(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 104508672 A
(43)申请公布日 2015.04.08

(21)申请号 201380039842.1
(22)申请日 2013.07.25
(30)优先权数据
2012-171164 2012.08.01 JP
(85)PCT国际申请进入国家阶段日
2015.01.27
(86)PCT国际申请的申请数据
PCT/JP2013/070188 2013.07.25
(87)PCT国际申请的公布数据
WO2014/021190 JA 2014.02.06
(71)申请人 三菱电机株式会社
地址 日本东京都
(72)发明人 植田武 樱井钟治
(74)专利代理机构 北京三友知识产权代理有限公司 11127
代理人 李辉 马建军
(51)Int.Cl.
G06F 21/12(2006.01)

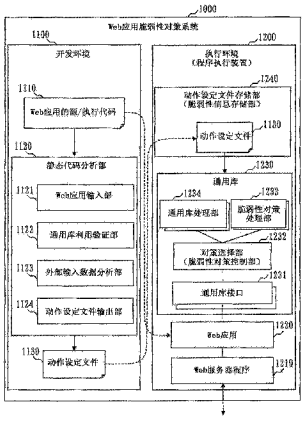
权利要求书2页 说明书12页 附图9页

(54)发明名称

程序执行装置以及程序分析装置

(57)摘要

本发明能够在瞄准脆弱性的攻击被实施以前可靠地执行针对脆弱性的对策处理。脆弱性对策处理部(1233)执行针对脆弱性库函数的脆弱性的对策处理，该脆弱性库函数是通用库(1230)包含的通用库函数中具有脆弱性的通用库函数。当在执行Web应用(1220)时被请求调用脆弱性库函数的情况下，对策选择部(1232)使脆弱性对策处理部(1233)执行针对脆弱性库函数的脆弱性的对策处理，在由脆弱性对策处理部(1233)执行了对策处理后，调用脆弱性库函数。



CN 104508672 A

1. 一种程序执行装置，其执行使用库函数的程序，其特征在于，该程序执行装置具有：

脆弱性对策处理部，其执行针对脆弱性库函数的脆弱性的对策处理，该脆弱性库函数是在所述程序中使用的库函数中具有脆弱性的库函数；以及

脆弱性对策控制部，当在执行所述程序时被请求调用所述脆弱性库函数的情况下，该脆弱性对策控制部使所述脆弱性对策处理部执行针对所述脆弱性库函数的脆弱性的对策处理，在由所述脆弱性对策处理部执行了对策处理后，调用所述脆弱性库函数。

2. 根据权利要求 1 所述的程序执行装置，其特征在于，

所述程序执行装置还具有存储脆弱性信息的脆弱性信息存储部，该脆弱性信息表示由所述脆弱性库函数处理的参数值中的非法值，

在被请求调用所述脆弱性库函数的情况下，所述脆弱性对策控制部判断由于该调用而由所述脆弱性库函数处理的参数值是否属于所述非法值，

当由所述脆弱性库函数处理的参数值属于所述非法值的情况下，所述脆弱性对策控制部使所述脆弱性对策处理部执行使得所述脆弱性库函数不处理所述非法值的对策处理。

3. 根据权利要求 2 所述的程序执行装置，其特征在于，

在被请求调用所述脆弱性库函数以外的库函数的情况下，所述脆弱性对策控制部不使所述脆弱性对策处理部执行对策处理即调用作为对象的库函数，

在被请求调用所述脆弱性库函数且由于该调用而由所述脆弱性库函数处理的参数值不属于所述非法值的情况下，所述脆弱性对策控制部不使所述脆弱性对策处理部执行对策处理即调用所述脆弱性库函数。

4. 根据权利要求 1 ～ 3 中的任意一项所述的程序执行装置，其特征在于，

所述程序执行装置与多个脆弱性库函数对应地具有多个脆弱性对策处理部，

当在执行所述程序时被请求调用任意一个脆弱性库函数的情况下，所述脆弱性对策控制部选择与作为调用对象的脆弱性库函数对应的脆弱性对策处理部，

所述脆弱性对策控制部使选择出的脆弱性对策处理部执行针对脆弱性的对策处理。

5. 根据权利要求 1 ～ 4 中的任意一项所述的程序执行装置，其特征在于，

所述程序执行装置使所述脆弱性库函数处理记述在从外部装置接收到的接收数据中的参数值，

所述程序执行装置还具有存储脆弱性信息的脆弱性信息存储部，该脆弱性信息表示由所述脆弱性库函数处理的参数值中的非法值，并且，按照由所述脆弱性库函数处理的参数值在接收数据中的每个记述位置表示对策处理，

在随着从外部装置接收到接收数据而被请求调用所述脆弱性库函数、且由于该调用而由所述脆弱性库函数处理的参数值属于所述非法值的情况下，所述脆弱性对策控制部判定该参数值在接收数据中的记述位置，使所述脆弱性对策处理部执行与该参数值的记述位置对应的对策处理。

6. 根据权利要求 1 ～ 5 中的任意一项所述的程序执行装置，其特征在于，

所述脆弱性对策处理部和所述脆弱性对策控制部包含在通用库中。

7. 一种程序分析装置，其特征在于，

该程序分析装置分析程序和在所述程序中使用的库函数，提取相对于外部输入数据具有脆弱性的库函数作为脆弱性库函数，并且，使用所述程序的设定文件的值、所述脆弱性库

函数以及利用外部输入数据的所述程序内的代码位置，生成用于检查所述设定文件的值是否恰当的试验数据和脚本中的至少任意一方。

程序执行装置以及程序分析装置

技术领域

[0001] 本发明涉及进行针对程序的脆弱性的对策处理的技术。

背景技术

[0002] 过去，通过攻击 OS(Operating System :操作系统）和应用程序（以下简称作“应用”）等中存在的安全上的脆弱性，产生数据和程序的篡改、计算机和应用的强制停止、对计算机的非法入侵和非法操作等危害。

[0003] 近年来，对在 Web 服务器上运行的 Web 应用的攻击不断增加，机密信息和个人信息的外漏和篡改、服务的非法利用等成为社会问题。

[0004] 对 Web 应用的攻击是通过在作为针对 Web 应用的来自客户端的输入数据的 HTTP(Hyper Text Transfer Protocol :超文本传输协议）的请求中混入非法数据而进行的。

[0005] Web 应用的脆弱性根据利用输入数据的处理内容而不同，公知有跨站点脚本攻击和 SQL 注入等多种类型。

[0006] 例如，跨站点脚本攻击是利用在动态地生成 HTML(Hyper Text Markup Language :超文本标记语言）数据的处理中的脆弱性的攻击，由于该脆弱性导致允许脚本混入 HTML 数据中。

[0007] 另外，SQL 注入是利用发行 SQL 语句的处理的脆弱性的攻击，该 SQL 语句用于进行关系数据库的数据操作，由于该脆弱性导致允许应用执行未假定的 SQL 语句。

[0008] 用于攻击脆弱性的非法数据根据脆弱性的类型而不同，其对策方法也不同。

[0009] 例如，作为用于进行跨站点脚本攻击的方法之一，有将 <SCRIPT> 标签作为非法数据包含在请求中的方法。

[0010] 作为跨站点脚本攻击的对策，按照 HTML 语法进行无害化处理，使得在生成的 HTML 中不将 <SCRIPT> 标签解释成 <SCRIPT> 标签。

[0011] 另外，在 SQL 注入中有这样的方法，通过混入用于将字符串括起来的“’”来终结字符串，然后混入任意的 SQL 语句。

[0012] 这种针对 SQL 注入的对策是按照 SQL 的语法进行无害化处理，使得输入数据中包含的“’”不表示将字符串括起来的记号。

[0013] 过去，为了防护对 Web 应用的脆弱性的攻击，主要采用以下两种对策。

[0014] 一种方法是验证产生各种脆弱性的输入数据，将使非法数据无害化的安全功能，按照可能产生脆弱性的处理内容编入 Web 应用自身。

[0015] 第二种防护方法是在到达 Web 应用之前检查对 Web 应用的 HTTP 请求，并利用进行有可能产生攻击的 HTTP 请求的阻挡或无害化的 Web 应用防火墙。

[0016] 关于第一种防护方法，公开有 OWASP(The Open Web Application Security Project :打开 Web 应用程序安全性项目）等组织编入的安全功能等的指南。

[0017] 另外，作为支持安全功能编入的方法，还公知有通过分析程序的源代码来检测有

可能产生脆弱性的部位的方法（例如，专利文献 1）。

[0018] 关于第二种防护方法的 Web 应用防火墙，通常的方法是验证是否包含事前登记的非法数据的模式，在 HTTP 请求中包含非法数据的模式的情况下，进行通信的切断和数据的无害化等（例如，非专利文献 1）。

[0019] 另外，还有使用实施各种脆弱性对策的程序库的方法（例如，专利文献 2）。

[0020] 在该方法中，读取记述有是否需要针对研发者生成的请求的各参数的各种脆弱性对策的设定文件，利用按照设定文件的内容在接收 HTTP 请求时实施对每个参数指定的脆弱性对策的程序库，能够高效地实施 Web 应用的脆弱性对策。

[0021] 现有技术文献

[0022] 专利文献

[0023] 专利文献 1 ：日本特开 2007-52625 号公报

[0024] 专利文献 2 ：日本特开 2007-47884 号公报

[0025] 非专利文献

[0026] 非专利文献 1 ：ModSecurity<URL:http://www.modsecurity.org/>

发明内容

[0027] 发明要解决的问题

[0028] 过去静态地分析 Web 应用内的源代码的方法（专利文献 1）存在如下问题，虽然检测有可能产生脆弱性的部位，但是针对脆弱性的对策方法的研究和修正必须由 Web 应用的研发者独立进行。

[0029] 另外，利用 Web 应用防火墙采取对策的方法（非专利文献 1）没有考虑 Web 应用进行的处理内容。

[0030] 因此，在特定类型的脆弱性根据 Web 应用的处理内容不成问题的情况下、已在 Web 应用中实施了脆弱性对策的情况下，导致虽然属于非法数据的模式但是没有危害的数据也被误检测成非法数据。

[0031] 因此，存在如下问题 ：在 Web 应用中切断实质上没有问题的通信，由于无害化处理而产生可正常处理的数据被变换的不良影响。

[0032] 另外，对请求的每个参数采取对策的方法（专利文献 2），需要由研发者对每个参数设定设定文件，存在花费功夫的问题。

[0033] 另外，由于设定错误等，有可能导致脆弱性的对策遗漏或选择不必要的对策。

[0034] 另外，被指定的脆弱性对策与 Web 应用防火墙同样是在接收 HTTP 请求时一并进行的，还存在已实施的脆弱性对策对其它有可能存在脆弱性的处理带来不良影响的问题。

[0035] 本发明将解决上述问题作为主要目的，其主要目的在于，得到能够在瞄准脆弱性的攻击被实施以前可靠地执行针对脆弱性的对策处理，防止对策遗漏，避免基于不必要的对策处理的不良影响的结构。

[0036] 用于解决问题的手段

[0037] 本发明的程序执行装置执行使用库函数的程序，其特征在于，该程序执行装置具有 ：

[0038] 脆弱性对策处理部，其执行针对脆弱性库函数的脆弱性的对策处理，该脆弱性库

函数是在所述程序中使用的库函数中具有脆弱性的库函数；以及

[0039] 脆弱性对策控制部，当在执行所述程序时被请求调用所述脆弱性库函数的情况下，该脆弱性对策控制部使所述脆弱性对策处理部执行针对所述脆弱性库函数的脆弱性的对策处理，在由所述脆弱性对策处理部执行了对策处理后，调用所述脆弱性库函数。

[0040] 发明效果

[0041] 根据本发明，当在执行程序时被请求调用脆弱性库函数的情况下，执行针对该脆弱性库函数的脆弱性的对策处理，因而能够在瞄准脆弱性的攻击被实施以前可靠地执行针对脆弱性的对策处理，能够防止对策遗漏，避免基于不必要的对策处理的不良影响。

附图说明

[0042] 图 1 是表示实施方式 1 的系统结构例的图。

[0043] 图 2 是表示实施方式 1 的 Web 应用脆弱性对策系统的结构例的图。

[0044] 图 3 是表示实施方式 1 的开发环境中的动作例的流程图。

[0045] 图 4 是表示实施方式 1 的执行环境中的动作例的流程图。

[0046] 图 5 是表示实施方式 2 的 Web 应用脆弱性对策系统的结构例的图。

[0047] 图 6 是表示实施方式 2 的开发环境中的动作例的流程图。

[0048] 图 7 是表示实施方式 3 的 Web 应用脆弱性对策系统的结构例的图。

[0049] 图 8 是表示实施方式 3 的开发环境中的动作例的流程图。

[0050] 图 9 是表示实施方式 1 ～ 3 的 Web 应用脆弱性对策系统的硬件结构例的图。

具体实施方式

[0051] 实施方式 1

[0052] 在本实施方式及以后的实施方式中说明的 Web 应用脆弱性对策系统，能够自动选择充分必要的 Web 应用的安全对策，防止对策遗漏和避免基于脆弱性对策的不良影响。

[0053] 图 1 是包含本实施方式的 Web 应用脆弱性对策系统的系统结构图。

[0054] 如图 1 所示，本实施方式的系统由 Web 应用脆弱性对策系统 1000、客户终端 2000、网络 3000（因特网等网络）构成。

[0055] Web 应用脆弱性对策系统 1000 和多个客户终端 2000 通过网络 300 连接。

[0056] 另外，图 2 是本实施方式的 Web 应用脆弱性对策系统 1000 的结构图。

[0057] 另外，在图 2 中，Web 应用脆弱性对策系统 1000 具有 CPU(Central Processing Unit：中央处理单元）、存储器、二次存储装置等硬件资源，但省略图示。

[0058] 并且，在二次存储装置中存储有实现以下说明的功能的程序，在执行程序时，将程序从二次存储装置下载到存储器，CPU 执行下载到存储器的程序。

[0059] 在图 2 中，Web 应用脆弱性对策系统 1000 由开发环境 1100 和执行环境 1200 构成。

[0060] 开发环境 1100 是用于研发者安装 Web 应用进行试验的环境。

[0061] 执行环境 1200 是用于执行已开始运行的 Web 应用的环境。

[0062] 执行环境 1200 相当于程序执行装置的示例。

[0063] 开发环境 1100 具有静态代码分析部 1120。

[0064] 静态代码分析部 1120 读取 Web 应用的源代码或者执行代码（记作“Web 应用的源

/ 执行代码 1101”），对 Web 应用的源 / 执行代码 1101 进行静态分析，并输出脆弱性对策用的动作设定文件 1130。

[0065] 另外，静态代码分析部 1120 由 Web 应用输入部 1121、通用库利用验证部 1122、外部输入数据分析部 1123 和动作设定文件输出部 1124 构成。

[0066] Web 应用输入部 1121 读取 Web 应用的源 / 执行代码 1101。

[0067] 通用库利用验证部 1122 根据对 Web 应用的源 / 执行代码 1101 的静态代码分析，调查正在调用通用库函数的代码位置和通用库接口 1231 的类型，并验证通用库函数的利用方法是否正确。

[0068] 外部输入数据分析部 1123 根据静态代码分析来分析来自 Web 应用外部的输入数据（外部输入数据）的数据流，并分析利用该外部输入数据的代码位置和可取的值（数据的值和数据的类型）。

[0069] 动作设定文件输出部 1124 根据通用库利用验证部 1122 和外部输入数据分析部 1123 的分析结果，输出用于决定在执行通用库时的动作的动作设定文件 1130。

[0070] 另外，在动作设定文件 1130 中记述有具有脆弱性的通用库函数（以下称作“脆弱性库函数”）、在利用脆弱性库函数进行处理时可能发生异常的参数的名称和该参数中的非法值、以及调用脆弱性库函数的代码位置中可能发生异常的代码位置。

[0071] 动作设定文件 1130 相当于脆弱性信息的示例。

[0072] 另外，执行环境 1200 由 Web 服务器程序 1210、Web 应用 1220、通用库 1230、动作设定文件存储部 1240 构成。

[0073] Web 服务器程序 1210 是用于执行 Web 应用 1220 的程序。

[0074] Web 应用 1220 调用通用库函数，并得到通用库函数的执行结果。

[0075] 通用库 1230 提供用于进行以下处理的功能 :HTTP 请求的发送接收、HTML 输出处理、对数据库的访问、文件的读写等在普通的 Web 应用中进行的处理。

[0076] 动作设定文件存储部 1240 存储从开发环境 1100 的静态代码分析部 1120 输出的动作设定文件 1130。

[0077] 动作设定文件存储部 1240 相当于脆弱性信息存储部的示例。

[0078] 通用库 1230 由通用库接口（接口）1231、对策选择部 1232、脆弱性对策处理部 1233、通用库处理部 1234 构成。

[0079] 通用库接口 1231 提供用于 Web 应用 1220 利用通用库 1230 的程序接口。

[0080] 另外，通用库接口 1231 是按照 HTML 输出处理和数据库访问等处理的每个分类而准备的。

[0081] 脆弱性对策处理部 1233 进行针对 Web 应用的脆弱性的对策处理。

[0082] 另外，按照脆弱性的每种对策准备多个脆弱性对策处理部 1233。

[0083] 即，脆弱性对策处理部 1233 是按照 HTML 输出处理和数据库访问等处理的每个分类而准备的。

[0084] 通用库处理部 1234 进行通用库 1230 提供的处理 (HTML 输出处理和数据库访问等）。

[0085] 通用库处理部 1234 相当于通用库函数。

[0086] 另外，按照处理内容准备多个通用库处理部 1234。

[0087] 对策选择部 1232 读取动作设定文件 1130，判定在通用库 1230 内部是否需要脆弱性的对策处理。

[0088] 更具体地讲，在 Web 应用 1220 请求调用通用库函数的情况下，即在被请求调用通用库处理部 1234 的情况下，对策选择部 1232 根据动作设定文件 1130 的内容，判定是否需要脆弱性的对策处理。

[0089] 另外，在需要脆弱性的对策处理的情况下，对策选择部 1232 首先使脆弱性对策处理部 1233 进行脆弱性的对策处理，在由脆弱性对策处理部 1233 进行了对策处理后，调用作为对象的通用库处理部 1234。

[0090] 对策选择部 1232 相当于脆弱性对策控制部的示例。

[0091] 下面，说明在 Web 应用脆弱性对策系统 1000 中进行的动作。

[0092] 在 Web 应用脆弱性对策系统 1000 中，首先由开发环境 1100 的静态代码分析部 1120 对 Web 应用的源 / 执行代码 1110 进行静态代码分析，并生成动作设定文件 1130。

[0093] 然后，对 Web 应用的源 / 执行代码 1110 进行编译而使其能够执行，将 Web 应用 1220 配置在执行环境 1200 中，并且将动作设定文件 1130 配置在执行环境 1200 中，Web 应用 1220 的运行开始。

[0094] 使用图 3 说明在开发环境 1100 中的动作例。

[0095] 在开发环境 1100 中，Web 应用输入部 1121 读取 Web 应用的源 / 执行代码 1110(S101)。

[0096] 然后，通用库利用验证部 1122 调查正在调用 Web 应用的源 / 执行代码 1110 利用的通用库函数的代码位置（行数等）、被调用的通用库函数的类型 (HTML 输出、数据库访问等)，然后验证对通用库函数提供的数据和使用方式是否恰当 (S102)。

[0097] 在判定为通用库函数被恰当地利用的情况下 (S102 :是)，外部输入数据分析部 1123 确定存储来自外部的输入数据的变量（外部输入数据），调查外部输入数据的数据流，并调查利用外部输入数据的一系列的代码位置和各代码中的外部输入数据可取的值的范围 (S103)。

[0098] 在判定为通用库函数未被恰当地利用的情况下 (S102 :否)，输出消息并结束处理，该消息包含未被恰当地利用的通用库函数的类型、代码上的位置、判定为未被恰当地利用的理由等 (S104)。

[0099] 在 S103 之后，动作设定文件输出部 1124 根据在 S102 中分析出的正在调用通用库函数的代码位置和被调用的通用库函数的类型、以及在 S103 中分析出的利用外部输入数据的一系列的代码位置和各代码中的外部输入数据可取的值的结果，输出动作设定文件 1130(S105)。

[0100] 该动作设定文件 1130 是用于选择对策的文件，包含调用有可能由于外部输入数据可取的值而产生脆弱性的通用库函数的代码位置。

[0101] 对 S105 的处理进行更具体的说明，提取在 S102 中分析出的正在调用通用库函数的代码位置中、在 S103 中分析出的利用外部输入数据的一系列的代码位置中包含的代码位置，按照外部输入数据的每个值，判定在向该代码位置的代码正在调用的通用库函数输入了在 S103 中分析出的外部输入数据可取的值的范围时脆弱性是否成为问题（是否发生异常）。

[0102] 并且，在判定为脆弱性成为问题的情况下，动作设定文件输出部 1124 生成动作设定文件 1130，在动作设定文件 1130 中记述有具有脆弱性的通用库函数（脆弱性库函数）的名称、调用脆弱性库函数的代码位置、外部输入数据的类型（变量的名称）、外部输入数据的非法值（发生异常的值）。

[0103] 并且，将生成的动作设定文件 1130 输出到动作设定文件存储部 1240，并存储在动作设定文件存储部 1240 中。

[0104] 下面，使用图 4 说明在执行环境 1200 中的动作例。

[0105] 首先，Web 服务器程序 1210 接收来自客户终端 2000 的 HTTP 请求 (S201)。

[0106] Web 服务器程序 1210 将接收到的 HTTP 请求转发给 Web 应用 1220(S202)。

[0107] Web 应用 1220 调用通用库接口 1231，在被请求调用通用库函数的情况下 (S203 ：是)，被调用的通用库接口 1231 调用对策选择部 1232(S204)。

[0108] 对策选择部 1232 读取动作设定文件 1130，从动作设定文件 1130 取得有可能混入使脆弱性明显化的外部输入数据的通用库函数（脆弱性库函数）的调用代码位置的信息 (S205)。

[0109] 然后，对策选择部 1232 根据堆栈轨迹等执行信息取得在 S203 中调用通用库接口 1231 以前执行的代码位置的信息 (S206)。

[0110] 然后，对策选择部 1232 判定是否需要执行脆弱性对策处理部 1233(S207)。

[0111] 具体而言，如果在 S205 中取得的代码位置中存在与在 S206 中取得的代码位置一致的代码位置，则从动作设定文件 1130 取得和与在 S206 中取得的代码位置一致的代码位置对应起来的脆弱性库函数的名称、外部输入数据的类型、外部输入数据的非法值。

[0112] 然后，对策选择部 1232 核对已取得的脆弱性库函数的名称和外部输入数据的类型和外部输入数据的非法值、与 Web 应用 1220 将要调用的通用库函数的名称和在该通用库函数中将要处理的外部输入数据的类型和外部输入数据的值，在 3 个要素全部一致的情况下，判定为需要执行脆弱性对策处理部 1233。

[0113] 另一方面，在 3 个要素中至少一个不一致的情况下，判定为不需要执行脆弱性对策处理部 1233。

[0114] 另外，对策选择部 1232 将与 Web 应用 1220 将要调用的脆弱性库函数对应的脆弱性对策处理部 1233 指定为调用的对象。

[0115] 对策选择部 1232 在判定为需要调用脆弱性对策处理部 1233 的情况下 (S207 ：是)，在进行通用库处理部 1234 的调用之前，进行脆弱性对策处理部 1233 的调用，使脆弱性对策处理部 1233 执行使得通用库处理部 1234 不处理外部输入数据的非法值的对策处理 (S208)。

[0116] 具体而言，使脆弱性对策处理部 1233 执行外部输入数据的无害化等对策处理。

[0117] 在脆弱性对策处理部 1233 执行的对策处理结束后，对策选择部 1232 调用通用库处理部 1234(S209)。

[0118] 在判定为不需要调用脆弱性对策处理部 1233 的情况下 (S207 ：否)，对策选择部 1232 不进行脆弱性对策处理部 1233 的调用而进行通用库处理部 1234 的调用 (S209)。

[0119] 每当在 Web 服务器程序 1210 向客户终端 2000 发送响应之前通用库接口 1231 被调用时，执行以上的步骤 S203 ～ S209(S210)。

[0120] 并且，在 Web 服务器程序 1210 向客户终端 2000 发送 HTTP 响应的情况下，Web 应用 1220 将 HTTP 响应转发给 Web 服务器程序 1210(S211)。

[0121] 并且，最后，Web 服务器程序 1210 向客户终端 2000 答复 HTTP 响应 (S212)。

[0122] 另外，在开发环境 100 中，在静态代码分析部 1120 使用 Web 应用的源代码生成了动作设定文件 1130 的情况下，在动作设定文件 1130 中记述有源代码上的代码位置（行编号等）。

[0123] 另一方面，在执行环境 1200 中，对策选择部 1232 根据堆栈轨迹等执行信息取得编译后的 Web 应用的代码位置。

[0124] 因此，在上述的 S207 中，更具体地讲，对策选择部 1232 将动作设定文件 1130 在源代码上的代码位置变换成编译后的代码位置，并核对变换后的代码位置和根据堆栈轨迹等取得的编译后的代码位置。

[0125] 或者，对策选择部 1232 将根据堆栈轨迹等取得的编译后的代码位置变换成源代码上的代码位置，并核对变换后的代码位置和动作设定文件 1130 在源代码上的代码位置。

[0126] 这样的代码位置的变换能够利用现有技术实现。

[0127] 如上所述，在本实施方式中提供具有如下功能的 Web 应用执行环境：在开始运行前对 Web 应用的源代码或者执行代码进行静态代码分析，根据通过静态代码分析而得到的数据流自动选择脆弱性对策。

[0128] 由此，能够自动选择充分必要的 Web 应用的脆弱性对策，因而能够防止对策遗漏，并且不执行不必要的对策处理，因而能够使基于对策处理的不良影响和处理速度的性能恶化为最小限度。

[0129] 在本实施方式中说明的 Web 应用脆弱性对策系统具有 Web 应用执行环境，该 Web 应用执行环境具有如下功能：使用 Web 应用和设定文件的静态代码分析的结果，判定有无 Web 应用的脆弱性对策的必要性，根据其判定结果执行 Web 应用的脆弱性对策

[0130] 并且，在本实施方式中说明了具有如下库的情况：对通过静态代码分析而得到的需要脆弱性对策的库的调用位置和在执行 Web 应用时取得的被调用的库的位置信息进行比较，根据比较结果变更在内部进行的动作。

[0131] 实施方式 2

[0132] 在以上的实施方式 1 中说明了如下结构：通过静态代码分析取得在开始运行前利用外部输入数据的通用库的调用位置，根据在执行 Web 应用时在开始运行前取得的通用库的调用位置的信息、和堆栈轨迹等 Web 应用的执行信息中包含的被调用的代码位置的信息，动态地变更是否需要在通用库的内部进行的脆弱性对策。

[0133] 下面，在本实施方式中示出如下实施方式：通过开始运行前的静态代码分析，调查在转发给通用库的数据 (HTML 和 SQL 语句等）的语法上的哪个位置利用了外部输入数据，由此，提高在执行时进行的脆弱性对策的正确性。

[0134] 图 5 是本实施方式中的 Web 应用脆弱性对策系统 1000 的结构图。

[0135] 图 5 相对于实施方式 1 的结构（图 2)，对静态代码分析部 1120 追加了语法分析部 1125。

[0136] 语法分析部 1125 提供分析被外部输入数据分析部 1123 调用且 HTML 语句和 SQL 语句等的与 Web 应用不同的语言的语法的功能。

[0137] 使用图 6 说明在开发环境 1100 中的动作。

[0138] 图 6 的 S301 ～ S304 是与图 3 的 S101 ～ S104 相同的流程，因而省略说明。

[0139] 在 S305 中，语法分析部 1125 调查在转发给通用库函数的数据 (HTML 和 SQL 语句等）的语法上的哪个位置利用了外部输入数据。

[0140] 例如，在 HTML 语句的情况下，调查利用外部输入数据的部位是哪个要素的内容、哪种属性的属性值等。

[0141] 然后，动作设定文件输出部 1124 根据如下信息输出动作设定文件 1130，所述信息指在 S302 中分析出的正在调用通用库函数的代码位置和被调用的通用库函数的名称、在 S303 中分析出的利用外部输入数据的一系列的代码位置和各代码中的外部输入数据可取的值、以及在 S305 中调查出的转发给通用库函数的外部输入数据被用在语法上的哪个位置 (S306)。

[0142] 在本实施方式生成的动作设定文件 1130 中，除了实施方式 1 的记述内容以外，还记述有转发给通用库函数的外部输入数据包含在 HTML 语句或者 SQL 语句的语法上的哪个位置。

[0143] 在本实施方式的动作设定文件 1130 中，列举出多个该语法上的位置，还在语法上的每个位置记述有对策处理。

[0144] 另外，实施方式 1 的动作设定文件 1130 的记述内容是具有脆弱性的通用库函数（脆弱性库函数）的名称、调用脆弱性库函数的代码位置、外部输入数据的类型（变量的名称）、以及外部输入数据的非法值（发生异常的值）。

[0145] 与实施方式 1 同样地将生成的动作设定文件 1130 输出到动作设定文件存储部 1240，并保存在动作设定文件存储部 1240 中。

[0146] 在执行环境 1200 的动作中，图 4 的处理流程的 S208 的处理与实施方式 1 不同。

[0147] 在判定为需要调用脆弱性对策处理部 1233 的情况下 (S207 ：是），对策选择部 1232 在 S208 中判定成为无害化对象的外部输入数据被记述在来自客户终端 2000 的 HTTP 响应中的 HTML 语句或者 SQL 语句的哪个位置。

[0148] 另外，对策选择部 1232 从记述在动作设定文件 1130 中的多个语法位置中提取与 HTTP 请求中的记述位置一致的语法位置，并提取已提取出的语法位置的对策处理。

[0149] 另外，对策选择部 1232 使脆弱性对策处理部 1233 执行已提取出的对策处理，使其进行外部输入数据的无害化等。

[0150] 除此以外的流程是与图 4 相同的处理流程，因而省略说明。

[0151] 如上所述，在本实施方式中，通过在静态代码分析中调查在转发给通用库的 HTML 语句或者 SQL 语句等的语法上的哪个位置利用了外部输入数据，能够更正确地选择在通用库的内部执行的脆弱性对策的内容。

[0152] 以上，在本实施方式中说明了具有根据如下信息变更在内部进行的动作的库的情况，该信息表示通过静态代码分析而得到的来自外部的输入数据被用在转发给库的数据的语法上的哪个位置。

[0153] 实施方式 3

[0154] 在以上的实施方式 1、2 中，通过静态代码分析取得在开始运行前利用外部输入数据的通用库的调用位置等信息，根据在执行 Web 应用时在开始运行前取得的信息和堆栈轨

迹等在执行时能够取得的信息，动态地变更是否需要在通用库的内部进行的脆弱性对策。

[0155] 与此相对，在本实施方式中示出输出动态试验的试验数据或者脚本的示例。

[0156] 本实施方式的试验数据或者脚本被用于动态试验中，该动态试验用于在开始运行前的静态代码分析中，检测由于不能判定是否需要脆弱性对策的 Web 应用的设定文件的错误而造成的脆弱性。

[0157] 图 7 是本实施方式的 Web 应用脆弱性对策系统 1000 的结构图。

[0158] 图 7 相对于实施方式 2 的结构（图 5），在开发环境 1100 中追加 Web 应用的设定文件 1111 和试验数据 / 脚本 1131，在静态代码分析部 1120 中追加设定文件分析部 1126、试验数据 / 脚本输出部 1127。

[0159] 设定文件分析部 1126 读取 Web 应用的设定文件 1111，验证设定文件有无错误，并提取利用设定文件指定的值。

[0160] 试验数据 / 脚本输出部 1127 使用静态代码分析部 1120 对 Web 应用的源 / 执行代码 1110 和 Web 应用的设定文件 1111 的分析结果，输出试验数据 / 脚本 1131。

[0161] 另外，本实施方式的开发环境 1100 相当于程序分析装置的示例。

[0162] 使用图 8 说明实施方式 3 的动作。

[0163] 在开发环境 1100 中，Web 应用输入部 1121 读取 Web 应用的源 / 执行代码 1110 和 Web 应用的设定文件 1111(S401)。

[0164] 然后，通用库利用验证部 1122 调查正在调用 Web 应用的源 / 执行代码 1110 利用的通用库函数的代码位置（行数等）、被调用的通用库函数的类型 (HTML 输出、数据库访问等)，然后验证对通用库函数提供的数据和使用方式是否恰当 (S402)。

[0165] 在判定为通用库函数被恰当地利用的情况下 (S402 :是)，设定文件分析部 1126 分析 Web 应用的设定文件 1111，验证 Web 应用的设定文件 1111 的设定内容中有无不在可设定值的范围内等明显的设定错误 (S403)。

[0166] 在通用库函数未被恰当地利用的情况下 (S402 :否)、或判定为 Web 应用的设定文件 1111 中具有设定错误的情况下 (S403 :否)，输出如下消息并结束处理，该消息包含未被恰当地利用的通用库函数的类型、代码上的位置、判定为未被恰当地利用的理由、Web 应用的设定文件 1111 的设定错误的内容等 (S404)。

[0167] 在判定为通用库函数被恰当地利用且 Web 应用的设定文件 1111 中没有设定错误的情况下 (S403 :否)，进行 S405 ～ S407 的处理。

[0168] 这些处理与图 6 的 S303 ～ S306 的处理相同，因而省略说明。

[0169] 然后，试验数据 / 脚本输出部 1127 根据在 S402、S403、S405 中进行分析的结果，对于在利用通用库函数的处理中根据 Web 应用的设定文件 1111 的设定来变更通用库函数的处理内容的处理，输出用于进行动态试验的试验数据 / 脚本 1131(S408)。

[0170] 即，试验数据 / 脚本输出部 1127 使用 Web 应用的设定文件 111 的值、脆弱性库函数、在 Web 应用中利用外部输入数据的代码位置等，生成用于检查 Web 应用的设定文件 111 的值是否恰当的试验数据或者脚本。

[0171] 如上所述，在本实施方式中，根据对 Web 应用及设定文件进行静态代码分析而得到的结果，生成用于检查由于 Web 应用的设定文件的错误而造成的脆弱性的试验数据或者脚本，由此容易在开始运行前发现在静态代码分析中不能检测出的设定文件的错误。

[0172] 以上在本实施方式中说明了具有静态代码分析部，在仅利用通过依赖于 Web 应用的设定文件的静态代码分析而得到的信息不能决定在库内部进行的动作的情况下，输出用于动态地测试设定文件有无错误的试验数据或脚本。

[0173] 最后，说明实施方式 1 ～ 3 所示的 Web 应用脆弱性对策系统 1000 的硬件结构例。

[0174] 图 9 是表示实施方式 1 ～ 3 所示的 Web 应用脆弱性对策系统 1000 的硬件资源的一例的图。

[0175] 另外，图 9 的结构毕竟只是表示 Web 应用脆弱性对策系统 1000 的硬件结构的一例，Web 应用脆弱性对策系统 1000 的硬件结构不限于图 9 所示的结构，也可以是其它结构。

[0176] 在图 9 中，Web 应用脆弱性对策系统 1000 具有执行程序的 CPU 911。

[0177] CPU 911 通过总线 912 与例如 ROM(Read Only Memory)913、RAM(Random Access Memory)914、通信板 915、显示装置 901、键盘 902、鼠标 903、磁盘装置 920 连接，并控制这些硬件装置。

[0178] 另外，CPU 911 也可以与 FDD904(Flexible Disk Drive)、CD 盘装置 905(CDD)、打印装置 906、扫描装置 907 连接。并且，也可以用 SSD(Solid State Drive)、光盘装置、存储卡（注册商标）读写装置等存储装置来取代磁盘装置 920。

[0179] RAM 914 是易失性存储器的一例。ROM 913、FDD 904、CDD 905、磁盘装置 920 的存储介质是非易失性存储器的一例。这些装置是存储装置的一例。

[0180] 在实施方式 1 ～ 3 中说明的“动作设定文件存储部 1240”利用 RAM 914、磁盘装置 920 等实现。

[0181] 通信板 915、键盘 902、鼠标 903、扫描装置 907 等是输入装置的一例。

[0182] 另外，通信板 915、显示装置 901、打印装置 906 等是输出装置的一例。

[0183] 通信板 915 如图 1 所示与网络连接。

[0184] 例如，通信板 915 与 LAN（局域网）、因特网、WAN（广域网）、SAN（存储区域网）等连接。

[0185] 在磁盘装置 920 中存储有操作系统 921(OS)、Windows 系统 922、程序组 923、文件组 924。

[0186] CPU 911 使用操作系统 921、Windows 系统 922 来执行程序组 923 的程序。

[0187] 另外，在 RAM 914 中临时存储使 CPU 911 执行的操作系统 921 的程序和应用程序的至少一部分。

[0188] 另外，在 RAM 914 中存储有 CPU 911 进行处理所需要的各种数据。

[0189] 另外，在 ROM 913 中存储有 BIOS(Basic Input Output System) 程序，在磁盘装置 920 中存储有根程序。

[0190] 在 Web 应用脆弱性对策系统 1000 起动时执行 ROM 913 的 BIOS 程序和磁盘装置 920 的根程序，利用 BIOS 程序和根程序起动操作系统 921。

[0191] 在上述程序组 923 中存储有执行在实施方式 1 ～ 3 的说明中作为“～部”（动作设定文件存储部 1240 除外，以下相同）而说明的功能的程序、“Web 应用 1220”、“Web 服务器程序 1210”。

[0192] 程序被 CPU 911 读出并执行。

[0193] 在文件组 924 中，表示在实施方式 1 ～ 3 的说明中作为“～的判断”、“～的判定”、

“～的提取”、“～的比较”、“～的验证”、“～的生成”、“～的设定”、“～的取得”、“～的核对”、“～的选择”、“～的生成”、“～的输入”、“～的输出”等而说明的处理结果的信息、数据、信号值、变量值，作为文件被存储在盘或存储器等存储介质中。

[0194] 另外，也可以将加密密钥／解码密钥、随机值、参数作为文件存储在盘或存储器等存储介质中。

[0195] “～文件”或“～数据库”被存储在盘或存储器等存储介质中。

[0196] 被存储在盘或存储器等存储介质中的信息、数据、信号值、变量值、参数，通过读写电路被 CPU 911 读出到主存储器或高速缓冲存储器中。

[0197] 另外，被读出的信息、数据、信号值、变量值、参数被用于提取、检索、参照、比较、运算、计算、处理、编辑、输出、打印、显示等 CPU 的动作中。

[0198] 在提取、检索、参照、比较、运算、计算、处理、编辑、输出、打印、显示的 CPU 的动作期间，信息、数据、信号值、变量值、参数被临时存储在主存储器、寄存器、高速缓冲存储器、缓冲存储器等中。

[0199] 另外，在实施方式 1 ～ 3 中说明的流程图中的箭头部分主要表示数据和信号的输入输出。

[0200] 数据和信号值被记录在 RAM 914 的存储器、FDD 904 的软盘、CDD 905 的 CD 盘、磁盘装置 920 的磁盘、其它光盘、蓝光（注册商标）光盘、DVD 等存储介质中。

[0201] 另外，数据和信号通过总线 912、信号线、线缆等其它传输介质被在线传输。

[0202] 另外，在实施方式 1 ～ 3 的说明中作为“～部”而说明的部分也可以是“～电路”、“～装置”、“～设备”，还可以是“～步骤”、“～工序”、“～处理”。

[0203] 另外，也能够将 Web 应用脆弱性对策系统 1000 的处理作为程序执行方法。

[0204] 另外，作为“～部”而说明的部分也可以利用存储在 ROM 903 中的固件来实现。

[0205] 或者仅利用软件实现，或者仅利用元件／装置／基板／配线等硬件实现，或者利用软件和硬件的组合来实现，或者利用软件、硬件和固件的组合来实现。

[0206] 固件和软件被作为程序存储在磁盘、软盘、光盘、CD 盘、蓝光（注册商标）光盘、DVD 等存储介质中。

[0207] 程序被 CPU 911 读出，并由 CPU 911 执行。

[0208] 即，程序是使计算机作为实施方式 1 ～ 3 的“～部”发挥作用的。或者，程序是使计算机执行实施方式 1 ～ 3 的“～部”的步骤或方法。

[0209] 这样，实施方式 1 ～ 3 所示的 Web 应用脆弱性对策系统 1000 是具有作为处理装置的 CPU、作为存储装置的存储器和磁盘等、作为输入装置的键盘和鼠标和通信板等、作为输出装置的显示装置、通信板等的计算机。

[0210] 另外，使用这些处理装置、存储装置、输入装置、输出装置实现如上所述作为“～部”而示出的功能。

[0211] 标号说明

[0212] 1000Web 应用脆弱性对策系统；1100 开发环境；1110Web 应用的源／执行代码；1111Web 应用的设定文件；1120 静态代码分析部；1121Web 应用输入部；1122 通用库利用验证部；1123 外部输入数据分析部；1124 动作设定文件输出部；1125 语法分析部；1126 设定文件分析部；1127 试验数据／脚本输出部；1130 动作设定文件；1131 试验数据／脚本；1200

执行环境 ;1210Web 服务器程序 ;1220Web 应用 ;1230 通用库 ;1231 通用库接口 ;1232 对策选择部 ;1233 脆弱性对策处理部 ;1234 通用库处理部 ;1240 动作设定文件存储部 ;2000 客户终端 ;3000 网络。

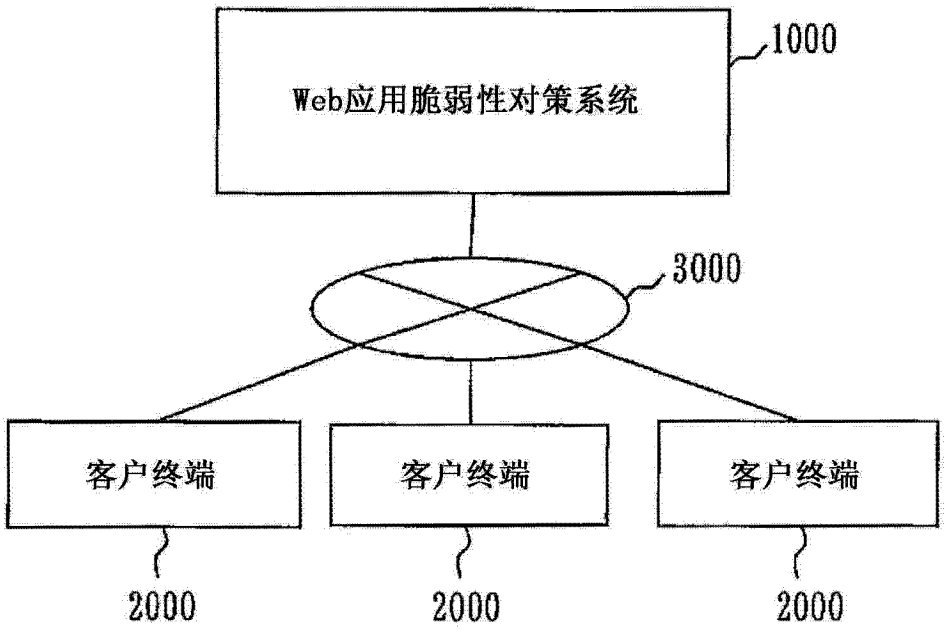


图 1

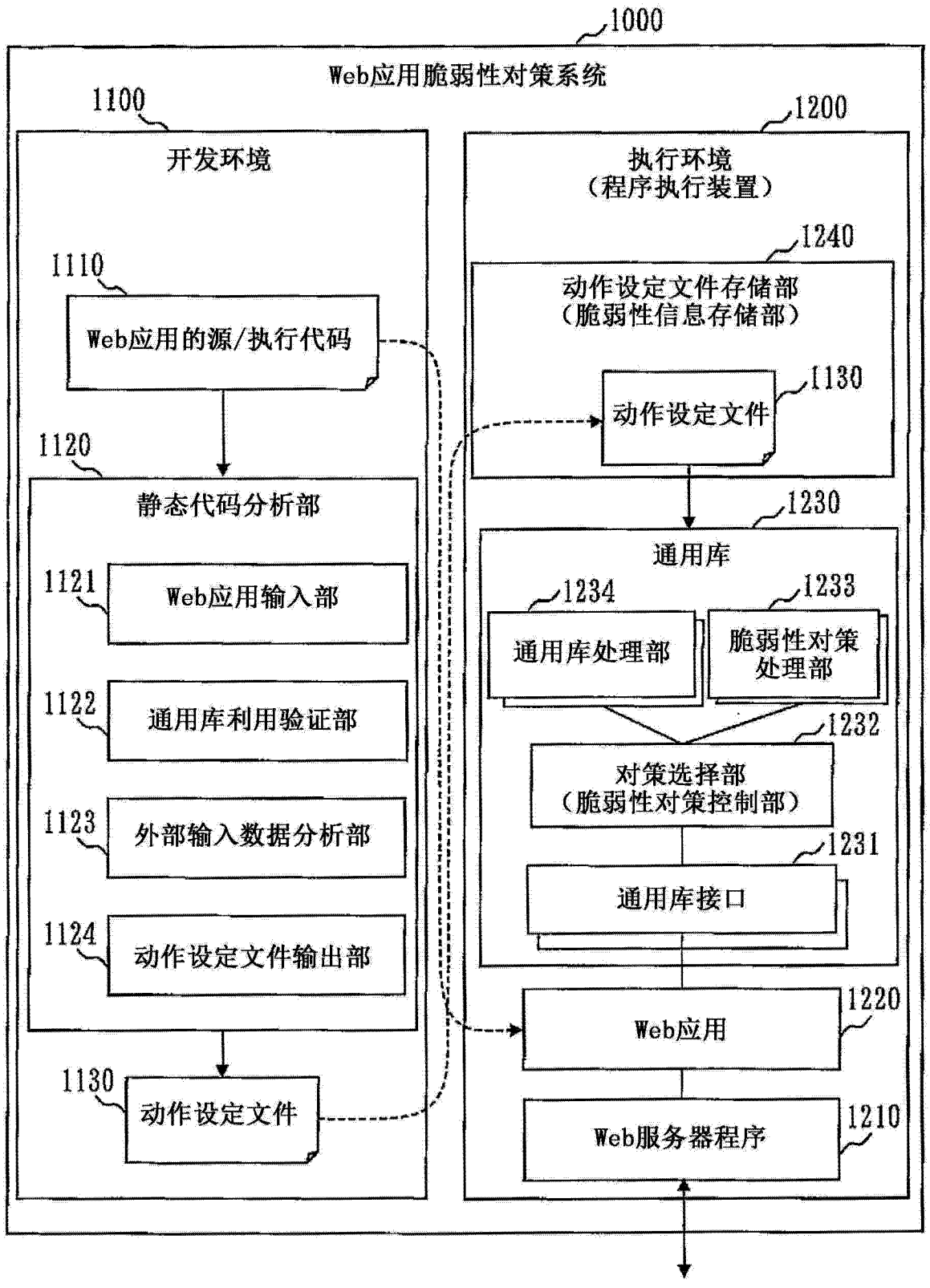

1000
Web应用脆弱性对策系统
1100
开发环境
1110
Web应用的源/执行代码
1120
静态代码分析部
1121
Web应用输入部
1122
通用库利用验证部
1123
外部输入数据分析部
1124
动作设定文件输出部
1130
动作设定文件
1200
执行环境
（程序执行装置）
1240
动作设定文件存储部
（脆弱性信息存储部）
1130
动作设定文件
1230
通用库
1234
通用库处理部
1233
脆弱性对策
处理部
1232
对策选择部
（脆弱性对策控制部）
1231
通用库接口
1220
Web应用
1210
Web服务器程序

图 2

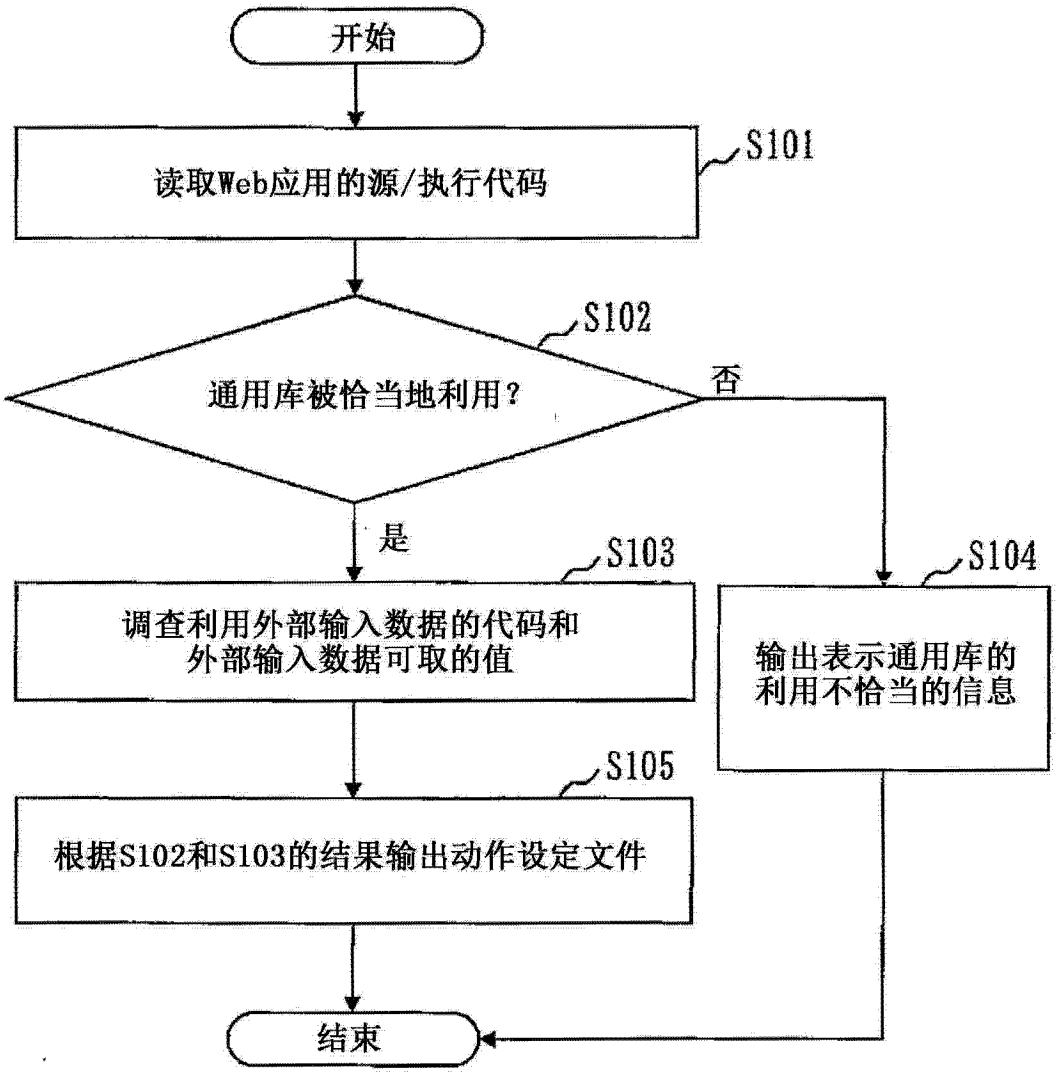

开始
读取Web应用的源/执行代码
S101
S102
通用库被恰当地利用？
否
是
S103
调查利用外部输入数据的代码和
外部输入数据可取的值
S104
输出表示通用库的
利用不恰当的信息
S105
根据S102和S103的结果输出动作设定文件
结束

图 3

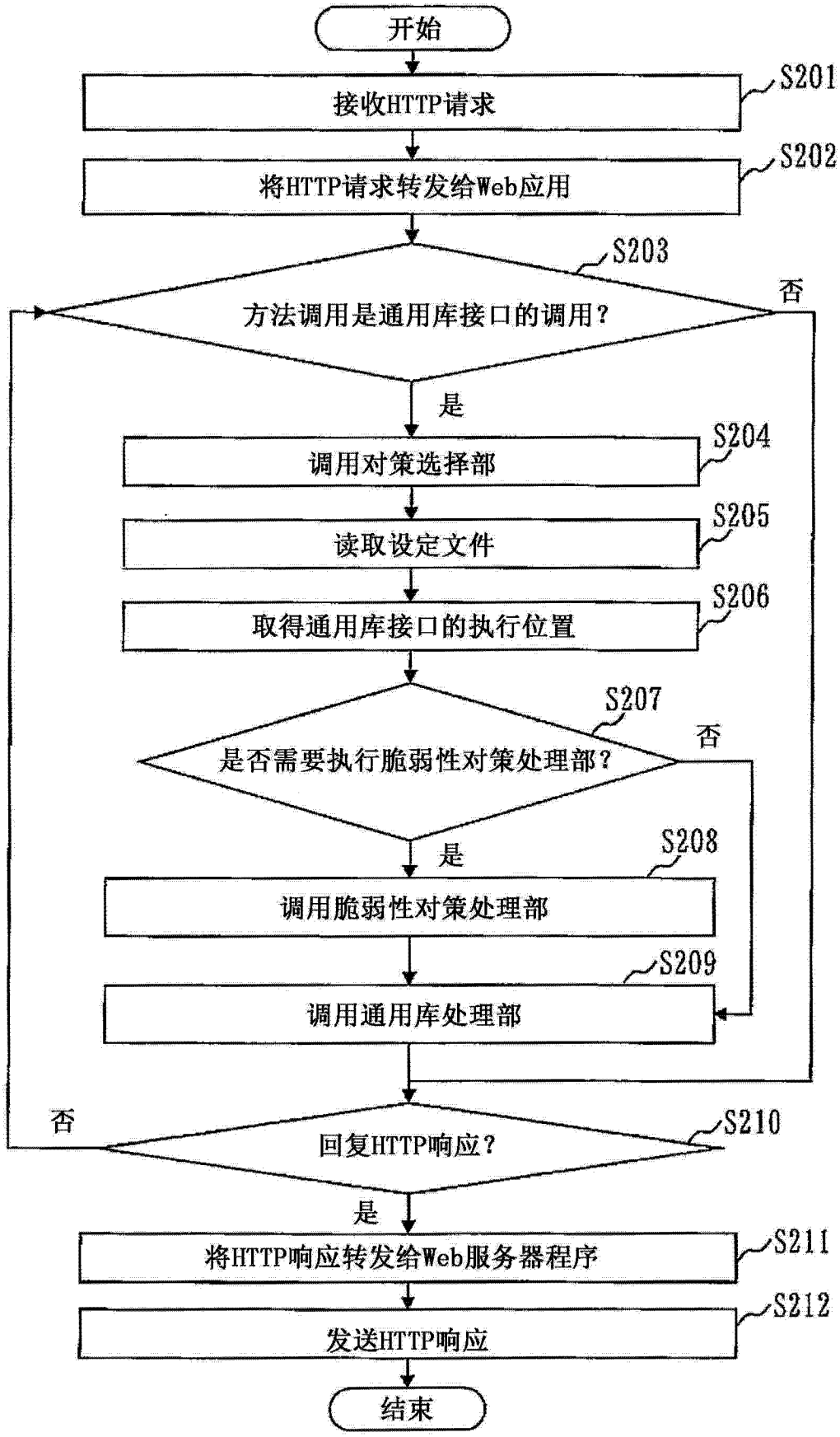

开始
接收HTTP请求
S201
将HTTP请求转发给Web应用
S202
方法调用是通用库接口的调用？
S203
否
是
调用对策选择部
S204
读取设定文件
S205
取得通用库接口的执行位置
S206
是否需要执行脆弱性对策处理部？
S207
否
是
调用脆弱性对策处理部
S208
调用通用库处理部
S209
回复HTTP响应？
S210
否
是
将HTTP响应转发给Web服务器程序
S211
发送HTTP响应
S212
结束

图 4

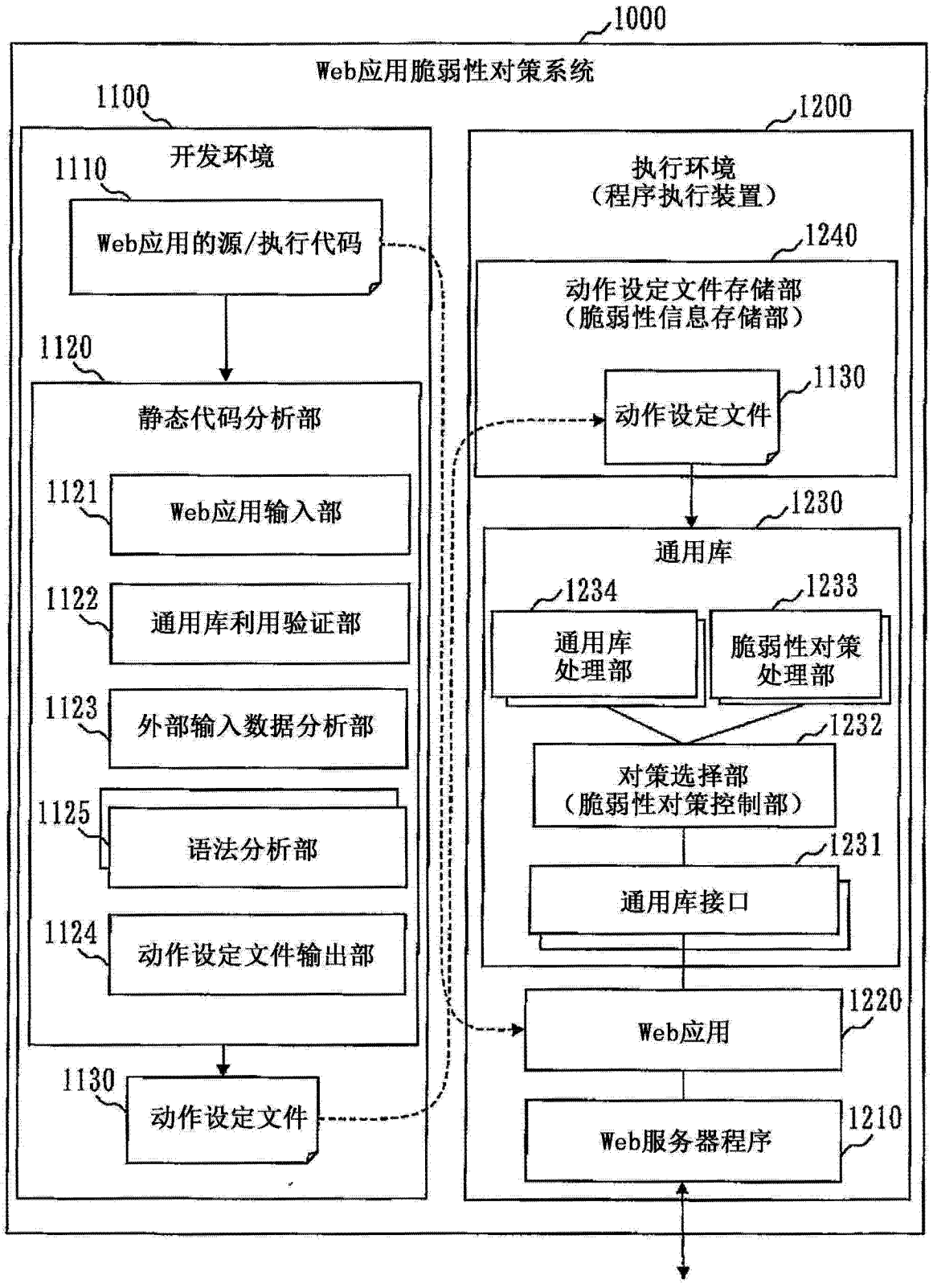

1000
Web应用脆弱性对策系统
1100
开发环境
1110
Web应用的源/执行代码
1120
静态代码分析部
1121
Web应用输入部
1122
通用库利用验证部
1123
外部输入数据分析部
1125
语法分析部
1124
动作设定文件输出部
1130
动作设定文件
1200
执行环境
（程序执行装置）
1240
动作设定文件存储部
（脆弱性信息存储部）
1130
动作设定文件
1230
通用库
1234
通用库
处理部
1233
脆弱性对策
处理部
1232
对策选择部
（脆弱性对策控制部）
1231
通用库接口
1220
Web应用
1210
Web服务器程序

图 5

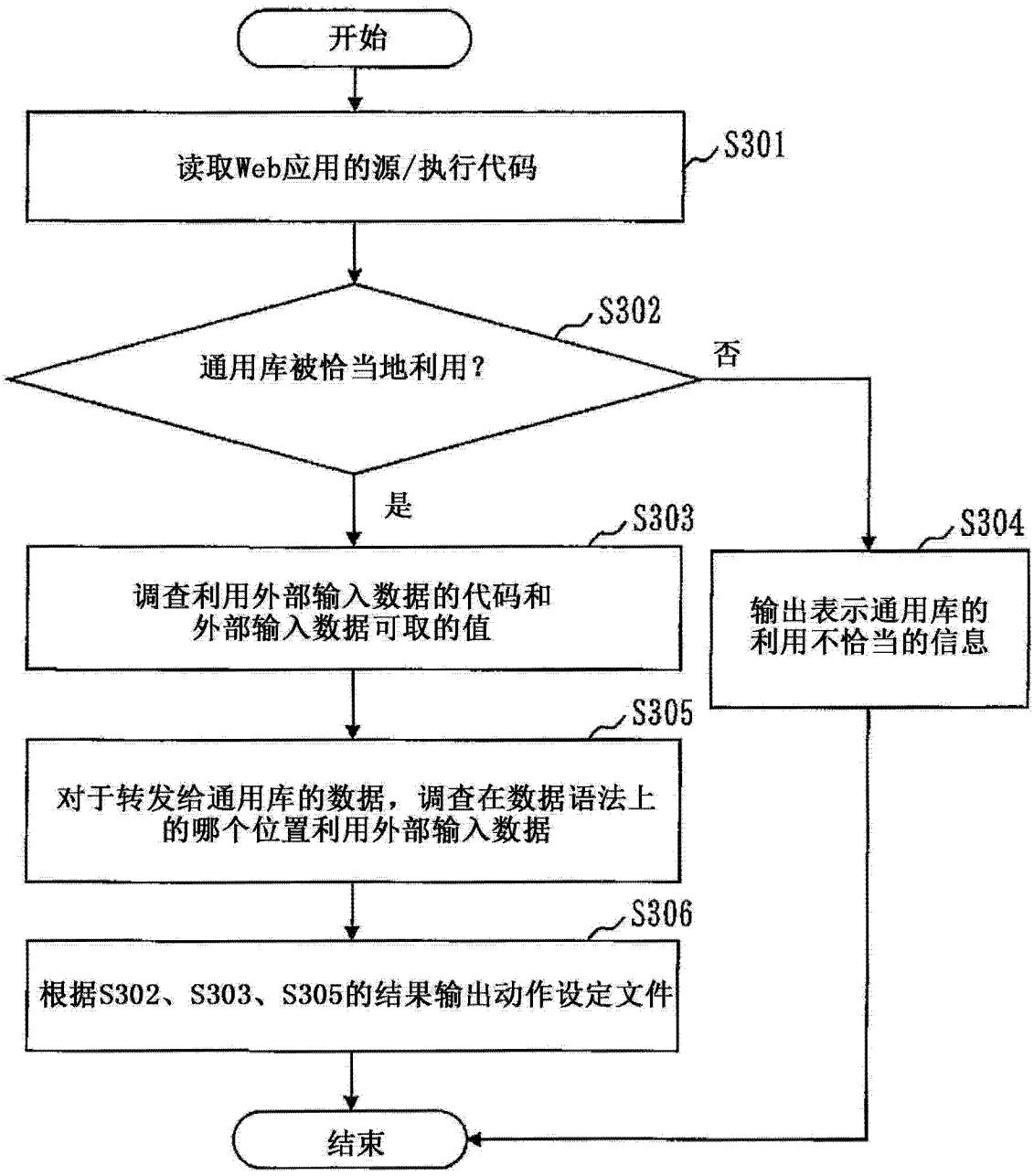

开始
读取Web应用的源/执行代码
S301
通用库被恰当地利用？
S302
否
是
S303
调查利用外部输入数据的代码和
外部输入数据可取的值
S304
输出表示通用库的
利用不恰当的信息
S305
对于转发给通用库的数据，调查在数据语法上
的哪个位置利用外部输入数据
S306
根据S302、S303、S305的结果输出动作设定文件
结束

图 6

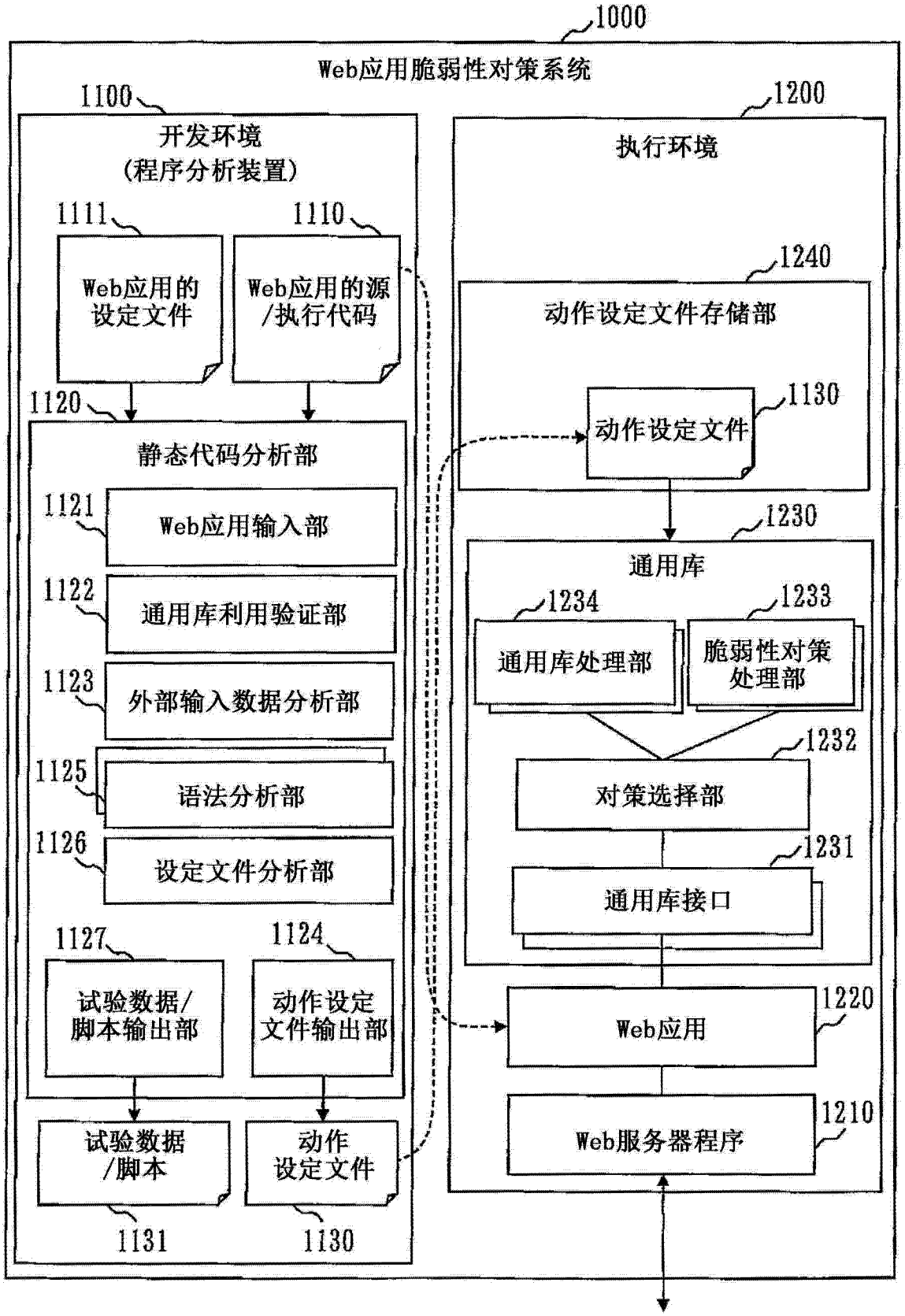

1000
Web应用脆弱性对策系统
1100
开发环境
(程序分析装置)
1111
Web应用的
设定文件
1110
Web应用的源
/执行代码
1120
静态代码分析部
1121
Web应用输入部
1122
通用库利用验证部
1123
外部输入数据分析部
1125
语法分析部
1126
设定文件分析部
1127
试验数据/
脚本输出部
1124
动作设定
文件输出部
试验数据
/脚本
动作
设定文件
1131
1130
1200
执行环境
1240
动作设定文件存储部
1130
动作设定文件
1230
通用库
1234
通用库处理部
1233
脆弱性对策
处理部
1232
对策选择部
1231
通用库接口
1220
Web应用
1210
Web服务器程序

图 7

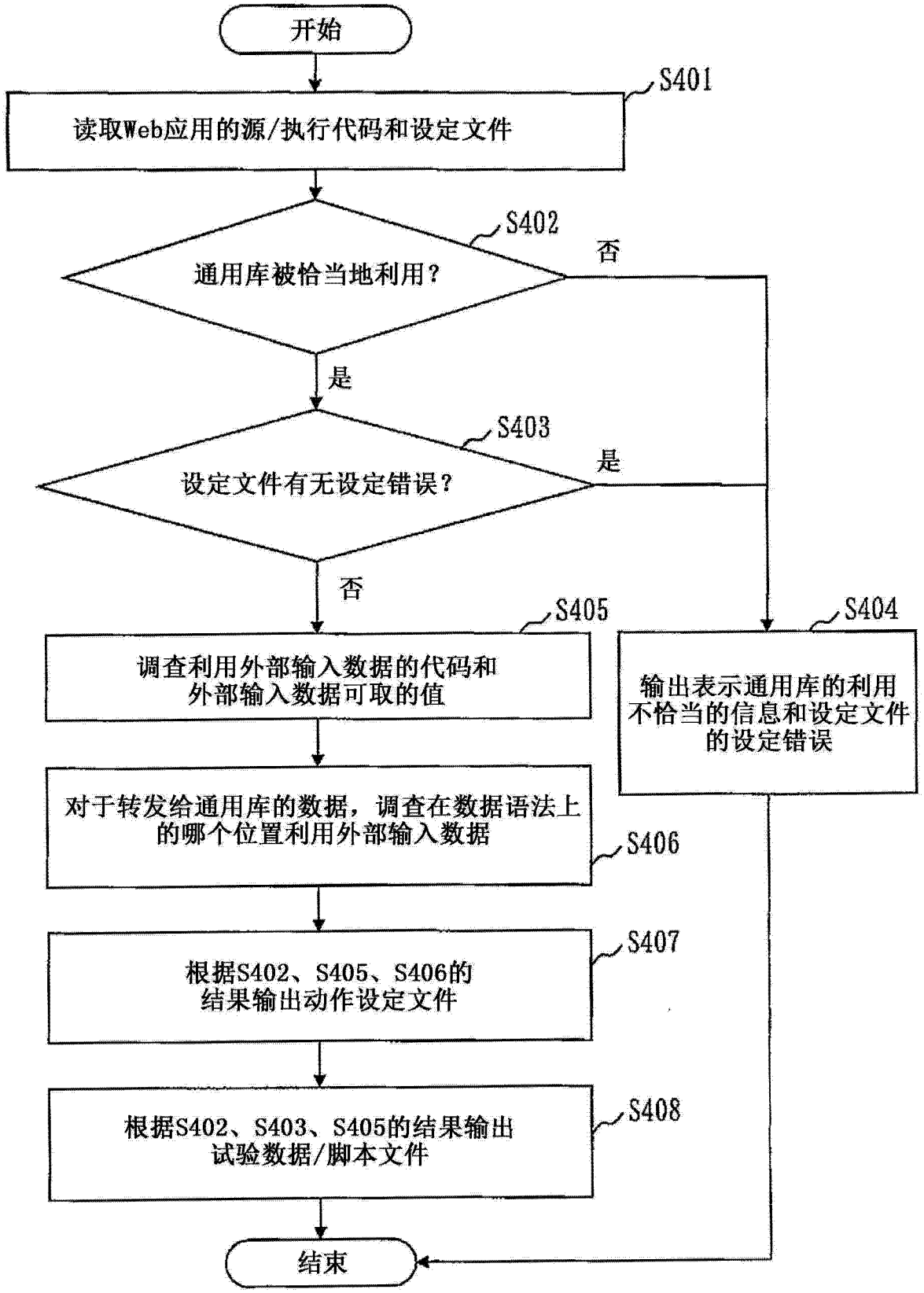

开始
S401
读取Web应用的源/执行代码和设定文件
S402
通用库被恰当地利用？
否
是
S403
设定文件有无设定错误？
是
否
S405
调查利用外部输入数据的代码和
外部输入数据可取的值
S404
输出表示通用库的利用
不恰当的信息和设定文件
的设定错误
对于转发给通用库的数据，调查在数据语法上
的哪个位置利用外部输入数据
S406
S407
根据S402、S405、S406的
结果输出动作设定文件
S408
根据S402、S403、S405的结果输出
试验数据/脚本文件
结束

图 8

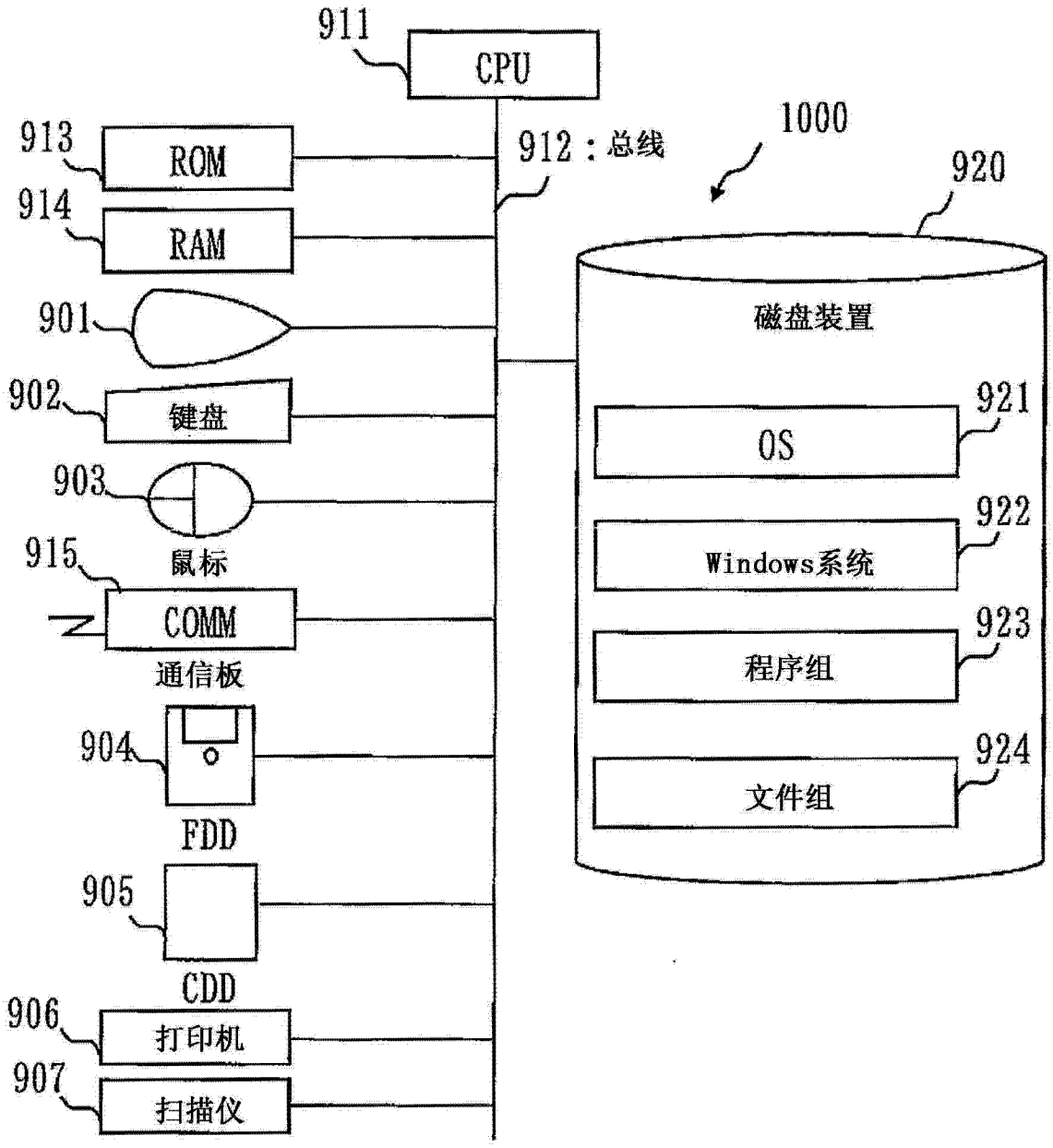

911
CPU
1000
913
ROM
912：总线
920
914
RAM
磁盘装置
901
902
键盘
921
OS
903
922
鼠标
Windows系统
915
COMM
923
通信板
程序组
904
924
文件组
FDD
905
CDD
906
打印机
907
扫描仪

图 9