(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107103213 B
(45)授权公告日 2018.08.31

(21)申请号 201710179465.1
(22)申请日 2017.03.23
(65)同一申请的已公布的文献号
申请公布号 CN 107103213 A
(43)申请公布日 2017.08.29
(73)专利权人 中国航天系统科学与工程研究院
地址 100048 北京市海淀区阜成路16号
(72)发明人 薛惠锋 王潇茵 巴峰 张伟
葛慧 李宁 高金梁
(74)专利代理机构 中国航天科技专利中心 11009
代理人 臧春喜
(51)Int.Cl.
G06F 21/14(2013.01)
G06N 3/12(2006.01)

审查员 颜佳

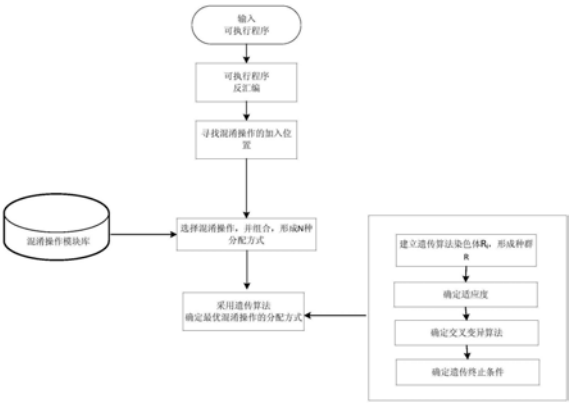
权利要求书1页 说明书4页 附图1页

(54)发明名称

一种基于遗传算法的软件代码混淆操作选择方法

(57)摘要

一种基于遗传算法的软件代码混淆操作选择方法，首先建立混淆操作模块库；然后对待混淆的可执行程序进行反汇编，从代码入口处开始寻找代码中的CALL、JUMP和XOR代码段，作为混淆操作的加入位置；从混淆操作模块库中随机选取多个混淆操作，并组合，形成N种混淆操作分配方式；最后采用遗传算法，对形成的N种混淆操作分配方式进行优化筛选，筛选出最优混淆操作分配方式。本发明方法能够综合考虑程序响应时间和代码混淆程度两个因素，对可执行程序的代码混淆操作进行筛选，在对软件代码进行有效充分混淆的前提下，尽量提高程序响应时间，极大的减少程序混淆对于程序执行效率的影响。



CN 107103213 B

1.一种基于遗传算法的软件代码混淆操作选择方法,其特征在于包括步骤如下:

(1)建立混淆操作模块库,所述混淆操作模块库由混淆操作构成,混淆操作包括控制流混淆操作、数据混淆操作以及综合控制流混淆和数据混淆的混淆操作;

(2)找到待混淆可执行程序的入口,对待混淆可执行程序进行反汇编;

(3)对反汇编后的代码进行遍历,寻找其中的JUMP、CALL、XOR代码,以此作为混淆操作的加入位置;

(4)从混淆操作模块库中随机选择若干个混淆操作,将所选的混淆操作分配给各个混淆操作的加入位置,形成N种混淆操作分配方式;

(5)采用遗传算法对N种混淆操作分配方式进行优化选择,选择最优的混淆操作分配方式作为本次软件代码混淆操作的选择结果。

2.根据权利要求1所述的一种基于遗传算法的软件代码混淆操作选择方法,其特征在于:所述步骤(5)的实现方法如下:

(2.1)确定混淆操作组合为遗传算法的特征,对N种混淆操作分配方式进行编码,形成N个染色体,N个染色体构成遗传算法的种群R,$R=\{R_1,R_2\cdots R_i\cdots R_N\}$;

(2.2)为每个染色体设定适应度,其中染色体R_i的适应度为F_i,F_i为经过染色体R_i混淆操作后的软件代码响应时间的倒数;

(2.3)每个染色体进行交叉或变异;

(2.4)当达到设定的遗传代数后,选择适应度最高的染色体R_H,作为最优的混淆操作分配方式。

3.根据权利要求2所述的一种基于遗传算法的软件代码混淆操作选择方法,其特征在于:所述步骤(2.1)中,染色体$R_i=\{H_1,H_2\cdots H_j\cdots H_m\}$,其中$H_j$表示第j个加入位置处分配的混淆操作组合。

4.根据权利要求2所述的一种基于遗传算法的软件代码混淆操作选择方法,其特征在于:所述步骤(2.3)中,对染色体R_i进行交叉的方法如下:

从R_i中随机选取一个特征值H_j,对于该特征值的两边特征H_{j-1}和H_{j+1}进行交叉处理,得到交叉之后的结果;

对染色体R_i进行变异的方法如下:

从R_i中随机选取一个特征值H_c,将H_c的混淆操作组合变异为混淆操作模块库中其他混淆操作的组合,实现变异,$c\in[1,m]$。

一种基于遗传算法的软件代码混淆操作选择方法

技术领域

[0001] 本发明属于信息安全技术领域，涉及一种基于遗传算法的软件代码混淆操作选择方法。

背景技术

[0002] 随着软件逆向技术的快速发展，软件出售之后，被非授权用户破解的可能性有所提高，一旦攻击者通常利用静态反汇编或动态调试等逆向工程技术对软件的可执行程序进行分析和破解，那么，将会盗取该软件核心算法等知识产权，还可能对软件和灌装参数表等进行篡改，谋取利益，甚至寻找软件中的安全漏洞，伺机利用。

[0003] 软件代码混淆技术是防止软件被非授权用户破解的一个有效手段，且在软件代码保护中逐渐发挥越来越大的作用。软件代码混淆将程序代码进行混淆处理，使得程序功能保持不变，但具体程序结构已经发生了比较大的变化，程序就转换成难以被他人理解和修改的程序，必须付出较大的代价才能够理解程序，这样就极大的保护了程序本身的知识产权。

[0004] 由于软件代码混淆对于软件本身的性能会有一定程度的影响，因此，要平衡性能和混淆程度，得到满意的代码混淆工作。当前研究者的工作主要集中在代码混淆方法和技术的选择和探索方面，甚少提出平衡软件可执行程序本身性能和代码混淆程度的优化方法。《基于垃圾代码的控制流混淆算法》一文中，曾经提出“当操作模块数过大时(大于某个值M)，则采用Hash函数选取操作模块，进行混淆操作，以限制混淆操作的次数，减小程序混淆后的时空开销”的方法，但是该方法随机性较强，在软件代码性能和混淆程度平衡的优化选择方面尚有欠缺。

[0005] 因此，需要提出一种优化算法，在进行软件代码混淆的过程中，对代码混淆操作进行优化选择，选择既能对软件代码进行有效的充分的混淆，又能够将对软件性能的影响降到最低的选择方法，提高软件代码混淆的功效。

发明内容

[0006] 本发明解决的技术问题是：克服现有技术的不足，提供一种基于遗传算法的软件代码混淆操作选择方法，实现了对软件代码有效的充分的混淆，同时将对软件性能的影响降到最低，提高了软件代码混淆的有效性。

[0007] 本发明的技术解决方案是：一种基于遗传算法的软件代码混淆操作选择方法，包括步骤如下：

[0008] (1)建立混淆操作模块库，所述混淆操作模块库由混淆操作构成，混淆操作包括控制流混淆操作、数据混淆操作以及综合控制流混淆和数据混淆的混淆操作；

[0009] (2)找到待混淆可执行程序的入口，对待混淆可执行程序进行反汇编；

[0010] (3)对反汇编后的代码进行遍历，寻找其中的JUMP、CALL、XOR代码，以此作为混淆操作的加入位置；

[0011] (4)从混淆操作模块库中随机选择若干个混淆操作，将所选的混淆操作分配给各个混淆操作的加入位置，形成N种混淆操作分配方式；

[0012] (5)采用遗传算法对N种混淆操作分配方式进行优化选择，选择最优的混淆操作分配方式作为本次软件代码混淆操作的选择结果。

[0013] 所述步骤(5)的实现方法如下：

[0014] (2.1)确定混淆操作组合为遗传算法的特征，对N种混淆操作分配方式进行编码，形成N个染色体，N个染色体构成遗传算法的种群R，$R=\{R_1, R_2 \cdots R_i \cdots R_N\}$；

[0015] (2.2)为每个染色体设定适应度，其中染色体R_i的适应度为F_i，F_i为经过染色体R_i混淆操作后的软件代码响应时间的倒数；

[0016] (2.3)每个染色体进行交叉或变异；

[0017] (2.4)当达到设定的遗传代数后，选择适应度最高的染色体R_H，作为最优的混淆操作分配方式。

[0018] 所述步骤(2.1)中，染色体$R_i=\{H_1, H_2 \cdots H_j \cdots H_m\}$，其中$H_j$表示第j个加入位置处分配的混淆操作组合，$H_j$可以为空。

[0019] 所述步骤(2.3)中，对染色体R_i进行交叉的方法如下：

[0020] 从R_i中随机选取一个特征值H_j，对于该特征值的两边特征H_{j-1}和H_{j+1}进行交叉处理，得到交叉之后的结果；

[0021] 对染色体R_i进行变异的方法如下：

[0022] 从R_i中随机选取一个特征值H_c，将H_c的混淆操作组合变异为混淆操作模块库中其他混淆操作的组合，实现变异，$c \in [1, m]$。

[0023] 本发明与现有技术相比的优点在于：

[0024] (1)本发明建立了包括多种混淆实现方式的混淆操作模块库，根据可执行程序中加入位置的不同，形成多种混淆操作分配方式，每种混淆操作分配方式均为不同类型不同实现方式的混淆操作组合，从而能够实现对待混淆可执行程序的充分有效混淆。

[0025] (2)本发明方法采用遗传算法对可执行程序混淆操作的多个混淆操作分配方式，进行选择、交叉和变异，优化选择出对可执行程序响应时间影响最小的混淆操作分配方式，避免了随机选择方式的不足，使得选择得到的软件代码混淆操作能够在充分进行代码混淆的前提下，尽可能小的影响代码本身的运行效率，将对软件性能的影响降到最低。

附图说明

[0026] 图1为本发明流程图。

具体实施方式

[0027] 如图1所示，本发明提出一种基于遗传算法的软件代码混淆操作选择方法，具体步骤包括：

[0028] (1)建立混淆操作模块库，混淆操作模块库由混淆操作构成，混淆操作包括控制流混淆操作、数据混淆操作以及综合控制流混淆和数据混淆的混淆操作。例如，控制流混淆操作包括在程序里面加入多余跳转、重新组织程序里面的控制流等，数据混淆操作包括对变量进行重组、对数值变量进行混合变换等。

[0029] (2)找到待混淆可执行程序的入口,对待混淆可执行程序进行反汇编。

[0030] (3)对反汇编得到的代码进行遍历,寻找其中的JUMP、CALL、XOR代码,以此作为混淆操作的加入位置。

[0031] 一种实现方式为:

[0032] (3.1)建立空的指令链表;

[0033] (3.2)判断第k行程序是否为JUMP、CALL、XOR指令,如果是,则将该代码段存入指令链表,如果不是,则进入步骤(3.3),k的初值为1;

[0034] (3.3)判断是否为最后一条指令,如果为否,则k的值加1,返回步骤(3.2);如果是最后一条指令,则进入步骤(4)。

[0035] (4)从混淆操作模块库中随机选择若干个混淆操作,并组合,将组合后的混淆操作分配给各个混淆操作的加入位置,形成N种混淆操作分配方式。

[0036] (5)采用遗传算法对N种混淆操作分配方式进行优化选择,选择最优的混淆操作分配方式作为本次软件代码混淆操作的选择结果。

[0037] 采用遗传算法对N种混淆操作分配方式进行优化选择的具体实现方法为:

[0038] 1)特征选择及编码

[0039] 针对遗传算法,确定混淆操作组合为遗传算法的特征。

[0040] 对N种混淆操作分配方式进行编码,形成N个染色体,N个染色体构成遗传算法的种群,$R_i=\{H_1,H_2\cdots H_j\cdots H_m\}$,其中$H_j$表示第j个加入位置处分配的混淆操作组合,$H_j$可以为空。$j\in[1,m]$。m表示待混淆可执行程序需要加入混淆操作的位置数量。$H_1,H_2\cdots H_j\cdots H_m$为遗传算法的特征。

[0041] N个染色体构成遗传算法的种群R,

[0042] $R=\{R_1,R_2\cdots R_i\cdots R_N\}$

[0043] 2)适应度确定

[0044] 采用遗传算法中的轮盘赌选择方法,每一个染色体占据虚拟轮盘中的一个扇区,各染色体占据的扇区面积正比于适应度数值,染色体R_i对应的适应度为F_i,F_i为经过染色体R_i混淆操作后的软件代码响应时间的倒数,软件代码响应时间越长,所对应的染色体适应度越低,反之亦然。

[0045] 3)特征交叉及变异

[0046] 采用简单交叉的方法,在每个染色体中随机设定一个交叉点,实行交叉的时候,该点前后两个特征进行交换,生成一个新的染色体。即,从染色体R_i中随机选取一个特征值H_j,对于特征值的两边特征H_{j-1}和H_{j+1}进行交叉处理,得到交叉之后的结果。

[0047] 变异过程中,随机选择染色体R_i一个特征值H_c,将H_c的混淆操作组合变异为混淆操作模块库中其他混淆操作组合,实现变异。$c\in[1,m]$。

[0048] 4)遗传终止条件

[0049] 预先设定遗传代数D,经过选择、交叉和变异操作,到规定代数D之后,选择适应度最高的染色体R_H,作为优化选择的结果。

[0050] (5)遗传算法结束后,R_H所对应的混淆操作分配方式即为本次软件代码混淆操作的选择结果。

[0051] 本发明首先建立混淆操作模块库,然后从反汇编后的代码中确定混淆操作的加入

位置，从混淆操作模块库中选择多个混淆操作，并组合，形成N种混淆操作分配方式；最后采用遗传算法，对形成的N种混淆操作分配方式进行优化筛选，筛选出适应程度最高的混淆操作分配方式。本发明对可执行程序的代码混淆操作进行筛选，能够综合考虑程序响应时间和代码混淆程度两个因素，在对软件代码进行有效充分混淆的前提下，尽量提高程序响应时间，极大的减少程序混淆对于程序执行效率的影响。

[0052] 本发明说明书中未作详细描述的内容属本领域技术人员的公知技术。

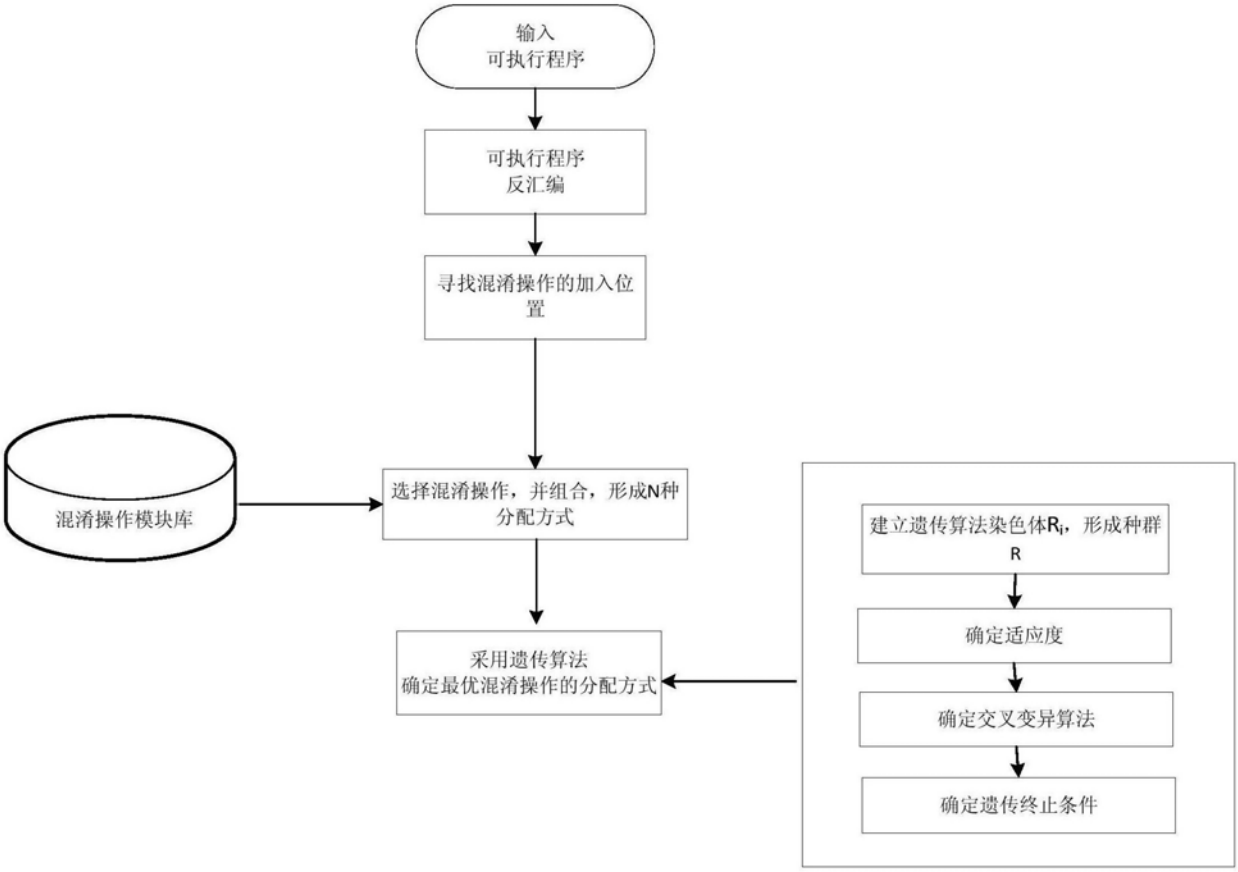

输入
可执行程序
可执行程序
反汇编
寻找混淆操作的加入位置
混淆操作模块库
选择混淆操作，并组合，形成N种分配方式
采用遗传算法
确定最优混淆操作的分配方式
建立遗传算法染色体R_i，形成种群R
确定适应度
确定交叉变异算法
确定遗传终止条件

图1