

结合多特征识别的恶意加密流量检测方法

李慧慧¹, 张士庚^{1,2}, 宋虹¹, 王伟平¹

¹中南大学计算机学院 长沙 中国 410083

²中国科学院信息工程研究所信息安全国家重点实验室 北京 中国 100093

摘要 随着加密流量的广泛使用, 越来越多恶意软件也利用加密流量来传输恶意信息, 由于其传输内容不可见, 传统的基于深度包分析的检测方法带来精度下降和实时性不足等问题。本文通过分析恶意加密流量和正常流量的会话和协议, 提出了一种结合多特征的恶意加密流量检测方法, 该方法提取了加密流量会话的包长与时间马尔科夫链、包长与时间分布及包长与时间统计等方面的统计特征, 结合握手阶段的 TLS 加密套件使用、证书及域名等协议特征, 构建了 863 维的特征向量, 利用机器学习方法对加密流量进行检测, 从而发现恶意加密流量。测试结果表明, 结合多特征的恶意加密流量检测方法能达到 98% 以上的分类准确性及 99.8% 以上召回率, 且在保持相当的分类准确性基础上, 具有更好的鲁棒性, 适用性更广。

关键词 加密流量, 恶意检测, TLS 协议分析, 鲁棒性

中图分类号 TP393 DOI 号 10.19363/J.cnki.cn10-1380/tn.2021.03.09

Robust Malicious Encrypted Traffic Detection based with Multiple Features

LI Huihui¹, Zhang Shigeng^{1,2}, Song Hong¹, Wang Weiping¹

¹School of Computer Science and Engineering, Central South University, Changsha 410083, China

²State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract With the widespread use of encrypted traffic, more and more malware also uses encrypted traffic to transmit malicious information. Since the transmission content is not visible, the traditional detection method based on deep packet inspection brings problems such as accuracy reduction and insufficient realtime performance. In this paper, by analyzing the protocol and the sessions of malicious encrypted traffic and normal traffic, a method for detecting malicious encrypted traffic combining multiple features is proposed. The method extracts the statistical characteristics of encrypted sessions such as the Markov chain of packet length and time, the distribution of packet length and time, and the statistical values of packet length and time. Combined with protocol features such as the use of TLS cipher suites in the handshake phase, certificates and domain names, an 863-dimensional feature vector is constructed. We use machine learning methods to detect encrypted traffic to discover malicious encrypted traffic. The test results show that the robust malicious encryption traffic detection method based on multiple features can achieve a classification accuracy of more than 98% and recall value of more than 99%, and the new method can receive better robustness while keeping the high classification accuracy and can be applied wider.

Key words encrypted traffic, malicious detection, TLS protocol analysis, robustness

1 引言

随着网络技术的快速发展, 互联网已经在军事、经济、教育、生活等各个领域都广泛应用。然而, 在互联网给我们的生活带来各种便利的同时, 也带来了各种安全问题, 各种计算机病毒、蠕虫等恶意软件的数量和种类也在快速增多, 为互联网用户的安

全带来了巨大挑战。

为了保护传输的数据, 加密传输已经成为现有广泛应用的方式。Cisco 的调查^[1]显示, 仅 2016—2017 年, 加密流量就增多了 90% 以上, 超过 50% 以上的流量都是加密流量。采用加密传输有益于保护普通用户的隐私, 然而这也给了恶意应用开发者可乘之机, 他们开始大量使用加密流量来逃避检测。

通讯作者: 宋虹, 博士, 副教授, Email: songhong@csu.edu.cn。

本课题得到国家自然科学基金项目(No. 61772559、No. 61672543), 中南大学研究生科研创新项目(No. 1053320183917)的资助。

收稿日期: 2020-04-30; 修改日期: 2020-07-12; 定稿日期: 2020-12-21

Cisco 预测到 2020 年, 将有 70% 的恶意软件将使用某种类型的加密来隐藏恶意软件的传递、远程控制以及数据泄露等恶意行为。其中最常用的加密方式是使用 TLS 协议加密。因此, 如何检测恶意 TLS 加密流量已成为恶意软件检测识别中的一个重要研究热点。

已有针对明文 HTTP 流量的恶意流量检测方法, 如基于签名的方法^[2-4]和基于语义特征的方法^[5-6], 分析了恶意流量在 HTTP 请求头部的签名信息和分割单词的语义特征, 发现恶意流量并对恶意流量进行分类。由于加密流量信息对于传输内容进行加密, 因此, 语义特征信息无法从流量中获取, 因此, 这些方法无法在加密流量的恶意性检测中应用。越来越多的加密流量恶意性检测采用数据包大小、方向、时间间隔等统计特征对流量进行分类^[7-10], 还有一些研究^[11-17]利用 TLS 握手阶段的可用特征, 包括握手消息类型、加密套件、扩展、公钥长度、SSL/TLS 版本号、加密方法等, 作为识别恶意流量的特征参量; 另外文献[18]主要考虑了 TLS 协议中的证书, 采用证书内容来识别正常流量和加密流量; 随着深度学习研究的发展, 也有一些研究直接采用了深度学习方法^[19-20], 将原始流量数据直接输入深度学习网络进行恶意流量识别。这些方法在恶意流量检测方面都获得了一些良好结果, 但也存在一些不足, 主要体现在: (1) 基于证书的方法对无证书传递的加密会话恶意性检测无效, 因为大多数加密会话可采用 TLS 会话复用方式传递; (2) 深度学习方法虽然无需复杂的特征提取工程, 但缺乏可解释性, 而且需要大量的训练数据; (3) 基于签名的方法和特征的方法大多需要分析流量内容中的信息, 而且只考虑流量某一方面的特征, 并未结合加密流量的特殊性以及加密协议的发展变化。

因此, 本文提出了一种结合多特征识别的恶意加密流量检测方法 RMETD-MF(Robust Malicious Encrypted Traffic Detection based with Multiple Features), 尝试在不对加密流量做解密的情况下实现恶意检测, 并能识别不同网络场景中新的恶意加密流量。本文的主要贡献如下:

(1) 通过监控网络流量, 分析了恶意流量和正常流量的区别, 提取了其中能明显区别正常流量和恶意流量的 69 个特征, 包括会话元数据特征、包长序列统计特征、TLS 握手特征、证书特征和域名特征, 用于恶意流量的识别和检测;

(2) 提出了结合多特征识别的恶意加密会话检测方法 RMETD-MF, 利用上述明显区别正常流量和

恶意流量的 69 个特征, 构建了 863 维特征向量, 实现加密会话的快速准确的分类。在收集的校园网数据集、企业数据集及学术数据集上进行十折交叉验证获得 99.9% 以上的分类准确度效果。

(3) 分析了 RMETD-MF 方法在不同时间流量数据的检测效果以及不同数据集下分类的效果, 验证了方法的稳定性和分类的准确度。

论文的剩余部分组织如下, 第 2 节分析了恶意加密流量和正常加密流量的特征比较; 第 3 节阐述了 RMETD-MF 方法的基本思想和详细设计; 第 4 节分析了方法在数据集上的性能测试和评估, 给出了我们所使用的数据集以及在这些数据集上进行的实验; 第 5 节给出了总结和下一步工作。

2 流量分析

通过对恶意加密流量和正常加密流量的深入分析, 可观察获得用于区分恶意流量与正常流量的显著特征。并选择其中比较稳定的、不易受时间和场景影响的特征作为后续 RMETD-MF 方法中使用的分类特征。

2.1 相关定义

定义 1: 一对通信结点交互产生的单向数据包集合称为流(Flow), 第 i 对通信节点之间的流 $Flow_i$ 定义为集合 $\{SrcIP, DstIP, SrcPort, DstPort, TP\}$, 其中, $SrcIP$ 为该流中的源 IP 地址、 $DstIP$ 是目的 IP 地址、 $SrcPort$ 为源端口号、 $DstPort$ 为目的端口号、 TP 表示通信节点间的传输层协议。

定义 2: 一对通信结点交互产生的双向网络包集合定义为会话(Session), 会话是源和目的可互换的双向流。一个会话为客户端与服务器的从建立连接到完成数据传输的完整过程。网络通信环境中, 多个用户通过建立多个连接发送和接收数据包, 因此会话是流量分析和检测的最小单位。

定义 3: 包长/时间序列 $S=\{P_1, P_2, \dots, P_n\}$ 为一个会话中数据包大小或数据包之间的时间间隔组成的序列。序列中的每个值 $P_i, i \in [1, n]$ 代表第 i 个数据包的大小或者第 i 个数据包与其之前的第 $(i-1)$ 个数据包之间的时间间隔, 排列顺序和会话中数据包的传输顺序一致。

2.2 TLS 协议分析

TLS 协议是加密流量传输中常用的协议, 它主要分为握手阶段和加密通信阶段, TLS 握手过程如图 1 所示, 完整的 TLS 握手过程如图 1(a) 所示, 会话复用时握手过程如图 1(b) 所示。

如图 1(a), 一个完整的握手过程首先由客户端向服务器端发送 Client Hello 消息, 服务器接收到该消息后, 将返回 Server Hello 消息, 同时传递服务器的证书、服务器的密钥交换信息, 并携带客户端的证书请求; 在客户端接收到服务器端的 Server Hello 消息后, 保存服务器端密钥交换信息, 并将自己的证书及验证信息、密钥交换信息和 Change Cipher Spec 消息发送给服务器端; 服务器端接收到信息后, 发送 Change Cipher Spec 消息给客户端, 由此建立了客户端和服务器的加密通信链路。

而在会话复用阶段(图 1(b)), 客户端和服务端之间握手同样也是从 Client Hello 开始, 以 Change Cipher Spec 结束为止。

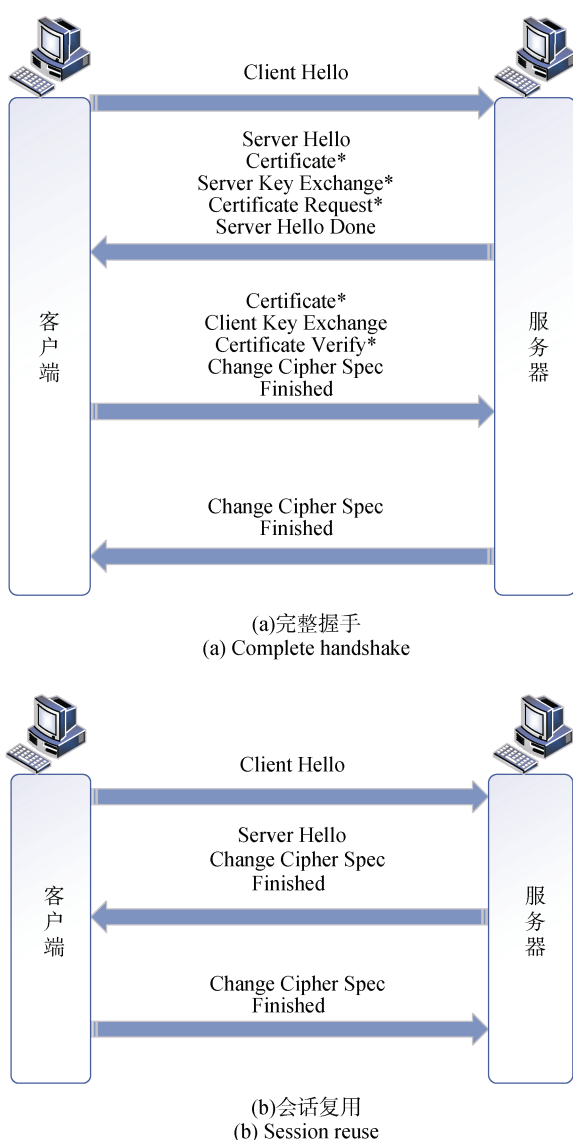


图 1 TLS 握手过程

Figure 1 The process of TLS handshake

握手阶段消息为明文信息, 加密通信阶段为加

密后的密文信息。在不对流量解密的情况下, TLS 加密会话中可观察到的为握手阶段传输的包的内容, 以及无论数据包是否加密都可得到的数据包的长度, 时间, 方向等传输层及以下各层的头部信息。握手阶段客户端会请求及验证服务器端证书, 可通过解析证书得到证书信息。加密会话的 URL 被加密, 但其 Client Hello 消息中的 Server Name Indication 扩展中指明了要访问的服务器域名, 服务器证书中也会显示服务器的公用名。由于域名和证书对分类结果的影响明显, 因此虽然这两个特征也是在 TLS 握手阶段得到的, 在特征向量中, 域名和证书作为单独特征向量进行考虑。

2.3 流量特征分析

大多数的恶意软件并不是从零开始编写的新型恶意软件, 而是通过对已有的恶意软件进行代码复用和修改而生成的变体。同一恶意软件的不同变体在代码和行为上都较为相似, 通常将这样功能、行为类似的恶意软件归为同一个恶意家族。同一个恶意家族的软件通常会调用相同或相似的函数, 执行类似的行为, 包括系统行为和网络行为。

2.3.1 会话的统计特征分析

为了获得恶意加密会话与正常加密会话的特征区别, 我们分别统计了 15 个恶意家族的数据包和正常加密流量数据包的信息。由于同一恶意家族的加密会话存在一定的相似性, 导致根据加密会话提取的统计特征在同一家族的加密会话中也十分相似。分析发现:

(1) 包数量的特征

从数据包的数量上看, 恶意加密流量和正常加密流量区别明显。绝大多数恶意家族的有效载荷发送数据包为 3~5 个; 除了恶意家族 Trojan.MSIL.Disfa 的约 80% 的加密会话接收了 10~30 个有效载荷的数据包之外, 大部分恶意家族的接收数据包数量为 3~5 个。而正常加密流量数据包的个数变化范围较大, 从 3 个到几千个, 大部分正常加密数据包个数为几十个。

(2) 包长序列的特征

恶意家族的包长序列具有一定的相似性, 如 trojan.win32.zbot 家族的加密会大部分都链接到网站 infoplusplus.com 或 ax100.net 网站, 其中, 链接到 infoplusplus.com 的会话包长序列为 {403, 105, 51, 176, 508}, 而链接到 ax100.net 网站的包长序列为 {396, 105, 51, 170, 509}。正常加密会话的包长序列是变化的, 并没有固定的长度。

(3) 会话持续时间的特征

观察恶意家族的加密会话持续时间, 可以发现

除几个恶意家族的会话持续时间较长(如 HEUR:Trojan.Win32.StartPage 的大部分加密会话持续时间超过 1min, 最长为 177s), 其他多数恶意家族约在 20s 内完成加密会话。而正常的加密会话没有此特征。

(4) 数据包顺序与大小特征

恶意家族的发送/接收数据包的顺序与大小较为相似。如恶意家族 Trojan.Win32.SelfDel 的 26836 个加密会话仅有 123 种不同的数据包序列, 恶意家族 Trojan-Spy.Win32.Zbot 的 16758 个加密会话中仅有 219 种不同的数据包序列。而且, 同一恶意家族的会话的不同数据包序列可能具有相同的数据包子序列, 会话的前几个数据包序列是相同的。

2.3.2 TLS 协议特征分析

建立 TLS 连接的第一步是客户端向服务器端发

送明文 Client Hello 消息, 并将自己所支持的按优先级排列的加密套件信息和扩展列表发送给服务器端, 这一消息的生成方式取决于构建客户端应用程序时所使用的软件包和方法。服务端反馈 Server Hello 消息, 包含选择使用的加密套件、扩展列表和随机数等, 这一消息基于服务器端所用库和配置以及 Client Hello 消息中的详细信息创建。由于大部分恶意软件会复用同一个恶意软件的代码, 因此许多恶意软件的 Client Hello 消息在一些特征上十分相似, 如加密套件、扩展等。

(1) 加密套件使用的特征

恶意会话和正常会话的客户端加密套件列表如表 1 所示, 其中加密套件列表中的数字是 TLS 为每个加密套件分配的唯一标识号。

表 1 客户端加密套件列表所占比例
Table 1 Proportion of the list of cipher suites provided by the client

客户端加密套件列表		所占比例(%)
恶意会话	47,53,5,10,49171,49172,49161,49162,50,56,19,4	67.85
	49196,49195,49200,49199,159,158,49188,49187,49192,49191,49162,49161,49172,49171,57,51,157,156,61,60,53,47,10,106,64,56,50,19	19.09
	60,47,61,53,5,10,49191,49171,49172,49195,49187,49196,49188,49161,49162,64,50,106,56,19,	2.96
	44865,4867,4866,49195,49199,52393,52392,49196,49200,49171,49172,47,53,10	2.07
	49200,49196,49192,49188,49172,49162,49186,49185,163,159,107,106,57,56,136,135,49202,49198,49194,49190,49167,49157,157,61,53,132,49170,49160,49180,49179,22,19,49165,49155,10,49199,49195,49191,49187,49171,49161,49183,49182,162,158,103,64,51,50,154,153,69,68,49201,49197,49193,49189,49166,49156,156,60,47,150,65,7,49169,49159,49164,49154,5,4,21,18,9,20,17,8,6,3,255	1.79
	49199,49200,49195,49196,52392,52393,49171,49161,49172,49162,156,157,47,53,49170,10	1.15
	49199,49195,49200,49196,49171,49161,49172,49162,156,157,47,53,49170,10	1.07
	49195,49199,49162,49161,49171,49172,51,57,47,53,10	18.21
	52393,52392,52244,52243,49195,49199,49196,49200,49161,49171,49162,49172,156,157,47,53,10	12.71
	49196,49195,49200,49199,159,158,49188,49187,49192,49191,49162,49161,49172,49171,157,156,61,60,53,47,10	7.08
正常会话	49200,49196,49192,49188,49172,49162,165,163,161,159,107,106,105,104,57,56,55,54,136,135,134,133,49202,49198,49194,49190,49167,49157,157,61,53,132,49199,49195,49191,49187,49171,49161,164,162,160,158,103,64,63,62,51,50,49,48,57361,57345,154,153,152,151,69,68,67,66,49201,49197,49193,49189,49166,49156,156,60,47,150,65,7,57363,57347,49169,49159,49164,49154,5,4,49170,49160,22,19,16,13,49165,49155,10,255	5.59
	49195,49199,158,49162,49161,49171,49172,51,57,156,47,53,10	4.46
	4,5,47,51,50,10,22,19,9,21,18,3,8,20,17,255	4.36
	49187,49191,60,49189,49193,103,64,49161,49171,47,49156,49166,51,50,49195,49199,156,49197,49201,158,162,49160,49170,10,49155,49165,22,19,255	3.85

由表 1 可知, 恶意流量的 129 种加密套件列表中, 67.85% 采用了加密套件列表 {47,53,5,10,49171,49172,49161,49162,50,56,19,4}; 19.09% 的加密会话使用的是 {49196,49195,49200,49199,159,158,49188,49187,49192,49191,49162,49161,49172,49171,57,51,157,156,61,60,53,47,10,106,64,56,50,19}。95% 的恶意加密会话使用相同的 7 个加密套件列表。正常流量中共 266 种不同的加密套件列表, 分布较为分散, 其中

使用较多的加密套件列表为: 18.21% 的加密会话使用的加密套件列表为 {49195,49199,49162,49161,49171,49172,51,57,47,53,10}; 12.71% 的加密会话使用的加密套件列表为 {52393,52392,52244,52243,49195,49199,49196,49200,49161,49171,49162,49172,156,157,47,53,10}; 恶意会话中使用最高的加密套件列表在正常会话中所占的比例仅为 2.20% 和 1.14%。可见恶意流量和正常流量在加密套件的使用上有明显区别。

客户端所用的单个加密套件占比例高的情况统计如图 2 所示, 纵坐标为占总数的比例。从图 2 中可以看出加密套件 4、5、19、50、56 在恶意流量中的使用明显多于正常流量中的使用。而这些加密套件被认为是比较弱的加密套件, 安全性不足, 在正常的软件中不被推荐使用。客户端所用加密套件个数如图 3 所示, 服务器端所选加密套件如图 4 所示。由

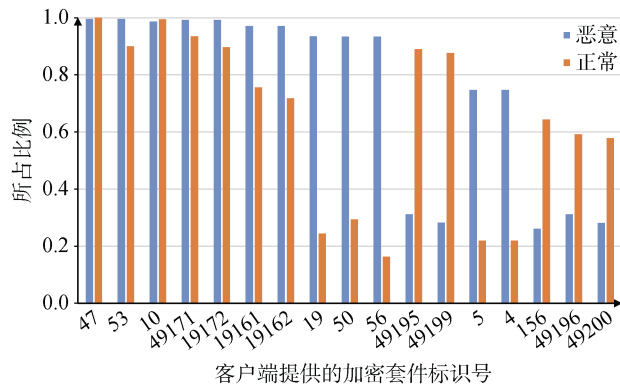


图 2 客户端加密套件比较

Figure 2 Comparison of client cipher suites

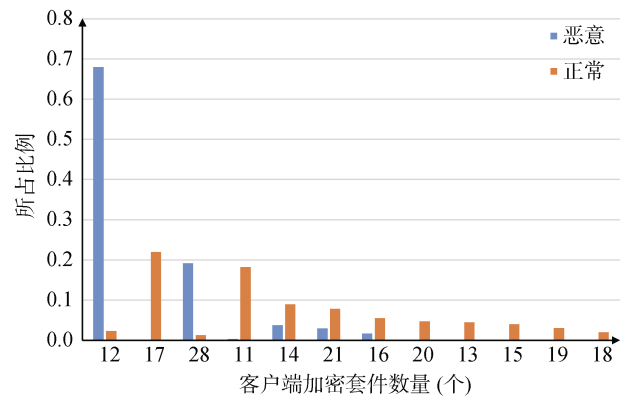


图 3 客户端加密套件个数比较

Figure 3 Comparison of the number of client cipher suites

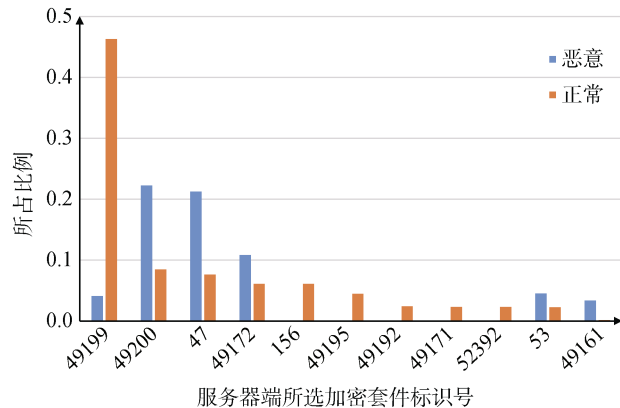


图 4 服务器端加密套件比较

Figure 4 Comparison of Server cipher suites

图可以看出, 恶意会话中客户端集中使用 12 个和 28 个加密套件, 服务器端选择最多的加密套件是 49200,47,49122; 而正常会话中, 客户端多使用 17,14,11 个加密套件, 服务器端选择最多的是 49199,49200,47,49122。

(2) 扩展加密套件使用的特征

扩展加密套件显示了支持协议、算法相关参数以及其他辅助信息, 正常会话和恶意会话在客户端及服务器端使用的情况如图 5~8 所示。

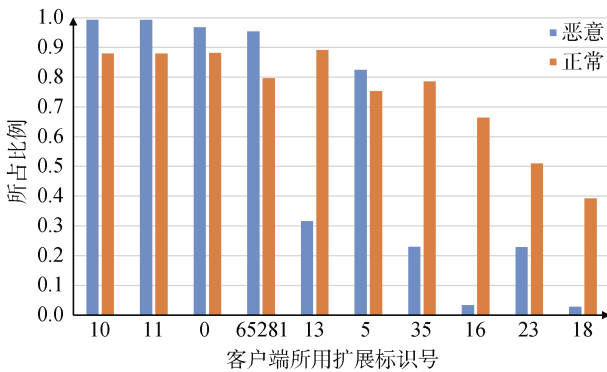


图 5 客户端扩展比较

Figure 5 Comparison of client extensions

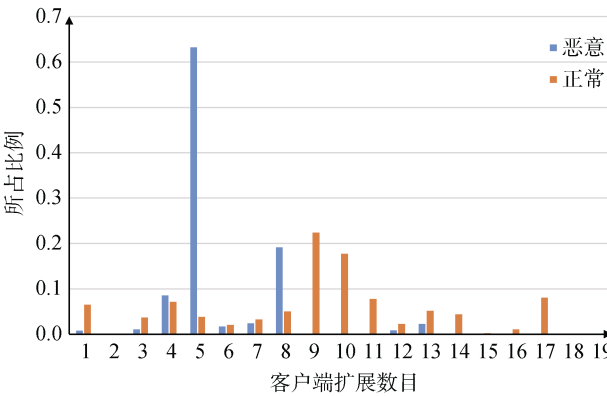


图 6 客户端扩展数目比较

Figure 6 Comparison of the number of client extensions

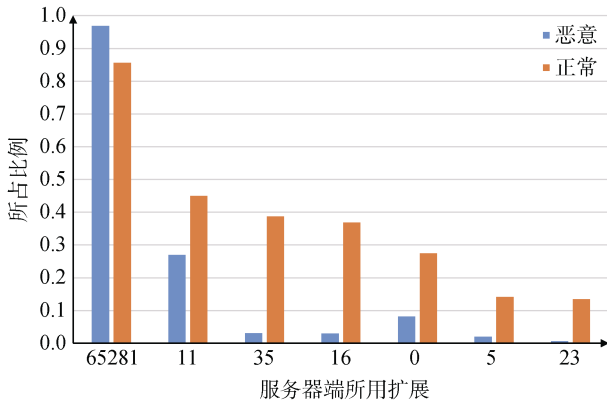


图 7 服务器端扩展比较

Figure 7 Comparison of server expansions

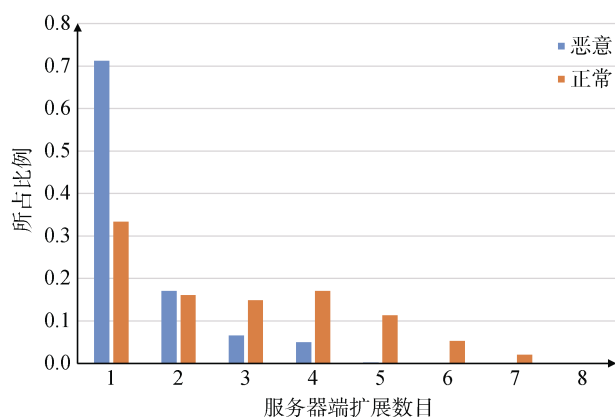


图 8 服务器端扩展数目比较

Figure 8 Comparison of the number of server extensions

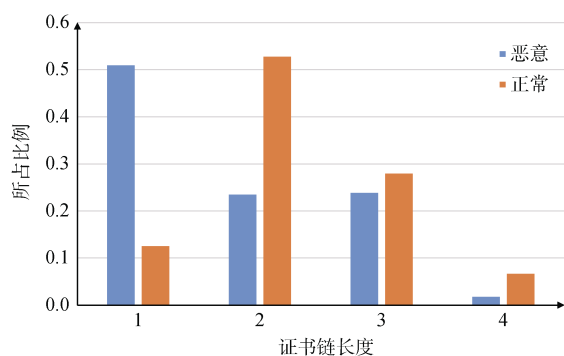
由图可知, 正常加密会话客户端常用的扩展种类较多, 恶意会话客户端最多使用的扩展标识号为 10,11,0,65281,5。恶意加密会话客户端所使用的扩展数目多为 5 个, 而正常会话的客户端扩展数目较为多样。正常会话与恶意会话在服务器端所用扩展最多的均为 65281, 数目最多的均为 1, 但是分布情况存在一定的差异。

2.3.3 服务器证书特征

服务器证书是 TLS 协议中用来对服务器身份进行验证的文件。由于会话复用的广泛使用, 36%的正常会话没有传输证书, 而 88%的恶意会话没有传输证书。常见的 CA 证书是指由受信任的 CA 机构颁发的证书, 申请时会对域名所有权和企业相关信息进行验证, 安全级别较高, 受各大浏览器的信任, 需要付费。而自签名证书不需要付费, 任何人都可以签发, 其 issuer 与 subject 相同。

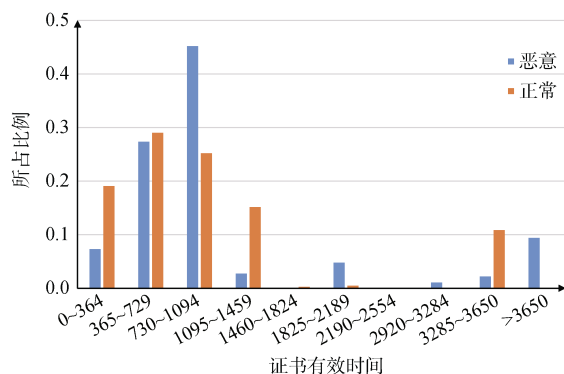
在传输了证书的加密会话中, 3.34%的恶意会话证书版本为 1, 而正常会话中基本都是版本 3。恶意证书与正常证书一个区别较大的特征是证书是否是自签名证书, 大部分恶意软件为了方便会选择使用自签名证书。约 48%的恶意加密会话为自签名证书, 而正常会话中约 12%的会话采用自签名证书。一般情况下, 证书的 Common Name 会填写证书的域名或子域名, 但是自签名证书可以随意填写。85%的正常会话所用证书中的 Common Name 为域名, 62%为 .com 域名, 而恶意会话中只有 53%为域名, 30%为 .com 域名。

图 9(a)~(d)显示了恶意会话与正常会话在证书的使用上的区别。由图 9 可知, 正常会话的证书链的长度常为 2, 3, 而恶意会话中证书链长度最常为 1, 这是因为恶意会话的证书常为自签名。恶意会话所用



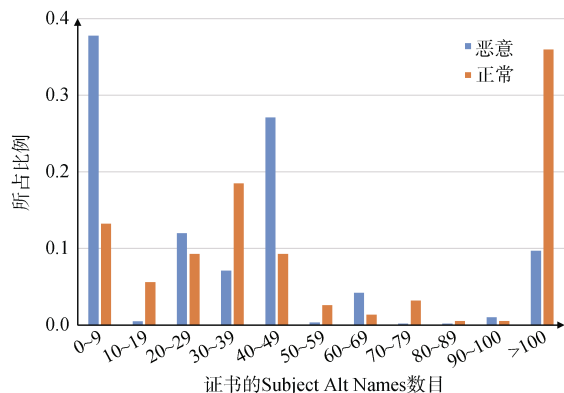
(a) 证书链长度对比

(a) Comparison of certificate chain length



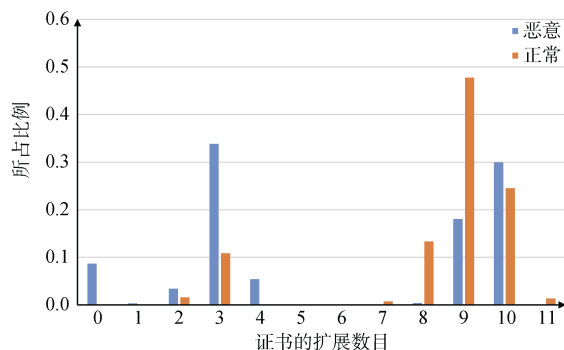
(b) 证书有效时间对比

(b) Comparison of certificate validity time



(c) 证书的Subject Alt Names数目对比

(c) Comparison of the number of Subject Alt Names of the certificate



(d) 证书的扩展数目对比

(d) Comparison of the number of certificate extensions

图 9 恶意会话与正常会话证书对比

Figure 9 Comparison of certificates for malicious sessions and normal sessions

证书的有效时间最多的是 2~3 年, 也存在很多超过十年的有效时间的恶意会话, 这是因为自签名证书支持超长有效期, 而正常会话所用证书的有效时间多分布在 0~4 年间。正常会话的证书的 Subject Alt Names 数目有接近 40% 的比例是大于 100 个的, 恶意会话最多的为 0~9 个, 以接近 40% 的比例。恶意会话中证书扩展数目最多的为 3、10、9、0, 而正常会话中最多的为 9、10、8、3、2。

2.3.4 服务器域名特征

Client Hello 中的 Server Name Indication 扩展用于指示客户端请求的服务器域名, 防止一个 IP 连接多个服务器而造成错误。当 Client Hello 中无域名指示时, 则取证书中的 Common Name 作为服务器域名。

所分析的 202559 个恶意加密会话中, 有约 2.6% 的会话既无 Server Name 又无 Common Name, 而正常加密会话中有约 6.6% 的会话既无 Server Name 又无 Common Name; 恶意会话中域名较为分散, 大多数为比较不常见的域名, 而正常会话中域名多为常见域名。恶意会话域名在 Alexa 排名如图 10 所示, 可以看出, 恶意会话所用域名有 85% 以上不在前一百万排名内, 与之相反, 正常会话中 85% 以上都位于前一百万排名内。

这是因为正常会话多连向一些常见的正常网站, 而恶意会话多连向一些由域名生成算法生成的不常见的网站, 则其域名排名较为靠后。

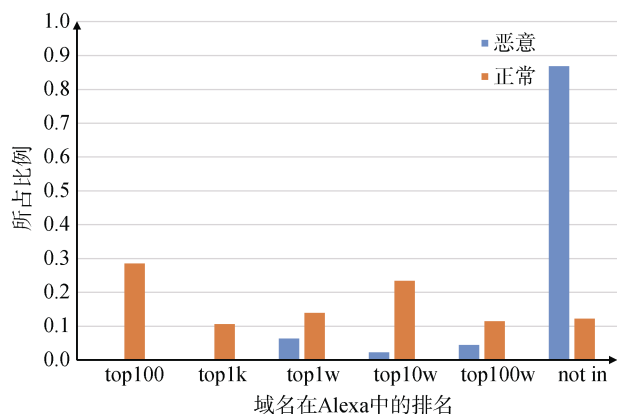


图 10 恶意会话与正常会话域名排名对比

Figure 10 Comparison of domain ranking of malicious sessions and normal sessions

3 RMETD-MF 方法

通过分析, RMETD-MF 方法的基本思想是通过监控和捕获网络加密流量, 提取流量中的区别比较

明显的会话统计特征和 TLS 协议相关特征, 构建 863 位特征向量, 训练机器学习分类模型, 并利用该训练模型对其他加密流量进行检测, 识别其是否是恶意流量。整体检测流程如图 11 所示。

如图 11 所示, 整个测试流程包括流量捕获、流量预处理、特征提取以及模型训练等四个部分。流量捕获部分主要是采用工具 wireshark、tcpdump 等抓包工具来完成采集, 为了得到训练阶段加密的恶意流量及正常流量, 流量捕获阶段将在沙箱中运行恶意软件和正常软件生成流量数据; 流量预处理阶段主要完成对流量的清洗, 过滤未加密流量和不完整的会话, 生成可用于流量检测的会话信息; 特征提取阶段主要是根据需求, 提取相关会话的统计特征和系数特征信息, 形成训练的 863 位特征向量; 而模型训练阶段主要是构建恶意加密流量分析的模型。

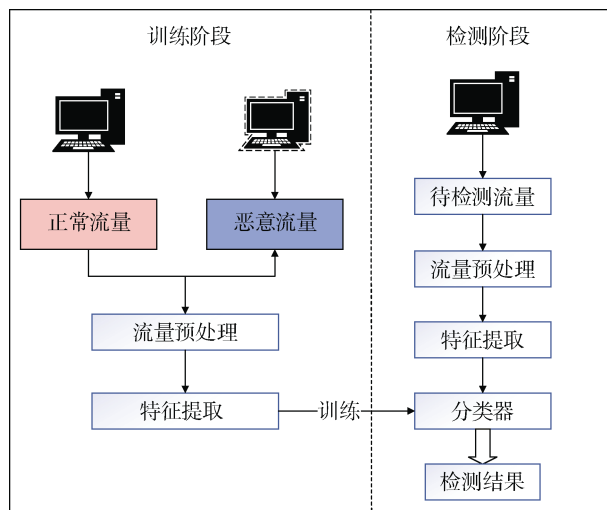


图 11 整体检测流程图

Figure 11 The overall detection process

3.1 流量捕获

为了获得训练用的纯净加密流量和检测阶段的实时流量数据, 构建了如图 12 所示的流量捕获模型。

图 12 中, 正常流量的获取通过在监控计算机上运行 wireshark 等工具捕获访问正常加密网站或运行正常软件产生的流量来获得, 或者通过监控较为干净的网络环境流量来获得, 并通过白名单过滤获得白名单中的会话作为正常流量。

恶意流量的获取采用沙箱方式, 在沙箱中运行恶意软件, 保存其运行期间产生的流量, 然后过滤掉沙箱间通信流量及系统白流量, 将剩余的流量作为恶意流量。

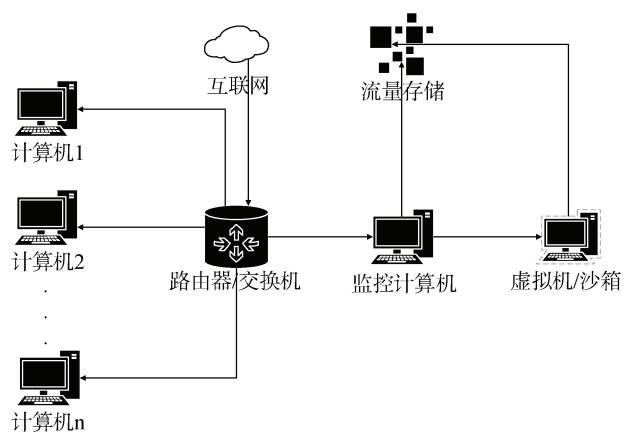


图 12 流量采集模型

Figure 12 Traffic collection model

3.2 流量预处理

为了提取出可用于加密流量恶意检测的会话, 对加密流量进行预处理:

(1) 过滤未加密的流量, 保留使用 SSL/TLS 协议的流量;

(2) 过滤会话, 从混杂的包中提取会话, 过滤未完成完整握手过程和未传输加密数据的会话。通过观察会话中是否有 Client Hello 消息和 Change Cipher Spec 消息, 来判断握手是否完成; 通过观察会话中是否有 Application Data 消息, 来判断会话是否传输了加密数据。

(3) 过滤重传包、确认包及传输丢失的坏包, 以避免对分类造成影响。

3.3 特征提取

提取每个加密会话有关流量的统计特征、SSL/TLS 握手特征、证书特征和域名特征, 形成特征向量, 作为恶意流量识别的输入。

3.3.1 会话的统计特征提取

(1) 元数据特征

元数据特征是指会话的一般信息, 包括客户端向服务器端发送的包数、服务器端向客户端发送的包数、客户端向服务器端发送的字节数、服务器端向客户端发送的字节数、会话持续时间、平均每个发送包的字节数以及平均每个接收包的字节数, 形成 7 维的特征向量。

(2) 包长与时间序列特征

会话中最大传输单位为 1500 字节, 将获取的会话中数据包长度分段统计, 10 个分段的范围分别是 $[0, 150)$, $[150, 300)$, \dots , $[1350, +\infty)$, 构建每个数据包有效载荷的长度及相邻包之间的转换关系矩阵,

采用 10×10 的马尔可夫状态转移概率矩阵, 并按行拼接维 100 维特征向量。

构建相邻数据包的时间间隔序列特征, 将时间间隔分为十个分段 $[0, 50\text{ms}]$, $[50\text{ms}, 100\text{ms}]$, \dots , $[450\text{ms}, +\infty]$, 根据相邻包之间的时间间隔所在的区间及转换关系构建 10×10 的马尔可夫转换矩阵, 将其也按行拼接作为 100 维的特征向量。

(3) 包长与时间分布特征

包长分布: 将包长分为 150 个不同的范围 $(0, 10)$, $(10, 20)$, \dots , $(1490, +\infty)$, 根据每个包的长度计算每个包长区间分布的包数, 作为 150 维的特征。

时间分布: 将时间间隔分为 100 个不同的范围 $(0, 0.005)$, $(0.005, 0.01)$, \dots , $(0.45, +\infty)$, 根据每个包与前一个包的时间间隔计算每个时间间隔区间分布的包数, 作为 100 维的特征向量。

(4) 包长与时间统计特征

分别计算包长序列和时间序列的统计特征: 个数, 最小值, 最小元素位置, 25% 分数, 中位数, 75% 分数, 均值, 最大值, 最大元素位置, 平均绝对方差, 方差, 标准差, 形成 24 维的特征向量。

3.3.2 TLS 握手特征提取

提取 Client Hello 中的加密套件列表信息和扩展信息, 结合 Server Hello 中的加密套件信息和扩展信息, 构建握手特征的特征向量。

(1) 客户端 tls 特征

观察发现客户端使用的加密套件共 260 种, 因此, 设置 260 维向量, 根据客户端提供的加密套件列表, 在对应的向量位上置 1 或 0, 即若加密套件被使用, 置为 1, 否则为 0; 计算加密套件列表中加密套件的个数, 也作为一维特征向量输入。

同时, 对客户端支持的扩展列表构建 43 维向量, 对应所使用的 43 个扩展加密套件; 计算扩展个数, 作为一维特征向量输入。

(2) 服务器端 tls 特征

将服务器端所选择的加密套件作为一维特征向量输入, 同时服务器端支持的扩展列表构建成 43 维特征向量, 而扩展个数也是 1 维特征向量。

3.3.3 证书特征提取

提取服务器证书的特征构建特征向量, 包括自签名属性、证书链长度、有效时间、平均长度、别名数量、扩展数目、证书版本、证书序列号、证书主体、证书颁发者、证书的 Subject 和 Issuer 特征等共 23 个, 形成 23 维的特征向量, 如表 2 所示。

表 2 证书特征列表

Table 2 List of certificate features

编号	特征名称	特征描述
1	是否自签名	证书的 issuer 与 subject 相同
2	证书链长度	证书链上的证书个数
3	有效时间	证书有效时间
4	平均长度	证书链上的所有证书长度与证书个数的比值
5	别名数目	证书扩展 Subject Alt Names 中的域名数量
6	扩展数目	证书的扩展数目
7	版本	证书版本
8	Subject 特征	提取 Subject 中 O、CO、ST、L 和 CN 等特征信息, 这些信息分别代表单位名称、公司名、省份、程序公用名等信息
		与 Subject 类似, 提取 Issuer 中的 O、CO、ST、L 及 CN 信息, 同时提取所包含的元素个数
9	Issuer 特征	

3.3.4 域名特征

提取以下两个域名特征作为特征输入:

(1) 域名特征

根据 DGA 生成算法可能导致恶意网站的域名与正常网站的字母数字等上的区别, 提取有关域名的特征, 包括域名中字母符号数目占有所有字符的比例、数字符号数目占有所有字符的比例以及非字母数字符号数目占有所有字符的比例。

(2) 排名特征

根据域名在 Alexa 前 100 万列表中的排名, 构建一个长度为 6 的向量, 根据其是否在 top100, top1000, top1w, top10w, top100w, not-in 列表中进行向量设置, 在则将该位置为 1, 如果都不在就置为 0, 若不在前 100w 列表就置 not-in 位为 1。

3.4 模型训练与测试

采用机器学习的方法对输入的特征向量进行二分类, 训练出分类模型, 并在测试阶段使用训练好的模型进行流量检测, 输出正常或恶意加密流量的分类检测结果。

4 实验评估

4.1 数据集

4.1.1 正常数据集

通过流量捕获模型, 共采集了三个来源的正常数据集:

(1) 校园网数据集: 分三个时间段采集的校园网内部数据, 分别是: ①2017 年 12 月 20 日至 2018 年 04 月 13 日; ②2019 年 03 月 18 日至 2019 年 03 月 21 日以及; ③2019 年 11 月 08 日至 2019 年 11 月 16 日的数据, 此数据集记为 Campus_normal。

(2) 企业数据集: 采集从 2019 年 8 月 1 日到 2019 年 8 月 21 日共三周的企业网络数据。并将此数据集标记为 Enterprise_normal。

(3) 学术数据集: 包括①2017 年 4 月到 2017 年 5 月期间, 从网站 <https://www.stratosphereips.org/datasets-normal> 下载的正常数据集, 标记为 CTU_normal; ②2016 年 9 月 14 日到 2016 年 9 月 26 日使用 Google 浏览器和 Chrome 浏览器访问网站 <http://betternet.lhs.inria.fr/datasets/https/index.html> 而下载的数据集, 标记为 Browser_normal。

流量预处理后, 得到正常数据集, 如表 3 所示。

4.1.2 恶意数据集

恶意数据集于 2019 年 5 月到 2019 年 8 月通过企业沙箱运行已知恶意软件而获得。每个恶意软件在沙箱中运行三分钟, 去除系统流量及沙箱通信流量等噪声, 获得原始数据; 对原始数据进行预处理后获得恶意数据集, 记为 Malware, 如表 4 所示。

表 3 正常流量数据集

Table 3 Normal traffic dataset

数据标记	会话数目
Campus_normal	521328
Enterprise_normal	359057
CTU_normal	49857
Browser_normal	474902
All_Normal	1405144

表 4 恶意流量数据集

Table 4 Malicious traffic dataset

统计	数目
沙箱数目	654
操作系统种类	7
恶意样本数	45148
恶意加密会话数	202559

4.2 评价指标

定义准确率等六个指标来估计方法的分类效果:

准确率(Acc.): $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

误报率(FPR): $FPR = \frac{FP}{FP + TN}$

漏报率(FNR): $FNR = \frac{FN}{TP + FN}$

查准率(Prec.): $Precision = \frac{TP}{TP + FN}$

查全率(Rec.): $Recall = \frac{TP}{TP + FN}$

$$F1\text{-score}(F1): F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.3 十折交叉验证

为了评估 RMETD-MF 方法的有效性, 首先进行十折交叉验证实验。所用的正常数据集为分别从三个正常数据集中选取时间上靠前的一部分加密会话混合构成, 其中从校园网数据集中选取 96277 个加密会话, 从企业数据集中选取 252467 个加密会话, 从学术数据集中选取 120724 个加密会话, 共 469468 个加密会话; 所用的恶意数据集为 5 月到 7 月在沙箱中运行的恶意软件产生的加密流量, 共 149374 个加密会话。

分别采用随机森林、逻辑回归、决策树等四种机器学习算法进行十折交叉验证, 结果如表 5 所示。

表 5 不同的机器学习分类算法实验结果

Table 5 Experimental results of different machine learning classification algorithms	
机器学习算法	交叉验证准确率(%)
L1 逻辑回归	99.69
L2 逻辑回归	86.89
决策树	99.49
随机森林	99.75

由表 5 可知, 各种机器学习算法都能达到 86.89%以上的识别准确率, 这说明 RMETD-MF 方法能够有效识别恶意加密会话和正常加密会话。其中, 分类效果最好的机器学习算法为随机森林, 因此后续实验评估选取了随机森林算法来完成。

为了分析不同特征组合对检测结果的影响, 我们使用上述实验的数据集分别测试了仅统计特征、仅握手特征、仅证书特征、仅域名特征和结合多特征识别的方法十折交叉验证结果, 如表 6 所示。

表 6 不同的特征组合十折交叉验证结果

Table 6 Ten-fold cross-validation results of different feature combinations				
特征组合	准确率	查准率	召回率	F1 值
仅统计特征	0.9985	0.9996	0.9944	0.9970
仅握手特征	0.9886	0.9675	0.9856	0.9765
仅证书特征	0.7941	0.9958	0.1466	0.2556
仅域名特征	0.9243	0.8407	0.8437	0.8437
全部特征组合	0.9997	0.9996	0.9991	0.9994

从表 6 中可以看出, 使用全部特征时, 恶意会话检测效果最好, 准确率达到 99.97%, 查准率达到 99.96%, 召回率达到 99.91%, F1 值达到 99.94%。同

时可以看出统计特征和握手特征的分类效果最好, 而仅证书特征时的分类效果最差。

文献[11]也采用了基于多种特征组合的方法, 但与 RMETD-MF 方法不同的是, 文献[11]未考虑会话中包大小和间隔时间的分布特征、域名特征、部分证书特征以及服务器端 TLS 的相关特征, 且文献[11]采用的是 L1 逻辑回归算法。分别使用两种方法对不同场景下的正常数据集和恶意数据集进行十折交叉验证准确率结果如表 7 所示, 可知, 两种方法都能获得 99.5%以上的准确率, 当混合所有正常数据集时, RMETD-MF 方法仍能保持 99.96%的准确性, 这说明 RMETD-MF 方法与文献[11]方法的识别效果相当。

表 7 两种方法在不同数据集上交叉验证结果

Table 7 Cross-validation results of two methods on different datasets			
数据集	恶意数据集	文献[11]方法(%)	RMETD-MF(%)
Campus_normal	Malware	99.79	99.97
Enterprise_normal	Malware	98.97	99.93
CTU_normal	Malware	99.98	99.98
Browser_normal	Malware	99.99	99.99
All_normal	Malware	99.58	99.96

4.4 鲁棒性测试

鲁棒性是方法实际应用的前提, 为了验证方法的鲁棒性, 设计了时间变化和场景变化对评价指标的影响。

(1) 时间变化测试

我们根据时间顺序将正常数据集和恶意数据集划为训练集和测试集。选取 2019 年 5 月~2019 年 7 月中旬的恶意流量、2017 年 12 月~2018 年 2 月的校园网流量及 2019 年 8 月前两周的企业流量数据作为训练集; 选取 2019 年 7 月下旬~2019 年 8 月的恶意流量、2019 年 3 月和 2019 年 11 月的校园网流量、以及企业 2019 年 8 月第三周数据作为测试集。

第一组实验将校园网训练集与恶意流量训练集组合进行训练, 验证在恶意流量测试集和校园网测试集上的效果, 记为时间测试 1; 第二组实验将企业训练集与恶意流量训练集组合进行训练, 验证在恶意流量测试集和企业测试集上的结果, 记为时间测试 2; 第三组实验选取多个正常场景下(校园网, 企业, 浏览器)时间靠前的流量和恶意训练集组合作为训练集, 验证相应场景下时间靠后的流量上的结果, 记为时间测试 3。测试结果如表 8 所示。

从表 8 的实验结果可以看出, 随着时间的推移, RMETD-MF 方法与文献[11]相比可保持更好的分类

效果, 分类准确率均可达到 98%以上。也可以看出, RMETD-MF 方法在包含多种不同场景的数据集时,

仍可以保持 99.9%的检测准确性, 且误报率和漏报率比文献[11]更低。

表 8 时间变化的测试结果
Table 8 Model test results over time

测试	时间测试 1		时间测试 2		时间测试 3	
	文献[11]方法	RMETD-MF	文献[11]方法	RMETD-MF	文献[11]方法	RMETD-MF
Acc.	0.9771	0.9854	0.9904	0.9991	0.9915	0.9990
FPR	0.0294	0.0188	0.0096	0.0010	0.0082	0.0007
FNR	0.0010	0.0006	0.0095	0.0007	0.0096	0.0024
Prec.	0.9103	0.9408	0.9810	0.9981	0.9573	0.9961
Rec.	0.9990	0.9993	0.9905	0.9982	0.9904	0.9976
F1-score	0.9526	0.9692	0.9857	0.9981	0.9736	0.9969

(2) 场景变化测试

为了测试 RMETD-MF 方法得到模型的适用范围, 我们测试了不同场景下的准确率、误报率、F1-score 值等表征有效性的指标。第一组实验测试选取 70%的校园网正常数据集与 70%的恶意数据集进行训练, 验证在企业正常数据集与 30%恶意数据集上的检测效果, 记为跨场景测试 1; 第二组实验选取

70%的企业正常数据集和 70%的恶意数据集进行训练, 测试在校园网数据集与剩余的 30%的恶意数据集上的检测效果, 记为跨场景测试 2; 由于数据集不相交, 为了降低误报率, 从测试数据集中随机划分 10%的正常数据集作为训练集的补充, 分别测试了随着补充的会话数增加对准确率等指标的影响, 结果如表 9、图 13 和图 14 所示。

表 9 跨场景稳定性测试
Table 9 Cross-scenario stability test

补充会话数	指标	跨场景测试 1		跨场景测试 2	
		文献[11]方法	RMETD-MF	文献[11]方法	RMETD-MF
0	Acc.	0.7964	0.7878	0.9074	0.3976
	FPR	0.2414	0.2519	0.1028	0.6785
	FNR	0.0014	0.0004	0.0118	0.0014
	Prec.	0.4374	0.4272	0.5492	0.1572
	Rec.	0.9985	0.9995	0.9881	0.9985
	F1-score	0.6083	0.5986	0.7060	0.2717
2000	Acc.	0.9128	0.9936	0.9809	0.9987
	FPR	0.0404	0.0073	0.0199	0.0012
	FNR	0.0021	0.0006	0.0120	0.0016
	Prec.	0.6452	0.9623	0.8629	0.9905
	Rec.	0.9978	0.9993	0.9879	0.9983
	F1-score	0.7837	0.9804	0.9212	0.9944
20000	Acc.	0.9653	—	0.9923	—
	FPR	0.0404	—	0.0069	—
	FNR	0.0041	—	0.0133	—
	Prec.	0.8224	—	0.9475	—
	Rec.	0.9958	—	0.9866	—
	F1-score	0.9008	—	0.9667	—

(其中“—”表示无限趋近于 1)

从表 9 可以看出, 当不补充会话数时, 即使用完全不同的数据集进行训练和测试时, 检测的

精度会下降很多, 特别是误报率很高, 这是由于不同场景下的正常流量特征相差较大而造成的。

由图 13 和图 14 可以看出, 随着测试场景下收集的会话数补充入训练集, 检测精度提高, 误报率降低, 整体性能提升。结合表 9 数据, 当补充了 2000 个会话时, 在跨场景测试 1 中, RMETD-MF 方法的召回率即可达到 98% 以上, 跨场景测试 2 中, 召回率可达到 99.4% 以上, 而文献[11]的方法召回率仅提升到 78.4% 和 92.1%; 只有在补充了 35000 个会话后, 第一种跨场景测试的召回率才

达到 93.1%, 第二种跨场景测试在补充了 20000 个会话后, 召回率才达到 96.6% 以上, 这是因为 RMETD-MF 方法考虑了会话中包顺序和大小分布特征及统计特征, 并加入了服务器 TLS 协议特征, 因此, 在补充少量会话包后, 能提取相关时间推移和场景变化的特征, 从而快速提高识别准确性。这也说明 RMETD-MF 方法在对场景变化情况下的鲁棒性要比文献[11]好。

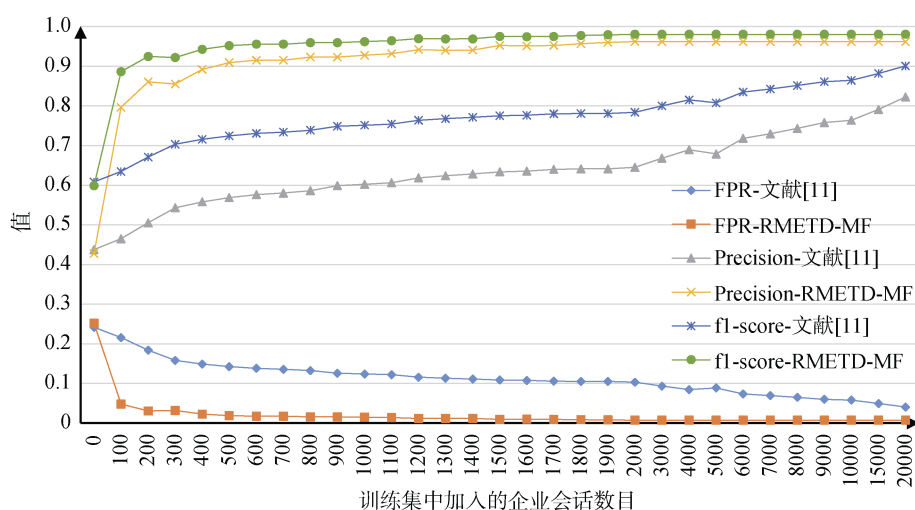


图 13 跨场景测试 1 改进测试

Figure 13 Improved test results for cross-scenario test 1

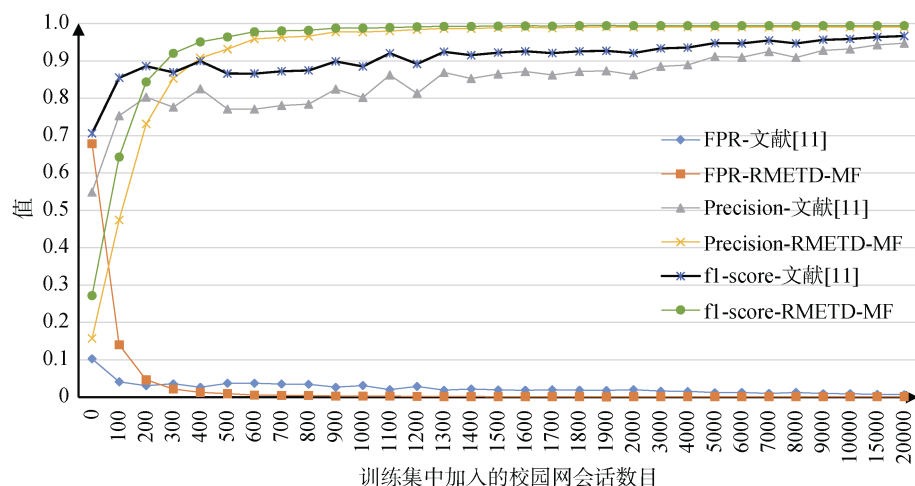


图 14 跨场景测试 2 改进测试

Figure 14 Improved test results for cross-scenario test 2

5 总结

本文提出了一种在不对加密流量做解密的情况下结合多种特征识别的恶意流量检测的方法 RMETD-MF。首先通过分析大量的正常和恶意加密流量, 从中提取出具有区分度的统计特征、TLS 握手特征、证书特征和域名特征。然后使用随机森林对

其进行训练, 利用训练的模型来对恶意流量进行检测。十折交叉验证结果表明, RMETD-MF 方法可达到 97.7% 以上的分类准确性及 99.8% 以上的识别效果。同时, 通过时间变化的实验和场景变化的测试实验, 可以看出 RMETD-MF 方法与文献[11]的方法相比, 具有更好的鲁棒性, 并在付出较少的代价(补充较少的会话信息)情况下, 能应对流量的时间变化,

能识别不同场景下的恶意流量。

由于目前仍然是使用 TLS1.2 协议作为流量加密的传输协议, 因此, 本研究是基于 TLS1.2 的分析与测试, 随着 TLS1.3 标准的推出与推广, 后续需要加强针对 TLS1.3 协议的加密流量检测方法研究。

致 谢 感谢国家自然科学基金项目 (No. 61772559、No. 61672543)、中南大学研究生科研创新项目 (No. 1053320183917) 的资助。

参考文献

- [1] CISCO. Encrypted Traffic Analytics [R/OL] (2019)
- [2] Perdisci R, Lee W, Feamster N. Behavioral Clustering of HTTP-based Malware and Signature Generation Using Malicious Network Traces[C]. *The 7th USENIX Symposium on Networked Systems Design and Implementation*. 2010: 391-404.
- [3] Griffin K, Schneider S, Hu X, et al. Automatic Generation of String Signatures for Malware Detection[M]. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 101-120.
- [4] Yao H Y, Ranjan G, Tongaonkar A, et al. SAMPLES: Self Adaptive Mining of Persistent LEXical Snippets for Classifying Mobile Application Traffic[C]. *The 21st Annual International Conference on Mobile Computing and Networking*, 2015: 439-451.
- [5] Wang S S, Yan Q B, Chen Z X, et al. Detecting Android Malware Leveraging Text Semantics of Network Flows[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(5): 1096-1109.
- [6] Wang S, Yan Q, Chen Z, et al. TextDroid: Semantics-based detection of mobile malware using network flows[C]. *Computer Communications Workshops (INFOCOM WKSHPS)*, 2017 IEEE Conference on. IEEE, 2017: 18-23.
- [7] Gu G F, Perdisci R, Zhang J J, et al. BotMiner: Clustering Analysis of Network Traffic for Protocol- And Structure-independent Botnet Detection[C]. *CCS'08*, 2008: 139-154.
- [8] Arora A, Garg S, Peddoju S K. Malware detection using network traffic analysis in android based mobile devices[C]. *Next generation mobile apps, services and technologies, 2014 eighth international conference on. IEEE*, 2014: 66-71.
- [9] Taylor V F, Spolaor R, Conti M, et al. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic[C]. *2016 IEEE European Symposium on. IEEE*, 2016: 439-454.
- [10] Saltaformaggio B, Choi H, Johnson K, et al. Eavesdropping on Fine-Grained User Activities Within Smartphone Apps Over Encrypted Network Traffic[C]. *10th USENIX Workshop on Offensive Technologies*, 2016.
- [11] Anderson B, Paul S, McGrew D. Deciphering Malware's Use of TLS (without Decryption)[J]. *Journal of Computer Virology and Hacking Techniques*, 2018, 14(3): 195-211.
- [12] Anderson B, McGrew D. Identifying Encrypted Malware Traffic with Contextual Flow Data[C]. *The 2016 ACM Workshop on Artificial Intelligence and Security*, 2016: 35-46.
- [13] Anderson B, McGrew D. Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-Stationarity[C]. *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017: 1723-1732.
- [14] Muehlstein J, Zion Y, Bahumi M, et al. Analyzing HTTPS Encrypted Traffic to Identify User Operating System, Browser and Application[EB/OL]. 2016: arXiv:1603.04865[cs.CR]. <https://arxiv.org/abs/1603.04865>
- [15] Korczyński M, Duda A. Markov Chain Fingerprinting to Classify Encrypted Traffic[C]. *Infocom, 2014 Proceedings IEEE*, 2014: 781-789.
- [16] Shen M, Wei M, Zhu L, et al. Certificate-aware encrypted traffic classification using second-order Markov chain[C]. *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on. IEEE*, 2016: 1-10.
- [17] Shen M, Wei M W, Zhu L H, et al. Classification of Encrypted Traffic with Second-Order Markov Chains and Application Attribute Bigrams[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(8): 1830-1843.
- [18] Torroledo I, Camacho L D, Bahnsen A C. Hunting Malicious TLS Certificates with Deep Neural Networks[C]. *The 11th ACM Workshop on Artificial Intelligence and Security*, 2018: 64-73.
- [19] Lotfollahi M, Jafari Siavoshani M, Shirali Hossein Zade R, et al. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning[J]. *Soft Computing*, 2020, 24(3): 1999-2012.
- [20] Rimmer V, Preuveneers D, Juarez M, et al. Automated Website Fingerprinting through Deep Learning[C]. *The 2018 Network and Distributed System Security Symposium*, 2018.



李慧慧 于 2017 年在中南大学信息安全专业获得学士学位。现在中南大学计算机技术专业攻读硕士学位。研究领域为网络安全。
Email: huihuili@csu.edu.cn



张士庚 于 2010 年在南京大学计算机科学与技术专业获得博士学位。现任中南大学计算机学院副教授。研究领域为物联网、强化学习、信息安全、深度学习。研究兴趣包括物联网、云计算、移动计算、物联网安全、无线射频(RFID)系统、无线网络定位等。
Email: sgzhang@csu.edu.cn



宋虹 于 2010 年在中南大学计算机应用技术专业获得博士学位。现任中南大学计算机学院副教授。研究领域为匿名通信, 透明计算, 信息安全。研究兴趣包括网络安全、透明计算等。Email: songhong@csu.edu.cn



王伟平 于 2004 年在中南大学计算机应用技术专业获得博士学位。现任中南大学计算机学院教授。研究领域为移动互联网安全、软件安全、恶意代码分析与识别。研究兴趣包括 Web 安全、移动终端安全、物联网安全、安全大数据分析、网络协议分析与行为监测等。Email: wpwang@csu.edu.cn