(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5038888号
(P5038888)
(45)発行日　平成24年10月3日(2012.10.3)　(24)登録日　平成24年7月13日(2012.7.13)

(51)Int.Cl.　　F I
G06F 21/00 (2006.01)　　G06F 21/00　155C
G06F 13/00 (2006.01)　　G06F 13/00　351Z
H04L 12/56 (2006.01)　　H04L 12/56　400Z

請求項の数 30　(全 16 頁)

(21)出願番号　特願2007-511653 (P2007-511653)
(86)(22)出願日　平成17年5月4日(2005.5.4)
(65)公表番号　特表2007-536646 (P2007-536646A)
(43)公表日　平成19年12月13日(2007.12.13)
(86)国際出願番号　PCT/US2005/015933
(87)国際公開番号　WO2005/107424
(87)国際公開日　平成17年11月17日(2005.11.17)
審査請求日　平成20年5月2日(2008.5.2)
(31)優先権主張番号　10/839,613
(32)優先日　平成16年5月4日(2004.5.4)
(33)優先権主張国　米国(US)

(73)特許権者　506363997
アークサイト，インク.
アメリカ合衆国　カリフォルニア　９５０１４，クパチーノ，リザルツ　ウェイ　５
(74)代理人　100077539
弁理士　飯塚　義仁
(72)発明者　サウラブ，クマール
アメリカ合衆国　カリフォルニア　９５０５１，サンタ　クララ，クラマス　アベニュー　２０１６，　アパートメント　１
(72)発明者　ティドウェル，ケニー
アメリカ合衆国　カリフォルニア　９４９２４，ロス　アルトス，エス．スプリンガー　ロード　９１４

最終頁に続く

(54)【発明の名称】ネットワーク・セキュリティ・システムにおけるパターン発見方法及びシステム

(57)【特許請求の範囲】
【請求項１】
コンピュータシステム及びそれに関連するネットワークを介して授受されるイベント・ストリーム内の或るイベント・パターンを発見するためにコンピュータによって実行される方法であって、前記イベント・ストリームは複数のセキュリティ・イベントを含んでおり、
前記コンピュータシステムが、トランザクション・パラメータに基づき前記イベント・ストリームをトランザクション・フィルタ処理することにより複数のトランザクションを作成することと、ここで、各トランザクションは前記複数のセキュリティ・イベントの部分集合を表しており、
前記コンピュータシステムが、前記作成された前記複数のトランザクションをツリー・ビルダで処理することによりトランザクション・ツリーを生成することと、ここで、前記トランザクション・ツリーは１つのルート・ノードと複数の非ルート・ノードとを含み、各非ルート・ノードはそれぞれセキュリティ・イベントを表しており、
前記コンピュータシステムが、ツリー解析処理を行うことにより前記トランザクション・ツリーから前記イベント・パターンを抽出することと、
を具備し、前記抽出することは、第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することを含み、ここで、１つの非ルート・ノードのサポートとは該非ルート・ノードで表される前記セキュリティ・イベントを含むトランザクションの数を表している、

ことを特徴とする方法。
【請求項２】
前記トランザクション・パラメータは時間を含み、各トランザクションは前記時間に受信されたイベントを表す、請求項１に記載の方法。
【請求項３】
前記トランザクション・パラメータはイベント・フィールドの値を含み、各トランザクションは前記イベント・フィールドの前記値を含むイベントを表す、請求項１又は２に記載の方法。
【請求項４】
前記イベント・フィールドは送信元アドレスを含む、請求項３に記載の方法。
【請求項５】
前記イベント・フィールドは送信先アドレスを含む、請求項３に記載の方法。
【請求項６】
前記トランザクション・パラメータは一つの時点を含み、各トランザクションは前記一つの時点の近傍で受信されたイベントを表す、請求項１乃至５のいずれかに記載の方法。
【請求項７】
前記複数のトランザクションに基づき前記トランザクション・ツリーを生成することは、前記複数のトランザクション内のセキュリティ・イベントのサポートに基づき前記トランザクション・ツリーを生成することを含む、請求項１乃至６のいずれかに記載の方法。
【請求項８】
前記イベント・パターンをイベント相関則に変換するオプションをユーザに提供すること、を更に具備する請求項１乃至７のいずれかに記載の方法。
【請求項９】
前記イベント・パターンを用いてイベント相関則を作成すること、を更に具備する請求項１乃至７のいずれかに記載の方法。
【請求項１０】
前記複数のイベントはイベント・データベースに記憶されている、請求項１乃至９のいずれかに記載の方法。
【請求項１１】
前記複数のセキュリティ・イベントの各イベントからフィールドを除外すること、を更に具備する請求項１乃至１０のいずれかに記載の方法。
【請求項１２】
前記フィールドの値は各イベントに固有である、請求項１１に記載の方法。
【請求項１３】
前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することは、
前記ルート・ノードで始まる前記トランザクション・ツリーの分岐をトラバースすることと、ここで、該分岐は該ルート・ノードから前記第１の非ルート・ノードを通って前記第２の非ルート・ノードまで延びており、
前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することと、
前記イベント・パターンが、前記第１の非ルート・ノードによって表されたセキュリティ・イベントを含み、かつ、前記第２の非ルート・ノードによって表されたセキュリティ・イベントを含まないということを判定することと
を含む、請求項１乃至１２のいずれかに記載の方法。
【請求項１４】
規則を生成するためにコンピュータによって実行される方法であって、
ユーザインターフェースを介して、予め記憶された複数のセキュリティ・イベントの部分集合をユーザが選択できるように構成されたパターン発見ツールを表示することと、
前記ユーザインターフェースを介した前記ユーザの選択に応じて、

コンピュータシステムが、トランザクション・パラメータに基づきイベント・ストリームをトランザクション・フィルタ処理することにより複数のトランザクションを作成し、ここで、各トランザクションは選択されたセキュリティ・イベントの部分集合を表しており、

前記コンピュータシステムが、前記作成された前記複数のトランザクションをツリー・ビルダで処理することによりトランザクション・ツリーを生成し、ここで、前記トランザクション・ツリーは１つのルート・ノードと複数の非ルート・ノードとを含み、各非ルート・ノードはそれぞれセキュリティ・イベントを表し、および

前記コンピュータシステムが、ツリー解析処理を行うことにより前記トランザクション・ツリーからイベント・パターンを抽出し、該イベント・パターンを抽出することは第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することを含み、ここで、１つの非ルート・ノードのサポートとは該非ルート・ノードで表される前記セキュリティ・イベントを含むトランザクションの数を表していることと、

前記ユーザインターフェースを介して、ユーザが操作できるように構成された規則生成ツールを表示することと、

前記ユーザの操作に応じて、前記コンピュータシステムが、選択された新たな特定パターンを相関則に変換する処理を行うことと、

を具備する方法。

【請求項１５】

前記ユーザの操作はシングルマウスクリック操作を含む、請求項１４に記載の方法。

【請求項１６】

前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することは、

前記ルート・ノードで始まる前記トランザクション・ツリーの分岐をトラバースすることと、ここで、該分岐は該ルート・ノードから前記第１の非ルート・ノードを通って前記第２の非ルート・ノードまで延びており、

前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することと、

前記イベント・パターンが、前記第１の非ルート・ノードによって表されたセキュリティ・イベントを含み、かつ、前記第２の非ルート・ノードによって表されたセキュリティ・イベントを含まないということを判定することと

を含む、請求項１４又は１５に記載の方法。

【請求項１７】

コンピュータシステム及びそれに関連するネットワークを介して授受されるイベント・ストリームのイベント・パターンを発見するためのシステムであって、前記イベント・ストリームは複数のセキュリティ・イベントを含んでおり、

トランザクション・パラメータに基づき複数のトランザクションを作成するように構成されたトランザクション・フィルタと、各トランザクションは前記複数のセキュリティ・イベントの部分集合を表しており、

前記複数のトランザクションに基づきトランザクション・ツリーを生成するように構成されたツリー・ビルダと、前記トランザクション・ツリーは１つのルート・ノードと複数の非ルート・ノードとを含み、各非ルート・ノードはそれぞれセキュリティ・イベントを表しており、

前記トランザクション・ツリーからイベント・パターンを抽出するように構成されたツリー解析部と、

を具備し、前記ツリー解析部は、第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することを含み、ここで、１つの非ルート・ノードのサポートとは該非ルート・ノードで表される前記セキュリティ・イベントを含むトランザクションの数を表している、

ことを特徴とするシステム。
【請求項１８】
前記トランザクション・パラメータは時間を含み、各トランザクションは前記時間に受信されたイベントを表す、請求項１７に記載のシステム。
【請求項１９】
前記トランザクション・パラメータはイベント・フィールドの値を含み、各トランザクションは前記イベント・フィールドの前記値を含むイベントを表す、請求項１７又は１８に記載のシステム。
【請求項２０】
前記トランザクション・パラメータは一つの時点を含み、各トランザクションは前記一つの時点の近傍で受信されたイベントを表す、請求項１７乃至１９のいずれかに記載のシステム。
【請求項２１】
前記複数のトランザクションに基づきトランザクション・ツリーを生成することは、前記複数のトランザクション内のセキュリティ・イベントのサポートに基づき前記トランザクション・ツリーを生成することを含む、請求項１７乃至２０のいずれかに記載のシステム。
【請求項２２】
前記イベント・パターンをイベント相関則に変換するツールをユーザに提供するユーザ・インタフェース、を更に具備する請求項１７乃至２１のいずれかに記載のシステム。
【請求項２３】
前記複数のセキュリティ・イベントの各イベントからフィールドを除外するように構成されたイベント・フィールド・フィルタ、を更に具備する請求項１７乃至２２のいずれかに記載のシステム。
【請求項２４】
前記ツリー解析部は、更に、
前記ルート・ノードで始まる前記トランザクション・ツリーの分岐をトラバースし、ここで、該分岐は該ルート・ノードから前記第１の非ルート・ノードを通って前記第２の非ルート・ノードまで延びており、
前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察し、
前記イベント・パターンが、前記第１の非ルート・ノードによって表されたセキュリティ・イベントを含み、かつ、前記第２の非ルート・ノードによって表されたセキュリティ・イベントを含まないということを判定する
ように構成されていることを特徴とする、請求項１７乃至２３のいずれかに記載のシステム。
【請求項２５】
コンピュータシステム及びそれに関連するネットワークを介して授受されるイベント・ストリーム内の或るイベント・パターンを発見するための手順をプロセッサに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、前記イベント・ストリームは複数のセキュリティ・イベントを含んでおり、
トランザクション・パラメータに基づき前記イベント・ストリームをトランザクション・フィルタ処理することにより複数のトランザクションを作成する手順と、ここで、各トランザクションは複数のセキュリティ・イベントの部分集合を表しており、
前記作成された前記複数のトランザクションをツリー・ビルダで処理することによりトランザクション・ツリーを生成する手順と、ここで、前記トランザクション・ツリーは１つのルート・ノードと複数の非ルート・ノードとを含み、各非ルート・ノードはそれぞれセキュリティ・イベントを表しており、
ツリー解析処理を行うことにより前記トランザクション・ツリーからイベント・パターンを抽出する手順であって、該抽出する手順は、第１の非ルート・ノードから第２の非ル

ート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することを含み、ここで、１つの非ルート・ノードのサポートとは該非ルート・ノードで表される前記セキュリティ・イベントを含むトランザクションの数を表している前記手順と、
を前記プロセッサに実行させるための前記プログラムを記憶したコンピュータ読み取り可能な記憶媒体。
【請求項２６】
前記複数のトランザクションに基づき前記トランザクション・ツリーを生成することは、前記複数のトランザクション内のセキュリティ・イベントのサポートに基づき前記トランザクション・ツリーを生成することを含む、請求項２５に記載の記憶媒体。
【請求項２７】
前記プログラムは、前記プロセッサに、前記イベント・パターンをイベント相関則に変換するオプションをユーザに提供する手順を更に実行させることを特徴とする、請求項２５又は２６に記載の記憶媒体。
【請求項２８】
前記プログラムは、前記プロセッサに、前記イベント・パターンを用いてイベント相関則を作成する手順を更に実行させることを特徴とする、請求項２５又は２６に記載の記憶媒体。
【請求項２９】
前記プログラムは、前記プロセッサに、前記複数のセキュリティ・イベントの各イベントからフィールドを除外する手順を更に実行させることを特徴とする、請求項２５乃至２８に記載の記憶媒体。
【請求項３０】
前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することにより前記トランザクション・ツリーを解析することは、
前記ルート・ノードで始まる前記トランザクション・ツリーの分岐をトラバースすることと、ここで、該分岐は該ルート・ノードから前記第１の非ルート・ノードを通って前記第２の非ルート・ノードまで延びており、
前記第１の非ルート・ノードから第２の非ルート・ノードまでのサポートの落ち込みを観察することと、
前記イベント・パターンが、前記第１の非ルート・ノードによって表されたセキュリティ・イベントを含み、かつ、前記第２の非ルート・ノードによって表されたセキュリティ・イベントを含まないということを判定することと
を含む、請求項２５乃至２９に記載の記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、異種及び同種のソースからセキュリティ・イベントを取り込み、取り込んだセキュリティ・イベントにおけるパターンを発見するコンピュータベースのシステムに関する。
【背景技術】
【０００２】
コンピュータ・ネットワークやシステムは今日のビジネスに不可欠なツールとなっている。今日、想像し得る殆ど全ての話題に関する無数の情報が世界中のユーザによって上記ネットワークに記憶され、上記ネットワークを通じてアクセスされる。該情報の多くは程度の差こそあれ機密情報であり、保護することが必要である。よって、アクセス権を持たない人々及び／あるいは装置がコンピュータ・ネットワーク及びその内部に記憶された情報にアクセスしようとする試みを発見する際に利用される侵入検出システム（ＩＤＳ）が開発されてきたことは当然のことである。更に、ルータやファイアウォール等のネットワーク装置は、上記のような試みを調べる際に利用できる動作履歴の記録をとっておく。

【０００３】
侵入検出とは、コンピュータ・ネットワークやシステム内の、あるいはそれに関する不適切、不正確、或いは異常な動作を検出する技術と考えることができる。侵入検知の最も一般的な方法は、統計的異常検出及びパターン照合検出である。ホスト上で動作して該ホスト上の不正な動作を検出するＩＤＳは、ホストベースＩＤＳ（ＨＩＤＳ）と呼ばれ、ホスト・ラッパー／パーソナル・ファイアウォールあるいはエージェントベースのソフトウェア形式で存在し得るものであり、ネットワーク・データフロー上で動作するものはネットワークベースＩＤＳ（ＮＩＤＳ）と呼ばれる。ホストベースの侵入検出では、監視対象のシステム（ホストコンピュータ）上のソフトウェアを読み込み、ログファイルあるいはホストの監査エージェントをデータソースとして用いる。これに対し、ネットワークベースの侵入検出システムは、そのネットワークセグメント上におけるトラフィックを監視し、該トラフィックをデータソースとして用いる。ネットワーク・インターフェースにより取り込まれたパケットは、シグニチャに一致した場合に目的のものであると考えられる。
【０００４】
データソースに関わらず、侵入を検出するには補完的な２つの方法がある。すなわち、知識ベースの方法と行動ベースの方法である。今日利用さられている殆ど全てのＩＤＳツールは知識ベースのものである。知識ベースの侵入検出技術では、取り込んだデータを、脆弱性につけ込む既知の技術に関連する情報と比較する。一致が検出されると警告が発せられる。一方、行動ベースの侵入検出技術では、システムやユーザの正常な行動あるいは予測される行動（そのモデルは、各種手段により収集された参照用情報から抽出される）からの逸脱を確認することにより侵入を発見する。疑わしい逸脱が確認されると警告が発せられる。
【０００５】
知識ベースの方法の利点は、誤った警告を発する率が非常に低いことであり、侵入検出システムが提案する文脈分析は詳細にわたっているため、そのような侵入検出システムを利用するセキュリティ担当者は予防的方法あるいは対処法をとることが容易になる。欠点は、既知の攻撃に関連する必要な情報を収集したり、新しい脆弱点や環境についていくのが難しいことである。
【０００６】
行動ベースの方法の利点は、予想外の新たな税弱点につけ込む試みを検出できることである。また、システム仕様に対する依存度が少ない。しかしながら、一般に、この方法の重大な欠点として誤った警告が発せられる率が高いことが挙げられ、行動は時とともに変化し得るため、このような誤った警告が発せられる頻度が上昇してしまう可能性がある。
【０００７】
ホストベース及びネットワークベースのどちらが採用されるかに関わらず、また知識ベースか行動ベースかに関わらず、侵入検出システムは、システムの正常な使用と実際の侵入とを区別する（適切な警告を伴う）機能を持たなければ役に立たない。迅速に侵入を検出し、適当な担当者に知らせることができれば、保護されているシステムが危険に晒されることを回避する措置を講じることができる。そうでなければ、そのような予防措置を講じることはできない。したがって、コンピュータ・ネットワークやシステムを脅かす試みを効果的に抑制するために正確及び適時に侵入を検出し、警告を発することができるシステムが求められている。
【発明の開示】
【０００８】
ネットワーク・セキュリティ・システムによって収集されたセキュリティ・イベントにおいてパターンを発見することができる。一実施例において、本発明は各種監視装置からのセキュリティ・イベントを収集し、記憶する。一実施例において、記憶されたセキュリティ・イベントのサブセットはイベント・ストリームとしてマネージャに供給される。一実施例において、本発明は、更に、イベント・ストリームにおける１以上の未知のイベント・パターンをマネージャによって発見する。

【０００９】
以下の添付図面において限定ではなく例示のために本発明の実施例を示す。
【００１０】
【００１１】
【００１２】
【００１３】
【発明を実施するための最良の形態】
【００１４】
以下、本発明について様々な実施例を参照して説明するが、これらの実施例をより広範である本発明の精神と範囲を限定するものとして解釈すべきではない。例えば、ここに挙げる例は、分散したエージェント、マネージャ、及びコンソールについて説明するが、これは本発明の一実施例にしか過ぎない。本発明の一般的概念及び範囲ははるかに広いものであり、いかなるコンピュータベースあるいはネットワークベースのセキュリティ・システムにも及ぶ。また、本発明を更に詳細に説明するために、システムの構成要素に渡す又はそこから渡されるメッセージと、システムの構成要素が使用できるデータスキーマの例を示すが、それらは包括的なものではないし、またそのように考えるべきものではない。
【００１５】
以下の詳細な説明の一部は、コンピュータメモリ中のデータに対する処理のアルゴリズム及び記号的表現によって示される。このようなアルゴリズムによる説明や表現は、コンピュータサイエンス技術分野における他の当業者に処理の内容を最も効果的に伝えるために当業者により用いられる手段である。ここで、及び一般的に、アルゴリズムはある所望の結果に至る首尾一貫した一連のステップだと考えられる。これらのステップは物理量の物理的処理を必要とする。必ずしもそうであるとは限らないが、通常、これらの物理量は、記憶、転送、合成、比較、または処理可能な電気または磁気信号の形をとる。主に一般的な慣習から、そのような信号をビット、値、要素、記号、文字、用語、数値等と称するのが時には便利であることが分かっている。しかしながら、これらの用語及び類似の用語は全て適当な物理量に関連付けられるものであり、該物理量に適用された便宜的な標識でしかないことを念頭に置くべきである。特に指定しない限り、詳細な説明全体にわたり、「処理」、「演算」、「計算」、「判定」、「表示」等の用語を用いた説明は、コンピュータ・システム及びメモリ内で物理（電子）量として表されたデータを同様にコンピュータ・システムのメモリやレジスタまたは伝送装置や表示装置等の他の情報記憶装置内で物理量として表された他のデータに変換するコンピュータ・システムや類似の電子計算機の動作及び処理を示すものである。
【００１６】
上記のように、本発明の一実施例は、コンピュータソフトウェア、即ち、１以上のコンピュータプロセッサ／システムにより実行されると該コンピュータプロセッサ／システムに指定された動作を行わせるコンピュータ可読の指示でインスタンス化される。このようなコンピュータソフトウェアは、１以上のコンピュータ可読媒体、例えばハードディスク、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、読み出し専用記憶装置、読取書き込み記憶装置等に常駐させることができる。このようなソフトウェアは、上記媒体のうち1以上の媒体上に分散することができ、あるいは、１以上のコンピュータ・ネットワーク（例えば、インターネット）を介したダウンロードに利用することができる。フォーマットに関わらず、ここで説明するコンピュータ・プログラミング、レンダリング、及び処理技術は、本発明の側面を実施するために利用することができるコンピュータ・プログラミング、レンダリング、及び処理技術の種類の一例に過ぎない。これらの例は本発明を限定するものではなく、本発明は本説明に続く請求の範囲を参照することにより最もよく理解することができる。
【００１７】
ところで、図１には、本発明の実施例に従って構築されたコンピュータベースのネットワーク・セキュリティ・システム１０の一例が示されている。システム１０は、エージェント１２、１以上のマネージャ１４、及び１以上のコンソール１６（ブラウザベースのも

のを備えてもよい）を備える。いくつかの実施例では、エージェント、マネージャ及び／あるいはコンソールを単一のプラットフォームにまとめたり、（図示例のように）２つ、３つ、あるいはそれ以上のプラットフォームに分散したりできる。この多層構造を用いることにより、コンピュータ・ネットワークあるいはシステムが成長する際の拡張性をサポートする。

【００１８】

エージェント１２は、各種ネットワーク・セキュリティ装置及び／あるいはアプリケーションから効率的なリアルタイムの（あるいは略リアルタイムの）ローカルイベント・データ・キャプチャ及びフィルタリングを行う。セキュリティ・イベント（安全性チェックの対象となるイベントデータ）の主なソースは、ファイアウォール等の共通のネットワーク・セキュリティ装置、侵入検出ステム、及びオペレーティングシステムのログである。エージェント１２は、イベント・ログやメッセージを生成する何れかのソースからイベントを収集し、固有装置、ネットワーク内の結合ポイント、及び／あるいは単純なネットワーク管理プロトコル（ＳＮＭＰ）トラップで動作することができる。

【００１９】

エージェント１２は、手動及び自動処理により、関連する構成ファイルを介して構成可能である。各エージェント１２には、正規化コンポーネント、時間補正コンポーネント、集約コンポーネント、バッチ・コンポーネント、リゾルバ・コンポーネント、トランスポート・コンポーネント、及び／あるいはその他のコンポーネントを含む１以上のソフトウェア・モジュールを含めることができる。これらのコンポーネントは、構成ファイル中の適当なコマンドにより起動及び／あるいは停止することができる。

【００２０】

マネージャ１４は、更に規則エンジン１８及び集中イベント・データベース２０を用いて、エージェントから受け取ったイベントを統合及び選別し、相互に関連づけるサーバベースのコンポーネントでもよい。マネージャ１４の役割の一つは、全てのリアルタイム及び履歴イベントデータを取り込み、記憶して、（データベース・マネージャ２２を介し）セキュリティ活動の企業規模における全体像を構築する。また、マネージャ１４は、集中管理、通知、及び（１以上の通知部２４を介して）報告を行うとともに、知識ベース２８及びケース管理ワークフローを有する。マネージャ１４は任意のコンピュータ・ハードウェア・プラットフォーム上に配置することができ、一実施例では、オラクル（登録商標）・データベース等のリレーショナルデータベース管理システムを利用してイベントデータ記憶コンポーネントを実施する。マネージャ１４とエージェント１２との間における通信は双方向とすることができ（例えば、マネージャ１４がエージェント１２のホストとしてのプラットフォームにコマンドを送信させる）、暗号化できる。いくつかの装置では、マネージャ１４が複数のエージェント１２の集線装置としての役割を果たすことができ、（企業の本社に配置された）他のマネージャに情報を転送することができる。

【００２１】

コンソール１６は、モニタリング、規則オーサリング、事件調査、及び報告等の日常的な管理及びオペレーション業務をセキュリティ専門家が行えるようにするコンピュータ（例えば、ワークステーション）ベースのアプリケーションである。アクセス制御リストにより、複数のセキュリティ専門家が、各々の責務に応じて、同一のシステム及びイベント・データベースを各々の表示で使用したり、相関則、警告、報告、及び知識ベースを使用したりすることが可能になる。複数のコンソール１６を単一のマネージャ１４によりサポートすることができる。

【００２２】

いくつかの実施例では、ブラウザベースのコンソール１６を用いてセキュリティ・イベント、知識ベースのアーティクル、レポート、通知、及びケースに対するアクセスを行うことができる。即ち、マネージャ１４は、（コンソール１６の代わりとなる）パーソナル・コンピュータあるいは手持ち型コンピュータをホストとするＷｅｂブラウザを介してアクセス可能なＷｅｂサーバ・コンポーネントを備えることにより、コンソール１６の機能

のいくつか、あるいは全てを実現することができる。ブラウザを介するアクセスは、コンソール１６から遠く離れたセキュリティ専門家や、パートタイム・ユーザにとって特に有効である。コンソール１６とマネージャ１２との通信は双方向とすることができ、暗号化可能である。

【００２３】

上述の構造により、本発明は集中型あるいは分散型環境をサポートできる。これは、企業組織がシステム１０の単一のインスタンスを実施し、アクセス制御リストを用いてユーザを区分したい場合があるため有効である。一方、企業組織では、多数のグループのうち各グループに個々のシステム１０を配置し、結果を「マスター」レベルでまとめることを望む場合がある。このような配置により、地理的に分散したピア・グループが、標準的な業務時間中に現在業務を行っているグループに主要な監督責任を移すことにより互いに連携する「フォローザサン（ｆｏｌｌｏｗ－ｔｈｅ－ｓｕｎ）」という仕組みが可能になる。また、システム１０は、企業の事業部が個別に業務を行い中央管理機能のロールアップをサポートするという企業組織に置くこともできる。

【００２４】

図１に例示したネットワーク・セキュリティ・システムは、この参照により全体が開示に含まれる２００２年１２月２日に出願された「複数のネットワークセキュリティ装置からのイベントのリアルタイム・モニタリング及び分析」というタイトルの米国出願第１０／３０８，４１５号（米国特許第７３７６９６９号）において更に詳しく説明されている。

【００２５】

本発明の一実施例において、ネットワーク・セキュリティ・システム１０は、更にパターン認識機能を備える。一実施例では、図１に示すように、パターン認識機能がパターン発見モジュール３０によってマネージャ１４上で実現される。図１に示すように、一実施例において、イベント・データベース２０に記憶された一組のセキュリティ・イベントを、データベース・マネージャ２２を介してパターン発見のためにパターン発見モジュール３０に供給することができる。

【００２６】

一実施例において、ユーザはコンソール・インターフェース１６を用いて、再生する既に記憶されたセキュリティ・イベントの時間を指定することによりイベント・データベース２０からイベントの再生を開始することができる。イベントの再生に関するいくつかの実施例は、この参照により全体が開示に含まれる２００２年１２月２日に出願された「ネットワーク・セキュリティ・システムの相関関係を使用しデバッグする方法及び装置」というタイトルの米国出願第１０／３０８，４１６号で更に詳しく説明されている。

【００２７】

一実施例において、再生された上記イベント・ストリームはパターン発見モジュール３０に供給される。別の実施例では、エージェント１２から収集されたセキュリティ・イベントがパターン発見モジュール３０に直接供給され、パターン発見モジュール３０は収集されたイベントのスナップショットを時々撮る。一実施例において、パターン発見モジュール３０は供給されたイベント・ストリームに対する関心のパターンを見つけ、言い換えれば、発見、検出する。

【００２８】

一実施例において、パターン発見モジュール３０が検出したパターンと規則エンジン１８が特定したパターンとの相違は、規則エンジン１８は収集されたセキュリティ・イベントにおける既知のパターンを発見する一方、パターン発見モジュール３０は収集されたセキュリティ・イベントにおける未知のパターンを見つける点である。言い換えれば、規則エンジン１８はイベント・ストリームに適用される相関則を有し、各規則は特定のパターンを探す。しかしながら、一実施例において、パターン発見モジュール３０は、既定の規則やフィルタ等、暗号化された領域知識に依存せず、そのように認識されたことがないパターンを発見する。一実施例において、パターン発見モジュールによって発見されたパタ

ーンは、その後セキュリティ・イベントを相互に関連付けるために規則エンジン１８によって利用される相関則に変換することができる。

【００２９】

一実施例においては、また、規則エンジン１８とパターン発見モジュール３０の動作も、規則エンジンは探している既知のパターンの特定インスタンスを見つけるという点で異なる。しかしながら、パターン発見モジュール３０は予想されるパターンを特定するように構成されているのみである。イベント・ストリームにおけるパターンを見つけるために、新たに検出されたパターンに基づく規則を規則エンジン１８に供給することができる。

【００３０】

以下、パターン発見モジュール３０の一実施例について、図２を参照して更に詳しく説明する。上述のように、パターン発見モジュール３０には、一実施例においてデータベース・マネージャ２２によってイベント・データベース２０から供給されるイベント・ストリーム（セキュリティ・イベントの集まり）が入力される。一実施例において、パターン発見モジュール３０は、各イベントに対して選択されたイベント・フィールドのみを保持するイベント・フィールド・フィルタ３２を備える。例えば、各イベントに固有のフィールド（例えば、固有イベント識別子）等、パターン発見には役に立たないイベント・フィールドをイベント・フィールド・フィルタ３２によって除外し、メモリを節約し、処理速度を上げることができる。

【００３１】

イベント・フィールド・フィルタは、少なくとも、トランザクション・フィルタ３４によって利用されるフィールドを保持するように構成される。一実施例において、イベント・フィールド・フィルタ３２は少なくともイベント名フィールド及び送信元及び送信先フィールドを保持する。別の実施例では、センサ発売元、製品、カテゴリ、及び送信元及び送信先ポート等、更に他のイベント・フィールドが保持される。このような実施例では、各イベント・フィールド・フィルタ３２が、少なくとも各トランザクション・フィルタ３４によって利用されるフィールドを保持する。イベント・フィールド・フィルタ３２は、任意の各イベント・フィールドや、上記のイベント・フィールドを組み合わせたものを含むイベント・フィールドのサブセットを除外あるいは保持することができる。

【００３２】

一実施例において、（保持されているイベント・フィールドによって表されるような）セキュリティ・イベントを用いてトランザクション（transaction: 処理、取引）を生成する。一実施例において、トランザクションは、何らかのトランザクション・パラメータによって関連付けられるセキュリティ・イベントを表す要素のグループとして定義される。例えば、ある時間枠（例えば、３０分、１日、１ヶ月・・・）内に受け取った全イベントを１つのトランザクションとすることができる。一実施例では、何れかの時間枠における全イベントであるイベント・ストリームは、そのような時間ベースの任意の数のトランザクションに分けることができる。

【００３３】

また、トランザクションはイベント発生率によって定義することもできる。一実施例において、トランザクションは集中的に発生するセキュリティ・イベントにより定義される。言い換えれば、イベント発生率に基づき、大きいトランザクションを更に小さいトランザクションに分割することができる。隠れマルコフモデルのようなイベント・ストリームあるいは大きいトランザクションの統計的分析によりイベント・ストリームが集中的に発生したものであると示されている場合、集中的に発生した各イベント・ストリームはトランザクションに選別される。

【００３４】

また、トランザクションはイベント・フィールドにより定義することもできる。一実施例において、同じ送信元アドレス（例えば、送信元のＩＰアドレス）からのイベントがトランザクションとして定義される。したがって、そのような各トランザクションは、イベント・ストリームの時間内における同一のＩＰアドレスからの全イベントを含む。

【００３５】
トランザクション・パラメータは、送信元及び送信先アドレスの両方とすることできる。したがって、いくつかのトランザクションを、同一の送信元及び送信先ＩＰアドレス（あるいはＩＰポート、あるいはその両方）を有する、ある時間枠の全イベントとして定義することができる。更に、いくつかのトランザクションを、２つの関連する送信元／送信先アドレス（あるいはポート）間でやり取りされたイベントとして定義することができる。このような実施例では、アドレスＡからＢ及びアドレスＢからＡへのセキュリティ・イベントが１つのトランザクションに含まれる。
【００３６】
一実施例において、トランザクションは、実際のセキュリティ・イベントを含まないで、セキュリティ・イベントを表す要素を含む。１つの要素により多数の同一セキュリティ・イベントを表すことができる。これにより、多くのデータを保存することができる。一実施例において、各要素は、各セキュリティ・イベントが表されている回数を示す。別の実施例では、各セキュリティ・イベントは、単に、トランザクション中の個々の数えられない要素によって表される。
【００３７】
一実施例において、トランザクションの定義は互いに相反するものではない。即ち、上述したトランザクションの定義、以下に述べるトランザクションの定義及びここでは説明しないその他のトランザクションの定義を、同じ一組のセキュリティ・イベントに対して行うことができる。
【００３８】
一実施例において、トランザクションは複数のトランザクション・フィルタ３４によって定義される。図２の省略記号は、セキュリティ・イベントが多数のトランザクションに弁別（フィルタ）され得ることを示しており、図２に示す４つのトランザクション・フィルタ３４に限られるものではない。トランザクション・フィルタ３４は、上述したあらゆるトランザクションを生成するように構成することができる。例えば、トランザクション・フィルタ３４ａが時間ベースのトランザクションを生成する一方、トランザクション・フィルタ３４ｂはイベント・フィールド・ベースのトランザクションを生成することができる。
【００３９】
一実施例において、トランザクション・フィルタは、更に、セキュリティ・イベントを各要素に変換する。上述のように、一実施例では各要素はイベント種類の記述として定義される。トランザクション・フィルタ３４がトランザクションに応じている際、上述したように、重複したセキュリティ・イベントは破棄され、または、要素数が増加する。
【００４０】
一実施例において、トランザクションを「ひとまとめにする」ことによって新たなトランザクションを生成することができる。例えば、トランザクション・フィルタ３４は、２つの重複するトランザクションをひとまとめにするように構成することができる。よって、例えば１つのトランザクションが送信元Ａから送信先Ｂへの要素をまとめ、別のトランザクションが送信元Ｂから送信先Ｃへの要素をまとめる場合、第３のトランザクションにはＡからＢ、ＢからＣへと渡される項目を含めることができる。その他のトランザクションも一部重複する場合がある。このようにひとまとめにされたトランザクションは、ワームの拡散というトランザクションや、送信先アドレスが攻撃に使う送信元アドレスになるような他の攻撃を検出する上で有効である。
【００４１】
一実施例において、トランザクション・フィルタ３４により生成されたトランザクションはパターン識別部３６に供給される。パターン識別部３６はトランザクションを解析し、イベント・ストリーム中のパターンを発見する。以下、パターン識別部３６の一実施例について図３を参照して更に詳しく説明する。
【００４２】

一実施例において、一緒に発生する、即ち、何らかの方法で関連付けられるセキュリティ・イベント群として、パターンを定義することができる。例えば、セキュリティ・イベントＡ、Ｂ及びＣが頻繁に略同時に発生する場合、イベントグループＡＢＣを１つのパターンとすることができる。一方、Ａ、Ｂ及びＣが略同時に起こらずに、時間の経過とともに同様の頻度で発生する場合にも、イベントグループＡＢＣを１つのパターンとすることができる。既知のパターンの例としては、一連のログイン失敗及びその後に続く正常なログインが挙げられる。このようなパターンは辞書攻撃が行われる際に生じる。緩慢で低調な攻撃は、発生時間は近くないがイベント発生頻度で関連付けられるイベントから成る場合がある。

【００４３】

パターン発見モジュール３０に供給されたセキュリティ・イベントにおけるイベント・パターンを特定するために、一実施例では、パターン識別部３６はまずツリー・ビルダ３８を用いて、トランザクション・フィルタ３４によって供給されたトランザクションからトランザクション・ツリーを構築する。一実施例において、ツリー・ビルダ３８は、まず要素サポートにより各トランザクションの要素を並べ替える。

【００４４】

一実施例において、これは、各要素が発生するトランザクションの数をカウントすることにより行われる。この数は、要素の「サポート」と呼ばれることがある。一実施例において、要素は、トランザクション内に留まるために最小限のサポートを必要とし、さもなければトランザクションのサイズを小さくするために削除される。

【００４５】

一実施例において、ツリー・ビルダ３８は、次に、各セキュリティ・イベントのサポートに基づき、トランザクションをツリー構造として構成する。以下、トランザクション・ツリーを構築ための方法の一つについて、図４を参照して説明する。図４に示す方法では、一度に１つのトランザクションを繰り返し処理する。一実施例において、トランザクションは、サポートの程度が高い要素から低い要素の順に整理される。

【００４６】

ブロック４０２でトランザクションが選択される。一実施例において、ツリーは空のルート・ノードを有する。ブロック４０４では、該ルート・ノードが選択され、ブロック４０６では、選択されたトランザクションにおける次の要素が選択されるが、この要素は、最初の繰り返しにおける第一番目（すなわち、サポートの程度が最も高い）の要素である。

【００４７】

ブロック４０８では、選択されたノード（最初の繰り返しにおけるルート・ノード）の子が、選択された要素と同じ要素を示すか否かについて判定する。答えがＹＥＳである場合、ブロック４１０でこのノードが選択ノードとなり、選択された要素のサポートが現在のノードのサポートに付加される。一方、答えがＮＯである場合にはブロック４１４に進み、選択された要素を表すために、選択ノードの新たな子ノードを作成し、ブロック４１６でこの新たなノードが選択ノードとなる。

【００４８】

次に、ブロック４１２で、選択されたトランザクションの処理が完了したかどうか、すなわち更に他の要素があるか否か判定する。トランザクションが完了していない場合（トランザクションに他の要素が残っている場合）はブロック４０６に進み、選択されたトランザクションの処理が継続され、次の要素がサポート順に選択される。一方、トランザクションが完了している場合（トランザクションに更に他の要素がない場合）、ブロック４０２で次に処理されるトランザクションが選択され、ツリー構築処理で全トランザクションが処理されるまで、図４のフローチャートに従って処理が続行される。

【００４９】

再び図３を参照すると、一実施例において、ツリー・ビルダ３８はトランザクション・ツリーをツリー解析部４０に供給し、ツリー解析部４０はトランザクション・ツリーから

対象パターンを抽出する。一実施例において、ツリー解析部４０は、ツリーの分岐を横断（トラバースすなわち探査）してサポートの大幅な落ち込みを探す。例えば、ツリーの１つの分岐～要素名（サポート）として示される～がＡ（２００）、Ｂ（１９０）、Ｃ（５０）、Ｄ（４５）である場合、要素Ｂと要素Ｃとの間にはサポートの大幅な落ち込みがある。この場合、ツリー解析部４０は、グループＡＢを１つのパターンとして特定するとともに、グループ　ＣＤを１つのパターンとして特定する。また、ツリー解析部は、グループＡＢＣＤを１つのパターンとして特定することもできる。一実施例において、１つのパターンとしてのＡＢＣＤは、該パターンにおいてこれらのセキュリティ・イベントの発生順序が保たれることを意味するものではない場合がある。

【００５０】

一実施例において、サポートの落ち込みは減少のパーセント値によって定義される。例えば、ツリー解析部４０は、１つの分岐中の１要素から次の要素までサポートが３０パーセント以上落ち込む場合には、常に、サポートの落ち込みであると定義する。他のパーセント値を用いることもできる。別の実施例においては、例えば１０の数値分のサポートの落ち込みにより、サポートの落ち込みが定義される。

【００５１】

更に別の実施例において、ツリーの分岐において次の要素が追加されることにより、それまでのサポートの標準偏差が閾値を越えてしまうと考えられる場合、サポートの落ち込みが検出される。例えば、最初の３つの分岐要素のサポート標準偏差が０．０３であるが第４の要素の追加により標準偏差が０．０３を超えてしまうような場合には、ツリー解析部４０は分岐上の第３と第４の要素との間にサポートの落ち込みを確認する。０．０３という数値は単なる一例であり、他の閾値を用いてもよい。更に、標準偏差の代わりに、分散量等、その他の変化の統計的尺度を用いることもできる。

【００５２】

一実施例において、ツリー解析部４０は、トランザクション・ツリーを横断しながらツリー内の相関要素を探す。例えば、トランザクション・ツリーの２箇所以上で一組のイベントが起こると、一実施例において、ツリー解析部４０はそのグループを１つのパターンであると宣言する。一実施例において、これは、ツリーにおける全ての出来事の代わりに、新たに特定されたパターンの表現を用い、結果として得られたツリーに対してパターン検出を行うことにより実行される。

【００５３】

一実施例において、パターン識別部３６は発見されたパターンをパターン発見モジュール３０に供給する。一実施例において、パターン発見モジュール３０は、コンソール・インターフェース１６を介して、新たに発見されたパターンをユーザに対して表示する。ユーザは、目的に応じた多数の方法でパターンを利用する。例えば、ユーザは、イベント・パターンを調べることにより、該イベント・パターンが悪意のある攻撃か、無害な行為かを判断することができる。

【００５４】

一実施例において、ユーザ・インターフェースは、ユーザが発見されたパターンのいずれかを規則エンジン１８によって利用される相関則に直接変換できるようにする規則生成ツールをユーザに提供する。そして、規則エンジン１８は、将来供給される（あるいは再生された）全てのイベント・ストリームにおけるパターンの特定なインスタンスを検出及び発見する。このように、新たに発見されたパターンが「既知の」パターンとなり得る。

【００５５】

一実施例において、上記変換はワンクリック・インターフェースを用いて行うができる。例えば、コンソール・インターフェース１６では、発見された各パターンの隣に、「規則に変換」と表記されているボタンを表示することができ、該ボタンはパターンを規則エンジン１６における相関則として実現する機能を果たす。また規則に名称を付けるようユーザに促してもよい。

【００５６】

上述の説明において、パターン発見モジュール３０は、セキュリティ・イベントを入力として取り込み、該セキュリティ・イベントにおけるパターンを出力として生成するものであると説明した。しかしながら、パターン発見モジュールは、パターン内のパターンを検出するように構成することもできる。このような実施例において、パターン発見モジュールは、各固有のパターンを固有の要素として扱い、イベントのパターンを発見する場合と同様の方法でパターンを発見する。

【００５７】

このように、パターン発見機能を有するネットワーク・セキュリティ・システムについて説明した。上述の説明において、様々な具体的な値及びデータ構造に「セキュリティ・イベント」、「トランザクション」、「トランザクション・ツリー」、「要素」等の名称がつけられていた。更に、「パターン発見モジュール」や「パターン識別部」等、様々な特定のモジュールについて説明した。しかしながら、これらの名称は単に本発明の各種側面を説明し、例示するためのものであり、本発明の範囲を何ら限定するものではない。更に、図３のツリー・ビルダ３８及びツリー解析部４０等、様々なモジュールをソフトウェアあるいはハードウェア・モジュールとして実現することができ、またはその機能をモジュールに分割することなく実現することができる。本発明は、上で説明したか否かに関わらず、ソフトウェアあるいはハードウェアのモジュール構造を限定するものではない。

【図面の簡単な説明】

【００５８】

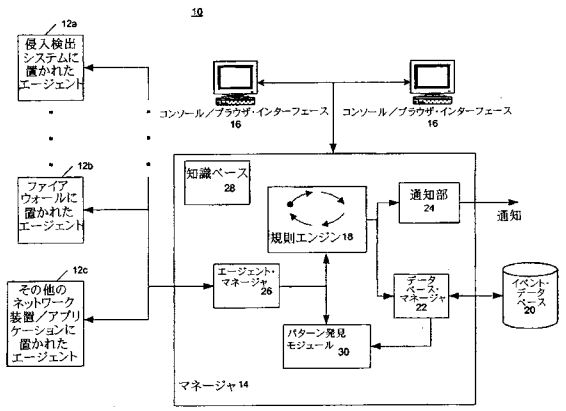
【図１】本発明の一実施例に係るネットワーク・セキュリティ・システムを示すブロック図である。

【図２】本発明の一実施例に係るパターン発見モジュール示すブロック図である。

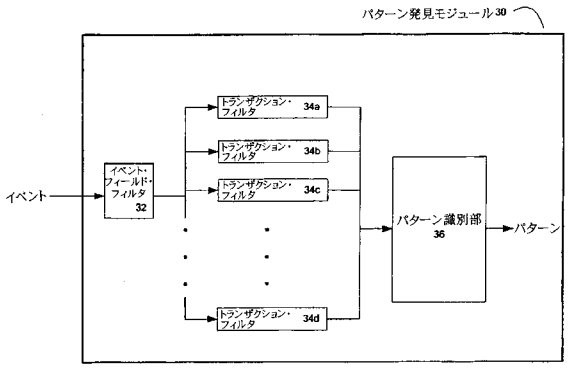
【図３】本発明の一実施例に係るパターン識別部を示すブロック図である。

【図４】本発明の一実施例に係るトランザクション・ツリーを構築する処理を示すフローチャートである。

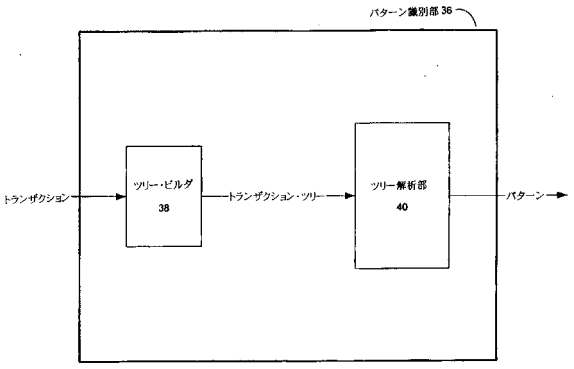
【図１】



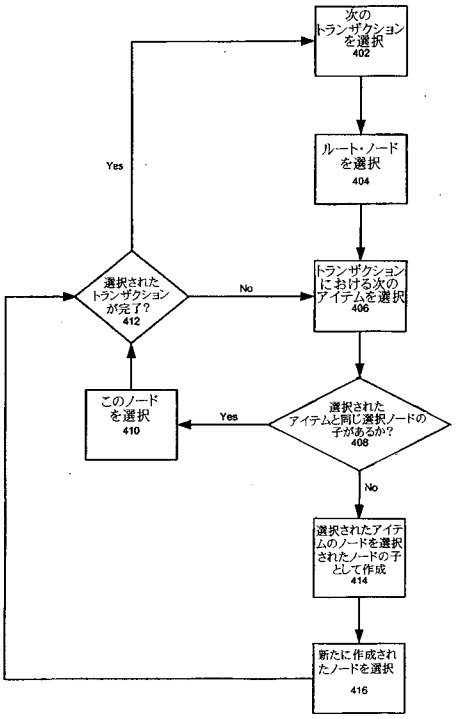
【図２】



【図３】



【図４】


次の
トランザクション
を選択
402
ルート・ノード
を選択
404
トランザクション
における次の
アイテムを選択
406
選択された
アイテムと同じ選択ノードの
子があるか?
408
Yes
No
このノード
を選択
410
選択された
トランザクション
が完了?
412
Yes
No
選択されたアイテム
のノードを選択
されたノードの子
として作成
414
新たに作成された
ノードを選択
416

フロントページの続き

審査官　和田　財太

(56)参考文献　米国特許出願公開第２００４／０１５７１９（ＵＳ，Ａ１）
英国特許出願公開第２３９１６５０（ＧＢ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 21/00-21/24
G06F 13/00
H04L 12/56