(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 103946855 B
(45)授权公告日 2017.03.08

(21)申请号 201280054578.4
(22)申请日 2012.11.07
(65)同一申请的已公布的文献号
申请公布号 CN 103946855 A
(43)申请公布日 2014.07.23
(30)优先权数据
13/290,932 2011.11.07 US
(85)PCT国际申请进入国家阶段日
2014.05.07
(86)PCT国际申请的申请数据
PCT/US2012/063953 2012.11.07
(87)PCT国际申请的公布数据
WO2013/070773 EN 2013.05.16
(73)专利权人 高通股份有限公司
地址 美国加利福尼亚州
(72)发明人 丹尼尔·科马罗米
亚历克斯·甘特曼
布莱恩·M·罗森贝格
阿伦·巴拉克里沙南 伦威·格
格雷戈里·G·罗丝
阿南德·帕拉尼古德
(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 宋献涛
(51)Int.Cl.
G06F 21/52(2006.01)
(56)对比文件
US 2010218251 A1,2010.08.26,
CN 102103483 A,2011.06.22,
WO 2006001574 A1,2006.01.05,
CN 1723448 A,2006.01.18,
审查员 郭少杰

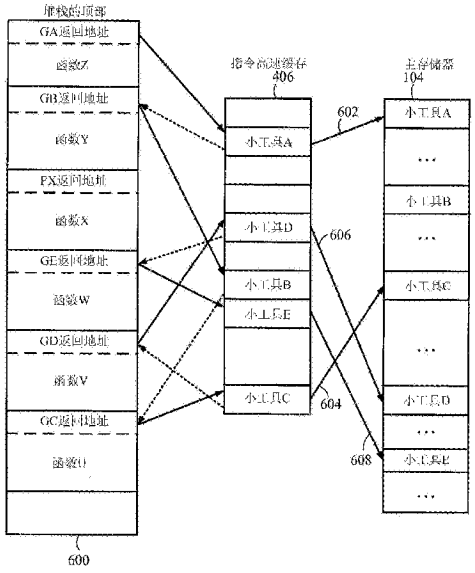
权利要求书4页 说明书13页 附图8页

(54)发明名称

用于检测面向返回编程攻击的方法、装置和系统

(57)摘要

本发明揭示用于检测面向返回编程ROP攻击的方法、装置和系统。系统包含处理器、主存储器和高速缓存存储器。高速缓存监视器通过监视对在所述高速缓存存储器中发现的经高速缓冲存储的指令的存取以及对当前不在所述高速缓存存储器中的指令的未中来形成指令加载简档。补救动作单元在所述指令加载简档指示涉及一或多个有效代码序列的ROP攻击的执行的情况下终止所述有效代码序列中的一或多者的执行。所述指令加载简档可为从相对于高速缓存未中监视高速缓存命中而导出的命中/未中比率。所述ROP攻击可包含若干代码片断，其各自包含来自有效代码序列的可执行指令和返回指令。



CN 103946855 B

1.一种在包含高速缓存存储器的处理电路中操作的方法,其包括:

在所述高速缓存存储器中加载可执行代码序列的至少若干部分;

执行从所述高速缓存存储器的所述可执行代码序列的指令获取;

响应于所述指令获取而检测一个或多个指令获取高速缓存未中,其中单个指令获取高速缓存未中代表针对所述可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中;

相对于所述一个或多个指令获取高速缓存未中监视所述指令获取以便动态地检测异常指令获取高速缓存未中活动;以及

确定所述异常指令获取高速缓存未中活动是否指示面向返回编程攻击。

2.根据权利要求1所述的方法,其进一步包括:

报告所述异常指令获取高速缓存未中活动,且响应于所述异常指令获取高速缓存未中活动而执行关于所述可执行代码序列的补救动作。

3.根据权利要求2所述的方法,其进一步包括:

通过响应于所述异常指令获取高速缓存未中活动的所述报告终止所述可执行代码序列的执行来执行所述补救动作。

4.根据权利要求1所述的方法,其中相对于所述一个或多个指令获取高速缓存未中监视所述指令获取包含

响应于所述指令获取而检测一个或多个指令获取高速缓存命中;

响应于所检测的指令获取高速缓存命中和指令获取高速缓存未中而确定指令获取高速缓存命中/未中比率;以及

通过检测命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率而选择的阈值来检测所述异常指令获取高速缓存未中活动。

5.根据权利要求4所述的方法,其中所述有效命中/未中比率是通过监视来自在所述监视期间不含面向返回编程攻击的多个所述可执行代码序列的指令获取高速缓存命中/未中比率来确定。

6.根据权利要求1所述的方法,其中所述监视在选定操作周期期间暂停。

7.根据权利要求6所述的方法,其中所述选定操作周期包含新过程的引导阶段。

8.根据权利要求1所述的方法,其中所述异常指令获取高速缓存未中活动是在预定义数目的先前指令或预定义时间周期上确定。

9.根据权利要求1所述的方法,其中所述异常指令获取高速缓存未中活动是在预定操作上下文内确定。

10.一种包含高速缓存存储器的处理装置,其包括:

处理电路,其经配置以获取和执行可执行代码序列;

高速缓存存储器系统,其可操作地耦合到所述处理电路且包含至少一个高速缓存存储器;

高速缓存监视器,其经配置以响应于指令获取而检测一个或多个指令获取高速缓存未中,其中单个指令获取高速缓存未中代表针对所述可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中,且进一步经配置以相对于所述一个或多个指令获取高速缓存未中监视指令获取以便动态地检测异常指令获取高速缓存未中活动;以及

补救动作单元，其经配置以确定所述异常指令获取高速缓存未中活动是否指示面向返回编程攻击。

11.根据权利要求10所述的处理装置，其中所述补救动作单元进一步经配置以在所述高速缓存监视器报告所述异常指令获取高速缓存未中活动的情况下终止所述可执行代码序列中的一或多者的执行。

12.根据权利要求11所述的处理装置，其中所述高速缓存监视器、所述补救动作单元、所述处理电路或其组合经配置以：

响应于所述指令获取而检测一个或多个指令获取高速缓存命中；

响应于所检测的指令获取高速缓存命中和指令获取高速缓存未中而确定指令获取高速缓存命中/未中比率；以及

通过检测命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率而选择的阈值来检测所述异常指令获取高速缓存未中活动。

13.根据权利要求10所述的处理装置，其中所述高速缓存存储器系统包含数据高速缓存和指令高速缓存，且所述高速缓存监视器经配置以从所述指令高速缓存形成指令加载简档。

14.根据权利要求10所述的处理装置，其中所述高速缓存存储器系统包含两级或两级以上高速缓存，且所述高速缓存监视器经配置以从所述两级或两级以上高速缓存中的至少一者形成指令加载简档。

15.根据权利要求10所述的处理装置，其中所述高速缓存监视器经配置以在选定操作周期期间暂停监视。

16.根据权利要求15所述的处理装置，其中所述选定操作周期包含新过程的引导阶段。

17.根据权利要求15所述的处理装置，其中所述高速缓存监视器在预定义数目的先前指令或预定义时间周期上检测所述异常指令获取高速缓存未中活动。

18.一种在包含高速缓存存储器的处理电路中操作的装置，其包括：

用于在所述高速缓存存储器中加载可执行代码序列的至少若干部分的装置；

用于执行从所述高速缓存存储器的所述可执行代码序列的指令获取的装置；

用于响应于所述指令获取而检测一个或多个指令获取高速缓存未中的装置，其中单个指令获取高速缓存未中代表针对所述可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中的；

用于相对于所述一个或多个指令获取高速缓存未中监视所述指令获取以便动态地检测异常指令获取高速缓存未中活动的装置；以及

用于确定所述异常指令获取高速缓存未中活动是否指示面向返回编程攻击的装置。

19.根据权利要求18所述的装置，其进一步包括：

用于报告所述异常指令获取高速缓存未中活动的装置；以及

用于响应于所述异常指令获取高速缓存未中活动而执行关于所述可执行代码序列的补救动作的装置。

20.根据权利要求18所述的装置，其进一步包括：

用于在预定义数目的先前指令或预定义时间周期上确定所述异常指令获取高速缓存未中活动的装置。

21.一种在包含高速缓存存储器的处理电路中操作的方法，其包括：

在处理电路中执行多个代码片断的非既定序列，每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述多个代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述序列中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断；

响应于在所述多个代码片断的非既定序列的执行期间执行的指令获取而检测一个或多个指令获取高速缓存未中，其中单个指令获取高速缓存未中代表针对可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中；

通过相对于指令获取高速缓存未中监视指令获取而形成指令加载简档；以及

基于所述指令加载简档来控制至少一个指令的执行，

其中基于所述指令加载简档来控制至少一个指令的执行包括确定所述指令加载简档是否对应于面向返回编程攻击。

22.根据权利要求21所述的方法，其中基于所述指令加载简档来控制至少一个指令的执行进一步包括：

在所述指令加载简档指示所述非既定序列的所述多个代码片断中的至少一个代码片断的执行的情况下，终止所述非既定序列的至少一个代码片断的执行。

23.根据权利要求21所述的方法，其中执行多个代码片断的所述非既定序列包含利用多级高速缓存作为所述高速缓存存储器，且经高速缓冲存储的指令是第一高速缓存级中的指令，且未经高速缓冲存储的指令是主存储器或第二高速缓存级中的指令。

24.根据权利要求21所述的方法，其进一步包括：

将所述指令加载简档形成为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率，所述命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率而选择的阈值。

25.根据权利要求24所述的方法，其进一步包括：

通过监视来自已知在所述监视期间不含面向返回编程攻击的多个可执行代码序列的命中/未中比率来确定所述有效命中/未中比率。

26.一种包含高速缓存存储器的处理装置，其包括：

处理电路，其经配置以获取和执行可执行代码序列，所述可执行代码序列包含多个代码片断的非既定序列，每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述多个代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令；

高速缓存存储器系统，其可操作地耦合到所述处理电路且包含所述高速缓存存储器，其中所述非既定序列的至少一个代码片断是在所述高速缓存存储器中未发现的未经高速缓冲存储的代码片断；

高速缓存监视器，其经配置以响应于在所述多个代码片断的非既定序列的执行期间执行的指令获取而检测一个或多个指令获取高速缓存未中，其中单个指令获取高速缓存未中代表针对可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中，所述高速缓存监视器进一步经配置以通过相对于指令获取高速缓存未中监视指令获取而形成指令加载简档；以及

补救动作单元，其经配置以基于所述指令加载简档来控制至少一个指令的执行，

其中所述补救动作单元进一步经配置以确定所述指令加载简档是否对应于面向返回编程攻击。

27.根据权利要求26所述的处理装置，其中所述高速缓存存储器系统包含数据高速缓存和指令高速缓存，且所述高速缓存监视器经配置以用于从所述指令高速缓存形成所述指令加载简档。

28.根据权利要求26所述的处理装置，其中所述高速缓存存储器系统包含两级或两级以上高速缓存，且所述高速缓存监视器经配置以用于形成所述两级或两级以上高速缓存中的至少一者的所述指令加载简档。

29.根据权利要求26所述的处理装置，其中所述高速缓存监视器、所述补救动作单元、所述处理电路或其组合经配置以将所述指令加载简档形成为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率，所述命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率而选择的阈值。

30.根据权利要求29所述的处理装置，其中所述高速缓存监视器、所述补救动作单元、所述处理电路或其组合经配置以通过监视来自已知在所述监视期间不含面向返回编程攻击的多个可执行代码序列的所述命中/未中比率来确定所述有效命中/未中比率。

31.根据权利要求26所述的处理装置，其中所述高速缓存监视器经配置以在选定操作周期期间暂停监视。

32.一种在包含高速缓存存储器的处理电路中操作的装置，其包括：

用于在处理电路中执行多个代码片断的非既定序列的装置，每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述多个代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述序列中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断；

用于响应于在所述多个代码片断的非既定序列的执行期间执行的指令获取而检测一个或多个指令获取高速缓存未中的装置，其中单个指令获取高速缓存未中代表针对可执行代码序列的相应的指令获取的所获取指令不在所述高速缓存存储器中

用于通过相对于指令获取高速缓存未中监视指令获取而形成指令加载简档的装置；以及

用于基于所述指令加载简档来控制至少一个指令的执行的装置，

其中所述用于基于所述指令加载简档来控制至少一个指令的执行的装置包括用于确定所述指令加载简档是否对应于面向返回编程攻击的装置。

33.根据权利要求32所述的装置，其中所述用于基于所述指令加载简档来控制至少一个指令的执行的装置进一步包括：

用于在所述指令加载简档指示所述非既定序列的所述多个代码片断中的至少一个代码片断的执行的情况下终止所述非既定序列的至少一个代码片断的执行的装置。

34.根据权利要求32所述的装置，其进一步包括：

用于将所述指令加载简档形成为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率的装置，所述命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率而选择的阈值。

用于检测面向返回编程攻击的方法、装置和系统

技术领域

[0001] 一个特征大体上涉及检测软件系统中的恶意代码，且更特定来说涉及检测软件系统中的面向返回编程攻击的方法和装置。

背景技术

[0002] 执行软件的计算系统正在经历增长的来自通常称为黑客的攻击者的大量攻击。这些攻击者已发现将恶意代码插入计算系统中且随后致使计算系统执行恶意代码的方式。恶意代码可执行许多不同操作，例如致使计算系统运行得比正常慢，监视计算系统上的活动，致使计算系统发射或接收用户可能不希望传送的信息，破坏持续和非持续存储器中的数据，以及使计算系统崩溃。

[0003] 最近已提出有时称为面向返回编程(ROP)攻击的攻击机制。一类ROP攻击经常称为返回到libc攻击，因为其使用驻留于许多软件系统中的标准C库。ROP攻击是有力的技术，其允许攻击者攻击软件程序中的有效代码序列而无需将任何新的恶意代码注入到处理器的地址空间中。有效代码序列的小片断(经常称为小工具)可能被攻击者发现，随后串接在一起以形成新的恶意代码序列，进而避开针对代码注入的防御。

[0004] 在ROP攻击中，小代码片断是代码的以返回指令结束的部分。当调用函数时，在向经调用函数的返回的地址完成时，在所述调用之后的指令的地址被推送到堆栈上。因此，所述堆栈可包含许多返回地址，用于处理器跳转到经调用函数完成的时间。如果攻击者可对堆栈写入信息，那么攻击者可用恶意返回地址覆写既定返回地址。此返回地址可到达由攻击者识别的小工具中的一者。

[0005] 通过操纵多个返回地址，控制调用堆栈的攻击者可将多个小工具链接在一起以产生恶意代码序列，而从不需要将任何新代码注入处理器地址空间。通过这些恶意代码序列及其布置的选择，攻击者可针对由小工具串组成的恶意程序而引起任意的(但仍是图灵完全的)行为。此类型的攻击是成功的，因为在大多数系统中代码和数据地址是可预测的。也就是说，攻击者可在其自己的计算机中加载特定代码，检视其堆栈以确定代码正如何加载，且当此代码加载在目标计算机中时使用此信息来攻击返回堆栈。此攻击可大体上依赖于在不同计算机上以相同方式加载代码。

[0006] 因此，需要稳健的计算机措施，其可检测对堆栈中的弱点的攻击且在检测到此些攻击时执行补救动作。

发明内容

[0007] 可在设备、方法和计算机可读媒体中实施本发明的各种特征。此些特征可包含检测对堆栈中的弱点的攻击以及在检测到此些攻击时执行补救动作。

[0008] 在一个实例中，提供一种在包含高速缓存存储器的处理电路中操作的方法。所述方法可包含在所述高速缓存存储器中加载可执行代码序列的至少若干部分。在一个实例中，可执行代码序列可包含一系列可执行代码序列且所述系列中的所述可执行代码序列中

的一或多者与调用堆栈上的指令相关联。

[0009] 所述方法还可包含：执行从所述高速缓存存储器的所述可执行代码序列的指令获取；以及相对于高速缓存未中监视所述指令获取以便动态地检测异常未中活动，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。在一个实例中，相对于高速缓存未中监视所述指令获取可包含监视所述指令获取中的一些是高速缓存命中，且所述异常未中活动指示所述高速缓存命中相对于所述高速缓存未中的命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率选择的阈值。

[0010] 可报告所述异常未中活动以便执行关于所述可执行代码序列的补救动作。可通过响应于所述异常未中活动的所述报告终止所述可执行代码序列的执行来执行所述补救动作。在一个实例中，所述有效命中/未中比率可通过监视来自在所述监视期间不含面向返回编程攻击的多个所述可执行代码序列的命中/未中比率来确定。根据一个方面，监视可在选定操作周期期间暂停。举例来说，所述选定操作周期可包含新过程的引导阶段。根据一个方面，所述异常未中活动可在预定义数目的先前指令、预定义时间周期上和/或预定操作上下文内确定。根据一个特征，可确定所述异常未中活动指示面向返回编程攻击的装置。

[0011] 在第二实例中，提供一种处理装置，其包含处理电路、高速缓存存储器系统和高速缓存监视器。处理电路可经配置以用于获取和执行可执行代码序列。高速缓存存储器系统可操作地耦合到所述处理电路且包含至少一个高速缓存存储器。高速缓存监视器可经配置以相对于高速缓存未中监视指令获取以便动态地检测异常未中活动，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。

[0012] 在第三实例中，提供一种方法，其包含在处理电路中执行多个代码片断的非既定序列，其中每一代码片断包含包括控制转移指令的至少一个可执行指令。所述代码片断中的一或多者可包含不同于原始既定控制转移指令的经修改控制转移指令，且所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。所述方法还可包含通过相对于高速缓存未中监视指令获取而形成指令加载简档，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。可在所述指令加载简档指示所述非既定序列的所述代码片断中的某些代码片断的执行的情况下终止所述非既定序列的至少一个代码片断的执行。执行代码片断的所述非既定序列包含利用多级高速缓存作为所述高速缓存存储器，且经高速缓冲存储的指令是较靠近所述处理电路的高速缓存级中的指令，且未经高速缓冲存储的指令是主存储器或较远离所述处理电路的高速缓存级中的指令。可将所述指令加载简档形成或产生为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率，所述命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率选择的阈值。可通过监视来自已知在所述监视期间不含面向返回编程攻击的多个可执行代码序列的命中/未中比率来确定所述有效命中/未中比率。

[0013] 在第四实例中，提供一种处理装置，其包含处理电路、高速缓存监视器和补救动作单元。处理电路可经配置以用于获取和执行可执行代码序列，且高速缓存存储器系统可操作地耦合到所述处理电路且包含至少一个高速缓存存储器。所述高速缓存监视器可经配置以通过相对于高速缓存未中监视指令获取而形成指令加载简档，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。补救动作单元可经配置以用

于在所述指令加载简档指示代码片断的非既定序列的情况下执行关于所述可执行代码序列的补救动作。每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。

[0014] 本发明的再其它方面包含一种处理器装置，其包含用于在处理电路中执行多个代码片断的非既定序列的装置。每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。所述处理装置还包含用于通过相对于高速缓存未中监视指令获取而形成指令加载简档的装置，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。

[0015] 本发明的其它方面包含一种其上存储有指令的计算机可读媒体，所述指令在由处理电路执行时致使所述处理电路执行多个代码片断的非既定序列。每一代码片断包含包括控制转移指令的至少一个可执行指令，其中所述代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。所述指令还致使所述处理电路通过相对于高速缓存未中监视指令获取而形成指令加载简档，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。

附图说明

[0016] 图1是图解说明包含可能的系统存储器组织的示范性处理系统100的框图。

[0017] 图2图解说明示范性一般调用堆栈。

[0018] 图3图解说明由来自有效代码序列的小代码片断构成的示范性小工具。

[0019] 图4图解说明包含高速缓存存储器系统的示范性处理系统。

[0020] 图5图解说明图4的高速缓存存储器系统的实例的细节。

[0021] 图6图解说明用于产生恶意代码序列的受破坏调用堆栈的实例，包含串接在一起以形成面向返回编程(ROP)攻击的小工具。

[0022] 图7是展示用以检测高速缓存存储器的异常未中活动以检测并终止ROP攻击的示范性过程的流程图。

[0023] 图8是展示形成指令加载简档并响应于对指令加载简档的分析而执行补救动作的示范性过程的流程图。

[0024] 图9是展示形成关于有效代码序列的执行的信息以用于分析指令加载简档的示范性过程的流程图。

具体实施方式

[0025] 在以下描述中，参考附图，其中借助于图解说明而展示其中可实践本发明的特定实例。实例既定以充分的细节描述本发明的方面以使所属领域的技术人员能够实践所描述特征中的一或多者。在不脱离本发明的范围的情况下可利用其它特征且可对所揭示实例做

出改变。不应在限制性意义进行以下详细描述，且范围仅由所附权利要求书界定。

[0026] 术语“数据”在本文可以可互换方式使用以指代可由处理器使用的计算指令和可由处理器操作的数据。术语“数据”的使用的上下文应当使得当“数据”涉及指令时是明显的。在需要的情况下，指令可明确地称为指令或指令数据。

[0027] 概述

[0028] 本发明涉及用于检测对堆栈中的弱点的攻击且在检测到此些攻击时执行补救动作的设备、方法和计算机可读程序。

[0029] 根据一个特征，处理器可用可执行代码(例如，来自外部存储装置或主存储器)加载高速缓存存储器的一或多个行。处理器随后通过从高速缓存存储器内的地址(位置)调用指令，且对于在高速缓存存储器中未发现的指令，在调用新指令之前或与调用新指令同时地将所述新指令加载到高速缓存存储器中(例如，从外部/主存储器)，来执行可执行代码。处理器可监视高速缓存存储器内发现的指令被调用的次数对高速缓存存储器内未发现的指令被调用的次数，以便产生命中/未中比率。举例来说，在常规代码中，加载新行的需要(“未中”)可例如仅大约每100个指令发生一次。相比之下，在ROP攻击中，加载新页的需要可每3个指令发生一次。处理器可在命中/未中比率下降到低于预定义阈值的情况下终止可执行代码序列的执行。可将阈值选择为低于用于合法程序的典型命中/未中比率。举例来说，可针对每一操作系统和/或处理器类型来经验上选择命中/未中阈值。因为在处理器或操作系统的初始引导期间，往往存在大量页切换或页加载，所以在处理器的引导阶段(或其它选定操作周期)期间可能暂停监视步骤。为了较快地检测ROP攻击的发生，可在固定数目的先前指令(例如，最后10、50、100个指令)上计算命中/未中比率。

[0030] 示范性数据结构

[0031] 攻击者已发现将恶意代码插入计算系统中的方式。作为响应，许多计算系统设计者已通过实施例如数据执行防止(DEP)和地址空间布局随机化(ASLR)等系统来使得攻击者较难这么做。

[0032] DEP涉及对处理器可使用的信息加标签为可执行信息或不可执行信息。在许多软件系统中，黑客可致使处理器获取被视为数据而不是指令的信息。处理器可解译此数据且将其作为并非软件编程者预期的指令来执行。此将信息加标签为可执行的使得黑客较难以致使处理器执行数据，因为所述数据被加标签为不可执行的。

[0033] 地址空间布局随机化(ASLR)涉及为关键的数据和指令区域(例如可执行代码的基部、库的位置、堆的位置和堆栈位置)随机地布置处理器的地址空间内的位置。

[0034] 然而，面向返回编程(ROP)攻击避开了这些保护机制，因为ROP攻击不需要插入任何代码。而是，其通过修改调用堆栈上的返回地址来攻击好代码的片断。

[0035] 图1是图解说明包含可能的系统存储器组织的示范性处理系统100的框图。当在处理系统100内执行软件时，处理单元102大体上存取持续存储装置106以读入呈代码形式的可执行指令。

[0036] 持续存储装置106可为用于存储呈指令和数据形式的信息的任何合适结构。举例来说而非限制，这些存储媒体可包含计算机可读媒体(CRM)。此CRM可包含但不限于磁性和光学存储装置，例如磁盘驱动器、磁带、CD(压缩光盘)、DVD(数字多功能光盘或数字视频光盘)，以及其它等效存储装置。

[0037] 可执行指令连同处理单元102可在执行代码时使用的其它数据结构一起存储在主存储器104中。图1中所示的一些实例数据结构包含堆、堆栈、恒定数据、初始化数据和指令。在各种软件程序可能在处理系统100内处于活动状态时，堆经常用于所述软件程序的动态分配。例如恒定数据和初始化数据等各种数据可存储在主存储器104中，供在处理系统100上运行的一或多个程序存取。与各种程序相关联的指令可存储在指令区域、堆或其组合中。

[0038] 处理系统100内可存在各种堆栈。特定关注的一个堆栈经常称为调用堆栈。调用堆栈可被ROP攻击破坏，通过本文论述的一或多个特征/方面响应于如下文更完整论述的对高速缓存活动的监视可检测所述ROP攻击。

[0039] 图2图解说明示范性一般调用堆栈200。调用堆栈200可用以存储在调用函数时可使用的各种信息。推送于堆栈上的每一函数占据帧202，如分别用于被调用函数A、B和C的帧202A、202B和202C指示。

[0040] 作为非限制性实例，调用堆栈200的每一帧202可包含例如从调用者函数传递到被调用函数的参数等信息。调用堆栈200还可包含用于存储可由被调用程序使用的各种局部变量的存储器区域。指示在被调用函数已完成执行之后调用函数执行应在何处继续的返回地址也可包含在调用堆栈200中。调用堆栈200作为后入先出(LIFO)缓冲器操作，意味着被推送于堆栈上的最后数据是从堆栈首先弹出的数据。调用堆栈200可为相当深的，指示许多函数调用嵌套在其它函数内。

[0041] 帧指针204大体上指向当前操作函数的帧202。堆栈指针206指向堆栈上的可用以被弹出且返回到处理器的下一数据位置。

[0042] ROP攻击通过将恶意地址写入到各种帧202的返回地址部分来利用堆栈数据结构。与调用堆栈200相关的被ROP攻击破坏的操作可通过本文论述的一或多个特征/方面响应于如下文更完整论述的对高速缓存活动的监视来检测。

[0043] 图3图解说明由来自有效代码序列300的小代码片断构成的小工具304和306。图3中将指令图解说明为包含一或多个字节的指令数据。作为非限制性实例，有效代码序列300(也可称为可执行代码序列)包含3字节的比较(CMP)指令、2字节的右移(SHR)指令、3字节的加载累积器(LDA)指令、2字节的相加(ADD)指令、4字节的从寄存器c移动到寄存器(MOV a，c)指令，以及2字节的返回(RET)指令。图3中所示的指令的各种字节中的指令和编码经制作且既定不指示任何特定指令集。

[0044] 可从其它函数调用的函数大体上以返回指令302结束。当处理器遇到返回指令302时，其从堆栈200(图2)弹出顶部返回地址以确定应在何处继续执行。堆栈上的顶部地址应大体上指向在调用函数中的调用指令之后调用函数中的下一指令的地址。然而，在使用ROP攻击的攻击中，返回地址可能已被修改为指向不同地址。

[0045] 攻击者可识别执行可对攻击者有用的特定函数的小工具304。小工具304以有效代码序列300的返回指令302结束。然而，其将大体上在有效代码序列300内的某个地方开始以在返回指令302的前方包含有用的指令。因此，小工具304包含相加指令，之后是移动指令，之后是返回指令302。

[0046] 一些指令集，例如精简指令集计算机(RISC)指令，通常是固定长度的，意味着所有指令由相同数目的字节(例如，4字节)组成。其它指令集，例如复杂指令集计算机(RISC)指令的长度可变化，如图3中图解说明。在可变长度指令的情况下，攻击者有机会创建在原始

有效代码序列300中不希望的新指令。作为非限制性实例，小工具306在有效代码序列300的移动指令内的位置开始，其可在处理器在所述点开始执行的情况下产生相减指令。因此，小工具306包含相减指令，之后是返回指令302。

[0047] 攻击者可检查将驻留在存储器中的代码，例如标准C库，或操作系统的部分。攻击者可随后识别这些小工具304和306(即，小代码片断)中的许多以产生ROP指令库。这些ROP指令可随后经串接在一起以产生有用的、非既定的且恶意的代码序列而无需将任何代码插入存储器。实际上，攻击者仅必须将调用堆栈200上的返回地址改变为指向所要小工具304和306的开始。

[0048] 示范性架构和组件

[0049] 图4图解说明包含高速缓存存储器系统404的处理系统100的实例。处理系统100可包含一或多个处理器402、高速缓存存储器系统404以及主存储器104。也可包含各种通信接口424和输入/输出(I/O)接口426。

[0050] 作为非限制性实例，通信接口424可包含用于蜂窝式网络上的通信的接口、串行端口、并行端口、以太网连接、通用串行总线(USB)连接、IEEE1394(“火线”)连接、蓝牙无线连接、802.1a/b/g/n型无线连接，以及其它合适的通信协议和接口。

[0051] 作为非限制性实例，I/O接口426可包含到例如键盘、鼠标、跟踪球、触觉装置、音频输入和输出以及显示器等装置的接口。

[0052] 包含高速缓存监视器414以监视对高速缓存存储器系统404的各种存取是“命中”还是“未中”，命中意味着所存取数据驻留在高速缓存存储器系统404中且有效，未中意味着所要位置不存在于高速缓存存储器系统404中或包含无效数据。因此，总线416可向高速缓存监视器414指示未中或命中以及其它信息。高速缓存监视器414可在预定义时间周期或预定义数目的指令获取内收集信息，例如命中数目和未中数目。关于命中和未中的所收集信息可在总线418上发射到补救动作单元420。补救动作单元420可处理命中和未中信息且经由总线422将结果报告给处理器402。如下文更完整阐释，为处理器402收集和识别的信息可针对不同实例或实施方案改变，且可与命中/未中比率一样简单到更复杂，例如指令加载简档。

[0053] 高速缓存414的功能、补救动作单元420的功能或其组合在不同实例中可以硬件或软件来执行。

[0054] 作为非限制性实例，高速缓存监视器414中的一或多个简单硬件计数器可用以跟踪高速缓存命中、高速缓存未中或其组合。硬件计数器可通过软件来清除、预设和读取以控制和分析在特定时间周期、特定数目的指令获取或其组合内的高速缓存命中或未中信息。作为一个实例，补救动作单元420可经配置以在软件控制下确定命中/未中比率或指令加载简档。控制可基于推送型模型，其中硬件基于满足硬件中的某种准则而产生中断。举例来说，如果在给定时间周期内请求某一阈值数目的命中或获取，那么硬件可产生中断。控制还可基于拉动型模型，其中软件可相对于例如逝去的时间、上下文切换和所获取指令的数目等某些事件监视硬件寄存器。在一个实例中，可在周期性时钟中断上检查硬件寄存器以确认是否检测到异常的高速缓存未中或指令获取条件。

[0055] 作为另一非限制性实例，可包含较多硬件，例如阈值寄存器或其它简档信息，其可由核或操作系统基于当前操作的预期行为来设定。硬件可随后在命中/未中比率超过或低

于阈值寄存器或简档信息时产生中断。

[0056] 在一些实例中，处理器402、高速缓存存储器系统404、高速缓存监视器414和补救动作单元420可全部在同一半导体装置或处理装置428上。在其它实例中，高速缓存系统404可在与处理器402分离的装置上。在此些实例中，高速缓存监视器414和补救动作单元420可与处理器402一起定位、与高速缓存存储器系统404一起定位，或在所述两个装置之间分裂。

[0057] 许多高速缓存存储器系统404包含多级高速缓存存储器，包括高速缓存存储器的两个或两个以上级412。图4的非限制性实例图解说明包含单独指令高速缓存406和数据高速缓存408的高速缓存的第一级。使指令与数据分离可通过产生用于获取信息的并行路径且利用对于指令和数据可不同的时间和空间接近度而产生性能增强。高速缓存的第二级经配置为统一高速缓存410，包含用于指令高速缓存406的指令和用于数据高速缓存408的数据。另外，高速缓存存储器系统404可包含用于高速缓存的不同配置，例如集合关联高速缓存和用于满的高速缓存的各种替换协议，如高速缓存设计领域的技术人员所知。

[0058] 高速缓存允许较快地检索与必须到达主存储器104相关的指令和数据。与传统存储器相比的折中是在大小上。大体上，较小的存储器具有较高的带宽、较低的等待时间或两者的组合。主存储器104通常实施为动态随机存取存储器(DRAM)，其可具有相对好的带宽，但可能具有相对长的等待时间。通过在高速缓存中高速缓存频繁使用的数据和指令，可快得多地且以较高带宽接收从处理器402的获取。作为一般规则，1级高速缓存(即，最靠近处理器402的高速缓存)较小、较快，且具有较低等待时间。2级高速缓存通常较大，可较慢，且可具有较长等待时间。然而，它们仍比主存储器104快，使得通过包含第二级高速缓存，性能改善是可能的。

[0059] 图4的高速缓存存储器系统404仅是一个实例。可使用较少级中的较多者且每一级可实施为统一高速缓存410或单独的指令高速缓存406和数据高速缓存408。本文论述的实例与指令最相关。因此，讨论可具体涉及指令高速缓存406。然而，所属领域的技术人员将了解，这些特征可在任一级的高速缓存上以及在指令高速缓存406和统一高速缓存410两者中实践。

[0060] 处理器攻击空间和时间接近度的性质以优化高速缓存中保持的内容以及必须从主存储器104检索的内容。举例来说，频繁调用的代码或函数可保持在指令高速缓存406中。而且，对于页中的函数或代码常见的是调用其它附近或空间上接近的代码(例如，在同一页内)。

[0061] 然而，此接近度的假设不一定对于ROP小工具成立，因为ROP攻击的基本性质是它们将位于存储器的相异且表面上随机的部分的小代码段缝合在一起。小工具的此分布可导致对于ROP攻击的大量“未中”。换句话说，处理器402必须频繁地将新的行加载到高速缓存中以找到被调用的小工具。相比之下，有效代码序列300(图3)可大体上具有少得多的未中和更多“命中”。换句话说，可在已加载于高速缓存中的相同行中发现被调用函数或代码。与有效代码序列300相比ROP攻击对高速缓存命中/未中特性的不同影响可用于建立防止ROP有效负载成功执行的检测机制。此系统(监视命中/未中特性)可对特定操作系统(OS)或过程进行训练。

[0062] 本发明的一些特征或方面允许在可能的最低级检测ROP攻击的执行，因此其不仅可在过程处应用，而且可在OS级粒度应用。而且，各种方面或特征可不联系到源代码或编译

时间修改。本发明的一些实例不一定依赖于防止攻击者读取存储器的任何部分或直接控制执行流。用于减轻ROP攻击的所有当前已知机制(尤其是ASLR)例行地被绕过,因为额外的漏洞将存储器布局泄露给攻击者,或存储器的某个部分变为仍容易可预测的,因此提供攻击表面来找到ROP有效负载。相比之下,即使攻击者将具有所攻击过程的存储器映射的完全知识,提出的方法也可检测ROP攻击执行。

[0063] 图5图解说明不同上下文中的图4的高速缓存存储器系统404的实例的细节以阐释高速缓存的操作,因为其涉及本发明的各种方面。指令高速缓存406包含行的集合,其中每一行包含索引部分502和数据部分504。索引部分包含部分地址(通常是最高有效位MSB),使得存储器的具有相同MSB的不同区域可映射到指令高速缓存406的特定行。因此,如图解说明,具有索引2的高速缓存行可含有指令数据2,其源自主存储器104中的指令数据2A或主存储器104中的指令数据2B。如果希望加载新指令且没有更多高速缓存行可用,那么可用较新近的指令(即,时间上较靠近)覆写(即,替换)每一高速缓存行。

[0064] 在操作中,处理器402请求检索指令的特定地址。如果所述地址匹配于指令高速缓存406的一行中的索引,那么在“命中”行处的指令数据(或其一部分)作为经高速缓冲存储的指令返回到处理器402。如果所请求地址不匹配于指令高速缓存406中的有效索引中的任一者,那么“未中”发生,且指令高速缓存406从主存储器104获取包含未经高速缓冲存储的指令506的足够信息以填充高速缓存的一行且将所请求地址的索引放置于高速缓存的所述行中。对具有所述特定指数的所述高速缓存行的后续请求现在将产生对经高速缓冲存储的指令的命中,且将不需要从主存储器104获取指令数据。

[0065] 长高速缓存行产生空间接近度,且用较新近使用的数据替换高速缓存行的能力产生时间接近度。有效代码序列300(图3)将大体上具有良好的空间接近度和良好的时间接近度。另一方面,ROP攻击将大体上具有不良的空间接近度和不良的时间接近度,原因是可位于存储器的不一定经常被存取的许多不同区域中的短片断的特征。

[0066] 示范性面向返回编程检测

[0067] 图6图解说明用于产生恶意代码序列的受破坏调用堆栈600的实例,包含串接在一起以形成面向返回编程(ROP)攻击的小工具。受破坏的调用堆栈600呈受破坏形式,原因是攻击者修改了受破坏调用堆栈600上的返回地址中的一或多者。而且图6中图解说明指令高速缓存406的一部分和主存储器104的一部分。

[0068] 参见图3、5和6,受破坏的调用堆栈600可包含用于函数U-Z的帧。从受破坏调用堆栈600到指令高速缓存406的实心箭头指示从堆栈弹出的返回地址,致使处理器402在特定小工具的开始处开始执行。从指令高速缓存406到受破坏调用堆栈600的虚线箭头指示在特定小工具的末尾处执行以从堆栈获取返回地址的返回指令。

[0069] 在受破坏调用堆栈600中,用于函数Z的返回地址已经修改为指向小工具A的开始地址。类似地,用于函数Y的返回地址已经修改为指向小工具B的开始地址,用于函数W的返回地址已经修改为指向小工具E的开始地址,用于函数V的返回地址已经修改为指向小工具D的开始地址,且用于函数U的返回地址已经修改为指向小工具C的开始地址。用于函数X的返回地址未经修改。

[0070] 由于这些修改,当函数Z完成其操作且执行返回指令302,不是返回到正确地点,控制在小工具A的开始处继续,对于所述开始处,地址已被放置于函数Z的返回地址中。每个小

工具以返回指令302结束。因此，当小工具A完成时，其返回指令302指向函数Y的返回地址。然而，函数Y的返回地址已被修改以指向小工具B的开始。因此，并非返回到正确的地点，控制在小工具B的开始处继续。继续小工具执行，在小工具B完成之后，并非返回到函数U的正确的地点，控制在小工具C的开始处继续。在小工具C完成之后，并非返回到函数V的正确的地点，控制在小工具D的开始处继续。在小工具D完成之后，并非返回到函数W的正确的地点，控制在小工具E的开始处继续。这种将小工具A到E串接在一起可执行形成ROP攻击的至少一部分的重要功能。

[0071] 然而，因为这些小工具A到E可在多样的地址处存在，所以其可造成比程序执行已正常发生的情况更大数目的高速缓存未中。在图6的非限制性实例中，小工具A不位于指令高速缓存406中，从而造成未中以及需要从主存储器104获取未经高速缓冲存储的代码片断602。类似地，小工具C、D和E未中指令高速缓存406，从而造成从主存储器104分别获取未经高速缓冲存储的代码片断604、606和608。小工具B的获取造成命中，从而允许返回小工具B的经高速缓冲存储的代码片断。

[0072] 在由较多代码片断构成的较长的ROP攻击有效负载的情况下，未中相对于命中大于预期数目的趋势变得较明显，且人们愈加相信ROP攻击可能是原因。

[0073] 图7是展示用以检测高速缓存存储器的异常未中活动以检测并终止ROP攻击的示范性过程700的流程图。将参考图7和4描述过程700。在操作框702中，过程包含在高速缓存存储器中加载可执行代码序列。换句话说，用可执行代码序列的至少若干部分加载高速缓存行中的至少一些行。在操作框704中，处理器402执行从高速缓存存储器的可执行代码序列的指令获取，其可包含高速缓存命中和高速缓存未中。这些代码序列可为有效代码序列，或其可为形成ROP攻击有效负载的一系列小工具。

[0074] 在操作框706中，高速缓存监视器414单独地、与补救动作单元420结合或与补救动作单元420和处理器402结合而相对于高速缓存未中监视指令获取，以动态地确定是否存在异常未中活动，其中高速缓存未中是在指令获取时不在高速缓存存储器中的所获取指令。在指令执行期间动态地监视指令获取，而非监视指令获取以尝试预先确定相对于特定指令序列的高速缓存命中/未中行为。可通过不同度量来确定异常未中活动。一些非限制性度量是形成命中/未中比率的相对于高速缓存未中的数目的高速缓存命中的数目、相对于指令获取的数目的高速缓存未中的数目、在某一时间周期中的高速缓存未中的数目，或在预期代码序列上的高速缓存未中的数目。作为非限制性实例，此命中/未中比率可为在获取指令时的移动平均值。当然，命中/未中比率可相反地表达为未中/命中比率。

[0075] 在一些实例中，可能需要某一预定义数目的指令获取来确定异常未中活动。决策框708指示发生测试以查看是否已达到所要数目的获取。如果为否，那么控制返回到操作框704，获取和执行更多指令。作为非限制性实例，所要数目的获取可设定于例如10、50和100个指令等数目。另外，可在预定义时间周期中确定所要数目的获取。作为另一非限制性实例，所要数目的获取可与预定操作上下文相关，例如特定软件程序的执行、特定软件库，或特定操作系统操作。

[0076] 如果已发生足够的指令获取，那么决策框710指示测试以查看是否有异常未中活动，其可经界定为低于阈值的命中/未中比率。如果为否，那么可能正在执行有效代码序列，且控制返回到操作框704以监视更多指令获取。

[0077] 如果存在异常未中活动，那么可能ROP攻击正在执行，且操作框712指示可报告异常未中活动，以使得可采取关于可执行代码序列的补救动作。如较早阐释，报告可以是向补救动作单元420或者向处理器402。

[0078] 操作框714指示可执行的一种类型的补救措施将是响应于异常未中活动的报告而终止可执行代码序列的执行。此终止可以许多不同方式终止，例如产生硬件或软件例外或产生硬件或软件中断。

[0079] 可将阈值选择为低于用于有效代码序列的典型命中/未中比率。举例来说，可针对每一操作系统和/或处理器类型来经验上选择命中/未中阈值。

[0080] 而且，可存在有效代码序列的执行周期或某些类型的执行，其中可预期较大量的未中，例如在高速缓存尚未被加载可造成高速缓存废弃的有效指令或特定有效代码序列的引导阶段期间。因此，可存在选定操作周期，例如当监视动作可暂停时在处理器402或操作系统的初始引导期间。如果这些类型的事件是已知的，那么可针对有效代码序列和ROP攻击而形成较复杂的指令加载简档。

[0081] 操作框716指示过程也可确定异常未中活动指示面向返回编程攻击。虽然在末尾处图解说明，但如果执行操作框716的动作，那么其可以操作712和714可响应于潜在ROP攻击的确定的方式，在过程中较早地执行。

[0082] 图8是展示形成指令加载简档并响应于对指令加载简档的分析而执行补救动作的示范性过程800的流程图。将参考图8和4描述过程800。操作框802指示过程在处理电路402中执行代码片断的非既定序列，每一代码片断包括包含控制转移指令的至少一个可执行指令。所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。过程执行非既定代码序列，因为代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令。作为非限制实例，控制转移指令可为例如跳转指令、返回指令和分支指令等指令。因此，原始既定控制转移函数例如是作为既定有效程序原始置于调用堆栈上的返回指令。在ROP攻击中，原始既定控制转移函数可经改变为经修改控制转移指令以执行导致代码片断的非既定序列的不同函数。

[0083] 操作框804指示过程通过相对于高速缓存未中监视指令获取而形成指令加载简档，其中高速缓存未中是在指令获取时不在高速缓存存储器中的所获取指令。换句话说，在有效代码序列和代码片断的执行期间监视高速缓存命中和高速缓存未中。在这些指令获取发生时，可形成指令加载简档。指令加载简档可能相当复杂，且跟踪例如高速缓存未中之间的时间分离和空间分离等事件和动作，且可甚至跟踪高速缓存行索引的替换以指示在ROP攻击中是否较可能涉及某些高速缓存行。另一方面，指令加载简档可与上文参见图7论述的命中/未中比率一样简单。

[0084] 操作框806指示指令加载简档可形成为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率。操作可监视以查看命中/未中比率是否低于相对于指示有效代码序列的有效命中/未中比率可确定或选择的阈值。

[0085] 操作框808指示可根据从已知在监视命中/未中比率期间不含面向返回编程攻击的多个可执行代码序列监视命中/未中比率来确定有效命中/未中比率，如下文参见图9更完全地阐释。

[0086] 决策框810指示可执行测试以查看当前指令加载简档是否类似于ROP攻击的指令

加载简档。如果为否，那么控制可返回到操作框804以监视额外指令获取以及其是否造成高速缓存命中或高速缓存未中。

[0087] 如果当前指令加载简档类似于ROP攻击的指令加载简档，那么操作框812指示，如果指令加载简档指示涉及可执行代码序列中的一或多者的面向返回编程攻击的执行，那么可执行相对于可执行代码序列的补救动作。作为非限制性实例，补救动作可为如果指令加载简档指示上文参见图7的序列的代码片断中的某些的执行，那么终止序列的至少一个代码片断的执行。

[0088] 可在用于在正常操作期间使用的测试环境中事先形成用于有效代码序列和ROP攻击两者的简档。

[0089] 图9是展示形成关于有效代码序列和ROP攻击的执行的信息以用于分析指令加载简档的示范性过程900的流程图。将参考图9和4描述过程900。在操作框902中，执行有效测试代码序列。在操作框904中，监视针对有效测试代码序列的高速缓存命中和高速缓存未中。在一个实例中，从有效测试代码序列的执行可形成有效命中/未中比率。操作框906指示针对有效代码序列可形成各种指令加载简档。以此方式，可具有相对低命中/未中比率的有效代码序列仍可具有将所述序列识别为有效代码序列而不是ROP攻击的其它命中/未中特性。可针对可能关注的任意数目的有效代码序列重复操作框902、904和906。

[0090] 在操作框908中，执行测试ROP攻击序列。在操作框910中，针对测试ROP攻击序列监视高速缓存命中和高速缓存未中。操作框912指示针对测试ROP攻击序列可形成各种指令加载简档。因为攻击者可能相当创新且进行无法预期的事，所以形成测试ROP攻击序列可比识别所关注有效代码序列更困难。然而，针对至少一些ROP攻击的指令加载简档的形成可帮助高速缓存监视器414、补救动作单元420和处理器402做出关于ROP攻击是否正在执行的更明智的决策。

[0091] 返回到图8，在用于有效代码序列300和ROP攻击两者的指令加载简档的情况下，决策框808可经扩展到测试当前指令加载简档是否类似于ROP攻击或不相似于有效代码序列。

[0092] 应当指出，本文相对于图7、8和9论述的过程是与通过监视高速缓存存储器系统404来检测ROP攻击相关的过程，且不一定是与在处理系统100上运行的软件程序的执行相关的过程。

[0093] 所展示和描述的特定实施方案仅是实例且不应阐释为实施本发明的仅有方式，除非本文另外指定。所属领域的技术人员容易了解，本发明中的各种实例可通过许多其它分割解决方案来实践。

[0094] 本文描述且在图中图解说明的组件、动作、特征和/或功能中的一或多者可经重新布置和/或组合为单个组件、动作、特征或功能或者在若干组件、动作、特征或功能中体现。在不脱离本文描述的各种特征和/或方面的情况下也可添加额外元件、部件、动作和/或功能。本文描述的算法可以软件有效地实施和/或嵌入硬件中。

[0095] 在描述中，可以框图形式展示元件、电路和功能，以免用不必要的细节混淆本发明。相反，所展示和描述的特定实施方案仅是示范性的且不应阐释为实施本发明的仅有方式，除非本文另外指定。另外，块界定以及各种块之间的逻辑的分割是特定实施方案的示范。所属领域的技术人员容易了解，本发明可通过许多其它分割解决方案来实践。对于大部分，已经省略关于时序考虑和类似的细节，其中此些细节对于获得本发明的完整理解不是

必要的，且在所属领域的技术人员的能力以内。

[0096] 还注意到，可将实例描述为过程，所述过程描绘为流程图、流图、结构图或框图。虽然流程图可将操作描述为顺序过程，但操作中的许多可并行地或同时地执行。另外，可重新布置操作的次序。当过程的操作完成时，过程终止。过程可对应于方法、函数、程序、子例程、子程序等等。当过程对应于函数时，其终止对应于所述函数到调用函数或主函数的返回。

[0097] 所属领域的技术人员将了解，可使用多种不同技艺和技术中的任一种来表示信息和信号。举例来说，可通过电压、电流、电磁波、磁场或磁性粒子、光场或光学粒子或者其任何组合来表示整个本描述中可能参考的数据、指令、命令、信息、信号、位、符号和码片。一些图式可能为了呈现和描述的清楚而将信号图解说明为单个信号。所属领域的技术人员将了解，信号可表示信号总线，其中总线可具有多种位宽度，且本发明可在任一数目的数据信号上实施，包含单个数据信号。

[0098] 本文描述的元件可包含同一元件的多个实例。这些元件可一般地由数字标示符(例如110)指示，且具体由跟随有字母标示符的数字指示符(例如110A)或前面带有“破折号”的数字指示符(例如110-1)来指示。为了便于理解描述内容，对于大部分，元件数字指示符以其上介绍或最完整讨论元件的图式的数字开始。

[0099] 应理解，本文使用例如“第一”、“第二”等标示对元件的任何参考均不限制这些元件的数量或次序，除非明确陈述此限制。实际上，这些标示在本文可用作在两个或两个以上元件或一元件的若干实例之间进行区分的方便方法。因此，对第一和第二元件的参考不意味着该处仅可采用两个元件或第一元件必须以某种方式在第二元件之前。另外，除非另外陈述，否则一组元件可包括一或多个元件。

[0100] 而且，存储媒体可表示用于存储数据的一或多个装置，包含用于存储信息的只读存储器(ROM)、随机存取存储器(RAM)、磁盘存储媒体、光学存储媒体、快闪存储器装置和/或其它机器可读媒体，以及处理器可读媒体和/或计算机可读媒体。术语“机器可读媒体”、“计算机可读媒体”和/或“处理器可读媒体”可包含(但不限于)能够存储、含有或载运指令和/或数据的非暂时性媒体，例如便携式或固定存储装置、光学存储装置和各种其它媒体。因此，本文描述的各种方法可完全地或部分地由可存储在“机器可读媒体”、“计算机可读媒体”和/或“处理器可读媒体”中且由一或多个处理器、机器和/或装置执行的指令和/或数据实施。

[0101] 此外，各种特征和/或方面可以硬件、软件、固件、中间件、微码或其任一组合来实施。当以软件、固件、中间件或微码实施时，用以执行必要任务的程序代码或代码段可存储在例如存储媒体等机器可读媒体或其它存储装置中。处理器可执行所述必要任务。代码段可表示过程、函数、子程序、程序、例程、子例程、模块、软件包、类或指令、数据结构或程序语句的任一组合。代码段可通过传递和/或接收信息、数据、自变量、参数或存储器内容而耦合到另一代码段或硬件电路。信息、自变量、参数、数据等等可经由任一合适手段传递、转发或发射，所述手段包含存储器共享、消息传递、令牌传递、网络发射等等。

[0102] 结合本文所揭示的实例描述的各种说明性逻辑块、模块、电路、元件和/或组件可用经设计以执行本文描述的功能的通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它可编程逻辑组件、离散门或晶体管逻辑、离散硬件组件或其任何组合来实施或执行。通用处理器可以是微处理器，但在替代方案中，所述处理

器可以是任何常规处理器、控制器、微控制器或状态机。处理器还可实施为计算组件的组合，例如DSP与微处理器的组合、若干微处理器、结合DSP核心的一或多个微处理器或任何其它此类配置。经配置以用于执行本文描述的特征的通用处理器可视为用于实行此些特征的专用处理器。类似地，通用计算机在经配置以用于实行本文描述的特征时可视为专用计算机。

[0103] 结合本文所揭示的实例描述的方法或算法可直接以硬件、以由处理器执行的软件模块或以所述两者的组合以处理单元、编程指令或其它指示的形式来实施，且可包含在单个装置中或分布于多个装置上。软件模块可驻留在RAM存储器、快闪存储器、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可装卸式盘、CD-ROM或此项技术中已知的任何其它形式的存储媒体中。存储媒体可耦合到处理器，使得处理器可从存储媒体读取信息和向存储媒体写入信息。在替代方案中，存储媒体可与处理器成一体式。

[0104] 所属领域的技术人员将进一步了解，结合本文所揭示的实例和特征描述的各种说明性逻辑块、模块、电路和算法步骤可实施为电子硬件、计算机软件或所述两者的组合。为了清楚地说明硬件与软件的这种可交换性，上文已大体上在其功能性方面描述了各种说明性组件、块、模块、电路和步骤。将此类功能性实施为硬件、软件还是其组合取决于特定应用和对整个系统施加的设计选择。

[0105] 本文描述的各种特征可在不同系统中实施。应注意，前述实例不应解释为限制性的。实例的描述既定为说明性的，且将不限制权利要求书的范围。由此，本发明教示可容易应用于其它类型的设备，且所属领域的技术人员将了解许多替代例、修改和变化。

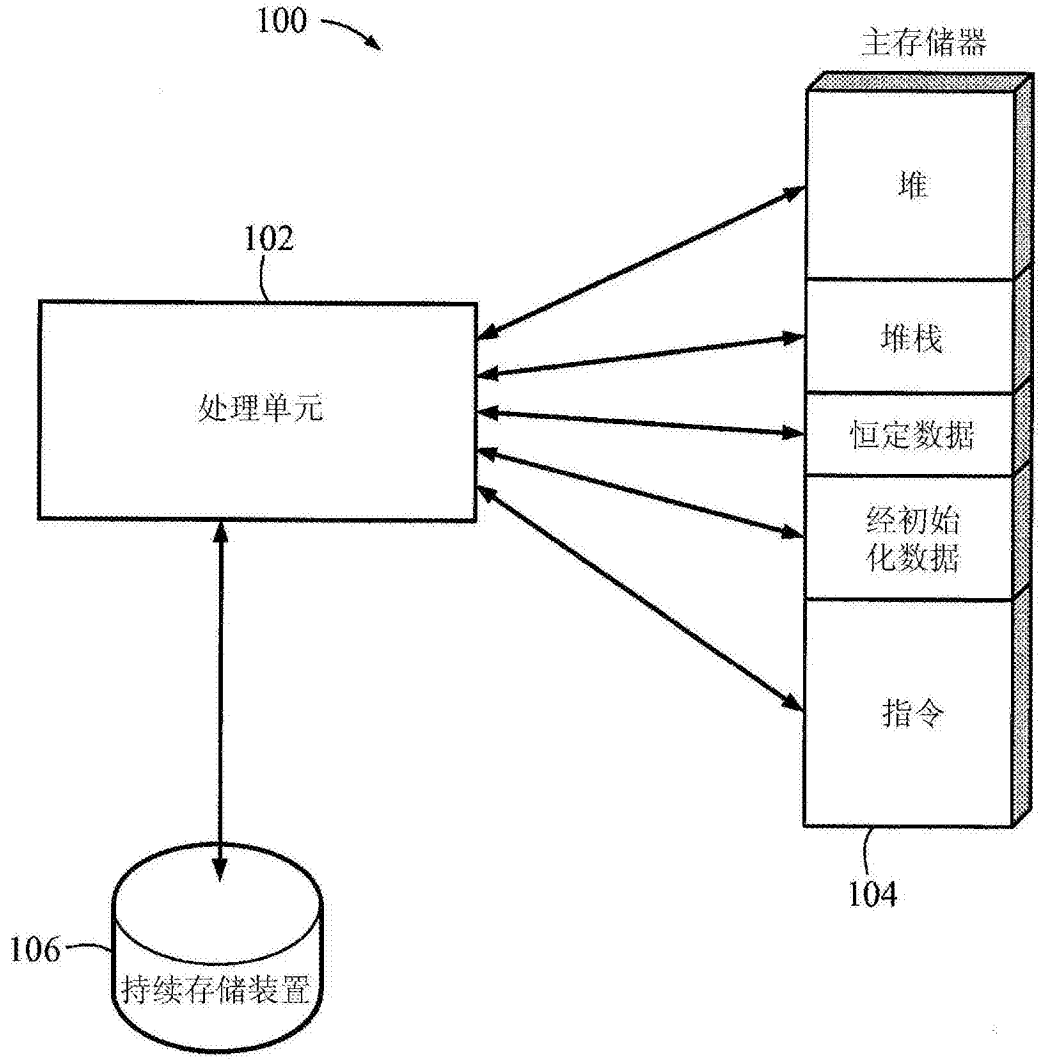


图1

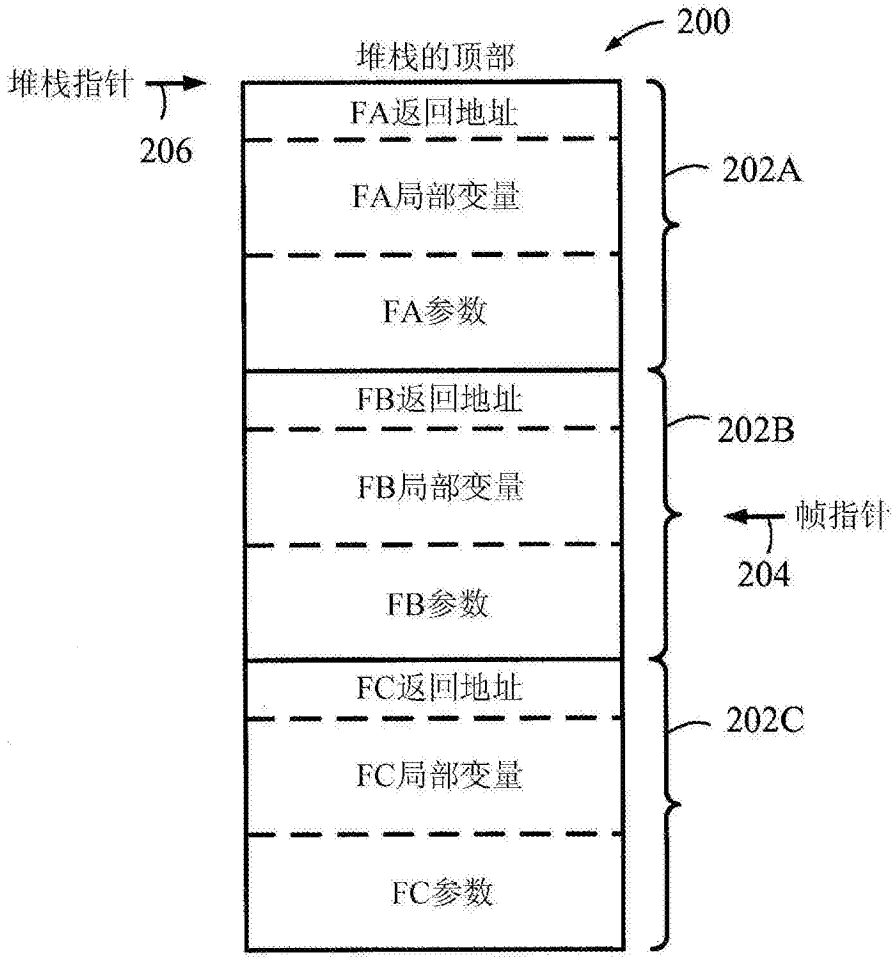

200
堆栈的顶部
堆栈指针
206
FA返回地址
FA局部变量
FA参数
202A
FB返回地址
FB局部变量
FB参数
202B
帧指针
204
FC返回地址
FC局部变量
FC参数
202C

图2

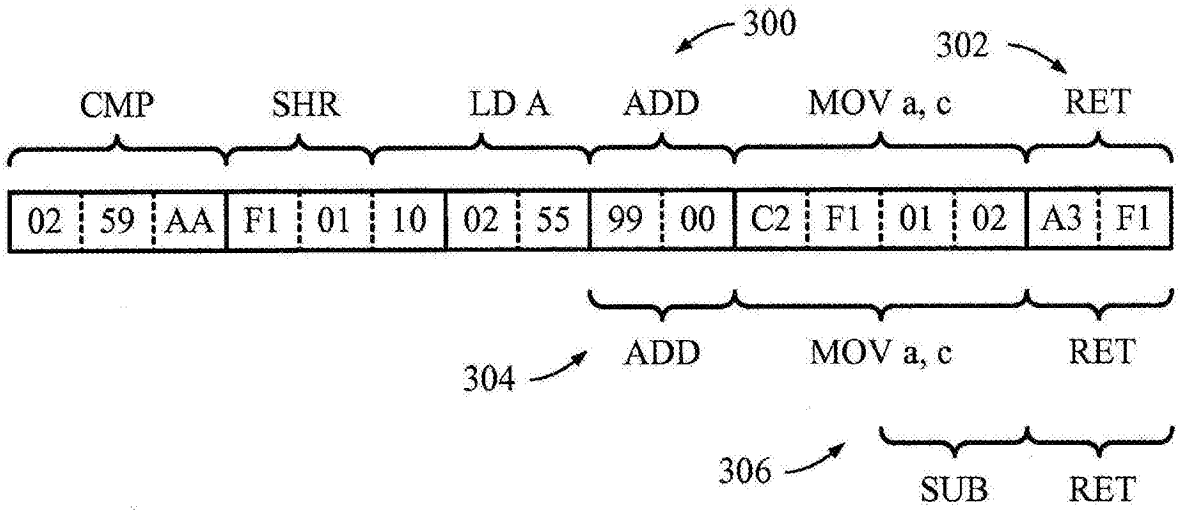

300
302
CMP
SHR
LD A
ADD
MOV a, c
RET
02 59 AA F1 01 10 02 55 99 00 C2 F1 01 02 A3 F1
304
ADD
MOV a, c
RET
306
SUB
RET

图3

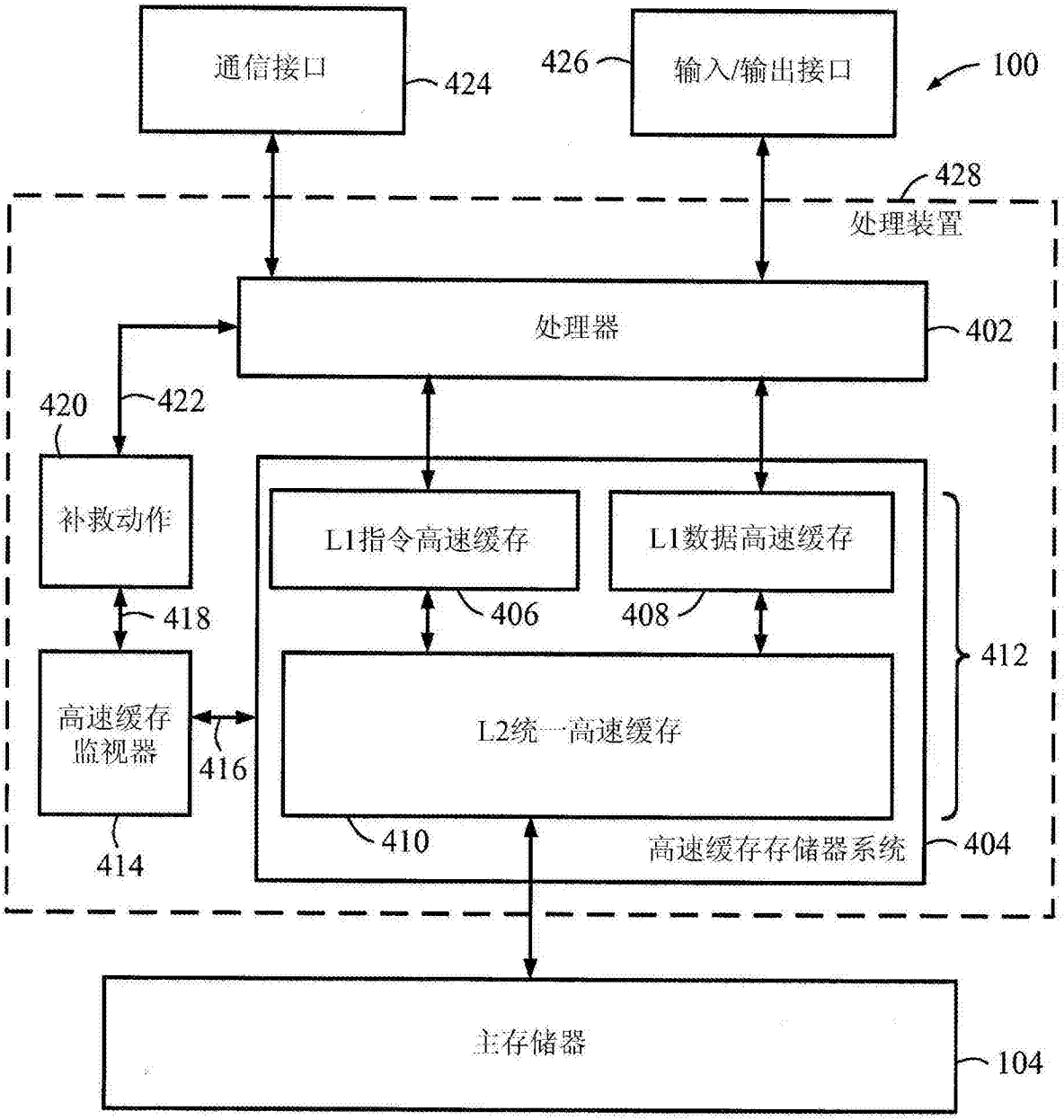

通信接口
424
426
输入/输出接口
100
428
处理装置
处理器
402
420
422
补救动作
L1指令高速缓存
L1数据高速缓存
406
408
418
412
高速缓存
监视器
416
L2统一高速缓存
410
414
高速缓存存储器系统
404
主存储器
104

图4

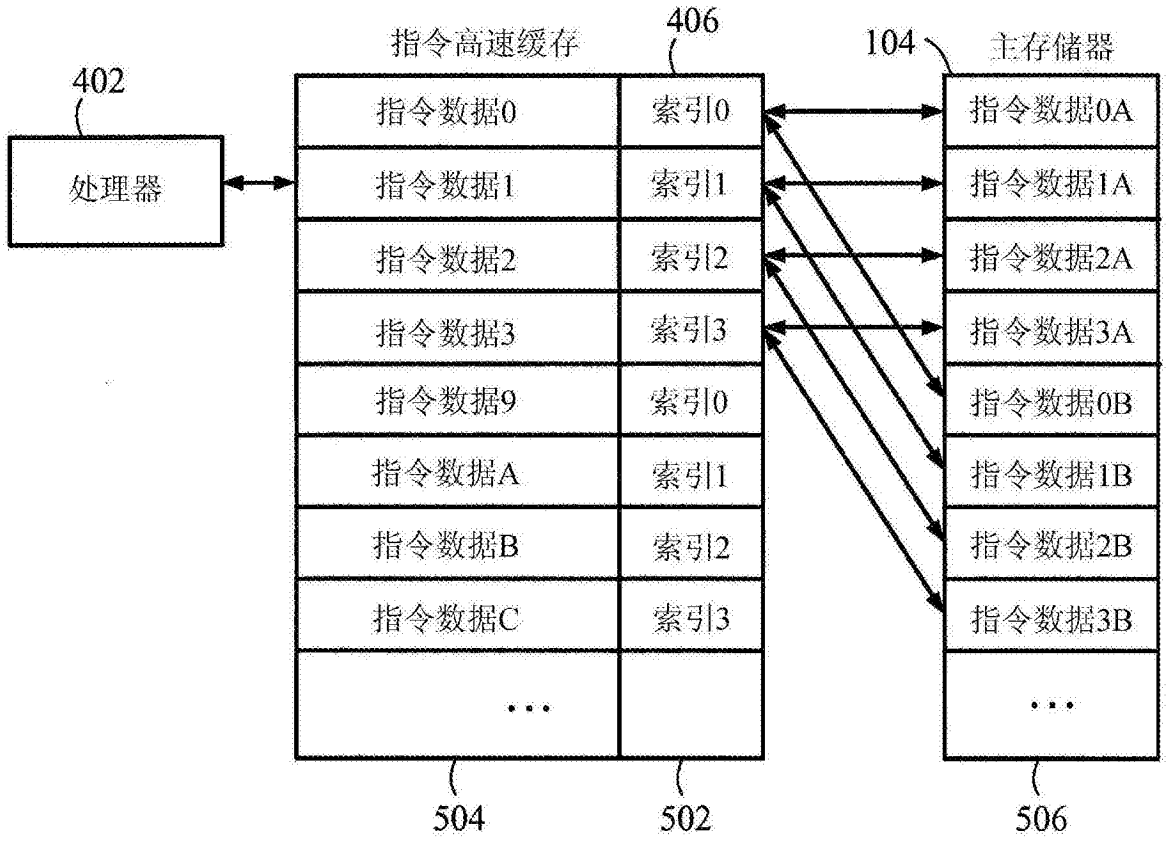

指令高速缓存
406
104
主存储器
402
处理器
指令数据0
索引0
指令数据0A
指令数据1
索引1
指令数据1A
指令数据2
索引2
指令数据2A
指令数据3
索引3
指令数据3A
指令数据9
索引0
指令数据0B
指令数据A
索引1
指令数据1B
指令数据B
索引2
指令数据2B
指令数据C
索引3
指令数据3B
504
502
506

图5

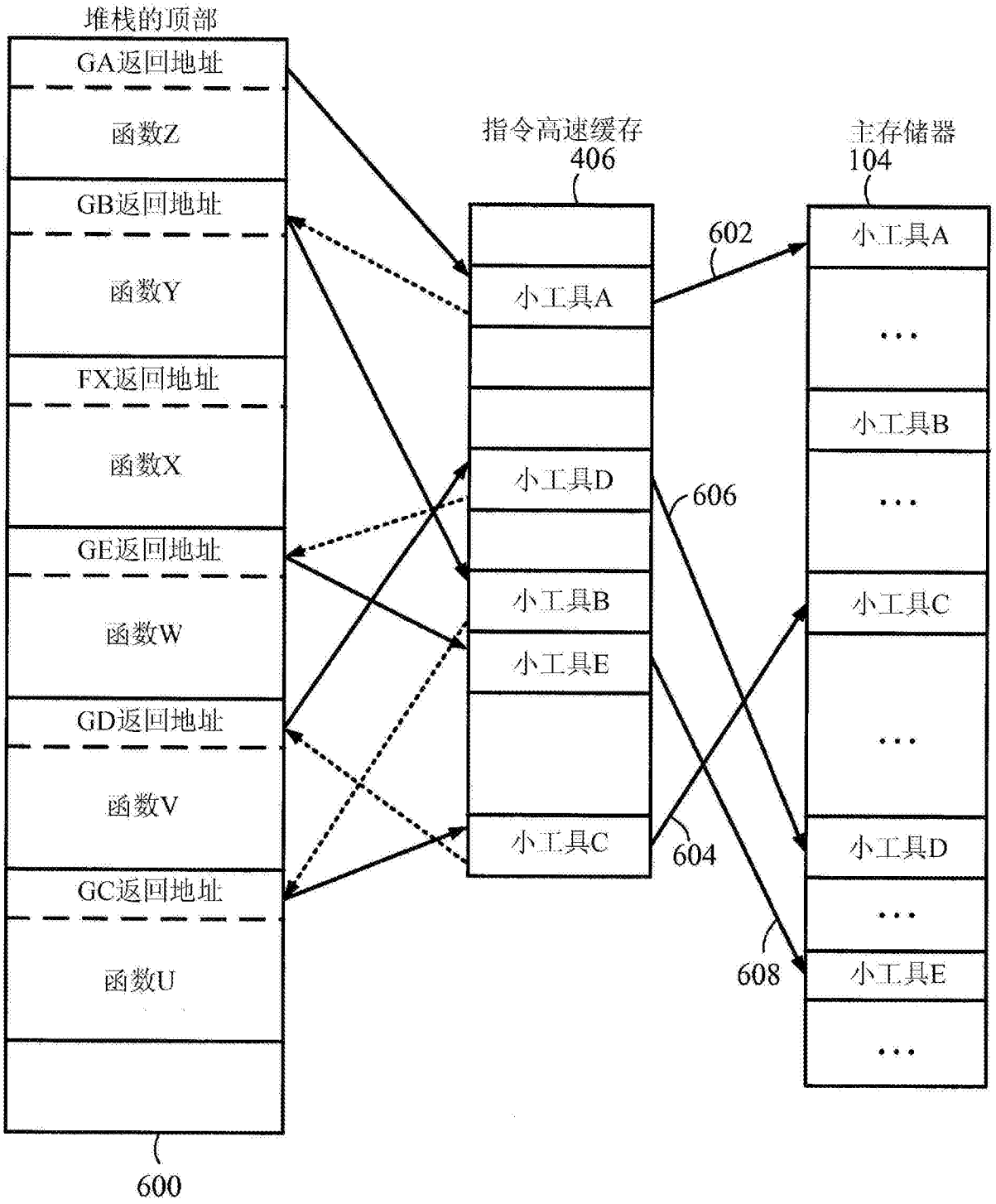

堆栈的顶部
GA返回地址
函数Z
GB返回地址
函数Y
FX返回地址
函数X
GE返回地址
函数W
GD返回地址
函数V
GC返回地址
函数U
600
指令高速缓存
406
小工具A
小工具D
小工具B
小工具E
小工具C
602
606
604
608
主存储器
104
小工具A
小工具B
小工具C
小工具D
小工具E

图6

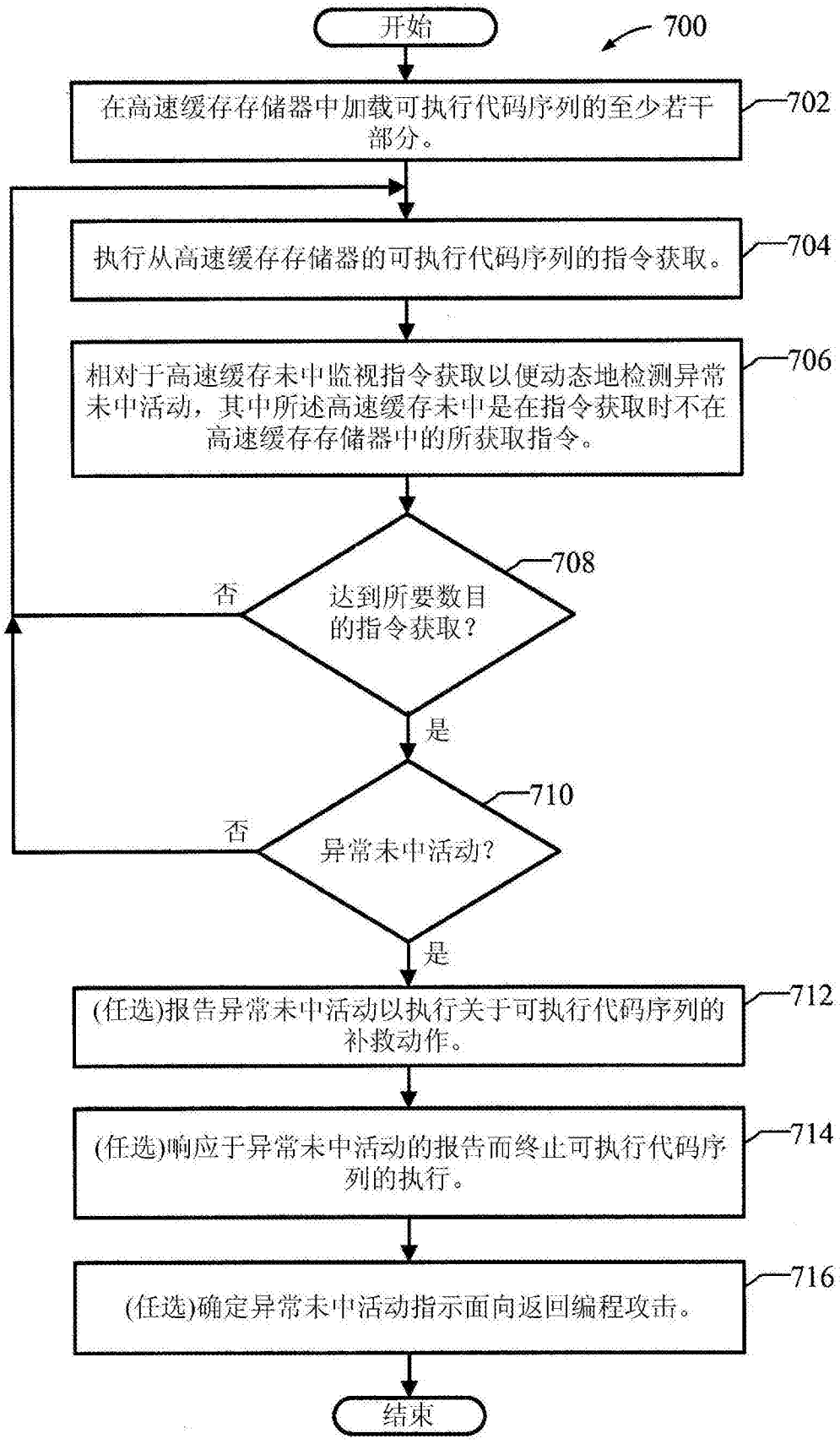

开始
700
在高速缓存存储器中加载可执行代码序列的至少若干部分。
702
执行从高速缓存存储器的可执行代码序列的指令获取。
704
相对于高速缓存未中监视指令获取以便动态地检测异常未中活动，其中所述高速缓存未中是在指令获取时不在高速缓存存储器中的所获取指令。
706
达到所要数目的指令获取？
708
否
是
异常未中活动？
710
否
是
(任选)报告异常未中活动以执行关于可执行代码序列的补救动作。
712
(任选)响应于异常未中活动的报告而终止可执行代码序列的执行。
714
(任选)确定异常未中活动指示面向返回编程攻击。
716
结束

图7

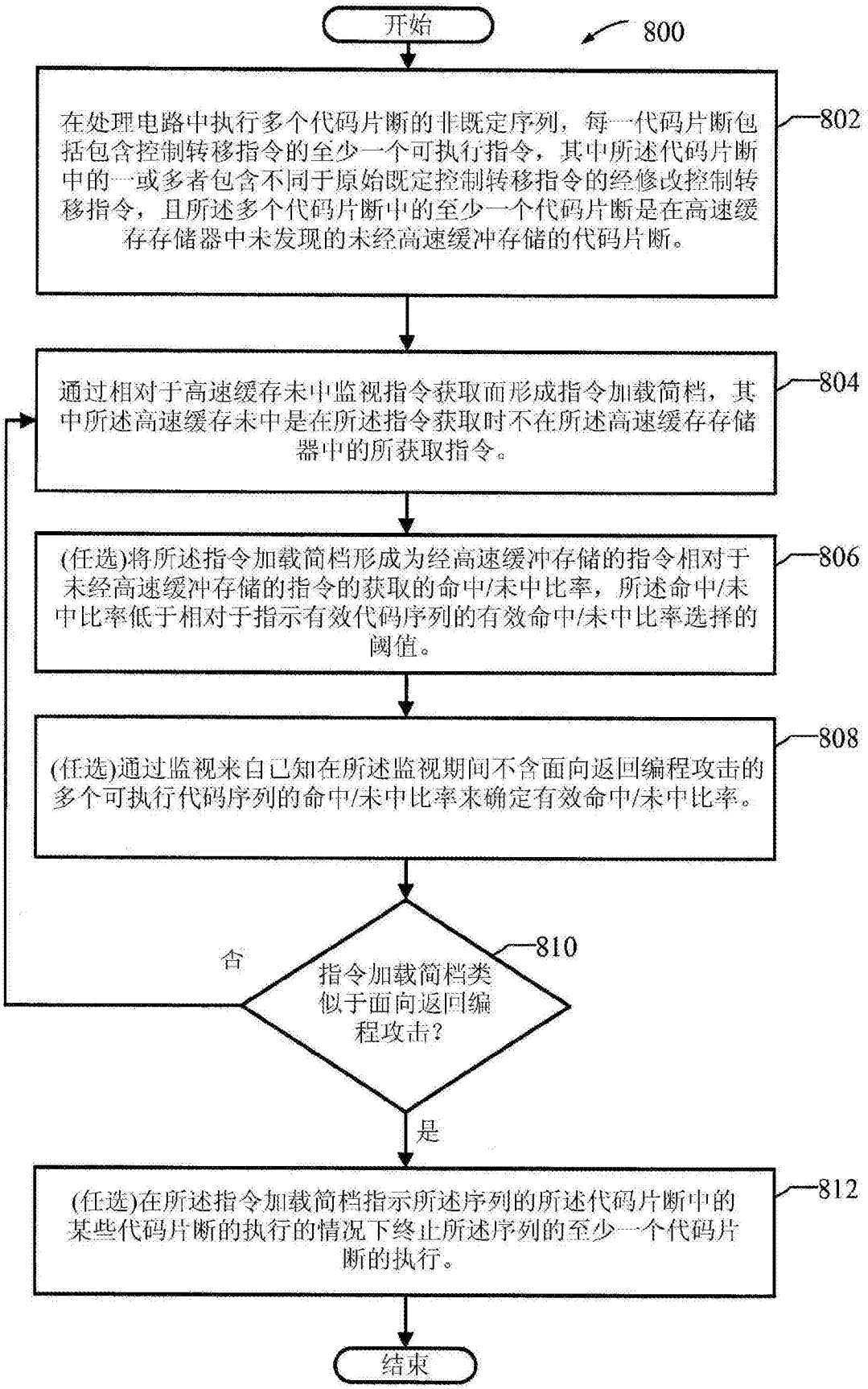

开始
800
在处理电路中执行多个代码片断的非既定序列，每一代码片断包括包含控制转移指令的至少一个可执行指令，其中所述代码片断中的一或多者包含不同于原始既定控制转移指令的经修改控制转移指令，且所述多个代码片断中的至少一个代码片断是在高速缓存存储器中未发现的未经高速缓冲存储的代码片断。
802
通过相对于高速缓存未中监视指令获取而形成指令加载简档，其中所述高速缓存未中是在所述指令获取时不在所述高速缓存存储器中的所获取指令。
804
(任选)将所述指令加载简档形成为经高速缓冲存储的指令相对于未经高速缓冲存储的指令的获取的命中/未中比率，所述命中/未中比率低于相对于指示有效代码序列的有效命中/未中比率选择的阈值。
806
(任选)通过监视来自已知在所述监视期间不含面向返回编程攻击的多个可执行代码序列的命中/未中比率来确定有效命中/未中比率。
808
指令加载简档类似于面向返回编程攻击?
810
否
是
(任选)在所述指令加载简档指示所述序列的所述代码片断中的某些代码片断的执行的情况下终止所述序列的至少一个代码片断的执行。
812
结束

图8

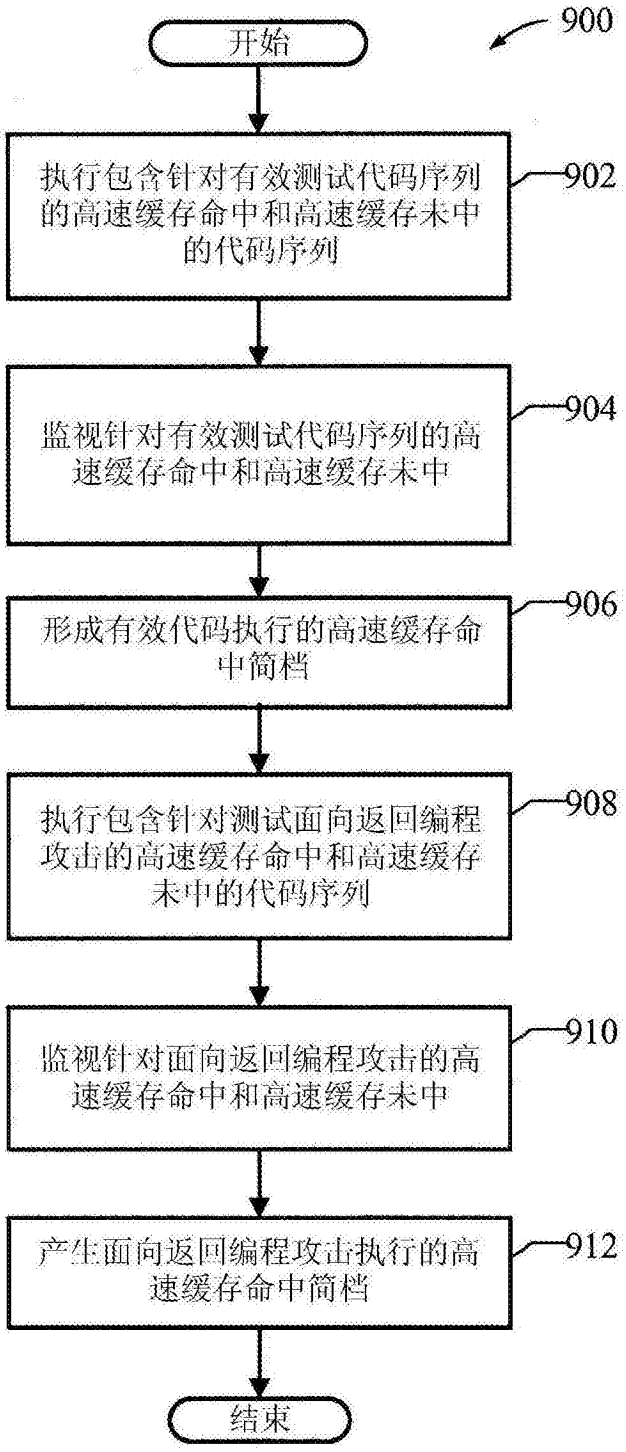

900
开始
执行包含针对有效测试代码序列的高速缓存命中和高速缓存未中的代码序列
902
监视针对有效测试代码序列的高速缓存命中和高速缓存未中
904
形成有效代码执行的高速缓存命中简档
906
执行包含针对测试面向返回编程攻击的高速缓存命中和高速缓存未中的代码序列
908
监视针对面向返回编程攻击的高速缓存命中和高速缓存未中
910
产生面向返回编程攻击执行的高速缓存命中简档
912
结束

图9