基于随机森林的命令混淆绕过检测研究

戚臻彦，孙永清
（公安部第三研究所，上海　200030）

摘　要：运维安全管理设备中的“命令过滤”功能只能过滤黑名单中的恶意代码，而无法有效识别并阻止使用特殊方法绕过该功能的行为。针对这一问题，提出了一种基于随机森林的算法，可以准确识别含有恶意代码的命令执行语句。首先，介绍了四种命令混淆绕过方法，它们用来规避黑名单中的关键词并进行命令执行。然后，为了解决这些风险，在模型的特征选择阶段将命令混淆代码纳入考虑范围，利用多种特征对模型进行训练并调整特征权重，以提高模型检测中对使用命令混淆攻击的识别率和准确度。实验结果表明，该方法能够及时识别并应对命令混淆攻击，从而更好地保证服务器安全运行。

关键词：命令混淆；运维管理设备；随机森林；网络安全
中图分类号：TP393　　**文献标识码：**A　　**DOI：**10. 19358/j. issn. 2097 – 1788. 2023. 06. 011
引用格式：戚臻彦，孙永清. 基于随机森林的命令混淆绕过检测研究［J］. 网络安全与数据治理，2023，42(6)：66 – 70.

Research on command obfuscation bypass detection based on random forest algorithm

Qi Zhenyan，Sun Yongqing
(The Third Research Institute of the Ministry of Public Security，Shanghai 200030，China)

Abstract：The "command filtering" function in operation and maintenance security devices can only filter malicious code in the blacklist，and cannot effectively identify and prevent the use of special methods to bypass this function. To address this problem，this paper proposes an algorithm based on random forest，which can accurately identify command execution statements containing malicious code. Firstly，this paper introduces four methods of command obfuscation bypass，which are used to evade keywords in the blacklist and perform command execution. Then，in order to solve these risks，the command obfuscation code is taken into account in the feature selection stage of the model，and various features are used to train and adjust the weights of the random forest model，so as to improve the recognition rate and accuracy of the model detection for adding command obfuscation attacks. The experimental results show that the method proposed in this paper can timely identify and deal with command obfuscation attacks，thus better ensuring the secure operation of servers.

Key words：command obfuscation；operation and maintenance management equipment；random forest；network security

0　前言

为了应对当下日益严峻的网络安全问题，各单位广泛使用运维安全管理设备（也称为堡垒机）来解决账号权限集中、审计难度大、运维管理困难等问题。《GB/T 22239—2019 信息安全技术 网络安全等级保护基本要求》[1]标准提出了安全通信网络、安全计算环境和安全管理中心等新的要求。运维安全管理设备的各项功能均能快速满足企业单位的需求，具有快速部署和管理权限分级等优点，使各单位在满足等级保护标准的同时，能够达到维护自身网络安全运行的目的。尽管运维安全管理设备的相关安全功能和技术不断提高，但随着对网络安全的深入研究仍会发现其存在诸多安全问题，如命令执行漏洞。

最近的研究表明，命令执行漏洞是当前计算机和网络系统中的一个重要安全问题。许多研究人员和公司都致力于寻找一种有效方法来防止这些漏洞的利用。例如，文献［2］提出一种目前常见的基于规则匹配的防御方式即 WAF 技术，但是其防御能力极大程度地依赖于规则的可靠性，容易发生误报或漏报等问题。文献［3］则针对 JAVA 静态分析的漏洞，对漏洞进行了分析和验证。文献

[4] 提出针对移动网络物理系统（如汽车、无人机和机器人车辆）的安全问题，开发了一种基于决策树的命令注入检测，该系统考虑了物理输入特征和网络输入特征，显著降低了假阳性率并提高了检测准确性。文献 [5] [6] 提出了基于决策树和贝叶斯算法的改进入侵检测。文献 [7] 提出了一种基于改进传统 K-means 的异常检测方法，并在 UCI 数据库上进行了实验。文献 [8] 提出了一种基于改进随机森林的算法，用于检测异常流量，但是算法复杂度较高。而文献 [9] 则提出一种基于改进随机森林和深度残差的 IoT 的入侵检测方法，但是对未知攻击的识别度不高。

基于上述研究的不足，本文首先提出了四种基于命令混淆的命令执行漏洞绕过方法，以提高对未知攻击的识别率，并降低漏报率；然后提出了一种基于随机森林算法的检测模型，通过特征提取和对算法的改进，使得运维安全管理设备的安全功能可以快速、高效地识别复杂的命令执行攻击。

1 相关知识

1.1 命令执行漏洞的危害

命令执行漏洞又称为命令注入漏洞，是一种常见的网络安全漏洞类型[10]。攻击者可以利用存在于网页中的系统命令执行方法，对服务器进行权限修改或对系统本身进行非法操作。命令执行漏洞造成的后果及危害主要有以下三点：（1）服务程序可能会去执行系统命令或者读写文件，甚至导致服务器关机或重启；（2）攻击者可能利用已知的服务器漏洞获取更高的权限，或者反弹 shell 进行进一步的攻击；（3）攻击者可以通过已经获取的服务器权限进一步控制整个网站服务器系统。

表 1 列举了在运维管理系统中，通常只有管理人员有权访问和使用的命令和部分敏感信息。恶意使用这些指令可能对服务器或远程主机造成严重的威胁，需要采取适当的措施来防止命令执行漏洞的利用，例如输入验证、过滤特殊字符、使用参数化查询等。

表 1 常被禁止的危险指令

指令	用途	示例
kill	用于杀死或中断任意进程	kill apache. exe
rm -rf	用于删除文件夹及其内容	rm -rf error. log
reboot	重启系统和服务器	reboot -d
wget	wget url -O - | sh 可以在下载恶意文件后直接运行脚本	wget http://malicious_ source -O - | sh

1.2 传统的命令执行漏洞防范措施

运维管理系统典型的应用场景如图 1 所示。

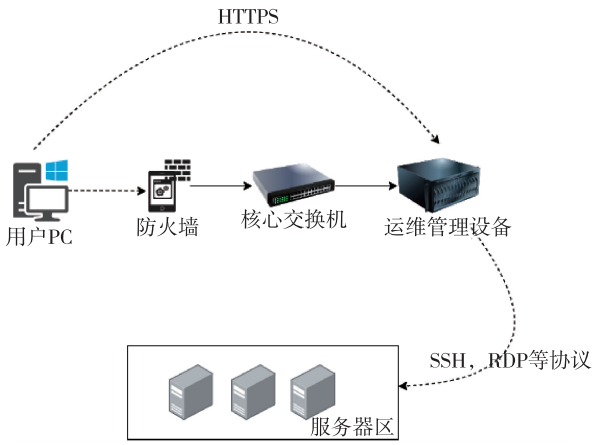

图 1 运维管理机部署拓扑图

命令执行漏洞经常会造成非常严重的危害，以本文所使用的某开源运维管理设备为例，是通过 WebSockets 协议和正则表达式进行防范恶意命令的。

运维管理系统常用的加密管理协议为 WebSockets 协议[11]。客户端和服务器端通信通常会使用 WebSockets 协议的两种方法，分别是 send（）和 close（）。通信连接建立后，用户可以使用 send（）方法从客户端向服务器发送消息，服务器端使用 onmessage（）方法监听收取消息。反之，服务端使用 send（）发送消息，客户端使用 onmessage（）方法进行监听接收，并在 web 端返回结果，这样实现了全双工的通信。当通信结束后使用 close（）方法关闭连接。

而为了对用户规定的恶意代码和命令进行过滤，采用正则表达式的方法[12]。正则表达式是一种对字符串操作的逻辑公式，即通过一些特定的字符及这些特定字符的组合，形成一个比较空余的字符串，该字符串用来表达对自身的一种过滤。

常见的命令过滤功能的执行流程是[13]：（1）用户设置“命令过滤”功能，将恶意代码、危险命令等列入黑名单。（2）鼠标键盘捕获模块检查输入的字符信息，利用正则表达式与黑名单中的命令进行匹配。（3）若匹配不成功，则通过 WebSockets 协议使用 SSH 方式向远程服务器发送命令；若匹配成功，则拒绝发送并返回警告信息，以此达到命令执行防范的效果。具体流程如图 2 所示。

1.3 基于命令混淆的绕过方法

在运维管理设备中，攻击者可以通过弱口令、未授权访问或社会工程学等安全漏洞获取普通用户权限。针对运维管理设备所管理的 Linux 系统和 Windows 系统，攻击者可以利用以下四种命令混淆方法，以规避和绕过服务器的正则表达式限制黑名单，执行恶意操作，并避免触发安全审计功能报警。

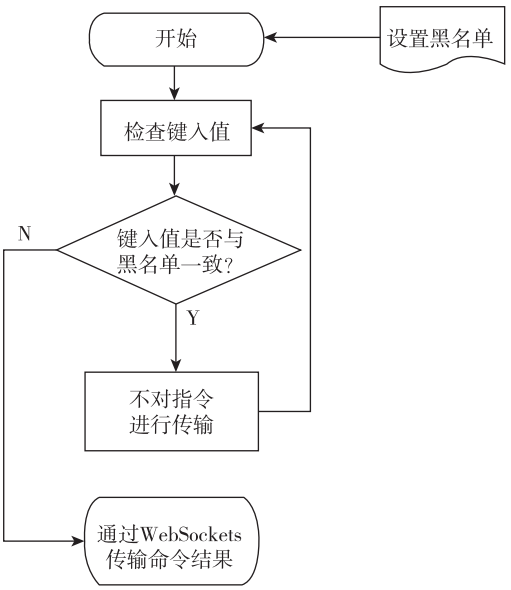

图 2　运维安全管理设备中命令过滤功能执行流程

（1）关键词拼接绕过

对命令进行关键词拼接绕过，是把多个环境变量分别定义后，将恶意命令截断并对变量赋值发送至远程服务器，再发送关键词将系统存储的环境变量按顺序拼接，例如“a = ki；b = ll；ab”。服务器会误将收到的代码拼接后作为命令执行，绕过鼠标键盘捕获模块键入值的检测。

类似地，在 Windows 系统中，可以使用“set”关键字将环境变量进行赋值，并使用“%%”符号将它们拼接在一起，还原恶意代码。例如，可以使用“set a = who&&set b = ami&&%a%%b%”的方式在命令提示符中执行“whoami”命令。

（2）关键词混淆绕过

使用关键词混淆绕过技术，是通过将黑名单中的关键词使用单引号、双引号和反斜杠等方式隔断后，操作系统会将这些符号默认为空字符，并将剩余字符进行拼接后执行攻击者构造的恶意命令。其中使用单引号、双引号进行隔断时必须确保前后闭合。

在 Windows 系统中，同样也可以使用双引号或“^”转义符进行隔断，如“who”a“mi”、“w^h^o^a^m^i”等。

（3）关键词编码绕过

关键词编码绕过是指将恶意代码进行加密或编码，然后通过操作系统认可的解密方式执行命令，从而对服务器造成恶意攻击。在 Linux 系统中，可以选择对关键词进行 base64 编码、十六进制编码等方式加密，然后使用系统中自带的“base64”或“xxd”关键词对编码后的关键词进行解码。服务器则通过“bash”命令对解码后的关键词作为命令执行。

而在 Windows 系统中，攻击者可以通过“set”命令定义一个字符串，并将关键词打乱插入任意字符后存储进字符串中。例如，攻击者需要执行“whoami”命令，可以使用“set a = whoqqqqqami&&%a：~0，4%%a：~-3%”达到目的。

（4）通配符混淆绕过

对命令进行通配符混淆绕过，是指将关键词利用通配符进行替换，绕过设备对关键词的匹配检测。在该方法中，通配符包括“*”“?”和“[ ]”三种，其中“*”可以匹配任意数量的任意字符，“?”可以在相应位置上匹配任意单个字符，“[ ]”可以匹配指定范围内的任意单个字符。

在 Windows 系统中，使用通配符无法用于执行任意命令。然而 Linux 系统中的通配符不仅可以用于文件名的匹配，还可以用于命令的执行。Linux 系统下的应用程序和命令通常存储于/bin、/sbin、/usr/bin、/usr/sbin 等应用程序目录下，攻击者可以配合多个通配符的使用，用于匹配任意目录下的命令和文件，从而绕过黑名单执行各类命令。例如，通过将命令“kill”替换为“/*in/?ill”，Linux 系统会自动匹配执行“/bin/kill”命令，即可成功绕过黑名单，而不会触发堡垒机的警告。

2　基于随机森林的命令混淆检测方法

2.1　随机森林模型相关概念

随机森林模型是一种 Bagging 类的有监督集成学习方法，是由多个决策树构成的分类器。随机森林模型通过对样本和特征进行随机选择，避免了单棵决策树的过拟合问题，具有较高的泛化性。首先，用 Bootstrap 算法对数据进行抽样，每一个决策树会从输入的 M 个变量中提取 N 个特征，其公式为：

$$N = \sqrt{M} \tag{1}$$

然后再分别以信息熵（E）和基尼系数（Gini）来衡量信息价值和特征的好坏：

$$E(D) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2}$$

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{3}$$

式中，p_i 表示第 i 件事发生的概率，n 为事件个数，D 为样本集合。

最后，模型根据结果进行投票，输出最优结果。随机森林的模型结构如图 3 所示[14]。

机器学习可以自动学习并发现训练集中的潜在模式，使精确度更高[15]。本次实验使用随机森林模型，是因为该模型在处理高维度、稀疏数据和特征相关性、非线性关系、缺失值、异常值和过拟合等方面具有很强的优

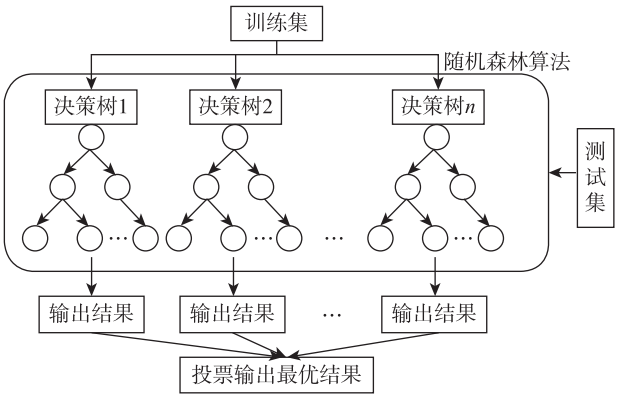

图 3　随机森林算法结构

势[16]，可以比较好地识别上一节提到的命令混淆方法中复杂或隐藏的攻击特征。同时，分别建立逻辑回归、支持向量机、朴素贝叶斯和 K 近邻 4 种模型，与随机森林模型进行对比分析。

2.2　模型设计

为验证文中提出方法的有效性，以恶意命令“kill”为例，使用了一个包含 1 200 条数据的数据集进行实验，其中 840 条用于模型训练，360 条用于验证训练模型在恶意指令上的检测能力。这些数据中包含添加了命令混淆方式的恶意指令数据集以及大量服务器中运维人员输入的常规指令，用于识别指令中是否存在命令混淆的恶意指令。

在数据预处理阶段，先使用正则表达式将数据集中可能出现的数字替换为占位符“<NUM>”，以保证数据的质量和特征的有效性。并且使用了 sklearn 库对数据进行了数值化处理，最终将数据转化为二维向量。

在特征提取阶段，用字符串长度、特殊字符和单词频率等特征来区分恶意指令和正常指令。

（1）字符串长度特征

Linux 系统中，由于攻击者可以组合本文中提到的命令混淆方法，导致字符串长度大于一般系统指令，用于添加和隐藏存在恶意指令的代码。所以在特征提取过程中，将这些命令的长度数据经过重新提取后，整理为一维数组：

$$\text{Len\_ features}_i = \text{length}(\text{commands}_i) \tag{4}$$

$$\text{Len\_ features} = \begin{bmatrix} \text{lenfeatures}_1 \\ \text{lenfeatures}_2 \\ \vdots \\ \text{lenfeatures}_n \end{bmatrix} \tag{5}$$

式中，length（）用于获取字符串长度。

（2）特殊字符特征

将文中提到的常用于命令混淆的关键词，如“’”“$”等，以及将“kill”进行 base64 或 hex 加密后的字符串，作为特殊字符提取到特征中，以提高模型的识别率。

（3）单词频率特征

由于常规指令中通常包含文件的名称、版本等无效的信息，需要将数据集中单词频率过低的指令定义为噪声，以防止出现过高的误报。公式如下：

$$\text{word\_ freq}(w_{i,j}) = \frac{\text{count}(w_{i,j})}{\sum_{k=1}^{k} \text{count}(w_{i,j})} \tag{6}$$

$$\text{word\_ noise}(w_{i,j}) = \begin{cases} 1, & \text{word\_ freq}(w_{i,j}) < 2 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

式中，word_ freq（）用于获取单词频率，count（）用于获取单词数量。

最后，将经过特征提取的样本数据分为 70% 的训练集和 30% 的测试集，对数据进行训练和测试。通过调试后，最终得到了一个具有高准确率的随机森林分类器。

2.3　实验设置与评价指标

本文实验将指令分为两类：添加了命令混淆方式的恶意指令和常规指令。对于二分类的问题，通常将测试样本分为四个部分[17-18]：

真正例（Ture Positive，TP），指正确识别添加了命令混淆代码的恶意指令的数量。

假正例（False Positive，FP），指将常规指令误识别为恶意指令的数量。

假负例（False Negative，FN），指未正确识别添加了命令混淆代码的恶意指令的数量。

真负例（Ture Negative，TN），指正确识别常规指令的数量。

在二分类问题中，一般使用准确率、精确度、召回率和 $F1$ 值进行算法评估[19]，这些指标可以反映出算法的识别性能。

$$A = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% \tag{8}$$

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \tag{9}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{10}$$

$$F1 = \frac{2\text{TP}}{P + R} \times 100\% \tag{11}$$

式中，A 为准确率，P 为精确度，R 为召回率。

2.4　仿真与对比结果

采用 2.2 节提到的训练集对基于随机森林的命令混淆检测进行训练，并使用测试集对其进行评估，计算机处理器为 Intel（R）Core（TM）i7 - 1165G7，操作系统为 Windows 11，内存为 16 GB，各类型算法使用基于 sklearn 机器学习库的算法进行设计并实现。同时分别与逻辑回归、支持向量机、朴素贝叶斯和 K 近邻 4 个模型在精确度、召回

率以及 *F*1 指标上进行比较，具体结果如表 2 所示。

表 2　仿真结果比较

算法	训练集	测试集	精确度/%	召回率/%	*F*1/%
随机森林	840	360	98. 80	96. 93	97. 90
逻辑回归	840	360	94. 12	86. 12	90. 34
支持向量机	840	360	70. 86	46. 47	56. 25
朴素贝叶斯	840	360	87. 12	86. 98	86. 11
K 近邻	840	360	78. 43	84. 29	81. 97

根据表 2 所示结果，本研究采用的随机森林模型在识别添加命令混淆的恶意代码方面表现优异，其精确度、召回率和 *F*1 值均超过 95%，其中精确度达到 98. 80%，*F*1 值达到 97. 90%。虽然逻辑回归算法的精确度和 *F*1 值都超过了 90%，但是其召回率仅为 86. 12%，显著低于随机森林模型。相比之下，其他机器学习模型在指标上的差异明显，均低于 90%，尤其是支持向量机模型的召回率仅为 46. 47%，*F*1 值仅为 56. 25%，远低于所使用的随机森林模型。因此，本文所采用的随机森林模型在用于命令混淆检测方面表现明显优于其他机器学习模型。

3　结论

针对使用命令执行漏洞的恶意代码的检测，传统的基于规则匹配的检测方法有很多局限性，本文使用一种基于随机森林模型的检测方法可以有效提高恶意代码识别率。该方法首先在数据预处理阶段，将数据进行降维和降噪处理；然后在特征提取阶段，进一步分析和识别命令混淆的特征；最后使用随机森林算法生成模型。实验结果表明，该方法能够有效地识别恶意命令中添加命令混淆方法的代码，精确度高于 95%，可以为恶意代码检测提供帮助。

后续工作将对模型继续优化，比如进一步增加命令混淆代码的数量级，并提取更多的特征用于对命令混淆的检测，从而进一步提升模型的性能。

参考文献

[1] GB/T 22239—2019 信息安全技术 网络安全等级保护基本要求 [S]. 2019.

[2] 王疆. 基于随机化的 Web 恶意代码注入防御方法研究 [D]. 郑州：战略支援部队信息工程大学，2021.

[3] 陈玮. Java 程序静态分析中的漏洞检测技术研究 [D]. 北京：北京邮电大学，2018.

[4] VUONG T P, LOUKAS G, GAN D, et al. Decision tree - based detection of denial of service and command injection attacks on robotic vehicles [C]//IEEE International Workshop on Information Forensics & Security. IEEE, 2016.

[5] LI M. Applicaation of CART decision tree combined with PCA algorithm in intrusion detection [C]//Proccedings of 2017 IEEE 8th International Conference on Software Engineering and Service. IEEE Beijing Section, 2017.

[6] CHORMALE A A, GHATULE A P. Cloud intrusion detction system using fuzzy clustering and artificial neural network [J]. Journal of Physics Conference Series, 2020. DOI: 10. 1088/1742 - 6596/1478/1/012030.

[7] 左近，陈泽茂. 基于改进 K 均值聚类的异常检测算法 [J]. 计算机科学，2016，43 (8)：258 - 261.

[8] 赵婵. 基于特征选择和改进随机森林的入侵检测技术研究 [D]. 兰州：兰州大学，2020.

[9] 潘桐. 基于改进随机森林和深度残差的 IoT 入侵检测方法 [D]. 南京：南京邮电大学，2022.

[10] 解宝琦，傅春林. 盘点常见网络攻击类型特点与区别 [J]. 网络安全和信息化，2022 (1)：124 - 127.

[11] 李孟辉. 基于 Web 的云开发平台的研究与实现 [D]. 成都：电子科技大学，2017.

[12] 王晓雨. 面向网络安全的多维正则表达式匹配算法分析 [J]. 数字通信世界，2019 (3)：262 - 263.

[13] 李继勇，周迎辉，闫岳明. 基于单向链路的无协议安全运维技术研究 [J]. 网络安全技术与应用，2022 (9)：18 - 20.

[14] 肖琦，苏开宇. 基于随机森林的僵尸网络流量检测 [J]. 微电子学与计算机，2019，36 (3)：43 - 47.

[15] YAVANOGLU O, AYDOS M, A review on cyber security datasets for machine learning algorithms [C]//IEEE International Conference on Big Data, Symposium on Data Analytics for Advanced Manufacturing. Boston, MA, USA, IEEE, 2017.

[16] 胡智杰，陈兴蜀，袁道华，等. 基于改进随机森林的 Android 广告应用静态检测方法 [J]. 信息网络安全，2023，23 (2)：85 - 95.

[17] JIN J, YAN Z W, YAN B P, et al. Botnet domain name detection based on machine learning [C]// International Conference on Wireless, Mobile and Multi-Media. Bejing, China, IET, 2016.

[18] STEVANOVIC M, PEDERSEN J M. On the use of machine learning for identifying botnet network traffic [J]. Journal of Cyber Security and Mobility, 2016, 4 (2): 1 - 32.

[19] 潘羿，李彬. 基于 DNSAE 和随机森林的电力信息网络入侵检测模型 [J]. 电力信息与通信技术，2022，20 (5)：23 - 29.

(收稿日期：2023 - 04 - 27)

作者简介：

戚臻彦 (1995 -)，男，本科，研究实习员，主要研究方向：信息安全。

孙永清 (1976 -)，男，硕士，副研究员，主要研究方向：信息安全。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部