

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2018 年 5 月 17 日 (17.05.2018)



(10) 国际公布号
WO 2018/086292 A1

(51) 国际专利分类号:
G06F 21/57 (2013.01)

(21) 国际申请号: PCT/CN2017/077984

(22) 国际申请日: 2017 年 3 月 24 日 (24.03.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201611036378.2 2016 年 11 月 14 日 (14.11.2016) CN

(71) 申请人: 平安科技 (深圳) 有限公司 (PING AN TECHNOLOGY(SHENZHEN)CO.,LTD.) [CN/CN];
中国广东省深圳市福田区八卦岭八卦三路平安大厦, Guangdong 518000 (CN)。

(72) 发明人: 王金锭 (WANG, Jinding); 中国广东省深圳市福田区八卦岭八卦三路平安大厦, Guangdong 518000 (CN)。

(74) 代理人: 深圳市世纪恒程知识产权代理事务所 (CENFO INTELLECTUAL PROPERTY AGENCY); 中国广东省深圳市南山区南山大道 3838 号设计产业园金栋二层 210-212 (原南头城工业村 11 栋), Guangdong 518052 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY,

(54) Title: METHOD AND SYSTEM FOR DETECTING SECURITY HOLE OF APPLICATION SOFTWARE, DEVICE, AND STORAGE MEDIUM

(54) 发明名称: 应用软件安全漏洞检测方法、系统、设备及存储介质

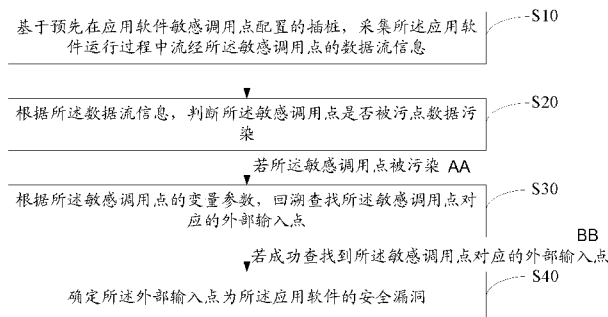


图 1

- S10 Acquire, on the basis of instrumentation configured on a sensitive calling point of application software in advance, information of data flow which flows through the sensitive calling point during a running process of the application software
- S20 Determine, according to the information of data flow, whether the sensitive calling point is polluted by tainted data
- S30 Backtrack to search for an external input point corresponding to the sensitive calling point according to a variable parameter of the sensitive calling point
- S40 Determine the external input point as a security hole of the application software
- AA If the sensitive calling point is polluted
- BB If the external input point corresponding to the sensitive calling point is successfully found

(57) Abstract: Disclosed is a method for detecting a security hole of application software. The method comprises: acquiring, on the basis of instrumentation configured on a sensitive calling point of application software in advance, information of data flow which flows through the sensitive calling point during a running process of the application software; determining, according to the information of data flow, whether the sensitive calling point is polluted by tainted data; if the sensitive calling point is polluted, backtracking to search for an external input point corresponding to the sensitive calling point according to a variable parameter of the sensitive calling

MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告 (条约第21条(3))。

point; and if the external input point corresponding to the sensitive calling point is successfully found, determining the external input point as a security hole of the application software. Also disclosed are a system for detecting a security hole of application software, a device, and a computer-readable storage medium. The present invention greatly improves the detection efficiency of a security hole of application software.

(57) 摘要: 本发明公开了一种应用软件安全漏洞检测方法, 该方法包括: 基于预先在应用软件敏感调用点配置的插桩, 采集所述应用软件运行过程中流经所述敏感调用点的数据流信息; 根据所述数据流信息, 判断所述敏感调用点是否被污点数据污染; 若所述敏感调用点被污染, 则根据所述敏感调用点的变量参数, 回溯查找所述敏感调用点对应的外部输入点; 若成功查找到所述敏感调用点对应的外部输入点, 则确定所述外部输入点为所述应用软件的安全漏洞。本发明还公开了一种应用软件安全漏洞检测系统、设备及计算机可读存储介质。本发明大大提高了应用软件的安全漏洞检测效率。

说明书

发明名称：应用软件安全漏洞检测方法、系统、设备及存储介质

[1] 技术领域

[2] 本发明涉及数据安全技术领域，尤其涉及一种应用软件安全漏洞检测方法、系统、设备及计算机可读存储介质。

[3] 背景技术

[4] 智能手机中安装的应用软件类型多样，为人们的生活、娱乐、工作带来了极大的便利。由于应用软件可能涉及各用户的隐私、财产安全，因此，应用软件的安全性一直是人们关注的重点。

[5] 现有应用软件安全漏洞检测大部分是基于静态的规则扫描和应用组件类的模糊测试实现，如墨贝系统和Drozer工具，是通过对应应用软件代码的静态扫描实现软件安全性的检测。这样的检测方法方案比较粗糙，仅仅是基于代码文字的规则匹配，存在大量的误报和漏报；另一方面，由于检测的覆盖面有限，并不能深入发现应用软件的功能逻辑所存在的安全隐患。

[6] 因此，如何提高应用软件的安全漏洞检测效率成为了亟需解决的问题。

[7] 发明内容

[8] 本发明的主要目的在于提供一种软件安全漏洞检测方法、系统、设备及计算机可读存储介质，旨在解决应用软件的安全漏洞检测效率低的技术问题。

[9] 为实现上述目的，本发明提供一种应用软件安全漏洞检测方法，所述应用软件安全漏洞检测方法包括以下步骤：

[10] 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；

[11] 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；

[12] 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；

[13] 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

- [14] 优选地，所述根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤包括：
- [15] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；
- [16] 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；
- [17] 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [18] 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [19] 优选地，所述应用软件安全漏洞检测方法还包括：
- [20] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [21] 优选地，所述根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤还包括：
- [22] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；
- [23] 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [24] 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [25] 优选地，所述基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息的步骤之前，还包括：
- [26] 当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。
- [27] 优选地，所述确定所述外部输入点为所述应用软件的安全漏洞的步骤之后，还包括：
- [28] 确定回溯找到所述外部输入点的路径为危险路径。
- [29] 此外，为实现上述目的，本发明还提供一种应用软件安全漏洞检测系统，所述应用软件安全漏洞检测系统包括：

- [30] 采集模块，用于基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；
- [31] 判断模块，用于根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；
- [32] 回溯模块，用于若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；
- [33] 确定模块，用于若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。
- [34] 优选地，所述回溯模块还用于，
- [35] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [36] 优选地，所述回溯模块还用于，
- [37] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [38] 优选地，所述回溯模块还用于，
- [39] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [40] 优选地，所述应用软件安全漏洞检测系统还包括：
- [41] 污点模块，用于当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。
- [42] 优选地，所述确定模块还用于，
- [43] 确定回溯找到所述外部输入点的路径为危险路径。
- [44] 此外，为实现上述目的，本发明还提供一种应用软件安全漏洞检测设备，所述

应用软件安全漏洞检测设备包括处理器及存储器，所述存储器中存储有应用软件安全漏洞检测程序；

[45] 所述处理器用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

[46] 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；

[47] 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；

[48] 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；

[49] 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[50] 优选地，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

[51] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；

[52] 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；

[53] 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

[54] 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[55] 优选地，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

[56] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

[57] 优选地，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

[58] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；

[59] 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用

点对应的外部输入点；

[60] 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[61] 此外，为实现上述目的，本发明还提供一种计算机可读存储介质，所述计算机可读存储介质存储有一个或者多个程序，所述一个或者多个程序可被一个或者多个处理器执行，以实现以下步骤：

[62] 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；

[63] 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；

[64] 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；

[65] 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[66] 优选地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：

[67] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；

[68] 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；

[69] 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

[70] 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[71] 优选地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：

[72] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

[73] 优选地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：

- [74] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；
- [75] 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [76] 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [77] 本发明实施例提出的一种应用软件安全漏洞检测方法、系统、设备和计算机可读存储介质，通过基于预先在应用软件敏感调用点配置的插桩，采集应用软件运行过程中流经敏感调用点的数据流信息，进行动态的数据采集，模拟应用软件实际运行的状态，相对于静态数据来说采集的数据更具有真实性，能够更好的进行应用软件的安全性检测，提高安全漏洞检测效率；然后，根据流经敏感调用点的数据流信息，判断敏感调用点是否被污点数据污染；若敏感调用点被污染，则敏感调用点可能被外部输入点关联调用，影响应用软件安全，此时，根据敏感调用点的变量参数，回溯查找敏感调用点对应的外部输入点，也即查找更够调用敏感调用点的外部输入点；若成功查找到敏感调用点对应的外部输入点，则确定此找到的外部输入点为应用软件的安全漏洞，会给应用软件带来风险。本发明中基于应用软件运行过程中的关键调用和数据处理操作的实时监控输出，通过离线自动污点数据传播分析，检测应用潜在的安全漏洞。通过采用动态插桩和离线数据流分析动静态结合的检测技术，能够覆盖应用软件更深层次功能和逻辑检查，不存在漏报，且采集的数据来自实际运行过程，真实有效，不存在误报，大大提高了应用软件的安全漏洞检测效率。
- [78] 附图说明
- [79] 图1为本发明应用软件安全漏洞检测方法第一实施例的流程示意图；
- [80] 图2为图1中根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤的细化流程示意图；
- [81] 图3为图1中根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤的细化流程示意图；
- [82] 图4为本发明应用软件安全漏洞检测方法第二实施例的流程示意图；

- [83] 图5为本发明应用软件安全漏洞检测方法第三实施例的流程示意图；
- [84] 图6为本发明应用软件安全漏洞检测系统第一实施例的功能模块示意图；
- [85] 图7为本发明应用软件安全漏洞检测系统第二实施例的功能模块示意图；
- [86] 图8是本发明实施例方案涉及的硬件运行环境的设备结构示意图。
- [87] 本发明目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。
- [88] 具体实施方式
- [89] 应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。
- [90] 参照图1，本发明应用软件安全漏洞检测方法第一实施例提供一种应用软件安全漏洞检测方法，所述应用软件安全漏洞检测方法包括：
- [91] 步骤S10、基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息。
- [92] 本发明通过在应用软件的函数敏感调用点配置插桩，在程序运行时，获取流经插桩点（也即敏感调用点）的数据流，用于检测敏感调用点是否安全，从而进行数据回溯。由于采集的数据是应用软件真实运行过程中的数据，因此真实有效，降低了漏洞漏报和误报的概率，提高了应用软件的安全漏洞检测效率。
- [93] 具体的，作为一种实施方式，首先，在不破坏待检测应用软件的程序原有逻辑完整性的前提下，在待检测应用软件需检测的敏感调用点配置插桩，以采集应用软件的运行时的关键调用数据。
- [94] 在配置程序插桩后，启动待检测的应用软件，获取插桩采集的流经敏感调用点的数据流信息。
- [95] 插桩采集的数据流信息包括应用软件运行过程中，调用此敏感调用点时传递给此敏感调用点的参数，以及敏感调用点根据传递的参数执行函数调用后输出的返回值等信息。
- [96] 需要说明的是，敏感调用点是应用软件中的关键调用点，或是安全性要求要高的调用点，可以由测试人员人为确定需检测的敏感调用点，在需检测的各敏感调用点配置插桩。
- [97] 步骤S20、根据所述数据流信息，判断所述敏感调用点是否被污点数据污染。

- [98] 需要说明的是，在通过外部输入点向应用软件输入污点数据后，污点数据会在应用软件运行的过程中，传递污染关联的参数，关联参数会被打上污染标签。
- [99] 则，若流经敏感调用点的参数携带有污点标签，则可以判定此敏感调用点被污点数据污染，也即，外部输入点输入的数据能够影响、控制此敏感调用点，存在风险。
- [100] 步骤S30、若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点。
- [101] 若敏感调用点被污染，则首先需要确定敏感调用点的变量参数，例如，敏感调用点使用形参执行函数输出返回值，则敏感调用点的形参即为变量参数，需要追踪传递给敏感调用点、对形参赋值的实参。
- [102] 然后，根据插桩采集的数据流信息中传递给此敏感调用点的实参，进行数据回溯，查找此变量参数数值的来源，以获取影响此变量参数的值的调用点，直至查找到的能够影响此变量参数的值的调用点为源节点，例如常量赋值、外部输入点等时，停止数据回溯。
- [103] 需要说明的是，源节点参数不再受应用软件内部其他调用点的影响。
- [104] 若能够影响此变量参数的值的调用点的源节点为外部输入点，则确定此外部输入点为敏感调用点对应的外部输入点，此外部输入点输入的数据能够影响、控制敏感调用点，给应用软件造成风险。
- [105] 作为一种实施方式，参照图2，所述根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤包括：
- [106] 步骤S31、根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；
- [107] 步骤S32、若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；
- [108] 步骤S33、若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [109] 步骤S34、若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [110] 例如，主函数调用当前敏感调用点时，传递给敏感调用点的参数为A，敏感调

用点根据A执行函数调用后会对应输出返回值，则确定敏感调用点的形参为变量参数。

[111] 然后，根据传递给敏感调用点的参数A，查找应用软件中返回或输出A的前驱调用点，也即查找将A作为返回值输出且执行顺序早于敏感调用点的调用点。

[112] 需要说明的是，前驱调用点在应用软件的各调用点执行顺序中，早于对比调用点执行函数调用。例如上述“查找输出A的前驱调用点”中，前驱调用点的函数执行顺序应当早于当前敏感调用点。

[113] 若前驱调用点1输出的返回值为A，则判断前驱调用点1输出的参数是否为外部输入点输入的变量。

[114] 若前驱调用点1输出的参数是外部输入点输入的变量，也即A是由前驱调用点1根据外部输入点输入的变量对应输出的，则可以确认此外部输入点为数据流的源节点，也就是当前敏感调用点对应的外部输入点。

[115] 若前驱调用点1是根据输入的实参B，执行函数调用运算得到A，则根据B继续查找返回B的前驱调用点，也即查找将B作为返回值输出，且执行顺序早于前驱调用点1的调用点，以此类推，直至找到数据流的源节点。

[116] 若源节点为外部输入点，则可以确定找到源节点为敏感调用点对应的外部输入点。

[117] 若源节点为常量赋值，则可以确定未回溯到外部输入点，当前敏感调用点安全，停止回溯。

[118] 进一步地，若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

[119] 在根据变量参数进行数据回溯时，可能会遇到多个调用点输出参数均为此变量参数，此时，首先根据各调用点的执行顺序，筛选出执行顺序先于敏感调用点的调用点，也可称为执行顺序早于敏感调用点的调用点。

[120] 若筛选后，得到的调用点只有一个，则就可以确定此调用点为当前查找的、用于进一步进行数据回溯前驱调用点；若筛选后，得到的调用点有多个，则此时根据这多个调用点的执行顺序，确定执行顺序最晚、但早于敏感调用点的调用

点为当前查找的、用于进一步进行数据回溯前驱调用点，也即确定执行顺序最接近敏感调用点的为前驱调用点。

[121] 例如，若输出A的调用点包括调用点2、调用点3，则获取调用点2、调用点3的执行顺序。若调用点2先于调用点3执行，敏感调用点晚于调用点3执行，则调用点3的执行顺序更加接近敏感调用点，确定调用点3为前驱调用点；若调用点3先于调用点2执行，敏感调用点晚于调用点2执行，则调用点2的执行顺序更加接近敏感调用点，确定调用点2为前驱调用点。

[122] 同理，在后续回溯数据时，若遇到多个调用点输出的参数均为当前回溯的参数、无法确定前驱调用点的情况时，选取执行顺序早于当前调用点、且最接近当前调用点的为前驱调用点，继续进行数据回溯。

[123] 由此，在有多个调用点输出变量参数时，可以唯一确定前驱调用点，实现了通过调用点时序，进行数据回溯和外部输入点的查找。

[124] 进一步地，作为另一种实施方式，参照图3，所述根据所述数据流信息中传递给所述敏感调用点的参数，回溯查找所述敏感调用点对应的外部输入点的步骤，还包括：

[125] 步骤S35、若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；

[126] 步骤S36、若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

[127] 步骤S37、若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[128] 若敏感调用点的变量参数值是由前驱参数的赋值决定的，也即传递给敏感调用点、对变量参数进行赋值的参数是前驱调用点的赋值，则回溯对应的右值，也即对传递给敏感调用点的变量参数进行赋值的右值。

[129] 若此右值是外部输入点输入的变量，也即前驱调用点是根据外部输入点输入的变量对输出的参数进行赋值的，则可以确定此外部输入点为敏感调用点对应的外部输入点。

[130] 若右值为常量赋值，则可以确定当前数据链安全，未回溯到外部输入点，停止

回溯。

[131] 由此，实现了对函数调用和赋值的数据追踪，保障了回溯调用点的完整性和准确性。

[132] 步骤S40、若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[133] 若成功回溯数据找到敏感调用点对应的外部输入点，则找到的外部输入点可能会被利用输入数据，调用此敏感调用点，给应用软件带来安全隐患。

[134] 由此，可以确定此外部输入点为应用软件的安全漏洞，进行漏洞上报。

[135] 在本实施例中，基于预先在应用软件敏感调用点配置的插桩，采集应用软件运行过程中流经敏感调用点的数据流信息，进行动态的数据采集，模拟应用软件实际运行的状态，相对于静态数据来说采集的数据更具有真实性，能够更好的进行应用软件的安全性检测，提高安全漏洞检测效率；然后，根据流经敏感调用点的数据流信息，判断敏感调用点是否被污点数据污染；若敏感调用点被污染，则敏感调用点可能被外部输入点关联调用，影响应用软件安全，此时，根据敏感调用点的变量参数，回溯查找敏感调用点对应的外部输入点，也即查找更够调用敏感调用点的外部输入点；若成功查找到敏感调用点对应的外部输入点，则确定此找到的外部输入点为应用软件的安全漏洞，会给应用软件带来风险。本实施例中基于应用软件运行过程中的关键调用和数据处理操作的实时监控输出，通过离线自动污点数据传播分析，检测应用潜在的安全漏洞。通过采用动态插桩和离线数据流分析动静态结合的检测技术，能够覆盖应用软件更深层次功能和逻辑检查，不存在漏报，且采集的数据来自实际运行过程，真实有效，不存在误报，大大提高了应用软件的安全漏洞检测效率。

[136] 进一步地，参照图4，本发明应用软件安全漏洞检测方法第二实施例提供一种应用软件安全漏洞检测方法，基于上述本发明应用软件安全漏洞检测方法第一实施例，所述步骤S10之前，还包括：

[137] 步骤S50、当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。

[138] 当应用软件的敏感调用点已经配置好插桩，启动运行时，或在应用软件运行的

过程中，通过应用程序的各个外部输入点，向应用程序中输入带有污染标签的污点数据。

[139] 污点数据再被输入应用程序后，跟随各调用点的调用，通过赋值、函数调用等方式，污染关联的参数。若被污染的参数流经调用点，则可认为此调用点被污点。

[140] 若应用程序运行的过程中敏感调用点被污染，则可以确定外部输入点输入的污点数据的关联参数能够流经此敏感调用点进行函数调用，也即此敏感调用点能够被外部输入点调用，存在安全漏洞。然后，根据敏感调用点的数据流进行数据回溯，查找能够调用敏感调用点的外部输入点，此能够调用敏感调用点的外部输入点即为应用程序的安全漏洞。

[141] 在本实施例中，当所述应用程序运行时，在应用程序的各外部输入点输入污点数据，用以检测敏感调用点是否被感染，从而根据被感染的敏感调用点查找对应的安全漏洞，实现了对应用程序漏洞的检测。本实施例通过向各外部输入点输入污点数据，使应用程序运行起来，能够快速、准确的标记出存在被外部输入点调用风险的敏感调用点，进而提高风险调用点的准确率，实现对应用程序深层逻辑的检测，减少漏洞的误报和漏报，提高漏洞检测的效率。

[142] 进一步地，参照图5，本发明应用程序安全漏洞检测方法第三实施例提供一种应用程序安全漏洞检测方法，基于上述本发明应用程序安全漏洞检测方法第一实施例或第二实施例（本实施例以上述本发明应用程序安全漏洞检测方法第一实施例为例），所述步骤S40之后，还包括：

[143] 步骤S60、确定回溯找到所述外部输入点的路径为危险路径。

[144] 当成功找到敏感调用点对应的外部输入点时，将找到外部输入点的路径信息提取出来，得到从外部输入点输入的污点数据流向敏感调用点的路径，以及路径上被感染的各调用点。

[145] 然后，标记此路径为危险路径，便于研发人员了解敏感调用点的具体风险，以提高应用程序的完全性。

[146] 在本实施例中，确定回溯找到敏感调用点对应外部输入点的路径为危险路径，以便于研发人员进行进一步地漏洞原因分析、提升应用程序的安全性。

- [147] 需要说明的是，本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。
- [148] 参照图6，本发明应用软件安全漏洞检测系统第一实施例提供一种应用软件安全漏洞检测系统，所述应用软件安全漏洞检测系统包括：
- [149] 采集模块10，用于基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息。
- [150] 本发明通过在应用软件的函数敏感调用点配置插桩，在程序运行时，获取流经插桩点（也即敏感调用点）的数据流，用于检测敏感调用点是否安全，从而进行数据回溯。由于采集的数据是应用软件真实运行过程中的数据，因此真实有效，降低了漏洞漏报和误报的概率，提高了应用软件的安全漏洞检测效率。
- [151] 具体的，作为一种实施方式，首先，在不破坏待检测应用软件的程序原有逻辑完整性的前提下，在待检测应用软件需检测的敏感调用点配置插桩，以采集应用软件的运行时的关键调用数据。
- [152] 在配置程序插桩后，启动待检测的应用软件，采集模块10获取插桩采集的流经敏感调用点的数据流信息。
- [153] 插桩采集的数据流信息包括应用软件运行过程中，调用此敏感调用点时传递给此敏感调用点的参数，以及敏感调用点根据传递的参数执行函数调用后输出的返回值等信息。
- [154] 需要说明的是，敏感调用点是应用软件中的关键调用点，或是安全性要求要高的调用点，可以由测试人员人为确定需检测的敏感调用点，在需检测的各敏感调用点配置插桩。
- [155] 判断模块20，用于根据所述数据流信息，判断所述敏感调用点是否被污点数据污染。
- [156] 需要说明的是，在通过外部输入点向应用软件输入污点数据后，污点数据会在应用软件运行的过程中，传递污染关联的参数，关联参数会被打上污染标签。
- [157] 则，若流经敏感调用点的参数携带有污点标签，则判断模块20可以判定此敏感

调用点被污点数据污染，也即，外部输入点输入的数据能够影响、控制此敏感调用点，存在风险。

[158] 回溯模块30，用于若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点。

[159] 若敏感调用点被污染，则回溯模块30首先需要确定敏感调用点的变量参数，例如，敏感调用点使用形参执行函数输出返回值，则敏感调用点的形参即为变量参数，需要追踪传递给敏感调用点、对形参赋值的实参。

[160] 然后，根据插桩采集的数据流信息中传递给此敏感调用点的实参，进行数据回溯，查找此变量参数数值的来源，以获取影响此变量参数的值的调用点，直至查找到的能够影响此变量参数的值的调用点为源节点，例如常量赋值、外部输入点等时，停止数据回溯。

[161] 需要说明的是，源节点参数不再受应用软件内部其他调用点的影响。

[162] 若能够影响此变量参数的值的调用点的源节点为外部输入点，则回溯模块30确定此外部输入点为敏感调用点对应的外部输入点，此外部输入点输入的数据能够影响、控制敏感调用点，给应用软件造成风险。

[163] 作为一种实施方式，所述回溯模块30还用于，

[164] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述源节点为常量赋值，则停止回溯根据，确定未找到所述敏感调用点对应的外部输入点。

[165] 例如，主函数调用当前敏感调用点时，传递给敏感调用点的参数为A，敏感调用点根据A执行函数调用后会对应输出返回值，则回溯模块30确定敏感调用点的形参为变量参数。

[166] 然后，回溯模块30根据传递给敏感调用点的参数A，查找应用软件中返回或输出A的前驱调用点，也即查找将A作为返回值输出且执行顺序早于敏感调用点的调用点。

[167] 需要说明的是，前驱调用点在应用软件的各调用点执行顺序中，早于对比调用

点执行函数调用。例如上述“查找输出A的前驱调用点”中，前驱调用点的函数执行顺序应当早于当前敏感调用点。

- [168] 若前驱调用点1输出的返回值为A，则回溯模块30判断前驱调用点1输出的参数是否为外部输入点输入的变量。
- [169] 若前驱调用点1输出的参数是外部输入点输入的变量，也即A是由前驱调用点1根据外部输入点输入的变量对应输出的，则回溯模块30可以确认此外部输入点为数据流的源节点，也就是当前敏感调用点对应的外部输入点。
- [170] 若前驱调用点1是根据输入的实参B，执行函数调用运算得到A，则根据B继续查找返回B的前驱调用点，也即查找将B作为返回值输出，且执行顺序早于前驱调用点1的调用点，以此类推，直至找到数据流的源节点。
- [171] 若源节点为外部输入点，则回溯模块30可以确定找到源节点为敏感调用点对应的外部输入点。
- [172] 若源节点为常量赋值，则回溯模块30可以确定未回溯到外部输入点，当前敏感调用点安全，停止回溯。
- [173] 进一步地，若回溯模块30找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [174] 回溯模块30在根据变量参数进行数据回溯时，可能会遇到多个调用点输出参数均为此变量参数，此时，回溯模块30首先根据各调用点的执行顺序，筛选出执行顺序先于敏感调用点的调用点，也可称为执行顺序早于敏感调用点的调用点。
- [175] 若筛选后，得到的调用点只有一个，则回溯模块30就可以确定此调用点为当前查找的、用于进一步进行数据回溯前驱调用点；若筛选后，得到的调用点有多个，则此时回溯模块30根据这多个调用点的执行顺序，确定执行顺序最晚、但早于敏感调用点的调用点为当前查找的、用于进一步进行数据回溯前驱调用点，也即确定执行顺序最接近敏感调用点的为前驱调用点。
- [176] 例如，若输出A的调用点包括调用点2、调用点3，则获取调用点2、调用点3的执行顺序。若调用点2先于调用点3执行，敏感调用点晚于调用点3执行，则调用

点3的执行顺序更加接近敏感调用点，确定调用点3为前驱调用点；若调用点3先于调用点2执行，敏感调用点晚于调用点2执行，则调用点2的执行顺序更加接近敏感调用点，确定调用点2为前驱调用点。

[177] 同理，在后续回溯数据时，若遇到多个调用点输出的参数均为当前回溯的参数、无法确定前驱调用点的情况时，选取执行顺序早于当前调用点、且最接近当前调用点的为前驱调用点，继续进行数据回溯。

[178] 由此，在有多个调用点输出变量参数时，回溯模块30可以唯一确定前驱调用点，实现了通过调用点时序，进行数据回溯和外部输入点的查找。

[179] 进一步地，作为另一种实施方式，所述回溯模块30还用于，

[180] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[181] 若敏感调用点的变量参数值是由前驱参数的赋值决定的，也即传递给敏感调用点、对变量参数进行赋值的参数是前驱调用点的赋值，则回溯模块30回溯对应的右值，也即对传递给敏感调用点的变量参数进行赋值的右值。

[182] 若此右值是外部输入点输入的变量，也即前驱调用点是根据外部输入点输入的变量对输出的参数进行赋值的，则可以确定此外部输入点为敏感调用点对应的外部输入点。

[183] 若右值为常量赋值，则回溯模块30可以确定当前数据链安全，未回溯到外部输入点，停止回溯。

[184] 由此，回溯模块30实现了对函数调用和赋值的数据追踪，保障了回溯调用点的完整性和准确性。

[185] 确定模块40，用于若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[186] 若成功回溯数据找到敏感调用点对应的外部输入点，则找到的外部输入点可能会被利用输入数据，调用此敏感调用点，给应用软件带来安全隐患。

[187] 由此，确定模块40可以确定此外部输入点为应用软件的安全漏洞，进行漏洞上

报。

[188] 在本实施例中，采集模块10基于预先在应用软件敏感调用点配置的插桩，采集应用软件运行过程中流经敏感调用点的数据流信息，进行动态的数据采集，模拟应用软件实际运行的状态，相对于静态数据来说采集的数据更具有真实性，能够更好的进行应用软件的安全性检测，提高安全漏洞检测效率；然后，判断模块20根据流经敏感调用点的数据流信息，判断敏感调用点是否被污点数据污染；若敏感调用点被污染，则敏感调用点可能被外部输入点关联调用，影响应用软件安全，此时，回溯模块30根据敏感调用点的变量参数，回溯查找敏感调用点对应的外部输入点，也即查找更够调用敏感调用点的外部输入点；若成功查找到敏感调用点对应的外部输入点，则确定模块40确定此找到的外部输入点为应用软件的安全漏洞，会给应用软件带来风险。本实施例中基于应用软件运行过程中的关键调用和数据处理操作的实时监控输出，通过离线自动污点数据传播分析，检测应用潜在的安全漏洞。通过采用动态插桩和离线数据流分析动静结合的检测技术，能够覆盖应用软件更深层次功能和逻辑检查，不存在漏报，且采集的数据来自实际运行过程，真实有效，不存在误报，大大提高了应用软件的安全漏洞检测效率。

[189] 进一步地，参照图7，本发明应用软件安全漏洞检测系统第二实施例提供一种应用软件安全漏洞检测系统，基于上述本发明应用软件安全漏洞检测系统第一实施例，所述应用软件安全漏洞检测系统还包括：

[190] 污点模块50，用于当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。

[191] 当应用软件的敏感调用点已经配置好插桩，启动运行时，或在应用软件运行的过程中，污点模块50通过应用软件的各个外部输入点，向应用软件中输入带有污染标签的污点数据。

[192] 污点数据再被输入应用软件后，跟随各调用点的调用，通过赋值、函数调用等方式，污染关联的参数。若被污染的参数流经调用点，则可认为此调用点被污点。

[193] 若应用软件运行的过程中敏感调用点被污染，则可以确定外部输入点输入的污

点数据的关联参数能够流经此敏感调用点进行函数调用，也即此敏感调用点能够被外部输入点调用，存在安全漏洞。然后，回溯模块30根据敏感调用点的数据流进行数据回溯，查找能够调用敏感调用点的外部输入点，此能够调用敏感调用点的外部输入点即为应用软件的安全漏洞。

[194] 在本实施例中，当所述应用软件运行时，污点模块50在应用软件的各外部输入点输入污点数据，用以检测敏感调用点是否被感染，从而根据被感染的敏感调用点查找对应的安全漏洞，实现了对应用软件漏洞的检测。本实施例通过向各外部输入点输入污点数据，使应用软件运行起来，能够快速、准确的标记出存在被外部输入点调用风险的敏感调用点，进而提高风险调用点的准确率，实现对应用软件深层逻辑的检测，减少漏洞的误报和漏报，提高漏洞检测的效率。

[195] 进一步地，本发明应用软件安全漏洞检测系统第三实施例提供一种应用软件安全漏洞检测系统，基于上述本发明应用软件安全漏洞检测系统第一实施例、第二实施例，所述确定模块40还用于，

[196] 确定回溯找到所述外部输入点的路径为危险路径。

[197] 当成功找到敏感调用点对应的外部输入点时，确定模块40将找到外部输入点的路径信息提取出来，得到从外部输入点输入的污点数据流向敏感调用点的路径，以及路径上被感染的各调用点。

[198] 然后，确定模块40标记此路径为危险路径，便于研发人员了解敏感调用点的具体风险，以提高应用软件的完全性。

[199] 在本实施例中，确定模块40确定回溯找到敏感调用点对应外部输入点的路径为危险路径，以便于研发人员进行进一步地漏洞原因分析、提升应用软件的安全性。

[200] 需要说明的是，在硬件实现上，以上采集模块10、判断模块20、回溯模块30、确定模块40以及污点模块50等可以以硬件形式内嵌于或独立于应用软件安全漏洞检测系统中，也可以以软件形式存储于应用软件安全漏洞检测系统的存储器中，以便于处理器调用执行以上各个模块对应的操作。该处理器可以为中央处理单元（CPU）、微处理器、单片机等。

[201] 如图8所示，图8是本发明实施例方案涉及的硬件运行环境的设备结构示意图。

- [202] 本发明实施例应用软件安全漏洞检测设备可以是PC，也可以是智能手机、平板电脑、电子书阅读器、便携计算机等终端设备。
- [203] 如图8所示，应用软件安全漏洞检测设备可以包括：处理器1001，例如CPU，以及存储器1002。这些组件之间的连接通信可以通过通信总线实现。存储器1002可以是高速RAM存储器，也可以是稳定的存储器（non-volatile memory），例如磁盘存储器。存储器1002可选的还可以是独立于前述处理器1001的存储装置。
- [204] 可选地，应用软件安全漏洞检测设备还可以包括用户接口、网络接口、摄像头、RF（Radio Frequency，射频）电路，传感器、音频电路、WiFi模块等等。用户接口可以包括显示屏（Display）、输入单元比如键盘（Keyboard），可选用户接口还可以包括标准的有线接口、无线接口。网络接口可选的可以包括标准的有线接口、无线接口（如WI-FI接口）。
- [205] 本领域技术人员可以理解，图8中示出的应用软件安全漏洞检测设备结构并不构成对应用软件安全漏洞检测设备的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。
- [206] 如图8所示，作为一种计算机存储介质的存储器1002中可以包括操作系统以及应用软件安全漏洞检测程序。其中，操作系统是管理和控制应用软件安全漏洞检测设备硬件与软件资源的程序，支持应用软件安全漏洞检测程序及其他软件或程序的运行。
- [207] 在图8所示的应用软件安全漏洞检测设备中，处理器1001可以用于执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [208] 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；
- [209] 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；
- [210] 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；
- [211] 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

- [212] 进一步地，处理器1001还可以执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [213] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；
- [214] 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；
- [215] 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [216] 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [217] 进一步地，处理器1001还可以执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [218] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [219] 进一步地，处理器1001还可以执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [220] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；
- [221] 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [222] 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [223] 进一步地，处理器1001还可以执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [224] 当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。
- [225] 进一步地，处理器1001还可以执行存储器1002中存储的应用软件安全漏洞检测程序，以实现以下步骤：
- [226] 确定回溯找到所述外部输入点的路径为危险路径。
- [227] 本发明应用软件安全漏洞检测设备的具体实施例与上述应用软件安全漏洞检测

方法和系统各实施例基本相同，在此不作赘述。

- [228] 本发明提供了一种计算机可读存储介质，所述计算机可读存储介质存储有一个或者多个程序，所述一个或者多个程序可被一个或者多个处理器执行，以实现以下步骤：
- [229] 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；
- [230] 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；
- [231] 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；
- [232] 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。
- [233] 进一步地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- [234] 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；
- [235] 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；
- [236] 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [237] 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [238] 进一步地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- [239] 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [240] 进一步地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- [241] 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；

- [242] 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- [243] 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [244] 进一步地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- [245] 当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。
- [246] 进一步地，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- [247] 确定回溯找到所述外部输入点的路径为危险路径。
- [248] 本发明计算机可读存储介质的具体实施例与上述应用软件安全漏洞检测方法和系统各实施例基本相同，在此不作赘述。
- [249] 还需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。
- [250] 上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质（如ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端设备（可以是手机，计算机，服务器，或者网络设备等）执行本发明各个实施例所述的方法。
- [251] 以上仅为本发明的可选实施例，并非因此限制本发明的专利范围，凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在

其他相关的技术领域，均同理包括在本发明的专利保护范围内。

权利要求书

- [权利要求 1] 一种应用软件安全漏洞检测方法，其特征在于，所述应用软件安全漏洞检测方法包括以下步骤：
- 基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；
- 根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；
- 若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；
- 若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。
- [权利要求 2] 如权利要求1所述的应用软件安全漏洞检测方法，其特征在于，所述根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤包括：
- 根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；
- 若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；
- 若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- 若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。
- [权利要求 3] 如权利要求2所述的应用软件安全漏洞检测方法，其特征在于，所述应用软件安全漏洞检测方法还包括：
- 若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。
- [权利要求 4] 如权利要求2所述的应用软件安全漏洞检测方法，其特征在于，所述根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点的步骤还包括：

若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；

若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 5] 如权利要求1所述的应用软件安全漏洞检测方法，其特征在于，所述基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息的步骤之前，还包括：
当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。

[权利要求 6] 如权利要求1所述的应用软件安全漏洞检测方法，其特征在于，所述确定所述外部输入点为所述应用软件的安全漏洞的步骤之后，还包括：
确定回溯找到所述外部输入点的路径为危险路径。

[权利要求 7] 一种应用软件安全漏洞检测系统，其特征在于，所述应用软件安全漏洞检测系统包括：
采集模块，用于基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；
判断模块，用于根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；
回溯模块，用于若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；
确定模块，用于若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[权利要求 8] 如权利要求7所述的应用软件安全漏洞检测系统，其特征在于，所述回溯模块还用于，
根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；若所

述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 9]

如权利要求8所述的应用软件安全漏洞检测系统，其特征在于，所述回溯模块还用于，

若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

[权利要求 10]

如权利要求8所述的应用软件安全漏洞检测系统，其特征在于，所述回溯模块还用于，

若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 11]

如权利要求7所述的应用软件安全漏洞检测系统，其特征在于，所述应用软件安全漏洞检测系统还包括：

污点模块，用于当所述应用软件运行时，在所述应用软件的各外部输入点输入污点数据。

[权利要求 12]

如权利要求7所述的应用软件安全漏洞检测系统，其特征在于，所述确定模块还用于，

确定回溯找到所述外部输入点的路径为危险路径。

[权利要求 13]

一种应用软件安全漏洞检测设备，其特征在于，所述应用软件安全漏洞检测设备包括处理器及存储器，所述存储器中存储有应用软件安全漏洞检测程序；

所述处理器用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；

根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；

若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；

若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[权利要求 14] 如权利要求13所述的应用软件安全漏洞检测设备，其特征在于，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；

若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；

若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 15] 如权利要求14所述的应用软件安全漏洞检测设备，其特征在于，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

[权利要求 16] 如权利要求14所述的应用软件安全漏洞检测设备，其特征在于，所述处理器还用于执行所述应用软件安全漏洞检测程序，以实现以下步骤：

若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；

若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 17] 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储有一个或者多个程序，所述一个或者多个程序可被一个或者多个处理器执行，以实现以下步骤：

基于预先在应用软件敏感调用点配置的插桩，采集所述应用软件运行过程中流经所述敏感调用点的数据流信息；

根据所述数据流信息，判断所述敏感调用点是否被污点数据污染；

若所述敏感调用点被污染，则根据所述敏感调用点的变量参数，回溯查找所述敏感调用点对应的外部输入点；

若成功查找到所述敏感调用点对应的外部输入点，则确定所述外部输入点为所述应用软件的安全漏洞。

[权利要求 18] 如权利要求17所述的计算机可读存储介质，其特征在于，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：

根据所述变量参数，回溯查找输出所述变量参数的前驱调用点；

若所述前驱调用点是基于函数调用输出的返回值，则再次回溯查找返回所述前驱调用点实参的前驱调用点，确定源节点；

若所述源节点为外部输入点，则确定所述外部输入点为所述敏感调用点对应的外部输入点；

若所述源节点为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

[权利要求 19] 如权利要求18所述的计算机可读存储介质，其特征在于，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：

若找到多个输出所述变量参数且执行顺序先于所述敏感调用点的调用点，则确定所述多个调用点中执行顺序最接近所述敏感调用点的为所述前驱调用点。

- [权利要求 20] 如权利要求18所述的计算机可读存储介质，其特征在于，所述一个或者多个程序可被所述一个或者多个处理器执行，还实现以下步骤：
- 若所述前驱调用点将所述变量参数作为赋值输出，则获取所述变量参数对应的右值；
- 若所述右值为外部输入点输入的变量，则确定所述外部输入点为所述敏感调用点对应的外部输入点；
- 若所述右值为常量赋值，则停止回溯，确定未找到所述敏感调用点对应的外部输入点。

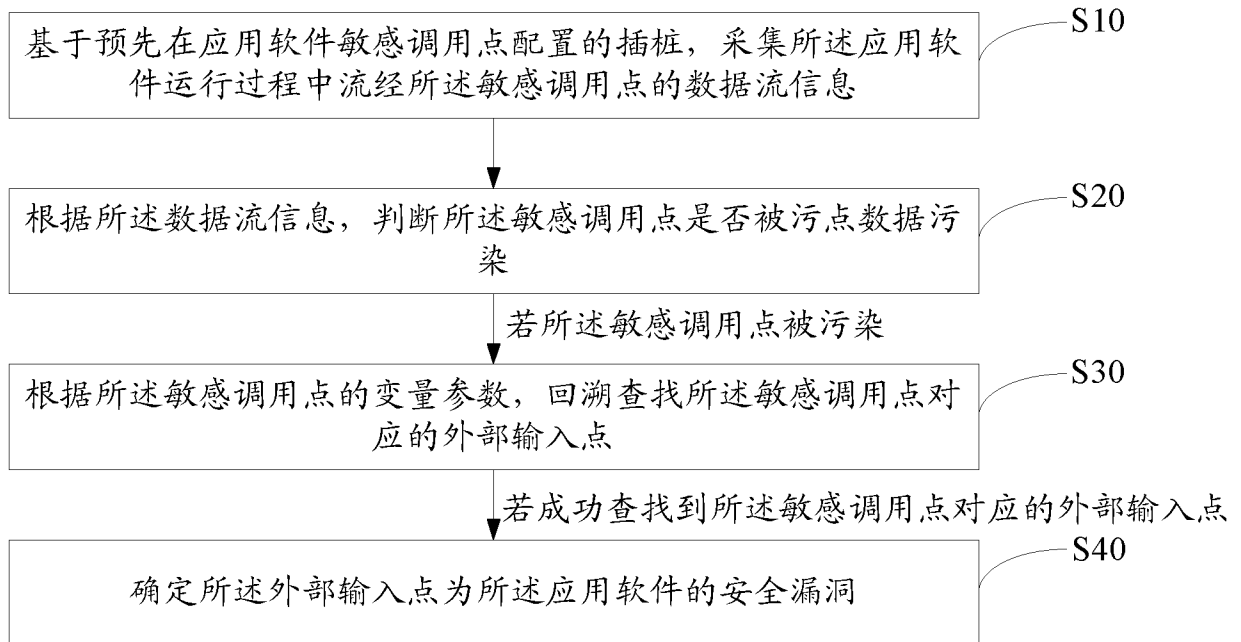


图 1

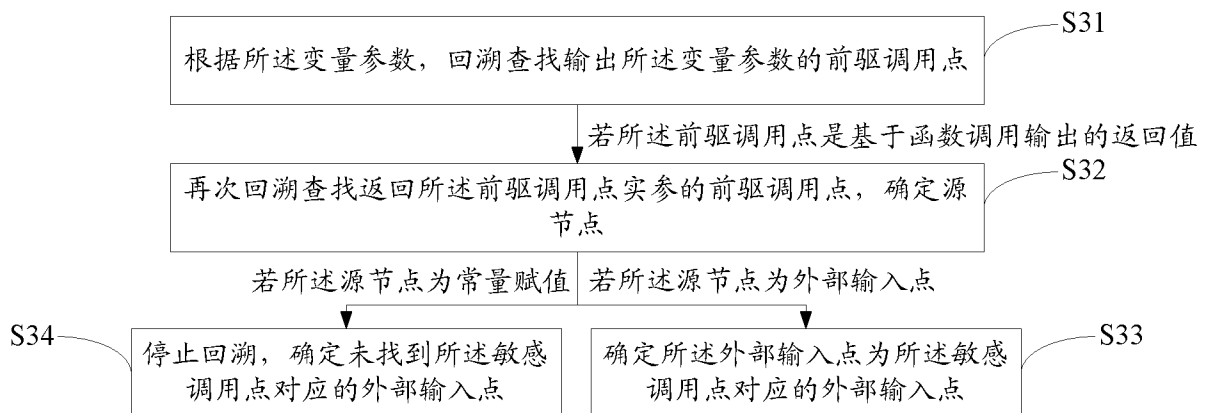


图 2

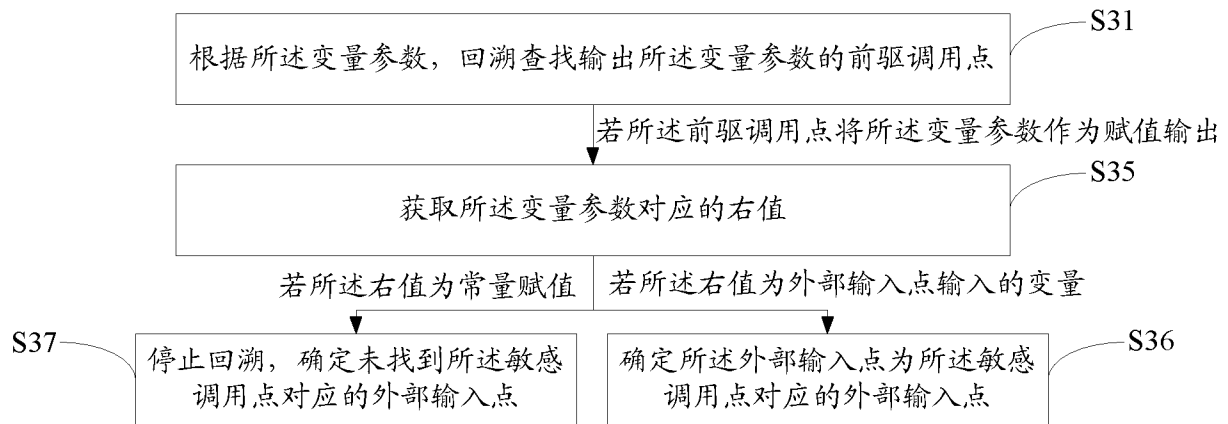


图 3

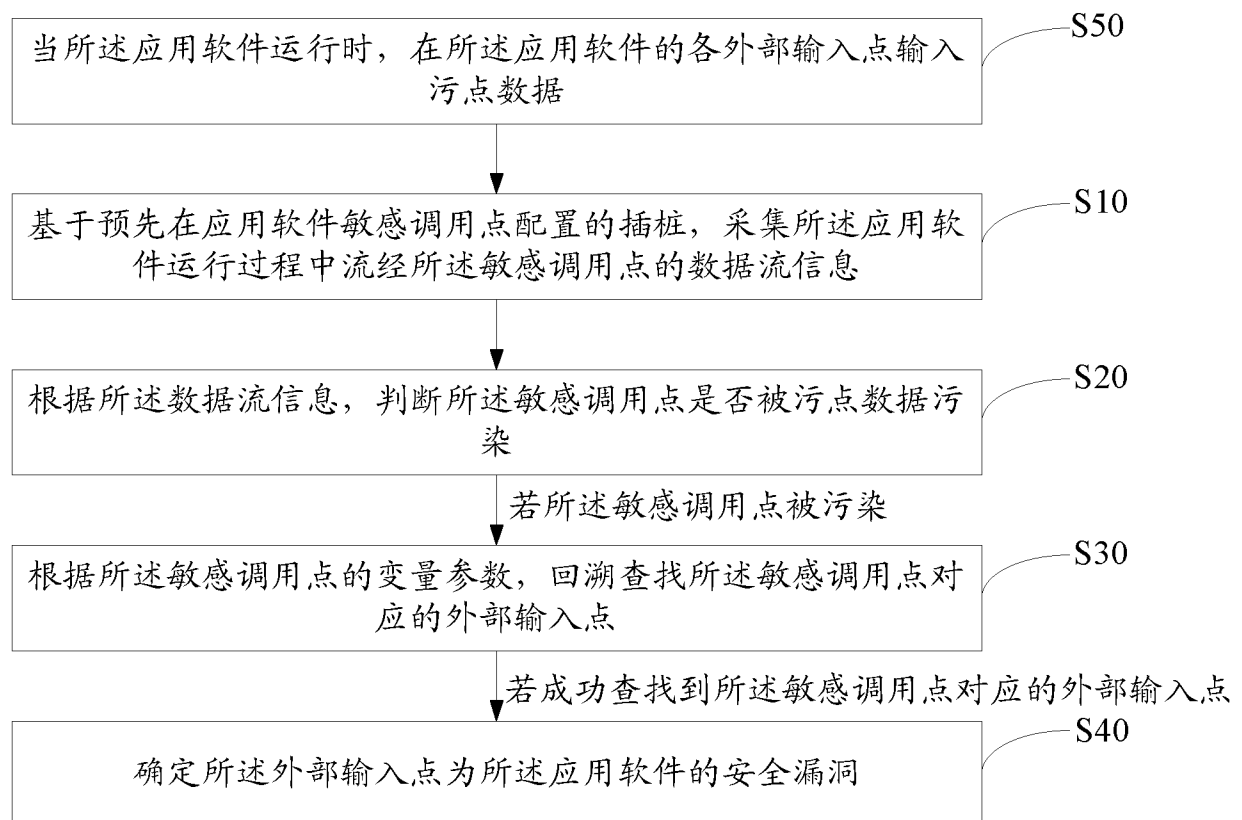


图 4

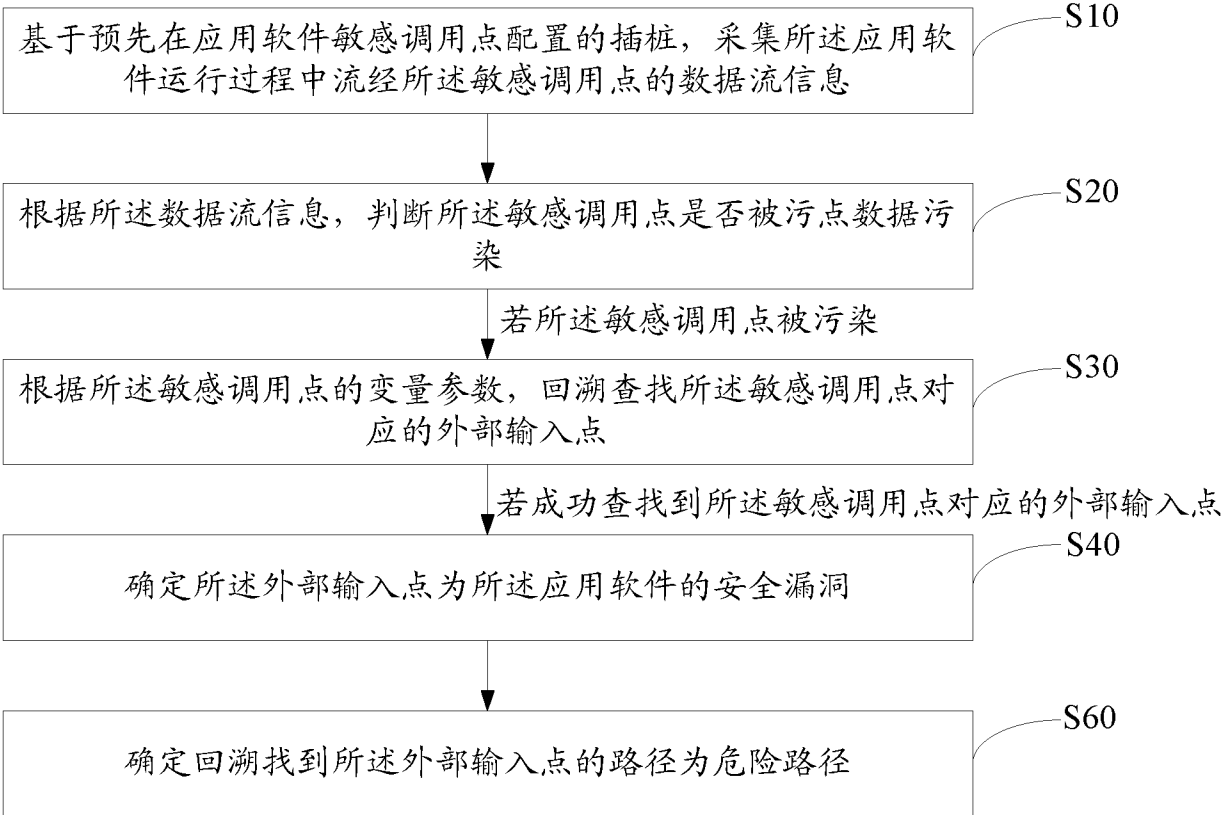


图 5

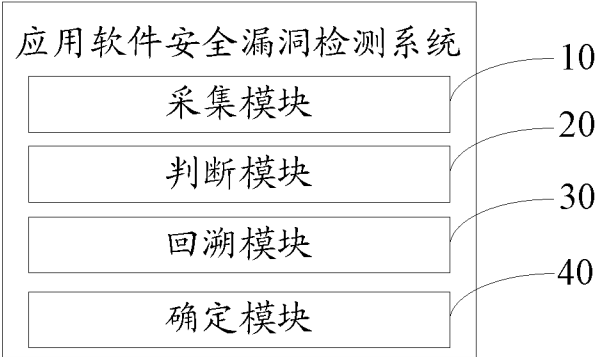


图 6



图 7

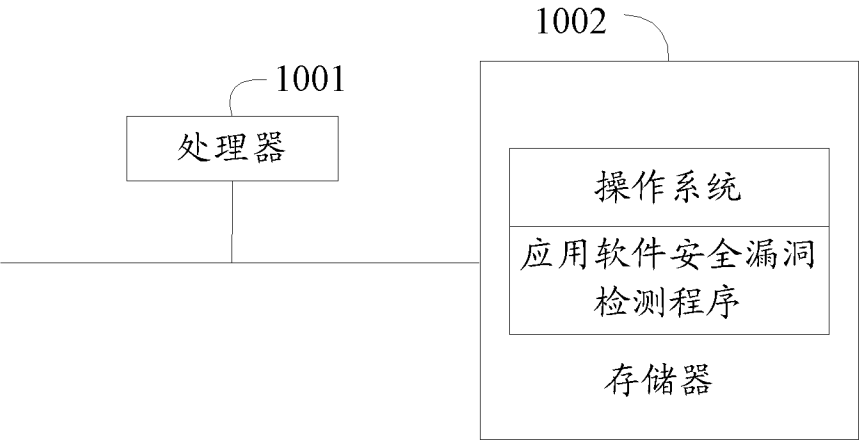


图 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/077984

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/57 (2013.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, CNKI, DWPI, SIPOABS: 安全, 检测, 检查, 漏洞, 污染, 插桩, 变量, 最近, safe, security, check+, examin+, bug, pollution, insert+, slot+, variable, nearest

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 106055980 A (NANJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS), 26 October 2016 (26.10.2016), description, paragraphs [0040]-[0057]	1-20
Y	CN 104462981 A (TENCENT INC.), 25 March 2015 (25.03.2015), description, paragraphs [0028]-[0029]	1-20
A	CN 102662840 A (TIANJIN UNIVERSITY), 12 September 2012 (12.09.2012), entire document	1-20
A	CN 103207969 A (BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.), 17 July 2013 (17.07.2013), entire document	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 04 August 2017	Date of mailing of the international search report 10 August 2017
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer FENG, Huiping Telephone No. (86-10) 62411838

International application No.
PCT/CN2017/077984

Form PCT/ISA/210 (patent family annex) (July 2009)

国际检索报告

国际申请号

PCT/CN2017/077984

A. 主题的分类 G06F 21/57 (2013.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类																	
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNABS、CNTXT、CNKI、DWPI、SIPOABS: 安全、检测、检查、漏洞、污染、插桩、变量、最近 safe、security、check+、examin+、bug、pollution、insert+、slot+、variable、nearest																	
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>CN 106055980 A (南京邮电大学) 2016年 10月 26日 (2016 - 10 - 26) 说明书第 [0040] - [0057] 段</td> <td>1-20</td> </tr> <tr> <td>Y</td> <td>CN 104462981 A (深圳市腾讯计算机系统有限公司) 2015年 3月 25日 (2015 - 03 - 25) 说明书第 [0028] - [0029] 段</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 102662840 A (天津大学) 2012年 9月 12日 (2012 - 09 - 12) 全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 103207969 A (百度在线网络技术北京有限公司) 2013年 7月 17日 (2013 - 07 - 17) 全文</td> <td>1-20</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	Y	CN 106055980 A (南京邮电大学) 2016年 10月 26日 (2016 - 10 - 26) 说明书第 [0040] - [0057] 段	1-20	Y	CN 104462981 A (深圳市腾讯计算机系统有限公司) 2015年 3月 25日 (2015 - 03 - 25) 说明书第 [0028] - [0029] 段	1-20	A	CN 102662840 A (天津大学) 2012年 9月 12日 (2012 - 09 - 12) 全文	1-20	A	CN 103207969 A (百度在线网络技术北京有限公司) 2013年 7月 17日 (2013 - 07 - 17) 全文	1-20
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
Y	CN 106055980 A (南京邮电大学) 2016年 10月 26日 (2016 - 10 - 26) 说明书第 [0040] - [0057] 段	1-20															
Y	CN 104462981 A (深圳市腾讯计算机系统有限公司) 2015年 3月 25日 (2015 - 03 - 25) 说明书第 [0028] - [0029] 段	1-20															
A	CN 102662840 A (天津大学) 2012年 9月 12日 (2012 - 09 - 12) 全文	1-20															
A	CN 103207969 A (百度在线网络技术北京有限公司) 2013年 7月 17日 (2013 - 07 - 17) 全文	1-20															
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。																	
<table border="0"> <tr> <td> * 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 </td> <td> “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件 </td> </tr> </table>			* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件													
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																
国际检索实际完成的日期	国际检索报告邮寄日期																
2017年 8月 4日	2017年 8月 10日																
ISA/CN的名称和邮寄地址	授权官员																
中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10) 62019451	冯慧萍 电话号码 (86-10) 62411838																

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2017/077984

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	106055980	A	2016年 10月 26日	无	
CN	104462981	A	2015年 3月 25日	无	
CN	102662840	A	2012年 9月 12日	无	
CN	103207969	A	2013年 7月 17日	CN 103207969 B	2016年 10月 5日