doi 10.3969/j.issn.1671-1122.2014.05.002

Linux 下基于 SVM 分类器的 WebShell 检测方法研究

孟正，梅瑞，张涛，文伟平
（北京大学软件与微电子学院，北京 100871）

摘 要：WebShell 是一种常见的网页后门，它常常被攻击者用来获取 Web 服务器的操作权限。文章首先分析了 Linux 下 WebShell 的实现机理，描述了 WebShell 的常见特征和特征混淆方法，然后以此为基础，提出了一种基于 SVM 分类器的检测方法，并在仿真平台下对其予以实现。文章从准确度、特定度和灵敏度 3 个方面比较了基于 SVM 分类器的 WebShell 检测方法、基于特征匹配的 WebShell 检测方法和基于决策树的 WebShell 检测方法。实验结果表明，文章提出的方法能够准确、高效地对 WebShell 进行检测。

关键词：WebShell 检测；SVM 分类器；特征提取
中图分类号：TP309 文献标识码：A 文章编号：1671-1122（2014）05-0005-05

Research of Linux WebShell Detection based on SVM Classifier

MENG Zheng, MEI Rui, ZHANG Tao, WEN Wei-ping
(*School of Software&Microelectronics, Peking University, Beijing* 100871, *China*)

Abstract: WebShell is a common webpage back door, which can be used by attackers to obtain Web server permissions. The realization mechanism of Linux WebShell is analyzed, the common characteristics and the characteristic mixed method are described in this paper. On this basis, a detection method based on SVM classifier is put forward and realized. From three aspects of accuracy, specificity and sensitivity, the WebShell detection methods individually based on SVM classifier, characteristic matching and decision tree are compared. The experimental result shows that the method proposed in this paper can detect WebShell accurately and efficiently.

Key words: WebShell detection; SVM classifier; characteristic extraction

0 引言

随着计算机网络技术的不断发展，以 B/S 架构为基础的 Web 应用程序逐渐普及，包括应用在政府门户网站、游戏网站、团购网站以及网上商城等。不同的 Web 系统研发人员在技术水平上存在差异，部分程序员在设计、编码过程中对安全问题考虑不周，造成 Web 应用问题出现安全问题。常见的安全威胁有 SQL 注入漏洞、跨站脚本攻击、提交表单漏洞、上传文件漏洞等。当前，Web 服务器所面临的安全问题已经十分严重，甚至对网络服务的正常运行产生了威胁。因此，检测服务器的漏洞，保障服务器的安全是十分必要的。

WebShell 是一种常见的网页后门，它常常被攻击者用来获取 Web 服务器的操作权限。攻击者在进行网站入侵时，通常会将 WebShell 文件与 Web 目录下的正常网页文件放置在一起，然后通过浏览器访问 WebShell 文件，从而获取命令执行环境，最终达到控制网站服务器的目的 [1,2]。当网站服务器被控制后，就可以在其上任意查看数据库、上传下载文件以及执行任意程序命令等。WebShell 与正常网页具有相同的运行环境和服务端口，它与远程主机是通过 www（80）端口进行数据交换的，因此能够很容易地避开杀毒软件的检测和穿透防火墙 [3]。另外，WebShell 是纯文本程序，相对于二进制编码程序，它在使

收稿日期：2014-04-10
基金项目：国家自然科学基金 [61170282]
作者简介：孟正（1990-），男，河北，硕士研究生，主要研究方向：漏洞分析和漏洞挖掘；梅瑞（1984-），男，安徽，硕士研究生，主要研究方向：系统与网络安全、软件安全漏洞分析、信息系统逆向工程等；张涛（1987-），男，江西，硕士研究生，主要研究方向：系统与网络安全、软件安全漏洞分析；文伟平（1976－），男，湖南，副教授，博士，主要研究方向：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等。

用上更加灵活多变，也易于进行混淆，这就使得基于特征匹配的方法很难对 WebShell 进行准确检测[4]。本文详细分析了 Linux下WebShell 的实现机理，描述了 WebShell 的特征，然后以此为基础，提出了一种基于 SVM 分类器的检测方法，并在仿真平台下对其予以实现。

1 WebShell 实现机理

1.1 WebShell概述

WebShell 可以协助管理员对远程服务器进行操作，与此同时，攻击者也会通过恶意 WebShell 实现对网站服务器的控制。对于攻击者来说，WebShell 就是一个后门程序，其编写语言通常是 ASP、PHP 或 JSP 等网页脚本，而网页脚本具有能够创建动态交互站点的特性[5]。当攻击者对一个网站实施入侵后，首先在网站服务器的 Web 目录下放置一些脚本程序，接下来就能够通过 Web 页面对网站服务器进行控制。通过 WebShell 进行服务器管理时，是不会在系统安全日志中留下记录的，普通管理员很难发现入侵痕迹。

1.2 WebShell攻击原理

WebShell 攻击有以下 5 种方式[6,7]：

1）首先通过系统前台上传 WebShell 脚本到网站服务器，此时，网站服务器会向客户端返回上传文件的整个 URL 信息；然后通过 URL 对这个脚本进行访问，使其可以执行；最终导致攻击者能够在网站的任意目录中上传 WebShell，从而获取管理员控制权限。

2）攻击者利用管理员密码登录进入后台系统，并借助后台管理工具向配置文件写入 WebShell，允许任意脚本文件上传。

3）通过数据库的备份和恢复功能获取 WebShell。在数据库备份时，可以将备份文件的扩展名更改为 .asp 类型。

4）系统中的其他站点遭受攻击，或者搭载在 Web 服务器上的 ftp 服务器遭受攻击之后，被注入了 WebShell，这些都会导致整个网站系统被感染。

5）攻击者利用 Web 服务器漏洞直接对其进行攻击，从而获得控制权限。

在上述攻击方式中，第 4 种方式可以导致整个网站系统被感染，危害性较大。现对 WebShell 感染[8]过程进行描述：

1）攻击者首先通过 SQL 注入、跨站脚本攻击等方式得到上传权限，然后将 WebShell上传至服务器。

2）通过 WebShell 完成对服务器的控制，实现植入僵尸木马、篡改网页以及获取敏感信息等恶意功能。

3）植入攻击木马，使其作为攻击“肉鸡”对整个网站系统进行感染。

1.3 Linux下WebShell攻击

WebShell 反弹命令行 Shell 的方式在 Linux 操作系统下Web 服务器入侵提权过程中得到了广泛应用。在 Linux 下，WebShell 可以执行命令，然而溢出却必须在交互环境中进行，否则即使提权成功，也不能获得完美利用。因此，为了完成 WebShell 攻击，需要反弹一个 Shell 命令行窗口，在命令行终端下执行溢出并进行提权。

多数 Linux 操作系统下的 PHP WebShell 都通过反弹连接功能获得一个继承当前 WebShell 权限的 Shell 命令行窗口。在使用反弹连接功能前，需要首先使用 NC 工具对一个未使用的端口进行监听，然后选择反弹连接方式。设置完成后，即可向主机返回一个 Shell 命令行窗口[9]，如图 1 所示。

```
listening on [any] 8080 ...
119.1.44.32: inverse host lookup failed: h_errno 11004: NO_DATA

connect to [119.1.58.126] from (UNKNOWN) [119.1.44.32] 55171: NO
_DATA
uname -a
Linux xiaoyao-desktop 2.6.28-13-generic #45-Ubuntu SMP Tue Jun 3
0 19:49:51 UTC 2009 i686 GNU/Linux
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

图1 WebShell反弹命令行Shell

2 WebShell 特征

2.1 WebShell常见特征

WebShell 通常具有以下常见特征[1,8,10]。

1）文件操作：查看文件属性、文件搜索、文件浏览列表、下载文件、上传文件、更改文件名、删除操作、复制操作等。

2）Web 应用程序被加密：使用 str_rotl3、gzuncompress、gzinflate、base64_decode、javascript.encode、jscrip.encode、vbscript 等方法加密一些敏感代码。

3）挂马功能。

4）Web 服务支持组件查询：Scripting.FileSystemObject、Adodb.Stream、wscript.shell、ADOX.Catalog。

5）查看系统信息：查看磁盘信息、系统环境变量、系统用户信息、本机 IP。

6）执行应用程序：如 Wscript.shell。

7）注册表操作：写入注册表、删除注册表、读取注册表和打开注册表。

8）网络功能：端口扫描。

9）数据库操作 数据库的连接、建立、压缩、添加、查询、删除和更改。

10）目录操作：更改目录名、删除目录。

2.2 WebShell关键代码分析

通过对大量 WebShell 程序源代码进行分析，可知常见的 WebShell 特征代码 [11] 有以下几种。

1）建立 Scripting.FileSystemObject 对象 ObjFSO，用于执行 FSO 对象，对文件和文件夹进行删除、新建、重命名、编辑等操作。代码如下：

Set ObjFSO = Server.CreateObject(“ Scripting.FileSystemObject ”)

或者直接指定注册表键值：

<object runat =“ server ”id =“ ObjFSO ”classid =“ clsid:0D43FE01-F093-11CF-8940-00A0C9054228 ” > </object>

2）获取服务器的基本信息，包括 Web 服务器版本、服务器操作系统、服务器 IP、服务器名称以及服务器端口号等信息。

服务器版本：IP Request. ServerVariables (“ LOCAL_ADDR ”)。

服务器 IP：IP Request. ServerVariables (“ LOCAL_ADDR ”)。

服务器操作系统：Request. ServerVariables (“ OS ”)。

3）建立 ADODB.Stream 对象 ObjStream，对服务器文件进行上传和下载。代码如下：

Set ObjStream = Server. CreateObject(“ ADODB.Stream ”)

或者直接指定注册表键值：

<Object runat =“ server ”id =“ ObjStream ”classid =“ clsid: 00000566-00000-0010-8000-00AA006D2EA4 ” > </object>

4）建立 Wscript.SHELL 对象 ObjShell，对注册表和任意程序指令进行读写。代码如下：

Set ObjShell = Server.Create(“ Wscript.SHELL ”)

或者直接指定注册表键值：

<object runat =“ server ”id =“ ObjShell ”classid =“ clsid:72C24DD5-D70A-438B-8A42-98424B88AFB8 ” > </object>

2.3 WebShell特征混淆

目前 Linux 操作系统下主流的 WebShell 检测工具都采用特征匹配的方式，这种检测方式依赖于特征库，检测精度差，智能化程度低，对于经过变形处理的 WebShell 无能为力。隐藏 WebShell 特征的方法主要有以下几种。

1）插入注释信息：在 PHP 代码段中插入注释信息，形如“ /*…*/ ”，注释信息不会影响 WebShell 的正常功能，但会对检测工具形成干扰。

2）字符串加密：对字符串等关键信息预先加密，将其可逆地映射成为另外一个字符串，在调用时动态解密。

3）外形混淆：在不影响功能的前提下，对程序中的函数名、变量名、常量名等标识做词法上的变换，从而阻止攻击者理解程序 [6]。

4）逻辑混淆：对程序代码的控制流程进行转换，使其变得复杂，增加程序代码分析的难度。常用的逻辑混淆方法有插入分支、插入循环和破坏循环等 [12]。

5）字符串拆分：WebShell 在对系统控件进行调用时，为避免控件名称被检测，先将其作为一个字符串进行拆分，然后再拼接起来。

6）文件包含：为防止 WebShell 特征过于集中，可以先将一个网页拆分成多个，再通过包含操作（如 include）进行整合。

3 基于 SVM 分类器的 WebShell 检测

3.1 检测框架

基于 SVM 分类器的 WebShell 特征检测模型对训练网页和测试网页分别进行特征提取，从而得到训练特征和测试特征，经 SVM 分类器完成分类操作，最终输出结果。模型框架如图 2 所示，它主要包括特征提取、SVM 分类器和输出结果 3 个部分。

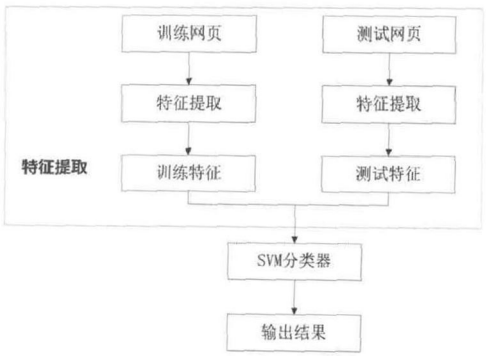

图2 基于SVM分类器的WebShell特征检测模型框架

3.2 特征提取

由图 2 可知，特征提取是整个检测框架的基础，选取特征的好坏直接关系检测结果的优劣。因此，在进行特征选取时，首先应对 Web 页面本身进行充分考虑，使得选取的特征能够很好地表现出静态页面。其次，选取的特征还应该具有动态特点，可以体现出页面所进行的操作[1]。

如果提取网页的全部特征进行处理，则无法检测变形的 WebShell，也会因为特征过多而对效率产生影响。如果检测特征过少，则有可能产生误报。本文主要从页面属性和页面操作两个方面对网页特征进行提取，页面属性特征包括页面长度、代码行数、注释个数等，页面操作特征包括加解密函数调用、system/eval/exec/shell_exec () 调用、字符操作函数调用、系统函数调用、文件操作、ftp 操作、数据库操作、Activex 控件调用等。

3.3 SVM分类器

SVM（支持向量机）是 1995 年由 Vapnik 和 Cortes 首先提出的，这种方法在解决非线性小样本模式识别时优势显著，其查准率和查全率几乎超过了现有的全部方法。此外，SVM 分类方法具有很好的泛化能力[7]。

当前主要有两种 SVM 分类器：1）一对一。2 类 SVM 分类器。2）一对多。将 2 类 SVM 分类器构造出 k 个，使每个类对应其中一个。本文只需对 WebShell 进行检测，因此选用了一对一的 SVM 分类器。特征提取后，针对每个 Web 页面均可以得到一个 10 维的特征向量，每个页面特征对应一个维度。下面通过 SVM 分类器建立特征向量和 WebShell 之间的对应关系[13,14]。

SVM 主要是通过结构风险最小化原则来解决分类问题的，它通过一个最优分类超平面将两类数据以最大间隔分开，如图 3 所示。

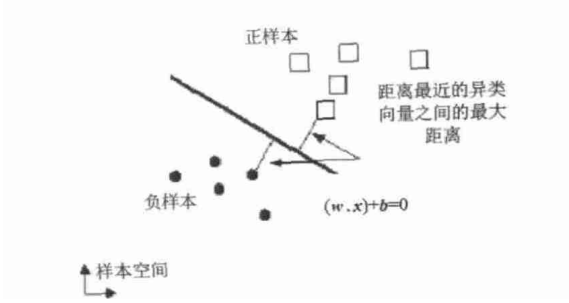

图3 最优分类超平面

设样本集 S 是线性可分的（$S=\{(x_i,y_i)\mid i=1,...,n\}$），其中 $x_i\in R^d$，$y_i=\{1,-1\}$ 表示 x_i 所对应的类别。$g(x)=wx+b$ 是 d 维空间中线性判别函数的一般形式，它对应的分类面方程为 $wx+b=0$。对 $g(x)$ 进行归一化操作，使得两类样本均满足如下条件：$|g(x)|\geq 1$。此时，分类间隔为 $\frac{2}{\|w\|}$。$\|w\|$ 越大，则分类间隔越小。为使分类面能够对全部样本进行正确分类，需满足以下条件：

$$y_i[(wx)+b]-1\geq 0\ (i=1,2,...,n) \quad \cdots\cdots (1)$$

使得上述条件均满足的分类面也可称为最优分类面，最优分类面问题等价于在满足公式（1）的前提下，求目标函数：

$$\phi(w)=\frac{1}{2}\|w\|^2=\frac{1}{2}(w\cdot w) \quad \cdots\cdots (2)$$

的最小值。与最优分类面的超平面 H_1、H_2 保持平行且过两类样本中距分类面最近的点的训练样本可以使公式（1）中的等号成立，这些训练样本被称为支持向量。对于线性不可分样本，引入了惩罚因子 C 和松弛变量 ξ_i，此时公式（2）可改写为：

$$\phi(w,\xi_i)=\frac{1}{2}(w\cdot w)+C(\sum_{i=1}^{N}\xi_i) \quad \cdots\cdots (3)$$

此时，引入 Lagrange 乘子 $\alpha_i(i=1,2,...,n)$，将 SVM 分类问题转换为有约束的二次函数极值问题，从而对最优分类面进行求解。最终解为 $w=\sum_i \alpha_i y_i x_i$，则最优分类函数可重写为[2,15,16]：

$$f(x)=sign\{(w\cdot x)+b\}=sign\{\sum_{i=1}^{N}\alpha_i y_i(x_i\cdot x)+b\} \quad \cdots\cdots (4)$$

3.4 输出结果

基于 SVM 分类器的 WebShell 检测是通过 PHP 编程语言实现的。在 CentOS 5.2 和 Ubuntu 12.04 上分别搭建 PHP+MySQL+Apache+Linux 开发环境，将训练网页放置于 /var/training 目录下，测试网页放置于 /var/www 目录下。在浏览器地址栏中输入网址 http://localhost/shelldetect.php，则会输出检测结果。检测结果是以网页的形式显示的，其中包含功能描述、测试文件个数、可疑文件个数以及可疑文件特征等。

4 实验结果与分析

4.1 实验数据

采用 ASP、PHP 和 JSP 3 种编程语言编写的 WebShell

具有极其相似的特征，我们只需选择其中一种进行测试即可。本文共选择了 142 个 PHP WebShell 样本作为训练网页进行测试，它们包含了常见 WebShell 的所有特征。另外，选择测试网页 850 个。

4.2 评价标准

在对 WebShell 进行检测时，可能存在以下 4 种情况[1]：

1）预测当前页面是 WebShell，实际上确实如此，这种情况记录为 TP（True Positive）。

2）预测当前页面是 WebShell，实际上不是，这种情况记录为 FP（False Positive）。

3）预测当前页面不是 WebShell，事实则不然，该页面 WebShell，这种情况记录为 FN（False Negative）。

4）预测当前页面不是 WebShell，事实上该页面确实不是 WebShell，这种情况记录为 TN（True Negative）。

WebShell 检测的评价标准主要有 3 个：准确度、特定度和灵敏度。准确度（Accuracy）定义为：

$$Accuracy=\frac{TN+TP}{TN+FN+TP+FP} \quad (5)$$

特定度（Specificity）定义为：

$$Specificity=\frac{TN}{FP+TN} \quad (6)$$

敏感度（Sensitivity）定义为：

$$Sensitivity=\frac{TP}{TP+FN} \quad (7)$$

4.3 实验结果

在 CentOS 5.2 上对 WebShell 进行检测，可疑文件特征主要涉及文件名、文件属主、文件大小、文件权限、文件最后访问时间、文件最后修改时间、哈希值、使用的可疑函数以及指纹信息等。

现从准确度、特定度和灵敏度 3 个方面对基于 SVM 分类器的 WebShell 检测方法、基于特征匹配的 WebShell 检测方法和基于决策树的 WebShell 检测方法进行比较，结果如表 1 所示。

表1 不同检测方法实验结果（单位%）

	准确度	特定度	灵敏度
基于 SVM 分类器的 WebShell 检测方法	92.81	99.64	98.62
基于特征匹配的 WebShell 检测方法	87.45	96.44	92.86
基于决策树的 WebShell 检测方法	91.07	99.53	98.03

由表 1 可知，基于 SVM 分类器的 WebShell 检测方法在准确度、特定度和灵敏度 3 个评价标准上均优于另外两种方法。在 3 种方法中，基于 SVM 分类器的 WebShell 检测方法最优，基于决策树的方法稍次之，基于特征匹配的方法最劣。

5 结束语

本文详细分析了 Linux 下 WebShell 的实现机理，描述了 WebShell 的常见特征和特征混淆方法，以此为基础，提出了一种基于 SVM 分类器的检测方法，并在仿真平台下对其予以实现。本文从准确度、特定度和灵敏度 3 个方面比较了基于 SVM 分类器的 WebShell 检测方法、基于特征匹配的 WebShell 检测方法和基于决策树的 WebShell 检测方法。实现结果表明，本文提出的方法能够准确、高效地对 WebShell 进行检测。（责编 马珂）

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，(06)：15-19.

[2] 袁勋，吴秀清，洪日昌，等 . 基于主动学习 SVM 分类器的视频分类 [J]. 中国科学技术大学学报，2009，39（05）：473-478.

[3] Xiao Yao. Large and Medium-sized Network Intrusions Cases Research[J]. Publishing House Of Electronics Industry, 2010,(10):301-310.

[4] J. Ross Quinlan. C4. 5: programs for machine learning[M]. San Francisco: Morgan Kaufmann, 1993.

[5] Yung-Tsung Hou, Yimeng Chang, Tsuhan Chen.Malicious web content detection by machine learning[J]. Expert Systems with Applications,2010,37(1):55-60.

[6] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines[C]//Proceedings of IEEE Workshop on Neural Networks for Signal Processing. Amelia Island, USA: IEEE Press, 1997: 276-285.

[7] Lin H T, Lin C J, Weng R C. A note on Plat tps probabilistic outputs for support vector machines[J]. Machine Learning, 2007, 68 (3): 267-276.

[8] Brinker K. On multiclass active learning with support vector machines[C]//Proceedings of European Conference on Artificial Intelligence. 2004: 969-970.

[9] Yuan X, Lai W, Mei T, et al. Automatic video genre categorization using hierarchical SVM[C]//IEEE International Conference on Image Processing. Atlanta: IEEE Press, 2006: 2905-2908.

[10] Tong S, Chang. E Support vector machine active learning for image ret rieval[C]//Proceedings of the 9th ACM International Conference on Multimedia. Ottawa, Canada: ACM Press, 2001, 9: 107-118.

[11] 唐银凤，黄志明，黄荣娟，等 . 基于多特征提取和 SVM 分类器的纹理图像分类 [J]. 计算机应用与软件，2011，28（06）：22-25.

[12] 骆剑承，周成虎，梁怡，等 . 支撑向量机及其遥感影像空间特征提取和分类的应用研究 [J]. 遥感学报，2002，6（01）：71-78.

[13] 刘冰 . 多类 SVM 分类算法的研究和改进 [J]. 电脑知识与技术，2007：1590-1593.

[14] 陈光英，张千里，李星 . 基于 SVM 分类机的入侵检测系统 [J]. 通信学报，2002，23（05）：51-56.

[15] CORTES C, VAPNIK V. Support vector network[J]. Machine Learning, 1995, (20):273-297.

[16] 李昆仑，黄厚宽，田盛丰 . 一种基于有向无环图的多类 SVM 分类器 [J]. 模式识别与人工智能，2003，16（02）：164-168.