摘要

随着互联网应用的快速发展，无论是企业还是个人用户越来越重视信息的安全，入侵检测系统作为信息安全重要的被动防御手段，受到国内外信息安全研究机构和安全厂商的关注。如何提高入侵检测系统的能力和效率是目前研究的热点。

为了提高入侵检测系统的检测能力，做了三个方面的工作，首先结合免疫入侵检测系统和分布式入侵检测系统的优点，设计了一个完整免疫代理分层布置的入侵检测系统模型，免疫代理分布在普通的工作站、代理服务器和专用服务器上，分别称为叶子代理、分支代理和主干代理，各司其职；其次对叶子代理、分支代理和主干代理功能进行了设计，其中在普通工作站上的叶子代理负责数据收集和初级免疫训练，并将数据和训练结果交给分支代理；在代理服务器上的分支代理收集叶子代理传过来的数据，使用免疫算法进行训练，将训练结果发送到服务器；主干代理负责对服务器的入侵检测，任务比较重，为了提高效率根据不同的服务设计有DNS代理、HTTP代理、FTP代理等，各种代理只收集相应的服务数据，并进行免疫训练、免疫细胞繁殖，结果发送到数据库服务器；最后将工作站、代理服务器主机参数，接入层、汇聚层、核心层网络参数，私有云中各服务器主要参数按照加权平均的方法计算整个网络系统的健康值。进行了两种对比试验，一是不分层的免疫代理系统，发现健康指数偏低；二是分层的免疫代理系统，健康指数较高。

由于得到了工作单位的软硬件资源和人力资源的大力支持，系统研究取得阶段性成果，在一些高校得到应用，将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵综合检测系统，提高检测的效率、减少网络和计算机系统的压力。

关键词：免疫代理 免疫算法 入侵检测 私有云

目录

请在此插入目录

图目录

请在此插入图目录

表目录

请在此插入表目录

1绪论

入侵检测系统作为信息安全重要的防御手段，一直受到国内外研究机构和软硬件生产厂家的关注，入侵检测相关技术也是近些年的研究热点。将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵检综合测系统，为入侵检测系统的研究提供了新思想和新方法。

本章首先介绍研究背景，然后总结国内外研究现状和本文研究的目的和意义，最后介绍了本文的研究方法和所做的工作。

1.1课题研究背景

1.1.1 Internet（国际互联网）发展迅速

Internet（国际互联网）的发展已经远远超出人们的想象，从2008年开始中国互联网络信息中心受工业和信息化部的委托每年1月和7月定期发布《中国互联网络发展状况统计报告》，2019年8月发布了第44次报告[1]，截至2019年6月，我国网民规模达8.54亿，互联网普及率为61.2%。

中国人在信息获取、商务交易、交流沟通、网络娱乐等方面已经无法离开互联网。

1.1.2 信息安全问题日益突出

现阶段网络安全的主要威胁包括：

① 钓鱼欺诈成为网络安全首害。

②网站拖库成为黑客主流盗号手段。

③ 由于针对普通用户和操作系统的一般性攻击成本高、利益低，高级持续性针对特定组织的多方位的攻击（APT）日渐增多。

① 病毒数量增长较快。

② 挂马网站、钓鱼网站增加较快。

③ 网络欺诈已成为危害网民上网安全的最主要威胁之一。

④大规模个人隐私信息泄露事件频发。

1.1.3 防护手段多样、效果有待提高

(1) 网络层防范手段包括：路由交换策略、VLAN划分、防火墙、隔离网闸、入侵检测、抗拒绝服务、传输加密等。

(2) 系统层防范手段包括：漏洞扫描、系统安全加固、SUS补丁安全管理、应用层、防病毒、安全功能增强等。

(3) 管理层防范手段包括：独立的管理队伍、统一的管理策略等。

入侵检测系统作为信息安全重要的被动防御手段，受到国内外研究机构和信息安全厂家的关注。将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵综合检测系统，提高检测的效率、减少网络压力、提高计算机处理能力。

1.2 入侵检测系统发展与研究现状

1.2.1 入侵检测系统发展状况

1) 入侵检测概念的出现

1980年James P．AnderSon博士发表《Computer Security Threat Monitoring and Surveillanc》[2]报告，他提出计算机系统的安全威胁主要来自三个方面，一是来自外部的渗透，二是来自内部的渗透，三是来自某些不合法的行为。首次提出要对三种威胁进行检测的概念，利用系统审计系统，对各种威胁进行跟踪和分析。

2) 系统模型的发展

利用审计数据分析系统入侵具有滞后性，上世纪80年代中期，有专家开始研究实时入侵检测系统模型，乔治敦大学的Dorothy Denning和SRI公司计算机科学实验室（SRI/CSL）的Peter Neumann提出了入侵检测专家系统（IDES）。该系统模型独立于特定系统平台和应用系统环境，提供一个通用框架，实时发现系统弱点和各种入侵威胁。

图1-1 IDS结构框架

1988年， Teresa Lunt等人改进并完善了Denning的入侵检测模型，并开发了一个功能更加完善的入侵专家系统。该系统主要包括两个部分，一个异常检测器，用于建立统计异常的模型，二是专家系统，按照规则对特征分析和检测，如图1-1所示。

3) 研究现状

1990年入侵检测系统研究翻开了新的一页，美国加州大学戴维斯分校的L. T. Heberlein等人开发出了网络安全监视器（Network Security Monitor，NSM）。该系统不需要将系统审计数据作为分析源，首次直接对网络数据流进行过滤和审计，提高了入侵检测系统的实时处理效果。从此之后，入侵检测系统发展为基于网络的IDS和基于主机的IDS等两大阵营。

1988年爆发了莫里斯蠕虫事件，整个网络遭受重创，军方、学术界和企业开始高度重视网络安全问题。为了提高入侵检测的效率，美国国家安全局等机构开始资助军队密码研究中心、国家实验室、重点高校的研究结构，开展了分布式入侵检测系统（DIDS）的研究，集成主机和网络，结合基于主机和基于网络的入侵检测方法。

DIDS的检测模型分为6层，包括数据、事件、主体、上下文、入侵类型、安全状态等，每层分工明确相互协调。

从1990年代到现在，由于入侵手段和方法的不断变化，入侵检测系统模型的研究和系统的研发在分布式和智能化两个方向不断进步和发展。目前，美国和欧洲的一些机构在智能分布式入侵检测系统研究中处于世界领先地位，主要包括SRI公司的计算机科学实验室、洛斯阿拉莫斯国家实验室、美国加州大学的戴维斯分校以及哥伦比亚大学等。

1.2.2 国内研究现状

1997年国内开始有免疫计算的文章出现。2001年开始，有关免疫系统的硕博论文涌现。通过在知网中查询发现2008年达到了创记录的619篇，以后每年递减。论文主要集中在免疫入侵检测系统模型研究和算法研究上，真正实现完整系统的比较少，没有实用价值，因此研究人数逐年减少。

目前开展免疫入侵检测系统研究和开发的学校主要有电子科技大学、西安电子科技大学、重庆大学、吉林大学、国防科技大学、华中科技大学、北京邮电大学、山东大学、上海交通大学、哈尔滨工程大学、四川大学、中南大学等，如图1-2所示。

图1-2 学校免疫入侵检测技术硕博士论文

1.3 课题的研究目的和研究目标

入侵检测系统作为信息安全重要的防御手段，一直受到国内外研究机构和软硬件生产厂家的关注，入侵检测相关技术也是近些年的研究热点。将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵检综合测系统，为入侵检测系统的研究提供了新思想和新方法。

1.4 本文的主要研究内容

随着互联网用户增加和应用的快速发展，信息安全问题日益突出，入侵检测系统作为信息安全重要的被动防御手段，受到国内外研究机构和信息安全厂家的关注。将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵综合检测系统，提高检测的效率、减少网络和计算机系统的压力。本文所做的工作包括下面几个部分：

（1）结合免疫学的自我修复能力和分布式强大处理能力，本文设计了一个基于分层免疫代理入侵检测系统模型，将免疫代理合理分布在普通的工作站、代理服务器和专用服务器上，分别称为叶子代理、分支代理和主干代理，各司其职。

（2）对叶子代理、分支代理和主干代理功能进行了设计，其中在普通工作站上的叶子代理负责数据收集和初级免疫训练，并将数据和训练结果交给分支代理；在代理服务器上的分支代理收集叶子代理传过来的数据，使用免疫算法进行训练，将训练结果发送到服务器；主干代理负责对服务器的入侵检测，任务比较重，为了提高效率根据不同的服务设计有DNS代理、HTTP代理、FTP代理等，各种代理只收集相应的服务数据，并进行免疫训练、免疫细胞繁殖，结果发送到数据库服务器。

（3）将工作站、代理服务器主机参数，接入层、汇聚层、核心层网络参数，私有云中各服务器主要参数按照加权平均的方法计算整个网络系统的健康值。进行了两种对比试验，一是不分层的免疫代理系统，发现健康指数偏低；二是分层的免疫代理系统，健康指数较高。

1.5 论文内容组织安排

全文共分为五章。

第一章：绪论，主要介绍了信息安全状况、入侵检测系统研究的目的和主要任务、论文组织安排等内容。

第二章：相关技术，主要介绍了入侵检测的基本概念及模型、生物免疫系统基础、免疫学习算法在网络入侵检测中的应用等。私有云系统，概要地描述私有云系统的组成和设计方案。

第三章：系统总体设计，介绍基于免疫代理分层设计情况，叶子代理、分支代理和主干代理的设计等。

第四章：详细设计和实现，对叶子代理、分支代理和主干代理功能进行了设计，并测试网络系统的健康指数。

第五章：总结与展望，总结系统设计的经验与不足，展望今后的工作。

2 相关技术

2.1 生物免疫入侵检测系统

2.1.1 生物免疫系统基础

1）生物免疫的概念

① 免疫应答和免疫

当机体外部有害病原体入侵，机体免疫细胞被激活，免疫细胞做出反应的过程称为免疫应答。免疫应答包括机体先天获得可以快速清除病原的固有免疫应答和获得性免疫应答两种，后者能够具有区分“自我”与“非我”的能力，能够区分和记忆，能够自我调节，具有特异性，多样性等优良特性[3，4]。

免疫是指免疫系统受到外界侵扰时，机体免受病原侵害的应答反应。

② “自我”与“非我”

“自我”主要是指机体自身的细胞，“非我”就是病原体，包括：侵入机体的毒性有机物、机体内突变细胞和死亡细胞等。生物免疫系统的首要任务是区分“自我”和“非我”，如果是“自我”，则不产生应答；如果遇到“非我”病原体，免疫细胞产生免疫应答，消除病原体。免疫细胞通过不断的进化、相互影响实现生物体的免疫功能。免疫系统通过基因选择、负选择选取特征，最后进行克隆选择从而完成免疫系统的构建。免疫系统不受其他器官的支配，不需要预先了解特定信息、自组织完成相应的任务。

2）生物免疫的特点

生物免疫系统具有自我控制、自我修复等特点，主要包括：

分布性：生物体的淋巴细胞分布在生物体的各个器官中，他们之间形成一个自我管理的分布式、自适应的自组织网络，没有一个中心控制器控制、指挥和统一他们的行为。

鲁棒性：生物免疫系统中过于庞杂，必定有冗余部分，因此即使某些组件的缺失一小部分也不会影响系统的主要功能。

适应性：生物免疫系统随外界环境的变化而变化，通过对新抗原的识别、学习和记忆，帮助后续识别相应的抗原。

人们希望通过对生物免疫机理的研究，将生物免疫系统的自我控制、自我修复等能力应用到IDS系统中，提高IDS系统的效率和性能。

3）生物免疫的功能

在自然界、生物界中，普遍存在着免疫现象，这对物种的生存，对物种的繁衍都发挥着相当重要的作用；

生物免疫的功能，主要是由参与免疫反应的细胞或由其构成的器官来完成的；

生物免疫主要有两种类型：特异性免疫（Specific Immunity）和非特异性免疫反应（Nonspecific Immunity）；

生物免疫的系统，是通过自我的识别、相互的刺激和制约，而构成了一个网络结构，这个结构式是动态平衡的。

4）免疫生物学的基本概念

（1） 抗原

是指能够刺激和诱导机体的免疫系统，使其产生免疫应答，并能与相应的免疫应答产物在体内或体外发生排斥反应的物质。

（2）抗体

是指免疫系统受抗原刺激后，免疫细胞转化为浆细胞并产生能与抗原发生特异性结合的免疫球蛋白，该免疫球蛋白即为抗体。

2.1.2 生物免疫系统在入侵检测系统中的应用

1）生物免疫技术应用网络安全现状

当前基于人工免疫技术主要应用到反病毒和反入侵等网络安全研究和产品方面。IBM公司研发了计算机免疫系统，只能用于病毒的高效查杀， S. Forrest等人提出了可用于病毒查杀和反入侵两个方面的非选择算法[5，6]。

（1）基于生物免疫能力的计算机免疫系统

IBM公司的J.O.Kephart等研究人员，通过模拟生物免疫系统的各个功能部件，设计了一种计算机病毒免疫模型和系统，系统可以对病毒特征自动进行识别、分析和清除[7-9]。

① 已知病毒的处理

对已知病毒，该系统通过对病毒库中病毒特征的匹配，判读病毒的存在，启动相应的病毒程序识别和清除计算机病毒。

② 未知病毒

未知病毒会感染“钓饵”程序，“钓饵”程序获取病毒样本、分析，通过提取病毒特征，自我设计相应的病毒查杀程序。病毒查杀程序将所产生的病毒特征传递到网上邻近计算机中，网络上的其它计算机很快就具有了该病毒的查杀能力。

（2）基于变化检测的负选择算法

S．Forrest等人通过对T细胞产生和行为机制进行深入研究[10]，提出了一个基于变化检测的负选择算法。T细胞在成熟过程中经过阴性选择，无法识别“自我”，而抗原性异物则被识别并清除[11，12]。

负选择算法不是一个自适应学习算法，是一个变化检测算法，具有很多优点，后人不断发展和完善了负选择算法 [13，14]。

（3）其它

还有许多其他较有影响的工作，如R. E. Marmelstein等人提出了计算机病毒免疫分层模型和系统，只能用于病毒的查杀；D. Dasgupt等人提出了基于生物免疫自主体的入侵检测系统框架等。[15，16]

2）生物免疫技术入侵检测模型研究

基于生物免疫的入侵检测系统把正常的网络活动和主机访问定义为“自我”，而把异常的网络访问和主机操作定义为“非我”，通常把“非我”判别为入侵活动。其原理可用下面公式描述[17]：



该公式表明，当一个行为被判读为“非我”时，可以判别为非法行为。

基于免疫原理的检测系统包括原始抗体、负选择、检测器、实时数据等部分，检测过程如图2-1所示。

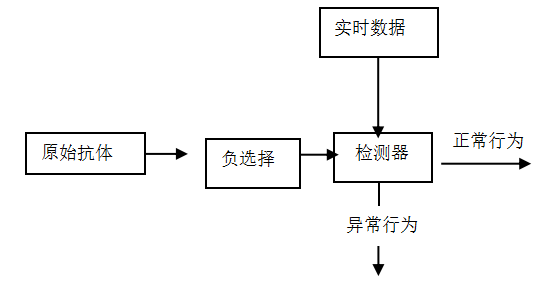


图2-1 基于免疫原理的检测系统

2.2 私有云系统

2.2.1 云计算技术基础

1）云计算定义

互联网中的许多计算资源、网络资源、存储资源处于闲置状态，将这些资源整合起来，形成“云”，云中的所有资源协同工作，平均分配能力，共同完成单机或网络无法完成的任务。云计算不是分布式不仅仅是分布式处理系统，它是一种全新的计算技术，是一种超级计算模式[18，19]。 “云”整合服务器集群，而服务器集群主要包括计算、存储等资源。它使用特定的软件将网络中所有的计算资源集中起来进行管理，对普通用户来说是透明的，用户根本不必考虑底层细节问题，使用它提供的业务简单而且高效，大大节约时间和成本。它的主要目标十分明确，普通用户通过使用任何终端（包括台式机、笔记本电脑、手机、PDA、等设备）连接Internet获取任何所需的计算服务和存储服务，云计算可以将简单终端复杂化，迁移到“云”中的终端可以获得比之强大数倍的服务能力[20，21]。

云计算的特点有很多，主要包括：

(1) 高效、安全虚拟化技术。

(2) 高可靠性。

(3) 协同性。

(4) 按需提供服务。

(5) 低成本 [22-24]。

2）云计算发展历程

随着社会各企事业单位对“云”服务需求的不断变化，需要企业信息管理人员、“云”服务研究和开发人员通力合作，云计算虽然是个新名词，实际上云技术经历了超过30年的发展，目前已经十分成熟。

（1）网格计算

20世纪 80 年代，单机时代被网络互连和互联网时代所替代，单一计算能力有限，无法满足大数据计算的需求，将计算通过网络分发到各个计算机上的网格计算技术就产生了，大大提高了计算能力。

（2）公用计算

上世纪 90 年代，互联网相关业务的出现，使得一些供应商提供虚拟化服务，其将一部分资源租给普通用户，普通用户不用花高额费用搭建自己的网格计算网络，只需通过互联网就能得到所需的计算资源。

（3） SaaS 软件即服务

用户不用购买实体的软件，通过付费租用软件提供商提供的软件，SaaS（Software-as-a-service）就应运而生了。SaaS是基于互联网提供软件服务的软件应用模式，它是一种新型的软件应用部署模式，用户通过少量的费用和管理成本就能获得自己的信息系统。

（4）云计算

有条件的部门将计算、网络和存储资源整合搭建自己的云计算平台，通过互联网租借给用户，云计算集合了网格计算、公用计算和SaaS软件服务等技术的优点。综合了以上几种计算模式的主要优点，云平台提供给用户高效、安全、低成本的服务，逐渐成为应用最广、发展前途最好的计算模式。

3）云计算关键技术

云计算有很多关键技术，主要包括虚拟化技术、分布式处理技术、海量分布式存储技术、协同工作技术等[29，30]。

（1）虚拟化技术

虚拟化技术将计算机、网络、存储等物理资源统一管理统一表示，它是云计算技术中最重要的技术，也是最关键的技术。

（2）分布式处理技术

它使用管理软件统一调配云中大量的廉价计算机、网络和存储资源，使它们协同、高效、安全工作，它结合了网格计算中和分布式处理技术。

（3）海量分布式存储技术

此技术虚拟化所有的存储，使得存储容量无限扩大，为大数据的存储和处理提供帮助。

（4）协同工作技术

云计算可以集合服务器群的能力，使云中数万甚至数十万服务器协同工作，大大提供处理能力。

2.2.2 私有云

为某个组织或机构（学校、企业、政府等）专门构建的云称为私有云，私有云为该组织所独享，相对于供应商托管的服务而言，私有云使得该组织能够获得更多的数据控制。私有云框架结构见图2-2。

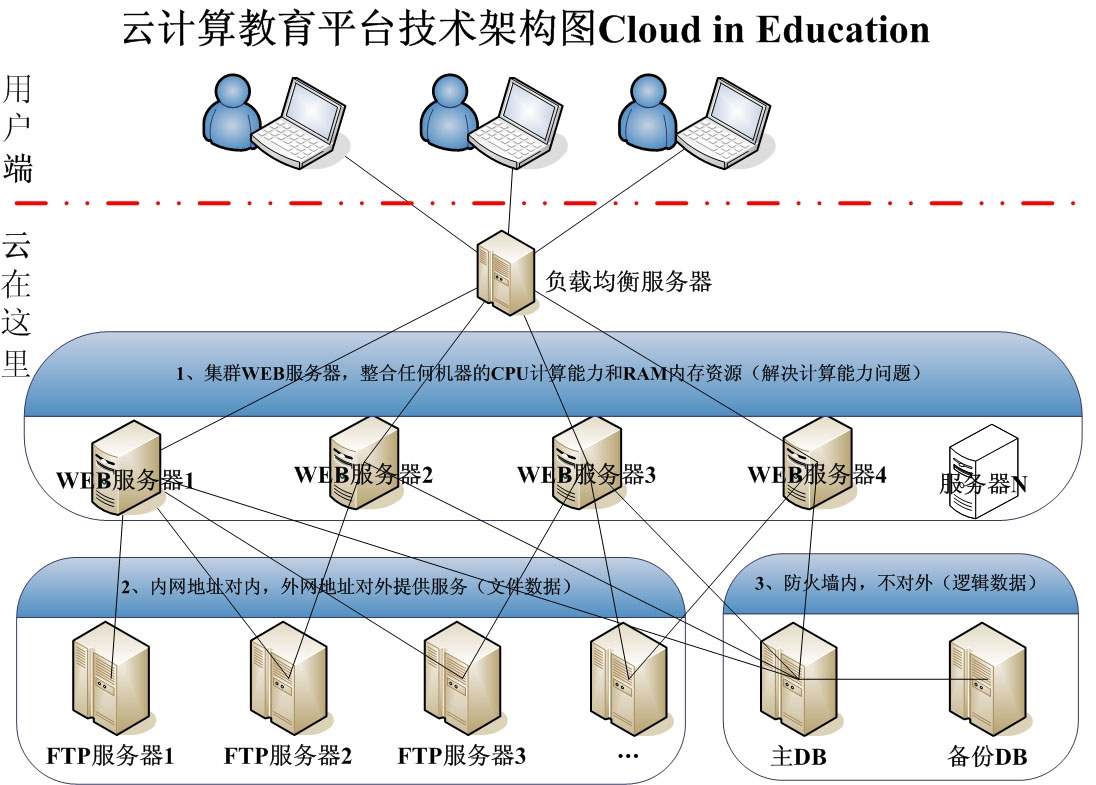


图2-2 私有云框架

1）VMware云计算基础架构

我们选择使用目前比较流行的VMware云计算基础架构，包括以下7个部分如图2-3所示。

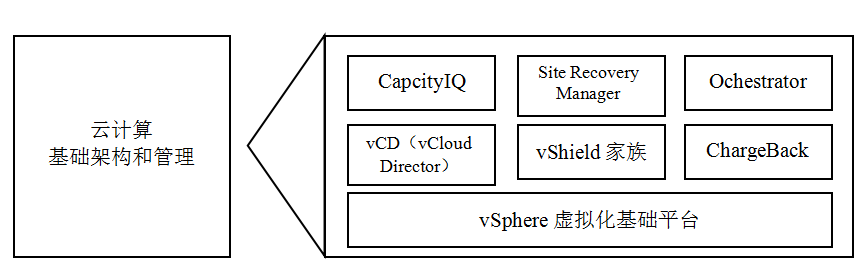


图2-3 VMware云计算基础架构

（1）vSphere：作为虚拟化基础软件，为其它软件提供底层支持。

（2）vCD（vCloud Director）: 云计算平台软件。

（3） vShield家族：虚拟化安全管理系列软件，主要包含vShield Zone、vShield Edge、vShield App、vShield Endpoint几个方面。

（4）ChargeBack：虚拟化资产管理及计费软件。

（5）CapcityIQ：容量规划管理软件。

（6）Ochestrator：虚拟化自动化部署软件。

（7）Site Recovery Manager(SRM)：虚拟化容灾管理软件。

2）VMware终端用户计算

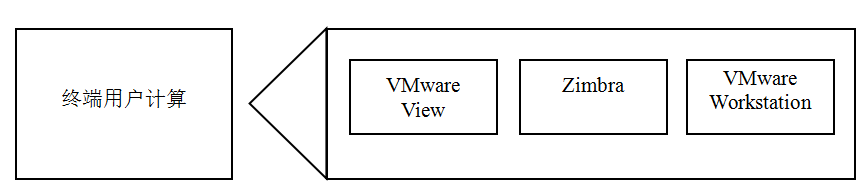
VMware终端用户包括VMware View、Zimbra、VMware Workstation等，如图2-4所示。

图2-4 VMware终端用户

（1）VMware View：桌面虚拟化产品（云终端）

是很著名的桌面虚拟化产品（云终端），VMware最早提出桌面虚拟化技术（VDI），并且现在作为公司除了vSphere之外的另一个重要技术进行推广。

（2）Zimbra

是从Yahoo收购的邮件管理及协同办公产品。Zimbra具有许多与众不同的特点，其“Zimlet”网络服务提供了更多的电子邮件功能。核心产品是Zimbra协作套件（Zimbra Collaboration Suite，简称ZCS）。除了它的核心功能是电子邮件和日程安排服务器，当然还包括许多其它的功能，就象是下一代的微软Exchange。在电子邮件和 日程安排之外，它还提供文档存储和编辑、即时消息以及一个利用获奖技术开发的全功能的管理控制台。ZCS同时也提供移动设备的支持，以及与部署于Windows、Linux或apple操作系统中的桌面程序的同步功能。

（3）VMware Workstation

VMware Workstation是早期的虚拟化软件，其具有非企业化的特点，主要应用到个人操作系统之上，得到广泛使用，操作相对来说比较简单。和它相类似的终端软件有VMware Server等。

2.3 本章小结

本章首先介绍生物免疫的概念、特点、功能，然后介绍了计算机免疫系统、负选择算法、基于生物免疫原理的入侵检测模型等，还详细介绍了一种免疫学习算法。本章还介绍了云计算、云计算的特点、云计算的发展历程，然后介绍了虚拟化技术、分布式处理技术、海量分布式存储技术、协作技术等相关云计算技术。最后以VMware为例子介绍了为某个组织或机构（学校、企业、政府等）专门构建的安全私有云技术。

3 系统总体设计

3.1 系统需求分析

随着互联网用户和应用的快速发展，信息安全问题日益突出，入侵检测系统作为信息安全重要的被动防御手段，受到国内外研究机构和信息安全厂家的关注。将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵综合检测系统，提高检测的效率、减少网络和计算机系统的压力。

3.2 系统总体架构设计

将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中，利用分层免疫入侵检测代理的基础计算和训练能力并结合云计算的强大计算和训练能力构建一个高效的入侵综合检测系统，提高检测的效率、减少网络和计算机系统的压力[25-27]。

（1）结合免疫入侵检测系统和分布式入侵检测系统的优点，本文设计了一个基于分层免疫代理入侵检测系统模型，免疫代理分布在普通的工作站、代理服务器和专用服务器上，分别称为叶子代理、分支代理和主干代理，各司其职[28-30]。

（2）对叶子代理、分支代理和主干代理功能进行了设计，其中在普通工作站上的叶子代理负责数据收集和初级免疫训练，并将数据和训练结果交给分支代理；在代理服务器上的分支代理收集叶子代理传过来的数据，使用免疫算法进行训练，将训练结果发送到服务器；主干代理负责对服务器的入侵检测，任务比较重，为了提高效率根据不同的服务设计有DNS代理、HTTP代理、FTP代理等，各种代理只收集相应的服务数据，并进行免疫训练、免疫细胞繁殖，结果发送到数据库服务器。

下面我们以某个高校为例进行分析，某高校典型网络拓扑图如图3-1所示。

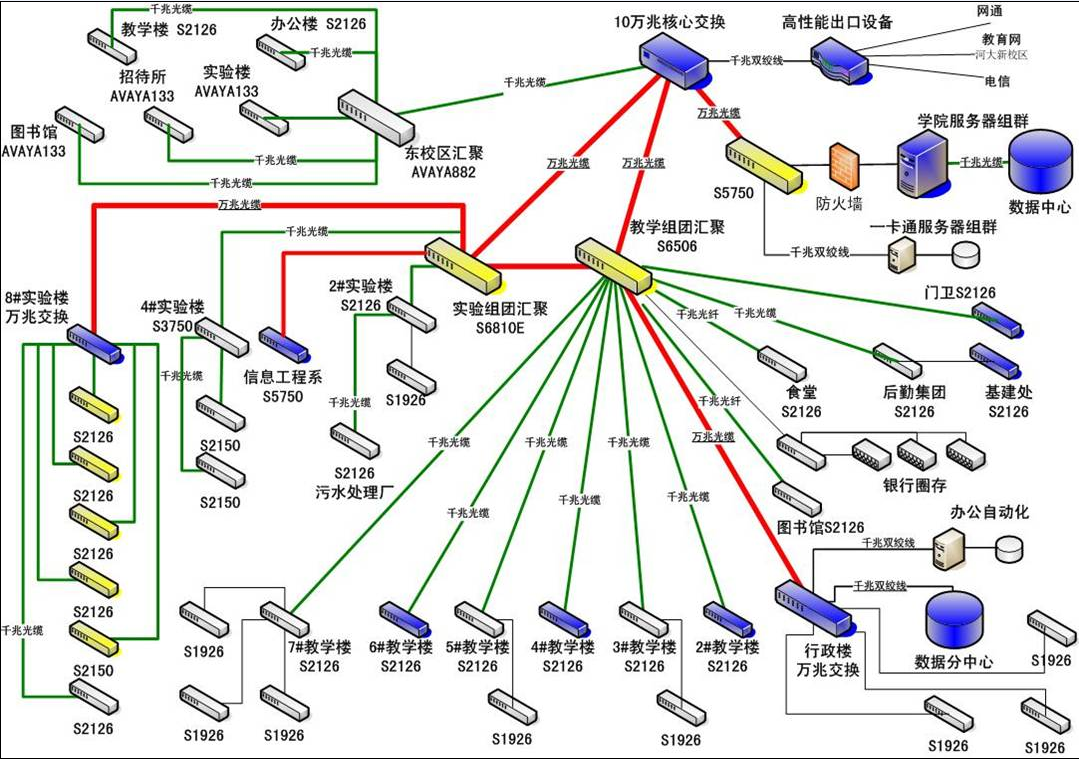


图3-1 某高校网络拓扑图

某高校相关部门网络拓扑图如图3-2所示。从图3-1和图3-2中可以看出，单位计算机的物理构造和逻辑构造类似一棵树，普通用户处于叶节点，特点是能力比较弱，功能单一，不能施加太大的压力；部门服务器和交换机处于分支节点，负责数据包的高速转发，需要排除一些干扰，不能把主要精力放在处理检测病毒和入侵上；中心交换机和各种应用服务器被放在主干节点上，主要任务是无障碍提供各种服务，但极易受到攻击，主干节点的防护十分重要，除了有效的外部防护，还应该做到自身防护的有的放矢，不能把主要精力用在处理病毒和入侵的检测上。

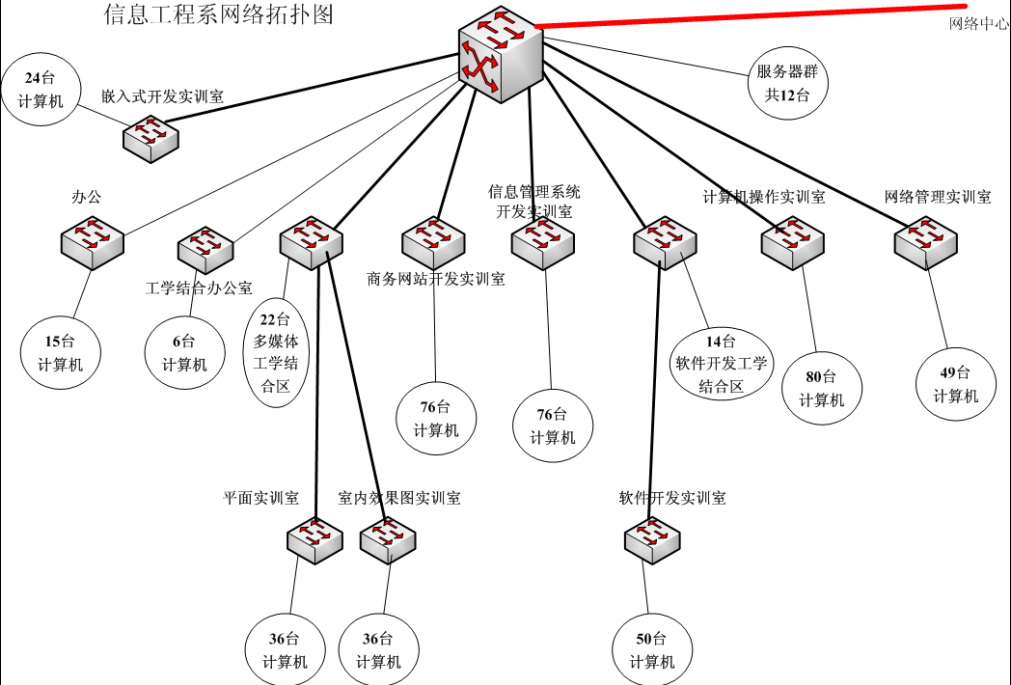


图3-2 某高校相关部门拓扑图

3.3 免疫代理设计

3.3.1 叶子代理的设计

如图3-3所示，叶子代理分布在各工作站上，叶子代理设计比较简单，负责收集对本代理宿机进行的攻击，针对普通工作站上的攻击比较少，并且普通工作站上安装有杀毒软件，可以阻挡大部分的蠕虫、ARP、木马攻击等，但有时工作站上病毒库升级较慢，有些新型病毒的攻击、以及一些未知的非病毒的攻击数据由叶子代理收集并训练，叶子代理负责将训练数据发送到分支代理，这样减轻了分支代理的训练强度。

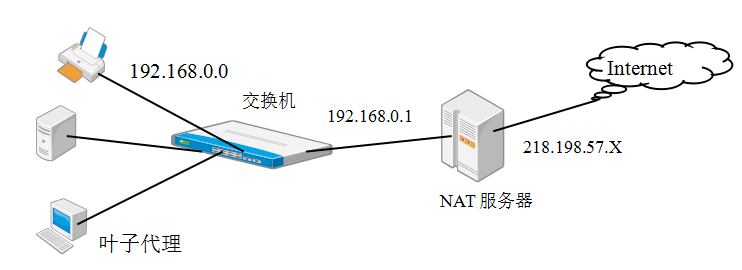


图3-3 叶子代理设计

3.3.2 分支代理的设计

如图3-4所示，分支代理分布在各NAT服务器上，分支代理设计相对复杂，负责收集对代理宿机服务器进行的攻击，代理服务器上安装有杀毒软件，病毒库升级比较及时，可以阻挡99%以上的蠕虫、ARP、木马攻击等，但有些代理服务器设置有策略，比如在某些时间段无法访问QQ、视频等，学生为了突破此限制，经常攻击叶子代理服务器，相对来说压力比较大，此代理收集未知的攻击并进行训练，分支代理还负责处理叶子代理传来的训练结果。

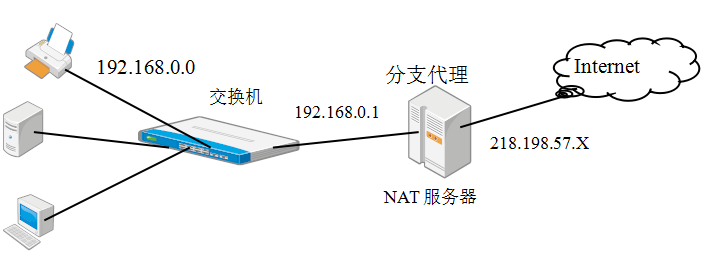


图3-4 分支代理设计

3.3.3 主干代理的设计

如图3-5所示，主干代理分布在各种应用服务器上，代理设计相对复杂，负责收集对代理宿机服务器进行的攻击，代理服务器上安装有杀毒软件，病毒库升级比较及时，可以阻挡99%以上的蠕虫、ARP、木马攻击等，但有代理服务器设置有策略，比如在某些时间段无法访问QQ、视频等，学生为了突破此限制，经常攻击叶子代理服务器，相对来说压力比较大，此代理收集未知的攻击并进行训练，分支代理还负责处理叶子代理传来的训练结果。

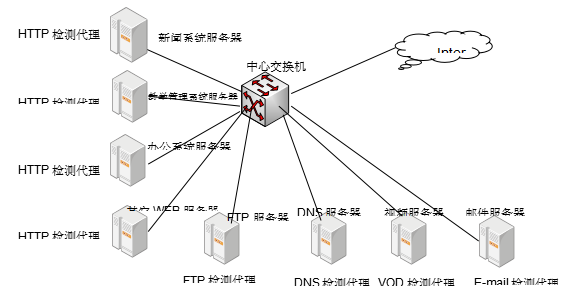


图3-5 主干代理设计

3.3.4 代理分层功能设计

免疫代理分为三个层次，每个层次的代理功能相对独立，叶子代理收集并训练入侵数据，并将训练结果送分支代理，这样可以有效减轻分支代理的负担，分支代理所在的宿主（NAT服务器）主机主要负责高速内网和外网的数据转发。分支代理如果有过多的训练任务，就会影响系统的内外网的数据转发速度，进而影响到系统的整体健康指数。

3.4 本章小结

本章首先介绍了系统需求分析，目的是将生物免疫系统和神经网络系统的并行处理能力、学习能力、记忆能力应用到入侵检测系统中。然后进行了设计了分层的系统总体模型。最后设计了叶子代理、分支代理和主干代理。最后介绍了每层代理的功能如何设计才能保证系统的健康指数最高。每个层次的代理功能相对独立，叶子代理收集并训练入侵数据，并将训练结果送分支代理，这样可以有效减轻分支代理的负担，分支代理所在的宿主（NAT服务器）主机主要负责高速内网和外网的数据转发。

4 详细设计和实现

4.1 代理捕获和过滤数据包的设计

1）捕获（Sniffer）模块设计

捕获（Sniffer）模块可以将每时每刻网卡收发的数据一并捕获并解析成用户可以接受的直观表示形式，捕获的信息包括：源IP地址、目的IP地址、协议类型、源端口、目的端口、数据包长度、等。选择任意一条数据都可以在右边显示该数据包在IP层的保存新式，并以IP包的格式显示，显示内容有：版本信息，包头长度信息，服务类型信息，数据报总长度信息，标识信息，分割片段起始偏移量信息，TTL信息，传输层协议类型信息，数据报头校验信息，起始IP信息，目标IP信息。目前Snifer模块可以解析的协议类型包括：IP ，ICMP，IGMP，GGP，IPV4，TCP，PUP，UDP，IDP，IPV6，ROUTING，FRAGMENT，ESP，AH，ICMPV6，NONE，ND，RAW。

2）程序实现

（1）创建一个Socket。

（2）获取本机IP地址并进行绑定。

（3）设置网卡为混杂模式以便能截获任何通过本机网卡的数据包，包括本机的和非本机的。

（4）增加一个新线程，新线程通过一个死循环来进行实施接收数据。

（5）主线程进行显示并处理已截获的数据包。

使用多线程的好处是可以实时显示抓包情况。而不像某些软件必须当抓包结束后才能显示出数据。

3）具体核心代码

// 创建监听套接字

SOCKET MonSock = socket ( AF\_INET, SOCK\_RAW, IPPROTO\_IP ) ;

// 绑定地址信息到套接字

if ( bind ( MonSock, (sockaddr\*)&LocalAddr, sizeof(sockaddr) ) == SOCKET\_ERROR )

return 0 ;

// 设置为混杂模式，收所有IP包

DWORD dwValue = 1 ;

if ( ioctlsocket ( MonSock, SIO\_RCVALL, &dwValue ) != 0 )

return 0 ;

// 检测控制标志，是否继续监视

while ( pDlg->isMonitor )

{

// .....

}

需要注意的是，用此种方法捕获的数据都是char\*型的字符数据，需要将其装换成为所需的数据格式。幸运的是，微软提供的接口将这些字符是按照协议类型字段长度按顺序分配在内存中的，所以只需要设计好接收数据结构便可以直接强制转换为我们想要的数据，可是，微软在MSDN中并没有对这些协议的数据结构进行声明，所以需要我们手动声明，下面就以IP协议为例进行代码介绍。

// 定义IP报头格式

typedef struct \_IP\_HEADER{

BYTE bVerAndHLen; // 版本信息（前位）和头长度（后位）

BYTE bTypeOfService; // 服务类型

USHORT nTotalLength; // 数据包长度

USHORT nID; // 数据包标识

USHORT nReserved; // 保留字段

BYTE nTTL; // 生成时间

BYTE bProtocol; // 协议类型

USHORT nCheckSum; // 校验和

UINT nSourIp; // 源IP

UINT nDestIp; // 目的IP

}IP\_HEADER,\*PIP\_HEADER;

// 定义我们关心的数据结构

typedef struct \_PACK\_INFO{

USHORT nLength; // 数据包长度

USHORT nProtocol; // 协议类型

UINT nSourIp; // 源IP

UINT nDestIp; // 目的IP

USHORT nSourPort; // 源端口号

USHORT nDestPort; // 目的端口号

}PACK\_INFO,\*LPPACK\_INFO;

// 取得数据包

nRecvSize = recv ( MonSock, szPackBuf, DEF\_BUF\_SIZE2, 0 ) ;

if ( nRecvSize > 0 )

{

// 解析IP包头

PIP\_HEADER pIpHeader = (PIP\_HEADER)szPackBuf ;

PackInfo.nLength = nRecvSize ;

PackInfo.nProtocol = (USHORT)pIpHeader->bProtocol ;

PackInfo.nSourIp = pIpHeader->nSourIp ;

PackInfo.nDestIp = pIpHeader->nDestIp ;

UINT nIpHeadLength = ( pIpHeader->bVerAndHLen & 0x0F ) \* sizeof(UINT) ; }

// 我们可以将IP包头进行进一步转化

// 这里以TCP和UDP为例，只检测TCP和UDP包

switch ( pIpHeader->bProtocol )

{

case IPPROTO\_TCP:

{// 取得TCP数据包端口号

PTCP\_HEADER pTcpHeader = (PTCP\_HEADER)&szPackBuf[nIpHeadLength] ;

PackInfo.nSourPort = pTcpHeader->nSourPort ;

PackInfo.nDestPort = pTcpHeader->nDesPort ;

pDlg->AddPackInfo ( &PackInfo ) ; // 输出到界面

}

break ;

case IPPROTO\_UDP:

{// 取得UDP数据包端口号

PUDP\_HEADER pUdpHeader = (PUDP\_HEADER)&szPackBuf[nIpHeadLength] ;

PackInfo.nSourPort = pUdpHeader->nSourPort ;

PackInfo.nDestPort = pUdpHeader->nDestPort ;

pDlg->AddPackInfo ( &PackInfo ) ; // 输出到界面

}

break ;

default:

{//取得其他协议数据包

PackInfo.nSourPort = 0;

PackInfo.nDestPort = 0;

pDlg->AddPackInfo(&PackInfo);

}

break;

}

4.2 数据的保存和显示

有了以上的数据，配合Timer计时器便可以方便的将数据以走势图和IP报文格式化的方式显示到界面了。这里采用了MFC提供的GDI+接口直接进行绘图操作。

传统的Sniffer程序只能对数据包进行捕获，但是却无法知道具体是由哪个活动在网络上的进程正在收发数据。而SockMon却可以做到这一点，它可以使用分层服务提供者LSP截取网络数据包。

1）服务提供者接口

服务提供者接口（Service Provider Interface，简称SPI）为需要网络通信的应用程序提供服务。SPI由两个部分组成：传输提供者（Transport Service Provider）和命名空间服务提供者（Name Spcae Provider）。

（1）传输提供者（Transport Service Provider）

传输提供者是提供建立连接，数据传输，出错控制的服务。它有如下两种类型。

① 基础服务提供者（Base Service Provider,简称BSP）

负责实现传输协议的实现，导出WinSock接口，此接口直接实现协议。

② 分层服务提供者（Layered Service Provider，简称LSP）

可以安装到基础服务提供者上面，截取应用程序的Winsock API调用。可以同时安装多个LSP。

（2）命名空间服务提供者（Name Spcae Provider）

此处不用，略去。

2）LSP截取网络数据原理

程序目标是截取不同协议的数据包，数据信息分为三个层次：

第一层：进程信息，包括进程网络数据通信的进程ID和进程路径全名。

第二层：套接字信息，包括套接字的协议类型，套接字的句柄和创建时间。

第三层：操作信息，包括传输操作所使用的接口，数据通信双方的地址，数据传输量等。

启动程序SockMon时首先会安装LSP，而当SockMon结束时会自动卸载LSP。在成功安装LSP后，一旦应用程序进行网络通信，就会把LSP模块加载到当前进程。此时，当前进程的所有网络通信的数据或操作都可以被LSP截取。程序把截取的数据保存在文件SockInfo.data,以内存映射文件的形式进行操作，如图4-1所示。

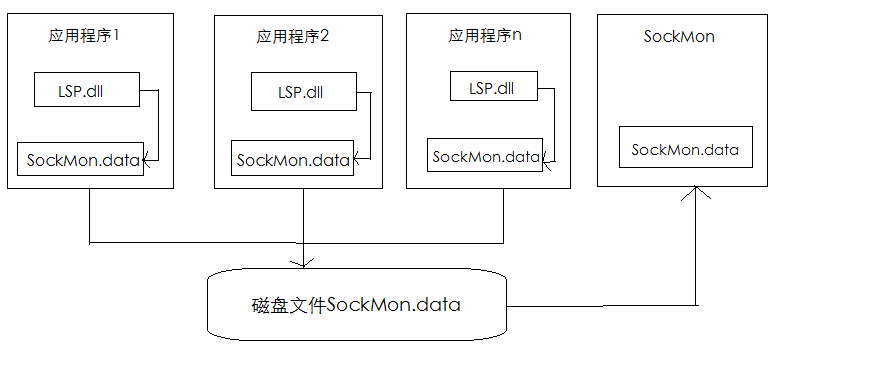


图4-1 使用LSP截取网络数据原理图

当有多个应用程序进行网络通信时，各自会把LSP.dll加载到进程。当LSP.dll被加载后，会创建SockInfo.data的内存映射文件。当应用程序进行网络通信时，LSP截取网络数据与操作并保存在SockData.data。与此同时，LSP向客户端发送一个通知消息，表明数据文件已被更新，这时SockMon就读取数据并显示到界面。

3）LSP模块数据信息结构

// 定义进程信息结构

typedef struct \_PROCESS\_INFO {

DWORD dwProcessId ; // 进程ID

WCHAR lpPath[MAX\_PATH\_LENGTH] ; // 进程路径全名

} PROCESS\_INFO, \*LPPROCESS\_INFO ;

// 定义套接字信息结构

typedef struct \_SOCKET\_INFO {

SOCKET s ; // 套接字句柄

int protocol ; // 协议类型

DWORD dwProcessId ; // 进程ID

SYSTEMTIME CreateTime ; // 创建时间

} SOCKET\_INFO, \*LPSOCKET\_INFO ;

根据上面的定义，可以知道实例内部处理的数据结构如图4-2。

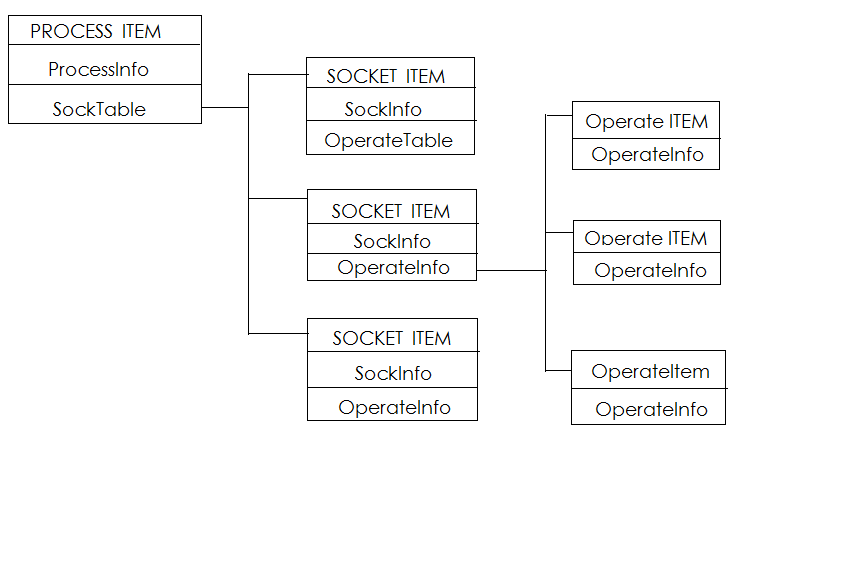


图4-2 内部处理的数据结构

当数据保存到磁盘文件SockInfo.data时，保存的是PROCESS\_INFO, SOCKET\_INFO和OPERATE\_INFO类型的数据，每一个结构的数据被视为一天记录。为了方便数据存取过程，实例中把每个记录的长度固定为4K，另外，在SockInfo.data中还保存着数据信息的总量，记为nCount。由于实力运行时可能有多个线程同时加载LSP模块并截取网络数据，可能会造成多个进程同时针对变量nCount进行操作，此时需要同步处理。

4）目录枚举

（1）SPI提供的协议

SPI提供的协议共有三个。

① 分层协议：在基础协议之上，是靠基础协议提供更高级的网络通信服务。

② 基础协议：它是相对于分层协议而言的，是能够安全、独立地和远程端点实现数据通信协议。

③ 协议链：将一系列的分层协议和基础协议特定结构连接在一起的链状结构。

在安装LSP和开发LSP过程中都需要枚举协议目录，可以使用SPI提供的函数WSCEnumProtocol进行枚举，定义如下：

int WSCEnumProtocols(IN LPINT lpiProtocols, \_\_out\_bcount\_opt);

如果枚举成功，返回协议数量。其中WSAPROTOCOL\_INFOW结构包含了服务提供者所支持的协议，地址家族以及套接字相关的信息。

（2）NativeAPI结构

本机系统服务NativeAPI，又称本机应用程序编程接口，它是由执行体为用户模式和内核模式的程序提供的系统服务集，从用户模式调用Native API是通过ntdll.dll实现的。通常情况下使用Win32函数一般是由kernel32.dll或gdi32.dll等库所导出的，表面上Win32函数为编程人员提供了很多接口来实现想要的功能，这些都仅仅是对Native API进行封装后提供的给用户使用的。

Native API本身也并不实现具体的功能，在Win NT以后的平台由于用户模式和内核模式的隔离，正常情况下普通应用程序无法进入内核模式，更不用谈及调用内核执行体所提供的功能，于是就有了现在的系统服务调用。

然而Native API跟普通Win32 API不同的是，所有东西都是未被导出的。如果使用Win32 API可以通过MSDN查询函数定义、参数说明、使用方法、注意事项，有时甚至还有源码实例。

① 调用步骤

在用户层调用Native API，主要分为3个步骤。

定义Native API函数NtQuerySystemInformation，用来枚举进程信息。

使用GetProcAddress取得NtQuerySystemInformation函数地址。

调用Native API函数NtQuerySystemInformation取得进程信息。

进程/线程/模块的查杀采用内核驱动“清0法”，每个“模块”句柄在应用层的调用都会引起内核对象计数的累加，同理句柄在应用层的释放也会引起内核对象计数的递减，当操作系统发现某个内核对象的计数为0时，就会清除该内核对象，而“清0法”则是通过驱动程序获得内核最高权限直接强制将内核对象计数清0，从而达到查杀的目的。

② 进程虚拟内存查看模块

取得与虚拟内存相关的系统信息，“页大小”，“最小地址”，“最大地址”。使用GetSystemInfo用于取得系统相关信息，定义如下：

void GetSystemInfo( \_\_out LPSYSTEM\_INFO lpSystemInfo );

取得内存状态信息，包括“总物理内存”，“可用物理内存”，“总页文件”，“可用页文件”，“总虚拟内存”和“可用虚拟内存”。使用GlobalMemoryStatus函数用于取得系统当前内存信息，定义如下：

void GlobalMemoryStatus( \_\_out LPMEMORYSTATUS lpBuffer );

查询指定进程的虚拟地址状态信息，使用VirtualQueryEx函数，定义如下：

SIZE\_T VirtualQueryEx(

\_\_in HANDLE hProcess,

\_\_in\_opt LPCVOID lpAddress,

\_\_out\_bcount\_part

);

4.3 系统健康指数

4.3.1 叶子工作站健康指数

每个工作站都有一个健康指数，工作站健康指数由受CPU、内存和网络影响，设定最小为0，最大为100。计算公式如下：



其中CPU是指宿主计算机CPU的使用率，MEM是指物理内存使用量（单位M），startMEM是指系统启动时使用的物理内存（单位M），totalMEM是指物理内存总量（单位M），NETWORK是指网卡使用率。

CPU、内存和网络使用等参数权重分别为60、20和20，CPU使用率是最主要参数，对系统性能影响较大，当CPU使用率超过80%时，系统几乎处于停滞状态，此时我们认为其健康指数为0。

4.3.2 分支服务器健康指数

每个NAT服务器健康指数由受CPU、内存和双网卡影响，设定最小为0，最大为100。计算公式如下：



其中CPU是指宿主服务器CPU的使用率，MEM是指物理内存使用量（单位M），startMEM是指系统启动时使用的物理内存（单位M），totalMEM是指物理内存总量（单位M），inNET是指内网卡使用率，outNET是指外网卡使用率。

CPU、内存、内网卡和外网卡使用等参数权重分别为60、10、20和10，CPU使用率是最主要参数，对系统性能影响较大，当CPU使用率超过80%时，系统几乎处于停滞状态，此时我们认为其健康指数为0。

4.3.3 主干服务器健康指数

每个服务器（包括WEB、FTP、DNS、E-mail、VOD等服务器）都有一个健康指数，服务器健康指数由受CPU、内存和网络影响，设定最小值为0，最大为100。计算公式如下：



因采用私有云构建服务器系统，因此系统中的CPU和内存都是虚拟的，其中CPU是虚拟服务器CPU的使用率，MEM是指物理内存使用量（单位M），startMEM是指系统启动时使用的物理内存（单位M），totalMEM是指物理内存总量（单位M），NETWORK是指网卡使用率（一般认为使用的是单网卡）。

CPU、内存和网络使用等参数权重分别为20、20和60，网卡使用率是最主要参数，对系统性能影响较大，当网卡使用率超过80%时，系统可能遭受DDOS等十分严重的攻击，系统对外服务几乎处于停滞状态，此时我们认为其健康指数为0，提醒系统必须采取积极有效的措施。

4.3.4 系统健康指数

整个网络系统运行是否通畅受到包括各种应用服务器、NAT服务器、工作站和网络传输设备的影响，但影响程度不同，由高到低排列顺序为应用服务器、NAT服务器、网络传输设备和工作站，因此计算系统健康指数是权重设置分别为40、30、20和10。健康指数计算公式是：



其中index\_trunk、index\_branch、CPU、index\_leaf分别为主干健康指数、分支健康指数、网络设备CPU使用率、叶子健康指数等，S是虚拟应用服务器的数量，N是NAT服务器的数量，R是网络设备的数量，C是工作站的数量。

整个系统的健康指数是应用服务器、NAT服务器、网络设备和工作站健康指数的加权平均，系统健康指数最小为0，最大为100。系统开始运行是系统指数为100，当系统正常运行和遭受攻击时都会是健康指数降低，表4-1显示健康指数和健康程度的关系。

表4-1 系统健康指数表

序号 健康等级 健康指数 备注

1 非常健康 90以上 防护措施得力

2 健康 75至90 可能工作站防护措施不力

3 亚健康 60至75 查找服务器和网络设备漏洞

4 不健康 60以下 遭受较强攻击，系统防护需要全面升级

4.4 系统的测试

系统的测试就是指将工作站、代理服务器主机参数，接入层、汇聚层、核心层网络参数，私有云中各服务器主要参数按照加权平均的方法计算整个网络系统的健康值。

4.4.1 系统实验和测试拓扑

系统设计和实现完成之后，需要做简单的软件测试。如图4-3所示的就是无分层代理的测试结构。

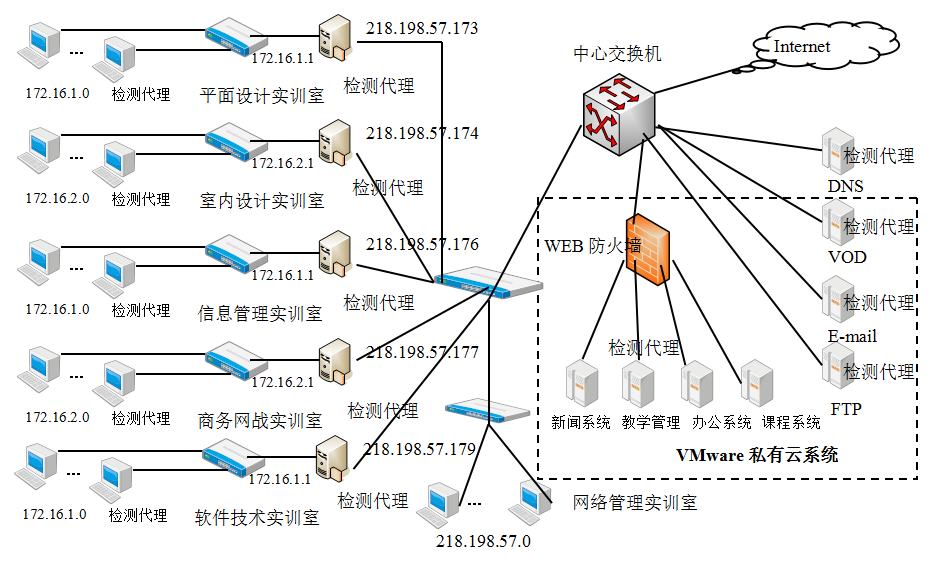


图4-3 无分层代理测试

如图4-4所示的就是分层代理的测试结构。测试内容为磁带记录档案管理系统的磁带记录档案及相关信息录入和数据库信息管理功能点。

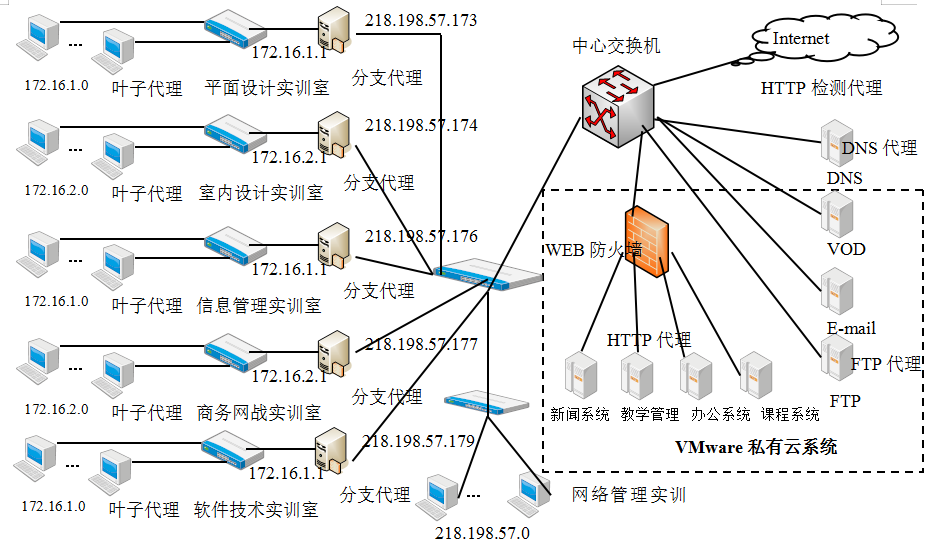


图4-4 分层代理测试

表4-2所示的就是参与测试的实训室的软件和硬件设备的详细列表。

表4-2 测试软硬件

|  |  |  |
| --- | --- | --- |
| 实训室 | 设备情况 | 担当角色 |
| 平面设计实训室 | 30台计算机 | 安装入侵检测代理、进行入侵检测 |
| 6台计算机 | 2台计算机入侵NAT服务器，4台计算机入侵其它计算机 |
| 1台NAT服务器 | 代理上网，安装入侵检测代理 |
| 室内设计实训室 | 30台计算机 | 安装入侵检测代理、进行入侵检测 |
| 6台计算机 | 2台计算机入侵NAT服务器，4台计算机入侵其它计算机 |
| 1台NAT服务器 | 代理上网，安装入侵检测代理 |
| 信息管理实训室 | 60台计算机 | 安装入侵检测代理、进行入侵检测 |
| 12台计算机 | 6台计算机入侵NAT服务器，6台计算机入侵其它计算机 |
| 1台NAT服务器 | 代理上网，安装入侵检测代理 |
| 商务网站实训室 | 60台计算机 | 安装入侵检测代理、进行入侵检测 |
| 12台计算机 | 6台计算机入侵NAT服务器，6台计算机入侵其它计算机 |
| 1台NAT服务器 | 代理上网，安装入侵检测代理 |
| 软件技术实训室 | 40台计算机 | 安装入侵检测代理、进行入侵检测 |
| 10台计算机 | 5台计算机入侵NAT服务器，5台计算机入侵其它计算机 |
| 1台NAT服务器 | 代理上网，安装入侵检测代理 |
| 网络管理实训室 | 48台计算机 | 6台入侵新闻系统，6台入侵教学管理系统，6台入侵办公自动化系统，6台入侵部门网站，6台入侵DNS系统，6台入侵VOD系统，6台入侵E-mai系统，6台入侵FTP系统 |
| 1台交换机 | 汇聚层交换机，锐捷三层S5750交换机，负责高速转发数据包到网络中心的锐捷S8600交换机上 |
| 网络中心 | 1台服务器 | DNS服务器，安装入侵检测代理，进行入侵检测 |
| 1套VMware虚拟系统 | 安装7套Windows server 2003或Windows Server 2008操作系统，分别安装7套服务（新闻系统、教学管理系统、办公自动化、各部门网站、VOD系统、E-mail系统、FTP系统、课程网站系统等） |
| 1台交换机 | 汇聚层交换机，锐捷三层S5750交换机，负责高速转发数据包到网络中心的锐捷S8600交换机上 |

4.4.2 无分层代理测试

2019年3月9日组织计算机网络技术专业81名学生进行无分层代理测试。被入侵的计算机安装同样的代理，接受测试，攻击设备情况见表4-3。

表4-3 无分层代理测试

|  |  |  |  |
| --- | --- | --- | --- |
| 实训室 | 使用设备 | 入侵目标 | 入侵工具 |
| 平面设计实训室 | 4台计算机 | 同实训室其它计算机 | 1.扫描工具（scanPOrt、superscan、nmap等）  2.上传工具（IIS put scanner、Nc.exe等）  3. 暴力破解工具（superdic.exe、Mysql\_pwd\_crack、X-Scan、SQLScanPass.exe、SMBcrack）  4.溢出类工具（Ms-08067、  Framework-3.2等）  5. Sql注入工具（阿D 、明小子、  蚂蚁注入工具）  6. Cookie注入工具（Cookie&Inject.exe、穿山甲cookie注入工具等）  7. 远程访问类工具（Putty、SecureCRT等） |
| 2台计算机 | NAT服务器 |
| 室内设计实训室 | 4台计算机 | 同实训室其它计算机 |
| 2台计算机 | NAT服务器 |
| 信息管理实训室 | 6台计算机 | 同实训室其它计算机 |
| 6台计算机 | NAT服务器 |
| 商务网站实训室 | 6台计算机 | 同实训室其它计算机 |
| 6台计算机 | NAT服务器 |
| 软件技术实训室 | 5台计算机 | 同实训室其它计算机 |
| 5台计算机 | NAT服务器 |
| 网络管理实训室 | 4台计算机 | DNS服务器 |
| 4台计算机 | 新闻服务器 |
| 4台计算机 | 教学系统服务器 |
| 4台计算机 | 办公自动化服务器 |
| 4台计算机 | VOD服务器 |
| 4台计算机 | 邮件服务器 |

此次无分层代理测试的健康指数波形图如图4-5所示。

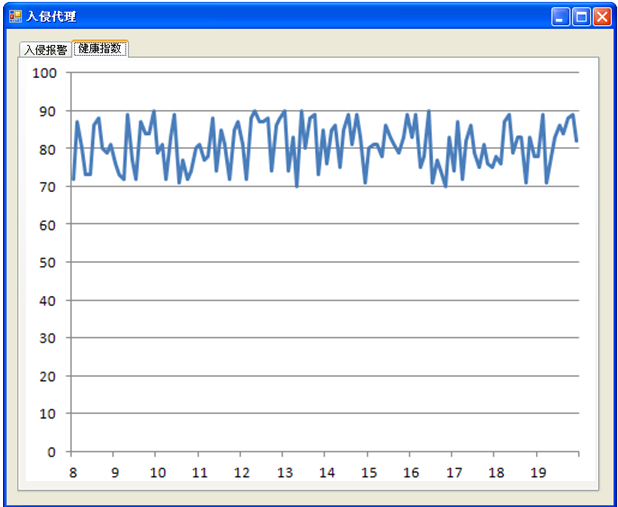


图4-5 入侵代理健康指数

4.4.3 分层代理测试

2019年3月10日组织计算机网络技术专业81名学生进行分层代理测试。被入侵的计算机安装不同的代理，而此次测试的攻击设备情况见表4-4。

表4-4分层代理测试

|  |  |  |  |
| --- | --- | --- | --- |
| 实训室 | 使用设备 | 入侵目标 | 入侵工具 |
| 主控室 | 8台服务器 | 服务器 | 1.扫描工具（scanPOrt、superscan、nmap等）  2.上传工具（IIS put scanner、Nc.exe等）  3. 暴力破解工具（superdic.exe、Mysql\_pwd\_crack、X-Scan、SQLScanPass.exe、SMBcrack）  4.溢出类工具（Ms-08067、  Framework-3.2等）  5. Sql注入工具（阿D 、明小子、  蚂蚁注入工具）  6. Cookie注入工具（Cookie&Inject.exe、穿山甲cookie注入工具等）  7. 远程访问类工具（Putty、SecureCRT等） |
| 平面设计实训室 | 4台计算机 | 同实训室其它计算机 |
| 2台计算机 | NAT服务器 |
| 室内设计实训室 | 4台计算机 | 同实训室其它计算机 |
| 2台计算机 | NAT服务器 |
| 信息管理实训室 | 6台计算机 | 同实训室其它计算机 |
| 6台计算机 | NAT服务器 |
| 商务网站实训室 | 6台计算机 | 同实训室其它计算机 |
| 6台计算机 | NAT服务器 |
| 软件技术实训室 | 5台计算机 | 同实训室其它计算机 |
| 5台计算机 | NAT服务器 |
| 网络管理实训室 | 4台计算机 | DNS服务器 |
| 4台计算机 | 新闻服务器 |
| 4台计算机 | 教学系统服务器 |
| 4台计算机 | 办公自动化服务器 |
| 4台计算机 | VOD服务器 |
| 4台计算机 | 邮件服务器 |

此次分层代理测试的健康指数波形图如图4-6所示。

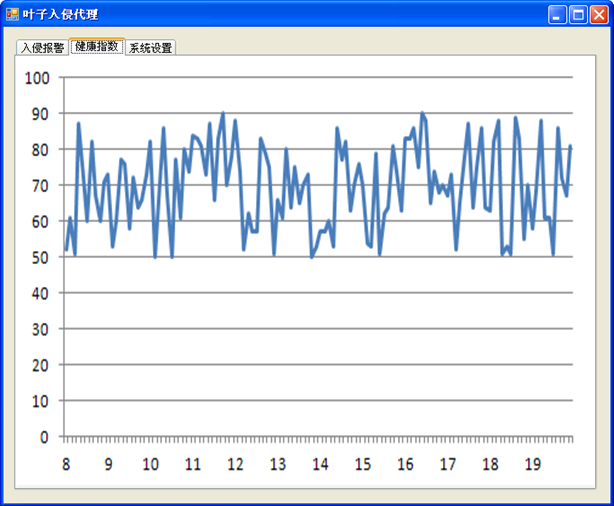


图4-6 叶子代理健康指数

此次分层代理测试的叶子代理的系统设置如图4-7所示。

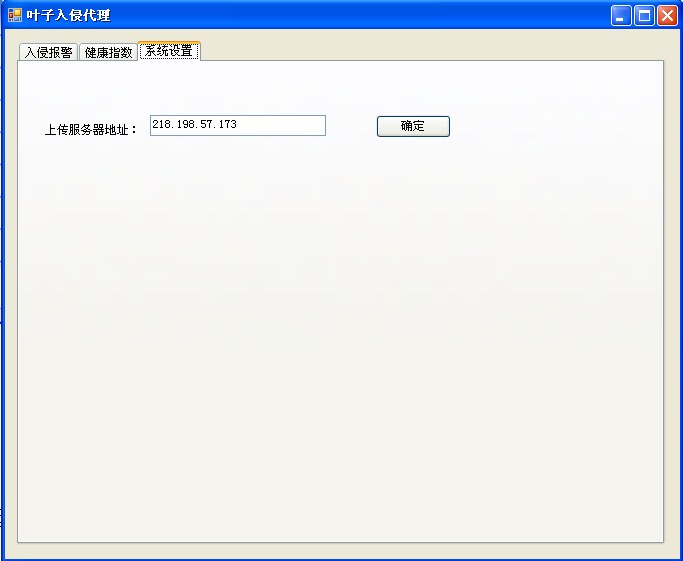


图4-7 系统设置

此次分层代理测试的分支代理健康指数如图4-8所示。

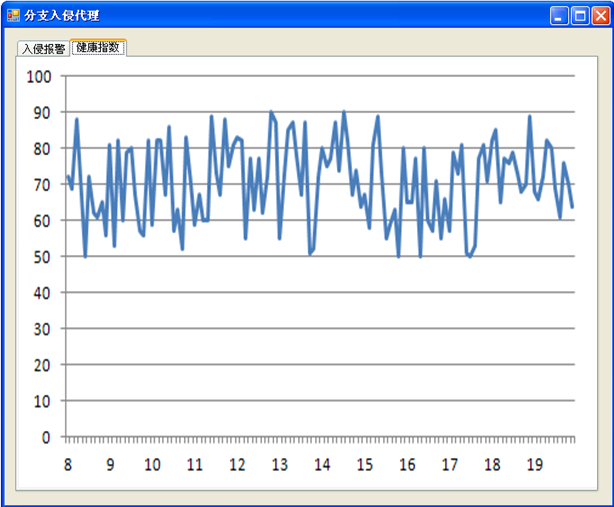


图4-8 分支代理健康指数

此次分层代理测试的DNS代理健康指数如图4-9所示。

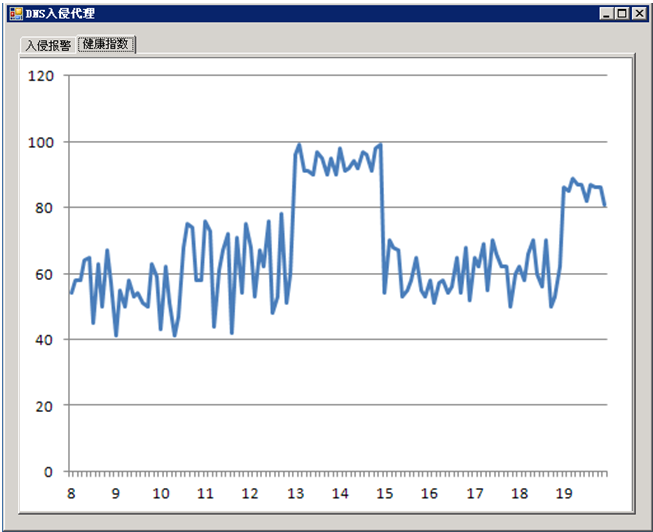


图4-9 DNS代理健康指数

4.4.4 健康指数对比

通过两次的测试，我们可以看出明显的效果。在无分层代理测试中，如图4-10所示无分层系统的健康指数在60左右，相对比较低。

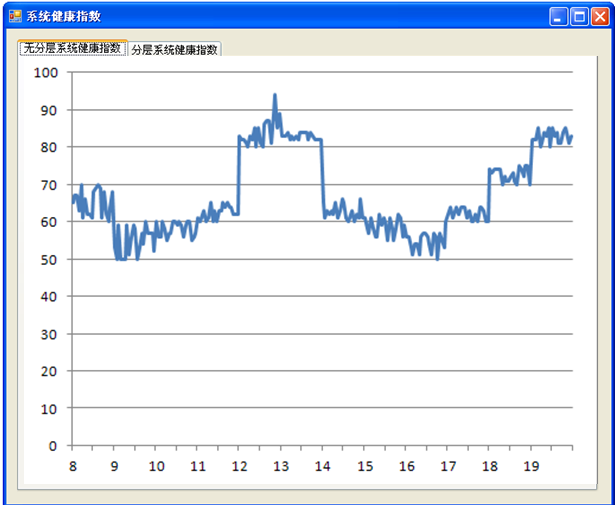


图4-10 无分层系统健康指数

而在分层代理测试中，如图4-11所示分层系统的健康指数在90左右，相比无分层系统的健康指数要高了很多。

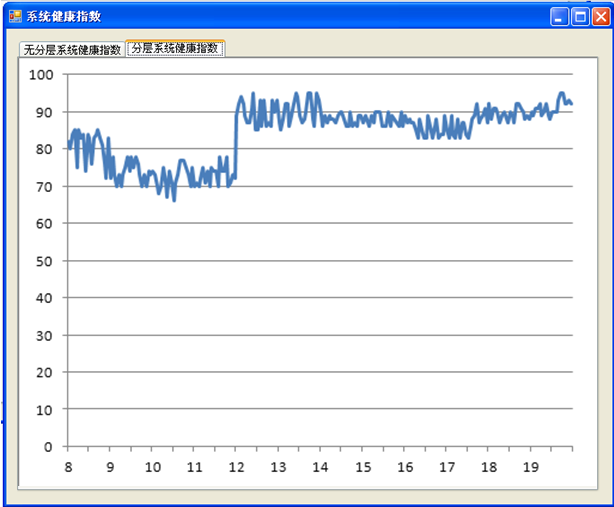


图4-11 分层系统健康指数

4.5 本章小结

将工作站、代理服务器主机参数，接入层、汇聚层、核心层网络参数，私有云中各服务器主要参数按照加权平均的方法计算整个网络系统的健康值。进行了两种对比试验，一是不分层的免疫代理系统，发现健康指数偏低；二是分层的免疫代理系统，健康指数较高。

5 总结与展望

5.1 总结

本文在前人研究成果的基础上， 对基于免疫原理的网络入侵检测技术做了进一步的研究与探讨，并提出分层代理模型，分析了此模型对网络健康指数的影响。本文得出的结论有以下几个方面：

(1) 结合免疫入侵检测系统和分布式入侵检测系统的优点，本文设计了一个基于分层免疫代理入侵检测系统模型，免疫代理分布在普通的工作站、代理服务器和专用服务器上，分别称为叶子代理、分支代理和主干代理，各司其职。

(2) 对叶子代理、分支代理和主干代理功能进行了设计，其中在普通工作站上的叶子代理负责数据收集和初级免疫训练，并将数据和训练结果交给分支代理；在代理服务器上的分支代理收集叶子代理传过来的数据，使用免疫算法进行训练，将训练结果发送到服务器；主干代理负责对服务器的入侵检测，任务比较重，为了提高效率根据不同的服务设计有DNS代理、HTTP代理、FTP代理等，各种代理只收集相应的服务数据，并进行免疫训练、免疫细胞繁殖，结果发送到数据库服务器。

(3) 将工作站、代理服务器主机参数，接入层、汇聚层、核心层网络参数，私有云中各服务器主要参数按照加权平均的方法计算整个网络系统的健康值。进行了两种对比试验，一是不分层的免疫代理系统，发现健康指数偏低；二是分层的免疫代理系统，健康指数较高。

5.2 展望

基于生物免疫算法的入侵检测系统是一个复杂的自适应、自我发展的系统， 本文只将已有的人工免疫算法上应用到一个复杂的网络系统中， 设计了一个分层代理入侵检测系统。然而， 如何增强入侵检测系统的工作效率， 还有很多地方值得研究和提高，准备在以下几个方面继续努力：

(1) 继续跟踪基于生物免疫学的入侵检测算法研究成果，将新的算法（比如支持向量机、神经网络等）应用到入侵检测中；

(2) 本文虽然将分布式技术和方法应用到入侵检测系统的研究中，但是比较初级，希望在未来对这一方面做更多的研究；

(3) 本模型的研究只是处于实验模拟阶段，下一步将继续研究代理之间如何更加有效进行对话，做更加实时和有效的网络入侵检测系统。

致 谢

完成本论文，首先要感谢我的导师王教授，从一开始的选定研究方向和开题、中期检查到论文的反复修改、润色，直到最后完成论文的撰写，王教授始终认真负责地给予我深刻而细致地指导，非常感谢王教授对我的一次次指导和帮助！王教授不但在我研究的过程中给与了宝贵的意见，还一直给我引导和鼓励。对于我遇到的难题，教授会耐心细致的逐个解决，并且提出相关的问题，使我可以及时发现错误进行纠正，使我的论文可以顺利、圆满的完成。

在整个过程中，王教授广博的学识、认真负责的工作态度都给我留下了深刻的印象，在之后的工作中，我也会时刻以王教授为榜样，认真、努力的完成每一件事。最后，再次对老师致以最真切的感谢！

此外，我还想感谢我的朋友、同学和同事，是他们对我的关心、帮助和支持，使我能够顺利的度过硕士学习的这些日子，能够在今后的人生中留下一段难忘的岁月。我衷心的谢谢他们，同时也祝愿他们幸福平安！

参考文献

[1]中国互联网络信息中心.中国互联网络发展状况统计报告[R].北京：中国互联网络信息中心， 2019：4-5.

[2]Anderson J E Computer security thread monitoring and survefllance[R]．Fort Washington,PA：Jame P,2010:1-5

[3]赵林惠主编. 基于人工免疫原理的检测系统模型及其应用[M]. 北京：清华大学出版社， 2016:123-143

[4]龚非力主编. 医学免疫学(第4版) [M]. 北京：科学出版社， 2019:1-30

[5]Joseph Migga Kizza编著. 计算机网络安全概论[M].北京：电子工业出版社， 2012:65-78

[6]李剑编著.入侵检测技术[M].北京：高等教育出版社， 2008:12-56

[7]郑成兴著.网络入侵防范的理论与实践[M].北京：机械工业出版社， 2006:125-146

[8]薛静锋等主编.入侵检测技术[M].北京：人民邮电出版社， 2016:77-98

[9]胡昌振编著. 网络入侵检测原理与技术（第2版）[M].北京：北京理工大学出版社， 2016:45-102

[10]Jinquan Zeng,Xiaojie Liu,Tao Li,Caiming Liu,Lingxi Peng,Feixian Sun Department of Computer Science,Sichuan University,Chengdu 610065,China. A self-adaptive negative selection algorithm used for anomaly detection[J]. Progress in Natural Science. 2009(02):28-33

[11]Zhou Ji,Dasgupta D.V-Detector:an efficient nega-tive selection algorithm with"probably adequate"detector coverage. Journal of Information Science . 2009:45-51

[12]李海. 否定选择算法在IDS中的应用研究[D]. 电子科技大学 2009:23-36

[13]Bereta M,Burczy′nski T.Comparing binary and re-al-valued coding in hybrid immune algorithm for fea-ture selection and classification of ECG signals. Engineering Applications of Artificial Intelligence . 2008:1021-1027

[14]Aydin I,Karakose M,Akin E.Chaotic-based hybrid negative selec-tion algorithm and its applications in fault and anomaly detection. Expert Systems With Applications . 2010:76-87

[15]Xu B,Luo W,Pei X, et al.On Average Time Complexity of Evolutionary Negative Selection Algorithms for Anomaly Detection. Proceedings of the 2009 World Summit on Genetic and Evolutionary Computation . 2009:102-105

[16]CHEN Y B,FENG C,ZHANG Q,et al.Negative selection algorithmwith variable-sized r-contiguous matching rule. International Con-ference on Progress in Informatics and Computing . 2010:78-86

[17]冯艳华. 基于免疫非我学习算法的入侵检测模型及方法研究[D]. 广西大学 2016:25-38

[18]张译云. 浅谈云计算的发展现状及应用[J]. 中国新通信. 2017(23): 3-5

[19]郭春梅，毕学尧，杨帆. 云计算安全技术研究与趋势[J]. 信息网络安全. 2010(04): 16-17

[20]Sims K.IBM introduces ready-to-use cloud computing collaboration services get clients started with cloud computing. http://www-03.ibm.com/press/us/en/pressrelease/22613.wss . 2007:26-32

[21]Boss G,Malladi P,Quan D,Legregni L,Hall H.Cloud computing. IBM White Paper . 2007:77-85

[22]黄昊晶，崔志明. 一种以vSpher为核心的私有云基础架构设计方案[J]. 微电子学与计算机. 2011(04) 38-41

[23]微软的私有云解决方案[J]. 中国新技术新产品. 2011(06): 14-15

[24]]刘菲，张波. 浅谈中小企业私有云计算解决方案[J]. 硅谷. 2010(14): 141

[25]杨均隆，俞鹤伟. 一个基于免疫机理的分布式入侵检测系统[J]. 微计算机信息. 2010(21): 42-43+94

[26]叶清，陈亚莎，黄高峰. 基于粗糙集和证据推理的网络入侵检测模型[J]. 计算机工程. 2011(05): 164-166

[27]王雷. 敌意规划模型在分布式入侵检测中的设计与研究[J]. 信息科技. 2011(21):14

[28]陈雪斌，刘峰，赵志宏，骆斌. 安全的分布式入侵检测系统框架[J]. 计算机工程与设计. 2009(14): 3266-3268+3277

[29]李春玲. 基于Agent的分布式入侵检测[D]. 北京：北京理工大学. 2013: 421-422+431

[30]马常楼，刘永庆. 一种基于多Agent的分布式入侵检测系统设计[J]. 计算机与数字工程. 2009(06): 102-106