软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2022,33(1):254−273 [doi: 10.13328/j.cnki.jos.006303]

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

基于网络轨迹的协议逆向技术研究进展*

王占丰[1], 程 光[2], 马玮骏[3], 张嘉玮[4], 孙中豪[4], 胡 超[5]

[1](东南大学 计算机科学与工程学院, 江苏 南京 211189)
[2](东南大学 网络空间安全学院, 江苏 南京 211189)
[3](南京莱克贝尔信息技术有限公司, 江苏 南京 210007)
[4](国家计算机网络应急技术处理协调中心, 北京 100020)
[5](解放军陆军工程大学 指挥控制工程学院, 江苏 南京 210007)
通信作者: 程光, E-mail: chengguang@seu.edu.cn

摘 要: 协议逆向广泛应用于入侵检测系统、深度包检测、模糊测试、僵尸网络检测等领域. 首先给出了协议逆向工程的形式化定义和基本原理, 然后针对网络运行轨迹的协议逆向方法和工具从协议格式提取和协议状态机推断两个方面对现有的协议逆向方法进行了详细分析, 阐释其基本模块、主要原理和特点, 最后从多个角度对现有算法进行了比较, 对基于网络流量的协议逆向技术的发展趋势进行了展望和分析.

关键词: 协议逆向工程; 多序列比对; 语法推断; 语义推断; 协议状态机
中图法分类号: TP303

中文引用格式: 王占丰, 程光, 马玮骏, 张嘉玮, 孙中豪, 胡超. 基于网络轨迹的协议逆向技术研究进展. 软件学报, 2022, 33(1): 254–273. http://www.jos.org.cn/1000-9825/6303.htm
英文引用格式: Wang ZF, Cheng G, Ma WJ, Zhang JW, Sun ZH, Hu C. Research Progress of Network Protocol Reverse Engineering Technologies Based on Network Trace. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 254–273 (in Chinese). http://www.jos.org.cn/1000-9825/6303.htm

Research Progress of Network Protocol Reverse Engineering Technologies Based on Network Trace

WANG Zhan-Feng[1], CHENG Guang[2], MA Wei-Jun[3], ZHANG Jia-Wei[4], SUN Zhong-Hao[4], HU Chao[5]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)
[2](College of Cyber Science and Engineering, Southeast University, Nanjing 211189, China)
[3](Nanjing Lexbell Information Technology Co. Ltd., Nanjing 210007, China)
[4](National Computer Network Emergency Response Technique Team/Coordination Center of China, Beijing 100020, China)
[5](College of Command Control Engineering, Army Engineering University of PLA, Nanjing 210007, China)

Abstract: Protocol reverse engineering is widely used in intrusion detection system, deep packet inspection, fuzzy testing, C & C malware detection, and other fields. First, the formal definition and basic principle of protocol reverse engineering are given. Then, the existing protocol reverse methods based on network trace are analyzed in detail from two aspects of protocol format extraction and protocol state machine inference. In addition, the basic modules, main principles, and characteristics of these algorithms are explained. Finally, the existing algorithms are compared from several aspects, and the development trend of protocol reverse technology is discussed.

Key words: protocol reverse engineering; multiple sequence alignment; syntax inference; semantic inference; protocol state machine

1 引 言

协议是两个或多个协议实体间完成通信所遵循的规则集合, 其包含 3 个基本要素: 语法 (syntax)、语义

* 基金项目: 国家重点研发计划 (2018YFB1800200, 2017YFB081703); 赛尔网络下一代互联网技术创新基金 (NGII20170406); 东南大学博士后创新人才培养基金 (2242019R20024); 浙江省公益性技术应用研究计划 (LGG20F020014)
收稿时间: 2020-03-02; 修改时间: 2020-05-16; 采用时间: 2020-09-02; jos 在线出版时间: 2021-02-07

(semantics)、时序逻辑 (temporal logic)[1]. 伴随着网络应用的丰富, 网络协议增长迅速, 然而多数的协议没有公开的协议规范或者缺乏完整协议描述, 特别是专用协议, 如工控协议、物联网协议等. 协议逆向工程 (protocol reverse engineering, PRE) 是指在不依赖于协议描述的情况下, 通过对协议实体的网络输入/输出、系统行为和指令执行流程进行监控和分析, 提取协议文法和交互过程. 它广泛地应用于网络入侵检测系统、深度包检测、模糊测试、僵尸网络检测、软件系统集成等领域[2].

协议逆向最早出现在 1950 年, 在 1990 年后伴随着软件和网络技术的快速发展, 由于网络安全管理、模糊测试等方面的巨大需求, 该技术得到了快速发展[3]. 由于二进制报文不可读, 数据信息熵大、协议格式动态可变、上下文背景的相关性以及协议报文加密, 协议逆向难度非常大. 最初协议逆向是通过人工方式实现的, 这是一个十分辛苦和耗费时间的过程. SAMBA 和 WINE 工程分别花费了 12 年、15 年的时间来对 SMB 协议和 API 协议规范进行了逆向分析[4,5]. 因此, 协议逆向技术的大规模应用不仅需要提高协议分析的正确性, 更要实现自动化. 一般协议而言, 协议逆向分为协议报文格式提取、语义推断以及协议行为分析. 其中, 协议报文格式提取和语义推断都属于协议格式分析的范畴, 其包括协议字段划分、关键字识别、分隔符识别等. 在协议报文格式识别过程中, 采用较多的是多序列比对技术、渐进多序列比等改进算法以及基于数据挖掘的方法, 关键字识别、分隔符识别则主要是针对的文本协议进行分析. 语义推断建立在报文格式推断基础上, 往往与其应用领域相关及上下文相关, 因此高级语义的推断往往需要结合协议实体交互过程[2]. 协议行为分析结果一般方通过构建协议的状态机, 其过程主要包括协议会话分割、不同类型报文序列构建和状态机约减等步骤.

协议逆向的方法很多, 按分析对象可以划分为文本协议逆向分析与二进制协议逆向分析, 按协议组成要素可以划分为消息格式逆向分析和协议状态机逆向分析, 按分析方法可以划分为消息序列分析方法和指令序列分析方法等. 在所有的协议逆向方法中, 基于指令序列的分析方法需要得到协议实体, 而基于网络流量的分析方法仅需要得到协议交互实体间的网络流量. 本文仅针对基于网络流量的协议逆向方法进行分析, 由于各类算法或工具的侧重点不同, 本文将所有的算法或工具按照协议逆向的两个阶段进行划分, 然后加以论述. 实际上, 许多面向协议格式的算法需要状态机的信息, 反之状态机的推断也需要协议语法信息, 抑或在推断中迭代使用协议语法信息与状态机信息. 有些算法同时解决了上述两个问题, 如 Biprominer[6]、NetZob[7]、AutoReEngine[8], 在下文中将所有算法的几个部分分别在相应部分进行论述.

本文第 2 节对协议逆向技术框架进行论述; 第 3 节对协议格式分析技术的算法和工具进行分析; 第 4 节论述协议状态机推断方法和工具; 第 5 节对所有算法进行综合对比, 并指出未来研究方向; 第 6 节总结全文.

2 基于网络流量的协议逆向技术框架

协议逆向目标就是获得正确、简洁的协议规范. 协议自动逆向技术的实现可以显著减少人工分析的工作量, 提高对私有协议的分析效率, 并使对网络安全事件的快速自动响应成为可能. 由于协议实体的内部机制不可知, 其协议规范只能通过实体间交互的一系列报文来进行推断, 首先需要会话的划分和报文组装, 然后分析报文格式与推断字段语义, 不同协议状态的格式有可能相同, 在获取协议状态机时需要结合前置格式序列对状态进行合并, 最后在此基础上进行协议状态机的推断. 根据以上分析, 理想的协议逆向模型应当包括预处理、协议格式提取和状态机推断 3 个阶段, 如图 1 所示.

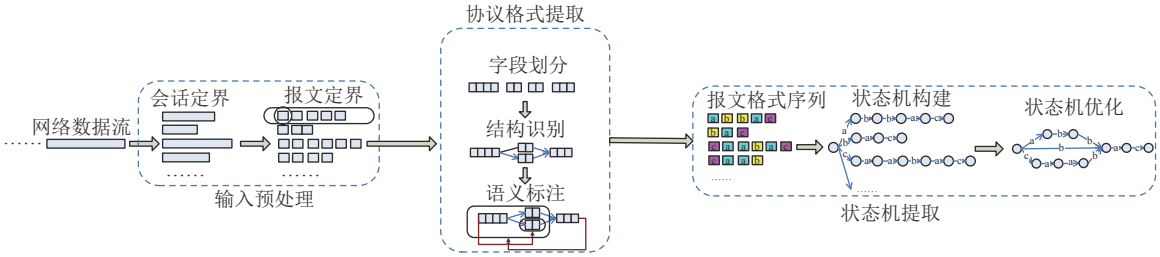


图 1 协议逆向模型

在输入预处理阶段, 要对所有的分组进行过滤, 得到要处理的目标协议报文序列. 一般是通过五元组进行过滤的, 即源目的主机地址、双方端口以及协议, 实际上在协议逆向中不考虑采用上述相同信息的不同协议报文. 之后, 就是将所有的分组构建为不同会话, 一般常用的方法是时间阈值法, 如 ReverX[9] 则将阈值设置为 64 s. 接下来就是要进行分组或报文的定界或组装, 对传输层之上的协议一般不存在这个问题, 而针对无线协议的分析则往往需要设计分组的定界方法[10,11]. 实际上, 这部分虽然是协议逆向实用化中不可或缺的部分, 却不是研究的重点, 经过上述过程后, 原始数据流被划分为由不同报文构成的会话序列集合.

协议格式提取阶段以分组序列为输入, 经过域划分、结构识别和语义属性推断 3 个步骤, 从而生成协议格式推断报告. 实际上, 多数研究集中在前两个阶段, 特别是第一个阶段, 字段划分最初采用的算法是序列对比、多序列对比[12] 等, 然后采用基于统计方法的字段划分方法, 如频率[7] 、信息熵[13,14]、n-gram[15]和概率模型[15] 等. 结构识别是在边界划分的结果上对报文结构进行抽象和整理, 如 Discoverer[12] 采用迭代策略优化字段划分, ProGraph[14] 等采用树形结构推断报文格式, HsMM[15] 等采用聚类的方法推断报文格式, 潘璠等[16] 提出的“正向定位, 反向确认”的策略对结果进行优化. 字段语义推断是所有算法的难点, 对于特定功能字段的识别目前已具有较为成熟的方法, 一般算法能够识别的主要包括时间、IP、MAC、长度、网站地址等[12], 然而对于具体具体的操作语义则需要与协议或软件的用途的支持, 文献 [11,17−19] 对报文字段语义推断进行了较为全面的研究.

在状态机推断阶段, 经过状态标注构建部分状态机, 最后经过消除状态冗余和优化化步骤获取对应的最小确定状态机. 状态机的推断既可以通过推断协议字段, 也可不借助于字段信息. 实际上, 由于无法获得完整的协议报文序列, 状态机的化简是个难点, 一般通过一些启发性规则来完成的, 如 PEXT[20]保证状态的唯一, ReverX[19]通过合并等价状态机来实现, BFS[21]通过报文字段特征等.

目前, 关于协议逆向的研究多达 20 多种算法和工具, 这些工具或算法有的既包含了报文格式和状态机推断, 有的仅是侧重于某个部分, 为了给协议逆向研究人员提供一个较为清晰的研究路线, 将所有的算法从两个角度进行研究, 即协议格式提取技术和状态机推断技术.

3 协议格式提取技术

协议格式提取主要完成协议字段划分以及字段语义推断. 字段划分又进一步可以分为字段提取以及字段关系推断或协议格式分析, 某些算法还包括格式优化过程. 协议字段的语义推断是一个难点, 往往需要背景知识的支撑, 相关研究较少. 在本节, 针对典型算法 PIP、Discoverer、Biprominer、ProDecoder、ProGraph、AutoReEngine、HsMM (hidden semi-Markov modelling) 等 22 种典型的算法进行分析, 为了理清研究思路和脉络, 按照算法的原理将这些算法分为 4 类, 即基于序列比对的格式推断算法、基于概率模型的格式推断算法、基于频繁集的格式提取算法和基于语义的格式推断算法.

3.1 基于序列比对的格式推断方法

3.1.1 PI

PI 是最早成功 APRE 工具之一[22], 它开创性地将生物信息学领域的方法引入到协议逆向中来. 该方法提出者认为生物基因序列与协议分组分析具有相似之处, 即都是在长数据序列中识别模式, 在生物信息学中, 该方算法的目标是确定特定的遗传信息序列, 而在协议分析中, 目标是确定消息中的字段. PIP 算法中所使用的消息对齐方法已经在许多 APRE 工具中重用.

PI 将 TCPDump 文件作为输入, 在分组中区分长度不变的关键字和可变长度的字段, 并以 Byte 作为分析对象. 分析结果说明每个字段的变化率, 0 变异率表示一个常量, 100% 变异率表示高度可变的变量. 该方法使用一种典型的动态规划方法 Needleman-Wunsch[23], 通过增加空白值来实现两个分组的对齐. 如两个序列 “GET/index.html HTTP/1.0”和“GET/home.html HTTP/1.0”, 应该对齐的关键字是 GET 和 HTTP/1.0, 而文件名称则会被识别为变量. 该算法的缺点在于当数据量较大时, 其计算量会增加. 为了简化计算量, 在进行对齐之前首先分组进行聚类, 将相似的分组划分到一起.

在计算不同分组之间的相似度时, 采用了无权分对分组算术平均方法 (unweighted pair group method using arithmetic averages, UPGMAA), 来计算不同类型之间的距离.

$$D=(clusterA,clusterB)=\frac{1}{|C_A||C_B|}\sum_{p\in C_A,q\in C_B}d_{p,q} \tag{1}$$

其中, $d_{p,q}$ 表示 *clusterA* 中 p 与 *clusterB* 中 q 分组的距离.

PI 将序列比对引入到协议逆向分析中来, 许多算法都是基于该算法进行改进的, 对该工具的全面理解, 有利于研究人员对该领域其他算法的快速认识. 其 3 个主要的不足[24]: 首先, 该算法仅对由相似字节序列构成的分组有效, 而许多协议在相同的消息中使用根本不同的字节序列. 其次, 针对特定协议或数据集相似性矩阵的优化十分具有挑战性. 最后, Needleman-Wunsch 算法对可变长度的分组比较存在缺陷.

3.1.2 Discoverer 算法

Discoverer[12] 可以处理网络运行轨迹文件来实现自动挖掘协议的格式, 它主要包括 3 个模块, 标记和初始化 (tokenization and initial clustering), 递归聚类 (recursive clustering) 和字段合并. 图 2 是 Discoverer 算法的主要思想.

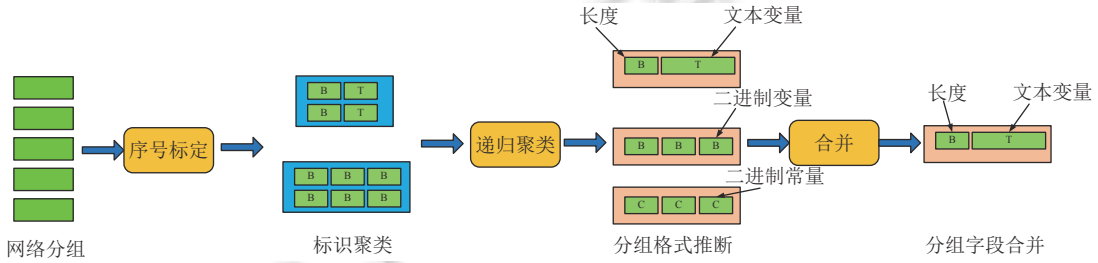


图 2 Discoverer 算法主要思想

由于协议往往无法简单的认为文本或者二进制字节构成, Discoverer 可以同时处理文本和二进制协议, 首先对接收到的协议字节流进行扫描, 通过与 ASCII 码表中所有可打印的字符进行对比, 从而将其标记为二进制或字符串; 为了避免将二进制数据误判为文本字段, 设置了字符串最小长度的阈值, 扫描后的数据构成了文本和二进制字符的序列. 字节序列比对采用的是 Needleman-Wunsch 算法[23]. 作者研究发现字节序列比对适应于具有相似字节模式的消息, 而不适用于具有相同格式的信息. 考虑到消息在不同两个方向格式的不同, 分别对两个方向的消息进行处理, 表示为多元组 (*dir*, *class_of_token*_1, *class_of_token*_2, …, *class_of_token_n*), 其中 *dir* 表示两个方向, 从服务器到客户端或反之, *class_of_token_i* 表示第 i 个格式, 如 (*client to server, text, binary, text*).

在递归聚类阶段, 主要依据的是格式区别标识 (format distinguisher tokens, FD tokens). 为了确定 FD 标识, 需要进行格式推断和格式对比. 格式从输入的消息集合推断出能够表示消息的标识, 实际上就是一组标识的语义及属性. 之所以采用标识语义, 是由于无法推断每个标识的语义而某些属性也可以表示消息格式. 标识的属性包括两种, 文本或二进制与者常量或变量. 作者将标识的类型定义为语义与属性的集合. 由于标识类别已经在标识识别阶段已经确定, 是否为常量与变量通过比较同类消息部分的值是否变化来区分. 采用 RolePlayer[24] 中方法进行语义推断, 包括长度、偏移量和 cookie. 在格式比较阶段, 将两个消息的格式从左到右对标识进行逐个比较, 比较其语义和属性. 在比较过程中, 定义了一些相容性规则, 如常量与变量可以认为相容, 以及向邻近字段合并等. 递归算法如下: (1) 首先统计消息集合中某一标识是否超过了一定的阈值; (2) 对于超过阈值的标识进行子类划分, 并且保证其必须超过一个最小的阈值, 以确保子类数目不会太少; (3) 对于所有满足第 2 个条件的将其与其他的子类进行比较判定其是否可以与其他子类进行合并.

在第 3 个阶段, 对于标识化和递归聚类阶段导致的过度分类问题, 寻找格式上相似的进行合并. 为此采用一种基于类型的序列比对, 在此对比阶段每一个类消息的各个表示只能具有相同的格式, 两者语义相同或具有相同的值. 为了避免在标识过程中引入的误差, 引入了“gap”来实现二进制与文本的对齐, 同时引入了限制规则防止过多

的加入“gap”.

该算法是较早一个比较完备的处理算法, 从原理看是 PI 算法改进, 不仅包括报文格式分析还包括字段语义分析, 为完整的协议格式提取算法和工具提供了参考依据. 然而, 该算法仍然无法避免多序列比对的缺陷, 即比对开销和格式进行分组所引入的误差.

3.1.3 NetZob

Bossert 等[25] 提出了一种基于语义的协议格式推断方法 (Semantic-PI), 它基于现有的 PIP 框架, 在序列比对和消息格式聚类两个阶段增加了语义的考虑. 算法分为会话切分、背景噪声过滤、基于北极的聚类、格式推断、合并等阶段.

在序列对齐时, 将 half-Byte 作为比较的单位, 对每个要素进行语义标注. 论文的难点在于如何挖掘语义信息, 论文提出了在消息中采用不同的编码格式, 如高字节序、低字节序、ASCII 码或 UTF-16, 进行搜索发现主机地址、时间戳或用户名等. 每个要素可以具有多个语义标签, 两个要素比较时不时简单的判定字符串是否相同, 而是语义标签是否完全相同, 或者最为包含与包含于的等关系. 在消息聚类阶段, 不仅考虑 NW 算法就算得到的相似度矩阵, 还考虑了元素在消息中的所占的比例. 在得到消息的格式后, 作者采用最大互信息系数 (maximal information coefficient, MIC) 来分析字段间的关系, 考虑到校验字段的非线性关系, 同时计算了皮尔逊相关系数 r (Pearson correlation coefficient). 文献 [26] 指出, 当 MIC-r^2 接近于零时, 则两个字段为线性相关, 反之则为非线性相关. 论文通过分析 MIC-r^2 系数, 在此基础上进行从而判定字段之间的关系. 状态推断模块则以 Dispatcher[27]中的改进 L*算法为基础, 利用报文格式中的特征信息实现 Mealy 机形式的状态机推断; 报文仿真模块利用输出的协议规范描述与未知协议终端进行通信, 并进一步实现协议实体伪造、智能模糊测试等应用.

作者将这一算法在开源项目 Netzo[25] 中, 进行了实现, 并进行了商业化应用, 它是目前持续更新和较为完善的协议逆向工具. 然而从算法的原理上来看, 它仍然属于第一代算法, 由于算法所采用的语义标注获得困难, 且种类有限, 这一机制需要进一步的完善.

3.1.4 其他基于多序列比对的改进方法

杜有翔等人[28] 提出了一种基于报文序列分析的半自动协议逆向方法结构提取方法. 该方法先将所有的分组进行初始聚类, 将原始样本集划分为格式不相交的子集, 在降低序列集规模的同时能够提高多序列比对的精确度; 对于基本块序列特征相同的样本子集, 按照由左至右的次序逐步在基本块级对报文数据进行对齐, 能够有效缩短比对序列长度, 并且保证比对结果与全局序列比对效果一致. 类似的算法还有张晓明等提出了一种在线协议格式分析算法[29]. 为减少协议分析的计算量, 改善在线协议逆向解析的用户体验, 首先将网络流量划分为不同的子流, 然后对每个子流采用渐进多序列比对方法获得协议的逆向格式, 同时引入误差判定机制, 对不同子流的分析结果进行对比, 当误差结果较大时则子流进行合并重新进行分析, 避免由于划分导致的局部误差, 从而保证全局的正确性. 该类方法的不足之处在于, 以平坦 (相对简单) 的报文字段序列为分析目标, 未考虑对报文结构信息的提取以及字段的语义.

为了解决上述问题, 潘璠等人[28] 提出一种基于递归聚类的自动化报文结构提取方法 (recursive-clustering based message structure extractor, RMSE). 算法采用一种“正向定位, 反向确认”的 FD 字段识别策略. 在字段划分的基础上, 选出候选的 FD 字段并根据字段取值对样本集进行分类, 在完成后续字段序列格式后再对候选 FD 字段进行确认. 采用这种方式可使 FD 字段识别时所有上层 FD 字段取值固定, 确保 FD 字段识别的准确性, 算法主要分为基本块划分、递归聚类和语义及取值约束推断 3 个阶段. 其中, 在语义及取值约束推断阶段, 作者针对分隔符、关键字、长度、计数以及格式标识 5 种语义在语法层次的特征, 分别设计了算法进行针对性识别. 该算法的不足与其他基于序列比对的算法类似, 在语义分析方面仅是部分字段的识别, 没有完整的字段语义谱系.

孟凡治等人[30] 提出了一种基于隐马尔可夫的多序列对齐算法 (multiple sequence alignment based on HMM, MSA-HMM), 算法主要用以分析二进制协议. 主要分为 3 个部分. 首先通过 k 均值聚类算法对所有分组进行划分, 起初将所有分组分为两类, 然后聚类评估算法中的轮廓系数 (silhouette coefficient) 调整 k 值, 并通过 UPGMA 算法进行迭代, 从而得到最终的分类数目. 其次, 对于每一类分组采用改进的导向树渐进式多序列比对 (guide tree

progressive multiple sequence alignment) 方法进行分组划分. 将两个对比的序列计算其 HMM 矩阵的转移概率, 然后构建渐进多序列对比树来对分组进行分割. 最后, 对于分割完毕的分组, 计算邻近字段的变化率、均值、方差等是否接近从而进行字段的合并处理. 在此基础上更加字段的变化率来决定是否为校验字段和序号字段.

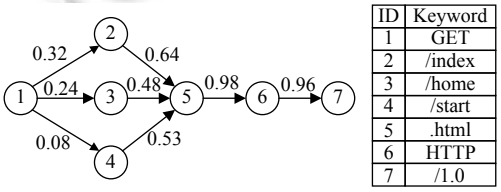
Esoul 等人[31] 为了避免由于报文长短不一而引起的判定误差, 提出了一种基于报文分割的格式提取算法 (segment-based NW). 其基本思想是, 将报文通过基于分割的多序列比对, 然后通过加权系数对分析结果进行合并. 加权系数是报文中包含的字段个数与整个报文中字段数的比值. 算法可以在一定程度上减少长序列比对的时间复杂度, 但准确性未必有明显的提高.

Razo 等人[32] 将多序列比多技术应用于安卓手机无线协议的分析, 提出了 SoundComm-CISEG 系统, 通过分析发现了 5 个固定字段和 2 两个动态字段, 具有较高的准确率. 作者没有对算法进行大幅改进, 也为分析协议的具体特征.

3.2 基于概率模型的格式推断算法

3.2.1 Biprominer 和 ProDecoder 算法

Wang 等人先后提出了 Biprominer[6]和 ProDecoder[13] 两种方法, 通过统计方法来寻找关键字. 与 Discoverer 算法不同的是, Biprominer 是指针对二进制协议而开发的, 它包括 3 个阶段: (1) 通过统计每个 Byte 出现的频率相关模式的相关性来确定字段长度和关键字. (2) 在第 2 个阶段, 将不同的消息与关键字进行关联. 其将应用于信息压缩领域的 Lempel-Ziv-Welch compression algorithm[33] 引入进来, 通过增加 n-gram 算法中 n 的数值来发现关键字. Biprominer 通过关键字的转移概率来发现消息关键字. 对于一个较大的消息集合, 计算阶段 1 中发现所有关键字间的转移概率. 图 3 给出了分析 HTTP 协议的一个实例, 通过关键字间的转移概率从而确定关键字间的顺序.



ID	Keyword
1	GET
2	/index
3	/home
4	/start
5	.html
6	HTTP
7	/1.0

图 3 Biprominer 算法推断示意图

Biprominer 算法仅适用于二进制协议, Wang 等人进行了扩展和改进, 提出了 ProDecoder, 其不仅支持二进制协议还支持文本协议. 算法分为 4 部分, 其中前两个步骤与 Biprominer 算法一致, ProDecoder 不再采用转移概率矩阵, 将关键字和转移概率作为特征, 在此基础上引入信息瓶颈 (information bottleneck, IB) 作为距离测度进行层次聚类. 采用信息瓶颈作为测度的优点在于: (1) 在计算复杂度与精度之间有个较好的平衡; (2) 不用预先计算各个类之间的相似度. 在进行聚类后, 将所有的信息采用 NW (Needleman-Wunsch) 算法进行序列比对从而推断出报文的格式.

该算法将概率统计和数据挖掘的思想引入到协议字段的识别和划分过程中来, 为协议格式字段分析带来了新的思路, 许多后续的算法都是以 Biprominer[6] 和 ProDecoder[13] 算法作为原始模型进行改进的. 此类方法可以挖掘非邻近字段的相关性, 但是对于邻近字段的关联性考虑的较少.

3.2.2 ProGraph 算法

该算法是第一个同时可以针对 bit 和 Byte 两个不同级别进行格式推断的工具[14]. 该算法是一个分组之间不同的字段存在依赖关系. 如在 DNS 协议中 bit 16 位“0”和“1”两个取值分别代表请求或者响应分组, 如果是一个分组是响应分组 (即 bit 16 为 1), bit 24 位“0”和“1”分别表示该 DNS 服务器是否支持递归查询; 否则, bit 24 位取值必须“0”. ProGraph 将目标协议的多个分组通过构建图模型来分析各个分组之间新的相关性, 该算法针是基于每个分组的且假设协议的信息总是包含在分组的最初几个 bit 中, 因此可以用于加密协议的分析.

该算法采用一种自底向上的方法, 将协议分组构建为一棵树, 节点为协议字段, 边表示不同字段的依赖性, 最

初所有的叶子节点和边都为空, 通过学习得出每个字段的合法取值范围, 并计算每两个叶子节点的互信信息熵是否超过一定的阈值从而决定是否为两个节点添加边. 然后对所有的字段进行合并, 合并的依据是: (1) 两个字段之间具有高度的相关性; (2) 两者在分组中的位置相近; (3) 两个字段都与具有相同的邻居节点, 满足前两个条件. 合并后, 对整个树进行更新, 重新计算字段间的相关性, 依次迭代直到停止更新. 在计算过程中, 阈值通过 MDL 算法进行自动设定, 字段之间的相关性通过有向边表示, 一般由取值较少的字段指向取值较多的字段. 图 4 给出了 DNS 协议 bit 18 到 bit 20 之间的合并过程. 由于 bit 18 和 bit 19、bit 20 相关, 且位置邻近因此可以进行合并, 同样的还有 bit 30 与 bit 31.

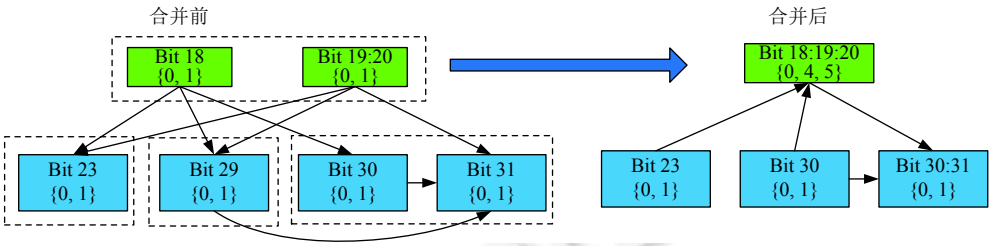


图 4 DNS 协议字段的合并过程

该算法可以处理 bit 和 Byte 级的细粒度的协议, 同时也带来了较大的计算量, 可以在改进算法中考虑协议的规范以及协议的特征, 从而减少搜索空间以提高协议分析的效率.

3.2.3 基于隐马尔可模型的算法

在文献 [15] 中, Cai 等人提出了一种基于隐半马尔可夫模型 HsMM 来实现关键字和协议字段的长度的判定. 与传统的隐马尔可夫模型不同隐半马尔科夫模型在原有的模型中引入了持续时间或者长度. 在此模型中, 每个字段表示一个隐含的状态 f_i, 而感测到的样本则为捕获的分组序列$\{m_1, m_2,\cdots, m_W\}$, 持续时间为每个字段的长度. 计算时要求各个字段的长度之和等于分组的长度为分组的长度, 换言之各个字段之间不存在交叠.

$$\lambda = (A, B, P, \pi) \tag{2}$$

其中, A 表示状态之间的转移概率, B 表示隐含状态的输出概率, P 表示状态持续分布, π 表示初始概率. 图 5 以 HTTP 协议为例给出了协议 HsMM 描述, 观测到的分组 (消息)m, 由一系列的字段构成 $m=\{f_1, f_2, \cdots, f_N\}$.每个字段 f_n 对应为一个状态 i_n, 其持续长度 d_n, a_{in}, i_{n+1} 表示从状态 i_n 到状态 i_{n+1} 的转移概率.

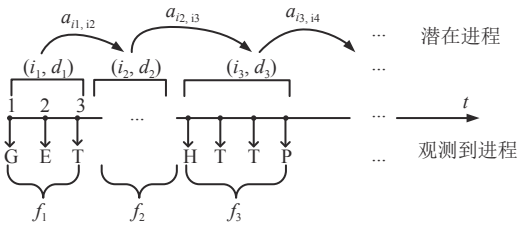


图 5 协议分析的 HsMM 模型

该算法通过 tshark[34]来实现原始分组的获取, 并假设所有捕获的分组都属于要分析的目标协议, 然后通过 5 个步骤会话重建、消息重组、HsMM、消息分割和类型推断. 算法整体分为 5 个阶段. 在会话重建阶段, 通过<传输协议, 源端口, 目的端口, 源 IP 地址, 目标 IP>5 元组来重建会话. 对于 TCP 协议通过识别协议字段 SYN 标志作为会话的开始分组, FIN 分组作为结束分组; 对于基于 UDP 的会话, 则通过 5 元组进行划分. 在消息重组阶段, TCP 分组按照序号和 ACK 序号进行重装, UDP 协议根据分组的时间戳和传输方向进行重组. HsMM 阶段是本算法的重点, 在参数估计中采用的是 Baum-Welch 算法[35], 采用最大似然估计的方法, 对状态转移概率、输出概率以及字段长度进行估计, HsMM 模型不仅可以估计协议的关键字格式同时还可推断协议关键字的长度, 它同时可以

将字段划分为固定位置的字段和位置变化的字段. 最后一个阶段, 论文引入一种无监督的 AP (affinity propagation) 聚类算法[36], 进行消息分类, 将每个分组看做网络中的一个节点, 然后通过计算和更新网络各个节点的吸引度 (responsibility) 和归属度 (availability) 更新节点间的距离, 最后收敛为若干个中心, 即为分类的数目. 图 6 是 HsMM 算法的主要思想.

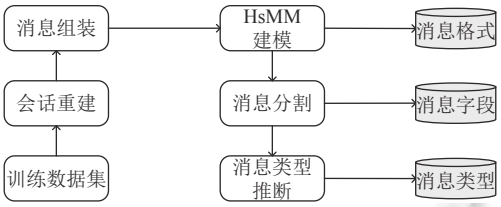


图 6 HsMM 算法主要思想

隐马尔可夫已经广泛应用于文本分词, 其与协议逆向中的格式分析由许多相似之处, 在协议格式推断方面具有一定的创新性, 论文主要关注的是协议格式和消息分类的推断, 在字段语义和状态机方面研究较少.

3.3 基于频繁集的格式提取算法

3.3.1 AutoReEngine 算法

Krueger 等人[37]提出了一种基于语义对网络攻击进行分析的框架 (automatic semantics-aware analysis of network payloads, ASAP), 主要包括 3 个步骤, 首先, 从网络流量中提取相关的字符串表, 然后对这些数据进行匹配, 如果可以匹配则将矩阵中对应的位加 1, 从而得到一个高维的向量矩阵; 第 2 步, 对得到的矩阵进行分解, 采用了 PCA 和 NMF 算法, 得到矩阵的主要成分对应的向量; 第 3 步, 将分解得到矩阵进行表示为析取表达式的方式并进行合并, 从而得到一个能够代表典型通信内容的语义模板. 该算法可以广泛地应用于网络入侵、蜜罐数据分析、恶意软件分析等. 该算法的主要目标不是进行完善的格式发现而是进行恶意软件发现.

AutoReEngine 算法[8] 借鉴了 ASAP 的字母表方法, 在模型中所有的词汇表示为, $D=\{\backslash 0x00,\backslash 0x01,\cdots,\backslash 0xFF\}$ 因此, 报文中的字符串表示为:

$$I=\{i_k|i_k=\overline{d_0d_1\cdots d_l\cdots d_{n_k}}\wedge d_l\in D,l=0,1,2,\cdots,n_k\}.$$

而报文则是由一个个字符域构成的:

$$M=\{m_k|m_k=<f_0,f_1,\cdots,f_l,\cdots,f_{n_k}>\wedge f_l\in D,l=0,1,2,\cdots,L_k\}.$$

一个会话则定义为通信双方所有交换信息的集合:

$$s=<m_0,m_1,m_2,\cdots,m_N>,m_l\in M,l=0,1,2,\cdots,N.$$

两个站点间的通信则定义为若干会话的集合:

$$\zeta=\{s_i|i=0,1,2,\cdots\}.$$

多数的协议逆向工具是从从左到右来识别协议的格式, 然后有些协议是从协议的右侧开始定义的. AutoReEngine 算法的贡献在于两点: 提出了一种基于字符串频率的关键字提取方法; 在确定字符串的长度时分别从消息和行的头部、尾部分别进行定位.

AutoReEngine 算法识别关键字基于这个直觉一个常见的模式的字串也是高频的. 算法分为 4 个部分, 首先将每个 Byte 视为一个关键字, 第 2 步计算每个字符串的频率是否超过一定的阈值, 如果超过就将其加入到频繁集中; 第 3 步增加一个 Byte 重新统计其是否超过一定的阈值, 对存在交叠的关键字进行合并, 计算其频率从而判定其是否为关键字. 最后一步当上述迭代停止, 直到频繁集中字符串都是完全字符串 (closed string) 时, 则为关键字集合.

为了确定关键字的准确位置, 该算法分别计算每个关键字到消息开始、结束的位置, 以及其到行首与行尾的位置, 将其表示为一个向量, 然后分别计算每个元素位置的方差, 从而确定关键字的准确位置. 通过比较关键字其实位置方差的变化来确定关键字, 关键字的结尾往往通过检测回车位置变化来确定. 采用 AutoReEngine 算法与 Discoverer 算法, 对 HTTP, POP3, SMTP, FTP, DNS 和 NetBIOS 等协议进行了对比, 由于其能够识别关键字的起始

位置, 因此具有更高的准确性. 该算法同时采用了 Apriori 算法来进行状态机的推断和构建.

3.3.2 基于 Apriori 的格式提取算法

Wang 等人[10]提出了一种无线网络的格式推断算法 (AC-FP 算法), 其主要思想如图 7. 算法分为两个主要的步骤通过改进的 AC 算法来确定帧边界, 通过 Apriori 算法[38]实现字符特征提取, FP-Growth 算法实现字段关联分析. 该算法分为如下几个阶段.

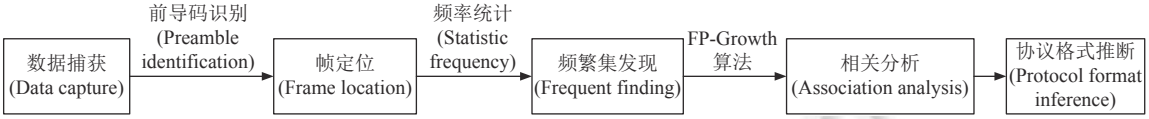


图 7 AC-FP 算法主要思想

首先, 数据捕获阶段获取无线信号过滤掉噪音将其转换为比特流. 然后, 采用信号的前导码进行帧边界的确定, 由于 802.11x 采用的是伪随机序列, 因此采用多模式进行匹配, 采用最长的重用序列, forcanset = {$cset_1$, $cset_2$,⋯,$cset_n$}, $cset_i$ 表示一个可能前导码序列. 第 3 步, 计算所有帧序列的频率, 首先对帧进行聚类, 同一个协议的帧也可能具有不同的格式, 如 ARP 的广播包与任播包是不同的. 其次采用改进的 AC 算法来识别 4 bit (half-Byte) 的序列. 第 4 步, 挖掘常见串之间的关联关系. 虽然无法猜测字符串的含义, 也无法区别在不同串中的含义, 但是可以通过统计不同字符串之间的出现概率来进行协议格式的关系, FP-Growth 算法基于字符串的概率以及不同字符串间的条件概率来推断字段间的关系. 第 5 步, 基于上述关系输出协议的格式字符串.

基于类似的想法, Hei 等人[39]提了一种基于 AC 和 Apriori 的算法, 可以发现任意长度 bit 位的关键字. 该算法分为 3 个步骤, 通过变化关键字的长度, 采用 Zipf 分布来对关键字进行统计, 仅排序在前面的作为关键字. 然后采用 AC 算法对所有分组中出现的关键字的频率进行统计, 然后通过 Apriori 算法进行频繁集的挖掘, 从而得到可能的报文格式, 然后根据报文的支持度将其进行约减. 该算法可以支持 bit 流, 适用与文本与二进制文件.

3.3.3 SPREA

Fan 等人[40]认为以往的格式推断算法忽略了字段前后顺序的关联关系, 因此提出了一种基于 FP-tree[41]的改进算法 SPREA (算法框架如图 8). 此外, 该算法利用信息熵对分组中的加密字段进行了推断, 研究了如何判定加密字段的边界. 此算法大致分为 4 个部分, 首先是分组的预处理, 然后是关键字顺序提取金额加密字段判定, 最后基于 Prospex 算法[42]进行了状态机的推断, 下面对其核心部分进行说明.

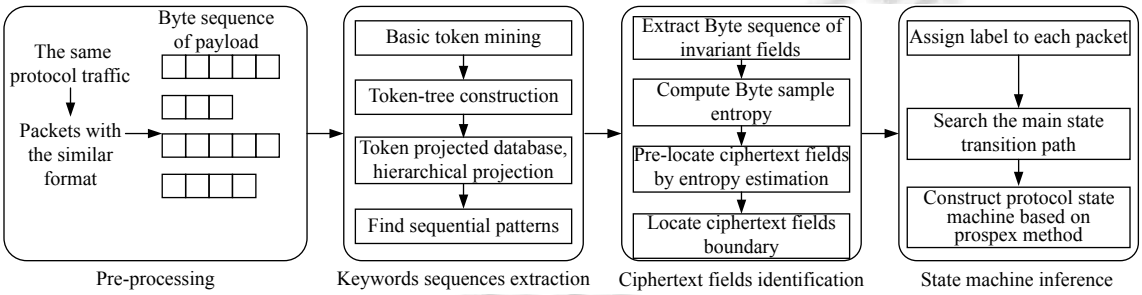


图 8 SPREA 算法主要架构

在 SPREA 中所有的关键字分为两类即位置固定的关键字, 和位置不固定的关键. 在初始阶段, 对个每个 Byte 的数据进行统计, 计算其支持度, 当其支持度大于 min_sup=0.6 时, 则认为其可以独立的成为一个关键字. 当两个邻近的字符串具有重叠部分时, 进行合并, 字符的最小长度 min_len 的阈值设置为 2. 对于识别为关键字的字节, 通过一个 5 元组进行标识 (标识名, 频率, 偏移量, 重叠数, 父节点), 构建树时采用了 FP-tree, 不同的是不按照支持度对所有的元素进行重排列, 从而保持元素前后之间的关系不变. 在协议推断时, 采用了改进的 PrefixSpan 算法[43],

由于考虑了字节的顺序且没有按照支持度进行排序, 挖掘频繁集时, 采用自左到右, 自上到下的构建方法, 要求频繁集的增加的项必须大于增加前最小的支持度. 从而完成频繁集的构建.

在加密字段识别时, 首先计算所有字节的信息熵, 然后通过最大似然估计就算分组中 N 个字节在加密情况下的熵值作为阈值, 然后根据字节的信息熵与阈值进行比较, 判定字节是否为加密字段, 然后对连续字段进行合并从而得到加密字段. 在对加密字段边界判定时, 将加密字段分为两类, 一类是固定长度的, 一类长度不固定的. 根据位置又可以分成位置固定和不固定的. 因此, 作者首先判定加密字段的长度, 对于可变长度的, 往往存在一个字段标记其加密字段的长度, 且加密长度标识字段一般不会超过两个字节, 且位置位于加密字段前, 因此从非加密字段中寻找这个字段.

3.3.4 HIERARCHICAL CSP 算法

Goo 等人[19]提出了一种基于 Apriori 的算法, 可以识别字段格式、消息格式和流格式, 即协议的格式、字段语义和状态机. 算法分为 5 个部分, 首先分组的预处理、消息组装、语法推断、语义推断和状态机的推断. 在预处理和消息组装阶段, 考虑了 TCP、UDP 协议的区别, 传输方向、重传分组、乱序分组以及 ACK 等握手信息的消除. 语法推断和语义推断是该算法的核心和创新之处.

在语法推断阶段, 作者将所有的字段分为 4 类: SF(v)、DF(v)、DF 和 GAP. SF(v) 表示静态有固定长度的字段; DF(v) 表示动态具有固定长度的字段. DF 和 GAP 表示随机性很强, 无法预测其取值范围的字段, 两者的区别在于 DF 字段的长度是变化在一定的范围内, 而 GAP 字段的长度则无法预测. 因此, 在分析过程中, SF(v)、DF(v)、DF 等 3 类字段的最小、最大和平均长度会进行存储. 在进行语法分析时, 采用了一种自底向上的分析方法, 算法分为 4 个步骤, 首先是 SF(v) 类字段的提取, 然后是 DF(v) 类字段的提取, 然后消息格式的和额外格式的提取. 在提取 SF(v) 字段时采用 Apriori 算法, 输入的字段的长度从一个 bit 到一个 Byte. 此阶段提取的结果, 将作为 DF(v) 字段的输入, 算法同样采取 Apriori 算法计算字符间的频繁集, 其包含 3 个特征位置变化不大, 支持度不为 1 (表示值是变化的), 字段的长度变化不大. 在此基础上, 再次使用 Apiori 算法, 从而得到根据字段间长度的方差, 以及最大最小值之间的差异来判定其为 DF 或 GAP 类型, 从而将分组中 4 中类型的字段都识别出来.

在字段的语义推断过程, 算法采用了 FieldHunter[18]中的方法, 详见第 3.5.1 节.

3.3.5 IPART 算法

IPART 算法[44] 是针对工控协议逆向分析而设计的算法, 作者分析了工控协议与普通协议的区别, 认为: (1) 工控协议更加扁平化, 一般工控协议都由一个固定的协议头和数据两大部分构造, 同时协议报文的字段位置往往是固定的. (2) 协议字段简短, 与传统协议相比协议字段更加简短, 如 IEC104 协议很多功能码就有个字节. (3) 协议字段间没有定界符. 该算法分为 3 个部分: 首先将分组分解为不同的字段顺序, 然后识别分组类型, 然后进行类型的推断和状态机的构建.

在字段推断部分, 首先通过 n-gram 算法选择出高频的字段, 然后采用全局投票算法 (global voting expert algorithm, GVE),判定将相邻的字段进行合并. 在字段识别时, 设计了 4 种不同的字段识别算法, 包括常量、长度字段、序列号和功能码. 论文在功能码识别时考虑了字段取值、偏移量等因素, 从而计算每个字段的得分, 将得分高的定为功能码. 最后, 算法根据数据传输的方向和功能码进行分类, 将同一种类型的分组格式进行合并, 从而推断协议的分组格式.

3.4 基于语义的格式提取算法

3.4.1 FieldHunter 算法

FieldHunter[18]可以处理文本协议和二进制协议, 主要包括两个模块字段提取器和字段类型推理工具, 如图 9 所示, 其中 T 表示文本类型, B 表示二进制. 字段提取器又分为两个模块键值和单字提取器. n-gram 提取器分布处理文本协议和二进制协议. 对于二进制的字符关键字的提取则通过变化 n 的取值来发现高频的字符串作为备选的协议关键字. 文本协议分析的关键在于确定分割符. 分隔符又分为字段分割符和键值对分隔符. 如报文”TIME-OUT: 60 # PORT: 54001” 中, “#”为字段分隔符, 而“: ”为关键字和值的分隔符.

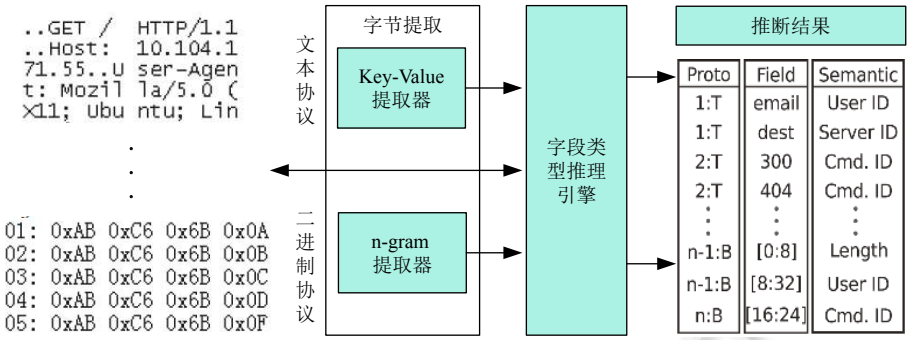


图 9 FieldHunter 系统架构

字段分隔符的确定: 一般字段分割符的特征包括 3 点: (1) 一般为的不是字符 1 或者 2 两个符号, 如“#”“&”等; (2) 分隔符在文本协议中相对于其他非字符符号出现的频率高; (3) 一个协议一般只有一个字段分隔符. 基于上述发现, 通过统计非字母的符号频率获得出现频率最高的符号即为该协议的字段分隔符.

键值对分隔符的确定. 算法分为 3 个步骤: (1) 通过最长前缀法提取对分组进行分类; (2) 对分类结果重新进行聚类, 避免具有少数共同关键字引起的误差, 如 Port: 与 Point: 具有两个“Po”相同的字符; (3) 在关键字中将共同非字符符号后缀提出来作为分隔符.

在确定了字段后, 字段类型推理工具负责确定每个字段的语义, 文献 [18] 中定义了 6 种类型的推断方法, 包括类型字段, 长度字段, Host 标识, 会话 ID, 状态 ID 和计数器 (accumulators). 类型字段的特征是变化范围不太大, 其与反方向数据量的变化没有什么关系. 长度字段计算该字段与数据长度之间的 Pearson 相关性系数, 如果该值达到一定阈值, 这说明其为长度字段. 主机 ID 计算其与主机源地址之间的相关性来进行判定. 会话 ID 通过计算该字段与会话五元组之间的关系来进行判定. 状态 ID 通过计算该字段与会话中该字段熵值足够高, 而且反向数据中的状态 ID 与之相同. 计数器通过判定该字段是否具有恒定的增长速率.

3.4.2 WASp 算法

Choi 等人针对基于 IEEE 802.15.4 的无线协议提出了 WASp (WPAN automatic spoofer)[11]. 该工具针对 Byte 级别进行识别, 并可以生成分组来进行分析和攻击. 802.15.4 是低速率无线个域网 (LR-WPAN) 协议, 规定了 LR-WPAN 的物理层和媒体访问控制 的标准. 该算法无法识别加密协议, 这是由于低功耗协议中加密计算需要消耗大量的能源. 图 10 是 WASp 算法的主要思想.

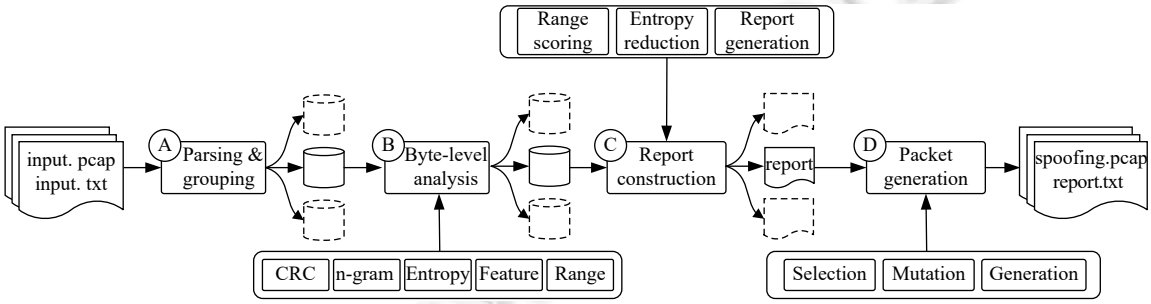


图 10 WASp 算法主要思想

图 10 给出了 WASp 算法的主要 4 个步骤: 解析分组阶段、Byte 级分析阶段、报告构建阶段以及分组发生阶段. 其中, 前 3 个阶段是针对协议分析的, 最后一个阶段是为了模糊测试或者攻击使用. 在解析分组阶段将 pcap 报文和描述文件, 首先对包含错误 MAC 的分组进行过滤, 将依据报文的长度进行分类, 然后再根据源目的 MAC 地址进行子类的划分. Byte 级分析阶段是该算法的重点, 在此阶段, 分别采用 CRC 检验、n-gram 检验、熵检验、特

征检验、范围检验对分组进行处理. 这些算法的检验是按照一定顺序进行的, 如范围检验就是针对 CRC、n-gram 以及 Zero 熵检验之后确定的字段之外字段进行检验. 报告生成阶段分为 3 个步骤, 分别是范围评分、熵约减和报告生成. 范围评分、熵约减是为了简化报文格式的生成过程, 而进行的数据处理. 在报告生成阶段, 依据采用的 5 种算法的置信率, 依次将 CRC 字段、n-gram 以及特征检验确定的字段, 最后是 Zero 熵检验确定的字段, 而对于范围检验确定的字段仅给出了参考评分, 并不确定其字段范围.

4 协议状态机推断技术

上一节讲述了如何推断协议的语法, 而本节则关注于协议状态机推断的相关算法和工具分析. 实际上, 前者是对一个消息内部的数据进行聚类, 而后者则是对不同消息本身进行聚类. 协议状态机的重构主要基于通信数据帧中状态相关字段的取值以及它们之间的顺序关系. 数据帧中不同的状态相关字段反映了协议实体所在的不同状态,数据帧间的顺序关系反映了状态之间的转换关系. 因此, FSM 工具的处理对象往往是两个通信实体间包含不同消息队列的会话集合. 一般此类工具包含 4 个步骤, 首先从网络通信数据中提取两个主机的通信数据包, 然后将这些信息进行聚类获得一个有限的类型集合称为状态, 第 3 步将观测到的不同聚类间的状态转移定义为它们之间的边, 第 4 步对边和状态进行简化. 图 11 给出了一个通用的状态机推断模型, 不同模型创新之处主要在于步骤 2 和步骤 4.

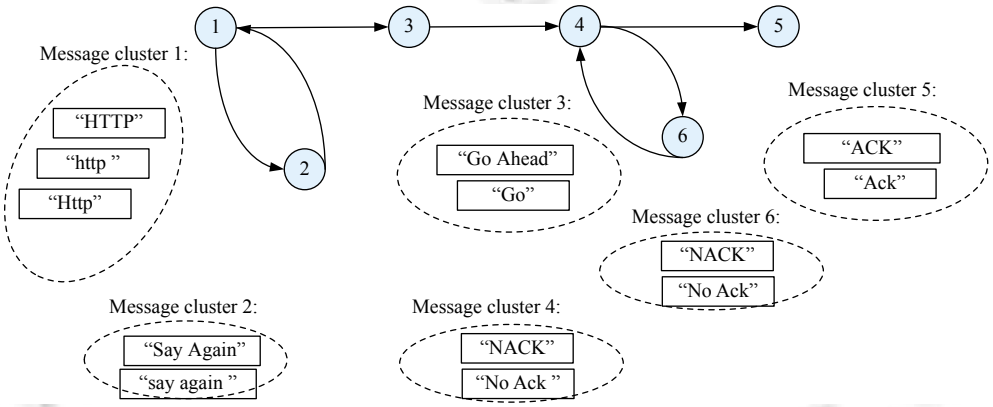


图 11 协议状态机推断的一般示例

本节将所有的状态机推断算法分为两类: 被动式协议状态机推断算法和主动状态机推断算法, 然后逐一进行分析.

4.1 被动式协议状态机推断方法

4.1.1 PEXT 算法

PETX (protocol extraction)[20]是一种较早的协议状态机推断方法, 该工具通过 LibPcap 工具来进行二进制分组的捕获, 然后将所有的分组采用 LCS (longest common string) 方法进行聚类, 可以基于不同的粒度如比特或者字节等, 然后将所有的数据流采用分组的聚类表示进行替换, 从而得到一个数据流的新表示方式, 在此基础上再次采用 LCS 方法进行聚类得到状态集合, 然后将各个数据流表示为有偏状态机, 并将这些部分状态机进行合并, 最后将合并到的状态机进行优化, 将状态机进行分解和合并, 从而得到最优的状态机. 其原则是必须保证从一个状态到一个状态的转换是唯一的, 对于违反上述原则的则从上述状态父节点中增加一个虚拟节点, 从而保证唯一. 此外, 还采用"pull-in"的策略将相邻的节点进行合并, 这些节点只有一个前驱和后继节点. 该算法缺陷在于无法对状态机的语义进行标注, 同时基于 LCS 的聚类算法的阈值需要进行选择.

4.1.2 ReverX 算法

ReverX 是 Joao 等人[9]提出的一种针对网络流量的自动推理工具, 可以自动推断协议规范和状态机推断分为

3 个步骤: (1) 提取会话的分组, 其主要的规律规则如下: 提取与主机具有相同目的和端口的分组, 对于 TCP 分组序号必须是单调递增的, 分组之间时间间隔不大于 1 个小时; (2) 部分状态机的构建, 按照分组的时序结合算法前期推断出来的分组类型构建初始的有偏状态机, 并对这些状态机进行合并; (3) 状态机约减, 包括两个步骤: 一是根据分组的类型进行状态合并, 将从同一类分组出发到达的状态进行合并; 二是对等价的状态进行合并, 如果两个状态互为因果则认为可以合并, 如果不存在直接的因果或者具有相同类型的转移分组, 则不可能等价. 基于上述两个规则, 对状态机进行反复迭代从而得到接近于真实情况的 PSM.

4.1.3 BFS 算法

Trifilo 在 PEXT 的基础上, 提出了基于二进制特征选择 (binary features selector, BFS) 状态机推断方法[21]. 该方法主要依据是会话过程中协议关键字的取值具有相似性, 而状态机转换模式是固定的, 其包括 3 个模块, 首先通过主机和端口对分组进行过滤将其存储在标准的 tcpdump 文件中. 在第二阶段, 基于分组利用统计方法提取各个字节的取值特性, 然后标记特征字段. 其通过统计分布方差变化 (variance of the distribution of variances, VDV) 来对协议的关键字进行判定, 方法不识别分组格式而是关键的标记字符. 最后一个阶段, 状态机的构建也是系统关键, 分为 3 个步骤: (1) 状态机初始化, 为每个协议状态机构建“START”和“END”状态, 为不同的状态分配编码, 根据分组的发送次序标记不同状态间的方向. (2) 状态分裂, 此阶段保证可能接受的正确状态数目最小, 首先按照分组顺序构建状态机, 然后构建滑动时间窗口生成子序列, 对子序列进行两两比较, 对于没有出现的状态转移进行删除, 或将其分裂为多个状态. (3) 状态剪枝, 为了减少状态分裂过程中产生的非必要状态分支得到最简状态机, 根据如下过程进行剪枝: 首先依据状态机对 trace 文件进行解析, 其次仅保留在 trace 文件中出现过的状态.

4.1.4 基于序列比对的状态机推断方法

Antunes 等人[45]基于偏序比对方法提出了两种状态机推断方法, 贪婪算法 (POA-Greedy) 和保守的对齐法. 传统的序列对比方法会丢失那些非高频的信息, 为了充分使用以前比对的效果, 在多序列比对中将以前的对比效果采用一个偏序图表示, 后到达的分组数据与偏序图进行对比, 那些在先前序列中不存在的信息会创建一个新的分支, 从而保证信息不丢失. 在贪婪的对比方法中, 尽快能减少节点构建, 从而导致节点间的连接很多, 在保守的构建方法中, 仅根据接收的报文的前缀不同时才创建新的分支, 因此必然导致状态机具有较多的叶子节点. 论文的不足之处在于没有给出如何将两种方法进行融合和状态机的化简方法.

4.1.5 Veritas 算法

Veritas[46]是一个基于统计的状态机自动推理算法, 该算法可以处理文本和二进制协议. 该工具分为 4 个模块 (如图 12): 网络流量采集、分组分析、协议状态信息推断和状态机推断. 网络流量采集模块根据 IP 和端口进行过滤.

在包分析阶段通过 K-S (Kolmogorov-Smirnov) 检测来发现高频的字节单元, 然后通过报文组装来实现报文格式推断, 其主要包括 3 个步骤: (1) 从流量中随机选择一组报文; (2) 将高频的信息字符串, 进行组合在这些报文中进行匹配; (3) 所有匹配的字符串作为协议格式消息.

$$J(a,b)=\frac{|a\cap b|}{|a\cup b|} \tag{3}$$

在协议状态推断阶段, 为了定义分组之间的相似性引入了相似性测度 Jaccard index, 在分类过程中引入了 PAM (partitioning around medoids) 算法, 与 k-means 算法相比对噪声和异常数据更加鲁棒, 在类别参数的选择中采用了邓恩指数 (Dunn index), 选择使得邓恩指数最大的 k 值. 由此, 每个类表示一个状态, 从而构建状态机图, 在数据处理过程中, 设定了数据流超时时间为 64 s. 在此概率模型中, 将不同状态转移的分组的频率作为边的权值, 对于那些概率小于 0.005 的则不考虑.

除此之外, 孟凡治等人[47]也研究了状态机构建以提取采用方法与 PEXT 类似, 但是并没有提出具体的状态机化简方法.

4.2 交互式状态机推断算法

4.2.1 自动回复工具

关于状态机推断的最早起源于基于到达的消息进行自动的信息回复, 典型算法如 ScriptGen[48]和 RolePlayer[24].

这些工具一般包括 5 个步骤: (1) 获取和存储所有两个通信实体的所有分组; (2) 对消息进行聚类; (3) 确定不同聚类之间的边; (4) 基于 FSM 的位置建立一个基于到达分组响应库; (5) 从库中选一个已经观测到的分组对于到来的分组进行响应. 此类模型的关键在于如何简化 FSM. ScriptGen 工具采用 Needleman-Wunsch 算法来对消息进行聚类, 然后基于 Byte 级的统计信息, 如字节概率、高频取值、方差等要素进行分类, 实际上这类算法的初衷不是为了进行协议逆向, 而是为了傀儡机的控制信息.

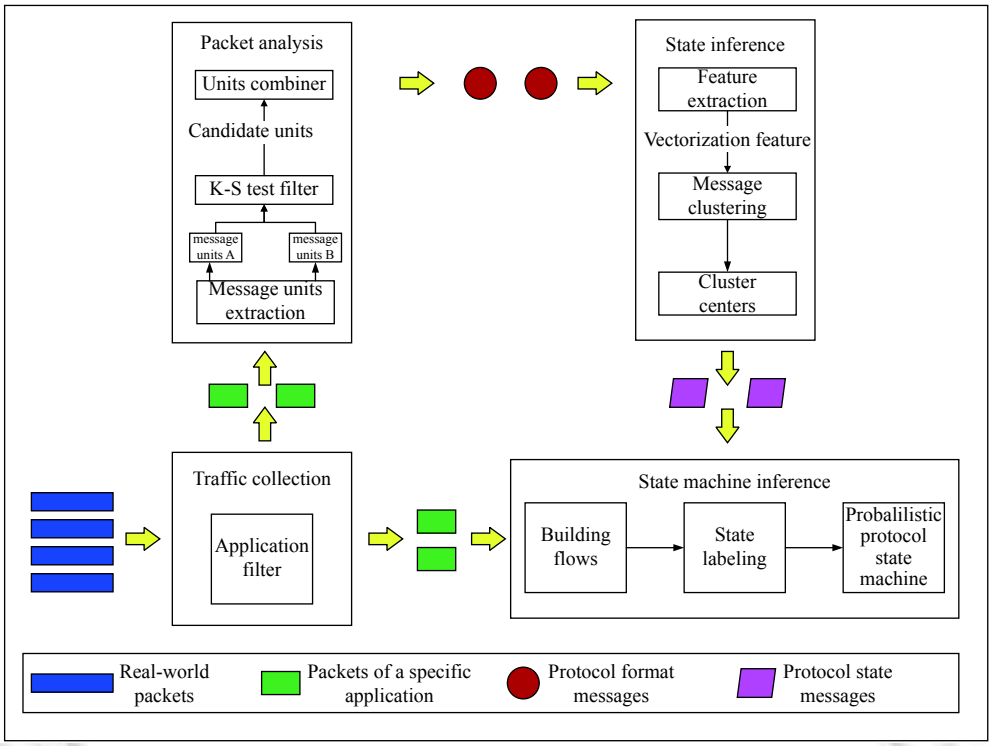


图 12 Veritas 系统主要模块

4.2.2 NimsGP 算法

上述算法都是基于采集的网络流量进行协议状态机的推断, Laroche 等人[49]提出了一种基于线性页面基因规划方法, 通过与协议实体进行交互获得更为全面的网络流量从而得到较为全面的状态机. 该方法基于已知的协议命令集合, 将与协议实体交互获得相应作为适应性的评价指标. 在初始阶段命令的交互顺序通过随机选择产生, 在每次迭代中随机产生一个命令集合来进行对比. 通过对比, 将选择更具有适应性的进行替换, 在变化中使用的选择算子包括交叉、交互和变异. 通过此种方法, 作者对 FTP 和 DHCP 协议进行了分析, 可以发现更加完善的协议状态机.

4.2.3 MQSM 算法

Zhang 等人[50]提出了一种改进的 QSM 方法[51], 该方法包括 3 个分组分析阶段、协议状态探索以及改进的 QSM 学习模块. 其主要工作过程如下: 首先采用 Wireshark 来获得分组数据, 并通过端口等要素进行过滤, 第 2 步分组分析阶段通过分析将数据结构采用 XML 格式进行表示, 第 3 步协议状态机空间探索者生成不同的探测顺序与协议实体进行交互, 最后, 将论文中提出的 QSM 中与算法 EDSM[51]进行结合, 从而提高协议状态机的推断准确性. MQSM 算法从以下几个方面进行了改进: 数据的交互对象由终端用户改为了协议实体; 在 blue-fring 阶段采用了 EDSM 算法; 支持人工编辑分组发送顺序; 基于先期生成的 XML 格式文件自动构建和解析分组从而简化实现难度.

5 算法对比与分析

5.1 协议格式提取算法对比

本文第 3 节系统梳理了协议逆向工程中关于格式分析的主要算法, 按照算法基本思想进行了分类, 并按照其演化过程对算法的主要思想进行了详述. 本节将对主要的算法从过个维度进行分析, 然后从不同的角度对现有的研究成果进行评述. 表 1 给出了各算法提出时间、分析粒度、协议类型、字段提取方法、协议格式提取、是否进行了语义和状态机的推断.

表 1 格式提取算法对比

序号	算法名称	时间	分析粒度	协议类型	字段提取	格式推断	语义推断	状态机构造	测试协议
1	PI[22]	2004	Byte	Text	多序列比对	UPGMA	No	No	HTTP, ICMP
2	Biprominer[6]	2011	Byte	Binary	统计	转移概率模型	No	Yes	Xunlei, QQLive, SopCast
3	ProDecoder[13]	2012	Byte	Binary+Text	n-grams	Needleman-Wunsch	No	No	SMB, SMTP
4	RMSE[28]	2012	Bit	Binary+Text	序列比对	递归聚类	Yes	No	FTP, DNS, Emule, SIP, SMB
5	AutoReEngine[8]	2013	Byte	Binary+Text	字母统计	Apriori	No	Yes	HTTP, POP3, SMTP, FTP, DNS, NetBIOS
6	Discoverer[12]	2013	Byte	Binary+Text	多序列比对	递归聚类	Yes	No	HTTP, RPC, SMB
7	AC-FP[10]	2013	Hex	Binary	AC	FP-tree	No	No	ARP, ICMP
8	Semantic-PI[25]	2014	Hex	Binary+Text	多序列比对	UPGMA	No	No	FTP, SMB, ZA, VoIP
9	ProGraph[14]	2015	Bit, byte	Binary+Text	统计频率	图论	No	No	HTTP, DNS, BitTorrent
10	FieldHunter[18]	2016	Byte	Binary+Text	统计频率, n-gram	启发式规则	Yes	No	STUN, FTP, HTTP, POP3, SMTP, MSNP, RTSP
11	HsMM[15]	2016	Byte	Binary+Text	字母统计	HMM	No	No	HTTP, SSDP, BitTorrent, QQ, DNS, NetBIOS
12	SoundComm-CISEG[32]	2016	Byte	Binary	序列比对	UPGMAA	No	No	Blooteeth
13	WASp[11]	2016	Byte	Binary	n-gram, Zero熵检验等	启发式规则	Yes	No	IEEE 802.15.4
14	Segment-Based NW[31]	2017	Byte	Binary+Text	序列比对	UPGMA	No	No	HTTP, SIP, TFTP, SMB
15	IPFRA[29]	2018	Bit	Binary+Text	序列比对	UPGMAA	No	No	HTTP, FTP
16	MSA-HMM[30]	2018	Byte	Binary	序列比对	HMM	No	No	EthernetIP-ICMP
17	SPREA[40]	2018	Byte	Binary	FP-tree	PrefixSpam	Yes	Yes	SSL, SSH, NS, SOF
18	HIERARCHICAL CSP[19]	2019	Byte	Binary+Text	Apriori	Apriori	Yes	Yes	HTTP, DNS
19	CFI[39]	2019	Bit	Binary+Text	AC	Apriori	No	No	—
20	IPART[44]	2019	Bit	Binary	n-gram	GVE	Yes	Yes	Modbus, IEC104, Ethernet/IP

从分析粒度来看, 多数算法分析粒度在 Byte 级别, 而少数算法在 bit 级别, 这主要是多数的研究是针对互联网协议设计的, 互联网协议的设计无论是应用层还是传输层都是以 Byte 作为单位的, 但是在物联网、工控协议中许多协议是 bit 作为基本单元的. 因此, 目前多数的算法在对于细粒度协议格式分析方面存在不足. 协议类型分析类型来看, 以文本协议、二进制和兼容二进制协议算法数目相差不大.

字段提取与格式推断是算法设计重点也是两个关键的阶段. 字段提取主要采要分为 4 类, 即多序列比对、频率统、n-grams 和频繁集算法, 其中后 3 者从原理上都是通过统计备选的字段出现的频率来进行可能字段的挖掘, 与最初的多序列比多在原理上有着较大的区别. 在格式推断方面, 主要分为 3 类型, 层次聚类方法、频繁集挖掘、以及概率模型, 第 1 类方法本质上强调了个字段在编辑距离上的关系, 而后两类则是关心的统计意义上的相关性.

协议字段的语义推断是个难点, 在此方面 FieldHunter、WASp[11]、HIERARCHICAL CSP[19]这 3 个算法进行系统的研究, 其余算法则是对于一些典型字段如长度、CRC、IP、MAC 等典型字段进行了初步研究. FieldHunter 和 WASp 算法对语义字段进行了较为系统详细的分类, 并给出了每类字段的判定方法, 以及当字段语义冲突时的优先判定规则, 而 HIERARCHICAL CSP 则对所有字段进行概念上的系统划分, 并结合 FieldHunter 中算法给出了较为全面的推断过程. 实际上, 目前在于语义推断方面, 都是关于互联网协议典型字段的语义的分析, 对于非公开协议最多的物联网、工控等方面的应用尚处于起步阶段[52].

第 3 节所关注的算法主要针对协议格式提取进行研究, 即便某些算法和系统具备状态机的推断能力, 也都是现有算法的应用. 从分析协议对象看, 主要是传统的应用层协议, 很少出现无线协议和物联网或工控协议.

5.2 状态机提取算法对比

状态机描述了协议实体间的动态交互行为, 既依赖于格式推断的效果, 同时又具有相对的独立性, 对协议格式的推断具验证作用. 协议状态机可以用于软件的交互测试, 从而发现漏洞或者更多的协议状态. 与协议格式提取算法相比, 协议状态机的研究较少. 表 2 从几个维度对主要的协议状态机进行了对比, 可以得到以下结论.

(1) 近年来在协议状态机推断方面的研究较少, 多数协议状态机推断算法同时支持二进制协议和文本协议, 且以被动方式为主, ScriptGen 和 RolePlayer 两个工具并不算严格的协议逆向工具, 然而它们的研究思路是状态机推断算法的起点, 因此也进行了论述.

(2) 在协议状态机提取方面, 目前采用的方法主要分为两类: 一类是按照消息的自然时序来提取协议部分状态机, 一类是通过聚类分析从而减少状态的数目避免状态膨胀. 只有一些综合性的算法如 NetZob 综合考虑了分组类型的划分.

(3) 在协议状态机构造方面, 多数的状态机化简都是基于启发式规则来进行的, 采用的自动化简方法较少, 或是与分组类型的分类相结合的.

表 2 协议状态机推断算法对比

序号	算法名称	时间	协议类型	交互方式	状态提取	状态机构造	分析对象
1	ScriptGen[28]	2005	Binary+text	主动	序列比对	时序构建	HTTP, NetBIOS, SMB
2	RolePlayer[24]	2006	Binary+text	主动	时序构建	—	SMTP, DNS, HTTP等
3	PEXT[20]	2007	Binary+text	被动	LCS聚类	启发式约减	FTP
4	POA-Greedy[45]	2009	Binary	被动	序列比对	时序构建	FTP
5	BFS[21]	2009	Binary	被动	VDV	时间窗口化简	ARP, DHCP, TCP
6	Veritas[40]	2011	Binary+text	被动	频率统计	PAM	SMTP,PPlive, Xunlei
7	ReverX[9]	2011	Binary+text	被动	分组划分	启发式约减	FTP, SMTP, POP
8	NimsGP[49]	2013	Binary	被动	线性页面基因规划方法	时序构建	FTP, DHCP
9	MQSM[50]	2012	Binary+Text	主动	EDSM	时序构建	HTTP, ISAKMP, SNMP
10	NetZob[25]	2014	Binary+ Text	主动	序列比对	时序构建	FTP, SMB, Zaccess等

5.3 未来研究方向

通过对于协议逆向主要算法分析发现, 总体而言针对协议格式分析相对较多, 状态机的较少, 但是总体而言协议逆向仍属于初级阶段, 但是由于其在网络入侵检测系统、深度包检测、模糊测试、僵尸网络检测等方面具有广泛的应用前景, 未来依然是个研究的热点. 建议从以下几个方面进行开展研究:

(1) 研究工控等物联网专有协议的特征, 设计更加有效的识别方法. 从格式提取算法演化过程看, 伴随着机器学习、人工智能技术的兴起, 大量自动学习算法被引入到协议分析中来, 而每种算法往往是针对特定的协议特征而设计的, 目前算法多是针对互联网协议而设计, 没有考虑物联网或工控协议, 协议简短、粒度细、无定界符的特征, 而这些协议数目在未知协议中比重较大, 因此需要考虑其特征对算法进行重新设计. 其次, 现有协议分析格式算法或系统中, 仅有少数考虑了一些协议设计的原理, 通过引入先验知识来解决识别中的冲突问题, 而完善协议分析的迭代过程, 可有效解决单纯依靠分类和数据挖掘算法不能解决的问题.

(2) 提出系统的协议语义体系, 丰富协议字段的语义推断方法. 由于字段语义的推断, 需要应用背景的支撑, 而在分析中很难得到, 协议语义一直没有得到系统的研究. 文献 [19] 将所有字段依据长度和变化情况将所有字段分为 4 类, 首次对协议的字段类型进行了一个全谱系的划分, 有益于实现协议字段语义的全面分析. 但是这种字段的划分过于简单, 没有体现协议设计的思想, 因此可结合应用层协议、无线协议、工控协议、物联网协议的特征建立完整的协议字段分层谱系, 实现从模糊到精确的语义推断, 并根据字段语义设计丰富多种的推断方法.

(3) 研究交互式增量协议逆向分析技术, 提高协议分析的完备性. 无论是协议格式提取还是协议状态机的构建都需要大量能够涵盖启动、运行、结束等全生命周期的报文, 然而基于网络流量的分析方法, 缺点就是无法获得完整的报文, 因此, 设计交互学习机制, 结合模糊测试技术采用主动的方法实现与分析对象报文交互, 从而提高协议逆向分析的完备性. 同时由于算法的效率较低, 无法对长周期、海量报文进行分析, 设计在线以及面向大量数据的高效分析方法也是未来研究方向之一.

(4) 研究多种方法综合集成机制. 从现有的协议逆向各个阶段所采用的方法看, 无论基于序列比对还是数据挖掘的方法, 其或考虑了协议字段的前后关系或者统计关系, 均有一定的效果. 在未来研究应当将这两种技术进行结合或进行交互验证, 从而提高整体的推断率. 如 Bossert 等人[25]提出了基于语义的协议格式推断方法, 结合了序列比对与信息挖掘的方法, 但是采用的语义要素有限, 字段间的邻居关系与语义关系的综合运用将有效改善分析效果.

6 总 结

伴随世界各国对网络空间安全日益重视, 对于网络流量监控、僵尸网络的识别、入侵流量过滤、恶意软件分析的需求变得更加迫切. 协议逆向技术有效可以实现对未知协议特征、格式以及交互行为的描述, 从而得到学术界和工业界的重视. 特别是伴随着 5G 网络技术和边缘计算的兴起, 物联网和工控协议以及新型应用必将大量涌现, 本文针对基于网络流量的分析技术进行了全面的分析, 对于协议逆向分析所涉及各个环节中的关键技术和演化过程进行了剖析, 指出了现有的研究不足与未来研究方向. 建议研究人员建立完善的分析理论, 充分考虑协议设计特点, 设计主被动的交互机制, 综合应用各种技术以提高协议逆向分析准确率和自动化程度.

References:

[1] Li MJ. Reverse extracting of protocol model based on dynamic binary analysis and its application [MS. Thesis]. Changsha: National University of Defense Technology, 2014 (in Chinese with English abstract). [doi: 10.7666/d.D01107920]

[2] Narayan J, Shukla SK, Clancy TC. A survey of automatic protocol reverse engineering tools. ACM Computing Surveys, 2016, 48(3): 1–26. [doi: 10.1145/2840724]

[3] Lee D. Yannakakis M. Principles and methods of testing finite state machines-a survey. Proc. of the IEEE, 1996, 84(8): 1090–1123. [doi: 10.1109/5.533956]

[4] Tridgell A. How SAMBA was written. 2003. http://www.samba.org/ftp/tridge/misc/french_cafe.txt

[5] Wine.org. About Wine. 2014. http://www.winehq.org/about/

[6] Wang YP, Li XJ, Meng J, Zhao Y, Zhang ZB, Guo L. Biprominer: Automatic mining of binary protocol features. In: Proc. of the 12th Int'l Conf. on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2011. Gwangju: IEEE, 2011. 179–184. [doi: 10.1109/PDCAT.2011.25]

[7] NetZob. 2020. https://netzob.org/

[8] Luo JZ, Yu SZ. Position-based automatic reverse engineering of network protocols. Journal of Network and Computer Applications,

2013, 36(3): 1070–1077. [doi: 10.1016/j.jnca.2013.01.013]

[9] Antunes J, Neves N, Verissimo P. Reverse engineering of protocols from network traces. In: Proc. of the the 18th Working Conf. on Reverse Engineering. Limerick: IEEE, 2011. 169–178. [doi: 10.1109/WCRE.2011.28]

[10] Wang Y, Zhang N, Wu YM, Su BB, Liao YJ. Protocol formats reverse engineering based on association rules in wireless environment. In: Proc. of the 12th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Melbourne: IEEE, 2013. 134–141. [doi: 10.1109/TrustCom.2013.21]

[11] Choi K, Son Y, Noh J, Shin H, Choi J, Kim Y. Dissecting customized protocols: Automatic analysis for customized protocols based on IEEE 802.15.4. In: Proc. of the 9th ACM Conf. on Security & Privacy in Wireless and Mobile Networks. Darmstadt: Association for Computing Machinery, 2016. 183–193. [doi: 10.1145/2939918.2939921]

[12] Cui WD, Kannan J, Wang HL. Discoverer: Automatic protocol reverse engineering from network traces. In: Proc. of the USENIX Security Symp. Berkeley: ACM, 2007. 1–14.

[13] Wang YP, Yun XC, Shafiq MZ, Wang LY, Liu AX, Zhang ZB, Yao DF, Zhang YZ, Guo L. A semantics aware approach to automated reverse engineering unknown protocols. In: Proc. of the 20th IEEE Int'l Conf. on Network Protocols (ICNP2012). Austin: IEEE, 2012. 1–10. [doi: 10.1109/ICNP.2012.6459963]

[14] Huang Q, Lee PPC, Zhang ZB. Exploiting intra-packet dependency for fine-grained protocol format inference. In: Proc. of the 2015 IFIP Networking Conf. (IFIP Networking). Toulouse: IEEE, 2015. 1–9. [doi: 10.1109/IFIPNetworking.2015.7145325]

[15] Cai J, Luo JZ, Lei FY. Analyzing network protocols of application layer using hidden semi-Markov model. Mathematical Problems in Engineering, 2016, 2016: 9161723. [doi: 10.1155/2016/9161723]

[16] Pan F, Hong Z, Du YX, Wu LF. Recursive clustering based method for message structure extraction. Journal of Sichuan University (Engineering Science Edition), 2012, 44(6): 137–142 (in Chinese with English abstract). [doi: 10.15961/j.jsuese.2012.06.016]

[17] Bermudez I, Tongaonkar A, Iliofotou M, Mellia M, Munafò M. Automatic protocol field inference for deeper protocol understanding. In: Proc. of the 14th IFIP Networking Conf. (IFIP Networking). Toulouse: IEEE, 2015. 1–9. [doi: 10.1109/IFIPNetworking.2015.7145307]

[18] Bermudez I, Tongaonkar A, Iliofotou M, Mellia M, Munafò M M. Towards automatic protocol field inference. Computer Communications, 2016, 84: 40–51. [doi: 10.1016/j.comcom.2016.02.015]

[19] Goo YH, Shim KS, Lee MS, Kim MS. Protocol specification extraction based on contiguous sequential pattern algorithm. IEEE Access, 2019, 7: 36057–36074. [doi: 10.1109/ACCESS.2019.2905353]

[20] Shevertalov M, Mancoridis S. A reverse engineering tool for extracting protocols of networked applications. In: Proc. of the 14th Working Conf. on Reverse Engineering (WCRE 2007). Vancouver: IEEE, 2007. 229–238. [doi: 10.1109/WCRE.2007.6]

[21] Trifilò A, Burschka S, Biersack E. Traffic to protocol reverse engineering. In: Proc. of the 2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications. Ottawa: IEEE, 2009. 1–8.[doi: 10.1109/CISDA.2009.5356565]

[22] Beddoe M. The protocol informatics project. 2020. http://www.phreakocious.net/PI/PI_Toorcon.pdf

[23] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 1970, 48(3): 443–453. [doi: 10.1016/0022-2836(70)90057-4]

[24] Cui WD, Paxson V, Weaver NC, Katz RH. Protocol-independent adaptive replay of application dialog. In: Proc. of the 13th Network and Distributed System Security Symp. (NDSS). San Diego: The Internet Society, 2006.

[25] Bossert G, Guihéry F, Hiet G. Towards automated protocol reverse engineering using semantic information. In: Proc. of the 9th ACM Symp. on Information, Computer and Communications Security. Kyoto: Association for Computing Machinery, 2014. 51–62. [doi: 10.1145/2590296.2590346]

[26] Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC. Detecting novel associations in large data sets. Science, 2011, 334(6062): 1518–1524. [doi: 10.1126/science.1205438]

[27] Cho C, Domagoj B, Shin E, Dawn S. Inference and analysis of formal models of botnet command and control protocols. In: Proc. of the ACM Conf. on Computer and Communications Security. 2010. 426–439.

[28] Du YX, Wu LF, Pan F, Hong Z. A semiautomatic protocol reverse method based on message sequence analysis. Computer Engineering, 2012, 38(19): 277–280 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-3428.2012.19.071]

[29] Zhang XM, Qiang Q, Wang WS, Wang ZF, Wei XL. IPFRA: An online protocol reverse analysis mechanism. In: Proc. of the 4th Int'l Conf. on Cloud Computing and Security. Haikou: Springer, 2018. 324–333.[doi: 10.1007/978-3-030-00015-8_28]

[30] Meng FZ, Zhang CR, Wu G. Protocol reverse based on hierarchical clustering and probability alignment from network traces. In: Proc. of the 2018 IEEE 3rd Int'l Conf. on Big Data Analysis (ICBDA). Shanghai: IEEE, 2018. 443–447.[doi: 10.1109/ICBDA.2018.8367724]

[31] Esoul O, Walkinshaw N. Using segment-based alignment to extract packet structures from network traces. In: Proc. of the 2017 IEEE Int'l Conf. on Software Quality, Reliability and Security (QRS). Prague: IEEE, 2017. 398–409.[doi: 10.1109/QRS.2017.49]

[32] Razo SIV, Anaya EA, Ambrosio PJE. Reverse engineering with bioinformatics algorithms over a sound android covert channel. In: Proc. of the 11th Int'l Conf. on Malicious and Unwanted Software (MALWARE). Fajardo: IEEE, 2016. 1–7. [doi: 10.1109/MALWARE.2016.7888724]

[33] Krueger T, Krämer N, Rieck K. ASAP: Automatic semantics-aware analysis of network payloads. In: Proc. of the Int'l ECML/PKDD Workshop on Privacy and Security Issues in Data Mining and Machine Learning. Barcelona: Springer, 2011. 50–63. [doi: 10.1007/978-3-642-19896-0_5]

[34] Wireshark. http://www.wireshark.org

[35] Baum LE, Petrie T, Soules G, Weiss N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The Annals of Mathematical Statistics, 1970, 41(1): 164–171. [doi: 10.1214/aoms/1177697196]

[36] Frey BJ, Dueck D. Clustering by passing messages between data points. Science, 2007, 315(5814): 972–976. [doi: 10.1126/science.1136800]

[37] Welch TA. 1984. A technique for high-performance data compression. Computer, 1984, 17(6): 8–19. [doi: 10.1109/MC.1984.1659158]

[38] Vyas K, Sherasiya S. Modified Apriori algorithm using hash based technique. Int'l Journal of Advance Research and Innovative Ideas in Education, 2016, 2(3): 1229–1234.

[39] Hei XH, Bai BB, Wang YC, Zhang L, Zhu L, Ji WJ. Feature extraction optimization for bitstream communication protocol format reverse analysis. In: Proc. of the 18th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications/13th IEEE Int'l Conf. on Big Data Science and Engineering (TrustCom/BigDataSE). Rotorua: IEEE, 2019. 662–669. [doi: 10.1109/TrustCom/BigDataSE.2019.00094]

[40] Fan YD, Zhu YN, Yuan L. Automatic reverse engineering of unknown security protocols from network traces. In: Proc. of the 4th IEEE Int'l Conf. on Computer and Communications (ICCC). Chengdu: IEEE, 2018. 1139–1148. [doi: 10.1109/CompComm.2018.8780606]

[41] Han JW, Pei J, Yin YW. Mining frequent patterns without candidate generation. ACM SIGMOD Record, 2000, 29(2): 1–12. [doi: 10.1145/335191.335372]

[42] Comparetti PM, Wondracek G, Krüegel C, Kirda E. Prospex: Protocol specification extraction. In: Proc. of the 30th IEEE Symp. on Security and Privacy. Oakland: IEEE, 2009. 110–125. [doi: 10.1109/SP.2009.14]

[43] Pei J, Han JW, Mortazavi-Asl B, Wang JY, Pinto H, Chen QM, Dayal U, Hsu MC. Mining sequential patterns by pattern-growth: The PrefixSpan approach. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(11): 1424–1440. [doi: 10.1109/TKDE.2004.77]

[44] Wang XW, Lv KZ, Li B. IPART: An automatic protocol reverse engineering tool based on global voting expert for industrial protocols. Int'l Journal of Parallel, Emergent and Distributed Systems, 2020, 35(3): 376–395. [doi: 10.1080/17445760.2019.1655740]

[45] Antunes J, Neves N. Building an automaton towards reverse protocol engineering. 2009. https://www.di.fc.ul.pt/~nuno/PAPERS/INFORUM09.pdf

[46] Wang YP, Zhang ZB, Yao DF, Qu BY, Guo L. Inferring protocol state machine from network traces: A probabilistic approach. In: Proc. of the Int'l Conf. on Applied Cryptography and Network Security. Berlin: Springer, 2011. 1–18. [doi: 10.1007/978-3-642-21554-4_1]

[47] Meng FZ, Liu Y, Zhang CR, Li T, Yue Y. Inferring protocol state machine for binary communication protocol. In: Proc. of the 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA). Ottawa: IEEE, 2014. 870–874. [doi: 10.1109/WARTIA.2014.6976411]

[48] Leita C, Mermoud K, Dacier M. ScriptGen: An automated script generation tool for honeyd. In: Proc. of the 21st Annual Computer Security Applications Conf. (ACSAC2005). Tucson: IEEE, 2005. 200–214. [doi: 10.1109/CSAC.2005.49]

[49] LaRoche P, Burrows A, Zincir-Heywood AN. How far an evolutionary approach can go for protocol state analysis and discovery. In: Proc. of the 2013 IEEE Congress on Evolutionary Computation. Cancun: IEEE, 2013. 3228–3235. [doi: 10.1109/CEC.2013.6557965]

[50] Zhang Z, Wen QY, Tang W. Mining protocol state machines by interactive grammar inference. In: Proc. of the 3rd Int'l Conf. on Digital Manufacturing & Automation. Guilin: IEEE, 2012. 524–527. [doi: 10.1109/ICDMA.2012.125]

[51] Dupont P, Lambeau B, Damas C, van Lamsweerde A. The QSM algorithm and its application to software behavior model induction. Applied Artificial Intelligence, 2008, 22(1-2): 77–115. [doi: 10.1080/08839510701853200]

[52] Ládi G, Buttyán L, Holczer T. Message format and field semantics inference for binary protocols using recorded network traffic. In: Proc. of the 26th Int'l Conf. Software, Telecommunications and Computer Networks. Split: IEEE, 2018. 1–6. [doi: 10.23919/SOFTCOM.2018.8555813]

附中文参考文献:

[1] 李美剑. 基于动态二进制分析的协议模型逆向提取及其应用研究[硕士学位论文]. 长沙: 国防科学技术大学, 2014. [doi: 10.7666/d.

D01107920]
[16] 潘璠, 洪征, 杜有翔, 吴礼发. 基于递归聚类的报文结构提取方法. 四川大学学报(工程科学版), 2012, 44(6): 137–142. [doi: 10.15961/j.jsuese.2012.06.016]
[28] 杜有翔, 吴礼发, 潘璠, 洪征. 一种基于报文序列分析的半自动协议逆向方法. 计算机工程, 2012, 38(19): 277–280. [doi: 10.3969/j.issn.1000-3428.2012.19.071]

王占丰(1982－), 男, 在站博士后, 主要研究领域为网络测量, 网络性能分析与建模, 网络安全.

张嘉玮(1991－), 男, 硕士, 主要研究领域为网络安全.

程光(1973－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为网络安全, 网络测量与行为学, 未来网络安全.

孙中豪(1989－), 男, 博士, 国家互联网应急中心工程师, 主要研究领域为工控系统网络安全, 关键基础设施安全防护.

马玮骏(1980－), 男, 博士, 主要研究领域为分布式存储, 大数据分析, 人工智能.

胡超(1984－), 男, 博士, 讲师, 主要研究领域为网络测量, SDN, 网络空间安全.