(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 103593617 B
(45)授权公告日 2016.08.17

(21)申请号 201310514456.5

(22)申请日 2013.10.27

(73)专利权人 西安电子科技大学
地址 710071 陕西省西安市太白南路2号

(72)发明人 朱辉 李强 陈晓峰 李晖 朱磊 黄橙 雷婉

(74)专利代理机构 陕西电子工业专利中心 61205
代理人 田文英 王品华

(51)Int.Cl.
G06F 21/57(2013.01)
G06F 21/53(2013.01)

(56)对比文件
CN 103093150 A,2013.05.08,
CN 101834860 A,2010.09.15,
US 2009/0172639 A1,2009.07.02,
US 2008/0114985 A1,2008.05.15,
US 2008/0244573 A1,2008.10.02,
US 2010/0023743 A1,2010.01.28,
CN 101593259 A,2009.12.02,
US 2009/0125974 A1,2009.05.14,
US 2008/0235754 A1,2008.09.25,

审查员 张峰

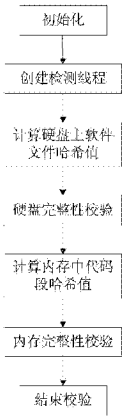
权利要求书1页 说明书4页 附图1页

(54)发明名称

基于VMM的软件完整性校验系统及其方法

(57)摘要

本发明公开一种基于VMM的软件完整性校验系统及方法。系统包括虚拟机和虚拟机监控器，虚拟机中包括初始检测模块和循环检测模块，虚拟机监控器包括硬盘完整性校验模块、内存完整性校验模块以及哈希值计块。方法步骤为：初始化、创建检测线程、计算硬盘上软件文件哈希值、硬盘完整性校验、计算内存中代码段哈希值、内存完整性校验、软件完整性校验结束。本发明通过采用主动陷入虚拟机监控器的方式能够高效的进行完整性校验，通过创建循环检测线程实现了对内存的完整性校验。本发明可用于检测对软件可执行文件本身的恶意修改、软件的防破解，检测内存补丁对软件在内存中映像的修改。



CN 103593617 B

1.一种基于VMM的软件完整性校验系统，包括虚拟机和虚拟机监控器，其特征在于，所述的虚拟机中包括初始化模块和循环检测模块，所述的虚拟机监控器包括硬盘完整性校验模块、内存完整性校验模块以及哈希值计算模块；其中：

所述的初始化模块，用于将当前操作系统迁移到虚拟机监控器上，调用硬盘完整性校验模块和内存完整性校验模块，进行完整性校验；

所述的循环检测模块，用于创建一个检测线程，每隔一定时间调用内存完整性校验模块对内存进行完整性校验，并对检测结果进行判断；

所述的硬盘完整性校验模块，用于调用哈希值计算模块完成软件在硬盘上存储文件的完整性校验，并将校验结果返回给所调用的模块；

所述的内存完整性校验模块，用于调用哈希值计算模块完成软件在内存中代码段映像的完整性校验，并将校验结果返回给所调用的模块；

所述的哈希值计算模块，用于通过给定的字节流，计算出该段字节流的哈希值，为硬盘完整性校验模块和内存完整性校验模块提供经过特定哈希算法运算之后的哈希值。

2.一种基于VMM的软件完整性校验方法，具体步骤如下：

(1)初始化：

初始化模块将操作系统迁移到虚拟机监控器上；

(2)创建检测线程：

2a)在循环检测模块中创建一个检测线程；

2b)检测线程每隔一定时间，进行内存完整性校验；

(3)计算硬盘上软件文件哈希值：

陷入虚拟机监控器，在虚拟机监控器中，读取软件可执行文件字节流，将该字节流传递给哈希值计算模块，计算软件可执行文件在硬盘上的哈希值；

(4)硬盘完整性校验：

在虚拟机监控器中，将哈希值与软件可执行文件的原始哈希值进行比较；对哈希值相同的结果，认为该软件在硬盘上未被修改，硬盘完整性校验通过；对哈希值不同的结果，认为该软件在硬盘上已被修改，硬盘完整性校验失败；

(5)计算内存中代码段哈希值：

陷入虚拟机监控器，在虚拟机监控器中，读取软件可执行文件代码段在内存中映像的字节流，将该字节流传递给哈希值计算模块，计算可执行文件在内存中代码段映像的哈希值；

(6)内存完整性校验：

在虚拟机监控器中，将哈希值与软件可执行文件代码段在内存中映像的原始哈希值进行比较；对哈希值相同的结果，认为该软件在内存中代码段未被修改，内存完整性校验通过；对哈希值不同的结果，认为该软件在内存中代码段已被修改，内存完整性校验失败；

(7)软件完整性校验结束。

3.根据权利要求2所述的基于VMM的软件完整性校验方法，其特征在于，步骤(3)和步骤(5)中所述的虚拟机监控器的陷入是通过硬件虚拟化指令主动陷入的。

4.根据权利要求2所述的基于VMM的软件完整性校验方法，其特征在于，步骤2b)中所述的一定时间的范围为5s～15s。

基于VMM的软件完整性校验系统及其方法

技术领域

[0001] 本发明属于计算机技术领域，更进一步涉及软件安全技术领域中的一种基于虚拟机监控器(Virtual Machine Monitor,VMM)的软件完整性校验系统及其方法。本发明通过调用虚拟化指令将软件的流程转到VMM，在VMM层对软件进行完整性校验，用于检测对软件可执行文件本身的恶意修改、软件的防破解，检测内存补丁对软件在内存中映像的修改。

背景技术

[0002] 随着计算机技术的快速发展，软件已经越来越成为人们生活中不可或缺的一部分。软件规模不断增大，复杂性越来越高，相应的，软件开发投入的资源也越来越多。为了保护软件开发中投入的巨大资金和人力资源，防止软件的破解与非法修改，软件的保护也越来越重要。

[0003] 北京航空航天大学拥有的专利技术“软件完整性验证方法及系统”(专利申请号CN200910087986.X，授权公告号CN101593259B)公开了一种软件完整性验证方法及系统，该方法包括以下步骤：1、在虚拟机监控器中，通过截获上层操作系统的系统调用，以获取加载的可执行文件的路径；2、根据所述可执行文件的路径，获取所述可执行文件的哈希值；3、并将所述可执行文件的哈希值与指纹库中的软件指纹比对；4、根据所述指纹比对结果，对所述可执行文件进行完整性验证。该方法能够实现对操作系统透明的软件完整性校验工作。但是，该方法仍然存在的不足之处是：1、该方法通过在VMM层拦截上层操作系统的系统调用来获取可执行文件的路径，由于现代操作系统中的系统调用很频繁，频繁拦截系统调用会对系统性能造成影响；2、该方法没有对内存中的完整性进行校验，不能防御内存补丁对软件在内存中映像的修改。该系统包括以下模块：1、获取路径模块；2、比对模块；3、完整性验证模块。该系统存在的不足之处是：该系统通过在VMM中截获上层操作系统的系统调用获取可执行文件路径，由于操作系统的系统调用非常频繁，导致被动陷入VMM的频率非常大，影响了系统的性能；该系统没有对软件在内存中的完整性进行校验，不能防御内存补丁对软件在内存中映像的修改。

发明内容

[0004] 本发明针对上述现有软件完整性校验技术的不足，提出一种基于VMM的软件完整性校验方法及系统。本发明具有安全、高效并且对系统性能影响较小的特点，可以准确完成软件的完整性校验，由于实现了内存完整性校验，也能检测到内存补丁对软件在内存中映像的修改。

[0005] 本发明的技术思路是：利用现代CPU支持的硬件虚拟化特性，利用虚拟化指令在操作系统之下增加一个新的软件层，即虚拟机监视器，应用软件通过调用特定的虚拟机指令主动陷入虚拟机监视器，完成软件的完整性校验。由于虚拟机监视器具有最高的权限，能够满足完整性校验的强制性要求；整个验证无需操作系统干预，能够实现对操作系统的透明性；由于是程序主动陷入虚拟机监视器之中，因此不会像拦截系统调用对系统性能造成影

响;考虑到内存补丁可能会对软件在内存中的映像做出修改,破坏软件的完整性,本专利还实现了周期性的内存完整性校验。

[0006] 本发明基于VMM的软件完整性校验系统,包括虚拟机和虚拟机监控器。虚拟机中包括初始化模块和循环检测模块。虚拟机监控器包括硬盘完整性校验模块、内存完整性校验模块以及哈希值计算模块。其中:

[0007] 所述的初始化模块,用于将当前操作系统迁移到虚拟机监控器上,调用硬盘完整性校验模块和内存完整性模块,进行完整性校验。

[0008] 所述的循环检测模块,用于创建一个检测线程,每隔一定时间调用内存完整性校验模块对内存进行完整性校验,并对检测结果进行判断。

[0009] 所述的硬盘完整性校验模块,用于调用哈希值计算模块完成软件在硬盘上存储文件的完整性校验,并将校验结果返回给所调用的模块。

[0010] 所述的内存完整性校验模块,用于调用哈希值计算模块完成软件在内存中代码段映像的完整性校验,并将校验结果返回给所调用的模块。

[0011] 所述的哈希值计算模块,用于通过给定的字节流,计算出该段字节流的哈希值,为硬盘完整性校验模块和内存完整性校验模块提供经过特定哈希算法运算之后的哈希值。

[0012] 本发明的校验方法包括如下步骤:

[0013] (1)初始化:

[0014] 初始化模块将操作系统迁移到虚拟机监控器上。

[0015] (2)创建检测线程:

[0016] 2a)在循环检测模块中创建一个检测线程;

[0017] 2b)检测线程每隔一定时间,进行内存完整性校验。

[0018] (3)计算硬盘上软件文件哈希值:

[0019] 陷入虚拟机监控器,在虚拟机监控器中,读取软件可执行文件字节流,将该字节流传递给哈希值计算模块,计算软件可执行文件在硬盘上的哈希值。

[0020] (4)硬盘完整性校验:

[0021] 在虚拟机监控器中,将哈希值与软件可执行文件的原始哈希值进行比较;对哈希值相同的结果,认为该软件在硬盘上未被修改,硬盘完整性校验通过;对哈希值不同的结果,认为该软件在硬盘上已被修改,硬盘完整性校验失败。

[0022] (5)计算内存中代码段哈希值:

[0023] 陷入虚拟机监控器,在虚拟机监控器中,读取软件可执行文件代码段在内存中映像的字节流,将该字节流传递给哈希值计算模块,计算可执行文件在内存中代码段映像的哈希值。

[0024] (6)内存完整性校验:

[0025] 在虚拟机监控器中,将哈希值与软件可执行文件代码段在内存中映像的原始哈希值进行比较;对哈希值相同的结果,认为该软件在内存中代码段未被修改,内存完整性校验通过;对哈希值不同的结果,认为该软件在内存中代码段已被修改,内存完整性校验失败。

[0026] (7)软件完整性校验结束。

[0027] 本发明与现有技术相比有以下优点:

[0028] 第一,本发明中的方法陷入虚拟机监控器通过采用主动调用虚拟化指令陷入,减

少了现有技术在虚拟机监控器中截获系统调用造成的性能开销，使得本发明的方法具有非常小的性能开销。

[0029] 第二，本发明中的方法创建循环检测线程周期性检测内存完整性，克服了现有技术无法检测到内存补丁修改软件在内存中映像行为的不足，使得本发明的方法能够检测内存补丁动态修改软件在内存中映像的行为。

[0030] 第三，本发明中的系统硬盘完整性模块和内存完整性模块，通过调用硬件虚拟化指令陷入虚拟机监控器，克服了现有系统被动拦截系统调用导致性能开销的不足，使得本发明的系统具有非常小的开销。

[0031] 第四，本发明中的系统循环检测模块，通过创建循环检测线程周期性进行内存完整性校验，克服了现有系统无法检测内存完整性的不足，使得本发明的系统能够检测内存补丁对软件可执行文件在内存中映像的修改。

附图说明

[0032] 图1为本发明系统的方框图；

[0033] 图2为本发明方法的流程图。

具体实施方式

[0034] 下面结合附图对本发明做进一步的详细描述。

[0035] 参照附图1，本发明系统包括虚拟机和虚拟机监控器，虚拟机中包括初始化模块和循环检测模块，虚拟机监控器包括硬盘完整性校验模块、内存完整性校验模块以及哈希值计算模块。其中：

[0036] 初始化模块完成初始化工作，包括检测当前CPU是否支持虚拟化特性，调用CPUID指令开启硬件虚拟化特性，调用硬件虚拟化指令，将操作系统迁移到虚拟机监控器，陷入虚拟机监控器之中，调用硬盘完整性校验模块和内存完整性校验模块。

[0037] 循环检测模块，周期性的陷入虚拟机监控器进行内存完整性的校验，它创建一个循环检测线程，该线程每隔10s调用硬件虚拟化指令，主动陷入虚拟机监控器之中，调用内存完整性校验模块，完成内存完整性的校验工作。

[0038] 硬盘完整性校验模块，通过调用哈希值计算模块完成软件在硬盘上存储文件的完整性校验，并将结果返回给所调用的模块。

[0039] 内存完整性校验模块，通过调用哈希值计算模块完成软件在内存中代码段映像的完整性校验，并将检测结果返回给所调用的模块。

[0040] 哈希值计算模块，通过给定的字节流，计算出这段字节流的哈希值，为硬盘完整性校验模块和内存完整性校验模块提供经过哈希算法运算之后的哈希值，是完整性校验的基础模块。

[0041] 下面结合附图2对本发明方法的具体步骤描述如下：

[0042] 步骤1，初始化。

[0043] 通过调用虚拟化指令，将操作系统迁移到虚拟机监控器上。

[0044] 步骤2，创建检测线程。

[0045] 在循环检测模块中创建一个检测线程，该线程每隔10s调用虚拟化指令，主动陷入

虚拟机监控器。陷入虚拟机监控器后，调用内存完整性校验模块，进行一次内存完整性校验。

[0046] 步骤3，计算硬盘上软件文件哈希值。

[0047] 调用虚拟化指令，主动陷入虚拟机监控器，在虚拟机监控器中，读取该软件在硬盘上的可执行文件，并将该文件文件头之后的代码和数据部分读入一个字节数组中，然后利用这个数组调用哈希值计算模块，计算出可执行文件在硬盘上的哈希值。

[0048] 步骤4，硬盘完整性校验。

[0049] 在虚拟机监控器中，将哈希值与软件可执行文件的原始哈希值进行比较；对哈希值相同的结果，认为该软件在硬盘上未被修改，硬盘完整性校验通过；对哈希值不同的结果，认为该软件在硬盘上已被修改，硬盘完整性校验失败。

[0050] 步骤5，计算内存中代码段哈希值。

[0051] 调用虚拟化指令，主动陷入虚拟机监控器，读取虚拟机中软件代码段在内存中的映像，将这些数据放到一个字节数组中，然后用这个数组作为参数调用哈希值计算模块，计算出软件代码段在内存中映像的哈希值。

[0052] 步骤6，内存完整性校验。

[0053] 在虚拟机监控器中，将哈希值与软件可执行文件代码段在内存中映像的原始哈希值进行比较；对哈希值相同的结果，认为该软件在内存中代码段未被修改，内存完整性校验；对哈希值不同的结果，认为该软件在内存中代码段已被修改，内存完整性校验失败。

[0054] 步骤7，软件完整性校验结束。

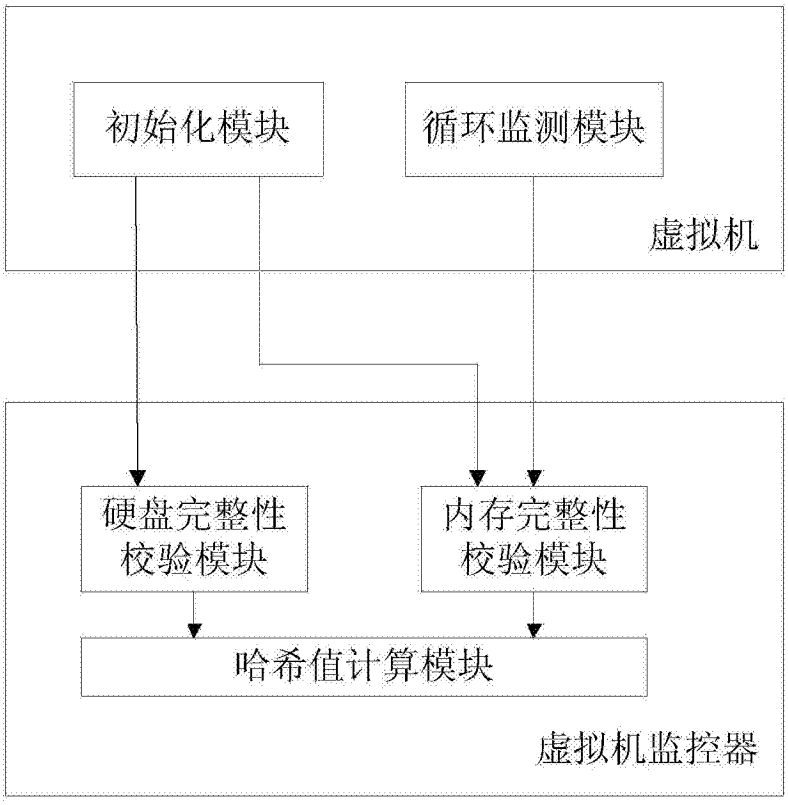

初始化模块
循环监测模块
虚拟机
硬盘完整性校验模块
内存完整性校验模块
哈希值计算模块
虚拟机监控器

图1

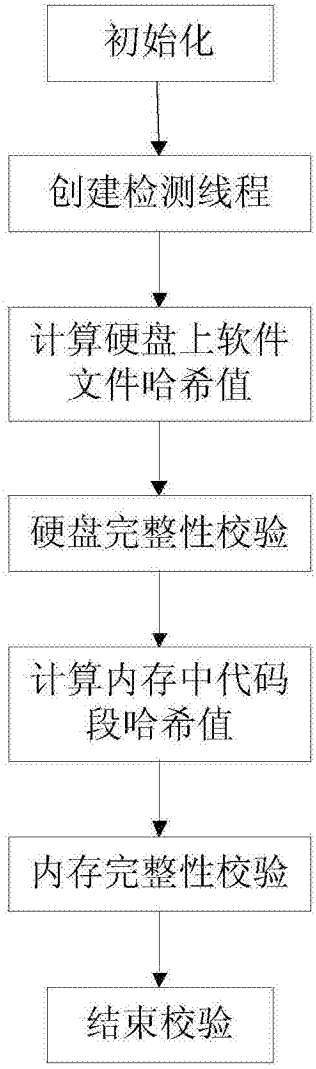

初始化
创建检测线程
计算硬盘上软件文件哈希值
硬盘完整性校验
计算内存中代码段哈希值
内存完整性校验
结束校验

图2