(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102034042 A
(43)申请公布日 2011.04.27

(21)申请号 201010585101.1

(22)申请日 2010.12.13

(71)申请人 四川大学
地址 610065 四川省成都市武侯区一环路南一段 24 号

(72)发明人 王俊峰 赵宗渠 白金荣 刘达富 方智阳

(74)专利代理机构 成都信博专利代理有限责任公司 51200
代理人 舒启龙

(51)Int.Cl.
G06F 21/00(2006.01)

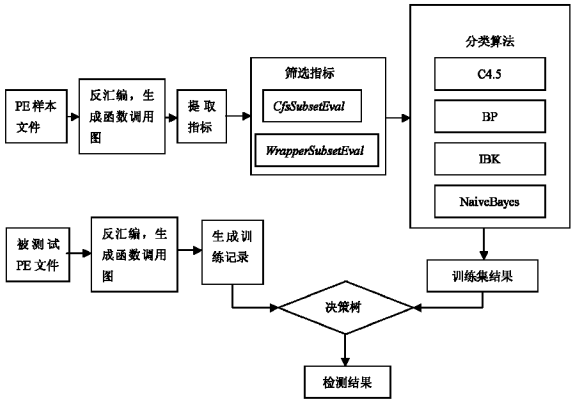
权利要求书 2 页 说明书 9 页 附图 1 页

(54)发明名称

基于函数调用关系图特征的恶意代码检测新方法

(57)摘要

一种基于函数调用关系图特征的恶意代码检测新方法。传统的特征码检测技术是通过局部特征对软件进行判断,其缺点是检测工具必须维护庞大的先验特征码以及这些检测方案缺少必要的稳定性和可靠性。本方法对于 Portable Executables(PE) 文件利用软件函数调用关系图,根据已有软件网络方面的研究成果,通过提取图特征信息来检测恶意代码。该方法主要过程分三步 :1、建立软件函数调用图 ;2、提取图中特征指标 ;3、通过有效的数据挖掘算法分类恶意软件和正常软件。本发明不但能够有效的检测出普通 zero-day 恶意代码,而且对于采用模糊和多态技术的恶意代码同样有较好的检测结果,对于蓄意攻击也保持良好的稳定性。



CN 102034042 A

1. 一种基于函数调用关系图特征的恶意代码检测方法，其特征是：分为以下 3 个阶段：

1、建立软件函数调用图；2、提取图中特征指标；3、通过数据挖掘算法分类恶意软件和正常软件；具体步骤如下：

1) 建立软件函数调用图：

装入 PE 格式文件，然后采用递归下降算法来处理文件，将文件进行反汇编操作，将文件转换成汇编代码，根据代码中的调用指令建立函数调用关系，然后将函数和这些调用关系保存在图这种数据结构中；

上述递归下降算法是通过控制流来逐条定位和分析指令及数据，根据顺序流指令，条件分支指令，无条件分支指令，函数调用指令和返回指令来定位后续指令的位置；

2) 提取图中特征指标：

2.1) 产生函数调用关系图特征集：

通过对输入文件处理，将文件中保存的函数调用图信息使用邻接链表的数据结构保存，然后在该结构中计算和统计定义特征集合 FeatureList 的值；文件处理步骤为：

a) 从输入文件中读入结点和边信息；

b) 向邻接链表中添加结点和边信息；统计结点类型及数量；直至读文件结束；然后，遍历所有连通有向子图，并统计结点和出、入度信息；遍历所有连通无向子图，并统计结点和出、入度信息；遍历定义的扩展连通图，并统计其信息，最后得到的计算和统计结果为产生的函数调用关系特征集；

2.2) 筛选指标以精确的反应函数调用图的特征：

三者择一地采用以下两种属性过滤算法以去除冗余属性区和分度较低的属性，合并相关较高的属性，进行指标的筛选；其一是，采用数据挖掘软件 Weka 中提供的 CfsSubsetEval 函数提供的属性过滤功能，衡量每一个属性的预测能力以及它们之间的冗余度，筛选出对预测目标关联度较高且相互之间低耦合的属性集合；其二是，选用与分类算法相关的属性筛选算法 WrapperSubsetEval，通过交叉验证的方法对属性进行衡量，最后得到该分类算法最有效的属性集合；

3) 通过数据挖掘算法分类恶意软件和正常软件：

3.1) 通过分类算法产生训练集结果：

该过程是生成训练集结果和决策树的过程；首先对大量的正常软件和恶意代码样本进行处理，将前面生成的函数关系调用图特征集的指标作为输入，通过机器学习过程，在选用的数据挖掘算法中得到训练集结果；上述数据挖掘算法采用基于决策树的 C4.5 算法，基于多层神经网络的 BP 算法，Lazy 分类算法中的 IBK 和贝叶斯分类算法中的 NaiveBayes 算法之一种；并按以下两种方式产生测试集结果和决策树：一种是按百分比划分样本和测试的数量，另一种是 m fold 交叉验证；训练集结果作为之后检测 PE 文件的依据；

3.2) 产生测试结果：

对于被测试的 PE 文件，经过产生函数调用图和提取图特征指标后，将这些指标格式化成符合检测要求的训练记录，用 arff 文件存储这些数据；这些训练记录文件作为输入，选用上述训练集结果就得到测试结果。

2. 根据权利要求 1 所述的基于函数调用关系图特征的恶意代码检测方法，其特征是：

所述数据挖掘算法优选为基于决策树的 C4.5 算法。

3. 根据权利要求 1 或 2 所述的基于函数调用关系图特征的恶意代码检测方法，其特征是：所述 m fold 交叉验证优选为 10fold 交叉验证。

4. 根据权利要求 1 或 2 所述的基于函数调用关系图特征的恶意代码检测方法，其特征是：所述筛选的指标为 28 个：

地址函数结点个数

外部函数结点个数；

内部函数结点个数；

导入名称结点个数；

入口结点个数；

有向连通图个数；

无向连通图个数；

入口结点序号；

有向连通图结点数绝对平均方差；

有向连通图平均度数；

入口结点后扩展图结点总数；

入口结点前扩展图结点总数；

入口结点有向连通图结点个数；

最大有向连通图结点数；

最大无向连通图结点数；

入口结点无向连通图结点数；

入口结点连通图是不是最大无向连通图；

结点最大度数；

结点平均度数；

度数的绝对平均方差；

孤立地址函数结点个数；

孤立外部函数结点个数；

孤立内部函数结点个数；

孤立导入名称结点个数；

终端地址子函数结点个数；

终端外部函数结点个数；

终端内部函数结点个数；

终端导入名称结点个数。

基于函数调用关系图特征的恶意代码检测新方法

技术领域

[0001] 本发明涉及计算机信息安全中的恶意软件检测，特别是一种新颖而实用的基于软件函数调用关系图特征的恶意代码检测方法。

背景技术

[0002] 随着计算机科学在社会各个领域的广泛应用，计算机软件的安全问题受到人们越来越多的关注。建立可信的软件系统成为维护计算机信息安全的一种有效手段，对于恶意代码的检测成为软件可信性分析的核心研究方向。

[0003] 传统的基于特征码的检测方式需要通过专用数据库来更新和维护事先提取相关特征码，通过扫描引擎查找软件的局部信息，并使用字符串匹配方法来对比这些信息和特征码的相似度，根据对比结果来得出检测结论。这种检测方法被广泛应用于现有的杀毒软件和系统防护软件中，属于比较成熟的技术，现在开发重点主要集中在提高代码的扫描速度和特征码提取的准确度。

[0004] 近年来提出的非特征码恶意代码检测方法中，有些是采用启发式分析或是基于软件行为来判断正常软件和恶意软件，这些方法在没有软件先验信息的情况下，对 zero-day 恶意软件有一定的检测效果。还有一些研究是通过在软件特殊结构属性信息来分类软件，比如通过提取PE(Portable Executables)文件的头部信息进行对比分类，在正常运行环境也取得了不错的检测结果。这些方法都是通过对软件的局部信息挖掘，期望得到能够将软件正确分类的指标集，然后用指标集检测软件。

[0005] 上述的恶意代码检测方法分别存在以下问题：

[0006] 第一，基于特征码的检测方法需要获得恶意代码的特征码，并将不断增加的特征码到用户端的数据库，对日益庞大的数据库维护成为使用者付出的代价。该检测方法最大的缺点是难以检测未知的恶意代码，用户不能够及时处理新的安全威胁。

[0007] 第二，采用启发式分析或是基于软件行为检测方法是通过获得代码的特殊局部信息对软件进行分类，但是对于使用模糊和多态的恶意代码，这种局部特征往往不固定，因此该检测方法在判断这些恶意代码时准确率不够。

[0008] 第三，使用标准格式信息来区分软件的检测方法，利用的是软件的外部描述信息，这些信息大多不直接涉及软件的行为，对于熟悉该方法的恶意代码设计者，能够通过对格式信息的特殊处理来大幅度降低该方法的检测效果。

发明内容

[0009] 本发明的目的在于提出并设计一种检出率高、稳定性好的基于软件函数调用关系图的恶意代码检测新方法。

[0010] 本发明的目的的是这样实现的：一种基于函数调用关系图特征的恶意代码检测方法，分为以下 3 个阶段：

[0011] 1、建立软件函数调用图；2、提取图中特征指标；3、通过数据挖掘算法分类恶意软

件和正常软件；具体步骤如下：

[0012] 1) 建立软件函数调用图：

[0013] 装入 PE 格式文件，然后采用递归下降算法来处理文件，将文件进行反汇编操作，将文件转换成汇编代码，根据代码中的调用指令建立函数调用关系，然后将函数和这些调用关系保存在图这种数据结构中；

[0014] 上述递归下降算法是通过控制流来逐条定位和分析指令及数据，根据顺序流指令，条件分支指令，无条件分支指令，函数调用指令和返回指令来定位后续指令的位置；

[0015] 2) 提取图中特征指标：

[0016] 2.1) 产生函数调用关系图特征集：

[0017] 通过对输入文件处理，将文件中保存的函数调用图信息使用邻接链表的数据结构保存，然后在该结构中计算和统计定义特征集合 FeatureList 的值；文件处理步骤为：

[0018] a) 从输入文件中读入结点和边信息；

[0019] b) 向邻接链表中添加结点和边信息；统计结点类型及数量；直至读文件结束；然后，遍历所有连通有向子图，并统计结点和出、入度信息；遍历所有连通无向子图，并统计结点和出、入度信息；遍历定义的扩展连通图，并统计其信息，最后得到的计算和统计结果为产生的函数调用关系特征集；

[0020] 2.2) 筛选指标以精确的反应函数调用图的特征：

[0021] 三者择一地采用以下两种属性过滤算法以去除冗余属性区和分度较低的属性，合并相关较高的属性，进行指标的筛选；其一是，采用数据挖掘软件 Weka 中提供的 CfsSubsetEval 函数提供的属性过滤功能，衡量每一个属性的预测能力以及它们之间的冗余度，筛选出对预测目标关联度较高且相互之间低耦合的属性集合；其二是，选用与分类算法相关的属性筛选算法 WrapperSubsetEval，通过交叉验证的方法对属性进行衡量，最后得到该分类算法最有效的属性集合；

[0022] 3) 通过数据挖掘算法分类恶意软件和正常软件：

[0023] 3.1) 通过分类算法产生训练集结果：

[0024] 该过程是生成训练集结果和决策树的过程；首先对大量的正常软件和恶意代码样本进行处理，将前面生成的函数关系调用图特征集的指标作为输入，通过机器学习过程，在选用的数据挖掘算法中得到训练集结果；上述数据挖掘算法采用基于决策树的 C4.5 算法，基于多层神经网络的 BP 算法，Lazy 分类算法中的 IBK 和贝叶斯分类算法中的 NaiveBayes 算法之一种；并按以下两种方式产生测试集结果和决策树：一种是按百分比划分样本和测试的数量，另一种是 m fold 交叉验证；训练集结果作为之后检测 PE 文件的依据；

[0025] 3.2) 产生测试结果：

[0026] 对于被测试的 PE 文件，经过产生函数调用图和提取图特征指标后，将这些指标格式化成符合检测要求的训练记录，用 arff 文件存储这些数据；这些训练记录文件作为输入，选用上述训练集结果就得到测试结果。

[0027] 上述数据挖掘算法优选为基于决策树的 C4.5 算法。

[0028] 上述 m fold 交叉验证优选为 10fold 交叉验证。

[0029] 上述筛选的指标为 28 个：

[0030] 地址函数结点个数

[0031] 外部函数结点个数；

[0032] 内部函数结点个数；

[0033] 导入名称结点个数；

[0034] 入口结点个数；

[0035] 有向连通图个数；

[0036] 无向连通图个数；

[0037] 入口结点序号；

[0038] 有向连通图结点数绝对平均方差；

[0039] 有向连通图平均度数；

[0040] 入口结点后扩展图结点总数；

[0041] 入口结点前扩展图结点总数；

[0042] 入口结点有向连通图结点个数；

[0043] 最大有向连通图结点数；

[0044] 最大无向连通图结点数；

[0045] 入口结点无向连通图结点数；

[0046] 入口结点连通图是不是最大无向连通图；

[0047] 结点最大度数；

[0048] 结点平均度数；

[0049] 度数的绝对平均方差；

[0050] 孤立地址函数结点个数；

[0051] 孤立外部函数结点个数；

[0052] 孤立内部函数结点个数；

[0053] 孤立导入名称结点个数；

[0054] 终端地址子函数结点个数；

[0055] 终端外部函数结点个数；

[0056] 终端内部函数结点个数；

[0057] 终端导入名称结点个数。

[0058] 本发明针对传统恶意代码检测方法中偏重于软件的局部或外部特征，对使用模糊和多态技术恶意代码检测效率不稳定的缺点，提出利用软件中更加稳定的函数调用关系来发掘软件的行为特征，使用图的形式来描述这些调用关系，进而利用软件网络和图论的相关知识对软件行为进行量化，通过指标分析和数据挖掘来实现恶意代码检测。

[0059] 本发明解决的关键问题在于：创新性的利用软件的函数调用关系图特征来判断软件行为，利用机器学习算法对量化的图特征指标进行筛选，通过数据挖掘的分类算法对样本处理得到软件分类的决策树，从而进行恶意代码检测。

附图说明

[0060] 图 1 是基于软件函数调用关系图特征的恶意代码检测方法的模型框图。

具体实施方式

[0061] 检测模型及基本思想：

[0062] 基于软件函数调用关系图的恶意代码检测方法处理的对象主要是 PE 格式的正常软件和恶意软件。软件工程中软件网络的理论认为，软件具有网络的拓扑结构，一般可以用图表示。在本方法中，将需要检测 PE 文件在函数级别用图结构来描述，图中的每个结点代表在文件中出现的函数，边代表函数之间的调用关系。与其他数据结构相比，图能够表达更加丰富的语义，且图论作为数学领域的一个重要分支，具有较长的研究历史及成熟完备的理论支持，基于图的数据挖掘技术主要用于发现图特征和软件分类之间的规律。本方法通过提取图中软件的特征信息来理解软件的行为，进而识别恶意代码。

[0063] 基于软件函数调用关系图的恶意代码检测方法使用的模型如图 1 所示，该模型分为 3 个阶段，①建立软件函数调用图，②提取图中特征指标，③通过有效的数据挖掘中分类算法对恶意软件和正常软件分类。

[0064] 建立软件函数调用图阶段需要对文件进行反汇编操作，将文件转换成汇编代码，根据代码中的调用指令建立函数调用关系，然后将函数和这些调用关系保存图这种数据结构中。这个阶段包含文件的装入、反汇编文件、检测函数之间的调用关系和生成函数调用图。

[0065] 我们处理的对象是 PE 格式的文件，由于一些恶意软件修改了 PE 文件的头部信息，在装入文件的时候必须考虑如何处理异常的装入信息。我们分析的主要数据是反汇编后得到的代码，因此反汇编的质量关系到数据的准确性，在本模型中采用递归下降算法来处理文件，这种算法的好处是能够有效的区分指令和数据，提高可信赖的结果。由于不同编译器生成的函数调用指令存在差异，在选取函数调用关系的时候，我们采用汇编指令中的 call 指令作为统一的调用标准，并生成调用图。为了在特征提取过程中更加方便，在本模型中函数调用图使用文件来保存。

[0066] 在提取函数调用图特定指标阶段，首先要制定能够反映图特征的指标，本模型中制定 47 个指标来量化图特征，这些指标是通过统计的方法、软件网络理论和图论中的一些算法计算得到，基本是能满足要求。为了防止在软件检测阶段使用数据挖掘分类时出现过度拟合的现象，我们使用了一些公开的特征过滤算法对 47 个指标进行了筛选，剔除了一些冗余指标和区分度不够的指标，合并了部分相联系的指标。指标的精简不但加快了模型整体的处理速度，而且在一定程度上提高了检测的准确率。

[0067] 在使用分类算法对文件进行分类操作阶段，首先需要建立机器学习结果，对收集的文件样本提取指标，选用一定的分类算法对样本进行处理生产训练结果，训练结果中包含了在机器学习中找到的最佳决策树。检测软件时，需要将文件按上述步骤提取图特征指标，依据决策树对软件作出判断结论。

[0068] 方法描述：

[0069] 相关概念及定义：

[0070] 在介绍具体的方法之前，首先给出相关概念及定义：

[0071] 定义一：函数的分类：对于反汇编后代码中的函数，依据在反汇编过程中的信息分为 4 类。第 1 类是由调用关系按内存地址命名的函数称为地址函数，第 2 类导入表中定义的函数，第 3 类是程序内部被识别的库函数，第 4 类是用户定义的函数；

[0072] 定义二：m 扩展子图：将内存地址相邻的 m 个连通子图合并成一个子图称为 m 扩展

子图。这样定义是根据程序局部性的原则，为了弥补静态反汇编中丢弃某些函数间接调用关系；

[0073] 定义三：m fold 交叉验证：在对文件进行分类时，将所有文件按随机方式平均分为 m 份，将其中 m-1 份作为机器学习的材料，而另外 1 份作为测试目标，这样进行 m 次，使每一份都能得到测试。

[0074] 1) 反汇编 PE 文件：

[0075] 本方法以静态分析文件为基础，需要将 PE 文件进行反汇编生成汇编格式的代码序列。采用反汇编引擎是利用递归下降算法来处理文件，该算法是通过控制流来逐条定位和分析指令及数据，它能够根据顺序流指令，条件分支指令，无条件分支指令，函数调用指令和返回指令来定位后续指令的位置。递归下降算法能够访问所有路径，将所有代码进行反汇编。作为一种基于控制流的算法，递归下降算法能够正确区分代码和数据，但是作为静态分析的缺点之一，对某些间接跳转指令的控制流处理不彻底，会忽略一些函数间的调用关系，对此我们在统一调用标准的前提的同时，在设计函数调用图指标时根据程序局部性的原理，做一些适当的处理；

[0076] 2) 提取函数调用图中特征：

[0077] 函数调用图反映了软件的功能，这些功能就是检测恶意代码的依据，提取函数调用图中的特征信息实际上是将这些功能通过图的指标来反映出来，也就是如何来度量软件功能的问题。

[0078] 现代度量理论属于数学的一个分支，度量是按照明确定义的规则，将数字或者符号赋予真实世界中实体的属性的过程，并通过这种方式来描述实体的属性，从而揭示其内在的意义。

[0079] 形式化地，度量是一个三元组 <Q，M，N>，其中：

[0080] ●经验关系系统 Q = <E，R>，E 为被度量实体（属性）的集合，R = $\{R_1, R_2, \cdots, R_n\}$ 为 E 上的关系集合；

[0081] ●数值关系系统 N = <S，P>，S 为数值或者符号集合，P = $\{P_1, P_2, \cdots, P_n\}$ 为 S 上的关系集合；

[0082] ●映射 M：Q → N，M(x) 表示实体 x 在被度量属性方面的度量值。

[0083] 上述定义中的经验关系系统是对被度量事物属性的描述与刻画，关系 R_i 必须能够真实、准确地反映被度量事物属性的性质。从数学的角度讲，关系 R_i 还可以被定义为被度量事物 E_i 上的运算，如果我们用一个映射 M 来给每一个 E_i 赋予一个实数的话，三者就构成了一个完整的度量。但是，这个映射必须满足一定的条件，即从 Q 到 N 的映射是同态的。

[0084] 函数调用图实际上是软件功能在语法上的一种抽象表示，即通过结构来发掘对应的功能，软件的功能是由多个子功能构成，表现为函数调用图的多个子图，提取函数调用图中特征就是通过指标来描述这些子图和子图之间的关系（即经验关系系统的 Q = <E，R>）。由于子功能和子图之间 m：n 的关系，所以在本阶段首先需要找到尽可能多的指标以及指标之间的关系（即数值关系系统 N = < S，P>）。函数调用关系图中定义的部分基本指标如图 2 表示。产生这些指标的过程描述如图 3 所示。

[0085] 函数调用图中的部分基本指标如下：

[0086] int sumSub；// 地址函数结点个数；

[0087] int exterSub ;// 外部函数结点个数 ;

[0088] int innerSub ;// 内部函数结点个数 ;

[0089] int remoteSub ;// 导入名称结点个数 ;

[0090] int stSub = 0 ;// 入口结点个数 ;

[0091] int sumGraphs ;// 有向连通图个数 ;

[0092] int sumNondirectGraphs ;// 无向连通图个数 ;

[0093] int startNodeId = -1 ;// 入口结点序号 ;

[0094] double cGraphsVar ;// 有向连通图结点数绝对平均方差 ;

[0095] double averageNodes ;// 有向连通图平均度数 ;

[0096] int extendGraphsNodes ;// 入口结点后扩展图结点总数 ;

[0097] int foreGraphsNodes ;// 入口结点前扩展图结点总数 ;

[0098] int startnodes = 0 ;// 入口结点有向连通图结点个数 ;

[0099] int maxGraphNodes ;// 最大有向连通图结点数 ;

[0100] int maxNondirectedGraph ;// 最大无向连通图结点数 ;

[0101] int startNondirectNodes ;// 入口结点无向连通图结点数 ;

[0102] boolean isStartMax ;// 入口结点连通图是不是最大无向连通图 ;

[0103] int maxDegree ;// 结点最大度数 ;

[0104] double averageDegree ;// 结点平均度数 ;

[0105] double degreeVar ;// 度数的绝对平均方差 ;

[0106] int abSub ;// 孤立地址函数结点个数 ;

[0107] int abExterSub ;// 孤立外部函数结点个数 ;

[0108] int abInnerSub ;// 孤立内部函数结点个数 ;

[0109] int abRemoteSub ;// 孤立导入名称结点个数 ;

[0110] int finalAdressSub ;// 终端地址子函数结点个数 ;

[0111] int finalExterSub ;// 终端外部函数结点个数 ;

[0112] int finalInnerSub ;// 终端内部函数结点个数 ;

[0113] int finalRemoteSub ;// 终端导入名称结点个数。

[0114] 筛选指标来精确的反应函数调用图的特征 :

[0115] 我们采用了若干种属性过滤算法，去除冗余属性区和分度较低的属性，合并相关较高的属性。例如本方法中采用数据挖掘软件 Weka 中提供的 CfsSubsetEval 函数提供的属性过滤功能，衡量每一个属性的预测能力以及它们之间的冗余度，筛选出对预测目标关联度较高且相互之间低耦合的属性集合。另外还选用了与分类算法相关的属性筛选算法 WrapperSubsetEval，它以指定的分类算法为参数，通过交叉验证的方法对属性进行衡量，最后得到该分类算法最有效的属性集合。与分类算法相关的属性筛选算法虽然需要耗费时间，但这种操作只是在创建训练集使用，对软件的检测时间影响不大。经过筛选后的指标，不但减轻了无效指标的干扰，通过降低数据的维度提高了分类算法在检测代码是的速度。

[0116] 产生函数调用关系图特征集的过程描述 :

[0117] 方法 :CreateFeatureList// 软件函数调用关系图特征创建方法。

[0118] 输入 :

[0119] ■函数调用图描述文件 F

[0120] 输出：

[0121] ■特征集合 FeatureList

[0122] 方法：

[0123] 通过对输入文件 F 处理，将文件中保存的函数调用图信息使用邻接链表的数据结构保存，然后在该结构中计算和统计定义特征集合 FeatureList 的值；

[0124] 调用 OperateGdl(File inputFile)；

[0125] procedure PatternsMining(inputFile)；

[0126] (1) FeatureList ← ∅;//FeatureList 为特征集合

[0127] (2) AllNodeList ← ∅;//AllNodeListt 为邻接链表

[0128] (3)repeat

[0129] (4) 从 inputFile 中读入结点和边信息

[0130] (5) 向 AllNodeList 中添加结点和边信息

[0131] (6) 统计结点类型及数量

[0132] (7)until 读文件结束

[0133] (8) 遍历所有连通有向子图，并统计结点和出、入度信息

[0134] (9) 遍历所有连通无向子图，并统计结点和出、入度信息

[0135] (10) 遍历定义的扩展连通图，并统计其信息

[0136] (11) 将计算和统计结果给 FeatureList 赋值

[0137] (12)return FeatureList；

[0138] 通过分类算法产生训练集结果

[0139] 该过程是生成训练集结果和决策树的过程。首先对大量的正常软件和恶意代码样本进行处理，按检测模型的前两步生成相应的指标，将这些指标作为输入，通过机器学习过程，在选用的数据挖掘算法中得到训练集结果。由训练样本生成训练集结果的过程中，本方法提供了基于决策树的 C4.5 算法，基于多层神经网络的 BP 算法（使用 Weka 提供的 MultilayerPerceptron 函数接口），Lazy 分类算法中的 IBK 和贝叶斯分类算法中的 NaiveBayes 算法。这些算法基本都能够达到较高的准确率。本方法提供了两种产生测试集结果的方式，一种是按百分比划分样本和测试的数量，另一种是m fold交叉验证，通过这两种方式都可以得到训练集结果和决策树，从中可以获知训练集的分类准确率，训练集结果可以作为之后检测 PE 文件的依据，在本模型中用 Weka 中使用的 arff 类型文件保存。图 4 表示了部分决策树结构。

[0140] 1) 产生测试结果

[0141] 对于被测试的 PE 文件，经过产生函数调用图和提取图特征指标后，将这些指标格式化成符合检测要求的训练记录，用 arff 文件存储这些数据。这些训练记录文件作为输入，选用前面提到的训练集结果就可以得到测试结果。

[0142] 决策树的部分结构：

[0143] SumofFinalExterSub <= 4

[0144] | SumofFinalExterSub <= 0

[0145] | | StartNondireeted6raph <= 0

[0146] | | | RemoteSub <= 0

[0147] | | | | AddressSub <= 2:begnign

[0148] | | | | AddressSub > 2:virus

[0149] | | | RemoteSub > 0:virus

[0150] 最后，由表 1 给出本发明方法与传统的基于特征码模式、最新的局部信息挖掘模式方法之间的简要对比与总结。

[0151] 表 1 本发明方法与传统方法间的对比总结

[0152]

检测方法	检测 zero-day 恶意代码	处理时间	准确率	样本维护	稳定性
传统基于特征码模式	不能	少	高	特征码数据率	较高
基于局部信息挖掘模式	能	较多	较高	数据挖掘后的决策树	一般
本发明方法	能	较多	较高	数据挖掘后的决策树	高

[0153]

[0154] 具体实施例：

[0155] 在表 2 中描述了本发明方法中采用的实验数据集的简要信息。该实验数据集正常文件是实验室中收集的 Windows XP sp3 中的系统文件和应用程序；恶意代码是从 Malfease datasets 网站下载的部分病毒文件。样本文件处理的过程包括反汇编，生成函数调用图，从其中提取图特征指标，将这些数据保存在 arff 类型文件中。在本发明方法验证过程采用 10 fold 交叉验证，在测试过程中使用了不同的分类算法。

[0156] 实验：

[0157] ●具体操作：分别采用 CfsSubsetEval 和 WrapperSubsetEval 筛选特征，然后使用 4 种分类算法 C4.5 算法、BP 算法、IBK 算法和 NaiveBayes 算法对数据进行 10fold 交叉测试，测试结果如表 3 所示。

[0158] 表 2 实验数据描述

[0159]

数据类型	数据名称	数量
PE 文件	正常文件	2070
	恶意代码	2051

[0160] 结果分析：由于 WrapperSubsetEval 筛选特征时，是根据筛选剩余特征都是根据相对算法组合而成，因此检测的准确率要远高于 CfsSubsetEval 方式。在所有的分类算法中，C4.5 算法得到的结果最好，因此在检测恶意代码时推荐使用该算法。

[0161] 表 3 实验结果描述

[0162]

分类算法	C4.5 算法		BP 算法		IBK 算法		NaiveBayes 算法		class
测试结果	TP Rate	FP Rate	TP Rate	FP Rate	TP Rate	FP Rate	TP Rate	FP Rate	
CfsSubsetEval	0.897	0.158	0.828	0.176	0.89	0.183	0.932	0.475	Virus
	0.842	0.103	0.824	0.172	0.817	0.11	0.525	0.068	begnign
	0.87	0.131	0.826	0.174	0.854	0.148	0.734	0.277	Weighted Avg
WrapperSubsetEval	0.898	0.075	0.854	0.169	0.896	0.108	0.77	0.186	Virus
	0.925	0.102	0.831	0.146	0.892	0.104	0.814	0.23	begnign
	0.911	0.088	0.843	0.158	0.894	0.106	0.791	0.208	Weighted Avg

[0163] 注 :TP Rate 指恶意代码被正确识别的准确率 ;FP Rate 指正常文件被当做恶意代码的误判率 ;Weighted Avg 指加权后的平均值。

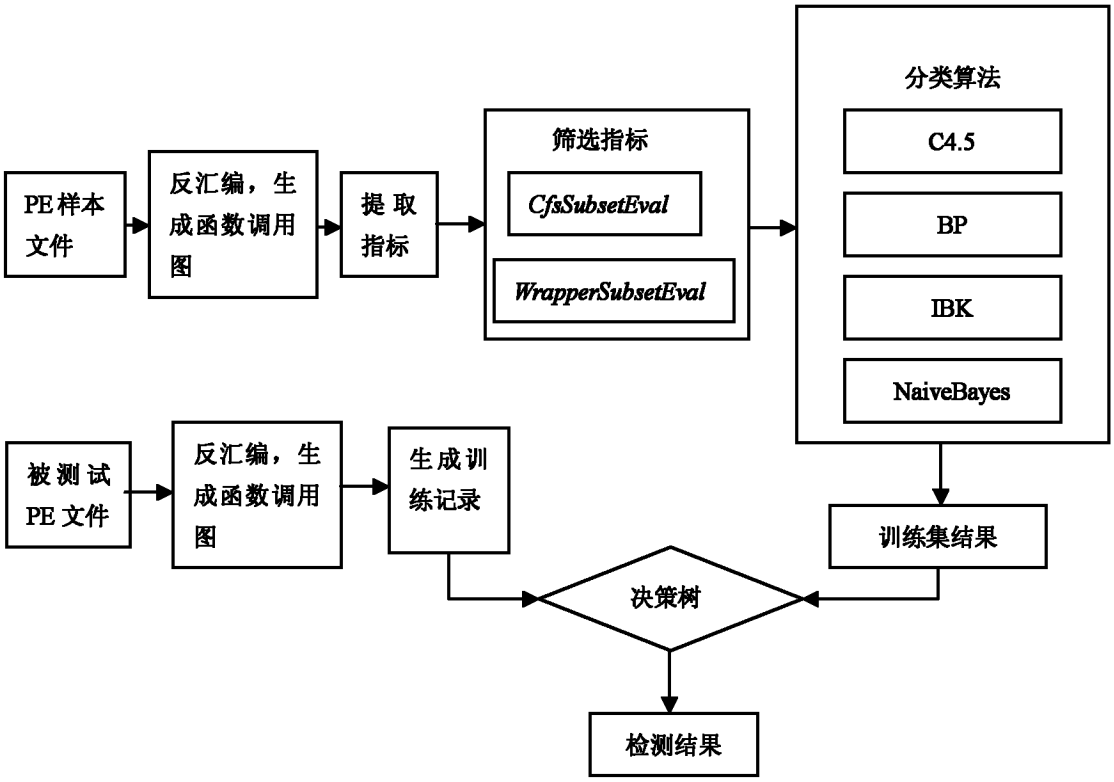

PE 样本文件
反汇编，生成函数调用图
提 取指标
筛选指标
CfsSubsetEval
WrapperSubsetEval
分类算法
C4.5
BP
IBK
NaiveBayes
被测试 PE 文件
反汇编，生成函数调用图
生成训练记录
训练集结果
决策树
检测结果

图 1