软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(3):604−617 [doi: 10.3724/SP.J.1001.2013.04243]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

抗噪的未知应用层协议报文格式最佳分段方法*

黎 敏, 余顺争

(中山大学 电子与通信工程系,广东 广州 510006)

通讯作者: 余顺争, E-mail: syu@mail.sysu.edu.cn

摘 要: 为了自动解析未知应用层协议的报文格式,提出一种未知应用层协议报文格式的最佳分段方法.这种方法不需要关于未知应用层协议的先验知识.它首先建立一种用于最佳分段的隐半马尔可夫模型(HSMM),并利用未知应用层协议在网络会话过程中传输的报文序列样本集来估计该模型的参数;再通过基于 HSMM 的最大似然概率分段方法,对报文中的各个字段进行最佳划分,同时获取代表各个字段语义的关键词.这种方法并不要求训练集绝对纯净.它能够基于观测序列的似然概率分布,发现混杂在训练集中的其他协议数据(噪声)并进行有效过滤.实验结果表明,该方法能够解析文本和二进制协议的报文格式,依据关键词构建的协议识别特征有很高的准确识别率,并能有效地检测出噪声.

关键词: 应用层协议;报文格式;分段;隐半马尔可夫模型

中图法分类号: TP393 **文献标识码**: A

中文引用格式: 黎敏,余顺争.抗噪的未知应用层协议报文格式最佳分段方法.软件学报,2013,24(3):604−617. http://www.jos.org.cn/1000-9825/4243.htm

英文引用格式: Li M, Yu SZ. Noise-Tolerant and optimal segmentation of message formats for unknown application-layer protocols. Ruanjian Xuebao/Journal of Software, 2013,24(3):604−617 (in Chinese). http://www.jos.org.cn/1000-9825/4243.htm

Noise-Tolerant and Optimal Segmentation of Message Formats for Unknown Application-Layer Protocols

LI Min, YU Shun-Zheng

(Department of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou 510006, China)

Corresponding author: YU Shun-Zheng, E-mail: syu@mail.sysu.edu.cn

Abstract: In order to automatically parse message formats of unknown application-layer protocols, this paper proposes an approach to optimally segment the message formats without a priori knowledge. A hidden semi-Markov model (HSMM) is established for the segmentation and its parameters are estimated from a set of message sequences collected from application sessions. By using the estimated HSMM in the maximum most likely segmentation, a message can be optimally divided into segments and keywords that provide semantic information about the segments can be extracted. This approach does not require the training set to be absolutely pure. The noise mixed in the training set can be filtered out based on its likelihood fitting to the HSMM. The experiments conducted in this paper show that the approach is suited to both text and binary protocols. The application-layer signatures constructed from the extracted keywords are highly accurate in identifying the protocols. The noise mixed in the training set can be efficiently detected and automatically filtered out.

Key words: application-layer protocol; message format; segmentation; hidden semi Markov model

近年来,媒体点播、网络电话、网上购物等新兴应用在人们的日常网络活动中所占的比例越来越大,新业务和新应用会使现有的网络环境发生变化,进而影响现有业务和应用的质量和性能.与此同时,针对应用层的新

* 基金项目: 国家自然科学基金(60970146); 国家高技术研究发展计划(863)(2007AA01Z449); 国家自然科学基金-广东联合基金(U0735002)

收稿时间: 2011-08-11; 定稿时间: 2012-04-09

型攻击也在不断增加,使得当前基于网络层和传输层的网络安全设备无法有效检测和防御.现有的协议分析工具,例如 Wireshark[1],只能分析已知的应用层协议.对于许多新型的应用,由于其协议规范一般是不公开的,已有的协议分析工具无法识别和分析,它们只能被统一标识为未知流量.在缺乏协议文档的情况下,通过人工分析的方法对未知协议进行反向解析[2],通常要耗费很多时间,而且容易出现错误.为了提高网络管理的效率和对新型攻击的反应速度,需要有新的方法自动识别和分析未知的应用层协议,进而做出相应的控制.

对应用层协议的分析研究开始于如何有效地识别网络中存在的各种应用层协议.Moore[3]给出了网络流在网络层和传输层的 249 种测度作为流量识别的特征.Nguyen[4]总结了多种基于流测度的机器学习方法.基于流测度的方法只能将网络流粗略的分成几大类,不能区分同一类中的不同协议.这一局限性促使许多研究关注应用层载荷(payload)的识别特征.针对应用层载荷的前 n 个字节,Haffner[5]和 Ma[6]利用不同的统计学方法来获取应用层协议的识别特征.赵咏[7]根据网络流量中文本内容的语义特点,将应用层载荷前 n 个字节分解成不同的语义单位,通过聚类的方法得到协议的识别模式.刘兴彬[8]利用数据挖掘的方法在同一协议的多个会话样本中提取频繁项,然后通过一系列的过滤规则得到最终的识别特征.

对于未知应用层协议分析来说,准确地从网络流量中找出未知应用仅仅是第 1 步,许多网络管理和安全应用需要对未知协议有更深入的了解.例如对于执行深度包检测(DPI)的安全系统[9],需要完整的协议规范作为输入.而要实现协议仿真和协议漏洞测试,则必须了解报文中各个字段的语法和语义规则.所以,自动的协议逆向工程就变得非常重要,其目的是要在无需了解协议规范的前提下,重构协议和应用的报文格式,以及推导出描述各种类型报文发送顺序的状态机.目前,国内外对网络协议逆向工程的研究根据分析数据的类型可划分为两个方向:基于服务器程序对报文的处理信息和基于在网络中传输的流量数据.

基于服务器处理信息的方法是在二进制代码层跟踪服务器程序处理报文各个字段所调用的不同指令和堆栈信息,进而推导各个字段的属性值和语法语义[10,11].在此基础上,文献[12]通过缓存的解密信息来处理加密的协议.文献[13]依据跟踪过程中程序处理信息的统计特征对报文进行聚类,最终解析出客户端和服务器的报文交互流程.依赖于从服务器程序运行过程中获得的额外信息,这类方法可以识别出报文中更多的字段,以及提取出这些字段更多的语法和语义信息,甚至可以处理加密协议.但是在实际环境中,要获取服务器程序并设置相应的网络环境来监视其运行过程并不容易实现,这限制了此类方法只能适用于某些特定的协议和应用环境.因此,将网络流量作为分析数据是适用范围更广的做法.

基于网络流量的协议逆向分析通过比对多个相同类型的报文来推导报文格式.这类方法首先要解决如何将样本集中不同的报文聚成不同的类,使得每一类中只包含一种类型的报文,从而避免得到混淆不清的报文格式.其次则是要考虑如何把报文划分为不同的字段,以及如何比对同一字段在多个报文中的取值来获取该字段的语法和语义.项目 Protocol Informatics Project[14]采用了基于字节的序列比对方法,这种方法适用于各字段都是固定长度的报文.如果报文中存在可变长度的字段,会对比对的结果有很大的影响.李伟明[15]把各个会话具有相同序号的报文看作是具有相同报文格式的报文组,然后依据简单的规则从报文组中过滤不符合要求的报文.该文献着重于报文组的多序列比对方法,以及如何将分析得到的结果应用于协议的漏洞测试,并没有详细介绍报文聚类的过程.

Cui[16]提出基于报文中文本和二进制字符的区别,根据一些常用的分界符(例如空格〈SP〉和回车换行‘/r/n’等)将文本内容分解成不同的语法单位(Token),对二进制内容则是每个字节划分为一个 Token,从而将各个报文映射成 Token 序列.然后根据这些 Token 序列的不同分布模式将报文划分为不同的类,每一类的 Token 序列代表了一种报文格式.基于 Token 的比对减小了可变长度字段的影响,但是这种预定义的字段划分方法存在过分类的问题.例如,对于 HTTP 协议报文中的日期头部‘Date: Wed, 23 Mar 2011 09:26:34 GMT\r\n’,在协议规范中是以分隔符‘:〈SP〉’将其划分为‘Date’和‘HTTP-date’两个字段,但是在文献[16]中则会被空格符划分为多个 Token.过于细化的划分会将相同类型的报文划分为多个子类,造成了大量的格式冗余,不利于下一步的协议逆向工作,需要在后期对多个报文格式进行合并.

一般来说,绝大多数的网络协议都会在报文格式中定义一个或多个协议控制字段来标识报文的类型和传

递相关的控制信息.文献[10]将这类字段定义为协议的关键词,并认为基于关键词可以有效地区分报文中协议的控制信息和用户数据.通常,在公开的协议文档中会对协议的关键词集合进行详细的定义.协议的关键词可以是协议的名称和版本号,也可以是协议的各种命令和响应码,其代表了协议的不同功能函数,适合于描述报文的类型.

如果将关键词作为报文字段划分的基础,把前后两个关键词之间的载荷内容划分为同一个变量字段,那么整个报文格式可以简化为"关键词+变量字段"的分段形式.这样的分段方法将协议规范中的多个字段合并成同一个分段,可以看成是一种层次化的划分方式.文献[11]在服务器程序的处理信息中发现报文的各个字段存在着层次结构,处于相同执行上下文的多个字段可以被划分为同一个分段.而从其实验结果看来,各分段之间通常都是以关键词作为边界.如果能够基于简化的报文格式自动提取报文中所包含的关键词,就可以利用报文所含的关键词集合来代表报文的类型,从而实现简单快捷的报文分类.在此基础上,可以对相同类型的报文作进一步的比对得到更为精确的字段格式,以及推导协议的状态机.

此外,另一个要解决的关键问题是如何消除噪声干扰对最终结果的影响.对于未知应用层协议,我们无法得知其准确的应用层识别特征.无论是基于流测度的流量分类方法,还是基于应用层载荷的数据挖掘方法,都无法保证百分之百的准确率,导致从网络流量中过滤得到的目标协议数据会混入其他协议的流量,即会引入一定比例的噪声.这会对最终的分析结果造成一定程度的偏差.例如,文献[8]的实验结果表明,噪声干扰会严重影响识别特征的准确性.为了在实际网络环境中取得理想的结果,分析算法必须能自动区分可能存在的噪声.

本文提出一种未知应用层协议报文格式的分段方法.在不需要协议先验知识的前提下,将包含了关键词的分段看作应用层报文的语义单位,通过建立隐半马尔可夫模型(HSMM)[17,18]来挖掘其中的最大似然概率的分段模式,并提取代表各个分段语义的关键词,同时利用观测序列相对于模型的似然概率来区分协议数据和噪声.

本文的第 1 节描述基于应用层载荷的 HSMM 建模过程.第 2 节介绍基于 HSMM 的关键词提取和噪声检测算法.第 3 节给出实验结果并进行分析.最后对本文进行总结并讨论下一步工作.

1 基于应用层载荷的 HSMM

应用层协议的通信是按照协议规范在网络中传输的一系列报文的交互过程.报文是应用层数据单元,一个报文可能封装在一个或多个数据包中进行传输.从网络上来看,一个应用层报文是由同一方向连续传输的数据包的载荷所组成.应用层会话一般分为不同的阶段,如连接建立、传输参数设置、数据交换和连接拆除等,在不同的阶段需要发送不同类型的报文来实现应用层协议的各种功能.在整个会话过程中,通信两端的应用程序解析不同类型的报文,保证整个会话过程的正常运行.由于我们是要对整个协议进行逆向分析,所以将以完整的应用层载荷作为分析对象,而不只是研究应用层会话的前几个报文,或者是应用层载荷的前 n 个字节.

包括应用层协议在内的 Internet 各层协议的报文格式,常见的有:(1) 二进制形式,即把若干比特(固定长)作为一个字段(field),每个字段代表一种属性,字段内比特的取值就是该属性的值.例如 IP 头部的前 4 个比特代表 IP 协议版本号,取值 4 就是 IPv4,取值 6 就是 IPv6;(2) TLV 形式,即 type-length-value 形式,其中,固定字节长度的 type 代表属性类型,固定字节长度的 length 表明后面跟的属性值 value 的字节数,可变字节数的 value 代表属性值,例如 IP 头部的选项部分;(3) ASCII 形式,即用特定的词表示字段的语义,这些词由 ASCII 字符组成,易于理解,通常为协议的命令或状态码.其后紧跟的字符串是内容,并以预定义的分界符(例如空格符或者回车换行符)作为该字段的结尾.例如 HTTP 协议的 GET,HEAD,POST 是命令,200 OK 是状态码;(4) 指针形式,即用 pointer 指出一个字段的开始或者结束位置,例如 DNS 的报文格式.

在上述 4 种形式的报文格式中,有的字段是固定长的,有的字段是可变长的.但无论是哪种形式,只要字段内出现某种固定模式的比特串或者字符串,这些固定模式的串都会在样本集中频繁出现,因而可以被挖掘出来.本文把这些固定模式的串统称为关键词.

关键问题是,对于未知协议,事先无法知道其报文格式是上述 4 种形式中的哪种形式,也不知道其字段是固定长的还是变长的,因而也不知道字段之间的分界(或分界符).因此,需要一种适用于上述 4 种格式的、不借助任

何先验知识的、通用的最佳分段方法.

假定报文格式是 $C_1||C_2||...||C_n$ 的形式,如图 1 所示.其中,C_i 代表一个定长或变长的字段,"||"代表串行拼接.C_i 本身又可以细分为 $key_i||data_i$ 的形式,key_i 是该字段的首部,代表应用层协议的一个关键词,可以是上述的 field、type、命令、状态码或 pointer 等;$data_i$ 是该字段的剩余部分,长度可以为 0.在需要的情况下,$data_i$ 还可以进一步细分为 $length_i||value_i$.于是,报文格式分析问题就变成了对报文中的字符串或比特串进行 $C_1,C_2,...,C_n$ 最佳分段的问题,以及对 C_i 进行 key_i 和 $data_i$ 划分的问题.至于 $data_i$ 如何进一步细分为 $length_i||value_i$,本文不做进一步讨论,因为 $length_i$ 的值等于 $value_i$ 的长度,$pointer_i$ 的值等于 $value_i$ 的位置,都可以通过简单验算来确定.

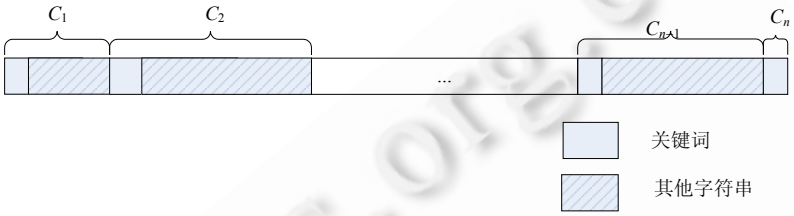

Fig.1 Segmentation of message format

图 1 报文格式的段划分

下面我们将研究如何采用最大似然概率作为目标函数,通过对报文的最大似然概率的分段完成对报文的解析.实际上,隐半马尔可夫模型的最大似然概率状态序列估计方法,就是一种非常适合于最佳分段的方法,其中每个隐状态都代表一种或几种字段结构,状态的持续长度间代表该字段的长度,一个状态向另一个状态的转移代表了从一个字段到另外一个字段的解析过程.

将应用层报文按顺序重组之后,可以得到完整的应用层载荷,称为观测序列.为了叙述方便,本文将应用层载荷看成是一个字节流,因此,观测序列是一个字符串.本文的方法同样适合于以比特或者几个比特(例如 4 个比特)为最小单位的串,因而可以用于分析二进制形式的报文格式.

将观测序列标记为 $O_{1:T}=o_1o_2o_3o_4...o_T$,$T$ 为观测序列的长度,即字符个数,其中,o_t 为观测序列的第 t 个字符.状态集合定义为 $S=\{1,2,...,M\}$.将观测序列 $O_{1:T}$ 对应的状态序列记为 $S_{1:T}=S_1S_2S_3S_4...S_T,S_t\in S$.用 $S_{[t:t']}$ 表示从 t 开始到 t' 结束的一个状态,$S_{1:T}$ 也可以用序列$(i_1,d_1),(i_2,d_2),...,(i_n,d_n)$表示,其中,$i_m$ 是状态,d_m 是状态 i_m 的持续长度,$S_{[1:d_1]}=i_1,S_{[d_1+1:d_1+d_2]}=i_2,...,\ i_1,i_2,...,i_n\in S$ 且满足 $\sum_{m=1}^{n}d_m=T$.对观测序列的段划分,就是求最大似然概率的状态序列 $i_1,i_2,...,i_n$.每个状态对应一个字段,状态的持续长度等于字段的长度,即各个字段的长度依次为 $d_1,d_2,...,d_n$.字段 C_m 对应于状态 i_m 和长度 d_m,从其起始位置开始的一个子字符串是 key_m,如图 2 所示.一个状态可以对应于多个不同的字段模式,每个字段模式有一个给定的 key 和 key 后跟随的一个可变长度与可变内容的 $data$ 部分.所以,可以把 d_m 和 key_m 都看作是给定状态 i_m 的输出值或观测值.必要的情况下,也可以把 $data_m$ 的属性(例如是二进制还是 ASCII 文本)看作给定状态 i_m 的输出值.由于在进行最佳分段和确定字段模式之前,不知道一个字段的起始位置、起始的 key 和字段长度,也不知道所对应的状态,所以由观测序列不能直接对应得到状态序列,因而状态序列是不可观测的,即隐性存在的.正因为如此,这里的状态也称为隐状态.

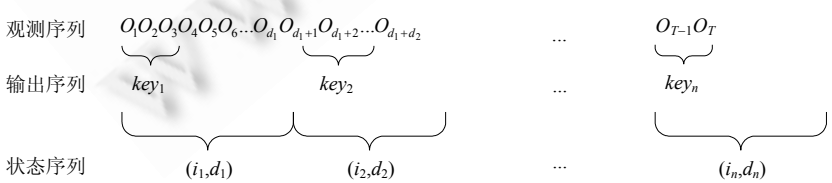

Fig.2 Framework of HSMM

图 2 HSMM 模型图

考虑到报文之间的关系以及报文中字段之间的关系代表了协议状态之间的转移关系,可以假设隐状态之间的跳转是一阶的马尔可夫过程.即,设状态 i 转移到状态 j 的概率为 a_{ij},满足 $\sum_j a_{ij}=1$ 且 $a_{ii}=0$.状态 i 的初始分布概率定义为 π_i.由于隐状态是无法观测到的,对于给定的观测序列,存在多种可能的段划分方式.同一个字段也可能选择不同的子字符串作为该字段的 *key*.所以,观测序列的所有子字符串都是一个状态的可能的输出值或观测值 *key*.定义状态的所有输出值 *key* 的集合为 *KEY*,在给定状态 j 的情况下以 *key* 为输出值的概率定义为 $k_j(key)$,满足 $\sum_{key\in KEY} k_j(key)=1$.

对于不同的 *key*,其后续的 *data* 长度一般会有不同的概率分布.例如在 SMTP 的协议规范[19]中,关键词 MAIL 和 RCPT 的 *data* 是邮箱地址,一般不会超过 30 个字符.而关键词 DATA 后面的 *data* 是邮件正文,其长度分布范围较为分散,从几百到几千个字符都有.所以,字段长度即状态持续长度不仅与状态有关,还与该分段所包含的 *key* 有关.在给定状态 j 和其输出值 $key\in KEY$ 的情况下,状态的持续长度为 d 的概率为 $l_{j,key}(d)$,满足 $\sum_d l_{j,key}(d)=1$,其中,$|key|\leqslant d\leqslant D_{\max}$,$D_{\max}$ 是状态的最大持续长度,$|key|$是 *key* 的长度.

因此,HSMM 的模型参数可以用 $\lambda=\{a_{ij},\pi_i,k_j(key),l_{j,key}(d),i,j\in S,key\in KEY\}$ 表示.给定训练集合,就可以利用 HSMM 的前后向算法完成模型参数 λ 的估计.

1.1 模型假设

在训练模型之前,我们首先依据应用层协议的特点对模型引入 3 个假设,使之更符合实际情况,同时也能大幅度地降低运算量.

假设 1:*key* 的长度有一定的限制.例如,FTP 协议[20]的命令和响应码的长度不超过 4 个字节.因此,我们引入了 *key* 的最大长度 $KL_{\max}$.一般来说,$KL_{\max}$ 取 10 个字节就足够了.当 *key* 的长度超过 $KL_{\max}$ 时,最后的结果会把该 *key* 看作多个长度小于 $KL_{\max}$ 的 key_x,但这些 key_x 总是与概率 1 同时出现.所以,只需要简单的一步,就可以把这些 key_x 合成一个长度大于 $KL_{\max}$ 的 *key*.实验结果表明,这种假设不会影响最终的分析结果.

假设 2:观测序列中的各个字段都不能跨越报文边界.相邻的两个报文由不同的方向发送,代表了协议不同的功能模块.所以,对于报文中出现的字段,其最大长度不能超过该报文的长度.如图 3 所示,某个应用层会话包含双向传输的多个报文 M_i,各个报文的起始位置为 $br(M_i)$,是已知的值.假设观测序列中的第 t 个字符属于报文 M_i,而且 t 是当前状态的结束位置,那么当前状态的最大持续长度为 $D_t=t-br(M_i)+1$.如果 t 不在报文末尾,则下一状态的最大持续长度为 $E_t=br(M_{i+1})-1-t$.如果 t 在报文末尾,则 $E_t=br(M_{i+2})-br(M_{i+1})$,即下一状态的最大持续长度为下一个报文的长度.

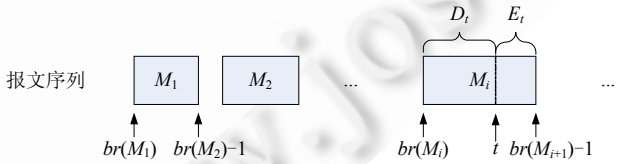

Fig.3 Maximum duration of a state at different position

图 3 不同位置的状态最大持续长度

假设 3:每个 *key* 会在训练集中的多个甚至是全部的观测序列中出现.对于在训练集中出现的某个字符串 X,假定 X 的支持度为 $n(X)$,代表训练集中包含 X 的观测序列的数目.当 X 的支持度小于给定阈值时,可以认为 X 不可能是协议的 *key*.因此,对于支持度小于阈值的所有字符串,在集合 *KEY* 中统一用特定项 *non-kw* 来不加区分地表示.其结果是,当整个字段都不包含协议所规定的 *key* 时,其前导字节是一个称为 *non-kw* 的特殊的 *key*.虽然 *non-kw* 的支持度很高,但它代表的是很多字符串的集合,每个字符串出现的频次很低,因而相对于集合 *KEY* 中的其他项,$k_j(non_kw)$在参数初始化和模型训练中会取得比较小的值.通过支持度的定义,可以精简待选字符串的个数,减少内存消耗和运算量.例如,令 $n(X)\geqslant 10$,即要求集合 *KEY* 中的各个元素至少在 10 个观测序列中出现.

也可以把 $n(X)/n$ 作为选择的依据,其中,n 是观测序列的总个数.例如,对于单字符,如果它是关键词,则其出现的频率应该远远大于 1/256,即把 $n(X)/n>>1/256$ 作为选取的标准.

1.2 模型的初始参数设置

模型的状态数目 M 决定了模型的结构.由于对目标协议没有任何的先验知识,我们只能随机选择一个不大的状态数目(例如 $M<10$),并通过增加或减少状态数目来训练不同的模型,最后比较不同模型下的实验结果来选择最优的状态数目.在实验部分我们发现,状态的数目对最终结果的影响不大.例如,我们选择 $M=3$ 和 $M=10$ 得到的结果基本上是一样的.但算法的运算时间和所需的内存相差比较大.

HSMM 模型的训练是一个求局部最优解的过程,模型的初始参数将会影响最终的训练结果.一般来说,a_{ij} 和 π_i 的初始值对最终训练得到的模型影响很小,可以简单地取等概率分布.对于多字节的协议关键词来说,其出现频率要小于其前缀.例如,对于 SMTP 协议中的关键词 EHLO,其前缀字符 E 在观测序列中出现的次数要高于 EHLO,但显然,E 不是 SMTP 协议的关键词.所以在初始化时,我们令 $k_j(key)=1/|KEY|$,$|KEY|$是 KEY 集合中的元素数目,key 是 KEY 中的任意一个元素,即所有的 key 取等概率分布.同时,设给定状态 j 和 key 的情况下,每个字段的长度 $d\geqslant|key|$的初始概率分布为 $l_{j,key}(d)=ce^{-\tau(d-|key|)}$,其中,$|key|$是 key 的长度;$d-|key|$是 key 后面所跟随的 $data$ 的长度;τ是一个待定的参数;c 是归一化因子,它使得 $\sum_d l_{j,key}(d)=1$.所以,在初始设置的时候,字段长度的概率随着 d 的增大而指数减小.另一方面,对于相同的字段长度 d,其概率大小与 key 的长度有关,即$|key|$越大,概率越大.这样的初始概率设置,使得在参数估计的初次迭代计算中,对于同一分段,会趋向于选取较长的字符串作为关键词.在实验过程中我们发现,τ的可选范围很宽,算法结果对τ的具体取值不敏感.但相比之下,选择$\tau<1$,会使得一些短字符串被当作关键词;而选择$\tau=5$ 则会取得很好的结果.需要说明的是,$l_{j,key}(d)=ce^{-\tau(d-|key|)}$只是用于对$l_{j,key}(d)$的初始值的选取,并不是假定其概率质量函数是这种指数型的或者参数化的分布.与 $k_j(key)$一样,$l_{j,key}(d)$也是一种非参数化的、离散的概率分布.

1.3 模型参数估计

用 $S_{t']}$表示到 t'结束的一个状态,$S_{[t}$表示从 t 开始的一个状态,$S_{[t:t'}$表示从 t 开始到 t'还没有结束的一个状态,$S_{t:t']}$表示在 t 已经开始到 t'结束的一个状态.对于在 t 时刻开始的状态,用 kl_t 表示其输出值 key 的长度.对于观测序列的某个段 $C_m=o_{t+1:t+d}$,隐状态输出值 key 的可能取值范围为 $\{o_{t+1:t+kl_{t+1}},1\leqslant kl_{t+1}\leqslant\min(d,KL_{\max})\}$.在给定 $S_{[t+1:t+d]}=j$ 的情况下,观测到 $o_{t+1:t+d}$ 的概率为

$$b_{j,d}(o_{t+1:t+d})=\sum_{kl_{t+1}=1}^{\min(d,KL_{\max})}k_j(o_{t+1:t+kl_{t+1}})l_{j,o_{t+1:t+kl_{t+1}}}(d) \tag{1}$$

定义前向变量$\alpha_t(j)=P[S_{t]}=j,o_{1:t}|\lambda]$,其迭代计算公式如下:

$$\alpha_t(j)=\begin{cases}\pi_j, & t=0\\ \sum_{i=1}^{M}\sum_{d=1}^{D_t}\alpha_{t-d}(i)a_{ij}b_{j,d}(o_{t-d+1:t}), & 1\leqslant t\leqslant T\end{cases} \tag{2}$$

其中,D_t是以第 t 个字符为结尾的状态 j 的最大持续长度.利用迭代得到的前向变量,我们可以计算出观测序列相对于给定模型λ的似然概率:

$$Lkh=P[O_{1:T}\mid\lambda]=\sum_{j=1}^{M}\alpha_T(j) \tag{3}$$

定义后向变量$\beta_t(i)=P[o_{t+1:T}|S_{t]}=i,\lambda]$,其迭代计算公式如下:

$$\beta_t(i)=\begin{cases}1, & t=T\\ \sum_{j=1}^{M}\sum_{d=1}^{E_t}a_{ij}b_{j,d}(o_{t+1:t+d})\beta_{t+d}(j), & 1\leqslant t<T\end{cases} \tag{4}$$

其中,E_t是以第 t+1 个字符为起始位置的状态 j 的最大持续长度.

定义中间变量:

$$\xi_t(i,j)=P[S_t=i,S_{[t+1}=j,O_{1:T}\mid\lambda]=\alpha_t(i)\sum_{d=1}^{E_t}a_{ij}b_{j,d}(o_{t+1:t+d})\beta_{t+d}(j),i\neq j,0\leqslant t\leqslant T-1 \tag{5}$$

$$\psi_t(j,len)=P[S_{[t+1:t+len}=j,kl_{t+1}=len,O_{1:T}\mid\lambda]=\sum_{i=1}^{M}\sum_{d=len}^{E_t}\alpha_t(i)a_{ij}k_j(o_{t+1:t+len})l_{j,o_{t+1:t+len}}(d)\beta_{t+d}(j),t\leqslant T-kl \tag{6}$$

$$\begin{aligned}\zeta_t(j,key,d)&=P[S_{[t-d+1:t]}=j,o_{t-d+1:t-d+|key|}=key,kl_{t-d+1}=|key|,O_{1:T}\mid\lambda]\\&=\sum_{i=1}^{M}\alpha_{t-d}(i)a_{ij}k_j(key)l_{j,key}(d)\beta_t(j)\delta(o_{t-d+1:t-d+|key|}-key),t\geqslant 1,D_t\geqslant d\geqslant|key|\geqslant 1\end{aligned} \tag{7}$$

其中,如果 $o_{t-d+1:t-d+|key|}=key$,则 $\delta(o_{t-d+1:t-d+|key|}-key)=1$;否则为 0.

我们采用多序列训练模型.假设训练集总共有 N 个观测序列,其中,$O_{1:T}^{(n)}$ 是第 n 个观测序列,$T^{(n)}$是其长度.由 $O_{1:T}^{(n)}$ 可以计算得到 $Lkh^{(n)}$, $\xi_t^{(n)}(i,j)$,$\psi_t^{(n)}(j,kl)$ 和 $\zeta_t^{(n)}(j,key,d)$.再利用下式估计模型参数:

$$\hat{a}_{ij}=\sum_{n=1}^{N}\frac{1}{Lkh^{(n)}}\sum_{t=0}^{T^{(n)}-1}\xi_t^{(n)}(i,j) \tag{8}$$

$$\hat{k}_j(key)=\sum_{n=1}^{N}\frac{1}{Lkh^{(n)}}\sum_{t=0}^{T^{(n)}-|key|}\psi_t^{(n)}(j,|key|)\cdot\delta(o_{t+1:t+|key|}^{(n)}-key) \tag{9}$$

$$\hat{l}_{j,key}(d)=\sum_{n=1}^{N}\frac{1}{Lkh^{(n)}}\sum_{t=d}^{T^{(n)}}\zeta_t^{(n)}(j,key,d) \tag{10}$$

$$\hat{\pi}_i=\sum_{n=1}^{N}\sum_{j=1}^{M}\frac{1}{Lkh^{(n)}}\xi_0^{(n)}(i,j) \tag{11}$$

最后进行归一化处理,就可以得到模型参数新的估计值 $\hat{\lambda}=(\hat{a}_{ij},\hat{\pi}_i,\hat{k}_j(key),\hat{l}_{j,key}(d))$.重复进行这种模型参数的估计过程,最终将收敛到一组固定的模型参数值.因为这种迭代估计的过程已经被证明为似然概率单调增长的过程,因此是必然会收敛的过程.根据经验,最多十几次迭代就可以以很高的精度收敛到固定的点.

2 基于 HSMM 的未知应用层协议分析

2.1 噪声区分

混杂在协议数据中的噪声是指在目标协议的会话序列样本集中存在其他协议的会话序列.由于我们事先对未知协议的特征一无所知,所以无法根据端口号、流量特征等把一种未知协议与其他协议的会话序列严格区分开来.但由于不同的应用层协议有不同的会话过程、报文结构、关键词集合、各字符(串)出现的概率分布和报文大小的分布[21],因此用以描述不同协议的模型参数必然是不同的.这使得适用于某种协议的模型参数,对于其他协议来讲,将是不适用的或者不那么适用的.

当使用混有噪声的目标协议会话序列样本集训练 HSMM 模型时,假定已经通过了某种精确的聚类分析,尽可能提纯了目标协议会话序列样本集,噪声所占的比例已经不大.这时,模型参数主要受目标协议数据的影响,而噪声数据对模型训练产生的影响很小.训练得到的模型更偏向于描述目标协议数据的统计特征,噪声序列与目标协议序列相对于模型的似然概率会有较大的不同.我们定义观测序列 O 相对于模型λ的平均对数似然概率(average log-likelihood,简称 ALH)为

$$ALH=\frac{1}{Msg(O)}\ln(P(O\mid\lambda)) \tag{12}$$

其中,$Msg(O)$是观测序列 O 的报文个数.在实验部分可以看出,目标协议数据和噪声的 ALH 会有很不同的分布,通过简单的聚类算法就可以实现对噪声的过滤.在进行初步的噪声过滤以后,可以重新进行模型参数的估计,然后用新的模型参数再进行一次噪声过滤,使得 ALH 的分布更加集中.

2.2 最大似然概率的段划分

在估计得到模型参数λ以后,我们可以开始进行最大似然概率的段划分.如前所述,对于观测序列 O,存在多种可能的段划分.对隐状态序列的最大似然概率估计,将是对观测序列的最优的段划分.因此,通过对 HSMM 的 Viterbi 算法作相应的修改,可以估计最大似然概率的状态序列;同时,由各状态确定报文所包含的各个关键词.

令 $G(j,key,d)=k_j(key)l_{j,key}(d)$,并定义前向变量

$$\begin{aligned}\delta_t(j,d) &\equiv \max_{S_{1:t-d}} P[S_{1:t-d}, S_{[t-d+1:t]}=j, o_{1:t}\mid\lambda] \\ &= \max_{i,d',kl}\delta_{t-d}(i,d')a_{ij}G(j,o_{t-d+1:t-d+kl},d), 1\leqslant t\leqslant T, 1\leqslant d\leqslant D_t, 1\leqslant d'\leqslant D_{t-d'}, 1\leqslant kl\leqslant KL_{\max}\end{aligned} \tag{13}$$

利用公式(13)完成$\delta_t(j,d)$的计算.用 $\Psi(t,j,d)$记录$\delta_t(j,d)$所选择的前一个状态及其持续长度;

同时,用 $Key_ML(t,j,d)$记录当前状态分段(j,d)所选择的关键词.

$$(i^*,d^*,kl^*)=\arg\max_{i,d',kl}\delta_{t-d}(i,d')a_{ij}G(j,o_{t-d+1:t-d+kl},d),$$

$$\Psi(t,j,d)=(t-d,i^*,d^*),$$

$$Key\_ML(t,j,d)=o_{t-d+1:t-d+kl^*}.$$

完成前向计算后,令 $t_1=T$,进行反向的状态路径回溯

$$(j_1,d_1)=\arg\max_{i,d}\delta_T(i,d),$$

$$kw_1=Key\_ML(t_1,j_1,d_1),$$

$$(t_2,j_2,d_2)=\Psi(t_1,j_1,d_1),$$

$$kw_2=Key\_ML(t_2,j_2,d_2),$$

$$\ldots,$$

$$(t_n,j_n,d_n)=\Psi(t_{n-1},j_{n-1},d_{n-1}),$$

$$kw_n=Key\_ML(t_n,j_n,d_n).$$

直到确定 $s_1=j_n$,算法结束.我们可以得到最大似然概率的状态序列$\{(j_n,d_n),(j_{n-1},d_{n-1}),\ldots,(j_1,d_1)\}$.观测序列最佳分段的各个段长度依次为 $d_n,d_{n-1},\ldots,d_1$.而观测序列所包含的最大似然概率关键序列为$\{kw_n,kw_{n-1},\ldots,kw_1\}$.

2.3 提取应用层识别特征

应用层协议的识别特征,首先必须保证既能匹配该协议的不同会话,又能区分其他协议的会话.关键词集合作为协议的常量字段,会在多个会话中频繁出现.其中,某些关键词子集或关键词子序列更是协议运行所必需的,在每一个会话中都会出现,代表了该协议最基本的会话规则.所以,协议的应用层识别特征可以由各个会话的公共关键词子集或关键词子序列组成.

对于训练集中的每一个观测序列,利用 Viterbi 算法可以提取出其最大似然概率的关键词序列,关键词序列中所有不同的关键词构成了训练集所包含的关键词集合.检查该集合中的各个元素或元素子集是否在所有的关键词序列中都出现.如果是,则将其标识为特征词.对于待识别的应用层会话,只有当应用层载荷包含了全部的特征词时,才认为是特征匹配成功.为了简单起见,本文并没有考虑特征词的时序关系.如果需要定义更严格的识别特征,可以在识别规则中加入各特征词的出现次数以及出现顺序.另外,我们可以限定在应用会话的前若干个(例如 10 个)数据包中确定特征词,从而在会话早期完成应用识别以跟踪和控制会话行为.

3 实验结果与分析

现有对于应用层载荷的分析绝大部分都是基于不公开的私有数据集,DARPA[22]数据集是目前唯一拥有完整应用层载荷的公开数据集.我们从中选取了占主要比例的浏览类协议 HTTP,文件传输类协议 FTP 和邮件类协议 SMTP.另外,我们从中山大学骨干网上捕获的流量数据中选取了邮件类协议 POP3,P2P 下载类协议 BT (BitTorrent)和网络管理协议 SMB.其中,HTTP,FTP,SMTP 和 POP3 属于文本协议,其载荷内容主要由 ASCII 字符组成.而 BT 和 SMB 则属于二进制协议,其载荷内容可以是任意的取值.

目前,基于网络流量的协议逆向分析都没有考虑噪声的影响,也没有公开的测试数据集,我们采用在目标协议纯净的样本集中依次添加不同类型和不同比例的噪声的方法,测试本文算法检测噪声的能力.

为了从网络流量中获取纯净样本集,我们首先将流量数据中的会话按照五元组(源 IP,目的 IP,源端口,目的端口和协议类型)进行分组,然后根据 L7-filter[23]的正则表达式进行识别,最后再用 Wireshark 进行辅助判断,确保每种协议的数据集中不包含其他协议的数据.

表 1 列出了各协议数据集的统计情况.其中,SMB 协议的数据集只有 300 个会话是因为在中山大学网络中使用这种协议的用户不多.

Table 1 Summary of data sets

表 1 各协议数据集的统计情况

协议	会话数	报文个数
HTTP	1 000	2 000
FTP	1 000	49 162
SMTP	1 000	17 020
POP3	1 000	13 249
SMB	300	5 955
BT	1 000	6 833

3.1 噪声检测

我们首先在FTP协议的数据集中随机选取了200个会话作为目标协议数据,然后从另外一种协议的数据集中随机选取 40 个会话作为噪声,此时的噪声比例(noise ratio,简称 NR)为 40/200=0.2.将目标协议数据和噪声混合后共同训练 HSMM 模型,混合数据中各个观测序列的 ALH 直方图分布如图 4 所示.

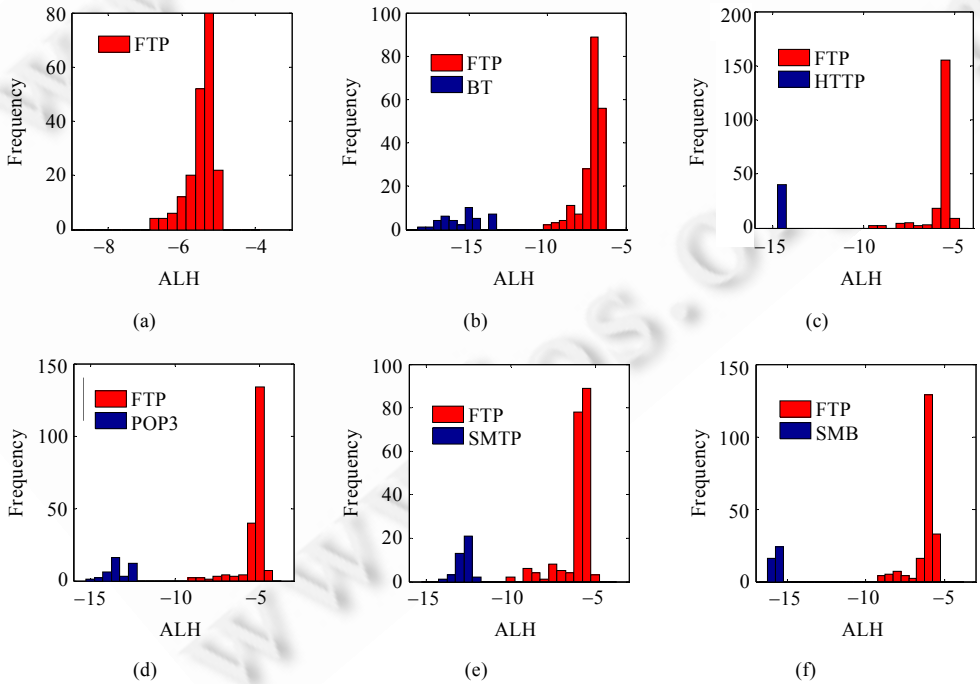

Fig.4 Distribution of ALH

图 4 平均对数似然概率分布

从图 4(a)可以看出,如果训练集中不存在噪声,目标协议数据的 ALH 有一个较为集中的分布区域.在混入噪声后,对目标协议数据的 ALH 分布有一定的影响.但是如图 4(b)~图 4(f)所示,噪声序列的 ALH 会落在不同的区

间,与目标协议数据序列的 ALH 的分布区间有一定的距离.因此,我们对训练集中各个观测序列的 *ALH* 采用欧式距离做简单的聚类运算,并将聚类的数目设为 2(分别代表目标协议和噪声).假设训练集中有 *m* 个目标协议会话和 *n* 个噪声会话,其中,被误判为噪声的目标协议会话数目为 TN(true negative),被误判为目标协议会话的噪声会话数目为 FP(false positive).我们采用两个误判率,*TNR*=*TN*/*m* 以及 *FPR*=*FP*/*n* 衡量聚类效果.

我们依次对 FTP,SMTP,HTTP,SMB 和 POP3 这 5 种协议进行噪声检测实验,并测试在不同的 *NR* 取值下的效果.从表 2 可以看出:在 *NR* 较小的情况下,FTP,SMTP 和 SMB 都能以 0 误判率区分出噪声序列,而 HTTP 和 POP3 则能以很小部分的目标协议会话被判定为噪声的代价区分出所有的噪声序列;当 *NR* 增大后,模型参数受噪声的影响也随之增大.见表 3,此时被误判为噪声的目标协议会话会增多,但所有的噪声序列都不会被误判为目标协议会话,这保证了能得到纯净的目标协议数据进行下一步的协议分析.

Table 2 TNRs and FPRs when *NR*=0.2

表 2 *NR*=0.2 时的误判率

协议	噪声类型											
	FTP		HTTP		POP3		SMTP		SMB		BT	
	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR
FTP	N/A	N/A	0	0	0	0	0	0	0	0	0	0
HTTP	0	0	N/A	N/A	0.02	0	0.05	0	0.015	0	0	0
POP3	0.015	0	0	0	N/A	N/A	0	0	0	0	0	0
SMTP	0	0	0	0	0	0	N/A	N/A	0	0	0	0
SMB	0	0	0	0	0	0	0	0	N/A	N/A	0	0

Table 3 TNRs and FPRs when *NR*=0.5

表 3 *NR*=0.5 时的误判率

协议	噪声类型											
	FTP		HTTP		POP3		SMTP		SMB		BT	
	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR	TNR	FPR
FTP	N/A	N/A	0	0	0	0	0	0	0	0	0	0
HTTP	0	0	N/A	N/A	0.06	0	0.3	0	0.05	0	0	0
POP3	0.05	0	0	0	N/A	N/A	0.04	0	0	0	0	0
SMTP	0.02	0	0	0	0	0	N/A	N/A	0	0	0	0
SMB	0	0	0	0	0	0	0	0	N/A	N/A	0	0

我们没有对 BT 协议做噪声过滤实验,这主要与 BT 协议的特殊性有关:首先,BT 协议的关键词很少,固定出现的只有前面两个握手报文中的 BitTorrent protocol 字段,此外,只有在某些可选的命令报文中用 1 个字节代表报文的类型;其次,BT 协议的一个会话所包含的报文数目是非常随机的,除了前面的两个握手报文会固定出现之外,后面的报文可以是不含任何关键词的数据交换报文,或者是可选的命令报文.这两点导致了即使是不含噪声的 BT 数据集,其 ALH 的分布依然会分散在不同的区间,这使得单纯依赖于 ALH 难以过滤 BT 会话中的噪声.

3.2 报文解析

现有的基于网络流量的应用层协议分析关注于协议逆向工程的不同方面,而且都是使用不公开的数据集,难以和本文方法进行比较.因此,我们通过人工分析将从报文中提取出来的关键词集合和公开的协议规范进行比较以检验对报文解析的准确率.

对于每一种协议,我们从其数据集中随机选取 100 个会话作为训练集.在提取出关键词集合后,将各个关键词与该协议的协议规范进行比较,发现提取出来的关键词可以分成 4 种类型:

类型 1:在协议规范中定义了的关键词;

类型 2:关键词由于输出长度不同造成的冗余,例如把‘HTTP/1.’与‘HTTP/1.1’同时作为关键词;

类型 3:单字符;

类型 4:受训练集样本的影响(例如会话序列样本取自对少数几个网站的访问),某些非协议定义的字符串会在固定的位置重复出现,例如网站的域名,其统计特征与协议定义的关键词很类似,所以也被提取成关键词.

表 4 统计了各协议提取出来的关键词的类型分布.对于 HTTP 协议,我们提取出了请求命令“GET”、协议版

本‘HTTP/1.’和响应码‘200 OK’以及‘Date:’, ‘Server:’, ‘User-Agent:’, ‘Content’, ‘Referer:’和‘Host:’共 6 个头部关键词.在其他 3 种文本协议中,我们提取了 FTP 协议的 10 种命令和 5 种响应码,POP3 协议的 7 种命令和响应码‘+OK’以及 SMTP 协议的 7 种命令和 4 种响应码.由于某些关键词后面总是紧跟着一些协议定义的分界符(例如空格符和回车换行符),这些分界符会被视为关键词的尾部一起被提取出来.二进制 BT 协议的关键词不多,我们只提取出来了‘0x13Bittoren’和‘t protocol’.由于这两个关键词在各个关键词序列中总是成对相邻出现,我们可以将其拼接成原有的关键词‘0x13Bittorent protocol’.这说明关键词的最大长度 $KL_{\max}$ 的设置并不影响最终的实验结果.对于另一个二进制协议 SMB,我们提取了 8 种不同的命令和协议的特殊标识‘SMB’.

Table 4 Different types of keywords

表 4 各协议的关键词类型

协议	类型 1	类型 2	类型 3	类型 4
FTP	15	3	12	2
HTTP	9	2	4	10
POP3	8	3	12	0
SMTP	11	1	5	12
SMB	9	4	1	6
BT	2	0	2	7

而在协议规范的定义之外,有部分的单字符会被提取成关键词.这是因为对于一个字节来说,只有 256 种可能的取值,某些单字符的出现频率很高.这造成一些多字节的关键词只提取了首字符,同时在某些非关键词部分可能会出现将单字符误判为关键词的分段.如果单字符是某个关键词的首字符,则可以简单地把该单字符排除在关键词集合之外;否则,就需要根据其他一些规则来进行过滤.例如,当大多数关键词都是多字节的字符串时,可以假定该协议的关键词都是多字节的,单字符应该排除在外.再就是在模型参数初始化时降低单字符作为关键词的概率.

由于本文算法只依赖于抓取的网络流量,实验结果与所选取的训练集有关,训练集样本中的非协议关键词内容会对实验结果有所影响.例如,在 DARPA 的网络环境中,SMTP 的服务器个数有限,而且采用基本相同的服务器软件.所以在样本集多个 SMTP 会话中,握手报文除了关键词 220 之外,还有另一个字符串 Sendmail 4.1 也会频繁出现,用以标识服务器的软件名称和版本号.这样的非协议定义的频繁字符串由于统计特性与协议关键词很类似,本文算法难以对其进行区分.而对于二进制协议 BT 和 SMB 来说,在应用层报头的预留字段或者在数据载荷部分,会因为零填充机制而包含一些全 0 的连续字节,这些全 0 的字段虽然不符合关键词的定义,但是由于频繁出现在相同的位置,也会被提取成协议的关键词.

虽然实验结果与协议规范有一定的误差,但从整体上看,提取出来的关键词大部分都是协议规范所定义的,也反映了报文中主要的协议控制信息,这表明本文算法能够对未知的应用层协议报文格式进行有效的解析.

3.3 应用层识别特征有效性检验

对于协议 A,假设测试数据集中有 n 个 A 数据,其中被正确识别为 A 的数目为 TP(true positive).同时,测试数据集中还有 m 个非 A 数据,其中被误识别为 A 的数目为 FP.定义正识别率为 $TPR=TP/n$,反映了识别特征对于协议 A 自身的准确性.另外定义负识别率为 $FPR=FP/m$,反映了识别特征对于其他协议的区分度.

对于每一个协议,我们从其数据集中随机选择 N 个观测序列来训练 HSMM 模型并构建识别特征,然后用其数据集中剩余的观测序列来评估 TPR.另外,我们用其他协议的所有观测序列来测试识别特征的 FPR.如图 5 所示,当样本数目较小时,正识别率会偏低.这主要是因为在小样本的情况下,训练集中会话的类型偏少,某些在协议规范中属于可选的关键词在所有的关键词序列中都有出现,被标识为特征词.此时,生成的识别特征只能代表部分类型的会话,从而造成 TPR 偏低.随着 N 的增大,训练集中会话类型的增多,可选的关键词被判定为特征词的概率变小,这样,TPR 也随之增加.当 N>100 时,基本可以保证 100%的 TPR.

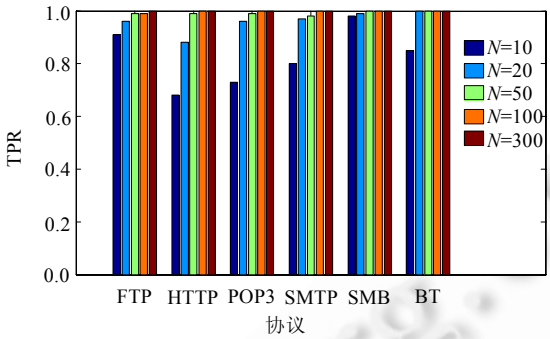

Fig.5 TPRs under different sample size N

图 5 不同训练集大小的 TPR

另一方面,无论样本集的大小,FPR 都为 0.这是因为识别特征是由多个特征词组成.虽然有可能去掉某些特征词后,识别特征的有效性依然不变,即识别特征包含了冗余信息,并不一定是最优和最高效的.但是特征词数目越多,其他协议的应用层载荷包含所有特征词的概率就越小,从而保证最低的 FPR.

3.4 模型状态数的影响

模型的状态数决定了模型的结构.对于给定的训练集,不同的状态数会使得提取出来的关键词序列集合有所不同.我们用不同的关键词序列集合之间的相似性来度量状态数对提取结果的影响.给定样本数为 N 的训练集合,设训练集中第 n 个观测序列 $O^{(n)}$在状态数为 i 时提取的关键词序列为 $K_i^{(n)}$,定义 $K_i^{(n)}$ 和 $K_j^{(n)}$ 的相似度为

$$Sim(K_i^{(n)},K_j^{(n)})=\frac{2\times|lcs(K_i^{(n)},K_j^{(n)})|}{|K_i^{(n)}|+|K_j^{(n)}|} \tag{14}$$

其中,$lcs(K_i^{(n)},K_j^{(n)})$ 是两个序列的最长公共子串,代表两个序列相同的部分,$|K|$是序列 K 的长度.设训练集在状态数为 i 下得到的关键词序列集合为 $K\_set(i)$,$K_set(i)$和 $K_set(j)$的相似度定义为

$$Sim(i,j)=\frac{1}{N}\sum_{n=1}^{N}Sim(K_i^{(n)},K_j^{(n)}) \tag{15}$$

对于每种协议,我们取 N=100,状态数分别设为 3,5,10 和 15.表 5 总结了各种协议在不同状态数下提取出来的关键词序列集合之间的相似性.即使状态数目分别取 3 和 15,这两者所提取出来的关键词序列集合有超过 95%是相同的.可以认为,状态数目对提取结果的影响不大,为了节省运算量,可以选择一个较小的状态数.

Table 5 Similarities for different pair of state number

表 5 不同状态数下的关键词集合的相似度

协议	*Sim*(3,5)	*Sim*(3,10)	*Sim*(3,15)	*Sim*(5,10)	*Sim*(5,15)	*Sim*(10,15)
FTP	0.971	0.961	0.961	0.99	0.988	0.998
HTTP	1	1	1	1	1	1
SMTP	0.963	0.956	0.953	0.977	0.989	0.981
POP3	0.998	1	0.998	0.997	0.996	1
SMB	0.956	0.952	0.951	0.978	0.983	0.994
BT	1	1	1	1	1	1

4 结论与展望

本文提出一种基于网络流量的未知应用层协议报文格式的最佳分段方法.该方法把包含关键词的分段看作是对应用层载荷进行解析的语法单位,并通过多个会话序列训练出代表协议解析规则的 HSMM 模型.与此同时,利用 Viterbi 算法可以得到各个观测序列的最大似然概率的段划分,以及提取出各分段所包含的关键词.另外,基于观测序列平均对数似然概率的分布,可以区分出混杂在协议数据中的噪声.实验结果表明,该方法不依

赖任何的协议先验知识,可以处理文本和二进制协议,并能有效地区分出噪声.虽然训练集样本中所包含的非协议关键词的内容会对最终结果有所影响,但是提取出来的关键词大部分都是协议规范所定义的,而且依据关键词构建的识别特征有很高的准确率.下一步的工作在于如何利用本文的结果提取更为精确的报文格式,以及推导协议的状态机,从而实现完整的协议逆向工程.

References:

[1] Wireshark. Network protocol analyzer. http://www.wireshark.org

[2] How samba was written. http://samba.org/ftp/tridge/misc/french_cafe.txt

[3] Moore AW, Zuev D, Crogan M. Discriminators for use in flow-based classification. Technical Report, RR-05-13, London: Queen Mary University of London, 2005.

[4] Nguyen TTT, Armitage G. A survey of techniques for Internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials, 2008,10(4):56−76. [doi: 10.1109/SURV.2008.080406]

[5] Haffner P, Sen S, Spatscheck O, Wang D. ACAS: Automated construction of application signatures. In: Sen S, ed. Proc. of the 2005 ACM SIGCOMM Workshop on Mining Network Data. New York: ACM Press, 2005. 197−202. [doi: 10.1145/1080173.1080183]

[6] Ma J, Levchenko K, Kreibich C, Savage S, Voelker GM. Unexpected means of protocol inference. In: Almeida J, ed. Proc. of the 6th ACM SIGCOMM on Internet Measurement. New York: ACM Press, 2006. 313−326. [doi: 10.1145/1177080.1177123]

[7] Zhao Y, Yao QL, Zhang ZB, Guo L, Fang BX. TPCAD: A text-oriented multi-protocol inference approach. Journal of Communications, 2009,30(S1):28−35 (in Chinese with English abstract).

[8] Liu XB, Yang JH, Xie GG, Hu Y. Automated mining of packet signatures for traffic identification at application layer with Apriori algorithm. Journal of Communications, 2008,29(12):51−59 (in Chinese with English abstract).

[9] Borisov N, Brumley DJ, Wang HJ, Guo C. A generic application-level protocol analyzer and its language. In: Proc. of the 14th Annual Network & Distributed System Security Symp. Internet Society, 2007.

[10] Caballero J, Yin H, Liang Z, Song D. Polyglot: Automatic extraction of protocol message format using dynamic binary analysis. In: Ning P, ed. Proc. of the 14th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2007. 317−329. [doi: 10.1145/1315245.1315286]

[11] Lin Z, Jiang X, Xu D, Zhang X. Automatic protocol format reverse engineering through context-aware monitored execution. In: Proc. of the 15th Symp. on Network and Distributed System Security. Internet Society, 2008.

[12] Wang Z, Jiang X, Cui W, Wang X, Grace M. ReFormat: Automatic reverse engineering of encrypted messages. In: Backers M, ed. Proc. of the 14th European Symp. on Research in Computer Security. Berlin, Heidelberg: Springer-Verlag, 2009. 200−215. [doi: 10.1007/978-3-642-04444-1_13]

[13] Comparetti PM, Wondracek G, Kruegel C, Kirda E. Prospex: Protocol specification extraction. In: Werner B, ed. Proc. of the 30th IEEE Symp. on Security and Privacy. IEEE, 2009. 110−125. [doi: 10.1109/SP.2009.14]

[14] The protocol informatics project. http://www.baselineresearch.net/PI/

[15] Li WM, Zhang AF, Liu JC, Li ZT. An automatic network protocol fuzz testing and vulnerability discovering method. Chinese Journal of Computers, 2011,34(2):242−255 (in Chinese with English abstract).

[16] Cui WD, Kannan J, Wang HJ. Discoverer: Automatic protocol reverse engineering from network traces. In: Provos N, ed. Proc. of the 16th Usenix Security Symp. USENIX Association, 2007. 199−212.

[17] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE, 1989,77(2):257−286. [doi: 10.1109/5.18626]

[18] Yu SZ. Hidden semi-Markov models. Artificial Intelligence, 2010,174:215−243. [doi: 10.1016/j.artint.2009.11.011]

[19] RFC 2821—Simple mail transfer protocol. http://www.ietf.org/rfc/rfc2821.txt

[20] RFC 959—File transfer protocol. http://www.ietf.org/rfc/rfc959.txt

[21] Yagi S, Waizumi Y, Tsunoda H, Nemoto Y. A reliable network identification method based on transition pattern of payload length. In: Miller RW, ed. Proc. of the IEEE Global Telecommunications Conf. IEEE, 2008. 1915−1919. [doi: 10.1109/GLOCOM.2008.ECP.370]

[22] Mahoney MV, Chan PK. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In: Vinga G, ed. Proc. of the 6th Symp. on Recent Advances in Intrusion detection. Berlin, Heidelberg: Springer-Verlag, 2003. 220–237. [doi: 10.1007/978-3-540-45248-5_13]

[23] L7-filter: Application layer packet classifier for linux. http://l7-filter.sourceforge.net/

附中文参考文献:

[7] 赵咏,姚秋林,张志斌,郭莉,方滨兴.TPCAD:一种文本类多协议特征自动发现方法.通信学报,2009,30(S1):28−35.

[8] 刘兴彬,杨建华,谢高岗,胡玥.基于 Apriori 算法的流量识别特征自动提取方法.通信学报,2008,29(12):51−59.

[15] 李伟明,张爱芳,刘建财,李之棠.网络协议的自动化模糊测试漏洞挖掘方法.计算机学报,2011,34(2):242−255.

黎敏(1982－),男,广东广州人,博士生,主要研究领域为网络协议分析,网络安全.
E-mail: sysu_lm@126.com

余顺争(1958－),男,博士,教授,博士生导师,主要研究领域为网络安全,下一代互联网.
E-mail: syu@mail.sysu.edu.cn