(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-146566
(P2010-146566A)
(43)公開日　平成22年7月1日(2010.7.1)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/22	(2006.01)	G06F	9/06	660N	5B276

審査請求 未請求　請求項の数 20　ＯＬ　外国語出願　（全 25 頁）

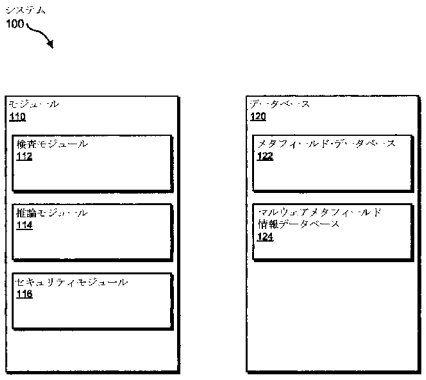
(21)出願番号　特願2009-286595 (P2009-286595)
(22)出願日　平成21年12月17日 (2009.12.17)
(31)優先権主張番号　12/338,877
(32)優先日　平成20年12月18日 (2008.12.18)
(33)優先権主張国　米国 (US)

(71)出願人　509345095
シマンテック　コーポレイション
アメリカ合衆国　カリフォルニア州　９４０４３　マウンテン　ビュー　エリスト　リート　３５０
(74)代理人　100147485
弁理士　杉村　憲司
(74)代理人　100134005
弁理士　澤田　達也
(74)代理人　100151677
弁理士　播磨　里江子
(72)発明者　マーク　ケネディー
アメリカ合衆国　カリフォルニア州　９０２７８　レドンドビーチ　モーガン　レーン　１９０４　ナンバービー

Ｆターム(参考)　5B276 FD08

(54)【発明の名称】マルウェア検出方法およびシステム

(57)【要約】
【課題】マルウェアを検出する方法を開示する。
【解決手段】本発明方法は、複数の既知の無害な実行可能ファイルにおける複数のメタデータフィールドの検査ステップを有する。本発明方法は、さらに、複数の既知の悪意がある実行可能ファイルにおける複数のメタデータフィールドの検査ステップも有する。本発明方法は、さらに、複数の既知の無害な実行可能ファイル、および既知の悪意がある実行可能ファイルにおける複数のメタデータフィールドの検査によって得られた情報に基づき、マルウェアを示すメタデータフィールド属性を推論するステップも有する。対応するシステムおよびコンピュータ読み取り可能な媒体も開示する。
【選択図】図１



【特許請求の範囲】
【請求項１】
マルウェア検出をコンピュータで実現する方法において、
複数の既知の無害実行可能ファイルにおける複数のメタデータフィールドを検査するステップと、
複数の既知の悪意実行可能ファイルにおける複数のメタデータフィールドを検査するステップと、
前記複数の既知の無害実行可能ファイル、および既知の悪意実行可能ファイルにおける複数のメタデータフィールドの検査から得られた情報に基づき、マルウェアを示すメタデータフィールド属性を推論するステップと
を有する、マルウェア検出をコンピュータで実現する方法。
【請求項２】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、さらに、
未知の実行可能ファイルを受け取るステップと、
前記未知の実行可能ファイルがマルウェアを示すメタデータフィールド属性を含んでいるか否かを決定することによって、前記未知の実行可能ファイルがマルウェアを含んでいるか否かを決定するステップと
を有する、方法。
【請求項３】
請求項２に記載のマルウェア検出をコンピュータで実現する方法において、さらに、
前記未知の実行可能ファイルがマルウェアを含んでいる場合、セキュリティアクションを実施するステップを有する、
方法。
【請求項４】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、
前記複数の既知の無害実行可能ファイルはポータブル実行可能ファイルを有し、
前記複数の既知の悪質実行可能ファイルはポータブル実行可能ファイルを有し、
前記メタデータフィールド属性はヘッダフィールド属性を有する、
マルウェア検出をコンピュータで実現する方法。
【請求項５】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、
前記マルウェアを示すメタデータフィールド属性は、以下の属性、すなわち、
デバッグ・セクション属性、
インポート属性、
シンボル・テーブル属性、
オプショナル・ヘッダ属性、
特性属性、
イメージ・サブシステム属性、
リンカ・バージョン属性、
サイズ属性、
リアル・バーチャル・アドレス属性、
エントリー・ポイント属性、
コード・セクション・ベース属性、
アライメント属性、
オペレーティング・システム・バージョン属性、
イメージ・バージョン属性、
ミニマム・サブシステム・バージョン属性、
ダイナミック・リンク・ライブラリ・特性属性、
スタック・サイズ属性、
ヒープ・サイズ属性、

セクション個数属性、
ノー・イン・アウト属性、
スレッド・レベル・スペキュレーション属性、
ベース・オブ・イメージ属性
のうち少なくとも一つを有する、方法。
【請求項６】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、マルウェアを示す前記メタデータフィールド属性は、前記複数の既知の無害実行可能ファイルおよび既知の悪意実行可能ファイル内で検査された、前記複数のメタデータフィールドからの属性のサブセットを有する、方法。
【請求項７】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、マルウェアを示す前記メタデータフィールド属性は静的属性を有する、方法。
【請求項８】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、マルウェアを示すメタデータフィールド属性を推論するステップは、マルウェアを示すメタデータフィールド属性の少なくとも一つの組み合わせを決定するステップを有する、方法。
【請求項９】
請求項１に記載のマルウェア検出をコンピュータで実現する方法において、マルウェアを示すメタデータフィールド属性を推論するステップは、マルウェアを示す第１メタデータフィールド属性が、マルウェアを示す第２メタデータフィールド属性よりも強い指標であることを決定するステップを有する、方法。
【請求項１０】
請求項１に記載のマルウェア検出をコンピュータで実施する方法において、少なくとも一つのコンピュータ読み取り可能な媒体における、コンピュータ実行可能命令として明確に具現化した、方法。
【請求項１１】
マルウェア検出をコンピュータで実現する方法において、
複数の既知の無害なポータブル実行可能ファイルにおける複数のメタデータフィールドを検査するステップと、
複数の既知の悪意があるポータブル実行可能ファイルにおける複数のメタデータフィールドを検査するステップと、
前記複数の既知の無害なポータブル実行可能ファイル、および既知の悪意があるポータブル実行可能ファイルにおける前記複数のメタデータフィールドの検査から得られた情報に基づき、マルウェアを示すメタデータフィールド属性を推論するステップと、
未知の実行可能ファイルを受け取るステップと、
前記未知の実行可能ファイルがマルウェアを示すメタデータフィールドを含んでいるか否かを決定することによって、前記未知の実行可能ファイルがマルウェアを含んでいるか否かを決定するステップと
を有する、マルウェア検出をコンピュータで実現する方法。
【請求項１２】
請求項１１に記載のマルウェア検出をコンピュータで実現する方法において、
マルウェアを示すメタデータフィールド属性は、以下の属性、すなわち
デバッグ・セクション属性、
シンボル・テーブル属性、
イメージ・サブシステム属性、
リンカ・バージョン属性、
リアル・バーチャル・アドレス属性、
オペレーティング・システム・バージョン属性、
イメージ・バージョン属性、

ミニマム・サブシステム・バージョン属性、
ダイナミック・リンク・ライブラリ・特性属性、
スタック・サイズ属性、
ヒープ・サイズ属性、
セクション数属性、
のうち、少なくとも１つを含むものとした、方法。
【請求項１３】
請求項１１に記載のマルウェア検出をコンピュータで実現する方法において、
マルウェアを示す前記メタデータフィールド属性は、以下の整数、すなわち
セクション・アライメント整数、
ファイル・アライメント整数、
メジャー・オペレーティング・システム・バージョン整数、
マイナー・オペレーティング・システム・バージョン整数、
メジャー・イメージ・バージョン整数、
マイナー・イメージ・バージョン整数、
メジャー・サブシステム・バージョン整数、
マイナー・サブシステム・バージョン整数、
イメージ・サイズ整数、
ヘッダ・サイズ整数、
グラフィカル・ユーザーインタフェース整数、
キャラクタ・ベース・ユーザーインタフェース整数、
サイズ・オブ・スタック・リザーブ整数、
サイズ・オブ・スタック・コミット整数、
サイズ・オブ・ヒープ・リザーブ整数、
サイズ・オブ・ヒープ・コミット整数、
シンボル・テーブル・ポインタ整数、
シンボル個数整数、
デバッグ・セクション整数、
メジャー・リンカ・バージョン整数、
マイナー・リンカ・バージョン整数、
コード・セクション・サイズ整数、
初期化・データ・サイズ整数、
非初期化・データ・サイズ整数、
リアル・バーチャル・アドレス・エントリー・ポイント整数、
リアル・バーチャル・アドレス・スタート・オブ・コード・セクション整数、
リアル・バーチャル・アドレス・ベース・オブ・コード・セクション整数、
リアル・バーチャル・アドレス・スタート・オブ・データ・セクション整数、
ベース・オブ・イメージ整数、
ハズ・インポート整数、
ハズ・ディレイ・インポート整数、
エクスターナル・バインディング・ファシリティ整数、
ノー・イン・アウト整数、
ｕｒｌｍｏｎ・インポート整数、
スレッド・レベル・スペキュレーション整数、
ｍｓｖｃｒｔ・インポート整数、
ｏｌｅａｕｔ３２・インポート整数、
ｓｅｔｕｐａｐｉ・インポート整数、
ｕｓｅｒ３２・インポート整数、
ａｄｖａｐｉ３２・インポート整数、
ｓｈｅｌｌ３２・インポート整数、

ｇｄｉ３２・インポート整数、
ｃｏｍｄｌｇ３２・インポート整数
ｉｍｍ３２・インポート整数
ハズ・サーティフィケート整数、
ノード整数
セクション個数整数
のうち、少なくとも２つを含むものとした、方法。
【請求項１４】
請求項１１に記載のマルウェア検出をコンピュータで実現する方法において、さらに、
前記未知の実行可能ファイルがマルウェアを含む場合にセキュリティアクションを実施するステップを有し、
前記セキュリティアクションは、以下のステップ、すなわち、
前記未知の実行可能ファイルを隔離するステップと、
前記未知の実行可能ファイルをセキュリティソフトウェアベンダーに報告するステップと、
前記未知の実行可能ファイルをマルウェアファイルのリストへ追加するステップと
のうち、少なくとも一つを有する、方法。
【請求項１５】
請求項１１に記載のマルウェア検出をコンピュータで実現する方法において、少なくとも一つのコンピュータ読み取り可能な媒体における、コンピュータ実行可能命令として明確に具現化した、方法。
【請求項１６】
検査モジュールであって、
複数の既知の無害な実行可能ファイルにおける複数のヘッダフィールドを検査し、
複数の既知の悪意がある実行可能ファイルにおける複数のヘッダフィールドを検査する
ようプログラムした、該検査モジュールと、
前記検査モジュールと通信し、前記検査モジュールによって得られた情報を記憶するよう構成したデータベースと、
前記検査モジュールから得られた前記情報に基づき、マルウェアを示すヘッダフィールド属性を推論するようプログラムした推論モジュールと
を備えたシステム。
【請求項１７】
請求項１６に記載のシステムにおいて、さらに、
セキュリティシステムであって、
未知の実行可能ファイルを受け取り、
前記未知の実行可能ファイルがマルウェアを示す前記ヘッダフィールド属性を含んでいるか否かを決定することによって、前記未知の実行可能ファイルがマルウェアを含んでいるか否かを決定する
ようにプログラムした、該セキュリティシステムを有する、システム。
【請求項１８】
請求項１７に記載のシステムにおいて、前記セキュリティモジュールは、さらに、前記未知の実行可能ファイルがマルウェアを含んでいる場合に、セキュリティアクションを実施するようプログラムした、システム。
【請求項１９】
請求項１６に記載のシステムにおいて、前記ヘッダフィールド属性は、以下の属性、すなわち、
デバッグ・セクション属性、
インポート属性、
シンボル・テーブル属性、

オプショナル・ヘッダ属性、
特性属性、
イメージ・サブシステム属性、
リンカ・バージョン属性、
サイズ属性、
リアル・バーチャル・アドレス属性、
エントリー・ポイント属性、
コード・セクション・ベース属性、
アライメント属性、
オペレーティング・システム・バージョン属性、
イメージ・バージョン属性、
ミニマム・サブシステム・バージョン属性、
ダイナミック・リンク・ライブラリ・特性属性、
スタック・サイズ属性、
ヒープ・サイズ属性、
セクション個数属性、
ノー・イン・アウト属性、
スレッド・レベル・スペキュレーション属性、
ベース・オブ・イメージ属性
のうち、少なくとも一つを含む、システム。

【請求項２０】
請求項１６に記載のシステムにおいて、マルウェアを示す前記ヘッダフィールド属性は静的属性を含む、システム。

【発明の詳細な説明】
【技術分野】
【０００１】
消費者および企業は、機密データを保存するのに、ますますコンピュータに依存するようになってきた。よって悪意あるプログラマーたちは、他者のコンピュータに対して違法な制御およびアクセスを得ることに、ますます力を傾注し続けているようである。悪意を抱くコンピュータプログラマーたちは、他人のコンピュータシステムおよびデータに危害を及ぼすウィルス、トロイの木馬、ワームおよびその他のプログラムをこれまでに作り、また作り続けている。これら悪意のあるプログラムは、通常マルウェアと称される。

【背景技術】
【０００２】
セキュリティソフトウェア会社は、顧客に対して日頃からマルウェア署名（例えば、マルウェアを識別するハッシュ機能）を作成し、また展開することによって、マルウェアの高まる風潮と闘っている。しかしながら、膨大な量のマルウェアがまだ識別されていない。従って必要なのは、未確認のマルウェアを検出する方法（プロセス）である。

【発明の概要】
【発明が解決しようとする課題】
【０００３】
本発明の実施形態は、悪意があるファイル（以下「悪意ファイル」と称する）の一つ以上のメタデータフィールド属性に基づく悪意があるデータ（以下「悪意データ」と称する）の検出を目的とする。例えば、検査モジュールは、複数の既知の無害な実行可能ファイルの複数のメタデータフィールドを検査することができる。検査モジュールはまた、複数の既知の悪意がある実行可能ファイル（以下「悪意実行ファイル」と称する）の複数のメタデータフィールドも検査することができる。推論モジュールは、複数の既知の無害な実行可能ファイル（以下「無害実行ファイル」と称する）および既知の悪意実行ファイルにおける複数のメタデータフィールドを検査することによって収集した情報に基づいて、マルウェアを示すメタデータフィールド属性を推論することができる。

【０００４】
いくつかの実施形態では、セキュリティモジュールは、複数の既知の無害実行ファイル、および既知の悪意実行ファイルにおけるメタデータフィールドの検査から推論された情報を使って、未知の実行可能ファイル（以下「未知実行ファイル」と称する）がマルウェアを含んでいるか否かを判定することができる。例えば、セキュリティモジュールは、未知実行ファイルを受け取る可能性がある。このとき、セキュリティモジュールは、その未知実行ファイルがマルウェアを示すメタデータフィールド属性を含んでいるか否かを決定することによって、その未知実行ファイルがマルウェアを含んでいるか否かを決定することができる。

【課題を解決するための手段】

【０００５】
少なくとも一つの実施形態では、セキュリティモジュールは、未知実行ファイルがマルウェアを含んでいるのであれば、セキュリティアクションを実施することができる。セキュリティモジュールは、未知実行ファイルの隔離、未知実行ファイルのセキュリティベンダーへの報告、未知実行ファイルのマルウェアファイルリストへの追加、および／または他の適切なセキュリティアクションを実施することによって、セキュリティアクションを実施することができる。上述の実施形態の機能は、本明細書に記載の一般原理に基づいて、互いに組み合わせて使用することができる。これら実施形態および他の実施形態、機能および利点は、添付図面および特許請求の範囲と合わせて下記の説明を読むことによって、より完全に理解することができるであろう。

【０００６】
添付図面は多数の実施例を示し、これら図面は本明細書の一部である。以下の説明とともに、これら図面は、本発明の様々な原理を提示し、また説明するものである。

【図面の簡単な説明】

【０００７】
【図１】ある実施形態によるマルウェア検出の例示的なシステムのブロック図である。
【図２】ある実施形態によるマルウェア検出の例示的な方法のフローチャートである。
【図３】ある実施形態による例示的な実行可能ファイルのブロック図である。
【図４】ある実施形態によるマルウェア検出の例示的な方法のフローチャートである。
【図５】本明細書に記載および／または説明する一つ以上の実施例を実行することのできるコンピューティングシステム例のブロック図である。
【図６】本明細書に記載および／または説明する一つ以上の実施形態を実行することのできる、例示的なコンピューティングネットワークのブロック図である。

【０００８】
図面全体にわたり、同一の参照符号および説明は、類似するが、必ずしも同じものとは限らない要素を示している。本明細書に記載の実施例は、様々な変更した形態または代替的形態となり得るが、図面には例として特定の実施形態を示し、本明細書において詳細を示す。しかし、本明細書に記載の実施例は、開示された特定の形態に限定することを意図するものではない。むしろ本発明は、添付する特許請求の範囲に該当する、あらゆる変更形態、等価形態、代替的形態をカバーする。

【発明を実施するための形態】

【０００９】
以下に、より詳細に示すように、本発明は、一般に、既知の無害な実行可能ファイル（無害実行可能ファイル）と既知の悪意のある実行可能ファイル（悪意実行可能ファイル）とを検査し、そして検査中に得られる情報を使ってマルウェアの検出を行って、未知のファイルがマルウェアを含んでいるか否かを決定する方法およびシステムに関する。図１は、既知の無害実行可能ファイルと既知の悪意実行可能ファイルを検査し、そして未知の実行可能ファイル（未知実行可能ファイル）がマルウェアを含んでいるか否かを決定する例示的なシステムの図である。図２は、既知の無害実行可能ファイルおよび既知の悪意実行可能ファイルを検査し、マルウェアを示すメタデータ属性を推論する例示的なプロセスを

示す。図３は例示的なポータブルの実行可能(ＰＥ)ファイルを示し、図４はＰＥファイルがマルウェアを含んでいるか否かを決定する例示的なプロセスを示す。本発明の実施形態を実現する例示的なコンピューティングシステムおよび例示的なネットワークを、図５および図６に示す。

【００１０】

図１は例示的なシステム１００のブロック図である。システム１００は一つ以上のタスクを実施する一つ以上のモジュール１１０を含むことができる。例えば、モジュール１１０は、既知の無害実行可能ファイルおよび既知の悪意実行可能ファイルのメタデータフィールドを検査する検査モジュール１１２を含むことができる。実行可能フィールドのメタデータフィールドは、実行可能ファイルのヘッダフィールドおよび／またはその他のフィールドを含むことができる。モジュール１１０はまた、既知の無害実行可能ファイルおよび既知の悪意実行可能ファイルを検査することによって得られる情報に基づいて、マルウェアを示すメタデータフィールド属性を推論する、推論モジュール１１４を含むこともできる。モジュール１１０は、マルウェアを示すメタデータフィールド属性を使って未知実行可能ファイルがマルウェアを含んでいるか否かを決定することのできる、セキュリティモジュール１１６を含むことができる。

【００１１】

ある実施形態では、図１の一つ以上のモジュール１１０は、コンピューティングシステムによって実行されるとき、本明細書に開示する一つ以上のステップをコンピューティングシステムに実施させる、一つ以上のソフトウェアアプリケーションまたはプログラムを表すことができる。例えば、以下により詳しく説明するように、一つ以上のモジュール１１０は、図５に示すコンピューティングシステム５１０のような一つ以上のコンピューティング装置および／または図６に示す例示的なネットワークアーキテクチャ６００の一部で実行するために構成した、ソフトウェアモジュールを表すことができる。図１の一つ以上のモジュール１１０はまた、本明細書に開示するステップと関連する一つ以上のタスクを実行するように構成した、一つ以上の特別用途コンピュータの全体または一部も表すことができる。

【００１２】

システム１００はデータベース１２０を含むことができる。データベース１２０は、既知の無害実行可能ファイルおよび既知の悪意実行可能ファイル内で検査される一つ以上のメタデータフィールドのリストを記憶することのできる、メタデータフィールド・データベース１２２を含むことができる。データベース１２０はまた、マルウェアメタデータフィールド情報データベース１２４を含むことができる。マルウェアメタデータフィールド情報データベース１２４は、マルウェアを示すメタデータフィールド属性および／またはメタデータフィールド属性がどのようにマルウェアを示すのかを識別するどんな情報も含むことができる。マルウェアを示すメタデータフィールド属性は、実行可能ファイル内で見つかった場合、その実行可能ファイルがマルウェアを含んでいることを示すことのできる一つ以上のメタデータフィールド属性を含むことができる。

【００１３】

いくつかの実施形態では、マルウェアメタデータフィールド情報データベース１２４は、マルウェアのメタデータフィールド属性をどのように利用して、未知の実行可能ファイルがマルウェアを含んでいるか否かを決定するのかを示す情報を含むことができる。ある実施形態によると、マルウェアメタデータフィールド情報データベース１２４は、任意の適切な機械学習アルゴリズムを既知の無害実行可能ファイルおよび既知の悪意実行可能ファイルに適用することから得られる情報を含むことができる。例えば、マルウェアメタデータフィールド情報データベース１２４は、マルウェアの検出におけるメタデータ属性の有用性を示す、一つ以上のメタデータフィールド属性の重み付け情報を含むことができる。

【００１４】

いくつかの実施形態では、マルウェアメタデータフィールド情報データベース１２４は

、閾値を示す閾値情報を含むことができる。セキュリティモジュール１１６は、マルウェアメタデータフィールド情報データベース１２４のメタデータフィールド属性と一致する実行可能ファイルのメタデータフィールド属性の数が閾値以上の場合、実行可能ファイルがマルウェアを含んでいると決定することができる。マルウェアメタデータフィールド情報データベース１２４は、付加的または代替的に、マルウェアを示すメタデータフィールド属性の一つ以上の組み合わせを示す情報を含むことができる。いくつかの実施形態では、マルウェアメタデータフィールド情報データベース１２４は、マルウェアを示すあるメタデータフィールド属性が、マルウェアを示す別のメタデータフィールド属性よりも強力なマルウェアの指標であることを示す情報を含むことができる。

【００１５】

図１の一つ以上のデータベース１２０は、一つ以上のコンピューティング装置の一部分を示す。一つ以上のデータベース１２０は、図５に示すコンピューティングシステム５１０の一部分および／または図６に示す例示的なネットワークアーキテクチャ６００のいくつかの部分を示す。あるいは、図１の一つ以上のデータベース１２０は、コンピューティング装置によってアクセスすることのできる一つ以上の物理的に分かれた装置を示す。

【００１６】

図２は検査モジュール１１２などの検査モジュール、推論モジュール１１４などの推論モジュールおよび／またはセキュリティモジュール１１６などのセキュリティモジュールによって実現することのできるプロセスを示す。検査モジュールは、複数の既知の無害実行可能ファイルにおける複数のメタデータフィールドを検査することができる(ステップ２１０)。例えば、検査モジュールは、２つ以上の既知の無害実行可能ファイルにおける２つ以上のメタデータフィールドを検査することができる。

【００１７】

既知の無害実行可能ファイルは、無害であると識別された任意のファイル（すなわち、マルウェアを含まないファイル）を含む。実行可能ファイルのメタデータフィールドは、実行可能ファイルに関するおよび／またはそれに関連する情報を持つどんなフィールドであってもよい。例えば、実行可能ファイルのメタデータフィールドにおける入力事項は、実行可能ファイルの属性を示すことができる。

【００１８】

いくつかの実施形態では、メタデータフィールドはまた、実行可能ファイルの静的属性を含むことができる。本明細書において、用語「静的属性」とは、ファイルが実行されていないときに観測できるファイルの属性に言及する。つまり、静的属性は、実行可能ファイルの情報の検査に基づいて決定できる属性である。逆に、動的属性は、ファイルの実行に基づいて観測される属性である。

【００１９】

実行可能ファイルは、コンピュータが実行できるコード（つまり、命令）を含む任意のファイルとすることができる。実行可能ファイルはまた、実行ファイルまたはバイナリと称することもできる。実行可能ファイルは、適切な実行可能ファイルフォーマットによってフォーマットすることができる。実行可能ファイルフォーマットの例として、ポータブル実行可能（ＰＥ）ファイルフォーマットがある。ＰＥファイルフォーマットは、ＷＩＮＤＯＷＳ（登録商標）オペレーティングシステムの３２ビットおよび６４ビットのバージョンに使用されるファイルフォーマットとすることができる。実行可能ファイルはまた、ＬＩＮＵＸ（登録商標）オペレーティングシステム、ＭＡＣ（登録商標）オペレーティングシステム、ＵＮＩＸ（登録商標）オペレーティングシステムおよび／または他のオペレーティングシステムの実行ファイルを含むことができる。

【００２０】

既知の無害実行可能ファイルの検査に加え、検査モジュールは、複数の既知の悪意実行可能ファイルにおける複数のメタデータフィールドを検査することができる(ステップ２２０)。例えば検査モジュールは、２つ以上の既知の悪意実行可能ファイルにおける２つ以上のメタデータフィールドを検査することができる。既知の悪意実行可能ファイルとし

ては、コンピュータシステムおよびデータに危害を及ぼすウィルス、トロイの木馬、ワームおよび／または他のプログラムのような、マルウェアを含むことで知られるどんな実行可能ファイルがある。いくつかの実施形態では、検査モジュールは、既知の無害実行可能ファイルおよび既知の悪意実行可能ファイル内で同じメタデータフィールドを検査することができる。他の実施形態では、検査モジュールは、既知の悪意実行可能ファイルよりも既知の無害実行可能ファイル内で、一つ以上の様々なメタデータフィールドを検査することができる。

【００２１】

検査モジュールは、実行可能ファイルのメタデータフィールド内の属性値を決定することによって、実行可能ファイルのメタデータフィールドを検査することができる。検査モジュールは、検査中に収集した情報（例えば、属性値）を、例えばマルウェアメタデータフィールド情報データベース１２４のようなデータベースに記憶することができる。

【００２２】

検査モジュールが既知の無害実行可能ファイルおよび既知の悪意実行可能ファイルを検査した後に、推論モジュールは、検査中に得た情報に基づいて、マルウェアを示すメタデータフィールド属性を推論することができる。先にも述べたように、検査中に得た情報は属性情報を含むことができる。推論モジュールは属性情報の処理を行い、どの属性がマルウェアを示すのかを決定することができる。推論モジュールは、適切な機械学習アルゴリズムを使って、どの属性がマルウェアを示しているかを決定することができる。推論モジュールはまた、未知の実行可能ファイルがマルウェアを含んでいるか否かを決定するのにメタデータ属性をどのように使用するのかを決定するために、任意の適切な機械学習アルゴリズムも使うことができる。例えば、推論モジュールは、メタデータ属性のどの組み合わせが悪意実行可能ファイルを示すのかを決定することができる。推論モジュールは、マルウェアメタデータフィールド情報データベース１２４のようなデータベースに、マルウェアを識別するためのメタデータ属性の使用法に関するあらゆる情報を記憶することができる。

【００２３】

実行可能ファイルは、様々なタイプのメタデータフィールドおよびそれに対応する属性を含むことができる。例えば、実行可能なファイルとしては、一つ以上のデバッグ・セクション属性に関する１つ以上のデバッグ・セクションフィールド、１つ以上のインポート属性に関する１つ以上のインポートフィールド、１つ以上のシンボル・テーブル属性に関する１つ以上のシンボル・テーブルフィールド、１つ以上のオプショナル・ヘッダ属性に関する１つ以上のオプショナル・ヘッダフィールド、１つ以上の特性属性に関する一つ以上の特性フィールド、１つ以上のイメージ・サブシステム属性に関する１つ以上のイメージ・サブシステムフィールド、１つ以上のベース・オブ・イメージ属性に関する１つ以上のベース・オブ・イメージフィールド、１つ以上のリンカ・バージョン属性に関する１つ以上のリンカ・バージョンフィールド、１つ以上のサイズ属性に関する１つ以上のサイズフィールド、および／または１つ以上のリアル・バーチャル・アドレス（ＲＶＡ）属性に関する１つ以上のＲＶＡフィールドがある。

【００２４】

実行可能ファイルとしては、さらに、１つ以上のエントリー・ポイント属性に関する１つ以上のエントリー・ポイントフィールド、１つ以上のコード・セクション・ベース属性に関する１つ以上のコード・セクション・ベースフィールド、１つ以上のノー・イン・アウト属性に関する１つ以上のノー・イン・アウトフィールド、１つ以上のスレッド・レベル・スペキュレーション（ＴＬＳ）属性に関する１つ以上のＴＬＳフィールド、１つ以上のハズ・サーティフィケート属性に関する１つ以上のハズ・サーティフィケートフィールド、１つ以上のノード属性に関する１つ以上のノードフィールド、１つ以上のアライメント属性に関する１つ以上のアライメントフィールド、１つ以上のオペレーティング・-システム・バージョン属性に関する１つ以上のオペレーティング・システム・バージョンフィールド、１つ以上のイメージ・バージョン属性に関する１つ以上のイメージ・バージョ

ンフィールド、１つ以上のミニマム・サブシステム・バージョン属性に関する１つ以上のミニマム・サブシステム・バージョンフィールド、１つ以上のダイナミック・リンク・ライブラリ（ＤＬＬ）特性属性に関する１つ以上のＤＬＬ特性フィールド、１つ以上のエクスターナルバインディングファシリティ（ＥＢＦｓ）属性に関する１つ以上のＥＢＦｓフィールド、１つ以上のスタック・サイズ属性に関する一つ以上のスタック・サイズフィールド、１つ以上のヒープ・サイズ属性に関する１つ以上のヒープ・サイズフィールド、および／または１つ以上のセクション数属性に関する１つ以上のセクション数フィールドがある。

【００２５】

本明細書に記載する属性は、任意の適切な種類のデータによっても表すことができる。例えば、デバッグ・セクション属性としては、デバッグ・セクション整数があり、このシンボル・テーブル属性としては、シンボル・テーブル・ポインタ整数および／またはシンボル整数の数があり、イメージ・サブシステム属性としては、グラフィカル・ユーザーインタフェース整数および／またはキャラクタ・ベース・ユーザーインタフェース整数があり、またリンカ・バージョン属性としては、メジャー・リンカ・バージョン整数および／またはマイナー・リンカ・バージョン整数がある。ＲＶＡ属性としては、ＲＶＡ・エントリー・ポイント整数、ＲＶＡ・スタート・オブ・コード・セクション整数、ＲＶＡ・ベース・オブ・コード-セクション整数、および／またはＲＶＡ・スタート・オブ・データ・セクション整数がある。

【００２６】

オペレーティング・システム・バージョン属性としては、メジャー・オペレーティング・システム・バージョン整数および／またはマイナー・オペレーティング・システム・バージョン整数がある。イメージ・バージョン属性としては、メジャー・イメージ・バージョン整数および／またはマイナー・イメージ・バージョン整数がある。ミニマム・サブシステム・バージョン属性としては、メジャー・サブシステム・バージョン整数および／またはマイナー・サブシステム・バージョン属性がある。サイズ属性としては、イメージ・サイズ整数、コード・セクション・サイズ整数、初期化・データ・サイズ整数、非初期化・データ・サイズ整数、および／またはヘッダ・サイズ整数がある。オプショナル・ヘッダ属性としては、実行可能ファイルのオプショナル・ヘッダにおける任意の属性がある。

【００２７】

エントリー・ポイント属性としては、エントリー・ポイント整数がある。コード・セクション・ベース属性としては、コード・セクション・ベース整数がある。アライメント属性としてはセクション・アライメント整数および／またはファイル・アライメント整数がある。ＤＬＬ・特性属性としては、ＤＬＬ・特性整数があり、スタック-サイズ属性としては、サイズ・オブ・スタック・リザーブ整数および／またはサイズ・オブ・スタック・コミット整数があり、ヒープ・サイズ属性としては、サイズ・オブ・ヒープ・リザーブ整数および／またはサイズ・オブ・ヒープ・コミット整数があり。ベース・オブ・イメージ属性としては、ベース・オブ・イメージ整数があり、ＥＢＦｓ属性としてはＥＢＦｓ整数がある。ノー・イン・アウト属性としては、ノー・イン・アウト整数があり、ハズ・サーティフィケート属性としては、ハズ・サーティフィケート整数があり、ナンバー・オブ・セクション属性としては、ナンバー・オブ・セクション整数があり、ノード属性としてはノード整数があり、ＴＬＳ属性としてはＴＬＳ整数がある。

【００２８】

インポート属性としては、ハズ・インポート整数、ハズ・ディレイド・インポート整数、ｕｒｌｍｏｎ・インポート整数、ｍｓｖｃｒｔ・インポート整数、ｏｌｅａｕｔ３２・インポート整数、ｓｅｔｕｐａｐｉ・インポート整数、ｕｓｅｒ３２・インポート整数、ａｄｖａｐｉ３２・インポート整数、ｓｈｅｌｌ３２・インポート整数、ｇｄｉ３２・インポート整数、ｃｏｍｄｌｇ３２・インポート整数および／またはｉｍｍ３２・インポート整数がある。特性属性としては、様々な実行可能ファイル特性用の１つ以上の特性フラグがある。

【００２９】
上述のように、メタデータフィールド属性は、実行可能ファイルの様々な特性を表すことができる。例えば、複数のセクション・アライメント整数を使用して、セクションをどこにロードする必要があるかを示すことができる。複数のファイル・アライメント整数は、セクション開始のオフセットとすることができる。メジャーおよびマイナーのオペレーティング・システム・バージョン整数は、実行可能ファイルの実行に必要なミニマム・オペレーティング・システムバージョンを示すことができる。メジャーおよびマイナーのイメージ・バージョン整数は、実行可能ファイルのバージョンを示す。メジャーおよびマイナーのサブシステム・バージョン整数は、実行可能ファイルの実行に必要とされる、最小限のサブシステム・バージョンを示すことができる。イメージ・サイズ整数は、セクション・アライメントを考慮したイメージのサイズを示すことができる。ヘッダ・サイズ整数は、実行可能ファイルにおけるヘッダの全体サイズを示すことができる。グラフィカル・ユーザーインタフェース整数は、実行可能ファイルがグラフィカル・ユーザーインタフェースを使用するか否かを示すフラグを含むことができる。キャラクターベースド・ユーザーインタフェース整数は、実行可能ファイルがキャラクター・ベースド・ユーザーインタフェースを使用するか否かを示すフラグを含むことができる。

【００３０】
サイズ・オブ・スタック・リザーブ整数は、スタックのためにとっておく必要のあるアドレススペースの量を示すことができる。サイズ・オブ・スタック・コミット整数は、スタックのためにコミットされる実際のメモリ量を示すことができる。サイズ・オブ・ヒープ・リザーブ整数は、ヒープのためにとっておく必要のあるアドレススペースの量を示すことができる。サイズ・オブ・ヒープ・コミット整数は、ヒープのためにコミットされる実際のメモリ量を示すことができる。シンボル・テーブル・ポインタ整数は、シンボル・テーブルのオフセットとすることができる。ナンバー・オブ・シンボル整数は、シンボル・テーブル内のシンボル数を示すことができる。デバッグ・セクション整数は、実行可能ファイルがデバッグ・セクションを有するか否かを示すことができる。メジャーおよびマイナーのリンカ・バージョン整数は、実行可能ファイルを生成したリンカのバージョンを示すことができる。コード・セクション・サイズ整数は、実行可能ファイルにおけるコードセクションのサイズを示すことができる。初期化・データ・サイズ整数は、実行可能ファイルの中の初期化・データセクションのサイズを示すことができる。非初期化・データ・サイズ整数は、実行可能ファイルの中の非初期化・データセクションのサイズを示すことができる。

【００３１】
ｕｒｌｍｏｎ・インポート整数は、実行可能ファイルがｕｒｌｍｏｎ．ｄｌｌファイルにリンクしているか否かを示すことができる。ｍｓｖｃｒｔ・インポート整数は、実行可能ファイルがｍｓｖｃｒｔ．ｄｌｌファイルにリンクしているか否かを示すことができる。ｏｌｅａｕｔ３２・インポート整数は、実行可能ファイルがｏｌｅａｕｔ３２．ｄｌｌファイルにリンクしているか否かを示すことができる。 ｓｅｔｕｐａｐｉ・インポート整数は、実行可能ファイルがｓｅｔｕｐａｐｉ．ｄｌｌファイルにリンクしているか否かを示すことができる。ｕｓｅｒ３２・インポート整数は、実行可能ファイルがｕｓｅｒ３２－ｉｍｐｏｒｔ．ｄｌｌ ファイルにリンクしている否かを示すことができる。ａｄｖａｐｉ３２・インポート整数は、実行可能ファイルがａｄｖａｐｉ．ｅｘｅファイルにリンクしているか否かを示すことができる。ｓｈｅｌｌ３２・インポート整数は、実行可能ファイルがｓｈｅｌｌ３２・ｉｍｐｏｒｔｓ．ｄｌｌファイルにリンクしているか否かを示すことができる。ｇｄｉ３２インポート整数は、実行可能ファイルが ｇｄｉ３２．ｄｌｌファイルにリンクしているか否かを示すことができる。ｃｏｍｄｌｇ３２・インポート整数は、実行可能ファイルがｋｏｍｄｌｇ３２．ｄｌｌファイルにリンクしているか否かを示すことができる。ｉｍｍ３２・インポート整数は、実行可能ファイルがｉｍｍ３２．ｄｌｌファイルにリンクしているか否かを示すことができる。

【００３２】

上述したように、実行可能ファイルはＰＥファイルフォーマットとすることができる。図３はＰＥファイルフォーマットの実行可能ファイル３００の例を示す。図３に示すように、実行可能ファイル３００は、ディスクオペレーティングシステム(ＤＯＳ）スタブ３１０を含むことができる。実行可能ファイル３００はまた、ファイルヘッダ３２０を含むことができる。ファイルヘッダ３２０は一つ以上のメタデータフィールドを含むことができる。例えば、ファイルヘッダ３２０は、バイナリが動作するよう意図したシステムを示すマシンフィールド、ナンバー・オブ・セクションフィールド、タイム・スタンプフィールド、シンボル・テーブル・ポインタフィールド、シンボル数フィールド、デバッグ・情報フィールド、オプショナル・ヘッダ・サイズフィールド、イメージ・ファイル・リロケーション・ストリップドフィールド、イメージ・ファイル・実行可能・イメージフィールド、イメージ・ファイル・ライン・ナンバー・ストリップドフィールド、ファイル・ローカル・シンボル・ストリップドフィールド、イメージ・ファイル・アグレッシブ・ウォーキング・セット・トリムフィールド、イメージ・ファイル・バイツ・リバースド・ローフィールド、イメージ・ファイル・３２-ビット・マシンフィールド、イメージ・ファイル・デバッグ・ストリップドフィールド、イメージ・ファイル・リムーバブル・ラン・フロム・スワップフィールド、イメージ・ファイル・ネット・ラン・フロム・スワップフィールド、イメージ・ファイル・システムフィールド、イメージ・ファイルダイナミックリンクライブラリ（ＤＬＬ）フィールド、および／またはイメージ・ファイル・アップ・システム・オンリーフィールドを含むことができる。

【００３３】

実行可能ファイル３００はまた、オプショナル・ヘッダ３３０を含むことができる。オプショナル・ヘッダ３３０は、一つ以上のメタデータフィールドを含む。例えば、オプショナル・ヘッダ３３０は、メジャー・リンカ・バージョンフィールド、マイナー・リンカ・バージョンフィールド、サイズ・オブ・コードフィールド、サイズ・オブ・初期化・データフィールド、サイズ・オブ・非初期化・データフィールド、オフセット・ツー・ザ-エントリー・ポイントフィールド、アドレス・オブ・エントリー・ポイントフィールド、オフセット・ツー・ザ・ベース・オブ・コードフィールド、ベース・オブ・データフィールド、セクション・アライメントフィールド、ファイル・アライメントフィールド、メジャー・オペレーティング・システム・バージョンフィールド、マイナー・オペレーティング・システム・バージョンフィールド、メジャー・イメージ・バージョンフィールド、マイナー・イメージ・バージョンフィールド、メジャー・サブシステム・バージョンフィールド、マイナー・サブシステム・バージョンフィールド、ＷＩＮＤＯＷＳ（登録商標）・３２-ビットグラフィカルユーザーインタフェース（ＧＵＩ）アプリケーションフィールド、ＷＩＮＤＯＷＳ（登録商標）・３２-ビット・バージョン・バリューフィールド、サイズ・オブ・イメージフィールド、サイズ・オブ・ヘッダフィールド、チェック・サムフィールド、イメージ・サブシステム・ネイティブフィールドを含むことができる。

【００３４】

オプショナル・ヘッダ３３０はまた、イメージ・サブシステム・ＷＩＮＤＯＷＳ（登録商標）・ＧＵＩフィールド、イメージ・サブシステム・ＷＩＮＤＯＷＳ（登録商標）キャラクターユーザーインタフェース（ＣＵＩ）フィールド、イメージ・サブシステム・ＯＳ／２・ＣＵＩフィールド、イメージ・サブシステム・ＰＯＳＩＸ・ＣＵＩフィールド、ＤＬＬ・特性フィールド、プロセス・アタッチメントフィールド、スレッド・デタッチメントフィールド、スレッド・アタッチメントフィールド、プロセス・デタッチメントフィールド、サイズ・オブ・スタック・リザーブフィールド、サイズ・オブ・スタック・コミットフィールド、サイズ・オブ・ヒープ・リザーブフィールド、サイズ・オブ・ヒープ・コミットフィールド、ローダフラグ、イメージ-ディレクトリフラグを含むことができる。

【００３５】

実行可能ファイル３００はまた、データディレクトリ３４０およびセクションヘッダ３５０を含むことができる。セクションヘッダ３５０は、１つ以上のメタデータフィールドを含むことができる。例えば、セクションヘッダ３５０は、イメージ・サイズショートニ

ング・フィールド、イメージ・セクション・ヘッダフィールド、バーチャル・アドレスフィールド、サイズ・オブ・ロー・データフィールド、ポインター・ツー・ローデータフィールド、ポインター・ツー・リロケーションフィールド、実行可能ファイル３００の一つ以上の属性を示す一つ以上のフラグを含むことのできる特性フィールドを含む。実行可能なファイル３００はまた、セクション１の３６０（１）からセクションＮの３６０（ｎ）を含むことができる。

【００３６】

図４は、ポータブル実行可能ファイルのマルウェアを検出する方法を示す。検査モジュールは、複数の既知のポータブルな無害実行可能ファイルの複数のメタデータフィールドを検査することができる（ステップ４１０）。検査モジュールはまた、複数の既知のポータブルな悪意実行可能ファイルにおける複数のメタデータフィールドを検査することができる（ステップ４２０）。検査モジュールは、既知のポータブルな無害実行可能ファイルおよび／または既知のポータブルな悪意実行可能ファイルを任意な数だけ検査することができる。例えば、検査モジュールは、数十、数百、数千、数万、数百万の実行可能ファイルを検査することができる。検査モジュールはまた、既知のポータブルな無害実行可能ファイルおよび／または既知のポータブルな悪意実行可能ファイルのメタデータフィールドを任意な数だけ検査することができる。実行可能ファイルが検査された後、推論モジュールは検査に基づいて、マルウェアを示すメタデータフィールド属性を推論することができる（ステップ４３０）。

【００３７】

セキュリティモジュールは、推論モジュールによって推論された情報を使って、未知の実行可能ファイルがマルウェアを含んでいるか否かを決定することができる。例えば、セキュリティモジュールは未知の実行可能ファイルを受け取る可能性がある（ステップ４４０）。セキュリティモジュールは、未知の実行可能ファイルが前もって識別されたマルウェアを示すメタデータフィールド属性を含んでいるか否かを決定することによって、未知の実行可能ファイルがマルウェアを含んでいるか否かを決定することができる。

【００３８】

いくつかの実施形態では、セキュリティモジュールはアンチウィルスセキュリティソフトウェアプログラムを含む、またはその一部とすることができる。少なくとも一つの実施形態によると、クライアントコンピューティング装置はセキュリティモジュールを含むことができ、そしてセキュリティモジュールは、クライアントコンピューティング装置の未知のファイルがマルウェアを含んでいるか否かを決定することによって、クライアントコンピューティング装置を保護することができる。セキュリティモジュールはまた、クライアントコンピューティング装置にダウンロードされるファイルがマルウェアを含んでいるか否かを決定することができる。その他の実施形態では、サーバまたは他の任意のコンピューティング装置もセキュリティモジュールを含むことができる。

【００３９】

一実施形態では、８５０，０００個の既知の無害実行可能ファイルおよび５００，０００個の既知の悪意実行可能ファイルを検査モジュールによって検査した。検査中に収集された情報は、推論モジュールによって処理され、マルウェアを示すヘッダフィールド属性を決定した。推論モジュールによって推論された情報に基づき、セキュリティモジュールは、０．５％以下の誤検出判定を返しながら、約５０～６０％の悪意実行可能ファイルを悪質なものとして識別することができた。このような結果を当業者は予期しなかったであろう。

【００４０】

図５は、本明細書に記載および／または説明する１つ以上の実施形態を実行することのできる例示的なコンピューティングシステム５１０のブロック図である。コンピューティングシステム５１０は、コンピュータの読み取り可能な命令を実行することのできる１つのまたは複数のマルチプロセッサコンピューティング装置またはシステムを広義に表している。コンピューティングシステム５１０の例としては、以下のものに限定しないが、ワ

ークステーション、ラップトップ、クライアント側端末、サーバ、分散コンピューティングシステム、ハンドヘルドデバイスまたは他のコンピューティングシステムまたは装置がある。その最も基本的な構成において、コンピューティングシステム５１０は少なくとも１個のプロセッサ５１４およびシステムメモリ５１６を備えることができる。

【００４１】

プロセッサ５１４は、一般に、データ処理または命令の機械言語翻訳および実行を行うことのできる任意のタイプまたは形式の処理ユニットを表す。ある実施形態では、プロセッサ５１４は、ソフトウェアアプリケーションまたはモジュールから命令を受け取る可能性がある。これらの命令によって、プロセッサ５１４は本明細書に記載および／または説明する一つ以上の実施例の機能を実施することができる。例えば、プロセッサ５１４は、単独で、またはその他の構成部品と組み合わせて、本明細書に記載する検査、推論、受け取り、決定および／または実行というステップのうち一つ以上のものを実施する、または実施する手段とすることができる。プロセッサ５１４は、さらに、本明細書に記載および／または説明する他のステップ、方法またはプロセスを実施する、または実施する手段とすることができる。

【００４２】

システムメモリ５１６は、一般に、データおよび／または他のコンピュータの読み取り可能な命令を記憶することのできる、任意のタイプまたは形式の揮発性もしくは不揮発性の記憶装置または媒体を表す。システムメモリ５１６の例としては、以下のものに限定しないが、ランダムアクセスメモリ（ＲＡＭ）、リードオンリメモリ（ＲＯＭ）、フラッシュメモリまたはその他のあらゆる適切なメモリ装置がある。ある実施形態では、必須ではないが、コンピューティングシステム５１０は、揮発性メモリユニット（例えばシステムメモリ５１６）および不揮発性記憶装置（例えば、以下に詳細を説明する主記憶装置５３２など）の双方を有することができる。

【００４３】

ある実施形態では、例示的なコンピュータシステム５１０は、さらに、プロセッサ５１４およびシステムメモリ５１６に加えて、１つ以上のコンポーネントまたは素子を備えることができる。例えば、図５に説明するように、コンピューティングシステム５１０は、メモリコントローラ５１８、入力／出力（Ｉ／Ｏ）コントローラ５２０および通信インタフェース５２２を備えることができ、それらはそれぞれ通信インフラ５１２を介して相互に接続する。通信インフラ５１２は、一般に、コンピューティング装置における１個以上のコンポーネント間の通信を容易化することのできる、任意のタイプまたは形式のインフラを表す。通信インフラ５１２の例としては、以下のものに限定しないが、通信バス（ＩＳＡ，ＰＣＩ，ＰＣＩｅまたは同様のバス）およびネットワークがある。

【００４４】

メモリコントローラ５１８は、一般に、メモリもしくはデータの処理、またはコンピューティングシステム５１０の１個以上のコンポーネント間の通信を制御することのできる、任意のタイプまたは形式の装置を表す。例えば、ある実施形態では、メモリコントローラ５１８は、通信インフラ５１２を介して、プロセッサ５１４、システムメモリ５１６、Ｉ／Ｏコントローラ５２０間の通信を制御することができる。ある実施形態では、メモリコントローラは、単独で、または他の素子と組み合わせて、例えば検査、推論、受け取り、決定、および／または実行などのような、本明細書に記載および／または説明する、一つ以上のステップまたは機能を実施する、または実施する手段とすることができる。

【００４５】

Ｉ／Ｏコントローラ５２０は、一般に、コンピューティング装置の入出力機能の調整および／または制御を行うことのできる、任意のタイプまたは形式のモジュールを表す。例えば、ある実施形態では、Ｉ／Ｏコントローラは、コンピューティングシステム５１０の１個以上の素子、例えば、プロセッサ５１４、システムメモリ５１６、通信インタフェース５２２、ディスプレイアダプタ５２６、入力インタフェース５３０および記憶インタフェース５３４間におけるデータ送信の制御または容易化を行うことができる。Ｉ／Ｏコン

トローラ５２０は、例えば、単独で、または他の素子と組み合わせて、本明細書に記載する検査、推論、受け取り、決定、および／または実施というステップのうち一つ以上を実施するために使用することができるか、または実施するための手段とすることができる。Ｉ／Ｏコントローラ５２０は、さらに、本明細書に示す他のステップまたは機能を実施するために使うことができる、または実施するための手段とすることができる。

【００４６】

通信インタフェース５２２は、例示的なコンピューティングシステム５１０と１つ以上の付加装置との間の通信を容易化することのできる、任意のタイプまたは形式の通信装置またはアダプタを広義に表す。例えば、ある実施形態では、通信インタフェース５２２は、コンピューティングシステム５１０および付加的コンピューティングシステムを備えるプライベートまたはパブリックのネットワークとの間の通信を容易化することができる。通信インタフェース５２２の例としては、以下のものに限定しないが、有線ネットワークインタフェース（例えばネットワークインタフェースカード）、無線ネットワークインタフェース（例えば無線ネットワークインタフェースカードなど）、モデムおよび他の適切なインタフェースがある。少なくとも一つの実施形態では、通信インタフェース５２２は、例えばインターネットなどのネットワークへの直接のリンクを介して、リモートサーバへの直接的な接続を提供することができる。通信インタフェース５２２はまた、例えば、ローカルエリアネットワーク（Ｅｔｈｅｒｎｅｔ（登録商標）ネットネットワークなど）、パーソナルエリアネットワーク、電話またはケーブルネットワーク、携帯電話接続、衛星データ接続またはその他の適切な接続を通して、このような接続を間接的に提供することもできる。

【００４７】

ある実施形態では、通信インタフェース５２２はまた、外部バスまたは通信チャネルを介して、コンピューティングシステム５１０と１つ以上の追加ネットワークまたは記憶装置との間で通信を容易化するように構成した、ホストアダプタを表すこともできる。ホストアダプタの例には、ＳＣＳＩホストアダプタ、ＵＳＢホストアダプタ、ＩＥＥＥ５９４ホストアダプタ、ＳＡＴＡとｅＳＡＴＡホストアダプタ、ＡＴＡとＰＡＴＡホストアダプタ、ファイバーチャネルインタフェースアダプタ、Ｅｔｈｅｒｎｅｔ（登録商標）アダプタなどが含まれるが、これらに限定されない。また通信インタフェース５２２によって、コンピューティングシステム５１０に分散またはリモートコンピューティングを行わせることもできる。例えば、通信インタフェース５２２はリモート装置から命令を受け取ったり、または実行のためにリモート装置へ命令を送ったりすることができる。ある実施形態では、通信インタフェース５２２は、単独で、またはその他の構成要素と組み合わせて、本明細書に開示する検査、推論、受け取り、決定および／または実施というステップのうち一つ以上のものを実施する、または実施する手段とすることができる。通信インタフェース５２２は、さらに、本明細書に示す他のステップおよび特徴を実施するために使用することができる、または実施する手段とすることができる。

【００４８】

図５に示すように、コンピューティングシステム５１０は、さらに、ディスプレイアダプタ５２６を介して通信インフラ５１２に接続する少なくとも１つのディスプレイ装置５２４を備えることができる。ディスプレイ装置５２４は一般に、ディスプレイアダプタ５２６によって転送される情報を視覚的に表示することのできる、任意のタイプまたは形式の装置を表す。同様に、ディスプレイアダプタ５２６は、一般に、ディスプレイ装置５２４で表示するように、通信インフラ５１２（または本発明の技術分野で既知のフレームバッファ）からのグラフィックス、テキストおよび他のデータを転送するよう構成した、任意のタイプまたは形式の装置を表す。

【００４９】

図５に示すように、例示的なコンピューティングシステム５１０は、さらに、入力インタフェース５３０を介して通信インフラ５１２に接続する、少なくとも１個の入力装置５２８を備えることができる。入力装置５２８は、一般に、コンピュータまたは人間が生成

した入力を例示的コンピューティングシステム５１０に供給することのできる、任意のタイプまたは形式の入力装置を表す。入力装置５２８の例としては、以下のものに限定しないが、キーボード、ポインティング装置、音声認識装置または他の入力装置がある。少なくとも一つの実施形態では、入力装置５２８は、単独で、または他の素子と組み合わせて、本明細書に開示する検査、推論、受け取り、決定、および／または実施というステップのうち一つ以上のものを実施することができるか、または実施するための手段とすることができる。入力装置５２８は、さらに、本明細書に示すその他のステップおよび特徴を実施するために使用することができるか、または実施するための手段とすることができる。

【００５０】

図５に説明するように、例示的なコンピューティングシステム５１０は、さらに、記憶インタフェース５３４を介して通信インフラ５１２に接続する、主記憶装置５３２およびバックアップ記憶装置５３３を備えることができる。記憶装置５３２および５３３は、一般に、データおよび／または他のコンピュータ読み取り可能な命令を記憶することのできる、任意のタイプおよび／または形式の記憶装置または記憶媒体を表す。例えば、記憶装置５３２，５３３は、磁気ディスクドライブ（例えば、いわゆるハードドライブ）、フロッピー（登録商標）ディスクドライブ、磁気テープドライブ、光ディスクドライブ、フラッシュドライブとすることができる。記憶インタフェース４２３は、一般に、記憶装置５３２，５３３と、コンピューティングシステム５１０における他の素子との間でデータの送信を行う、任意のタイプまたは形式のインタフェースまたはデバイスを表す。

【００５１】

ある実施形態では、記憶装置５３２，５３３は、コンピュータソフトウェア、データ、または他のコンピュータ読み取り可能な情報を記憶するように構成した、リムーバブル記憶装置に対する読み出しおよび／または書き込みを行うよう構成することができる。適切なリムーバブル記憶ユニットの例としては、以下のものに限定しないが、フロッピー（登録商標）ディスク、磁気テープ、光ディスク、フラッシュメモリ装置がある。記憶装置５３２，５３３はまた、コンピュータソフトウェア、データまたは他のコンピュータの読み取り可能な命令をコンピューティングシステム５１０にロードするための他の類似な構造またはデバイスを備えることができる。例えば、記憶装置５３２，５３３は、ソフトウェア、データまたは他のコンピュータの読み取り可能な情報を読み出しおよび書き込みするよう構成することができる。記憶装置５３２，５３３は、さらに、コンピューティングシステム５１０の一部とする、または他のインタフェースシステムを通してアクセスする個別のデバイスとすることもできる。

【００５２】

ある実施形態では、本明細書に開示する例示的なファイルシステムを主記憶装置５３２に記憶するとともに、本明細書に開示する例示的なファイルシステムバックアップをバックアップ記憶装置５３３に記憶することができる。記憶装置５３２，５３３は、例えば、本開示に示す他のステップや特徴を実施するために使用することができる、または実施するための手段とすることができる。

【００５３】

他の多数のデバイスまたはサブシステムをコンピューティングシステム５１０に接続することができる。逆に、本明細書に記載および／または説明する実施形態を実施するために、図５に示すコンポーネントおよびデバイスの全てが存在しなければならないということではない。上述したデバイスおよびサブシステムは、さらに、図５に示すものとは異なる方法で相互に接続することができる。コンピューティングシステム５１０は、さらに、ソフトウェア、ファームウェア、および／またはハードウェア構成を任意の数だけ使用することができる。例えば、本明細書に開示する一つ以上の実施例は、コンピュータ読み取り可能な媒体にコンピュータプログラム（または、コンピュータソフトウェア、ソフトウェアアプリケーション、コンピュータの読み取り可能な命令またはコンピュータ制御ロジックとも称することができる）として符号化することができる。用語「コンピュータ読み取り可能な媒体」とは、一般に、コンピュータの読み取り可能な命令を記憶または担持す

ることができる任意の形式の装置、キャリア、または媒体に言及する。コンピュータ読み取り可能な媒体としては、以下のものに限定しないが、例えば、搬送波のような伝送タイプの媒体、および磁気記憶媒体のような物理的媒体(例えば、ハードディスクドライブやフロッピー（登録商標）ディスク)、光記憶媒体(例えば、ＣＤ－ＲＯＭまたはＤＶＤ－ＲＯＭ)、電子記憶媒体(例えばソリッドステートドライブおよびフラッシュ媒体)および他の分散システムがある。

【００５４】

コンピュータプログラムを含むコンピュータ読み取り可能な媒体は、コンピューティングシステム５１０にロードすることができる。コンピュータ読み取り可能な媒体に記憶されるコンピュータプログラムの全体または一部は、それからシステムメモリ５１６および／または記憶装置５３２，５３３の様々な部分に記憶される。プロセッサ５１４によって実行されると、コンピューティングシステム５１０にロードされたコンピュータプログラムは、プロセッサ５１４に、本明細書に記載および／または説明する一つ以上の実施例の機能を実施させる、または実施させるための手段となるようにすることができる。付加的または代替的に、本明細書に記載および／または説明する一つ以上の実施例は、ファームウェアおよび／またはハードウェアで実現することができる。例えば、コンピューティングシステム５１０は、本明細書に開示する一つ以上の実施例を実現するよう構成した、特定用途向け集積回路（ＡＳＩＣ）として構成することができる。

【００５５】

図６は、例示的なネットワークアーキテクチャ６００のブロック図であり、クライアントシステム６１０，６２０，６３０およびサーバ６４０，６４５をネットワーク６５０に接続することができる。クライアントシステム６１０，６２０，６３０は、一般に、例えば図５の例示的なコンピューティングシステム５１０のような、任意のタイプまたは形式のコンピューティング装置またはシステムを表す。同様に、サーバ６４０，６４５は、一般に、様々なデータベースサービスの提供および／または特定のソフトウェアアプリケーションを実行するよう構成した、アプリケーションサーバまたはデータベースサーバのような、コンピューティング装置またはシステムを表す。ネットワーク６５０は、一般に、例えば、イントラネット、広域ネットワーク（ＷＡＮ）、ローカルエリアネットワーク（ＬＡＮ）、パーソナルエリアネットワーク（ＰＡＮ）またはインターネットを含む、あらゆる電気通信またはコンピュータネットワークを表す。

【００５６】

図６に示すように、１台以上の記憶装置６６０（１）～（Ｎ）をサーバ６４０に直接取り付けることができる。同様に、１台以上の記憶装置６７０（１）～（Ｎ）をサーバ６４５に直接取り付けることができる。記憶装置６６０（１）～（Ｎ）および記憶装置６７０（１）～（Ｎ）は一般に、データおよび／または他のコンピュータ読み取り可能な命令を記憶することのできる、任意のタイプもしくは形式の記憶装置または媒体を表している。ある実施形態では、記憶装置６６０（１）～（Ｎ）および記憶装置６７０（１）～（Ｎ）は、ＮＦＳ，ＳＭＢまたはＣＩＦＳのような様々なプロトコルを使って、サーバ６４０および６４５と通信を行うように構成したネットワーク接続ストレージ（ＮＡＳ）装置を表すことができる。

【００５７】

サーバ６４０，６４５は、さらに、ストレージエリアネットワーク（ＳＡＮ）ファブリック６８０に接続することができる。ＳＡＮファブリック６８０は一般に、複数の記憶装置間の通信を容易化することのできる、任意のタイプもしくは形式のコンピュータネットワークまたはアーキテクチャを表す。ＳＡＮファブリック６８０は、サーバ６４０，６４５、複数の記憶装置６９０（１）～（Ｎ）、および／またはインテリジェントストレージアレイ６９５間の通信を容易化することができる。ＳＡＮファブリック６８０は、さらに、ネットワーク６５０およびサーバ６４０，６４５を介して、クライアントシステム６１０，６２０，６３０と記憶装置６９０（１）～（Ｎ）および／またはインテリジェントストレージアレイ６９５との間の通信を、記憶装置６９０（１）～（Ｎ）およびアレイ６９

５がクライアントシステム６１０，６２０，６３０に局部的に取り付けられた装置として現れるようにして、容易化する。記憶装置６６０（１）～（Ｎ）および記憶装置６７０（１）～（Ｎ）のように、記憶装置６９０（１）～（Ｎ）およびインテリジェントストレージアレイ６９５は、一般に、データおよび／または他のコンピュータ読み取り可能な命令を記憶することのできる、任意のタイプもしくは形式の記憶装置または媒体を表す。

【００５８】

ある実施形態では、図５の例示的なコンピューティングシステム５１０につき説明すると、図５の通信インタフェース５２２のような通信インタフェースは、クライアントシステム６１０，６２０，６３０のそれぞれとネットワーク６５０との間に接続能力を付与するために使用することができる。クライアントシステム６１０，６２０，６３０は、例えば、ウェブブラウザまたはその他のクライアントソフトウェアを使って、サーバ６４０または６４５の情報にアクセスすることができる。このようなソフトウェアによって、クライアントシステム６１０，６２０，６３０は、サーバ６４０，サーバ６４５、記憶装置６６０（１）～（Ｎ）、記憶装置６７０（１）～（Ｎ）、記憶装置６９０（１）～（Ｎ）および／またはインテリジェントストレージアレイ６９５によってホストされるデータにアクセスすることができる。図６はデータ交換のネットワーク（例えばインターネットのような）の使用を説明するものであるが、本明細書に記載および／または説明する実施形態は、インターネットまたは特定のネットワークベースの環境に限定されるものではない。

【００５９】

少なくとも一つの実施形態では、本明細書に開示する一つ以上の実施例の全てまたは一部を、コンピュータプログラムとして符号化し、サーバ６４０、サーバ６４５、記憶装置６６０（１）～（Ｎ）、記憶装置６７０（１）～（Ｎ）、記憶装置６９０（１）～（Ｎ）および／またはインテリジェントストレージアレイ６９５またはそれらの任意の組み合わせにロードし、そして実行することができる。本明細書に開示する一つ以上の実施例の全てまたは一部もまたコンピュータプログラムとして符号化し、サーバ６４０に記憶し、サーバ６４５によって実行し、ネットワーク６５０によってクライアントシステム６１０，６２０，６３０に分散することができる。よって、ネットワークアーキテクチャ６００は、単独で、または他の素子と組み合わせて、本明細書に開示する検査、推論、受け取り、決定および／または実行というステップのうち１つ以上を行うことができる、および／または行う手段とすることができる。ネットワークアーキテクチャ６００は、さらに、本開示に示すその他のステップや機能の特徴を実施するために使用される、または実施するための手段とすることができる。

【００６０】

詳細に上述したように、コンピューティングシステム４１０および／またはネットワークアーキテクチャ５００の一つ以上のコンポーネントは、単独または他の素子との組み合わせで、本明細書に記載および／または説明する例示的な方法の一つ以上のステップを実施することができる、または実施するための手段とすることができる。例えば、コンピューティングシステムは、複数の既知の無害実行可能ファイルにおける複数のメタデータフィールドを検査することができる。コンピューティングシステムは、さらに、複数の既知の悪意実行可能ファイルにおける複数のメタデータフィールドも検査することができる。コンピューティングシステムは、複数の既知の無害な、および既知の悪意実行可能ファイルにおける複数のメタデータフィールドを検査することから得た情報に基づいて、マルウェアを示すメタデータフィールド属性を推論することができる。

【００６１】

ある実施形態では、コンピューティングシステムは悪意実行可能ファイルを受け取る可能性がある。コンピューティングシステムは、未知の実行可能ファイルがマルウェアを示すメタデータフィールド属性を含んでいるか否かを決定することにより、未知の実行可能ファイルがマルウェアを含んでいるか否かを決定することができる。少なくとも一つの実施形態では、コンピューティングシステムは、未知の実行可能ファイルがマルウェアを含んでいると、セキュリティアクションを実施することができる。

【００６２】
様々な実施形態によると、複数の既知の無害実行可能ファイルは、ポータブルの実行可能ファイルを含むことができ、複数の既知の悪意実行可能ファイルはポータブルの実行可能ファイルを含むことができる。少なくとも一つの実施形態では、メタデータフィールド属性はヘッダフィールド属性を含むことができる。ある実施形態によると、ヘッダフィールド属性は、デバッグ・セクション属性、インポート属性、シンボル・テーブル属性、オプショナル・ヘッダ属性、特性属性、イメージ・サブシステム属性、リンカ・バージョン属性、サイズ属性、リアル・バーチャル・アドレス属性、エントリー・ポイント属性、コード・セクション・ベース属性、アライメント属性、オペレーティング・システム・バージョン属性、イメージ・バージョン属性、ミニマム・サブシステム・バージョン属性、ダイナミック・リンク・ライブラリキャラクテリスティック属性、スタック・サイズ属性、ヒープ・サイズ属性、セクション数属性、ノー・イン・アウト属性、スレッド・レベル・スペキュレーション属性および／またはベース・オブ・イメージ属性のうち少なくとも一つを含む。
【００６３】
いくつかの実施形態では、マルウェアを示すメタデータフィールド属性は、複数の既知の無害な実行可能ファイルと既知の悪意実行可能ファイル内で検査される複数のメタデータフィールドのサブセットを含む。少なくとも一つの実施形態では、マルウェアを示すメタデータフィールド属性は、静的属性を含むことができる。様々な実施形態では、マルウェアを示すメタデータフィールド属性の推論は、マルウェアを示すメタデータフィールド属性の少なくとも一つの組み合わせの決定を含む。少なくとも一つの実施形態によると、マルウェアを示すメタデータフィールド属性の推論には、マルウェアを示す第１メタデータフィールド属性が、マルウェアを示す第２メタデータフィールド属性よりも強いマルウェアの指標であるという決定を含む。
【００６４】
いくつかの実施形態では、マルウェアを示すメタデータフィールド属性は、デバッグ・セクション属性、シンボル・テーブル属性、イメージ・サブシステム属性、リンカ・バージョン属性、リアル・バーチャル・アドレス属性、オペレーティング・システム・バージョン属性、イメージ・バージョン属性、ミニマム・サブシステム・バージョン属性、ダイナミック・リンク・ライブラリ・キャラクテリスティック属性、スタック・サイズ属性、ヒープ・サイズ属性、および／またはセクション数属性のうち少なくとも一つを含む。
【００６５】
いくつかの実施形態では、マルウェアを示すメタデータフィールド属性は、セクション・アライメント整数、ファイル・アライメント整数、メジャー・オペレーティング・システム・バージョン整数、マイナー・オペレーティング・システム・バージョン整数、メジャー・イメージ・バージョン整数、マイナー・イメージ・バージョン整数、メジャー・サブシステム・バージョン整数、マイナー・サブシステム・バージョン整数、イメージ・サイズ整数、ヘッダ・サイズ整数、グラフィカル・ユーザーインタフェース整数、キャラクタ・ベース・ユーザーインタフェース整数、サイズ・オブ・スタック・リザーブ整数、サイズ・オブ・スタック・コミット整数、サイズ・オブ・ヒープ・リザーブ 整数、サイズ・オブ・ヒープ・コミット整数、シンボル・テーブル・ポインタ整数、シンボル数整数、デバッグ・セクション整数、メジャー・リンカ・バージョン整数、マイナー・リンカ・バージョン整数、コード・セクション・サイズ整数、初期化・データ・サイズ整数、非初期化・データ・サイズ整数、リアル・バーチャル・アドレス（「ＲＶＡ」）エントリー・ポイント整数、ＲＶＡ・スタート・オブ・コード・セクション整数、ＲＶＡ・ベース・オブ・コード・セクション整数、ＲＶＡ・スタート・オブ・データ・セクション整数、ベース・オブ・イメージ整数、ハズ・インポート整数、ハズ・ディレイ・インポート整数、エクスターナル・バインディング・ファシリティ整数、ノー・イン・アウト整数、ｕｒｌｍｏｎ・インポート整数、スレッド・レベル・スペキュレーション整数、ｍｓｖｃｒｔ・インポート整数、ｏｌｅａｕｔ３２・インポート整数、ｓｅｔｕｐａｐｉ・インポート整数、

ｕｓｅｒ３２・インポート整数、ａｄｖａｐｉ３２・インポート整数、ｓｈｅｌｌ３２・インポート整数、ｇｄｉ３２・インポート整数、ｃｏｍｄｌｇ３２・インポート整数、ｉｍｍ３２・インポート整数、 ハズ・サーティフィケート整数、ノード整数および／またはセクション数整数のうち少なくとも２つを含む。

【００６６】

ある実施形態によると、システムは、複数の既知の無害実行可能ファイルにおける複数のヘッダフィールド、および複数の既知の悪意実行可能ファイルにおける複数のヘッダフィールドを検査するようにプログラムした検査モジュールを有する。このシステムは、さらに、検査モジュールによって得た情報を記憶するように構成したデータベースと、検査モジュールから取得した情報に基づいて、マルウェアを示すヘッダフィールド属性を推論するようにプログラムした推論モジュールとを有する。

【００６７】

ある実施形態では、システムは未知の実行可能ファイルを受け取る、および／またはその未知の実行可能ファイルがマルウェアを示すヘッダフィールド属性を含んでいるか否かを決定することによって、その未知の実行可能ファイルがマルウェアを含んでいるか否かを決定するようにプログラムしたセキュリティシステムを有することができる。少なくとも一つの実施形態では、セキュリティモジュールは、さらに、未知の実行可能ファイルがマルウェアを含んでいる場合、セキュリティアクションを実施するようにプログラムすることができる。

【００６８】

上述の開示は特定のブロック図、フローチャートおよび例を使って様々な実施形態を示すが、本明細書に記載および／または説明する各ブロック図の構成部分、フローチャートステップ、動作および／またはコンポーネントは、広範囲のハードウェア、ソフトウェアまたはファームウェア（またはそれらの組み合わせ）を使って個々におよび／または集合的に実現することができる。さらに、他のコンポーネントに含まれる構成部品の開示はどのようなものであっても、同じ機能を達成するために多くの他のアーキテクチャを実現することができるので、本来は例示的なものと見なすことができる。

【００６９】

本明細書に記載および／または説明するプロセスパラメータおよびステップの順序は、単なる例であり、所望に応じて修正することができる。例えば、本明細書に説明または記載のステップは特定の順序で示す、または論じたが、これらのステップは必ずしも説明または論じた通りの順序で行う必要はない。本明細書に記載または説明する様々な例示的方法もまた、本明細書に記載または説明する１つ以上のステップを削除することもできるし、または開示されたものに加えて、追加のステップを含むこともできる。

【００７０】

さらに、様々な実施形態を完全に機能的なコンピューティングシステムとして本明細書に記載および／または説明してきたが、分散を行うために実際に使用される特定の種類のコンピュータ読み取り可能な媒体に関わらず、一つ以上のこれらの実施例は、様々な形態のプログラム製品として流通させることができる。本明細書に開示する実施形態はまた、特定のタスクを実行するソフトウェアモジュールを使って実現することもできる。これらのソフトウェアモジュールは、コンピュータ読み取り可能な記憶媒体またはコンピューティングシステムに記憶することのできるスクリプト、バッチまたはその他の実行可能ファイルを含むことができる。いくつかの実施形態では、これらのソフトウェアモジュールは、本明細書に開示する一つ以上の実施例を実行するコンピューティングシステムとして構成することができる。

【００７１】

上述の説明は、当業者が本明細書に記載する実施例の様々な態様を最大限に利用できるように行った。この例示的説明は網羅的、または開示した通りの形式に限定することを意図したものではない。本発明の要旨または範囲から逸脱することなく、多くの変更および改変が可能である。本明細書に記載する実施形態は、あらゆる面において例示的であり、

限定するものではないと見なし、また、添付の特許請求の範囲および特許請求の範囲が決定する等価物を参照して本発明の範囲を決定するのが望ましい。

【００７２】

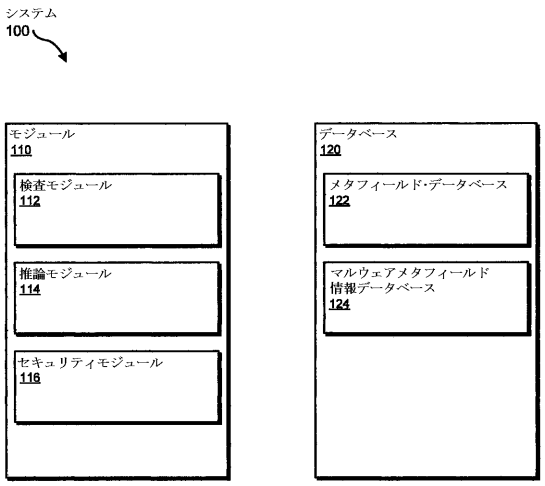
別段に記載のない限り、明細書または特許請求の範囲において、各要素は複数存在し得る。さらに、簡単のために、明細書および請求項で使用される「～を含む」および「～を有する」という言葉は、同じ意味を持つ言葉である「～を備える」と置き替えることができる。

【符号の説明】

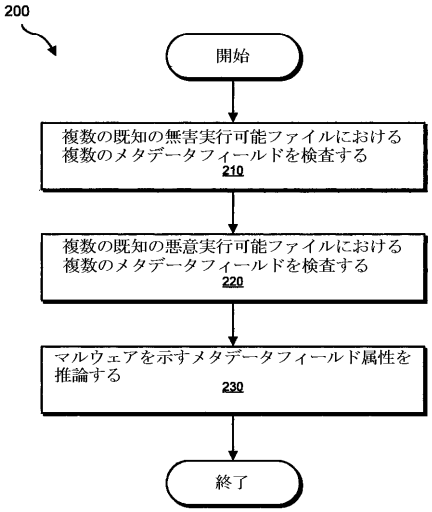
【００７３】

１００　システム
１１０　モジュール
１１２　検査モジュール
１１４　推論モジュール
１１６　セキュリティモジュール
１２０　データベース
１２２　メタデータフィールド・データベース
１２４　マルウェアメタデータフィールド情報データベース
３００　実行可能ファイル
３１０　ＤＯＳスタブ
３２０　ファイルヘッダ
３３０　オプショナル・ヘッダ
３４０　データディレクトリ
３５０　セクションヘッダ
３６０（１）　セクション１
３６０（２）　セクション２
３６０（ｎ）　セクションＮ
５１０　コンピューティングシステム
５１４　プロセッサ
５１６　システムメモリ
５１８　メモリコントローラ
５２０　Ｉ／Ｏコントローラ
５２２　通信インタフェース
５２４　ディスプレイ装置
５２６　ディスプレイアダプタ
５２８　入力装置
５３０　入力インタフェース
５３２　主記憶装置
５３３　バックアップ記憶装置
５３４　記憶インタフェース
６００　ネットワークアーキテクチャ
６１０、６２０、６３０　クライアント
６４０、６４５　サーバ
６５０　ネットワーク
６６０（１）～６６０（Ｎ）、６７０（１）～６７０（Ｎ）、６９０（１）～６９０（Ｎ）装置
６８０　ＳＡＮファブリック
６９５　インテリジェント記憶アレイ

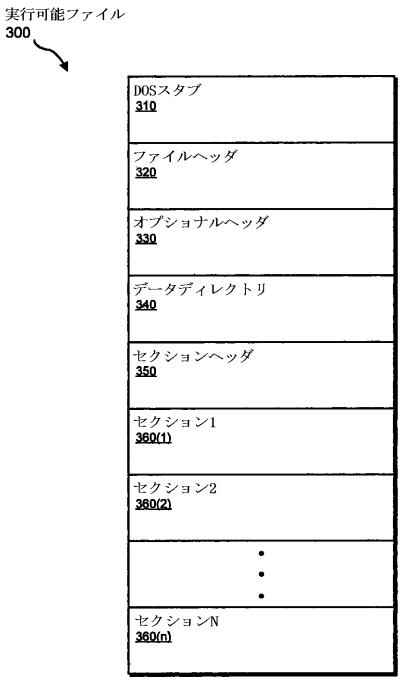
【図１】


システム
100
モジュール
110
検査モジュール
112
推論モジュール
114
セキュリティモジュール
116
データベース
120
メタフィールド･データベース
122
マルウェアメタフィールド
情報データベース
124

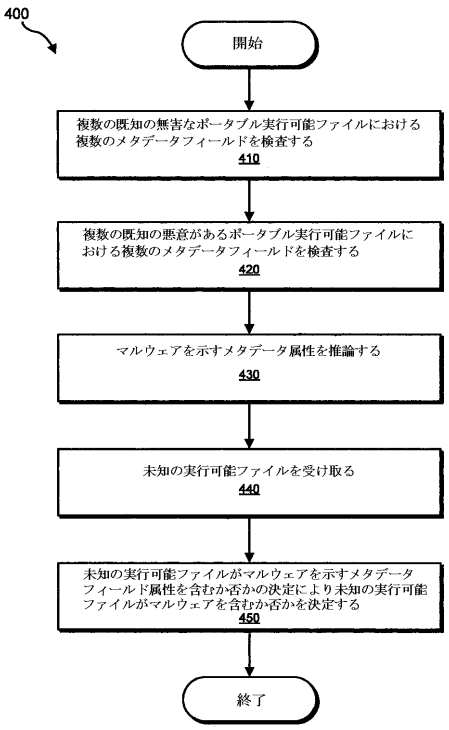
【図２】


200
開始
複数の既知の無害実行可能ファイルにおける
複数のメタデータフィールドを検査する
210
複数の既知の悪意実行可能ファイルにおける
複数のメタデータフィールドを検査する
220
マルウェアを示すメタデータフィールド属性を
推論する
230
終了

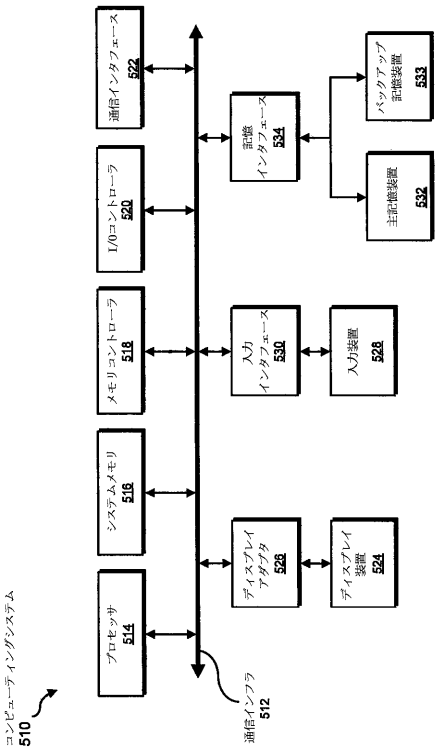
【図３】


実行可能ファイル
300
DOSスタブ
310
ファイルヘッダ
320
オプショナルヘッダ
330
データディレクトリ
340
セクションヘッダ
350
セクション1
360(1)
セクション2
360(2)
セクションN
360(n)

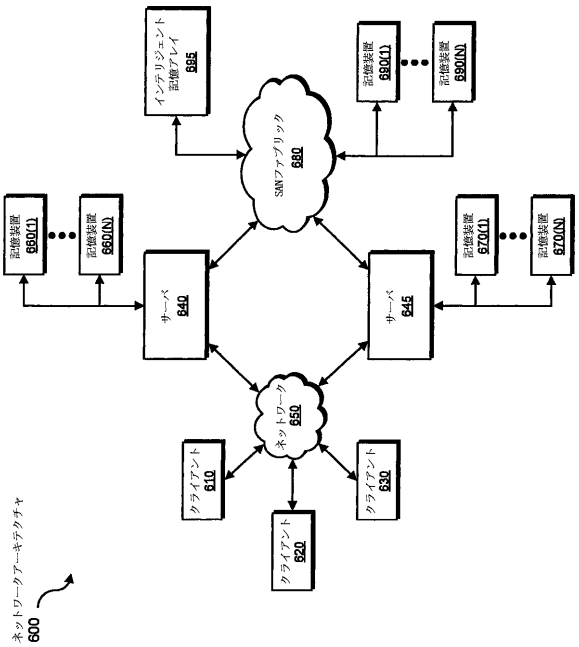
【図４】


400
開始
複数の既知の無害なポータブル実行可能ファイルにおける
複数のメタデータフィールドを検査する
410
複数の既知の悪意があるポータブル実行可能ファイルに
おける複数のメタデータフィールドを検査する
420
マルウェアを示すメタデータ属性を推論する
430
未知の実行可能ファイルを受け取る
440
未知の実行可能ファイルがマルウェアを示すメタデータ
フィールド属性を含むか否かの決定により未知の実行可能
ファイルがマルウェアを含むか否かを決定する
450
終了

【図５】


コンピューティングシステム
510
プロセッサ
514
システムメモリ
516
メモリコントローラ
518
I/Oコントローラ
520
通信インタフェース
522
ディスプレイアダプタ
526
ディスプレイ装置
524
入力インタフェース
530
入力装置
528
記憶インタフェース
534
主記憶装置
532
バックアップ記憶装置
533
通信インフラ
512

【図６】


ネットワークアーキテクチャ
600
サーバ
640
サーバ
645
ネットワーク
650
クライアント
610
クライアント
620
クライアント
630
SANファブリック
680
インテリジェント記憶アレイ
685
記憶装置
690(1)
記憶装置
690(N)
記憶装置
660(1)
記憶装置
660(N)
記憶装置
670(1)
記憶装置
670(N)

【外国語明細書】
2010146566000001.pdf