第 47 卷第 6 期
2015 年 12 月

南 京 航 空 航 天 大 学 学 报
Journal of Nanjing University of Aeronautics & Astronautics

Vol. 47 No. 6
Dec. 2015

DOI:10.16356/j.1005-2615.2015.06.020

基于支持向量机的 Webshell 黑盒检测

叶 飞[1,2] 龚 俭[1,2] 杨 望[1,2]

(1.东南大学计算机科学与工程学院,南京,210096; 2.江苏省计算机网络技术重点实验室,南京,210096)

摘要:Webshell 是一种基于 Web 的网站后门程序。当前已有的 Webshell 检测方法都需要根据脚本程序源代码来检测,因此只能部署在服务器主机上,而且只能检测本机的网站代码。本文通过分析 Webshell 的 HTML 页面特征,提出了一种基于支持向量机(Support vector machine,SVM)分类算法的黑盒检测方法。该方法是一种有监督的机器学习系统,对先验网页的 HTML 页面进行学习,可以在未知脚本源代码的情况下对 Webshell 进行检测。实现结果表明,该方法在黑盒的条件下达到了较高的准确率和极低的误报率,并且取得了与白盒检测方法相近的检出率,可以部署在基于网络的入侵检测系统中,同时监测多台服务器是否包含 Webshell,从而帮助监控入侵趋势和网络安全态势。

关键词:Webshell;网站后门;支持向量机;入侵检测;机器学习

中图分类号:TP393 **文献标志码**:A **文章编号**:1005-2615(2015)06-0924-07

Black Box Detection of Webshell Based on Support Vector Machine

Ye Fei[1,2], *Gong Jian*[1,2], *Yang Wang*[1,2]

(1. School of Computer Science and Technology, Southeast University, Nanjing, 210096, China;
2. Key Laboratory of Computer Network Technology of Jiangsu Province, Nanjing, 210096, China)

Abstract: Webshell is a kind of backdoor program based on Webpages. Existing Webshell detection methods rely on the script source code, therefore they can only be deployed on the server in which the pages are scanned. By analyzing the HTML feature of Webshell pages, a black box detecting method based on support vector machine (SVM) classification algorithm is proposed. The method is one sort of supervised machine learning system which can detect unknown Webshell without the knowledge of source code. The experimental result indicates that the black box method has a high accuracy along with a low false positive rate, and reaches an approximate detect rate as the white box detection methods. Therefore, it can be deployed in intrusion detection system(IDS) based on network and monitor more than one server from being injected Webshell, thus helping to monitor the intrusion trends and network security situation.

Key words: Webshell; backdoor; support vector machine(SVM); intrusion detection; machine learning

近年来,CSDN 密码泄密门、Apache Struts2 漏洞、棱镜门等一系列网络安全事件进入人们的视野,这其中绝大部分都与 Web 网站安全息息相关。根据 CNCERT 每年发布的年度安全报告,与 Web 网站相关的入侵行为呈逐年上升趋势。在所有安全威胁中,尤其以网站后门为甚。

网站后门又被称为 Webshell,是一种基于 Web 服务的后门程序。据国家互联网应急中心发布的《2013 年我国互联网网络安全态势综述》显示,中国互联网站总量近 350.7 万个,但受攻击明显增多。其中网站被植入 Webshell 后门等隐蔽性攻击事件呈增长态势,网站用户信息成为黑客窃取

收稿日期:2014-09-30;**修订日期**:2015-01-03

通信作者:龚俭,男,教授,博士生导师,E-mail:jgong@njnet.edu.cn。

的重点。与以往通过明显篡改网页内容以表达诉求或炫耀技术不同的是,2013 年,黑客倾向于通过隐蔽的、危害更大的后门程序,获得经济利益和窃取网站内存储的信息。据 CNCERT/CC 监测,2013 年,我国境内(我国海关关境以内)被植入 Webshell 后门的网站数量为 76 160 个/年,较 2012 年增长 46%,其中政府网站有 2 425 个。

Webshell 危害巨大,如果发现网站已经被植入 Webshell 后,则意味着攻击者已经利用漏洞掌握了服务器的控制权限。因此检测 Webshell 对于及时掌握网络安全态势具有极其重要的意义。

1　Webshell 的特点和分类

攻击者通过 Webshell 获得 Web 服务的管理权限,从而达到对 Web 网站服务器的渗透和控制目的。从攻击者的角度看,WebShell 就是一个 ASP(ASP. NET)、JSP 或者 PHP 脚本木马后门。攻击者在人侵一个网站后,常常将这些脚本木马后门文件放置在网站服务器的 Web 目录中,然后就可以用 Web 页面的方式,利用脚本木马后门控制网站服务器。通过 Webshell,黑客可以上传、查看、修改和删除网站服务器上的文件,可以读取并修改网站数据库的数据,甚至可以直接在网站服务器上运行系统命令。

WebShell 最大的优点就是可以穿越防火墙。由于访问 Webshell 的行为和访问普通 Web 的行为特征几乎一致,所以可逃避传统防火墙和杀毒软件的检测;并且使用 WebShell 进行服务器管理操作不在系统安全日志中留下记录,只会在网站的 Web 日志中留下一些数据提交记录与正常的网页文件混在一起,因此管理员很难看出入侵痕迹的。而且随着各种用于反检测特征混淆隐藏技术应用到 Webshell 上,使得传统基于特征码匹配的检测方式很难及时检测出新的变种。正因为 Webshell 具有方便使用、难以检测的特点,几乎所有的 Web 攻击者在通过这种手段取得服务器权限后,都会通过上传一个 Webshell 来进行更深层次的攻击行为。

根据脚本程序的大小和功能,攻击者通常将 Webshell 分为“大马”、“小马”、“一句话木马”三类。

(1) 大马:具有最全面的功能,通常包含友好的操作界面,可以在图形界面下进行文件操作、命令执行和数据库操作等,更强大的可能包括漏洞利用代码。大马往往文件较大,代码使用混淆手段防止检测。此外有的大马会包含一个登陆界面,只有正确输入口令才能使用。

(2) 小马:仅包含单一功能的 Webshell 被称为小马。小马通常提供文件上传或者数据库提权等操作,在 Web 网站对于上传文件大小有限制时,攻击者会使用小马作为上传跳板。小马的文件大小往往在 5 KB 以内,并且没有口令保护。

(3) 一句话木马:一句话木马指经过混淆或者变形的一句脚本代码,通常是命令执行代码,例如 eval()函数。由于只有一行代码,所以一句话木马具有更好的隐藏特性,可以插入 Web 网站原先已有的代码中而很难被管理员发现。

2　相关工作

白盒检测和黑盒检测是检测的两种常见方法。白盒检测即在已知源代码的情况下进行检测;而黑盒检测则对于程序源代码一无所知,仅仅根据程序的输入输出进行检测。

现有检测 Webshell 的方法都是白盒检测,即针对 Webshell 脚本文件的源代码进行检测,具体可以分为基于主机的检测和基于网络的检测两类。

2.1　基于主机的检测

工业界较普遍的检测方法为直接使用已知关键词作为特征,使用 grep 语句搜索可疑文件后人工分析,或者使用程序定期检查已有文件的 MD5 值以及检查是否有新文件产生。这种直观的检测方法容易被攻击者使用混淆手段规避。

Emposha 的 Web Shell Detector 是一个 php 脚本,使用一个较大的 Webshell 特征库,可以用来识别 php/cgi/asp/aspx 的 Webshell,宣称识别率高达 99%。

InfoSec Institut 的开源检测程序 NeoPI 针对 5 种常见 Webshell 的混淆特征进行检测,包括:信息熵、最长单词、重合指数、特征和压缩比。

D 盾_Web 查杀可以识别加密变形后的 Webshell,使用自行研发不分扩展名的代码分析引擎,能分析更为隐藏的 WebShell 后门行为。引擎特别针对、一句话后门、变量函数后门、$\{\}$执行、\` 执行、preg_replace 执行、call_user_func、file_put_contents 和 fputs 等特殊函数的参数进行针对性的识别,能查杀更为隐藏的后门。

胡建康等[1]提出通过提取文档属性、基本属性和高级属性 3 类特征数据,使用决策树 C4.5 分类器进行分类判定,检测 Webshell 的方法。

2.2 基于网络的检测

目前的研究主要集中于在网络入口处配置入侵检测系统(Intrusion detection system,IDS)或者WAF来检测Webshell。Fireeye提出使用Snort配置特征规则来检测一句话木马"中国菜刀"。Yang等[2]通过配置ModSecurty的核心规则集来检测上传Webshell的行为。

上述两种方法都是通过分析HTTP请求中是否包括特殊关键词(例如<form、<%、<?、<php等)来判断攻击者是否正在上传HTML或者脚本文件,即将所有上传HTML或脚本的行为认为是一次上传Webshell的攻击。这种方法需要很大的开销,存在误报的可能性;并且只能检测到正在发生上传Webshell的行为,对于已有的Webshell则无能为力。

3 基于SVM的检测

3.1 SVM分类算法

常见的网页文本分类算法有以下几种:

(1) 概率模型方法:例如朴素Bayes方法[3]。

(2) 关系学习方法:例如Cohen的FLIPPER系统,以及Parson使用PROGOL算法结合WordNeyt中的语义信息进行网页分类,Slattery等[4-5]将FOIL归纳算法用于网页分类。

(3) 支持向量机方法:Joachim[6]将SVM[7]方法应用到文本分类,并取得好的分类性能。

根据第2节的分析,由于本文涉及的特征维度较高,对于朴素贝叶斯而言,需要假设各维度相互独立。维度高的一个可能问题就是维度之间的独立性会变差,造成朴素贝叶斯的假设不成立,从而效果不好;而支持向量机SVM则适用于高维度的二分问题[8]。此外文献[9]显示使用关系学习方法归纳速度慢,并且用学习到的规则进行分类时,通常会出现查准率高、查全率低的现象,因此本文选用SVM作为本文的分类算法。

SVM的应用是文本分类近年来最重要的进展之一[10]。相对于其他方法,SVM在解决小样本、非线性及高维模式识别中表现出许多特有的优势。SVM方法是通过一个非线性映射p,把样本空间映射到一个高维乃至无穷维的特征空间中(Hilbert空间),使得在原来的样本空间中非线性可分问题转化为在特征空间中的线性可分问题。一般的升维都会带来计算的复杂化,SVM方法巧妙地解决了这个难题:应用核函数的展开定理,就无需知道非线性映射的显式表达式;由于是在高维特征空间中建立线性学习机,所以与线性模型相比,不但几乎不增加计算的复杂性,而且在某种程度上避免了"维数灾难"。

选择不同的核函数,可以生成不同的SVM。本文将选取默认的RBF核函数(即径向基核函数)和线性核函数进行对比实验。

3.2 特征选取

由于Webshell的编写目的是为了方便入侵者进一步攻击,与合法页面为了提供信息给用户的目的完全不同;因此,Webshell的HTML页面结构和合法页面的结构也有较大的不同。参考胡建康等[1]在白盒检测提取的Webshell测度,本文选取的SVM分类测度分为结构特征和文本特征两种。

3.2.1 结构特征

Webshell的页面某些结构特征与正常的合法页面存在较大的不同。本文将结构特征分为以下5类:

(1) 文件特征:包括文件名称,文件大小,规格化后的行数。大部分攻击者并不自己编写Webshell,而是直接使用网络上提供的版本,因此文件名称特征具有较大的可识别性。此外,经过统计,70%的合法网页的文件大小超过10 000字节,而只有34%的Webshell大于10 000字节。同时,86%的合法网页HTML文件行数大于300行,而只有42%的Webshell包含超过300行的HTML代码。

(2) 编码特征:包括网页本身声明的编码以及HTML解析器检测出的实际编码。一般而言,合法的网页往往会声明自己的编码,特别对于中国高校而言,大多数会声明为中文编码如"gbk"、"gb2312"或者"gb18030"。而境外的入侵者为了显示方便,会在Webshell中使用本国的语言编码,例如"en"或者"ru"等。有时会出现网页不声明编码或者声明的编码和实际编码不一致的情况,因此还使用了HTML解析器来探测网页实际的编码。笔者统计了常见Webshell的编码情况,并与中国高校网页的编码作对比,结果如表1和表2所示,可以看出大多数的Webshell都没有声明编码,而合法网页往往都声明了正确的编码,并且中文编码占到了一定比例。

(3) 语言特征:即网页文本使用的自然语言。几乎所有的Webshell都使用英语作为主要的现实语言,少量的会使用本国语言例如俄语、希腊语、土耳其语和中文等。统计结果如表3所示,其中标注的语言除了un和xxx代表无法识别的语言,其

余皆为 ISO 639 标准规定的缩写。由表可以看出,一半以上的 Webshll 都由英语编写,而合法网页中使用中文的占到了绝大多数。

(4) 标签特征:即页面使用的 HTML 标签种类数,以及具体使用了哪些标签。由于 Webshell 页面的编写往往不考虑美观和整洁,因此标签的使用要远远少于正常的合法页面,据统计,在 HTML5 规定的 117 种标签中,90%以上的 Webshell 使用了少于 15 种标签,而合法页面往往至少使用了 20 种标签。

(5) 静态文件特征:即页面是否包含针对图片文件、样式文件和脚本文件的引用。静态文件对于页面有锦上添花的效果,例如图片会增加信息量,美化页面、样式可以方便网页制作者统一风格,脚本能够增加页面的功能。而对于 Webshell 来说,这 3 类静态文件完全是多余的。本文将静态文件的引用情况分为 4 类:①无。页面完全没有引用任何静态文件;②本地。页面包含了对本地静态文件的应用;③外部。页面包含了对外部静态文件的应用;④皆有。同时符合本地和外部的状况。

表 1　网页声明编码分类统计(Top 5)

Tab. 1　Declared charset of webpages

Webshell		合法网页	
声明编码	百分比/%	声明编码	百分比/%
None	61.50	utf-8	47.11
windows-1251	19.91	None	27.07
windows-1256	6.19	gb2312	25.56
utf-8	4.42	us-ascii	0.12
windows-1254	3.54	iso-8859-1	0.06

表 2　网页实际编码分类统计(Top 5)

Tab. 2　Actual charset of webpages

Webshell		合法网页	
实际编码	百分比/%	实际编码	百分比/%
utf-8	60.18	utf-8	72.37
windows-1251	19.91	gb18030	25.26
windows-1252	7.08	windows-1252	2.08
windows-1256	6.19	us-ascii	0.12
windows-1254	3.54	iso-8859-1	0.06

表 3　网页语言分类统计(Top 5)

Tab. 3　Language of webpages

Webshell		合法网页	
语言	百分比/%	语言	百分比/%
en	54.87	zh	83.27
un	30.09	en	8.28
xxx	4.42	un	2.37
tr	2.65	ru	2.27
ru	1.77	el	2.08

3.2.2　文本特征

本文所选取的文本特征和典型的网页分类方法类似。网页数据是一种半结构化的数据,这与纯文本数据不同。对网页中的任一特征而言,有两个影响特征权值的因素:(1) 词是否出现在 HTML 文档里,(2) 出现的词在该文档中位置。因此,也特别考虑了出现在某些特殊位置的文本。经过分析,本文选取的文本特征如下

(1) Title:网页的标题由〈title〉标签确定,用来说明网页的主要内容。Webshell 的标题往往含有入侵者个性内容,和高校页面的标题有较大区别。

(2) Meta:网页的元信息通常包括 keywords 和 description 等,反映了网页的摘要内容。

(3) 关键词:通常的网页分类采用 bag-of-words model 即词袋模型,将网页的文本信息分词后直接作为词袋的元素。为了减少分类的维度,本文提取网页的关键词作为词袋的元素。

3.2.3　词袋模型和关键词提取

词袋模型最早出现在 NLP 和 IR 领域。在信息检索中,词袋模型假定对于一个文本,忽略其词序、语法和句法,将其仅仅看做是一个词集合,或者说是词的一个组合。文本中每个词的出现都是独立的,不依赖于其他词是否出现,或者说当这篇文章的作者在任意一个位置选择一个词汇都不受前面句子的影响而可以独立选择。本文使用词袋模型对于网页 Web 的文本特征进行特征提取。

为了减少特征维度,需要提取文本的关键词。目前流行的关键词提取算法是 TF-IDF。TF 指的是 Term Frequency,即词频的英文缩写,即某个关键词在文本中出现的频率。IDF 是 Inverse Document Frequency,即逆文本频率指数,表达式为 $\log\left(\frac{D}{D_w}\right)$,其中 D 为全部网页数,D_w 为出现关键词 w 的网页数。利用 TF-IDF,便可以计算所有关键词对于整个文本的相关度,TF-IDF 值越大说明该单词相对文本的相关性越大,其计算公式为

$$\text{TF-IDF}_w = \text{TF}_w \times \text{IDF}_w$$

4　实　验

4.1　实验数据集

实验用的 Webshell 样本来自 Github 上的 Webshell 收集项目。由于一句话木马不会生成 HTML 页面,因此实验只考虑了大马和小马。在所有可以正常使用的 226 个 Webshell 文件中,经过人工分类整理,将这些脚本文件分为 3 类:大马

(需登录)、大马(不需登录)和小马,样本数分别如表4所示。

表4 Webshell样本分类

Tab.4 Categories of webshell samples

Webshell类型	样本数量
大马(需登录)	51
大马(不需登录)	153
小马	22
总计	226

实验用的合法网页来自江苏省网内的高校网站,使用网页爬虫下载了1 816个网页。由于Webshell需要由攻击者上传,因此合法网页上不存在至Webshell的链接,即网页爬虫无法通过链接抓取到Webshell页面。所以认为这1 816个网页全部是合法网页。

下文中将Webshell样本称为恶意样本,合法样本称为良性样本。

4.2 实验方法

根据前文的叙述,Webshell的检测实验。分为特征提取、训练和测试3部分。

为了能够评估实验效果,首先需要训练样本和实验样本。本文使用分层抽样的方法,即将上述恶意和良性样本分别随机抽取2/3作为训练数据,其余1/3作为测试数据。为了更好地体现实验效果,本文重复了10次随机抽取工作,共得到了10个训练数据样本以及10个对应的测试数据,下文将统称为样本1～样本10。

接着根据第3节提出的特征对于这10个样本分别提取特征,并保存在文件中。

对于机器分类的训练和测试,本文使用了libsvm[7]和liblinear[11]这两个开源的SVM库,它们使用的数据格式是相同的。其中libsvm使用RBF核的SVM,liblininear使用线性核函数的SVM。根据官方文档,libsvm适合一般的分类问题,而liblinear适合维度较大的分类问题。因此,对于实验来说,由于实验维度远远大于样本数,因此理论上liblinear的分类效果要好于libsvm。将上述提取的特征转换为libsvm和liblinear可以识别的格式,并且使用libsvm自带的参数选择工具grid.py选择分类时使用的参数:对于libsvm,即RBF核函数的SVM,其惩罚系数c取2.0,gamma值g取3.051 757 812 5e−05;对于liblinear,即线性核函数的SVM,其c取0.031 25,不使用gamma值。

4.3 效果评估

本实验中将Webshell表示为1,普通合法网页表示为0,分类的所有情况如表5所示。

表5 Webshell检测算法的分类情况

Tab.5 Classification results of webshell detection algorithm

		预测	
		1	0
实际	1	True positive(TP)	False negative(FN)
	0	False positive(FP)	True negative(TN)

为了评估实验效果,采用了准确率、召回率和误报率3项指标。其中准确率定义为$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$,表示分类的准确度;召回率定义为$\mathrm{RECALL}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$,表示针对Webshell的检出率;误报率定义为$\mathrm{FALLOUT}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$,表示算法的错误命中率。

对于第3节得到的10个样本集分别进行了两种核函数的SVM分类实验,表6和表7分别表示了使用线性核的SVM分类结果和使用RBF核的SVM分类结果。

表6 线性SVM分类效果

Tab.6 Classify effect of linear SVM

%

分类	准确率	召回率	误报率
1	98.69	92.00	0.40
2	98.53	86.84	0.00
3	96.62	69.74	0.00
4	96.03	64.47	0.00
5	97.21	76.25	0.00
6	97.06	73.68	0.00
7	97.35	76.32	0.00
8	97.21	75.00	0.00
9	98.38	85.53	0.00
10	97.35	76.32	0.00
平均	97.44	77.62	0.04

表7 RBF-SVM分类效果

Tab.7 Classify effect of RBF-SVM

%

分类	准确率	召回率	误报率
1	96.79	73.00	0.00
2	96.76	71.05	0.00
3	97.50	77.63	0.00
4	95.88	63.16	0.00
5	96.91	73.75	0.00
6	97.21	75.00	0.00
7	98.09	82.89	0.00
8	96.76	71.05	0.00
9	97.21	75.00	0.00
10	97.35	76.32	0.00
平均	97.05	73.89	0.00

由表中可以看出，对于绝大多数样本，两种核函数都取得了较好的准确率，并且拥有极低的误报率，说明检测算法对于合法网页的识别较为准确。而在召回率方面，不同样本出现了不同的效果：对于线性 SVM，召回率最高的可以达到 92%，最低只有 64%，相差较大；RBF-SVM 表现整体弱于线性 SVM，召回率的的变化幅度在 63%～82%之间；两种核函数的平均召回率都在 75%左右。两种核函数的对比可见图 1。

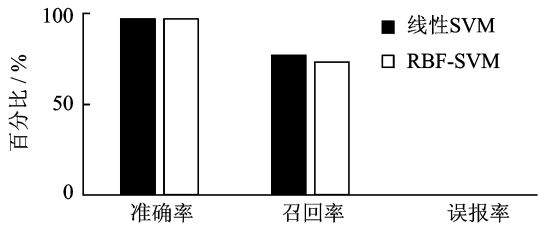

图 1　不同核函数分类效果比较

Fig. 1　Classify effect of different kernel functions

因此可以得出结论：SVM 分类算法可以较好地完成 Webshell 后门的检测，使用线性核函数的 SVM 在准确率和召回率方面都要好于使用 RBF 核函数的 SVM。这也符合前文的推断，此外两种核函数的误报率都较低。

将本文提出的方法与市面上现有的检测工具作对比（见表 8）。本文选取了 9 种不同的软件：包括 7 种在市场上占有率较高的杀毒软件和 2 种专用的 Webshell 检测工具：Web Shell Detector 和 D 盾_Web 查杀。对比实验分为白盒实验和黑盒实验两种。黑盒实验即在不获知脚本源代码的情况下仅仅根据 HTML 文档进行检测；而白盒实验则是通过脚本源代码进行检测。根据之前的讨论，本文提出的方法使用黑盒检测。

表 8　与其他检测软件的对比结果

（Webshell 样本总数：226）

Tab. 8　Comparison with other detection software

项目	黑盒检出数	黑盒检测率/%	白盒检出数	白盒检测率/%
360 杀毒	0	0.00	0	0.00
Avast	54	15.93	189	83.63
Avira	64	28.32	165	73.01
MSE	15	6.64	59	26.11
QQ 安全管家	15	6.64	13	5.75
百度杀毒	19	8.41	136	60.18
金山毒霸	17	7.52	2	0.88
D 盾_Web 查杀	1	0.44	218	96.46
Web Shell Detector	19	8.41	149	65.93
本文方法	171	75.67	N/A	N/A

由表 8 与其他检测软件的对比结果（Webshell 样本总数为 226）可以看出，如果仅仅根据 HTML 文档的信息，即黑盒实验，绝大多数工具无法很好地检测出 Webshell，而本文提出的方法可以达到 75%的检测率；而对于白盒实验，即针对源代码的检测，各个工具的检测效果不一，大多数在 60%～90%之间。由此可以得出结论：由于本文的方法并不是简单的特征匹配，而是根据 Webshell 页面的结构和文本特点来进行检测；因此在不获知脚本源代码（即黑盒检测）的情况下，本文提出的检测方法可以达到其他检测软件白盒检测的效果。

5　结束语

本文分析归纳了 Webshell 的页面特点，总结归纳了其 HTML 的结构特征和文本特征，提出并实现了基于 SVM 支持向量机的检测方法，该模型可以在未知源代码的情况下，仅仅通过 HTML 页面检测 Webshell。经过实验表明，这种黑盒检测的方法达到了传统基于特征匹配的白盒检测方法的检测效果。

致谢　*我们向对本文工作给予支持和建议的同行，尤其是江苏省网络技术重点实验室的老师和同学表示感谢。*

参考文献：

[1] 胡建康，徐震，马多贺，等. 基于决策树的 Webshell 检测方法研究[J]. 网络新媒体技术，2012，1(6)：15-19.
Hu Jiankang，Xu Zhen，Ma Duohe ，et al. Research of webshell detection based on decision tree[J]. Journal of Network New Media，2012，1(6)：15-19.

[2] Yang C H，Shen C H. Implement web attack detection engine with snort by using modSecurity core rules[C]// Fourth the E-Learning and Information Technology Symposium (EITS 09). [S. l.]：Graduate Institute of Information and Computer Education，National Kaohsiung Normal University Kaohsiung，2009.

[3] McCallum A，Nigam K. A comparison of event models for naive bayes text classification[R]. AAAI-98，1998：41-48.

[4] Slattery S. Hypertext classification[D]. Pittsburgh: School of Computer Science, Carnegie Mellon University, 2002.

[5] Craven M, Slattery S. Relational learning with statistical predicate invention: Better models for hypertext[J]. Machine Learning, 2001, 43(1/2): 97-119.

[6] Joachims T. Text categorization with support vector machines: Learning with many relevant features[C] // Chemnitz: European Conference on Machine Learning(ECML). 1998:137-142.

[7] Chang Chihchung, Lin Chihjen. LIBSVM: A library for support vector machines[EB/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm,2011.

[8] 刘永芬. 支持向量机在入侵检测中的应用[D]. 厦门:福建师范大学, 2010.
Liu Yongfen, Application ofsupport vector machine to network intrusion detection[D]. Xiamen: Fujian Normal University, 2010.

[9] 孙建涛, 沈抖, 陆玉昌, 等. 网页分类技术[J]. 清华大学学报: 自然科学版, 2004, 44(1): 65-68.
Sun Jiantao, Shen Dou, Lu Yuchang, et al. Web document classification techniques[J]. Journal of Tsinghua University: Science and Technology, 2004, 44(1): 65-68.

[10] 苏金树, 张博锋, 徐昕. 基于机器学习的文本分类技术研究进展[J]. 软件学报, 2006, 17(9): 1848-1859.
Su Jinshu, Zhang Bofeng, Xu Xin. Advances in machine learning based text categorization[J]. Journal of Software, 2006, 17(9):1848-1859.

[11] Fan R E, Chang K W, Hsieh C J, et al. Liblinear: A library for large linear classification[J]. The Journal of Machine Learning Research, 2008(9): 1871-1874.