一种针对规则集不一致性的测试数据包选取算法

李 林 卢显良
(电子科技大学计算机科学与工程学院 成都 610054)

摘 要 在防火墙规则集正确性测试中,现有的测试数据包选取算法大多随机选取数据包和从规则顶点选取数据包。然而,这种做法忽略了存在规则不一致性的区域,从而导致不能检测出所有因规则不一致性而产生的配置错误。针对这一情况,提出了一种针对规则集不一致性的测试数据包选取算法。该算法以两条规则为基本单位,计算其不一致性区域。算法不但从规则顶点选取数据包,而且从规则集不一致性区域选取数据包。测试表明,与常见测试数据包选取算法相比,该算法只需增加少量测试数据包,就能检测出所有因规则不一致性而带来的配置错误。

关键词 规则不一致性,测试数据包,防火墙,安全漏洞
中图法分类号 TP393 **文献标识码** A

Test Packets Choice Algorithm Aiming at Filter Conflicts

LI Lin LU Xian-liang
(Department of Computer Science of UESTC, Chengdu 610054, China)

Abstract Because of firewall filter conflicts, filters may not be in accordance with administrators' meaning so that this leads to security vulnerabilities. Therefore we need correctness test to solve this problem. Most of the current test packets choice algorithms choose packets at random or from the apex of filters in the correctness test. However these methods neglect the areas that contain conflicting filters and hence cannot detect all error produced by filter conflicts. This paper presented a test packets choice algorithm aiming at filter conflicts to address this problem. The algorithm treats two filters as the basic processed object and computes their area that contains conflicting filters. We not only choose test packets from the apex of filters but from the areas that contain conflicting filters as well. Compared to current test packets choice algorithms, the algorithm proposed by this paper can detect all error produced by filter conflicts with adding only a little packets. This paper proves the algorithm and experiments verify its good performance.

Keywords Filter conflicts, Test packets, Firewall, Security vulnerabilities

1 引言

报文分类技术是路由器、防火墙等许多网络设备的基本技术之一。该技术的原理是:根据预先设置的规则集,检查数据包的相关字段,例如协议号、源IP地址、目的IP地址、源端口号以及目的端口号等信息,以判断数据包匹配哪条规则。最后,按照规则所定义的动作处理数据包。在报文分类过程中,可能出现这样一种现象,即一个数据包能同时匹配两条或以上的规则,且规则处理动作不一致。通常,这些规则被称为不一致规则。常见的处理方式是给每条规则赋予不同的优先级。在不一致的规则中,按照优先级高的规则所定义的动作处理数据包。

近年来,随着防火墙规则数目不断增多,有效管理这些规则变得越来越困难。大型企业级防火墙通常包含有数百条规则,而在这类规则集中,通常都存在着许多不一致规则。面对这样的规则集,管理员即使只完成一些最基本的任务,诸如弄清每条规则的含义以及规则间的关系,也不是一件轻松的事。因此在这种情况下,规则不一致性很容易导致规则配置错误。下面通过一个例子来加以说明。

假设某二维规则集中包含有两条规则:第一条,允许192.168.2.55到192.168.2.160中的节点访问192.168.10.97到192.168.10.189中的节点;第二条,禁止192.168.2.10到192.168.2.230中的节点访问192.168.10.130到192.168.10.150中的节点。假设管理员的本意是要禁止192.168.2.10到192.168.2.230中的节点访问192.168.10.130到192.168.10.150中的节点,而允许192.168.2.55到192.168.2.160中的节点访问192.168.10.97到192.168.10.129中的节点以及192.168.10.151到192.168.10.189中的节点。由于管理员没有察觉到这两条规则是不一致的,错误地给第二条规则赋予了一个较低的优先级,使得192.168.2.55到192.168.2.160中的节点依然可以访问192.168.10.130到192.168.10.150中的节点,从而造成潜在的安全漏洞。

从上述例子可以看出,由于规则间存在不一致性,导致了管理员配置的规则集有可能不能正确表达其配置意图,从而

到稿日期:2010-10-11 返修日期:2011-02-27 本文受信息产业部生产发展基金资助。
李 林(1981-),男,博士,讲师,主要研究领域为网络安全,E-mail:lilin@uestc.edu.cn;卢显良(1944-),男,教授,博士生导师,主要研究领域为计算机网络、操作系统。

造成潜在的安全漏洞。显然，要解决这一问题，必须对规则集进行正确性测试。规则集正确性测试的目的，是判断某个给定的规则集是否符合管理员配置意图。而目前最常见的测试方法是：首先按照某种算法选取测试数据包，并由管理员提供这些数据包的预期处理动作，即放行或者丢弃；然后使用报文分类模拟器处理以上测试数据包，并记录每个数据包的处理动作；最后将上述记录结果与事先由管理员输入的预期结果进行比对，若发现二者不一致，则说明存在规则配置错误。从上述测试方法可以看出，如何选取测试数据包，即如何设计测试数据包选取算法，是一个关键点。而现阶段有关规则集正确性测试的研究，也集中于此。

目前，有关防火墙测试方面的研究，主要集中在功能测试、性能测试、抗攻击能力测试等几个方面，只有少量的研究针对规则集正确性测试。

在防火墙测试方面的研究工作中，文献[1]对各类防火墙测试方法进行了分析比较，指出了各自的应用环境以及优缺点。

文献[2]设计了一种模拟防火墙以及网络环境的工具。该工具能自动生成针对一系列安全漏洞的测试用例。

文献[3]设计了一种分析工具来模拟防火墙的行为模式。该工具有利于管理员及早发现配置错误，但不能找出所有因规则不一致性而带来的漏洞。

文献[4]分析了分布式系统中防火墙安全性同性能之间的关系，并提出了一些防火墙性能测试和安全性测试的方法。

文献[5]提出了一种基于数据流的防火墙漏洞分析框架。该框架对防火墙漏洞进行了分类，并据此提出了不同的抗攻击能力测试方法。

文献[6]提出了一种针对防火墙漏洞的测试方法。该方法使用两类测试用例：一类是自动生成的，而另一类是手工添加的。

文献[7]提出了一种规则集正确性测试方法。该方法从规则边界选取测试数据包(实际上大多数情况下，都是简单地从规则顶点选取数据包)。从规则顶点处选取测试数据包，即不考虑规则不一致性区域，将导致不能找出所有因规则不一致性而带来的配置错误。

另外，关于规则不一致性的研究方面，文献[8]讨论了主机型防火墙规则不一致性的几种形式，对不一致性进行了详细的分类，并采用了一种十分简单的查找检测算法，即线性不一致性规则检测算法。该算法类似于报文分类的线性查找算法，即顺序地检测每条规则。

从以上讨论可知，在规则集正确性测试方面，以文献[7]算法为代表的现有测试数据包选取算法，由于没有考虑规则不一致性区域，从而导致其不能检测出所有因规则不一致性而带来的配置错误。由此可见，有必要研究能够完全检测出这类配置错误的测试数据包选取算法。

本文从计算几何的角度分析了规则不一致性，并按照规则的空间位置关系对规则不一致性进行了分类；然后在此基础之上，提出了一种针对规则集不一致性的测试数据包选取算法。本文算法以两条规则为基本单位，计算其不一致性区域。算法不但从规则顶点选取数据包，而且从规则集不一致性区域选取数据包。测试表明，与文献[7]算法相比，本文算法只需增加少量测试数据包，就能检测出所有因规则不一致性而带来的配置错误。

2 规则不一致性的定义

本文约定：所讨论的规则集是一个包含有 n 条规则的 m 维规则集，记为 I。

定义 1 规则 R_i 定义为：$R_i=\{T_{i1},T_{i2},\cdots,T_{ik},\cdots,T_{im}\}$，$1\leqslant k\leqslant m$。其中 $T_{ik}=[T_{ik}^{S},T_{ik}^{E}]$，即 $T_{ik}=\{x\mid T_{ik}^{S}\leqslant x\leqslant T_{ik}^{E},T_{ik}^{S},x,T_{ik}^{E}$为非负整数$\}$。规则优先级定义为 $pri(R_i)$，处理动作定义为 $act(R_i)$。对于防火墙而言，规则动作通常是放行或者丢弃。

定义 2 一个数据包 P，其对应的 m 维分量为 $T(P)=(t_{P1},t_{P2},\cdots,t_{Pk},\cdots,t_{Pm})$，$t_{Pk}$ 是一个第 k 维值域上的非负整数，$1\leqslant k\leqslant m$。$\exists R_i$，对于 $\forall j,1\leqslant j\leqslant m$，均使得 $t_{Pj}\in T_{ij}$，则称数据包 P 匹配规则 R_i，记为 $P\in R_i$，否则称 P 不匹配规则 R_i，记为 $P\notin R_i$。P 的处理动作记为 $A(P)$。

定义 3 数据包 P 和规则 R_i,R_j，若 $P\in R_i,P\in R_j$，且 $act(R_i)\neq act(R_j)$，则称 R_i 和 R_j 不一致，记为 $R_i\int R_j$。

从定义 3 可知，数据包 P 同时匹配上了规则 R_i 和 R_j，且这两条规则的处理动作不一致。这样就存在一个模棱两可的情况：究竟根据哪条规则来处理 P。通常的解决方案是给规则赋予不同的优先级，按照优先级高的规则的处理动作执行，即定义 4。

定义 4 数据包 P 和规则集合 $Match(P)$，$\forall R_p\in Match(P)$，均有 $P\in R_p$，而 $\forall R_q\notin Match(P),R_q\in I$，均有 $P\notin R_q$，则称 $Match(P)$ 是 P 的匹配集合。$A(P)=act(R_r),R_r\in Match(P)$ 且 R_r 是 $Match(P)$ 中优先级最高的规则。

从定义 1 可知，规则分量 T_{ik} 在数轴上相当于一条线段，规则 R_i 在 m 维空间中相当于一个超长方形。而规则不一致性可以理解为超长方形之间有交叠。由于规则等同于超长方形，因此在本文的上下文环境中，两者可互换。下面从计算几何角度分析规则不一致性。

定义 5 数轴上两条线段 T_{ik} 和 T_{jk}。若 $T_{ik}^{E}<T_{jk}^{S}$ 或者 $T_{jk}^{E}<T_{ik}^{S}$，则称 T_{ik} 和 T_{jk} 无关，记为 $T_{ik}\infty T_{jk}$。若 $T_{ik}^{S}<T_{jk}^{S}\leqslant T_{ik}^{E},T_{jk}^{S}\leqslant T_{ik}^{E}<T_{jk}^{E}$ 或者 $T_{jk}^{S}<T_{ik}^{S}\leqslant T_{jk}^{E},T_{ik}^{S}\leqslant T_{jk}^{E}<T_{ik}^{E}$，则称 T_{ik} 和 T_{jk} 交叉，记为 $T_{ik}\sigma T_{jk}$。若 $T_{ik}^{S}\leqslant T_{jk}^{S}\leqslant T_{jk}^{E}\leqslant T_{ik}^{E}$，则称 T_{ik} 包含 T_{jk}，记为 $T_{ik}\partial T_{jk}$。若 $T_{ik}\sigma T_{jk}$ 或者 $T_{ik}\partial T_{jk}$ 或者 $T_{jk}\partial T_{ik}$，则称 T_{ik} 和 T_{jk} 相关，记为 $T_{ik}\infty T_{jk}$。

定义 6 两个超长方形 R_i 和 R_j。若对于 $\forall k,1\leqslant k\leqslant m$，都有 $T_{ik}\infty T_{jk}$，且 $act(R_i)\neq act(R_j)$，则称 R_i 和 R_j 不一致，即 $R_i\int R_j$。

下面证明定义 3 和定义 6 是等价的。

定理 1 定义 3 和定义 6 是等价的。

证明：首先从定义 3 推导定义 6。

∵数据包 P，$T(P)=\{t_{P1},t_{P2},\cdots,t_{Pk},\cdots,t_{Pm}\}$，$P\in R_i$，$P\in R_j$

∴对于 $\forall k,1\leqslant k\leqslant m$，均有 $t_{Pk}\in T_{ik},t_{Pk}\in T_{jk}$

∴T_{ik} 和 T_{jk} 的关系是交叉关系或者包含关系

∴$T_{ik}\infty T_{jk}$，且 $act(R_i)\neq act(R_j)$

下面从定义 6 推导定义 3。

∵对于 $\forall k,1\leqslant k\leqslant m,T_{ik}\infty T_{jk}$

∴$\exists t_k$，使得 $t_k\in T_{ik},t_k\in T_{jk}$

∴可以构造一个数据包 P,其对应的 m 维为

$T(P)=\{t_1,t_2,\cdots\cdots,t_k,\cdots\cdots,t_m\}$

又∵ $\forall k,1\leqslant k\leqslant m$,均有 $t_k\in T_{ik},t_k\in T_{jk}$

∴ $P\in R_i,P\in R_j$,且 $act(R_i)\neq act(R_j)$

∴命题得证。

3 规则不一致性的分类

文献[8]对规则不一致性进行了详细分类,其类型多达 7 种。对于规则集正确性测试而言,这显然过于复杂。本文从规则的空间位置关系以及规则集正确性测试的角度,简化了规则不一致性的分类。规则不一致性分为以下两类。

定义 7 两个超长方形 R_i 和 R_j,$R_i\int R_j$。$\forall k,1\leqslant k\leqslant m$,均有 $T_{jk}\partial T_{ik}$,则称 R_j 覆盖 R_i,记为 $R_j\triangledown R_i$。

定义 8 两个超长方形 R_i 和 R_j,$R_i\int R_j$。$\exists k,1\leqslant k\leqslant m$,使得 $T_{ik}\sigma T_{jk}$,则称 R_i 部分遮挡 R_j,记为 $R_i\ \Im\ R_j$。

4 测试数据包选取算法

正如前面所述,如何选取测试数据包,即如何设计一种测试数据包选取算法,是规则集正确性测试中的一个关键点。目前以文献[7]为代表的现有测试数据包选取算法,从规则顶点选取测试数据包。使用这种方式选取的数据包,能够检测出因定义 7 讨论的覆盖关系而导致的配置错误,但不能检测出因定义 8 讨论的部分遮挡关系而导致的配置错误(注意,文献[7]从规则顶点选取测试数据包,其主要目的是检测单条规则是否配置正确;而这种选取方式,恰好可以检测出覆盖关系导致的配置错误)。因此,本文提出的测试数据包选取算法,也会从规则顶点选取数据包。由于本文算法是建立在选点映射基础之上的,因此在分析算法之前,首先讨论这个映射。

4.1 选点映射

定义 9 选点映射 $f:L\times L\to 2^N$,其中,L 是数轴上所有线段组成的集合,N 是数轴上所有点组成的集合,2^N 是 N 的幂集。$\forall l\in L,l=[l^S,l^E]$,其中,$l^S$ 表示 l 的左端点,l^E 表示 l 的右端点。

$\forall l_1,l_2\in L$,不妨假设 $l_2^S\leqslant l_1^S$,

$$f(\langle l_1,l_2\rangle)=\begin{cases}\{l_1^S,l_2^E\}, & l_2^S\leqslant l_1^S<l_2^E<l_1^E\\ \{l_1^S,l_1^E\}, & l_2^S\leqslant l_1^S<l_1^E\leqslant l_2^E\\ \{l_1^S\}, & l_2^S\leqslant l_1^S=l_1^E<l_2^E\\ \{l_2^E\}, & l_2^E=l_1^S\\ \Phi, & l_2^E<l_1^S\end{cases}$$

显然,f 是从 $L\times L$ 到 2^N 的映射,故不再对此证明。

4.2 两条不一致规则的测试数据包选取

本文提出的测试数据包选取算法,是以两条不一致规则为基本处理单位的。因此,首先讨论两条不一致规则的测试数据包选取。从第 3 节可知,规则不一致性被划分成两种类型。因此,本节就从这两种类型出发,分别讨论如何选取测试数据包。如前所述,从规则顶点选取测试数据包,可以检测出因定义 7 讨论的覆盖关系而导致的配置错误。这类数据包的选取方法,即定义 11。

定义 10 $\forall R_i\in I,1\leqslant i\leqslant n$,其 m 维分量构成 m 个端点集合,每个集合包括两个元素。其中,第 k 个集合 $N_{ik}=\{T_{ik}^S,T_{ik}^E\}$。

定义 11 $\forall R_i\in I,1\leqslant i\leqslant n$。从顶点选取的所有测试数据包组成的集合为 $Packet_{\triangledown}(R_i)=\prod_{k=1}^{m}N_{ik}$($\Pi$ 为笛卡儿集运算),$1\leqslant k\leqslant m$。

$Packet_{\triangledown}(R_i)$ 中的每个元素,即规则的顶点。下面讨论针对部分遮挡关系,如何选取测试数据包。

定义 12 $\forall R_i,R_j\in I,R_i\ \Im\ R_j,1\leqslant i,j\leqslant n$。对 R_i 和 R_j 的每一维分量进行选点映射,得到 m 个选点集合。其中,第 k 个集合 $f_k(R_i,R_j)=f(\langle T_{ik},T_{jk}\rangle),1\leqslant k\leqslant m$。

定义 13 $\forall R_i,R_j\in I,R_i\ \Im\ R_j,1\leqslant i,j\leqslant n$。针对部分遮挡关系而选取的所有测试数据包组成的集合为 $Packet_{\Im}(R_j,R_i)=Packet_{\Im}(R_i,R_j)=\prod_{k=1}^{m}f_k(R_i,R_j),1\leqslant k\leqslant m$。

$Packet_{\Im}(R_j,R_i)$ 中的每个元素,即针对部分遮挡关系而选取的测试数据包。注意,本文算法会从规则顶点选取数据包,而 $Packet_{\Im}(R_j,R_i)$ 中的一些数据包有可能是规则顶点,因此需要从 $Packet_{\Im}(R_j,R_i)$ 中删去这些数据包,以避免重复计算。

下面分析 $Packet_{\triangledown}(R_i)$ 和 $Packet_{\Im}(R_j,R_i)$ 中的测试数据包是否能够检测出因规则不一致性而带来的配置错误,即定理 2、定理 3 和定理 4。

定理 2 $\forall R_i,R_j\in I,1\leqslant i,j\leqslant n,R_i\triangledown R_j$。$Packet_{\triangledown}(R_i)$ 和 $Packet_{\triangledown}(R_j)$ 中的测试数据包能够检测出 I 中因覆盖关系而导致的规则配置错误。

证明:不妨假设管理员规则配置的本意是 $pri(R_i)<pri(R_j)$。

∵ $R_i\triangledown R_j$

∴ $\forall P\in Packet_{\triangledown}(R_i),A(P)=act(R_i)$

$\forall P'\in Packet_{\triangledown}(R_j),A(P')=act(R_j)$

当管理员规则配置错误,即 $pri(R_i)>pri(R_j)$ 时

$A(P)=act(R_i),A(P')=A(P)=act(R_i)$

∴显然,通过比较测试数据包不同的处理结果,能够发现因覆盖关系而导致的配置错误。

∴命题成立。

定理 3 $\forall R_i,R_j\in I,1\leqslant i,j\leqslant n,R_i\ \Im\ R_j$。$Packet_{\triangledown}(R_i)$ 和 $Packet_{\triangledown}(R_j)$ 中的测试数据包不能完全检测出 I 中因部分遮挡关系而导致的规则配置错误。

证明:不妨假设管理员规则配置的本意是 $pri(R_i)<pri(R_j)$,$m=2$,$T_{i1}\partial T_{j1}$ 且 $T_{j2}\partial T_{i2}$。

∴ $\forall P\in Packet_{\triangledown}(R_i),A(P)=act(R_i)$

$\forall P'\in Packet_{\triangledown}(R_j),A(P')=act(R_j)$

当管理员规则配置错误,即 $pri(R_i)>pri(R_j)$ 时

$A(P)=act(R_i),A(P')=act(R_j)$

∴显然,无论管理员是否配置错误,测试数据包的处理动作均一致。

∴命题成立。

定理 4 $\forall R_i,R_j\in I,1\leqslant i,j\leqslant n,R_i\ \Im\ R_j$。$Packet_{\Im}(R_j,R_i)$ 中的测试数据包能够检测出 I 中因部分遮挡关系而导致的规则配置错误。

证明:不妨假设管理员规则配置的本意是 $pri(R_i)<pri(R_j)$。

∵ $R_i\ \Im\ R_j$

∴ $\forall P\in Packet_{\Im}(R_j,R_i),A(P)=act(R_j)$

当管理员规则配置错误，即 $pri(R_i)>pri(R_j)$ 时

$A(P)=act(R_i)$

∴显然，通过比较测试数据包不同的处理结果，能够发现因遮挡关系而导致的配置错误。

∴命题成立。

4.3 测试数据包的选取

前面描述了两条不一致规则的测试数据包选取，本节将讨论针对整个规则集的测试数据包选取。本文提出的测试数据包选取算法，使用线性检测算法查找所有的不一致规则对，然后根据前面讨论的内容，针对每一个不一致规则对，选取其测试数据包。下面给出本文算法的伪代码。

算法 1 本文算法

```
输入：规则集 I
输出：测试数据包集合 Packet
PACKET SearchTestingPacket(RuleSet I){
PACKET Packet;
//从每条规则顶点处选取测试数据包，无重复地放入 Packet 中
for(int i=1;i<|I|;i++)
GetTestingPacket(Packet,Packet▽(Ri));
//FindOnePairOfConflictRules 使用线性检测算法，返回一对不一致规则
//GetTestingPacket 将 Packet3(Ri,Rj)中的测试数据包无重复地放入 Packet 中
while(<Ri,Rj>=FindOnePairOfConflictRules(I'))
if(Ri ℑ Rj)
    GetTestingPacket(Packet,Packet3(Ri,Rj));
  return Packet;
}
```

下面对本文算法进行分析。从定理 2－定理 4 易证，本文算法能够检测出 I 中因规则不一致性而导致的配置错误。

下面分析算法的时空复杂度。本文算法的空间复杂度主要取决于测试数据包的个数。定理 5 给出了在最坏的情况下测试数据包个数的数量级。

定理 5 在最坏的情况下，本文算法选取的测试数据包个数的数量级是 n^m。

证明：测试数据包的选取主要针对覆盖关系和部分遮挡关系。显然，针对前者选取的数据包个数为 $n2^m$。下面主要分析针对部分遮挡关系而选取的测试数据包个数。

对第 k 维而言，$1\leqslant k\leqslant m$，$\forall R_i,R_j\in I,1\leqslant i,j\leqslant n$，均有 $T_{ik}\infty T_{jk}$ 时，$\bigcup_{1\leqslant i,j\leqslant n} f_k(R_i,R_j)$ 包含的元素达到最多，即 $|\bigcup_{1\leqslant i,j\leqslant n} f_k(R_i,R_j)|_{\max}=2(n-1)$。

∴ $|\prod_{k=1}^{m}(\bigcup_{1\leqslant i,j\leqslant n} f_k(R_i,R_j))|_{\max}\leqslant[2(n-1)]^m$

∴本文算法选取的测试数据包个数的数量级是 n^m。

∴命题得证。

从定理 5 可知，在最坏情况下，本文算法的空间复杂度是 $O(n^m)$。即空间复杂度只与规则个数和规则维数相关，而与不一致区域的大小无关。另一方面，本文算法的时间复杂度主要取决于不一致规则检测时间复杂度。由于本文采用线性检测算法，因此根据文献[8]可知，时间复杂度为 $O(n^2)$。实际上，对于测试数据包选取算法的应用环境而言，绝大多数都不太关心其运行时间。

虽然在理论上最坏的情况下，本文算法的空间复杂度很高，但是在实际应用中，其选取的测试数据包个数与以文献[7]算法为代表的现有算法选取的测试数据包个数相差无几。这一点将在测试部分详细说明。

5 测试

测试环境：Intel2. 4G，512M 内存，linux2. 4 内核，使用 C++语言实现本文算法和文献[7]算法。测试使用的不一致规则检测算法是线性检测算法。测试所用规则集来自本文统计的 24 个企业级防火墙和 95 个个人防火墙，其中规则集的维数从 2 到 5 不等。在这些规则集中，根据文献[8]讨论的规则不一致分类，人为地引入了错误配置。下面首先讨论这些规则集的特性。

第一，防火墙规则数目较少。24 个企业级防火墙中，72%的规则集包含有 200 至 500 条规则，有两个规则集超过 500 条(分别为 523 条和 647 条)，其余规则集都在 200 条以下。而 95 个个人防火墙中，83%的规则集包含有 50 到 100 条规则，仅有 3 个规则集超过 100 条，其余规则集都在 50 条以下，最小的规则集包含 10 条规则。造成这种情况的主要原因在于：目前，大多数企业级防火墙规则，主要是由防火墙厂商事先预设以及管理员手工配置；而个人防火墙规则，主要是由厂商事先预设。厂商事先预设的规则出于安全考虑，主要针对一些常见的攻击，其规则数量有限。而管理员配置的规则是人工配置，其规则数量也不会过多。文献[9]统计的规则集与本文统计的规则集有相似的性质。在文献[9]统计的规则集中，平均规则个数只有 50 条。

第二，规则集包含的不一致规则对的个数很少，平均而言，其比例为规则集大小的 10%。在本文统计的规则集中，平均而言，覆盖关系的规则对个数与规则集大小之比为8.6%，部分遮挡关系的规则个数与规则集大小之比为 1.4%。造成这种情况的主要原因是：1)防火墙厂商预设的规则主要针对特定的端口、协议等相关信息，因此在这些规则中，很少存在不一致规则。2)针对不同的受保护的 IP 地址区域(即受保护的某个 IP 地址或者 IP 地址段)，管理员会配置相同或者不相同的规则。而这些不同的受保护的 IP 地址区域，大多数情况下都是不相交的，只有极少数 IP 地址区域有交叠。又因为覆盖关系的不一致规则对，大多数存在于同一 IP 地址区域；而部分遮挡关系的不一致规则对，大多数存在于两个或多个有交叠的 IP 地址区域。因此，部分遮挡关系的不一致规则对，远远少于覆盖关系的不一致规则对；且部分遮挡关系的不一致规则对个数极少。3)针对某一个受保护的 IP 地址区域，管理员通常只会配置一条或者数条规则来加以保护，这就使得一条规则最多与数条规则不一致。由于在同一个受保护的 IP 地址区域内，不一致规则对大多数属于覆盖关系，因此覆盖关系的不一致规则对数量受到了很大限制。在本文统计的规则集中，一条规则最多与 4 条规则不一致，而在文献[9,10]中，该值是 5。

从上述分析可以看出，大多数防火墙规则集都应具有上述两个性质。

测试项目：

(1)本文算法和文献[7]提出的测试数据包选取算法的对比测试。测试方法：分别使用上述两种算法进行规则集正确性测试，并记录各自的测试结果。测试结果：使用文献[7]算法产生的测试数据包，不能完全检测出所有因规则不一致性而导致的配置错误，表 1 列出了其中部分未被检测出的配置错误(出于安全和隐私考虑，表 1 使用 # 号代替部分 IP 地

址);使用本文算法产生的测试数据包,能检测出所有因规则不一致性而导致的配置错误,因此表1未列出其测试结果。表1中的二维规则包括源IP地址段和目的IP地址段两个分量。在表1中,R_1 和 R_2 不一致,R_3 和 R_4 不一致。管理员的本意是要 R_2 的优先级高于 R_1,R_4 的优先级高于 R_3,然而却恰好配置错误。使用文献[7]算法产生的测试数据包,未能检测出上述配置错误。

表1 使用文献[7]算法的测试结果

	R_1	R_2	R_3	R_4
源IP起始值	#.#.2.45	#.#.2.4	#.#.45.28	#.#.45.30
源IP结束值	#.#.2.49	#.#.2.50	#.#.45.35	#.#.45.33
目的IP起始值	#.#.10.2	#.#.10.5	#.#.3.25	#.#.3.23
目的IP结束值	#.#.10.30	#.#.10.10	#.#.3.30	#.#.3.34
规则处理动作	放行	丢弃	放行	丢弃

R_1 和 R_2,R_3 和 R_4 均是部分遮挡关系。使用文献[7]算法产生的测试数据包,不能检测出这类配置错误的原因即定理3,因此不再赘述。

(2)本文算法与文献[7]算法空间性能对比测试。测试方法:使用上述两种算法处理上述规则集,记录各自产生的测试数据包个数。理论上最坏的情况下,本文算法产生的数据包个数会以 n^m 的数量级增加。而在实际应用中,并不会产生过多测试数据包。图1描述了测试结果,横坐标表示规则集的大小,即包含的规则个数;纵坐标表示本文算法产生的数据包个数与文献[7]算法产生的数据包个数之比。注意:若有两个或者多个规则集大小相同,则纵坐标表示本文算法产生的数据包个数的平均值与文献[7]算法产生的数据包个数的平均值之比。

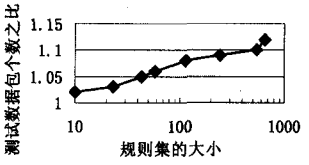

图1 规则增加的个数

当规则个数为58条(仅有一个规则集包含58条规则)时,本文算法产生的测试数据包是896,而文献[7]算法产生的测试数据包数量是845,数据包数量之比为1.06。当规则个数为243条(仅有一个规则集包含243条规则)时,本文算法产生的测试数据包是4157,而文献[7]算法产生的测试数据包数量是3814,数据包数量之比为1.09。当规则个数为647条(仅有一个规则集包含647条规则)时,本文算法产生的测试数据包是11472,而文献[7]算法产生的测试数据包数量是10243,数据包数量之比为1.12。

可以看出,本文算法产生的测试数据包略多于文献[7]算法采用的数据包。从定理5的分析可知,在最坏的情况下,本文算法产生的测试数据包应远多于文献[7]算法产生的数据包,但实际情况却相去甚远。造成这种情况的主要原因是:文献[7]算法只选取规则顶点作为测试数据包,以检测出覆盖关系的不一致规则对,而本文算法除了选取顶点作为测试数据包外,还要针对部分遮挡关系的不一致规则对选取数据包;然而,部分遮挡关系的规则对个数与规则集大小之比仅为1.4%,导致了针对这类关系而选取的测试数据包数量很少。因此,本文算法选取的测试数据包略多于文献[7]算法选取的测试数据包。

另外,对本文算法进行了时间性能测试。当规则集个数达到647条时,本文算法运行时间不到1s。由于在规则集正确性测试中,往往都不太关心测试数据包选取算法的运行时间,因此本文略去其他相关数据。

根据以上讨论可知:尽管在理论上最坏情况下,本文算法选取的测试数据包过多,然而在实际应用中,算法的时空性能良好,并能检测出所有因规则不一致性而带来的配置错误。

结束语 本文从计算几何的角度分析了规则不一致性,并简化了不一致性的分类,在此基础之上,提出了一种针对规则集不一致性的测试数据包选取算法。算法在选点映射基础之上,以两条规则为基本单位,计算其不一致性区域,并选取测试数据包。与以文献[7]算法为代表的现有测试数据包选取算法相比,本文算法不仅从规则顶点选取数据包,而且从规则集不一致性区域选取数据包。测试表明,本文算法只需增加少量测试数据包,就能检测出所有因规则不一致性而带来的配置错误。

参考文献

[1] Allen J. The CERT Guide to System and Network Security Practices[M]. Addison-Wesley,2001

[2] Jurjens J,Wimmel G. Specification-based Testing of Firewalls [C]//Proceedings of the 4th International Conference on Perspectives of System Informatics. 2001:308-316

[3] Wool A,Mayer A,Ziskind E. Fang:A Firewall Analysis Engine [C]//Proceedings of the 2000 IEEE Symposium on Security and Privacy. August 2000:85-97

[4] Lyu M, Lau L. Firewall Security: Policies, Testing and Performance Evaluation[C]//Proceedings 2000 International Conference on Computer Systems and Applications. October 2000: 116-121

[5] Schultz E,Frantzen M,Kerschbaum F,et al. A Framework for Understanding Vulnerabilities in Firewalls Using a Dataflow Model of Firewall Internals[J]. Computers and Security,2001, 20(3):263-270

[6] Al-Tawil K, Al-Kaltham I. Evaluation and Testing of Internet Firewalls[J]. International Journal of Network Management,9 (3):135-149

[7] El-AtawyA,IbrahimK. PolicySegmentationforIntelligentFirewall Testing[C]//The 1st IEEE Workshop on Secure Network Protocols. Nov. 2005:67-72

[8] Al-Shaer E,Hamed H. Taxonomy of conflicts in network security policies[J]. IEEE Communications Magazine,2006,44(3): 134-141

[9] Gupta P, et al. Packet Classification on Multiple Fields [C]// Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. Massachusetts,1999:147-160

[10] Taylor D E. Survey & taxonomy of packet classification techniques[R]. WUCSE-2004-24. Department of Computer Science & Engineering, Washington University in Saint Louis, May 2004

[11] Diana S,David B,Germano C. Firewall Conformance Testing[C]// The 17th IFIP TC6/WG 6.1 International Conference. May, 2005:226-241

[12] 田原,云晓春,朱晓辉. 防火墙性能基准测试研究[J]. 计算机仿真,2003,20(7):123-126