文章编号:1671-4229(2022)02-0060-07

基于数据包特征的加密流量分类

杨彦召[1],丁 杰[2],仇 晶[2],张光华[3*]
(1. 中汽智创科技有限公司,江苏 南京 211100;2. 广州大学 网络空间先进技术研究院,广东 广州 510006;
3. 河北科技大学 信息科学与工程学院,河北 石家庄 050091)

摘 要:随着加密技术在网络应用中的广泛应用,如何在不侵犯用户隐私的情况下对加密流量进行分类成为新的挑战。文章提出了一种基于数据包的加密流量分类方法,这种方法不仅安全,而且可以有效防止恶意攻击流量。在流量传输过程中,每个数据包的任务是不同的。有些数据包负责维护连接,有些负责数据交互。以往的研究忽略了数据包行为对流量分类的影响。基于数据包的方法旨在通过聚类算法来区分正常和恶意数据包对流量分类的影响,以识别加密的恶意流量。该方法使用公共流量数据集和实验室收集的流量数据集进行验证,并与其他2种方法进行比较,证明了基于数据包的加密流量分类方法的有效性。

关键词:恶意软件;加密恶意流量检测;数据包;机器学习
中图分类号:TP 183 **文献标志码**:A

Encrypted traffic classification based on packet characteristics

YANG Yan-zhao[1], *DING Jie*[2], *QIU Jing*[2], *ZHANG Guang-hua*[3*]
(1. China Automotive Innovation Corporation, Nanjing 211100, China;
2. Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China;
3. School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050091, China)

Abstract: With the widespread application of encryption technology in network applications, how to classify encrypted traffic without infringing on user privacy has become a new challenge. This study, proposes a packet-based encryption traffic classification method. This method is not only safe but also effective in preventing malicious attack traffic. In the process of traffic transmission, the task of each data packet is different. Some data packets are responsible for maintaining the connection and some are responsible for data interaction. Previous research has neglected the influence of packet behavior on traffic classification. The method based on the packet aims to distinguish the impact of normal and malicious data packets on traffic classification through a clustering algorithm to identify encrypted malicious traffic. The method uses public flow data sets and laboratory-collected flow data sets for verification. The comparison with the other two methods proves the effectiveness of the encryption traffic classification method based on data packets.

Key words: malware; encrypt malicious traffic detection; packet; machine learning

流量分类是一种将网络流量划分为相应类别的任务,在网络入侵检测系统(NIDS)等许多软件程序中具有至关重要的应用。它可用于识别网络中恶意软件产生的流量,从而帮助防火墙识别恶意连接入侵。恶意软件可以定义为旨在破坏计算机系统的程序,它是当今信息安全的最大威胁[1]。流量分类已经被研究了20年,早期网络传输的数据包多为明文和基于端口的数据包,因此,深度数据包检测(DPI)技术被广泛采用。加密流量

基金项目:国家自然科学基金资助项目(U1636215,U1803263);国家重点研发计划资助项目(2018YFB1800702);广东省重点研发计划资助项目(2019B010136003)
作者简介:杨彦召(1985—),男,工程师. E-mail:yangyanzhao@t3caic.com
*通信作者. E-mail:xian_software@163.com
引文格式:杨彦召,丁杰,仇晶,等. 基于数据包特征的加密流量分类[J]. 广州大学学报(自然科学版),2022,21(2):60-66.

是一种对原始数据报文加密后,在网络上进行信息传输的技术。随着网络的发展,加密流量方法受到大范围推广,过去依赖明文内容的流量分类方法使用率不断下降。

加密旨在保护网络通信的安全性和隐蔽性,但这种隐蔽性往往被攻击者用于隐藏和部署恶意代码、远程命令控制或者造成数据泄露。因此,如何在不解密数据包的情况下对加密流量进行分类成为网络空间安全研究的主要问题。

网络上主要采用 TLS 协议对流量加密。TLS 协议保障了互联网安全通信,其在 2 个通信的应用程序之间进行握手,从而建立起可靠的加密通信信道。由于传统基于端口的方法以及深度包检测技术对流量分类时依赖明文数据,因此,这些方法在加密流量面前失去作用。这也导致了利用机器学习对加密流量进行分类的方法逐渐引起了大多数学者的关注。机器学习算法已被证明是处理加密流量分类任务最合适的方法。它以时间序列和统计数据为特征,既可以对未加密流量进行分类,也可以通过对加密流量的加密协议进行分析以实现流量分类的目的[2]。

思科提供了一种基于对 TLS 握手元数据和 2 个上下文流深入分析的方法来识别加密流量中的威胁[3]。TLS 协议在握手协议阶段,双方发送的数据仍然是以明文的形式进行传输,直到加密通信信道建立为止。鉴于此,本文的目标是设计一种流量分类方法,避免解密数据包的隐私信息,充分利用网络流量的元数据。正常流量的通信行为与恶意软件的通信行为存在很大差异。这种差异可以反映在与 TLS、证书或流元数据相关的几个参数中[4],是能够实现高精度的流量分类系统的理论基础。

传统的机器学习模型选取的特征基于流或者会话。流是指具有相同源 IP 地址和目标 IP 地址的网络流量。与之相对,会话是指具有相同源 IP 地址和目标 IP 地址但可以相互交换的双向网络流量。网络流量中解析的特征包括数据包信息(包长度)、TLS 信息、DNS 信息或可用于流量分类的 HTTP 信息。传统的机器学习算法通过对流或会话上的数据包进行一系列的计算,得到上下行数据包数和上下行字节数的统计数据。

通常来说,客户端和服务器之间的通信流量中只有部分数据包涉及数据交互。也就是说,即使是恶意流量,也不是所有的数据包行为都是恶意的。而正常的流量偶尔也会产生行为异常的数据包。本研究认为流量中数据包的正常行为和恶意行为的比例也是对加密流量进行分类的重要依据,之前的研究中都忽略了流量包行为对加密流量分类效果的影响。本研究提出的基于数据包的流量分类方法可以发现这个缺陷并解决这个问题。因此,本研究主要的贡献总结如下:

(1)提出了一种以数据包为分类特征的加密流量分类方法,使用流或者会话作为训练特征将忽略数据包行为对分类的影响。

(2)支持多版本 TLS 协议。随着网络的发展,加密协议也在不断迭代,如 TLS1.3 协议自 2018 年提出以来得到广泛应用。考虑到加密协议的升级,提出了一种新的特征选择方法。

(3)通过与现有的加密流量检测模型对比,实验结果表明,提出的模型具有良好的流量分类能力,分类准确率达到 99.96%。

本文的其余部分安排如下:第二节介绍了相关工作;第三节描述了使用的数据特征和算法;第四节概述了使用的数据集以及实验结果;第五节给出了结论和下一步工作计划。

1　相关工作

加密流量检测是流量分类研究领域的一个重要研究方向,同时也是网络安全领域的基础组成部分。网络流量的爆发式增长使安全监控系统承受着巨大的压力。传统的基于 DPI 技术的流量识别方法可以准确识别未加密的电信流量,但无法有效识别加密流量。Sen 等[5]分析了基于端口和深度包检测方法的局限性,并强调了统计的优点。随着 TLS 加密协议的出现和应用,这些传统方法逐渐失去了原有的优势,识别准确率急剧下降。不过得益于此,基于端口和深度包检测方法的局限性促进了流量分类研究的发展,并推动了使用机器学习方法来处理加密流量的趋势。

机器学习算法相比传统流量分类方法具有更多的优点。比如,机器学习算法能够处理加密流量分类,同时具备了很高的准确率。除此之外,机器学习算法能够适应复杂多变的网络环境,识别未知的样本类别。因此,它可以对不断升级的加密协议和新兴的网络应用进行分类[6]。基于机器学习的加密流量检测方法通过对数据流元数据的统计分析,构建加密流量的统计属性组合作为指纹,对加密流量进行分类识别。现有的研究主要分为基于规则的加密流量检测和基于机器学习的加密流量检测方法。

1.1　基于规则的流量检测

基于规则的加密流量检测主要利用加密流量的数据包字段或字段组合特征(如排序或固定模式)作为指纹进行规则匹配。Kim 等[7]提出了一种从加密的流量有效载荷数据中自动生成服务签名的新方法。使用证

书交换过程中的证书颁发信息字段对服务进行签名,并构建证书、会话ID和IP地址应关系列表,以匹配流量的类别。

建立映射表是基于规则的加密流量检测的常用方法。Shbair等[8]将服务器名称指示(SNI)与IP对应的域名信息进行比较,依靠可信的DNS服务来验证真实目标服务器与声明的SNI值的一致性,从而监控HTTPs流量。Husák等[9]通过分析TLS握手数据中支持的密码套件列表和来自HTTP头的用户代理来构建密码套件列表和字典,以识别加密通信的客户端。Papadogiannake等[10]提出了一种模式语言,通过定期匹配固定的加密模式(例如相关数据包的出现频率或数据包的位置)来识别加密流量的类型。

一般来说,基于规则的方法需要对字段特征进行人工过滤,匹配提取的规则对加密流量进行分类。这种方法具有轻量快速且容易构建的优点,但缺点在于需要人工筛选特征字段,仅可以对已知类别的流量构造映射表进行对应匹配,容易被攻击者采用数据包相关字段拼接或伪造的方法绕过,具有很高的误报率。

1.2 基于机器学习的加密流量检测

基于机器学习的加密流量检测方法(对数据流元数据进行统计分析)与基于规则的流量分类方法不同,基于机器学习算法通过构造加密流量的特征矩阵,搭建机器学习模型进行分类。机器学习算法构建的特征矩阵各不相同。Bilge等[11]使用NetFlow和外部信誉评分对僵尸网络流量进行分类。他们的检测模型也可以应用于加密恶意网络流量的检测,但该模型没有使用与TLS数据相关的特征。

机器学习算法采用的特征更多来源于网络流量的原始特征。Panchenkoa等[12]基于数据包大小,通过统计和计算变换衍生后的特征,对加密攻击流量执行网站指纹识别。Wurzinger等[13]通过分析数据包的大小和数据包发送到达时间来检测恶意流量。流量包数据流统计也可以作为识别加密流量恶意软件类别特征的模型。Mcgrew等[14]使用入站字节数、出站字节数、入站数据包数、出站数据包数、源端口号和目的端口号以及数据流的持续时间作为特征。

除了流(Flow),会话(Session)也是一个常见的选择。流和会话都是一个流量单位,流量单位可以分为五元组(源IP、源端口、目的IP、目的端口和传输层协议)。不同之处在于流是一个方向,会话的源IP和目标IP是可以互相交换的[15]。Anderson等[16]提出了一种通过检测与加密流量相关的DNS和HTTP中的关键信息来判断加密恶意流量的方法,使用的特征数据包括完整的TLS握手包和与TLS握手包相同的服务端和客户端5分钟窗口内的DNS和HTTP信息。

Prasse等[17]从HTTPS日志中提取基于数据流的特征,然后对其进行扩展,生成的特征包括端口热编码值、流持续时长、包时间间隔以及发送和接收的数据包字节数。但是在最新版本的TLS1.3协议中,证书握手数据已经被加密,无法获取证书信息。

除了传统机器学习,很多研究人员也将深度学习模型对特征的自适应学习能力研究逐步迁移到加密流量检测研究中。Wang等[18]将PCAP格式的流量文件进行特征筛选融合,裁剪头部前784字节大小的数据作为数字矩阵,构建CNN模型从而识别流量的类别。

Razaei等[19]使用了基于半监督的一维卷积神经网络来分类谷歌的5个应用,这个模型使用的整个流的数据特征中包含了大量的无标签数据,学习到的权重传递给了一个新的模型,伴随着少部分的标记数据一起作为训练数据重新被训练来完成对应用软件的分类。这篇论文展示了使用标记的时间序列作为特征的可能性,而不是以往采用前N个数据包作为训练样本的方法,优点在于分析高带宽的网络应用时具有更高可用性。

基于机器学习的加密流量检测方法可以根据不同的应用场景调整相应的特征结构,但这种方法依赖于专业的人员来分析和构建特征矩阵,非常依赖知识和经验。基于深度学习方法的输入不需要人工构建,可以自动化提取特征,但存在考虑信息不全面,仅能针对单一任务分类的缺点。

上面提到的论文大部分都将数据包作为流量分类的一个特征,但是他们并没有进一步分析数据包行为是否是恶意的,从而忽略了数据包行为对于流量分类的影响。此外,TLS 1.3加密协议的提出和应用是网络空间安全和性能方面的重大进步。虽然主流浏览器还没有默认开启,但是使用TLS1.3协议对传输消息进行加密已经成为一种趋势。

2 基于数据包的加密流量分类系统

基于以上原因,本文提出了一种基于数据包粒度的流量分类方法,认为机器学习常用的特征可以分为时空特征、头部特征、负载特征和统计特征。是否将数据包的性质作为流量分类的特征是基于数据包方法的重要依据,也是本研究的方法与以往研究的最大区别。下面介绍流量分类的特点和模型的组成部分。

2.1 特征表示

目前的流量分类方法至少使用时空特征、头部特征、负载特征和统计特征中的一个或多个类别。

(1)时空特征一般指网络流量传输过程中正常发送的数据包时间和空间属性。数据包的时间属性有数据包发送时间和包间时延等。数据包的空间特征包括数据包长度、数据包发送方向和数据包个数等[20]。

(2)头部特征,包括流量五元组、DNS 和 HTTP 信息。DNS 包括 DNS 域名、返回码、DNS 地址和 TTL 生存期。除此之外,还可以从 DNS 中提取其他特征,比如网站的受访问欢迎度排名(如网站 Alexa 排名)以及网站域名的长度以及字符分布规律(如域名的高斯分布)。HTTP 是使用最广泛的协议,常用于 web 浏览器和 SMTP 邮件服务中,能够提取出的特征包括 HTTP 协议类型、请求方式、状态码以及 Content-Type 字段等。

(3)负载特征,是指封装在流数据上的内容,例如加密协议。在建立安全的加密通信信道之前,客户端和服务端必须交换数据报文以确认对方身份信息,这一过程通常被称为加密协议的握手阶段。在 TLS 协议中,客户端和服务端需要交换彼此支持的密码套件从而选择一种合适的加密算法、加密数据报文。为了认证客户端和服务端的身份,服务端会发送证书给客户端,验证通信双方的身份。值得一提的是,目前对加密流量的研究大多基于 TLS1.2 协议,而 TLS1.3 已经开始流行。相比于 TLS1.2 协议而言,TLS1.3 协议在握手过程发送的报文以及握手次数更少,也给加密流量分类任务带来更多挑战。TLS1.2 和 TLS1.3 协议的区别见图 1。

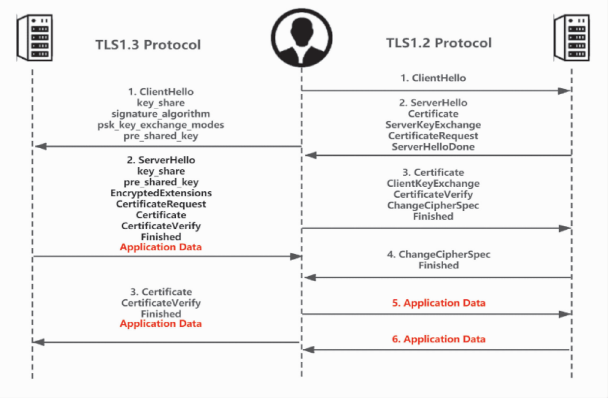

图 1　TLS1.2 和 TLS1.3 协议握手过程对比

Fig. 1　Comparison of protcol hand TLS1.2 and TLS1.3

(4)统计特征,可以从流量中获取。单个数据包具有 3 个属性:字节数、传输方向和包间延迟。根据数据包本身的属性,可以计算得到平均包长、最大包长、平均包间时延等属性。根据客户端和服务端发送数据的来源可以简单定义流量的方向,上行流量为客户端发送给服务端的流量,下行流量为客户端接收服务端的流量。由此,可以进一步计算出上下行数据包数的比值和上下行字节数的比值。统计特征可以从数值上看出正常流量和恶意流量的区别,这也是对加密流量进行分类的一个重要特征。但是为了获得统计特征,分类器必须在一个流或会话中获得许多数据包,因此,它只能用于离线分类[21]。

如前所述,现有研究主要使用以上 4 种特征,却忽略了数据包的行为特征对流量分类的影响。本文主张数据包行为有正常和恶意 2 类,涉及数据窃取或网络攻击的数据包属于恶意行为数据包。下面介绍如何区分数据包行为的类别并将行为特征输入到模型当中。

2.2　模型架构

上述介绍了负载特征,在加密协议的握手阶段,客户端和服务器需要以明文方式协商相关的加密参数。在这个阶段,可以得到很多客户端和服务端相关的数据。TLS 握手阶段的协议和加密流量的特征可以在一定程度上描述应用程序的行为,例如恶意软件倾向于使用低版本的加密协议和弱密码套件。流持续时间、时间间隔、总字节数等数据流统计特征可以描述客户端和服务器的网络行为。网络行为分析对检测新的恶意软件和零日威胁特别有效[3]。

为了提取有效特征,需要处理原始数据包文件。在处理 TLS 加密流量数据时,需对超时、重传和失序的数据包进行处理。还有一些流量数据缺少关键功能,例如服务器证书或提供的密码套件列表。最后需要过滤一些错误的数据包,但不能设置太多的强制特征,以免从数据集中删除过多的流,使数据集容易过拟合。

此外,还需要对正常流量和恶意流量的 IP 进行脱敏处理,以防止数字对分类的影响。对于源端口号和目的端口号,保留原始端口号是因为恶意流量的偏好,99% 的恶意流量更喜欢使用 443 端口[3]。同时,一条流或者会话中,PCAP 文件中的流量包含的数据包数量和连接时间不同,这会导致特征提取困难以及造成训练样本的偏差。所以本研究选择使用流量中的前 20 个数据包来保证训练数据集的一致性,不足 20 个数据包的流或者会话会被剔除。

得到可靠的数据包来源后,需要判别出数据包是正常行为的数据包还是恶意行为的数据包。因为数据集标签是建立在流量上的,为了获取数据包的行为类别需要对其进行聚类。一般认为正常流量的大多数数据包行为是正常的,而恶意流量的恶意行为数据包的比例要高得多。统计一条流上正常数据包和恶意数据包个数之后,用流的标签验证聚类方法的有效性。但是传统的数据集标签是针对整个流或会话的,也就是说,无法直接判断数据包是正常数据包还是恶意数据包。为了得到数据包行为的标签,会使用到聚类算法。在得到所有需要的特征之后,就可以将提取的特征输入到模型当中,模型的结构将在下面进行介绍。

本文提出的模型架构图见图 2。首先,以流或会话的形式提取 PCAP 文件格式的数据集。之后,按流或会话中的字段提取特征。从 TLS 流的关联 DNS 响应中收集域名、返回码、响应记录和 TTL 生存时间。根据这些信息,可以进一步提取域名的长度和高斯分布,以及网站流量的排名。

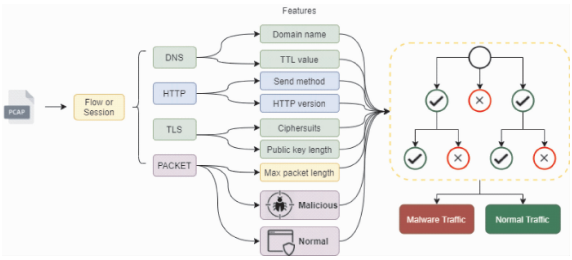

图 2 模型架构

Fig. 2 The model architecture

其次,可以从 HTTP 报文头部和 TLS 有效载荷中提取更多特征。例如 HTTP get 和 post 方法,或 Content-Type、User-Agent、Accept Language 和 Server 等属性。TLS 字段包括协议版本、支持的密码套件和支持的扩展数量等。数据包信息包括流量五元组、字节数、发送方向和发送时间,还可以计算平均包长、最大包长和最小包间延迟等属性。

本文提出的模型侧重于数据包行为的特征,认为不同的数据包有不同的行为,正常流量会有异常流量包,恶意流量也会有正常流量包。本文采用 k-means 聚类方法对正常数据包和恶意数据包进行分类。

输入的数据集格式为 $D=\{x_1,x_2,\cdots x_m\}$,输出是分类的结果 $C=\{C_1,C_2\}$,C_1 和 C_2 分别是正常流量和恶意流量的标签。首先,从数据集 D 中随机选择 2 个样本作为质心集$\{\mu_1,\mu_2\}$,μ_j 是集合的质心,之后计算每个样本 x_i 和质心 μ_j 的距离,距离计算方法为

$$d_{ij}=||x_i-\mu_j||_2^2,$$

之后重新计算集合 C 中的质心,计算公式为

$$\mu_j=\frac{1}{|C_j|}\sum_{x\in C_j}x。$$

计算每个样本与 2 个质心之间的距离,将每个对象分配给距离它最近的质心,质心与它分配的样本就代表一个聚类。所有的样本被分配之后,如果所有的质心向量都没有被改变,则输出聚类的结果。最终,输出簇划分为

$$C=\{C_1,C_2\}。$$

得到流量中正常行为数据包和异常数据包的类别之后,进一步计算出正常行为数据包和异常数据包在流量中的比例以及二者之间的比值,将它们作为参数添加到特征矩阵之中。最终得到样本集合 $S=\{S_1,S_2\,|\,x_i\in S\}$,其中,$x_i$ 是集合 S 中的一个样本。

输入特征集之后,使用 LightGBM 模型进行分类。LightGBM 是一种实现 GBDT 算法的框架,主要思想是使用弱分类器迭代训练以得到最佳模型,支持高效的计算,并且具有更快的训练速度、更高的准确率等优点。LightGBM 使用基尼系数而不是信息增益比。基尼系数越小,杂质越低,特性越好。概率分布的基尼系数表达式为

$$Gini(p)=2p(1-p),$$

p 是属于正常流量的概率。本文使用的损失函数是对数似然损失函数,其计算公式如下:

$$L=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log p_i+(1-y_i)\log(1-p_i)),$$

L 是损失函数,N 是样本数,是输入实例的真实类别,是输入实例属于正常流量类别的预测概率。

3 实验评估

本文提出的分类器已经在真实数据集上进行了实验评估并与其他加密流量分类器进行了比较。因此,本节提供了对实验环境的完整描述。具体来说,本研究第一步将介绍数据集和从原始 PCAP 数据中提取分类特征的方法。然后,将与相关工作的实验结果进行比较。

3.1 数据集

为了区分恶意流量和正常流量,需要为分类器提供加密数据集,用于训练和测试。实验设置采用 Czech Technical University 为恶意软件捕获设施(MCFP)项目收集的 CTU 恶意软件数据集,并以带有 4 个标签的原始 PCAP 格式提供。MCFP 项目通过长期监控收集捕获真实网络环境中的恶意流量和正常流量,并对捕获的流量进行标记。标签包括僵尸网络流量、攻击流量、正常流量和后台流量。目前,数据集包含 300 多个子集,数据格式为 PCAP 文件。

原始 PCAP 文件中会存在大量因为超时、错误或者重传的数据包,这些数据包的数据如果直接引用到本文提出的模型中会严重干扰模型的准确率。所以第一步要处理这些错误数据包,可以借助 Wireshark 工具直观地看到这些错误数据包的存在,错误数据包的格式见图 3。

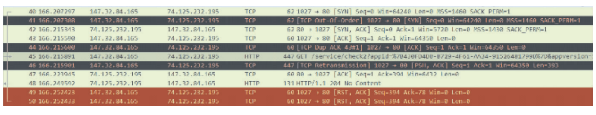

图 3 错误数据包

Fig. 3 Error packets

处理完错误数据包之后,从流量数据中提取有效的字段作为输入特征集。这里的流量提取工具使用 Joy

将 PCAP 文件形式的数据集解析为 JSON 格式,Joy[16]是一种用于网络研究、取证和安全监控的开源数据包分析工具。该工具从网络流量和 JSON 格式文件中提取数据特征,可以得到许多分析工具和编程环境(如 Python 和机器学习框架)的支持。之后,可以将过滤后的文件解析为数据包信息、TLS 加密信息、HTTP 信息或 DNS 信息(更详细的在 2.1 中)。每个流或会话提取的 json 格式数据见图 4。最后一步,将这些提取的特征、统计特征和聚类后的数据包信息保存在 CSV 文件中。


{"sa":"147.32.84.165","da":"184.154.132.106","pr":6,"sp":1398,"dp":9541,"bytes_out":0,
"num_pkts_out":77,"bytes_in":320,"num_pkts_in":37,"time_start":1312972736.776499,
"time_end":1312972767.872890,"packets":[{"b":11,"dir":"<","ipt":471},{"b":5,"dir":"<",
"ipt":689},{"b":11,"dir":"<","ipt":1655},{"b":11,"dir":"<","ipt":345},{"b":11,"dir":"<",
"ipt":1098},{"b":5,"dir":"<","ipt":394},{"b":11,"dir":"<","ipt":1070},{"b":5,"dir":"<",
"ipt":612},{"b":22,"dir":"<","ipt":316},{"b":5,"dir":"<","ipt":568},{"b":5,"dir":"<",
"ipt":532},{"b":11,"dir":"<","ipt":494},{"b":16,"dir":"<","ipt":303},{"b":5,"dir":"<",
"ipt":1749},{"b":5,"dir":"<","ipt":304},{"b":11,"dir":"<","ipt":255},{"b":11,"dir":"<",
"ipt":306},{"b":5,"dir":"<","ipt":288},{"b":5,"dir":"<","ipt":550},{"b":5,"dir":"<",
"ipt":254},{"b":11,"dir":"<","ipt":543},{"b":5,"dir":"<","ipt":848},{"b":5,"dir":"<",
"ipt":528},{"b":16,"dir":"<","ipt":280},{"b":5,"dir":"<","ipt":3262},{"b":11,"dir":"<",
"ipt":342},{"b":5,"dir":"<","ipt":801},{"b":5,"dir":"<","ipt":3843},{"b":11,"dir":"<",
"ipt":262},{"b":5,"dir":"<","ipt":1191},{"b":11,"dir":"<","ipt":3079},{"b":5,"dir":"<",
"ipt":527},{"b":11,"dir":"<","ipt":314},{"b":11,"dir":"<","ipt":1169},{"b":11,"dir":"<",
"ipt":329},{"b":11,"dir":"<","ipt":1226},{"b":5,"dir":"<","ipt":283}],"ip":{"out":
{"ttl":128,"id":[42985,42987,42987,42998,42998,43006,43006,43009,43009,43014,43014,43017,
43017,43019,43019,43020,43020,43024,43024,43025,43025,43026,43026,43030,43030,43033,43033,
43034,43034,43043,43043,43044,43044,43046,43046,43047,43047,43048,43048,43050,43050,43053,
43053,43054,43054,43057,43057,43061,43061,43063]},"in":{"ttl":108,"id":[31811,21306,9915,
21464,23336,2882,10843,1306,12342,234,19406,5222,17097,17368,29406,6966,19618,28136,8420,
18645,10663,11270,682,11145,18493,29812,22968,13159,21885,26256,23237,6111,17245,29572,8024,
18929,27878]}},"debug":{"tcp_retrans":2},"expire_type":"a"}

图 4　数据包提取的 JSON 数据

Fig. 4　JSON data extracted from packet

3.2　实验结果

在这部分,主要实现了 2 个部分的实验内容:第一部分是使用 SVM、随机森林和 LightGBM 的对比实验;第二部分是本研究的方法与其他加密流量检测方法的比较。

SVM 是一种有监督的学习模型,支持线性分类和非线性分类,其决策边界是对学习样本求解的最大边距超平面。随机森林算法是一种基于 Bagging 思想的集成学习方法,它是一种基于决策树作为分类器的集成算法,主要思想是使用多个弱分类器通过投票形成一个强分类器。LightGBM 是一个决策树模型,主要思想是利用弱分类器进行迭代训练,得到最优模型,具有训练效果好,不易过拟合的优点。对比实验见表 1。

表 1　不同算法分类效果对比

Table 1　Comparison of classification effects of different algorithm　%

算法	准确率
SVM	92.03
RF	97.34
LightGBM	99.96

从表 1 中可以看出 LightGBM 实验效果最好。在同一数据集上,本研究与其他研究结果进行了对比实验。本文选取 Frantisek 和 EncrCatcher[22]模型作为对比实验,实验结果见表 2。

表 2　其他模型分类效果对比

Table 2　Comparison of classification effects of other models　%

算法	评价指标		
	准确率	精确率	召回率
Frantisek	97.45	97.34	96.54
EncrCatcher	98.21	98.03	97.55
Our method	99.96	99.27	97.72

本研究的方法使用数据包行为作为加密流分类特征,实验准确率达到 99.96%,优于其他模型。同时,本研究提出的模型准确率为 99.27%,召回率为 97.72%。

4　结　论

如何在不解密数据的情况下,对加密流量进行分类已经成为了时下流行的研究方向。在传统的方法诸如基于端口号以及深度数据包检测技术日趋衰落的情况下,机器学习方法逐渐展露自身的优势。通过对传统机器算法和深度学习算法进行比较,本研究最终选用半监督的传统机器学习模型完成加密流量分类的任务,任务的目标是识别网络流量是正常的还是恶意的。

本文提出了一种基于数据包加密流量分类的机器学习模型,目标是提供一种有效的方法来利用原始 PCAP 文件的信息。本研究除了获得基本的时空特征、头部特征、负载数据和统计特征外,还提出了数据包行为特征。数据包的行为表现了正常流量和恶意流量的区别。通过聚类模型提取正常和恶意流量包,并将结果作为训练特征添加到模型中进行训练。

为了评估模型的有效性,本研究在公共数据集上进行了模型训练,并对机器学习模型进行了实验。同时,对比了现有的加密流量检测模型,实验结果表明,本文提出的方法对加密恶意流量具有更好的检测能力。在此研究基础上,接下来的工作计划是实现多类加密流量分类模型和加密流量检测模型的实时分类。

参考文献:

[1] Chen L, Gao S, Liu B, et al. THS-IDPC: A three-stage hierarchical sampling method based on improved density peaks clustering algorithm for encrypted malicious traffic detection[J]. The Journal of Supercomputing, 2020, 76(5):1-30.

[2] Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview[J]. IEEE Communications Magazine, 2019, 57(5):76-81.

[3] Anderson B, McGrew D. Identifying encrypted malware traffic with contextual flow data[C]//Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. New York:ACM, 2016: 35-46.

[4] Roques O. Detecting malware in TLS traffic[D]. London: Imperial College, 2019.

[5] Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of p2p traffic using application signatures[C]// Proceedings of the 13th International Conference on World Wide Web. New York:ACM, 2004: 512-521.

[6] Kong B, Liu Z, Zhou G, et al. A method of detecting the abnormal encrypted traffic based on machine learning and behavior characteristics[C]//ICCNS 2019: 2019 the 9th International Conference on Communication and Network Security. New York:ACM, 2019:47-50.

[7] Kim S M, Gooy H, Kimm S, et al. A method for service identification of ssl/tls encrypted traffic with the relation of session id and server ip[C]//2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS). Piscataway: IEEE, 2015: 487-490.

[8] Shbair W M, Chole Z T, François J, et al. Improving sni-based https security monitoring[C]//2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW). Piscataway:IEEE, 2016: 72-77.

[9] Husák M,Čermák M, Jirsík T, et al. Https traffic analysis and client identification using passive ssl/tls fingerprinting[J]. EURASIP Journal on Information Security, 2016(1):1-14.

[10] Papadogiannake E, Halevidis C, Akritidis P, et al. Otter: A scalable highresolution encrypted traffic identification engine [C]//International Symposium on Research in Attacks, Intrusions, and Defenses. Berlin: Springer, 2018: 315-334.

[11] Bilge L, Balzarotti D, Robertson W, et al. Disclosure: Detecting botnet command and control servers through large-scale netflow analysis[C]//Proceedings of the 28th Annual Computer Security Applications Conference. New York: ACM, 2012: 129-138.

[12] Panchenkoa A, Lanze F, Zinnen A, et al. Website fingerprinting at internet scale[C]//Network & Distributed System Security Symposium. Reston:ISOC, 2016:1-15.

[13] Wurzinger P, Bilge L, Holz T, et al. Automatically generating models for botnet detection[C]//European symposium on research in computer security. Barlin: Springer, 2009: 232-249.

[14] Mcgrew D, Anderson B. Enhanced telemetry for encrypted threat analytics[C]//2016 IEEE 24th International Conference on Network Protocols (ICNP). Piscataway: IEEE, 2016: 1-6.

[15] Wei W, Ming Z, Wang J, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]//2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Piscataway: IEEE, 2017, doi:10.11091ZSI.2017.8004872.

[16] Anderson B, Mcgrew D. Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity[C]//Acm Sigkdd International Conference. New York: ACM, 2017:1723-1732.

[17] Prasse P, Machlica L, Pevny T, et al. Malware detection by analysing network traffic with neural networks[C]//2017 IEEE Security and Privacy Workshops (SPW). Piscataway: IEEE, 2017:205-210.

[18] Wang W, Zhu M, Zeng X, et al. Malware traffic classification using convolutional neural network for representation learning [C]//2017 International Conference on Information Networking (ICOIN). Piscataway: IEEE, 2017: 712-717.

[19] Rezaei S, Liu X. How to achieve high classification accuracy with just a few labels: A semisupervised approach using sampled packets[EB/OL]. (2018-12-23)[2022-03-08]. http://arxiv.org/abs/1812.09761.

[20] Lopez-Martin M, Carro B, Sanchez-Esguevillas A, et al. Network traffic classifier with convolutional and recurrent neural networks for internet of things[J]. IEEE Access, 2017,5(18):42-50.

[21] Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview[J]. IEEE Communications Magczin, 2019,57(5):76-81.

[22] Střasák F. Detection of https malware traffic[EB/OL]. [2022-03-29]. https://dspace.cvut.cz/bitstream/ handle/10467/68528/F3-BP-2017-Strasak-Frantisek-strasak_thesis_2017.pdf.

【责任编辑:孙向荣】