(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5507699号
(P5507699)
(45)発行日　平成26年5月28日(2014.5.28)　(24)登録日　平成26年3月28日(2014.3.28)

(51)Int.Cl.　　F I
G06F 21/56 (2013.01)　　G06F 21/00 156G
G06F 21/44 (2013.01)　　G06F 21/20 144D

請求項の数 13　(全 11 頁)

(21)出願番号　特願2012-537803(P2012-537803)
(86)(22)出願日　平成22年11月1日(2010.11.1)
(65)公表番号　特表2013-510361(P2013-510361A)
(43)公表日　平成25年3月21日(2013.3.21)
(86)国際出願番号　PCT/KR2010/007608
(87)国際公開番号　WO2011/055945
(87)国際公開日　平成23年5月12日(2011.5.12)
審査請求日　平成24年6月22日(2012.6.22)
(31)優先権主張番号　10-2009-0105344
(32)優先日　平成21年11月3日(2009.11.3)
(33)優先権主張国　韓国(KR)

(73)特許権者　512117199
アンラブ，インコーポレイテッド
大韓民国、ギョンギ－ド　４６３－４００、ソンナム－シ、ブンダン－グ、パンギョヨク－ロ（サムピョン－ドン）　２２０
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100088683
弁理士　中村　誠
(74)代理人　100109830
弁理士　福原　淑弘
(74)代理人　100075672
弁理士　峰　隆司
(74)代理人　100095441
弁理士　白根　俊郎

最終頁に続く

(54)【発明の名称】悪性サイト検出装置及び方法

(57)【特許請求の範囲】
【請求項１】
悪性サイト検出装置において、コンピューティング装置内で実行されるあらゆるプロセスを監視する監視部と、
前記監視部によりブラウザの実行が検知された場合、前記ブラウザで実行されるプロセスにフックコードを挿入するフックコード挿入部と、
ウェブサイトの移動が検知される場合、前記フックコードを利用して前記ウェブサイトの移動に応じて発生するプロセスのスタック構造を検査、前記スタック構造を検査したか否かを判断し、前記移動したウェブサイトが悪性サイトであるか否かを判断する危険値判断部と、
前記判断された悪性サイトのリストが格納されるデータベースとを含む悪性サイト検出装置。
【請求項２】
前記フックコードは第１フックコードと第２フックコードを含み、前記フックコード挿入部は前記ブラウザで実行されるプロセスの実行進入点に前記第１フックコードを挿入し、前記ブラウザで実行されるプロセスの実行中間点に前記第２フックコードを挿入するように構成され、
前記危険値判断部は、前記第１フックコードを利用して前記ウェブサイトの移動に応じて発生するプロセスのスタック構造を検査するプロセス判断部と、前記第２フックコードを利用して前記第１フックコードを利用したスタック構造を検査したか否かを確認するフ

ックコード実行如何判断部とを含むことを特徴とする請求項１に記載の悪性サイト検出装置。

【請求項３】

前記プロセス判断部は、前記第１フックコードを利用して前記ウェブサイトの移動に応じて発生するプロセスのプログラム内に認証書が存在するか否かを確認し、

前記プログラム内に認証書がない場合に、特定プロセスのスタックに含まれるＤＬＬファイルの製造会社情報と、各ウェブサイトの固有プロセス及び前記固有プロセス内の固有コールスタック構造情報を含むスタック構造情報に基づいて前記スタック構造を検査するように構成されたことを特徴とする請求項２に記載の悪性サイト検出装置。

【請求項４】

前記危険値判断部は、前記スタック構造の検査完了後、検査完了の標識を該当プロセス内に挿入し、前記スタック構造の検査結果、前記スタック構造が異常であると判断される場合、前記移動したウェブサイトを悪性サイトと判断するように構成されたことを特徴とする請求項３に記載の悪性サイト検出装置。

【請求項５】

前記フックコード実行如何判断部は、前記第２フックコードを利用して前記検査完了標識の存否を検査し、前記検査完了の標識が存在しない場合、前記移動したウェブサイトを悪性サイトと判断するように構成されたことを特徴とする請求項４に記載の悪性サイト検出装置。

【請求項６】

前記フックコード判断部は、既設定プロセス関数を用いて前記第１フックコード及び前記第２フックコードを利用した２つの検査の完了如何を確認し、前記両検査のうちいずれか１つでも行われていない場合、前記移動したウェブサイトを悪性サイトと判断するよう構成されたことを特徴とする請求項２に記載の悪性サイト検出装置。

【請求項７】

悪性サイト検出装置において、

コンピューティング装置内で実行されるあらゆるプロセスを監視する中でブラウザの実行が検知される場合、前記ブラウザで実行されるプロセスにフックコードを挿入する段階と、

ウェブサイトの移動が検知される場合に前記フックコードを利用してウェブサイトの移動に応じて発生するプロセスのスタック構造を検査し、前記スタック構造を検査したか否かを判断する段階と、

移動したウェブサイトが悪性サイトであるか否かを判断する段階と、

判断された悪性サイトをデータベースに格納する段階とを含む悪性サイト検出方法。

【請求項８】

前記フックコードは第１フックコードと第２フックコードを含み、前記フックコードを挿入する段階は、

前記ブラウザで実行されるプロセスの実行進入点に前記第１フックコードを挿入する段階と、

前記ブラウザで実行されるプロセスの実行中間点に前記第２フックコードを挿入する段階を含み、

前記移動したウェブサイトが悪性サイトであるか否かを判断する段階は、

前記第１フックコードを利用して前記スタック構造を検査する段階と、

前記第２フックコードを利用して前記第１フックコードを利用したスタック構造を検査したか否かを確認する段階とを含むことを特徴とする請求項７に記載の悪性サイト検出方法。

【請求項９】

前記移動したウェブサイトが悪性サイトであるか否かを判断する段階は、

前記スタック構造を検査する前に前記ウェブサイトの移動に応じて発生するプロセスのプログラム内に認証書が存在するか否かを確認する段階を更に含み、

前記プログラム内に認証書が存在しない場合に、特定プロセスのスタックに含まれるＤＬＬファイルの製造会社情報と、各ウェブサイトの固有プロセス及び前記固有プロセス内の固有コールスタック構造情報を含むスタック構造情報に基づいて前記スタック構造を検査することを特徴とする請求項８に記載の悪性サイト検出方法。

【請求項１０】

前記移動したウェブサイトが悪性サイトであるか否かを判断する段階は、

前記スタック構造の検査完了後、検査完了の標識を前記プロセス内に挿入する段階と、

前記スタック構造の検査結果、前記スタック構造が異常であると判断される場合、前記移動したウェブサイトを悪性サイトと判断する段階とを更に含むことを特徴とする請求項９に記載の悪性サイト検出方法。

【請求項１１】

前記移動したウェブサイトが悪性サイトであるか否かを判断する段階は、

前記第２フックコードを利用して前記検査完了標識の存否を検査する段階と、

前記検査完了の標識が存在しない場合、前記移動したウェブサイトを悪性サイトと判断する段階とを更に含むことを特徴とする請求項１０に記載の悪性サイト検出方法。

【請求項１２】

前記移動したウェブサイトが悪性サイトであるか否かを判断する段階は、

既設定プロセス関数を用いて前記第１フックコード及び前記第２フックコードを利用した両検査の完了如何を確認する段階と、

前記両検査のうち、いずれか１つでも行われていない場合に前記移動したウェブサイトを悪性サイトと判断する過程とを更に含むことを特徴とする請求項１１に記載の悪性サイト検出方法。

【請求項１３】

請求項７～請求項１２のいずれか一項の悪性サイト検出方法を行うコンピュータプログラムが記録された記録媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明はコンピュータ及びユーザー機器に存在する悪性プログラムを検知する技術に関し、特に、コンピュータ及びユーザー機器におけるブラウザでプロセスを実行した時点で悪性サイトであるか否かを判断するに適した悪性サイト検出装置及び方法に関するものである。

【背景技術】

【０００２】

一般的に、セキュリティソフトベンダー或いは関連セキュリティ企業により分析されていない新種の悪性プログラム及びマルウェアはセキュリティソフトがインストールされているコンピューティング装置であっても、該当コンピューティング装置を感染させてしまう。特に、ユーザーがコンピューティング装置を通じてインターネットに接続した状態で特定のウェブサイトにアクセスするだけでもマルウェア（例えば、脆弱性攻撃（Ｅｘｐｌｏｉｔ）コード）に感染し得る。

【０００３】

これは攻撃者が該当ウェブサイトをハッキングしてそのサイトにアクセスした人々にマルウェアを配布したり、攻撃者が直接ウェブサイトを運営してマルウェアを配布しているためである。

【０００４】

また、攻撃者は検索エンジンを通じて攻撃の対象となるウェブページを自動で収集し、実際の攻撃のために自動化したＳＱＬインジェクション攻撃ツールを用いる。最近用いられているＳＱＬインジェクション攻撃は、特定のウェブサイトにマルウェアリンクを密かに隠しておき、アクセス者を感染させる。このようなＳＱＬインジェクション攻撃ツールは検索エンジンで脆弱性サイトを検索した後、脆弱性サイトに実際にＳＱＬインジェクシ

ョン攻撃を行ってＤＢに悪意あるスクリプト（ｓｃｒｉｐｔ）ファイルを連結するスクリプトを埋め込む。

【０００５】

ユーザーが悪性スクリプトが埋め込まれているウェブサイトに接続する際、ウェブサーバーはＤＢからユーザーに示すコンテンツを呼び出すが、埋め込まれているスクリプトも同時に呼び出す。ウェブブラウザを通じてスクリプトが実行された後、攻撃者のウェブサーバーからもう１つの悪性スクリプトを呼び出して実行する。この悪性スクリプトにはユーザーのコンピューティング装置を攻撃する脆弱点攻撃コード（Ｅｘｐｌｏｉｔ Ｃｏｄｅ）が搭載されており、セキュリティレベルの低いユーザーのコンピューティング装置にマルウェアをインストールして、重要な情報を流出する。

【０００６】

ただし、このようなＳＱＬインジェクション攻撃に備えるために、ウェブサイトに対するリアルタイム検査及び有害性の有無の検討を通じてマルウェアの流入経路を遮断する方式に対する多くの研究がなされている。

【０００７】

従来技術において、特定のサイトで悪性ページか否かを判断する方式では、悪性パターンがスクリプトにあるかどうかを検査したり、接続したサイトのページ構造が悪性ページと類似しているか否かだけを検査するため、特定の悪性パターンが存在しなかったり、新規悪性プログラムに対しては検査を実行しても検出されないこともあり得るという問題点があった。関連する先行技術が下記の特許文献１に開示されている。

【先行技術文献】

【特許文献】

【０００８】

【特許文献１】韓国公開特許第２００９-００３４６４８号（公開日：２００９年４月８日）

【発明の概要】

【発明が解決しようとする課題】

【０００９】

そこで、本発明は上記事情を鑑みてなされたものである。その目的は、悪性サイトからダウンロードしたプログラムの最終目的であるプロセスの実行時に一般的なプロセスの実行と異なる点を捕捉して悪性サイトであるか否かを判断できる悪性サイト検出装置及び方法を提供することにある。

【００１０】

また、本発明の他の目的は、特定のサイトからダウンロードしたプログラムのプロセスの実行時点で該当プロセス内に認証書が含まれているか、スタック構造が正常か否かを確認して、現在のサイトが危険なサイトであるか或いは現在コンピュータで実行しているプロセスが異常であるかを判断できる悪性サイト検出装置及び方法を提供することにある。

【課題を解決するための手段】

【００１１】

本発明の態様による悪性サイト検出装置は、コンピューティング装置内で実行されるあらゆるプロセスを監視する監視部と、前記コンピューティング装置内のブラウザが実行された場合に前記プロセスにフックコードを挿入し、スタック構造を検査し、前記スタック構造を検査したか否かを判断して悪性サイトを判断する危険値判断部、前記判断された悪性サイトのリストが格納されるデータベースを含む。

【００１２】

本発明の他の態様による悪性サイト検出方法は、コンピューティング装置内で実行されるあらゆるプロセスを監視する段階と、前記コンピューティング装置内のブラウザが実行される場合に前記プロセスにフックコードを挿入しスタック構造を検査し前記スタック構造を検査したか否かを判断して悪性サイトを判断する段階、判断された悪性サイトのリストをデータベースに格納する段階とを含む。

【発明の効果】
【００１３】
本発明の実施形態による悪性サイト検出装置及び方法により、実際のブラウザ上で実行されるスクリプトパターン及びマルウェアを探知して迅速な分析及び駆除を可能にし、まだ広く知られていない新規マルウェアを検出できるので、悪性サイトを容易に区別できるという効果を奏する。
【図面の簡単な説明】
【００１４】
【図１】コンピューティング装置と連結されるウェブサービス装置を示す図である。
【図２】本発明の実施形態による悪性サイト検出装置の構造を示すブロック図である。
【図３】本発明の実施形態による悪性サイト検出装置の動作手順を示すフローチャートである。
【発明を実施するための手段】
【００１５】
以下、本明細書の一部をなす添付の図面を参照して本発明の実施形態を詳細に説明する。
【００１６】
図１は、コンピューティング装置と連結されるウェブサービス装置を示す図である。
【００１７】
図１を参照すると、コンピューティング装置１００は有無線通信網１０２を介してウェブサービス装置１０４と接続されている。このとき、コンピューティング装置１００とウェブサービス装置１０４は、クライアント端末とサーバー関係になり得、少なくとも１つのコンピューティング装置１００が有無線通信網１０２に連結され、少なくとも１つのウェブサービス装置１０４と相互間で連結され得る。
【００１８】
コンピューティング装置１００は例えば、パソコン（ＰＣ）、ノートブックパソコンと、ＰＭＰ（Ｐｏｒｔａｂｌｅ Ｍｕｌｔｉｍｅｄｉａ Ｐｌａｙｅｒ）、ＰＤＡ（Ｐｅｒｓｏｎａｌ Ｄｉｇｉｔａｌ Ａｓｓｉｓｔａｎｔｓ）、携帯電話、スマートフォンを含むユーザー機器などのような有無線通信網１０２に接続してウェブサービス装置１０４で提供するウェブサイトにアクセス可能なあらゆる電子機器を含み得る。
【００１９】
ユーザーがコンピューティング装置１００において、ブラウザを実行してウェブサービス装置１０４のウェブサイトに接続すると、ウェブサービス装置１０４はユーザーが要請するウェブページ情報をユーザーのコンピューティング装置１００に提供する。
【００２０】
ここでいうウェブページは、動的ウェブページ、静的ウェブページなどを含む。動的ウェブページとは、株式情報、天気予報、掲示板などのようにコンテンツが動的に登録、修正、削除されるページを意味し、このような動的ウェブページは、ＡＳＰ（Ａｃｔｉｖｅ Ｓｅｒｖｅｒ Ｐａｇｅ）ソース、ＰＨＰ（ｐｒｏｆｅｓｓｉｏｎａｌ ＨＴＭＬ ｐｒｅｐｒｏｃｅｓｓｏｒ）ソースなどの形態で示す。静的ウェブページは動的ウェブページとは異なり、ページ生成時にコンテンツが固定されたページを意味し、例えば、ＨＴＭＬ（ｈｙｐｅｒｔｅｘｔ ｍａｒｋｕｐ ｌａｎｇｕａｇｅ）ソースなどの形態で示す。
【００２１】
そして、このようなＨＴＭＬソースにはスクリプトが含まれ得、スクリプトには悪性パターン又は悪性スクリプトがリンクされ得る。悪性スクリプトが埋め込まれているウェブページがユーザーに提供されるとき、該当ウェブページのＨＴＭＬにリンクされている悪性スクリプトがユーザーのコンピューティング装置１００で実行される。
【００２２】
これにより、悪性スクリプトは特定のプログラム及びプロセスを実行するようになり、ユーザーのコンピューティング装置１００で実行されるプロセスを用いて情報流出、ファ

イル感染、システム破壊などの攻撃を行う。

【００２３】

コンピューティング装置１００で実行される悪性スクリプトとして脆弱性攻撃コードの最終目的は、悪性プログラムのダウンロードとの実行にある。実際にブラウザを用いて多くのファイルがダウンロードされるが、実際に実行されるプログラムは限定されている。正常なプロセスを実行する場合にはプロセスが同一のスタック構造を有するようになるので、コンピューティング装置１００ではこのような事項を参照してプロセスの実行時点でプロセスに認証書が含まれているか、又はスタック構造が正常か否かを確認する。これを通じて、コンピューティング装置１００は、現在のウェブサイトが悪意あるウェブサイトであるか、或いは現在のコンピューティング装置１００で実行したプロセスが異常か否かの判断が可能になる。

【００２４】

図２は、本発明の実施形態による悪性サイト検出装置の構造を示すブロック図である。

【００２５】

図２を参照すると、悪性サイト検出装置２００は１つのコンピューティング装置１００であるか、又はコンピューティング装置１００内に含まれる装置であり、プロセス実行監視部２０２、フックコード挿入部２０４、危険値判断部２０６、危険ＵＲＬリストＤＢ２１２を含む。

【００２６】

具体的に、プロセス実行監視部２０２はコンピューティング装置１００が駆動されれば駆動時から実行されるプロセスを周期的にモニタリングする。

【００２７】

ここでいうプロセスとは、プログラムがメモリに積載されて実行されている状態をいう。あらゆるプログラムは実行時に少なくとも１つのプロセスを有する。１つのプロセスには複数の命令語とカウント、ＣＰＵレジスタ、そしてルーチン因子、復帰アドレス、格納されている変数、関数などを入れたスタックが含まれている。このような各プロセスは固有の権限と責任のもと相互通信を行いシステムで動作中である１つのプロセスが誤った演算を行いエラーが発生しても他のプロセスは正常に作動する。

【００２８】

プロセス実行監視部２０２はコンピューティング装置１００でブラウザの実行だけでなくブラウザが実行された後、特定のウェブサイトに接続することにより発生するあらゆるプロセスをリアルタイムで監視するようになる。プロセス実行監視部２０２でウェブページ、例えば、ハイパーテキスト文書を示すブラウザが実行されたことを検知すると、フックコード挿入部２０４はブラウザで実行されるプロセスの実行進入点に第１フック（Ｈｏｏｋ）コードを挿入し、プロセスの実行中間点に第２フックコードを挿入する。挿入されたフックコードはその後、ユーザーが他のウェブサイトに移動することにより生成するプロセスの実行時に共に実行されて該当プロセスを検査する。

【００２９】

第１フックコードは実行されるプログラムに認証書が含まれているかと、プロセスが異常なスタック構造の有無を判断するためのものであり、第２フックコードは第１フックコードによる検査の実行有無を判断するためのものである。実行進入点及び中間点はプロセスの既設定される位置であり、実現方式によってその位置が変わり得る。特に、中間点は第１フックコードによる検査の完了確認が可能な位置であれば、どの位置でも可能である。危険値判断部２０６は、フックコード挿入部２０４で挿入したフックコードを利用してユーザーが移動するサイトが危険であるか否かを判断するためのものであり、プロセス判断部２０８と、フックコード実行如何判断部２１０とを含む。

【００３０】

プロセス判断部２０８は第１フックコードを利用してプロセスを検査するものであり、ユーザーがウェブサイトを移動する際に、これにより実行されるプログラムが認証書を含むか否かを確認する。プログラムが公認認証機関からプログラムに対する公認認証を受け

た場合、該当プログラム内に認証書が挿入されているので、プログラムの実行時に該当認証書を含むか否かの確認を通じて該当プログラムのプロセスは正常であると判断できる。

【００３１】

また、このような場合には後述する第１フックコードを利用したスタック構造の検査、及び第２フックコードを利用した第１フックコードの実行如何検査の手順を省略でき、これらのフックコードを利用した不要な検査を防止できる。

【００３２】

更に、プロセス判断部２０８は、第１フックコードを利用して実行されるプロセスのスタック構造を検査して該当スタック構造が異常なスタック構造であるか否かを判断する。スタックには一種の検査エンジンのようにパターン情報が格納されているので、開始スタックとスタックに含まれているＤＬＬ（Ｄｙｎａｍｉｃ Ｌｉｎｋ Ｌｉｂｒａｒｙ）などの製造会社の確認（例えば、マイクロソフト、アドビアクロバットなど）を通じてスタック構造の異常の有無が分かる。下記（表１）のようにマルウェアで実行されたプロセスのスタック構造は、特定名からなっているＤＬＬファイルが繰り返されていることが分かる。

【表１】

- 脆弱性攻撃（Ｅｘｐｌｏｉｔ）コードで実行されたプロセススタック構造

kernel32.dll|SHELL32.dll|OLEAUT32.dll|SHELL32.dll|msadco.dll|vbscript.dll
kernel32.dll|ntdll.dll|kernel32.dll|ntdll.dll

- 一般的なプロセススタック構造

kernel32.dll|IEShims.dll|AFCSTA~1.OCX|MFC42.DLL|JScript.dll|MFC42.DLL
kernel32.dll|SHELL32.dll|npkcx.ocx|MFC42.DLL|npkcx.ocx|USER32.dll
kernel32.dll|suipre.dll|aosmgr.ocx|MFC42.DLL|jscript.dll|USER32.dll
kernel32.dll|SHELL32.dll|IssacWebSE2.dll|mshtml.dll|jscript.dll|mshtml.dll

【００３３】

また、特定プロセス毎に特定コールスタック構造の情報を含んでいるので、プロセス判断部２０８は該当スタック構造の異常の有無を判断した後、該当スタック構造が該当プロセスとマッチしない場合には悪性サイトとして該当サイトのアドレス（Ｕｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒ、以下ＵＲＬという）を危険ＵＲＬと判断する。この危険ＵＲＬ情報は危険ＵＲＬリストＤＢ２１２に格納されるか、コンピューティング装置１００に又は有無線通信網１０２と連動し、危険ＵＲＬリストを管理する管理サーバーがある場合には管理サーバーに送信される。危険ＵＲＬと判断された該当サイトのプログラムは遮断されて駆除され得る。プロセス判断部２０８は、スタック構造を検査した後、スタック構造の検査完了を知らせる既に設定された標識を該当プロセス又はスタック構造に挿入する。

【００３４】

一方、各ウェブサイトにある特定のプロセスが特定のコールスタック構造を含むという情報は、モニタリングサービスのためのクライアントやクライアント-サーバモデルで適用可能である。即ち、モニタリングサービスのためのクライアントは関連情報をエンジンの形態で有しているが周期的にチェックでき、クライアント-サーバーはクライアントで特定のサイト内に既に報告されていた特定のプロセスのコールスタック構造と異なる情報が収集されるか否かを検査する。

【００３５】

フックコード実行如何判断部２１０は、第２フックコードを利用して第１フックコード

が実行されたか否かを検査する。第１フックコードの実行如何は、プロセス判断部２０８が第１フックコードでスタック構造を検査した後は既に設定された標識を該当プロセス又はスタック構造に挿入するようになるので、標識が残っているかを確認することで分かる。標識が残っている場合には第１フックコードが実行されて検査が完了したものであり、該当サイトは正常であると判断できる。

【００３６】

しかし、標識が残っていない場合には、悪性コードにより第１フックコードが代替、変更及び削除された可能性があるので、この際には該当ＵＲＬを危険ＵＲＬと判断して危険ＵＲＬリストＤＢ２１２に格納するか、管理サーバーにこれを伝達する。

【００３７】

一方、危険値判断部２０６ではＥｎｕｍＰｒｏｃｅｓｓのような関数を利用して第１及び第２フックコードを利用したスタック構造の検査及び標識検査がいずれも行われたかどうかを判断する。これにより、両検査がいずれも行われた場合には正常なプロセスであると判断できるが、少なくとも１つの検査が行われていない場合には該当プロセスが実行されたウェブサイトのＵＲＬを危険ＵＲＬと判断する。

【００３８】

図３は、本発明の実施形態による悪性サイト検出装置の動作手順を示すフローチャートである。

【００３９】

図３を参照すると、悪性サイト検出装置２００内のプロセス実行監視部２０２はコンピューティング装置１００が実行後に実行されるプロセスを周期的に確認する。この段階でプロセス実行監視部２０２がユーザーによるブラウザの実行を検知する場合（３００段階）、フックコード挿入部２０４はブラウザで実行されるプロセスの実行進入点に第１フックコードを挿入し（３０２段階）、プロセスの実行中間点に第２フックコードを挿入する（３０４段階）。

【００４０】

ユーザーによるウェブサイトの移動が検知される場合、危険値判断部２０６内のプロセス判断部２０８は第１フックコードを利用してウェブサイトの移動に応じて実行されるプロセスを検査する。まず、３０６段階において第１フックコードでプログラム内に認証書が存在するか否かを判断し、認証書が存在する場合には正常なプロセスであるので悪性サイトの検出手順を終了する。しかしながら、プログラム内に認証書が存在しない場合、３０８段階において第１フックコードでプロセスのスタック構造を検査しその異常有無を判断する。プロセス判断部２０８には正常なプロセススタック構造との比較ができるように開始スタックとスタックに含まれているＤＬＬファイルなどの製造会社の確認を可能にする情報と、サイト別の特定プロセス及び特定のコールスタック構造情報が格納されている。３１０段階でプロセス判断部２０８の前記検査が完了した場合、該当プロセス又はスタック構造内に検査完了の標識を挿入する。

【００４１】

その後、３１２段階で検査されたスタック構造に異常がある場合は３１４段階に進み、ユーザーが移動したウェブサイトを悪性サイトと判断し、該当ＵＲＬを危険ＵＲＬに分類して危険ＵＲＬリストＤＢ２１２に格納するか、管理サーバーに伝達する。

【００４２】

ただし、３１２段階で検査されたスタック構造に異常がない場合又は検査が行われていない場合には３１６段階に進み、フックコード実行如何判断部２１０は第２フックコードによる標識検査を行う。標識検査は第１フックコードによる検査実行有無を判断するためのものであり、標識が残されていない場合にはマルウェアにより第１フックコードが代替、変更及び削除されたと判断できる。

【００４３】

従って、３１８段階で標識が存在しないと判断された場合には３１４段階に進んで該当サイトを悪性サイトと判断し、このサイトのＵＲＬを危険ＵＲＬに分類して危険ＵＲＬリ

ストＤＢ２１２に格納するか、管理サーバーに伝達する。

【００４４】

しかし、３１８段階で標識が存在すると判断された場合には３２０段階に進み、ＥｎｕｍＰｒｏｃｅｓｓ関数により周期的にブラウザで実行されるプロセスがあるとチェックされると、第１及び第２フックコードを利用した検査の実行有無を判断する。これにより、両検査の過程がいずれも行われたと判断された場合、正常なプロセスと判断して終了するが、いずれか１つでも行われていない場合には３１４段階に進み、該当サイトのＵＲＬを危険ＵＲＬに分類して危険ＵＲＬリストＤＢ２１２に格納するか、管理サーバーに伝達する。

【００４５】

以上説明した通り、本発明の実施形態による悪性サイト検出装置及び方法は、悪性サイトでダウンロードしたプログラムのプロセス実行時に一般的なプロセスの実行と異なる点を捕捉して悪性サイトであるか否かを判断するものであり、プロセスの実行時点でプログラムに認証書が含まれているか、スタック構造が正常か否かを確認して現在のサイトが危険な悪性サイトであるか、或いは現在のコンピュータで実行したプロセスが異常か否かの判断を可能にする。

【００４６】

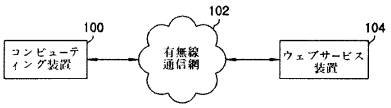
本発明の範疇内の実施形態は、コンピュータ実行可能な命令語又はデータ構造を格納するコンピュータ読み取り可能な媒体を含むことができる。前記コンピュータ読み取り可能な媒体は一般的な機能又は特別な機能をするコンピュータにより制御可能な媒体であり得る。前記コンピュータ読み取り可能媒体は、例えば、ＲＡＭ（Ｒａｎｄｏｍ-Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、ＲＯＭ（Ｒｅａｄ Ｏｎｌｙ-Ｍｅｍｏｒｙ）、ＥＥＰＲＯＭ（Ｅｌｅｃｔｒｉｃａｌｌｙ Ｅｒａｓａｂｌｅ Ｐｒｏｇｒａｍｍａｂｌｅ Ｒｅａｄ-Ｏｎｌｙ Ｍｅｍｏｒｙ）、ＣＤ-ＲＯＭ及び光ディスク、マグネチックディスク及びマグネチック格納装置などを含む。また、前記コンピュータ読み取り可能媒体は、前記例を上げた媒体以外に、コンピュータ実行可能な命令又はデータ構造の形態におけるプログラムコードを格納できる媒体も含むことができる。

【００４７】

以上、本発明の好適な実施形態が説明されたが、本発明はこれらの特定の実施形態に限定されず、後続する請求範囲の範疇から逸脱することなく、多様な変更及び変形がなされ得、それも本発明の範疇内に属すると言える。

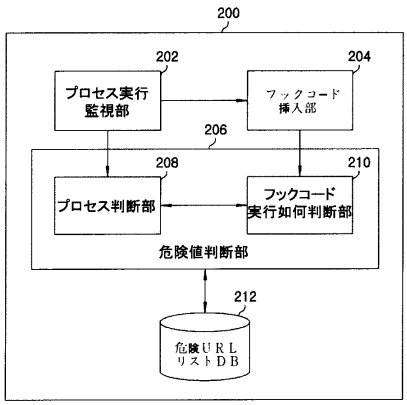
【図１】

図 1



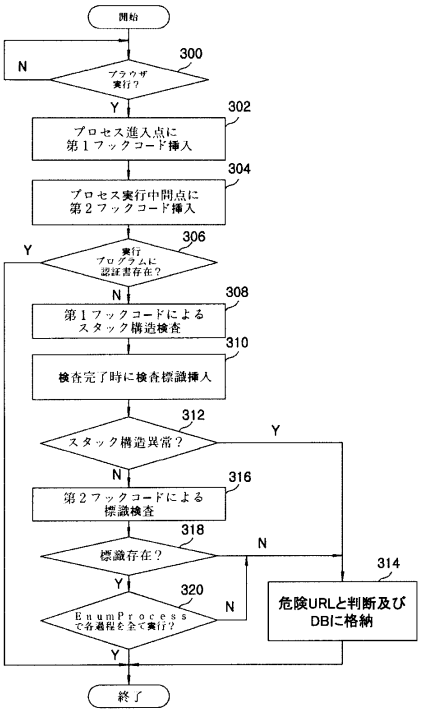
【図２】

図 2



【図３】

図 3



フロントページの続き

(74)代理人 100103034
弁理士 野河 信久
(74)代理人 100140176
弁理士 砂川 克
(72)発明者 オー、ジュ・ヒュン
大韓民国、ソウル １５０－０７３、ヨンドゥンポーグ、デリム ３－ドン ６５２、スンウォン・アパートメント １０１－１３０１
(72)発明者 リー、チャン・ウォ
大韓民国、ソウル １２１－８０２、マポーグ、ゴンドク ２－ドン １７５－１００ ２０３
(72)発明者 パク、チョン・フィル
大韓民国、ソウル １５６－０３１、ドンジャクーグ、サンド １－ドン ５２２ ３１４

審査官 木村 励

(56)参考文献 特開２００３－３３７７９７（ＪＰ，Ａ）
特開平０７－０９３１８４（ＪＰ，Ａ）
国際公開第２００４／０７５０６０（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／３０ － ２１／４６
Ｇ０６Ｆ ２１／５０ － ２１／５７