

(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2017-505944
(P2017-505944A)

(43) 公表日 平成29年2月23日(2017.2.23)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 21/56 (2013.01)	G06F 21/56 330	5B042
G06F 9/44 (2006.01)	G06F 9/06 620A	5B376
G06F 11/30 (2006.01)	G06F 11/30 140G	
G06F 11/34 (2006.01)	G06F 11/34 176	

審査請求 有 予備審査請求 未請求 (全 27 頁)

(21) 出願番号 特願2016-541324 (P2016-541324)
 (86) (22) 出願日 平成25年12月30日 (2013.12.30)
 (85) 翻訳文提出日 平成28年6月17日 (2016.6.17)
 (86) 国際出願番号 PCT/CN2013/090887
 (87) 国際公開番号 W02015/100538
 (87) 国際公開日 平成27年7月9日 (2015.7.9)

(71) 出願人 315002955
 ノキア テクノロジーズ オーユー
 フィンランド共和国 02610 エスポ
 ー カラボルッティ 3
 (74) 代理人 100127188
 弁理士 川守田 光紀
 (72) 発明者 ヤン テイ
 中華人民共和国 710071 陝西省
 西安市 西安電子科技大学 南太白通り
 No. 403 No. 2 ビルディング
 69
 Fターム(参考) 5B042 GA12 JJ06 JJ29 KK13 MA14
 MC12 MC25
 5B376 BC38 BC71 FA15 GA03

最終頁に続く

(54) 【発明の名称】 マルウェア検出検査方法及び装置

(57) 【要約】

オフラインマルウェア検出検査、及び追加的にリアルタイムマルウェア検出検査を行うアプローチが提供される。オフラインマルウェア検出検査は、アプリケーションの関数呼出しマップをオフラインで検出すること；関数呼出しマップはアプリケーションが呼び出す関数間の呼出し関係を記録すること；関数呼出しマップからアプリケーションの関数呼出しのパターンを抽出すること；抽出されたパターンを正常なアプリケーションの基本パターンと比較することを含んでもよい。リアルタイムマルウェア検出検査は：実環境でアプリケーションを実行すること；アプリケーションの実行中にアプリケーションの挙動を記録すること；記録された挙動から挙動パターンを抽出すること；抽出された挙動パターンを正常なアプリケーションの基本パターン又は以前に記録されたアプリケーションのパターンと比較することを含んでもよい。

【選択図】 図2

200

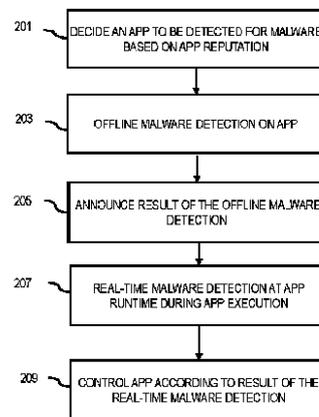


FIG. 2

【特許請求の範囲】**【請求項 1】**

アプリケーションに関するオフラインマルウェア検出検査を行うことを含む方法であって、前記オフラインマルウェア検出検査は：

前記アプリケーションの少なくとも 1 つの関数呼出しマップをオフラインで検出することであって、関数呼出しマップは前記アプリケーションが呼び出す関数間における呼出しの関係を記録する、前記検出することと；

前記少なくとも 1 つの関数呼出しマップから前記アプリケーションの関数呼出しのパターンを抽出することと；

前記抽出されたパターンを正常なアプリケーションの少なくとも 1 つの基本パターンと比較することと；

を含む、方法。

10

【請求項 2】

前記検出することは、仮想環境で前記アプリケーションのコードの少なくとも一部を実行することと、前記アプリケーションの関数呼出しのログを獲得することを含む、請求項 1 に記載の方法。

【請求項 3】

前記抽出することは、前記ログを解析するためにデータマイニング法を使用することを含む、請求項 2 に記載の方法。

【請求項 4】

20

前記少なくとも 1 つの関数呼出しマップは、次の 3 種類の関数呼出しマップ：

前記実行中に前記アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップ；

特定の異なる時点での呼出しマップであって、前記実行中に前記アプリケーションが前記異なる時点よりも前に呼び出した関数間の呼出し関係を含む、前記特定の異なる時点での呼出しマップ；

前記実行中に前記アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップ；

のうちの少なくとも 1 つを含む、請求項 3 に記載の方法。

【請求項 5】

30

前記オフラインマルウェア検出検査は、

前記関数呼出しマップのログを獲得する関数モジュールの追加によって、前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることと；

前記ログを獲得するために、前記仮想環境で前記再コンパイルされたコードを実行することと；

を更に含む、請求項 1 に記載の方法。

【請求項 6】

前記アプリケーションの評価に従って前記オフラインマルウェア検出検査をスケジュールすることを更に含む、請求項 1 に記載の方法。

【請求項 7】

40

高評価アプリケーションは、低評価アプリケーションよりも優先して前記オフラインマルウェア検出検査を受けるようにスケジュールされる、請求項 6 に記載の方法。

【請求項 8】

前記アプリケーションの潜在的不正の脅威を示す前記オフラインマルウェア検出検査の結果を告知することを更に含む、請求項 1 に記載の方法。

【請求項 9】

前記アプリケーションが実環境で実行される間に前記アプリケーションに関するリアルタイムマルウェア検出検査を行うことを更に含み、前記リアルタイムマルウェア検出検査は：

前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；

50

前記記録された挙動から挙動パターンを抽出することと；

前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；
を含む、請求項1から8の何れかに記載の方法。

【請求項10】

前記記録された挙動は、次の3種類の挙動；

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び/又は下りトラフィックに関連する挙動；

のうちの少なくとも1つを含む、請求項9に記載の方法。

10

【請求項11】

前記リアルタイムマルウェア検出検査は、

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含む、請求項9又は10に記載の方法。

【請求項12】

実環境でアプリケーションを実行することと；

前記アプリケーションの実行時における前記アプリケーションの挙動を記録することと；

前記記録された挙動から挙動パターンを抽出することと；

20

前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；
を含む、方法。

【請求項13】

前記記録された挙動は、次の3種類の挙動；

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び/又は下りトラフィックに関連する挙動；

のうちの少なくとも1つを含む、請求項12に記載の方法。

【請求項14】

30

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含む、請求項12又は13に記載の方法。

【請求項15】

少なくとも1つのプロセッサと、コンピュータプログラムコードを含む少なくとも1つのメモリを備える装置であって、

前記少なくとも1つのメモリ及び前記コンピュータプログラムコードは、前記少なくとも1つのプロセッサを用いて、前記装置に少なくとも、アプリケーションに関するオフラインマルウェア検出検査を実行させるように構成され、

前記オフラインマルウェア検出検査は；

40

前記アプリケーションの少なくとも1つの関数呼出しマップをオフラインで検出することであって、関数呼出しマップは前記アプリケーションが呼び出す関数間における呼出しの関係を記録する、前記検出することと；

前記少なくとも1つの関数呼出しマップから前記アプリケーションの関数呼出しのパターンを抽出することと；

前記抽出されたパターンを正常なアプリケーションの少なくとも1つの基本パターンと比較することと；

を含む、装置。

【請求項16】

前記検出することは、仮想環境で前記アプリケーションのコードの少なくとも一部を実

50

行することと、前記アプリケーションの関数呼出しのログを獲得することを含む、請求項 15 に記載の装置。

【請求項 17】

前記抽出することは、前記ログを解析するためにデータマイニング法を使用することを含む、請求項 16 に記載の装置。

【請求項 18】

前記少なくとも 1 つの関数呼出しマップは、次の 3 種類の関数呼出しマップ：

前記実行中に前記アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップ；

特定の異なる時点での呼出しマップであって、前記実行中に前記アプリケーションが前記異なる時点よりも前に呼び出した関数間の呼出し関係を含む、前記特定の異なる時点での呼出しマップ；

前記実行中に前記アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップ；

のうちの少なくとも 1 つを含む、請求項 17 に記載の装置。

【請求項 19】

前記オフラインマルウェア検出検査は、

前記関数呼出しマップのログを獲得する関数モジュールの追加によって、前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることと；

前記ログを獲得するために、前記仮想環境で前記再コンパイルされたコードを実行することと；

を更に含む、請求項 15 に記載の装置。

【請求項 20】

前記装置は更に、

前記アプリケーションの評価に従って前記オフラインマルウェア検出検査のスケジュールを組ませられる、請求項 15 に記載の装置。

【請求項 21】

高評価アプリケーションは、低評価アプリケーションよりも優先して前記オフラインマルウェア検出検査を受けるようにスケジュールされる、請求項 20 に記載の装置。

【請求項 22】

前記装置は更に、

前記アプリケーションの潜在的不正の脅威を示す前記オフラインマルウェア検出検査の結果を告知させられる、請求項 15 に記載の装置。

【請求項 23】

前記装置は更に、前記アプリケーションが実環境で実行される間に前記アプリケーションに関するリアルタイムマルウェア検出検査をさせられ、前記リアルタイムマルウェア検出検査は：

前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；

前記記録された挙動から挙動パターンを抽出することと；

前記抽出された挙動パターンを正常なアプリケーションの少なくとも 1 つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；

を含む、請求項 15 から 22 の何れかに記載の装置。

【請求項 24】

前記記録された挙動は、次の 3 種類の挙動：

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び / 又は下りトラフィックに関連する挙動

のうちの少なくとも 1 つを含む、請求項 23 に記載の装置。

【請求項 25】

前記リアルタイムマルウェア検出検査は、

15	行することと、前記アプリケーションの関数呼出しのログを獲得することを含む、請求項 15 に記載の装置。
	【請求項 17】
	前記抽出することは、前記ログを解析するためにデータマイニング法を使用することを含む、請求項 16 に記載の装置。
	【請求項 18】
	前記少なくとも 1 つの関数呼出しマップは、次の 3 種類の関数呼出しマップ：
	前記実行中に前記アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップ；
10	特定の異なる時点での呼出しマップであって、前記実行中に前記アプリケーションが前記異なる時点よりも前に呼び出した関数間の呼出し関係を含む、前記特定の異なる時点での呼出しマップ；
	前記実行中に前記アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップ；
	のうちの少なくとも 1 つを含む、請求項 17 に記載の装置。
	【請求項 19】
	前記オフラインマルウェア検出検査は、
	前記関数呼出しマップのログを獲得する関数モジュールの追加によって、前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることと；
20	前記ログを獲得するために、前記仮想環境で前記再コンパイルされたコードを実行することと；
	を更に含む、請求項 15 に記載の装置。
	【請求項 20】
	前記装置は更に、
	前記アプリケーションの評価に従って前記オフラインマルウェア検出検査のスケジュールを組ませられる、請求項 15 に記載の装置。
	【請求項 21】
	高評価アプリケーションは、低評価アプリケーションよりも優先して前記オフラインマルウェア検出検査を受けるようにスケジュールされる、請求項 20 に記載の装置。
30	【請求項 22】
	前記装置は更に、
	前記アプリケーションの潜在的不正の脅威を示す前記オフラインマルウェア検出検査の結果を告知させられる、請求項 15 に記載の装置。
	【請求項 23】
	前記装置は更に、前記アプリケーションが実環境で実行される間に前記アプリケーションに関するリアルタイムマルウェア検出検査をさせられ、前記リアルタイムマルウェア検出検査は：
	前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；
	前記記録された挙動から挙動パターンを抽出することと；
40	前記抽出された挙動パターンを正常なアプリケーションの少なくとも 1 つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；
	を含む、請求項 15 から 22 の何れかに記載の装置。
	【請求項 24】
	前記記録された挙動は、次の 3 種類の挙動：
	前記アプリケーションの関数呼出しに関連する挙動；
	前記アプリケーションが行ったローカルデータアクセスに関連する挙動；
	前記アプリケーションが起こした上り及び / 又は下りトラフィックに関連する挙動
	のうちの少なくとも 1 つを含む、請求項 23 に記載の装置。
	【請求項 25】
	前記リアルタイムマルウェア検出検査は、
50	

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含む、請求項 23 又は 24 に記載の装置。

【請求項 26】

少なくとも 1つのプロセッサと、コンピュータプログラムコードを含む少なくとも 1つのメモリを備える装置であって、前記少なくとも 1つのメモリおよび前記コンピュータプログラムコードは、前記少なくとも 1つのプロセッサを用いて、前記装置に少なくとも：
実環境でアプリケーションを実行することと；

前記アプリケーションの実行時における前記アプリケーションの挙動を記録することと；

10

前記記録された挙動から挙動パターンを抽出することと；

前記抽出された挙動パターンを正常なアプリケーションの少なくとも 1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；
を実行させるように構成される、装置。

【請求項 27】

前記記録された挙動は、次の 3種類の挙動：

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び / 又は下りトラフィックに関連する挙動；

のうちの少なくとも 1つを含む、請求項 26 に記載の装置。

20

【請求項 28】

前記装置は更に、

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルするように構成される、請求項 26 又は 27 に記載の装置。

【請求項 29】

1つ又は複数のプロセッサにより実行されると、装置に、請求項 1 から 14 の何れかに記載の方法を少なくとも実行させる、1つ又は複数の命令の 1つ又は複数のシーケンスを担持する、コンピュータ可読記憶媒体。

【請求項 30】

請求項 1 から 14 の何れかに記載の方法を実行する手段を備える、装置。

30

【請求項 31】

1つ又は複数のプロセッサにより実行されると、装置に、請求項 1 から 14 の何れかに記載の方法を少なくとも実行させる、1つ又は複数の 1つ又は複数のシーケンスを含む、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、概してコンピュータ及びネットワークのセキュリティに関し、より具体的には、マルウェア検出検査に関する。

40

【背景】

【0002】

モバイルデバイスは様々なアプリケーションを実行するためにオープンプラットフォームに発展している。アプリケーションは「アプリ」とも呼ばれるが、一般にモバイルデバイス等のコンピュータデバイスで実行可能なソフトウェアアプリケーションを指す。アプリ、特にモバイルアプリは、例えば、インターネットを通じて豊富な情報に即座にアクセスしたり様々な機能を提供したりすることによって、人々の日常活動の多くを向上させている。モバイルアプリの急速な成長は、将来のモバイルインターネットとその経済的な成功に対する決定的な役割を担っている。現在では百万種を超えるモバイルアプリがあり、

50

毎日2000種の新アプリが市場に出されている。

【0003】

しかし、モバイルアプリの中には不正やバグがあったり、不測の動作をしたり、モバイルデバイスにセキュリティホールを開けたりするものもある。加えて、マルウェアアプリは、モバイル・無線通信ネットワークだけでなくインターネットに接続する他のホストにとっても脅威となりうる。モバイルマルウェアは、モバイルデバイスにおいて悪意のある挙動を示す悪質なソフトウェアと見做されている。こうしたモバイルマルウェアの悪質な挙動は、大きくはモバイルデバイスに寄生するウイルスやボットネット、ワーム、トロイの木馬に分類される。これらは当初、モバイル通信システムのセキュリティ脆弱性を浮き彫りにしていた。しかし最近では、悪意あるモバイルコードがユーザの本人証明を盗んだり、ユーザの情報を売ったり、コンテンツ配信を操作したり、SMSスパムを送信したりする等によって、大規模な経済的利益を得る手段と化している。モバイルアプリが購入するユーザにとって信頼できるものであるということは、アプリとモバイルインターネットの最終的な成功に影響を与え、モバイルネットワーク関連の経済にも影響を及ぼす極めて重大な問題となっている。

10

【0004】

したがって、マルウェアを効果的かつ効率的に検出可能な方法及びシステムを提供することは、本技術分野において先進的なことである。

【例示的实施形態】

【0005】

前述の問題を解消し、先行技術から読み取って理解される限界を克服するために、オフラインと実行時の何れかまたは両方でマルウェアを効果的かつ効率的に検出するアプローチを開示してゆく。

20

【0006】

ある実施形態に従う方法は、アプリケーションに関するオフラインマルウェア検出検査を含む。オフラインマルウェア検出検査は、アプリケーションの少なくとも1つの関数呼出しマップをオフラインで検出することであって、関数呼出しマップは前記アプリケーションが呼び出す関数間における呼出しの関係を記録する、前記検出することと；前記少なくとも1つの関数呼出しマップから前記アプリケーションの関数呼出しのパターンを抽出することと；前記抽出されたパターンを正常なアプリケーションの少なくとも1つの基本パターンと比較することを含む。オフラインマルウェア検出検査の結果は、前記アプリケーションの潜在的な不正の脅威を示すように告知されてもよい。

30

【0007】

例示的实施形態によっては、前記検出することは、仮想環境で前記アプリケーションのコードの少なくとも一部を実行することと、前記アプリケーションの関数呼出しのログを獲得することを含んでもよい。前記抽出することは、前記ログを解析するためにデータマイニング法を使用することを含んでもよい。

【0008】

例示的实施形態によっては、前記少なくとも1つの関数呼出しマップは、次の3種類の関数呼出しマップ：前記実行中に前記アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップ；特定の異なる時点での呼出しマップであって、前記実行中に前記アプリケーションが前記異なる時点よりも前に呼び出した関数間の呼出し関係を含む、前記特定の異なる時点での呼出しマップ；及び前記実行中に前記アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップの少なくとも1つを含んでもよい。

40

【0009】

例示的实施形態によっては、前記オフラインマルウェア検出検査は、前記関数呼出しマップのログを獲得する関数モジュールの追加によって、前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることと、前記ログを獲得するために、前記仮想環境で前記再コンパイルされたコードを実行することを更に含んでもよい。

50

【0010】

例示的实施形態によっては、前記方法は、前記アプリケーションの評価に従って前記オフラインマルウェア検出検査をスケジュールすることを更に含んでもよい。高評価アプリケーションは、低評価アプリケーションよりも優先して前記オフラインマルウェア検出検査を受けるようにスケジュールされてもよい。

【0011】

例示的实施形態によっては、前記方法は、前記アプリケーションが実環境で実行される間に前記アプリケーションに関するリアルタイムマルウェア検出検査を行うことを更に含んでもよい。リアルタイムマルウェア検出検査は、前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；前記記録された挙動から挙動パターンを抽出することと；前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することを含む。前記記録された挙動は、次の3種類の挙動：前記アプリケーションの関数呼出しに関連する挙動；前記アプリケーションが行ったローカルデータアクセスに関連する挙動；及び前記アプリケーションが起こした上り及び/又は下りトラフィックに関連する挙動の少なくとも1つを含んでもよい。前記リアルタイムマルウェア検出検査は、前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含んでもよい。

10

【0012】

別の実施形態に従う方法は、実環境でアプリケーションを実行することと；前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；前記記録された挙動から挙動パターンを抽出することと；前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することを含む。

20

【0013】

別の実施形態に従う装置は、少なくとも1つのプロセッサと、コンピュータプログラムコードを含む少なくとも1つのメモリを備え、前記少なくとも1つのメモリ及び前記コンピュータプログラムコードは、前記少なくとも1つのプロセッサを用いて、前記装置に少なくとも部分的に、実環境でアプリケーションを実行することと；前記アプリケーションの実行時における前記アプリケーションの挙動を記録することと；前記記録された挙動から挙動パターンを抽出することと；前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較すること
を実行させるように構成される。

30

【0014】

別の実施形態に従うコンピュータ可読記憶媒体は、1つ又は複数のプロセッサにより実行されると、装置に、前述の方法の1つを少なくとも部分的に実行させる、1つ又は複数の命令の1つ又は複数のシーケンスを担持する。

【0015】

別の実施形態に従う装置は、前述の方法を実行する手段を備える。

40

【0016】

コンピュータプログラム製品は、1つ又は複数のプロセッサにより実行されると、装置に前述の方法の1つを少なくとも実行させる、1つ又は複数の1つ又は複数のシーケンスを含む。

【0017】

本発明のさらなる側面や特徴、利点が、以下の詳細説明によって容易に明らかになる。以下の詳細説明では、本発明を実施するための最良の形態であると考えられているものも含め、種々の具体的な実施形態や実装形態が例示される。本発明はまた、さらに多くの様々な異なる実施形態を取りうることができ、その幾つかの詳細部分は、本発明の思想や範囲を逸脱することなく、多くの自明な観点から修正可能なものである。本明細書による説

50

明や図面は例示的な性質を有するものと考えられるべきであり、制限的なものとみなされるべきではない。

【図面の簡単な説明】

【0018】

添付の図面には、本発明の実施形態が例示されている。これらはいくまでも例示を目的とするものであって、限定の目的のためのものではない。

【0019】

【図1】本発明の実施形態に従う、モバイルデバイスに対してマルウェアを検出するためのアーキテクチャを示す。

【0020】

【図2】本発明の実施形態に従う包括的マルウェア検出検査の手順を示すフローチャートである。

【0021】

【図3】本発明の実施形態に従う、評判評価によってオフラインマルウェア検出検査を推進する手順を示す。

【0022】

【図4】本発明の実施形態に従うオフラインマルウェア検出検査のフローチャートである。

【0023】

【図5】本発明の実施形態に従うリアルタイムマルウェア検出検査のフローチャートである。

【0024】

【図6】本発明の様々な例示の実施形態が適用されうる装置の例示的ブロック図を示す。

【詳細説明】

【0025】

マルウェア検出検査を行う方法、装置及びシステムの実施例を説明する。以降の記述では、説明目的上、多くの具体的かつ詳細な構成が示されているが、これらは本発明の実施形態の深い理解に繋げるためのものである。なお、当業者には明らかなことであるが、本発明の実施形態は、これら特定の詳細構成以外でも実施可能であり、均等な構成によって実施することもできる。また、既知の構成やデバイスがブロック図の形で示されるが、これは、本発明の実施形態をいたずらに不明瞭にすることを避けるためである。本明細書および図面を通じて同様の符号は同様の要素を表す。

【0026】

以降の記述では、マルウェアの脅威にさらされるコンピューティングデバイスの一例としてモバイルデバイスが用いられる。しかし、こうしたデバイスにはラップトップコンピュータやデスクトップコンピュータ、ホームオートメーションデバイス、ホーム制御デバイスの何れか又は全てのような他の種類のコンピューティングデバイスでも起こりうる。また、本明細書ではモバイルマルウェアが用いられる。しかしマルウェアとしては、限定的ではないがラップトップコンピュータのマルウェアや未知のマルウェアを含むその他のマルウェアもありうる。

【0027】

セキュリティを主たる懸念事項と捉えると、アプリが悪質か良質かを見極めるために各アプリを精査して全アプリの追跡を維持することは侮り難いタスクである。一方、多数のアプリストアでは、レコメンデーションとしてアプリの信頼性や評価を示すダウンロードスタットを使用している。しかし、ある調査によると、モバイルアプリの26%はダウンロード後、一度試されただけで棄てられていることが判明している。これから明らかであるように、ダウンロードスタットは不完全かつ誇張されて見えることが多いため、ダウンロード追跡は、アプリの成功のセキュリティを計るためには全く正確ではない。膨大な数のアプリがあるため、マルウェアを効果的かつ効率的に検出することは非常に困難である。第一に、膨大な数のアプリケーションによって、(例えば、F-secureや中国の360等の

10

20

30

40

50

）セキュリティサービスプロバイダが各アプリケーションのセキュリティを検証してどのアプリケーションが妥当であるべきかを優先して決定することは、困難かつコストの掛かることとなっている。第二に、アプリケーションによっては、インストールして暫く使用した後に悪質になるものも存在する。現在、実行時にモバイルマルウェアを動的に検出できるような状況ではない。

【0028】

多数のマルウェア検出技術が提案されてきたが、その大部分は包括的検査において取り上げられてきた。しかし、モバイルマルウェアに関する研究はまだ緒についたばかりである。マルウェア検出検査技術は以下のカテゴリに分類される。モバイルマルウェア及び他のセキュリティ脆弱性を検出するために利用可能な技術には様々な長所・短所がある。

10

【0029】

(1) 静的解析

【0030】

静的解析は、アプリケーションにおける悪質な特徴や不正なコードセグメントを、それらを実行せずに発見する方法である。これらは一般に、明白なセキュリティ上の脅威を検出するために疑わしいアプリケーションが最初に評価される際の予備的解析において用いられる。静的マルウェア検出検査技術の第1の種類は、最初にモバイルアプリケーション及び抽出システムコールを逆アセンブルする(特徴抽出)。次いで、セントロイドマシン(Centroid Machine)と呼ばれる軽量クラスタリング機構の一種を用いて、モバイルアプリケーションを悪質か良質かの何れかに分類する(異常検出)。静的マルウェア検出検査技術の第2の種類は、最初にモバイルアプリケーションを逆アセンブルして処理フローグラフ(CFG)を構成することによって静的テイント解析(static taint analysis)を実行する。この解析は、アドレス帳や現GPS座標、キーボードキャッシュ、固有デバイスID、その他電話機に関連する情報のような機密性の高いソースからの経路を考慮する。データフロー解析は、こうしたソースから伝送されるあらゆる機密データに対して、ユーザへ通知せずに同期してプライバシー漏洩を引き起こしていないかを調べる。この方法は単一アプリケーション内でのプライバシー漏洩のみを検査するため、2つ以上のアプリケーションが互いに他動的に連鎖されている場合には機能しない。静的マルウェア検出検査技術の第3の種類は、(例えば、Androidコード用の)逆コンパイラを用いてアプリケーションのインストールイメージからJava(登録商標)のソースコードを生成し、静的コード解析パッケージソフトを用いて復元したソースコードを評価する。この技術の適用は、比較的少数の権限とAPIコールを用いるアプリケーションに限定される。

20

30

【0031】

静的解析は高速かつ安価なアプローチであるが、長い間問題なく動作した後で自己書換えを行い、突然悪質な挙動を示すコードによって引き起こされるセキュリティ脅威を検出することは困難である。そして、モバイルボットネットマスタやボットネット、ウイルスによる攻撃や命令を克服することは不可能である。

【0032】

(2) 動的解析

【0033】

動的解析は、モバイルアプリケーションの動的挙動を調査員が監視できるように、仮想マシンやエミュレータ等の隔離環境でのアプリケーションの実行を伴う。調査員は主に、テイント追跡やシステムコール追跡で動的解析を用いる。例えば、TaintDroidはAndroid用でシステム全体に亘る動的テイント追跡を提供する。モバイルアプリケーションはDalvik仮想マシン(Dalvik virtual machine)に通され、テイント伝搬に関する4つの粒度である変数、メソッド、メッセージ、ファイルの各レベルを実行する。テイント追跡は、位置やマイクロフォン、カメラ、その他の電話機識別子等の機密ソースからのあらゆる不明瞭なデータをマークする。この技術は、全てのネイティブライブラリが仮想マシンから確実に呼び出されるようにネイティブライブラリロードを変更し、信用のないアプリケーションがネイティブメソッドを直接実行するのを防止できる。最終的に、動的解析は、シス

40

50

テムをネットワークインタフェース、即ちテイントシンクに残す前に、機密データ漏洩の恐れのある影響を受けたデータをスクリーニングする。しかし、TaintDroidはフォールスネガティブ及びフォールスポジティブを引き起こす可能性がある。さらに、TaintDroidは専らデータフローに焦点を当てており、その他の脆弱性については考慮していない。Android Application Sandbox (AASandbox) システムはAndroidアプリケーション用に二段階解析を提供する。モバイルアプリケーションはAASandboxに通され、オフラインモードで静的解析と動的解析を実行する。静的解析はアプリケーションイメージバイナリを逆アセンブルし、逆アセンブルコードを用いて不審パターンを探す (<http://bit.ly/171Mnl>)。動的解析はAndroidエミュレータでこのバイナリを実行し、システムコールのログを取る。調査員が入力を生成するためにAndroid Monkey (ADB Monkey) を使用したとしても、実際のユーザによるテストと比べて殆ど影響は無い。しかも、このアプローチは、多様な挙動を示すマルウェアやコードフラグメント暗号化に対してはテストされていない。

10

【0034】

この実質的数量には、より綿密な(手動の)解析に相応しいサンプルと様々な既知の脅威であるサンプルとの間を高速で区別する自動化されたアプローチが必要とされる。こうした自動解析は二通りの方法で行われる。動的解析は、サンプルを実行し、そのサンプルが実際に実行する動作を検証する技術を指し、静的解析は、実際にサンプルを実行せずにそのタスクのみを実行する。静的解析ではプログラムを通じて実行されるフロー全てをカバーできるのに対して、動的解析では経路のカバーが不完全であるという問題がある。

20

【0035】

(3) アプリケーション権限解析

【0036】

アプリケーション権限解析は、権限確認を通じてモバイルアプリの悪質な挙動を発見することを目的としている。例えば、KirinはAndroidプラットフォーム用認定アプリケーションである。Kirinは、アプリケーションをインストールする間に権限確認を実行する。ユーザがアプリケーションをインストールするとき、Kirinはアプリケーションのセキュリティ構成を抽出し、その構成がそのアプリケーションが既に有しているセキュリティポリシーに反していないかを調べる。アプリケーションが何れかのセキュリティポリシー規則に合致しなかった場合、Kirinはそのアプリケーションを削除するか、ユーザに警告することができる。Kirinは専らアプリケーション作成者の権限要求を確認するだけであって、そのアプリケーションがこうした権限をどのように使用するかについては精査しない。したがって、悪意をもって機密ユーザデータを開示したり、(許可された権限で)望まないコンテンツを導入したりする一部のアプリのセキュリティホールを調べるには効果が無い。

30

【0037】

(4) クラウドベース検出検査

【0038】

スマートフォンは、計算能力とエネルギー源の制約のため、簡易ファイルスキャン等の具備するセキュリティ機構を完全に実行することができないこともある。クラウドベースのマルウェア保護技術はセキュリティ解析と計算をリモートサーバに移し、このリモートサーバは、エミュレータ上で動作している複数の携帯電話のレプリカをホストする。スマートフォンに搭載されているトラッカーは、モバイルアプリケーションの実行を再現するのに必要な全ての情報を記録する。トラッカーは記録した情報をクラウドベースリプレイヤーに伝送し、クラウドベースリプレイヤーはエミュレータで実行を再現する。リプレイヤーは、クラウドの豊富なリソースから動的マルウェア解析やメモリスキャナ、システムコール異常検出検査、商用アンチウイルススキャン等の複数のセキュリティ検査を配備することができる。しかし、このアプローチを最初に用いても、サンプルサイズが依然として非常に小さいため、フォールスポジティブの結果を生じさせる可能性はある。また、ユーザがアプリケーションの挙動をサードパーティーに送信するように求められるときにユーザがどのように反応するかは不明確であり、ユーザの振舞いに全面的に依存したとして

40

50

も、正確な結果を得られるとは限らない。そしてユーザのプライバシー問題も未解決のままである。さらに、このアプローチは検出検査を後回しにもする。したがって、ネットワーク接続が利用できないかマルウェアによって損なわれた場合には適用することができない。

【 0 0 3 9 】

(5) バッテリ寿命監視

【 0 0 4 0 】

スマートフォンはバッテリー容量に制限があるため、時折電力消費を測定することによって悪質なアプリケーションを特定することができる。こうした悪質なアプリケーションは良質なものよりも多くの電力を消費するからである。通常ユーザ行動と現在のバッテリー状態、信号強度やネットワークトラフィック等、その他のドメイン固有の詳細情報が分かる場合、より精密に隠れた悪質な挙動を検出することができる。不測のユーザ行動とマルウェア侵入による偽装イベントがあると、電力モデルの確度に影響が及ぶ可能性がある。また、このアプローチはマルチタスク機能を有するスマートフォンには適用できない。

10

【 0 0 4 1 】

前述のように、こうしたアプローチは各タイプにそれぞれの短所があって、総合的によいものは一つも無い。また、モバイルマルウェア検出検査として現存するアプローチの殆どは、モバイルマルウェアが実行される時にリアルタイムで検出することはできない。しかも、悪質なモバイルアプリによっては、長い間使用された後に突然、又はコードの自己書換えを通じてモバイルデバイスに侵入するものもある。こうした脅威は、モバイルアプリの信頼性管理に関する研究に課題を与えている。

20

【 0 0 4 2 】

様々な例示的实施形態に従って、効率的かつ効果的なマルウェア検出検査がアプリケーションのオフライン検査において提供され、さらにアプリケーションの実行時におけるリアルタイム検査でも提供される。マルウェアの「オフライン」検査とは、アプリケーションを実環境で実際に実行せずにアプリケーション内の異常を検出することを意味する。実施形態によっては、アプリケーションの関数呼出しのログを獲得するために、検査すべきアプリケーションのコードの少なくとも一部が仮想環境で実行されてもよい。関数呼出しのログから、アプリケーションが呼び出す関数間での呼出し関係を反映する関数呼出しマップが得られる。アプリケーションの関数呼出しマップには、アプリケーションが呼び出した関数を表わすノードと、ノード間の呼出し関係を示すエッジが含まれる。エッジは、それに対応する呼出し関係の詳細情報であって、呼出し稠密度 (calling density) や呼出し頻度等に応じて更に重み付けすることもできる。関数呼出しマップをデータマイニング技術等で統計的解析することで、関数呼出しマップから統計的特徴又はパターンが抽出されてもよい。次いで、検査対象アプリケーションから異常を発見するために、抽出されたパターン又は特徴が、正常なアプリケーションの基本パターン又は規則と比較されてもよい。こうして、アプリケーションの関数呼出しの構造の解析に基づいて、オフラインマルウェア検出検査を自動的に構築することができる。オフラインマルウェア検出検査の結果は、例えばセキュリティ証明書を発行することによって告知されてもよい。セキュリティ証明書は検査対象アプリケーションによる潜在的不正の脅威を知らせるものであって、ユーザがそのアプリケーションのダウンロードを決断する手助けをするものである。

30

40

【 0 0 4 3 】

実施形態によっては、(オフライン検査済み又は未検査を問わず) アプリケーションが実環境のモバイルデバイスにダウンロード、インストール、そして実際に実行されたとしても、モバイルデバイスはアプリケーションの悪質な挙動を自動的に検出することができる。これに関して、そのパターンや統計的特徴を抽出するために、アプリケーションの実行時における実行挙動がリアルタイムで取り出されてもよい。異常を発見するために、抽出パターンは、そのアプリケーションの正常パターン及び予め記録済みのパターンと比較されてもよい。悪質なアプリケーションによっては、インストール後暫くは正常に動作するが、ユーザの信用を獲得した後、突然悪意をもつものもある。したがって、パターン比

50

較はこの種の潜在的脅威も効果的に検出することができる。これにより、マルウェアを検出する包括的方法が提供される。

【 0 0 4 4 】

実施形態によっては、オフラインマルウェア検出検査を自動的に推進するために、アプリケーション毎の評価が利用されてもよい。これに関する基本概念は、マルウェア検出検査がされていない最高評価のアプリケーションこそ、疑いを払拭するべく慎重に解析される必要があるということである。アプリケーションの評価は、そのアプリケーションに対するユーザの中での信頼度と需要を示している。例えば、アプリケーションの評価が高い程、そのアプリケーションはより信頼され需要もあることを意味する。実施形態によっては、例えばモバイルデバイスにおけるアプリケーションの使用挙動に応じて、アプリケーションに対するユーザ個別の信頼が自然と生じることもある。こうした個別ユーザの信頼と、ユーザのフィードバック等アプリケーションに関する他の情報に基づいて、そのアプリケーションの評価が更に作られてもよい。アプリケーションの評価に基づいて、高評価で需要のある（広く使用されている）未検査アプリケーションは、低評価であり使用されていないアプリケーションよりも優先してマルウェア検査が行われる。

図 1 は、実施形態に従う、モバイルデバイスに対してマルウェアを検出するためのアーキテクチャの機能ブロック図である。図 1 に示すように、システム 1 0 0 は (UE) は、通信ネットワーク 1 0 5 を介してコンピューティングデバイス 1 0 1 a、セキュリティサービスプロバイダ 1 0 9、評価センター 1 1 1、その他の通信エンティティ (他のコンピューティングデバイス 1 0 1 b) を備え、コンピューティングデバイス 1 0 1 a はアプリケーションストア 1 0 7 と接続している。例として、システム 1 0 0 の通信ネットワーク 1 0 5 には、データネットワークや無線ネットワーク、電話ネットワーク (何れも図示せず)、又はそれらの組合せのような 1 つ又は複数のネットワークが含まれる。データネットワークは、ローカルエリアネットワーク (LAN) やメトロポリタン エリア ネットワーク (MAN)、広域ネットワーク (WAN)、公衆データネットワーク (例えばインターネット)、自己管理モバイルネットワーク (self-organized mobile network) 等でもよく、他の適切なパケット交換ネットワークでもよい。パケット交換ネットワークには、商用利用可能なものもあれば、個別の光ケーブルや光ファイバネットワークのような、私有のパケット交換ネットワークもある。さらに無線ネットワークは、例えばセルラネットワークでもよく、EDGE (enhanced data rates for global evolution) や GPRS (general packet radio service), GSM (登録商標)、IMS (Internet protocol multimedia subsystem)、UMTS (universal mobile telecommunications system) など、様々な技術を利用したものでもよい。あるいは他の適切な無線媒体が利用されてもよく、こうした無線媒体として、例えば WiMAX (worldwide interoperability for microwave access) や LTE (Long Term Evolution)、CDMA (符号分割多元接続)、WCDMA (登録商標)、WiFi、衛星、MANNET (モバイルアドホックネットワーク) 等がある。

【 0 0 4 5 】

コンピューティングデバイス 1 0 1 a・1 0 1 b (以下、一般に 1 0 1 とする) は、プロセッサ等でソフトウェアアプリケーションを実行可能なあらゆるタイプのデバイスでもよい。例えば、コンピューティングデバイス 1 0 1 は、スマートフォンやタブレット、ラップトップコンピュータ、ノートブック、携帯情報端末 (PDA) のような携帯型 (モバイル) デバイス、ステーションやユニット、マルチメディアコンピュータ、マルチメディアタブレット、インターネットノード、デスクトップコンピュータのような据置型デバイス、又は組込型デバイスでもよく、こうしたデバイスの組合せでもよい。図 1 に示すように、コンピューティングデバイス 1 0 1 はアプリケーションストア 1 0 7 からアプリケーション 1 0 3 a・1 0 3 b をダウンロードし、ダウンロードしたアプリケーションを実行する。コンピューティングデバイス 1 0 1 は、アプリケーションの使用に関するフィードバックをアプリケーションストア 1 0 7 や評価センター 1 1 1 等の他の団体、他の機関の何れか又は全てに提供するために利用されてもよい。

【 0 0 4 6 】

10

20

30

40

50

アプリケーションストア107は、種々のアプリケーションのアップロード、ダウンロード、更新等のためにそれらをキャッシュして管理してもよい。例えばスマートフォン用として、Windows PhoneシステムやAndroidシステム、iOSシステムのようなオペレーションシステム毎に複数のアプリケーションストアが存在する。図1にはアプリケーションストアが1つしか示されていないが、アプリケーションストアの数は任意である。

【0047】

セキュリティサービスプロバイダ(SSP)109は、複数のアプリケーションをオフラインでスキャンしてアプリケーションの異常及びマルウェアを検出するために提供される。実施形態によっては、SSP109は、スキャンされるべきアプリケーションをアプリケーションストア107から直接又は間接でダウンロードしてもよい。しかし、SSP109は、ソフトウェアアプリケーションの開発者や起業、政府機関、ユーザ、その他のエンティティの何れか又は全てのようなアプリケーションのソースから、スキャンされるべきアプリケーションの実行コードを取得してもよい。スキャン又はマルウェア検出検査の結果は、アプリケーションをダウンロードするというユーザの決定を助けるために発行されてもよい。例えば、F-secureや360等、ソフトウェアアプリケーションのセキュリティサービスを提供する複数の企業又は機関が存在する。実施形態によっては、SSP109は、ソフトウェアアプリケーションのセキュリティを検査するような企業又は機関のサーバとして具現化されてもよく、他の関係者がアクセス可能な公的又は私的クラウドサービスとして配備されてもよい。実施形態によっては、SSP109は、こうしたアプリケーションをそれ自体が実際に実行可能なコンピューティングデバイスに配備することもできる。

10

20

【0048】

さらに、SSP109におけるオフラインマルウェア検出検査又はスキャンは、こうしたアプリケーションの評価順位に基づいて行われてもよい。膨大な数のアプリケーションがある場合、SSP109が各々のアプリケーションのセキュリティを検証するのは困難でコストも掛かる可能性がある。したがって、優先して検査すべきアプリケーションを決定する必要がある。評価順位に従えば、最高評価で需要のあるアプリケーションが疑いを払拭するために最初に検査される。

【0049】

評価センター(RC)111は、アプリケーションの評価を生成してSSP109に提供するように備えられる。RC111は、例えばユーザのコンピューティングデバイスから、アプリケーションの使用に関する情報とアプリケーションに関するユーザからのフィードバックを集める。こうした情報に従って、アプリケーションの信頼性と需要を反映するアプリケーションの評価値が生成されてもよい。RC111は、SSP109を提供するクラウドサービスプロバイダに配備されてもよく、あるいは、アプリケーションストア107又はSSP109に組み込まれてもよい。

30

【0050】

図2は、実施形態に従う包括的マルウェア検出検査の手順を示すフローチャートである。包括的マルウェア検出検査は、いわゆるオフラインマルウェア検出検査とリアルタイムマルウェア検出検査という2段階を含む。203で、アプリケーションに対してオフラインマルウェア検出検査が行われ、例えばSSP109のツールキットによって行われてもよい。SSP109は、アプリケーションの関数呼出しに異常がないかを調べてもよい。オフラインマルウェア検出検査の詳細な手順については、この後の図4を参照して説明する。

40

【0051】

マルウェア検出検査の結果は、例えば205で検査結果(ポジティブ又はネガティブ)の証明書を発行することによって告知されてもよい。マルウェア検出検査の結果がネガティブ、即ちアプリケーションに何らかの異常が存在する場合、SSP109は、検出された問題を告知して検査結果を認定してもよい。マルウェア検出検査の結果がポジティブである場合、SSP109は、ポジティブな検査結果を認定する証明書を発行してもよい。こうした結果は、ユーザのモバイルデバイス101にそのアプリケーションをダウンロードするか否かの決定を助けるために、ユーザに提供されてもよい。またこうした結果は、その

50

アプリケーションの棚卸しやアプリケーション用パッチの開発の何れか又は両方というような、アプリケーションの管理を助けるために、アプリケーションストア 107 に提供されてもよい。オフラインマルウェア検出検査によって、使用に入る前に一部のマルウェアを除去することが可能となる。

【0052】

オフラインマルウェア検出検査は、それぞれのアプリケーションの評価に従ってスケジュールされてもよい。実施形態によっては、SSP 109 は、複数のアプリケーションの中から最も評価の高いアプリケーションを優先して検査するように構成されてもよい。例えば、RC 111 は、マルウェア検査が必要な高評価アプリケーションを決定し、SSP 109 でそのアプリケーションに関するオフラインマルウェア検出検査をトリガしてもよい。

10

【0053】

アプリケーションがオフラインマルウェア検出検査に合格し、コンピューティングデバイス 101 a 等の実環境で実際に実行される場合、207 でアプリケーションに対してリアルタイムマルウェア検出検査を行うことができる。コンピューティングデバイス 101 a は、関数呼出しやデータアクセス挙動、ネットワーク挙動のような、そのアプリケーションの実行時における挙動を監視し、こうした挙動が正常であることを調べてもよい。リアルタイムマルウェア検出検査の詳細な手順については、この後の図 5 を参照して説明する。

【0054】

リアルタイムマルウェア検出検査の結果がネガティブ、即ちアプリケーションの挙動に何らかの異常が存在する場合、コンピューティングデバイスは 209 で、検出された悪質なアプリケーションの制御を再現する。例えば、コンピューティングデバイスのユーザは、このアプリケーションの削除を通知されてもよく、対応する行動をとってもよい。

20

【0055】

前述のように、オフラインマルウェア検出検査は評判評価によって推進することができる。図 3 は、実施形態に従うような処理手順を示している。301 で、RC 111 は、例えばモバイルデバイス 101 からアプリケーションに関するユーザの使用データを収集してもよい。使用データには、アプリ使用挙動データ、アプリ反応挙動データ、アプリ相関挙動データが含まれてもよい。アプリ使用挙動データは、正常なアプリケーションの使用に関連し、主に経過使用時間や使用回数、使用頻度が反映する。アプリ反応挙動データは、ユーザがアプリケーションの問題 / エラーに直面したり、使用体験が良かった / 悪かった後の使用挙動に関連したりする。そしてアプリ相関挙動データは、同じ様に機能するアプリケーションの数に関連する使用挙動に関連する。収集された使用データに基づいて、アプリケーションに対する個々のユーザの信頼を示す信頼度が決定されてもよい。303 で、アプリケーションの信頼度とユーザの主観的なフィードバックの何れか又は両方に基づいて、そのアプリケーションの評価が生成されてもよい。評判の生成又は評価を行うアルゴリズムは多数存在する。例えば、本願の発明者の論文にそうしたアルゴリズムが記載されている。論文のタイトルは "TruBeRepec: A Trust-Behavior-Based Reputation and Recommender System for Mobile Applications (TruBeRepec: モバイルアプリケーションのための信頼-挙動に基づく評価及び推奨システム)" (Z. Yan, P. Zhang, R.H. Deng, Journal of Personal and Ubiquitous Computing, Springer, Vol. 16, Issue 5, pp. 485-506, 2012) である。アプリの評判を生成又は評価する他のアプローチは既知のものもあるが今後も開発されており、こうした方法も利用することができる。

30

40

【0056】

305 で、各アプリケーションの評価に従って、RC 111 は、マルウェア検出検査を行うべきアプリケーションを、例えば評価値の降順で順位付けしてもよい。307 で、RC 111 は、未検査アプリケーションに対するマルウェア検出検査の優先度を決定してもよい。例えば、最初の N 位 (N は検査閾値) までのアプリケーションは、優先してオフラインマルウェア検出検査を受けるように決定されてもよい。309 で、この決定により SSP 109 は、例えば SSP 109 のオフラインマルウェア検出検査ツールキットで検査するために

50

、これら決定されたアプリケーションのコードを1つずつロードするようにトリガされてもよい。

【0057】

アプリの評判評価により、SSP 109は、オフラインマルウェア検出検査の合理的スケジュールを容易に組めるようになり、高評価で需要のある、最も価値の高いアプリケーションの検査に集中することができる。これにより、SSP 109のコストを削減し、マルウェア検出検査をより効率的に行うことができる。例えば、好まれないアプリケーションに対してSSP 109は、そうしたアプリケーションの問題の検出検査を急ぐ必要はなく、検査自体も不要となる。一方、需要のあるアプリケーションに対しては、SSP 109は速やかにそうしたアプリケーションのセキュリティを検査し、それらが引き起こす潜在的リスクを大きく低減することができる。こうして、SSP 109は市場の要求対応に集中することができる。

10

【0058】

オフラインマルウェア検出検査の手順について、図4を参照して説明する。オフラインマルウェア検出検査では、アプリケーション中の異常を発見するために、アプリケーションの関数呼出しの構造が解析される。401で、アプリケーションの関数呼出しマップがオフラインで検出される。関数呼出しマップは、アプリケーションが呼び出す関数間における呼出しの関係を記録してもよい。また、アプリケーションが呼び出す関数を表わすノードと、ノードを結ぶエッジが含まれてもよい。エッジはノード間における関数呼出し関係を示す。エッジはまた、呼出しの向きと順序を示すように向きを有してもよい。マップ中の関数は、アプリケーションが呼び出した関数の何れかであって、アプリケーション自身が設計・分割した関数及びサブ関数が含まれてもよい。あるいは又は加えて、マップ中の関数には、他のアプリケーションが提供するサービス関数や内部のオペレーションシステムが提供するシステムレベルの関数又はサービスの何れか又は全てが含まれてもよい。システムレベルの関数又はサービスには、携帯電話の電話帳機能や写真撮影機能、ネットワーク機能、測位機能等のような機能が含まれる。ノードが表わす粗い粒度の関数は、より細かい粒度の関数の組を含むこともできる。2つのノード間のエッジは、その2つのノードが表わす2つの関数の間での呼出し特性に従って重み付けされてもよい。例えばエッジは、呼出し稠密度と呼出し頻度に基づいて重み付けされてもよい。

20

【0059】

実施形態によっては、関数呼出しのログを獲得し易くするために、アプリケーションの実行コードが逆コンパイルされてから再コンパイルされてもよい。例えば、ツールキットを利用して関数呼出しログを獲得する関数モジュールが、アプリケーションに埋め込まれてもよい。再コンパイルしたアプリケーションの実行コードを仮想環境で実行することを通じて、アプリケーションの関数呼出しのログをオフラインで取得することができる。この関数呼出しログに基づいて、アプリケーションの関数呼出しマップが生成され、アプリケーションの異常を発見するために解析されてもよい。

30

【0060】

403で、関数呼出しマップから、アプリケーションの関数呼出しのパターンが抽出されてもよい。例えば、SSP 109は、データマイニング技術を用いて関数呼出しログを自動的に取り出してもよい。アプリの異常を検出するには、関数呼出しのパターンを抽出するのが効果的である。次いで405で、SSP 109は、抽出パターンと正常なアプリケーションの基本パターン又は規則を比較し、抽出パターンが基本パターン又は規則と一致するかを決定してもよい。例えば、抽出パターンと正常なアプリケーションの基本パターンとの間に有意な差がある場合、又は抽出パターンが正常なアプリケーションの規則に従っていない場合、検査対象アプリケーションは異常と決定されてもよく、マルウェアの可能性もある。

40

【0061】

この段階で、オフラインでのマルウェア検出検査を行うために、動的方法及び静的方法の何れか又は両方が適用されてもよい。実施形態によっては、SSP 109は、別の種類の

50

関数呼出しマップを調べ、動的及び静的の何れか又は両方の方式で、時間経過に伴う呼出し関係や呼出し順序、呼出し濃度、呼出し頻度を解析してもよい。これに関して、SSP 109は、初めから終わりまでの完全シミュレーションの実行を通じて、アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップを調べてもよい。アプリケーションに関する完全呼出しマップはほぼ静的であるため、静的方式で解析することができる。SSP 109は、完全呼出しマップが正常なアプリケーションの基本パターンに一致するかを決定してもよい。例えば、冪乗則次数分散を伴うスケールフリー性及びスモールワールド性複合ネットワーク (Scale-free and Small-world complex network) に一致するかを決定してもよい。

【0062】

あるいは又は加えて、異なる時点での呼出しマップであって、シミュレーション中にアプリケーションがその時点よりも前に呼び出した関数間の呼出し関係を含むものが調べられてもよい。例えば、10分間実行するアプリケーションのシミュレーションを仮定する。ある時点での呼出しマップは、シミュレーションの開始から1分後までにアプリケーションが呼び出した関数間の呼出し関係を含むものであり、それが2分後、3分後、...というような場合とは異なっている。これに対して、完全呼出しマップは、初めから10分後までの全体を通じてアプリケーションが呼出した関数間の呼出し関係を含んでもよい。SSP 109は、異なる時点における呼出しマップが正常なアプリケーションの基本パターンであって、例えば稠密化冪乗則 (Densification Power Law) に一致するかを決定してもよい。

【0063】

あるいは又は加えて、シミュレーション実行中に選択アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップが用いられてもよい。前述のシミュレーションが実施例として行われる場合、部分呼出しマップは、最初の2分間にアプリケーションが呼び出した関数間の呼出し関係を含んでもよく、あるいは第2の2分間、第3の2分間、...というような場合が含まれてもよい。部分呼出しマップは、それが正常なアプリケーションの基本パターンであって、例えば稠密化冪乗則等の特定の安定した挙動パターンに一致するかを決定するために解析されてもよい。

【0064】

正常なアプリケーションの基本パターンは、一般的に、正常なアプリケーションが通常含む特質を反映する。例えば、近年、現実世界のネットワークの発展が一貫した傾向を示すことが多く、次式で表現可能であることが分かっている。

$$e(t) \propto n(t)^a$$

ここで、 $e(t)$ 及び $n(t)$ は、それぞれ時刻 t におけるネットワークのエッジ及びノードの数であり、 a は1から2の間の指数である。この関係は稠密化冪乗則と呼ばれる。また、良質なアプリケーションの呼出しマップを調べたところ、その全てが稠密化冪乗則に従って成長していることを示した研究もある。部分呼出しマップの稠密化冪乗則 (Densification Power Law) という性質は、ソフトウェアシステムのダイナミクスに関する本質的機構に関連する。したがって、アプリケーションの部分呼出しマップが、例えば稠密化冪乗則から大きく逸脱している場合、そのアプリケーションは異常でマルウェアの可能性があると決定することができる。

【0065】

403のパターン抽出及び405のパターン比較は、それぞれ異なる種類の関数呼出しマップに関して行われてもよい。何らかの異常が発見されると、SSP 109は、追加検査のために警告を上げ、例えば、より厳密なマルウェア検出検査処理に移行してもよい。

【0066】

アプリケーションがコンピューティングデバイスにインストールされて実行される場合

、実施形態によっては、悪質なアプリケーションがオフラインマルウェア検出検査に合格したとしても、悪質なアプリケーションの挙動を実行時にでも発見することができる。リアルタイムマルウェア検出検査は、リアルタイムのアプリ挙動からの取出しと、正常なアプリケーションやアプリ挙動の事前記録の何れか又は両方との比較に基づいて行うことができる。リアルタイムマルウェア検出検査の手順について、図5を参照して説明する。こうした実施形態では、1つ又は複数のコンピューティングデバイス（例えば、コンピューティングデバイス101a・101b）で処理500が実行される。この処理は、例えば、図6に示されるようにプロセッサ及びメモリを備えるチップセットとして実装される。そのため、コンピューティングデバイスは処理500の種々の部分を実行する手段に加え、他の要素と協働する他の処理を実行する手段をも提供することができる。

10

【0067】

501で、コンピューティングデバイスは、そのコンピューティングデバイスで実際に実行されるアプリケーションの実行時に、ある時間におけるアプリケーションの挙動を（例えば、コンピューティングデバイスのプロセッサで）記録してもよい。記録されたアプリケーションの挙動情報は、アプリケーション又はコンピューティングデバイスのセキュリティに関連するあらゆる動作、処理、データに関係しうる。例えば、アプリの実行時にコンピューティングデバイスは、部分呼出しマップ及び現時点の呼出しマップのような、関数呼出しに関連する関数呼出し挙動を記録してもよい。コンピューティングデバイスは更に、コンピューティングデバイスのメモリからのデータ読取り動作やコンピューティングデバイスのメモリへのデータ書込み動作のような、ローカルデータアクセスに関連する

20

【0068】

アプリケーションの挙動は、実行時のアプリケーションの実行ログを読み取って記録されてもよい。実施形態によっては、実行時にログを取得するために、対応する機能モジュールを追加することによって、コンピューティングデバイスがアプリケーションの実行コードを再コンパイルしてもよい。あるいは、こうしたアプリケーションの再コンパイルが

30

【0069】

503で、コンピューティングデバイスは、記録されたアプリケーション挙動から、例えばデータマイニング技術を通じて、アプリケーションの実行時における挙動パターンを抽出してもよい。実行時にデータマイニング技術を用いて自動的にアプリ挙動を取り出すと効率的である。次いで、抽出した挙動パターンは、正常なアプリケーションの基本パターン又は規則や、検査されたアプリケーションそれ自体に対して最後に実行した時やオフラインの何れか又は両方で解析されたパターンのような良質なパターンと比較されてもよい。アプリケーションの挙動が通常正常なアプリケーションの挙動又はそのアプリケーション自身の以前の（正常とみられる）挙動と有意に異なる場合、そのアプリケーション

40

【0070】

例えば、コンピューティングデバイスは、（505で）部分呼出しマップ及び現時点の呼出しマップを、前回実行時やオフラインの何れか又は両方で解析されたものに対応する良質なパターンと比較し、（507で）呼出しの挙動に対する抽出パターンが対応する良質なパターンに一致するかを検査してもよい。この検査結果がネガティブである場合、コ

50

ンピューティングデバイスは、509で警告を上げてよい。

【0071】

507での検査結果がポジティブである場合、コンピューティングデバイスは更に、(511で)データアクセス挙動を、前回実行時やオフラインの何れか又は両方で解析されたものに対応する良質なパターンと比較し、(513で)データアクセス挙動に対する抽出パターンに対応する良質なパターンに一致するかを検査してもよい。この検査結果がネガティブである場合、デバイスは、509で警告を上げ、この異常に対応するようユーザに提案してもよい。

【0072】

一方、513での検査結果がポジティブである場合、コンピューティングデバイスは更に、(515で)アプリネットワーク挙動を、前回実行時やオフラインの何れか又は両方で解析されたものに対応する良質なパターンと比較し、(517で)データアクセス挙動に対する抽出パターンに対応する良質なパターンに一致するかを検査してもよい。この検査結果がネガティブである場合、デバイスは、509で警告を上げ、この異常に対応するようユーザに提案してもよい。

【0073】

アプリケーションの挙動に関する上記検査の各々がポジティブである場合、この処理は、517から501への連結線で示されるように、記録されたアプリ挙動の周期的な監視及び解析に戻ってもよい。上記検査は図5を参照して特定の順序で記述されていたが、当然のことながら、こうした動作が異なる順序で実行されてもよく、一部の動作は調整、統合、あるいは削除も可能である。例えば、ネットワーク挙動の検査がアプリ呼出し挙動の検査よりも先に、又は並行して行われるように調整されてもよい。

【0074】

次に図6を参照する。図6は、本発明の様々な実施形態が適用されうる装置600の例示的ブロック図を示す。これはセキュリティサービスプロバイダやコンピューティングデバイスでもよく、コンピューティングデバイスはサーバ、ユーザ装置(UE)、携帯端末、その他のコンピューティングデバイス等を含んでもよい。装置600の一般的な構成は、処理モジュール601と、処理モジュール601に接続する通信インタフェースモジュール609を備える。装置600は更に、共に処理モジュール601に接続するユーザインタフェースモジュール611と不揮発性メモリ613を備える。通信インタフェースモジュール609とユーザインタフェースモジュール611、不揮発性メモリ613も互いに通信してもよい。

【0075】

処理モジュール601はプロセッサ603とメモリ605を備える。処理モジュール601はメモリ605に格納されるソフトウェア607を更に含み、このソフトウェアはプロセッサ603にロードされて実行される。ソフトウェア607は、コンピュータプログラム製品の形式をとりうる1つ又は複数のソフトウェアモジュールを含んでもよい。処理モジュール601は、アプリケーションソフトウェア又はデータ用と装置600の通常動作とで別々の処理・メモリ領域を備えてもよい。

【0076】

通信インタフェース609は、例えばWLANやBluetooth(登録商標)、GSM(登録商標)、GPRS、CDMA、WCDMA(登録商標)、LTE(ロング・ターム・エボリューション)の無線モジュールであってもよい。通信インタフェースモジュール609は装置600自身に統合されていたり、装置600の適切なスロットやポートに挿入されるアダプタやカードのようなものに統合されていたりしてもよい。通信インタフェース609は、単一の無線インタフェース技術をサポートするものであってもよいし、複数の無線インタフェース技術をサポートするものであってもよい。図6において、通信インタフェース609は一つしか描かれていないが、装置600は、複数の通信インタフェースモジュール609を備えている場合もある。

【0077】

プロセッサ 603 は、例えば、中央処理ユニット (CPU)、マイクロプロセッサ、デジタルシグナルプロセッサ (DSP)、グラフィックスプロセッシングユニット等であってもよい。図 6 において、プロセッサ 603 は 1 つしか描かれていないが、装置 600 は、複数のプロセッサを備えてもよい。

【0078】

メモリ 605 には、例えば、不揮発性メモリや揮発性メモリが含まれてもよく、例えば、読み取り専用メモリ (ROM) やプログラム可能な読み取り専用メモリ (PROM)、消去・プログラム可能な読み取り専用メモリ (EPROM)、ランダムアクセスメモリ (RAM)、フラッシュメモリ、データディスク、光学記憶装置、磁気記憶装置、スマートカード等が含まれてもよい。装置 600 は複数のメモリを備えていてもよい。メモリ 605 は、装置 600 の一部として構成されてもよいし、装置 600 のスロットやポート等に挿入されるものであってもよい。メモリ 605 は、データの格納という単一の目的のために使用されてもよいし、データ処理又はマルウェア検出検査等様々な目的のために働く装置の一部として構成されてもよい。不揮発性メモリ 613 はフラッシュメモリ等でもよく、例えば、ソフトウェアの更新を受け取って格納する目的に適うものでもよい。不揮発性メモリ 613 は、装置 600 の一部として構成されてもよいし、装置 600 のスロットやポート等に挿入されるものであってもよい。

【0079】

ユーザインタフェースモジュール 611 は、装置 600 のユーザから入力を受け取る回路手段や、ユーザへ出力を提供する回路手段を備えてもよい。ユーザからの入力は、例えばキーボードや、装置 600 のディスプレイ上に表示されるグラフィカル・ユーザインタフェース、音声認識回路、ヘッドセット等の周辺機器を介して入力されてもよく、ユーザへの出力は、グラフィカル・ユーザインタフェースやスピーカーを通じて行われてもよい。

【0080】

当業者には明らかなことであろうが、装置 600 は、図 6 に描かれた要素以外にも様々な要素を備えることができる。例えばマイクロフォンやディスプレイ、入出力 (I/O) 回路のような追加回路、メモリチップ、特定用途向け集積回路 (ASIC)、特定目的のための処理回路等を備えることができる。また、そのような処理回路には、ソースの符号化 / 復号用回路、チャンネル符号化 / 復号用回路、暗号化 / 平文化回路等が含まれてもよい。さらに装置 600 は、外部からの電源供給が不可能な場合に電力を供給するための、使い切り式や充電式のバッテリーを備えていてもよい (図示されていない)。

【0081】

一般に、多くの典型的実施形態は、ハードウェアまたは特定用途向け回路、ソフトウェア、ロジック、またはそれらの組み合わせで実装される。例えば、ある場合ではハードウェアで実装されてもよく、一方別の場合では、コントローラやマイクロプロセッサ等のコンピュータデバイスによって実行されるファームウェアやソフトウェアで実装されてもよい。ただし、本発明はこれらに限定されるものではない。本発明の典型的実施形態の種々の側面は、ブロック図、フローチャート、または他の図的記述を使用して記述ないし示され得る。これらのブロック、装置、システム、技術、またはここで記述される方法は、非限定的な例として、ハードウェア、ソフトウェア、ファームウェア、特定用途向け回路やロジック、汎用ハードウェア、コントローラや他のコンピュータデバイス、またはそれらの組み合わせで実装されてもよいと理解されるべきである。

【0082】

当然ながら、本発明の例示的实施形態の少なくとも一部の態様はコンピュータ可読命令で具現化することができ、例えば 1 つ又は複数のプログラムモジュールで具現化され、1 つ又は複数のコンピュータ又は他のデバイスで実行することができる。一般に、プログラムモジュールには、コンピュータ又は他のデバイスのプロセッサが実行すると特定のタスクを実行したり、特定の抽象データ型を実装したりするルーチンやプログラム、オブジェクト、コンポーネント、データ構造等が含まれる。コンピュータ可読命令は、ハードディ

10

20

30

40

50

スクや光ディスク、取り外し可能な記憶媒体、ソリッドステートメモリ、RAM等のようなコンピュータ可読媒体に格納されてもよい。当業者には明らかなことであるが、プログラムモジュールの機能は、種々の実施形態で望まれる通りに統合又は分配することができる。加えて、こうした機能は、集積回路やフィールドプログラマブルゲートアレイ（FPGA）等のようなファームウェア又はハードウェア等価物の全部又は一部に具現化されてもよい。

【0083】

本発明の種々の実施形態における様々な特徴は、様々な利点をもたらす。実施形態によっては、マルウェアをオフラインと実行時の両方で検査することによって、マルウェアのリスクを最小限に低減することができる。オフラインマルウェア検出検査中では、マルウェアを発見するために、静的完全呼出しマップと部分呼出しマップ、別の時点での呼出しマップを検査することができる。リアルタイムマルウェア検出検査中では、関数呼出しによって生じるセキュリティ漏洩を発見するために、呼出しマップのパターンを検査することができる。また、リスクの高いローカルデータアクセス、特にこれまでとは異なる異常なアクセスを発見するために、データアクセス挙動を検査することもできる。さらに、アプリケーションのネットワーク挙動の検査を通じて、潜在的な命令を発見するために、アプリケーションの上りトラフィックを検査でき、突然の攻撃で生じる感染で、例えばコンピューティングデバイスをボットに変えてしまうような潜在的感染を解明するために、アプリケーションの下りトラフィックも検査することができる。こうして、コンピューティングデバイスにおけるユーザ情報の盗み取りや売却、デバイス操作、コンテンツ配信、スパム送信、突然の命令を行うマルウェアの対処を可能にし、包括的な検出検査と保護を行うことができる。

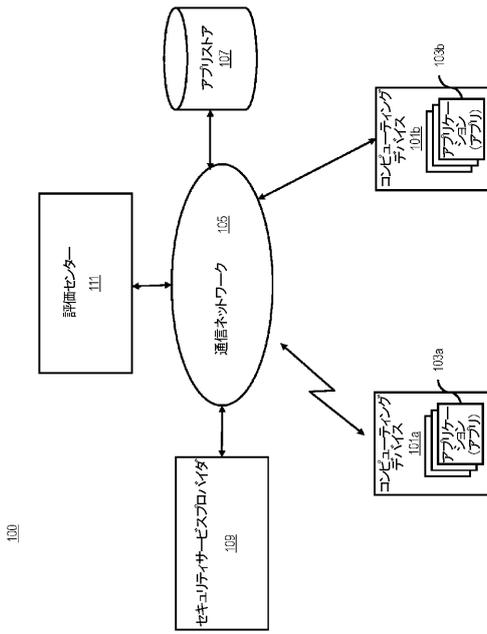
10

20

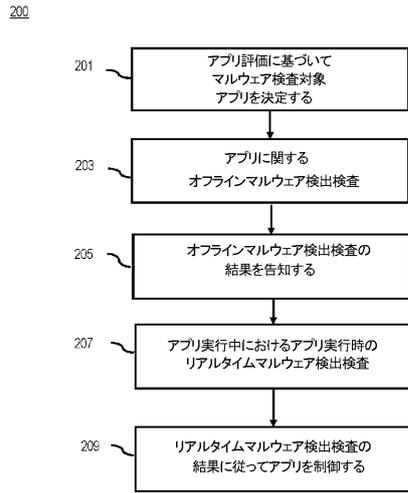
【0084】

本発明には、本明細書に開示された新規の特徴やその組合せが明示的に、又は一般化された形で含まれている。上述した本発明の例示的实施形態への種々の修正や変更は、添付図面と併せて上の説明を考慮すれば、本願に関連する技術分野の当業者には明らかになるだろう。そして、如何なる全ての修正変更も本発明の非限定かつ例示的な実施形態の範囲内である。

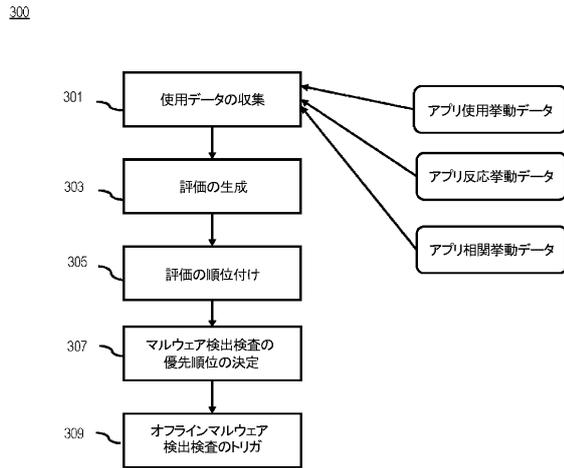
【 図 1 】



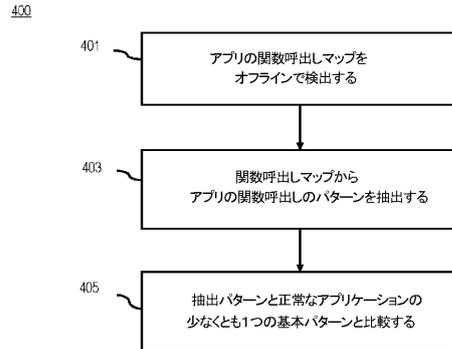
【 図 2 】



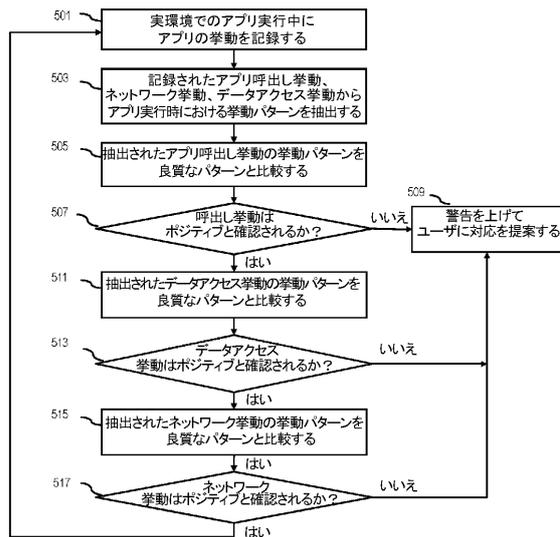
【 図 3 】



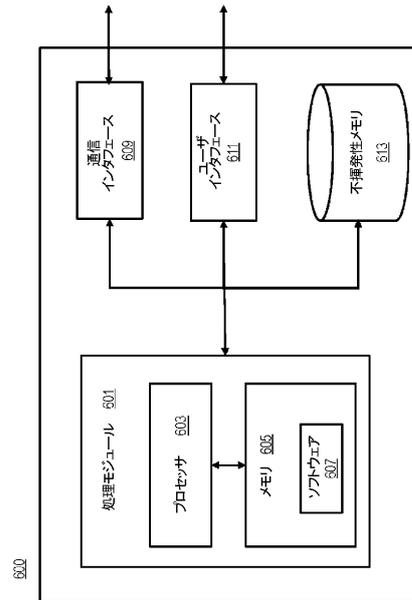
【 図 4 】



【 図 5 】



【 図 6 】



【 手続 補正 書 】

【 提出 日 】 平成 28 年 6 月 17 日 (2016.6.17)

【 手続 補正 1 】

【 補正 対 象 書 類 名 】 特 許 請 求 の 範 囲

【 補正 対 象 項 目 名 】 全 文

【 補正 方 法 】 変 更

【 補正 の 内 容 】

【 特 許 請 求 の 範 囲 】

【 請 求 項 1 】

アプリケーションに関するオフラインマルウェア検出検査を行うことを含む方法であって、前記オフラインマルウェア検出検査は：

前記アプリケーションの少なくとも1つの関数呼出しマップをオフラインで検出することであって、関数呼出しマップは前記アプリケーションが呼び出す関数間における呼出しの関係を記録する、前記検出することと；

前記少なくとも1つの関数呼出しマップから前記アプリケーションの関数呼出しのパターンを抽出することと；

前記抽出されたパターンを正常なアプリケーションの少なくとも1つの基本パターンと比較することと；

を含む、方法。

【 請 求 項 2 】

前記検出することは、仮想環境で前記アプリケーションのコードの少なくとも一部を実行することと、前記アプリケーションの関数呼出しのログを獲得することを含む、請求項1に記載の方法。

【 請 求 項 3 】

前記抽出することは、前記ログを解析するためにデータマイニング法を使用することを

含む、請求項 2 に記載の方法。

【請求項 4】

前記少なくとも 1 つの関数呼出しマップは、次の 3 種類の関数呼出しマップ：

前記実行中に前記アプリケーションが呼出し可能な関数間の全ての呼出し関係を含む完全呼出しマップ；

特定の異なる時点での呼出しマップであって、前記実行中に前記アプリケーションが前記異なる時点よりも前に呼び出した関数間の呼出し関係を含む、前記特定の異なる時点での呼出しマップ；

前記実行中に前記アプリケーションが特定の時間内に呼び出した関数間の呼出し関係を含む部分呼出しマップ；

のうちの少なくとも 1 つを含む、請求項 3 に記載の方法。

【請求項 5】

前記オフラインマルウェア検出検査は、

前記関数呼出しマップのログを獲得する関数モジュールの追加によって、前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることと；

前記ログを獲得するために、前記仮想環境で前記再コンパイルされたコードを実行することと；

を更に含む、請求項 1 に記載の方法。

【請求項 6】

前記アプリケーションの評価に従って前記オフラインマルウェア検出検査をスケジュールすることを更に含む、請求項 1 に記載の方法。

【請求項 7】

高評価アプリケーションは、低評価アプリケーションよりも優先して前記オフラインマルウェア検出検査を受けるようにスケジュールされる、請求項 6 に記載の方法。

【請求項 8】

前記アプリケーションの潜在的不正の脅威を示す前記オフラインマルウェア検出検査の結果を告知することを更に含む、請求項 1 に記載の方法。

【請求項 9】

前記アプリケーションが実環境で実行される間に前記アプリケーションに関するリアルタイムマルウェア検出検査を行うことを更に含む、前記リアルタイムマルウェア検出検査は：

前記アプリケーションの実行中に前記アプリケーションの挙動を記録することと；

前記記録された挙動から挙動パターンを抽出することと；

前記抽出された挙動パターンを正常なアプリケーションの少なくとも 1 つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；

を含む、請求項 1 から 8 の何れかに記載の方法。

【請求項 10】

前記記録された挙動は、次の 3 種類の挙動：

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び/又は下りトラフィックに関連する挙動；

のうちの少なくとも 1 つを含む、請求項 9 に記載の方法。

【請求項 11】

前記リアルタイムマルウェア検出検査は、

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含む、請求項 9 又は 10 に記載の方法。

【請求項 12】

実環境でアプリケーションを実行することと；

前記アプリケーションの実行時における前記アプリケーションの挙動を記録することと

;

前記記録された挙動から挙動パターンを抽出することと；

前記抽出された挙動パターンを正常なアプリケーションの少なくとも1つの基本パターン又は以前に記録された前記アプリケーションのパターンと比較することと；
を含む、方法。

【請求項13】

前記記録された挙動は、次の3種類の挙動：

前記アプリケーションの関数呼出しに関連する挙動；

前記アプリケーションが行ったローカルデータアクセスに関連する挙動；

前記アプリケーションが起こした上り及び/又は下りトラフィックに関連する挙動；

のうちの少なくとも1つを含む、請求項12に記載の方法。

【請求項14】

前記アプリケーションの挙動のログを獲得する関数モジュールの追加によって前記アプリケーションの実行コードの少なくとも一部を再コンパイルすることを更に含む、請求項12又は13に記載の方法。

【請求項15】

処理手段及び記憶手段を備える装置であって、前記記憶手段はプログラム命令を格納し、前記プログラム命令は、前記処理手段に実行されると、前記装置に、請求項1から14のいずれかに記載の方法を遂行させるように構成される、装置。

【請求項16】

装置の処理手段に実行されると、前記装置に、請求項1から14のいずれかに記載の方法を遂行させるように構成されるプログラム命令を備える、コンピュータプログラム。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/CN2013/090887
A. CLASSIFICATION OF SUBJECT MATTER G06F 21/56(2013.01)i 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) G06F; H04W Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) CNPAT, WPI, EPODOC: malware or virus or botnet or worm or horse or malicious, code or software or program or app or application, detect or monitor or inspect or supervise or check or examine, virtual or simulate or emulate or simulation or emulation or simulator or emulator or offline or sandbox, perform or run or operation, call, sequence or map or relation or diagram, real or reality or true or realism		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 102034042 A (SICHUAN UNIVERSITY) 27 April 2011 (2011-04-27) description paragraphs [0011], [0024], [0026], [0059], [0062], [0067], [0077], and figure 1	1-8, 15-22, 29-31
Y	CN 102034042 A (SICHUAN UNIVERSITY) 27 April 2011 (2011-04-27) description paragraphs [0011], [0024], [0026], [0059], [0062], [0067], [0077], and figure 1	9-11, 23-25
X	CN 103369532 A (HUANG, YUHUI) 23 October 2013 (2013-10-23) description paragraphs [0004], [0005], [0011], [0015], and figure 1	12-14, 26-31
Y	CN 103369532 A (HUANG, YUHUI) 23 October 2013 (2013-10-23) description paragraphs [0004], [0005], [0011], [0015], and figure 1	9-11, 23-25
A	US 8434151 B1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 30 April 2013 (2013-04-30) the whole document	1-31
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* 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 11 September 2014		Date of mailing of the international search report 30 September 2014
Name and mailing address of the ISA/ STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN) 6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China Facsimile No. (86-10)62019451		Authorized officer NING,Bo Telephone No. (86-10)62413288

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2013/090887

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 102034042 A	27 April 2011	Non e	
CN 103369532 A	23 October 2013	Non e	
US 8434151 B1	30 April 2013	Non e	

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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, US

(特許庁注：以下のものは登録商標)

- 1 . A N D R O I D
- 2 . W I N D O W S