



(12) 发明专利

(10) 授权公告号 CN 1574839 B

(45) 授权公告日 2010.05.26

(21) 申请号 200410048884.4

(22) 申请日 2004.06.04

(30) 优先权数据

10/456,766 2003.06.06 US

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 B·D·斯瓦恩德 P·G·枚菲尔德

(74) 专利代理机构 上海专利商标事务所有限公司
31100

代理人 张欣

(51) Int. Cl.

H04L 29/06 (2006.01)

H04L 12/56 (2006.01)

(56) 对比文件

CN 1406351 A, 2003.03.26, 全文.

US 6347376 B1, 2002.02.12, 说明书第 5 栏
第 11 行至第 6 栏第 28 行, 第 8 栏第 44 行至第 9
栏第 58 行.

W0 02/37730 A2, 2002.05.10, 说明书第 5 页
最后一段至第 6 页第 9 行, 第 8 页第 3 行至第 23
行, 第 11 页第 9 行至第 12 页第 4 行, 第 16 页第 10
行至第 17 页第 4 段, 第 19 页第 2.1 节.

US 5835726 A, 1998.11.10, 说明书第 9 栏、
图 5.

审查员 苏宁

权利要求书 3 页 说明书 23 页 附图 8 页

(54) 发明名称

多层防火墙结构

(57) 摘要

本发明提供了一种方法和系统,用于在网络设备中执行防火墙结构。该防火墙结构包括多个网络层、第一个防火墙引擎以及一个或多个呼出模块。这些层将信息包和信息包信息发送到第一个防火墙引擎,保持信息包上下文并将其传递到随后的各个层,并且处理这些信息包。第一个防火墙引擎将该信息包信息与一个或多个被安装的过滤器进行比较,并将动作返回到这些层,指出如何处置该信息包。这些呼出提供额外的功能性(例如,侵入检测、记录和双亲控制特点)。

1. 一种在计算机系统内加以执行的防火墙构架,用于为信息包提供多层过滤,其特征在于,所述防火墙构架包括:

一组层处理器,每个层处理器能够处理和所述层处理器关联的信息包的层参数,并且,每个层处理器还能够发出包括相对应的层参数的分类请求;以及

第一防火墙引擎,它包括:

层接口,用于从请求层处理器接收与其对应的第一层参数,并用于将动作返回到所述请求层,所述请求层处理器是所述层处理器组中正在处理所述信息包的层处理器;

一组被安装的过滤器;以及

查找部件,用于从所述被安装的过滤器组中标识至少一个匹配过滤器,并从所述匹配过滤器中标识将由所述层接口返回的动作,其中,所述请求层从已经处理过所述信息包的前一层处理器那里接收信息包上下文,所述信息包上下文包括和所述前一层对应的第二层参数,所述分类请求包括所述信息包上下文,所述查找部件进一步使用所述信息包上下文来标识所述至少一个匹配过滤器。

2. 如权利要求 1 所述的防火墙构架,其特征在于,所述第一防火墙引擎进一步包括:

呼出接口,用于将这些第一层参数、所述信息包上下文和所述信息包发送到呼出,其中,所述呼出分析所述信息包,并将所述动作返回到所述第一防火墙引擎。

3. 如权利要求 1 所述的防火墙构架,其特征在于,所述请求层将所述信息包上下文传递到将要处理所述信息包的下一层。

4. 如权利要求 1 所述的防火墙构架,其特征在于,所述防火墙构架进一步包括:

策略供应器,用于建立新的过滤器;以及,

第二防火墙引擎,用于将这个新的过滤器加入这个被安装的过滤器组。

5. 如权利要求 1 所述的防火墙构架,其特征在于,所述被安装的过滤器中的每个过滤器包括:

过滤器条件,包括类型-数据对,所述类型定义所述数据的尺寸,所述数据包括过滤器参数;以及,

所述动作。

6. 如权利要求 5 所述的防火墙构架,其特征在于,所述过滤器参数包括一个数值范围。

7. 如权利要求 5 所述的防火墙构架,其特征在于,所述过滤器包括定义所述过滤器的优先级的加权因子。

8. 如权利要求 1 所述的防火墙构架,其特征在于,所述层处理器组进一步包括:

链路层,它具有层参数,这些层参数包括接口号以及源 MAC 地址和目的 MAC 地址;

网络层,它具有层参数,这些层参数包括源和目的 IP 地址;

传输层,它具有层参数,这些层参数包括源端口和目的端口;以及,

应用层,它具有层参数,这些层参数包括数据流。

9. 如权利要求 8 所述的防火墙构架,其特征在于,所述网络层被进一步分成片段层和完全装配层,所述片段层处理 IP 信息包片段,所述完全装配层处理完整的 IP 信息包,所述片段层和所述完全装配层中的每种层发出所述分类请求。

10. 如权利要求 1 所述的防火墙构架,其特征在于,所述第一防火墙引擎按操作系统核心模式来执行。

11. 如权利要求 1 所述的防火墙构架,其特征在于,所述第一防火墙引擎按操作系统用户模式来执行。

12. 一种用于在操作系统中的正在处理信息包的第一层进程与防火墙进程之间进行通信的方法,其特征在于,所述方法包括以下各个步骤:

由所述第一层进程发出具有多个参数的分类请求,所述多个参数包括协议信息包、至少一个层参数和来自第二层进程的信息包上下文,所述第二层进程是已经处理过所述信息包的前一层进程,并且所述信息包上下文包括来自所述第二层进程的第二层参数;

由所述防火墙进程接收所述分类请求;

基于所述分类请求,由所述防火墙进程标识一个或多个匹配过滤器;以及,

由所述防火墙进程基于所标识的一个或多个匹配过滤器发出动作。

13. 如权利要求 12 所述的方法,其特征在于,所述第一层进程是新的进程,进一步包括:

由所述第一层进程发出增加层调用。

14. 如权利要求 12 所述的方法,其特征在于,所述第一层进程是现存的进程,进一步包括:

由所述第一层进程发出删除层调用。

15. 一种用于在操作系统中的防火墙进程与呼出进程之间进行通信的方法,其特征在于,所述方法包括以下各个步骤:

由所述防火墙进程发出具有多个参数的分类请求,所述多个参数包括协议信息包、至少一个层参数、信息包上下文和匹配过滤器标识,所述分类请求接收自正在处理信息包的第一层进程,所述信息包上下文包括来自已经处理过所述信息包的所述第二层进程的第二层参数;

由所述呼出进程接收所述分类请求;以及

由所述呼出进程发出动作,所述动作由所述分类请求中的多个参数加以标识。

16. 如权利要求 15 所述的方法,其特征在于,所述呼出进程是保持可接受的资源位置的高速缓存的父控制模块,进一步包括:

检查所述信息包,以标识资源位置,并且将资源位置与所述可接受的位置进行比较。

17. 如权利要求 16 所述的方法,其特征在于,所述呼出进程所发出的这个动作是阻断,并防止所述协议信息包进行网络遍历。

18. 如权利要求 16 所述的方法,其特征在于,所述呼出进程所发出的这个动作是允许,并允许所述协议信息包进一步穿过网络。

19. 如权利要求 15 所述的方法,其特征在于,所述呼出进程是记录模块,进一步包括:将所述协议信息包存储在存储器中。

20. 如权利要求 15 所述的方法,其特征在于,所述呼出进程是安全模块,进一步包括:

由所述安全模块确定所述协议信息包需符合安全协议;

验证所述协议信息包是符合所述安全协议的。

21. 如权利要求 20 所述的方法,其特征在于,所述安全协议是 IPSec 安全协议。

22. 一种用于允许请求层获得关于信息包的策略的功能接口,所述请求层是多个层中正在处理所述信息包的层,所述功能接口包括:

一种分类模块,所述分类模块包括:

由所述请求层接收的所述信息包;

与所述信息包关联的第一参数集,所述第一参数集包括由所述请求层处理的数据;

由所述请求层从多个层中的另一层接收的信息包上下文,所述另一层是多个层中已经处理过所述信息包的前一层,所述信息包上下文包括所述另一层要处理的第二参数集;以及,

用于将一个动作返回到所述请求层的装置,所述动作标识将要被应用于所述信息包的第一项策略,其中所述动作是基于一个或多个匹配过滤器的,所述一个或多个匹配过滤器是基于所述第一参数集和所述信息包上下文而被标识的。

23. 如权利要求 22 的功能接口,其特征在于进一步包括:

增加层模块,用于允许将新的层加入这多个层。

24. 如权利要求 22 的功能接口,其特征在于进一步包括:

删除层模块,用于从多个层中除去现存的层。

25. 如权利要求 22 的功能接口,其特征在于进一步包括:

策略上下文,定义要被应用于该信息包的第二项策略。

26. 如权利要求 25 的功能接口,其特征在于:其中,所述第一项策略是防火墙策略,所述第二项策略是非防火墙策略。

多层防火墙结构

[0001] 相关申请

[0002] 本申请所包含的主题与跟本申请同日提交的专利申请《用于执行网络防火墙的基于多层的方法》(代理人标签号:221039)、《用于结合多项网络策略的方法和构架》(代理人标签号:221041)和《用于管理基于网络过滤器的策略的方法》(代理人标签号:221037)的主题有关,这些专利申请的揭示说明被特别包括于此,用作参考。

技术领域

[0003] 本发明一般涉及计算机系统和网络安全。更具体地说,本发明涉及在一个或多个网络设备中所执行的防火墙结构。

[0004] 发明背景

[0005] 网络协议的设计用于促进通过公开的数据交换而在各个网络设备之间进行通信。这种公开的数据交换大大增强了网络设备完成任务的用途,同时,它也引起一些问题,这是因为网络协议不是为网络安全性而设计,因此一般不提供网络安全性。被耦合到公用网和专用网(例如,局域网(LANs)、广域网(WANs)、内联网和因特网)的计算机容易受到直接或间接地与该网络耦合的其他网络设备的恶意攻击。这类恶意的攻击包括盗窃数据、“服务拒绝”(DOS)攻击、计算机病毒扩散和类似的攻击。当将计算机耦合到网络时,会出现其他有关的问题(例如,控制儿童对不合需要的或不适当的web站点的访问)。

[0006] 防火墙一般是用来保护用户个人、网络设备和网络避免遭遇恶意攻击的一种工具,同时,也增加了通过实施策略来控制网络上的数据交换的能力。通过检查网络信息包,并通过根据该检查来确定是应该允许还是相反地阻断这些信息包进一步穿过网络,防火墙可执行策略。

[0007] 经由防火墙而加以执行的这项策略由一个或多个过滤器来定义。每个过滤器包括过滤器参数和关联的动作。这些过滤器参数被用来识别受制于该防火墙策略的网络信息包,并包括诸如硬件地址(例如,“媒体访问控制”(MAC)地址)、网络地址(例如,“网际协议”(IP)地址)、协议类型(例如,“传输控制协议”(TCP))、端口号和类似物等信息。该动作定义应该如何处置具有与这些过滤器参数相匹配的参数的信息包。一个特殊的例子是:该过滤器包括“统一资源定位器”(URL)地址(例如,“http://www.foo.com”),作为它的参数。该过滤器进一步使阻断动作(即,丢弃该信息包)与那个URL地址相关联。只要防火墙检查信息包并且通过那个检查而将URL地址“http://www.foo.com”识别为被嵌入该信息包中,服务器就丢弃该信息包,从而防止它穿过网络。

[0008] 通过包括分层网络结构的网络堆栈来发送和接收信息包,网络设备可以交换数据。存在不同的网络结构模型,但大多数至少包括应用层、传输层、网络层和链路层。网络信息包序贯地穿过每个层,并且,当每个层被穿过时,信息包经历处理。关于出站信息包,应用层根据应用协议(例如,其中的一些是:“超文本传输协议”(HTTP)、“文件传输协议”(FTP)和“简单邮件传输协议”(SMTP))来处理数据。其他的层(例如,网络层和传输层)通过将数据嵌入TCP头部和IP头部中,来对其进行分组(packetize)。这些层通过(例如)分

析头部、为数据解除分组 (unpacketize) 等,来为入站信息包执行互换处理。由这些层执行的分层“堆栈”结构和处理功能会产生动态信息包结构,由此,当信息包穿过网络协议堆栈时,包括这些信息包参数的信息包内容会发生变化。

[0009] 防火墙检查位于分层网络堆栈内的检查点处的信息包。一方面,该检查点在教育层处。例如,防火墙被用作“分层服务供应者”(LSP)。应用层处的信息包包括基础数据,该基础数据将被传送到另一个网络设备或者已从另一个网络设备那里被加以接收。通过检查该应用层处的该信息包,可允许防火墙识别应用层参数(例如,URL 地址)并将这些应用层参数与这些过滤器参数进行比较。但是,其他的信息包参数(例如,IP 地址、端口号、MAC 地址和类似物)不可用,这是因为它们要么还没有被加入出站信息包,要么已从入站信息包被分析掉。

[0010] 另一方面,在网络堆栈(作为被置于链路层与网络层之间的中间驱动器)的较低层次处执行防火墙检查点。网络堆栈的这些较低层次处的信息包包括最大数量的参数(例如,接口号、MAC 地址、IP 地址、协议类型、端口和有效载荷数据)。虽然这些信息包包括这类参数,但是,这并不表示认为:可以容易地识别这些参数。在该防火墙接收该信息包之后,该防火墙需要分析并解释该有关的信息包参数,用于和这些过滤器参数进行比较。这样,网络堆栈中的这些层以及该防火墙两者都执行多余的信息包分析和解释功能。

[0011] 发明概述

[0012] 本发明针对一种防火墙结构,该防火墙结构允许将防火墙过滤器应用于网络堆栈内的所有层处的网络信息包。该结构包括能够处理来自网络信息包的层参数的设置层处理器。这些层参数是与该层处理器关联的参数;这些层参数包括该层处理器从该网络信息包中分析的、加入该网络信息包的或从该网络信息包中导出的参数。这些层处理器进一步能够发出包括这些层参数的分类请求。

[0013] 该结构也包括第一个防火墙引擎,该防火墙引擎包括层接口、一组被安装的过滤器和查找部件。该层接口接收作为该分类请求的一部分而被加以发送的这些层参数。这第一个防火墙引擎查找部件使用这些层参数来从这个被安装的过滤器组中识别一个或多个匹配过滤器。每个匹配过滤器包括一个动作,该动作指示该层处理器:是否允许该信息包进一步穿过网络,或者是否阻断信息包。这第一个防火墙引擎经由该层接口而至少将一个动作返回到该层处理器。

[0014] 在本发明的实施例中,层处理器保持信息包上下文。层处理器从前一层处理器那里接收信息包上下文,并将信息包上下文发送到第一个防火墙引擎。第一个防火墙引擎中的该查找部件结合这些层参数来使用信息包上下文,以识别这一个或多个匹配过滤器。该层处理器也通过增加这些层参数、然后将修改过的信息包上下文发送到下一层处理器,来修改信息包上下文。

[0015] 在本发明的另一个实施例中,一个或多个呼出(callout)被包括在内,作为该防火墙结构的一部分。这一个或多个呼出模块中的每个呼出模块提供(允许和阻断以外的)增加的功能性,例如信息包记录功能、“网际协议安全性”(IPSec)验证和双亲控制的执行。当呼出模块被识别为这些匹配过滤器之一中的那个动作时,在该信息包上执行该呼出模块。

[0016] 在本发明的另一个实施例中,提供了第二个防火墙引擎。第二个防火墙引擎经由

管理 API 而将新的过滤器加入这个被安装的过滤器组。第二个防火墙引擎也包括第一个防火墙引擎的实例，它被称作“过滤器模块”，该过滤器模块在第二个防火墙引擎内复制第一个防火墙引擎的各项服务。

[0017] 通过以下参照附图而对说明性实施例的详细描述，本发明的额外的特点和优点将会变得一目了然。

[0018] 附图简述

[0019] 所附如权利要求书详细地陈述了本发明的这些特点，但通过以下结合附图的详细说明，可以最佳程度地理解本发明及其目的和优点。在这些附图中：

[0020] 图 1 是框图，大体说明了其上驻留本发明的示范计算机系统；

[0021] 图 2 是框图，大体说明了使用本发明的示范网络环境；

[0022] 图 3 是框图，大体说明了本发明的防火墙结构；

[0023] 图 4 是框图，说明了用于本发明的示范过滤器；

[0024] 图 5 是框图，说明了用于本发明的示范信息包上下文的数据结构；

[0025] 图 6 是框图，说明了用于本发明的示范应用编程接口组；

[0026] 图 7 是框图，说明了用于本发明的示范应用编程接口；

[0027] 图 8 是框图，说明了根据本发明的、由网络层来执行的功能；以及，

[0028] 图 9 是框图，说明了用于本发明的示范呼出组。

[0029] 发明的详细说明

[0030] 揭示了一种防火墙结构，该防火墙结构允许多层处理（在这里被称作“层”）处的过滤。该结构包括用户模式处理和操作系统中执行的核心模式处理。作为选择，在操作系统以外的一个或多个程序模块中或在单一操作系统模式以内执行该防火墙结构。

[0031] 该核心模式处理包括多个层，这多个层包括协议堆栈、核心防火墙引擎以及一个或多个呼出。协议堆栈包括应用层、传输层、网络层和链路层。按需要增加或从该结构中删除额外的层。这些层每个都构成请求层，请求层能够从前一个层或处理那里接收网络信息包和对应的信息包上下文数据。然后，请求层经由层 API，将分类请求发给核心防火墙引擎。分类请求包括由请求层接收的信息包、该信息包上下文和与请求层关联的一组层参数。核心防火墙引擎处理该请求，并返回动作。举例来讲，该动作指示该请求层如何处置该信息包（例如，允许或阻断）。如果该动作是允许，则该请求层根据层协议来处理该信息包，将该信息包上下文修改成包括这些层参数，并将信息包和信息包上下文传递到下一个层。如果该动作是阻断，则请求层丢弃信息包，并且不将信息包传递到这下一个层。作为阻断动作的结果，请求层可能会执行额外的功能（例如，在丢弃 TCP 信息包的情况下，拆毁 TCP 连接）。

[0032] 核心防火墙引擎包括层 API、一组被安装的过滤器和呼出 API。这个被安装的过滤器组中的每个过滤器包括一组过滤器条件和一个关联的动作。核心防火墙引擎通过识别一个或多个匹配过滤器，来处理从该请求层那里被发送的分类请求。这些匹配过滤器具有跟这些层参数和信息包上下文相匹配的过滤器条件。一旦识别这些匹配过滤器，就按过滤器优先级的顺序来应用它们。如果正在被应用的过滤器的这个动作是允许或阻断，则将这个动作返回到请求层。如果该动作是呼出，则将该请求层所发出的该分类请求连同匹配过滤器标识一起传递到这些呼出模块中的一个呼出模块。该呼出模块执行其被编程的功能，并将动作返回到核心防火墙引擎。

[0033] 核心防火墙引擎按优先级顺序将匹配过滤器应用于信息包,直到至少指定终止动作为止。一旦指定该终止动作,就将它返回到请求层,并且不应用额外的匹配过滤器。如果没有为信息包识别匹配过滤器,则向该请求层通知:不曾发现匹配过滤器;然后,请求层决定如何处置信息包。通常,在没有识别匹配过滤器的情况下,请求层处置信息包,好象返回允许动作一样。

[0034] 示范用户模式处理包括用户模式防火墙引擎以及一个或多个策略供应器。这些策略供应器从任何合适的来源(例如,易失或非易失存储器)获得策略,或者通过图形用户界面或类似的工具而被用户输入。该策略是用于呈现新的过滤器的信息源,包括过滤器条件和关联动作的集合。

[0035] 该用户模式也包括该用户防火墙引擎内的核心防火墙引擎的实例,从而允许创建用户模式层。然后,用户模式层使用核心防火墙引擎的用户模式实例来安装过滤器,并且识别与一组参数相匹配的过滤器,这组参数允许在该用户模式以内应用过滤。

[0036] 在本发明的实施例中,从防火墙引擎到一组呼出模块的呼出接口实质上允许这些防火墙性能的无限扩展。举例来讲,HTTP 上下文呼出通过识别可接受的和不能接受的 URL 地址,来提供双亲控制特点。IPSec 呼出验证:应该使用 IPSec 的信息包已经适当地经历 IPSec 处理。记录呼出记入符合所建立的标准的信息包,从而促进以后对信息包的检查。侵入检测呼出根据已知的算法来识别可疑的信息包。

[0037] 参考这些附图(其中,相似的参考数字提及相似的元件),本发明被展示为在合适的计算环境中加以执行。虽然未作要求,但是,将在正由个人计算机执行的计算机可执行指令(例如,程序模块)的一般上下文中描述本发明。通常,程序模块包括执行特殊任务或实施特殊的抽象数据类型的例行程序、程序、对象、部件、数据结构等。也可以在分布式计算环境中实践本发明,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于本地记忆存储设备和远程记忆存储设备中。

[0038] 图 1 展示了可以在其上执行本发明的合适的计算系统环境 100 的例子。计算系统环境 100 只是合适的计算环境的一个例子,它并不意在对本发明的使用或功能性的范围提出任何限制。也不应该将计算环境 100 解释为具有涉及示范操作环境 100 中所展示的任何部件或部件组合的任何依赖性要求。

[0039] 本发明可用于众多其他的通用或专用计算系统环境或配置。可能适用于本发明的众所周知的计算系统、环境和/或配置的例子包括(但不限于)个人计算机、服务器计算机、手持设备或便携式设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络 PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境,以及类似物。

[0040] 可以在正由计算机执行的计算机可执行指令(例如,程序模块)的一般上下文中描述本发明。通常,程序模块包括执行特殊任务或实施特殊的抽象数据类型的例行程序、程序、对象、部件、数据结构等。也可以在分布式计算环境中实践本发明,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括记忆存储设备的本地计算机存储介质和远程计算机存储介质中。

[0041] 参照图 1,用于执行本发明的示范系统包括采取计算机 110 的形式的通用计算设

备。计算机 110 的部件可以包括（但不局限于）处理单元 120、系统存储器 130 和系统总线 121，系统总线 121 将包括系统存储器的各种系统部件耦合到处理单元 120。系统总线 121 可以是几种类型的总线结构（包括存储总线或存储控制器、外围总线和使用各种总线构造中的任何总线构造的局域总线）中的任何总线结构。举例来讲（不作限制），这类结构包括“工业标准结构”（ISA）总线、“微通道结构”（MCA）总线、“增强的 ISA”（EISA）总线、“视频电子标准协会”（VESA）局域总线和也被称作“中层楼（Mezzanine）总线”的“外围部件互连”（PCI）总线。

[0042] 计算机 110 通常包括各种计算机可读介质。计算机可读介质可以是可由计算机 110 存取的任何可用介质，它包括易失和非易失介质、可移动和不可移动的介质。举例来讲（不作限制），计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括易失和非易失的可移动和不可移动的介质，这些介质用关于信息（例如，计算机可读指令、数据结构、程序模块或其他数据）存储的任何方法或技术来加以执行。计算机存储介质包括（但不局限于）RAM、ROM、EEPROM、快闪存储器或其他存储技术、CD-ROM、数字通用光盘（DVD）或其他光盘存储器、盒式磁带、磁带、磁盘存储器或其他磁性存储设备、或可以被用来存储所需信息并可以由计算机 110 来进行存取的其他任何介质。通信介质通常具体表现为在调制数据信号（例如，载波或其他传送机制）中的计算机可读指令、数据结构、程序模块或其他数据，它包括任何信息传递介质。术语“调制数据信号”意味着一种信号，该信号的一个或多个特征按这样的方式来加以设置或更改，以便为该信号中的信息编码。举例来讲（不作限制），通信介质包括有线介质（例如，有线网络或直线连接）和无线介质（例如，声音、RF、红外线和其他无线介质）。以上任何内容的组合也应该被包括在计算机可读介质的范围以内。

[0043] 系统存储器 130 包括采取易失和 / 或非易失存储器（例如，只读存储器（ROM）131 和随机存取存储器（RAM）132）的形式的计算机存储介质。基本输入 / 输出系统 133（BIOS）通常被存储在 ROM 131 中，该基本输入 / 输出系统包含有助于在计算机 110 内的各个元件之间传送信息（例如，在启动期间）的这些基本例行程序。RAM 132 通常包含可立即由处理单元 120 存取并且 / 或者目前正由处理单元 120 进行操作的数据和 / 或程序模块。举例来讲（不作限制），图 1 展示了操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。

[0044] 计算机 110 也可以包括其他可移动 / 不可移动的易失 / 非易失计算机存储介质。只举例来讲，图 1 展示了从不可移动的非易失磁性介质读取或对其写入的硬盘驱动器 141、从可移动的非易失磁盘 152 读取或对其写入的磁盘驱动器 151，以及从可移动的非易失光盘 156（例如，CD ROM 或其他光学介质）读取或对其写入的光盘驱动器 155。可以被用于该示范操作环境中的其他可移动 / 不可移动的易失 / 非易失计算机存储介质包括（但不局限于）卡型盒式磁带机、快闪存储卡、数字通用光盘、数字录像带、固态 RAM、固态 ROM 和类似的存储介质。硬盘驱动器 141 通常通过不可移动的存储接口（例如，接口 140）而被连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常由可移动的存储接口（例如，接口 150）连接到系统总线 121。

[0045] 以上所讨论的和图 1 中所展示的这些驱动器及其关联的计算机存储介质为计算机 110 提供计算机可读指令、数据结构、程序模块和其他数据的存储。在图 1 中，例如，硬盘

驱动器 141 被展示为存储操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。注意,这些部件可以等同于或不同于操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。这里为操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 提供不同的号码,以展示:它们至少是不同的副本。用户可以通过输入设备(例如,键盘 162)和定点设备 161(通常被称作“鼠标”、“跟踪球”或“触垫”),来将命令和信息输入计算机 110。其他输入设备(未示出)可以包括话筒、操纵杆、游戏垫、圆盘式卫星电视天线、扫描仪或类似的输入设备。这些和其他的输入设备经常通过被耦合到该系统总线的用户输入接口 160 而被连接到处理单元 120,但也可以由其他接口和总线结构(例如,并行端口、游戏端口或通用串行总线(USB))来加以连接。监视器 191 或其他类型的显示设备也经由接口(例如,视频接口 190)而被连接到系统总线 121。除该监视器以外,计算机也可以包括其他外围输出设备(例如,扬声器 197 和打印机 196),这些外围输出设备可以通过输出外围接口 195 来加以连接。

[0046] 计算机 110 可以在使用与一台或多台远程计算机(例如,远程计算机 180)的逻辑连接的联网环境中进行操作。远程计算机 180 可以是另一台个人计算机、服务器、路由器、网络 PC、对等设备或其他共同的网络节点,它通常包括以上相对于计算机 110 而描述的许多或所有这些元件,尽管图 1 中只展示了记忆存储设备 181。图 1 中所描绘的这些逻辑连接包括局域网(LAN) 171 和广域网(WAN) 173,但也可以包括其他网络。这类联网环境在办公室、企业范围的计算机网络、内联网和因特网中很普遍。

[0047] 当被用于 LAN 联网环境中时,计算机 110 通过网络接口或适配器 170 而被连接到 LAN 171。当被用于 WAN 联网环境中时,计算机 110 通常包括调制解调器 172 或用于在 WAN 173(例如,因特网)上建立通信的其他装置。调制解调器 172(可能是内置的,也可能是外置的)可以经由用户输入接口 160 或其他合适的机制而被连接到系统总线 121。在联网环境中,相对于个人计算机 110 或其各个部分而描绘的程序模块可以被存储在该远程记忆存储设备中。举例来讲(不作限制),图 1 将远程应用程序 185 展示为驻留在存储设备 181 上。将会理解:所示的这些网络连接起示范的作用,可以使用在各台计算机之间建立通信链路的其他装置。

[0048] 在下文中,除非另有指示,将参照动作和一台或多台计算机所执行的操作的符号表示来描述本发明。照此,将会理解:这类动作和操作(有时被称作“是计算机执行的”)包括用结构化形式来表现数据的电信号的计算机的处理单元所执行的操作。这项操作变换该数据或将其保存在计算机的存储系统中的各个位置,这样,可按精通该技术领域的人所熟悉的方式来重新配置或改变计算机的这个操作。那里保存有数据的这些数据结构是存储器的物理位置,存储器具有由该数据的格式定义的特定属性。但是,尽管前述上下文中描述本发明,但这并不意在起限制的作用,如精通该技术领域的人将会理解的,也可以在硬件中执行下文中所描述的这各种动作和操作。

[0049] 现在,将参照图 2 来描述其中使用本发明的防火墙结构的一种网络环境。该网络本质上起示范的作用,因为可在被耦合到任何网络配置的任何网络设备中执行本发明的防火墙结构。该网络环境包括专用网 200 和公用网 202。专用网 200 和公用网 202 属于任何合适的类型(例如,局域网(LANs)、广域网(WANs)、内联网、因特网或其任何组合)。

[0050] 该网络环境包括多个网络设备 204、206、208、210 和 212。网络设备 204、206 被耦

合到专用网 200。网络设备 210、212 被耦合到公用网 202。网络设备 208 被耦合到专用网 200 和公用网 202,并在这两个网络之间提供接口。这些网络设备被耦合到使用任何合适的技术(例如,以太网、1394 或 802.11(b))的公用网和专用网。这些网络设备被进一步加以执行,作为任何合适的计算设备(例如,个人计算机、服务器、手持设备、打印机、交换器、路由器、桥接器、转发器或类似的设备)。

[0051] 网络设备 208 包括防火墙 214 以及一个或多个过滤器 216。防火墙 214 是根据本发明的结构来加以执行的一个程序模块或一组程序模块,这些程序模块检查在被耦合到专用网 200 的网络设备 204、206、208 与被耦合到公用网 202 的网络设备 210、212 之间被交换的网络信息包。在本发明的实施例中,防火墙 214 也检查从专用网 200 内的网络设备那里被发送的以本地为目的地的网络信息包以及去往那些网络设备的网络信息包。

[0052] 防火墙 214 在网络设备 208 中被加以执行,以保护并控制在专用网 200 与公用网 202 之间被交换的网络信息流通量,这被称作“边缘防火墙”。作为选择,防火墙 214 在如网络设备 210 中所展示的单一网络设备中被加以执行,并保护该单一网络设备,这被称作“主机防火墙”。该防火墙也能够按同步方式、作为主机和/或边缘防火墙的中央管理集来加以执行,这被称作“分布式防火墙”。较佳地选择执行防火墙 214 的这个或这些网络设备的布置,以便防火墙 214 检查它应该保护的、以这些网络设备为目的地的所有网络信息流通量。

[0053] 过滤器 216 作为防火墙 214 的一部分来加以执行。作为选择,过滤器 216 作为防火墙 214 可以使用的单独的数据结构的一部分来加以执行。防火墙 214 和过滤器 216 执行一种防火墙策略,该防火墙策略被设计成保护网络设备 204、206、208 避免遭遇来源于被耦合到该公用网的网络设备 210、212 的恶意攻击。防火墙 214 也提供增加的功能性(例如,促进双亲控制、侵入检测、记录网络信息包和其他基于增加的过滤器的功能性)。

[0054] 每个过滤器 216 包括一组过滤器条件以及一个或多个关联的动作。这些过滤器条件包括参数和信息,这些参数和信息可以从网络信息包(例如,接口号、硬件地址、网络地址、协议类型、端口号和有效载荷数据)中被加以分析或从网络信息包中获得。这一个或多个关联的动作定义执行该防火墙的该网络设备应该如何处置与这些过滤器条件相匹配的信息包。典型的动作包括允许(即,允许该信息包继续网络遍历)和阻断(即,通过丢弃信息包,来排除进一步的网络遍历)。

[0055] 防火墙 214 检查当在网络设备 208 处被接收时穿过网络的网络信息包;并且,通过将这些信息包参数与这些过滤器条件进行比较,防火墙 214 识别一个或多个匹配过滤器。当这些过滤器条件与这些信息包参数相匹配时,产生匹配过滤器。和过滤器条件一样,这些信息包参数包括从信息包中被加以分析或获得的信息。当防火墙识别匹配过滤器时,执行与这些过滤器条件关联的一个或多个动作。

[0056] 图 3 表现了本发明的防火墙结构的实施例的总体视图。该结构提供了一种能力——在网络堆栈的所有层处,使信息包经历过滤器。防火墙提供了中央管理性能,这些中央管理性能允许增加和删除过滤器,并用于识别和解决的过滤器冲突。防火墙结构是可扩展的,这体现在:过滤器层可按需要来被增加和删除,并可以扩大,以包括允许和阻断动作以外的专用功能性。虽然具体参照防火墙和防火墙过滤器来描述本发明,但是,也使用本发明来促进和管理其他的过滤器和策略。作为特殊的例子,本发明适合用来促进和管理被用于“服务质量”(QoS)、“网际协议安全性”(IPSec)套组以及其他的加密协议、鉴定协议和密

钥管理协议的过滤器。

[0057] 防火墙结构包括用户模式处理 250 和核心模式处理 252。用户模式处理 250 和核心模式处理 252 作为网络设备中的操作系统的一部分来加以执行。精通该技术领域的人将会理解：操作系统的用户模式处理 250 和核心模式处理 252 包括额外的部件，为简单起见，没有示出这些额外的部件。作为选择，在操作系统以外作为一个或多个程序模块或应用程序或在单一操作系统处理以内整体或部分地执行防火墙结构。

[0058] 核心模式处理 252 包括网络堆栈 254、在这里被称作“核心防火墙引擎 256”的第一个防火墙引擎，以及任选的呼出 258。总的来说，核心模式处理 252 通过为网络信息包识别匹配过滤器、根据已知的协议来处理信息包并对如匹配过滤器所指定的信息包执行其他动作，来实施被建立的防火墙策略。

[0059] 网络堆栈 254 包括多个层处理，它们在这里被称作“层”，包括数据流层 268、传输层 270、网络层 272 和链路层 274。本发明的防火墙结构是可扩展的，并且，可按需要来动态地增加和删除额外的层。增加的层的例子包括文件存取层 276，该文件存取层根据“服务器主块”(SMB) 协议来加以执行。这些层可以与其他程序模块（例如，“超文本传输协议”(HTTP) 分析程序模块 278) 协同运作。

[0060] 网络堆栈 254 中的这些层处理入站网络信息包和出站网络信息包。出站网络信息包是正从执行防火墙结构的网络设备那里被传送到网络上的信息包。入站信息包是在执行防火墙结构的网络设备处被接收到的信息包。如图 3 中所示的相互箭头所指出的，入站信息包从下到上地穿过网络堆栈 254，出站信息包从上到下地穿过网络堆栈 254。

[0061] 网络信息包序贯地穿过这些网络层，并由这些网络层序贯地加以处理。根据已知的技术，网络堆栈 254 中的每个层能够从前一个层或处理那里接收信息包，能够根据规范或协议来处理该信息包，并能够将这个处理过的信息包发送到下一个层或处理。根据本发明，网络堆栈 254 中的每个层也保持信息包上下文，将该信息包上下文传递到下一个层，将分类请求发给核心防火墙引擎 256，并根据防火墙策略来对该信息包采取行动。

[0062] 信息包上下文是从一个层到另一个层地跟随信息包的数据结构。每个层通过将该楼层被设计成要加以处理的一组参数（例如，该层被设计成从入站信息包中加以分析的、加入出站信息包的或从信息包的内容中导出的信息）加入该上下文数据结构，来保持该上下文。参照图 5 来描述被用于信息包上下文的示范数据结构。

[0063] 由网络堆栈 254 的各个层执行的这些操作中的一项操作是：通过发出分类请求，来调用核心防火墙引擎。该分类请求是网络堆栈 254 中的层所执行的调用，请求：识别与信息包相匹配的任何过滤器，并将任何关联的动作（例如，允许或阻断）返回到该层。发出该分类请求的这个层在这里被称作“请求层”。每个层也对由核心防火墙引擎 256 返回的信息包采取行动。

[0064] 核心防火墙引擎 256 包括层 API 280、一组被安装的过滤器 282 和呼出 API 284。核心防火墙引擎 256 执行各种功能，包括：(1) 保持定义防火墙策略的被安装的过滤器组 282；(2) 从网络堆栈 254 中的这些层接收分类请求；(3) 根据该分类请求，来识别一个或多个匹配过滤器；以及 (4) 根据这些匹配过滤器，来向该请求层指示要对信息包采取的任何动作。

[0065] 这个被安装的过滤器组中的每个过滤器包括一组过滤器条件以及一个或多个关联的动作。如参照图 2 的描述，这些过滤器条件识别经历这个关联的过滤器动作的这些网

络信息包。被安装的过滤器组 282 中所规定的这些动作包括允许和阻断。经由任选的呼出 258 来增加额外的功能性。参照图 4 来描述这些过滤器的示范形式。

[0066] 层 API 280 在网络堆栈 254 中的各个层与核心防火墙引擎 256 之间提供接口。通过层 API 280, 请求层将分类请求发给核心防火墙引擎 256。分类请求包括如请求层所接收的信息包、如请求层所接收的信息包上下文, 以及层参数。这些层参数是由请求层处理 (例如, 增加或分析) 的信息包参数。作为特殊的例子, 源网际协议 (IP) 地址和目的网际协议 (IP) 地址是当执行该 IP 协议时由网络层 272 发送的层参数。层参数也可以包括被加入信息包或从信息包中被加以分析的这些信息包参数以外的信息。作为特殊的例子, 这些层参数包括局部地址类型。该局部地址类型由 IP 层来确定, 并作为该分类请求的一部分来加以发送。局部地址类型包括单播、广播、多播、任意点传送 (anycast) 和类似的类型。参照图 6 来描述层 API 280 的特殊实施。

[0067] 随意地, 可使用呼出 258 来执行允许和阻断过滤器动作以外的增加的功能性。当核心防火墙引擎 256 为该信息包识别匹配过滤器 (包括作为关联的动作的对这些呼出模块之一的呼出) 时, 执行呼出。核心防火墙引擎将如请求层所发出的分类请求 (即完全信息包、层参数和信息包上下文) 连同匹配过滤器的标识一起经由呼出 API 284 发送到呼出模块。在本发明的实施例中, 防火墙结构包括呼出基本组 258。和层一样, 按需要将额外的呼出加入可扩展的防火墙结构。参照图 6 来描述呼出 API 284 的特殊实施。

[0068] 用户模式处理 250 包括被称作“用户防火墙引擎 260”的第二个防火墙引擎, 以及被识别为“PP1”、“PP2”和“PP3”的一个或多个策略供应器 262。策略供应器 262 是将防火墙策略 (即被安装的过滤器 282) 加入防火墙结构的处理。可以使用任何处理来完成这项任务。一个例子是传统 (legacy) IPSec 策略服务 (LIPS)。传统 IPSec 策略服务增加对应该使用 IPSec 协议 (例如, “封装安全协议” (ESP) 和 “鉴定头部协议” (AH)) 的网络信息流量进行定义的过滤器。作为特殊的例子, 传统 IPSec 策略增加防火墙策略, 该防火墙策略指出: 必须根据 ESP 协议, 来为所有未经请求的明文的入站信息包加密。该策略进一步规定: 应该阻断任何未经请求的入站信息包 (即未加密的信息包)。策略供应器 262 从任何合适的来源 (例如, 易失或非易失存储器中的数据, 或允许网络管理员或系统用户直接输入策略的 “图形用户界面” (GUI)) 获得该策略。用户防火墙引擎 260 将该策略转换成新的过滤器, 即, 按照过滤器条件来定义该策略, 并将这个新的过滤器加入被安装的过滤器组 282。

[0069] 用户防火墙引擎 260 也执行过滤器判优和冲突解决功能。当策略供应器 262 为用户模式防火墙引擎 260 提供新策略时, 用户防火墙引擎确定: 起因于这项新策略的这个新的过滤器是否与被安装的过滤器 282 中的任何过滤器有冲突。标题为《用于管理基于网络过滤器的策略的方法》的美国专利申请 (代理人标签号: 221037) 中描述了一种适合在本发明的构架中使用的、用于识别和解决冲突的方法的例子。

[0070] 该结构进一步包括过滤器引擎 API 266, 过滤器引擎 API 266 构成用户模式防火墙引擎 260 与核心防火墙引擎 256 之间的接口。过滤器引擎 API 266 提供用于用户防火墙引擎 260 的机制, 以将新的过滤器加入被安装的过滤器组 282 或从被安装的过滤器组 282 中除去现存的过滤器, 并用于检查被安装的过滤器 282, 以便可以检测 and 解决过滤器冲突。可由策略供应器 262 获得的管理 API 290 也提供一种机制, 以增加过滤器并从该结构中除去过滤器。

[0071] 用户模式防火墙引擎 260 也包括过滤器模块 294。过滤器模块 294 是用户模式 250 中的核心防火墙引擎 256 的实例。用户模式防火墙引擎 260 中的过滤器模块 294 的这个实例允许用户防火墙引擎 260 为在这里被称作“用户模式层 282”的一个或多个用户模式层处理复制核心防火墙引擎 256 的各项服务。用与创建核心模式层相同的方法来增加用户模式层 282。由于过滤器模块 294 是核心防火墙引擎 256 的用户模式实例,因此,将会理解:这里为核心防火墙引擎 256 而描述的任何功能性也可应用于过滤器模块 294。

[0072] 密码模块 API 288 在用户策略引擎 260 与密码模块 296 之间提供接口。使用密码模块 API 288 来用信号通知该密码模块:需要建立 SA。

[0073] 该防火墙结构包括引导时间策略 286 和故障保险策略 264。执行引导时间策略 286 和故障保险策略 264,以提供被安装的过滤器基本组 282,同时,该系统处于过渡状态,例如当该网络设备被初始化(例如,被开启、重置或被重新启动)时。在此过渡状态期间,存在潜伏期,在此潜伏期间,不建立用户模式 250,并且,不可能经由策略供应器 262 和用户防火墙引擎 260 来安装过滤器。结果,该网络设备容易受到来自(例如)未经请求的入站信息包的恶意攻击。引导时间策略 286 驻留在核心模式 252 中,并在建立用户模式 250 时间之前被载入核心防火墙引擎 256。引导时间策略 296 提供可配置的被安装的基本过滤器 282,以提供任何合适的过滤器条件和关联的动作,用于在此过渡状态期间保护系统。引导时间策略 286 的例子是:阻断所有入站网络信息包和出站网络信息包。

[0074] 一旦建立用户模式 250,该过渡状态就继续保持,同时,策略供应器 262 正在识别策略并尝试将被安装的过滤器 282 加入核心防火墙引擎 256。当建立用户模式 250 时,用户防火墙引擎 260 在策略供应器 262 接受任何策略之前,首先将故障保险策略 264 载入核心防火墙引擎 256。然后,禁止引导时间策略 286。和引导时间策略 286 一样,故障保险策略 264 是基础过滤器,它被设计成:当这些策略供应器和用户防火墙引擎仍然处于安装过滤器的过程中时,用于在过渡状态期间保护该系统避免遭遇攻击。一旦策略供应器 262 将其各自的过滤器成功地安装到网络设备中,就禁止故障保险策略 264。

[0075] 参考图 4,现在将描述被安装的过滤器组 282。被安装的过滤器组 282 中的每个过滤器 310 具有多个字段,这些字段包括过滤器 Id 312、加权 314、一个或多个动作 316、策略上下文 328 和一组过滤器条件 318。过滤器 Id 312 为过滤器提供唯一标识。例如,过滤器 Id 312 被用作供核心防火墙引擎 256 将匹配过滤器信息返回到用户防火墙引擎 260 和呼出 258 的一种工具。在本发明的实施例中,过滤器被分配给网络堆栈 254 中的各个层中的一个层。过滤器 Id312 被核心防火墙引擎 256 用来跟踪将哪个过滤器分配给哪个层。

[0076] 加权字段 314 包括识别过滤器 310 的优先级的值。加权字段 314 中的这个值越大,该过滤器的优先级就越高。过滤器优先级确定核心防火墙引擎 256 将匹配过滤器应用于该信息包的顺序。

[0077] 在本发明的实施例中,首先应用具有最高优先级(即最大的加权值)的那个过滤器,然后应用优先级位于其次的过滤器,等等,直到遇到具有终止动作的匹配过滤器为止。以下更加详细地描述了终止动作。一旦应用具有终止动作的匹配过滤器,核心防火墙引擎 256 就停止应用匹配过滤器。这样,在应用该终止动作之后,不对信息包采取由较低优先级匹配过滤器规定的动作 316。作为选择,防火墙引擎 256 识别单一匹配过滤器,并返回来自该单一匹配过滤器的一组动作。

[0078] 过滤器条件集 318 确定信息包是否与过滤器 310 相匹配。每个过滤器条件 318 包括类型 320、数据 322 和层 Id : 字段 Id 324。类型 320 定义对应的数据 322 中所包括的变量的长度和数目。该结构规定预定义的已知变量类型（例如，“字节”、“短”、“长”、“8 个字节”、“字符串”、“网际协议版本 4 (IPv4) 地址”、“网际协议版本 6 (IPv6) 地址”、“IPv4 地址加上掩码”、“IPv6 地址加上掩码”和“地址范围”）。

[0079] 数据字段 322 包括与类型相匹配的数据。例如，如果类型是“IPv4 地址”，则关于数据字段 322 的可接受的值是如有点的十进制记数法中所表达的 00.00.00.00 ~ 255.255.255.255 的范围内的 32 位数字。在一些实例中，类型 320 规定数据字段 322 中的多个值。“地址范围”、“IPv4 地址加上掩码”和“IPv6 地址加上掩码”这些类型允许采用两个 IP 地址值，从而定义 IP 地址的始、末范围。为了最大的灵活性，该结构也允许采用用户定义的类型。作为选择，将额外的类型手动加入该系统结构。

[0080] 使用层 Id : 字段 Id 324 来分别识别起源层和来自该起源层的参数。该起源层和来自该起源层的这个参数定义信息包参数，即当识别匹配过滤器时与数据 322 进行比较的层参数和信息包上下文。起源层识别网络堆栈中的一个层。来自起源层的参数识别与该起源层关联的特殊参数。过滤器条件 326 展示了特殊的例子。类型是 IPv4，从而指出：数据 322 是 32 位 IP 地址。层 Id 是“IP”，表示：32 位数字是 IP（即网络）层参数。字段 Id 是“Src IP Addr”，它在该例中代表源 IP 地址。数据字段 322 中所提供的 IP 地址是“123.3.2.1”，指出：具有那个源 IP 地址的任何信息包都符合该过滤器条件，从而与该过滤器相匹配。过滤器 310 可以包括多个过滤器条件 318，在这种情况下，只有当符合所有的过滤器条件时，信息包才与过滤器 310 相匹配。

[0081] 过滤器 310 中的动作 316 是允许、阻断或呼出。如果过滤器 310 中的动作 324 是允许或阻断，并且该信息包与过滤器 310 相匹配，那么，该允许或阻断动作被核心防火墙引擎 256 返回到请求层。如果动作 316 是呼出，则核心防火墙引擎 256 将其自己的分类请求发给指定的呼出模块 258，分类请求包括该完全信息包、层参数、上下文和匹配过滤器的标识。呼出模块 258 对该信息包执行其被编程的功能（例如，侵入检测）。该呼出可以将动作（允许或阻断）返回到该核心防火墙引擎，这又将该动作转达给该请求层。该呼出也能够保持信息包上下文，该信息包上下文同样经由核心防火墙引擎 256 而被返回到请求层。该动作也可能是值（例如空值，从而指出：不存在防火墙策略）。

[0082] 动作被指定为终止或非终止。如果该动作是终止，那么，一旦为信息包识别那个动作，该动作就被返回到该请求层，并且没有额外的过滤器被应用于该信息包。如果该动作是非终止，则额外的过滤器查找继续进行，直到识别关于该信息包的终止动作为止。默认的情况是，“允许”和“阻断”被指定为终止动作。

[0083] 使用策略上下文 328 来存储除防火墙策略以外的策略（例如，安全性策略或 QoS 策略）。该策略上下文是任何合适的数据结构。例如，该策略上下文是由加入过该策略上下文的处理来解释的 64 位数字。该策略上下文和 / 或动作可能是空值。

[0084] 图 5 展示了被用于信息包上下文的数据结构 330 的例子，信息包上下文由网络堆栈 254 和呼出模块 258 中的各个层来保持，并被传递到这些层。当这个入站或出站网络信息包穿过这些层并包括被标注为 336-340 的一个或多个项目时，信息包上下文 330 跟随网络信息包。每个项目包括层 Id : 字段 Id 332 和对应的值 334。

[0085] 层 Id : 字段 Id 332 的含义等同于作为过滤器 310 中的过滤器条件 318 的一部分而加以提供的层 Id : 字段 Id 324 的含义 (图 4)。也就是说, 层 Id : 字段 Id 322 为值字段 334 中的数据识别起源层和来自起源层的层参数。值字段 334 包括特殊的层参数。

[0086] 作为特殊的例子, 项目 336 包括层 Id : 字段 Id 332 “NDIS : Src. MAC Addr. ”。“NDIS”表示链路层 274 的“网络驱动器接口规范”实施 (图 1)。“Src MAC addr.”表示源 MAC 地址。这样, 层 : 字段 Id 332 指出 : 值字段 334 中的数据是曾由 NDIS (链路) 层处理的源 MAC 地址。值字段 334 包括这个实际的源 MAC 地址, 在该例中, 该源 MAC 地址是如十六进制记数法中所表达的“00. 08. 74. 4F. 22. E5”。

[0087] 作为第二个例子, 项目 338 具有“NDIS : IF No”的层 Id : 字段 Id 332。这再次将该层识别为 NDIS, 但在此情况下, 将该参数识别为把接口号表示为该特殊 NDIS 参数的“IF No”。值字段 334 包括这个实际的接口号, 该接口号在此情况下是 2。

[0088] 作为第三个例子, 项目 340 具有“IP : Dst IP Addr”的层 Id : 字段 Id 332。“IP”表示使用 IP 协议的网络层, 并且, “Dst IP Addr”将目的 IP 地址表示为 IP 层参数。值字段 334 包括“123. 3. 2. 1”的这个实际的目的 IP 地址。

[0089] 已描述了本发明的基础防火墙结构, 现在来注意使用这里所描述的该基础防火墙结构来加以执行的系统和示范方法的这些功能接口。这些功能接口作为多个应用编程接口 (APIs) 来加以执行。这些 APIs 包括如图 6 和图 7 所示的层 API 280、呼出 API 284、过滤器引擎 API 266 和密码模块 API 288。

[0090] 层 API 280 促进网络堆栈 254 内的各个层中的每个层与核心防火墙引擎 256 之间的数据交换。如所示, 层 API 280 包括“分类”方法 402、“增加层”方法 404 和“删除层”方法 406。层 API 280 也可以在过滤器模块 294 内执行, 以促进这些用户模式层与过滤器模块 294 之间的通信。

[0091] 请求层使用“分类”方法 402, 将层参数、如请求层所接收的信息包以及信息包上下文发送到核心防火墙引擎 256。核心防火墙引擎 256 将 (1) 来自请求层的这些层参数和 (2) 信息包上下文项目跟被分配给请求层的每个过滤器 310 中的过滤器条件 318 进行比较, 以识别匹配过滤器。以下是该“分类”方法的示范实施。将会理解 : 以下这些方法被描述成接收或返回数据值。根据已知的编程技术, 这些方法可以使用指向数据值的指针, 而不是实际的数据值。

[0092] NTSTATUS

[0093] WFPClassify

[0094] (

[0095]	IN ULONG	LayerId,
[0096]	IN WFP_INCOMING_VALUES*	pInFixedValues,
[0097]	IN WFP_INCOMING_CONTEXT_VALUE*	pInContext,
[0098]	PVOID	pPacket,
[0099]	OUT WFP_ACTION_TYPE*	pActionType,
[0100]	OUT UINT64*	pOutContext

[0101]);

[0102] 其中, 以下内容表现了这些所列举的参数的特征。

[0103] LayerId 识别发出分类请求的网络层,即请求层。参考图 3,该“层”Id 将该层识别为数据流层 268、传输层 270、网络层 272 或链路层 274。如果被加入该系统,则其他层有效。例如,如果加入 SMB 层 276,则 SMB 层具有其自己的唯一标识。本发明的防火墙结构进一步允许在网络堆栈 254 中的层处的多项协议实施。例如,该堆栈具有两个传输层 270——第一传输层使用 TCP 协议,第二传输层使用 UDP 协议。也可以加入用户模式层 282,从而形成有效层。

[0104] pInFixedValues 包括由请求层处理的这些层参数的子集。将该 pInFixedValues 连同这些信息包上下文项目与这些过滤器条件进行比较,以确定信息包是否与过滤器相匹配。以下的表格 A 中识别用于每个层的 pInFixedValues 中所包括的默认层参数的例子。将会理解:以下这些值只是起示范的作用,因为该结构的优点是:它允许这些层使用那个层可以获得的任何值。

[0105] 表格 A

[0106]

层	默认层参数
链路层	源 MAC 地址和目的 MAC 地址 ; 接口号
网络层	源 IP 地址和目的 IP 地址 ; 协议类型 ;局部地址类型
传输层	源端口号和目的端口号 ;
应用程序	被解密的应用层协议有效载荷

[0107] pInContext 包括如请求层所接收的上下文数据结构 330 (图 5)。核心防火墙引擎 256 使用信息包上下文并结合这些层参数来识别匹配信息包。

[0108] pPacket 包括如请求层所接收的整个信息包。pPacket 没有被核心防火墙引擎 256 用来识别匹配过滤器。如前所述,核心防火墙引擎 256 使用 pInFixedValues 和 pInContext 来识别匹配过滤器。pPacket 被包括在“分类”方法中,以便核心防火墙引擎 256 可以将它发送到被识别为匹配过滤器中的动作 316 的一个或多个呼出模块 258。

[0109] pActionType 包括被返回到请求层的动作 316。所返回的动作 316 是如在匹配过滤器中所识别的允许或阻断、或由匹配过滤器执行的呼出模块。

[0110] pOutContext 包括策略上下文数据。如前所述,使用该策略上下文来推动跟 IPSec、QOS 和任何其他基于非防火墙过滤器的策略有关联的网络策略。

[0111] “增加层”404 和“删除层”406 方法被分别用来增加层和从该防火墙结构中除去层。以下是“增加层”404 方法的示范形式。

[0112] NTSTATUS

[0113] AddExtensionLayer(OUT PULONG pLayerId) ;

[0114] 其中,以下内容表现了这些所列举的参数的特征。

[0115] pLayerId 是被返回到正在添加的那个层 (即执行“增加层”方法的那个层) 的唯

一层标识值。

[0116] 以下是“删除层”406 方法的示范形式。

[0117] NTSTATUS

[0118] RemoveExtensionLayer (ULONG LayerId) ;

[0119] 其中, 以下内容表现了这些所列举的参数的特征。

[0120] LayerId 识别正在被除去的那个层, 即执行“删除层”方法的那个层。

[0121] 呼出 API 284 促进核心防火墙引擎 256 与呼出 258 之间的数据交换。“呼出”API 284 包括“分类”方法 410、“通知”方法 412、“登记呼出”方法 414 和“撤消登记呼出”方法 416。“呼出”API 284 的“分类”方法 410 类似于“层”API 280 的“分类”方法 402, 除“它还包括匹配过滤器数据”这一点以外。以下是被用来执行呼出的“分类”法 410 的示范形式。

[0122] typedef NTSTATUS (*WFP_CALLOUT_CLASSIFY_FN)

[0123] (

[0124] IN const WFP_INCOMING_VALUES* fixedValues,

[0125] IN WFP_INCOMING_CONTEXT_VALUE*w fpContext,

[0126] IN VOID* packet,

[0127] IN WFP_FILTER* matchedFilter,

[0128] OUT WFP_ACTION_TYPE* action,

[0129] OUT UINT64* outContext

[0130]);

[0131] 其中, 以下内容表现了这些所列举的参数的特征。

[0132] fixedValues 包括从请求层那里被发送的这些层参数。fixedValues 是 pInFixedValues 数据中的请求层所提供的相同的数据, pInFixedValues 数据作为层 API 280 中的“分类”方法 402 的一部分来加以发送。

[0133] wfpContext 包括上下文数据结构 330 (图 5)。该数据等同于如 pInContext 中的请求层所发送的数据, pInContext 作为层 API 280 中的“分类”方法 402 的一部分来加以发送。

[0134] packet 包括如请求层所接收的整个信息包。该数据等同于如 pPackett 中的请求层所发送的数据, pPacket 作为层 API 280 中的“分类”方法 402 的一部分来加以发送。

[0135] matchedFilter 识别请求呼出的过滤器。通常, 该匹配过滤器由启动呼出 API 284 的“分类”方法 410 的匹配过滤器 310 的过滤器 Id 312 来加以识别。

[0136] pActionType 包括从呼出 258 被返回到核心防火墙引擎 256 的那个动作。如果 pActionType 是允许或阻断, 则将它返回到请求层, 作为由“层”API 280 返回的 pActionType。该呼出也可以返回继续动作, 该继续动作指示核心防火墙引擎 256 继续将匹配过滤器应用于信息包。

[0137] pOutContext 包括策略上下文数据 (例如, 安全性或 QOS 策略数据)。

[0138] 当过滤器 310 被加入被安装的过滤器组 282 (将呼出模块 258 识别为其动作 316 之一) 或被删除时, 使用“通知”方法 412 来通知呼出。该通知为该呼出提供了一种机会——采取任何所要求的动作 (例如, 当核心防火墙引擎 256 执行该动作时, 分配或解除分配将由呼出 258 使用的缓冲器)。以下是“通知”方法的示范形式。

```
[0139] typedef NTSTATUS (*WFP_CALLOUT_NOTIFY_FN)
[0140] (
[0141]     IN WFP_NOTIFY_ENUM notify,
[0142]     IN WFP_FILTER* filter
[0143] );
```

[0144] 其中,以下内容表现了这些所列举的参数的特征。

[0145] notify 包括一个数值,该数值指出是正在增加还是正在删除过滤器。例如,1 的值指出正在增加过滤器,2 的值指出正在删除过滤器。

[0146] filter 识别正在被唯一值增加或删除的过滤器。通过提供作为过滤器 310 的一部分而被包括在内的过滤器 Id 312,可以实现这一点。

[0147] 分别使用呼出“登记”方法 414 和呼出“解除登记”416 方法来增加和除去呼出模块。登记方法 414 的示范形式如下所述:

```
[0148] NTSTATUS WfpRegisterCallout
[0149] (
[0150]     IN const GUID* calloutId,
[0151]     IN const WFP_CALLOUT* callout,
[0152]     IN const SECURITY_DESCRIPTOR* sd
[0153] );
```

[0154] 其中,以下内容表现了这些所列举的参数的特征。

[0155] callout Id 提供关于登记呼出模块的唯一标识。

[0156] callout 提供任何呼出特定信息(例如,驱动器服务名称、设备名称,以及指向这些呼出分类和通知函数的指针)。

[0157] sd 提供关于呼出的安全描述符。该安全描述符提供存取控制清单(ACL),该存取控制清单规定哪些处理可以修改和删除呼出。

[0158] 呼出解除登记方法 416 的示范形式如下所述:

```
[0159] NTSTATUS WfpDeregisterCallout
[0160] (
[0161]     IN const GUID* calloutId
[0162] );
```

[0163] 其中,以下内容表现了这些所列举的参数的特征。

[0164] callout Id 是将被除去的呼出的唯一 Id。

[0165] 过滤器引擎 API 266 促进用户模式防火墙引擎 260 与核心模式防火墙引擎 256 之间的数据交换。如所示,管理 API 266 包括“增加过滤器”方法 418、“删除过滤器”方法 420 和“列举层”方法 422。也提供了过滤器引擎 API 266 的这些方法,作为管理 API 290 的一部分。

[0166] 分别使用“增加过滤器”418 和“删除过滤器”418 方法,来将新的过滤器加入被安装的过滤器组 282,并从被安装的过滤器组 282 中删除现存的过滤器。以下是“增加过滤器”法 418 的示范形式。

```
[0167] NTSTATUS
```

```
[0168]   AddFilterToLayer
[0169]   (
[0170]       ULONG           LayerId,
[0171]       WFP_FILTER*      pFilter
[0172]   );
```

[0173] 其中,以下内容表现了这些所列举的参数的特征。

[0174] LayerId 识别为其分配过滤器的那个层。

[0175] pFilter 是正在被加入被安装的过滤器组 282 的过滤器 310。

[0176] 以下是“删除过滤器”法 420 的示范形式。

```
[0177]   NTSTATUS
[0178]   DeleteFilterFromLayer
[0179]   (
[0180]       ULONG           LayerId,
[0181]       ULONG           FilterId
[0182]   );
```

[0183] 其中,以下内容表现了这些所列举的参数的特征。

[0184] LayerId 识别为其分配过滤器的那个层。

[0185] pFilter 是正从这个被安装的过滤器组中被删除的过滤器。

[0186] “列举层”方法 422 为用户防火墙引擎 260 提供一种机制,以识别与一套标准相匹配的所有过滤器。这样,可允许过滤器引擎 API 识别发生冲突的过滤器,用于过滤器判优和冲突解决。以下是“列举层”方法 422 的示范形式。

```
[0187]   LayerStartEnum
[0188]   (
[0189]       PWFP_ENUM_TEMPLATE      pEnumTemplate,
[0190]       OUT PULONG              pMatchCount,
[0191]       OUT PWFP_ENUM_HANDLE    pEnumHandle
[0192]   )
```

[0193] 其中,以下内容表现了这些所列举的参数的特征。

[0194] pEnumTemplate 包括定义应该被返回的这些过滤器的一种数据结构。例如,它包括这些过滤器条件必须与要被返回的过滤器匹配的参数。

[0195] pMatchCount 包括基于这个规定的 pEnumTemplate 的过滤器匹配数目。

[0196] pEnumHandle 包括对这些匹配的过滤器项目的引用。

[0197] 密码模块 API 288 在密码模块与用户防火墙引擎 260 之间提供接口。密码模块 API 288 包括“IPSec SA 获取”方法 430、“期满通知”方法 432、“IPSecSA 获取完成”方法 434、“密码模块登记”方法 436、“密码模块解除登记”方法 438、“IPSec 入站获得 SPI”方法 440、“增加入站 SA”方法 444、“增加出站 SA”方法 446、“入站 SA 期满”448 方法和“密码模块启动”方法 350。密码模块 API 288 也可以包括先前为层 API 而描述的各种方法(例如,“分类”方法 402、“增加层”方法 404 和“删除层”方法 406)。

[0198] 使用密码模块 API 来促进已知安全协议(例如,由 IPSec 定义,由启动计算机和响

应计算机使用)的运用。IPSec 包括诸如 AH 和 ESP 等协议。ESP 协议是鉴定和加密协议,该协议使用密码机制来提供完整性、来源鉴定和数据机密性。AH 协议是鉴定协议,该协议使用信息包头部中的散列签名来证实信息包数据的完整性和发送者的可靠性。

[0199] IKE 协议提供一种关于启动计算机和响应计算机的方法,以商议用于 AH 协议和 ESP 协议的安全设置。所商议的这些安全设置构成被称作“安全关联”(SA)的数据结构。SA 定义被 ESP 或 AH 用来保护 IP 信息包的内容的参数(例如,鉴定算法、加密算法、密钥和密钥的使用期限)。由于 ESP 和 AH 需要被建立的 SA,因此,在启动和响应计算机使用 ESP 或 AH 协议之前,执行 IKE 商议。由被称作“安全参数索引”(SPI)的值来识别给定的 SA。

[0200] 启动计算机和响应计算机中的每种计算机包括 IPSec 驱动器,该 IPSec 驱动器根据 IPSec 策略来确定在该启动计算机与响应计算机之间被发送的数据是要求加密还是要求鉴定。该 IPSec 策略是一组过滤器,它定义该网络设备如何使用 IPSec,并包括过滤器清单、鉴定方法和其他信息。在本发明的实施例中,由核心防火墙引擎或过滤器模块 294 内的被安装的过滤器组中所包括的过滤器来定义 IPSec 策略。通过调用使用“分类”方法来识别匹配过滤器的密码模块层,来识别应用于信息包的策略。

[0201] 用户防火墙引擎 260(经由客户代理)调用“IPSec SA 获取”方法 430,来将驱动器获取或外部启动请求传递到作为密码模块层来加以执行的用户模式层 282。密码模块层返回这个调用,并异步地执行商议。一旦密码模块层已完成商议,密码模块层就调用“IPSec SA 获取完成”方法 343,以通知用户策略引擎 260:商议完成。以下是“IPSec SA 获取”方法的示范形式。

[0202] typedef WIN32_ERR(*PROCESS_IPSEC_SA_ACQUIRE0)

[0203] (

[0204] IN FWP_IPSEC_ACQUIRE_CONTEXT0 ipsecContext,

[0205] IN const FWP_IPSEC_SA_ACQUIRE0* acquire,

[0206] IN FWP_IPSEC_SPI inboundSAspi

[0207]);

[0208] 其中,以下内容表现了这些所列举的参数的特征。

[0209] ipsecContext 是将获取与正在被加入的 SA 连接起来的句柄。

[0210] acquire 包括用于根据已知协议(例如, IKE)来商议 SA 的必要信息。

[0211] inboundSAspi 包括被用于入站 SA 的 SPI。

[0212] 调用“期满通知”方法 432,以便将期满通知传递到增加过入站 SA 的那个密码模块层。以下是“期满通知”方法的示范形式。

[0213] typedef VOID(*PROCESS_IPSEC_SA_EXPIRE0)

[0214] (

[0215] IN const FWP_IPSEC_SA_EXPIRE_NOTIFY0*expireNotify

[0216]);

[0217] 其中,以下内容表现了这些所列举的参数的特征。

[0218] expireNotify 包含识别期满 SA 的信息。例如,在出站 SA 的情况下,提供 SPI。

[0219] 密码模块层调用“IPSec SA 获取完成”434 方法,以便在它已结束商议并增加所有这些 SA 之后,或在它已碰到错误之后,关闭用户防火墙引擎的上下文。在执行该方法之后,

密码模块层没有再使用这个 `ipsecContext` 用于其他任何 API 方法。以下是“IPSec SA 获取完成”方法的示范形式。

```
[0220] WIN32_ERR
[0221] FwpIPSecSAAcquireComplete0
[0222] (
[0223]     IN FWPM_ENGINE_HANDLE engineHandle,
[0224]     IN FWP_IPSEC_ACQUIRE_CONTEXT0 ipsecContext,
[0225]     IN const FWP_IPSEC_NEGOTIATION_STATUS0* status
[0226] );
```

[0227] 其中,以下内容表现了这些所列举的参数的特征。

[0228] `engineHandle` 提供用户防火墙引擎 260 的句柄。

[0229] `ipsecContext` 是由利用“IPSec 获取”方法的用户防火墙引擎传递的上下文。

[0230] `status` 提供 SA 商议的状态和其他细节。如果经由 `FwpKeyingModuleInitiate0` 在外部启动获取,则用户防火墙引擎 260 返回该状态。

[0231] 由密码模块层来调用“密码模块登记”方法 436,以便向用户防火墙引擎 260 登记,并传递其功能指针。以下是“密码模块登记”方法的示范形式。

```
[0232] WIN32_ERR
[0233] FwpKeyingModuleRegister0
[0234] (
[0235]     IN FWPM_ENGINE_HANDLE engineHandle,
[0236]     IN const GUID* keyingModuleID,
[0237]     IN const FWP_KEYING_MODULE_INFO0* keymodInfo
[0238] );
```

[0239] 其中,以下内容表现了这些所列举的参数的特征。

[0240] `engineHandle` 提供用户防火墙引擎 260 的句柄。

[0241] `keyingModuleID` 是关于密码模块层的唯一 ID。

[0242] `keymodInfo` 包括关于密码模块层的登记信息(例如,用于处理“IPSec SA 获取”和“IPSec SA 期满”函数的指针)。

[0243] 由密码模块层来调用“密码模块解除登记”方法 438,以便从用户防火墙引擎 260 中解除登记密码模块。以下是“密码模块解除登记”方法的示范形式。

```
[0244] WIN32_ERR
[0245] FwpKeyingModuleDeregister0
[0246] (
[0247]     IN FWPM_ENGINE_HANDLE engineHandle,
[0248]     IN const GUID* keyingModuleID
[0249] );
```

[0250] 其中,以下内容表现了这些所列举的参数的特征。

[0251] `engineHandle` 是用户防火墙引擎 260 的句柄。

[0252] `keyingModuleID` 是密码模块层的唯一 ID。

[0253] 由密码模块层来调用“IPSec 入站获得 SPI”方法 440,以获得关于新的入站 SA 的 SPI。当密码模块层在响应的网络设备中执行时,通常使用“IPSec 入站获得 SPI”方法 440。以下是该“IPSec 入站获得 SPI”方法的示范形式。

```
[0254] WIN32_ERR
[0255] FwpIPSecSAInboundGetSpi0
[0256] (
[0257]     IN FWPM_ENGINE_HANDLE          engineHandle,
[0258]     IN const FWP_IPSEC_TRAFFIC0*    ipsecTrafficDescription,
[0259]     IN const FWP_IPSEC_UDP_ENCAP0*  udpEncapInfo,
[0260]     OUT FWP_IPSEC_SPI*              inboundSpi
[0261] );
```

[0262] 其中,以下内容表现了这些所列举的参数的特征。

[0263] engineHandle 是用户防火墙引擎 260 的句柄。

[0264] ipsecTrafficDescription 是用于创建入站幼体 (larval)SA 的 5 元组说明。该 5 元组包括源 IP 地址与目的 IP 地址、源端口与目的端口,以及传输层协议类型。

[0265] udpEncapInfo 是用于创建幼体 SA 的 UDP 封装数据。UDP 封装是将根据安全协议而加以格式化的信息包嵌入未加密的 UDP 信息包的一种已知的方法。

[0266] inboundSpi 是关于入站 SA 的 SPI。

[0267] 由密码模块层来调用“增加入站 SA”方法 444,以增加入站 SA(即,更新幼体 SA)。用户防火墙引擎 260 使用 SA 中的 SPI 来将此调用映射到其内部状态,并将 SA ioctl(输入输出控制)down 到 IPSec 驱动器中。以下是“增加入站 SA”方法的示范形式。

```
[0268] WIN32_ERR
[0269] FwpIPSecSAInboundAdd0
[0270] (
[0271]     IN FWPM_ENGINE_HANDLE          engineHandle,
[0272]     IN const FWP_IPSEC_SA_STRUCT0 *inboundSA
[0273] );
```

[0274] 其中,以下内容表现了这些所列举的参数的特征。

[0275] engineHandle 是用户防火墙引擎 260 的句柄。

[0276] inboundSA 包括入站 SA。

[0277] 由密码模块层来调用“增加出站 SA”446 方法,以增加出站 SA。用户防火墙引擎使用入站 SPI 参数来将此调用映射到其内部状态,并将该 SA ioctl(输入输出控制)到 IPSec 驱动器中。以下是“增加出站 SA”方法的示范形式。

```
[0278] WIN32_ERR
[0279] FwpIPSecSAOutboundAdd0
[0280] (
[0281]     IN FWPM_ENGINE_HANDLE          engineHandle,
[0282]     IN FWP_IPSEC_SPI              inboundSpi,
[0283]     IN const FWP_IPSEC_SA_STRUCT0* outboundSA
```

[0284]);

[0285] 其中,以下内容表现了这些所列举的参数的特征。

[0286] engineHandle 是用户防火墙引擎 260 的句柄。

[0287] inboundSpi 是关于入站 SA 的 SPI,入站 SA 与出站 SA 配对。

[0288] outboundSA 包括出站 SA。

[0289] 由密码模块层来调用“入站 SA 期满”方法 448,以截止先前增加的入站 SA。以下是“入站 SA 期满”方法 448 的示范形式。

[0290] WIN32_ERR

[0291] FwpIPSecSAInboundExpire0

[0292] (

[0293] IN FWPM_ENGINE_HANDLE engineHandle,

[0294] IN const FWP_IPSEC_SA_EXPIRE0* expire

[0295]);

[0296] 其中,以下内容表现了这些所列举的参数的特征。

[0297] engineHandle 是用户防火墙引擎 260 的该句柄。

[0298] expire 包括关于要期满的 SA 的数据。

[0299] 由已知的外部应用程序(比如 RAS、Winsock API 和类似物)来调用“密码模块启动”方法 450,以便在应用程序开始发送其网络信息流通量之前,启动密码模块层并设置 SA。用户防火墙引擎 260 异步地挂起该 RPC 调用,从 IPsec 驱动器那里获得 SPI,并将该获取传递到合适的密码模块。一旦密码模块层调用 FwpIPSecSAAcquireComplete0,用户防火墙引擎就完成具有商议状态的异步 RPC。以下是“密码模块启动”法的示范形式。

[0300] WIN32_ERR

[0301] FwpKeyingModuleInitiate0

[0302] (

[0303] IN FWPM_ENGINE_HANDLE engineHandle,

[0304] IN const FWP_IPSEC_SA_ACQUIRE0* acquire,

[0305] IN HANDLE waitEvent,

[0306] OUT FWP_IPSEC_NEGOTIATION_STATUS0* negotiationStatus

[0307]);

[0308] 其中,以下内容表现了这些所列举的参数的特征。

[0309] engineHandle 是用户防火墙引擎 260 的句柄。

[0310] acquire 包括商议 SA 所必要的的数据。

[0311] waitEvent 是给当商议状态可用时被触发的事件的句柄。如果客户(即,进行调用的外部应用程序)没有兴趣等待该商议完成,则它可以将这个参数设置为 NULL。在内部,该客户代理随意地将这个事件传递到 RPC,并且,一旦该异步 RPC 调用完成,就请求它设置这个事件。

[0312] negotiationStatus 包括商议的结果。如果 waitEvent 是 NULL,则 negotiationStatus 是 NULL。否则,negotiationStatus 保持有效,直到 waitEvent 被触发为止。

[0313] 图8展示了根据本发明的、由网络堆栈254中的这些层使用的各种方法。图8中所展示的方法也可以由一个或多个用户模式层282跟过滤器模块294和用户防火墙引擎260协力来加以使用。

[0314] 每个层能够执行多项功能,包括:处理网络信息包;将分类请求发给核心防火墙引擎256;以及,管理信息包上下文。在本发明的实施例中,在网络堆栈254中所安装的薄垫片460、462、464、466内的各个层中的每个层处执行这些功能。作为选择,该功能性被直接置入这些单独的层,而不需要这些薄垫片。

[0315] 网络堆栈254包括数据流层268、传输层270、网络层272和链路层274。出于展示本发明的目的,链路层274作为NDIS驱动器来加以执行,网络层272作为IP层来加以执行,传输层270作为TCP层来加以执行,数据流层268作为HTTP层来加以执行。将会理解:可以根据任何协议来执行层。例如,传输层也配合“用户数据报协议”(UDP)。应用层支持“文件传输协议”(FTP)、“远程过程调用”(RPC)、“简单邮件传输协议”(SMTP)、“服务器主块”(SMB)等。如前所述,可以将额外的层加入该结构,并可以删除层。例如,使用参照图6来加以描述的该“增加层”方法和“删除层”方法,来增加和删除层。

[0316] 被标注为468(a)-(d)的网络信息包在它穿过网络堆栈254中的各个层并由这些层来处理时,展示了网络信息包。如果信息包468(a)-(d)是入站信息包,则它从下到上穿过网络堆栈。如果信息包468(a)-(d)是出站信息包,则它从上到下穿过网络堆栈。这类处理众所周知,但出于展示本发明的目的,对其进行简要的描述。

[0317] 假设在网络设备(例如,web浏览器)中执行的应用程序启动关于位于另一个网络设备上的网页的内容的请求,则该应用程序将该请求发给数据流层268。在该例中,数据流264根据HTTP协议来对该请求进行格式化,并将该请求发送到信息包468(a)中的传输层。传输层270接收信息包468(a)。传输层270(执行TCP协议)将该数据放置在一个或多个信息包中,并且,为每个信息包提供TCP头部。该TCP头部包括诸如源端口与目的端口、协议类型(即TCP、序号、标志和检验和)等信息。然后,传输层将被标注为468(b)的信息包发送到该网络层。

[0318] 网络层执行IP协议,并将该数据装入IP头部,IP头部包括源IP地址与目的IP地址、标志、检验和以及其他已知信息。IP头部也指出是否将信息包分段。当IP信息包的尺寸超过关于被用来传送信息包的网络技术的“最大传输单位”(MTU)尺寸时,将信息包分段。例如,以太网技术规定:该MTU是1500个字节。如果IP信息包长度超过MTU,则将它分成两个或多个IP信息包——每个IP信息包具有其自己的IP头部,与MTU相比,它们都具有相等的或较短的长度。

[0319] 在本发明的实施例中,网络层被分成第一层和第二层。这第一层(被称作“片段层”)处理IP信息包片段。例如,在为出站IP信息包分段之前,以及在将入站IP信息包重新装配成单一IP信息包之后,第二层(被称作“完全装配层”)处理完全的IP信息包。在网络层处理和可能的分段之后,信息包468(c)被发送到链路层274。链路层274通过提供具有源MAC地址和目的MAC地址以及其他信息的MAC头部,来进一步对数据实行分组(packetizes)。然后,将该信息包发送到网络接口卡(NIC),在那里,该信息包在物理上被传送到网络上。

[0320] 用相反方式来处理入站信息包。信息包468(d)由NIC来接收,并被发送到链路层

274。如果必要的话,则除去 MAC 头部,并将信息包 468(c) 发送到网络层,在那里,如果需要重新装配这些 IP 信息包片段;并且,分析 IP 头部。然后,网络层将信息包 468(b) 发送到传输层,在那里,除去 TCP 头部;并且,如果在多个 TCP 信息包中发送过数据流,则重新装配数据流。最后,将数据流 468(a) 发送到数据流层 268,在那里,该数据由应用协议(在这种情况下是 HTTP 协议)来进行解密。

[0321] 对于每个出站信息包,网络堆栈的各个层保持信息包上下文 470(a)-(c)。对于每个入站信息包,网络堆栈的各个层保持信息包上下文 472(a)-(c)。当每个信息包穿过这些网络层时,信息包上下文跟随它。信息包上下文也被传递到呼出 258,并可以由这些呼出来加以修改(图 1)。

[0322] 当在每个层处处理这些信息包时,更新信息包上下文。每个层将其层参数加入信息包上下文,从而将该信息提供给随后的各个层或处理。如所示,链路层 274 加入如上下文 472(a) 所展示的源 MAC 地址和目的 MAC 地址以及关于入站信息包的接口号。那个上下文由网络层 272 来接收,网络层 272 加入如上下文 472(b) 所展示的源 IP 地址和目的 IP 地址。传输层 266 接收该上下文,并加入如上下文 472(a) 所展示的这些端口号。

[0323] 对于和出站信息包关联的上下文 470(a)-(c) 而言,会发生类似的处理。数据流层 268 加入诸如来自如上下文 470(a) 所展示的信息包有效载荷的 URL 地址等信息,传输层 270 进一步加入如上下文 470(b) 所展示的源端口号和目的端口号,网络层加入如上下文 470(c) 所展示的源 IP 地址和目的 IP 地址。

[0324] 将会理解:每个层可以将可用的任何上下文信息加入那个层。通常,这包括该层被设计成要加以处理(即,增加信息包或分析信息包或从信息包中导出)的任何信息。

[0325] 当在每个层处接收信息包及其对应的上下文时,通过识别这些层参数并发送被标注为 474 的分类请求,该层可用作请求层。分类请求 474 包括层参数 476、从该先前层接收的信息包上下文 478 和完全信息包 480。被用来发出分类请求的示范方法是参照层 API 280 来加以描述的“分类”方法 402(图 7)。

[0326] 响应于每个分类请求,核心模式防火墙引擎 256 将层参数 476 和信息包上下文 478 跟被分配给请求层的这些过滤器的过滤器条件 318(图 4) 进行比较。核心防火墙引擎 256 将被标注为 482 的响应(具有来自拥有最高加权 314 的匹配过滤器 310 的动作 484) 发送到请求层。核心防火墙引擎 256 也返回策略上下文 482。如果核心防火墙引擎 256 没有识别匹配过滤器,则该核心防火墙引擎通知该请求层:不存在匹配过滤器。核心防火墙引擎 256 继续识别匹配过滤器,直到匹配过滤器指定终止动作(即允许或阻断)为止,或者直到检验被分配给该请求层的所有过滤器(不管首先检验哪个过滤器)为止。作为选择,核心防火墙引擎 256 识别所有匹配,并在单一响应中将这些动作返回到该请求层。

[0327] 如一般所展示的,层参数的识别作为由网络堆栈 254 的各个层执行的标准层处理的一部分。不要求额外的信息包分析,从而将对系统性能的影响减到最小。而且,由于这些层在保持信息包上下文的过程中进行合作,因此,防火墙引擎 256 有可能将过滤器条件与通常无法使用那些信息包参数的各个层处的信息包参数进行比较。例如,网络层 268 从包括源 MAC 地址和目的 MAC 地址的链路层 274 那里接收关于入站信息包的上下文。由于网络层 272 发出具有网络层参数(例如,源 IP 地址和目的 IP 地址以及该信息包上下文)的分类请求,因此,即使这些 MAC 地址通常不可用,核心防火墙引擎 256 也可以在网络层 272 处

的 IP 地址和 MAC 地址上进行过滤。

[0328] 参考图 9, 现在将描述与本发明的防火墙结构一同包括在内的示范的呼出模块组 258。呼出模块 258 包括 HTTP 上下文呼出 500、侵入检测呼出 502、IPSec 呼出 504 和记录呼出 506。呼出 API 284 构成核心防火墙引擎 256 与呼出 258 之间的接口。

[0329] HTTP 上下文呼出 500 保持可接受的或（作为选择）不能接受的 URL 地址的高速缓存 508。HTTP 上下文呼出 508 定期访问与公用网连接的服务器 510, 公用网保持 URL 地址并将它们分类成“可接受的”或“不能接受的”。当核心防火墙引擎 256 执行 HTTP 上下文呼出时, 该呼出检查信息包, 对 URL 地址进行解密（如果必要的话）, 并根据高速缓存 508 中的信息确定 : 它是否是可接受的。然后, 如果 URL 地址是可接受的, 则 HTTP 呼出返回允许, 作为动作 316 ; 如果 URL 地址对于核心模式防火墙引擎 256 而言是不能接受的, 则返回阻断 ; 这又经由层 API 280 将动作 316 返回到请求层。HTTP 上下文呼出在执行双亲控制功能的过程中可发挥作用。

[0330] 侵入检测呼出 502 使用可用的算法和技术来检查信息包, 以识别病毒或可疑信息包的征候。如果检测到可疑信息包, 则返回阻断动作 316。可疑信息包的例子是这样一种信息包——其中, IP 头部和 TCP 头部中的所有标志被设置为值 1。信息包是可疑的, 这是因为它永远不会有效, 并且会指出攻击签名。作为选择, 侵入检测呼出 502 修改信息包上下文, 以便标记信息包的可疑性质, 从而推迟了关于是否阻断这个信息包到网络堆栈中的随后各层的决定。

[0331] IPSec 呼出 504 被设计成确定是否曾将适当的安全协议应用于信息包。IPSec 呼出 504 与 IPSec 处理进行通信, 并根据 IPSec 策略来确定信息包是否曾被期望经历 IPSec 处理。如果是, 则 IPSec 呼出 504 根据信息包上下文来验证 : 信息包实际上是否经历过 IPSec 处理。如果信息包曾被期望根据 IPSec 来加以处理, 但却不是如此（例如, 信息包在明文中）, 则返回阻断动作。如果信息包经历过 IPSec 处理, 则 IPSec 呼出验证 : 应用过适当的“安全关联”（SA）。如果没有应用过该适当的 SA, 则呼出返回阻断动作 316, 否则, 返回允许动作 316。作为选择, IPSec 呼出修改信息包上下文, 从而推迟了关于是否阻断还是允许对随后的层的许可的决定。

[0332] 使用记录呼出 506 来保存关于信息包（例如, 供以后使用的完全信息包）的信息。例如, 这类以后的使用可能是例如关于某个网络信息流通量（该网络信息流通量出乎意料地没有穿过网络, 因为它正受到阻断）或用于跟踪该系统上的恶意攻击的诊断程序。

[0333] 这里所引用的所有这些参考资料被包括于此, 用作整体参考。

[0334] 鉴于可以应用本发明的这些原理的这许多可能的实施例, 应该认识到 : 这里根据这些附图而描述的实施例意在只起说明的作用, 而不应该被视作限制本发明的范围。例如, 精通该技术领域的人将会认识到 : 可以在硬件中执行软件中所示的该说明性实施例的这些元件, 反之亦然 ; 或者, 在不脱离本发明的精神的前提下, 可以在配置和细节方面修改说明性实施例。所以, 如这里所描述的本发明计划预期将所有这类实施例纳入以下的如权利要求书及其相等物的范围以内。

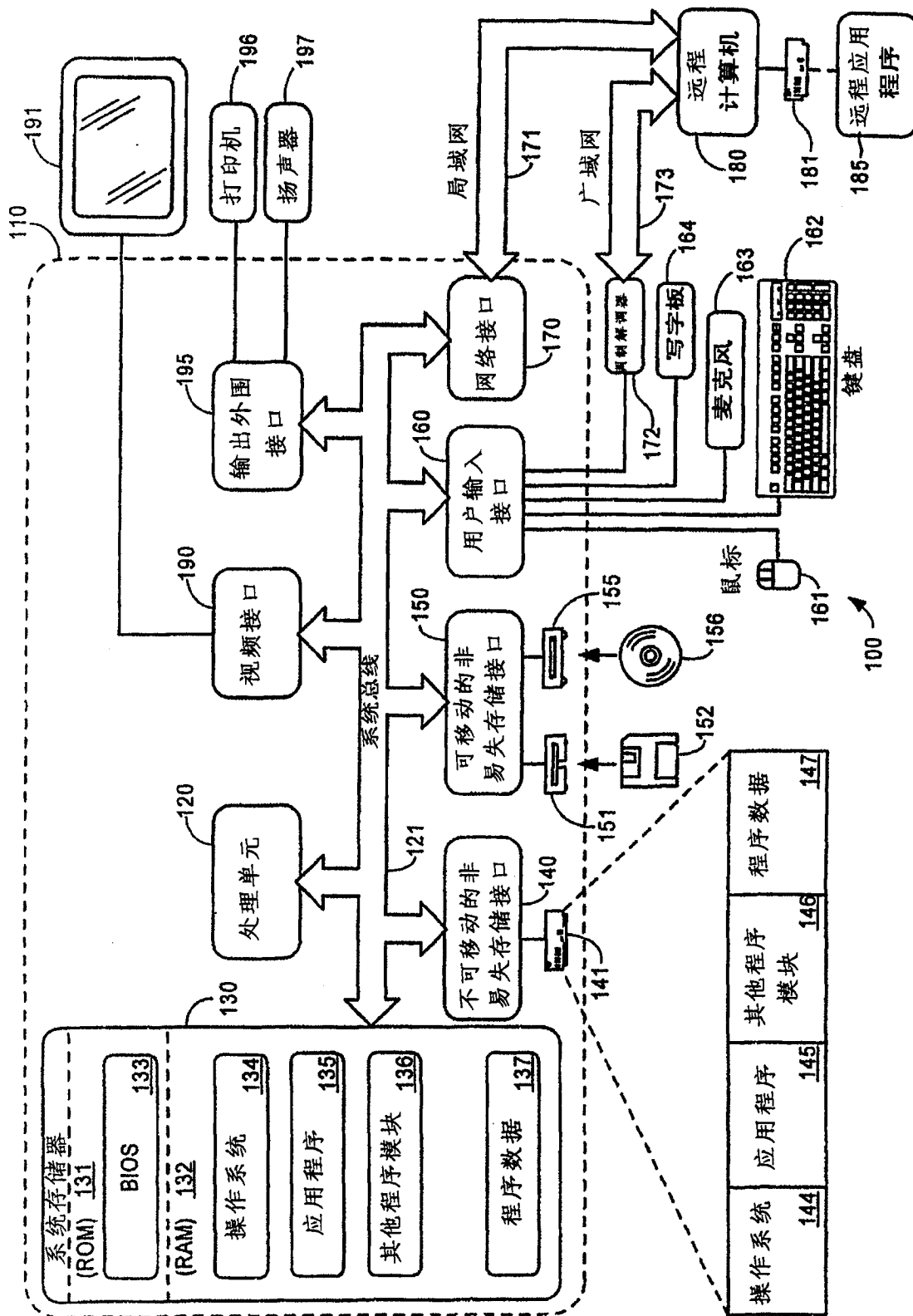


图 1

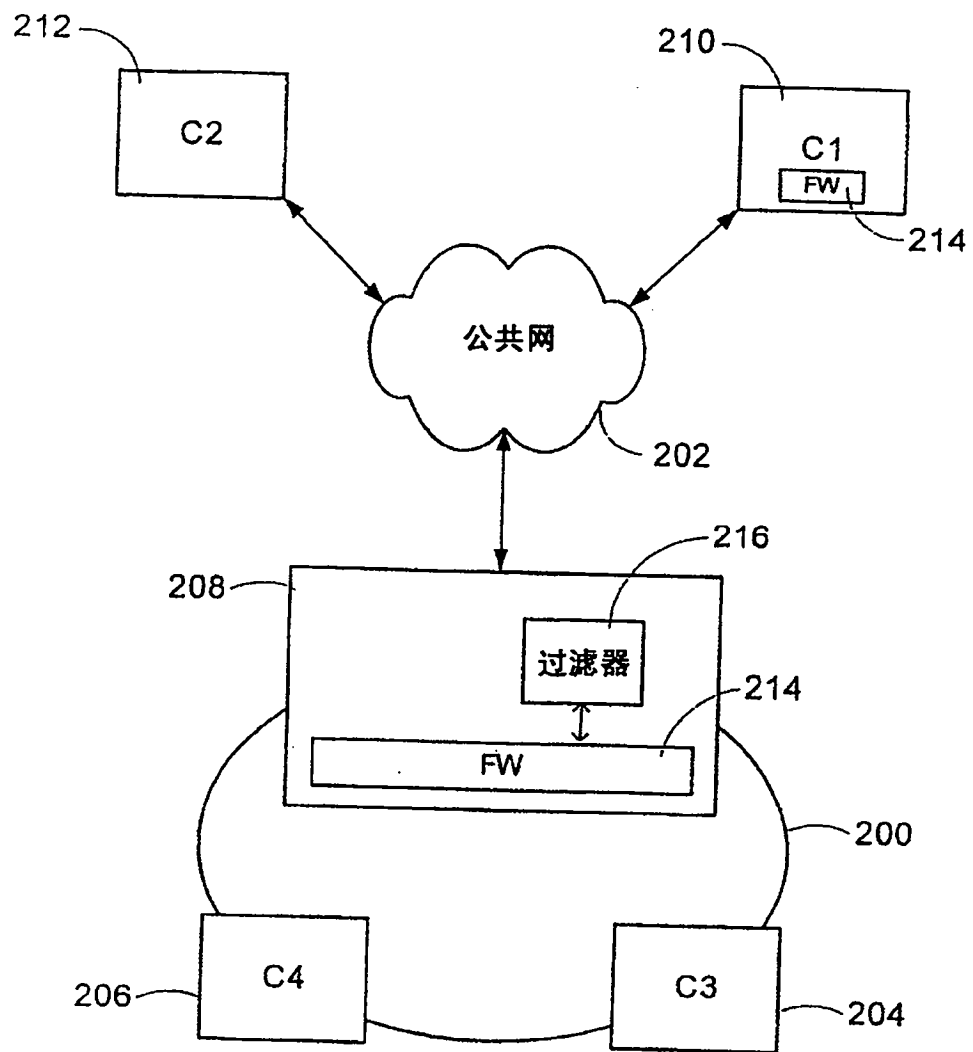


图 2

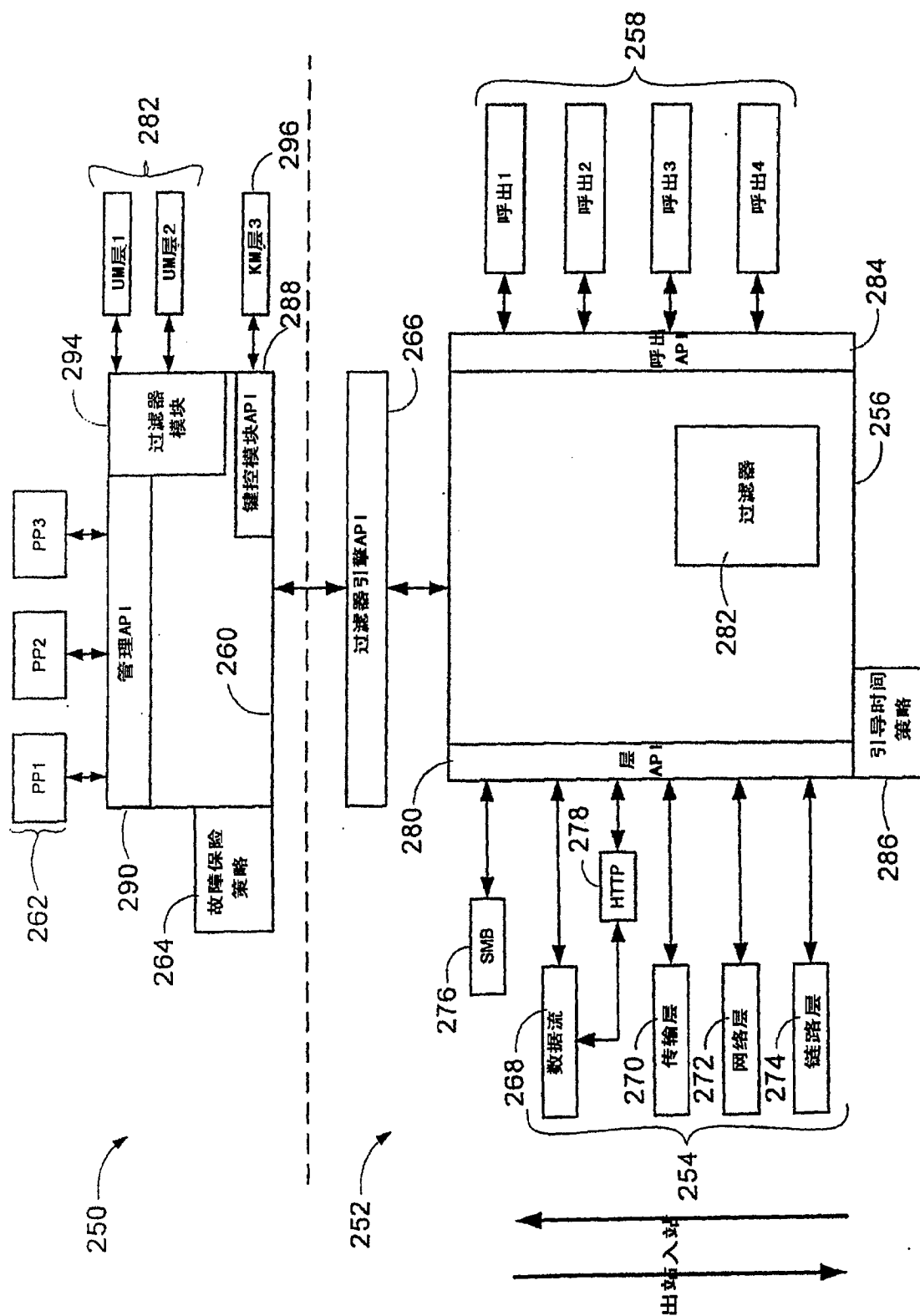


图 3

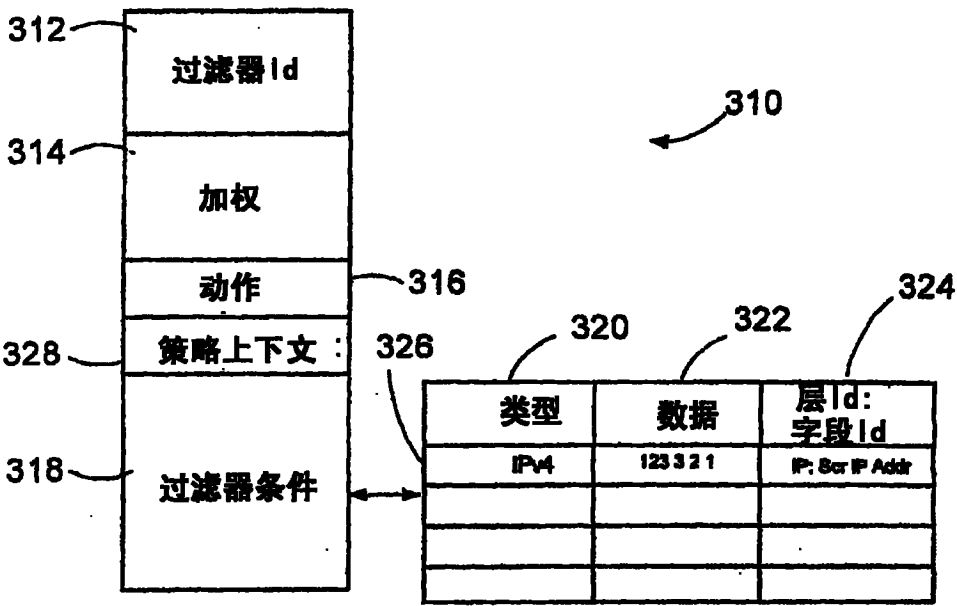


图 4

层Id: 字段Id	值
NDIS:Src MAC Addr.	00.80.74.4F.22.E5
NDIS:IF No.	2
IP:Dst IP Addr.	123.3.2.1

图 5

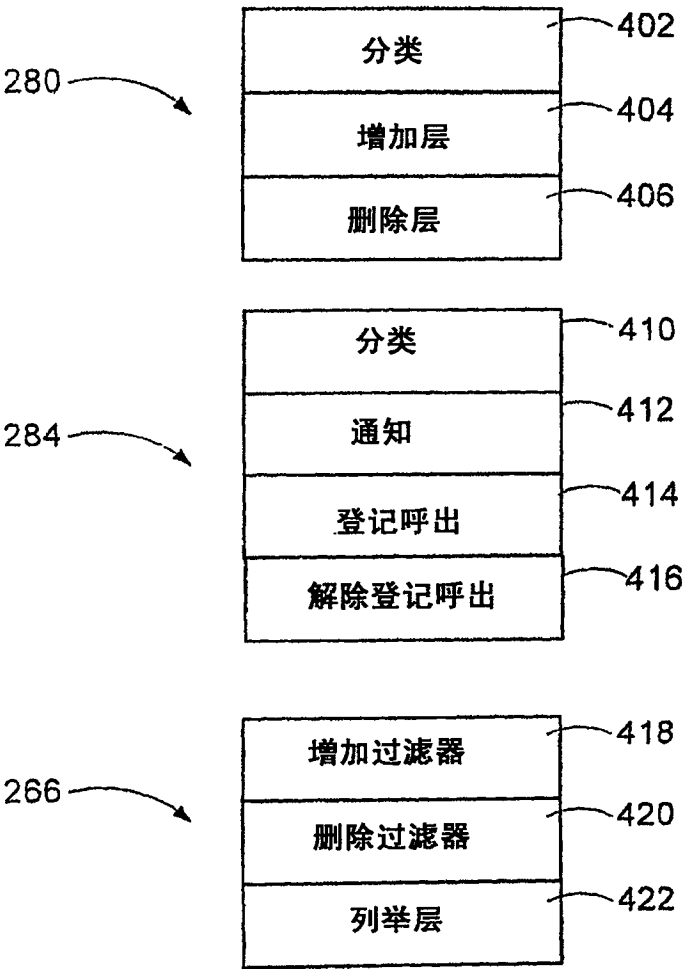


图 6

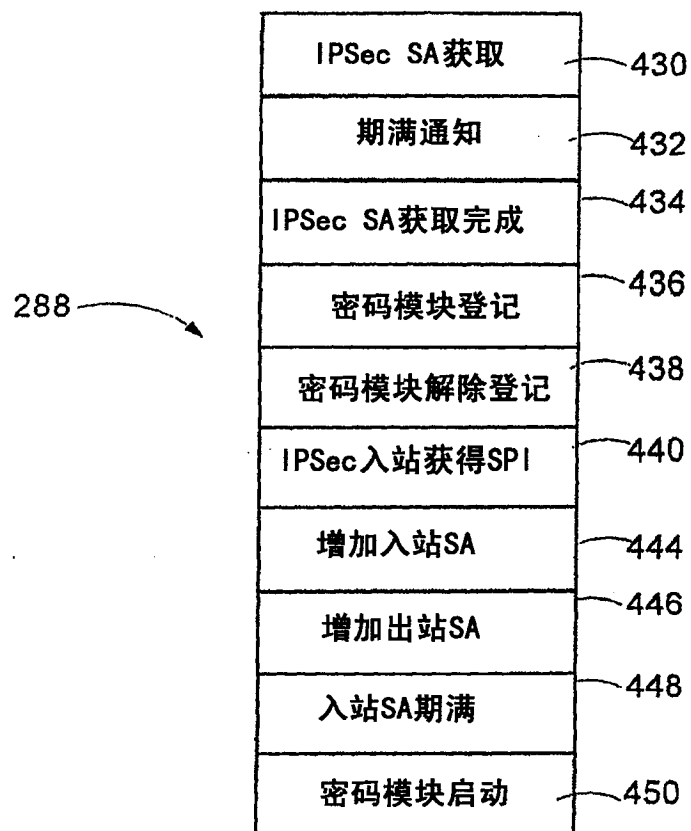


图 7

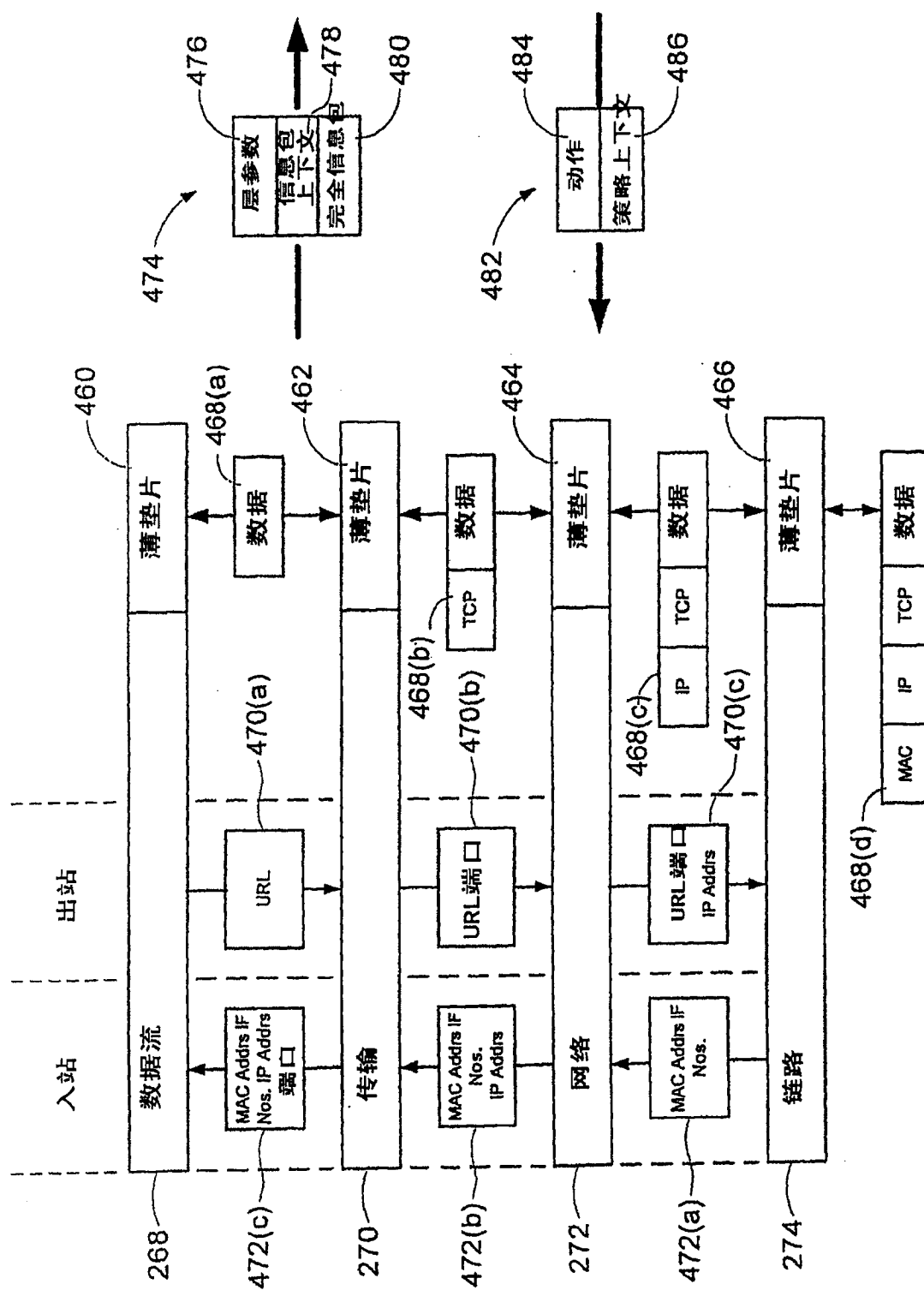


图 8

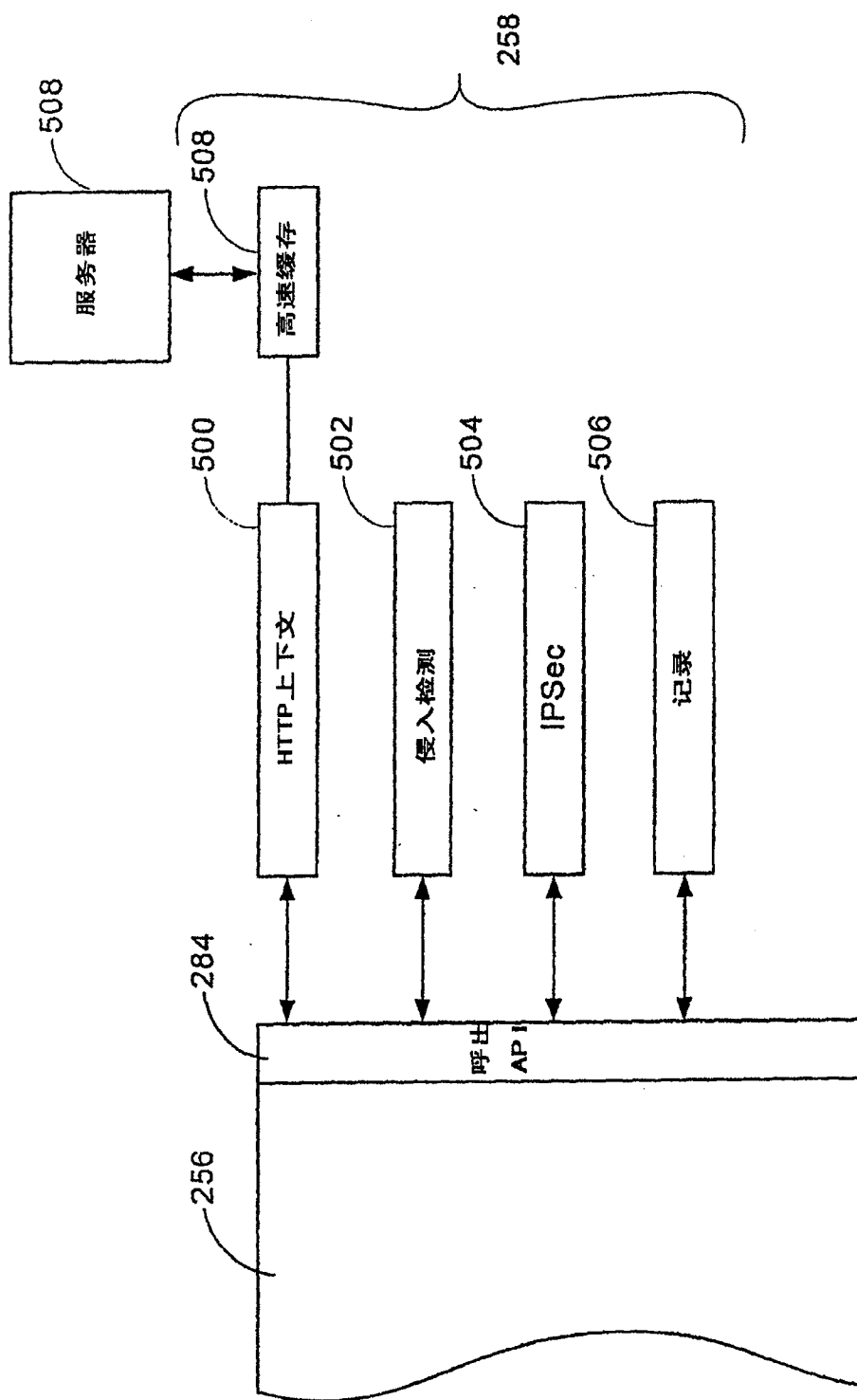


图 9