华中科技大学

硕士学位论文

未知网络蠕虫检测与特征码自动生成的研究

姓名：蔡新

申请学位级别：硕士

专业：计算机系统结构

指导教师：李汉菊

20070605

摘　要

网络蠕虫以其快速、多样化的传播方式不断给网络世界带来灾害，与传统的主机病毒相比，网络蠕虫具有更强的繁殖能力和破坏能力。从蠕虫爆发到蠕虫被消灭的时间却越来越长，但从发现漏洞到蠕虫爆发的时间越来越短，采用人工方法控制蠕虫的快速传播是不现实的，这就亟需蠕虫的自动检测系统。新的网络蠕虫层出不穷，蠕虫变种也越来越多，破坏力也越来越大，而且更加隐蔽，传统的基于模式匹配算法的检测系统也难以胜任，这就需要能够检测未知蠕虫的系统。

网络蠕虫脱胎于病毒，但又于病毒有所不同，在定义了网络蠕虫之后，介绍了蠕虫的行为特点、实体结构和工作流程。从蠕虫检测方面考虑，重点分析了网络蠕虫的扫描方法，比较蠕虫自动扫描与黑客人为扫描的区别，并简要介绍了蠕虫的传染模型。

以蠕虫扫描失败的特征、感染主机的特征、蠕虫爆发时的网络特征为依据，主要研究在网络层检测未知网络蠕虫，采用基于出入口双向流量分析的方法来检测。目前多数蠕虫检测系统利用网络流量的某一属性来检测蠕虫的攻击，可能存在单点时效性，故对入口流量进行多属性的相似度分析；考虑到静态时间窗口的流量统计失去了前后的相关性，故对出口流量进行基于滑动窗口的失败连接的统计分析。二者共同构建可疑属性数据库，作为检测未知蠕虫的依据。

捕获网络流量，参照可疑属性数据库，从中找出异常流量，认为其为可疑的未知蠕虫数据包，将其导入可疑流量池。当可疑流量池中的流量超过一定规模时，触发蠕虫特征码自动生成系统。特征码自动生成系统以可疑流量池为数据来源，输出未知蠕虫特征码到特征码数据库中。系统采用两种不同的方法来生成特征码：最多令牌特征码和最少令牌特征码，其基本思想都是根据同类蠕虫传播时数据包的相似性，从中提取公共的部分，构建令牌集，自动产生未知蠕虫的特征码。根据上述的研究，开发出生成未知网络蠕虫检测与特征码自动生成的系统原型，并进行简单的系统测试与问题分析。

关键字：网络蠕虫，异常流量，可疑流量池，可疑属性数据库，特征码，令牌

Abstract

Network worms pose a serious threat to the security of Internet infrastructures by their rapid and variety propagation. Compared with the traditional host virus, worms infect more quickly and destroy more severity. When the time between detecting vulnerability and the outbreak of worms is shorter and for long time to exterminate the worms, so it is not realistic to control the infection of worms by artificial methods, therefore automatic detection system is needed. Especially under the environment of Internet, the variety of the propagation ways result in worm with more worms variation, much deeper latency and much wider coverage. The traditional detection system which is based on signature is not suiting new various worms, so the system is able to detected unknown worms.

Network worms come from virus, but it is different with virus. After define the network worms and introduce the pivotal character include the behavior, the structure and the working flow of worms. Form the aspect of detecting the worms, we analyze the scanning methods and distinguish the difference between the automatic scanning of worms and the artificial scanning of hackers, and intro the propagation model.

Then based on the worms' behavior, we mainly study the unknown worms in the network layer. We analyze the network traffic which obtained by network trap, take into the multi-attribute of inbound traffic (e.g. source IP, destination port) account, at the same time compute the outbound traffic depended on moving windows of the failing traffic, then construct the suspicious attribute database. After that, we find the suspicious unknown worms in the inbound traffic which refer to the suspicious attribute database, and put it into the suspicious pool.

When the accounts of suspicious pool exceed the critical account, trigger the automated signature generation system. The system read data from the pool and export worms' signature to the database of signature. The system implements two methods to generate signature, both of them are apply the same principle. After eliminate the noise in the pool, system then extract the common substring in the pool and auto generate the signature of the unknown worms based on the similarity among worms from the same types. Essentially, I propose a model to detect unknown worms and automated generate signature, and then test it and discuss the emergent problems.

Keywords: network worms, anomaly traffic, suspicious flow, suspicious attribute database, signature, and tokens

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本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和汇编本学位论文。

本论文属于　保　密□，在_____年解密后适用本授权书。

　　不保密□。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　年　月　　日　　日期：　年　月　日

1 绪言

1.1 研究背景与意义

20世纪90年代以来，计算机技术的发展和互联网的扩大，给整个社会带来了日新月异的变化，促进了整个社会的进化，然而网络技术的飞速发展也带来了很多安全问题[1]。在网络安全问题中，恶意代码(Malicious Code)]造成的经济损失占有最大的比例[2]。恶意代码主要包括计算机病毒，网络蠕虫等等[3]。恶意代码流行日趋泛滥，蠕虫作为当前恶意代码中广为传播的一类，构成了当前对网络最大的威胁，几乎每次蠕虫发作都会因其造成的巨大经济损失而给人们留下深刻印象。

网络蠕虫虽然早在1988年就显示出来它的巨大破坏力和危害性，但当时网络没有普及，因而也没有引起人们更多的注意。1988年，Morris Worm感染了6千台机器，约占当时互联网上机器数量的10%，由于当时网络规模小，应用不广泛，因此蠕虫的爆发造成的损失还很小，也没有引起人们的重视。从1990年开始，对抗恶意代码破坏的主要内容锁定在个人电脑的防病毒上，科研人员的主要精力放在如何预防、检测和消除攻击个人电脑文件系统的病毒。邮件病毒的出现，使人们认识到了网络已经使病毒的性质发生了一些变化，需要调整研究方法和目标；另一方面，新蠕虫层出不穷，危害越来越大，其造成的危害程度远远超过传统的病毒，而由于对蠕虫的研究的滞后，使人们在蠕虫面前显得有些力不从心。

网络应用的迅速普及，网络规模的急剧扩大，为蠕虫的快速传播创造了条件，蠕虫对网络安全和应用的威胁与日俱增[4]。尤其是蠕虫由于快速的传播对网络造成了的极大的危害。不同时期蠕虫的屡次爆发，重新吸引了人们的注意，应付蠕虫的威胁已经是一项必须的任务，因此，对蠕虫的研究又多了起来。由于蠕虫的泛滥，应对蠕虫对于计算机网络安全问题有着重要的现实意义。

近几年随着互联网应用的深入，更加激发了计算机病毒传播的活力。根据CERT(Computer Emergency Response Team，计算机紧急响应小组)从1988年到2003年的统计数据表明，Internet安全威胁事件每年以指数增长[5]，近年来的增长态势变得尤为迅速，见图1.1；从1995年至今发现的漏洞个数也是每年以指数增长，见图1.2。

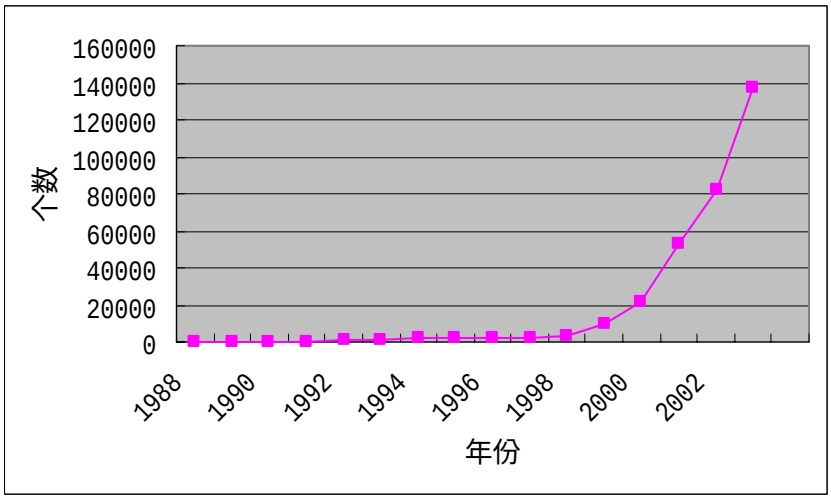

图1.1 Internet安全威胁事件示意图

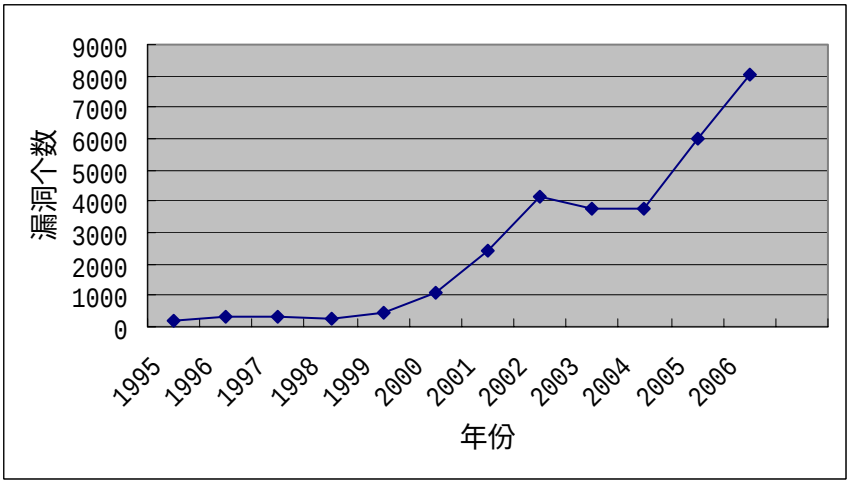

图1.2 漏洞个数示意图

由于网络本身是一个开放的多变量的复杂巨系统[6]，而蠕虫充分利用了网络开放、复杂的特点，它的传播与很多因素相关联。在这样复杂坎坷的环境下，如何对蠕虫进行检测和预警，从而对主机系统和网络进行最大限度的保护，从而有效的遏制蠕虫在大规模网络上的爆发，将蠕虫对网络造成的破坏降到最低[7]，是一个值得研究的话题。

随着网络革命的到来，网络在为发展带来巨大机会与可能性的同时，也带来了恶意入侵的风险。蠕虫在网络中的传播破坏了资源的完整性、可用性或有效性，它是一种可以传播的入侵手段。针对蠕虫的检测与遏制是入侵检测的一个重要方面。

蠕虫的传播速度非常快，而且蠕虫爆发的频率越来越高，但对于蠕虫的研究和防

御到目前为止还比较少，对于蠕虫的爆发还缺乏有力的手段进行监控。蠕虫的泛滥及其传播手段的改进，留给用于检测的反应时间越来越短，因此，构架快速的蠕虫检测平台已经是当务之急，这对于减少蠕虫对网络的破坏，并将其最小化，有着很大的现实意义。

1.2 国内外研究现状

1.2.1 网络蠕虫检测的研究现状

常见蠕虫检测技术主要有如下两种[9]：

(1) 异常检测技术(anomaly-based)

假定所有蠕虫行为都是与正常行为不同的。如果建立系统正常行为的轨迹，那么理论上可以把所有与正常轨迹不同的系统状态视为可疑企图。对于异常特征与阀值的选择是异常发现技术的关键。比如，通过流量统计分析将异常时间的异常网络流量视为可疑。异常发现技术的局限是并非所有的入侵都表现为异常， 而且系统的轨迹难于计算和更新。

(2) 模式匹配技术(signature-based)

假定所有入侵行为和手段(及其变种) 都能够表达为一种模式或特征，那么所有已知的入侵方法都可以用匹配的方法发现。模式发现的关键是如何表达入侵的模式，把真正的入侵和正常行为区分开来。模式发现的优点是误报少，局限是它只能发现已知的攻击，对未知的攻击无能为力。

对于未知蠕虫检测，本文简要总结了主要几种未知蠕虫发现系统：

EarlyBird System通过观察在正常的网络流量模式下的变化来发现蠕虫。它利用了Rabin指纹方法找出数据包载荷中重复内容的出现，同时用地址数的指数增长来鉴别蠕虫流量[10]。

Honeypots通过监视未用的地址空间，捕获与这些地址通信的流量，将其当作蠕虫入侵，蠕虫检测采用了黑名单的方法[11]。

DEWP采用两步检测，首先进行端口匹配，确定双向的端口是否一致，然后进行目的地址计数，确定目的地址个数是否有较大增长[12]，以其发现蠕虫。

Autograph采用黑名单方式初步确定正进行随机扫描的IP地址，监控这些地址发送的所有数据包，采用指纹识别的方法，提取数据包中可疑串，作为蠕虫特征串[13]。

1.2.2 网络蠕虫特征码自动生成的研究现状

目前特征码自动生成主要有以下几种方法：(1)不检测蠕虫攻击，直接从流量中生成特征码；(2)在网络层检测蠕虫攻击，捕获可疑流量，之后从中生成特征码；(3) 在

主机上检测蠕虫攻击，获取可疑数据，之后生成特征码；(4) 同时在网络层与主机上检测蠕虫攻击，之后生成特征码。

对于第一种方法，成型的系统有Polygraph[14]，它主要针对多态蠕虫，提取多态蠕虫的特征码。Polygraph利用流量分类器对流量进行分类，流量分类器以端口为单位重组流量，将可疑流量放入可疑流量池中。特征码生成系统的数据输入来源是可疑流量池，其输出结果蠕虫特征码。文中定义了三种不同形式的特征码：联合令牌特征码、序列令牌特征码和Bayes特征码，三种特征码的基本元素都是字符子串，即令牌。

还有Earlybird[10]，其原理是在网络的边界路由器上进行流量监测,并计算网络上所有长度为β的字符串的Rabin签名。如果某些数据包频繁出现在所监控的网络上，就会得到大量重复的签名。当重复次数超过某个阈值时，系统就认为网络上出现了新蠕虫，并根据得到的签名进行数据过滤。注意这里不是对蠕虫样本进行签名，而是对观测到的数据，计算其所有长度为β的子字符串的Rabin签名，然后根据签名内容进行监测。Earlybird没有使用包过滤器，分析所有进入的数据包，检测所有数据包的流行度，基于公用的攻击字符序列，来提取Rabin签名；使用源地址离散分布的特性，来减少系统的误报率。

对于第二种方法，比较著名的有Autograph[13]、PADS[15]和PAYL[16]系统。

Autograph采用网络层检测，从中提取可疑流量以生成特征码的方法。Autograph使用一个原始的，基于端口扫描的流量分类器，找出异常可疑的流量，在此基础上执行内容流行度分析(COPP)以产生特征码。

PADS(Position-aware distribution signatures)系统使用双密罐系统以追踪本地网络中的恶意行为，设置两个蜜罐：高交互式的密罐和低交互式的密罐。从高交互式的密罐中收到的所有连接都转向到低交互式的密罐中，其依据是：一个受感染的密罐迟早会感染其它系统，因此蠕虫代码的多重变量可以很容易的在低交互式密罐上生成。

PAYL是异常检测器，检测入口流量的异常负载，与同一端口上的出口流量相关联。PAYL使用无人监管的机器学习技术，对主机上正常的入口流量和出口流量的特征进行建模；在训练学习阶段，PAYL计算n-grams的统计概率，n-grams是指在一个负载中n个邻近字节的序列。之后这样假设：如果PAYL在出口流量中检测到发往端口i的异常包，这个包与PAYL在端口i接受的入口异常包相象，那么这个入口流量数据包是蠕虫数据包的概率就很大。据此可从中提取出蠕虫的特征码。

主机层生成特征码则是所谓的蜜罐系统或者陷阱系统，如Honeyd系统[17]。蜜罐

技术是一种欺骗入侵者以达到采集黑客攻击方法和保护真实主机目标的诱骗技术。该不同于大多数传统安全机制，它的安全资源的价值是在于它主动去被探测，被攻击，或者被威胁。Honeyd是一个小的防护程序，它能够产生虚拟的主机，这些主机能够被配置以提供任意的服务，系统特征也是与之相适应，以至于使之看起来像真实的系统在 运行。在一个局域网的网络仿真中，Honeyd能够使单个主机拥有许多IP（多达65536个）。通过分析蠕虫对Honeyd系统的攻击数据包，从中提取出蠕虫特征码。

同时在主机层和网络层检测蠕虫的系统是HoneyStat[18]。它主要针对局域网的攻击检测。HoneyStat主要监测三种事件：内存事件，网络事件和硬盘事件。内存事件由运行在密罐上的缓冲区溢出检测软件发现；当密罐产生出口流量时，触发网络事件；如果一个进程向文件系统的特殊部分写数据，触发硬盘事件。HoneyStat 并不从事件中生成特征码，但是当事件发生时，记录以下信息：操作系统/补丁的版本；事件的种类及相关捕获的数据；所有优先的网络活动的日志文件。收集到的信息都发往中央分析节点，该节点收到所有事件信息后，执行逻辑回归分析(logistic regression)。

1.3 主要研究内容

模式匹配的优点在于准确性高，而对特征库中没有的攻击无法检测；异常检测能够发现新出现的攻击，但准确性低。目前对恶意代码的防治通常使用模式匹配。然而随着人们对系统机理、网络的了解越来越深入，编制更加复杂的恶意代码已经越来越容易。网络上越来越多、形形色色的恶意代码，对网络的破坏非常严重，尤其是新的蠕虫的出现总是很突然，很难把握它们的规律，预测什么样的蠕虫会出现，从而给传统的基于特征的模式匹配的方法造成了很大的困扰，因此本文必须研究基于异常检测的早期发现蠕虫的有效手段。

蠕虫的响应自动化是蠕虫防御的必然趋势。从模式匹配和异常检测两个角度来对蠕虫进行检测，是为了应对当前蠕虫泛滥的趋势而必须做的工作。其中模式匹配为了能够准确的检测已知蠕虫；而异常检测则在大规模网络下对新蠕虫进行预警。蠕虫的自动防御则要求在异常发现的基础上，发现大规模网络上的蠕虫爆发，并生成其特征，从而进行特征检测。对新出现的疫情爆发能够及时的发现，则可以为进一步的控制提供基础，并显著减小蠕虫所造成的损失，且使用模式匹配与异常发现相结合的手段，其模型和系统实现在实际环境中有着广泛的实用价值。

蠕虫检测可以在网络层，通过捕获网络流量来分析；也可以在应用层，通过监控

主机上的异常操作来发现蠕虫。本文的研究针对网络层，在路由器、网关、防火墙等等设备上，利用网络分接器捕获网络流量，对流进流出的网络流量进行分析。

根据网络流量的异常，发现可疑的流量，检测出未知的网络蠕虫，之后将可疑的流量导入可疑流量池。可疑流量池是特征码生成的数据来源，特征码生成系统从中生成蠕虫特征码。如何生成蠕虫特征码是本文的又一主要内容。

根据研究内容，可以确定系统的基本运行过程，即本系统从网关设备上获取网络流量，在独立的系统上运行，分析网络流量，从中找出可疑的网络流量，之后，将可疑的流量导入可疑流量池，由特征码自动生成系统生成未知蠕虫的特征码，生成的特征码与其它设备交互，如图1.3所示。

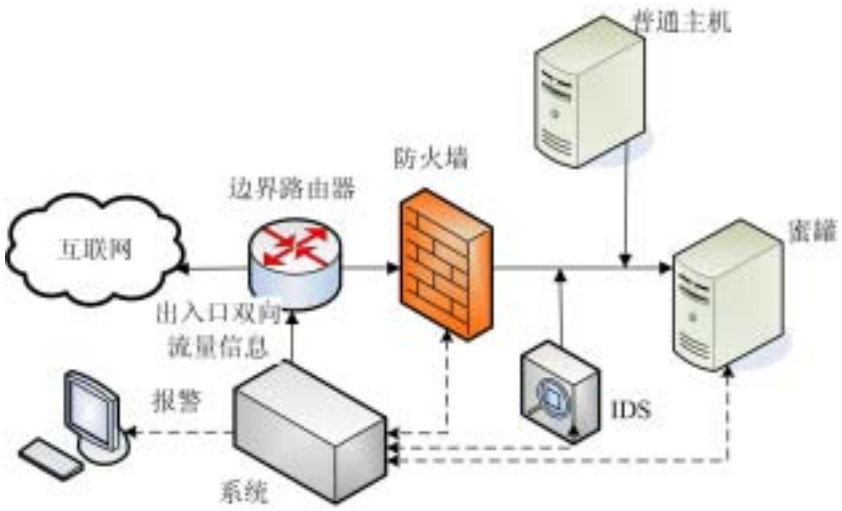

图1.3 系统运行过程示意图

2 网络蠕虫的关键特征

2.1 网络蠕虫的定义与特点

2.1.1 网络蠕虫的定义

网络蠕虫是无须计算机使用者干预即可运行的独立程序，它通过不停的获得网络中存在漏洞的计算机上的部分或全部控制权来进行传播。而计算机病毒是一段代码，能把自身加到其它程序包括操作系统上。它不能独立运行，需要由它的宿主程序运行来激活它[19]。

计算机蠕虫和计算机病毒都具有传染性和复制功能，这两个主要特性上的一致，导致二者之间是非常难区分的，尤其是近年来，越来越多的病毒采取了部分蠕虫的技术，另一方面具有破坏性的蠕虫也采取了部分病毒的技术，更加剧了这种情况。但对计算机蠕虫和计算机病毒进行区分还是非常必要的，因为通过对它们之间的区别、不同功能特性的分析，如表2.1所示。可以确定谁是对抗计算机蠕虫的主要因素、谁是对抗计算机病毒的主要因素；可以找出有针对性的有效对抗方案；同时也为对它们的进一步研究奠定初步的理论基础。

表2.1 病毒和蠕虫的一些差别

	病毒	**蠕虫**
存在形式	寄生	独立个体
复制机制	插入到宿主程序（文件)中	自身的拷贝
传染机制	宿主程序运行	系统存在漏洞
搜索机制	针对本地文件	针对网络上的其它计算机
触发传染	计算机使用者	程序自身
影响重点	文件系统	网络性能、系统性能
用户角色	病毒传播中的关键环节	无关
防治措施	从宿主文件中摘除	为系统打补丁（Patch)
对抗主体	计算机使用者、反病毒厂商	系统提供商、网络管理人员

2.1.2 网络蠕虫的行为特点

网络蠕虫有一些明显的外在行为特点，这些特征是分析蠕虫的基础。

(1) 主动攻击：蠕虫在本质上已经演变为黑客入侵的自动化工具，当蠕虫被释放后，从搜索漏洞，到利用搜索结果攻击系统，到复制副本，整个流程全由蠕虫自身主动完成[20]。

(2) 行踪隐蔽：由于蠕虫的传播过程中，不象病毒那样需要计算机使用者的辅助

工作（如执行文件、打开文件、阅读信件、浏览网页等等)，所以蠕虫传播的过程中计算机使用者基本上不可察觉[21]。

(3) 利用系统、网络应用服务漏洞：计算机系统存在漏洞是蠕虫传播的前提，利用这些漏洞，蠕虫获得被攻击的计算机系统的相应权限，完成后继的复制和传播过程。这些漏洞有的是操作系统本身的问题，有的是应用服务程序的问题，有的是网络管理人员的配置问题。正是由于漏洞产生原因的复杂性，导致面对蠕虫的攻击防不胜防[21]。

(4) 造成网络拥塞：蠕虫进行传播的第一步就是找到网络上其它存在漏洞的计算机系统，这需要通过大面积的搜索来完成，搜索动作包括：判断其它计算机是否存在；判断特定应用服务是否存在；判断漏洞是否存在。这不可避免的会产生附加的网络数据流量。即使是不包含破坏系统正常工作的恶意代码的蠕虫，也会因为它产生了巨量的网络流量，导致整个网络瘫痪，造成经济损失[22]。

(5) 降低系统性能：蠕虫入侵到计算机系统之后，会在被感染的计算机上产生自己的多个副本，每个副本启动搜索程序寻找新的攻击目标。大量的进程会耗费系统的资源，导致系统的性能下降。这对网络服务器的影响尤其明显。

(6) 产生安全隐患：大部分蠕虫会搜集、扩散、暴露系统敏感信息（如用户信息等)，并在系统中留下后门。这些都会导致未来的安全隐患。

(7) 反复性：即使清除了蠕虫在文件系统中留下的任何痕迹，如果没有修补计算机系统漏洞，重新接入到网络中的计算机还是会被重新感染。这个特性在Nimda蠕虫的身上表现的尤为突出，计算机使用者用一些声称可以清除Nimda的防病毒产品清除本机上的Nimda蠕虫副本后，很快就又重新被Nimda蠕虫所感染[23]。

(8) 破坏性：从蠕虫的历史发展过程可以看到，越来越多的蠕虫开始包含恶意代码，破坏被攻击的计算机系统，而且造成的经济损失数目越来越大[24]。

2.2 网络蠕虫的结构与流程

2.2.1 网络蠕虫的实体结构

蠕虫程序相对于一般的应用程序，在实体结构方面体现更多的复杂性，通过对多个蠕虫程序的分析，可以粗略的把蠕虫程序的实体结构分为如下的六大部分，具体的蠕虫可能是由其中的几部分组成[25]：

(1) 未编译的源代码：由于有的程序参数必须在编译时确定，所以蠕虫程序可能包含一部分未编译的程序源代码；

(2) 已编译的链接模块：不同的系统（同族)可能需要不同的运行模块，例如不同

的硬件厂商和不同的系统厂商采用不同的运行库，这在UNIX族的系统中非常常见；

(3) 可运行代码：整个蠕虫可能是由多个编译好的程序组成；

(4) 脚本：利用脚本可以节省大量的程序代码，充分利用系统shell的功能；

(5) 受感染系统上的可执行程序：受感染系统上的可执行程序如文件传输等可被蠕虫作为自己的组成部分；

(6) 信息数据：包括已破解的口令、要攻击的地址列表、蠕虫自身的压缩包等等；

鉴于所有蠕虫都具有相似的功能结构，本文给出了蠕虫程序的统一功能模型，统一功能模型将蠕虫程序分解为基本功能模块和扩展功能模块。实现了基本功能模块的蠕虫程序就能完成复制传播流程，包含扩展功能模块的蠕虫程序则具有更强的生存能力和破坏能力。如图2.1所示。

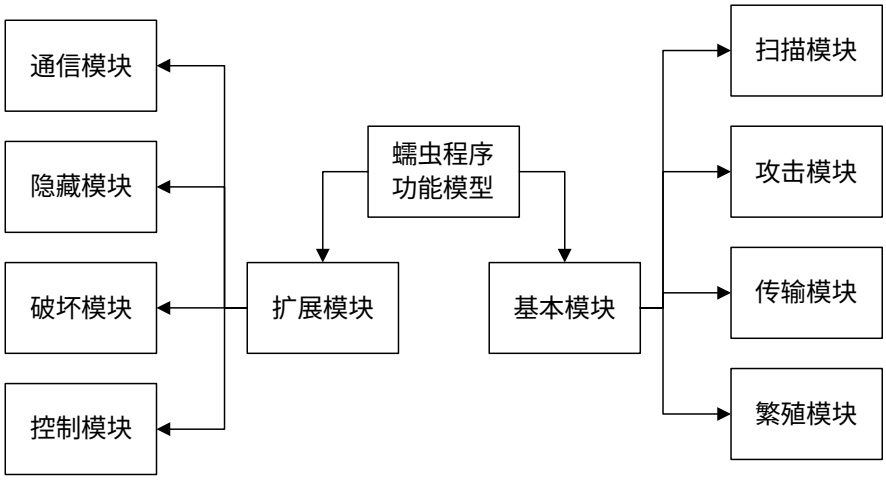

图2.1 蠕虫程序功能模型图

基本功能由五个功能模块构成：

(1) 扫描模块：寻找下一台要传染的机器；为提高搜索效率，可以采用一系列的搜索算法。

(2) 攻击模块：在被感染的机器上建立传输通道（传染途径)；为减少第一次传染数据传输量，可以采用引导式结构。

(3) 传输模块：计算机间的蠕虫程序复制；

(4) 繁殖模块：建立自身的多个副本；在同一台机器上提高传染效率、判断避免重复传染。

扩展功能又四个功能模块构成：

(1) 隐藏模块：隐藏蠕虫程序，使简单的检测不能发现。

(2) 破坏模块：摧毁或破坏被感染计算机，或在被感染的计算机上留下后门程序

等等；

(3) 通信模块：蠕虫间、蠕虫同黑客之间进行交流，可能是未来蠕虫发展的侧重点；

(4) 控制模块：调整蠕虫行为，更新其它功能模块，控制被感染计算机，可能是未来蠕虫发展的侧重点；

2.2.2 网络蠕虫的工作流程

蠕虫作为个体，其一般传播过程为：

(1) 扫描：由蠕虫的扫描功能负责探测存在漏洞的主机。当程序向某个主机发送探测漏洞信息并收到成功的反馈信息后，就得到一个可传播的对象。

(2) 攻击：攻击模块自动攻击在第1步中找到的对象，取得该主机的权限（一般为管理员权限)。

(3) 现场处理：使计算机在被感染后保留一个后门以便发动分布式拒绝服务攻击。

(4) 复制：复制模块通过原主机和新主机的交互将蠕虫程序复制到新主机并激活。蠕虫病毒的工作流程如图所示。

图2.2 蠕虫工作流程图

2.3 网络蠕虫的扫描

端口扫描是通过对目标系统端口试探性的访问来判断端口是否开放的行为。它往往是蠕虫传播行为的第一步。因此分析蠕虫的扫描行为是检测蠕虫的主要手段。

2.3.1 端口扫描的种类

端口扫描可以分成两类：水平扫描和垂直扫描[26]。水平扫描是对多个主机的同一个端口进行扫描．垂直扫描是对同一个主机的多个端口进行扫描，对于水平攻击者在不同主机上访问的端口有共同之处，而正常网络应用在不同土机上访问的端口很可能不同。由于扫描者事先不知道端口是否开放，所以正常主机建立连接的成功率要大于扫描者建立连接的成功率[27]。

2.3.2 端口扫描的方法

蠕虫传播之前一般用ICMP Ping包、TCP SYN、FIN、RST和ACK包来进行扫描检测[28]。影响蠕虫传播的主要因素是如何能快速找到新的目标主机，所以扫描方法的性能直接决定着蠕虫传播的速度。了解蠕虫的扫描方法对分析蠕虫的扫描行为大有裨益。

(1) 随机扫描法（Random Scanning)

所谓随机扫描是感染蠕虫的计算机在寻找新的攻击目标时不知道哪些计算机系统有漏洞，随机地选择网上计算机进行扫描，所以扫描的目标为IPv4所有地址空间。Code Red蠕虫和Slammer蠕虫都是采用随机扫描的方法[29]。

(2) Hit-list 扫描

这种蠕虫在传播之前先搜集一些有漏洞的计算机地址列表，传播时先感染这些地址列表中的计算机，然后这些感染的计算机再随机扫描互联网上的其他计算机。采用Hit-list扫描的蠕虫要比采用随机扫描的蠕虫传播的速度快得多[29]。

(3) 路由扫描法（Routing Scanning)

采用BGP 路由表中的地址来减小要扫描的地址空间。由于在BGP 路由表中，只包含了28. 6%的IPv4地址空间 [30] ，所以，若采用路由扫描法，其探测地址空间将比随机扫描减小，所以传播速度会增加。

(4) 分而治之扫描（Divide-Conquer Scanning)

在释放蠕虫之前，作者需要收集一个有10 000 ～ 50 000个漏洞且网络连接良好主机的列表，在传播时，蠕虫给每个受害主机发送一个子列表，受害主机就按照子列表进行扫描[31]。

(5) DNS 扫描法（DNS Scanning)

蠕虫作者利用从DNS 服务器得到的IP地址建立目标主机地址列表，它的优点是这些地址都是可用的，提高了扫描的准确度。但缺点也很多：a.这个地址列表不易得到；b.列表涉及范围太小，因为都是来自同一个DNS服务器；c.蠕虫需要携带一个大的地址列表，传播速度受限[31]。

(6) 顺序扫描(sequential scan)

顺序扫描是指被感染主机上蠕虫会随机选择一个C类网络地址进行传播。根据本地优先原则，蠕虫一般会选择它所在网络内的IP地址.若蠕虫扫描的目标地址IP为A，则扫描的下一个地址IP为A+1 或者A-1。一旦扫描到具有很多漏洞主机的网络时就会达到很好的传播效果.该策略的不足是对同一台主机可能重复扫描，引起网络拥塞[31]。

2.3.3 扫描方法的性能比较

网络蠕虫传播速度的关键影响因素有4个：目标地址空间选择、是否采用多线程搜索、易感染主机是否有易感染主机列表(hit-list)以及传播途径的多样化。各种扫描策略的差异主要在于目标地址空间的选择。网络蠕虫感染一台主机的时间取决于蠕虫搜索到易感染主机所需要的时间。

一般情况下[21]，采用DNS扫描传播的蠕虫速度最慢，选择性扫描和路由扫描比随机扫描的速度要快；对于Hit-list扫描，当列表超过1M字节时，蠕虫传播的速度就会比路由扫描蠕虫慢；当列表大于6M时，蠕虫传播速度比随机扫描还慢。分治扫描目前还没有找到易于实现且有效的算法。目前，网络蠕虫首先采用路由扫描，再利用随机扫描进行传播是最佳选择。

这几种扫描方法的根本不同之处在于目的主机列表的选择上，当列表大小相同时，蠕虫感染其他主机的时间主要取决于这个列表的有效率，因为这个列表可能包括未分配或保留的地址。地址列表的粒度越小，蠕虫扫描就越隐蔽，但是蠕虫代码也就越大，所以需要在地址列表大小和蠕虫传播速度间找一个合理的平衡度[32]。

2.3.4 蠕虫扫描与黑客扫描的区别

网络蠕虫的核心工作机制与黑客渗透方法极其类似，它们都有信息收集、扫描探测、攻击渗透；不同在于蠕虫在渗透到被感染主机后，会自动控制被感染主机，通过目标选择策略选择向多台其他网络上的主机再进行渗透，形成多米诺效应[33]。

蠕虫传播与黑客攻击也存在不同：蠕虫代码的传播只有少量的攻击行为，因为是在全球范围内传播，故通常采用一些动态传播模型；而黑客的攻击更加复杂，通常攻击目标是一个或一个特定集合的主机，而不是遵循一个定义良好的动态模型[34]。

黑客扫描一般在时间上没有限制，为防止防火墙和IDS之类设备的检测，黑客可以使用慢速扫描等方法来逃避网络安全设备的检测；而蠕虫传播一般却不会采用这种方式。黑客扫描一般以垂直扫描为主，结合水平扫描；而蠕虫扫描一般是水平扫描，扫描某一端口的漏洞。

2.4 网络蠕虫的传播模型

了解蠕虫的工作流程和扫描方法之后，就要关注蠕虫的传播模型。如果说工作流程是个体运动的规律，那么传播模型就是整体运动的规律。如果说扫描方法是微观方面，那么传播模型就是宏观方面。

理想的网络蠕虫传播模型能够充分反映蠕虫的传播行为，识别网络蠕虫传播链中存在的薄弱环节，同时可以预测网络蠕虫可能带来的威胁。蠕虫的传播适合采用传

染病传播模型。常见的传染病传播模型是经典传染病模型(Simple Epidemic Model，简称SEM)[35]，如图2.3所示。

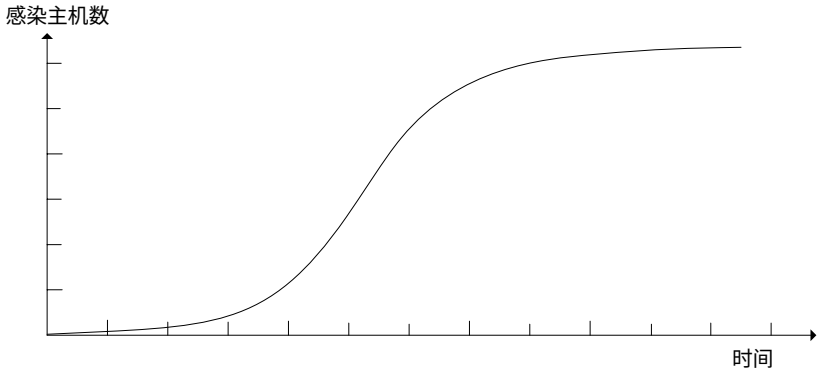

图2.3 蠕虫传染示意图

在SEM中，每台主机保持两种状态：易感染的和感染的。模型假定一台主机一旦被感染就始终保持被感染的状态，因此状态转变过程是：易感染--被感染。SEM模型的微分方程表达式为：

$$dI(t)/dt = \beta I(t)[N - I(t)] \tag{2.1}$$

在公式(1)中，I(t)为时刻t已被感染的主机数；N 为网络中主机总数；b为主机感染率。当t=0时，I(0)为已感染的主机数，N-I(0)为易感染主机数。

令 $a(t)=I(t)/N$，根据式(1)可以得出下面的方程：

$$da(t)/dt = KI(t)[1-a(t)], K = \beta N \tag{2.2}$$

取节点数N=10000000，感染概率因子为b=1/10000000，即K=b，N=1，当蠕虫繁殖副本数量I(0)=3 时，仿真结果如图2.4所示，横坐标为传播时间，纵坐标为整个网络被感染的百分比。

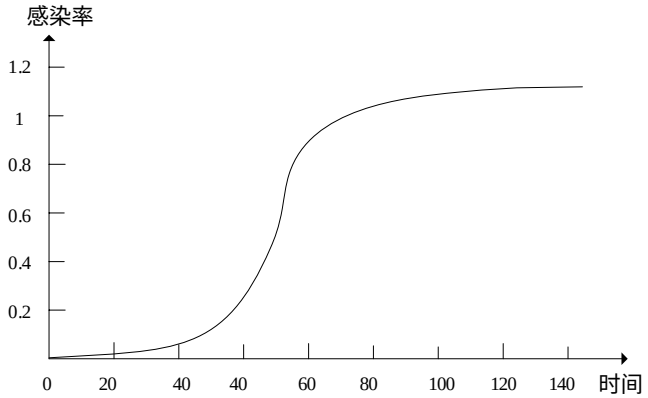

图2.4 SEM模型中网络蠕虫传播趋势仿真图

3 未知网络蠕虫的检测

本文的未知蠕虫检测与特征码自动生成存在两个基本的假设：1)内网是安全的，而外网是不安全的，蠕虫是从外网向内网攻击；2)在某一时刻，只有一种蠕虫在攻击，这种蠕虫可能存在多个变种。第一种假设决定了蠕虫检测的基本方向，第二种假设确定了特征码自动生成的基调。

3.1 未知网络蠕虫检测的基本依据

3.1.1 蠕虫扫描失败时的特征

据前文所述，在蠕虫传染时，一台被蠕虫感染的机器会持续不断的产生大量连接失败信息，正是这大量的失败连接造成了ICMP和TCP协议状态的异常[36]。失败连接由主机或路由器产生，主要有以下两种：

(1) ICMP-T3报文

ICMP报文的作用是当目的IP地址或路由出现问题时提供一个反馈信息，蠕虫传播时产生的报文类型主要是“目的不可达”报文，在各种形式的目的不可达报文中主要有目的主机不可达、目的网络不可达和目的端口不可达报文。由于目的不可达报文在ICMP报文头部信息的类型码值为3，就把该类报文称为ICMP-T3报文。RFC文档规定路由器必须要提供产生ICMP-T3类型报文的功能，但不是必须要使用这个功能[37]。

ICMP-T3主机不可达报文是由本地网络路由器产生的。当一个报文到达它的本地网络时，如果没有主机应答，那么路由器就会返回给发送者一个ICMP-T3目的主机不可达的报文。ICMP-T3网络不可达报文产生于传输层的路由器，路由器查询路由表失败，就会丢弃该报文并返回一个ICMP-T3网络不可达的报文给发送者。

使用和分析ICMP-T3类型报文中包含的原始报文头部就会得到有用的信息。该类型报文包括了原始报文中源IP地址、目的IP地址、源端口和目的端口等内容，这部分信息能够反应发送报文者的企图。蠕虫扫描报文的响应种类有：连接响应、连接RESET报文、目的不可达，还有大部分根本就没有任何回应(路由器不是必须要进行响应或者是防火墙之类的策略设置使然)。从历次蠕虫扫描的统计数据来看，其中的ICMP-T3报文在响应报文中占了约83%。

(2) TCP RESET报文

TCP连接失败是当一个报文的目的主机不存在或者目的主机存在但目的端口不

开放时产生。如果一个SYN报文发送给一个根本不存在的主机，那么ICMP主机不可达报文就会返回；如果一个SYN报文发送给地址存在但是目的端口关闭的主机，那么TCP RESET报文就会返回。统计数据表明TCP RESET报文在响应报文中所占的比例约10%。

由于发送TCP RESET报文的是真实的响应主机，就可以利用本地网络的网关重构报文和蜜罐技术来分析得到真实的网络蠕虫扫描源，把真实蠕虫扫描源和大量的伪造扫描源地址区分开来。这既可以减少蠕虫检测监视系统的计算负荷， 又可以提高蠕虫报警的准确度。

由上可知，检测ICMP-T3报文和TCP RESET报文能够捕获绝大部分的扫描流量。对大型网络而言分析ICMP-T3类型的报文来建立模型；如果本地网络的地址空间是密集空间，那么利用传染模型来检测监视RESET报文就足够了。这样即可以从宏观上把握蠕虫传播的网络流量状况，又可以从具体网络中得到蠕虫的行为特点和较精确的扫描源集合。

3.1.2 受感染主机的特征

通过对蠕虫传播机制进行分析，可以发现，蠕虫传播具有比较明显的、异常的网络传输特征，感染网络蠕虫病毒的主机主要有以下三个特征[38]：

(1) 以尽可能的最大速度向IP地址不同的计算机发起连接请求。一台感染网络蠕虫病毒的主机的新连接发起频率可能达到至少10OHZ以上，而一台正常的主机通常仅为0.5HZ到2HZ。

(2) 网络连接之间缺乏地址相关性。感染网络蠕虫病毒的主机将尝试向大量不同IP 地址的主机发起连接，而正常的主机在连接之间具有一定的地址相关性，如会重复向同一个IP 地址发起连接请求。

(3) 感染网络蠕虫病毒的主机所发出的网络连接请求，绝大部分都不会收到应答；而正常的网络连接，只有小部分会没有应答。

3.1.3 蠕虫爆发时的网络特征

本文主要检测未知的网络蠕虫，通过研究当前各种蠕虫，如CodeRed，以及MS SQL Slammer等，可以发现蠕虫所共有的一些特征[10]。

(1) 大量相同或相似的数据流量

在蠕虫爆发初期，由于试图感染更多的主机而产生的扫描流量占用了相当大的网络带宽，而且这些流量有相似性。网络蠕虫在传播的各个阶段，感染节点传出的数据具有相似性，数据包都包含了相应的请求、攻击代码或蠕虫的主体程序，内容相

对稳定，因此其传输数据的大小基本不变。例如Code Red利用IIS 漏洞进入被感染节点的主机后，产生100个线程，前面99个线程都利用随机产生的IP 地址，探测其它节点主机是否存在IIS 的Indexing Service缓冲区溢出漏洞。在一定时间内每个线程对目的主机的请求内容都是相似的。而对于蠕虫变种而言，虽然攻击代码某些地方发生改变，但是其基本负载仍然保持不变。

(2) 感染规模不断增大

由于蠕虫的自传播特性，被感染的主机数量以及蠕虫的扫描流量都呈指数级迅速增长。

(3) 随机扫描

被感染的主机会随机探测某个地址的固定端口，来试图感染其他主机，所以常常会去试图连接一些未用的地址空间。

(4) 大量的无效IP地址和无效服务请求

网络蠕虫为了在网络中迅速传播和扩散，攻击目标的选择具有盲目性。信息的收集和探测都会导致大量无效IP地址的产生，由于攻击目标IP地址的无效性，因此相应的服务请求也得不到应答，或者产生失败的连接信息。这一点请参展前文。

从上面对网络蠕虫攻击行为模式的分析可以看出，如果网络之外的某一主机，在短时间内对内网进行大量的TCP连接请求，这些连接请求的目标端口相同、数据包大小一定、数据包内容相似，目标IP地址和目标服务都得不到应答，则可判断该网络可能被某种蠕虫侵入；如果在同时，内网向外传出很多失败连接的数据包，而数据包的源端口相同，目标IP或端口集中于某一段IP或端口，则也可判断网络正在被某种蠕虫入侵。结合上述两种的信息，进行一定的关联分析，可检测出未知网络蠕虫攻击的相关模式，如IP，端口，协议等等，这就是检测网络蠕虫的基本思想。

3.2 基于出入口双向流量分析的未知网络蠕虫检测

3.2.1 网络蠕虫检测的阶段与位置

结合上文，蠕虫的传播可用经典传染病模型粗略描述：在爆发阶段受感染主机数目急剧增加，短时间内达到相对稳定的峰值，如图3.1所示。爆发稳定在峰值以后，网络将会发生严重的阻塞。换言之，蠕虫传播共分为三阶段：慢速启动，快速传播，慢速结束[34]。本文提出的蠕虫爆发检测算法就是要在蠕虫传染爆发阶段的早期检测出异常，从而使管理人员有足够时间在网络发生阻塞之前采取措施。

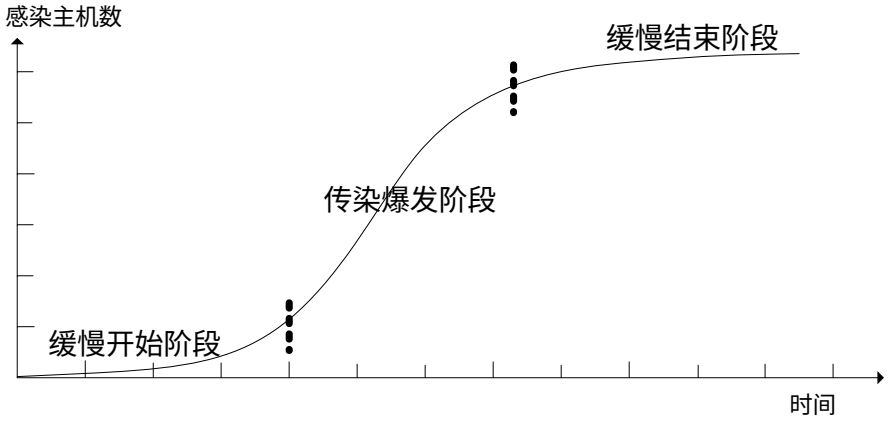

图3.1 蠕虫传染示意图

此算法检测的对象是对网络造成影响的蠕虫，它们应具有以下特征：(1) 爆发性：传播极其迅速，受感染主机的数目在很短时间内猛烈增加；(2) 持续性：如果不加控制，可以在较长时期内持续发挥作用；(3) 破坏性：导致网络严重阻塞甚至瘫痪[39]。

通过在路由器等网关设备上布置网络分接器，监测路由器出入口双向流量，对出口和入口的流量执行不同的检测。对路由器入口流量，执行数据包的多属性的分析；对出口流量，则主要关注于失败的连接，基本模型如图3.2所示。

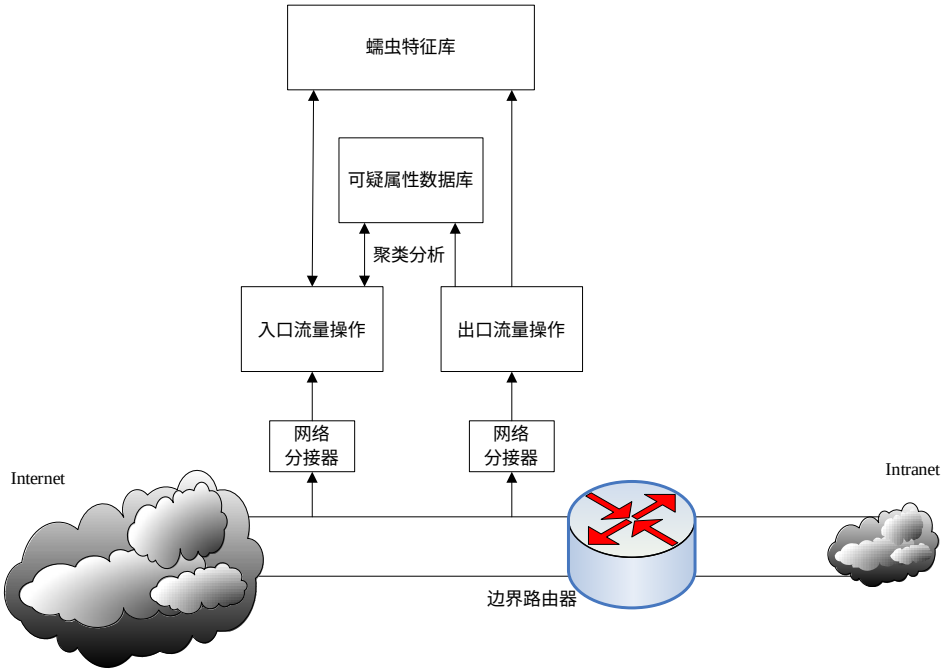

图 3.2 系统基本模型示意图

3.2.2 网络流量分析的基本原理

目前端口扫描的基本检测算法可以归结为：在一个时间窗 T 秒内，如果检测到从同一个源 X 发出，发往不同目的地Y(IP/PORT组合)的包多于N个，就判断为一次扫描。这种检测方法有一些缺陷：(1) 无法检测分布式的扫描[40]；(2) 无法处理慢速扫描，因为时间窗T是一个固定值，而攻击者只要使扫描速度低于这个阈值，就可以避开扫描检测；(3) 很难确定静态阈值T与N到底取何值才最合适[41]。

本文以此为基础，对于入口流量，对源与目的地进行具体的分析，分别考虑端口与IP，同时考虑协议的变化与数据包长度的变化，并将简单的个数判断转化为变化率的判断。大多数检测系统只是根据网络上数据包的单一属性进行检测，这样会存在单点失效问题，也就是说如果所利用的属性(唯一属性)没有出现异常，或者产生的异常没有达到预先设置的异常条件时就会产生漏报。为了避免这种单点失效问题，本文选取了多个属性(目的端口、协议、包尺寸、源IP)，并且用相似度的方法综合考虑在蠕虫爆发的时候前三个属性的变化。与此同时，对于源IP，仍然采用前文所说的基本检测算法。

而对于出口流量，只考虑失败连接的次数。这是基于如下考虑：(1)从内网发往外网的失败连接本身就有问题；(2)相比正常流量而言，从内网发往外网的失败连接较少，在数据量不大的情况下，如果考虑变化率则计算过于烦琐。同时将固定的时间窗移动，进行动态分析，以提高判断的准确性。

这两种不同的计算，关键都是要进行数据包的协议分析。协议分析技术[42]是在传统模式匹配技术基础上发展起来的一种新的检测技术。其工作原理是依据在OSI 7层协议模型，数据包按照从上到下的层层协议封装的有序性，先对数据包进行网络层的协议识别，再进行组包还原，然后按照OSI的7层模型的顺序，从下到上逐层脱去每层的协议头，一直分析到应用层。

3.2.3 可疑属性数据库

对出入口双向流量进行两种不同的计算，其结果就是生成一个可疑属性数据库。可疑属性数据库以入口流量分析时使用的四个属性作为基本要素，构建四个表：可疑目的端口表、可疑协议表、可疑包长表、可疑源IP表，每个表记录对应属性队列中可疑的元素，及其权值和生存时间(Time-To-Live，下文简称TTL)，表中的元素以权值为基，降序排序。所谓权值，是指该元素可疑的概率；所谓TTL，是指该元素可能可疑的存在时间。

记录入口流量的四个属性后，将出口流量分析的两个属性转化为入口流量分析的属性，同样添加进可疑属性数据库。可疑属性数据库作为随后检测流量的依据，要

保证其数据的时效性，设定一个定时器，对其定时更新，修改TTL，将TTL为0的项删除。

3.3 基于多属性相似度的入口流量分析

在正常网络流量下，流量属性（目的端口、协议、包尺寸)的分布是一定的，因此前后时刻的属性相关性会比较高。在异常网络流量下，属性分布会被破坏，因此前后时刻的属性相关性会比较低。本文通过多属性相似度分析的方法来研究在不同的网络状况下多属性在时间上的相关性，并提出一个异常检测算法。

3.3.1 多属性相似度定义

定义1：相似度即相关系数，其定义为

$$\psi(X,Y)=\frac{Cov(X,Y)}{\sqrt{DX}\sqrt{DY}} \tag{3.1}$$

其中 $X:\{x_1,x_2,.....x_n \mid x_i \in R\}$ 和 $Y:\{y_1,y_2,.....y_n \mid y_i \in R\}$，是两个随机变量序列，$Cov(X,Y)$ 是随机变量X和Y的协方差，DX和DY分别是随机变量X和Y的方差[43]。

定义2：数据流属性空间是一个数据流属性的集合，记为

$$TCS=\{C_1,C_2,.....C_n\} \tag{3.2}$$

其中属性包括端口号、报文大小和协议类型等。由于属性的不同，其值域也不尽相同，例如 $C_{port}=\{c_i \mid c_i \in N\}$ 和 $C_{size}=\{c_i \mid c_i \in R\}$。

定义3：若有一个数据流属性 $C \in TCS$，在某一时刻 τ 有 $\{c_1,c_2,.....c_3\} \in \{C\}$，则数据流属性C在时刻 τ 的属性分布记为 $\{\tau_1,\tau_2,.....\tau_3\}$，其中 τ i为ci对应的流量占总流量的百分比。(总流量如何定义，是所有流量，还是top n 的所有流量)

定义4：若给定一个数据流属性C以及其在时刻 τ 和 τ' 上的属性分布，则数据流属性C 在时刻 τ' 相对时刻 τ 的单属性相似度为 $\psi(\tau,\tau')$，记为 $S_{\tau,\tau'}^{C}$。

定义5：对所有的数据流属性 $C_i \in TCS$，时刻 τ' 相对 τ 的多属性相似度为 $\sum_{i=1}^{n}\varepsilon^{c_i}\delta^{c_i}S_{\tau,\tau'}^{C_i}$，记为 $S_{\tau,\tau'}^{TCS}$。

3.3.2 多属性相似度分析

本文采集流量，汇集不同时刻的流量并将其按照某些流属性（如目的端口、包长和协议)进行统计。然后在多个数据流属性间计算统计数据的多相似度。通过前后时刻的多相似度的变化，反映网络实际状况的异常，从而发现未知的网络蠕虫。以 IP

地址为例，其计算过程大致如下：

(1) 在时刻 τ 按照所占流量的百分比从大到小列出前 n 个百分比，得到了时刻 τ 的属性分布 $\{\tau_1, \tau_2, ...\tau_3\}$，同时得到前 n 个百分比对应的 IP地址。

(2) 在时刻 τ' 列出在时刻 τ 出现的 IP 地址所对应的流量百分比。如果这个 IP 地址没有出现在时刻 τ' 的前 n 个中，则其对应的百分比为 0，因为可以认为其流量在时刻 τ' 太小可以合理地忽略。如果这个 IP 地址在 τ' 时刻出现，则其对应的百分比为 τ' 时刻的百分比。因此得到了时刻 τ' 的属性分布 $\{\tau_1', \tau_2', ...\tau_3'\}$。

(3) 按照定义 4 计算 IP 地址的时刻 τ' 相对时刻 τ 的单属性相似度 $S_{\tau,\tau'}^{IP} = \psi(\tau, \tau')$。

(4) 交换时刻 τ 和 τ' 的属性分布，重复步骤 1、2、3。即在步骤 1，列出在时刻 τ' 前 n 个百分比，得到一个新的属性分布。在步骤 2，在时刻 τ 列出在时刻 τ' 出现的 IP 地址所对应的流量百分比。计算时刻 τ 相对时刻 τ' 的单属性相似度 $S_{\tau',\tau}^{IP} = \psi(\tau', \tau)$。

$S_{\tau,\tau'}^{IP}$ 和 $S_{\tau',\tau}^{IP}$ 的值越接近 1，在时刻 τ 和 τ' 的 IP 分布就越相似，在时刻 τ 和 τ' 的关联性越大。因此通过不断地计算多属性相似度，就可以研究网络流的属性在时间上的相关性，找出可疑的属性，进而更新可疑属性数据库。

3.3.3 基于多属性相似度的分析算法

实际中多相似度上的两个变化可以帮助识别网络异常。一方面，当网络异常刚刚开始发生时，在多相似度曲线上会有一个很大的下降，如果这个下降超过预定的阈值，那么就可以认为发现并识别了异常。另一方面，当网络异常持续一段时间之后，网络属性分布再次稳定。多相似度曲线或者返回到原先的相似度水平上，或者变得再次稳定。多属性相似度的方法可有效应对慢速扫描与随机化扫描。

基本算法流程如下：

(1) 接收入口数据包，采用协议分析的方法，去掉数据帧头部，获取IP头部和TCP头部，从中提取目的端口、协议和应用层数据包长，源IP及其对应的目的IP；

(2) 对于目的端口、协议和数据包长，对于源IP，转到5)建立三个队列，在单位时间内，统计其个数，找去前N名，并将其百分比设定为其权值；不在前N名之内，则从队列中删除；

(3) 以该单位时间内的前N名的序列S1为基准，调整前一个采样的单位时间内 前N名流量序列S2的顺序与权值。调整原则为：在S1中出现而S2中没有的，以及在S2

中出现而S1中没有的，都将其权值设为0；

(4) 计算S1和S2序列的相似度，若相似度超过阀值，则认为出现异常。根据S1的排名，计算其权值，将S1序列加入可疑属性数据库，根据项目不同设定不同的生存时间值；

(5) 对于源IP，则以源IP为单位，统计每一个源IP在一个采样的单位时间内对应的不同目的IP的个数；当同一源IP对应的目的IP超过某一阀值时，则认为该源IP正对内网进行扫描，该源IP即为可疑源IP，将其加入可疑属性数据库。其依据是：正常情况下，一个IP只会与一个IP通信；如果一个IP与多个IP通信，可疑。

(6) 设定一个定时器，根据设定的时间，定时查看可疑属性数据库，修改不同项的生存时间值。

3.4 基于滑动窗口的出口流量分析

3.4.1 滑动窗口的定义

对于出口流量，主要检测失败连接。在一般的扫描算法中，都是基于一个固定的时间窗口，统计这个时间窗口的流量，而这种静态的固定时间窗口难以有效应对复杂多变的网络环境。此时可以参考TCP的滑动窗口。

TCP的滑动窗口协议是一种流量控制的协议[20]，在开始的传输控制阶段，确定传输的窗口值；数据传输时，接收端有一个接收窗口，大小固定，但不一定与发送窗口相同。接收窗口的上界表示允许接收的序号最大的帧，下界表示希望接收的帧。接收窗口表示允许接收的信息帧，落在窗口外的帧均被丢弃。序号等于下界的帧被正确接收，并产生一个响应帧，下界加1。接收窗口大小不变。

滑动窗口本质上是描述接受方的TCP数据报缓冲区大小的数据，发送方根据这个数据来计算自己最多能发送多长的数据。如果发送方收到接受方的窗口大小为0的TCP数据报，那么发送方将停止发送数据，等到接受方发送窗口大小不为0的数据报的到来。

原来的静态窗口每次统计一个单位时间内的流量，之后统计下一个单位时间内的流量，而这种方法失去了这两个相邻的单位时间的相关性。参展滑动窗口协议，将流量统计时的静态固定窗口改为滑动窗口，窗口大小设为固定的，而窗口的移动则改为小于单位时间的滑动。如图3.3所示，每隔一个单位时间，向前滑动一个窗口；而图3.4则是以更小的粒度向前滑动，这样，前一个单位时间的部分信息就可以保存到下一个统计中。

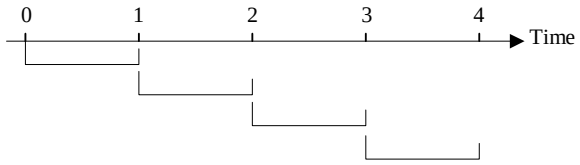

图3.3 静态窗口示意图

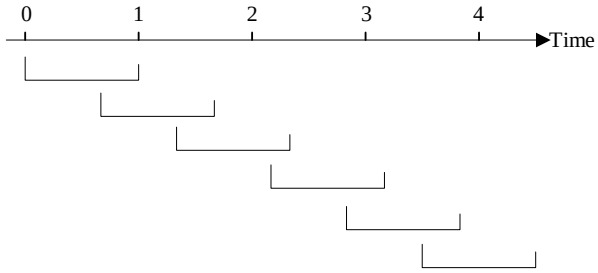

图3.4 滑动窗口示意图

3.4.2 基于滑动窗口的分析算法

基本算法流程如下：

(1) 接收入口数据包，采用协议分析的方法，获取失败连接的目的IP和源端口；将该IP和端口插入相应的对列；设定单位时间为 t ，滑动的时间值为 s，s＜ｔ；

(2) 统计目的IP和源端口在单位时间 t 内出现的次数，同时记录其出现的时间值；

(3) 若出现的次数超过阀值，该目的IP或该源端口可疑，计算其权值，写入可疑属性数据库；

(4) 经过时间ｓ后，向前移动窗口，删除队列中不在窗口中的值，重新开始统计；

(5) 设定一个定时器，根据设定的时间，定时查看可疑属性数据库，修改不同项的生存时间值；

在出口流量中检测的目的IP和源端口，可以认为是入口流量的源IP和目的端口。因此，此时在写入可疑属性数据库之前，进行一定的聚类分析与转化，将其加入到入口流量的源IP和目的端口中。加入的原则为：如果可疑属性数据库中不存在该项，直接加入，按照出口流量的要求，生成其TTL值。如果可疑属性数据库中存在该项，数据库中的权值加上新项的权值；按照出口流量的要求，生成该项其TTL值，然后与数据库中已有的权值进行比较，取二者中的较大的值。

3.5 未知蠕虫检测

3.5.1 流量分析与检测

一般异常检测都是假定所有蠕虫行为都是与正常行为不同的，继而建立系统正常行为的轨迹，然后把所有与正常轨迹不同的系统状态视为可疑企图，以发现蠕虫的攻击。其关键点是建立系统正常的行为规律，然后比较与正常行为不同的状态。而网络状态无时无刻不在动态的变化，之前认为是正常的流量状态，不一定适用于后期的情况；而网络的前后状态却是存在一定的关联，故本文采用的是多属性相似度的方法分析入口流量的前后状态的变化，同时考虑蠕虫的行为特点，有针对性的分析网络流量。

本文检测未知蠕虫的方式首先是进行出入口流量分析，分析的结果是生成了可疑属性数据库，这是进行未知蠕虫检测的参考依据。之后参照可疑数量数据库，检测网络流量，提取出可疑流量，导入可疑流量池，可疑流量池保存检测到的异常数据包，可认为是一个数据包的队列。

生成可疑属性数据库之后，在入口捕获网络流量时，提取出该数据包的源IP、目的端口、协议和包长，与可疑属性数据库进行匹配，计算其权值之和，如果权值之和超过阀值，则认为该数据包可疑，将其导入可疑流量池中，并记录导入的时间值。阀值的设定是关键的因素，决定了后续采用哪种特征码生成方法。

再设定一个定时器，指定有效时间值，当可疑流量池中的数据包超过了有效时间值后，将其删除。如此继续检测进入的数据包，将可疑的数据包导入可疑流量池。当可疑流量池中的数据包个数超出一定规模时，触发特征码自动生成模块。

3.5.2 定时器的使用

前文无论是构建可疑属性数据库，还是将流量导入可疑流量池，都提到了定时器，定时器的使用在本系统中是个重要因素，是系统能否成功运行的关键。

在可疑属性数据库中，需要定时修改TTL，其理论依据是：1)当已经检测出某种未知蠕虫后，其特征码加入特征码数据库，之后该蠕虫即为已知蠕虫，可以通过模式匹配来检测，而不需要进行可疑属性的分析；2)在分析可疑属性时，某些属性可能具有时效性，如源IP，一定时刻后，之前是蠕虫宿主机的源IP可能不再攻击内网系统，那么其IP不再可疑。

在导入可疑流量池时，对于导入的可疑流量也需定时删除失效者，其理论依据是蠕虫的传播只是在一定时间内发生的，类似于慢速扫描之类的蠕虫传播还没有出现。

4 蠕虫特征码的自动生成

综合分析目前典型的特征码自动生成技术，总体上有两个基本步骤：(1)获取可疑的数据；(2)对可疑数据进行分析和比较，从中提取出共性的、不变的成分，形成攻击的特征。本章主要讨论第二步。

在网络层检测蠕虫，根据网络流量中的可疑数据包自动生成蠕虫特征码。通过对路由器出入口双向流量的分析，检测出到可疑的网络流量后，将其导入可疑流量池，以待生成特征码。蠕虫在爆发时会在网络中产生大量内容相同或相似的数据包，这是蠕虫特征码生成的理论基础。

之所以在检测出未知蠕虫的目的端口、协议和包长等属性后，依然要生成蠕虫的特征码，是为了更准确的描述未知蠕虫，这是基于以下考虑：(1) 源IP只是一定时间范围内有效；(2) 前期检测到的可能是蠕虫扫描包，其与爆发时的数据包可能有所不同；(3) 在目前网络蠕虫变种频繁的情况下，同一类蠕虫的目的端口可能发生改变，采用的协议可能改变，数据包的长度更可能改变，而其核心的功能代码变更的可能性很小。因此为了更好的检测未知的网络蠕虫，仍然要快速的自动生成未知蠕虫的特征码。

4.1 特征码生成的方法及其种类

4.1.1 特征码生成的两种方法

有两种方法可以生成蠕虫特征代码[44]：(1)由专家人工生成，这种方法准确率高，误报率低，然而这种方法的反映速度慢，待其生成特征代码之后，往往蠕虫已经泛滥开来；(2)自动生成，这种方法实时性较好，发现蠕虫的同时很快就可生成特征代码，便于防御抑制蠕虫，但是其准确率不尽如人意，然而在机器学习的能力越来越强的情况下，自动生成特征代码也成为一种选择。

特征码自动生成技术主要研究在没有人工帮助的情况下，如何快速准确地检测新攻击并自动生成出其特征。从2003年开始，攻击特征自动生成得到越来越多研究者的关注，已经成为蠕虫研究的一个新热点[45]。

4.1.2 特征码的两种类别

(1) 单一字符串特征码

现阶段，蠕虫特征代码多是单一的字符串。该字符串是从蠕虫样本中提取的固定不变的字符串，可以作为蠕虫的特征，其有效长度可很好的匹配蠕虫，而不能匹配

正常流量；字符串越短，误报的可能性越大；字符串越长，匹配消耗的系统资源越多，速度越慢。

单一字符串特征代码是基于这样的假设：蠕虫在攻击过程中，存在一个单一的负载子串，其内容保持不变，且该字符串对于蠕虫是独一无二的。该假设是基本成立的，一个蠕虫的基本攻击代码是不变的。

(2) 多重子串特征码

然而，检测未知蠕虫时，蠕虫的特征代码难以有效的用单一的字符串确定，但是会存在多个模糊的特征。同时，对于多态蠕虫而言，其传播时，会改变其负载；而负载改变了，则单一字符串特征代码存在的基本假设不成立，难以有效应对多态蠕虫。因此提出了多重子串特征代码。

该特征代码是一个由多个字符串组成的集合，集合的元素为较短的字符串，其长度一般要小于单一字符串特征代码（为与之前的单一字符串特征代码相区分，将集合中的元素称之为令牌，同时保证令牌的独特性，即一个令牌不会是另一个令牌的子串)，同时每个令牌可以取不同的权值。令牌的个数越多，每个令牌的长度越长，匹配时就越困难，漏报率的可能性也就越高；令牌的个数越少，每个令牌的长度越短，在提高匹配速度的同时，也提高了误报率；问题的关键是在令牌的个数与长度间取得平衡，同时为每个令牌赋以合适的权值。据此将多重子串特征码分为如下两类：

a 最多令牌特征码（模糊特征码)

该类特征码由尽可能多的令牌组成，根据每个令牌在样本集中出现的概率，赋予每个令牌以一定的权值，同时设定一个阀指。数据包与特征码匹配时，对出现在数据包中的所有令牌，计算其权值的总和，将其与指定的阀值比较。如果权值总和超过阀值，则认为该数据包为网络蠕虫。不要求特征码集合中所有令牌都出现在蠕虫数据包中，这样就可以尽可能的丰富特征码令牌集，因此把最多令牌特征码称为模糊特征码。

b 最少令牌特征码（精确特征码)

与上述特征码相反，该类特征码在与数据包相匹配时，要求特征码中的所有令牌都出现在数据包中，即可以认为每个令牌中的权值为1，因此也把这类特征码叫做精确特征码。为达到这个要求，同时将漏报率控制在一个可承受的范围内，特征码中的令牌个数必须尽可能的少，而令牌也应以短为佳。

4.2 最多令牌特征码

4.2.1 令牌的提取

生成特征码的关键是令牌的提取。本类特征码在最大公共子串的基础上，加以改进采用最多公共字串匹配法来提取令牌。

最大公共子串字符串是处理中目前最多的字符串算法，这种方法求两个字符串的最长公共子串，其结果为一个子串。本文提出的最多公共子串不要求是最长的公共字串，只要是长度满足一定要求的公共子串就行。这里定义为子串的长度为Lsize，最短的长度为Lmin ，Lsize >= Lmin，这里Lmin=3。

定义两个字符串为a[M]，b[N]，临时公共子串为temp[]，其提取算法的伪码表示为：

```
for(i=0; i + Lmin -1 <M;i++)
{
temp=a[i, i + Lmin -1]
    If(temp  在b[N]  中匹配)
    {
        继续寻找，以获得更长的公共子串
    }
    else //  不匹配
    {
        进入下一轮循环
    }
}
```

该算法只找出尽可能多的公共子串，而不考虑子串所处的位置。提取的公共子串即为令牌。

4.2.2 令牌集的构建

令牌提取之后，就是如何用令牌构成令牌集，本特征码采用三轮循环的方式构建令牌集。生成令牌集之后，从生成的令牌集剔除不符合定义的令牌，去除不符合要求的令牌，如此就生成了蠕虫特征码。三轮循环法的理论基础是：蠕虫扫描攻击数据包的相似性。

在有n个样本的可疑流量池中（n为4的倍数)，对这n个数据包，以数据包长度由大到小排序为{L1，L2，L3，.....，Ln}。按长度为序的原因是长度相似的数据包为同一类蠕虫的可能性很大。再将n个数据包划为4组P1、P2、P3、P4，每组n/4个。

第一轮循环，选取两个数据包{Li，Li+1}1≤i≤n-1，i=i+2，采用最多公共子串算法，提取这两个数据包的公共子串集H1i={m1，m2，.....，ms}。如此循环计算n/2次之后，将{H1i}合并为H1={s1，s2，.....，sm}。依次统计公共子串si出现的次数mi，如果mi<Min(设定的最小阀指，2<Min<n-2)，那么将si从公共子串集H1中删除；否则保留。如此构建完备的第一轮循环的公共子串集H1={x1，x2，.....，xm}。

第二轮循环，P1和P2归并计算，P3和P4归并计算，选取两个数据包P1i和P2i、P3i和P4i，其中1≤i≤n/4、i=i+1，生成公共子串集H2i，并将{H2i}合并整理为H2。

第三轮循环，P1和P2归并计算，P3和P4归并计算，选取两个数据包P1i和P3i、P2i和P4i，其中1≤i≤n/4、i=i+1，生成公共子串集H3i，并将{H3i}1≤i≤n-3合并整理为H3。

在三轮循环的过程中，提取的公共子串的长度应该限制在一个范围内，即MinLength < Length(xi) < MaxLength，可以确定的是MinLength>2，而MaxLength须经过统计分析才能确定。

前文提过，可疑流量数据库中数据包个数为4的倍数，这里为简化起见，设n为8，其算法运行过程如图4.1所示：

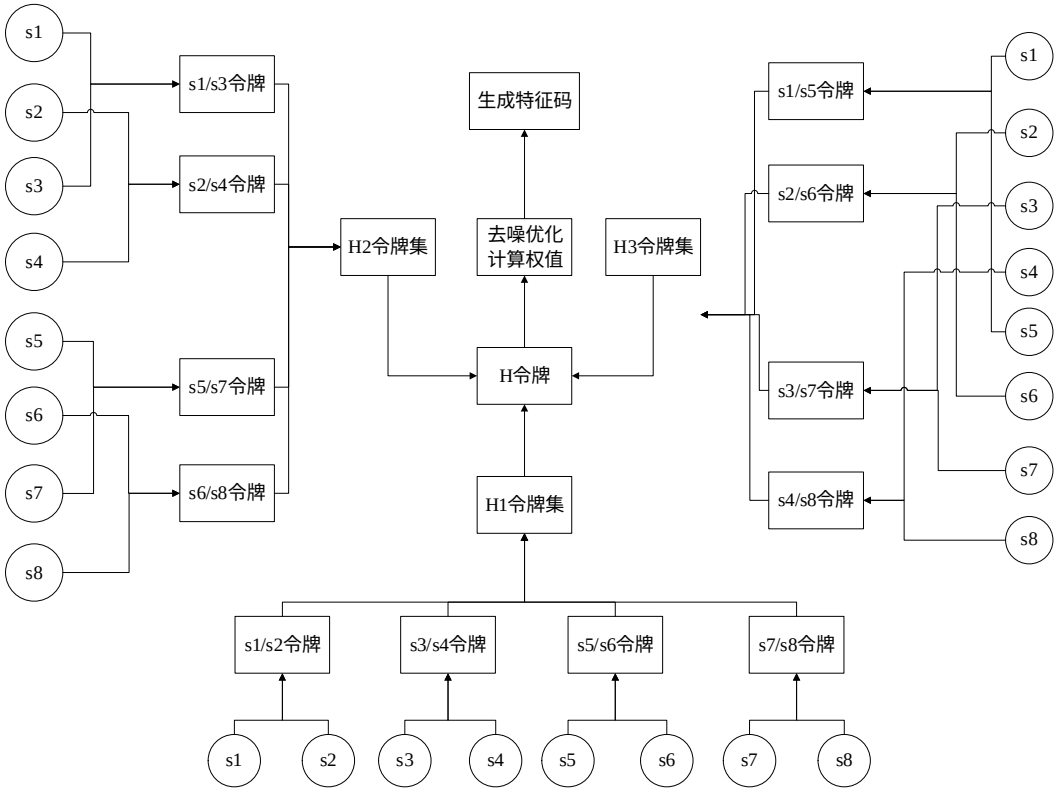

图 4.1 最多令牌特征码生成算法示意图

三轮循环之后，再将H1、H2和H3合并整理为H＝{h1，h2，.....hg}，到此已构建完整的令牌集。

4.2.3 特征码的生成

首先，将根据误报率与漏报率的要求，去除令牌集中不合格的令牌。计算hi在可疑流量池中出现的概率pi，以及在正常流量池中出现的概率qi：

$$p_i = \frac{h_i\text{在可疑样本中出现的次数}}{\text{可疑流量池中样本总数}n} \tag{4.1}$$

$$q_i = \frac{h_i\text{在正常样本中出现的次数}}{\text{正常流量池中样本总数}m} \tag{4.2}$$

如果hi在同一个样本中出现多次，就重复计数。

再计算Vi=pi-qi，如果Vi≤k(k为设定的阀值，一般取0)，将对应的Hi删除。最终得出一个令牌集S＝{S1，S2，.....，Sm}，该令牌集对应的权值V＝{V1，V2，.....，Vm}， S以令牌权值的降序排序。

其次根据独特性的描述（每个字符串都不会是另一个字符串的子串)，将H整理为符合独特性的公共子串集。其算法基本思路为：将H中的令牌按照长度升序排序，之后，参照冒泡排序的思想，依次进行字符串匹配，若是长度较短的令牌可以是长度较长令牌的子串，则删除其一，删除原则：将两个令牌的长度乘以其权值，得出两个乘积，删除乘积较小的令牌。如此循环，直到令牌集中的令牌都符合独特性。

确定最终的令牌集和对应的权值之后，构建特征码T={(S1， V1)，(S2，V2)，……，(Sm ，Vm)}。T即为特征码，在与数据包相匹配时，Vresult为特征码中出现在数据包中的所有令牌权值相加的结果，Vmin为设定的阀指，当Vresult>=Vmin时，即判断数据包为蠕虫数据包。

4.3 最少令牌特征码

4.3.1 剔除流量噪声

对于以精确匹配为要求的最少令牌特征码而言，混进可疑流量池中的正常流量无疑是影响精度的噪声。因此，在提取令牌之前，应剔除可疑流量池中的噪音。

根据流量四个属性的综合判断，把网络数据包中的异常流量导入可疑流量池中，再将这四个属性分开分析，可以作为剔除噪声的基准。根据四个属性的重要性，依次取端口、协议和包长，进行去噪处理。

去噪过程中，基于以下原因不考虑源IP：(1) 端口、协议和包长都可以转化为数字，可以采用概率统计的方法进行分析，而IP地址是一个32位的数，难以有效转换

为数字，计算较复杂；(2) 蠕虫可能会在不同的源对内网攻击，IP地址作为网络流量的属性之一，允许存在离散分布性。

对于端口、协议和包长，都可以采用数学分析的方法，将其转化为有序序列，之后采用概率统计的方法，去除噪声。如果在某一属性上，剔除了足够多的噪声；或者认为所有流量都是可疑的、没有噪声，则余下属性，就不必分析。

4.3.2 令牌的提取

本类特征码采用字符串对齐的方法，来提取令牌。

字符串对齐技术是指，在两个字符串中找到其最大的公共字符串集。比如说，在字符串“ XXhelloXXXXXXok ”和“ YhelloYYYokYYYY ”中，最大的公共字符串集就是“ hellook ”。就是说，给定两个字符串，以某种方式对齐，使之可以最大化公共字符串集，作为前例，其对齐方式为：

X X h e l l o X X X X X X o k -- -- -- --

Y -- h e l l o Y Y Y -- -- -- o k Y Y Y Y

使用常规表达式，这中对齐就表示为：“ *hello*ok* ”。

但是，最长子串的匹配可能并不能最优化顺序的匹配。比如说，考虑字符串“ oXnXeXzXtwoX ”和“ YtwoYoYnYeYz ”。最长子串的匹配是：

-- - -- -- -- o X n X e X z X t w o X

Y t w o Y o Y n Y e Y z -- - -- -- --

该特征码就是“ *o*n*e*z ”。但是，此例中，还可以如此匹配以生成特征代码“ *two* ”：

o X n X e X z X t w o X - - - - - - -

- -- -- -- -- -- - Y t w o Y o Y n Y e Y z

如上所示，第二种对齐方式生成了更短的子串，而这个子串中字节都是邻近的，因而产生的特征代码更好。因此就需要一种字符串对齐的算法，以生成字节邻近的字符串特征码。

本文将每种对齐方式赋予一个分值，分值的计算是这样的：对齐后生成的字符串中每个字符为1分，任意两个邻近的字符之间如有“ * ”，则减去一个Wg。本例将Wg赋值为0.8。那么，第一种对齐方式生成的特征代码“ *o*n*e*z ”，其分值为：4-3*0.8＝1.6；第二种对齐方式生成的特征代码“ *two* ”，其分值为：3-0*0.8＝3。相比而言，第二种对齐方式的分值更高，产生的特征代码更好。

4.3.3 令牌集的构建

采用字符串对齐技术生成的令牌有位置关系，令牌提取之后，就是如何用令牌构

成令牌集，再从令牌集中选取合适的令牌生成特征码。

本类特征码采用迭代法计算令牌集，不同于前文所述三轮循环法。其理论基础也是蠕虫扫描攻击数据包中核心代码的不变性。

在提出噪声之后的可疑流量池中，将n个数据包，以长度由大到小排序为{L1，L2，L3，.....，Ln}，此时有两种计算模式，完全计算模式和增量计算模式。

完全计算模式是在已完全获取n个报文时，或者说可疑流量池中采集到一定数量的报文后作统一分析，其基本过程如图4.2所示。举例说明：假设从L1和L2中通过字符串对齐得到令牌S(1，2)“ *two*one* ”，从L3和L3中得到令牌S(3，4)“ *woo*nello* ”，那么把“ * ”当作普通字符，继续执行字符串对齐，则得到令牌S(1，2，3，4)“ *wo*ne* ”，如此迭代计算，得出最终的令牌集。

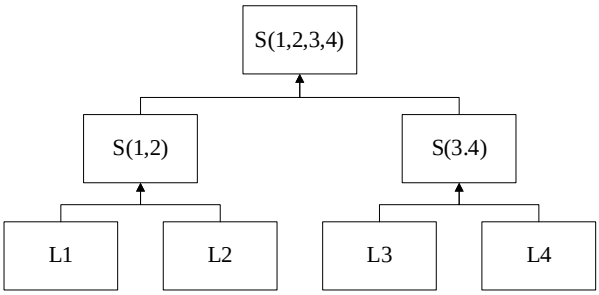

图4.2 完全计算算法流程图

增量计算是当逐步收到可疑数据包时，求解令牌集，即当收到两个数据包时，就求出令牌S(1，2)；收到下一个数据包时，将上次求得的令牌当作字符串，与数据包相对齐，求出新的令牌集。如此迭代，当不再接收数据包时，就得出了最终的令牌集，如图4.3所示。比较对齐中采用的数据包个数越多，产生的令牌就越少，每个令牌就越小。

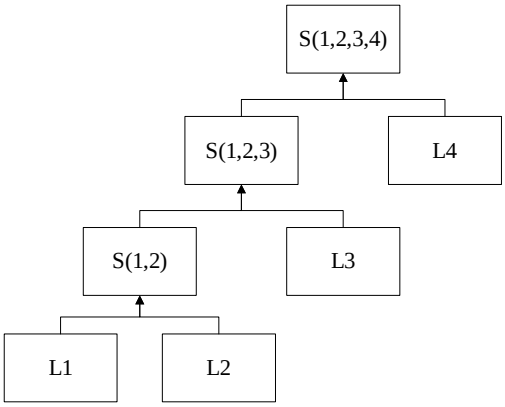

图 4.3 增量计算算法流程图

根据这两种方法生成的令牌集即为蠕虫的特征码。

4.3.4 特征码的转换

本文所研究的系统通过网络流量分接器获取网络流量，对网络流量本身不产生任何影响，所生成的特征码写入特征库，系统本身并不能防范蠕虫，阻断蠕虫攻击，还必须与其它系统——如IDS，firewall，IPS等等——联动，特征库必须转换成这些系统可以理解的规则，而目前常用的规则即为Snort规则。

Snort是一个开放源码的、强大的轻量级的网络入侵检测系统。它具有实时数据流量分析和日志IP网络数据包的能力，能够进行协议分析，对内容进行搜索/匹配。它能够检测各种不同的攻击方式，对攻击进行实时报警。此外，Snort具有很好的扩展性和可移植性。Snort使用一种简单的规则描述语言，这种描述语言易于扩展，功能也比较强大。由于它的可扩展性，使得它备受欢迎。如下所示，为一个简单的Snort规则。

alert tcp any any -> 192.168.1.0/24 111(content："|00 01 86 a5|"；msg："mountd access"；)

最少令牌特征码采用字符串对齐算法，考虑了子串位置信息，从而反映特征片段之间的相对位置关系。而snort的规则选项“ offset ”和“ distance ”恰好可以准确地表达此关系。而在蠕虫检测阶段发现可疑流量所生成的四个属性，除包长之外，都可以应用。特征码转换为可用的Snort规则，应用到IDS，IPS中。

而最多令牌采用的算法只考虑共同子串，不考虑子串位置关系，表达方式不同与最少令牌特征码，使其难以转换为Snort规则。

4.4 两种特征码的区别

(1) 噪声的处理

这里的噪声可以指混进可疑流量池中的正常数据流，也可以指因为可疑流量池中正常流量的存在而导致的无效令牌。最少令牌特征码采用前者，而最多令牌特征码则认可后一种理解。所以最少令牌特征码才令牌提取之前，进行噪声处理，通过分析可疑流量池的四个属性，剔除噪声。而最大特征码则是在生成令牌集之后、特征码生成之前，采用概率的方式，剔除无效的令牌。

(2) 令牌提取的算法

二者的提取算法也不同，最多令牌特征码采用的是最多公共子串算法，将出现次数较多的公共子串作为令牌；而最少令牌特征码采用的是字符串对齐算法来提取令牌。前者的令牌仅仅是字符串、与权值关联，而后者的令牌与其在数据包中的位置相关。前者的令牌最小值为3，后者的令牌最小值为2。

(3) 令牌集构建的方法

最多令牌特征码采用的是三轮循环法构建令牌集，从算法示意图来看，不管可疑流量的多少，都只有四个层级；而最少令牌特征码的层级与可疑流量的个数直接相关。可疑流量越多，前者中令牌的个数越多，后者中令牌的个数越少。

(4) 检测可疑流量的阀值

最多令牌特征码不关心令牌个数的多少，而最少令牌特征码的令牌以少为佳，所以在导入可疑流量池之前、设定阀值时，前者设定的阀值较小，以便导入更多的可疑流量；而后者设定的阀值较大，只将更为可疑的流量导入可疑流量池中。

(5) 特征码的结构

最多令牌特征码中的特征码中的构成元素为(令牌，权值)等，而最少令牌特征码中的构成元素则为令牌。前者的令牌比较短长，个数较多，而后者的令牌比较小，个数较少。前者中元素的排列以权值为基，降序排序；后者以令牌在蠕虫数据包中出现的顺序为序。

(6) 匹配的方式

二者匹配的方式不同，最多令牌特征码是按照令牌的权值、将数据包尽可能多的匹配其中的令牌，而一旦其权值之和超过阀值，则认为数据包为蠕虫；最少令牌特征码则是需要完全的匹配特征码中的所有令牌。给前者为模糊匹配，后者为精确匹配。

5 系统原型设计

5.1 系统流程分析

系统主要有两大功能：未知蠕虫检测和特征码自动生成。这两个功能是相关联的，未知蠕虫检测的结果是生成可疑流量多特征库，并捕获可疑流量将其导入可疑流量池。特征码自动生成的数据来源是可疑流量池，生成特征后，将其转化为Snort规则时，还必须参考可疑流量多特征库。

系统在网关设备上运行，同时生成的特征码要转换为Snort规则，与防火墙等设备进行交互，所以系统运行环境为Linux操作系统。

其基本模块结构如图5.1所示，需要说明的是，在流量异常发现部分，出入口流量分析都需要进行；而在特征码生成部分，最多令牌特征码和最少令牌特征码只需选择一个就行。

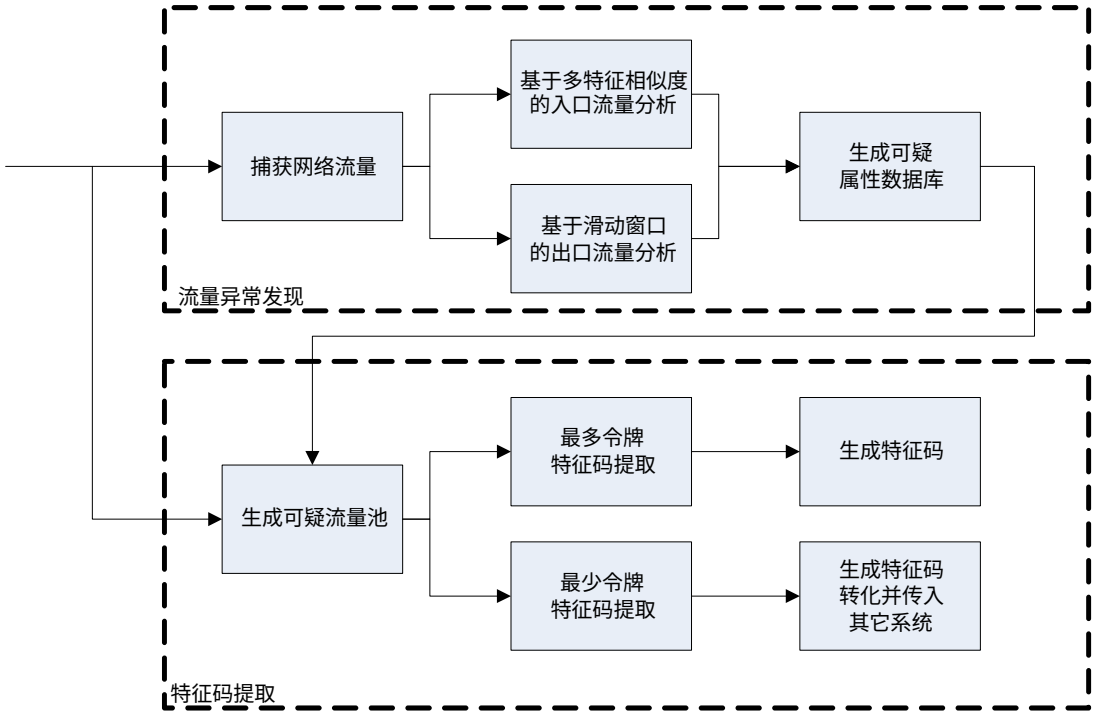

图 5.1 系统模块结构图

系统的运行是首先进行出入口流量分析，根据分析的结果构建可疑数据库，然后再根据可疑数据库来检测网络流量，从中提取出可疑的未知网络蠕虫，将其导入可疑流量池，然后从中生成特征码。系统流程如图5.2所示。

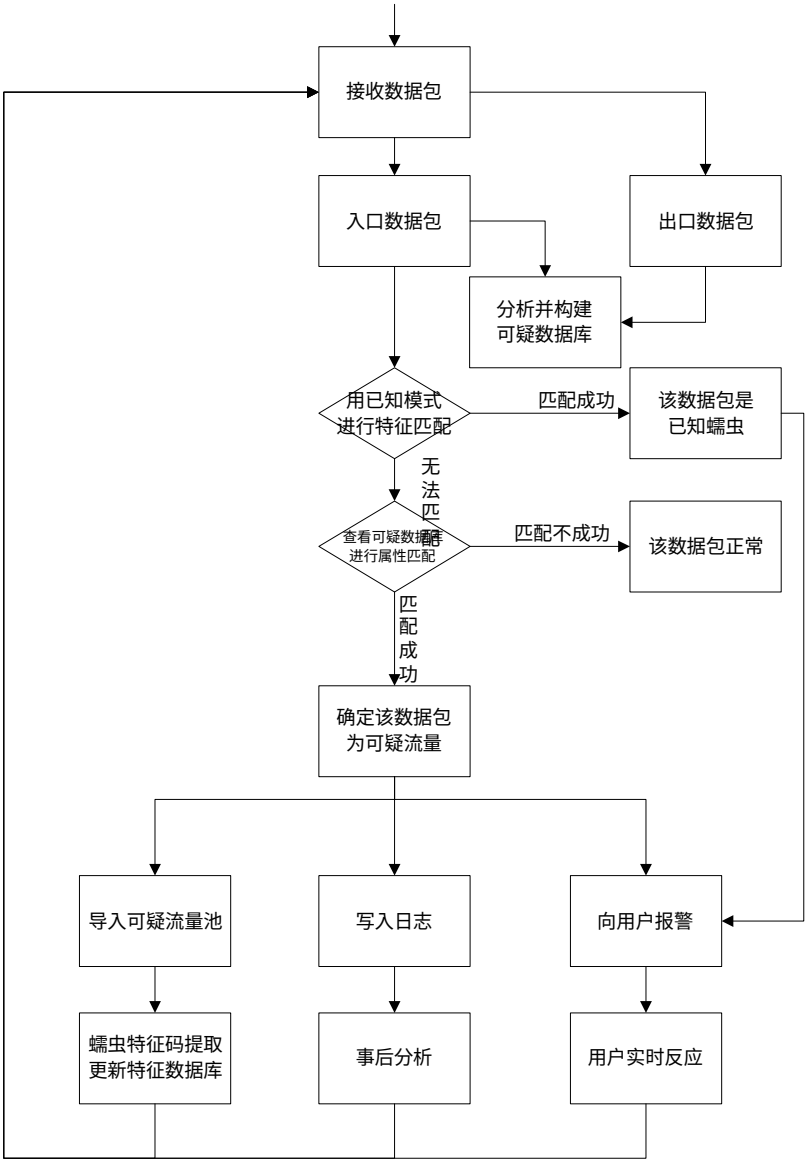

图 5.2 系统流程图

5.2 模块结构划分

根据系统功能分析及流程，可将系统划分为如下几个主要模块。

5.2.1 入口流量分析模块

从执行流程上看，首先在流量采集阶段分别对目的端口、包尺寸和协议这三个属性进行统计，找出流量排名前N的对应项；其次与前一时刻的排名相比较，分别计算它们的相似度；然后如果相似度的值超过指定的阀值，那么判定该次排名中存在可疑的流量属性，更新可疑属性数据库。而对于源IP，只考虑与其对应的目的IP的个数，

当一个源IP对应的目的IP过多，失去了对称性，即认为其该IP可疑。其过程如图5.3所示。

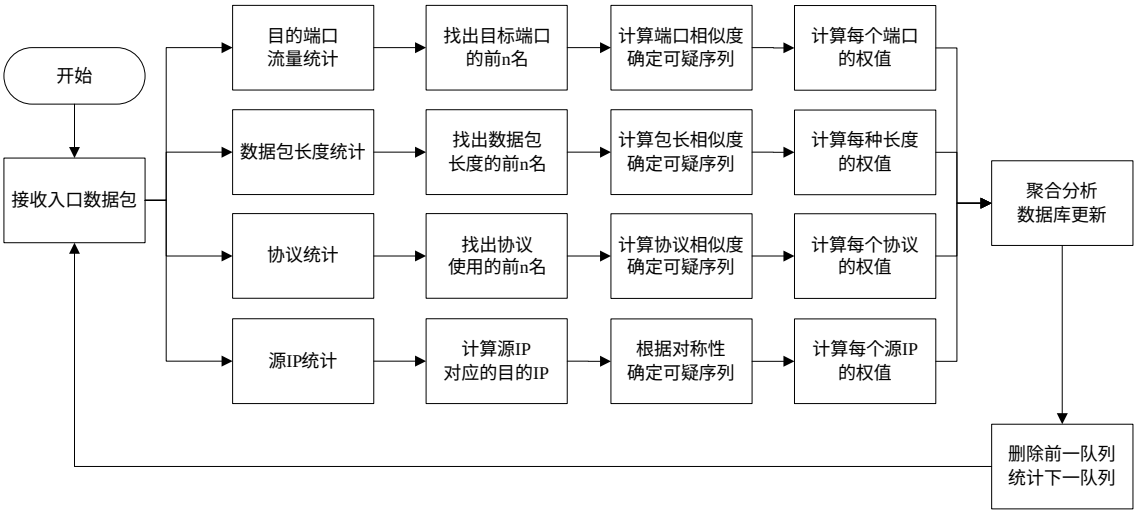

图 5.3 入口流量执行流程图

5.2.2 出口流量分析模块

出口流量模块主要统计失败连接，如前所述，在蠕虫传染时，一台被蠕虫感染的机器攻击内网主机时，会持续不断的产生大量连接失败信息，正是这大量的失败连接造成了ICMP和TCP协议状态的异常。对失败连接的统计分析即可检测出异常的数据通信。出口流量统计模块执行流程图如图5.4所示。

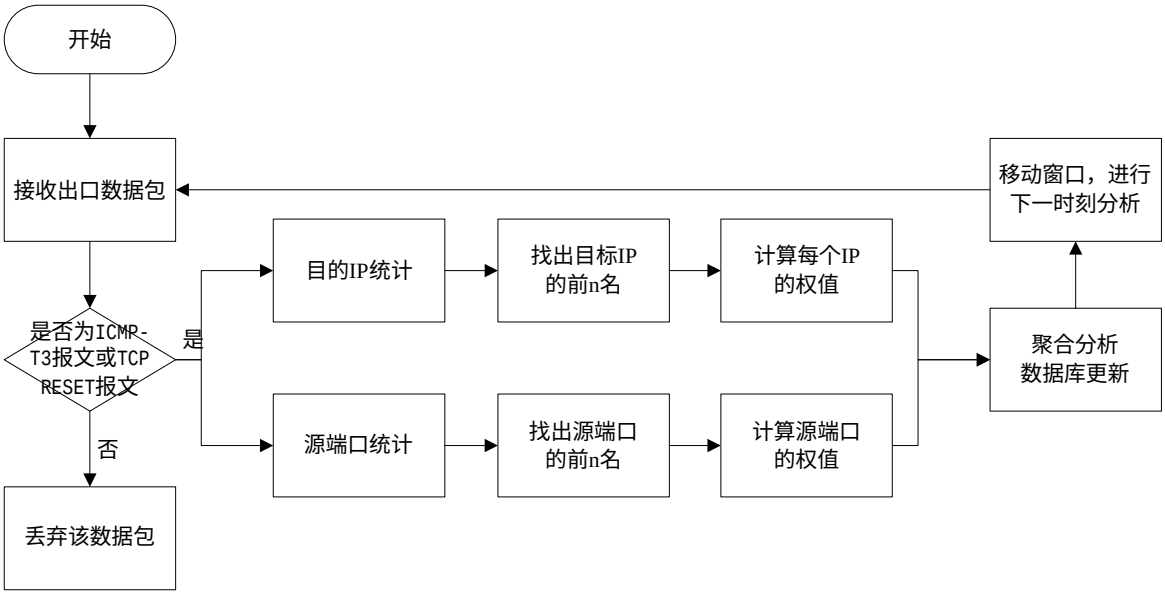

图 5.4 出口流量流程图

5.2.3 最多令牌特征码生成模块

对于可疑流量池中的可疑数据包，以包的长度为序降序排列；之后以三轮循环法、按照最多公共子串算法提取令牌，构建令牌集；之后根据独特性描述和令牌出现的概率，剔除无效令牌，生成特征码。其运行过程如图5.5所示。

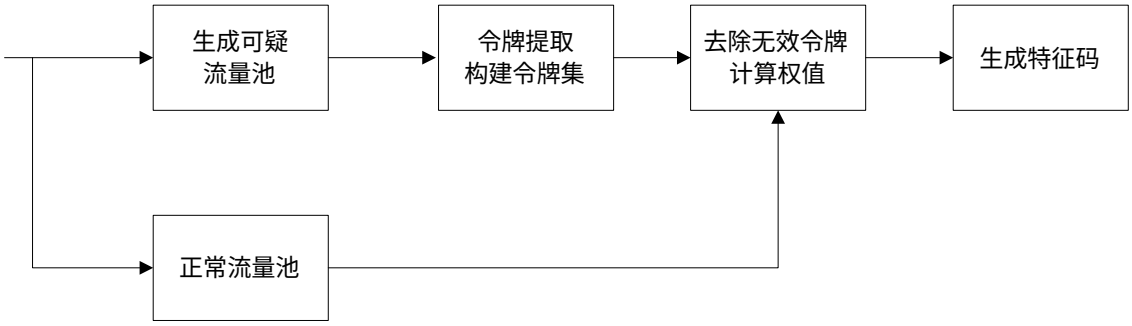

图 5.5 最多令牌特征码流程图

5.2.4 最少令牌特征码生成模块

最少令牌特征码与最多令牌特征码不同，在生成令牌之前，首先经过聚类分析、去除可疑流量中的噪声；之后在干净的流量中提取令牌，构建令牌集，生成特征码，然后根据联动系统的要求，将其转化成Snort规则。系统流程如图5.6所示。

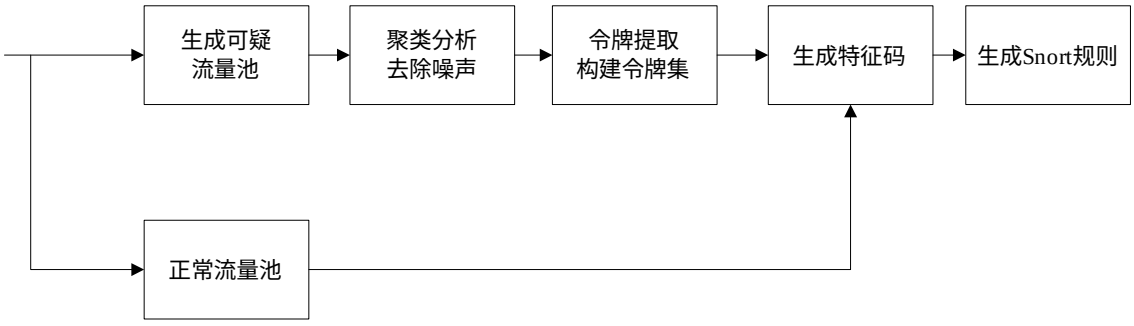

图 5.6 最少令牌特征码流程图

5.3 系统接口说明

5.3.1 出入口双向流量属性队列的格式定义

入口流量分析采用四个属性：目的端口、协议、包长和源IP。四个属性构建四个队列，其定义如下：

(1) 目的端口队列

```
struct Dport
{
    int dportID；              // 端口号
    int times；                // 端口出现的次数
    struct Dport *next；       // 下一个端口
}
```

(2) 协议队列

```
struct Protocol
{
    int proID；                // 协议号
    int times；                // 该类协议出现的次数
    struct Protocol *next；    // 下一个协议号
}
```

(3) 包长队列

```
struct PacketSize
{
    int sizeID；               // 包长，为便于计算，将包长划分为多个区间
    int times；                // 该类包长出现的次数
    struct PacketSize *next；// 下一个包长
}
```

(4) 源IP队列

```
struct SIP
{
    in_addr sipID；                // 源IP号
    struct DIP
    {
        in_addr dipID；            // 目的IP号
        int times；                // 该IP出现的次数
        struct DIP *next；         //该源IP对应的下一个目的IP
    }
    int times；                    // 为struct DIP队列中所有times之和
    struct SIP *next；             // 下一个IP
}
```

出口流量检测只考虑两个属性：目的IP和源端口，分别构建两个队列，其定义如

下：

(1) 目的IP队列：

```
struct DIP
{
    in_addr dipID；          // 目的IP号
    int times；              // 目的IP出现的次数
    struct DIP *next；       // 下一个IP
}
```

(2) 源端口队列：

```
struct Sport
{
    int sportID；            // 端口号
    int times；              // 端口出现的次数
    struct Sport *next；     // 下一个端口
}
```

5.3.2 可疑属性数据库的格式

出入口双向流量检测中，共提到了六个属性，这六个属性在检测蠕虫数据包时各有不同的权值，其权值根据这六个属性的重要性来确定，如下所示：

$$W_{SIP} < W_{Dport} = W_{PacketSize} = W_{Protocol} < W_{DIP} = W_{Sport} < 1 \quad (1)$$

$$W_{Dport} + W_{PacketSize} + W_{Protocol} + W_{SIP} + W_{Sport} + W_{DIP} = 1 \quad (2)$$

而这六个属性中，Dport、PacketSize 、Protocol和SIP应用于入口流量检测，Sport和DIP应用于出口流量检测，DIP可以理解为SIP，Sport可以理解为Dport。

根据以上设定，如表5.1定义这些值。

表5.1 属性权值定义初值

属性项	目的端口	包长	协议	源 IP	源端口	目的 IP
权值	0.18	0.15	0.15	0.12	0.20	0.20

将上述定义进行归纳，其最终定义如表5.2。

表5.2 属性权值定义终值

属性项	包长	协议	源 IP	目的端口
权值	0.15	0.15	0.32	0.38

再为每一个属性项设定其单独的可疑属性数据库，如表5.3。

表5.3 可疑属性数据库定义

序号	SIP	权值	TTL
1			
2			
3			
⋮			
n			

其中TTL以时间为单位，权值都小于1。

其它三个属性也如此表，表的更新原则为：

(1) 设定定时器，定时查看TTL值，时间到，则删除该项；

(2) 当一个可疑可疑序列到达时，按照该次可疑序列的排名，计算其权值与TTL；表中的权值与TTL都是在检测流量、确定流量可疑时计算确定，其它时刻不再更改；

(3) 当下一个可疑序列到达时，添加新的值，如果新项在表中已经存在，则比较新项与旧项中权值与TTL，并二者之中较大者更新数据库；

5.3.3 可疑流量池的格式

可疑流量池存储可疑流量序列，使用队列来实现。

```
struct Traffic
{
    String   packet;          // 数据包
    Time     inittime;        // 导入流量池的时间
    struct   Traffic  *next;  // 下一个数据包
}
```

5.3.4 最多令牌特征码数据库的格式

最多令牌特征码数据库采用格式，如表5.4。

表5.4 最多令牌特征码数据库格式

序号	生成时间	目的端口	包长	协议	（令牌，权值）
1	20070506 1438	1434	370	17(UDP)	(68 2E 64 6C,0.8),(65 6C 33,0.6),(6B 65 72,0.5)
2	20070510 0225	445	110	6(TCP)	(37 58 34,0.7),(66 25 34 37,0.6),(34 25 441,0.2)

对于表5.4，说明如下：生成时间记录生成蠕虫特征码的时间，时间格式为以数字形式记录当前的年月日时分，即200705061438表示2007年5月6日14点38分；包长为范围值，370表示该蠕虫数据包的包长在370~379之间；协议从IP包头部提取。

5.3.5 最少令牌特征码数据库的格式

最少令牌特征码数据库采用格式，如表5.5。

表5.5 最少令牌特征码数据库格式

序号	生成时间	目的端口	包长	协议	令牌
1	200705120935	80	300	6(TCP)	68 2E 64, 65 6C, 38 29
2	200705121234	139	120	17(UDP)	25 21 65,68 68 69, 34 24

对于该表的说明同5.4表，而令牌的格式二者不同。

6 系统测试

6.1 未知蠕虫检测的测试

6.1.1 测试环境

前文提过，系统的运行测试环境为Linux，数据库系统采用开源数据库MySQL。测试使用的程序为模拟蠕虫程序，根据前文讨论的蠕虫特征，模拟蠕虫的扫描过程，扫描方式为Hit-list扫描，采用TCP方式，设置内网交换机，允许发送出口的失败连接数据包。在此基础上，根据系统的要求，构建如图6.1的封闭测试环境：

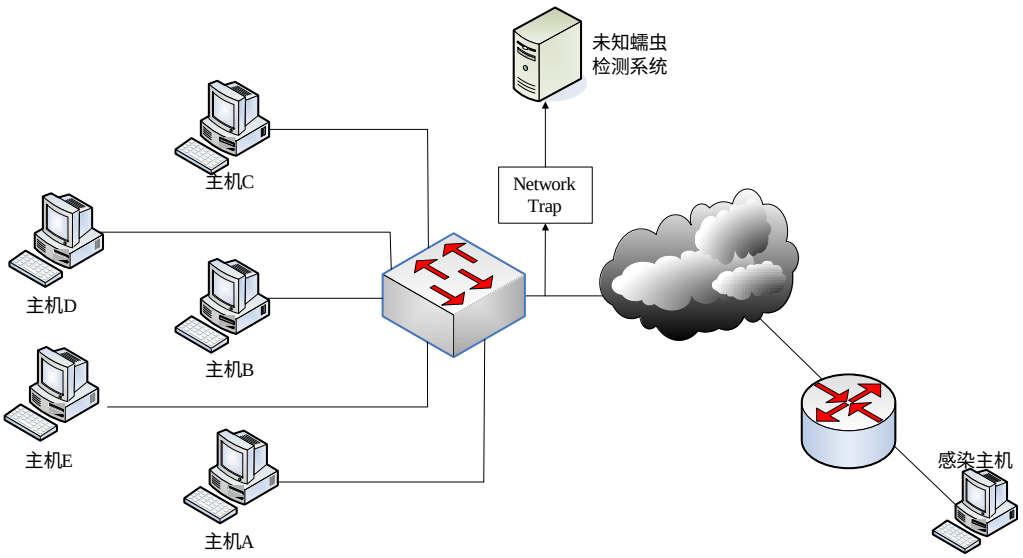

图6.1 系统测试环境

6.1.2 参数定义

本系统采用出入口流量双向分析，构建可疑属性数据库的方法来检测未知蠕虫。

(1) 入口流量分析相关参数

N：目的端口、协议和包长的属性队列的统计前N名，N设为10；

f_{SIP}：源IP统计的阀值，设为5，即当一个源IP在一定时间内与超过5个的目的IP通信，即判断该IP可疑；

T_{in}：采样单位时间，设为10分钟，即每隔十分钟进行一次数据统计，并刷新统计系列。

(2) 出口流量分析相关参数

T_{out}：统计单位时间，设为12分钟；

T_{move}：滑动时间，设为4分钟；

f_{DIP}：统计目的IP个数的阀值，设为5；

f_{port}：统计源端口个数的阀值，设为5；

(3) 可疑数据库相关参数

TTLip：IP的存活时间，设为30分钟；

TTLport：port的存活时间，设为60分钟；

TTLprotocol：protocol的存活时间，设为60分钟；

TTLpacketsize：packetsize的存活时间，设为30分钟；

Ttimer：定时器，每隔一定时间对数据库中的TTL值更新，并删除过期者，设为10分钟；

W_{SIP}: 源IP的权值，设为0.32；

W_{Dport}: 目的端口的权值，设为0.38；

$W_{PacketSize}$: 数据包包长的权值，设为0.15；

$W_{Protocol}$: 数据包协议的权值，设为0.15。

6.1.3 测试步骤

首先，启动系统，通过网络分接器开始捕获网络流量并进行分析。然后在感染主机上运行模拟蠕虫程序，即模拟程序只是模拟蠕虫的扫描过程，对于蠕虫的传播过程暂时不予考虑。

模拟蠕虫程序以不同的扫描强度、不同的扫描时间、不同的间隔时间对内部网络进行扫描，即以每秒钟发送m个扫描数据包，每次扫描n分钟，n分钟之后停止扫描，再过k分钟后又重新开始扫描。同时对每个扫描数据包进行标记，以便后期统计分析，每次共扫描30分钟。每次扫描之前先清空可疑属性数据库，以便之前测试的结果不会影响本次测试。

6.1.4 测试结果

当模拟蠕虫以不同的扫描方式对内网主机进行扫描时，查看系统报警的次数，与扫描次数相比较，据此分析系统的漏报率。同时分析系统日志文件，根据扫描数据包的标记，分析系统的误报率。统计数据如表6.1：

表6.1 模拟蠕虫扫描结果

序号	扫描强度 次/秒	扫描时间 分钟	间隔时间 分钟	误报率	漏报率
1	10	1	2	0.132	0.144
2	10	4	1	0.130	0.146
3	50	2	5	0.125	0.134
4	50	3	5	0.120	0.122
5	100	3	3	0.115	0.107
6	500	2	2	0.109	0.084
7	1000	1	1	0.085	0.052

分析漏报率这一栏，可以发现，随扫描强度的增加，系统的漏报率降低的幅度较小。系统运行过程是先分析流量，构建可疑属性数据库，继而进行流量检测，再查看日志文件，可以得出结论，系统的漏报多是在蠕虫扫描的初期，此时正在进行流量的分析。而系统的误报率与正常流量有关系，与此时系统正常的应用存在一定关联。

6.2 特征码自动生成测试

6.2.1 测试环境

特征码自动生成系统其数据来源是可疑流量池，输出是特征码数据库。对于可疑流量池，本文直接将几种已知蠕虫的数据包放入可疑流量池，然后提取蠕虫特征码，与现有系统提取的特征码进行比较。

6.2.2 参数定义

(1) 最多令牌特征码的参数定义

f_{min}：可疑流量池中可疑数据包个数的阀值，当数据包超过该值，处罚特征码自动生成器，此处设为16；

L_{min}：令牌最短的长度，即令牌集中所有令牌都必须不小于此值，此处设为3；不存在令牌的最大长度值；

k：k为设定的阀值，在剔除令牌中的噪声时使用，一般取0；

Vmin：设定的阀指，数据包与特征码数据库匹配时使用，与特征码中令牌的权值相关，在此设为：Vmin = max(0.5 * (V1+V2+……+Vm)，min(V1,V2,……,Vm))

(2) 最少令牌特征码的参数定义

f_{min}：可疑流量池中可疑数据包个数的阀值，当数据包超过该值，处罚特征码自

动生成器，此处设为8；

L_{min}：令牌最短的长度，即令牌集中所有令牌都必须不小于此值，此处设为2；存在令牌的最大长度值；

6.2.3 测试步骤

首先进行模拟测试，用Apache-Knacker和BIND-TSIG这两种蠕虫攻击作为测试用例，并利用病毒变形引擎Clet和ADMmutate进行了变形处理，使得每种攻击按照不同生成算法的要求生成不同数量的流量，导入流量池进行测试。

然后再以Worm.Welchir蠕虫测试。在封闭环境中，在感染主机运行Welchir蠕虫。被感染主机会产生大量的ICMP，数据包长度为92，数据内容如下：

0800 5c08 0200 44a2 aaaa aaaa aaaa aaaa aaaa
aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa
aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa
aaaa aaaa c714 e03f flb1 0300 6000

6.2.4 测试结果

对于Apache-Knacker和BIND-TSIG这两种漏洞，其结果如表6.2。

表6.2 蠕虫攻击生成特征码测试

算法	Apache-Knacker 特征码	BIND-TSIG 特征码
最长公共字符串算法	HTTP/1.1\r\n	\x00\x00\xFA
最多令牌特征码	(‘GET ’, 0.9), (‘\r\nHost: ’,0.9), (‘\r\n’, 0.8), (‘HTTP/1.1\r\n’,0.7), (‘\xFF\xBF’,0.7)	(‘\xFF\xBF’, 0.9) (‘\x00\x00\xFA’,0.8)
最少令牌特征码	GET, HTTP/1.1\r\n., \r\nHost, \r\n., \r\nHost, \xFF\xBF, \r\n	\xFF\xBF, \x00, \x00, \xFA
Bayes 算法	‘\r\n’: 0.0000, ‘: ’: 0.0000, ‘\r\nHost: ’: 0.0022,‘GET ’: 0.0035, ‘ HTTP/1.1\r\n’: 0.1108,‘\xFF\xBF’: 3.1517. Threshold: 1.9934	‘\x00\x00\xFA’: 1.7574, ‘\xFF\xBF’: 4.3295 Threshold: 4.2232

根据结果分析，特征码生成系统可以有效的提取蠕虫特征码，将生成特征码加入特征码数据库，再次模拟这两种蠕虫的攻击过程，查看能否根据特征码准确的检测蠕虫。最多令牌特征码其结果如表6.3。

表6.3 蠕虫最多令牌特征码检测测试

	Apache-Knacker	BIND-TSIG
漏报率	0.049	0.053
误报率	0.069	0.078

最少令牌特征码其结果如表6.4。

表6.4 蠕虫最少令牌特征码检测测试

	Apache-Knacker	BIND-TSIG
漏报率	0.057	0.051
误报率	0.048	0.069

根据结果可看出，与表6.1相比，特征码检测的漏报率和误报率都有所降低。表6.3与表6.4相比较，可看出最多令牌特征码的漏报率小，而误报率偏大；最少令牌特征码的漏报率偏大，而误报率较小。

对于Worm.Welchir蠕虫，运行特征码生成系统，采用最多令牌特征码和最少令牌特征码，分别生成蠕虫特征码。对于Welchir蠕虫，两种方法都提取了同样的令牌，而且令牌只有一个，令牌如下：

aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa
aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa
aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

生成同样令牌的原因是蠕虫传播的数据包相同，采用最多公共子串算法和字符串对其算法得到了同样的结果。

6.3 系统问题分析

根据上述测试的过程和测试的结果，发现系统存在如下问题：

(1) 系统运行的速度

网络安全设备的处理速度一直是影响网络性能的一大瓶颈，虽然本系统是以并联方式接入网络的，不影响网络本身的性能，但是如果系统速度跟不上网络数据的传输速度，那么检测系统就会漏掉其中的部分数据包，从而导致漏报而影响系统的准确性和有效性。对于1G/s的网络而言，处理每个数据包需要12us，而现在一般主干网的网络交换设备速度甚至多达10G、40G，多用硬件实现交换与路由，但本系统是以软件实现，故其难以应用于主干网环境中。

本文提出了两种特征码，其生成算法虽简单易行，但计算相对复杂，在高速的网络环境中能否及时的生成特征码，以便在随后的蠕虫爆发期检测蠕虫，尚是个疑问。在系统中，截获网络的每一个数据包，并分析、匹配其中是否具有某种蠕虫攻击的特征需要花费大量的时间和系统资源。目前大部分现有的网络安全设备只有百兆的检测速度，随着千兆、万兆网络的大量应用，网络安全检测技术发展的速度已经远远落后于网络速度的发展。

(2) 蠕虫检测的准确性

基于异常发现的检测模块通过出入口双向流量统计分析，系统能否准确运行依赖于阀值的设定，阀值则是由前期的统计分析来确定，很难取得一个准确的值；系统的流量分析的单位时间该如何确定，也是一个难题；对于多属性中包长的范围值是个模糊值，而这都造成了流量检测很难准确进行。

目前应用系统越来越复杂，网络环境也越来越复杂，许多蠕虫活动很难以用简单的蠕虫传染统计模型来刻画，而现有的复杂的统计模型在计算量上不能满足实时的检测要求！需要提出新的蠕虫传染模型来对网络流量活动进行实时有效的统计分析与建模。

(3) 系统运行的局限性

本文提出了出入口双向流量检测，在出口流量上专门分析失败连接，通过计算失败连接来发现可疑的蠕虫扫描，但是这种方法却有相当大的依赖性，依赖于主机或路由设备的设置。前文所说，TCP等失败连接报文由主机或路由器产生，但是很多主机或路由器上可以设置不发出这种报文，如果没有这种报文，那么失败连接的检测就无法实现。

另外，本系统是基于单包检查的，协议分析得不够，因此无法识别伪装或变形的网络攻击，对于IP分片攻击也没有予以足够的重视，这也造成大量漏报和误报。同时统计方法中的阀值难以有效确定，太小的值会产生大量的误报，太大的值会产生大量的漏报；当使用机器训练学习的方法进行确定时，又需要前期大量的操作时间，且很难具有通用性。因此如何提高监测的准确度是必须要解决的一个问题。

7 总结与展望

7.1 总结

本本文对蠕虫定义，蠕虫攻击模式，蠕虫典型传播模型理论进行了研究与探讨，对在蠕虫检测方面的主要研究理论方法成果做了综述，对传统蠕虫检测方法做了介绍。

在上述基础上，本文在网络层研究未知的网络蠕虫，主要研究的是未知网络蠕虫的检测与特征码的自动生成。本文主要采用数学方法分析网络流量的变化，发现其异常，聚类分析后，构建可疑属性数据库。根据可疑属性数据库，从网络数据流中检测出未知的网络蠕虫流量。当发现异常的蠕虫流量之后，向用户报警，将异常流量写入日志，同时将其导入到可疑流量池中，进行特征码的生成。本文介绍了两种不同的方法来生成特征码，这两种方法都是先提取令牌，构建令牌集，之后生成特征码。

本文的研究中，有如下两个创新点

(1) 本文提出了出入口双向流量分析，在入口上进行多属性相似度的检测，在出口上进行滑动窗口的失败连接检测，最后进行相关的聚类分析，写入可疑属性数据库。

(2) 本文提出了两种不同的特征码自动生成算法，最多令牌特征码和最少令牌特征码，针对这两种特征码分别进行了定义和算法流程分析。

7.2 展望

一个能够在高速网络环境中，准确检测到网络蠕虫的传播，及时生成蠕虫的特征码，并且能够自动应对的网络蠕虫检测系统是目前急需的。针对这一个目标，后续研究工作可以在四个方面展开：

(1)提高检测实时性：随着网络速度越来越快，现有的网络数据捕获技术需要进一步的提高，从而提高网络蠕虫检测的实时性。网络流量的实时采集、实时分析等技术可以作为研究的方向。

(2)提高检测精度：针对各种检测技术的应用，网络蠕虫的传播越来越隐蔽，目前的网络蠕虫检测技术还存在一定的误报率和漏报率。网络蠕虫检测有效性、准确性的提高，误报率、漏报率的降低是追求的目标。

(3)简化特征码生成算法：随着网络速度越来越快，蠕虫的传播也越来越迅捷，

缓慢开始阶段将缩短，而高速爆发阶段将提前，而本文采取的特征码生成算法难以有效如此应对高速的网络环境，简化生成算法的复杂度，将是下一步的重点。

(4)研究防范应对技术：由于高速网络的出现，网络蠕虫的传播速度也越来越快，传统的人工应对技术已经不能对付网络蠕虫的快速传播。在研究网络蠕虫检测技术的同时，需要研究网络蠕虫的自动应对技术，防止网络蠕虫的大规模泛滥。网络蠕虫的防范和控制还主要是采用人工手段，针对主机主要采用手工检查、清除，利用软件检查、清除，给系统打补丁、升级系统，采用个人防火墙，断开感染蠕虫的机器等方法；针对网络主要采用在防火墙或边缘路由器上关闭与蠕虫相关的端口、设置访问控制列表和设置内容过滤等方法。

蠕虫存在的最根本的原因是一是人们观念的疏忽，二是软件中存在漏洞。

人作为计算机的使用者和最终的指挥者，是信息安全框架中的重要组成部分，这一点在过去却往往被忽视，人们将信息安全的保障经常寄希望于购置各种必需的安全 产品，以为有了这些安全产品就可以高枕无忧，所以导致即使花大量费用购置了安全产品的政府以及企业仍然没有摆脱遭受攻击的烦恼。一般计算机用户缺乏计算机专业知识，安全意识不够，在复杂的上网环境下很难保证主机有效的安全；对于计算机专业人员而言，每天去浏览追踪自己系统中的成百上千个应用软件供应厂商的网站查询最新的漏洞公告显然也不现实。而人们对于网络安全的惰性和侥幸心，也容易无意识的忽略安全问题。随着人们网络安全意识的提高，蠕虫的爆发也将越来越少见。

增强人的安全意识，是一种从外围来加强安全防范的方法；提高软件产品的安全性，尽可能减少软件中出现漏洞的可能，却是从内部提高软件和系统的安全性。高质量的软件产品不仅能减少软件被网络加以利用的可能，而且能够减轻软件使用与维护者的负担。由于软件的编写过程中漏洞存在的不可避免性，导致在现有的软件设计和分发体系下，蠕虫的产生和肆虐也是不可避免的。如果想彻底消灭蠕虫，必然要改变现有软件的设计和分发模式。Web Services技术的出现，可能会导致软件架构设计及分发体制的根本改变，这为彻底避免蠕虫的爆发提供了一线曙光。

要彻底的把网络蠕虫拒之网络之外，确保网络的安全运行，不但要加强对未来蠕虫技术的深入研究，在检测和防御蠕虫的技术上还需要更多科研人员的共同努力进行蠕虫检测技术的创新。

致谢

研究生生活即将结束，回首这两年的生活点滴，感慨颇多。两年来，华中科技大学的优美校园与严谨学风，为我的人生留下一段美丽的篇章；那些一直关心、鼓励、帮助我的所有师长和朋友更是我心中珍贵的回忆。值此论文完成之际，谨向曾经给予我关心和帮助的老师、同学和亲友表示衷心的感谢。

衷心感谢导师李汉菊教授和李之棠教授两年来对我的关心、指导和教诲。本论文的写作是在两位李老师的直接指导下完成的，李汉菊老师渊博的知识、敏捷的思维、平易近人的工作作风使我受益匪浅，深深的影响了我，李之棠老师对科学研究的执著精神、严谨的治学风范和豁达宽广的心胸永远是我学习的榜样。两年来，两位李老师给我创造了许多学习机会，为我提供了良好的研究环境和具有挑战性的课题，使我掌握了计算机网络和网络安全方面的知识，提高了自己的动手能力，而且在生活中给予我关心、帮助和鼓励。这两年内我取得的每一点进步都和他们的辛勤工作密不可分，谨此向两位李老师表示诚挚的谢意！

在实验室学习的两年内，我得到了老师们无私的关心和帮助。衷心感谢涂浩老师的热心指导和帮助，涂老师开朗豁达，视野开阔，思维敏捷，不愧是青年教师中的楷模。感谢黎耀老师，我的成长离不开黎老师的关怀与帮助。感谢李战春老师、陈琳老师、柳斌老师、刘蜀豫老师、陈琳老师和周丽娟老师等，在学习和工作中他们给了我特别的关心、指导和建议。衷心感谢网络中心的工作人员：汪燕老师、秦山秀老师、杨勇老师、汪翔老师、李汉华老师等等，感谢你们的关心和帮助。

感谢李文瑾、马媛和姜浩，在同一个毕业设计组和项目组中，他们给了我很多帮助；感谢同一宿舍的三位兄弟：李维、张雄刚和章勇，让我两年的宿舍生活在快乐中度过；感谢刘文芹和李佳，同样来自武汉理工的我们，相识中多了一份亲切。感谢李奕、林晓、孙飞、乐艳辉、陈剑斌和张记评，研究生生活因为有你们而精彩。我们在一起营造了一个充满自由、朝气和富于创新精神的学习和科研环境。

衷心感谢父母与弟弟对我多年的教诲和关爱，对我一贯的支持和鼓励。他们是我能顺利完成学业的精神支柱，在我多年的求学生涯中，给予了我精神上的支持和物质上的资助。我能有今天的一切，都是源于他们在生活上的帮助。

最后衷心感谢对本论文进行评审、提出宝贵意见的李汉菊老师、李战春老师、胡福林老师、邹德清老师等等，你们无私的奉献与严谨的工作作风给我们留下了深刻的影响。

参考文献

[1] 赖英旭, 李征.未知病毒检测技术的研究.计算机科学, 2006,33(8):300~303

[2] Lo R, Kerchen P and Crawford R. Towards a test bed for malicious code detection. Compcon Spring '91 Digest of Papers, 1991, 32(5): 160~166

[3] Karresand M. Separating Trojan Horses, Viruses, and Worms--a Proposed Taxonomy of Software Weapons. Information Assurance 2003,25(3):127~134

[4] 郑辉. Internet蠕虫研究:[博士学位论文].天津市:南开大学,2003.

[5] CERT/CC Statistics 1988-2007,http://www.cert.org/stats/cert_stats.html

[6] 戴汝为, 操龙兵. Internet——一个开放的复杂巨系统. 中国科学E辑,2003,3 (4):289~296

[7] 王平.大规模网络蠕虫检测与传播抑制:[博士学位论文].哈尔滨:哈尔滨工业大学,2006.

[8] Denning D.E. An intrusion-detection model. IEEE Transactions on Software Engineering, 1987,5(13):222~232

[9] 肖政宏, 尹浩. 基于网络流量统计分析的入侵检测研究. 微电子学与计算机,2003,23(5):289~296

[10] Singh Sumeet, Estan Cristian, Varghese George, et al. The EarlyBird System for Real-time Detection of Unknown Worms. CS2003-0761, 2003,10(9):289~296

[11] Spitzner L. Honeypots:Tracking Hackers. Addison-Wesley Professional, 2003,15(8):239~256

[12] Chen Xuan, Heidemann John. Detecting Early Worm Propagation through Packet Matching. Information Sciences,2004,25(3):189~196

[13] Kim H. A. and Karp B. Autograph. Toward Automated, Distributed Worm Signature Detection, In : Proceedings of the 13th USENIX Security Symposium, San Diego, CA. 2004.59~66

[14] Newsome J, Karp B, and Song D. Polygraph: Automatically generating signatures for polymorphic worms. In: IEEE Security and Privacy Symposium, Oakland, California, 2005.135~146

[15] Tang Y. and Chen S. Defending against internet worms: A signature-based approach. In: Proceedings of IEEE INFOCOM'05, Hong Kong, 2005.13~23

[16] Wang S. Anomalous payload-based network intrusion detection. In: Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection, Sophia Antipolis, France, 2004. 45~53

[17] Kreibich C. and Crowcroft J. Honeycomb creating intrusion detection signatures using honeypots. In: Proceedings of the Second Workshop on Hot Topics in Networks, Boston, 2003.119~127

[18] Dagon D, Qin X, W. Lee, et al. Honeystat: Local worm detection using honeypots. In: Proceedings of RAID'04, volume 3224 of Lecture Notes in Computer Science, Springer, 2004.39~58

[19] Eugene H, The Internet worm programs. ACM Computer, 1989,23(3):17~57.

[20] Douglas E.Comer著. 计算机网络与互联网.第一版.徐良贤,张声坚等译.北京:电子工业出版社,2004.125~135

[21] 文伟平, 卿斯汉, 蒋建春等.网络蠕虫研究与进展.软件学报,2004,15(8):35~39

[22] 翟光群, 张玉凤.网络蠕虫病毒分析与防范研究.河南科学,2005,23(6):136~137

[23] 栾新民, 廖闻剑. Nimda蠕虫分析与防范. 计算机应用研究, 2002, 19 (11):155~158

[24] Chen X. and Heidemann J. Detecting early worm propagation through packet matching, USC ISI Technical,2004,16(13): 127~131

[25] 郑辉, 李冠一, 涂菶生. 蠕虫的行为特征描述和工作原理分析, 第三届中国信息与通信安全学术会议, 武汉, 2003. 225~231

[26] Lee Cynthia Barley, Roedel Chris, Silenok Elena．Detection and characterization of port scan attacks. In: Proceedings of IEEE INFOCOM'02, New York, 2003.35~46

[27] 赖海光, 许峰. 基于Dempster-Sharer证据理论的端口扫描检测方法. 电子学报,2005,34 (17) :58~68

[28] Kephart JO, White SR. Measuring and Modeling Computer Virus Prevalence. In: Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, Okaland, 1993. 2~15

[29] Staniford S, Paxson V, Weaver N. How to Own the Internet in Your Spare Time.In: 11th Usenix Security Symposium, San Francisco, 2002.38~45

[30] Zou C, Towsley D. Routing Worm: A Fast Selective Attack Worm Based on IP Address Information. Univ. Massachusetts Technical Report, 2003,3(6):234~245

[31] J. Jung,V. Paxson, A. W. Berger, et al. Fast portscan detection using sequential hypothesis testing, IEEE Symposium on Security and Privacy, Los Alamitos, Calif. ,2004.179~188

[32] 张运凯, 王方伟, 张玉清等. 蠕虫病毒的传播机制研究.计算机应用研究,2005, 4(7):39~45

[33] 刘晟,李玲. 网络蠕虫分析与检测防御. 现代计算机,2005,20(5):135~139

[34] Gao Lixin, Gong Weibo, Towsley Don, Monitoring and Early Warning for Internet

Worms, In: Proceedings of CCS'03, Washington, DC, USA. 2003.27~31

[35] Streftaris G, Gibson GJ. Statistical inference for stochastic epidemic models. In: Proceedings of the 17th Workshop on Statistical, Chania, 2002: 609~616

[36] 田雪峰, 钟求喜, 苏金树. 蠕虫早期检测系统研究.信息安全与通信保密, 2005, 9(5):68~75

[37] RFC 792, Internet Message Control Protocol, Volume 792 of Request for Comments, 1981

[38] 西安交大捷普网络科技有限公司. 一种实时检测网络蠕虫病毒的方法.中国, 发明,CN1674530A, 2005.1~9

[39] 温世强, 段海新, 吴建平. 网络蠕虫爆发的检测算法及其应用.计算机工程与设计, 2004, 26(5):56~61

[40] Chunmei Yin, Mingchu Li, JianboMa, et al. Honeypot and scan detection in intrusion detection system. Electrical and Computer Engineering, 2004,3(2):1107~1110

[41] 刘庭华, 宋华, 戴一奇. 自适应分布式端口扫描检测方法. 计算机工程与设计. 2003, 27(9):29~36

[42] 宋安利, 唐成, 武维善. 一种基于异常流量的蠕虫入侵检测.网络安全技术与应用. 2004,27(5):44~50

[43] 李贤平. 概率论基础. 第二版. 北京:高等教育出版社,1997.188~197

[44] Zhichun Li, Manan Sanghi, Yan Chen Ming, et al. Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience, the 2006 IEEE Symposium on Security and Privacy, San Diego,CA,USA, 2006:254~263

[45] Y. Tang and S. Chen. Defending against internet worms: A signature-based approach. In: Proceedings of Infocom, 2003:39~47

未知网络蠕虫检测与特征码自动生成的研究

作者： 蔡新
学位授予单位： 华中科技大学