Journal of Computer Applications
计算机应用,2015,35(5):1373 - 1378
ISSN 1001-9081
CODEN JYIIDU
2015-05-10
http://www.joca.cn

文章编号:1001-9081(2015)05-1373-06
doi:10.11772/j.issn.1001-9081.2015.05.1373

基于消息序列图的协议交互过程构建方法

石　旺[1,2*],杨英杰[1,2],唐慧林[1,2],董丽鹏[1,2]
(1.信息工程大学,郑州 450001;　2.河南省信息安全重点实验室,郑州 450001)
(*通信作者电子邮箱 shiwang1313@163.com)

摘　要:为了有效掌握协议的交互行为,提出一种基于消息序列图的协议交互过程自动构建方法。首先,根据协议交互过程的特点,定义依赖关系图来表示消息序列中事件的偏序关系,将网络流转换为依赖关系图;然后,使用基本消息序列描述协议的交互行为片段,通过定义事件最大后缀来挖掘基本消息序列;最后,搜索出最大依赖关系图并将其连接合并,构建出消息序列图。实验结果表明,该方法具有较高的准确性,构建出的消息序列图可以直观地表示协议的交互过程。

关键词:消息序列图;网络流;依赖关系图;事件最大后缀;协议交互过程
中图分类号:TP393.04　　**文献标志码**:A

Building protocol interactive process based on message sequence chart

SHI Wang[1,2*], YANG Yingjie[1,2], TANG Huilin[1,2], DONG Lipeng[1,2]
(1. *Information Engineering University, Zhengzhou Henan 450001, China;*
2. *Henan Province Key Laboratory of Information Security, Zhengzhou Henan 450001, China*)

Abstract: In order to effectively master protocol interactive behavior, a method to automatically build protocol interactive process based on message sequence chart was proposed. Firstly, according to the characteristics of the protocol interactive process, the dependency graph was defined to represent the partial order of events in message sequence, and the network flows were converted to dependency graphs. Secondly, the basic message sequences were used to describe protocol interactive behavior fragments, and the basic message sequences were mined by defining event maximum suffix. Finally, the maximum dependency graphs that were found out were connected and merged to build a message sequence chart. The experimental results show that the proposed method has a high accuracy and the built message sequence chart can visually represent the protocol interactive process.

Key words: message sequence chart; network flow; dependency graph; event maximum suffix; protocol interactive process

0　引言

协议交互过程是网络行为的具体体现,通常其严格按照协议规范执行实施。然而,实际网络中的协议交互过程并不都是严格按照协议规范进行的,如一些黑客对协议的漏洞进行攻击,导致网络中存在一些协议异常交互的情况[1]。同时,有些私有或专用协议的规范不公开,对其交互过程的分析也无法通过协议规范文档得到。事实上,针对如上非规范和未知协议的交互过程的分析,对于掌握网络运行状况有着关键性的支撑和重要价值。同时,协议交互过程的构建也是一些安全应用的基础,如渗透测试可以利用协议交互过程构建测试样本来发现协议漏洞,安全检测通过掌握协议的交互过程来发现隐蔽隧道和恶意攻击行为[2]。

通过对协议交互过程特点的分析研究,本文提出一种基于消息序列图的协议交互过程构建方法,其原理是通过对网络数据流进行分析,根据协议通信双方发送和接收的消息序列,利用消息序列图描述协议的交互动作及其以何种顺序发生,以图形或序列形式形象化地表示网络中消息的传递行为,从而可以很直观准确地掌握协议的交互过程。

1　相关研究

协议交互过程是协议的重要组成部分。目前,有关协议交互过程的研究工作主要分为两个方向[3-4]:一个方向是利用动态污点分析技术,对现网络协议的软件程序进行二进制的动态跟踪分析,通过跟踪二进制文件对报文的处理流程,以获取协议交互过程,但是该方法技术难度较高,在实际应用中难以获取协议的软件进行分析代码分析。另外一个方向是利用网络流来推测网络协议交互过程,主要通过分析通信双方之间的报文,以得到协议交换过程的相关信息;该方法以网络数据流为分析对象,容易获取且适用性强。因此,本文方法也是基于该原理实现协议交互过程的构建。

基于网络流对协议交互过程的研究起步较晚,相关研究主要有协议状态机推断[5-6]。目前,大部分针对协议状态机的推断方法采用人工分析来推断其交互过程,但也有少量自

收稿日期:2014-11-27;**修回日期**:2015-01-13。
基金项目:国家973计划项目(2011CB311801);河南省科技创新人才计划项目(114200510001)。
作者简介:石旺(1987 -),男,湖南湘潭人,助理工程师,硕士研究生,主要研究方向:网络与信息安全;　杨英杰(1971 -),男,河南郑州人,副教授,博士,主要研究方向:网络与信息安全;　唐慧林(1980 -),男,安徽安庆人,讲师,博士研究生,主要研究方向:信息安全;　董丽鹏(1987 -),男,山西阳泉人,工程师,硕士研究生,主要研究方向:网络与信息安全。

动推断的方法[6]。一些自动推断协议状态机的方法只是针对单方向报文状态的转换,分别展示客户端或服务器的行为,在推断出的状态机中主要是体现状态转换之间的约束关系,而不能较好地体现出协议通信双方的交互过程[7]。

消息序列图由国际电信联盟(International Telecommunication Union, ITU)提出,主要用于形式化描述消息之间的通信行为和传递顺序,适合直观描述协议实体进行通信时的消息交互过程[8]。因此,可以将双向报文信息加入到协议消息序列中,利用消息序列图对通信双方报文的交互过程进行描述,以消息来标识报文类型,用序列来表示交互的状态及其关联性,构建出协议交互过程的消息序列图。

2 协议交互过程分析

一个消息序列图表示一个具有偏序性质的事件集合,包括消息发送、消息接收。偏序关系可以反映协议事件的时序特征。在正常的协议交互过程中,通过获取协议消息的偏序关系可以准确掌握协议的交互过程。偏序具有两层含义[9]:第一层含义是在一个行为中事件的所有顺序关系,即事件随着时间总是从开始一步一步到结束;第二层含义是表示消息的发送与接收的关系,即发送事件总是在接收事件之前。

协议通过特定的消息语义来完成某项任务,语义由协议的消息类型来定义,消息按照一定的时序进行传输。协议语义关键字是指在协议中规定的一类具有特殊语义的字符或字符串[10]。语义关键词可以区分协议的不同消息类型,代表协议交互一方的一次请求或者响应。语义关键词的提取方法可以通过查阅标准规范或从网络流中利用相关技术提取,相关研究已有很多[10-11]。为了便于描述,用语义关键词 *keyword* 来标识消息的类型,从而协议的每个交互过程可表示为一个消息标识符序列。用 m 标识消息,1 或 -1 分别表示消息的发送和接收方向,则用户 p 向 q 发送一个消息 m_1 可以表示为〈1, $keyword_1$〉,用户 p 接收到 q 的一个消息 m_2 表示为〈-1, $keyword_2$〉。消息序列是一个将多个消息用箭头连接的图,客户端 p 与服务器 q 的一个简单协议交互过程及其对应的消息序列示例如图 1 所示,其中单向箭头表示消息之间的时序关系。

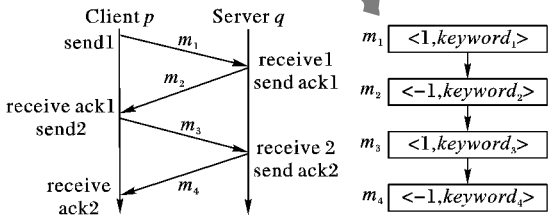

图 1 协议交互过程及其消息序列

通过以上分析,协议在网络中体现为一系列消息的传递过程,根据协议交互和消息序列图的特点,可以通过对协议的网络流进行收集分析,利用消息序列图来有效地描述协议交互过程,用语义关键字来代表某个消息类型,根据消息传递顺序得到消息的偏序关系,最后得到一个由消息组成的序列 $\{m_1, m_2, \cdots, m_n\}$。

协议实体发送消息或接收消息的过程表示为事件。事件可以形式化表示为:用户 p 向 q 发送一个消息 m_1 为一个发送事件,表示为(p,q,1,$keyword_1$);用户 p 接收到 q 的一个消息 m_2 为一个接收事件,表示为(p,q,-1,$keyword_2$)。其中,1 表示消息的发送,-1 表示消息的接收,语义关键词 *keyword* 表示消息 m 的类型。一个协议消息序列代表一组事件,包括消息发送和消息接收以及这些事件之间的偏序关系,以准确反映网络流中消息的时序特征。协议消息序列定义如下。

定义 1 协议消息序列。一个协议消息序列 M 可以看作是两个协议实体之间具有偏序关系的事件集合,表示为 $M = (L, \{E_l\}_{l \in L}, \leqslant, \delta, \Sigma)$,其中:$L$ 是一组由多个消息组成的路径集合;E_l 是一个路径 l 中的事件集合;Σ 是消息发送和接收事件的标识符集;$\delta: \{E_l\}_{l \in L} \rightarrow \Sigma$ 是给每个发送事件或接收事件分配一个标识符的函数;$\leqslant$ 表示 $\{E_l\}_{l \in L}$ 中事件出现的偏序关系,$\leqslant_l$ 表示 E_l 中事件按路径 l 依次排序,即沿着路径 l 依次从开始到结束。

协议交互行为通常是由多个动作形成的一个有序的消息传输过程。协议的一次执行路径就是协议在一个交互过程中经历的具有先后顺序关系的消息序列。本文基于网络流来构建消息序列,一个消息序列图可表示为一个有向图 (V, E, V_s, V_f),其中 V 是顶点集,E 是边的集合,V_s 是开始顶点,V_f 是结束顶点。消息序列图中的一个消息序列可表示为 $(v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n)$,其中,$v_1 \in V_s$ 并且 $v_n \in V_f$。

3 协议交互过程构建方法

协议会话是协议实体之间一系列消息的交互过程,从开始到结束的各个阶段都有特定的消息类型。本文所指的协议会话为传输层会话,即认为一个传输层会话是一个完整的协议会话。协议交互过程的构建就是通过对协议交互行为的网络流进行收集处理,提取出目标协议会话,通过标识消息类型,根据消息之间的偏序关系,自动生成一个消息序列图来对其进行描述。这个过程中主要面临的挑战在于,如何从网络流中挖掘协议交互行为,并用消息序列对其进行描述。本文采取以下方法,使用依赖关系图来表示消息序列中事件的偏序关系,并为给定的网络流引入事件最大后缀和最大依赖关系图,以获取基本消息序列来构建消息序列图。协议交互过程构建的整个过程分为 3 个阶段,具体步骤如图 2 所示。

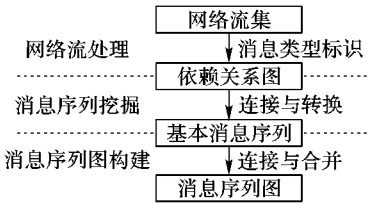

图 2 协议消息序列图构建步骤

1)网络流处理。收集目标协议的网络流,提取出目标协议会话集,利用协议语义关键词识别出协议会话中不同的消息类型,根据消息间的时序约束关系,将网络流转变成依赖关系图。

2)基本消息序列挖掘。从依赖关系图确定基本的消息序列,用最大依赖关系图表示,通过一系列算法将每一个依赖关系图转变成一个基本消息序列。

3)消息序列图构建。将所有的基本消息序列组合连接,合并到一个消息序列图中。

3.1 网络流处理

网络流通过网络数据采集工具采集,包括协议双方所有发送和接收的消息。本文将一个传输层会话作为一个完整的协议会话,从而根据传输层会话的特点从网络流中提取出目

标协议的会话集。传输层会话包括传输控制协议(Transmission Control Protocol, TCP)会话和用户数据报协议(User Datagram Protocol, UDP)会话。TCP 会话是指一个 TCP 连接从开始到结束的数据包集合; UDP 会话是指将发送的第一个数据包作为会话的开始,如果会话中等待 60 s 后没有出现的新数据包,则将最后收到的一个数据包作为会话的结束[12]。如果同时存在多个协议会话,通过提取传输层会话可以将多个协议会话区分开来,排除不同协议会话对事件偏序关系的影响。

网络流是一连串事件的有限集合,包括发送事件和接收事件,用符号 Σ 表示。一个网络流事件为发送 m 或接收 m,每一个事件必须包含一个时间戳,以记录事件发生的线性时间顺序。通过消息类型标识的各个事件是互不相干的,可以利用事件之间的时序关系将它们联系起来; 然后,将网络流得到的事件顺序转换成偏序关系来表示协议的整个行为。

一个协议消息序列 $M = (L, \{E_l\}_{l\in L}, \leqslant, \delta, \Sigma)$ 规定了一个事件集的偏序 $\leqslant$, $\leqslant$ 被定义为每条路径的事件之间的顺序关系。观察到协议的有些行为频繁重复地出现在交互过程中,并且符合 $\leqslant_l$ 规定的顺序。客户端和服务器的协议会话是一个交互进行的过程。客户端给服务器发送一条命令请求,服务器返回一个相应的响应。在协议会话的网络流中,这些发送命令请求和返回响应是一起出现的,即存在一个相互依赖的关系。因此,定义一个依赖关系图来获取这些行为,以表示交互过程中消息发送与接收之间的偏序关系。

定义 2　依赖关系图(Dependency Graph, DG)。一个依赖关系图可以定义为 $DG = (L, \{V_l\}_{l\in L}, R, \delta', \Sigma)$,其中,每个 $v_i \in V_l$ 对应于一个事件 $e_i \in E_l$; 存在一条有向边 v_1Rv_2,当且仅当其对应的事件 e_1 和 e_2,$(e_1 \rightarrow e_2)$ 满足偏序关系; $\delta'(v_i) = \delta(e_i)$ 为依赖关系图中每一个事件 e_i 及其对应的顶点 v_i。

由于路径 L 和事件符号 Σ 与后续分析不相关,可以使用 (V,R,δ) 作为依赖关系图的一个简化表示。依赖关系图作为网络流的图形表示,其相关属性定义如下。

定义 3　等效 $\equiv$。对于依赖关系图 $DG_1 = (V_1, R_1, \delta_1)$ 和 $DG_2 = (V_2, R_2, \delta_2)$,$DG_1 \equiv DG_2$ 当且仅当存在一个映射 f: $V_1 \rightarrow V_2$,其中: $\forall v_1 \in V_1, \delta_1(v_1) = \delta_2(f(v_2))$; $\forall v_1, v_2 \in V_1$, $v_1R_1v_2 \Leftrightarrow f(v_1)R_2f(v_2)$。

定义 4　依赖关系图的连接 $|$。对于两个图 $DG_1 = (V_1, R_1, \delta_1)$ 和 $DG_2 = (V_2, R_2, \delta_2)$ 的连接,$DG_1 \mid DG_2 = (V, R, \delta)$,其中; $V = V_1 \cup V_2$; $\delta = \delta_1 \cup \delta_2$; $R = R_1 \cup R_2$。

图 3 显示出的 $DG_1 \mid DG_2$ 为依赖关系图 DG_1 和 DG_2 连接的结果。

定义 5　子图。依赖关系图中的子图定义为: $DG' \subseteq DG$,当且仅当存在图 x 和 y 使得 $DG \equiv (x \mid DG') \mid y$。

定义 6　前缀和后缀。一个子图 $DG' \subseteq DG$ 是一个前缀,当且仅当存在 y,$DG \equiv DG' \mid y$。类似地,DG' 是一个后缀,当且仅当存在 x,$DG \equiv x \mid DG'$。

在图 3 中,DG_a,DG_b 和 DG_c 是依赖关系图连接图 $DG_1 \mid DG_2$ 中的子图,其中子图 DG_a 是一个前缀,DG_c 是一个后缀。

对于网络流 t,根据协议交互过程的特点,定义一个函数 $d(t)$ 来构造一个依赖关系图。构造依赖关系图时,首先,为一个事件的每次出现创建一个唯一的顶点,即不同的事件采用不同的顶点; 然后,添加有向边将满足时序关系的相邻事件连接成一个路径; 最后,将发送和接收的事件以向后扩展的方式连接到路径后面。利用依赖关系图构造函数 $d(t)$,将可用的网络流集合转变成依赖关系图集合,其中每个依赖关系图对应协议执行的情况。在函数执行过程中,为了保证构建序列的准确性,作以下两个假设:1)协议会话传输过程中没有消息丢失;2)先发送的消息先到达,传输过程中不会造成消息的乱序。

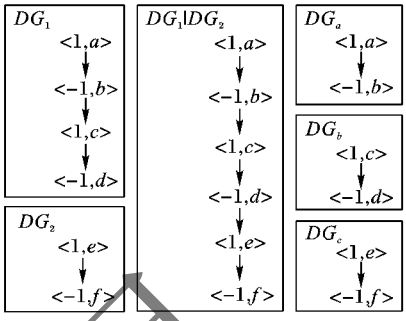

图 3　依赖关系图的连接

3.2　基本消息序列挖掘

基本消息序列用来描述协议会话过程中的交互行为片段,可以反映各消息之间的时序关系和连接依赖关系。一个消息序列就是协议实体双方进行一次交互行为所发送或接收的消息序列,节点表示消息事件,有向边表示事件之间的连接依赖关系。在同一协议的不同交互过程中,其基本消息序列是不同的。本节的目标是要发现尽可能全且完整的基本消息序列。因此,为每一个事件定义一个事件最大后缀来表示基本消息序列。

定义 7　事件最大后缀。对于事件 $e \in \Sigma$,当给定一个网络流 $t \subseteq T$,其最大后缀 $max\_suffix(e)$ 是一个由事件 e 开始的依赖关系图,该最大后缀包含一个标记为 e 的顶点 v_e。除了该顶点外,它也包含了与 e 相连并在 e 之后发生的具有偏序关系的所有事件。

事件最大后缀是 v_e 的所有后缀的最大图,具有以下两个性质:1) $max\_suffix(e)$ 包含至少一个顶点 v_e;2) $max\_suffix(e)$ 不能往后继续扩展。

事件最大后缀在网络流中体现为由 e 开始到事件结束的整个消息序列。对于事件 e 和网络流集合 T,T_e 是由 e 开始的网络流后缀。T_e 可以很容易地衍生出一个后缀树。通过对每一个 $t_i \in T_e$ 确定 $d(t_i)$,可以从 T_e 得到后缀图形的集合。在该图中,让 v_e 对应于事件 e 第一次发生的顶点。然后根据条件去除不属于最大后缀的顶点。在遍历中,顶点和边逐渐添加到最大后缀中,直到所有满足条件的顶点和边都添加完毕。事件最大后缀的挖掘算法描述如算法 1 所示。

算法 1　挖掘事件最大后缀。

输入　网络流集合 T,T 中出现的事件集 Σ;

输出　$max\_suffix(e)$。

```
1)  for each e ∈ Σ do
2)      get T_e;                          // T_e 是由 e 开始的所有后缀集
3)      T_e = {t_1, t_2, ···, t_n};
4)      max_suffix(e) = ∅;                //max_suffix 初始为空
5)      for i = 1···n do
6)          (V,R,δ) = d(t_i);             // 将 t_i 转化为依赖关系图
7)          v_e = first(e);               // v_e 对应于第一个事件 e
8)          for each v ∈ V
```

```
9)      if (v_e → v) ∉ R then
10)       V = V - {v};          // 去除不属于最大后缀的顶点
11)     end for
12)     if max_suffix(e) = ∅ then
13)       max_suffix(e) = (V,R,δ); // 初次得到 max_suffix
14)     else
15)      max_suffix(e) = max_suffix(e) | (V,R,δ);
        //将满足条件的顶点和边添加到最大后缀中
16)   end for
17) end for
```

3.3 消息序列图构建

3.3.1 搜索最大依赖关系图

随着协议交互次数的增加和交互过程的变化,每一个事件都有其最大后缀,通过网络流搜索出的事件最大后缀模式数量会迅速增加。事件最大后缀只能表示从某一事件 e 开始的消息序列,并不一定能表示协议一次交互的完整过程。因此,引入最大依赖关系图,对事件最大后缀进行连接合并。

定义 8 最大依赖关系图。对于一个给定的网络流集合 $T = \{t_1, t_2, \cdots, t_n\}$,最大依赖关系图表示为 $DG_{\max} = (V, R, \delta)$,当且仅当:

1)存在 $t \in T$ 使得 $DG_{\max} \subseteq d(t)$;

2)对于不同的 $v_1, v_2 \in V, (v_1 \rightarrow v_2) \in R$;

3)不存在满足以上 2 个条件的 DG',使得 $DG_{\max} \subseteq DG'$。

条件 2)要求 $DG_{\max}$ 中的事件是相互连接的,即 $DG_{\max}$ 中的消息属于同一个协议交互行为。条件 3)保证了 $DG_{\max}$ 的最大性,即表示该 $DG_{\max}$ 是一个完整的消息序列。这些附加的条件可用于简化搜索过程。

事件最大后缀连接合并的一个基本思想是如果事件最大后缀中某一事件 e 的前缀与另一事件最大后缀中事件 e 的部分前缀相同,则将两个最大后缀连接合并。

由于最大后缀已经是最大的,不能在后面扩展。因此,通过给其加前缀来扩展 DG_1。对于每个事件 e',判断 $max\_suffix[e']$ 是否可以合并到 DG_1,将 $max\_suffix[e']$ 作为 DG_2。不失一般性,将两个最大后缀表示为:$DG_1 \equiv DG^{comm} \mid DG_1^{suff}$,$DG_2 \equiv (DG_2^{pref} \mid DG^{comm}) \mid DG_2^{suff}$。其中,$DG^{comm}$ 是这两个依赖关系图中共有的。如果 DG^{comm} 是空的,则不进行任何合并。如果 DG^{comm} 非空,将 $DG_2^{pref} \mid DG^{comm} \mid DG_1^{suff} \mid DG_2^{suff}$ 作为合并的图。如果没有更多的事件最大后缀可以合并到依赖关系图 DG_1 中,则它是一个 $DG_{\max}$。最大依赖关系图的搜索算法描述如算法 2 所示。

算法 2 搜索最大依赖关系图。

输入 $max\_suffix(e)$ for all $e \in \Sigma$;

输出 $DG_{\max}(e_1)$。

```
1)  for each e_1 ∈ Σ do
2)    Q = Σ - {e_1};                    // 去除事件 e_1
3)    DG_1 = max_suffix(e_1);
4)    while ∃e_2 ∈ Q do
5)      DG_2 = max_suffix(e_2) = (DG_2^pref | DG^comm) | DG_2^suff;
6)      if DG^comm = ∅ then
7)        return fail          //不满足连接合并条件,返回失败
8)      else
9)        if DG_2^pref = ∅ then
10)         return fail        //不满足连接合并条件,返回失败
11)       else
12)         DG_1 = merge(max_suffix(e_2), DG_1) ≡
              DG_2^pref | DG^comm | DG_1^suff | DG_2^suff;
            // 将满足条件的事件最大后缀连接合并
13)         Q = Q - {e_2};                  // 去除事件 e_2
14)       end if
15)     end if
16)   end while
17)   DG_max(e_1) = DG_1;
18) end for
```

3.3.2 合并成消息序列图

一个协议在一段时间内会进行多种不同的协议交互行为,协议的每一个交互行为对应一个最大依赖关系图。最大依赖关系图作为消息序列图的基本组成部分,根据网络流中消息之间的约束关系,利用自动学习技术将其连接起来,形成消息序列图,为了掌握网络中某个协议一段时间的交互过程,需要将同一协议的不同交互行为得到的不同消息序列连接合并,形成一个协议消息序列图,即一个由不同消息类型组成的有向图,以描述一段时间内某个协议的全部交互行为。

为了表示一个完整的协议消息序列图,在后续步骤中将具有相连约束关系的最大依赖关系图连接起来。对于一个协议,在进行不同的交互行为时,会话开始阶段会有相同的认证或握手动作,在消息序列中即体现为相同的前缀。最大依赖关系图连接合并的一个基本思想是如果 DG_{max}' 中事件 e 的前缀与 DG_{max} 中事件 e 的前缀相同,则将 DG_{max}' 中事件 e 的后缀连接到 DG_{max} 中。将所有的最大依赖关系图连接合并后,即形成协议交互过程的消息序列图。具体算法如算法 3 所示,输入最大依赖关系图序列 DG_{max}_List,将具有约束关系的最大依赖关系图连接合并后,输出消息序列图 MSC。

算法 3 生成消息序列图。

输入 DG_{max}_List;

输出 MSC。

```
1)  MSC = DG_max[0];                  // 初始化 MSC
2)  temp = ∅;
3)  for i = 1,2,…,DG_max_List.size() do
4)    DG_max[0] ≡ DG_max0^pref | DG_max0^suff;
5)    temp = DG_max[i] ≡ DG_maxi^pref | DG_maxi^suff;
6)    if DG_maxi^pref = DG_max0^pref then   //存在相同的前缀
7)      MSC = MSC | temp;          //将 DG_max 合并到 MSC 中
8)      temp = DG_max[i+1];
9)    end if
10) end for
11) return MSC
```

4 实验及结果分析

为了验证本文方法的有效性,选取常用的文件传输协议(File Transfer Protocol, FTP)和简单邮件传输协议(Simple Mail Transfer Protocol, SMTP),利用 Serv-U 和 WebMail 配置了 FTP 和 SMTP 服务器,搭建了如图 4 的实验环境。利用用户终端对服务器进行了一系列的协议操作,模拟产生了大量的协议网络流数据集。

实验过程主要分为两步:首先,利用 Wireshark 工具采集目标协议的网络流数据集,对网络流进行预处理,根据语义关键字识别出不同的消息类型,并将网络流转变成依赖关系图;然后,根据本文提出的消息序列图构建方法,构建出协议的消息序列图,并验证其准确性。

4.1 网络流处理

为了能有效识别出协议的消息类型,提取到的语义关键

词应尽可能覆盖协议所有的消息类型。本文提取出 FTP 的部分语义关键词为 USER、PASS、RETR、PORT、PASV、LIST、QUIT、200、220、226、230、250、331、332、530、500 等。SMTP 的部分语义关键词为 EHLO、AUTH LOGIN、MAIL FROM、RCPT TO、DATA、QUIT、220、221、235、250 等。

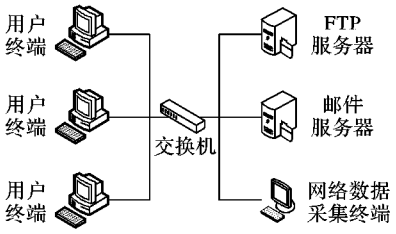

图 4　实验环境

在采集到目标协议的网络流数据集后,过滤掉非目标协议的数据,提取出网络流中的目标协议会话集。本文根据传输层会话特点提取出 FTP 和 SMTP 的协议会话数量各 100 个。利用提取出的语义关键词,识别出网络流中的消息类型,用消息方向和语义关键词标识消息类型,根据消息之间的偏序关系,构建出目标协议的依赖关系图。FTP 的部分依赖关系图示例如图 5 所示。图 5 中,DG_1 表示 FTP 会话开始时的认证过程,DG_2 表示 FTP 的获得文件操作,DG_3 表示 FTP 的会话结束过程。

4.2　协议消息序列图构建

为了验证本文方法的有效性,本文通过人为正常操作生成有限的网络流样本来进行实验。FTP 的部分操作动作及消息序列如表 1 所示。

表 1　FTP 操作动作及消息序列

网络流	操作动作	消息序列
t_1	登录、获得文件、退出	220、USER、331、PASS、230、RETR、150、226、QUIT、221
t_2	登录、系统、改变工作目录、等待、列表、退出	220、USER、331、PASS、230、SYST、215、CWD、250、PASV、227、LIST、150、226、QUIT、221
t_3	登录、数据端口、退出	220、USER、331、PASS、230、PORT、200、QUIT、221

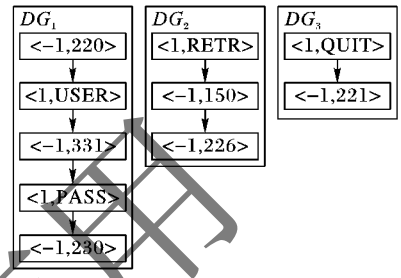

图 5　FTP 部分依赖关系图

根据本文方法对表 1 中的 3 个网络流构建出 FTP 的消息序列图如图 6 所示。SMTP 的交互过程比较简单,利用本文方法对邮件的一次发送过程构建出 SMTP 的消息序列图如图 7 所示。

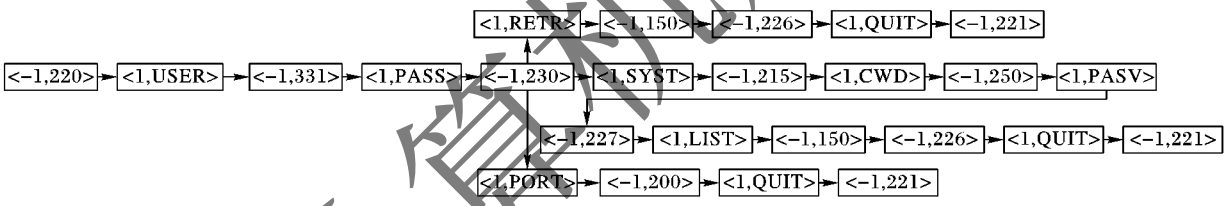

图 6　FTP 消息序列图

图 7　SMTP 消息序列图

从图 6 和图 7 可以看出,构建出的消息序列图比协议状态图更容易理解,可以直观地表示出网络流中协议实体双方进行会话的所有交互过程。

为了验证协议消息序列图构建的准确性,使用准确率来进行评估。由于本文的目的是构建出协议在一段时间内的交互过程,因此构建出的消息序列图并不是描述完整的协议规范。利用 RFC 文档推断出符合协议规范的协议状态机,将其作为参考标准,用构建出的消息序列图与之对比:如果构建出的协议消息序列图符合协议状态机,则认为正确;否则认为不正确。因此,准确率 R 的计算公式如下:

$$R = P/N \times 100\%$$

其中:P 为构建出的消息序列图正确的数量,N 为构建出的消息序列图的总数量。图 8 给出了本文方法与文献[13]方法的准确率对比结果。

通过实验结果分析可知,本文方法的准确率基本都在 90% 以上,高于文献[13]的方法。文献[13]的方法在具有大量训练集的情况下,推断协议状态机时准确率较高,当训练集比较小时准确率有所降低,因此,不适合描述协议在一段时间内的交互过程。本文方法主要考虑获取协议交互过程中消息之间的偏序关系,从而能准确反映协议的实际交互过程。同时,由于实验是在实验室环境下进行的,采集的数据都是人工进行的正常操作生成的符合协议规范的网络流,且没有其他协议和恶意攻击的干扰,因此得到的结果准确率比较高。

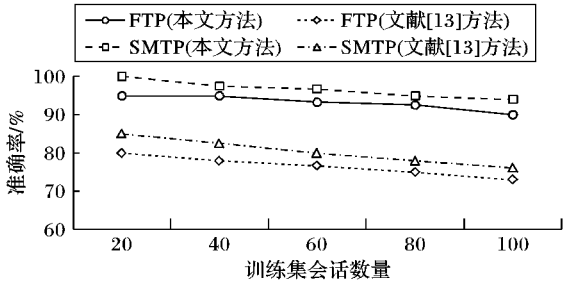

图 8　消息序列图构建准确率比较

5　结语

对网络中的协议交互过程进行构建,可以掌握协议的具体行为,对于协议分析、网络管理和安全检测等具有重要意

义。本文关注于构建网络流中协议的交互过程,基于消息序列图的思想,利用消息序列挖掘算法,根据消息之间的约束关系,构建出协议的消息序列图,能直观准确地表示出协议的实际交互过程,在协议分析方面具有很好的应用价值。在下一步工作中,为更细致地描述协议的交互过程,考虑为消息序列图的有向边添加权值,以表示消息序列出现的次数,同时将研究如何基于本文的方法对协议的异常行为进行检测。

参考文献:

[1] ZHU J, GUO C, WU Q. Conformance checking method for Web service interaction behaviors based on Petri nets[J]. Computer Engineering and Science, 2013, 35(1): 24 - 25. (朱俊, 郭长国, 吴泉源. 基于 Petri 网的 Web 服务交互行为一致性检测方法[J]. 计算机工程与科学, 2013, 35(1): 24 - 25.)

[2] HOOPER E, HOLLOWAY R. Intelligent techniques for effective network protocol security monitoring, measurement and prediction [J]. International Journal of Security and Its Applications, 2008, 2 (4): 1 - 10.

[3] PAN F, WU L, DU Y, *et al.* Overviews on protocol reverse engineering[J]. Application Research of Computers, 2011, 28(8): 2801 - 2806. (潘璠, 吴礼发, 杜有翔, 等. 协议逆向工程研究进展[J]. 计算机应用研究, 2011, 28(8): 2801 - 2806.)

[4] DAINOTTI A, PESCAPÉA, CLAFFY K C. Issues and future directions in traffic classification[J]. IEEE Network, 2012, 26(1): 35 - 40.

[5] ZHANG Z, WEN Q, TANG W. Survey of mining protocol specifications[J]. Computer Engineering and Applications, 2013, 49(9): 1 - 9. (张钊, 温巧燕, 唐文. 协议规范挖掘研究综述[J]. 计算机工程与应用, 2013, 49(9): 1 - 9.)

[6] LI X, LI C. A survey on methods of automatic protocol reverse engineering[C]// Proceedings of the 7th International Conference on Computational Intelligence and Security. Piscataway: IEEE, 2011: 685 - 689.

[7] WANG Y, ZHANG Z, YAO D, *et al.* Inferring protocol state machine from network traces: a probabilistic approach[C]// Proceedings of the 2011 International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2011: 1 - 18.

[8] ITU-T. Message sequence chart (MSC)[S]. Geneva: ITU-Telecommunication Standardization Sector, 1999.

[9] DONG G, PEI J. Mining partial orders from sequences[J]. Sequence Data Mining, 2007, 33(10): 89 - 112.

[10] WANG Y, YUN X, SHAFIQ M Z, *et al.* A semantics aware approach to automated reverse engineering unknown protocols[C]// Proceedings of the 20th IEEE International Conference on Network Protocols. Piscataway: IEEE, 2012: 1 - 10.

[11] HUANG X, CHEN X, ZHU N, *et al.* Protocol state machine reverse method based on labeling state[J]. Journal of Computer Applications, 2013, 33(12): 3486 - 3489. (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.)

[12] LI N. Application protocol identification based on flow statistics [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2013: 10 - 11. (李宁. 基于流统计特性的应用协议识别技术研究[D]. 南京: 南京邮电大学, 2013: 10 - 11.)

[13] ANTUNES J, NEVES N, VERISSIMO P. Reverse engineering of protocols from network traces[C]// Proceedings of the 18th Working Conference on Reverse Engineering. Piscataway: IEEE, 2011: 17 - 20.

(上接第1327页)

[4] JANNACH D, ZANKER M, FELFERNING A, *et al.* Recommender systems: an introduction [M]. New York: Cambridge University Press, 2011: 13 - 22.

[5] WANG B. Mining of massive datasets [M]. Beijing: Post and Telecom Press, 2012: 115 - 143. (王斌. 大数据: 互联网大规模数据挖掘与分布式处理[M]. 北京: 人民邮电出版社, 2012: 115 - 143.)

[6] XIANG L, CHEN Y, WANG Y. Recommendation system practice [M]. Beijing: Posts and Telecom Press, 2012: 73 - 77. (项亮, 陈义, 王益. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012: 73 - 77.)

[7] XING C, GAO F, ZHAN S, *et al* . A collaborative filtering recommendation algorithm incorporated with user interest change [J]. Journal of Computer Research and Development, 2007, 44 (2): 296 - 301. (邢春晓, 高凤荣, 战思南, 等. 适应用户兴趣变化的协同过滤推荐算法[J]. 计算机研究与发展, 2007, 44 (2): 296 - 301.)

[8] ZHENG X, CAO X. Research on lineal gradual forgetting collaborative filtering algorithm[J]. Computer Engineering, 2007, 33(6): 72 - 73. (郑先荣, 曹先彬. 线性逐步遗忘协同过滤算法的研究[J]. 计算机工程, 2007, 33(6): 72 - 73.)

[9] WANG L, ZHAI Z. Collaborative filtering algorithm based on time weight[J]. Journal of Computer Applications, 2007, 27(8): 2302 - 2303. (王岚, 翟正军. 基于时间加权的协同过滤算法[J]. 计算机应用, 2007, 27(8): 2302 - 2303.)

[10] GORI M, PUCCI A. ItemRank: a random-walk based scoring algorithm for recommender engines[C]// Proceedings of the 20th International Joint Conference on Artifical Intelligence. San Francisco: Morgan Kaufmann Publishers, 2007: 2766 - 2771.

[11] SHANG S, KULKANMI S, CUFF P, *et al.* A random walk based model incorporating social information for recommendations[C]// MLSP2012: Proceedings of the 2012 IEEE International Workshop on Machine Learning for Signal Processing. Piscataway: IEEE, 2012: 23 - 26.

[12] GUO Y, BAI S, YANG Z, *et al.* Analyzing scale of Web logs and mining users' interests[J]. Chinese Journal of Computers, 2005, 28(9): 1483 - 1495. (郭岩, 白硕, 杨志峰, 等. 网络日志规模分析和用户兴趣挖掘[J]. 计算机学报, 2005, 28(9): 1483 - 1495.)

[13] HUANG X . Cognitive psychology [M]. Beijing : China Light Industry Press, 2000: 70 - 80. (黄希庭. 认知心理学[M]. 北京: 中国轻工业出版社, 2000: 70 - 80.)

[14] Recommendation algorithm contest of TianChi big data[EB/OL]. [2014 - 06 - 20]. http://102. alibaba. com/competition/addDiscovery/index. htm. (天池大数据推荐算法比赛[EB/OL]. [2014 - 06 - 20]. http://102. alibaba. com/competition/addDiscovery/index. htm.)