(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2016-534460
(P2016-534460A)
(43)公表日　平成28年11月4日(2016.11.4)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/56	(2013.01)	G06F	21/56	360	5B376
G06F 9/445	(2006.01)	G06F	9/06	640A	
G06F 9/54	(2006.01)	G06F	9/06	640B	
G06F 11/00	(2006.01)	G06F	9/06	630B	
G06F 21/53	(2013.01)	G06F	21/53		

審査請求 有　予備審査請求 未請求　（全 26 頁）　最終頁に続く

(21)出願番号　特願2016-537815 (P2016-537815)
(86)(22)出願日　平成26年8月27日 (2014.8.27)
(85)翻訳文提出日　平成28年2月24日 (2016.2.24)
(86)国際出願番号　PCT/US2014/052932
(87)国際公開番号　WO2015/031488
(87)国際公開日　平成27年3月5日 (2015.3.5)
(31)優先権主張番号　14/012,520
(32)優先日　平成25年8月28日 (2013.8.28)
(33)優先権主張国　米国 (US)

(71)出願人　515004599
アマゾン　テクノロジーズ　インク
アメリカ合衆国　ワシントン州　９８１０８－１２２６　シアトル　ピー．オー．ボックス　８１２２６
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(72)発明者　ヨハンソン、　ジェスパー、　ミカエル
アメリカ合衆国　ワシントン州　９８１０９－５２１０　シアトル　テリー　アヴェニュー　ノース　４１０

最終頁に続く

(54)【発明の名称】動的アプリケーションセキュリティ検証

(57)【要約】
セキュリティ検証を動的アプリケーションについて行なう種々の実施形態が開示される。アプリケーションのインスタンスが実行される。実行時、アプリケーションが、動的に読み込まれるコードにネットワークサイトからアクセスしようとしているかどうかが確認される。１つの実施形態では、アクセスは、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）を使用することにより検出することができる。別の実施形態では、アクセスは、ダウンロードデータを実行可能メモリ部分に読み込むことにより検出することができる。セキュリティ分析を動的に読み込まれるコードに対して行ない、そして操作を、セキュリティ分析が行なわれると開始する。

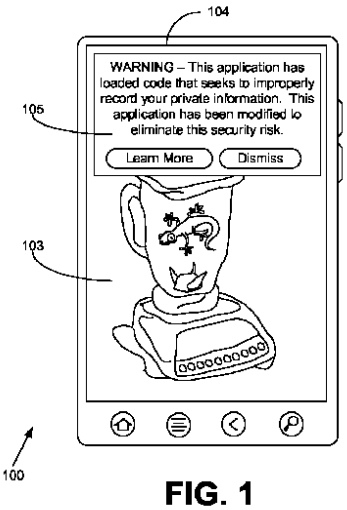


FIG. 1

【特許請求の範囲】
【請求項１】
少なくとも１つのコンピューティング装置と；
前記少なくとも１つのコンピューティング装置内で実行することができるセキュリティ検証サービスと、を備え、前記セキュリティ検証サービスは：
アプリケーションのインスタンスを実行するロジックであって、前記アプリケーションが、アプリケーションマーケットプレイスの中から提供される、前記実行するロジックと；
前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれるコードにネットワークサイトからアクセスしようとしていることを確認するロジックと；
前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードにアクセスしようとしていることが確認されると、セキュリティ分析を動的に読み込まれる前記コードに対して行なうロジックと；
前記セキュリティ分析の結果を、前記アプリケーションマーケットプレイスに送信するロジックと、を含む、システム。
【請求項２】
前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードに前記ネットワークサイトからアクセスしようとしていることを確認する前記ロジックは更に：
前記アプリケーションの前記インスタンスが、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを行なっていることを確認するロジックであって、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードを前記特定のＡＰＩコール以外のコールで取得することが禁止される、前記確認するロジック；
前記アプリケーションの前記インスタンスが、ダウンロードデータを、サンドボックス環境内のメモリの実行可能領域に読み込んでいるかどうかを確認するロジック；または
前記アプリケーションの前記インスタンスが、認識可能な実行可能コードを含むダウンロードデータを有しているかどうかを確認するロジックのうちの少なくとも１つのロジックを含む、請求項１に記載のシステム。
【請求項３】
前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードに前記ネットワークサイトからアクセスしようとしていることを確認する前記ロジックは更に：
前記アプリケーションの前記インスタンスが前記ネットワークサイトからダウンロードしたデータを追跡して、前記データが実行可能コードを含んでいることを確認するロジックを含む、請求項１に記載のシステム。
【請求項４】
前記セキュリティ分析を行なう分析ロジックは：
前のセキュリティ分析が既に、動的に読み込まれる前記コードに対して行なわれているかどうかを確認し；そして
前記前のセキュリティ分析が既に行なわれている場合に、前記前のセキュリティ分析を、前記セキュリティ分析として利用するように構成される、請求項１に記載のシステム。
【請求項５】
前のセキュリティ分析が既に行なわれているかどうかを確認する際に更に：
動的に読み込まれる前記コードの指紋を確認し；そして
前記指紋を、動的に読み込まれるコードに対する複数の前のセキュリティ分析に関連する指紋ライブラリと比較する、請求項４に記載のシステム。
【請求項６】
前記セキュリティ検証サービスは更に：
セキュリティリスクが前記セキュリティ分析で特定されると、前記アプリケーションの

前記インスタンスを終了させるロジック、または動的に読み込まれる前記コードが、前記アプリケーションの前記インスタンスによって読み込まれるのを防止するロジックと；
セキュリティリスクが前記セキュリティ分析で特定されない場合に、前記アプリケーションの前記インスタンスに許可して、動的に読み込まれる前記コードを実行させるロジックと、を含む、請求項１に記載のシステム。
【請求項７】
前記セキュリティ検証サービスは更に、動的に読み込まれる前記コードを修正して、前記セキュリティ分析で特定されるセキュリティリスクを無くすロジックを含む、請求項１に記載のシステム。
【請求項８】
前記セキュリティ検証サービスは更に、前記アプリケーションマーケットプレイスの中から提供される前記アプリケーションの提供アプリケーションを無効化する、または前記提供アプリケーションにフラグを立てるロジックを含む、請求項１に記載のシステム。
【請求項９】
前記アプリケーションの前記インスタンスは、移動体通信端末装置内で、前記実行するロジックによって実行される、請求項１に記載のシステム。
【請求項１０】
前記アプリケーションの前記インスタンスは、サーバ環境内で、前記実行するロジックによって実行され、前記サーバ環境は、前記アプリケーションマーケットプレイスに代わって動作する、請求項１に記載のシステム。
【請求項１１】
少なくとも１つのコンピューティング装置によって、アプリケーションのプロバイダから受信するアプリケーションのインスタンスをサンドボックス環境内で実行し；
前記少なくとも１つのコンピューティング装置によって、前記アプリケーションの前記インスタンスが、ネットワークサイトからダウンロードしたデータを有していることを確認し；
前記少なくとも１つのコンピューティング装置によって、ダウンロードデータが、実行可能コードを含んでいるかどうかを確認し、前記確認する際に：
前記アプリケーションの前記インスタンスが、前記ダウンロードデータの少なくとも一部を、前記サンドボックス環境内のメモリの実行可能領域に既に読み込んでいるかどうかを確認する；または
前記アプリケーションの前記インスタンスが、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを行なって、前記ダウンロードデータを既に取得しているかどうかを確認して、前記アプリケーションの前記インスタンスが、前記実行可能コードを前記特定のＡＰＩコール以外のコールで取得することを禁止されるようにし；
前記ダウンロードデータが前記実行可能コードを含んでいることが確認されると、前記少なくとも１つのコンピューティング装置によって、セキュリティ分析を前記ダウンロードデータの少なくとも一部に対して行ない；そして
前記少なくとも１つのコンピューティング装置から、前記セキュリティ分析の結果を前記アプリケーションの前記プロバイダに送信する、方法。
【請求項１２】
更に：
前記少なくとも１つのコンピューティング装置によって、前記ダウンロードデータの少なくとも一部に関連するコード署名を確認し；そして
前記少なくとも１つのコンピューティング装置によって、前記コード署名に対応する前のセキュリティ分析の結果を受信する、請求項１１に記載の方法。
【請求項１３】
前記アプリケーションの前記プロバイダはアプリケーションマーケットプレイスであり、前記アプリケーションマーケットプレイス及び前記ネットワークサイトは、異なるエンティティによって制御される、請求項１１に記載の方法。

【請求項１４】
更に：
前記セキュリティ分析が行なわれると、前記少なくとも１つのコンピューティング装置によって、前記ダウンロードデータの前記少なくとも一部を修正してセキュリティリスクを無くす；または
前記セキュリティ分析が行なわれると、前記少なくとも１つのコンピューティング装置によって、前記サンドボックス環境の構成を変更して、前記セキュリティリスクを無くす、請求項１１に記載の方法。
【請求項１５】
動的アプリケーション通知を前記アプリケーションの前記プロバイダから受信して、前記動的アプリケーション通知が前記アプリケーションに関連付けられると、前記アプリケーションの前記インスタンスを前記サンドボックス環境内で実行する、請求項１１に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、移動体用アプリケーションについてのセキュリティ検証に関する。
【背景技術】
【０００２】
移動体用アプリケーションは通常、非常に多くの開発者が提供するアプリケーションを特徴とするアプリケーションマーケットプレイスを通じて取得される。顧客は、アプリケーションマーケットプレイスを多種多様な理由から利用することができる。例えば、顧客の移動体通信端末装置は、特定のアプリケーションマーケットプレイスを利用することにより、アプリケーション群を取得する操作を、特定のアプリケーションマーケットプレイスを介して最も容易な選択肢として行なうように予め構成しておくことができる。幾つかの場合では、顧客の移動体通信端末装置は、アプリケーション群を、特定のアプリケーションマーケットプレイスを介してのみ取得するように予め構成しておくことができる。最終的に、顧客は、マーケットプレイスを介して取得されるアプリケーション群は比較的安全であるという感覚があるので、所定のアプリケーションマーケットプレイスを利用することを好む。別の表現をすると、顧客は、アプリケーションマーケットプレイスの所有者に信頼を置けるので、アプリケーションマーケットプレイスを介して提供されるアプリケーション群に信頼を置けるという感覚を持っている可能性がある。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
顧客は、所定のアプリケーションマーケットプレイスを介して提供されるアプリケーション群が安全であり、かつアプリケーションマーケットプレイスの所有者が、提供アプリケーション群のセキュリティを保証しているという感覚を持っている可能性がある。従って、所有者の関心は、セキュリティ検証を、提供するアプリケーション群について行なうことにある。しかしながら、幾つかのアプリケーションについてのセキュリティ分析は、解決困難な課題である。幾つかのアプリケーションが一体化されていることにより、これらのアプリケーションのコードをこれらのアプリケーション全体について、顧客が使用する前に分析することができるが、他のアプリケーション群は動的アプリケーションである可能性があり、コードの一部はアプリケーションを顧客の端末にインストールした後にダウンロードされる。別の表現をすると、アプリケーションマーケットプレイスの所有者は、後でダウンロードしたコードを、セキュリティリスクについて分析する機会を決して持つことができない。
【課題を解決するための手段】
【０００４】
態様の１つは、少なくとも１つのコンピューティング装置と、前記少なくとも１つのコ

ンピューティング装置内で実行することができるセキュリティ検証サービスと、を備える。前記セキュリティ検証サービスは、アプリケーションのインスタンスを実行するロジックであって、前記アプリケーションが、アプリケーションマーケットプレイスの中から提供される、前記実行するロジックと、前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれるコードにネットワークサイトからアクセスしようとしていることを確認するロジックと、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードにアクセスしようとしていることが確認されると、セキュリティ分析を動的に読み込まれる前記コードに対して行なうロジックと、前記セキュリティ分析の結果を、前記アプリケーションマーケットプレイスに送信するロジックと、を含む。

【発明の効果】

【０００５】

本開示は、セキュリティ検証を動的アプリケーション群に対して行なうアプローチを提供することができる。

【図面の簡単な説明】

【０００６】

【図１】本開示の１つの実施形態によるユーザインターフェースをディスプレイにレンダリングするクライアント装置の一実施例の図面である。

【図２】本開示の種々の実施形態によるネットワーク構成環境の図面である。

【図３】本開示の種々の実施形態による図２のネットワーク構成環境のコンピューティング環境内で実行されるアプリケーションマーケットプレイスシステムの一部として実行される機能の一実施例を示すフローチャートである。

【図４】本開示の種々の実施形態による図２のネットワーク構成環境のコンピューティング環境内で実行されるセキュリティ検証サービスの一部として実行される機能の一実施例を示すフローチャートである。

【図５】本開示の種々の実施形態による図２のネットワーク構成環境に用いられるコンピューティング環境を表わす１つの実施例を与える模式ブロック図である。

【図６】本開示の種々の実施形態による図２のネットワーク構成環境に用いられるクライアント装置を表わす１つの実施例を与える模式ブロック図である。

【発明を実施するための形態】

【０００７】

本開示の多くの態様は、以下の図面を参照することにより一層深く理解することができる。これらの図面に含まれる構成要素群は、必ずしも寸法通りにはなっていないが、その代り、本開示の原理を明確に示すために誇張されている。更に、これらの図面では、同様の参照番号は、対応する構成要素群を、幾つかの図面を通じて指している。

【０００８】

本開示は、移動体用アプリケーションについてのセキュリティ検証に関するものである。顧客は、所定のアプリケーションマーケットプレイスを介して提供されるアプリケーション群が安全であり、かつアプリケーションマーケットプレイスの所有者が、提供アプリケーション群のセキュリティを保証しているという感覚を持っている可能性がある。従って、所有者の関心は、セキュリティ検証を、提供するアプリケーション群について行なうことにある。しかしながら、幾つかのアプリケーションについてのセキュリティ分析は、解決困難な課題である。幾つかのアプリケーションが一体化されていることにより、これらのアプリケーションのコードをこれらのアプリケーション全体について、顧客が使用する前に分析することができるが、他のアプリケーション群は動的アプリケーションである可能性があり、コードの一部はアプリケーションを顧客の端末にインストールした後にダウンロードされる。別の表現をすると、アプリケーションマーケットプレイスの所有者は、後でダウンロードしたコードを、セキュリティリスクについて分析する機会を決して持つことができない。

【０００９】

非限定的な実施例として、トカゲをミキサーに入れて混ぜ合わせるゲームとして現われるアプリケーションが提供される可能性がある。このアプリケーションは、種々のセキュリティ許可を顧客の端末に対して害を及ぼすことなく要求するように見える。しかしながら、一旦、このアプリケーションが顧客の端末で実行されると、アプリケーションは、顧客の知らないうちに、パスワード、クレジットカード番号、及び他の個人情報を収集して送信する不正コードをダウンロードしてバックグランドで実行する虞れがある。この問題に対する１つの単純な解決策では、動的コードを有する如何なるアプリケーションもアプリケーションマーケットプレイスから提供されることがないようにアプリケーションをブロックする。セキュリティリスクを未然に防ぐが、このような解決策では、動的コードを使用することにより開発者に与えられる多くの恩恵を無視することになる。例えば、動的コードを使用すると、より多くのフレキシビリティを開発者に提供することができ、かつ開発者が新製品リリースをより迅速に行なうことができる。動的コードを使用しない場合には、コードの微調整のたびに、開発者は、新バージョンのアプリケーションを提供して、アプリケーションマーケットプレイスに承認されるようにする必要がある。

【００１０】

本開示の種々の実施形態は、セキュリティ検証を動的アプリケーション群に対して行なうアプローチを提供する。動的アプリケーションで取得されるコードが特定される。１つの実施形態では、アプリケーション群は、取得コードを取得データから識別するアプリケーションプログラミングインターフェース（ＡＰＩ）を使用して書き込まれる。別の実施形態では、アプリケーション群を動作させて挙動を観測し、そしてサンドボックス実行環境で、取得データが、システムメモリの実行可能コード領域に、またはシステムメモリのデータ領域に格納されるかどうかを追跡する。一旦、取得コードが確認されると、当該コードをセキュリティリスクについて分析することができる、または既に分析されているコードと比較することができる。幾つかの場合では、検出されたセキュリティ問題は、自動的に修復することができる、または当該アプリケーションを無効化する、かつ／またはアプリケーションマーケットプレイスから排除することができる。

【００１１】

図１を参照するに、図示されているのは、１つの実施形態によるユーザインターフェース１０３をディスプレイ１０４にレンダリングするクライアント装置１００の一実施例である。ユーザインターフェース１０３は、前述の実施例において説明したトカゲをミキサーに入れて混ぜ合わせるゲームアプリケーションにより生成される。本開示の原理によれば、セキュリティ検証サービスは、個人情報をクライアント装置１００から不正に記録しようとしている実行可能コードをゲームアプリケーションが読み込んでしまっていることを検出したところである。その結果、セキュリティ検証サービスは、当該アプリケーションを修正して、セキュリティリスクを無くしてしまっている。ユーザインターフェース１０３にレンダリングされるユーザインターフェースコンポーネント１０５は、ユーザに、当該アプリケーションが修正されてセキュリティリスクを解消してしまっていることを通知する。続いて、ユーザは、不正部分が無くなった後に、当該アプリケーションを使用し続けることができる。ユーザインターフェースコンポーネント１０５が図１の実施例に図示されているが、他の実施例では、当該アプリケーションは、ユーザ通知を行なうことなく、修正する、ブロックする、終了させるなどとすることができる。以下の説明では、システム、及び当該システムのコンポーネントについての概要説明が行なわれ、続いてシステム及びコンポーネントの動作についての説明が行なわれる。

【００１２】

次に、図２を参照するに、図示されているのは、種々の実施形態によるネットワーク構成環境１１０である。ネットワーク構成環境１１０は、コンピューティング環境１１３と、コンピューティング環境１１６と、データ通信を、ネットワーク１１９を介して行なう１つ以上のクライアント装置１００と、を含む。ネットワーク１１９は、例えばインターネット、イントラネット、エキストラネット、ワイドエリアネットワーク（ＷＡＮ）、ローカルエリアネットワーク（ＬＡＮ）、有線ネットワーク、無線ネットワーク、ケーブル

ネットワーク、衛星ネットワーク、または他の適切なネットワークなどを含む、または２つ以上のこのようなネットワークの任意の組み合わせを含む。

【００１３】

コンピューティング環境１１３は、例えばサーバコンピュータ、または計算機能を提供する他のいずれかのシステムを備えることができる。別の構成として、コンピューティング環境１１３は、複数のコンピューティング装置を用いることができ、これらのコンピューティング装置は、例えば１つ以上のサーババンクに、またはコンピュータバンクに、或いは他の機構に配置される。このようなコンピューティング装置は、単一の設置箇所に格納するか、または多くの異なる地理的位置に分散配置することができる。例えば、コンピューティング環境１１３は、複数のコンピューティング装置を含むことができ、これらのコンピューティング装置は一体となって、ホスト上の利用可能なコンピューティングリソース、グリッドコンピューティングリソース、及び／または他のいずれかの分散コンピューティング機構を構成することができる。幾つかの場合では、コンピューティング環境１１３は、エラスティックコンピューティングリソースに対応させることができ、この場合、割り当て処理能力、ネットワーク、ストレージ、または他の計算関連リソースは経時的に変わる可能性がある。

【００１４】

種々のアプリケーション、及び／または他の機能は、種々の実施形態によるコンピューティング環境１１３で実行することができる。また、種々のデータは、コンピューティング環境１１３にアクセス可能なデータストア１２２に格納される。データストア１２２は、図から分かるように、複数のデータストア１２２を表わすことができる。データストア１２２に格納されるデータは、例えば以下に説明する種々のアプリケーション及び／または機能エンティティの動作に関連付けられる。

【００１５】

コンピューティング環境１１３で実行されるコンポーネント群は、例えばアプリケーションマーケットプレイスシステム１２５、セキュリティ検証サービス１２８、サンドボックス環境１２９、及び本明細書において詳細には説明されない他のアプリケーション、サービス、プロセス、システム、エンジン、または機能を含む。アプリケーションマーケットプレイスシステム１２５が実行されると、アプリケーション群１３１を複数の開発者から容易に供給することができる。１つの実施形態では、アプリケーションマーケットプレイスシステム１２５は、本明細書において、所有者（ｐｒｏｐｒｉｅｔｏｒ）と表記される単一のエンティティによって管理される。アプリケーションマーケットプレイスシステム１２５は、セキュリティ検証サービス１２８を用いてセキュリティ分析を、アプリケーション群１３１に対して行なうことができる。セキュリティ検証サービス１２８で、アプリケーション１３１が、セキュリティリスクを全く含んでいないことが確認される場合、アプリケーション１３１をアプリケーションマーケットプレイスシステム１２５から提供することができる。

【００１６】

セキュリティ検証サービス１２８を実行してこれらの分析を行なうことができる。この分析を行なうために、セキュリティ検証サービス１２８は、種々のツールを用いて、アプリケーション１３１が、セキュリティリスクを含んでいるかどうかを確認することができる。例えば、セキュリティ検証サービス１２８は、アプリケーション１３１が、既知の不正サーバと通信しているかどうかを検出することができる、セキュリティ検証サービス１２８は、アプリケーション１３１内の既知の不正コードの署名を検出することができる、またはセキュリティ検証サービス１２８は、既知の挙動または挙動パターンを、不正なアプリケーション１３１から検出することができる。幾つかの実施形態では、セキュリティ検証サービス１２８は、他の種類のコンテンツ検査またはコード検査を実施するサードパーティツールを用いることができる。幾つかの実施形態では、セキュリティ検証サービス１２８は、アプリケーション１３１を修復して、またはその他には、修正して、検出されるセキュリティリスクを取り除くか、またはその他には、無くすように構成することがで

きる。
【００１７】
アプリケーション１３１が一体化されている、または内蔵されている場合、セキュリティ分析は比較的簡単に行なうことができる。セキュリティ分析の種々の実施例は、２０１３年６月２５日に出願され、かつ“ＡＮＡＬＹＺＩＮＧ　ＳＥＣＵＲＩＴＹ　ＯＦ　ＡＰＰＬＩＣＡＴＩＯＮＳ（アプリケーションのセキュリティ分析）”と題する米国特許出願第１３／９２６，２１１号に開示されており、この米国特許出願は、本明細書において参照されることにより、当該米国特許出願全体が本明細書に組み込まれる。しかしながら、アプリケーション１３１が、外部サイトから取得されるコードを読み込む動的アプリケーションである場合、セキュリティ分析は、アプリケーション１３１のコードが、アプリケーション１３１がクライアント装置１００にダウンロードされた後に変化する可能性があるので一層極めて困難となる。
【００１８】
１つの実施形態では、セキュリティ検証サービス１２８は、アプリケーション取り込み処理の一部として、アプリケーションインスタンス１３４をサンドボックス環境１２９内で実行することができる。サンドボックス環境１２９は、この状況では、クライアント装置１００をエミュレートする仮想化環境に対応させることができる。別の構成として、サンドボックス環境１２９は、コードの実行を監視し、かつアプリケーションからデータ及び／またはシステムサービスへのアクセスを禁止する実行環境に対応させることができる。サンドボックス環境１２９は、実行可能コードメモリ１３７と、データメモリ１４０と、を含むことができる。アプリケーションインスタンス１３４が、サンドボックス環境１２９内で実行されると、セキュリティ検証サービス１２８は、アプリケーションインスタンス１３４により外部ネットワークサイト１４７から取得されて動的に読み込まれるコード１４３を検出することができる。従って、セキュリティ検証サービス１２８は、セキュリティ分析を、動的に読み込まれるコード１４３に対して行なうことができる。コンピューティング環境１１３に常駐するセキュリティ検証サービス１２８について説明してきたが、セキュリティ検証サービス１２８の種々のサービス構成部分は、これらのクライアント装置１００においても実行することができることを理解されたい。更に、クライアント装置１００においてではなく、セキュリティ検証サービス１２８の少なくとも１つのサービス構成部分は、ネットワーク１１９の内部のネットワーク装置、例えばファイアウォール、負荷バランサ、ルータなどにおいて実行してもよい。
【００１９】
データストア１２２に格納されるデータは、例えばアプリケーションマーケットプレイスデータ１５０、アプリケーション群１３１、動的に読み込まれるコードデータ１５３、前のセキュリティ分析結果１５７、セキュリティ分析設定データ１６０、及び場合によっては他のデータを含む。アプリケーションマーケットプレイスデータ１５０は、アプリケーションマーケットプレイスシステム１２５の機能をサポートするデータを含み、当該データは、アプリケーション群１３１からなる複数の提供アプリケーション１６３を含む。提供アプリケーション群１６３の各提供アプリケーションは、タイトル、記述、価格、デバイス互換性情報、スクリーンショット、顧客審査、顧客評価、ダウンロード統計、アプリケーション１３１が要求するデバイス優先度、及び／または他の情報に関連付けることができる。これらの提供アプリケーション１６３は更に、アプリケーション１３１のセキュリティ検証レベルの指標値に関連付けることができる。セキュリティ分析により完全には検証することができないアプリケーション群１３１は、完全に分析されているアプリケーション群１３１と比較して、相対的に低いセキュリティ検証レベルに関連付けることができる。
【００２０】
アプリケーション群１３１は、アプリケーションマーケットプレイスシステム１２５を介して提供されるアプリケーション１３１を実行するパッケージ群またはコードに対応する。アプリケーション群１３１は、移動体用アプリケーション、例えばスマートフォン、

タブレット、電子ブックリーダなどに搭載されるアプリケーション、またはデスクトップ用アプリケーション、例えば場合によっては、デスクトップコンピュータ、ラップトップコンピュータなどに搭載されるアプリケーションとすることができる。アプリケーション群１３１は、アプリケーション１３１のソース、完全性、及び／またはバージョンを検証するために使用することができる署名、指紋、チェックサム、バージョン情報、及び／または他の情報に関連付けることができる。アプリケーション群１３１は、ネイティブアプリケーションまたは一体型アプリケーション、アプリケーションマーケットプレイスから提供される動的アプリケーション用のコンテナ、顧客から提供される動的アプリケーション用のコンテナ、及び／または他の種類のアプリケーションに対応させることができる。

【００２１】

動的に読み込まれるコードデータ１５３は、動的に読み込まれるコード１４３のバージョンの、またはパッケージを含むことができ、これらのコードをセキュリティ検証サービス１２８が既に処理している。幾つかの実施形態では、動的に読み込まれるコードデータ１５３は、動的に読み込まれるコード１４３のソース、完全性、及び／またはバージョンを検証するために使用することができる署名、指紋、チェックサム、バージョン情報、及び／または他の情報を含むことができる。前のセキュリティ分析結果１５７は、動的に読み込まれる種々のバージョンのコード１４３に対してセキュリティ検証サービス１２８によって行なわれた前のセキュリティ分析の結果に対応する。セキュリティ分析設定データ１６０は、不正コードの検出、不正コードの修復を設定する、かつ／またはセキュリティ検証サービス１２８によって行なわれる、または開始される他の操作を設定するデータを含むことができる。

【００２２】

コンピューティング環境１１６は、例えばサーバコンピュータまたは計算機能を提供する他のいずれかのシステムを備えることができる。別の構成として、コンピューティング環境１１６は、複数のコンピューティング装置を用いることができ、これらのコンピューティング装置は、例えば１つ以上のサーババンクに、またはコンピュータバンクに、或いは他の機構に配置される。このようなコンピューティング装置は、単一の設置箇所に格納するか、または多くの異なる地理的位置に分散配置することができる。例えば、コンピューティング環境１１６は、複数のコンピューティング装置を含むことができ、これらのコンピューティング装置は一体となって、ホスト上で利用可能なコンピューティングリソース、グリッドコンピューティングリソース、及び／または他のいずれかの分散コンピューティング機構を構成することができる。幾つかの場合では、コンピューティング環境１１６は、エラスティックコンピューティングリソースに対応させることができ、この場合、割り当て処理能力、ネットワーク、ストレージ、または他の計算関連リソースは経時的に変わる可能性がある。

【００２３】

種々のアプリケーション、及び／または他の機能は、種々の実施形態によるコンピューティング環境１１６で実行することができる。また、種々のデータは、コンピューティング環境１１６にアクセス可能なデータストアに格納することができる。コンピューティング環境１１６で実行されるコンポーネント群は、例えば外部ネットワークサイト１４７、及び本明細書において詳細には説明されない他のアプリケーション、サービス、プロセス、システム、エンジン、または機能を含む。外部ネットワークサイト１４７は、アプリケーションマーケットプレイスシステム１２５の所有者とは異なるエンティティによって動作させられ、かつ動的に読み込まれるコード１４３を、クライアント装置１００及び／またはコンピューティング環境１１３で実行されているアプリケーション群１３１のインスタンスに付与するように構成される。外部ネットワークサイト１４７は、アプリケーション１３１の開発者によって、またはサードパーティによって動作させることができる。幾つかの場合では、外部ネットワークサイト１４７は、マルウェアを、動的に読み込まれるコード１４３を利用して拡散させようとする不正ユーザによって動作させられる、または悪用される可能性がある。

【００２４】
クライアント装置１００は、ネットワーク１１９に接続することができる複数のクライアント装置１００を表わしている。クライアント装置１００は、例えばコンピュータシステムのようなプロセッサ利用システムを備えることができる。このようなコンピュータシステムは、デスクトップコンピュータ、ラップトップコンピュータ、携帯情報端末、携帯電話機、スマートフォン、セットトップボックス、ミュージックプレーヤ、ウェブパッド、タブレットコンピュータシステム、ゲームコンソール、電子ブックリーダ、または同様の機能を備える他のデバイスの形態で具体化することができる。クライアント装置１００はディスプレイ１０４を含むことができる。ディスプレイ１０４は、例えば液晶表示（ＬＣＤ）ディスプレイ、ガスプラズマ放電を行う方式のフラットパネルディスプレイ、有機発光ダイオード（ＯＬＥＤ）ディスプレイ、電気泳動インク（Ｅ Ｉｎｋ）を用いた電気泳動ディスプレイ、ＬＣＤプロジェクタ、または他の種類のディスプレイ装置などのような１つ以上の装置を備えることができる。

【００２５】
クライアント装置１００は、マーケットプレイスクライアントアプリケーション１６６、サンドボックス環境１２９、アプリケーションインスタンス１３４、セキュリティ検証クライアントサービス１６９、及び／または他のアプリケーションのような種々のアプリケーションを実行するように構成することができる。マーケットプレイスクライアントアプリケーション１６６は、例えばクライアント装置１００内で実行することにより、コンピューティング環境１１３、１１６、及び／または他のサーバから与えられるネットワークコンテンツにアクセスして、ユーザインターフェース１０３をディスプレイ１０４にレンダリングすることができる。詳細には、マーケットプレイスクライアントアプリケーション１６６をユーザがクライアント装置１００に用いて、アプリケーションマーケットプレイスシステム１２５と対話する。マーケットプレイスクライアントアプリケーション１６６は、アプリケーション１３１の検索、アプリケーション１３１の購入、アプリケーション１３１のダウンロード、アプリケーション１３１のインストール、及び／またはアプリケーションマーケットプレイスシステム１２５に対する、そしてクライアント装置１００に対する他の操作を容易にすることができる。幾つかの場合では、マーケットプレイスクライアントアプリケーション１６６は、例えばブラウザ、移動体用アプリケーションなどに対応させることができ、ユーザインターフェース１０３は、ネットワークページ、移動体用アプリケーションスクリーンなどに対応させることができる。

【００２６】
種々の実施形態では、アプリケーションインスタンス群１３４は、クライアント装置１００内で、またはサンドボックス環境１２９内で別々に実行することができる。サンドボックス環境１２９を用いて、動的に読み込まれるコード１４３を監視し、当該コードは、アプリケーションインスタンス群１３４によってダウンロードされてメモリに読み込まれる。コンピューティング環境１１３のサンドボックス環境１２９と同じように、クライアント装置１００のサンドボックス環境１２９は、実行可能コードメモリ１３７と、データメモリ１４０と、を含むことができる。クライアント装置１００のサンドボックス環境１２９は、エミュレーション、及び／または仮想化を実現することができる、または実現する必要はなく；実現しない場合には、サンドボックス環境１２９は単に、アプリケーションインスタンス群１３４がクライアント装置１００のリソース群に直接アクセスするのを防止するレイヤとすることができる。

【００２７】
セキュリティ検証クライアントサービス１６９は、セキュリティ検証サービス１２８のクライアント実行部分に対応する。アプリケーションインスタンス群１３４宛てに動的に読み込まれるコード１４３は、実行時には変化している可能性があるので、動的に読み込まれるコード１４３のセキュリティ分析の少なくとも一部をクライアント装置１００内で行なうと有利となる。例えば、セキュリティ検証クライアントサービス１６９は、動的に読み込まれるコード１４３のバージョンを検証して、前のセキュリティ分析１５７が確実

に行なわれた状態を維持する。セキュリティ検証クライアントサービス１６９は更に、修復機能を含むことにより、動的に読み込まれるコード１４３について検出されるあらゆるセキュリティリスクを除去しようとする。幾つかの実施例では、セキュリティ検証クライアントサービス１６９の少なくとも１つのサービス構成部分を、ネットワーク１１９内のネットワーク装置により、例えばファイアウォール、負荷バランサ、ルータ、及び／または他の装置により実行することができる。

【００２８】

クライアント装置１００は、マーケットプレイスクライアントアプリケーション１６６、サンドボックス環境１２９、アプリケーションインスタンス群１３４、及びセキュリティ検証クライアントサービス１６９以外のアプリケーション群、例えばブラウザ、移動体用アプリケーション、ｅｍａｉｌアプリケーション、ソーシャルネットワーキングアプリケーション、及び／または他のアプリケーションを実行するように構成することができる。

【００２９】

次に、ネットワーク構成環境１１０の種々のコンポーネントの動作についての概要説明を行なう。最初に、開発者または他のユーザは、アプリケーション群１３１をアプリケーションマーケットプレイスシステム１２５に供給する。アプリケーションマーケットプレイスシステム１２５は、提供されて承認されるアプリケーション１３１の種類を確認することができる。アプリケーション群１３１がネイティブアプリケーションまたは一体型アプリケーションである場合、セキュリティ検証サービス１２８は、セキュリティ分析をアプリケーション群１３１に対して行ない、セキュリティリスクが検出されると、検出されたセキュリティリスクを含むアプリケーション群１３１を拒否する、修復する、アプリケーション群１３１にフラグを立てるなどとすることができる。

【００３０】

アプリケーション群１３１が動的アプリケーションである場合、セキュリティ検証サービス１２８は、種々のアプローチをセキュリティ分析に適用することができる。第１の一連のアプローチでは、アプリケーション１３１をサンドボックス環境１２９内でアプリケーションインスタンス１３４として実行し、セキュリティ検証サービス１２８は、アプリケーションインスタンス１３４が、動的に読み込まれるコード１４３に外部ネットワークサイト１４７からアクセスすることを要求している、またはアクセスしようとしているかどうかを監視する。アプリケーションインスタンス１３４が、動的に読み込まれるコード１４３にアクセスすることを要求している、またはアクセスしようとしている場合、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３をセキュリティリスクについて分析することができる。

【００３１】

適用可能である場合、セキュリティ検証サービス１２８は、検出されるセキュリティリスクに対応する問題を修正または修復しようとすることができる。更に、セキュリティ検証サービス１２８は、セキュリティリスクが検出されると、アプリケーションマーケットプレイスシステム１２５内のアプリケーション１３１の提供アプリケーション１６３を修正する、提供アプリケーション１６３にフラグを立てる、または提供アプリケーション１６３を中断することができる。セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３のチェックサム、署名、指紋、プロファイルなどを、動的に読み込まれるコードデータ１５３に格納して、当該コードを将来時点で識別することができるようにする。更に、セキュリティ検証サービス１２８は、セキュリティ分析の結果を、前のセキュリティ分析結果１５７に格納することができる。

【００３２】

第２の一連のアプローチでは、動的アプリケーション１３１は、クライアント装置１００にダウンロードしてインストールすることができる。アプリケーション１３１は、アプリケーションインスタンス１３４として別々に実行することができる、またはクライアント装置１００のサンドボックス環境１２９内で実行することができる。アプリケーション

インスタンス１３４が、動的に読み込まれるコード１４３にアクセスすることを要求している、またはアクセスしようとしている場合、セキュリティ検証クライアントサービス１６９は、動的に読み込まれるコード１４３をセキュリティリスクについて分析することができる。セキュリティ検証クライアントサービス１６９は、セキュリティ分析の結果をセキュリティ検証サービス１２８に報告することができる。

【００３３】

セキュリティ分析をクライアント装置１００内で行なうと、プロセッサ利用効率、メモリ利用効率、及び／またはバッテリ消費量の点でコストが比較的高く付いてしまう。幾つかの実施形態では、セキュリティ分析をクライアント装置１００内で行なうのではなく、セキュリティ検証クライアントサービス１６９は、動的に読み込まれるコード１４３、または動的に読み込まれるコード１４３のソースに対応するユニフォームリソースロケータ（ＵＲＬ）を、セキュリティ検証サービス１２８に送信して、分析をコンピューティング環境１１３内で行なうようにしてもよい。別の構成として、セキュリティ検証クライアントサービス１６９は、動的に読み込まれるコード１４３のチェックサム、指紋、署名、または他のバージョン識別子を確認し、次にセキュリティ検証サービス１２８にクエリを送信して、前のセキュリティ分析１５７の結果を、利用可能な場合に、確認するようにしてもよい。１つの実施形態では、Ｃｅｒｔｉｆｉｃａｔｅ　ｐｉｎｎｉｎｇ（証明書のピン留め）を利用して、外部ネットワークサイト１４７から取得されるデータの署名を検証することができる。前のセキュリティ分析結果１５７を利用することができない場合、セキュリティ検証サービス１２８及び／またはセキュリティ検証クライアントサービス１６９は、セキュリティ分析を、動的に読み込んで新たに取得されるコード１４３に対して行なって、これらの結果を次に、前のセキュリティ分析結果１５７として格納することができる。

【００３４】

セキュリティ検証クライアントサービス１６９は、修復機能を実行して、セキュリティ検証クライアントサービス１６９及び／またはセキュリティ検証サービス１２８で発見されるセキュリティ問題を解決するように構成することができる。セキュリティ検証クライアントサービス１６９は、動的に読み込まれる攻撃コード１４３を修復する、置き換える、または消去することにより、セキュリティリスクを無くすことができる。これにより、アプリケーションインスタンス１３４の動作を停止することになるが、アプリケーションインスタンス１３４を停止する方が、セキュリティリスクが検出された状態で動作を継続するよりも望ましい。幾つかの場合では、セキュリティ検証クライアントサービス１６９は、アプリケーションインスタンス１３４を終了することを選択してもよい。セキュリティリスクが検出される、かつ／または無くなる場合、クライアント装置１００のユーザに、これらの結果を通知することができる。

【００３５】

セキュリティリスクを検出して無くす非限定的な実施例として、セキュリティ検証クライアントサービス１６９は、アプリケーションインスタンス１３４が、データを、不正ユーザに関連していることが判明しているサイトに対応する“ｗｗｗ．ｍａｌｉｃｉｏｕｓｕｓｅｒｓ．ｓｉｔｅ，”に送信しようとしていることを検出することができる。アプリケーションインスタンス１３４を修復するために、セキュリティ検証クライアントサービス１６９は、アプリケーションインスタンス１３４のドメイン名前解決を変更して“ＮＸＤＯＭＡＩＮ”とすることにより、ドメイン名前解決が不正サイトの実際のネットワークアドレスで行なわれるのを防止することができる。別の構成として、クライアント装置１００のファイアウォールルールは、ドメイン名前に関連するネットワークホストとの通信をブロックするように設定することができる。攻撃を行なうネットワークホストとの通信をブロックすることにより、アプリケーションインスタンス１３４のエンドユーザ機能に影響を与える場合もあるし、または影響を与えない場合もあるが、セキュリティリスクは無くすことができる。

【００３６】

アプリケーションインスタンス１３４が、動的に読み込まれるコード１４３にアクセスしようとしているかどうかの判断は、幾つかの方法で行なうことができる。例えば、アプリケーションマーケットプレイスシステム１２５は、リモートデータを取得するアプリケーション群１３１が、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを、アプリケーションマーケットプレイスシステム１２５から提供される条件として使用するように要求することができる。この条件を強制的に適用するために、セキュリティ検証サービス１２８は、アプリケーション１３１が、外部ネットワークサイト１４７に、特定のＡＰＩコールを使用することなくコンタクトするように設定されているかどうかを検出し、そのように設定されている場合、アプリケーション１３１の承認を拒否する。これらのＡＰＩコールは、単なるデータを取得する処理を、動的に読み込まれるコード１４３を含むデータを取得する処理から識別することができる。アプリケーションインスタンス１３４が実行時に、コードのＡＰＩコールを行なう場合、セキュリティ検証クライアントサービス１６９及び／またはセキュリティ検証サービス１２８は、次にセキュリティ分析を、動的に読み込まれて取得されるコード１４３に対して行なうように設定することができる。

【００３７】

更に、セキュリティ検証サービス１２８及び／またはセキュリティ検証クライアントサービス１６９は、データが、データメモリ１４０ではなく、実行可能コードメモリ１３７に読み込まれる時点を検出するように設定することができる。例えば、セキュリティ検証サービス１２８及び／またはセキュリティ検証クライアントサービス１６９は、アプリケーション１３１によりネットワーク１１９を経由して、サンドボックス環境１２９を使用することによりダウンロードされるデータを追跡して当該データにラベリングすることができる。データメモリ１４０に読み込まれるデータが禁止されて非実行可能データとなるのに対し、実行可能コードメモリ１３７に読み込まれるデータは実行することができる。検出は、サンドボックス環境１２９を使用することにより行なうことができ、サンドボックス環境１２９は、ダウンロードされるデータの読み込み先がいずれのメモリ領域であるかを追跡するように構成することができる。データが実行可能コードメモリ１３７にアプリケーションインスタンス１３４により読み込まれると、セキュリティ検証サービス１２８及び／またはセキュリティ検証クライアントサービス１６９は、セキュリティ分析を当該データに対して行なうことができる。１つの実施形態では、アプリケーション１３１は、サンドボックス環境１２９によって、実行可能コードをメモリに、特定のＡＰＩコール以外で読み込むことが禁止される。

【００３８】

幾つかの場合では、コードは、ダウンロードされたデータを分析することにより検出することができる。しかしながら、殆どの方法はアーキテクチャ固有である。例えば、特定の分析は、ｘ８６コード、ＡＲＭコード、ハイパーテキストマークアップ言語（ＨＴＭＬ）５コードなどを検出するために行なうことができる。

【００３９】

別の実施形態では、クライアント装置１００のサンドボックス環境１２９は、全ての外部データ要求をルーティングするように構成することができる、またはセキュリティ検証サービス１２８から提供されるプロキシサービスでアプリケーションインスタンス１３４により行なわれる要求であって、動的に読み込まれるコード１４３に対する要求として特定される外部データ要求をルーティングするように構成することができる。従って、セキュリティ分析は、セキュリティ検証サービス１２８により、クライアント装置１００内で行なうのではなく、コンピューティング環境１１３内のサーバ側に対して行なうことができる。これらのデータ要求をプロキシすることにより、種々の利点をもたらすことができ、これらの利点として、動的に読み込まれるコード１４３を動的に読み込まれるコードデータ１５３にキャッシュすることができること、及びダウンロード速度を高速化することができることを挙げることができる。

【００４０】

セキュリティ分析の結果から、セキュリティ検証サービス１２８及び／またはセキュリティ検証クライアントサービス１６９により開始される操作を指示することができる。幾つかの場合では、検出されるセキュリティリスクの重要度は変わる可能性がある。リスクが低い場合、アプリケーションマーケットプレイスシステム１２５は、当該アプリケーションを完全に除去するのではなく、単にアプリケーション１３１の提供アプリケーション１６３にフラグを立てるだけである。幾つかの場合では、アプリケーション１３１の優先レベルは、アプリケーション１３１に対して行なわれるセキュリティ分析のセキュリティレベル、及び／またはアプリケーション１３１に検出されるセキュリティリスクレベルに基づいて指定することができる。優先レベルにより、いずれの優先度をクライアント装置１００のアプリケーションインスタンス１３４に付与するかについて設定することができる。

【００４１】

幾つかの場合では、アプリケーションマーケットプレイスシステム１２５の所有者は、特定のアプリケーション群１３１またはアプリケーション群１３１のベンダー群をホワイトリストに登録して、セキュリティ分析を行なわないで済ませるように選択を行なうことができる。このようなアプリケーション群１３１またはアプリケーション群１３１のベンダー群は、所有者に信頼されて、追加のセキュリティ分析が、アプリケーションマーケットプレイスシステム１２５の支援を受けて、不必要であると見なされるようになる。このようなアプリケーション群１３１には、認証の署名を付与することができ、そして署名を検証してソースを確認することができる。

【００４２】

次に、図３を参照するに、図示されているのは、種々の実施形態によるアプリケーションマーケットプレイスシステム１２５の一部の動作の一実施例を提供するフローチャートである。図３のフローチャートは、多くの異なる種類の機能配置の一実施例を示しているに過ぎず、これらの機能配置を行なって、本明細書において記載されるアプリケーションマーケットプレイスシステム１２５の一部の動作を実行することができることを理解されたい。別の構成として、図３のフローチャートは、１つ以上の実施形態によるコンピューティング環境１１３（図２）において実行される方法のステップ群の一実施例を図示しているものとして見ることができる。

【００４３】

ボックス３０３から始まって、アプリケーションマーケットプレイスシステム１２５は、アプリケーション１３１（図２）を受信する。アプリケーション１３１は、開発者またはユーザによりアップロードすることができる。別の構成として、アプリケーション１３１は、データストア１２２（図２）に、アプリケーションマーケットプレイスシステム１２５の所有者によってダウンロードすることができる。ボックス３０６では、アプリケーションマーケットプレイスシステム１２５は、アプリケーション１３１の種類、アプリケーション１３１がネイティブ／スタンドアローンアプリケーション１３１または動的アプリケーション１３１であるかどうかを確認する。例えば、アプリケーションマーケットプレイスシステム１２５は、セキュリティ検証サービス１２８（図２）を用いて、アプリケーション１３１が、動的に読み込まれるコード１４３（図２）のダウンロードに関連するＡＰＩコールを行なうかどうかを確認することができる。

【００４４】

ボックス３０９では、アプリケーションマーケットプレイスシステム１２５は、アプリケーション１３１が動的アプリケーション１３１であるかどうかを確認する。動的アプリケーション１３１が、開発者により行なわれるような、自己宣言アプリケーションとすることができる、またはアプリケーション１３１は、クライアント装置１００からのアプリケーション実行時の挙動の報告により、ＡＰＩ分析により、アプリケーションをサンドボックス環境１２９（図２）内で実行することにより、または別のアプローチにより動的アプリケーションであることを確認することができる。アプリケーション１３１が動的アプリケーション１３１である場合、アプリケーションマーケットプレイスシステム１２５は

、動作を継続してボックス３１２に進み、そして動的アプリケーションセキュリティ分析アプローチを用いる。

【００４５】

このアプローチでは、アプリケーション１３１がアプリケーションマーケットプレイスシステム１２５に、当該アプリケーションの挙動がコード更新とともに変化している可能性があるときに受け入れられた後に、検証手順を継続することができる。動的アプリケーションセキュリティ分析アプローチでは更に、動的アプリケーション群１３１が通常、セキュリティリスクについて分析される可能性があるコンテナのような所定のネイティブコードを含んでいるので、セキュリティ分析を、ネイティブアプリケーションセキュリティ分析アプローチにより行なうことができる。その後、アプリケーションマーケットプレイスシステム１２５は動作を継続してボックス３１５に進む。アプリケーション１３１が動的アプリケーション１３１ではない場合、アプリケーションマーケットプレイスシステム１２５は、動作を継続してボックス３０９からボックス３１８に進み、そしてネイティブアプリケーションセキュリティ分析アプローチを用いる。次に、アプリケーションマーケットプレイスシステム１２５は、動作を継続してボックス３１５に進む。

【００４６】

ボックス３１５では、アプリケーションマーケットプレイスシステム１２５は、アプリケーションの提供アプリケーション１６３（図２）を、セキュリティ分析結果に少なくとも部分的に基づいて、生成する、修正する、または除去する。例えば、セキュリティリスクを持たないことが検証されたアプリケーション１３１は、アプリケーションマーケットプレイスに追加することができる。別の構成として、セキュリティリスクを含んでいることが確認されるアプリケーション１３１は、修正可能な場合に、修正することによりセキュリティリスクを除去することができる、またはアプリケーションマーケットプレイスから完全に除去することができる。幾つかの場合では、提供アプリケーション１６３は、セキュリティ問題が生じている可能性があることを示すフラグに関連付けることができる。１つの実施形態では、提供アプリケーション１６３にフラグを立てて、当該提供アプリケーションが、更に別の検証手順を受ける動的アプリケーション１３１であることを通知することができる。このようなフラグは、テキスト警告、アイコン、及び／または他の証印を含むことができる。提供アプリケーション１６３は、アプリケーション１３１に許可される優先レベルを、既に行なわれているセキュリティ検証のレベルに少なくとも部分的に基づいて通知することができる。その後、アプリケーションマーケットプレイスシステム１２５の一部が終了する。

【００４７】

図４に移って図４を参照するに、図示されているのは、種々の実施形態によるセキュリティ検証サービス１２８の一部の動作の一実施例を提供するフローチャートである。図４のフローチャートは、多くの異なる種類の機能配置の一実施例を示しているに過ぎず、これらの機能配置を用いて、本明細書において記載されるセキュリティ検証サービス１２８の一部の動作を実行することができることを理解されたい。別の構成として、図４のフローチャートは、１つ以上の実施形態によるコンピューティング環境１１３（図２）において実行される方法のステップ群の一実施例を図示しているものとして見ることができる。図４に示すタスク群の幾つかのタスク、または全てのタスクは、セキュリティ検証クライアントサービス１６９（図２）により、クライアント装置１００（図１）内で、代わりに実行するか、または追加で実行することができる。

【００４８】

ボックス４０３から始まって、セキュリティ検証サービス１２８は、アプリケーション１３１（図２）をアプリケーションインスタンス１３４（図２）として実行する。ボックス４０６では、セキュリティ検証サービス１２８は、アプリケーション１３１が、動的に読み込まれるコード１４３（図２）に、実行時にアクセスしていることを確認する。例えば、アプリケーション１３１は、動的コード使用に対応するＡＰＩコールを行なうことができる、アプリケーション１３１は、ダウンロードデータを実行可能コードメモリ１３７

に読み込むことができる、アプリケーション１３１は、認識可能な実行可能コードを含むデータをダウンロードすることができるなどである。

【００４９】

ボックス４０９では、セキュリティ検証サービス１２８は、外部ネットワークサイト１４７（図２）から要求されて動的に読み込まれるコード１４３を取得する。１つの実施形態では、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３に対する要求を、クライアント装置１００に代わってプロキシすることができる。別の実施形態では、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３に対する外部ネットワークサイト１４７からの当該サービス固有の要求を開始することができる。更に別の実施形態では、セキュリティ検証クライアントサービス１６９は実際に、動的に読み込まれるコード１４３を外部ネットワークサイト１４７から取得することができ、しかも次に、バージョン識別子をセキュリティ検証サービス１２８に送信することができる。

【００５０】

ボックス４１２では、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３のバージョンを確認する。例えば、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３に関連する指紋、コード署名、チェックサムなどを確認することができる。ボックス４１５では、セキュリティ検証サービス１２８は、前のセキュリティ分析１５７（図２）が、動的に読み込まれるコード１４３のバージョンについて行なわれているかどうかを確認する。例えば、セキュリティ検証サービス１２８は、動的に読み込まれるコード１４３の指紋を、前に分析されたコードに関連する動的に読み込まれるコードデータ１５３（図２）に含まれる指紋ライブラリと比較することができる。前のセキュリティ分析１５７が行なわれている場合、セキュリティ検証サービス１２８は、ボックス４１５からボックス４１８に移動して、前のセキュリティ分析１５７の結果を、データストア１２２（図２）から取得する。次に、セキュリティ検証サービス１２８は、動作を継続してボックス４２１に進む。前のセキュリティ分析１５７が特定のバージョンについて行なわれていない場合、セキュリティ検証サービス１２８は、ボックス４１５からボックス４１８に移動するのではなく、ボックス４１５からボックス４２４に移動して、セキュリティ分析を、動的に読み込まれるコード１４３に対して行なう。幾つかの実施形態では、セキュリティ検証サービス１２８は、セキュリティ分析の結果をクライアント装置１００から受信することができる。セキュリティ検証サービス１２８は、動作を継続してボックス４２１に進む。

【００５１】

ボックス４２１では、セキュリティ検証サービス１２８は、セキュリティリスクが、動的に読み込まれる所定バージョンのコード１４３について検出されるかどうかを確認する。セキュリティリスクが検出されない場合、アプリケーション１３１は問題がないことが検証されて、アプリケーション１３１を実行し続けることができる。その後、セキュリティ検証サービス１２８は終了する。セキュリティリスクが検出される場合、セキュリティ検証サービス１２８は、ボックス４２１からボックス４２７に進んで、１つ以上の操作をセキュリティリスクが検出されると開始する。セキュリティ分析の結果は、アプリケーションマーケットプレイスに関連するエンティティに送信することができる。例えば、セキュリティ検証サービス１２８は、アプリケーションマーケットプレイスのアプリケーション１３１の提供アプリケーション１６３（図２）を修正する、提供アプリケーション１６３にフラグを立てる、または提供アプリケーション１６３を除去することができる。

【００５２】

セキュリティ検証サービス１２８は、アプリケーション１３１を修復または修正して、セキュリティリスクを無くすことができる。セキュリティ検証サービス１２８は、クライアント装置１００に指示して、アプリケーション１３１の実行を終了させる、かつ／またはアプリケーション１３１をアンインストールさせることができる。幾つかの場合では、セキュリティ検証サービス１２８は、サンドボックス環境１２９の構成を変更して、セキュリティリスクを無くすことができる。例えば、ネットワークフィルタリングルールを更

新して、サンドボックス環境１２９の不正ネットワークサイトへのトラフィックをブロックすることができる、またはアプリケーション１３１にサンドボックス環境１２９を介してアクセス可能なコンタクトを、ダミーコンタクトに置き換えることができる。その後、セキュリティ検証サービス１２８の一部が終了する。

【００５３】

図５を参照するに、図示されているのは、本開示の１つの実施形態によるコンピューティング環境１１３の模式ブロック図である。コンピューティング環境１１３は、１つ以上のコンピューティング装置５００を含む。各コンピューティング装置５００は、少なくとも１つのプロセッサ回路を含み、当該プロセッサ回路は、例えばプロセッサ５０３と、メモリ５０６と、を有し、プロセッサ５０３及びメモリ５０６は共に、ローカルインターフェース５０９に接続される。この構成を実現するために、各コンピューティング装置５００は、例えば少なくとも１つのサーバコンピュータまたは同様の装置を備えることができる。ローカルインターフェース５０９は、図から分かるように、例えばアドレス／制御バスまたは他のバス構造が付随したデータバスを備えることができる。

【００５４】

メモリ５０６に格納されているのは、プロセッサ５０３で実行可能なデータ及び幾つかのコンポーネントの両方である。詳細には、メモリ５０６に格納され、かつプロセッサ５０３で実行することができるのは、セキュリティ検証サービス１２８、アプリケーションマーケットプレイスシステム１２５、サンドボックス環境１２９、及び場合によっては、他のアプリケーションである。更に、メモリ５０６に格納されているのは、データストア１２２及び他のデータとすることができる。更に、オペレーティングシステムをメモリ５０６に格納して、プロセッサ５０３で実行することができる。

【００５５】

図６を参照するに、図示されているのは、本開示の１つの実施形態によるクライアント装置１００の模式ブロック図である。クライアント装置１００は、少なくとも１つのプロセッサ回路を含み、当該プロセッサ回路は、例えばプロセッサ６０３と、メモリ６０６と、を有し、プロセッサ６０３及びメモリ６０６は共に、ローカルインターフェース６０９に接続される。ローカルインターフェース６０９は、図から分かるように、例えばアドレス／制御バスまたは他のバス構造が付随したデータバスを備えることができる。ディスプレイ１０４が更に、ローカルインターフェース６０９に接続される。

【００５６】

メモリ６０６に格納されているのは、プロセッサ６０３で実行可能なデータ及び幾つかのコンポーネントの両方である。詳細には、メモリ６０６に格納され、かつプロセッサ６０３で実行することができるのは、マーケットプレイスクライアントアプリケーション１６６、サンドボックス環境１２９、アプリケーションインスタンス１３４、セキュリティ検証クライアントサービス１６９、及び場合によっては、他のアプリケーションである。更に、メモリ６０６に格納されているのは、データストア及び他のデータとすることができる。更に、オペレーティングシステムをメモリ６０６に格納して、プロセッサ６０３で実行することができる。

【００５７】

次に、図５及び図６の両方を参照するに、図から分かるように、メモリ５０６、６０６に格納され、かつそれぞれのプロセッサ５０３、６０３で実行することができる他のアプリケーションを設けることができることを理解されたい。本明細書において説明されるいずれのコンポーネントも、ソフトウェア構成で実現される場合、複数のプログラミング言語のうちのいずれか１つのプログラミング言語を用いることができ、これらのプログラミング言語として、例えばＣ、Ｃ＋＋、Ｃ＃、Ｏｂｊｅｃｔｉｖｅ Ｃ、Ｊａｖａ（登録商標）、ＪａｖａＳｃｒｉｐｔ（登録商標）、Ｐｅｒｌ、ＰＨＰ、Ｖｉｓｕａｌ Ｂａｓｉｃ（登録商標）、Ｐｙｔｈｏｎ（登録商標）、Ｒｕｂｙ、Ｆｌａｓｈ（登録商標）、または他のプログラミング言語を挙げることができる。

【００５８】

複数のソフトウェアコンポーネントは、メモリ５０６、６０６に格納され、かつそれぞれのプロセッサ５０３、６０３で実行することができる。この点に関して、“ｅｘｅｃｕｔａｂｌｅ（実行可能な）”という用語は、最終的にプロセッサ５０３、６０３で実行することができる構成のプログラムファイルを意味している。実行可能プログラムの実施例として、例えばメモリ５０６、６０６のランダムアクセス部分に読み込むことができ、かつプロセッサ５０３、６０３で実行することができるフォーマットの機械コードに変換することができるコンパイルプログラム、メモリ５０６、６０６のランダムアクセス部分に読み込むことができ、かつプロセッサ５０３、６０３で実行することができるオブジェクトコードのような正しいフォーマットで表現することができるソースコード、または別の実行可能プログラムにより解釈されて命令群を、プロセッサ５０３、６０３で実行されることになるメモリ５０６、６０６のランダムアクセス部分に生成することができるソースコードなどを挙げることができる。実行可能プログラムは、メモリ５０６、６０６のいずれかの部分またはコンポーネントに格納することができ、これらのメモリとして、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）、ハードドライブ、固体ドライブ、ＵＳＢフラッシュドライブ、メモリカード、コンパクトディスク（ＣＤ）またはデジタル多用途ディスク（ＤＶＤ）のような光ディスク、フロッピー（登録商標）ディスク、磁気テープ、または他のメモリコンポーネントを挙げることができる。

【００５９】

メモリ５０６、６０６は、本明細書において、揮発性メモリ及び不揮発性メモリの両方のメモリ、及びデータストレージコンポーネントを含むものとして定義される。揮発性コンポーネントは、データ値を電源遮断時に保持しないコンポーネントである。不揮発性コンポーネントは、データを電源遮断時に保持するコンポーネントである。従って、メモリ５０６、６０６は、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）、ハードディスクドライブ、固体ドライブ、ＵＳＢフラッシュドライブ、メモリカードリーダでアクセス可能なメモリカード、接続先のフロッピーディスクドライブでアクセス可能なフロッピーディスク、光ディスクドライブでアクセス可能な光ディスク、適切なテープドライブでアクセス可能な磁気テープ、及び／または他のメモリコンポーネント、またはこれらのメモリコンポーネントのうちのいずれか２つ以上のメモリコンポーネントの組み合わせを含むことができる。更に、ＲＡＭは、例えばスタティックランダムアクセスメモリ（ＳＲＡＭ）、ダイナミックランダムアクセスメモリ（ＤＲＡＭ）、または磁気ランダムアクセスメモリ（ＭＲＡＭ）、及び他のこのようなメモリ装置を含むことができる。ＲＯＭは、例えばプログラマブルリードオンリメモリ（ＰＲＯＭ）、消去可能なプログラマブルリードオンリメモリ（ＥＰＲＯＭ）、電気的消去可能なプログラマブルリードオンリメモリ（ＥＥＰＲＯＭ）、または他の同様なメモリ装置を含むことができる。

【００６０】

また、プロセッサ５０３、６０３はそれぞれ、複数のプロセッサ５０３、６０３及び／または複数のプロセッサコアを表わすことができ、メモリ５０６、６０６はそれぞれ、処理回路群をそれぞれ同時に動作させる複数のメモリ５０６、６０６を表わすことができる。このような場合、ローカルインターフェース５０９、６０９は、複数のプロセッサ５０３、６０３のいずれか２つのプロセッサの間、いずれかのプロセッサ５０３、６０３とメモリ５０６、６０６のいずれかのメモリとの間、またはメモリ５０６、６０６のいずれか２つのメモリの間などの通信を容易にする適切なネットワークとすることができる。ローカルインターフェース５０９、６０９は、この通信を調整するように設計される更に別のシステムを備えることができ、これらのシステムは、例えば負荷バランスを実行する。プロセッサ５０３、６０３は、電気的構成とする、または他の所定の利用可能な構成とすることができる。

【００６１】

本明細書において記載されるセキュリティ検証サービス１２８、アプリケーションマーケットプレイスシステム１２５、サンドボックス環境１２９、マーケットプレイスクライアントアプリケーション１６６、アプリケーションインスタンス１３４、セキュリティ検

証クライアントサービス１６９、及び他の種々のシステムは、ソフトウェアで具体化することができる、または上に説明した汎用ハードウェアで実行されるコードとすることができるが、別の構成として、これらの同じ構成要素は、専用ハードウェアで、またはソフトウェア／汎用ハードウェア及び専用ハードウェアの組み合わせで具体化することもできる。専用ハードウェアで具体化される場合、各構成要素は、複数の技術のいずれか１つの技術、または複数の技術の組み合わせを用いる回路またはステートマシンとして実現することができる。これらの技術は、これらには限定されないが、種々の論理機能を１つ以上のデータ信号が印加されると実行する論理ゲートを有する個別論理回路、適切な論理ゲートを有する特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、または他のコンポーネントなどを含むことができる。このような技術は、この技術分野の当業者に広く知られているので、ここで詳細に説明することはしない。

【００６２】

図３及び図４のフローチャートは、アプリケーションマーケットプレイスシステム１２５及びセキュリティ検証サービス１２８の種々の構成部分の実施形態の機能及び動作を示している。ソフトウェアで具体化される場合、各ブロックは、モジュール、セグメントを表わすか、または指定される論理機能（群）を実行するプログラム命令群を含むコード部分を表わすことができる。プログラム命令群は、プログラミング言語で記述されるヒューマン可読命令文を含むソースコードの形式、またはコンピュータシステムまたは他のシステムのプロセッサ５０３、６０３のような適切な実行システムが認識可能な数値命令を含む機械コードの形式で具体化することができる。機械コードは、ソースコードなどから変換することができる。ハードウェアで具体化される場合、各ブロックは、指定される論理機能（群）を実行する回路または複数の相互接続回路を表わすことができる。

【００６３】

図３及び図４のフローチャートは、特定の実行順序を示しているが、この実行順序は、図示の実行順序と異なっていてもよいことを理解されたい。例えば、２つ以上のブロックの実行順序は、図示の順序をごちゃ混ぜにしたものとしてもよい。また、図３及び図４に連続して図示される２つ以上のブロックは、同時に実行してもよい、または一部を同時に実行してもよい。更に、幾つかの実施形態では、図３及び図４に示すブロック群のうちの１つ以上のブロックは、飛ばしてもよい、または省略してもよい。更に、任意の数のカウンタ、状態変数、警告セマフォ、またはメッセージを、本明細書において記載される論理フローに追加して、可用性改善、会計処理、性能測定、またはトラブルシューティング支援などを行なう。このような変更は、本開示の範囲に含まれることを理解されたい。

【００６４】

また、セキュリティ検証サービス１２８、アプリケーションマーケットプレイスシステム１２５、サンドボックス環境１２９、マーケットプレイスクライアントアプリケーション１６６、アプリケーションインスタンス１３４、及びセキュリティ検証クライアントサービス１６９を含み、かつソフトウェアまたはコードを含む本明細書に記載の任意のロジックまたはアプリケーションは、例えばコンピュータシステムまたは他のシステムのプロセッサ５０３、６０３のような命令実行システムにより使用される、または命令実行システムに接続される任意の非一時的なコンピュータ可読媒体で具体化することができる。この意味において、ロジックは、例えばコンピュータ可読媒体からフェッチすることができ、かつ命令実行システムにより実行することができる命令及び宣言を含む命令文を含むことができる。本開示の状況では、“ｃｏｍｐｕｔｅｒ－ｒｅａｄａｂｌｅ　ｍｅｄｉｕｍ（コンピュータ可読媒体）”とは、命令実行システムにより使用される、または命令実行システムに接続され、かつ本明細書において記載されるロジックまたはアプリケーションを収容する、格納する、または保持することができる任意の媒体とすることができる。

【００６５】

コンピュータ可読媒体は、例えば磁気媒体、光媒体、または半導体媒体のような多くの物理媒体のうちのいずれか１つの媒体を含むことができる。適切なコンピュータ可読媒体の更に具体的な実施例として、これらには限定されないが、磁気テープ、磁気フロッピー

ディスケット、磁気ハードドライブ、メモリカード、固体ドライブ、ＵＳＢフラッシュドライブ、または光ディスクを挙げることができる。また、コンピュータ可読媒体は、例えばスタティックランダムアクセスメモリ（ＳＲＡＭ）、及びダイナミックランダムアクセスメモリ（ＤＲＡＭ）、または磁気ランダムアクセスメモリ（ＭＲＡＭ）を含むランダムアクセスメモリ（ＲＡＭ）とすることができる。更に、コンピュータ可読媒体は、リードオンリメモリ（ＲＯＭ）、プログラマブルリードオンリメモリ（ＰＲＯＭ）、消去可能なプログラマブルリードオンリメモリ（ＥＰＲＯＭ）、電気的消去可能なプログラマブルリードオンリメモリ（ＥＥＰＲＯＭ）、または他の種類のメモリ装置とすることができる。

【００６６】

本開示の種々の実施形態は、以下の条項に記載することができる：

条項１．

少なくとも１つのコンピューティング装置で実行することができるプログラムを具体化する非一時的なコンピュータ可読記憶媒体であって：

アプリケーションのインスタンスをサンドボックス環境内で実行するコードであって、前記アプリケーションが、アプリケーションマーケットプレイスから受信される、前記実行するコードと；

前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれるコードにアクセスしようとしているかどうかを確認するコードであって、前記確認するコードが：

前記アプリケーションの前記インスタンスが、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを行なって、実行可能コードをネットワークサイトから取得しようとしているかどうかを確認するコードであって、前記アプリケーションの前記インスタンスが、前記実行可能コードを前記特定のＡＰＩコール以外のコールで取得することが禁止される、前記確認するコード；

前記アプリケーションの前記インスタンスが、ダウンロードデータを前記サンドボックス環境内のメモリ実行可能領域に読み込んでいるかどうかを確認するコード；または

前記アプリケーションの前記インスタンスが、認識可能な実行可能コードを含むダウンロードデータを有しているかどうかについて確認するコードのうちの少なくとも１つのコードを含む、前記確認するコードと；

前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードにアクセスしようとしていることが確認されると、セキュリティ分析を動的に読み込まれる前記コードに対して行なうコードと；

前記セキュリティ分析の結果を、前記アプリケーションマーケットプレイスに関連するエンティティに送信するコードと；

前記セキュリティ分析で検出されるセキュリティリスクを無くす操作を開始するコードと、を備える、非一時的なコンピュータ可読記憶媒体。

条項２．

前記操作では、前記アプリケーションマーケットプレイスの前記アプリケーションの提供アプリケーションを修正する、条項１に記載の非一時的なコンピュータ可読記憶媒体。

条項３．

前記少なくとも１つのコンピューティング装置はクライアント装置に対応する、条項１または２に記載の非一時的なコンピュータ可読記憶媒体。

条項４．

前記セキュリティ分析を行なう前記コードは更に、動的に読み込まれる前記コードに対する前のセキュリティ分析の結果を前記アプリケーションマーケットプレイスから受信するコードを含む、条項１乃至３に記載の非一時的なコンピュータ可読記憶媒体。

条項５．

少なくとも１つのコンピューティング装置と；

前記少なくとも１つのコンピューティング装置内で実行することができるセキュリティ検証サービスと、を備え、前記セキュリティ検証サービスは：

アプリケーションのインスタンスを実行するロジックであって、前記アプリケーションが、アプリケーションマーケットプレイスの中から提供される、前記実行するロジックと；

前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれるコードにネットワークサイトからアクセスしようとしていることを確認するロジックと；

前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードにアクセスしようとしていることが確認されると、セキュリティ分析を動的に読み込まれる前記コードに対して行なうロジックと；

前記セキュリティ分析の結果を、前記アプリケーションマーケットプレイスに送信するロジックと、を含む、システム。

条項６．

前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードに前記ネットワークサイトからアクセスしようとしていることを確認する前記ロジックは更に：

前記アプリケーションの前記インスタンスが、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを行なっていることを確認するロジックを含み、前記アプリケーションの前記インスタンスは、動的に読み込まれる前記コードを前記特定のＡＰＩコール以外のコールで取得することが禁止される、条項５に記載のシステム。

条項７．

前記アプリケーションの前記インスタンスの実行時に、前記アプリケーションの前記インスタンスが、動的に読み込まれる前記コードに前記ネットワークサイトからアクセスしようとしていることを確認する前記ロジックは更に：

前記アプリケーションの前記インスタンスが前記ネットワークサイトからダウンロードするデータを追跡して、前記データが実行可能コードを含んでいることを確認するロジックを含む、条項５または６に記載のシステム。

条項８．

前記セキュリティ分析を行なう前記分析ロジックは：

前のセキュリティ分析が既に、動的に読み込まれる前記コードに対して行なわれているかどうかを確認し；そして

前記前のセキュリティ分析が既に行なわれている場合に、前記前のセキュリティ分析を、前記セキュリティ分析として利用するように構成される、条項５乃至７に記載のシステム。

条項９．

前記前のセキュリティ分析が既に行なわれているかどうかを確認する際に更に：

動的に読み込まれる前記コードの指紋を確認し；そして

前記指紋を、動的に読み込まれるコードに対する複数の前のセキュリティ分析に関連する指紋ライブラリと比較する、条項４に記載のシステム。

条項１０．

前記セキュリティ検証サービスは更に、セキュリティリスクが前記セキュリティ分析で特定されると、前記アプリケーションの前記インスタンスを終了させるロジック、または動的に読み込まれる前記コードが、前記アプリケーションの前記インスタンスによって読み込まれるのを防止するロジックを含む、条項５乃至９に記載のシステム。

条項１１．

前記セキュリティ検証サービスは更に、動的に読み込まれる前記コードを修正して、前記セキュリティ分析で特定されるセキュリティリスクを無くすロジックを含む、条項５乃至１０に記載のシステム。

条項１２．

前記セキュリティ検証サービスは更に、セキュリティリスクが前記セキュリティ分析で特定されない場合に、前記アプリケーションの前記インスタンスに許可して、動的に読み

込まれる前記コードを実行させるロジックを含む、条項５乃至１１に記載のシステム。
条項１３．
前記セキュリティ検証サービスは更に、前記アプリケーションマーケットプレイスの中から提供される前記アプリケーションの提供アプリケーションを無効化する、または前記提供アプリケーションにフラグを立てるロジックを含む、条項５乃至１２に記載のシステム。
条項１４．
前記アプリケーションの前記インスタンスは、移動体通信端末装置内で、前記実行するロジックによって実行される、条項５乃至１３に記載のシステム。
条項１５．
前記アプリケーションの前記インスタンスは、サーバ環境内で、前記実行するロジックによって実行され、前記サーバ環境は、前記アプリケーションマーケットプレイスに代わって動作する、条項５乃至１４に記載のシステム。
条項１６．
少なくとも１つのコンピューティング装置によって、アプリケーションのプロバイダから受信するアプリケーションのインスタンスをサンドボックス環境内で実行し；
前記少なくとも１つのコンピューティング装置によって、前記アプリケーションの前記インスタンスが、ネットワークサイトからダウンロードしたデータを有していることを確認し；
前記少なくとも１つのコンピューティング装置によって、前記ダウンロードデータが、実行可能コードを含んでいるかどうかを確認し、前記確認する際に：
前記アプリケーションの前記インスタンスが、前記ダウンロードデータの少なくとも一部を、前記サンドボックス環境内のメモリの実行可能領域に既に読み込んでいるかどうかを確認する；または
前記アプリケーションの前記インスタンスが、特定のアプリケーションプログラミングインターフェース（ＡＰＩ）コールを行なって、前記ダウンロードデータを既に取得しているかどうかを確認して、前記アプリケーションの前記インスタンスが、前記実行可能コードを前記特定のＡＰＩコール以外のコールで取得することを禁止されるようにし；
前記ダウンロードデータが前記実行可能データを含んでいることが確認されると、前記少なくとも１つのコンピューティング装置によって、セキュリティ分析を前記ダウンロードデータの少なくとも一部に対して行ない；そして
前記少なくとも１つのコンピューティング装置から、前記セキュリティ分析の結果を前記アプリケーションの前記プロバイダに送信する、方法。
条項１７．
更に：
前記少なくとも１つのコンピューティング装置によって、前記ダウンロードデータの少なくとも一部に関連するコード署名を確認し；そして
前記少なくとも１つのコンピューティング装置によって、前記コード署名に対応する前のセキュリティ分析の結果を受信する、条項１１に記載の方法。
条項１８．
前記アプリケーションの前記プロバイダはアプリケーションマーケットプレイスであり、前記アプリケーションマーケットプレイス及び前記ネットワークサイトは、異なるエンティティによって制御される、条項１６または１７に記載の方法。
条項１９．
更に：
前記セキュリティ分析が行なわれると、前記少なくとも１つのコンピューティング装置によって、前記ダウンロードデータの前記少なくとも一部を修正してセキュリティリスクを無くす；または
前記セキュリティ分析が行なわれると、前記少なくとも１つのコンピューティング装置によって、前記サンドボックス環境の構成を変更して、前記セキュリティリスクを無くす

、条項１６乃至１８に記載の方法。

条項２０．

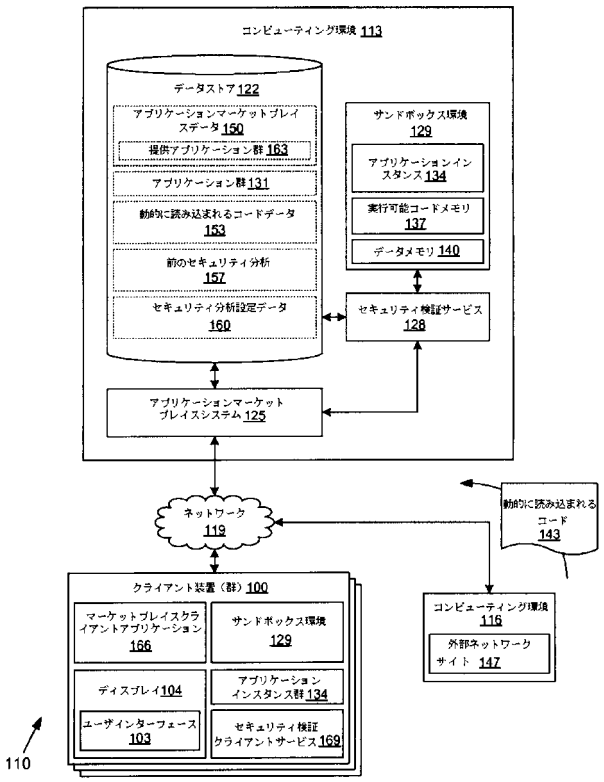
動的アプリケーション通知を前記アプリケーションの前記プロバイダから受信して、前記動的アプリケーション通知が前記アプリケーションに関連付けられると、前記アプリケーションの前記インスタンスを前記サンドボックス環境内で実行する、条項１６乃至１９に記載の方法。

【００６７】

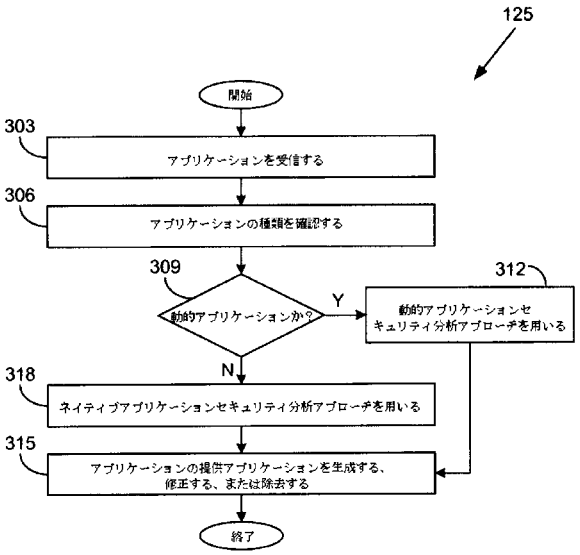
上に説明した本開示の種々の実施形態は、本開示の原理を明確に理解するために開示される種々の実施形態の適用可能な実施例に過ぎないことを強く認識されたい。多くの変更及び変形は、上に説明した実施形態（群）に対して、本開示の思想及び原理から大きく逸脱しない限り行なうことができる。全てのこのような変形及び変更は、本明細書においては、本開示の範囲に含まれ、かつ以下の請求項により保護されるべきである。

【図１】

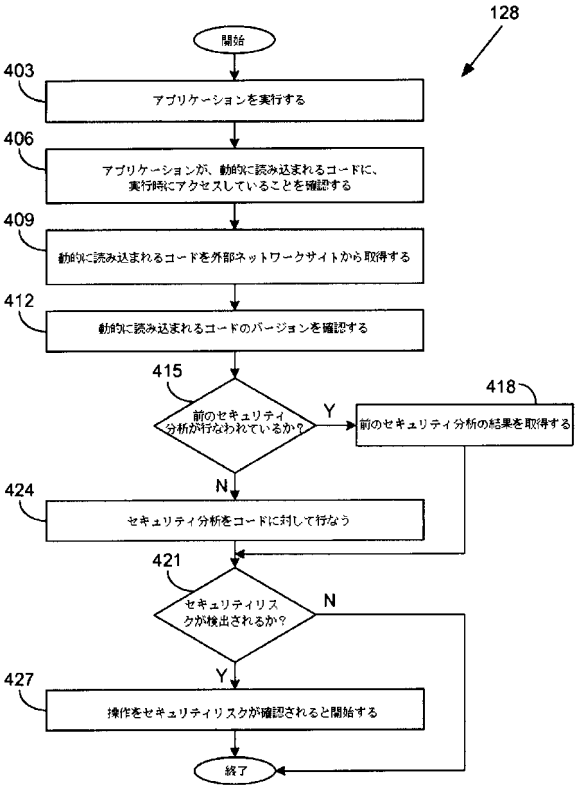
【図２】



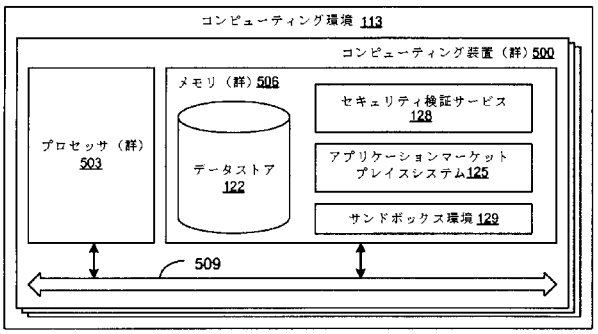
【図３】


125
開始
303
アプリケーションを受信する
306
アプリケーションの種類を確認する
309
動的アプリケーションか？
Y
312
動的アプリケーションセキュリティ分析アプローチを用いる
N
318
ネイティブアプリケーションセキュリティ分析アプローチを用いる
315
アプリケーションの提供アプリケーションを生成する、修正する、または除去する
終了

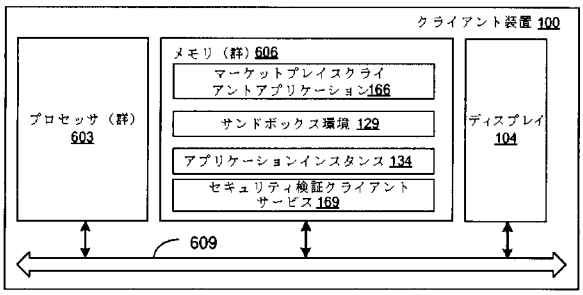
【図４】


128
開始
403
アプリケーションを実行する
406
アプリケーションが、動的に読み込まれるコードに、実行時にアクセスしていることを確認する
409
動的に読み込まれるコードを外部ネットワークサイトから取得する
412
動的に読み込まれるコードのバージョンを確認する
415
前のセキュリティ分析が行なわれているか？
Y
418
前のセキュリティ分析の結果を取得する
N
424
セキュリティ分析をコードに対して行なう
421
セキュリティリスクが検出されるか？
N
Y
427
操作をセキュリティリスクが確認されると開始する
終了

【図５】


コンピューティング環境 113
コンピューティング装置（群）500
プロセッサ（群）503
メモリ（群）506
データストア 122
セキュリティ検証サービス 128
アプリケーションマーケットプレイスシステム125
サンドボックス環境 129
509

【図６】


クライアント装置 100
プロセッサ（群）603
メモリ（群）606
マーケットプレイスクライアントアプリケーション166
サンドボックス環境 129
アプリケーションインスタンス 134
セキュリティ検証クライアントサービス 169
ディスプレイ 104
609

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/052932

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 29/06 (2014.01)
CPC - H04L 63/0245 (2014.09)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - H04L 29/06 (2014.01)
USPC - 726/1, 24-27, 29, 30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - H04L 63/0245; H04L 63/20; H04L 63/102 (2014.09) (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Orbit, Google Patents, Google

Search terms used: emulator, mobile application security, mobile software market, instrumented emulation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/0117651 A1 (EDERY et al.) 10 May 2012 (10.05.2012) entire document	1-15
Y	US 2012/0291022 A1 (MEHTA et al.) 15 November 2012 (15.11.2012) entire document	1-15
A	US 2010/0107252 A1 (MERTOGUNO) 29 April 2010 (29.04.2010) entire document	1-15

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 October 2014	18 DEC 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

フロントページの続き

(51)Int.Cl.	ＦＩ	テーマコード（参考）
	Ｇ０６Ｆ　21/56　　３８０	

(81)指定国　AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG

(72)発明者　ブランドワイン、　エリック、　ジェイソン
アメリカ合衆国　ワシントン州　９８１０９－５２１０　シアトル　テリー　アヴェニュー　ノース　４１０

Ｆターム(参考) 5B376 AB06　AE44　AE51　CA42　GA03