文章编号:1001-9081(2001)05-0050-03

具有高速过滤算法的 IP 防火墙

申震生,龚向阳,王文东,阙喜戎
(北京邮电大学 程控交换与通信网国家实验室,北京 100876)

摘 要:IP 防火墙是一种具有报文过滤能力的安全设备,在实现中需要解决的主要问题是如何提高包过滤的性能,特别是在规则数据库的规模较大,网络传输速率较高的情况下。本文分析了报文过滤的性能问题,并给出了一种简单有效的解决方案,利用改进的递归流分类(RFC)算法来实现 IP 防火墙中的数据包过滤。

关键词:IP 防火墙;包过滤;递归流分类
中图分类号:TP393.08 **文献标识码**:A

1 前言

近年来随着网络安全受到越来越多的重视,防火墙已经成为大部分网络中必不可少的安全设备。防火墙使用的技术一般有两种类型:包过滤和应用代理[1]。比较常见的方式是同时使用这两种技术构成综合性的防范系统,如图 1 所示的屏蔽子网体系结构。两台屏蔽路由器,又称 IP 防火墙,使用包过滤技术控制对内部子网的访问;同时,在两台 IP 防火墙之间,被称作非军事区(DMZ)的地方,另有一台或多台堡垒主机,它们使用应用代理技术控制内外网络之间的数据通信。

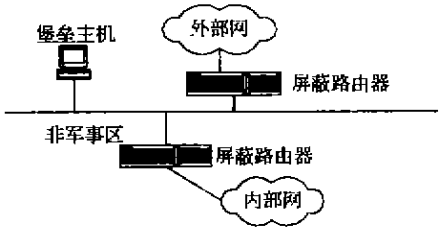

图 1 屏蔽子网防火墙系统

在上面的例子里,IP 防火墙在外部网和内部网之间担当了基于规则的路由器的角色。它维护了一个包过滤规则数据库,根据规则对不同的数据包采取不同的动作,如转发,丢弃,或者利用 NAT(网络地址映射)进行转换。大部分过滤规则不仅要处理 IP 包头,还要处理传输层(TCP/UDP)包头信息,因此 IP 防火墙已经从单纯的基于目标的第三层交换变为更为复杂的第四层交换[2]。对每个到来的数据包,IP 防火墙从中提取关键信息,送去与规则逐条比较,直至与某条规则相匹配,或者到达规则库的末尾为止。但是,这种简单操作可能会导致过滤算法变得效率低下。特别是在高速网络中,如果规则库的规模很大,低效的过滤算法将使 IP 防火墙将成为网络性能的瓶颈。

2 高速防火墙的需求分析

经过对高性能过滤算法和 IP 防火墙环境的研究,在高速网络中要想兼顾 IP 防火墙的安全和效率,必须满足以下要求:

1) 在高速网络中,过滤算法的速度应当足够快,以避免使用额外的存储队列或采取丢弃算法来处理涌来的数据流。随着千兆比以太网越来越流行,将来的防火墙完全有可能工作在这样的网络环境中。在 IP 包的平均大小为 252 个八位组的条件下,过滤算法需要具有每秒至少处理 0.5 兆数据包的能力。

2) 理想的过滤算法应当能够对任意层的数据域进行匹配,包括数据链路层,网络层,传输层,甚至应用层的头部。

3) 过滤算法应当能够支持匹配规则所具有的各种表示方法,如规则域可以是确定值,前缀,整数范围,或者无限制。

4) 应用于网络安全环境中,过滤算法应当具有很好的实时性,能够立即对规则的改变做出反应,以便及时阻挡来自外部网络的攻击。

到目前为止,大部分 IP 防火墙在实现时采用的是线性搜索的策略,或者同时使用 cache 技术来提高过滤性能,如 ipfilter-3.2.10,ipchains-1.3.9 等等。将最近使用的规则存入 cache,使用内容匹配方式查找,给 IP 防火墙效率带来了一定的提高。但是当防火墙的规则很多,建立的连接变化频繁时,这种好处是有限的,它的效率在很大程度上决定于防火墙所在网络的流量。

近年来,在 IP 高速路由技术领域出现了一些流分类算法,如 Grid of Tries[2], Tuple Space Search[3], Recursive Flow Classification[4]等,设计之初主要是为了满足高速路由器中 IP 流分类的需要,但是从本质上来说,流分类和包过滤并无区别,它们也可以应用于 IP 防火墙。GoT 算法比较适合于二维前缀匹配,查找速度快,空间要求小;但是扩展到多维匹配后,查找效率大大降低。TSS 算法将很大的规则空间划分为多个较小空间的集合,在较小的规则表中进行哈希查找,提高效率;但是它在最坏情况下的查找效率与线性搜索相同,应用受到限制。RFC 算法的搜索过程非常简单快速,满足了高性能防火墙的大部分要求,但是预处理需要大量时间空间,因此我们对它做了一些优化后引入,使之更加适合 IP 防火墙的特定环境。

收稿日期:2001-02-19(修改稿)
作者简介:申震生(1976-),男,硕士研究生,主要研究方向:宽带通信网,网络安全.

3　RFC 算法描述

从原理上说,包过滤问题可以看作一种映射动作,即将数据包头部 S 比特的关键信息映射为 T 比特的类标识(classID),这里 T = logN,且T << S,N 为过滤规则的数目。如果能够预先计算出所有 2S 个不同的分组头所对应的 classID,存在一个线性表中,搜索时只需要一步查表(一次内存访问)就能得到结果。尽管这种实现方法看起来简单快捷,但是它需要 $O(2^S)$的内存空间,实际中 S 往往较大(100 左右),并不可行。RFC 算法的基本思想就是分步完成这样的映射,每步完成类似的操作,即进行一次空间压缩,把一个较大的等价类集合映射为较小的等价类集合,如图 2 所示。使用这种方法,虽然分几步才能得到最终的结果,但是消耗的内存减少了许多。

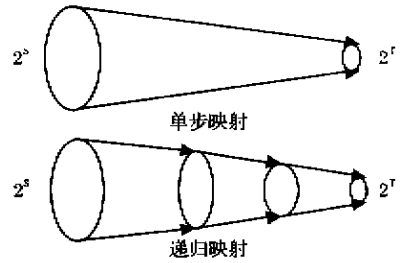

图 2　RFC 算法的基本思想

3.1　算法简介

RFC 算法利用了实际应用中规则库的结构特性和冗余度,它由两个阶段组成,即预处理阶段和搜索阶段。RFC 对于查找过程中可能产生的所有中间结果进行预处理,当分组到达时,可以直接利用预处理的结果,从而简化搜索过程,其代价是增加了预处理过程在时间和空间上的复杂度。

预处理阶段如图 3 所示。

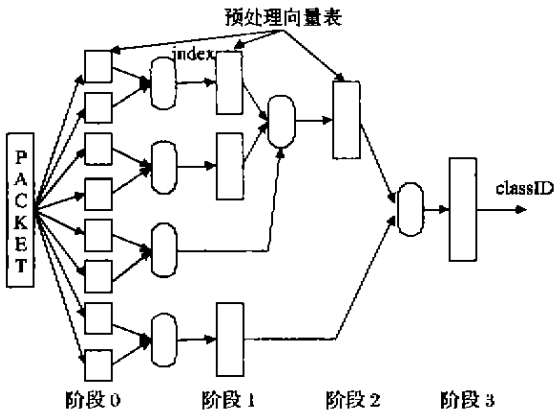

图 3　RFC 算法中数据流处理方式

1) 在阶段 0,从规则库中读取规则,规则域的值被投影到数轴上,形成一组无重叠的间隔,对每个间隔赋予一个等价集标识(eqID),即唯一的比特向量,从而构成比特向量表。

2) 其后的阶段 1 至阶段 3 是一组递归的操作,对每两个源向量表,分别对这两表中的每一项进行按位与操作,得到所有可能的新的比特向量,并把它们添加到目的表中。每次生成的目的表长度必然比两个源向量表长度之和要短,从本质上说这是一棵减少树的形成过程(如图 3)。

3) 每个阶段生成的目的表是一个预处理向量表,直至最后得到类标识(classID)。

相比之下,搜索阶段很简单,因而具有较高的效率。对于每个到来的分组,提取头部的关键域,分别在相应的向量表中找到对应项,然后按照减少树合并的结构,把得到的值进行简单计算,以所得结果作为下阶段数组索引在下一级表中查找。最后一个阶段到达减少树的根部,从表中查到的数值就是所匹配规则的规则号,从而得到规则对应的动作。

3.2　算法优化

从下一节的实现结果可以看出,RFC 在搜索阶段具有速度快和受规则数影响较小的优点,但是 RFC 的搜索阶段的优异性能是以预处理过程的复杂化为代价的,在实际应用中 RFC 的预处理阶段占用了大量的时间和内存空间,无法对规则的变化做出及时的反应。对 RFC 算法做出一些优化,会使它更加适合 IP 防火墙的安全环境。

1) 简化预处理过程

在预处理阶段,需要进行长度为 O(n)的比特向量之间的按位"与"操作(这里 n 为规则数),大量的比特逻辑与运算一般会耗费较长的时间。因为按字(word)访问内存和按比特访问内存耗用的时间相同,而且在通常情况下比特向量中往往有较多的"0",可以首先以字为单位来访问内存。在向量中以 16 比特为一个单元,首先测试这个字是否为 0,若是,可以直接跳往下一个字;否则就逐个访问这 16 比特做逻辑与运算。应用这种策略,RFC 预处理的时间可以缩短许多。

RFC 算法利用了实际规则库中的结构化和冗余性,故算法所需的空间大小与规则总数及规则库的具体结构特性有关。在预处理的前几个阶段中空间的收敛速度不够快,会造成空间需求过大。采用规则合并的方法,把仅有一个匹配域不同而其他域都相同,并且动作也相同的规则合并在一起,从而减少待搜索的规则数,可以减少内存空间的消耗。

2) 省略不必要的预处理过程

把预处理的结果以数据文件的形式存储下来,当防火墙重新启动,RFC 算法初始化的时候,首先检测规则库和相应的数据文件的创建时间。如果规则文件从上次防火墙重启以来并无变化,那么无需对规则库作预处理,只需直接从数据文件中读取结果即可。

3) 对报文过滤方式的调整

RFC 算法的预处理阶段耗时很长,应用于高速流分类时不可避免会受到限制,即不适合改动频繁的规则库。依照目前的发展趋势,防火墙作为一种使用静态,被动保护方式的安全产品,正在向自适应,动态安全和快速反应的综合性防范系统过渡。比如将 IP 防火墙与入侵检测系统(IDS)结合应用于网络安全中,一旦入侵检测系统检测到外部网的攻击,将立即生成一条新规则并插入防火墙规则库中以阻拦攻击。在这种情况下,新规则应当马上生效,对攻击做出快速反应,但是 RFC 算法太长的预处理时间大大降低了网络的安全性能。因此在 IP 防火墙中可以将耗用内存小,反应时间快的哈希查找应用于规则库的改变部分。它利用的是哈希化的线性查找方式,因为没有预处理阶段,新规则可以立即产生效果。同时由于哈希查找涉及的规则只是规则库内发生变化的部分,不会对 IP 防火墙的性能造成很大的影响。

4　RFC 算法性能

对 RFC 算法性能方面的一系列测试验证了 RFC 算法的

效率。当在 350MHz 的 Pentium－II 微机上实现 RFC 算法，规则库数目为 1000 条时，RFC 算法每次查找大约耗时 1 微秒。相关的性能结果如表 1 所示。

表 1 RFC 查找性能结果

规则数	预处理时间(分)	搜索时间(微秒)
1,000	1	0.98
2,000	6	1.37
3,000	65	1.54

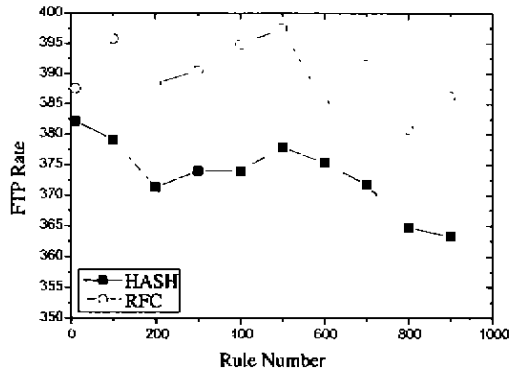

图 4 RFC 与线性查找的性能比较

当 RFC 算法应用于 IP 防火墙中时，同样表现出高效率。在规则库的大小从 10 条变化至 1000 条时，我们用 FTP 跨越防火墙传输文件，并记下传输速率。单独使用RFC算法和单独使用哈希算法(线性搜索方式)得到的 FTP 速率如图 4 所示。从中我们得出了如下结论：在相同的网络和规则库条件下，RFC 算法具有比哈希查找(线性搜索)更高的效率；尤其在规则数目很多时，效率差别较为明显。RFC 算法的效率与规则库的大小关系较小，当规则库变大时，RFC 性能的降低程度要比哈希查找低的多。

5 结束语

随着 Internet 的发展，为不同数据流提供不同服务的 DiffServ 体系结构将是未来网络的发展趋势，防火墙只是其众多应用中的一种。本文介绍了一种用于高速 IP 防火墙的简单快速的分类算法，RFC 算法，并对其在实际中的应用做出了一些优化。相信随着宽带网络的发展，高性能的防火墙将会在其中起到非常重要的作用。

参考文献

[1] Chris Hare, Karanjit Siyan. Internet 防火墙和网络安全(第二版)[M]. Prentice Hall, May 1998.

[2] V. Srinivasan, G. Varghese, S. Suri, M. Waldvogel. Fast and Scalable Layer Four Switching[A]. Proceeding ACM SIGCOMM '98[C], September 1998.

[3] V. Srinivasan, S. Suri, G. Varghese. Packet Classification using Tuple Space Search[A]. Proceedings of ACM SIGCOMM '99[C], September 1999.

[4] Pankaj Gupta, Nick McKeown. Packet Classification on Multiple Fields[A]. Proceeding ACM SIGCOMM'99[C], September 1999.

(上接第 49 页)

别的说明，这里主要考虑通过"修改"来破坏完整性的情况(尽管其发生的可能性比较小)。虽然 FTP 协议的安全扩展[5]提供了对 FTP 协议的完整性和机密性保护，但是在 FTP 客户与 FTP 服务器之间插入了代理型防火墙后，安全扩展中的用于建立安全集合(Security Association)的关键命令 ADAT 不能够被转发，因而也就难以获得它提供的安全服务。在寻求其它的解决方案前，先分别对 FTP 的控制连接和数据连接的完整性和机密性问题进行讨论。首先来看传输命令的控制连接，其实上一节中用于防止 TCP 会话劫持攻击的 AH 安全协议已经在提供认证的同时提供了完整性保护，而对于命令的机密性我们觉得并没有很大的必要进行保护；其次再来看传输文件的数据连接，文件的完整性和机密性都需要采取一定的措施来保护。我们已经知道，ESP 作为一个安全协议可以同时对数据包提供完整性和机密性保护，因而在防火墙上实现 ESP 是一个比较完善的解决方案。

与 3.3 中采用 AH 所面临的问题一样，由于 ESP 的实现同样处于 IP 层，属于操作系统的内核部分，其运行还需实现相关的密匙分发协议，同时需要通信两端都在 IP 层实现 ESP，所以在短期内我们不太可能完全采取这种方案。

考虑在 FTP 协议之上保护数据文件的完整性和机密性，通常校验和是一种非常简单和高效的验证数据文件完整性的机制，不过从安全角度看，诸如算术校验和、多项式校验和(CRC)这些类型的校验和都是很弱的，更安全的方法是使用单向散列函数来计算数据文件的摘要以用作校验；至于机密性，可以通过对数据文件进行加密来提供。这些对文件的处理对通信协议(FTP)透明，收端若拥有密匙，在接收后只要对数据解密即可获得原始文件，并进行校验。这种方法的安全性能完全取决于算法的强度以及密匙的选择。

4 结论

增强的用户认证机制应用在代理型防火墙上能够提供给防火墙更大的安全性和方便性，而一次性口令认证作为一种强认证机制，其优点是口令不重用，防止被动攻击和重用攻击的效果显著，因而各种基于 OTP 的代理型防火墙越来越多地被使用。强认证并不是万能的，本文的焦点在于对该类防火墙的攻击与防止，从字典攻击、竞争攻击、TCP 会话劫持攻击以及完整性和机密性四个部分加以阐述，为进一步提高基于 OTP 的代理型防火墙的安全性能打下基础。

参考文献

[1] N. Haller, C. Metz, P. Nesser and M. Straw. A One－Time Password System[S]. RFC2289, IETF, February 1998.

[2] L. Joncheray. A Simple Active Attack Against TCP[A]. Proceedings of the Fifth Usenix UNIX Security Symposium[C], Salt Lake City, April 1995.

[3] S. Kent, R. Atkinson. Security Architecture for the Internet Protocol [S]. RFC2401, IETF, November 1998.

[4] B Fraser. Site Security Handbook[EB/OL]. http://info.internet.isi.edu/in－drafts/files, July 1996.

[5] M Horowitz, S. Lunt. FTP Security Extensions[S]. RFC2228, IETF, October 1997.