doi:10.11887/j.cn.202305021 http://journal.nudt.edu.cn

利用字节模式二维特征的ROP链智能检测方法*

王 剑,黄恺杰,张梦杰,刘星彤,杨 刚
(国防科技大学 电子科学学院,湖南 长沙 410073)

摘 要:面向返回编程(return oriented programming,ROP)攻击是网络攻击者突破操作系统安全防护、实现漏洞攻击的一种主要手段,ROP链是ROP攻击的重要组成部分。为检测网络流量中的ROP链,提出了一种能自动提取ROP链特征、具有良好泛化性能的智能检测方法。该方法采用顺序抽取的方式将被测流量分成多个序列,利用滑动窗口和数值量化将输入的一维流量数据转换为二维特征向量,基于卷积神经网络模型实现对ROP链的检测。不同于已有的静态检测方法,该方法不依赖程序内存地址的上下文信息,实现简单、部署方便,且具有优异的检测性能。实验结果表明,模型最高准确率为99.4%,漏报率为0.6%,误报率为0.4%,时间开销在0.1 s以内,对真实ROP攻击流量的漏报率为0.2%。

关键词:面向返回编程;静态检测;序列抽取;图像特征
中图分类号:TN918 **文献标志码**:A **开放科学(资源服务)标识码(OSID)**:
文章编号:1001-2486(2023)05-184-09

Intelligent detection method of ROP chain using two-dimensional feature of byte pattern

WANG Jian, *HUANG Kaijie*, *ZHANG Mengjie*, *LIU Xingtong*, *YANG Gang*
(College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China)

Abstract: ROP(return oriented programming) attack is an important method for network attackers to break through the protection of operating system and realize vulnerability attacks, and ROP chain is the main component of ROP attack. In order to detect the ROP chain in network traffic, an intelligent detection method that can automatically extract the characteristics of ROP chain and has good generalization performance was proposed. The sequential extraction method was adopted to divide the measured network traffic into multiple sequences, one-dimensional traffic data was converted into two-dimensional feature vectors by using sliding window and numerical quantization, and the detection of ROP chain was realized based on the convolution neural network model. Different from the existing static detection methods, the proposed method did not rely on the context information of the program memory address, was simple to implement, easy to deploy, and had excellent detection performance. The experimental results show that the highest accuracy rate of the model is 99.4%, the false negative rate is 0.6%, the false positive rate is 0.4%, the time cost is within 0.1 s, and the false negative rate for the real ROP attack traffic is 0.2%.

Keywords: return oriented programming; static detection; sequence extraction; image feature

数据执行保护(data execution prevention,DEP)、地址空间布局随机化(address space layout randomization,ASLR)等内存防护机制的普遍应用,使得传统的代码注入攻击难以奏效。Shacham[1]提出了面向返回编程(return oriented programming,ROP)的概念,打破了传统注入攻击需要注入外部可执行代码的局限,通过选取并拼接目标系统软件内存空间中的已有代码片段(称之为Gadget)来实现恶意功能。Gadget是目标程序内存空间已存在的以跳转指令为结尾的代码片段,可以实现内存读写、数据处理、系统调用和函数调用等功能。Gadget是构成ROP链的基本单元,ROP链通过拼接多个Gadget,完成恶意攻击过程中的特定操作。由于ROP链中没有显性表示的恶意代码,因此可以突破DEP的防护。ROP链在实际漏洞攻击中通常作为攻击代码的重要组成部分,如在针对CVE-2014-0322、CVE-2016-10190、2021-22555、CVE-2022-0995等漏洞攻击的代码中均包含了ROP链[2]。

ROP链的检测分为动态检测与静态检测:

动态检测通过建立蜜罐、沙箱等目标环境,对外部输入数据进行动态执行监控,获取其动态特

* **收稿日期**:2023-02-23
基金项目:教育部中国移动科研基金资助项目(MCM20200103)
作者简介:王剑(1975—),男,湖南邵阳人,教授,博士,博士生导师,E-mail:jwang@nudt.edu.cn

征,如寄存器或内存状态、指令执行状态、控制流完整性等,采用行为规则匹配、执行特性统计分析、内存状态推演等方法实现对 ROP 的检测。DROP 基于动态二进制插桩对代码指令的执行情况进行监控,通过检查“ret”指令间隔的指令数,判断是否存在 ROP 恶意代码片段[3]。蒋廉[4]使用基于内核的虚拟机(kernel-based virtual machine,KVM),利用动态插桩,结合软硬件分层过滤,嵌套检测 ROP 攻击。SCRAP 在 DROP 的基础上根据指令指针的跳转行为定义攻击行为签名,利用签名匹配检测 ROP 攻击[5]。Ropecker 基于硬件辅助的最近分支记录,通过识别间接分支跳转的频率实现对 ROP 攻击的判定[6]。ROPdefender 通过识别软件控制流是否出现异常来检测 ROP 攻击[7]。CFIMon 利用硬件辅助的方法监控 CPU 指令执行,检测破坏控制流完整性的异常跳转事件[8]。ROPDetector[9] 和 MIBChecker[10]利用硬件性能管理单元的事件触发机制,针对预测失败的分支进行实时 ROP 检测。ROPStop 通过对间接分支跳转事件的统计分析,有效地捕获控制流异常事件,进而识别 ROP 攻击[11]。ROPscan 推测性地驱动目标进程地址空间中已经存在的代码的执行,并在运行时检测 ROP 代码的执行[12]。动态检测方法能够得到较为可靠的检测结果,但其需要构建目标软件动态环境并监控执行样本的动态特征,存在环境构建困难、算法开销大、应用部署复杂等缺点。

静态检测根据 ROP 在数据层面上的取值范围和具体数值的排布顺序,基于特征码匹配、字节流统计分析、模拟执行推演等方法直接分析流量数据以实现对 ROP 攻击的检测。EavesROP 利用滑动窗口筛选潜在的可疑数据段,并使用快速傅里叶变换匹配滤波的方式对网络流量中的疑似地址值与已知库文件中的 Gadget 地址值进行匹配,以发现 ROP 攻击[13]。Strop 根据 Gadget 地址的统计特征,如静态地址数量、静态地址距离、相同地址数量、地址范围等,实现对 ROP 链的检测[14]。DeepReturn 采用反汇编的方法分析可疑 ROP 攻击链,并将其编码输入到卷积神经网络中进行分类检测[15]。ROPminer 通过在目标代码段和动态链接库中收集可能的 Gadget,学习 ROP 组件的字节取值特性和组件次序,并运用隐性马尔可夫模型实现对输入数据流的检测[16]。CodeStop 通过模拟执行数据中的可疑 Gadget 地址,对比其语义是否类似调用 Windows 特定函数时对寄存器的赋值模式,从而判断是否存在 ROP 攻击[17]。静态检测方法较之动态检测方法具有开销小、架构简单、部署方便等优点,但是当前的静态检测方法一方面依赖程序内存地址的上下文信息,包括特殊的函数调用地址信息或是受保护进程的内存地址信息,方法的泛用性较差;另一方面存在检测效率低、真实 ROP 攻击检测性能差等问题。

本文提出一种基于字节模式二维特征的 ROP 链检测方法。该方法基于 ROP 链的字节波动特征,通过序列抽取、滑动分组、数值量化,将输入的一维流量数据转换为二维特征向量,保留了更多的数据信息,并采用卷积神经网络实现对 ROP 链的检测。该方法解决了现有方法地址内存信息依赖的问题,无须考虑目标系统环境,包括操作系统版本、目标软件版本,且具有较高的检测准确率、较低的时间开销和系统开销。

1　ROP 链检测模型

1.1　模型设计

本文提出的模型适用于 32 位和 64 位操作系统。为便于理解,下面以 4 B 地址空间的 32 位操作系统为例进行阐述。检测模型的总体方案如图 1所示,包含数据准备、数据预处理、模型训练和模型检测四个部分。数据准备主要是建立 ROP

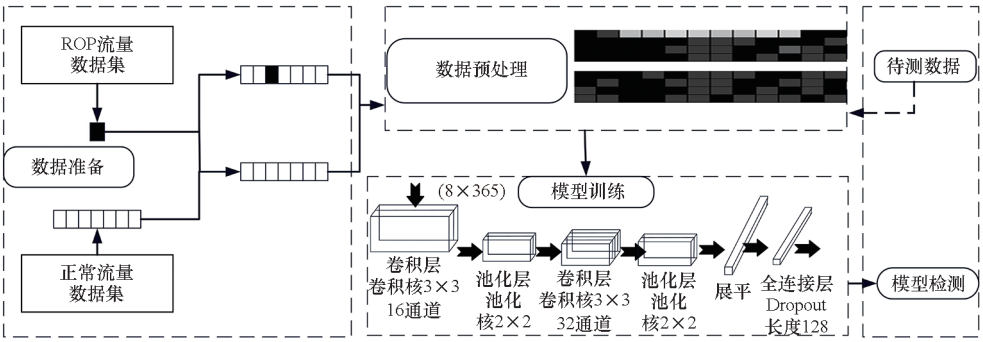


图 1　模型设计总体方案

Fig. 1　General design program of the model

流量数据集和非 ROP 流量数据集;数据预处理部分将流量数据包的有效载荷转换为类灰度图像的二维数值矩阵;模型训练部分将经过预处理后的数值矩阵输入到神经网络模型中进行训练;模型检测部分将测试数据预处理后输入到测试模型中,从而得到所属类别。神经网络模型相较于传统数学模型以及机器学习模型具有更强的特征提取能力,能够自动地、智能地提取类灰度图像的二维数值矩阵特征。同时基于神经网络的检测模型具有优秀的鲁棒性和泛化性,对于未出现在训练集的新 ROP 链具有良好的检测性能。因此,采用神经网络模型对 ROP 链进行检测。

1.2 数据集构建

当前,ROP 链的检测还没有可用的公开数据集。ROPgadget[18]、Ropper[19] 等工具能自动生成 ROP 链。但是一方面这些工具生成 ROP 链的方法过于标准化,使得生成的不同 ROP 链所包含的 Gadget 数量区别不大;另一方面用于生成 ROP 链的 Gadget 通常选自同一代码段,功能较为单一。因此采用这些工具生成的 ROP 链并不能很好地反映真实 ROP 链的特点,不适合神经网络模型的学习训练。

由于构成 ROP 链的 Gadget 的第一字节与其他字节的分布差异大,因此本文基于 ROP 链所在区域的字节波动特征实现对 ROP 攻击的检测,并通过对真实 ROP 链进行分析,依据真实 ROP 链的结构特点来构建数据集。ROP 链数据的生成方式是将真实 Gadget 的首字节与三个随机的字节组合形成扩展的 Gadget,并将若干个扩展 Gadget 拼接组成 ROP 链。通过从 CVE 和 Expliot Database 等收集并分析大量真实攻击样本,发现 ROP 链中 Gadget 的选择来源往往小于 6 个不同的代码区域。为提高模型检测能力,本数据集提取了 10 个不同首字节的代码地址段,每个代码段有 64 MB 的地址空间。研究表明,在 x86 架构下,攻击者实现简单的条件转移逻辑也需要串接 11 个不同的 Gadget[20],因此本数据集设定 ROP 链包含 Gadget 的数量在 10 ~ 50 个不等。ROP 链数据集的构建如表 1 所示,ROP 链的长度表示其包含的 Gadget 数量,地址段数表示 ROP 链包含的 Gadget 的地址段数量,如:地址段数为 1 表示 Gadget 都处于相同的地址段,其对应的首字节相同;地址段数为 2 表示 ROP 链中的 Gadget 有 2 个不同的首字节。该数据集既考虑了 ROP 链的长度,又考虑了 Gadget 的地址范围,总样本量在 12 000 以上,能满足神经网络训练需求。

表 1 ROP 链数据集

Tab. 1 ROP chains data set

地址段数	ROP 链长度			
	10 ~ 20 个	21 ~ 30 个	31 ~ 40 个	41 ~ 50 个
1	400	400	400	0
2	600	600	600	0
3	400	400	400	400
4	400	400	400	400
5	400	400	400	400
6	600	600	600	600
7	0	200	200	200
8	0	200	200	200
9	0	200	200	200
10	0	200	200	200

正常流量数据由 USTC-TFC 数据集构建[21]。USTC-TFC 数据集包含异常流量和正常流量两部分,本文选用其中的正常流量部分进行实验,包含点对点(peer-to-peer,P2P)流量、多媒体流量、文件传输流量、电子邮件流量等。

1.3 数据预处理

ROP 链包含 Gadget 内存地址和填充数据两种数据,均为 4 B,因此 ROP 链在流量中表现为多个 4 B 数据组成的代码串。ROP 链的核心单元是 Gadget,它的字节数值表示所调用的代码段在内存空间的具体位置。ROP 链具有明显的字节模式特征,即当 Gadget 是从同一个代码段或相邻代码段中选取时,Gadget 的第一个字节数值大小相同或相近,且 Gadget 后三个字节的数值是随机的[17]。基于 ROP 链的字节模式特征,本文采用顺序抽取的方法将流量数据分成四组,并采用滑动窗口进行数值量化,最后将一维流量数据转化为二维矩阵,共 4 个步骤,如图 2 所示。

Step 1:对待测流量数据进行标准化清洗。去除数据包协议头部,保留有效负载,输出流量序列 T。假定序列 T 总长 $4n$,舍弃多余的字节。

Step 2:将 ROP 链中所有 Gadget 字节有序分离并组合。将流量序列 T 从首字节开始,以 0、1、2、3 顺序依次标号,并将同一标号数据组合为一组,共输出 4 组数据 T_1、T_2、T_3、T_4。经过第二步操

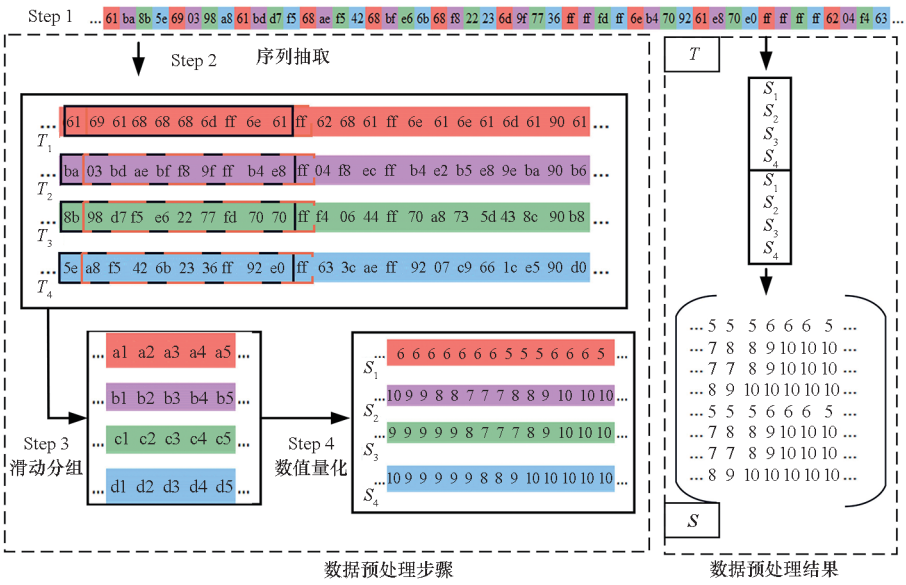


图 2　二维特征数据预处理

Fig. 2　Preprocessing of two-dimensional feature data

作,ROP 链中的 Gadget 的首字节一定属于这 4 组数据中的某一组,这组数据的数值变化较小,波动平稳。另外 3 组数据分别包含 Gadget 的后 3 B,其数值波动较大。

Step 3:在 T_1、T_2、T_3、T_4 上选取窗长为 L、步长为 1 的滑动窗口,对滑动窗口内的字节波动大小进行量化,量化方式有类化、熵化、拟熵化等。类化是统计滑动窗口内不同字节数值的个数,熵化是计算滑动窗口内字节的熵值,拟熵化是对熵化的改进。$H_1(x)$、$H_2(x)$、$H_3(x)$ 分别表示采用类化、熵化、拟熵化的滑动窗口内的波动量化值。

$$H_1(x) = n \tag{1}$$

$$H_2(x) = -\sum_{i=1}^{n} \frac{p(x_i)}{L} \log_2 \frac{p(x_i)}{L} \tag{2}$$

$$H_3(x) = -\sum_{i=1}^{n} p(x_i) \cdot p(x_i) \tag{3}$$

其中,n 表示滑动窗口内不同字节数值的个数,$p(x_i)$ 表示滑动窗口内不同字节出现的次数。

Step 4:单步滑动窗口,输出 4 组数据 S_1、S_2、S_3、S_4,输出序列长度为 365,长度不足则补零,叠加组合输出矩阵 $\boldsymbol{S}$。这一步是得到神经网络所用的标准数据,对数据进行叠加组合可以提取数据边缘信息。

本文所提数据预处理方法,是在分析 ROP 链中 Gadget 不同字节波动特征的基础上,通过序列抽取的方式,并结合滑动窗口对数据进行量化,从而实现数据特征的有效提取。该方法将一维流量数据转为类灰度图像的二维数值矩阵,包含了更为丰富的数据信息,且不需要预知目标软件的地址信息。

1.4　卷积神经网络结构

ROP 链在网络流量中的局部特征明显,且 ROP 链的检测结果与其在流量中的位置无关。卷积神经网络具有很强的局部上下文特征提取能力,并具有平移不变性的特点。经实验验证,卷积神经网络在 ROP 链检测任务中的准确率优于深度神经网络以及循环神经网络。因此,将卷积神经网络作为检测模型。

ROP 链的检测有实时性要求。本文设计了简单且轻量级的卷积神经网络结构,交替使用卷积层和池化层提取局部特征并减少模型参数,同时应用 Dropout 正则化增强模型对噪声的鲁棒性。检测模型的结构如图 3 所示。输入 8 × 365 的定长二维数值矩阵,在第 1 个二维卷积层中先对输入进行卷积,卷积核为 3 × 3,共 16 个通道,生成 16 个长为 6 × 363 的特征图。然后通过二维池化核 2 × 2 的最大池化层生成 16 个长为 3 × 181 的特征图。在第 2 个二维卷积层中,卷积核为 3 × 3,共 32 个通道,生成 32 个长为 2 × 181 的特征图,然后通过二维池化核 2 × 2 的最大池化层,生成 32 个长为 1 × 90 的特征图。完成上述操作后,将特征序列展平,通过两个全连接层将序列依次转换为 128 和 1。模型使用的超参数如下:使用参数为 0.6 的 Dropout 正则化,输出层采用 Sigmoid 函数作为激活函数,其他层使用 ReLU 函数作为激活函数,ReLU 函数没有复杂的指数运算,计算简单、运算效率高。学习率设为 1×10^{-4},训练轮次为 30,训练总参数为 37 万余个,损失函数采用二分类交叉熵损失函数。

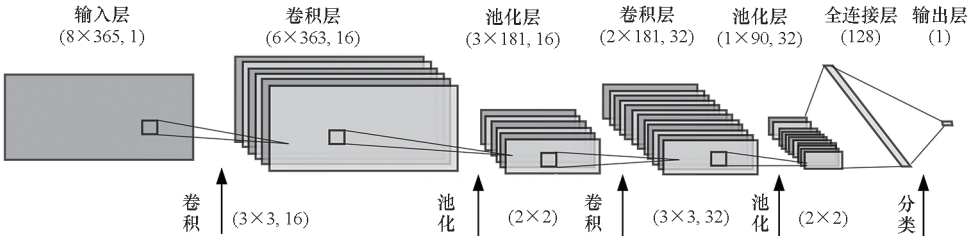


图 3　卷积神经网络结构

Fig. 3　Structure of convolutional neural network

2　实验分析

2.1　数据预处理参数选择实验

滑动窗口长度 L 以及量化边界 D 是影响模型数据预处理效果的两个关键特征。为探究这两个变量对算法性能的影响，依据相邻代码段首字节数值大小相近的特点，设置了长度范围为 5 ~ 24 和量化边界为 0 ~ 3 的循环实验，共 80 个训练模型进行性能对比，对比结果如图 4 ~ 6 所示。

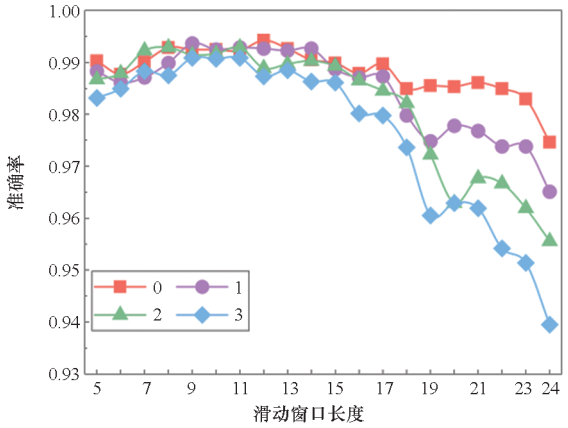


图 4　窗口长度与量化边界对准确率的影响

Fig. 4　Influence of window length and quantization boundary on accuracy rate

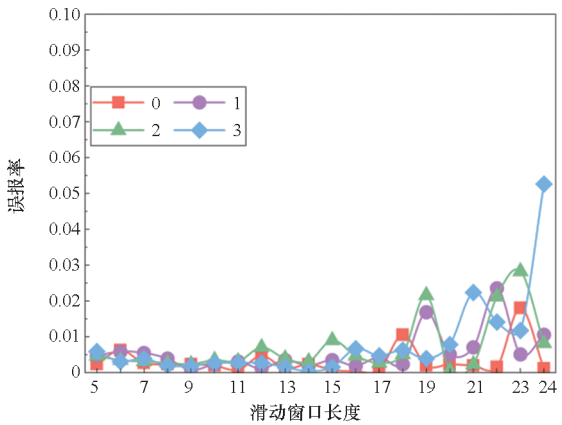


图 5　窗口长度与量化边界对误报率的影响

Fig. 5　Influence of window length and quantization boundary on false alarm rate

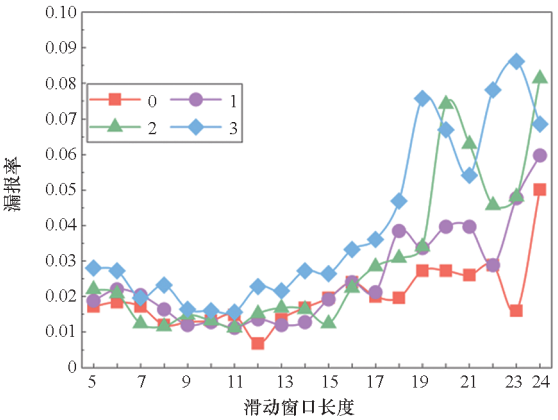


图 6　窗口长度与量化边界对漏报率的影响

Fig. 6　Influence of window length and quantization boundary on miss alarm rate

总体来讲，采用不同窗长和量化边界均能有效区分 ROP 链，准确率在 94% 以上，最高可达 99.4%。通过对不同窗长和量化边界的模型性能进行分析，可以发现：当窗长大于 15 时，模型的准确率下降较快，且量化边界越大准确率越低；当窗长大于 12 时，模型的漏报率上升明显，且量化边界越大漏报率越高；模型的误报率受窗口长度和量化边界的影响较小；当窗长大于 18 时，模型性能波动剧烈，因为窗长过长容易涵盖 ROP 链区域外的正常流量字节，造成模型性能不稳定。模拟训练时的损失函数收敛曲线如图 7 所示，随着训练轮次的增加，检测准确率稳定上升，误差损失值平稳下降。

2.2　模型性能评价实验

实验使用的神经网络框架为 TensorFlow。实验硬件 GPU 为 GeForce RTX 3060 Laptop，CPU 为 11th Gen Intel（R） Core（TM） i7 - 11800H @ 2.30 GHz，内存为 16 GB。

实验选取长度在 150 ~ 1 550 范围内的数据包，滑动窗口长度设为 12。为验证不同预处理方法模型的检测效率，将数据包分别输入采用不同量化方法的二维特征检测模型中，测算从数据输

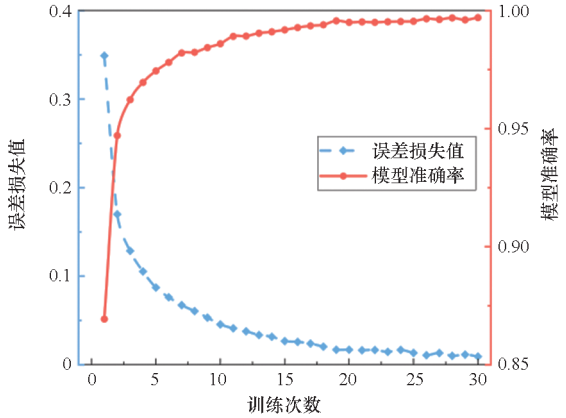


图 7　损失函数收敛曲线

Fig. 7　Loss function convergence curve

入到检测结果输出的时间开销,实验结果如图 8 所示,图中散点表示对于单个数据包检测的时间开销,而直线是对这些单个数据包检测开销的拟合结果。三种量化方式的时间开销均在 0.14 s 以内,类化方式所用时间开销明显低于熵化和拟熵化,类化方式大部分数据包的检测时间在 0.1 s 以内,拟熵化模型次之,熵化模型时间开销最大。

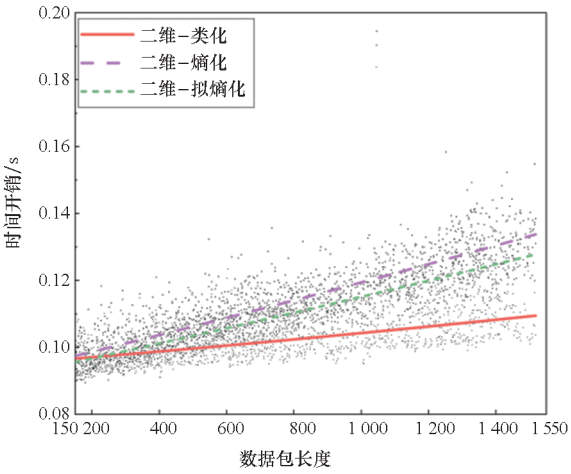
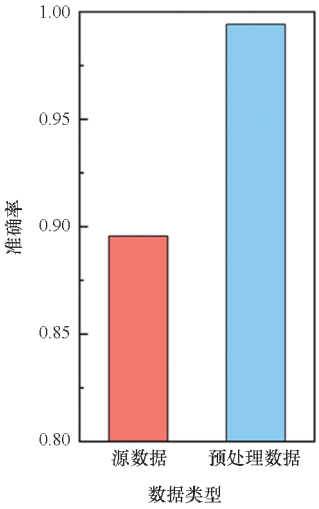


图 8　不同量化方法的时间开销

Fig. 8　Time costs of different quantization methods

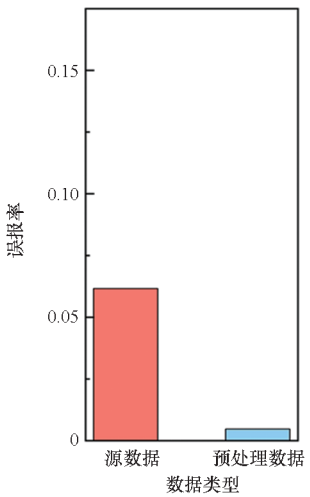
为验证数据预处理的有效性,开展了未经数据预处理的源数据和数据预处理后数据的模型性能对比实验,如图 9 所示。实验结果表明:经过数据预处理后,模型的准确率提升 9.9%,误报率降低 5.7%,漏报率降低 14.1%,充分验证了数据预处理方法和检测模型的有效性。

为验证模型对真实 ROP 链的检测性能,分别选取 ROPgadget、Exploit Database、CVE 中的 ROP 链进行检测实验。通过 ROPgadget 生成的 ROP 链 188 个,分别模拟 10 次攻击。Exploit Database 漏洞库中漏洞利用程序的 ROP 链共计 445 个,分别模拟 10 次攻击。在 Github 中收集



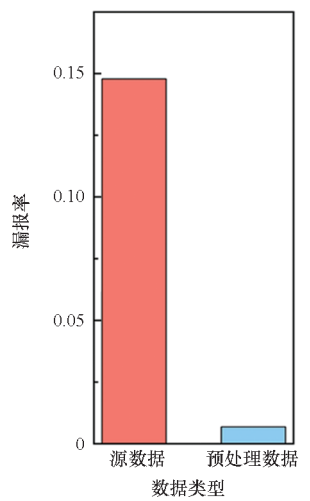
(a) 准确率对比

(a) Accuracy rate comparison



(b) 误报率对比

(b) False alarm rate comparison



(c) 漏报率对比

(c) Miss alarm rate comparison

图 9　二维特征预处理前后模型性能对比

Fig. 9　Comparison of model performance before and after two-dimensional feature preprocessing

的 CVE 漏洞利用程序的 ROP 链 49 个，分别模拟 100 次攻击。实验表明，经过数据预处理后的模型具有较强的泛化性能，能够以极低的漏报率实现对真实 ROP 链的有效检出，而未经数据预处理的模型漏报率非常高，如表 2 所示。实验发现一个有趣的结果，两种模型检测由 ROPgadget 工具产生的 ROP 链的漏报率均为 0，因为 ROP 链自动生成工具的生成逻辑往往会遵循某些固定范式，产生的 ROP 链也非常规范，容易被检测。如果采用此类工具产生的 ROP 链来训练神经网络模型，模型的训练性能非常好，但是实际检测性能往往非常差。

表 2　不同来源真实 ROP 链检测的漏报率

Tab. 2　False negative rate of real-world ROP chains

ROP 链来源	数量	模拟攻击频次	预处理后的漏报率/%	数据未处理漏报率/%
ROPgadget	188	10	0	0
Exploit Database	445	10	0.15	25
CVE	49	100	0.65	17

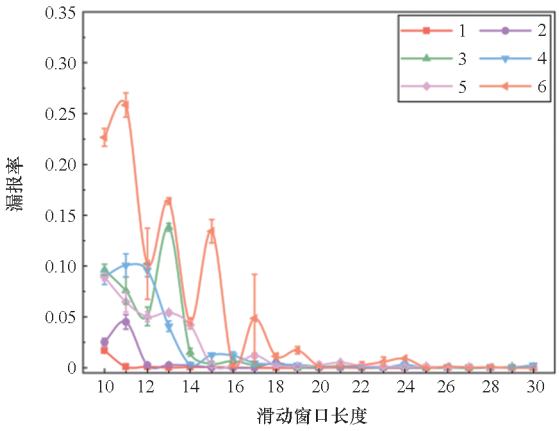
2.3　模型能力探究实验

ROP 链的长度决定了目标区域的长度，其长度越长，特征也越明显。以往的研究很少关注 ROP 链的长度对模型性能的影响。实验分析二维特征检测模型对 6 个分组（每个分组的地址段数不同，如表 1 所示）中不同长度（10 ~ 30）ROP 链的检测能力（用漏报率衡量）。模型类别采用 $L-D$ 表示，L 为窗长，D 为量化边界。选用最高准确率模型 12 -0、最短窗长模型 5 -0 和较长窗长模型 17 -0 进行实验。由于量化边界为 0 时，模型性能较优，因此模型的量化边界统一选择为 0。

实验表明窗长和地址段数对模型的检出性能影响较大，如图 10 所示，随着 ROP 链长度的增加，漏报率均可降低至 1% 以下。在窗长相同的情况下，地址段数越多，则模型漏报率越高、波动越大，因为地址段数越多，ROP 链的字节特征与正常流量字节特征更接近。窗长越短，模型对于不同地址段数的漏报率的波动越大，因为短窗口对连续区域的波动量化能力有限，无法很好地规避波动干扰。窗长越长，模型对于不同地址段数的漏报率变化趋势越相似，因为长的窗口可以抵消地址段数增加带来的负面影响。

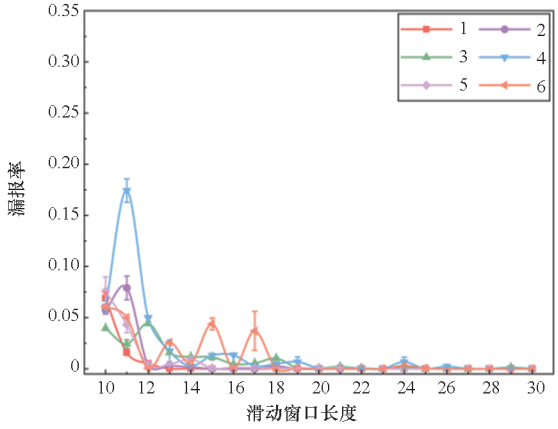
2.4　对比分析实验

ROPminer、SPRT[22]、DeepReturn 等模型虽然



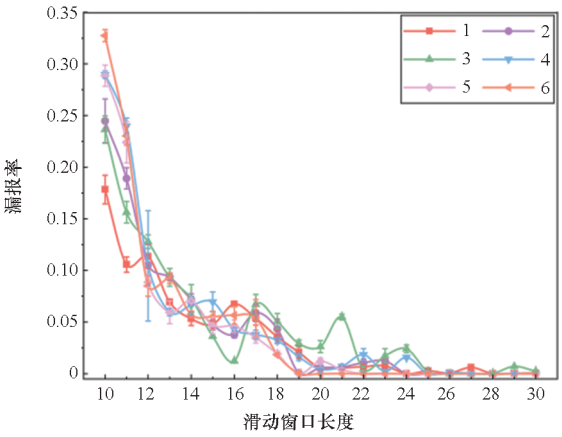
（a）5 -0 模型

（a）5 -0 model



（b）12 -0 模型

（b）12 -0 model



（c）17 -0 模型

（c）17 -0 model

图 10　模型漏报率

Fig. 10　Miss alarm rate of model

具有较高的准确率，但这些方法都依赖程序的内存地址信息。本文提出的方法与已有方法最关键的区别在于不依赖程序的内存地址信息，具有更低的复杂度和更广泛的应用场景，对比数据如表 3所示。虽然 Strop 方法也不依赖内存地址信

息,但该方法对于 68 个 ROP 样本仅检测到 51 个,漏报率高达 25%。本文方法、SPRT 和 ROPminer 都是完全静态的方法,仅根据网络数据流就可实现 ROP 链的检测。DeepReturn 需要对流量字节进行反汇编,虽然其误报率较低,但该方法需要结合内存地址信息,通用性较差。SPRT 和本文方法探索了 ROP 链长度对算法性能的影响。在 SPRT 中,当训练数据集 ROP 链长度为 10 ~ 50 时,准确率仅为 93.2%;当训练数据集 ROP 链长度为 20 ~ 100 时,准确率才达到 99.0%。本文方法在训练数据集 ROP 链长度为 10 ~ 50 时,准确率可达 99.4%。因此本文方法在检测较短长度的 ROP 链时具有更高的准确率。Strop、ROPminer、DeepReturn 等方法均采用真实 ROP 样本验证方法的有效性,但本文用于实验验证的真实样本数量远多于其他方法。以上方法中,ROPminer 进行了时间开销实验,单文件的时间开销为 0.9 s,本文方法的时间开销在 0.1 s 以内。

表 3　与现有方法的对比分析

Tab. 3　Comparison with existing methods

检测方法	是否需要内存地址信息	真实 ROP 样本验证数量	真实 ROP 样本漏报率/%	正常流量误报率/%	所用 ROP 链长度	单数据包时间开销
Strop	否	68	25	1.3	未说明	未说明
ROPminer	是	15	0	3.0	未说明	0.9 s
DeepReturn	是	100	0	0.01	未说明	未说明
SPRT	是	未说明	1	0.4	20 ~ 100	未说明
本文方法	否	494	0.2	0.4	10 ~ 50	0.1 s

3　结论

本文提出了一种基于字节模式二维特征的 ROP 链检测方法,首先对原始流量数据包进行数据清洗,然后对清洗后的数据进行序列抽取,经滑动窗口采样后对数据进行量化处理,将输入的一维数据转换为二维特征向量,并采用卷积神经网络模型实现对 ROP 链的检测。该方法能自动从样本中提取 ROP 链字节模式二维特征,并具有良好的泛化能力,能检测训练集中未出现过的新 ROP 链。模型的检测准确率达到 99.4%,单数据包的时间开销在 0.1 s 以内,且模型对真实 ROP 样本的测试漏报率接近于 0,相较于 Strop、ROPminer、SPRT 等传统方法,在检测性能上具有明显优势。同时,该方法可直接对流量数据进行检测,不需要预知代码段加载内存地址,部署方便并具有良好的可解释性。下一步将深入分析不同类型漏洞样本的特征,建立更为丰富的漏洞攻击样本库,并研究基于语义分析的检测模型,进一步提高模型的泛化性能,以及模型对不同类型漏洞攻击的检测能力。

参考文献(References)

[1] SHACHAM H. The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86)[C]// Proceedings of the 14th ACM Conference on Computer and Communications Security, 2007: 552 - 561.

[2] 乔延松, 杜皓睿, 赵绪营. ROP 攻击原理与检测方法研究[J]. 北京电子科技学院学报, 2021, 29(4): 51 - 56.
QIAO Y S, DU H R, ZHAO X Y. Research on the principle and detection method of return-oriented programming attack[J]. Journal of Beijing Electronic Science and Technology Institute, 2021, 29(4): 51 - 56. (in Chinese)

[3] CHEN P, XIAO H, SHEN X B, et al. DROP: detecting return-oriented programming malicious code [C]// Proceedings of the 5th International Conference on Information Systems Security, 2009: 163 - 177.

[4] 蒋廉. 一种基于程序运行期间指令特征的 ROP 攻击检测方法[D]. 成都: 电子科技大学, 2021.
JIANG L. A method based on instruction characteristics during program operation for detecting ROP attack[D]. Chengdu: University of Electronic Science and Technology of China, 2021. (in Chinese)

[5] KAYAALP M, SCHMITT T, NOMANI J, et al. SCRAP: architecture for signature-based protection from code reuse attacks [C]//Proceedings of IEEE 19th International Symposium on High Performance Computer Architecture (HPCA), 2013: 258 - 269.

[6] CHENG Y Q, ZHOU Z W, YU M, et al. ROPecker: a generic and practical approach for defending against ROP attack [C]// Proceedings of the Network and Distributed System Security Symposium, 2014.

[7] DAVI L, SADEGHI A R, WINANDY M. ROPdefender: a detection tool to defend against return-oriented programming attacks[C]//Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, 2011: 40 - 51.

[8] XIA Y B, LIU Y T, CHEN H B, et al. CFIMon: detecting violation of control flow integrity using performance

counters[C]// Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2012.

[9] 牛伟纳, 赵成洋, 张小松, 等. ROPDetector: 一种基于硬件性能计数器的ROP攻击实时检测方法[J]. 计算机学报, 2021, 44(4): 761 - 772.
NIU W N, ZHAO C Y, ZHANG X S, et al. ROPDetector: a real-time detection method of ROP attack based on hardware performance counter [J]. Chinese Journal of Computers, 2021, 44(4): 761 - 772. (in Chinese)

[10] 李威威, 马越, 王俊杰, 等. 基于硬件分支信息的ROP攻击检测方法[J]. 软件学报, 2020, 31(11): 3588 - 3602.
LI W W, MA Y, WANG J J, et al. ROP attack detection approach based on hardware branch information[J]. Journal of Software, 2020, 31(11): 3588 - 3602. (in Chinese)

[11] JACOBSON E R, BERNAT A R, WILLIAMS W R, et al. Detecting code reuse attacks with a model of conformant program execution [C]//Proceedings of International Symposium on Engineering Secure Software and Systems, 2014: 1 - 18.

[12] POLYCHRONAKIS M, KEROMYTIS A D. ROP payload detection using speculative code execution[C]//Proceedings of the 6th International Conference on Malicious and Unwanted Software, 2011: 58 - 65.

[13] JÄMTHAGEN C, KARLSSON L, STANKOVSKI P, et al. EavesROP: listening for ROP payloads in data streams[C]// Proceedings of Conference on Information Security, 2014: 413 - 424.

[14] CHOI Y H, LEE D H. STROP: static approach for detection of return-oriented programming attack in network[J]. IEICE Transactions on Communications, 2015, E98. B(1): 242 - 251.

[15] LI X S, HU Z S, WANG H Z, et al. DeepReturn: a deep neural network can learn how to detect previously-unseen ROP payloads without using any heuristics [J]. Journal of Computer Security, 2020, 28(5): 499 - 523.

[16] USUI T, IKUSE T, OTSUKI Y, et al. ROPminer: learning-based static detection of ROP chain considering linkability of ROP gadgets [J]. IEICE Transactions on Information and Systems, 2020, E103. D(7): 1476 - 1492.

[17] OCONNOR T, ENCK W. Code-Stop: code-reuse prevention by context-aware traffic proxying [C]//Proceedings of Conference on Internet Monitoring and Protection, 2016.

[18] SALWAN J. ROPgadget [CP/OL]. [2023 - 02 - 01]. https://github.com/JonathanSalwan/ROPgadget.

[19] SCHIRRA S. Ropper-rop gadget finder and binary information tool[CP/OL]. [2023 - 02 - 01]. https://scoding.de/ropper/.

[20] 张梦杰, 王剑, 黄恺杰, 等. 一种基于字节波动特征的ROP流量静态检测方法[J]. 信息网络安全, 2022(7): 64 - 72.
ZHANG M J, WANG J, HUANG K J, et al. A static detection method of ROP traffic based on bytes fluctuation characteristics[J]. Netinfo Security, 2022(7): 64 - 72. (in Chinese)

[21] WANG W, ZHU M, ZENG X W, et al. Malware traffic classification using convolutional neural network for representation learning [C]// Proceedings of International Conference on Information Networking (ICOIN), 2017: 712 - 717.

[22] HO J W. Efficient and robust detection of code-reuse attacks through probabilistic packet inspection in industrial IoT devices[J]. IEEE Access, 2018, 6: 54343 - 54354.