

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-27710

(P2012-27710A)

(43) 公開日 平成24年2月9日(2012.2.9)

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

審査請求 未請求 請求項の数 13 O L (全 13 頁)

(21) 出願番号 特願2010-166150 (P2010-166150)
(22) 出願日 平成22年7月23日 (2010.7.23)(71) 出願人 000004226
日本電信電話株式会社
東京都千代田区大手町二丁目3番1号
(71) 出願人 899000068
学校法人早稲田大学
東京都新宿区戸塚町1丁目104番地
(74) 代理人 100070150
弁理士 伊東 忠彦
(74) 代理人 100124844
弁理士 石原 隆治
(72) 発明者 森 達哉
東京都千代田区大手町二丁目3番1号 日
本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 ソフトウェア検出方法及び装置及びプログラム

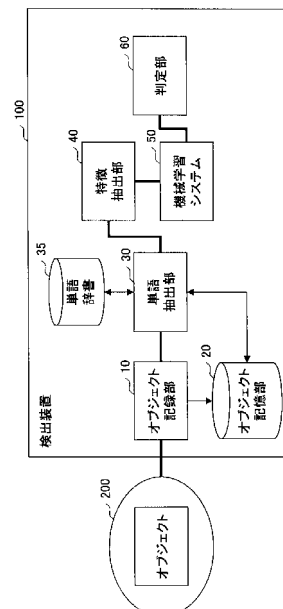
(57) 【要約】

【課題】 未知のパターンにも対応可能な高速なマルウェア及びマルウェアを含む通信を検出する。

【解決手段】 本発明は、オブジェクトを記憶するオブジェクト記憶手段と、オブジェクト記憶手段に記録されたオブジェクトより文字列として印字及び可読なバイト列を単語として抽出する単語抽出手段と、単語抽出手段によって抽出された単語の集合を元にオブジェクトごとに特徴ベクトルを構成する特徴抽出手段と、特徴ベクトルに機械学習システムを適用することにより新たに観測した特徴ベクトルを元に前記オブジェクトがマルウェアであるか否かを判定する判定手段と、を有する。

【選択図】 図1

本発明の一実施の形態におけるシステムの構成例



【特許請求の範囲】

【請求項 1】

悪意のあるソフトウェア（以下、「マルウェア」と記す）あるいはマルウェアを送受信する通信を検出する悪意のあるソフトウェアを検出するソフトウェア検出装置であって、ファイルあるいは送受信する通信データ（以下、ファイルあるいは通信データをまとめて「オブジェクト」と記す）を記憶するオブジェクト記憶手段と、

前記オブジェクト記憶手段に記録されたオブジェクトより文字列として印字及び可読なバイト列を単語として抽出する単語抽出手段と、

前記単語抽出手段によって抽出された単語の集合を元にオブジェクトごとに特徴ベクトルを構成する特徴抽出手段と、

前記特徴ベクトルに機械学習システムを適用することにより新たに観測した特徴ベクトルを元に前記オブジェクトがマルウェアであるか否かを判定する判定手段と、
を有することを特徴とするソフトウェア検出装置。

10

【請求項 2】

前記機械学習システムは、

任意に設定が可能な周期が到来する毎に新たな学習データをフィードバックすることによって再学習を可能とする

請求項 1 記載のソフトウェア検出装置。

【請求項 3】

前記単語抽出手段は、

任意の単語辞書を参照して、印字可能な文字から可読な単語集合を抽出する手段を含む
請求項 1 記載のソフトウェア検出装置。

20

【請求項 4】

前記単語抽出手段は、

前記単語辞書として、予め用意した静的な辞書、または、ネットワーク上の検索エンジンの結果を用いる

請求項 3 記載のソフトウェア検出装置。

【請求項 5】

前記判定手段は、

マルウェアであるか否かのみならず、マルウェアの種別に関して学習並びに判定を行う
手段を含む

請求項 1 記載のソフトウェア検出装置。

30

【請求項 6】

前記判定手段は、

前記オブジェクトの全体または一部のみを使って判定する手段を含む

請求項 1 または 5 記載のソフトウェア検出装置。

【請求項 7】

オブジェクト記憶手段、単語抽出手段、特徴抽出手段、判定手段を有し、悪意のあるソフトウェア（以下、「マルウェア」と記す）あるいはマルウェアを送受信する通信を検出する装置における、悪意のあるソフトウェアを検出するソフトウェア検出方法であって、

前記単語抽出手段は、ファイルあるいは送受信する通信データ（以下、ファイルあるいは通信データをまとめて「オブジェクト」と記す）を記憶する前記オブジェクト記憶手段に記録されたオブジェクトより文字列として印字及び可読なバイト列を単語として抽出する単語抽出ステップと、

前記特徴抽出手段が、前記単語抽出ステップによって抽出された単語の集合を元にオブジェクトごとに特徴ベクトルを構成する特徴抽出ステップと、

前記判定手段が、前記特徴ベクトルに機械学習システムを適用することにより新たに観測した特徴ベクトルを元に前記オブジェクトがマルウェアであるか否かを判定する判定ステップと、

を行うことを特徴とするソフトウェア検出方法。

40

50

【請求項 8】

前記判定ステップにおいて、

任意に設定が可能な周期が到来する毎に新たな学習データをフィードバックすることによって再学習を可能とする前記機械学習システムを用いる、

請求項 7 記載のソフトウェア検出方法。

【請求項 9】

前記単語抽出ステップにおいて、

任意の単語辞書を参照して、印字可能な文字から可読な単語集合を抽出する

請求項 7 記載のソフトウェア検出方法。

【請求項 10】

10

前記単語抽出ステップにおいて、

前記単語辞書として、予め用意した静的な辞書、または、ネットワーク上の検索エンジンの結果を用いる

請求項 9 記載のソフトウェア検出方法。

【請求項 11】

前記判定ステップにおいて、

マルウェアであるか否かのみならず、マルウェアの種別に関して学習並びに判定を行う

請求項 7 記載のソフトウェア検出方法。

【請求項 12】

20

前記判定ステップにおいて、

前記オブジェクトの全体または一部のみを使って判定する

請求項 7 または 11 記載のソフトウェア検出方法。

【請求項 13】

請求項 1 乃至 6 のいずれか 1 項に記載のソフトウェア検出装置を構成する各手段としてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ソフトウェア検出方法及び装置及びプログラムに係り、特に、教師付き機械学習システムを用いたマルウェアあるいはマルウェアを送受信する通信の検出を行うためのソフトウェア検出方法及び装置及びプログラムに関する。

30

【背景技術】

【0002】

ワームやボットネットと呼ばれるマルウェア（悪意のあるソフトウェア）による被害が拡大・深刻化している。マルウェアに感染したコンピュータはネットワークに接続された他のコンピュータに対して不正あるいは有害な動作を行うことが特徴であり、迷惑メールの大量送信や、サーバへの不正な大量アクセスによるサービス妨害攻撃といった悪質な行動を行うためのツールとして使われる。マルウェアの脅威は外部に対する攻撃のみならず、感染したコンピュータからクレジット番号やアドレス帳などの個人情報抽出し、外部のコンピュータに送信する活動も存在する。このようなマルウェアによる被害を未然に防ぐためにはマルウェア本体、あるいはマルウェア本体を送受信している通信を検出する技術が必要となる。

40

【0003】

マルウェアにはパターンマッチングによる検出や逆アセンブラによる静的解析を避けるために意図的に複雑な暗号化や圧縮（パッキングと呼ばれる）を施したものが多い。パッキングされた実行ファイルは実行時に自分自身を復号してメモリ上に展開する。

【0004】

一般にマルウェアの詳細な分析を実施するためには、パッキングされた実行ファイルを様々な技術を駆使して復号する必要がある。

【0005】

50

更に復号したファイルに対して、逆アセンブルやデバッガを用いる手法が一般的に利用されている（例えば、比特許文献1参照）。

【先行技術文献】

【非特許文献】

【0006】

【非特許文献1】K. Rozinov. "Reverse Code Engineering: An In-Depth Analysis of the Bagle Virus". Bell Labs - Government communication Laboratory - Internet Research, August 2004. http://old.idapalace.net/papers/bagle_analysis_v.1.0.pdf

【発明の開示】

【発明が解決しようとする課題】

10

【0007】

しかしながら、上記のような復号・逆アセンブル、デバッガを用いる手法は時間と演算リソースがかかるため、リアルタイムでマルウェアを含む通信を検出する用途には向かない。

【0008】

マルウェアを実際のオペレーティングシステム上で実行し、その挙動を分析することによってマルウェアであるかを検出するアプローチとして、例えば、文献「伊藤光恭、針生剛男、谷本直人、岩村誠、八木毅、川古谷裕平、青木一史、秋山満昭、中山心太、"マルウェア対策技術", NTT技術ジャーナル2010年3月号、pp. 40-44. <http://www.ntt.co.jp/journal/1003/files/jn201003040.html>」に記載された方法もあるが、やはり時間と演算リソースを消費するため、即時的な検出・判定ができないという問題があった。

20

【0009】

本発明は、上記の点に鑑みなされたもので、未知のパターンにも対応可能な高速かつ処理の軽いマルウェア及びマルウェアを含む通信の検出を実現することが可能なソフトウェア検出方法及び装置及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【0010】

上記の課題を解決するため、本発明（請求項1）は、悪意のあるソフトウェア（以下、「マルウェア」と記す）あるいはマルウェアを送受信する通信を検出する悪意のあるソフトウェアを検出するソフトウェア検出装置であって、

30

ファイルあるいは送受信する通信データ（以下、ファイルあるいは通信データをまとめて「オブジェクト」と記す）を記憶するオブジェクト記憶手段と、

前記オブジェクト記憶手段に記録されたオブジェクトより文字列として印字及び可読なバイト列を単語として抽出する単語抽出手段と、

前記単語抽出手段によって抽出された単語の集合を元にオブジェクトごとに特徴ベクトルを構成する特徴抽出手段と、

前記特徴ベクトルに機械学習システムを適用することにより新たに観測した特徴ベクトルを元に前記オブジェクトがマルウェアであるか否かを判定する判定手段と、を有する。

【0011】

また、本発明（請求項2）は、前記機械学習システムにおいて、

40

任意に設定が可能な周期が到来する毎に新たな学習データをフィードバックすることによって再学習を可能とする。

【0012】

また、本発明（請求項3）は、前記単語抽出手段において、

任意の単語辞書を参照して、印字可能な文字から可読な単語集合を抽出する手段を含む。

【0013】

また、本発明（請求項4）は、上記請求項3の前記単語抽出手段において、

前記単語辞書として、予め用意した静的な辞書、または、ネットワーク上の検索エンジンの結果を用いる。

50

【0014】

また、本発明（請求項5）は、前記判定手段において、マルウェアであるか否かのみならず、マルウェアの種別に関して学習並びに判定を行う手段を含む。

【0015】

また、本発明（請求項6）は、上記請求項1または5の前記判定手段において、前記オブジェクトの全体または一部のみを使って判定する手段を含む。

【0016】

また、本発明（請求項7）は、オブジェクト記憶手段、単語抽出手段、特徴抽出手段、判定手段を有し、悪意のあるソフトウェア（以下、「マルウェア」と記す）あるいはマルウェアを送受信する通信を検出する装置における、悪意のあるソフトウェアを検出するソフトウェア検出方法であって、

前記単語抽出手段は、ファイルあるいは送受信する通信データ（以下、ファイルあるいは通信データをまとめて「オブジェクト」と記す）を記憶する前記オブジェクト記憶手段に記録されたオブジェクトより文字列として印字及び可読なバイト列を単語として抽出する単語抽出ステップと、

前記特徴抽出手段が、前記単語抽出ステップによって抽出された単語の集合を元にオブジェクトごとに特徴ベクトルを構成する特徴抽出ステップと、

前記判定手段が、前記特徴ベクトルに機械学習システムを適用することにより新たに観測した特徴ベクトルを元に前記オブジェクトがマルウェアであるか否かを判定する判定ステップと、を行う。

【0017】

また、本発明（請求項8）は、前記判定ステップにおいて、

任意に設定が可能な周期が到来する毎に新たな学習データをフィードバックすることによって再学習を可能とする前記機械学習システムを用いる。

【0018】

また、本発明（請求項9）は、前記単語抽出ステップにおいて、

任意の単語辞書を参照して、印字可能な文字から可読な単語集合を抽出する。

【0019】

また、本発明（請求項10）は、上記請求項9の前記単語抽出ステップにおいて、

前記単語辞書として、予め用意した静的な辞書、または、ネットワーク上の検索エンジンの結果を用いる。

【0020】

また、本発明（請求項11）は、前記判定ステップにおいて、

マルウェアであるか否かのみならず、マルウェアの種別に関して学習並びに判定を行う。

【0021】

また、本発明（請求項12）は、上記請求項7または11の前記判定ステップにおいて、

前記オブジェクトの全体または一部のみを使って判定する。

【0022】

また、本発明（請求項13）は、請求項1乃至6のいずれか1項に記載のソフトウェア検出装置を構成する各手段としてコンピュータを機能させるためのプログラムである。

【発明の効果】

【0023】

上記のように、本発明では、ソフトウェアファイルに含まれる可読な文字列に着目し、マルウェアを構成するプログラム中に含まれるメッセージ、通信に利用するために用いられるホスト名、実行コマンド名、関数名やライブラリ名などのマルウェアを特徴付ける情報を（特徴ベクトル）を収集しておき、この内容と検査対象のファイルに含まれる文字列とを比較することにより、マルウェアを検出するため、未知のパターンにも対応可能な高

速なマルウェア及びマルウェアを含む通信の検出が可能になる。

【図面の簡単な説明】

【0024】

【図1】本発明の一実施の形態におけるシステムの構成例である。

【図2】本発明の一実施の形態における文字列処理のフローチャートである。

【発明を実施するための形態】

【0025】

本発明は、マルウェアと通常のデータの比較にあたり、データに含まれる印字可能な文字列を適切に処理し、可読な単語集合とした際に単語集合間に差異が存在するという性質を利用する。可読な文字列にはマルウェアを構成するプログラム中に含まれるメッセージ、通信に利用するために用いられるホスト名、実行コマンド名に加え、関数名やライブラリ名などがあり、マルウェアに関する様々な情報を収集することができる。

10

【0026】

マルウェアにはコードの一部のみを改変した亜種と呼ばれるソフトウェアが存在する。今日のマルウェアの大多数はオリジナルが開発された後に、多数の亜種が開発されることが知られているが、亜種のコードの大部分はオリジナルと変わらないため、印字可能な文字列の特徴はオリジナルとほぼ同様となる可能性が高い。

【0027】

以下に、本発明の概要を示す。

【0028】

20

(1) はじめにファイルあるいは送受信する通信データ（以下、ファイルあるいは通信データをまとめて「オブジェクト」とする）を後述するオブジェクト記録部によって記録する。

【0029】

(2) 次に、後述する単語抽出部が、上記の記録したオブジェクトにより文字列として印字及び可読なバイト列を単語として抽出し、単語集合とする。

【0030】

(3) 次に、上記の単語集合を元にオブジェクト毎に特徴ベクトルを構成し、さらに上記の特徴ベクトルに機械学習システムを適用することによって、判定機能により新たに観測した特徴ベクトルを元に該オブジェクトがマルウェアであるか否かを判定する。ここで、特徴ベクトルはある単語が存在したか否かを「0」、「1」で表現したもの、あるいは、単語毎の頻度、あるいは、単語毎の頻度を正規化した値、あるいは単語毎の $t f - i d f$ 値など、任意に定めることができる。

30

【0031】

上記の教師付き機械学習システムは、任意の設定が可能な周期が到来する毎に、新たなデータをフィードバックすることによって再学習が可能である。ここで、新たなデータとは、先行技術などの方法でリアルタイムではないものの、時間をかけてマルウェアあるいは通常ファイルと判定されたファイルに関するデータなどである。

【0032】

上記の(2)の単語抽出部は、任意の単語辞書を利用することによって、印字可能な文字列から可読な単語集合を抽出する。このとき、予め用意した静的な辞書の他、ネットワーク上の検索エンジンの結果を利用することもできる。すなわち、印字可能な文字列を任意に指定が可能な検索エンジンで検索し、その結果が1件以上、あるいは、一般にX件以上であったときに可読であると判定する。

40

【0033】

上記の判定機能では、マルウェアであるか通常ファイルであるのかの判定のみならず、他のクラスの分類を実現する機械学習によってマルウェアの種別に関しても判定を行う機能を有する。

【0034】

オブジェクトに対する判定は、そのオブジェクトの全体ではなく、一部のみを使って判

50

定を実施することも可能である。例えば、通信データの場合、全てのデータを受信し終わる前に、現在までに受信したデータを用いて判定することができる。

【0035】

以下図面と共に、本発明の実施の形態を詳細に説明する。

【0036】

図1は、本発明の一実施の形態におけるシステム構成を示す。

【0037】

検出装置100は、ネットワーク200と接続され、オブジェクト記録部10、オブジェクト記憶部20、単語抽出部30、単語辞書35、特徴抽出部40、機械学習システム50、判定部60から構成される。ここで、オブジェクト記憶部20及び単語辞書35は、ハードディスク等の記憶媒体である。

10

【0038】

オブジェクト記録部10は、マルウェアの実行ファイル（バイナリ）あるいはマルウェアを送受信している通信をオブジェクトとしてオブジェクト記憶部20に記録する。

【0039】

単語抽出部30は、オブジェクト記憶部20からオブジェクトを読み出して、オブジェクトを解析し、はじめに印字可能な文字コードがX文字以上連続する文字列を抽出する。Xは任意の正数（ $X = 1, 2, \dots$ ）に設定が可能である。ここで、印字可能であるとは、例えば、ASCIIコードにおいては、制御文字を除いた 0×20 から $0 \times 7e$ までの空白及び図形文字の文字コードを指す。表1に得られた文字列の例を示す。

20

【0040】

【表1】

表1 抽出した印字可能な文字列

文字列
.text
SUVM3
_set_app_type
L\$^L^]
ResetKeySecurity

30

なお、表1に示す文字列は単語抽出部30内のメモリ（図示せず）に格納されるものとする。

【0041】

40

さらに、抽出された文字列に対し、任意に設定が可能な規則に従って文字列を処理し、単語の候補とする。図2に文字列処理の例を示す。

【0042】

ステップ101) 文字列の英数字以外を区切り文字に置換する。

【0043】

ステップ102) 大文字が2文字以上連続している場合は、ステップ103に移行し、連続していない場合は、ステップ104に移行する。

【0044】

ステップ103) 該当した大文字を一つの単語としてマークする。

【0045】

50

ステップ 104) マークされた以外に大文字がある場合は、ステップ 105 に移行し、ない場合はステップ 106 に移行する。

【0046】

ステップ 105) 該当した大文字の前に区切り文字を挿入する。

【0047】

ステップ 106) 文字列を区切り文字によって分割し、分割した各々の文字列を単語とする。

【0048】

上記のフローに従って文字列処理を実施した結果、上記の表 1 に示す文字列は表 2 に示す単語候補となる。

【0049】

【表 2】

10

表2

単語候補
text
suvn
set
app
type
reset
key
security

20

なお、表 2 に示す文字列は単語抽出部 30 内のメモリ（図示せず）に格納されるものとする。

【0050】

単語抽出部 30 は、上記の表 2 に示す単語候補に対して予め準備しておいた単語辞書 35 を用いて、単語辞書 35 に登録されている単語にマッチした単語のみ抽出する。あるいは、インターネットの検索エンジンで該当する単語候補を検索し、検索結果の数がある任意に設定が可能な閾値以上であったときのみ単語として抽出する。単語辞書 35 とマッチした結果、表 3 に示す単語集合を得る。

【0051】

40

【表 3】

表3

単語候補
text
set
app
type
reset
key
security

10

なお、表 3 に示す文字列は単語抽出部 30 内のメモリ（図示せず）に格納されるものとする。 20

【0052】

特徴抽出部 40 は、各々のファイルについて表 3 で得られた単語集合を単語抽出部 30 のメモリから読み出して、特徴ベクトルを構成する。すなわち、

ファイル X_i ($i = 1, 2, \dots, N$) の特徴ベクトル

$$V_i = \{v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{iM}\}$$

を各々のファイル X_i に含まれる各単語 w_j ($j = 1, 2, \dots, M$) を用い、

$$v_{ij} = g(w_j, X_i, X, a)$$

のように、 w_j , X_i , X , a の関数で定義する。ここで、 X は全てのファイルの集合であり、 a は任意に定義可能な重みパラメータである。 30

【0053】

v_{ij} の構成例としては、例えば、公知の $tf-idf$ 値を用いることができる。すなわち、

$$v_{ij} = tf_{ij} * idf_j$$

であり、

$$tf_{ij} = n_{ij} / \sum_k n_{kj}$$

$$idf_j = \log(N / c)$$

である。ここで、

- ・ n_{ij} はファイル F_i において単語 w_j が生起した回数；
- ・ $\sum_k n_{kj}$ は単語 w_j が全てのファイルで生起した回数の合計；
- ・ N は全てのファイルの総数；
- ・ c は単語 w_j を含むファイルの総数；

である。なお、 $*$ は乗算を示す演算子である。 40

【0054】

判定部 60 は、特徴ベクトル V_i に機械学習システム 50 を適用する。

【0055】

特徴ベクトル V_i に教師付き機械学習システム 50 を適用することによって、ファイルを「マルウェア」、「通常ファイル」に分類する。具体的には、予め「マルウェア」、「通常ファイル」のそれぞれのファイルの特徴ベクトルを学習し、新たに得られたファイルについてその特徴ベクトルを元に「マルウェア」であるか「通常ファイル」であるかを判定する。 50

【0056】

機械学習システム50の機械学習の手法としては、公知の教師あり学習の手法を用いることができる。教師あり学習では事前に教師データ (V_1, y_1) , (V_2, y_2) , ..., (V_N, y_N) が与えられる。ここで、 V_i は前述のファイル X_i に対する特徴ベクトルである。 y_i は同ファイルのクラスであり、「0」か「1」の二値をとる。「0」なら通常ファイル、「1」ならマルウェアのように定義する。

【0057】

上記の教師あり学習は、新たに観測した V_m に対するクラス y_m を高精度に推定する手法であり、サポートベクターマシン(SVM)、ナイーブベイズ法、ニューラルネットワーク、決定木など様々な公知の手法が広く使われている。

10

【0058】

以下では、SVMを教師あり学習として適用した結果を例示するが、本発明における機械学習の適用範囲はこの例に限定されることなく、上記の種々の方法が適用可能である。

【0059】

表4は、機械学習システム50において、各ファイルに対し、出現する単語及びその出現単語数を分析した結果であり、判定部60のメモリ(図示せず)に格納される。

【0060】

【表4】

20

表4

ファイル名(hash値)	出現単語:回数
00c4fc2fd0ac1d2b2fa43d8c5a087104	dll:4,get:2,sip:2,kernel:2,address:1,change:1,...
013a5ba10e3fc8a039b045530381d957	dll:4,get:2,kernel:2,rhd:1,bde:1,amu:1,range:1,...
notepad.exe	get:47>window:20,set:19,file:16,message:13,...
.....	

30

次に、表5は上記の表4を元に、特徴ベクトルを構成した例である。単語は単語辞書35を元に番号が付けられている。特徴ベクトルの表記は" $j: v_{ij}$ "とする。すなわち、{単語の番号: tf-idf値}である。ここで、クラスは"1"はマルウェア、"-1"は通常ファイルとした。

【0061】

【表5】

40

表5

ファイルID	特徴ベクトル	クラス
1	119:0.0008, 196:0.0077, 307:0.0077,...	-1
2	119:0.0024, 168:0.0051, 213:0.0382,...	-1
3	84:0.006, 94:0.002, 113:0.00258,...	+1
....		

50

上記の特徴ベクトルにSVMを適用した結果の一例を表6に示す。ここでは、マルウェアを221ファイル、通常ファイルを1526ファイルを利用した。マルウェアは実際のインターネットで収集したものであり、通常ファイルは市販のオペレーティングシステムにインストールされているファイルを用いた。

【0062】

また、学習には各々100ファイルを用い、残り全てのファイルについて判定を実施した。すなわち、既知の100+100 = 200ファイルを用いて機械学習を行い、その200ファイルも除いた未知の1547ファイルに対して判定を行った。得られた結果は精度(precision)が99.37%、再現率(Recall)が99.86%と高い精度を得ることができる。

【0063】

【表6】

実際	推定	判定数
+1	+1	1424
-1	+1	9
-1	-1	112
+1	-1	2

上記の例のように2クラス、すなわちマルウェアであるか通常ファイルであるかの判定のみならず、他のクラスの分類を実現する機械学習によってマルウェアの種別に関しても機械学習で判定が可能である。すなわち、マルウェアの種別 c_k ($k=1, 2, \dots, K$)をクラスとし、機械学習を所与の特徴ベクトルに対してクラス C_k を判定する問題設定で適用することにより、そのような分類が可能となる。

【0064】

また、機械学習システム50は、任意の設定が可能な周期が到来する毎に新たなデータをフィードバックすることによって再学習が可能である。ここで、新たなデータとは、先行技術などの方法でリアルタイムではないもの、時間をかけてマルウェアあるいは通常ファイルと判定されたファイルに関するデータなどである。

【0065】

上記のオブジェクトに対する判定はそのオブジェクト全体ではなく、一部のみを使って判定を実施することも可能である。例えば、通信データの場合、全てのデータを受信し終わる前に、現在までに受信したデータを用いて判定する。

【0066】

上記のように、本発明では、ソフトウェアファイルに含まれる可読な文字列に着目し、マルウェアを構成するプログラム中に含まれるメッセージ、通信に利用するために用いられるホスト名、実行コマンド名、関数名やライブラリ名などのマルウェアを特徴づける情報を(特徴ベクトル)を収集しておき、この内容と検査対象のファイルに含まれる文字列とを比較することにより、マルウェアを検出することができる。

【0067】

上記の図1に示す検出装置の各構成要素の動作をプログラムとして構築し、検出装置として利用されるコンピュータにインストールして実行させる、または、ネットワークを介して流通させることが可能である。

【0068】

また、構築されたプログラムをハードディスクや、フレキシブルディスク、CD-ROM等の可搬記憶媒体に格納し、コンピュータにインストールする、または、配布することが可能

である。

【0069】

なお、本発明は上記の実施の形態に限定されることなく、特許請求の範囲内において種々変更・応用が可能である。

【符号の説明】

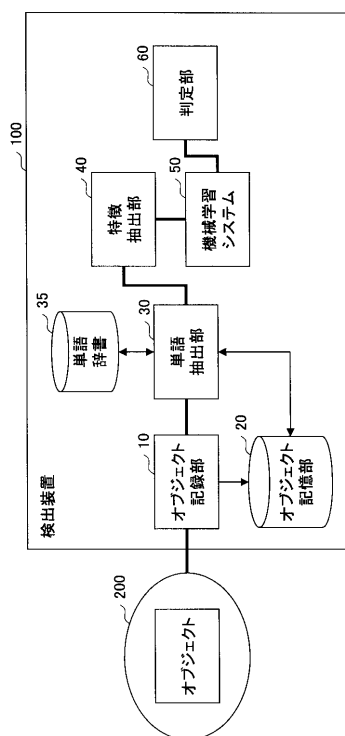
【0070】

- 10 オブジェクト記録部
- 20 オブジェクト記憶部
- 30 単語抽出部
- 35 単語辞書
- 40 特徴抽出部
- 50 機械学習システム
- 60 判定部
- 100 検出装置
- 200 ネットワーク

10

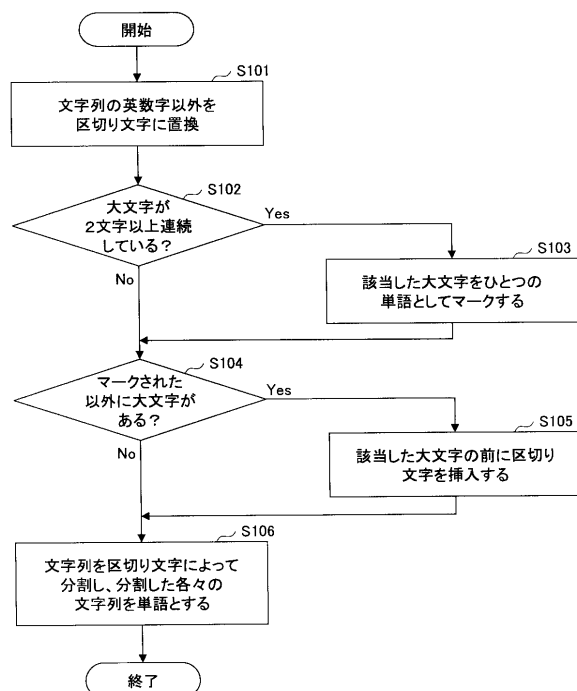
【図1】

本発明の一実施の形態におけるシステムの構成例



【図2】

本発明の一実施の形態における文字列処理のフローチャート



フロントページの続き

- (72)発明者 戸部 和洋
東京都新宿区戸塚町1丁目104番地 学校法人早稲田大学内
- (72)発明者 下田 晃弘
東京都新宿区戸塚町1丁目104番地 学校法人早稲田大学内
- (72)発明者 後藤 滋樹
東京都新宿区戸塚町1丁目104番地 学校法人早稲田大学内
- Fターム(参考) 5B276 FD08

(19) Japan Patent Office (JP)

(12) **Japanese Unexamined Patent
Application Publication (A)**(11) Japanese Unexamined Patent
Application Publication Number**Japanese Patent Application****2012-27710****(P2012-27710A)**(43) Publication date **February 9, 2012 (2.9.2012)**

(51) Int. Cl.

G06F 21/22**(January
2006)**

F1

G06F

9/06

660N

Theme codes (reference)

5B276

	Request for examination	Not yet requested	Number of claims 13 OL (Total of 13 pages)
(21) Application Number	Japanese Patent Application 2010-166150 (P2010-166150)	(71) Applicant 000004226 Nippon Telegraph and Telephone Corp. 2-3-1 Otemachi, Chiyoda-ku, Tokyo	
(22) Date of application	July 23, 2010 (7.23.2010)	(71) Applicant 899000068 Waseda University 1-104 Totsuka-cho, Shinjuku-ku, Tokyo	
		(74) Agent 100070150 Patent attorney Tadahiko Ito	
		(74) Agent 100124844 Patent attorney Takashi Ishihara	
		(72) Inventor Tatsuya Mori 2-3-1 Otemachi, Chiyoda-ku, Tokyo, Japan Telegraph and Telephone Corp.	
			Continue to last page

(54) [Title of the invention] Software Detection Methods and Devices and Programs

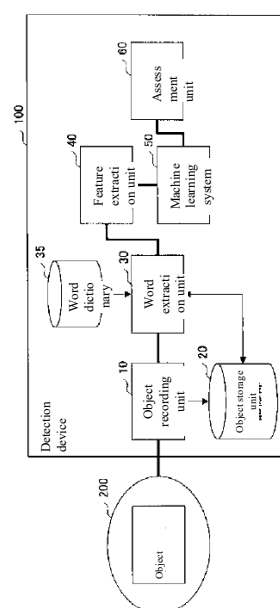
(57) [Abstract]

[Problem] Detects high-speed malware and malware-containing communications that can respond to unknown patterns.

[Means for solving] The present invention includes an Object storage means for storing Objects, a word extraction means for extracting a string of printed and readable bytes as words from Objects recorded in the Object storage means, feature extraction means for configuring feature vectors for each Object based on a set of words extracted by the word extraction means, and determination means for determining whether the Object is malware based on newly observed feature vectors by applying a machine learning system to the feature vectors.

[Representative Drawing] Figure 1

Example configuration of a system for an embodiment of the present invention



[Scope of Patent Claims]

[Claim 1]

A software detection device that detects malicious software (hereinafter referred to as “malware”) or communications that send or receive malware, with said software detection device comprised of:

An Object storage means for storing files or communication data to be transmitted or received (hereinafter referred to as “objects”);

A word extraction means for extracting, as words, byte strings that can be printed and read as character strings from an Object recorded in the Object storage means;

A feature extraction means for configuring feature vectors for each Object based on a set of words extracted by the word extraction means; and

A determination means for determining whether the Object is malware based on newly observed feature vectors by applying a machine learning system to the feature vector. 10

[Claim 2]

The machine learning system of Claim 1, wherein the machine learning system,

Allows for re-learning by providing feedback on new learning data for each configurable cycle.

[Claim 3]

The word extraction method of the previous Claim, wherein the word extraction means,

Includes a means for extracting a set of readable words from printable characters while referring to a word dictionary.

[Claim 4]

The software detection means of Claim 3, wherein the word extraction means,:

Uses a static dictionary prepared in advance or search engine results on a network as said word dictionary. 20

[Claim 5]

The software detection means of Claim 1, wherein said determination means,

Includes methods for learning and determining not only whether any malware is present, but also the type of such malware.

[Claim 6]

The software detection means of Claim 1, wherein said determination means,

Includes means for determination using all or only a part of said Object.

[Claim 7]

A word extraction step whereby a software detection method for detecting malicious software in a device having Object storage, word extraction, feature extraction, and determination means detects malicious software (hereinafter referred to a “malware”) or communications transmitting and/or receiving malware, with said word extraction means extracting a string of printed and readable bytes as words from an Object recorded in the Object storage means that stores files or communication data (hereinafter referred to as “Object”) where the word extraction means is a string; 30

A feature extraction step wherein the feature extraction means configures feature vectors for each Object based on a set of words extracted by the word extraction step; and

A determination step in which the determining means determines whether or not the Object is malware based on newly observed feature vectors by applying a machine learning system to the feature vector.

[Claim 8]

The software detection means of Claim 7,
wherein said determination step uses a machine learning system to enable re-learning by providing feedback on new learning data upon each arbitrarily configurable cycle arrives.

[Claim 9]

The software detection means of Claim 7,
wherein said word extraction step involves the use of a word dictionary to extract sets of readable words from printable characters.

[Claim 10]

The software detection means of Claim 9,
wherein said word extraction step involves use of a static dictionary prepared in advance or search engine results on a network as said word dictionary. 10

[Claim 11]

The software detection means of Claim 7,
wherein said determination step involves determining not only whether malware is present, but also the type of said malware if present.

[Claim 12]

The software detection means of Claims 7 and 11,
wherein in said determination step, determination is made using all or only part of the Object.

[Claim 13]

A program for causing a computer to function as a means for configuring a software detection device according to any one of Claims 1 to 6. 20

[Detailed description of invention]

[Technical field]

[0001]

The present invention relates to a software detection method, apparatus and program, in particular to a software detection method, apparatus and program for detecting malware or communications transmitting and/or receiving malware using supervised machine learning systems.

[Background Art]

[0002]

Malware (malicious software) called worms and botnets are becoming more and more damaging. Malware-infected computers are characterized by malicious or harmful behavior aimed at other computers connected to the network, and are used as a tool to perform malicious actions such as sending spam or denial of service attacks through unauthorized mass access to servers. Malware threats are not limited to attacks to the external environment, but, rather, may also involve extracting personal information such as credit card numbers and address books from infected computers, with these then sent to external computers. In order to prevent damage caused by such malware, it is necessary to use technology to detect the malware itself or the communication that sends and receives the malware. 30

[0003]

Malware often uses deliberately complex encryption and compression (called packing) to avoid detection by pattern matching or static analysis by inverse assemblers. Packed executable files decrypt themselves at runtime and are then deployed in memory.

[0004]

In general, in order to perform a detailed analysis of malware, packed executable files need to be decrypted using various techniques. 40

[0005]

Furthermore, methods using inverse assembly and debuggers are generally used for decrypted files (see, for example, Reference 1 of the Specific Patent).

[Prior Art Literature]

[Non-patent Literatures]

[0006]

[Non-Patent Literature 1] K. by Rozinov. Reverse Code Engineering: An In-Depth Analysis of the Bagle Virus
Bell Labs - Government communication Laboratory - Internet Research, August 2004.
http://old.idapalace.net/papers/bagle_analysis_v.1.0.pdf

[Disclosure of the invention]

[Problem to Be Solved by the Invention]

[0007]

However, methods such as the above that use decryption, disassembly, and debuggers are time-consuming and require computational resources, and so are not suitable for use in detecting communications containing malware in real time.

[0008]

As an approach to detecting whether malware is present by executing it on an actual operating system and analyzing its behavior, problems such as those described in the literature were evident. [Michiya Ito, Ryuo Neiko, Naoto Tanimoto, Makoto Iwamura, Ei Yagi, Hirohira Kawakotani, Kaokiichi, Michiaki Akiyama, Minta Nakayama, "Antimalware Technology", NTT Technical Journal March 2010 issue, pp. 40-44. <http://www.ntt.co.jp/journal/1003/files/jn201003040.html>]. Such issues included being time-consuming and requiring computational resources, meaning that timely detection and determination could not be performed.

[0009]

The present invention is directed at providing software detection methods and devices and programs that take the above points into consideration, and is capable of detecting high speed, light processing malware and malware-containing communications that can handle unknown patterns.

[Means for Solving the Problems]

[0010]

In order to resolve the above issues, the present invention (Claim 1) constitute a software detection device that detects malicious software (hereinafter referred to as "malware") or communications involving sending or receiving such malware;

An Object storage means for storing files or communication data to be transmitted or received (hereinafter referred to as "objects");

A word extraction means for extracting a string of bytes printed and readable as words from an Object recorded in the Object storage means;

A feature extraction means for configuring feature vectors for each Object based on a set of words extracted by the word extraction means; and

A determination means for determining whether the Object is malware based on newly observed feature vectors by applying a machine learning system to the feature vectors.

[0011]

In addition, the present invention (Claim 2) includes, in the machine learning system:

Relearning made possible by providing feedback on new learning data for each arbitrary configurable cycle.

[0012]

In addition, the present invention (Claim 3) includes, a means to:

extract a set of readable words from printable characters by referring to an arbitrary word dictionary.

[0013]

In addition, with the word extraction means of Claim 3 above, the present invention

(Claim 4) involves using a predetermined static dictionary or search engine results on the network.

10

20

30

40

50

[0014]

In addition, the present invention (Claim 5) includes, in the determination means, methods for determining not only whether malware is present, but also what types of malware this is, if present.

[0015]

In addition, the present invention (Claim 6) includes in the aforementioned determination means of Claims 1 or 5,

means for determination using either the whole Object or only part of it.

[0016]

In addition, the present invention (Claim 7) has Object storage means, word extraction means, feature extraction means, and determination means, and a software detection method for detecting malicious software (hereinafter referred to as “malware”) or communications that send and receive malware, with these methods consisting of:

a word extraction means that extracts as a word a string of printed and readable bytes from an Object recorded in the Object storage means that stores files or communication data to be transmitted (hereinafter the file or communication data are collectively referred to as “Object”); and

a feature extraction step wherein the feature extraction means configures feature vectors for each Object based on a set of words extracted by the word extraction step; and

a determining step that determines whether or not the Object is malware based on newly observed feature vectors by applying a machine learning system to the feature vector.

[0017]

In addition, the present invention (Claim 8) includes, in the determination step,

a machine learning system used to enable re-learning by providing feedback on new learning data upon each optionally configurable cycle.

[0018]

In addition, the present invention (Claim 9) includes, in the word extraction step,

referring to an arbitrary word dictionary to extract readable word sets from printable characters.

[0019]

In addition, the present invention (Claim 10) in the word extraction step of claim 9 above,

involves a predetermined static dictionary set up in advance or search engine results on the network to be used as a word dictionary.

[0020]

In addition, the present invention (Claim 11) includes, in the determination step,

making determinations not only about whether or not any malware is present, but also what type of malware this may be.

[0021]

In addition, with the present invention (Claim 12) in the determination step of Claims 7 and 11 above,

a determination is made using only a part of, or the entire the Object.

[0022]

In addition, the present invention (Claim 13) is a program for causing a computer to function as a means for configuring a software detection device according to any one of Claims 1 to 6.

[Effect of Invention]

[0023]

As described above, the present invention focuses on readable strings contained in software files, collects information (characteristic vectors) that characterizes malware, such as messages contained in programs that constitute malware, host names, execution command names, function names and library names used for communication, and compares this content with strings contained in files being inspected to detect malware, allowing for the timely detection of malware and communications that include malware at a high rate,.

10

20

30

40

50

[Brief Description of the Drawings]

[0024]

[Figure 1] An example of a system configuration in one embodiment of the present invention.

[Figure 2] A flowchart of character string processing in one embodiment of the present invention.

[Embodiments of the Invention]

[0025]

The present invention, in comparing malware and normal data, appropriately processes the printable character strings contained in the data and utilizes the nature of differences between the word sets when they are converted into readable word sets. Readable strings include messages contained in programs that make up malware, host names used for communication, execution command names, function names, library names, and so on, and can collect a variety of information about malware.

10

[0026]

Malware has software called variants where only part of the code has been modified. Although the majority of malware today is known to develop a large number of variants after the original is developed, most of the code of the variants is not the same as the original, so it is likely that the characteristics of the printable strings will be similar to the original.

[0027]

An overview of the present invention is shown below.

[0028]

(1) First, the file or communication data to be sent or received (hereinafter, the file or communication data is collectively referred to as the "Object") is recorded by the Object recording unit, which is described below.

20

[0029]

(2) Next, a word extraction unit, which is described below, extracts the printed and readable byte strings as words from the Object recorded above, and uses it as a word set.

[0030]

(3) Next, by configuring a feature vector for each Object based on the above word set, and applying a machine learning system to the above feature vector, the determination function determines whether the Object is malware based on newly observed feature vectors. Here, the feature vector can optionally be defined as whether a word existed or not by assigning it "0," or "1," the frequency per word, the normalized frequency per word, or the tf-idf value per word.

30

[0031]

The supervised machine learning system outlined above can be re-learned by providing feedback of new data each time an arbitrary configurable cycle arrives. Here, new data is data related to, for example, files that are determined to be malware or normal files over time, although not in real time. by methods such as prior art.

[0032]

The word extraction unit in (2) above refers to the extraction of a readable set of words from a printable string by utilizing an arbitrary word dictionary. At this time, in addition to the static dictionary provided in advance, the results of search engines on the network can also be used. That is, a printable string is searched using a search engine that can be arbitrarily specified, and is determined to be readable when the result is one or more or, in general, X or more.

[0033]

40

The determination function described above not only determines whether or not a file contains malware or is a normal file, but also determines the type of malware by machine learning that enables classification of other classes.

[0034]

The determination for an Object can also be performed using only a portion of the Object, as opposed to the entire Object. For example, in the case of communication data, the data received to date may be used to make a determination before all data is received.

50

[0035]

The embodiments of the present invention will be described in detail with the following drawings.

[0036]

Figure 1 shows a system configuration in one embodiment of the present invention.

[0037]

Detection device 100 is connected to network 200 and consists of Object recording unit 10, Object storage unit 20, word extraction unit 30, word dictionary 35, feature extract unit 40, machine learning system 50, and assessment unit 60. Here, the Object storage unit 20 and word dictionary 35 are storage media such as a hard disk.

[0038]

The Object recording unit 10 records the malware executable file (binary) or the malware to be sent/received as an Object in Object storage unit 20.

[0039]

Word extraction unit 30 retrieves the Object from Object storage unit 20, analyzes it, and extracts a string of consecutive characters with an initial printable character code of X characters or more. X can be set to any positive number (X = 1, 2, ...). Here, printable refers, for example, to the character codes of blank and graphic characters from 0x20 to 0x7e, excluding control characters, in ASCII codes. An example of the resulting string is shown in Table 1.

[0040]

[Table 1]

10

20

Table 1 Extracted printable character strings

String
.text
SUVM3
_set_app_type
L\$^L^]
ResetKeySecurity

30

The character strings shown in Table 1 are to be stored in memory (not shown) in Word extract unit 30.

[0041]

In addition, the extracted character strings are processed according to rules that can be arbitrarily set, and are candidates for words. Figure 2 shows an example of string processing.

[0042]

Step 101) Replace non-alphanumeric characters with delimiters in the string.

[0043]

Step 102) If the uppercase letters are two or more consecutive characters, go to step 103, and if not, go to step 104.

[0044]

Step 103) Mark the appropriate uppercase letter as a word.

[0045]

Step 104) If there are uppercase letters other than those marked, go to step 105, otherwise go to step 106.

50

[0046]

Step 105)

Insert a delimiter before the appropriate uppercase letter.

[0047]

Step 106)

Split the string by a delimiter and use each of the divided strings as a word.

[0048]

As a result of performing string processing according to the flow noted above, the character string shown in Table 1 above becomes the word candidate shown in Table 2.

[0049]

[Table 2]

10

Table 2

Word candidates
text
suvn
set
app
type
reset
key
security

20

The character strings shown in Table 2 are to be stored in the memory (not shown) of word extract unit 30.

30

[0050]

Word extract unit 30 uses word dictionary 35 prepared in advance for the word candidates shown in Table 2 above, extracting only those words that match the words in word dictionary 35. Alternatively, an Internet search engine is used to search for appropriate word candidates, with words extracted only when the number of search results exceeds an arbitrary configurable threshold. Matching with word dictionary 35 results in the word set shown in Table 3.

[0051]

40

50

[Table 3]

Table 3

Word candidates
text
set
app
type
reset
key
security

10

The character strings shown in Table 3 are to be stored in the memory (not shown) of word extract unit 30. [0052]

20

Feature extraction unit 40 reads out the word set obtained in Table 3 for each file from the memory of word extract unit 30 and configures the feature vector. In other words,

Feature vector of file X_i ($i=1, 2, \dots, N$)

$$V_i = \{v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{iM}\}$$

Using each word w_j ($j=1, 2, \dots, M$) in each file X_i ,

$$v_{ij} = g(w_j, X_i, X, a)$$

with this defined by the functions w_j , X_i , X , and a , where X is the set of all files and a is an optionally definable weight parameter.

[0053]

30

For example, known tf-idf values can be used as a configuration example of v_{ij} . In other words,

$$v_{ij} = tf_{ij} * idf_j$$

and

$$tf_{ij} = n_{ij} / \sum_k n_{kj}$$

$$idf_j = \log(N / c)$$

. Here,

- n_{ij} is the number of times the word w_j comes up in file F_i ;
- $\sum_k n_{kj}$ is the total number of times the word w_j comes up in all files;
- N is the total number of all files; and
- c is the total number of files containing the word w_j

40

. Note that $*$ is an operator that indicates multiplication.

[0054]

Assessment unit 60 applies machine learning system 50 to the feature vector V_i .

[0055]

By applying supervised machine learning system 50 to the feature vector V_i , the file is classified as “malware” and “normal file.” Specifically, each “malware” and “normal file” file feature vector is learned in advance, and the newly obtained file is determined to be “malware” or “normal file” based on that feature vector.

50

[0056]

As a method of machine learning for machine learning system 50, known supervised learning methods can be used. In supervised learning, teacher data (V₁, y₁), (V₂, y₂), ..., (V_N, y_N) is given in advance, with V_i serving as a feature vector for said file X_i. y_i is of the same file class and has a value of "0" or "1." "0" is usually defined as a file, and "1" is defined as malware.

[0057]

The supervised learning noted above is a highly accurate method for estimating class y_m for the newly observed V_m, with a variety of known methods widely used, including support vector machines (SVMs), naive Bayesian methods, neural networks, and decision trees.

[0058]

While the results of applying the SVM as supervised learning are illustrated below, the scope of machine learning in the present invention is not limited to this example, with the various methods described above also being applicable.

[0059]

Table 4 is the result of analyzing the words that appear in each file as well as the word counts for each file in machine learning system 50, and is stored in the memory (not shown) of assessment unit 60.

[0060]

[Table 4]

Table 4

File name (hash value)	Appearance of word: number of times
00c4fc2fd0ac1d2b2fa43d8c5a087104	dll:4,get:2,sip:2,kernel:2,address:1,change:1,...
013a5ba10e3fc8a039b045530381d957	dll:4,get:2,kernel:2,rhd:1,bde:1,amu:1,range:1,...
notepad.exe	get:47>window:20,set:19,file:16,message:13,...
.....	

Table 5 is an example of configuring feature vectors based on Table 4 above. Words are numbered based on word dictionary 35. The character vector is called "j: v_{ij}." In other words, now we have {word number: tf-idf value}. Here, the class is "1" and malware, is "-1" as a normal file.

[0061]

[Table 5]

Table 5

File ID	Feature Vector	Class
1	119:0.0008, 196:0.0077, 307:0.0077,...	-1
2	119:0.0024, 168:0.0051, 213:0.0382,...	-1
3	84:0.006, 94:0.002, 113:0.00258,...	+1
....		

10

20

30

40

50

An example of the results of applying SVM to the above feature vector is shown in Table 6. Here we used 221 malware files, and 1,526 normal files. Malware was collected from the actual Internet, and files were usually installed on commercial operating systems.

[0062]

In addition, 100 files each were used for learning, and all remaining files were adjudicated. In other words, machine learning was performed using the known $100+100=200$ files, and judgments were made on the unknown 1,547 files that excluded those 200 files. The results obtained are highly accurate, with a precision rate of 99.37% and recall of 99.86%.

[0063]

[Table 6]

10

Actual	Estimated	Number of determinations
+1	+1	1424
-1	+1	9
+1	-1	112
-1	-1	2

20

As in the example above, not only can we determine whether the subject is malware or a normal file, but also use machine learning to classify other classes of malware and determine the malware type. That is, by using the malware type c_k ($k=1, 2, \dots, K$) as a class and applying machine learning to a given feature vector for problem setting the determination of class C_k , such classification is possible.

[0064]

Machine learning system 50 can also perform re-learning via feedback of new data upon each arbitrary configurable cycle. Here, new data includes data about files that are not real-time, such as through prior art, or that, over time, have been determined to be malware or normal files.

[0065]

Determinations regarding the Object described above may be arrived at using only a portion of the Object, as opposed to all of it. For example, in the case of communication data, the data received to date is used to make a determination before all data is received.

30

[0066]

As described above, the present invention can detect malware by focusing on the readable strings contained in software files, collecting information (feature vectors) that characterizes malware, such as messages contained in programs that make up malware, host names, execution command names, function names, library names, and so on, comparing this content with strings contained in the files being inspected.

[0067]

The operation of each component of the detection device shown in FIG. 1 above can be programmatically constructed and then installed and executed on a computer utilized as the detection device, or distributed over a network.

40

[0068]

In addition, the built program can be stored on a hard disk, flexible disk, CD-ROM, or other portable storage medium, and installed on a computer or distributed.

[0069]

50

It should be noted that the present invention is not limited to the above embodiments, and various changes and applications are possible within the scope of the claims.

[Explanation of References]

[0070]

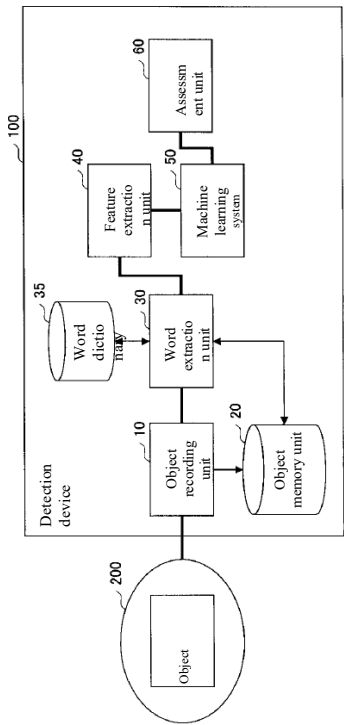
- 10 Object recording unit
- 20 Object storage unit
- 30 Word extraction unit
- 35 Word dictionary
- 40 Feature extraction unit
- 50 Machine learning systems
- 60 Assessment unit
- 100 Detection device
- 200 Network

10

20

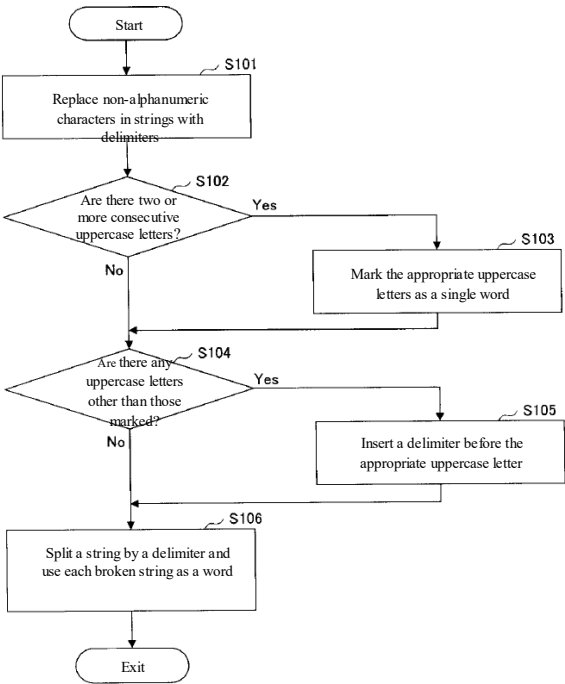
[Figure 1]

Example of a system configuration in the embodiment of the present invention



[Figure 2]

Flow chart of string processing in the embodiment of the present invention



30

40

50

Continued from the front page

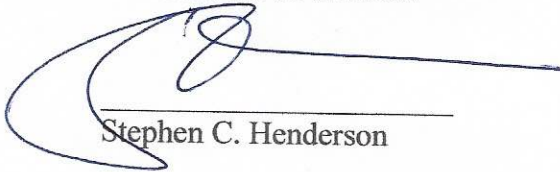
(72) Inventor Kazuhiro Tobe
1-104 Totsuka-cho, Shinjuku-ku, Tokyo Waseda University
(72) Inventor Akihiro Shimoda
1-104 Totsuka-cho, Shinjuku-ku, Tokyo Waseda University
(72) Inventor Shigeki Goto
1-104 Totsuka-cho, Shinjuku-ku, Tokyo Waseda University
F term (reference) 5B276 FD08

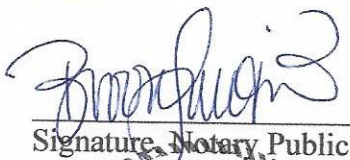
City of New York, State of New York, County of New York

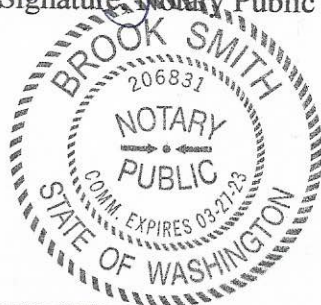
Translation Certificate

I, Stephen C. Henderson, declare under penalty of perjury under the laws of the United States of America that that I am competent to translate from Japanese to English, and verify that the document JP2012027710A is true and accurate, and that the foregoing is true and correct.

Executed on 11/16/2022



Stephen C. Henderson

Signature, Notary Public

Stamp, Notary Public

LANGUAGE AND TECHNOLOGY SOLUTIONS FOR GLOBAL BUSINESS
1250 BROADWAY, 32ND FLOOR, NEW YORK, NY 10001 | T 212.689.5555 | F 212.689.1059 |
WWW.TRANSPERFECT.COM
OFFICES IN 90 CITIES WORLDWIDE