(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6253333号
(P6253333)
(45)発行日　平成29年12月27日(2017.12.27)　(24)登録日　平成29年12月8日(2017.12.8)

(51)Int.Cl.　　F I
G06F 9/445 (2006.01)　　G06F 9/06 650C
G06F 11/00 (2006.01)　　G06F 9/06 610L
G06F 9/06 630B

請求項の数 17　(全 23 頁)

(21)出願番号　特願2013-211424(P2013-211424)
(22)出願日　平成25年10月8日(2013.10.8)
(65)公開番号　特開2014-96143(P2014-96143A)
(43)公開日　平成26年5月22日(2014.5.22)
審査請求日　平成28年9月13日(2016.9.13)
(31)優先権主張番号　特願2012-224575(P2012-224575)
(32)優先日　平成24年10月9日(2012.10.9)
(33)優先権主張国　日本国(JP)

(73)特許権者　000104652
キヤノン電子株式会社
埼玉県秩父市下影森１２４８番地
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】情報処理装置、情報処理システムおよび情報処理方法

(57)【特許請求の範囲】
【請求項１】
プログラムの起動およびプログラムの生成または変更の検知、もしくは、プログラムの検索を行う検知手段と、
プログラムを識別する識別手段と、
ネットワークアクセスが許可されたプログラムのリストであるホワイトリスト、およびネットワークアクセスが禁止されたプログラムのリストであるブラックリストのうち一方のリストにプログラムを登録する登録手段とを有し、
前記識別手段は、前記検知手段によって起動が検知されたプログラムまたは前記検知手段の検索によって検出されたプログラムのプログラム情報に基づき、前記プログラムがプログラム情報に関する所定の基準を満たすか否かを判定し、
前記登録手段は、前記所定の基準を満たすと判定されたプログラムを前記一方のリストに登録し、
前記登録手段は、前記所定の基準を満たすと判定されたプログラムに昇格権限を付与するために、前記プログラムと前記昇格権限の対応を前記一方のリストに登録し、
前記検知手段によって前記昇格権限が付与されたプログラムによる子プログラムの生成または変更が検知された場合、前記登録手段は、前記子プログラムを前記一方のリストに登録することを特徴とする情報処理装置。
【請求項２】
前記識別手段は、前記検知または検出されたプログラムが前記ホワイトリストおよび前

記ブラックリストのうち他方のリストに登録されているか否かを判定し、
前記登録手段は、前記他方のリストに登録されていないと判定された前記検知または検出されたプログラムに前記昇格権限を付与する請求項１に記載された情報処理装置。
【請求項３】
前記登録手段は、前記検知手段によって前記昇格権限が付与されたプログラムによる次の世代のプログラムの生成または変更が検知されると、前記次の世代のプログラムを前記一方のリストに登録する処理を再帰的に繰り返すことを特徴とする請求項１に記載された情報処理装置。
【請求項４】
前記検知手段によって前記昇格権限が付与されたプログラムによる前記次の世代のプログラムの起動が検知された場合、前記登録手段は、前記次の世代のプログラムに前記昇格権限を付与することを特徴とする請求項３に記載された情報処理装置。
【請求項５】
前記検知手段によって前記昇格権限が付与された前記次の世代のプログラムによる他の次の世代のプログラムの起動が検知された場合、前記登録手段は、前記他の次の世代のプログラムに前記昇格権限を付与することを特徴とする請求項３に記載された情報処理装置。
【請求項６】
前記検知手段によって前記昇格権限を付与されたプログラムによるインストーラの起動が検知された場合、前記登録手段は、前記インストーラに前記昇格権限を付与することを特徴とする請求項３に記載された情報処理装置。
【請求項７】
前記登録手段は、予め記憶された生成プログラム群に属するプログラムから生成され、かつ、予め記憶された起動プログラム群に属するプログラムから起動されたプログラムに、前記昇格権限を付与することを特徴とする請求項３に記載された情報処理装置。
【請求項８】
前記識別手段は、前記検知手段によって前記昇格権限が付与されたプログラムによる次の世代のプログラムの生成または変更が検知されると、前記次の世代のプログラムのプログラム情報に基づき、前記次の世代のプログラムが前記ホワイトリストおよび前記ブラックリストのうち他方のリストに登録されているか否かを判定し、
前記登録手段は、前記他方のリストに登録されていないと判定された次の世代のプログラムに前記昇格権限を付与することを特徴とする請求項４から請求項７の何れか一項に記載された情報処理装置。
【請求項９】
前記登録手段は、前記他方のリストに登録されていると判定された次の世代のプログラムの登録を前記一方のリストから削除することを特徴とする請求項８に記載された情報処理装置。
【請求項１０】
前記検知手段によって前記昇格権限が付与されたプログラムによる次の世代のプログラムの起動が検知された場合、前記識別手段は、前記起動が検知されたプログラムのプログラム情報に基づき、前記プログラムが、前記ホワイトリストおよび前記ブラックリストのうち他方のリストに登録されているか否かの判定、または、前記一方のリストに登録されているか否かの判定を行い、
前記登録手段は、前記他方のリストに登録されていないと判定されたプログラム、または、前記一方のリストに登録されていないと判定されたプログラムについて、前記一方のリストに登録するための処理を実行することを特徴とする請求項１から請求項９の何れか一項に記載された情報処理装置。
【請求項１１】
前記識別手段は、前記所定の基準を満たすと判定されたプログラムと関連するプログラムの設定情報を取得し、

前記登録手段は、前記所定の基準を満たすと判定されたプログラムと、前記識別手段によって取得された設定情報と、を関連付けて登録することを特徴とする請求項１０に記載された情報処理装置。

【請求項１２】

さらに、前記一方のリストに基づいてプログラムのネットワークアクセスを許可する、または前記ホワイトリストおよび前記ブラックリストのうち他方のリストに基づいてプログラムのネットワークアクセスを禁止する制御手段を有することを特徴とする請求項１から請求項１１の何れか一項に記載された情報処理装置。

【請求項１３】

前記プログラム情報にはディジタル署名が含まれることを特徴とする請求項１から請求項１２の何れか一項に記載された情報処理装置。

【請求項１４】

プログラムの起動およびプログラムの生成または変更の検知、もしくは、プログラムの検索を行う検知手段、プログラムを識別する識別手段、ネットワークアクセスが許可されたプログラムのリストであるホワイトリスト、およびネットワークアクセスが禁止されたプログラムのリストであるブラックリストのうち一方のリストにプログラムを登録する登録手段を有する情報処理装置の情報処理方法であって、

前記識別手段が、前記検知手段によって起動が検知されたプログラムまたは前記検知手段の検索によって検出されたプログラムのプログラム情報に基づき、前記プログラムがプログラム情報に関する所定の基準を満たすか否かを判定し、

前記登録手段が、前記所定の基準を満たすと判定されたプログラムを前記一方のリストに登録し、

前記登録手段は、前記所定の基準を満たすと判定されたプログラムに昇格権限を付与するために、前記プログラムと前記昇格権限の対応を前記一方のリストに登録し、

前記検知手段によって前記昇格権限が付与されたプログラムによる子プログラムの生成または変更が検知された場合、前記登録手段は、前記子プログラムを前記一方のリストに登録することを特徴とする情報処理方法。

【請求項１５】

請求項１から請求項１３の何れか一項に記載された情報処理装置と、

ネットワークを介して、前記情報処理装置に前記一方のリストのデータを送信するサーバ装置とを有することを特徴とする情報処理システム。

【請求項１６】

前記サーバ装置は、前記ネットワークを介して、前記情報処理装置から前記一方のリストに登録すべきプログラムに関する情報を受信することを特徴とする請求項１５に記載された情報処理システム。

【請求項１７】

コンピュータを請求項１から請求項１３の何れか一項に記載された情報処理装置の各手段として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、ネットワークアクセス制御を行う情報処理に関する。

【背景技術】

【０００２】

近年、企業に対するサイバー攻撃の新しい形態として、企業内の特定の社員が使用または所有するコンピュータを標的にし、そのコンピュータにマルウェアを感染させ、企業内の情報を盗み出す、標的型攻撃という手法が増えている。

【０００３】

従来のアンチウィルスソフトなどは、ブラックリスト方式によるウィルス定義ファイルを用いる。しかし、コンピュータウィルスを含むマルウェアの種類は日に万単位で増加し

ている。そのため、ウィルス定義ソフトの更新は、マルウェアの急増に追いつかない状況にあり、従来のアンチウィルスソフトによる標的型攻撃への対処は難しい状況にある。

【０００４】

一方、既知のプログラムによるネットワークアクセスを許可し、その他のプログラムのプログラムによるネットワークアクセスを制限する、いわゆるホワイトリスト型のネットワークアクセス制御を用いる標的型攻撃への対処が存在する（例えば、特許文献1）。

【０００５】

ホワイトリスト型のネットワークアクセス制御を開始する際、許可対象のプログラムをホワイトリストに登録する。しかし、セキュリティパッチなどのアップデートが行われると、ホワイトリストに登録した許可対象のプログラムが更新され、更新後のプログラムをホワイトリストに再登録する必要がある。さらに、アップデータによって新規にプログラムが生成される場合、新規に生成されるプログラムもホワイトリストに登録する必要がある。

【０００６】

従来、システム管理者などは、アップデートの都度、更新されるプログラムやアップデータが生成するプログラムをホワイトリストに登録する作業を行っていた。この方法によれば、アップデータが信頼できるか否かの判断だけでなく、更新されるプログラムや生成されるプログラムをホワイトリストに登録する負担が大きい作業が必要になる。

【先行技術文献】

【特許文献】

【０００７】

【特許文献１】特開2009-259160号公報

【発明の概要】

【発明が解決しようとする課題】

【０００８】

本発明は、リスト型のネットワークアクセス制御を行う場合の、リストの更新にかかる作業を軽減することを目的とする。

【課題を解決するための手段】

【０００９】

本発明は、前記の目的を達成する一手段として、以下の構成を備える。

【００１０】

本発明にかかる情報処理装置は、プログラムの起動およびプログラムの生成または変更の検知、もしくは、プログラムの検索を行う検知手段と、プログラムを識別する識別手段と、ネットワークアクセスが許可されたプログラムのリストであるホワイトリスト、およびネットワークアクセスが禁止されたプログラムのリストであるブラックリストのうち一方のリストにプログラムを登録する登録手段とを有し、前記識別手段は、前記検知手段によって起動が検知されたプログラムまたは前記検知手段の検索によって検出されたプログラムのプログラム情報に基づき、前記プログラムがプログラム情報に関する所定の基準を満たすか否かを判定し、前記登録手段は、前記所定の基準を満たすと判定されたプログラムを前記一方のリストに登録し、前記登録手段は、前記所定の基準を満たすと判定されたプログラムに昇格権限を付与するために、前記プログラムと前記昇格権限の対応を前記一方のリストに登録し、前記検知手段によって前記昇格権限が付与されたプログラムによる子プログラムの生成または変更が検知された場合、前記登録手段は、前記子プログラムを前記一方のリストに登録することを特徴とする。

【発明の効果】

【００１１】

本発明によれば、リスト型のネットワークアクセス制御を行う場合の、リストの更新にかかる作業を軽減することができる。

【図面の簡単な説明】

【００１２】

【図１】ホワイトリスト制御システムの構成例を示すブロック。
【図２】サーバが存在しないホワイトリスト制御システムの構成例を示すブロック図。
【図３】ホワイトリストの一例を示す図。
【図４】サーバに存在するホワイトリストマスタの一例を示す図。
【図５】ホワイトリスト制御処理の一例を説明するフローチャート。
【図６】インストーラが起動された場合のホワイトリスト制御処理の一例を説明するフローチャート。
【図７】ネットワークアクセス制御を説明するフローチャート。
【図８】プログラム起動をトリガとするホワイトリストの更新処理を説明するフローチャート。
【図９】プログラム検索をトリガとするホワイトリストの更新処理を説明するフローチャート。
【図１０】上書アップデートによるホワイトリストの更新処理を説明するフローチャート。
【図１１】置換アップデートによるホワイトリストの更新処理を説明するフローチャート。
【図１２】登録プログラムが情報を登録するリストの一例を示す図。
【図１３】置換アップデートによるホワイトリストの更新処理を説明するフローチャート。

【発明を実施するための形態】

【００１３】
以下、本発明にかかる実施例の情報処理システムの情報処理を図面を参照して詳細に説明する。

【００１４】
［システムの構成］
図1のブロックによりホワイトリスト制御システムの構成例を示す。ホワイトリスト制御システムは、情報処理装置と、情報処理装置を管理するサーバ装置を含む。

【００１５】
図1において、クライアントコンピュータ（以下、クライアント）10は、ホワイトリスト制御システムにおける情報処理装置である。クライアント10は、例えば、企業、学校、行政機関または家庭などに設置されたパーソナルコンピュータ(PC)や、個人が使用または所有するタブレット端末やスマートフォンなどのコンピュータ機器である。

【００１６】
サーバコンピュータ（以下、サーバ）20は、ホワイトリスト制御システムにおける情報処理装置を管理するサーバ装置である。サーバ20は、複数のクライアント10からホワイトリスト120の情報を取得してデータベース化したり、定期的にクライアント10にホワイトリストデータを送信してホワイトリスト120の更新を行う。

【００１７】
ネットワーク300は、インターネットやイントラネットなどのコンピュータネットワークである。クライアント10は、ネットワーク300を介して、サーバ20や、図示しないウェブサーバやFTPサーバなどと接続する。

【００１８】
なお、簡潔化のため図1にはクライアント10とサーバ20を一台ずつ示すが、実際には、ホワイトリスト制御システムに複数のクライアントや複数のサーバが存在することができる。

【００１９】
●クライアント
クライアント10において、演算装置10Cはマイクロプロセッサ(CPU)である。演算装置10Cは、メモリ10EのROMに格納されたBIOSなどのブートプログラムに従い記憶装置10Bに格納されたオペレーティングシステム(OS)を起動し、さらにOSに従い各種の常駐プログラム（

例えば制御プログラム113など）を起動する。その際、演算装置10Cは、メモリ10EのRAMをワークエリアとして使用する。また、OSは例えばWindows（登録商標）、Mac OS（登録商標）、Linux（登録商標）、iOS（商標）、Android（商標）などである。

【００２０】

記憶装置10Bは、ハードディスクドライブ(HDD)やソリッドステートドライブ(SSD)などであり、OSのほかにクライアント10上で稼働する各種のプログラム100やデータ101を格納する。詳細は後述するが、記憶装置10Bが格納する各種プログラム100には識別プログラム110、登録プログラム111、検知プログラム112、制御プログラム113、ファイル検索ツール114などが含まれる。また、詳細は後述するが、記憶装置10Bが格納する各種データ101にはホワイトリスト120、昇格基準ルール130、ブラックリスト140などが含まれる。

【００２１】

なお、プログラム100は、識別プログラム110など各種機能ごとのプログラムを複数備えてもよいし、各種機能を備えた一つのプログラムでもよい。

【００２２】

I/Oデバイス10Aは、ポインティングデバイス（マウスなど）やキーボードに接続するための入出力インタフェース(I/F)、または、タッチパネルを組み込んだディスプレイなどである。なお、キーボードはソフトウェアキーボードでもよい。また、I/Oデバイス10Aは、入力された操作者の音声を音声認識機能によって認識し、認識した音声を演算装置10Cへ伝達する、マイク等を含む音声式入力部でもよい。また、I/Oデバイス10Aは、情報を表示するためのユーザインタフェース(UI)としても機能する。

【００２３】

ネットワークI/F 10Dは、ネットワーク300とのインタフェースであり、他のコンピュータと通信するための通信回路である。演算装置10Cは、ネットワークI/F 10Dを介して、例えばホワイトリスト120の一部データなどの情報をサーバ20から受信し、また、各種情報をサーバ20に送信する。

【００２４】

●サーバ

サーバ20において、演算装置20Cはマイクロプロセッサ(CPU)である。演算装置20Cは、メモリ20EのROMに格納されたBIOSなどのブートプログラムに従い記憶装置20Bに格納されたOSを起動する。さらに、演算装置20Cは、記憶装置20Bから管理コンソール210をメモリ20EのRAMにロードする。そして、複数のクライアント10から情報（例えば、ホワイトリスト120の情報など）を取得してデータベース化したり、逆に、クライアント10に対して情報を送信してホワイトリスト120の更新などを行う。

【００２５】

記憶装置20Bは、HDDやSSDなどであり、OSのほかにサーバ20上で稼働する管理コンソール210を含む各種のプログラム200やデータ201を格納する。詳細は後述するが、記憶装置10Bが格納する各種データ201にはホワイトリストマスタ220、昇格基準ルール230、ブラックリストマスタ240、ホワイトリスト候補250などが含まれる。

【００２６】

I/Oデバイス20Aは、ポインティングデバイス（マウスなど）、キーボード、モニタに接続するためのインタフェース(I/F)であり、モニタは情報を表示するためのUIとして機能する。ネットワークI/F 20Dは、ネットワーク300とのインタフェースであり、クライアント10など他のコンピュータと通信するための通信回路である。

【００２７】

演算装置20Cは、ネットワークI/F 20Dを介して、複数のクライアント10からホワイトリスト120やブラックリスト140に関する情報を受信し、受信した情報に基づき、ホワイトリストマスタ220やブラックリストマスタ250を管理する。

【００２８】

ホワイトリスト制御システムにおいて、サーバ20は必須の構成ではない。図2のブロック図によりサーバ20が存在しないホワイトリスト制御システムの構成例を示す。図2の構

成においては、クライアント10とサーバ20の通信は不要になるため、ネットワーク300およびネットワークI/F 10Dもオプションである。

【００２９】

また、ホワイトリスト制御システムはシンクライアント（例えば、ターミナルサービスなど）を利用した構成としてもよい。シンクライアントは、クライアントがサーバにリモート接続し、サーバ上に生成された仮想デスクトップ環境を利用してサーバ上でアプリケーションプログラムを実行できるようにするシステムである。

【００３０】

［プログラムおよびデータ］

識別プログラム110は、演算装置10Cによって実行され、別途起動されるプログラムからファイル名（プログラム名）、ハッシュ値、バージョン情報、ファイルサイズ、ファイルパス、ディジタル署名などの情報（以下、プログラム情報）を取得する。また、プログラム情報には、当該プログラムが格納されているPC名も含まれる。そして、取得したプログラム情報に基づき当該プログラムを識別する識別機能を有する。また、識別プログラム110は、取得したプログラム情報と後述する昇格基準ルール130を照会して、当該プログラムが昇格基準を満たすか否かを判定する。

【００３１】

ホワイトリスト120は、ネットワークアクセスを許可するプログラムに関する情報をリスト化したものである。ホワイトリスト120を構成する情報には、識別プログラム110によって取得されたプログラム情報が用いられる。

【００３２】

図3によりホワイトリスト120の一例を示す。ホワイトリスト120は、各プログラムについて、プログラム名（実行ファイル名）、ハッシュ値、バージョン情報、ファイルサイズ、接続先IPアドレス、接続先ポート番号、後述する昇格権限フラグの七種類の情報を保持する。なお、図3は一例であり、ホワイトリスト120として保持する情報の種類と数は図3に限定されるものではない。

【００３３】

ホワイトリストマスタ220は、複数のホワイトリスト120をリスト化したものである。図4によりサーバ20に存在するホワイトリストマスタ220の一例を示す。ホワイトリストマスタ220は、複数のクライアントのホワイトリスト120に関連する情報をクライアントの名称や符号に対応付けて保持する。図4の例では、PC003に対応するホワイトリスト003として、複数のプログラムのプロセス名、ハッシュ値、接続先IPアドレス、接続先ポート番号、登録日時、最終起動日時が保持された例を示す。なお、図4は一例であり、ホワイトリストマスタ220として保持する情報の種類と数は図4に限定されるものではない。

【００３４】

検知プログラム112は、演算装置10Cによって実行され、プログラムの起動と、起動されたプログラムによる別のプログラムの生成を監視する監視機能と、それらを検知する検知機能を有する。

【００３５】

登録プログラム111は、演算装置10Cによって実行され、検知プログラム112に起動や生成が検知されたプログラムの、識別プログラム110が取得したプログラム情報に基づき、当該プログラムをホワイトリスト120に登録する登録機能を有する。

【００３６】

制御プログラム113は、演算装置10Cによって実行され、ホワイトリスト120に基づき、クライアント10上で起動されたプログラムのネットワークアクセスを許可または禁止する制御機能を有する。また、制御プログラム113は、プログラムがアクセスしようとする接続先のIPアドレスやポート番号を判断基準としてネットワークアクセスを制御する制御機能も有する。

【００３７】

昇格基準ルール130は、プログラムやファイルが、信頼できる発行者によって発行され

たものか否かを判断するためのルールである。昇格基準ルール130は、プログラム情報を元に、管理者やユーザなどが定義したルールである。ツールには、例えば、プログラムやファイルに付加されたディジタル署名、ディジタル証明書が正当か否かの検証、ファイルの署名者名が予め記憶された名前か否かの判定、ファイル名が指定条件を満たすか否かの判定などがある。

【００３８】

また、昇格基準ルール130として適用するルールの種類や数は、前記の具体例に限定されるものではない。例えば「ディジタル署名やディジタル証明書が正当であり、かつファイル名に"Setup"または"Update"という文字が含まれている」という複数の組み合わせのルールを適用することもできる。この場合「画像ビューワ(Viewer.exe)の脆弱性を突いてマルウェアを生成させるマルウェア」が動作したとき、画像ビューワのディジタル署名が正当でも、そのファイル名に前記の文字列が含まれない。その結果、画像ビューワの脆弱性を突く攻撃を防ぐ効果が得られる。

【００３９】

ブラックリスト140は、ネットワークアクセスが禁止されたプログラムをリスト化したものである。ブラックリスト140のデータ構造は、図3に示すホワイトリスト120と略同一である。また、ブラックリスト140は必須ではなくクライアント10上に存在しなくてもよいし、ブラックリスト140を使用しない場合はサーバ20のブラックリストマスタ240も必須ではない。

【００４０】

［ホワイトリスト制御処理］

●処理の概要

検知プログラム112は、グローバルフック（APIフック、フィルタドライバ）などを用いて、クライアント10におけるプログラムの起動を検知し、プログラムの起動を検知すると識別プログラム110を呼び出す。識別プログラム110は、起動されるプログラムのプログラム情報を取得して、当該プログラムが昇格基準ルール130を満たすか否かを検証する。

【００４１】

検証の結果、当該プログラムが昇格基準ルール130を満たすと判断された場合、制御プログラム113は、当該プログラムがアップデータであると判断し、登録プログラム111を呼び出す。登録プログラム111は、識別プログラム110から当該プログラムのプログラム情報を受け取り、当該プログラムをホワイトリスト120に登録する。この登録により、当該プログラムのネットワークアクセスが可能になる。

【００４２】

なお、起動されるプログラムが昇格基準ルール130を満たすとしても、当該プログラムがブラックリスト140に登録されている場合、制御プログラム113は登録プログラム111に当該プログラムの登録を実行させない。つまり、当該プログラムのネットワークアクセスの禁止が維持される。なお、ネットワークアクセスの禁止が維持されれば、起動されるプログラムの実行を阻止、もしくは、許可してもよい。

【００４３】

この時点では、プログラムがアクセスする接続先のIPアドレスやポート番号を知ることはできない。そのため、登録プログラム111は、接続先のIPアドレスやポート番号を制限しない設定のレコード（例えば図3の三行目）をホワイトリスト120に登録する。従って、アップデータと判断されたプログラムは、ネットワークアクセスの制限を受けず、正常に動作することが可能である。

【００４４】

次に、登録プログラム111は、ホワイトリスト120に登録したプログラムに「昇格権限」を与える。昇格権限の有無は、例えば、ホワイトリスト120の昇格権限フラグに設定する。あるいは、ホワイトリスト120に対応するテーブルを記憶装置10Bに格納し、当該テーブルの各レコードに昇格権限の有無を登録してもよい。昇格権限は、以下のように定義される権限である。

【００４５】
昇格権限を有するプログラム（親プログラム）が何らかのプログラム（子プログラム）を生成した場合、子プログラムは無条件にホワイトリスト120に登録される。そして、親プログラムが子プログラムを起動した場合、子プログラムにも昇格権限が与えられる。この昇格権限に関する登録処理を、図1に示す構成が行う場合、以下のような処理になる。
【００４６】
検知プログラム112は、親プログラムの挙動をグローバルフックなどを用いて監視する。検知プログラム112は、親プログラムによる子プログラムの生成を検知すると、識別プログラム110に子プログラムのプログラム情報を取得させる。識別プログラム110は、取得したプログラム情報を登録プログラム111に渡す。登録プログラム111は、受け取ったプログラム情報に基づきホワイトリスト120に追加するレコードのデータを作成し、作成したデータをホワイトリスト120に追加する。
【００４７】
このとき、次の方式の採用も可能である。つまり、検知プログラム112は、子プログラムの生成を検知すると、生成されたファイルを解析し、生成されたファイルに関する情報をホワイトリストへ登録する必要の有無を判断する。そして、登録不要と判断した場合、識別プログラム110によるプログラム情報の取得を行わずに以降の処理を打ち切る。登録不要と判断される場合は、例えば、生成されたファイルのバイナリヘッダを解析し、その構成がPE (Portable Executable)形式ではなく、実行ファイルではないと判断することができる場合などである。
【００４８】
次に、検知プログラム112は、親プログラムによる子プログラムの起動を監視する。検知プログラム112は、親プログラムによる子プログラムの起動を検知すると、登録プログラム111を呼び出す。呼び出された登録プログラム111は、子プログラムの昇格権限フラグに「有」を設定する。また、制御プログラム113は、子プログラムのネットワークアクセスを許可する。
【００４９】
昇格権限を利用すれば、例えばソフトウェアのセキュリティパッチなど、子プログラムを生成する挙動を有するプログラムが生成したプログラムを自動的にホワイトリスト120に追加することができる。言い換えれば、更新されるプログラムや生成される子プログラムをホワイトリスト120に登録する、ホワイトリストの更新にかかる作業を軽減することができる。
【００５０】
なお、昇格権限に関する処理は、親プログラムから子プログラムを生成した場合に限らず、親プログラムに対する変更（例えば、リネーム）、または、子プログラムに対する変更の場合でも可能である。
【００５１】
また、昇格基準ルール130に「プログラムに正当なディジタル署名が付加されているかを判定する」を適用する。こうすれば、信頼できる発行元が発行したプログラムであり、マルウェアのような悪意のあるプログラムではないことを操作者の意識を介さずに判断可能である。なお、正当なディジタル署名が付加されているか否かは、例えばWindows（登録商標）アプリケーションプログラミングインタフェース(API)などを用いる方法などがある。
【００５２】
また、昇格基準ルール130を満たさないプログラムと判断された場合、制御プログラム113は一般的なホワイトリスト制御を行う。つまり、識別プログラム110は、当該プログラムが昇格基準ルール130を満たさなと判断すると、当該プログラムがホワイトリスト120に登録されているか否かを判定する。制御プログラム113は、当該プログラムがホワイトリスト120に登録されていると判定されると、昇格権限を有するか否か判定し、当該プログラムのネットワークアクセスを許可する。また、当該プログラムがホワイトリスト120に

登録されていないと判定されると、グローバルフックやパケットフィルタリングなどにより当該プログラムのネットワークアクセスを禁止する。

【００５３】

また、識別プログラム110により、起動されるプログラムがブラックリスト140に登録されているか否かを判定する。そして、ブラックリスト140に登録されている場合は当該プログラムのネットワークアクセスを禁止し、ブラックリスト140に登録されていない場合は当該プログラムのネットワークアクセスを許可する方式の採用も可能である。

【００５４】

なお、以下の説明において、プログラムなどがホワイトリスト120またはブラックリスト140に登録されている状態を「リストに存在する」、登録されていない状態を「リストに存在しない」と表現する場合がある。

【００５５】

●処理の詳細

図5のフローチャートによりホワイトリスト制御処理の一例を説明する。

【００５６】

検知プログラム112はプログラムの起動を監視し(S201)、プログラムの起動を検知すると処理をステップS202に進める。

【００５７】

プログラムの起動が検知されると、識別プログラム110は、当該プログラムのプログラム情報を取得し(S202)、当該プログラムがブラックリスト140に存在するか否か判定する(S203)。当該プログラムがブラックリスト140に存在すると判定された場合、制御プログラム113は、当該プログラムのネットワークアクセスを禁止し(S204)、処理をステップS201に戻す。なお、ブラックリスト140が存在しない場合、識別プログラム110は、ステップS203の判定をスルーパスする。

【００５８】

また、当該プログラムがブラックリスト140に存在しない場合、識別プログラム110は、当該プログラムが昇格基準ルール130を満たすか否かを判定する(S205)。当該プログラムが昇格基準ルール130を満たさない場合、識別プログラム110は、当該プログラムがホワイトリスト120に存在するか否かを判定する(S206)。

【００５９】

当該プログラムがホワイトリスト120に存在すると判定された場合、識別プログラム110は、当該プログラムが昇格権限を有するか否かを判定する(S206B)。昇格権限を有すると判定された場合、制御プログラム113は当該プログラムのネットワークアクセスを許可する(S210)。

【００６０】

昇格権限が無いと判定された場合、制御プログラム113は当該プログラムのネットワークアクセスを許可し(S207)、処理をステップS201に戻す。また、当該プログラムがホワイトリスト120に存在しないと判定された場合、制御プログラム113は当該プログラムのネットワークアクセスを禁止し(S204)、処理をステップS201に戻す。

【００６１】

一方、当該プログラムが昇格基準ルール130を満たすと判定された場合、制御プログラム113は、当該プログラムのプログラム情報を登録プログラム111に渡す。これにより、登録プログラム111は、当該プログラム（親プログラム）をホワイトリスト120に登録し(S208)、昇格権限フラグに「有」を設定する(S209)。続いて、制御プログラム113は、親プログラムのネットワークアクセスを許可する(S210)。

【００６２】

次に、検知プログラム112は、昇格権限が付与された親プログラムによる子プログラムの生成を監視する(S211)。検知プログラム112は、親プログラムが子プログラムを生成した場合は処理をステップS212に進め、親プログラムが子プログラムを生成しない場合は処理をステップS201に戻す。

【００６３】
なお、ステップS211において、検知プログラム112は、子プログラムの生成だけでなく、子プログラム、もしくは、親プログラムに対する変更（例えば、リネーム）を監視してもよい。親プログラムの変更が検知された場合、当該プログラム（親プログラム）に対して、子プログラムと同様の処理を行う。
【００６４】
子プログラムの生成が検知されると、識別プログラム110は、子プログラムのプログラム情報を取得し(S212)、取得したプログラム情報を登録プログラム111に渡す。これにより、登録プログラム111は、子プログラムをホワイトリスト120に登録する(S213)。
【００６５】
続いて、検知プログラム112は、昇格権限が付与された親プログラムが子プログラムを起動するか否かを監視する(S214)。検知プログラム112は、親プログラムが子プログラムを起動する場合は処理をステップS215に進め、親プログラムが子プログラムを起動しない場合は処理をステップS201に戻す。
【００６６】
親プログラムによる子プログラムの起動が検知されると、識別プログラム110は、子プログラムがブラックリスト140に存在するか否かを判定する(S215)。子プログラムがブラックリスト140に存在すると判定された場合、制御プログラム113は子プログラムのネットワークアクセスを禁止する(S216)。そして、登録プログラム111は、ホワイトリスト120から子プログラムの登録を削除し(S217)、処理をステップS201に戻す。
【００６７】
他方、子プログラムがブラックリスト140に存在しないと判定された場合、処理はステップS209に戻る。従って、子プログラムの昇格権限フラグに「有」が設定され(S209)、子プログラムのネットワークアクセスが許可され(S210)、昇格権限が付与された子プログラムによる孫プログラムの生成が監視される(S211)。そして、子プログラムが孫プログラムを生成すると、孫プログラムをホワイトリスト120に登録し、昇格権限が付与された子プログラムが孫プログラムを起動すると孫プログラムに昇格権限を与える処理（S209からS215）が再帰的に繰り返される。
【００６８】
なお、ブラックリスト140が存在しない場合、識別プログラム110は、ステップS215の判定をスルーパスする。その場合、親プログラムによって起動された子プログラムや子プログラムによって起動された孫プログラムの昇格権限フラグに「有」が設定され(S209)、子プログラムのネットワークアクセスが許可される(S210)。
【００６９】
なお、親プログラム/子孫プログラムをホワイトリスト120に登録する際に、当該プログラムがホワイトリスト120に登録されているか否かを判定し、未登録と判定した場合に当該プログラムをホワイトリスト120に登録してもよい。また、ホワイトリスト120に既登録の場合は登録を行わずに、次の処理に進む。ホワイトリスト120に登録されているか否かを判定することにより、プログラムの重複登録を防ぐことができる。
【００７０】
以下では、第一の世代の親プログラムが元になって生成される子プログラムや孫プログラムなど、第二の世代以降のプログラムを「子孫プログラム」と呼ぶことにする。
【００７１】
●インストーラへの対応
ステップS201において、検知プログラム112は、リストとの比較により、起動されるプログラムが記憶装置10Bに予め格納されているインストーラプログラム（例えばWindows（登録商標）におけるmsiexec.exe）か否かを判定する。そして、インストーラプログラム（以下、インストーラ）の起動と判定された場合、インストーラの動作が他のプログラムの動作と異なるため、ステップS202以降の処理を切り替える。
【００７２】

操作者がインストーラパッケージファイル（msiファイル、mspファイル、msuファイルなど）の実行を指示するとインストーラが起動され、インストーラは、インストーラパッケージファイル（以下、パッケージ）に格納されたファイルを展開する。従って、昇格基準ルール130に基づく判定は、インストーラではなく、パッケージに対して行う必要があり、図5に示す処理を直接適用することができない。

【００７３】

図6のフローチャートによりインストーラが起動された場合のホワイトリスト制御処理の一例を説明する。

【００７４】

識別プログラム110は、パッケージのファイル情報を取得し(S221)、パッケージがブラックリスト140に存在するか否かを判定する(S222)。当該パッケージがブラックリスト140に存在すると判定された場合、制御プログラム113は、インストーラの起動を禁止し(S223)、処理をステップS201に戻す。なお、ブラックリスト140が存在しない場合、識別プログラム110は、ステップS222の判定をスルーパスする。

【００７５】

また、当該パッケージがブラックリスト140に存在しない場合、識別プログラム110は、当該パッケージが昇格基準ルール130を満たすか否かを判定する(S224)。当該パッケージが昇格基準ルール130を満たさない場合、制御プログラム113はインストーラの起動を禁止し(S223)、処理をステップS201に戻す。つまり、ブラックリスト140に登録されたパッケージや昇格規準ルール130は満たさないパッケージのインストールは中止される。

【００７６】

また、当該パッケージがブラックリスト140に存在しない場合、識別プログラム110は、当該パッケージが昇格基準ルール130を満たすか否かを判定する(S224)。当該パッケージが昇格基準ルール130を満たさない場合、識別プログラム110は、当該パッケージがホワイトリスト120に存在するか否かを判定する(S225)。当該パッケージがホワイトリスト120に存在しないと判定された場合はインストーラの起動を阻止し(S223)、処理をステップS201に戻す。

【００７７】

一方、当該パッケージが昇格基準ルール130を満たす、または、ホワイトリスト120に存在する場合、制御プログラム113はインストーラの起動を許可する(S226)。その後の処理は図5に示すステップS209からS217と同様であり、インストーラの昇格権限フラグに「有」が設定され(S209)、インストーラのネットワークアクセスが許可される(S210)。そして、インストーラによってパッケージから取り出されたプログラムは、親プログラム（この場合はインストーラ）によって生成された子プログラムと同等に扱われる。つまり、パッケージから取り出されたプログラムは子孫プログラムとしてホワイトリスト120に登録され、子孫プログラムが次の世代のプログラムを起動すると、起動されるプログラムに昇格権限を与える処理（S209からS215）が再帰的に繰り返される。

【００７８】

なお、ステップS211において、親プログラムによる子プログラムの生成ではなく、当該プログラム（親プログラム）によるインストーラパッケージが生成された場合は、そのパッケージファイルに対して、図6の処理を適用することも可能である。

【００７９】

［親プログラムが生成した子プログラム以外に昇格権限を継承/付与する場合］

昇格権限を持った親プログラムから生成された子プログラムが親プログラムによって起動された場合、昇格権限を継承するが、以下のような場合も昇格権限を継承または付与してもよい。

【００８０】

●パターン1

図5において、同一の親プログラムから複数の子孫プログラムが生成される場合を考える。このとき、子孫プログラムの中で親プログラムから昇格権限を継承したものがあり、

昇格権限を継承した子孫プログラムが、他の子孫プログラムを起動した場合、昇格権限を継承してもよいものとする。

【００８１】

例えば、Windows（登録商標）のアップデート動作の中には上記のような動作がある。パターン1の手法を実施することで、Windows（登録商標）のアップデートをブロックすることなく行うことが可能になる。

【００８２】

パターン1を実施する場合、図5に示すステップS214において「親プログラム、または、同じ親プログラムから生成され、当該親プログラムから昇格権限を継承したプログラムによって、子孫プログラムが起動されるか？」という判定が行われる。

【００８３】

●パターン2

図6において、インストーラが他のプログラムから起動される場合を考える。このとき、昇格権限をもつプログラムがインストーラを起動した場合、インストーラが昇格権限を継承してもよいものとする。

【００８４】

市販のソフトウェアには、インストール時にインストーラを起動するものがある。パターン2の手法を実施することで、そのようなソフトウェアのインストールをブロックすることなく行うことが可能になる。

【００８５】

●パターン3

図5において、昇格権限をもつプログラムから生成された子プログラムを、昇格権限をもたないプログラムが起動する場合を考える。このとき「生成プログラム群」と「起動プログラム群」という二種類のプログラム群を予め定義する。そして、生成プログラム群に属すプログラムから生成され、かつ、起動プログラム群に属すプログラムから起動されたプログラムには昇格権限を付与する処理を行ってもよいものとする。

【００８６】

市販のソフトウェアには、インストール時に上記のような動作を行うものがある。パターン3の手法を実施することで、そのようなソフトウェアのインストールをブロックすることなく行うことが可能になる。

【００８７】

［ネットワークアクセス制御］

図7のフローチャートによりネットワークアクセス制御を説明する。ネットワークアクセス制御はOSを実行する演算装置10Cによって実行される。プログラムごとのネットワークアクセスの許可または禁止を示す情報（例えばアクセス制御リスト(ACL)）は、制御プログラム113により、メモリ10EのRAMや記憶装置10Bの所定領域にテーブルとして格納されている。

【００８８】

演算装置10Cは、プログラムによるネットワークアクセスを監視する(S501)。演算装置10Cは、プログラムがネットワークアクセス要求を発行すると、ACLを参照して、当該プログラムのネットワークアクセスの許可または禁止を判定する(S502)。そして、判定結果に従い、当該プログラムのネットワークアクセスを制御する。

【００８９】

演算装置10Cは、当該プログラムのネットワークアクセスが許可されている場合、ネットワークアクセスを許可する(S503)。つまり、当該プログラムが発行したコマンドやデータをネットワークI/F 10Dに転送し、ネットワークI/F 10Dが受信した当該プログラムあてのデータを当該プログラムに転送する。なお、ACLには、プログラムごとに、接続先のIPアドレスやポート番号の制限が設定されている場合がある。その場合、演算装置10Cは、当該制限に従いフィルタリングを行う。

【００９０】

また、演算装置10Cは、当該プログラムのネットワークアクセスが禁止されている場合、ネットワークアクセスを許可しない(S504)。つまり、当該プログラムとネットワークI/F 10Dの間のデータ転送を実行せず、当該プログラムのネットワークアクセス要求に対してエラーメッセージを返す。

【００９１】

［ホワイトリストの作成］

ホワイトリスト制御システムの運用には、運用環境に適したホワイトリスト120を作成する必要がある。

【００９２】

●プログラム起動をトリガとする処理

図5においては、プログラムが昇格基準ルール130を満たさず、かつ、ホワイトリスト120に存在しない場合（S206のNO）、制御プログラム113はプログラムのネットワークアクセスを禁止する(S204)と説明した。また、図6においては、パッケージが昇格基準ルール130を満たさず、かつ、ホワイトリスト120に存在しない場合（S225のNO）、制御プログラム113はインストーラの起動を禁止する(S223)と説明した。しかし、そのようなプログラムやパッケージ（以下、未登録プログラム）が検出された場合、ホワイトリスト120を更新を試みることが可能である。

【００９３】

図8のフローチャートによりプログラム起動をトリガとするホワイトリストの更新処理を説明する。

【００９４】

登録プログラム111は、未登録プログラムが検出されると(S301)、識別プログラム110から渡された未登録プログラムのプログラム情報（以下、未登録情報）をサーバ20に送信する(S302)。そして、処理をステップS301に戻す。

【００９５】

なお、未登録プログラムの検出直後に未登録情報を送信せずに、所定のサイクルで未登録情報をサーバ20に送信するようにしてもよい。例えば、登録プログラム111は、未登録情報を例えば記憶装置10Bやメモリ10Eの所定領域に一時保存する。そして、所定のサイクルで（例えば五分ごとや一時間ごとに）保存された未登録プログラム情報があるか否かを判定し、未登録プログラム情報が保存されている場合は当該情報をサーバ20に送信する。

【００９６】

サーバ20の管理コンソール210は、クライアント10から未登録情報を受信すると(S311)、受信した未登録情報に一致する情報がホワイトリスト候補250に存在するか否かを判定する(S312)。受信した未登録情報に一致する情報がホワイトリスト候補250に存在する場合は処理をステップS311に戻す。

【００９７】

一方、受信した未登録情報に一致する情報がホワイトリスト候補250に存在しない場合、管理コンソール210は、受信した未登録情報をホワイトリスト候補250へ追加する(S313)。そして、例えば電子メールやアラートウィンドウなどにより、ホワイトリスト候補250に追加した未登録情報をサーバ20の操作者に提示する(S314)。

【００９８】

操作者は、提示された情報を参照して、未登録プログラムをホワイトリストに登録するか否かを判断し、判断結果に応じた指示を管理コンソール210に入力する。管理コンソール210は、操作者の指示が当該プログラムの登録を示すか否かを判定する(S315)。操作者の指示が登録を示す場合は、当該プログラムをホワイトリストマスタ220に登録し(S316)、ホワイトリスト候補250の当該プログラムのレコードに「登録済」を記録する(S317)。また、操作者の指示が非登録を示す場合は、当該プログラムをホワイトリストマスタ220に登録せずに、ホワイトリスト候補250の当該プログラムのレコードに「非登録」を記録する(S318)。そして、処理をステップS311に戻す。

【００９９】

ホワイトリスト120の作成や更新については、例えば、図8に示す処理を適切な期間（二週間から一月程度）実施して得られたホワイトリストマスタ220に基づき、ホワイトリスト120のデータを作成または更新し、クライアント10に配布すればよい。ホワイトリスト120の更新期間は任意であり、ホワイトリスト制御システムを運用する企業、学校、行政組織などの規模によって変化する。

【０１００】

●プログラム検索をトリガとする処理

上記では、プログラム起動をトリガとしてホワイトリスト候補250にプログラムを追加する処理を説明した。しかし、例えばOS標準のファイル検索ツールなどを用いてプログラムを検索し、検出したプログラムをホワイトリスト候補250の登録することもできる。このようにすれば、クライアント10の記憶装置10Bに格納されたすべてのプログラムをホワイトリスト候補250に追加することができる。

【０１０１】

図9のフローチャートによりプログラム検索をトリガとするホワイトリストの更新処理を説明する。

【０１０２】

クライアント10の操作者（またはサーバ20）の指示により、ファイル検索ツール114が起動され、指示された検索条件に基づきファイル検索が実行される(S401)。ホワイトリスト候補250にプログラムを追加する場合、記憶装置10Bに格納されたすべてのプログラムが検索され、識別プログラム110は検索結果を受け取る(S402)。

【０１０３】

検索結果を受け取った識別プログラム110は、検出されたプログラムのプログラム情報を取得する(S403)。そして、検出されたプログラムから、プログラムが昇格基準ルール130を満たさず、かつ、ホワイトリスト120に存在しない未登録プログラムを抽出する(S404)。登録プログラム111は、未登録プログラムが抽出されると(S405)、未登録プログラムのプログラム情報をサーバ20に送信する(S406)。

【０１０４】

管理コンソール210の処理は、プログラム起動をトリガとする場合の処理（図8のS311からS318）と同様であり、詳細説明を省略する。

【０１０５】

また、図2に示すサーバ20が存在しない構成の場合、図8に示す管理コンソール210の処理（S311からS318）もクライアント10上で実行されることは言うまでもない。

【０１０６】

［変形例］

上記では、登録プログラム111によってホワイトリスト120に登録されるレコードには、接続先のIPアドレスやポート番号を制限を設定しない例（例えば図3の三行目）を説明した。しかし、例えば、アップデート前とアップデート後のプログラム名が同じ場合でも、アップデートされるプログラムが特定可能な場合、アップデート前のプログラム（関連するプログラム）と、同一/一部同一の設定情報（接続先のIPアドレスやポート番号）に制限することができる。なお、設定情報は、IPアドレスやポート番号に限定されず、例えばMACアドレス等、端末情報を用いて制限することも可能である。

【０１０７】

例えば、アップデート前のプログラムをアップデート後のプログラムで上書きする処理やアップデート前のプログラムをアップデート後のプログラムに置き換える処理を検知し、アップデート前後のプログラムを関連付ける。そして、この関連付けに基づき、ホワイトリスト120を更新すればよい。これにより、アップデート前のプログラムと同一設定のレコードをホワイトリスト120に登録することができ、よりセキュリティの高いホワイトリストの更新機能とすることができる。以下に詳細を説明する。

【０１０８】

●上書きを検知してホワイトリストをアップデート

例えば、昇格権限を有するプロセスによってホワイトリスト120に登録されたプログラムが上書アップデートされる場合に、プロセスの書込処理を監視することで、ホワイトリスト120を適切に設定する例を説明する。なお、上書アップデートとは、例えば、アップデート前のデータ（ファイル）にアップデート後のデータ（ファイル）を上書きすることである。

【０１０９】

図10のフローチャートにより上書アップデートによるホワイトリスト120の更新処理を説明する。

【０１１０】

任意のプロセスからファイルの書込要求が発生すると、検知プログラム112は、当該書込要求を検知し(S1001)、当該書込要求を行ったプロセスが昇格権限をもつか否かを判定する(S1002)。当該プロセスが昇格権限をもたない場合、検知プログラム112は、当該書込要求に対する処理を終了し、再び、プロセスからの書込要求を監視する。

【０１１１】

一方、書込要求を行ったプロセスが昇格権限をもつ場合、識別プログラム110は、書込要求から書込対象のプログラム（以下、対象プログラム）を特定する。例えば、ファイルパス、ファイル名から対象プログラムを特定すればよい。そして、対象プログラムのハッシュ値を取得し、そのハッシュ値に基づき対象プログラムがホワイトリスト120に登録されているか否かを判定する(S1003)。

【０１１２】

ホワイトリスト120に未登録の対象プログラムの場合、処理は図5または図6に示すホワイトリスト制御処理に移行し、対象プログラムがホワイトリスト120に追加され(S1007)、書込要求に対する処理が終了する。

【０１１３】

他方、ホワイトリスト120に既登録の対象プログラムの場合、識別プログラム110は、書込前の対象プログラムの情報を取得する(S1004)。取得する情報には、例えば、ハッシュ値、ファイルパス、署名者名、作成者名、作成会社名などが含まれる。

【０１１４】

プロセスによる書き込みが終了すると、識別プログラム110は、書込後の対象プログラムの情報を取得する(S1005)。取得する情報には、例えば、ハッシュ値、ファイルパス、署名者名、作成者、作成会社名などが含まれる。そして、書き込み前後の対象プログラムの情報を比較して、上書アップデートが行われたか否かを判定する(S1006)。この判定は、例えば、ファイルパスが同一か、ファイル名が同一か、または、署名者名が同一かなどの条件で行えばよく、判定条件は一つでもよいし、複数でもよい。

【０１１５】

上書アップデートではないと判定された場合、処理は前述したステップS1007に移行する。また、上書アップデートと判定した場合、識別プログラム110は、書込前（上書アップデート前）のハッシュ値に基づきホワイトリスト120の許可情報（設定情報）を取得する(S1008)。許可情報は、例えば、昇格権限、通信が許可されている接続先ポート、通信が許可されている接続先IPアドレスなどである（図3参照）。

【０１１６】

次に、登録プログラム111は、ステップS1005で取得された書込後（上書アップデート後）のハッシュ値とステップS1008で取得された許可情報を識別プログラム110から受け取る。そして、上書アップデート後のハッシュ値と許可情報を、上書アップデートされた対象プログラムに関連付けてホワイトリスト120に登録し(S1009)、書込要求に対する処理が終了する。

【０１１７】

以上の処理により、上書アップデート前のホワイトリスト120の設定を継続するアップデートが可能になる。なお、ステップS1009において、上書アップデート前のハッシュ値や許可情報などは、ホワイトリスト120から削除してもよいし、ホワイトリスト120に残し

てもよい。上書アップデート前のハッシュ値や許可情報などをホワイトリスト120に残せば、アップデートされたプログラムの不具合等によりアップデート前のプログラムに戻す場合や、他のユーザが当該アップデートを未実施の場合の対応が可能になる。

【０１１８】

●置き換えを検知してホワイトリストをアップデート

上記では上書アップデートの例を説明したが、以下では、置換アップデートについて説明する。置換アップデートとは、例えば、ファイルを削除または移動、あるいは、ファイル名を変更した後、アップデート後のファイルを作成または移動、あるいは、ファイル名を変更するなどにより、プログラムをアップデートする処理のことである。

【０１１９】

図11のフローチャートにより置換アップデートによるホワイトリスト120の更新処理を説明する。図11は元ファイルの削除要求またはファイル名の変更要求に対する処理例を示す。

【０１２０】

任意のプロセスからファイルの削除要求またはファイル名の変更要求が発生すると、検知プログラム112は、当該削除要求または変更要求（以下、要求）を検知し(S1101)、当該要求を行ったプロセスが昇格権限をもつか否かを判定する(S1102)。当該プロセスが昇格権限をもたない場合、検知プログラム112は、当該要求に対する処理を終了し、再び、プロセスからの削除要求または変更要求を監視する。

【０１２１】

一方、要求を行ったプロセスが昇格権限をもつ場合、識別プログラム110は、要求から削除またはファイル名を変更するプログラム（以下、対象プログラム）を特定する。例えば、ファイルパス、ファイル名から対象プログラムを特定すればよい。そして、対象プログラムのハッシュ値を取得し、そのハッシュ値に基づき対象プログラムがホワイトリスト120に登録されているか否かを判定する(S1103)。

【０１２２】

ホワイトリスト120に未登録の対象プログラムの場合、識別プログラム110は、要求がファイル名の変更要求か否かを判定する(S1106)。ファイル名の変更要求ではない（つまり削除要求）場合、識別プログラム110は当該要求に対する処理を終了し、検知プログラム112は、再び、プロセスからのファイルの削除要求またはファイル名の変更要求を監視する。また、ファイル名の変更要求の場合、処理は図5または図6に示すホワイトリスト制御処理に移行し、対象プログラムがホワイトリスト120に追加され(S1107)、要求に対する処理が終了する。

【０１２３】

他方、ホワイトリスト120に既登録の対象プログラムの場合、識別プログラム110は、削除またはファイル名変更前の対象プログラムの情報を取得する(S1104)。取得する情報には、例えば、ハッシュ値、ファイルパス、署名者名、作成者名、作成会社名などが含まれる。

【０１２４】

次に、登録プログラム111は、要求を発行したプロセスのプロセス名およびプロセスID、対象プログラムのファイル名、並びに、ステップS1104で取得された情報を識別プログラム110から受け取る。そして、受け取った情報をリストに登録し(S1105)、要求に対する処理が終了する。図12により登録プログラム111が情報を登録するリストの一例を示す。

【０１２５】

図13のフローチャートにより置換アップデートによるホワイトリスト120の更新処理を説明する。図13はファイルの作成要求またはファイル名の変更要求に対する処理例を示す。

【０１２６】

任意のプロセスからファイルの作成要求またはファイル名の変更要求が発生すると、検知プログラム112は、当該作成要求または変更要求（以下、要求）を検知し(S1201)、当該

要求を行ったプロセスが昇格権限をもつか否かを判定する(S1202)。当該プロセスが昇格権限をもたない場合、検知プログラム112は、当該要求に対する処理を終了し、再び、プロセスからの作成要求または変更要求を監視する。

【０１２７】

一方、要求を行ったプロセスが昇格権限をもつ場合、識別プログラム110は、ファイルの作成またはファイル名の変更が終了したか否かを判定する(S1203)。そして、ファイルの作成またはファイル名の変更が終了すると、作成されたファイルまたはファイル名が変更されたファイル（以下、対象プログラム）の情報を取得する(S1204)。取得する情報には、例えば、対象プログラムのファイル名、ハッシュ値、ファイルパス、署名者名、作成者名、作成会社名などが含まれる。

【０１２８】

次に、識別プログラム110は、要求を発行したプロセスのプロセス名およびプロセスID、並びに、ステップS1204で取得した情報と、図12に示すリストの登録情報を比較する(S1205)。そして、置換アップデートが行われたか否かを判定する(S1206)。この判定は、例えば、プロセスIDが同一か、ファイルパスが同一か、ファイル名が同一か、または、署名者名が同一かなどの条件で行えばよく、判定条件は一つでもよいし、複数でもよい。

【０１２９】

置換アップデートではないと判定された場合、処理は図5または図6に示すホワイトリスト制御処理に移行し、対象プログラムがホワイトリスト120に追加され(S1207)、要求に対する処理が終了する。

【０１３０】

他方、置換アップデートと判定した場合、識別プログラム110は、リストの対象プログラムに対応するレコードのハッシュ値に基づきホワイトリスト120の許可情報を取得する(S1208)。許可情報は、例えば、昇格権限、通信が許可されている接続先ポート、通信が許可されている接続先IPアドレスなどである（図3参照）。

【０１３１】

次に、登録プログラム111は、置換アップデート後の対象プログラムのハッシュ値とステップS1208で取得された許可情報を識別プログラム110から受け取る。そして、置換アップデート後のハッシュ値と許可情報を、置換アップデートされた対象プログラムに関連付けてホワイトリスト120に登録し(S1209)、要求に対する処理が終了する。

【０１３２】

以上の処理により、置換アップデート前のホワイトリスト120の設定を継続するアップデートが可能になる。なお、ステップS1208において、置換アップデート前のハッシュ値や許可情報などは、ホワイトリスト120から削除してもよいし、ホワイトリスト120に残してもよい。置換アップデート前のハッシュ値や許可情報などをホワイトリスト120に残せば、アップデートされたプログラムの不具合等によりアップデート前のプログラムに戻す場合や、他のユーザが当該アップデートを未実施の場合の対応が可能になる。

【０１３３】

上記では、クライアント10の例としてPC、タブレット端末、スマートフォンを例に挙げた。しかし、ポインティングデバイスを持たない端末（例えばネットワークスキャナ）や、ディスプレイを持たない端末（例えば組込端末）などをクライアントとして、上記処理を適用することができる。

【０１３４】

上記では、ホワイトリストを用いたホワイト制御システムについて説明したが、ホワイトリストに限定されず、ブラックリストを用いたブラックリスト制御システムにも上記処理を適用することができる。

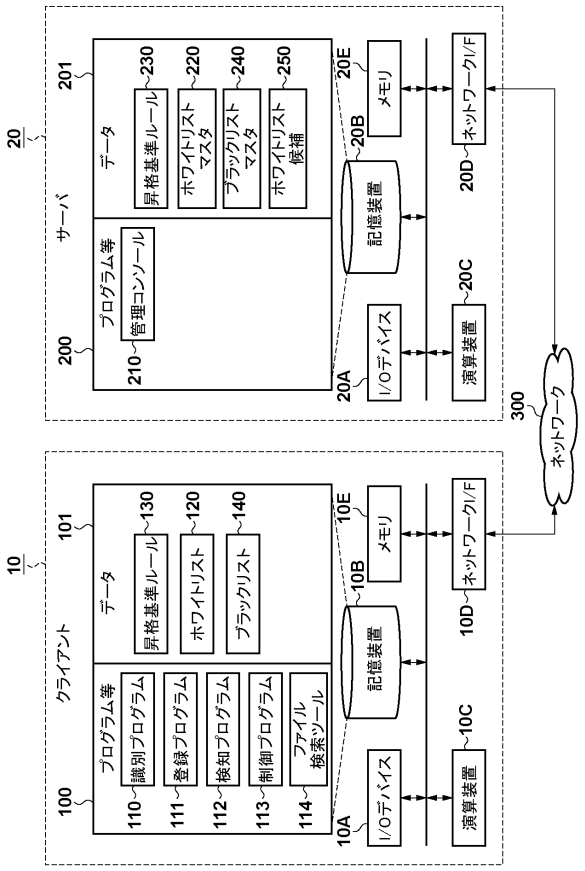
【０１３５】

［その他の実施例］

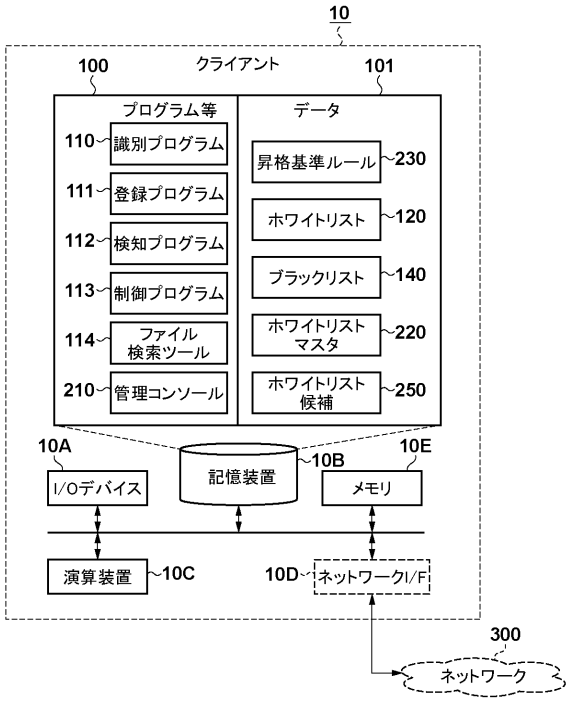
また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記録媒体

を介してシステム或いは装置に供給し、そのシステムあるいは装置のコンピュータ（又はCPUやMPU等）がプログラムを読み出して実行する処理である。

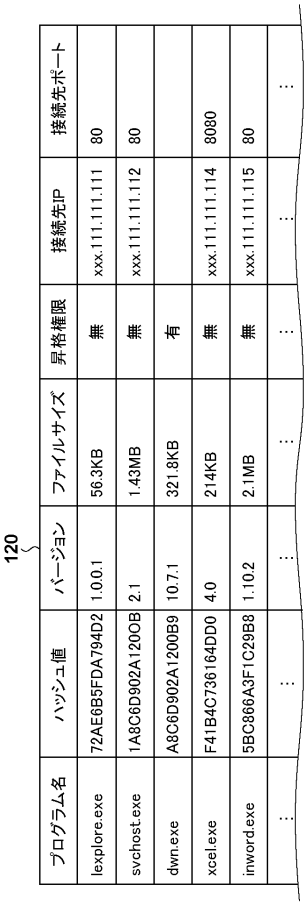
【図１】



【図２】



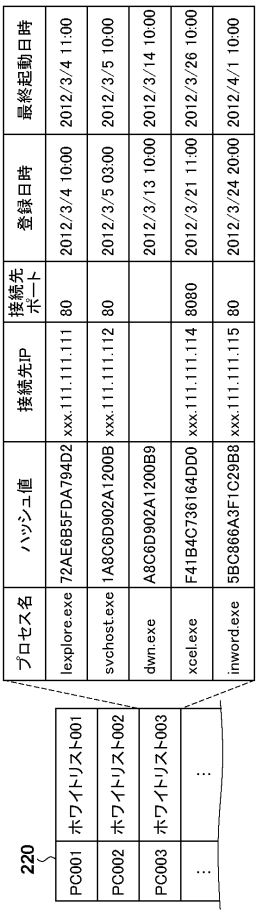
【図３】



120

プログラム名	ハッシュ値	バージョン	ファイルサイズ	昇格権限	接続先IP	接続先ポート
lexplore.exe	72AE6B5FDA794D2	1.0.0.1	56.3KB	無	xxx.111.111.111	80
svchost.exe	1A8C6D902A120OB	2.1	1.43MB	無	xxx.111.111.112	80
dwn.exe	A8C6D902A1200B9	10.7.1	321.8KB	有		
xcel.exe	F41B4C7361164DD0	4.0	214KB	無	xxx.111.111.114	8080
inword.exe	5BC866A3F1C29B8	1.10.2	2.1MB	無	xxx.111.111.115	80
…	…	…	…	…	…	…

【図４】

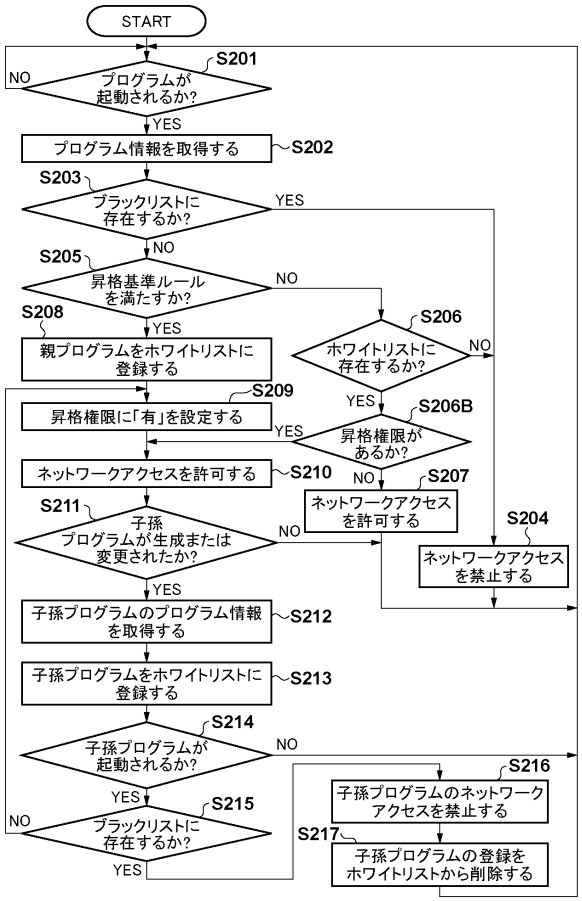


プロセス名	ハッシュ値	接続先IP	接続先ポート	登録日時	最終起動日時
lexplore.exe	72AE6B5FDA794D2	xxx.111.111.111	80	2012/3/4 10:00	2012/3/4 11:00
svchost.exe	1A8C6D902A1200B	xxx.111.111.112	80	2012/3/5 03:00	2012/3/5 10:00
dwn.exe	A8C6D902A1200B9			2012/3/13 10:00	2012/3/14 10:00
xcel.exe	F41B4C736164DD0	xxx.111.111.114	8080	2012/3/21 11:00	2012/3/26 10:00
inword.exe	5BC866A3F1C29B8	xxx.111.111.115	80	2012/3/24 20:00	2012/4/1 10:00

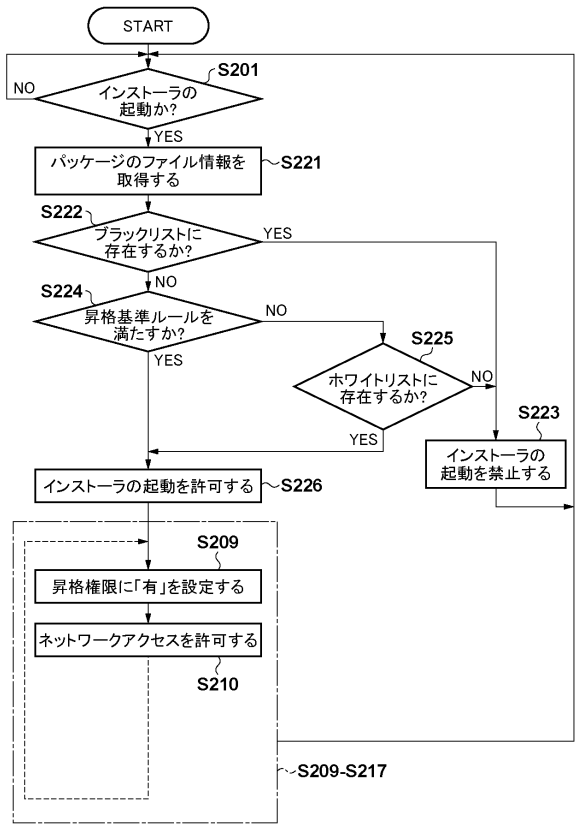
220

PC001	ホワイトリスト-001	ホワイトリスト-002	ホワイトリスト-003	…
PC002	…			
PC003	…			

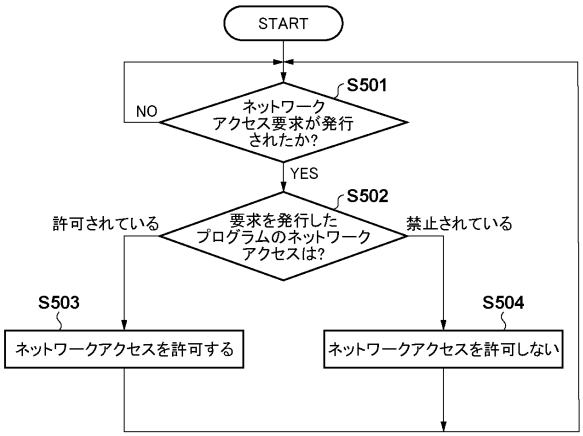
【図５】



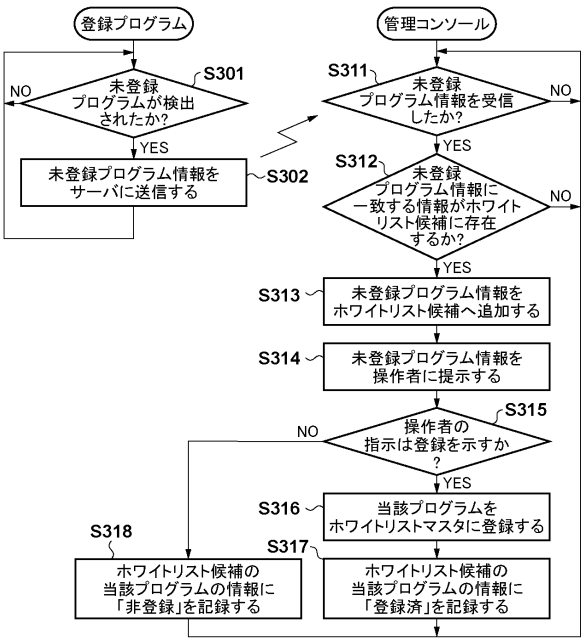
【図６】



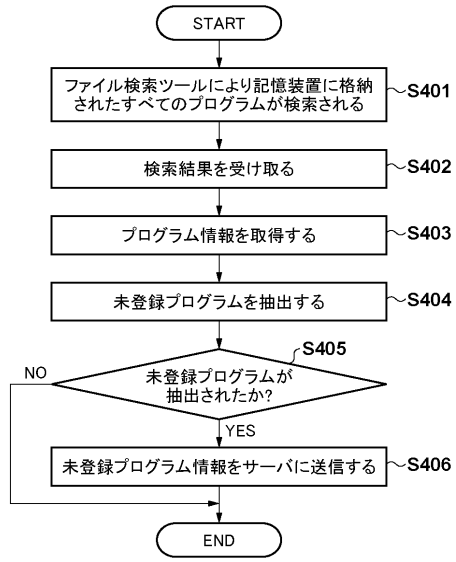
【図７】


START
S501
ネットワーク
アクセス要求が発行
されたか?
NO
YES
S502
要求を発行した
プログラムのネットワーク
アクセスは?
許可されている
禁止されている
S503
ネットワークアクセスを許可する
S504
ネットワークアクセスを許可しない

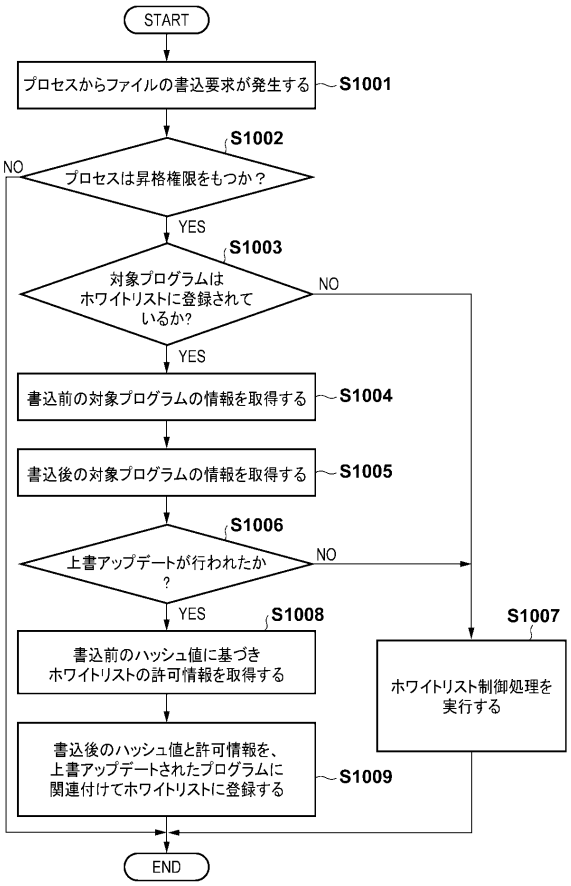
【図８】


登録プログラム
管理コンソール
S301
未登録
プログラムが検出
されたか?
NO
YES
未登録プログラム情報を
サーバに送信する
S302
S311
未登録
プログラム情報を受信
したか?
NO
YES
S312
未登録
プログラム情報に
一致する情報がホワイト
リスト候補に存在
するか?
NO
YES
S313
未登録プログラム情報を
ホワイトリスト候補へ追加する
S314
未登録プログラム情報を
操作者に提示する
S315
操作者の
指示は登録を示すか
?
NO
YES
S316
当該プログラムを
ホワイトリストマスタに登録する
S317
ホワイトリスト候補の
当該プログラムの情報に
「登録済」を記録する
S318
ホワイトリスト候補の
当該プログラムの情報に
「非登録」を記録する

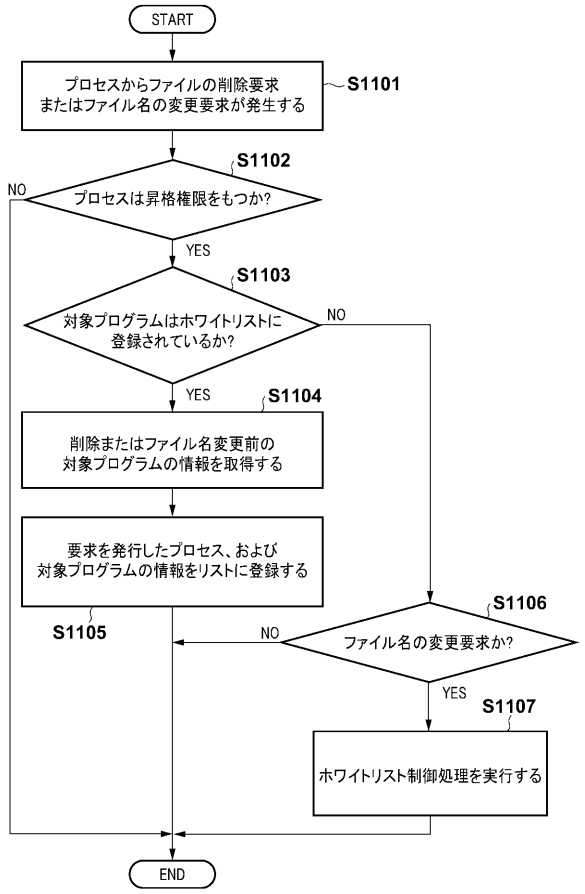
【図９】


START
ファイル検索ツールにより記憶装置に格納
されたすべてのプログラムが検索される
S401
検索結果を受け取る
S402
プログラム情報を取得する
S403
未登録プログラムを抽出する
S404
S405
未登録プログラムが
抽出されたか?
NO
YES
未登録プログラム情報をサーバに送信する
S406
END

【図１０】


START
プロセスからファイルの書込要求が発生する
S1001
S1002
プロセスは昇格権限をもつか？
NO
YES
S1003
対象プログラムは
ホワイトリストに登録されて
いるか?
NO
YES
書込前の対象プログラムの情報を取得する
S1004
書込後の対象プログラムの情報を取得する
S1005
S1006
上書アップデートが行われたか
?
NO
YES
S1008
書込前のハッシュ値に基づき
ホワイトリストの許可情報を取得する
書込後のハッシュ値と許可情報を、
上書アップデートされたプログラムに
関連付けてホワイトリストに登録する
S1009
S1007
ホワイトリスト制御処理を
実行する
END

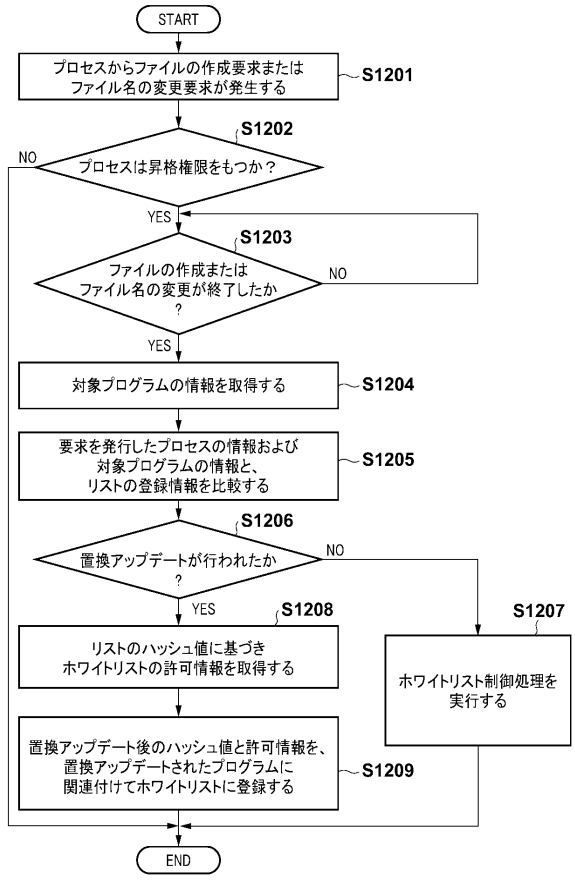
【図１１】



【図１２】

作成会社名	ABC Inc.	BCD Inc.	ABC Inc.	・・・
作成者名	ABC	BCD	CDE	・・・
署名者名	A Sign	Sign B	A Sign	・・・
ファイルパス	C:\test\A.exe	C:\test\B.exe	C:\test\C.exe	・・・
ハッシュ値	72ABC758...	1AFC412...	EFD1C32...	・・・
ファイル名	A.exe	B.exe	C.exe	・・・
プロセスID	100	200	300	・・・
プロセス名	TestA.exe	TestB.exe	TestC.exe	・・・

【図１３】



フロントページの続き

(72)発明者　米川　智
埼玉県秩父市下影森１２４８番地　キヤノン電子株式会社内
(72)発明者　高野　和希
埼玉県秩父市下影森１２４８番地　キヤノン電子株式会社内

審査官　大塚　俊範

(56)参考文献　特開２０１２－１８５７４５（ＪＰ，Ａ）
特開２０１０－２３８１６８（ＪＰ，Ａ）
国際公開第２０１２／０４６４０６（WO，Ａ１）
特開２００７－３１６７８０（ＪＰ，Ａ）
植松 建至，構造計算書不正検知システムの提案，情報処理学会論文誌　論文誌ジャーナル［ＣＤ－ＲＯＭ］，日本，社団法人情報処理学会，２００８年　９月１５日，第49巻，第9号，第3199-3208頁，ISSN:1882-7837

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４４５
Ｇ０６Ｆ　１１／００