协议逆向工程研究进展 *

潘 璠, 吴礼发, 杜有翔, 洪 征
(解放军理工大学 指挥自动化学院, 南京 210007)

摘 要: 首先给出了协议逆向工程的形式化定义,并探讨了主要应用领域的特定需求;然后从报文序列分析和指令执行序列分析两个方面介绍了协议逆向技术的研究现状,并对两类技术的优劣进行了比较;最后结合当前方案的缺陷和实际应用的需求,对协议逆向技术的发展趋势进行了展望。

关键词: 协议逆向工程; 多序列比对; 文法推断; 动态污点分析; 数据流分析
中图分类号: TP393.08 **文献标志码**: A **文章编号**: 1001-3695(2011)08-2801-06
doi:10.3969/j.issn.1001-3695.2011.08.001

Overviews on protocol reverse engineering

PAN Fan, WU Li-fa, DU You-xiang, HONG Zheng
(*Institute of Command Automation, PLA University of Science & Technology, Nanjing* 210007, *China*)

Abstract: First, this paper gave the formal definition of protocol reverse engineering, and discussed the specified requirements of application domains. From the two aspects of network trace analysis and execution trace analysis, introduced the key technologies of protocol reverse engineering, followed with the evaluation on the two kinds of technology. Based on the disadvantages of the current solutions and requirements from practical applications, discussed the future of protocol reverse engineering.

Key words: protocol reverse engineering; multiple sequences alignment; grammatical inference; dynamic taint analysis; data flow analysis

0 引言

协议是为进行网络数据交换而建立的一系列的规则、标准和约定,是计算机网络及数据通信的核心,也是网络安全领域的重点研究对象。当前许多网络安全相关的应用都以协议的详细描述信息为基础,如入侵检测[1~3]、模糊测试[4~7]、协议重用[8~10]和一致性测试[11]等。但由于目前使用的协议大部分都是私有协议,缺乏正式的描述文档,研究人员越来越多地采用协议逆向的手段实现协议信息的提取。协议逆向工程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、语法和语义的过程。开源项目 Samba[12]花费约 12 年时间对 Windows 平台上的 SMB/CIFS 协议进行逆向,成功实现了跨平台的文件和打印共享机制。除此之外,公开的协议逆向应用还有 Rdesktop[13]、Pidgin[14]等。这些针对非公开协议的逆向应用取得了较为理想的效果,但都存在一些共同的缺陷:过度依赖人工分析、过程冗长耗时。随着网络规模的扩大和应用种类的增多,对协议逆向的准确性和时效性的要求越来越高,研究如何提高协议逆向的自动化程度和准确度具有重要的现实意义。

1 问题的描述与定义

目前对协议逆向技术的研究主要通过自然语言对问题域和逆向方案进行描述,在表达上缺乏精确性和简明性,同时难以从理论上对技术路线的可行性进行验证。为了剖析协议逆向的本质,分析现有方案的优势与局限性,本文首先给出协议逆向的形式化定义。

1.1 协议逆向目标

协议系统由协议实体和通道组成,协议实体之间利用通道事件进行通信。现有网络安全应用需要模拟协议实体之间的报文交互行为,其关注的重点为协议实体所有可接受的输入报文序列。因此,报文格式和响应规则是协议逆向的主要目标,而通道事件的时序关系和协议实体的内部事件属于无关紧要的内容。下文中提到的协议逆向都是指对协议实体的逆向,而不针对整个协议系统。

对于不同的输入报文,协议实体不仅改变内部状态,还生成与状态和输入相关的报文输出。本文在 Mealy 机[15]的基础上,将协议实体符号化为一系列状态,以报文的接收和发送替代系统事件作为输入和输出,忽略内部事件对系统状态的影响。对协议实体作如下扩展定义:

收稿日期: 2011-03-21; **修回日期**: 2011-04-25 **基金项目**: 解放军理工大学预先研究基金资助项目

作者简介: 潘璠(1987-),男,安徽芜湖人,博士研究生,主要研究方向为协议逆向分析、漏洞挖掘(dynamozhao@163.com);吴礼发(1968-),男,教授,博导,博士,主要研究方向为网络安全;杜有翔(1986-),男,硕士研究生,主要研究方向为协议逆向分析;洪征(1979-),男,副教授,博士,主要研究方向为网络安全.

定义 1　协议实体可定义为六元组 $M=(Q,\Sigma,\Delta,\delta,\lambda,q_0)$，其中：

a) $Q=\{q_0,q_1,\cdots,q_n\}$ 是有限状态集合；

b) $\Sigma=\{\sigma_0,\sigma_1,\cdots,\sigma_m\}$ 是有限输入报文格式的集合；

c) $\Delta=\{a_0,a_1,\cdots,a_r\}$ 是有限输出报文格式的集合；

d) $\delta: Q\times\Sigma\rightarrow 2^Q$ 是状态转换函数；

e) $\lambda: Q\times\Sigma\rightarrow\Delta$ 是输出函数；

f) $q_0\in Q$ 是初始状态。

结合现有应用的需求，协议逆向的目标可定义为协议实体对应的集合 Σ 与状态转换函数 δ。

协议报文由若干个域组成，域具有域类型、域边界、语义和取值约束等属性[16]。域类型包括常量、整型变量、文本字符串和二进制流等；域边界为确定域在报文中位置与长度的属性；域的语义表示协议将如何使用该域，包括分隔符、长度域、校验和、时间戳和会话标志符等[1]；域的取值约束为域在协议可允许的取值范围。另外，报文的解析是一个层次化、序列化的过程，域之间存在包含关系、顺序关系和并列关系[17]。考虑到域结构和域属性在协议逆向应用中的必要性，对集合 Σ 中的输入报文格式 σ_i 作如下定义：

定义 2　协议在某一状态可接受的输入报文格式 σ，可定义为一个属性文法 (G,V,F)，由一个上下文无关文法和一系列语义规则构成。其中：

a) G 为上下文无关文法，可用扩展巴科斯范式（ABNF）来表示。一个简单的示例如下：

$G::=C_1(C_2|C_3|C_4)[C_5]C_6$

$C_5=\{C_7|C_8|\cdots\}*$

符号 C_i 对应于协议的基本域。扩展巴科斯范式是 IETF 用来定义通信协议的元语言符号表示法，具体语法规则参见 RFC2234 文档。

b) V 是为文法符号 C_i 配备的穷属性集，由一组基础属性 $\{v_1,v_2,\cdots,v_k\}$ 和语义属性 v_s 组成。基础属性如长度、偏移等为所有域的共有属性，可在语法分析的过程中进行计算和传递；语义属性如端口、校验和等表示符号 C_i 的意义，并且每个域只能拥有一种语义属性。

c) $F=\{f_1,f_2,\cdots,f_k\}$ 是关于属性规则的有穷集，f_i 为对应属性 v_i 的取值提供解释。对于基础属性，属性规则将其定义为自身。对于语义属性，f_i 可以表达同一报文内的域约束关系，如指针属性 v_p 对应的 f_p 为 $C_1.v_l=\text{offset}(C_2)$；$f_i$ 还可以表达输入报文之间的域约束关系，如序号属性 v_s 对应的 f_s 为 $(\sigma_j.C_1.v_s=\text{inc}(\sigma_i.C_1.v_s))\wedge(\sigma_j=\text{next}(\sigma_i))$。需要特别指出的是，不仅输入报文之间存在域约束关系，输入与输出报文之间也存在域约束关系。

1.2　协议逆向模型

在给出协议逆向目标的基础上，对协议逆向作如下定义：

定义 3　定义 X 为输入的网络数据流，LOG 为程序解析 X 的指令记录序列，那么协议逆向可以定义为：通过分析 X 或对应的 LOG，获取 Σ 中各个 σ_i 对应的 (G_i,V_i,F_i) 和状态转换函数 δ 的过程。

协议实体间的会话由一系列报文组成，单个会话的报文序列对应于状态机的一条路径。为了获取未知的状态转换函数 δ，首先需要将初始数据流以会话为单位进行划分。由于协议实体的内部机制不可知，其状态只能用已接收的报文格式序列来表示。因此要获取状态转换函数 δ，必须先获取格式 σ_i。另外，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。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提取三个阶段，如图 1 所示。

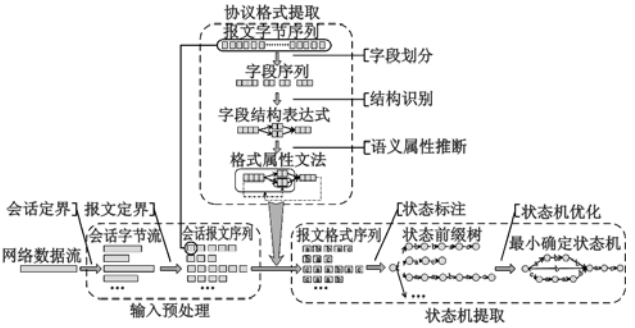

图1　协议逆向模型

在输入预处理阶段，经过会话定界和报文定界两个步骤，原始数据流被划分为与独立会话中各个报文对应的子序列；协议格式提取阶段以子序列为输入，经过域划分、结构识别和语义属性推断三个步骤，逐步获取报文格式 σ_i 对应的 (G_i,V_i,F_i)；在状态机提取阶段，经过状态标注构建状态前缀树以合并状态冗余，再经过状态机优化步骤获取对应的最小确定状态机。能否实现会话定界决定了能否获取状态转换序列，能否实现报文定界则决定了能否对状态进行标注，因此输入预处理阶段是协议逆向的基础。

虽然协议逆向以输入报文为目标，但由于输入报文与输出报文之间可能存在域约束关系，因此还需要对输出报文格式进行提取。输出报文的格式提取算法与输入报文的提取算法相同，但在最终输出协议描述时仅需要保留与输入报文关联的部分。

1.3　协议逆向需求

随着研究的不断深入，协议逆向技术逐渐在多个网络安全领域得到了应用。不同应用对协议逆向结果的需求有所不同，所采用的技术路线也有所差别。常见的协议逆向应用包括如下几种：

a) Fuzzing 测试。Fuzzing 技术是一种融合了协议知识、软件知识、漏洞知识，采用试探性攻击方法的软件漏洞挖掘技术。当前的高级 Fuzzing 技术分为两类[18]：白盒 Fuzzing 技术和基于知识的 Fuzzing 技术。白盒 Fuzzing 技术主要基于代码覆盖的思想，而基于知识的 Fuzzing 技术主要利用文件格式知识或者协议格式知识构造测试用例。根据协议知识生成的测试用例可以有效地突破目标程序中的检查和验证，如关键字、校验和、长度计算、加/解密算法等。为了实现对未知协议的 Fuzzing 测试，协议逆向技术至少需要完成输入报文格式集 Σ 的提取。

b) 网络入侵检测。网络入侵检测作为一种积极主动的

安全防护技术,通过检查网络中是否存在安全策略的行为和被攻击的迹象,对可能的入侵进行响应和拦截。入侵检测可分为异常检测和误用检测,两者都需要大量的专家知识对正常行为和入侵行为进行区别定义。传统的入侵检测系统无法理解网络事件之间存在的关联关系,并且无法识别使用未知协议的网络行为。为了提高入侵检测系统识别的准确率和扩展性,协议逆向技术至少需要提取输入报文格式集 Σ 和状态转换函数 δ。

c)协议重用。网络应用的跨平台移植和重用通常需要了解未知协议的工作机制,如上文中提到的 SMB 协议的重用。在僵尸网络的跟踪和监测中,为了伪装成被控端加入到控制网络,事先需要分析 C&C 协议的工作原理。由于工作量巨大且耗时极长,人工提取协议信息的方式远不能满足协议重用对时效性的要求。若要实现自动化的协议重用,协议逆向技术至少需要提取输入报文格式集 Σ 中与输出存在约束关系的部分。

2　研究现状与分析

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。在协议逆向工程领域,国内外已进行了较为深入的研究。根据分析对象的不同,现有的协议逆向技术大致分为两类[4]:报文序列(network trace)分析和指令执行序列(execution trace)分析。

2.1　报文序列分析技术

报文序列分析技术以嗅探得到的网络数据流 X 为分析对象,其可行性在于以下两点:a)单个报文样本的数据流为报文格式的一个实例,同一报文格式对应的多个报文样本具有相似性;b)会话是协议实体之间的完整交互过程,协议实体在一次会话中的状态转换序列是协议状态机的一个子集。同一会话内报文的时序关系包含了部分协议状态转换的信息。

国外对基于报文序列分析的协议逆向方法研究较早,最为典型的是 Beddoe[19] 于 2004 年启动的 PI 项目(protocol information project)。在生物信息学中,需要从 DNA 中搜寻产生蛋白质的特定基因,与此相类似,很多网络应用需要从大量的网络数据包中寻找具有特定含义的域。由于两者的这种相似性,PI 项目通过引入生物信息学的序列比对算法,尝试对目标协议的结构信息进行分析。其具体流程包括以下几步:a)通过局部序列比对算法计算报文字节序列之间的相对距离,获得距离矩阵;b)采用非加权成对群算术平均法(unweighted pair group method with arithmetic mean, UPGMA),根据序列顺序构建系统树;c)采用渐进比对算法遍历系统树实现多序列比对。由于实际应用中的网络报文具有序列长、数量多的特点,因此实现多序列精确匹配需要耗费大量时间和计算资源。PI 利用构造系统树的启发式方法引导多序列比对的执行,可以有效降低序列比对的时间复杂度,提高算法的执行效率。根据序列比对结果,PI 的局限性在于:a)仅可获得域的边界和基本属性,要获得语义属性还依赖于人工分析;b)算法仅针对协议的报文结构,并不能得到协议状态机信息;c)对于紧凑、简单、与基因序列较为类似的协议,PI 的识别效果较好,而对于复杂、冗余域较多的协议,其效率和准确度较低。

Leita 等人[8] 以 PI 项目中的算法流程为基础,实现了蜜网 Honeyd 配置脚本自动提取工具 ScriptGen。ScriptGen 首先获取报文会话序列,并为每次会话维护一个状态机;在此基础上通过 PI 中的协议域识别算法对状态机进行合并和简化;最后利用状态机,生成 Honeyd 可用的 Python 配置脚本。为了实现报文正确重放的需求,ScriptGen 在域划分之后还对协议域之间的约束关系进行了识别。每一种约束关系都对应于一条识别规则,如会话标志域为请求报文和应答报文中取值始终一致的变量域,而序号域为连续报文中取值递增的数值域。ScriptGen 考虑了输入与输出之间的约束关系,实现了部分语义的提取,其局限性在于构建初始状态机之前没有对报文进行聚类,在样本报文数较多时会导致初始状态机过于庞大。此外,这种方法对样本的数量和多样性要求较高,而且状态机的精确程度依赖于最大状态数和状态最大出度值的设定。

采用序列比对算法进行域划分需要获得大量的报文样本,但在一些环境下该条件无法得到满足。RolePlayer[9] 对样本集的完备性没有很高的要求,但需要拥有协议的一些先验知识。RolePlayer 不对协议的完整结构进行分析,其重点在于识别报文结构中用户参数、状态标志、长度等动态域,以此为基础,RolePlayer 能够对接收到的新报文进行协议类别的判断,并更新输出报文中的动态域,实现报文自动重放。RolePlayer 的局限性在于其识别效果主要依赖于先验知识的丰富程度。此外,这种技术对变长域和嵌套结构较多的协议识别效果较弱。

李伟明等人[20] 在 PI 的基础上,提出了自动化网络协议模糊测试的完整方案。该方案首先采用类型匹配提取同类型的报文序列,再通过多序列比对算法将报文中的不变域和可变域分离开来,进而推断出报文中的文本域、二进制域和长度域,最后依据报文格式自动生成 SPIKE 测试脚本。该方案针对性强,但没有考虑域之间序号、cookie 等约束语义,不能保证所生成的 Fuzzing 测试数据绝对有效。此外,方案没有提取状态机信息,无法对整个报文的交互过程进行模糊测试。

对于绝大部分协议而言,报文解析是一个层次化的过程,在逐层解析的过程中通常有一些格式标志域决定了子结构的解析方式。针对这一特征,Cui 等人[21] 提出了以递归分类为核心的协议逆向方案 Discoverer。该方案首先按照文本和二进制两种属性对报文字节流进行分词,再采用序列比对算法对报文属性序列进行初始聚类;进而对各个域进行语义推断,根据识别出的格式标志域取值进行再分类;不断重复这一过程,直到子类中的报文数目小于某个阈值。为了避免对样本集过分类,Discoverer 还会对属性和语义序列相似度高的子类进行合并。Discoverer 通过分词实现初始域划分,并以域为基元进行序列比对,相比于 PI 采用字节作为基元,Discoverer 对协议的逆向更具针对性。此外,Discoverer 还能够识别格式标志、长度、偏

移和 cookie 等语义。每一种语义的识别采用了相应的一套启发式识别规则,如格式标志域的取值较为固定,根据域值聚类后报文的属性序列相同;又如长度域通常为固定长度的二进制域,并且取值与子序列的长度相关联。Discoverer 最终得到带有域语义、层次化的完整协议格式描述,但没有考虑到状态机信息的提取。Deyoung 在假设协议格式已知的前提下,引入文法推断中的 k-RI 和 k-TSSI 算法提取状态机信息[22],弥补了 Discoverer 的不足。

除了基于域划分的分析方法之外,报文序列分析技术中还有一些针对特定应用的分析方法。PEXT[23] 首先以最长公共子序列长度为相似度指标对报文样本进行聚类和状态标记,进而生成对应于会话的状态转换序列,最终合并得到涵盖所有测试用例的最小确定状态机。与 PEXT 采用聚类方法对报文进行状态标注不同,BFS[24] 通过分析报文中各字节的取值分布来识别状态标志域,并构建协议状态机。BFS 认为协议报文中通常存在一些报文状态域标志了当前的状态逻辑,如 HTTP 请求 GET /pub/WWW/TheProject. html HTTP/1. 1 中仅 GET 代表了当前状态,/pub/WWW/TheProject. html 指定要获取的页面,而 HTTP/1. 1 为版本说明。会话中状态转换模式相对固定,不同会话内报文状态域的取值分布具有相似性,而这种相似性即是 BFS 识别状态域的依据。Antunes 等人[25] 提出一种以偏序比对算法为基础,构建有穷自动机识别报文的方法。协议格式可看做一种正则语言,所有合法的报文都遵守语言的句法规则,因此对报文格式正确与否的判断可以由相应的识别自动机来完成。与协议状态机不同,识别自动机的状态和状态转换并没有具体的语义,仅表示字符识别的过程。以上三种方法没有考虑报文格式的提取,最终结果只包括协议状态机信息,因此只能满足协议识别的基本需求。

2.2 指令执行序列分析技术

指令执行序列分析技术是指以数据解析过程中的指令执行序列为分析对象的一类技术,其理论依据在于:a)协议实体接受报文的过程即报文解析的过程,通过对程序处理域边界及域的使用方式可以获得协议格式的表达式和属性;b)完整会话的指令序列可划分为单个报文指令序列的排列,子序列之间的顺序关系包含了状态转换信息。

为了获取数据处理流程的指令执行序列,现有研究均以动态污点分析(taint data analysis)[26] 为基础。动态污点分析是一种在指令级对外部数据传播过程进行跟踪与分析的技术,近年来在漏洞分析、恶意代码检测和测试用例生成等领域得到广泛应用。指令执行序列分析技术的基本思想是将所有网络传输数据作为污点数据源,监控协议实体处理这些污点数据的流程,并对指令执行序列进行分析从而获取协议描述。

Jcaballero 等人[27] 于 2007 年提出采用动态污点分析实现协议逆向,并设计了原型系统 Polyglot。出于执行效率和可回溯性的考虑,系统采用离线分析的方式对报文格式中的分隔符、定位符和关键字进行识别。Polyglot 首先识别域为定长还是变长,并划分域边界。定长域的识别比较简单:只要域的解析范围由指令中的常数参数指定,可直接认定其为定长域。变长域的识别方式大致有两种:一种为定位方式,通过长度或者指针实现对变长域边界的定位;另一种为分隔符方式,通过特定的字符标志变长域的结束。除了划分报文的域边界之外,Polyglot 还通过跟踪污点数据与非污点的常量字符(串)之间发生的比较操作识别关键字。与 Polyglot 类似,何永君等人[28] 也提出基于动态污点分析的网络协议逆向解析方法,并在 DynamoRIO 平台上实现了相应的原型系统。

Polyglot 逆向得到的报文格式为域的线性序列,而实际上协议实体对报文的处理是一种层次化的解析过程,域与域之间可能存在包含、并列以及序列等关系。Lin 等人[17] 结合域解析的上下文环境,提出了基于污点数据分析的域结构识别方案 AutoFormat。该方案在污点跟踪过程中,不仅记录所有与污点数据相关的操作指令,还保存对应的函数调用栈。每条记录以四元组$\langle o,c,s,l\rangle$的形式保存。其中:o 为域对应于报文起始位置的偏移;c 为域的内容;s 为操作指令执行时的函数调用栈;l 为操作指令的地址。在识别过程中,AutoFormat 通过判断偏移范围是否覆盖对域之间的包含关系进行识别,通过判断指令子序列和上下文环境是否相似对域之间的并列关系进行识别,通过判断指令子序列的调用顺序对域之间的序列关系进行识别。Wondracek 等人[29] 在 Polyglot 的基础上提出了一种改进方案,基于多次监控的分析结果,将所有格式相同的报文进行语义信息融合,从而提取通用性更好的报文结构。

AutoFormat 中采用启发式策略对预定义的域属性和依赖关系进行识别,但无法处理复杂结构和未知语义。为了更具一般性,Tupni[16] 在 Polyglot 与 AutoFormat 算法思想的基础上,根据处理复杂结构的控制流特征进行识别,并通过动态数据流分析获得未知语义类型的符号谓词约束,最终以 BNF 的形式输出协议逆向结果。Tupni 还尝试利用平台相关的 API 函数规范描述进行辅助分析,一方面函数的参数可以确定域的类型和语义,另一方面对函数功能的描述可以辅助识别约束关系。

Comparetti 等人[4] 借鉴了 AutoFormat 对域结构的识别策略,提出了一个较为完整的协议逆向方案 Prospex。与 AutoFormat 相比,Prospex 主要在三个方面进行了改进:a)结合指令操作的上下文环境对相同状态下报文的聚类,结果更精确;b)对报文序列进行状态标注,实现了协议状态机的逆向;c)根据逆向得到的协议描述,可以自动化生成模糊测试工具 Peach 所需的测试脚本。Prospex 的局限性在于没有考虑域的语义和取值约束关系,无法保证生成的模糊测试数据的有效性。

出于安全性的考虑,越来越多的协议采用加密技术对报文进行安全防护。以上的各项指令序列分析方法都是通过对明文数据进行动态污点跟踪实现的,对加密报文无能为力。Wang 等人[30] 对多个加密协议进行了跟踪分析,发现报文解密过程通常独立于解析过程,并且解密过程中的算术指令与比特位操作指令要明显多于解析过程。根据这一规律,他们提出了基于缓冲区数据生命周期分析的解密报文识别方案 ReFormat。该方案主要包括两个步骤:a)对于指令序列中的每个函数,统计其中算术与比特位操作指令在所有指令中所占的比例;如果自某个函数开始,之后所有函数中的算术与比特位操作指令所占的比例明显减少,则该函数的起始位置被认定为解密过程的

结束;b)根据记录的缓冲区生命周期信息,把那些在解密过程发生写操作、在解析过程中发生读操作的缓冲区识别为解密报文缓冲区;如果有多个缓冲区符合要求,则将这些缓冲区按照读取顺序排序,共同作为解密报文缓冲区;在确定解密报文缓冲区以后,即可采用针对明文数据的解析方法完成报文格式提取。

在实际应用中,某些加密协议的解密过程与解析过程交替进行,此时 ReFormat 所使用的解密报文缓冲区识别策略就会失效。Dispatcher[1] 沿用了 ReFormat 中的缓冲区数据生命周期分析和算术指令统计的思想,但识别的粒度细化到编码函数级(包括加/解/密函数、hash 函数、解压缩函数以及混淆函数),并允许多个加密过程与解析过程交替,识别的准确度更高。在已有的对输入报文格式识别技术的基础上,Dispatcher 提出了包含输出报文格式提取的双向协议逆向方案。与输入报文格式识别采用的动态污点分析不同,输出报文的格式识别通过跟踪缓冲区生命周期内的复制与合并实现。此外,Dispatcher 对函数参数与指令操作数语义进行了预定义,并跟踪语义类型在执行过程中的传播,实现了对多种字段语义识别。虽然初步的实验证明了 Dispatcher 的可行性,但其识别策略的通用性和准确性还需要进一步的验证。

2.3　两类技术的比较

由以上介绍可以看出,对两类协议逆向技术的研究都取得了较大进展,并在多个领域得到了初步应用。结合理想的协议逆向流程,本文分别从逆向能力、准确度、限制条件和分析速度四个方面对两类技术进行比较。

1)逆向能力　在报文格式提取阶段,报文序列分析方法根据取值的变化频率和特征对字节位进行合并,域划分的本质为对协议格式的文法推断;指令执行序列分析方法根据对报文数据块的分段读取进行域划分,其本质为指令级的数据流分析。指令执行序列分析是自顶向下的过程,可以利用域划分的过程信息获取域之间的结构关系;而报文序列分析是自底向上的过程,无法实现对域结构关系的提取。

由于报文数据流中只包含协议报文的句法信息,因此报文序列分析难以实现域的语义属性推断。即使在有先验知识的条件下,也只能识别部分语义[9]。而指令执行序列分析可根据指令子序列处理和使用数据的方式,实现对字段语义属性和约束关系的识别。

对于加密协议,由于报文各字节位的频率和取值特征被破坏,无法采用报文序列分析方法实现协议逆向。而指令执行序列分析通过识别解密过程并对解密报文进行数据流分析,可实现对加密协议的逆向。

2)限制条件　报文序列分析只需截取协议报文的数据包,指令执行序列分析则要求在指令级监控协议实体对报文的解析过程。由于实现复杂且有可能无法获得协议实体的控制权,在实际应用时后者的局限性要明显大于前者。在输入预处理阶段,报文序列分析需要依赖底层协议以实现会话报文序列的划分,而指令执行序列分析则通过对系统调用的识别来实现。协议格式提取阶段,报文序列分析方法需要大量同格式的报文,而指令执行序列分析方法则只需要单个报文的指令执行序列。

3)准确度　在报文格式提取阶段,报文序列分析的准确度依赖于样本集的完备程度。Gold 证明了在仅提供正例的情况下,正则语言不可能完全通过学习得到[31]。报文样本集只能提供协议实体可接受的正例,报文序列分析一般仅能得到协议格式的近似描述。而在指令执行序列中,存在数据流和控制流信息与域划分、结构关系和语义属性一一对应。理论上来说,指令执行序列分析可获取协议格式的精确描述,在应用中的准确度取决于具体实现策略。在状态机提取阶段,指令执行序列分析结合上下文环境对状态进行标注,可有效避免状态的错误合并,相比报文序列分析准确度更高。由于两类方法都没有将协议实体的内部事件考虑在内,因此无法完全消除状态机中冗余。

4)分析速度　报文序列分析的速度取决于域划分时的多序列比对速度,而指令执行序列分析的速度取决于动态污点跟踪和数据流分析的速度。在同等样本集规模的条件下,报文序列分析的速度要远快于指令执行序列分析的速度。

就逆向能力和准确度而言,指令执行序列分析要优于报文序列分析;但考虑到限制条件和分析速度,后者又优于前者。因此在应用时,需要根据应用的特定需求和环境来选择何种技术路线。

3　结束语

随着网络应用种类逐渐增多和规模日趋庞大,自动化的协议逆向技术将得到更多的关注,相关的成果也将在入侵检测、模糊测试和协议重用等领域广泛应用。由研究现状可以看出,协议逆向工程并不是一种全新的技术,而是对多序列比对、文法推断、动态污点分析和数据流分析等多种技术的综合和扩展。虽然协议逆向技术的可行性已得到充分验证,但是由于方法本身的局限性和实现的复杂性,逆向得到的协议描述与理想流程的目标还有较大的差距。将来协议逆向工程的研究方向主要集中于以下几个方面:

a)现有方案中缺乏对语义属性提取的研究,获取的协议逆向描述有待进一步完善。考虑到分析对象仅含有句法信息,报文序列分析只能通过引入外部先验知识实现语义属性提取;而指令执行序列分析面临的挑战,则是如何定义语义属性的数据流特征,在确保准确度的同时提高识别效率。

b)现有方案都依赖于样本集的完备程度,如果对应于某种格式的报文不在样本集中出现,则无法逆向得到该格式的描述。由于依赖于样本的取值和频率特征,报文序列分析方法无法克服这一缺陷。指令执行序列分析方法则可在动态污点分析的基础上,结合对协议实体目标代码的静态分析,实现新样本的自动生成。

c)随着安全意识的不断提升,越来越多的协议开始采用加密传输机制,针对加密协议的逆向技术将成为未来的研究重点。当前方案多数都没有考虑加密协议的逆向,少数指令执行序列方案也仅依据算术指令占全部指令的比重识别解密过程,其准确度无法保证。考虑到解密过程与解析过程的数据流特征不同,下一步应结合数据流特征匹配以提高解密过程识别的

准确率。在此基础上还可建立已知加密算法特征库,以实现加密算法类型的识别和解密报文的快速定位。

d)当前研究仅针对较为理想的单通道双端协议模型,没有考虑多通道、多交互方协议等应用场景[23]。为了提高协议逆向技术的实用性,后续研究应在现有技术路线的基础上进行扩展,通过完善识别策略实现对复杂协议的逆向。

随着准确度和实用性的逐步提高,协议逆向工程将成为网络安全研究方向不可或缺的一类技术。随着网络安全需求的不断扩大,除了当前研究主要关注的互联网外,协议逆向技术还将在 SCADA、卫星通信网等环境中发挥越来越重要的作用。

参考文献:

[1] CABALLERO J, POOSANKAM P, KREIBICH C, *et al*. Dispatcher: enabling active botnet infiltration using automatic protocol reverse-engineering[C]//Proc of ACM Conference on Computer and Communications Security. 2009:621-634.

[2] LEITA C, DACIER M, MASSICOTTE F. Automatic handling of protocol dependencies and reaction to 0-day attacks with ScriptGen-based honeypots[C]//Proc of Symposium on Recent Advances in Intrusion Detection. 2006:185-205.

[3] DREGER H, FELDMANN A, MAI M, *et al*. Dynamic application-layer protocol analysis for network intrusion detection[C]//Proc of the 15th USENIX Security Symposium. 2006: 257-272.

[4] COMPARETTI P M, WONDRACEK G, KRUEGEL C, *et al*. Prospex: protocol specification extraction[C]//Proc of the 30th IEEE Symposium on Security and Privacy. 2009:110-125.

[5] BRUMLEY D, CABALLERO J, LIANG Zhen-kai, *et al*. Towards automatic discovery of deviations in binary implementations with applications to error detection and fingerprint generation[C]//Proc of the 16th USENIX Security Symposium. 2007:213-228.

[6] GROSSO C D, ANTONIOL G, PENTA M D, *et al*. Improving network applications security: a new heuristic to generate stress testing [C]//Proc of Data Genetic and Evolutionary Computation Conference. 2005:1037-1043.

[7] MCMINN P, HARMAN M, BINKLEY D, *et al*. The species per path approach to search-based test data generation[C]//Proc of International Symposium on Software Testing and Analysis. 2006:13-24.

[8] LEITA C, MERMOUD K, DACIER M. ScriptGen: an automated script generation tool for honeyd[C]//Proc of the 21st Annual Computer Security Applications Conference. 2005:203-214.

[9] CUI Wei-dong, PAXSON V, WEAVER N C, *et al*. Protocol-independent adaptive replay of application dialog[C]//Proc of the 13th Annual Network and Distributed System Security Symposium. 2006.

[10] NEWSOME J, BRUMLEY D, FRANKLIN J, *et al*. Replayer: automatic protocol replay by binary analysis[C]//Proc of ACM Conference on Computer and Communications Security. 2006:311-321.

[11] LEE D, SABNANI K. Reverse-engineering of communication protocols[C]//Proc of International Conference on Network Protocols. 1993:208-216.

[12] TRIDGELL. How samba was written[EB/OL]. (2003-08)[2011-03]. http://samba.org/ftp/tridge/misc/french_cafe.txt.

[13] CHAPMAN, MATTHEW. Rdesktop: a remote desktop protocol client [EB/OL]. (2006-09)[2011-03]. http://sourceforge.net/projects/rdesktop/.

[14] CONTRIBUTORS. About pidgin[EB/OL]. (2007-08)[2011-03]. http://www.pidgin.im/about/.

[15] MEALY G H. A method for synthesizing sequential circuits[J]. Bell System Technical Journal, 1955, 34(5): 1045-1079.

[16] CUI Wei-dong, PEINADO M, CHEN K, *et al*. Tupni: automatic reverse engineering of input formats[C]//Proc of ACM Conference on Computer and Communications Security. 2008:391-402.

[17] LIN Zhi-qiang, JIANG Xu-xing, XU Dong-yan, *et al*. Automatic protocol format reverse engineering through context-aware monitored execution[C]//Proc of the 15th Symposium on Network and Distributed System Security. 2008.

[18] 吴志勇,王红川,孙乐昌,等. Fuzzing 技术综述[J]. 计算机应用研究,2010,27(3):829-832.

[19] BEDDOE M. Protocol information project[EB/OL]. (2004-10-05) http://www.4tphi.net/~awalters/PI/PI.html.

[20] 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.

[21] CUI Wei-dong, KANNAN J, WANG H J. Discoverer: automatic protocol reverse engineering from network traces[C]//Proc of the 16th USENIX Security Symposium. 2007:199-212.

[22] DEYOUNG M E. Dynamic protocol reverse engineering: a grammatical inference approach[D]. [S.l.]: Air Force Institute, 2008.

[23] SHEVERTALOV M, MANCORIDIS S. A reverse engineering tool for extracting protocols of networked applications[C]//Proc of the 14th Working Conference on Reverse Engineering. 2007:229-238.

[24] TRIFIL'O A, BURSCHKA S, BIERSACK E. Traffic to protocol reverse engineering[C]//Proc of the 2nd IEEE International Conference on Computational Intelligence for Security and Defense Applications. 2009:257-264.

[25] ANTUNES J, NEVES N. Building an automation towards reverse protocol engineering [EB/OL]. (2009-10)[2011-03]. http://homepages.di.fc.ul.pt/~nuno/PAPERS/INFORUM09.pdf.

[26] NEWSOME J, SONG D. Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software[C]//Proc of Network and Distributed System Security Symposium. 2005.

[27] JCABALLERO, YIN Heng, LIANG Zhen-kai, *et al*. Polyglot: automatic extraction of protocol format using dynamic binary analysis [C]//Proc of the 14th ACM Conference on Computer and Communications Security. 2007:317-329.

[28] 何永君,舒辉,熊小兵. 基于动态二进制分析的网络协议逆向解析[J]. 计算机工程,2010,36(9):268-270.

[29] WONDRACEK G, COMPARETTI M P, KRUEGEL C, *et al*. Automatic network protocol analysis[C]//Proc of the 16th Symposium on Network and Distributed System Security. 2008.

[30] WANG Zhi, JIANG Xu-xian, CUI Wei-dong, *et al*. ReFormat: automatic reverse engineering of encrypted messages[C]//Proc of European Symposium on Research in Computer Security. 2009:200-215.

[31] GOLD E M. Language identification in the limit[J]. Information and Control, 1967, 10(5):447-474.