(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2010-182020
(P2010-182020A)
(43)公開日　平成22年8月19日(2010.8.19)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/22	(2006.01)	G06F	9/06	660N	5B276
G06F 21/20	(2006.01)	G06F	15/00	330A	5B285

審査請求 未請求　請求項の数 8　ＯＬ　（全 17 頁）

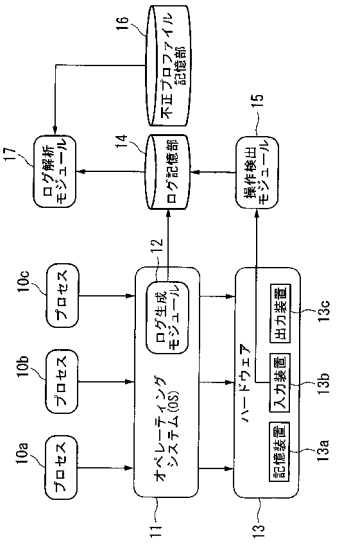
(21)出願番号　特願2009-23850 (P2009-23850)
(22)出願日　平成21年2月4日 (2009.2.4)

(71)出願人　000208891
ＫＤＤＩ株式会社
東京都新宿区西新宿二丁目３番２号
(71)出願人　304023318
国立大学法人静岡大学
静岡県静岡市駿河区大谷８３６
(74)代理人　100106909
弁理士　棚井　澄雄
(74)代理人　100064908
弁理士　志賀　正武
(74)代理人　100146835
弁理士　佐伯　義文
(74)代理人　100138759
弁理士　大房　直樹

最終頁に続く

(54)【発明の名称】不正検知装置およびプログラム

(57)【要約】
【課題】ボットを検知すると共にボットの種別を知ることができる不正検知装置およびプログラムを提供する。
【解決手段】正常プロファイル記憶部１６は、既知のボットに感染している端末で生成された第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、ボットの種別を示すボット種別情報とを記憶する。ログ生成モジュール１２は、監視対象の端末で生成された第２のプロセスの挙動を監視し、当該第２のプロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報を生成する。ログ解析モジュール１７は、第１の識別情報と第２の識別情報を比較し、両者が一致した場合に、第１の識別情報に対応したボット種別情報を正常プロファイル記憶部１６から取得する。
【選択図】図１



【特許請求の範囲】
【請求項１】
既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、
監視対象の端末で生成された第２のプロセスの挙動を監視し、当該第２のプロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報を生成する情報生成手段と、
前記第１の識別情報と前記第２の識別情報を比較し、両者が一致した場合に、前記第１の識別情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、
を備えたことを特徴とする不正検知装置。
【請求項２】
既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、
監視対象の端末で生成された親プロセスの挙動と、当該親プロセスによって起動される子プロセスの挙動とを監視し、前記親プロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報と、前記子プロセスが操作を行ったファイルまたはフォルダを識別する第３の識別情報とを生成する情報生成手段と、
前記第１の識別情報と前記第２の識別情報を比較すると共に前記第１の識別情報と前記第３の識別情報を比較し、前記第１の識別情報と前記第２の識別情報が一致した場合、前記第１の識別情報と前記第３の識別情報が一致した場合、または前記第１の識別情報と前記第２の識別情報が一致すると共に前記第１の識別情報と前記第３の識別情報が一致した場合に、前記第１の識別情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、
を備えたことを特徴とする不正検知装置。
【請求項３】
既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報を含み、前記第１のプロセスがファイルまたはフォルダに操作を行ったときの手順を示す第１の手順情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、
監視対象の端末で生成された第２のプロセスの挙動を監視し、当該第２のプロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報と時刻情報を含む情報を生成する第１の情報生成手段と、
前記第１の情報生成手段が生成した情報に基づいて、前記第２の識別情報を含み、前記第２のプロセスがファイルまたはフォルダに操作を行ったときの手順を示す第２の手順情報を生成する第２の情報生成手段と、
前記第１の手順情報と前記第２の手順情報を比較し、両者が一致した場合に、前記第１の手順情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、
を備えたことを特徴とする不正検知装置。
【請求項４】
既知のボットに感染している端末で生成された、前記ボットによる第１の親プロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記第１の親プロセスによって起動される、前記ボットによる第１の子プロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報とを含み、前記第１の親プロセスと前記第１の子プロセスがファイルまたはフォルダに操作を行ったときの手順を示す第１の手順情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、
監視対象の端末で生成された第２の親プロセスの挙動と、当該第２の親プロセスによって起動される第２の子プロセスの挙動とを監視し、前記第２の親プロセスが操作を行ったファイルまたはフォルダを識別する第３の識別情報および時刻情報を含む情報と、前記第

２の子プロセスが操作を行ったファイルまたはフォルダを識別する第４の識別情報および時刻情報を含む情報とを生成する第１の情報生成手段と、
前記第１の情報生成手段が生成した情報に基づいて、前記第３の識別情報と前記第４の識別情報を含み、前記第２の親プロセスと前記第２の子プロセスがファイルまたはフォルダに操作を行ったときの手順を示す第２の手順情報を生成する第２の情報生成手段と、
前記第１の手順情報と前記第２の手順情報を比較し、両者が一致した場合に、前記第１の手順情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、
を備えたことを特徴とする不正検知装置。
【請求項５】
前記第２のプロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする請求項１または請求項３に記載の不正検知装置。
【請求項６】
前記親プロセスと前記子プロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする請求項２に記載の不正検知装置。
【請求項７】
前記第２の親プロセスと前記第２の子プロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする請求項４に記載の不正検知装置。
【請求項８】
請求項１～請求項７のいずれかに記載の不正検知装置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、ボットによる不正を検知する不正検知装置に関する。また、本発明は、本不正検知装置としてコンピュータを機能させるためのプログラムにも関する。
【背景技術】
【０００２】
近年、ウィルスに感染したコンピュータに悪質な動作を実行させる、ボットと呼ばれるウィルスによる被害が拡大している。ボットは、外部の指令サーバに通信セッションを確立して新たなコードをダウンロードする機能や、攻撃のための指令を受ける機能、指令に従って攻撃する機能などを持つ悪意のコードで構成されている。
【０００３】
しかし、パターンマッチング型のウィルス対策ソフトで検知できないボットが増えている。そこで、ボットを検知する手法として、ボットがPC内の複数のファイルに感染するときに自身のコードを読み込む（Read）行為や証拠隠滅のためにコードを消去する（Delete）行為に着目した検知手法が提案されている（例えば非特許文献１参照）。
【先行技術文献】
【非特許文献】
【０００４】
【非特許文献１】酒井崇裕、長谷巧、竹森敬祐、西垣正勝、“自己ファイルREAD/DELETEの検出によるボット検知の可能性に関する一検討”、マルウェア対策研究人材育成ワークショップ2008（MWS2008）、セッションM6-2、2008年10月
【発明の概要】
【発明が解決しようとする課題】
【０００５】
ボットは、ユーザファイルにアクセスする情報漏洩型、既存のEXEファイル（実行ファイル）にアクセスするトロイの木馬型、スパムメールに利用する宛先や本文を保存したファイルにアクセスするスパムメール型などに分類される。上記の手法によりボットの検知は可能であるが、ボットの種別を把握することはできない。

【０００６】
本発明は、上述した課題に鑑みてなされたものであって、ボットを検知すると共にボットの種別を知ることができる不正検知装置およ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】
【０００７】
本発明は、上記の課題を解決するためになされたもので、既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、監視対象の端末で生成された第２のプロセスの挙動を監視し、当該第２のプロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報を生成する情報生成手段と、前記第１の識別情報と前記第２の識別情報を比較し、両者が一致した場合に、前記第１の識別情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、を備えたことを特徴とする不正検知装置である。
【０００８】
また、本発明は、既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、監視対象の端末で生成された親プロセスの挙動と、当該親プロセスによって起動される子プロセスの挙動とを監視し、前記親プロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報と、前記子プロセスが操作を行ったファイルまたはフォルダを識別する第３の識別情報とを生成する情報生成手段と、前記第１の識別情報と前記第２の識別情報を比較すると共に前記第１の識別情報と前記第３の識別情報を比較し、前記第１の識別情報と前記第２の識別情報が一致した場合、前記第１の識別情報と前記第３の識別情報が一致した場合、または前記第１の識別情報と前記第２の識別情報が一致すると共に前記第１の識別情報と前記第３の識別情報が一致した場合に、前記第１の識別情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、を備えたことを特徴とする不正検知装置である。
【０００９】
また、本発明は、既知のボットに感染している端末で生成された、前記ボットによる第１のプロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報を含み、前記第１のプロセスがファイルまたはフォルダに操作を行ったときの手順を示す第１の手順情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、監視対象の端末で生成された第２のプロセスの挙動を監視し、当該第２のプロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報と時刻情報を含む情報を生成する第１の情報生成手段と、前記第１の情報生成手段が生成した情報に基づいて、前記第２の識別情報を含み、前記第２のプロセスがファイルまたはフォルダに操作を行ったときの手順を示す第２の手順情報を生成する第２の情報生成手段と、前記第１の手順情報と前記第２の手順情報を比較し、両者が一致した場合に、前記第１の手順情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、を備えたことを特徴とする不正検知装置である。
【００１０】
また、本発明は、既知のボットに感染している端末で生成された、前記ボットによる第１の親プロセスが操作を行ったファイルまたはフォルダを識別する第１の識別情報と、前記第１の親プロセスによって起動される、前記ボットによる第１の子プロセスが操作を行ったファイルまたはフォルダを識別する第２の識別情報とを含み、前記第１の親プロセスと前記第１の子プロセスがファイルまたはフォルダに操作を行ったときの手順を示す第１の手順情報と、前記ボットの種別を示すボット種別情報とを記憶する記憶手段と、監視対象の端末で生成された第２の親プロセスの挙動と、当該第２の親プロセスによって起動される第２の子プロセスの挙動とを監視し、前記第２の親プロセスが操作を行ったファイルまたはフォルダを識別する第３の識別情報および時刻情報を含む情報と、前記第２の子プ

ロセスが操作を行ったファイルまたはフォルダを識別する第４の識別情報および時刻情報を含む情報とを生成する第１の情報生成手段と、前記第１の情報生成手段が生成した情報に基づいて、前記第３の識別情報と前記第４の識別情報を含み、前記第２の親プロセスと前記第２の子プロセスがファイルまたはフォルダに操作を行ったときの手順を示す第２の手順情報を生成する第２の情報生成手段と、前記第１の手順情報と前記第２の手順情報を比較し、両者が一致した場合に、前記第１の手順情報に対応した前記ボット種別情報を前記記憶手段から取得する情報取得手段と、を備えたことを特徴とする不正検知装置である。

【００１１】

また、本発明の不正検知装置において、前記第２のプロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする。

【００１２】

また、本発明の不正検知装置において、前記親プロセスと前記子プロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする。

【００１３】

また、本発明の不正検知装置において、前記第２の親プロセスと前記第２の子プロセスは、ユーザが前記監視対象の端末を操作していないときに生成されたプロセスであることを特徴とする。

【００１４】

また、本発明は、上記の不正検知装置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】

【００１５】

本発明によれば、ボットに感染した正常な端末で生成されたプロセスによるファイルまたはフォルダの操作と同一の操作が検知された場合に、その操作の元となったボットの種別をボット種別情報から識別することが可能となるので、ボットを検知すると共にボットの種別を知ることができる。

【図面の簡単な説明】

【００１６】

【図１】本発明の一実施形態による不正検知装置の構成を示すブロック図である。
【図２】本発明の一実施形態におけるログに含まれる情報を示す参考図である。
【図３】本発明の一実施形態におけるログに含まれる情報を示す参考図である。
【図４】本発明の一実施形態における処理のイメージを示す参考図である。
【図５】本発明の一実施形態における処理のイメージを示す参考図である。
【図６】本発明の一実施形態におけるファイル操作の手順を示す参考図である。
【図７】本発明の一実施形態におけるファイル操作の手順を示す参考図である。
【図８】本発明の一実施形態におけるファイル操作の手順を示す参考図である。
【図９】本発明の一実施形態におけるファイル操作の手順を示す参考図である。

【発明を実施するための形態】

【００１７】

以下、図面を参照し、本発明の実施形態を説明する。図１は、本実施形態による不正検知装置の構成を示している。本不正検知装置は、ファイルまたはフォルダの状態を変更する操作と、その操作を実行するプロセスとを関連付けることによって、不正の可能性がある操作の詳細を簡易に抽出する。これを達成するために、システムコール処理をフックして、ファイル操作（ファイルの変更・削除）に関するログを生成し、そのログを解析する仕組みが設けられている。

【００１８】

監視対象の端末で動作するOS１１のカーネル部分には、ログ生成モジュール１２が設けられている。このログ生成モジュール１２は、各種のアプリケーションプロセスであるプ

ロセス１０ａ，１０ｂ，１０ｃが記憶装置１３ａ、入力装置１３ｂ、出力装置１３ｃなどのハードウェア１３に対してアクセスを行う際にOS１１に発行したシステムコール処理をフックしてログを生成する。

【００１９】

Linux（登録商標）には、カーネルにおいてセキュリティ機能を拡張するフレームワークであるLinux（登録商標） Security Module（LSM）が実装されている。LSMでは、ファイルやプロセスの操作が行われた際に、ユーザが定義したセキュリティ検証機構を呼び出して権限の検証やログの生成を行うための監視ポイントが設けられている。本ログ解析システムのシステムコール処理のフックは、LSMの監視ポイントにおけるセキュリティ検証機構として実装される。

【００２０】

ログ生成モジュール１２が生成したログはログ記憶部１４に格納され、記憶される。操作検出モジュール１５は、ユーザがマウスやキーボード等の入力装置１３ｂを操作したことを検出し、操作時刻を含むログ（以下、操作ログと記載する）を生成する。操作検出モジュール１５が生成した操作ログはログ記憶部１４に格納され、記憶される。

【００２１】

不正プロファイル記憶部１６は、ボットに感染した端末で取得された、ファイルまたはフォルダに対する不正な操作の情報を含む不正プロファイルを記憶する。不正プロファイルは、ボットに感染した端末のプロセス（ボットによるプロセス）の挙動を監視した結果に基づいて生成されたものである。プロセスの挙動を監視する手法は、ログ生成モジュール１２がプロセスの挙動を監視する手法と同様である。ログ解析モジュール１７は、ログ記憶部１４に格納されたログを解析し、不正の有無を判定する。

【００２２】

次に、ログ生成モジュール１２が生成するログの詳細を説明する。LSMには、ファイル処理、プログラムの実行処理、通信処理など、およそ160の処理に関して、監視ポイントが設けられている。本実施形態において、ログ生成モジュール１２は、ファイル操作に関する監視ポイントによりログを生成する。本実施形態では、ファイルの読み書きを監視する監視ポイントである「file_permission」と、ファイルの削除を監視する監視ポイントである「inode_delete」とがログを生成する。

【００２３】

図２および図３は、ログ生成モジュール１２が生成するログに含まれる情報を示している。このログに含まれる情報はヘッダ情報と監視ポイント固有情報に大別される。ヘッダ情報は、各監視ポイントに対応するログに共通して記録される情報である。図２はヘッダ情報の内容を示している。具体的には、ログが記録された時刻２００、監視ポイントの名称２０２、処理を行ったプロセスのID２０４（pid）、ユーザID２０６（uid）、グループID２０８（gid）、親プロセスのID２１０（parent）、親プロセスの名称２１２（parent cmd）、および処理を行ったプロセスの名称２１４（cmd）が記録される。

【００２４】

監視ポイント固有情報は、フック処理に渡される引数の情報に応じて監視ポイント毎に記録される情報である。図３は、ファイルの読み書きを監視する「file_permission」によって記録される監視ポイント固有情報の内容を示している。情報３００（inode_num）は、ファイルに割り当てられた固有の識別子である。情報３０２（fowner）は、ファイルの所有者を示す固有の識別子である。情報３０４（fgrp）は、ファイルの所属するグループを示す固有の識別子である。情報３０６（mode）は、ファイルに対する読み込み・書き込みを識別するである。情報３０６の値はOSに固有の値であるが、この値を読み取ることで、操作内容（読み込み／書き込み）を把握することが可能である。情報３０８（path）は、操作対象となるファイルの名称と、ファイルが存在するフォルダの名称とを含む情報である。図３に示した例の場合、「/home/example/」がフォルダの名称であり、「path.txt」がファイルの名称である。

【００２５】

以下では、独立して識別可能なファイル操作に関する１つのヘッダ情報と１つの監視ポイント固有情報からなる情報を単位ログとする。ログ記憶部１４に格納されているログは単位ログの集合体である。

【００２６】

次に、ログ解析モジュール１７によるログの解析方法を説明する。以下では、ファイルに対する操作を不正検知の対象として説明を行うが、フォルダに対する操作を不正検知の対象とする場合も同様である。

【００２７】

ユーザがマウスやキーボード等を操作することによるファイルの操作を誤って検知する可能性がある。そこで、本実施形態では、ユーザがマウスやキーボード等を操作していない期間のファイルの操作を不正検知の対象とする。ただし、この期間のファイル操作を異常検知の対象とすることは必須ではなく、この期間以外のファイル操作も異常検知の対象としてもよい。

【００２８】

ログ解析モジュール１７は、ログ生成モジュール１２が生成したログをログ記憶部１４から読み出すと共に、操作検出モジュール１５が生成した操作ログをログ記憶部１４から読み出す。操作ログには、ユーザがマウスやキーボード等を操作した時刻が記録されており、ログ解析モジュール１７は、ユーザが操作を行った時刻を基準とする所定期間を操作期間であると認識する。ログ解析モジュール１７は、ログ生成モジュール１２が生成したログのうち、時刻（図２の時刻２００）が操作期間に含まれないログを抽出する。このようにして抽出されたログが以降の処理で使用される。以下では、上記のようにして抽出されたログを処理対象のログとする。

【００２９】

上記以降の処理として、以下では４つの処理例を説明する。

【００３０】

（第１の処理例）

まず、第１の処理例を説明する。第１の処理例では、既知のボットによるプロセスがアクセスするファイルにプロセスがアクセスした場合に、ボットによる不正が発生したと判定する。図４は、第１の処理例による処理のイメージを示している。

【００３１】

不正プロファイル記憶部１６に格納されている不正プロファイルには、ボットに感染した端末のプロセス（ボットによるプロセス）がアクセスしたファイルを識別する情報（本実施形態ではファイル名称）が、ボットの種別（情報漏洩型／トロイの木馬型／スパムメール型）を示す情報（以下、ボット種別情報と記載する）と関連付けられて記録されている。図４では、ユーザがアクセス可能なユーザファイルの名称が情報漏洩型のボット種別情報と共に記録され、既存のEXEファイル（実行ファイル）の名称がトロイの木馬型のボット種別情報と共に記録され、スパムメールに利用する宛先や本文を保存したファイルの名称がスパムメール型のボット種別情報と共に記録されている。ボットに感染した端末では、ボットによるプロセスだけでなく、正常なプロセスも動作する。そこで、プロセスの挙動を監視して得られた情報をユーザがチェックし、ボットによるプロセスであると確認できたプロセスの情報から不正プロファイルを構成することが望ましい。

【００３２】

監視対象の端末ではプロセス４００，４１０，４２０が動作している。プロセス４００はユーザファイルにアクセスしているため、情報漏洩型のボットによる不正が発生していると判定される。また、プロセス４１０はEXEファイルにアクセスしているため、トロイの木馬型のボットによる不正が発生していると判定される。また、プロセス４２０は、スパムメールに利用する宛先や本文を保存したファイルにアクセスしているため、スパムメール型のボットによる不正が発生していると判定される。

【００３３】

第１の処理例では、ログ解析モジュール１７は以下のように動作する。まず、ログ解析

モジュール１７は不正プロファイル記憶部１６から不正プロファイルを読み出す。続いて、ログ解析モジュール１７は、処理対象のログに含まれる単位ログに記録されているファイル名称（図３の情報３０８）と、不正プロファイルに含まれるファイル名称とを比較する。

【００３４】

不正プロファイルには複数のファイル名称が記録されている。単位ログに記録されているファイル名称が、不正プロファイルに記録されているいずれかのファイル名称と一致した場合、ログ解析モジュール１７は、不正が発生したと判定する。さらに、ログ解析モジュール１７は、不正が発生したと判定したときのファイル名称と関連付けられているボット種別情報を不正プロファイルから取得し、ボット種別情報に基づいてボットの種別を判定する。

【００３５】

単位ログに記録されているファイル名称が、不正プロファイルに記録されているいずれのファイル名称とも一致しなかった場合、ログ解析モジュール１７は、不正が発生していないと判定する。処理対象のログに複数の単位ログが存在する場合には、ログ解析モジュール１７は上記の処理を繰り返す。

【００３６】

上記の処理によって、ボットを検知すると共にボットの種別を知ることができる。なお、単位ログに記録されているファイル名称とプロセス名称の組合せが、不正プロファイルに記録されているファイル名称とプロセス名称の組合せと一致した場合に不正が発生したと判定するようにしてもよい。

【００３７】

（第２の処理例）

次に、第２の処理例を説明する。ボットに感染した端末では、親プロセスが子プロセスを起動して、親プロセスと子プロセスが所望の処理を共同で行う場合がある。第２の処理例では、親子のプロセスによる不正の有無を判定する。図５は、第２の処理例による処理のイメージを示している。

【００３８】

第１の処理例と同様に、不正プロファイル記憶部１６に格納されている不正プロファイルには、ボットに感染した端末のプロセスがアクセスしたファイルを識別する情報（ファイル名称）が、ボット種別情報と関連付けられて記録されている。また、監視対象の端末では、親プロセス５００と、親プロセス５００によって起動された子プロセス５１０，５２０とが動作している。

【００３９】

親プロセス５００は、ボットに感染した端末のプロセスがアクセスしたファイルと異なるファイルにアクセスしているため、不正ではないと判定されるが、子プロセス５１０，５２０は、ボットに感染した端末のプロセスがアクセスしたファイルと同一のファイルにアクセスしているため、不正であると判定される。この結果、親プロセス５００、子プロセス５１０，５２０を含むグループが不正（ボット）であると判定される。

【００４０】

また、子プロセス５１０は、情報漏洩型のボットによるプロセスであると判定され、子プロセス５２０は、スパムメール型のボットによるプロセスであると判定される。この結果、親プロセス５００、子プロセス５１０，５２０を含むグループは、情報漏洩型かつスパムメール型のボットによるプロセスのグループであると判定される。このように、親プロセスまたは子プロセスの単位で複数種類のボットが検知された場合には、親子のプロセスを含むグループは、複数種類のボットによるプロセスのグループであると判定される。

【００４１】

第２の処理例では、ログ解析モジュール１７は以下のように動作する。単位ログには、ファイル操作を行ったプロセスのID（図２のID２０４）や名称（図２の名称２１４）のほか、そのプロセスを起動した親プロセスのID（図２のID２１０）や名称（図２の名称２１

２）が記録されている。ログ解析モジュール１７は、これらの情報に基づいて、任意の２つの単位ログに記録されたプロセスの親子関係を把握する。

【００４２】

具体的には、ログ解析モジュール１７は、一方の単位ログに含まれるプロセスのIDまたは名称が、他方の単位ログに含まれる親プロセスのIDまたは名称と一致する場合に、両者の単位ログを関連付ける。ただし、プロセスIDは、一時点においては、その時点で動作中の各プロセスに固有な情報であるものの、異なる時点において各プロセスに固有な情報であることを保証するものではないため、上記の処理にはプロセス名称を使用することがより望ましい。

【００４３】

ログ解析モジュール１７は、上記のようにして関連付けた２つの単位ログの親子関係を示す情報（以下、親子関係情報と記載する）を生成する。例えば、親プロセスのIDまたは名称と子プロセスのIDまたは名称とを関連付けた親子関係情報を生成する。続いて、ログ解析モジュール１７は不正プロファイル記憶部１６から不正プロファイルを読み出し、処理対象のログに含まれる単位ログに記録されているファイル名称と、不正プロファイルに含まれるファイル名称とを比較する。

【００４４】

不正プロファイルには複数のファイル名称が記録されている。単位ログに記録されているファイル名称が、不正プロファイルに記録されているいずれかのファイル名称と一致した場合、ログ解析モジュール１７は、不正が発生したと判定する。さらに、ログ解析モジュール１７は、不正が発生したと判定したときのファイル名称と関連付けられているボット種別情報を不正プロファイルから取得し、ボット種別情報に基づいてボットの種別を判定する。

【００４５】

また、単位ログに記録されているファイル名称が、不正プロファイルに記録されているいずれのファイル名称とも一致しなかった場合、ログ解析モジュール１７は、不正が発生していないと判定する。処理対象のログに複数の単位ログが存在する場合には、ログ解析モジュール１７は上記の処理を繰り返す。

【００４６】

ログ解析モジュール１７は、不正が発生したと判定したときに用いた単位ログから、ファイル操作を行ったプロセスのIDまたは名称を抽出し、そのIDまたは名称を含む親子関係情報に基づいて、親子のプロセスを含むグループが不正であると判定する。ログ解析モジュール１７は、不正であると判定したグループに関連する単位ログの情報を関連付けてログ記憶部１４に格納する。上記の処理では、親プロセスが不正ではないかつ子プロセスが不正であると判定される場合、親プロセスが不正であるかつ子プロセスが不正ではないと判定される場合、親プロセスが不正であるかつ子プロセスが不正であると判定される場合があるが、いずれの場合も、親子のプロセスを含むグループが不正であると判定される。

【００４７】

上記の処理によって、ボットを検知すると共にボットの種別を知ることができる。特に、ボットによって親子のプロセスが共同して所望の処理を行う場合に、不正と判定したグループに関連するログの情報から、親子のプロセスの挙動を知ることができる。なお、単位ログの処理において、単位ログに記録されているファイル名称とプロセス名称の組合せが、不正プロファイルに記録されているいずれかのファイル名称とプロセス名称の組合せと一致した場合に不正が発生したと判定するようにしてもよい。

【００４８】

（第３の処理例）

次に、第３の処理例を説明する。第３の処理例では、ボットによるプロセスがファイルにアクセスするときの手順と同一の手順でプロセスがファイルにアクセスした場合に、ボットによる不正が発生したと判定する。図６は、情報漏洩型のボットによるプロセスがファイルにアクセスするときの手順の例を示している。

【００４９】
図６では、プロセス６００がユーザフォルダのファイル６１０，６２０，６３０にアクセスする様子が示されている。矢印は、プロセス６００による各ファイルへのアクセスを示しており、矢印の近傍に記載された文字はアクセスの種類と順番を示している。アクセスの種類には、「r」（読み込み）、「w」（書き込み）、「d」（削除）がある。また、順番は数字で表され、数字が小さいほど順番が早い。例えば、「ｒ１」はファイルの読み込みであることと、順番が１番目であることとを示している。
【００５０】
図６に示したプロセス６００は以下のステップ１～ステップ３の手順で各ファイルにアクセスする。
ステップ１：ファイル６１０を読み込む。
ステップ２：ファイル６２０に書き込みを行う。
ステップ３：ファイル６３０に書き込みを行う。
【００５１】
図７は、トロイの木馬型のボットによるプロセスがファイルにアクセスするときの手順の例を示している。矢印の意味と、矢印の近傍に記載された文字の意味は図６と同様である。図７では、プロセス７００がシステムフォルダのファイル７１０，７２０，７３０にアクセスする様子が示されている。図７に示したプロセス７００は以下のステップ１～ステップ３の手順で各ファイルにアクセスする。
ステップ１：ファイル７１０を読み込む。
ステップ２：ファイル７２０に書き込みを行う。
ステップ３：ファイル７３０に書き込みを行う。
【００５２】
図８は、スパムメール型のボットによるプロセスがファイルにアクセスするときの手順の例を示している。矢印の意味と、矢印の近傍に記載された文字の意味は図６と同様である。図８では、プロセス８００がシステムフォルダのファイル８１０とユーザフォルダのファイル８２０，８３０にアクセスする様子が示されている。図８に示したプロセス８００は以下のステップ１～ステップ５の手順で各ファイルにアクセスする。
ステップ１：ファイル８１０を読み込む。
ステップ２：ファイル８２０に書き込みを行う。
ステップ３：ファイル８３０に書き込みを行う。
ステップ４：ファイル８２０を読み込む。
ステップ５：ファイル８３０を読み込む。
【００５３】
不正プロファイル記憶部１６に格納されている不正プロファイルには、ボットに感染した端末のプロセスがファイルにアクセスしたときの手順を示す情報が、ボット種別情報と関連付けられて記録されている。監視対象の端末で検知されたファイル操作の手順が、不正プロファイルに記録されている手順と同一である場合、監視対象の端末で検知されたファイル操作を行ったプロセスが不正であると判定されると共に、ボットの種別が判定される。また、監視対象の端末で検知されたファイル操作の手順が、不正プロファイルに記録されている手順と同一ではない場合、監視対象の端末で検知されたファイル操作を行ったプロセスは不正ではないと判定される。
【００５４】
第３の処理例では、ログ解析モジュール１７は以下のように動作する。まず、ログ解析モジュール１７は、処理対象のログに基づいて、監視対象の端末で検知されたファイル操作の手順を示す情報を生成する。具体的には、ログ解析モジュール１７は、同一のプロセスのID（図２のID２０４）や名称（図２の名称２１４）が記録されている単位ログを抽出し、時刻（図２の時刻２００）の順に単位ログを並べる。この結果、単位ログはファイル操作の順番に並ぶことになり、各単位ログの情報が、一連の手順を構成する各ステップの情報となる。

【００５５】
続いて、ログ解析モジュール１７は不正プロファイル記憶部１６から不正プロファイルを読み出し、監視対象の端末で検知されたファイル操作の手順を示す情報と、不正プロファイルに含まれる情報とを比較する。不正プロファイルには、１または複数のステップからなる手順の情報が記録されている。ログ解析モジュール１７は、ステップ毎に情報を比較する。まず、ログ解析モジュール１７は、監視対象の端末で検知されたファイル操作の最初のステップの情報と、不正プロファイルに記録されているファイル操作の最初のステップの情報とを比較する。
【００５６】
各ステップの情報には、少なくとも、ファイル操作を行ったプロセスの名称、ファイルの名称、ファイル操作の種別（読み込み／書き込み／削除）が含まれる。これらの情報が１つでも一致しなかった場合、監視対象の端末で検知されたファイル操作の手順が、不正プロファイルに記録されている手順と同一ではないため、ログ解析モジュール１７は、不正が発生していないと判定する。また、これらの情報が全て一致した場合、次のステップの情報が比較される。各ステップに関して、上記の処理が行われる。
【００５７】
全てのステップに関して、監視対象の端末で検知されたファイル操作の情報と、不正プロファイルに記録されているファイル操作の情報とが一致した場合、ログ解析モジュール１７は、不正が発生したと判定する。さらに、ログ解析モジュール１７は、不正が発生したと判定したときのファイル名称と関連付けられているボット種別情報を不正プロファイルから取得し、ボット種別情報に基づいてボットの種別を判定する。不正プロファイルには、複数のファイル操作について、各ファイル操作の手順を示す情報が記録されており、ログ解析モジュール１７は各ファイル操作について上記の処理を繰り返す。
【００５８】
上記の処理によって、ボットを検知すると共にボットの種別を知ることができる。
【００５９】
（第４の処理例）
次に、第４の処理例を説明する。第４の処理例では、親子のプロセスによる不正の有無を判定する。図９は、スパムメール型のボットによるプロセスがファイルにアクセスするときの手順の例を示している。
【００６０】
図９では、親プロセス９００と子プロセス９１０がシステムフォルダのファイル９２０，９３０とユーザフォルダのファイル９４０，９５０にアクセスする様子が示されている。矢印の意味と、矢印の近傍に記載された文字の意味は図６と同様である。図９に示した親プロセス９００と子プロセス９１０は以下のステップ１～ステップ６の手順で各ファイルにアクセスする。
ステップ１：親プロセス９００がファイル９２０を読み込む。
ステップ２：親プロセス９００がファイル９４０に書き込みを行う。
ステップ３：親プロセス９００がファイル９５０に書き込みを行う。
ステップ４：子プロセス９１０がファイル９３０を読み込む。
ステップ５：子プロセス９１０がファイル９４０を読み込む。
ステップ６：子プロセス９１０がファイル９５０を読み込む。
【００６１】
不正プロファイル記憶部１６に格納されている不正プロファイルには、ボットに感染した端末のプロセスがファイルにアクセスしたときの、親プロセスと子プロセスによる一連の手順を示す情報が記録されている。監視対象の端末で検知された、親プロセスと子プロセスによる一連のファイル操作の手順が、不正プロファイルに記録されている手順と同一である場合、監視対象の端末で検知されたファイル操作を行ったプロセスは不正であると判定される。また、監視対象の端末で検知されたファイル操作の手順が、不正プロファイルに記録されている手順と同一ではない場合、監視対象の端末で検知されたファイル操作

を行ったプロセスは不正ではないと判定される。

【００６２】

第４の処理例では、ログ解析モジュール１７は以下のように動作する。まず、ログ解析モジュール１７は、第２の処理例と同様の処理により、親子のプロセスの関係を把握し、親プロセスのIDまたは名称と子プロセスのIDまたは名称とを関連付けた親子関係情報を生成する。

【００６３】

続いて、ログ解析モジュール１７は、処理対象のログに基づいて、監視対象の端末で検知されたファイル操作の手順を示す情報を生成する。具体的には、ログ解析モジュール１７は、同一のプロセスのID（図２のID２０４）や名称（図２の名称２１４）が記録されている単位ログを抽出する。また、ログ解析モジュール１７は、親子関係情報に基づいて、このプロセスの親プロセスまたは子プロセスのIDまたは名称と同一のIDまたは名称を含む単位ログも抽出する。これによって、親子関係にあるプロセスの単位ログが抽出される。

【００６４】

続いて、ログ解析モジュール１７は、抽出した単位ログを時刻（図２の時刻２００）の順に並べる。この結果、単位ログはファイル操作の順番に並ぶことになり、各単位ログの情報が、一連の手順を構成する各ステップの情報となる。

【００６５】

続いて、ログ解析モジュール１７は不正プロファイル記憶部１６から不正プロファイルを読み出し、監視対象の端末で検知されたファイル操作の手順を示す情報と、不正プロファイルに含まれる情報とを比較する。不正プロファイルには、１または複数のステップからなる手順の情報が記録されている。ログ解析モジュール１７は、ステップ毎に情報を比較する。まず、ログ解析モジュール１７は、監視対象の端末で検知されたファイル操作の最初のステップの情報と、不正プロファイルに記録されているファイル操作の最初のステップの情報とを比較する。

【００６６】

各ステップの情報には、少なくとも、ファイル操作を行ったプロセスの名称、ファイルの名称、ファイル操作の種別（読み込み／書き込み／削除）が含まれる。これらの情報が１つでも一致しなかった場合、監視対象の端末で検知されたファイル操作の手順が、不正プロファイルに記録されている手順と同一ではないため、ログ解析モジュール１７は、不正が発生していないと判定する。また、これらの情報が全て一致した場合、次のステップの情報が比較される。各ステップに関して、上記の処理が行われる。

【００６７】

全てのステップに関して、監視対象の端末で検知されたファイル操作の情報と、不正プロファイルに記録されているファイル操作の情報とが一致した場合、ログ解析モジュール１７は、不正が発生したと判定する。さらに、ログ解析モジュール１７は、不正が発生したと判定したときのファイル名称と関連付けられているボット種別情報を不正プロファイルから取得し、ボット種別情報に基づいてボットの種別を判定する。不正プロファイルには、複数のファイル操作について、各ファイル操作の手順を示す情報が記録されており、ログ解析モジュール１７は各ファイル操作について上記の処理を繰り返す。

【００６８】

上記の処理によって、ボットを検知すると共にボットの種別を知ることができる。

【００６９】

上述した第１～第４の処理例において、処理結果を表示装置に表示してもよい。例えば、第１の処理例においては、不正と判定されたプロセスが操作を行ったファイルの情報とボットの種別を表示してもよい。第２の処理例においては、不正と判定されたプロセスが操作を行ったファイルの情報や、不正と判定されたプロセスを含む親子のプロセスグループの情報とボットの種別を表示してもよい。第３の処理例においては、不正と判定された手順とボットの種別を表示してもよい。第４の処理例においては、不正と判定された手順や、その手順による操作を行った親子のプロセスグループの情報とボットの種別を表示し

てもよい。

【００７０】

また、ホスト型侵入検知システムや不正プロセス検知システムで得られる情報を利用してもよい。ホスト型侵入検知システムは、監視対象の端末のファイルやディレクトリの正常な状態を保存して、定期的に整合性のチェックを行うことで、システムファイルの改ざんを検知するシステムである。不正プロセス検知システムは、マルウェアに感染した端末が、ユーザのキーボードやマウスの操作と関係なく、意図しないパケットを自動的もしくは外部からの制御によって送信する特徴に注目して、正常な端末の無操作状態の通信の特徴をプロファイル化して、これに該当しない通信を異常と判定して、不正プロセスを検知するシステムである。

【００７１】

ホスト型侵入検知システムでは、改ざんを検知したファイルの名称が得られる。第１～第４の処理例において、不正が発生したと判定された場合に、ログ解析モジュール１７は、その不正に関するファイルの名称と、ホスト型侵入検知システムが取得したファイルの名称とを比較する。両者が一致する場合、ボットによる不正が発生している可能性がより高いことを知ることができる。

【００７２】

不正プロセス検知システムでは、不正なプロセスの名称が得られる。第１～第４の処理例において、不正が発生したと判定された場合に、ログ解析モジュール１７は、その不正に関するプロセスの名称と、不正プロセス検知システムが取得したプロセスの名称とを比較する。両者が一致する場合、ボットによる不正が発生している可能性がより高いことを知ることができる。

【００７３】

また、第３～第４の処理例では、閾値θを設けて、監視対象の端末で検知されたファイル操作の手順を構成するステップと、不正プロファイルに記録されているファイル操作の手順を構成するステップとがθ個以上同一である場合に不正が発生したと判定するようにしてもよい。

【００７４】

また、以下のようにして、ボットによる不正を検知してもよい。ボットは、そのプロセスが自分自身のファイル（実行ファイル）を複製（自己複製）するという特徴を有する。特に、自己複製の際には、OSに関連するファイルが格納されるシステムフォルダにファイルが複製されるという特徴がある。

【００７５】

ボットの自己複製では、ファイル操作（複製）を行ったプロセスの元となったボットのファイル（およびそのファイルが格納されたフォルダ）と、そのプロセスがファイル操作（複製）を行ったファイル（およびそのファイルが格納されたフォルダ）とが同一となる。そこで、ファイル操作を行ったプロセスの元となったボットのファイル、またはそのファイルが格納されたフォルダ（以下、操作元プロセスのファイルまたはフォルダとする）と、そのプロセスがファイル操作を行ったファイル、またはそのファイルが格納されたフォルダ（以下、操作対象のファイルまたはフォルダとする）との相対関係（より具体的には相対パス）を不正プロファイルとする。

【００７６】

監視対象の端末で検出された操作元プロセスのファイルまたはフォルダを基準とする操作対象のファイルまたはフォルダの相対パスが不正プロファイル中の相対パスと同一である場合（実際には、監視対象の端末で検出された操作元プロセスのファイルまたはフォルダが操作対象のファイルまたはフォルダと同一である場合）、不正が発生したと判定される。相対パスは、操作元プロセスのファイルまたはフォルダの絶対パスと操作対象のファイルまたはフォルダの絶対パスとを演算して求めればよい。

【００７７】

さらに、ボットの自己複製ではシステムフォルダにファイルが複製されることから、監

視対象の端末で上記により検出された相対パスと不正プロファイルの相対パスとが一致し、かつ、操作対象のフォルダがシステムフォルダであった場合に不正が発生したと判定してもよい。

【００７８】

上述したように、本実施形態によれば、ボットに感染した正常な端末で生成されたプロセスによるファイルまたはフォルダの操作と同一の操作が検知された場合に、その操作の元となったボットの種別をボット種別情報から識別することが可能となるので、ボットを検知すると共にボットの種別を知ることができる。また、ボットの種別、すなわちPC内でのボットの挙動を把握できるため、ボットが検知されたときの正しい対処を迅速に図ることができる。

【００７９】

以上、図面を参照して本発明の実施形態について詳述してきたが、具体的な構成は上記の実施形態に限られるものではなく、本発明の要旨を逸脱しない範囲の設計変更等も含まれる。例えば、上記の不正検知装置の動作および機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータに読み込ませ、実行させてもよい。

【００８０】

ここで、「コンピュータ」は、ＷＷＷシステムを利用している場合であれば、ホームページ提供環境（あるいは表示環境）も含むものとする。また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ＲＯＭ、ＣＤ－ＲＯＭ等の可搬媒体、コンピュータに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムが送信された場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリ（ＲＡＭ）のように、一定時間プログラムを保持しているものも含むものとする。

【００８１】

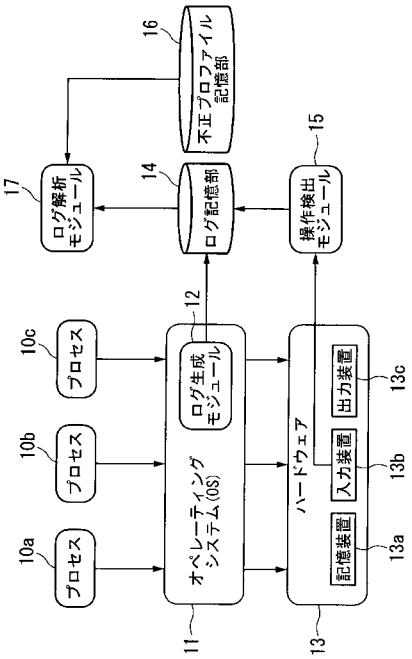
また、上述したプログラムは、このプログラムを記憶装置等に格納したコンピュータから、伝送媒体を介して、あるいは伝送媒体中の伝送波により他のコンピュータに伝送されてもよい。ここで、プログラムを伝送する「伝送媒体」は、インターネット等のネットワーク（通信網）や電話回線等の通信回線（通信線）のように、情報を伝送する機能を有する媒体のことをいう。また、上述したプログラムは、前述した機能の一部を実現するためのものであってもよい。さらに、前述した機能を、コンピュータに既に記録されているプログラムとの組合せで実現できるもの、いわゆる差分ファイル（差分プログラム）であってもよい。

【符号の説明】

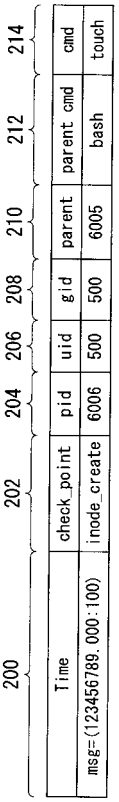
【００８２】

１１・・・OS、１２・・・ログ生成モジュール（情報生成手段、第１の情報生成手段）、１３・・・ハードウェア、１３ａ・・・記憶装置、１３ｂ・・・入力装置、１３ｃ・・・出力装置、１４・・・ログ記憶部、１５・・・操作検出モジュール、１６・・・不正プロファイル記憶部（記憶手段）、１７・・・ログ解析モジュール（第２の情報生成手段、情報取得手段）

【図１】



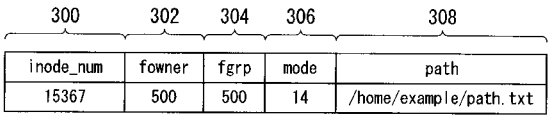
【図２】



200 202 204 206 208 210 212 214

Time	check_point	pid	uid	gid	parent	parent cmd	cmd
msg=(123456789.000:100)	inode_create	6006	500	500	6005	bash	touch

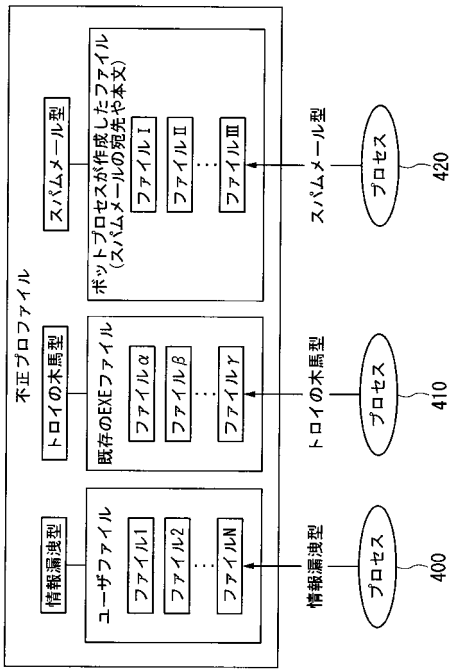
【図３】



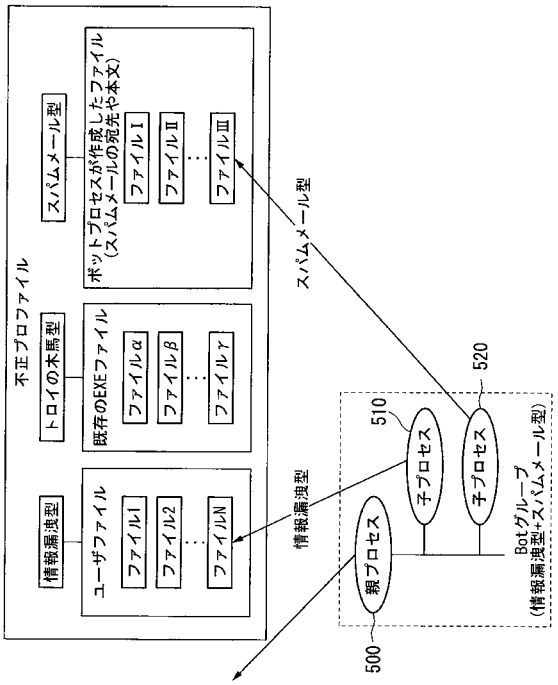
300 302 304 306 308

inode_num	fowner	fgrp	mode	path
15367	500	500	14	/home/example/path.txt

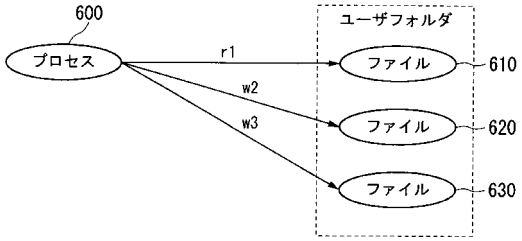
【図４】



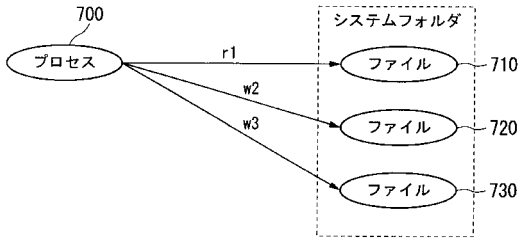
【図５】


不正プロファイル
スパムメール型
ファイルI
ファイルII
ファイルIII
ボットプロセスが作成したファイル
(スパムメールの宛先や本文)
トロイの木馬型
既存のEXEファイル
ファイルα
ファイルβ
ファイルγ
情報漏洩型
ユーザファイル
ファイル1
ファイル2
ファイルN
スパムメール型
情報漏洩型
520
510
子プロセス
子プロセス
親プロセス
500
Botグループ
(情報漏洩型+スパムメール型)

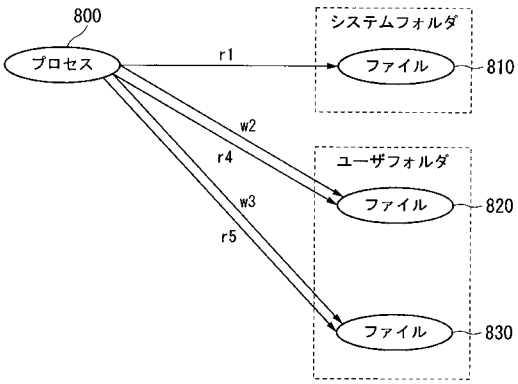
【図６】


600
プロセス
r1
w2
w3
ユーザフォルダ
ファイル
610
ファイル
620
ファイル
630

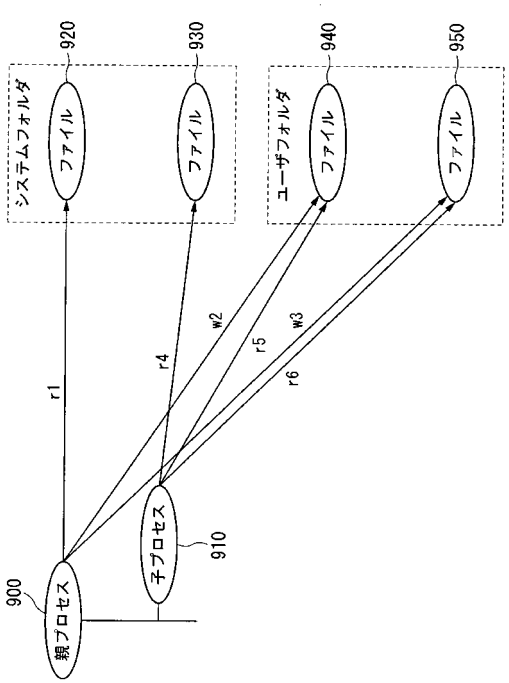
【図７】


700
プロセス
r1
w2
w3
システムフォルダ
ファイル
710
ファイル
720
ファイル
730

【図８】


800
プロセス
r1
w2
r4
w3
r5
システムフォルダ
ファイル
810
ユーザフォルダ
ファイル
820
ファイル
830

【図９】


920
930
940
950
システムフォルダ
ファイル
ファイル
ユーザフォルダ
ファイル
ファイル
r1
r4
w2
r5
w3
r6
子プロセス
910
900
親プロセス

フロントページの続き

(72)発明者　竹森　敬祐
埼玉県ふじみ野市大原２丁目１番１５号　株式会社ＫＤＤＩ研究所内
(72)発明者　磯原　隆将
埼玉県ふじみ野市大原２丁目１番１５号　株式会社ＫＤＤＩ研究所内
(72)発明者　三宅　優
埼玉県ふじみ野市大原２丁目１番１５号　株式会社ＫＤＤＩ研究所内
(72)発明者　酒井　崇裕
静岡県浜松市中区城北３丁目５－１　国立大学法人静岡大学情報学部内
(72)発明者　長谷　巧
静岡県浜松市中区城北３丁目５－１　国立大学法人静岡大学大学院情報学研究科内
(72)発明者　西垣　正勝
静岡県浜松市中区城北３丁目５－１　国立大学法人静岡大学創造科学技術大学院内

Ｆターム(参考) 5B276 FD08
5B285 AA06　BA07　CA32　CA37