面向行为多样期的挖矿恶意软件早期检测方法

2024-09-29 18:12:53

# 摘要

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）。

该方法可以在软件开始运行后10s内以96.55%的F1-score值判别其是良性软件还是挖矿恶意软件。

# 引言

传统的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。一些诸如代码混淆、更换域名、限制进程 CPU 利用率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，该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，该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 和 TFIDF （Term Frequency-Inverse Document Frequency）生成特征向量，通过分类算法建立检测模型。

该文的主要贡献如下：

* 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。
* 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。
* 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。

# 相关工作

## 静态检测方法

* 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。
* 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。
* 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。

总之，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。

## 动态检测方法

* 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。（需要等待系统性能受到影响后才能检测出挖矿恶意软件）
* 使用 API 序列结合深度学习实现检测。
* 使用 n-gram 模型从 API 序列中提取特征

总之，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。

## 流量监测方法

* 使用 51 个通信行为统计特征的挖矿加密流量检测，对来自 xmr-stak 和 xmrig 的流量进行检测。
* 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。
* 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。

总之，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。

# 挖矿恶意软件行为分析

## 挖矿恶意软件攻击过程

挖矿恶意软件会在运行初期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。

## 挖矿恶意软件API序列分析

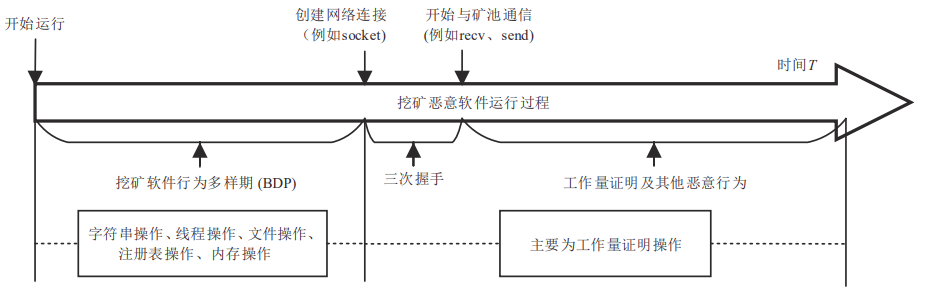
该文收集了覆盖多种加密货币类型的75款挖矿恶意软件，在 VMware 创建的 Windows10 虚拟机中收集各样本在其默认的进程 CPU 利用率阈值下运行1分钟的 API 序列，从 API 调用的角度对这些 API 序列进行分析。该文将对应特定行为且相对固定的 API 调用称为特征 API 调用。

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，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。

## 挖矿软件行为多样期

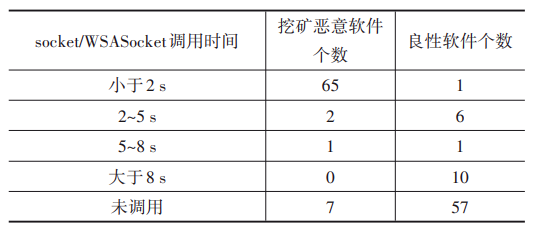
挖矿恶意软件的每秒 API 序列熵值在运行初期时较大，然后快速下降并出现波动，一定时间之后则较为稳定。良性软件各时刻的每秒 API 序列熵值均在波动，与挖矿恶意软件区别较大。

该文以网络连接建立与否来判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，该文提出 BDP 的概念，其在挖矿恶意软件运行过程中所处时间段如下图所示：



挖矿软件行为多样期示意图

下表统计了 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：

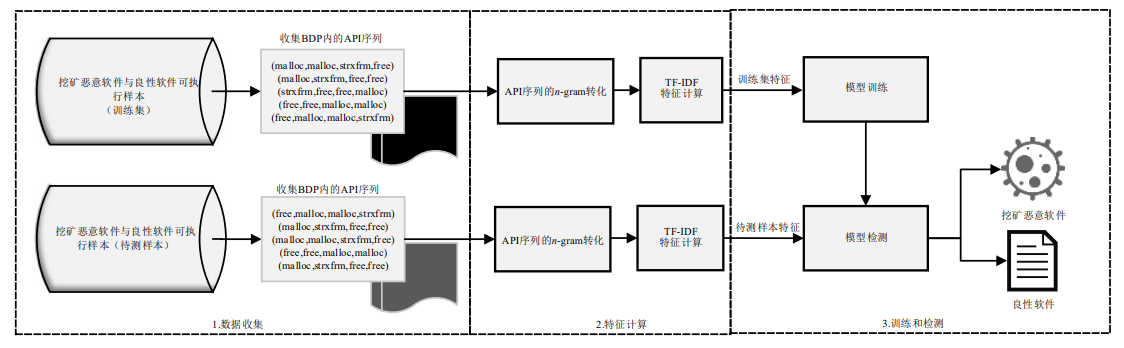


不同软件调用socket/WSASocket的时间统计

上表显示绝大多数挖矿恶意软件在开始运行 2s 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 。 上表还显示仅有少部分良性软件在 1min 内存在 socket/WSASocket 调用且调用时间多在 2s 之后 。 基于以上分析，该文以首次调用 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。

# CEDMB

基于上一章的分析，该文提出了 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。框架如下图：



CEDMB框架

## 数据收集

该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，该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。

收集 API 序列的时间区间为程序运行到 T （T为自定阈值）。

## 特征计算

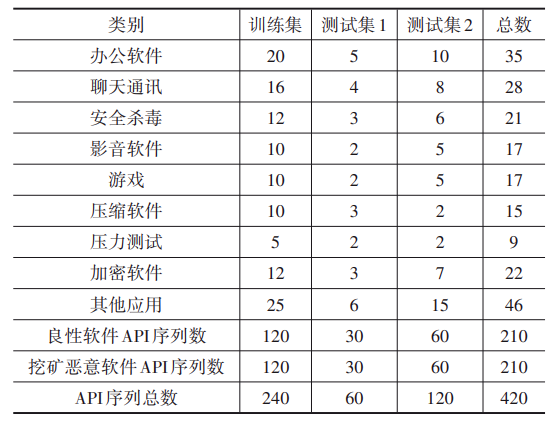
一些特征 API 调用能反映挖矿恶意软件的行为特性，为了提取这些特征 API 调用， CEDMB 使用 n-gram 。 n-gram 将长度为 n 的子序列作为独立特征考虑，可以保留 API 之间的调用关系。经过 n-gram 处理后会存在部分冗余 n-gram 序列。因此， CEDMB 运用 TFIDF 算法对 n-gram 序列进行筛选及计算特征值。

## 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。

# 实验及结果

## 实验环境与样本



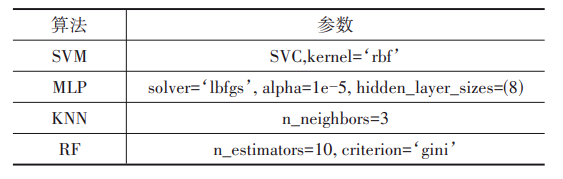
实验数据集划分

## 评估指标

该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）。

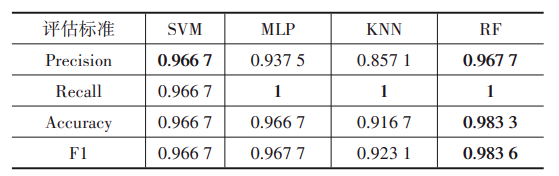
## 实验结果

模型参数细节如下：



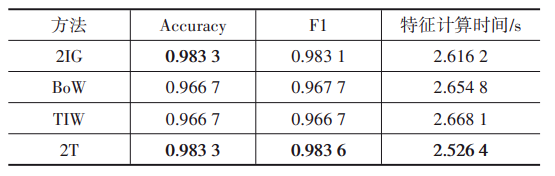
实验中分类算法的参数

### 测试集1的检测结果



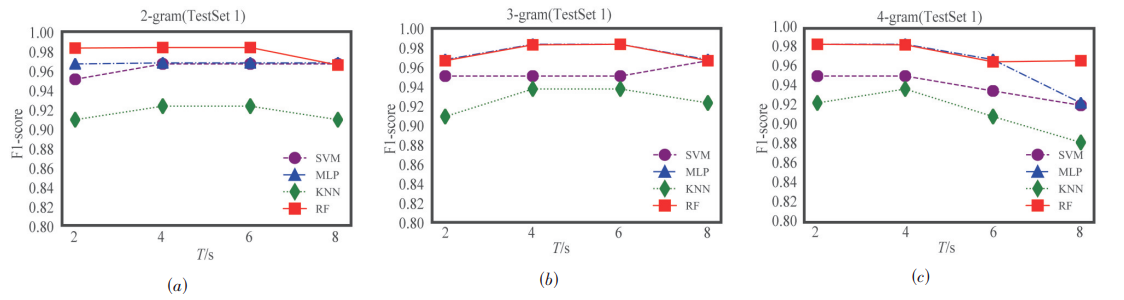
CEDMB在2-gram和T为4s时在测试集1上使用不同分类算法得到的检测结果

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%。



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

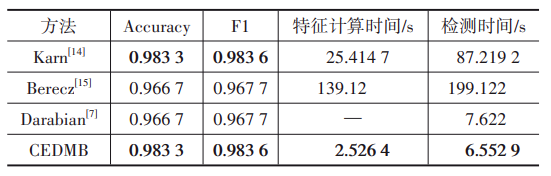
2IG 指代 CEDMB 使用 2-gram 和信息增益计算特征； BoW 指代 CEDMB 使用词袋模型计算特征； TIW 指代 CEDMB 使用 word2vec 生成 API 的词向量，再使用API的 TF-IDF 值对 API 词向量进行加权平均来得到特征向量； 2T 指代 CEDMB 使用 2-gram 和 TF-IDF 计算特征。特征计算时间为全部测试 API 序列转换为特征向量的时间。



CEDMB在不同T值和n-gram下使用4种分类算法得到的F1值

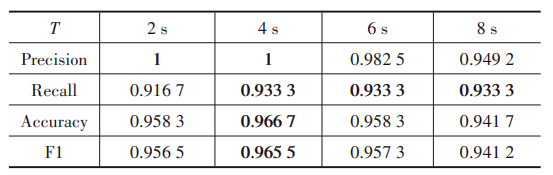
CEDMB 使用不同 n-gram 以及不同分类算法时大多在T为4s时获得最佳的 F1 值。大多数挖矿恶意软件的调用时间在开始运行后2s内，少部分在2s到5s。因此，以T为4s收集的 API 序列囊括了绝大多数挖矿恶意软件样本在建立网络连接前的行为，而这些行为与良性软件存在明显区别。

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。

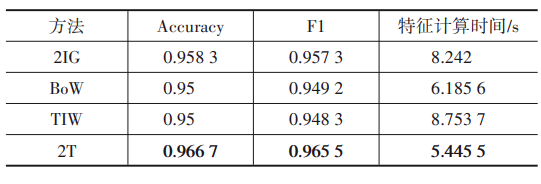


不同检测方法在测试集1上的检测结果

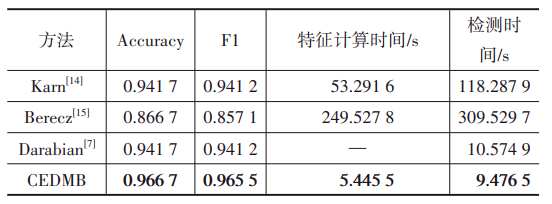
### 测试集2的检测结果



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



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



不同检测方法在测试集2上的检测结果

# 评价

## 创新点

* 使用了不同的特征计算方法，并且证明2T方法效果最好。

## 新点子

* 提取出来的API短序列应该是只包含API名，如果提取出来的API短序列还包含API参数、API调用者和API调用时间等信息，效果会不会更好？