计算机科学
COMPUTER SCIENCE

一种基于顺序和频率模式的系统调用轨迹异常检测框架

魏辉，陈泽茂，张立强

引用本文

魏辉，陈泽茂，张立强. 一种基于顺序和频率模式的系统调用轨迹异常检测框架[J]. 计算机科学, 2022, 49(6): 350-355.

WEI Hui, CHEN Ze-mao, ZHANG Li-qiang. Anomaly Detection Framework of System Call Trace Based on Sequence and Frequency Patterns[J]. Computer Science, 2022, 49(6): 350-355.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210500031

一种基于顺序和频率模式的系统调用轨迹异常检测框架

魏 辉 陈泽茂 张立强
空天信息安全与可信计算教育部重点实验室(武汉大学国家网络安全学院) 武汉 430072
(weihui@whu.edu.cn)

摘 要 针对现有的基于系统调用的异常入侵检测方法使用单一轨迹模式无法准确反映进程行为的问题,基于系统调用轨迹的顺序和频率模式对进程行为进行建模,设计了一个数据驱动的异常检测框架。该框架可以同时检测系统调用轨迹的顺序异常和定量异常,借助组合窗口机制,通过满足离线训练和线上检测对提取轨迹信息的不同需求,可以实现离线细粒度学习和线上异常实时检测。在ADFA-LD入侵检测标准数据集上进行了针对未知异常检测性能的对比实验,结果表明,相比4类传统机器学习方法和4类深度学习方法,该框架的综合检测性能提高了10%左右。

关键词:基于主机型入侵检测系统;系统调用;深层神经网络;长短期记忆神经网络
中图法分类号 TP393

Anomaly Detection Framework of System Call Trace Based on Sequence and Frequency Patterns

WEI Hui,CHEN Ze-mao and ZHANG Li-qiang
Key Laboratory of Aerospace Information Security and Trusted Computing,Ministry of Education,School of Cyber Science and Engineering,Wuhan University,Wuhan 430072,China

Abstract The existing system call-based anomaly intrusion detection methods can't accurately describe the behavior of the process by a single trace pattern. In this paper,the process behavior is modeled based on the sequence and frequency patterns of system call trace,and a data-driven anomaly detection framework is designed. The framework could detect both sequential and quantitative anomalies of the system call trace simultaneously. With the help of combinational window mechanism,the framework could realize offline fine-grained learning and online anomaly real-time detection by meeting different requirements of offline training and online detection for extracting trace information. Performance comparison experiments of unknown anomalies detection are conducted on the ADFA-LD intrusion detection standard dataset. The results show that,compared with the four traditional machine learning methods and four deep learning methods,the comprehensive detection performance of the framework improves by about 10%.

Keywords Host-based intrusion detection systems,System calls,Deep neural network,Long and short-term memory neural network

1 引言

目前,随着个人主机的广泛使用,主机入侵事件时有发生且增长趋势明显,给个人隐私安全带来了严峻挑战。基于此,入侵检测系统成为了保障计算机安全的关键组件之一[1]。与基于网络的入侵检测系统(NIDS)相比,基于主机的入侵检测系统(HIDS)具有检测粒度细且能够检测内部攻击的优势[2]。入侵检测方法分为基于签名的检测方法和基于异常的检测方法。基于签名的检测方法指根据已知的入侵行为模式来构建攻击轮廓库,在检测时使用系统实时行为来匹配已知攻击模式,从而判断是否出现了入侵活动。该方法的缺点是漏报率较高,而且无法检测未知攻击。基于异常的检测方法指构建系统正常行为模型,当检测到系统行为偏离正常行为轨迹时,即判定为出现了异常活动。该方法的优点是能够检测未知异常,更适用于当下的检测需求。本文提出的检测框架属于后者。

系统调用是进程与操作系统的原始交互,使得系统调用轨迹可以用于检测云环境[3]和容器环境[4]下的进程异常活动。系统调用轨迹异常可以分为两类:顺序异常和定量异常。程序根据逻辑流顺序执行,如果系统调用轨迹偏离程序流的正常模式,则会发生顺序异常。程序中的循环、跳转以及用户需完成任务的统计学特性决定了程序执行具有一些恒定的线性关系,可以通过系统调用轨迹中的定量关系来捕获它们。如果一组系统调用发生定量异常,则会破坏这些线性关系。

到稿日期:2021-05-07 返修日期:2021-07-30
基金项目:湖北省重点研发项目(2020BAA001)
This work was supported by the Key R&D Projects of Hubei Province(2020BAA001).
通信作者:陈泽茂(chenzemao@whu.edu.cn)

现有的基于系统调用的异常检测方法大致可以分为两类：基于传统机器学习方法和基于深度学习方法。传统机器学习方法是在得到完整系统调用轨迹后，基于分类思想来判断是否出现了异常活动[2]，但这种事后分析异常的方法已经无法满足当下的异常实时检测需求[5]。深度学习方法利用滑动窗口机制，并依靠其神经网络强大的学习能力来挖掘局部轨迹片段的细粒度特征，从而做到异常实时检测[6]。上述两类方法虽然在各自的实验环境下都取得了不错的实验结果，但都只围绕系统调用轨迹的单一特征维度进行详细的研究，导致对进程行为建模不完整，在面对未知异常场景时检测效果可能会变差。

针对上述情况，本文提出了一个数据驱动的异常检测框架(SysAnomaly)。该框架的核心思路是依靠长短期记忆神经网络(Long Short-Term Memory，LSTM)在学习时间序列数据的时序性和长程依赖性上具有的优势，使用系统调用轨迹的顺序和频率模式建模进程行为以检测顺序异常和定量异常。由于建模进程行为时使用的特征信息更全面，因此，相比现有方法，该框架在未知异常的检测上综合检测性能有明显提高。本文的主要贡献如下：

(1)针对使用系统调用轨迹单一模式难以完整反映进程行为的问题，提出了使用其顺序和频率模式来准确建模进程行为。

(2)设计了一个基于 LSTM 神经网络的异常检测框架 SysAnomaly，该框架能自动同时检测系统调用轨迹的顺序异常和定量异常。

(3)提出了组合窗口机制，能满足离线训练和线上检测对提取轨迹信息的不同需求，实现了离线细粒度学习和线上异常实时检测。

2　相关工作

Forrest 等[7]首次提出了使用系统调用进行主机系统异常入侵检测。其方法是使用固定长度的正常系统调用序列来标识彼此之间的相对位置，并将它们存储在数据库中。在运行程序时，将其轨迹与数据库中的已知序列进行比较，若出现与数据库数据的累积偏差超过阈值的序列，则表示可能存在异常行为。但该方法创建程序正常行为数据库非常耗时，而且不准确的偏差阈值会导致较高的误报率。

许多研究都开始使用系统调用来开展主机系统异常入侵检测。基于分类思想的传统机器学习方法，如 k 最近邻(kNN)[8]、决策树(Decision Tree)[9]、支持向量机(SVM)[10]和朴素贝叶斯(NaiveBayes)[11]已经在基于系统调用的主机入侵检测系统上得以实现。虽然这些传统机器学习方法都取得了较好的检测结果，但它们都是在得到完整的系统调用轨迹后，基于分类思想来分析是否出现异常，再采取防御措施。然而，目前网络攻击复杂多变，这类事后分析问题的方法会因攻击得不到及时处理而造成严重损失。因此，这类方法已难以满足当下异常实时检测的需求。

近年来，基于系统调用的异常入侵检测研究从传统机器学习逐渐转向了深度学习。深度学习依靠其深层神经网络强大的学习能力，可以发现局部系统调用轨迹的细粒度特征，从而能够应用于异常实时检测[1]。Creech 等[12]提出基于上下文语法来创建基于系统调用轨迹的语义模型，但是该创建过程非常耗时。Xie 等[13]发现使用系统调用轨迹的频率特征构建检测模型比创建基于语义的检测模型更快，但是其误报率较高。Sanjeev 等[14]提出一种基于语义的频率集中模型(Frequency-Centralized Model，FCM)，该模型可以对带参数的系统调用进行分类以检测恶意活动。但该模型处理参数的流程十分复杂，难以应用于异常实时检测。Lv 等[15]提出了使用序列到序列模型进行系统调用轨迹的预测，以达到检测异常的目的。但该模型的预测范围太大，导致面对未知异常时检测率较低。Kolosnjaii 等[16]发现，使用一个基于 CNN 和 RNN 的神经网络对系统调用序列建模可以实现较好的分类效果，但该方法的缺点是检测过程耗时较长。随后，Chawl 等[6]在此思路上提出 CNN/RNN 模型，使用 GRU 来缩短训练时间以达到高效检测的目的，该方法提高了检测速度却损失了一定的检测精度。考虑到系统调用数据量足够大，Sun 等[5]提出使用卷积神经网络结合长短期记忆神经网络来捕获系统调用轨迹的顺序特征，以检测异常，但该方法只考虑了系统调用的一个特征维度，存在检测率较低的问题。Zhan 等[17]提出基于连续时间窗口对主机复合资源进行采样，采用主成分分析方法(PCA)将高维数据映射到低维空间，然后在特征空间的不同维度上查看每个数据点与其他数据的偏差以实现异常检测，但该方法在未知异常场景中会出现检测率骤降的情况。Xu 等[18]发现，使用固定大小的滑动窗口算法扫描完整的系统调用跟踪可以发现异常活动，但是其工作没有考虑线下训练和线上检测对数据处理方式的不同需求。

3　SysAnomaly 设计

3.1　设计概观

SysAnomaly 的目标是以一种自反馈的监督学习方式，实现实时、自动、准确地检测系统调用轨迹的顺序异常和定量异常。其结构如图 1 所示。

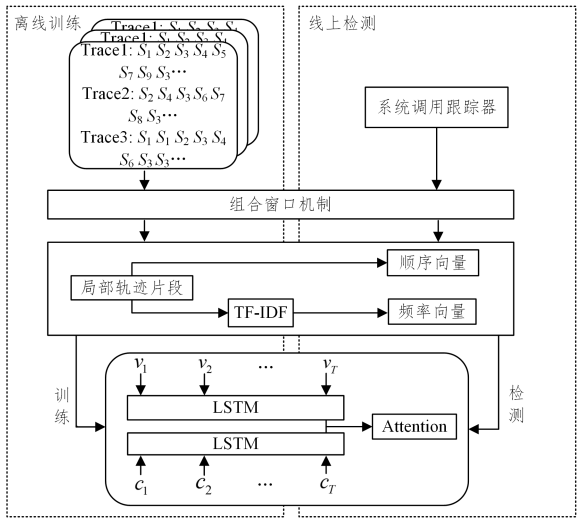

图 1　SysAnomaly 结构图

Fig. 1　Structure diagram of SysAnomaly

该异常检测框架存在两种状态：离线训练状态和线上检测状态。在离线训练状态下，SysAnomaly 首先使用传统滑动

窗口从历史系统调用轨迹中提取局部轨迹片段，然后使用LSTM神经网络学习该局部系统调用轨迹片段的顺序和频率模式。离线训练定期进行，例如，每周一次，以便将新出现的异常活动特征定期合并到新学习的离线模型中。在线上检测状态下，由于系统调用是实时跟踪的，在使用跳步滑动窗口实时获取最新局部轨迹片段后，SysAnomaly使用离线状态下训练得到的检测模型来确定实时局部轨迹是否出现异常活动。线上检测出现误报或漏报情况时，该框架会记录人工反馈的正确结果，然后在离线状态下定期训练时，自动输入正确的数据样本，以便及时修正参数。

3.2 模式向量化

系统调用轨迹的顺序模式和频率模式，并将其反映在其顺序信息和定量信息中。模式向量化的作用是将顺序信息和定量信息映射为固定维度的顺序向量和频率向量。其工作流程如图2所示，其中包括使用滑动窗口来获取轨迹片段和使用TF-IDF加权聚合来区分局部活动特征。

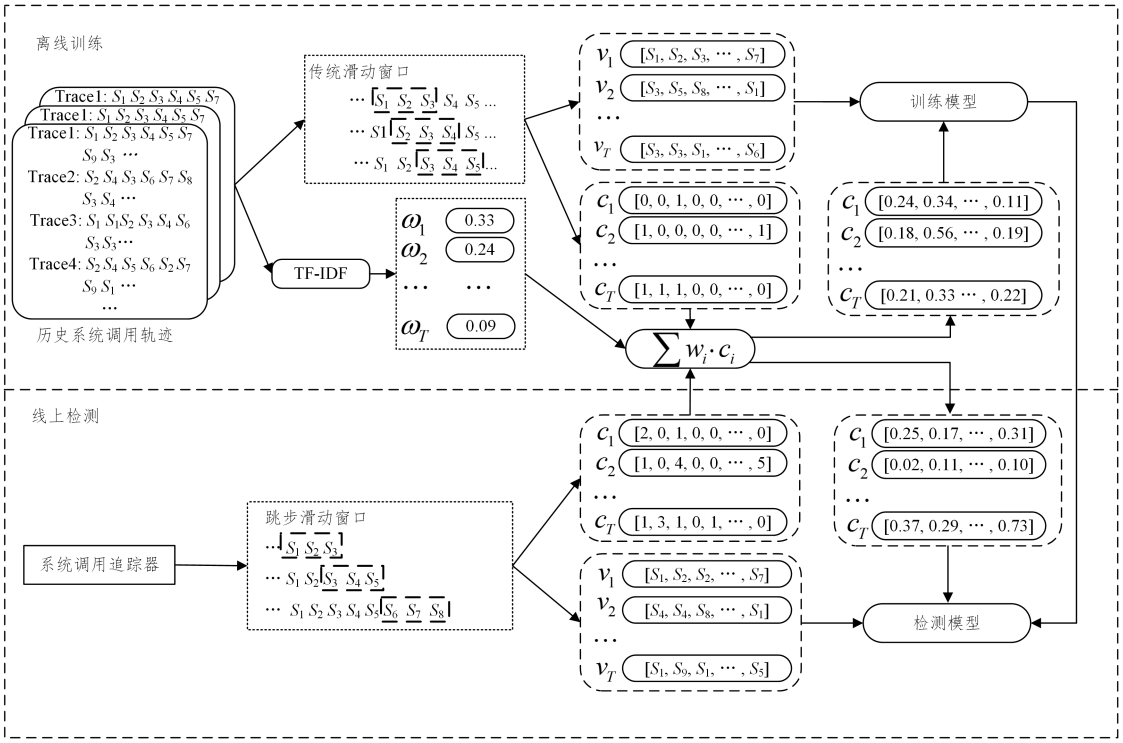

图2 模式向量化

Fig. 2 Pattern vectorization

Forrest等[7]首先提出采用滑动窗口的方法处理系统调用轨迹数据。Xu等[18]验证了使用滑动窗口方法处理数据的可行性。通过观察攻击的特点，Wunderlich等[19]发现，在被标记为异常的轨迹中，并不是每一部分序列都表现出恶意行为。通常，异常活动不应是在结束后才被检测到，因此使用滑动窗口提取轨迹片段来检测局部异常活动成为了常用方法。

然而，经过实验观察，采用移动距离固定的滑动窗口(传统滑动窗口)会产生线上系统调用实时跟踪和异常实时检测不同步的问题。也就是说，移动距离太小，异常实时检测会落后于实时跟踪；移动距离太大，训练模型的特征粒度会过大，从而导致难以学习隐藏的异常特征。为此，在综合分析系统调用轨迹特性和异常实时检测需求后，本文在传统滑动窗口的基础上提出了组合窗口机制，即线上检测使用移动距离变化的滑动窗口(跳步滑动窗口)，而离线训练仍采用传统滑动窗口。如图2所示，离线训练采用传统滑动窗口在历史系统调用轨迹上每次滑动1个系统调用，以学习轨迹片段的细粒度特征。线上检测使用跳步滑动窗口实时跟踪最新系统调用，以检测轨迹片段是否存在异常活动。相比文献[18]的方法，本文方法考虑了训练和检测状态下对提取轨迹片段的不同需求，更具实用性。

在使用滑动窗口获取到局部轨迹片段后，SysAnomaly会将轨迹片段的顺序信息和定量信息转化为固定维度的顺序向量和频率向量。令$\Omega=\{S_1, S_2, \cdots, S_n\}$是不同系统调用的整个集合。用于检测的滑动窗口大小为T，则轨迹片段的顺序向量$\boldsymbol{V}=\{S_i\}$，$S_i\in\Omega$，$\boldsymbol{V}\in\Omega^{\mathrm{T}}$。在生成频率向量时，SysAnomaly使用TF-IDF进行加权聚合。TF-IDF权重能够有效地度量系统调用轨迹中各个系统调用在不同轨迹片段中的重要性，以区分局部轨迹活动。例如，如果一个系统调用频繁出现在系统调用轨迹片段中，则表示此系统调用更能代表此局部行为；如果该系统调用频繁出现在所有轨迹片段中，则表示其太常见从而无法区分这些局部行为，因此应减小其权重。计算每个系统调用的权重的表达式如式(1)所示：

$$\omega_{\mathrm{syscall}}=\frac{\#\,syscall/\#\,total}{\#\,L/\#\,L_{\mathrm{syscall}}} \tag{1}$$

其中，$\#\,syscall$表示轨迹片段中具体的系统调用个数，$\#\,total$表示整个轨迹片段的系统调用个数，$\#\,L$表示所有轨迹片段的个数，$\#\,L_{\mathrm{syscall}}$表示包含此系统调用的轨迹片段的个数。

最终，根据式(2)得到对应轨迹片段的频率向量$\boldsymbol{C}\in R^{\mathrm{T}}$。

$$\boldsymbol{C}=\frac{1}{N}\sum_{i=1}^{N}\omega_i\cdot v_i \tag{2}$$

使用组合窗口机制和TF-IDF加权聚合，既能满足离线

训练和线上检测获取轨迹片段的不同需求，又能准确地表示各个局部活动的不同特征。因此，SysAnomaly 既能保证深层神经网络学习到隐藏的细粒度特征，又能保证线上异常检测的时效性。

3.3　异常检测模型

通过模式向量化后，每个局部轨迹片段 L 被转化为顺序向量 $\boldsymbol{V}$ 和频率向量 $\boldsymbol{C}$。以顺序向量 $\boldsymbol{V}$ 和频率向量 $\boldsymbol{C}$ 为输入，SysAnomaly 采用基于注意力机制的 LSTM 神经网络来学习局部轨迹片段的隐藏特征。LSTM[20] 是循环神经网络（Recurrent Neural Network，RNN）的一种变体，也是专门为学习具有长依赖关系的时间序列数据而设计的。相比 LSTM，门控循环单元（Gate Recurrent Unit，GRU）只具有 update 和 reset 两个门，参数更少且更易收敛，因此更适合小样本集。本文使用的标准数据集的数据量十分充裕，因此 LSTM 更适合于本文方法的应用场景。如图 3 所示，$\boldsymbol{h}_t^S$ 和 $\boldsymbol{h}_t^C$ 分别是时间步长 t 处的隐藏向量，其中，时间步长 t 表示输入系统调用轨迹的位置，然后将这两者连接得到隐藏状态 $\boldsymbol{h}_t$。

$$\boldsymbol{h}_t = concat(\boldsymbol{h}_t^S + \boldsymbol{h}_t^C) \tag{3}$$

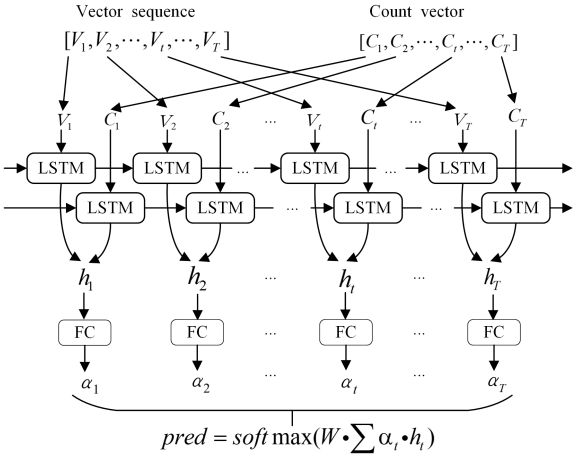

图 3　基于注意力机制的长短期记忆神经网络异常检测模型

Fig. 3　Attention-based long short-term memory neural network anomaly detection model

由于不同的系统调用对分类结果的影响不同，本文在 LSTM 模型中引入了注意力机制，对不同的系统调用赋予不同的权重。通过这种方式，对一些系统调用噪声给予更低的关注，可以减小其对整个异常检测过程的影响。各个系统调用的重要性可以从注意层自动学习。更具体地说，本文添加了一个完全连接层（即图 3 的 FC 层）作为连接隐藏状态 $\boldsymbol{h}_t$ 的注意层，其输出是注意的权重（表示为 α），它反映了系统调用的重要性。α 越大，模型越关注这个系统调用。α 的计算式如式(4)所示，其中 $\boldsymbol{W}_t^a$ 是在时间步长 t 处注意层的权重。

$$\alpha_t = \tanh(\boldsymbol{W}_t^a \cdot h_t) \tag{4}$$

最后，对所有的 α 的隐藏状态求和，然后构造一个 softmax 层来输出分类结果。如式(5)所示，W 是 softmax 层的权重，T 是滑动窗口内系统调用的轨迹长度。

$$pred = \mathrm{softmax}(W \cdot (\sum_{t=0}^{t=T} \alpha_t \cdot h_t)) \tag{5}$$

在训练阶段，使用预测输出和数据集的真实标签来计算交叉熵，并将其作为损失函数，使用随机梯度下降算法来训练模型的参数。

4　实验及分析

4.1　实验数据集及评价指标

为了与其他方法进行对比，本文使用主机入侵检测领域标准数据集 ADFA-LD[21]，产生该数据集的操作系统环境是 x86 架构上的 Ubuntu 11.04，与现代主流主机环境相符。其包含 3 个部分，即正常训练数据集（Train Data）、正常验证数据集（Validation Data）和攻击数据集（Attack Data），详细信息如表 1 所列。为了保证实验的一致性和科学性，将各个文件的系统调用轨迹分别导入 Excel 软件中，进行去重处理后得到实验数据。为了对比各类方法在面对未知异常时的检测性能，对攻击数据集进行了分配，即一部分攻击数据集用于训练，另一部分未知攻击数据集用于检测。

表 1　ADFA-LD 数据集

Table 1　ADFA-LD dataset

类型	轨迹数	去重后轨迹数
Train Data	833	780
Validation Data	4372	2420
Hydra_FTP	162	161
Hydra_SSH	176	174
Adduser	91	90
Java_Meterpreter	124	122
Meterpreter	75	73
Web_Shell	118	118

为了衡量 SysAnomaly 在异常检测中的有效性，本文使用精确率（Precision）、召回率（Recall）和 F1 分数（F1-Score）作为指标来评估模型。计算各项指标的公式如下。

精确率：指检测为异常的样本中真实异常样本所占比例。

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

召回率：指异常样本被正确识别出来的百分比，该值等于检测率（DR）。

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

F1 分数：是精确率和召回率的调和平均值，反映模型的综合性能。

$$F1\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

式(6)、式(7)中，TP 是检测正确的异常样本数，FP 是检测错误的正常样本数，FN 是检测错误的异常样本数，TN 是检测正确的正常样本数。

4.2　基准方法

本文比较了 SysAnomaly 和其他几种基准方法，即 4 类传统机器学习方法和 4 类深度学习方法。k 最近邻（kNN）[8] 是一种基于分类思想进行异常检测的方法，其主要思路是针对检测样本寻找最邻近的 k 个数据样本的类别以判断该检测样本的类别。决策树（Decision Tree）[9] 是一个使用系统调用频率及其标签来构建的树结构图，通过遍历到达叶子节点来反映其预测状态以进行异常检测。支持向量机（SVM）[10] 使用多维空间中的超平面来分离样本以达到异常检测效果。朴

素贝叶斯(NaiveBayes)[11]也是一种基于分类思想的异常检测方法,其思路是根据某个对象的先验概率计算出其后验概率,后验概率最大的类为该对象所属的类。LSTM(one-hot)[21]将各个系统调用表示为340维的one-hot向量后按序输入到LSTM神经网络。LSTM(semantics)[19]使用嵌入层将系统调用表示为8维语义向量后按序输入到LSTM神经网络中。CNN/RNN[6]首先使用一维卷积神经网络(1D-CNN)提取系统调用轨迹的局部特征,然后输入门控循环单元(GRU)以加快训练和检测。Seq2Seq(Sequence-to-Sequence)[16]使用GRU作为神经单元来构建序列到序列的预测模型,以预测系统调用轨迹,然后比较预测结果与实际情况来判断是否出现异常活动。

4.3 实验设置

本文实验都是在Intel Xeon 2.40 GHz CPU和64 GB内存的Linux服务器上进行的。SysAnomaly使用Python 3.7和Pytorch 1.5实现,并在Nvidia Tesla V100 16 GB上训练。其双层LSTM神经网络的隐藏层神经元个数均设置为32。使用*mini-batch*的方式对网络进行小批量梯度更新,设置*mini-batch*为1024。为了避免模型过拟合,经实验调试,在最后的全连接层设置*dropout*为0.8时检测效果最佳。滑动窗口过大会无法检测细粒度特征,过小则训练和检测都会十分耗时,参考文献[21]和经过实验调试对比后发现,设置滑动窗口大小为20时SysAnomaly的检测性能最佳。本文使用Adam优化算法对模型进行优化求解。

4.4 实验结果与分析

为了对比各方法在面对未知异常时的检测性能,实验使用一部分的异常样本作为训练集让模型学习异常特征,使用另一部分不同异常样本作为测试集。具体来说,使用Hydra_SSH,Adduser和Meterpreter 3种攻击类型的337条异常轨迹(去重后有331条轨迹)混合相应比例的正常轨迹作为训练集;使用Hydra_FTP,Java_Meterpreter和Web_Shell 3种攻击类型的401条异常轨迹混合相应比例的正常轨迹作为测试集。这种实验数据设置方案可以模拟真实未知异常场景,有效对比各方法面对未知异常时的检测性能。

4类传统机器学习方法和SysAnomaly的对比检测结果如表2所列。总体来说,SysAnomaly取得了最好的综合性能,即其F1-Score最高。与4类传统机器学习方法相比,SysAnomaly的召回率平均高出了10%左右。也就是说,面对未知异常的检测,SysAnomaly的检测率比传统机器学习方法提高了10%。虽然KNN,SVM和Decision Trees的精确率比SysAnomaly高,但这是它们在牺牲检测未知异常的条件下得到的。也就是说,面对某种未知异常,这3类方法更倾向于判断其为正常,这会直接导致模型异常感知能力下降而无法应对未知异常的实时检测。

表2 与传统机器学习方法的对比

Table 2 Comparison with traditional machine learning

Methods	Precision	Recall	F1-Score
Decision Tree	0.88	0.66	0.76
kNN	**0.91**	0.67	0.77
SVM	0.86	0.74	0.79
NaiveBayes	0.64	0.78	0.70
SysAnomaly	0.83	**0.86**	**0.85**

4类深度学习方法和SysAnomaly的对比检测结果如表3所列。在4种方法对比中,LSTM(semantics),CNN/RNN,Seq2Seq的精确率比SysAnomaly平均高3%。但是,SysAnomaly的检测率却比LSTM的检测率高26%,比CNN/RNN高9%,比Seq2Seq高13%。对于异常检测领域,高检测率比高精确率更重要,错过一次异常活动的应对处理可能会产生非常严重的后果。同时,实验结果也说明了面对未知异常时,SysAnomaly的异常感知能力要优于4类深度学习模型。从整体上看,SysAnomaly的综合检测性能相比其他4类方法提高了10%左右。

表3 与深度学习方法的对比

Table 3 Comparison with deep learning

Methods	Precision	Recall	F1-Score
LSTM(one-hot)	0.77	0.62	0.69
LSTM(semantics)	**0.87**	0.60	0.71
CNN/RNN	0.85	0.76	0.80
Seq2Seq	0.86	0.73	0.79
SysAnomaly	0.83	**0.86**	**0.85**

为了验证顺序和频率模式比单一模式建模进程行为更准确,本文对比测试了没有顺序向量的SysAnomaly(SysAnomaly w/o Vector Sequence)和没有频率向量的SysAnomaly(SysAnomaly w/o Count Vector),它们分别表示没有使用顺序模式和没有使用频率模式的SysAnomaly。表4所列的实验结果表明,使用顺序和频率模式的SysAnomaly比只使用顺序模式的SysAnomaly的综合检测性能提高了8%,比只使用频率模式的SysAnomaly的综合检测性能提高了27%。从整体来看,使用顺序和频率模式的SysAnomaly在各项对比指标上均表现更佳,这也充分说明了使用更全面的轨迹信息作为检测依据更有利于发现异常。

表4 与使用单一模式的SysAnomaly对比

Table 4 Comparison with SysAnomaly with single mode

Methods	Precision	Recall	F1-Score
SysAnomaly w/o vector sequence	0.71	0.84	0.77
SysAnomaly w/o count vector	0.65	0.52	0.58
SysAnomaly	**0.83**	**0.86**	**0.85**

结束语 本文提出了一个数据驱动的异常检测框架(SysAnomaly),它使用系统调用轨迹的顺序和频率模式对进程行为进行准确建模,以提高异常检测率。实验结果表明,相比于使用轨迹单一模式,使用顺序和频率模式的SysAnomaly在各项评价指标上均表现更佳。在未知异常检测方面,与4类传统机器学习方法和4类深度学习方法相比,SysAnomaly在检测率和综合性能两个指标上均有更好的表现。

本文实验没有对比测试不同参数设置情况下的检测性能。在未来的研究中,将使用更丰富的标准数据集和更完善的对比实验来充分验证SysAnomaly的检测性能。同时,将探索在异常检测系统中引入分布式架构来缩短检测响应时间。

参考文献

[1] MORA-GIMENOF J, MORA-MORA H. Intrusion Detection

System Based on Integrated System Calls Graph and Neural Networks[J]. IEEE Access, 2021(9): 9822-9833.

[2] LIU M, XUE Z, XU X, et al. Host-Based Intrusion Detection System with System Calls: Review and Future Trends[J]. ACM Computing Surveys, 2018, 51(5): 98-136.

[3] CHEN X S, CHEN J X, JIN X, et al. Process Abnormal Detection Based on System Call Vector Space in Cloud Computing Environments[J]. Journal of Computer Research and Development, 2019, 56(12): 2684-2693.

[4] CHEN X S, JIN Y L, WANG Y L, et al. Anomaly Detection of Processes Behavior in Container Based on LSTM Neural Network[J]. Acta Electronica Sinica, 2021, 49(1): 149-156.

[5] SUN P, LIU P, LI Q, et al. DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System [J]. Security and Communication Networks, 2020, 5(55): 639-652.

[6] CHAWL A, LEE B, FALLON S, et al. Host Based Intrusion Detection System with Combined CNN/RNN Model[C]// ECML PKDD 2018 Workshops. Lecture Notes in Computer Science. Cham: Springer, 2019: 149-158.

[7] FORREST S, HOFMEVRS A, SOMAYAJI A, et al. A sense of self for Unix processes[C]// Proceedings of IEEE Symposium on Security and Privacy. Oakland: IEEE press, 1996: 120-128.

[8] DING Y X, YUAN X B, ZHOU D, et al. Feature representation and selection in malicious code detection methods based on static system calls[J]. Computers & Security, 2011, 30(6): 514-524.

[9] JOHNSON R, TONG Z. Learning Nonlinear Functions Using Regularized Greedy Forest [J]. IEEE Transactions Pattern Analysis and Machine Intelligence, 2014, 36(5): 942-954.

[10] WEAL K, SYED S M, ABEDL H L, et al. Combining heterogeneous anomaly detectors for improved software security[J]. Journal of Systems and Software, 2017, 137(MAR.): 415-429.

[11] DARREN M, FREDRIK V, GIOVANNI K, et al. Anomalous system call detection[C]// ACM Transactions on Information and System Security. 2006: 61-93.

[12] CREECH G, HU J. A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Discontinuous System Call Patterns[J]. IEEE Transactions on Computers, 2014, 63(4): 807-819.

[13] XIE M, HU J, YU X, et al. Evaluating Host-Based Anomaly Detection Systems: Application of the Frequency-Based Algorithms to ADFA-LD[C]// Network and System Security. Cham: Springer, 2015: 542-549.

[14] SANJEEV D, YANG L, WEI Z, et al. Semantics-based online malware detection: Towards efficient real-time protection against malware[J]. IEEE Transaction on Information Forensics and Security, 2016, 11(2): 289-302.

[15] LV S H, JIAN W, YANG Y Q, et al. Intrusion prediction with system-call sequence-to-sequence model[J]. IEEE Access, 2018 (6): 71413-71421.

[16] KOLOSNJAII B, ZARRAS A, WEBSTER G, et al. Deep Learning for Classification of Malware System Call Sequences[C]// Advances in Artificial Intelligence. Cham: Springer, 2016: 137-149.

[17] ZHAN J, TONG Y, XU M D, et al. A Method for Data Collection and Real-Time Anomaly Detection of Lightweight Hosts [J]. Journal of Xi'an Jiaotong University, 2017, 51(4): 97-102.

[18] XU L F, ZHANG D P, ALVAREZ M A, et al. Dynamic android malware classification using graph-based re-presentations[C]// IEEE International Conference on Cyber Security and Cloud Computing. IEEE. 2016: 220-331.

[19] WUNDERLICH S, RING M, LANDES D, et al. Comparison of System Call Representations for Intrusion Detection[C]// Computational Intelligence in Security for Information Systems and International Conference on European Transnational Education. Cham: Springer, 2019: 14-24.

[20] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[21] CREECH G, HU J. Generation of a new IDS test dataset: Time to retire the KDD collection[C]// IEEE Wireless Communications and Networking Conference (WCNC). 2013: 4487-4492.

WEI Hui, born in 1998, postgraduate. His main research interests include network security and deep learning.

CHEN Ze-mao, born in 1975, Ph.D, professor. His main research interests include information system security, trusted computing and equipment information security.

（责任编辑：李亚辉）