(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 102224505 B
(45)授权公告日 2014.06.04

(21)申请号 200980145309.7
(22)申请日 2009.11.19
(30)优先权数据
61/199,728 2008.11.19 US
(85)PCT国际申请进入国家阶段日
2011.05.12
(86)PCT国际申请的申请数据
PCT/US2009/065171 2009.11.19
(87)PCT国际申请的公布数据
WO2010/059843 EN 2010.05.27
(73)专利权人 安全工程有限公司
地址 美国佐治亚州
(72)发明人 A·达文波特 H·金 J·R·拉姆泽
(74)专利代理机构 北京润平知识产权代理有限公司 11283
代理人 南毅宁 周建秋
(51)Int.Cl.
G06F 21/54(2013.01)
H04L 29/06(2006.01)

(56)对比文件
US 2007/0136811 A1,2007.06.14,
US 2004/0098623 A1,2004.05.20,
US 5949973 A,1999.09.07,
CN 1981289 A,2007.06.13,
US 2005/0011947 A1,2005.01.20,
US 2006/0075494 A1,2006.04.06,

审查员 田民丽

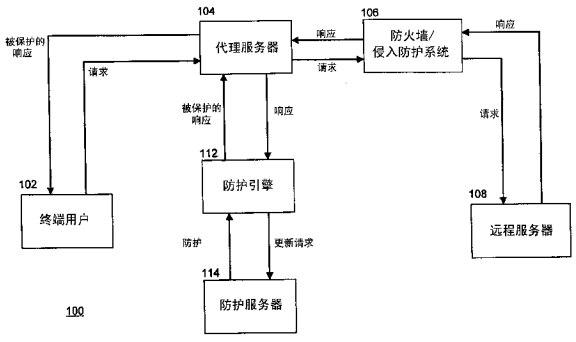
权利要求书3页 说明书14页 附图10页

(54)发明名称

用于运行时攻击预防的系统和方法

(57)摘要

防止对计算机在运行时的攻击。至少一个构造成可存取计算机的功能的内容由计算机接收。相应于功能的防护被添加到内容,其中,该防护覆盖功能。随后,内容和防护被发送给计算机。功能可能暴露计算机的漏洞,而传递给功能的参数可能利用那个漏洞。当内容被执行时,防护被执行,并确定内容传递给功能的参数是否构成威胁。响应于确定出参数构成威胁,内容的执行在功能不被执行的情况下被终止。



CN 102224505 B

1. 一种用于防护计算机的方法，该方法包括：

在防护设备处，接收以计算机为目的地的内容，其中所述内容被配置成存取所述计算机的功能，并且其中所述内容从服务器接收，并且其中所述内容包括对所述功能的调用；

在所述防护设备处，为相应于所述功能的内容添加防护，其中所述防护被配置成监视所述功能并且其中添加所述防护进一步包括：

确定与所述计算机相关联的计算环境；

识别与所述计算环境相对应的第一防护；

将所述第一防护添加到所述内容；

向所述计算机传送所述内容和防护；以及

响应于对所述功能的调用执行所述防护，其中执行所述防护包括确定内容传递到功能的参数是否代表对所述计算机构成威胁。

2. 根据权利要求 1 所述的方法，其中，监视所述功能包括覆盖所述功能。

3. 根据权利要求 1 所述的方法，其中，监视所述功能包括分析传递到所述功能的参数。

4. 根据权利要求 1 所述的方法，其中，所述功能包括暴露所述计算机的漏洞的功能。

5. 根据权利要求 1 所述的方法，其中，传递给所述功能的参数利用所述计算机的漏洞。

6. 根据权利要求 1 所述的方法，进一步包括，响应于确定出参数代表构成威胁，在功能不被执行的情况下终止内容的执行的步骤。

7. 根据权利要求 1 所述的方法，进一步包括，响应于确定出参数代表构成威胁，修改所述内容的步骤。

8. 根据权利要求 1 所述的方法，进一步包括，响应于确定出不存在威胁，允许所述功能正常执行的步骤。

9. 根据权利要求 1 所述的方法，进一步包括，响应于确定出参数代表构成威胁，向终端用户、网络管理员和第三方中的至少一者警告该威胁。

10. 根据权利要求 1 所述的方法，其中，计算环境包括安装在所述计算机上的操作系统、互联网浏览器和应用程序中的至少一个。

11. 根据权利要求 1 所述的方法，进一步包括如下步骤：

确定所述内容的类型；

识别与所述类型相应的第一防护；以及

将所述第一防护添加到所述内容。

12. 根据权利要求 11 所述的方法，其中，所述内容的类型包括用于 JavaScript、Flash 和 Silverlight 中的至少一个的可执行代码。

13. 根据权利要求 1 所述的方法，其中，添加所述防护的步骤包括为所述内容预设防护。

14. 一种用于防护计算机的系统，该系统包括：

被配置成提供内容的第一计算机；

被配置成从所述第一计算机接收内容的第二计算机，所述内容被配置成存取所述第二计算机的功能；

耦连于所述第一计算机的防护引擎，该防护引擎被配置成在所述内容被所述第二计算机接收之前拦截该内容、为所述内容添加防护、向所述第二计算机传送所述防护和内容，且

响应于对所述功能的调用执行该防护，其中执行所述防护包括确定内容传递到功能的参数是否代表对所述计算机构成威胁，

其中在所述添加防护时，所述防护引擎还被配置成：

确定与所述第二计算机相关联的计算环境；

识别与所述计算环境相对应的第一防护；

将所述第一防护添加到所述内容；以及

其中，所述防护覆盖所述功能，以使得在所述功能代表对所述第二计算机构成威胁时该防护在内容调用功能时被执行。

15. 根据权利要求 14 所述的系统，其中，为内容添加防护包括为内容预设防护。

16. 根据权利要求 14 所述的系统，其中，所述第一计算机包括所述防护引擎。

17. 根据权利要求 14 所述的系统，其中，所述第一计算机包括所述第二计算机。

18. 根据权利要求 14 所述的系统，其中，所述防护被配置成一旦确定出所述功能包括威胁，则修改所述内容。

19. 根据权利要求 14 所述的系统，其中，所述防护被配置成一旦确定出所述功能包括威胁，则终止所述内容的执行。

20. 根据权利要求 14 所述的系统，其中，所述防护被配置成一旦确定出所述功能包括威胁，则警告终端用户、网络管理员和第三方中的一者。

21. 一种用于运行时防止对计算机的攻击的方法，该方法包括：

在服务器处，接收对来自于计算机的可执行内容的请求，其中，所述可执行内容被配置成存取所述计算机的功能；

将所述内容从所述服务器发送到所述计算机；

在防护设备处拦截所述内容；

在所述防护设备处，为所述内容添加防护，其中，所述防护覆盖所述可执行内容可存取的功能，其中所述添加防护进一步包括：

确定与所述计算机相关联的计算环境；

识别与所述计算环境相对应的第一防护；以及

将所述第一防护添加到所述内容；

从所述防护设备向所述计算机传送所述防护和内容；以及

响应于对所述功能的调用，执行所述防护，其中执行所述防护包括确定内容传递到功能的参数是否代表对所述计算机构成威胁。

22. 根据权利要求 21 所述的方法，进一步包括执行内容和防护的步骤，该步骤包括：

利用防护确定可执行内容传递给所述功能的参数是否代表对所述计算机构成威胁。

23. 根据权利要求 22 所述的方法，进一步包括响应于确定出所述参数代表对所述计算机构成威胁，终止所述内容的执行。

24. 根据权利要求 22 所述的方法，进一步包括响应于确定出所述参数代表对所述计算机构成威胁，修改所述内容。

25. 根据权利要求 22 所述的方法，进一步包括向终端用户、网络管理员和第三方中的一者警告所述威胁。

26. 根据权利要求 21 所述的方法，其中，所述内容包括 JavaScript、Flash 和

Silverlight 中的一个。

27. 根据权利要求 21 所述的方法，其中，添加所述防护包括预设所述防护。

28. 根据权利要求 21 所述的方法，其中，所述防护在所述服务器处被添加。

29. 一种用于防护计算机的系统，该系统包括：

耦连于网络的计算机，暴露至少一个功能，该计算机被配置成经由所述网络接收被配置成存取所述功能的内容，且一旦接收到，则执行该内容，所述功能包括至少一个参数；

防护系统，耦连于所述网络并被配置成：

在所述内容被所述计算机接收之前拦截该内容；以及

为该内容预设防护，其中在为所述内容预设防护的情况下，所述防护系统进一步被配置为；

确定与所述计算机相关联的计算环境；

识别与所述计算环境相对应的第一防护；以及

将所述第一防护添加到所述内容；以及

从所述防护设备向所述计算机传送所述防护和内容；

其中所述计算机被配置成响应于对所述功能的调用，执行所述防护，其中执行所述防护包括确定所述参数是否代表对所述计算机构成威胁；

其中所述防护被配置成当所述内容试图执行所述功能时被所述计算机执行，

其中所述防护被进一步配置成分析所述参数以确定在所述参数被传递给所述功能的情况下该参数是否包括对所述计算机的威胁，

其中所述防护被进一步配置成响应于确定出所述参数在被传递给功能时包括对所述计算机的威胁，修改所述内容的执行，以及

其中所述防护被进一步配置成，响应于确定出所述参数在被传递给功能时不包括对所述计算机的威胁，允许根据所述内容的调用来执行该功能。

30. 根据权利要求 29 所述的系统，其中修改所述内容的执行的步骤包括终止所述内容的执行。

31. 根据权利要求 29 所述的系统，其中所述内容包括 JavaScript、Flash 和 Silverlight 中的至少一个。

32. 根据权利要求 29 所述的系统，其中所述内容包括一旦被接收则被执行的 ASCII 文本。

33. 根据权利要求 29 所述的系统，其中所述计算机进一步包括被配置成一旦接收则执行所述内容的网络浏览器，且所述功能由该网络浏览器暴露。

34. 根据权利要求 29 所述的系统，其中所述防护进一步被配置成响应于确定出所述参数包括在被传递给功能的情况下对所述计算机的威胁，向终端用户、网络管理员和第三方中的至少一者传送警告。

用于运行时攻击预防的系统和方法

[0001] 交叉引用相关专利申请

[0002] 本申请要求根据 35 U.S.C. §119 要求在 2008 年 11 月 19 日申请的，名称为“虚拟安全库”的 US 临时专利申请 61/199,728 的优先权。因此上述在先申请的全部内容全部合并于此作为参考。

技术领域

[0003] 本发明大体涉及计算机和网络安全的领域，且更具体地涉及识别恶意的程序和代码，并防止其执行。

背景技术

[0004] 几十年来，黑客们已寻找到将恶意程序（“恶意软件”）安装到计算机或其他联网设备的途径。通常，恶意软件运行以损坏或妨害安装了其恶意软件的计算机（“主机”）的功能。恶意软件也可为黑客提供呈现在主机上的敏感信息的通路，以及主机所连接到的网络上的途径。通过侵占主机的电子邮件和其他网络通信工具，恶意软件也可使用主机去发布另外的恶意软件。

[0005] 恶意程序一般透过否则看来良性的程序或文件渗入而被安装到主机上。例如，恶意软件可被包含在电子邮件消息中，且可在打开电子邮件消息或查看消息内的图像时自动运行。恶意软件也可与主机的使用者有意图地下载和 / 或安装的程序相关联。

[0006] 通常，检测这些威胁的主要方法为识别与各种恶意软件有关的签名。签名通常为可执行的字符序列，其在攻击已发生并被报告给受过专门训练的分析者之后被识别。分析者识别恶意软件和与其有关的签名。随后分析者以可执行代码或以网络通讯设计识别该签名的软件。对于最常规的病毒防护、反间谍软件和入侵检测系统，这是典型的工作模式。

[0007] 黑客非常清楚这种常规的防御，且已开发了新的攻击，其被设计成使用新的技术来规避基于签名的检测。一种这样的攻击使用可执行的互联网代码，诸如 JavaScript。JavaScript 为设计成为嵌入在超文本标记语言 (HTML) 页面的编程语言。JavaScript 提供接近网络浏览器外的可执行目标的途径。例如，JavaScript 允许网页打开存储在主机硬盘的文件以及修改该文件。JavaScript 还允许在主机的存储器中的数据结构的创建。由于 JavaScript 为脚本语言，它不被转化成机器语言，而是以文本的形式被发送给终端用户的网络浏览器，随后在那里其被执行。

[0008] 这些特征为黑客提供了可利用的途径。例如，可能是已知的，在指定的操作系统或网络浏览器中，创建特定大小的阵列可允许黑客接近计算机存储器的特别敏感的区域。JavaScript 允许这样的阵列的创建。创建数据结构需要特定的已知语法，所以基于签名的检测仍旧可能是可行的。然而，由于程序员在表达变量的名称上和设置在阵列中的数据上有广泛的自由，基于签名的检测变得更困难，甚至不可行，因为攻击可能直到恶意代码已在主机上运行时才变得明显。

[0009] 然而，在该假设的示例中，在基于签名的检测可行的程度上，诸如 JavaScript 的

语言为黑客提供另外的途径来躲避基于签名的检测方案。具体地，JavaScript 提供可随后被执行的字符串的运行时评估。例如，假设叫作 document.write(“hack”) 的功能为已知的溢出漏洞。黑客可利用 JavaScript 的运行时评估的特征如下所述地混淆该溢出漏洞：

[0010] var X =“ha”;

[0011] var Y =“k”;

[0012] var Z =“doc”;

[0013] var A =“ument.wr”;

[0014] var B =“ite(”;

[0015] var C =“)”;

[0016] var D =“c”;

[0017] eval(Z+A+B+X+D+Y+C);

[0018] 上述代码将字符串估值为 document.write(“hack”) 并执行该恶意代码。使该问题复杂化的是，JavaScript 几乎没有规定对于变量的名称或者它们出现的顺序的限制。因此，对于混淆恶意代码的途径，黑客们只受他们的想象力限制。而且，由于 JavaScript 可在网页被访问时自动产生，如上所述的混淆手法可以这样的方式响应于访问随机产生，以确保每一次攻击与上一次具有不同的签名。因此，即使是可行的，常规的基于签名的检测也会很困难检测到这种新型的互联网攻击。

[0019] 这样的攻击的进一步问题为，其可被隐藏在从其他可信源接收的代码中。例如，得到广泛使用的网站可售卖它网站上的广告空间给广告客户。当终端用户浏览网页时，显现在那个空间的广告典型地从广告客户或第三方广告服务器所加载。广告也可在终端用户浏览网页之前被加载到与网站相关联的服务器上，在此情况下，当终端用户浏览该网站时，该广告也可被提供给用户而不用连接到外部服务器。除了契约上的控制之外，运行该网站的公司对广告的内容可能只有很少的控制。如果恶意代码被设置在广告中，网站可能直到它已经提供了无数的受感染的网页才发现问题。因此，一般用于阻止来自于不可信的来源的网络通讯或者仅仅允许来自于可靠来源的通讯的方法可能并不足够用于阻止攻击。

[0020] 阻止这样的攻击的常规方法是禁止基于网络的可执行代码的执行，该可执行的代码包括但不限于 JavaScript、Adobe Flash 和 Microsoft Silverlight。然而，阻止这样的代码的执行，可能使得某些网站不能正确地运行，且减损了该代码允许的丰富的网络内容。

[0021] 因此，现时存在对于可识别和阻止在可执行代码中传递的攻击的系统的需求。进一步地，现时存在对于下列的系统的需求，该系统在即使当攻击被混淆以至于直到运行时才能检测到时，也可识别和阻止这种攻击。现时更存在信誉良好的网站经营者对访问者的保护，以使访问者不受可能隐匿在经由他们网站公布的内容中，诸如由第三方广告客户控制的调幅广告，的攻击。

发明内容

[0022] 本发明通过提供一种用于预防运行时攻击的系统和方法来满足上述需求。在本发明的一个方面中，提供了一种可接收计算机指定的内容方法。该内容可被构造成可接近使计算机漏洞暴露的至少一个功能。覆盖该功能或者修改该内容的执行的防护随后可被添加到该内容。包括该防护的该内容随后被发送给计算机。

[0023] 计算机一旦接收了该内容和该防护，便会执行该内容。如果该内容调用使计算机漏洞暴露的功能时，该防护便会被执行。通过执行该防护，该内容试图传递到该功能的任何参数都被分析，以判定这些参数是否带威胁性。如果确定该参数对计算机构成威胁，该内容的执行在该功能未被执行时被终止。另一方面，如果该参数不带威胁性，该漏洞功能将被允许正常执行。

[0024] 如果传递给该功能的参数被确定带有威胁性，而该内容的执行被终止，警报将被发送给网络管理员。警报也可被发送给终端用户或者第三方。

[0025] 当增加防护时，该方法可考虑与计算机有关的计算环境。计算环境可包括与计算机的操作系统有关的信息、在计算机上运行的网络浏览器的类型和版本、和 / 或安装在计算机上的应用程序。于是该方法可识别与计算环境相应的防护并将这些防护添加到该内容。当添加防护时，该方法也可考虑该内容的类型，其可包括 JavaScript、Flash 和 Silverlight。该方法可识别与该内容的类型相应的防护，并可将这些防护添加到该内容。

[0026] 在本发明的另一方面，提供了一种用于防止计算机在运行时被攻击的系统。该系统包括被构造成可接收内容的第一计算机。防护引擎被耦连到第一计算机，并被构造成可拦截该内容，并添加防护到该内容。该防护覆盖易被该内容接近的功能，以使得当该内容调用该功能时，执行该防护。该防护可被构造成能确定该内容传递给该功能的参数是否一种攻击。该防护也可被构造成在确定该参数为攻击时，终止该内容的执行。该防护也可修改该内容的执行，以便阻挠依赖于该内容的执行的特定特性的攻击。该防护也可被构造成，在确定对该功能的调用为攻击时，发送警报给终端用户、网络管理员和 / 或另外的第三方。

[0027] 本发明的另一个方面提供一种用于防止在对计算机运行时被攻击的附加的方法。在该方法中，远程服务器接收来自计算机的可执行内容的请求。防护被添加到该内容中，其中该防护覆盖易被该内容接近的功能。随后，该防护和该内容被发送给计算机。

[0028] 计算机于是执行该内容和该防护。该防护确定该内容传递给该功能的参数是否对计算机带有威胁性。响应于该参数对计算机带有威胁性的判定，该防护终止该内容的执行。该防护也可修改该内容的执行，以便阻挠依赖于该内容的执行的特定特性的攻击。该防护也可将该威胁警告发放给终端用户、网络管理员或者第三方。

[0029] 通过参考下面的示例实施例的详细说明，本发明的另外的方面、目的、特征和优点对于本领域技术人员将变得显而易见。现在结合下述附图参考随后的说明，以便更全面理解本发明的示例实施例及其优点。

附图说明

[0030] 图 1 为描述网络结构的框图，该网络结构实施根据本发明的示例实施例的用于运行时攻击预防的系统；

[0031] 图 2 为描述利用图 1 的用于运行时攻击预防的系统在网络通讯中插入防护的不例方法的流程图；

[0032] 图 3 为描述，利用图 1 的用于运行时攻击预防的系统，创建防护的示例方法的流程图；

[0033] 图 4 为描述，利用图 1 的用于运行时攻击预防的系统，在网络中插入防护的示例方法的流程图；

[0034] 图 5 为描述用于预先计划防护的示例方法的流程图；

[0035] 图 6 为描述用于执行为了运行时攻击预防的被保护内容的示例方法的流程图；

[0036] 图 7 为描述网络结构的框图，该网络结构实施根据本发明的第二示例实施例的用于运行时攻击预防的系统；

[0037] 图 8 为描述利用图 7 的用于运行时攻击预防的系统在服务器中插入防护的示例方法的流程图；

[0038] 图 9 为描述网络结构的框图，该网络结构实施根据本发明的第三示例实施例的用于运行时攻击预防的系统；

[0039] 图 10 为描述利用图 9 的运用运行时攻击预防的系统在终端用户的计算机上插入防护的示例方法的流程图。

具体实施例

[0040] 用于预防运行时攻击的创新系统可截断包含可执行内容的网络通讯，该可执行内容的指定目的地为利用来自于网络的内容的终端用户，代理或设备。终端用户的计算机，也称为主机，暴露了许多该内容可接近的功能。这些功能中的一些可能暴露与计算机、它的操作系统或安装在计算机上的其他应用程序有关的漏洞。通过截断该内容，该创新的系统将防护添加到覆盖（或“遮掩”）该功能的内容。当该内容试图调用计算机所暴露的功能时，该防护被执行。如此处所使用的，术语计算机是指智能终端，诸如连接到诸如互联网的外部网络的个人计算机。术语计算机也指任何可联网的设备，其具有直接或通过网络耦连于存储装置的处理器或控制器，并能在网络上接收可执行代码并执行该代码，例如包括，膝上型电脑、手持电脑、移动电话、个人数码助手 (PDA)、便携式音频播放器以及全球定位系统。正如在整个本申请中所使用的，术语“计算机”仅仅为了方便而使用，且如上所述的，术语“计算机”在本文还包含任何计算机或联网设备。

[0041] 防护被构造成可基于该内容试图传递给该功能的参数或其它信息（包括状态信息，诸如特定功能已被调用的次数）来确定该内容是否包含对计算机的威胁。如果防护确定由该内容实行的特定功能调用构成威胁，防护可终止该内容的执行。防护也可修改该内容的执行以便阻挠依赖于该内容的执行的特定特性的攻击。另一方面，如果防护确定具体功能调用没有威胁性，该功能可被允许正常执行（如，由该内容所调用）。该防护也可就该威胁警告终端用户、网络管理员或其他第三方。在执行该内容之前在网络传递的可执行内容中插入防护可防止针对隐藏在该内容中所暴露的功能的攻击。

[0042] 在添加防护之前，该创新的系统可考虑该内容和请求的计算机的某几方面，以有助于简化该过程。例如，如果内容的请求由互联网浏览器的特定版本发送，该系统仅仅可将防护添加到与特定互联网浏览器有关的内容。类似地，如果该内容为某种类型的可执行代码（如，JavaScript），该系统仅仅可添加适于 JavaScript 内容的那些防护。

[0043] 现在转到附图，其中，用样的参考标记引证同样的元件，图 1 为描述网络结构的框图，该网络结构实施根据本发明的示例实施例的用于运行时攻击预防的系统 100。一个或多个终端用户 102 在网络内操作计算机。

[0044] 终端用户 102，经由它们的智能终端，在网络上请求来自于远程服务器 108 的内容。在示例实施例中，请求在当终端用户 102 打开它们计算机上的诸如 Microsoft Internet

Explorer、Mozilla Firefox、Mozilla Flock、Goole Chrome、Opera 或 Apple Safari 的网络浏览器并浏览网址时产生。终端用户 102 的计算机发送请求到代理服务器 104 或网络上的诸如路由器的其他设备，其评价该请求并将该请求发送给防火墙或侵入防护系统 106，随后防火墙或侵入防护系统将该请求传递给远程服务器 108。

[0045] 远程服务器 108 接收并响应该请求产生发送回终端用户 102 的内容。在示例实施例中，远程服务器 108 为万维网服务器，其存储可在计算机上显示给终端用户 102 的各种类型的内容。在该实施例中，该内容由 ASCII 格式的 HTML（超文本标记语言）代码组成，该代码可由终端用户 102 的浏览器显示。该内容也可由诸如图像的二进制编码信息组成。该图像可以许多传统的图像格式储存，诸如联合图像专家组格式 (JPEG 或 JPG)，可交换的图像文件 (GIF)、标签图像文件格式 (TIFF)、可移植文件格式 (PDF) 或其他可在网络上传递的图像格式。

[0046] 该内容也可由可被浏览器或其他应用程序执行的代码组成，该浏览器或其他应用程序可在终端用户 102 的计算机上执行该代码。现时存在几个对于传递可执行代码给浏览器的标准，其包括 JavaScript、Adobe Flash 及它的相关汇编语言 ActionScript、以及 Microsoft Silverlight。例如，JavaScript 由 ASCII 编程指令所代表，并在由浏览器收到时执行。

[0047] 远程服务器 108 指引响应内容返回到终端用户 102。然而，在该内容到达终端用户 102 之前，它必须穿过防火墙 / 侵入检测系统 106。防火墙 106 可惯例地基于一组易于识别的特性，诸如通讯的源或目的地址，被设置成用于阻止有害的网络通讯。假设响应成功地穿过防火墙 106，其也可由侵入检测系统 (IDS) 所检查，该侵入检测系统典型地被构造成，对网络通讯尽力实施额外的、更严格的检验以便识别（和防止）威胁。例如，IDS 可储存与网络通讯有关的信息和签名以便识别匹配于与已知威胁相关联的模式的通讯。

[0048] 假设响应成功地穿过 IDS，该响应将被传递给代理服务器 104。在传统的网络中，代理服务器 104 储存与终端用户 102 发送的导致响应的请求有关的信息，并将响应发送给适当的终端用户 102 的计算机。然而，在本发明的示例实施例中，防护引擎 112 介入该过程并为终端用户 102 增加额外的防护层。

[0049] 在示例实施例中，防护引擎 112 与运转代理服务器 104 在相同的服务器上运行。然而，在另一个示例实施例中，防护引擎 112 可为完全独立的服务器，其被构造成在代理服务器 104 将内容发送回给终端用户 102 之前，接收来自于代理服务器 104 的内容。在进一步的另一个示例实施例中，防护引擎 112 可被设置在网络数据路径内的任何位置。例如，防护引擎 112 可被设置在防火墙 /IDS 106 之前、在防火墙 /IDS 106 和代理服务器 104 之间或者代理服务器 104 和终端用户 102 之间。防护引擎 112 的最终部署不必处于任何特定的位置，且可依赖于多个与特定网络拓扑结构相关的因素。

[0050] 再一次参考示例实施例，其中，防护引擎 112 与代理服务器 104 相关联，当响应传递到代理服务器 104 时，防护引擎 112 阻截该响应。防护引擎 112 接收该响应并确定该响应是否包括可能携带有攻击的内容类型。在本发明的示例实施例中，防护引擎 112 被构造成可识别包含在响应内，作用为将特定类型的内容的出现通知给终端用户的浏览器的文本或其他信号。例如，如果响应包括 JavaScript，那么该响应将包含诸如“<script type＝”text/javascript”>”的标志。类似地，如果响应包括动画内容，那么它将包含诸如

“<object data =”movie.swf”type =”application/x-shockwave-flash”>”。通过接收这样的标志(或者与其他类型可执行内容的有关联的其他类似标志),防护引擎112会将该响应识别为包含可能携带攻击的内容。

[0051] 一旦防护引擎112该响应识别为具有可能携带攻击的内容的类型,防护引擎112预先为该内容设置防护。在示例实施例中,防护为在隐藏的恶意代码执行之前执行,并提供抵抗隐藏在代码中的攻击的屏障的代码。示例的防护覆盖具有新功能的核心功能,其被构造成可观测内容的功能性并在允许该核心功能运行前识别攻击。术语“覆盖”表示,替代如此处所述的核心功能的正常功能的防护。该术语还包含了替代或处理现有的核心功能的功能的任何技术,包括“隐藏”核心功能。

[0052] JavaScript为这样的语言的一个示例,对于该语言,核心功能可被接近以至于允许攻击且也可被覆盖以便抵抗攻击。仅仅以示例的方式,JavaScript包括一种名为document.write(<arguments>)的功能,其可被用于利用互联网浏览器中的溢出漏洞—安全弱点。防护可覆盖document.write功能,从而在执行document.write核心功能之前,为防护提供对传递给功能的参数提供分析的机会。在该示例实施例中,在网页开始被加载时,核心document.write功能可通过为它分配新的对象或变量(例如,document.write.old)而被覆盖。通过这种方式,原始的document.write功能变为document.write.old。于是,当内容调用document.write时,防护被执行,如果指示核心功能没有攻击,该核心功能可被执行。

[0053] 用于document.write的防护的伪代码表示如下所述:

[0054]

```
var document.write.old = document.write;
document.write(string input) {
    if (analyze_for_attacks(input) = true) {
        alert;
        terminate javascript;
    }
    else {
        document.write.old(input);
    }
}
```

[0055] 如上述示例的伪代码所指示的,当所接收的内容中JavaScript包含对document.write的调用时,终端用户102的浏览器将运行防护而不是直接运行功能。防护功能分析内容试图传递给document.write作为威胁的参数。如果该参数代表威胁,功能将该威胁向用户报警,并在不曾调用核心的document.write功能的情况下终止功能的执行。而通过这种方式,攻击不曾在终端用户102的计算机上执行。另一方面,如果参数不代表威胁,document.write.old功能将被调用,以允许核心的document.write功能正常执行。

[0056] 尽管上述示例覆盖document.write()功能,防护可被添加到覆盖任何功能的内容,其包括与单独地从终端用户102的浏览器执行的应用程序相关联的功能。确实,在许多

情况下，汇编语言在浏览器内启动单独的应用程序的能力可为攻击提供额外的通道，这是由于许多应用程序包含可被利用的弱点。因此，防护不仅仅限于浏览器功能，恰恰相反，他们可用于可被响应内的可执行代码或脚本调用的任何功能。而且，预设防护的能力不限于以 JavaScript 所写的功能，而是可应用于可被包含在内容中的任何可执行代码。

[0057] 在本发明的示例实施例中，防护引擎 112 将与包含在响应内的内容的类型有关的所有防护添加到该内容中。通过这种方式，防护引擎 112 可确保任何潜在的危险功能也被覆盖。这可能特别重要，这是因为，黑客的常规技术，此处被称为“混淆”，可使得很难确定哪个功能调用被包含在响应内。

[0058] 混淆的一个示例包含 eval 功能的使用。Eval 功能是有用的功能，其允许通过将字符串传递给 eval 功能，使 eval 功能像可执行代码一样处理字符串。由于被传递给 eval 功能的字符串可在运行时产生，可能直到要被调用的功能被执行时，才能识别该功能。

[0059] 例如，假设功能调用 document.write(“hack”) 是已知的溢出漏洞。黑客可如下利用 eval 功能来混淆溢出漏洞：

[0060] var X = ″ ha″ ;

[0061] var Y = ″ k″ ;

[0062] var Z = ″ doc″ ;

[0063] var A = ″ ument.wr″ ;

[0064] var B = ″ ite(″ ;

[0065] var C = ″)″ ;

[0066] var D = ″ c″ ;

[0067] eval(Z+A+B+X+D+Y+C) ;

[0068] 上述代码会将字符串评价为 document.write(“hack”)。document.write(“hack”) 会被运行这一点，直到实际运行时间才会变得明显。如利用常规系统，此时防止攻击已经太迟。类似地，执行传递给 eval 功能的参数的“预评价”将是无效的，且会不必要地减慢程序的运行。

[0069] 然而，在本发明的该实施例中，攻击在不昂贵地（就数据处理和网络资源而言）预评价响应时被检测到。具体地，eval 功能被允许运行并执行 document.write(“hack”)。然而，防护，而不是浏览器功能被执行。随后防护将 document.write(在这种情况下为“hack”) 的参数识别，识别出攻击，以及在恶意代码可被运行前终止执行。

[0070] 在另一示例实施例中，防护可以修改内容以在阻挠攻击的同时，仍旧允许内容继续执行，而不是在执行恶意代码前终止内容的执行。仅仅举例来说，许多攻击依赖特定的时机和系统运行的知识来实施攻击。一个这样的已知攻击为“heap spray”攻击，其分配大的内存区块，并在具体的位置设置特定的值。heap spray 攻击依赖于以堆形式的内存，该堆以特定的方式分配给功能。然而，如果防护确定某种功能调用或功能调用组代表 heap spray 攻击，防护例如可通过插入分配内存的延迟以修改堆的结构来修改内容的执行。这将修改攻击所依赖的结构，并将阻挠攻击。本领域普通技术人员会认识到尽管内容的这种修改可适于阻挠 heap spray 攻击，可适于阻挠其他依赖于时机或内存分配的类似的攻击的其他修改也位于本发明的范围内。

[0071] 在另一实施例中，防护引擎 12 可进一步分析请求并根据请求的特定方面有选择

性地预设防护。在另一个实施例中，防护引擎 112 对终端用户 112 的操作环境，诸如操作系统、作出请求的浏览器的类型和版本、以及可能出现的其他应用程序的有关信息的请求进行检查。例如，请求通常携带识别请求来自的浏览器的类型和版本、终端用户 102 的计算机的操作系统的元数据，并也可识别其他在终端用户 102 的计算机上允许的软件。利用该信息，防护引擎 112 可制定为请求系统预设的防护。例如，网络浏览器开发商经常更新它们的浏览器以便消除潜在的溢出漏洞。如果已知特定的浏览器版本已经纠正了特定的溢出漏洞，可不必预设针对那个溢出漏洞的防护。类似地，特定的溢出漏洞可针对执行除了网络浏览器之外的其他应用程序的代码。如果防护引擎 112 从请求中确定出特定溢出漏洞的对象的程序没有被安装在终端用户 102 的计算机上，那么它可确定不预设针对那个溢出漏洞的防护。

[0072] 在本发明的另一个示例实施例中，防护引擎 112 可在预设防护之前考虑外部数据。例如，防护引擎 112 可确定远程服务器 108 是否位于白名单上。通常，白名单为已知对网络通讯安全的服务器数据库（由名字、互联网地址或其他身份指示符）。如果远程服务器 108 在白名单上，不预设防护或可改善其性能。

[0073] 在本发明的示例实施例中，防护引擎 112 把防护预设于响应中的可执行代码前，这意味着防护将由浏览器在任何可被浏览器执行的可执行代码之前执行。最常见的内容包括能用于识别于该内容内可执行代码的开头部分的位置的标签。例如，JavaScript 代码的开头部分通常利用 <script> 标签识别。根据示例实施例，JavaScript 防护可紧接在 <script> 标签之后，但在标签内的可执行代码之前预设。在另一示例实施例中，当在要被执行的代码中的功能可利用不必在内容中的代码之前出现的防护覆盖时，那么，防护可在任何适当的位置被添加到代码中，该位置由内容和可执行代码的特定类型规定。

[0074] 除了在潜在的恶意功能调用发生之前修改或终止内容的执行之外，防护也可包括警告功能，其可将恶意代码的存在通知一个或多个感兴趣的当事人。在本发明的示例实施例中，警报会发送通知给特定网络的管理员或其他制定的第三方。该警报可包含与攻击有关的信息，其可帮助网络管理员确定行动的正当步骤，仅仅举例来说，其包括，识别出攻击、攻击的类型、内嵌有攻击的代码、请求最终包含有攻击的内容的计算机的身份、传送攻击的远程服务器 108 的身份、以及其他可以任何可想到的矫正措施来帮助警报接收者的信息的防护。在另一示例实施例中，警报可通过出现在终端用户 102 的计算机上的通知的形式发送给终端用户 102。在另一个实施例中，警报可发送给第三方，其可以各种方式利用与警报有关的信息以便提供更有效的服务。仅仅举例来说，在确定远程服务器 108 的地址是否应该被添加到有害的发送者名单（有时称为黑名单或危险列表）中时，第三方可利用关于远程服务器 108 的身份的信息。第三方也可为防护的提供者，且如果这样的话，第三方可利用警报中的信息来改善它的防护。

[0075] 一旦为响应预设了防护，受保护的响应将被发送回代理服务器 104。受保护的响应于是被发送给终端用户 102，并于终端用户 102 的浏览器上执行。

[0076] 防护引擎 112 被耦连于防护服务器 114。防护服务器 114 被构造成可为防护引擎 112 提供更新的防护。安全分析员不断地注意新的溢出漏洞。在某些情况下，溢出漏洞由寻找软件中的缺陷以试图先于黑客行动的安全分析员或软件开发者发现。在其他情形中，黑客识别并利用该缺陷。无论那种情况，安全分析员一旦察觉到利用易受攻击的功能的溢出

漏洞，他们都能利用上述的防护覆盖这些功能。在示例实施例中，覆盖功能包括，检查黑客可用来利用溢出漏洞的方法，以及产生代码以便检查这些方法的证据的属性。包装器也可包括检查状态信息的代码，该状态信息例如为利用特定参数，特定功能已被调用的次数，并可确定该状态信息是否代表攻击。更新的包装器被设置在防护服务器 114 上，并被定期地加载或推送到防护引擎 112 上。

[0077] 现在参考图 7，其示出了根据本发明的第二示例实施例，描述实施用于运行时攻击预防的系统 700 的网络结构的框图。和图 1 的系统一样，图 7 的用于运行时预防的系统通过给响应预设防护来运转。图 7 的系统与图 1 的系统不同在于，不是在网络级预设防护，而是由远程服务器 108 预设防护。

[0078] 在图 7 中所描述的用于运行时攻击预防的系统 700 的示例实施例中，终端用户 702 产生对内容的请求。请求被发送给终端用户 702 的计算机被连接到的局域网 (“LAN”) 704。请求于是通过互联网 706 被递送给远程服务器 708，其被指定为响应该请求。通常，如果请求为对互联网 / 万维网内容的请求，则响应为一个或多个构成网页的文本或数据包。在本发明的示例实施例中，在发送响应之前，远程服务器 708 为响应预设防护。远程服务器 708 随后通过互联网 706 和 LAN 704 将受保护的响应发送回终端用户 702。正如图 1 的系统中一样，终端用户 702 的浏览器接收受保护的响应并执行防护。

[0079] 在该实施例中，通过在将响应发送给终端用户 702 之前，为响应预设防护，远程服务器 708 得到不会不慎向连接到服务器的终端用户 702 提供攻击的保证。通过示例的方式，许多商业网站在网站上包括提供给访问者的广告。通常，商业网站几乎不直接控制广告的内容。反而，商业网站仅仅从互联网上的其他位置下载包括广告的内容。如果那个内容含有恶意代码，远程服务器 708 直到终端用户 702 抱怨来自于由远程服务器 708 提供的网站的攻击前，可能不会有任何指示，。然而，通过预设防护，不管终端用户 702 是否与可能也提供防护的系统（如图 1 和 9 中描述的系统）相关联，远程服务器 708 也已对终端用户 702 作出防护以免其终端用户 702 受远程服务器可能不慎发送的任何攻击。

[0080] 远程服务器 708 被耦连到防护服务器 710。在示例实施例中，防护服务器 710 和图 1 所描述的防护服务器 114 类似，并为远程服务器 708 提供更新的防护。

[0081] 现在参考图 9，其示出了根据本发明的第三示例实施例，描述实施用于运行时攻击预防的系统 900 的网络结构的框图。和图 1 和 7 所示的系统一样，图 9 所示的用于运行时攻击预防的系统通过为响应预设防护来操作。图 9 的系统与图 1 和 7 的系统的区别在于，防护由终端用户 702 的计算机，而不是在网络级预设。

[0082] 在示例实施例中，终端用户 902 的计算机发送请求，其穿过 LAN 904 和互联网 906 传播到远程服务器 908。远程服务器 908 将响应穿过互联网 906 和 LAN 904 发送回给终端用户 902 的计算机。终端用户 902 的计算机一经接收到响应，会在将响应传递给网络浏览器之前，为响应预设防护。受保护的响应于是被允许传递给终端用户 902 的浏览器并在那里被执行。

[0083] 终端用户 902 的计算机可被耦连于提供更新的防护的防护服务器 910。在示例实施例中，终端用户 902 的计算机经由互联网 906 并通过耦连于终端用户 902 的计算机的 LAN 904 耦连于防护服务器 910。在另一示例实施例中，防护服务器 910 对于终端用户 902 的计算机可为局部的，或者可位于耦连于终端用户 902 的计算机的 LAN 内。

[0084] 在图 9 中所阐释的系统 900 的另一示例实施例中，防护除了可被预设到从远程服务器 908 接收的响应之外，也被预设到可为可执行的代码。仅仅通过示例的方式，防护本身可为可执行的二进制代码，并且可被预设于现有的可执行程序中。

[0085] 在另一个可替换示例实施例中，防护不是（或者除了）在可执行代码中预设覆盖功能，而是可被预设成可修改可执行代码曾接近系统层级的功能的入口。例如，在 Linux 操作系统中，特定的系统性能仅仅可由已被授予特定许可的用户或进程接近。即使进程开始了已接近所有可获得的功能的执行（通常称为“根”或“超级用户”通路），随着执行的进行，进程可程式化地放弃特定的许可。一旦进程已经放弃了许可，就不可能要求恢复那个许可。因此，在任何包含于代码中的恶意代码可被执行之前，可修改可执行代码到系统的入口的防护可被预设到该代码，而且要求恢复其，从系统的角度，曾自愿放弃的许可将被阻止。

[0086] 例如，进程必须具有特定的一组许可以便能接近表示输入 / 输出端口和系统内存的内存地址。如果有的话，很少的合法进程需要这样的入口。因此，可为迫使进程放弃该入口的可执行代码预设防护，由此抢先防止恶意代码操纵这些资源。

[0087] 类似地，也可在进程可获得的系统资源方面限制进程。例如，除了别的以外，进程也可被限于接近某最大尺寸的文件、某数量的内存或处理器工作量、某数量的打开文件、某数量的磁盘用量以及某数量的子进程。一些攻击的标志，例如拒绝服务攻击，为占用系统资源的多个子进程的大量生成，其阻止系统响应合法的请求。可为可执行代码预设防护，以限制进程可创建或接近的子进程（或其他系统资源）数量，从而减小这种攻击将发生的可能性。

[0088] 现在参考图 2，其为描述，利用图 1，7 和 9 的运行时攻击预防的系统，在网络通讯中插入防护的示例方法 200 的流程图，本发明的附加特征将被描述。图 2 将与图 1 相关连地论述。该方法突出用于运行时攻击预防的系统的关键功能特征。

[0089] 本领域普通技术人员会理解到，用于运行时攻击预防的系统 100 执行的进程功能或步骤可包括程控的通用计算机、形成电子器件的电子电路、在微控制器或微处理器上执行的固件代码；以特定应用程序或可编程逻辑形式执行的状态机；或者不脱离本发明的精神和范围的多个其他形式。换句话说，本发明可被设置成可包括机器可读介质的计算机程序，该机器可读介质已在其上存储有用于使计算机（或其他电子器件）编程以便执行根据本发明的进程的指令。

[0090] 机器可读介质可包括但不限于，软盘、光盘、CD-ROM，以及磁光盘、ROMs、RAMs、EPROMs、EEPROMs、磁或光卡、闪存、或适于储存电子指令的其他类型的介质 / 机器可读介质。

[0091] 所有下面所参考的逻辑流程图中所描述的方法中的某些步骤必须自然地处于其他步骤之前以便本发明如所描述的起作用。然而，如果这些的顺序或次序不会改变本发明的功能的话，本发明并不限于所描述的步骤的顺序。也就是说，某些步骤可在其他步骤之前、之后或与其他步骤并行执行而不脱离本发明的范围和精神。

[0092] 此外，某些步骤可以不同的次序重新设置或被完全删除，而不脱离本发明的范围和精神。换句话说，在流程图中所述的步骤仅代表实现防护网络的期望结构的一种方式。其他方式或可包括附加的不同的步骤或步骤的消除、或者消除步骤和添加对于本领域普通技术人员显而易见的不同步骤的结合。

[0093] 进一步，基于流程图和申请文本中的相关说明，从事编程的普通技术人员将能轻易地编写这样的计算机程序或，例如，识别适当的硬件电路来实施所公开的发明。因此，特定的程序代码指令组或详细的硬件器件的公开对于适当的理解如何制造和使用本发明是并不被认为是必需的。在结合阐述其他进程流程的剩余附图的随后说明中，要求保护的实施进程的计算机的独创的功能将被更详细的阐明。

[0094] 再一次参考图 2，步骤 205 为创建防护的方法 200 的第一步。创建防护的过程将透过图 3 进一步详细论述。判断步骤 210 确定防护是否要被插入在网络处。如果防护要被插入在网络处，那么方法 200 沿着“是”分支行进到步骤 215，其中，防护被插入在网络处。将防护插入在网络处的过程将透过图 4 进一步详细的论述。方法 200 于是前进到盘道步骤 220。另一方面，如果在判断步骤 210 确定防护将不被插入在网络处，那么方法 200 沿着“否”分支行进到判断步骤 220。

[0095] 判断步骤 220 确定防护是否要被插入在远程服务器 108 处。如果防护要被插入在远程服务器处 108，那么沿着“是”分支行进到步骤 225，其中，防护被插入在远程服务器 108 处。在远程服务器 108 处插入防护的过程将透过图 8 进一步详细讨论。方法 200 于是前进到判断步骤 230。另一方面，如果判断步骤 220 确定防护将不被插入在远程服务器 108 处，那么沿着“否”分支行进到判断步骤 230。

[0096] 判断步骤 230 确定防护是否被插入在终端用户 102 的计算机处。如果防护被插入在终端用户 102 的计算机处，那么方法 200 沿着“是”分支行进到步骤 235，其中，防护被插入在终端用户 102 处。在终端用户 102 处插入防护的过程将透过图 10 进一步详细讨论。方法 200 于是前进到判断步骤 240。另一方面，如果判断步骤 230 确定防护将不被插入在终端用户 102 处，那么方法 200 沿着“否”分支行进到判断步骤 240。

[0097] 判断步骤 240 确定系统是否要继续装上防护。如果确定系统将继续装上防护，那么方法 200 沿着“是”分支行进到步骤 205。另一方面，如果确定系统不继续装上防护，那么方法 200 沿着“否”分支行进，以结束方法 200。

[0098] 现在参考图 3，其描述了利用图 1 的用于运行时攻击防护的系统创建防护的方法 205。图 3 将与图 1 和 2 相关连地讨论。步骤 300 为方法 205 中的第一步，在其中识别易受攻击的方法。如上所述，在本发明的示例实施例中，易受攻击的方法可由安全分析员和找出并识别软件中的漏洞的黑客来识别出。漏洞也可通过阅读程序中的漏洞的发表报告、逆向开发已经发生的攻击、使程序“模糊不清”或对程序“模糊测试”、以及研究程序制造商已发布的程序的补丁、修复或其他更新来识别，使程序“模糊不请”或对程序“模糊测试”通常涉及将大量随机或其他的无效数据传递给程序并确定所述数据是否导致失效，而研究程序制造商已发布的程序的补丁、修复或其他更新是因为，程序制造商已发布的程序的补丁、修复或其他更新通常矫正程序的较早版本中的漏洞并因此可用于识别这些漏洞。

[0099] 方法 205 随后前进到步骤 305，步骤 305 编写代码以便覆盖易受攻击的方法且必要时以便修改内容的执行。方法 205 随后前进到步骤 310，在该步骤中产生代码以检查用于已知攻击的被覆盖的易受攻击的方法的属性。方法 205 随后前进到步骤 315，在该步骤中产生代码来检查状态信息以便识别攻击。举例来说，代码可被产生以便计算 NOP 功能被调用的次数，且如果 NOP 的数量超过某个预定的阈值，识别出潜在的攻击。方法 205 随后前进到步骤 320，在该步骤中产生代码以警告终端用户 102、网络供应商和 / 或诸如第三方的其他相

关当事人可利用与警报有关的信息来矫正现有的攻击并利用改进的防护来抵御将来的攻击。方法 205 随后前进到图 2 的步骤 210。

[0100] 现在参考图 4，其描述了利用图 1 的用于运行时攻击预防的系统在网络处插入防护的方法 215。图 4 将与图 1 和 2 相关连地讨论。步骤 400 为方法 215 中的第一步，在其中，终端用户 102 请求给予内容。如上所述，在示例实施例中，请求来源于安装在终端用户 102 的计算机上的网络浏览器，然而在另一实施例中，请求可为向远程服务器 108 请求最终将在终端用户 102 的计算机上执行的内容的任何请求。方法 215 随后前进到步骤 405，其中网络接收内容。随后方法 215 前进到步骤 410，在该步骤中为响应预设防护。在该示例实施例中，防护引擎 112 被耦连于安装在网络上的代理服务器 104，其拦截响应并如上述透过图 1 所描述的，在其上添加或预设防护。预设的防护将透过图 5 进一步详细讨论。

[0101] 方法 215 随后前进到步骤 415，其中受保护的内容被传递给终端用户 102。在示例实施例中，防护引擎 112 将受保护的内容发送回代理服务器 104（或发送回网络上拦截响应的位置），代理服务器 104 将响应发送回终端用户 102。随后方法 215 前进到步骤 420，其中受保护的内容在终端用户 102 的计算机上被执行。在示例实施例中，受保护的内容在终端用户 102 的网络浏览器内被执行。然而，在另一示例实施例中，受保护的内容可在浏览器之外被执行。

[0102] 方法 215 随后前进到步骤 425，其中防护引擎 112 更新它的防护。在示例实施例中，防护引擎 112 从防护服务器 114 接收更新过的防护。随后方法 215 前进到判断步骤 430，在步骤中确定是否继续装上防护。如果系统被确定应该继续装上防护，那么沿着“是”分支返回到步骤 400。另一方面，如果系统被确定不继续装上防护，那么沿着“否”分支行进，方法 215 回到图 2 的步骤 220。

[0103] 现在参考图 5，示出了利用图 1，7 和 9 的用于运行时攻击预防的系统预设防护的方法 410/810/1015。图 5 将与图 1，4，8 和 10 相关连地讨论。判断步骤 500 为方法 410/810/1015 中的第一步，方法 410/810/1015 在该步骤确定所接收的内容是否来自于可信源。仅仅举例来说，如果内容的源地址出现在已知可靠的发送者（通常称为白名单）的数据库中，内容可被认定成来自于可信源。如果确定内容接收自可信源，那么在一个示例实施例中，沿着“是”分支行进，方法 410/810/1015 回到步骤 415/815/1020。在另一例实施例中，系统可被构造成忽视与内容的来源有关的信息，并可不顾来源而预设防护。

[0104] 再一次回到判断步骤 500，如果内容被确定不是来自于可信源，那么沿着“否”分支行进到步骤 505，以确定所接收的内容的类型。仅仅举例来说，步骤 505 确定所接收到的内容是否包含任何潜在地可被应用在攻击中的可执行代码，诸如 JavaScript、Flash 或 Sliverlight。随后方法 410/810/1015 前进到步骤 510，以识别相应于在步骤 505 中识别出的内容类型的防护。在某种情形中，特定的防护仅可应用于可执行内容的特定形式。因此，其可改善性能以仅仅识别可应用于即将到来的内容的防护。

[0105] 随后方法 410/810/1015 前进到判断步骤 515，其中，防护引擎 112 确定其是否了解终端用户的环境。在示例实施例中，防护引擎 112 从来自于对内容的请求取得与终端用户有关的信息。另一方面，防护引擎 112 可储存与在网络上的终端用户 102 的构造有关的信息，其包括终端用户的环境信息。如果确定防护引擎 112 不了解终端用户 102 的环境，那么沿着“否”分支行进到步骤 525，以为内容预设在步骤 510 中识别出的防护。方法

410/810/1015 随后回到步骤 415/815/1020。

[0106] 再次回到判断步骤 515，如果确定防护引擎 112 了解终端用户 102 的环境，那么方法 410/810/1015 沿着“是”分支行进到步骤 520，在该步骤识别与终端用户 102 的环境相应的防护。例如，如果终端用户 102 的环境包括作为网络浏览器的 Google Chrome，那么不必预设针对抵御仅仅存在于微软网络浏览器第 6 版的溢出漏洞的防护。在示例实施例中，相应于终端用户 102 的环境的防护从相应于内容类型的防护组中选择。在另一实施例中，相应于终端用户 102 的环境的防护可单独从相应于内容类型的防护中选择。随后方法 410/810/1015 前进到步骤 525，在该步骤为内容预设所有识别出的防护。随后方法 410/810/1015 回到步骤 415/815/1020。

[0107] 现在转到图 6，示出了利用图 1，7 和 9 的用于运行时攻击预防的系统的执行受保护的内容的方法 420/820/1020。图 6 将与图 1，4，8 和 10 相关连地描述。步骤 600 为方法 420/820/1020 的第一步，在其中，提供给内容的系统入口被修改。随后方法 420/820/1020 前进到步骤 605，以执行功能封装器。在示例实施例中，当可执行代码调用受保护的功能（即，已经被创建了防护的功能）时，功能封装器便会被执行。随后功能 420/820/1020 前进到步骤 610，而在该步骤就攻击方面分析传递给受保护的功能的参数。随后方法 420/820/1020 前进到步骤 615，及在该步骤分析状态信息。仅仅举例来说，状态信息可与在给定的浏览会话或时间段内功能曾被调用的次数、对受保护的功能和其他受保护的功能的调用之间的时间关系、或者可指示攻击的其他基于状态的信息相关。

[0108] 随后方法 420/820/1020 前进到步骤 620，在该步骤确定状态信息或参数是否指示存在潜在威胁。如果确定潜在的威胁并不存在，那么方法 420/820/1020 沿着“否”分支行进到步骤 625，在该步骤中，被覆盖的功能被允许正常执行。随后方法 420/820/1020 前进到判断步骤 630，在该步骤中确定内容是否包含更多要运行的代码。如果确定功能包含更多要运行的代码，那方法 420/820/1020 么沿着“是”分支行进，且前进到步骤 605。另一方面，如果内容不包含更多的要运行的代码，方法 420/820/1020 沿着“否”分支行进，且回到步骤 425/825/1025。

[0109] 再一次参考判断步骤 620，如果确定检测到威胁，那么方法 420/820/1020 沿着“是”分支行进到步骤 635，其中内容的执行被终止或修改。在本发明的示例实施例中，内容的执行在受保护的功能不被执行的情况下终止。在另一示例实施例中，内容的执行可被修改以便阻挠攻击而不终止该内容的执行。方法 420/820/1020 随后前进到步骤 640，在其中播送警报。多个警报可被播送，且警报可被播送向多个接收者。在示例实施例中，警报被播送到网络管理员，其可确定如何响应警报。在另一示例实施例中，警报被播送到终端用户 102。在另一个示例实施例中，警报被播送到向第三方。方法 420/820/1020 随后回到步骤。

[0110] 现在转到图 8，其示出了描述利用图 7 的用于运行时攻击预防的系统 700 在服务器处插入防护的方法 225 的流程图。图 8 将与图 7 相关连地讨论。步骤 800 为方法 225 的第一步，在其中，远程服务器 708 接收对内容的请求。随后方法 225 前进到步骤 805，且在该步骤产生对请求的响应。随后方法 225 前进到步骤 810，且在该步骤为响应预设防护。上文已透过图 5 描述了为响应预设防护的过程。

[0111] 方法 225 随后前进到步骤 815，其中，受保护的响应经由互联网传递给终端用户 702。随后方法 225 前进到步骤 820，在该步骤中，受保护的内容被执行。执行受保护的内容

的过程将透过图 6 进一步详细的描述。随后方法 225 前进到步骤 825，在其中防护被更新。随后方法 225 前进到判断步骤 830，在该步骤确定系统 700 应否继续装上防护。如果确定系统 700 应继续装上防护，那么方法 225 沿着“是”分支行进到步骤 800。另一方面，如果确定系统应停止装上防护，那么方法 225 沿着“否”分支行进，且回到步骤 230。

[0112] 现在转到图 10，示出了描述利用图 9 的用于运行时攻击预防的系统 900 在终端用户 902 的计算机处插入防护的方法 235 的流程图。图 10 将与图 9 相关连地讨论。步骤 1000 为方法 235 中的第一步，其中，防护被下载到终端用户 902 的计算机上。随后方法 235 前进到步骤 1005，其中，终端用户 902 的计算机请求来自于远程服务器 908 的内容。随后方法 235 前进到步骤 1010，其中终端用户 902 的计算机接收内容。随后方法 235 前进到步骤 1015，其中终端用户 902 的计算机为内容预设防护。预设防护的方法透过图 5 以更详细的方式描述。随后方法 235 前进到步骤 1020，并在该步骤执行受保护的内容。本文透过图 6 以补充的细节的方式描述了用于执行受保护的内容的过程。方法 235 随后前进到步骤 1025，其中更新被保护。随后方法 235 前进到判断步骤 1030，并在该步骤中确定是否请求额外的内容。如果确定请求额外的内容，那么方法 235 沿着“是”分支行进到步骤 1010。另一方面，如果在步骤 630 确定没有请求额外的内容，那么方法 235 沿着“否”分支行进，且回到步骤 240。

[0113] 用于运行时攻击预防的系统和方法的另一些实施例对于本领域普通技术人员将变得显而易见，本发明与其相关且并不脱离该公开的精神和范围。因此，虽然以示例的方式利用某种程度的细节已描述了本发明，应理解，本文仅仅以示例的方式作出本发明的公开，并可采取构造的细节中和部件或步骤的组合及设置中的各种变化，而不脱离本发明的精神和范围。因此，本发明的范围由附属的权利要求所限制而不是前述说明。

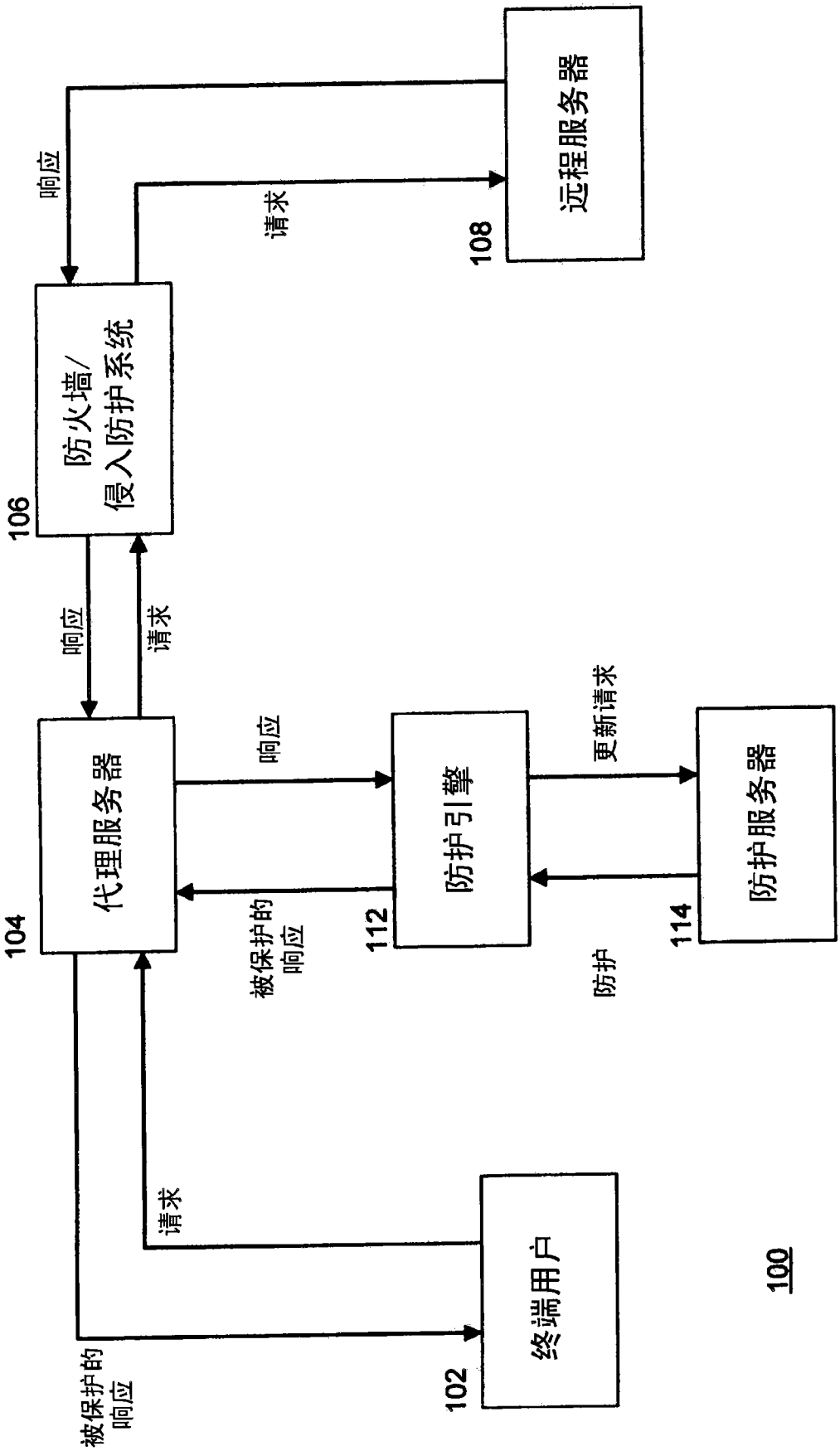

远程服务器
108
响应
请求
防火墙/
侵入防护系统
106
响应
请求
代理服务器
104
响应
被保护的
响应
防护引擎
112
更新请求
防护
防护服务器
114
请求
被保护的
响应
终端用户
102
100

图 1

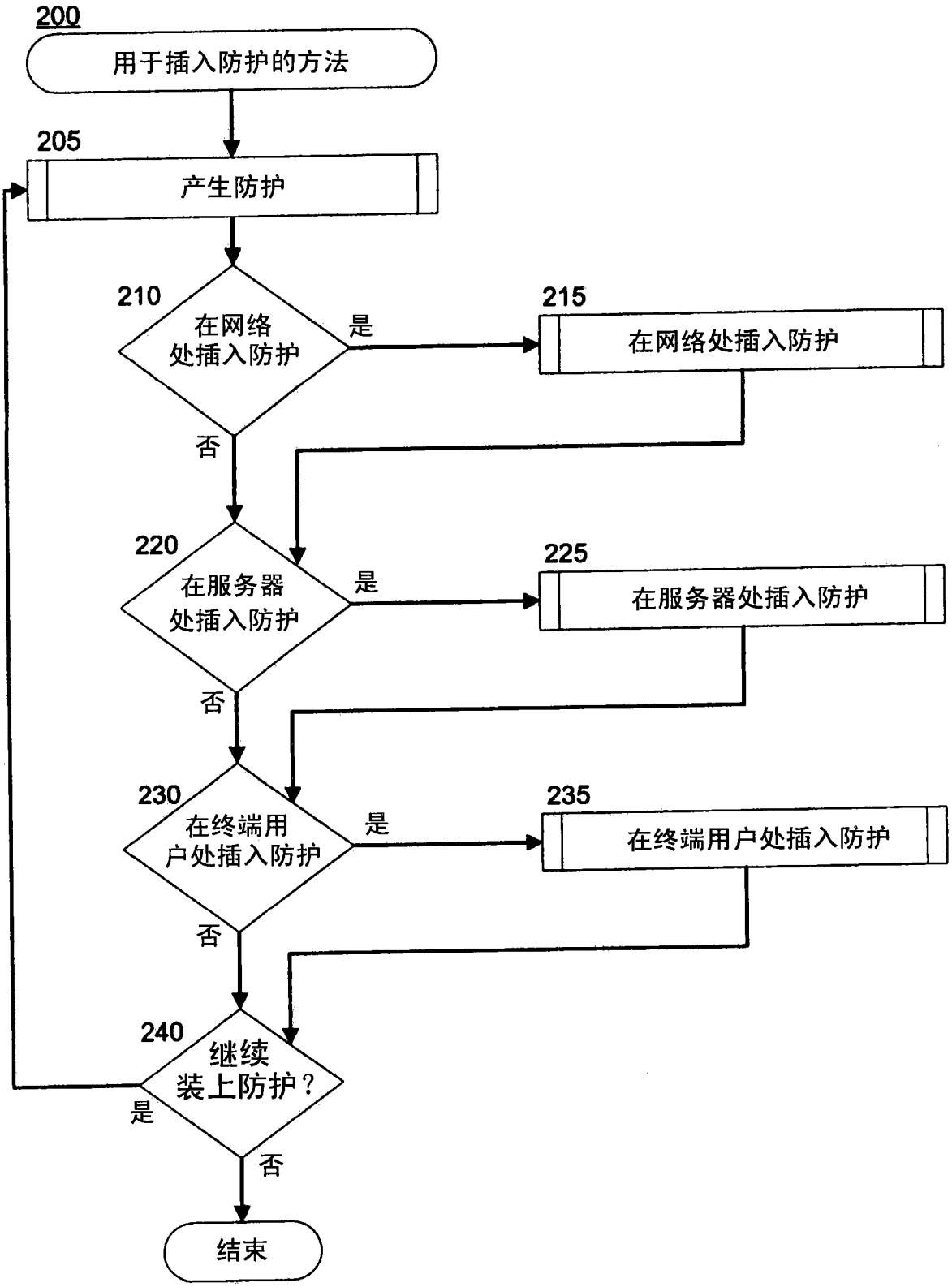

200
用于插入防护的方法
205
产生防护
210
在网络处插入防护
是
215
在网络处插入防护
否
220
在服务器处插入防护
是
225
在服务器处插入防护
否
230
在终端用户处插入防护
是
235
在终端用户处插入防护
否
240
继续装上防护？
是
否
结束

图 2

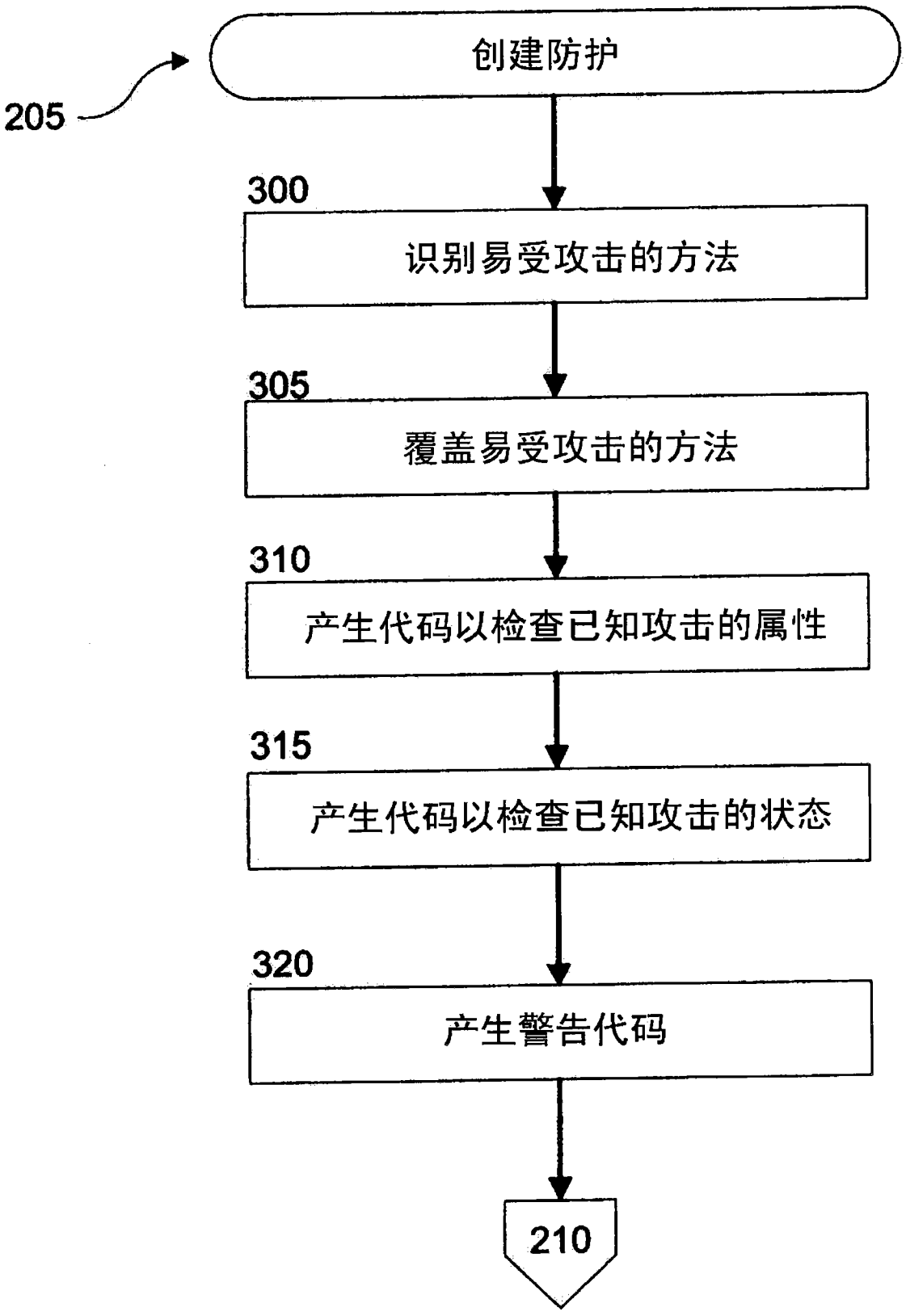

创建防护
205
300
识别易受攻击的方法
305
覆盖易受攻击的方法
310
产生代码以检查已知攻击的属性
315
产生代码以检查已知攻击的状态
320
产生警告代码
210

图 3

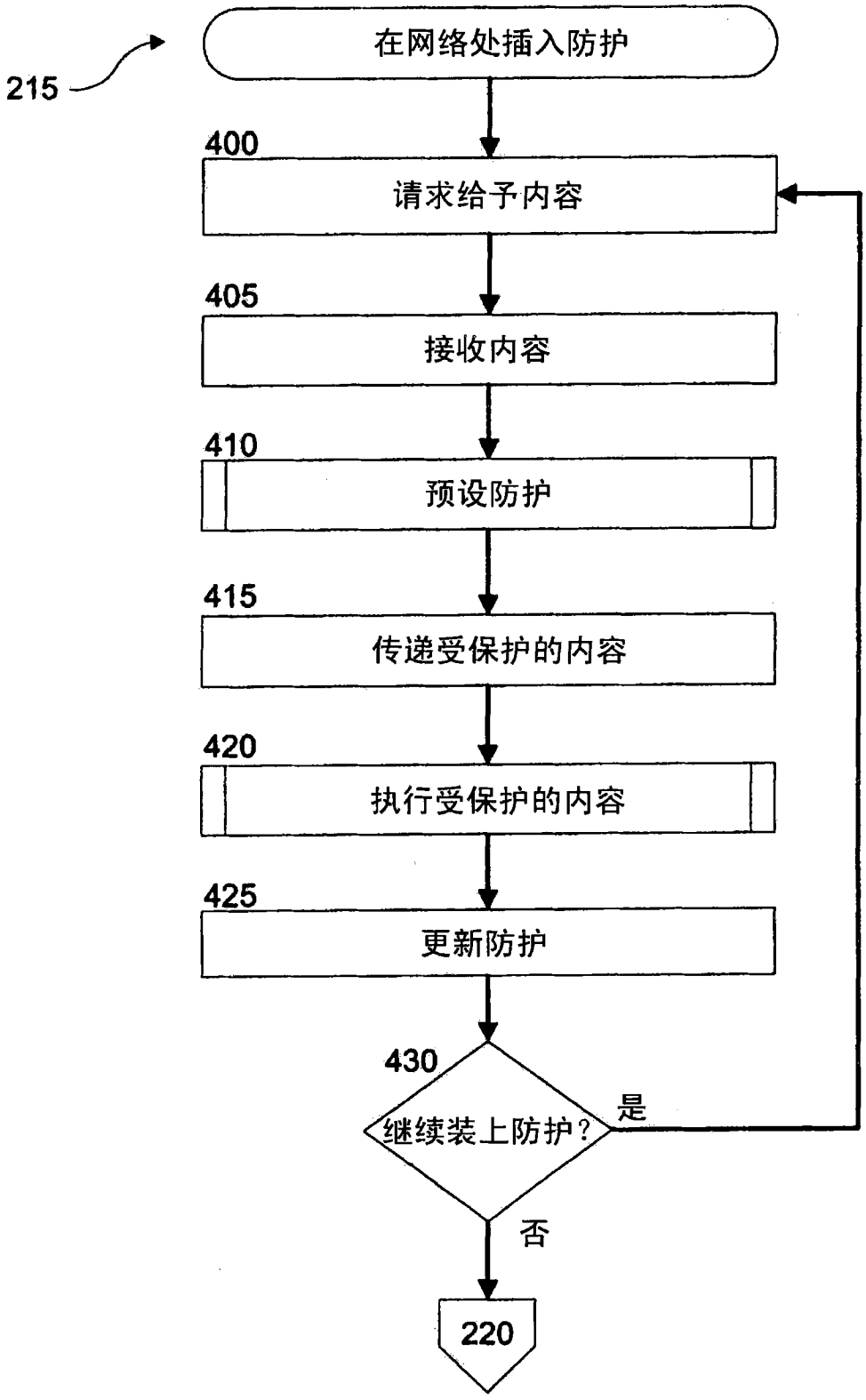

215
在网络处插入防护
400
请求给予内容
405
接收内容
410
预设防护
415
传递受保护的内容
420
执行受保护的内容
425
更新防护
430
继续装上防护？
是
否
220

图 4

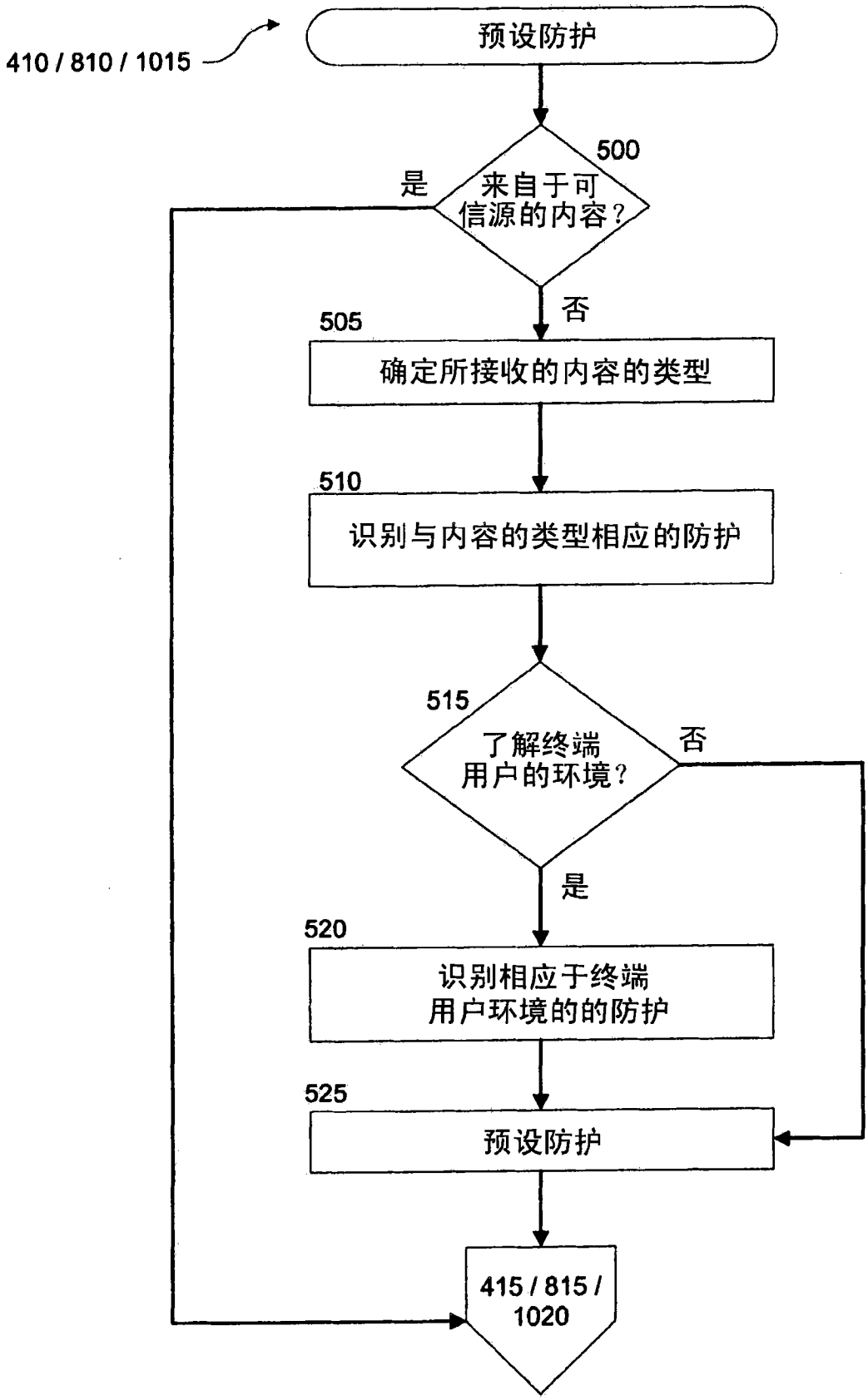

410 / 810 / 1015
预设防护
500
来自于可信源的内容？
是
否
505
确定所接收的内容的类型
510
识别与内容的类型相应的防护
515
了解终端用户的环境？
否
是
520
识别相应于终端用户环境的的防护
525
预设防护
415 / 815 / 1020

图 5

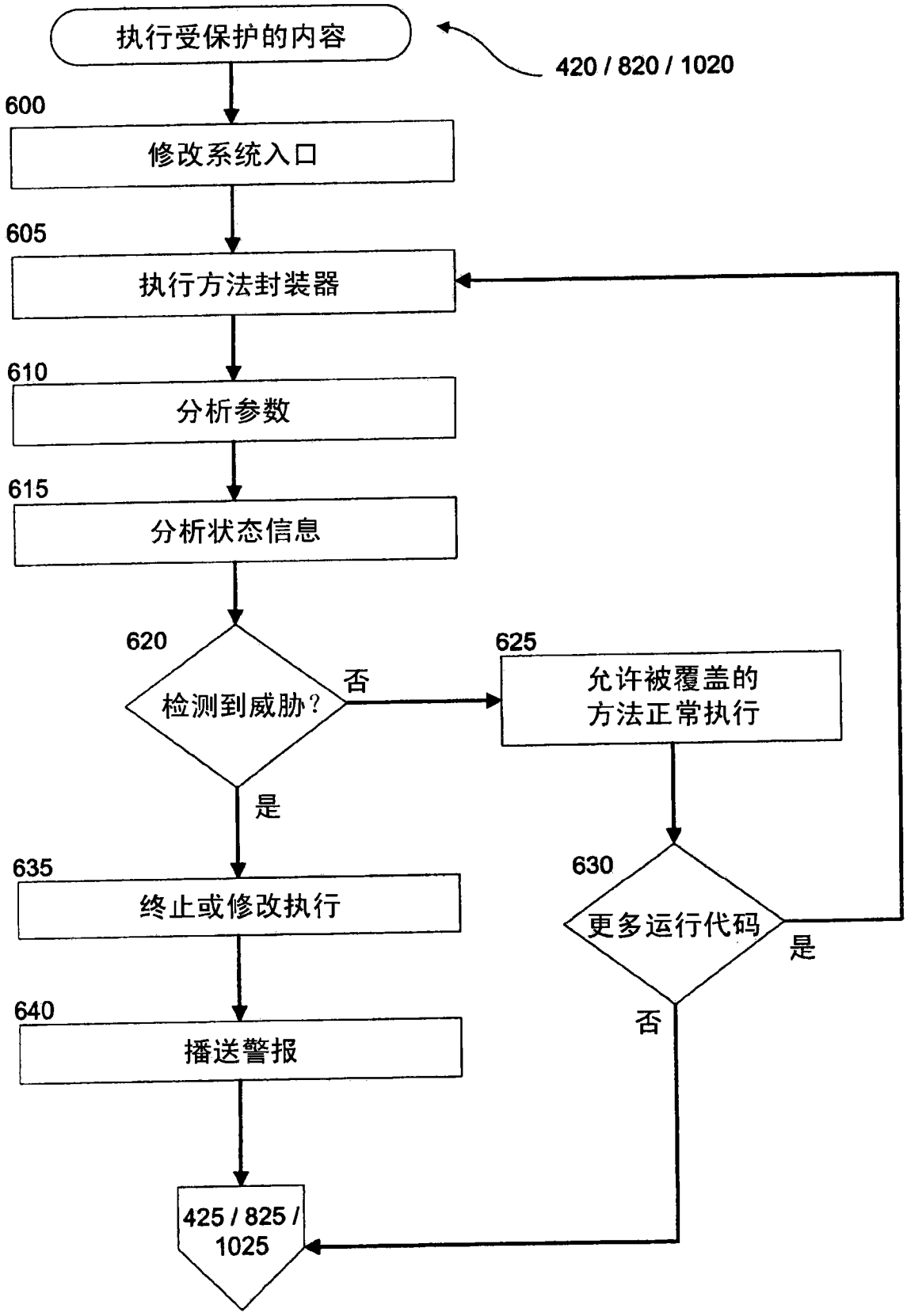

执行受保护的内容
420 / 820 / 1020
600
修改系统入口
605
执行方法封装器
610
分析参数
615
分析状态信息
620
检测到威胁？
否
625
允许被覆盖的
方法正常执行
是
635
终止或修改执行
630
更多运行代码
是
否
640
播送警报
425 / 825 /
1025

图 6

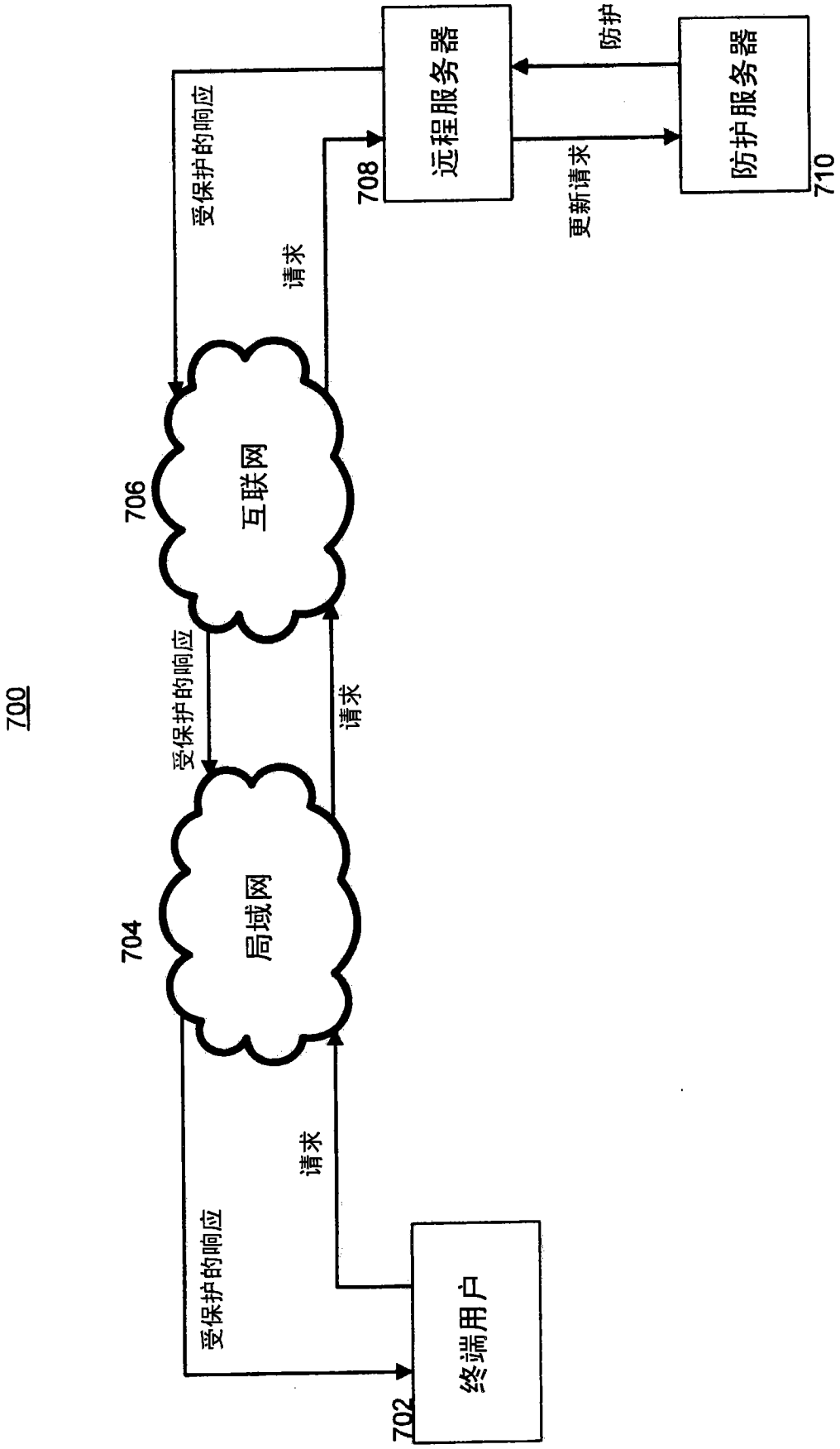

700
远程服务器
708
防护
防护服务器
710
更新请求
受保护的响应
请求
互联网
706
受保护的响应
请求
局域网
704
受保护的响应
请求
终端用户
702

图 7

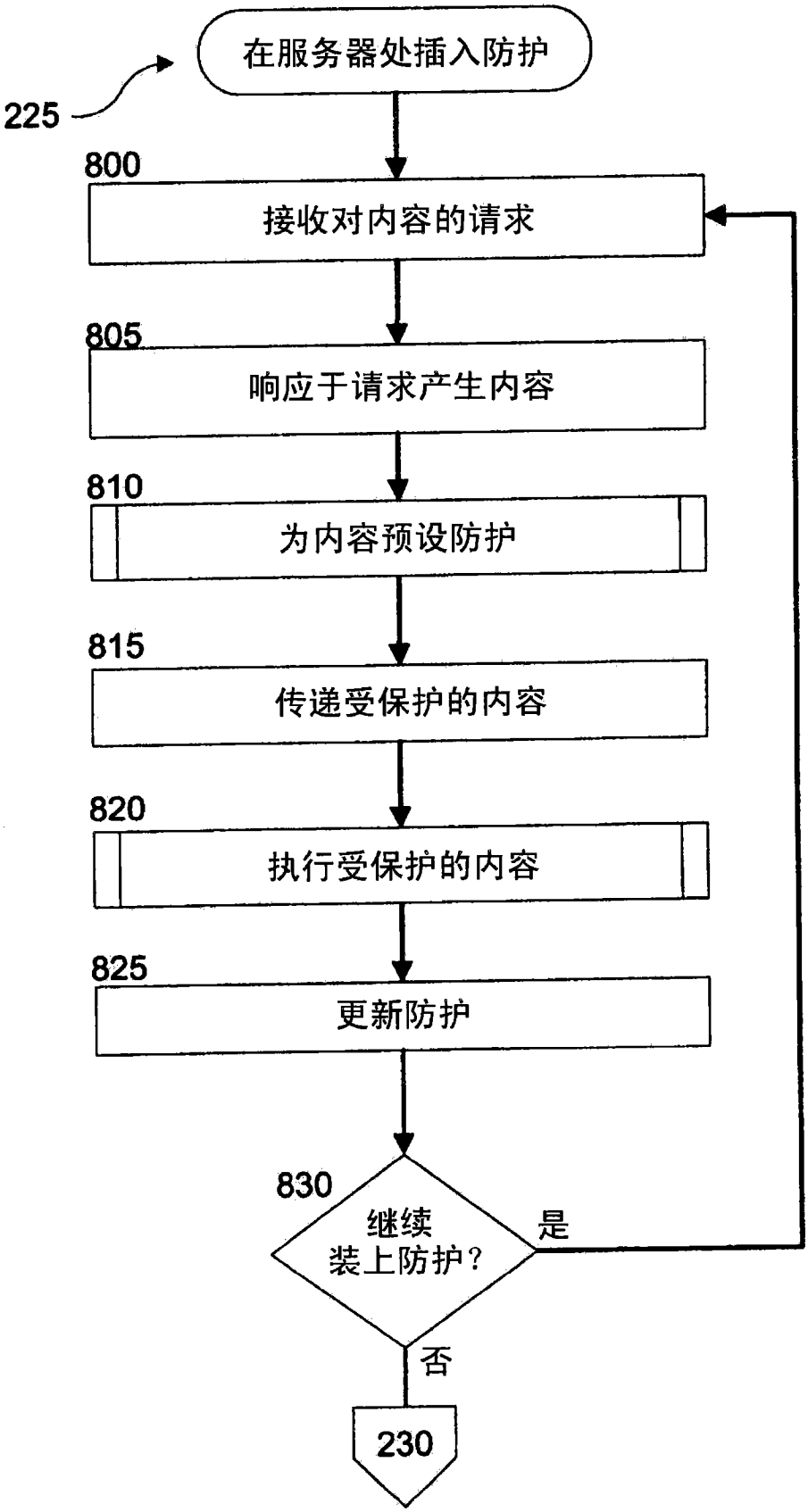

在服务器处插入防护
225
800
接收对内容的请求
805
响应于请求产生内容
810
为内容预设防护
815
传递受保护的内容
820
执行受保护的内容
825
更新防护
830
继续
装上防护？
是
否
230

图 8

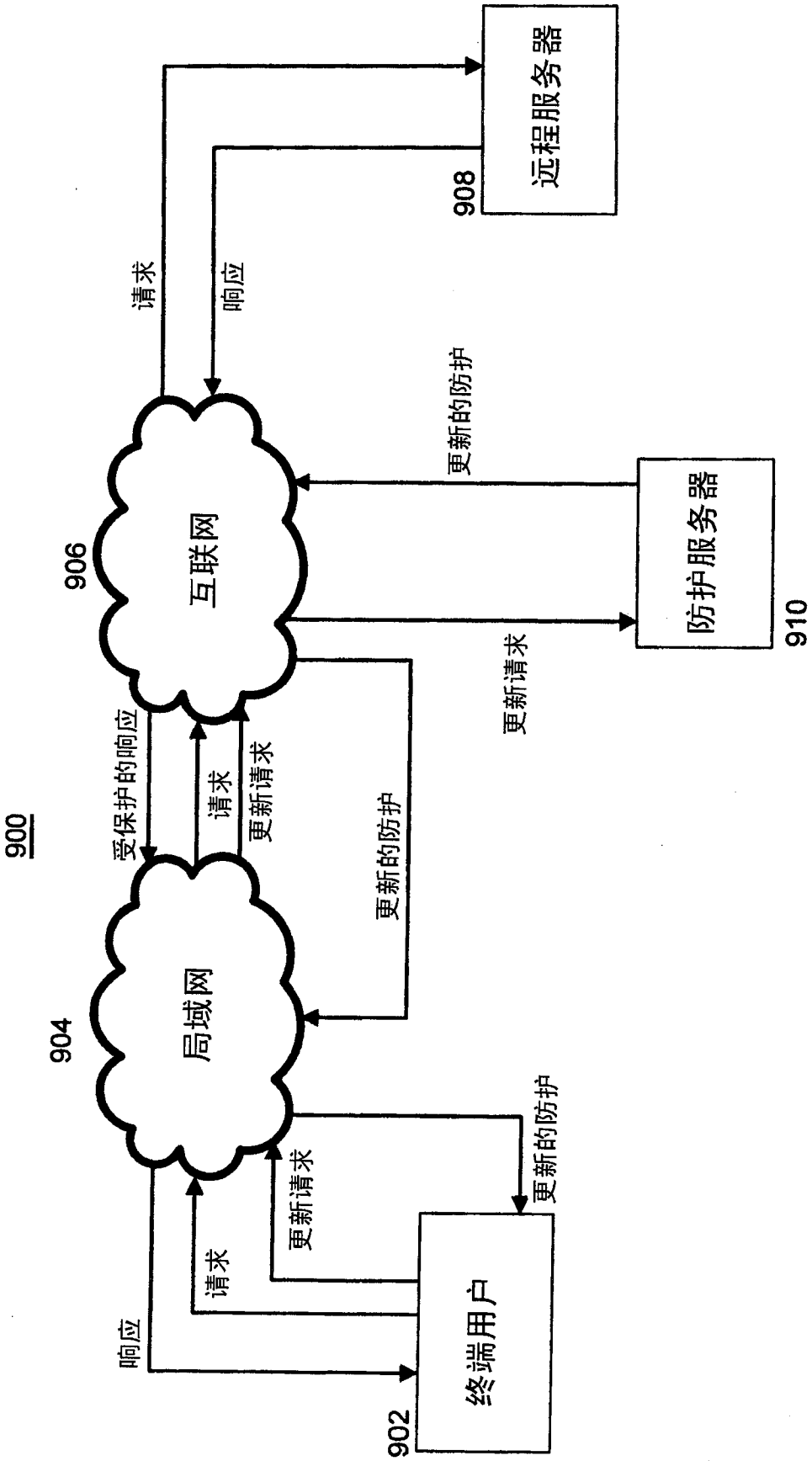

900
远程服务器
908
请求
响应
更新的防护
互联网
906
防护服务器
910
更新请求
受保护的响应
请求
更新请求
更新的防护
局域网
904
更新的防护
更新请求
请求
响应
终端用户
902

图 9

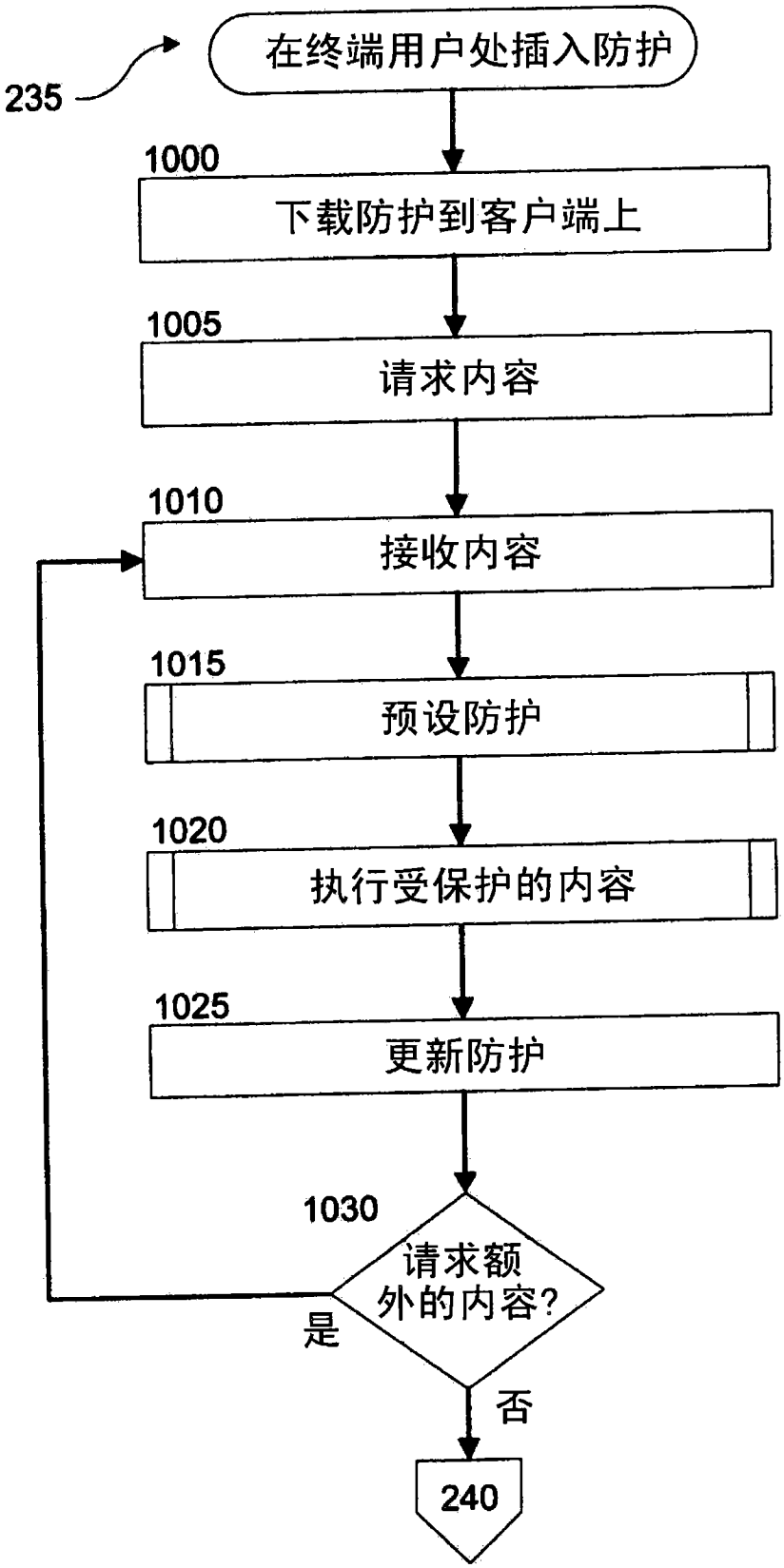

235
在终端用户处插入防护
1000
下载防护到客户端上
1005
请求内容
1010
接收内容
1015
预设防护
1020
执行受保护的内容
1025
更新防护
1030
请求额外的内容?
是
否
240

图 10