摘　要

随着互联网的快速发展，越来越多的应用通过网络来实现，同时网络的安全也面临着巨大的挑战。快速的网络为攻击者提供了方便，攻击模式和方法越来越复杂，攻击者的技术水平也在不断提高，攻击规模日益扩大，越来越多的系统受到攻击，如何保护系统与可信网络不受入侵成为目前迫切需要解决的问题。传统的网络安全措施，如防火墙或入侵检测技术(IDS)显得力不从心，这就需要引入一种全新的主动入侵防护 （Intrusion Prevention System，IPS）技术。入侵防护作为一种主动的安全防护手段，它的优势体现在如下两方面：首先它可以在攻击发生前主动阻断它们，它不仅可进行检测，还能在攻击造成损坏前阻断它们；另一方面在检测到攻击时就需要具备相应的响应能力，入侵防护系统在成功的检测到攻击事件后，它可以记录攻击者的详细的攻击行为因而对攻击有更有效的响应。

蜜罐是近几年才发展起来的一种主动安全技术。它设置专门让黑客攻击的应用系统以记录黑客的活动，让黑客来告诉我们所面临的威胁。分析蜜罐采集的信息，人们可以了解黑客攻击的方式和手段，发现威胁所在。蜜罐提供了一个丰富的认识黑客攻击手段的信息源。

蜜罐是网络安全的一个全新领域。它通过构造一个有着明显安全漏洞的系统来引诱入侵者对其进行攻击，并在攻击的过程中对入侵者的入侵动机、入侵手段、使用工具等信息进行详细地一记录。根据收集到的入侵者信息，我们就可以分析得到入侵者所使用的最新技术、发现系统中的安全漏洞，从而对系统中存在的问题及时予以解决。

在学位论文工作期间，我对蜜罐进行了系统的理论研究，在此基础上，初步设计了一个包含蜜罐系统的网络安全架构。将我的蜜罐系统放到校园网环境中，采集到许多宝贵信息，在一定程度上提高校园网的安全性。通过对这些信息的分析，我对蜜罐有了更深入的理解，特别在它对信息分析能力、对黑客的了解方面也有了较大提高。

关键词：蜜罐；蜜网；陷阱系统

Abstract

With the rapid development of Internet, more and more applications are realized through the network, at the same time the security of the network also faces the enormous challenge too. The fast network has facilitated intruders. The intruder's engineering skill is improving constantly too. How to protect systems and the believed networks from intrusion is an urgent problem to be solved at present. The traditional safe strategy now manifest their limitations, so a new system named Intrusion Prevention System has been come up. As a new active technology Intrusion Prevention System has two distinct advantages: first it can actively block the attacks before hacker started. On the other hand, intrusion prevention system can give a more effective response by tracing hacker and collecting detailed attacking information.

Honeypot is a new technology developed in recent years. Honeypot can simulate real services or application so as to induce the hacker to attack it, thus it can collect all of the activities done by hack. We can learn the new methods and technologies by analyzing the information, thus we can discover what, where and how the threatens are existed. Honeypot offers abundant valuable information.

Honeypot has a lot of drawbacks in security, so hacker will attack it, during this process honeypot will record all motives, methods and tools used by bad intruders. We can discover bugs and improve the network security performances to some extent by analyzing the information.

During my work, I have generally studied the honeypot, I design a network security architecture by using honeypots. The campus network performance will be improved by deploying honeypots in different positions, the honeypots can collect a lot of valuable information. The more I comprehended about honeypot, especially in the capability of analyzing information, the more I learn about hacker' attack meantime

Key Words: Honeypot; Honeynet; Deception System

第一章 绪 论

1.1 引言

随着因特网技术的发展，基于网络的应用信息系统也越来越多，所涉及到的应用范围已经涵盖了电子商务、电子政务、电子税务、电子银行、电子海关、电子证券、网络书店、网上拍卖、网络购物、网络防伪、网上交易和网上选举等诸多方面。网络信息系统在政治、军事、金融、商业、交通、电信、文教等方面的作用日益扩大，社会对网络信息系统的依赖也日益增强，网络与人们的日常生活也变的密不可分。伴随着网络如此广泛的应用，借助于网络来对各个方面所进行的攻击也日益严重，网络安全与保密问题也已经成为人们要面对的首要的问题。从大的方面来说，网络安全问题关系到一个国家的安全和主权、社会的稳定、民族的文化等方面。从小的方面来说，信息安全问题也是人们能否保护自己隐私的关键。近十几年以来，网络上的各种安全性问题越来越多，也越来越严重。

现有的网络安全措施主要以防火墙和入侵检测系统为核心，它们在一定程度上改善网络的安全保障。但防火墙和入侵检测系统并不是万能的，它们在很多方面存在弱点。防火墙防范的前提是对各种已识别类型的攻击进行正确的配置。这就要求防火墙知识库不断更新以识别各种新类型的攻击。入侵检测系统一方面像防火墙一样需要知识库不断更新，另一方面对有违反安全策略的恶意使用行为进行识别和响应。从根本上说，防火墙和入侵检测系统滞后于各种各样的黑客攻击。这就决定了以防火墙和入侵检测系统为核心的网络安全体系不可能完全、有效地解决网络安全问题。这些安全技术中，大多数技术都是在攻击者对网络进行攻击时对系统进行被动的防护，而蜜罐技术的引入，可以将防护从被动方式变为主动方式。即在蜜罐中用特有的特征吸引攻击者，同时对各种攻击行为进行分析并找到有效的对付方法。

1.2 传统的网络安全防护手段

随着网络规模的不断扩大，人们对网络知识的了解越来越深入，网络上的攻击行为变得越来越多，已经严重威胁到网络与信息的安全。计算机网络信息安全已经成为一个倍受关注的问题。

网络与信息安全技术的核心问题是对计算机系统和网络进行有效地防护。网络安全防护涉及面很广，从技术层面上讲主要包括防火墙技术、入侵检测技术、病毒防护技术、数据加密技术和认证技术等，这些安全技术中，为了防止各种入侵手段，提高系统的安全程度，人们采取了多种入侵防护手段，目前经常使用到的有以下几种：

1.2.1 防火墙

防火墙是一种用来加强网络之间访问控制、防止外部网络用户以非法手段通过外部网络进入内部网络，访问内部网络资源，保护内部网络操作环境的特殊网络互联设备。它对两个或多个网络之间传输的数据包和链接方式按照一定的安全策略进行检查，来决定网络之间的通信是否被允许，并监视网络运行状态。防火墙实际上是一个独立的进程或一组紧密联系的进程，运行于路由、网关或服务器上来控制经过防火墙的网络应用服务的通信流量。安全、管理、速度是防火墙的三大要素。防火墙的目的在于实现安全访问控制，按照 OSI/RM，防火墙可以这七层中五层设置。虽然防火墙可以提高来自外部网络非法用户对内部网络攻击的安全性，但随着网络攻击技术及工具的发展，防火墙在网络安全防护中的弱点逐渐暴露出来，最明显的防火墙无法防护来自内部网络用户的攻击，无法防备病毒攻击，无法保护那些绕过防火墙的攻击[1]。

1.2.2 入侵检测

入侵检测是最近 20 多年发展起来的一种动态监控、预防或抵御系统入侵行为的安全机制。入侵检测系统与防火墙安全策略相比，是一种不同的安全策略，主要通过监控网络、系统的状态、行为以及系统的使用情况，来检测系统用户的

越权使用及系统外部的入侵者利用系统的安全缺陷对系统进行入侵的企图。入侵检测系统在识别入侵和攻击时具有一定的智能，主要体现在入侵特征的提取和汇总、响应的合并与融合，在检测到入侵后能在一定程度上采取相应的响应措施。入侵检测是一种事后处理方案，具有智能监控、实时探测、动态响应、易于配置等特点[2]。入侵技术的引入使得网络、系统的安全性得到进一步的提高。

现在的入侵检测系统通常分为基于主机和基于网络两类。基于主机的入侵检测系统的主要特征是使用主机传感器监控系统的信息，主要用于保护某台主机的资源不被破坏。基于网络的入侵检测系统主要是网络监控传感器监控包监听器收集的信息，用于保护整个网络不被破坏。它不能审查加密数据流的内容，对高速网络不是特别有效。但是入侵检测会出现漏报和错报

1.2.3 加密传输

加密就是为了安全的目的对信息进行编码和解码。数据加密的基本过程就是将可读信息(明文)译成密文(或密码)的代码形式。加密的逆过程即为解密。加密传输技术是一种十分有效的网络安全技术，它能够防止重要的信息在网络上被拦截和窃取[3]。

IPSec （IP 安全体系结构)技术在 IP 层实现加密和认证，实现了数据传输过程中的完整性和机密性，可为 IP 及其上层协议(TCP 和 UDP 等)提供安全保护。

虚拟专用网(VPN)技术能够在公共网络中为两台通信的计算机建立一个逻辑上的安全通道(tunnel)，通过数据的加密和认证使得数据包即使被截获也不容易破译，提供了很好的安全性。

现有的这些入侵防御手段中，都有一个共同的特点就是采用拒绝型防御策略，即根据特定的需要指定一系列的访问策略，不符合指定的安全策略就拒绝访问。比如进入防火墙的数据不符合防火墙的规则，则不让通过；没有密钥就无法通过正常渠道得到解密的数据等等。这些防护手段相当于在需要保护的系统外部建立了一道保护屏障，一旦所使用的防御手段有效，就把黑客成功地阻止在被保护系统之外，从而使系统的安全性得到了充分的保证。

1.3 本论文所做的主要工作

传统的网络安全技术如防火墙、入侵监测、加密传输等在信息安全领域发挥了很大的作用，这些被动的防御措施一般都是基于规则或者特征匹配的方式工作，并且都是针对现有攻击技术的。随着攻击技术的不断发展，新的攻击方法层出不穷，攻击的发起时间、攻击者、攻击发起地点和攻击目标都具有很大的不确定性。被动防御技术对新的攻击方法往往不能正确识别，从而陷入被动的地位。

主动防御技术逐渐开始受到人们的关注。主动防御技术试图牵制和转移网络攻击行为，并且对攻击方法进行技术分析，有可能获得未知的攻击技术资料，对网络攻击进行取证并且对攻击者进行监视和跟踪。蜜罐(Honeypot)技术是一种主动防御的安全技术，在网络防火墙、入侵检测系统等安全措施的配合下,能够弥补原有安全防御的不足,提升网络的安全性能。

鉴于传统防御手段中所存在着不能够对入侵者的入侵行为进行详细跟踪、记录并分析这一不足之处，我们需要有一种既能够保证重要信息不被窃取、系统不被破坏，又能够对黑客的入侵行为做进一步监视、记录的手段。我此次论文的目的，也就是设计一个具有这种特性的网络安全系统。

蜜罐系统是本次论文工作的重点。它是应用蜜罐技术实现的一种设置好的网络或主机，通过模拟一些常见的系统漏洞，制造一个容易被入侵的网络环境，诱导入侵者对系统发生攻击。在攻击的过程中对此环境中的进出数据进行捕获与控制，并对被捕获数据进行分析，从而理解和研究入侵者们所使用的工具、入侵的方法以及入侵的动机，由此获取此次入侵的第一手入侵数据。这样做，一方面可以转移攻击目标，让攻击者在该系统中浪费时间，这在一定程度上保护目标机；另一方面可以通过对收集的入侵数据分析来对系统进行评估、优化入侵检测系统、防火墙系统，为指定强有力的安全决策提供依据。通过学习他们使用的工具和思路，我们可以更好的来建立我们的安全系统模型。

蜜罐的思想是建立一个陷阱系统，这个系统有着一个真实的或者建立在别的系统上的操作系统，它看上去有许多漏洞，可以很轻易的获取其资源。蜜罐应该以一种与真实系统相似的方式建立，应当有许多假的文件、文件夹和其它信息，以便使得这个系统与真实系统看起来非常相似。通过使用合法的文件使蜜罐看起

来象一个合法的主机，这样就会让黑客相信他们在获取一些重要的信息。黑客在“honeypot”中待的时间越长，他们所使用的技术就会暴露的越多，而这些信息、可以被用来评估他们的技术水平，了解他们使用的攻击工具。理想的蜜罐提供一个入侵者可以被捕获的环境，或者提供一些可以被入侵的弱点，这些弱点都是以真实的系统为背景建立的。建立诱捕系统的目的不是为了抓住入侵者，而是要监视和学习它们的行为，找出它们是如何探测和入侵系统的，以及如何才能在真实的系统中阻止类似入侵行为的发生。

1.4 本论文的内容安排

第一章 分析了网络安全的现状，简单介绍传统安全防御措施的各自的优缺点，提出采用蜜罐技术同时结合防火墙和入侵检测技术来构筑新的网络安全防护系统；

第二章 介绍蜜罐概念及发展历史，并对现有几种蜜罐系统性能进行分析与对照；

第三章 提出此次设计的总体框架，对其中各部件进行介绍；

第四章 分析蜜罐特性并由此给出此系统的概要设计；

第五章 给出系统各部分组件的简单实现；

第六章 对该系统的测试及分析；

第二章　蜜罐系统概述

2.1 蜜罐的概念

2.1.1 蜜罐技术的发展背景

当今世界计算机网络的广泛应用也深入到人们生活的各个领域，对人们的生活方式和工作方式产生着前所未有的影响，它已经成为人们生活中不可缺少的部分，与此同时网络规模的不断扩大，网络上的攻击行为变得越来越多，如何应对这些攻击已经成为目前人们最关注的问题。

网络安全防护涉及面很广，从技术层面上讲主要包括防火墙技术、入侵检测技术、病毒防护技术、数据加密技术和认证技术等，这些安全技术中大部分都是在攻击者对网络进行攻击时对系统进行被动响应。这些传统的防御手段在一定时期内曾占有非常重要的地位，在网络安全防护中曾起着不可替代的作用，但它们都有一个同样的缺陷：发现和预防能力不够。因为这些防御手段都是通过把入侵者阻挡在被保护系统之外来完成系统防护的，也就是说，入侵者在刚刚侵入到系统或者还没有侵入系统的时候就已经被发现并被禁止做进一步的入侵行为，所以也就不可能知道入侵者的确切的入侵目的，以及入侵者想要进行的进一步入侵行为。而在很多情况下，我们想要收集入侵者的详细入侵踪迹和入侵手段，以作为对系统安全性分析的重要数据，或者在以后法律程序上作为入侵行为的重要证据。而使用传统的防御措施就不能够达这个目的，因此我们需要有一种新的技术来弥补现有防御手段的不足。而蜜罐技术正是在对这种新技术的要求下提出的。

2.1.2 蜜罐的概念

关于蜜罐的定义一直以来都没有一个准确的定义，不同人对蜜罐有不同的定义。在本文中我采用美国 Project Honeynet 研究组的成员之一 Lance Spitzner 给出的关于蜜罐的定义：

蜜罐是一个信息系统资源，它的价值在于未授权或非法的使用该资源[4]。这是一个总的定义，它包含了所有不同的蜜罐表现形式。以下对蜜罐技术的讨论将根据上述 Lance Spitzner 对蜜罐的定义，蜜罐的价值在于攻击者与它们的交互。概念上讲，几乎所有的蜜罐工作都类似。它们是一种没有被授权的活动的资源，它们没有任何产品价值；理论上讲，一个蜜罐应该看不到流量，因为它没有合法的活动。这就意味着任何与蜜罐的交互都有可能是未授权或恶意的行为。任何试图对蜜罐的连接极有可能是一次探测或攻击。这种概念听起来很简单，也正是这种简洁性使蜜罐有许多优点和缺点。

蜜罐是一种资源，它的价值就在于其可以被攻击、被入侵。这也就是说蜜罐具有被探测、被攻击甚至被利用的可能。因为蜜罐不会修补任何东西，这样就给使用者提供了额外的、有价值的信息。蜜罐不会直接提高计算机网络安全，但它却是其他安全策略不可替代的一种主动防御技术。

2.1.3 蜜罐的优点和缺点

蜜罐并不是针对特定问题的解决，而是一种面向整体安全架构的工具。蜜罐是一种伪装成为真实目标的资源。它有着被攻击、被入侵的可能性。它的主要目的是为了分散攻击者的注意力，并且获得攻击者本身以及其攻击行为的相关信息。蜜罐本身也具有一定优势和劣势，这会影响它的价值。

优点：蜜罐是一个非常简单的定义，这样使得它有强大的实力。高价值的小数据集：蜜罐收集信息量少，它们一天只记录 1MB 的信息并且只发出 10 次警告。蜜罐只捕获恶意攻击，任何与蜜罐的交互极有可能是未授权的或恶意的行为。蜜罐通过收集少量数据信息来减少噪音，然而这些信息有较高的价值，因为它是攻击者的信息。这就意味着更容易分析蜜罐收集的数据，并从中得到有价值的信息；新的工具和策略：蜜罐是设计用来捕获任何对它们的攻击的任何行为，包括从未见过的新的工具策略；最小的资源：蜜罐要求最小的资源，它们只捕获恶意攻击，这意味着一台旧的计算机能容易的处理完全的 B 类网络；加密或 IPV6：不像大多数安全技术，蜜罐对加密和 IPV6 环境也有很有效。不管黑客对蜜罐进行任何攻击蜜罐都将能检测并捕获到这些攻击；信息：蜜罐能够收集进一步的与攻击者的相关信息；简单性：最后，蜜罐的概念很简单。没

有特殊的运算规则来发现，维护稳定的表格，或者更新的签名。技术越简单，错误率和错误配置的可能性就越小。

缺点：像任何其他技术，蜜罐也有缺点，它不能代替当前的技术，而是与现存的技术配合工作。局限的视野：蜜罐只可以跟踪和捕获与它们直接交互的活动，蜜罐将不能捕获对其它系统的攻击，除非攻击者或威胁与蜜罐有交互；指纹识别：蜜罐具备一些特定的预期特征或者行为，因而能够被攻击者识别出其真实身份并对它进行攻击；威胁：所有的安全技术都有危险，防火墙有被渗透的危险，加密有被破解的危险，IDS 传感器有检测不出攻击的危险。蜜罐也如此，它也有危险。具体地讲，蜜罐有被黑客利用并危及其他系统的危险，不同的蜜罐有不同的风险，而且风险也是变化的，这取决于它的构建和部署方式。

现有的各种安全防御机制都有自己的局限性，因此针对网络的安全不能只依靠单一的安全防御技术和防御机制。只有通过在对网络安全防御体系和各种网络安全技术和工具的研究基础上，制定具体的系统安全策略，通过设立多道的安全防线、集成各种可靠的安全机制，建立完善的多层安全防御体系，才能抵御来自系统内、外的入侵攻击，达到维护网络系统的安全。

2.2 蜜罐系统的分类

根据不同标准可以对蜜罐技术进行不同分类，下面讨论三种分类方式：

2.2.1 根据设计目标分类

根据产品设计的目的可以分为产品型蜜罐和研究型蜜罐[5]。产品型蜜罐的目的是减轻受保护组织将受到的攻击威胁，它所做的工作就是检测并对付恶意攻击者。它所代表的是这样一种系统：它有助于减轻组织或环境中所存在的风险，可以为系统及网络的安全保障提供特定的价值。在阻止安全方面，它采取欺骗和威慑技术。欺骗就是让攻击者在进攻蜜罐上浪费时间和资源；而威慑是告诉攻击者组织中已经部署了一些蜜罐，以此来吓跑它攻击者。在检测方面，由于蜜罐没有任何产品流量，任何发往蜜罐的连接请求都被认为是可疑的活动，都会被蜜罐检测到，所以对蜜罐而言不存在误报和错报。在对攻击事故作出响应时，组织所面

临的主要难题是证据的收集，当攻击者进入系统时，他们的活动会留下证据，可能用来判定攻击者是如何进入系统的，进一步看到在它获得了对系统的控制权之后又从事了什么活动。即使攻击者采取了一些隐藏活动的措施，如修改日志文件，这些活动还是可以被追踪到的。而且蜜罐能够帮助解决提供反抗能力面临的问题。蜜罐中没有任何产品活动，这是前提，借助于这些前提来协助解决数据污染的问题。当某个蜜罐被攻破时，系统中惟一的实际的活动就是攻击者的活动，这有助于维护其完整性。一般情况下，商业组织运用产品型蜜罐对自己的网络进行防护。

研究型蜜罐专门以研究和获取攻击信息为目的而设计，只有那些需要进行研究的组织，例如大学、政府、军队或安全研究组织才需要使用研究型蜜罐。它通过为大家提供一个可用于了解计算机威胁的平台，在信息收集方面提供了广泛的价值。这种类型的蜜罐并不能减少组织的风险，但可以应用所了解的信息，来改进预防、检测和反应。

2.2.2 根据工作方式分类

根据蜜罐的工作方式不同可以分为牺牲型蜜罐、外观型蜜罐和测量型蜜罐[6]。牺牲型蜜罐就是一台简单的为某种特定攻击设计的计算机。牺牲型蜜罐实际上是放置在易受攻击的地点，假扮为攻击的受害者，它为攻击者提供了极好的攻击目标。它所使用的数据收集信息者是蜜罐附近配置的网络嗅探器，但它不会提供任何关于主机配置的数据。需要手动或运用各种第三方跟踪分析工具进行额外的检验。还须考虑用防火墙或其他网络控制设备来隔离并控制牺牲型蜜罐。牺牲型蜜罐提供真实的攻击目标，所以得到的结果都是真实系统上会发生的状况。牺牲型蜜罐可以对被入侵前的系统进行分析但是系统一旦被攻陷就不可能再正常工作，所以需要管理员要定期检验蜜罐系统，判断整个系统是否已被入侵，在被入侵的情况下还需要判断蜜罐所遭受的攻击类型。外观型蜜罐是一个呈现目标主机的虚假映像系统，仅仅是对网络服务进行仿真而不会导致机器真正被攻击，从而蜜罐安全不会受到威胁。它是最简单的蜜罐，通常由某些应用服务的仿真程序构成，以欺骗攻击者。它们只能够提供潜在威胁的基本信息。测量型蜜罐结合了对外观型蜜罐的低成本和牺牲型蜜罐的细节深度两方面的优点，通过对现有系统进行大规模的操作系统层次或内核层次更改以及应用程序开发，它已作为一种有效的网

络防御方法，包括进行高级数据收集、攻击活动击落、基于策略的告警和企业的管理功能。

2.2.3 根据交互性分类

交互性是蜜罐的一个重要属性。交互性体现了入侵者和实现这个蜜罐的操作系统之间的交互程度。交互级别为我们提供了一种可以对蜜罐进行测度和比较的标尺。蜜罐所能做的事情越多，以及攻击者对蜜罐所能做的事情越多，从蜜罐上所获得的信息就会越多，但同时攻击者对蜜罐所做的事情越多，他所能造成的危害就会越大。交互性定义了蜜罐允许攻击者的行为水平。按照交互性把蜜罐分为三类，低交互度的蜜罐、中交互度的蜜罐和高交互度蜜罐[6]。这个分类有助于我们理解所使用的蜜罐的类型，强度和弱点。

1 对低交互度蜜罐，它设计简单且功能有限，安装配置部署和维护都很容易。它仅仅是模拟了大量的服务，攻击者和它的交互也就限于这些预指定的服务。这种蜜罐没有真实的操作系统，它的价值主要在于检测，具体来说就是对未授权扫描或者未授权连接尝试的检测。它只提供了有限的功能，因此大部分可以用一个程序来模拟。只需要将该程序安装在一台主机系统中，配置管理希望提供的服务就可以了。管理员所要做的工作就是维护程序的补丁并监视所有的预警机制。这种蜜罐所提供的功能非常少，因而出错的地方也很少，所具有的风险也很低。同时它能够为我们提供的关于攻击者的信息量也有限。通过模拟服务和操作系统实现。攻击行为受到蜜罐模拟水平的限制。例如，一个模拟的 FTP 服务在 21 端口监听，也许只能模拟 FTP 登录或者可能支持一些额外的 FTP 命令。低交互度蜜罐的缺点主要是它们只记录有限的信息，只能捕获已知的攻击行为，并且以一种预定的方式进行响应。模拟服务仅能做到这样。而且它容易被攻击者识破，不管模拟服务做得多好，有技术的黑客最终都能检测到它的存在。

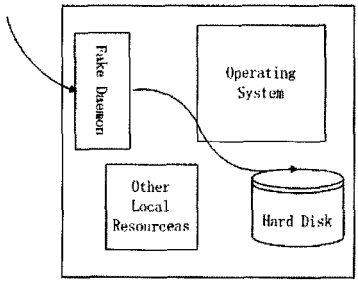

图 2.1 低交互度蜜罐

Figure2.1 Low Interacting Honeypot

2 中交互度蜜罐为攻击者提供的交互能力比低交互度的蜜罐多些，它们能够预期一些活动，并且旨在可以给出一些低交互度蜜罐所无法给予的响应。它通常要花费更多的时间去安装和配置。与低交互度蜜罐相比它的部署和维护是一具更为复杂的过程，攻击者得到更多的交互，因此必须要以一个安全的方式来部署这种交互。必须开发相应的机制以确保攻击者不会危害其他系统，并且这种增加的功能不会成为攻击者进行攻击的易受攻击环节。攻击者可能会访问到实际操作系统，但他们的能力是受限的，这种类型蜜罐必须要进行日常维护，以应对新的攻击。由于它具有较大的复杂度，所以出错的风险也相应增大，另一方面它可以收集到更多攻击者的信息。

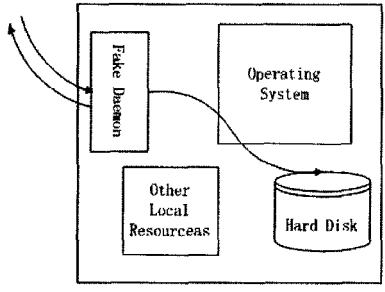

图 2.2 中交互度蜜罐

Figure2.2 Secondary Interacting Honeypot

3 高交互度蜜罐是目前蜜罐技术的极限。它能提供大量的关于攻击者的信息但是构建和维护它们是极为耗时的，并且与之相伴的是最高级别的风险。它常常有复杂的解决方法。因为它为攻击者提供真正的操作系统和应用程序与之

交互，一切都不是模拟的，它给攻击者提供真实的环境。但是这些系统定义为没有任何产品价值。一旦攻击者掌握了对某个蜜罐的控制权，就拥有了一个完整的可操作系统进行交互。最为常见的高交互度的蜜罐往往被放置在一种受控环境中，如防火墙之后。借助于这些访问控制设备来控制攻击者的攻击能力，防火墙允许攻击者攻破位于其后的某个蜜罐，但不允许攻击者使用该蜜罐启动对外的攻击。这种构架的部署和维护十分复杂，而且这种类型蜜罐还要求对防火墙有一个恰当的过滤规则库。同时还要求配合 IDS 的功能，要求 IDS 的签名数据库进行更新，而且要不停监视蜜罐的活动。它为攻击提供的交互越多出错的地方就越多。一旦进行了正确的实现，高交互度的蜜罐就能够最大程度的洞察攻击者。例如想在一个 Linux 蜜罐上运行一个 FTP 服务器，那么你建立一个真正的 Linux 系统来运行 FTP 服务。这种解决方案优势是双重的，首先，你能够捕获大量信息。这可以通过给攻击者提供真实的系统与之交互来实现，你能够了解到攻击者的整个攻击行为，从新的工具包到 IRC 的会话的整个过程。高交互蜜罐的第二个优势是对攻击者如何攻击不是假设，它们提供一个能够捕获行为的开放的环境，高交互度解决方法将使得我们能学到意外的攻击行为。一个极好的例子是一个蜜罐在一个非标准的 IP 协议上捕获密码后门命令。然而，这也增加了蜜罐的风险，因为攻击者能够用这些真实的操作系统来攻击非蜜罐系统。结果，要采用附加技术来阻止对非蜜罐系统的攻击。一般来讲，高交互度的蜜罐在安全防护性上优于前两种交互度蜜罐，然而，高交互度蜜罐的开发和维护都较复杂。

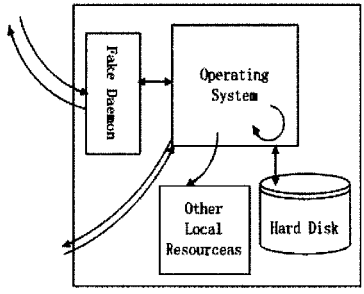

图 2.3 高交互度蜜罐

Figure2.3 High Interacting Honeypot

表 2-1 三种不同交互度的蜜罐对比

table2-1 Diference between Interacting of the Three Honeypot

	低交互性蜜罐	中交互性蜜罐	高交互性蜜罐
交互性	低	中	高
真实的操作系统	无	无	有
所受威胁的程度	低	中	高
信息收集	少	中	多
是否希望被入侵	否	否	是
运行难度	低	低	高
开发难度	低	高	较高
维护所需时间	少	少	多

2.3 现有蜜罐系统

2.3.1 常见的几种蜜罐系统

蜜罐是一个可以模拟具有一个或多个攻击弱点的主机系统，为攻击者提供一个易于被攻击的目标。蜜罐中所有的假终端和子网都经过精心设计，以吸引攻击者的攻击。蜜罐监视攻击者的行径，收集相关的数据。

市场上的现有的蜜罐产品虽然总的来说功能繁多，类别覆盖从商业到研究，从低交互度蜜罐到高交互度蜜罐，从单个蜜罐系统到复杂的蜜网，这些现有的蜜罐产品使用起来确非常容易，各个开发商在产品的易用性方面做了不少的工作。常见的有以下一些常用产品[7]：

. BackOfficer Friendly

. Specter

. Honeyd

. 自制 Honeypot

. ManTrap

. Honeynet

BackOfficer Friendly 是一种免费的低交互度的 honeypot。它是一种主要用于检测攻击的产品型 honeypot，其优点在于它易于安装、配置和维护；可运行在任何基于 Windows 或者 Unix 的平台上，包括大部分桌面系统或笔记本系统；由于其简捷性故带来的风险小。其缺点是仅能对 7 个端口的服务进行模拟，而无法对指定端口进行定制，因而增加了指纹识别的可能性，没有对任何远程的日志、预警或配置远程功能；因此不适合于企业级应用。

Specter 是一种商用的、低交互度的产品型 honeypot，其主要价值在于检测。它在阻止方面也具有价值即欺骗或威慑攻击者。其优点在于它不仅易于安装、配置和部署，而且它可以模拟众多的服务，可以监视两倍于 BOF 的端口，有出色的通知能力和远程管理。

Honeyd 是一种开放源码的低交互度 honeypot，它引入了几个新特征。首先，它具备对数百万个系统进行监视并同时充当数千个受害者身份的功能。Honeyd 可以确定出哪些系统是有效的，哪些系统是不存在的，然后在执行中充当这些不存在的系统的身份。它还具备在应用层和 IP 栈层进行操作系统模拟功能。它的优点是可以对所有的 TCP 或者 UDP 端口以及整个网络进行监视，作为一种开放的源码的解决方案，它是免费的并且会随着安全界中其他人的输入和开发而迅速发展起来，通过在 IP 栈级和应用级模拟操作系统，可以阻止指纹识别。作为一种低交互度的解决方案，它无法为攻击者提供实时的操作系统进行交互。作为一种开发的源码解决方案，没有为维护和故障诊断提供任何的正式支持，没有任何内置用于预警的机制，也没有任何用于捕获大量信息的机制。

现有的自制的 honeypot 主要集中于两种类型。第一种是端口监视，这是一种低交互度的 *honeypot* 主要用于捕获恶意攻击和载荷。第二种类型自制 honeypot 是 jail，这是一种较高等交互度的解决方案。这项功能主要限制在使用 chroot(1) 的 Unix 系统。Jail 并不是作为一种 honeypot 解决方案而设计的，而是一种用于削减风险的安全机制。Jail 具有被检测 和突破的可能性。因此风险较大，只有高级的 Unix 安全专家才能使用它。

Mantrap 是一种商用的高交互度的 *honeypot*。它的优点是可以使用内置的的嗅探器检测任意端口上的活动，为攻击者提供了一个完全的操作系统进行交互，经由内核空间捕获所有的攻击者活动，包括诸如 SSH 之类的加密流量，有出色的日志记录功能，远程功能，包括 email 预警以及远程管理，使得其成为一种企业

级解决方案。由于它是一种高交互度的 honeypot 意味着攻击者具有利用系统危害其他系统或组织的可能，攻击者可能会对其访问的 Mantrap 牢笼进行指纹识别或者突破，且它受限于 Solaris 操作系统，使用完整的开发员安装。

Honeynet 是一种高交互度的 honeypot。是现有交互级别最高的一种解决方案。它创建了一种高可控环境的架构。它最大的优点是它的灵活性——任何系统或应用都可置于 Honeynet 中，而且它对已知或未知和工具与战术都有广泛的数据捕获能力，适合于众多的组织和环境。但同时 Honeynet 的部署及维护所需的资源比前几类蜜罐都复杂，高交互性的功能引入了攻击者使用系统攻击、损害或摧毁其他系统或组织的风险，由于 honeynet 是一项新发展起来的技术，其中有些不成熟的技术会带来引入错误的风险。

2.3.2 蜜罐与蜜网区别

Honeypot 是一个故意设计为有缺陷的系统或“伪”服务，通常是用来对入侵者的行为进行报警或者诱骗。一般情况下，传统的 Honeypot 模拟其它操作系统或者一些常见漏洞，而 Honeynet 则有所不同，它是一个用来对黑客的入侵行为进行研究、学习的工具。

Honeynet 是一个网络系统，而并非某台单一主机，这个网络系统是隐藏在防火墙后面的，所有进出的数据都受到监视、捕获和控制。这些被捕获的数据可以用来研究和分析入侵者们使用的工具、方法以及入侵动机。在 Honeynet 中，研究者可以使用各种不同的操作系统以及设备，如 Solaris，Linux，WindowsNT，Cisco Switch 等等。这样建立的网络环境看上去会更加真实可信，同时还在不同的系统平台上面运行着不同的服务，比如 Linux 的 DNS server, WindowsNT 的 webserver 或者一个 Solaris 的 FTP server。研究者可以学习使用不同的工具以及不同的策略。由于某些入侵者的原始目标可能仅仅定位于几个特定的系统漏洞，配置这种多样化的网络系统，就有可能揭示他们更多的一些特性[8]。

在 Honeynet 中的所有系统都是标准的主机，运行的都是真实完整的操作系统以及应用程序,就像能在网络上找到的系统一样，没有刻意地模拟某种不安全环境或者故意地使系统存在明显的漏洞。在 Honeynet 的系统上存在的安全风险，与网络上一些企业内部的网络存在的安全风险完全相同。

因为一个典型蜜罐系统是通过模仿一个操作系统来实现的，因此有经验的攻击者知道如何识别哪些是正确的迹象，而哪些不是。对于 Honeynet 来说，因为它使用的是标准的操作系统，所以如果设计得当，入侵者将很难发现 Honeynet 是一种陷阱，这样就可以使得对入侵者的入侵信息的收集工作更加顺利地进行。

一方面，从蜜罐到蜜网，系统的复杂程度越来越高，因此所需要的技术含量也就越来越大；另一方面，如果我们已经拥有一个比较成熟的蜜罐，那么想要以蜜罐为基础构造一个蜜网系统也相对来说比较容易。因此，在此次的设计过程中，我把蜜罐做为了设计的重点。

2.4 蜜罐操作系统(OS)的选择

2.4.1 纯蜜罐和虚拟蜜罐

在讨论利用哪种操作系统来实现蜜罐之前，我先大概介绍下关于纯蜜罐和虚拟的蜜罐的含义。

简单的来讲，纯蜜罐就是指安装了一种操作系统，同时对其行为进行监控的一台主机。而虚拟蜜罐则指的是在一个已经存在的操作系统(主操作系统)上所安装的另外一个寄生的操作系统或者叫子操作系统[9]。

纯蜜罐实现起来比较容易，理论上来说，它可以用任何的一种操作系统来实现。但是以这种方式实现的蜜罐存在着一些缺点。比如在同一时刻，同一台物理主机上只能运行一个操作系统，资源没有被充分利用；另外，一旦系统被入侵，那么重新恢复这个被入侵的系统将会是一件比较麻烦的事情；同时，对这种系统的监控一般需要有外部设备的参与，因为对实现蜜罐所用的操作系统的任何修改都有可能有指纹识别，有可能被经验丰富的入侵者觉察到。

而使用虚拟技术实现的虚拟蜜罐有比较大的灵活性。根据实现蜜罐所使用的主操作系统和虚拟技术的不同，子操作系统可以拥有和主操作系统完全不同的特性。子操作系统可以被并行的安装在一台物理主机上，甚至几个完全不同的子操作系统都可以并存。虚拟机这种实现方式使得管理员可以在主操作系统环境下来监视子操作系统的运行，而不用担心会被入侵者发现。此外，一旦子系统遭到破坏，管理员也可以轻松的从主操作系统中将其恢复。

当然，虚拟蜜罐上述优点的前提是要求主操作系统本身安全性是较高的，而且主操作系统是不应当让入侵者发现的。这一点可以通过将主操作系统同一个单独的网络接口连接来解决，只有蜜罐的管理员才有使用这个网络接口权限。

2.4.2 Windows 操作系统

微软的操作系统家族中主要包含有两类操作系统：基于 MS-DOS 的操作系统：如 Windows 3.1, 95, 98 等；基于 NT 内核的操作系统：如 Windows NT，2000, XP。一般来讲，基于 NT 内核的操作系统被商业领域来做专门的服务器，而非 NT 内核的则被家庭用户广泛使用。然而，随着 Windows 2000 和 WindowsXP 的出现，家庭用户也逐渐开始使用基于 NT 内核的操作系统。

在过去十几年中，随着互联网的迅速发展，越来越多的主机开始使用 Windows 操作系统。同时，也有越来越多的报告不断揭露 Windows 主机受到了诸如病毒、蠕虫、木马之类的恶意攻击。正是由于 Windows 操作系统存在着大量的安全漏洞，所以许多攻击者都把它选做自己的攻击对象。虽然微软发布了许多很多的补丁来弥补这些安全漏洞，但是在人们的心目中，它还被认为是一种易受攻击的操作系统。

既然 Windows 系统有如此多的安全漏洞，那么用它来实现蜜罐对入侵者的数据进行收集应该是一个不错的主意。而事实却不是我们所想象的那样，Windows 操作系统的结构使得用它来进行数据收集相当困难。与 UNIX 操作系统不同，在 Windows 中入侵者和系统之间没有一个交互的过程，因此也就不可能记录下任何的交互信息。大多数入侵者都使用一些预先编译好的二进制文件，通过系统的漏洞而入侵主机。之后，通过安装后门的方法，得到对系统的最高控制权。这个过程就使得蜜罐管理员很难发现究竟在蜜罐上发生了什么事情。

此外，由于 Windows 操作系统的源代码不是公开的，这样就使得要对操作系统本身来做修改就变得几乎不可能。系统事件的记录必须被保存到用户空间，而这些记录又对包括入侵者在内的所有用户都是可见的。因此蜜罐管理员能做的就是诸如记录当前运行的程序列表、定期的用 MD-5 校验和来检查系统文件的完整性、以及通过 NT 事件记录来远程监控应用程序和内核事件等一些有限的工作了[10]。

2.4.3 Unix操作系统

UNIX 最早是在 1969 年 AT&T 的 Bell 实验室中诞生的，作者是 Ken Thompson。后来又经过 Ken Thompson 及 Demmis Rithchie 用 C 进行改良，大幅增加可移植性后，开始了它的蓬勃发展。目前一些基于 UNIX 的操作系统，如：Linux, BSD 等，也已广泛地被广大用户所接受。虽然在这些操作系统中，也有一些安全漏洞被发现，但这大多是由于一些运行在用户模式下的后台程序编写不当所造成的，因此这些操作系统的核心仍然被认为是非常稳定的。

因为大多数以 UNIX 为内核的操作系统都是开放源代码的，因此用以 UNIX 为内核的操作系统来实现蜜罐非常方便。我们可以通过修改系统的源代码来实现我们自己的事件记录机制，然后将源代码重新编译，植入到我们所要实现的蜜罐中去。要发现这种重新编译过的可执行代码是比较困难的，因此入侵者也就不会怀疑他所处的环境正在被我们所监视。

大多数的入侵者在侵入一台主机之后，会在这台主机上安装 root-kits。这些 root-kits 中包含了事先编译过的系统代码，这些代码然后就会被拷贝到系统中以覆盖掉原先的系统代码。这样，我们原先的信息收集方式可能就会被破坏，从而使得可以收集到的数据量大大减少，甚是根本收集不到任何数据。为了解决这个问题，我们可以将信息采集的部分植入系统内核中，这样，对入侵者来说就很难觉察到。同时，对于以普通用户入侵的入侵者来说，他是不能够对系统内核进行编译的，这样也就保证了我们的信息采集手段不被破坏[10]。

第三章　系统总体结构

3.1 单机蜜罐系统结构

根据蜜罐技术的特点和应用要求，综合考虑多方面的因素，设计了一种基于主机的蜜罐系统。系统被放置在防火墙内部通过防火墙与外部网络连接，可以充分发挥防火墙的作用，设置内部网络主机和蜜罐系统的对外连接属性。蜜罐系统以系统程序的形式安装并运行在主机上。蜜罐内部主要由网络服务、数据收集和日志记录三个模块组成，如图所示[11]。

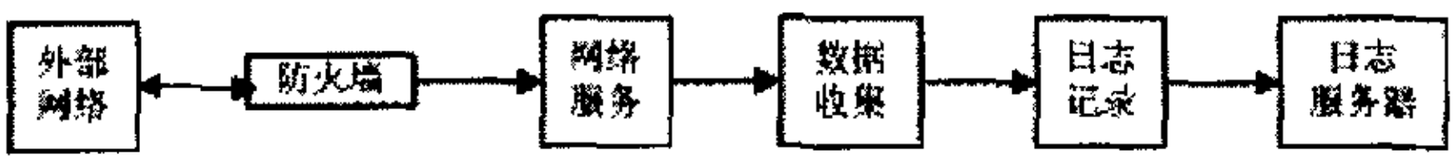

图 3.1 单机蜜罐构成

Figure3.1 Architecture of Single Honeypot

网络服务是蜜罐系统提供的经过伪装的服务，应该能够使攻击者相信这些都是正常的服务。此外，为了吸引攻击者的注意力，还可以提供一些便于入侵者进入蜜罐系统的漏洞。如果开放已知的安全漏洞，可能会导致一般类型的攻击；如果堵上已知漏洞，则可能发现新型的和意料之外的攻击方式，但是同时也降低了诱惑性。因此，蜜罐设计时需要精心设计开放的漏洞类型，以便起到欺骗作用并且有利于管理员进行精确的监视。

数据捕获模块是系统中最重要的组成部分，必须能够获得所有入侵者的行动记录，这些记录最终将有助于分析攻击者所使用的工具、策略以及攻击的目的。该模块的目的是在入侵者不被发现的情况下，捕获尽可能多的数据信息。捕获时可能需要对系统进行必要的修改(尽可能的少)，以免引起入侵者的怀疑。此外，还也可以增加对数据进行过滤的功能，在分析前如果能滤掉无用的信息，会给以后的数据分析带来方便。

日志记录模块将捕获到的信息按照既定的格式形成日志文件。在系统中运行着一个随时都处于激活状态的记录进程，由这个记录进程来监视是否有数据被捕获。如果没有，则继续等待；如果有新的数据被捕获，就立即用一个网络连接把处

理过的数据发送到日志服务器上。记录进程本身应该具有一定的隐藏性，不能轻易的被攻击者发现。

不同于传统的网络安全系统，蜜罐系统采用主动防御的方法，力图记录入侵者的攻击过程，然后分析其攻击方法，更好地维护网络的安全。因此，蜜罐系统自身的安全性就显得非常重要。如果轻易被攻破，系统不但失去了其应有的价值，反而可能给整个系统带来更大的安全威胁。蜜罐系统在设计时除了综合考虑功能性和实用性等多方面的因素外，还必须在更大程度上提高系统的安全性。增加了安全性改进的蜜罐系统如图 3.2 所示。

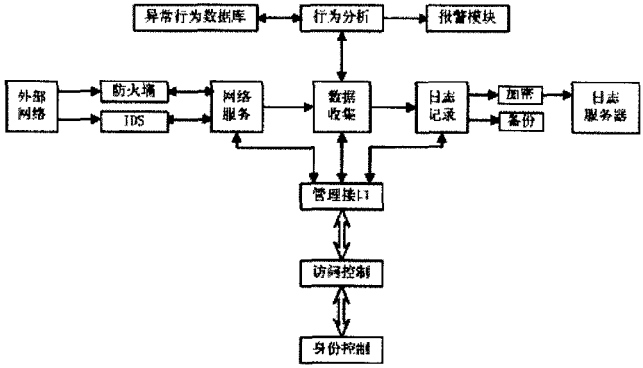

图 3.2 增加安全性的单机蜜罐结构

Figure3.2 Architecture of Single Honeypot with Security

3.2 蜜网

蜜网（Honeynet）是一种高交互级别的蜜罐。它的最基本目的就是收集潜在威胁的信息、发现新的工具、确定攻击特征以及研究攻击者的动机。蜜网实际上是一种蜜罐技术，是一种高交互的用来获取广泛的威胁信息的蜜罐。通过广泛的交互可以收集来自内部或外部的对目标组织机构的威胁信息。蜜网是由多个计算机系统组成的一整套网络体系。honeynet 的概念很简单，它仅仅包含一个或多个蜜罐，它也不具有任何产品价值，所以它本身不会有任何活动，也没有任何授权的服务。一切与它的交互都意味着恶意的或者未授权的行为。建立一个标准产品系统的网络，并将这个网络置于某种访问控制设备之后，同时进行观察。攻击者

可以探测、攻击并利用 honeynet 中的任何系统，honeynet 提供完整的操作系统和应用程序与获得者进行交互。Honeynet 中的产品系统可以是任何一种系统如：运行 Oracle 数据库的 Slaris 服务器，或运行 IIS Web Server 的 WindowsXP 服务器，或一个 Cisco 路由器，即 honeynet 中的系统必须是一个真正的产品系统。

Honeynet 是一种非常灵活的工具，它可以实现现有的 honeypot 的所有功能。在阻断攻击方面，由于它使用的是带有真正的应用程序的可信的系统，因而它具有更强的欺骗攻击者的能力。它几乎可以在任何操作系统和应用程序上运行，这使得它对系统攻击做出响应时有更大的灵活性。

从概念上讲 honeynet 是一种简单的机制，它的工作原理同 honeypot。它也是没有产品流量的资源，任何发送给它的信息都是可疑的，都可能是攻击者的一次探测、扫描或攻击，而从它发出的任何信息都暗示着它可能受到了攻击。这种思想更便于对攻击者活动的分析。它进一步扩充了 honeypot 的思想，它是由多个系统组成的物理网络，不是安装的一个产品，或放在网络上的一个设备，而是一种架构。在该架构上建立了一种高度可控的网络，网络中的机器上可以安装任何系统或应用软件。Honeynet 架构的有三个重要的元素：数据控制、数据捕获和数据采集。数据控制是对黑客攻击活动的控制，一旦黑客控制了 honeynet 中的某个 honeypot，就必须将他的活动包含起来，以使它的活动不会危害非 honeynet 系统。数据捕获是对 honeynet 中发生的所有活动的捕获。数据采集是多个 honeynet 捕获的数据的集合，主要用在分布式 Honeynet 部署中，对各个 Honeynet 收集的信息进行整合。

目前已有的 honeynet 有两种不同的架构。第一代的 honeynet 主要用来捕获黑客的活动，它能捕获大量的信息，并能捕获未知的攻击和技术。它在捕获自动攻击和经验不足的黑客方面特别有效，它主要用来捕获机会目标的攻击，它的局限性是指纹识别和数据捕获。第二代 honeynet 在数据控制和数据捕获方面取得较大突破。它容易部署且不易被检测到，它最大变化就是数据控制的改变，它是专门用于捕获和分析 Internet 上的威胁架构。

3.3 蜜罐的部署

3.3.1 蜜罐在网络中的部署位置

蜜罐被看作是一个不提供任何真实服务的标准服务器，因此它不需要被放置在网络中固定的位置。但是对于特定的用途来说，将蜜罐放置在某一个特定的位置可能会得到比其它位置更好的效果[12][13][14]。

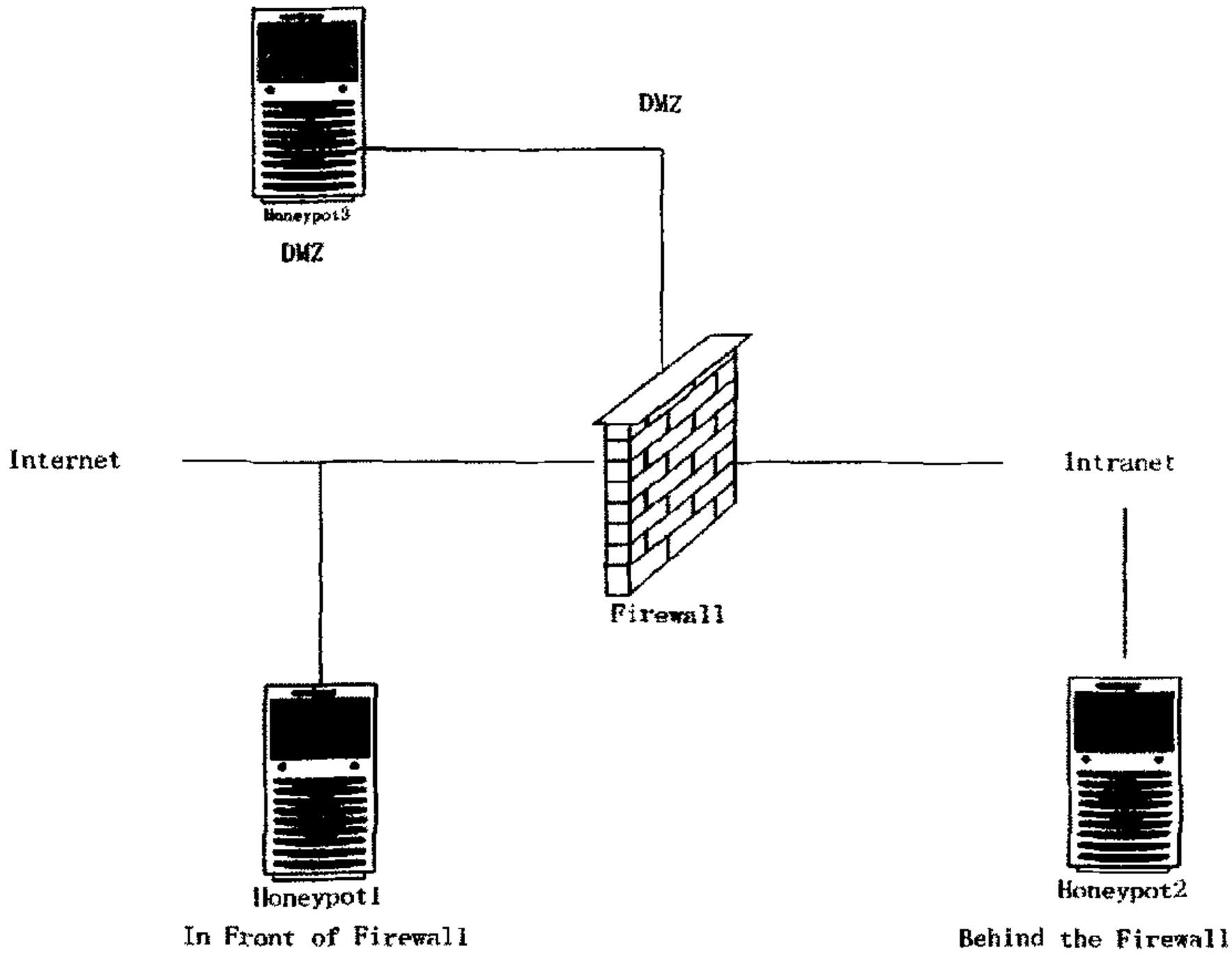

图 3.3 蜜罐在网络中的位置

Figure3.3 Position of the Honeypot in Network

蜜罐系统可以放置在网络的任何位置，可以在防火墙内或防火墙外。蜜罐一般是单个软件或系统，通常与要保护的网络系统放在一起，吸引攻击者的注意力。如果要保护内部主机的安全，可以将蜜罐系统部署在防火墙的网络内部。如果要保护 Web、Ftp、Http 等服务器的安全，则可以将蜜罐部署在防火墙的停火区，增强服务器的安全性能。

根据不同的需求，蜜罐可以被部署在防火墙之外，也可以被部署在防火墙

之内，或者放置在防火墙停火区，如图 3.3 所示。不同的放置位置有着不同的特点。

1. 位于防火墙之前(Internet)

将蜜罐放置在防火墙之前，会消除由于向内部网加入不安全设备而引入的种种安全性问题。同时，由于蜜罐位于防火墙的外部，因此它可能会吸引到大量的入侵者来对其进行扫描和入侵，而防火墙和内部网的入侵检测系统却不会对这些入侵事件进行记录，也不会产生相应的报警信号。如果将蜜罐放置在外部网，那么如此多的入侵事件就会使防火墙和入侵检测系统产生大量的报警信号。

除此之外，将蜜罐放置在防火墙的外部还能带来一个非常大的好处，那就是不需要再花费额外的时间去调整防火墙和入侵检测系统中的相应规则。因为在这种情况下，蜜罐被看作是位于内部网之外的一个单独的网络设备，这样就可以在不增加内部网威胁程度的前提下来运行一个蜜罐了。

放置在防火墙外部的蜜罐也有着其自身的缺点。在这种情况下，对于那些源于内部网的入侵者来说，蜜罐系统可能就已经失去其效应了。特别是在某些对输出连接进行限制的防火墙中，这个缺点就越发明显。

2. 位于防火墙之后(Intranet)

将蜜罐放置于防火墙之后会给内部网引入新的安全问题。特别是当在蜜罐没有同内部网安全隔离开的情况下，蜜罐的引入就很可能影响到内部其它网络设备的安全。

蜜罐中经常提供一些虚假的服务以吸引入侵者对其进行攻击，为了使入侵者能够访问到这些服务，就不可避免的要对防火墙中的相应规则进行调整。同时，入侵检测系统的特征库也需要进行调整，否则，可能会因为入侵蜜罐的事件太多，而产生大量的报警信号。

一旦蜜罐被侵入，那么随之而来就会产生一个新的问题：入侵者可能会通过把蜜罐作为一个跳板来对其它非蜜罐主机进行攻击。而在我们的防火墙看来，这些通信都是从我们内部网中的合法成员——蜜罐中发出的，自然也不会对其进行阻止。因此将蜜罐放置在防火墙之后，则蜜罐自身的安全性问题就是首要考虑的，在高交互性的蜜罐中更是如此。

虽然将蜜罐放置在防火墙之后会带来如此多的麻烦，但这样的部署也有它

特有的优势。在这种部署情况下，我们可以通过蜜罐来检测到源于内部的攻击者，同时，还可以通过蜜罐来检测防火墙规则配置的正确性。

3. 位于停火区(DMZ)

如果我们可以保证蜜罐对 DMZ 中的其它提供公共服务的网络设备来说是安全的，那么将蜜罐部署在 DMZ 中。在大多数的 DMZ 中，并不是所有的资源都可以被访问到的，只是一些被允许的服务才会被防火墙所允许。在这种情况下，如果把蜜罐放置在 DMZ 之中，为了要实现蜜罐的设计目的，就必然要使防火墙允许对 DMZ 中所有端口的访问，这种做法是非常危险的。

我们可以通过设置一个蜜罐自己的 DMZ 来解决这个问题。通过使用蜜罐自己的 DMZ 可以加强对蜜罐的控制，同时还可以增强其灵活性和安全性。然而这样做必然会需要增加相应的硬件设备，从而使成本提高。

对于一个单独的蜜罐来说，直接将其放置在网络之中，就可以引诱入侵者对其进

行攻击，从而对入侵资料进行收集。然而在这种情况下，只有蜜罐遭到入侵者的探测、攻击，相应的攻击信息才能被记录下来，而那些对受保护网络的攻击却只能按照传统的防御方法将其拒绝在受保护网络之外。

3.3.2 蜜罐在校园网中的部署

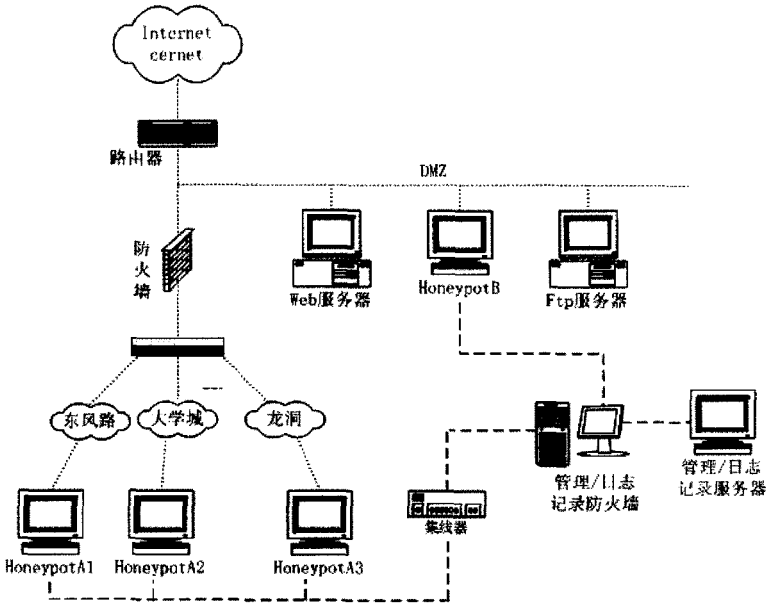

图 3.4 校园网 honeypot 部署总体结构图

Figure3.4 System Function Architeture of the Campus Honeypot

在校园网的安全防护中，要成功实现 honeypot 的首先要先确定使用它的目标，即希望它如何来实现防护功能，honeypot 不能取代防火墙、入侵检测系统，它是整个安全架构的一部分。由于校园网拓扑的复杂性，我在选择了在两个不同位置部署不同类型蜜罐，从而在一定程度上对现有校园网的安全防护进行改进与提高。其总体设计部署如上图 3.4 所示。

图中 honeypotA1 至 honeypotA3 部署在校园网各个校区。主要目标是要实现当攻击者成功穿越防火墙对校园网进行扫描或探测等攻击时，系统管理员要能立即收到警报。即此处部署的蜜罐可以检测到对校园内部网的成功的攻击。

对于 DMZ 区的蜜罐 honeypotB 主要用于协助对攻击事件的响应，此处的蜜罐是公共服务器的镜像。蜜罐必须以每种可能的方式来给真实服务器提供镜像，

以了解攻击是怎样进入系统的，从而对攻击行为作出响应。如将蜜罐作为 Web 服务器的镜像来保护真实的 Web 服务器，此时蜜罐有与真实的 Web 服务器一样的操作系统构成、相同的应用和相同的配置。所了解的对蜜罐攻击的每一件事情都将应用于 Web 服务器，这里应尽量减少到蜜罐的流量，同时还要提高黑客攻击蜜罐的可能性，让黑客以为自己攻击的是 Web 服务器[15]。从而在一定程度上保护真实服务器。

第四章 蜜罐系统的分析与设计

4.1 蜜罐的设计要求与目标

4.1.1 设计要求

虽然蜜罐有许多不同的类型及配置方式，但无论设计何种蜜罐都需考虑如下三方面的要求：

1. 适应性要求：这要求所设计的蜜罐系统必须能够适用于多种不同的网络环境，并且能够适应系统环境的改变，如增加环境中的计算机数量，改变计算机操作系统类型时，这时只要对蜜罐系统增加相应的蜜罐主机即可正常工作。适应性也包括蜜罐系统本身对其宿主平台的适应性，即：跨平台工作的能力，适应其宿主平台软、硬件配置的各种不同情况[16]。

2. 安全性与可用性要求：蜜罐本身是一种诱骗系统，理论上不提供任何产品流量，任何从蜜罐发出的连接都意味着该蜜罐系统可能被攻破了，此时应该能将该蜜罐系统隔离开，避免攻击者利用它作为跳板来攻击其他非蜜罐系统。而且对蜜罐本身来说也必须是完善和健壮的，不应该向所属的计算机环境中引入新的安全问题及安全隐患[16]。

3. 有效性要求：能够证明根据某种要求所设计建立的陷阱系统是切实有效的。即：对于攻击事件记录，能够保证它的真实性。这就使得有可能查找出攻击者的位置，能够记录下攻击者攻击过程的全部活动，并用这些记录作为起诉证据等。可使安全人员尽早知道哪些服务系统可能招致攻击，以及黑客攻击系统的目的等并对此做出防范。并可将攻击过程的技术资料作为研究黑客攻击技术的教材[16]。

4.1.2 设计目标

在我设计的校园网的安全防护体系架构中，引入蜜罐技术从阻止(prevention)、检测(detection)、响应(response)、这三个安全层次上提高对校园网的安全防护。具体地讲：通过欺骗或威慑来阻止攻击者；检测攻击时作为防盗自动警铃；响应攻击时要求能收集攻击活动的数据和证据，还要求能够通过捕获的数据和信息来研究攻击者在攻击时所采用的工具、战术和动机。

4.2 蜜罐系统的设计

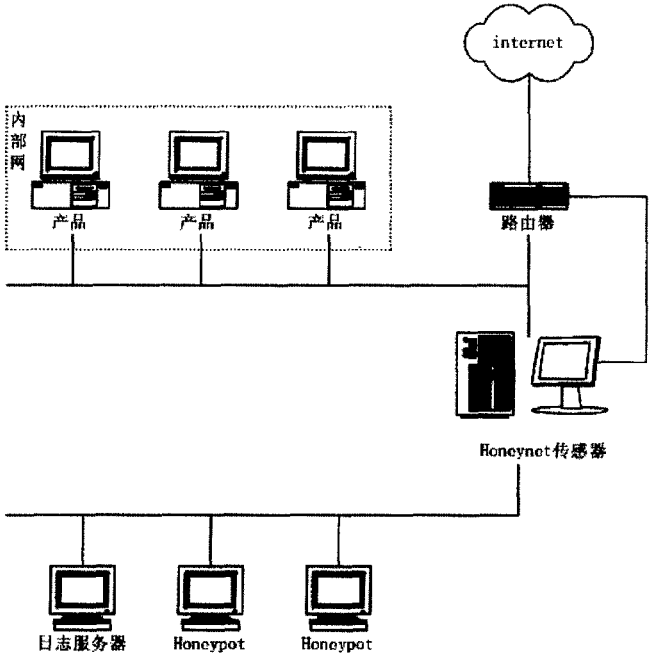

图 4.1 蜜网系统结构图

Figure4.1 Structure of the Honeynet

4.2.1 设计时的主要技术

蜜罐系统实现起来比较复杂，需要使用很多关键技术，如端口重定向技术、数据控制技术、数据捕获与数据采集技术等[17]。利用端口重定向技术，可以在工作系统中模拟一个非工作服务。数据控制用来减少危险，它通过限制 honeypot 的流入和流出来控制攻击者的行为。危险就是一旦攻击者攻击了某个 honeypot 系统，他就可以使用该系统攻击其他的非 honeypot 系统，数据控制就是控制通过已攻入的 honeypot 系统对非 honeypot 系统进行攻击，对数据控制有如下要求：数据控制既可以自动实现也可以通过人干预实现；为防止失败至少要有两层数据控制；要能够维持所有出入境连接状态；能够控制任何未授权的活动；它的执行可以由管理员随时进行配置；控制方式要尽可能能被攻击者检测到；当蜜罐系统被攻破时至少要有两种活动预警方法；要能够对数据进行远程管理，能够过程访问并管理数据控制机制。数据捕获技术要求在不被入侵者发现的情况下，尽可能的捕获攻击者的所有活动，数据捕获要求与收集技术获取尽可能多的信息，包括输入输出信息包、键盘和屏幕记录等，以便从中分析所使用的工具和策略。数据采集的目的在于集中管理由多个蜜罐捕获的信息，要保证捕获的数据从传感器传送到采集器时必须以一种安全的方式进行，要保证数据的机密性、完整性和真实性。

4.2.2 蜜罐系统的数据控制

如上章中的系统结构图所示，我设计的蜜罐系统使用了单一的 honeypot 传感器，并且它将 IDS 传感器和防火墙的功能结合在一起，在硬件上只用一个设备，因而在部署和管理上都相对简单。而且它本身是一个二层设备，这样难以被探测到，该设备对攻击者来说几乎是不可见的。每一个进出 Honeypot 的包都必须要经过 honeypot 传感器。由于使用的是二层设备，所以实际上所有的系统都是同一个网络的一部分，真正产品系统和蜜罐系统的分离仅在第二层发生。

它采用 IDS 网络技术，而不是依赖于第三层防火墙。IDS 网络关控制谁可以访问网络上的什么资源，它不但能根据这些服务来切断连接，而且能够区分恶意攻击和合法行为。IDS 网关具有签名数据库，当一个已知攻击与数据库匹配时，就会被切断连接，利用这种技术进行数据控制时，它的优势如下：首先它利用 IDS

网关来检测未授权活动，不是通过利用防火墙对出境连接数进行计算来阻断攻击者，采用这种技术跟踪攻击者的所进行的活动，有更强的智能性。第二个优点体现在对未授权活动做出响应的方式上，它不仅切断连接，而且可以修改甚至扼杀黑客的活动。通过修改第二层网关的包使黑客难以发现蜜罐系统对它的响应。例如，攻击者或能会对非 honeynet 系统进行扫描或发起拒绝服务攻击，这些攻击可以穿过第二层网关，但在这个过程中会丢失包甚至丢失所有的连接，网关还可以产生虚假的响应，如在切断连接的同时，给攻击者返回一个 RST 包，伪造一个连接[18][19]。总之，这种技术具有较难被发现的灵活的响应机制。

4.2.3 蜜罐系统的数据捕获

在蜜罐系统中实现数据捕获的难点是要在不被黑客发现的情况下捕获尽可能多的数据。实现数据捕获时要保证尽可能不对 honeypot 做任何修改，以防止黑客通过指纹发现蜜罐的存在。另外对所捕获的数据通信不能存储在本地 honeypot 中，以防被黑客发现并破坏其中的数据信息。在我设计的蜜罐系统中，我采用分层机制来捕获数据。即从各种来源中采集数据，在增加所收集信息的同时降低失败的风险。在分层捕获中的第一层借助防火墙，因为所有的流量必须通过防火墙，它所记录的信息只包一些包头信息，如攻击时间/日期、源和目的 IP 地址以及源和目的端口，这些信息有助于趋势分析和统计建模。第二层是 IDS 系统，honeynet 传感器集成了 IDS 的功能，它有两个界面：一个与蜜网连接，一个与真实内部网连接。IDS 传感器与蜜网的界面没有附带 IP 地址，所以攻击者无法攻击它，它能够捕获网络上的所有活动。它可以捕获并记录所有通过蜜网的包和载荷，它可以使蜜网在网络层对攻击进行分析，同时也可以捕获击键行为、工具包及黑客间的通信[20][21][22]。另一方面，它与产品网络连接可以达到远程管理和数据采集的功能。数据捕获的第三个层次是蜜罐本身，这主要包括捕获发生有蜜罐上的所有系统和用户活动，包括本地的和远程日志服务器。对于 windows 系统，可以借助第三方应用程序来记录系统日志信息。现在大多数的攻击者都会使用加密来与被黑系统进行通信。要捕获击键行为，需要从蜜罐主机中获得，如可以通过修改系统库或者开发内核模块来修改内核从而记录下黑客的行为。

4.2.4 蜜罐系统的数据采集

校园网的构建都是基于分布式网络的，因此在构建校园网安全的同时也必须是面向分布式环境设计开发的。在该环境中，部署在不同位置的多个蜜网由一个组织控制。此时对数据采集的要求就较高。对于这种分布式蜜网的部署，要求有对系统进行远程管理和采集捕获的方法，要保证多个蜜网能协同工作并共享结果。在数据采集过程中还要保证信息以一种安全的方式采集，主要保证信息的完整性、真实性和保密性。这可以通过在分布式蜜网到中央位置建立 IPSec 隧道来达到这个要求。加密保证传送的数据不会被篡改，每个蜜网都要向中央服务器验证其身份，而且保证没有第三方能看到这些数据。在系统总体结构图中从 Honeynet 传感器到路由器有一条连线，它用于数据采集和远程管理。它是与动态 IP 栈的第三层接口，允许网关与不在该网络上的其他系统进行通信，通过此连接，所有的数据都可以远程发送给中央采集点，同时实现对所有分布式蜜网的管理。在数据采集时还要考虑对所发送数据的格式的标准化。保证从多个蜜网中采集到的数据能够简单地实现共享和聚合[23][24]。

4.2.5 对收集数据的集中处理

由于我们的校园网是一个典型的大型分布式网络，这客观决定了将蜜罐部署在校园网中有其特殊性。为了在这个大型分布式网络中合理部署蜜罐，同时将各个子网的安全威胁进行收集，这里我用到蜜场和动态蜜罐的技术。将所有的蜜罐部署到蜜场中，在各个校区子网内部设置一些重定向器，借助于入侵检测组件检测到网络中有黑客发起的数据流，则通过内部网设置的重定向器将这些恶意流量重定向到蜜场中某个蜜罐机上，由蜜场中部署的数据控制和数据捕获机制对这些恶意流量进行跟踪、收集和分析。蜜场相当于一个集中的数据管理中心，将蜜罐集中部署在蜜场中在一定程度上减小了各个子网内的安全风险，并有利于对引入的风险进行集中控制。在重定向技术的引入使蜜罐具有更大的价值。

4.3 蜜罐系统的安全性改进

这是一个包含数据链路层、网络层、应用层控制的多级综合安全机制，实现对进入和流出蜜罐系统信息的安全控制和保护。连接控制主要是针对对蜜罐系统的启动的外出连接加以控制，以防止黑客利用蜜罐系统作跳板攻击其它非蜜罐系统主机；路由控制对来自蜜罐系统所在网络的包进行初步处理，以防黑客利用蜜罐系统对外进行地址欺骗(spoofed IP)；数据捕获从链路层抓包、分析，对出入蜜罐系统的数据包进行较详细的记录；远程日志由蜜罐机本身完成对系统日志的远地存储[25]。

从安全机制整体上看，连接控制拥有蜜罐系统外来和外出连接的记录，远程日志中记录了黑客在蜜罐系统中的操作过程，远程日志系统是一个有较高安全配置的系统，即使黑客能够发现并破坏了远程日志，数据捕获模块还留有出入蜜罐系统数据包的详细记录。从连接控制和数据捕获所得记录仍能对攻击所采用的工具、过程有全面了解。

蜜罐系统本身是一个应用系统，必然存在自身的安全性问题。同时，由于系统的特殊性，必然会受到有经验的攻击者的特别注意，因此，蜜罐系统自身的安全问题非常重要。蜜罐系统作为一种主动防御的网络安全系统，其软件和模块安装在一定的系统和网络上，可能遇到的攻击除了具有一般信息安全系统的特点以外，还有其他不同的特点。下面着重探讨如何针对这些恶意攻击行为进行防御的策略和方法。

对未经认证而对蜜罐系统的访问：入侵者没有预先经过同意，就关闭蜜罐系统程序进程的运行，使系统失去效用，或停止蜜罐主机系统，改变其监控设置，改变在蜜罐系统模块之间的通信行为。其表现形式有：假冒身份攻击、非法用户利用漏洞进入正常服务系统或蜜罐系统进行违法操作等。此外，入侵者可能通过各种渠道获得被攻击者的蜜罐系统，研究该系统的特点和效能，这样就有可能找到系统的弱点，或者绕过系统的一些安全机制，从而使蜜罐系统无法监测其入侵活动[26]。可以使用身份鉴别技术和访问控制技术来解决这种问题。身份鉴别技术通过对访问者身份的识别，确认访问者的身份，减少假冒身份攻击的可能，可以通过采用密码或者数字签名等方法来实现。访问控制技术则是针对越权使用资源

的防御措施,根据用户的权限来决定其行为，使系统能够在合法的范围内使用。对蜜罐系统进程的隐藏和保护可以减小被入侵者发现的可能，从而保证系统的正常运行。通过特定的技术在系统级隐藏和保护进程，一般的方法将无法发现进程的存在，即使是管理员也无法任意终止程序的运行，只有特定的用户才具有操纵进程的权限。这样，在一定程度上保证了系统的安全性。

对数据丢失和完整性的破坏：蜜罐获得的关于入侵的日志或其他数据可能被泄漏和丢失；通常包括日志数据在传输中丢失或泄漏、数据在存储介质中丢失或泄漏、通过非法手段篡改异常日志数据等。数据完整性的破坏包括：以非法手段窃取数据的使用权；删除、修改、插入或重复某些重要日志信息，以取得有益于攻击者的响应或者隐藏其攻击行为;恶意添加、修改配置数据，以干扰蜜罐系统的正常运行。在蜜罐系统中,网络传输的数据采用适当的加密措施，并且在应用程序之间传递数据时也进行加密和验证，防止攻击者通过监听或其他方式观察到数据的存在，即使攻击者获得了数据，也很难进行解密以获得原始的信息[26]。同时,对系统的日志文件和重要的配置文件也要进行加密和保护,隐藏相应的文件和目录，使一般的方法和机制无法发现其存在，只能由特定的用户使用特定的程序才能访问。另外,还可以对日志等重要文件进行备份，即使遭到破坏，也可以在一等程度上恢复原来的数据。

对拒绝服务攻击：拒绝服务攻击不断对主机系统和蜜罐系统进行干扰，改变其正常的工作流程，或者执行无关程序使系统响应减慢甚至瘫痪，影响正常监控功能程序的运行，甚至使管理用户被排斥而不能进入蜜罐系统或不能进行有响应的操作。由于蜜罐系统和日志系统对网络通信都非常敏感 ，因拒绝服务攻击而产生的大量数据包，会给系统带来不可预计的后果。解决办法是:除了在系统设计时增加防范措施的模块以外，在编程时还要加强对代码的分析与测试工作，确保在内存及其他资源的使用方面没有潜在的问题，并减少对系统资源的依赖；同时,还要进行系统漏洞的及时修补工作，必须具有细致的完整性检查和周全的错误处理机制，确保不会在处理一些恶意的及非正常行为时发生意料之外的错误[26]。

4.3.1 连接控制

借助防火墙和入侵检测系统对流入和流出蜜罐系统的流量进行控制。使出入蜜罐的任何信息都必须经过防火墙，防火墙规则定义为允许所有数据包从外网进入蜜罐系统，而对由蜜罐系统发出的连接要进行控制。具体可采取攻击包抑制和连接数限制这两种方法。攻击包抑制就是借助入侵检测系统中 snort_inline 对流出的数据包进行检测，如果检测到含有攻击特征的数据包，发出报警信息并将攻击数据包丢弃；限制对外连接数就是借助 LINUX 防火墙内置的 IPtables 中设置连接数最大值，当由蜜罐发出的向外连接数超过这个最大值时，将其记录同时阻断以后的所有连接，以防止黑客将利用蜜罐作跳板对其他系统进行攻击[28]。黑客完成对蜜罐系统的扫描、探测、或利用某种安全漏洞成功完成权限升级后，通常，其下一步会将蜜罐系统作为跳板，对别的主机发起攻击。上述连接控制就是为解决这个问题提出的，要保证黑客通过蜜罐系统对其它主机的攻击企图无法完成。根据入侵者发出的连接请求和操作命令，作出决策是否切断连接。连接控制的另一个功能是记录所有进入蜜罐的流量和从蜜罐发出的连接的情况，这些信息是很重要的，因为进出蜜罐系统所有连接都是可疑的，这些数据也是对蜜罐系统进行攻击分析和来源分析的初步资料。

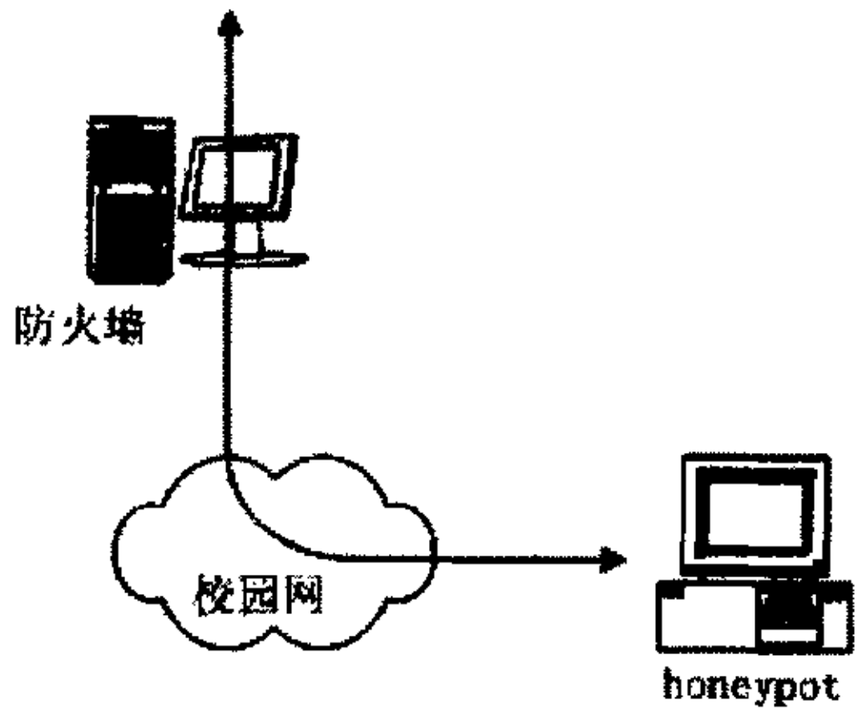

图 4.2 借助防火墙实现数据控制

Figure4.2 Data Control by Firewall

4.3.2 数据捕获

数据捕获是对所有经过蜜罐的流量进行记录，并以规定的格式对所有记录的

信息保存。捕获的数据用于分析黑客的活动，它们是有关攻击过程的详细信息，包括了攻击工具、命令等细节。可以针对特定的服务(如 telnet, ftp)在应用层对取得的数据加以处理，得到有关黑客操作的 ASCII 码信息，实时显示黑客的攻击过程。二进制日志和 ASCII 日志按日期分类存放，以便于数据分析。

对于数据捕获也借助于入侵检测组件，在获取网络数据的同时，对一些可疑的操作进行报警，提醒蜜罐系统的防护人员关注黑客的入侵，更好地对入侵过程进行监控。

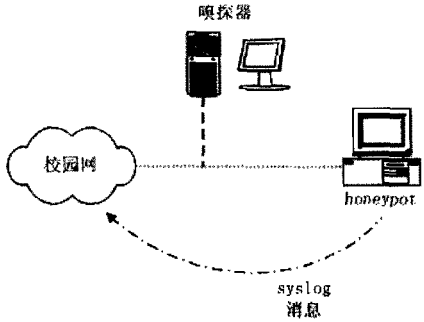

图 4.3 为 honeypot 部署一个嗅探器以捕获更多数据

Figure4.3 Depolying a Honeypot for Capturing More Data

4.3.3 远程日志

系统日志记录了系统内部所有进程的活动和用户的操作。对于蜜罐主机的日志采用异机远程存放。采用较强访问控制并修补了漏洞的 windows 系统存放远程日志，并运行 syslog 服务。

经验丰富的黑客可能会对远程日志服务器发起攻击，这一点我们无须担忧，因为：远程日志服务器是一个安全设置相对较强的系统，如果黑客能够攻破它的安全防护，必会用到更高明的攻击方法和技巧。

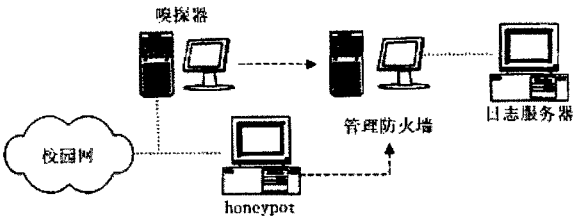

图 4.4 日志记录

Figure4.4 System Log

第五章 系统各部分组件的实现

5.1 总体设计

5.1.1 目标确定

在校园网安全防护中，引入蜜罐目标是提高识别成功攻击的能力以及分析攻击者怎样攻入的能力。利用蜜罐对攻击者活动进行提前预警和事后响应。根本上是想利用蜜罐技术直接提高校园网的安全性，所以在这里我选择产品型蜜罐。

对于攻击者恶意攻击活动的提前预警可以根据对所收集的攻击信息进行分析研究，借用现有的数学方法如运用过程控制原理，回归分析，移动平均自回归求和，及我提出来的将要介绍的神经网络 BP 学习算法构造并改进预警模型。当有攻击者成功地穿越了防火墙并访问校园网时，这时要求设计部署的蜜罐要能即时发现攻击者并立即向管理员发送警告。校园网内部主要由基于 Windows 的桌面系统组成，特别是 Windows2000 和 XP。一般认为内部攻击者是从某个某个局域网的桌面系统攻入并获得了校园网的内部信息。因此在校园网内部网上部署的 honeypot 主要用于检测，要能够检测成功的攻击。鉴于此目的，我在每个校区的网络中部署一个蜜网，因每个校区位于不同的地理位置上。在部署蜜网时同时要考虑后期的维护工作，所以对部署的蜜网要求引入的风险要尽可能地小，要求容易部署、可以远程管理并且仍然能够可靠地检测到攻击的方案。因为在校园网内部大部分系统都是基于 Windows 的，因此在部署蜜网时我采用 Windows 操作系统平台，这样可以确保在底层的 IP 栈的操作系统类型和蜜网模拟的服务是一样的。这里我的蜜罐选择 Specter，用它来构建蜜网，因为它可以进行远程管理，而且有全部的预警、日志记录和检测选项。

增强校园网安全的第二个目标是指出攻击者怎样进入的能力。这同样可以通过部署主要用于对事件进行响应的 honeypot，通过对攻击者作出反应来确定攻击者怎样进入系统，同时跟踪并记录它们的活动。鉴于此要求这里部署的蜜罐有高的交互性，从而能够捕获大量的信息，包括攻击者的击键行为、工具包及攻击者

间的通信。一般认为外部攻击者通过 DMZ，比如 Web 服务器来穿越校园网，我们学校园网 Web 服务器操作系统为 Solaris，所以我建立一个蜜罐来提供 Web 服务器的镜像。如果攻击者再次进入，他很可能会以相同方式通过 Web 服务器和 honeypot，这样就可以检测并捕获攻击者的所有活动了。要求此处用于响应的蜜罐有高的交互性，因此需要和 Web 服务器相同的操作系统，不能仅仅模拟一个完整的操作系统的所有可能性。在此我选择 ManTrap 这种蜜罐，它具有高的交互性同时可以作为一个 Web 服务器的镜像。它能够捕获攻击者的击键行为和内核的系统活动，它还可以创建多达 4 个逻辑系统的能力[13][15][30]。

5.1.2 部署 honeypot

通过上面对安全要求目标分析，决定蜜罐部署方案，在内部网中使用蜜罐主要目的是用于检测，因此选择部署低交互度的产品型蜜罐 Specter；在 DMZ 区要求能够对检测的事件做出响应，在此选择高交互度的 ManTrap 蜜罐，同时要求部署的蜜罐能够镜像提供公共服务的服务器。

计划在每个校区内部署一个检测型 honeypot 组成的蜜网。如果攻击者访问任何一个校区的网络，都应该有一个蜜网中的 honeypot 能够检测到攻击。对于 DMZ 区的蜜罐主要用于协助对攻击事件的响应，此处的蜜罐是公共服务器的镜像。蜜罐必须以每种可能的方式来给 Web 服务器提供镜像，以了解攻击是怎样进入系统的，从而对攻击行为作出响应。蜜罐需要和 Web 服务器有一样的操作系统构成、相同的应用和相同的配置。所了解的对蜜罐攻击的每一件事情都将应用于 Web 服务器，这里应尽量减少到蜜罐的流量，同时还要提高黑客攻击蜜罐的可能性，让黑客以为自己攻击的是 Web 服务器。

对检测型蜜罐效率的优化比较直接，由于校园内部绝大部分系统都是基于 Windows 的，因此让蜜罐模拟 Windows 系统。由于想要检测出所有的攻击，因此需要监听全部的服务。使用 Specter 蜜罐可以监听全部的 13 个缺省端口，对于可选端口，我设置监听 TCP139 或者 NetBIOS，这些端口常用来提供基于 Windows 服务，并且经常被攻击。这样设置能提高检测的机会。这些蜜罐不提供任何产品流量，所有到蜜罐的流量都有可能是一次探测、扫描或攻击。

对响应型蜜罐，要区分开真正的 Web 服务器和对它镜像的蜜罐是非常关键

的。首先需要尽量减少进入蜜罐的流量，蜜罐没有在校园网的任何一条 DNS 记录中列出。当外部网用户达到校园网主页 http://www.gdut.edu.cn 时，他将不会在 DNS 中找到蜜罐的 IP 地址，并且会被定向到 Web 服务器中。这意味着蜜罐中即使有，也只有少量的产品流量，这使得可以很容易的检测可疑的或者未授权的活动。可以在提供公共服务的服务器上创建一个到蜜罐的 http 连接，并假设这个连接是一个留言簿。这个留言簿是一个基于 Web 的应用程序，它允许每个访问你的 Web 站点的访问者在这个页面上签名。当有攻击者浏览 Web 服务器时，他将看到一个要求他在这个留言簿上的签名链接。如果他点击这个链接，他将到真实的留言簿。但他不会意识到这个留言簿是在 Web 服务器 honeypot 上运行。留言簿的价值是捕获高级攻击者。高级攻击者通常不会扫描整个校园网的 DMZ 中的每一个 IP 地址，他们会耐心检查每个提供公共服务的机器，寻找特定的弱点。通过增加公共服务器到蜜罐的连接，让高级攻击者跟踪这个链接，并将注意力放到蜜罐上。当攻击者进入 Web 服务器上的留言簿时，他们同时也就陷入了蜜罐。

对于各校区内所使用的低交互度的蜜罐而言，降低其危险性有限。模拟服务限制了攻击者访问操作系统或者使用蜜罐攻击其他系统的能力。这里必须保证 Windows 平台是绝对安全的。为了防止对蜜罐系统的指纹识别需要将蜜罐和真实产品系统使用相同的命名规则，统一在校园网内标识。

对于 DMZ 区部署的蜜罐，要保证一旦蜜罐系统受到攻击，能有有效的数据控制机制确保蜜罐不被作为跳板来攻击内部网。其数据控制可借助于对防火墙的规则库进行设置来实现。限制 Web 服务器蜜罐遭到了攻击后不能启动到内部网的连接。这样防止黑客将蜜罐作为跳板去攻击内部网。一旦有蜜罐发向内部网的连接，防火墙立即阻塞它，同时给安全管理员以警报。部署在这里的蜜罐同时配置为包括运行具有留言簿功能的高交互度的 Web 服务器蜜罐，从而防止黑客对它进行指纹识别。

在这个部署方案中还包括对蜜网和蜜罐的集中管理和日志记录。在整个校园网中由各校区分布在不同地理位置，所以需要对它们进行远程管理，除此之外还需要一些集中收集所有预警和所捕获的数据的方法。每个蜜网都会有两个接口：一个用于连接到真实产品网络，一个用于管理。蜜网中的所有管理和日志记录都发生在专用网中。通过在各内部蜜网与部署在 DMZ 区的蜜网中布置日志防火墙将两个范围的蜜网分离，保证某个蜜网中某蜜罐被攻破了不影响蜜网中其他蜜罐

或其他蜜网的可以正常检测与响应工作。

5.1.3 维护 honeypot

为了集中捕获数据和进行预警，让所有的蜜罐将全部的日志和预警转发到中心日志服务器中，这个中心日志服务器将对这些信息进行监视和归档。当蜜罐检测到一次探测、扫描或者攻击时，让它直接向系统管理员发送 email 预警。Specter 和 ManTrap 均支持这项功能；也可以让进程监视中心日志服务器，任何时候只要有系统产生了一条日志，它就会通过 syslogd(UDP514)将其发送给远程的日志服务器。远程日志服务器上的一个进程会监视这些日志、检测新的记录项并通过寻呼系统对管理员进行预警。

对于检测型蜜罐，预警机制是很简单的。无论何时只要有人发起了对蜜罐的连接，它们就会发出预警。由于它们不提供任何产品服务，所以任何发往它们的连接都很有可能是未授权的活动，这时意味着有人入侵了网络防线。对于这样一种预警，需要关于攻击者是谁及其具体活动的信息。假如攻击者向某内部 192.168.1.0/24 网络企图发起一次到 honeypot 的未授权 FTP 连接，这时由 Specter 会产生预警，提示我们攻击者正试图攻击 192.168.1.101，并且在扫描内部网中的其他系统。

对于部署在 DMZ 区的 Web 服务器蜜罐来说，预警要稍微复杂些。因为这个蜜罐提供一个带有留言簿登录的 Web 服务器，它只会得到少量 HTTP 流量，而此时不能只对 HTTP 连接进行预警。防火墙应该允许入境 HTTPS，或端口 443 流量，因为校园网有可能会升级到 SSL 以实现到 Web 服务器的加密连接。如果在端口 80 或者端口 443 这外的任何端口上出现了连接，也要进行预警。意味着有未授权活动。还可以使用防火墙本身进行预警。如 HoneypotB 企图连接到内部中的其他任何系统，防火墙就会检测并阻断这些企图。

下一步是对接收到的预警进行响应，对于检测型蜜罐的反应过程比较简单：识别出源攻击者并降低威胁。如果检测型某个蜜网中的蜜罐向内部某系统的发出连接请求，意味着内部某系统很有可能被攻破了。这时应该立即断开蜜罐与系统的连接，使攻击者不再使用蜜罐作为跳板对其他系统进行攻击。然后对系统进行分析以了解攻击者的活动。如果某个检测型蜜罐检测到了来自外网的一次成功攻

击，很有可能意味着攻击者已经绕过了防火墙。该检测过程查看防火墙规则库，确认攻击是有效的，如果威胁是真实存在的就修改防火墙规则库，从而达到在防火墙端阻断攻击者。

响应型蜜罐一旦发现了一次成功的攻击，就断开蜜罐到系统的网线。这样来确保攻击者无法再次攻击或者危害其他系统，并且还可以进行一次完整的分析，以了解攻击者是如何进来的及其所进行的活动是什么，以免在未来遭受同样的攻击。同时会对真实 Web 服务器进行查看，了解是否出现了任何成功的攻击；并对其进行严密监视，直至对反应蜜罐的分析完成为止。

最后还需要制定一个确保蜜罐保持最新的计划。需要一个过程来保证蜜罐软件及其底层平台是最新的，并且对于新发现的脆弱点来说是安全的。为了维护蜜罐，遵循用于所有真实系统的相同的补丁策略。校园网有一个定义良好的过程，可以保证所有的真实的系统都是每周更新的。管理/日志网络为我们提供了一种在所有的蜜罐系统上远程安装的这些补丁的简单有效的方式。

5.1.4 Honeypot 的模拟服务及配置简介

1．Specter 简介

Specter 可以模拟 FTP、TELNET、SMTP、FINGER、HTTP、NETBUS、POP3 七种攻击者可能会与它交互的服务。除了七种预定义的服务外，Specter 还提供了一种称为 trap 的事件，它是一些对所有连接进行侦听、检测和日志记录的预指定端口。攻击者没有与端口或者服务进行交互，所尝试的连接均被 honeypot 悄悄记录下来。Specter 本身带有 6 个预定义的 trap 和一个可定制的 trap，从而允许指定一个端口对所有的活动进行监听和检测。可以利用这个可定制的 trap 调整 honeypot 以适应那些新的威胁和漏洞。可以为 Specter 赋予自己的命名、域地址或者一条定制的警告消息[31]。

Specter 会监听所选定的服务和 trap 端口。使用 netstat -a 命令会给出选定并启用为打开和监听的那些端口。它无法侦听已属于另一应用程序的端口，而只能对尚未被其他应用程序所占有的端口进行监视。在模拟操作系统时，Specter 只动作在应用程序级。意味着它只能根据 7 种模拟服务来模拟所选的操作系统。IP 栈没有被加以模拟。对于 Windows 类型的 honeypot，一种有效方法为：让基础操作

系统和模拟的 Windows honeypot 保持完全匹配，从而使得 Specter 年上去更像一个真实的目标。[31]

在 Specter 图形界面中选择所用的操作系统为 windows 2000，对于特征我这里选择 Open，此时将会创建一个带有大量易于访问服务的系统。如果攻击者打算连接到 SMTP 端口并验证 honeypot 是否具备邮件中继功能，honeypot 就会模拟出存在的漏洞。

在服务中将七种模拟的服务全部选择，FTP：文件传输协议,监听端口 TCP21。Telnet：一种用于远程管理系统的明文协议，监听 TCP23。SMTP 简单邮件传输协议，一种用于收发 email 的明文协议，监听端口 TCP25。FINGER 用于获取远程系统中的用户信息，监听端口 TCP79。HTTP 监听 TCP80，在 Specter 所提供的全部服务中，这是最有可能被探测和攻击的服务。NETBUS windows 木马，监听端口 TCP12345。POP3 邮局协议。客户端用于收取 email 的明文协议，监听端口 TCP110。在这些服务中，有一些具有极高的可定制性。如 POP3 是邮件客户端。攻击者会经常对这项服务进行探测，如果发现漏洞就会试图窃取帐户、密码甚至邮件。利用 Specter 可以定制用于让攻击者从模拟 POP3 服务中获取的 email 信息了。这就赋予用户对服务进行自定义的灵活性，从而创建了一个更为真实的 honeypot。[31]

智能选项是一组带有 11 个选项的选择，可以让 honeypot 主动获得更多关于攻击者的信息。Traps 是 Specter 进行监听的按选定端口，只是对连接进行了日志记录。Specter 有一个可选的第 7 个端口，可以将其配置为监听所选的任一 TCP 端口。如下有 6 个预定义的端口：DNS 域名服务，用于解析域名或传输域文件，监听端口 TCP53。IMAP4 internet 消息访问协议，由客户端用于收取 email 的明文协议，监听端口 TCP143。SUN-RPC 端口映射或者远程过程调用，监听端口 TCP111。SSH 安全 shell,用于对系统进行安全远程管理或者文件传输的加密协议，监听端口 TCP22。SUB-7 windows 木马，监听端口 TCP27374。BO2K 一种 windows 木马，监听端口 TCP54320[31]。

密码类型主要用于欺骗攻击者，前面对于 honeypot 的特征我设置为 Open，此时攻击者可能会从 honeypot 处获得一份伪造的密码文件。此处可以选择攻击可获得的密码文件类型，从而让攻击者在捕获密码文件时花费大量时间去破解密码。

通知一栏为以何种方式将攻击通知给管理员。选择 Incident Database，确保

所有的检测活动都会被日志记录在 honeypot 本地，且日后可由 Specter 的 Log Analyzer 功能检索。同时选择 Alert 和 Short email 用于远程通知机制。两个额外的日志选项 Event 和 Syslog。Event 将信息日志记录到本地的 windows 操作系统中，并由事件查看器来查看。Syslog 将信息记录到一个远程的日志服务器，使用的是标准的 Syslog（1M）功能。

在控制台右边，是附加配置和定制选项。第一项给系统提供一个主机名，可以对其身份进行定制，下面是几项用于配置邮件服务器 IP 地址和通知 email 地址的选项。如果需要过程管理 Specter 需要配置管理服务器的远程 IP 地址、确定希望让管理连接发生的端口、并给出用于远程管理的密码。

对于 Specter 检测和信息收集有 3 个核心部分。第一部分为预警，告知管理者何时检测到了攻击活动。第二部分为通知后发出查看捕获信息。Specter 支持几个日志方法，包括 Log Analyzer，这是一种用于对保存在 honeypot 本地的日志活动进行分析的单个应用程序。第三部分为智能收集，这是 honeypot 用于主动获取攻击者信息的一项特殊功能。

Specter 可以获取相关启动攻击者和系统的信息。支持 11 种不同的选项，以主动获取与其连接的任何系统的消息。Finger 远程系统，用于获取用户信息。Tracer 对那些远程连接到攻击 honeypot 的系统进行 finger。Portscan 扫描远程系统中的开放端口，Whois 会使用 ARIN 或者 RIPE 数据库查询远程系统。DNS 解析攻击 IP 的地址名。Telnet Banner 连接到远程系统的 Telnet 端口并获取其 Telnet 登录标识。FTP Banner 连接到远程系统的 FTP 端口并获取 FTP 登录标识。SMTP Banner 远程系统的 SMTP 端口并获取其 sendmail 标识。HTTP Server Header 连接到远程主机的 HTTP 端口并发出 HEAD 命令。HTTP Document 连接到远程主机的 HTTP 端口并发出 GET 命令。Traceroute 使用 ICMP 跟踪到远程主机的路由。

对于 Specter 的预警主要采取如下两种通知方法：Short Mail 和 Alert Mail。Short Mail 目的在于通过传呼或者蜂窝电话向个人发出通知，内容限制在少量字符上。优点是能够通过用于向配有传呼的网络管理员进行预警，但它只能告知检测到了一次攻击。Alert Mail 是包含详细攻击信息的标准电子邮件，管理员通过电子邮件接收到关于试图探查或破坏的关键信息，这种预警机制提供了大量的攻击信息，从而能够有效的对攻击者做出反应。

Specter 本身带有内置的用于对探测和攻击进行日志记录和归档的数据库，该

数据库为一组文本文件的集合，每次攻击都会被赋予记录每次攻击的文本文件。Log Analyzer 为一种构建在 Specter 中的 GUI，允许对这些日志归档进行查询和查看。Log Analyzer 提供了一种查询机制，使得可以根据时间和来源对攻击进行查询和检索。Specter 还支持 Event Log 这种日志记录。这是大部分 Windows 系统具有的日志机制，并且 Specter 活动可以被记录到日志。这种日志记录也只记录少量信息，优点在于如果有第三方的应用程序监视这些日志文件，这些应用程序也可以对 Specter 活动进行监视和回应了。Syslog 是另一种用于对 Specter 所捕获的事故进行日志记录的方法。这是一种标准的 UNIX 实用工具，可用于在网络上从一个系统到另一个系统进行信息的日志记录。优点是可以从多个 honeypot 处收集载入日志的攻击并集中保存到一台远程的日志服务器中。日志服务器上的进程或者应用程序可以监视所记录的活动，并根据特定的签名或者预定的活动发送预警。

2．ManTrap 简介

ManTrap 是由 Recourse Technologies 创建、维护的高交互度的蜜罐。它的设计目标在于不仅被攻击而且还要被摧毁。ManTrap 创建了一种可以和攻击者进行交互的高度受控的操作环境。它创建了一种含有虚拟牢笼的操作环境，这种环境只是在逻辑上受控制，攻击者一旦进入则无法退出并且也无法攻击主机系统，每个牢笼都是一个全功能的操作系统，而且可以对牢笼进行定制，可以在这个环境上创建用户、安装应用程序、运行进程、甚至编译自己的二进制文件。当入侵者攻击并获得了某个牢笼的访问权时，对攻击者来说这个牢笼是一个隔离的物理系统，它并不知道自己正处于一种牢笼，不知道它的攻击行为已被记录。在这种环境中主操作系统能够对攻击者进行监视和控制。

ManTrap 可以在一个单独的物理系统上创建出四个虚拟的操作环境。实际上就是在一个物理系统中创建了四个 honeypot。ManTrap 没有模拟系统而是使用了牢笼技术，因此逻辑牢笼与基础操作系统是相同的。ManTrap 的缺陷是只能支持某些特定的操作系统。ManTrap 在提高安全性方面的的作用主要表现在它的强大的检测与响应。ManTrap 使用了两种不同的功能进行检测。首先它具有侦听特定端口的应用程序，由应用程序进行攻击检测和交互，它运行的是有效的应用程序而不是模拟服务；它的检测功能还表现在它可以检测不侦听的端口或者服务，这主要通过借助一个嗅探器被动检测所有发往任一牢笼任一端口的连接。ManTrap 运行着真实的服务。它对攻击的响应体现在一旦攻击者摧毁并获得了对某个牢笼

的访问权，他们的所有活动都被记录下来，包括新进程的启动和击键，即使他们使用了诸如 SSH 之类的加密手段来访问系统，他们的活动仍然能够被捕获到，在对攻击进行响应，这些信息就显得特别关键。

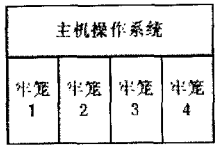

图 5.1 一个带有 4 个逻辑牢笼的 ManTrap

Figure5.1 A ManTrap with 4 Jails

5.1.5 系统功能模块图

蜜罐系统整体结构如图 5.1 所示，由主程序模块、数据控制模块、数据捕获模块和日志记录模块四大部分构成。

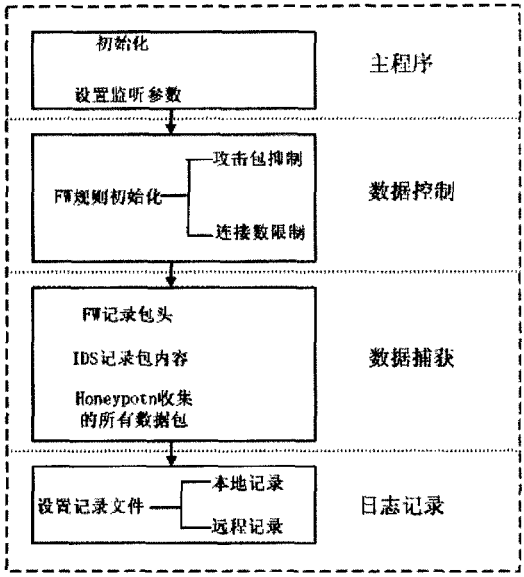

图 5.1 蜜罐功能模块图

Figure5.1 Function Architecture of the Honeypot

1. 主模块主要负责系统初始化工作，并对收到的命令进行分析转到相应的监听界面，对需要进行的监听的端口、服务等设置初始值，同时转入对数据控制和数据捕获模块的调用。其功能图如图 5.2。

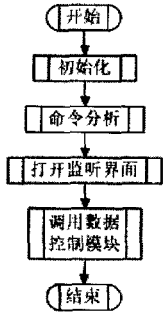

图 5.2 主程序功能图

Figure5.2 Function Architecture of Main Programme

数据控制和数据捕获是系统的核心模块，这两个模块在一定程度上无明显的分界，数据控制主要完成对流经蜜罐系统的数据的控制。这从两方面来考虑，首先对流入蜜罐的数据，由于蜜罐设计的最初目的就是让攻击者顺利进入并攻击它，所以对于流入蜜罐的数据不进行任何限制，而且由于蜜罐本身不提供真实产品服务，任何进入蜜罐的流量都被认为是有攻击企图的；另一方面，对从蜜罐发出的连接要做限制，要能设置适当的外出连接数目，如果设置太少，由蜜罐引入的风险相对较小但对有经验的黑客较容易识别到蜜罐的存在，如果设置较多，让黑客难以对蜜罐进行指纹识别，可以获得更多的黑客与蜜罐系统的交互信息，但同时也带来的风险也较大。数据控制就是要控制蜜罐向外发出连接数量，防止黑客将蜜罐作为跳板对其他产品系统进行攻击。

数据捕获是将流经蜜罐的数据全部捕获，收集黑客对蜜罐攻击的全部信息。对黑客攻击信息的捕获可来源于三个设备，即防火墙、入侵检测系统和蜜罐本身。防火墙捕获的数据信息主要包括攻击日期/时间、源和目的 IP 地址以及源和目的端口等数据包头信息；IDS 可以捕获所有的网络活动，包括数据包载荷，而且它可以捕获攻击者的全部击键行为，工具包及攻击者间的通信，IDS 系统可以提供特定连接的详细信息；由蜜罐本身也可以捕获所有关于黑客的攻击活动。它的功能模块如下图 5.3。

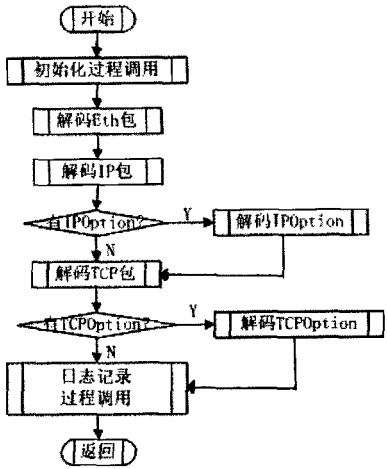

图 5.3 数据控制和捕获功能图

Figure5.3 Function Architecture of Data Control and Data Capture

系统最后一个模块是日志记录模块，对收集到的信息要进行日志记录，要在本地和远程分别进行记录，在本地进行记录时尽可能不对系统应用程序或服务进行修改，以防止被攻击者发现，引起黑客的怀疑，一旦被黑客发现他们被跟踪并记录，刚他们会想方设法销毁他们的攻击迹像，会对本地的日志记录修改或者删除，或者提供虚假信息。因此还要在远地进行日志记录备份，以便对黑客的真正攻击动机，手段和技术进行分析。日志记录功能图 5.4 所示。

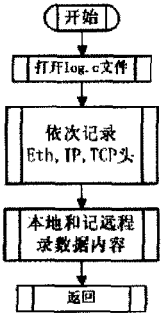

图 5.4 日志记录

Figure5.4 System Log Record

最后对收集的数据进行分析，借用神经网络的学习功能，将初步分析的原始数据作为神网络学习的样本输入，通过构造神经网络前向多层 BP 模型（如图 5.5 所示）对初始样本进行训练学习，采用这种方式既可以对现有预警模型进行优化使它更准确，也可以按照原始数据经过学习训练形成新的预警模型。由此可见将蜜罐技术引入网络安全架构中，在对攻击发生前的预警比传统安全设备的防护有更大的可靠性。

5.1.6 BP 算法构造预警模型

网络安全的第一道防线要求在攻击者进入系统前能对其进行有效阻止。蜜罐的误报率很低，因此在很多方面比传统的检测技术更容易使用，更有效。而现存的对攻击检测的技术对攻击预警还不够精确，因此，在该系统中我将借用神经网络中 BP 网络模型的算法来精确构造安全预警模型，在阻止威胁这个层次上提高安全性能。

神经网络模型以神经元的数学模型为基础来描述。神经网络最大的优点在于他的学习能力，它能够通过自动学习训练达到对模型的精确化。BP 神经网络是一种反向传播的神经网络，是目前最重要，应用最多的神经网络算法。我在用 BP 神经网络实现构造预警模型时，我采用典型的三层结构：输入层——中间层（隐层）——输出层，其中隐层只有一层。输入层神经元的个数为蜜罐系统收集并初始分析过的信息量数目，对应每个初始攻击信息给出相应的初始权系数，输出神经元个数为一个，用于输出并构造精确的预警模型，中间神经元个数与输入神经元数目相同。其具体结构如下图 5.5 所示。

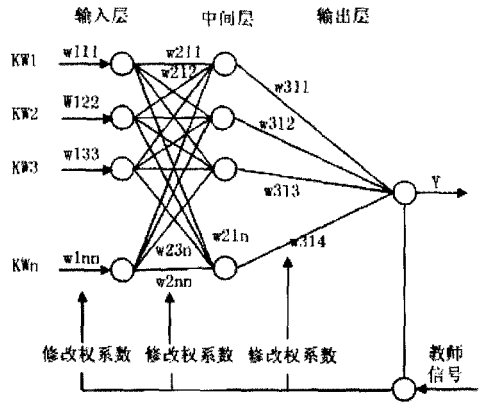

图 5.5 BP 神经网络模型

Figure5.5 Neural Networks Structure

算法描述为：

设 *m*[*i*]为第 *i* 层的神经元的个数；*Input* 为 BP 网络的输入；*KW*[*n*]为输入样本；*fact_Output* 为实际输出；*wish_Output* 为教师信号，即期望输出；*studyrate* 为学习率；*changerate* 为权系数修正常数；*w*[*i*][*j*][*k*]为第 *i* 层的第 *j* 个神经元的第 *k* 个输入的权系数；*F*[*i*][*j*]为第 *i* 层的第 *j* 个神经元的输出；*dlt*[*i*][*j*]为第 *i* 层的第 *j* 个神经元的权值改变；*w_change*[*t*]为第 *t* 个权系数的上一次的改变值；*jg_maxErr* 为结果最大误差；*jg_avgErr* 为结果平均误差；*qxs_avgErr* 为权系数平均误差；*qxs_minErr* 为权系数最小误差。

算法步骤为：

1．对权系数 *w*[*i*][*j*][*k*]置初值。

对各层的权系数 *w*(*i*,*j*,*k*)置一个小于 1 的随机数，并选择学习率 *studyrate* 为（0.1~0.4），权系数修正常数 *changerate* 赋值（0.7～0.9）；

2．　在训练样本集中取一个样本值 *X*，也就是训练样本的特征相权值 *X*=（*KW*[1],*KW*[2],*KW*[3],...,*KW*[*n*]）。求出对应的期望输出 *wish_Output*。

3．计算各层的输出。

对第 *i* 层第 *j* 个神经元的输出 X_i^k，有

```
for(i=1;i<=3;i++){                                    //第 i 层
  for(j=1;j<=m[i];j++){                               //第 j 单元
```

```
    u=0;
  for (k=1;k<=m[i-1];k++)                    //第 k 个输入
    {u=u+F[i-1][k]*w[i][j][k];}
    F[i][j]=1/(1+exp(-u));                   //输出
  }
}
```

神经网络实际输出为 *fact_Output=F*[3][1]；

4．求各层的学习误差 *dlt*[*i*][*j*]

对输出层有 *k=m*,有

dlt[5][1]=*fact_Output**(1-*fact_Output*)*(*fact_Output-wish_Output*);

对其它各层，有

```
for(i=4;i>=1;i--){                                   //第 i 层
   for(j=1;j<=m[i];j++){                             //第 j 单元
     add=0;
     for(k=1;k<=m[i+1];k++)
{
add=add+dlt[i+1][k]*w[i+1][k][j];                    //第 k 个输出
}
     dlt[i][j]=F[i][j]*(1-F[i][j])*add;
    }
 }
```

5．修正权系数

```
for(i=1;i<=5;i++){
   for(j=1; j<=m[i];j++){
     for(k=1;k<=m[i-1];k++){
       Err=-studyrate*dlt[i][j]*F[i-1][k]+changerate*w_change[t++];
       w[i][j][k]=w[i][j][k]+Err;
       w_change[t-1]=Err;
       qxs_avgErr+=Err;
       }
```

```
        }
    }
```

6.求出了各层各个权系数之后，可按误差判别是否满足要求。如果满足要求，则算法结束；如果未满足要求，则返回 2 执行。本程序的学习结束的控制条件是最大权系数的变化误差小于等于 0.005 且结果的误差小于等于 0.005 且学习次数要大于 3000 次以上。即（*times*>3000&&fabs(*qxs_avgErr*)<0.005&&fabs(*jg_Err*)<0.005 为真时，学习结束。在经过对由蜜罐收集的初始攻击信息的 3000 次学习训练后，最终形成精确的攻击预警模型并输出。

5.2 其他组件的设计

5.2.1 防火墙组件

防火墙连接控制是数据控制的第一层，允许所有进入蜜罐系统的网络连接，而对外出连接进行限制，通过设置允许最大连接数对外出连接进行控制，当达到最大数时自动阻断其后的所有连接。防火墙也是数据捕获的第一层，防火墙对所有进出蜜罐系统的数据包进行日志记录，并把日志信息存放在防火墙本地。

防火墙截获所有流经防火墙的 IP 包，从其 IP 头、以及传输层协议头中获取并过滤所需的相关信息，然后依次按顺序与事先设定的访问控制规则进行一一匹配比较，如果与某条规则相匹配，便执行其规定的动作(如:DENY, ACCEPT, REJECT 等)。为追使每一个 IP 包都能在访问规则集中找到与其匹配的规则，一般在访问规则中都要定义缺省策略以实现某种缺省动作。防火墙的主要功能是:

1.设定单向地址拦截或双向地址拦截，在单向地址拦截时，一方到另一方的资料访问被禁止，但反向的数据访问依然能正常进行，不会受到影响。

2.采用先进的状态监测数据包过滤技术，不仅仅是依靠单个的 IP 包来过滤，而是对每一个对话和连接进行分析和监控，在系统中自动维护其当前状态，根据连接的状态来对 IP 包进行高效快速安全过滤。

3.对蜜罐系统的外出连接进行控制。当外出连接达到一定数量时，阻断以后的连接，防止蜜罐系统被黑客攻破后用来作为发起攻击的“跳板”。

4.对所有出入蜜罐系统的连接进行日志记录。

5.2.2 IDS 组件

IDS 是蜜罐系统的必备功能，它不但可以实现及时报警加强管理员对蜜罐系统的监视程度，还可以从另一个角度对网络数据流进行分析和保存。IDS 对系统事件和网络上传输的数据进行实时监视和分析；IDS 抓取蜜罐系统内所有网络通讯 IP 包，并将其记录到数据库中，以便进一步分析统计。同时检测网络入侵行为，并进行声音报警[33]。

IDS 系统的输入数据是蜜罐系统内的所有数据流信息(数据链路层数据帧)。IDS 主要完成的工作有：

1. 输入数据的有效性检查；
2. 包的规则匹配；
3. 网络攻击包报警；

第六章　系统测试

蜜罐研究的最终目的是利用蜜罐了解黑客，这就要进行蜜罐实验。以下我只是初步实现了一个蜜罐系统，并对它的功能进行初步的测试，蜜罐实验也可以看作是对蜜罐系统的一个集成测试。

6.1 蜜罐实验网络环境

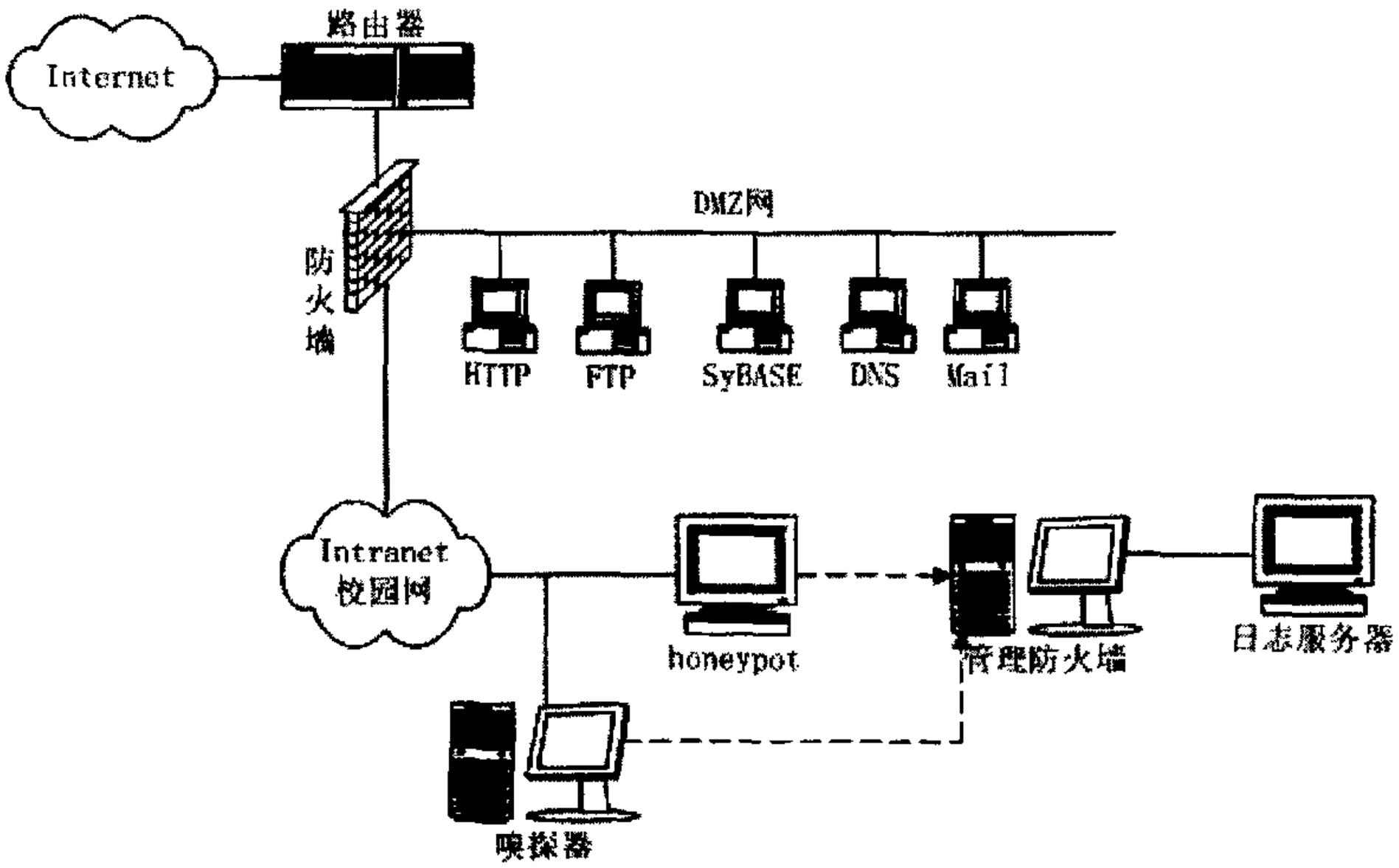

图 6.1 蜜罐实验网络环境

Figure6.1 Environment of the Experiment

为了方便采集信息，充分发挥蜜罐在安全防护中的主动防御功能，应将蜜罐放到因特网上黑客容易访问到的地方。在前面的章节中我提出的系统结构图中在在校园网不同位置部署不同类型的蜜罐,以获得不同目的的安全防护功能。限于客观条件，也为了实现方便，我在校园网内部东风路校区四号楼一台机器上部署蜜罐，操作系统基于 windows2000，在这种环境下对蜜罐作用进行实验。实验的网络环境如上图 6.1 所示。

6.2 蜜罐实验过程

我将蜜罐部署在内部网中，此时主要的优势是利用它来检测和阻止攻击者对校园网的进行攻击，并利用它捕获攻击者的所有攻击活动，借此分析攻击者使用的攻击动机、工具和技术。

在利用 Specter honeypot 实现对攻击活动进行检测时，由于 Specter 可以通过对指定端口的监听来检测和记录发往这些端口的任何连接（均被认为是可疑的探测或扫描）。并且 Specter 在检测到这些可疑的连接后有多种方式远程预警给管理员。在阻止攻击者方面，Specter 可以通过欺骗和威慑手段来达到目的。具体地讲可以通过将虚假信息反馈给攻击者或者分发假冒密码文件的能力；还可以把 Specter 的特征设置为 Strange，通过对攻击者进行古怪响应或者奇特动作来迷惑攻击者和浪费它们的时间，这时 Specter 会以一种非预期的方式行动。另外 Specter 可以模拟漏洞，假设攻击者启动一个 Specter 成功模拟的一个漏洞。攻击者会紧随此次攻击，此时 Honeypot 会模拟期待访问被模拟的服务，这样攻击者会被误认为攻入真实系统漏洞。

将蜜罐技术引入网络安全整体架构中不仅应该让它捕获尽可能多的信息，而且应该使用不同的方法来捕获该信息。由于在校园网内部可以部署多个 honeypot，因而需要一个集中的架构来聚合收集的所有信息，这可以通过建立一个蜜场来管理和记录网络来实现。

6.3 黑客攻击案例

我的蜜罐部署到校园网内部以后，多次被黑客探测和侵入。在这个过程中，我的蜜罐收集了攻击者的攻击信息，通过分析这些入侵信息来了解黑客入侵所采取的技术和工具。以下是我部署的蜜罐所捕获到的一次黑客攻击过程。

在校园网东风路校区，我采用 *Specter* 蜜罐，它的操作系统基于 *windows2000*。该蜜罐的主要目的是用于检测成功的攻击。我用作蜜罐的机器 IP 地址为 202.116.137.93，用作远程记录日志的机器 IP 为 202.116.137.92，假定攻击者 IP 为 202.116.137.95。

Specter 本身带有 7 个完全模拟的服务、6 个 trap 和一个可定制的 trap。这种

灵活性可以对 13 个预定义的端口和一个可配置的端口进行攻击检测，通过覆盖更大范围的端口，Specter 能够检测到更大数目的不同攻击。而且 Specter 能够对应用程序进行模拟，其中的服务模拟了实际的应用程序的交互，具有更大的真实性和交互性。更容易让攻击者相信。它们不仅可以模拟交互还可以模拟漏洞。黑客可以对 Specter 服务发起一次攻击并认为攻击已经成功，而实际上并非如此。然后 Specter 可以将伪造的信息反馈给攻击者。以下以 Specter 模拟的服务之一 FTP 为例，此时 Specter 充当一个易受攻击的 FTP 服务器，入侵者可以 FTP 到 honeypot 并执行大量的命令[33][34]。很多攻击者会使用的一种常见战术是下载被攻破系统的密码文件。该文件用于获取关于用户账户的更多信息，有时包括用户密码的信息。Specter 对这种漏洞进行了模拟，为攻击者提供了一份伪造的攻击者可能会用来进行分析甚至加以破解的密码文件。这种模拟漏洞与攻击者进行交互，为入侵者提供他们所期待的从易受攻击服务中获得的反馈。

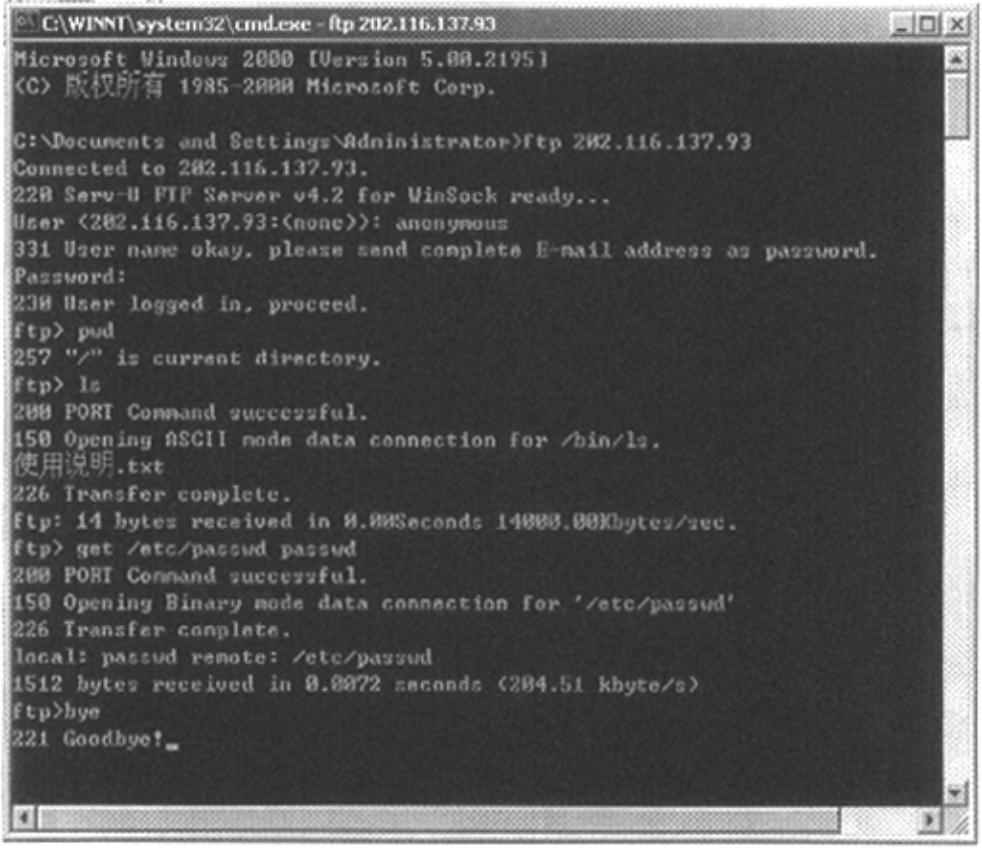

图 6.2 模拟一个易受攻击的 FTP 服务器的 Specter Honeypot

Figure6.2 FTP Server Simulated alike Specter Honeypot

当黑客攻击了模拟 FTP 的 Specter 蜜罐时，Specter 提供虚假服务，同时收集攻击者的所有活动，并向管理员发出预警信号。Specter 支持两种预警信号：Short Mail 和 Alert Mail。对 Short Mail 来讲，目的在于通过传呼或者蜂窝电话向个人发出通知，内容限制在少量字符上。优点是能够用于向那些配有传呼的管理员进行预警。局限性是仅能告诉管理员检测到了一次攻击。在对这样的预警作出反应前，管理员还必须获得更多的信息。下图为一次从 FTP 攻击处收到的 Alert Mail 通知。图 6.3 中没有给出攻击者的活动，而只知道发生了一次 FTP 连接。

```
Date :Sun, 10 Apr 2005 10:56:22 -0600
From: SPECTER ON gleaner <gleaner@gdut.edu.cn>
To: gzdusq@gdut.edu.cn
Subject: FTP connection (202.116.137.95 ) -Attempt 7/16 (FTP/16)

FTP connection from 202.116.137.95(FTP Attempts: 7,Total attempts: 16)
on Sun Apr 10 10:56:12 2005
```

图 6.3 为 FTP 生成的 Short Mail

Figure6.3 Short Mail Created for FTP

对 Alert Mail 来讲，它包含详细攻击信息的标准电子邮件。管理员通过电子邮件接收到关于试图破坏的关键信息。这种预警机制的优点是提供了大量的攻击信息。管理员可以有效地就预警中的信息作出反应。在这种预警机制中可以判定攻击者的目的。下图显示了 Alert Mail 预警信息。

```
Date :Sun, 10 Apr 2005 10:56:22 -0600
From: SPECTER ON gleaner <gleaner@gdut.edu.cn>
To: gzdusq@gdut.edu.cn
Subject: FTP connection (202.116.137.95 ) -Attempt 7/16 (FTP/16)

FTP connection
Host: 202.116.137.95
Login: anonymous
pass: hax0r
Time: Sun Apr 10  10:56:12 2005

Log:
Client connecting :202.116.137.95
Client tries anonymous Login
-->331 Guest login ok, send your complete e-mail address
as password 'hax0r'
-->230 User anonymous logged in.
Client asks about current directory.
-->257 "/" is current directory.
Client set port to 40985, IP to 202.116.137.95
-->200 PORT command successful.
Client asks for directory listing, sending fake listing.
-->150 Opening ASCII mode data connection for '/bin/ls'
Opened data connection to 202.116.137.95 on port 40985,
sent directory listing
-->226 Transfer complete.
Client set port to 40986, IP to 202.116.137.95
-->200 PORT command successful.
Client wants to transfer file /etc/passwd
-->150 Opening binary mode data connection for 'etc/passwd'
Sending passwd file with normal passwords
Transfer of file /etc/passwd to 202.116.137.95 on port 40986
complete.
-->226 Transfer complete.
Client closed connection.
-->221 Goodbye!
Closing connection with 202.116.137.95
```

图 6.4 为 FTP 攻击生成的 Alert Mail

Figure6.4 Alert Mail Created for FTP

一旦被告知有攻击发生，我们会希望查看下日志以获取更多信息。Specter 支持如下三种日志机制：Log Analyzer，Event Log 和 syslog。

Log Analyzer 本身带有内置的用于对探测和攻击进行日志记录和归档的数据库。该数据库为一组文本文件的集合，每次攻击都会被赋予记录每次攻击的文本

文件。Log Analyzer 为一种构建在 Specter 之中的 GUI，它允许对这些日志归档进行查询和查看。Log Analyzer 从主控制台启动。Log Analyzer 提供了一种查询机制，从而使得可以根据时间和来源对攻击者进行查询和检索。

Event Log 这种日志机制是大部分 windows 系统具有的日志机制，并且 Specter 活动可以被记录到日志。但这些日志也仅有少量信息，类似于 Short Mail。只能表明有活动在进行，但是没有更多信息。这部分信息被具体记录进了应用程序日志。优点在于如果有第三方的应用程序监视这些日志文件，这些应用程序也可以对 Specter 活动进行监视和回应了。

Syslog 代表了另一种对 Specter 所捕获的事故进行日志记录的方法。这是一种标准的 UNIX 实用工具，可用于在网络上从一个系统到另一个系统进行信息的日志记录。优点是可以从多个 honeypot 处收集载入日志的攻击并集中保存到一台远程的日志服务器中。日志服务器上的进程或者应用程序可以监视所记录的活动，并根据特定的签名或者预定的活动发送预警。但是要使用这项功能就必须有一个 syslog 服务器构架。根据该日志可以获得挖掘出更多信息所需的足够信息，但可能并不足以直接对攻击者进行回击。类似于 Short Mail。优点在于如果有第三方的应用程序在监视在监视这些日志文件，这些应用程序同时也能够对 Specter 活动进行监视和反应。

6.4 蜜罐实验结果综述

通过蜜罐实验，我对蜜罐、对黑客都有了更深一层的认识。实验过程中，在因特网上把 Windows 默认安装的 HTTP 服务打开，很快就会受到黑客攻击。由此可见因特网上黑客活动的泛滥程度。通过实验，还发现一些侵入蜜罐的黑客不仅有较高的技术水平，而且非常谨慎小心。要采集到更多与黑客相关的信息，需要进一步提高蜜罐的伪装程度、改进信息采集方法、探索适度的控制策略。

要使得 honeypot 有效运作起来，首先要先定义一套清晰的安全策略。即组织如何着手、实现以及执行安全措施，以降低对其环境所带来的风险。在我设计的这次 honeypot 部署中，对于内部网主要用于检测，因此其价值就是检测出那些未授权的活动，如扫描、探测或者攻击。Honeypot 的成功之处在于它检测出了这种探测的并向安全管理员进行了预警。

总之 honeypot 是一种可以应用于很多环境中的高度灵活的技术。作为一种安全工具，honeypot 只收集到少量的数据，但是这些数据中的大部分都是具有很高价值的信息。Honeypot 具备在资源紧张环境中有效工作的能力，通过检测和捕获未授权活动，honeypot 会迅速展示出其价值。

结 论

一、 技术优势与不足

系统的优势：

将蜜罐技术运用到网络安全防护架构中是一种更加有效而实用的方法，一方面它通过模拟有漏洞的系统或服务，提供易受攻击的目标，在一定程度上转移了入侵者对真实主机的攻击，让黑客在蜜罐机器中浪费时间，也在一定程度上保护了真实目标机器，另一方面通过让黑客与蜜罐机器交互，可以在蜜罐中收集到更多的关于入侵者信息，从而为入侵的取证提供重要的线索和信息。蜜罐可以是只模拟有限的服务或应用的低交互度蜜罐，或是一台默认的操作系统安装的有着明显安全漏洞的高交互度蜜罐。不管哪种类型都可以在不同程度上收集攻击者的攻击信息。通过对这些攻击信息的分析，可以暴露出系统存在的漏洞，并对其加以修正，从而起到提高系统安全性的目的。本次设计的系统有如下优点：

1. 通过构造有明显安全漏洞的蜜罐，并利用它来提供对入侵者易受攻击的目标并让入侵者对其进行攻击。在此过程中收集入侵者攻击的数据，这有助于对入侵者的入侵方法、入侵动机等的进一步研究，同时还可以通过对所收集数据的分析来找出我们系统中可能存在的安全漏洞，从而进一步完善我们的安全防护体系；

2. 因为在设计蜜罐时要求考虑到蜜罐自身的隐蔽性，即在理想情况下，入侵者不会发现自己正处于别人的监视之下，因而对由蜜罐所捕获得的数据不易被入侵者发现并篡改，从而使捕获的数据更加真实可靠；

系统的不足：

1. 蜜罐的跨平台性。在我的蜜罐系统的设计方案中，主要是对 windows 操作系统和应用服务进行模拟。但在现实中 Linux 是一种开放源代码的操作系统，所以对它的服务进行模拟，通过可加载内核模块的方式来实现系统功能。而对于目前使用的非常广泛的 Windows 操作系统，由于其内核源代码不公开，所以实现对 Windows 操作系统的模拟也非常困难，甚至是不可能实现的。但是由于 Windows 操作系统在目前使用的非常广泛，所以获取和分析入侵者入侵 Windows 系统的原

始信息是非常有必要的。同时，如果对于其它每种操作系统都重新设计蜜罐，那么工作量也将非常大。针对这一矛盾，我们可以考虑通过以下两种手段来对其进行处理。

2. 虚拟 honeynet

虚拟蜜网想通过在单一的物理系统上虚拟地运行多个蜜罐系统构成的蜜网，而目前常用的两种虚拟蜜网和 VMware 和 User Mode Linux[35]，前者是商用的虚拟蜜网，设计用于同时运行多个操作系统，后者只运行基于 Intel 操作系统上，目前只限于 Linux。

3. 蜜罐自身的安全性

在设计蜜罐的时候，已经考虑到蜜罐自身的安全性，例如通过进程的隐藏和保护、文件的隐藏和保护等手段对蜜罐的安全性进行了一定的保护，然而这些保护措施都是用来防止入侵者对蜜罐系统本身的关键部件进行破坏的。入侵者在入侵到一台主机以后，可能有另外一种选择，那就是利用已经攻陷的蜜罐作为跳板，来对其他主机进行进一步的破坏，比如将其作为发动 DDoS 攻击的一台傀儡主机。

为了增加蜜罐的可信度，尽可能地不让攻击者对蜜罐进行指纹识别，让入侵者能够建立适当的连接，不让其对蜜罐产生怀疑，又不至于对其他主机产生威胁，可以通过对蜜罐向外发出的连接进行控制的方法来加以解决。因为所有进出蜜罐的数据包都要经过防火墙，所以我们可以在防火墙上添加适当的规则，这些规则允许任何进入或外出的连接，但是对由蜜罐主机所外发的连接的数量进行限制。如果当前的连接数达到这个上限，那么任何超出的部分将被其所阻塞。这样就可以防止蜜罐对其它系统进行拒绝服务、端口扫描之类的常见攻击。这种方法可通过 CheckPoint 开发的 FireWall-1 实现，也可以利用 Linux 操作系统的开放源码的防火墙 iptables 来实现。FW-1 是一种可以运行于各种平台上的解决方案，该防火墙很容易管理，图形界面好，缺点是花费高，源码不开放。而 Linux 的 iptables 实现起来也比较容易，是一个开放源码的自由防火墙，如下用简单的命令行就可以实现这项功能：

```
iptables -I FORWARD -s $HONEYPOT -m limit -limit 3/minute
-limit-burst3\
-j LOG -log-level DEBUG -log-prefix "iptables FORWARD packet
died[39]:"
```

上面的命令说明对所有由 Honeypot 向外输出的连接进行控制，限定每分钟最多能够建立的连接数为 3 个，一旦超出限制，则对其进行记录。对于相同的记录，每分钟最多只能有 3 条，并且在每条记录的开始部分加上“iptables FORWARD packet died:”这样的前缀。当然，这里所允许的并发连接数和所允许的记录频度不是一成不变的，要经过不断的实验以后才能够确定。但用这种防火墙不足之处是仅限于 Linux 上使用。

二、 创新点

设计中，综合运用蜜罐、防火墙、入侵检测技术来构筑一个全新的网络安全防护体系。其创新点在于：

1、 防护的主动性。构造有特征的蜜罐，并通过这些特征吸引攻击者，对攻击者各种攻击行为分析，从而找到有效对付方法。如在本系统设计中为了吸引攻击者，通常在蜜罐系统上留下一些安全后门以吸引攻击者上钩，或者放置一些网络攻击者希望得到的敏感信息，当然这些信息都是虚假的信息。

2、 提前的预警机制与模型。提出一个基于神经网络 BP 算法来构造更精确的预警模型，从而在攻击发生前更有效地对攻击进行预警。

3、 较强的事后响应机制。由于收集的信息较长全面，因而能够通过对这些信息的分析，提出合理有效的应对攻击的方法。在校园网中部署蜜罐，一旦蜜罐被入侵，可以将蜜罐从网络中断开进行脱机工作，这样并不影响其他系统所提供的产品服务。蜜罐可以对入侵进行响应，它提供了一个具有低数据污染的系统和牺牲系统可以随时进行脱机工作。此时，系统管理员将可以对脱机的系统进行分析，并且把分析的结果和经验运用于以后的系统中。

三、发展趋势

本文从对传统网络安全防御手段的分析入手，针对传统防御措施中所存在的不足，提出了将 IDS、防火墙和蜜罐结合起来的新型主动防御结构。

作为传统的网络安全产品，IDS 和防火墙在防止黑客入侵这一方面可以说已经做的比较成熟，能够发现入侵行为并将大部分入侵行为阻止在受保护的网络之外。然而如果要对入侵者的入侵行为进行研究和一定程度的响应，那么用传统的网络安全产品实现起来就比较困难，因为这些安全产品一旦发现入侵行为，要么将其连接进行复位，要么直接将其拒绝在被保护网络之外，这样就无法对入侵者

后续入侵行为进行更进一步的研究。

蜜罐作为一种新型的主动的安全措施，与传统的安全防护措施不同，蜜罐被设计为一个具有明显安全漏洞的系统，以吸引大量的黑客对其进行攻击。通过收集这些入侵者的入侵信息，就可以对他们的入侵动机、入侵手段等进行进一步的分析，根据分析结果对攻击过程中所暴露出来的问题进行完善，从而提高系统的安全性。

蜜罐是网络安全领域的一种比较新的技术，目前国外已有不少机构正在对其进行研究和讨论，也已经有公司开始发布用于商业目的的蜜罐产品。在研究过程中，我们发现，自制的蜜罐能够提供更大的灵活性，但是在开发、配置与维护方面也需要更多的专业知识。同时，如果能将蜜罐同传统的安全防护措施紧密地结合起来，那么就可以使得整个系统的安全性能得到更进一步的加强。

蜜罐是一种可以用各种方式应用的高度灵活的的技术，如前所述它可以是一个模拟几项服务的简单的 Windows 系统，也可以是一个产品系统的网络。蜜罐的最初研究领域之一是早期预警和预测，因为它消除了错报和漏报。而且蜜罐不仅能检测出新的攻击趋势，还能潜在地捕获新的攻击工具。

只有配置和使用还不够，还必需维护蜜罐一旦环境变了，这个更有挑战性。不仅因为蜜罐技术的快速发展和提高，同时也由于所使用的网络环境的变化，常常加入新的系统而旧的系统被删除或者更新，新的应用程序的使用，旧的服务停止，你的网络经常发生变化。为了适应情况，要求部署的蜜罐必须能自动调节以适合这些变化。传统上需要人为的更新或修改蜜罐以更好的映射产品环境，而未来需要一种新的动态蜜罐技术，只要简单地插入它，蜜罐会为你做所有的工作。它会自动决定部署蜜罐的数目和方式，在你的环境中如何工作，甚至那些部署好的蜜罐可以变化以适应你的环境[36][37][38]。例如：你在网络中加入 LINUX 系统，你马上就可以得到 LINUX 蜜罐； 你去掉网络中的挪威网，相应的蜜罐也会很快消失； 你替换路由器，相应的蜜罐路由器也会变； 它可以自动训练并学习周围的网络环境，安排适当数量和蜜罐配置来适应你网络的变化。

动态蜜罐技术中最重要的就是它如何学习并适应周围的网络环境。环境中所有用的系统类型，如何用这些系统。动态蜜罐能够智能地映射系统和对网络环境作出响应。 一个可能的方法是动态探测你的环境，以决定什么系统是活动的，它们是哪种系统，它们所用的服务。

虽然蜜罐是一种比较新型的安全手段，但是我们也要认识到这种思想已经被许多人所了解，同时入侵者也在千方百计地探测蜜罐的存在，并企图对其进行破坏。因此我们在研究主动防御措施的同时也要关注入侵技术的最新动向，以在入侵与反入侵的对抗中占优势地位[40][41]。