DOI:10.7544/issn1000-1239.2015.20150568
52(10):2200-2211,2015

SPFPA:一种面向未知安全协议的格式解析方法

朱玉娜　韩继红　袁霖　陈韩托　范钰丹
(解放军信息工程大学　郑州　450001)
(zyn_qingdao@126.com)

SPFPA: A Format Parsing Approach for Unknown Security Protocols

Zhu Yuna, Han Jihong, Yuan Lin, Chen Hantuo, and Fan Yudan
(*PLA Information Engineering University*, *Zhengzhou* 450001)

Abstract　Format parsing for unknown security protocols is a critical problem that needs to be solved in the information security field. However, previous network-trace-based format parsing methods have only considered the plaintext format of payload data, and have not been suitable for security protocols which include a large number of ciphertext data. In this paper, to infer the message format of unknown security protocols from a large mount of network traces, we propose a novel format parsing approach-named SPFPA (security protocols format parsing approach). SPFPA presents a hierarchical method to extract the protocol keywords sequences using sequential pattern mining for the first time, which provides a new idea for plaintext format parsing. On this basis, SPFPA introduces a set of heuristics to search the possible ciphertext length fields, and then identifies ciphertext length fields and the corresponding ciphertext fields by using the randomness feature of ciphertext data. Finally we evaluate SPFPA on four classical security protocols, i.e. SSL protocol, SSH protocol, Needham-Schroeder (NS) public key protocol and sof protocol. Our experimental results show that without using dynamic binary analysis, SPFPA can parse true protocol format effectively, i.e. invariant fields, variable fields, ciphertext length fields and ciphertext fields, purely from network traces, and the inferred formats are highly accurate in identifying the protocols.

Key words　security protocol; protocol format parsing; sequential pattern; data mining; ciphertext feature

摘　要　针对未知安全协议的格式解析方法是当前信息安全技术中亟待解决的关键问题.现有基于网络报文流量信息的方法仅考虑报文载荷中的明文信息,不适用于包含大量密文信息的安全协议.针对该问题,提出一种新的面向未知安全协议的格式解析方法(security protocols format parsing approach, SPFPA).SPFPA首次利用序列模式挖掘方法层次化、序列化提取协议的关键词序列特征,为明文信息格式解析提供一种新的解决思路,并在此基础上给出查找协议密文长度域的启发式规则,进而利用密文数据的随机性特征确定密文域.实验结果表明,该方法在不借助任何主机运行特征的基础上,仅依靠网络报文数据即能够有效解析未知安全协议的不变域、可变域、密文长度域及相应的密文域,并具有较高的准确率.

关键词　安全协议;协议格式解析;序列模式;数据挖掘;密文信息特征

中图法分类号　TP393.08

收稿日期:2015-06-16;**修回日期**:2015-08-12
基金项目:国家自然科学基金项目(61309018)

随着密码技术的广泛应用,安全协议被大量应用在互联网各种核心、关键应用中,与安全协议相关的各种数据在网络流量中比重日益增加.与此同时,针对安全协议的新型攻击不断增加.传统方法通过对安全协议进行形式化分析或自动化验证的方法检测协议自身缺陷和潜在攻击,需要基于特定的攻击者模型和若干假设,只能给出理想情况下的安全性分析结果,对于协议运行过程中的某些动态因素往往无法准确判断.因此仅仅依靠传统方法难以真正保证安全协议在复杂、多变的实际运行环境中不存在安全风险.以经典的 SSL/TLS 协议为例,该协议已发布 5 个版本——SSL2.0,SSL3.0,TLS1.0,TLS1.1,TLS1.2.基本上每个版本都有形式化方法证明安全,但随后又发现漏洞.由此,安全协议在线动态安全性分析技术已经成为新一代信息安全技术中亟待深入研究的关键问题.

要实现协议动态安全性分析,需要在线识别信息系统中报文数据的协议类型、重构协议会话实例、获取协议当前运行状态信息,这需要以协议的格式描述信息为基础.而目前存在很多私有协议,协议细节未公开,无法基于已有的协议分析工具(例如著名的 Wireshark)分析协议安全性.为此,需要自动解析未知安全协议格式信息,为协议动态安全性分析提供支撑.

安全协议运行过程中,频繁使用数据加密、数字签名、公钥证书、散列等各种密码技术对关键信息进行加密和保护,报文包含大量密文信息.攻击者在无法解密密文的情况下,常常通过重放、转发密文进行攻击.因此,对未知安全协议进行格式解析时,不仅需要解析可用明文格式特征,还需要充分发掘和利用协议报文中包含的密文数据特征.

目前,网络协议格式解析技术主要包括 2 类方法:基于目标主机程序执行轨迹的方法和基于网络报文流量信息的方法.1)基于目标主机程序执行轨迹的方法借助特定二进制分析平台,基于目标主机上协议相关的应用程序运行状态特征解析网络协议格式,该类方法虽然可以处理加密报文,但是需要在目标主机上获取执行协议的应用程序信息,并部署特定监测工具,应用局限性较大,无法真正满足网络环境中对数据报文监测需求;2)基于网络报文流量信息的方法是使用范围更广的方法,该类方法主要以捕获的网络流量数据为分析对象,依据协议字段的取值变化频率和特征推断得到协议格式,但目前仅解析报文中的明文信息格式,不考虑协议中的密文信息格式.

针对上述问题,本文提出了一种面向未知安全协议的格式解析方法(security protocols format parsing approach, SPFPA),基于网络报文数据识别协议的不变域、可变域、密文长度域及相应的密文域.本文主要贡献有:1)给出了 SPFPA 总体框架,阐述了框架的 3 个阶段:数据预处理、关键词序列提取、密文格式解析.2)在关键词序列提取阶段,结合协议序列特点,首次提出面向协议的序列模式挖掘方法,发现具有时序关系的关键词序列集合,解析协议不变域、可变域.3)在密文格式解析阶段,以挖掘的关键词序列为基础,给出查找密文长度域的启发式规则,并利用密文随机性特征确定密文域,为密文数据信息的有效利用提供了新的解决思路.

1 相关工作

本节主要介绍现有的 2 类协议格式解析方法.

1) 基于目标主机程序执行轨迹的方法[1-4].是以数据解析过程中协议相关应用程序的执行轨迹(即指令执行序列)为分析对象的技术,目前可实现对密码协议的逆向.Wang 等人[5]提出了基于缓冲区数据生命周期的解密报文识别方案 ReFormat,并利用动态二进制平台 Valgrind 开发相应的分析工具,对程序中网络消息的加密函数进行捕捉,达到识别密码算法和解析明文信息格式的目的,但不适用于解密过程与解析过程交替进行的安全协议.为解决该问题,Dispatcher[6-7]沿用了 ReFormat 中缓冲区数据生命周期的算术指令统计思想,识别力度细化到编码函数级(包括加/解密函数、散列函数等),允许多个加密过程与解析过程交替,但识别策略的通用性和准确性需要进一步验证.该类方法要求在指令级监控协议实体对报文的解析过程,其实现复杂且可能无法获得协议实体的控制权,应用受限.

2) 基于网络报文流量信息的方法.是指依据协议在线运行数据推断协议具体语法或语义的过程.与基于目标主机程序执行轨迹的方法相比,该类方法在实际应用中只需捕获协议报文,局限性较小,使用范围较广.文献[8-10]研究报文载荷的字符串特征,提取由公共字符串组成的子序列作为应用特征.文献[11]利用网络流量文本内容的语义特点,提取文本协议字符串特征.鉴于数据挖掘是目前最有效的数据分析手段,可用于发现大量数据所隐含的各种规律,人们将经典关联规则挖掘方法应用于协议

特征提取,挖掘协议会话前 N 个字节的固定位置特征,并取得初步研究成果[12-14].上述方法主要提取未知协议的识别特征,对于未知安全协议在线安全性分析而言,还需要进一步解析和推断协议格式,以便重构协议会话实例.PI 项目(protocol information project)[15]借助于生物信息学中序列比对算法,对目标协议的结构信息进行解析.文献[16-17]提出了以循环聚类为核心思想的协议逆向方案,采用基于类型的序列比对算法,实现了针对性更强的报文格式逆向;文献[18]结合字节类型推断和 PI 的序列比对思想,得到具有共同类型的报文,并最终实现了部分域语义属性的推断.上述方法主要利用序列比对技术,易受数据集噪音和比对顺序影响,导致早期不频繁信息的丢失,不适用于序列较长、格式复杂的协议.此外,目前该类方法仅考虑协议中的明文格式解析,不能完全适用于包含大量密文信息的安全协议.

2 未知安全协议格式解析问题描述

定义 1. 协议会话.协议参与方之间的一次协议完整交互过程,包括协议一次通信建立和结束之间的所有报文.

定义 2. 协议报文载荷序列.组成报文载荷的字节的排列次序,记为 $p=\langle s_1,s_2,\cdots,s_i,\cdots,s_n\rangle$,其中,$s_i$ 属于 2 位 16 进制数组成的集合 $\Sigma=\{00,01,\cdots,FF\}$,$1\leqslant i\leqslant n$,$n=|p|>0$,$|p|$ 为序列 P 的长度.同一种协议的报文载荷序列组成的有限序列集称为协议序列集合.

定义 3. 协议关键词.在报文格式中用于标识协议报文类型和传递相关控制信息的协议字段.文献[1]指出,一般而言,绝大多数的网络协议都会在报文格式中定义一个或多个关键词,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.

协议的关键词可以是协议包头中的特征字符串,包括协议名称、版本号等,也可以是协议控制信息的特征字符串,包括命令码、标识信息等.按照关键词出现的位置,可将关键词分为固定偏移关键词和非固定偏移关键词.固定偏移关键词在协议报文中位置固定;非固定偏移关键词在协议报文中位置可变.

关键词在协议报文中频繁出现,是组成协议特征的基本元素.现有基于关联规则挖掘的方法不考虑事务的顺序性,不能提取协议有序的非固定偏移关键词.序列模式挖掘由 Agrawal 和 Srikant[19]首次提出,用于发现序列集合中项集的时序关联规则,是对关联规则挖掘的进一步推广,目前已广泛应用于 DNA 序列分析、交易数据分析、Web 用户访问模式预测等领域.具体而言,给定序列集合和支持度阈值,序列模式挖掘就是找出所有的频繁子序列,即在序列集合中出现次数不低于最小支持度阈值的子序列.

本文首先基于序列模式挖掘方法提取协议序列集合中具有时序关系的关键词序列.关键词字段为协议的不变域,前后 2 个关键词之间为协议的可变域.随后对可变域进一步解析,确定密文长度域以及相应的密文域,从而得到协议格式,如图 1 所示:

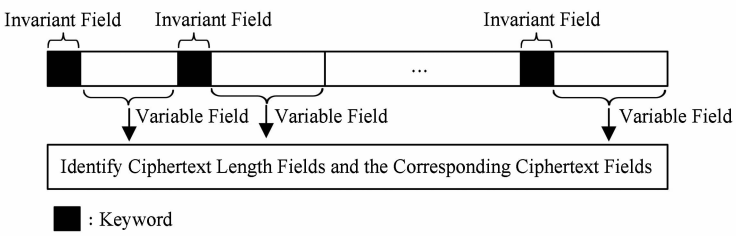

Fig. 1 Security protocol format parsing problem.

图 1 未知安全协议格式解析问题

3 SPFPA 总体框架

SPFPA 总体框架包括 3 个阶段,如图 2 所示.

1) 数据预处理

为获取纯净的同类协议流量,在网络各个主机上运行该协议的应用程序,并采用 Wireshark 捕获报文;对公开数据集,则根据流量过滤规则过滤出指定协议的网络流量.随后对同类协议流量进行处理,进一步获取相同类型的报文.具体而言,同一种协议相同位置的消息大都具有相似的报文格式.按照五元组(源 IP,目的 IP,源端口,目的端口,传输层协议类型)进行分流,将不同流相同位置的数据报文组成一个报文组,并对每一个报文组中的报文提取载荷

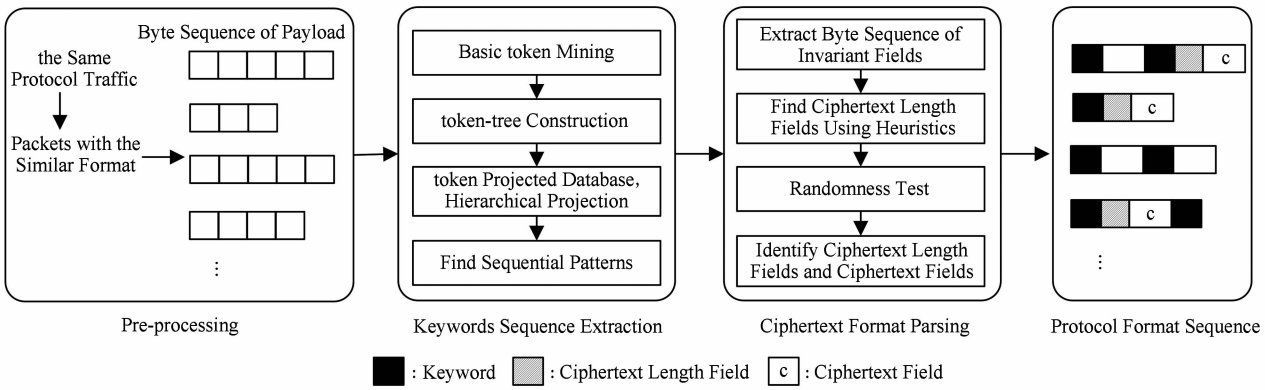

Fig. 2 SPFPA architecture.

图 2 SPFPA 总体框架

字节序列,用于后续格式解析.

2) 协议关键词序列提取

基于序列模式挖掘提取协议中具有时序关系的关键词序列,识别协议的不变域和可变域.具体而言:

① 挖掘报文载荷序列集合中具有约束条件的频繁模式(基本 token)——固定位置的频繁模式 1-position-token 和相邻 *min_len* 个字节的频繁模式 min-length-token;

② 结合网络流量字节有序的特征构建 token-tree,保存序列的 token 信息;

③ 基于 token-tree 建立 token 投影库,层次化、序列化挖掘协议序列模式,生成协议关键词特征.

3) 密文格式解析

在确定关键词序列的基础上,对前后关键词之间的可变域进一步处理.

① 结合协议密文特点,给出查找密文长度域的启发式规则;

② 依据密文数据的随机性特点确定密文域和密文长度域.

4 基于序列模式挖掘的关键词序列提取方法

本节结合协议序列特点和实际应用背景,分析面向协议的序列模式挖掘特点和需求,在此基础上提出协议序列模式挖掘方法,提取具有时序关系的关键词序列.

4.1 面向协议的序列模式挖掘特点

现有序列模式挖掘方法仅考虑序列元素的值,忽略序列元素之间的顺序,认为在序列中相邻项之间是独立的、无依赖关系的.而协议序列模式具有特定需求,具体如下:

1) 协议中相邻字节频繁模式和固定位置频繁模式更具有实际意义.

① 相邻位置的频繁模式更可能具有语义信息.协议的关键词一般由多个字节组成(例如 SSL 中协议版本号关键词"0x03 0x01",SSH 中协议名称关键词"0x53 0x53 0x48"等).

② 协议中固定偏移关键词与位置密切相关.例如 SSL 协议中 client_hello 报文载荷的第 1 个字节 0x16 表示协议名称,而其他位置的 0x16 表示其他意义或者仅是偶然的组合.

由于协议本身的规范以及上下文之间的关系,单个字节值可能在一个报文或者不同报文中频繁出现,并且很多协议采用零填充机制,报文存在很多"00"字节.直接进行序列模式挖掘,会产生很多不属于协议特征的冗余频繁短模式,导致效率不高.因此应充分利用相邻字节频繁模式和固定位置频繁模式,降低协议序列模式挖掘的复杂度.

2) 对协议关键词序列的挖掘是一个层次化、序列化的过程,无需挖掘所有的序列模式.

协议报文载荷数据是有序、连续的字节序列,关键词之间大都存在明确的顺序关系.因此应结合网络流量字节有序的特征层次化挖掘关键词序列.

综上所述,为满足协议序列挖掘的特定需求,应设计面向协议的序列模式挖掘算法,提取协议关键词序列.

4.2 基本 token 挖掘

为挖掘固定位置频繁模式和相邻字节频繁模式,提取 2 类基本 token——1-position-token 和 min-length-token.前者针对固定偏移关键词,挖掘固定位置的单个频繁字节;后者针对非固定偏移关键词,挖掘最小长度为 *min_len* 的频繁字节序列.

定义 4. 支持度 *sup*.给定协议序列集合 P(见定义 2),$p \in P$ 为其中的一个序列,若序列 x 为 p 的

子序列,则 $s(x)=1$,否则 $s(x)=0$. x 在 P 上的支持度记为 $sup(x)=\sum_{p\in P}s(x)/|P|$,其中 $|p|$ 为 P 中包含的 p 总数.

定义 5. 1-position-token. 给定最小支持度阈值 min_sup,若某个字节始终出现在固定位置,且在协议序列集合中的支持度不小于 min_sup,则称该字节为 1-position-token.

定义 6. min-length-token. 给定最小支持度阈值 min_sup 和最小长度参数 min_len,若 s 为长度为 min_len 的相邻字节序列,且在协议序列集合中的支持度不小于 min_sup,则称序列 s 为 min-length-token.

1) 1-position-token 挖掘

对报文载荷原始序列进行编码. 每个字节同样作为序列中的一个元素,采用 5 个字符表示. 前 3 个字符为该字节偏移的 16 进制表示,从零开始计数,其中字节偏移是指字节在所属报文载荷的序号位置(网络数据传输采用 IP 分片技术,数据包长度小于等于最大传输单元(maximum transmission unit, MTU). 以太网中 $MTU=1500$,相应的报文载荷长度为 0~1460. 为此,采用 3 位 16 进制表示字节偏移). 第 4,5 个字符为该字节的 16 进制表示值. 随后对编码后的序列集合进行频繁 1-项挖掘,根据频繁 1-项确定相应的 1-position-token 及其位置信息.

2) min-length-token 挖掘

min-length-token 中的字节可能为 1-position-token. 为挖掘与 1-position-token 集合不相交的 min-length-token,对报文载荷原始序列中 1-position-token 以外的字节进行编码,如图 3 所示:

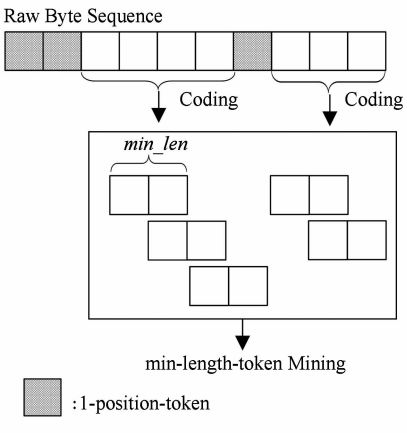

Fig. 3 min-length-token mining.

图 3 min-length-token 挖掘

具体而言,记前后 2 个 1-position-token 之间的序列为 $X=b_1,b_2,\cdots,b_k$(对最后一个位置的 1-position-token,X 为该 token 之后的字节序列). 以 min_len 长度字节为元素对序列 X 进行编码,编码后的序列 $X'=b_1\cdots b_{min\_len}$, $b_2\cdots b_{min\_len+1}$, …, $b_{k-min\_len+1}\cdots b_k$. 随后对编码后的序列集合挖掘 min-length-token.

定义 7. token 连接. 若 t_1 和 t_2 为基本 token, $t_1=b_1b_2\cdots b_l$, $t_2=b'_1b'_2\cdots b'_{l'}$,且 t_1 的后 m 个字节和 t_2 的前 m 个字节相同($m<l, m<l'$),则称 t_1 和 t_2 的连接为 $b_1b_2\cdots b_mb'_{m+1}\cdots b'_{l'}$,记为 $t_1\|_m t_2$.

命题 1. 记 $T_k=b_1b_2\cdots b_n(n>min\_len)$ 为 n 个字节长度的 token,则 T_k 可以表示成 t 个基本 token 的连接($\left\lceil\frac{n}{min\_len}\right\rceil\leqslant t\leqslant n-min\_len+1$).

证明. T_k 为频繁序列,则 T_k 的子序列 $b_ib_{i+1}\cdots b_{min\_len+i-1}$($1\leqslant i\leqslant n-min\_len+1$)也为频繁序列,为 min-length-token. T_k 对应 $n-min\_len+1$ 个 min-length-token.

当 $n=min\_len+k$ 时,分 2 种情况讨论:

① 当 $k \bmod min\_len\neq 0$ 时,T_{min_len+k} 最小连接集合为 $\left\lceil\frac{min\_len+k}{min\_len}\right\rceil$ 个互不相交的 min-length-token 和 $b_kb_{k+1}\cdots b_{min\_len+k}$ 的连接. $\left\lceil\frac{n}{min\_len}\right\rceil=\left\lceil\frac{min\_len+k}{min\_len}\right\rceil=\left\lceil\frac{min\_len+k}{min\_len}\right\rceil+1$. 最大连接集合为 $k+1=n-min\_len+1$ 个 min-length-token 的连接. 命题 1 成立.

② 当 $k \bmod min\_len=0$ 时,T_{min_len+k} 最小连接集合为 $\left\lceil\frac{min\_len+k}{min\_len}\right\rceil$ 个互不相交的 min-length-token 的连接. $\left\lceil\frac{n}{min\_len}\right\rceil=\left\lceil\frac{min\_len+k}{min\_len}\right\rceil$. 最大连接集合为 $k+1=n-min\_len+1$ 个 min-length-token 的连接. 命题 1 成立. 证毕.

4.3 token-tree 构建

结合网络流量字节有序的特点,借鉴 FP-tree[20] 提出一种新的树结构 token-tree,用于存储协议序列的基本 token 信息. 具体而言,将协议序列中基本 token 的频度、偏移、相邻、关联信息压入到一个 token-tree 中,为挖掘协议序列模式提供基础.

定义 8. token-tree.

① 由 3 部分组成:树根(记为 root)、基本 token 构成的前缀子树、基本 token 头表.

② 除根结点以外的每一个结点由 5 个域组成:(*token-name*, *count*, *offset*, *overlap*, *pnode*).

其中,*token-name* 记录该结点所代表的基本 token 名称;*count* 记录包含该 token 结点的序列数目;*offset* 记录 token 的位置,若为固定偏移,则记录偏移位置,否则为 null;*overlap* 记录与父结点之间的重叠字节数;*pnode* 表示父结点指针. 由命题 1 可知,为使 token-tree 能够包含所有基本 token,*overlap* 取值如下:

$$overlap=\begin{cases}0, & \text{与父结点 token 位置相邻且不相交;}\\ min\_len-1, & \text{与父结点 token 位置相邻,}\\ & \text{且重叠字节为 } min\_len-1;\\ \text{null}, & \text{与父结点 token 位置不相邻.}\end{cases}$$

③ token 头表的每个元组由 2 部分组成:*token-name*,*Node-link*(指向 token-tree 中有相同 *token-name* 的第 1 个结点). 相同 token 的指针连接在一条链上.

④ 鉴于协议字节序列的有序性,token-tree 各分支上的结点按照在协议序列中出现的顺序关系排列. 在基本 token 构成的前缀子树中,对每个序列检查是否与 token-tree 中已有分支表示的序列共享前缀,如果否,则在根结点创建一个分支.

满足以上特征的树称为 token-tree.

一个 token-tree 树结构实例如图 4 所示:

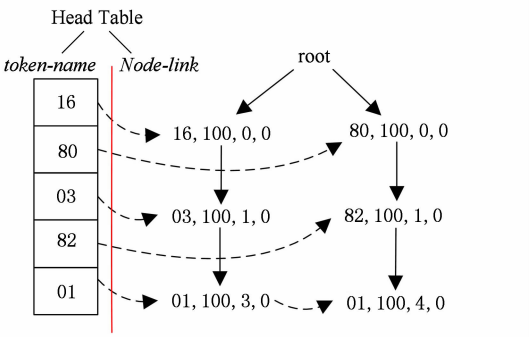

Fig. 4 token-tree example.

图 4 token-tree 例子

定义 9. token 序列. 记 token-tree 结点 $node_k=(t_k, count_k, offset_k, overlap_k, pnode_k)$. 对 token-tree 分支上任意 2 个结点 $node_i$ 和 $node_j$ 而言,结点之间的路径 $node_i\rightarrow node_{i+1}\rightarrow\cdots\rightarrow node_j$ 表示 token 序列 $t_i\|_{overlap_{i+1}}t_{i+1}\|_{overlap_{i+2}}\cdots\|_{overlap_j}t_j$.

token-tree 具有以下性质:

性质 1. 在 token-tree 中,除根结点外,任何结点的支持度都不小于其子孙结点的支持度.

证明. 当多个序列包含相同的前缀 token 时,token-tree 采用同一个分支保存这些相同的 token. 因此,只要协议中某序列其包含 token-tree 上某个结点,则必包含该结点的所有祖先结点. 相应地,祖先结点的支持度大于等于该结点的支持度. 证毕.

性质 2. 对从根结点 root 到某 token 结点 $node_k$ 的分支路径而言,该路径对应序列的支持度等于结点 $node_k$ 的支持度.

证明. 协议序列是有序序列. 该路径遵循协议的有序性,由性质 1 可知,当分支包含结点 $node_k$ 时,则必包含 $node_k$ 的所有祖先结点 $node_1, node_2, \cdots, node_{k-1}$. 因此 root$\rightarrow node_1\rightarrow node_2\rightarrow\cdots\rightarrow node_k$ 对应序列的支持度等于结点 $node_k$ 的支持度. 证毕.

4.4 协议序列模式挖掘

根据协议报文从左到右层次解析的特性,基于 token-tree 生成 token 投影库,层次化挖掘协议关键词序列.

4.4.1 token 投影库

Pei 等人[21]提出的 PrefixSpan 算法(prefix projected sequential pattern mining)不需要产生候选频繁项集,是目前序列模式挖掘中效率较高的一种方法. 其基本思想是:频繁序列的前缀序列也是频繁的,且该序列可以由其前缀子序列经有限步扩展得到. 因此先扫描序列集合查找单个频繁项,产生投影数据库的集合,其中每个投影数据库关联一个频繁项,然后重复上述过程对每个投影库进行单独挖掘.

受 PrefixSpan 算法启发,本文构建协议 token 投影库,挖掘协议序列模式.

定义 10. token 投影、token 投影库. 给定 2 个 token 序列 x_1 和 x_2,$x_1=t_1\|_{l_1}t_2\|_{l_2}\cdots t_{n-1}\|_{l_{n-1}}t_n$,$x_2=t'_1\|_{l'_1}t'_2\|_{l'_2}\cdots t'_{m-1}\|_{l'_{m-1}}t'_m$,$(m<n, l_i$ 为 t_{i+1} 和 t_i 的重叠字节数,l'_i 为 t'_{i+1} 和 t'_i 的重叠字节数),x_2 是 x_1 的子序列,$x_2\neq x_1$,若存在 token 序列 $x_3=t''_1\|_{l''_1}t''_2\|_{l''_2}\cdots t''_{m-1}\|_{l''_{m-1}}t''_m$,$(m<k=n)$,且满足:$x_2$ 是 x_3 的前缀,x_3 是 x_1 中满足上述条件的最大子序列,则称 x_3 是 x_1 关于 x_2 的投影. 协议序列数据库中所有 x_3 中 x_2 的后缀序列组成的集合称为 x_2 的投影数据库.

例如对于序列 $x_1=16\|_0 03\|_0 01\|_{\text{null}}$c014$\|_1 1400$,其子序列 $x_2=03\|_0 01$ 的投影为 $x_3=03\|_0 01\|_{\text{null}}$c014$\|_1 1400$. 在 x_2 的投影数据库中,该序列对应 c014$\|_1 1400$.

根据上述定义,协议序列任一 token 结点 t_i 的投影为 token-tree 中该结点至叶子结点路径对应的序列. 协议序列库中 t_i 的投影数据库为 token-tree 中与 t_i 相关的结点对应的后缀,即 t_i 对应结点之后的所有子树对应序列的集合.

4.4.2 基于 token-tree 的序列挖掘

PrefixSpan 方法对所有频繁 1-项进行迭代投影,而协议序列是有序的,在 token-tree 一个分支中,任一结点的投影包含其子孙结点的投影.因此无需产生所有频繁 token 的投影,而是按照 token-tree 的分层顺序进行层次化迭代投影,如图 5 所示.具体而言:

1) 扫描 token-tree,查找 token-tree 中第 1 层 token 的集合 1-tokenset,该集合记为 $Token_1$.

2) 对 $Token_1$ 中基本 token 分别进行投影.在 $token_1$($token_1 \in Token_1$)的投影库中,根据 token-tree 分支进行纵向搜索,查找与之相连的第 2 层 token(即 $token_1$ 的子孙结点,记为 $token_2$)的支持度.

① 若在 $token_1$ 的投影库中,$token_2$ 不小于给定的支持度阈值 min_sup,则对 $token_2$ 继续进行迭代投影.即在 $token_2$ 的投影库中,纵向搜索 token-tree 中与之相连的第 3 层 token(即 $token_2$ 的子孙结点,记为 $token_3$)的支持度.

② 若在 $token_1$ 的投影库中,$token_2 < min\_sup$,根据 4.3 节性质 1,与 $token_2$ 相连的子孙结点的支持度也小于 min_sup,因此不再对该分支进行层次化投影.

3) 在 $token_i$ 的投影库中,根据 token-tree 中分支,查找与之相连的第 $i+1$ 层 token(即 $token_i$ 的子孙结点,记为 $token_{i+1}$)的支持度.依此类推,直到到达 token-tree 各个路径中的叶子结点.

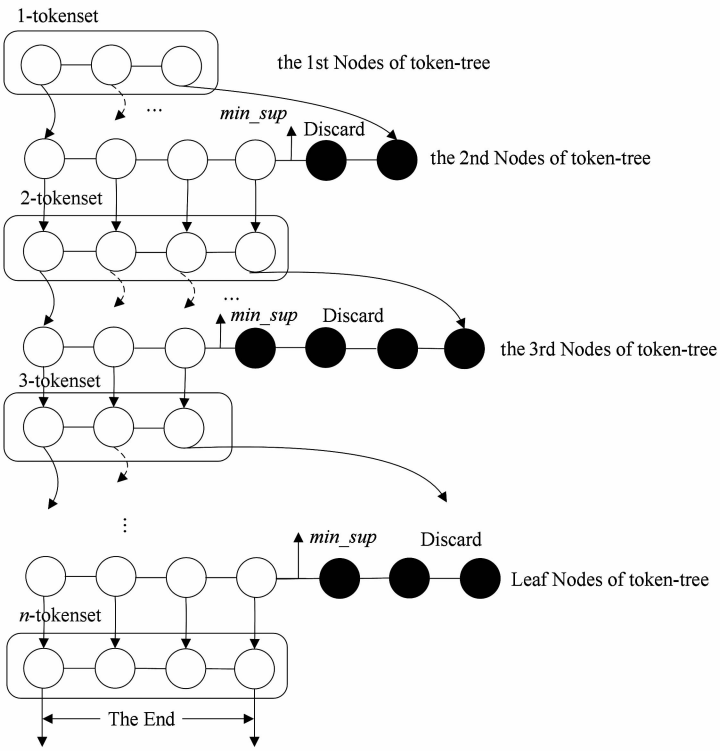

Fig. 5 Sequential pattern mining.

图 5 序列模式生成

只有存在于同一个会话中的关键词序列才能构成协议特征.在生成报文序列模式的基础上,将每个会话作为一个序列,每个报文载荷的序列模式作为序列的一个元素,进一步挖掘协议会话的序列模式,提取协议会话关键词序列.

5 基于关键词序列的密文格式解析方法

密文长度与所采用的密码算法、明文长度、密钥长度相关,即使在协议规范中密文格式确定,例如经典 sof 协议中的$\{h(N_b),N_a,A,K\}_{K_{ab}}$,在协议报文中该密文长度也是变化的.因此在密文域前面,通常采用长度域来标识密文的长度,以便接收方可以确定密文的开始位置和接收位置,进而解析密文.

本节在提取关键词序列的基础上,对前后 2 个关键词之间的可变域进一步进行解析,识别密文长度域和相应的密文域.

5.1 密文长度域启发式识别规则

密文长度域的识别策略基于以下启发式规则:

1) 密文长度域一般采用 1~2 B 长度的 16 进制

数值表示.密码运算计算量较大,并且密码算法不等同与安全.根据协议设计的加密准则,协议设计者需要正确地使用密码算法,避免冗余.因此,协议密文具有明确的规范,其长度在一定的范围内.结合协议实际情况,通常情况下,协议密文长度为 128～4 096 b.由于密文长度为 8 b 的倍数,相应的长度域十进制取值范围为 16～512.

2) 密文长度域及其对应的密文域一定在前后 2 个关键词之间.由于密文是随机的,在密文域中不存在频繁项,也不可能存在协议关键词,因此密文长度域值一定小于前后 2 个关键词之间的长度.

3) 密文长度域与密文的长度相关联.当密文的长度发生变化时,长度域随之改变.

4) 密文长度域随后的密文具有随机性.连续 5 B 第 1 位同时为 0 或者同时为 1 的概率为 $\left(\frac{1}{2}\right)^5=0.03125<0.05$,为小概率事件.对文本协议而言,其明文为 ASCII 码,取值范围为 0～127,其字节第 1 位为 0,明文中经常出现 5 个连续 ASCII 码.对于二进制协议,由于协议规范具有特定语义,也经常出现连续 5 B 的第 1 位为 0 的情况.因此,可利用该特点大致判断密文域位置.

5.2 密文长度域及相应密文域解析方法

密文长度域及相应密文域的解析方法如图 6 所示:

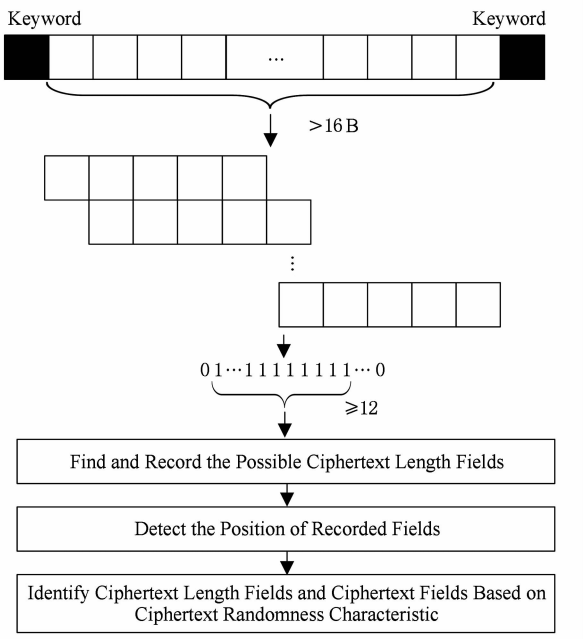

Fig. 6 Parse ciphertext length field and its ciphertext field.

图 6 密文长度域及相应密文域解析方法

具体步骤如下:

Step1. 对每个序列查找可能的密文域.判断前后 2 个关键词之间的长度,若大于 16 B,对 2 个关键词之间的字节进行编码.连续 5 B 的第 1 位相同则编码为 0,不同则编码为 1.鉴于密文的随机性,密文数据编码后出现 0 为小概率事件,又由于假设密文最小长度为 16 B,因此若连续为 1 的长度大于等于 12,则关键词之间可能存在密文域,执行 Step2.

Step2. 查找可能的密文长度域.若关键词之间的字节序列可能存在密文域,分别计算字节序列中各个字节值 n 和连续 2 B(字节以及相邻的前一字节)对应的值 m.记某一字节至下一关键词之间的长度为 l(单位为 B).① 若 $n\leqslant l$,且 $n\geqslant 16$ B,记录(字节所在起始关键词、字节所在终止关键词、字节值 n、n 对应的字节序列);② 若 $m\leqslant l$,且 $m\leqslant 512$ B,记录(字节所在起始关键词、字节所在终止关键词、字节值 m、m 对应的字节序列);③ 若 $m=n$,(例如“00 C0”和“C0”都作为长度域标识密文域),长度域本身重叠,优先选取 m 作为长度域.

Step3. 基于密文随机性确定密文格式.在序列中记录的 n 值和 m 值可能为一种巧合,进一步进行处理.若在所有序列的相同 2 个关键词之间存在记录的字节值和相应的字节序列,计算字节序列累积和校验[22]的 *p-value* 值,并设定显著性水平 α.若通过随机性测试,则该字节序列为密文域,字节值为密文长度域.

6 实验评估

SSL 协议和 SSH 协议是网络中广泛应用的安全通信协议;NS(Needham-Schroeder)公钥协议和 sof 协议则属于经典基础安全协议.本文选取这 4 个经典的安全协议进行实验,结果表明 SPFPA 方法所解析的格式具有较高的准确率.

6.1 实验数据

协议流量数据集如表 1 所示,主要由 3 部分组成:1)来源于广域网和公开数据集 MACCDC[23],为包含 SSL 协议的网络报文流量;2)来源于公开数据集 InfoVisContest[24],为包含 SSH 协议的网络报文流量;3)由实验室局域网环境产生,为包含 NS 公钥协议和 sof 协议[25]的网络报文流量.其中 NS 公钥协议、sof 协议的应用程序采用 Spi2Java[25]工具生成,并在各个主机上运行.

Table 1 Protocol Dataset

表 1 协议数据集

Protocol	Flow Number	Packet Number	Data Source
SSL	49 300	350 874	WAN, MACCDC[23]
SSH	270 548	846 382	InfoVisContest[24]
NS	2 000	24 000	LAN
sof	2 000	24 000	LAN

Wireshark 软件内嵌一个 Lua 语言执行引擎，基于 Lua 脚本可以获取 Wireshark 提供的与协议相关的信息(例如通信双方 IP、端口、载荷内容等). 在 Wireshark 捕获报文后，基于 Lua 脚本获取报文原始数据，并利用 SPFPA 方法进行实验.

6.2 参数确定

针对每个协议，选取完整的 100 个会话，用于解析协议格式. 实验需要确定 3 个重要的参数，分别为 min-length-token 的 *min_len* 值、支持度阈值 *min_sup* 以及显著性水平 α 值.

其中，*min_len* 和 *min_sup* 参数对关键词提取的质量产生关键影响，参数设置太小会产生很多冗余频繁模式，太大会漏掉很多特征. 文献[26]指出绝大多数识别特征包含于流中第 1 条报文内，本文依据第 1 条报文选取阈值. 图 7 为协议第 1 条报文中 1-position-token 数量随 *min_sup* 的变化关系，图 8 为基本 token 数量随 *min_sup* 和 *min_len* 的变化关系. 由图 7 和图 8 可知，当支持度 $0.6 \leqslant sup \leqslant 0.9$，挖掘的基本 token 数量趋于稳定，设置 $min\_sup = 0.6$. 当 $min\_len = 1$ 时，协议提取到很多冗余频繁项；当 $min\_len \geqslant 2$，协议提取到的基本 token 数量变化不大，设置 $min\_len = 2$.

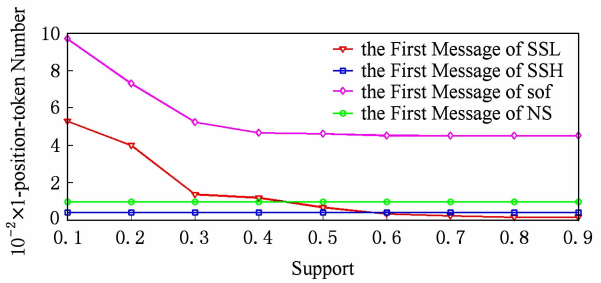

Fig. 7 1-position-token number under different *min_sup*.

图 7 1-position-token 数量随 *min_sup* 的变化关系图

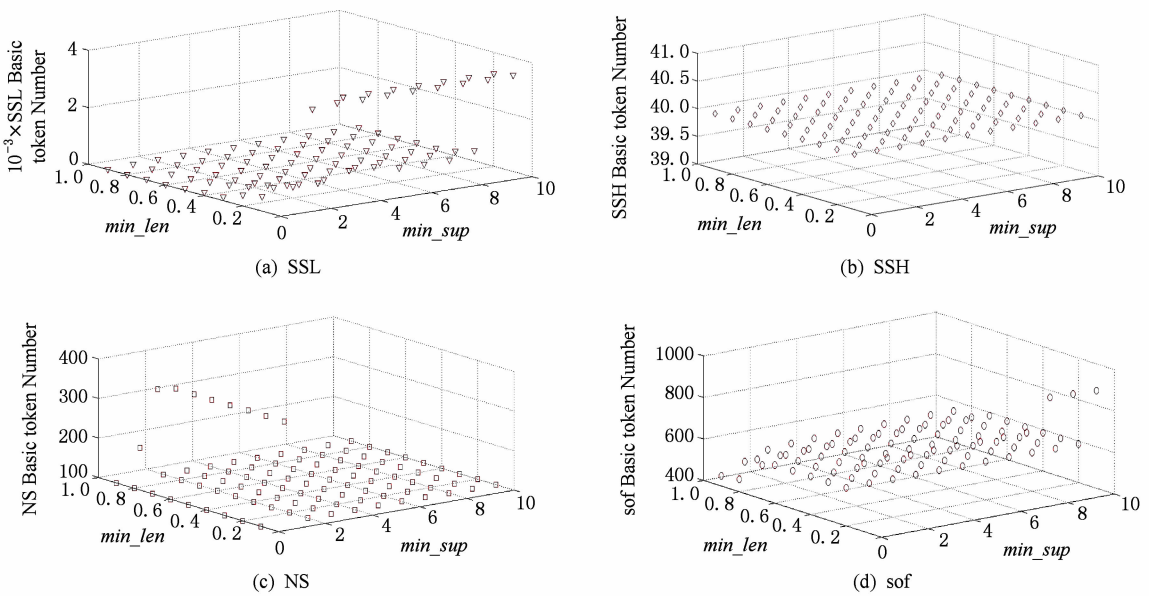

Fig. 8 Basic token number under different *min_sup* and *min_len*.

图 8 基本 token 数量随 *min_sup* 和 *min_len* 的变化关系

显著性水平 α 值主要用于确定密文域，太小会导致漏报，太大会造成误报. 根据密文算法随机性测试的常用值，本文设定 $\alpha = 0.1$.

6.3 实验结果

现有基于网络报文流量信息的方法使用不公开的数据集，且主要解析协议的明文格式特征，难以和本文方法进行比较. 为检验 SPFPA 方法的正确性，将本文解析的报文格式与公开的协议规范及 Wireshark 解析结果进行比较，例如图 9 为所解析的协议第 1 条报文格式. 结果表明 SPFPA 方法可以较好地解析未知协议格式.

为进一步验证本文方法的有效性，针对每个协议随机选取完整的 M 个会话作为训练集，用于解析协议格式；其余部分作为测试集，用于评估所解析的

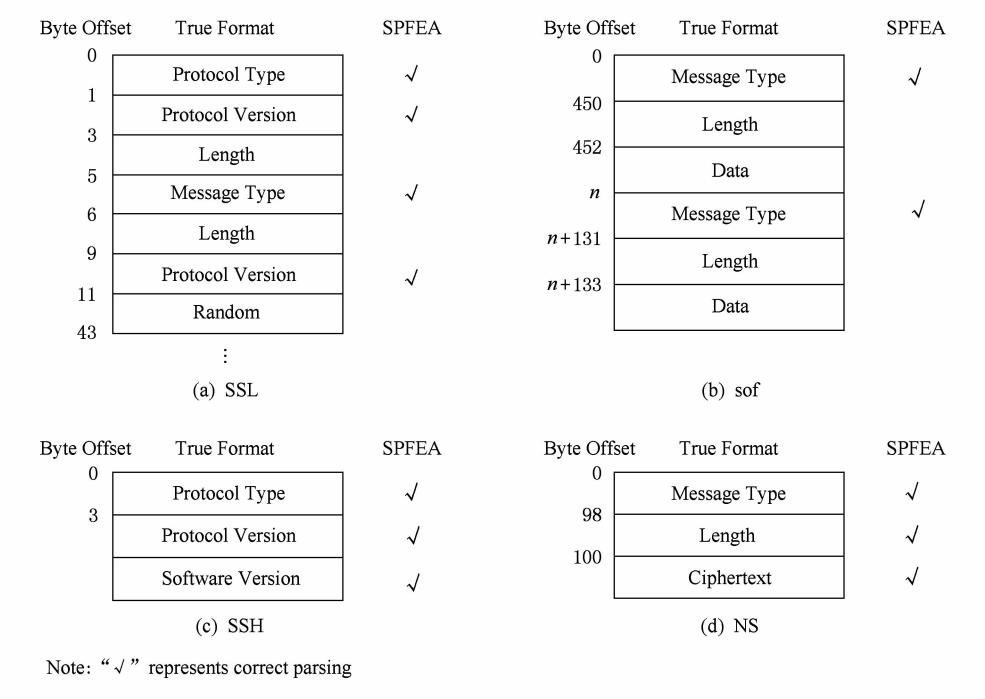

Fig. 9 Protocol format parsing result of the first packet.

图 9 协议第 1 条报文格式解析结果

格式.本文采用如下性能指标,记测试集中某协议 A 的样本数目为 N,N_1 表示被正确识别为 A 的样本数,N_2 表示非 A 被错误识别为 A 的样本数,识别率为 N_1/N,误识别率为 $N_2/(N_1+N_3)$.识别率越高,误识别率越低,相应的识别效果越好.

不同训练样本数的识别率如图 10 所示.在小样本情况下,协议会话类型偏少,所解析的格式只能代表协议部分类型的会话,识别率偏低.随着 M 增加,选取的会话类型增多,解析的格式特征更为精确,识别率也随之增加.当 $M=100$ 时,识别率基本在 94%以上.另一方面,不同训练样本数目下,都可以提取有效的协议格式特征,与其他协议可以相区分,误识别率都为 0.由此可知,本文方法可以较好地识别协议.

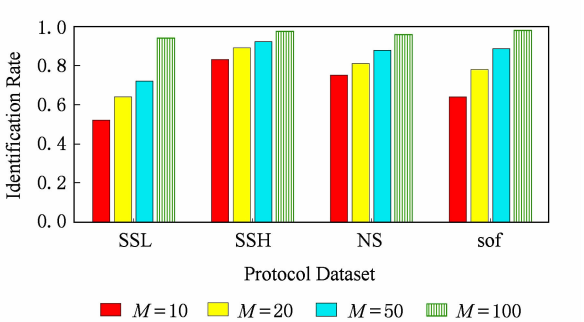

Fig. 10 Identification rate under different training set size.

图 10 训练集不同大小时的识别率

7 未来工作

SPFPA 虽然可以有效提取未知安全协议格式,但还存在一定的局限性:

1) 采用流中相同偏移的报文进行关键字提取.对多方安全协议而言,一次会话过程分布在多个单向流中;对双方安全协议而言,在网络传输中可能存在丢包、乱序等情况.因此不能完全保证流中相同偏移的报文具有相同格式,在以后工作中将利用数据包大小、方向、偏移位置等特征对同一类协议流量进行聚类,获取更精确的相同类型报文组,进一步提高关键词挖掘的准确性.

2) 参数 min_len 和 min_sup 通过实验测试进行选取.对于不同的安全协议,由于其关键词分布不同,参数设置可能不同.本文参数值对其他安全协议可能并不是最优值.尽管可以通过不同测试重新获取最优值,但此过程需要人工参与,比较耗时.应设计自适应选取参数机制,随协议的不同自适应调整参数.

3) 仅解析协议的语法格式,需在此基础上构建未知安全协议状态机,给出协议的状态转换过程,刻画协议的行为关系,为协议动态安全性分析进一步提供前提和基础.

8 结束语

本文针对未知安全协议格式解析问题提出SPFPA方法.该方法首先基于序列模式挖掘提取协议的关键词序列,识别协议的不变域、可变域,为明文信息格式解析提供一种新的解决思路;随后依据协议密文特点,在可变域中解析密文长度域和相应密文域;最后对4个经典安全协议进行实验,结果表明SPFPA基于网络报文数据即可有效解析未知安全协议格式,并具有较高的准确率.

参 考 文 献

[1] Caballero J, Yin H, Liang Zhenkai, et al. Polyglot: Automatic extraction of protocol message format using dynamic binary analysis [C] //Proc of the 14th ACM Conf on Computer and Communications Security. New York: ACM, 2007: 317-329

[2] Cui Weidong, Peinado M, Chen K, et al. Tupni: Automatic reverse engineering of input formats [C] //Proc of the 15th ACM Conf on Computer and Communications Security. New York: ACM, 2008: 391-402

[3] Comparetti P M, Wondracek G, Kruegel C, et al. Prospex: Protocol specification extraction [C] //Proc of the 30th IEEE Symp on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, 2009: 110-125

[4] Pan Fan, Hong Zheng, Zhou Zhenji, et al. Protcol format extraction at semantic level [J]. Journal on Communications, 2013, 33(10): 162-173 (in Chinese)
(潘璠, 洪征, 周振吉, 等. 语义层次的协议格式提取方法[J]. 通信学报, 2013, 33(10): 162-173)

[5] Wang Zhi, Jiang Xuxian, Cui Weidong, et al. ReFormat: Automatic reverse engineering of encrypted messages [C] // Proc of the 4th European Symp on Research in Computer Security. Berlin: Springer, 2009: 200-215

[6] Caballero J, Poosankam P, Kreibich C, et al. Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering [C] //Proc of the 16th ACM Conf on Computer and Communications Security. New York: ACM, 2009: 621-634

[7] Caballero J, Song D. Automatic protocol reverse-engineering: Message format extraction and field semantics inference [J]. Computer Network, 2013, 57(2): 451-474

[8] Byung-Chul P, Won Y J, Myung-Sup K, et al. Towards automated application signature generation for traffic identification [C] //Proc of the Network Operations and Management Symp. Piscataway, NJ: IEEE, 2008: 160-167

[9] Ye Mingjiang, Xu Ke, Wu Jianping, et al. AutoSig-automatically generating signatures for applications [C] // Proc of the 9th IEEE Int Conf on Computer and Information Technology. Los Alamitos, CA: IEEE Computer Society, 2009: 104-109

[10] Wang Yu, Xiang Yang, Zhou Wanlei, et al. Generating regular expression signatures for network traffic classification in trusted network management [J]. Journal of Network and Computer Applications, 2012, 35(2): 992-1000

[11] Zhao Yong, Yao Qiulin, Zhang Zhibin, et al. TPCAD: A text-oriented multi-protocol inference approach [J]. Journal on Communications, 2009, 30(10): 28-35 (in Chinese)
(赵咏, 姚秋林, 张志斌, 等. TPCAD: 一种文本类多协议特征自动发现方法[J]. 通信学报, 2009, 30(10): 28-35)

[12] Liu Xingbin, Yang Jianhua, Xie Gaogang, et al. Automated mining of packet signatures for traffic identification at application layer with apriori algorithm [J]. Journal on Communications, 2008, 29(12): 51-59 (in Chinese)
(刘兴彬, 杨建华, 谢高岗, 等. 基于Apriori算法的流量识别特征自动提取方法[J]. 通信学报, 2008, 29(12): 51-59)

[13] Wang Bianqin, Yu Shunzheng. Adaptive extraction method of network application signatures [J]. Journal on Communications, 2013, 34(4): 127-137 (in Chinese)
(王变琴, 余顺争. 自适应网络应用特征发现方法[J]. 通信学报, 2013, 34(4): 127-137)

[14] Yuan Zhenlong, Xue Yibo, Dong Yingfei. Harvesting unique characteristics in packet sequences for effective application classification [C] //Proc of the 1st IEEE Conf on Communications and Network Security. Los Alamitos. CA: IEEE Communication Society, 2013: 341-349

[15] Beddoe M. The Protocol information project [EB/OL]. [2004-10-05]. http://www.tphi.net/awalters/PI.html

[16] Cui Weidong, Paxson V, Weaver N, et al. Protocol-independent adaptive replay of application dialog [C] //Proc of the 13th Annual Network and Distributed System Security Symp. Reston, Virginia: Internet Society, 2006

[17] Cui Weidong, Kannan J, Wang H J. Discoverer: Automatic protocol reverse engineering from network traces [C] //Proc of the 16th USENIX Security Symp. Berkeley, CA: USENIX Association, 2007: 199-212

[18] Li Weiming, Zhang Aifang, Liu Jiancai, et al. An automatic network protocol fuzz testing and vulnerability discovering method [J]. Chinese Journal of Computers, 2011, 34(2): 242-255 (in Chinese)
(李伟明, 张爱芳, 刘建财, 等. 网络协议的自动化模糊测试漏洞挖掘方法[J]. 计算机学报, 2011, 34(2): 242-255)

[19] Agrawal R, Srikant R. Mining sequential patterns [C] //Proc of the 11th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 1995: 3-14

[20] Han Jiawei, Pei Jian, Yin Yiwen. Mining frequent patterns without candidate generation [C] //Proc of ACM SIGMOD 2000. New York: ACM, 2000: 1-11

[21] Pei Jian, Han Jiawei, Mortazavi-Asl B, et al. Mining sequential patterns by pattern growth: The PrefixSpan approach [J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(10): 1-17

[22] Zhao Bo, Guo Hong, Liu Qinrang, et al. Protocol independent identification of encrypted traffic based on weighted cumulative sum test [J]. Journal of Software, 2013, 24(6): 1334-1345 (in Chinese)
(赵博, 郭虹, 刘勤让, 等. 基于加权累积和校验的加密流量忙识别算法[J]. 软件学报, 2013, 24(6): 1334-1345)

[23] NETRESEC. MACCDC traces [DB/OL]. [2012-03-16]. http://www.netresec.com/?page=MACCDC

[24] Hack lu. InfoVisContest traces [DB/OL]. [2009-01-15]. http://2009.hack.lu/index.php/InfoVisContest

[25] Pironti A, Pozza D, Sisto R. Spi2Java User Manual-Version 3.1 [R]. Turin, Piedmont, Italy: Polytechnic University of Turin, 2008

[26] Aceto G, Dainotti A, Donato W, et al. PortLoad: Taking the best of two worlds in traffic classification [C] //Proc of IEEE Int Conf on Computer Communications. New York: IEEE Communications Society, 2010: 1-5

Zhu Yuna, born in 1985. PhD candidate. Her main research interests include security protocol identification and reverse engineering.

Han Jihong, born in 1966. Professor, PhD supervisor. Her main research interests include network security and security protocol formal analysis.

Yuan Lin, born in 1981. PhD, associate professor. His main research interests include security protocol formal analysis and software trustworthy analysis.

Chen Hantuo, born in 1990. Master candidate. His main research include protocol security online analysis.

Fan Yudan, born in 1981. Master, lecturer. Her main research include security protocol formal analysis.