(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 106815524 B
(45)授权公告日 2020.05.15

(21)申请号 201510866427.4

(22)申请日 2015.11.27

(65)同一申请的已公布的文献号
申请公布号 CN 106815524 A

(43)申请公布日 2017.06.09

(73)专利权人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 邵睿

(74)专利代理机构 北京博浩百睿知识产权代理有限责任公司 11134
代理人 宋子良

(51)Int.Cl.
G06F 21/56(2013.01)

(56)对比文件
CN 101667230 A,2010.03.10,说明书第2页第1段-第10页第1段,附图1-4.
CN 102945347 A,2013.02.27,说明书第5-180段,附图1-3.
CN 103258163 A,2013.08.21,全文.
US 2014/0181975 A1,2014.06.26,全文.

审查员 李佳曦

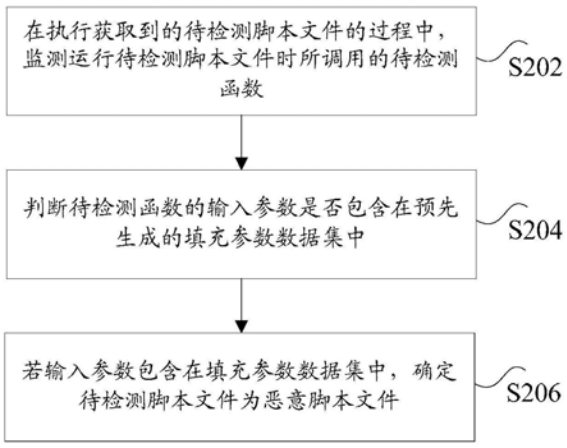
权利要求书2页 说明书8页 附图3页

(54)发明名称

恶意脚本文件的检测方法及装置

(57)摘要

本申请公开了一种恶意脚本文件的检测方法及装置。其中,该方法包括:在执行获取到的待检测脚本文件的过程中,监测运行待检测脚本文件时所调用的待检测函数;判断待检测函数的输入参数是否包含在预先生成的填充参数数据集中,其中,填充参数数据集包括用于页面交互的填充参数,填充参数为根据预先挂钩的预设函数及预设解释引擎生成的,预设函数用于输入填充参数,预设解释引擎用于检测是否需要输入填充参数;若输入参数包含在填充参数数据集中,确定待检测脚本文件为恶意脚本文件。本申请解决了由于基于静态特征提取的恶意脚本文件检测方法容易漏报潜在的恶意脚本文件造成的网页服务器安全性较低的技术问题。



CN 106815524 B

1.一种恶意脚本文件的检测方法,其特征在于,包括:

在执行获取到的待检测脚本文件的过程中,监测运行所述待检测脚本文件时所调用的待检测函数;

判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中,其中,所述填充参数数据集包括用于页面交互的填充参数,所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的,所述预设函数用于输入所述填充参数,所述预设解释引擎用于检测是否需要输入所述填充参数;

若所述输入参数包含在所述填充参数数据集中,确定所述待检测脚本文件为恶意脚本文件;

其中,在所述监测运行所述待检测脚本文件时所调用的待检测函数之前,所述方法还包括:挂钩所述预设函数及所述预设解释引擎,所述预设函数包括恶意脚本疑似函数以及用于变形所述输入参数的变形函数;通过所述预设解释引擎,检测是否需要输入用于页面交互的所述填充参数;若需要输入所述填充参数,调用所述恶意脚本疑似函数或所述变形函数输入所述填充参数;生成包含所述填充参数的所述填充参数数据集。

2.根据权利要求1所述的方法,其特征在于,所述挂钩所述预设函数及所述预设解释引擎包括:

通过超文本预处理器PHP扩展法挂钩所述预设函数及所述预设解释引擎,其中,所述PHP扩展法用于修改所述待检测脚本文件的执行逻辑。

3.根据权利要求1至2中任一项所述的方法,其特征在于,所述恶意脚本疑似函数包括以下一种或几种:用于将字符串作为PHP语法执行的函数、用于判断条件是否正确的函数、用于调用执行系统命令的函数以及用于进程执行的函数。

4.根据权利要求1至2中任一项所述的方法,其特征在于,所述变形函数包括以下一种或几种:用于编码解密的函数、用于解压缩的函数以及用于字符串旋转解密的函数。

5.一种恶意脚本文件的检测装置,其特征在于,包括:

监测单元,用于在执行获取到的待检测脚本文件的过程中,监测运行所述待检测脚本文件时所调用的待检测函数;

判断单元,用于判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中,其中,所述填充参数数据集包括用于页面交互的填充参数,所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的,所述预设函数用于输入所述填充参数,所述预设解释引擎用于检测是否需要输入所述填充参数;

确定单元,用于若所述输入参数包含在所述填充参数数据集中,确定所述待检测脚本文件为恶意脚本文件;

其中,所述装置还包括:挂钩单元,用于挂钩所述预设函数及所述预设解释引擎,所述预设函数包括恶意脚本疑似函数以及用于变形所述输入参数的变形函数;检测单元,用于通过所述预设解释引擎,检测是否需要输入用于页面交互的所述填充参数;调用单元,用于若需要输入所述填充参数,调用所述恶意脚本疑似函数或所述变形函数输入所述填充参数;生成单元,用于生成包含所述填充参数的所述填充参数数据集。

6.根据权利要求5所述的装置,其特征在于,所述挂钩单元用于执行以下步骤挂钩所述预设函数及所述预设解释引擎:

通过超文本预处理器PHP扩展法挂钩所述预设函数及所述预设解释引擎，其中，所述PHP扩展法用于修改所述待检测脚本文件的执行逻辑。

7.根据权利要求5至6中任一项所述的装置，其特征在于，所述恶意脚本疑似函数包括以下一种或几种：用于将字符串作为PHP语法执行的函数、用于判断是否正确的函数、用于调用执行系统命令的函数以及用于进程执行的函数。

8.根据权利要求5至6中任一项所述的装置，其特征在于，所述变形函数包括以下一种或几种：用于编码解密的函数、用于解压缩的函数以及用于字符串旋转解密的函数。

恶意脚本文件的检测方法及装置

技术领域

[0001] 本申请涉及信息安全领域，具体而言，涉及一种恶意脚本文件的检测方法及装置。

背景技术

[0002] 网页服务器被黑客入侵后，通常会植入一段恶意脚本文件，作为黑客使用的后门。常用的建站语言PHP(Hypertext Preprocessor，超文本预处理器)、ASP(Active Server Page，动态服务器页面)、JSP(Java Server Pages，Java服务页面)都会有相应的恶意脚本文件，其中，以PHP变化做多。PHP的语法灵活，对相同的实现可以用不同的脚本变形，导致传统的PHP恶意脚本文件检测的难度增加。

[0003] 目前的恶意脚本文件检测，大多使用的静态特征提取方式，然而，其对潜在的恶意脚本文件(例如变形的PHP恶意脚本文件)的检测效果不佳，从而容易造成漏报，这将导致网页服务器存在很大的安全隐患。

[0004] 针对上述的问题，目前尚未提出有效的解决方案。

发明内容

[0005] 本申请实施例提供了一种恶意脚本文件的检测方法及装置，以至少解决由于基于静态特征提取的恶意脚本文件检测方法容易漏报潜在的恶意脚本文件造成的网页服务器安全性较低的技术问题。

[0006] 根据本申请实施例的一个方面，提供了一种恶意脚本文件的检测方法，包括：在执行获取到的待检测脚本文件的过程中，监测运行所述待检测脚本文件时所调用的待检测函数；判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中，其中，所述填充参数数据集包括用于页面交互的填充参数，所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，所述预设函数用于输入所述填充参数，所述预设解释引擎用于检测是否需要输入所述填充参数；若所述输入参数包含在所述填充参数数据集中，确定所述待检测脚本文件为恶意脚本文件。

[0007] 根据本申请实施例的另一方面，还提供了一种恶意脚本文件的检测装置，包括：监测单元，用于在执行获取到的待检测脚本文件的过程中，监测运行所述待检测脚本文件时所调用的待检测函数；判断单元，用于判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中，其中，所述填充参数数据集包括用于页面交互的填充参数，所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，所述预设函数用于输入所述填充参数，所述预设解释引擎用于检测是否需要输入所述填充参数；确定单元，用于若所述输入参数包含在所述填充参数数据集中，确定所述待检测脚本文件为恶意脚本文件。

[0008] 在本申请实施例中，采用在执行获取到的待检测脚本文件的过程中，监测运行待检测脚本文件时所调用的待检测函数；判断待检测函数的输入参数是否包含在预先生成的填充参数数据集中，其中，填充参数数据集包括用于页面交互的填充参数，填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，预设函数用于输入填充参数，预设解释引擎

用于检测是否需要输入填充参数;若输入参数包含在填充参数数据集中,确定待检测脚本文件为恶意脚本文件的方式,通过动态执行待检测脚本文件,监测待检测脚本文件的待检测函数的输入参数,与基于预设函数及预设解释引擎的填充参数数据集相匹配,并不基于待检测脚本文件的特征值,而是从待检测脚本文件在动态执行过程中的输入参数入手,达到了确定待检测脚本文件是否为恶意脚本文件的目的,从而实现了增强网页服务器安全性的技术效果,进而解决了由于基于静态特征提取的恶意脚本文件检测方法容易漏报潜在的恶意脚本文件造成的网页服务器安全性较低的技术问题。

附图说明

[0009] 此处所说明的附图用来提供对本申请的进一步理解,构成本申请的一部分,本申请的示意性实施例及其说明用于解释本申请,并不构成对本申请的不当限定。在附图中:

[0010] 图1是根据本申请实施例的一种运行恶意脚本文件的检测方法的计算机终端的硬件结构框图;

[0011] 图2是根据本申请实施例的一种可选的恶意脚本文件的检测方法的流程示意图;

[0012] 图3是根据本申请实施例的另一种可选的恶意脚本文件的检测方法的流程示意图;

[0013] 图4是根据本申请实施例的一种可选的恶意脚本文件的检测装置的结构示意图;

[0014] 图5是根据本申请实施例的另一种可选的恶意脚本文件的检测装置的结构示意图。

具体实施方式

[0015] 为了使本技术领域的人员更好地理解本申请方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分的实施例,而不是全部的实施例。基于本申请中的实施例,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,都应当属于本申请保护的范围。

[0016] 需要说明的是,本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0017] 首先,本实施例涉及的技术术语解释如下:

[0018] 脚本文件:类似于DOS操作系统中的批处理文件,它可以将不同的命令组合起来,并按确定的顺序自动连续地执行。脚本文件是文本文件,用户可使用任一文本编辑器来创建脚本文件。脚本是批处理文件的延伸,是一种纯文本保存的程序,一般来说的计算机脚本程序是确定的一系列控制计算机进行运算操作动作的组合,在其中可以实现一定的逻辑分支等。脚本程序相对一般程序开发来说比较接近自然语言,可以不经编译而是解释执行,利

于快速开发或一些轻量的控制。

[0019] CS(Client/Server,客户机/服务器):是软件系统体系结构,通过它可以充分利用两端硬件环境的优势,将任务合理分配到Client端和Server端来实现,降低了系统的通讯开销。其基本原则是将计算机应用任务分解成多个子任务,由多台计算机分工完成,即采用“功能分布”原则。客户端完成数据处理,数据表示以及用户接口功能;服务器端完成DBMS(数据库管理系统)的核心功能。

[0020] BS(Browser/Server,浏览器/服务器):是WEB兴起后的一种网络结构模式,WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端,将系统功能实现的核心部分集中到服务器上,简化了系统的开发、维护和使用。

[0021] Webshell:是以asp、php、jsp或者cgi等网页文件形式存在的一种命令执行环境,也可以将其称做为一种网页后门。黑客在入侵了一个网站后,通常会将asp或php后门文件与网站服务器WEB目录下正常的网页文件混在一起,然后就可以使用浏览器来访问asp或者php后门,得到一个命令执行环境,以达到控制网站服务器的目的。

[0022] 挂钩(hook):是Windows中提供的一种用以替换DOS下“中断”的系统机制,在对特定的系统事件进行hook后,一旦发生已hook事件,对该事件进行hook的程序就会受到系统的通知,这时程序就能在第一时间对该事件做出响应。

[0023] PHP(Hypertext Preprocessor,超文本预处理器):是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点,利于学习,使用广泛,主要适用于Web开发领域。

[0024] 实施例1

[0025] 根据本申请实施例,还提供了一种恶意脚本文件的检测方法的方法实施例,需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0026] 本申请实施例一所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。以运行在计算机终端上为例,图1是本申请实施例的一种恶意脚本文件的检测方法的计算机终端的硬件结构框图。如图1所示,计算机终端10可以包括一个或多个(图中仅示出一个)处理器102(处理器102可以包括但不限于微处理器MCU或可编程逻辑器件FPGA等的处理装置)、用于存储数据的存储器104、以及用于通信功能的传输装置106。本领域普通技术人员可以理解,图1所示的结构仅为示意,其并不对上述电子装置的结构造成限定。例如,计算机终端10还可包括比图1中所示更多或者更少的组件,或者具有与图1所示不同的配置。

[0027] 存储器104可用于存储应用软件的软件程序以及模块,如本申请实施例中的恶意脚本文件的检测方法对应的程序指令/模块,处理器102通过运行存储在存储器104内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的应用程序的漏洞检测方法。存储器104可包括高速随机存储器,还可包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器104可进一步包括相对于处理器102远程设置的存储器,这些远程存储器可以通过网络连接至计算机终端10。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0028] 传输装置106用于经由一个网络接收或者发送数据。上述的网络具体实例可包括

计算机终端10的通信供应商提供的无线网络。在一个实例中，传输装置106包括一个网络适配器(Network Interface Controller,NIC)，其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中，传输装置106可以为射频(Radio Frequency,RF)模块，其用于通过无线方式与互联网进行通讯。

[0029] 在上述运行环境下，本申请提供了如图2所示的恶意脚本文件的检测方法。图2是根据本申请实施例一的恶意脚本文件的检测方法的流程图。

[0030] 步骤S202，在执行获取到的待检测脚本文件的过程中，监测运行待检测脚本文件时所调用的待检测函数。

[0031] 本申请上述步骤S202中，在执行待检测脚本文件的过程中，可以实时监测运行待检测脚本文件时所调用的待检测函数，其中，该待检测脚本文件可以是主机发送来的。

[0032] 需要说明的是，本申请实施例的主机可以是云主机，也可以是本地主机，还是CS(Client/Server,客户机/服务器)架构下的客户机，或者可以BS(Browser/Server,浏览器/服务器)架构下的客户机，本实施例对此不作限定。

[0033] 步骤S204，判断待检测函数的输入参数是否包含在预先生成的填充参数数据集中。

[0034] 本申请上述步骤S204中，监测待检测函数的输入参数，判断该输入参数是否包含在上述填充参数数据集中。其中，填充参数数据集包括用于页面交互的填充参数，填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，预设函数用于输入填充参数，预设解释引擎用于检测是否需要输入填充参数。

[0035] 可选地，恶意脚本疑似函数包括以下一种或几种：用于将字符串作为PHP语法执行的函数(例如eval函数)、用于判断条件是否正确的函数(例如assert函数)、用于调用执行系统命令的函数(例如syste函数、exec函数、shell_exec函数)以及用于进程执行的函数(例如proc_open函数)。

[0036] 可选地，变形函数包括以下一种或几种：用于编码解密的函数(例如base64_decode函数)、用于解压缩的函数(例如gzinflate函数、gzuncompress函数、zlib_decode函数)、用于字符串旋转解密的函数(例如str_rot13函数)以及(+.)函数。

[0037] 步骤S206，若输入参数包含在填充参数数据集中，确定待检测脚本文件为恶意脚本文件。

[0038] 本申请上述步骤S206中，如果输入参数包含在填充参数数据集中，则认为待检测脚本文件为恶意脚本文件。其中，恶意脚本文件具体可以是指Webshell文件，本实施例对此不作限定。

[0039] 由上可知，本申请上述实施例一所提供的方案，通过动态执行待检测脚本文件，监测待检测脚本文件的待检测函数的输入参数，与基于预设函数及预设解释引擎的填充参数数据集相匹配，并不基于待检测脚本文件的特征值，而是从待检测脚本文件在动态执行过程中的输入参数入手，达到了确定待检测脚本文件是否为恶意脚本文件的目的，从而实现了增强网页服务器安全性的技术效果，进而解决了由于基于静态特征提取的恶意脚本文件检测方法容易漏报潜在的恶意脚本文件造成的网页服务器安全性较低的技术问题。

[0040] 可选地，如图3所示，在监测运行待检测脚本文件时所调用的待检测函数之前，方法还包括：

[0041] 步骤S302,挂钩预设函数及预设解释引擎,预设函数包括恶意脚本疑似函数以及用于变形输入参数的变形函数。

[0042] 本申请上述步骤S302中,执行待检测脚本文件,通过PHP扩展法挂钩预设函数以及变形函数,其中,预设函数可以包含恶意脚本疑似函数(例如Webshell疑似函数),如eval函数、assert函数、syste函数、exec函数、shell_exec函数、proc_open函数等;变形函数可以包含base64_decode函数、gzinflate函数、gzuncompress函数、zlib_decode函数、str_rot13函数和字符串相加(+.)函数等。

[0043] 进一步地,可以挂钩预设解释引擎(例如PHP解释引擎),如“==”或者“!=”。

[0044] 简单而言,挂钩预设函数及预设解释引擎就是修改待检测脚本文件原生方法的代码,以及修改待检测脚本文件原生代码的执行逻辑,其中,预设解释引擎就是解释待检测脚本文件语法的引擎,把待检测脚本文件转化为程序能识别的标识。

[0045] 可选地,挂钩预设函数及预设解释引擎包括:通过超文本预处理器PHP扩展法挂钩预设函数及预设解释引擎,其中,PHP扩展法用于修改待检测脚本文件的执行逻辑。

[0046] 步骤S304,通过预设解释引擎,检测是否需要输入用于页面交互的填充参数。

[0047] 本申请上述步骤S304中,web页面包含大量的交互,例如get函数、post函数、cookie函数等,当检测时如果没有输入数据会执行不下去。通过挂钩的预设函数的方法,可以探知待检测脚本文件什么时候要求填充数据,根据类型(具体可以是指get函数、post函数、cookie函数这三种需要客户端发送的数据类型)分别填充。其中,这个探知是根据预设解释引擎做的,当运行到需要外部参数的代码时,就知道了要填充数据。

[0048] 步骤S306,若需要输入填充参数,调用恶意脚本疑似函数或变形函数输入填充参数。

[0049] 本申请上述步骤S306中,输入参数为字符串,其中有大量的字符串变形函数,通过对变形函数的动态执行,加上静态分析。精准的定位该变形的输入结果。例如:

[0050] $a=&_POST[a];

[0051] $b=&_POST[b];

[0052] eval($a.$b);

[0053] 上述待检测脚本文件在执行的过程中,被填入填充参数,设para1,para2,对$a.$b的判断推导出para3.证实为可利用,为恶意脚本文件($a.$b可以是客户端可控的,都是参数输入结合起来的,对于脚本的开发者,$a,$b是人为的拆分参数,可以用一个参数来表示,这个过程是要确认这个$a.$b是不是属于刻意拆分而导致可以利用)。

[0054] 当第三句为eval($a.$b.'aa')时,因为待检测脚本文件加入了第三变量,推导不出来输入自符串;则不为恶意脚本文件。

[0055] 如果字符串‘aa’为一条完整PHP语句时,如“echo1;”结合语法分析,可以把“echo1”作为独立整句,不影响上下文,则eval($a.$b.'aa')可以推导出第三输入变量para3。该待检测脚本文件为恶意脚本文件。

[0056] 步骤S308,生成包含填充参数的填充参数数据集。

[0057] 由此可知,现有技术存在的基于静态特征提取的恶意脚本文件的检测方法容易出现漏报,导致网页服务器安全性较差的问题,本申请提出一种基于动态执行待检测脚本文件的方法,在执行待检测脚本文件的过程中,监测待检测脚本文件的待检测函数的输入参

数，与基于预设函数及预设解释引擎的填充参数数据集相匹配，从待检测脚本文件在动态执行过程中的输入参数入手，达到了确定待检测脚本文件是否为恶意脚本文件的目的，从而实现了增强网页服务器安全性的技术效果。

[0058] 需要说明的是，对于前述的各方法实施例，为了简单描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本申请并不受所描述的动作顺序的限制，因为依据本申请，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定是本申请所必须的。

[0059] 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根据上述实施例的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，本申请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)执行本申请各个实施例所述的方法。

[0060] 实施例2

[0061] 根据本申请实施例，还提供了一种用于实施上述方法实施例的装置实施例，本申请上述实施例所提供的装置可以在计算机终端上运行。

[0062] 图4是根据本申请实施例的恶意脚本文件的检测装置的结构示意图。

[0063] 如图4所示，该恶意脚本文件的检测装置可以包括监测单元502、判断单元504以及确定单元506。

[0064] 其中，监测单元502，用于在执行获取到的待检测脚本文件的过程中，监测运行所述待检测脚本文件时所调用的待检测函数；判断单元504，用于判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中，其中，所述填充参数数据集包括用于页面交互的填充参数，所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，所述预设函数用于输入所述填充参数，所述预设解释引擎用于检测是否需要输入所述填充参数；确定单元506，用于若所述输入参数包含在所述填充参数数据集中，确定所述待检测脚本文件为恶意脚本文件。

[0065] 由上可知，本申请上述实施例二所提供的方案，通过动态执行待检测脚本文件，监测待检测脚本文件的待检测函数的输入参数，与基于预设函数及预设解释引擎的填充参数数据集相匹配，并不基于待检测脚本文件的特征值，而是从待检测脚本文件在动态执行过程中的输入参数入手，达到了确定待检测脚本文件是否为恶意脚本文件的目的，从而实现了增强网页服务器安全性的技术效果，进而解决了由于基于静态特征提取的恶意脚本文件检测方法容易漏报潜在的恶意脚本文件造成的网页服务器安全性较低的技术问题。

[0066] 此处需要说明的是，上述监测单元502、判断单元504以及确定单元506对应于实施例一中的步骤S202至步骤S206，三个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中，可以通过软件实现，也可以通过硬件实现。

[0067] 可选地，如图5所示，恶意脚本文件的检测装置还包括：挂钩单元602、检测单元604、调用单元606以及生成单元608。

[0068] 其中，挂钩单元602，用于挂钩所述预设函数及所述预设解释引擎，所述预设函数包括恶意脚本疑似函数以及用于变形所述输入参数的变形函数；检测单元604，用于通过所述预设解释引擎，检测是否需要输入用于页面交互的所述填充参数；调用单元606，用于若需要输入所述填充参数，调用所述恶意脚本疑似函数或所述变形函数输入所述填充参数；生成单元608，用于生成包含所述填充参数的所述填充参数数据集。

[0069] 此处需要说明的是，上述挂钩单元602、检测单元604、调用单元606以及生成单元608对应于实施例一中的步骤S302至步骤S308，四个模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例一所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例一提供的计算机终端10中，可以通过软件实现，也可以通过硬件实现。

[0070] 可选地，所述挂钩单元602用于执行以下步骤挂钩所述预设函数及所述预设解释引擎：通过超文本预处理器PHP扩展法挂钩所述预设函数及所述预设解释引擎，其中，所述PHP扩展法用于修改所述待检测脚本文件的执行逻辑。

[0071] 可选地，恶意脚本疑似函数包括以下一种或几种：用于将字符串作为PHP语法执行的函数(例如eval函数)、用于判断条件是否正确的函数(例如assert函数)、用于调用执行系统命令的函数(例如syste函数、exec函数、shell_exec函数)以及用于进程执行的函数(例如proc_open函数)。

[0072] 可选地，变形函数包括以下一种或几种：用于编码解密的函数(例如base64_decode函数)、用于解压缩的函数(例如gzinflate函数、gzuncompress函数、zlib_decode函数)、用于字符串旋转解密的函数(例如str_rot13函数)以及(+.)函数。

[0073] 由此可知，现有技术存在的基于静态特征提取的恶意脚本文件的检测方法容易出现漏报，导致网页服务器安全性较差的问题，本申请提出一种基于动态执行待检测脚本文件的方法，在执行待检测脚本文件的过程中，监测待检测脚本文件的待检测函数的输入参数，与基于预设函数及预设解释引擎的填充参数数据集相匹配，从待检测脚本文件在动态执行过程中的输入参数入手，达到了确定待检测脚本文件是否为恶意脚本文件的目的，从而实现了增强网页服务器安全性的技术效果。

[0074] 实施例3

[0075] 本申请的实施例还提供了一种存储介质。可选地，在本实施例中，上述存储介质可以用于保存上述实施例一所提供的恶意脚本文件的检测方法所执行的程序代码。

[0076] 可选地，在本实施例中，上述存储介质可以位于计算机网络中计算机终端群中的任意一个计算机终端中，或者位于移动终端群中的任意一个移动终端中。

[0077] 可选地，在本实施例中，存储介质被设置为存储用于执行以下步骤的程序代码：在执行获取到的待检测脚本文件的过程中，监测运行所述待检测脚本文件时所调用的待检测函数；判断所述待检测函数的输入参数是否包含在预先生成的填充参数数据集中，其中，所述填充参数数据集包括用于页面交互的填充参数，所述填充参数为根据预先挂钩的预设函数及预设解释引擎生成的，所述预设函数用于输入所述填充参数，所述预设解释引擎用于检测是否需要输入所述填充参数；若所述输入参数包含在所述填充参数数据集中，确定所述待检测脚本文件为恶意脚本文件。

[0078] 可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：挂钩所述预设

函数及所述预设解释引擎，所述预设函数包括恶意脚本疑似函数以及用于变形所述输入参数的变形函数；通过所述预设解释引擎，检测是否需要输入用于页面交互的所述填充参数；若需要输入所述填充参数，调用所述恶意脚本疑似函数或所述变形函数输入所述填充参数；生成包含所述填充参数的所述填充参数数据集。

[0079] 可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：通过超文本预处理器PHP扩展法挂钩所述预设函数及所述预设解释引擎，其中，所述PHP扩展法用于修改所述待检测脚本文件的执行逻辑。

[0080] 可选地，在本实施例中，上述存储介质可以包括但不限于：U盘、只读存储器(ROM，Read-Only Memory)、随机存取存储器(RAM，Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0081] 可选地，本实施例中的具体示例可以参考上述实施例1中所描述的示例，本实施例在此不再赘述。

[0082] 上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

[0083] 在本申请的上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

[0084] 在本申请所提供的几个实施例中，应该理解到，所揭露的技术内容，可通过其它的方式实现。其中，以上所描述的装置实施例仅仅是示意性的，例如所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，单元或模块的间接耦合或通信连接，可以是电性或其它的形式。

[0085] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0086] 另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0087] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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中，包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U盘、只读存储器(ROM，Read-Only Memory)、随机存取存储器(RAM，Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0088] 以上所述仅是本申请的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本申请原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本申请的保护范围。

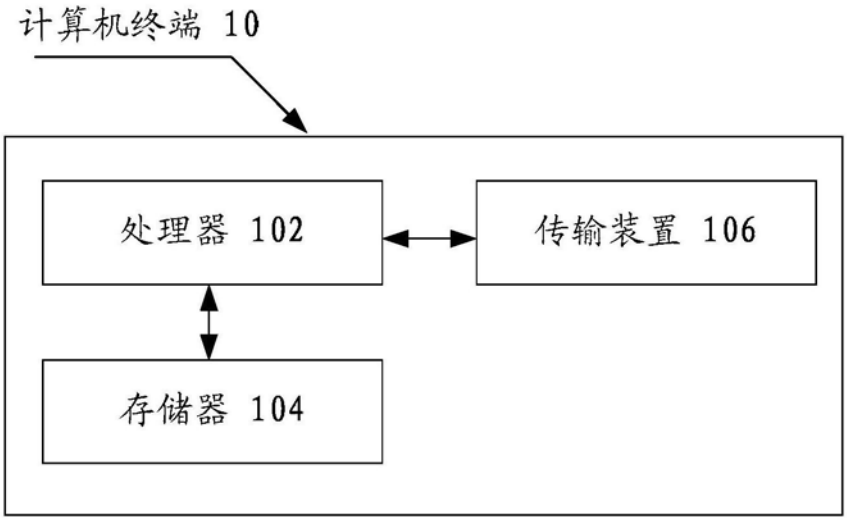

计算机终端 10
处理器 102
传输装置 106
存储器 104

图1

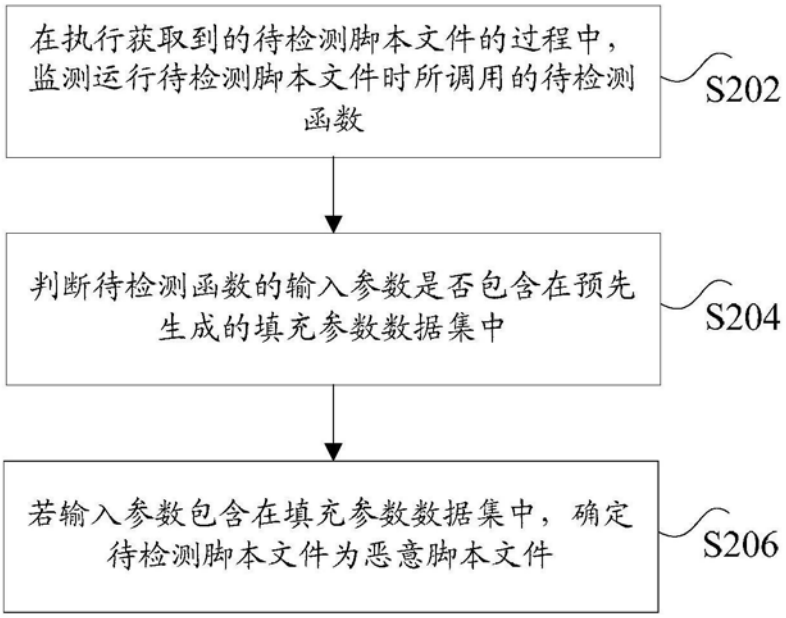

在执行获取到的待检测脚本文件的过程中，监测运行待检测脚本文件时所调用的待检测函数
S202
判断待检测函数的输入参数是否包含在预先生成的填充参数数据集中
S204
若输入参数包含在填充参数数据集中，确定待检测脚本文件为恶意脚本文件
S206

图2

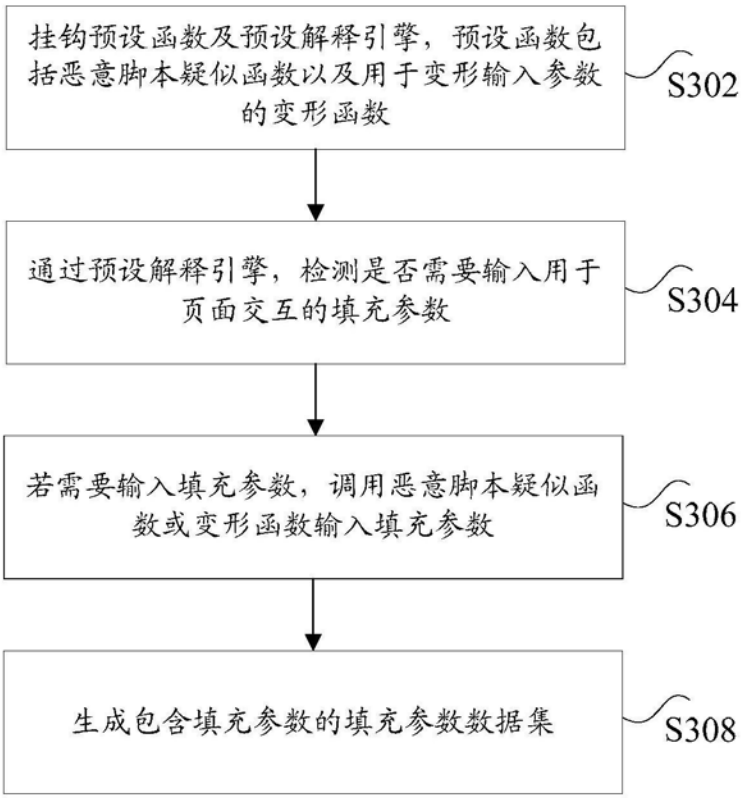

挂钩预设函数及预设解释引擎，预设函数包括恶意脚本疑似函数以及用于变形输入参数的变形函数
S302
通过预设解释引擎，检测是否需要输入用于页面交互的填充参数
S304
若需要输入填充参数，调用恶意脚本疑似函数或变形函数输入填充参数
S306
生成包含填充参数的填充参数数据集
S308

图3


监测单元502
判断单元504
确定单元506

图4

图5