(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 110868405 A
(43)申请公布日 2020.03.06

(21)申请号 201911071345.5

(22)申请日 2019.11.05

(71)申请人 南方电网数字电网研究院有限公司
地址 511458 广东省广州市南沙区丰泽东路106号城投大厦1301房(自编1301-12159)

(72)发明人 梁志宏 胡朝辉 陈佳捷 罗强 高健 伍思廉 郑伟文 吴佩泽 彭伯庄 王金贺 陈鹏

(74)专利代理机构 广州华进联合专利商标代理有限公司 44224
代理人 曹瀚青

(51)Int.Cl.
H04L 29/06(2006.01)
G06F 21/56(2013.01)

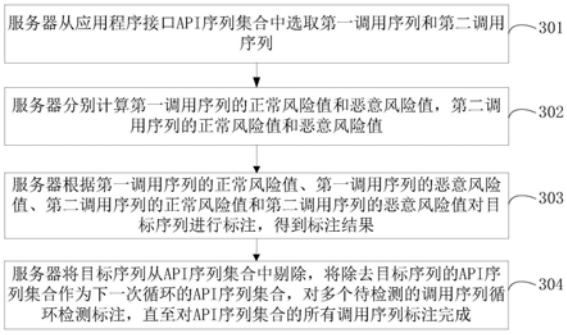
权利要求书2页 说明书13页 附图4页

(54)发明名称

恶意代码检测方法、装置、计算机设备及存储介质

(57)摘要

本申请公开了一种恶意代码检测方法、装置、计算机设备及存储介质，涉及信息安全技术领域。该方法中，网络设备的服务器可以从应用程序接口API序列集合中选取第一调用序列和第二调用序列，分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，根据第一调用序列的正常风险值和恶意风险值、第二调用序列的正常风险值和恶意风险值对目标序列进行标注，得到标注结果；将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。本申请技术方案可以提高对恶意代码的处理效率。



CN 110868405 A

1.一种恶意代码检测方法,其特征在于,所述方法包括:

从应用程序接口API序列集合中选取第一调用序列和第二调用序列,所述API序列集合包括多个待检测的调用序列;

分别计算所述第一调用序列的正常风险值和恶意风险值,所述第二调用序列的正常风险值和恶意风险值,所述正常风险值表示调用序列中存在恶意代码的概率;所述恶意风险值表示调用序列中不存在恶意代码的概率;

根据所述第一调用序列的正常风险值、所述第一调用序列的恶意风险值、所述第二调用序列的正常风险值和所述第二调用序列的恶意风险值对目标序列进行标注,得到标注结果,所述目标序列为所述第一调用序列或者所述第二调用序列,所述标注结果包括存在恶意代码或者不存在恶意代码;

将所述目标序列从所述API序列集合中剔除,将除去所述目标序列的API序列集合作为下一次循环的API序列集合,对所述多个待检测的调用序列循环检测标注,直至对所述API序列集合的所有调用序列标注完成。

2.根据权利要求1所述的方法,其特征在于,所述计算所述第一调用序列的正常风险值和恶意风险值,包括:

获取恶意样本集和正常样本集,所述恶意样本集中包括多个已知含有恶意代码的恶意调用序列;所述正常样本集中包括多个已知不含有恶意代码的正常调用序列;

针对所述恶意样本集中的每个所述恶意调用序列,分别计算所述第一调用序列与所述恶意调用序列的恶意相似度;针对所述正常样本集中的每个所述正常调用序列,分别计算所述第一调用序列与所述正常调用序列的正常相似度;

获取各所述恶意调用序列对应的恶意预测值、各所述正常调用序列对应的正常预测值以及所述第一调用序列的预测值;

根据所述恶意预测值、所述恶意相似度和所述第一调用序列的预测值计算所述第一调用序列的所述恶意风险值;根据所述正常预测值、所述正常相似度和所述第一调用序列的预测值计算所述第一调用序列的所述正常风险值。

3.根据权利要求1所述的方法,其特征在于,所述根据所述第一调用序列的正常风险值和恶意风险值,所述第二调用序列的正常风险值和恶意风险值对目标序列进行标注,得到标注结果,包括:

从所述第一调用序列的正常风险值、所述第一调用序列的恶意风险值、所述第二调用序列的正常风险值和所述第二调用序列的恶意风险值中选择出最小的风险值;

将所述最小的风险值对应的调用序列确定为所述目标序列;

根据所述最小的风险值对所述目标序列进行标注,得到标注结果。

4.根据权利要求3所述的方法,其特征在于,所述根据所述最小的风险值对所述目标序列进行标注,得到标注结果,包括:

当所述最小的风险值为所述第一调用序列的正常风险值或者所述第二调用序列的正常风险值时,所述标注结果为所述目标序列不存在恶意代码;

当所述最小的风险值为所述第一调用序列的恶意风险值或者所述第二调用序列的恶意风险值时,所述标注结果为所述目标序列存在恶意代码。

5.根据权利要求1所述的方法,其特征在于,所述从所述API序列集合中选取第一调用

序列和第二调用序列,包括:

分别计算所述API序列集合中每两个所述待检测的调用序列的汉明距离;

选取汉明距离最大的两个调用序列作为所述第一调用序列和所述第二调用序列。

6.根据权利要求1所述的方法,其特征在于,所述从所述API序列集合中选取第一调用序列和第二调用序列之前,所述方法还包括:

在虚拟沙箱中运行接收到的待检测的目标文件,获取所述目标文件的API函数所对应的调用序列;

对各所述调用序列,获得所述调用序列的特征向量,形成所述API序列集合。

7.根据权利要求1所述的方法,其特征在于,所述将除去所述目标序列的API序列集合作为下一次循环的API序列集合,包括:

当所述下一次循环的API序列集合只包括一个待检测的调用序列时,停止检测;

向人工检测终端发送检测指令,所述检测指令用于指示对所述下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

8.一种恶意代码检测装置,其特征在于,所述装置包括:

调用序列选取模块,用于从应用程序接口API序列集合中选取第一调用序列和第二调用序列,所述API序列集合包括多个待检测的调用序列;

风险计算模块,用于分别计算所述第一调用序列的正常风险值和恶意风险值,所述第二调用序列的正常风险值和恶意风险值,所述正常风险值表示调用序列中存在恶意代码的概率;所述恶意风险值表示调用序列中不存在恶意代码的概率;

标注模块,用于根据所述第一调用序列的正常风险值、所述第一调用序列的恶意风险值、所述第二调用序列的正常风险值和所述第二调用序列的恶意风险值对目标序列进行标注,得到标注结果,所述目标序列为所述第一调用序列或者所述第二调用序列,所述标注结果包括存在恶意代码或者不存在恶意代码;

循环处理模块,用于将所述目标序列从所述API序列集合中剔除,将除去所述目标序列的API序列集合作为下一次循环的API序列集合,对所述多个待检测的调用序列循环检测标注,直至对所述API序列集合的所有调用序列标注完成。

9.一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述的方法的步骤。

10.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的方法的步骤。

恶意代码检测方法、装置、计算机设备及存储介质

技术领域

[0001] 本申请涉及信息安全技术领域，特别是涉及一种恶意代码检测方法、装置、计算机设备及存储介质。

背景技术

[0002] 网络攻击是利用网络信息系统存在的漏洞和安全缺陷对网络设备的信息系统和数据资源进行攻击的行为，具体的，网络攻击可以篡改被攻击的网络设备的权限，从而窃取文件；还可以使被攻击的网络设备拒绝服务，以致用户无法正常使用网络设备，从而给用户带来巨大损失。

[0003] 现有技术，提供了一种检测网络设备接收到的目标文件中是否存在恶意代码的检测方法，该方法是：获取待检测的目标文件的动态动作信息，动态动作信息包括目标文件运行后产生的动作信息以及访问信息，当动作信息不能通过安全基线审核时，或者访问信息指向网络设备的核心单元时，判断该目标文件中存在恶意代码。

[0004] 然而，当目标文件中存在恶意代码时，对目标文件的源代码进行再次检测以确定恶意代码所在位置，需要花费额外的时间和人工，导致对恶意代码的处理效率较低。

发明内容

[0005] 基于此，有必要针对上述存在的不能确定出的恶意代码在目标文件中的具体位置的问题，提供一种恶意代码检测方法、装置、计算机设备及存储介质。

[0006] 第一方面，本申请实施例提供了一种恶意代码检测方法，该方法包括：

[0007] 从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列；

[0008] 分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率；

[0009] 根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码；

[0010] 将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0011] 在其中一个实施例中，计算第一调用序列的正常风险值和恶意风险值，包括：

[0012] 获取恶意样本集和正常样本集，恶意样本集中包括多个已知含有恶意代码的恶意调用序列；正常样本集中包括多个已知不含有恶意代码的正常调用序列；

[0013] 针对恶意样本集中的每个恶意调用序列，分别计算第一调用序列与恶意调用序列的恶意相似度；针对正常样本集中的每个正常调用序列，分别计算第一调用序列与正常调

用序列的正常相似度；

[0014] 获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值；

[0015] 根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值。

[0016] 在其中一个实施例中，根据第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值对目标序列进行标注，得到标注结果，包括：

[0017] 从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值；

[0018] 将最小的风险值对应的调用序列确定为目标序列；

[0019] 根据最小的风险值对目标序列进行标注，得到标注结果。

[0020] 在其中一个实施例中，根据最小的风险值对目标序列进行标注，得到标注结果，包括：

[0021] 当最小的风险值为第一调用序列的正常风险值或者第二调用序列的正常风险值时，标注结果为目标序列不存在恶意代码；

[0022] 当最小的风险值为第一调用序列的恶意风险值或者第二调用序列的恶意风险值时，标注结果为目标序列存在恶意代码。

[0023] 在其中一个实施例中，从API序列集合中选取第一调用序列和第二调用序列，包括：

[0024] 分别计算API序列集合中每两个待检测的调用序列的汉明距离；

[0025] 选取汉明距离最大的两个调用序列作为第一调用序列和第二调用序列。

[0026] 在其中一个实施例中，从API序列集合中选取第一调用序列和第二调用序列之前，该方法还包括：

[0027] 在虚拟沙箱中运行接收到的待检测的目标文件，获取目标文件的API函数所对应的调用序列；

[0028] 对各调用序列，获得调用序列的特征向量，形成API序列集合。

[0029] 在其中一个实施例中，将除去目标序列的API序列集合作为下一次循环的API序列集合，包括：

[0030] 当下一次循环的API序列集合只包括一个待检测的调用序列时，停止检测；

[0031] 向人工检测终端发送检测指令，检测指令用于指示对下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

[0032] 第二方面，本申请实施例提供了一种恶意代码检测装置，该装置包括：

[0033] 调用序列选取模块，用于从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列；

[0034] 风险计算模块，用于分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率；

[0035] 标注模块，用于根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第

二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码；

[0036] 循环处理模块，用于将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0037] 第三方面，提供了一种计算机设备，包括存储器和处理器，该存储器存储有计算机程序，该计算机程序被该处理器执行时实现上述第一方面的方法的步骤。

[0038] 第四方面，提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现上述第一方面的方法的步骤。

[0039] 本申请实施例提供的技术方案带来的有益效果至少包括：

[0040] 网络设备的服务器(以下简称服务器)可以从应用程序接口API(英文：Application Programming Interface，缩写:API)序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列。服务器可以分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，其中，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率。服务器可以根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，其中目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码。服务器可以将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。由此可知，本申请实施例中，网络设备的服务器对所有调用序列进行标注，这样就可以确定多个待检测的调用序列中的每个调用序列存在恶意代码或者不存在恶意代码，从而在确定目标文件中是否存在恶意代码的过程中可以直接确定恶意代码存在于哪个调用序列中，用户可以直接对存在恶意代码的调用序列进行处理，相比于现有技术，提高了对恶意代码的处理效率。

附图说明

[0041] 图1为本申请实施例提供的恶意代码检测方法的实施环境的示意图；

[0042] 图2为本申请实施例提供的恶意代码检测方法的另一种实施环境的示意图；

[0043] 图3为本申请实施例提供的一种恶意代码检测方法的流程图；

[0044] 图4为本申请实施例提供的另一种恶意代码检测方法的流程图；

[0045] 图5为本申请实施例提供的另一种恶意代码检测方法的流程图；

[0046] 图6为本申请实施例提供的另一种恶意代码检测方法的流程图；

[0047] 图7为本申请实施例提供的一种恶意代码检测装置的框图。

具体实施方式

[0048] 为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

[0049] 随着计算机技术的发展，网络设备的使用越来越广泛，网络设备之间传输的文件的数量激增。其中，部分文件有可能被嵌入恶意代码，恶意代码是指故意编制或设置的、对网络或系统会产生威胁或潜在威胁的计算机代码，例如：计算机病毒、特洛伊木马等等。这些恶意代码会进行匿名广告推送、静默下载软件，甚至偷偷扣费等行为，当网络设备打开携带有恶意代码的文件时，可能会受到网络攻击。网络攻击是利用网络信息系统存在的漏洞和安全缺陷对网络设备的信息系统和数据资源进行攻击的行为，具体的，网络攻击可以篡改被攻击的网络设备的权限，从而窃取文件；还可以使被攻击的网络设备拒绝服务，以致用户无法正常使用网络设备，从而给用户带来巨大损失。

[0050] 现有技术，提出了一种恶意代码的检测方法，该方法是对网络设备接收到的目标文件进行检测，获取待检测的目标文件的动态动作信息，动态动作信息包括目标文件运行后产生的动作信息以及访问信息，当动作信息不能通过安全基线审核时，或者访问信息指向网络设备的核心单元时，判断该目标文件中存在恶意代码。但是该方法不能在确定目标文件存在恶意代码时直接确定出恶意代码所在的位置，因此，当目标文件中存在恶意代码时，需要对目标文件的源代码进行再次检测以确定恶意代码所在位置，并对恶意代码进行处理。

[0051] 上述方法，当目标文件中存在恶意代码时，对目标文件的源代码进行再次检测以确定恶意代码所在位置，需要花费额外的时间和人工，导致对恶意代码的处理效率较低。

[0052] 本申请实施例提供的恶意代码检测方法、装置、计算机设备及存储介质，可以提高对恶意代码的处理效率。该方法中，网络设备的服务器(以下简称服务器)可以从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列。服务器可以分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，其中，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率。服务器可以根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，其中目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码。服务器可以将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。由此可知，本申请实施例中，网络设备的服务器对所有调用序列进行标注，这样就可以确定多个待检测的调用序列中的每个调用序列存在恶意代码或者不存在恶意代码，从而在确定目标文件中是否存在恶意代码的过程中可以直接确定恶意代码存在于哪个调用序列中，用户可以直接对存在恶意代码的调用序列进行处理，相比于现有技术，提高了对恶意代码的处理效率。

[0053] 下面，将对本申请实施例提供的恶意代码检测方法所涉及到的实施环境进行简要说明。

[0054] 请参考图1，图1是本申请实施例提供的恶意代码检测方法所涉及到的一种实施环境的示意图，该实施环境可以如图1所示，包括在服务器上安装有恶意代码检测程序的网络设备(图1中示出了一台电脑)，其中，恶意代码检测程序可以被网络设备的服务器调用以实现本申请实施例提供的恶意代码检测方法。

[0055] 可选的，本申请实施例中，网络设备可以是路由器、电脑以及交换机等。

[0056] 请参考图2,提供一种网络设备的服务器(以下简称为服务器),该服务器的内部结构图可以如图2所示,该服务器包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中,该服务器的处理器用于提供计算和控制能力。该服务器的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该服务器的数据库用于存储恶意样本集和正常样本集,恶意样本集中包括多个已知含有恶意代码的恶意调用序列;正常样本集中包括多个已知不含有恶意代码的正常调用序列。该服务器的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种恶意代码检测方法。

[0057] 图2中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的网络设备的限定,具体的网络设备可以包括比图2中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0058] 请参考图3,其示出了本申请实施例提供的一种恶意代码检测方法的流程图,该恶意代码检测方法可以应用于图2所示的服务器中。如图3所示,该恶意代码检测方法可以包括以下步骤:

[0059] 步骤301、服务器从应用程序接口API序列集合中选取第一调用序列和第二调用序列。

[0060] 本申请实施例中,API序列集合包括多个待检测的调用序列,第一调用序列和第二调用序列可以是多个待检测的调用序列中的两个调用序列。

[0061] 在一种可选的实现方式中,如图4所示,服务器从API序列集合中选取第一调用序列和第二调用序列之前,还包括以下步骤401-步骤402:

[0062] 步骤401、网络设备接收到目标文件之后,服务器可以在虚拟沙箱中运行接收到的待检测的目标文件,获取目标文件的API函数所对应的调用序列。

[0063] 其中,虚拟沙箱是指虚拟系统程序,可以运行在虚拟环境中运行目标文件,并可以将运行目标文件所产生的变化进行删除。虚拟沙箱可以通过重定向技术,把运行目标文件所生成和修改的文件定向到自身的文件夹中,从而避免目标文件对本地系统文件进行修改。从而可以避免目标文件中可能出现的恶意代码对本地系统进行攻击。

[0064] 服务器获取的API调用序列是指API调用的组合,多个API调用基于前后依赖关系组成API调用序列。

[0065] 本申请实施例中,当外界通过网络协议向网络设备发送目标文件时,服务器将所接收的目标文件放置于虚拟沙箱中运行,在运行过程中,服务器可以获取目标文件的静态动作信息,并根据静态动作信息获取目标文件的源代码,从目标文件的源代码中提取API调用序列。

[0066] 本申请实施例中,静态动作信息包括目标文件的MD5(英文:MD5Message-Digest Algorithm;缩写:MD5信息摘要算法)值,服务器根据静态动作信息获取目标文件的源代码的过程可以是:服务器根据目标文件的MD5值,判断是否调用脱壳工具,当目标文件的MD5值大于阈值时,采用脱壳工具获取目标文件的源代码。当目标文件的MD5值小于等于阈值时,则不需要采用脱壳工具。需要说明的是,“脱壳”是对软件加壳的逆操作。软件加壳是指在写好的软件上设置专门负责保护软件不被非法修改或反编译的程序。

[0067] 步骤402、对各调用序列，服务器获得调用序列的特征向量，形成API序列集合。

[0068] 本申请实施例中，可以通过局部敏感哈希sim-hash算法将API函数所对应的调用序列进行处理，获得二进制的API调用序列的特征向量H_i，多个调用序列的特征向量可以构成API序列集合。

[0069] 在一种可选的实现方式中，本申请实施例中可以从API序列集合中任意选取两个调用序列作为第一调用序列和第二调用序列。

[0070] 在一种可选的实现方式中，为了增大第一调用序列和第二调用序列的差异化，以便于对第一调用序列和第二调用序列进行区分，服务器从API序列集合中选取第一调用序列和第二调用序列的过程可以包括以下步骤B1-步骤B2：

[0071] 步骤B1、服务器分别计算API序列集合中每两个待检测的调用序列的汉明距离。

[0072] 汉明距离用于表示两个(相同长度)字对应位不同的数量，例如：码字A为10001001，码字B为10110001，那么码字A与码字B中不同的字符数为3，表示码字A与码字B的汉明距离就是3。

[0073] 服务器可以计算出API序列集合中的任意的两个调用序列的汉明距离。

[0074] 可选的，可以采用公式(1)计算API序列集合中每两个调用序列的汉明距离：

[0075] $$D_{ham}(y,z)=\sum_{r=1}^{m} y_r \oplus z_r$$ 公式（1）。

[0076] 在公式(1)中，y_r为API序列集合中的一个调用序列所对应的比特值，z_r为API序列集合中的另一个调用序列所对应的比特值，$D_{ham}(y,z)$为汉明距离，r为API序列集合中两两成组的组数，m为样本容量。

[0077] 步骤B2、服务器选取汉明距离最大的两个调用序列作为第一调用序列和第二调用序列。

[0078] 汉明距离越大，表示两个码字之间的相似度越低，汉明距离越小，表示两个码字之间的相似度越高。

[0079] 本申请实施例中通过选取汉明距离最大的两个调用序列，即选取相似度最低的两个调用序列分别作为第一调用序列和第二调用序列。

[0080] 步骤302、服务器分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值。

[0081] 其中，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率。其中，调用序列的正常风险值越大，表示调用序列为正常序列的可能性低。调用序列的正常风险值越小，表示调用序列为正常序列的可能性高。调用序列的恶意风险值越大，表示调用序列为恶意序列的可能性低。调用序列的恶意风险值越小，表示调用序列为恶意序列的可能性高。

[0082] 本申请实施例中，服务器可以分别对第一调用序列和第二调用序列计算各自的正常风险值和恶意风险值。在一种可选的实现方式中，以第一调用序列为例，本申请实施例中，如图5所示，服务器计算第一调用序列的正常风险值和恶意风险值的过程可以包括以下步骤：

[0083] 步骤501、服务器获取恶意样本集和正常样本集。

[0084] 本申请实施例中，可以对高级持续性威胁APT(英文：Advanced Persistent

Threat;缩写:APT)团伙进行追踪,通过VXHeavens、Malshare等恶意代码共享网站收集Backdoor(电脑病毒木马,简称“后门”)、Trojan(中文:木马病毒)、Virus(中文:病毒)和Worm(中文:蠕虫)等多种主要类型的恶意代码,并获取恶意代码对应的API函数所对应的恶意调用序列,通过sim-hash算法对已知的含有恶意代码的恶意调用序列进行处理,获得二进制的恶意调用序列的特征向量H_{i-}',多个恶意调用序列的特征向量H_{i-}'组合形成恶意样本集。

[0085] 与此同时,本申请实施例中,服务器可以获取已知的不存在恶意代码的正常代码对应的API函数所对应的的正常调用序列,通过sim-hash算法对已知的含有正常代码的正常调用序列进行处理,得到二进制的正常调用序列的特征向量H_{i+}',多个正常调用序列的特征向量H_{i+}'组合形成正常样本集。

[0086] 步骤502、针对恶意样本集中的每个恶意调用序列,服务器分别计算第一调用序列与恶意调用序列的恶意相似度;针对正常样本集中的每个正常调用序列,服务器分别计算第一调用序列与正常调用序列的正常相似度。

[0087] 本申请实施例中,可以采用公式(2)计算第一调用序列分别与每个恶意调用序列和每个正常调用序列的相似度。本申请实施例中,为了便于区分,将第一调用序列与恶意调用序列的相似度称为恶意相似度,将第一调用序列与正常序列的相似度称为正常相似度。

[0088] $$sim(H_i,H_i')=1-(\sum_{r=1}^{m}y_r\oplus z_r)/n$$ 公式(2)。

[0089] 其中,sim(H_i,H_i')为相似性度量,y_r为恶意样本集(或者正常样本集)中的一个恶意(或者正常)调用序列所对应的比特值,z_r为第一调用序列所对应的比特值,r为API序列集合中两两成组的组数,m为样本容量。

[0090] 可选的,为了对恶意样本集与正常样本集进行区分,本申请实施例中,可以使用H_{i+}'表示正常样本集对应的正常调用序列,H_{i-}'表示恶意样本集对应的恶意调用序列。那么,第一调用序列与恶意调用序列的恶意相似度可以表示为:sim(H_i,H_{i-}'),第一调用序列与正常调用序列的正常相似度可以表示为:sim(H_i,H_{i+}')。

[0091] 举例而言,本申请实施例中,假设正常样本集中包括5个特征向量H_{i+}',分别用L1、L2、L3、L4和L5表示。恶意样本集中包括5个特征向量H_{i-}',分别用L6、L7、L8、L9和L10表示。第一调用序列用A1表示,那么服务器可以计算出A1L1、A1L2、A1L3、A1L4和A1L5之间的正常相似度,以下简称为A1L1、A1L2、A1L3、A1L4和A1L5。相应的,第一调用序列与每个恶意调用序列的恶意相似度可以表示为:A1L6、A1L7、A1L8、A1L9和A1L10。

[0092] 步骤503、服务器获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值。

[0093] 本申请实施例中,建立包含随机森林算法的分类器C,将恶意样本集和正常样本集分别输入分类器C中进行训练,并获得分类器C的预测结果C_i。本申请实施例中,分类器对恶意样本集中恶意调用序列进行分类的结果可以用C_{i-}表示,分类器对正常样本集中正常调用序列进行分类的结果可以用C_{i+}表示。预测结果表示调用序列中不存在恶意代码的概率或者调用序列中存在恶意代码的概率。

[0094] 承接上文举例,针对正常样本集中L1、L2、L3、L4和L5,进行分类之后可以得到五个预测结果,分别用L1C_{i+}、L2C_{i+}、L3C_{i+}、L4C_{i+}和L5C_{i+}表示。

[0095] 针对恶意样本集中的L6、L7、L8、L9和L10，进行分类之后可以得到五个预测结果，分别用$L6C_{i-}$、$L7C_{i-}$、$L8C_{i-}$、$L9C_{i-}$和$L10C_{i-}$表示。

[0096] 同时，服务器还可以将第一调用序列A1输入分类器中，获得对A1的预测结果，用$A1C_i'$表示。

[0097] 步骤504、服务器根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值。

[0098] 本申请实施例中，服务器可以根据公式(3)计算第一调用序列的正常风险值，根据公式(4)计算第一调用序列的恶意风险值。

[0099] $R_{S+}=\sum (C_{i+}-C_i)^2/\mathrm{sim}(H_i,H_i')$ 公式(3)。

[0100] $R_{S-}=\sum (C_{i-}-C_i)^2/\mathrm{sim}(H_i,H_i')$ 公式(4)。

[0101] 承接上文举例，第一调用序列的正常风险值$A1R_{S+}$可以表示为：

[0102] $$A1R_{S+}=\frac{(L1C_{i+}-A1Ci')^2}{A1L1}+\frac{(L2C_{i+}-A1Ci')^2}{A1L2}+\frac{(L3C_{i+}-A1Ci')^2}{A1L3}+\frac{(L4C_{i+}-A1Ci')^2}{A1L4}+\frac{(L5C_{i+}-A1Ci')^2}{A1L5}$$

[0103] 第一调用序列的恶意风险值$A1R_{S-}$可以表示为：

[0104] $$A1R_{S-}=\frac{(L6C_{i-}-A1Ci')^2}{A1L6}+\frac{(L7C_{i-}-A1Ci')^2}{A1L7}+\frac{(L8C_{i-}-A1Ci')^2}{A1L8}+\frac{(L9C_{i-}-A1Ci')^2}{A1L9}+\frac{(L10C_{i-}-A1Ci')^2}{A1L10}$$

[0105] 基于步骤501-步骤504相同的原理，本申请实施例中，服务器可以计算得到第二调用序列的正常风险值和恶意风险值，分别用$A2R_{S+}$和$A2R_{S-}$表示。

[0106] 步骤303、服务器根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果。

[0107] 其中，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码。

[0108] 在一种可选的实现方式中，如图6所示，服务器对目标序列进行标注，得到标注结果的过程可以包括以下步骤：

[0109] 步骤601、服务器可以从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值。

[0110] 承接上文举例，第一调用序列的正常风险值为$A1R_{S+}$，第一调用序列的恶意风险值为$A1R_{S-}$，第二调用序列的正常风险值为$A2R_{S+}$，第二调用序列的恶意风险值为$A2R_{S-}$。从$A1R_{S+}$、$A1R_{S-}$、$A2R_{S+}$、$A2R_{S-}$中选择最小的风险值。

[0111] 例如$A2R_{S-}$为最小的风险值。

[0112] 步骤602、服务器可以将最小的风险值对应的调用序列确定为目标序列。

[0113] $A2R_{S-}$对应的调用序列为第二调用序列，即第二调用序列为目标序列。

[0114] 步骤603、服务器根据最小的风险值对目标序列进行标注，得到标注结果。

[0115] 本申请实施例中，当最小的风险值为第一调用序列的正常风险值或者第二调用序列的正常风险值时，标注结果为目标序列不存在恶意代码。

[0116] 当最小的风险值为第一调用序列的恶意风险值或者第二调用序列的恶意风险值时，标注结果为目标序列存在恶意代码。

[0117] 承接上文举例，本申请实施例中，最小的风险值$A2R_S$-为第二调用序列的恶意风险值，因此标注结果为第二调用序列中存在恶意代码。

[0118] 步骤304、服务器将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0119] 本申请实施例中，服务器将第二调用序列A2标记为存在恶意代码，并将第二调用序列A2从API序列集合中剔除。

[0120] 例如：API序列集合中包括A1至A10个待检测的调用序列，将A2剔除之后，除去目标序列的API序列集合包括A1和A3至A10。将除去目标序列的API序列集合作为下一次循环的API序列集合，然后服务器从A1和A3至A10中选择新的第一调用序列和第二调用序列，循环执行上述步骤，以实现对API序列集合中的每个调用序列的标注。

[0121] 其中，将除去目标序列的API序列集合作为下一次循环的API序列集合，还包括：

[0122] 当下一次循环的API序列集合只包括一个待检测的调用序列时，停止检测。

[0123] 其中，API序列集合中除去目标序列后，只剩下一个待检测的目标序列时，作为下一次循环的API序列集合在下一次循环时，已经无法满足从API序列集合中选取第一调用序列和第二调用序列的条件，因此循环检测过程无法继续进行，此时，服务器检测到除去目标序列的API序列集合中只包括一个待检测的调用序列时，停止检测。

[0124] 服务器可以向人工检测终端发送检测指令。

[0125] 检测指令用于指示对下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

[0126] 即服务器可以向人工检测终端发送检测指令，以及最后一个待检测的调用序列的相关代码，工作人员可以对最后一个待检测的调用序列进行人工检测并标注，得到标注结果，标注结果包括存在恶意代码或者不存在恶意代码。人工检测终端可以向服务器反馈标注结果。

[0127] 本申请实施例提供的恶意代码检测方法中，网络设备的服务器(以下简称服务器)可以从应用程序接口API(英文:Application Programming Interface，缩写:API)序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列。服务器可以分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，其中，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率。服务器可以根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，其中目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码。服务器可以将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。由此可知，本申请实施例中，网络设备的服务器对所有调用序列进行标注，这样就可以确定多个待检测的调用序列中的每个调用序列存在恶意代码或者不存在恶意代码，从而在确定目标文件中是否存在恶意代码的过程中可以直接确定恶意代码存在于哪个调用序列中，用户可以直接对存在恶意代码的调用序列进行处理，相比于现有技术，提高了对恶意代码的处理效率。

[0128] 进一步的，本申请实施例中，可以准确高效地检测出目标文件中是否存在恶意代码，并确定出存在恶意代码的调用序列，这样极大地提高了用户对存在恶意代码的目标文件的分析追踪和定位能力。对于用户追踪APT攻击者的身份有极大帮助。

[0129] 请参考图7，其示出了本申请实施例提供的一种恶意代码检测装置的框图，该恶意代码检测装置可以配置在图2所示实施环境中的服务器中。如图7所示，该恶意代码检测装置可以包括调用序列选取模块701、风险计算模块702、标注模块703和循环处理模块704，其中：

[0130] 调用序列选取模块701，用于从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列；

[0131] 风险计算模块702，用于分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率；

[0132] 标注模块703，用于根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码；

[0133] 循环处理模块704，用于将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0134] 在本申请的一个实施例中，风险计算模块702还用于获取恶意样本集和正常样本集，恶意样本集中包括多个已知含有恶意代码的恶意调用序列；正常样本集中包括多个已知不含有恶意代码的正常调用序列；针对恶意样本集中的每个恶意调用序列，分别计算第一调用序列与恶意调用序列的恶意相似度；针对正常样本集中的每个正常调用序列，分别计算第一调用序列与正常调用序列的正常相似度；获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值；根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值。

[0135] 在本申请的一个实施例中，标注模块703还用于从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值；将最小的风险值对应的调用序列确定为目标序列；根据最小的风险值对目标序列进行标注，得到标注结果。

[0136] 在本申请的一个实施例中，标注模块703还用于当最小的风险值为第一调用序列的正常风险值或者第二调用序列的正常风险值时，标注结果为目标序列不存在恶意代码；当最小的风险值为第一调用序列的恶意风险值或者第二调用序列的恶意风险值时，标注结果为目标序列存在恶意代码。

[0137] 在本申请的一个实施例中，调用序列选取模块701还用于分别计算API序列集合中每两个待检测的调用序列的汉明距离；选取汉明距离最大的两个调用序列作为第一调用序列和第二调用序列。

[0138] 在本申请的一个实施例中，调用序列选取模块701还用于在虚拟沙箱中运行接收

到的待检测的目标文件，获取目标文件的API函数所对应的调用序列；对各调用序列，获得调用序列的特征向量，形成API序列集合。

[0139] 在本申请的一个实施例中，循环处理模块704还用于当下一次循环的API序列集合只包括一个待检测的调用序列时，停止检测；向人工检测终端发送检测指令，检测指令用于指示对下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

[0140] 在本申请的一个实施例中，提供了一种计算机设备，包括存储器和处理器，存储器存储有计算机程序，处理器执行计算机程序时实现以下步骤：

[0141] 从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列；分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率；根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码；将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0142] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：获取恶意样本集和正常样本集，恶意样本集中包括多个已知含有恶意代码的恶意调用序列；正常样本集中包括多个已知不含有恶意代码的正常调用序列；针对恶意样本集中的每个恶意调用序列，分别计算第一调用序列与恶意调用序列的恶意相似度；针对正常样本集中的每个正常调用序列，分别计算第一调用序列与正常调用序列的正常相似度；获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值；根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值。

[0143] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值；将最小的风险值对应的调用序列确定为目标序列；根据最小的风险值对目标序列进行标注，得到标注结果。

[0144] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：当最小的风险值为第一调用序列的正常风险值或者第二调用序列的正常风险值时，标注结果为目标序列不存在恶意代码；当最小的风险值为第一调用序列的恶意风险值或者第二调用序列的恶意风险值时，标注结果为目标序列存在恶意代码。

[0145] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：分别计算API序列集合中每两个待检测的调用序列的汉明距离；选取汉明距离最大的两个调用序列作为第一调用序列和第二调用序列。

[0146] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：在虚拟沙箱中运行接收到的待检测的目标文件，获取目标文件的API函数所对应的调用序列；对各调用序列，获得调用序列的特征向量，形成API序列集合。

[0147] 在本申请的一个实施例中，处理器执行计算机程序时还可以实现以下步骤：当下

一次循环的API序列集合只包括一个待检测的调用序列时，停止检测；向人工检测终端发送检测指令，检测指令用于指示对下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

[0148] 本申请实施例提供的计算机设备，其实现原理和技术效果与上述方法实施例类似，在此不再赘述。

[0149] 在本申请的一个实施例中，提供了一种计算机可读存储介质，其上存储有计算机程序，计算机程序被处理器执行时实现以下步骤：

[0150] 从应用程序接口API序列集合中选取第一调用序列和第二调用序列，API序列集合包括多个待检测的调用序列；分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值，正常风险值表示调用序列中存在恶意代码的概率；恶意风险值表示调用序列中不存在恶意代码的概率；根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果，目标序列为第一调用序列或者第二调用序列，标注结果包括存在恶意代码或者不存在恶意代码；将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成。

[0151] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：获取恶意样本集和正常样本集，恶意样本集中包括多个已知含有恶意代码的恶意调用序列；正常样本集中包括多个已知不含有恶意代码的正常调用序列；针对恶意样本集中的每个恶意调用序列，分别计算第一调用序列与恶意调用序列的恶意相似度；针对正常样本集中的每个正常调用序列，分别计算第一调用序列与正常调用序列的正常相似度；获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值；根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值。

[0152] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值；将最小的风险值对应的调用序列确定为目标序列；根据最小的风险值对目标序列进行标注，得到标注结果。

[0153] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：当最小的风险值为第一调用序列的正常风险值或者第二调用序列的正常风险值时，标注结果为目标序列不存在恶意代码；当最小的风险值为第一调用序列的恶意风险值或者第二调用序列的恶意风险值时，标注结果为目标序列存在恶意代码。

[0154] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：分别计算API序列集合中每两个待检测的调用序列的汉明距离；选取汉明距离最大的两个调用序列作为第一调用序列和第二调用序列。

[0155] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：在虚拟沙箱中运行接收到的待检测的目标文件，获取目标文件的API函数所对应的调用序列；对各调用序列，获得调用序列的特征向量，形成API序列集合。

[0156] 在本申请的一个实施例中，计算机程序被处理器执行时还可以实现以下步骤：当下一次循环的API序列集合只包括一个待检测的调用序列时，停止检测；向人工检测终端发送检测指令，检测指令用于指示对下一次循环的API序列集合中包括的待检测的调用序列进行人工检测。

[0157] 本申请实施例提供的计算机可读存储介质，其实现原理和技术效果与上述方法实施例类似，在此不再赘述。

[0158] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的计算机程序可存储于一非易失性计算机可读取存储介质中，该计算机程序在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限，RAM以多种形式可得，诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0159] 以上所述实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

[0160] 以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对申请专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。

图1

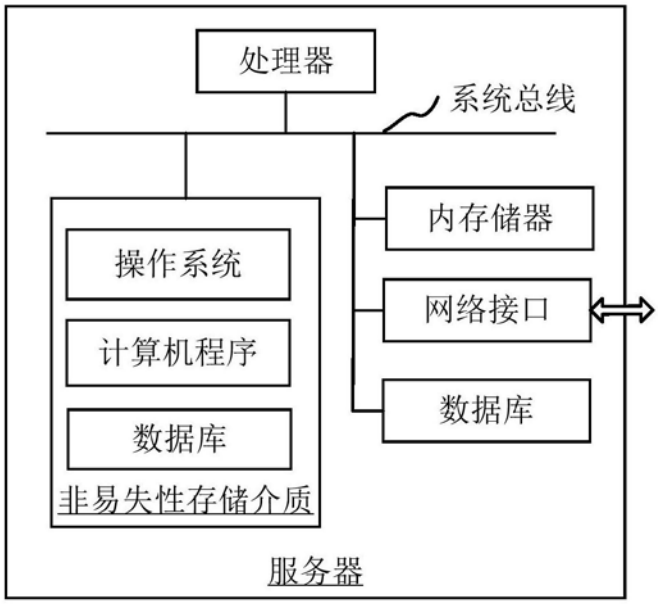


图2

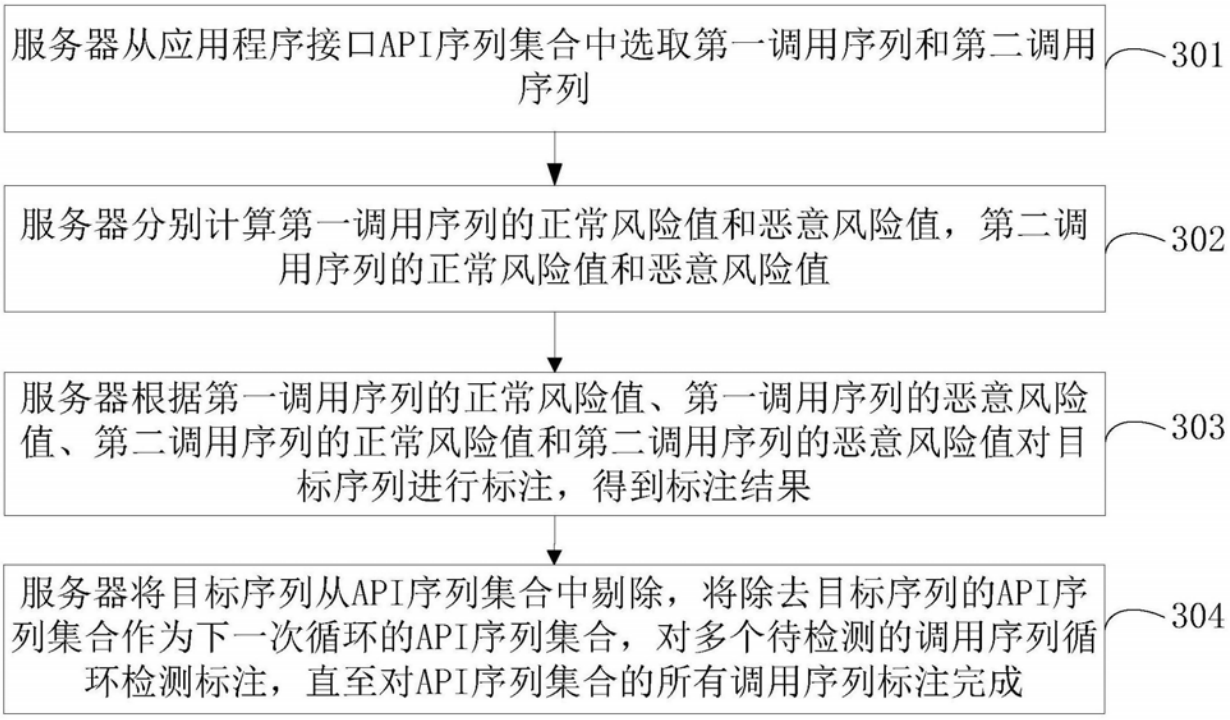

服务器从应用程序接口API序列集合中选取第一调用序列和第二调用序列
301
服务器分别计算第一调用序列的正常风险值和恶意风险值，第二调用序列的正常风险值和恶意风险值
302
服务器根据第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值对目标序列进行标注，得到标注结果
303
服务器将目标序列从API序列集合中剔除，将除去目标序列的API序列集合作为下一次循环的API序列集合，对多个待检测的调用序列循环检测标注，直至对API序列集合的所有调用序列标注完成
304

图3

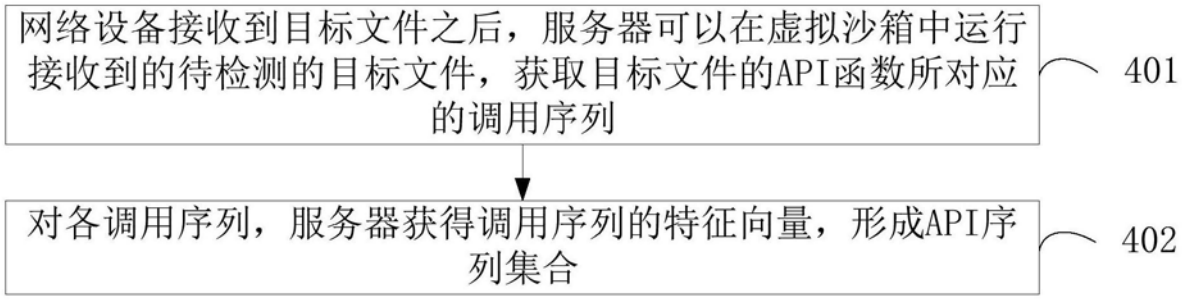

网络设备接收到目标文件之后，服务器可以在虚拟沙箱中运行接收到的待检测的目标文件，获取目标文件的API函数所对应的调用序列
401
对各调用序列，服务器获得调用序列的特征向量，形成API序列集合
402

图4

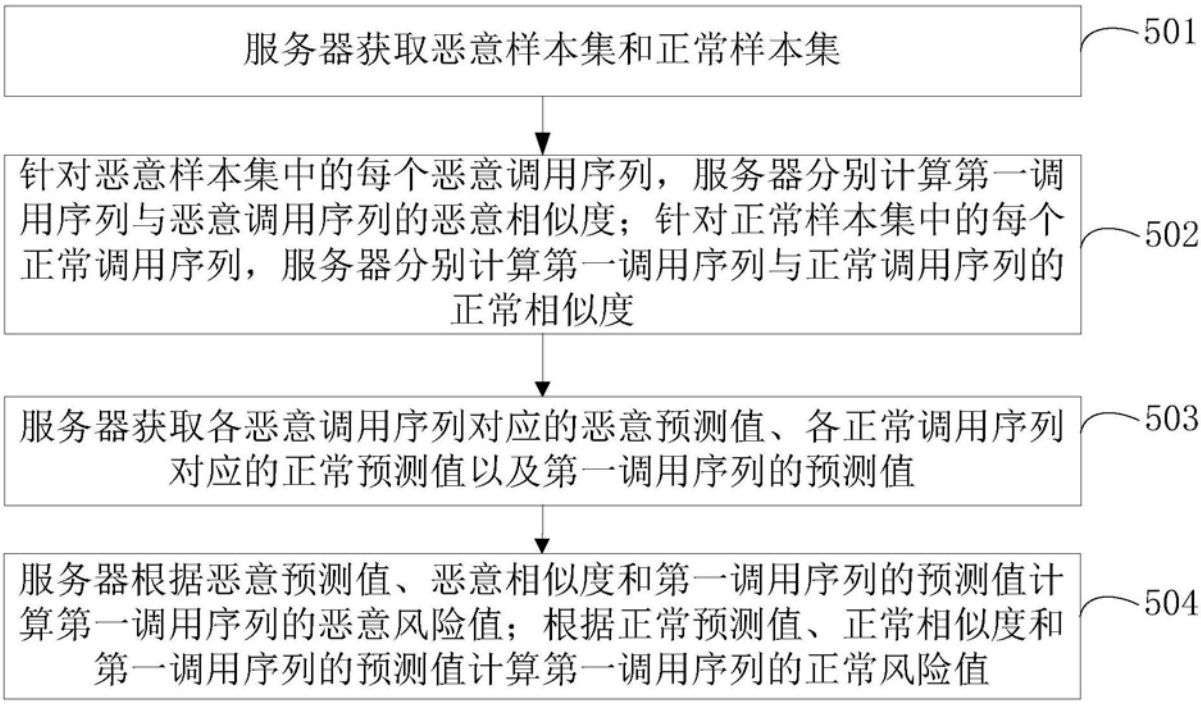

服务器获取恶意样本集和正常样本集
501
针对恶意样本集中的每个恶意调用序列，服务器分别计算第一调用序列与恶意调用序列的恶意相似度；针对正常样本集中的每个正常调用序列，服务器分别计算第一调用序列与正常调用序列的正常相似度
502
服务器获取各恶意调用序列对应的恶意预测值、各正常调用序列对应的正常预测值以及第一调用序列的预测值
503
服务器根据恶意预测值、恶意相似度和第一调用序列的预测值计算第一调用序列的恶意风险值；根据正常预测值、正常相似度和第一调用序列的预测值计算第一调用序列的正常风险值
504

图5

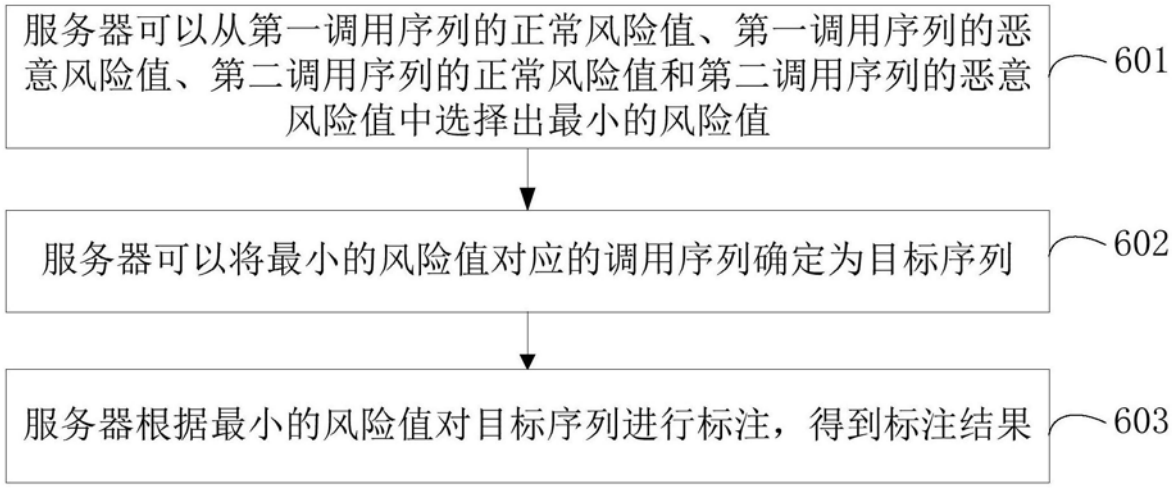

服务器可以从第一调用序列的正常风险值、第一调用序列的恶意风险值、第二调用序列的正常风险值和第二调用序列的恶意风险值中选择出最小的风险值
601
服务器可以将最小的风险值对应的调用序列确定为目标序列
602
服务器根据最小的风险值对目标序列进行标注，得到标注结果
603

图6

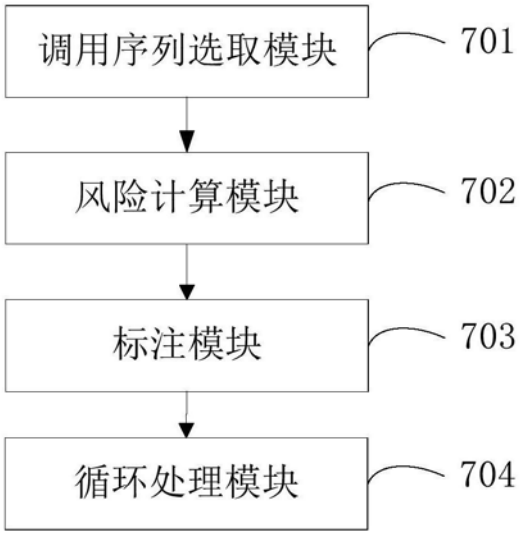


图7