计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(11):132−138 [doi: 10.15888/j.cnki.csa.006063]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

面向高速混杂网络的被动式多维度主机指纹模型①

张凯翔[1], 刘　琰[1], 方文渊[1], 刘莺迎[2]

[1](数学工程与先进计算国家重点实验室, 郑州 450000)
[2](河南牧业经济学院 信息与电子工程学院, 郑州 450000)
通讯作者: 刘　琰, E-mail: ms_liuyan@aliyun.com

摘　要: 主机识别对于计算机网络犯罪取证、抵御匿名攻击具有重要意义. 为了精确识别网络上的目标主机, 首先给出了多维度主机指纹模型的定义和性质并进行了形式化描述, 然后针对传统方法在主机指纹获取中存在可靠性及准确性不足的问题, 综合主机硬件特征信息、主机软件环境特征信息和主机网络行为特征信息, 提出了一种面向高速混杂网络流量的多维度主机指纹模型构建方法. 实验结果表明, 该模型在高速混杂网络下可以灵活有效提取主机特征信息, 使用该模型构建多维度主机指纹模型, 主机识别准确率达到 93.33%, 相比单维度主机指纹识别提高了近 8 个百分点, 具有更高的可靠性和准确率, 且不受 IP 地址变化的影响.

关键词: 主机识别; 主机指纹; 特征提取; 特征选择

引用格式: 张凯翔,刘琰,方文渊,刘莺迎.面向高速混杂网络的被动式多维度主机指纹模型.计算机系统应用,2017,26(11):132–138. http://www.c-s-a.org.cn/1003-3254/6063.html

Passive Multi-Dimensional Host Fingerprint Model in High-Speed Hybrid Network

ZHANG Kai-Xiang[1], LIU Yan[1], FANG Wen-Yuan[1], LIU Ying-Ying[2]

[1](China State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450000, China)
[2](Information and Electronic Engineering of Henan Animal Husbandry College, Zhengzhou 450000, China)

Abstract: Host identification is very important for computer forensics and anonymous attack resistance. In order to accurately identify the target host on the network, the definition and properties of the multi-dimensional host fingerprint model are given and formalized. Then, in view of the problem of reliability and accuracy of fingerprint acquisition, this paper proposes a multi-dimensional host fingerprint model for high-speed hybrid network traffic, which integrates the hardware characteristic information, host software environment characteristic information and host network behavior characteristic information. The experimental results show that the proposed model can extract data flexibly and efficiently in the high-speed hybrid network, and the multi-dimensional host fingerprint model can effectively identify the host with the accuracy of 93.33% , which has increased by nearly 8 percent compared with the single-dimension host fingerprint identification, and the multi-dimensional host fingerprint model is not affected by IP address changes. In general, the multi-dimensional host fingerprint model has higher reliability and accuracy compared with the single-dimensional host fingerprint identification.

Key words: host identification; host fingerprint; characteristics extraction; characteristics selection

随着互联网的飞速发展, 人们的社会活动越来越倾向于网络化, 网络成为了传播、存储和交互各种信息的新平台. 然而随之而来的大量安全隐患也开始不断出现. 网络攻击者和窃密者往往会实施网络伪装以逃避追查, 由于数据报报头在网络中传输时的透明性, 上网主机可以修改自身的网络标识, 因此依赖 IP 地

① 基金项目: 国家自然科学基金 (61309007, U1636219); 国家重点研发计划课题 (2016YFB0801303)
收稿时间: 2017-02-16; 修改时间: 2017-03-16; 采用时间: 2017-03-23

址、MAC 地址等手段已不能确切标识网络主机. 然而识别主机对于计算机网络犯罪取证、抵御匿名攻击等具有重要意义.

现有的方法多依据单维度的主机特征信息进行主机识别, 如主机硬件特征信息、主机软件环境特征信息等. T.Kohno 等人[1]最早提出了通过主机通信报文中携带的时间戳信息测算主机时钟偏移率作为主机硬件特征识别主机的方法, 并对主机时钟偏移率的稳定性、可测性、可区分性进行了测量. Polčák L 等人[2,3]分析了影响时间戳和主机时钟偏移率的原因和解决办法. 隋新[4]实现了一种基于被动嗅探获取 Banner 信息从而构建主机特征信息库来识别主机的方法. P.Eckersley 等人[5]通过请求客户端主机软件环境特征信息, 如主机操作系统类型及版本号、浏览器类型及版本号、主机安装的应用程序、字体等构建主机指纹信息, 进而识别主机. A.Soltani 等人[6]提出一种基于长存 cookies 的方法对主机进行识别, 该方法通过在不同的位置使用不同的格式存储多分 cookies 的副本, 从而保证识别主机的准确率.

然而, 在现有的研究中基于主机硬件特征的主机识别虽然具有较高的识别准确率且能够识别出 NAT 后的主机, 但主机时钟偏移率很难生成, 由于精度的影响无法进行大规模部署. 基于主机软件环境特征的主机识别主要是通过主动的方式获取主机软件环境特征信息, 如使用 Nmap[7]、Xprobe[8]、p0f[9]等工具或采用 cookies 的方式[10], 而主动探测的方式往往会受到网络过滤设备、主机防火墙和主机安全配置的影响, 且探测到大多是普适性的主机软件环境特征, 无法直接用于主机识别.

为此, 本文提出了一种基于可扩展插件被动式获取主机特征信息的方法, 并给出了多维度主机指纹模型的构建方法. 实验证明, 使用多维度主机指纹库识别主机准确率达到 93.33%.

1 主机指纹模型

本节给出多维度主机指纹模型构建所涉及到的概念定义及形式化的描述.

定义 1. 主机特征信息: 指可以通过网络获取且对主机有识别作用的信息集合.

从国内外相关研究来看, 主机特征信息主要包含三类: 主机硬件特征信息、主机软件环境特征信息和主机网络行为特征信息. 为了便于理解, 分别定义如下:

定义 2. 主机硬件特征: 指主机硬件本身特有的或基于硬件信息提炼的特征信息集合, 用 S 表示. 如 MAC 地址、主机硬盘序列号、手机 IMEI 串号、主机时钟偏移率等.

定义 3. 主机软件环境特征: 指主机使用的软件环境及安装的应用程序信息集合, 用 O 表示. 如操作系统类型及版本号、浏览器类型及版本号、主机安装的应用程序及版本号等.

定义 4. 主机网络行为特征: 指主机访问网络的行为习惯所反映出的主机特征信息集合, 用 B 表示. 如主机访问应用程序使用的账号信息等.

定义 5. 主机指纹信息: 指可以唯一标识主机的主机特征信息集合, 用 c 表示, $c=\{s, o, b\}$, 其中 s、o、b 是 S、O、B 的一个子集.

由此给出多维度主机指纹模型定义.

定义 6. 多维度主机指纹模型: 假设 $P=\{p_1, p_2, p_3, \ldots, p_n\}$ 代表一台主机含有特征信息的数据包集合; $C=\{\{c_1\},\{c_2\},\{c_3\},\ldots\{c_n\}\}$ 表示主机指纹信息, F 代表主机指纹获取函数, 则 $\boldsymbol{F}:\boldsymbol{P}\rightarrow\boldsymbol{C}$, 若满足以下性质:

可测性: $\forall \boldsymbol{p}\in\boldsymbol{P}$, 存在 $\boldsymbol{c}\in\boldsymbol{C}$, 使得 $\boldsymbol{F}(\boldsymbol{p})=\boldsymbol{c}$. 说明主机特征集合是可以测量的.

可区分性: 对于任意两台主机数据包 P_a 和 P_b, 存在主机特征集合 $C_a\neq C_b$, 使得 $F(P_a)=C_a, F(P_b)=C_b$.

则这样的 C 可作为主机的指纹.

本文通过被动监听的方式获取网络数据流量, 由于网络数据流量具有传输速度快、流量混杂、数据格式不统一等特性, 被动式多维度主机指纹模型构建面临两个关键问题:

(1) 高速混杂流量中主机特征信息提取问题. 在高速混杂流量中, 高效处理数据流量是系统可靠性的基础, 由于不同类型的主机特征信息的提取没有统一的提取方法, 相同类型的特征信息也有多种识别特征, 如何设计一种通用可扩展的主机特征提取框架实现高效准确的特征提取是本文面临的一个关键问题. 第三节提出了基于可扩展插件在高速混杂流量下不同维度主机特征信息高效、并行处理的方法.

(2) 高维主机特征选择问题. 主机的特征信息具有多样性, 对于一台主机来说, 由于安装软件环境和网络行为的不同均会导致出现不同格式重复的特征信息, 而应用协议的规定也会导致不同的主机下出现大量相同的特征信息, 这都给主机指纹模型的构建增加了难度. 因此, 如何针对不同维度的主机特征信息构建主机

指纹提取函数 F 是本文面临的一个关键问题. 由此, 第四节提出了 MAP-SCORE 算法构建多维度主机指纹模型的方法.

2 高速混杂流量下主机特征信息提取

由于对主机硬件指纹的提取国内外已经有较为成熟的方法, 如 Polčák 等[11]人开发了一款主机硬件指纹工具, 该工具通过获取主机网络通信流量中携带的时间戳信息, 然后通过最小二乘法计算得到主机时钟偏移率. 因此, 主机硬件指纹提取不作为本文研究的重点. 在网络环境下, 主机与网络应用程序进行信息交互的过程中, 服务器端往往会请求主机操作系统类型及版本、浏览器类型及版本、系统语言、字体、分辨率等软件环境信息, 从而为主机提供最佳的服务, 同时应用协议也规定了主机与服务器间的通信行为, 为了能够分析用户访问应用程序的行为习惯和偏好, 应用程序往往采用 cookies 对用户的行为进行记录, 研究发现在主机进行网络通信过程中会频繁出现主机访问应用程序所使用到的账号信息, 但是由于不同的应用具有不同的应用协议结构和识别特征, 随着应用数量的增加, 获取满足定义的主机特征信息的难度也随着增加. 为了能够高效、并行地对不同维度的特征信息进行提取, 本节提出了基于插件的可扩展主机特征提取方法.

图 1 给出了高速网络环境下主机特征信息提取的框图. 基于模块化思想, 为不同维度的主机特征信息分别定制插件以提高数据处理的灵活性, 当面对同维度的数据提取需求时, 直接修改配置文件即可, 由此来提供开放、高扩展性的数据处理能力. 插件的具体实现是构建一颗识别特征树, 父节点代表解析主机特征信息类型, 中间节点代表在该维度主机特征信息下包含的应用程序, 叶子节点则代表解析该应用协议下主机特征信息所使用到的识别特征.

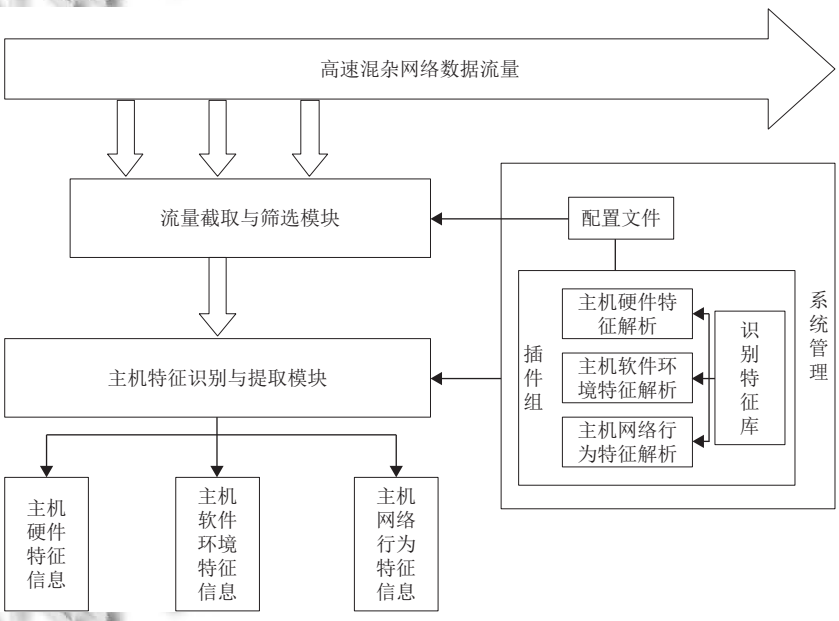

图 1 高速混杂流量下主机特征信息提取框图

系统共分为两个模块: 流量截取与筛选模块和主机特征识别与提取模块. 流量截取与筛选模块主要采用五元组策略对高速混杂的网络原始流量进行初步筛选和过滤, 去除噪音, 缩小数据量. 为了防止硬件的截取速度过快导致主机特征识别与解析模块无法同步处理导致丢包, 系统采用了零拷贝[12]机制减少数据拷贝次数和系统调用, 将网卡数据直接送达上层应用, 实现 CPU 的零参与, 彻底消除 CPU 在这方面的负担, 提高数据处理能力. 主机特征识别与提取模块主要负责应用协议识别, 提取主机特征信息. 该模块通过插件定义的树形结构从根节点依次对应用数据进行识别匹配, 最终通过叶子节点的识别特征解析出主机特征信息. 系统管理主要从三个方面进行管理: (1) 插件的管理, 可以定制插件功能, 定期更新维护插件. (2) 插件状态管理, 控制插件的运行和停止, 监控插件运行状态防止出现空转或者崩溃. (3) 统一管理和调度, 当单个节点无法满足数据处理需求时, 可以进行分布式部署, 对原始数据流量进行分流, 系统统一调配调度插件.

该方法的好处在于, 插件解析的内容可以通过配置文件进行灵活配置, 对于同维度主机特征信息的解析需求只需要增加识别特征即可, 而对于增加识别特征不能满足需求的则可以新增插件. 基于模块化的思想方便系统管理和维护, 只需定期对识别特征库进行更新就可以保证主机特征信息提取的准确性, 且系统支持分布式多点部署.

3 基于 MAP-SCORE 的主机指纹选择算法

通过第二节提取得到不同维度主机特征信息, 但是由于应用协议的规定, 在主机与应用程序交互的过程中不同行为携带的主机特征信息不尽相同, 进而导致主机特征信息标识主机的强度也不同, 标识性弱的主机特征信息会大量频繁地出现在多台主机上, 标识性强的主机特征信息则集中出现在少数的主机上. 而对于一台主机来说, 在一定时间段内主机软硬件环境较为稳定, 这也就为主机指纹信息的提取提供了条件, 如果能够找出频繁出现在一台主机上不同维度的主机特征信息, 且该特征极少出现在其他主机上, 则在一定程度上可以说明该主机特征信息具有较强的主机识别能力, 根据该思想设计针对主机软件环境特征信息 O 和主机网络行为特征信息 B 的主机指纹获取函数 $F_{(O,B)}$, 则可得到$F^{(P)}_{(O,B)}=C$.

由此本文提出了, 基于 MAP-SCORE 的主机指纹选择算法, 其作用是对提取到的不同维度主机特征信息进行评估, 找出符合多维度主机指纹模型定义条件的主机特征信息集合. 该算法主要有两个方法: (1) MAP 方法, 用来构建机器能够计算的主机特征矩阵. (2) SCORE 方法, 基于 MAP 算法构建出的主机特征矩阵评估每一个特征信息对应不同主机的关联度.

3.1 MAP 方法

在以太网环境下, 假设在一定的时间段内, 每一个 IP 对应一台主机. 对提取到的每一条主机特征信息通过 MAP 算法构建如表 1 所示主机特征矩阵, 特征矩阵中每一行表示一条特征 t_i, 每一列表示出现过的主机 x_i, 特征矩阵中MAP_{t_i,x_i}的值则表示特征 t_i 出现在主机 x_i 下的次数.

表 1 主机特征矩阵结构

主机特征信息	总次数(*)	HostA	HostB	HostC	……
P_INFO=m1@163.com	500	500	0	0	……
tracknick=mskx	600	550	0	10	……
Mozilla/5.0	2 000	500	300	200	……
WeChat/6.3.16.17	400	150	50	0	……
……	……	……	……	……	……

可以预见的是, 构建完整的主机特征矩阵是一个高维稀疏矩阵, 随着主机特征信息数量的增加, 数据的检索和匹配会占用大量系统资源, 同时如果使用传统的方式对其存储会导致大量资源的浪费, 而当矩阵维度越来越多, 数据的存储和提取也将受到影响, 为了便于 SCORE 算法的计算, 提高数据存取效率, 提出通过采用带索引的十字链表进行 MAP 矩阵的存储, 如图 2 所示. 首先定义一个 Index 数组, 通过 BKDR 哈希算法对特征信息进行哈希计算, 哈希值作为数组的下标, 而该项的值是一个指针, 指向该主机特征信息, 主机特征信息项存储着该特征出现的总次数且使用双指针分别指向该特征出现的主机项和下一个特征项, 主机项存储该特征在该主机上出现的次数并指向下一台出现该特征的主机, 以此类推, 直到数据包处理完成没有新加项. 这种数据结构对于稀疏矩阵来说可以极大压缩其占用的系统空间, 同时通过哈希值提高特征检索的速度, 冲突率极低.

3.2 SCORE 方法

从表 2 中可以看到, 特征“P_INFO=m1@163.com”全部出现在 HostA 下, 则该特征很有可能成为主机指纹信息, 而特征“Mozilla/5.0”虽然出现的次数很多, 但是覆盖主机的范围也很广, 因此该特征成为主机指纹信息的可能性较低. 由此提出 SCORE 算法的主要思想.

(1) 对于某一台主机而言, 特征出现在一个主机流量中的频率越高且在其他主机流量中频率越低则更能代表这台主机.

(2) 对于一个主机特征信息而言, 出现在不同主机的数量越多越不能代表一台主机.

因此, 假设 SA 代表主机的集合, a 是 SA 中的一台主机即 $a\in SA$, $\delta(t,a)$ 表示特征 t 是否在主机 a 出现过, 出现为 1, 未出现则为 0, 则$\sum_{a\in SA}\delta(t,a)$表示特征 t 在多少台主机下出现过的次数. 因此, 特征 t 对应主机 x 的 SCORE 值可以表示为:

$$SCORE_{(t,x)}=\frac{\frac{MAP_{t,x}}{MAP_{t,*}}}{\sum_{a\in SA}\delta(t,a)} \quad (1)$$

$\frac{MAP_{t,x}}{MAP_{t,*}}$值越大说明特征 t 出现在主机 x 下的次数越多, 特征 t 作为主机指纹的可能性也就越大, $\sum_{a\in SA}\delta(t,a)$越大说明特征 t 作为主机指纹的可能性越小. 以此类推, 对主机特征矩阵中每一条特征 t 分别计算对应不同主机 x 的 SCORE 值, 当 $SCORE(t,x)>\mu$ 时, 认为特征 x 是主机的指纹. μ 是根据大量实验给出的一个经验值.

通过上述 MAP 和 SCORE 方法即完成了主机指纹获取函数 $F_{(O,B)}$ 的整个过程, 得到主机 x 对应的主机指纹信息 C.

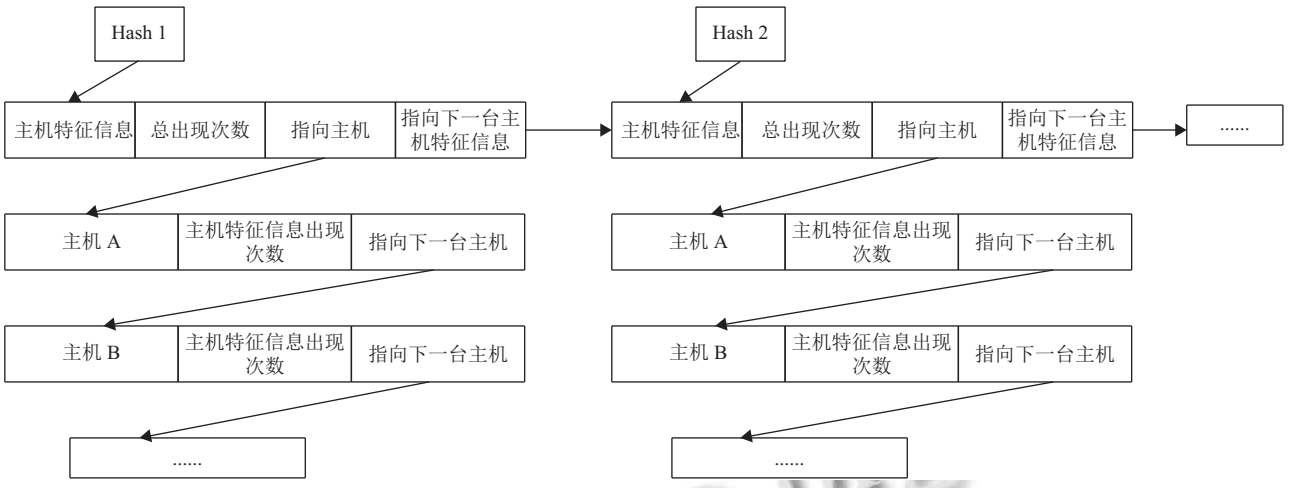

图 2 带索引的十字链表结构

4 多维度主机指纹模型在主机识别中的应用

为了验证方法的有效性, 在校园网的环境下针对文本型数据和二进制数据进行了系统实现和测试. 这两种数据主要由 HTTP 协议承载. 根据对校园网数据统计分析发现, HTTP 协议在校园网网络数据中占比达到 90% 以上. 经过对 HTTP 报文的分析发现具有以下特点:

(1) HTTP 报文内容的每一行都是独立的, 以\r\n 结尾.

(2) 每一行中 Key-Value(键值对) 对以冒号“: ”分割, 冒号前为 Key, 冒号后为 Key 对应的 Value 值.

(3) Cookie 字段中包含有用户账号特征字段和系统唯一编号等唯一标识字段.

(4) Cookie 字段包含有应用协议的 Key-Value 对, 这些 Key-Value 对大多以“=”, “: ”进行分割, 以“; ”, “&”, “|”, “, ”作为结束符.

HTTP 协议主要使用 GET 和 POST 两种方法, 每一种请求方法后都会跟随一个 URL(Uniform Resource Locator, 统一资源定位符), 每一个 URL 标识这个应用程序的一个行为, 如用户登录、信息更新、退出等, 不同的应用程序相同的操作对应着不同的报文头, 而相同的应用相同的操作对应的报文头则较为固定. 在 HTTP 协议包头部分的 User-Agent 字段承载了操作系统类型及版本、浏览器类型和版本信息和应用程序版本等信息, 该信息由网络应用程序规定填充格式, 在通信的过程中由客户端进行填充. 由于其在一定程度上可以反映主机的软件环境, 因此符合主机软件环境特征的定义, 在系统实现时主要以该字段作为主机软件环境特征信息.

现如今在主机上运行的网络应用程序大多需要使用账号进行登录, 对于一台主机而言, 主机访问网络应用程序使用到的账号信息是较为稳定的, 频繁出现的一组账号信息不会经常出现在其他主机下. 通过对近 20 种网络应用协议的分析发现在 cookies 中会存在大量的账号信息和应用程序为主机分配的唯一编号, 因此, 主机访问网络使用的账号信息可以作为主机网络行为的特征信息. 表 2 给出了部分应用账号提取的识别特征关键字.

表 2 不同应用程序账号提取识别特征关键字

应用名称	识别特征	结束符	应用名称	识别特征	结束符
163邮箱	P_INFO=	|	qq空间	ptui_loginuin=	;
126邮箱	P_INFO=	|	淘宝	Tracknick=	;
yeah邮箱	P_INFO=	|	京东	mp=	;
Sina邮箱	freeName=	;	Weibo	un=	;
qq邮箱	qm_username=	;	豆瓣	ue=”	“
搜狐邮箱	Pprecoverinfo=”	“	Csdn	UserName=	;
189邮箱	ACCOUNT=	|	战旗	acf_nickname=	;

原型系统主要从主机软件环境特征信息和主机网络行为特征信息两个维度构建主机指纹库. 具体实现流程如图 3 所示, 高速混杂的网络数据流量到达主机网卡后, 首先采用零拷贝技术对数据进行处理, 根据五元组过滤规则去除噪音数据和非 HTTP 数据, 然后通过主机软件环境解析插件提取 User-Agent 信息, 同时通过主机网络行为特征解析插件加载应用账号识别特征关键字提取主机访问网络所使用到的应用账号, 主机特征信息提取完成后送入主机指纹获取函数 $F_{(O, B)}$, 经过 MAP-SCORE 算法输出主机指纹库 C, 完成多维度主机指纹库模型的构建.

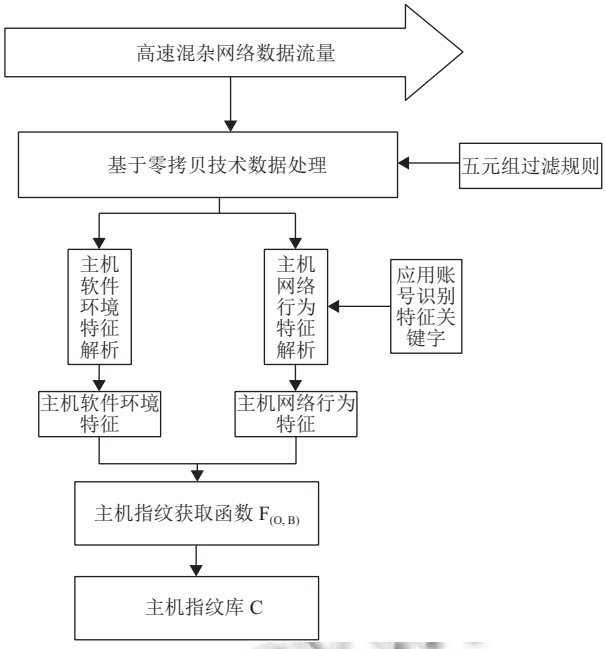

图 3 原型系统实现流程图

5 实验及分析

5.1 实验准备

实验数据通过校园网采集了 35 台主机不同时间段的通信数据. 其中采集 30 台主机 1 个小时的正常通信流量作为构建主机指纹库的样本数据, 对主机的正常上网行为不做要求. 然后在不同的时间段通过校园网再次采集 35 台 (含样本集中的 30 台主机) 主机 30 分钟的正常通信流量作为测试数据. 样本集外的 5 台主机数据作为噪音数据. 由于校园网采用 DHCP 服务为主机分配 IP 地址, 因此在测试集中有 10 台主机的 IP 与样本集中的 IP 不同.

通过多维主机指纹模型对样本集中 30 台主机流量进行多维度主机指纹库构建, 设置 μ=1, 分别通过 $F_{(O)}$、$F_{(B)}$ 和 $F_{(O,B)}$ 构建得到 2 个单维度主机指纹库和一个多维度主机指纹库, 然后分别采用单维度主机软件环境指纹、单维度主机网络行为指纹和多维度主机指纹进行实验, 计算识别主机的准确率和召回率. 假设 P 表示主机识别率, R 表示主机召回率, 计算公式如下:

$$P = \frac{\text{正确识别主机数量}}{\text{系统识别出的主机数量}} \quad (2)$$

$$R = \frac{\text{正确识别主机数量}}{\text{成功构建指纹库的主机数量}} \quad (3)$$

为了进一步验证本文方法的准确性和可靠性, 实验以目前最流行的指纹识别工具 Nmap[7]作对比, 分别与本文方法构建的 2 个单维度主机指纹库和一个多维度主机指纹库在进行主机识别的识别结果进行对比.

5.2 基于 Nmap 的主机识别结果分析

Nmap 工具主要是通过探测主机操作系统来进行识别, 通过对实验所用 35 台主机进行探测结果如表 3 所示.

表 3 Nmap 指纹探测结果

Nmap指纹	相同指纹主机数量
Microsoft Windows Embedded Standard 7	1
Microsoft Windows Vista Home Premium SP1, Windows 7, or Windows Server 2008	1
Microsoft Windows Vista SP2, windows 7 SP1, or Windows Server 2008	1
Microsoft Windows 10 build 10586-14393	2
Microsoft Windows 7 SP0-SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows8, or Windows 8.1 update 1	6
Microsoft Windows Vista, Windows 7 Sp1, or Windows 8.1 Update 1	3
Microsoft Windows Server 2008 SP2 or Windows 10 or Xbox one	3
Microsoft Windows Vista	2
Microsoft Windows Server 2008 R2 or Windows 8.1	3
Microsoft Windows 7	5
Microsoft Windows Server 2008 or 2008 Beta 3	3
Linux 3.2-4.6	5

从表 3 中可以看出, 在 35 台实验主机中只有 3 台主机具有唯一的指纹信息, 因此通过该方法对主机进行识别并不能达到很好的识别效果, 准确率和召回率分别为 25% 和 8.57%. 受限于 Nmap 指纹探测的精度和探测方式, 该方法在现有网络环境下很受限, 识别效率较低.

5.3 基于单维度主机指纹的主机识别结果分析

基于主机软件环境指纹的主机识别结果如表 4 所示. 其中, 未识别到的主机是由于测试集较小未出现软件环境指纹导致未能识别, 而错误识别的主机是由于样本数据较小, 在指纹库构建时引入了非指纹信息. 这种情况只需扩大样本集, 全面获取主机数据就可以解决.

表 4 单维度主机软件环境指纹识别结果

测试数据	完成指纹构建主机	识别出主机数量	正确识别主机数量	准确率	召回率
35台	30台	29台	25台	86.21%	83.33%

基于主机网络行为指纹的主机识别结果如表 5 所示.

表 5 单维度主机网络行为主机识别结果

测试数据	完成指纹构建主机	识别出主机数量	正确识别主机数量	准确率	召回率
35台	17台	10台	10台	100%	58.82%

一般而言, 一台主机的网络行为较为稳定, 如果能够全面采集到主机的网络行为指纹, 则对主机的识别效果会很好. 上述实验结果可以发现, 采用单维度主机网络行为指纹对主机进行识别的准确率很高, 但是由于在采集测试集的过程中, 网络行为并不一定发生, 因此导致召回率较低.

5.4 可变 IP 地址流中多维主机指纹库主机识别结果分析

基于多维主机指纹库的主机识别结果如表 6 所示.

通过结果可以看出, 多维主机指纹库在识别主机准确率和召回率上相比单维度主机指纹识别都有近 9% 的提升, 主机软件环境指纹和主机网络行为指纹在识别主机的过程中可以相互补充, 弥补由于单维度指纹获取不到导致的无法识别主机的情况. 同时在测试集中有 10 台主机的 IP 与样本集中使用的 IP 不同, 测试结果均正确识别, 说明多维主机指纹库在对主机识别时可以容忍主机 IP 地址发生变化. 通过与传统方法的比较, 本文提出的方法在保证采集的主机通信流量完整的情况下, 主机指纹模型具有良好的健壮性和可靠性.

表 6 多维主机指纹库主机识别结果

测试数据	完成指纹构建主机	识别出主机数量	正确识别主机数量	准确率	召回率
35台	30台	30台	28台	93.33%	93.33%

6 结束语

本文研究了基于多维度指纹模型构建主机指纹库的方法, 给出了网络主机多维度指纹模型的设计思想和设计方案, 解决了高速混杂流量下多维度主机特征信息的提取和多维度主机指纹模型构建问题. 通过从多维度描述主机进而提高了识别主机的准确率得到了较好的结果. 未来的工作: (1) 扩充可以描述主机特征的类型并研究设计高效的特征提取算法. (2) 目前不同类型的特征并没有考虑其对主机的重要程度, 如何识别不同类型的主机并根据其类型定向采集特征并在构建多维度主机指纹模型时考虑特征所占的权重是下一步研究的重点.

参考文献

1 Kohno T, Broido A, Claffy KC. Remote physical device fingerprinting. IEEE Trans. on Dependable and Secure Computing, 2005, 2(2): 93–108. [doi: 10.1109/TDSC.2005.26]

2 Polčák L, Franková B. On reliability of clock-skew-based remote computer identification. Proc. of the 11th International Conference on Security and Cryptography (SECRYPT). Vienna, Austria. 2014. 1–8.

3 Polčák L, Franková B. Clock-skew-based computer identification: Traps and pitfalls. Journal of Universal Computer Science, 2015, 21(9): 1210–1233.

4 隋新. 主机特征信息被动识别的研究与实现. 科学技术与工程, 2013, 13(3): 652–658.

5 Eckersley P. How unique is your web browser? Proc. of the 10th International Symposium on Privacy Enhancing Technologies. Berlin Heidelberg, Germany. 2010. 1–18.

6 Soltani A, Canty S, Mayo Q, *et al*. Flash cookies and privacy. AAAI Spring Symposium Series: Intelligent Information Privacy Management. 2010. 158–163.

7 NMAP free security scanner, 2017. http://www.insecure.org/nmap/.

8 Ma JL, Wang XL, Xiao B. Research of the enhanced anti-xprobe2. Proc. of the 2016 International Conference on Networking and Network Applications (NaNA). Hakodate, Japan. 2016. 122–125.

9 Zalewski M. p0f v3: Passive fingerprinter, 2012. http://lcamtuf.coredump.cx/p0f3/README.

10 Yen TF, Xie YL, Yu F, *et al*. Host fingerprinting and tracking on the web: Privacy and security implications. Proc. of the 19th Annual Network and Distributed System Security Symposium (NDSS). San Diego, CA, USA. 2012.

11 Polčák L, Jirásek J, Matoušek P. Comment on “remote physical device fingerprinting”. IEEE Trans. on Dependable and Secure Computing, 2014, 11(5): 494–496. [doi: 10.1109/TDSC.2013.26]

12 吕民强, 吕丹丹. 一种高效的零拷贝报文捕获系统. 航空计算技术, 2015, 45(5): 102–105.