软件逆向工程分析技术研究及应用

刘 明

(中国航空计算技术研究所,陕西,西安 710068)

摘 要:软件逆向工程对软件维护、复用和创新提供了可靠保证,但一直未见成熟的软件逆向工程理论和方法。介绍了软件逆向工程的基本概念,主要研究了软件逆向分析中常用的静态分析方法和动态分析方法,以汽车实时嵌入式控制系统软件为例介绍了两种分析方法的应用,并对两种分析方法进行比较,可为软件逆向工程的发展和深入研究提供借鉴。

关键词:软件逆向工程;静态分析方法;动态分析方法;应用
中图分类号:TP311 文献标识码:A 文章编号:1671-654X(2011)02-0093-03

Research on Analysis Technology in Software Reverse Engineering and Application

LIU Ming

(Aeronautical Computing Technique Research Institute, Xi'an 710068, China)

Abstract Software reverse engineering provides a reliable protection for software maintenance, reuse, and innovation, however it has no mature theory and methods. This paper introduces the basic concepts of software reverse engineering, mainly research static analysis and dynamic analysis which are commonly used in software reverse analysis, then introduce the application of two methods in real-time embedded control system for the automobile, and compare two methods, lays a solid foundation for developing and further studying of software reverse engineering.

Key words software reverse engineering; methods of static analysis; methods of dynamic analysis; application

引言

面对日益复杂和庞大的软件需求,软件逆向工程在软件工程中发挥着越来越重要的作用。借助已有的设计良好,性能优越的软件系统能够快速开发出一个有效的复杂软件,然而由于过去软件过程化的不规范,造成文档缺失严重,给软件的识别和使用带来了很大困难,甚至有些软件一直作为黑盒使用,阻碍了软件技术的探索和发展。软件逆向工程通过对软件的重新理解和分析探索软件实现原理,生成相关文档,为软件的维护、复用以及创新提供了可靠保证。

国内外对软件逆向工程的研究已经有三四十年,但一直未形成完整的理论和方法,在软件逆向工程领域还有很多内容需要深入研究。本文主要研究了软件逆向分析中常用的静态分析方法和动态分析方法以及两者的应用,对两种方法进行了简单比较。

1 软件逆向工程基本概念

软件逆向工程(Software Reverse Engineering)又称软件反向工程,是指从可运行的程序系统出发,运用反汇编、系统分析、程序理解等多种计算机技术,对软件的结构、流程、算法、代码等进行逆向拆解和分析,推导出软件产品的源代码、设计原理、结构、算法、处理过程、运行方法及相关文档等[1]。通常,人们把对软件进行反向分析的整个过程统称为软件逆向工程,把在这个过程中所采用的技术都统称为软件逆向工程技术。

现实中,人们并不总是完全需要逆向出目标软件的所有功能[2],如果那样的话将会是一个艰苦而漫长的过程。大多数情况下是意图通过对软件进行逆向,

收稿日期:2010-10-08 修订日期:2011-03-03
基金项目:总装预研项目资助(513150401)
作者简介:刘 明(1983-),女,陕西西安人,助理工程师,硕士研究生,研究方向为计算机软件。

从中获取软件的算法, 或破解软件及进行功能扩展等。软件逆向工程包括逆向分析和再工程, 本文只关注逆向分析过程, 对软件再工程暂时不予考虑。

2 软件逆向分析方法

2.1 软件逆向分析流程

软件逆向工程是从可执行的程序系统出发, 逆向分析可执行程序的源代码或反汇编的伪汇编代码, 运用程序理解等技术手段, 还原出目标程序的源代码、系统架构及相关设计文档等[3]。

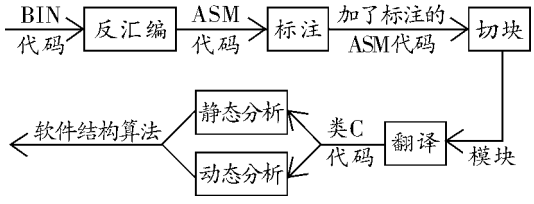

图 1 软件逆向工程流程

图 1所示为软件逆向工程的流程, 首先是对目标 BIN 代码进行反汇编, 一般通过工具完成, 再对反汇编后的代码进行标注、切块和翻译等预处理, 得到类 C 代码, 之后采用静态分析方法和动态分析方法对类 C 代码进行分析, 得到软件结构算法, 达到分析目的。

2.2 静态分析方法

对软件的分析一般都从静态分析开始, 通过阅读程序, 跟踪数据变化过程和控制过程, 对代码有一个初步认识, 静态分析方法[4]主要有以下 4种:

1)词法和语法分析。对程序最基本的分析首先是词法和语法分析。通过分析代码中的分支情况, 循环情况, 以及调用关系, 语句含义, 变量之间的联系等等, 对代码功能有一个初步的认识。

2)图形、图表。在程序分析过程中, 为了能够对程序功能有更为清晰的表达, 一般也可采用图形化的方法。在面向对象的软件逆向分析过程中, 组件之间关系比较复杂, 包括继承、泛化、访问、多态等, 所以常采用类图、协作图等表示不同对象或类之间的关系; 传统的软件中一般采用流程图、结构图等表示程序的功能模块之间的关系或内部流程。

3)复用代码分析。对代码中常出现的代码段或函数功能段进行深入分析, 对重复代码的大小, 次数, 上下文信息等的分析能够帮助我们更好地理解使用这些重复代码的程序功能。

4)程序切片。这主要是针对较长的程序段的分析方法, 根据程序中的跳转语句、判断语句等, 将程序分成较小的片段, 这样每一个小片段的功能比较单一, 而且只涉及尽量少的代码, 在功能理解上更为容易, 唯一不足地是, 要关注变量取值, 注意上下文关系。此部分可以使用工具帮助完成。

2.3 动态分析方法

只通过静态模型很难理解一个复杂系统的行为, 因此监视并获取系统运行时产生的动态信息是十分重要的。动态分析部分主要有 3种实现方式:

1) 采用在源代码中植入 (instrument)语句, 称之为植入法。目前各种工具中收集动态信息主要采用这种方法。其主要原理是利用代码的结构信息, 依据固定的规则, 将软件触发器添加到代码中。所谓软件触发器, 是指在源程序中相应的位置添加的一些代码, 运行时由这些代码按特定协议将指定的动态信息传递到指定位置, 或传递给动态信息收集机制, 从而提供产生动态模型所需的对象之间的消息传递信息。

2) 采用调试器获取动态信息。调试器 (Debugger)使用户可以在程序运行时控制代码的执行, 检查程序状态以及变量的值等信息。

3) 在二进制码中植入信息。这方面需要良好的工具支持。

3 分析方法应用

3.1 实例描述

本文以汽车运行时嵌入式实时控制系统软件分析为例说明软件逆向分析技术的应用。在实例中除可执行代码外, 可参考资料有汽车相关理论, 硬件芯片资料和 I/O 变量说明以及 workbench仿真环境。workbench 仿真环境可以运行目标码获取运行时信息。通过对控制系统软件的分析, 了解汽车运行的自动控制方法, 为进行代码维护或功能扩充奠定基础。

3.2 静态分析方法应用

静态分析主要是通过程序的词法语法分析, 提取数据结构、控制等相关信息。在对控制系统软件类 C 代码静态分析的过程中, 本文主要分析三类信息。

3.2.1 函数相关信息

1)函数调用关系。在类 C语言代码中, 函数的调用都是以“ call函数名”出现。在本文分析的类 C代码中主要有两种调用方式存在: 一种是显示的函数调用, 如 call 0x643f0, 在反汇编代码中, 函数名均为地址表示; 另一种是函数指针调用, 形式均为 call CTR, CTR 作为函数指针存储函数名称, 程序运行过程中根据不同情况, 赋值不同的函数地址, 调用完成不同处理。

2)函数参数。在分析过程中, 要弄清楚参数的个数, 参数的类型和用途。在本系统中, 函数参数主要有地址, 堆栈, 寄存器数据, 对应的参数传递方式也分为堆栈方式、堆栈和寄存器结合方式。程序中, 将 R1寄

存器作为栈基址, 在每段函数开头和结尾都有对应的入栈和出栈操作, 代码中凡是出现 R1的操作均为栈操作。另外程序过有 R13为基址的全局变量, 这是一个全局数据结构, 对此结构中变量的取值和赋值均以“R13 + 偏移”的形式出现。通过全局数据结构也可以进行数据传递。

3)局部变量。局部变量即函数内部使用的变量, 只在函数运行过程中有效, 函数执行完后即被释放。参数和局部变量的分配如图 2所示。

4)返回值。函数的返回值中往往记录了计算结果或标志等重要信息, 在本系统中, 返回值有两种形式, 一种是以 LR 链接寄存器的方式返回值, 另一种则是通过给地址赋值, 由地址进行参数返回。

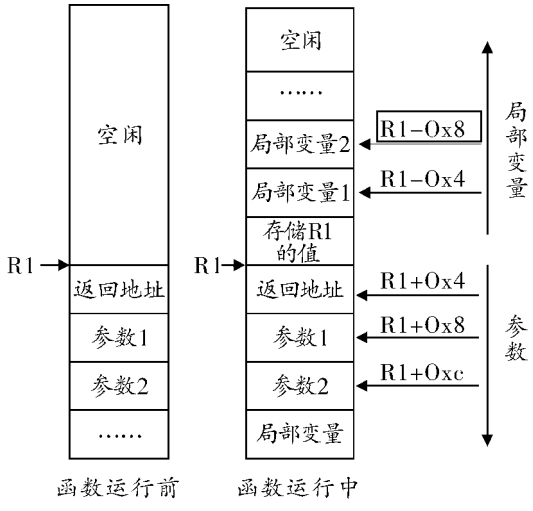

图 2　参数和局部变量分配

3.2.2　数据结构

软件由数据和算法构成。在对程序的逆向分析中, 数据的分析至关重要, 它是软件的必备要素之一。代码中常用的数据结构有数组、结构体和指针, 其他复杂的数据类型都可以通过这三种数据结构表示。在反汇编的代码中, 指针已经转换为地址, 所以在分析的过程中重点分析数组和结构体。数组的识别比较简单, 在代码中有很多以“基址 + 偏移”方式存在的数据, 这些数据长度一致且地址连续, 可以认定为数组; 而对结构体的识别则稍微复杂一些, 但仔细查看每个数据的长度和对应的操作, 如果长度不一致或变量类型不同则认定为结构体。

3.2.3　循环、控制语句

循环、控制语句是软件的另一重要组成部分, 软件的算法主要通过循环、控制语句完成。通过对这些语句的分析可以有效地分析程序的数据流和控制流, 从而达到对函数功能的理解。常用的循环、控制语句有 while, for, if, switch等语句, 在分析时主要查看其判断条件和循环执行条件。例如在程序中有很多数据块搬家, 即将数据从 flash中移到活存中, 通过查看循环条件就可以确定数据块的长度; 通过判断语句的取值也能够判断某些开关量当前状态, 中断使能 /禁止或变量取值范围等等。这些信息的正确性对系统的分析有巨大的作用。

3.3　动态分析方法应用

由于控制系统代码量大, 分支多, 在对程序未进行透彻分析时很难进行代码植入, 而且代码植入对分析人员的汇编语言水平要求较高, 经常要借助工具的帮助才能够完成, 所以在动态分析过程中, 主要借助调试器获取信息。通过调试器运行程序, 在程序运行过程中主要通过以下途径获取信息:

1)设置断点。当代码的流程比较复杂时, 可以设置断点单步跟踪程序流程, 在跟踪的过程中可以查看变量值的变化过程, 帮助程序功能理解。常用的设置断点方法[5]有代码定位断点、数据断点和条件断点, 分别跟踪代码的运行, 内存数据和条件判断。

2)内存修改。通过运行代码找到对应内存地址, 修改内存值, 观察修改后程序的变化, 从而判断数据的意义。在分析系统的应用层功能时此方法非常有效, 根据数据的变化能够很快对应数据段到应用模块。

3)代码屏蔽。屏蔽一段代码, 观察程序运行时的变化判断被屏蔽的代码段的功能; 或当某一段代码因为条件不满足无法正常执行时, 将其屏蔽[5], 使得下面的代码能够正常运行。屏蔽代码的方法也比较多, 可以采用 NOP覆盖, 或者修改栈地址中函数返回地址, 对于中断可以设置硬件中断屏蔽等。本文分析的系统在仿真运行时总是在减一中断处死循环, 程序无法继续运行, 猜测是错误处理, 此处就通过屏蔽硬件中断, 使得程序直接越过此中断, 执行后面代码。

4)改变程序流程。在分析过程中发现 x643f0, 0x63dc0和 0x64270等函数存在难理解, 调用次数多, 输入输出明确的特点, 而部分函数如 0且在实际程序运行中, 总是很难满足全部的分支条件进入到程序的执行中。借助仿真平台, 在 workbench中编写代码直接传递参数调用 flash中的函数: 其核心代码如下: asm(“0x643f0(ID, Addr)”), 其中 ID, Addr是输入参数, 从函数 0x643f0被调用处提取并进行简单预处理, 通过 ID计算, 将计算结果记录在 Addr中输出。再通过分析输入输出数据理解程序功能。

(下转第 104页)

3 结束语

进入 21世纪,单兵作战在当前战争发展中扮演着越来越重要的角色。嵌入式通信技术正不断成长完善,成为嵌入式技术发展的热点领域。将二者结合,着手研究一种适用于嵌入式通信系统的军用网络协议栈,具有重要的现实意义。本文提出了基于嵌入式 Linux的军用 TCP/IP网络协议栈设计并达到了预期效果。该协议栈简洁、可靠、安全,也适用于民用或商用的小型局域网设备,具有一定实用性。

参考文献:

[1] (美)Douglas E Comer著. 用 TCP/IP进行网际互联,第一卷:原理、协议与结构[M]. 林瑶,蒋慧,杜蔚轩译. 北京:电子工业出版社, 2004

[2] (美) Douglas E Comer用 TCP/IP进行网际互联,第三卷:客户 -服务器编程与应用 (Linux) [M]. 北京:电子工业出版社, 2001.

[3] Lauren A Chappell, Ed Tittel著. TCP/IP 协议原理与应用 Guide to TCP/IP[M]. 马海军,吴华译. 北京:清华大学出版社, 2005.

[4] 谢兵,面向嵌入式系统的网络通信协议 [J]. 电子设计应用, 2003(12): 60- 62

[5] (美) Douglas E Comer, David L Stevens著. 用 TCP/IP进行网际互联,第二卷:设计、实现与内核[M]. 张娟,王海,黄述真译. 北京:电子工业出版社, 2003.

[6] AtulKahate著. 密码学与网络安全[M]. 邱仲潘译. 北京:清华大学出版社, 2005.

[7] William Stallings 密码编码学与网络安全:原理与实践[M]. 北京:电子工业出版社, 2005.

[8] 徐含乐,张科,田进. 基于 Xscale腕带式单兵通讯系统设计与实现[J]. 计算机测量与控制, 2010(6).

(上接第 95页)

5)读取运行时数据。在分析中发现有很多数据表记录变量的默认值、换算比率或临时中间值等,通过静态分析很容易跟踪地址表结构,但其中的值的变化却很难分析;另外还有一些在程序中经常使用的特殊地址。针对这些情况,可以借助调试器获取这些地址中的值,推进分析工作。如 DSPI通信接口自测试,通过代码分析只能发现发送数据的地址和接收数据的地址,是自测还是数据通信不清楚,通过 workbench运行程序查看,发现发送和接收地址中的数据一样,从而肯定是自测。另外可以通过 workbench获取地址结构中的所有运行值,当然还要通过对控制流的跟踪才能确定所抓取的值在何种情况下有效。

3.4 分析方法比较

静态分析是对程序的全面分析,它以程序为中心,不改变程序,通过目标系统推断系统的行为;动态分析则以输入为中心,不同的输入对应不同的输出,根据这些输入输出判断系统行为。在软件逆向分析中,通常的做法是将静态分析和动态分析结合使用,不断推动分析深入,从而达到对系统功能的理解。例如静态分析可以有效分析程序的数据结构,常用地址,分支控制条件和循环等,但静态分析的推断可能并不十分准确,通过动态分析可以对静态分析推断进行验证、获取数据结构、地址中的运行值,跟踪程序流程等。通过动态分析获得更多信息达到对当前程序段的理解后就可以继续分析其他相关程序段,如此不断进行,使得分析工作不断向前推进。

静态分析和动态分析不但相互配合,也相互互补推进分析。动态分析时代码的运行和输入有关,满足条件的路径才执行,在实验过程中,有很多路径的条件总是不能得到满足,这部分的分析就要通过静态分析完成。静态分析是对程序所有路径的分析,它能够覆盖全部路径,这样静态分析和动态分析互相补充,最终得到对程序的全面理解。

4 总结和展望

本文主要研究了软件静态分析方法和动态分析方法以及在软件分析中的具体应用。在软件的逆向分析中,分析方法的使用是基础,同时也要借助资料和工具,尤其涉及领域专家方面,然而这些也是目前软件逆向工程发展还不成熟的地方,在工具支持通用性和自学习机制等方面还有待进一步发展和完善。

参考文献:

[1] 陈昊鹏. 软件逆向工程技术研究[D]. 西安:西北工业大学, 2001: 1- 4.

[2] 李伟华,李由. 实时软件逆向工程技术研究[J]. 西北工业大学学报, 2003(8).

[3] 袁望洪,陈向葵,谢涛,等. 逆向工程研究与发展[J]. 计算机科学, 1999(5).

[4] 严秀,李龙澍. 软件逆向工程技术研究[J]. 计算机技术与发展, 2009(4).

[5] 张晓锋. 软件逆向工程相关技术研究与实现[M]. 成都:电子科技大学出版社, 2007.