(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101964025 B
(45)授权公告日 2016.02.03

(21)申请号 200910089786.8
(22)申请日 2009.07.23
(73)专利权人 北京神州绿盟信息安全科技股份有限公司
地址 100089 北京市海淀区北洼路4号益泰大厦三层
(72)发明人 刘光旭 温玉杰 周大 王晓明 刘晓霞
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 谢建云 刘红
(51)Int.Cl.
G06F 21/57(2013.01)
H04L 29/06(2006.01)

(56)对比文件
CN 101459548 A,2009.06.17,
US 7343626 B1,2008.03.11,
CN 101471781 A,2009.07.01,

审查员 王晓飞

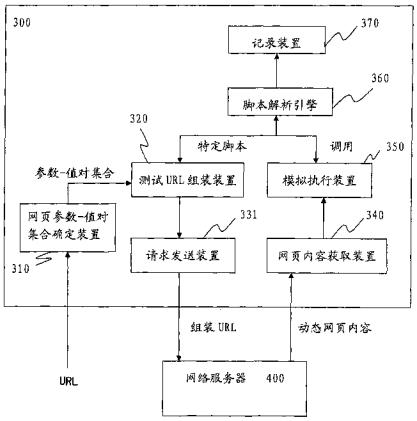
权利要求书1页 说明书6页 附图3页

(54)发明名称

XSS检测方法和设备

(57)摘要

本发明公开了一种检测网页中的XSS漏洞的XSS漏洞检测方法,包括为网页可接收的参数-值对集合中的每个参数-值对:构造在值中插入了特定脚本的参数-值对,基于该插入了特定脚本的参数-值对来组装对应于所述网页的URL,获取对应于所组装的URL的动态网页内容,以及模拟执行所获取的动态网页内容,如果执行了所述特定脚本,则认为所述网页中对该参数的处理存在XSS漏洞。本发明还公开了一种相应的XSS漏洞检测设备和使用该设备的网站安全扫描系统和网络扫描系统。



CN 101964025 B

1. 一种检测网页中的 XSS 漏洞的 XSS 漏洞检测方法，包括步骤：

确定网页可接收的参数 - 值对集合；以及

为所述参数 - 值对集合中的每个参数 - 值对：

构造在值中插入了特定脚本的参数 - 值对；

基于该插入了特定脚本的参数 - 值对来组装对应于所述网页的 URL；

将所组装的 URL 发送到网络服务器；

接收从所述网络服务器返回的对应于所组装的 URL 的动态网页内容；以及

使用脚本解析引擎来模拟执行所获取的动态网页内容，如果执行了所述特定脚本，则认为所述网页中对该参数的处理存在 XSS 漏洞，其中，该脚本解析引擎被构造为基于该特定脚本是否被触发来确定是否存在 XSS 漏洞，并对其它脚本的执行进行简化处理。

2. 如权利要求 1 所述的 XSS 漏洞检测方法，其中所述特定脚本为 alert 函数。

3. 如权利要求 1 所述的 XSS 漏洞检测方法，其中在所述组装对应于网页的 URL 的步骤中，通过在 URL 中改变参数 - 值对次序和插入其它特殊代码来组装多个 URL，并且为所述多个 URL 分别执行所述获取动态网页内容和模拟执行动态网页内容的步骤。

4. 如权利要求 1-3 中任一所述的 XSS 漏洞检测方法，还包括步骤：

记录所述参数 - 值对集合中的每个参数是否存在 XSS 漏洞。

5. 一种检测网页中的 XSS 漏洞的 XSS 漏洞检测设备，包括：

网页参数 - 值对集合确定装置，用于确定所述网页可以接收的参数 - 值对集合；

测试 URL 组装装置，为所述参数 - 值对集合中的每个参数 - 值对组装测试用的 URL，其中在组织所述测试用的 URL 时，在所述值中插入特定的脚本；

请求和接收装置，用于将所述测试用的 URL 发送到网络服务器，并接收从所述网络服务器返回的动态网页内容；以及

脚本解析引擎，模拟执行所获取的动态网页内容，如果执行了所述特定脚本，则认为所述网页中对该参数的处理存在 XSS 漏洞，其中，该脚本解析引擎被构造为基于该特定脚本是否被触发来确定是否存在 XSS 漏洞，并对其它脚本的执行进行简化处理。

6. 如权利要求 5 所述的 XSS 漏洞检测设备，其中所述特定脚本为 alert 函数。

7. 如权利要求 5 所述的 XSS 漏洞检测设备，其中所述测试 URL 组装装置在为某个参数 - 值对组装测试用的 URL 时，通过在 URL 中改变参数 - 值对次序和插入其它特殊代码来组装多个 URL，并且将每个组装的 URL 发送给所述请求和接收装置，以便为每个组装的 URL 进行 XSS 漏洞检测。

8. 如权利要求 5-7 中任一所述的 XSS 漏洞检测设备，还包括记录装置，记录所述参数 - 值对集合中的每个参数是否存在 XSS 漏洞。

9. 一种网站安全扫描系统，包括如权利要求 5-7 中的任一项所述的 XSS 漏洞检测设备。

10. 一种网络扫描系统，包括如权利要求 5-7 中的任一项所述的 XSS 漏洞检测设备。

XSS 检测方法和设备

技术领域

[0001] 本发明涉及网站安全扫描和分析领域，尤其涉及一种用于对网站中的网页是否具有 XSS（跨站脚本攻击）漏洞进行检测的方法和设备。

背景技术

[0002] 从二十世纪九十年代开始，XSS 漏洞开始被披露，其发现和利用为人们所关注。XSS，即跨站脚本攻击，是利用网站漏洞从用户那里恶意盗取信息的方式之一。用户在浏览网站、使用即时通讯软件、或者在阅读电子邮件时，通常会点击其中的链接。恶意攻击者在链接中插入恶意代码，当用户点击这些链接时，生成相应网页的网络服务器由于没有过滤这些恶意代码而具有 XSS 漏洞，因此生成包含恶意代码的页面，而这个页面看起来就像是那个网站应当生成的合法页面一样，从而导致这些恶意代码最终在用户计算机上执行，绕过用户本地的安全机制来盗取用户信息，甚至在用户机器上进行挂马攻击而远程获得用户机器的控制权等。攻击者通常会用十六进制（或其他编码方式）将链接编码，以免用户怀疑它的合法性。XSS 在目前互联网站点上普遍存在，给直接用户带来极大的威胁。近年来，XSS 一举超过缓冲区溢出而成为最流行的安全漏洞之一。大约至少有 68％的网站存在 XSS 漏洞。

[0003] 检测 XSS 从检测的途径来看可分为远程主动检测和本地被动检测两种方式。本地被动检测技术主要应用在浏览器里，目前 IE8，firefox 的 noscript 插件都支持 XSS 检测。远程主动检测则主要应用在远程漏洞扫描器类的检测类工具里。本发明主要是针对远程检测技术进行改进。

[0004] 随着网站程序员对安全的认识也有所增强，会对用户输入的参数进行一些特别处理，这部分特别处理给远程扫描 XSS 漏洞带来了一定的难度，尤其是更容易给远程扫描带来误报。

[0005] 目前已经提出了几种用于远程扫描网络服务器的 XSS 漏洞的方法。美国专利 US 7343626B1 公开了一种测试网站是否具有 XSS 漏洞的自动化方法和系统，其中包括 ：对于网络服务器的网页，找出其所有参数 - 值对 ；对于每个参数 - 值对，构造特定的跟踪值，并且将构造的参数 - 值对提交到网络服务器以请求网页 ；如果返回的网页中包含特定的跟踪值，则说明该网页可能具有 XSS 漏洞 ；此时基于网页中特定跟踪值出现的位置，构造包括脚本的第二特定跟踪值，并再次提交到网络服务器，并根据返回的网页是否执行该脚本来判断网页是否具有 XSS 漏洞。然而，美国专利 US 7343626B1 所公开的方法需要两次提交参数 - 值对，因此执行效率不高。此外该方法还需要基于特定跟踪值出现的位置来构造包括脚本的第二特定跟踪值，由于随着网络技术的发展，XSS 漏洞可能以其他位置出现，这也会导致该方法不能完整地检测出 XSS 漏洞。

[0006] 一些开源软件中也公开了其它用于远程检测 XSS 漏洞的方法，其大致原理如下，对于某个要检测的网页，首先确定该网页接受的参数 - 值对，然后对于每个参数，构造特定的值，并且利用这些特定构造的参数 - 值对向网络服务器请求该网页，最后根据返回信息

来判断漏洞是否存在。对返回信息的分析方法，这些方法采用的是基于正则表达式的匹配。在这些开源软件的检测方法中，通过基于特征串的正则表达式匹配来分析返回信息以判断是否具有 XSS 漏洞，这在某些情况下会产生一些误报或者漏报。此外，在基于 DOM 的 XSS 漏洞中，该方法无法判断漏洞能否被触发。例如，返回的网页虽然包含了构造的特定值，但是该特定值不会被执行时，该方法仍然认为网页存在 XSS 漏洞，但是实际上并非如此。

[0007] 可以看出，在本技术领域，还没有一种方法和设备可以完全且高效地检测 XSS 漏洞，本发明力图通过对开源软件中所提出的方法进行改进来提供一种完全自动化的、可以全面且高效地检测 XSS 漏洞的方案。

发明内容

[0008] 本发明的申请人发现，XSS 漏洞的最终结果是要在被攻击者的机器上执行非预期的脚本代码，因此，如果利用 javascript 解析引擎来确定非预期的脚步代码是否在被攻击者的机器上执行，则可以非常全面地检测 XSS 漏洞。本发明基于此做出。

[0009] 根据本发明的一个方面，提供了一种检测网页中的 XSS 漏洞的 XSS 漏洞检测方法，包括步骤：确定网页可接收的参数 - 值对集合；以及为所述参数 - 值对集合中的每个参数 - 值对：构造在值中插入了特定脚本的参数 - 值对；基于该插入了特定脚本的参数 - 值对来组装对应于所述网页的 URL；将所组装的 URL 发送到网络服务器；接收从所述网络服务器返回的对应于所组装的 URL 的动态网页内容；以及模拟执行所获取的动态网页内容，如果执行了所述特定脚本，则认为所述网页中对该参数的处理存在 XSS 漏洞。在所述模拟执行所获取的动态网页内容的步骤中，使用脚本解析引擎来模拟执行所述网页内容中，该脚本解析引擎被构造为基于特定脚本是否被触发来确定是否存在 XSS 漏洞，并对其它脚本的执行进行简化处理。

[0010] 根据本发明的另一方面，提供了一种检测网页中的 XSS 漏洞的 XSS 漏洞检测设备，包括：网页参数 - 值对集合确定装置，用于确定所述网页可以接收的参数 - 值对集合；测试 URL 组装装置，为所述参数 - 值对集合中的每个参数 - 值对组装测试用的 URL，其中在组织所述测试用的 URL 时，在所述值中插入特定的脚本；请求和接收装置，用于将所述测试用的 URL 发送到网络服务器，并接收从所述网络服务器返回的网页内容；以及模拟执行装置，用于模拟执行所述网页内容，并且在所述特定脚本被执行时，确定相应参数存在 XSS 漏洞。该 XSS 漏洞检测设备还包括脚本解析引擎，所述模拟执行装置在模拟执行所述网页内容时使用所述脚本解析引擎来执行脚本，所述脚本解析引擎根据所述特定脚本是否被触发来确定所述相应参数是否存在 XSS 漏洞，并对其它脚本的执行进行简化处理。

[0011] 本发明通过组装具有特定脚本的参数值的 URL，并通过模拟执行该 URL 返回的动态网页内容是否会触发该特定脚本来确定该动态网页是否具有 XSS 漏洞，并对其它脚本的执行进行简化处理。和传统的检测方法相比，其更加高效。

附图说明

[0012] 通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。其中在附图中，参考数字

之后的字母标记指示多个相同的部件，当泛指这些部件时，将省略其最后的字母标记。在附图中：

[0013] 图 1 示出了根据本发明实施例的、用于检测网页的 XSS 漏洞的方法的流程图；

[0014] 图 2 示出了根据本发明实施例的、用于确定网络服务器对某个参数的处理是否存在 XSS 漏洞的方法的流程图；以及

[0015] 图 3 示出了根据本发明实施例的、用于检测网页的 XSS 漏洞的设备的示意图。

具体实施例

[0016] 下面结合附图和具体的实施方式对本发明作进一步的描述。

[0017] 图 1 示出了根据本发明实施例的、用于检测网页 XSS 漏洞的方法的流程图。

[0018] 在步骤 S110 中，获取要进行 XSS 漏洞检测的网页的 URL，然后在步骤 S120 中，确定该网页能够接收的参数 - 值对集合。根据 XSS 的原理，具有 XSS 漏洞的网页为网络服务器动态生成的网页，因此其通常可以根据 HTTP 协议接收一些参数和相应的值。例如如果要检测的网页 URL 为 http://www.test.com/test.asp？ id ＝ 1&name ＝ test，则可以确定该网页可以接收的参数包括 id 和 name。可以通过各种方式来确定动态 URL 可以接收的参数集，如通过监视访问该动态网页的详细 URL 内容，根据 HTTP 协议检测传递到网络服务器的 FORM 表单的内容等。所以这些都在本发明的保护范围之内。

[0019] 在步骤 S120 获取了该参数 - 值对集合之后，在步骤 S130，选取该集合中的第一组参数 - 值对，然后在步骤 S140 中，对所选择的参数 - 值对中的值进行修改，以组装测试用的 URL。具体而言，根据本发明的实施例，在该值中嵌入特定的 javascript 脚本，如果该动态网页存在 XSS 漏洞，则该嵌入的 javascript 脚本将未被网络服务器所处理，并存在于返回的动态网页内容中。本发明通过监控该 javascript 脚本是否会存在于动态网页中并被执行来确定该网页是否存在 XSS 漏洞。为了防止所嵌入的 javascript 脚本和本来动态网页中就有的 javascript 脚本相冲突，因此所嵌入的 javascript 脚本应当是唯一的，例如其中包含了唯一的脚本参数内容。另外，由于 javascript 脚本中的 alert 函数的作用是弹出消息框而不会对文档内容产生其它影响，因此，在本发明的进一步实施例中，将 alert 函数嵌入到所选择的参数 - 值对中。

[0020] 例如，在上面给出的 URL：

[0021] http://www.test.com/test.asp？ id ＝ 1&name ＝ test 中，

[0022] 在对参数 id 进行修改时，可以在其对应的值中嵌入函数 <script>alert(0)</script>。因此，所构造得到的 URL 变为：

[0023] http://www.test.com/test.asp？ id ＝ 1<script>alert(0)</script>&name ＝ test。

[0024] 可选的是，根据 HTTP 协议，提交给网络服务器的参数并没有先后次序，因此，可以修改参数的前后次序以把嵌入值放到最后，即：

[0025] http://www.test.com/test.asp？ name ＝ test&id ＝ 1<script>alert(0)</script>。

[0026] 另外，在修改参数值时，除了添加 javascript 脚本之外，还可以添加一些特殊字符，如“＞”、“＜”和“％ 20”等，因此，构造出的 URL 还可以是：

[0027] http://www.test.com/test.asp ? name = test&id = 1 % 20<script>alert(0)</script> 或者

[0028] http://www.test.com/test.asp ? name = test&id = 1 % 20 > <script>alert(0)</script>等。

[0029] 此外，在修改参数值时，为了确保对该参数的处理存在 XSS 漏洞时，所插入的脚本能被 javascript 解析引擎所执行，还需考虑 html 语法的恢复问题，例如，如果动态网页生成的 html 代码为

[0030] <pre> 你输入的名字是 test</pre>

[0031] 其中 test 是由动态网页根据参数 id 的值而产生的。在这种情况下，如果仅仅在参数 id 的值中添加 javascript 脚本，即 id = <script>alert(0)</script>，则此时由动态网页生成的 javascript 脚本包含在 <pre></pre> 的 html 标记符中，并不能由 javascript 解析引擎来执行，为此，我们需要将参数 id 值中插入的脚本修改为 id = </pre><script>alert(0)</script><pre>，这样动态网页生成的 html 代码变为：

[0032] <pre> 你输入的名字是 </pre><script>alert(0)</script><pre>

[0033] 从而确保插入的脚本（如 alert 函数）能被解析引擎执行。

[0034] 上面的 <pre> 标签只是一个例子，还存在其他可能会导致插入的脚本的情况，因此，在参数值中插入 javascript 脚本时，还必须考虑 html 语法的恢复问题，以确保在对参数的处理存在 XSS 漏洞时，所插入的 javascript 脚本会被执行。因此，需要在插入 javascript 脚本时进行多种方式的插入，从而更精确地确定 XSS 漏洞。

[0035] 在步骤 S140 组装了测试用的 URL 之后，在步骤 S150 将组装好的 URL 发送到网络服务器以请求动态网页内容，并且在步骤 S160 获取作为响应的动态网页内容。随后，在步骤 S170 中，利用根据本发明的 javascript 解析引擎来模拟执行动态网页内容，并根据在步骤 S140 中插入的特定脚本是否有 javascript 解析引擎执行来确定该参数是否具有 XSS 漏洞。下面将参考图 2 详细描述步骤 S170 中的处理，这里不再进行赘述。

[0036] 在步骤 S170 对该参数是否存在 XSS 漏洞进行了判断之后，在步骤 S180 确定参数集中是否还有需要进行判断的参数，如果有，则在步骤 S190 获取参数集中下一个要处理的参数 - 值对，并且返回到步骤 S140 来处理该参数 - 值对。如果在步骤 S180 中判断没有要处理的参数，则在步骤 S210 输出对参数集中的所有参数的处理结果，并结束对该网页的 XSS 检测。

[0037] 应当注意的是，在上述步骤 S140 中，已经说明了针对一个参数可以组装出多个特定 URL。根据本发明的另一个实施例，可以重复执行步骤 S140-S170 来将每个特别组装的 URL 发送到网络服务器以确定网页是否存在 XSS 漏洞。这可以更全面地对网页进行测试。

[0038] 图 2 示出了在步骤 S170 中执行的、用于确定网络服务器对某个参数的处理是否存在 XSS 漏洞的方法 1700 的流程图。

[0039] 在步骤 S1710 中，将所获取的网页内容转换为 DOM 模型。为了在用户端呈现网页内容，将网页内容转换为 DOM 模型是常用的技术手段之一，本发明需要模拟执行所获取的网页内容，因此首先将网页内容转换为 DOM 模型。随后在步骤 S1720 中，利用根据本发明的 javascript 解析引擎来执行 DOM 模型中的 javascript 脚本。如上所述，在针对某个参数组装特定 URL 时，已经把特定的 javascript 脚本插入到参数值中了。因此，根据在步骤 S1730

中，根据 javascript 解析引擎是否执行了该特定的脚本来判断网络服务器针对该参数的处理是否存在 XSS 漏洞。如果该特定脚本被触发执行了，则说明存在 XSS 漏洞 (S1740)，否则则说明不存在 XSS 漏洞 (S1750)。此后，在步骤 S1760 中记录对该参数的处理是否存在 XSS 漏洞的判断，并且结束该方法。

[0040] 应当注意的是，图 2 所示方法的主要目的是利用根据本发明的 javascript 解析引擎来模拟执行返回的网页内容，因此所有可以对网页内容进行模拟执行的方式都在本发明的保护范围之内。

[0041] 另外，如上所述，在 javascript 脚本中，alert 函数的功能为弹出一个提示窗口，其不会对网页内容产生影响，因此，优选地，插入到参数值中的脚本为 alert 函数，而根据本发明的 javascript 解析引擎也对 alert 函数的处理进行修改，以便根据 alert 函数是否被触发来确定对该参数的处理是否存在 XSS 漏洞。

[0042] 图 3 示出了根据本发明实施例的、用于检测网页的 XSS 漏洞的 XSS 漏洞检测设备的示意图。

[0043] 如图 3 所示，XSS 漏洞检测设备 300 包括网页参数 - 值对集合确定装置 310、测试 URL 组装装置 320、请求发送装置 330、网页内容获取装置 340、模拟执行装置 350、javascript 解析引擎 360 和记录装置 370。

[0044] 网页参数 - 值对集合确定装置 310 确定某个动态网页可以接收的参数 - 值对集合，如上面参考步骤 S110 所述的那样，可以通过各种方式来确定动态网页可以接收的参数 - 值对集合。随后网页参数 - 值对集合确定装置 310 将所确定的参数 - 值对集合发送到测试 URL 组装装置 320。

[0045] 测试 URL 组装装置 320 为所接收到的参数 - 值对集合中的每个参数 - 值对构造测试用的 URL。在为某个参数 - 值对构造测试 URL 时，测试 URL 组装装置 320 参考本发明定制的 javascript 引擎 360 在值中插入相应的 javascript 脚本，诸如上面所述的 <script>alert(0)</script> 等，以便如果该参数存在 XSS 漏洞，则该相应的 javascript 脚本就会在 javascript 引擎 360 中触发。因此，本发明的 javascript 解析引擎 360 中提供了要插入到值中的 javascript 脚本。

[0046] 如上所述，测试 URL 组装装置 320 可以为一个参数 - 值对构造出多个包含特定 javascript 脚本和不同特殊字符的 URL，并且可以分别对这些组装的 URL 进行测试。

[0047] 请求发送装置 330 接收由测试 URL 组装装置 320 组装的 URL，并且将该 URL 发送到网络服务器 400 以请求动态网页，作为响应，网络服务器 400 将所生成的动态网页发送到网页内容获取装置 340。网页内容获取装置获取该动态网页，并且将动态网页内容发送给模拟执行装置 250 以检测该动态网页是否具有 XSS 漏洞。

[0048] 在实践中，可以将请求发送装置 330 和网页内容获取装置 340 的功能并入到同一个请求和接收装置中，以统一执行网络相关功能。这些都在本发明的保护范围之内。

[0049] 模拟执行装置 250 以各种方式模拟执行所获取的网页内容，例如，可以诸如 IE、Firefox 或者 Chrome 之类浏览器内核的方式来处理网页内容，并且将网页内容转换为 DOM 模型，然后利用 javascript 解析引擎 360 来执行网页中的 javascript 脚本。

[0050] Javascript 解析引擎 360 在执行 javascript 脚本时，判断所提供的、要插入到值中的 javascript 脚本是否被触发了，如果被触发了，则认为网络服务器对该动态网页中的

参数的处理存在 XSS 漏洞，如果未被触发，则认为对该参数的处理不存在 XSS 漏洞。随后，javascript 解析引擎 360 将该确定结果发送到记录装置 370 进行记录。

[0051] 在上述XSS漏洞检测设备中，公开了为一个动态网页进行XSS漏洞检测的方式。这个 XSS 漏洞检测设备可以并入到网站安全扫描系统或者网络扫描系统中，以便为网站安全扫描系统或者网络扫描系统所扫描的每个动态网页进行 XSS 漏洞检测，从而为某个或者多个网站进行 XSS 漏洞扫描。

[0052] 由于本发明中的 javascript 解析引擎 360 的主要目的就是为了判断特定的 javascript 函数是否在返回的动态网页中被触发了，因此，为了提高 javascript 解析引擎 360 的执行效率，可以对其它 javascript 函数的执行进行简化处理，例如仅仅实现其接口而不执行任何实质性的操作，从而加快模拟执行动态网页内容的速度。

[0053] 根据本发明的 XSS 漏洞检测方法和设备通过组装具有特定脚本的参数值的 URL，并检测该 URL 返回的动态网页内容是否会触发该特定脚本来确定该动态网页是否具有 XSS 漏洞。和传统上仅仅通过判断相应内容是否会出现在动态网页中的特征匹配相比，其准确度更高。另外，随着 AJAX 技术等的发展，客户端网页内容呈现和处理方式变得日益复杂，传统的特征匹配方式不能准确判断 XSS 漏洞。由于 XSS 漏洞最终会通过执行脚本来进行，因此本发明可以很好地应用于富客户端的网页内容中。

[0054] 应该注意的是，上述实施例对本发明进行说明而不是对本发明进行限制，并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中，不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中，这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

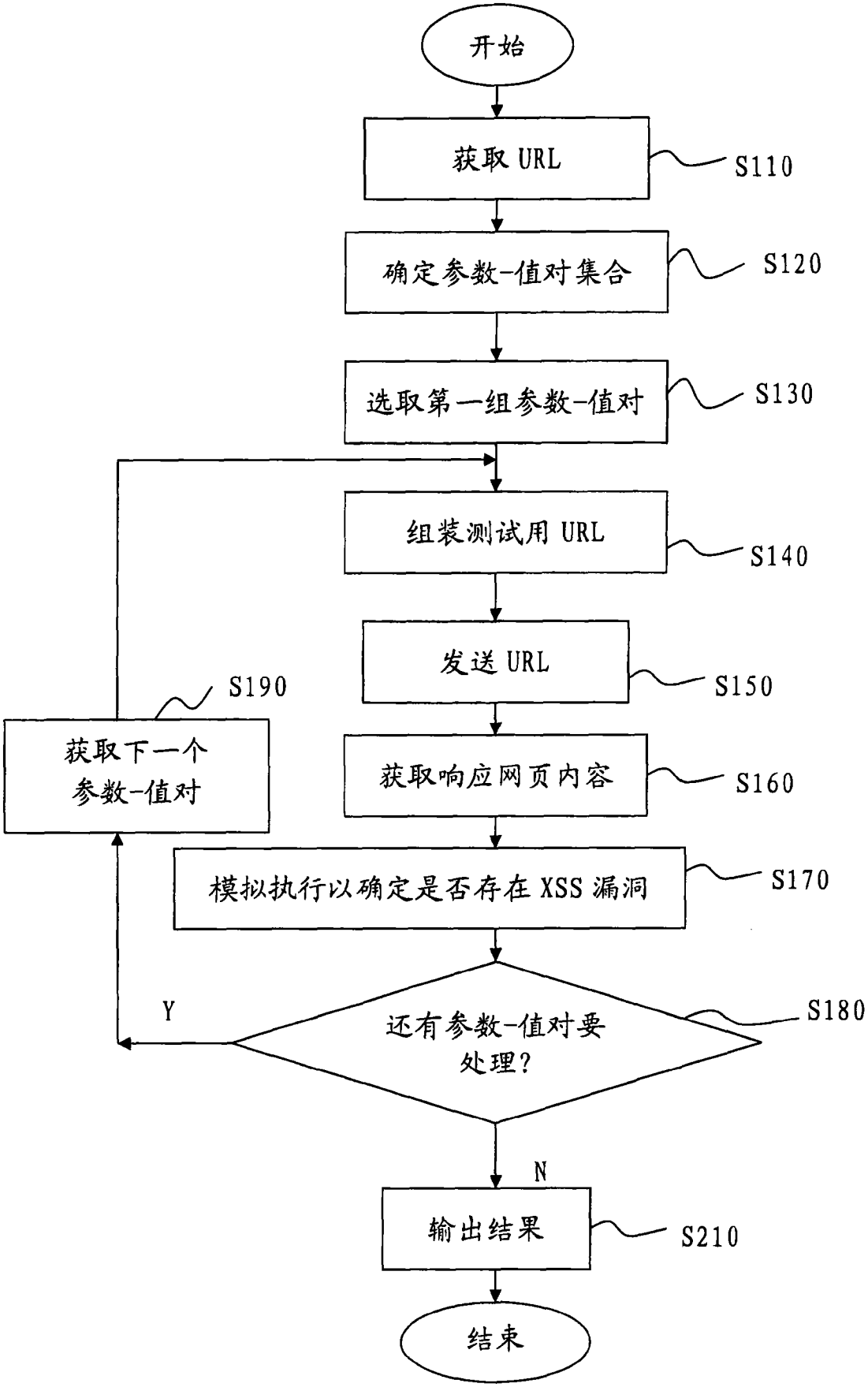

开始
获取 URL
S110
确定参数-值对集合
S120
选取第一组参数-值对
S130
组装测试用 URL
S140
发送 URL
S150
获取响应网页内容
S160
模拟执行以确定是否存在 XSS 漏洞
S170
还有参数-值对要处理?
S180
Y
N
获取下一个参数-值对
S190
输出结果
S210
结束

图 1

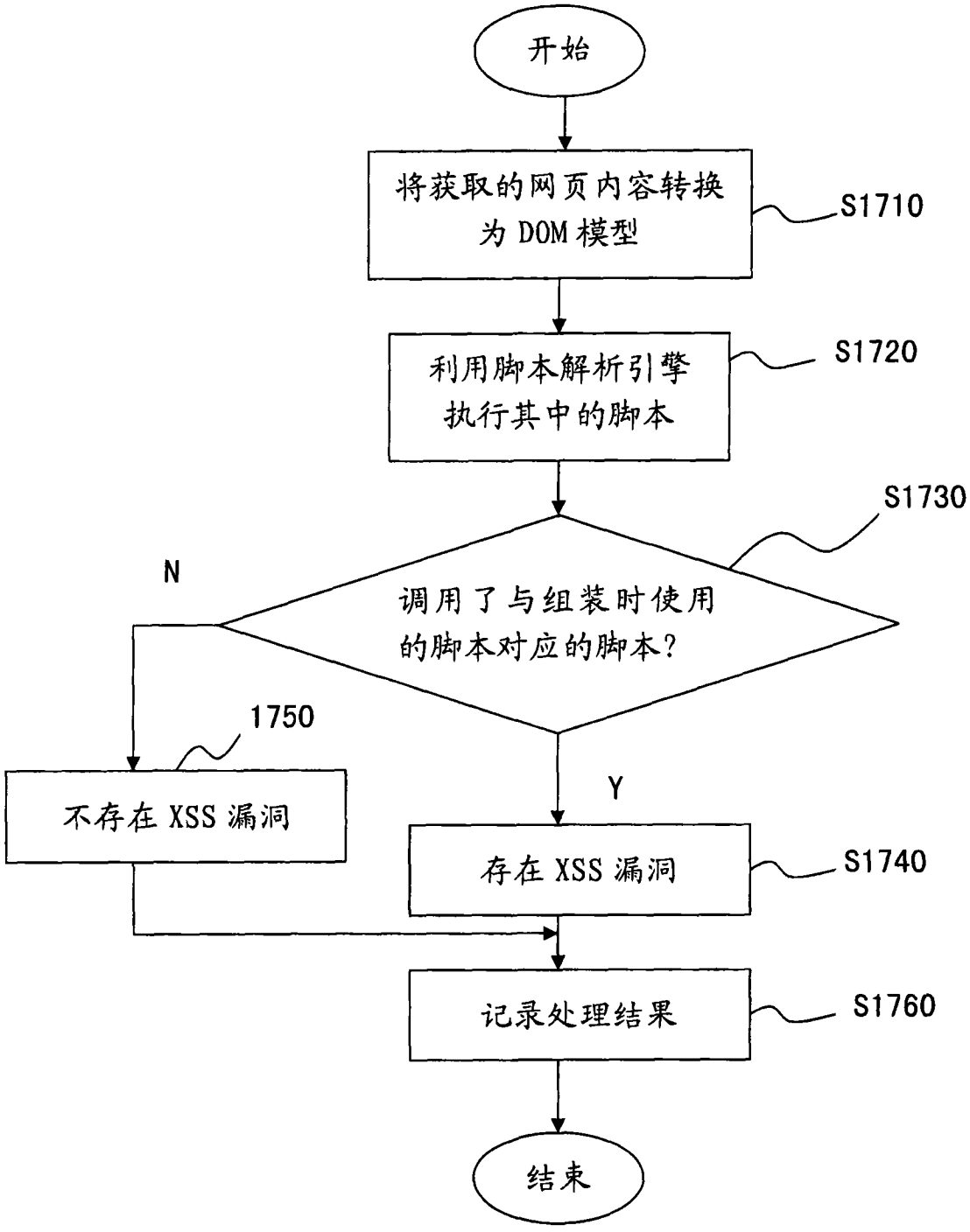

开始
将获取的网页内容转换为 DOM 模型
S1710
利用脚本解析引擎执行其中的脚本
S1720
S1730
调用了与组装时使用的脚本对应的脚本?
N
Y
1750
不存在 XSS 漏洞
存在 XSS 漏洞
S1740
记录处理结果
S1760
结束

图 2

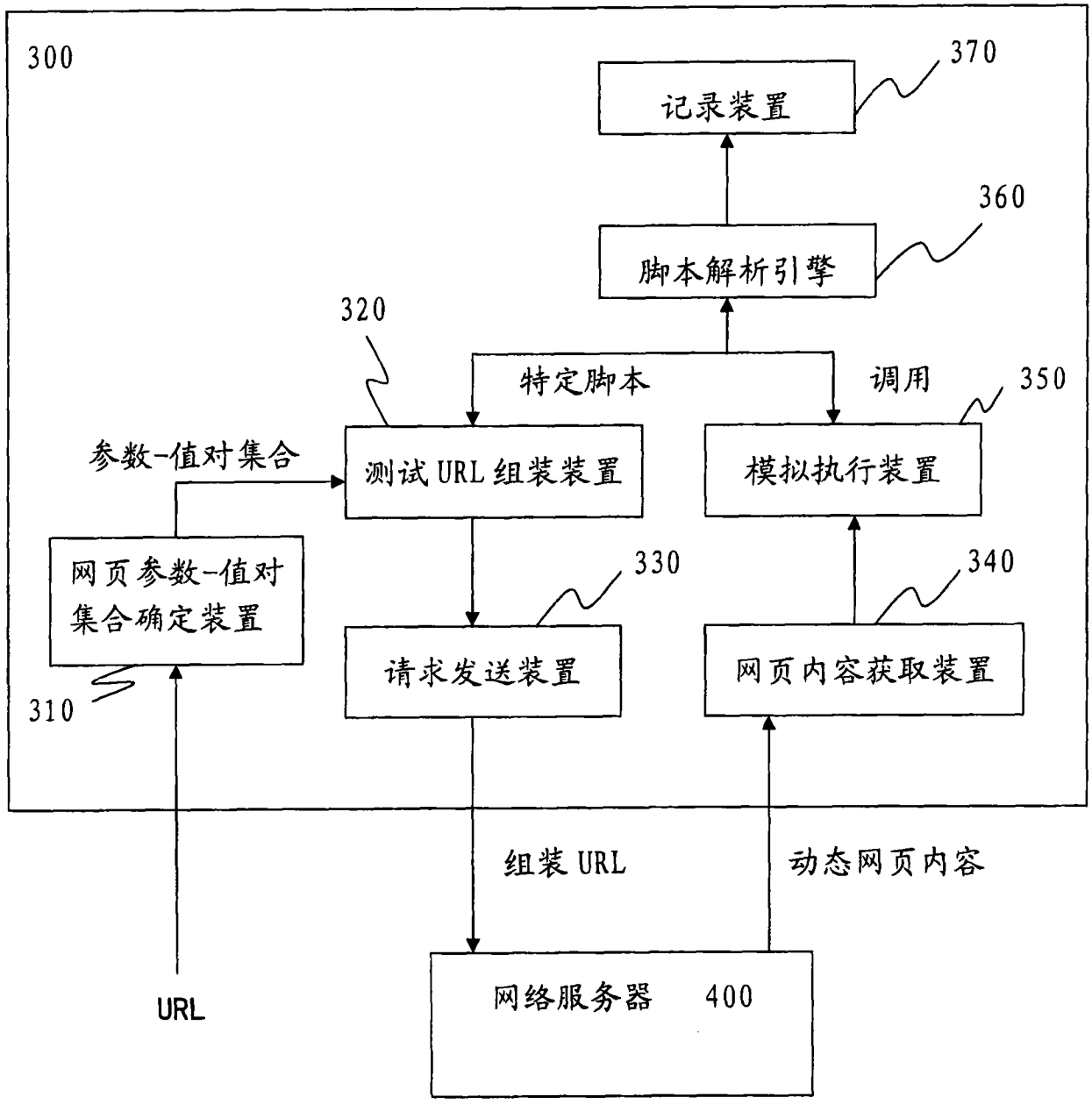

300
370
记录装置
360
脚本解析引擎
320
特定脚本
调用
350
参数-值对集合
测试 URL 组装装置
模拟执行装置
网页参数-值对集合确定装置
330
340
请求发送装置
网页内容获取装置
310
组装 URL
动态网页内容
URL
网络服务器 400

图 3