应用层协议识别算法综述 *)

陈 亮 龚 俭 徐 选
(东南大学计算机科学与工程学院 南京 210096) (江苏省计算机网络技术重点实验室)

摘 要 能够标识出 Internet 上每个流所使用的应用层协议是一系列网络应用的前提和基础。然而随着网络的高速化和协议的复杂化,传统的基于端口识别应用层协议的算法已经不够准确,因此各种新的协议识别算法成为研究热点。本文介绍了协议识别问题的几个基本概念,将目前正在使用或研究的协议识别算法总结为三类并分析了各自的优缺点及关键技术和难点,最后指出了两个进一步研究的方向。

关键词 Internet 流量,识别,应用层协议,算法

A Survey of Application-Level Protocol Identification Algorithm

CHEN Liang GONG Jian XU Xuan
(School of Computer Science and Engineering, Southeast University, Nanjing 210096)
(Jiangsu Province Key Laboratory of Computer Networking Technology)

Abstract The ability to accurately identify the Internet traffic associated with different application-level protocols is essential to a broad range of network applications. Due to high speed of network and complexity of new protocols, traditional port-based application-level protocol identification method is becoming much more inaccurate. Therefore, a variety of protocol identification algorithms become a research hotspot. In this paper, several basic concepts of application-level protocol identification are introduced first; then the identification algorithms being used or researched now are classified into three categories; their main difficulties, advantages and disadvantages are also analyzed; finally two future research directions are presented.

Keywords Internet traffic, Application-level protocol, Identification, Algorithm

1 引言

能够准确地识别 Internet 上每个流所使用的应用层协议对于网络管理员、研究员以及服务提供商、用户都具有重要的意义,其是研究区分服务、QoS、入侵检测、流量监控、计费管理以及用户行为分析的前提和基础。但是,随着 Internet 底层环境和上层应用的发展,下面两个问题变得越来越突出:一、识别的可行性问题。当前 Internet 主干带宽已升至 40 Gb,CERNET 主干也已达到 10 Gb。这样高速高带宽的网络 1 分钟的流量就达到上百万个流。如何设计算法以处理数量庞大且不断增长的网络流量,使协议识别可行,是识别算法需要面对的首要问题。二、识别的有效性问题。在过去的网络环境中,绝大部分网络流量被 Web、FTP、SMTP、Telnet 等协议所占据。而近年来,应用层协议的形式与种类都较过去更加复杂,传统协议的流量在总流量中的比重越来越少,相反,P2P、流媒体、网络游戏等新应用协议不断涌现,且已经占据了网络流量的 60 %以上[1,2]。更重要的是,这些新协议的规范往往不公开并且不遵守默认固定端口的约定,因此,如何能够正确的识别这些复杂的协议也是现在协议识别算法必须解决的问题。

正是由于 Internet 飞速发展所带来的新问题,各种应用层协议识别算法成为现在研究的热点,并且取得了一系列的成果。本文总结了协议识别问题的一些基本概念,综述了现在正在使用或研究的协议识别算法的种类,并分析了每类算法的优缺点。文章第 2 部分首先介绍了协议识别所涉及到的几个基本问题;第 3 部分分别介绍了每类算法并分析了各自的优缺点;最后对全文进行总结并提出了未来的研究方向。

2 协议识别的基本概念

概念 1 流 ——指在某一段固定时间间隔内通过网络上一个观测点的 IP 报文集合。属于一个特定流的所有报文有一些相同的属性[3]。

应用层协议识别的对象不是单个报文,而是将"流"作为一个整体考虑。

概念 2 协议识别 ——标识出网络上每个流所使用的应用层协议,其是基于使用类型的流分类的延伸和精化。在基于使用类型的流分类问题中,每个类别可能包含某些属性类似的多种协议,但协议识别问题必须对流进行更精细的分类,使得每个类别中的流只使用一种应用层协议。

概念 3 流分类 ——指利用流以及流中报文的某些信息将网络上的流分成既定的若干类别(如长流/短流,快流/慢流,或者各种使用类型的流),其是报文分类的扩展[4]。

概念 4 解决协议识别问题的基本思想 ——从本质上看,协议识别问题是多元统计学中的判别分析在实际中的应

*) 本文受国家 973 计划课题(2003CB314804)、教育部科学技术重点研究项目(105084)和江苏省网络与信息安全重点实验室(BM2003201)资助。陈 亮 博士生,主在研究方向为网络行为学;龚 俭 工学博士,东南大学计算机系教授,博士生导师,主要研究方向包括网络管理、网络行为学、网络安全等。

用。首先根据所选择的 n 维流信息将流分为 k 个类别,每个类别对应一个协议,对于新到来的流,计算其自身的 n 维流信息值,根据结果将其划分到相应的类别中,给出类别号即协议名。从理论上说,流中每个报文的任意字段或流传输过程中的任何特性都可以作为一维的流信息即协议识别的依据。但实际使用中,如何选择最有效的流信息维度是面临的最大困难。

概念 5 算法评价原则——一般来说,评价一个识别算法好坏的标准有五个:

1) 能够识别的协议的数量,特别是新出现的协议的数量(各种 P2P 协议、流媒体等)。

2) 准确性:一个好的算法应该有较高的准确性,较低的误报率和漏报率。

3) 时间复杂度:为了能够处理高速网络,一个好的算法必须有较低的时间复杂度。

4) 空间复杂度:为了能够处理高带宽,一个好的算法必须有较低的空间复杂度。

5) 更新难易度:这包括两个方面(1)某协议的规范发生变化时,一个好的算法应该尽可能少地被修改和重新配置,以便能继续识别该协议;(2)某协议已不再使用或有新协议出现时,一个好的算法也应该尽可能少地被修改和重新配置,以便删除或添加该协议的识别。

以上五个标准往往互相制约。例如,若增加识别协议的数量就会增加时间和空间复杂度,若减少时间复杂度就会增加空间复杂度或降低准确性,等等。在实际评价一个算法好坏时,要根据实际的背景在这五个指标中做出权衡。在下一部分中,我们将介绍现在正在使用或研究的几类协议识别算法。

3 几类协议识别算法

3.1 基于端口识别协议

原理——传统的应用层协议识别算法只利用了端口号一维信息,其根据各个应用层协议在 IANA[5] 中注册的端口号来标识协议。例如,若某个 TCP 流使用了端口号 80、8080 或 443,则将其标记为 Web 流量。

评价——协议数量:算法所能识别的协议数量为在 IANA 中注册端口号的协议的数量,但是由于新出现的协议都不在 IANA 中注册其端口号,算法所能识别的协议在总协议数量中所占的比重越来越少。准确性:文献[6,7]中详细论述了导致基于端口算法失效的原因。正是由于这些原因,基于端口识别协议的准确性已经低于 50 %[8],算法的错误率高于正确率。时空复杂度:由于算法简单,所需信息少,端口算法的时空复杂度是所有算法中最低的。更新:由于新协议均不在 IANA 中注册端口号,实际中必须用实验等方法获得待识别协议的端口号,更新较复杂,并且,若协议使用动态端口,则算法不可行。

3.2 基于负载识别协议

为了提高识别的准确性,2002 年至 2004 年间有很多研究利用报文的负载部分识别应用层协议[9~11],并且目前绝大部分厂商所研发的协议识别的设备也均采用此类算法。

原理——基于负载的算法仍是一个一元判别问题,其需要事先详细分析待识别的应用层协议,找出其交互过程中不同于其他任何协议的字段,作为该协议的特征。在识别的过程中,该类算法检查流中每个报文 TCP 首部之后的负载部分,若匹配到某协议的特征,则将该流标记为相应的协议。基于负载的算法不仅能识别出使用单一连接进行通信的协议,而且能够识别出如 PASV FTP、流媒体等使用多个连接、动态端口进行通信的协议。在这些协议中,数据传输所使用的端口是在事先建立的控制连接中协商的。基于负载的算法检查控制连接中的每个报文,找出协商得到的端口号,并以此端口识别数据连接。

评价——协议数量:从理论上来说总可以通过分析协议规范和实际交互的报文得到协议的特征,因此只要有足够的工作量,该类算法可以识别所有的协议。准确性:负载算法的准确性是目前所有算法中最高的,文[9~11]中所研究的算法的误报率均小于 10 %。时空复杂度:由于需要逐报文的匹配所有协议的特征以及额外的存储报文的负载部分,该类算法的时空复杂度是目前所有算法中最高的,并且随着待识别协议数量的增长而增长。更新:使用基于负载的算法需要不断地跟踪待识别协议的发展,当协议规范发生变化或者新协议出现时,寻找特征的工作必须重新进行,工作量非常大,更新困难,因此,该类算法通常只被用在需准确识别数量较少的协议时,且需要有相当的工作量。

3.3 基于测度识别协议

上述两类算法由于固有的不可克服的缺陷,已经逐渐不被研究。而从 2004 年开始至今,基于测度识别协议的各种算法逐渐成为研究的热点[12~14]。

原理——基于测度识别协议的算法利用协议规范的不同所造成的流测度的差异区别各个协议。例如,Web 流一般为短流小报文,而 P2P 流一般为长流大报文。基于测度的算法要求事先有标准的训练集可用,即要用已按各个协议分类的报文集合来训练识别器,使其在使用的过程中根据已知的标准答案和新计算的流测度,按照某种判别算法得出当前流所属的类别,即所使用的协议。

评价——协议数量:从理论上说,每种协议由于其规范不同,测度总会有所不同,因此,基于测度的算法理论上也可以识别所有的协议。但是,当前所研究的算法都只能将流分成几大类(例如 bulk transfer, single and multiple transaction, interactive traffic 等等),没有达到识别协议的最终目的。准确性:文[12~14]所实现的算法的准确性都在 70 %左右,高于端口算法,低于负载算法。时空复杂度:由于要进行多元判别分析,该类算法的时空复杂度高于端口算法,但是远低于负载算法。更新:基于测度识别协议的算法不需要知道协议的具体细节,因此当协议规范的细节发生变化时不需要更新相应的算法。当有待识别的新协议时,需要用新协议的标准集训练识别器,算法更新的难度要稍高于端口算法,远低于负载算法。

由上面的分析可以看出,基于测度识别协议的算法有很多优点。但是,该类算法的研究还很不成熟,目前主要存在两个难点:一、如何选取测度。和前两类识别算法不同,基于测度的算法是一个多维空间判别问题。文[15]中为流和流中报文定义了 249 种测度,显然实际不可能使用全部这些测度,因此,选择尽可能少的哪些测度是影响算法效果的一个重要因素。二、如何选择判别算法。多元判别问题需要一个判别方程,根据新样本在多维空间的取值决定其应属的类别。如何为算法选择一个判别方程,使其在测度集合一定的情况下达到最好的判别效果是影响算法的另一个重要因素。

3.4 综合算法

为了弥补上述三类算法各自的缺点,一些研究试图将它们结合使用:文[6]首先通过测度将流分类,然后在每一类中寻找代表端口以标识协议。虽然该方法可以提高基于端口识别协议的准确性,但是如果端口的选择完全随机,方法将失效。文[8]和[16]结合了端口和负载的算法以提高识别的准确性。特别是文[8]中的算法迭代地使用九个子方法,识别的准确性达到 99 %,但是其时间和空间复杂度过高,不可能应用于实际。

小结 新的应用层协议识别算法的研究是现在计算机网络的一个热点研究方向,其必须有高准确性和高效率以克服由于新复杂协议及高速网络给协议识别所带来的困难。本文将目前正在使用或研究的协议识别算法归纳为三类并分析了它们的优缺点和难点:传统的基于端口识别协议的算法简单,但是随着网络的不断发展其准确性已经低于 50 %,不可用;基于负载识别协议的算法具有 90 %以上的准确性,但开发和维护需要很大的工作量,且该类算法的时空复杂度过高,限制了其在实际环境中的使用;刚出现的基于测度识别协议的算法不需要知道协议的具体细节,易于扩展与维护,时空复杂度都远低于基于负载的算法,但是目前算法还不成熟,识别不细,且准确性只达到 70 %左右。鉴于三类算法各有优缺点,未来的研究可以向两个方向发展:一、研究基于测度的算法中如何挑选测度及判别方程并证明算法的最优;二、进一步研究如何更好地结合各类算法,使其优势互补。例如,可以结合基于测度和负载的算法:先使用基于测度的算法对流进行粗分,然后在每一类中使用基于负载的算法标识出协议。这样既可以避免在大范围中使用基于负载的算法以提高速度,又可以增加基于测度的算法的精度和准确性。

参 考 文 献

1 Sen S, Wang J. Analyzing Peer-to-Peer Traffic across Large Networks[C]. IEEE/ACM Transactions on Networking. NJ: IEEE Press, 2004. 219～232
2 Plissonneau L, Costeux J L, Brown P. Analysis of Peer-to-Peer Traffic on ADSL[J]. In PAM 2005, volume 3431 of LNCS Springer, 2005. 69～82
3 RFC3971. Requirements for IP Flow Information Export (IPFIX)[S]
4 Hifn, Inc. Why You Need Flow Classification, Technical White Paper[EB/OL]. http://www.hifn.com/docs/a/WP-0001-00-Why-Your-Need-Flow-Classification.pdf. September 2001
5 IANA. http://www.iana.org/assignments/port-numbers[S]
6 Kim M S, Won Y J, Hong J W K. Application-Level Traffic Monitoring and an Analysis on IP Networks[J]. ETRI journal, 2005, 27(11): 22～42
7 Roughan M, Sen S, Spatscheck O, Duffield N. Class-of-service mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification. In: Proc. ACM. SIGCOMM IMC 2004, Taormina, Italy, Oct. 2004. 135～148
8 Moore A W, Papagiannaki K. Toward the Accurate Identification of Network Applications[C]. In PAM2005. Boston, MA, 2005. 41～54
9 Kang H J, Kim M S, Hong J W-K. A Method on Multimedia Service Traffic Monitoring and Analysis[J]. In DSOM 2003. Lecture Notes in Computer Science 2867. Heidelberg, Germany, Oct. 2003. 93～105
10 van der Merwe J, Caceres R, Chu Y-H, Sreenan C. mmdump-A Tool for Monitoring Internet Multimedia Traffic[C]. ACM Computer Communication Review, Oct. 2000, 30: 48～59
11 Sen S, Spatscheck O, Wang D. Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures[C]. In: Proceedings of the 13th international conference on World Wide Web. NY: ACM Press, 2004. 512～521
12 Zander S, Nguyen T, Armitage G J. Self-Learning IP Traffic Classification Based on Statistical Flow Characteristics[C]. In: PAM2005. Boston, MA, 2005. 325～328
13 Zuev D, Moore A W. Traffic Classification Using a Statistical Approach[C]. In: PAM2005. Boston, MA, 2005. 321～324
14 McGregor A, Hall M, Lorier P, Brunskill J. Flow Clustering Using Machine Learning. Techniques. In: PAM2004, France, April 2004. 205～214
15 Moore A W, Zuev D, Crogan M. Discriminators for use in flow-based classification[R]. Department of Computer Science, Queen Mary, University of London, August 2005
16 Choi T S, Kim C H, Yoon S H, Park J S, Lee B J, Kim H H, Chung H S, Jeong T S. Content-aware Internet Application Traffic Measurement and Analysis[C]. In: Proc. NOMS, 2004. 511～524

(上接第 72 页)

抽象接口,将网卡与文件数据封装在一起,对外提供统一接口,实现无差别访问。另外,借鉴 Windows 消息机制思想,在数据包到达后,由数据包捕获器向各个处理模块发送消息,通知各模块处理新到的数据包。运用设计模式及消息传递机制,系统的结构简单清晰,易于维护。第二,读写磁盘的速度系统,处理的速度存在较大差异,而本系统对时间要求较高。为解决数据包存储的速度瓶颈问题,本系统采用缓存技术。数据包到达时,不是立即存储到磁盘中,而是先存储在内存里事先开辟出来的缓冲区中。当缓冲区数据满时,再一次性写入磁盘中,这能显著地减少磁盘读写次数。另外,结合多线程技术,另开一线程,专门负责数据写操作。缓冲技术结合多线程技术的运用,使得系统实时性得到显著的改善[6]。

结束语 网络数据包捕获及协议分析技术是防范网络攻击的基础,良好的底层捕获实现为网络入侵检测奠定坚实的基础[7]。本文运用 UML 建模技术,设计数据包捕获及协议分析系统的总体结构和功能模块,将 UML 应用到数据包捕获及协议分析系统的设计与开发中,使开发者能够准确理解系统各部分之间的内在联系,加快软件开发过程,提高软件开发水平和软件质量,对网络攻击防范软件开发具有借鉴意义。

参 考 文 献

1 周治平,夏娟,纪志成. 基于 UML 的实时系统设计方法的分析与比较[J]. 计算机工程, 2005, 31(13): 99～101
2 王强,贾素玲,等. UML 系统分析设计[M]. 北京:高等教育出版社, 2005
3 马蔷,杨南海,等. UML 软件开发[M]. 北京:电子工业出版社, 2005
4 姚淑珍. UML 和模式应用 - 面向对象分析与设计导论[M]. 北京:机械工业出版社, 2002
5 Alexandrescu A. Modern C++ Design(影印版)[M]. 北京:中国电力出版社, 2003
6 高光勇,谢志恒. 网络入侵检测系统中的包截获和报文解析[J]. 齐齐哈尔大学学报, 2004, 20(4): 47～50
7 刘文涛. Linux 网络入侵检测系统[M]. 北京:电子工业出版社, 2004