# Web进程链及SSH操作序列异常检测技术设计文档

## 问题描述

针对Web进程链/SSH异常操作序列，检测上下文关系中的子进程异常。

Web进程链：同一Web服务应用的进程调用关系中，父进程为JAVA，其派生的所有子进程序列。

SSH操作序列：同一台主机系统中，父进程为TTY，其派生的所有子进程序列。（可以当作是管理员的输入）

异常定义：根据历史行为建立基线模型，识别有异于正常行为的异常行为

## 研究意义

如今绝大多数企业及个人用户都选择使用云计算来部署、运营业务。

Web进程链及SSH异常操作检测技术研究将应用在的主机型入侵检测系统，通过分析程序访问的内存、文件系统、日志存储等信息，结合算法能力，精确检测程序异常或恶意入侵行为，以为系统提供主动性的防御机制。

该项技术可通过捕获Linux系统下用户态和内核态通信过程中产生的信息，挖掘更深层次的特征，以进程为粒度进行入侵或异常行为的检测，能够抵御新型攻击手段，特别是隐蔽而又高危的APT攻击，为用户提供安全可靠的云计算环境。

## 已有最新技术描述

实现恶意指令检测的技术，传统一般采用的是基于模式匹配和规则的方法。此种类型的检测算法依赖于模式和规则的构造，而且对新出现的恶意行为检测能力较差。机器学习算法在分类任务与自然语言处理方面取得的成就可以应用到此问题中。

当前的应用模式主要是使用自然语言处理的技术，提取命令特征，然后使用机器学习算法与神经网络来构建模型进行命令识别。

## 拟采取的技术

特征提取：将指令当做文本，提取向量特征，随后使用算法训练模型进行测试。

机器学习中的分类算法（knn、svm、decision tree）经常用于分类任务，所以可以使用这些算法完成恶意指令的分类任务。常见的异常检测算法比如Iforest，也可用于此问题。

恶意样本的缺失导致监督学习不适合此场景，尽管如此，传统的机器学习算法也值得一试为我们后来的系统提供准确性参考。考虑使用无监督学习算法是可行的，我们也对此进行了尝试。使用聚类算法，在正常样本的基础上进行聚类分析，计算获得聚类中心，设置距离阈值，通过比较测试数据与聚类中心的距离，来决定测试样本是否恶意。

## 方案设计

使用自然语言处理的相关技术，构建命令语言字典，进行命令向量化，使用常见的异常检测算法以及机器学习算法进行测试。

## 已进行的工作

### **6.1** 使用常见的异常检测算法测试

使用Iforest做了测试，使用所有样本训练模型，去检测所有数据集中的恶意指令，检测出的恶意指令占比较低，效果不理想。可以选择出部分恶意命令，但是召回率较低。

### 6.2自然语言处理

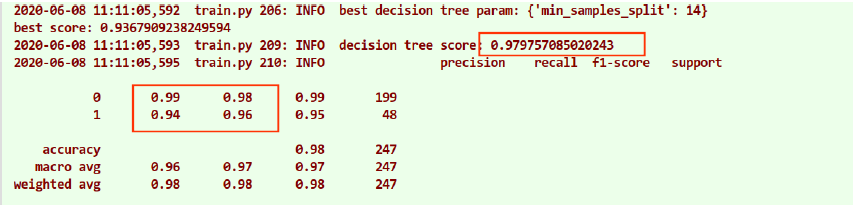
使用基于nlp的机器学习技术，以情感分析为原型，进行恶意命令识别。  
将命令看做shell与计算机交互的语言，对该语言进行分析，判断文本极性，随后可用各种机器学习算法进行测试，比如knn、决策树、svm等。Knn表现不好，决策树在训练数据集的表现良好，但是拿所有的数据进行测试的时候，准确率下降的非常快。分类效果一般。在训练数据集上的准确率可达95%，但是使用更多数据测试时只有60%。

#### 特征提取

使用sklearn的TfidfVectorizer或CountVectorizer进行command的向量化，将向量化后的结果，取1000条正常指令，和所有的异常指令训练模型。

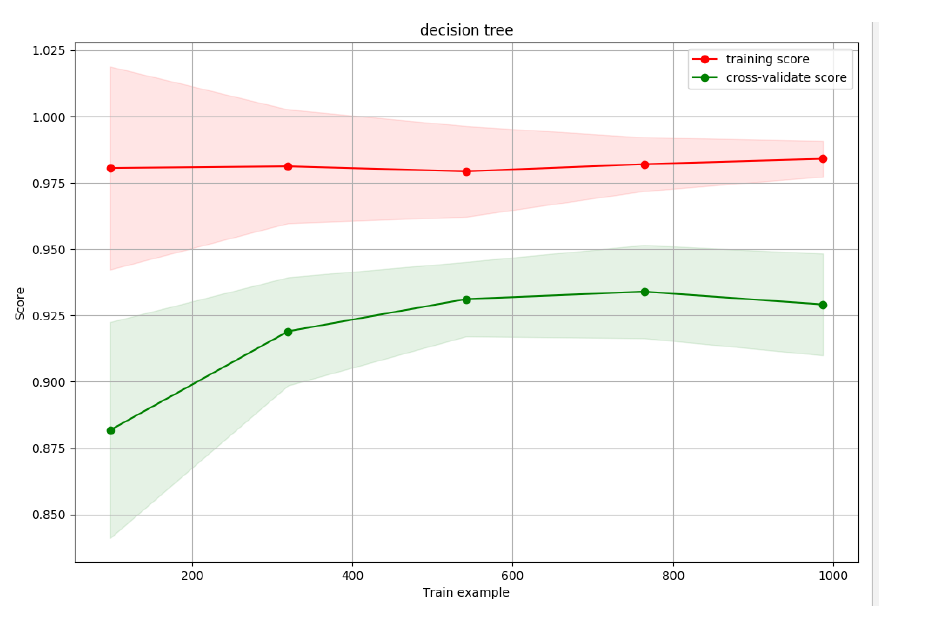
#### 决策树测试结果

准确性分析如下



1决策树模型准确率

在测试集上获得了较高准确率，高达98%，模型学习曲线如下



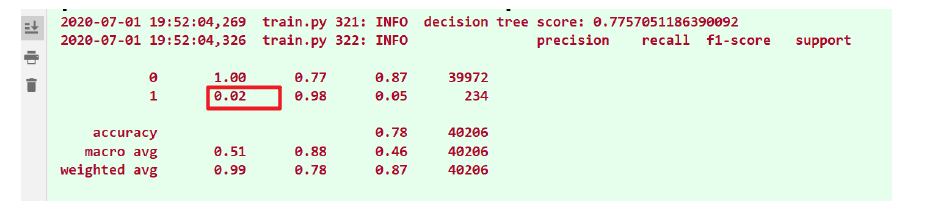
2决策树模型学习曲线

在交叉验证数据集上的准确性已经出现了下降的趋势，模型已收敛，并且稍微有些过拟合。但是准确性很高，目前表现还可以，kfold交叉验证结果如下



3 kfold5折交叉验证准确率

后来使用所有数据对该模型进行测试

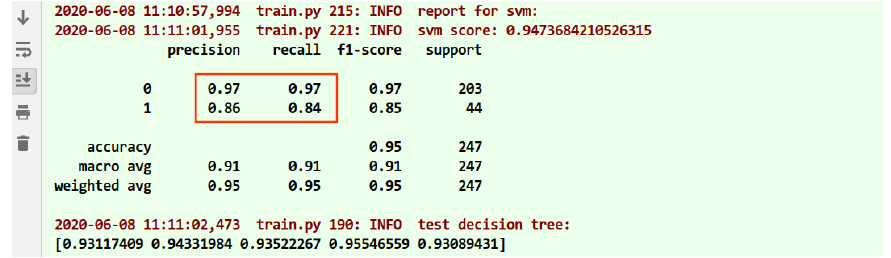


4 决策树模型在所有数据集上的准确率

模型表现在恶意命令识别准确率上非常低。

#### SVM测试结果

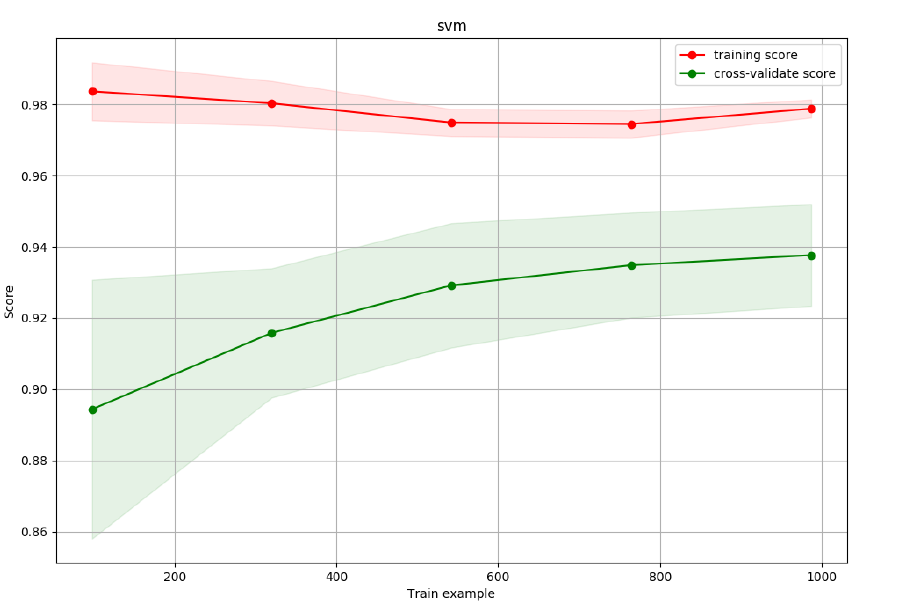
使用支持向量机模型，经过参数选择，最优参数为kernel=rbf，gamma=0.22449同样的步骤，模型准确率如下



5 SVM模型准确率

可以看到svm模型在恶意命令准确性不如决策树模型高

学习曲线如下



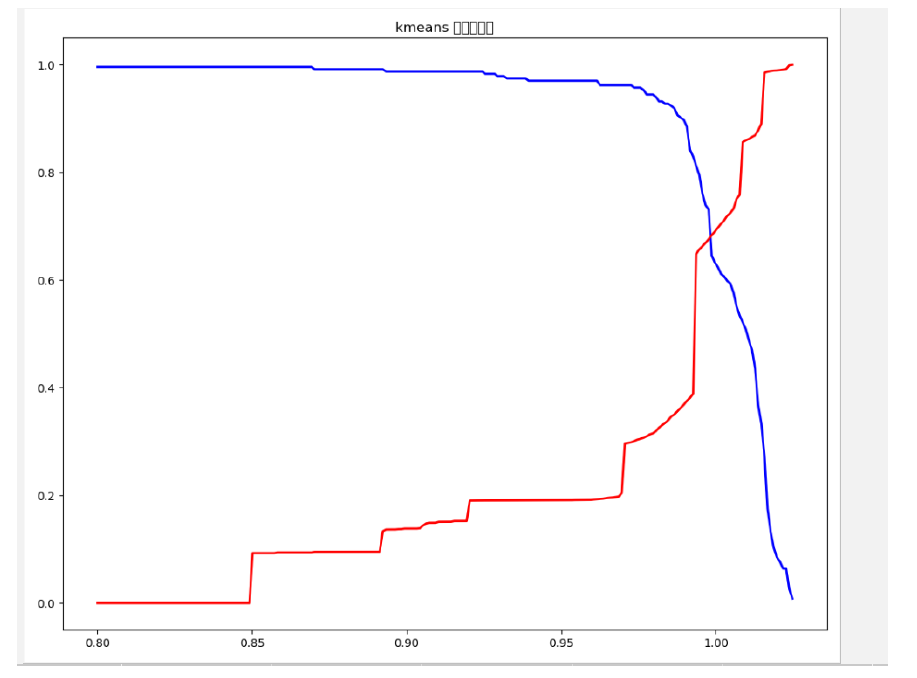
6 SVM模型学习曲线

模型未收敛，准确性在上升，但是其上限也比较明显。效果整体不如决策树算法。

### 6.3 无监督学习算法

使用聚类算法，仅仅针对正常样本建模  
由于恶意数据量少，仅使用正常数据建模，把恶意数据当做异常点，也是可以尝试的方法。我们用kmeans做了测试，使用正常命令样本做聚类，随后通过某条命令和聚类中心的距离，判断该命令是否是恶意指令。设置合适的阈值，可以让正常指令和异常指令的准确率达到80%。但是还有改进的空间。

结果示意图如下



7 Kmeans 聚类算法准确性示意图

横轴是采用的阈值，红色曲线是正常数据准确性，蓝色曲线是异常数据准确性，随着阈值升高，正常指令准确性逐渐提高，异常指令准确性逐渐下降，两者相遇在0.7左右，效果不理想。其主要原因在与计算命令距离的算法，后面拟进行对距离算法的选择优化，进一步提高这种方法的准确性。

## **未来拟采取的方案**

项目的难点主要在特征提取方面，如果可以综合考虑更多的命令执行特征，机器学习算法就可以得到大幅优化。

基于n-gram的knn，是Detection of Malicious Remote Shell Sessions中提出的一种恶意命令识别的方法，n-gram计算两条指令的距离，使用knn做最终分类，并且考虑了命令上下文。下一步的目标是复现文中的异常检测系统，测试系统性能。

## 8. 参考文献

[1]. Dumont, P., Meier, R., Gugelmann, D., & Lenders, V. (2019, May). Detection of Malicious Remote Shell Sessions. In *2019 11th International Conference on Cyber Conflict (CyCon)* (Vol. 900, pp. 1-20). IEEE.

[2]. Tian, Y., Wang, J., Zhou, Z., & Zhou, S. (2017, December). CNN-webshell: malicious web shell detection with convolutional neural network. In *Proceedings of the 2017 VI International Conference on Network, Communication and Computing* (pp. 75-79).

[3]. Hendler, D., Kels, S., & Rubin, A. (2018, May). Detecting malicious PowerShell commands using deep neural networks. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 187-197).