第37卷 第16期
Vol.37 No.16

计 算 机 工 程
Computer Engineering

2011年8月
August 2011

·软件技术与数据库·

文章编号：1000—3428(2011)16—0036—03 文献标识码：A 中图分类号：TP311

基于模糊匹配的专用库函数识别技术

吴 滨[1]，蒋烈辉[1]，舒 辉[1]，方 霞[2]

(1. 解放军信息工程大学信息工程学院，郑州 450002；2. 海军航空工程学院，山东 烟台 264001)

摘 要：针对传统库函数识别方法无法有效识别专用库函数的问题，提出基于模糊匹配的专用库函数识别技术。在库文件快速识别与鉴定技术(FLIRT)的函数签名机制的基础上做出改进，提取目标文件的有效函数集并利用专用库函数特征库进行模糊匹配，确定需要加载的库签名，加载签名完成精确匹配。实验结果证明，该技术在专用库函数的识别方面效果较好。

关键词：专用库函数；库文件快速识别与鉴定技术；模糊匹配；函数签名；有效函数集

Specific Library Function Identification Technology Based on Vague Matching

WU Bin[1], JIANG Lie-hui[1], SHU Hui[1], FANG Xia[2]

(1. Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China;
2. Navy Aeronautical Engineering Academy, Yantai 264001, China)

【Abstract】Aiming at the problem that traditional library function identification technology can not recognize specific library function effectively, this paper proposes a specific library function identification technology based on vague matching. The technique improves the function signature mechanism produced by Fast Library Identification and Recognition Technology(FLIRT), extracts a valid function congregation from the target file and does vague matching using professional library function feature database. Vague matching determines the signature to be loaded, loads the signature and fulfils the accurate matching. Experimental results show that the method does well at specific library function identification.

【Key words】specific library function; Fast Library Identification and Recognition Technology(FLIRT); vague matching; function signature; valid function congregation

DOI: 10.3969/j.issn.1000-3428.2011.16.012

1 概述

库函数识别是反编译过程中的一个重要环节。经统计，高级语言编写的程序使用的函数中平均有50%是库函数，识别出库函数模块能够有效减少反编译的工作量，提高反编译结果的正确性及可读性，并且有利于更好地了解程序意图和分析程序功能。

随着技术的不断进步，专用函数库的应用领域日益宽广，各种专业技术领域，如图形图像处理、密码学、网络应用、嵌入式应用等，都在不同程度上使用了本领域专用的函数库(如zlib、tomcrypt等)，成功识别专用库函数会给相关领域的软件逆向工作带来巨大便利。然而，传统的库函数识别技术大多针对系统库函数和各类标准库，在专用库函数的识别方面效果很不理想，亟待改进。

本文在库文件快速识别与鉴定技术(Fast Library Identification and Recognition Technology, FLIRT)的基础上做出改进，提出一种利用模糊匹配进行专用库函数识别的方法，并给出该方法的具体实现。文中所指的库函数均为静态链接至最终程序的静态库函数，动态链接情况较为简单，不做说明。

2 相关研究

库函数的识别技术作为反编译的重要内容之一，处于不断的发展和完善中。总结现阶段国内外已有的研究成果，主要的识别方法有2种。

2.1 基于特征数据库的模式匹配方法

这种方法最早是由澳大利亚昆士兰大学反编译研究小组于1995年提出的，其基本思想是提取库函数目标代码的某些属性作为模式特征，将提取出的模式统一存放至一个特征数据库中；识别时提取待识别函数的相应特征，在特征数据库中进行模式匹配，若匹配成功，则输出该库函数名。

国内外很多研究人员针对该方法做出了改进，改进大多体现在函数特征的选取方面：文献[1]提出以操作码序列作为库函数特征。文献[2]提出在函数特征中加入可变操作码，以便于识别C++模板函数。文献[3]提出基于基本块划分的库函数识别技术，以基本块以及块间关系作为函数的特征。

以上提到的方法虽然在具体实现上各有不同，但核心思想区别不大，在对系统库函数进行识别时效果良好。然而，专用函数库的种类繁多、数目庞大，使用的编译器、针对的体系结构等更是不尽相同。使用模式匹配的方法直接进行专用库函数的识别，会严重降低识别的效率，系统资源的使用量也将无法控制。

2.2 FLIRT

反编译工具IDA使用的FLIRT是迄今为止应用最广泛的库函数识别技术。该技术提取函数开头的32个字节作为模式特征，辅以函数长度、外部符号引用等属性作为辅助特征；使用哈希算法对函数特征进行签名，并将得到的签名保存至特定的签名文件中。通常每种编译器对应一个签名文件；进行函数识别时，只需加载对应的签名文件，即可完成识别

作者简介：吴 滨(1987－)，男，硕士研究生，主研方向：软件逆向工程；蒋烈辉，教授、博士生导师；舒 辉，副教授、博士；方 霞，硕士

收稿日期：2011-01-25 **E-mail：**wubin19870516@163.com

工作[4]。

签名机制避免了不必要的匹配，大大提高了函数识别的效率。IDA 使用启动签名识别的方法自动选择需要加载的签名文件，其原理即根据不同编译器的特征识别出该文件使用的编译器，并加载该编译器对应的签名文件。系统函数库与编译器联系紧密，该方法在识别系统库函数时是行之有效的。但是，在软件中使用专用函数库并不会影响最终生成的可执行文件的编译器特征，因此，启动签名机制无法有效识别并自动加载专用函数库签名。面对未知的二进制文件，手动选择需要加载的专用库签名是不现实的。

综上所述，传统的库函数识别方法在对专用库函数进行识别时都存在一定的缺陷，需要针对专用库的特点加以改进。

3 基于模糊匹配的专用库函数识别

针对专用函数库种类繁杂、数目庞大、结构多样等特点，本文提出了一种基于模糊匹配预处理的函数识别方法。以下是 2 个基本定义。

定义 1 有效函数集

有效函数集是指待识别的可执行文件中的一个符合约束条件的函数模块集合 S，设该可执行文件的全部函数模块集合为 U，则有 $S \subseteq U$；对于任意的函数模块 $f \in U$，f 需要进行匹配当且仅当 $f \in S$。

定义 2 相关库和相关函数

对于任意的函数库 L，若存在函数模块 $l \in L$ 使得 l 与待识别文件中的某函数模块 f 具有完全相同的属性特征，称该函数库 L 为待识别文件的相关库，函数模块 f 则称为函数库 L 的相关函数。

3.1 识别方法

由于专用函数库数目庞大且采用的编译器不尽相同，如果能够判断出待识别的可执行文件使用了哪些库，再针对这些库进行函数识别，无疑会大大提高识别效率和准确性——这就是基于模糊匹配的函数识别方法的基本思路。模糊匹配指的是判断可执行文件使用的函数库的过程，这是针对最终的精确匹配而言的。

本文中函数识别是建立在反汇编得到的汇编代码的基础上的，采用的反汇编引擎是 IDA，所有工作均基于 IDA 得到的反汇编结果进行。

整个识别过程可分为 3 个步骤：库函数特征提取，可执行文件的预处理以及函数匹配识别。首先对专用函数库进行特征提取，将获得的库函数特征存储至专用库函数特征数据库中，以备识别时使用；然后在分析具体的可执行文件时，对其进行预处理，提取出待识别文件的有效函数集，对有效函数集中的函数进行特征抽取；接下来在专用库函数特征数据库中匹配有效函数集中的函数，进行模糊匹配，获得该可执行文件使用到的函数库的相关信息；再根据模糊匹配的结果加载库签名，进行精确匹配，得到最终结果。

上述识别流程如图 1 所示。需要说明的是，这种识别方法是对 FLIRT 识别机制的一种改进，最终的精确匹配步骤使用了 FLIRT 的签名匹配机制，即采用最小完美哈希算法完成函数特征匹配。文献[5]阐述了该算法的原理，在此不予赘述。

自动确定需要加载的库签名文件，需要对提取出的有效函数特征进行模糊匹配。模糊匹配的核心思想在于利用函数的一些属性，快速准确地在数据库中查找到匹配项；需加载的签名信息则被事先记录在数据项中，匹配成功后即可获得该信息。模糊匹配作为函数识别的一项辅助策略，不应影响函数识别的效率。因此，需要采取一些措施加快匹配速度。本文采用了多级索引、对相关库及相关函数进行特殊处理等方法来提高匹配速度。

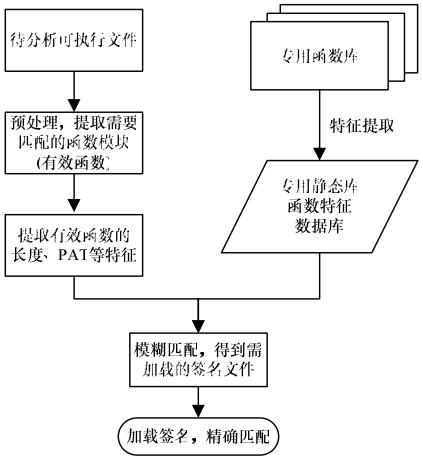

图 1 专用库函数识别过程示意图

3.2 特征选取

在库函数识别技术中，函数特征的选取是关键所在。本文中的函数特征分为 2 种：(1)模糊匹配时使用的函数特征，即函数长度、基本块数、函数的 PAT 组成的三元组；(2)进行精确匹配时使用的函数特征，即 FLIRT 采用的函数特征。后者可参考文献[4]，在此不展开说明。

选择函数长度和基本块数作为特征是为了建立多级索引，提高模糊匹配的速度。而函数的 PAT 这一概念同样出自 FLIRT，但由于 PAT 格式过于复杂，会影响到匹配效率，因此在实践中对其进行了简化。简化后的 PAT 内容如下：

(1)函数的前 32 Byte，可变操作数用通配符代替，函数长度不足 32 Byte 的用通配符补足。

(2)从第 33 Byte 开始到第一个外部引用(或全局变量)处的所有字节的 CRC16 值。

(3)用来计算 CRC 值的字节数。

在专用库函数特征数据库建立的过程中，除了上述的函数特征之外，每个数据项还需要包括函数名、对应的库文件名以及对应的签名文件等信息，后两项用于对相关库和相关函数进行特殊处理。需要说明的是，此数据库仅用于在模糊匹配阶段查找需加载的库签名，后续的精确匹配并不依赖于该数据库。

4 具体实现及算法描述

4.1 有效函数集的提取

可执行文件由大量的函数模块组成，根据文件格式以及文件对应的指令集，可以比较准确地从可执行文件中将这些函数模块提取出来。然而，原始的函数模块数量巨大，而专用库函数在其中所占的比例却很小，对原始函数模块逐一进行匹配会产生大量无效操作，使识别效率大大降低。因此，本文提出了有效函数集的概念，见定义 1。

本文提出的约束条件有 3 个，满足这 3 个约束条件的函数即可加入有效函数集中：(1)函数的长度必须大于阈值 x(可设置)，避免短函数的识别；(2)函数中不能有对程序数据段的直接或间接访问，因为只有用户函数才会访问程序的数据区；(3)将标准库函数排除，避免重复识别，可通过预加载相应的标准库签名实现。

有效函数集提取的算法如下：

输入 反汇编后的目标文件函数模块集合 U

输出 有效函数集 S

```
S: = ∅    //初始状态为空集
for each f ∈U
 do
  if f_length > x    //判断该函数的长度是否小于阈值 x
   then
      if f_data() = false //判断该函数是否引用了.data 段的数据
       then
          if f_standard() = false //判断该函数是否为标准库函数
             then
                S = S U {f};
          endif
      endif
  endif
 endfor
Output S
```

限于篇幅，本算法中的关键函数 f_data()不再给出详细算法，仅给出大致原理：判断是否引用数据段内容，只需依次遍历函数中的每条指令，遇到跳转指令时进行递归，只要在递归过程中跳转到数据段所在地址，则停止递归返回 true，否则返回 false。

4.2 模糊匹配的实现

模糊匹配模块以从有效函数集中提取的函数特征集合作为输入，通过与特征数据库中的数据项进行匹配，输出需要加载的库文件签名集合。整个模块的工作流程如图 2 所示。

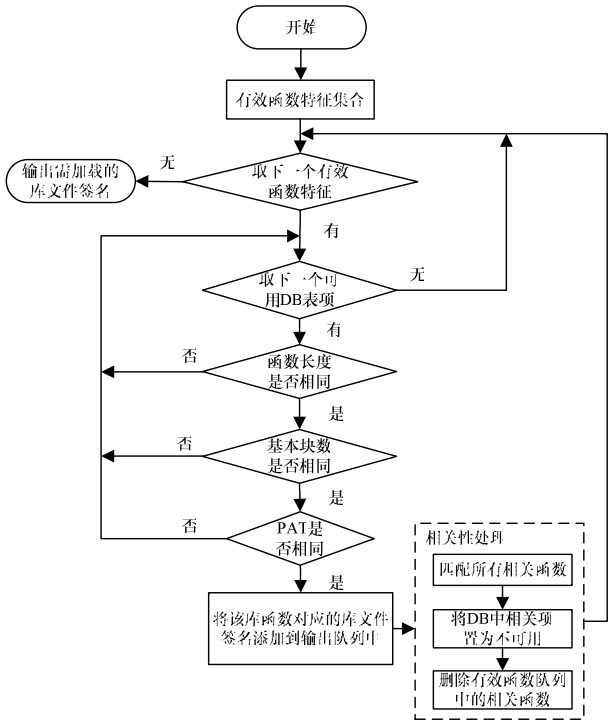

图 2 模糊匹配流程

在对数据库进行查询匹配时，使用多级索引的方法提高查询速度，分别以函数长度和基本块数目作为前两级索引。同时，为了进一步提高匹配效率，引入了相关性处理的步骤：当某一个库函数匹配成功之后产生了相关库时，对该相关库中的所有函数进行优先匹配，匹配结束后将与该相关库有关的所有数据项置为不可用(数据项中的可用属性置为 false)；同时，将处理过程中产生的相关函数从有效函数集中删除。

事实上，只要有一个函数匹配成功，该函数所对应的函数库的签名就必须被加载。因此，使用相关性处理将该库的相关函数从有效函数集中排除出去，可以减少大量不必要的匹配操作，当数据库规模庞大时，这种处理将显著提高模糊匹配的效率。

5 实验结果

为了验证上述方法的有效性，选取了若干具有代表性的目标程序进行测试。测试程序中均静态链接有专用库函数，主要测试库函数的识别率。测试结果如表 1 所示。

表 1 实验有效性测试结果

程序	使用的专用函数库	库函数数量	识别数量	识别率/(%)
WinCrypto	mcrypt/libpng	30	29	96
BOAST	ScalePort/MKL	89	87	97
IMPacker	tomcrypt	29	29	100
WinRAR	zlib/BZip2	76	76	100

从实验结果可以看出，本文的识别方法可以比较准确地识别出程序中使用的专用库函数，能够满足实际应用中的需要。

6 结束语

本文提出一种基于模糊匹配自动加载函数库签名的识别机制，弥补了 FLIRT 启动签名机制无法有效识别专用库签名的缺陷，并且具有较高的识别效率，本文提出的方法在专用库函数识别方面效果良好。下一步的研究方向在于寻找新的匹配约束条件，进一步提高模糊识别的速度。

参考文献

[1] 许向阳, 雷 涛, 朱 虹. 反编译中的静态库识别研究[J]. 计算机工程与应用, 2004, 40(9): 37-39.

[2] 胡 政. C++逆编译中的模板库函数识别研究[D]. 合肥: 中国科学技术大学, 2006.

[3] 邱 景. 基于基本块划分的库函数快速识别技术[J]. 计算机工程, 2009, 35(21): 88-90.

[4] Guilfanov I. Fast Library Identification and Recognition Technology[EB/OL]. (2007-01-10). http://www.DataRescue.com/idaflirt.htm.

[5] Czech Z J, Havas G, Mahewski B S. An Optimal Algorithm for Generating Minimal Perfect Hash Functions[J]. Information Processing Letters, 1992, 43(5): 257-264.

编辑 顾逸斐

(上接第 29 页)

参考文献

[1] 白星振, 李晓梅, 吴 娜. 基于 WSN 目标跟踪的移动 Agent 路由算法[J]. 计算机工程, 2009, 35(10): 11-14.

[2] 耿 峰, 祝小平. 一种改进的多传感器多目标跟踪联合概率数据关联算法研究[J]. 系统仿真学报, 2007, 19(10): 4671-4675.

[3] 薛 锋, 刘 忠, 曲 毅. 无线传感器网络中的分布式目标被动跟踪算法[J]. 系统仿真学报, 2007, 19(8): 3499-3502.

[4] Wu Yuanxin, Hu Dewen, Wu Meiping, et al. Quasi-Gaussian Particle Filtering[M]//Alexandrov V N. Lecture Notes in Computer Science, Volume 3991. Heidelberg, Germany: Springer, 2006.

[5] Doucet S, Freitas J F G, Gordon N J. Sequential Monte Carlo Methods in Practice[M]. New York, USA: Springer-Verlag, 2001.

[6] 杨小军, 邢科义, 施坤林, 等. 传感器网络下机动目标动态协同跟踪算法[J]. 自动化学报, 2007, 33(10): 1029-1035.

编辑 张正兴