分级结构的 AdaBoost 入侵检测方法研究

王 勇[1,2]，陶晓玲[1]

(1. 桂林电子科技大学 网络中心，广西 桂林 541004；
2. 北京航空航天大学 计算机学院，北京 100083)

摘要：针对目前智能入侵检测方法存在不能同时满足检测精度和检测速度的要求问题，提出一种分级结构的智能入侵检测方法. 该方法将改进的 AdaBoost 算法用于入侵特征的选择及构造每一级的 Ada-域值分类器，并通过级连多个分类器来共同完成检测任务. 设计并实现了 Linux 实时入侵检测实验平台，在此平台上训练和测试分级结构的智能入侵检测器. 实验结果表明，该方法降低了运算复杂度；在保证高的检测率的同时，降低了虚警率；提高了处理速度，更适合入侵检测系统的实时处理要求.

关键词：入侵检测；AdaBoost 算法；特征选择；Ada-域值分类器；分级结构

中图分类号：TP393；TP181 **文献标识码**：A **文章编号**：1001-2400(2008)02-0345-06

Study of the intrusion detection method based on AdaBoost with a hierarchical structure

WANG Yong[1,2], *TAO Xiao-ling*[1]

(1. Network Information Center, Guilin Univ. of Electronic Technology, Guilin 541004, China; 2. School of Computer Science and Eng., BeiHang Univ., Beijing 100083, China)

Abstract: An intelligent hierarchical intrusion detection method is proposed for getting both high precision and high speed. With this method, an improved AdaBoost algorithm is used in selecting intrusion features and constructing an Ada threshold-classifier at every level, and several hierarchical classifiers are combined for detection. A Linux IDS experimental platform is designed and implemented to train and test the intelligent intrusion detector. Experimental results show that the method reduces the complexity of computation, and that the false negative rate is reduced greatly while maintaining the high detection rate. Moreover, the method improves the processing speed and is especially appealing for the real-time processing of the intrusion detection system.

Key Words: intrusion detection; AdaBoost algorithm; feature selection; Ada threshold-classifier; hierarchical structure

网络的广泛普及和应用，使得网络安全问题成为这个时代的永恒话题. 入侵检测作为网络安全的第二道防线，也一直是学者们热衷的研究课题. 入侵检测方法是入侵检测研究的重点. 随着人工智能技术的不断发展，它的许多先进方法应用到了新一代的入侵检测系统中. 文献[1,2]列举了当前研究中主流的智能入侵检测方法，这些方法从不同的角度来处理入侵检测问题，利用各种技术构建入侵检测的正常模型或攻击模型. 它们各有特点，也存在不足. 入侵检测性能的主要量度是检测精度及检测速度，当一种方法的检测精度很高而且检测速度很快时才被认为其性能很好. 当前的研究存在的普遍问题是这些方法不能同时满足检测精度和检测速度的要求. 这就需要更有效的技术来改进这些智能入侵检测方法. 这里引入分级结构的思想，分类器分级的思路在人脸检测中取得了成功[3]，证明其在处理目标检测这类问题时是行之有效的，因而可以尝试利用它来解决入侵检测问题. 另一方面，多数入侵检测方法都是使用 KDD 99 数据集进行评估，然而，面对日

收稿日期：2007-08-17

基金项目：教育部高校博士点基金资助(20040251010)；广西自然科学基金资助(桂科基 0575094)

作者简介：王 勇(1964-)，男，桂林电子科技大学教授，博士，E-mail: ywang@guet.edu.cn.

益复杂的网络环境,该数据集有一定的局限性.因此笔者在借鉴麻省理工学院林肯实验室成功经验的基础上,设计并实现了在新的软硬件环境下的 Linux 实时入侵检测实验环境,用于验证分级结构智能入侵检测方法的有效性.

1 基于 AdaBoost 算法的特征选择及分类器的构造

AdaBoost 算法是机器学习领域中的重要算法之一[4].该算法通过依次训练一组弱分类器,将它们集成为一个强分类器.基本过程是:每个训练样本被赋予一个权值,表明它被某个弱分类器选入训练集的概率.当一个弱分类器训练完成后,根据其在训练集上的分类结果对所有的样本权值进行调整.如果某个样本被当前弱分类器准确分类,那么它的权重就会被降低,则在构造下一个弱分类器的训练集时,它被选中的概率就被降低;相反,如果某个样本没有被正确分类,则它的权重就相应被提高,它入选下一个弱分类器的训练集的概率被提升.通过这种方式,Adaboost 能够"聚焦于"那些比较困难(容易出现错分)的样本.集成后的强分类器的判决结果是所有弱分类器的判决结果的加权和.Schapire R. E.等证明,使用 AdaBoost 算法能够得到既在训练集上具有低错误率又具备相当泛化能力的分类器[5].

AdaBoost 算法的目标是提高给定的学习算法的分类准确率,它对提高入侵检测识别率方面也是有成效的[6,7].而笔者从特征选择和分类的角度对 AdaBoost 算法进行了修改,所得到算法的基本思想是:反复选择入侵特征构建两值弱分类器,选出一些分类能力优的弱分类器(也就是选择一些重要特征),然后根据挑选出的这些弱分类器的加权组合构成最终的强分类器.训练过程中的每个弱分类器都是基于单特征的;而且采用轮盘赌的方法选择训练子集;每一轮循环挑选出在当前权重分布下误差最小的弱分类器,即选出具有最佳分类表现的相应特征;根据误差来更新权重分布;得到的对应每一个弱分类器的权重,在一定程度上还可以用来衡量不同特征对分类的贡献度.

具体算法描述如下:

(1) 选取训练样本 $(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)$,$x_i$ 表示每个入侵检测数据的特征向量,$y_i=\pm 1$ 为类别标识,其中 $y_i=1$ 代表正常样本,$y_i=-1$ 代表异常样本.

(2) 初始化权值,每个训练样本具有起始权值 $w_{1,i}=\begin{cases}1/(2p) & , \quad y_i=1\\ 1/(2q) & , \quad y_i=-1\end{cases}(i=1,\cdots,m)$,其中 p 和 q 分别表示正常样本和异常样本的数目.

(3) for $t=1,\cdots,T$

1) 规一化权值 $w_{t,i}=w_{t,i}\Big/\sum_{i=1}^{m}w_{t,i}$,使得 w_t 为一概率分布;

2) 对于每一种未被选择的特征 j,采用轮盘赌的方法选一训练子集训练弱分类器 h_j(只限于单个特征),分类误差计为 ε_j:$\varepsilon_j=\sum_i w_{t,i}\{h_j(x_i)\neq y_i\}$;

3) 选择具有最小误差的分类器 h_j 作为 h_t,对特征 j 附加已选择标记(其在特征向量中的位置用 l_t 表示);

4) 更新权值 $w_{t+1,i}=w_{t,i}\beta_t^{1-e_i}$,当 x_i 被正确分类时,$e_i=0$;否则 $e_i=1$,并且 $\beta_t=\varepsilon_t/(1-\varepsilon_t)$;

5) $t++$.

(4) 最终找出 T 个最优特征的分类器 $h_1,h_2,\cdots,h_T$,其在特征向量中的位置是 $l_1,l_2,\cdots,l_T$,每个分类器的权值 $\alpha_t=\log(1/\beta_t)$,最后的强分类器的判决函数为:$H(x)=\mathrm{sign}\Big(\sum_{t=1}^{T}\alpha_t h_t(x)-\theta\Big)$,其中 sgn() 为符号函数;若 $H(x)\geqslant 0$,则判断 x 为正常数据;反之判断 x 为异常数据.θ 为判决域值,初始值设定为所有弱分类器权值的平均值,即 $\theta=\frac{1}{T}\sum_{t=1}^{T}\alpha_t$.

依据上述算法构造的分类器,将其称为 Ada-域值分类器.针对算法描述中判决域值 θ 初始值的设置,文

献[3]中将其初始值设定为:$\theta = \frac{1}{2}\sum_{t=1}^{T}\alpha_t$,即传统两类分类问题中对应最小错误率处的域值;对判决域值 θ 的选取是与分级结构中要求的检测率和虚警率相对应的,笔者经过反复的实验验证,得出:当 θ 的初始值设定为所有弱分类器权值的平均值时,即$\theta = \frac{1}{T}\sum_{t=1}^{T}\alpha_t$ 时,比较合适.

2　分级结构的分类方法设计

对实时的网络数据或主机数据进行入侵分类是一个复杂的两类问题,很难得到一个单一的强分类器同时满足检测精度和计算时间的要求.为了解决准确率和效率之间的矛盾,采取分级结构的思想.

分级结构入侵检测器的框架图如图 1 所示.该检测器由多级分类器构成.检测过程就像一颗退化的决策树.只有前面一级的分类器判决为异常的样本才被送入后面一级的分类器继续处理,反之则被认为是正常样本直接排除.最后,只有那些被每一级分类器都判决为异常的样本才作为检测到的输出结果.

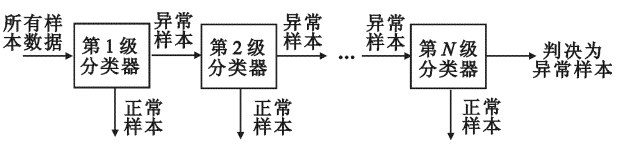

图 1　分级结构入侵检测器的框架图

在分级结构中,每一级都是用改进的 AdaBoost 算法训练得到的一个 Ada-域值分类器.通过调整分类器的判决阈值 θ,使得每一级都能让几乎全部的异常样本通过,而拒绝很大一部分正常样本;从第二级之后的各级正常样本的采集方法很方便,只需将前面级次中错分的正常样本作为下一级次分类器的正常样本;靠近前面各级的分类器,只使用小部分重要特征,结构比较简单,却拒绝了大部分的正常样本;靠近后面各级的分类器,虽然采用大量的特征来排除近似异常样本的正常样本的干扰,但是由于需要处理的样本数目很少,对于整体运算时间的耗费很小.在实时的入侵检测系统中,检测到的多数是正常数据,异常数据占有比较小的比例.如果采用分级结构的检测方法,在前面几级的分类器中就已经滤除了大部分的正常数据,只有小部分的数据要通过所有级的分类器,大大降低了运算复杂度,更能满足实时检测的要求.

设计分级结构的检测器时,需要解决以下 3 个问题:(1)分类器的级数;(2)每一级分类器的特征数;(3)每一级分类器的判决域值.先引入几个性能指标,其定义如下:

(1)总检测率　$D = \prod_{i=1}^{K} d_i$,其中 d_i 为每一级分类器的检测率,即 d_i 为分对的异常样本数 / 异常样本总数;K 为分类器的级数.

(2)总虚警率　$F = \prod_{i=1}^{K} f_i$,其中 f_i 为每一级分类器的虚警率,即 f_i 为分错的正常样本数 / 正常样本总数;K 为分类器的级数.

要构造分级结构的入侵检测器,必须要先进行训练,其训练框架图如图 2 所示.整个训练过程要满足事先设定的总虚警率 F 的要求,F 设置的越低,所需的分类器的级数就越多;在每级分类器进行训练时,对检测率 d_i 和虚警率 f_i 也都有要求:要满足高的检测率 d_i,这要经过调整分类器的域值 θ 来实现;而要满足比较低的虚警率 f_i,则要通过增加弱分类器的个数来实现.这样使得分类器组成分级结构后,仍然能够保持较高的检测率.同时,保证绝大多数的虚警样本在后面的级次中被正确地识别出来,以满足系统对虚警率的要求.

分级结构智能入侵检测器的训练方法描述如下:

(1) 确定检测性能目标:设定总虚警率 F,每级可接受的最大虚警率 f 和最小检测率 d;

(2) 初始化:分级结构分类器总虚警率的值 $F_0 = 1.0$,取训练样本集:P 为正常样本集,N 为异常样本集;分类器的级数 $i = 0$;

(3) while $F_i > F$

　　1) i ++;

　　2) 初始化:特征数 $n_i = 0$;虚警率 $f_i = 1.0$;

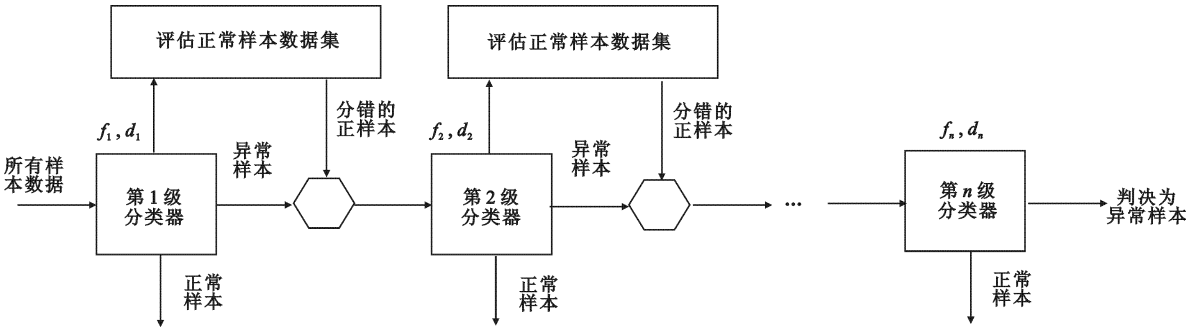

图 2 分级结构入侵检测器的训练框架图

3) while $f_i > f$

① n_i ++;

② 用 P 和 N 作为训练样本,采用改进的 AdaBoost 算法选取 n_i 个特征(即弱分类器),并构造第 i 级 Ada-域值分类器;

③ 计算第 i 级分类器的虚警率 f_i 和检测率 d_i;

④ 调整该级分类器的判决域值 θ,使其检测率 $d_i \geqslant d$;

4) P 置为空集;

5) 如果 $F_i > F$,则用第 i 级分类器评估正常样本数据集,并且把所有分错的正常样本放入 P 集.

3 实验的建立及结果分析

3.1 实验环境

借鉴麻省理工学院林肯实验室在收集 IDS 的标准测试数据时提出的提供服务以及攻击可控制原则,这里的 Linux 入侵检测系统测试平台的构建原则为:(1)整个实验在正常的可控的网络环境中进行,服务器方提供 FTP,HTTP,SMTP,SSH/TELNET,SAMBA 等正常服务;(2)借助当前应用普遍的入侵检测系统以及防火墙,保证攻击的可控性和样本的性质. 基于以上原则, 设计实验环境如图 3 所示.

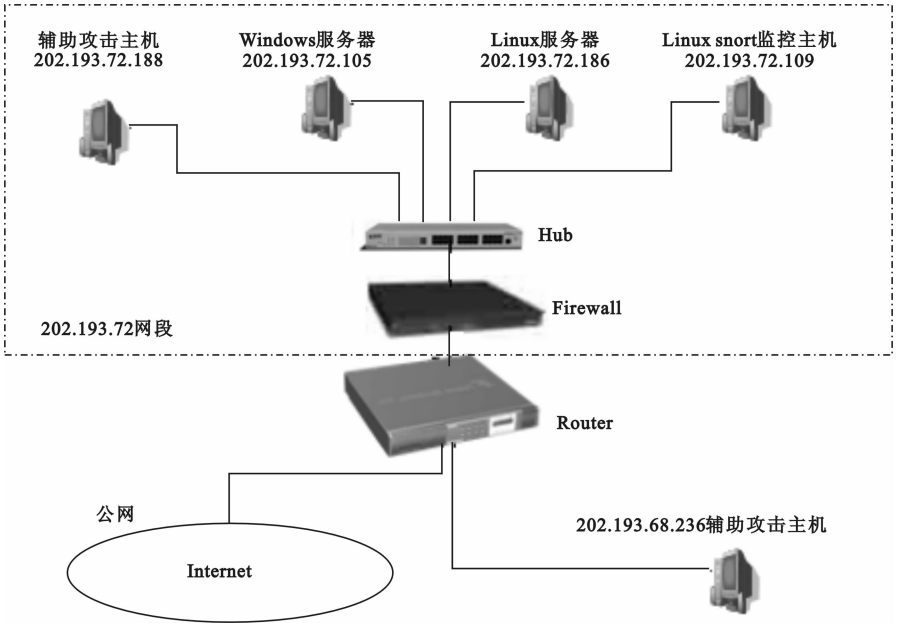

图 3 入侵检测实验环境

在 Linux 服务器(202.193.72.186)上,安装了 Red Hat Linux 9 操作系统,在该服务器上分别开启了

Apache 以及 Tomcat 的 HTTP,Vsftpd 的 FTP,SSH,Samba,Sendmail 的 SMTP,MySQL 的数据库等服务.对该服务器进行攻击并在其上进行入侵检测的数据采集工作.

3.2　网络攻击设计

图 3 的实验环境中 202.193.72.188 和 202.193.68.236 是两台辅助攻击的主机,当然,辅助攻击是在可控范围之内的.选择了多种有效的攻击工具,这些工具也是当前网络上黑客常用的或者是经典的工具:(1)综合扫描类工具:X-Scan,superscan,Retina,Hscan,netcat,bluescan,amap,nmap,流光;(2)拒绝服务攻击类工具:tfn2k,Jolt,udpflood,teardrop.c,Smurf,SYN Flood;(3)远程攻击工具:proft_put_down,httpd,Brutus;(4)普通授权用户提升到特权用户的攻击工具:hatorihanzo.c,Root.c;(5)后门程序:netcat,stcpshell,tcpbackdoor,adore,Q-2.4;(6)其他恶意脚本:sars.

3.3　特征提取

参考 LIDS,snort 等入侵检测系统的特征选取方法,Linux 入侵检测系统所选定的原始特征共 5 个类别 27 个参数,其中包含了对入侵比较敏感的主机数据特征以及网络数据特征,如表 1 所示.而在特征提取的数据实际采集方面,系统的设计主要针对设定的时间窗,对主机以及网络数据进行统计.时间窗的选择直接影响整个系统的检测率以及开销.笔者经过理论计算和实验验证,最终将系统的时间窗设定为1 s[8].

表 1　入侵系统采集的特征

特征类别	特征含义(位置号)	取值类型	特征类别	特征含义(位置号)	取值类型
系统级特征	系统级 CPU 利用率(1)	实数	端口连接特征	TCP 开放端口数(5)	整数
	用户级 CPU 利用率(2)	实数		UDP 开放端口数(6)	整数
	物理内存利用率(3)	实数		RAW 开放端口数(7)	整数
	SWAP 利用率(4)	实数		特殊端口连接数(8)	整数
	敏感资源被占用的次数(11)	整数		TCP 连接数(23)	整数
	敏感系统调用次数(26)	整数		UDP 连接数(24)	整数
用户级特征	时间窗内用户错误登陆次数(13)	整数		RAW 连接数(25)	整数
	登陆用户的数目(14)	整数	网络传输特征	SYN 包所占比例(17)	实数
	root 用户在登录用户中的比例(15)	实数		RST 包所占比例(18)	实数
	敏感指令所占比例(16)	实数		URG 包所占比例(19)	实数
进程级特征	运行进程的总数(9)	整数		FLAG 包所占比例(20)	实数
	运行状态的进程所占比例(10)	实数		出站流量(21)	实数
	敏感资源占用次数与占有敏感资源的进程数的比率(12)	实数		入站流量(22)	实数
				HTTP 传输错误数(27)	整数

注:1. 敏感资源指下列目录/文件中的资源:/bin,/sbin,/usr/bin,/usr/sbin,/etc/rc.d,passwd,shadow.

2. 敏感指令指下列指令:su,chmod,chown,chgrp,fork,exec,insmod,rmmod.

3. 敏感系统调用指:sys_get_kernel_sysms,sys_chmod,sys_chown, sys_fchmod,sys_fchown,sys_create_module,sys_delete_module,sys_query_module.

4. 特殊端口指除了 28 个常规端口之外的端口.

3.4　实验结果分析

通过使用内核模块加载(LKM),proc 文件系统等 Linux 高级编程技术,从 Linux 服务器中采集原始数据,并对原始数据进行预处理,形成样本数据.利用改进的 AdaBoost 算法进行特征选择(其中弱分类方法采用支持向量机),并构造分级结构的入侵检测方法,在上述入侵检测系统中进行实验测试.这些算法和方法均在 PC 机(CPU 为 PIV 1.60 GHz,内存为512 MB)上,利用 Visual C++6.0 开发工具实现.

通过 AdaBoost 特征选择算法,对 Linux 入侵检测系统所选定的 27 个原始特征进行了挑选和排序,结果为(重要性从大到小):{23,17,1,5,4,22,15,19,2,3,18,14,6,7,8,9, 10,11,26,12,13,20,16,21,24,25,27}.算法实现中采用的训练样本集为3 100条样本数据,其中正常样本1 600条,异常样本1 500条.每轮的训练子集用轮盘赌方法选择1 000条样本数据,算法实现的时间为653 s.根据得到的特征重要性排序序列,对入侵特征进行约减尝试(这部分将另文描述),并结合数据采集的成本以及实际的需要,Linux 入侵检测系统约减掉原始特征中的{27,25,24,21,16,20},保留 21 个特征.

接下来要采用图 2 所示的训练方法,训练一分级结构智能入侵检测器. 首先就是确定检测性能目标:设定总虚警率$F=3\%$,每级可接受的最大虚警率$f=30\%$和最小检测率$d=95\%$;训练第 1 级分类器,采用了2 520条样本数据,其中正常样本1 479条,异常样本1 041条;用第 1 级分类器评估一正常样本数据集(有3 079个样本),将分错的正常样本和经过第 1 级分类器分对的负样本作为第 2 级分类器的训练样本,共2 348条,其中正常样本1 307条,异常样本1 041条;第 3 级分类器的训练样本为:第 2 级分类器评估正常样本数据集得到的分错的正常样本,和经过第 2 级分类器分对的负样本,共1 424条,其中正常样本 387 条,异常样本1 037条. 采用 3 级结构便满足了检测性能目标,训练结果如表 2 所示.

表 2 分级结构入侵检测器的训练结果

	第 1 级分类器	第 2 级分类器	第 3 级分类器	分级结构分类器
所用特征个数	5	12	21	21
检测率/%	100.00	99.62	96.82	96.45
虚警率/%	14.13	29.61	8.79	0.37

对上述训练得到的分级结构智能入侵检测器,按照图 1 所示的方法进行测试. 测试样本集采用4 000条样本数据,其中正常样本3 000条,异常样本1 000条. 测试结果如表 3 所示. 经过第 1 级分类器,就有2 029条(占 67.63%)正常样本数据被直接排除. 也就是通过数量很少的几个特征训练得到的分类器,在保证高的检测率的同时,能够拒绝相当数量的正常(干扰)样本,降低了运算复杂度;经过第 2 级分类器后,有 89.73%的正常样本被排除了;整体分级结构的分类器与单级的分类器相比,检测率的下降不多,但通过采用多个分类器的分级结构,虚警率则有大幅度的改善.

表 3 分级结构入侵检测器的测试结果

	第 1 级分类器	第 2 级分类器	第 3 级分类器	分级结构分类器
所用特征个数	5	12	21	21
检测率/%	100.000	99.700	96.890	96.600
虚警率/%	32.370	31.720	9.740	1.000
测试时间/s	1.261	3.044	4.266	8.571

表 4 两种分类器的测试性能比较

	未分级的分类器	分级结构分类器
检测率/%	96.6	96.6
虚警率/%	9.33	1.00
测试时间/s	9.743	8.571

如果不采用分级结构,而是直接采用由改进的 AdaBoost 算法训练得到的一个含 21 个弱分类器的 Ada-域值分类器,针对同样的测试样本集,与分级结构分类器的测试性能对比如表 4 所示. 可知,分级结构的检测器在保证高的检测率的同时,大大降低了虚警率,整体运算时间的耗费也变少.

4 结 束 语

提出一种分级结构的基于 AdaBoost 算法的入侵检测方法. 该方法先从特征选择和分类的角度,将改进的 AdaBoost 算法用于入侵特征的选择,并在此基础上构造每一级的 Ada-域值分类器,最后将这些分类器串联在一起形成一高性能分类器进行入侵检测. 建立了一个模仿林肯实验室的 Linux 实时入侵检测实验平台,将分级结构的智能入侵检测方法在其上进行验证,实验结果表明该方法是有效的.

参考文献:

[1] 马传香, 李庆华, 王卉. 入侵检测研究综述[J]. 计算机工程, 2005, 31(3): 4-6.
Ma Chuanxiang, Li Qinghua, Wang Hui. A Study Survery of Intrusion Detection[J]. Journal of Computer Engineering, 2005, 31(3): 4-6.

(下转第 361 页)