(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 112422581 A
(43)申请公布日 2021.02.26

(21)申请号 202011377833.1

(22)申请日 2020.11.30

(71)申请人 杭州安恒信息技术股份有限公司
地址 310000 浙江省杭州市滨江区西兴街道联慧街188号

(72)发明人 钱仕鹏 范渊 黄进

(74)专利代理机构 北京集佳知识产权代理有限公司 11227
代理人 张春辉

(51)Int.Cl.
H04L 29/06(2006.01)
G06F 8/30(2018.01)
G06F 8/53(2018.01)

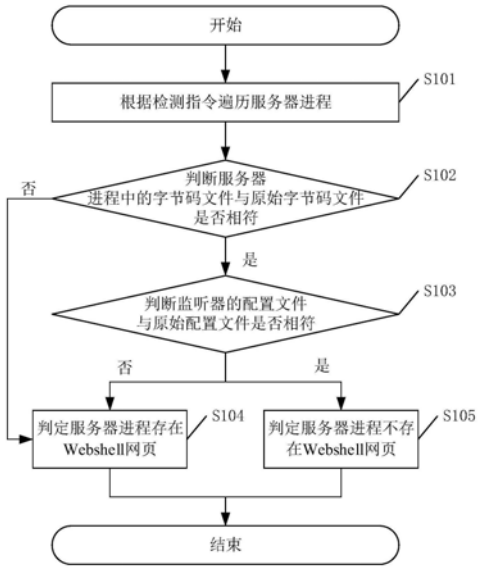
权利要求书2页 说明书8页 附图3页

(54)发明名称

JVM中的Webshell网页检测方法、装置及设备

(57)摘要

本申请公开了一种JVM中的Webshell网页检测方法，包括根据检测指令遍历服务器进程；判断服务器进程中的字节码文件与原始字节码文件是否相符；若字节码文件与原始字节码文件不相符，则判定服务器进程存在Webshell网页；若字节码文件与原始字节码文件相符，则判断监听器的配置文件与原始配置文件是否相符；若配置文件与原始配置文件不相符，则判定服务器进程不存在Webshell网页；该JVM中的Webshell网页检测方法可以实现服务器中Webshell的检测，保证服务器的安全运行。本申请还公开了一种JVM中的Webshell网页检测装置、设备及计算机可读存储介质，均具有上述有益效果。



CN 112422581 A

1.一种JVM中的Webshell网页检测方法,其特征在于,所述方法包括:

根据检测指令遍历服务器进程;

判断所述服务器进程中的字节码文件与原始字节码文件是否相符;

若所述字节码文件与所述原始字节码文件不相符,则判定所述服务器进程存在Webshell网页;

若所述字节码文件与所述原始字节码文件相符,则判断监听器的配置文件与原始配置文件是否相符;

若所述配置文件与所述原始配置文件不相符,则判定所述服务器进程不存在所述Webshell网页。

2.根据权利要求1所述的方法,其特征在于,还包括:

当所述字节码文件与所述原始字节码文件不相符时,将所述字节码文件转化为java代码,并定位所述java代码中的Webshell网页。

3.根据权利要求2所述的方法,其特征在于,所述将所述字节码文件转化为java代码,包括:

通过反编译技术将所述字节码文件转化为所述java代码。

4.根据权利要求2所述的方法,其特征在于,所述定位所述java代码中的Webshell网页,包括:

通过正则匹配技术定位所述java代码中的Webshell网页。

5.根据权利要求1至4任意一项所述的方法,其特征在于,还包括:

当所述字节码文件与所述原始字节码文件不相符时,判断服务器是否处于业务处理状态;

若否,则重启服务器;

若是,则利用所述原始字节码文件替换所述字节码文件。

6.根据权利要求1所述的方法,其特征在于,还包括:

当所述配置文件与所述原始配置文件不相符时,判断服务器是否处于业务处理状态;

若否,则重启服务器;

若是,则删除所述配置文件。

7.根据权利要求1所述的方法,其特征在于,还包括:

根据所述检测指令执行日志记录操作,获得Webshell网页检测日志。

8.一种JVM中的Webshell网页检测装置,其特征在于,包括:

进程遍历模块,用于根据检测指令遍历服务器进程;

第一文件判断模块,用于判断所述服务器进程中的字节码文件与原始字节码文件是否相符;

第一结果判定模块,用于若所述字节码文件与所述原始字节码文件不相符,则判定所述服务器进程存在Webshell网页;

第二文件判断模块,用于若所述字节码文件与所述原始字节码文件相符,则判断监听器的配置文件与原始配置文件是否相符;

第二结果判定模块,用于若所述配置文件与所述原始配置文件不相符,则判定所述服务器进程存在所述Webshell网页。

9.一种JVM中的Webshell网页检测设备,其特征在于,包括:

存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序时实现如权利要求1至7任一项所述的JVM中的Webshell网页检测方法的步骤。

10.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1至7任一项所述的JVM中的Webshell网页检测方法的步骤。

JVM中的Webshell网页检测方法、装置及设备

技术领域

[0001] 本申请涉及互联网安全技术领域，特别涉及一种JVM中的Webshell网页检测方法，还涉及一种JVM中的Webshell网页检测装置、设备及计算机可读存储介质。

背景技术

[0002] 随着互联网的飞速发展，网络安全逐渐成为一个潜在的巨大问题，网络安全是一个涉及面很广泛的问题，其中也会涉及到是否构成犯罪行为的问题。现如今的服务器安全尤为重要，信息泄露、挖矿病毒等字眼逐渐进入人们的视线当中，Webshell的出现更是为恶意攻击者带来了巨大的便利。Webshell是一种以网页文件形式存在的恶意网页后门文件，其可以通过很小的文件、很少的流量实现对服务器的完全控制。然而，现有的针对Webshell的检测工具都只是对相关文件的检测，对服务器内存Webshell的检测并不支持，但服务器内存Webshell的产生对服务器安全同样具有很大威胁，无文件落地的特征使其隐蔽性更好，导致服务器安全性能无法得到保证。

[0003] 因此，如何实现服务器中Webshell的检测，保证服务器的安全运行是本领域技术人员亟待解决的问题。

发明内容

[0004] 本申请的目的是提供一种JVM中的Webshell网页检测方法，该JVM中的Webshell网页检测方法可以实现服务器中Webshell的检测，保证服务器的安全运行；本申请的另一目的是提供一种JVM中的Webshell网页检测装置、设备及计算机可读存储介质，均具有上述有益效果。

[0005] 第一方面，本申请提供了一种JVM中的Webshell网页检测方法，包括：

[0006] 根据检测指令遍历服务器进程；

[0007] 判断所述服务器进程中的字节码文件与原始字节码文件是否相符；

[0008] 若所述字节码文件与所述原始字节码文件不相符，则判定所述服务器进程存在Webshell网页；

[0009] 若所述字节码文件与所述原始字节码文件相符，则判断监听器的配置文件与原始配置文件是否相符；

[0010] 若所述配置文件与所述原始配置文件不相符，则判定所述服务器进程不存在所述Webshell网页。

[0011] 优选的，所述JVM中的Webshell网页检测方法还包括：

[0012] 当所述字节码文件与所述原始字节码文件不相符时，将所述字节码文件转化为java代码，并定位所述java代码中的Webshell网页。

[0013] 优选的，所述将所述字节码文件转化为java代码，包括：

[0014] 通过反编译技术将所述字节码文件转化为所述java代码。

[0015] 优选的，所述定位所述java代码中的Webshell网页，包括：

[0016] 通过正则匹配技术定位所述java代码中的Webshell网页。

[0017] 优选的，所述JVM中的Webshell网页检测方法还包括：

[0018] 当所述字节码文件与所述原始字节码文件不相符时，判断服务器是否处于业务处理状态；

[0019] 若否，则重启服务器；

[0020] 若是，则利用所述原始字节码文件替换所述字节码文件。

[0021] 优选的，所述JVM中的Webshell网页检测方法还包括：

[0022] 当所述配置文件与所述原始配置文件不相符时，判断服务器是否处于业务处理状态；

[0023] 若否，则重启服务器；

[0024] 若是，则删除所述配置文件。

[0025] 优选的，所述JVM中的Webshell网页检测方法还包括：

[0026] 根据所述检测指令执行日志记录操作，获得Webshell网页检测日志。

[0027] 第二方面，本申请还公开了一种JVM中的Webshell网页检测装置，包括：

[0028] 进程遍历模块，用于根据检测指令遍历服务器进程；

[0029] 第一文件判断模块，用于判断所述服务器进程中的字节码文件与原始字节码文件是否相符；

[0030] 第一结果判定模块，用于若所述字节码文件与所述原始字节码文件不相符，则判定所述服务器进程存在Webshell网页；

[0031] 第二文件判断模块，用于若所述字节码文件与所述原始字节码文件相符，则判断监听器的配置文件与原始配置文件是否相符；

[0032] 第二结果判定模块，用于若所述配置文件与所述原始配置文件不相符，则判定所述服务器进程存在所述Webshell网页。

[0033] 第三方面，本申请还公开了一种JVM中的Webshell网页检测设备，包括：

[0034] 存储器，用于存储计算机程序；

[0035] 处理器，用于执行所述计算机程序时实现如上所述的任一种JVM中的Webshell网页检测方法的步骤。

[0036] 第四方面，本申请还公开了一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现如上所述的任一种JVM中的Webshell网页检测方法的步骤。

[0037] 本申请所提供的一种JVM中的Webshell网页检测方法，包括根据检测指令遍历服务器进程；判断所述服务器进程中的字节码文件与原始字节码文件是否相符；若所述字节码文件与所述原始字节码文件不相符，则判定所述服务器进程存在Webshell网页；若所述字节码文件与所述原始字节码文件相符，则判断监听器的配置文件与原始配置文件是否相符；若所述配置文件与所述原始配置文件不相符，则判定所述服务器进程不存在所述Webshell网页。

[0038] 可见，本申请所提供的JVM中的Webshell网页检测方法，通过对服务器中的字节码文件和监听器的配置文件进行匹配判断，实现了对服务器中Webshell网页的检测，为Webshell网页检测结果提供了双重保证，可以更为准确的检测到服务器中是否存在

Webshell网页，以便对Webshell网页进行及时清理，降低Webshell网页对服务器安全运行所带来的威胁，有效提升了服务器的安全性能。

[0039] 本申请所提供的一种JVM中的Webshell网页检测装置、设备及计算机可读存储介质，均具有上述有益效果，在此不再赘述。

附图说明

[0040] 为了更清楚地说明现有技术和本申请实施例中的技术方案，下面将对现有技术和本申请实施例描述中需要使用的附图作简要的介绍。当然，下面有关本申请实施例的附图描述的仅仅是本申请中的一部分实施例，对于本领域普通技术人员来说，在不付出创造性劳动的前提下，还可以根据提供的附图获得其他的附图，所获得的其他附图也属于本申请的保护范围。

[0041] 图1为本申请所提供的一种JVM中的Webshell网页检测方法的流程示意图；

[0042] 图2为本申请所提供的另一种JVM中的Webshell网页检测方法的流程示意图；

[0043] 图3为本申请所提供的一种JVM中的Webshell网页检测装置的结构示意图；

[0044] 图4为本申请所提供的一种JVM中的Webshell网页检测设备的结构示意图。

具体实施方式

[0045] 本申请的核心是提供一种JVM中的Webshell网页检测方法，该JVM中的Webshell网页检测方法可以实现服务器中Webshell的检测，保证服务器的安全运行；本申请的另一核心是提供一种JVM中的Webshell网页检测装置、设备及计算机可读存储介质，也具有上述有益效果。

[0046] 为了对本申请实施例中的技术方案进行更加清楚、完整地描述，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介绍。显然，所描述的实施例仅仅是本申请一部分实施例，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，都属于本申请保护的范围。

[0047] 请参考图1，图1为本申请所提供的一种JVM中的Webshell网页检测方法的流程示意图，该JVM中的Webshell网页检测方法可包括：

[0048] S101：根据检测指令遍历服务器进程；

[0049] 本步骤旨在实现服务器进程的遍历，以获取服务器中所存在的服务器进程。具体的，当接收到检测指令时，即可对服务器进程开始遍历，其中，该检测指令是指用于实现Webshell网页检测的指令，其获取方式并不唯一，可以由用户通过服务器对应的前端直接输入，也可以是根据预设响应条件自动触发，如定时条件，以实现周期性的Webshell网页检测，因此，本申请对于上述检测指令的获取方式不做限定。此外，当设定进行周期性Webshell网页检测时，其检测周期的具体取值也不唯一，由技术人员根据实际需求进行设定即可，本申请对此同样不做限定。

[0050] S102：判断服务器进程中的字节码文件与原始字节码文件是否相符；若是，则执行S103，若否，则执行S104；

[0051] 本步骤旨在实现字节码文件(class文件)的正确性判断，该字节码文件为服务器进程中的文件，可以从服务器进程中提取获得。具体而言，可以先加载各服务器进程中的字

节码文件,进而将其与对应原始字节码文件进行匹配以判断二者是否相符,其中,原始字节码文件是服务器启动时所加载的标准字节码文件,由此,若服务器进程中当前的字节码文件与原始字节码文件不相符,即可证明该服务器进程存在Webshell网页,严重影响服务器的安全运行,反之,则继续进行下一步的判断,以确定该服务器进程中是否真的不存在Webshell网页。

[0052] 作为一种优选实施例,该JVM中的Webshell网页检测方法还可以包括:当字节码文件与原始字节码文件不相符时,将字节码文件转化为java代码,并定位java代码中的Webshell网页。

[0053] 当检测到服务器进程中当前的字节码文件与原始字节码文件不相符时,已经可以毫无疑义的确定该服务器进程存在Webshell网页,此时,可以对Webshell网页进行定位,以便清除Webshell网页,即清除服务器中的安全隐患。具体的,可以先将字节码文件转化为java代码,再在该java代码中定位Webshell网页,在java代码中定位Webshell网页更加方便快捷。

[0054] 其中,上述将字节码文件转化为java代码,可以包括:通过反编译技术将字节码文件转化为java代码。

[0055] 本优选实施例提供了一种字节码文件至java代码的转换方法,具体可基于反编译技术实现,即计算机软件还原工程,其具体实现过程可参照已有技术,本申请在此不再赘述。

[0056] 其中,上述定位java代码中的Webshell网页,可以包括:通过正则匹配技术定位java代码中的Webshell网页。

[0057] 本优选实施例提供了一种从java代码中定位Webshell网页的实现方法,即基于正则匹配技术实现,其具体实现过程参照已有技术即可,本申请在此不再赘述。可以理解的是,正则匹配技术仅为本优选实施例所提供的一种实现方式,并不唯一,还可以为采用其他技术实现,如hook技术(钩子技术)等,具体定位方式可由技术人员根据实际情况进行设定,本申请对此不做限定。

[0058] 作为一种优选实施例,该JVM中的Webshell网页检测方法还可以包括:当字节码文件与原始字节码文件不相符时,判断服务器是否处于业务处理状态;若否,则重启服务器;若是,则利用原始字节码文件替换字节码文件。

[0059] 除检测和定位Webshell网页之外,为保证服务器安全,还可对Webshell网页进行清理操作。具体的,当判定字节码文件与原始字节码文件不相符时,可以通过重启服务器的方式清理Webshell网页,因为重启服务器时会重新加载原始字节码文件,实现当前字节码文件的清理。当然,为避免服务器业务中断,在重启服务器之前,还可以判断服务器当前是否处于业务处理状态,若未处于业务处理状态,直接重启服务器即可;若处于业务处理状态,则可以直接利用原始字节码文件替换当前字节码文件,通过替换文件的方式实现Webshell网页的清理,同时也避免了重启服务器所造成的的业务中断问题。

[0060] S103:判断监听器的配置文件与原始配置文件是否相符;若否,则执行S104,若是,则执行S105;

[0061] S104:判定服务器进程存在Webshell网页;

[0062] S105:判定服务器进程不存在Webshell网页。

[0063] 当判定当前字节码文件与原始字节码文件不相符时，还无法直接毫无疑义的确定服务器中不存在Webshell网页，为给服务器安全提供进一步保证，可以继续对服务器中监听器的配置文件进行匹配判断，该配置文件是服务器中间件中的文件，可以从服务器中间件中检测获得。具体而言，可以从中间件中检测获得配置文件，其具体形式一般为数组形式；进一步，将其与对应的原始配置文件进行匹配以判断二者相符，其中，原始配置文件是服务器启动时用于配置监听器的标准配置文件，由此，若服务器中间件中当前的配置文件与原始配置文件不相符，即可证明该服务器中存在Webshell网页，严重影响服务器的安全运行，反之则可以确定服务器中不存在Webshell网页。

[0064] 作为一种优选实施例，该JVM中的Webshell网页检测方法还可以包括：当配置文件与原始配置文件不相符时，判断服务器是否处于业务处理状态；若否，则重启服务器；若是，则删除配置文件。

[0065] 同样的，当判定当前配置文件与原始配置文件不相符时，为保证服务器安全，也可对Webshell网页进行清理操作。具体的，当判定字节码文件与原始字节码文件不相符时，同样可以通过重启服务器的方式清理Webshell网页，因为重启服务器时会重新加载原始配置文件，并根据其中的配置信息重新配置监听器，实现当前监听器的删除，进而实现Webshell网页的清理。当然，为避免服务器业务中断，在重启服务器之前，同样可以判断服务器当前是否处于业务处理状态，若未处于业务处理状态，直接重启服务器即可；若处于业务处理状态，则直接删除当前监听器即可，以有效避免重启服务器所造成的的业务中断问题。

[0066] 作为一种优选实施例，该JVM中的Webshell网页检测方法还可以包括：根据检测指令执行日志记录操作，获得Webshell网页检测日志。

[0067] 本优选实施例所提供的JVM中的Webshell网页检测方法可实现日志记录功能，以便在Webshell网页检测检测过程中生成完整的日志报告，使得技术人员可以更加方便直观的了解检测结果。具体的，当接收到检测指令时，即可执行实时的日志记录操作，直至检测结束，获得完整的Webshell网页检测日志。进一步，还可对其进行分类存储或可视化展示。

[0068] 可见，本申请所提供的JVM中的Webshell网页检测方法，通过对服务器中的字节码文件和监听器的配置文件进行匹配判断，实现了对服务器中Webshell网页的检测，为Webshell网页检测结果提供了双重保证，可以更为准确的检测到服务器中是否存在Webshell网页，以便对Webshell网页进行及时清理，降低Webshell网页对服务器安全运行所带来的威胁，有效提升了服务器的安全性能。

[0069] 本申请实施例提供了另一种JVM中的Webshell网页检测方法。

[0070] 具体而言，内存webshell的植入有两种方法，第一种是修改已经加载到内存的class字节码，将无害的字节码修改为可执行命令的一句话木马。另一种内存webshell的植入则是针对服务器而言，通过相关服务器自带的监听器实现webshell的植入，在服务器的监听中新增一个监听器用来拦截请求执行命令。因此，可针对这两种服务器内存webshell植入方式，提供相应的Webshell网页检测方法。

[0071] 请参考图2，图2为本申请所提供的另一种JVM中的Webshell网页检测方法的流程示意图，其具体实现流程如下：

[0072] (1)运行检测程序，寻找当前服务器上的JVM(Java Virtual Machine，Java虚拟机)进程(服务器进程)，并通过附加进程对JVM中的相关信息进行访问获得class文件；

[0073] (2)将存在同类名的原始class文件与JVM中的class文件进行十六进制对比,以确定JVM中的class文件是否被恶意修改,若已被修改,则执行(3),否则执行(4);

[0074] (3)用户选择确定是否能重新启动服务器,若是,则重新启动服务器,JVM将会重新加载原始class文件,达到清除webshell的目的;若不能重启,则选择原始class文件,并在内存中对恶意字节码进行替换,达到不重启服务器即可清除webshel的目的;

[0075] (4)对中间件的监听器进行检测,每一个中间件都存在监听器的定义,即用一个数组来存储所配置的监听器,然后通过循环调用保证每一个监听器的运行;由于附加了JVM进程,因此,可以通过执行java代码将中间件中存储监听器的数组读取出来,然后和配置文件进行对比,以判断监听器数据是否发生了新增或者修改,若是,则选择确定是否能重新启动服务器,且重启服务器后,当前监听器数组会根据配置文件重新进行加载,达到清除webshell的目的;若不重启服务器,则可以在当前JVM执行相关代码,将恶意监听器移除,达到清除webshell的目的;

[0076] (5)记录日志,生成检测报告。

[0077] 可见,本申请实施例所提供的JVM中的Webshell网页检测方法,通过对服务器中的字节码文件和监听器的配置文件进行匹配判断,实现了对服务器中Webshell网页的检测,为Webshell网页检测结果提供了双重保证,可以更为准确的检测到服务器中是否存在Webshell网页,以便对Webshell网页进行及时清理,降低Webshell网页对服务器安全运行所带来的威胁,有效提升了服务器的安全性能。

[0078] 为解决上述技术问题,本申请还提供了一种JVM中的Webshell网页检测装置,请参考图3,图3为本申请所提供的一种JVM中的Webshell网页检测装置的结构示意图,该JVM中的Webshell网页检测装置可包括:

[0079] 进程遍历模块1,用于根据检测指令遍历服务器进程;

[0080] 第一文件判断模块2,用于判断服务器进程中的字节码文件与原始字节码文件是否相符;

[0081] 第一结果判定模块3,用于若字节码文件与原始字节码文件不相符,则判定服务器进程存在Webshell网页;

[0082] 第二文件判断模块4,用于若字节码文件与原始字节码文件相符,则判断监听器的配置文件与原始配置文件是否相符;

[0083] 第二结果判定模块5,用于若配置文件与原始配置文件不相符,则判定服务器进程存在Webshell网页。

[0084] 可见,本申请实施例所提供的JVM中的Webshell网页检测装置,通过对服务器中的字节码文件和监听器的配置文件进行匹配判断,实现了对服务器中Webshell网页的检测,为Webshell网页检测结果提供了双重保证,可以更为准确的检测到服务器中是否存在Webshell网页,以便对Webshell网页进行及时清理,降低Webshell网页对服务器安全运行所带来的威胁,有效提升了服务器的安全性能。

[0085] 作为一种优选实施例,该JVM中的Webshell网页检测装置还可包括Webshell定位模块,用于当字节码文件与原始字节码文件不相符时,将字节码文件转化为java代码,并定位java代码中的Webshell网页。

[0086] 作为一种优选实施例,上述Webshell定位模块可具体用于通过反编译技术将字节

码文件转化为java代码。

[0087] 作为一种优选实施例，上述Webshell定位模块可具体用于通过正则匹配技术定位java代码中的Webshell网页。

[0088] 作为一种优选实施例，该JVM中的Webshell网页检测装置还可包括第一Webshell清理模块，用于当字节码文件与原始字节码文件不相符时，判断服务器是否处于业务处理状态；若否，则重启服务器；若是，则利用原始字节码文件替换字节码文件。

[0089] 作为一种优选实施例，该JVM中的Webshell网页检测装置还可包括第二Webshell清理模块，用于当配置文件与原始配置文件不相符时，判断服务器是否处于业务处理状态；若否，则重启服务器；若是，则删除配置文件。

[0090] 作为一种优选实施例，该JVM中的Webshell网页检测装置还可包括日志记录模块，用于根据检测指令执行日志记录操作，获得Webshell网页检测日志。

[0091] 对于本申请提供的装置的介绍请参照上述方法实施例，本申请在此不做赘述。

[0092] 为解决上述技术问题，本申请还提供了一种JVM中的Webshell网页检测设备，请参考图4，图4为本申请所提供的一种JVM中的Webshell网页检测设备的结构示意图，该JVM中的Webshell网页检测设备可包括：

[0093] 存储器10，用于存储计算机程序；

[0094] 处理器20，用于执行计算机程序时可实现如上述任意一种JVM中的Webshell网页检测方法的步骤。

[0095] 对于本申请提供的设备的介绍请参照上述方法实施例，本申请在此不做赘述。

[0096] 为解决上述问题，本申请还提供了一种计算机可读存储介质，该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可实现如上述任意一种JVM中的Webshell网页检测方法的步骤。

[0097] 该计算机可读存储介质可以包括：U盘、移动硬盘、只读存储器(Read-Only Memory，ROM)、随机存取存储器(Random Access Memory，RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0098] 对于本申请提供的计算机可读存储介质的介绍请参照上述方法实施例，本申请在此不做赘述。

[0099] 说明书中各个实施例采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言，由于其与实施例公开的方法相对应，所以描述的比较简单，相关之处参见方法部分说明即可。

[0100] 专业人员还可以进一步意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

[0101] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块，或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存

储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM或技术领域内所公知的任意其它形式的存储介质中。

[0102] 以上对本申请所提供的技术方案进行了详细介绍。本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想。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,还可以对本申请进行若干改进和修饰,这些改进和修饰也落入本申请的保护范围内。

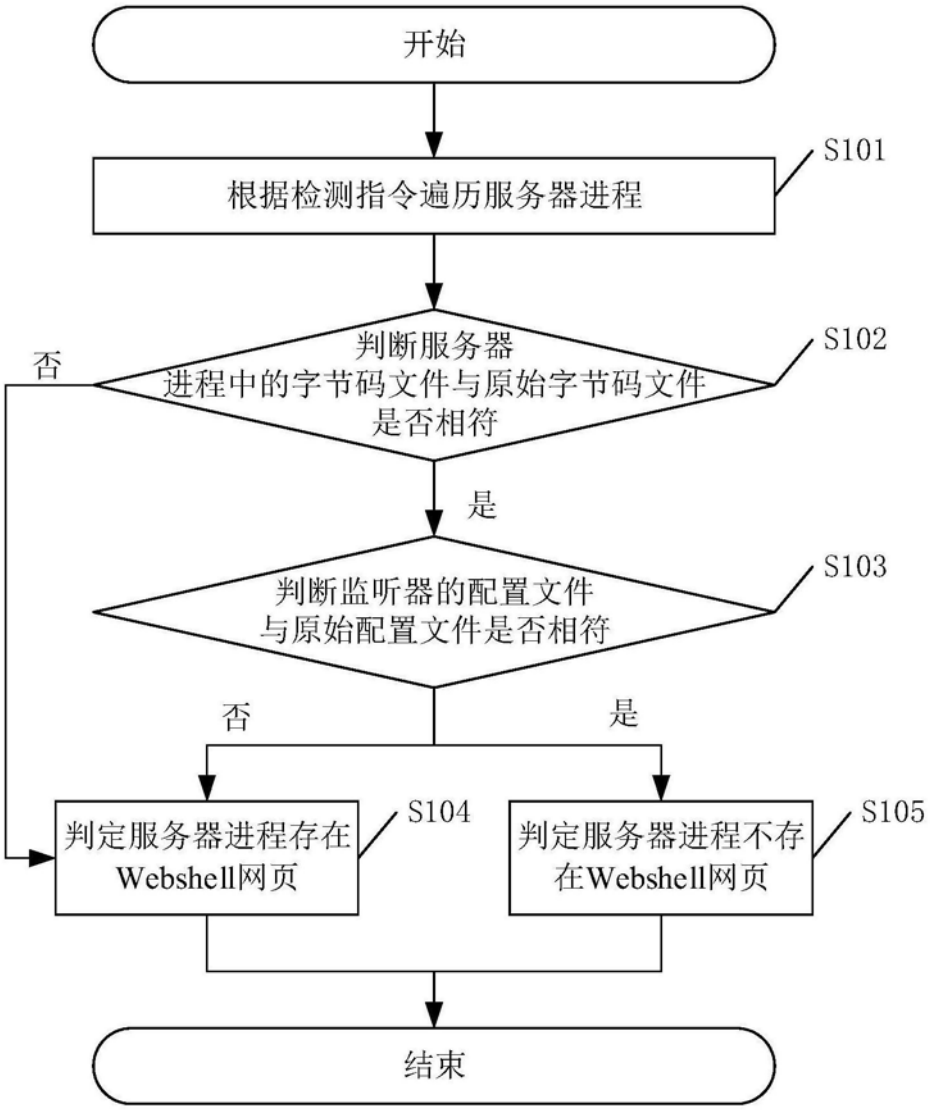

开始
S101
根据检测指令遍历服务器进程
S102
判断服务器
进程中的字节码文件与原始字节码文件
是否相符
否
是
S103
判断监听器的配置文件
与原始配置文件是否相符
否
是
S104
判定服务器进程存在
Webshell网页
S105
判定服务器进程不存
在Webshell网页
结束

图1

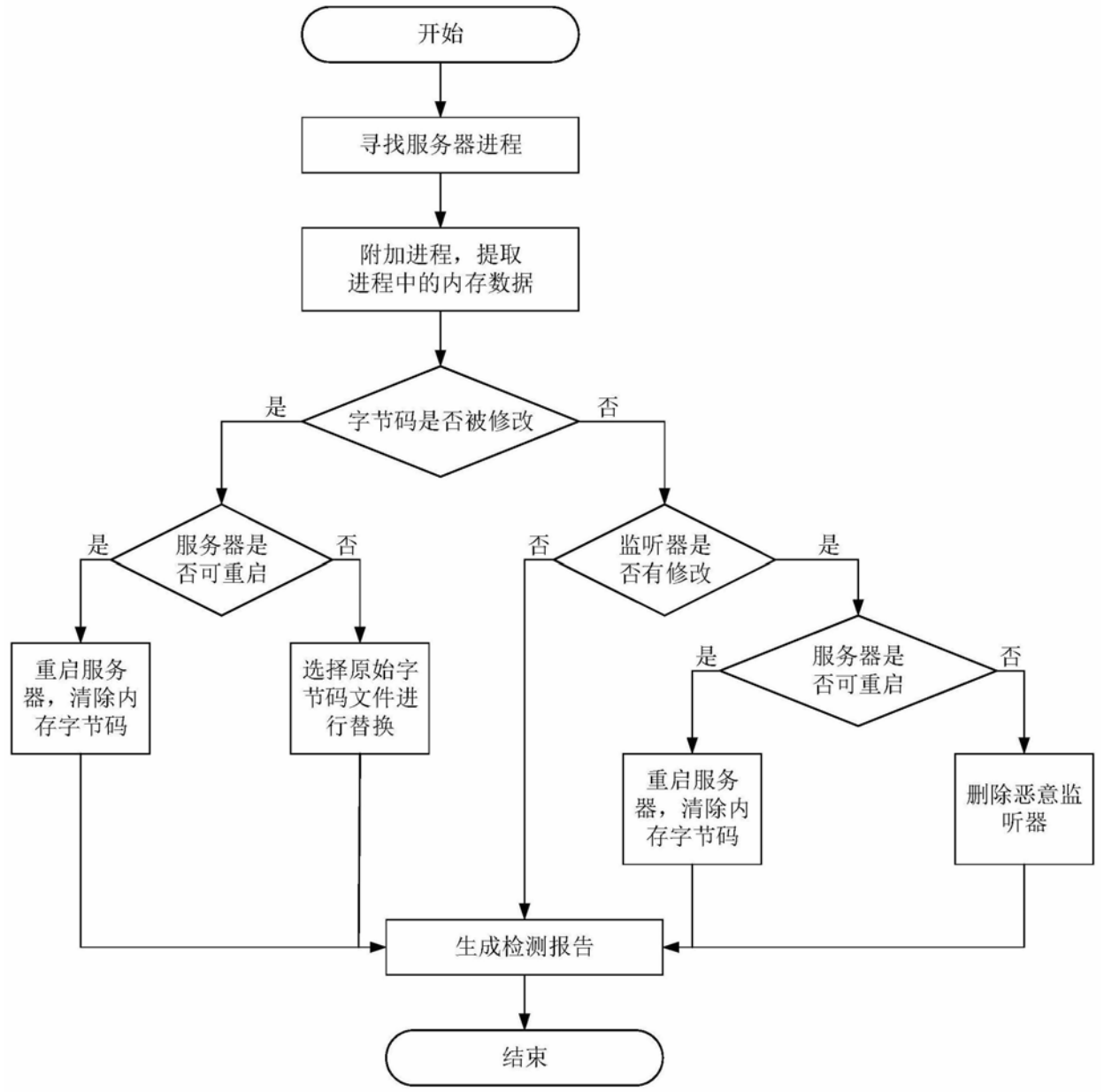

开始
寻找服务器进程
附加进程，提取
进程中的内存数据
字节码是否被修改
是
否
服务器是
否可重启
是
否
监听器是
否有修改
否
是
重启服务
器，清除内
存字节码
选择原始字
节码文件进
行替换
服务器是
否可重启
是
否
重启服务
器，清除内
存字节码
删除恶意监
听器
生成检测报告
结束

图2

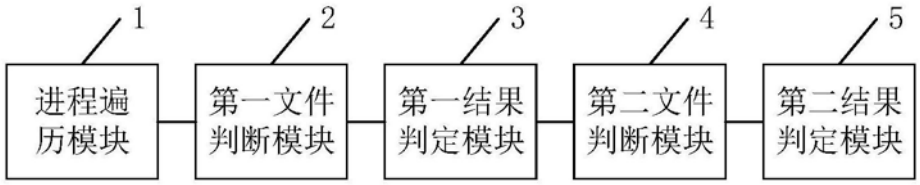

1
2
3
4
5
进程遍
历模块
第一文件
判断模块
第一结果
判定模块
第二文件
判断模块
第二结果
判定模块

图3

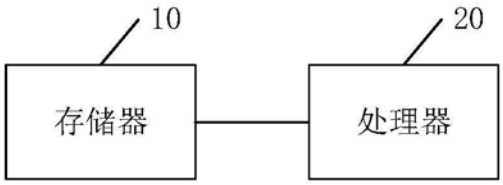


图4