(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4880674号
(P4880674)
(45)発行日　平成24年2月22日(2012.2.22)　(24)登録日　平成23年12月9日(2011.12.9)

(51)Int.Cl.　　F I
G06F 21/22 (2006.01)　　G06F 21/22 112J
G06F 21/00 (2006.01)　　G06F 21/00 156E

請求項の数 13　(全 15 頁)

(21)出願番号　特願2008-507840 (P2008-507840)
(86)(22)出願日　平成18年4月20日 (2006.4.20)
(65)公表番号　特表2008-538638 (P2008-538638A)
(43)公表日　平成20年10月30日 (2008.10.30)
(86)国際出願番号　PCT/US2006/014743
(87)国際公開番号　WO2006/115935
(87)国際公開日　平成18年11月2日 (2006.11.2)
審査請求日　平成21年3月19日 (2009.3.19)
(31)優先権主張番号　11/112,507
(32)優先日　平成17年4月21日 (2005.4.21)
(33)優先権主張国　米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　マルク　イー．サインフェルド
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ワン　マイクロソフト　ウェイ　マイクロソフト　コーポレーション内

最終頁に続く

(54)【発明の名称】ウェブサービスを提供するコンピュータをマルウェアから保護する方法

(57)【特許請求の範囲】
【請求項１】
要求元コンピュータと、ウェブサービスを提供するコンピュータとを含むネットワーキング環境において、前記要求元コンピュータによって生成されたマルウェアから前記ウェブサービスを提供するコンピュータを保護する方法であって、
前記ウェブサービスを提供するコンピュータが、前記ウェブサービスに関するプログラム実行のためのリクエストを前記要求元コンピュータから受信するステップと、
前記リクエストと関連付けられた高水準コードをコンパイルするオンデマンドコンパイルシステムによって前記高水準コードをバイナリコードにコンパイルするステップと、
前記バイナリコードが実行されることがスケジュールされた場合、前記オンデマンドコンパイルシステムによって、前記バイナリコードが実行されることがスケジュールされた時期をアンチウィルスソフトウェアに通知するステップと、
当該スケジュールされたバイナリコードが実行される前に、前記アンチウィルスソフトウェアによって前記スケジュールされたバイナリコードをマルウェアスキャンするステップと、
マルウェアが前記スケジュールされたバイナリコードにおいて識別された場合には、前記ウェブサービスを提供するコンピュータによって前記スケジュールされたバイナリコードが実行されることを防止するステップと
を備えることを特徴とする方法。
【請求項２】

マルウェアが前記バイナリコードにて識別されなかった場合には、前記バイナリコードが実行されることを許可するステップをさらに備えたことを特徴とする請求項１に記載の方法。

【請求項３】

前記リクエストは、前記ウェブサービスを実施するソフトウェアルーチンによって処理されることに先立って前記リクエストがマルウェアかを判定する前記ウェブサービスコンピュータ上のフィルタにおいて、バイナリコードにコンパイルされることを特徴とする請求項１に記載の方法。

【請求項４】

前記フィルタは、

前記リクエストと関連付けられた前記高水準コードが前記バイナリコードにコンパイルされることを引き起こすコンパイラと、

前記バイナリコードが実行される前に前記バイナリコードをマルウェアスキャンするアンチウィルスソフトウェアと

を含むＩＳＡＰＩ（Internet Server Application Program Interface）フィルタであることを特徴とする請求項３に記載の方法。

【請求項５】

バイナリコードにコンパイルされる前記リクエストと関連付けられた前記高水準コードは、ＸＳＬ（Extensible Style Sheet programming language）であることを特徴とする請求項１に記載の方法。

【請求項６】

前記バイナリコードをマルウェアスキャンするステップは、

ハッシュ関数を使用して前記バイナリコードの署名を生成するステップと、

前記署名と既知のマルウェアから生成された署名とを比較するステップと

を含むことを特徴とする請求項１に記載の方法。

【請求項７】

前記バイナリコードをマルウェアスキャンするステップは、マルウェアの特徴であるバイナリコードにおける発見的な要素を識別するステップを含むことを特徴とする請求項１に記載の方法。

【請求項８】

前記識別された発見的な要素は、前記ウェブサービスを提供するコンピュータにインストールされたオペレーティングシステムに対してバイナリコードによってされる呼び出しのタイプであることを特徴とする請求項７に記載の方法。

【請求項９】

前記リクエストは、前記要求元コンピュータと前記ウェブサービスを提供するコンピュータとの間でＨＴＴＰ（Hypertext Transfer Protocol）を使用して送信されることを特徴とする請求項１に記載の方法。

【請求項１０】

前記リクエストは、ＸＭＬ(Extensible Markup Language)プロトコルに適合するようにフォーマットされることを特徴とする請求項１に記載の方法。

【請求項１１】

要求元コンピュータとウェブサービスを提供するコンピュータとを含むネットワーキング環境において、前記要求元コンピュータによって生成されたマルウェアから前記ウェブサービスを提供するコンピュータを保護する方法を実行するためのコンピュータ実行可能命令を格納したコンピュータ読み取り可能記録媒体であって、前記方法は、

前記ウェブサービスを提供するコンピュータが、前記ウェブサービスに関するプログラム実行のためのリクエストを前記要求元コンピュータから受信するステップと、

前記リクエストと関連付けられた高水準コードをコンパイルするオンデマンドコンパイルシステムによって前記高水準コードをバイナリコードにコンパイルするステップと、

前記バイナリコードが実行されることがスケジュールされた場合、前記オンデマンドコ

ンパイルシステムによって、前記バイナリコードが実行されることがスケジュールされた時期をアンチウィルスソフトウェアに通知するステップと、

当該スケジュールされたバイナリコードが実行される前に、前記アンチウィルスソフトウェアによって前記スケジュールされたバイナリコードをマルウェアスキャンするステップと、

マルウェアが前記スケジュールされたバイナリコードにおいて識別された場合には、前記ウェブサービスを提供するコンピュータによって前記スケジュールされたバイナリコードが実行されることを防止するステップと

を備えたことを特徴とするコンピュータ読み取り可能記録媒体。

【請求項１２】

前記方法は、マルウェアが識別されなかった場合には、前記バイナリコードが実行されることを許可するステップをさらに含むことを特徴とする請求項１１に記載のコンピュータ読み取り可能記録媒体。

【請求項１３】

前記バイナリコードの前記マルウェアスキャンを実行するステップは、マルウェアの特徴であるバイナリコードにおける発見的な要素を識別するステップを含むことを特徴とする請求項１１に記載のコンピュータ読み取り可能記録媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明はコンピュータに関し、より詳細には、ウェブサービスを提供するコンピュータをマルウェアから保護することに関する。

【背景技術】

【０００２】

コンピュータネットワークの接続性、そして、より詳細には、インターネットの接続性が、商品やサービスが提供される方法に革命をもたらした。ほとんど全世界のネットワークおよびインターネットの接続性より前に、人間の情報のやり取りは、典型的にはソフトウェアを購入すること、または情報のデータベースのデータを提供することのようなトランザクションを完了することが必要とされていた。さらには、ある製品を配布するには、購入者に製品を配布し、包装するためのシステムが必要であった。しかしながら、ソフトウェアおよび同種の製品を配布するシステムを作り出すことは、通常、提供者にとって費用のかかることであった。さらには、コンピュータ化されていない手段を通じて大容量の情報を交換することは、少なくとも何人かの人間の関与が必要であり、結果として、時間を浪費し、また費用がかかってしまっていた。

【０００３】

現代のコンピュータネットワークによって提供される相互接続性は、通常、コンピュータが情報を交換することを許可することの助けとなる一方で、相互接続性は、また、コンピュータが攻撃に対してより脆弱にさせている。当業者が気付くであろうように、それらの攻撃は、決して限定されるわけではないが、コンピュータウィルス、コンピュータワーム、システムコンポーネントリプレースメント、ＤｏＳ（denial of service）攻撃、合法コンピュータシステム特性の誤用／不正使用でさえも含み、それらの全ては非合法の目的のために一または複数のコンピュータシステムの脆弱性を不正に利用する。当業者は、様々なコンピュータ攻撃は技術的に互いにはっきりと区別されることを理解するので、本発明の目的および説明の簡素のために、全ての悪意のコンピュータプログラムは、一般的にコンピュータマルウェアとして、あるいは、単にマルウェアとして以下で参照されるものとする。

【０００４】

コンピュータがコンピュータマルウェアによって攻撃されまたは“感染された”ときは、不運な結果は、システムデバイスを無効にすること、ファームウェア、アプリケーション、またはデータファイルを消去または破損すること、潜在的に機密事項データをネット

ワークの他の場所に送信すること、コンピュータをシャットダウンすること、またはコンピュータを破壊させることを含み、多様である。全てではないが多くのコンピュータマルウェアのさらにもうひとつの悪意の側面は、感染されたコンピュータが別のコンピュータを感染するために使用されることである。

【０００５】

インターネットのために開発されたプロトコルを使用してネットワークコンピュータの間のデータ通信を促進するひとつのシステムは、ウェブサービスである。当業者およびその他の者は、ウェブサービスが、他のソフトウェアシステムに変わってアクションを実行するネットワークアクセス可能インタフェースでソフトウェアシステムを参照することに気付くであろう。ウェブサービスは通常はＳＯＡＰ（Simple Object Access Protocol）のような標準的なプロトコルを使用してアクセスされる。リモートコンピュータに配置されたソフトウェアシステムは、ウェブサービスとのコミュニケーションのための方法を定義するサービス記述にて提供される定義によって指示された方法でウェブサービスと情報のやり取りをすることができる。また、ソフトウェアシステム間の情報のやり取りは、通常は、ＨＴＴＰ(HyperText Transfer Protocol)のようなインターネットベースのプロトコルを介して交換されるＸＭＬ（Extensible Markup Language）ベースのメッセージを使用することを生じる。たとえば、ウェブサービスとコミュニケーションするためのひとつの方法としては、エンドポイントのセットとしてのウェブサービスを記述するために使用されるＸＭＬベース言語であるＷＳＤＬ（Web Services Description Language）を使用することが挙げられる。この方法では、コンピュータ上にてデータにアクセスしまたは動作を実行するため、あるいはウェブサービスを提供するコンピュータのクラスタのために、ウェブサービスはプロセスをリモートソフトウェアシステムに公開することができる。通常は、ウェブサービスは、ＵＲＩ（Uniform Resource Indictor）を使用して識別されることができるネットワークの特定位置における他のソフトウェアシステムとの情報のやり取りをサポートする。ウェブサービスは、ソフトウェアシステム間のコミュニケーションのために、開発者が異なるオペレーションシステムおよびプログラミング言語を使用することを許可する。さらに、ウェブサービスによって提供されるプロセスは、たとえば、ウェブインタフェースを介してＸＭＬデータを変換することによって利用可能である。結果として、異なるプログラムは、複雑な動作を成し遂げるためのゆるく結合された方法に組み合わせられることができる。

【０００６】

残念なことに、ウェブサービスへリクエストをするエンティティ（以下、“サービス要求元”、または“要求元コンピュータ”という）は、悪意の効果を奏する一または複数のメッセージを生成することができる。換言すれば、ウェブサービスを提供するコンピュータは、サービス要求元によって生成されたマルウェアに感染しやすい。たとえば、マルウェア作者は、サーバコンピュータのリクエストにおけるＸＭＬ文法を解析することの計算に関する複雑性に起因して、ＤｏＳ攻撃を引き起こす有効に形成されたリクエストをウェブサービスに渡すことができる。前述したように、この合法コンピュータシステム特性の誤用／不正使用のタイプは、送信を受信するコンピュータにネガティブな効果を引き起こすものであるが、本願においてはマルウェアとして分類される。当業者や他の者は、コンピュータとネットワークがネットワーク帯域幅、メモリ、ディスクスペース、およびＣＰＵ（central processing unit）へのアクセスなどのような動作をするために一定のリソースを必要とすることに気付くだろう。ＤｏＳ攻撃において、ウェブサービスを提供するコンピュータ上の十分ではないリソースを消耗し、制圧するために設計されたリクエストがウェブサービスに対して作成される。結果として、他のサービス要求元は攻撃されているウェブサービスに拒否され、またはアクセスすることを制限される。当業者や他の者は、ＤｏＳ攻撃は、ウェブサービスへのリクエストにて生成されたマルウェアによって攻撃される可能性があるウェブサービスを提供するコンピュータにおける、単なる一例であることに気付くであろう。

【発明の開示】

【発明が解決しようとする課題】
【０００７】
先行技術の状況に伴う前述の問題は、本発明の原理によって克服することができ、本発明は、ウェブサービスを提供するコンピュータをマルウェアから保護するシステム、方法、およびコンピュータ読み取り可能媒体に関する。
【課題を解決するための手段】
【０００８】
本発明の一態様は、要求元コンピュータによって生成されたマルウェアからウェブサービスを提供するネットワーキング環境におけるコンピュータを保護する方法である。より詳細には、その方法は、ウェブサービスを提供するコンピュータにおいてリクエストを受信すること、リクエストと関連付けられた高水準コードが、実行され得るバイナリコードにコンパイルされることを引き起こすこと、バイナリコードをマルウェアスキャンすることを備える。その方法がリクエストにおいてマルウェアを識別した場合には、リクエストに関連付けられたコードは実行されない。逆に、マルウェアが識別されなかった場合には、リクエストは満足したものとなる。
【０００９】
本発明の他の態様は、マルウェアがウェブサービスを提供するコンピュータ上において実行されることを防止するソフトウェアシステムである。本発明の一実施態様においては、ソフトウェアシステムは、リクエストを受け付けることが可能なネットワークアクセス可能なインタフェースを含む。リクエストが受信されると、オンデマンドコンパイルシステムは、リクエストに関連付けられた高水準コードを、実行されることが可能なバイナリコードへコンパイルすることができる。しかしながら、リクエストに関連付けられたバイナリコードが実行される前に、アンチウィルスソフトウェアがバイナリコードをマルウェアスキャンする。一実施態様においては、アンチウィルスソフトウェアはバイナリコードと周知のマルウェアから生成されたマルウェア署名とを比較するスキャンエンジンを含む。
【００１０】
さらにもうひとつの実施態様においては、コンピュータ読み取り可能媒体がコンテンツ、すなわち、前述した方法に従ってコンピュータを動作させるプログラムを伴って提供される。
【００１１】
前記態様および本発明の付随する多くの利点は、添付の図面とあわせて以下の詳細な説明を参照することでよりよく理解され、より容易に認識される。
【発明を実施するための最良の形態】
【００１２】
本発明によれば、ウェブサービスへのリクエストにおけるマルウェアを識別するためのシステム、方法、およびコンピュータ読み取り可能媒体が提供される。本発明の一態様は、ウェブサービスを提供するコンピュータをウェブリクエストに実装されたマルウェアから保護するコンピュータ実装された方法である。リクエストが受信されると、オンデマンドコンパイルシステムは、リクエストに関連付けられた高水準コードを、実行され得るバイナリコードへとコンパイルする。しかしながら、コードが実行される前に、リクエストと関連付けられたマルウェアを識別するために設計されたアンチウィルスソフトウェアがバイナリコードをマルウェアスキャンする。マルウェアが識別された場合には、アンチウィルスソフトウェアは、バイナリコードが実行されることを防止する。
【００１３】
本発明は、主にウェブサービスに対して送信されたマルウェアを識別することに関連して説明されるものであるが、関連する当業者およびその他の者は、本発明が説明される以外のソフトウェアシステムにもまた適用することができることに気付くだろう。以下の説明は、最初に従来技術の態様および本発明が実装され得るソフトウェアシステムの概要を提供する。それから、本発明が実装される方法を説明する。以下で提供される説明に役立

つ実例は、網羅されたことを意図するものではなく、また、開示された正確な形式に限定されるものではない。同様に、以下で説明されるいずれのステップは同じ結果を実現するために他のステップまたステップの組み合わせに変更することができる。

【００１４】

図１および以下の考察は、先行技術において形成されたネットワーク環境１００において実装されたウェブサービスの簡潔な概要を提供するものである。図１に示すように、ネットワーキング環境１００は、要求元コンピュータ１０２およびウェブサービスプロバイダコンピュータ１０４と、からなる。また、要求元コンピュータ１０２およびウェブサービスプロバイダコンピュータ１０４は、ネットワーク１０６を介して通信で接続される。当業者および他の者は、ネットワーク１０６がＬＡＮ(Local area network)、ＷＡＮ(Wide area network)、セルラーネットワーク、ＩＥＥＥ８０２．１１、ブルートゥース（登録商標）無線ネットワークなどとして実装されることができると気付くだろう。しかしながら、通常、ネットワーク１０６は、インターネットまたはWWW（World Wide Web）として一般に知られているグローバルネットワークである。

【００１５】

本発明は、コンピュータ１０２および１０４のようなパーソナルコンピュータと連動する動作に関して一般的に説明されるが、説明するためのものであり、本発明を限定するものとして解釈されてはならないことに留意されたい。当業者はコンピュータのほとんどすべてのタイプがウェブサービスと情報のやりとりをし、または実装され得ることにすぐに気付くだろう。従って、本発明は、限定されるわけではないが、コンピュータの多数のタイプ、コンピュータ機器、または、パーソナルコンピュータ、タブレットコンピュータ、ノートブックコンピュータ、ミニおよびメインフレームコンピュータ、サーバコンピュータなどを含むコンピューティングシステムを保護するために有利に実装される。

【００１６】

さらに図１に示すように、要求元コンピュータ１０２は、プログラム実行の例示的なフロー１０８を保持する。現在のネットワークのインフラより前には、プログラムはひとつのコンピュータ上において完全に実行された。しかしながら、図１に示すように、ウェブサービスプロバイダコンピュータ１０４は、ウェブサービスにアクセス可能なネットワーク１０６を提供する。当業者または他の者は、標準ネットワークプロトコルを使用してリモートコンピュータからアクセスされることができる“ブラックボックス機能”をウェブサービスが提供することに気付くだろう。たとえば、要求元コンピュータ１０２のようなひとつのコンピュータ上で実行しているアプリケーションは、リクエストを出すことによって、イベント１１０におけるウェブサービスを提供するコンピュータの機能を呼び出すことができる。その結果、プログラム実行のフロー１０８は、要求元コンピュータ１０２からウェブサービスプロバイダー１０４に転送される。この場合、機能を呼び出すことは、通常、ウェブサービスプロバイダー１０４においてプログラムコードが実行されることを引き起こすだろう。当業者およびその他の者は、ウェブサービスプロバイダー１０４においてマルウェアが実行されることを引き起こす方法でリクエストが構成され得ることに気付くだろう。ウェブサービスにおいて呼び出された機能が完了したときに、イベント１１２において、プログラム実行のフロー１０８は、要求元コンピュータ１０２に返送される。通常、ウェブサービスは標準のネットワークプロトコルを使用してレスポンスの形式のデータが要求元コンピュータ１０２に転送されることを引き起こす。図１に示すように、ウェブサービス１１０は、２リンクソフトウェアコンポーネントであるネットワーク１０６を使用する仮想アプリケーションの一種である。

【００１７】

さて、図２に関連して、従来技術にて実装されていた図１で示したウェブサービスプロバイダコンピュータ１０４のコンポーネントが、簡潔に説明される。図２に示すように、ウェブサービスプロバイダコンピュータ１０４は、インタフェース２００、実行環境２０２、およびオンデマンドコンパイルシステム２０４を含む。ウェブサービスリクエスト２０８のようなリクエストがリモートコンピュータから受信されると、インタフェース２０

０は、限定されるものではないが、リクエストを解析すること、およびそのリクエストと関連付けられたデータを実行環境２０２に渡すことを含むアクションを実行する。通常、ウェブサービスリクエスト２０８は、ＨＴＴＰのようなインターネットベースのプロトコルを介して交換されたＸＭＬベースのメッセージである。

【００１８】

実行環境２０２は、リクエスト２０８がウェブサービスによって受信されたときにプログラムコードの実行を管理するための論理およびサービスを提供すると一般的に説明される。当業者およびその他の者は、ウェブサービスを提供し、ウェブサービスと情報のやりとりをするプログラムコードが、多数の異なる高水準プログラミング言語で記述され得ることに気付くだろう。従来技術のあるシステムでは、実行環境２０２は、実行環境２０２において生成されたオブジェクトコード２１０からプログラムコードを中間プログラミング言語に翻訳する。

【００１９】

通常、ウェブサービスによってリクエストが満足されるユニットは、ページである。たとえば、リクエスト２０８は、ウェブページの形式で要求元コンピュータ１０２に返信されるアルゴリズムの結果、ウェブサービスによってアルゴリズムが実行されることを引き起こすことができる。当業者およびその他の者はページの形式でレスポンス２１２を生成するために、様々な時におよび異なる状況でオンデマンドコンパイルシステム２０４がオブジェクトコード２１０をバイナリコード２１４にコンパイルすることに気付くであろう。さらに、オブジェクトコード２１０のコンパイルは、通常、コードが最初に要求され将来の使用のためにキャッシュされたときに、発生する。従来技術のシステムの中には、オンデマンドコンパイルシステム２０４が、次のリクエストを満足させるためにメモリにキャッシュされるＤＬＬ（Dynamically Linked Library）にオブジェクトコード２１０をコンパイルするシステムがある。結果として、バイナリＤＬＬのみが“オンデマンドで”コンパイルされるので、オンデマンドコンパイルシステム２０４によって実行されるコンパイルの数は最小化される。

【００２０】

図２と関連して提供されるウェブサービスプロバイダコンピュータ１０４の説明は、非常に簡素化されたものであることを理解すべきである。さらに、図２で示されたウェブサービスプロバイダコンピュータ１０４のコンポーネントアーキテクチャーは、例示的なものとして解釈されるべきであり、なんら限定して解釈してはならない。実際、ウェブサービスプロバイダコンピュータ１０４、インタフェース２００、実行環境２０２、およびオンデマンドコンパイルシステム２０４は、図２で示されてなく付随する文章で説明されていない追加のコンポーネントおよび機能を有する。

【００２１】

さて、図３に関連して、本発明の実施態様を実装することができるウェブサービスプロバイダコンピュータ１０４の構成が説明される。図３に示すように、ウェブサービスプロバイダコンピュータ１０４は、インタフェース２００、実行環境２０２、および図２で示したオンデマンドコンパイルシステム２０４を含む。さらに、図２に関連して提供された説明と同様に、ウェブサービスリクエスト２０８は、ウェブサービスプロバイダコンピュータ１０４においてリモートコンピュータから受信される。結果として、オブジェクトコード２１０が実行環境２０２によって生成される。しかしながら、この場合に、本発明の態様では、オンデマンドコンパイルシステム２０４が、バイナリコード２１４が実行されることをスケジュールされた時期をアンチウィルスソフトウェア３００に通知することを引き起こさせる。アンチウィルスソフトウェア３００の構成は、スキャンエンジン３０２および署名データベース３０４を含み、オンデマンドコンパイルシステム２０４によって生成されたバイナリコード２１４がマルウェアを含んでいるかを判定する。

【００２２】

本発明の態様によれば、オンデマンドコンパイルシステム２０４は、バイナリコード２１４が実行されることをスケジュールされた時期をアンチウィルスソフトウェア３００に

通知するために構成される。それに応じて、アンチウィルスソフトウェア３００は、バイナリコード２１４の分析を実行して、コード２１４がマルウェアの機能を実装しているかを決定する。本発明の一実施態様においては、アンチウィルスソフトウェア３００は、マルウェアを見つけるために署名ベースシステムを実装する。この種のシステムにおいてマルウェアを識別するための技術として周知の技術は、マルウェアのコピーを“野生で(in the wild)”取得することを含む。それから、マルウェアを実装するプログラムコードは、マルウェアを固有に識別するために使用され得る“署名”にプログラムコードを変換する機能で処理される。図３で示されたスキャンエンジン３０２は、この周知の技術を採用して、バイナリコード２１４をマルウェア署名スキャンしてもよい。たとえば、署名データベース３０４に記憶されたマルウェア署名がバイナリコード２１４と比較されてもよい。しかしながら、スキャンエンジン３０２は、バイナリコード２１４がマルウェアに感染されたかを決定するために分析の追加タイプを実行するために構成されてもよい。このように、本稿で説明されていないマルウェア検出システムの他の種類がアンチウィルスソフトウェア３００にて実行されることができることを十分に理解されるべきである。

【００２３】

さて、図４に関連して、本発明の他の態様では、ウェブサービスによって処理される前にウェブサービスへのどのリクエストがマルウェアスキャンされるかが説明される。図４にて示すように、ウェブサービスプロバイダコンピュータ１０４は、図３で示したものと同様の多くのコンポーネントを含む。しかしながら、本発明の本態様においては、ウェブサービスリクエスト２０８がインタフェース２００において受信されたときには、リクエスト２０８およびリクエスト２０８に関連付けられたデータがアンチウィルスソフトウェア４００に渡される。図４に示すように、アンチウィルスソフトウェア４００は、スキャンエンジン４０２、署名データベース４０４、およびコンパイラ４０６を含む。当業者およびその他の者は、スキャンエンジン４０２および署名データベース４０６は、通常、図３に関連して上記で説明された同様の名前を有するコンポーネントと同様の機能を有することに気付くだろう。しかしながら、図４で示すスキャンエンジン４０２は、図４に関連して上記で説明されたスキャンエンジン３０２よりも追加の機能を実行することができる。たとえば、スキャンエンジン４０２は、リクエスト中のＸＭＬ構造を、悪意のＸＭＬ構造以外の正当な署名と一致するパターンを求めてスキャンする。換言すれば、スキャンエンジン４０２は、リモートコンピュータから受信したＸＭＬソースデータを、周知のマルウェアと一致するパターンまたは署名を求めて検索することができる。たとえば、当業者またはその他の者は、ＸＭＬリクエストは、組み込みＤＴＤ（Document Type Definition）構造を含むことがあることに気付くだろう。しかしながら、ＤＴＤ構造は、ＸＭＬパーサが、ウェブサービスプロバイダコンピュータ１０４において、過度のコンピュータリソースを順番に使用することを引き起こす方法で形成される場合がある。ＸＭＬパーサを使用するコンピュータを保護する本発明の一態様においては、ウェブサービスに入力された生データがマルウェアと関連付けられた認識できるパターンと一致するＸＭＬ構文スキャンされる。結果として、ＸＭＬパーサを圧倒するために設計されたネスト化されたＤＴＤ構造のようなＸＭＬ入力は、ウェブサービスによって処理される前にマルウェアとして識別される。マルウェアが誤って識別されたとき、“偽陽性(false positive)”またはインスタンスの発生を防ぐために、ウェブサービスによってマルウェアとして正常に識別され処理されたＸＭＬ構成を許可する設定パラメータが構築されることができる。このように、システム管理者のような管理のエンティティは、アンチウィルスソフトウェア４００を設定して組織の必要を満たすことができる。

【００２４】

図４にて示した本発明の態様においては、ウェブサービスリクエスト２０８が受信されると、アンチウィルスソフトウェア４００にそれが渡される。ＸＭＬソースデータがスキャンエンジン４０２によって解析された後に、コンパイラ４０６　４００は、リクエストの結果として実行され得るバイナリコードを生成する。それからスキャンエンジン４０２は署名データベース４０４からマルウェア署名を取得し、その署名と、コンパイラ４０６

によって生成されたバイナリコードとを比較する。当業者およびその他の者は、図４で示す本発明の態様は、同じ高水準コードが２回コンパイルされることを引き起こすことができることに気付くだろう。たとえば、コンパイラ４０６は、マルウェア検出の目的でバイナリコードを生成することができる。同様に、オンデマンドコンパイルシステム２０４は、アンチウィルスソフトウェア４００がマルウェアを検出しない場合には、同じバイナリコードが生成されることを引き起こすことができる。本態様において最適化は、コンパイラ４０６によって生成されるバイナリコードを、コンパイルシステム２０４を要求するために利用することができるメモリの領域にキャッシュし、または記憶する。

【００２５】

ウェブサービスプロバイダコンピュータ１０４に対して送信されるリクエストをインターセプトするために採用された技術は、ＩＳＡＰＩ（Internet Server Application Program Interface）フィルタとして実装されることができる。ＩＳＡＰＩフィルタは、ソフトウェアモジュールがイベントを登録し、ウェブサービスに対して送信されているデータストリームを編集することを許可することに、当業者および他の者は気付くだろう。本発明に関しては、ウェブサービスに対するリクエストは、マルウェア検出の目的のためにリクエストの前処理を実行するＩＳＡＰＩフィルタにおいてインターセプトされる。本発明の一実施態様において、前処理は、リクエストの結果として実行されるであろうバイナリコードを生成すること、およびこのバイナリコードがマルウェアの機能性を実装するかを決定することを含む。

【００２６】

前述したように、本発明は、バイナリコードが実行される前にバイナリコードをマルウェアスキャンする。ソースコードのような高水準言語のコードとは対照的なバイナリ形式のコードをスキャンすることによって、本発明は、マルウェアを検出するためのバイナリコードの署名ベース技術のような伝統的な技術を使用することができる。しかしながら、本発明の他の態様においては、マルウェアスキャンは、リクエストと関連付けられたコードが高水準言語である間に実行される。たとえば、上記説明されたウェブサービスにされるリクエストは、典型的にはＸＭＬメッセージプロトコルを使用する。この場合において、本発明によって提供されるアンチウィルスソフトウェアは、マルウェアの特徴である特有のＸＭＬベース文法を求めてリクエストをスキャンすることができる。さらに、ある実行環境において、リクエストと関連付けられたコードは、バイナリコードにコンパイルされる前に、中間言語に翻訳される。この場合において、マルウェアスキャンは、中間言語に翻訳された後に、コードにおいて実行される。

【００２７】

さて、図５に関連して、ウェブサービスに対するリクエストにおけるマルウェアを識別するスキャン方法５００の例示的な態様を説明する。本発明の例示的な態様において、スキャン方法５００は、ウェブリクエストを受信しコードを実行してリクエストを満足させるために設計された現存するシステムに実装される。要約すると、スキャン方法５００は、マルウェアに対するウェブサービスを提供するコンピュータを潜在的に危険にさらすことを防ぐために実行されることを必要とするマルウェアスキャンにおいて、インスタンスを識別する。コンピュータがマルウェアに危険にさらされている可能性があるときには、方法５００は、リクエストに関連付けられたいずれのコードが実行される前に、スキャンが実行されることを引き起こす。図１から４および付随する説明に継続的に関連して、例示的なスキャン方法５００は、説明されるだろう。

【００２８】

図５に示すように、スキャン方法５００は、ウェブサービスがアクセス可能にされたときにブロック５０２において開始する。当業者およびその他の者は、ウェブサービスが、ウェブサービスリクエストのようなイベントに応答してアクションが実行されるイベント駆動システムであることに気付くだろう。このように、ウェブサービスを提供するコンピュータをマルウェアから保護するために、スキャン方法５００は、ウェブサービスを提供するコンピュータがリクエストを受け付けることが可能であるときはいつでもリクエスト

と関連付けられたコードをスキャンすることができる。換言すれば、本発明の態様においては、ウェブサービスに対するリクエストを受け付けるためにコンピュータが設定されている場合はいつでも、コンピュータを保護するシステムサービスとして実装することができる。

【００２９】

判定ブロック５０４において、スキャン方法５００は、ウェブリクエストが本発明を実装するコンピュータにおいて受信されるまで、アイドル(idle)となっている。前述したように、ウェブリクエストは、多くの異なったソフトウェアシステムおよび通信プロトコルを使用して生成されることができる。当業者およびその他の者は、ウェブサービスは、通常、ウェブサービスがアクセスされ得る方式を含むウェブサービスと通信するための方法を定義するインタフェースまたはサービス記述を提供することに気付くだろう。要求元コンピュータにおけるソフトウェアシステムは、通常、ＵＲＩ（Uniform Resource Indicator）を使用するウェブサービスを識別し、そのウェブサービスによって定義されるインタフェースに対する一または複数の機能呼び出しをする。あるシステムでは、機能呼び出しをするためのデータは、ＨＴＴＰまたはＨＴＴＰＳのようなウェブベースのプロトコルを介して送信される。しかしながら、当業者およびその他の者は、本発明の要旨を逸脱しない範囲で、他のネットワークプロトコルを使用するデータが送信されることが可能であることに気付くだろう。さらには、送信されたときは、そのリクエストは、ＸＭＬのようなソフトウェアシステムの間でデータの交換を容易にする高水準マークアップ言語である。しかしながら、当業者およびその他の者は、他のマークアップ言語がウェブリクエストを作成するために使用されてもよく、ここに提供された例は、例示的なものとして解釈すべきであり、なんら限定すべきものではないことに気付くだろう。

【００３０】

図５に示すように、ブロック５０４において受信されたウェブリクエストを満足させるために実行される高水準コードが、ブロック５０６においてバイナリコードにコンパイルされる。前述したように、ウェブサービスを提供するコンピュータにおいて、バイナリコードは様々なときおよび異なる状況において生成されることができる。たとえば、図２に関連して上記で説明されたオンデマンドコンパイルシステム２０４は、バイナリコードが実行されることを必要とするときに、高水準コードをバイナリコードにコンパイルする。それから、バイナリコードは、引き続きのウェブリクエストが受信されたときに、再使用のためにメモリにキャッシュされ、または記憶される。しかしながら、高水準コードをバイナリコードにコンパイルすることは、従来技術として一般的に既知のソフトウェアシステムおよび方法を使用して実行されることができるので、方法５００の本態様の説明については、ここではさらなる詳細な説明はしない。

【００３１】

判定ブロック５０８において、スキャン方法５００は、ブロック５０６にて生成されたバイナリコードが実行されることをスケジュールされたかを判定する。高水準コードは、様々なときおよび異なる状況においてバイナリコードにコンパイルされるので、スキャン方法５００は、マルウェアスキャンを実行する前に、プログラム実行が発生することが予定されることの通知があるまで待機する。たとえば、オンデマンドコンパイルシステム２０４（図２）は、プログラム実行が発生することをスケジュールされる前に複数のバイナリＤＬＬをコンパイルすることができる。本発明の一実施態様においては、リクエストと関連付けられたバイナリコードが実行されることをスケジュールされたときに、オンデマンドコンパイルシステム２０４は、本発明を実装するアンチウィルスソフトウェアに通知する。この場合において、リクエストと関連付けられたバイナリコードが実行されることをスケジュールされると、方法５００は、以下でさらに詳細に説明されるブロック５１０に進む。リクエストと関連付けられたバイナリコードが実行をスケジュールされていなかった場合には、スキャン方法５００は、ブロック５０６に戻り、そしてブロック５０６、５０８は、リクエストと関連付けられたバイナリコードの全てが利用可能でありかつ実行されることをスケジュールされるまで繰り返す。

【００３２】
図５に示すように、ブロック５１０において、方法５００は、実行されることをスケジュールされたバイナリコードをマルウェアスキャンする。前述したように、スキャンエンジン３０２（図３）におけるソフトウェア実装されたルーチンは、バイナリコードをマルウェアスキャンするために使用されることができる。本発明の一実施態様においては、ブロック５１０で実行されるスキャンは、マルウェア“署名”に対するコードのパターンと一致することを含む。たとえば、署名データベース３０４に保持される署名は、ブロック５０４にて受信されたリクエストの結果として生成されるバイナリコードと比較されることができる。しかしながら、ブロック５１０で実行されるスキャンには、以下でより詳細に説明される発見的なマルウェア検出技術のような追加的なマルウェア識別技術が含まれてもよい。

【００３３】
多くの既知のツールによって、開発者が高水準言語であるコード上において困難または不可能であるバイナリコードの分析を実行することが許可される。たとえば、バイナリコードの分析は、バイナリコードによってされるオペレーティングシステムのＡＰＩに対する機能呼び出しを識別することを実行され得る。さらに、オペレーティングシステムの中には、個別ＡＰＩは、それぞれＡＰＩを実行するために要求される特権を示す許可レベルを割り当てられるものがある。ウェブサービスに対してされるリクエストは、通常、システムまたは管理者特権を必要とするオペレーティングシステムに対する呼び出しを要求しない。より、一般的には、ウェブサービスリクエストは、リソースの限られた“三度ボックス(sandbox)”にアクセスすることを期待される。逆に言えば、高い特権的なレベルを必要とするオペレーティングシステムに対するＡＰＩ呼び出しは、“疑わしい”ものであり、マルウェアの特徴であるかもしれない。このように、ブロック５１０で実行されるスキャンは、これらの“疑わしい”発見的な要因のタイプを識別することを含むことができる。この関連で、バイナリコードの分析を実行する既知のツールは、バイナリコードの特徴を検出するために使用される。

【００３４】
マルウェアのスキャンされるバイナリコードは異なるソースから由来するものであり得ることには留意されたい。たとえば、前述したように、マルウェア作者はＤｏＳ攻撃を引き起こすリクエストをウェブサービスに対して渡すかもしれない。この場合において、ウェブサービスプロバイダコンピュータにおけるソフトウェアルーチンは、リクエストにおいて提供されたデータを受け入れ、情報のやりとりを行う。しかしながら、リクエストは、ウェブサービスプロバイダコンピュータに実装されたコードが過度のコンピュータリソースを消費することを引き起こす方法で構成される。換言すれば、ウェブサービスプロバイダコンピュータによって実装された良性コード以外は、マルウェアの機能性を実装するために操作される。

【００３５】
あるいは、マルウェアは、ウェブサービスに対するリクエストにおいてリモートコンピュータから直接取得されるかもしれない。たとえば、あるウェブサービスは、ＸＭＬフォーマットにおけるデータがどのように表示されるかを定義するＸＳＬ(Extensible Stylesheet Language)におけるコードを受け付ける。さらに、ＸＳＬはデータから書式設定を切り離しているため、ＸＭＬをＨＴＭＬのような他のマークアップ言語に変換するために一般的に使用される。いずれのイベントにおいても、ＸＳＬのような言語からの高水準コードがウェブサービスによって受け付けられるとき、高水準コードは、バイナリコードにコンパイルされ、最終的には実行される。この場合に、ウェブサービスに対するリクエストをしたリモートコンピュータから完全にマルウェアが生み出される。本発明は、バイナリコードをスキャンするので、マルウェアが生み出される場所にかかわらずマルウェアは検出されることができる。

【００３６】
図５に示すように、判定ブロック５１２において、方法５００は、マルウェアがブロッ

ク５１０で識別されたかを判定する。マルウェアが識別された場合には、方法５００は、ブロック５１４に進む。ブロック５１４では、ウェブサービスを提供するコンピュータにおいてマルウェアの受信が処理される。当業者およびその他の者は、ウェブサービスにおけるマルウェアの受信は、多くの異なる方法で処理されることが可能であることに気付くだろう。たとえば、マルウェアリクエストを生成したコンピュータの同一性が確認され、既知のマルウェア作者の“ブラックリスト”に追加されるだろう。この場合に、リクエストが生み出されたコンピュータは将来ウェブサービスへアクセスすることを拒否されるであろう。しかしながら、マルウェアの受信および同一性確認は従来技術において周知の他の方法を使用することで処理されることもできる。その後、方法５００は、終点のブロック５１８に進む。マルウェアがブロック５１０で識別されなかった場合には、スキャン方法５００は、ブロック５１６に進み、そこでブロック５０４で受信したウェブリクエストを満足させることを必要とされるバイナリコードが実行される。ウェブリクエストの結果として生成されるバイナリコードを実行するシステムは、従来技術において周知であるため、これらのシステムのさらなる詳細な説明はここでは提供されない。

【００３７】

本発明の実装は、図５で示した例示的な方法５００に限定されることはない。他の方法は、追加のアクションを含んでもよいし、または、示されたいくつかのアクションを削除してもよい。また、他の方法は、図５で示したよりも異なる順序でアクションを実行してもよい。たとえば、図５で示す例示的な方法５００は、高水準コードがオンデマンドコンパイルシステムによってバイナリコードにコンパイルされるシステムとの関連で説明される。一旦、バイナリコードが実行されることをスケジュールされると、方法５００によってバイナリコードのスキャンが実行される。しかしながら、図４と関連して前述したように、本発明は、ウェブサービスに向けられるデータストリームがインターセプトされるフィルタとして実装されることができる。この場合において、高水準コードはコンパイルされ、ウェブサービスによって受信される前にマルウェアスキャンをする。このように、図５にて描かれた方法５００および付随して説明される文章は、本発明の一実施態様であり、他の実施態様も可能である。

【００３８】

好ましい発明の実施態様を説明したが、本発明の精神から逸脱することなく様々な変更がなされることが可能であることを理解されたい。

【図面の簡単な説明】

【００３９】

【図１】先行技術の態様を説明するに適した、要求元コンピュータおよびウェブサービスプロバイダコンピュータを含むネットワーキング環境の描写図である。

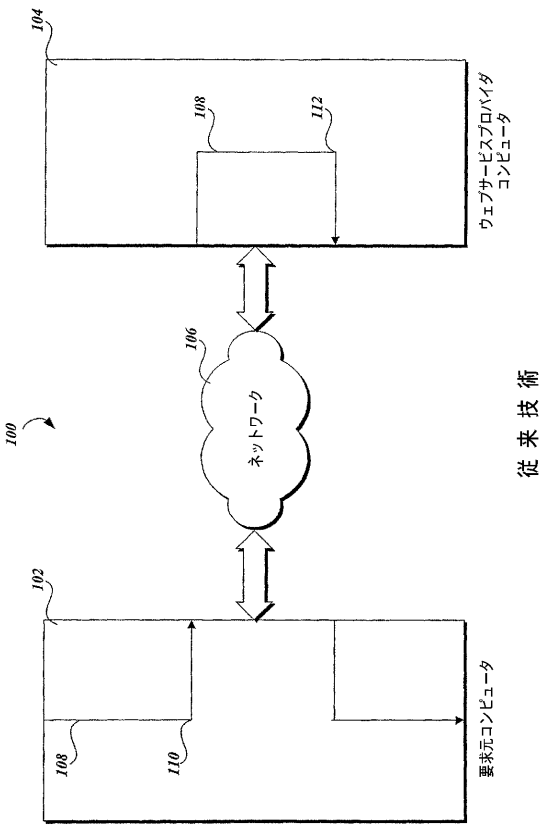
【図２】先行技術におけるウェブサービスリクエストを満足させることができるウェブサービスプロバイダコンピュータの構成を説明するブロック図である。

【図３】本発明の一実施態様におけるマルウェアに感染されていないウェブサービスリクエストを満足させることができるウェブサービスプロバイダコンピュータの構成を説明するブロック図である。

【図４】本発明の他の実施態様におけるマルウェアに感染されていないウェブサービス利ウエストを満足させることができるウェブサービスプロバイダコンピュータの構成を説明するブロック図である。

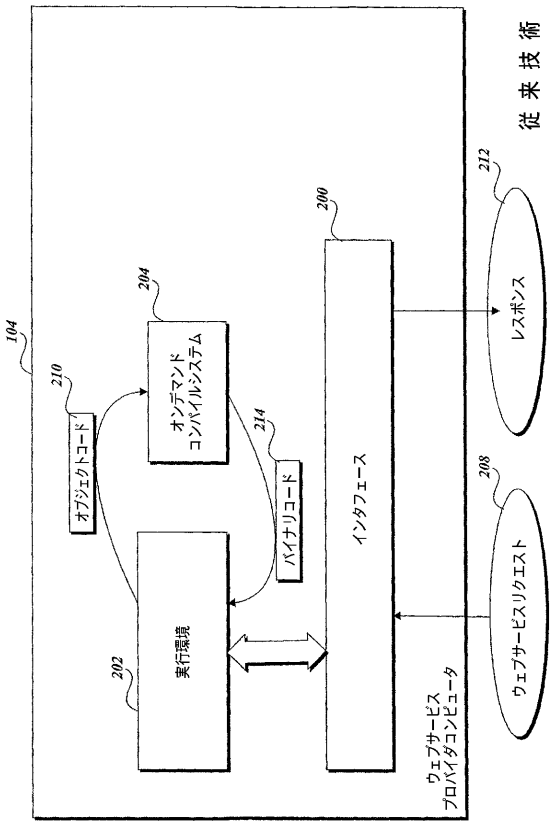
【図５】本発明におけるウェブサービスに対して作成されたリクエストにおけるマルウェアを識別するための方法を実装されたソフトウェアの例示的な一態様を説明するフロー図である。

【図１】



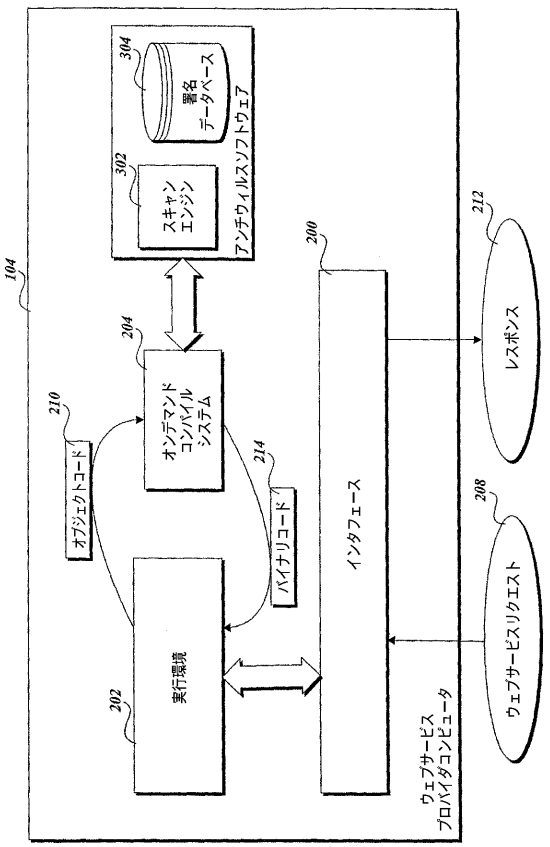
従来技術

【図２】

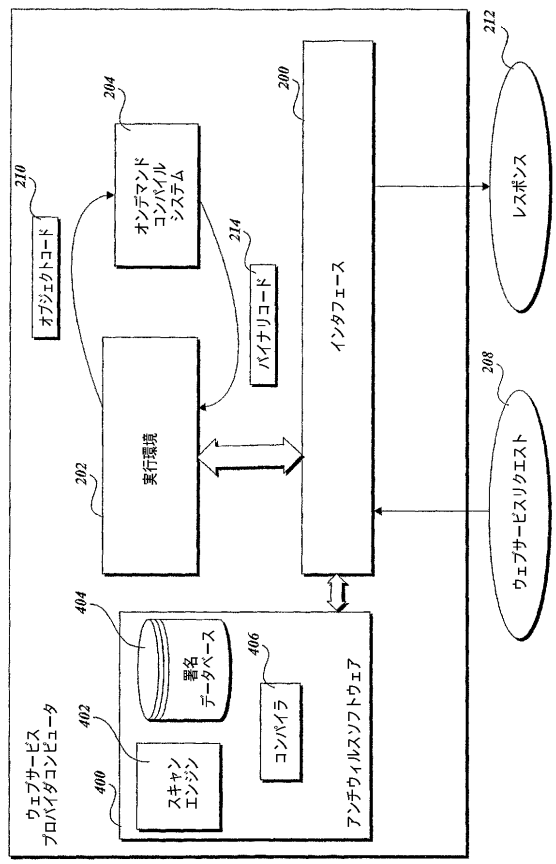


従来技術

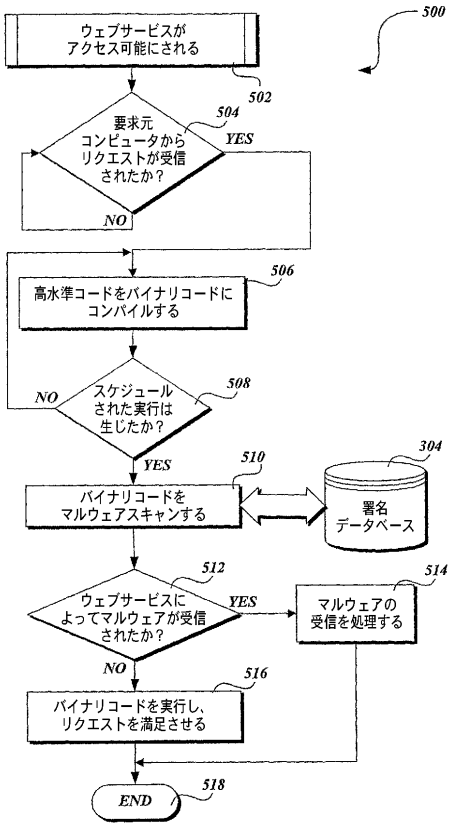
【図３】



【図４】



【図５】


500
ウェブサービスが
アクセス可能にされる
502
504
要求元
コンピュータから
リクエストが受信
されたか？
YES
NO
506
高水準コードをバイナリコードに
コンパイルする
508
NO
スケジュール
された実行は
生じたか？
YES
510
304
バイナリコードを
マルウェアスキャンする
署名
データベース
512
514
ウェブサービスに
よってマルウェアが受信
されたか？
YES
マルウェアの
受信を処理する
NO
516
バイナリコードを実行し、
リクエストを満足させる
518
END

フロントページの続き

(72)発明者 アドリアン エム. マリネスク
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 チャールズ ダブリュ. カウフマン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ジェフリー エム. クーパースタイン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 マイケル クレイマー
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

審査官 後藤 彰

(56)参考文献 特開２００２－３４２２７９（ＪＰ，Ａ）
特開２００５－０５６００７（ＪＰ，Ａ）
特開平１１－１１９９２７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 21/22
G06F 21/00