(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 104331663 B
(45)授权公告日 2017.09.01

(21)申请号 201410602604.3

(22)申请日 2014.10.31

(65)同一申请的已公布的文献号
申请公布号 CN 104331663 A

(43)申请公布日 2015.02.04

(73)专利权人 北京奇虎科技有限公司
地址 100088 北京市西城区新街口外大街28号D座112室(德胜园区)
专利权人 奇智软件(北京)有限公司

(72)发明人 唐海 陈卓 邢超 杨康

(74)专利代理机构 北京鼎佳达知识产权代理事务所(普通合伙) 11348
代理人 王伟锋 刘铁生

(51)Int.Cl.
G06F 21/56(2013.01)
H04L 29/06(2006.01)

(56)对比文件
CN 103905422 A,2014.07.02,
CN 101673326 A,2010.03.17,
CN 103905422 A,2014.07.02,
KR 2009-0031393 A,2009.03.25,
CN 101692267 A,2010.04.07,
CN 102955913 A,2013.03.06,
CN 102955913 A,2013.03.06,
CN 102043919 A,2011.05.04,
CN 103559447 A,2014.02.05,
CN 102609649 A,2012.07.25,
李剑等.脚本引擎的简单实现.《电脑编程技巧与维护》.2009,

审查员 王慧敏

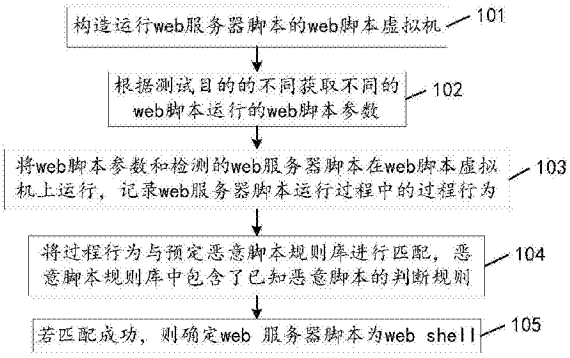
权利要求书2页 说明书9页 附图3页

(54)发明名称

web shell的检测方法以及web服务器

(57)摘要

本发明公开了一种web shell的检测方法以及web服务器,涉及信息安全领域,主要目的在于快速、精确地检测出web服务器中的web shell,从而能够保护web服务器的安全。本发明的主要技术方案为:构造运行web服务器脚本的web脚本虚拟机;根据测试目的的不同获取不同的web脚本运行的web脚本参数;将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行,记录web服务器脚本运行过程中的过程行为;将过程行为与预定恶意脚本规则库进行匹配;若匹配成功,则确定web服务器脚本为web shell。本发明主要应用于检测web shell的过程中。



CN 104331663 B

1.一种web shell检测方法,其特征在于,包括:

构造运行web服务器脚本的web脚本虚拟机,包括:构造web服务器脚本词法和语法分析器,所述词法和语法分析器用于对检测的web服务器脚本进行分析,得到语法树,所述语法树的根结点是运行web服务器脚本的程序入口;建立所述语法树的执行环境,所述执行环境至少包括:变量空间、内置对象以及内置函数;

根据测试目的的不同获取不同的web脚本运行的web脚本参数;

将所述web脚本参数和检测的web服务器脚本在所述web脚本虚拟机上运行,记录所述web服务器脚本运行过程中的过程行为,所述过程行为包括:调用的函数、调用函数的次序,调用的参数,参数处理的过程,参数处理的中间结果,变量的处理以及进行加法、减法运算;

将所述过程行为与预定恶意脚本规则库进行匹配,所述恶意脚本规则库中包含了已知恶意脚本的判断规则;

若匹配成功,则确定所述web服务器脚本为web shell。

2.根据权利要求1所述的方法,其特征在于,根据测试目的的不同获取不同的web脚本运行的web脚本参数包括:

根据测试的目的确定待获取参数的类型;

根据所述待获取参数的类型查询web脚本参数库,获取web脚本运行的web脚本参数,所述web脚本参数库存储有根据经验积累的黑客经常使用的web脚本参数,所述web脚本参数根据黑客入侵的类型进行分类。

3.根据权利要求2所述的方法,其特征在于,将所述web脚本参数和检测的web服务器脚本在所述web脚本虚拟机上运行,记录所述web服务器脚本运行过程中的过程行为包括:

依次从获取的web脚本参数中读取web脚本参数;

每次从语法树的根结点开始,逐一执行所述语法树中的每一条语句;

记录所述语法树执行过程中的过程行为。

4.根据权利要求1-3中任一项所述的方法,其特征在于,将所述过程行为与预设恶意脚本规则库进行匹配包括:

获取预定恶意脚本规则库中的规则;

利用所述规则对所述过程行为进行分析;

若得到分析结果,则确定检测的web服务器脚本为web shell。

5.一种web服务器,其特征在于,包括:

构造单元,用于构造运行web服务器脚本的web脚本虚拟机,包括:构造模块,用于构造web服务器脚本词法和语法分析器,所述词法和语法分析器用于对检测的web服务器脚本进行分析,得到web服务器脚本语法树,所述语法树的根结点是运行web服务器脚本的程序执行入口;建立模块,用于建立所述构造模块得到的语法树的执行环境,所述执行环境至少包括:变量空间、内置对象以及内置函数;

获取单元,用于根据测试目的的不同获取不同的web脚本运行的web脚本参数;

操作单元,用于将所述获取单元获取的web脚本参数和检测的web服务器脚本在所述构造单元构造的web脚本虚拟机上运行,记录所述web服务器脚本运行过程中的过程行为,所述过程行为包括:调用的函数、调用函数的次序,调用的参数,参数处理的过程,参数处理的中间结果,变量的处理以及进行加法、减法运算;

匹配单元，用于将所述操作单元记录的过程行为与预定恶意脚本规则库进行匹配，所述恶意脚本规则库中包含了已知恶意脚本的判断规则；

确定单元，用于当所述匹配单元匹配成功时，确定所述web服务器脚本为web shell。

6.根据权利要求5所述的web服务器，其特征在于，所述获取单元包括：

确定模块，用于根据测试的目的确定待获取参数的类型；

获取模块，用于根据所述确定模块确定的待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数，所述web脚本参数库存储有根据经验积累的黑客经常使用的web脚本参数，所述web脚本参数根据黑客入侵的类型进行分类。

7.根据权利要求6所述的web服务器，其特征在于，所述操作单元包括：

读取模块，用于依次从获取的web脚本参数中读取web脚本参数；

执行模块，用于每次从语法树的根结点开始，逐一执行所述语法树中的每一条语句；

记录模块，用于记录所述语法树在执行模块执行所述语法树中的每一条语句过程中的过程行为。

8.根据权利要求5-7中任一项所述的web服务器，其特征在于，所述匹配单元包括：

获取模块，用于获取预定恶意脚本规则库中的规则；

分析模块，用于利用所述获取模块获取的规则对所述过程行为进行分析；

确定模块，用于当分析模块得到分析结果时，确定检测的web服务器脚本为web shell。

web shell的检测方法以及web服务器

技术领域

[0001] 本发明涉及信息安全领域，特别是涉及一种web shell的检测方法以及web服务器。

背景技术

[0002] 随着互联网技术的高速发展，互联网信息安全成为人们日益关注的焦点。web shell是web入侵的脚本攻击工具。简单的说来，web shell就是一个动态服务器页面(Active Server Page，asp)或超级文本预处理语言(Hypertext Preprocessor，php)木马后门，黑客在入侵了一个网站后，常常在将这些asp或php木马后门文件放置在网站服务器的web目录中，与正常的网页文件混在一起。然后黑客就可以用web的方式，通过asp或php木马后门控制网站服务器，包括上传下载文件、查看数据库、执行任意程序命令等。

[0003] 目前，我们常常通过静态检测的方式检测是否存在web shell。该静态检测方式是通过将web服务器端的脚本与恶意特征码进行二进制匹配，若匹配成功，则确定该脚本为web shell。该种检测方法不能快速、精确地检测出web服务器中的web shell，增加了对web服务器端web shell的误判率，从而使web服务器的安全受到威胁。

发明内容

[0004] 有鉴于此，本发明实施例提供一种web shell的检测方法以及web服务器，主要目的在于快速、精确地检测出web服务器中的web shell，从而能够保护web服务器的安全。

[0005] 依据本发明一个方面，提供了一种web shell检测方法，包括：

[0006] 构造运行web服务器脚本的web脚本虚拟机；

[0007] 根据测试目的的不同获取不同的web脚本运行的web脚本参数；

[0008] 将所述web脚本参数和检测的web服务器脚本在所述web脚本虚拟机上运行，记录所述web服务器脚本运行过程中的过程行为；

[0009] 将所述过程行为与预定恶意脚本规则库进行匹配，所述恶意脚本规则库中包含了已知恶意脚本的判断规则；

[0010] 若匹配成功，则确定所述web服务器脚本为web shell。

[0011] 根据本发明的另一个方面，提供了一种web服务器，包括：

[0012] 构造单元，用于构造运行web服务器脚本的web脚本虚拟机；

[0013] 获取单元，用于根据测试目的的不同获取不同的web脚本运行的web脚本参数；

[0014] 操作单元，用于将所述获取单元获取的web脚本参数和检测的web服务器脚本在所述构造单元构造的web脚本虚拟机上运行，记录所述web服务器脚本运行过程中的过程行为；

[0015] 匹配单元，用于将所述操作单元记录的过程行为与预定恶意脚本规则库进行匹配，所述恶意脚本规则库中包含了已知恶意脚本的判断规则；

[0016] 确定单元，用于当所述匹配单元匹配成功时，确定所述web服务器脚本为web

shell。

[0017] 借由上述技术方案，本发明提供的web shell的检测方法以及web服务器，当对web shell进行检测的时候，先构造能运行web服务器脚本的web脚本虚拟机，web shell的检测是基于该web脚本虚拟机实现的；再将获取的web脚本参数和检测的web服务器脚本在该web脚本虚拟机上运行，并且记录web服务器脚本在运行过程中的过程行为，将记录下来的过程行为与预定恶意脚本规则库进行匹配，若能够匹配成功，那么该web服务器脚本为web shell，整个过程按照规则流程自动执行，与现有技术中通过静态检测的方式检测是否存在web shell相比快速、准确。

[0018] 上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

[0019] 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。在附图中：

[0020] 图1示出了本发明实施例提供的一种web shell检测方法的流程图；

[0021] 图2示出了本发明实施例提供的构建web脚本虚拟机的方法流程图；

[0022] 图3示出了本发明实施例提供的根据测试目的的不同获取不同的web脚本运行的web脚本参数的流程图；

[0023] 图4示出了本发明实施例提供的一种操作web脚本虚拟机的方法流程图；

[0024] 图5示出了本发明实施例提供的另一种web shell检测方法的流程图；

[0025] 图6示出了本发明实施例提供的一种web服务器的组成框图；

[0026] 图7示出了本发明实施例提供的另一种web服务器的组成框图；

[0027] 图8示出了本发明实施例提供的另一种web服务器的组成框图；

[0028] 图9示出了本发明实施例提供的另一种web服务器的组成框图；

[0029] 图10示出了本发明实施例提供的另一种web服务器的组成框图。

具体实施方式

[0030] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

[0031] 本发明实施例提供的一种web shell检测方法，如图1所示，该方法包括：

[0032] 101、构造运行web服务器脚本的web脚本虚拟机。

[0033] web shell检测是在web服务器端进行的，其检测的目的是防止web服务器在运行的时候存在web shell。在设置检测方法时，一般不能以正常运行的web服务器以及web服务器脚本为测试用例，需要根据构建一个与正常运行的web服务器脚本一样的运行环境，并在此基础上进行训练测试。这样就需要构造运行web服务器脚本的web脚本虚拟机。该虚拟机

的构造包括web服务器脚本运行环境的构造以及读写和解析web服务器脚本模块的构造等。

[0034] 102、根据测试目的的不同获取不同的web脚本运行的web脚本参数。

[0035] web脚本参数存储在已知的web脚本参数库中，该已知的web脚本参数库记录了黑客入侵常使用的脚本参数，并根据黑客入侵的类型进行分类，其中，每个类型包含多种web脚本参数。

[0036] 在根据测试目的的不同获取不同的web脚本运行的web脚本参数时，可以先根据测试的目的确定待获取参数的类型，之后再根据所述待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数。

[0037] 在根据所述待获取参数的类型查询web脚本参数库时，可以根据测试的目的遍历并获取已知web脚本参数库中该测试目的对应的web脚本参数，在一定程度上相对于人为的输入测试参数，使得获取的测试参数更加全面和准确，从而增加了检测web shell的准确性，能够保护web服务器的安全。

[0038] 103、将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行，记录web服务器脚本运行过程中的过程行为。

[0039] 在对web服务器端进行检测web shell时，没有用户访问web服务器，web脚本虚拟机无法获取所需的web脚本参数，因此需要将已知web脚本参数库中的web脚本参数引入。

[0040] 在将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行时，记录其运行的整个过程行为，该整个过程行为包括调用的函数、调用函数的次序，调用的参数，参数处理的过程，参数处理的中间结果，变量的处理以及进行加法、减法运算等等，具体的对运行过程中的实际操作不进行限定。

[0041] 104、将过程行为与预定恶意脚本规则库进行匹配，恶意脚本规则库中包含了已知恶意脚本的判断规则。

[0042] 预定恶意脚本规则库为已知的预定恶意脚本规则库，该恶意脚本规则库记录了黑客攻击web服务器时常用的规则，该恶意脚本规则库中的内容根据经验设定。将过程行为与预定恶意脚本规则库进行匹配，首先根据过程行为依次遍历该预定恶意规则库中的规则，若过程行为能匹配该预定恶意规则库中的其中一条规则，则执行105，若匹配不成功，则检测的web服务器脚本为无恶意脚本。

[0043] 105、若匹配成功，则确定web服务器脚本为web shell。

[0044] 本发明实施例中，当对web shell进行检测的时候，先构造能运行web服务器脚本的web脚本虚拟机，web shell的检测是基于该web脚本虚拟机实现的；再将获取的web脚本参数和检测的web服务器脚本在该web脚本虚拟机上运行，并且记录web服务器脚本在运行过程中的过程行为，将记录下来的过程行为与预定恶意脚本规则库进行匹配，若能够匹配成功，那么该web服务器脚本为web shell，整个过程按照规则流程自动执行，与现有技术中通过静态检测的方式检测是否存在web shell相比快速、准确。

[0045] 基于上述方法，虚拟机的构造包括web服务器脚本运行环境的构造以及读写和解析web服务器脚本模块的构造等。在构造web脚本虚拟机时可以通过但不局限于以下的方法实现，如图2所示，构建web脚本虚拟机的方法包括：

[0046] 201、构造web服务器脚本词法和语法分析器，所述词法和语法分析器用于对检测的web服务器脚本进行分析，得到语法树，语法树的根结点是运行web服务器脚本的程序入

口。

[0047] 由于web服务器脚本都有各自的语法，例如，变量名的写入，变量之间进行加、减、乘、除运算以及函数之间的调用等，因此在对检测的web服务器脚本进行分析时，可以通过词法和语法分析器对检测的web服务器脚本进行分析，得到语法树；也可以通过链表实现对检测的web服务器脚本的分析，具体的实现方式，本发明实施例不进行限制。

[0048] 本发明实施例以通过词法和语法分析器对检测的web服务器脚本进行分析，得到语法树为例进行具体的阐述。得到的语法树的每一个结点为一条语句，按照分析出web服务器脚本的内容，可将该语法树分为一个或多个左子树以及一个或多个右子树，也可以将该语法树分为二叉树或者完全二叉树，具体的，本发明实施例对此不进行限制。

[0049] 202、建立语法树的执行环境，执行环境至少包括：变量空间、内置对象以及内置函数。

[0050] 在通过词法和语法分析器对检测的web服务器脚本进行分析，得到语法树之后，需要建立语法树的执行环境。该执行环境为可变更的执行环境，其包含的内容可根据用户的需求自行变更，具体的执行环境中包含的内容，本发明实施例对此不进行限制。

[0051] 其中，变量空间为web服务器脚本变量初始化所需要的存储空间，关于该存储空间的大小可根据在实际操作中所需要的存储空间进行确定，具体的本发明实施例不进行限制；内置对象至少包括环境变量对象，通过环境变量对象可以得到当前运行的目录、当前运行的配置等信息；内置函数至少包括字符串搜索、加法、减法、乘法、除法等函数。其中，内置对象和内置函数不仅仅包括上述的内容，上述内容为示例性的举例，具体包含的内容，本发明实施例不进行限制。

[0052] 进一步的，基于web脚本虚拟机实现对web shell的检测，具体的在执行102根据测试目的的不同获取不同的web脚本运行的web脚本参数时，本发明实施例可以采用但不局限于以下的方法实现，该方法如图3所示，包括：

[0053] 301、根据测试的目的确定待获取参数的类型。

[0054] web脚本参数的类型至少包括：注入测试、信息泄露、提权等。关于web脚本参数的具体类型本发明实施例对此不进行限制。根据测试的目的确定待获取参数的类型，例如，当检测web服务器脚本是否为关于信息泄露类型的web shell时，可以先遍历web脚本参数库中关于信息泄露类型的web脚本参数获取该类型的web脚本参数之后，在进行后续的检测。

[0055] 302、根据待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数。

[0056] 进一步的，在获取web脚本运行的web脚本参数之后，将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行，正如前面所述的，基于web脚本虚拟机，并在web脚本虚拟机上运行该web服务器脚本实现对web shell的检测，同时记录web服务器脚本运行过程中的过程行为。具体的，本发明实施例提供一种操作web脚本虚拟机的方法：如图4所示，该方法包括：

[0057] 401、依次从获取的web脚本参数中读取web脚本参数。

[0058] 黑客进行攻击web服务器时的类型可以有多种web脚本参数，在对web服务器脚本进行检测时，我们需要依次读取根据测试的目的确定待获取参数的类型中包含的所有可用的web脚本参数，再逐一的将获取的所有可用的web脚本参数进行后续检测。

[0059] 402、每次从语法树的根结点开始，逐一执行语法树中的每一条语句。

[0060] 在进行检测web服务器脚本是否为web shell时，将步骤401获取的待获取参数的类型中包含的所有可用的web脚本参数分别进行检测，例如，将获取的待获取参数的类型中包含的所有可用的web脚本参数中的第一个参数输入到检测的web服务器脚本中，执行该web服务器脚本对应的语法树，从该语法树的根结点开始运行，直到遍历完语法树中所有的结点；在将获取的待获取参数的类型中包含的所有可用的web脚本参数中的第二个参数输入到检测的web服务器脚本中，执行该web服务器脚本对应的语法树，从该语法树的根结点开始运行，直到遍历完语法树中所有的结点，直到用获取的待获取参数的类型中包含的所有可用的web脚本参数遍历完语法树中所有的结点，确保获取的待获取参数的类型中包含的所有可用的web脚本参数都能被运行，从而提高了检测web shell的准确性。

[0061] 其中，逐一执行语法树中的每一条语句，即逐一执行语法树中的每一个结点。在执行语法树中的每一个结点时，可以通过前序遍历、后序遍历，也可以通过中序遍历。具体的本发明实施例对此不进行限制。

[0062] 403、记录语法树执行过程中的过程行为。

[0063] 通过web虚拟机中调用的内置函数记录读取的web脚本参数在运行的整个过程行为，例如，在进行sql注入测试时，脚本参数为sql，在web虚拟机中运行通过词法和语法分析器对web脚本参数sql分析得到的语法树，调用web虚拟机中的内置函数，通过该内置函数传入字符串，该字符串为用户输入的脚本参数；语法树执行过程中的过程行为，即在内置函数中所调用的参数，例如，在进行测试时调用用户输入的参数abc，同时内置函数记录该调用的参数，若调用的参数被改动过，且对web服务器进行攻击，判断为web shell；若调用的字符串被改动过，但没有任何恶意行为，判断为正常脚本。

[0064] 其中，读取的web脚本参数在进行运行之前，通过一个内置函数对该读取的web脚本参数进行安全处理，例如将/以及%等加入到读取的web脚本参数中；在将安全处理后的web脚本参数运行时，将该脚本参数通过web虚拟机的内置函数进行解密。

[0065] 进一步的，在执行步骤104将过程行为与预定恶意脚本规则库进行匹配时，本发明实施例还提供一种web shell检测方法，如图5所示，该方法包括：

[0066] 501、获取预定恶意脚本规则库中的规则。

[0067] 该预定恶意脚本规则库为如前所述的预定恶意脚本规则库，关于预定恶意脚本规则库的相关描述，此处将不再对此进行赘述。

[0068] 502、利用规则对过程行为进行分析。

[0069] web虚拟机中的内置函数获取调用的参数后，再去获取预定恶意规则库，根据输入的参数和输出的参数的不同，对预定恶意规则库依次进行遍历，如果匹配了预定恶意脚本规则库其中的一条规则，则执行503。

[0070] 503、若得到分析结果，则确定检测的web服务器脚本为web shell。

[0071] 基于上述方法实施例，本发明实施例提供一种web服务器，如图6所示，web服务器包括：

[0072] 构造单元61，用于构造运行web服务器脚本的web脚本虚拟机；

[0073] 获取单元62，用于根据测试目的的不同获取不同的web脚本运行的web脚本参数；web脚本参数存储在已知的web脚本参数库中，该已知的web脚本参数库记录了黑客入侵常

使用的脚本参数，并根据黑客入侵的类型进行分类，其中，每个类型包含多种web脚本参数。在根据测试目的的不同获取不同的web脚本运行的web脚本参数时，可以先根据测试的目的确定待获取参数的类型，之后再根据所述待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数。在根据所述待获取参数的类型查询web脚本参数库时，可以根据测试的目的遍历并获取已知web脚本参数库中该测试目的对应的web脚本参数，在一定程度上相对于认为的输入测试参数，使得获取的测试参数更加全面和准确，从而增加了检测web shell的准确性，能够保护web服务器的安全。

[0074] 操作单元63，用于将获取单元62获取的web脚本参数和检测的web服务器脚本在构造单元61构造的web脚本虚拟机上运行，记录web服务器脚本运行过程中的过程行为。其中，在将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行时，记录其运行的整个过程行为，该整个过程行为包括调用的函数、调用函数的次序，调用的参数，参数处理的过程，参数处理的中间结果，变量的处理以及进行加法、减法运算等等，具体的对运行过程中的实际操作不进行限定。

[0075] 匹配单元64，用于将操作单元63记录的过程行为与预定恶意脚本规则库进行匹配，恶意脚本规则库中包含了已知恶意脚本的判断规则。预定恶意脚本规则库为已知的预定恶意脚本规则库，该恶意脚本规则库记录了黑客攻击web服务器时常用的规则，该恶意脚本规则库中的内容根据经验设定。

[0076] 确定单元65，用于当匹配单元64匹配成功时，确定web服务器脚本为web shell。

[0077] 进一步的，如图7所示，构造单元61包括：

[0078] 构造模块611，用于构造web服务器脚本词法和语法分析器，所述词法和语法分析器用于对检测的web服务器脚本进行分析，得到语法树，语法树的根结点是运行web服务器脚本的程序执行入口；其中，由于web服务器脚本都有各自的语法，例如，变量名的写入，变量之间进行加、减、乘、除运算以及函数之间的调用等，因此在对检测的web服务器脚本进行分析时，可以通过词法和语法分析器对检测的web服务器脚本进行分析，得到语法树；也可以通过链表实现对检测的web服务器脚本的分析，具体的实现方式，本发明实施例不进行限制。

[0079] 建立模块612，用于建立构造模块611得到的语法树的执行环境，执行环境至少包括：变量空间、内置对象以及内置函数。其中，变量空间为web服务器脚本变量初始化所需要的存储空间，关于该存储空间的大小可根据在实际操作中所需要的存储空间进行确定，具体的本发明实施例不进行限制；内置对象至少包括环境变量对象，通过环境变量对象可以得到当前运行的目录、当前运行的配置等信息；内置函数至少包括字符串搜索、加法、减法、乘法、除法等函数。其中，内置对象和内置函数不仅仅包括上述的内容，上述内容为示例性的举例，具体包含的内容，本发明实施例不进行限制。

[0080] 进一步的，如图8所示，获取单元62包括：

[0081] 确定模块621，用于根据测试的目的确定待获取参数的类型；

[0082] 获取模块622，用于根据确定模块621确定的待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数，web脚本参数库存储有根据经验积累的黑客经常使用的web脚本参数，web脚本参数根据黑客入侵的类型进行分类。

[0083] 进一步的，如图9所示，操作单元63包括：

[0084] 读取模块631，用于依次从获取的web脚本参数中读取web脚本参数；黑客进行攻击web服务器时的类型可以有多种web脚本参数，在对web服务器脚本进行检测时，我们需要依次读取根据测试的目的确定待获取参数的类型中包含的所有可用的web脚本参数，再逐一的将获取的所有可用的web脚本参数进行后续检测。

[0085] 执行模块632，用于每次从语法树的根结点开始，逐一执行语法树中的每一条语句；在进行检测web服务器脚本是否为web shell时，将读取模块631获取的待获取参数的类型中包含的所有可用的web脚本参数分别进行检测，例如，将获取的待获取参数的类型中包含的所有可用的web脚本参数中的第一个参数输入到检测的web服务器脚本中，执行该web服务器脚本对应的语法树，从该语法树的根结点开始运行，直到遍历完语法树中所有的结点；在将获取的待获取参数的类型中包含的所有可用的web脚本参数中的第二个参数输入到检测的web服务器脚本中，执行该web服务器脚本对应的语法树，从该语法树的根结点开始运行，直到遍历完语法树中所有的结点，直到用获取的待获取参数的类型中包含的所有可用的web脚本参数遍历完语法树中所有的结点，确保获取的待获取参数的类型中包含的所有可用的web脚本参数都能被运行，从而提高了检测web shell的准确性。

[0086] 记录模块633，用于记录语法树在执行模块632执行语法树中的每一条语句过程中的过程行为。

[0087] 进一步的，如图10所示，匹配单元64包括：

[0088] 获取模块641，用于获取预定恶意脚本规则库中的规则；

[0089] 分析模块642，用于利用获取模块641获取的规则对过程行为进行分析；

[0090] 确定模块643，用于当分析模块642得到分析结果时，确定检测的web服务器脚本为web shell。

[0091] 本发明提供的一种web shell的检测方法以及web服务器，当对web shell进行检测的时候，先构造能运行web服务器脚本的web脚本虚拟机，webshell的检测是基于该web脚本虚拟机实现的；再将获取的web脚本参数和检测的web服务器脚本在该web脚本虚拟机上运行，并且记录web服务器脚本在运行过程中的过程行为，将记录下来的过程行为与预定恶意脚本规则库进行匹配，若能够匹配成功，那么该web服务器脚本为web shell，整个过程按照规则流程自动执行，与现有技术中通过静态检测的方式检测是否存在web shell相比快速、准确。

[0092] 此外，在根据所述待获取参数的类型查询web脚本参数库时，可以根据测试的目的遍历并获取已知web脚本参数库中该测试目的对应的web脚本参数，在一定程度上相对于人为的输入测试参数，使得获取的测试参数更加全面和准确，从而增加了检测web shell的准确性，能够保护web服务器的安全。

[0093] 进一步的，每次从语法树的根结点开始，逐一执行语法树中的每一条语句，直到用获取的待获取参数的类型中包含的所有可用的web脚本参数遍历完语法树中所有的结点，确保获取的待获取参数的类型中包含的所有可用的web脚本参数都能被运行，从而提高了检测web shell的准确性。

[0094] 在上述实施例中，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。

[0095] 可以理解的是，上述方法及装置中的相关特征可以相互参考。另外，上述实施例中

的“第一”、“第二”等是用于区分各实施例，而并不代表各实施例的优劣。

[0096] 所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

[0097] 在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述，构造这类系统所要求的结构是显而易见的。此外，本发明也不针对任何特定编程语言。应当明白，可以利用各种编程语言实现在此描述的本发明的内容，并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0098] 在此处所提供的说明书中，说明了大量具体细节。然而，能够理解，本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

[0099] 类似地，应当理解，为了精简本公开并帮助理解各个发明方面中的一个或多个，在上面对本发明的示例性实施例的描述中，本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而，并不应将该公开的方法解释成反映如下意图：即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说，如下面的权利要求书所反映的那样，发明方面在于少于前面公开的单个实施例的所有特征。因此，遵循具体实施方式的权利要求书由此明确地并入该具体实施方式，其中每个权利要求本身都作为本发明的单独实施例。

[0100] 本领域那些技术人员可以理解，可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件，以及此外可以把它们分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外，可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述，本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0101] 此外，本领域的技术人员能够理解，尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征，但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如，在下面的权利要求书中，所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0102] 本发明的各个部件实施例可以以硬件实现，或者以在一个或者多个处理器上运行的软件模块实现，或者以它们的组合实现。本领域的技术人员应当理解，可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的web shell的检测方法以及web服务器中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如，计算机程序和计算机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上，或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到，或者在载体信号上提供，或者以任何其他形式提供。

[0103] 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。

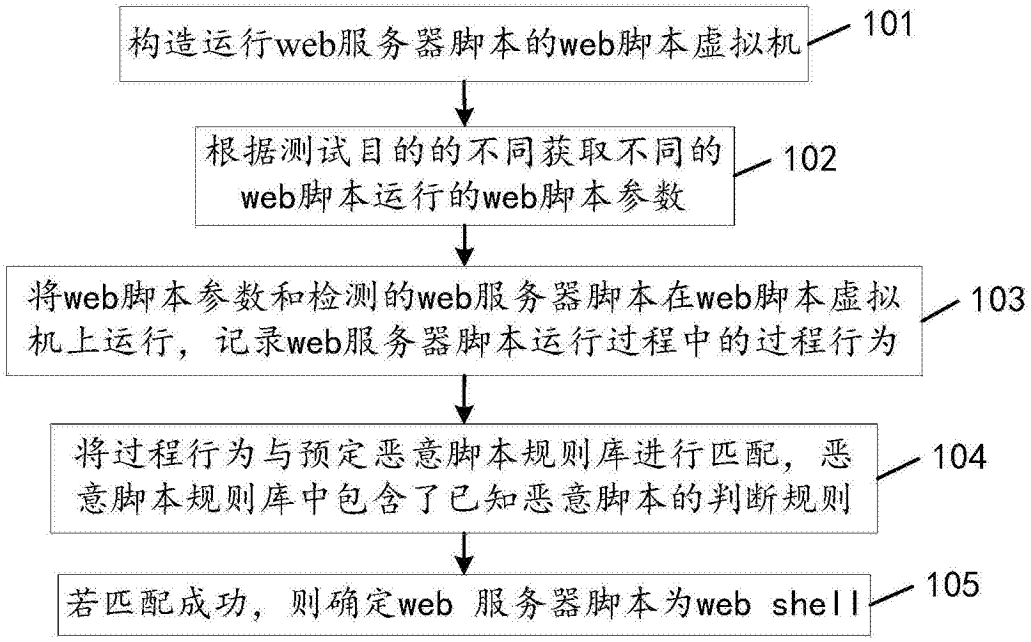

构造运行web服务器脚本的web脚本虚拟机
101
根据测试目的的不同获取不同的web脚本运行的web脚本参数
102
将web脚本参数和检测的web服务器脚本在web脚本虚拟机上运行，记录web服务器脚本运行过程中的过程行为
103
将过程行为与预定恶意脚本规则库进行匹配，恶意脚本规则库中包含了已知恶意脚本的判断规则
104
若匹配成功，则确定web 服务器脚本为web shell
105

图1

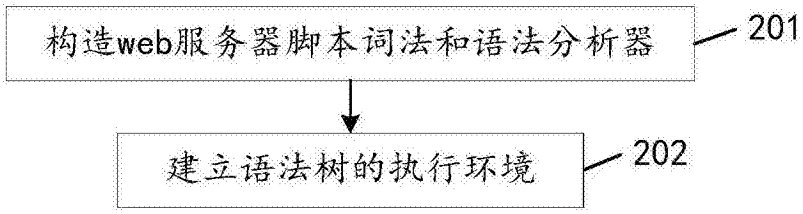

构造web服务器脚本词法和语法分析器
201
建立语法树的执行环境
202

图2

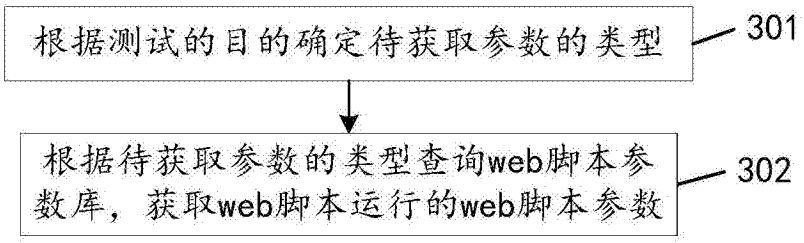

根据测试的目的确定待获取参数的类型
301
根据待获取参数的类型查询web脚本参数库，获取web脚本运行的web脚本参数
302

图3

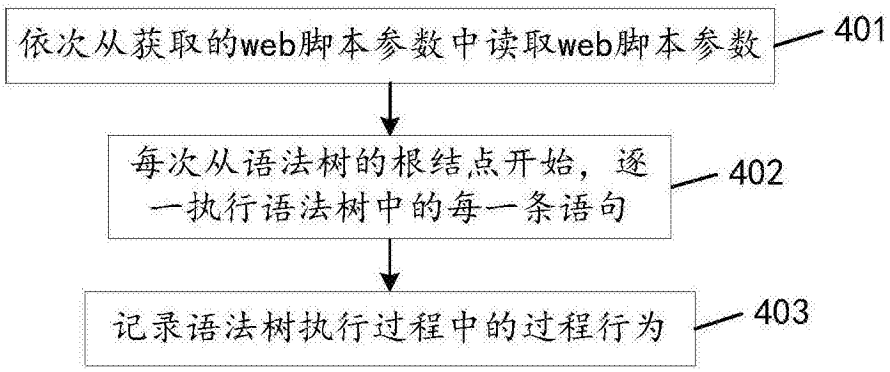

依次从获取的web脚本参数中读取web脚本参数
401
每次从语法树的根结点开始，逐一执行语法树中的每一条语句
402
记录语法树执行过程中的过程行为
403

图4

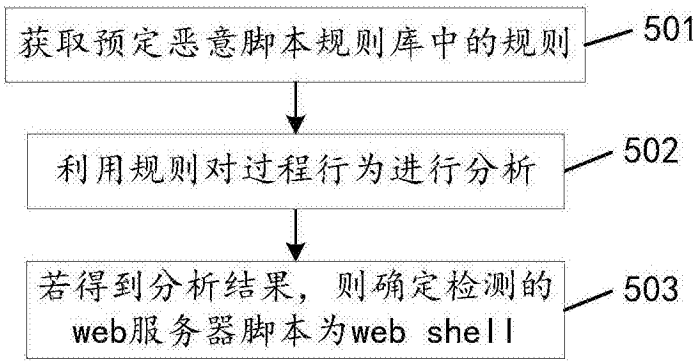

获取预定恶意脚本规则库中的规则
501
利用规则对过程行为进行分析
502
若得到分析结果，则确定检测的web服务器脚本为web shell
503

图5

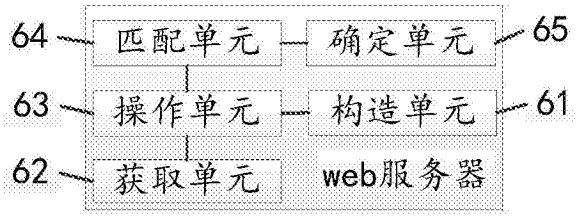

64
匹配单元
确定单元
65
63
操作单元
构造单元
61
62
获取单元
web服务器

图6

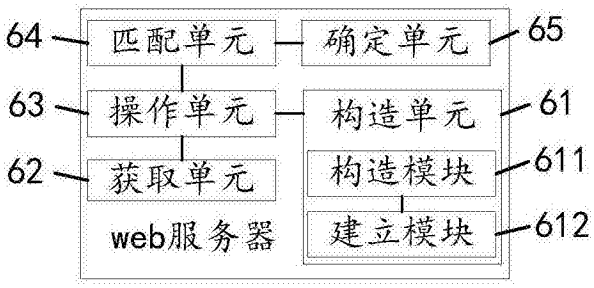

64
匹配单元
确定单元
65
63
操作单元
构造单元
61
62
获取单元
构造模块
611
web服务器
建立模块
612

图7

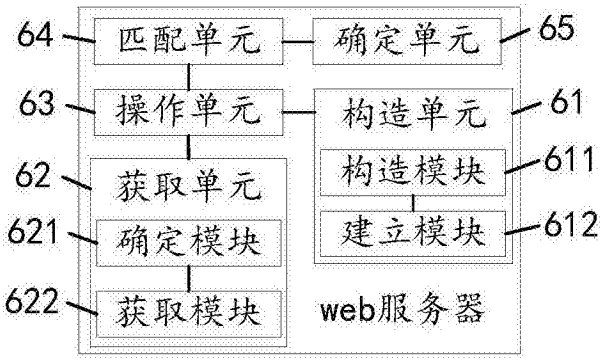

64
匹配单元
确定单元
65
63
操作单元
构造单元
61
62
获取单元
构造模块
611
621
确定模块
建立模块
612
622
获取模块
web服务器

图8

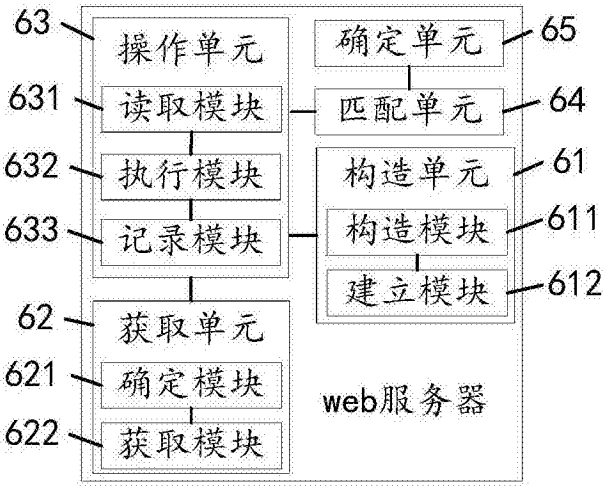

63
操作单元
确定单元
65
631
读取模块
匹配单元
64
632
执行模块
构造单元
61
633
记录模块
构造模块
611
建立模块
612
62
获取单元
621
确定模块
web服务器
622
获取模块

图9

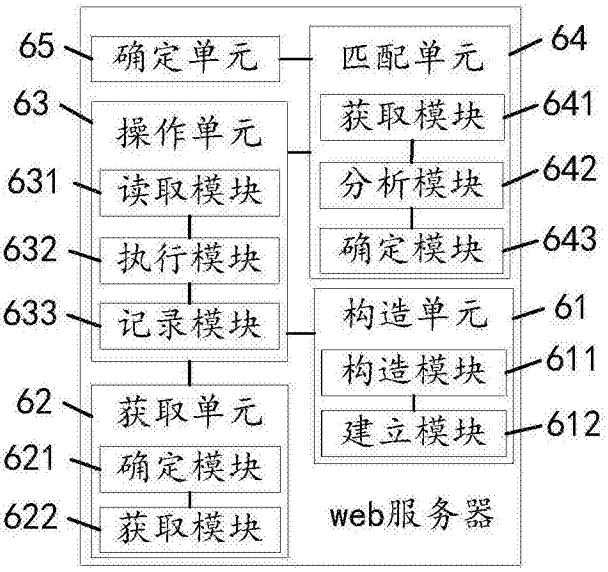

65
确定单元
匹配单元
64
63
操作单元
获取模块
641
分析模块
642
631
读取模块
确定模块
643
632
执行模块
构造单元
61
633
记录模块
构造模块
611
建立模块
612
62
获取单元
621
确定模块
web服务器
622
获取模块

图10