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

(21) **PI0616018-2 A2**

BRPI0616018A2

(22) Data de Depósito: 27/07/2006
(43) Data da Publicação: 07/06/2011
(RPI 2109)

(51) Int.Cl.:
G06F 12/14 2006.01

(54) Título: **SISTEMAS E MÉTODOS DE SEGURANÇA PARA REDES DE COMPUTADOR**

(30) Prioridade Unionista: 29/07/2005 US 11/193,291, 29/07/2005 US 11/193,292, 29/07/2005 US 11/193,295, 29/07/2005 US 11/194,075, 29/07/2005 US 11/194,078

(73) Titular(es): Bit9, Inc

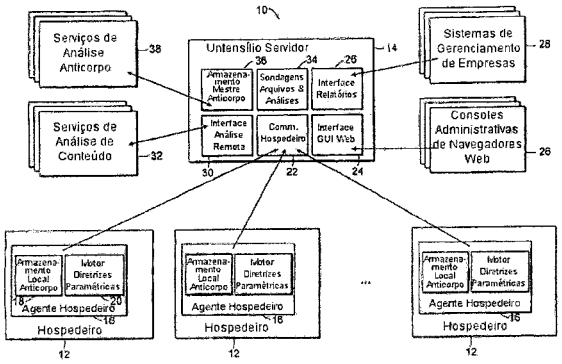
(72) Inventor(es): Allen Hillery, John Hanratty, Todd F. Brennan

(74) Procurador(es): Símbolo Marcas e Patentes Ltda

(86) Pedido Internacional: PCT US2006029714 de 27/07/2006

(87) Publicação Internacional: WO 2007/016478de 08/02/2007

(57) Resumo: SISTEMAS E METODOS DE SEGURANÇA PARA REDES DE COMPUTADOR, que consistem em sistemas e métodos de segurança que proporcionam uma defesa contra vírus conhecidos e desconhecidos, programas maliciosos, softwares espião, invasores, e softwares indesejados ou desconhecidos. O sistema pode implementar diretrizes centralizadas que permitem que um administrador aprove, bloqueie, coloque emquarentena, ou registre atividades de arquivos, O sistema mantém meta- informações de arquivos nos hospedeiros e no servidor. Um hospedeiro detecta operações de arquivos que podem provocar mudanças no conteúdo do arquivo ou no nome do arquivo, e atualiza as meta-informações do hospedeiro e/ou do servidor como resultado. As modificações das meta-informações do servidor são disponibilizadas para os hospedeiros.



PI - 0616018-2

"SISTEMAS E MÉTODOS DE SEGURANÇA PARA REDES DE COMPUTADOR"

Campo Técnico

Grandes empresas aplicam vultosas verbas em tecnologias de informação (IT) e em sistemas de segurança com camadas IT, no entanto, compromissos de rede e danos decorrentes de vírus, programas maliciosos e de softwares espião são normais. As tecnologias de segurança IT atuais são dispendiosas para serem mantidas e não proporcionam proteção contra ameaças novas ou desconhecidas, enquanto novas ameaças são distribuídas, detectadas e relatadas em taxas crescentes.

Soluções de segurança que estão localizadas nos perímetros das redes, tais como *firewalls*, têm visibilidade limitada do tráfego de rede que passa diretamente por elas. Vetores de entrada, tais como vírus de e-mails, exploradores de falhas de navegadores web, acessos sem fio, VPN's, mensageiros instantâneos e compartilhadores de arquivos geram um perímetro poroso crescente que ignora estas tecnologias. É difícil definir um perímetro em uma rede moderna que proporcione suficiente proteção e visibilidade. Muitos ataques geram tráfego na rede somente depois que eles comprometeram uma máquina ou uma rede. Por exemplo, no instante em que um vírus começa a enviar e-mails a partir de uma máquina da rede, esta máquina já está comprometida. Para parar os ataques antes que eles sejam executados, é normalmente necessário proteger os arquivos, não apenas o tráfego da rede.

Visibilidade e proteção podem ser proporcionadas por um agente hospedeiro, que é um software, algumas vezes utilizado em conjunto com um hardware, que opera em múltiplos computadores individuais, os "hospedeiros" dentro da rede. Agentes hospedeiros geralmente trabalham em paralelo, utilizando alguns dos recursos do hospedeiro para executar funções de segurança em segundo plano. Por terem potencialmente acesso a todas as funções internas e significativas de um hospedeiro, agentes hospedeiros podem teoricamente detectar e parar ameaças em hospedeiros antes que quaisquer danos sejam perpetrados. Sistemas de segurança de agentes hospedeiros são algumas vezes denominados Sistemas de Segurança de Ponto Final, por que eles operam nas

"extremidades" da rede.

Os sistemas atuais de segurança de ponto final de empresas freqüentemente tentam detectar e bloquear ataques com padrões conhecidos de bits, tais como varreduras ou escaneamentos antivírus (AV) e varreduras ou escaneamentos anti-software espião (AS). Varreduras com padrões utilizam listas negras de padrões que são identificados previamente como sendo ruins. Similarmente, alguns sistemas de segurança utilizam perfis detectados de comportamentos conhecidos, que podem ser descritos como uma lista negra de padrões de comportamento ruim. Em ambos os casos, as listas negras estão permanentemente desatualizadas, incapazes de responder aos ataques que são novos ou desconhecidos. Listas negras também são ineficazes contra ataques, tais como de novos vírus, que podem se disseminar mais rapidamente do que a capacidade de originar, testar e distribuir atualizações de listas negras. Como dúzias de novos vírus são descobertos a cada semana, listas negras de todos os tipos se tornam cada vez mais ineficazes. Padrões de comportamento são complexos para serem desenvolvidos e testados, e como resultado apresentam alta taxa de alarmes falsos; isto é, eles erroneamente concluem que um comportamento é ruim quando de fato ele é benigno. Como os novos ataques evoluem, o comportamento muda, o que leva a erros e a falhas de detecção. Ficar aguardando até que um ataque, como o de um vírus, exiba mau comportamento, faz com que a máquina afetada já esteja comprometida. Em resumo, listas negras tentam rastrear o que já se sabe que está errado, enquanto que o que está errado está constantemente mudando.

Outra tecnologia de ponto final de empresas é a detecção de anomalias. Esta pode ser vista como uma listagem negra comportamental que é determinada estatisticamente através da observação do comportamento por determinado tempo. Ao herdar adicionalmente as deficiências das listas negras comportamentais, a detecção de anomalias adiciona novos modos de erros por serem os comportamentos bons e ruins estimados estatisticamente, portanto erros estimativos certamente existem. Este processo freqüentemente conduz a altas e inaceitáveis taxas de alarmes falsos e de falhas de detecção.

Outra classe de segurança de ponto final limita a

execução apenas de programas que constem em listas brancas, que são listas de padrões de programas conhecidamente bons. Se um programa não estiver incluído na lista, ele não roda. Um sistema deste tipo não é suficientemente flexível para uma empresa tipicamente moderna, e as listas brancas resultantes são difíceis de serem mantidas. Por exemplo, a maioria das grandes empresas prepara programas personalizados que são desenvolvidos na casa e que mudam freqüentemente. Além disso, estes programas podem conter propriedade intelectual sensível e riscos de segurança que não devem ser expostos a uma terceira parte. É improvável que um vendedor de listas brancas tenha acesso para pré-aprovar um software deste tipo em tempo hábil. Outros exemplos são sistemas operacionais e outras atualizações. Novamente, não existe uma instituição central ou uma autoridade central que possa certificar que certos programas ou atualizações são bons para todas as empresas. Os modos de falha de sistemas de listas brancas são severos, bloqueando o acesso a aplicações críticas, mas ainda não aprovadas, e a funções comerciais.

Como resultado, sistemas que classificam o acesso ao conteúdo de arquivos de modo centralizado em apenas dois estados, Aprovado e Banido, terão problemas com condições de disputa (temporização). Uma grande quantidade de softwares não se encaixa com clareza em nenhuma destas categorias, e não existe uma autoridade central que seja acreditada universalmente para todos os softwares dentro de uma empresa. Mesmo que isto não seja um fator relevante, pode levar tempo para classificar o software intermediário. No caso de um novo vírus, pode-se levar de 6 a 48 horas, ou mais, para classificar um vírus como sendo ruim, e neste tempo gerar uma condição de pandemônio. Portanto, mesmo com uma forte conectividade de rede do hospedeiro à autoridade central de aprovação, pode levar mais do que alguns minutos para detectar e analisar um novo software. Para adicionar transparentemente esta autorização baseada em conteúdo a um sistema operacional em segundo plano, os retardos devem ser tipicamente menores que um minuto, ou de outra maneira o intervalo de tempo do sistema de arquivos pode se esgotar, e erros falsos de bloqueio de acesso podem ocorrer.

Fundamentos da Invenção

Sistemas de segurança como os aqui descritos permitem que um administrador detecte, monitore, localize, identifique e controle arquivos instalados em uma grande rede de computadores. O sistema pode proporcionar uma defesa contra vírus conhecidos e desconhecidos, programas maliciosos, softwares espião, invasores, softwares não aprovados / não desejados (por exemplo, aplicações de software que são contrárias à diretriz comercial corrente) e ataques de engenharia social. Os administradores podem acessar informações detalhadas e estatísticas de novos executáveis, roteiros, e roteiros embarcados à medida que eles aparecem e se espalham pelos sistemas em rede. O sistema pode implementar diretrizes centralizadas que permitem que um administrador aprove, bloqueie, coloque em quarentena e registre atividades de arquivos. O sistema também pode coletar informações detalhadas indicadas para diagnosticar e localizar arquivos problemáticos ou ataques. O sistema oferece visibilidade, controle e proteção para instalações de computadores de grande porte.

A arquitetura do sistema preferencialmente inclui softwares agente que rodam em cada hospedeiro protegido, e um servidor, referenciado como sendo um "servidor", que proporciona diretrizes centralizadas de gerenciamento, monitoramento de eventos, coordenação de agentes e varredura de vírus. O servidor pode ser implementado como sendo um utensílio (o que geralmente sugere um dispositivo de funcionalidade mais limitada). Um único utensílio pode suportar muitos hospedeiros, por exemplo, 10.000 hospedeiros. Um servidor adicional ou utensílio, algumas vezes denominado "super servidor", pode monitorar múltiplos utensílios.

O software agente rodando em cada computador hospedeiro protegido, analisa a atividade do sistema de arquivos e toma ações baseadas nas diretrizes configuradas nos servidores. Em uma implementação, quando um hospedeiro tenta abrir ou escrever um arquivo, o software agente calcula uma tabela de espalhamento (*hash)* do conteúdo do arquivo para identificar o arquivo com exclusividade para o sistema. O software agente utiliza esta tabela de espalhamento para verificar o estado e as diretrizes para o arquivo. Baseado nesta informação, o software agente pode bloquear uma operação, registrar um evento, colocar um arquivo em quarentena, ou tomar outras ações

específicas.

O sistema também incorpora muitos outros recursos que podem ser úteis de modo combinado ou individualmente, incluindo a habilidade de extrair arquivos de outros arquivos, a habilidade de extrair macros de arquivos, o rastreamento centralizado e a análise de conteúdo, e uma função de "encontrar arquivos" descrita neste documento.

Os sistemas aqui descritos podem utilizar pelo menos dois estados adicionais: Pendente, que representa um nível de ameaça intermediário e menos definido, e Localmente Aprovado, que é Aprovado para um hospedeiro mas não necessariamente Aprovado para a autoridade central (e conseqüentemente para todos os outros hospedeiros). O último permite que hospedeiros divirjam levemente da linha que serve de base. O estado Pendente permite que hospedeiros bloqueiem ou permitam acesso a novos conteúdos, baseados em vários níveis de ameaças e em diretrizes empresariais de utilização. Apesar de utilizarem a terminologia binária comum de aprovação, Aprovado ou Banido, a divisão da aprovação em 3 - 4 estados resulta em capacidades diferentes e melhoradas para cada estado individual. De um modo geral, um software que é novo e que ainda não foi classificado está Pendente. Estados tradicionais de acessos binários para softwares (banido / aprovado) não são suficientemente flexíveis, e sistemas de classificação deste tipo também não são escaláveis.

A designação de um software como novo / Pendente é conveniente. A maioria das empresas utiliza alguma forma de diretriz "não executar o novo", tal como "funcionários não estão autorizados a baixar e a rodar softwares não aprovados da Internet". Mesmo assim, as empresas não têm como detectar softwares novos enquanto eles se propagam, até que seja tarde demais, não sabem quando suas diretrizes estão sendo violadas, e não têm meios de efetivamente reforçar suas diretrizes. Através do rastreamento de novos programas Pendentes enquanto eles estão sendo modificados / escritos em um sistema de arquivos, um agente hospedeiro pode detectar e relatar um novo conteúdo, em tempo real, quando ele entra na rede através de qualquer meio, seja por e-mail, mensageiro instantâneo, *download*, memória USB, laptop móvel,

etc. Com a identificação de programas como Pendentes, algumas diretrizes escaláveis, simples e efetivas são possíveis, tais como: "Permita, mas alerte quando o hospedeiro rodar novos executáveis" ou "Nenhum programa novo não aprovado pode ser instalado ou rodar neste grupo de hospedeiros" ou "Alerte quando o mesmo programa novo não autorizado aparecer em mais de N hospedeiros em 24 horas". Desta maneira, programas novos podem ser localizados com segurança, rastreados e analisados enquanto estão bloqueados. Outros softwares comerciais aprovados continuam rodando. Softwares novos e aprovados podem ser instalados e rodar, tais como atualizações AV e atualizações de segurança. Este enfoque é uma resposta proativa, protegendo contra softwares desconhecidos e possivelmente maliciosos, permitindo produtividade e ganhando tempo de análise por não requerer atualizações críticas em tempo de listas negras ou listas brancas.

Os sistemas existentes de arquivos de listas brancas ou listas negras tendem a ser de natureza global, visto que é muito difícil manter um grande número de listas separadas centralmente, uma para cada hospedeiro. Como descrito aqui, os hospedeiros podem manter as suas próprias listas, que podem divergir da lista central. Particularmente, este pode ser o caso com os estados Local Aprovado e Pendente, e é freqüentemente verdadeiro com estados baseados em nomes, tais como NomeBanir e NomeAprovar. Como "nome" é geralmente uma propriedade local, estes estados podem divergir dos estados controlados centralmente. Por exemplo, se um arquivo "foo" tiver uma determinada tabela de espalhamento=x *(hash=x)* e um estado Pendente de um servidor central, em um hospedeiro o arquivo poderia ser Local Aprovado ou Nome-Banido ou Nome-Aprovado, os últimos dois dependendo do nome local do arquivo no hospedeiro. Os sistemas aqui descritos possibilitam o gerenciamento eficiente e a implementação de diretrizes de milhares de propriedades de nomes, aplicadas simultaneamente a cada arquivo de cada hospedeiro. O estado NomeAprovar permite capacidades flexíveis de aprovação local e de aprovação central, baseadas nos locais onde arquivos são criados no hospedeiro. Em conjunto com grupos de hospedeiros, isto permite especificações exatas, flexíveis e eficientes de onde e em qual hospedeiro o novo conteúdo é aprovado.

Mesmo com este sistema novo e flexível de diretrizes, as empresas normalmente necessitam reforçar diferentes diretrizes para diferentes regras e situações. Por exemplo, administradores IT e desenvolvedores internos de software podem precisar rodar cuidadosamente novos softwares, enquanto que outros funcionários requerem apenas um pequeno conjunto padronizado de aplicações relativamente estáticas. Esta situação pode mudar rapidamente sob condições de ataque. Por exemplo, se um vírus for detectado em mais do que N hospedeiros, pode fazer sentido expandir o escopo da diretriz "não executar o novo". Esta flexibilidade e a resposta incremental é uma vantagem do sistema de "Controle Paramétrico de Conteúdo", aqui descrito, comparado com sistemas rígidos que não conseguem se adaptar a diretrizes variáveis dentro de uma empresa e sob diferentes condições. O "Controle Paramétrico de Conteúdo" permite um modo flexível de travamento que pode ser gerenciado centralmente e ser rapidamente variado, baseado nas condições da rede e do hospedeiro. E isto permite restrições e aprovações incrementais de conteúdo de arquivos e/ou de arquivos baseados em nomes.

Ao contrário de outras tecnologias de segurança de ponto final que processam credenciais de usuários de hospedeiros, identificadores de processos, fontes de dados (URL), estruturas de diretórios e descritores de segurança de sistemas operacionais, o sistema aqui descrito não necessita utilizar estes fatores como parte das diretrizes do hospedeiro. Em um hospedeiro, estes fatores podem ser inseguros e podem ser vulneráveis a concessões, e podem impedir a escalabilidade. Estes fatores resultam em diretrizes menos escaláveis, pois diretrizes afinadas podem interagir através de uma variedade de hospedeiros em modos complexos. Mesmo que um sistema operacional esteja comprometido e um ataque ganhe privilégios administrativos com todos os descritores associados de segurança, uma diretriz "não executar o novo" como a descrita aqui irá proporcionar substancial proteção.

O sistema de "Rastreamento de Conteúdo" utiliza estados adicionais, tais como Pendente, para monitorar e analisar novos conteúdos à medida que eles se movem através da rede. As tecnologias atuais não permitem visibilidade global central e o rastreamento de cada arquivo executável novo,

através de um grande número de hospedeiros, em tempo real. Sistemas de Ponto Final baseados em varredura de arquivos de sistema, tais como varredores AV e aplicativos de inventário de hospedeiros tais como o Tripwire, rastejam lentamente e periodicamente através de grandes sistemas de arquivos procurando por softwares novos ou que tenham sido modificados. Esta é uma operação tipicamente disruptiva para o hospedeiro, pode levar horas de execução, e normalmente é programada para ser executada ao menos uma vez por dia. Ao focalizar aquilo que é novo e armazenar esta informação na memória, o sistema de Rastreamento de Conteúdo é mais escalável e responsivo. Como é raro que novos softwares apareçam e que nunca tenham sido visualizados por um hospedeiro em um grande número N, e como é raro que muitos hospedeiros M tenham visualizado este novo software em um curto período de tempo, relatórios, respostas e análises são facilitados por essa distinção.

Uma vez que um novo software é detectado, pode ser útil localizá-lo e identificá-lo em um tempo relativamente curto. Se um novo software for um ataque e estiver se espalhando, é desejável responder muito rapidamente. Novamente, as tecnologias correntes conseguem localizar arquivos novos únicos em hospedeiros únicos em uma rede, em uma escala de tempo de vários minutos a horas. Mesmo em um hospedeiro único, encontrar um arquivo muito novo pelo nome ou pelo conteúdo pode levar 15 - 60 minutos, o que terá impacto negativo no desempenho do disco do hospedeiro enquanto a consulta é processada. Nos últimos 20 anos, os discos rígidos se tornaram muito maiores em termos de capacidade de armazenamento de bytes, porém não aumentaram proporcionalmente em termos de velocidade. O recurso "Consulta de Meta-Informação Distribuída" acelera a localização e a identificação de atributos chave de arquivos em segundos, através de um grande número de hospedeiros (milhares), com consultas centralmente especificadas e resultados centralmente reportados, e com pequeno ou nenhum impacto nos discos dos hospedeiros. Ao contrário das tecnologias tradicionais de rastreamento que fazem o rastreamento em todos os arquivos, incluindo aqueles que não foram modificados, esta invenção rastreia modificações de arquivos na memória enquanto os arquivos estão sendo modificados, e isto proporciona um meio eficiente para consultar

hospedeiros por meta-informações de arquivos da memória. Processar esta informação centralmente proporciona, pela primeira vez, visualizações responsivas globais do movimento de arquivos individuais através de coleções de sistemas hospedeiros de arquivos. Finalmente, sendo um serviço de segurança, é importante que o hospedeiro conecte ao, poste no, e faça consultas ao servidor central. Esta é uma parte importante da invenção, pois ela permite que hospedeiros sejam separados de servidores por um ou mais *firewalls* ou dispositivos NAT, e o difícil problema de proteger o soquete de uma rede adicional de hospedeiros nos modos aceite / escute é evitado.

Sistemas atuais de agentes hospedeiros de ponto final que utilizam análise de conteúdo têm problemas com a atualização de agentes hospedeiros. Por exemplo, para que sistemas de varredura AV sejam mais eficientes, eles devem ser atualizados dentro de horas ou minutos após uma atualização ter sido colocada à disposição. Quaisquer hospedeiros com sistemas AV atrasados encontram-se em risco, e muitos sistemas AV estão inadequadamente configurados, resultando em atrasos de atualização. Em razão deles não rastrearem eficientemente modificações em arquivos, sistemas de varredura AV geralmente levam um tempo relativamente longo para responder a novos conteúdos escritos em um sistema de arquivos. Além disso, as tecnologias atuais de análise de conteúdo de hospedeiros reanalisam desnecessariamente arquivos sem levar em conta fatores de segurança. Por exemplo, é mais importante analisar novos conteúdos mais freqüentemente, quanto mais novos eles forem. Se um arquivo permaneceu completamente inalterado em uma rede por 2 anos, é muito provável que ele não necessite ser escaneado a cada dez minutos. Entretanto, se um novo arquivo está se espalhando através de uma rede, com início há dez minutos atrás, o escaneamento freqüente do arquivo nos primeiros dois dias faz sentido. Com o passar do tempo existem geralmente menos e menos novas informações sobre novos arquivos maliciosos executáveis. O recurso "Análise Centralizada em Tempo Determinado" trata destas questões. Apenas um agente de análises necessita ser atualizado, o agente central, e todos os hospedeiros se beneficiam disto imediatamente. Existe uma menor possibilidade de uma configuração de hospedeiro interferir com as atualizações

de análises de conteúdo. Através do rastreamento de apenas arquivos novos e através da programação de análises baseadas na idade (tempo) exposta à rede, novos conteúdos ruins podem ser localizados e identificados eficientemente e mais rapidamente. Finalmente, muitas tecnologias de análise de conteúdo de ponto final, tais como tecnologias AV, são fortemente integradas com o sistema operacional. Como resultado, pode ser dificultoso colocar diversos agentes de inspeção de conteúdo de diferentes vendedores em um hospedeiro. A diversificação das tecnologias de análise melhora a exatidão de detecção e classificação. Novamente, a invenção soluciona este problema com a utilização de um servidor central para despachar análises a diferentes servidores, se necessário.

Conteúdos executáveis (arquivos exe) e macros embarcadas (macros embarcadas dentro de documentos Microsoft Office) tendem a se propagar em conjuntos ou grupos. Um documento de processamento de textos pode conter 10 macros, e ter tamanho superior a 30 MB, no entanto as macros apenas ocupam uma fração deste espaço. Um grande pacote de instalação pode ter centenas de MB de tamanho, e ainda assim as porções executáveis de sua arquitetura interna ocupam tipicamente apenas uma pequena parte do tamanho total. Vírus viajam freqüentemente através de e-mails em arquivos anexos, tais como arquivos zip, para evitar a detecção. Dentro destes arquivos, a carga útil do vírus pode ser pequena. Em todos estes casos, arquivos "contêineres" maiores podem obscurecer a propagação de novos códigos não desejados. O recurso "Extrator de Conteúdo" atende a uma variedade das limitações correntes através da preservação das relações (aninhadas) de contêineres, e simultaneamente facilita: o rastreamento do conteúdo, o rastreamento de contêineres similares, o rastreamento de associações de produtos, a minimização de reanálises desnecessárias, a minimização da largura de banda na transferência de arquivos, preservando a compatibilidade com outras tecnologias de análise e reempacotando o conteúdo de acordo com outros tipos de arquivos conhecidos. O armazenamento central, o rastreamento de conteúdo e a programação central de análises relacionadas ao tempo da primeira aparição do conteúdo, proporcionam vantagens poderosas em termos de segurança,

visibilidade global, integração de sistemas de gerenciamento de empresas, e futuras expansões.

Enquanto que os sistemas aqui descritos foram diferenciados de outros sistemas, tais diferenciações não significam renúncia à cobertura de reivindicações a estes sistemas. Os sistemas e recursos aqui descritos podem ser fornecidos como um grupo ou separadamente, e em muitos casos, podem ser integrados em sistemas existentes e conhecidos, incluindo os identificados acima.

Outros recursos e vantagens se tornarão aparentes a partir dos desenhos seguintes, descrições detalhadas e reivindicações.

Breve Descrição dos Desenhos

A Figura 1 é um diagrama em blocos mostrando uma visão geral de um sistema de segurança como o aqui descrito.

A Figura 2 é um diagrama em blocos muito mais detalhado mostrando componentes do sistema da Figura 1.

A Figura 3 é um fluxograma ilustrando um processo para executar uma análise.

As Figuras 4-5 são esquemas de processos realizados pelo sistema.

A Figura 6 é um gráfico mostrando um exemplo de uma análise em tempo.

A Figura 7 é um fluxograma de passos executados durante uma análise em tempo.

A Figura 8 é um esquema de um processo de extração de conteúdo.

Descrição Detalhada da Invenção

Com referência à Figura 1, um sistema, também referenciado como um sistema digital anticorpo (DAS) 10, permite que um administrador monitore, entenda e controle arquivos instalados em uma grande rede de computadores, e proporciona uma defesa contra vírus conhecidos e desconhecidos, programas maliciosos, softwares espião, invasores e ataques de engenharia social, bem como contra softwares não aprovados (por exemplo,

compartilhadores de arquivos de uso não comercial). O sistema inclui um ou mais servidores, um dos quais é mostrado aqui como servidor 14 (utensílio). Este servidor fornece diretrizes centralizadas de gerenciamento, monitoramento de eventos, coordenação de agentes e análise de conteúdos (por exemplo, varredura de vírus e softwares espião). Um servidor pode suportar inúmeros hospedeiros 12, por exemplo, centenas ou milhares de hospedeiros. O servidor também mantém uma base de dados de metadados relacionados a análises, tais como históricos de varreduras e estados de aprovação, com respeito a arquivos e programas. Estes metadados são referenciados como um "anticorpo" para cada um dos arquivos e programas.

Cada hospedeiro protegido 12 tem um agente hospedeiro 16, preferencialmente implementado por software. Ele analisa a atividade de sistemas de arquivos e toma ações baseadas em diretrizes configuradas em um servidor. Estas diretrizes, descritas com mais detalhes abaixo, identificam se devem bloquear, registrar, permitir ou colocar em quarentena, ações tais como acessos a arquivos e a execução de executáveis. Cada agente hospedeiro 16 tem um armazenamento local "anticorpo" 16, que é um *cache* ou memória rápida de meta-informações relacionadas a arquivos, e um motor de diretrizes paramétricas 20 para a implementação das diretrizes do servidor 14.

O servidor 14 tem um determinado número de funções e interfaces. As interfaces incluem a interface de comunicação do hospedeiro 22 para a comunicação com os hospedeiros, uma interface gráfica de usuário (GUI) baseada na *web* para a comunicação com consoles administrativas 26 através de um programa navegador, uma interface de relatórios 26 para servir de interface com sistemas de gerenciamento de empresas 28, e uma interface de análise remota 30 para a comunicação com serviços de análise de conteúdo 32 (por exemplo, varredores de vírus e softwares espião). O servidor 14 também inclui um bloco de análises 34 e um armazenamento mestre anticorpo 36, que se comunica com serviços de análise anticorpo 38 e que armazena uma lista mestre de anticorpos para o hospedeiro associado. Os serviços 38 podem incluir uma autoridade de certificação externa com informações adicionais associadas aos anticorpos, como por exemplo, a classificação de um anticorpo como sendo um

membro de um determinado pacote de produtos, tal como o Microsoft Office.

A Figura 2 mostra uma vista expandida do sistema e de seus componentes, incluindo o servidor 14, o hospedeiro 12 com porções de usuário e de núcleo, uma outra rede e serviços *web* 40. Como ilustrado aqui, o servidor incorpora um bloco de processamento e armazenamento de novos arquivos 42, que inclui cópias de arquivos recentes que apareceram na rede, um motor de análises programadas 44 para a identificação de arquivos e espalhamentos que devem ser analisados, um assinador de conteúdo 46 para criar espalhamentos criptografados de conteúdos utilizando algoritmos tais como o MD5 e o SHA-1, um armazenamento mestre anticorpo 36, um gerenciador de configurações 50 e serviços de registros e relatórios 52. O servidor interage com os serviços de rede e *web* 40 incluindo a execução de análises 54, varredura AV (ou de outro conteúdo) 56, e serviços de gerenciamento 57.

A porção usuário 60 do hospedeiro 12 conta com uma memória *cache* anticorpo 64 para manter as atualizações da base de dados 34 por ambos o nome e a data, com o processamento de arquivos e eventos 66, com um motor de análises 68, com um extrator de conteúdo 70 para a extração de conteúdo de interesse e para a associação de grupos de conteúdo individual em um pacote, com um assinador de conteúdo 72 para a criação de espalhamentos criptográficos dos conteúdos, com um resolvedor de estados 74 de meta-informações (MI) do servidor, para verificar a memória *cache* anticorpo 64 para o anticorpo e para verificar o servidor para o anticorpo, e com um resolvedor de estados de arquivos 76, para verificar o progresso do carregamento de conteúdo para o servidor e para a verificar a certificação do servidor no carregamento.

A porção núcleo 80 do servidor 12 tem uma memória *cache* 82 para o salvamento de anticorpos organizados por nome de arquivo, e uma memória *cache* 84 de operações recentes com arquivos e informações de arquivos. O núcleo também tem uma função de interceptação / bloqueio 86 que recebe e intercepta requisições de operação de arquivos, e que fornece estas requisições a um filtro de estado 88 que primeiramente verifica a memória *cache* de operações recentes de arquivos 84. Se não existir coincidência, ele verifica os disparadores e o bloco de ações 90 que mantém as diretrizes de segurança. Este

bloco 90 é acoplado a um bloco "*defcon*" 92, que tem um valor que indica um nível de segurança para o sistema, e a um motor de diretrizes 94 que governa os blocos 82, 90 e 92, para controlar várias operações com arquivos incluindo execuções, leitura de arquivos, escrita de arquivos e outras ações. Os disparadores e o bloco de ações 90 se comunicam com a memória *cache* anticorpo, que verifica por meta-informações em arquivos baseadas em seus nomes. O motor de diretrizes 94 também controla ações, tais como bloqueios, relatórios, ou a permissão de execução de arquivos e a geração de relatórios para o usuário.

O sistema incorpora numerosos métodos e aspectos para a utilização deste sistema de segurança, muitos dos quais podem ser utilizados individualmente ou combinados com outros. Estes métodos e aspectos serão descritos com mais detalhes abaixo.

Um aspecto é a utilização da varredura centralizada para a verificação de documentos e executáveis, e para manter um espalhamento que indica se os dados foram previamente verificados. Os valores de espalhamento podem ser armazenados em um banco de dados e também verificados em um hospedeiro local.

Outro aspecto é a utilização de um parâmetro definido centralmente, algumas vezes designado como "D" ou "*Defcon*", que controla as diretrizes dos hospedeiros. Esta diretriz e o parâmetro central podem ser aplicados a todos os hospedeiros, ou a um grupo selecionado de hospedeiros. O parâmetro pode ser definido manualmente por um operador ou pode ser ajustado pelo sistema sem intervenção humana, tipicamente em resposta a algum evento. As diretrizes podem incluir bloqueios ou permitir determinadas ações, ou podem fazer uma ação pendente, que a faz autorizada, sujeita ainda a operações de monitoramento tais como registro. Um estado pendente apresenta múltiplos benefícios, incluindo o ato de levar em conta latências do sistema, bem como a implementação de diretrizes que não se enquadram nos modelos binários tradicionais de aprovar / banir. Estas latências incluem o tempo antes que códigos perigosos sejam identificados, durante defeitos no sistema, ou o tempo em que um hospedeiro está desconectado da rede.

Ainda em outro aspecto, um servidor central pode especificar uma consulta de meta-informações e distribuir esta consulta a todos ou a um grupo selecionado de hospedeiros. Estes hospedeiros realizam a consulta a partir de um armazenamento local de meta-informações e enviam os resultados de volta ao servidor, o que pode fazer com que o servidor ajuste o parâmetro.

Em outro aspecto posterior, o sistema inclui um método de proteção contra a disseminação de vírus de macros que podem estar embarcados dentro de outros documentos. Esta funcionalidade pode ser utilizada com macros Visual Basic, mas os métodos também podem ser aplicados a qualquer outra linguagem de macros que o Visual Basic.

Em um outro aspecto, todas as cópias de novos arquivos são mantidas em um diretório especial em um servidor 42. Análises adicionais podem ser executadas baseadas em temporizadores, e podem ser executadas dias após o arquivo ter sido visualizado pela primeira vez. Após um período de tempo da primeira aparição de um arquivo, tal como 30 dias, o arquivo pode ser reescaneado contra vírus, softwares espião ou outros problemas, e o sistema pode tomar ações dependendo dos resultados. Por exemplo, uma análise que indica que um vírus está contido em um arquivo faz com que a entrada correspondente do Banco de Dados Anticorpo 36 seja incluída em um estado Banido. Esta mudança, em conjunto com outras mudanças no Banco de Dados Anticorpo, será propagada para os Hospedeiros.

Parâmetro Definido Centralmente e Diretriz de Conteúdo Paramétrico

A segurança do sistema é baseada em diretrizes que são definidas em cada servidor e propagadas a todos os hospedeiros associados ou grupos de hospedeiros, através de técnicas empurra e/ou puxa. Estas diretrizes se relacionam com o que pode ser feito com executáveis e arquivos, tal como leitura, execução e escrita, com o que fazer quando eles são alterados por hospedeiros, como escaneamentos são rodados, como os registros são feitos, e muitas outras funções, e para cada diretriz (por exemplo, quais operações podem ser feitas com um executável recentemente visualizado) pode existir um número de opções de diretrizes (tais como banir, permitir, ou permitir e registrar). As

diretrizes podem ser baseadas no conteúdo (dados) de um arquivo, ou no nome do arquivo, ou em uma combinação. O conteúdo pode ser definido por uma assinatura, tal como um ou mais espalhamentos criptográficos. Uma listagem não exclusiva de amostras de diretrizes inclui:

1. Bloquear / registrar a execução de novos executáveis e roteiros destacados (por exemplo, *.exe ou *.bat)

2. Bloquear / registrar a leitura / execução de novo conteúdo embarcado (por exemplo, macros em *.doc)

3. Bloquear / registrar a instalação / modificação de conteúdo *Web* (alteração de conteúdo em arquivos *.html ou *.cgi)

4. Permitir atualizações para diretrizes tais como (3) acima

5. Autoaprovação de arquivos que tenham passado por duas varreduras de vírus (por exemplo, definir o estado correspondente do arquivo para Aprovado)

6. Bloquear / registrar a instalação / execução de arquivos especificamente banidos pelo administrador

7. Colocar em quarentena / apagar / registrar arquivos infectados pelos dados

8. Colocar em quarentena / registrar arquivos infectados pelo nome

9. Bloquear / registrar a execução de novos arquivos em uma "classe" administrativamente definida; por exemplo, um administrador poderia querer bloquear protetores de tela *.scr, mas não a classe inteira de executáveis *.exe, *.dll, *.sys, etc...

10. Registrar quando arquivos específicos são copiados para mídias removíveis

11. Bloquear / registrar a execução de novos executáveis, roteiros e conteúdos embarcados, exceto em um determinado diretório, isto é, permitir que um usuário crie novos roteiros ou executáveis em um diretório especial mas protegendo o restante do sistema de arquivos

12. Diferentes diretrizes para hospedeiros desconectados,

conectados remotamente ou conectados localmente

13. Listar hospedeiros / caminhos que contém um arquivo específico por dados ou por nome

14. Listar hospedeiros com executáveis bloqueados, roteiros, e roteiros embarcados

15. Listar hospedeiros / caminhos com arquivos infectados ou banidos

16. Auto-aprovar arquivos de serviços definidos de atualização, por exemplo, se forem de fontes confiáveis

17. Bloquear / registrar a execução de arquivos especificamente banidos pelo administrador para grupos específicos de hospedeiros (isto é, existe mais de um grupo)

18. Desativar completamente o sistema hospedeiro por razões de desempenho e testes.

19. Auto-aprovar um arquivo após um período de tempo (configurável pelo usuário)

20. Permitir que um novo arquivo seja instalado / executado até x vezes (configurável pelo usuário). Bloquear qualquer outra instalação e/ou execução até a aprovação.

21. Aprovar localmente novos arquivos à medida que eles são escritos

22. Aprovar centralmente novos arquivos à medida que eles são escritos

O servidor pode manter uma ou mais diretrizes para cada grupo de hospedeiros, e cada diretriz é variavelmente reforçada de acordo com um parâmetro que é centralmente definido e que indica as opções para as diretrizes. Estas diretrizes e opções podem ser logicamente organizadas em um arranjo bidimensional, no qual o parâmetro de fato se move ao longo de uma dimensão para selecionar as opções de diretrizes das variadas diretrizes. Este parâmetro é referenciado aqui como um valor D. Todos os hospedeiros podem ter um valor para D, ou subgrupos lógicos de hospedeiros podem ter seus próprios valores D, por exemplo, hospedeiros do departamento de vendas podem ser

atribuídos com D=1 e hospedeiros do departamento de marketing podem simultaneamente ser atribuídos com D=2. Em uma implementação, hospedeiros verificam (sondam) o servidor para determinar se o valor D mudou. Assim que cada um dos hospedeiros descobre que o valor de D mudou, cada um deles começa a se "mover" para o novo valor de D. Este movimento pode ser feito em etapas. Estas sondagens podem ser providenciadas como mensagens de rede do hospedeiro para o servidor. O valor D controla as ações das diretrizes. Para uma dada diretriz (por exemplo "Nenhuns Novos Executáveis" ou "Nenhuns Novos Roteiros"), D=2 bloqueia ações de violação de diretrizes (neste caso, a execução de um "novo executável"), D=4 alerta (alarme silencioso para o servidor) mas permite, e D=6 permite e não alerta de forma alguma. Indiferentemente se D=2, 4 ou 6, o hospedeiro continua preferencialmente a notificar e a gravar novos executáveis à medida que eles são escritos. Enquanto que os exemplos aqui descritos utilizam um valor numérico para D, D pode ter um "valor" expresso em letras, palavras, ou qualquer combinação de letras e numerais.

O valor D também controla a ativação das diretrizes. Para uma dada diretriz (por exemplo "Nenhuns Novos Executáveis" ou "Nenhuns Novos Roteiros"), D=1 habilita a diretriz "proteção contra escrita", e assim novos executáveis não podem ser escritos de forma alguma, D=8 desabilita completamente todas as diretrizes e os casos D=2, 4 e 6 podem ser definidos como explicitado acima. Neste caso, D=8 pode inclusive desabilitar a notificação da diretriz quando novos executáveis são escritos.

Enquanto que o valor de D pode ser definido centralmente em um servidor, ele é implementado localmente em um hospedeiro. Ele pode ser definido por um administrador através de uma interface gráfica de usuário (GUI) em uma console administrativa com a utilização de um navegador conectado a um servidor, ou através do Protocolo de Gerenciamento de Redes Simples (SNMP). Os valores D são considerados valores "alvo"; os hospedeiros tentam se aproximar o máximo possível desse valor, o que pode demandar segundos ou minutos. Em alguns casos, um hospedeiro pode divergir localmente de um valor alvo de acordo com a definição do servidor. Um programa com linha de comando pode ser invocado no hospedeiro, ou o usuário pode ser questionado por certos

valores de D, e o valor alvo de D pode ser sobrescrito. Este recurso pode ser útil, por exemplo, nos casos nos quais uma máquina individual requer que a sua segurança seja desabilitada (D=8) e quando não existe conectividade de rede com o servidor. Determinadas ações podem mudar automaticamente o valor de D no hospedeiro, tal como a detecção de uma atualização de um programa autorizado (por exemplo, uma atualização antivírus).

As diretrizes refletem um compromisso entre a segurança e a usabilidade. Nos exemplos acima D=8 é maximamente utilizado, e minimamente seguro – nenhuma restrição é ativada, e os agentes hospedeiros são efetivamente desabilitados de bloqueios e rastreamentos. À medida que D se move em direção à máxima segurança (D=1), mais e mais diretrizes restritivas são ativadas, e as ações executadas quando as diretrizes são violadas se tornam mais e mais severas. Estados solicitados são desejáveis porque eles são mais fáceis de visualizar e testar (geralmente, alguém pode testar os pontos finais que devem ser testados, tais como D=1 e D=8). Com estados solicitados o número de arquivos e usuários se torna sucessivamente mais acessível ou mais restritivo quando o valor é aumentado ou diminuído. Estes estados solicitados naturalmente refletem compromissos entre a segurança e a usabilidade.

À medida que D é modificado em um sistema vivo, condições de disputa podem ocorrer. O problema básico é que uma instalação de múltiplos arquivos pode se tornar "meio-bloqueada" ou "meio-instalada" se o valor de D for mudado de 8 → 1 durante a instalação de um programa. Como resultado, certas transições D podem disparar estados de reanálise de arquivos anticorpo e transformações de estados de massa de arquivos anticorpo.

Modificações locais D podem ser, algumas vezes, causadas por disparos locais de diretrizes. Normalmente, D é definido centralmente no servidor. Mas, algumas vezes, uma diretriz de um hospedeiro local é disparada, o que então faz com que o valor D do hospedeiro local mude. Isto é útil, por exemplo, para completar uma instalação em um sistema bloqueado (D=2). Continuando com este exemplo, a instalação de *drivers* de impressora em D=2 pode de outra forma resultar em problemas, porque alguns dos novos arquivos de instalação desempacotados necessitam ser executados para

completar a instalação. Além disso, diferentes máquinas hospedeiro podem ser necessárias para desempacotar e executar diferentes programas para completar a instalação (por exemplo, Windows 2000 e Windows XP). Neste caso, a execução de um determinado tipo de arquivo anticorpo, um programa "printer_setup.exe", irá mover o local D deste hospedeiro de 2 → 3, o que é um estado levemente mais fraco e que automaticamente aprova localmente apenas estes novos arquivos de instalação e suas descendências.

O valor de D pode ser modificado dependendo do tipo de conectividade, seja ela local (em uma LAN cabeada), remota tal como através de um modem de telefone ou rede virtual privada (VPN), ou completamente desconectada. O agente hospedeiro armazena, portanto, um conjunto de valores especificados de D para estes tipos de conectividade e então automaticamente os seleciona do conjunto em resposta a uma mudança, por exemplo, quando um usuário desconecta o hospedeiro da LAN. Também, diferentes valores de D podem resultar em diminuições de relatórios, registros e rastreamento de detalhes.

Diretrizes também podem ser definidas a partir de um servidor central, algumas vezes referenciado como sendo um "super servidor", que pode controlar muitos servidores / servidores. Assumindo-se que cada servidor controla 2.000 hospedeiros, e que existem 1.000 super servidores, é improvável que um comando do super servidor definido como D=1 seja apropriado para todos os 2.000.000 de hospedeiros. Ao invés disso, o super servidor pode comandar todos os servidores e hospedeiros para ter um D forte como localmente permitido. Desta forma alguns servidores, e seus hospedeiros conectados, irão aos seus limites, por exemplo, D=2. Outros servidores poderiam ir para D=1, mas então talvez alguns de seus grupos de hospedeiros seriam limitados a D=4, assim estes hospedeiros iriam ser tão fortes, mas não mais fortes que D=4. A mesma limitação é verdadeira para a outra extremidade do espectro. Se o super servidor comandar D=8, alguns servidores e hospedeiros poderiam ir somente para D=6. Como D é um estado selecionado, estas restrições são simples faixas de inteiros (máximos e mínimos).

O valor de D pode mudar baseado na detecção de algum

evento, tal como a propagação de arquivos. Se um número excessivo de cópias de um novo arquivo estiver se propagando entre os hospedeiros de um servidor, o servidor pode opcionalmente aumentar o valor de D para parar a propagação (por exemplo, ir para D=2). Este evento pode ser especificado como demais com um determinado nome (por exemplo, uma lista dos 10 por nome) ou por demais com conteúdo único (por exemplo, uma lista dos 10 por um espalhamento de dados).

Este valor também pode ser modificado por uma requisição do servidor em resposta a um novo evento percebido pelo servidor, tal como um novo arquivo entrante ou um potencial ataque de vírus. Na maioria dos casos, é o administrador (uma pessoa) que inicia a mudança de D, seguindo uma operação planejada de usuário, ou uma observação de eventos de determinados arquivos. D pode ser automaticamente modificado, por exemplo, durante o processo de uma operação, e neste caso, o hospedeiro / servidor irá retornar o valor de D de volta ao seu nível original assim que a operação tiver terminado. Disparadores externos podem modificar o valor de D tais como SNMP.

Uma outra resposta é adequada para que o servidor aprove automaticamente o conteúdo que está em um número menor do que um certo número limiar de hospedeiros, ainda que automaticamente impeça o acesso a este conteúdo quando o número de hospedeiros é excedido. Uma diretriz deste tipo pode ser utilizada para limitar o número de cópias de qualquer conteúdo ou arquivo na rede. Uma diretriz deste tipo também pode ser utilizada para somente relatar o conteúdo que exceder um certo número de hospedeiros.

Servidores podem manter um conjunto de diretrizes separadamente para cada grupo lógico de hospedeiros, tais como hospedeiros de vendas, hospedeiros de marketing, e hospedeiros de engenharia. Conjuntos de diretrizes podem ter números únicos de identificação que são similares a números de versões anticorpo. A diferença é que, uma vez organizado, um conjunto de diretrizes se torna "somente leitura" para reconciliar problemas posteriores com um conjunto de diretrizes e para desfazer um problema de organização. Isto também pode ser feito para diferenciar configurações e outras atualizações, utilizando tecnologia similar aos utilitários Unix "diff" e "patch". Hospedeiros podem questionar o servidor pelo numero ID do atual conjunto de diretrizes para o

seu grupo, e, se houver um desacordo, eles podem enviar ao servidor uma requisição "Receber Conjunto de Diretrizes".

Um conjunto de diretrizes pode incluir múltiplas diretrizes, tais como uma diretriz "Novos Executáveis" e uma diretriz "Novo Roteiro". Cada diretriz pode estar em modo ativo (ligado), inativo (desligado), ou em modo de teste (quando bloqueios são permitidos mas uma mensagem "foram bloqueados" é enviada ao servidor). Cada diretriz pode conter múltiplas regras, cada regra com um modelo básico "disparo e ação". Disparos são padrões que são testados. Se os padrões coincidirem, as ações resultantes são executadas. Por exemplo, "bloquear a execução de novos executáveis em D=2" pode ser especificado como segue:

Disparo=(D=2 & ArquivoOp=Executar & Estado=Pendente & ClasseExtensãoArquivo=ClasseExecutável) onde ClasseExecutável = (*.exe | *.sys | *.dll | ...)

Ação=(Bloquear & Reportar & Notificar(P)) onde "Bloquear" pára a operação, "Reportar" envia uma notificação para o servidor, e "Notificar" alerta o usuário com o conjunto de parâmetros P.

Com esta estrutura, o núcleo pode reforçar todas as diretrizes sem interação com o espaço do usuário, exceto no caso de atualizações da memória *cache* anticorpo do núcleo, atualizações D, e atualizações do conjunto de diretrizes. Conjuntos de diretrizes necessitam ser armazenadas em apenas um lugar, e elas necessitam apenas ser interpretadas dentro do núcleo nesta implementação. Conjuntos de diretrizes podem ser autenticados e armazenados em um contexto seguro (o núcleo), resultando em maior segurança contra violações.

Diretrizes e ações são parametrizadas por D pois D permite diferentes regras para combinar diferentes disparadores. Arquivos com determinados estados podem ter certas operações bloqueadas. Estes estados podem ser uma combinação de propriedades de nomes e dados. Estes estados são determinados no espaço do usuário, espelhados no espaço do núcleo, e finalmente, os estados são determinados pelo servidor. Uma diretriz útil é bloquear arquivos banidos, e em alguns valores D, bloquear a execução de

arquivos de arquivos (novos) pendentes.

As diretrizes podem ser providenciadas como um conjunto de listas de diretrizes, sobre uma faixa de compromissos de acessibilidade e segurança. O servidor pode então fornecer informações que fazem com que os hospedeiros selecionem uma das listas. Estando as listas presentes nos hospedeiros e permitindo que os hospedeiros atualizem diretrizes utilizando o enfoque "puxar", os hospedeiros podem convenientemente atualizar diretrizes de segurança sob controle do servidor.

A tabela seguinte mostra um exemplo de como o valor D pode afetar várias diretrizes em um conjunto mestre de diretrizes, onde as linhas representam diretrizes no conjunto mestre, as colunas representam ações, e as células tem faixas numéricas de D para indicar as ações. As ações especificadas na tabela e outros detalhes são resumidos abaixo:

Valor D x Nome Diretriz	D=10 Aprovação Global	D=8 Proteção Desabilitada	D=6 Apenas Rastreamento	D=4 Alarme Silencioso	D=3 Aprovação Local	D=2 Travamento	D=1 Proteção Escrita
Executáveis Novos/ Pendentes *.exe. *.sys, ...	Aprovação Auto Global Novo, Reporta	Permite	Permite	Permite execução, Reporta	Aprovação Auto Local Novo, Reporta	Execução em Bloco, Notifica, Reporta	Bloco Escrita/ Execução, Notifica, Reporta
Roteiros Autônomos Novos/ Pendentes *.vbs, *.bat, ...	Aprovação Auto Global Novo, Reporta	Permite	Permite	Permite execução, Reporta	Aprovação Auto Local Novo, Reporta	Execução em Bloco, Notifica, Reporta	Bloco Escrita/ Execução, Notifica, Reporta
Roteiros Embarcados Novos/ Pendentes em *.doc, *.xls, ...	Aprovação Auto Global Novo, Reporta	Permite	Permite	Permite execução, Reporta	Aprovação Auto Local Novo, Reporta	Execução em Bloco, Notifica, Reporta	Bloco Escrita/ Execução, Notifica, Reporta
Novo Conteúdo Web *.html, *.asp, ...	Aprovação Auto Global Novo, Reporta	Permite	Permite	Permite Escrita, Reporta	Permite Escrita, Reporta	Proteção Escrita, Reporta	Proteção Escrita, Reporta
Aprovado (espalhamento e/ou nome) Exes/Roteiros/ Embarcados	Permite	Permite	Permite	Permite	Permite	Permite	Permite
Banido/Não Aprovado (por espalhamento) Exes/Roteiros/ Embarcados	Permite	Permite	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta
Banido/Não Aprovado (por espalhamento) Exes/Roteiros/ Embarcados	Permite	Permite	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta	Execução em Bloco, Notifica, Reporta
Rastreamento mudança de Conteúdo e criação de Conteúdo	Rastreia, Reporta	Permite	Rastreia Reporta	Rastreia Reporta	Rastreia Reporta	Rastreia Reporta	Rastreia Reporta

(1) Permitir: Permite a operação, caso contrário silêncio

(2) Bloquear: Bloqueia a operação, caso contrário silêncio

(3) Rastrear: Rastreia a operação e o conteúdo resultante

(se o conteúdo é Pendente ou Banido), do contrário silêncio. Conteúdo Aprovado geralmente não é rastreado.

(4) Reportar: Envia uma notificação para o servidor

(5) Notificar: Indica para o usuário do ponto final do hospedeiro por que a operação foi bloqueada / interrompida

(6) Auto Aprovação Local: Novos arquivos de hospedeiro e/ou novos conteúdos com o estado local do hospedeiro=Pendente, são definidos localmente para o estado do hospedeiro=Aprovado ou para o estado=LocalmenteAprovado somente no hospedeiro local, enquanto os arquivos / conteúdos são criados / modificados.

(7) Auto Aprovação Global: Novos arquivos de hospedeiros e/ou novos conteúdos com o estado local=Pendente são definidos globalmente para o estado servidor=Aprovado no servidor enquanto os arquivos / conteúdos são criados / modificados.

Introdução Anticorpo, Arquivo Meta-Informação

Com referência particular à Figura 2, para ações que são permitidas, o servidor do sistema inclui uma base de dados anticorpo 36 que é utilizada primariamente para manter o rastreamento dos históricos de varreduras de arquivos e dos estados de aprovação para cada um destes arquivos. Um anticorpo é um bloco de dados de um arquivo (isto é, metadados ou meta-informações) que podem incluir alguns ou todos os seguintes campos:

- Tempo Primeiramente Visto. Quando o arquivo ou espalhamento foi visto pela primeira vez pelo hospedeiro e reportado ao servidor.
- ID de Arquivo. Um identificador único pára o arquivo, incluindo um ou mais espalhamentos de conteúdo tais como MD5, SHA-1, e OMAC.
- Tipo de Arquivo. A classe do arquivo (por exemplo, executável, roteiro, documento de escritório, arquivo, etc.). Esta é derivada do nome do arquivo quando ele foi visto pela primeira vez (veja abaixo) e também da análise do conteúdo do arquivo.
- Condição / Estado. O estado corrente do arquivo, incluindo Aprovado, Pendente, ou Banido.
- Método. A maneira pela qual o servidor aprendeu sobre o arquivo

(automaticamente, manualmente, etc.).

- Nome do Arquivo. O nome do arquivo, como foi visto pela primeira vez e reportado ao servidor. Este pode não ser o nome corrente do arquivo, mas é apenas o nome da primeira instância vista na rede.
- Caminho do Arquivo. O caminho do arquivo, como foi visto pela primeira vez e reportado ao servidor.
- Nome do arquivo do hospedeiro / caminho / extensão quando foi visto / reportado pela primeira vez.
- Nome do arquivo do hospedeiro / caminho / extensão quando foi visto / reportado pela última vez.
- Endereço IP do arquivo do hospedeiro quando foi visto / reportado pela primeira vez.
- Primeiro Hospedeiro Visto. O nome do hospedeiro no qual o arquivo ou espalhamento foi visto e reportado pela primeira vez.
- Resultados de Análises. O resultado das últimas varreduras ou outras análises.
- Primeira Análise. O tempo da primeira varredura / análise do arquivo.
- Última Análise. O tempo em que o arquivo foi escaneado / analisado pela última vez.
- Última Atualização. O tempo em que o estado do arquivo foi modificado pela última vez.
- Contêineres Pai. Links para outros arquivos que foram associados com o arquivo.
- Atributos dos Contêineres Pai. Nome do arquivo, tempo em que foi visto pela primeira vez, primeiro hospedeiro visto, caminho do arquivo, classificação de produtos, e estado de um arquivo contêiner associado.
- Contêineres Raiz. Links para outros arquivos que foram associados com o arquivo. Um contêiner raiz é um contêiner que não está contido em outro contêiner.
- Atributos dos Contêineres Raiz. Nome do arquivo, tempo em que foi visto pela primeira vez, primeiro hospedeiro visto, caminho do arquivo,

classificação de produtos, e estado de um arquivo de contêiner raiz associado.

- Contêineres Pai Referência, se conhecidos. Estes são utilizados para manter associações contidas tais como: "o arquivo deste espalhamento=y foi observado dentro de um arquivo do espalhamento=x".
- Tipo de conteúdo de arquivo (determinado por análise de conteúdo) tais como executáveis, arquivos roteiro, macros embarcadas.

O servidor tem um conjunto completo de anticorpos de sistema para o sistema. Enquanto que cada hospedeiro pode conter um subconjunto local de anticorpos na memória *cache* de usuário 64 e na memória *cache* de núcleo 82, o servidor é a autoridade que define e modifica determinados estados. Por exemplo, o servidor é a autoridade que centralmente inicia e propaga modificações (para Hospedeiros) incluindo transições de estado de Pendente para Aprovado ou Banido (estes três estados preferencialmente associados com espalhamentos de conteúdo), enquanto que os hospedeiros são as únicas autoridades que podem definir um estado para localmente aprovado. Cada entrada da base de dados 36 é persistente e preferencialmente é facilmente acessível com um índice de espalhamento de dados de arquivos. A base de dados pode ser opcionalmente indexada por outras chaves, tais como nomes de arquivos, datas da primeira visualização, estados, resultados de análises, ID de hospedeiros, ou contagens de hospedeiros, de tal forma que um administrador possa facilmente navegar pela base de dados anticorpo.

Enquanto que a base de dados com anticorpos é descrita como sendo localizada dentro ou no servidor, deve ser entendido que isto significa que a base de dados está associada com o servidor. Ela pode residir fisicamente no mesmo gabinete com as funcionalidades de processamento do servidor, ou ela pode residir em um gabinete diferente ou mesmo em uma localização remota. Se for em uma localização remota, devem existir conexões adequadas com ou sem fio para a obtenção dos dados.

Introdução ao Rastreamento de Anticorpos (AB)

À medida que novos arquivos são criados ou arquivos existentes são modificados, diretrizes de rastreamento podem ser disparadas,

com isto definindo uma cadeia de eventos de análises de arquivos e anticorpos. Primeiro, o hospedeiro executa uma série de etapas para determinar se houve uma significativa modificação de conteúdo que corresponda com o conteúdo que já foi analisado e para o qual um anticorpo já foi armazenado na memória *cache* do hospedeiro. Se o conteúdo anticorpo não estiver na memória *cache* do hospedeiro, o servidor é questionado para determinar se o servidor já fez a análise do conteúdo. Se o servidor não tiver um anticorpo correspondente, o conteúdo pode ser carregado para o servidor para análises adicionais. Até que o servidor possa determinar o estado definitivamente, o estado associado ao conteúdo é definido como sendo pendente ou ainda não determinado. O subseqüente acesso ao conteúdo pendente pode ser limitado. O servidor executa análises no conteúdo baseado em um tempo desde que o conteúdo foi visto pela primeira vez no servidor. Baseado nas análises ou em outras determinações externas o servidor pode definitivamente determinar mudanças no estado. Estas mudanças podem ser indicadas pelos hospedeiros para posterior recuperação, de tal forma que os hospedeiros possam atualizar suas memórias *cache* anticorpo com os estados modificados.

Rastreamento de Anticorpos do Hospedeiro

Com referência à Figura 3, o hospedeiro intercepta operações de arquivos (501), incluindo executar, ler, renomear, ou escrever, e providencia a operação para um filtro de operação de arquivos (502). Se o nome do arquivo não estiver na memória *cache* do núcleo e existir uma ausência de memória *cache* no núcleo (510) e se existir uma possível modificação do arquivo ou conteúdo (511), o estado é invalidado. O arquivo então vai a um extrator de conteúdo, que, como descrito com mais detalhes abaixo, extrai o conteúdo ativo de interesse (503) para produzir um arquivo reduzido, e fornece o arquivo reduzido a um assinador de conteúdo (504). O assinador de conteúdo aplica um espalhamento criptográfico, tal como MD5, ao arquivo reduzido. Este espalhamento é associado com o arquivo e com o nome do arquivo. Uma operação de arquivo pode ser atrasada / travada enquanto o espalhamento e outras análises (falha de resolução da memória *cache*) são completadas.

O hospedeiro também faz uma verificação local baseada

no conteúdo do espalhamento para tentar receber um estado (505). Se o conteúdo e o estado não forem encontrados o estado é definido como pendente. Isto pode significar que a operação de arquivo está autorizada a proceder, apesar de que monitoramentos adicionais, tais como registros, também podem ocorrer. Se o conteúdo for encontrado, o nome, o conteúdo, o contêiner (arquivo que continha o conteúdo ativo) e o estado são todos associados em conjunto (507). Se o conteúdo não for encontrado, o hospedeiro solicita que o servidor verifique pelo conteúdo em sua própria memória (506). Se ele for ali encontrado, o nome, conteúdo, contêiner (arquivo contendo o conteúdo ativo) e o estado são todos associados em conjunto (507). Se o conteúdo e o estado não forem encontrados, o estado é definido como pendente, e o conteúdo é carregado para o servidor (508), que confirma o carregamento (509). O servidor também pode procurar por um "super servidor" associado com um número dos servidores. Relacionamentos de contêineres são armazenados e associados com arquivos e outros contêineres. A informação de contêiner é também enviada a servidores e hospedeiros, bem como enviada para análises. Um "Contêiner Raiz" é um contêiner que não está contido em outro contêiner. Contêineres são identificados por seus arquivos associados bem como por espalhamentos criptográficos.

Geralmente, estados anticorpo são atribuídos a um espalhamento ou assinatura das partes "ativas" do conteúdo de um arquivo ou do conteúdo inteiro do arquivo. Assim geralmente, ESPALHAMENTO (Arquivo Dados / Conteúdos) → Estado. Isto mapeia Dados → Estado. O Estado (S) pode conter muitas partes de informações, tais como "aprovado" (lista branca) ou "banido" (lista negra) ou "pendente" (uma "lista cinza" tal como um arquivo recentemente visto e que ainda não foi completamente analisado).

Uma vantagem deste sistema é a combinação de nomes de estados com estados de conteúdos. Por exemplo, o servidor pode especificar e armazenar múltiplos nomes banidos, tais como *msblast.exe. O servidor armazena diretrizes de nomes de estado como listas de expressões regulares e meta-informações associadas. Qualquer drive / caminho / nome / extensão de arquivo que coincidir com a expressão regular irá então herdar o nome meta-informação. Esta informação é atualizada toda vez que nomes de arquivos são

modificados ou as especificações do nome meta-informação mudar. Estados de nomes e diretrizes são propagadas do servidor para os hospedeiros. Por exemplo, adicionando *msblast.exe → NomeBanir, o servidor irá sentir a nova diretriz / estado, e irá propagar esta especificação para os hospedeiros. Os hospedeiros irão então procurar em suas memórias *cache* de nomes de meta-informações, por coincidências com *msblast.exe, e aqueles arquivos que coincidirem herdarão o estado NomeBanir. O estado dos arquivos de um hospedeiro é uma superposição de estados de nome e data: por exemplo, se temp_msblast.exe tiver estado de conteúdo=Pendente, seu estado combinado é Banido pois NomeBanir tem precedência sobre Pendente. Estados de aprovação de nomes são tratados de uma maneira similar.

Anticorpos são armazenados nas bases de dados hierarquicamente. Existem 4 localizações principais de armazenamento para anticorpos como mencionado acima. Em um agente hospedeiro uma memória *cache* de núcleo anticorpo 82 mapeia arquivos NOME → ESTADO anticorpo. Por exemplo, NOME=c:\windows\bar.exe → ESTADO=aprovado. Resumidamente, este mapeamento é N → S. O núcleo pode e de fato reforça diretrizes baseadas no estado sem necessitar acesso ao conteúdo do arquivo. Isto é útil porque o arquivo pode ser encriptado no núcleo mas visível de forma não encriptada em nível maior. O núcleo tem acesso direto ao nome, mas não ao espalhamento. A memória *cache* do núcleo pode ser levemente consistente com outras memórias *cache*, e finalmente com o servidor, pelo fato de poderem existir longas latências (segundos, minutos, horas, dias).

O agente hospedeiro tem uma memória *cache* de usuário de nomes anticorpo (UN) e uma memória *cache* de usuário de dados anticorpo (UD) 60. O UN mapeia o nome do arquivo para um espalhamento dos conteúdos do arquivo (Dados), isto é, UN mapeia N → Dados. E, similarmente, o UD mapeia dados para o estado Dados → S. Geralmente, o mapeamento de N → Dados é muitos-para-um, e UN espelha a estrutura do sistema de arquivos local. O mapeamento de Dados → S é geralmente um-para-um, pois colisões de espalhamentos são raras com os espalhamentos fortes que são

preferencialmente utilizados, tais como MD5. As memórias *cache* UN e UD são também fracamente consistentes com o servidor, mas ambas UN e UD são fortemente consistentes com o sistema local de arquivos do hospedeiro, como é a memória *cache* do núcleo. UN e UD podem ser combinadas como segue: N → Dados → S = N → S.

Um servidor tem uma base de dados anticorpo 34 de cada espalhamento único que foi reportado de modo geral por qualquer um de seus hospedeiros, e um super servidor (se um existir) tem uma base de dados anticorpo de cada espalhamento único que foi visualizado de modo geral em qualquer um de seus servidores. A limitação a espalhamentos únicos limita o armazenamento e o processamento, apesar de que mais espalhamentos podem ser armazenados com conseqüentes melhorias no armazenamento e no processamento. Também, a limitação a espalhamentos únicos resulta em análises mais eficientes e menor tráfego de rede.

Geralmente, novos arquivos se propagam do hospedeiro para o servidor e para o super servidor em resposta a eventos "Novo Arquivo" e "Arquivo Sujo", e estados anticorpo recentemente computados se propagam em sentido reverso, do super servidor para o servidor, para o usuário do hospedeiro, para o núcleo do hospedeiro, na forma de atualizações anticorpo. Deste modo, anticorpos são centralmente controlados, gerenciados, e verificados. Os servidores "possuem" e certificam os anticorpos, e os servidores fornecem a autenticação de que os anticorpos não foram alterados ou forjados. Os hospedeiros mantêm seus próprios anticorpos que geralmente, mas não necessariamente, correspondem com aqueles do servidor. Assim um hospedeiro compromissado ou em mau funcionamento não consegue degradar uma coleção anticorpo de um servidor ou super servidor, nem pode um hospedeiro compromissado degradar os anticorpos de outros hospedeiros.

No hospedeiro, o estado anticorpo é preferencialmente armazenado de tal forma que ele não seja associado com espalhamentos / dados, mas sim pelo nome. O núcleo processa, interpreta, e reforça diretrizes, e o estado de um arquivo é verificado pelo seu nome. Deve ser entendido que a implementação preferida reforça diretrizes no núcleo, mas outras implementações

podem reforçar diretrizes no espaço do usuário. Ao verificar o estado, tanto no espaço do usuário como no núcleo, o que determina o estado resultante é de fato uma mistura. Por exemplo, se os dados anticorpo para foo.exe estão pendentes, mas o nome anticorpo foi banido baseado em seu nome, então o RecebeABEstado(foo.exe) retorna um resultado de "banido pelo nome". Existe uma diretriz em separado para bloquear execuções de arquivos com o estado anticorpo = NomeBanir. As ações para esta diretriz são parametrizadas pelo valor de D como mencionado acima. Uma diferença é que as diretrizes que bloqueiam "Banido pelo Nome" estão ativas em níveis mais baixos de segurança D. Por exemplo, em D=4, arquivos "pendentes" serão executados (com alarme silencioso) mas arquivos banidos não serão executados.

Eliminadores de nomes são representados como uma lista de expressões regulares e podem incluir um curinga (*), por exemplo, "*oo.exe" ou "*msblast.exe", no servidor. Estas listas têm números de versões. À medida que os hospedeiros fazem sondagens, eles verificam seus números de versões. Quando um hospedeiro detecta um desacordo, ele envia uma requisição RecebeNomeBanir para o servidor (isto é, o hospedeiro puxa os novos dados banir do servidor). Estas expressões regulares são então reavaliadas contra os anticorpos nome. O nome Banir é um atributo de estado, e somente deve ser recomputado quando a listagem de nomes banir mudar ou quando o nome do arquivo mudar. A lista de curingas não necessita ser comparada a cada operação de arquivo. Assim a natureza dupla dos dados anticorpo e dos nomes anticorpo é útil. Além disso, centenas ou milhares de expressões regulares de nomes podem estar simultaneamente ativas sem requerer milhares de expressões regulares de coincidências computacionais no núcleo para cada operação de arquivo, o que poderia ser extremamente caro.

Rastreamento de Conteúdos de Arquivos

Com referência à Figura 2, uma função interceptar / bloquear 86 pode interceptar e ler requisições de acesso a arquivos. Ela pode suspender requisições enquanto obtém informações de diretrizes, requisições de bloqueio baseadas em diretrizes internas de núcleo, e retornar códigos de erros apropriados para requisições bloqueadas. A função 86 lê das requisições de

acesso a arquivos o nome do processo requisitado, horário local do sistema da requisição, o arquivo requisitado (incluindo o caminho completo), e a ação requisitada (por exemplo, ler, escrever, ou executar). Em outra modalidade preferida de execução, a função 86 encaminha todas as requisições de acessos a arquivos ao "filtro de estado" 88 e todas as operações são bloqueadas até que o filtro 88 retorne um sinal indicando que a operação foi bloqueada ou permitida.

O filtro 88 intercepta as requisições de acesso a arquivos da função 86 e retorna uma ação de "bloquear" ou "permitir" para a maioria das requisições de acesso a arquivos. Qualquer requisição de acesso a arquivos que não possa ser associada com requisições de acesso a arquivos já aprovados é encaminhada aos disparadores do núcleo e módulos de ação 90, que retornam uma ação "bloquear" ou "permitir". Esta ação é armazenada pelo filtro 88, e é preferencialmente retornada para a função 86 para qualquer requisição de acesso a arquivos associada e similar.

O filtro 88 mantém uma memória *cache* 84 dos arquivos já abertos (indexados por um identificador único de núcleo; por exemplo, um manipulador de arquivos de núcleo no Windows NT). Cada entrada da memória *cache* contém um identificador de arquivo (manipulador de arquivos de núcleo) e bloqueia ou permite permissões para uma leitura, escrita ou execução.

Se múltiplos processos acessarem o mesmo arquivo, cada um deles terá a sua própria entrada de memória *cache*. Se um dado processo tentar um acesso a um novo arquivo, o filtro experimentará uma falta de memória *cache* para este arquivo, o que fará com que a requisição de acesso ao arquivo seja submetida aos disparadores e módulos de ação. A identificação para a operação requisitada (leitura, escrita, ou execução) deve ser definida em "permitir" se o módulo 90 o permitir. De outro modo, ele deve ser definido em "bloquear". Se um processo que obteve apenas um tipo de permissão (por exemplo, leitura) então tentar outro tipo de acesso (por exemplo, escrita), o módulo 90 será novamente contatado.

As entradas da memória *cache* cuja idade excedem um determinado valor (por exemplo, 60 segundos) podem ser apagadas. Isto permite a seleção de entradas que por alguma razão não foram removidas. Isto também

permite reavaliações periódicas de um arquivo pelo módulo 90.

Neste exemplo, uma operação de escrita de arquivo é capturada no núcleo em uma camada de bloqueio 86 pelo programa de núcleo do hospedeiro agente (HK) para um arquivo "foo.exe". Em um valor de D=4, a operação de arquivo, aqui uma operação de escrita de arquivo, é capturada por uma diretriz ativada de "rastreamento sujo", e isto define um evento "sujo" do programa núcleo do hospedeiro para o programa do espaço do usuário do agente hospedeiro (HU). Este evento especifica o nome do arquivo e a operação suja. A memória *cache* de núcleo 82 não é consultada para esta operação, e a diretriz de rastreamento sujo tem este campo anulado.

HU então executa uma variedade de operações de análises locais em arquivos e processamento de eventos 66, e analisa o motor 68 no foo.exe. Primeiro, foo.exe é verificado para confirmar que se ele existe, se ele pode ser lido, e se em realidade é um executável. Outras operações podem ser executadas, tais como a extração de "dados de interesse" no filtro 88; por exemplo, comentários de roteiros podem ser removidos se o arquivo for foo.bat. Os dados extraídos de foo.exe são então criptograficamente espalhados, e este espalhamento é utilizado para tentar uma verificação na memória *cache* anticorpo HU 60. Se o nome e os dados já existirem, nada mais é feito. Se o nome é novo mas os dados forem conhecidos, então um novo nome anticorpo é criado na memória *cache* UN. Este processo todo faz parte do que é chamado de "fila de análises do Estágio 1". Muitos arquivos podem ser colocados à espera em fila para serem espalhados na fila do Estágio 1 do hospedeiro. A fila do Estágio 1 tem apenas nomes anticorpo e meta-informações, pois os dados ainda não são conhecidos nem foram analisados.

Se o hospedeiro tiver visto estes dados de arquivo e espalhamentos, então a correspondente e conhecida meta-informação deste espalhamento é associada com a meta-informação do arquivo do hospedeiro para este arquivo, recuperada da memória local UD ou de memórias locais de discos, nesta ordem. Se o hospedeiro não viu estes dados, a memória *cache* UD "falhou". O espalhamento é colocado em uma fila de análises do Estágio 2. Na realidade, existem dados anticorpo, isto é, estados com dados lógicos de rastreamento, tais

como "Aprovado", "Banido", ou "Pendente", e também existem anticorpos Nome, por exemplo "Banido por Nome". Por exemplo, se o servidor banir "*oo.exe", então o anticorpo nome para foo.exe indicará "NomeBanido" e diretrizes de banimento de nomes podem ser bloqueadas baseadas nisso. Desta forma, mesmo que as memórias *cache* possam saber que foo.exe já foi banido (por nome), a resolução de rastreamento sujo continua. Esta distinção dos anticorpos nome e dados é de escopo local para os hospedeiros individuais, mas se torna importante para a função AchaArquivo (descrita abaixo) e para o reforço das diretrizes. O anticorpo dados é colocado portanto na fila do Estágio 2.

As análises do Estágio 2 tentarão primeiro resolver informações de estados locais de memórias *cache*, depois de armazenadores de dados locais baseados em discos e finalmente do servidor. Se o servidor estiver conectado, a fila do Estágio 2 será esvaziada à medida que as meta-informações são resolvidas. Quando foo.exe é removido desta lista, o servidor é perguntado se ele viu este espalhamento de dados, se este espalhamento não for encontrado localmente. Se a resposta for não, foo.exe, o seu espalhamento e outras meta-informações são colocadas na fila do Estágio 3 para carregamento no servidor. Adicionalmente, o servidor envia um estado anticorpo padronizado para o hospedeiro, que está em situação "pendente", caso o servidor não tenha visto o espalhamento ou a análise do servidor não tiver sido completada suficientemente para determinar outros estados. Se o servidor já tiver computado um anticorpo e um estado válido, ele retorna esta meta-informação de anticorpo. Se o servidor nunca tiver visto estes dados para foo.exe, é novo no sentido de que todas as máquinas, na experiência do servidor, nunca viram este arquivo.

Quando foo.exe é removido da fila do Estágio 3, ele é enviado ao servidor utilizando transferência encriptada de uma via. Isto é, utilizando FTPS (protocolo de transferência segura de arquivos) e um diretório apenas de escrita no servidor, onde arquivos podem ser transferidos para o servidor mas não dele descarregados. Quando a transferência tiver sido completada com sucesso, o hospedeiro informa ao servidor que foo.exe foi transferido. Esta transferência é referenciada como espalhamento, de tal forma a minimizar o vazamento de informações e para proporcionar segurança adicional.

Quando o servidor entender que foo.exe foi transferido, ele inicia a análise do arquivo através de diversos estágios, da mesma maneira como o hospedeiro faz. Neste caso um novo anticorpo é criado, e o servidor utiliza seu relógio verificado e sincronizado para datar sua primeira aparição. Além disso, são executados a extração e o espalhamento, e seus resultados são sobrepostos aos do hospedeiro.

As análises do servidor seguem uma programação que é especificada e armazenada no servidor. Esta programação se relaciona com o tempo da primeira aparição do arquivo ou de seu espalhamento no servidor. Por exemplo, se um arquivo chegar ao meio dia e a programação for "verificar espalhamento em +o e varredura AV em +0 e varredura AV em +2 horas", então ao meio dia, o espalhamento do arquivo será computado e verificado utilizando um serviço externo de verificação de espalhamento. Então, uma varredura AV é executada. Duas horas após as 14 horas, outra varredura AV deste arquivo é executada. Outra maneira de descrever a programação, é que ela é relativa à "idade do arquivo no servidor".

Quando um anticorpo muda de estado no servidor, um valor de um contador incremental é escrito para o anticorpo. Este contador é utilizado para selecionar apenas a faixa de anticorpos que mudou desde que qualquer hospedeiro particular ou super servidor fez a verificação. Por exemplo, se uma modificação prévia de anticorpo foi glorp.bat mudando de Pendente → Aprovado e a versão do contador global anticorpo foi 277, o anticorpo do servidor correspondendo com o espalhamento de glorp.bat receberá o número de versão 277 e o contador será 278. Assim sendo, o número da versão correspondendo ao anticorpo foo.exe é 278 e o do contador é 279.

Quando os hospedeiros periodicamente sondam, eles fornecem seu último número de versão anticorpo, e o servidor envia todos os anticorpos que sofreram modificações desde a última sondagem. Preferencialmente, o servidor envia o número corrente, e quando o hospedeiro detecta a desconformidade, ele solicita ao servidor uma atualização anticorpo, e a lista de dados de anticorpos é retornada. Estes são então juntados aos anticorpos do hospedeiro, e modificações também são enviadas para dentro do núcleo.

Apesar do hospedeiro poder receber e armazenar alguns anticorpos para dados que ele nunca viu, geralmente apenas aqueles anticorpos que correspondem com arquivos existentes no hospedeiro são juntados. Os outros são geralmente descartados. O servidor coloca na memória *cache* os últimos poucos minutos de atualizações, para minimizar o efeito da adaptação customizada de todas as atualizações para cada hospedeiro. Novamente, uma vez que hospedeiros geralmente recebem mais anticorpos do que eles necessitam, e porque novos anticorpos são raros, este tráfego é limitado. Atualizações anticorpo são pequenas, pois são geralmente de outras mensagens.

Anticorpos podem permanecer sincronizados com um super servidor de um modo similar. Aqui, o super servidor pode sondar servidores e receber listas de atualização de anticorpos. O super servidor pode juntá-las, e enviar atualizações personalizadas para cada servidor. Estas atualizações são todas fracamente consistentes, pois elas podem estar atrasadas em minutos ou dias, mas devem existir intertravamentos e salvaguardas para evitar "buracos" nas atualizações.

Existem outros aspectos e recursos relacionados à combinação de anticorpos. Por exemplo, alguns servidores podem não aceitar determinadas atualizações de anticorpos do super servidor. Também, hospedeiros podem não permitir determinadas mudanças nos estados locais de acordo com determinados estados especificados pelos servidores.

Um problema está relacionado com os estados iniciais da memória *cache* e com as diretrizes iniciais. A memória *cache* do servidor pode ser pré-carregada com espalhamentos de anticorpos conhecidamente bons e ruins, ou ela pode estar vazia, e tudo está bem. Entretanto, hospedeiros devem ocasionalmente "Carregar Devagar". Por exemplo, quando um hospedeiro primeiro conecta a um determinado servidor, este fato é detectado, e o hospedeiro irá executar um carregamento devagar no qual cada arquivo único de interesse do sistema de arquivos do hospedeiro é inserido dentro da fila do Estágio 1. Um valor especial de D é envolvido durante este processo para assegurar que a memória *cache* indeterminada não irá causar problemas. Todos os anticorpos geralmente iniciam com o estado "pendente", e eles lentamente

sincronizam com o servidor. Também, todos os anticorpos dos hospedeiros, informações de filas e aspectos globais relacionados são persistidos periodicamente sobre as reinicializações.

Consistência da Memória *Cache* do Núcleo

Durante o *boot* ou outra inicialização do agente hospedeiro, o núcleo é carregado com todos os anticorpos válidos e conhecidos do espaço do usuário, para cada arquivo do hospedeiro conhecido e existente que tiver meta-informações válidas. Algumas atualizações anticorpo são enviadas para dentro do núcleo assim que elas são recebidas do servidor ou de listas de análises do espaço do usuário. No entanto, algumas atualizações são resultado de ausências de memória *cache* no núcleo. Se uma diretriz foi determinada para estar ativa, e o estado anticorpo é necessário, e se este estado não estiver disponível, o núcleo irá geralmente travar a operação por algum tempo e enviar um evento de falha de núcleo para o espaço do usuário. Alguns eventos podem ser travados mesmo que o anticorpo não seja necessário. Este é o caso quando uma diretriz permitir que o usuário do hospedeiro sobreponha um estado restritivo (Pendente) através da interação com uma interface de usuário (janela de mensagem *popup*), por exemplo, clicando sim para sobrepor uma operação Pendente bloqueada e para causar operações subseqüentes restritas, para prosseguir sem bloqueio por algum tempo.

Em um exemplo, um programa de instalação desempacota um novo programa chamado inst.exe e então o renomeia e executa. O núcleo evita as inconsistências temporárias atrasando o renomear e atrasando a execução enquanto a análise está em curso. O anticorpo resultante é enviado para baixo assincronicamente, do espaço do usuário, e então as operações pendentes são desbloqueadas e a diretriz é avaliada com a necessária informação de estado, tão logo a atualização assíncrona tenha sido completada.

A memória *cache* do núcleo contém anticorpos para praticamente todos os arquivos do sistema de arquivos na inicialização. Operações que podem deixar buracos na memória *cache* do núcleo ou outras inconsistências, mesmo por curto espaço de tempo, são atrasadas e intertravadas de tal forma que a consistência é mantida. As memórias *cache* do espaço do

usuário são otimizadas para resolver falhas de núcleo com latências muito baixas. Enquanto que o núcleo e as memórias *cache* do espaço do usuário são relativamente insensíveis a latências do lado do servidor, a memória *cache* do núcleo é sensível a intertravamentos e a persistências adequadas.

AchaArquivo

Em função das memórias *cache* UN e UD serem preferencialmente otimizadas para verificações de baixa latência, estas memórias *cache* podem ser utilizadas como parte de uma lista distribuída de anticorpos do servidor, referenciada aqui como sendo a função "AchaArquivo", para produzir uma visualização de quais arquivos estão em quais hospedeiros. Uma requisição AchaArquivo pode ser especificada por um administrador submetendo um formulário de navegador *web* através de uma interface *web* em um servidor ou super servidor. Por exemplo, os seguintes qualificadores podem ser conjuntamente especificados:

(1) uma especificação regular de um padrão de expressão para o nome do arquivo,

(2) uma especificação regular de um padrão de expressão para o caminho do arquivo,

(3) um espalhamento de conteúdos de interesse de um arquivo,

(4) um espalhamento ou outro ID de um contêiner que está associado com um arquivo,

(5) um espaço de tempo no qual um arquivo ou o espalhamento do arquivo foi visto pela primeira vez pelo hospedeiro,

(6) nome do hospedeiro,

(7) endereço IP do hospedeiro,

(8) tipo do arquivo

(9) um ou mais estados de arquivo associados com o arquivo a partir de um conjunto de pelo menos 3 estados: aprovado, banido, análise pendente. Por exemplo, um conjunto TodosBanidos=(NomeBanir, BanirPorEspalhamento).

(10) se determinadas operações de arquivo foram

executadas pelo hospedeiro no arquivo, e

(11) um grupo de hospedeiros.

Com referência à Figura 4, uma requisição AchaArquivo completada é análoga a um e-mail no qual o servidor postou uma requisição de recuperação posterior a hospedeiros determinados. À medida que os hospedeiros se identificam, eles são informados se existirem mensagens AchaArquivo do servidor esperando por eles. Quando o hospedeiro é informado que ele tem uma requisição AchaArquivo pendente, ele recupera a requisição utilizando ReceberRequisiçãoAchaArquivo, como mostrado em linhas (1) na Figura 4. Em outras palavras, a requisição é preferencialmente executada como uma "puxada" do servidor. Isto permite uma implementação mais segura sem a necessidade de soquetes de escuta de hospedeiros.

Cada um dos hospedeiros conectados processa sua requisição AchaArquivo através do acesso de dados aplicáveis de sua memória *cache* anticorpo, e posta as listas de resultados em uma base de dados de resultados mostrados como PosteResultadosAchaArquivo (linhas (2) na Figura 4), incluindo algumas ou todas das seguintes informações para cada arquivo retornado:

(1) um nome de arquivo,

(2) um caminho de arquivo,

(3) um espalhamento de conteúdos de interesse de um arquivo,

(4) um tempo quando um arquivo ou o espalhamento do arquivo foi visto pela primeira vez pelo hospedeiro,

(5) o nome do hospedeiro,

(6) o endereço IP do hospedeiro,

(7) o tipo do arquivo,

(8) a informação contêiner para o arquivo,

(9) um ou mais estados de arquivos de hospedeiro associados com o arquivo de um conjunto de pelo menos 3 estados: aprovado, banido, análise pendente,

(10) se determinadas operações de arquivo foram

executadas pelo hospedeiro no arquivo, e

(11) um grupo de hospedeiros.

Em uma implementação, todas as comunicações hospedeiro – servidor (não apenas AchaArquivo) são executadas pelo hospedeiro primeiramente se conectando ao servidor e enviando uma ou mais mensagens de rede, e recebendo respostas do servidor para as mensagens do hospedeiro, antes de desconectar. Novamente este processo tem a vantagem de ser mais seguro porque nenhum soquete de escuta de hospedeiro é necessário. Existe uma vantagem adicional que reside no fato de que apenas o endereçamento e as rotas do servidor necessitam ser mantidas, ao invés de manter o endereçamento do hospedeiro, as rotas, e de reduzir a necessidade da descoberta de tais informações de hospedeiros.

O servidor junta e constrói uma lista mestre dos resultados das listas AchaArquivo dos hospedeiros. A união destas listas é a resposta completa para a requisição AchaArquivo, e é construída durante um tempo, geralmente se completando em menos de um minuto. Uma vez que o processamento local do hospedeiro somente acessa as memórias *cache* anticorpo, e não o sistema de arquivos do hospedeiro, estas consultas podem ser rápidas. O nome duplo, o sistema de associação de dados anticorpo e as memórias *cache* permitem isto. O servidor então exporta os resultados para o administrador, por exemplo, através de uma interface *web*. Também, determinados resultados AchaArquivo podem afetar e disparar o SNMP, os registros de atividades de sistema, os alarmes, e outros sistemas de notificação.

O super servidor também pode postar requisições para serem acessadas por servidores de um modo similar, ou um super servidor pode submeter requisições AchaArquivo diretamente para os servidores. Os servidores, então, podem retornar os resultados juntados para o super servidor, que pode então juntar estes em um resultado mestre ainda maior. Isto é similar ao relacionamento entre servidores e hospedeiros durante o processamento de uma requisição AchaArquivo.

Análises Centrais Disparadas por Tempo

Com referência à Figura 5, um servidor pode executar

análises baseadas em eventos, por exemplo, uma análise a cada vez que o hospedeiro enviar um conteúdo, ou o sistema pode executar estas análises baseadas no tempo. Como mencionado acima, o novo conteúdo pode ser enviado para o servidor, e análises são executadas com agentes de análises externos e/ou internos para criar metadados ou meta-informações que são armazenadas em uma base de dados. O sistema pode então verificar por análises programadas adicionais, por exemplo, depois de determinados intervalos de tempo relativos a uma primeira observação de um arquivo quando o novo conteúdo é transferido. Os servidores e super servidores podem executar muitos tipos de análises adicionais baseadas em tempo.

Com referência à Figura 6, à medida que um arquivo é visto pela primeira vez e seu anticorpo é adicionado a uma base de dados do servidor, o efeito é o mesmo como se um temporizador fosse iniciado para cada arquivo. Assim, por exemplo, os intervalos de tempo podem ser (t=0=imediato, t=12 horas depois, t=2 dias depois, e t=30 dias depois da primeira visualização ou relatório para o servidor), e podem ser baseados no relógio do servidor. Ações periódicas, além de ações únicas de tempo em tempo, podem ser especificadas. Como mostrado aqui, varreduras antivírus (AV) e anti-software espião (AS) podem ser executadas em diferentes tempos, e outras análises podem ser executadas. Para intervalos de tempo posteriores, isto pode ser uma comparação com outros servidores que possam ter verificado os mesmos arquivos. Tipicamente, as análises posteriores podem ser baseadas em todos os arquivos primeiramente vistos dentro de um determinado período de tempo. Por exemplo, todos os arquivos primeiramente vistos dentro do tempo de 1 hora, receberão a análise das 12 horas, a 12 horas do último arquivo no período de tempo.

Com referência à Figura 7 o sistema seleciona arquivos para análise e envia os arquivos para executar as análises especificadas. Para cada intervalo de tempo podem ser especificadas diferentes operações. Uma vez que os arquivos são mantidos por um período de tempo nos servidores, estas análises ativadas em tempo podem proceder independentemente do Hospedeiro original continuar conectado ou não. Exemplos de análises de servidores com tempo central que podem ser executados, incluem:

(1) Computar espalhamentos alternativos (por exemplo, utilizando algoritmos MD5 ou SHA1), verificar os espalhamentos reportados, e armazenar todos os espalhamentos.

(2) Autenticar e assinar conteúdos com as credenciais do servidor ou com outras credenciais de terceiros.

(3) Verificar espalhamentos contra bases de dados ruins conhecidas (lista negra) ou localmente ou via consulta a outro servidor.

(4) Verificar espalhamentos contra bases de dados boas conhecidas (lista branca) ou localmente ou via consulta a outro servidor.

(5) Verificar espalhamentos contra bases de dados conhecidas de classificação de produtos para identificar o produto (e outras informações) que correspondam com o espalhamento do arquivo.

(6) Enviar arquivos para varredura de vírus (por exemplo, por FTP ou SMTP como anexos MIME) ou executar localmente.

(7) Enviar arquivos para varredura de software espião como em (4) ou executar localmente.

(8) Enviar arquivos para análises personalizadas em sites específicos como em (4) ou executar localmente.

(9) Exportar arquivos para um subdiretório especial com acesso-restrito-à-rede em um servidor de arquivos de rede (por exemplo, Samba autenticado ou FTPS).

(10) Enviar sinalizações SNMP informando que novos arquivos necessitam análise, e especificar suas localizações.

(11) Enviar registros de atividade de sistema ou mensagens de e-mail informando que novos arquivos necessitam análise, e especificar suas localizações.

(12) Controlar determinados diretórios para verificar se um outro sistema aprovou ou desaprovou o arquivo.

(13) Executar análises personalizadas no servidor.

(14) Executar automaticamente uma segunda análise condicionada aos resultados de uma primeira análise.

(15) Receber mensagens de rede autenticadas contendo

resultados de análises de sistemas de análises externos.

Os resultados das análises acima são resumidos no servidor, que atualiza o estado no armazenador de meta-informações (124), particularmente o estado para a difusão para os hospedeiros. O servidor faz recomendações quanto a um arquivo ser aprovado ou banido. A informação é resumida para que administradores possam aprovar o banir grupos de arquivos com uma ação de um navegador *web*. Opcionalmente os resultados das análises acima podem ser utilizados para aprovar ou banir automaticamente arquivos com determinados anticorpos. O servidor pode providenciar relatórios, alarmes, ou outras informações, e pode alterar o valor paramétrico de D para todos ou um ou mais grupos de hospedeiros. O servidor sinaliza as mudanças de estado para posterior distribuição através de atualizações (130), preferencialmente de uma maneira que os hospedeiros puxem a atualização do servidor.

Análise Anticorpo / Serviços de Aprovação

Uma vez que o sistema se concentra em novos arquivos, serviços terceirizados de análises de arquivos podem se tornar práticos e úteis. Estes serviços podem ser automatizados (por exemplo, com chamadas de serviços SOAP/*Web*) ou manuais (seguir links autenticados para os servidores de um provedor de serviços). Estes serviços, que podem ser executados localmente ou em locais externos utilizando servidores remotos, podem incluir:

(1) Entrar um espalhamento manualmente ou seguir um link *web* pré-computado para receber resultados de consultas de verificações de bases de dados conhecidas como boas e ruins.

(2) Encontrar anticorpos relacionados a um anticorpo particular (por exemplo, grupos de arquivos associados com a mesma aplicação ou aplicações similares).

(3) Identificar o vendedor e a aplicação associada com o espalhamento.

(4) Descobrir quantas companhias e computadores têm este arquivo e por quanto tempo. Estas companhias não são identificadas por nome, apenas contadas. O provedor de serviços recebe esta informação confidencialmente como parte do serviço. O provedor de serviços cria uma base

de dados duplamente cega dos resultados e do serviço.

(5) Descobrir quantas companhias baniram ou aprovaram o arquivo, e quais arquivos eles aprovaram junto com ele. Novamente, todos eles são cegos e feitos por espalhamento, da perspectiva do usuário final. O provedor de serviços não necessita receber ou armazenar nomes de arquivos ou dados de arquivos, apenas a meta-informação na forma de anticorpos. De fato, nomes de arquivos, e certamente os próprios arquivos devem ser considerados como informação proprietária.

(6) Aprovação automática pelo lado do servidor baseada no resultado das consultas acima, bem como baseadas também nas análises do servidor.

Extrator de Conteúdo (CE)

O conteúdo geralmente forma grupos ou pacotes de conteúdo. Exemplos disto incluem programas executáveis e vírus dentro de arquivos .zip ou macros dentro de documentos Microsoft Office (por exemplo, Word, Excel e arquivos Power Point), ou arquivos dentro de pacotes de instalação tais como arquivos Microsoft .msi. Com referência à Figura 8, um arquivo é recebido e o extrator de conteúdo verifica tipos embarcados de conteúdo, por exemplo, macros dentro de um documento Office. Preferencialmente apenas estes tipos "ativos" de conteúdos são extraídos.

Após detectar uma possível modificação de arquivo (600) ou de estado desconhecido, o extrator de conteúdo toma as porções extraídas e as converte em um tipo de arquivo de conteúdo válido, por exemplo, um arquivo Word (.doc) sem textos ou figuras, para reempacotá-los. Este processo é ilustrado nas etapas 600-605. O arquivo reempacotado resultante é geralmente muito menor que o arquivo original (o "contêiner") que é referenciado como sendo uma "redução". Um espalhamento da redução é computado (603) e os espalhamentos da redução são associados com o espalhamento do contêiner (604). Contêineres podem ser aninhados e suas associações são também rastreadas. Posteriormente, se o conteúdo necessita ser transferido, apenas as reduções são transferidas. Opcionalmente, o arquivo contêiner e suas meta-informações podem ser transferidos baseado no resultado da análise da extração. Contêineres raiz e

suas meta-informações podem ser transferidos baseado no resultado da análise da extração. Por exemplo, um arquivo set up.exe contém um arquivo main.cab, que por sua vez contém um arquivo install.exe. Em relação ao install.exe o arquivo main.cab é o contêiner pai para o arquivo install.exe, e o arquivo setup.exe é o contêiner raiz para o arquivo install.exe bem como o contêiner pai para o arquivo main.cab. Todas estas associações são armazenadas e preferencialmente salvas como relacionamentos entre os espalhamentos dos arquivos individuais.

Este processo reduz o tráfego de rede e as áreas de cobertura dos estágios de análises, e ele permite apenas o rastreamento do conteúdo embarcado e não de macros associadas com outros arquivos (por exemplo, modelos de documentos herdados). Isto não é verdade em métodos que interceptam macros durante o seu carregamento. O extrator permite a detecção e o rastreamento de macros embarcadas, independentemente da localização.

O reempacotamento de reduções como outros tipos de arquivos válidos tem a vantagem que as reduções são compatíveis com sistemas de análises de terceiros, por exemplo, macros reempacotadas como pequenos documentos Word podem ser enviadas como anexos de e-mail para uma porta de varredura de vírus de e-mails. Outro exemplo é um arquivo zip, temp.zip, contendo 5 arquivos, apenas um dos quais é ativo, foo.exe. A redução de temp.zip pode ser um arquivo zip chamado foo.zip com apenas foo.exe dentro dele, ou a redução pode ser o próprio arquivo foo.exe. A assinatura de foo.zip ou a assinatura de foo.exe é preferencialmente associada como a assinatura correspondendo ao temp.zip. A redução pode novamente ser enviada por e-mail para uma porta AS de varredura de e-mails. Alguns contêineres são isentos de conteúdo ativo, e como tais podem não ser rastreados. Existem vantagens de eficiência no rastreamento de reduções, mas também existem vantagens em detectar e analisar somente o novo conteúdo. Deste modo, estatísticas mais exatas, alarmes, e análises podem ser produzidas. A detecção automática e especificamente antecipada de conteúdo não classificado, tais como arquivos de estado Pendente, permite diretrizes poderosas e gerenciamentos de conteúdo.

Interface de Usuário do Servidor

A interface de usuário do servidor proporciona um número de "painéis", cada um dos quais permite a configuração e o gerenciamento de um aspecto diferente do sistema. Nesta seção o termo "usuário" é utilizado para indicar um administrador que tem acesso à interface de usuário do servidor. A interface de usuário pode ser acessível através de um navegador *web* padronizado, através de uma conexão SSL criptografada. A autenticação e o controle de acesso são providenciados para manter a integridade do servidor e para determinar o nível de privilégio de um usuário em particular.

Quando o primeiro usuário acessa o sistema, o usuário é autenticado e a ele é atribuído um nível de privilégio baseado nesta autenticação. Este nível de privilégio determina se o usuário tem permissão de acesso ilimitado ou acesso apenas de leitura; menor granularidade de acesso também pode ser proporcionada. Ações de usuários são rastreadas pelo nome do usuário e pelo tempo. Certificados instalados no servidor podem ser utilizados para controlar e encriptar tanto o acesso à interface do usuário e também para providenciar assinaturas para e possíveis encriptações de informações retornadas aos servidores. Estes certificados podem ser instalados e atualizados em um painel de manutenção. Toda a entrada para a interface deve ser apropriadamente validada para assegurar que o servidor está fornecendo informações corretas para os hospedeiros em suas configurações.

Uma interface de estado de rede proporciona uma visão geral do sistema em execução, incluindo: eventos recentes e informações associadas, incluindo identificadores únicos de arquivos, datação de eventos, tipos de eventos, prioridade de eventos , tipos de arquivos e nomes, e sistemas hospedeiros identificados por ambos os nomes e os identificadores únicos. A interface também fornece um resumo de informações do estado do sistema durante certos períodos de tempo (por exemplo, última hora, último dia). Informações mais detalhadas encontram-se disponíveis em um painel de estatísticas. As informações mostradas aqui incluem os números de novos executáveis detectados, novos roteiros detectados, arquivos com novo conteúdo embarcado, arquivos não aprovados, e arquivos infectados.

Um painel de estatísticas mostra as estatísticas mais

detalhadas coletadas pelo sistema. Esta informação inclui o número dos seguintes eventos em vários períodos de tempo (por exemplo, última hora, últimas 24 horas, última semana). Ela pode incluir, por exemplo, o número de novos executáveis vistos na rede, novos roteiros, arquivos como novo conteúdo embarcado, novos arquivos *web* (HTML, ASP, etc.), arquivos que necessitam de aprovação, manualmente ou por varredura, arquivos aprovados pelo processo de varredura, arquivos aprovados manualmente ou através de auto-aprovação, arquivos que não estão passando por uma varredura, arquivos que são conhecidamente infectados e que foram bloqueados, executáveis que são banidos e foram bloqueados, total de eventos processados pelo servidor desde que ele foi instalado, e eventos desde a última reinicialização.

Junto com as estatísticas de cada categoria, o usuário pode visualizar "listas top dez" de um item realçando as instâncias mais freqüentemente vistas de cada um, através de todos os hospedeiros gerenciados pelo servidor. Exemplos de listas Top10 incluem arquivos top dez recentemente descobertos, classificados pela contagem de quantos hospedeiros têm pelo menos uma cópia do arquivo, com variantes desta lista incluindo contagem-por-espalhamento-único, contagem-por-nome-único-de-arquivo, contagem-Banido-por-espalhamento, contagem-Banido-por-nome, contagem-recentemente-banido, contagem-recentemente-atualizado/modificado, contagem-por-grupo/contêiner/contêiner-raiz/produto-único. Listas Top10 são atualizadas e exportadas via SNMP. Um painel de configuração pode ser utilizado para configurar alarmes e respostas automáticas baseadas em contagens Top10 e outras variáveis atualizadas. Alarmes incluem registros de atividades, armadilhas SNMP, mensagens de registro de sistemas, notificações de e-mail e outras mensagens de rede. Respostas incluem banir arquivos, aprovar arquivos, mudar o parâmetro D para um ou mais grupos de hospedeiros, mudar a diretriz para um ou mais grupos de hospedeiros, mudar a atribuição do grupo de hospedeiros para um ou mais hospedeiros, e analisar arquivos.

O painel de estatísticas também mostra informações generalizadas sobre o sistema, incluindo: o número total de hospedeiros servidos por este servidor, divididos em ativos e inativos (um hospedeiro inativo é um que

não contatou o servidor recentemente); o número total de anticorpos na base de dados do servidor; e o tempo ativo, isto é, quanto tempo o sistema tem estado ativo desde a última reinicialização.

A informação estatística exibida neste painel também está disponível através de consulta via SNMP (Protocolo Simples de Gerenciamento de Rede) ao servidor, permitindo a integração com sistemas de gerenciamento de redes.

Um painel de plotagem permite que o usuário plote e imprima gráficos e tabelas de atividades recentes. Este painel pode ser combinado com o painel de estatísticas. A informação de plotagem também pode estar disponível em formato XML para exibição em aplicações externas. Exemplos de gráficos que podem ser plotados incluem a atividade durante um determinado período de tempo (uma hora por minuto, uma semana por hora, etc.), ou exibições gráficas da "lista top dez".

Podem existir algumas limitações na variedade de plotagens disponíveis, em função das restrições nas informações estatísticas retidas pelo servidor. Onde um administrador estiver utilizando um sistema de gerenciamento SNMP, também pode ser possível providenciar plotagens de estatísticas em um formato que já está em uso dentro da organização.

O painel da base de dados anticorpo permite que o usuário interaja diretamente com a base de dados anticorpo armazenada no servidor. O conteúdo da base de dados é exibido e o usuário pode escolher a classificação da exibição por diferentes critérios, ou limitar a exibição através da escolha de padrões de filtros. O usuário também pode interagir com os próprios anticorpos; estas operações serão detalhadas abaixo.

O servidor pode utilizar uma base de dados informacional auxiliar, que inclui campos que não são requeridos na base de dados anticorpo principal. Um exemplo de campos desta base de dados pode ser o primeiro nome de arquivo visto ou a classe inicial do arquivo.

Para cada arquivo, a seguinte informação é exibida neste painel:

- Tempo Primeiramente Visto. Quando o arquivo ou espalhamento foi visto

pela primeira vez pelos hospedeiros e reportado ao servidor.

- ID de Arquivo. Um identificador único para o arquivo, incluindo um ou mais espalhamentos de conteúdo tais como MD5, SHA-1, e OMAC.
- Tipo de Arquivo. A classe do arquivo (por exemplo, executável, roteiro, documento Office, arquivo, etc.). Esta é derivada do nome do arquivo como foi primeiramente visto (veja abaixo) e também da análise do conteúdo do arquivo.
- Estado/Estado. O estado corrente do arquivo incluindo Aprovado, Pendente, Banido.
- Método. O método pelo qual o servidor aprendeu sobre o arquivo (automaticamente, manualmente, etc.).
- Nome do Arquivo. O nome do arquivo, como primeiramente visto e reportado para o servidor. Este pode não ser o nome corrente do arquivo, mas apenas o nome da primeira instância visualizada na rede.
- Caminho do Arquivo. O caminho do arquivo como primeiramente visto e reportado para o servidor.
- Primeiro Hospedeiro Visto. O nome do hospedeiro no qual o arquivo ou espalhamento foi visto pela primeira vez e reportado.
- Resultados de Análises. O resultado dos últimos escaneamentos ou outras análises.
- Primeira Análise. O tempo da primeira varredura / análise do arquivo.
- Última Análise. O tempo que o arquivo foi escaneado / analisado pela última vez.
- Última Atualização. O tempo que o estado do arquivo foi modificado pela última vez.
- Contêineres Pai. Links para outros arquivos que foram associados com o arquivo.
- Atributos do Contêiner Pai. Nome do arquivo, tempo primeiramente visto, primeiro hospedeiro visto, caminho do arquivo, classificações do produto, e estado de um arquivo contêiner associado.
- Contêineres Raiz. Links para outros arquivos que foram associados com o

arquivo. Um contêiner raiz é um contêiner que não está contido em outro contêiner.

- Atributos do Contêiner Raiz. Nome do arquivo, tempo primeiramente visto, primeiro hospedeiro visto, caminho do arquivo, classificações do produto, e estado de um arquivo contêiner pai associado.

As seguintes operações podem ser executadas em um arquivo selecionado da lista:

- Detalhe de Arquivo. Proporciona informação adicional do arquivo a partir da base de dados anticorpo, incluindo o usuário da interface que aprovou ou baniu o arquivo, onde o arquivo foi visto pela primeira vez e quaisquer comentários adicionados por usuários.
- Aprovar. Explicitamente aprova os arquivos correntemente selecionados. Esta opção deve providenciar alertas adequados para o usuário, pois ela aprova os arquivos em todos os hospedeiros.
- Desaprovar. Explicitamente desaprova arquivos que já estão aprovados, preferencialmente transitando o estado para Pendente.
- Banir. Bane explicitamente um arquivo. Isto faz com que o arquivo seja banido em todos os hospedeiros.
- Análise / Escaneamento de Vírus. Força a programação de uma análise / escaneamento para os arquivos selecionados.
- Apagar. Remove informações neste arquivo. Isto faz com que o servidor trate o arquivo como novo da próxima vez que ele for visto.
- Acha Arquivos em Hospedeiros. Esta operação conecta o localizador de arquivos, fornecendo os nomes selecionados de arquivos como entradas.
- Acha Contêineres. Verifica possíveis contêineres para o arquivo e informações para estes contêineres.
- Acha Contêineres Raiz. Verifica possíveis contêineres raiz para o arquivo e informações para estes contêineres.
- Acha Informações Serviços *Web*. Requisita que vários outros servidores de rede encontrem informações adicionais sobre o arquivo e/ou seus contêineres / produtos.

Um painel Acha Arquivos permite que o usuário inicie um

processo com os melhores esforços para encontrar as localizações de um arquivo em particular em todos os hospedeiros gerenciados. Como este processo pode consumir tempo, o usuário será notificado antes de iniciar uma nova procura. O localizador de arquivos pode não ser implementado em todas as versões do produto. O progresso do AchaArquivo pode ser exibido durante uma consulta parcialmente completada.

Este processo também pode ser iniciado a partir do painel da base de dados anticorpo (veja seção 0) com a seleção de um arquivo particular ou arquivos, o que então leva o usuário para o painel Acha Arquivos com a informação apropriada preenchida automaticamente.

Este processo requer que todos os hospedeiros que estão em comunicação com o servidor retornem estados assincronicamente, assim o painel irá abrir uma nova visualização para exibir dinamicamente os resultados à medida que eles são recebidos. Se o usuário iniciar uma outra procura, a procura corrente será encerrada. Múltiplas procuras de arquivos podem ser implementadas em futuras versões.

Um painel grupo de hospedeiros permite que os hospedeiros conhecidos pelo servidor sejam associados com um grupo lógico particular. A funcionalidade completa de grupos pode não estar disponível em versões iniciais da interface, e neste caso esta tela irá exibir informações sobre o grupo único suportado por este servidor.

O painel suporta a manipulação de grupos, incluindo:

- Adição de novos grupos.
- Remoção de grupos existentes. Quando um grupo é removido, os hospedeiros não são removidos da base de dados do servidor, mas são reatribuídos a um grupo padrão.
- Mover hospedeiros de um grupo ao outro.

A seguinte informação é exibida neste painel sobre cada hospedeiro:

- Hospedeiro. O nome DNS dos hospedeiros.
- ID Único. O identificador único dos hospedeiros.
- Endereço IP. O último endereço IP conhecido deste hospedeiro.

- Estado. O estado *online* do hospedeiro.
- Ultimamente Visto. A última vez que o hospedeiro se registrou com o servidor.
- Sistema Operacional. O sistema operacional do hospedeiro.
- Versão. A versão do sistema operacional do hospedeiro.

Um painel de classe de arquivo permite a visualização e a edição das extensões do arquivo que estão mapeadas para cada classe. Algumas classes, como abaixo, são definidas por extensões. Outras classes são determinadas por análises de conteúdo. Algumas classes são determinadas por ambos, a extensão e a análise. Estas extensões são apenas de leitura.

Algumas extensões predefinidas são:

- Executáveis. Extensões incluindo exe, com, dll, pif, scr, drv, e ocx.
- Roteiros. Extensões incluindo vbs, bat e cmd.
- Conteúdo Embarcado em Macros. Extensões incluindo doc, dot, xls, xla, xlt, xlw, ppt, pps e pot.
- Conteúdo *Web*. Extensões incluindo htm, html, asp e cgi.

Um painel de diretrizes é o núcleo da configuração do servidor. O usuário pode exibir e editar as diretrizes reforçadas em todos os hospedeiros gerenciados, agrupados por grupos de hospedeiros. Este painel também pode exibir a programação global corrente D para o grupo correntemente selecionado.

Esta seção permite que o usuário defina o nível global D para o grupo correntemente selecionado. Quando um novo nível D é escolhido, a modificação não é imediatamente aplicada, mas deve ser selecionada explicitamente. Escolher um novo nível D proposto modifica a exibição das informações das diretrizes e ações para mostrá-las para este novo nível. Navegar para longe do painel não aplica as mudanças.

A lista de diretrizes exibe as várias ações e efeitos de níveis particulares D em classes de arquivos particulares (por exemplo, executáveis, roteiros, etc.). Diretrizes podem ser habilitadas ou desabilitadas, mas não editadas. As seguintes diretrizes estão incluídas na lista:

- Novos Executáveis

- Novos Roteiros Autônomos
- Novos Roteiros Embarcados
- Novo Conteúdo *Web*
- Arquivos não Aprovados
- Ignorar Agente de Atualização (aprova automaticamente novo conteúdo de determinadas fontes / processos / localizações de atualização)
- Arquivos Infestados por Vírus / Software Espião

Toda vez que uma diretriz é desabilitada o rastreamento de arquivos desta classe continua, mas nenhuma ação é tomada pelos sistemas dos hospedeiros afetados.

Para cada diretriz, uma grade de ações é exibida. A grade indica qual programação de diretriz se aplica ao nível D correntemente selecionado.

- Ação
- Bloquear Execução. A execução desta classe de arquivo será bloqueada?
- Bloquear Escrita. A escrita em arquivos desta classe de arquivos será bloqueada? Esta programação é utilizada somente para conteúdo *web* e arquivos não aprovados. Ela é utilizada apenas para sistemas fortemente controlados e não para a operação normal.
- Quarentena. Arquivos desta classe serão colocados em quarentena? Arquivos podem ser colocados em quarentena através de bloqueio de leitura, ao invés de serem enviados a um diretório separado. No caso de arquivos infectados por vírus, eles podem ser escritos, mas posteriormente apagados, mas esta funcionalidade pode também não estar implementada inicialmente.
- Registro. O acesso a arquivos desta classe será registrado?
- Aprovação
- Aprovação Implícita. Os arquivos serão aprovados implicitamente neste nível D? Uma aprovação implícita muda o estado aprovado do arquivo após varreduras apropriadas e tempos de espera.
- Aprovação Explícita. Os arquivos serão aprovados explicitamente neste

nível D?

Uma grade de ações similar à ilustrada acima, mostra ao usuário uma representação dos efeitos de níveis particulares de D em combinação com diretrizes pré-fabricadas. As tabelas abaixo mostram um exemplo da combinação de ações e diretrizes pré-fabricadas nos vários níveis D (zero a sete).

Parâmetros Notificadores

Quando o acesso ao conteúdo é bloqueado, o usuário do hospedeiro é notificado. Para cada diretriz da lista, e para cada grupo de hospedeiros, as seguintes programações estão disponíveis:

- Mensagem exibida. O texto exibido no diálogo interativo do usuário. Mensagens múltiplas são listadas em uma caixa de listagem.
- Texto de botão. O texto exibido em um único botão no diálogo interativo do usuário.
- Intervalo. Quanto tempo o diálogo será exibido para o usuário. Um intervalo zero indica aceito pelo usuário, e o diálogo permanece exibido indefinidamente.
- Opcionalmente, para certos valores de D, existe um botão para sobrepor restrições de conteúdo por um período de tempo.
- Link URL com mais informações sobre a diretriz.

Os parâmetros de notificação também incluem uma programação global que define a imagem exibida no hospedeiro em conjunto com a mensagem de notificação. Estas programações são configuráveis para uma das diretrizes pré-fabricadas individualmente. Parâmetros de notificação são editados na interface administrativa do servidor. Estes parâmetros são associados com diretrizes, que por sua vez são atribuídas a grupos de hospedeiros, e propagadas a hospedeiros quando as diretrizes mudam.

Parâmetros de Idade de Escaneamentos

Esta seção permite que o usuário configure o período de tempo quando um arquivo foi visto pela primeira vez e quando foi aprovado (escaneamento de auto-aprovação), o tempo que o segundo escaneamento (aprovação) foi conduzido e o tempo que um terceiro escaneamento (repetição)

ocorreu. Mais escaneamentos e tempos podem ser especificados como na Figura 7.

Manutenção

A seção de Manutenção permite que o usuário configure programações globais para o próprio servidor.

- Configuração de Sistema. Configurações relacionadas com a interação do servidor com a rede local e com os sistemas de hospedeiros.
- Endereço IP e máscaras de sub-rede. Máscaras de sub-rede permitem a classificação de hospedeiros nos tipos Remoto e Local. Hospedeiros remotos têm comunicações mais restritas para conservar largura de banda. Grupos de hospedeiros podem ter diferentes diretrizes e parâmetros de programação D, que são especificados para cada tipo de conexão, Remoto, Local, ou Desconectado. Hospedeiros remotos geram menor tráfego de rede, por exemplo, menos relatórios de servidor. Hospedeiros remotos também reportam preferencialmente espalhamentos de novos conteúdos para o servidor, mas não transferem o conteúdo.
- Informação de roteamento IP.
- Senhas. Define ou reseta senhas de acesso à interface do servidor.
- Certificados. Instala certificados de mídias removíveis (e opcionalmente da rede). Estes são utilizados pelos hospedeiros para verificar a identidade do servidor e também para a interface SSL do servidor.
- SNMP. Define uma lista de servidores SNMP que recebem armadilhas e que têm permissão para consultar a configuração dos servidores.
- Seleção de armadilha SNMP. Seleciona que tipo de evento, quais armadilhas e para qual serviço SNMP a armadilha será enviada (e também define a prioridade critica, alta, média, baixa, informacional, etc...).
- Registrador de atividades de sistema. Define uma lista de servidores

que recebem informações de registro via o registrador de atividades de sistema, para vários tipos de eventos e prioridades.

- Servidor de sincronização de tempo NTP. Define uma lista de servidores para sincronização de tempo. O tempo no servidor é retirado do seu relógio interno durante a inicialização e então sincronizado com sua fonte de tempo externa NTP. Desvios de tempo do hospedeiro do tempo do servidor serão rastreados pelo servidor.

- Estado de Sistema (servidor)
  - Tempo Ativo. Exibe o tempo desde a última reinicialização do sistema.
  - Versão do Software. Exibe a informação de versão do software do servidor.
  - Espaço de disco. Exibe estatísticas de discos locais e de armazenamento do servidor.
- Atualizações de Assinaturas de Vírus / Software Espião
  - Atualização da Última Assinatura. O tempo da atualização da última assinatura.
  - Configuração de serviços de atualização. Configura o serviço de atualização para o software antivírus instalado, incluindo localizações de *downloads* e cronogramas.
  - Atualização Scanner. Atualiza o software de escaneamento de vírus.
  - Atualização Assinaturas. Força uma atualização das assinaturas de vírus.
- Atualização do Software do Servidor
  - Versão Corrente. Exibe a versão corrente do software do servidor.
  - Reinicialização. Reinicializa o servidor utilizando a imagem correntemente instalada.
  - Carregar nova imagem. Carrega uma nova imagem do software para o servidor a partir de mídia removível ou rede (por exemplo, via FTP).

- Reverter para a versão anterior. Reverte para a imagem do software previamente utilizada.

- Configuração de Serviços Externos.
  - Endereço de rede, tipo de serviço, e autoridade de aprovação para sistemas de escaneamento de conteúdo.
  - Endereço de rede, tipo de serviço, e autoridade de aprovação para serviços de compartilhamento de meta-informações.
  - Endereços de servidores de arquivos externos, protocolos, registros, e diretórios para a transferência de conteúdo externo e análises definidas pelo usuário.
  - Configurações SNMP de serviços externos de notificação de conteúdo, atividades de sistema, e-mail e notificação SOAP de novo conteúdo.
- Backup. Faz backup e restaura a configuração para mídias removíveis (e também para a rede).
  - Salvar configuração e base de dados. Salva a configuração e a base de dados anticorpo (por exemplo, via XML).
  - Carregar configuração e base de dados. Carrega a configuração e a base de dados anticorpo (por exemplo, via XML).

O servidor incorpora capacidades de processamento tais como um microprocessador programado, processador digital de sinais (DSP), ou processamento e memória específicos de aplicações. Hospedeiros podem incluir computadores pessoais ou computadores similares, ou outros dispositivos de processamento, incluindo computadores de mão, PDAs, ou outros dispositivos em uma rede.

Tendo descrito aqui as modalidades preferidas de execução da invenção, deve ficar aparente que modificações podem ser feitas sem divergir do escopo da invenção conforme reivindicado.

REIVINDICAÇÕES

1. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, *caracterizado pelo fato* de ser utilizado em um sistema de computador composto por uma pluralidade de computadores hospedeiros (hospedeiros) e um servidor associado aos hospedeiros, que compreende:

- o servidor, que propaga para os hospedeiros um conjunto mestre de diretrizes relacionadas às operações de arquivos, e opções de diretrizes indicando pelo menos se e com quais condições tais operações são permitidas ou banidas;

- o servidor, que propaga um valor para os hospedeiros;

- o valor armazenado no hospedeiro, que indica qual subconjunto de diretrizes e opções de diretrizes deve ser implementado no hospedeiro a partir do conjunto mestre de diretrizes e opções de diretrizes;

- o hospedeiro, que implementa as diretrizes da operação de arquivo indicadas pelo valor.

2. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com a reivindicação 1, *caracterizado pelo fato* de cada diretriz ter um único parâmetro de configuração que indica uma das opções da diretriz, sendo que o valor é propagado por meio da seleção da opção da diretriz para cada uma das diversas diretrizes.

3. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com a reivindicação 2, *caracterizado pelo fato* de o conjunto mestre incluir listas de diretrizes e opções, sendo que o valor é propagado por meio da seleção de uma destas listas

4. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com a reivindicação 1, 2 ou 3, *caracterizado pelo fato* de o hospedeiro modificar o valor automaticamente em resposta a um relatório de diretriz no mesmo hospedeiro, ou em resposta a um evento detectado no hospedeiro, ou em resposta a um comando executado no hospedeiro.

5. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com a reivindicação 1, 2, 3 ou 4, *caracterizado pelo fato* de as opções de diretrizes incluírem automaticamente operações de execução de permissão ou bloqueio e/ou leitura de arquivos, com um estado de meta-informações indicando quais operações estão aprovadas.

6. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com a reivindicação 1, 2, 3, 4 ou 5, *caracterizado pelo fato* de as opções de diretrizes incluírem operações de execução de permissão ou bloqueio e/ou leitura de arquivos com um estado de meta-informações pendente e associado, indicando quais ações ainda não foram determinadas para serem permitidas ou banidas.

7. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de o hospedeiro manter meta-informações para cada um dos arquivos no hospedeiro, sendo que as meta-informações incluem um estado no qual há pelo menos três valores possíveis: Aprovado, Banido e Pendente.

8. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de pelo menos algumas das diretrizes e opções de diretrizes indicarem uma ação baseada no nome e/ou no conteúdo de um arquivo.

9. MÉTODO DE SEGURANÇA PARA REDES DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de o servidor modificar o valor, por meio do registro de um novo valor de uma maneira acessível para os hospedeiros, sendo que os hospedeiros acessam o novo valor, comparam o novo valor com o valor que os hospedeiros têm e modificam seu valor para o novo valor.

10. SISTEMA DE COMPUTADOR *caracterizado pelo fato* de compreender:

- uma pluralidade de computadores hospedeiros (hospedeiros);

- um servidor para propagar, para os hospedeiros, um conjunto mestre de diretrizes relacionadas às operações de arquivos, e opções de diretrizes indicando pelo menos se e com que condições estas operações são permitidas ou banidas;

- servidor para também propagar um valor para os hospedeiros, para o armazenamento nos hospedeiros;

- um valor armazenado nos hospedeiros indicando qual o subconjunto de diretrizes e opções de diretrizes deve ser implementado nos hospedeiros a partir do conjunto mestre de diretrizes e opções de diretrizes;

- um hospedeiro para a implementação das diretrizes de operações de arquivos indicadas pelo valor.

11. SISTEMA DE COMPUTADOR, de acordo com a reivindicação 10, *caracterizado pelo fato* de a informação propagada pelo servidor incluir um valor que indica um conjunto de opções de diretrizes para cada uma das diversas e diferentes diretrizes.

12. SISTEMA DE COMPUTADOR, de acordo com a reivindicação 10 ou 11, *caracterizado pelo fato* de os hospedeiros serem organizados em múltiplos grupos de hospedeiros, sendo que o servidor propaga as modificações do valor para um ou mais, mas não para todos os grupos de hospedeiros.

13. SISTEMA DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de as operações de arquivos incluírem acessos de escrita aos arquivos e execução de arquivos, sendo que as opções incluem uma pluralidade de opções em um conjunto ordenado de restrições que gradativamente aumentam ou reduzem a capacidade dos hospedeiros de executar as operações de arquivos.

14. SISTEMA DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de o conjunto mestre incluir listas de diretrizes e opções de diretrizes, sendo que o servidor fornece informações, incluindo um valor que indica uma das listas.

15. SISTEMA DE COMPUTADOR, de acordo com qualquer uma das reivindicações precedentes, *caracterizado pelo fato* de o servidor postar as informações em um local acessível aos hospedeiros e os hospedeiros acessarem as ditas informações e atualizarem seus valores.

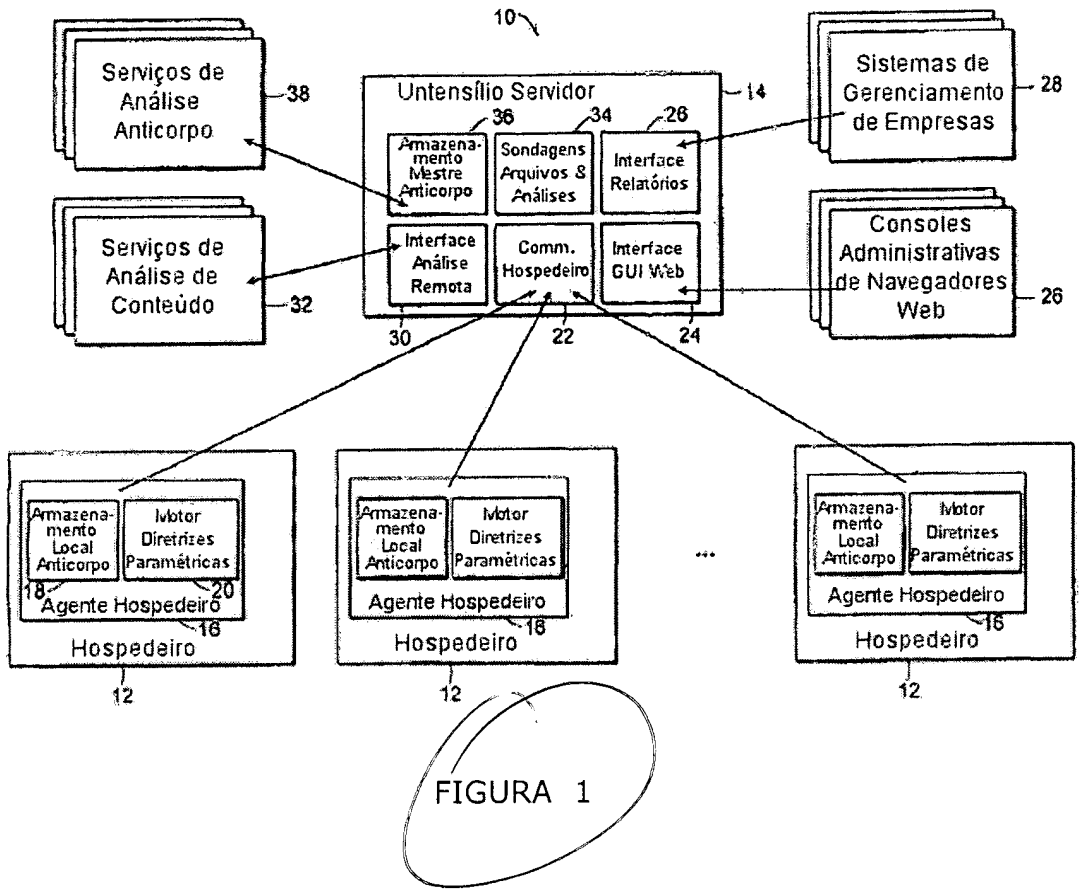


FIGURA 1

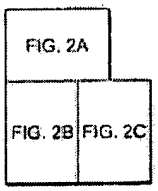


FIGURA 2

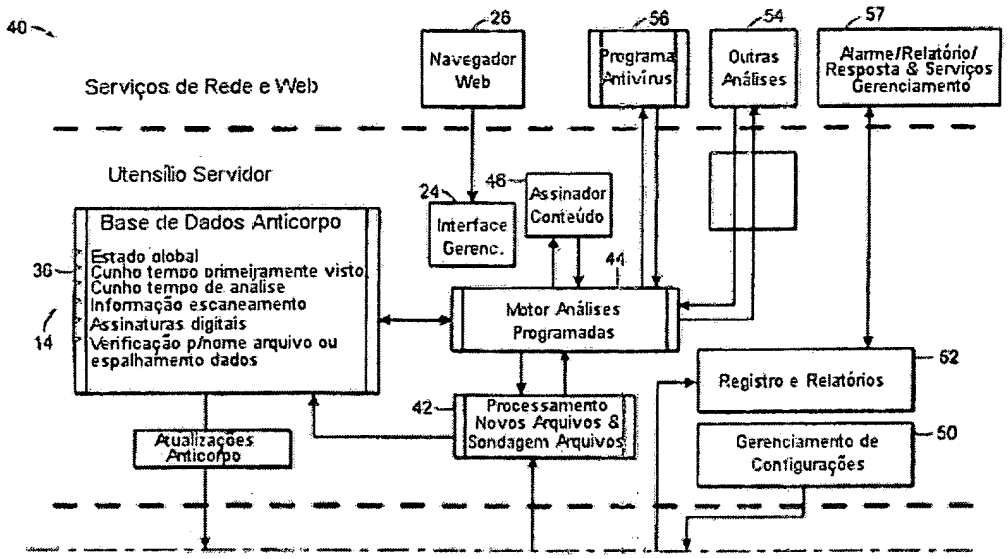


FIGURA 2A

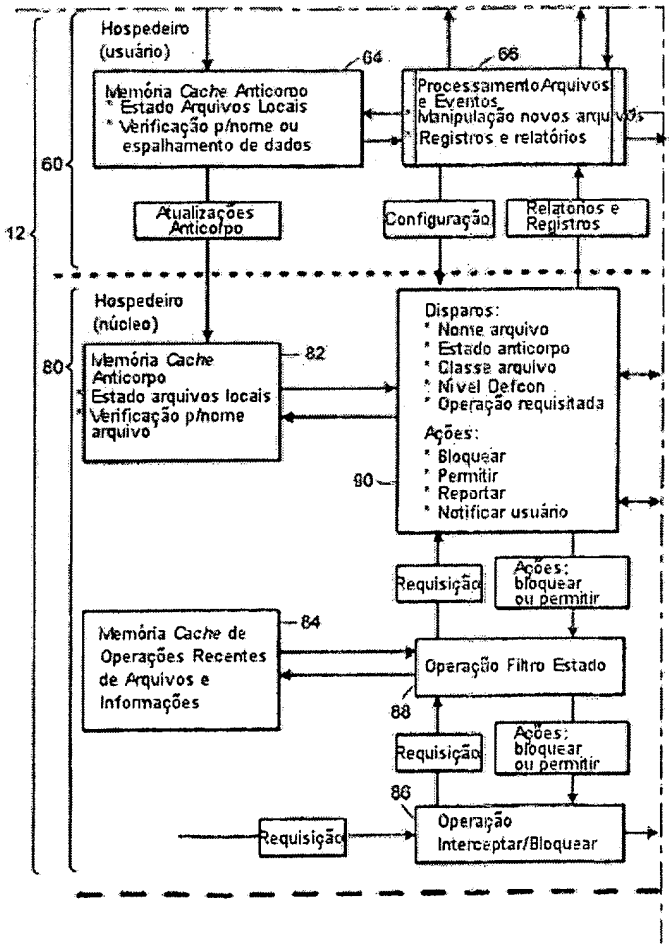


FIGURA 2B

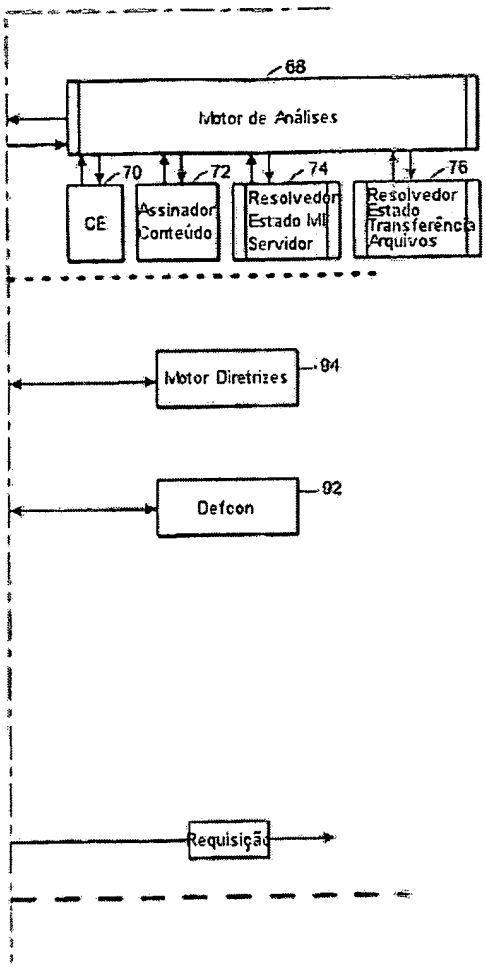


FIGURA 2C

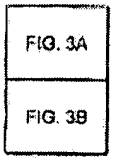


FIGURA 3

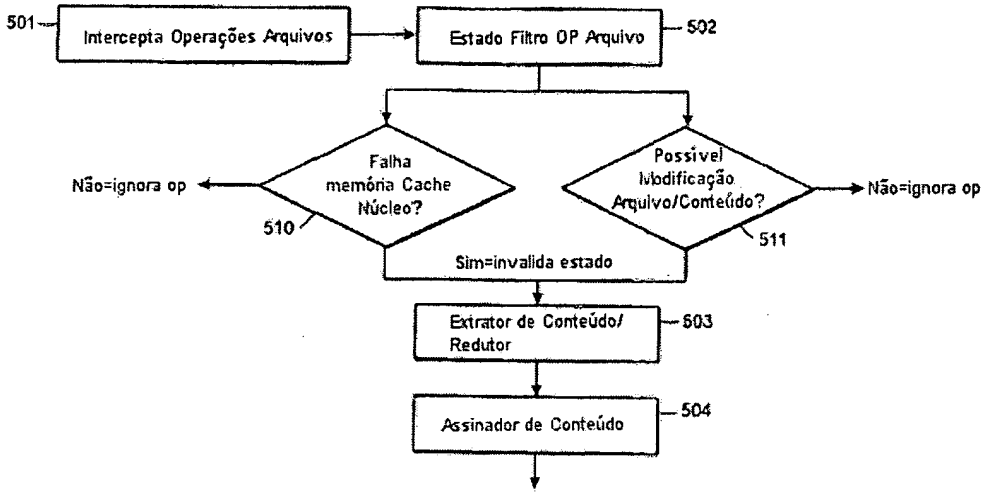


FIGURA 3A

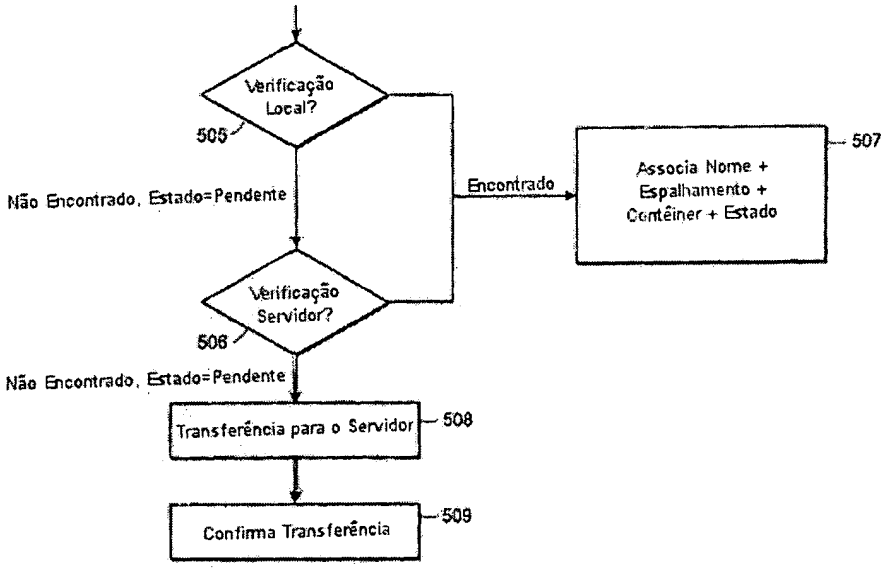


FIGURA 3B

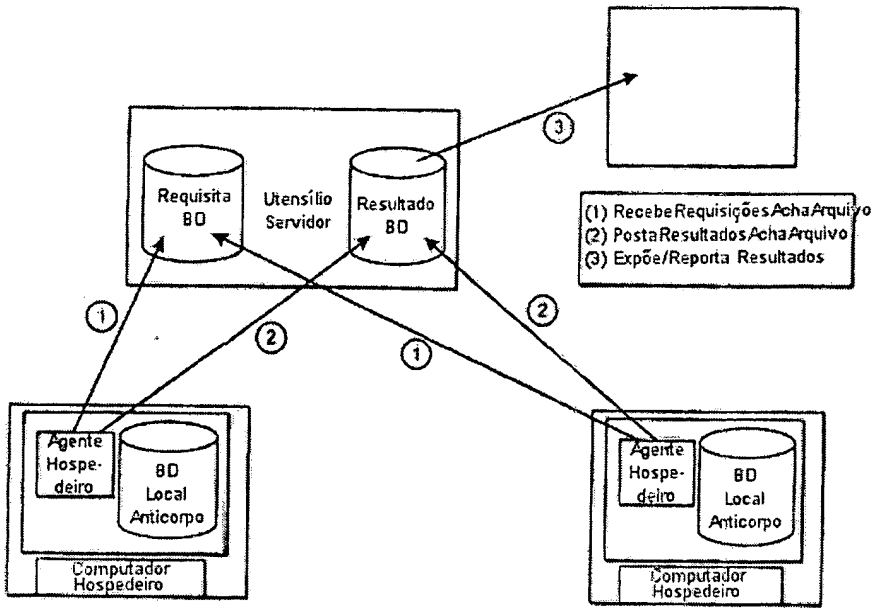


FIGURA 4

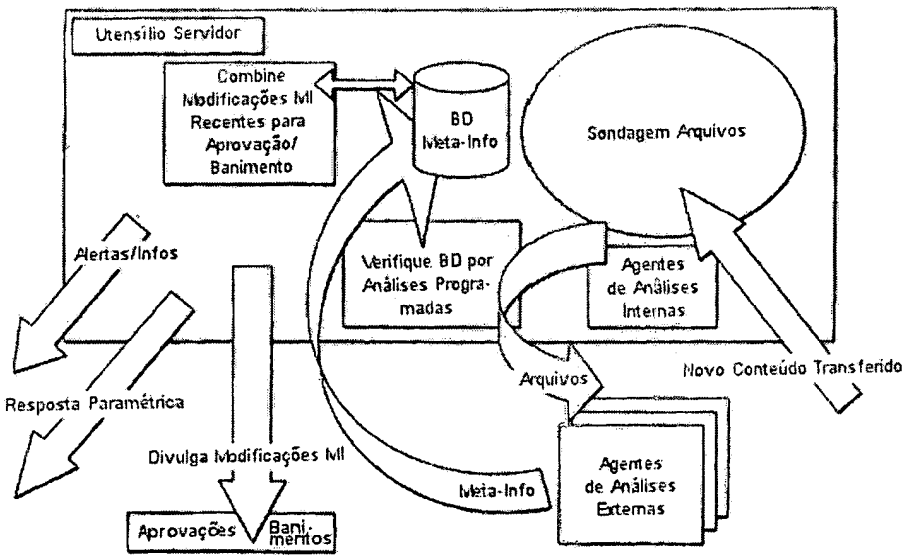


FIGURA 5

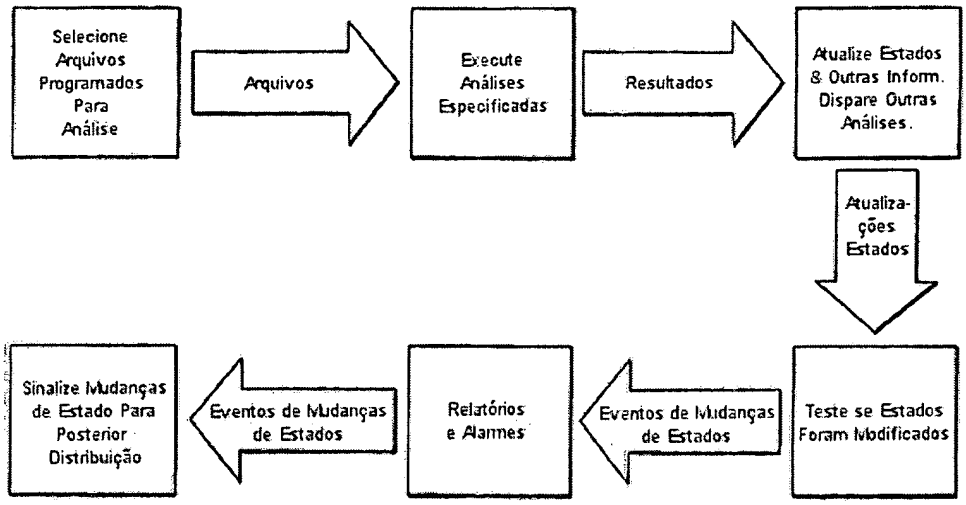


FIGURA 6

$\Delta t = t_{corrente} - t_{arquivo\_primeiramente\_visto\_na\_rede}$

$\Delta t=0$	$\Delta t=12$ h	$\Delta t=2$ dias	$\Delta t=30$ dias
Certificação Espalhamento			
Escaneam. AV #1 Escaneam. AV #2	Escaneam. AV #1 Escaneam. AV #2	Escaneam. AV #1 Escaneam. AV #2	Escaneam. AV #1 Escaneam. AV #2
Escaneam. AS #1 Escaneam. AS #2	Escaneam. AS #1 Escaneam. AS #2	Escaneam. AS #1 Escaneam. AS #2	Escaneam. AS #1 Escaneam. AS #2
Outras Análises #1 (Analise Novos)			
		Outras Análises #2 (Analise Persistentes)	Outras Análises #2 (Analise Persistentes)

FIGURA 7

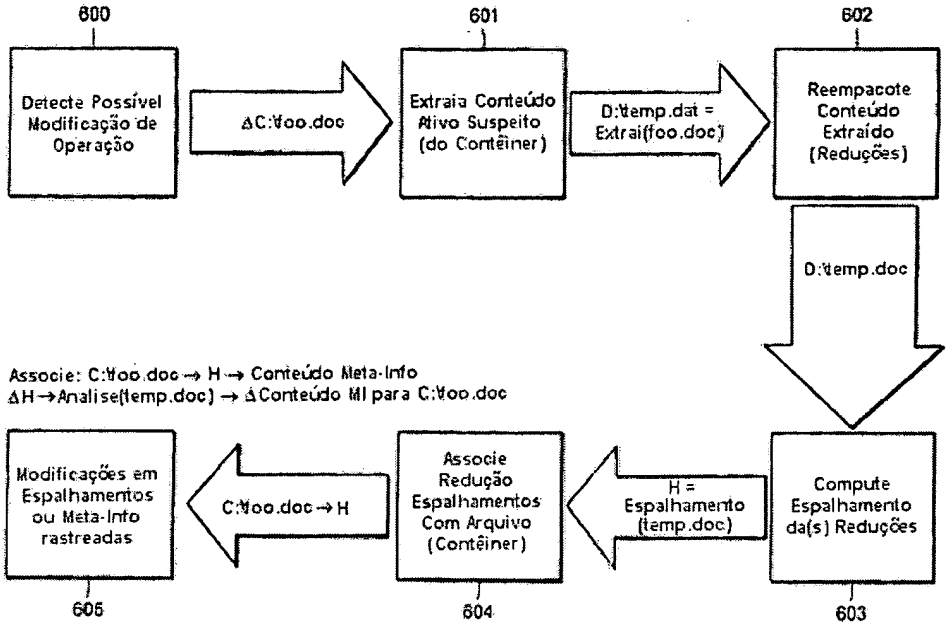

600
Detecte Possível Modificação de Operação
ΔC:\foo.doc
601
Extraia Conteúdo Ativo Suspeito (do Contêiner)
D:\temp.dat = Extrai(foo.doc)
602
Reempacote Conteúdo Extraído (Reduções)
D:\temp.doc
Associe: C:\foo.doc → H → Conteúdo Meta-Info
ΔH → Analise(temp.doc) → ΔConteúdo MI para C:\foo.doc
Modificações em Espalhamentos ou Meta-Info rastreadas
C:\foo.doc → H
Associe Redução Espalhamentos Com Arquivo (Contêiner)
H = Espalhamento (temp.doc)
Compute Espalhamento da(s) Reduções
605
604
603

FIGURA 8

PI 0616018-2

RESUMO

"SISTEMAS E MÉTODOS DE SEGURANÇA PARA REDES DE COMPUTADOR", que consistem em sistemas e métodos de segurança que proporcionam uma defesa contra vírus conhecidos e desconhecidos, programas maliciosos, softwares espião, invasores, e softwares indesejados ou desconhecidos. O sistema pode implementar diretrizes centralizadas que permitem que um administrador aprove, bloqueie, coloque em quarentena, ou registre atividades de arquivos. O sistema mantém meta-informações de arquivos nos hospedeiros e no servidor. Um hospedeiro detecta operações de arquivos que podem provocar mudanças no conteúdo do arquivo ou no nome do arquivo, e atualiza as meta-informações do hospedeiro e/ou do servidor como resultado. As modificações das meta-informações do servidor são disponibilizadas para os hospedeiros.