(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 102346828 A
(43)申请公布日 2012.02.08

(21)申请号 201110278710.7

(22)申请日 2011.09.20

(71)申请人 海南意源高科技有限公司
地址 570203 海南省海口市海府路 73 号农建友谊大厦 12 楼
申请人 华中科技大学

(72)发明人 王振江 金海

(51)Int.Cl.
G06F 21/00(2006.01)
G06F 17/30(2006.01)

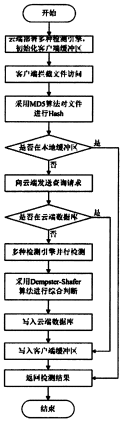
权利要求书 1 页 说明书 6 页 附图 3 页

(54)发明名称

一种基于云安全的恶意程序判断方法

(57)摘要

本发明属于计算机领域，具体涉及一种基于云安全的恶意程序判断方法。本发明通过在云端部署多种不同类型的检测引擎，对用户提交的文件进行并行检测。由于检测引擎具有不同的准确率，检测结果可能各不相同。云端采用综合判断算法对检测结果进行综合判断。为了提高检测的效率，在本地建立文件检测结果缓冲区，同时在云端建立文件检测结果数据库。总体来说，本发明通过多种检测引擎综合判断方法来提高检测结果的正确性，同时客户端采用缓冲区和云端采用数据库来缓存文件检测结果来提高检测过程的高效性。



CN 102346828 A

1. 一种基于云安全的恶意程序判断方法，其特征在于：其步骤为：

a. 在云端部署至少两个不同类型的检测引擎；

b. 在云用户终端运行时，拦截用户程序文件访问或者执行操作，对该文件的唯一标识进行 Hash；

c. 若文件的 Hash 值存在本地缓冲区，则直接返回决策结果；否则，向云端发送查询消息，若找到则返回决策结果，否则通知云用户终端进行上传该文件；

d. 云用户终端上传该文件，将该文件复制多份后，云端启动多种检测引擎进行并行检测，并返回检测结果；

e. 针对各种检测引擎的检测结果，采用综合判断算法进行综合决策，并向云用户终端反馈检测信息；

f. 将该文件的文件名、文件大小、Hash 值、检测结果、查询次数等信息写入云端数据库，方便下次查询。

2. 根据权利要求 1 所述的一种基于云安全的恶意程序判断方法，其特征在于，所述的检测引擎是指对恶意程序进行查杀的开源杀毒软件。

3. 根据权利要求 1 所述的一种基于云安全的恶意程序判断方法，其特征在于，所述的多种检测引擎部署在物理机或虚拟机中，或物理机和虚拟机中皆部署。

4. 根据权利要求 1 所述的一种基于云安全的恶意程序判断方法，其特征在于，所述的在 b 步骤中对该文件的唯一标识采用 MD5 或 SHA1 算法进行 Hash。

5. 根据权利要求 1 所述的一种基于云安全的恶意程序判断方法，其特征在于，所述的 e 步骤中的综合判断算法为 Dempster-Shafer 算法。

一种基于云安全的恶意程序判断方法

技术领域

[0001] 本发明属于计算机领域，具体涉及一种基于云安全的恶意程序判断方法。

背景技术

[0002] 随着计算机及其应用的快速发展和网络结构的复杂化，计算系统的弱点和漏洞将趋于分布式。随着黑客入侵水平的提高，其攻击行为也不再是单一的行为，单个网络安全防御工具在应对分布式、协同式、复杂模式的攻击行为时，就显得十分势单力薄。目前的网络攻击行为的典型特点是：

[0003] (1) 恶意代码数目呈爆炸式增长。目前，全球的恶意程序已经超过 1100 万个，而且这个数据还在处于不断增长中。在 2005 年，每天只有大约 50 种恶意程序特征码被添加到特征库中。而到了 2010 年，这一数字已经增加到了 40000 个。而主流的杀毒软件都是基于病毒特征码，特征码的更新速度远远赶不上新病毒产生的速度。

[0004] (2) 终端有限的资源与庞大恶意程序特征码数量之间的矛盾。虽然计算机的运算速度和存储容量有明显地提高，但是与呈几何级数增长的病毒相比还是远远不够的。用户需要持续不断地更新特征库，才能保证查杀恶意程序的效果。但是随着终端病毒库的增大，需要更多的存储空间，同时消耗了大量计算资源用于杀毒，影响了计算机的运行速度。

[0005] (3) 恶意攻击持续时间长，单个攻击行为不明显。例如，为了刺探某站点的具体信息（如所提供的服务，运行操作系统等信息），多个攻击者协同进行扫描。扫描的持续时间可能会有几天或几个月之久，但每次的扫描活动与正常活动并无明显差异。

[0006] (4) 恶意攻击者范围分布广泛，攻击危害性大。比如，各个攻击者从地理上分散的位置同时向某个网站发动攻击（如拒绝服务攻击），这些攻击积累的结果会导致该网站瘫痪。

[0007] (5) 恶意攻击工具多种多样，攻击者之间及时交流攻击信息，将缩短攻击时间和优化攻击手段。比如各攻击者采用不同的刺探工具，从不同方面获取目标的脆弱点信息，并互相交流，以便优化下一轮攻击措施。

[0008] 面对这些趋势，现有的各自为营的安全防御方法暴露出严重的缺陷。例如，每个终端都安装了入侵防御系统和杀毒软件，但是仍然不能有效地抵御最新的恶意程序。常用的安全组件只能针对独立的入侵行为，而难以防范大规模的、有组织的协同攻击行为；在大规模分布式系统中，各安全组件缺乏协同工作和互动的防御机制。随着大规模协同攻击的危害性日益严重，构建一种能联合各安全组件的可扩展框架是当前的迫切需要。

[0009] 云安全的出现成为解决上述安全问题的有效手段。利用云计算平台强大的处理能力和存储能力，云安全建立专业的信息安全服务平台，它能够集中对信息安全的相关威胁进行处理，提供相应的信息安全服务。在云安全领域，反病毒行业是目前进展较快、影响较大的一个领域。“云杀毒”是病毒防范的一种新模式，它本质上是一种基于互联网的防病毒体系。例如趋势公司采用的是大量服务器作为云端的方式，将复杂的复合式攻击拦截交给云端处理，减轻了用户终端的负担。而瑞星公司采用的是大量用户终端作为云端的方式，将

用户终端作为样本收集机制，实现安全信息的及时发现和共享。目前，已经有多家反病毒公司已经采用相应的技术提供服务，包括卡巴斯基、赛门铁克、趋势、瑞星、金山、江民和奇虎360等。与反病毒领域类似，在防火墙、入侵检测、防垃圾邮件、Web安全等领域，同样也可以利用云计算的方式。防火墙、入侵检测、防垃圾邮件和Web安全的威胁分析服务器可以充分利用云进行动态实时的威胁信息集中采样与共享，从而最终实现主动应变的安全服务。

[0010] 安全厂商在云端部署多个检测引擎，当用户进行系统访问时，客户端进行拦截并上传至云端，由云端检测后向用户反馈检测结果。这种方式改变了传统安全防御的思路，将检测移至云端，实现了恶意程序特征库共享，并降低了用户终端的性能开销。然而，由于商业模式的限制，趋势和瑞星等公司在云端都采用单一类型的多个检测引擎。但是每种检测引擎都有一定的误报率，虽然这种方法能够提高检测的效率，但是并不能提高检测的准确率。此外，由于单一类型的引擎可能存在某种固有缺陷，对某些恶意程序非常有效，其检测的覆盖面非常有限。与此同时，随着客户端病毒库的增大，需要更多的存储空间，同时消耗了大量计算资源用于杀毒，影响了客户端计算机的运行速度。

发明内容

[0011] 本发明的目的在于克服上述不足之处，提供一种基于多种检测引擎的恶意程序判断方法，它具有低开销、多样性、准确性、高效性等特点。

[0012] 本发明的目的是通过如下途径实现的：一种基于云安全的恶意程序判断方法，其步骤为：

[0013] a. 在云端部署至少两个不同类型的检测引擎；

[0014] b. 在云用户终端运行时，拦截用户程序文件访问或者执行操作，对该文件的唯一标识进行Hash；

[0015] c. 若文件的Hash值存在本地缓冲区，则直接返回决策结果；否则，向云端发送查询消息，若找到则返回决策结果，否则通知云用户终端进行上传该文件；

[0016] d. 云用户终端上传该文件，将该文件复制多份后，云端启动多种检测引擎进行并行检测，并返回检测结果；

[0017] e. 针对各种检测引擎的检测结果，采用综合判断算法进行综合决策，并向云用户终端反馈检测信息；

[0018] f. 将该文件的文件名、文件大小、Hash值、检测结果、查询次数等信息写入云端数据库，方便下次查询。

[0019] 更进一步的，所述的检测引擎是指对恶意程序进行查杀的开源杀毒软件。

[0020] 更进一步的，所述的多种检测引擎部署在物理机或虚拟机中，或物理机和虚拟机中皆部署。

[0021] 更进一步的，所述的在b步骤中对该文件的唯一标识采用MD5或SHA1算法进行Hash。

[0022] 更进一步的，所述的e步骤中的综合判断算法为Dempster-Shafer算法。

[0023] 本发明具有以下优点及效果：

[0024] 1. 客户端的的低开销。在用户终端不需要安装任何入侵检测工具或者杀毒软件，只需要安装轻量级客户端。所有的检测功能都在云端实现，云端包含各种恶意程序最新的

特征码。客户端只需要对用户访问的文件进行拦截，并将没有检测过的文件进行上传。同时，采用两级缓冲的方法，提高了查找检测文件的命中率。每个文件只需要上传 1 次，由云端检测后所有用户终端进行共享。因此，客户端只需要进行查询，从而降低了用户终端的性能开销。

[0025] 2. 检测引擎的多样性。由于单一引擎可能对某种类型的恶意程序检测十分有效，但是对于其他类型的恶意程序可能存在缺陷。如果云端采用多个单一类型的检测引擎将会存在一定的限制。本发明提出在云端部署不同类型的多检测引擎，保证检测引擎的多样性，从而提高了检测恶意程序的覆盖面。

[0026] 3. 检测结果的准确性。当采用了多检测引擎进行并行检测后，各个检测引擎的检测结果可能不同，本发明提出了采用综合判断算法（例如 Dempster-Shafer，决策树等）对检测结果进行综合判断，并将判断结果反馈给用户。由于采用了多种引擎进行检测，同时利用综合判断算法，从而提高了检测结果的准确性。

[0027] 4. 检测过程的高效性。当访问或者执行某个文件之前，先对其进行 Hash，根据其 Hash 值来判断该文件是否被检测过。如果已经检测，那么由客户端或者云端直接反馈判断结果；如果没有检测，则上传至云端进行检测。也就是说，对所有用户而言，并不是每次访问文件或者执行程序都需要上传进行检测，只有在没有命中时才上传。当大量用户同时运行系统时，命中率可以高达 95％以上，因此整个检测过程具有极高的效率。

附图说明

[0028] 下面结合附图对本发明作进一步详细说明：

[0029] 图 1 为云模式安全总体架构图；

[0030] 图 2 为云安全的多检测引擎综合判断方法总体流程图；

[0031] 图 3 为本发明的客户端结构图；

[0032] 图 4 为本发明的系统配置实例图。

具体实施方式

[0033] 下面结合附图对本发明进一步作详细的说明。

[0034] 图 1 说明了云模式安全总体架构图。客户端在系统运行过程中对文件访问进行拦截，将文件上传到云端，进行恶意程序判断。云端部署多种类型的检测引擎，对提交的文件进行并行检测和综合判断。所有文件的检测都在云端实现，云端将安全以服务的形式向用户提供。

[0035] 本发明通过在云端部署多种不同类型的检测引擎，对用户提交的文件进行并行检测。由于检测引擎具有不同的准确率，检测结果可能各不相同。云端采用综合判断算法对检测结果进行综合判断。为了提高检测的效率，在本地建立文件检测结果缓冲区，同时在云端建立文件检测结果数据库。总体来说，本发明通过多种检测引擎综合判断方法来提高检测结果的正确性，同时客户端采用缓冲区和云端采用数据库来缓存文件检测结果来提高检测过程的高效性。

[0036] 实施例：如图 2 所示，本发明一种基于云安全的恶意程序判断方法，其步骤为：

[0037] (1) 在云端部署至少两个不同类型的检测引擎，这些检测引擎可以部署在物理机

中，也可以部署在虚拟机中，部署在虚拟机中可以提高资源的利用率；我们使用的检测引擎是指对恶意程序进行查杀的开源杀毒软件。

[0038] (2) 客户端拦截程序的执行操作，计算该程序的唯一标识，该唯一标识可以采用 MD5 算法进行哈希计算得到，也可以采用 SHA1 等类似算法；

[0039] (3) 若该程序的唯一标识值存在于本地缓冲区，则直接返回最终判断结果，转入步骤 (9)。否则，向云端发送查询消息，云端收到查询消息后，在云端数据库中进行查找，若查找到该唯一标识值则向客户端返回最终判断结果，转入步骤 (8)；否则通知客户端上传该文件，进入步骤 (4)；

[0040] (4) 客户端向云端上传该文件；

[0041] (5) 云端采用多个检测引擎，将该文件复制多份，进行并行检测，并返回初步检测结果；

[0042] (6) 云端将每个检测引擎的初步检测结果，采用综合判断算法（如 Dempster-Shafer 算法）进行计算，得到该程序是否为恶意程序的最终判断结果，并向客户端返回最终判断结果；

[0043] (7) 将该文件的唯一标识值、最终判断结果等信息写入云端数据库；

[0044] (8) 将该文件的唯一标识值、最终判断结果等信息写入客户端缓冲区；

[0045] (9) 结束。

[0046] 在步骤 (2) 中，客户端需要对用户终端运行过程中的文件访问进行拦截，图 3 说明了客户端的结构图。客户端包含 2 部分，其中位于内核态的拦截模块实现对系统执行过程中文件访问、加载等操作进行拦截，而位于用户态的部分主要是用户接口，方便用户配置和反馈结果。在拦截到文件后，计算其唯一标识，如利用 MD5 算法对其进行哈希 (Hash)。MD5 算法是一种消息摘要算法 (Message Digest Algorithm)，此算法以任意长度的信息 (Message) 作为输入进行计算，产生一个 128 位 (16 字节) 的指纹或报文摘要。

[0047] 针对步骤 (6) 中的 Dempster-Shafer 综合判断算法，其基本知识：

[0048] 设 Θ 是一个识别框架，在识别框架 Θ 上的基本概率分配 (Basic Probability

[0049]

$$m(\varnothing)=0$$

[0050] Assignment，简称 BPA) 是一个 $2^{\Theta} \rightarrow [0,1]$ 的函数 m，并且满足：

[0051] $$\sum_{A\subseteq\Theta} m(A)=1$$

[0052] 对于 $\forall A \subseteq \Theta$，识别框架 Θ 上的有限个 m 函数 m1，m2，...，mn 的 Dempster 合成规则为：

[0053] $$(m_1 \oplus m_2 \oplus \cdots \oplus m_n)(A)=\frac{1}{K}\sum_{A_1 \cap A_2 \cap \cdots \cap A_n = A} m_1(A_1)\cdot m_2(A_2)\cdots m_n(A_n)$$

[0054]

$$K=\sum_{A_1 \cap \cdots \cap A_n \neq \varnothing} m_1(A_1)\cdot m_2(A_2)\cdots m_n(A_n)$$

[0055] 其中

[0056] 其融合的结果表明经过 n 个主体对识别框架 Θ 中结果为 A 的概率。

[0057] 实例

[0058] 下面举例说明本方法实施过程中的配置情况。

[0059] 为了提高资源的利用率,所有检测引擎都部署在虚拟机中,当然也可以部署在物理机中。首先,在 2 个物理节点上安装虚拟机管理器 -Xen,每个物理节点的硬件及系统配置如表 1 所示。

[0060]

物理节点	CPU	内存	硬盘	操作系统	虚拟机管理器
Node1	2 个 IntelXeonE5310	4GB	160GB	Fedora Core 8	Xen
Node2	2 个 IntelXeonE5310	4GB	160GB	Fedora Core 8	Xen

[0061] 表 1 物理平台的硬件及系统配置

[0062] 在每个物理节点上部署了 1 个管理域和 2 个虚拟机,每个虚拟机中部署 1 种检测引擎。各个检测引擎的基本配置情况如表 2 所示。

[0063]

检测引擎	CPU 数目	内存 (MB)	操作系统	虚拟机标识	物理节点
avast	1	1024MB	Window XP	Domain 1	Node1
avg	1	1024MB	Window XP	Domain 2	Node1
kaspersky	1	1024MB	Window XP	Domain 1	Node2
金山毒霸	1	1024MB	Windows XP	Domain 2	Node2

[0064] 表 2 检测引擎的配置情况

[0065] 图 4 说明了系统配置实例图,每个节点上部署 2 种检测引擎,这些检测引擎都是运行于 Windows 平台上。假设 4 种检测引擎 (avast, avg, kaspersky,金山毒霸) 的准确率 (Probability) 分别为 :Pavast = 0.6,Pavg = 0.85,Pkaspersky = 0.9,Pking = 0.7。每个检测引擎的运行环境都是配置 1 个 CPU 和 1024MB 内存。在虚拟计算平台中,虚拟机的 CPU 数目和内存大小都可以动态调整,从而提高检测效率。

[0066] 在用户终端上安装了轻量级的云安全客户端。当系统运行过程中需要加载动态链接库 netmsg.dll 时,首先通过拦截 Windows 内核态的 ZwCreateSection 函数来获取其文件名。采用 MD5 算法来计算其唯一标识 (Hash 值),查找本地缓冲区。发现其不存在,则发送请求至云端。云端通过查询数据库,发现其已经被检测过。云端返回判断结果 (正常)。客户端允许该动态链接库加载,系统正常运行。在云端显示的结果如表 3 所示。

[0067]

文件名	文件大小 (字节)	MD5 值	检测结果	查询次数
netmsg.dll	245248	ff6726a57d76010c9a963d896bf1fbd4	正常	3

opel9.exe	25600	fe86e69b490f24c975db66215f03db7e	恶意	1

[0068] 表 3 云端检测结果实例

[0069] 当系统运行过程中需要执行文件 opel9.exe 时，通过相同的方法进行拦截。首先度量其唯一标识 (Hash 值)，查找本地缓冲区，发现该文件没有被检测过。然后发送查询请求至云端，发现也不在云端数据库中。因此，客户端将该文件上传至云端，由云端的 4 个检测引擎进行并行检测。检测结果为 avast 和 avg 判断为正常 (Normal)，kaspersky 和金山毒霸判断为恶意 (Malicious)。根据 Dempster-Shafer 算法来进行数据融合。客户端上传的文件存在 2 种互斥的状态 :正常 (N) 和恶意 (M)，而且$N \bigcap M = \varnothing$。识别框架为 Θ = {N, M}。有 4 个证据源的基本概率分别为 :

[0070] S1 :正常概率 mavast(N) = 0.6，恶意概率 mavast(M) = 0.4

[0071] S2 :正常概率 mavg(N) = 0.85，恶意概率 mavg(M) = 0.15

[0072] S3 :正常概率 mkaspersky(N) = 0.1，恶意概率 mkaspersky(M) = 0.9

[0073] S4 :正常概率 mking(N) = 0.3，恶意概率 mking(M) = 0.7

[0074] 计算归一化常数 K :

[0075]

$$K = \sum_{A \cap B \cap C \cap D \neq \varnothing} m_{avast}(A) \square m_{avg}(B) \square m_{kaspersky}(C) \square m_{king}(D)$$

[0076]

$$= m_{avast}(N) \square m_{avg}(N) \square m_{kaspersky}(N) \square m_{king}(N) + m_{avast}(M) \square m_{avg}(M) \square m_{kaspersky}(M) \square m_{king}(M)$$

[0077]

$$= 0.6 \times 0.85 \times 0.1 \times 0.3 + 0.4 \times 0.15 \times 0.9 \times 0.7 = 0.0531$$

[0078] 将 4 种检测引擎的检测结果进行融合，根据 Shafer 融合规则得到恶意程序的概率为 :

[0079]

$$(m_{avast} \oplus m_{avg} \oplus m_{kaspersky} \oplus m_{king})(M) = \frac{1}{K} \sum_{A \cap B \cap C \cap D = M} m_{avast}(M) \square m_{avg}(M) \square m_{kaspersky}(M) \square m_{king}(M)$$

[0080]

$$= \frac{0.4 \times 0.15 \times 0.9 \times 0.7}{0.0531} = 0.712$$

[0081] 因此，根据 4 种引擎的判断结果，云端发现该文件有 0.712 的概率为恶意的。云端向客户端反馈该文件为恶意程序的判断结果，同时更新云端数据库。

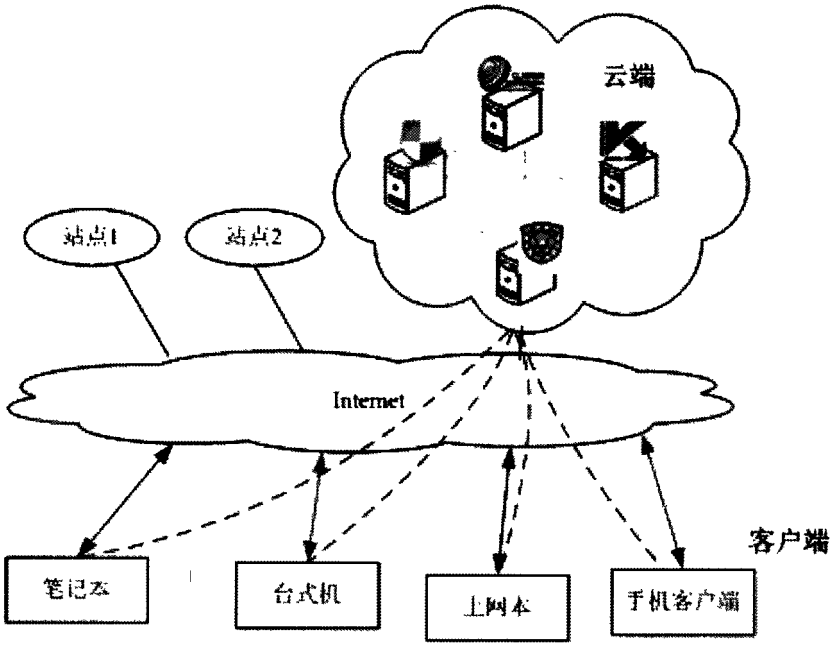


图 1

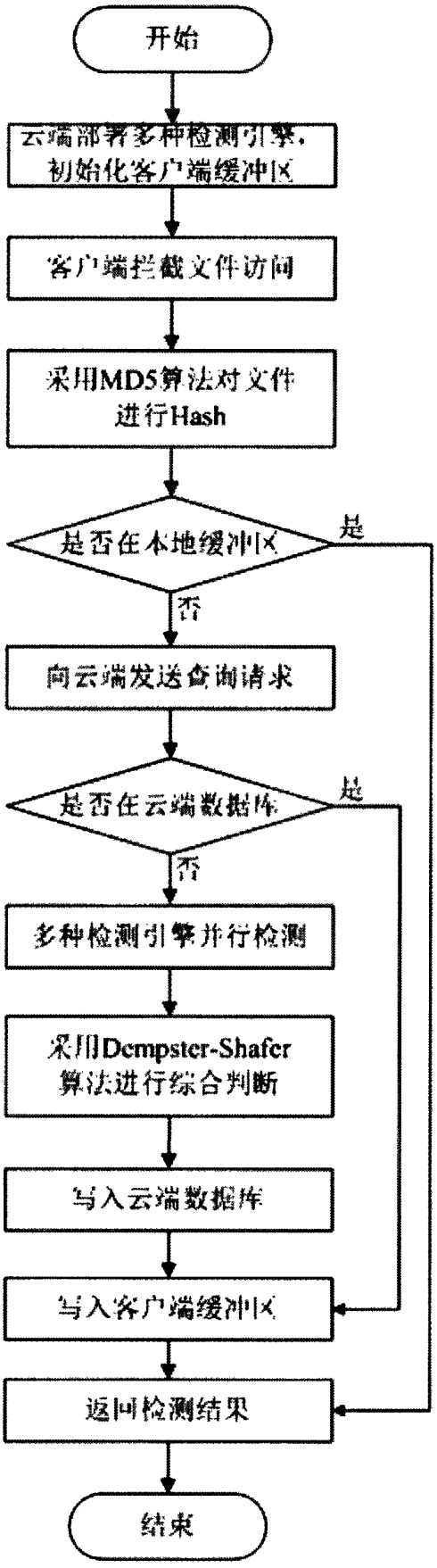

开始
云端部署多种检测引擎，初始化客户端缓冲区
客户端拦截文件访问
采用MD5算法对文件进行Hash
是否在本地缓冲区
是
否
向云端发送查询请求
是否在云端数据库
是
否
多种检测引擎并行检测
采用Dempster-Shafer算法进行综合判断
写入云端数据库
写入客户端缓冲区
返回检测结果
结束

图 2

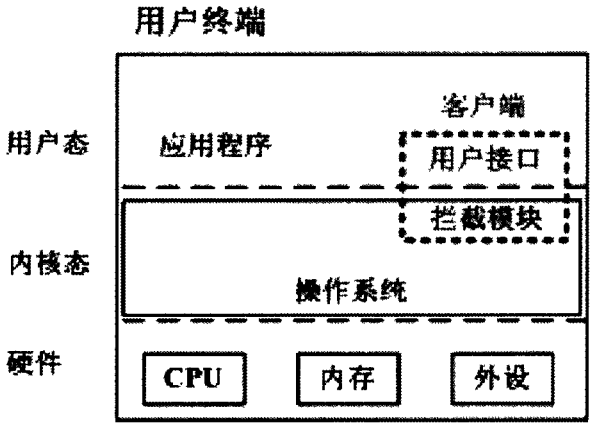

用户终端
用户态
内核态
硬件
应用程序
客户端
用户接口
拦截模块
操作系统
CPU
内存
外设

图 3

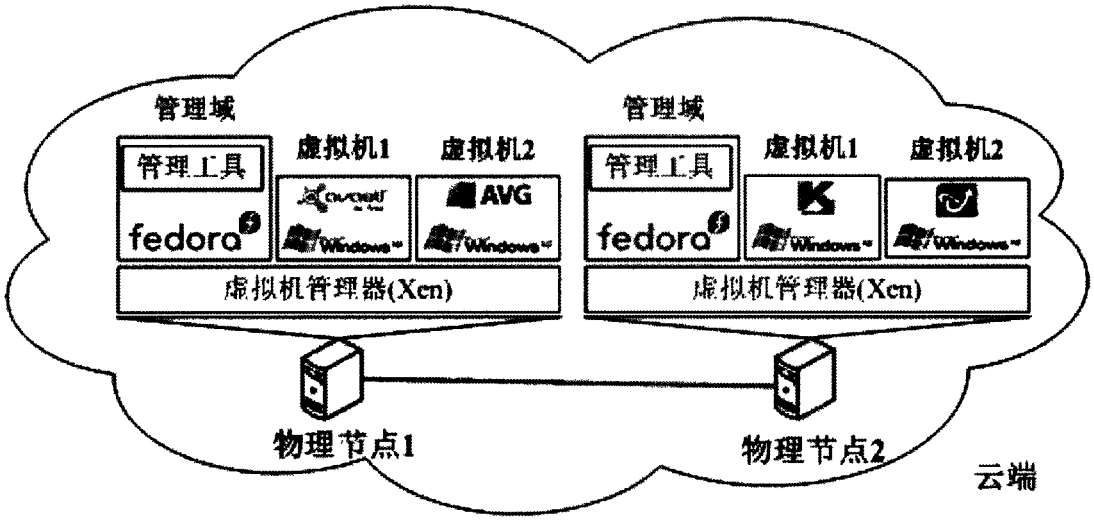

管理域
管理工具
fedora
虚拟机1
虚拟机2
AVG
Windows
虚拟机管理器(Xen)
物理节点1
管理域
管理工具
fedora
虚拟机1
虚拟机2
Windows
虚拟机管理器(Xen)
物理节点2
云端

图 4