(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6526842号
(P6526842)
(45)発行日　令和1年6月5日(2019.6.5)　(24)登録日　令和1年5月17日(2019.5.17)

(51)Int.Cl.　F I
G06F 21/56 (2013.01)　G06F 21/56 360

請求項の数 20 (全 19 頁)

(21)出願番号　特願2017-566815(P2017-566815)
(86)(22)出願日　平成28年5月25日(2016.5.25)
(65)公表番号　特表2018-519604(P2018-519604A)
(43)公表日　平成30年7月19日(2018.7.19)
(86)国際出願番号　PCT/US2016/033977
(87)国際公開番号　WO2017/003587
(87)国際公開日　平成29年1月5日(2017.1.5)
審査請求日　平成30年2月5日(2018.2.5)
(31)優先権主張番号　14/752,901
(32)優先日　平成27年6月27日(2015.6.27)
(33)優先権主張国　米国(US)

(73)特許権者　517378810
マカフィー，エルエルシー
アメリカ合衆国、９５０５４　カリフォルニア州、サンタ　クララ　ミッション　カレッジ　ブーレバード　２８２１
(74)代理人　110000877
龍華国際特許業務法人
(72)発明者　エドワーズ、ジョナサン　エル.
アメリカ合衆国、９５０５４　カリフォルニア州、サンタ　クララ　ミッション　カレッジ　ブーレバード　２８２１　マカフィー、インコーポレイテッド内

最終頁に続く

(54)【発明の名称】マルウェアの検出

(57)【特許請求の範囲】
【請求項１】
少なくとも１つのプロセッサに、
ハイパーバイザを用いて、実行中のプロセスを監視する手順と、
前記プロセスが１または複数のシステム関数を検索するために解析しているかどうかを判断する手順と、
前記プロセスが１または複数のシステム関数を検索するために解析しているという判断に基づいて、マルウェアについて前記プロセスを分析するべく前記プロセスをネットワーク要素に送信する手順と
を実行させ、
前記判断する手順は、前記プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合に、前記プロセスが１または複数のシステム関数を検索するために解析していると判断することを含む、
コンピュータプログラム。
【請求項２】
前記プロセスはシェルコードを含む、請求項１に記載のコンピュータプログラム。
【請求項３】
前記少なくとも１つのプロセッサに、前記プロセスがブラックリストに見つかることを判断する手順をさらに実行させる、請求項１または２に記載のコンピュータプログラム。

【請求項４】
前記少なくとも１つのプロセッサに、前記プロセスがブラックリストに見つかるという判断に基づいて、前記プロセスが動作できないようにする手順をさらに実行させる、請求項３に記載のコンピュータプログラム。
【請求項５】
前記少なくとも１つのプロセッサに、前記プロセスが１または複数のシステム関数を検索するために解析していないという判断に基づいて、前記プロセスが動作することを許可する手順をさらに実行させる、請求項１から４のいずれか一項に記載のコンピュータプログラム。
【請求項６】
装置であって、
ネットワーク要素との通信のための回路と、
電子コードを格納することができるメモリ要素と、
ハイパーバイザと、
前記装置が、
前記ハイパーバイザを用いて、実行中のプロセスを監視し、
前記プロセスが１または複数のシステム関数を検索するために解析しているかどうかを判断し、
前記プロセスが１または複数のシステム関数を検索するために解析しているという判断に基づいて、マルウェアについて前記プロセスを分析するべく前記プロセスをネットワーク要素に送信する
ように構成されるように、前記電子コードに関連付けられる命令を実行することができるプロセッサと
を備え、
前記判断することは、前記プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合に、前記プロセスが１または複数のシステム関数を検索するために解析していると判断することを含む、
装置。
【請求項７】
前記プロセスはシェルコードを含む、請求項６に記載の装置。
【請求項８】
前記プロセッサは、前記装置が、前記プロセスがブラックリストに見つかることを判断するようにさらに構成されるように、前記電子コードに関連付けられるさらなる命令を実行することができる、請求項６または７に記載の装置。
【請求項９】
前記プロセッサは、前記装置が、前記プロセスがブラックリストに見つかるという判断に基づいて、前記プロセスが動作できないようにするようにさらに構成されるように、前記電子コードに関連付けられるさらなる命令を実行することができる、請求項８に記載の装置。
【請求項１０】
前記プロセッサは、前記装置が、前記プロセスが１または複数のシステム関数を検索するために解析していないという判断に基づいて、前記プロセスが動作することを許可するようにさらに構成されるように、前記電子コードに関連付けられるさらなる命令を実行することができる、請求項６から９のいずれか一項に記載の装置。
【請求項１１】
ハイパーバイザを用いて、実行中のプロセスを監視する段階と、
前記プロセスが１または複数のシステム関数を検索するために解析しているかどうかを判断する段階と、
前記プロセスが１または複数のシステム関数を検索するために解析しているという判断

に基づいて、マルウェアについて前記プロセスを分析するべく前記プロセスをネットワーク要素に送信する段階と
を備え、
前記判断する段階は、前記プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合に、前記プロセスが１または複数のシステム関数を検索するために解析していると判断することを含む、
方法。
【請求項１２】
前記プロセスはシェルコードを含む、請求項１１に記載の方法。
【請求項１３】
前記プロセスがブラックリストに見つかることを判断する段階をさらに備える、請求項１１または１２に記載の方法。
【請求項１４】
前記プロセスがブラックリストに見つかるという判断に基づいて、前記プロセスが動作できないようにする段階をさらに備える、請求項１３に記載の方法。
【請求項１５】
前記プロセスが１または複数のシステム関数を検索するために解析していないという判断に基づいて、前記プロセスが動作することを許可する段階をさらに備える、請求項１１から１４のいずれか一項に記載の方法。
【請求項１６】
マルウェアを検出するためのシステムであって、
前記システムは、
ネットワーク要素と、
ハイパーバイザと、
メモリ要素と、
プロセッサであって、
前記ハイパーバイザを用いて、実行中のプロセスを監視し、
前記プロセスが１または複数のシステム関数を検索するために解析しているかどうかを判断し、
前記プロセスが１または複数のシステム関数を検索するために解析しているという判断に基づいて、マルウェアについて前記プロセスを分析するべく前記プロセスをネットワーク要素に送信する
命令を実行することができるプロセッサと
を有する電子デバイスと、
前記ネットワーク要素と前記電子デバイスとを接続するネットワークと
を備え、
前記判断することは、前記プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合に、前記プロセスが１または複数のシステム関数を検索するために解析していると判断することを含む、
システム。
【請求項１７】
前記プロセスはシェルコードを含む、請求項１６に記載のシステム。
【請求項１８】
前記プロセッサは、前記プロセスがブラックリストに見つかることを判断するさらなる命令を実行することができる、請求項１６または１７に記載のシステム。
【請求項１９】
前記プロセッサは、前記プロセスがブラックリストに見つかるという判断に基づいて、前記プロセスが動作できないようにするさらなる命令を実行することができる、請求項１

８に記載のシステム。
【請求項２０】
前記プロセッサは、前記プロセスが１または複数のシステム関数を検索するために解析していないという判断に基づいて、前記プロセスが動作することを許可するさらなる命令を実行することができる、請求項１６から１９のいずれか一項に記載のシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
［関連出願の相互参照］
本願は、２０１５年６月２７日に出願された「マルウェアの検出」と題する、米国非仮（実用）特許出願第１４／７５２，９０１号の利益およびそれに基づく優先権を主張し、その全体が本明細書に参照として組み込まれる。
【０００２】
本開示は、概して、情報セキュリティの分野に関し、より具体的には、マルウェアの検出に関する。
【背景技術】
【０００３】
ネットワークセキュリティの分野は、現代社会において、ますます重要になってきている。インターネットは、世界中の種々のコンピュータネットワークの相互接続を可能にしている。特に、インターネットは、様々な種類のクライアントデバイスを介して、種々のコンピュータネットワークに接続された種々のユーザ間でデータを交換するための媒体を提供する。インターネットの使用が、ビジネスコミュニケーションおよびパーソナルコミュニケーションを変化させてきた一方、それはまた、悪意のあるオペレータが、コンピュータおよびコンピュータネットワークへ不正アクセスを得るための、および機密情報の意図的なまたは不注意な開示のための手段として使われている。
【０００４】
ホストコンピュータを感染させる悪意のあるソフトウェア（「マルウェア」）は、ホストコンピュータと関連付けられる企業または個人から機密情報を盗取すること、他のホストコンピュータに伝播すること、および／または分散サービス妨害攻撃に支援すること、ホストコンピュータからスパムまたは悪意のある電子メールを送信することなどの任意の数の悪意のある動作を実行することが可能であり得る。従って、悪意のあるソフトウェアおよびデバイスによる、悪意があって、不注意な不当利用から、コンピュータおよびコンピュータネットワークを保護するために、重要な管理上の課題が残る。
【図面の簡単な説明】
【０００５】
本開示およびその特徴ならびに利益に対するより完全な理解を提供するため、添付の図と併せて以下の説明が参照される。同様の参照番号は、同様の部分を表す。
【０００６】
【図１】本開示の一実施形態に係る、マルウェアの検出のための通信システムの簡略ブロック図である。
【０００７】
【図２】本開示の一実施形態に係る、マルウェアの検出のための通信システムの一部の簡略ブロック図である。
【０００８】
【図３】一実施形態に係る、通信システムと関連付けられ得る潜在的な動作を例示する簡略フローチャートである。
【０００９】
【図４】一実施形態に係る、通信システムと関連付けられ得る潜在的な動作を例示する簡略フローチャートである。
【００１０】

【図５】一実施形態に係る、ポイントツーポイント構成で配置される例示的なコンピューティングシステムを例示するブロック図である。
【００１１】
【図６】本開示の例示的なＡＲＭエコシステムシステムオンチップ（ＳＯＣ）と関連付けられる簡略ブロック図である。
【００１２】
【図７】一実施形態に係る、例示的なプロセッサコアを例示するブロック図である。
【００１３】
図面の図は、それらの寸法が、本開示の範囲から逸脱することなく大幅に変更され得ることから、必ずしも縮尺通りに描かれていない。
【発明を実施するための形態】
【００１４】
［例示的な実施形態］　図１は、本開示の一実施形態に係る、マルウェアの検出のための通信システム１００の簡略ブロック図である。図１に例示されているように、通信システム１００の一実施形態は、電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６を含んでよい。電子デバイス１０２は、オペレーティングシステム（ＯＳ）１１０、メモリ１１２、プロセッサ１１４、ハイパーバイザ１１６、セキュリティモジュール１１８、および少なくとも１つのアプリケーション１２０を含んでよい。ＯＳ１１０は、ＯＳ関数１２２およびＯＳ変数１２４を含んでよい。メモリ１１２は、共用ライブラリ１２６を含んでよい。セキュリティモジュール１１８は、システム処理監視モジュール１２８、ホワイトリスト１３０、およびブラックリスト１３２を含んでよい。クラウドサービス１０４およびサーバ１０６は、ネットワークセキュリティモジュール１３４をそれぞれ含んでよい。ネットワークセキュリティモジュール１３４は、ホワイトリスト１３０およびブラックリスト１３２を含んでよい。電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６は、ネットワーク１０８を使用して通信を行い得る。一例において、悪意のあるデバイス１３６は、悪意のあるコード１３８で電子デバイス１０２を感染させるために、ネットワーク１０８または何らかの他の手段（例えば、物理的な接続）を使用することを試み得る。
【００１５】
例示的な実施形態において、通信システム１００は、プロセスのスレッドを監視し、スレッドが、プロセスが既に知っているべき関数を検索することを試みているかを判断するよう構成され得る。共通する関数は、公開されているライブラリで利用可能で、エクスポートライブラリに対してリンクされ、ダイナミックリンクライブラリ（ＤＬＬ）ローダが自動的にアドレスを解決できるので、概して、正規のソフトウェアまたは正規のアプリケーションの一部であるコードは、オペレーティングシステムとインターアクトするべく、共通する関数を検索する必要がない。しかしながら、悪意のあるコードは、多くの場合、様々な関数呼出しの場所を知らず、悪意のあるコードが実行され得る前に、まず関数を見つけなければならない。システム関数に関連する特定のファイルおよび領域を読み取り不可能とマーク付けすることにより、システムは、何がシステム関数の位置を特定するためにファイルおよび領域を読み取っているかを分析でき、コードが信頼できるものか、悪意のあるものかの判断を行うことができる。
【００１６】
図１の要素は、ネットワーク（例えば、ネットワーク１０８）通信のための実行可能な経路を提供する任意の好適な接続（有線または無線）を使用する１または複数のインタフェースを通じて互いに結合されてよい。さらに、図１のこれらの要素のうち任意の１または複数のものは、特定の構成の必要性に基づいて組み合わせられる、またはアーキテクチャから取り除かれてよい。通信システム１００は、ネットワークにおけるパケットの送信または受信のための伝送制御プロトコル／インターネットプロトコル（ＴＣＰ／ＩＰ）通信が可能な構成を含み得る。通信システム１００は、適切な場合、かつ特定の必要性に基づいて、ユーザデータグラムプロトコル／ＩＰ（ＵＤＰ／ＩＰ）または任意の他の好適な

プロトコルと併せて動作してもよい。

【００１７】

通信システム１００の特定の例示的な技術を例示する目的では、ネットワーク環境をトラバースし得る通信を理解することが重要である。以下の基礎的情報は、本開示が適切に説明され得る基礎と見なされてよい。

【００１８】

悪意のあるコード１３８は、ホストコンピュータ（例えば、電子デバイス１０２）を感染させ、ホストコンピュータと関連付られる企業または個人から機密情報を盗取すること、他のホストコンピュータに伝播すること、および／または、分散サービス妨害攻撃を支援すること、ホストコンピュータからスパムまたは悪意のある電子メールを送信することなどの任意の数の悪意のある動作を実行させる、マルウェアまたは悪意のあるソフトウェアであり得る。マルウェアの共通の特徴の１つは、機械上において動作しているソフトウェアの脆弱性を利用するためにシェルコードを使用することである。シェルコードは、ソフトウェアの脆弱性の不当利用において、ペイロードとして使用される１つのコードである。それは、攻撃者が感染した機械を制御できるようにするために、コマンドシェルを一般に起動するので、「シェルコード」と呼ばれる。シェルコードが機械を効果的に感染させる前に、ＯＳ関数またはＯＳルーチン（例えば、ＬｏａｄＬｉｂｒａｒｙ、ＣｒｅａｔｅＦｉｌｅなど）を見つけて、そのペイロードを実行する必要がある。ＯＳルーチンを見つけるためにシェルコードは、ＧｅｔＰｒｏｃＡｄｄｒｅｓｓを呼び出す、またはｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅ（ＰＥ）ヘッダを解析し、ＤＬＬのインポートおよびエクスポートテーブルを見つけて解釈することができる。シェルコードを検出し、悪意のあるアクティビティを特定するためのシステムおよび方法を提供するセキュリティソリューションが必要とされている。

【００１９】

図１で概説されているように、マルウェアの検出のための通信システムが、これら（およびその他）の課題を解決する。通信システム１００は、コードが実行され、データにアクセスするときにコードを監視するべく、ハイパーバイザ（例えば、ハイパーバイザ１１６）メモリベースの監視を使用するよう構成され得る。例えば、メモリ読み込み監視は、マルウェアが実行され得る前に必要とされ得るＯＳ関数を見つけるために、マルウェアが読み取る必要があるデータ構造に対して使用され得る。ＤＬＬが、何らかの関数をプロセスにエクスポートした場合、関数の開始の場所についての情報、ならびに、テーブル（例えば、エクスポートテーブル）に格納されている関数の名前が見つけられ得、それらは、ＤＬＬの始めの部分に、周知の構造により示される。通信システム１００は、ハイパーバイザを使用して、これらの構造およびテーブルを読み取り不可能にし、これにより、プロセスがそれらを読み取った場合、システムがプロセスを分析し、アクセスのパターン、および構造またはテーブルにアクセスしているコードのパターンを調べるよう構成され得る。アクセスされているパターンおよびバイトから、システムは、どの関数が検索されているかを判断でき、コードが、関数を見つけるための悪意のある試みであるか判断できる。

【００２０】

例えば、システム処理監視モジュール１２８は、ＯＳ関数（例えば、ＯＳ関数１２２）およびＯＳ変数（例えば、ＯＳ変数１２４）を検索する（例えば、アプリケーション１２０からの）コードを分析するよう構成され得る。システムの共用ライブラリ（例えば、共用ライブラリ１２６）において、コードを読み取り不可能にすることによる利益はないので、どこでＯＳ関数またはＯＳ変数を見つけるかを示す構造だけが、読み取り不可能にされる。保護され、読み取り不可能とマーク付けされたメモリのエリアは、インポートおよびエクスポートテーブル、ＤＬＬ、ＰＥファイルなどを含んでよい。

【００２１】

図１のインフラストラクチャを参照すると、例示的な一実施形態に係る通信システム１００が示されている。概して、通信システム１００は、任意の種類またはトポロジのネットワークにおいて実装され得る。ネットワーク１０８は、通信システム１００を通じて伝

播する情報パケットを受信するおよび送信するための、相互接続された通信経路の一連のポイントまたはノードを表す。ネットワーク１０８は、ノード間の通信インタフェースを提供し、任意のローカルエリアネットワーク（ＬＡＮ）、仮想ローカルエリアネットワーク（ＶＬＡＮ）、ワイドエリアネットワーク（ＷＡＮ）、無線ローカルエリアネットワーク（ＷＬＡＮ）、メトロポリタンエリアネットワーク（ＭＡＮ）、イントラネット、エクストラネット、仮想プライベートネットワーク（ＶＰＮ）、およびネットワーク環境において通信を容易にする任意の他の適切なアーキテクチャもしくはシステム、または、有線および／または無線通信を含むそれらの任意の好適な組み合わせとして構成され得る。

【００２２】

通信システム１００において、パケット、フレーム、信号、データなどを含むネットワークトラフィックは、任意の好適な通信メッセージングプロトコルに従って送受信され得る。好適な通信メッセージングプロトコルは、オープンシステム間相互接続（ＯＳＩ）モデルのような多層スキーム、またはそれらのあらゆる派生例もしくは変形例（例えば、伝送制御プロトコル／インターネットプロトコル（ＴＣＰ／ＩＰ）、ユーザデータグラムプロトコル／ＩＰ（ＵＤＰ／ＩＰ））を含んでよい。さらに、セルラーネットワークを介した無線信号通信が、通信システム１００において提供されてもよい。好適なインタフェースおよびインフラストラクチャは、セルラーネットワークとの通信を可能にするために提供され得る。

【００２３】

本明細書で使用されている「パケット」という用語は、パケット交換ネットワーク上で、ソースノードと宛先ノードとの間でルーティングされ得るデータ単位を指す。パケットは、ソースネットワークアドレスおよび宛先ネットワークアドレスを含む。これらのネットワークアドレスは、ＴＣＰ／ＩＰメッセージングプロトコルにおけるインターネットプロトコル（ＩＰ）アドレスであってよい。本明細書で使用されている「データ」という用語は、電子デバイスおよび／またはネットワークにおいて１つのポイントから別のポイントに伝達され得る、任意の種類のバイナリ、数値、音声、ビデオ、テキストもしくはスクリプトのデータ、または任意の種類のソースコードもしくはオブジェクトコード、または、任意の適切なフォーマットの任意の他の好適な情報を指す。さらに、メッセージ、要求、応答およびクエリは、ネットワークトラフィックの形式であり、従って、パケット、フレーム、信号、データなどを備え得る。

【００２４】

例示的な一実装例において、電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６は、ネットワーク機器、サーバ、ルータ、スイッチ、ゲートウェイ、ブリッジ、ロードバランサ、プロセッサ、モジュール、またはネットワーク環境で情報を交換するために動作可能な任意の他の好適なデバイス、コンポーネント、要素またはオブジェクトを包含するよう意図されたネットワーク要素である。ネットワーク要素は、それらの動作を容易にする、任意の好適なハードウェア、ソフトウェア、コンポーネント、モジュールまたはオブジェクト、ならびにネットワーク環境において、データもしくは情報を受信、送信および／またはその他の方法で伝達するための好適なインタフェースを含み得る。これは、データまたは情報の効果的な交換を可能にする適切なアルゴリズムおよび通信プロトコルを含み得る。

【００２５】

通信システム１００と関連付けられる内部構造に関して、電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６の各々は、本明細書に概説されている動作において使用されるべき情報を格納するためのメモリ要素を含んでよい。電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６の各々は、任意の好適なメモリ要素（例えば、ランダムアクセスメモリ（ＲＡＭ）、リードオンリメモリ（ＲＯＭ）、消去可能プログラマブルＲＯＭ（ＥＰＲＯＭ）、電気的消去可能プログラマブルＲＯＭ（ＥＥＰＲＯＭ）、特定用途向け集積回路（ＡＳＩＣ）など）、ソフトウェア、ハードウェア、ファームウェア、または、適切でかつ特定の必要性に基づいて任意の他の好適なコンポーネント、

デバイス、要素、もしくはオブジェクトに情報を保存し得る。本明細書に記載されているメモリアイテムの何れも、「メモリ要素」という広義の用語内に包含されていると解釈されるべきである。さらに、通信システム１００において使用されている、追跡されている、送信されている、または受信されている情報は、任意のデータベース、レジスタ、キュー、テーブル、キャッシュ、制御リスト、または他のストレージ構造にて提供され得、その全ては、任意の好適なタイムフレームにおいて参照され得る。そのようなストレージの選択肢の何れも、本明細書で使用されている「メモリ要素」という広義の用語内に含まれ得る。

【００２６】

特定の例示的な実装例において、本明細書に概説されている機能は、非一時的コンピュータ可読媒体を含み得る１または複数の有形媒体内に符号化されているロジック（例えば、ＡＳＩＣ内に設けられる埋込みロジック、デジタル信号プロセッサ（ＤＳＰ）命令、プロセッサ、または、他の同様の機械などにより実行される（オブジェクトコードおよびソースコードを潜在的に含む）ソフトウェア）により実装され得る。これらの例のいくつかにおいて、メモリ要素は、本明細書に説明されている動作に使用されるデータを格納できる。これは、本明細書に説明されているアクティビティを行うために実行されるソフトウェア、ロジック、コードまたはプロセッサ命令を格納することを可能にするメモリ要素を含む。

【００２７】

例示的な一実装例において、電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６のような通信システム１００のネットワーク要素は、本明細書に概説されている動作を実現または促進するためのソフトウェアモジュール（例えば、セキュリティモジュール１１８、システム処理監視モジュール１２８、およびネットワークセキュリティモジュール１３４）を含み得る。これらのモジュールは、特定の構成および／またはプロビジョニングの必要性に基づき得る、任意の適切な態様で、好適に組み合わされ得る。例示的な実施形態において、そのような動作は、意図される機能を実現するべく、これらの要素の外部に実装されるハードウェア、または、何らかの他のネットワークデバイスに含まれるハードウェアにより実行され得る。さらに、モジュールは、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の好適な組み合わせとして実装され得る。これらの要素は、本明細書に概説されているような動作を実現するために、他のネットワーク要素と連携できるソフトウェア（またはレシプロケーティングソフトウェア）も含み得る。

【００２８】

さらに、電子デバイス１０２、クラウドサービス１０４、およびサーバ１０６の各々は、本明細書に記載されているアクティビティを実行するためのソフトウェアまたはアルゴリズムを実行し得るプロセッサを含んでよい。プロセッサは、本明細書に詳述されている動作を実現するために、データと関連付けられた任意の種類の命令を実行できる。１つの例において、プロセッサは、要素または物品（例えばデータ）を１つの状態または物から別の状態または物へ変換できる。別の例において、本明細書に概説されているアクティビティは、固定ロジックまたはプログラマブルロジック（例えば、プロセッサにより実行されるソフトウェア／コンピュータ命令）で実装され得る。本明細書で識別される要素は、デジタルロジック、ソフトウェア、コード、電子命令、またはそれらの任意の好適な組み合わせを含む、何らかの種類のプログラマブルプロセッサ、プログラマブルデジタルロジック（例えば、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、ＥＰＲＯＭ、ＥＥＰＲＯＭ）、またはＡＳＩＣであり得る。本明細書に説明されている、潜在的な処理要素、モジュールおよび機械の何れも、「プロセッサ」という広義の用語内に包含されていると解釈されるべきである。

【００２９】

電子デバイス１０２は、ネットワーク要素であってよく、例えばデスクトップコンピュータ、ラップトップコンピュータ、モバイルデバイス、パーソナルデジタルアシスタント

、スマートフォン、タブレット、または他の同様のデバイスを含む。クラウドサービス１０４は、電子デバイス１０２にクラウドサービスを提供するよう構成される。クラウドサービスは、概して、インターネットようなネットワークを介したサービスとして供給されるコンピューティングリソースを使用することと定義される。通常、計算、ストレージ、およびネットワークリソースは、クラウドインフラストラクチャにおいて提供され、作業負荷をローカルネットワークからクラウドネットワークへ効果的にシフトさせる。サーバ１０６は、サーバまたは仮想サーバのようなネットワーク要素であり得、何らかのネットワーク（例えば、ネットワーク１０８）を介して通信システム１００における通信を開始したいクライアント、顧客、エンドポイントまたはエンドユーザと関連付られ得る。「サーバ」という用語は、通信システム１００内で、クライアントの要求を果たすため、および／またはクライアントに代わって何らかの計算タスクを実行するために使用されるデバイスを含む。セキュリティモジュール１１８は、電子デバイス１０２内に位置するように図１には示されているが、これは例示目的に過ぎない。セキュリティモジュール１１８は、任意の好適な構成で組み合わせられてよく、または分離されてよい。さらに、セキュリティモジュール１１８は、クラウドサービス１０４またはサーバ１０６のような、電子デバイス１０２によりアクセス可能な別のネットワークと統合され得、または、別のネットワーク内に分散され得る。

【００３０】

図２を参照すると、図２は、マルウェアの検出のための通信システム１００の一部の簡略ブロック図である。図２に例示されているように、電子デバイス１０２は、ＯＳ１１０、メモリ１１２、セキュリティモジュール１１８、およびアプリケーション１２０を含んでよい。ＯＳ１１０は、ＯＳ関数１２２およびＯＳ変数１２４を含んでよい。メモリ１１２は、ＤＬＬ１４０、インポートおよびエクスポートテーブル１４２、１または複数のＰＥファイル１４４、ならびにＧｅｔＰｒｏｃＡｄｄｒｅｓｓ１４８を含んでよい。セキュリティモジュール１１８は、システム処理監視モジュール１２８、ホワイトリスト１３０、およびブラックリスト１３２を含んでよい。アプリケーション１２０は、シェルコード１４６を含んでよい。各ＰＥファイル１４４は、ヘッダ１５０を含んでよい。ＧｅｔＰｒｏｃＡｄｄｒｅｓｓ１４８は、ＤＬＬ１４０からエクスポートされた関数または変数のアドレスを取得できる。

【００３１】

アプリケーションが悪意のあるものである、または悪意のあるコード１３８を含む場合、シェルコード１４６が機械を効果的に感染させる前に、シェルコード１４６は、オペレーティングシステム関数またはルーチン（例えば、例示的なＬｏａｄＬｉｂｒａｒｙ、ＣｒｅａｔｅＦｉｌｅなど）を見つけて、そのペイロードを実行する必要がある。ＯＳルーチンを見つけるためにシェルコードは、ＧｅｔＰｒｏｃＡｄｄｒｅｓｓ１４８を呼び出す、またはＰＥファイル１４４からＰＥヘッダを検索するために解析し、ＤＬＬのインポートおよびエクスポートテーブル１４２を見つけて解釈することができる。例えば、ＤＬＬ１４０が、何らかの関数をプロセスにエクスポートする場合、関数の開始についての情報、ならびに関数の名前が見つけられ得る。関数の名前は、ＤＬＬ１４０の始めに周知の構造により示されるインポートおよびエクスポートテーブル１４２に格納され得る。ホワイトリスト１２２は、既知のクリーンなまたは信頼できるアプリケーション、コード、ストリングなどのエントリを含んでよく、誤検知を削減するために使用され得る。ブラックリスト１２４は、既知の悪意のあるまたは信頼されないアプリケーション、コード、ストリングなどのエントリを含んでよい。

【００３２】

図３を参照すると、図３は、一実施形態に係る、マルウェアの検出と関連付けられ得るフロー３００の考えられる動作を例示する例示的なフローチャートである。３０２において、プロセスが動作を開始する。３０４においてシステムは、プロセスが監視されるべきかを判断する。プロセスが監視されるべきではない場合、次にプロセスは、３１０で示されるようにフラグ設定されない。例えば、プロセスは、ホワイトリスト１３０内に見つか

り得、信頼できると分類され得る。加えて、プロセスは、通常、マルウェアについて監視されないプロセスであり得る。プロセスが監視されるべきである（例えば、アプリケーションが不明である、またはブラックリスト１３２内に見つかった）場合、次にシステムは、３０６で示されるように、プロセスがシステム関数を手動で検索（例えば、検索のために解析）しているかを判断する。プロセスが手動でシステム関数を検索（例えば、検索のために解析）していない場合、次にプロセスは、３１０で示されるようにフラグ設定されない。プロセスが手動でシステム関数を検索（例えば、検索のために解析）している場合、次にプロセスは、３０８で示されるようにフラグ設定される。プロセスをフラグ設定することにより、プロセスは、セキュリティモジュール１１８によりマルウェアについて分析され得、または（例えばネットワークセキュリティモジュール１３４による）さらなる分析のためにネットワーク要素に送信され得る。

【００３３】

図４を参照すると、図４は、一実施形態に係る、マルウェアの検出と関連付けられ得るフロー４００の考えられる動作を例示する例示的なフローチャートである。４０２において、アプリケーションが、実行を開始する。４０４において、ＤＬＬテーブルを手動で（例えば、解析し）見つけて解釈するために、アプリケーションは、ＰＥファイルの解析を開始する。４０６において、アプリケーションが悪意のあるものか判断するさらなる分析のために、アプリケーションはフラグ設定される。例えば、プロセスは、セキュリティモジュール１１８によりマルウェアについて分析され得、または（例えばネットワークセキュリティモジュール１３４による）さらなる分析のためにネットワーク要素に送信され得る。

【００３４】

図５は、一実施形態に係る、ポイントツーポイント（ＰｔＰ）構成で配置されるコンピューティングシステム５００を例示する。特に、図５は、プロセッサ、メモリ、および入出力デバイスが、多数のポイントツーポイントインタフェースにより相互接続されるシステムを示す。概して、通信システム１００のネットワーク要素のうち１または複数は、コンピューティングシステム５００と同一または同様の態様で構成されてよい。

【００３５】

図５に例示されているように、システム５００は、いくつかのプロセッサを含み得るが、明確にするために、それらのうちプロセッサ５７０および５８０の２つのみが示されている。２つのプロセッサ５７０および５８０が示されている一方、システム５００の一実施形態は、そのようなプロセッサを１つだけ含んでもよいことが理解されるだろう。プロセッサ５７０および５８０の各々は、プログラムの複数のスレッドを実行するための一連のコア（すなわち、プロセッサコア５７４Ａおよび５７４Ｂ、ならびにプロセッサコア５８４Ａおよび５８４Ｂ）を含んでよい。コアは、図１から図５を参照して上述されているものと同様の態様で、命令コードを実行するよう構成され得る。各プロセッサ５７０、５８０は、少なくとも１つの共有キャッシュ５７１、５８１を含んでよい。共有キャッシュ５７１、５８１は、プロセッサコア５７４および５８４のような、プロセッサ５７０、５８０の１または複数のコンポーネントにより利用されるデータ（例えば、命令）を格納してよい。

【００３６】

プロセッサ５７０および５８０は、メモリ要素５３２および５３４と通信するための集積メモリコントローラロジック（ＭＣ）５７２および５８２をそれぞれ含んでもよい。メモリ要素５３２および／または５３４は、プロセッサ５７０および５８０により使用される様々なデータを格納してよい。代替的な実施形態において、メモリコントローラロジック５７２および５８２は、プロセッサ５７０および５８０とは分離したディスクリートロジックであってよい。

【００３７】

プロセッサ５７０および５８０は、任意の種類のプロセッサであり得、それぞれ、ポイントツーポイントインタフェース回路５７８および５８８を使用して、ポイントツーポイ

ント（ＰｔＰ）インタフェース５５０を介してデータを交換し得る。　プロセッサ５７０および５８０は、ポイントツーポイントインタフェース回路５７６、５８６、５９４および５９８を使用して、個々のポイントツーポイントインタフェース５５２および５５４を介して、チップセット５９０とそれぞれデータを交換し得る。チップセット５９０は、ＰｔＰインタフェース回路であり得るインタフェース回路５９２を使用して、高性能グラフィックスインタフェース５３９を介して、高性能グラフィックス回路５３８とデータを交換してもよい。代替的な実施形態において、図５に例示されている、任意のまたは全てのＰｔＰリンクは、ＰｔＰリンクではなく、マルチドロップバスとして実装され得る。

【００３８】

チップセット５９０は、インタフェース回路５９６を介して、バス５２０と通信を行ってよい。バス５２０は、バスブリッジ５１８およびＩ／Ｏデバイス５１６のような、それを介して通信する１または複数のデバイスを有してよい。バス５１０を介して、バスブリッジ５１８は、キーボード／マウス５１２（またはタッチスクリーン、トラックボールのような他の入力デバイスなど）、通信デバイス５２６（モデム、ネットワークインタフェースデバイス、またはコンピュータネットワーク５６０を通じて通信し得る、他の種類の通信デバイスのような）オーディオＩ／Ｏデバイス５１４、および／またはデータストレージデバイス５２８のような他のデバイスと通信を行ってよい。データストレージデバイス５２８は、プロセッサ５７０および／または５８０により実行され得るコード５３０を格納し得る。代替的な実施形態において、バスアーキテクチャの任意の部分は、１または複数のＰｔＰリンクで実装され得る。

【００３９】

図５に図示されているコンピュータシステムは、本明細書に記載されている様々な実施形態を実装するために利用され得る、コンピューティングシステムの一実施形態の概略図である。図５に図示されているシステムの様々なコンポーネントは、システムオンチップ（ＳｏＣ）アーキテクチャで、または任意の他の好適な構成で組み合わされ得ることが理解されるであろう。例えば、本明細書に開示されている実施形態は、スマートセルラーフォン、タブレットコンピュータ、パーソナルデジタルアシスタント、携帯型ゲーム機などようなモバイルデバイスを含むシステム内に組み込まれ得る。少なくともいくつかの実施形態において、これらのモバイルデバイスには、ＳｏＣアーキテクチャが設けられることが理解されるであろう。

【００４０】

図６を参照すると、図６は、本開示の例示的なＡＲＭエコシステムＳＯＣ６００と関連付けられる簡略ブロック図である。本開示の少なくとも１つの例示的な実装例は、本明細書に記載されているマルウェアの検出の特徴、およびＡＲＭコンポーネントを含んでよい。例えば、図６の例は、任意のＡＲＭコア（例えばＡ－７、Ａ－１５など）と関連付けられてよい。さらに、アーキテクチャは、任意の種類のタブレット、スマートフォン（Ａｎｄｒｏｉｄ（登録商標）フォン、ｉＰｈｏｎｅ（登録商標）を含む）、ｉＰａｄ（登録商標）、Ｇｏｏｇｌｅ　Ｎｅｘｕｓ（登録商標）、Ｍｉｃｒｏｓｏｆｔ　Ｓｕｒｆａｃｅ（登録商標）、パーソナルコンピュータ、サーバ、ビデオ処理コンポーネント、ラップトップコンピュータ（任意の種類のノートブックを含む）、Ｕｌｔｒａｂｏｏｋ（商標）システム、任意の種類のタッチ式入力デバイスなどの一部であってよい。

【００４１】

図６のこの例において、ＡＲＭエコシステムＳＯＣ６００は、ＬＣＤに結合する、モバイルインダストリプロセッサインタフェース（ＭＩＰＩ）／高解像度マルチメディアインタフェース（ＨＤＭＩ（登録商標））リンクと関連付けられ得る、複数のコア６０６－６０７、Ｌ２キャッシュ制御６０８、バスインタフェースユニット６０９、Ｌ２キャッシュ６１０、グラフィックス処理ユニット（ＧＰＵ）６１５、相互接続６０２、ビデオコーデック６２０、および液晶ディスプレイ（ＬＣＤ）Ｉ／Ｆ６２５を含んでよい。

【００４２】

ＡＲＭエコシステムＳＯＣ６００は、加入者識別モジュール（ＳＩＭ）Ｉ／Ｆ６３０、

ブートリードオンリメモリ（ＲＯＭ）６３５、シンクロナスダイナミックランダムアクセスメモリ（ＳＤＲＡＭ）コントローラ６４０、フラッシュコントローラ６４５、シリアル周辺インタフェース（ＳＰＩ）マスタ６５０、好適な電力制御６５５、ダイナミックＲＡＭ（ＤＲＡＭ）６６０、およびフラッシュ６６５も含んでよい。加えて、１または複数の例示的な実施形態は、１または複数の通信能力、インタフェースならびにＢｌｕｅｔｏｏｔｈ（登録商標）６７０、３Ｇモデム６７５、全地球測位システム（ＧＰＳ）６８０、および８０２．１１Ｗｉ－Ｆｉ（登録商標）６８５の例のような特徴を含む。

【００４３】

動作において、図６の例は、比較的低い消費電力とともに処理能力を提供でき、様々な種類のコンピューティング（例えば、モバイルコンピューティング、ハイエンドデジタルホーム、サーバ、無線インフラストラクチャなど）を可能とする。加えて、このようなアーキテクチャは、任意の数のソフトウェアアプリケーション（例えば、Ａｎｄｒｏｉｄ（登録商標）、Ａｄｏｂｅ（登録商標）Ｆｌａｓｈ（登録商標）Ｐｌａｙｅｒ、Ｊａｖａ（登録商標）Ｐｌａｔｆｏｒｍ Ｓｔａｎｄａｒｄ Ｅｄｉｔｉｏｎ（Ｊａｖａ（登録商標）ＳＥ）、Ｊａｖａ（登録商標）ＦＸ、Ｌｉｎｕｘ（登録商標）、Ｍｉｃｒｏｓｏｆｔ Ｗｉｎｄｏｗｓ（登録商標）Ｅｍｂｅｄｄｅｄ、Ｓｙｍｂｉａｎ、およびＵｂｕｎｔｕなど）を可能にし得る。少なくとも１つの例示的な実施形態において、コアプロセッサは、結合された低レイテンシレベル２キャッシュを有するアウトオブオーダスーパースカラパイプラインを実装し得る。

【００４４】

図７は、一実施形態に係るプロセッサコア７００を例示する。プロセッサコア７００は、マイクロプロセッサ、組込みプロセッサ、デジタル信号プロセッサ（ＤＳＰ）、ネットワークプロセッサ、または、コードを実行する他のデバイスのような、任意の種類のプロセッサ用のコアであってよい。図７において、１つのプロセッサコア７００のみが例示されているが、プロセッサは、図７に例示されているプロセッサコア７００の１つより多くを代替的に含んでよい。例えば、プロセッサコア７００は、図５のプロセッサ５７０および５８０に関連して示され、説明されている、プロセッサコア５７４ａ、５７４ｂ、５８４ａ、および５８４ｂの１つの例示的な実施形態を表す。プロセッサコア７００は、シングルスレッドコアであってよく、または、少なくとも１つの実施形態に関して、プロセッサコア７００は、コアごとに１つより多くのハードウェアスレッドコンテキスト（または「ロジカルプロセッサ」）を含み得るという点で、マルチスレッドコアであってよい。

【００４５】

図７はまた、一実施形態に係る、プロセッサコア７００に結合されたメモリ７０２を例示する。メモリ７０２は、既知のまたはそうでなければ当業者に利用可能な多種多様なメモリ（メモリ階層の様々な層を含む）の何れかであってよい。メモリ７０２は、プロセッサコア７００により実行される１または複数の命令であり得るコード７０４を含んでよい。プロセッサコア７００は、コード７０４により示される、命令のプログラムシーケンスに従ってよい。各命令は、フロントエンドロジック７０６に入り、１または複数のデコーダ７０８により処理される。デコーダは、その出力として、予め定義されたフォーマットで固定幅のマイクロオペレーションのようなマイクロオペレーションを生成してよく、または元のコード命令を反映する他の命令、マイクロ命令、もしくは制御信号を生成してよい。フロントエンドロジック７０６はまた、レジスタリネーミングロジック７１０およびスケジューリングロジック７１２を含み、これらは、概して、リソースを割り当て、実行の命令に対応する動作をキューに登録する。

【００４６】

プロセッサコア７００は、一連の実行ユニット７１６－１から７１６－Ｎを有する実行ロジック７１４を含んでもよい。いくつかの実施形態は、特定の機能または複数の機能セット専用の多数の実行ユニットを含み得る。他の実施形態は、１つの実行ユニットのみ、または特定の機能を実行できる１つの実行ユニットを含み得る。実行ロジック７１４は、コード命令により指定される動作を実行する。

【００４７】
コード命令により指定される動作の実行が完了した後、バックエンドロジック７１８は、コード７０４の命令をリタイアできる。１つの実施形態において、プロセッサコア７００は、アウトオブオーダ実行を許可するが、命令のインオーダリタイアメントを必要とする。リタイアメントロジック７２０は、（例えば、リオーダバッファまたは同様の）様々な既知の形式を取ってよい。このように、デコーダ、レジスタリネーミングロジック７１０により利用されるハードウェアレジスタおよびテーブル、ならびに実行ロジック７１４により変更される任意のレジスタ（不図示）により生成される出力に少なくとも関して、プロセッサコア７００は、コード７０４の実行中に変換される。
【００４８】
図７に例示されていないが、プロセッサは、プロセッサコア７００とともに他の要素をチップ上に含んでよく、少なくともそれらのうちいくつかは、図５を参照して本明細書に示され、説明されている。例えば、図５に示されているように、プロセッサは、プロセッサコア７００とともにメモリ制御ロジックを含んでよい。プロセッサは、Ｉ／Ｏ制御ロジックを含んでよく、および／またはメモリ制御ロジックと統合されるＩ／Ｏ制御ロジックを含んでよい。
【００４９】
本明細書に提供されている例に関して、インタラクションは、２つ、３つまたはそれより多くのネットワーク要素に関して説明され得ることに留意されたい。しかしながら、これは、明確性および例示目的のためだけになされたものである。特定の場合には、限定的な数のネットワーク要素のみを参照することにより、所与のセットのフローの１または複数の機能を説明することがより容易になる場合がある。通信システム１００およびその教示は、容易にスケーラブルであり、多数のコンポーネント、ならびにより複雑な／洗練された配置及び構成に適応可能であることを理解されたい。従って、提供されている例は、潜在的に無数の他のアーキテクチャに適用される通信システム１００の範囲を限定する、またはその広範な教示を阻むべきではない。
【００５０】
前述されたフロー図（すなわち、図３から図４）の動作は、通信システム１００により実行され得る、または通信システム１００内に考えられる相関シナリオおよびパターンのうちいくつかのものだけを例示することに留意することも重要である。これらの動作のうちいくつかのものは、適切な箇所で削除されもしくは取り除かれてよく、または、これらの動作は、本開示の範囲から逸脱することなく、大幅に修正もしくは変更されてよい。加えて、これらの動作の多数は、１または複数の追加的な動作と同時にまたは並行して実行されていると説明されている。しかしながら、これらの動作のタイミングは、大幅に変更され得る。前述された動作のフローは、例示および説明目的のために提供されている。任意の好適な配置、時系列、構成、およびタイミングのメカニズムが、本開示の教示から逸脱することなく提供され得るという点で、大きな柔軟性が、通信システム１００により提供される。
【００５１】
本開示は、特定の配置および構成を参照して詳細に説明されているが、これらの例示的な構成および配置は、本開示の範囲から逸脱することなく著しく変更され得る。さらに、特定のコンポーネントが、特定の必要性および実装に基づいて、組み合わされ、分離され、排除され、または追加され得る。さらに、通信システム１００は、通信処理を容易にする特定の要素および動作に関連して例示されているが、これらの要素および動作は、通信システム１００の意図される機能を実現する、任意の好適なアーキテクチャ、プロトコル、および／またはプロセスによって置き換えられてよい。
【００５２】
多数の他の変更、代替、変形、改変および修正が、当業者に確認され得て、本開示は、全てのそのような変更、代替、変形、改変および修正を添付の特許請求の範囲内に含まれるものとして包含することが意図される。米国特許商標庁（ＵＳＰＴＯ）を補助するため

、さらに、本明細書に添付の特許請求の範囲の解釈において、本願に基づいて発行された任意の特許のあらゆる読者を補助するため、出願人は、（ａ）「の手段（ｍｅａｎｓ　ｆｏｒ）」または「の段階（ｓｔｅｐ　ｆｏｒ）」という文言が、特定の特許請求の範囲において具体的に使用されない限り、出願日において本明細書に存在するよう、添付の特許請求の範囲の何れかに米国特許法第１１２条第６段落を援用することを出願人が意図しないこと、および（ｂ）明細書におけるあらゆる記述によって、決して、本開示をそうでなければ添付の特許請求の範囲に反映されないように限定することを出願人が意図しないことに留意することを望んでいる。

【００５３】

［他の留意事項および例］

例Ｃ１は、少なくとも１つのプロセッサにより実行された場合、プロセスを監視することと、プロセスが１または複数のシステム関数を検索するために解析しているかを判断することと、プロセスが１または複数のシステム関数を検索するために解析している場合、プロセスをフラグ設定することとを少なくとも１つのプロセッサに実行させる１または複数の命令を有する少なくとも１つの機械可読媒体である。

【００５４】

例Ｃ２において、例Ｃ１の主題は、プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。

【００５５】

例Ｃ３において、例Ｃ１からＣ２の何れか１つの主題は、プロセスがＧｅｔＰｒｏｃＡｄｄｒｅｓｓを呼び出す場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。

【００５６】

例Ｃ４において、例Ｃ１からＣ３の何れか１つの主題は、プロセスがシェルコードを含むことを任意で含んでよい。

【００５７】

例Ｃ５において、例Ｃ１からＣ４の何れか１つの主題は、少なくとも１つのプロセッサにより実行される場合、１または複数の命令は、マルウェアについてプロセスを分析することを少なくとも１つのプロセッサにさらに実行させることを任意で含んでよい。

【００５８】

例Ｃ６において、例Ｃ１からＣ５の何れか１つの主題は、少なくとも１つのプロセッサにより実行される場合、１または複数の命令は、プロセスがホワイトリスト内に見つかった場合にフラグ設定を除去することを少なくとも１つのプロセッサにさらに実行させることを任意で含んでよい。

【００５９】

例Ａ１において、装置は、システム処理監視モジュールを含んでよい。システム処理監視モジュールは、プロセスを監視し、プロセスが１または複数のシステム関数を検索するために解析しているかを判断し、プロセスが１または複数のシステム関数を検索するために解析している場合、プロセスをフラグ設定するよう構成され得る。

【００６０】

例Ａ２において、例Ａ１の主題は、プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合、プロセスは、１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。

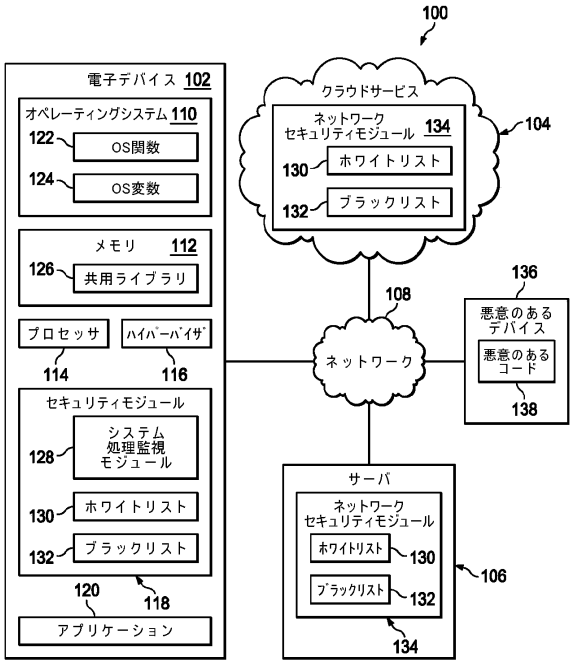
【００６１】

例Ａ３において、例Ａ１からＡ２の何れか１つの主題は、プロセスがＧｅｔＰｒｏｃＡｄｄｒｅｓｓを呼び出す場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。

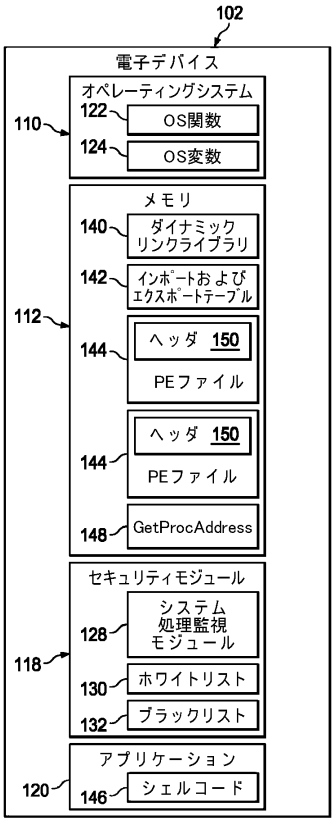
【００６２】
例Ａ４において、例Ａ１からＡ３の何れか１つの主題は、プロセスがシェルコードを含むことを任意で含んでよい。
【００６３】
例Ａ５において、例Ａ１からＡ４の何れか１つの主題は、システム処理監視モジュールが、マルウェアについてプロセスを分析するようさらに構成されることを任意で含んでよい。
【００６４】
例Ａ６において、例Ａ１からＡ５の何れか１つの主題は、プロセスがホワイトリスト内に見つかった場合、システム処理監視モジュールは、フラグ設定を除去するようさらに構成されることを任意で含んでよい。
【００６５】
例Ｍ１は、プロセスを監視する段階と、プロセスが１または複数のシステム関数を検索するために解析しているかを判断する段階と、プロセスが１または複数のシステム関数を検索するために解析している場合、プロセスをフラグ設定する段階とを含む方法である。
【００６６】
例Ｍ２において、例Ｍ１の主題は、プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。
【００６７】
例Ｍ３において、例Ｍ１からＭ２の何れか１つの主題は、プロセスがＧｅｔＰｒｏｃＡｄｄｒｅｓｓを呼び出す場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。
【００６８】
例Ｍ４において、例Ｍ１からＭ３の何れか１つの主題は、プロセスがシェルコードを含むことを任意で含んでよい。
【００６９】
例Ｍ５において、例Ｍ１からＭ４の何れか１つの主題は、マルウェアについてプロセスを分析することを任意で含んでよい。
【００７０】
例Ｓ１は、マルウェアを検出するためのシステムであって、システム処理監視モジュールを含んでよい。システム処理監視モジュールは、プロセスを監視し、プロセスが１または複数のシステム関数を検索するために解析しているかを判断し、プロセスが１または複数のシステム関数を検索するために解析している場合、プロセスをフラグ設定するよう構成され得る。
【００７１】
例Ｓ２において、例Ｓ１の主題は、プロセスがダイナミックリンクライブラリテーブルを見つけて解釈するべく、ｐｏｒｔａｂｌｅ　ｅｘｅｃｕｔａｂｌｅヘッダを解析する場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。
【００７２】
例Ｓ２において、例Ｓ１およびＳ２の何れか１つの主題は、プロセスがＧｅｔＰｒｏｃＡｄｄｒｅｓｓを呼び出す場合、プロセスが１または複数のシステム関数を検索するために解析していると判断されることを任意で含んでよい。
【００７３】
例Ｘ１は、例Ａ１－Ａ６または例Ｍ１－Ｍ５の何れか１つに示されるように、方法を実装する、または装置を実現する機械可読命令を含む機械可読記憶媒体である。例Ｙ１は、例示的な方法Ｍ１－Ｍ５の何れかを実行するための手段を備える装置である。例Ｙ２において、例Ｙ１の主題は、プロセッサおよびメモリを備える方法を実行するための手段を任

意で含んでよい。例Ｙ３において、例Ｙ２の主題は、機械可読命令を備えるメモリを任意で含んでよい。

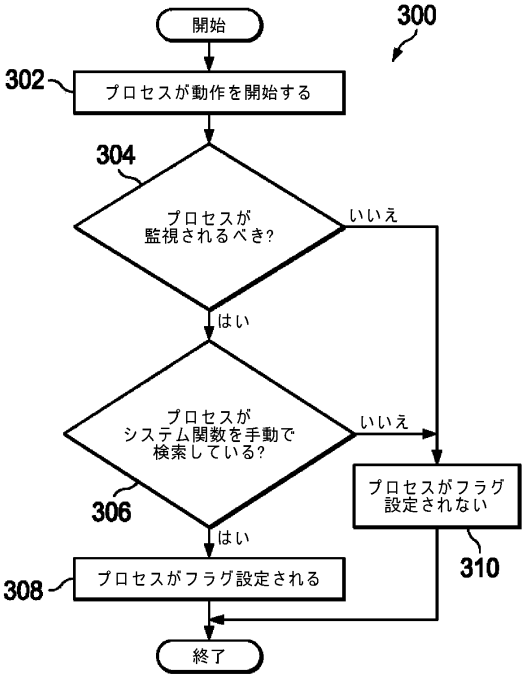
【図１】



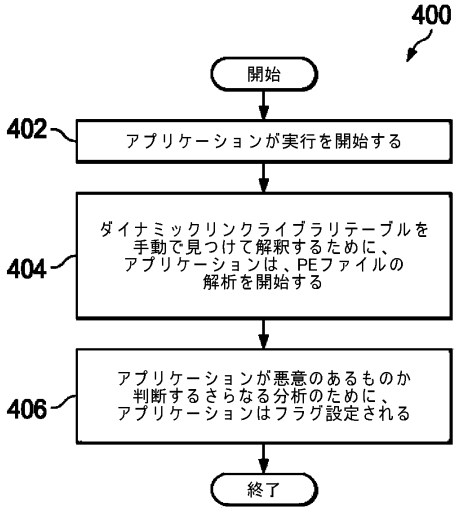
【図２】



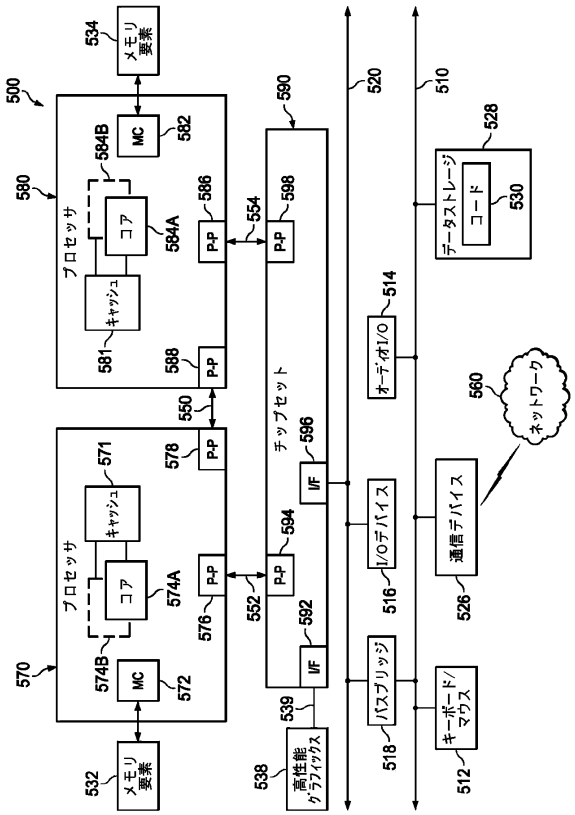
【図３】


300
開始
302
プロセスが動作を開始する
304
プロセスが
監視されるべき?
いいえ
はい
プロセスが
システム関数を手動で
検索している?
いいえ
306
はい
プロセスがフラグ
設定されない
310
308
プロセスがフラグ設定される
終了

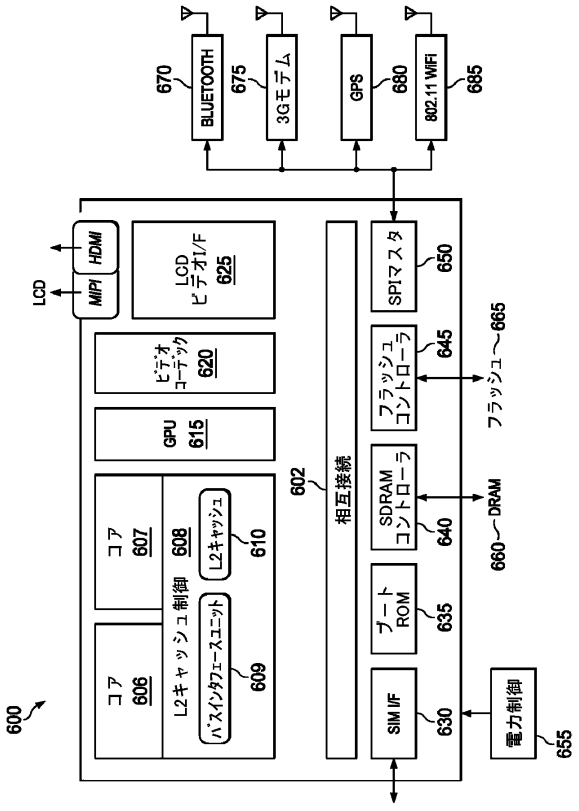
【図４】


400
開始
402
アプリケーションが実行を開始する
404
ダイナミックリンクライブラリテーブルを
手動で見つけて解釈するために、
アプリケーションは、PEファイルの
解析を開始する
406
アプリケーションが悪意のあるものか
判断するさらなる分析のために、
アプリケーションはフラグ設定される
終了

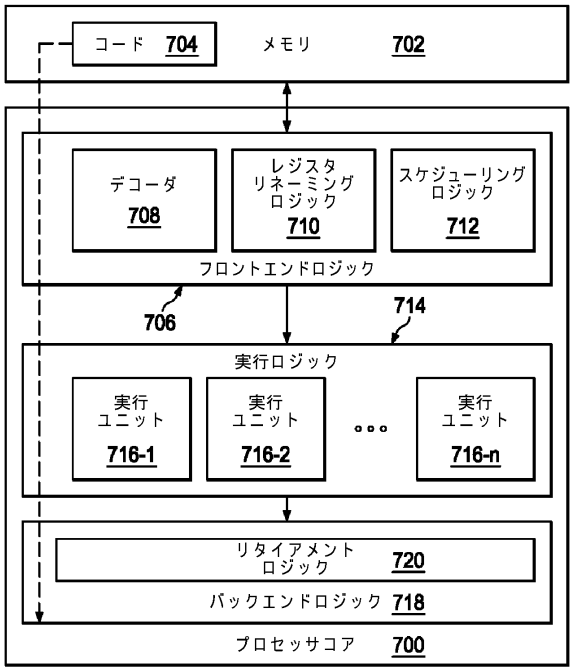
【図５】


500
580
プロセッサ
584B
MC
582
コア
584A
キャッシュ
581
P-P
586
588
554
550
570
プロセッサ
571
キャッシュ
コア
574A
574B
MC
572
578
576
552
590
チップセット
598
596
594
592
IF
539
538
高性能グラフィックス
520
510
528
データストレージ
コード
530
514
オーディオI/O
516
I/Oデバイス
518
バスブリッジ
526
通信デバイス
560
ネットワーク
512
キーボード/マウス
534
メモリ要素
532
メモリ要素

【図６】


600
670
BLUETOOTH
675
3Gモデム
680
GPS
685
802.11 WiFi
LCD
HDMI
MIPI
LCD
ビデオI/F
625
ビデオ
コーデック
620
GPU
615
コア
607
608
L2キャッシュ
610
コア
606
L2キャッシュ制御
バスインタフェースユニット
609
602
相互接続
SPIマスタ
650
フラッシュ
コントローラ
645
フラッシュ～665
SDRAM
コントローラ
640
660～DRAM
ブート
ROM
635
SIM I/F
630
電力制御
655

【図７】


コード 704
メモリ 702
デコーダ 708
レジスタリネーミングロジック 710
スケジューリングロジック 712
フロントエンドロジック
706
714
実行ロジック
実行ユニット 716-1
実行ユニット 716-2
実行ユニット 716-n
リタイアメントロジック 720
バックエンドロジック 718
プロセッサコア 700

フロントページの続き

(72)発明者　スパーロック、ジョーエル　アール.
アメリカ合衆国、９５０５４　カリフォルニア州、サンタ　クララ　ミッション　カレッジ　ブーレバード　２８２１　マカフィー、インコーポレイテッド内

審査官　宮司　卓佳

(56)参考文献　米国特許第０８３０７４３２（ＵＳ，Ｂ１）
米国特許出願公開第２０１２／０２９１１３１（ＵＳ，Ａ１）
特開２０１１－２３３１２６（ＪＰ，Ａ）
特表２０１４－５１４６５１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５６