基于模糊层次分析法的软件保护有效性评价

李士群,王崑声,经小川,王潇茵,巴 峰,梁光成
(中国航天系统科学与工程研究院,北京 100000)

摘 要:软件频频遭到逆向破解和攻击,因此各种保护技术应运而生并得到长足发展。然而,保护有效性如何度量一直备受业界关注。通过对现状分析,当前有效性评价研究处于探索阶段,且多是针对单一保护技术的有效性进行评价,难以适用于当前多保护技术综合使用的保护方案。对此,在对软件逆向攻击和保护技术深入研究的基础上,建立了软件保护有效性综合评价的指标体系,并选用模糊层次分析法建立了软件保护有效性综合评价模型。首先,采用层次分析法构建层次化的评价指标体系结构,并确定指标权重;然后,采用模糊综合评价法将评价数据进行综合模糊运算,得到量化的评价结果;最后,运用建立的评价模型结合实际案例进行了评价研究,实验结果表明该模型具有一定的可行性和有效性。

关键词:软件逆向;软件保护;模糊层次分析法;保护有效性评价
中图分类号:N945.16 **文献标识码**:A **文章编号**:1673-629X(2018)04-0133-08
doi:10.3969/j.issn.1673-629X.2018.04.028

Evaluation for Effectiveness of Software's Protection Based on FAHP

LI Shi-qun, WANG Kun-sheng, JING Xiao-chuan, WANG Xiao-yin,
BA Feng, LIANG Guang-cheng
(China Aerospace Academy of System Science and Engineering, Beijing 100000, China)

Abstract: A series of protection technologies have emerged and developed rapidly as the software is reverse-cracked and attacked frequently. However, how to evaluate the effectiveness of software protection draws much attention of researchers and practitioners. The analysis of status shows that the current effectiveness evaluation is still in the exploratory stage and for single protection technology mostly, which is difficult to apply to protection scheme of multi-technology. For this, based on further studying of software reverse attack and software protection, we set up a comprehensive evaluation index system, and establish comprehensive assessment model by FAHP (fuzzy analytical hierarchy process). Firstly, the hierarchical index system is built by analytical hierarchy process for determining of index weight. Secondly, the fuzzy comprehensive evaluation method is used to conduct the comprehensive fuzzy operations to evaluation data, getting the quantitative evaluation result. At last, the established evaluation model is researched combined with a actual case. The experiments show that the model is feasible and valid certainly.

Key words: software reverse; software protection; FAHP; evaluation of effectiveness

0 引 言

为应对软件逆向、破解等攻击而造成的安全威胁,各种保护技术成了研究热点。然而,如何对软件保护的有效性进行综合评价,是研究者和软件开发方在设计和选择保护方案时十分关心的。科学客观的评价结果能够指导设计更为完善的保护方案,为保护方案的选择提供决策支持。

现有的评价方法多是从一个角度进行有效性评价,然而,科学的评价模型应该对软件保护和逆向进行综合考量。另外,保护方案往往会综合使用多种保护技术,单纯对某一种保护技术进行评价分析也是当前评价方法的局限之处。

文中对多保护技术综合使用的保护方案进行评价研究,在对软件逆向和保护深入研究的基础上建立了软件保护有效性评价的综合评价体系,选用模糊层次分析法建立了综合评价模型,并结合实际案例对模型的实际效果进行了验证。

收稿日期:2017-04-10 **修回日期**:2017-08-11 **网络出版时间**:2017-12-05
基金项目:国家自然科学基金(9141820013)
作者简介:李士群(1991-),男,硕士研究生,CCF 会员(74353M),研究方向为系统工程;王崑声,研究员,研究方向为系统工程;经小川,研究员,研究方向为软件安全。

1 相关现状

1.1 软件攻防现状

1.1.1 软件攻击现状

当前,软件攻击的目的通常是对程序逆向分析,进而破除软件的权限限制、窃取软件思想或者在其中嵌入恶意代码[1]。逆向分析是指从可执行程序出发,推导出软件产品设计原理的过程[2]。逆向分析可以分为文件获取、反汇编、动静态分析、软件篡改五个子行为序列[3]。逆向分析模型如图 1 所示。

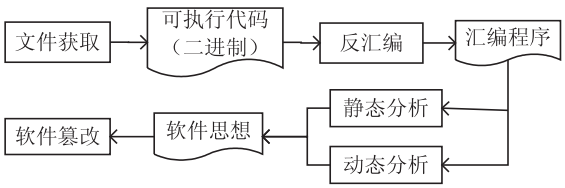

图 1 软件逆向过程

(1)文件获取:为实现攻击目的,攻击者首先需要获得攻击的目标程序。通常获取的是二进制可执行文件。

(2)反汇编:获取的二进制可执行文件难以分析,需要反汇编成相对容易理解的汇编程序,以便于后续分析。

(3)静态分析:静态分析是在程序不执行的状态下进行逆向分析的分析技术。通过控制流分析、数据流分析等技术对软件进行分析,对程序代码进行结构框架上的分析认识。

(4)动态分析:动态分析是指通过动态调试对目标程序进行分析的技术。与静态分析不同,动态分析是对目标程序的执行流程、执行结果实时跟踪,可以收集到更为详尽和真实的信息。

(5)软件篡改:在对目标程序分析完成之后,攻击者会将程序中的关键算法进行篡改,实现权限破解或为己所用等目的。

1.1.2 软件保护现状

为应对逆向分析造成的安全威胁,一系列的保护技术先后被提出并得到发展。目前,主流的保护技术有软件加密、软件混淆、反调试和防篡改等。

(1)软件加密是指将目标程序加密,在运行之前再解密还原的技术。软件加密能够使得目标程序不能直接反汇编,提高攻击者反汇编的难度。

(2)软件混淆即通过对目标程序进行重新组织,使得处理后的程序与原来相比在功能上变化不大,但在语义上更难理解,对应对动静态分析都有很好的效果。

(3)反调试是应对逆向分析进行动态调试的技术,现普遍基于“检测—响应”机制。即检测程序是否处于被调试状态,然后做出相应的响应[4]。

(4)防篡改是防止攻击者在破解程序关键思想之后对程序进行恶意篡改的技术。防篡改也是基于“检测—响应”的机制[5]。

另外,综合分析文献[6-8]可以看出,在设计保护方案时,往往会将上述技术综合使用以实现各项技术优势互补,进一步提高保护效果。

1.2 软件保护有效性评价现状

软件保护有效性评价即度量目标软件在施加保护措施后其所能应对逆向分析的能力。现有的评价方法主要分为两类:一是基于程序属性度量的评价;二是基于逆向分析角度的评价。

1.2.1 基于程序属性度量的评价

基于程序属性度量的评价是指提取目标程序的关键属性为指标,通过分析保护前后指标的变化来度量保护方案的有效性。

Collberg 等提出了强度、弹性、开销和隐蔽性 4 项针对软件混淆有效性的评价指标[9]。Collberg 的成果为代码混淆有效性评价研究奠定了基础,但没有给出这些指标具体的度量方法。

M. Dalla 将语义变化和保护有效性联系起来,提出比较混淆前后程序分析结果的不等性来描述软件混淆的有效性[10]。但 M. Dalla 没能指明语义的哪种变化对程序的理解难度有提高效果,因此这种评价方法尚不成熟。

赵玉洁等将逆向分析分为反汇编、控制流分析和数据流分析三个关键环节;提取表征程序属性的指标:指令执行率、控制流循环复杂度、扇入/扇出复杂度,通过保护前后指标的变化评价软件混淆的有效性[11]。该方法为软件混淆有效性评价提供了进步性思路,但局限于单一技术是该方法的不足之处。

1.2.2 基于逆向角度的评价

基于逆向角度的评价是指从逆向分析出发,根据逆向的成本或难度来度量保护方案的有效性。

M. Ceccato 等通过对混淆后的程序进行手动逆向实验,从实际逆向体验上验证混淆技术的有效性[12]。该研究分析了影响逆向分析的关键因素,对评价思路转移到逆向角度具有很好的启发意义。

Collberg 等提出从攻击建模角度对软件保护方案进行有效性评价的思路[13]。通过对逆向攻击建立攻击模型,提取逆向过程中的指标来评价软件保护的有效性,给有效性评价研究指引了新的方向。

王妮在 Collberg 的基础上,建立了攻击模型,并提出了攻击成本的评价指标[14]。建立的关键算法的逆向攻击模型能够描述整个逆向的先后过程、逆向状态信息及状态间转换的条件要素。在评价指标方面,将各逆向技术的攻击成本进行分级,并给每一等级赋予

单位开销值;通过逆向所需使用的技术来量化攻击成本,继而反映出保护有效性。该方法思路新颖,但逆向过程具有很高的灵活性,且逆向技术成本包括攻击者的体智耗费、机器损耗等难以量化的因素,因此,该方法的说服力还有待提高。

综合上述评价研究,当前软件保护有效性评价思路主要存在以下两方面的问题:

(1)现有评价方法多是针对软件混淆技术的有效性进行研究,其他如软件加密、反调试、防篡改等主流保护技术的有效性评价研究很少。

(2)针对单一保护技术有效性进行评价研究难以适用于当前综合多技术的保护方案,因此,现有评价研究的实用意义存在很大局限。

2　软件保护有效性综合评价指标体系

软件保护技术致力于提高逆向分析的难度。其中,软件加密提高反汇编的难度;软件混淆提高动静态分析的难度;反调试提高动态调试的难度;防篡改提高在理解程序思想后进行篡改的难度。因此,面向综合多技术保护方案的有效性评价可以从保护方案的防反汇编能力、语义混淆能力、反调试能力和防篡改能力 4 个方面进行研究。基于此,提取如下指标来度量多技术保护方案的保护有效性:

1. 防反汇编能力。

(1)加密算法强度:是指加密后,密文数据本身的信息特征十分微弱,单纯从密文出发解密出明文难以实现。

(2)加密算法多样性:是指采用的加密算法种类是否多样。多样的加密算法能够提高攻击者逆向分析的知识需求量和工作量,从而增强保护的效果。

(3)密钥隐蔽性:密钥存放的越隐蔽越能提高攻击者解密的难度,可从密钥的存储方式来评价密钥的隐蔽性。

(4)密钥强度:是指密钥难以被破译的程度,攻击者可能使用穷举法对密钥进行爆破,因此可以采用密钥的长度来评价密钥的强度。

(5)解密程序隐蔽性:目标程序运行之前需要解密,攻击者往往会跟踪分析解密程序来获取目标程序的明文,因此解密程序需要具有一定的隐蔽性。

(6)目标程序隐蔽性:是指目标程序加密之后是否能够与其他数据混合并以分布的形式存储,使得目标程序具有一定的隐蔽性。

2. 语义混淆能力。

(1)控制流循环复杂度:用控制流图描述程序的控制流程,然后采用图论的知识和计算方法来评估程序的执行复杂度[7]。控制流图中节点代表程序中的基本块,边代表程序中基本块之间的跳转。控制流循环复杂度记为 $V(G)$,其计算公式如下:

$$V(G) = e - n + 2 \tag{1}$$

其中,e 表示边的数量;n 表示节点的数量。

(2)数据流复杂度:数据流程越复杂,攻击者逆向分析的难度越大,也即保护效果越好。Henry 提出数据流复杂度度量方法,数据流复杂度记为 DC,计算公式如下:

$$DC = (\text{fan-in} \times \text{fan-out})^2 \tag{2}$$

其中,fan-in 为一个模块输入的数据流之和;fan-out 为一个模块输出的数据流之和。

(3)数据结构复杂度:程序中有大量的数据结构用来保存重要的数据信息,是逆向分析过程中理解程序中关键数据的重点。数据结构复杂度记为 SC,计算公式如下:

$$SC = IN \times D \tag{3}$$

其中,IN 为数据结构中的条目数;D 为数据结构深度。

(4)指令执行率:程序中的假指令会对静态分析造成干扰,从而提高静态分析的难度。指令执行率记为 IE,计算公式如下:

$$IE = Id/Is \tag{4}$$

其中,Is 为反汇编生成的指令条数;Id 为实际执行的指令条数。

3. 反调试能力。

(1)技术丰富度:是指目标程序中嵌入的检测和响应模块所采用技术的多样性。技术丰富度越高,就需要攻击者有更多的反调试知识和经验。

(2)模块数量:是指检测和响应模块的数量。数量越多,攻击者去除反调试机制的工作量就会越大,逆向分析的成本也就越高。

(3)响应强度:是指当检测到程序被调试之后响应方式的强度,即程序强制退出、程序自毁和执行假分支等方式所能起到的保护效果。

(4)模块隐蔽性:是指检测和响应模块与程序中其他模块具有相似性,难以被攻击者定位和识别。

(5)弹性:是指检测模块之间和响应模块之间的关联度,即其中的部分模块被发现并破坏后,整个反调试机制正常运行所受影响的程度。

4. 防篡改能力。

(1)技术丰富度:是指检测和响应模块所采用技术的多样性。技术丰富度越高,就需要攻击者有更多的反防篡改技术的知识和经验。

(2)模块数量:是指检测和响应模块的数量。数量越多,攻击者去除防篡改机制的工作量就越大,逆向分析的成本也就越高。

(3)响应机制强度:是指当检测到程序被篡改之后响应方式的强度,即程序自修复、程序自毁等方式所能起到的保护效果。

(4)模块隐蔽性:是指检测和响应模块与程序中其他模块具有相似性,难以被攻击者定位和识别。

(5)弹性:是指检测模块之间和响应模块之间的关联度,即其中的部分模块被发现并破坏后,整个防篡改机制正常运行所受影响的程度。

由于同是基于"检测—响应"机制,反调试和防篡改能力的评价指标在名称上相同,但评价内容不同,因此不会造成评价指标的冗余。

3　软件保护有效性综合评价模型

软件保护有效性评价的指标体系呈现出多层次、定性定量指标共存,并且指标具有模糊性的特点;文中采用模糊层次分析法对保护有效性进行评价研究。采用层次分析法建立指标体系的层次结构,判断矩阵法确定指标的权重;采用模糊综合评价法建立评语集,确定隶属度、构造评价矩阵和模糊综合评价。软件保护有效性综合评价模型如图 2 所示。

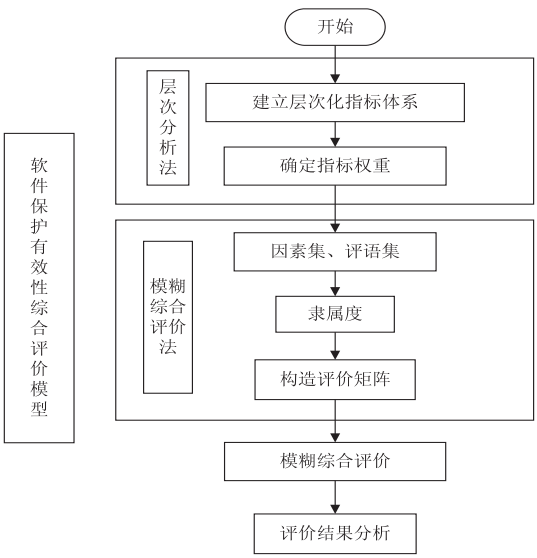

图 2　综合评价模型

3.1　保护有效性综合评价指标体系权重

相对于有效性评价目标,需要给各指标分配不同的权重。确定指标体系权重的步骤如下:

(1)建立指标体系的层次结构。

软件保护有效性综合评价指标体系分为目标层、准则层和指标层,层次结构如图 3 所示。

(2)构建比较判断矩阵。

通过调查问卷的方式,由专家对同一目标或准则下的评价指标进行两两重要性比较评判,并采用 1 ~ 9 分制标示重要度,从而构建判断矩阵。

对目标层 A 和准则层的元素 B_1,B_2,B_3,B_4,专家进行两两重要性比较评判,构造出判断矩阵 $\boldsymbol{B}$:

$$\boldsymbol{B}=\begin{bmatrix}1 & b_{12} & b_{13} & b_{14}\\ b_{21} & 1 & b_{23} & b_{24}\\ b_{31} & b_{32} & 1 & b_{34}\\ b_{41} & b_{42} & b_{43} & 1\end{bmatrix} \tag{5}$$

同理,即可构造出准则 B_1,B_2,B_3,B_4 下指标层的判断矩阵 $\boldsymbol{C}_1,\boldsymbol{C}_2,\boldsymbol{C}_3,\boldsymbol{C}_4$。

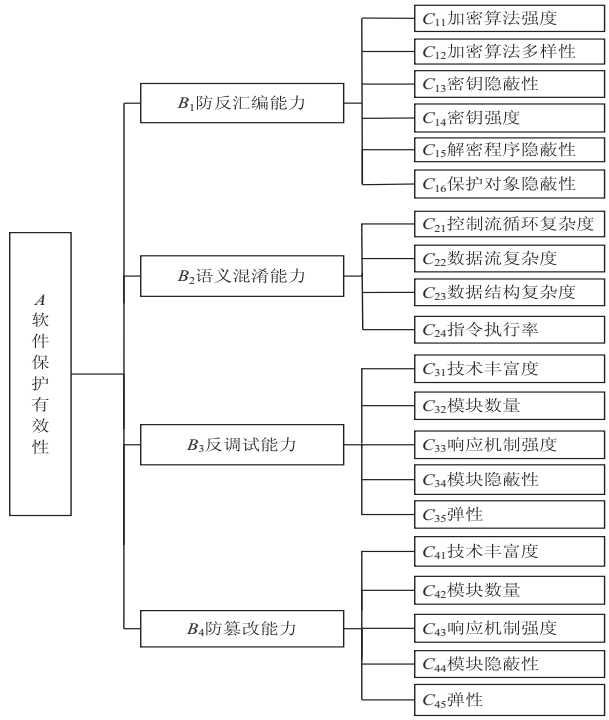

图 3　指标体系层次结构

(3)求解单层指标权重。

在目标层 A 下,计算判断矩阵 $\boldsymbol{B}$ 的最大特征根 $\lambda\max$ 和对应的特征向量 $\boldsymbol{\omega}$,特征向量 $\boldsymbol{\omega}$ 的元素即为指标 B_1,B_2,B_3,B_4 的权重。计算公式如下:

$$\boldsymbol{B\omega}=\lambda_{\max}\boldsymbol{\omega} \tag{6}$$

同理可得准则 B_1,B_2,B_3,B_4 下各指标的权重。

(4)一致性检验。

判断矩阵来源于专家的主观评判,判断结果难免有所差异。因此,需要对判断矩阵进行一致性检验。一致性检验方法如下:

一致性指标:

$$\mathrm{CI}=\frac{\lambda_{\max}-n}{n-1} \tag{7}$$

其中,CI 为一致性指标;$\lambda_{\max}$ 为判断矩阵的最大特征根;n 为判断矩阵阶数。

一致性比率:

$$\mathrm{CR}=\frac{\mathrm{CI}}{\mathrm{RI}} \tag{8}$$

其中,CR 为一致性比率,CR<0.10 时,判断矩阵具有满意的一致性;RI 为平均随机一致性指标。

3.2　保护有效性综合评价

3.2.1　确定因素集和评语集

(1)因素集。

$U=\{U_1,U_2,U_3,U_4\}=$ {防反汇编能力,语义混淆能力,反调试能力,防篡改能力};

$U_1=\{U_{11},U_{12},\cdots,U_{16}\}=$ {加密算法强度,密钥隐蔽性,密钥强度,加密算法多样性,解密程序隐蔽性,目标程序隐蔽性};

$U_2=\{U_{21},U_{22},U_{23},U_{24}\}=$ {控制流循环复杂度,数据流复杂度,数据结构复杂度,指令执行率};

$U_3=\{U_{31},U_{32},\cdots,U_{35}\}=$ {技术丰富度,模块数量,响应机制强度,模块隐蔽性,弹性};

$U_4=\{U_{41},U_{42},\cdots,U_{45}\}=$ {技术丰富度,模块数量,响应机制强度,模块隐蔽性,弹性}。

(2)评语集。

采用五级评价等级,评语集 $V=\{V_1,V_2,V_3,V_4,V_5\}=$ {高,较高,一般,较低,低},各等级对应分值区间为[80,100),[60,80),[40,60),[20,40),[0,20)。

3.2.2　确定隶属度和评价矩阵

采用调查问卷的方式,由专家对指标体系中的各指标进行等级评判。然后,对各指标隶属于各评价等级的频次进行统计,隶属度为:

$$r=\frac{n}{m} \tag{9}$$

其中,n 为评定指标为 V_i 的频次;m 为专家总数。

根据各指标的隶属度构造指标层的评价矩阵:

$$\boldsymbol{R}_i=\begin{bmatrix} r_{i11} & r_{i12} & \cdots & r_{i15} \\ r_{i21} & r_{i22} & \cdots & r_{i25} \\ \vdots & \vdots & \ddots & \vdots \\ r_{in1} & r_{in2} & \cdots & r_{in5} \end{bmatrix} \tag{10}$$

3.2.3　模糊综合评价

(1)一级模糊运算。

通过各指标层的模糊评价矩阵右乘权重向量对其进行模糊综合计算,一级模糊综合评价集为:

$$\boldsymbol{B}_i=(\omega_{i1}\ \omega_{i2}\ \cdots\ \omega_{in})\begin{bmatrix} r_{i11} & r_{i12} & \cdots & r_{i15} \\ r_{i21} & r_{i22} & \cdots & r_{i25} \\ \vdots & \vdots & \ddots & \vdots \\ r_{in1} & r_{in2} & \cdots & r_{in5} \end{bmatrix}=$$

$$(b_{i1}\ b_{i2}\ \cdots\ b_{i5}) \tag{11}$$

其中,i 为准则层指标的序号;n 为准则 $\boldsymbol{B}_i$ 下指标的个数;$\boldsymbol{B}_i$ 为综合模糊运算结果。

由此,可得到指标层的综合评价矩阵:

$$\boldsymbol{B}=\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{15} \\ b_{21} & b_{22} & \cdots & b_{25} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{i5} \end{bmatrix} \tag{12}$$

(2)二级模糊运算。

由一级模糊运算结果构成的模糊评价矩阵右乘权重向量来对准则层各因素进行模糊综合计算。二级模糊综合评价集为:

$$\boldsymbol{A}=(\omega_1\ \omega_2\ \cdots\ \omega_n)\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{15} \\ b_{21} & b_{22} & \cdots & b_{25} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{i5} \end{bmatrix}=$$

$$(a_1\ a_2\ \cdots\ a_5) \tag{13}$$

(3)综合评价结果的量化分析。

模糊运算得到的结果是评价目标在评语集上的隶属分布,与各等级分值相乘即可得到评价分值。

$$\boldsymbol{G}=\boldsymbol{A}\cdot\boldsymbol{V}=(a_1\ a_2\ \cdots\ a_5)\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} \tag{14}$$

其中,$\boldsymbol{A}$ 为评价结果在评语集上的隶属分布;$\boldsymbol{V}$ 为各评语集对应的量化分值;$\boldsymbol{G}$ 为最终评价分值。

4　应用案例研究

4.1　评价对象概述

为验证评价模型的有效性,选取针对人脸识别算法设计的保护方案为评价对象,实施了有效性评价的案例研究。其中,保护方案综合使用了软件加密、混淆、反调试和防篡改技术。其保护和执行流程如图 4 所示。

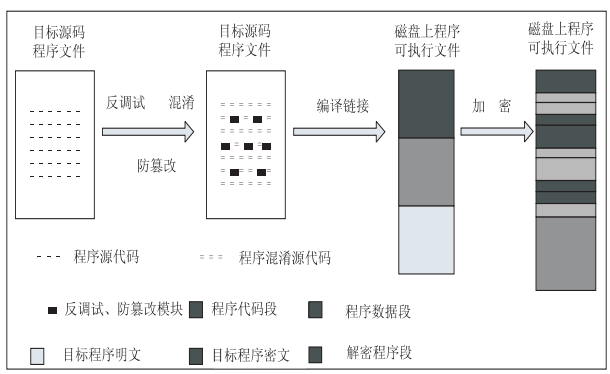

图 4　保护方案模型

将保护方案的关键设计和保护前后软件关键属性的变化提供给专家参考,由专家对各指标评价打分。该案例中关键评价指标的参考如表 1 所示。

表 1 保护方案关键评价指标参考说明

指标	说明
加密算法强度 C_{11}	采用 Gost 和 Blowfish 算法
加密算法多样性 C_{12}	采用了两种加密算法
密钥隐蔽性 C_{13}	密钥与数据混合存储
密钥强度 C_{14}	均为 256 比特
解密程序隐蔽性 C_{15}	解密程序分布存储于代码段
目标程序隐蔽性 C_{16}	切割成三块,分布存储于数据段
控制流循环复杂度 B_{21}	保护前 19,保护后 54
数据流复杂度 B_{22}	保护前 563,保护后 3186
数据结构复杂度 B_{23}	保护前 27,保护后 48
指令执行率 B_{24}	保护前后执行率为 74.683%,47.629%
技术丰富度 C_{31}	5 种检测技术,3 种响应方式
模块数量 C_{32}	9 个检测模块,5 个响应模块
响应机制强度 C_{33}	直接退出程序和死循环
模块隐蔽性 C_{34}	部分调用库函数,部分与普通程序相似
弹性 C_{35}	模块之间无关联
技术丰富度 C_{41}	2 种检测技术,3 种响应方式
模块数量 C_{42}	7 个检测模块,7 个响应模块
响应机制强度 C_{43}	直接退出程序、死循环和程序自毁
模块隐蔽性 C_{44}	与普通程序相似
弹性 C_{45}	模块之间无关联

4.2 评价实施

4.2.1 确定指标权重

相对于软件保护有效性,准则层判断矩阵如下:

$$\boldsymbol{B}=\begin{bmatrix}1 & 2.220\,0 & 2.459\,5 & 2.550\,8\\ 0.450\,3 & 1 & 1.219\,8 & 1.176\,1\\ 0.406\,6 & 0.819\,8 & 1 & 1\\ 0.390\,2 & 0.850\,5 & 1 & 1\end{bmatrix}$$

$\boldsymbol{W}_B=(0.202\,6,0.444\,3,0.174\,8,0.174\,7)$,并且 CR=0.000 2<0.1,通过一致性检验。

相对于防反汇编能力,指标层判断矩阵如下:

$$\boldsymbol{C}_1=\begin{bmatrix}1 & 0.440 & 1.148 & & 0.294 & 0.608\\ 2.267 & 1 & 1.782 & 0.308 & 0.461 & 0.724\\ 0.870 & 0.561 & 1 & 0.294 & 0.870 & 0.35\\ 3.393 & 3.245 & 3.33 & 1 & 0.488 & 0.659\\ 1.643 & 2.168 & 1.148 & 2.047 & 1 & 0.870\\ 3.393 & 1.379 & 2.825 & 1.515 & 1.148 & 1\end{bmatrix}$$

$\boldsymbol{W}_{C_1}=(0.081\,7,0.129\,3,0.089\,5,0.319\,7,0.173\,0,0.209\,8)$,且 CR=0.038 9<0.1,通过一致性检验。

相对于语义混淆能力,指标层判断矩阵如下:

$$\boldsymbol{C}_2=\begin{bmatrix}1 & 1.147\,8 & 1 & 1.551\,8\\ 0.870\,6 & 1 & 0.870\,6 & 1.148\,7\\ 1 & 1.148\,7 & 1 & 0.644\,4\\ 0.644\,4 & 0.870\,6 & 1.551\,8 & 1\end{bmatrix}$$

$\boldsymbol{W}_{C_2}=(0.285\,8,0.237\,2,0.233\,6,0.243\,3)$,并且 CR=0.027 8<0.1,通过一致性检验。

相对于反调试能力,指标层判断矩阵如下:

$$\boldsymbol{C}_3=\begin{bmatrix}1 & 0.922\,1 & 1.148\,7 & 2.352\,2 & 1.974\,4\\ 1.084\,5 & 1 & 1.148\,7 & 1.148\,7 & 1\\ 0.870\,6 & 0.870\,6 & 1 & 1.148\,7 & 1\\ 0.425\,1 & 0.870\,6 & 0.870\,6 & 1 & 1.319\,5\\ 0.506\,5 & 1 & 1 & 0.757\,9 & 1\end{bmatrix}$$

$\boldsymbol{W}_{C_3}=(0.271\,6,0.210\,8,0.188\,5,0.166\,0,0.163\,1)$,且 CR=0.024 7<0.1,通过一致性检验。

相对于防篡改能力,指标层判断矩阵如下:

$$\boldsymbol{C}_4=\begin{bmatrix}1 & 0.880\,7 & 1.102\,8 & 2.292\,5 & 1.839\,2\\ 1.135\,4 & 1 & 1.102\,9 & 1.127\,0 & 1\\ 0.906\,8 & 0.906\,7 & 1 & 1.181\,9 & 1\\ 0.436\,2 & 0.887\,3 & 0.846\,1 & 1 & 1.340\,0\\ 0.543\,7 & 1 & 1 & 0.746\,3 & 1\end{bmatrix}$$

$\boldsymbol{W}_{C_4}=(0.269\,6,0.224\,3,0.179\,5,0.163\,8,0.164\,0)$,且 CR=0.026 3<0.1,通过一致性检验。

4.2.2 确定隶属度和评价矩阵

防反汇编能力下的指标评价矩阵:

$$\boldsymbol{R}_1=\begin{bmatrix}0.224 & 0.376 & 0.280 & 0.072 & 0.048\\ 0.272 & 0.448 & 0.120 & 0.104 & 0.056\\ 0.024 & 0.136 & 0.488 & 0.200 & 0.152\\ 0.104 & 0.288 & 0.344 & 0.200 & 0.064\\ 0.032 & 0.072 & 0.128 & 0.416 & 0.352\\ 0.424 & 0.496 & 0.080 & 0 & 0\end{bmatrix}$$

语义混淆能力下的指标评价矩阵:

$$\boldsymbol{R}_2=\begin{bmatrix}0.120 & 0.249 & 0.150 & 0.449 & 0.032\\ 0.072 & 0.188 & 0.216 & 0.428 & 0.096\\ 0.024 & 0.345 & 0.235 & 0.324 & 0.075\\ 0.031 & 0.226 & 0.296 & 0.379 & 0.068\end{bmatrix}$$

反调试能力下的指标评价矩阵:

$$\boldsymbol{R}_3=\begin{bmatrix}0.296 & 0.360 & 0.248 & 0.096 & 0\\ 0 & 0.200 & 0.304 & 0.376 & 0.120\\ 0.320 & 0.432 & 0.248 & 0 & 0\\ 0.208 & 0.392 & 0.344 & 0.056 & 0\\ 0.440 & 0.344 & 0.112 & 0.064 & 0.040\end{bmatrix}$$

防篡改能力下的指标评价矩阵:

$$\boldsymbol{R}_4=\begin{bmatrix}0.048 & 0.216 & 0.408 & 0.256 & 0.072\\ 0.192 & 0.624 & 0.184 & 0 & 0\\ 0.104 & 0.360 & 0.248 & 0.224 & 0.064\\ 0 & 0.016 & 0.184 & 0.424 & 0.376\\ 0.408 & 0.512 & 0.080 & 0 & 0\end{bmatrix}$$

4.2.3 软件保护有效性模糊综合评价

(1)一级模糊综合评价。

指标层防反汇编能力 U_1 模糊综合评价如下:

$$\boldsymbol{B}_1 = (0.081\ 7\ \ 0.129\ 3\ \ 0.089\ 5\ \ 0.319\ 7\ \ 0.173\ 0\ \ 0.209\ 8) \cdot \begin{bmatrix} 0.224 & 0.376 & 0.280 & 0.072 & 0.048 \\ 0.272 & 0.448 & 0.120 & 0.104 & 0.056 \\ 0.024 & 0.136 & 0.488 & 0.200 & 0.152 \\ 0.104 & 0.288 & 0.344 & 0.200 & 0.064 \\ 0.032 & 0.072 & 0.128 & 0.416 & 0.352 \\ 0.424 & 0.496 & 0.080 & 0 & 0 \end{bmatrix} =$$

$$(0.145\ 2\ \ 0.419\ 9\ \ 0.242\ 8\ \ 0.142\ 2\ \ 0.046\ 2)$$

$$\boldsymbol{G}_1 = \boldsymbol{B}_1 \cdot \boldsymbol{V} = (0.145\ 2\ \ 0.419\ 9\ \ 0.242\ 8\ \ 0.142\ 2\ \ 0.046\ 2) \begin{pmatrix} 90 \\ 70 \\ 50 \\ 30 \\ 10 \end{pmatrix} = 59.329$$

指标层语义混淆能力 U_2 模糊综合评价如下:

$$\boldsymbol{B}_2 = (0.285\ 8, 0.237\ 2, 0.233\ 6, 0.243\ 3) \cdot \begin{bmatrix} 0.120 & 0.249 & 0.150 & 0.449 & 0.032 \\ 0.072 & 0.188 & 0.216 & 0.428 & 0.096 \\ 0.024 & 0.345 & 0.235 & 0.324 & 0.075 \\ 0.031 & 0.226 & 0.296 & 0.379 & 0.068 \end{bmatrix} =$$

$$(0.065\ \ 0.251\ 3\ \ 0.221\ 0\ \ 0.397\ 7\ \ 0.066\ 0)$$

$$\boldsymbol{G}_2 = \boldsymbol{B}_2 \cdot \boldsymbol{V} = (0.065\ \ 0.251\ 3\ \ 0.221\ 0\ \ 0.397\ 7\ \ 0.066\ 0) \begin{pmatrix} 90 \\ 70 \\ 50 \\ 30 \\ 10 \end{pmatrix} = 47.082$$

指标层反调试能力 U_3 模糊综合评价如下:

$$\boldsymbol{B}_3 = (0.271\ 6, 0.210\ 8, 0.188\ 5, 0.166\ 0, 0.163\ 1) \begin{bmatrix} 0.296 & 0.360 & 0.248 & 0.096 & 0 \\ 0 & 0.200 & 0.304 & 0.376 & 0.120 \\ 0.320 & 0.432 & 0.248 & 0 & 0 \\ 0.208 & 0.392 & 0.344 & 0.056 & 0 \\ 0.440 & 0.344 & 0.112 & 0.064 & 0.040 \end{bmatrix} =$$

$$(0.247\ 0\ \ 0.401\ 9\ \ 0.253\ 6\ \ 0.125\ 0\ \ 0.031\ 8)$$

$$\boldsymbol{G}_3 = \boldsymbol{B}_3 \cdot \boldsymbol{V} = (0.247\ 0\ \ 0.401\ 9\ \ 0.253\ 6\ \ 0.125\ 0\ \ 0.031\ 8) \begin{pmatrix} 90 \\ 70 \\ 50 \\ 30 \\ 10 \end{pmatrix} = 67.111$$

指标层防篡改能力 U_4 模糊综合评价如下:

$$\boldsymbol{B}_4 = (0.269\ 6, 0.224\ 3, 0.179\ 5, 0.163\ 8, 0.164\ 0) \cdot \begin{bmatrix} 0.048 & 0.216 & 0.408 & 0.256 & 0.072 \\ 0.192 & 0.624 & 0.184 & 0 & 0 \\ 0.104 & 0.360 & 0.248 & 0.224 & 0.064 \\ 0 & 0.016 & 0.184 & 0.424 & 0.376 \\ 0.408 & 0.512 & 0.080 & 0 & 0 \end{bmatrix} =$$

$$(0.139\ 7\ \ 0.425\ 2\ \ 0.240\ 0\ \ 0.182\ 1\ \ 0.092\ 4)$$

$$\boldsymbol{G}_4 = \boldsymbol{B}_4 \cdot \boldsymbol{V} = (0.139\ 7\ \ 0.425\ 2\ \ 0.240\ 0\ \ 0.182\ 1\ \ 0.092\ 4) \begin{pmatrix} 90 \\ 70 \\ 50 \\ 30 \\ 10 \end{pmatrix} = 60.734$$

(2)二级模糊综合评价。

准则层软件保护有效性 U 模糊综合评价如下:

$$\boldsymbol{A}=(0.202\,6,0.444\,3,0.174\,8,0.174\,7)\cdot\begin{bmatrix}0.145\,2 & 0.419\,9 & 0.242\,8 & 0.142\,2 & 0.046\,2\\ 0.065\,0 & 0.251\,3 & 0.221\,0 & 0.397\,7 & 0.066\,0\\ 0.247\,0 & 0.401\,9 & 0.253\,6 & 0.125\,0 & 0.031\,8\\ 0.139\,7 & 0.425\,2 & 0.240\,0 & 0.182\,1 & 0.092\,4\end{bmatrix}=$$

$$(0.326\,2\ \ 0.328\,0\ \ 0.135\,5\ \ 0.158\,6\ \ 0.060\,0)$$

$$\boldsymbol{G}=\boldsymbol{A}\cdot\boldsymbol{V}=(0.326\,2\ \ 0.328\,0\ \ 0.133\,5\ \ 0.158\,6\ \ 0.060\,0)\begin{pmatrix}90\\70\\50\\30\\10\end{pmatrix}=64.351$$

4.3 评价结果分析

该案例中软件保护有效性的评价结果为 64.351,处于较高等级,仍有一定提升空间。防反汇编能力、语义混淆能力、反调试能力和防篡改能力的评价结果分别为 59.329,47.082,67.111,60.734,防反汇编能力处于一般水平,但也十分接近较高水平;语义混淆能力处于一般水平;反调试能力处于较高水平;防篡改能力刚刚达到较高水平。

整体看来,该案例中保护方案对人脸识别算法有着较高的保护效果,如有更高需求的话,可以从四个方面继续提高保护强度。其中语义混淆能力对软件保护方案的有效性有着较高的权重,但并没有取得较高的保护效果。因此,在对保护方案进行改进的工作中可以着重提升语义混淆的能力。

5 结束语

面向多保护技术综合使用的保护方案,在对软件攻防深入研究的基础上,建立了保护有效性综合评价指标体系;基于模糊层次分析法建立了有效性综合评价模型。该模型不仅能够得出综合保护有效性的量化评价结果,而且能够评价保护方案各项能力的有效性,在给保护方案的选择提供决策支持的同时,还能够给保护方案的改进提供指导建议。

由于有效性评价研究处于探索阶段,文中建立的指标体系难免有不足之处,接下来还需要在指标体系方面进一步研究;另外在施加保护措施之后,将对目标程序的实时性、可靠性等的影响纳入评价模型也是下一步研究的重要内容。

参考文献:

[1] 薛芳芳,房鼎益,王怀军,等. 基于攻击目的的软件攻击分类方法研究[J]. 计算机应用与软件,2015,32(2):283-287.

[2] 严 秀,李龙澍. 软件逆向工程技术研究[J]. 计算机技术与发展,2009,19(4):20-24.

[3] 赵玉洁,汤战勇,王 妮,等. 代码混淆算法有效性评估[J]. 软件学报,2012,23(3):700-711.

[4] 刘 昕. 基于 Windows 内核反调试的软件保护系统[D]. 北京:北京邮电大学,2009.

[5] 汤战勇,郝朝辉,房鼎益,等. 基于进程级虚拟机的软件防篡改方法[J]. 华中科技大学学报:自然科学版,2016,44(3):65-70.

[6] 徐江陵. 基于反跟踪和自修改代码技术的软件保护系统设计[D]. 成都:电子科技大学,2012.

[7] 鲁晓成. 嵌入式软件保护关键技术研究与应用[D]. 武汉:武汉理工大学,2011.

[8] 赵路华. 软件保护技术研究[J]. 科技信息,2013(22):264.

[9] COLLBERG C,THOMBORSON C,LOW D. Manufacturing cheap,esilient and stealthy opaque constructs[C]//Proceedings 25th annual SIGPLAN SIGACT symposium on principles programming languages. New York,NY,USA:ACM,1998:184-196.

[10] DALLA M,GIACOBAZZI R. Control code obfuscation by abstract interpretation[C]//Proceedingof the 32nd international colloquium on automata,languages and programming. [s. l.]:[s. n.],2005:1325-1336.

[11] 赵玉洁. 面向逆向工程的代码混淆有效性研究与实践[D]. 西安:西北大学,2011.

[12] CECCATO M,PENTA M D,NAGRA J,et al. The effectiveness of source code obfuscation:an experimental assessment[C]//17th international conference on program comprehension. Vancouver,BC,Canada,IEEE,2009:178-187.

[13] COLLBERG C,DAVIDSON J,GIACOBAZZI R,et al. Toward digital asset protection[J]. IEEE Intelligent Systems,2011,26(6):8-13.

[14] 王 妮. 基于攻击建模的软件保护有效性评估方法研究[D]. 西安:西北大学,2012.

[15] MACABE T J. A complexity measure[J]. IEEE Transactions on Software Engineering,1976,2(4):308-320.