in Web 2.0 Era [OL]. http://www.raid-symposium.org/raid2011/fileslWeb2Bot.pdf

[54] S Nagaraja，P Mittal，C. Y. Hong，et al. BotGrep: Finding P2P Bots with Structured Graph Analysis[C]. USENIX Security Symposium. 2010: 95-110.

[55] G Gu，J Zhang，W Lee. BotSniffer: Detecting botnet command and control channels in network traffic [J]. 2008.

[56] G Gu，R Perdisci，J Zhang，et al. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection[C]. USENIX Security Symposium. 2008: 139-154.

[57] L Zhuang，J Dunagan，D. R. Simon，et al. Characterizing Botnets from Email Spam Recordsc[J]. LEET, 2008, 8: 1-9.

[58] 李翔，胡华平，刘波等. 基于行为相似性的pep僵尸网络检测模型[J] . 现代电子技术，2010，33(15): 132-138.

[59] 王涛，余顺争. 中心式结构僵尸网络的检测方法研究[J].计算机系统应用，2010, 31(3):510-516.

[60] 李晓祯，程佳，胡军. 基于聚类分析的僵尸网络识别系统[J].计算机系统应用，2009,18(8): 131-135.

[61] 梁其川，吴礼发. 一种新颖的pep僵尸网络检测技术[J].电脑知识与技术，2009, 22(5):6186-6188

[62] H Wang，Z Gong. Collaboration-based botnet detection architecture[C]. Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on. IEEE, 2009, 2: 375-378.

[63] 肖斌，张众，汪永益.基于蠕虫的大规模BotNet传播与控制研究[J].电脑与信息技术 2009,17(3).

攻硕期间取得的研究成果