西安电子科技大学

硕士学位论文

基于Snort的入侵防御系统的设计与实现

作者姓名 李梦钰

校内指导教师姓名、职称 覃桂敏 副教授

企业指导教师姓名、职称 武志强 高工

申请学位类别 工程硕士

学校代码 10701 学 号 1310122669
分 类 号 TP311.5 密 级 公开

西安电子科技大学

硕士学位论文

基于 Snort 的入侵防御系统的设计与实现

作者姓名：李梦钰

领　　域：软件工程

学位类别：工程硕士

校内指导教师姓名、职称：覃桂敏 副教授

企业指导教师姓名、职称：武志强 高工

学　　院：软件学院

提交日期：2015 年 12 月

Design and Implementation of Intrusion Prevention System Based on Snort

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Software Engineering

By

Li Mengyu

Supervisor: Qin Guimin Associate Professor

December 2015

摘要

随着网络技术的快速发展，网络环境日新月异，网络给大众所提供的服务越来越多，人们对于网络的依赖也越来越严重。与网络技术相对应，黑客攻击技术也在快速的发展。伴随网络普及的同时，用户被攻击事件也在不断地增长。不只是普通用户，越来越多的黑客把自己的攻击目标锁定为政府机构、社会团体甚至是各国军方的机密文件。于是网络安全的发展，被越来越多的人们所关注。各国政府也在这方面投入了大量的人力和资本，以求维护本国网络的安全，使本国的机密文件可以处于安全的环境中。

由于杭州华三通信技术有限公司（H3C）的主营业务是为各个高校和企业网络的搭建提供相应的设备，因此在现如今的网络环境下，公司对于其产品在网络安全方面的功能也提出了相应的要求，于是决定开发自己的入侵防御系统（Intrusion Prevention System ，IPS）。由于以前没有这方面的设计，所以主要将会面临以下三个问题：

1、已有的 IPS 产品与公司的产品不兼容，无法直接使用，因此需要进行移植；

2、已有的 IPS 大多数是应用层的，效率比较低，需要进行改进；

3、IPS 内部算法效率低，需要进行改进；

为了解决以上问题，本文通过对国内外 IPS 发展现状的了解，结合 H3C 公司产品的需求以及需要部署网络的环境等方面内容的研究，设计实现了一个基于 Snort 的 IPS。主要完成了以下三个方面：

1、通过对 IPS 发展过程的研究，选定了将要进行移植的系统——Snort 系统。这是一款入侵检测系统（Intrusion Detection System ，IDS），IDS 是 IPS 的基础，也是 IPS 的核心部分。Snort 系统是世界上最广泛使用的一种 IDS，具有高效、灵活和简洁的特点。作为开源系统，使用该系统进行移植，也可以降低开发成本。

2、为了提高 IPS 的效率，也为了满足公司产品的需求，对 Snort 系统进行了改进，从原来的用户层面修改到了内核层面，因此对 Snort 的一些模块进行了修改，并增加了新的模块。

3、对于内部算法效率低的问题，引入了多模式匹配算法，通过该算法可以只用一次比较，就得出某一字符串中是否包含另外几种字符串，大大的提高了对于报文进行比对的效率。

本文通过对基于 Snort 的 IPS 进行需求分析，了解到系统需要实现的各个功能点。然后针对各个功能点，结合 Snort 系统，对 IPS 进行模块划分和设计，并实现了各个模块。最后搭建环境对各个功能点进行了详细的测试，修改了测试中出现的错误，保证最后的系统可以实现需求中全部的功能。

关键词：入侵防御系统， 入侵检测系统， Snort 技术， 多模式匹配算法

ABSTRACT

With the rapid development of network technology, network environment is changing with each passing day. The network provides the public with an increasing number of services, thus people rely more on the network. As the correspondence of network technology, hacker technology is also experiencing a rapid development. However, with the expansion of network, number of users attacked is constantly growing. Not only ordinary users, more and more hackers are targeting government agencies, community groups and even national military confidential documents. So the development of network security draws more people's attention. Governments put a lot of manpower and capital in this in order to safeguard their network security so that their confidential files can be in a safe environment.

H3C, with the main business of providing the appropriate equipment for building network for various universities and enterprises, has also made the corresponding requirements for its products in terms of the function of the network security under the current network environment. Therefore, the company decided to develop its own Intrusion Prevention System (IPS). Without previous design, the main problems that will be faced with are as follows:

1. The existing IPS products are not compatible with the company's products and can not be used directly, so transplantation is needed;

2. Most of the existing IPS are working in application layer, so they have relatively low efficiency, so improvement is needed;

3. Internal algorithm of IPS has low efficiency, which needs to be improved;

In order to solve the problems above, with the understanding of development status of IPS both at home and abroad, the author designed a Snort-based IPS on the product demand of H3C. The design has mainly completed the following three aspects:

1. After researching the development of IPS, we selected Snort system to be transplanted.

This is an Intrusion Detection System (IDS), the foundation and a core part of the IPS. Snort is the most widely used IDS system in the world with characters of high efficiency, flexibility and simplicity. As an open source system, using the system for transplantation can also reduce development cost.

2. In order to improve the efficiency of IPS and meet the needs of the company's products, Snort system has been improved: modified from the original user level to the kernel level, therefore some modules have been modified, and new modules added.

3. As for the problem of low efficiency of internal algorithm, this paper introduced multi-pattern matching algorithm, which can judge if a string contains several other strings only compared once, greatly improving the efficiency of packets comparison.

Based on the demand analysis of Snort-based IPS, the thesis illustrates each function point the system needs to be achieved. Then for each function point, the author carried out module partition and design for IPS under Snort system, and programmed various modules. Finally, the author set up the environment for testing functionality, and modified the error in the test to ensure that the final system can achieve all the functions in demand.

Keywords: IPS, IDS, Snort technology, multi pattern matching algorithm

插图索引

表格索引

缩略语对照表

缩略语	英文全称	中文对照
WiFi	Wireless Fidelity	无线保真
DoS	Denial of Service	拒绝服务
VPN	Virtual Private Network	虚拟专用网络
IDS	Intrusion Detection System	入侵检测系统
IPS	Intrusion Prevention System	入侵防御系统
HIDS	Host Intrusion Detection System	基于主机的入侵检测系统
NIDS	Net Intrusion Detection System	基于网络的入侵检测系统

目录

第一章 绪论

现如今，随着支付宝、百度钱包、微信钱包等各种网上支付手段的普及，互联网已经成为人们生活的一部分。与此同时，免费 WiFi 的出现更加大大方便了人们的生活。然而，在人们的生活越来越离不开互联网的同时，网络安全也成为了人们越来越关注的问题。

为了让人们可以放心的使用网络，在一个安全的网络环境中生活，对入侵防御系统的研究就显得愈发的重要。

1.1 IPS 的研究背景及意义

现如今，处在一个通信技术和网络技术快速发展的时代，在我国大力发展信息化技术浪潮的推动下，我国的经济建设和人民的生活也在越来越依赖于通信网络。包括政府和企业在内的许多团体，都搭建有自己内部的网络系统，有些甚至直接与互联网相连。网络的出现，大大的方便了人们的工作与生活，因此随着互联网时代的来临，文化、经济、军事和社会的方方面面都将会对互联网产生强烈的依赖。

伴随着网络快速发展的同时，黑客攻击事件的发生频率也在不断增长，尤其是大规模蠕虫和 DoS 攻击，为网络环境的安全带来了大量的隐患。于是人们研制出了许多的网络安全产品，如：公钥基础设施[1]、鉴别与认证[2]、防火墙[3]、入侵检测系统（IDS）[4]、虚拟专用网络 VPN[5]、防病毒网关[6]、物理隔离卡[7]等产品。其中得到广泛应用的是防火墙和入侵检测系统，现在基本上每一台电脑都会有防火墙，每一款杀毒软件都有其自身的入侵检测系统，但是它们都有自身的不足之处[8]。根据 Vandyke Software（美国网络安全软件开发商）的调查显示，遭受到网络攻击的用户中，有 86%使用了防火墙，有 42%使用了 IDS[9]。由此可以看出，防火墙和 IDS，在现如今复杂的网络环境中，对于保护用户的安全，已经难以起到作用。研发一款更为强大的安全系统，实现对用户网络环境的保护，成为了当今社会，日益紧迫的事情。于是乎，入侵防御系统（IPS）应运而生。

由于在 H3C 的业务中，各个企业、政府和学校网络环境的搭建占有很大的比重，基于现在网络攻击事件频发的现状，公司提出了在产品中加入 IPS 的需求。分析了目前主流的几款 IPS 产品后，发现其中存在一些问题。首先大多数的 IPS 是基于应用层的，这样会影响报文转发的效率；其次对于 IPS 内部的检测算法效率低下，这样会影响检测每个报文所花费的时间，从而造成报文的延迟；还有很多 IPS 对于其规则的设计不够灵活，用户不能对其进行修改，这样当网络环境发生变化或者攻击手段发生变化的时候就不能及时的对系统进行配置，从而造成难以防御攻击的后果。因此公司决

定设计一款内核态运行的、可与公司产品兼容的、灵活简洁的 IPS。

1.2 IPS 的国内外研究现状

IPS，入侵防御系统，作为计算机的安全系统，是对防病毒软件和防火墙的补充，是 IDS 的扩展[10]。在各种主流的安全产品中，防病毒软件，是部署在个人终端中对电脑中的病毒进行查杀的；防火墙部署在内网与外网之间，通过禁止端口的形式，对攻击报文起到一定的防御作用，但是如果攻击隐藏在报文内部，就可以逃避防火墙的检测；而 IPS 是部署在防火墙之后，作为防火墙的补充存在的，其作用是在背后实时的保护网络和系统的安全，它通过不断地监控网络和系统的活动，在入侵行为即将发动时进行阻止。

术语“入侵防御系统（Intrusion Prevention System)”由 NetworkICE 的技术作者和顾问 Andrew Plato 发明[11]。并且，由该公司设计的世界上第一款 IPS 产品——BlackICE Guard，于 2000 年 9 月 18 日问世[12]。它首次实现了在线模式的 IPS，该系统可以直接对网络中的报文进行处理，检测出具有攻击性的报文，可以直接丢弃。

在国外 IPS 已经日趋成熟，作为防火墙和 IDS 的取代品，它有着显著的优势，并且已经被市场认可，在某些地方已经取代了防火墙和 IDS 的应用。随着 IPS 的不断发展，其在市场中所占的比重也在不断加大，受到了越来越多的青睐，总的来说，其应用厂商可以分为三类[13]。

第一类是入侵检测和入侵保护厂商[14]。他们的业务范围主要是，对于入侵检测和入侵保护技术的研究，代表的公司有 ISS 公司、McAfee 公司和绿盟公司等。

第二类是集成网关厂商[15]。他们主要是依靠在网关中加入 IPS 的功能，来达到保护网络安全的目的。代表公司有 NetScreen 公司、SonicWALL 公司、安氏公司、启明星辰和港湾等。

第三类是负载均衡厂商[16]。他们在深入了解网络传输协议的基础上，在其产品中加入了 IPS 模块，用于保证在网络中安全的传输。代表公司有 TopLayer 公司和 Radware 公司等。

与国外成熟的市场相比较所不同的是，国内正处于蓬勃发展的时期。首先由绿盟科技于 2005 年 9 月，生产出了国内首款具有自主知识产权的 IPS 产品——冰之眼网络入侵保护系统（ICEYE NIPS）[17]。紧接着，启明星辰、华为、网威、天融信等公司也相继推出了自己的 IPS 产品[18]。现在越来越多的公司都加入到了研发拥有自主产权 IPS 的行列中。因此，各公司越来越多优秀的技术人才加入了对于 IPS 的研究。相信在不久的将来，在一代又一代国人的努力下，国内的 IPS 市场也会逐步的成熟。

从整体来看，在未来的市场中，拥有自主知识产权的 IPS 是一个必然的趋势。而对于 IPS 来说，配置越来越灵活，检测越来越高效，可适应的环境越来越复杂，也是其发展的一个必然趋势。

对于各个公司的 IPS 产品进行分析和比较，得出了各个公司产品的特点和功能，如表 1.1 所示。

表 1.1 各公司 IPS 产品特点和功能的表

公司	产品特点	主要功能
Cisco	可以准确的判断、分类和阻断相应的攻击报文	具有多个接口，可以同时运行于混合模式和内部模式，可保护多个子网的安全
网威	部署灵活，可检测多种报文，具有多种响应模式，安全可靠	入侵检测、实时响应、策略管理、安全加固、报警查询、报表统计、管理功能、升级维护、系统监控
绿盟技术	高性能、实时的主动防御、准确的检测/防护、优异的产品性能、高安全性、高可靠性和易操作性	入侵防护、Web 安全、流量控制、上网管理
启明星辰	深层防御、精确阻断	防御网络蠕虫、间谍软件、溢出攻击、数据库攻击；具有防火墙、防病毒、上网行为管理、抗拒绝服务攻击、内容过滤日志审计等功能
IBM	可阻挡已知与未知的攻击行为	动态阻断；优异的侦测技术
天融信公司	网络适用性、攻击检测、病毒检测、应用识别、URL 过滤、流量管理、防火墙、高可用性、日志和报表、系统管理	抗 DOS/DDOS 攻击；蠕虫、后门、木马、漏洞、间谍软件、Web 攻击防御能力；流行 P2P/IM、热门游戏的过滤；针对不同的网络环境和安全需求，制定不同的防御规则和响应方式

1.3 论文主要研究内容

本论文是作者在 H3C 实习期间所完成的项目，该项目的目标是设计并实现一款可以与公司平台兼容、灵活、简洁、高效的 IPS 产品。作为研发团队的成员之一，主要完成了系统移植整体框架的设计、各个模块的设计与实现和核心模块中算法的改进等工作。主要完成了以下几个方面：

1、前期准备，搜集相关资料，研究 IPS 的背景和发展现状，结合公司平台架构，确定移植方案。

2、需求分析，通过绘制活动图对系统业务进行了描述；使用用例图具体描述了各个模块的功能点；对系统中用到的数据进行了分析；通过绘制数据流图描述了系统中数据的流向和处理过程。

3、设计实现，从对于整体和对于各模块的设计两方面详细叙述了系统的设计方案；各模块中通过绘制功能分解图和流程图明确了各个模块的功能和实现过程。

4、测试分析，搭建测试环境，从配置解析、报文解析和报文检测三方面对系统进行测试。列出测试中的测试用例，描述测试项、测试目的、测试步骤和预期结果。总结测试结果，得出对于系统的评价。

5、总结展望，对本次项目进行总结，并根据系统的不足之处，对系统今后的改进方案提出展望。

1.4 论文组织结构

本论文总共由六章构成，每一章节的内容如下所示：

第一章，绪论。在这一章介绍了 IPS 相关的背景以及其研究现状；通过对当今网络环境的介绍，说明了研究 IPS 的重要意义；结合自己在公司实习的经历，介绍了论文的研究内容和自己所做的工作。

第二章，基础理论与技术。在这一章介绍了与 IPS 相关的 IDS、Snort 技术和多模式匹配算法。对于 IDS，介绍了它的不同分类，系统组成部分以及它的不足之处；对于 Snort 技术，则介绍了它的系统架构和其检验规则；对于多模式匹配算法，通过举例介绍了其基础算法和扩展算法。

第三章，系统需求分析。在这一章介绍了对于系统进行需求分析的过程，首先对于系统的业务进行了陈述，绘制了活动图，清晰的描述了系统所要实现的总体功能。然后绘制用例图，分析系统整体和各模块所要实现的具体功能点。之后对系统中需要使用到的数据进行了分析，确定了数据的结构和数据项。最后使用数据流图，对系统进行了三层分解，描述了系统中数据的流向和处理过程。

第四章，系统的设计与实现。首先介绍了系统的整体架构和处理流程；然后介绍了各个模块的功能和处理流程，通过功能分解图，明确了各个模块需要完成的具体功能，并通过流程图对实现过程进行了详细描述。

第五章，系统测试及分析。首先介绍了系统的运行环境，由于是基于公司的平台，因此只能在 H3C 的设备上运行；然后介绍了本系统的测试用例和测试过程；最后针对测试的结果进行分析。

第六章，结束语。首先对论文的工作进行了总结，客观分析在项目中取得的收获和一些不足之处；最后对于该项目的日后研究方向进行了合理的规划。

第二章 相关理论与技术

本章节将对此次课题中所涉及的一些理论与技术进行介绍，主要有入侵检测系统、Snort 技术和多模式匹配算。这些理论是理解该课题的基础，可以帮助读者快速的读懂课题所研究的内容。

2.1 入侵检测技术

入侵检测技术，是检测和识别针对计算机系统和网络系统，或者更广泛意义上的信息系统的非法攻击，或者违反安全策略事件的过程[19]。它所分析的数据，来源于网络和计算机系统中，通过对数据进行分析，摘取出其中的异常数据，从而判断出异常事件和可疑攻击行为，进而针对各种不同的攻击行为，采取相应的措施，或者预警，或者丢弃报文等，以此来达到保护计算机系统和网络环境安全的目的。

2.1.1 入侵检测技术分类

从数据的来源看，入侵检测技术可以分为两类：基于主机的入侵检测技术和基于网络的入侵检测技术[20]。

基于主机的入侵检测系统（HIDS）出现在 1980 年左右，那时没有现在的大规模网络环境，并且仅有的小规模网络也是相对独立的[21]。在这种情况下，对计算机各种行为的审计记录进行检查是轻而易举的。而在没有网络的环境中，入侵行为在当时是非常少见的，攻击的手段也比较单一，因此仅仅对攻击进行事后分析，就可以抑制住进一步的攻击。

与以往的相同，现在的 HIDS 依然对审计记录进行检测，但是，与以往不同的是，主机自动的对各种行为的审计记录进行检测，并且可以及时的对相应的入侵行为进行准确的判断，进而作出正确的响应。

通常，HIDS 通过对主机的系统、事件和安全记录进行检测和分析，从而获取所需的数据源。例如，当计算机系统有新的事件发生时，HIDS 将其发生后的审计记录与攻击特征进行比较，看是否匹配，如果匹配，系统就会根据匹配的结果作出对应的响应。

在 HIDS 中，通常都会对重要的系统文件和可执行文件进行定期的校验检查，以便可以及时的发现入侵行为。此外，大多数 HIDS 产品都会对端口的活动进行监听，在敏感的端口被访问时向管理员报警。

基于网络的入侵检测系统（NIDS），通过对流通在网络上的各种报文进行捕获，来获取数据源，它将系统放置在网络中的关键位置处，通过实时的捕获和解析在网络

中流通的数据报文，来判断入侵行为的存在与否[22]。

NIDS 通常将关键位置的主机网卡设置为混乱模式，并通过它来对网络进行实时的检测，将捕获的报文进行分析，来辨别通过网络的通信业务是否存在入侵行为。

NIDS 用于辨别攻击的模块，是基于四种常用的技术实现的：模式、表达式或自己匹配、频率或穿越阀值、低级事件的相关性[23]。

NIDS 一旦检测到了有入侵行为发生，就会通过其响应模块发出警报，其发出警报的方式是多种多样的，可以根据不同的入侵行为发出相应的警报，并采取不同的处理方式。

从数据分析手段看，入侵检测技术可以分为两类：滥用入侵检测技术和异常入侵检测技术[24]。

滥用入侵检测技术，是建立在对以往各种已经发生的入侵行为进行分析或者对于各种系统缺陷的积累的基础之上的[25]。需要不断的维护一个存储入侵行为特征和系统缺陷特征的数据库。将收集到的数据与数据库中的特征进行比对，如果比对成功则判定为入侵行为。

滥用入侵检测技术的优点在于，它可以准确的找到入侵行为，很少对入侵行为进行虚报；由于匹配规则可以清晰的描述，利于对其进行维护。但是相应的，它的缺点在于，对于入侵行为的搜集十分繁琐，需要耗费大量的精力；并且其数据库具有局限性，因为入侵行为会根据不同的操作系统、软件平台和应用类型作出相应的调整，所以滥用入侵检测技术没有良好的可移植性。

异常入侵检测技术，是建立在如下假设基础上的，即任何一种入侵行为都能由于其偏离正常或者所期望的系统和用户的活动规律而被检测出来[26]。因此异常入侵检测技术，会对正常合法的活动进行大量的搜集，通过分析其数据得出相应的规律，然后将需要检测的活动与其进行比对，如果发现偏离了正常的活动，就判定为是入侵行为。

异常入侵检测技术的优点在于，它可以发现一些对系统漏洞进行发掘、试探的行为，并且对环境的要求低，利于移植。但是相应的，由于对于正常行为的建模不一定完全，并且系统的行为是不断变化的，所以需要不断地学习，建立新的模型。可是如果在学习的过程中遭到攻击，很可能会将这种攻击判定为正常行为进行学习，这样以后就检测不出这类攻击了。

2.1.2 入侵检测系统的组成部分

从功能逻辑上讲，入侵检测系统由探测器（Sensor）、分析器（Analyzer）和用户接口（User Interface）组成[27]。

1)探测器

探测器主要用于对数据的收集，它的输入包含任何可能成为入侵检测行为的线索数据，如系统的审计记录、安全文件和网络报文等，这些数据被探测器收集整理后，传送到分析器进行解析判断。

2）分析器

分析器，即检测引擎，它将探测器收集到的数据进行解析，提取出其中的关键字，跟自己的信息库进行比对，从而来判断入侵行为的存在与否。如果发现了入侵行为，它将根据入侵的种类发出一个对应的提示信号，如警告等。系统会根据提示作出相应的措施，如丢弃报文等。

3）用户接口

用户接口是用于用户和系统之间进行交互的，用户通过该接口接收系统发送的消息，并且通过该接口对系统发送命令，实现对于系统的控制。“控制台”、“管理器”或者“控制器”是用户接口的别称。

除了上述的三个必要组成部分外，一些入侵检测系统还会有一个称为“蜜罐”的部分。该部分是作为诱饵存在的，它具有明显的安全漏洞，来引诱黑客对其进行攻击，然后将这次入侵的信息完全记录下来，从而使入侵检测系统可以对这类入侵行为进行检测。

2.1.3 入侵检测系统的不足

随着网络的不断发展，高速网络的进一步普及，网络环境千变万化。如何在高速复杂的网络中实现对入侵行为的实时检测将是对入侵检测系统的一个考验。

现如今大规模分布式系统的技术已经成熟。因此，将来的入侵检测会面临更为复杂的系统环境。如何承载大规模的不同性质的网络上的复杂入侵行为，将是入侵检测技术的一个瓶颈。

由于入侵检测技术是在入侵行为发生之后才能做出相应的反应，所以如何提前发现入侵行为，避免不必要的损失，将是限制入侵检测技术的一个重要因素。

2.2 Snort 技术

Snort 系统是一个以开放源代码形式发行的网络入侵检测系统，由 Martin Roesch 编写，并由遍布世界各地的众多程序员共同维护和升级[28]。Snort 以 Libpcap 库函数为基础，具有良好的可移植性，可以在多种软硬件系统平台上运行。它具有非常清晰的系统架构，因此利于模块化设计。因为以上优点，它被广泛的运用在各种入侵检测系统中。

2.2.1 Snort 系统架构

Snort 系统架构有三个特点：性能高、简洁和灵活。它由数据包解析器、检测引擎和日志/报警三个子系统构成[29]。所有的子系统都是基于 Libpcap 库函数的，系统先进行各种配置、规则解析和数据结构的初始化，然后再对报文逐个进行分析和检测，而不是每来一个报文进行一次初始化。这样可以提高性能，减少处理时间。

Snort 主函数的流程是，首先对命令行进行解析，提取出关键的信息；初始化检测引擎和协议解析器。然后打开接口对报文进行捕获。之后对插件进行初始化，包括输出插件、检测插件、预处理插件等，主要就是将插件名称和对应的处理函数关联起来，以便后续的使用。接着初始化规则架构，将分散的规则搭建成一个二维链表结构，为以后的检测做准备。然后初始化快速匹配数据结构，以便高效的检测出入侵行为。最后将报文送到数据包解析器对报文进行处理。

在数据包解析器中，将会调用各种网络协议的解析函数，按照自下而上的顺序，逐层对报文进行解析，从数据链路层到传输层。为了使得解析可以快速的进行，只是将各个关键信息的指针保存下来，以便下一步检测引擎对其进行分析。

Snort 的检测引擎采取的是“插件”模式。它将各种检测功能做成一个个的插件，每个插件都有一个特定的关键字与其相对应，检测的时候直接调用插件进行检测。这么做的目的是，使检测更为的方便灵活，并且可以随意的添加新的插件，来完善检测。

Snort 的检测规则，最终将会储存在一个二维的链表结构中。该链表由链表头和链表选项两部分构成。在链表头中，定义了一类报文，它们有相同的源 IP 地址、目的 IP 地址、源端口号和目的端口号等信息。每一个链表头与其他链表头连接，并且在每个链表头下都挂接着多个不同的链表选项。检测的时候会将这类报文与选项中的内容进行匹配，如果匹配成功，就会触发相应的动作。

日志/警报子系统是可选择配置的，在运行系统的时候，可以使用命令行进行配置，打开则会生成日志或者产生警报，关闭则不会。如果打开日志功能，将会有两种格式的日志可供选择，可读格式和 Tcpdump 二进制格式。可读格式将会对数据包进行快速的分析，而 Tcpdump 二进制格式可以提高系统的性能。对于警报信息，有完整和快速两种模式。完整模式下，会将警报的全部信息和完整的传输层包头信息记录下来；而在快速模式下，会把包头信息进行压缩后记录。

2.2.2 Snort 规则

Snort 使用了一种简单但是灵活、高效的规则描述语言来对检测规则进行描述[30]。Snort 规则分为两部分：规则头和规则选项[31]。Snort 规则示意图如图 2.1 所示，规则头和规则选项用一对圆括号“（）”隔开，圆括号外的是规则头，相应的圆括号内部的就是规则选项，规则选项是可以没有的。

```
drop tcp any any -> any any (msg:”bad traffic”; content:”xxx”; sid:524; rev:8;)
```

图 2.1 Snort 规则示意图

1）规则头

规则头由规则动作、协议、IP 地址、端口号和方向操作符五部分构成。

规则动作是指匹配成功后所要执行的动作，共有三类：Alert，记录该报文，并依据警报方式发出信号；Log，记录该报文；Pass，忽略报文。

协议字段标明是针对哪种协议进行检测，如 TCP、UDP、ICMP 等。

IP 地址表明检测所针对的 IP 地址。可以使用 any 表示所有地址，也可以使用求反算子“！”来表示除了某类地址外的地址。

端口号有多种不同的指定方法。如 any 指定、求反操作指定和范围指定等。范围指定用范围操作符“：”来表示，如 1:1024 表示从 1 到 1024 范围内的所有端口号。

方向操作符“->”，左侧的是源 IP 地址和源端口号，右侧的是目的 IP 地址和目的端口号。双向操作符“<>”表明左右都可以作为源 IP 地址/源端口号或者目的 IP 地址/目的端口号。

2）规则选项

规则选项是 Snort 系统的核心部分，它的设计体现了易用性和灵活性。不同的规则选项之间用分号“；”间隔开。选项中的关键字与关键字所对应的内容之间使用冒号“：”分隔开。Snort 中主要规则选项关键字及其功能如表 2.1 所示。

表 2.1 Snort 规则选项表

关键字	功能
msg	在警报信号和数据包日志中显示一条消息
log	将数据包记录到用户指定名称的文件，而不是标准输出文件
ttl	测试 IP 数据包的 TTL 字段值
id	测试 IP 数据包分组的 ID 字段是否等于指定值
dsize	测试数据包的载荷段大小是否等于指定值
content	在数据包负载中搜索指定模式
offset	选项 content 的修饰符，设定模式搜索的起始偏移量
depth	选项 content 的修饰符，设定某一指定模式匹配尝试的最大搜索深度
nocase	在进行字符串模式匹配时，不区分大小写
seq	测试 TCP 序列号字段是否等于指定值
ack	测试 TCP 确认字段是否等于指定值
itype	测试 ICMP 类型字段是否等于指定值
icode	测试 ICMP 代码字段是否等于指定值
session	转储某一指定会话的应用层信息
icmp_id	测试 ICMP Echo ID 字段是否等于指定值
icmp_seq	测试 ICMP Echo 序列号是否等于指定值
ipoption	监控 IP 选项字段中特定代码的出现情况

2.3 多模式匹配算法

作为众多入侵检测技术中最为重要的方法之一，模式匹配算法是一种基于攻击特征的检测技术[32]。所谓模式匹配，就是提取出入侵行为的特征值，然后将其与对应的特征库进行比对，从而判断出入侵行为[33]。例如在入侵检测系统中，解析出从网络获取到的报文中的关键字，然后与入侵检测系统中的模式库进行比对，如果匹配成功，就可判定为入侵行为。

2.3.1 基本算法描述

在与特征关键字进行匹配时，算法的效率将会是影响深度检测的瓶颈之一，在此处算法拟采用 N-tree 结构来进行字符串的匹配，N-tree 的结构如图 2.2 所示。

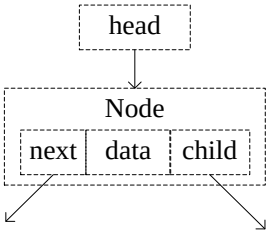

图 2.2 N-tree 的基本结构图

next 指向不同的字符，child 则是指向同一字符串的后续字符，例如要放入 w.a.c 和 w.b.c，其插入结果如图 2.3 所示。

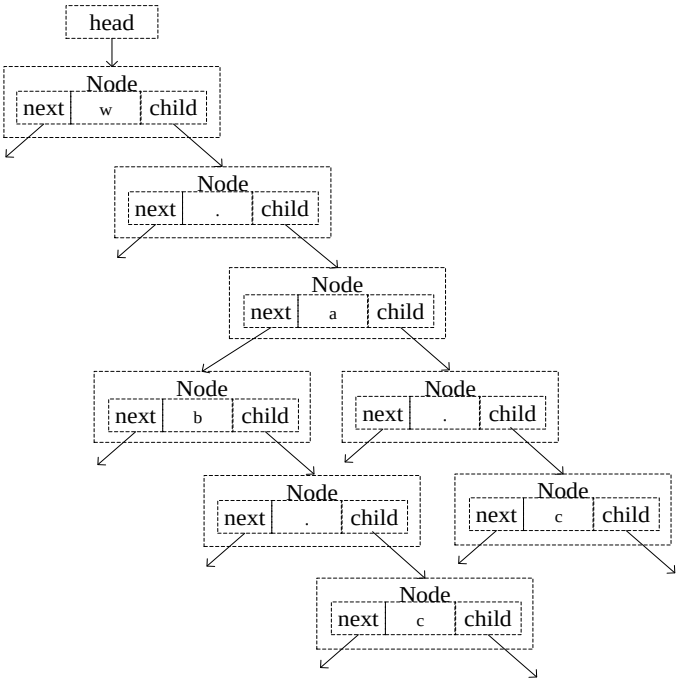

图 2.3 多模式匹配结构的插入结果示意图

第一个节点指向“w”字符，而他的 child 指向“.”这个节点，由于放入的前两个

字符串都有相同的“w.”，因此“w”节点和“.”节点 next 指向空，对于接下来的字符，由于不相同，并属于两个字符串，因此“.”节点的 child 指向“a”节点，而“a”节点的 next 指向“b”节点，如果查找一个字符串，首先检查第一个字符，如果字符匹配，接着只需要查找此节点的 child，如果不匹配，则对节点的 next 进行查找。

2.3.2 扩展算法描述

因为基本算法的搜索效率有限，于是扩展至有限状态机，提高搜索的效率。扩展后的算法在节点中添加了状态，可以记录各个节点的状态。扩展后的 N-tree 结构如图 2.4 所示。

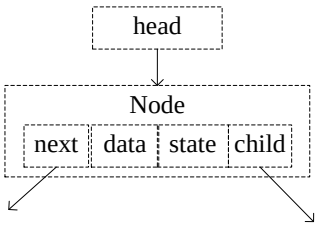

图 2.4 N-tree 扩展后的结构图

以 he、she、his 和 hers 为例，其添加结果如图 2.5 所示：

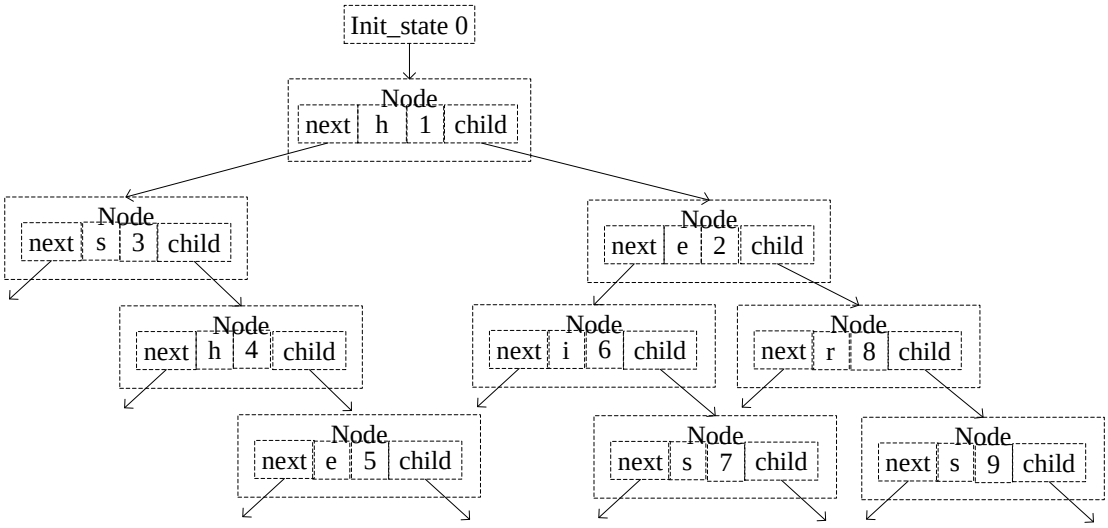

图 2.5 多模式匹配结构扩展后插入结果示意图

当得到一个字符的输入后，会有两种输出，输出一个状态或者输出 fail。例如，当状态为 0 时，输入一个字母 h，对应状态表可以得到状态 1，因此下一步就将自动机转换到状态 1。然而，如果输入的字母是 a，查找状态表得不到相应的状态，就会得到返回结果为 fail。如表 2.2 所示，每一个在自动机中的字母，都会对应一个状态，并且会对其进行相应的处理。当输入的字符没有匹配成功时，会发生状态转换，它的

作用是在不产生回溯的情况下匹配有相同子串的两个字符串，这样可以提高效率。

表 2.2 多模式匹配状态转换表

state	1	2	3	4	5	6	7	8	9
f(state)	0	0	0	1	2	0	3	0	3

为了能够清楚直观的了解其工作原理，用“ushers”作为例子，来进行具体的描述。

初始化自动机状态为 0；

输入字符“u”，遍历节点“h”和“s”，没有符合的，返回 fail；

输入字符“s”，遍历节点“h”和“s”，发现节点“s”，设置状态为 3，此状态的输出为空；

输入字符“h”，遍历节点“s”的子节点 child，发现节点“h”，设置状态为 4，此状态的输出为空；

输入字符“e”，遍历节点“h”的子节点 child，发现节点“e”，设置状态为 5，根据多模式匹配输出表 2.3，此状态的输出为 output(5) = {she，he}，根据多模式匹配状态转换表 2.2，置状态为 f(5) = 2；

表 2.3 多模式匹配输出表

state	2	5	7	9
output(state)	{he}	{she, he}	{his}	{hers}

输入字符“r”，遍历状态为 2 的节点“e”的子节点，发现节点“r”，设置状态为 8，根据多模式匹配输出表 2.3，此状态的输出为空；

输入字符“s”，遍历节点“r”的子节点 child，发现节点“s”，设置状态为 9，根据多模式匹配输出表 2.3，此状态的输出为 output(9) = {hers}。

在具体的实现中，如果需要更好的节省时间，提高效率，当函数 output 输出不为空，就可以认为有关键字被匹配，不需要再进行搜索了。

为了便于理解，将图 2.5 转换了一下变成图 2.6（实际这两个图是一样的），状态转换表是根据树的深度依次计算的，首先计算深度为 1 的“h”和“s”，凡是输入字符不为“h”或“s”的，均会返回状态为 0，并输入下一个字符。因此 f(1) = 0，f(3) = 0。接下来计算深度为 2 的“e”，“i”和“h”，要想知道状态转换，必须先知道上一级的状态转换，节点“e”的上一级为“h”，f(1) = 0，在 0 状态下无字符“e”，因此字符“e”的状态转换 f(2) = 0，同理字符“i”的状态转换 f(6) = 0。由于字符“h”的上一级状态转换 f(3) = 0，在 0 状态下有字符“h”，并且状态为 1，因此 f(4) = 1。然后计算深度为 3 的“r”，“s”和“e”，节点“r”上一级的状态转换 f(2) = 0，搜索状态 0 没有字符“r”，因此节点“r”的状态转换 f(8) = 0。节点“s”上一级的状态转换 f(6) = 0，搜索状态 0 得到字符“s”，并且状态为 3，因而节点“s”的状态转换 f(7) = 3。节点“e” 上一级的状态转换 f(4) = 1，搜索状态 1 得到字符“e”，并且状态为 2，因而节点“e”的状态转

换 f(5) = 2。最后计算深度为 4 的节点“s” 它的上一级的状态转换 f(8) = 0，搜索状态 0 得到字符“s”，因此节点“s”的状态转换 f(9) = 3。

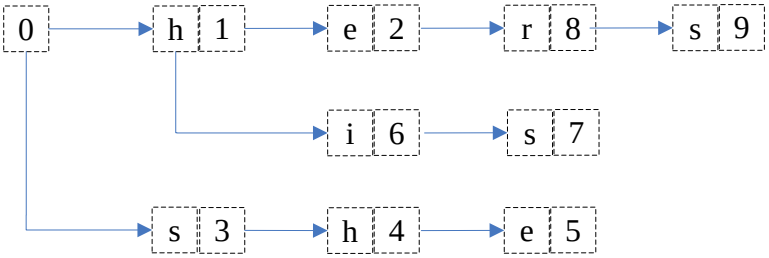

图 2.6 多模式匹配结构扩展后树的结构图

2.4 本章小结

本章节介绍了入侵防御系统的一些基础知识，有入侵检测系统、Snort 技术和模式匹配算法。

入侵检测系统，是入侵防御系统的前身。按数据源，可以分为基于主机的入侵防御系统和基于网络的入侵防御系统；按数据分析方法，可以分为滥用入侵检测系统和异常入侵检测系统。入侵检测系统主要由探测器、分析器和用户接口三方面构成。探测器用于收集待检测的信息；分析器用于解析检测收集到的信息；用户接口用于将检测的结果输出。入侵检测系统也有其不足之处，如何适应高速网络、如何适应复杂的系统和如何提早预警，都是入侵检测系统有待提升的方面。

Snort 系统，是一个入侵检测系统的实例，它以自身简洁、灵活和高性能的优势，成为了全球最受欢迎，应用最广泛的入侵检测系统。它由包解析器、检测引擎和日志/报警三个子系统构成。包解析器主要负责解析捕获的报文，取出其中的关键字；检测引擎主要负责针对解析后的关键字进行匹配，并得出报文是否为入侵报文的结论；日志/报警主要负责将检测的结果进行反馈，记录或者报警等。

多模式匹配算法，是用于优化检测过程的。它首先要对一组字符串搭建模式匹配结构树，树的左孩子是与其不在同一个字符串上的字符，右孩子是其所在字符串的下一个字符；然后对字符串进行匹配，这样只用经过一次匹配，就可以判断出一组字符串中是否有匹配成功的字符串了。

第三章 入侵防御系统需求分析

需求分析是任何研发项目都必不可少的一步，本章节通过业务陈述、需求建模、数据分析和过程建模四部分对需求进行了详细的分析和介绍。本章节的目的在于，通过本章节的分析，明确系统的各个功能点，明确系统中输入数据和输出数据的类型，明确系统中数据的流向。

3.1 入侵防御系统业务陈述

本系统是在公司产品平台 Comware7 上研究开发的，主要是为了研究一款基于内核态的 IPS，在保证防御攻击的基础上提高 IPS 的效率。通过使用本系统，可以达到高效的防御网络中大多数情况下攻击的目的。

通过分析可以看出，本系统主要完成了两个业务过程：第一，实现用户对系统的配置；第二，实现报文的检测。

用户对于系统的配置，首先用户对系统进行配置；配置信息需要进行解析才能让系统明白用户所要进行的配置；解析好的信息要进行处理才能被后续的操作所使用；使用处理好的配置信息搭建相应的特征树为检测报文服务，完成配置。如图 3.1 所示。

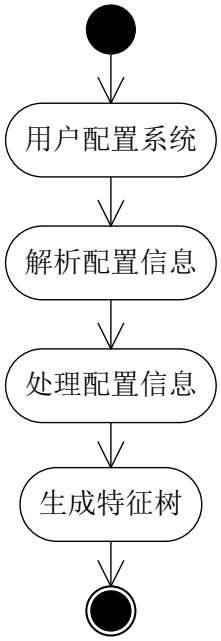

图 3.1 用户配置系统活动图

对于报文的检测，首先防火墙将按 IP 地址进行分类的报文送到系统；系统要对报文进行解析，取出其中需要的部分；将解析的结果进行处理，生成可以检测识别的信息；然后对报文的信息进行检测；最后输出检测的结果。完成报文的检测。如图 3.2 所示：

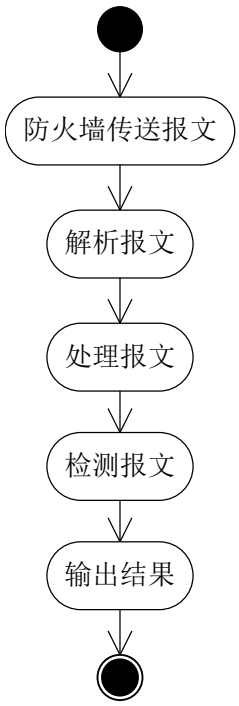

图 3.2 报文检测活动图

3.2 入侵防御系统需求建模

通过对于系统业流程的分析，可以看到，与本系统进行交互的有两个外部实体，用户和防火墙。用户可以对 IPS 系统进行管理，打开、关闭系统和配置信息等操作；也可以查看系统输出的检测结果，通过日志的方式反馈给用户。防火墙向系统提供将要进行检测的报文，如图 3.3 所示。

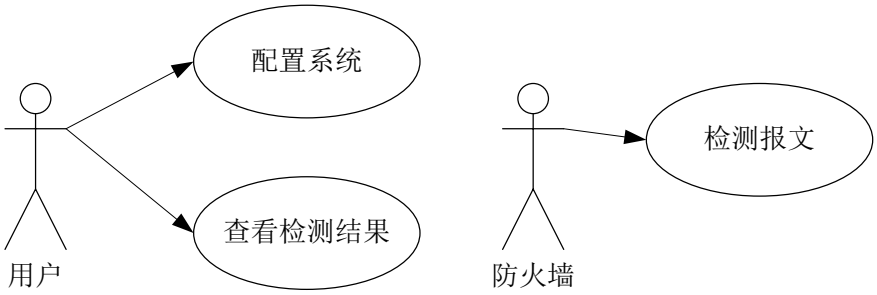

图 3.3 IPS 系统用例图

用户打开 IPS 系统，确定系统启动后，进入系统的配置页面，根据自己不同的需求，按照规定的格式对系统进行配置，系统会将配置后的结果反馈给用户，配置错误会返回相应的错误信息，配置成功就没有显示的信息。对于配置系统用例的描述，如表 3.1 所示。

配置好系统后，系统会对接收到的报文进行检测，然后将检测的结果以日志的方式反馈给用户，用户通过查看日志来查看检测结果。对于查看检测结果用例的描述，

如表 3.2 所示。

配置好系统后，防火墙会上送报文，系统首先对报文进行解析，提取出其中的关键信息，然后对报文进行处理，如分片重组等，之后根据由配置信息搭建成的特征树，对报文进行检测，最后将结果反馈给用户。对于检测报文用例的描述，如表 3.3 所示。

表 3.1 配置系统用例描述表

用例名称	配置系统
用例说明	本用例描述了用户对系统进行配置的过程
前置条件	用户打开 IPS 系统
基本事件流	1、用户启动 IPS 系统，进入配置页面。 2、用户根据需求，使用不同的命令对系统进行配置。 3、系统收到用户的配置命令，判断命令的类型。 4、根据不同的命令类型，调用不同的解析函数对命令进行解析。 5、将解析结果存入相应的数据结构中，完成配置。如果中间出现错误，给用户反馈错误的信息；如果配置成功，则不显示信息。

表 3.2 查看检测结果用例描述表

用例名称	查看检测结果
用例说明	本用例描述了用户查看系统检测报文结果的过程
前置条件	用户打开 IPS 系统；对系统进行配置；防火墙上送报文； 系统完成对报文的检测；
基本事件流	1、用户启动 IPS，并对其进行配置。 2、防火墙上送接收到的报文。 3、系统对报文进行检测。 4、系统将检测结果加入输出事件列表。 5、遍历输出列表，去除重复输出。 6、将结果进行输出，以日志的方式反馈给用户查看

表 3.3 检测报文用例描述表

用例名称	检测报文
用例说明	本用例描述了防火墙上传报文，系统对报文进行检测的过程
前置条件	用户打开 IPS 系统；对系统进行配置；防火墙上送报文；
基本事件流	1、用户启动 IPS，并对其进行配置。 2、防火墙上送接收到的报文。 3、系统对报文进行解析，提取出 IP 地址、端口号、协议等重要信息。 4、将解析结果与系统特征库进行比对。 5、如果匹配成功，则判定为攻击报文，根据用户的配置作出相应的处理；如果匹配不成功则判定为安全报文，不作处理。 6、将攻击报文的检测结果进行输出，反馈给用户知道。

3.3 入侵防御系统数据分析

通过以上的分析，可以看出本系统需要处理的数据分为两类，输入数据和输出数

据，下面将分别对其进行介绍。

3.3.1 输入数据

对于系统来说，与系统进行交互的共有两个外部实体，用户和防火墙，因此将会产生两种输入数据，用户输入的配置信息和防火墙输入的报文信息。

1）配置信息

根据约定，用户输入的配置信息需要按照规定好的格式进行输入，主要分为以下几类：

（1）文件路径定义，以 include 为关键字的信息，在关键字之后，定义的是文件名称和文件的路径，将文件名称和路径对应起来，主要用于批量的将特征文件下发至系统。

（2）IP 地址定义，以 var 为关键字的信息，在关键字之后，定义的是 IP 地址的代称，即就是为某个或者某一类 IP 地址起了一个名字，方便对于这类 IP 地址的使用。如 var home_net 1.0.0.1，在以后的配置中就可以使用 home_net 来取代 IP 地址 1.0.0.1。

（3）端口定义，以 portvar 为关键字的信息，在关键字之后，定义的是端口的代称，即就是为某个或者某一类端口起了一个名字，方便对于这类端口的使用。如 portvar http_port 80，在以后的配置中就可以使用 http_port 来取代端口号 80。

（4）预处理定义，以 preprocessor 为关键字，在关键字之后，定义的是插件名称和插件的配置信息，用于配置系统将会使用的插件，如分片重组插件等。

（5）约束门限定义，以 threshold 为关键字，在关键字之后，定义的是对于某类报文进行限制的信息，如受限的 IP 地址、受限数据的流向、受限的时间等信息，主要用于在某个时间段内，对某一类报文进行限制。

（6）约束禁止定义，以 suppress 为关键字，在关键字之后，定义的是禁止某类报文通过的信息。如受限的 IP 地址、受限数据的流向、受限的优先级等信息，主要用于禁止某类报文的通过。

（7）特征定义，无关键字，用于定义检测规则，分为特征头和特征选项两部分，与 Snort 规则相对应。

（8）特征关联定义，以 reference 为关键字，在关键字之后，定义的是特征库名称和特征库索引信息，用于将特征库的信息和系统相关联，增强系统的检测能力。

（9）特征分类定义，以 classification 为关键字，在关键字之后，定义的是特征分类信息，如特征名、特征 ID、特征分类 ID 等，主要用于输出，这些信息将会反馈给用户。

2）报文信息

防火墙会将接收到的报文传递到 IPS 系统，进行进一步的分析。因此系统接收到

的报文就是实际的 TCP/IP 协议报文，系统将会对四种报文进行处理：

（1）TCP 报文，IP 层协议号为 6 的报文。TCP 协议头存储着端口号、序号、确认号等信息。

（2）UDP 报文，IP 层协议号为 17 的报文。UDP 协议头存储着端口号等信息。

（3）ICMP 报文，IP 层协议号为 1 的报文。ICMP 协议头存储着类型等信息。

（4）IP 报文，下层协议不是 TCP、UDP 或 ICMP 的其他报文。

3.3.2 输出数据

输出数据，指的是系统根据用户的配置信息，对报文进行检测的结果。其中包含，特征分类名称、特征分类优先级、源 IP 地址、目的 IP 地址、源端口号、目的端口号，协议类型等信息。根据不同的协议会有不同的显示信息。如 TCP 报文将会显示序号、确认序号等信息。

如果报文为安全报文，系统会将报文输出到转发模块，进行后续的转发处理。

3.4 入侵防御系统过程建模

通过前面的分析，已经了解了系统所要完成的功能和需要得到的一些数据，下面将介绍如何得到这些数据，即就是系统的处理过程。

3.4.1 系统顶层数据流图

通过以上分析可得，本系统有三个外部实体，如图 3.4 所示。

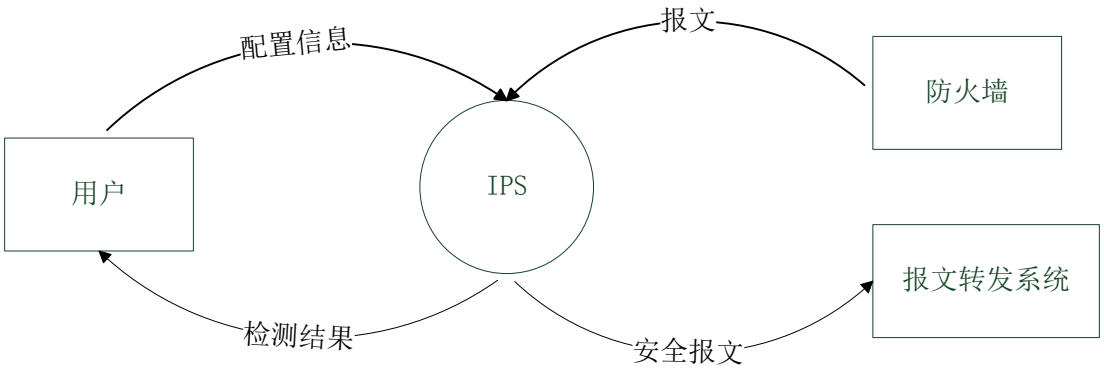

图 3.4 IPS 系统顶层数据流图

用户对系统进行配置，防火墙将报文送入系统，系统根据用户的配置对报文进行检测，如果检测后判定报文是攻击报文，系统根据不同的攻击类型进行不同的处理，如警告、记录或丢弃等，然后将检测的结果反馈给用户；如果检测后判定报文属于安全报文，则会送到转发模块进行转发。

3.4.2 系统 1 层数据流图

将系统功能展开，绘制出系统的 1 层数据流图，如图 3.5 所示。

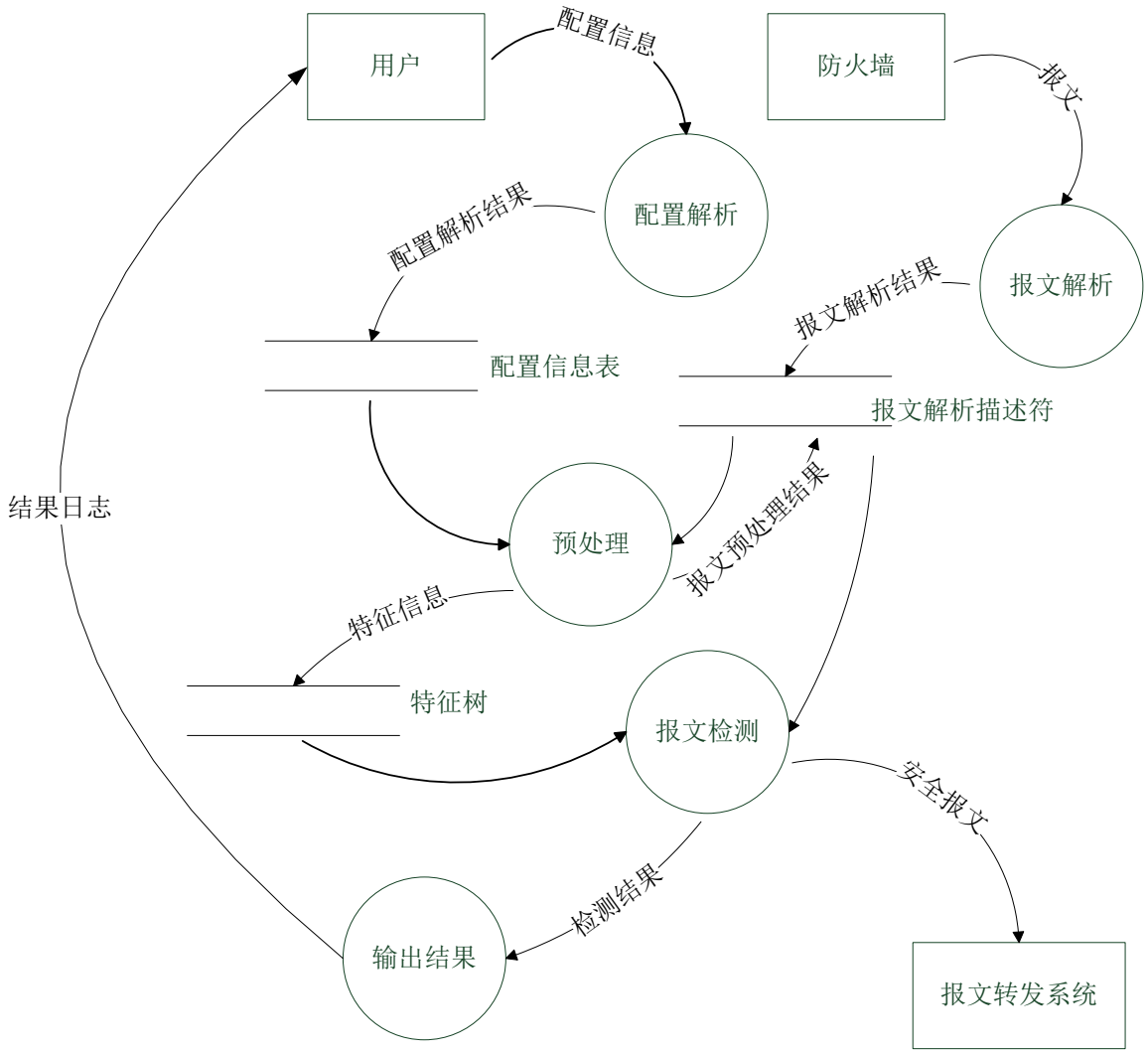

图 3.5 IPS 系统 1 层数据流图

用户下发配置信息，系统将配置信息进行解析，然后将解析好的信息送到预处理模块；预处理模块对配置信息进行处理，调用各种插件进行处理，并且将特征信息存储成特征树的形式，传递给检测模块，进行后续的操作。

防火墙将报文送达系统，系统对报文进行解析，按协议对报文进行分类，然后送到预处理模块进行处理；预处理模块会根据报文的类型调用不同的处理插件对报文进行处理，最后将报文信息储存成检测模块可以使用的形式传递给检测模块进行后续处理。

检测模块将特征信息和报文信息进行比对，将最终的结果送到输出模块；输出模块汇总检测结果，将重复的检测结果删除后，将结果反馈给用户，将判定为安全的报文送到转发模块进行后续处理。

3.4.3 配置解析 2 层数据流图

展开配置解析加工过程，绘制配置解析 2 层数据流图，如图 3.6 所示。

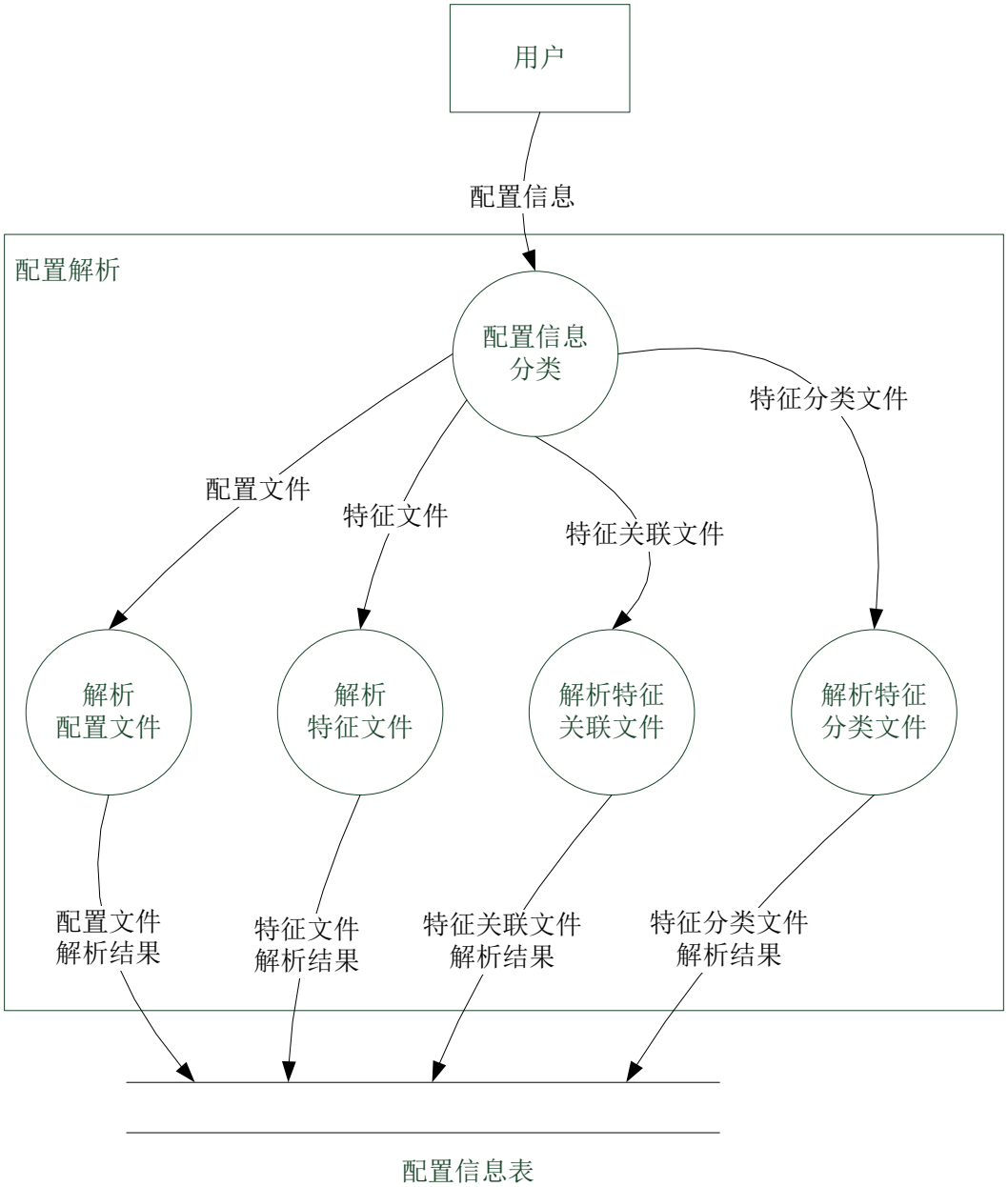

图 3.6 配置解析 2 层数据流图

配置解析模块，将对四类文件进行解析：系统配置文件、特征文件、特征关联文件、特征分类文件。

系统配置文件中包含了 IP 变量的定义、端口变量的定义、预处理定义、文件路径的定义、约束门限和约束限制的定义。配置解析模块将解析好的信息送到预处理模块，预处理模块会根据信息对相应的内容进行处理。

特征文件中包含了各个特征的规则信息，这是检测的基础信息。解析好的信息将被送到预处理模块进行进一步加工。

特征关联文件中包含漏洞库的名称和相应的 ID，用于将漏洞库与系统进行关联，

强化系统的检测能力。解析好的信息将会存入相应的数据结构中，以便后续的使用。

特征分类文件中包含特征类型、特征 ID、特征名称、特征优先级等信息。这些信息主要是用于输出的。解析好的信息会存到相应的数据结构中。

3.4.4 预处理 2 层数据流图

展开预处理加工过程，绘制预处理 2 层数据流图，如图 3.7 所示。

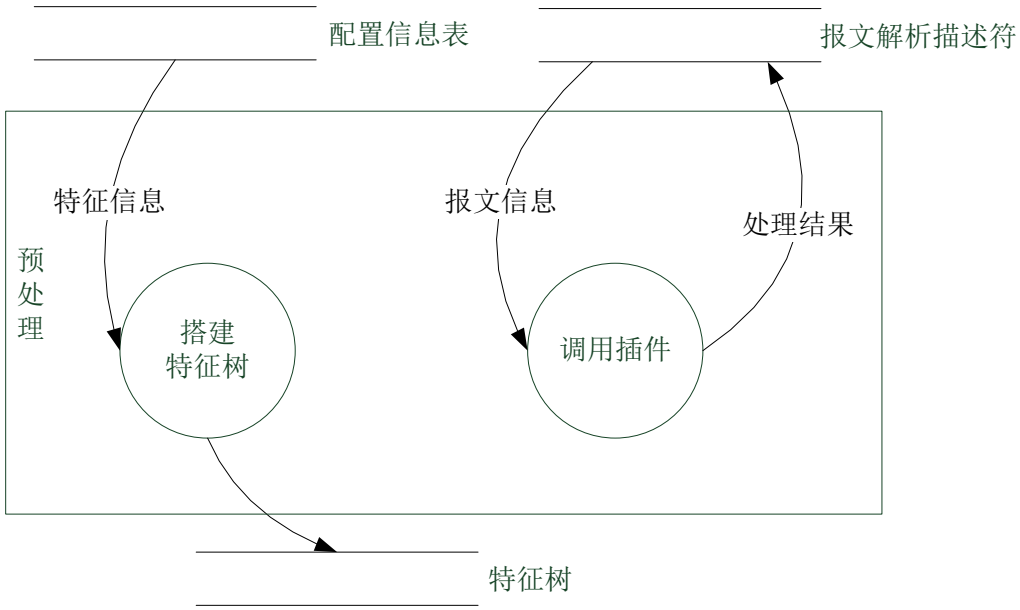

图 3.7 预处理 2 层数据流图

预处理主要分为两个过程，对特征信息的处理和对报文信息的处理。

对特征的信息处理，主要是利用特征信息搭建特征树，这是检测模块的基础，因此搭建好的特征树将送到检测模块进行下一步操作。

对报文信息的处理，主要是根据报文类型，调用不同的插件进行处理，例如处理分片报文的插件等。处理好的信息将送到检测模块进行下一步操作。

3.4.5 报文检测 2 层数据流图

展开报文检测加工过程，绘制报文检测 2 层数据流图，如图 3.8 所示。

报文检测主要分为两个过程，快速搜索引擎的搭建和对报文进行检测。

检测模块会将特征树中的特征信息根据协议进行分类，分别搭建协议为 IP 类型、协议为 TCP 类型、协议为 UDP 类型和协议为 ICMP 类型的快速搜索引擎。搭建引擎使用的是多模式匹配算法。

检测模块将接收到的报文信息根据协议分别连接到相应的快速搜索引擎中进行检测。并将检查结果发送到输出模块，进行输出反馈。

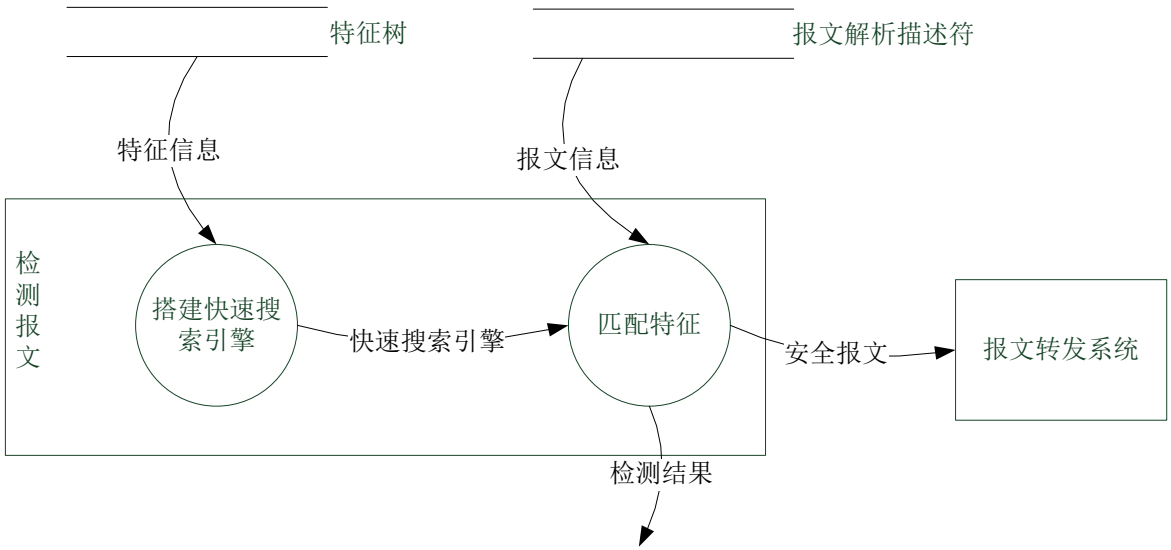

图 3.8 报文检测 2 层数据流图

3.5 本章小结

本章节介绍了系统需求分析方面的内容，首先对系统所要完成的业务进行了陈述，通过活动图介绍了系统所要完成的工作。然后对系统进行了用例建模分析，通过绘制系统总体用例图、配置解析模块用例图、报文解析模块用例图、预处理模块用例图、检测模块用例图和输出模块用例图，对系统总体和各个模块的功能点进行了介绍。然后通过对系统中数据的分析，明确了各个模块的任务。最后对系统的过程进行建模分析，通过绘制数据流图，自顶向下的对系统的处理过程和中间数据的流向进行了分析。

第四章 入侵防御系统设计与实现

本章节对系统的设计进行了详细的介绍，首先介绍系统的总体框架和处理流程，然后分模块介绍了各个模块的功能和详细设计思路。

4.1 入侵防御系统架构

本系统是在 Snort 系统的基础上设计与实现的，但是由于 Snort 系统是一款 IDS 产品，因此在设计的时候，对其架构进行修改，并对其中的模块进行删除和修改。

4.1.1 Snort 移植过程

从软件系统的角度来看，软件系统主要与用户和硬件系统有着密切关系。用户通过第三方软件、命令行或者 Web 对这个系统进行配置、管理和监控；硬件系统主要为系统提供硬件支撑，硬件系统与软件系统之间既有控制流也有数据流。

本系统是在 Snort 的架构基础上进行的开发，但是由于本系统是基于 Comware7 平台的，所以与传统的 Snort 架构有所不同。图 4.1 是传统 Snort 的架构图。

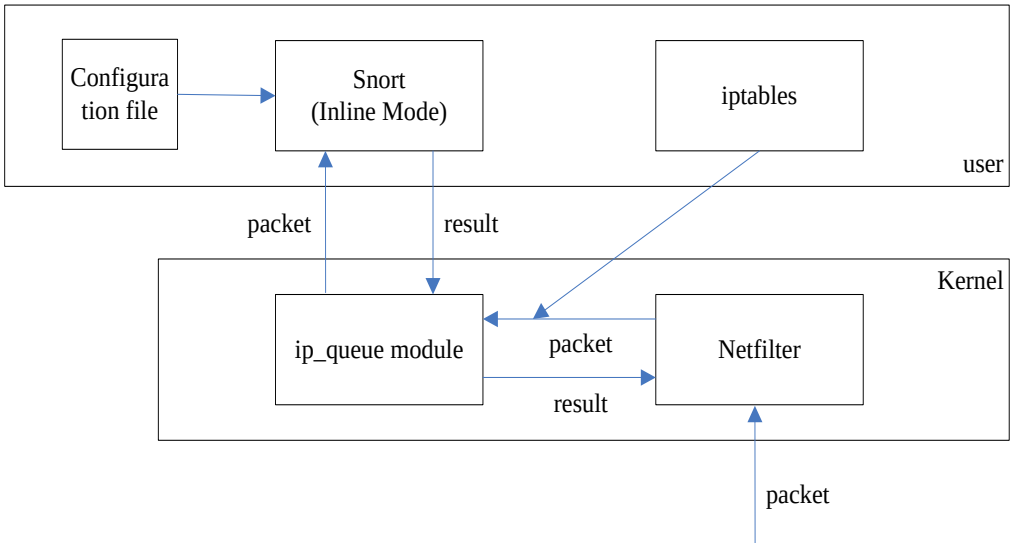

图 4.1 传统 Snort 的架构图

传统的 Snort 系统实现在用户态，报文通过防火墙传递到报文队列，然后再送到 Snort 中进行处理。Snort 的配置信息通过配置文件的方式传递到 Snort 中。该系统多数实现在应用层，报文经过多层传递，因此效率较低；通过配置文件的方式对 Snort 进行配置，不够灵活，更新不便利，也会导致效率低下。

因此，为了提升处理的能力和效率，设计了在 Comware7 平台下、内核态中的 IPS。图 4.2 是 Comware7 平台下 Snort 的架构。

Comware7 平台下的 IPS 利用防火墙的包过滤规则，进行网络数据包分类和过滤。这样省去了内核态和用户态之间的转换，大大提高了系统的执行效率。通过用户管理模块的命令和控制平台的文件两种方式对 Snort 进行配置，使得规则的更新和配置更为的方便，利于用户随时更新规则，能够更好地针对新的攻击进行防御。

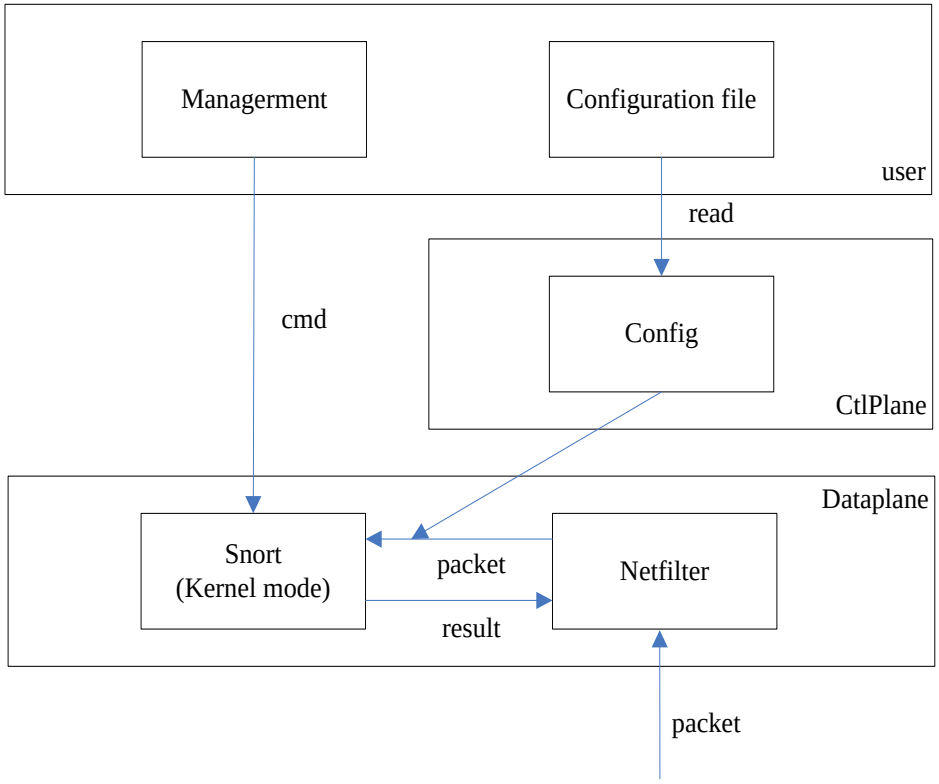

图 4.2 Comware7 平台下 snort 的架构图

由于 IPS 是根据配置信息对报文进行检测的，因此在移植过程中，剔除了探测器部分的内容；对于数据包解析器的内容进行了重新编写；增加了配置解析模块；在检测模块中，修改了检测引擎的搭建方式，并且引入了多模式匹配算法来提高检测效率；在输出模块中，增加了对于匹配报文的处理过程，采用插件形式注册各个处理函数。

因此，总的来说，本系统删除了 Snort 作为 IDS 特有的部分，对于 IPS 与 IDS 通用的部分进行了大量的修改和重写，使其可以良好的应用于 Comware7 的平台中。对于 Snort 系统仅保留了它定义的特征规则和连接漏洞库的方式。

4.1.2 系统架构

这里从两个角度分别对系统的架构及其处理流程进行阐述：

1）从整个系统的角度来分析，数据报文由防火墙传来，经过 IPS 模块，进行慢转的处理，在处理过程中挂接快转结构，图 4.3 是 Comware7 平台下 IPS 的系统框架。

在对系统进行初始化的时候，会在系统的应用链表中加入 IPS 的相关信息，这是 IPS 模块的总入口。当用户希望对 IPS 模块进行操作的时候，会通过输入设备敲入相

应的命令行，命令行输入后，系统就会在应用链表中进行查找，找到 IPS 的相关信息后，会调用相关的初始化函数，将需要用到的信息进行初始化。通过防火墙的筛选，会将符合条件的报文挑选出来，送到 IPS 模块进行处理。首次处理会搭建相应的处理通道，以后再有类似的报文进来，就会利用搭建好的通道快速的进行处理了。

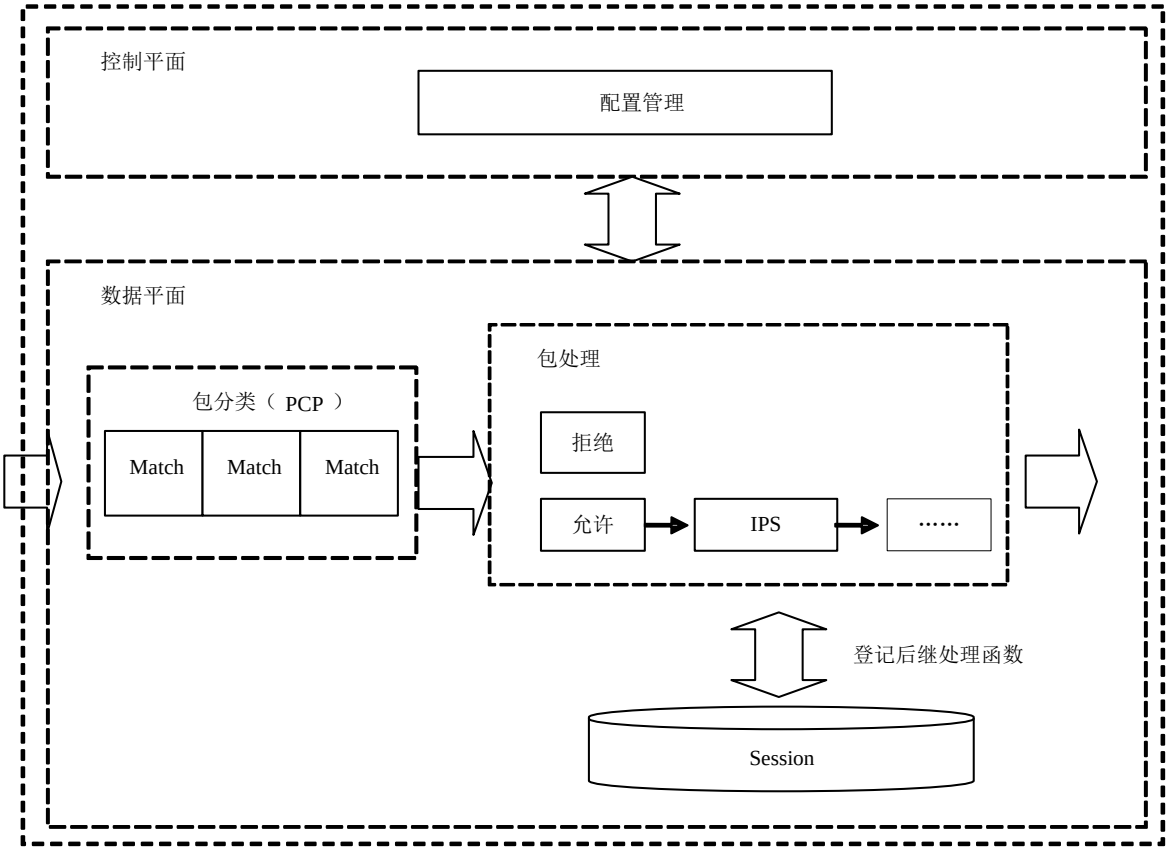

图 4.3 Comware7 平台下 IPS 的系统框架图

2）从 IPS 内部的角度进行描述，如图 4.4 所示。

首先用户下发配置信息，配置信息由配置解析模块进行解析，取出其中的关键信息存入相应的全局数据链表中，预处理模块会从不同的全局数据链表中取出相应的配置信息，然后调用不同的插件对信息进行处理，其中的特征信息，会被搭建成特征树，为检测模块以后快速搜索引擎的搭建提供数据。

然后系统开始接收报文，当防火墙将符合条件的报文送达 IPS 模块后，首先经过解析，然后送到预处理模块，此模块的主要工作是在报文进入检测引擎之前，对其进行前期的处理，例如：分片数据包的重组、HTTP 请求 URL 字符串需要统一格式化等等；数据报文经过预处理后，进入检测模块，检测模块会根据特征树的信息搭建快速搜索引擎，并且连接 Snort 所提供的漏洞库，这里将会涉及到多模式匹配算法，之后利用快速搜索引擎对报文进行检测，最终返回检测的结果。报文经过检测后到达输出模块，此模块会根据检测结果，进行相应的处理，例如记录日志、报警、丢弃等操作。如果报文是安全的，就送到转发模块进行转发处理。

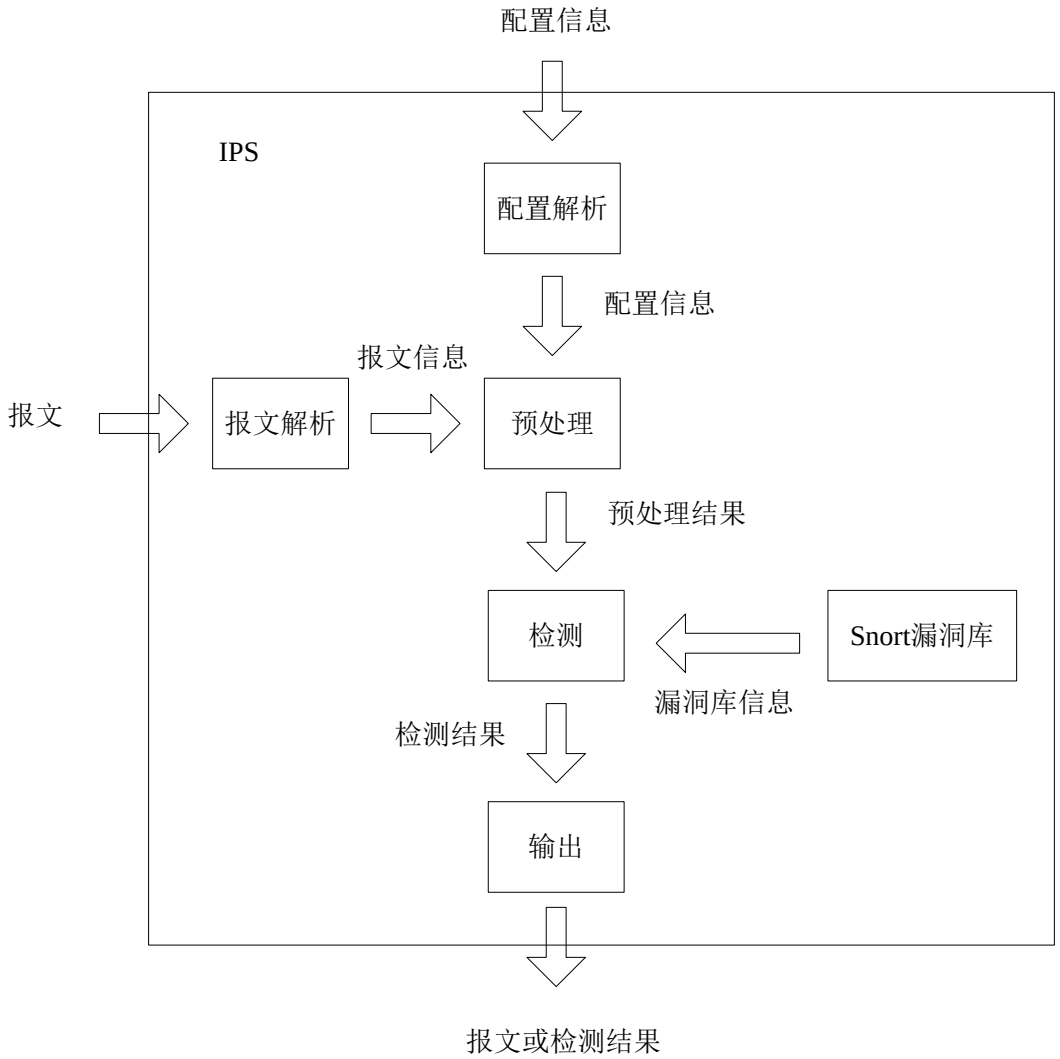

图 4.4 IPS 内部框架图

根据以上分析可以将本系统分为配置解析模块、报文解析模块、预处理模块、检测模块和输出模块五部分。每一部分如下描述：

配置解析模块，主要用于解析用户下发的配置文件和配置命令，并将解析好的结果存入相应的结构体中，然后交给预处理模块进行处理。

报文解析模块，主要用于解析防火墙传送过来的报文，将解析出来的结果存入相应的数据结构中，然后交给预处理模块进行处理。

预处理模块，主要用于处理解析好的配置信息和报文信息，根据配置信息完成初始化相关结构、注册插件、初始化插件、搭建特征树等相关操作；根据报文信息调用相应的报文处理插件，将报文信息处理成检测模块可用的规范化形式；然后将处理好的报文信息和特征信息送到检测模块进行检测。

检测模块，根据预处理模块提供的特征信息搭建快速搜索结构，然后将从预处理模块得到的报文信息与之进行比对，将比对的结果送到输出模块进行处理。

输出模块，处理检测模块发送到结果，去除其中重复的部分，将最终的结果输出，反馈给用户。如果报文是安全的，就送到转发模块进行转发处理。

下面将分模块的介绍对其的设计与实现。这里主要介绍了作者在项目中所完成的内容，其他组员的工作内容并没有在文中出现。

4.2 配置解析模块的设计与实现

配置解析模块主要负责解析用户定义的配置文件和用户输入的配置信息。首先会将读入的信息进行分类，不同的类别有与之相应的解析函数；通过对应的解析函数解析出配置信息中需要的部分，然后将结果存储在对应的数据结构中；最后将存储好结果的数据结构挂接在相应的链表中，为后续的处理做准备。

本模块分为解析配置文件、解析特征文件、解析特征关联文件和解析特征分类文件四部分。如图 4.5 所示。

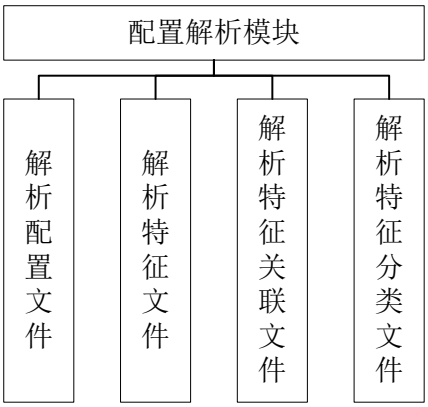

图 4.5 配置解析模块功能分解图

4.2.1 解析配置文件

解析配置文件，这部分主要负责，初始化配置相关数据结构；解析文件路径；解析 IP 地址定义；解析端口定义；解析预处理定义；解析约束门限定义；解析约束禁止定义。就是通过文件中的不同关键字，将文件进行分类，然后调用不同的解析函数进行解析。如图 4.6 所示。

按关键字可以将命令行分为以下 6 类：

（1）解析配置文件中包含关键字 include 的命令，如果配置的关键字是 include，先判断 include 后面是否为变量代称形式。如果不是变量代称形式，则直接获取包含的文件路径和文件名称字符串；如果是变量代称形式，则调用 Ips_Parser_VarGet 函数查找 include 关键字后面的变量是否包含在 pstVarHead 链表中，如果包含，说明该变量已经定义，则获取包含的文件路径和变量名称字符串，如果未包含，说明该变量

未定义，返回 NULL。然后，将获取到的包含文件路径和文件名称传入解析文件函数 Ips_Parser_ParseRuleFile，使用该函数对这类文件进行解析。

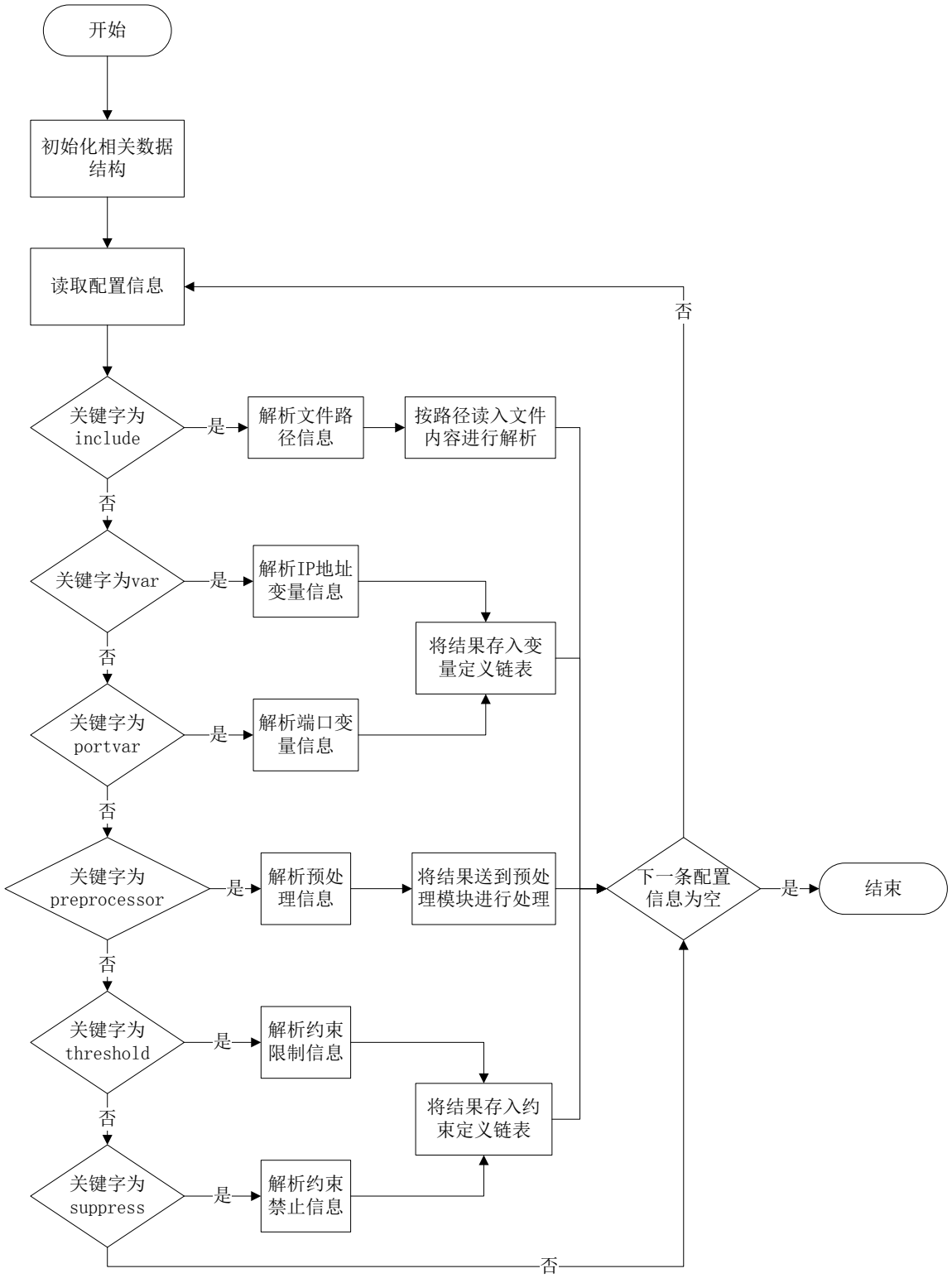

图 4.6 解析配置文件流程图

（2）解析配置文件中包含关键字 var 的命令。如果关键字为 var 则定义的是 IP 地址变量，调用 Ips_Parser_VarDefine 函数进行解析，参数是定义的变量名称和变量

值。该函数将变量名称和变量值进行关联，如有定义 var home_net 10.1.26.1, var http_server $home_net，该函数会将$home_net 与 10.1.26.1 进行关联，在判断没有重复储存后，会将其存储到 IPS_VARENTRY_S 类型结构里，并将该结构作为结点插入全局变量链表 g_pstVarHead 中。

（3）解析配置文件中的端口定义关键字 portvar。如果关键字为 portvar 则定义的是端口号变量，调用 Ips_Parser_PortVarDefine 函数进行解析，参数是定义的变量名和变量值。首先查看该变量是否重复定义。若没有重复定义，判断变量的值是否为 any，即是否是任意端口。由于端口值不允许为!any，所以先判断 any 前是否有！号，如果有则出错退出；如果没有则将该端口信息存入以该变量名命名的端口对象中；如果变量的值不是 any，则建立以该变量名命名的端口对象。将上述建立的端口对象加入到全局端口对象哈希表 g_pstPortVarTable 中。

（4）解析配置文件中的预处理定义关键字 preprocessor，如果配置的关键字是 preprocessor，则调用 Ips_Parser_ParseProcessor 函数进行解析，该函数先解析出预处理插件名称和该预处理插件所配置的内容，然后将解析的结果送到预处理模块进行相应的处理。

（5）解析规则约束文件中的门限关键字 threshold，如果配置的关键字是 threshold，则调用 Ips_Parser_ParseSFThreshold 函数进行解析，然后将解析出来的关键字存入 IPS_THDXSTRUCT_S 类型数据结构 stthdx 中，在存入前需要判断是否重复存储。

（6）解析规则约束文件中的限制关键字 suppress，如果配置的关键字是 suppress，则调用 Ips_Parser_ParseSFSuppress 函数进行解析，将解析的结果存到与（5）相同的数据结构中，与其操作相同。

4.2.2 解析特征文件

解析特征文件，首先根据配置文件的定义，读取特征文件；然后依次解析文件中的每个特征；接着解析特征的动作、协议，对于本系统而言，特征的动作是另外处理的，这里所定义的动作并不作为特征的处理动作。之后解析特征的网络信息，由于 IPS 的设置与包分类策略进行关联，因此特征的地址信息实际上是没有用的（端口在标识特征时使用），这里为了移植的原因，不做修改，但在检测时不会对地址信息进行检测；然后解析特征选项，遍历系统选项链表，初始化选项检测模块；最后将以上信息传递给初始化模块搭建特征树。

首先对特征头中的动作进行解析，解析出 drop、alert、pass、log 四种动作，并将其结果进行记录。

然后对特征头中的协议进行解析，解析出 TCP、UDP、IP 和 ICMP 四中协议，

并将解析出的协议类型标志值进行记录。

然后对特征头中的 IP 地址进行解析，记录解析结果。由于特征中必须包括源 ip 和目的 ip，所以需要对源 ip 和目的 ip 分别进行解析。

之后解析特征头中的端口号，记录解析结果。由于特征中必须包括源端口和目的端口，所以要对源端口和目的端口分别进行解析。

之后解析特征选项中的内容，将特征选项的关键字和选项内容分别解析出来并存入相应的数据结构。现支持的特征选项有 msg、ttl、id、dsize、content、seq、ack、itype、icode、icmp_id、icmp_seq 等。

最后将解析出来的结果送到预处理模块搭建特征树。

如图 4.7 所示，是解析特征文件的流程图。

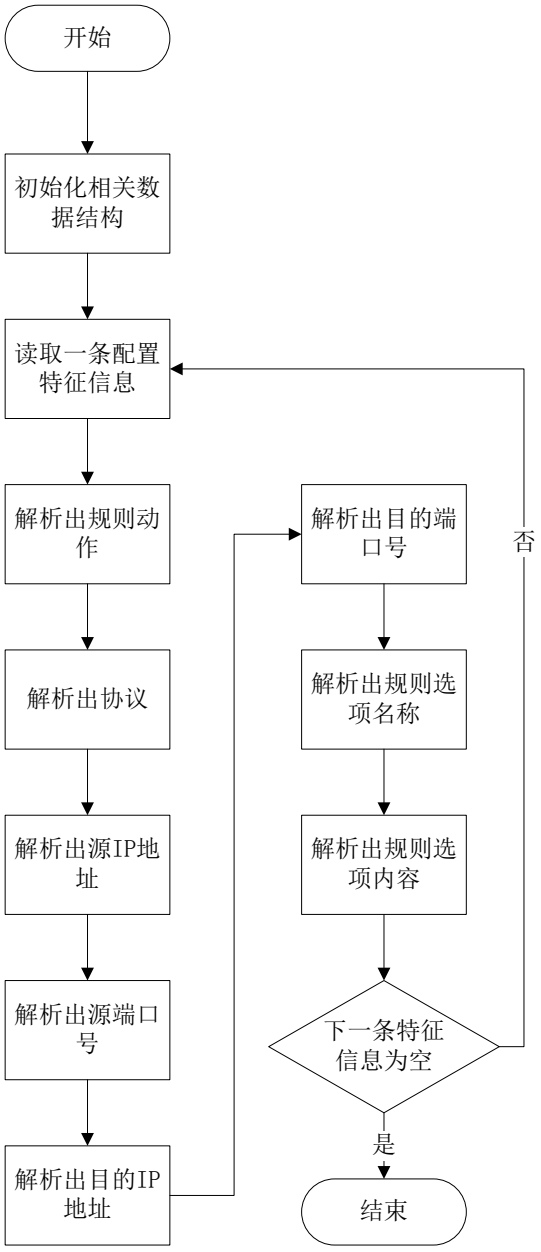

图 4.7 解析特征文件流程图

4.2.3 解析特征关联配置文件

很多漏洞库中都详细的阐述了 IPS 特征的解决办法、受影响范围等信息，例如 cve、bugtraq 等。IPS 特征在选项中也定义了此特征在漏洞库的位置。但为了减化配置，对于漏洞库相同的情况，没有必要输入完整的 url 地址，只需要输入漏洞库的名称和特征 ID 即可，为了应对这种情况，定义了一个特征关联文件，此文件定义了一组 url 以及标识这些 url 的名称，这样我们就可以在特征选项中作如下定义了：reference:cve, 2000-0922。本模块主要负责的是，根据配置文件的定义，读取特征关联命令，分别获取名称与 url，最后将其存放在相关数据结构中。

本操作分为以下三部分：

（1）解析配置内容中包含特征关联关键字 reference 的命令，由 Ips_Parser_ParseReferenceSystemConfig 函数负责解析，过程如下：

先解析出特征关联 url 名称和 url 地址，分别赋给 url 名称和地址变量。如果未设置 url 地址，则将 url 地址变量置为 NULL。

将 url 名称和 url 地址交给 Ips_Parser_ReferenceSystemAdd 函数做进一步解析。

（2）特征关联内容解析结果的保存，由 Ips_Parser_ReferenceSystemAdd 函数进行解析，过程如下：

将 url 名称和地址存入 IPS_REFSYSNODE_S 类型结构中。

然后将填充好的 IPS_REFSYSNODE_S 类型结构插入系统特征关联链表中。

（3）系统特征关联结构链表，用于存放系统特征关联结构的链表。

4.2.4 解析特征分类文件

特征分类文件，定义了分类信息以及缺省的优先级。在输出日志信息时，可以根据优先级，分类进行打印。

本模块主要负责的是，根据配置文件的定义，读取特征分类文件；解析所定义的分类名称、分类名称注释信息和优先级；将解析的信息存入相关的数据结构。

本操作分为以下三部分：

（1）解析配置内容中的特征分类关键字 classification，由 Ips_Parser_ParseClassificationConfig 函数负责解析，该函数先解析出特征类型、特征名称和特征优先级，再创建 IPS_CLASSTYPE_S 类型数据结构，并将上述解析出的各项存入该数据结构中。最后调用 Ips_Parser_ClassificationConfigAdd 函数，将上述填充好的数据结构作为一个结点加入特征分类链表中。

（2）特征分类解析结果的保存，由 Ips_Parser_ClassificationConfigAdd 函数完成，其实现过程为，遍历特征分类结构链表，检查该链表中是否已经存在此类特征，如果

存在则返回 SSP_ERR，如果不存在则将结果填充到对应的数据结构链表中。结点 id 号随结点数的增加依次加 1。结点 id 号一般为输出日志所用。

（3）特征分类结构链表，用于存放特征分类的结构链表。

4.3 报文解析模块的设计与实现

报文解析模块，处理由防火墙包分类策略进行分类后的报文，将其中需要进行 IPS 处理的报文，按协议进行解析处理，并将解析后的协议头以及数据区信息，填充到报文数据结构中，然后根据报文的类型调用相应的插件进行处理。

本操作分为以下两部分：

（1）解析报文，将报文按协议进行解析；将解析好的协议头信息、数据信息等存入报文数据结构。即就是，报文信息存储单元，存放报文根据协议解析后的信息；后续预处理和检测模块都以此结构进行处理。其处理过程，如图 4.8 所示。

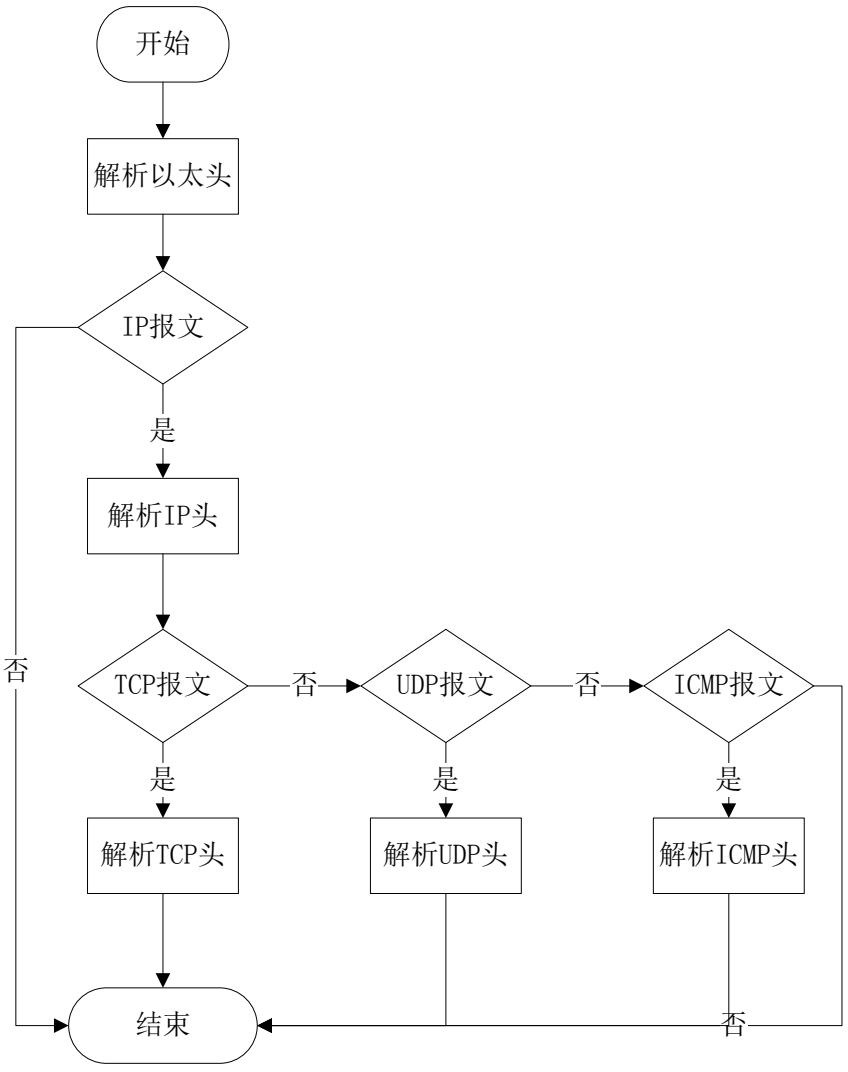

图 4.8 报文解析流程图

首先解析报文以太头的信息，其方式是，将报文强制转化成以太头类型的数据结

构，取出其中的协议信息进行判断，如果是 IP 报文，指向报文的指针偏移一个以太头的距离后进行后续解析；如果是非 IP 报文，不是本系统处理的报文类型，结束解析，让报文通过进行后续处理。

然后解析报文 IP 头中的信息，将报文强制转化成 IP 头类型的数据结构，取出其中的源、目的 IP 地址、ttl、IP 标识、分片信息、协议等信息，将这些信息存入 Packet 类型的数据结构中。

然后判断 IP 头中的协议信息是否是 TCP、UDP 或者 ICMP 中的一种，如果是，将指向报文的指针偏移一个 IP 头的长度后进行下一层的解析；如果不是结束解析。

根据 IP 头中不同的协议类型，调用不同的解析函数进行解析，方法与 IP 头、以太头类似，将报文强制转换成该协议类型的数据结构然后取出其中源、目的端口号等信息，存入 Packet 类型的数据结构中。

（2）调用插件进行处理，包含处理 http 报文的插件、处理流重组报文的插件、处理分片报文的插件、处理 RPC 报文的插件、处理 BO 报文的插件、处理 Telnet 报文的插件、处理 FTP 报文的插件、处理端口扫描报文的插件、处理 SMTP 报文的插件、处理 SS1 报文的插件、处理 SSH 报文的插件、处理 DNS 报文的插件等。

4.4 预处理模块的设计与实现

预处理模块分为初始化、插件管理和搭建特征树三部分。

4.4.1 初始化

本部分用于初始化将要用到的全局变量。主要完成了以下内容的初始化：

初始化用户自定义特征标志变量 ulCustomFlag 为 0，代表没有自定义特征。之后根据命令行命令的指示设置该标志：如果没有自定义特征，该标志保持初始值 0；如果有自定义特征，命令行会下发一条命令通知控制平面，收到命令后则将该标志设置为 1，代表有自定义特征。

内存中特征树的初始化，判断用户自定义特征标志是否有效。如果用户自定义特征标志无效，即没有用户自定义特征，则直接初始化特征树。如果用户自定义特征标志有效，即有用户自定义特征，则先清除原有的特征树和优化结构，再初始化特征树。

初始化各种数据结构和表，用于存放将来解析的结果。

4.4.2 插件管理

插件管理，分为预处理插件管理、检测插件管理。其功能是注册插件，根据配置信息进行插件的查找，并对插件进行初始化，其中包括注册为插件提供功能的函数。它将控制平台的配置信息下发到对应的预处理插件、检测插件和输出插件。

预处理插件管理，根据配置文件中的预处理定义对插件进行配置。其功能有，预处理插件初始化，统一调用各个预处理模块的初始化函数；预处理插件注册，将各个预处理模块统一注册到全局链表；预处理插件的清除；预处理功能链表的清除；根据处理目的不同，提供 API 将预处理插件的各个函数加入到相应处理链表，例如，模块的清除函数，应注册在相应的清除链表，在 IPS 模块卸载时进行调用；模块的功能函数，应注册在功能链表，在报文经过时进行调用。

检测插件管理，根据特征文件中的选项对检测插件进行相应的处理。其功能有，检测插件初始化，统一调用各个检测模块的初始化函数；检测插件注册，将各个检测模块统一注册到全局链表；检测插件的清除；检测功能链表的清除；根据检测不同，提供 API 将检测插件的各个功能函数加入到相应功能链表。

4.4.3 搭建特征树

特征树将由配置解析模块解析的用户配置信息按动作、协议存放到相应的结构中，并将储存好的结构送到检测模块用于搭建快速搜索引擎。

特征树以分类的思想存放特征的所有信息。如何高效的建立 IPS 的特征树，是提高 IPS 运行效率、降低误报、漏报的关键。图 4.9 就描述了 IPS 特征树的主要结构，特征树主要分为四层，分别对应动作、协议、特征网络信息（源 IP 地址、目的 IP 地址、源端口号和目的端口号）和特征选项信息。这样设计能使特征树尽可能精简、有相同交集的特征部分不需要重复存放，例如相同动作、协议的特征放到一起，最大程度的节省内存资源。

另外特征树可以根据 IPS 的特点加以修改。由于 IPS 对于特征的处理是可以进行动作的修改，而基于此特征树，如果要修改动作较为耗时，因此设计的特征动作全部是 Alert（此动作不涉及真正的处理动作），另外设计一个动作处理结构来关联特征的处理动作，在修改动作时只需要根据特征 ID 查找并修改相关结构就行了，不需要对特征树进行操作；而对于地址部分的处理，由于 IPS 是根据包分类策略进行地址、端口匹配的。这就意味着特征的地址部分实际上是没有用的（端口在标识特征时使用），因此在处理时可以保留此部分的解析，但在检测时则不需要检查地址。保留动作与地址部分的原因在于对特征以及解析部分不用作大的修改，同时又不影响后续的处理，解析所做的额外处理也不是很多。

初始化的时候会将特征树的前两层进行初始化，而对于后两层的搭建，数据来源于配置解析模块解析出的特征信息。

预处理模块接收到特征信息的解析结果，首先会根据解析结果搭建特征树的第三层链表。首先定义特征树第三层链表结构类型指针 pstRtn_idx。如果协议类型为 tcp 协议，则让 pstRtn_idx 指向 tcp 协议链表头；如果协议类型为 udp 协议，则让 pstRtn_idx

指向 udp 协议链表头；如果协议类型为 icmp 协议，则让 pstRtn_idx 指向 icmp 协议链表头；如果协议类型为 ip 协议，则让 pstRtn_idx 指向 ip 协议链表头。如果 pstRtn_idx 指向地址为空，说明当前使用的协议链表中尚无结点。为当前使用的协议链表开辟新结点做为链表头。如果 pstRtn_idx 指向地址不空，说明当前使用的协议链表中已经有结点。遍历该协议链表，逐个结点比较 pstRtn_idx 的 iplist 和 stProto_node 的 iplist 是否完全匹配（包括 ip 地址的匹配和端口的匹配），若匹配说明链表中已经存在该结点，不做结点的挂接，释放结点空间。若不匹配说明链表中不存在该结点，则为其分配内存空间，将新节点插入链表中。

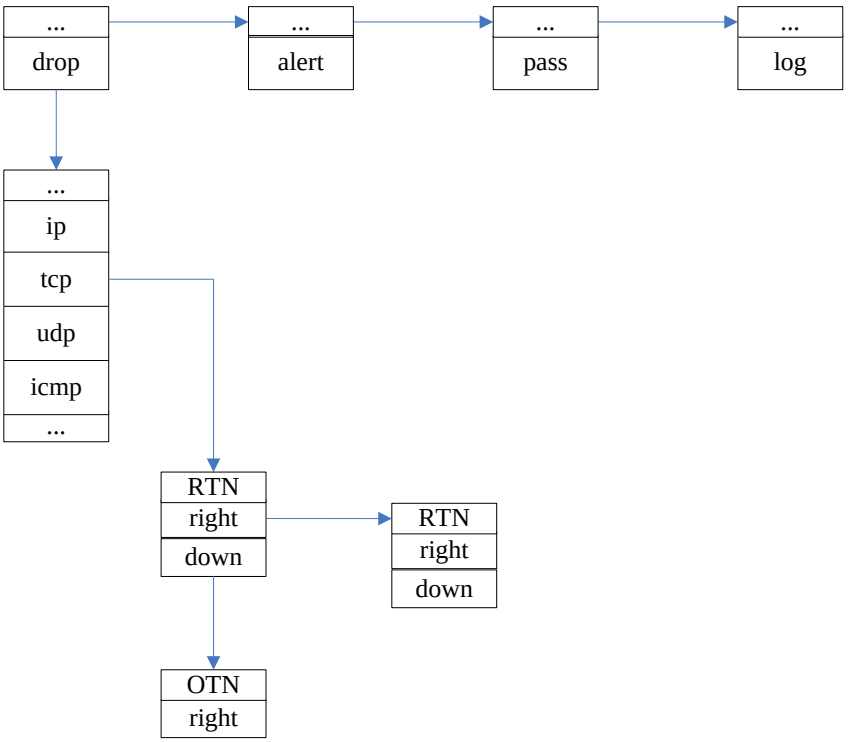

图 4.9 特征树结构示意图

然后根据解析的结果搭建特征树的第四层链表。首先为其分配空间并初始化。然后根据各个选项所解析出来的结果对其中的选项关键字和对应的内容进行填充。

之后将特征树的第三层链表和第四层链表添加到特征树中，构成完整的特征树，完整的特征树，包括四层链表：第一层，IPS_RULELISTNODE_S 类型结构链表，即特征动作链表；第二层，IPS_PROTOLIST_S 类型结构，成员是特征协议链表表头；第三层，IPS_RULETREENODE_S 类型结构链表，即特征协议和特征端口信息链表；第四层，RuleOptionNode 类型结构链表，即特征选项链表。

4.5 检测模块的设计与实现

检测模块作为本系统的核心模块，主要包含了快速搜索引擎初始化、快速搜索引

擎搭建和报文检测三部分。

4.5.1 快速搜索引擎初始化

本模块由内部快速搜索结构的创建和映射引擎两部分组成。它从特征树和端口表中获取配置，根据特征的协议、端口搭建快速搜索结构。

内部快速搜索结构的创建，从配置解析模块中获取解析的端口结构，根据协议，为每个端口对象创建端口组，为端口组创建多模式匹配结构，为多模式结构添加特征中的关键字，将端口组与特征关联。

映射引擎，由于创建的搜索引擎是在端口结构中，而端口结构对于进行报文的匹配来说过于复杂，无法根据端口来进行高效率的检测，因此需要映射到一个按协议、端口进行划分的结构中，以便于报文的快速检测。它根据内部快速搜索创建的结构，将端口结构中的 TCP 协议，按源端口将端口组映射到一个 TCP 全局变量中的数组，下标为源端口值；将端口结构中的 TCP 协议，按目标端口将端口组映射到一个 TCP 全局变量中的数组，下标为目标端口值；将端口结构中的 UDP 协议，按源端口将端口组映射到一个 UDP 全局变量中的数组，下标为源端口值；将目的端口结构中的 UDP 协议，按目的端口将端口组映射到一个 UDP 全局变量中的数组，下标为目的端口值；将端口结构中的 IP 协议，按上层协议值将相应组映射到一个 IP 全局变量中的数组，下标为上层协议值；将端口结构中的 ICMP 协议，按 ICMP 类型值将相应组映射到一个 ICMP 全局变量中的数组，下标为 ICMP 类型值。

4.5.2 快速搜索引擎搭建

快速搜素引擎，按协议、端口存放配置的特征，按端口搭建模式匹配结构。通过对规则树进行二次分类，以提高规则的检测匹配效率。按照特征的源、目的端口进行搭建，对同一类的特征关键字采取多模式匹配的算法，即报文只需通过一次检测就可以知道是否命中这一类关键字中的一个，无须对每个关键字进行检测，从而大大节省检测的时间。针对不同协议，会有不同的端口处理方式。对于 TCP 或者 UDP 报文，根据源端口为特定值，加入源端口值对应的子集；根据目的端口为特定值，加入目的端口值对应的子集；如果源端口、目的端口都为任意值 any，则加入通用子集中。对于 ICMP 报文，如果在规则选项中指定了 ICMP 类型值，则目的端口为其类型值，加入目的端口值对应子集，如果没有指定，则为任意值 any，加入通用子集。对于 IP 报文，如果在规则选项中指定了 IP 高层协议类型值，则目的端口为其协议类型值，加入目的端口值对应子集，如果没有指定，则为任意值 any，加入通用子集。

4.5.3 报文检测

报文检测分为快速检测、端口检测和选项检测三部分。根据搭建的搜索引擎对报

文进行检测，不匹配则说明报文属于安全报文，允许报文通过，进行后续的处理，匹配则说明报文属于攻击报文，返回匹配的特征分类 ID，输出模块会根据不同的特征类型作出相应的处理。

1）对特征头的检测

为了提高检测效率，首先对报文的协议与端口号进行匹配，如果匹配成功，则继续匹配特征选项的内容，这样可以保证比较的次数最小，从而提高匹配效率。其具体流程如图 4.10 所示。

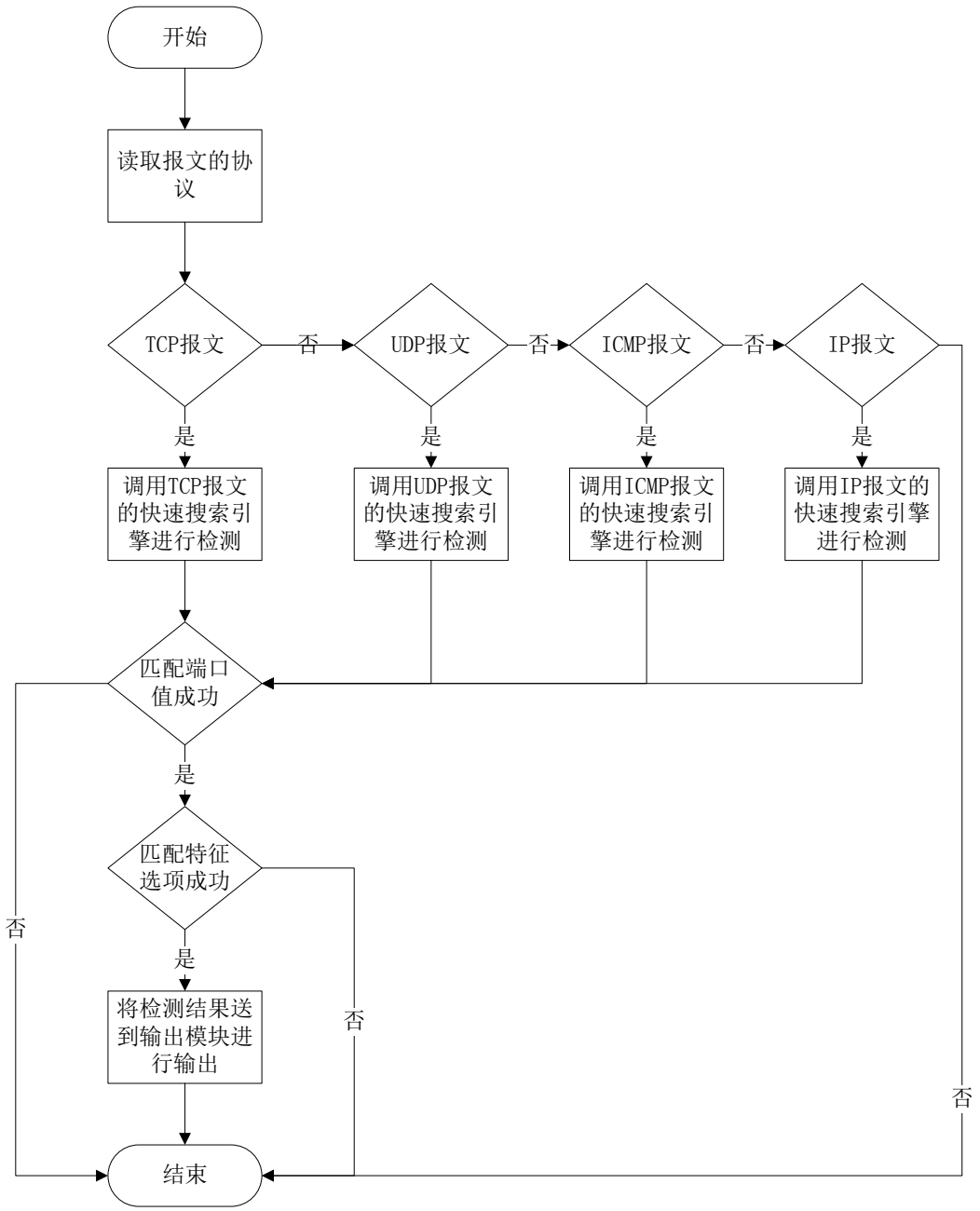

图 4.10 报文检测流程图

首先进行快速检测，根据报文协议、IP 地址，从快速搜索引擎中获取对应的多模式匹配结构；调用模式匹配算法，进行匹配；

然后进行端口检测，特征中网络信息部分的检测，由于地址不需要进行检测，因此只对端口进行检测。

最后进行选项检测，进行完快速检测后，对于匹配的报文，还需要检测特征选项中其他条件是否也满足。根据快速搜索引擎关联的特征选项链表，遍历每个特征选项的名称和内容，如果有符合的情况，则说明报文属于攻击报文，返回匹配成功的特征分类号。

2）对特征选项的检测

对特征选项中的内容进行匹配，主要分为两类，一类是匹配协议头中除了端口和协议类型的信息，如 ttl、IP 报文的标识等信息；一类是匹配报文负载内容的信息。

本系统支持 11 种特征选项，其中 msg 选项中的内容是需要在输出中显示的，因此不需要匹配。在剩下的 10 种特征选项中，只有 content 一种是需要比较报文负载信息的。因此第二类特征选项只包括 content 一个。

对于第一类的匹配比较简单，其具体步骤如下：

（1）取出特征选项的名称，判断是哪一个特征选项，以 ttl 为例，如果判定为是 ttl 特征选项，则取出特征选项中的内容；

（2）将报文信息中的 ttl 取出与特征选项中的内容进行比较，如果相等，则匹配成功，返回相应的特征分类 ID；如果不相等，则匹配失败继续匹配剩下的特征选项。

（3）如果将所有的特征选项都匹配完了，没有一个是成功的，则说明该报文属于安全报文，允许报文通过进行后续的处理。

对于第二类的匹配，首先要将本层中所有特征选项名称是 content 的选项提取出来，然后将其中的内容搭建成一个多模式匹配结构。其结构如第二章中 2.3 小结中的扩展算法结构，由状态、左子树和右子树三部分构成，状态表明单个字符匹配成功后所处的状态，设置状态主要为了避免在匹配过程中产生回溯的问题；左子树指向不同的字符；右子树则是指向同一字符串的后续字符。按照上述规则将所有 content 规则选项内容中的字符串构造成一个多模式匹配结构。详细过程在第二章中有详细的描述，这里不再赘述。

然后将报文的负载部分放到多模式匹配结构中进行匹配，如果与其中一条匹配成功，则判定为匹配成功，该报文是攻击报文，返回特征分类 ID，由输出模块进行输出；如果全部匹配结束后也没有匹配成功的，则判定为安全报文，允许报文通过，进行后续处理。

4.6 输出模块的设计与实现

输出模块，用于将检测的结果进行输出，根据不同的类型（警告、记录日志或丢弃报文等形式）进行相应的处理。本模块由匹配事件添加、匹配事件的过滤、输出事

件和清除事件四部分组成，其处理流程如图 4.11 所示。

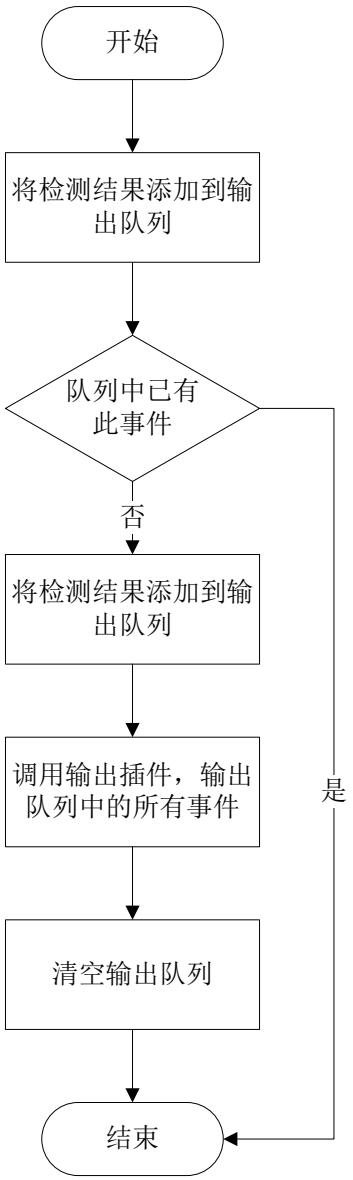

图 4.11 输出模块流程图

匹配事件添加，根据类型（Alert、Log、Drop 等），将匹配事件添加到相应事件队列中；在添加的时候，会将需要添加的事件与队列中已有的事件进行比较，对于相同的匹配事件（相同特征触发的事件），不重复添加；判断队列的长度，超过最大长度不进行添加。

匹配事件的过滤，根据匹配特征的优先级进行事件的排序；再次检查是否有相同的匹配事件，不重复添加；将事件加入事件链表。

输出事件，从事件链表中遍历事件信息；调用注册的输出插件输出事件，进行记录。这里会根据不同的特征信息，对报文进行不同的处理，如发出警告、记录日志或者丢弃报文等。

清除事件，清除事件队列。

4.7 本章小结

本章节是本论文的核心部分，主要从系统总体架构和各个模块两个方面介绍了本系统的设计思路。

在系统架构设计中，主要说明本系统中的 Snort 和传统的 Snort 在系统架构中的区别，并说明了本系统设计的优势。然后对处理流程进行了介绍，从系统整体处理流程和系统内部处理流程两个方面介绍了 IPS 的处理过程，从总体上对于系统有了一个框架的设计。

在各个模块的设计中，从配置解析模块、报文解析模块、预处理解析模块、检测模块和输出模块五部分对系统进行了具体的介绍，介绍了其中的功能模块和具体的实现。

第五章 入侵防御系统测试及分析

本章节主要介绍了针对本系统所做的测试方面的内容，首先介绍了测试的环境，然后根据测试用例介绍了测试的过程，最后根据测试结果进行了分析。

5.1 系统测试环境

由于该系统是基于 H3C 公司的 Comware7 平台设计的，所以选取了比较常用的设备 WX5540E 对系统的功能进行测试。图 5.1 为测试环境拓扑图。

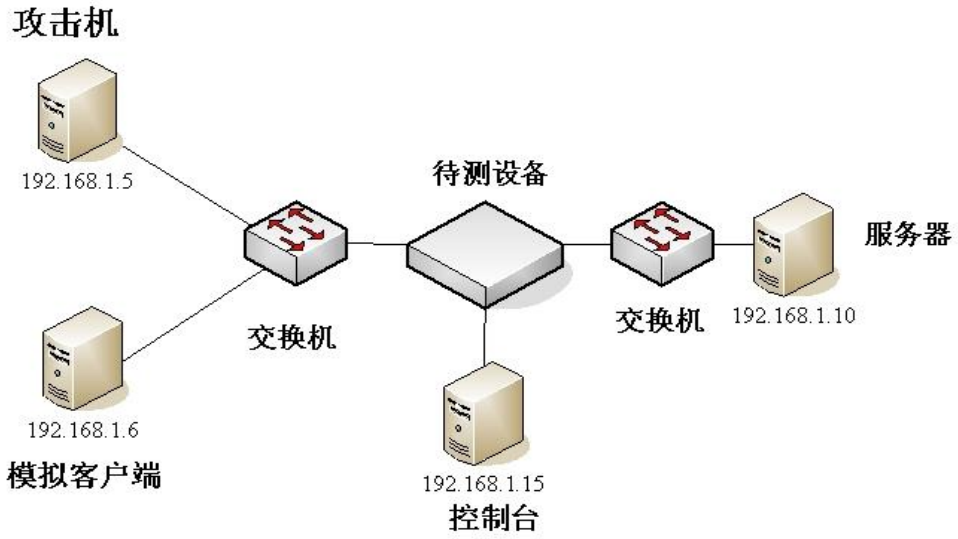

图 5.1 测试环境拓扑图

所有主机中都安装 Windows2000 以上系统，按照图中 IP 地址进行配置。

然后对待测设备进行配置，首先启动 IPS 系统：

<H3C>sy

[H3C]ips enable

[H3C]dis this

…

ips enable

…

启动设备后，输入 sy（system 的简写）命令进入系统配置视图，然后输入命令 ips enable 启动 IPS 系统，输入 dis this 进行查看查看，如果在显示的选项中看到了 ips enable 这一项表明 IPS 系统已经启动了。

然后配置一条防火墙包分类策略，由于测试攻击机的 IP 地址为 192.168.1.5，因此以源 IP 地址为 192.168.1.5 为分类标准：

[H3C]pcp pcp1

[pcp1]source ip 192.168.1.5/16

新建一个名称为 pcp1 的包分类策略，策略的内容是源 IP 地址为 192.168.1.5 的报文。

之后，新建一个 IPS 的策略，并与包分类策略相关联：

[H3C]ips

[IPS]policy policy1

[policy1]quit

[IPS]quit

[H3C]ips pcp pcp1 apply policy policy1

进入 IPS 配置视图，新建一个名称为 policy1 的 IPS 策略，然后退出到系统配置视图，将上面建立的包分类策略与 IPS 策略相关联。这样就可以通过防火墙对报文进行分类，然后对符合标准的报文进行检测了。

5.2 测试用例

针对本系统，可以从配置解析、报文解析和报文检测三方面进行单元测试，这样可以覆盖所有的功能点。其测试用例如下描述。

5.2.1 配置解析功能测试用例

对于配置解析功能的测试，是系统的基础功能测试，主要测试系统对于配置命令的解析是否正确。其测试用例如表 5.1 所示。

表 5.1 配置解析功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
文件路径定义的解析	测试 IPS 可否正确解析文件路径的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义文件路径的命令 4、查看 debug 信息与输入的信息是否一致	文件路径配置信息	解析结果	debug 显示的信息与输入的信息一致
IP 地址变量定义的解析	测试 IPS 可否正确解析 IP 地址变量的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义 IP 地址的命令 4、查看 debug 信息与输入的信息是否一致	IP 地址定义配置信息	解析结果	debug 显示的信息与输入的信息一致
端口号变量定义的解析	测试 IPS 可否正确解析端口号变量的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义端口号的命令 4、查看 debug 信息与输入的信息是否一致	端口号定义配置信息	解析结果	debug 显示的信息与输入的信息一致

续表 5.1 配置解析功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
预处理定义的解析	测试 IPS 可否正确解析预处理的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义预处理的命令 4、查看 debug 信息与输入的信息是否一致	预处理配置信息	解析结果	debug 显示的信息与输入的信息一致
约束门限定义的解析	测试 IPS 可否正确解析约束门限的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义约束门限的命令 4、查看 debug 信息与输入的信息是否一致	约束门限配置信息	解析结果	debug 显示的信息与输入的信息一致
约束禁止定义的解析	测试 IPS 可否正确解析约束禁止的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义约束禁止的命令 4、查看 debug 信息与输入的信息是否一致	约束禁止配置信息	解析结果	debug 显示的信息与输入的信息一致
特征定义的解析	测试 IPS 可否正确解析特征的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义特征的命令 4、查看 debug 信息与输入的信息是否一致	特征定义配置信息	解析结果	debug 显示的信息与输入的信息一致
特征关联定义的解析	测试 IPS 可否正确解析特征关联的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义特征关联的命令 4、查看 debug 信息与输入的信息是否一致	特征关联配置信息	解析结果	debug 显示的信息与输入的信息一致
特征分类定义的解析	测试 IPS 可否正确解析特征分类的定义	1、进入 IPS 系统配置视图 2、打开 debug 模块 3、输入定义特征分类的命令 4、查看 debug 信息与输入的信息是否一致	特征分类配置信息	解析结果	debug 显示的信息与输入的信息一致

5.2.2 报文解析功能测试用例

对于报文的解析，是报文检测的基础，报文检测的测试主要是查看解析结果是否正确。其测试用例如表 5.2 所示。

表 5.2 报文解析功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
IP 协议头解析	测试 IPS 对于报文 IP 头的解析是否正确	1、从攻击机向待测设备发送 IP 报文 2、打开测试设备报文解析部分的 debug 模块 3、查看 debug 输出的信息与报文信息是否一致	IP 报文	解析结果	debug 输出的信息与报文信息一致
TCP 协议头解析	测试 IPS 对于报文 TCP 头的解析是否正确	1、从攻击机向待测设备发送 TCP 报文 2、打开测试设备报文解析部分的 debug 模块 3、查看 debug 输出的信息与报文信息是否一致	TCP 报文	解析结果	debug 输出的信息与报文信息一致

续表 5.2 报文解析功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
UDP 协议头解析	测试 IPS 对于报文 UDP 头的解析是否正确	1、从攻击机向待测设备发送 UDP 报文 2、打开测试设备报文解析部分的 debug 模块 3、查看 debug 输出的信息与报文信息是否一致	UDP 报文	解析结果	debug 输出的信息与报文信息一致
ICMP 协议头解析	测试 IPS 对于报文 ICMP 头的解析是否正确	1、从攻击机向待测设备发送 ICMP 报文 2、打开测试设备报文解析部分的 debug 模块 3、查看 debug 输出的信息与报文信息是否一致	ICMP 报文	解析结果	debug 输出的信息与报文信息一致

5.2.3 报文检测功能测试用例

对于报文的检测，是本项目的核心部分，也是本项目的主要功能，对于其测试，主要是测试系统可否检测出符合特征的报文。其测试用例如表 5.3 所示。

表 5.3 报文检测功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
log 动作 msg 选项测试	测试 IPS 可否正确记录规定的消息	1、配置 IPS 特征 log 动作的 msg 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并按照 msg 中的配置输出一条消息
drop 动作 ttl 选项	测试 IPS 可否判断出规定 ttl 的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 ttl 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 id 选项	测试 IPS 可否判断出规定 IP 标识的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 id 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 dsize 选项	测试 IPS 可否判断出规定数据包载荷的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 dsize 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 content 选项	测试 IPS 可否判断出含有规定字段的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 content 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文

续表 5.3 报文检测功能测试用例表

测试项	测试目的	测试步骤	输入	输出	预期结果
drop 动作 seq 选项	测试IPS可否判断出规定TCP 序列号的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 seq 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 ack 选项	测试IPS可否判断出规定TCP 确认序列号的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 ack 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 itype 选项	测试IPS可否判断出规定ICMP 类型的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 itype 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 icode 选项	测试IPS可否判断出规定ICMP 代码的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 icode 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 icmp_id 选项	测试IPS可否判断出规定ICMP 标识的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 icmp_id 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文
drop 动作 icmp_seq 选项	测试IPS可否判断出规定ICMP 序列号的报文，并丢弃报文	1、配置 IPS 特征 drop 动作的 icmp_seq 选项 2、从攻击机发送一条攻击报文 3、检测测试结果	攻击报文	测试结果	IPS 成功检测出攻击报文，并丢弃报文

5.3 测试过程及结果

对于配置解析的测试和对于报文解析的测试，测试方法和内容基本相同，下面以配置特征为例进行说明。

首先进入系统 debug 配置界面，打开配置解析的 debug 功能：

<H3C>debug

[DEBUG]ips parser config

然后进入 IPS 的策略视图配置一条特征，系统会将配置解析的结果输出：

[H3C]ips

[IPS]policy policy1

[policy1]log tcp 192.168.1.5 any -> any any(msg:“log success！”);

action : log

protocol : tcp

source ip : 192.168.1.5

source port : any

destination ip : any

destination port : any

rule option : msg

option items : log success!

开启 debug 功能，在编写代码的时候使用 debug 输出，就可以在测试的时候看到输出的内容，通过检查就能看出解析的是否正确了。

配置解析的其他测试用例和报文解析的测试用例与此相同，在此不再赘述。

下面介绍对于报文检测功能的测试。以阻断连接为例进行说明，其具体测试过程和现象如下：

（1）在 IPS 特征策略中添加特征：

[H3C]ips

[IPS]policy policy1

[policy1]drop tcp 192.168.5 any -> 192.168.1.6 any (sid:1024; rev:6;)

（2）从攻击机向测试设备发送 TCP 报文。

现象：测试设备接收不到报文。

此时查看日志文件目录，发现生成了新的日志文件，其内容是刚才丢弃报文的数据信息：

10:02:31.953086

192.168.1.5:54835 -> 192.168.1.6:80 TCP TTL:127

TOS:0x0 ID:1536 IpLen:20 DgmLen:482 DF***AP***

Seq:0x112BDD12 Ack:0x11B38D8A Win:0x4510 TcpLen:20

其中第一行为接收到攻击报文的时间；第二行显示的是源 IP 地、源端口号、目的 IP 地址、目的端口号、协议类型和 ttl；第三行显示的是 IP 协议头的信息；第四行显示的是 TCP 协议头的信息。

按照以上方法对于测试用例中的所有用例进行了详细的测试，最终修改了所有的问题，全部测试用例都通过了测试。通过以上的测试和对结果的分析，可以看出，系统在可以达到预期的目的，可以正确的解析配置信息和报文信息，可以检测报文并阻

断攻击，保证安全的网络环境，保护计算机的安全。对于系统性能方面的测试，是由公司测试小组的人员进行测试的，根据最终反馈的结果，系统在性能方面达到了预期的标准，由于这部分不是本人测试的，因此在论文没有相关的描述。

5.4 本章小结

本章节主要介绍了对于系统测试方面的内容，首先介绍了测试的环境；然后从配置解析功能、报文解析功能和报文检测功能三方面对系统进行了测试。

配置解析功能的测试，主要测试了系统对于配置信息的解析结果是否正确，其测试方法为，使用 debug 功能输出解析结果，查看与输入的信息是否一致。

报文解析功能的测试，主要测试系统解析报文的能力，其结果是否与报文中的信息一致，这里利用了 debug 的显示功能，将解析结果显示出来，直接与报文中的信息进行比较，看是否一致。

报文检测功能的测试，配置特征，发送符合特征的报文，查看检测结果，通过检测结果与预期结果是否一致来判断测试是否通过。

第六章 总结和展望

本文主要设计并实现了一个基于 Snort 的 IPS。首先从背景入手，对于 IPS 的发展历史和现如今的研究状况进行了分析，通过对网络发展的介绍，得出网络在生活中十分重要的结论，但是网络环境并不安全。从而引出对于 IPS 研究的重要意义。然后介绍了一些有关 IPS 的理论基础。从它的前身 IDS 开始介绍，介绍了 IDS 的分类和其组成部分，并提出了 IDS 的不足之处；之后介绍了一个广泛使用的 IDS 系统——Snort 系统，从系统结构和 Snort 规则两个方面对其进行了较为详细的介绍；接着为了提高检查的效率，引入了模式匹配算法。之后对于系统进行了需求分析，从业务描述、用例建模、数据分析和过程建模四部分对系统的需求进行了详细的介绍。然后介绍了系统的整体架构，从整体和内部两个角度，对系统的流程进行介绍。之后分模块进行介绍，详细介绍了各个模块的内部组成、功能和实现方式。最后针对系统进行了测试，主要从配置解析、报文解析和报文检测三方面对系统进行了测试，得出了本系统可以防护网络中大部分攻击的结论。

然而由于时间和个人能力有限，系统还有很多需要改进的地方：

（1）除了防火墙和入侵检测技术外，还可以加入“蜜罐”技术，补充系统的不足之处，也使得系统更加的安全。

（2）目前对于系统的配置，需要较高的专业素质，因此本系统还不能面向大众，以后可以设计更为方面简单的人机界面，使得对于 IPS 的配置更为方便简单，更大众化。

（3）由于技术有限，对于 Snort 系统的移植还不完美，有些地方还不够简洁，在以后的研究中还有待改进。

总体而言，本系统实现了入侵防御的基本功能，并且针对网络中流行的攻击方式，有良好的防御措施。但是也有其不足之处，随着网络的发展，本系统的不足将越来越明显，因此需要不断地了解新鲜的安全知识，不断地对其进行改进，以适应更为复杂的环境和攻击手段。

参考文献

[1] Younglove, W Roger. Public Key Infrastructure: How It Works[J]. Computing & Control Engineering Journal, 2011, 12(2): 99-102.

[2] Hunt Ray. PKI and Digital Certification Infrastructure[J]. IEEE Computer Society, 2013(1): 234-239.

[3] Zalenski R. Firewall technologies[J]. IEEE Potentials, 2012, 21(1): 24-29.

[4] 蒋建春, 马恒太, 任党恩等. 网络安全入侵检测：研究综述[J]. 软件学报, 2000, 11(11): 1460-1466.

[5] 魏广科. VPN 技术及其应用的研究[J]. 计算机工程与设计, 2005, 2(3): 714-715.

[6] 思维世纪公司. 防病毒网关如何帮助你防毒[J]. 计算机安全, 2003, 1(31): 58.

[7] 胡志新, 王英惠, 尹用等. 集成网络功能的 PCI 接口物理隔离卡开发[J]. 计算机工程, 2007, 33(11): 249-250.

[8] David Newman Joel Snyder Rodney Thayer Crying Wolf False Alarms Hide Attacks[EB/OL]. http://www.nwfusion.com/techinsider/2002/0624securityl.html. [2015-09-23]

[9] Vandyke Software™. Survey ShowsHow IT Perceives & Responds to Constantly Changing Security Threates[EB/OL]. http://www.vandyke.com. [2015-8-13]

[10] 唐正军. 入侵检测技术导论[M]. 北京: 机械工业出版社, 2004.

[11] Marc Norton. Optimizing pattern matching for intrusion detection[J]. Sourcefire, 2014, 1(15): 36.

[12] 黄刚. 入侵防御系统关键技术的研究[J]. 网络安全技术与应用, 2008, 1(5): 32-34.

[13] 边歆. IPS 的快跑与慢走—简析 IPS 国内外市场状况[EB/OL]. http://cnw2005.cnw.com.cn/store/detail/detail_feedback.asp?articleId=41643&ColumnId=1144&pg=&view=.[2015-9-18]

[14] VIGNA G, KEMMERER R A.NetSTAT: A network-based intrusion detection system[J]. Journal of Computer Security, 1999, 7(1): 37-71.

[15] 杨昌振. 网络入侵检测原理与技术[M]. 北京: 北京理工大学出版社, 2006

[16] 薛静铎, 朱烈煌, 阎慧等. 入侵检测技术[M]. 北京: 人民邮电出版社, 2007

[17] 绿盟科技. 国产 IPS 问世及 IPS 的发展简述[EB/OL]. http://www.soft6.com/tech/5/57525.html. (2006-9-15) [2015-08-26].

[18] 胡华. 基于 Snort 的网络入侵防御系统的研究与设计[D]. 武汉: 武汉科技大学计算机科学与技术学院, 2010.

[19] 唐谦, 张大方. 基于 Snort 的入侵检测引擎比较分析[J]. 计算机工程与设计, 2005, 26(11):

2884-2885.

[20] 李飞, 甘刚, 陈艾东. 基于 Linux 的入侵防御系统的研究与实现[J]. 计算机应用研究, 2007, 24(9): 102-103.

[21] Advanced Engineering Forum. HIDS and NIDS Hybrid Intrusion Detection System Model Design[C]. New York: New York University Press, 2012.

[22] Yao-Min Chen, Yanyan Yang. Policy Management for Network-based Intrusion Detection and Prevention[J]. Network Operation and Management Symposium, 2004, 1(2): 219-232.

[23] 宋普选, 应锦鑫. 入侵检测技术研究综述[J]. 军民两用技术与产品, 2005, 1(7): 38-40.

[24] 肖竞华, 卢娜. 基于网络的入侵检测系统的研究及实现[J]. 计算机科学与发展, 2007, 17(2): 242-244.

[25] 张晓光. 基于模式匹配的入侵检测系统应用研究[D]. 大连: 大连海事大学软件学院, 2010.

[26] Measuring Technology and Mechatronics Automation 2013 Fifth International Conference on. An improved multi-pattern matching algorithms in intrusion detection[C]. New York: Computer Society Press, 2013.

[27] Andreas Fuchsberger. Intrusion Detection System and Intrusion Prevention Systems[J]. Information Security Technical, 2005, 1(10): 134-139.

[28] Brian Caswell, Jay Beale, James C. Foster, Jeffery Poslumns. Snort 2.0 Intrusion Detection[M]. New York: Syngress. 2003.

[29] 李晓芳, 姚远. 入侵检测工具 Snort 的研究与使用[J]. 计算机应用与软件, 2006, 23(3): 123-124.

[30] Koziol J, 吴溥峰, 孙默等. Snort 入侵检测实用解决方案[J]. 计算机应用与软件, 2005, 22(16): 75-98.

[31] 孙敏, 古晓明, 张志丽. Snort 规则链表结构的改进与仿真[J]. 计算机工程, 2009, 35(11): 120-121.

[32] Dharmapurikar S, Lockwood J W. Fast and Scalable pattern matching for network intrusion detection systems[J]. IEEE Computer Society, 2006, 24(10): 1781-1792.

[33] Computer Security Conference. A pattern matching model for misuse instrusion detection[C]. Baltimore: Williams and Wilkins, 1994.

致谢

时光飞逝，转眼间，两年多的研究生生活也将近尾声了。回顾这两年多的生活，有对于往昔的不舍和留念；想一想今后的生活，又充满了对未知的向往和好奇。但是时光终会过去，于是过去的日子就成为了我们生活中点点滴滴的回忆。

想想自己在读研究生的日子里，有许多的人给与了自己帮助与鼓励，是他们陪伴着自己，让自己一点一点长大，从而取得今天的成绩。

首先要感谢的是我的指导老师，覃桂敏覃老师。她兢兢业业，勤勤恳恳，对于我的论文，从开始选题，到最终定稿，都进行了详细的指导，并且提出了很多宝贵的意见，使我的论文更为充实，结构更加清晰，内容更加合理。没有她的指导，我的论文将很难达到毕业的水平。

其次要感谢的，是我的企业导师武志强老师。我在 H3C 实习的十个月时间里，他不但在技术上给与我无微不至的讲解，在生活中更是为我提出了很多有建设性的意见。他总是在闲暇的时间，跟我谈论许多人生的话题，比如说择业、就业、工作地点、工作环境、发展空间等。并且结合自身的亲身经历，给我许多指导性的建议。这些意见无不令我获益匪浅，是他的关怀和体贴，让我逐步长大，适应了社会的工作，褪去了在校学生青涩的外壳。

然后还要感谢 H3C 的各位同事，在我实习期间，给与了我极大的帮助，无论是生活方面，还是工作方面，无不体现着对我的关怀。

当然还有陪伴我度过研究生生活的各位同学，我们一起学习、娱乐和生活，这份友谊将永远的留在我们心中。

接下来要感谢的就是我的家人，无论什么时候，你们都对我关怀备至，给与我最大的鼓励、包容和支持。

最后感谢所有帮助、关怀和支持我的人们！

西安電子科技大學

XIDIAN UNIVERSITY

地址：西安市太白南路2号

邮编：710071

网址：www.xidian.edu.cn