República Federativa do Brasil
Ministério da Indústria, Comércio Exterior
e Serviços
Instituto Nacional da Propriedade Industrial

(11) PI 0403261-6 B1

(22) Data do Depósito: 05/08/2004

(45) Data de Concessão: 13/06/2017

(54) Título: SISTEMA ADAPTADO PARA EXECUTAR SISTEMAS OPERACIONAIS PLURAIS EM UM ÚNICO PROCESSADOR E MÉTODO PARA MONITORAR UM AMBIENTE OPERACIONAL NÃO CONFIÁVEL DESTE SISTEMA

(51) Int.Cl.: G06F 9/46

(30) Prioridade Unionista: 07/08/2003 US 10/638.199

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC

(72) Inventor(es): BRYAN MARK WILLMAN; PAUL ENGLAND; KENNETH D. RAY; KEITH KAPLAN; VARUGIS KURIEN; MICHAEL DAVID MARR

"SISTEMA ADAPTADO PARA EXECUTAR SISTEMAS OPERACIONAIS PLURAIS EM UM ÚNICO PROCESSADOR E MÉTODO PARA MONITORAR UM AMBIENTE OPERACIONAL NÃO CONFIÁVEL DESTE SISTEMA".

CAMPO DA INVENÇÃO

[001] Esta invenção se relaciona em geral com o campo de segurança de computador. Mais particularmente, esta invenção relaciona-se com o uso de vários ambientes de execução (por exemplo, sistemas operacionais) em um único dispositivo de computação e proporciona técnicas que suportam a confiabilidade de tais sistemas operacionais ou ambientes.

FUNDAMENTO DA INVENÇÃO

[002] Os primeiros computadores somente estavam aptos a executar um único programa por vez. Entretanto, nos tempos modernos, os computadores são esperados de estarem aptos a executarem várias partes de software de uma vez. Por exemplo, os sistemas operacionais com múltiplas tarefas típicos podem executar vários programas de aplicação de uma vez em uma única máquina. Em vista disto e da evolução de uma rede compartilhada, aberta (isto é, a Internet), a segurança e a privacidade se tornaram dois pontos importantes e difíceis encarando a indústria de computadores. À medida que o computador pessoal se desenvolve mais central para lares, trabalho e escola, os consumidores e os clientes de negócios da mesma forma estão cada vez mais atentos às questões de privacidade e segurança. Aperfeiçoar a habilidade do software e do hardware para proteger a integridade da informação digital e a privacidade dos usuários de computador se tornou um foco crítico tanto para os desenvolvedores de software como para os fabricantes de hardware. A Microsoft Corporation, Redmond, WA, introduziu a plataforma de computador pessoal Base de Computação Segura de Próxima Geração (NGSCB), a qual proporciona segu-

rança e privacidade em um sistema operacional.

[003] Na NGSCB convencional dentro de um computador 110, como apresentado na Figura 2, um sistema de segurança "do lado direito" (RHS) funciona em conjunto com um sistema "lado esquerdo" (LHS) tradicional e com a unidade de processamento central (CPU). O RHS é projetado para proteger contra software malicioso enquanto preservando a acessibilidade do sistema operacional. Com o NGSCB, as aplicações rodam em um espaço de memória protegido que é altamente resistente à falsificação e interferência no software. Tipicamente, existe um conjunto de chips no computador 110 que tanto o LHS como o RHS utilizam. O LHS e o RHS são uma divisão ou particionamento lógico, porém fisicamente imposto, do computador 110.

[004] O LHS compreende as aplicações tradicionais 205, 210, tal como o Microsoft® Word® e o Microsoft® Excel®, junto com um sistema operacional convencional 201, tal como o sistema operacional Microsoft® Windows®. Apesar de duas aplicações serem apresentadas, tipicamente qualquer número pode ser implementado.

[005] O RHS compreende os agentes confiáveis 255, 260, junto com um "nexus" 251. Um nexus é um sistema operacional de "alta segurança" que proporciona um certo nível de segurança quanto para o seu comportamento e pode compreender todo o código de modo de núcleo (Kernel) no RHS. Por exemplo, um nexus pode ser empregado para trabalhar com informação secreta (por exemplo, chaves criptográficas, etc.) que não deve ser divulgada, por se proporcionar uma memória encoberta que é garantida de não divulgar informação para o mundo do lado de fora do nexus e por permitir somente certas aplicações certificadas se executem sob o nexus e acessem a memória encoberta. O nexus 251 não deve interagir com o sistema operacional principal 201 de qualquer modo que permitiria que eventos acontecendo no sistema operacional principal 201 comprometessem o compor-

tamento do nexus 251. O nexus 251 pode permitir a todas as aplicações executarem ou o dono de uma máquina pode configurar uma política de máquina na qual o nexus 251 permite que somente certos agentes rodem. Em outras palavras, o nexus 251 irá executar qualquer agente que o dono da máquina deixar executar. O dono da máquina também pode indicar ao nexus o que não executar.

[006] O nexus 251 isola os agentes confiáveis 255, 260, gerencia as comunicações para e a partir dos agentes 255, 260 e criptograficamente lacra os dados armazenados (por exemplo, armazenados em um controlador de disco rígido). Mais particularmente, o nexus 251 executa no modo de núcleo no espaço confiável e proporciona serviços básicos para os agentes confiáveis 255, 260, tal como o estabelecimento dos mecanismos de processo para comunicação com os agentes confiáveis e com as outras aplicações e serviços de confiança especiais tal como atestação de uma plataforma de hardware / software ou ambiente de execução e a lacre e deslacre dos segredos. A atestação é a habilidade de uma parte de código digitalmente assinar ou de outro modo atestar uma parte de dados e adicionalmente garante ao receptor que os dados foram construídos por uma pilha de software criptograficamente identificada, que não pode ser falsificada.

[007] Um ambiente confiável é um programa, uma parte de um programa ou um serviço que roda no modo de usuário no espaço confiável. Um agente confiável 255, 260 chama o nexus 251 para serviços relacionados à segurança e serviços gerais críticos, tal como gerenciamento da memória. Um agente confiável está apto a armazenar segredos utilizando o armazenamento lacrado e se autentica utilizando os serviços de atestação do nexus. Cada agente ou entidade confiável controla seu próprio domínio de confiança e eles não precisam confiar um no outro.

[008] O RHS adicionalmente compreende um componente de su-

porte de segurança (SSC) 253 que utiliza um par de chaves da infraestrutura de chave pública (PKI) junto com funções de criptografia para proporcionar um estado segura.

[009] A NGSCB proporciona características tal como "atestação", "armazenamento lacrado" e "isolamento de processo forte". A atestação permite à outros computadores saberem que um computador é realmente o computador que diz ser e está executando o software que diz estar executando. Devido ao fato do software e hardware NGSCB serem criptograficamente verificáveis para o usuário e para outros computadores, programas e serviços, o sistema pode verificar que outros computadores e processos são de confiança antes de empregá-los ou compartilhar informação. Assim, a atestação permite ao usuário revelar as características selecionadas do ambiente operacional para requisitantes externos.

[0010] O armazenamento lacrado permite ao usuário criptografar informação de modo que ela somente pode ser acessada por uma aplicação de confiança. Isto pode incluir somente a aplicação que criou a informação no primeiro lugar ou qualquer informação que é confiada pela aplicação que possui os dados. Portanto, o armazenamento lacrado permite à um programa armazenar segredos que não podem ser recuperados por programas não confiáveis, tal como um vírus ou cavalo de Tróia.

[0011] O isolamento de processo forte proporciona um espaço confiável por se esculpir uma área segura (o RHS). As operações que rodam no RHS são protegidas e isoladas do LHS, o que as torna significativamente mais seguras contra ataques.

[0012] A NGSCB também proporciona entrada e saída segura. Com a NGSCB, os pressionamentos de teclas são criptografados antes deles poderem ser lidos por software e decriptografados uma vez que eles alcançam o RHS. Isto significa que o software malicioso não

pode ser utilizado para gravar, roubar ou modificar os pressionamentos de teclas. A saída segura é similar. A informação que aparece na tela pode ser apresentada ao usuário de modo que mais ninguém pode interceptá-la e lê-la. Levadas em consideração juntas, estas funções permitem a um usuário saber com um alto grau de confiança que o software no seu computador está fazendo o que é suposto de se fazer.

[0013] A despeito dos recursos de confiança substanciais disponíveis para o RHS, o LHS permanece não confiável. A presente invenção endereça esta e outras deficiências dos sistemas de computação confiáveis correntes.

SUMÁRIO DA INVENÇÃO

[0014] A presente invenção proporciona um mecanismo para projetar a confiabilidade das entidades em um ambiente confiável para entidades em um ambiente não confiável.

[0015] São descritos sistemas e métodos nos quais um ambiente não confiável e um ambiente confiável são proporcionados. Um agente de monitoramento base roda no ambiente confiável. O agente de monitoramento base monitora o ambiente não confiável.

[0016] De acordo com uma modalidade, o agente de monitoramento está associado com uma aplicação e o agente de monitoramento monitora sua aplicação associada para eventos ou comportamentos que podem indicar um ataque. A natureza confiável do agente monitor permite que estes eventos / comportamentos sejam detectados e relatados de forma confiável, deste modo projetando a confiabilidade do ambiente confiável no ambiente não confiável. O agente de monitoramento base pode aprovar, proibir ou modificar um evento do ambiente não confiável relatado ou descoberto pelo agente de monitoramento. Relatórios para cobrir casos tal como o hardware relatando uma tentativa de mover a GDT (tabela descritora global) para o nexus, o qual por sua vez iria relatar isto para o agente de monitoramento base, por

exemplo. A descoberta seria um caso no qual o agente de monitoramento base (para o OS) ou um agente de monitoramento (para alguma aplicação) descobre um problema por se analisar a memória da aplicação não confiável, por exemplo.

[0017] Para outra modalidade, o agente de monitoramento base responde para a entrada recebida a partir de uma entrada segura. Por exemplo, o agente de monitoramento base pode recusar permitir alterações junto ao ambiente não confiável sem receber aprovação via uma entrada segura. Como outro exemplo, o agente de monitoramento base pode recusar permitir alterações junto ao ambiente não confiável a menos que as alterações sejam descritas por um pacote que é assinado por uma parte aprovada.

[0018] Ainda para outra modalidade, o agente de monitoramento utiliza o armazenamento lacrado para manter um segredo para um sistema operacional ou uma aplicação residindo no ambiente não confiável. O agente de monitoramento pode recusar revelar o segredo para o sistema operacional ou para a aplicação a menos que o sistema operacional ou aplicação tenham um transformador que associa o dono do segredo. De forma alternativa, o agente de monitoramento pode recusar revelar o segredo para o sistema operacional ou aplicação a menos que o sistema operacional ou aplicação estejam em uma lista de transformadores que podem ler o segredo.

[0019] De acordo com outras características, o agente de monitoramento utiliza um teste para determinar se uma entidade legítima está requisitando o segredo. Um tal teste inclui examinar as pilhas da entidade e garantir que as pilhas possuem conteúdos de pilhas legais. Além do mais, o agente de monitoramento pode editar um estado do ambiente não confiável para torná-lo seguro ou de outro modo aceitável. O estado pode compreender uma configuração inicial ou uma opção de relatório de erros.

[0020] As características e vantagens adicionais da invenção irão ser feitas aparentes a partir da descrição detalhada seguinte das modalidades ilustrativas que prosseguem com referência aos desenhos acompanhantes.

BREVE DESCRIÇÃO DOS DESENHOS

[0021] O sumário precedente, bem como a descrição detalhada das modalidades preferidas, são melhor entendidos em conjunto com os desenhos anexos. Para o propósito de ilustrar a invenção, são apresentadas nos desenhos as construções ilustrativas da invenção; entretanto, a invenção não está limitada aos métodos específicos e instrumentos reveladas. Nos desenhos:

A Figura 1 é um diagrama de blocos apresentando um ambiente de computação ilustrativo no qual os aspectos da invenção podem ser implementados;

A Figura 2 é um diagrama de blocos de um sistema NGSCB existente possuindo ambientes tanto confiáveis como não confiáveis;

A Figura 3 é um diagrama de blocos de um sistema de projeção ilustrativo de acordo com a presente invenção; e

A Figura 4 é um fluxograma de um método ilustrativo de projeção de acordo com a presente invenção.

DESCRIÇÃO DETALHADA DAS MODALIDADES PREFERIDAS

VISTA GERAL

[0022] Em uma única máquina que possui entidades executando em um ambiente não confiável e entidades executando em um ambiente confiável, a presente invenção proporciona um mecanismo para projetar a confiabilidade das entidades no ambiente confiável para as entidades no ambiente não confiável A invenção é direcionada para mecanismos utilizados quando um primeiro ambiente de execução (por exemplo, um sistema operacional) hospeda um segundo ambiente de execução. A invenção aplica-se à situações tal como a Base de Com-

putação Segura de Próxima Geração (NGSCB) da Microsoft®, onde o sistema operacional normal (por exemplo, o sistema operacional Windows®) hospeda um sistema operacional seguro (por exemplo, o nexus). Vários mecanismos são descritos, os quais permitem ao segundo ambiente projetar sua confiabilidade para o primeiro ambiente.

AMBIENTE DE COMPUTAÇÃO ILUSTRATIVO

[0023] A Figura 1 ilustra um exemplo de um ambiente de sistema de computação adequado 100 no qual os aspectos da presente invenção podem ser implementados. O ambiente de sistema de computação 100 é somente um exemplo de um ambiente de computação adequado e não é pretendido para sugerir qualquer limitação quanto ao escopo de uso ou quanto a funcionalidade da invenção. Nem deve o ambiente de computação 100 ser interpretado como possuindo qualquer dependência ou requerimento relacionando-se com qualquer um ou com uma combinação dos componentes ilustrados no ambiente operacional ilustrativo 100.

[0024] A invenção é operacional com vários outros ambientes ou configurações de sistema de computação de propósito geral ou especial. Exemplos de sistemas, ambientes e/ou configurações de computação bem conhecidos que podem ser adequados para uso com a invenção incluem, mas não estão limitados aos computadores pessoais, aos computadores servidores, aos dispositivos de mão ou laptop, aos sistemas com múltiplos processadores, aos sistemas baseados em microprocessador, às caixas decodificadoras, aos telefones móveis, aos eletrônicos programáveis de consumidor, aos PCs de rede, aos minicomputadores, aos computadores de grande porte, aos ambientem de computação distribuída que incluem qualquer um dos sistemas ou dispositivos acima e coisa parecida.

[0025] A invenção pode ser descrita no contexto geral de instruções executáveis por computador, tal como módulos de programa,

sendo executados por um computador. Geralmente, os módulos de programa incluem rotinas, programas, objetos, componentes, estruturas de dados, etc. que executam tarefas particulares ou implementam tipos de dados abstratos particulares. A invenção também pode ser praticada em ambientes de computação distribuída, onde as tarefas são executadas por dispositivos de processamento remotos que estão ligados através de uma rede de comunicações ou por outro meio de transmissão de dados. Em um ambiente de computação distribuída, os módulos de programa e os outros dados podem estar localizados tanto no meio de armazenamento local como remoto, incluindo os dispositivos de armazenamento em memória.

[0026] Com referência à FIG. 1, um sistema ilustrativo para implementar a invenção inclui um dispositivo de computação de propósito geral na forma de um computador 110. Os componentes do computador 110 podem incluir, mas não estão limitados à unidade de processamento 120, a uma memória do sistema 130 e ao barramento do sistema 121 que acopla vários componentes do sistema, incluindo a memória do sistema com a unidade de processamento 120. O barramento do sistema 121 pode ser qualquer um dentre vários tipos de estruturas de barramento incluindo um barramento de memória ou controlador de memória, um barramento periférico e um barramento local utilizando qualquer uma dentre uma variedade de arquiteturas de barramento. A título de exemplo e não de limitação, tais arquiteturas incluem o barramento da Arquitetura Padrão da Industria (ISA), o barramento da Arquitetura de Micro Canal (MCA), o barramento ISA Aperfeiçoado (EISA), o barramento local da Associação de Padrões de Componentes Eletrônicos de Vídeo (VESA) e o barramento de Interconexão de Componente Periférico (PCI) (também conhecido como barramento Mezanino).

[0027] O computador 110 tipicamente inclui uma variedade de meios legíveis por computador. O meio legível por computador pode

ser qualquer meio disponível que possa ser acessado pelo computador 110 e inclui tanto o meio volátil como o não volátil. o meio removível e não removível. A título de exemplo e não de limitação, o meio legível por computador pode compreender o meio de armazenamento do computador e o meio de comunicação. O meio de armazenamento do computador inclui tanto o meio volátil como o não volátil e o meio removível como o não removível implementado em qualquer método ou tecnologia para armazenamento de informações tal como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. O meio de armazenamento do computador inclui, mas não está limitado à RAM, ROM, EEPROM, memória flash ou a outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento ótico, cassetes magnéticos, fita magnética, armazenado em disco magnético ou a quaisquer outros dispositivos de armazenamento magnético, ou a qualquer outro meio que possa ser utilizado para armazenar a informação desejada e que possa ser acessado pelo computador 110. O meio de comunicação tipicamente incorpora as instruções legíveis por computador, estruturas de dados, módulos de programa e outros dados em um sinal de dados modulado tal como uma onda portadora ou outro mecanismo de transporte e inclui qualquer meio de distribuição de informação. O termo "sinal de dados modulado" significa um sinal que possui uma ou mais dentre suas características estabelecidas ou alteradas de uma maneira tal a codificar a informação no sinal. A título de exemplo e não de limitação, o meio de comunicação inclui meio com fios tal como uma rede com fios ou conexão com fios direta e meio sem fios tal como um meio sem fios acústico, RF, infravermelho e outros meios sem fios. Combinações de qualquer um dos ditos acima também devem estar incluídas dentro do escopo de meio legível por computador.

[0028] A memória do sistema 130 inclui o meio de armazenamento

do computador na forma de memória volátil e/ou não volátil tal como a ROM 131 e a RAM 132. Um sistema básico de entrada/saída (BIOS) 133, contendo as rotinas básicas que ajudam a transferir informação entre os elementos dentro do computador 110, tal como durante a inicialização, tipicamente está armazenado na ROM 131. A RAM 132 tipicamente contém dados e/ou módulos de programa que são imediatamente acessíveis e/ou atualmente sendo operados pela unidade de processamento 120. A título de exemplo e não de limitação, a Figura 1 ilustra o sistema operacional 134, os programas de aplicação 135, outros módulos de programa 136 e os dados de programa 137.

[0029] O computador 110 também pode incluir outro meio de armazenamento do computador removível, não removível, volátil/não volátil. Somente a título de exemplo, a Figura 1 ilustra um controlador de disco rígido 140 que lê e grava junto ao meio magnético não removível, não volátil, um controlador de disco magnético 151 que lê ou grava junto a um disco magnético removível, não volátil 152 e um controlador de disco ótico 155 que lê ou grava junto a um disco ótico removível, não volátil 156, tal como um CD-ROM ou outro meio ótico. Outros meios de armazenamento do computador removíveis/não removíveis, voláteis/não voláteis que podem ser utilizados no ambiente operacional ilustrativo incluem, mas não estão limitados aos cassetes de fita magnética, cartões de memória flash, discos versáteis digitais, fita de vídeo digital, RAM de estado sólido, ROM de estado sólido e coisas parecidas. O controlador de disco rígido 141 tipicamente está conectado com o barramento do sistema 121 através de uma interface de memória não removível tal como a interface 140 e o controlador de disco magnético 151 e o controlador de disco ótico 155 tipicamente estão conectados com o barramento do sistema 121 por uma interface de memória removível, tal como a interface 150. É adicionalmente contemplado que a presente invenção também pode ser implementada em um mi-

croprocessador embutido no qual a CPU e toda a memória estão em uma único matriz em um único pacote.

[0030] Os controladores e seus meios de armazenamento do computador associados discutidos acima e ilustrados na Figura 1 proporcionam armazenamento das instruções legíveis por computador, das estruturas de dados e dos módulos de programa e de outros dados para o computador 110. Na Figura. 1, por exemplo, o controlador de disco rígido 141 é ilustrado como armazenando o sistema operacional 144, os programas de aplicação 145, os outros módulos de programa 146 e os dados de programa 147. Observe que estes componentes podem ser os mesmos ou diferentes do sistema operacional 134, dos programas de aplicação 135, dos outros módulos de programa 136 e dos dados de programa 137. São dados números diferentes aqui dentro para o sistema operacional 144, para os programas de aplicação 145, para os outros módulos de programa 146 e para os dados de programa 147 para ilustrar que no mínimo, eles são cópias diferentes. Um usuário pode entrar com comandos e informação dentro do computador 110 através dos dispositivos de entrada tal como um teclado 162 e um dispositivo de apontamento 161, normalmente referido como um mouse, uma trackball ou plataforma de toque. Outros dispositivos de entrada (não apresentados) podem incluir um microfone, joystick, plataforma de jogo, antena de satélite, digitalizador, ou coisa parecida. Estes e outros dispositivos de entrada são frequentemente conectados com a unidade de processamento 120 através de uma interface de entrada do usuário 160 que está acoplada com o barramento do sistema, mas pode estar conectada por outra interface e estruturas de barramento, tal como uma porta paralela, uma porta de jogo ou um barramento serial universal (USB). Um monitor 191 ou outro tipo de dispositivo de exibição também está conectado com o barramento do sistema 121 via uma interface, tal como uma interface de vídeo 190. Em adição ao mo-

nitor, os computadores também podem incluir outros dispositivos periféricos de saída tal como alto-falantes 197 e a impressora 196, os quais podem estar conectados através de uma interface de periférico de saída 195.

[0031] O computador 110 pode operar em um ambiente em rede utilizando conexões lógicas com um ou mais computadores remotos, tal como um computador remoto 180. O computador remoto 180 pode ser um computador pessoal, um servidor, um roteador, um PC de rede, um dispositivo par ou outro nó comum de rede e tipicamente inclui vários ou todos os elementos descritos acima em relação ao computador 110, apesar de somente um dispositivo de armazenamento em memória 181 ter sido ilustrado na Figura 1. As conexões lógicas descritas incluem uma rede de área local (LAN) 171 e uma rede de área ampla (WAN) 173, mas também podem incluir outras redes. Tais ambientes em rede são comuns em escritórios, em rede de computadores de grandes empresas, intranets e na Internet.

[0032] Quando utilizado em um ambiente em rede LAN, o computador 110 está conectado com a LAN 171 através de uma interface ou adaptador de rede 170. Quando utilizado em um ambiente em rede WAN, o computador 110 tipicamente inclui um modem 172 ou outro dispositivo para estabelecer as comunicações através da WAN 173, tal como a Internet. O modem 172, o qual pode ser interno ou externo, pode ser conectado com o barramento do sistema 121 via a interface de entrada do usuário 160, ou por outro mecanismo apropriado. Em um ambiente em rede, os módulos de programa descritos em relação ao computador 110, ou partes dos mesmos, podem ser armazenados no dispositivo de armazenamento em memória remoto. A título de exemplo e não de limitação, a Figura 1 ilustra os programas de aplicação remotos 185 como residindo no dispositivo de memória 181. Será apreciado que as conexões de rede apresentadas são ilustrativas e

que outros dispositivos para estabelecer uma ligação de comunicação entre os computadores podem ser utilizados.

MODALIDADES ILUSTRATIVAS

[0033] Como descrito previamente, é conhecido na técnica que um computador pode ser configurado para proporcionar dois ambientes distintos: confiável e não confiável. O código comum cuja confiabilidade não foi verificada (isto é, código cujo comportamento não foi verificado, ou que não pode ser controlado de possivelmente servir um propósito malévolo) roda no ambiente não confiável. O software de aplicação comum, tal como jogos, processadores de texto, planilhas, etc. bem como sistemas operacionais comuns, controladores de dispositivo e depuradores, geralmente caem na categoria não confiável. O código cuja confiabilidade foi verificada de alguma maneira pode executar no ambiente confiável. Alguma parte da memória do computador (isto é, a memória "isolada" ou "encoberta") é projetada como sendo acessível somente para o ambiente confiável.

[0034] Para a discussão seguinte, um agente é "confiável" se ele foi instanciado de acordo com um procedimento seguro projetado para preservar sua integridade ou tornar aparente qualquer brecha de sua integridade. Por exemplo, o agente pode ser iniciado através de um procedimento confiável que verifica a identidade do agente e o ambiente no qual ele está executando (atestação), pode ser designada uma localização de memória segura (memória encoberta) que não é acessível para nenhum outro agente, confiável ou não confiável e ele pode ser capaz de lacrar segredos. Tal agente confiável pode ser unicamente e de forma confiável identificado.

[0035] No ambiente confiável, existem limitações em relação ao que o código está permitido de fazer. Por exemplo, existem menos APIs confiáveis (versus o conjunto bastante rico de API em um LHS típico), agentes executando no ambiente confiável somente podem se

comunicar uns com os outros via os mecanismos de Comunicação de Inter-Processo formal restrito (IPC) e os agentes podem ter acesso à um conjunto de APIs e serviços mais restrito e primitivo para apresentar texto e imagens para o usuário. Estas limitações reduzem a complexidade e, por consequência, a superfície de ataque do ambiente confiável e dos agentes confiáveis que operam dentro da mesma. O ambiente não confiável, por outro lado, é similar ao ambiente tipicamente criado pelo sistema operacional em um sistema de computação "aberto" (por exemplo, um computador pessoal, um computador portátil, etc.) – isto é, quase qualquer código é permitido de executar em tal ambiente não confiável e o código executando no ambiente padrão possui acesso total à um grande e rico conjunto de serviços de programação e interfaces. O ambiente não confiável e o ambiente confiável podem ser adicionalmente divididos em subambientes. Por exemplo, o ambiente não confiável pode ser dividido em um modo de usuário não confiável (onde a aplicação comum executa) e um modo de núcleo não confiável (onde o sistema operacional comum executa). De forma similar, o ambiente confiável pode ser dividido em um modo de usuário confiável (onde aplicações especiais, confiáveis executam) e um modo de núcleo confiável (onde o sistema operacional confiável que cria o ambiente confiável para aplicações confiáveis executa).

[0036] Quando os ambientes confiáveis e não confiáveis coexistem no mesmo sistema de computador, o ambiente confiável pode realizar etapas para garantir que sua confiabilidade não possa ser afetada por nada que acontece no ambiente não confiável, ou por qualquer outro código de modo do usuário no ambiente confiável. As modalidades da presente invenção proporcionam um mecanismo para projetar ou de outro modo utilizar a confiabilidade do lado confiável para o benefício do lado não confiável.

[0037] A Figura 3 é um diagrama de blocos de uma modalidade de

um sistema de projeção de acordo com a presente invenção e a Figura 4 é um fluxograma de uma modalidade do método de projeção de acordo com a presente invenção. O LHS do sistema executando no computador 110 é similar à este descrito acima com respeito á Figura 2. Duas aplicações 305, 310 estão executando em conjunto com um sistema operacional 301. Partes do RHS também são similares em relação à esta descrita com respeito à Figura 2. Dois agentes confiáveis 355, 360 estão executando junto com um nexus 351 e o SSC 353. É contemplado que qualquer número de aplicações pode ser executado no LHS e qualquer número de agentes confiáveis pode executar no RHS.

[0038] A Figura 3 apresenta um sistema no qual o sistema operacional 301 e o nexus 351 executam em um único computador 110. Uma separação lógica 350 entre o sistema operacional 301 e o nexus 351 permite que certas comunicações ocorram, enquanto protegendo o nexus 351 contra eventos que se originam no sistema operacional 301.

[0039] Na modalidade ilustrada pela Figura 3, o sistema operacional 301 é um sistema operacional hospedeiro e o nexus 351 é um hóspede hospedado pelo OS 301. Isto é, o OS 301 proporciona certos serviços e recursos para o nexus 351, tal como memória e tempo do processador. Para uma modalidade, a separação lógica 350 permite ao nexus 351 confiar somente em certos recursos do sistema operacional 301, enquanto ainda permitindo ao nexus 351 se proteger de ações (maliciosas ou inocentes) que surgem no sistema operacional 301 e podem causar que o nexus 351 se comporte de uma maneira contrária em relação às suas especificações comportamentais. Por exemplo, o nexus 351 e os recursos confiáveis associados com ele, por exemplo, o SSC 353, podem gerenciar a separação lógica. Entretanto, será entendido que a invenção não está limitada à forma particular do nexus 351. São contemplados mecanismos que permitem à se-

paração 350 ser construída de modo a permitir este equilíbrio de interação e proteção.

[0040] Deve ser notado que a Figura 3 apresenta o sistema operacional 301 como um "hospedeiro" e o nexus 351 como um "hóspede". Em geral, esta caracterização refere-se ao fato de que, nestes exemplos, o sistema operacional 301 proporciona certa infraestrutura de sistema operacional que é utilizada pelos sistemas operacionais 301 e pelo nexus 351 (por exemplo, controladores de dispositivos, programação de execução, etc.). O nexus 351 é um "hóspede" no sentido que ele pode contar com certos recursos de infraestrutura do sistema operacional 301 ao invés dele próprio proporcioná-los. Entretanto, deve ser notado que os parâmetros do que faz de um sistema operacional um "hospedeiro" ou um "hóspede" são flexíveis. Deve ser apreciado que as técnicas descritas aqui dentro podem ser aplicadas para a interação de quaisquer dois ou mais sistemas operacionais executando na mesma máquina (ou mesmo no mesmo conjunto de máquinas conectadas). Dois ou mais sistemas operacionais que rodam em uma única máquina são exemplos de "ambientes" que podem precisar interagir uns com os outros em uma única máquina, apesar de que será entendido que a invenção não está limitada aos sistemas operacionais tradicionais.

[0041] A projeção é o mecanismo pelo qual algumas das capacidades e propriedades dos agentes confiáveis (no RHS) podem ser estendidas para o código LHS. De acordo com um exemplo, a projeção permite que as capacidades da plataforma de computador pessoal NGSCB sejam aplicadas ao código existente. Por exemplo, ao invés de portar uma aplicação tal como o Microsoft® Excel® para o RHS, a projeção de acordo com a invenção permite a construção de um agente de monitoramento (também referido aqui dentro como um anjo) para a aplicação (também referida aqui dentro como um mortal), o qual por

sua vez permite que a aplicação existente rode com várias das mesmas propriedades úteis como um agente confiável. A projeção pode ser aplicada tanto para o sistema operacional LHS (por exemplo, Microsoft® Windows®) como para quaisquer programas de aplicação LHS (por exemplo, Microsoft® Office®) para os quais algum nível de operação confiável é desejado. A projeção também pode ser aplicada aos controladores de dispositivo LHS. Assim, como descrito adicionalmente abaixo, a projeção permite aos agentes confiáveis protegerem, garantirem, atestarem e estenderem, os sistemas operacionais, serviços e programas LHS.

[0042] A Figura 3 apresenta um agente de monitoramento 390 que corresponde à aplicação 305 e o agente de monitoramento 395 que corresponde à aplicação 310 (etapa 400 na Figura 4). Cada agente ou anjo de monitoramento protege sua aplicação associada.

[0043] Para uma modalidade, o criador da entidade LHS de interesse (por exemplo, uma aplicação) também cria um anjo que guarda a entidade LHS. Isto permite ao criador proporcionar ao anjo um profundo conhecimento da aplicação que ele monitora. Tal anjo pode ser mais sensível em relação à anomalias na aplicação que ele monitora e então protegê-la e validá-la de forma mais eficaz. Por exemplo, um agente de monitoramento base criado por um desenvolvedor de sistema operacional pode incorporar conhecimento detalhado a cerca do gerenciamento de memória do sistema operacional que permite ao mesmo identificar operações de memória suspeitas rapidamente.

[0044] Para outra modalidade, um anjo pode executar ação corretiva ou preventiva se ele detectar uma atividade anômala ou suspeita em sua aplicação associada. Por exemplo, o anjo pode detectar uma tentativa pela sua aplicação associada de alterar uma variável chave na memória julgada invariável pelo criador da aplicação e interceptar a escrita para a variável. Tal operação de escrita provavelmente indica-

ria, pelo menos, corrupção do código da aplicação, se não subversão imediata por malévolo, por exemplo, código, de vírus. Em resumo, um anjo atua como um agente de monitoramento para observar atividades negativas ou suspeitas em sua aplicação associada e executar ação corretiva ou preventiva apropriada. Suas ações podem ser circunscritas para impedir o anjo de danificar sua aplicação associada. Um anjo pode ser anexado à uma entidade, programa ou aplicação particular, por exemplo, ou à um grupo de tais entidades, programas e / ou aplicações.

[0045] Um agente de monitoramento base (também referido aqui dentro como um arcanjo) 380 é associado com o sistema operacional base (isto é, o OS LHS 301) (bloco 410). Para uma modalidade, o agente de monitoramento base 380 é escrito pelo criador do sistema operacional LHS. Isto permite ao agente de monitoramento base 380 incorporar conhecimento detalhado a cerca do sistema operacional LHS, o que o torna mais sensível em relação à comportamento anômalo pelo sistema operacional associado.

[0046] Por exemplo, um arcanjo poderia saber o formato da base de dados de endereços virtuais, da base de dados de processos e da base de dados PFN (número de quadro de página) e baseado nisto, detectar casos nos quais controladores de dispositivos trapaceiros fizeram mapeamentos ilegais para processos por se mapear PFNs que eles não deveriam possuir. Assim, o arcanjo poderia detectar mapeamentos que não foram feitos pelo gerenciador de memória (por um controlador de dispositivo trapaceiro, por exemplo) e poderia detectar mapeamentos de processo cruzados que não deveriam estar lá.

[0047] Em tal caso, um arcanjo poderia conspirar com um OS mortal alterado. O OS e o arcanjo podem concordar, por exemplo, que a base de dados PFN deve sempre ser consistente sempre que um bloqueio particular não é manipulado e que esta consistência deve ser

representável via soma de verificação (checksum). Então, em intervalos periódicos, o arcanjo poderia inspecionar o bloqueio e encontrá-lo desbloqueado (ele é uma variável de memória e assim fácil de se testar), ir e executar soma de verificação na base de dados PFN. Se o arcanjo descobrir que a soma de verificação não corresponde, então ele sabe que a base de dados PFN foi falsificada.

[0048] Além do mais, um arcanjo poderia conhecer as variáveis de controle para o depurador do núcleo e forçar as variáveis de controle para desabilitar o uso do depurador do núcleo.

[0049] Um exemplo adicional inclui o carregamento de processo: monitorar o carregador, o gerenciador de memória cache, o manipulador de falha de página, etc. para garantir que os bits corretos sejam corretamente carregados em um processo de modo do usuário (ou qualquer outro módulo carregado no sistema), ou de forma apropriada assinado, talvez listado em uma lista de códigos numéricos mantidos dentro de uma tabela conhecida pelo arcanjo. O arcanjo estaria apto a antecipar quando o carregador, o manipulador de falha de página, etc. precisaria mapear código / dados para dentro ou para fora de um processo (paginação, etc.). O RHS poderia manter as páginas físicas LHS para este processo bloqueado (mesmo para o OS LHS) a menos que o OS estivesse executando funções boas conhecidas. O RHS controla as tabelas de página para os processos LHS. Assim, existe uma série de mecanismos que um escritor de arcanjo poderia incorporar no arcanjo para restringir comportamento ruim.

[0050] Um exemplo adicional inclui o endurecimento de processo. Existem mecanismos conhecidos e aprovados para um processo modificando outro processo. O arcanjo pode garantir que todos os mapas de memória compartilhada, bem como a cópia de dados para dentro ou para fora de um espaço de processo diferente sejam restritas. Outro exemplo envolve o núcleo somente de leitura, no qual todas as páginas

de "texto" (páginas de código) do núcleo e dos controladores de dispositivo são bloqueadas.

[0051] O arcanjo 380 também suporta a projeção por processo (acesso restrito) para anjos. Isto significa que os anjos, os quais são agentes iguais pelo fato de que o sistema irá executar qualquer anjo que o usuário solicitar (consistente com a política do usuário) e os quais não são parte do vetor de atestação, como definido abaixo (isto é, o arcanjo é na verdade parte da configuração da máquina), poderiam descarregar danos, invalidando a privacidade de lado esquerdo, intrometendo-se em aplicações mortais que eles não são supostos de se aplicarem, por exemplo. Portanto, é desejável que os anjos que sejam de forma forte limitados às aplicações particulares (mortais). Isto é de preferência feito por se permitir à um anjo somente afetar um mortal que o anjo inicia, ou, permitir à um anjo somente se aplicar à um mortal que se associa com um transformador declarado no manifesto do anjo, com a verificação de transformador feita pelo arcanjo e somente após as chamadas da aplicação mortal com o transformador do anjo para iniciá-la. Isto é desejável porque torna seguro e prático permitir à qualquer vendedor de aplicação escrever um anjo para sua aplicação e deixar qualquer usuário utilizá-lo, sem arriscar causar danos ou destruir a privacidade para tudo mais.

[0052] Assim, o arcanjo é tanto o agente que observa através do LHS como um agente que oferece serviços para outros anjos. Devido ao fato do arcanjo possuir o conhecimento mais detalhado das estruturas de processo LHS, o arcanjo irá provavelmente ser o que decide qual anjo pode se ligar à qual processo LHS. A restrição significa que um anjo (o qual não é parte do vetor de atestação do nexus) pode somente tocar processos que ele inicia, ou que chamam por ele para protegê-los. Isto impede os anjos de agirem ao acaso nos processos LHS. Esta divisão (o arcanjo obtém as capacidades de nível do OS e é vali-

dade assim como o nexus, os anjos obtém capacidades de nível de aplicação restritas e podem ser executados livremente como qualquer outro agente) é desejável.

[0053] Para uma modalidade, os anjos podem incluir o arcanjo (e por extensão o OS base LHS) em seus vetores de atestação. Um vetor de atestação é uma lista de transformadores de componentes relevantes de segurança que estabelecem a configuração relevante de segurança de uma entidade. Por exemplo, o transformador para um agente pode incluir a própria máquina ou placa mãe, o nexus e o próprio agente, junto com outras informações. Esta pilha de números é um indicador forte, confiável, do que o agente é e em qual ambiente o agente está executando. Ele permite à outra entidade confiar que ela está lidando ou não com o "agente real". A pilha de vetores de atestação (então o transformador do agente não é parte do vetor para o nexus, mas o transformador do nexus é parte do transformador para o agente). Então, quando o vetor de atestação de alguma coisa é incluído em outra coisa, isto significa que todos eles estão sendo limitados à uma configuração de segurança que pode ser reconhecida. Uma propriedade de uma atestação é que ela fortemente identifica a configuração relevante de segurança de um sistema.

[0054] Colocado de outra maneira, um vetor de atestação é uma lista de valores de transformador que define uma identidade de software do RHS. De preferência, o software carregado no RHS é transformado antes de ser carregado e o próprio processo é bem isolado de modo que ele não possa alterar-se. Isto é um processo indutivo: o hardware assina o transformador do nexus (atesta para o transformador do nexus) e o nexus por sua vez atesta o agente. Desta maneira, uma parte externa pode validar estes transformadores junto à uma lista conhecida para determinar se esta parte exterior aprova o software executando no sistema. O anjo e o arcanjo, devido ao fato de estarem

executando no RHS, possuem identidades de código bem definidas. Por esta razão, estas identidades de código podem ser listadas no vetor de atestação, o qual descreve o ambiente no qual o código LHS está executando. Devido ao fato do anjo não poder controlar totalmente a execução do código LHS, esta declaração de identidade de código não é tão forte como uma declaração de identidade de código de um agente RHS, mas isto realmente significa que a dada parte de código LHS está executando sob as restrições do anjo, do arcanjo e do nexus que possuem fortes identidades de código.

[0055] As modalidades de um arcanjo podem expor algum conjunto de APIs para os anjos para proporcionar suporte para algumas funções e / ou características dos anjos. Por exemplo, para qualquer operação de memória, o arcanjo irá desejavelmente intermediar. Um anjo pode desejar examinar o código da aplicação coberta no endereço virtual VA=100. Entretanto, pode não ser conhecido para qual endereço físico ele mapeia. O nexus não conhece a cerca de tais estruturas. Portanto, ao invés disso, o arcanjo (o qual conhece como o OS LHS funciona) utiliza os serviços de nexus básicos (que somente os arcanjos podem chamar) para ler a memória do núcleo LHS relevante. O arcanjo utiliza dados a partir da memória OS LHS para calcular mapeamentos corretos para a memória de aplicação LHS. O anjo é então informado a cerca de qual endereço de aplicação coberta corresponde ao endereço do anjo e o anjo pode então inspecionar estes conteúdos e continuar o processamento. Em resumo, para anjos de limite de processos (isto é, anjos que somente aplicam-se à processos autorizados ao invés de percorrer através do estado LHS aleatoriamente), é desejável que o arcanjo interprete as estruturas de dados LHS.

[0056] Uma função ilustrativa adicional inclui proporcionar um canal IPC seguro que irá somente permitir à aplicação LHS e ao anjo RHS verem os dados. O núcleo LHS normalmente estaria apto a ver

todas as páginas que atravessam um canal IPC entre o LHS e o RHS, mas se estas páginas somente puderem ser acessadas sob o olho observador do arcanjo, então é proporcionada a alta garantia de que somente o processo em questão (o processo controlado pelo dado anjo) pode ver os dados no canal. Outra função ilustrativa fornece ao anjo a habilidade de controlar quais módulos (por exemplo, DLLs) e quais versões destes módulos podem ser carregadas no espaço de processo de um dado processo.

[0057] Como uma entidade confiável, o arcanjo 380 possui acesso à memória associada com o LHS e é notificado sempre que algo acontece no LHS. O arcanjo 380 é pré-programado com um corpo de conhecimento que ele utiliza para detectar inconsistências para determinar se qualquer ação deve ser executada no interesse de segurança ou proteção. Por exemplo, o arcanjo 380 pode capturar certos conjuntos de eventos LHS. Estes podem ser eventos que são permitidos pelo LHS e que não são impedidos pelo nexus 351 ou pelo ambiente de confiança que ele gerencia. Por exemplo, o arcanjo 380 pode detectar mapeamentos impróprios no LHS (os quais o nexus 351 de outra maneira permitiria), o que indica uma possível questão de ataque ou de segurança. O arcanjo 380 também pode executar uma verificação de consistência.

[0058] Para a modalidade apresentada na Figura 3, cada anjo está limitado ou de outro modo supervisionado pelo arcanjo 380 e pelo nexus 351 (bloco 420). O arcanjo 380 impõem a ligação entre um anjo e seu código LHS associado, o que limita a habilidade dos anjos de afetarem a privacidade e a segurança, por exemplo, no LHS.

[0059] É desejável que o comportamento dos anjos seja restrito à afetar somente os processos com os quais eles estão supostos de estarem ligados, porque o nexus 351 e o arcanjo 380 irão executar qualquer anjo que o usuário direcioná-los a executar, de acordo com a polí-

tica do usuário. O arcanjo possui capacidades em condições iguais do nexus e será examinado minuciosamente no mesmo nível. Para os anjos, como para qualquer outro agente, o nexus irá executar sempre que o usuário os solicitar. Então, enquanto que o nexus e os arcanjos são restritos, os anjos comuns (como os agentes) não são (apesar do usuário poder estabelecer políticas que dizem para o nexus executar ou não executar agentes ou anjos assinados por um avaliador particular, por exemplo).

[0060] É desejável que os anjos sejam restritos. Por exemplo, um anjo com um bloco de assinatura que diz "anjo para um primeiro programa" não deve ser permitido de utilizar a memória OS base LHS ou utilizar a memória de outros programas. Permitir isto violaria vários direitos do usuário e faria dos anjos perigosos ao invés de úteis. Então, o arcanjo certifica-se que os anjos somente obtenham acesso aos programas LHS que eles são supostos de estarem aptos a acessarem.

[0061] Um agente confiável de preferência não possui mais capacidade do que qualquer programa LHS. Em particular, um agente confiável não pode olhar dentro do OS LHS nem controlar nem editar o estado de configuração do OS LHS. Ao invés disso, os anjos de preferência são somente permitidos de inspecionar ou modificar a memória dos mortais para os quais eles se aplicam. Adicionalmente, em algumas modalidades, o arcanjo poderia não mais permitir que um anjo altere o código do mortal, restringindo o anjo a ler qualquer coisa no espaço de endereço do modo do usuário do seu mortal e permitindo-o escrever no espaço de memória não compartilhado de leitura-gravação do mortal. Entretanto, alguns mecanismos requerem que uma chamada do mortal pelo anjo seja permitida de retornar não para o ponto de chamada, mas ao invés disso, para um ponto de retorno calculado. Isto permite ao anjo forçar alguns eventos para iniciarem em endereços corretos conhecidos no mortal – uma forte maneira de se combater

ataques de rede de acrobatas baseados em pilhas corrompidas alterando os endereços de retorno.

[0062] Um anjo somente pode monitorar sua entidade ou grupo de entidades associadas (bloco 430) e não é mais confiável do que qualquer outro agente. Um anjo não pode monitorar ou de outro modo olhar nas entidades não associadas. Em particular, um anjo possui uma ou mais das seguintes propriedades:

a. O anjo pode monitorar a memória do modo de usuário somente do processo ou processos com os quais ele está ligado (isto é, o mortal) (não um código RHS de capacidade normalmente proporcionado – veja acima).

b. Somente o anjo pode ver a memória de modo núcleo do OS LHS com a qual ele está ligado.

c. O anjo pode ser aplicado somente para estes processos LHS que chamam e solicitam ao mesmo ou é somente aplicado aos processos LHS que ele inicia.

d. O anjo pode ser restrito por imposição declarativa. Por exemplo, o nexus e / ou o arcanjo podem restringir o anjo para projetar-se somente nestes processos que contém executáveis que se associam com os executáveis declarados no manifesto para o anjo. Assim, por exemplo, um anjo para "hackertool" não pode se projetar em uma aplicação LHS por acidente ou por malícia sem alguém alterar o manifesto para o anjo. Tal alteração do manifesto seria óbvia para uma ferramenta de política.

[0063] O arcanjo 380 pode impor as restrições acima (blocos 440 e 450). Para este propósito, pode ser dado para o arcanjo acesso extensivo ao LHS e, caso este em que ele está sujeito à um nível de exame minucioso similar à este do nexus (isto é, exame minucioso intenso). Por exemplo, o arcanjo possui poder sobre o OS LHS e assim sobre qualquer coisa que roda no LHS. Posto de outro modo, o arcanjo pode

ler qualquer memória LHS, mas não possui capacidades especiais RHS, tal como acesso à memória de núcleo RHS ou habilidade de ver dentro dos processos de outros agentes, ou restringir, aumentar, modificar, etc. o nexus ou outros agentes RHS. Um anjo pode somente ler o espaço de endereço do programa para o qual ele se aplica (isto é, os anjos possuem capacidades especiais, as quais aplicam-se somente aos mortais aos quais eles se aplicam). O arcanjo também pode ler toda a memória LHS (etapa 440), enquanto oferecendo processar serviços específicos de modo que os anjos somente possam ver dentro do espaço de endereço dos programas que eles monitoram e protegem.

[0064] Um anjo pode "projetar" sua proteção pelo menos dos seguintes modos (etapas 430 e 450):

a. Ele pode bloquear ou marcar como somente-leitura vários elementos da memória, possivelmente em coordenação com o comportamento de proteção, para impedir certas alterações (por exemplo, ataques de vírus) na proteção.

b. Ele pode executar algumas operações chave para a proteção, dentro de seu espaço confiável.

c. Ele pode insistir nas proteções específicas da proteção, tal como limitar quais alterações de configuração podem ser feitas, ou permitir tais alterações de serem feitas se aprovadas por um humano autorizado utilizando um mecanismo de entrada seguro.

d. Ele pode examinar a memória de proteção e relatar em intervalos desejados procurando por erros de inconsistência, corrupções e assim por diante e notificar o usuário ou parar a proteção antes do dano adicional ou ação não intencional / não autorizada ocorrer.

e. Ele pode liberar dados lacrados / criptografados para a proteção somente quando preciso para minimizar a quantidade de tais dados que podem ser atacados em qualquer hora.

1. Ele pode utilizar o armazenamento lacrado para manter

segredos lacrados para o LHS (ou uma aplicação LHS) e recusar fornecer estes segredos para qualquer LHS (ou aplicação LHS) que não possui um transformador associando-se com o dono do segredo ou listado como permitido pelo dono do segredo.

f. Dada a API apropriada, ele pode alterar o estado de execução da proteção; isto é, ele pode direcionar processos para pontos de execução conhecidos, direcionar o fluxo de controle na aplicação alvo ou executar a computação e execução de ramificação para a aplicação alvo. Ele também pode editar o estado de configuração, o estado de inicialização ou coisa parecida, para forçar as coisas em modos aceitáveis para operação segura / correta da proteção.

g. um anjo pode chamar o arcanjo e solicitar ao arcanjo para executar a prevenção, proteção, descoberta ou reação a favor da proteção.

h. Um anjo pode extrair (por chamada ou por inspeção de memória, por exemplo) dados de saída a partir da aplicação, validar tais dados (por exemplo, executar soma de verificação, etc.) e então apresentar estes dados utilizando o hardware de saída seguro.

[0065] Parte da funcionalidade da entidade ou aplicação pode ser movida para o anjo. Do mesmo modo, parte da funcionalidade do núcleo LHS pode ser movida para o arcanjo. Um criador da aplicação pode implementar algumas das funções da aplicação no anjo. Apesar de que isto aumentaria a carga RHS, permitiria às funções transferidas serem executadas no ambiente confiável. De forma similar, uma parte do OS LHS 301 pode ser movida no arcanjo 380.

[0066] Um anjo pode ser carregado ou invocado de várias maneiras. Um programa LHS, tal como a aplicação 305 pode chamar pelo seu anjo 390. Desta maneira, por exemplo, quando da inicialização de uma aplicação, o anjo correspondente é carregado. Alternativamente, um anjo pode ser carregado a partir do RHS e o anjo então invoca o

processo ou aplicação LHS correspondente. O anjo utiliza o arcanjo para chamar através do LHS e requisitar que a aplicação seja iniciada. O arcanjo então liga o agente com a aplicação. Para uma modalidade, as APIs que o nexus e o arcanjo oferecem para o anjo da aplicação deixam-no ver somente o processo que ele cria e talvez os filhos do mesmo.

[0067] Como outra alternativa, o programa LHS, pode ser invocado por manifesto e então desviado para o RHS que inicia o anjo, o qual chama de volta para o LHS para iniciar o processo ou aplicação LHS correspondente. Tipicamente, um programa LHS é iniciado por se denominar o arquivo que contém o mesmo (uma API que é, por exemplo, "run c:\somedir\someotherdir\someprogram.exe"). Para o código RHS (um agente ou anjo), ele é iniciado por se denominar um manifesto e o manifesto denomina o binário. Isto é independente de localização. Além disso, os manifestos são tipicamente assinados e certificados, por exemplo, de modo que eles são muito mais difíceis se fraudar. Assim, um mecanismo ilustrativo seria para apresentar um manifesto de esquerda / direita combinado para o RHS (nexus), o qual iniciaria tanto a aplicação LHS como o anjo relacionado e os ligaria. Além do mais, o anjo pode ser utilizado para iniciar a aplicação a partir do LHS ou do RHS.

[0068] Em uma modalidade da invenção, o arcanjo pode confirmar que a imagem de código carregada inicialmente do processo LHS se associa com uma imagem de código alvo declarada associada com o anjo. A imagem de código alvo declarada pode ser proporcionada através do manifesto do anjo. Isto impede o código que reivindica ser um anjo para uma aplicação particular de iniciar outra aplicação no lugar, o que proporciona segurança adicional contra ataque.

[0069] De acordo com algumas modalidades da invenção, um anjo é impedido de editar a imagem de código da aplicação ou processo

LHS com o qual ele está associado. O anjo pode ler / gravar dados, mas ele somente pode ler código.

[0070] Estas e políticas similares podem ser empregadas para impedir os anjos de executarem sem supervisão ou restrições sobre o LHS e os anjos trapaceiros são impedidos de fraude utilizando os programas e aplicações LHS.

[0071] Em adição aos mecanismos de iniciação descritos acima, existem outras maneiras para ter certeza de que o anjo correto está ligado à aplicação LHS (ou RHS) correta e permanece ligado à mesma. Uma aplicação executando pode ser alterada por um atacante antes de fazer uma chamada para o seu agente ou um vírus LHS pode interceptar e permutar sua chamada para almejar algum outro anjo.

[0072] As modalidades da presente invenção podem endereçar isto por processar chamadas a partir de uma aplicação para seu anjo através de uma autoridade confiável como o arcanjo ou o nexus. Por exemplo, o arcanjo pode transformar a aplicação LHS que chama e comparar a transformação com uma lista de transformadores "aprovados" associados com o anjo RHS. Se eles realmente não se associarem, devido ao fato da aplicação LHS ter sido permutada ou devido ao fato da chamada ter sido modificada para almejar um anjo diferente, a chamada falha e o sistema pode notificar o usuário e / ou executar qualquer outra série de ações.

[0073] A política do sistema pode ser utilizada para especificar quais anjos podem se ligar com quais aplicações LHS. Utilizar um mecanismo de política forte proporciona um mecanismo difícil de fraudar, difícil de inicializar de maneira errada para estabelecer tais dependências.

[0074] Em algumas modalidades, um anjo de preferência possui vários níveis de inspeção programáveis ou ajustáveis para almejar ameaças à aplicação associada. A sensibilidade do anjo em relação à

uma ameaça ou ataque percebidos pode ser ajustada.

[0075] Em adição à proporcionar projeção (defesa, proteção, conselho, por exemplo) para o OS ou aplicações LHS, um anjo também pode ser aplicado para um agente executando no ambiente de computação confiável. Em tal caso, um agente alvo (normalmente uma entidade paranoide) confia no anjo que se liga à ele. Isto permite à um processo observador externo interceptar várias falhas e façanhas no agente alvo. O anjo pode impor invariáveis de segurança oposto à examinar os erros de segurança (por exemplo, como em uma tecnologia antivírus convencional) e o uso da forte separação e proteção de processo que um nexus proporciona.

[0076] Para uma modalidade, o agente é uma máquina virtual, apresentando uma "cópia duplicata efetivamente idêntica" de alguma máquina real, dentro da qual uma imagem do OS foi lançada. Um ambiente confiável pode permitir à um agente acessar a memória de processo da máquina virtual. O agente que acessa pode monitorar a memória do processo para proteger a máquina virtual contra ataques a partir da imagem que ele contém. Um ambiente confiável pode permitir à um anjo proteger a imagem do OS na máquina virtual e permitir aos anjos protegerem aplicações na máquina virtual. É contemplado que os mesmos mecanismos normalmente aplicados às aplicações LHS sejam aplicados ao invés disso junto ao ambiente de máquina virtual.

[0077] Para uma modalidade da invenção, o nexus proporciona ao arcanjo uma API para inspeção e alteração da memória (pelo menos). A API suporta a captura e reação à tentativas de alterar as estruturas de controle que facilitam a projeção. Por exemplo, na arquitetura x86, a proteção para as estruturas de controle tal como GDT, LDT, IDT, registradores de depuração, TR, etc. pode ser proporcionada através de uma API. O GDT se refere à tabela descritora global e o LDT refere-se à tabela descritora local. Bloquear a GDTR (registro de tabela descrito-

ra global) para os ataques que dependem da deformação do significado dos endereços virtuais de modo a permitir saltos para locais que o atacante normalmente não poderia saltar. A IDT é a tabela de despacho de interrupção, a qual controla o roteamento das interrupções. A localização do IDT é indicada pelo IDTR (registro da tabela de despacho de interrupção). Bloquear o IDTR torna a projeção mais poderosa por parar os ataques nos quais o atacante utiliza a IDT e uma interrupção anunciada para forçar uma ramificação para o código que eles de outro modo não alcançariam.

[0078] É desejável que o Ambiente Confiável (isto é, o RHS) e o Ambiente Aberto (isto é, o LHS) estejam conectados de alguma maneira. A conexão permite ao Ambiente Confiável examinar o estado e ser informado de eventos no Ambiente Aberto. Os ensinamentos aqui trabalham para estruturas incluindo, mas de modo algum limitadas às, estruturas abaixo:

1. O RHS e o LHS estão na mesma máquina e o RHS pode diretamente examinar a memória LHS (enquanto que o LHS não pode examinar a memória RHS sem permissão).

2. O RHS e o LHS estão em processadores diferentes, possivelmente com memórias diferentes, porém um barramento, rede, porta ou outra interconexão permite ao RHS ver dentro da memória LHS. Por exemplo, um processador de serviço ARM poderia executar uma pilha totalmente confiável e a pilha confiável poderia estar apta a inspecionar a memória principal de um sistema MP x86. Por exemplo, alguém poderia possuir uma máquina com processadores principais x86 e um ARM ou powerPC como um processador de serviço e utilizar os mecanismos da presente invenção para permitir ao processador de serviço observar pelo software nos processadores principais.

3. Se o RHS pode receber a notificação de eventos LHS (por exemplo, alterações de mapas) mas não os altera ou os impede,

ou não pode ver dentro da memória LHS, alguma parte de projeção (por exemplo, uma parte fraca) ainda é possível.

4. O RHS pode inspecionar a memória LHS à vontade, pode controlar (isto é, impedir ou alterar) as edições LHS para o mapeamento de memória LHS e as estruturas de tradução de endereço, controlar para onde o vetor de despacho de interrupção aponta (porém não precisam controlar o controlador de interrupção, apesar de que se tal controle for oferecido, existe aperfeiçoamento nisto). É contemplado que determinar uma lista de estados / eventos que o RHS de forma desejável está apto a controlar totalmente para suportar projeção forte é uma tarefa a ser feita para cada arquitetura de processador e um trabalhador com conhecimento na técnica irá entender que a lista é diferente para arquiteturas diferentes.

5. Em uma modalidade, as alterações do registro e estabelecimento TR x86 dos registradores de depuração do hardware também são controláveis pelo RHS.

[0079] No hardware da técnica anterior, o ambiente confiável não é garantido de executar porque ele pode depender do hardware de interrupção comum, da tabela de despacho de interrupção e assim por diante.

[0080] No hardware listado acima, estando apto a controlar o IDT (em um x86, ou o equivalente em outro lugar) permite ao RHS garantir que alguma interrupção de sua escolha irá sempre executar o código que chama o RHS.

[0081] Entretanto, um atacante ou erro LHS poderia corromper o controlador de interrupção, desligar a interrupção e assim por diante. É contemplado que o ATC (controle de tradução de endereço) é utilizado para garantir que o RHS roda frequentemente. Se o RHS estiver utilizando o ATC, ele pode modificar o ATC para incrementar um contador. O contador é estabelecido para algum valor sempre que o RHS pro-

grama a execução do arcanjo. Se o contador alcançar zero, o ATC então sabe que o arcanjo não rodou por "muito tempo" e chama um ponto de entrada do nexus que de forma forçada roda o arcanjo. Esta técnica não garante que o arcanjo execute em qualquer momento particular, mas garante que ele irá executar após uma série de operações de edição da memória LHS. Assim, um LHS que é ativo irá eventualmente ter que deixar o arcanjo executar.

[0082] Se o RHS pode bloquear o IDT e o sistema possui uma fonte confiável de NMIs (interrupções que não podem ser mascaradas), então o RHS pode forçar o manipulador NMI a chamar certo.

[0083] Em uma modalidade ilustrativa, o hardware possui um temporizador que força uma interrupção para a RHS após vários momentos.

[0084] A presente invenção proporciona mecanismos que permitem que a confiabilidade de um ambiente de computação seja projetada em um segundo ambiente de computação. Dois ou mais sistemas operacionais que rodam em uma única máquina são exemplos de "ambientes" que podem precisar interagir um com o outro em uma única máquina, apesar de que será entendido que a invenção não está limitada à um sistema operacional tradicional. Além do mais, pelo menos algumas das técnicas descritas aqui dentro podem ser utilizadas, no caso geral, para projetar a confiabilidade a partir de qualquer tipo de entidade executável (por exemplo, qualquer parte de software) até qualquer outro tipo de entidade.

[0085] No caso onde duas entidades existem lado à lado em uma única máquina e precisam se interagir uma com a outra, a interação pode tomar várias formas. Por exemplo, as duas entidades podem precisar comunicar os dados de um lado para outro uma com a outra. No caso onde as entidades são sistemas operacionais (ou certos outros tipos de ambientes de execução, tal como mecanismos de script que

executam scripts em uma máquina virtual), as entidades podem precisar interagir umas com as outras de certas outras maneiras - por exemplo, compartilhando memória, compartilhando tempo em um processador, compartilhando recursos e manipulando interrupções. A invenção proporciona técnicas pelas quais duas entidades podem empenhar-se nestes tipos de interações uma com a outra, enquanto permitindo à uma entidade projetar sua confiabilidade na outra entidade.

[0086] As modalidades descritas acima se focalizam na memória como o recurso monitorado, mas a invenção não está limitada à isto. Se os monitores de segurança estiverem disponíveis para outros recursos além da memória, um agente de monitoramento base (por exemplo, arcanjo) pode empregar tais monitores como delegados confiáveis para estender sua esfera de confiança. Por exemplo, se um NIC seguro estiver disponível, o agente de monitoramento base pode utilizá-lo para impedir enviar pacotes com certos cabeçalhos. Em geral, tal delegado confiável somente precisa entender uma invariável de medida, por exemplo, cabeçalhos associando <regexp> e de forma confiável alertar o agente de monitoramento quanto às alterações da invariável.

[0087] É observado que os exemplos precedentes foram proporcionados meramente para o propósito de explicação e não são de forma alguma para serem construídos como limitação da presente invenção. Enquanto que a invenção foi descrita com referência à várias modalidades, é entendido que as palavras que foram utilizadas aqui dentro são palavras de descrição e ilustração, ao invés de palavras de limitações. Adicionalmente, apesar da invenção ter sido descrita aqui dentro com referência à meios, materiais e modalidades particulares, a invenção não é pretendida para estar limitada aos particulares revelados aqui dentro; ao invés disso, a invenção se estende para todas as estruturas, métodos e usuários de funcionalidade equivalente, tal como es-

tão dentro do escopo das reivindicações anexas. Estes com conhecimento na técnica, possuindo o benefício dos ensinamentos desta especificação, podem efetuar numerosas modificações na mesma e as alterações podem ser feitas sem se afastar do escopo e espírito da invenção em seus aspectos.

REIVINDICAÇÕES

1. Sistema adaptado para executar sistemas operacionais plurais (301, 501) em um único processador, o sistema compreendendo:

um ambiente operacional não confiável compreendendo pelo menos um primeiro sistema operacional (301);

um ambiente operacional confiável compreendendo um segundo sistema operacional (351) para executar uma pluralidade de agentes de monitoramento (390, 395), em que cada agente de monitoramento é associado a uma aplicação (305, 310), no ambiente operacional não confiável, e cada agente de monitoramento monitora (430) sua aplicação associada,

o ambiente operacional confiável compreendendo ainda um agente de monitoramento base (380), o dito agente de monitoramento base sendo associado (410) com o sistema operacional (301) do ambiente operacional não confiável e supervisionando o pelo menos um agente de monitoramento (390, 395),

CARACTERIZADO pelo fato de que o agente de monitoramento base (380) não permite o agente de monitoramento a alterar o código da aplicação associada com o agente de monitoramento por restrição do agente de monitoramento a ler qualquer coisa no espaço de endereço de modo de usuário de sua aplicação associada e permitindo o agente de monitoramento a escrever no espaço de memória não compartilhado de leitura-gravação de sua aplicação associada.

2. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que cada agente de monitoramento (390, 395) é adaptado para compreender uma parte da aplicação (305, 310) que é associada com o mesmo.

3. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que cada agente de monitoramento

possui um nível ajustável de inspeção para almejar ameaças para a aplicação associada.

4. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que cada agente de monitoramento (390, 395) é adaptado para receber entrada segura e transferir a entrada segura para a aplicação associada.

5. Sistema, de acordo com a reivindicação 2, **CARACTERIZADO** por adicionalmente compreender outro agente de monitoramento (390, 395) é adaptado para executar no ambiente confiável, em que os agentes de monitoramento estão em comunicação um com os outros.

6. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o agente de monitoramento base (380) é adaptado para detectar inconsistências no ambiente não confiável.

7. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que pelo menos um dos agentes de monitoramento base (380) é adaptado para aprovar ou desaprovar um evento do ambiente operacional não confiável.

8. Sistema, de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que pelo menos um agente de monitoramento base (380) compreende uma entrada segura para receber entrada, o agente de monitoramento base aprovando ou desaprovando baseado na entrada recebida.

9. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que pelo menos um dos agentes de monitoramento base (380) é adaptado a recusar-se a permitir alterações no ambiente operacional não confiável sem receber aprovação via uma entrada segura.

10. Sistema, de acordo com a reivindicação 1,

CARACTERIZADO pelo fato de que pelo menos um dos agentes de monitoramento base (380) é adaptado a recusar-se a permitir alterações no ambiente operacional não confiável a menos que as alterações sejam descritas por um pacote que é assinado por uma parte aprovada.

11. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o agente de monitoramento utiliza o armazenamento lacrado para manter um segredo para um sistema operacional ou uma aplicação residindo no ambiente operacional não confiável.

12. Sistema, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390, 395) é adaptado a recusar-se a revelar o segredo para o sistema operacional ou a aplicação a menos que o sistema operacional ou a aplicação possuam um transformador que se associa com o dono do segredo.

13. Sistema, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390, 395) é adaptado a recusar-se a revelar o segredo para o sistema operacional ou a aplicação a menos que o sistema operacional ou a aplicação estejam em uma lista de transformadores que podem ler o segredo.

14. Sistema, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390, 395) é adaptado a utilizar um teste predeterminado para determinar se uma entidade legítima está requisitando o segredo.

15. Sistema, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que o teste predeterminado inclui examinar as pilhas da entidade e garantir que as pilhas possuem conteúdos de pilhas legais.

16. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390, 395) é adaptado a editar um estado do ambiente operacional não confiável para torná-lo seguro ou de outro modo aceitável.

17. Sistema, de acordo com a reivindicação 16, **CARACTERIZADO** pelo fato de que o estado compreende uma configuração inicial ou uma opção de relatório de erros.

18. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o agente de monitoramento base (380) é adaptado a zerar a memória física que não pertence à configuração boa conhecida do ambiente operacional não confiável ou ao ambiente operacional confiável.

19. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o ambiente operacional não confiável compreende um sistema básico de entrada / saída (BIOS), firmware ou carregador,

20. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o sistema operacional de alta segurança (351) é adaptado a executar o agente de monitoramento base (380) no momento da partida.

21. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um contador no ambiente operacional confiável, o contador utilizado para determinar se o agente de monitoramento base (380) deve ser executado.

22. Sistema, de acordo com a reivindicação 21, **CARACTERIZADO** pelo fato de que o contador é adaptado a contar o número de operações de edição de memória não confiáveis.

23. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o ambiente operacional confiável executa uma primeira arquitetura de processador e o ambiente opera-

cional não confiável executa em uma segunda arquitetura de processador, adicionalmente compreendendo um agente de monitoramento base (380) executando no primeiro processador.

24. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o ambiente operacional confiável e o ambiente operacional não confiável executam no mesmo processador, adicionalmente compreendendo um agente de monitoramento base (380) executando no ambiente operacional confiável.

25. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o ambiente operacional confiável executa em um primeiro processador e o ambiente operacional não confiável executa em um segundo processador, o primeiro e o segundo processadores sendo capazes de executarem em um modo confiável ou em um modo não confiável.

26. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um sistema operacional de alta segurança (351) e um agente de monitoramento base (380) residindo no ambiente operacional confiável, o agente de monitoramento base sendo confinado, ligado ou compilado dentro do sistema operacional de alta segurança (351).

27. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um sistema operacional de alta segurança (351) e um agente de monitoramento base (380) residindo no ambiente operacional confiável, em que o agente de monitoramento base (380) é um processador de modo de usuário executando no sistema operacional de alta segurança (351).

28. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um agente de monitoramento base (380) residindo no ambiente operacional confiável e desenvolvido por e com e no mesmo ou em um ambiente de constru-

ção relacionado como um sistema operacional do ambiente operacional não confiável.

29. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um agente de monitoramento base (380) residindo no ambiente operacional confiável, o agente de monitoramento base sendo parte de uma base de computação confiável para avaliação de segurança.

30. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** por adicionalmente compreender um agente de monitoramento base (380), uma primeira parte do agente de monitoramento base (380) residindo no ambiente operacional confiável e uma segunda parte do agente de monitoramento base (380) residindo em uma máquina fisicamente remota, a primeira e segunda partes conectadas por uma ligação segura.

31. Método para monitorar um ambiente operacional não confiável de um sistema adaptado para executar sistemas operacionais plurais (301, 351) em um único processador, o método compreendendo as etapas de:

proporcionar um ambiente operacional não confiável compreendendo pelo menos um primeiro sistema operacional (301);

proporcionar um ambiente operacional confiável compreendendo um segundo sistema operacional (351) para executar uma pluralidade de agentes de monitoramento (390, 395), em que cada agente de monitoramento é associado a uma aplicação (305, 310) no ambiente operacional não confiável, e cada agente de monitoramento monitora (430) sua aplicação associada; e

proporcionar ainda no ambiente operacional confiável um agente de monitoramento base (380), o dito agente de monitoramento base sendo associado (410) com o sistema operacional (301) do ambiente operacional não confiável e supervisionando o pelo menos um agente de

monitoramento (390, 395),

CARACTERIZADO pelo fato de que o agente de monitoramento base (380) não permite o agente de monitoramento a alterar o código da aplicação associada com o agente de monitoramento por restrição do agente de monitoramento a ler qualquer coisa no espaço de endereço de modo de usuário de sua aplicação associada e permitindo o agente de monitoramento a escrever no espaço de memória não compartilhado de leitura-gravação de sua aplicação associada.

32. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender associar uma aplicação (305, 310) com um dos agentes de monitoramento e transferir uma parte da aplicação para o agente de monitoramento de modo que a parte resida no ambiente operacional confiável.

33. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender associar (400) uma aplicação (305, 310) com o agente de monitoramento (390, 395) e ajustar um nível de inspeção no agente de monitoramento para almejar ameaças para a aplicação associada.

34. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender associar (400) uma aplicação (305, 310) com o agente de monitoramento (390, 395) e receber entrada segura no agente de monitoramento e transferir a entrada segura para a aplicação.

35. Método, de acordo com a reivindicação 31, **CARACTERIZADO** pelo fato de que os agentes de monitoramento (390, 395) estão em comunicação uns com os outros.

36. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender o agente de monitoramento base (380) aprovando ou desaprovando um evento do ambiente operacional não confiável.

37. Método, de acordo com a reivindicação 36, **CARACTERIZADO** por adicionalmente compreender o agente de monitoramento base (380) recebendo entrada a partir de uma entrada segura.

38. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender o agente de monitoramento base (380) recusando-se a permitir alterações no ambiente operacional não confiável sem receber aprovação via uma entrada segura.

39. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender o agente de monitoramento base (380) recusando-se a permitir alterações no ambiente operacional não confiável a menos que as alterações sejam descritas por um pacote que é assinado por uma parte aprovada.

40. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender:

proporcionar uma pluralidade de agentes de monitoramento (390, 395) executando no ambiente operacional confiável; e

um dos agentes de monitoramento utilizando o armazenamento lacrado para manter um segredo para um sistema operacional (301) ou uma aplicação (305, 310) residindo no ambiente operacional não confiável.

41. Método, de acordo com a reivindicação 31, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390, 395) se recusa a revelar o segredo para o sistema operacional (301) ou aplicação (305, 310) a menos que o sistema operacional ou aplicação possuam um transformador que se associa com o dono do segredo.

42. Método, de acordo com a reivindicação 31, **CARACTERIZADO** pelo fato de que o agente de monitoramento (390,

395) se recusa a revelar o segredo para o sistema operacional (301) ou aplicação (305, 310) a menos que o sistema operacional ou aplicação estejam em uma lista de transformadores que podem ler o segredo.

43. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender utilizar um teste predeterminado para determinar se uma entidade legítima está requisitando o segredo.

44. Método, de acordo com a reivindicação 43, **CARACTERIZADO** pelo fato de que o teste predeterminado inclui examinar as pilhas da entidade e garantir que as pilhas possuem conteúdos de pilha legais.

45. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender:

proporcionar uma pluralidade de agentes de monitoramento (390, 395) executando no ambiente operacional confiável; e

um dos agentes de monitoramento (390, 395) editando um estado do ambiente operacional não confiável para torná-lo seguro ou de outro modo aceitável.

46. Método, de acordo com a reivindicação 45, **CARACTERIZADO** pelo fato de que o estado compreende uma configuração inicial ou uma opção de relatório de erros.

47. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender o agente de monitoramento base (380) zerando a memória física que não pertence à configuração boa conhecida do ambiente operacional não confiável ou do ambiente operacional confiável.

48. Método, de acordo com a reivindicação 31, **CARACTERIZADO** pelo fato de que ambiente operacional não confiável compreende um sistema básico de entrada / saída (BIOS), firmwa-

re ou carregador.

49. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender executar o agente de monitoramento base (380) no momento da partida via um sistema operacional de alta segurança.

50. Método, de acordo com a reivindicação 31, **CARACTERIZADO** por adicionalmente compreender determinar se o agente de monitoramento base (380) deve ser executado responsivo à um contador.

51. Método, de acordo com a reivindicação 50, **CARACTERIZADO** pelo fato de que o contador conta o número de operações de edição de memória não confiáveis.

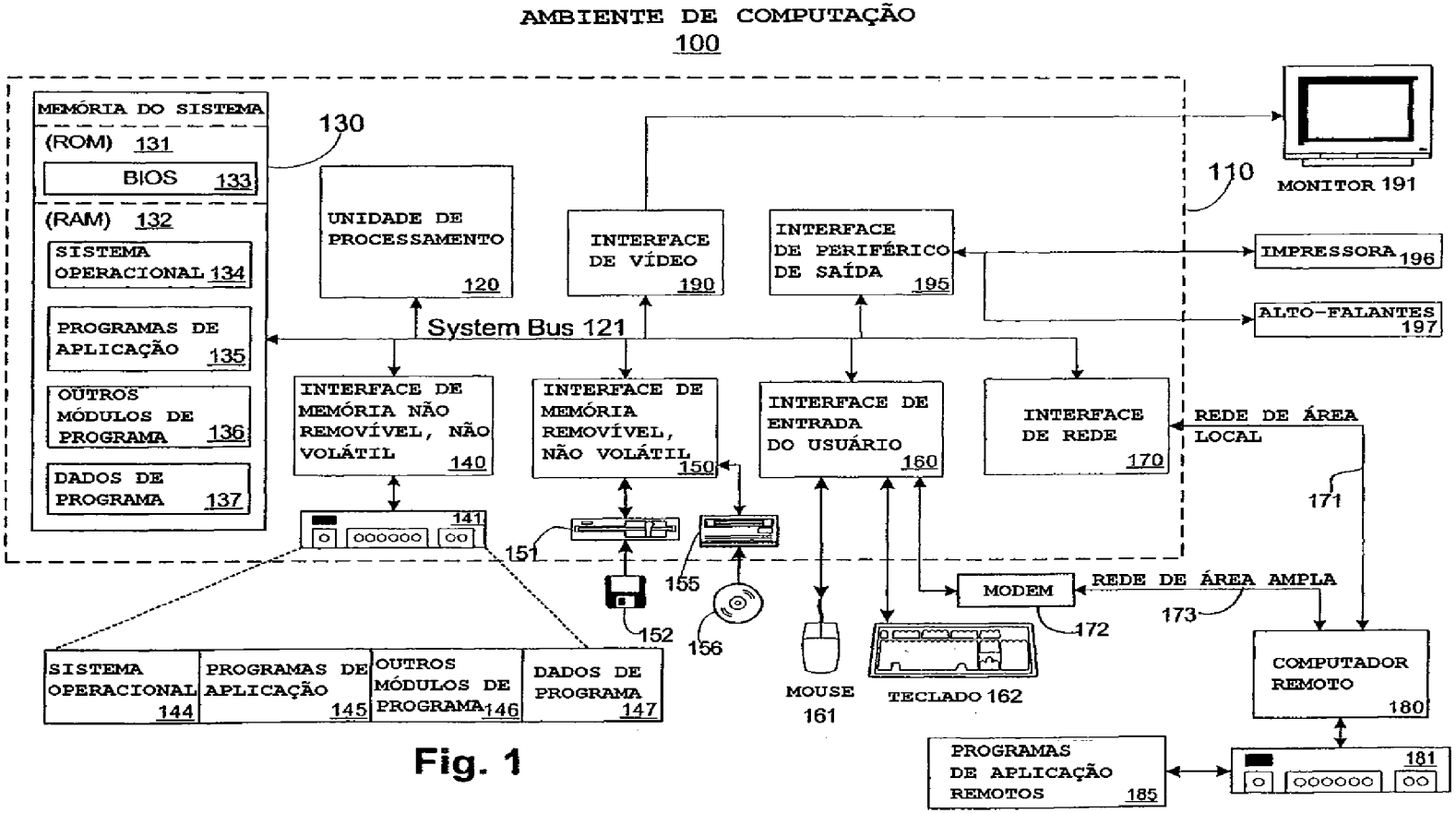
AMBIENTE DE COMPUTAÇÃO
100
MEMÓRIA DO SISTEMA
130
(ROM) 131
BIOS 133
(RAM) 132
SISTEMA OPERACIONAL 134
PROGRAMAS DE APLICAÇÃO 135
OUTROS MÓDULOS DE PROGRAMA 136
DADOS DE PROGRAMA 137
UNIDADE DE PROCESSAMENTO 120
INTERFACE DE VÍDEO 190
INTERFACE DE PERIFÉRICO DE SAÍDA 195
MONITOR 191
110
IMPRESSORA 196
ALTO-FALANTES 197
System Bus 121
INTERFACE DE MEMÓRIA NÃO REMOVÍVEL, NÃO VOLÁTIL 140
INTERFACE DE MEMÓRIA REMOVÍVEL, NÃO VOLÁTIL 150
INTERFACE DE ENTRADA DO USUÁRIO 160
INTERFACE DE REDE 170
REDE DE ÁREA LOCAL
171
141
151
155
152
156
MODEM
172
REDE DE ÁREA AMPLA
173
SISTEMA OPERACIONAL 144
PROGRAMAS DE APLICAÇÃO 145
OUTROS MÓDULOS DE PROGRAMA 146
DADOS DE PROGRAMA 147
MOUSE 161
TECLADO 162
COMPUTADOR REMOTO 180
181
PROGRAMAS DE APLICAÇÃO REMOTOS 185

Fig. 1

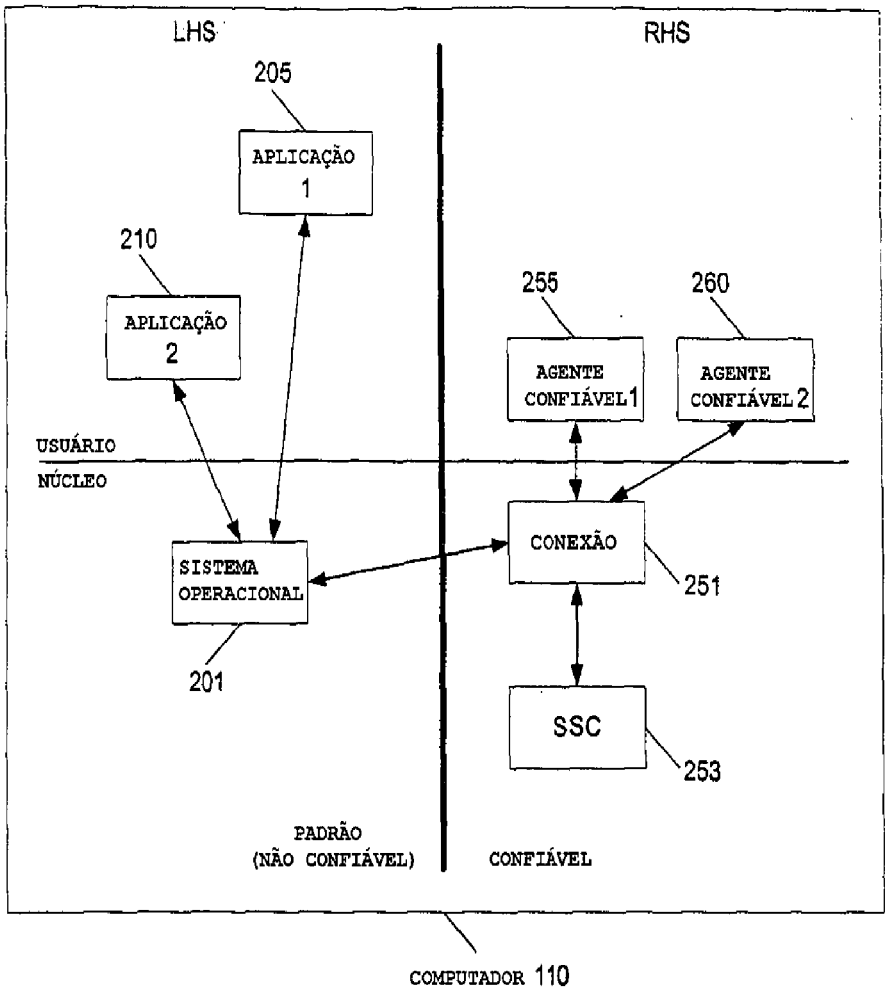

Fig. 2

(TÉCNICA ANTERIOR)

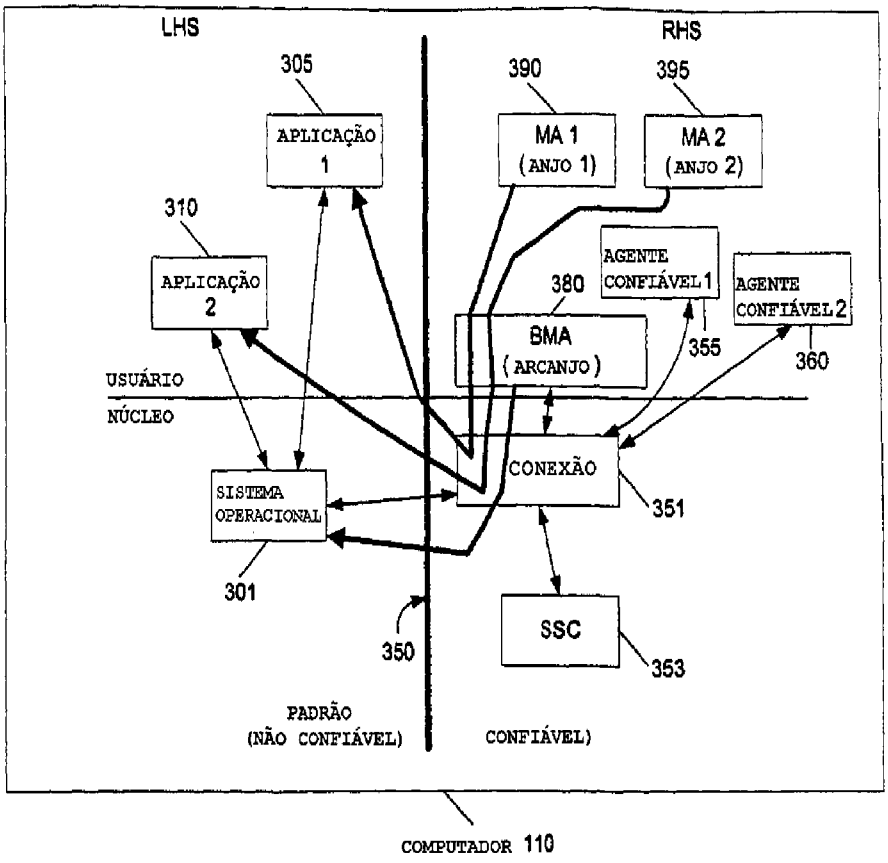
LHS
RHS
305
390
395
APLICAÇÃO 1
MA 1 (ANJO 1)
MA 2 (ANJO 2)
310
APLICAÇÃO 2
AGENTE CONFIÁVEL 1
AGENTE CONFIÁVEL 2
380
BMA (ARCANJO)
355
360
USUÁRIO
NÚCLEO
CONEXÃO
351
SISTEMA OPERACIONAL
301
350
SSC
353
PADRÃO (NÃO CONFIÁVEL)
CONFIÁVEL)
COMPUTADOR 110

Fig. 3

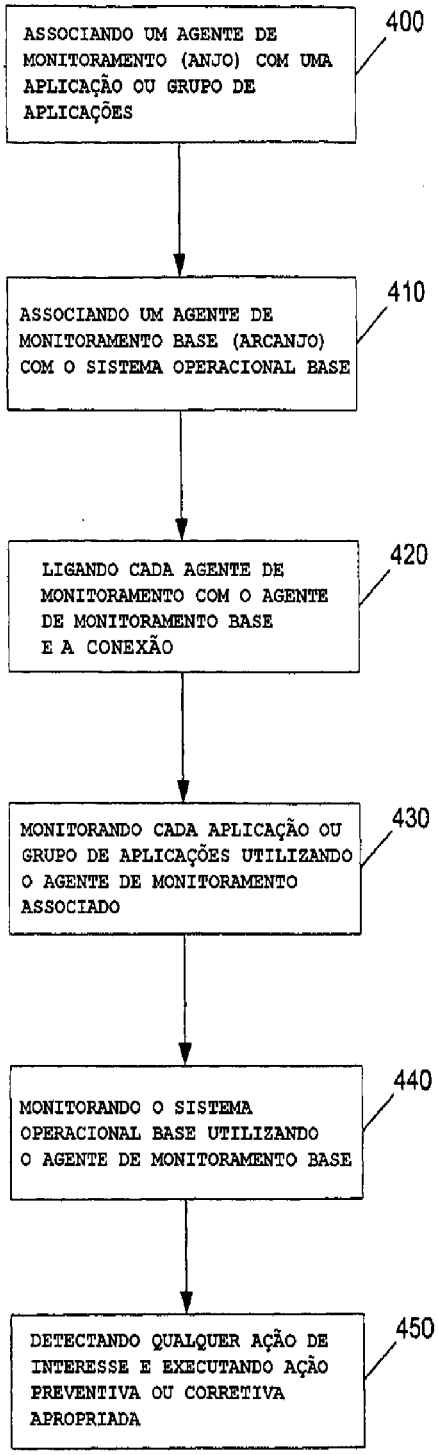
ASSOCIANDO UM AGENTE DE MONITORAMENTO (ANJO) COM UMA APLICAÇÃO OU GRUPO DE APLICAÇÕES
400
ASSOCIANDO UM AGENTE DE MONITORAMENTO BASE (ARCANJO) COM O SISTEMA OPERACIONAL BASE
410
LIGANDO CADA AGENTE DE MONITORAMENTO COM O AGENTE DE MONITORAMENTO BASE E A CONEXÃO
420
MONITORANDO CADA APLICAÇÃO OU GRUPO DE APLICAÇÕES UTILIZANDO O AGENTE DE MONITORAMENTO ASSOCIADO
430
MONITORANDO O SISTEMA OPERACIONAL BASE UTILIZANDO O AGENTE DE MONITORAMENTO BASE
440
DETECTANDO QUALQUER AÇÃO DE INTERESSE E EXECUTANDO AÇÃO PREVENTIVA OU CORRETIVA APROPRIADA
450

Fig. 4