面向行为多样期的挖矿恶意软件早期检测方法

曹传博[1],郭 春[1],申国伟[1],崔允贺[1],平 源[2]
(1. 贵州大学计算机科学与技术学院公共大数据国家重点实验室,贵州贵阳 550025;2. 许昌学院信息工程学院,河南许昌 461000)

摘 要: 挖矿恶意软件是一种隐匿在受害主机中,在未经用户许可的情况下使用系统资源挖掘加密货币的恶意软件,其不仅影响计算机系统的正常运行也会危害系统安全.目前基于动态分析的挖矿恶意软件检测方法主要以挖矿恶意软件的工作量证明行为为检测对象,难以实现对此类软件的及时检测.针对上述问题,通过分析挖矿恶意软件的运行过程,发现挖矿恶意软件在建立网络连接前行为多样,由此提出"挖矿软件行为多样期(Behavioral Diversity Period of Cryptominer,BDP)"的概念并进一步提出面向行为多样期的挖矿恶意软件早期检测方法(Cryptomining Malware Early Detection Method in Behavioral Diversity Period,CEDMB).CEDMB使用*n*-gram模型和TF-IDF(Term Frequency-Inverse Document Frequency)算法从BDP内的API(Application Programming Interface)序列中提取特征以训练检测模型.实验结果显示,CEDMB使用随机森林算法时可以在软件开始运行后10 s内以96.55%的F1-score值判别其是良性软件还是挖矿恶意软件.

关键词: 挖矿恶意软件;动态分析;早期检测;随机森林;API序列
基金项目: 国家自然科学基金(No.62162009);贵州省科技支撑计划(No.黔科合支撑[2022]一般071);河南省科技攻关计划项目(No.212102210084)
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112(2023)07-1850-09
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220926

Cryptomining Malware Early Detection Method in Behavioral Diversity Period

CAO Chuan-bo[1], GUO Chun[1], SHEN Guo-wei[1], CUI Yun-he[1], PING Yuan[2]
(1. *State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou* 550025, *China*;
2. *School of Information Engineering, Xuchang University, Xuchang, Henan* 461000, *China*)

Abstract: Hiding in victim hosts, cryptomining malware utilizes system resources to mine cryptocurrencies without permission. It not only affects the normal operations of computer systems but also endangers system security. The existing dynamic analysis based cryptomining malware detection methods mainly focus on proof-of-work behaviors as the detection object, which can hardly prevent mining and other malicious behaviors from damaging the system in time. To tackle this issue, by analyzing the running process of cryptomining malware, we find that the cryptomining malware performs diverse behaviors before establishing a network connection. Based on this, we give a concept of behavioral diversity period of cryptominer (BDP) and then propose a cryptomining malware early detection method in behavioral diversity period (CEDMB). CEDMB trains the detection model with features extracted from application programming interface sequences following *n*-gram and TF-IDF. Experimental results show that when a random forest algorithm is adopted, the proposed CEDMB can determine whether a sample is a cryptomining malware sample in 10 seconds with an F1-score of 96.55%.

Key words: cryptomining malware; dynamic analysis; early detection; random forest; API sequence
Foundation Item(s): National Natural Science Foundation of China (No.62162009); Science and Technology Support Program of Guizhou Province (No.[2022]071); Key Technologies R&D Program of Henan Province(No.212102210084)

收稿日期:2022-08-05;修回日期:2023-03-09;责任编辑:李勇锋

1 引言

攻击者通过挖矿劫持攻击可在未经用户许可的情况下,使用受害主机的系统资源挖掘加密货币来获益.该类攻击会影响系统正常运行及减损设备寿命.根据实现方式的不同可以将挖矿劫持攻击分为基于浏览器[1]、基于可执行文件[2]和无文件挖矿三类.本文的研究对象为基于可执行文件的挖矿劫持攻击,即挖矿恶意软件.近两年来,挖矿恶意软件呈现出明显的增长趋势,《NTT Security 2021 全球威胁情报报告》显示,2020年挖矿恶意软件取代间谍软件[3]成为全球最常见的恶意软件类型[4].

挖矿恶意软件运行中占比最高的行为是工作量证明,即利用系统资源进行大量hash计算,一些基于行为的恶意软件检测方法往往会将挖矿恶意软件误判为高负载的良性软件而发生漏报.另外,一些诸如代码混淆、更换域名[2]、限制进程CPU利用率[1]等逃逸技术的使用也使其难以被传统的基于黑名单或规则匹配的检测方法准确检测.基于静态分析的检测方法难以应对代码混淆等逃逸技术.基于流量的检测方法则需要等待挖矿恶意软件产生挖矿流量后才能实现检测.基于机器学习的动态检测方法是目前的研究热点.但是现有该类方法所关注的行为主要是挖矿恶意软件执行工作量证明的行为,因此难以在挖矿恶意软件执行工作量证明行为之前将其检出.

针对上述问题,本文从及时检测挖矿恶意软件的目标出发,从API(Application Programming Interface)的角度分析其运行过程.基于发现的挖矿恶意软件在建立网络连接前行为多样、连接建立后行为相对单一的现象,本文创新性的将挖矿恶意软件的检测重点放在挖矿恶意软件建立网络连接前的行为,提出"挖矿软件行为多样期(Behavioral Diversity Period of cryptominer,BDP)"的概念并通过socket/WSAsocket定位该时间段,再以此为基础提出面向行为多样期的挖矿恶意软件早期检测方法(Cryptomining malware Early Detection Method in Behavioral diversity period,CEDMB).CEDMB兼顾挖矿恶意软件检测的及时性和准确性,以BDP内产生的API序列作为分析对象,使用*n*-gram[5]和TF-IDF[6](Term Frequency-Inverse Document Frequency)生成特征向量,通过分类算法建立检测模型.

本文主要工作如下.

(1)从API调用、API序列熵值等方面分析挖矿恶意软件的运行过程,发现挖矿恶意软件在其运行初期表现出区别于良性软件的行为特性,以此为基础提出BDP的概念.将挖矿恶意软件的检测重点聚焦于挖矿恶意软件建立网络连接前的行为,为设计挖矿恶意软件早期检测方法奠定了基础.

(2)提出面向BDP的挖矿恶意软件早期检测方法CEDMB.CEDMB采集待测对象在BDP内调用的API序列,运用*n*-gram和TF-IDF算法对API序列进行特征计算,最后使用机器学习算法进行模型训练及检测.

(3)使用涉及12个币种的多款挖矿恶意软件以及多种类型的良性软件组成的样本集对CEDMB进行测试.实验结果表明,CEDMB能够准确地检测出处于运行初期的已知和未知的挖矿恶意软件样本.

2 相关工作

目前挖矿劫持攻击检测领域主要针对基于浏览器和基于可执行文件两种类型展开检测研究,检测方法可分为基于静态分析的检测方法、基于动态分析的检测方法和基于流量分析的检测方法.

2.1 静态检测方法

静态检测方法分析挖矿恶意软件的代码或文件属性,可以直接对静态文件进行检测而无需运行样本.Darabian等人[7]和Yazdinejad等人[8]提取挖矿恶意软件的操作码序列并使用深度学习算法建立检测模型.Naseem等人[9]将挖矿网页的WebAssembly脚本编译生成的二进制文件转换为灰度图,结合卷积神经网络算法构建了一个轻量高效的检测模型.郑锐等人[9]提出一种基于威胁情报层次特征集成的挖矿恶意软件检测方法,使用多种静态特征训练检测模型.静态检测方法具有较高的检测效率,但是难以对抗一些采用了混淆等绕过技术的挖矿恶意软件[11].

2.2 动态检测方法

基于主机的动态检测方法在受控的主机环境中执行挖矿恶意代码或挖矿恶意软件,分析其执行过程中表现出的特征.Ning等人[12]和Mani等人[13]关注挖矿恶意软件运行时对主机性能的影响,将处理器、内存等硬件资源的使用情况作为数据训练检测模型,取得了不错的效果.但该类检测方法需要等待系统性能受到影响后才能检测出挖矿恶意软件,且存在易受同一硬件环境中其他进程影响以及移植性差等缺点.Darabian等人[7]基于挖矿恶意软件运行时会调用Windows加密函数库的API这一发现,使用API序列结合深度学习实现检测.Karn等人[14]使用*n*-gram模型从8种挖矿恶意软件和8种良性软件运行产生的API序列中提取特征,并使用决策树算法构建检测模型.Berecz等人[15]选择与挖矿行为相关的10个API和5个动态链接库(Dynamic Link Library,DLL)以及另外5个PE(Portable Executable)文件属性作为特征,使用机器学习建立检测模型.总体来说,基于API的动态检测方法可以较好地应对应用了混淆技术的挖矿恶意软件,但目前该类方法或需要较长的API序列采集时间,或主要关注样本执行工作

量证明的行为,难以兼顾挖矿恶意软件检测的及时性和准确性.

2.3 流量检测方法

基于流量的检测方法大多关注挖矿流量通信行为的统计特征或序列特征,结合机器学习算法进行检测. Pastor等人[16]提出了一种使用51个通信行为统计特征的挖矿加密流量检测方法,对来自xmr-stak和xmrig的流量进行检测,取得了较好的检测结果. Zhang等人[17]提出了一种基于时间序列的挖矿流量检测方法,根据网络流序列和区块创建序列的相似度实施对xmrig挖矿流量的检测. 基于流量分析的检测方法可以监控网络入口的流量,覆盖所监控网络内的所有设备. 但是该类方法需要在挖矿恶意软件成功连接矿池之后才能收集到流量来进行检测. 同时,一些诸如延迟发送数据包、增加无用数据包、添加代理等逃逸手段的出现也影响了该类检测方法的效果[17]. 为实施挖矿劫持的早期检测,Sun等人[18]提出了一种挖矿早期流量的卷积特征提取方法. 该方法借助卷积函数从挖矿流量的前数个数据包构成的包负载大小序列中提取特征,再使用机器学习建立检测模型. 但该方法同样需等待挖矿恶意软件成功连接矿池并产生流量后才能实施检测.

综上,如何兼顾挖矿恶意软件检测的及时性和准确性,是当前挖矿恶意软件检测领域的一个技术挑战. 针对该挑战,本文从提高动态检测方法检测及时性的意图出发,重点关注挖矿恶意软件运行初期所调用API序列的特性,并以此为基础提出挖矿恶意软件早期检测方法.

3 挖矿恶意软件行为分析

3.1 挖矿恶意软件攻击过程

挖矿恶意软件的攻击过程可以分为准备阶段、入侵阶段、攻击阶段[2]. 攻击者在准备阶段开发挖矿恶意软件. 在入侵阶段,攻击者会使用钓鱼网站、弱口令爆破、漏洞利用等方式将挖矿恶意软件植入受害者主机. 在攻击阶段,挖矿恶意软件会密集地执行一系列行为,如信息收集、检查防护进程或系统监控进程的存在、生成副本、修改计划任务和自启动项等. 之后,挖矿恶意软件首先会利用socket与矿池建立连接[16],接着不断从矿池接收计算任务[19],然后提交结果,最后其收益由攻击者获得[13].

从上述攻击过程可以看到,挖矿恶意软件会在运行初期密集地执行一系列行为,然后才连接矿池并进行重复单一的工作量证明. API调用可以反映软件运行时的行为轨迹,因此本文接下来从API调用的角度分析挖矿恶意软件行为.

3.2 挖矿恶意软件API序列分析

我们收集了覆盖多种加密货币类型的75款挖矿恶意软件,在VMware创建的Windows10虚拟机中收集各样本在其默认的进程CPU利用率阈值下运行1分钟的API序列,从API调用的角度对这些API序列进行分析. 本文将对应特定行为且相对固定的API调用称为特征API调用. 分析结果显示上述挖矿恶意软件在运行初期会频繁调用与内存操作、线程读写操作相关的特征API调用,以及与3.1节所提到的行为对应的特征API调用,如与修改计划任务相关的NtNotifyChangeKey→RtlInitAnsiStringEx(“→”表示API间存在调用关系). 之后挖矿恶意软件会调用socket/WSAsocket来连接矿池. 在与矿池建立网络连接之后,其调用的API逐渐变得单一,具体表现为不断重复调用与工作量证明相关的特征API调用,如sleep→NtDelayExecution→memcpy. 该结果与3.1节中挖矿恶意软件的运行过程一致.

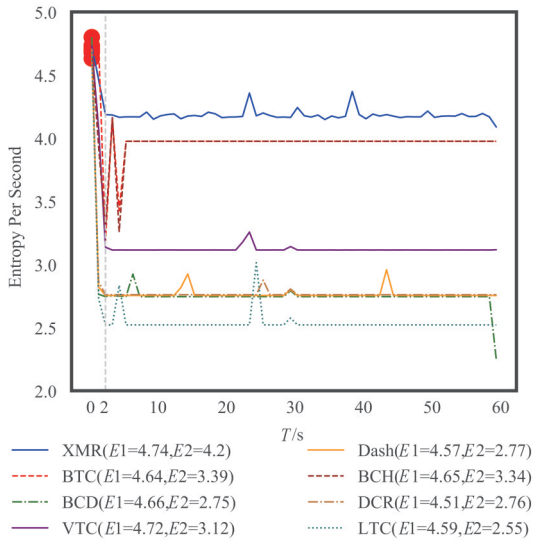
3.3 挖矿软件行为多样期

通过上述分析,我们发现:挖矿恶意软件普遍需要建立网络连接,且以隐蔽、持久化、逃逸检测为目的的大量相关行为主要存在于建立网络连接之前. 为了进一步分析该行为特性,我们分析了XMR、BTC、BCD、VTC、Dash、BCH、DCR、LTC8种类型的8个挖矿恶意软件样本的API序列. 为进行比较,也分析了8种不同类型的良性软件,包括影音软件、办公软件、压测软件、远控软件等. API序列的熵值可以体现特征API调用的多样性,熵值低说明API序列中存在大量重复的API调用[20],即有不断重复的行为. API序列的熵值可使用式(1)计算,其中n表示序列中有n种字符,$p(x_j)$为字符x_j在序列中出现的频率. 各软件每秒API序列的熵值如图1所示,图中$E1$为样本前2 s的API序列的熵值,$E2$为2 s到60 s的API序列的熵值.

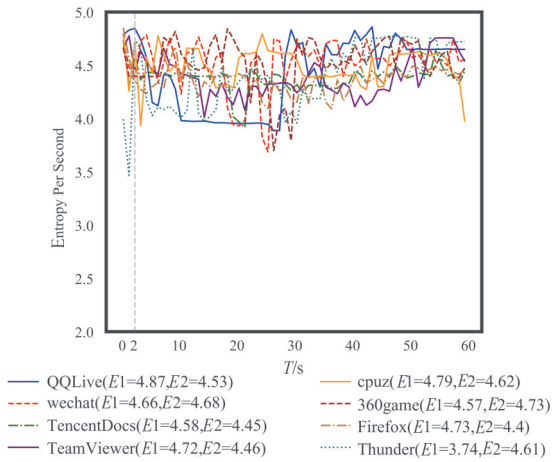
$$E=-\sum_{j}^{n}p(x_j)\log\left(p(x_j)\right) \tag{1}$$

从图1(a)可以看到,各挖矿恶意软件的每秒API序列熵值在运行初期时较大,然后快速下降并出现波动,一定时间之后则较为稳定. 同时各挖矿恶意软件在前2 s的熵值$E1$均明显高于其2 s至60 s的熵值$E2$. 值得注意的是,所分析的挖矿恶意软件均调用socket/WSASocket来建立流套接字以实现网络连接且调用时间(在图1(a)中用红点标记)均在熵值稳定前. 该结果表明:所分析的挖矿恶意软件在建立网络连接前行为多样,连接建立后行为相对单一. 与此不同,图1(b)显示所分析的良性软件各时刻的每秒API序列熵值波动不断,以及$E1$与$E2$数值接近.

挖矿恶意软件需要通过连接矿池来执行其挖矿操作[2,16,18],故建立网络连接是挖矿恶意软件开始挖矿操作前的必要步骤. 因此,本文以网络连接建立与否来判



(*a*) 8个挖矿恶意软件样本的每秒API序列熵值



(*b*) 8个良性软件样本的每秒API序列熵值

图1　不同样本的每秒API序列熵值

断挖矿恶意软件是否进入挖矿阶段,并关注其挖矿阶段前的行为. 基于以上分析,本文提出BDP的概念,其在挖矿恶意软件运行过程中所处时间段如图2所示.

定义1　挖矿软件行为多样期(BDP)

挖矿恶意软件开始运行到其初次建立网络连接之间的时间段.

由于目前矿池所使用的主流通信协议为stratum[21],其使用流套接字和JSON消息编码进行通信[22]. 因此,本文考虑将创建socket作为挖矿恶意软件创建网络连接的标志. 从3.2节中API序列分析可知,挖矿恶意软件创建socket的API为socket/WSASocket. 表1统计了75款挖矿恶意软件以及多种类型的75款良性软件开始运行后1分钟内调用socket/WSASocket的情况.

表1　不同软件调用socket/WSASocket的时间统计

socket/WSASocket调用时间	挖矿恶意软件个数	良性软件个数
小于2 s	65	1
2~5 s	2	6
5~8 s	1	1
大于8 s	0	10
未调用	7	57

表1显示绝大多数挖矿恶意软件在开始运行2 s内就调用了socket/WSASocket来创建流套接字以连接矿池. 有7个挖矿恶意软件因为缺少运行环境、矿池域名解析失败等未调用socket/WSASocket. 表1还显示仅有少部分良性软件在1 min内存在socket/WSASocket调用且调用时间多在2 s之后. 基于以上分析,本文以首次调用socket/WSASocket作为挖矿恶意软件建立网络连接的标志,此时挖矿恶意软件还未执行其挖矿操作,因此利用BDP内的API序列可对其进行早期检测.

4　CEDMB

基于第三章的分析,本文提出了CEDMB. CEDMB从BDP内的API序列中提取特征,然后使用分类算法构建检测模型. CEDMB包含如图3所示的三个阶段:数据收集、特征计算、训练和检测. 下面分别进行介绍.

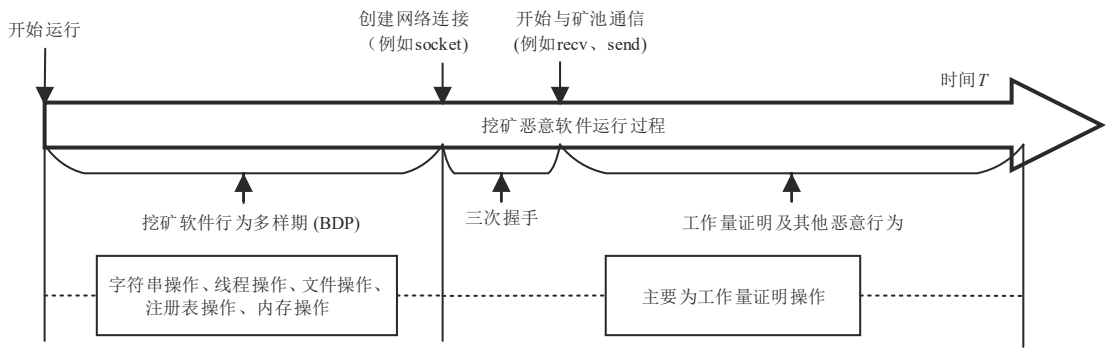


图2　挖矿软件行为多样期示意图

4.1 数据收集

本文发现挖矿恶意软件使用的挖矿算法和选用的进程CPU利用率阈值会影响其产生的API序列：一是部分挖矿恶意软件支持多种挖矿算法且在使用不同的挖矿算法时产生的API序列在每秒API序列熵值、长度以及频繁调用的API等方面存在差异；二是挖矿恶意软件设定不同的进程CPU利用率阈值时运行所产生的API序列有区别. 因此，本文采用如下方式收集用于训练的挖矿恶意软件的API序列：一是分别收集同一软件使用不同挖矿算法所产生的API序列；二是将同一软件的进程CPU利用率阈值分别设置为25%、50%、75%、100%后运行并收集各自的API序列.

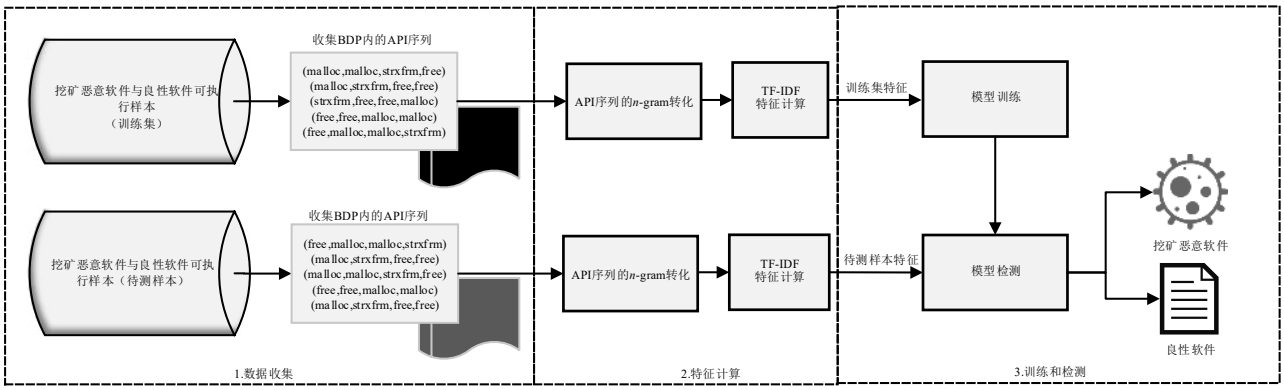


图3 CEDMB框架

我们收集样本在BDP内的API序列时，考虑到一些未能成功实施挖矿的挖矿恶意软件以及未调用socket/WSASocket的良性软件，需要确定一个API序列的数据收集时间上限T. 若样本在0~T内调用了socket/WSASocket，则在初次调用该API的时刻结束API序列的收集；若样本在0~T内未调用socket/WSASocket，则在T时刻结束收集.

4.2 特征计算

在特征计算阶段，CEDMB对上阶段收集的API序列进行特征计算. 从第三章对挖矿恶意软件的API序列分析可知，一些特征API调用能反映挖矿恶意软件的行为特性. 为了提取这些特征API调用，CEDMB使用n-gram. n-gram将长度为n的子序列作为独立特征考虑，可以保留API之间的调用关系. 经过n-gram处理后会存在部分冗余n-gram序列. 因此，CEDMB运用TF-IDF算法对n-gram序列进行筛选及计算特征值. TF-IDF是文本分类领域常用的一种算法，可以衡量一个词的重要程度. 其基本思想是如果一个单词在一篇文章中出现频率高且其他文章中出现少，则认为该词的类别区分能力强. 该算法的特点使其不易遗漏那些能够反映样本特有行为的特征API调用. 计算过程如下：

(1)对样本的每个n-gram序列，可以通过式(2)和式(3)分别计算其TF值和IDF值. 式(2)中TF_{ij}表示在样本j的n-gram序列集合中n-gram$i$的出现频率，$n_{i,j}$表示在样本j的n-gram序列集合中n-gram$i$的出现次数，$\sum_k n_{k,j}$表示样本j的n-gram序列集合中n-gram的总数. 式(3)中IDF_i可以衡量n-gram$i$在所有样本的n-gram序列集合中的稀有程度. $|D|$表示样本总数，$\left|\{j: i \in j | j \in D\}\right|$表示包含$n$-gram$i$的样本$j$的个数.

$$\mathrm{TF}_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

$$\mathrm{IDF}_i = \log_2 \frac{|D|}{\left|\{j: \mathrm{i} \in j | j \in D\}\right|} \tag{3}$$

(2)n-gram$i$在样本j上的TF-IDF值通过式(4)计算得到. 对每个n-gram序列，计算其在所有样本中的TF-IDF值并得到总和. 然后将所有n-gram序列按照其TF-IDF总和从大到小进行排序，选取排名靠前的一定数量的n-gram序列作为特征，并使用其在样本中的TF-IDF值作为该样本的特征向量.

$$\mathrm{TF-IDF}_{ij} = \mathrm{TF}_{ij} \times \mathrm{IDF}_i \tag{4}$$

4.3 模型训练与检测

CEDMB在本阶段将利用上阶段计算出的特征向量，结合机器学习分类算法构建检测模型，用以区分良性软件和挖矿恶意软件. 因为本文的检测模型用于实施挖矿恶意软件与良性软件的二分类，所以需要给每个训练样本的特征向量赋类别标签——良性软件和挖矿恶意软件分别标记为0和1，得到训练集D_{tr}($D_{tr}=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$)，其中$y_i \in \{0,1\}$，$x_i$为特征向量，$m$为样本数量.

在训练步骤，使用训练集结合分类算法构建一个挖矿恶意软件检测模型. 在检测步骤，先获取待检测软件BDP内的API序列并进行特征向量计算，然后由检测模型判别待检测软件的特征向量属于良性软件还是

挖矿恶意软件.

5 实验及结果

5.1 实验环境与样本

样本的运行环境为VMware中创建的Windows 10虚拟机,Intel i7 10700F CPU和4 GB内存.使用API Monitor收集软件的API序列.检测模型所在主机为IntelXeon Gold 5220CPU,251 GB内存,采用Python 3.9并使用scikit-learn 0.24.1库实现.

挖矿恶意软件样本来源于github项目、virusshare(https://virusshare.com)、微步在线云沙箱(https://s.threatbook.cn)、malware traffic analysis(https://malware-traffic-analysis.net)和any.run(https://app.any.run).良性软件样本包括办公软件、安全杀毒、加密软件等类别.良性软件样本来源于各软件的官网、360软件管家.

为验证CEDMB对挖矿恶意软件的检测效果,本文将75款挖矿恶意软件(如4.1节所述每款采集4条API序列)和210款良性软件的在BDP内产生的API序列划分为表2所示的训练集和两个测试集.

表2 实验数据集划分

类别	训练集	测试集1	测试集2	总数
办公软件	20	5	10	35
聊天通讯	16	4	8	28
安全杀毒	12	3	6	21
影音软件	10	2	5	17
游戏	10	2	5	17
压缩软件	10	3	2	15
压力测试	5	2	2	9
加密软件	12	3	7	22
其他应用	25	6	15	46
良性软件API序列数	120	30	60	210
挖矿恶意软件API序列数	120	30	60	210
API序列总数	240	60	120	420

首先随机选取60款挖矿恶意软件的各2条API序列,结合随机选择的120条良性软件API序列,组成包含240条API序列的训练集.接着从训练集中出现过的挖矿恶意软件随机选取15款,将各软件未被选入训练集的2条API序列,与随机选择的30条良性软件API序列组成包含60条API序列的测试集1.之后将在训练集中未出现过的15款挖矿恶意软件的各4条API序列,和另外60条良性软件API序列组成包含120条API序列的测试集2.测试集1和测试集2分别可评估CEDMB对设置了不同进程CPU利用率阈值的已知挖矿恶意软件和未知挖矿恶意软件的检测能力.

5.2 评估指标

本文使用了恶意软件检测领域中常用的评估指标对CEDMB进行评估,包括准确率(Accuracy),精确率(Precision),召回率(Recall)和F1-score(F1)[9].

5.3 实验结果

我们给出了CEDMB使用支持向量机(Support Vector Machine,SVM)、K近邻(K-Nearest Neighbor,KNN)、多层感知机(Multilayer Perceptron,MLP)、随机森林(Random Forest,RF)4种分类算法在测试集1、2上的检测结果,各算法的具体参数如表3所示.我们还给出了CEDMB使用不同数据收集时间上限T、不同n-gram序列长度时的检测结果.

表3 实验中分类算法的参数

算法	参数
SVM	SVC,kernel='rbf'
MLP	solver='lbfgs', alpha=1e−5, hidden_layer_sizes=(8)
KNN	n_neighbors=3
RF	n_estimators=10, criterion='gini'

5.3.1 测试集1的检测结果

CEDMB在2-gram且T为4 s时在测试集1上使用不同分类算法的检测结果如表4所示.表4显示CEDMB使用RF算法时得到的检测结果最佳,此时F1值达到了98.36%.

表4 CEDMB在2-gram和T为4 s时在测试集1上使用不同分类算法得到的检测结果

评估标准	SVM	MLP	KNN	RF
Precision	**0.966 7**	0.937 5	0.857 1	**0.967 7**
Recall	0.966 7	**1**	**1**	**1**
Accuracy	0.966 7	0.966 7	0.916 7	**0.983 3**
F1	0.966 7	0.967 7	0.923 1	**0.983 6**

为评估CEDMB使用不同特征计算方法的检测结果,我们测试了CEDMB使用4种不同特征计算方法时所得到的结果,该实验中T为4s,分类算法为RF,结果如表5所示.其中,2IG指代CEDMB使用2-gram和信息增益计算特征;BoW指代CEDMB使用词袋模型计算特征;TIW指代CEDMB使用word2vec生成API的词向量,再使用API的TF-IDF值对API词向量进行加权平均来得到特征向量;2T指代CEDMB使用2-gram和TF-IDF计算特征.特征计算时间为全部测试API序列转换为特征向量的时间.由表5可知CEDMB使用2T时获得的F1值和所需的特征计算时间优于其使用其他特征计算方法.

图4对比了CEDMB在不同T值和n-gram长度时的检测结果.当T在{2,4,6,8}内取值时,图4显示CEDMB使用不同n-gram以及不同分类算法时大多在T为4 s时获得最佳的F1值.依据3.3节给出的挖矿恶意软件调用socket/WSASocket的情况,即所分析的绝大多

表5 CEDMB使用T=4 s和RF时在测试集1上使用不同特征计算方法得到的检测结果

方法	Accuracy	F1	特征计算时间/s
2IG	**0.983 3**	0.983 1	2.616 2
BoW	0.966 7	0.967 7	2.654 8
TIW	0.966 7	0.966 7	2.668 1
2T	**0.983 3**	**0.983 6**	**2.526 4**

数挖矿恶意软件的调用时间在开始运行后2 s内,少部分在2 s到5 s. 因此,以T为4 s收集的API序列囊括了绝大多数挖矿恶意软件样本在建立网络连接前的行为,而这些行为与良性软件存在明显区别. 绝大多数挖矿恶意软件在运行5 s后主要执行工作量证明行为,而这个时间点良性软件也开始越来越多的执行内存、注册表操作,这导致CEDMB在T为6 s和8 s时得到的F1值,相比于其在T为4 s时得到的F1值未进一步提升,甚至还略有下降.

当n-gram的n在{2, 3, 4}内取值时,图4显示CEDMB在n为2和3时使用不同分类算法获得的F1值总体上略优于n为4时获得的F1值. 这主要是因为挖矿恶意软件的特征API调用的长度大多为2和3. 由于更短的n-gram长度对应更高的特征计算效率,因此在后续实验中CEDMB均使用2-gram.

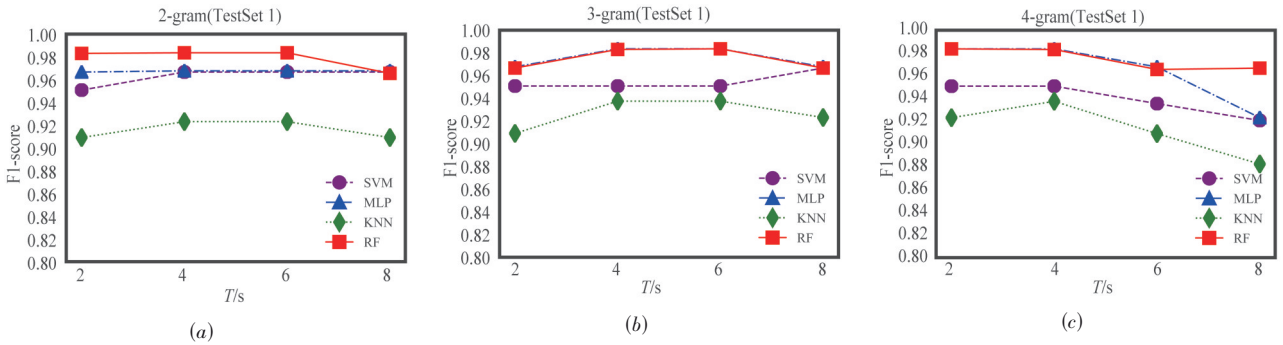


图4 CEDMB在不同T值和n-gram下使用4种分类算法得到的F1值

CEDMB与其他挖矿恶意软件检测方法在测试集1上的检测结果如表6所示,其中检测时间为单个方法所需的数据收集时间、全部测试API序列的特征计算时间及分类时间的总和,"—"表示无此项. 表6显示CEDMB和文献[14]的方法在测试集1上得到的Accuracy和F1值高于文献[15]和文献[7]的方法. 在检测时间上,CEDMB在BDP内收集数据并使用TF-IDF进行特征计算,所需检测时间在几种方法中最少. 文献[14]的方法所需的数据收集时间和特征计算时间均长于CEDMB. 文献[15]采用动静态特征结合的方法但未说明其动态行为数据收集时间(对比实验中设置为1分钟),所需的特征计算时间远大于另外几种方法. 文献[7]也未给出其数据收集时间,当该文献方法的数据收集时间设置为与CEDMB相同的4 s时,其采用深度学习方法所需的检测时间较CEDMB略长.

5.3.2 测试集2的检测结果

CEDMB使用2-gram和RF算法在测试集2上得到的检测结果如表7所示. 当T在{2,4,6,8}内取值时,表7显示CEDMB在T为4 s时获得的检测结果最佳.

表6 不同检测方法在测试集1上的检测结果

方法	Accuracy	F1	特征计算时间/s	检测时间/s
Karn[14]	**0.983 3**	**0.983 6**	25.414 7	87.219 2
Berecz[15]	0.966 7	0.967 7	139.12	199.122
Darabian[7]	0.966 7	0.967 7	—	7.622
CEDMB	**0.983 3**	**0.983 6**	**2.526 4**	**6.552 9**

表7 CEDMB在不同T值下使用2-gram和RF时在测试集2上得到的检测结果

T	2 s	4 s	6 s	8 s
Precision	**1**	**1**	0.982 5	0.949 2
Recall	0.916 7	**0.933 3**	**0.933 3**	**0.933 3**
Accuracy	0.958 3	**0.966 7**	0.958 3	0.941 7
F1	0.956 5	**0.965 5**	0.957 3	0.941 2

CEDMB使用不同特征计算方法在测试集2上得到的检测结果如表8所示. 表8显示在4种特征计算方法中,CEDMB使用2T得到的Accuracy和F1值最佳且所需的特征计算时间最少.

表9给出了CEDMB与其他挖矿恶意软件检测方法在测试集2上的检测结果. 如表9所示,CEDMB在测试集2上得到的Accuracy和F1值优于其他文献的方法,

表8 CEDMB使用T=4 s和RF时在测试集2上使用不同特征计算方法得到的检测结果

方法	Accuracy	F1	特征计算时间/s
2IG	0.958 3	0.957 3	8.242
BoW	0.95	0.949 2	6.185 6
TIW	0.95	0.948 3	8.753 7
2T	**0.966 7**	**0.965 5**	**5.445 5**

所需的检测时间也最少. CEDMB能够对测试集2中的未知挖矿恶意软件样本具有很高的检测精度,一是因为CEDMB关注的是挖矿恶意软件在BDP内的行为,这些行为主要为以隐蔽、持久化、逃逸检测为目的的相关行为,属于挖矿恶意软件在其运行初期的必要行为;二是因为CEDMB使用的特征计算方法能较好地保留BDP内的特征API调用;三是因为CEDMB使用的RF算法是一种包含多个决策树的分类器,其输出的类别由各个决策树输出类别的众数决定而不容易产生过拟合.

表9　不同检测方法在测试集2上的检测结果

方法	Accuracy	F1	特征计算时间/s	检测时间/s
Karn[14]	0.941 7	0.941 2	53.291 6	118.287 9
Berecz[15]	0.866 7	0.857 1	249.527 8	309.529 7
Darabian[7]	0.941 7	0.941 2	—	10.574 9
CEDMB	**0.966 7**	**0.965 5**	**5.445 5**	**9.476 5**

6　结论

本文针对当前动态检测方法检测及时性不佳的问题,提出了一个对及时检测挖矿恶意软件具有重要意义的概念——BDP,构建了挖矿恶意软件早期检测方法CEDMB. 实验结果表明,CEDMB只需收集至多4 s的API序列、使用2-gram和RF算法即可分别以98.36%、96.55%的F1值检测数据集中已知和未知的挖矿恶意软件样本. 相比其他现有的挖矿恶意软件动态检测方法,CEDMB在检测及时性方面优势明显,这在降低挖矿恶意软件对受害主机造成的损害方面具有重要意义. 本文后续将继续探索如何在更短的时间内实现对挖矿恶意软件高准确率的检测.

参考文献

[1] BIJMANS H L J, BOOIJ T M, DOERR C. Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at Internet scale[C]//Proceedings of the 28th USENIX Conference on Security Symposium. New York: ACM, 2019: 1627-1644.

[2] TEKINER E, ACAR A, ULUAGAC A S, et al. SoK: cryptojacking malware[C]//2021 IEEE European Symposium on Security and Privacy (EuroS&P). Piscataway: IEEE, 2021: 120-139.

[3] 郭春, 罗迪, 申国伟,等.一种基于诱导机制的间谍软件检测方法[J]. 电子学报, 2022, 50(4): 1014-1024.
GUO C, LUO D, SHEN G W, et al. A spyware detection method based on inducement mechanism[J]. Acta Electronica Sinica, 2022, 50(4): 1014-1024. (in Chinese)

[4] Nttsecurity. 2021 global threat intelligence report[EB/OL]. (2021-08-19) [2022-08-05]. https://services. global. ntt/en-gb/insights/2021-global-threat-intelligence-report.

[5] CAVNAR W B, TRENKLE J M. N-gram-based text categorization[C]//Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994:14.

[6] ZHANG W, YOSHIDA T, TANG X J. A comparative study of TF*IDF, LSI and multi-words for text classification[J]. Expert Systems with Applications, 2011, 38(3): 2758-2765.

[7] DARABIAN H, HOMAYOUNOOT S, DEHGHANTANHA A, et al. Detecting cryptomining malware: A deep learning approach for static and dynamic analysis[J]. Journal of Grid Computing, 2020, 18(2): 293-303.

[8] YAZDINEJAD A, HADDADPAJOUH H, DEHGHANTANHA A, et al. Cryptocurrency malware hunting: A deep recurrent neural network approach[J]. Applied Soft Computing, 2020, 96: 106630.

[9] NASEEM F, ARIS A, BABUN L, et al. MINOS: A lightweight real-time cryptojacking detection system[C]//Proceedings 2021 Network and Distributed System Security Symposium. Reston, VA: Internet Society, 2021: 21-25.

[10] 郑锐, 汪秋云, 林卓庞,等. 一种基于威胁情报层次特征集成的挖矿恶意软件检测方法[J]. 电子学报, 2022, 50(11): 2707-2715.
ZHENG R, WANG Q Y, LIN Z P, et al. Cryptojacking malware hunting: A method based on ensemble learning of hierarchical threat intelligence feature[J]. Acta Electronica Sinica, 2022, 50(11): 2707-2715. (in Chinese)

[11] CHOUDHARY S P, VIDYARTHI M D. A simple method for detection of metamorphic malware using dynamic analysis and text mining[J]. Procedia Computer Science, 2015, 54: 265-270.

[12] NING R, WANG C, XIN C S, et al. CapJack: Capture in-browser crypto-jacking by deep capsule network through behavioral analysis[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. Piscataway: IEEE, 2019: 1873-1881.

[13] MANI G, PASUMARTI V, BHARGAVA B, et al. DeCrypto pro: Deep learning based cryptomining malware detection using performance counters[C]//2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). Piscataway: IEEE, 2020: 109-118.

[14] KARN R R, KUDVA P, HUANG H, et al. Cryptomining detection in container clouds using system calls and ex-

plainable machine learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(3): 674-691.
[15] BERECZ G, CZIBULA I G. Hunting traits for cryptojackers[C]//Proceedings of the 16th International Joint Conference on e-Business and Telecommunications. Setubal: SCITEPRESS, 2019: 386-393.
[16] PASTOR A, MOZO A, VAKARUK S, et al. Detection of encrypted cryptomining malware connections with machine and deep learning[J]. IEEE Access, 2020, 8: 158036-158055.
[17] ZHANG S Z, WANG Z L, YANG J H, et al. MineHunter: A practical cryptomining traffic detection algorithm based on time series tracking[C]//ACSAC'21: Annual Computer Security Applications Conference. New York: ACM, 2021: 1051-1063.
[18] SUN P F, LYU M D, LI H, et al. An early stage convolutional feature extracting method using for mining traffic detection[J]. Computer Communications, 2022, 193: 346-354.
[19] LI Z, LIU W J, CHEN H B, et al. Robbery on DevOps: Understanding and mitigating illicit cryptomining on continuous integration service platforms[C]//2022 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2022: 2397-2412.
[20] PEKTAŞ A, ACARMAN T. Malware classification based on API calls and behaviour analysis[J]. IET Information Security, 2018, 12(2): 107-117.
[21] Systems Braiins. Stratum V2 | The next generation protocol for pooled mining[EB/OL]. (2020-01-23) [2022-08-05]. https://stratump-rotocol.org.
[22] RUSSO M, ŠRNDIĆ N, LASKOV P. Detection of illicit cryptomining using network metadata[J]. EURASIP Journal on Information Security, 2021, 2021(1): 1-20.

作者简介

曹传博 男,1998年生,湖北孝感人. 贵州大学计算机科学与技术学院硕士研究生. 主要研究方向为计算机网络与信息安全.
E-mail: cbcao3842@163.com

郭 春(通讯作者) 男,1986年生,贵州贵阳人. 博士,贵州大学计算机科学与技术学院教授. 主要研究领域为恶意代码检测、入侵检测.
E-mail: gc_gzedu@163.com