·移动互联与通信技术· 文章编号：1000-3428(2019)07-0126-08 文献标志码：A 中图分类号：TP393

基于状态相关字段的二进制协议状态机推断

闫小勇，李 青，莫有权

（信息工程大学 信息系统工程学院，郑州 450000）

摘 要： 在通信协议规范中，报文的格式类型与状态类型不存在一一映射关系，通过聚类较难将格式类型相同、状态类型不同的报文分离。为此，提出一种基于状态相关字段的二进制私有协议状态机推断方法。根据最长公共子序列距离进行状态相关字段识别，以获取协议会话的行为逻辑相似性。构建基于邻接表的初始状态机，对其进行异常会话去除与相似状态合并，从而降低协议状态机的规模。在 TCP 协议和 SMB 协议数据集上的测试结果表明，该方法能够有效推断二进制私有协议状态机，其准确率与召回率均较高。

关键词： 协议状态机；二进制协议；最长公共子序列距离；邻接表；异常会话去除；相似状态合并

中文引用格式： 闫小勇，李青，莫有权. 基于状态相关字段的二进制协议状态机推断[J]. 计算机工程，2019，45(7)：126-133.

英文引用格式： YAN Xiaoyong，LI Qing，MO Youquan. State machine inference for binary protocol based on state-related field[J]. Computer Engineering，2019，45(7)：126-133.

State Machine Inference for Binary Protocol Based on State-related Field

YAN Xiaoyong，LI Qing，MO Youquan

（School of Information System and Engineering，Information Engineering University，Zhengzhou 450000，China）

【Abstract】 As the one-to-one mapping relationship does not exist between the message format type and the message status type in the communication protocol specification，it is difficult to separate messages with the same format type and different status type by clustering. Therefore，a state machine inference method for binary private protocol based on state-related field is proposed. State-related field are identified according to the Longest Common Subsequence Distance (LCSD) to obtain the logical similarity of protocol sessions. An initial state machine based on adjacency table is constructed，and its abnormal session is removed and similar state is merged to reduce the size of protocol state machine. Test results on TCP and SMB protocol datasets show that the proposed method can effectively infer the state machine of binary private protocol，and both its accuracy and recall rate are high.

【Key words】 protocol state machine；binary protocol；the Longest Common Subsequence Distance(LCSD)；adjacency table；abnormal session removal；similar state merging

DOI：10.19678/j.issn.1000-3428.0050589

0 概述

协议状态机推断需要获取协议会话中的报文状态类型，该状态类型标识了报文在会话中的逻辑顺序与功能[1-3]。例如，TCP 协议报文中的 6 位控制代码：URG，ACK，PSH，RST，SYN，FIN，其不同的取值组合代表了不同的状态类型。能够表征协议状态类型的字段称为状态相关字段[4]。

根据获取报文状态类型方法的不同，现有关于协议状态机推断的相关研究可以分为基于聚类和基于状态相关字段 2 种类型。文献[5]通过层次聚类为每条报文分配类别标签，将协议会话报文序列转化为类别标签序列，构建初始状态机并对其进行化简，得到最小确定状态机。但是，其将报文之间的格式相似性作为相似性度量标准，对于格式相似但状态类型不同的报文，该方法将无法进行区分。文献[6]采用 Dunn 指数[7]与 Jaccard 指数[8]分别确定聚类类别数与报文相似性，最终利用 PAM 算法[9]实现报文聚类。但是，其同样以聚类后的类别标签代替协议报文，与文献[5]方法面临相同的问题。上述方法虽然易于实现，但报文的格式类型与状态类型间不存在一一映射关系，一些格式相似但状态类型不同的报文可能被聚为一类，最终导致状态机推断错误。

文献[4]认为同种协议的会话具有相似的逻辑，协议的状态信息由状态相关字段唯一确定。该文将二进制协议会话报文序列转化为状态相关字段序

基金项目： 国家自然科学基金“多天线无线携能通信系统中的物理层安全传输技术研究”(61601516)。

作者简介： 闫小勇(1993—)，男，硕士研究生，主研方向为协议逆向技术、数据挖掘；李 青、莫有权，副教授。

收稿日期： 2018-03-05 **修回日期：** 2018-05-21 **E-mail：** yanxiaoyong2016@163.com

列,并构建初始状态机。文献[10]利用统计方法提取状态相关字段,通过邻接矩阵表示协议报文间的时序关系。文献[11-12]基于指令序列划分协议字段,从中选取状态相关字段用于协议状态机推断,但文中并未详细说明状态相关字段的具体选取方法。上述研究利用状态相关字段的不同取值来表示协议报文的状态类型,较符合实际情况,得到的状态机准确性较高。但面向私有协议时,因为缺少先验信息,导致难以准确获取状态相关字段。

在协议字段划分的基础上[13],筛选状态相关字段是推断二进制协议状态机的关键步骤。本文提出一种基于状态相关字段的二进制私有协议状态机推断方法 BSMISRF。根据最长公共子序列距离(the Longest Common Subsequence Distance,LCSD)进行协议状态相关字段识别,构建初始状态机并去除异常会话,然后基于出度、入度进行相似状态合并,在此基础上,化简初始状态机以构建概率协议状态机。

1　问题描述

不同于文本类协议,二进制协议的字段不受字节长度的限制,字段之间不存在明显的定界符,导致其字段划分比文本类协议困难[13]。协议状态机只与协议的状态相关字段有关。状态相关字段的取值可变,同种类型的协议字段中往往存在多个可变字段。在无先验知识的条件下,状态相关字段极易与其他可变字段混淆,从而给协议状态相关字段识别造成困难。

文献[14-16]根据所得的状态相关字段,构建状态相关字段序列,由该序列生成 APTA 树,然后化简状态机。在会话数量较多时,构造的状态前缀树往往过于庞大,状态机化简需要大量的比较操作。

由训练集协议会话构建的状态机是一个特化的状态机,只能接受训练集中的协议会话。异常报文序列的存在导致协议状态机中可能出现错误的状态转移。此外,当协议状态类型较多时,所构建的状态机过于复杂,不易于理解。为构造正确、具有较小规模的协议状态机,需要对初始状态机进行化简。因此,进行二进制私有协议状态机推断需要解决协议字段划分、状态相关字段选取、初始状态机构建以及状态机化简等关键问题,其流程如图 1 所示。

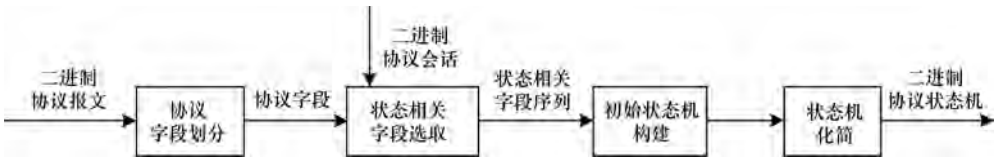

图 1　二进制私有协议状态机推断流程

假设输入的同种类型的协议报文集合为 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\}$,$x_i=b_{i1}b_{i2}\cdots b_{ij}\cdots b_{im}$表示第 i 条二进制协议报文,$b_{ij}\in\{\backslash 0b0,\backslash 0b1\}$。经过协议字段划分后,第 i 条二进制协议报文表示为 $x_i'=f_{i1}\parallel f_{i2}\parallel\cdots\parallel f_{ik}\parallel\cdots\parallel f_{ip}$,$f_{ik}=b_{ij_1}b_{i(j_1+1)}\cdots b_{ij_2}$表示第 i 条协议报文的第 k 个字段,"$\parallel$"表示链接。由协议报文集合 X 重构的协议会话集为 $S=(s_1,s_2,\cdots,s_i,\cdots,s_N)$,$s_i=(x_{i1},x_{i2},\cdots,x_{ij},\cdots,x_{iM})$表示第 i 个协议会话,x_{ij}表示第 i 个协议会话的第 j 条协议报文。结合协议字段划分结果,第 i 个协议会话为 $s_i'=(x_{i1}',x_{i2}',\cdots,x_{ij}',\cdots,x_{iM}')$,第 i 个协议会话的第 j 个字段序列为 $s_{ij}'=(f_{i1j},f_{i2j},\cdots,f_{ijj},\cdots,f_{iMj})$,$f_{ijj}$表示第 i 个协议会话第 j 条协议报文的第 j 个字段。假定选取的状态相关字段为协议报文的第 k 个字段,则与协议会话 s_i 相对应的状态相关字段序列为 $s_{if}=(f_{i1k},f_{i2k},\cdots,f_{ijk},\cdots,f_{iMk})$。以 s_{if}作为输入,构建初始状态机,通过协议状态的删除、合并操作实现状态机化简。

2　BSMISRF 算法

2.1　基于 LCSD 的状态相关字段识别

在构建初始状态机时无法直接使用协议会话 $s_i=(x_{i1},x_{i2},\cdots,x_{ij},\cdots,x_{iM})$,需要将协议会话用状态相关字段进行表示,构建与协议会话对应的状态相关字段序列 $s_{if}=(f_{i1k},f_{i2k},\cdots,f_{ijk},\cdots,f_{iMk})$,并作为初始状态机的输入,如图 2 所示。在面向私有协议时,先验信息未知,无法直接获取协议的状态相关字段,需要先进行字段划分,将 s_i 转换为 $s_i'=(x_{i1}',x_{i2}',\cdots,x_{ij}',\cdots,x_{iM}')$,再根据划分结果识别状态相关字段,最终构建状态相关字段序列 s_{if}。

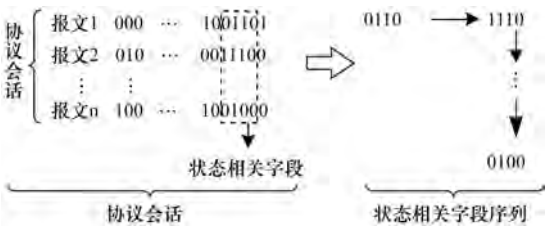

图 2　状态相关字段序列示意图

文献[4]总结出协议会话的特性:

1)同种协议的会话具有相似的行为逻辑。

2)协议行为逻辑由状态相关字段表示。

因此,同种协议会话的状态相关字段序列具有相似的行为逻辑。文献[4]首先计算同种协议中不同会话的各字节方差,如式(1)所示。然后计算不同会话的同一字节方差分布的方差(Variance of the Distribution of Variances,VDV),如式(2)所示。最终,将 VDV 最小的字节作为状态相关字节。

$$\sigma_{ik}^2=\frac{(f_{i1k}-\bar{f}_{ik})^2+(f_{i2k}-\bar{f}_{ik})^2+\cdots+(f_{iMk}-\bar{f}_{ik})^2}{M} \quad (1)$$

其中,σ_{ik}^2表示第 i 个会话第 k 个字节的方差,f_{i1k}表示第 i 个会话第 1 个报文的第 k 个字节,$f_{i1k}=b_{i1j_1}b_{i1(j_1+1)}\cdots b_{i1j_2}$,$j_1=8\times k-7$,$j_2=8\times k$,$M$ 表示会话中的报文数。

$$\sigma_k^2=\frac{(\sigma_{1k}^2-\bar{\sigma}_k^2)^2+(\sigma_{2k}^2-\bar{\sigma}_k^2)^2+\cdots+(\sigma_{Nk}^2-\bar{\sigma}_k^2)^2}{N} \quad (2)$$

其中,σ_k^2 表示第 k 个字节的 VDV,N 表示会话数。

文献[4]算法(简称为 VDV 算法)对于协议状态机推断有一定效果,但方差仅能反映样本的离散程度,很难体现样本间的顺序约束关系与行为逻辑。例如,在某会话集中,只有同种类型协议的 2 个会话:会话 1 与会话 2,会话的第 2 个字段序列如图 3 所示。采用 VDV 算法,会话 1 和会话 2 的第 2 个字段序列相似,因为会话 1 第 2 个字段序列的元素全部包含在会话 2 第 2 个字段序列中。根据式(1)、式(2)可知,σ_2^2 取值较小时,第 2 个字段可能被识别为状态相关字段。但事实上,会话 1 和会话 2 的第 2 个字段序列所包含的元素虽然相似,但不同元素间的顺序差别较大,即会话 1 和会话 2 的行为逻辑存在较大差异。由此可见,VDV 算法仅能刻画序列所包含元素的相似性,而无法体现序列元素的顺序相似性。

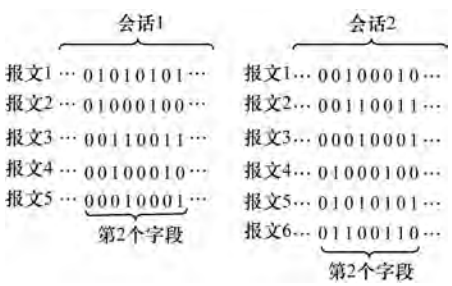

图 3 同种类型协议会话集示意图

为解决上述问题,本文提出一种基于 LCSD 的协议状态相关字段识别算法。LCSD 是基于最长公共子序列的一种距离,对于一个序列 s,如果其分别是 2 个或多个已知序列的子序列,且在所有符合此条件的序列中长度最长,则称 s 为已知序列的最长公共子序列。最长公共子序列可以通过动态规划进行求解,式(3)是 LCSD 的计算公式。

$$d(s'_{i_1j},s'_{i_2j})=1-\frac{|LCS(s'_{i_1j},s'_{i_2j})|}{\min(|s'_{i_1j}|,|s'_{i_2j}|)} \quad (3)$$

其中,$d(s'_{i_1j},s'_{i_2j})$表示字段序列 s'_{i_1j}和 s'_{i_2j}之间的距离,$LCS(s'_{i_1j},s'_{i_2j})$表示 s'_{i_1j}和 s'_{i_2j}的最长公共子序列,$|LCS(s'_{i_1j},s'_{i_2j})|$表示最长公共子序列的长度,$|s'_{i_1j}|$和$|s'_{i_2j}|$分别表示 s'_{i_1j}和 s'_{i_2j}的长度。

例如,有状态相关字段序列 $s'_{12}=(1,2,3,4,5,6)$和 $s'_{22}=(1,2,3,4,6,5)$,$LCS(s'_{12},s'_{22})=(1,2,3,4,5)$,$d(s'_{12},s'_{22})=1-5/6=0.17$,因此,序列 s'_{12}和 s'_{22}最长公共子序列距离为 0.17。有序列 $s'_{12}=(1,2,3,4,5,6)$和序列 $s'_{32}=(6,5,4,3,2,1)$,$LCS(s'_{12},s'_{32})=\varnothing$,$d(s'_{12},s'_{32})=1-0/6=1$,因此,序列$s'_{12}$和 s'_{32}最长公共子序列距离为 1。由此可见,LCSD 不仅能够刻画序列所包含元素的相似性,同时能反映序列元素的顺序相似性,即行为逻辑相似性。序列所包含的元素相似,元素逻辑顺序也相似,则 LCSD 较小。在会话集中,LCSD 之和最小的字段序列为状态相关字段序列,LCSD 之和的计算如式(4)所示。

$$D_j=\sum_{i_1,i_2\in(1,\cdots,|S|),i_2>i_1}d(s'_{i_1j},s'_{i_2j}) \quad (4)$$

其中,$|S|$表示会话集中的会话数量,D_j 表示会话集中所有会话的第 j 个字段序列的 LCSD 之和。最小 D_j 对应的字段为状态相关字段,用 f_s 来表示。

基于 LCSD 进行状态相关字段识别的具体步骤如下:

步骤 1 获取同种协议的会话,构造协议会话集 $S=\{s_1,s_2,\cdots,s_N\}$。

步骤 2 设置指针指向协议报文的第 0 个字段。

步骤 3 判断指针是否指向最后一个字段,若是,转到步骤 5;否则,指针指向下一个字段,转到步骤 4。

步骤 4 提取会话集中不同会话的第 j 个字段序列$\{s_{1j},s_{2j},\cdots,s_{|S|j}\}$,计算 D_j,转到步骤 3。

步骤 5 根据规则选取最小 D_j 对应的字段,将其作为状态相关字段。

识别出状态相关字段后,将会话集中的所有会话用状态相关字段序列进行表示,并作为初始状态机的输入,如图 2 所示。

2.2 基于邻接表的初始状态机构建

构建初始状态机的主流方法是构建 APTA 树,然后接受所有状态相关字段序列并进行合并化简。该过程会导致初始构建的 APTA 树过于庞大,需要大量的比较操作[10]。本文利用邻接表构建初始状态机,省去 APTA 树的相关过程。假设协议会话用状态相关字段序列表示为:

0001→0010→0011→0100→0101→0010

初始协议状态邻接表如表 1 所示。状态 0 为起始状态,没有对应的状态相关字段,表示不存在一个状态在接受一个报文后能够转移到初始状态。输入状态相关字段序列的第 1 个值“0001”,状态相关字段所在的列不存在“0001”,因此,添加“0001”到列

表中,同时添加状态 1,表示在接受到状态相关字段“0001”代表的报文后,协议由之前的状态转移到状态 1。在状态 0 对应的出度列中添加状态 1,表示状态 0 可以转移到状态 1,更新后的邻接表如表 2 所示。将整个状态相关字段序列输入到状态邻接表中,结果如表 3 所示。最后一个值为“0010”,已经在状态相关字段列中存在,因此,只需更新状态 5 的出度,将状态 2 添加到状态 5 的出度列表中即可。

表 1　初始协议状态邻接表

状态相关字段	状态	出度
—	0	—

表 2　协议状态邻接表 1

状态相关字段	状态	出度
—	0	1
0001	1	—

表 3　协议状态邻接表 2

状态相关字段	状态	出度
—	0	1
0001	1	2
0010	2	3
0011	3	4
0100	4	5
0101	5	2

协议状态机的终止状态往往不止一个,为构建便于理解的协议状态机,本文标识终止状态,同时记录每个转移出现的次数,最终构造概率协议状态机,如表 4 所示。其中,2(T)表示状态 2 可以作为终止状态,1:1 表示由状态 0 到状态 1 的转移出现了 1 次,其余标识同上。将会话集中所有会话对应的状态相关字段序列添加到协议状态邻接表中,当输入的状态相关字段已存在于状态邻接表中时,只需更新出度列表;否则,先添加新的状态再更新出度列表。

表 4　协议状态邻接表 3

状态相关字段	状态	出度
—	0	1:1
0001	1	2:1
0010	2(T)	3:1
0011	3	4:1
0100	4	5:1
0101	5	2:1

由表 4 状态邻接表还原出的概率协议状态机如图 4 所示,其中,状态 2 可以作为终止状态,用 2 个同心圆表示,“50% ”表示状态 2 有 50% 的概率成为终止状态,转移弧线下方的百分比表示各出度的比例。例如,状态 1 只有一个出度状态 2,该出度的概率如式(5)所示。

$$p_1(1,2)=\frac{n(1,2)}{n(1,:)}=\frac{n(1,2)}{n(1,2)}=100\% \tag{5}$$

其中,$p_1(1,2)$表示状态 1 的出度状态 2 出现的概率,$n(1,2)$表示状态 1 到状态 2 的转移出现的次数,$n(1,:)$表示状态 1 的所有出度的总次数。

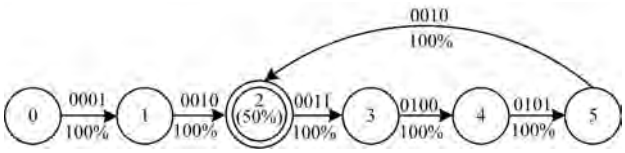

图 4　概率协议状态机示意图

2.3　协议状态机化简

2.3.1　基于概率统计的异常会话去除

在真实环境中,用户操作失误或者传输环境变化通常会导致不完整或错误会话的出现,本文将其统称为异常会话。包含小概率转移的会话为异常会话,异常会话在整个数据集中比例较小。利用含有异常会话的数据集推断的协议状态机会存在错误。本文统计各个转移出现的概率,发现概率小的转移后定位转移所在的会话,将该会话剔除,然后在不包含异常会话的协议数据集上重新构建状态邻接表。

表 5 所示为当前构建的状态邻接表,输入状态相关字段序列 0001→0011→0100→0101 后,状态邻接表如表 6 所示。其中,状态 1 有 2 个出度,分别为状态 2 和状态 3,状态 1 到状态 3 的概率如式(6)所示,状态 1 到状态 2 的概率如式(7)所示。状态 1 到状态 3 的概率仅为 1% ,为小概率事件,设置阈值 θ_{th},当转移概率小于 θ_{th}时,对应转移为异常转移,对应会话为异常会话。如果直接删除异常转移,不做其他处理,则异常会话的剩余部分仍然存在于状态邻接表中。为彻底消除异常会话的影响,本文在确定异常转移与异常会话后,将所有的异常会话从原始数据集中进行消除,再对剩余的会话重新构建新的状态邻接表。

表 5　协议状态邻接表 4

状态相关字段	状态	出度
—	0	1:100
0001	1	2:100
0010	2	3:100
0011	3	4:100
0100	4	5:100
0101	5(T)	—

表 6　协议状态邻接表 5

状态相关字段	状态	出度
—	0	1:101
0001	1	2:100,3:1
0010	2	3:100
0011	3	4:101
0100	4	5:101
0101	5(T)	—

$$p_1(1,3)=\frac{n(1,3)}{n(1,:)}=\frac{n(1,3)}{n(1,3)+n(1,2)}=1\% \tag{6}$$

$$p_2(1,2)=\frac{n(1,2)}{n(1,:)}=\frac{n(1,2)}{n(1,3)+n(1,2)}=99\% \tag{7}$$

2.3.2 基于出度、入度的相似状态合并

经过前文的处理,协议状态邻接表中不存在相同的协议状态相关字段。虽然没有相同的状态,但可能存在相似的状态。为最大限度地化简状态机,需要将相似状态进行合并。协议状态之间存在 2 种关系:顺序和并列,如图 5 所示。顺序关系指状态之间存在先后顺序,例如状态 1 在状态 0 后,两者是顺序关系。并列关系指状态之间没有先后顺序,例如在状态 2 之后状态 3 ~ 状态 5 都有可能出现,状态 3 ~ 状态 5 之间没有先后顺序,属于并列关系。具有并列关系的状态属于相似状态,可以合并,图 6 所示为合并图 5 中相似状态后的状态机。

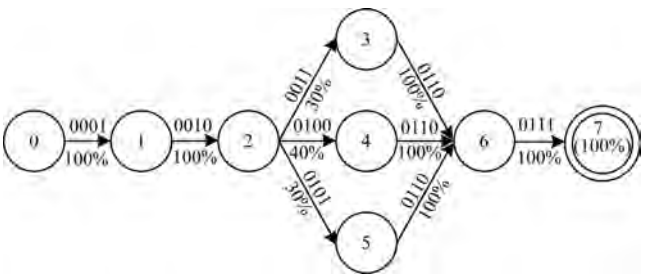

图 5 存在相似状态的协议状态机示意图

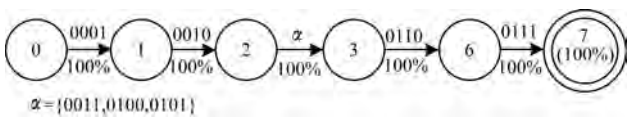

图 6 合并相似状态后的协议状态机示意图

相似状态在状态邻接表中表现为入度和出度均相同。表 7 所示为一个协议状态邻接表,各状态相应的出度、入度如表 8 所示。状态 3 ~ 状态 5 的入度均为状态 2,出度均为状态 6,因此,状态 3 ~ 状态 5 为相似状态,可以合并。合并后的状态邻接表如表 9 所示,状态 3 ~ 状态 5 合并为状态 3,状态相关字段合并到状态 3 的相关字段,包含状态 4 和状态 5 的出度列表统一更新为状态 3,并对相同出度对应的次数进行合并。

表 7 协议状态邻接表 6

状态相关字段	状态	出度
—	0	1:100
0001	1	2:100
0010	2	3:100,4:100,5:100
0011	3	6:100
0100	4	6:100
0101	5	6:100
0110	6	7:100
0111	7(T)	—

表 8 协议状态的入度、出度信息

状态	入度	出度
0	—	1
1	0	2
2	1	3,4,5
3	2	6
4	2	6
5	2	6
6	3,4,5	7
7(T)	6	—

表 9 协议状态邻接表 7

状态相关字段	状态	出度
—	0	1:100
0001	1	2:100
0010	2	3:300
0011/0100/0101	3	6:300
0110	6	7:100
0111	7(T)	—

基于概率统计的异常会话去除和基于出度、入度的相似状态合并,可以分别看作从原数据集和初始状态机 2 个方面对协议状态机进行化简。

3 实验结果与分析

为验证 BSMISRF 算法对二进制协议状态机推断的有效性,选取 TCP 协议和 SMB 协议进行协议状态机推断。实验环境为一台 PC(CPU i7-4720HQ,内存 8 GB),操作系统为 Windows 7,利用 Python 编程语言进行程序编写测试。通过 Wireshark 软件在校园骨干网上采集数据包,再通过五元组实现 TCP 协议会话分割,获取 TCP 协议会话集。在 PC 上设置共享文件夹,在手持客户端上安装文件共享软件,通过文件共享软件对共享文件夹进行操作,如文件创建、删除等,利用 Wireshark 软件采集操作过程中产生的 SMB 协议数据包,再通过五元组实现 SMB 协议会话分割,获取 SMB 协议会话集。实验中使用的训练数据集和测试数据集信息如表 10、表 11 所示。

表 10 训练数据集信息

协议	报文数量	会话数量	大小/MB
TCP	1 103	100	0.228
SMB	7 012	109	1.710

表 11 测试数据集信息

协议	报文数量	会话数量	大小/MB
TCP	1 971	200	0.438
SMB	6 066	100	1.470

在实验过程中,仅涉及一个参数 θ_{th},本文设定

θ_{th}为 0.01。评价指标采用协议逆向工程中常用的准确率和召回率,如式(8)、式(9)所示。

$$precision_p = \frac{n_p^t}{n_p} \tag{8}$$

$$recall_p = \frac{n_p^t}{n^p} \tag{9}$$

其中,$precision_p$ 和 $recall_p$ 分别表示协议 p 的准确率和召回率,n_p 表示测试集数据被识别为协议 p 的会话数,n^p 表示测试集数据中真正属于协议 p 的会话数,n_p^t 表示被识别为协议 p 且真正属于协议 p 的会话数。

3.1　协议状态机推断结果

由 BSMISRF 算法推断的 TCP 和 SMB 协议状态机分别如图 7、图 8 所示。在 TCP 协议状态机中,路径 0→1→2→3 表示协议中 3 次握手建立连接的过程,4→5→3→5→3 和 3→5→3→5→3 均为 TCP 协议 4 次挥手关闭连接的过程。SMB 协议状态相关字段对应的命令码如表 12 所示,路径 0→1→2→3 表示 SMB 协议的会话建立过程。

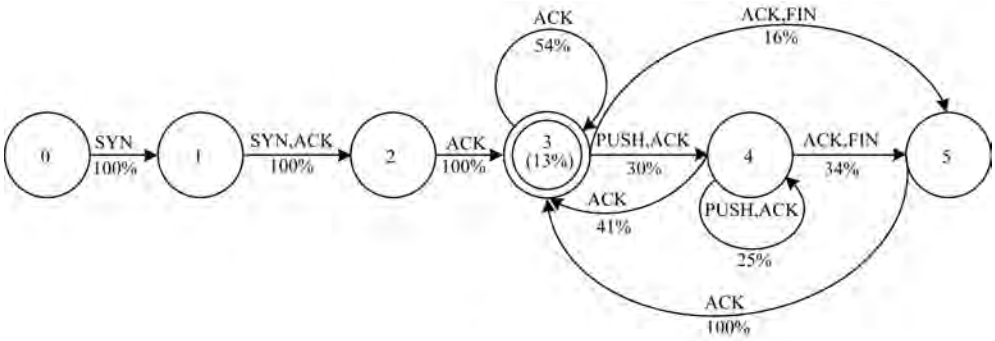

图 7　TCP 协议状态机

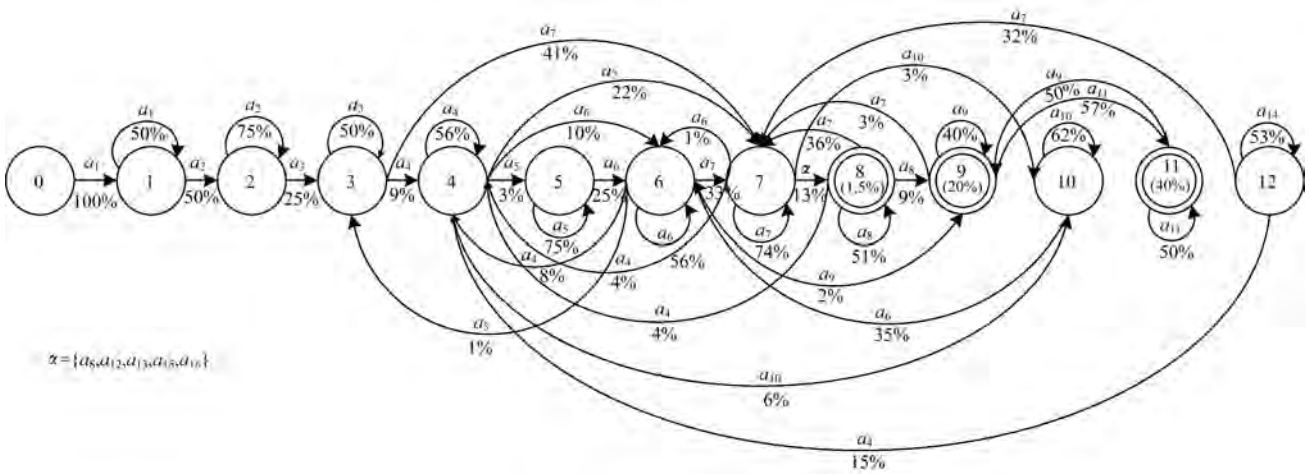

图 8　SMB 协议状态机

表 12　SMB 协议命令码

符号	状态相关字段	Command 字段
a_1	72	NEGOTIATE
a_2	73	SESSION SETUP ANDX
a_3	75	TREE CONNECT ANDX
a_4	*a*2	NT CREATE ANDX
a_5	25	TRANSACTION
a_6	04	CLOSE
a_7	32	TRANSACTION2
a_8	34	FIND CLOSE2
a_9	71	TREE DISCONNECT
a_{10}	2*e*	READ ANDX
a_{11}	74	LOGOFF ANDX
a_{12}	06	DELETE
a_{13}	07	RENAME
a_{14}	2*f*	WRITE ANDX
a_{15}	00	CREATE DIRECTORY
a_{16}	01	DELETE DIRECTORY

3.2　VDV 算法与 LCSD 算法性能比较

VDV 算法和 LCSD 算法均是状态相关字段选取算法,前者只能刻画序列元素的相似性,后者既能体现序列元素的相似性,又能反映序列元素行为逻辑的相似性。图 9、图 10 所示分别为 TCP 协议和 SMB 协议的状态相关字节/段选取结果。其中,纵坐标中的"log"表示对纵坐标值取以 e 为底的对数。VDV 算法以 *VDV* 最小取值对应字节为协议状态相关字节(*VDV* 取值为 0 的是不可变字节)。LCSD 算法以 *D* 最小取值对应字段为协议状态相关字段(*D* 取值为 0 的是不可变字段)。对于 TCP 协议,能够表示协议状态的是 6 位控制代码:URG,ACK,PSH,RST,SYN,FIN,在 TCP 协议报文的第 14 个字节第 7 个字段。VDV 和 LCSD 均能找

到 TCP 协议的状态相关字节/段，如图 9(a)和 9(b)。对于 SMB 协议，能够表示协议状态的是 Command 字段，在 SMB 协议报文的第 5 个字节第 2个字段。VDV 算法无法找到 SMB 协议的状态字节，如图 10(a)，LCSD 算法能够找到 SMB 协议的状态相关字段，如图 10(b)。

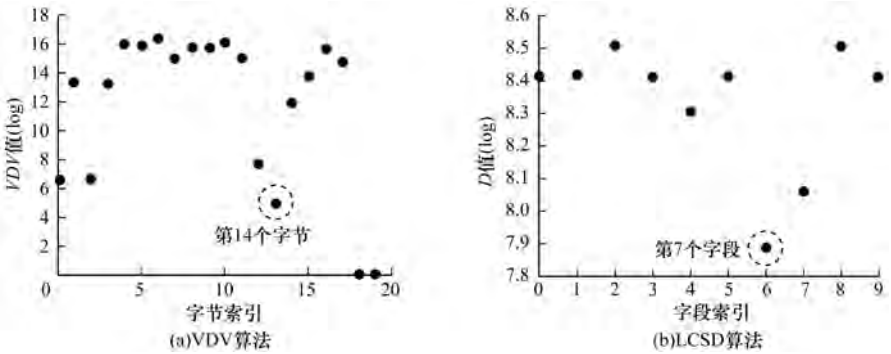

图 9 TCP 协议状态相关字段选取结果

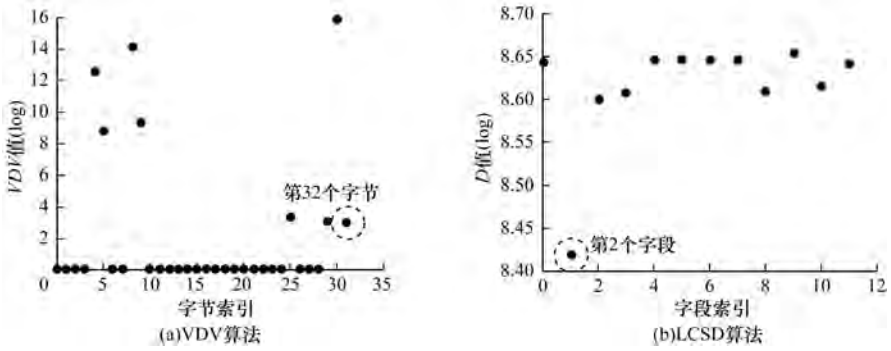

图 10 SMB 协议状态相关字段选取结果

3.3 化简前后的协议状态机规模比较

本文基于概率统计进行异常会话去除，基于出度、入度实现相似状态合并，以对初始状态机进行化简。化简前后的状态机规模如表 13、表 14 所示。其中，NS_{TCP}和 NS_{SMB}分别表示 TCP 和 SMB 协议的初始状态机，S_{TCP}和 S_{SMB}分别表示 TCP 和 SMB 协议的最终状态机。TCP 协议数据集中不包含相似状态，但存在异常会话，因此，状态数和终止状态数没有变化，只有状态转移数得到了约简。SMB 协议数据集中存在相似状态，也存在异常会话，因此，状态数、状态转移数以及终止状态数均得到了约简。2 种协议的约简比例如图 11 所示。

表 13 TCP 协议状态机规模统计

协议状态机	状态数	状态转移数	终止状态数
NS_{TCP}	6	12	1
S_{TCP}	6	10	1

表 14 SMB 协议状态机规模统计

协议状态机	状态数	状态转移数	终止状态数
NS_{SMB}	17	62	5
S_{SMB}	12	39	3

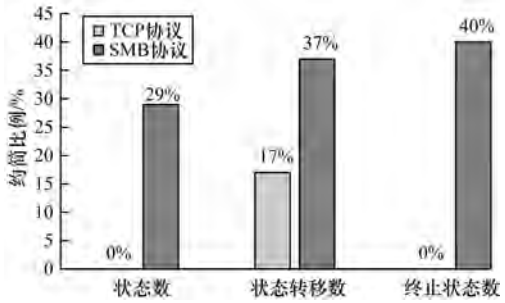

图 11 2 种协议的约简比例

3.4 协议状态机性能评估

将 TCP 协议和 SMB 协议的混合会话集作为测试数据集，如表 11 所示，利用推断的 TCP 协议状态机和 SMB 协议状态机分别对测试数据集中的会话进行识别，准确率和召回率结果分别如表 15、表 16 所示。由表 15、表 16 可以看出，本文协议状态机对测试数据集有较好的识别效果，与真实协议状态机接近。

表 15 准确率统计结果

协议状态机	识别的会话数	正确的会话数	准确率/%
S_{TCP}	198	198	100
S_{SMB}	90	90	100

表 **16**　召回率统计结果

协议状态机	真实的会话数	正确的会话数	召回率/%
S_{TCP}	200	198	99
S_{SMB}	100	90	90

4　结束语

本文提出一种基于状态相关字段的二进制私有协议状态机推断方法 BSMISRF。在 TCP 协议和 SMB 协议数据集上进行测试,结果验证了该方法良好的识别性能。在实际应用中,BSMISRF 方法对字段划分的准确度要求较高,因此,对二进制协议字段格式提取进行分析并获取高准确度的字段划分结果,将是下一步的研究方向。

参考文献

[1] 吴礼发,洪征,潘☒. 网络协议逆向分析及应用[M]. 北京:国防工业出版社,2016.

[2] 吴礼发,王辰,洪征,等. 协议状态机推断技术研究进展[J]. 计算机应用研究,2015,32(7):1931-1936.

[3] 王军. 基于 EDSM 的二进制协议状态机逆向[D]. 哈尔滨:哈尔滨工业大学,2016.

[4] TRIFILO A,BURSCHKA S,BIERSACK E. Traffic to protocol reverse engineering[C]//Proceedings of IEEE International Conference on Computational Intelligence for Security and Defense Applications. Washington D. C., USA: IEEE Press, 2009:1-8.

[5] SHEVERTALOV M, MANCORIDIS S. A reverse engineering tool for extracting protocols of networked applications [C]//Proceedings of the 14th Working Conference on Reverse Engineering. Washington D. C., USA:IEEE Press,2007:229-238.

[6] WANG Yipeng, ZHANG Zhibin, YAO Danfeng, et al. Inferring protocol state machine from network traces: a probabilistic approach [C]//Proceedings of International Conference on Applied Cryptography and Network Security. Berlin,Germany:Springer,2011:1-18.

[7] DUNN J C. Well-separated clusters and optimal fuzzy partitions[J]. Journal of Cybernetics, 1974, 4(1): 95-104.

[8] JACCARD P. The distribution of the flora in the alpine zone[J]. New Phytologist,1912,11(2):37-50.

[9] KAUFMAN L,ROUSSEEUW P J. Finding groups in data: an introduction to cluster analysis[M]. Washington D. C., USA:John Wiley and Sons,Inc.,2009.

[10] 黄笑言,陈性元,祝宁,等. 基于状态标注的协议状态机逆向方法[J]. 计算机应用,2013,33(12):3486-3489.

[11] XIAO Mingming, YU Shunzheng, WANG Yu. Automatic network protocol automaton extraction [C]// Proceedings of the 3rd International Conference on Network and System Security. Washington D. C., USA: IEEE Press,2009:336-343.

[12] 肖明明,余顺争. 基于文法推断的协议逆向工程[J]. 计算机研究与发展,2013,50(10):2044-2058.

[13] TAO Siyu, YU Hongyi, LI Qing. Bit-oriented format extraction approach for automatic binary protocol reverse engineering[J]. IET Communications, 2016, 10(6): 709-716.

[14] COMPARETTI P M, WONDRACEK G, KRUEGEL C, et al. Prospex: protocol specification extraction [C]// Proceedings of IEEE Symposium on Security and Privacy. Washington D. C., USA:IEEE Press,2009:110-125.

[15] 张黎. 基于 Net-trace 的网络协议逆向工程方法研究[D]. 武汉:华中科技大学,2011.

[16] 肖明明,余顺争,张世龙. 文法推断网络协议状态机[J]. 科学技术与工程,2014,14(19):100-105.

编辑　吴云芳