(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 113259322 A
(43)申请公布日 2021.08.13

(21)申请号 202110419638.9

(22)申请日 2021.04.19

(71)申请人 山东英信计算机技术有限公司
地址 250101 山东省济南市高新区浪潮路1036号浪潮科技园S05号楼北3层北区

(72)发明人 刘力

(74)专利代理机构 北京权智天下知识产权代理事务所(普通合伙) 11638
代理人 王新爱

(51)Int.Cl.
*H04L 29/06*(2006.01)
*H04L 29/08*(2006.01)

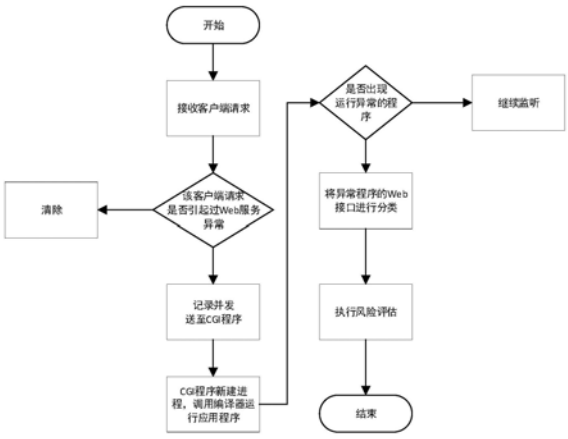
权利要求书2页 说明书6页 附图3页

(54)发明名称

一种预防Web服务异常的方法、系统及介质

(57)摘要

本发明公开了一种预防Web服务异常的方法，应用于Web服务器，包括以下步骤：创建规则库，建立存储异常Web接口的访问记录及接口信息的规则库，记录Web接口的访问次数；通过Web接口接收客户端请求，并执行过滤操作，根据过滤操作结果设置环境变量，建立进程，并调用编译器运行应用程序；存储异常接口：将触发进程异常的Web接口记为异常Web接口，并将该异常Web接口的访问记录及接口信息存入所述规则库内；异常风险评估：根据规则库内的访问记录及接口信息评估Web接口的风险值和异常类型的百分比，本发明能够拦截触发Web异常的客户端请求，提高Web服务的安全性和稳定性，并计算Web接口风险值及异常类型占比。




CN 113259322 A

1.一种预防Web服务异常的方法,应用于Web服务器,其特征在于,包括以下步骤:

创建规则库:建立存储异常Web接口的访问记录及接口信息的规则库,记录Web接口的访问次数;

过滤客户端请求:通过所述Web接口接收客户端请求,并对所述客户端请求执行过滤操作,根据过滤操作结果执行运行应用程序步骤;

运行应用程序:根据所述客户端请求设置环境变量,建立进程,并调用编译器运行应用程序;

存储异常接口:监听所述进程,将触发所述进程异常的所述Web接口记为异常Web接口,并将该异常Web接口的访问记录及接口信息存入所述规则库内;

异常风险评估:根据所述规则库内的访问记录及接口信息评估所述Web接口的风险值和异常类型的百分比。

2.根据权利要求1所述的预防Web服务异常的方法,其特征在于:所述过滤客户端请求的步骤进一步包括:若所述规则库存在与所述客户端请求对应的所述Web接口的访问记录,则将所述客户端请求清除;

若所述规则库不存在与所述客户端请求对应的所述Web接口的访问记录,则执行所述运行应用程序步骤。

3.根据权利要求1所述的预防Web服务异常的方法,其特征在于:所述异常风险评估的步骤进一步包括:

将所述Web接口的所述接口信息按照接口类型和异常类型分类;

计算所述Web接口的所述异常类型的风险值,并根据所述异常类型的风险值执行累加操作,得出所述Web接口的风险值。

4.根据权利要求3所述的预防Web服务异常的方法,其特征在于:所述异常风险评估的步骤进一步还包括:

获取所述异常Web接口的访问次数;

获取所述异常Web接口触发所述进程出现异常的所述异常类型;

将所述异常类型及所述访问次数执行统计操作,得出所述异常类型的百分比。

5.根据权利要求4所述的预防Web服务异常的方法,其特征在于:所述计算所述Web接口的所述异常类型的风险值的步骤进一步包括:

获取所述Web接口的异常类型的异常率;

获取所述Web接口的访问次数;

根据所述Web接口的异常类型的异常率及所述Web接口的访问次数执行加权操作,得出所述Web接口的所述异常类型的风险值。

6.根据权利要求1所述的预防Web服务异常的方法,其特征在于:在时间阈值内执行所述异常风险评估步骤。

7.一种预防Web服务异常的系统,应用于Web服务器,其特征在于,所述系统包括:请求处理模块、异常分析模块、规则模块、风险评估模块及CGI模块;

所述规则模块分别与所述请求处理模块、所述异常分析模块及所述风险评估模块相连;所述异常分析模块与所述规则模块相连;

所述请求处理模块分别与所述Web服务器和所述CGI模块相连;

所述Web服务器通过Web接口发送客户端请求；

所述请求处理模块用于根据所述规则模块中存储的异常Web接口，对所述客户端请求执行过滤操作；

所述CGI模块用于建立进程并根据所述客户端请求填写环境变量；

所述进程用于调用编译器运行所述应用程序；

所述异常分析模块用于监听所述进程运行所述应用程序的运行状态；

所述规则模块用于存储触发所述进程异常的所述Web接口的访问记录及接口信息；

所述风险评估模块用于评估所述Web接口的风险值及其异常类型百分比。

8.根据权利要求7所述的预防Web服务异常的系统，其特征在于：

所述请求处理模块还用于将所述Web接口的访问记录发送至风险评估模块；

当所述异常分析模块监听到所述进程出现异常状态时，所述异常分析模块用于将触发所述进程出现异常状态的所述Web接口记录并发送至所述规则模块。

9.根据权利要求7所述的预防Web服务异常的系统，其特征在于：

所述规则模块还用按照所述Web接口的接口类型、接口参数及异常类型执行分类操作和统计操作。

10.一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1-6任一项所述的预防Web服务异常的方法步骤。

# 一种预防Web服务异常的方法、系统及介质

## 技术领域

[0001] 本发明涉及网站安全技术领域，特别是涉及一种预防Web服务异常的方法、系统及介质。

## 背景技术

[0002] 网站服务的后端一般分为两个部分，第一部分为Web服务器，第二部分为应用程序模块，Web服务器与应用程序模块通过CGI程序完成数据交换，CGI(Common Gateway Interface)是公共网关接口，CGI作为Web服务器与应用程序模块的一个标准接口，根据CGI标准，编写外部扩展应用程序，可以对客户端浏览器输入的数据进行处理，完成客户端与Web服务器的交互操作。

[0003] 在CGI标准中，Web服务器负责监听来自客户端的请求，外部扩展应用程序负责完成实际业务逻辑相关功能，由于在编写外部扩展应用程序时，代码的编写不规范，或者客户端用户输入处理不安全，及运行环境复杂等各种原因，外部扩展应用程序在接受到客户端请求并运行数据时可能发生各种未知的异常，这种异常对Web服务器的安全性及稳定性造成极大的威胁。

## 发明内容

[0004] 本发明主要解决的是外部扩展应用程序在接受到客户端请求并运行数据时发生各种未知的异常，对Web服务器的安全性及稳定性造成极大的威胁的问题。

[0005] 为解决上述技术问题，本发明采用的一个技术方案是：提供一种预防Web服务异常的方法，应用于Web服务器，包括以下步骤：

[0006] 创建规则库：创建规则库，存储异常Web接口的访问记录及接口信息，并记录所述Web接口的访问次数；

[0007] 过滤客户端请求：通过Web接口接收客户端请求，并对所述客户端请求执行过滤操作，根据过滤操作结果执行运行应用程序步骤；

[0008] 运行应用程序：根据所述客户端请求设置环境变量，建立进程，并调用编译器运行应用程序；

[0009] 存储异常接口：监听所述进程，并获取触发所述进程异常的所述异常Web接口的访问记录及接口信息；

[0010] 异常风险评估：在时间阈值内，根据所述访问记录及所述接口信息评估所述Web接口的风险值和异常类型的百分比。

[0011] 进一步，所述过滤客户端请求的步骤进一步包括：若所述规则库存在所述客户端请求对应的所述异常Web接口的访问记录，则将所述客户端请求清除；

[0012] 若所述规则库不存在所述客户端请求对应的所述异常Web接口的访问记录，则执行所述运行应用程序步骤。

[0013] 进一步，所述异常风险评估的步骤进一步包括：

[0014] 将所述Web接口的所述接口信息按照接口类型和异常类型分类；

[0015] 计算所述Web接口的所述异常类型的风险值，并根据所述异常类型的风险值执行累加操作，得出所述Web接口的风险值。

[0016] 进一步，所述异常风险评估的步骤进一步还包括：

[0017] 获取所述异常Web接口的访问次数；

[0018] 获取所述异常Web接口触发所述进程出现异常的所述异常类型；

[0019] 将所述异常类型及所述访问次数执行统计操作，得出所述异常类型的百分比。

[0020] 进一步，所述计算所述Web接口的所述异常类型的风险值的步骤进一步包括：

[0021] 获取所述Web接口的异常类型的异常率；

[0022] 获取所述Web接口的访问次数；

[0023] 根据所述Web接口的异常类型的异常率及所述Web接口的访问次数执行加权操作，得出所述Web接口的所述异常类型的风险值。

[0024] 进一步，在时间阈值内执行所述异常风险评估步骤。

[0025] 本发明还提供一种预防Web服务异常的系统，应用于Web服务器，所述系统包括：请求处理模块、异常分析模块、规则模块、风险评估模块及CGI模块；

[0026] 所述规则模块分别与所述请求处理模块、所述异常分析模块及所述风险评估模块相连；所述异常分析模块与所述规则模块相连；

[0027] 所述请求处理模块分别与所述Web服务器和所述CGI模块相连；

[0028] 所述Web服务器通过Web接口发送客户端请求；

[0029] 所述请求处理模块用于根据所述规则模块中存储的异常Web接口，对所述客户端请求执行过滤操作；

[0030] 所述CGI模块用于建立进程并根据所述客户端请求填写环境变量；

[0031] 所述进程用于调用编译器运行所述应用程序；

[0032] 所述异常分析模块用于监听所述进程运行所述应用程序的运行状态；

[0033] 所述规则模块用于存储触发所述进程异常的所述Web接口的访问记录及接口信息；

[0034] 所述风险评估模块用于评估所述Web接口的风险值及其异常类型百分比。

[0035] 进一步，所述请求处理模块还用于将所述Web接口的访问记录发送至风险评估模块；

[0036] 当所述异常分析模块监听到所述进程出现异常状态时，所述异常分析模块用于将触发所述进程出现异常状态的所述Web接口记录并发送至所述规则模块。

[0037] 进一步，所述规则模块还用按照所述Web接口的接口类型、接口参数及异常类型执行分类操作和统计操作。

[0038] 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计算机程序，所述计算机程序被处理器执行时，实现所述的预防Web服务异常的方法步骤。

[0039] 本发明的有益效果是：

[0040] 1、本发明所述的预防Web服务异常的方法，可以实现拦截客户端请求，并将客户端请求与触发进程的客户端请求进行匹配，而且还可以实现对Web接口执行分类和统计操作，并计算出Web接口的风险值及异常类型占比。

[0041] 2、本发明所述的预防Web服务异常的系统，可以实现拦截触发Web异常的客户端请求，提高Web服务的安全性和稳定性，同时根据实际Web接口的访问情况，计算Web接口风险值及异常类型占比。

[0042] 3、本发明所述的预防Web服务异常的介质，可以实现拦截客户端请求，并对Web接口执行分类和统计操作。

## 附图说明

[0043] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案，下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施方式，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0044] 图1是本发明实施例1所述的预防Web服务异常的方法的流程图；

[0045] 图2是本发明实施例1所述的预防Web服务异常的方法的示意图；

[0046] 图3是本发明实施例2所述的预防Web服务异常的系统的拓扑图。

## 具体实施方式

[0047] 下面将结合附图对本发明的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0048] 需要说明的是，在本发明的描述中，

[0049] Web(World Wide Web)是全球广域网。

[0050] 实施例1

[0051] 本发明实施例提供一种预防Web服务异常的方法，应用于Web服务器，请参阅图1和图2，包括以下步骤：

[0052] S21、创建规则库。

[0053] 该步骤进一步包括：创建规则库，存储异常Web接口的访问记录及接口信息，并记录所述Web接口的访问次数；

[0054] 为了保证Web服务器接收到的客户端请求未曾引起过Web服务异常，本实施例创建一种规则库，可以使表格也可以是文本，该规则库可以将Web服务器接收到的所有客户端请求、运行的所有程序记录。

[0055] S22、过滤客户端请求。

[0056] 该步骤进一步包括：通过Web接口接收客户端请求，并对所述客户端请求执行过滤操作，根据过滤操作结果执行运行应用程序步骤。

[0057] 客户通过客户端向Web服务器发送客户端请求，为了保证客户端请求的完整性与安全性，首先对客户端请求进行过滤操作，验证该客户端请求通过的Web接口曾经是否引起过Web服务的异常，将曾经引起过Web服务异常的客户端请求对应的Web接口的访问记录存储，若曾经引起过Web服务，则将该客户端请求清除，保证不会再次影响Web服务，若未引起过Web服务异常，则将该客户端请求发送至CGI程序。

[0058] S23、运行应用程序。

[0059] 该步骤进一步包括:根据所述客户端请求填写环境变量,建立进程,并调用编译器运行应用程序。

[0060] CGI模块根据客户端请求新建一个进程,根据该进程调用编译器,编译器运行应用程序,应用程序根据客户端请求运行,因为一个应用程序一般为多个Web接口导入的数据构成,所以需要时刻监听进程的运行状态。

[0061] 获取Web服务器上所有的进程的Web接口的接口信息,并按照接口类型及导致进程异常的异常类型进行分类。

[0062] S24、存储异常接口。

[0063] 该步骤进一步包括:监听所述进程,并获取触发所述进程异常的异常Web接口的访问记录及接口信息。

[0064] 因为进程运行状态有可能因为Web接口出现异常,所以要在一段时间内对进程的运行状态进行监测,该监测时间可以根据Web服务器的安全等级设定,若在这段时间内,进程出现异常,则将该进程相关的所有Web接口进行记录,并将该接口信息及异常类型进行记录。

[0065] S25、异常风险评估。

[0066] 该步骤进一步包括:在时间阈值内,根据所述访问记录及所述接口信息评估所述Web接口的风险值及异常类型的百分比。

[0067] 在人为规定的时间阈值内,该阈值由Web服务器的安全等级设定,将导致进程异常的接口信息按照Web接口的接口类型及异常类型进行分类,并对该异常的Web接口的访问次数进行统计。

[0068] 根据访问次数及异常发生的次数计算该异常类型的异常率,因为一个Web接口可能有多种异常类型,所以需要对单独的类型进行统计,并得出该异常类型的特定异常类型风险值。

[0069] 特定异常类型风险值的计算方法为根据该异常类型的异常率及该Web接口的访问量进行加权得出。

[0070] 得出单种的异常类型后,对Web接口出现的所有异常类型进行累加,得出该Web接口的风险值。

[0071] 或者,根据整个Web服务器的Web接口的异常类型执行统计操作,得出每个Web服务器的异常类型的占比。

[0072] 实施例2

[0073] 本发明实施例提供一种预防Web服务异常的系统,应用于Web服务器,请参阅图3,包括:

[0074] 请求处理模块、异常分析模块、规则模块、风险评估模块及CGI模块。

[0075] 规则模块分别与请求处理模块、异常分析模块及风险评估模块相连;异常分析模块与规则模块相连;

[0076] 请求处理模块分别与Web服务器和CGI模块相连;

[0077] 因CGI模块作为Web服务器及应用程序模块之间的转接模块,所以当客户端向Web服务器发送客户端请求时,Web服务器将该客户端请求通过若干Web接口发送至CGI模块时,请求处理模块可以截取客户端请求,因客户端请求为一连串的字符组成,且相同的请求对

应的字符相同，所以在本实施例中，为了方便描述将设定若干个客户端请求，设定第一客户端请求及第二客户端请求，即第一客户端请求对应的字符为0x00，第二客户端请求对应的字符为0x01，若在规则模块中存在第一客户端请求对应的字符0x00，则证明该第一客户端请求曾经出现过异常，该第一客户端请求有可能会触发Web服务异常，则请求处理模块将拦截第一客户端请求并将其清除。

[0078] 若在规则模块中不存在第一客户端请求对应的字符0x00，则证明该第一客户端请求未曾引起过Web服务异常，请求处理模块则不会清理该第一客户端请求，请求处理模块将该第一客户端请求发行至CGI模块，同时将该第一客户端记录。

[0079] 相同的，对于第二客户端请求，若在规则模块中存在第二客户端请求对应的字符0x01，则证明该第二客户端请求曾经引起Web服务出现过异常，导致过Web服务异常，则请求处理模块将拦截第二客户端请求并将其清除。

[0080] 若在规则模块中不存在第二客户端请求对应的字符0x01，则证明该第二客户端请求未曾引起过Web服务异常，请求处理模块则不会清理该第二客户端请求，请求处理模块将该第二客户端请求发行至CGI模块，同时将该第一客户端记录。

[0081] CGI模块首先根据客户端请求建立一个新的进程，该进程调用编译器，因为编译器需要设定环境变量，所以CGI模块根据客户端请求将编译器的环境变量进行设定，编译器作为运行进程的一个功能性部件，编译器运行应用程序，应用程序根据客户端请求运行，该进程面向客户端，并将客户端请求进行运算等一系列操作，一系列的Web接口发送的数据提供给进程，该进程进行运算有可能出现异常，所以本实施例，提出异常分析模块及规则模块对引起进程异常的一系列Web接口进行处理。

[0082] 异常分析模块用于监听进程的运行状态，对所有的新建的进程进行监听，若监听到编译器抛出的异常信息后，则将对应的Web接口以及引起进程调用编译器异常的Web接口的信息记录至规则模块。

[0083] 规则模块用于存储请求处理模块记录的每一个客户端请求的访问记录。

[0084] 规则模块还用于存储异常分析模块监听到的导致进程产生异常的Web接口的接口信息，本实施例中所提到的接口信息包括该Web接口的环境参数及该Web接口引起进程出现异常的异常类型。

[0085] 规则模块还用于将Web接口的访问记录以及导致进程产生异常的接口信息输出到请求处理模块及风险评估模块。

[0086] 因在服务器集群中，每个Web服务器都有对外响应客户端请求的Web接口，每个Web接口的安全性也是不相同的，所以需要对所有的Web接口进行风险评估，本发明实施例中，提出了风险评估模块可以对Web接口进行风险评估。

[0087] 风险评估模块用于对进程中的所有Web接口进行风险评估，因为Web服务器对用户端提供安全性是有差异的，即在安全性能要求较高的一段时间，需要对此进行风险评估，所以风险评估是具有时效性的，即在当前的时间段内谈论风险评估才有实际意义。

[0088] 在Web服务器内，为了加快对客户端的请求的响应，进程同时设有多种类型的Web接口，若干类型的Web接口对应客户端请求。

[0089] 所以风险评估模块首先计算出单个Web接口的风险值，计算原理为评估每种接口类型的的风险值，然后将该进程的所有的接口类型的接口进行加权得出此接口的风险值。

[0090] 本实施例中每一个接口类型的Web接口的风险值的计算原理为,在一段时间内所有服务器的该类型的异常率,以及该接口类型的Web接口的访问次数进行加权平均得出。

[0091] 通过单个接口的风险值的计算方法,将进程的所有Web接口根据接口类型进行计算,得出整个Web服务器的风险值。

[0092] 同样,在Web服务器中,对于风险评估的策略不只有一种,我们还可以通过占比的形式,统计异常类型所占的百分比。

[0093] 基于与前述实施例中方法同样的发明构思,本说明书实施例还提供一种计算机可读存储介质,计算机可读存储介质上存储有计算机程序,计算机程序被处理器执行时实现如前述公开的一种预防Web服务异常的方法的步骤。

[0094] 上述本发明实施例公开实施例序号仅仅为了描述,不代表实施例的优劣。

[0095] 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。

[0096] 以上所述仅为本发明的实施例,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。

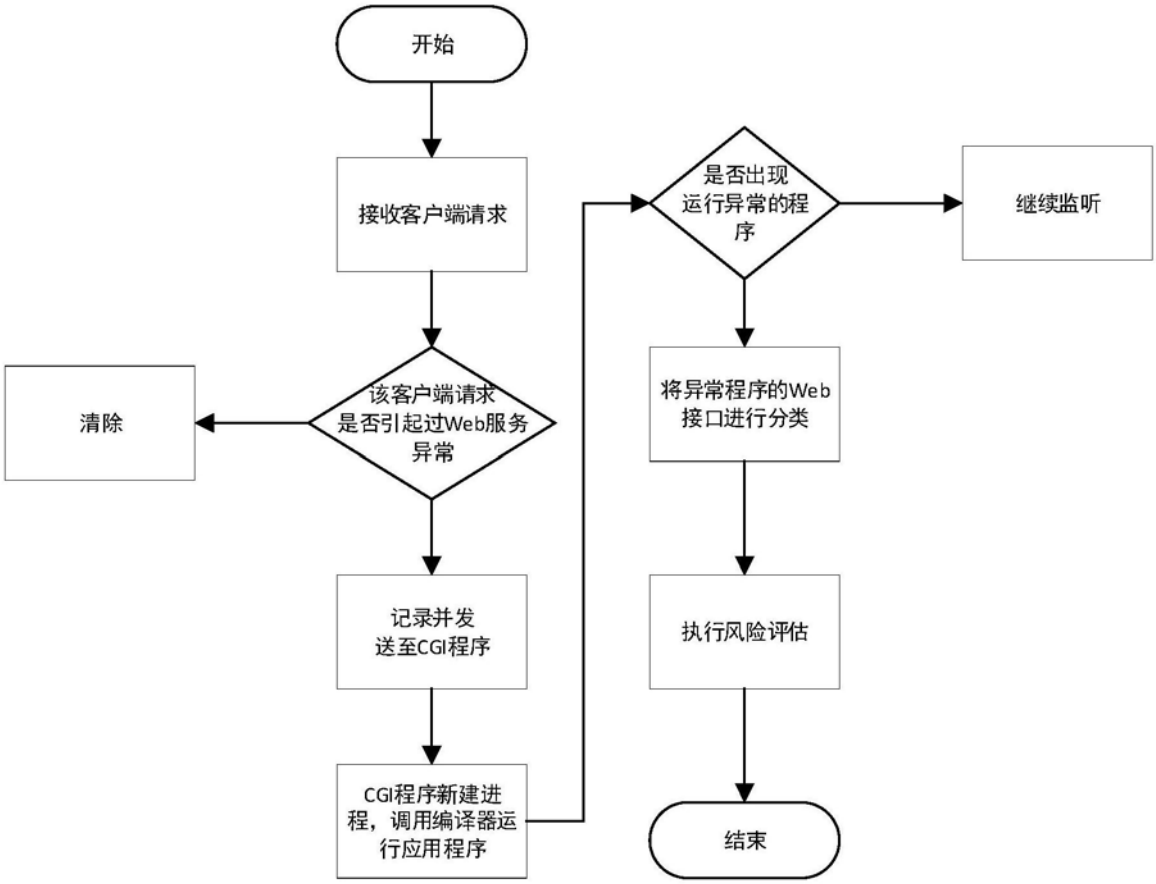

开始
接收客户端请求
该客户端请求是否引起过Web服务异常
清除
记录并发送至CGI程序
CGI程序新建进程，调用编译器运行应用程序
是否出现运行异常的程序
继续监听
将异常程序的Web接口进行分类
执行风险评估
结束

图1

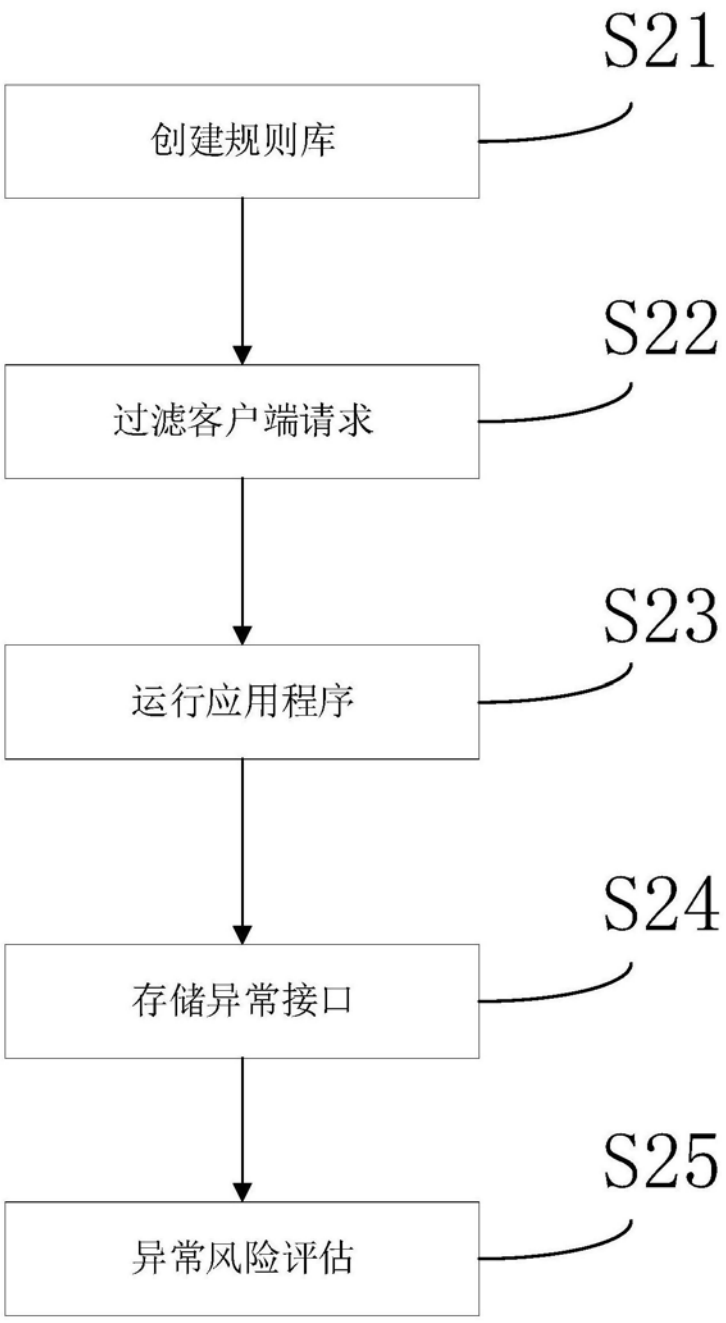



图2

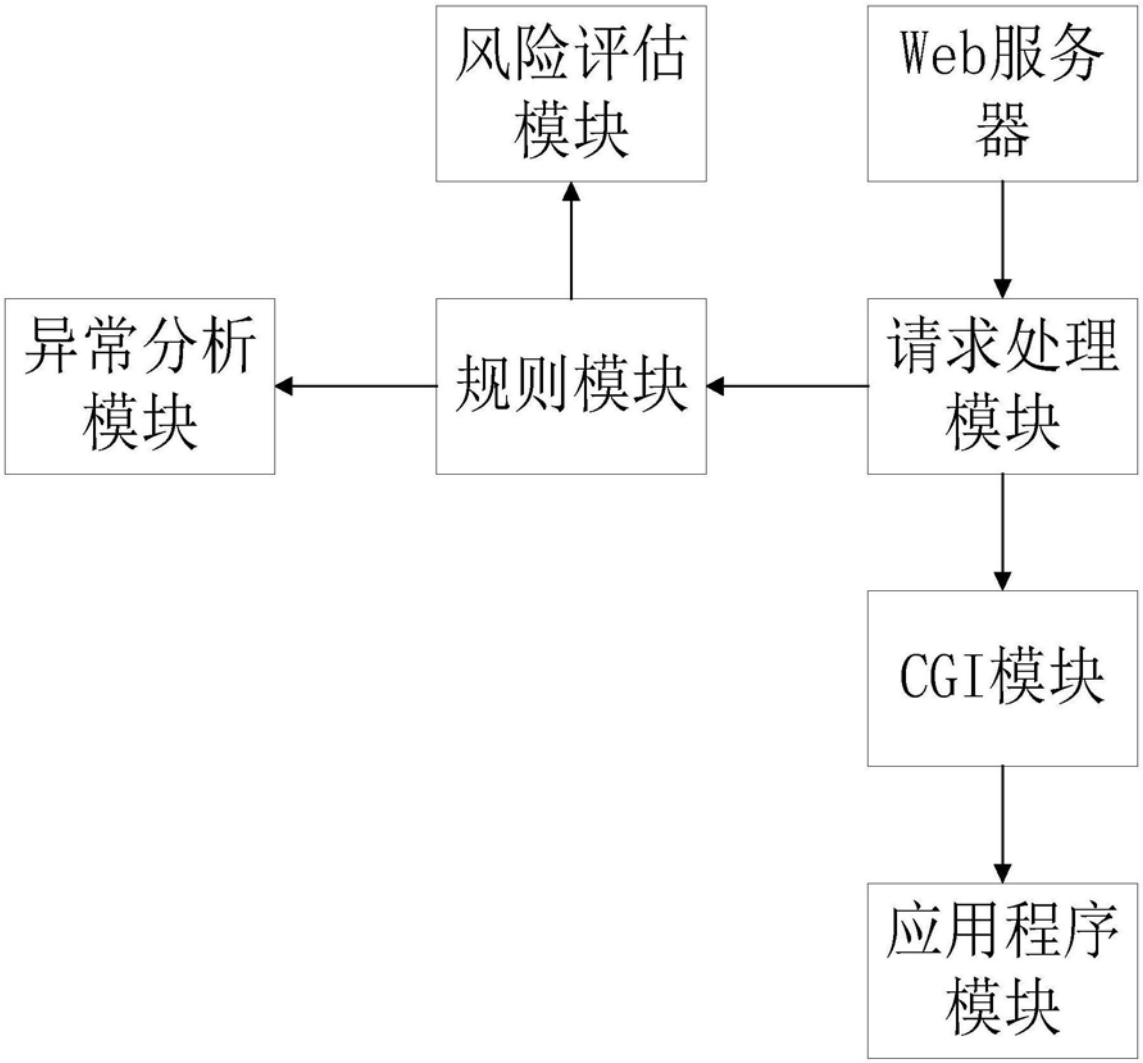



图3