(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2016-534479
(P2016-534479A)
(43)公表日　平成28年11月4日(2016.11.4)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 21/56 (2013.01)	G06F 21/56	

審査請求 未請求　予備審査請求 未請求　（全 35 頁）

(21)出願番号	特願2016-542827 (P2016-542827)
(86)(22)出願日	平成26年9月12日 (2014.9.12)
(85)翻訳文提出日	平成28年5月6日 (2016.5.6)
(86)国際出願番号	PCT/US2014/055469
(87)国際公開番号	WO2015/038944
(87)国際公開日	平成27年3月19日 (2015.3.19)
(31)優先権主張番号	61/960,209
(32)優先日	平成25年9月12日 (2013.9.12)
(33)優先権主張国	米国 (US)

(71)出願人　516074252
ヴァーセック・システムズ・インコーポレーテッド
ＶＩＲＳＥＣ　ＳＹＳＴＥＭＳ，ＩＮＣ．
アメリカ合衆国，カリフォルニア州　９５０５４，サンタ　クララ，スイート　４３０，オールド　アイアンサイズ　ドライブ　４６９９
(74)代理人　100087941
弁理士　杉本　修司
(74)代理人　100086793
弁理士　野田　雅士
(74)代理人　100112829
弁理士　堤　健郎

最終頁に続く

(54)【発明の名称】マルウェアのランタイム中の自動検出

(57)【要約】
【課題】マルウェアがその悪質な目的を成功裏に実行する前にマルウェア攻撃を正確に識別する方法を提供する。
【解決手段】ロードタイム中にコンピュータアプリケーションのモデルを抽出することと、コンピュータアプリケーションのモデルを格納することと、ランタイムにデータを収集するように、コンピュータアプリケーションに指示を挿入することと、１つまたは複数のセキュリティイベントを検出するように、ランタイムに収集されたデータをコンピュータアプリケーションの格納されたモデルと照合して解析することと、状態マシンを用いて、１つまたは複数のセキュリティイベントを追跡することと、を含む。
【選択図】図７

【特許請求の範囲】
【請求項１】
コンピュータにより実行される方法であって、
ロードタイム中にコンピュータアプリケーションのモデルを抽出することと、
前記コンピュータアプリケーションの前記モデルを格納することと、
ランタイムにデータを収集するように、前記コンピュータアプリケーションに指示を挿入することと、
１つまたは複数のセキュリティイベントを検出するように、ランタイムに収集された前記データを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析することと、
状態マシンを用いて、前記１つまたは複数のセキュリティイベントを追跡することと、
を含む方法。
【請求項２】
請求項１に記載の方法において、前記コンピュータアプリケーションの前記モデルを抽出することが、前記コンピュータアプリケーションから遷移マッピングデータを抽出すること、前記コンピュータアプリケーションからメモリマッピングデータを抽出すること、前記コンピュータアプリケーションからソフトスポットデータを抽出すること、および前記コンピュータアプリケーションによって呼び出されるＯＳ関数およびシステムコールを抽出すること、のうち１つまたは複数を含む方法。
【請求項３】
請求項１に記載の方法において、前記コンピュータアプリケーションの前記モデルを抽出することが、少なくとも一部においてコード逆アセンブラを用いて達成される方法。
【請求項４】
請求項１に記載の方法において、前記コンピュータアプリケーションがバイナリ形式またはインタプリタ形式である方法。
【請求項５】
請求項１に記載の方法において、さらに、
ロードタイム中に保全性のために前記コンピュータアプリケーションをチェックすることを含む方法。
【請求項６】
請求項５に記載の方法において、保全性のために前記コンピュータアプリケーションをチェックすることが、チェックサムを計算することを含む方法。
【請求項７】
請求項１に記載の方法において、前記モデルを格納することが、前記コンピュータアプリケーションをモデル化する１つまたは複数のテーブルを含むデータベースに前記モデルを格納することを含む方法。
【請求項８】
請求項１に記載の方法において、前記モデルを格納することが、前記モデルをリモートシステム上のデータベースに格納することを含む方法。
【請求項９】
請求項８に記載の方法において、さらに、
前記コンピュータアプリケーションの前記モデルを、前記データベースに格納する前記リモートシステムへの伝送用にパッケージ化することを含む方法。
【請求項１０】
請求項９に記載の方法において、さらに、
前記伝送を保証するように、前記伝送内にカナリアを配置することを含む方法。
【請求項１１】
請求項１に記載の方法において、前記コンピュータアプリケーションに指示を挿入することが、少なくとも一部においてダイナミックバイナリ解析エンジンまたはバイトコード・インストルメンテーション・エンジンを用いて達成される方法。

【請求項１２】
請求項１に記載の方法において、さらに、
ランタイムに収集した前記データを、１つまたは複数のプロセスへの伝送用にパッケージ化することを含む方法。
【請求項１３】
請求項１２に記載の方法において、１つまたは複数のプロセスがリモートシステム上に存在する方法。
【請求項１４】
請求項１２に記載の方法において、さらに、
前記伝送を保証するように、前記伝送内にカナリアを配置することを含む方法。
【請求項１５】
請求項１に記載の方法において、ランタイムに収集された前記データが、前記コンピュータアプリケーションの１つまたは複数のスレッド用のデータを含む方法。
【請求項１６】
請求項１に記載の方法において、ランタイムに収集された前記データを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析することが、遷移データを解析すること、ＯＳ関数を解析すること、システムコールを解析すること、メモリ書込みを解析すること、およびソフトスポットデータを解析することのうちの１つまたは複数を含む方法。
【請求項１７】
請求項１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、定義済みのシーケンスに基づいて前記１つまたは複数のセキュリティイベントを相関させることを含む方法。
【請求項１８】
請求項１７に記載の方法において、前記定義済みのシーケンスが不変チェーンに基づく方法。
【請求項１９】
請求項１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、当該イベント用のフォレンジックデータを捕捉することを含む方法。
【請求項２０】
請求項１に記載の方法において、前記１つまたは複数のセキュリティイベントが、重大度レベルを用いて追跡される方法。
【請求項２１】
請求項１に記載の方法において、さらに、
前記１つまたは複数のセキュリティイベントを追跡することに応答して、１つまたは複数の措置をとることを含む方法。
【請求項２２】
請求項２１に記載の方法において、前記１つまたは複数の措置がシステムによって自動的にとられる方法。
【請求項２３】
請求項２１に記載の方法において、前記１つまたは複数の措置がユーザによってとられる方法。
【請求項２４】
請求項２１に記載の方法において、前記１つまたは複数の措置が、前記コンピュータアプリケーションを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッドを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッド上のソケットを閉じること、および前記１つまたは複数のセキュリティイベントに応答して警報を発することのうちのいずれかを含む方法。
【請求項２５】
請求項２１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡

することに応答してとられる前記１つまたは複数の措置が、調整可能である方法。
【請求項２６】
コンピュータにより実行される方法であって、
ロードタイム中にコンピュータアプリケーションのモデルを抽出することと、
前記コンピュータアプリケーションの前記モデルを格納することと、
ランタイムにおいてデータを収集するように、前記コンピュータアプリケーションに指示を挿入すること、を含む方法。
【請求項２７】
請求項２６に記載の方法において、前記コンピュータアプリケーションの前記モデルを抽出することが、前記コンピュータアプリケーションから遷移マッピングデータを抽出すること、前記コンピュータアプリケーションからメモリマッピングデータを抽出すること、前記コンピュータアプリケーションからソフトスポットデータを抽出すること、および前記コンピュータアプリケーションによって呼び出されるＯＳ関数およびシステムコールを抽出すること、のうち１つまたは複数を含む方法。
【請求項２８】
請求項２６に記載の方法において、前記コンピュータアプリケーションのモデルを抽出することが、少なくとも一部においてコード逆アセンブラを用いて達成される方法。
【請求項２９】
請求項２６に記載の方法において、前記コンピュータアプリケーションはバイナリ形式またはインタプリタ形式である方法。
【請求項３０】
請求項２６に記載の方法において、さらに、
ロードタイム中に保全性のために前記コンピュータアプリケーションをチェックすることを含む方法。
【請求項３１】
請求項３０に記載の方法において、保全性のために前記コンピュータアプリケーションをチェックすることが、チェックサムを計算することを含む方法。
【請求項３２】
請求項２６に記載の方法において、前記モデルを格納することが、前記コンピュータアプリケーションをモデル化する１つまたは複数のテーブルを含むデータベースに前記モデルを格納することを含む方法。
【請求項３３】
請求項２６に記載の方法において、前記モデルを格納することが、前記モデルをリモートシステム上のデータベースに格納することを含む方法。
【請求項３４】
請求項３３に記載の方法において、さらに、
前記コンピュータアプリケーションの前記モデルを、前記データベースに格納する前記リモートシステムへの伝送用にパッケージ化することを含む方法。
【請求項３５】
請求項３４に記載の方法において、さらに、
前記伝送を保証するように、前記伝送内にカナリアを配置することを含む方法。
【請求項３６】
請求項２６に記載の方法において、前記コンピュータアプリケーションに指示を挿入することが、少なくとも一部においてダイナミックバイナリ解析エンジンまたはバイトコード・インストルメンテーション・エンジンを用いて達成される方法。
【請求項３７】
請求項２６に記載の方法において、さらに、
ランタイムに収集した前記データを、１つまたは複数のプロセスへの伝送用にパッケージ化することを含む方法。
【請求項３８】

請求項３７に記載の方法において、１つまたは複数のプロセスがリモートシステム上に存在する方法。
【請求項３９】
請求項３７に記載の方法において、さらに、
前記伝送を保証するように、前記伝送内にカナリアを配置することを含む方法。
【請求項４０】
請求項２６に記載の方法において、ランタイムに収集された前記データが、前記コンピュータアプリケーションの各スレッド用のデータを含む方法。
【請求項４１】
コンピュータにより実行される方法であって、
１つまたは複数のセキュリティイベントを検出するように、コンピュータアプリケーションのランタイムに収集されたデータを前記コンピュータアプリケーションの格納されたモデルと照合して解析することと、
状態マシンを用いて、前記１つまたは複数のセキュリティイベントを追跡することと、
を含む方法。
【請求項４２】
請求項４１に記載の方法において、さらに、
ランタイムに収集された前記データをリモートシステム上のプロセスから受信することを含む方法。
【請求項４３】
請求項４１に記載の方法において、さらに、
前記コンピュータアプリケーションの前記格納されたモデルをリモートシステム上のプロセスから受信することを含む方法。
【請求項４４】
請求項４１に記載の方法において、ランタイムに収集された前記データを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析することが、遷移データを解析すること、ＯＳ関数を解析すること、システムコールを解析すること、メモリ書込みを解析すること、およびソフトスポットデータを解析することのうちの１つまたは複数を含む方法。
【請求項４５】
請求項４１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、定義済みのシーケンスに基づいて前記１つまたは複数のセキュリティイベントを相関させることを含む方法。
【請求項４６】
請求項４５に記載の方法において、前記定義済みのシーケンスが不変チェーンに基づく方法。
【請求項４７】
請求項４１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、フォレンジックデータを捕捉することを含む方法。
【請求項４８】
請求項４１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、重大度レベルの使用を含む方法。
【請求項４９】
請求項４１に記載の方法において、さらに、
前記１つまたは複数のセキュリティイベントを追跡することに応答して、１つまたは複数の措置をとることを含む方法。
【請求項５０】
請求項４９に記載の方法において、前記１つまたは複数の措置がシステムによって自動的にとられる方法。
【請求項５１】

請求項４９に記載の方法において、前記１つまたは複数の措置がユーザによってとられる方法。
【請求項５２】
請求項４９に記載の方法において、前記１つまたは複数の措置が、前記コンピュータアプリケーションを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッドを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッド上のソケットを閉じること、および前記１つまたは複数のセキュリティイベントに応答して警報を発することのうちのいずれかを含む方法。
【請求項５３】
請求項４９に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することに応答してとられる前記１つまたは複数の措置が、調整可能である方法。
【請求項５４】
ロードタイム中にコンピュータアプリケーションのモデルを抽出するクライアントであって、
前記コンピュータアプリケーションの前記モデルを格納し、
ランタイムにデータを収集するように、前記コンピュータアプリケーションに指示を挿入する、クライアントと、
１つまたは複数のセキュリティイベントを検出するように、ランタイムに収集されたデータを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析する解析エンジンであって、
状態マシンを用いて、前記１つまたは複数のセキュリティイベントを追跡する解析エンジンと、を備えたシステム。
【請求項５５】
請求項５４に記載のシステムにおいて、前記コンピュータアプリケーションの前記モデルを抽出することが、前記コンピュータアプリケーションから遷移マッピングデータを抽出すること、前記コンピュータアプリケーションからメモリマッピングデータを抽出すること、前記コンピュータアプリケーションからソフトスポットデータを抽出すること、および前記コンピュータアプリケーションによって呼び出されるＯＳ関数およびシステムコールを抽出すること、のうち１つまたは複数を含むシステム。
【請求項５６】
請求項５４に記載のシステムにおいて、前記コンピュータアプリケーションの前記モデルを抽出することが、少なくとも一部においてコード逆アセンブラを用いて達成されるシステム。
【請求項５７】
請求項５４に記載のシステムにおいて、前記コンピュータアプリケーションがバイナリ形式またはインタプリタ形式であるシステム。
【請求項５８】
請求項５４に記載のシステムにおいて、前記クライアントが、さらに、ロードタイム中に保全性のために前記コンピュータアプリケーションをチェックするシステム。
【請求項５９】
請求項５８に記載のシステムにおいて、保全性のために前記コンピュータアプリケーションをチェックすることが、チェックサムを計算することを含むシステム。
【請求項６０】
請求項５４に記載のシステムにおいて、前記モデルを格納することが、前記コンピュータアプリケーションをモデル化する１つまたは複数のテーブルを含むデータベースに前記モデルを格納することを含むシステム。
【請求項６１】
請求項６０に記載のシステムにおいて、前記モデルを格納することが、前記モデルをリモートシステム上のデータベースに格納することを含むシステム。
【請求項６２】

請求項６１に記載のシステムにおいて、前記クライアントが、さらに、前記コンピュータアプリケーションの前記モデルを、前記データベースに格納する前記リモートシステムへの伝送用にパッケージ化するシステム。
【請求項６３】
請求項６２に記載のシステムにおいて、前記クライアントが、さらに、前記伝送を保証するように、前記伝送内にカナリアを配置するシステム。
【請求項６４】
請求項５４に記載のシステムにおいて、前記コンピュータアプリケーションに指示を挿入することが、少なくとも一部においてダイナミックバイナリ解析エンジンまたはバイトコード・インストルメンテーション・エンジンを用いて達成されるシステム。
【請求項６５】
請求項５４に記載のシステムにおいて、前記クライアントが、さらに、ランタイムに収集した前記データを、前記解析エンジンへの伝送用にパッケージ化するシステム。
【請求項６６】
請求項６５に記載のシステムにおいて、前記クライアントが、さらに、前記伝送を保証するように、前記伝送内にカナリアを配置するシステム。
【請求項６７】
請求項６５に記載のシステムにおいて、前記解析エンジンへの伝送が、前記クライアントと前記解析エンジンとの間のトランスポートチャネル上で収集された前記データを送信することを含むシステム。
【請求項６８】
請求項５４に記載のシステムにおいて、ランタイムに収集された前記データが、前記コンピュータアプリケーションの１つまたは複数のスレッド用のデータを含むシステム。
【請求項６９】
請求項５４に記載のシステムにおいて、ランタイムに収集された前記データを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析することが、遷移データを解析すること、ＯＳ関数を解析すること、システムコールを解析すること、メモリ書込みを解析すること、およびソフトスポットデータを解析することのうちの１つまたは複数を含むシステム。
【請求項７０】
請求項５４に記載のシステムにおいて、前記１つまたは複数のセキュリティイベントを追跡することが、定義済みのシーケンスに基づいて前記１つまたは複数のセキュリティイベントを相関させることを含むシステム。
【請求項７１】
請求項７０に記載のシステムにおいて、前記定義済みのシーケンスが不変チェーンに基づくシステム。
【請求項７２】
請求項５４に記載のシステムにおいて、前記１つまたは複数のセキュリティイベントを追跡することが、当該イベント用のフォレンジックデータを捕捉することを含むシステム。
【請求項７３】
請求項５４に記載のシステムにおいて、前記１つまたは複数のセキュリティイベントが、重大度レベルを用いて追跡されるシステム。
【請求項７４】
請求項５４に記載のシステムにおいて、前記クライアントが、１つまたは複数のセキュリティイベントを追跡することに応答して、１つまたは複数の措置をとるシステム。
【請求項７５】
請求項７４に記載のシステムにおいて、前記１つまたは複数の措置がシステムによって自動的にとられるシステム。
【請求項７６】

請求項７４に記載のシステムにおいて、前記１つまたは複数の措置がユーザによってとられるシステム。
【請求項７７】
請求項７４に記載のシステムにおいて、前記１つまたは複数の措置が、前記コンピュータアプリケーションを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッドを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッド上のソケットを閉じること、および前記１つまたは複数のセキュリティイベントに応答して警報を発することのうちのいずれかを含むシステム。
【請求項７８】
請求項７４に記載のシステムにおいて、前記１つまたは複数のセキュリティイベントを追跡することに応答してとられる前記１つまたは複数の措置が、調整可能であるシステム。
【請求項７９】
請求項５４に記載のシステムにおいて、前記クライアントおよび前記解析エンジンの少なくとも一方が、スマートフォン、タブレット、ラップトップ、デスクトップ、およびハイエンドサーバのうちのいずれかにおいて動作するシステム。
【請求項８０】
請求項５４に記載のシステムにおいて、前記解析エンジンが、１つまたは複数のプロセッサからなるプロセッサファブリックを含むシステム。
【請求項８１】
請求項５４に記載のシステムにおいて、前記クライアントが、前記クライアントの１つまたは複数のプロセッサが停止または無応答状態の場合に、前記１つまたは複数のプロセッサを再始動させるクライアントデーモンを含むシステム。
【請求項８２】
請求項５４に記載のシステムにおいて、前記解析エンジンが、前記解析エンジンの１つまたは複数のプロセッサが停止または無応答状態の場合に、前記１つまたは複数のプロセッサを再始動させるアプリケーションデーモンを含むシステム。
【請求項８３】
請求項５４に記載のシステムにおいて、前記解析エンジンが、前記コンピュータアプリケーションの状態を表示するダッシュボードを含むシステム。
【請求項８４】
第１のプロセスおよび第２のプロセスを実行するプロセッサを備えた装置であって、
前記第１のプロセスが、ロードタイム中にコンピュータアプリケーションのモデルを抽出し、
前記第１のプロセスがさらに、前記コンピュータアプリケーションの前記モデルを格納し、
前記第２のプロセスが、ランタイムにデータを収集するように、前記コンピュータアプリケーションに指示を挿入する、装置。
【請求項８５】
請求項８４に記載の装置において、前記コンピュータアプリケーションの前記モデルを抽出することが、前記コンピュータアプリケーションから遷移マッピングデータを抽出すること、前記コンピュータアプリケーションからメモリマッピングデータを抽出すること、前記コンピュータアプリケーションからソフトスポットデータを抽出すること、および前記コンピュータアプリケーションによって呼び出されるＯＳ関数およびシステムコールを抽出すること、のうち１つまたは複数を含む装置。
【請求項８６】
請求項８４に記載の装置において、前記コンピュータアプリケーションの前記モデルを抽出することが、少なくとも一部においてコード逆アセンブラを用いて達成される装置。
【請求項８７】
請求項８４に記載の装置において、前記コンピュータアプリケーションがバイナリ形式

またはインタプリタ形式である装置。
【請求項８８】
請求項８４に記載の装置において、さらに、
ロードタイム中に保全性のために前記コンピュータアプリケーションをチェックする第３のプロセスを備えた装置。
【請求項８９】
請求項８８に記載の装置において、保全性のために前記コンピュータアプリケーションをチェックすることは、チェックサムを計算することを含む装置。
【請求項９０】
請求項８４に記載の装置において、データベースが、コンピュータアプリケーションをモデル化する１つまたは複数のテーブルを含む装置。
【請求項９１】
請求項８４に記載の装置において、前記データベースがリモートシステム上に存在する装置。
【請求項９２】
請求項９１に記載の装置において、さらに、
前記コンピュータアプリケーションの前記モデルを、前記データベースに格納する前記リモートシステムへの伝送用にパッケージ化する前記第１のプロセスを備えた装置。
【請求項９３】
請求項９２に記載の装置において、前記第１のプロセスが、さらに、前記伝送を保証するように、前記伝送内にカナリアを配置する装置。
【請求項９４】
請求項９１に記載の装置において、前記コンピュータアプリケーションに指示を挿入することが、少なくとも一部においてダイナミックバイナリ解析エンジンまたはバイトコード・インストルメンテーション・エンジンを用いて達成される装置。
【請求項９５】
請求項９１に記載の装置において、さらに、
ランタイムに収集した前記データを、１つまたは複数のプロセスへ伝送用にパッケージ化する第３のプロセスを備えた装置。
【請求項９６】
請求項９５に記載の装置において、前記１つまたは複数のプロセスがリモートシステム上に存在する装置。
【請求項９７】
請求項９５に記載の装置において、前記第３のプロセスが、さらに、前記伝送を保証するように、前記伝送内にカナリアを配置する装置。
【請求項９８】
請求項９１に記載の装置において、ランタイムに収集されたデータが、前記コンピュータアプリケーションの各スレッド用のデータを含む装置。
【請求項９９】
第１のプロセスおよび第２のプロセスを実行するプロセッサを備えた装置であって、
前記第１のプロセスが、１つまたは複数のセキュリティイベントを検出するように、コンピュータアプリケーションのランタイムに収集されたデータを前記コンピュータアプリケーションの格納されたモデルと照合して解析し、
前記第２のプロセスが、状態マシンを用いて、前記１つまたは複数のセキュリティイベントを追跡する装置。
【請求項１００】
請求項９９に記載の装置において、ランタイムに収集された前記データがリモートシステム上のプロセスから受信される装置。
【請求項１０１】
請求項９９に記載の装置において、前記コンピュータアプリケーションの前記格納され

たモデルが、リモートシステム上のプロセスから受信される装置。
【請求項１０２】
請求項９９に記載の装置において、ランタイムに収集された前記データを前記コンピュータアプリケーションの前記格納されたモデルと照合して解析することが、遷移データを解析すること、ＯＳ関数を解析すること、システムコールを解析すること、メモリ書込みを解析すること、およびソフトスポットデータを解析することのうちの１つまたは複数を含む装置。
【請求項１０３】
請求項９９に記載の装置において、前記１つまたは複数のセキュリティイベントを追跡することが、定義済みのシーケンスに基づいて前記１つまたは複数のセキュリティイベントを相関させることを含む装置。
【請求項１０４】
請求項１０３に記載の装置において、前記定義済みのシーケンスが不変チェーンに基づく装置。
【請求項１０５】
請求項９９に記載の装置において、前記１つまたは複数のセキュリティイベントを追跡することが、フォレンジックデータを捕捉することを含む装置。
【請求項１０６】
請求項９９に記載の装置において、前記１つまたは複数のセキュリティイベントを追跡することが、重大度レベルを用いることを含む装置。
【請求項１０７】
請求項９９に記載の装置において、さらに、
前記１つまたは複数のセキュリティイベントを追跡することに応答して、１つまたは複数の措置をとる第３のプロセスを備えた装置。
【請求項１０８】
請求項１０７に記載の装置において、前記１つまたは複数の措置が自動的にとられる装置。
【請求項１０９】
請求項１０７に記載の装置において、前記１つまたは複数の措置はユーザによってとられる装置。
【請求項１１０】
請求項１０７に記載の装置において、前記１つまたは複数の措置は、前記コンピュータアプリケーションを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッドを終了すること、前記コンピュータアプリケーションの１つまたは複数のスレッド上のソケットを閉じること、および前記１つまたは複数のセキュリティイベントに応答して警報を発することのうちのいずれかを含む装置。
【請求項１１１】
請求項１０７に記載の装置において、前記１つまたは複数のセキュリティイベントを追跡することに応答してとられる前記１つまたは複数の措置が、調整可能である装置。
【請求項１１２】
請求項１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、さらに、重大度および前記状態マシンを用いた追跡用の措置を伴う１つまたは複数の新しいセキュリティイベントを追加することを含む方法。
【請求項１１３】
請求項４１に記載の方法において、前記１つまたは複数のセキュリティイベントを追跡することが、さらに、重大度および前記状態マシンを用いた追跡用の措置を伴う１つまたは複数の新しいセキュリティイベントを追加することを含む方法。
【請求項１１４】
請求項５４に記載のシステムにおいて、前記クライアントが、さらに、重大度および前記状態マシンを用いた追跡用の措置を伴う１つまたは複数の新しいセキュリティイベント

を追加すること含むシステム。
【請求項１１５】
請求項９９に記載の装置において、前記第２のプロセスが、さらに、重大度および前記状態マシンを用いた追跡用の措置を伴う１つまたは複数の新しいセキュリティイベントを追加することを含む装置。
【発明の詳細な説明】
【関連出願】
【０００１】
本出願は、2013年9月12日に出願された米国仮出願第61/960,209号の利益を主張する。この米国仮特許出願の全教示内容は、参照をもって本明細書に取り入れたものとする。
【背景技術】
【０００２】
サイバー攻撃は日々ますます高度化している。大量の攻撃が特定のアプリケーションにおける特定の脆弱性に付け込むことを対象としている。これらの攻撃は、明らかに不正にみえるネットワーク動作を引き起こさないので、ネットワーク層では識別不可能である。これらの標的型攻撃に対処するために、ベンダの多くは製品を開発している。このような製品としては、例えば、アプリケーションの挙動の追跡を試みる次世代ファイアウォールや、サンドボックスで不審なコードの実行を試みてその不審なコードが不正動作を起こすのを待つというサンドボックス技術などがある。しかし、これらの場合、マルウェアは、単にその挙動にわずかに変更するか、あるいはその悪意ある目的を実行するために長時間待機する。このような挙動の変化によって、製品は攻撃の挙動を認識しにくくなるため、製品のマルウェア検出能力が著しく低下する。
【発明の概要】
【課題を解決するための手段】
【０００３】
単一および複数アプリケーション、閉鎖および分散型アプリケーション、独立型アプリケーション、ウェブ上のアプリケーション、およびクラウド上のアプリケーションを含むコンピュータアプリケーションであるが、これらに限定はされるわけではないコンピュータアプリケーションは、マルウェア攻撃に対して脆弱である。今日、コンピュータアプリケーションに対するマルウェア攻撃の大多数は、コンピュータアプリケーションの実行プロセスに不正コンテンツを挿入してその後に実行するという悪意ある行為者の才能に帰結する。そのような不正コンテンツを挿入する工程では、不適当な入力検証を行う不十分に設計されたコードを識別しこれに付け込む。現時点のサイバーセキュリティー技術では、サンドボックスにおいて、アプリケーションにおける不正コンテンツを観察すること、アプリケーションの挙動を追跡すること、または不審なコードの挙動を選別することを試みている。これら技術には、悪質なコンテンツの挿入を示すイベントを確実に検出するために、十分に小さい粒度でリアルタイムにコンピュータアプリケーションを調べる能力は無い。また、これら技術には、マルウェアがその悪質な目的を成功裏に実行する前にマルウェア攻撃を正確に識別するために、そのようなイベントを長い時間かけて追跡し相関させる能力は無い。
【０００４】
ある例示的な方法およびその装置は、ロードタイム中（読込み時）にコンピュータアプリケーションのモデルを抽出して格納する。この例示的な方法およびその装置はまた、ランタイムにデータを収集するように、ロードタイムにコンピュータアプリケーションに指示を挿入する。ランタイムに収集されたデータは、コンピュータアプリケーションの格納されたモデルと照合して解析されて、１つまたは複数のセキュリティイベントが検出される。この方法およびその装置は、状態マシンを用いて、攻撃者によって引き起こされた１つ以上のセキュリティイベントを追跡する。
【０００５】
いくつかの実施形態において、上記方法およびその装置は、コンピュータアプリケーシ

ョンデータのモデルの一部として、以下のうちの１つ以上を抽出してもよい：遷移マッピングデータ、メモリマッピングデータ、ソフトスポット（脆弱箇所）データ、および／またはコンピュータアプリケーションが呼び出す、アクセス許可および権限に影響するＯＳ関数またはシステムコール。これら情報はモデルデータベースに格納されてもよい。さらに、上記方法およびその装置は、少なくとも一部においてコード逆アセンブラを用いてコンピュータアプリケーションのモデルを抽出してもよい。抽出されたコンピュータアプリケーションはバイナリ形式やインタプリタ形式を含む種々の形式であってもよい。

【０００６】

例示的な実施形態において、上記方法およびその装置は、ロードタイム中に保全性のためにコンピュータアプリケーションをチェックしてもよい。上記方法およびその装置は、コードのＭＤ５ハッシュなどのチェックサムを計算することにより、または信頼できるチェックサム検査サービスを用いて、保全性のために前記コンピュータアプリケーションをチェックしてもよい。

【０００７】

いくつかの実施形態において、モデルデータベースは、コンピュータアプリケーションをモデル化する１つ以上のテーブルを含む。さらに、モデルデータベースはローカルまたはリモートシステム上にあってもよい。モデルデータベースがリモートシステム上にある場合、上記方法およびその装置は、コンピュータアプリケーションのモデルを、リモートシステムへ伝送してデータベースに格納するために、パッケージ化してもよい。データベースのパッケージ化されたモデルは、ＴＣＰ／ＩＰまたはＵＤＰなどの標準ベースのトランスポートプロトコルを用いて送信されてもよい。

【０００８】

例示的な実施形態において、上記方法およびその装置は、ダイナミックバイナリ解析エンジンまたはバイトコード・インストルメンテーション・エンジン（バイトコードデータ・データ記録手段装備・エンジン）を用いて、ロードタイムにおいてコンピュータアプリケーションにインストルメンテーション（データ記録手段装備）の指示を挿入する。データ記録手段が装備されたアプリケーションを実行すると、ランタイムに収集されたデータを、解析用に他のプロセスへの伝送のために、パッケージ化してもよい。この解析用の他のプロセスは、ローカルまたはリモートシステム上にあってもよい。さらに、ランタイムに収集されたデータは、コンピュータアプリケーションの１つ以上のスレッドのデータを含んでもよい。

【０００９】

いくつかの実施形態において、ランタイムに収集されたデータをコンピュータアプリケーションの格納されたモデルと照合して解析する時、上記方法およびその装置は、以下のうちの１つ以上を解析してもよい：遷移データ、アクセス許可および権限に影響する重要なＯＳ関数およびシステムコール、メモリ書込み、ヒープ割当てまたは割当て解除、および／またはソフトスポットデータ。

【００１０】

例示的な実施形態において、状態マシンを用いて１つ以上のセキュリティイベントを追跡することは、定義済みのシーケンスに基づいてイベントを相関させることを含む。１つ以上のセキュリティイベントを追跡することは、そのイベントのためのフォレンジックデータを入手することも含んでよい。ある実施形態では、１つ以上のセキュリティイベントを、重大度レベルを用いて追跡してもよい。さらに、１つ以上の措置を、１つ以上のセキュリティイベントの発生に応じてとってもよい。ある例示的な実施形態では、イベントに応じた１つ以上の措置を、システムにより自動的にとってもよい。他の例示的な実施形態では、１つ以上の措置を、ユーザによる手動介入の後にとることもできる。いくつかの実施形態では、１つ以上の措置は、以下のうちの１つ以上を含んでもよい：コンピュータアプリケーションの１つ以上のスレッドを終了すること、コンピュータアプリケーションの１つ以上のスレッド上の通信ソケットを閉じること、アプリケーションを終了すること、イベントを記録すること、および／または１つ以上のセキュリティイベントに応答して警

報を発すること。
【００１１】
前述の内容は、以下の本発明の例示的な実施形態のさらに具体的な説明から明らかになるであろう。添付図面において、異なる図をとおして、同一の構成要素は同一の符号で示す。図面は必ずしも原寸に比例しておらず、本発明の実施形態の説明に重点が置かれている。
【図面の簡単な説明】
【００１２】
【図１】高度な持続的マルウェア脅威の構成の一例を示す図である。
【図２】コード実行型攻撃を示す一連の不変イベントを示す図である。
【図３】クライアントにより実行されるロードタイム動作のフローチャートである。
【図４】クライアントと解析エンジンとの間でデータを送信するために使用されるプロトコルデータユニット（ＰＤＵ）を示す図である。
【図５】ランタイムに収集したデータに基づくイベントを生成するために解析エンジンが用いるプロセスのフローチャートである。
【図６】生成されたイベントを追跡するために使用されるイベントチェーン状態マシンを示す図である。
【図７】クライアントおよび解析エンジンのブロック図である。
【図８】本発明の各実施形態が実装されるコンピュータネットワークまたは同様のデジタルプロセス環境を示す図である。
【図９】図８に示すコンピュータシステムにおけるコンピュータ（例えば、クライアントプロセッサ/装置、またはサーバコンピュータ）の内部構造を示す図である。
【発明を実施するための形態】
【００１３】
発明の例示的な実施形態を以下に説明する。
本願明細書で参照する、全ての特許、全ての公開された出願、および全ての引例の教示内容は、参照をもって本明細書に取り入れたものとする。
【００１４】
図１に、高度な持続的マルウェア脅威の一例を示す。この場合、悪意ある行為者（すなわちハッカ）が自身のインフラストラクチャ１０２から、エヌマップ（ｎｍａｐ）１０４などのスキャンツールを用いて、企業のインフラストラクチャ１０８に向けてウェブを遠隔的にスキャンする。このスキャンにより脆弱性（既知の脆弱性またはゼロデイ脆弱性）を有するサーバを発見すると、行為者はシェルコード１０６をインストールして離れた企業サーバ１１０を乗っ取り、企業ネットワークにアクセスする。ネットワークの内部に入ると、悪意ある行為者は追加のツール１０４をロードする。これらツールとしては、エヌマップ、ポートスキャナー、パスワード・クラッキングツール、ｆｔｐクライアント、圧縮ツール、ハッシング、および／または暗号化・復号化ツールが挙げられる。
【００１５】
行為者は、企業インフラストラクチャにアクセスし、高い権限を有するユーザがデータベースやファイルリポジトリなどの価値ある標的にログインするためのマシン１１４または１１６を探して、ユーザのアクセス認証情報（access credentials）を無効化し、追加のハッキングツールを忍ばせる所を探す。脆弱なアプリケーションを有するマシン１１４または１１６を見つけると、悪意ある行為者は認証情報を無効化し、侵入し、その後標的１１８にアクセスする。標的がアクセスされると、ハッキングのための追加のツール１０４が標的にロードされる。マルウェアはまた、権限を有するユーザが自身のマシンをコーヒーショップ、空港、ホテルなどの無防備なネットワークを介して持ち込んだ際に、ユーザのスマートフォンやラップトップなどの移動端末装置に入り込める。これとは別に、内部ユーザが標的マシンを感染させるかもしれない。
【００１６】
高度なパスワード・クラッキングツールまたはスヌーピング・イントラネット・トラフ

ィックを用いることによって、マルウェアは管理ユーザ１１６の認証情報を取得し得る。悪意ある行為者は、認証情報を取得した後、データベースやファイルリポジトリ１１８に罰せされることなく接続でき、実名、自宅住所、社会保障番号、運転免許証、生年月日、医療記録、クレジット/デビットカードなどの金融情報、電話番号、電子メールアドレス、ユーザ名およびパスワード、ならびに保険情報などの価値あるデータを抽出し得る。悪意ある行為者は、これら情報を随意に圧縮および暗号化でき、企業セキュリティ分析者の注意を引かないように小さいサイズに分割してハッカ・コマンド・コントロールセンター１１２にアップロードし得る。目的を達成するために、悪意ある行為者は、ハッカ・コマンド・コントロールセンター１１２のＩＰアドレスを毎日変更したり、プロキシを使用したりしているため、企業の侵入検知システムはパターン化した状況を確認できない。企業は通常毎日１０ＧＢを超えるデータを送受信しているため、短いバーストでの比較的少量のデータのアップロードは気付かれないことが多い。

【００１７】

このような場合の一連のイベントを確実に検出しブロックできることが、高度な持続的脅威を妨害するための鍵となる。最近のサイバー・セキュリティツールには４つの主要な欠点がある。まず、これらツールは十分に小さい粒度でアプリケーションを検査していない。この機能なしでは、攻撃の多くの兆候を確実に見分けることができない。次に、これらツールは長い時間をかけて攻撃を追跡する機能を有しない。通常のセキュリティ情報およびイベント管理（ＳＩＥＭ）システムは、通常２４時間イベント（大きすぎる粒度のイベント）を相関させるだけである。マルウェアは、検出を逃れるために十分長い時間休止状態に留まることがある。長時間確実に攻撃を追跡する機能なしでは、攻撃に寄与する個々の不正イベントは独立した相関のないイベントにみえる。

【００１８】

また、これらサイバー・セキュリティツールは、攻撃の前兆となるイベントの閾値の設定をセキュリティ分析者に頼っている。例えば、セキュリティ分析者が、毎時ある回数のポートスキャンおよびログインの試みが日常的に行われることを当然とみなすかもしれない。この場合、問題は、警報のきっかけとなる、発信元毎の毎時のポートスキャンの試みの数が一体いくつであれば多すぎるのか、ということである。警報の発生が早すぎる場合、分析者は、悪意のないポートスキャンまたは合法的なユーザによるログインの失敗までも調査するかもしれない。警報の発生が遅すぎる場合、マルウェアによる攻撃はすでに成功しているかもしれない。さらに、これらツールは、攻撃を効果的に検出するには基本情報が不十分である。真正な警報を発生するには、良性イベントと不正イベントを確実に識別するツールが必要である。悪意ある行為者は、同じサインもしくは既知の不審なネットワークまたはアプリケーション挙動を認識するであろう。したがって、悪意ある行為者は攻撃の挙動を微調整できる。例えば、データをコード化する暗号化を用いたり、ＩＰとポートの組み合わせを変更したり、基本情報を用いて指定した挙動に基づく検出を逃れるために攻撃速度を落としたりできる。

【００１９】

図２は、コード実行攻撃（コードを実行することによる攻撃）の一連の不変（動かし難い）イベントを示す。複数のコード実行攻撃のような種々のマルウェア攻撃について、マルウェア攻撃の対応するイベントチェーンは一連の不変イベントによって表現される。これらイベントは、特定のマルウェア攻撃を追跡するコンピュータのプロセスメモリ内に格納される。以下に、コード実行攻撃の符号２０２から符号２１６までの一連の不変イベントによって表現された挙動について述べる。

【００２０】

コード実行攻撃を実行するにあたり、悪意ある行為者がパッチされていないアプリケーションまたはいわゆるゼロデイ脆弱性を探し当てると、行為者は特別に練ったペイロードである不良コンテンツを作成する。このペイロードは２０２において、直接またはネットワークを介して、コンピュータシステムの中央処理装置（ＣＰＵ）上で実行される標的プロセスに送信され、ＣＰＵがコンピュータアプリケーションの指示を呼び出すことを阻止

し、このペイロードは代わりに不正ペイロードの指令による指示を呼び出す。この特別に練られたペイロードは、標的となるアプリケーションに応じて、多くのメカニズム、例えばネットワークソケットやキーボードさらにはファイルまでを介して、アプリケーションに挿入される。

【００２１】

２０６において、攻撃中に挿入された不正ペイロードを起動するために、マルウェアは多数の攻撃ベクトルのうちの1つを強化する。コード実行攻撃の場合、２０４において、マルウェアはバッファのエラーまたはユーザの無知に付け込む。攻撃ベクトルの他の例としては、単純なスタック・スマッシュ（Stack Smashing）アプローチ、フォーマット指定子の使用、擬似乱数スタック・カナリアの発見、例外ハンドラ・テーブルのオーバーラン、またはリターン・オリエンテッド・プログラミング（ＲＯＰ）ガジェット、ならびに多数の追加のベクトルが挙げられる。

【００２２】

本格的な不正動作を開始する前に、マルウェアは検出技術が働かなくなるように十分に長い時間休止状態になる。例えば、不審な動作に対してＥメール検査するサンドボックス技術は、最終的に諦めて受信者にＥメールを届けるに違いない。その後のある時点で、マルウェアは２１２において、その意図することの実行を開始するために、既存のアプリケーションスレッドを用いるかまたは1つ以上の新規のスレッドをスピンする。場合によっては、既存のスレッドの使用はユーザの注意を引き、新しいスレッドのスピンはユーザに気付かれない。これは、ほとんどのセキュリティ対策が、新しいスレッドのスピンが不正動作であるか無害動作であるかを判断できないからである。アプリケーション内に糸口となるものを作成して、マルウェアは２１０において、悪意ある行為者のコマンド・コントロールセンター（Ｃ＆Ｃ）との接続性を確立する。企業ファイアウォール内部からの接続性が確立すると、マルウェアは２０８において、上記スレッド上で、パスワード・クラッキングツール、ポート・スキャンツール、暗号化ツールなどのさらなるマルウェアを、ファイル・ブラックリスティング対策の注意を引かないように複数のビットやデータ単位に分けてダウンロードする。

【００２３】

ツールがダウンロードされると、マルウェアは、感染したマシンおよび感染したマシンから到達可能なすべてのマシンからの有用なコンテンツの抽出を図る。図２に示すコード実行攻撃では、マルウェアは、高い権限を有するユーザを見つけるまで企業を検索し続けるかもしれない。代わりに、企業においてＳＭＢ／ＮＥＴＢＩＯＳ／ＣＩＦＳトラフィックのようなトラフィックを嗅ぎ分けて、権限を有するユーザの名前を見つけ、その認証情報にログインするかもしれない。代わりに、パスワード・クラッキングツールを使用したり、推測したパスワードの寄せ集めとパスワードファイルのコンテンツとを単純に比較したりするかもしれない。現在の処理能力であれば、８ワード長のパスワードは数時間で解読可能である。

【００２４】

権限を有するユーザの認証情報の抽出後、マルウェアは十分に強化されて、感染したマシンおよび感染したマシンから到達可能なすべてのマシンから有用なコンテンツを抽出するという作業に着手する。到達可能なマシンのリストには、データベースサーバ、コードリポジトリ、または価値ある設計書類を有するＣＡＤマシンが含まれる。価値あるコンテンツが抽出されると、マルウェアは、攻撃が最高潮に達する前に、データを暗号化または２１６においてコマンド・コントロールセンターにアップロードするかもしれない。データが暗号化された場合、悪意ある行為者は身代金要求を用いて標的に接触するかもしれない。

【００２５】

図３は、本開示の原理に従った、例示的なクライアント（ここでは解決クライアントと称す）が、ロードタイムにおいてマルウェア動作の検出に備えるために行う操作を示す。経路妥当性確認エンジンは解決クライアントの一部であり、マルウェアが起動した時点か

らマイクロ秒以内にマルウェア動作を確実に検出できる。解決クライアントはまず保全性（integrity）を検証し、その後アプリケーションのモデルを抽出するためにアプリケーションの各モジュールを分析する。アプリケーションのモデルはアプリケーション・マップ・データベースに格納される。アプリケーション・マップ・データベースは以下のテーブルを含んでもよい：コードテーブル、エクスポートテーブル、Ｖテーブル、アザーテーブル（他のテーブル）、基本ブロックテーブル、ソフトスポットテーブル、メモリ被演算子テーブル、遷移テーブル、逆アセンブリテーブル、重要ＯＳ関数テーブル。図３に示す実施形態では、アプリケーション・マップ・データベースは、解決クライアントからのリモートシステム上に配置される。他の実施形態では、アプリケーション・マップ・データベースは、アプリケーションが実行されているハードウェア上、または解決クライアントおよび解析エンジンの外部にあるハードウェア上に格納される。解決クライアントは、ストリーミングエンジンを使用して、解析システムのアプリケーション・マップ・データベースに格納すべきデータを送付するように、抽出されたアプリケーションモデルを複数の解決プロトコル・データユニット（ＰＤＵ）にパッケージ化する。ＰＤＵ構造を図４に示す。

【００２６】

３０２において解決クライアントがロードタイムにアプリケーションの処理を開始した後、３０４および３０６において、同じ操作がコンピュータアプリケーションの各モジュールのループにおいて行われる。アプリケーションの各モジュールがメモリへロードされる時、解決クライアントは、マシンコード逆アセンブラまたはバイトコード逆アセンブラなどの逆アセンブラを用いて所定のモジュールのすべての実行ファイルおよびライブラリを検査する。アプリケーションファイルのモジュールは、実行リンク可能フォーマット（ＥＬＦ）や共通オブジェクト・ファイル・フォーマット（ＣＯＦＦ）などの標準ファイル・フォーマットである。このフォーマットのアプリケーションのモジュールは、コードセクション、エクスポートされたデータセクション、ｖテーブル・セクション、および他の追加的セクションなどのセクションに整理される。アプリケーションの各モジュールがメモリへロードされる時、解決クライアントは、これらデータセクションをアプリケーションモデルの一部として抽出する。モジュールのコードセクションの境界およびアクセス属性は、３１４において、コードテーブル内のアプリケーション・マップ・データベースに送付（ディスパッチ）され格納される。このテーブルの各レコードは｛開始アドレス，終了アドレス｝という形式を有する。モジュールのコードセクション内の各基本ブロックの境界および指示数（指示の数）は、３３０において、基本ブロックテーブル（Basic Block Table）内のアプリケーション・マップ・データベースに送付され格納される。このテーブルの各レコードは｛開始アドレス，終了アドレス，および指示数｝という形式である。モジュールのエクスポートされたデータセクションの境界およびアクセス属性は、３１８において、エクスポートテーブル内のアプリケーション・マップ・データベースに格納される。このテーブルの各レコードは｛開始アドレス，終了アドレス｝形式である。モジュールのｖテーブルセクション（もしあれば）の境界およびアクセス属性は、３２２において、Ｖテーブル（V Table）内のアプリケーション・マップ・データベースに送付され格納される。このテーブルの各レコードは｛開始アドレス，終了アドレス｝形式である。モジュールのすべての他のセクションの境界およびアクセス属性は、３２６において、アザーテーブル内のアプリケーション・マップ・データベースに送付され格納される。このテーブルの各レコードは｛開始アドレス，終了アドレス，および保護属性｝という形式である。

【００２７】

各モジュールがメモリへロードされる時、解決クライアントはアプリケーションのモジュールから、他のメモリマッピングデータおよびソフトスポットデータをも抽出する。メモリマッピングデータは、メモリ割当て、メモリ割当て解除、およびメモリの重要セグメントへのメモリ書込みの指示を含む。ソフトスポットデータは、ループを実行する指示（例えば、ＲＥＰスタイル演算コード付き指示）を含む、大規模メモリバッファ（ソフトス

ポット）を操作するための指示を含む。ソフトスポット指示のアドレスおよび各メモリ書込みのサイズは、３３４において、ソフトスポットテーブル内のアプリケーション・マップ・データベースに送付され格納される。このテーブルの各レコードは｛アドレス，書込みサイズ｝という形式である。このアドレスおよび書込みサイズは、宛先がメモリ被演算子であるメモリ書込み指示のために格納されることになる。このデータは、３４０において、メモリ被演算子書込みテーブル内のアプリケーション・マップ・データベースに格納される。このテーブルの各レコードは｛発信元アドレス，メモリ書込みサイズ｝という形式である。

【００２８】

アプリケーションの各モジュールがメモリへロードされる時、解決クライアントはモジュールから遷移マッピングデータ（分岐転送または遷移データ）をも抽出する。遷移マッピングデータは、標的アドレスへの遷移指示をその時点において判定できる直接遷移マッピング、または標的アドレスへの遷移指示がランタイム依存性を有する間接メモリマッピングのためのものである。ランタイム依存性を有するため、遷移指示は、完全に判定されることがランタイムまで阻止される。間接遷移が生じる指示の完全な逆アセンブリは、３２４において、逆アセンブリテーブル内のアプリケーション・マップ・データベースに送付され格納される。抽出されたすべての遷移マッピングもまた、３２４および３３２において、遷移テーブル内のアプリケーション・マップ・データベースに送付され格納される。このテーブルの各レコードは｛発信元アドレス，宛先アドレス｝という形式である。また、３２０において、ランタイム以前に、オペレータはマップ遷移テーブルに手動で遷移マッピングデータを追加できる。マップ遷移テーブルに手動でレコードを追加する際、オペレータは、マルウェアによる遷移テーブルの変更を排除するために、２要素認証プロセスを用いた認証が必要である。

【００２９】

遷移マッピングは、マルウェアが起動した時点からマイクロ秒以内にマルウェア動作を確実に検出するために、経路妥当性確認エンジンの機能の中心をなす。遷移マッピングの概念は、ソースコードの検査により一層理解される。以下に示すサンプルソースコードでは、関数main()は、ライブラリで定義された関数printf()を呼び出すが、関数notCalled()は呼び出さない。コンパイラおよびリンカがこのコードを処理し、作成されたバイナリが検査された後、関数main()、printf()、およびnotCalled()の相互関係またはその欠如が保存される。関数main()は関数printf()に対して「遷移」を有すると言える。この遷移は{Address $_{SRC}$ -> Address $_{DST}$}で表され、Address $_{SRC}$は関数main()において関数printf()が呼び出された指示のアドレスを示し、Address $_{DST}$は関数printf()のアドレスを示す。発信および標的はシステムコールまたは例外ハンドラであってもよい。上記のようなレコードは、アプリケーションの遷移マップテーブルにおいて単一のレコードである。

【００３０】

【数１】

```
//C hello world example
#include <stdio.h>

int main()
{
printf("Hello world\n");
return 0;
}

int notCalled()
{
printf( “Feeling lonely !\n” );
return 0;
}
```

【００３１】
上記例はコンパイル言語であるＣ／Ｃ＋＋で記述されているが、ソースコードにおける遷移間の類似性は、インタープリタコードやＪＩＴコンパイルされたコードなどを含む、その他の言語で記述されたコードにおいても想定できる。Ｊａｖａ（登録商標）のような解釈型言語で記述した同じ例を以下に示す。

【００３２】

```
/*
Java Hello World example.
*/

public class HelloWorldExample {

   public static void main(String args[]){
      System.out.println("Hello World !");
  }

   public static void notCalled() {
      System.out.println("Feeling lonely !");
  }
```

}

【００３３】

アプリケーションの各モジュールがメモリへロードされる時、解決クライアントは３０８において、保全性のためにアプリケーションをチェックする。一実施形態においては、このチェックは、モジュールがロードされている時にコードのＭＤ５ハッシュなどのチェックサムを計算し、このチェックサムをチェックサム・データベースに格納された対応する既知のグッドチェックサムと比較することによって行う。代わりに、信頼できるチェックサム検査サービスを活用してもよい。これは、現在ロード中のモジュールのコードが未だマルウェアにより破損されていないことを保証する。解決クライアントは、保全性チェックが失敗した場合に、３１０において警報を発するよう構成されてもよい。

【００３４】

ロードタイムに、３１２および３１６において、アクセス許可および権限に影響する特定のＯＳ関数およびシステムコールが識別され、これらのアドレスが重要ＯＳ関数テーブルに送付され格納される。解決クライアントにより送付された特定のＯＳ関数およびシステムコールは、実行ファイルの実行経路に遠大な影響を及ぼすものである。これら管理用の重要ＯＳ関数およびシステムコールは、メモリセグメントのアクセス許可を変更し、アクセス権限を増大し、非実行ポリシーを変更し、構造化された例外ハンドラ保護を変更し、アドレス・スペース・レイアウト乱数化ポリシーを遮断し、メモリの割当ておよび割当て解除を行い、新しいプロセスを作成し、新しいスレッドを作成し、またはデータの暗号化および解読に関与する。

【００３５】

アプリケーションの各モジュールがメモリへロードされる時、解決クライアントは、ランタイムにデータを収集するために、アプリケーションのモジュールに挿入される指示に追加的にデータ記録手段を装備する。データ記録手段が装備されたコードは、動的バイナリ解析エンジンおよび／またはバイトコード・インストルメンテーション・エンジンを用いてアプリケーションのモジュールに挿入される。ソフトスポット指示には、３３８において、ループを実行する指示などのマルウェアが攻撃しそうな領域において、データ記録手段が装備されて、ランタイムにこれら領域で動作を追跡するためのデータを収集する。直接的・間接的遷移マッピング指示は、３２８において、モジュールにデータ記録手段が装備されて、ランタイムに遷移マッピングに関与する動作を追跡するためのデータを収集する。メモリ被演算子書込み指示には、３３６において、モジュールにデータ記録手段が装備されて、ランタイムにメモリ書込み動作に関するデータを収集する。自己修飾コードがある場合は、基本ブロックがランタイムにおいて変化するかもしれない。また、３１２および３１６において、アプリケーション内に指示にはデータ記録手段が装備されて、重要ＯＳ関数テーブルに格納されたＯＳ関数およびシステムコールに関与する動作のためのデータを収集する。

【００３６】

ロードタイムにおいて挿入されるデータ記録手段を装備した結果、ランタイムにおいて重要な情報が生成され分析のために収集される。遷移マッピングデータ関連のデータ記録手段の装備がアクセスされた時、解決クライアントは、スレッドＩＤ、現在の指示アドレス、宛先指示アドレス、および、随意に、各汎用レジスタに収納されたデータを収集する。指示が実行される前にソフトスポットのデータ記録手段の装備がアクセスされるため、解決クライアントは、適当なレジスタを介してスレッドＩＤおよびスタックの境界を入手する。ソフトスポットのデータ記録手段の装備が完了した時、解決クライアントは、スレッドＩＤ、およびこの書込み動作により更新されたメモリの領域の推定を可能とする数個の汎用レジスタを入手する。コールが実行される前に重要なＡＰＩまたはＯＳコールのデータ記録手段の装備がアクセスされる場合、解決クライアントはスレッドＩＤ、ＡＰＩ名またはシステムコール番号、および入力パラメータを入手する。コールが実行された後に重要なＡＰＩまたはＯＳコールのデータ記録手段がアクセスされる場合、解決クライアントはスレッドＩＤ、ＡＰＩ名またはシステムコール番号、および戻り値を入手する。メモ

リの割当てまたは割当て解除を行うＯＳ関数またはシステムコールにおけるデータ記録手段の装備は、アプリケーションが作成したかもしれない種々のヒープに現在関与しているメモリの領域の追跡を援助する。このメモリ・エンベロープは、マルウェアがヒープにおけるコントロール構造をオーバーランさせようとしているか否かを見つけるために、ランタイムにおいて間接的メモリ書込みの標的を追跡するのに活用される。また、キャッシュを用いて基本ブロックの境界を追跡することにより、解析エンジンは基本ブロックが変化したか否かを判定できる。この判定が肯定である場合、モデルデータベース内の基本ブロックテーブルが更新される。

【００３７】

この例示的な実施形態では、解決クライアントは入手した情報をストリーミングエンジンへ送付して、解析エンジンに伝送すべくＰＤＵとしてパッケージ化する。ストリーミングエンジンは、パイプまたはローカル・プロシージャ・コールなどの極めて低いオーバーヘッドＯＳアーティファクトを用いて、種々のデータ記録手段装備により生成されたデータを別のプロセスに移動させる。これにより、データ記録手段が装備されたプロセスはその通常の動作を継続できる。この例示的な実施形態では、ストリーミングエンジンは、適切な標準ベースのトランスポートプロトコルを用いて、データ記録手段から収集した情報を、解析エンジンへのさらなる伝送のために解決ＰＤＵにおいてパッケージ化してもよい。一実施形態では、トランスポートプロトコルはＴＣＰ／ＩＰでもよい。他の実施形態では、トランスポートプロトコルはＵＤＰでもよい。さらに他の実施形態では、トランスポートプロトコルは、パイプやローカル・プロシージャ・コールなどの共有メモリ技術を用いることを含んでもよい。

【００３８】

図４に解決ＰＤＵを示す。解決クライアントおよび解析エンジンは互いに効果的に作業するために、解決ＰＤＵを用いて相互通信する。解決ＰＤＵは、特に解決クライアントにより、抽出されたアプリケーションモデルおよび／または収集されたランタイムデータを解析エンジンへの伝送用にパッケージ化するために用いられる。解決ＰＤＵは、解決クライアントと解析エンジンとの間で送信される各種情報のためのフィールドを含む。解決ＰＤＵは、アプリケーション提供データセクション、ＨＷ／ＣＡＥ生成セクション、およびコンテンツ解析エンジンまたは生データ（ローデータ）セクションに分割される。

【００３９】

アプリケーション提供データセクションは、種々のレジスタからのデータと、このセクションの種々のフィールドに配置された発信元アドレスおよび標的アドレスとを含む。プロトコルバージョンは解決ＰＤＵ４０２のバージョン番号を含む。解決プロトコルバージョンが経時変化する場合、発信元および宛先は互いの通信を継続できる必要がある。この８ビットフィールドは、発信元エンティティにより生成された際の解決パケットのバージョン番号を表す。現在未使用の予約されたフィールド４０４はプロトコルバージョンフィールドに続く。

【００４０】

アプリケーション提供データセクションにおける次のフィールドはメッセージ発信元／宛先識別子４０６、４０８および４１０である。これらは図７に示す解析エンジンのインフラストラクチャ内でのトラフィックの交換に用いられる。図７に示す種々のエンティティは、時々トラフィックを交換する。これら装置のすべてがＩＰアドレスを有するかまたは必要とするわけではないので、２つの（ハードウェアおよびホスト）照会ルータ・エンジンは、内部でトラフィックをルーティングするためにメッセージ発信元および宛先フィールドを用いる。いくつかのメッセージは、解析エンジンのエンティティに辿り着くためにネットワークを横切る必要がある。そのために、エンティティには以下のＩＤが割り当てられる。所定の解析エンジン機器は２つ以上のアクセラレータ・カードを有してもよい。各カードは固有のＩＰアドレスを有する。したがって、種々のエンティティは固有のＩＤを有する。上述したインフラストラクチャは２つ以上のアプリケーションを実行してもよい。各アプリケーションサーバは固有のＩＰアドレスを有することになるので、対応す

る解決クライアント側エンティティもまた固有のＩＤを有する。
【００４１】
＜解決クライアント側エンティティ＞
１．解決ＧＵＩ（Resolve GUI）
２．インストルメンテーションおよび解析エンジン（Instrumentation and Analysis Engine）
３．クライアント・メッセージ・ルータ（Client Message Router）
４．ストリーミング・エンジン（Streaming Engine）
５．クライアント側デーモン（Client Side Daemon）
６．ＣＬＩエンジン（CLI Engine）
７．クライアント監視（Client Watchdog）
８．クライアント圧縮ブロック（Client Compression Block）
９．クライアント・ｉワープ・イーサネット（登録商標）・ドライバ（１００Ｍｂ／１Ｇｂ／１０Ｇｂ）（Client iWarp Ethernet Driver (100 Mb/1Gb/10Gb)）
＜ＰＣＩ毎・カード・エンティティ（先頭アドレス＝２０＋ｎ×２０）＞
２０．セキュラライザＴＯＥブロック（Securalyzer TOE block）
２１．セキュラライザＰＣＩブリッジ（Securalyzer PCI Bridge）
２２．解凍ブロック（Decompression Block）
２３．メッセージ検査ブロック（Message Verification Block）
２４．パケット・ハッシング・ブロック（Packet Hashing Block）
２５．タイムスタンプ・ブロック（Time-Stamping Block）
２６．メッセージ・タイムアウト・タイマ・ブロック（Message Timeout Timer Block）
２７．統計カウンタ・ブロック（Statistics Counter Block）
２８．セキュラライザ照会ルータ・エンジン（Securalyzer Query Router Engine）
２９．セキュラライザ・アシスト（Securalyzer Assist）
＜セキュラライザ・ホスト・エンティティ＞
２００．セキュラライザＰＣＩｅドライバ（Securalyzer PCIe Driver）
２０１．ホスト・ルーティング・エンジン（Host Routing Engine）
２０２．コンテンツ解析エンジン（Content Analysis Engine）
２０３．ログ・マネージャ（Log Manager）
２０４．デーモン（Daemon）
２０５．ウェブ・サービス・エンジン（Web Service Engine）
２０６．監視（Watchdog）
２０７．ＩＰＣメッセージング・バス（IPC Messaging Bus）
２０８．コンフィギュレーション・データベース（Configuration Database）
２０９．ログ・データベース（Log Database）
＜ＳＩＥＭコネクタ＞
２２０．ＳＩＥＭコネクタ１－Ｖｉｒｓｅｃダッシュボード（SIEM Connector 1－Virsec Dashboard）
２２１．ＳＩＥＭコネクタ２－ＨＰ　ＡｒｃＳｉｇｈｔ（SIEM connector 2－HP ArcSight）
２２２．ＳＩＥＭコネクタ３－ＩＢＭ　ＱＲａｄａｒ（SIEM connector 3－IBM QRadar）
２２３．ＳＩＥＭコネクタ４－Ａｌｉｅｎ　Ｖａｕｌｔ　ＵＳＭ（SIEM connector 4－Alien Vault USM）
＜セキュラライザ・インフラストラクチャ・エンティティ＞
２３０．Ｖｉｒｓｅｃダッシュボード（Virsec dashboard）
２３１．ＳＭＴＰサーバ（SMTP Server）
２３２．ＬＤＡＰサーバ（LDAP Server）

２３３．ＳＭＳサーバ（SMS Server）
２３４．資格サーバ（Entitlement Server）
２３５．データベース・バックアップ・サーバ（Database Backup Server）
２３６．ＯＴＰクライアント（OTP Client）
２３７．ＯＴＰサーバ（OTP Server）
２３８．チェックサム・サーバ（Checksum Server）
２３９．チケッティング・サーバ（Ticketing Server）
２４０．Ｖｉｒｓｅｃルール・サーバ（Virsec Rules Server）
２４１．Ｖｉｒｓｅｃ更新サーバ（Virsec Update Server）
＜オール・ユーザ・アプリケーション＞
２５５．ユーザアプリケーション‐照会を発行するアプリケーションを識別するためにアプリケーションＰＩＤが使用される。

【００４２】
アプリケーション提供データセクションの他のフィールドには、送信されているデータのタイプを示すメッセージ・タイプ・フィールド４１２がある。最上位レベルで、種々のローカルな解決クライアント側エンティティ間、解析エンジン機器側エンティティ間、およびクライアント側エンティティと機器側エンティティの間を流れる３つの異なるタイプのメッセージがある。さらに、ネットワークを移動する必要のあるメッセージは、ＯＳＩモデルおよび他のプロトコルに適合する必要がある。

【００４３】
アプリケーション提供データセクションにおける次のフィールドは、パケットのシーケンス識別子を含むパケットシーケンス番号フィールド４１４である。ストリーミングエンジンは、損失パケットのエラー回復が可能である。そのために、ストリーミングエンジンは一意にパケットを識別する必要がある。インクリメントする符号付の６４ビットパケットシーケンス番号は、ストリーミングエンジンにより挿入され、残りの解析エンジンのインフラストラクチャを単に通過する。シーケンス番号が６４ビット境界で完了すると、０からリスタートする。ハートビート、ログメッセージ等の非アプリケーションパケットの場合は、パケットシーケンス番号は－１であってもよい。

【００４４】
アプリケーション提供データセクションは、暗号化のために使用されるカナリア（canary）を含む解決カナリア・メッセージフィールド４２２をも含む。解決クライアントおよび解析エンジンは、アプリケーション・ラウンチタイム、ＰＩＤ、ライセンス・ストリング、および許可されたユーザ名などの一般的な情報（ただし、新しい性質の情報）からのカナリア計算方法を認識している。

【００４５】
アプリケーション提供データセクションはさらに、すべてのメッセージにおいて使用される汎用フィールドを含む。アプリケーション発信元指示アドレス４５８、アプリケーション宛先指示アドレス４１６、メモリ開始アドレスポインタ４１８、メモリ終了アドレスポインタ４２０、アプリケーションＰＩＤ４２４、スレッドＩＤ４２６、解析エンジン到着タイムスタンプ４２８、および解析エンジン出発タイムスタンプ４３０など、汎用アプリケーションデータを保持するフィールドである。

【００４６】
解決ＰＤＵはＨＷ／ＣＡＥ生成セクションをも含む。解析を容易にしかつ固定の時間制約を守るために、解析エンジンは発信元・宛先アドレスフィールドをハッシュし、処理の前に解決ＰＤＵを更新する。解決ＰＤＵのＨＷ／ＣＡＥ生成セクションは、後に使用するハッシュデータが配置されるものである。このセクションは、ハッシュ・アプリケーション発信元指示アドレス４３２、ハッシュ・アプリケーション宛先指示アドレス４３４、ハッシュ・メモリ開始アドレス４３６、およびハッシュ・メモリ終了アドレス４３８のフィールドを含む。ＨＷ／ＣＡＥ生成セクションはさらに、解決カナリア４４２に関する他のフィールドを含む。これらフィールドは、ハードコード化コンテンツ開始マジックヘッダ

、ＡＰＩ名マジックヘッダ、コール・コンテキスト・マジックヘッダ、およびコール生データ・マジックヘッダであって、すべての解決ＰＤＵパケットに存在する。

【００４７】

ＨＷ／ＣＡＥ生成セクションは、他のコンフィギュレーションおよびエラーのデータを識別するためのフィールド４４０をも含む。このフィールド４４０は、結果、コンフィギュレーションビット、オペレーティングモード、エラーコード、およびオペレーティングモードのデータを含む。このフィールドの結果部分は、遷移プレイバック、コードレイアウト、メモリ（スタックまたはヒープ）オーバーラン、およびディープ検査などの異なる解析エンジン照会に対してブーリアン結果を返送するためにセグメント化されている。このフィールドのコンフィギュレーションビット部分は、圧縮フラグ、デモフラグ、または共通配置フラグが設定された時を示す。このフィールドにおけるフラグの存在は、パケットを圧縮モードで返送するべきか否かを解析エンジンに対して示す。デモフラグは、システムに有効なライセンスがないためにシステムがデモモードであることを示す。このモードでは、ログおよびイベントの全体を入手できない。共通配置フラグは、アプリケーションが解析エンジンにおいて実行中であることを示し、これによりホスト照会ルータ・エンジンはアプリケーションへ返送されるべきパケットの宛先を判定できる。このフラグがセットされている場合、パケットはＰＣＩブリッジを介して送信され、そうでない場合は、パケットはＰＣＩカード上のイーサネット（登録商標）インターフェースを介して送信される。このフィールドのオペレーティングモード部分は、システムがパラノイド（最高セキュリティ）モード、モニタモード、学習モードのいずれにあるかを示す。これらモードについては、後に詳述する。最後に、このフィールドのエラーコード部分はシステムにおけるエラーを示す。エラーコードの最初の８ビットはメッセージ発信元に対応することになる。残りの１２ビットは各サブシステムにより報告される実際のエラーに対応することになる。

【００４８】

解決ＰＤＵは、また、コンテンツ解析エンジンまたは生データを含む。ＯＳライブラリコールおよびシステムコールの、引き数や戻り値などのすべての可変データは、解決ＰＤＵのこのセクションに配置される。このセクションのデータは、アプリケーションから収集されたデータのコンテンツを含み、主にコンテンツ解析エンジンに向けられる。このセクションは、サイズ可変なＡＰＩ名またはＡＰＩ番号４４４、コール・コンテンツ・マジックヘッダ４４６、サイズ可変コール・コンテンツ４５０、コール・ローデータ・マジックヘッダ４５２、サイズ可変ローデータ・コンテンツ４５６、および２つのリザーブド４４８および４５４のフィールドを含む。さらに、これらフィールドは管理メッセージに対してオーバーロードされる。

【００４９】

図５は、ランタイムにセキュリティイベントを検出するために解析エンジンが用いるプロセスを示す。この例示的な実施形態において、解析エンジンは、解決クライアントと同一または異なるハードウェア上で実行される別プロセスである。データ解析部分からデータ収集を分離することにより、データ記録手段を装備するプロセスにより生成された情報の解析に伴う処理のオーバーヘッドが実質的に削減される。これにより、解析エンジンは、同時に複数のアプリケーションのランタイム解析を行うことができる。

【００５０】

ランタイム中、コードにデータ記録手段が装備されることで、スレッドＩＤ、スタック開始およびベースアドレス、汎用レジスタ、発信元アドレス、および宛先アドレスなどの適切なコンテキストを検査する機会が得られる。コンテンツを検査するための一連の動作に特定の順序はなく、入替え可能であり、解析エンジンの複数のスレッドにより並列に行える。図５に示すプロセスを用いて、安全なエンジンがアプリケーションにおける各スレッドの個々のイベントを生成することになり、これらイベントは特定かつ個別のイベントチェーン状態マシン上で収集でき、これにより攻撃の進行をリアルタイムで追跡できる。解析エンジンにより生成されるイベントは、ＳＹＳＬＯＧなどの標準ベースのイベントフ

ォーマットに従う。これにより、標準ベースのイベント管理システムは、解決クライアントにより生成されたイベントを直接、またはＳＹＳＬＯＧをコモン・イベント・フォーマット（ＣＥＦ）などの標準標的フォーマットに変換するコネクタを介して使用できる。解析エンジンは、以前に格納したテーブル付アプリケーションマップにアクセスできるので、実行された遷移指示がそのコンテキストを報告した時、解析エンジンは、標的アドレスが正当な宛先か否かを示すデータを含む抽出されたアプリケーションモデルにアクセスできる。

【００５１】

５０２においてプロセスが開始した後、５０４において、解決クライアントからの解決ＰＤＵパケットで新たなランタイム情報が受信される。５０８においてこのパケットをバッファに保存する前に、解析エンジンは５０６において、パケットに含まれるカナリアおよびタイムスタンプを検証する。解決ＰＤＵが未だバッファされている間に、解析エンジンは５１０において、アドレスフィールドをハッシュし、ハッシュしたデータを解決ＰＤＵのＨＷ／ＣＡＥセクションに入れる。５１２において、処理のためにパケットをパケットバッファリングから抜かれる。パケットの処理が完了すると、５５４においてプロセスは次のパケットを待つ。

【００５２】

解析エンジニアにより使用されるプロセスは、遷移マッピングデータを調査する。５２０において遷移指示により報告されたランタイム情報が直接遷移からのものであり、かつ５２８において標的アドレスが遷移マップテーブル中に見つからない場合、解析エンジンは５４４においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。５２２において遷移タイプ指示により報告されたランタイム情報が間接遷移からのものであり、かつ５３０において標的アドレスがコードテーブル中にあり、かつ５４８において標的アドレスが基本ブロックの中央部分にある場合、解析エンジンは５５０においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。５２２において遷移タイプ指示により報告されたランタイム情報が間接遷移からのものであり、かつ５３０および５４８において標的アドレスがヒープメモリの領域にある場合、解析エンジンは５５０においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。５２２において遷移タイプ指示により報告されたランタイム情報が間接遷移からのものであり、かつ５５２において標的アドレスがメモリの非コード・非インポートテーブル領域にある場合、解析エンジンは５５６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。

【００５３】

５１６において、解析エンジンにより使用されるプロセスはメモリ書込みデータを調査する。報告されたランタイム情報がメモリ書込み指示からのもので、かつ５２４において書込み標的アドレスがＶテーブルのメモリ領域内にある場合、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。報告されたランタイム情報がメモリ書込み指示からのもので、かつ５２４において書込み標的アドレスがエクスポートテーブルのメモリ領域内にある場合、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。報告されたランタイム情報がメモリ書込み指示からのもので、かつ５３８において書込み標的アドレスがヒープメモリ領域の標的コントロールセクション内にある場合、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。報告されたランタイム情報がメモリ書込み指示からのもので、かつ５３２において、書込み標的アドレスがスタックメモリ領域の標的コントロールセクションにある場合、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。

【００５４】

５１４において、解析エンジンにより使用されるプロセスはソフトスポットデータを調査する。指示がソフトスポット指示であって、かつ５２４において書込み標的アドレスが

Ｖテーブルのメモリ領域にある場合は、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。指示がソフトスポット指示であって、かつ５２４において書込み標的アドレスがエクスポートテーブルのメモリ領域にある場合は、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。指示がソフトスポット指示であって、かつ５３８において書込み標的アドレスがヒープメモリ領域の標的コントロールセクションにある場合は、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。指示がソフトスポット指示であって、かつ５３２において書込み標的アドレスがベースポインタ下方の記憶場所におけるスタックのコントロール領域に格納された指定ポインタを上書きする場合は、解析エンジンは５３６においてイベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。

【００５５】

５１８において、解析エンジンにより使用されるプロセスはＯＳ関数およびシステムコールを調査する。５４２における無効なＯＳ関数またはシステムコール動作が原因で、５２６において重要なＯＳ関数またはシステムコールのエントリポイントのデータ記録手段（インストルメンテーション）が呼び出された場合、解析エンジンは５４４において、その重要ＯＳ関数テーブルにおける動作が発生した時に、イベントを生成する。このイベントは、それが発生したスレッドの詳細を含む。この生成されたイベントにより、メモリアクセス許可を変更する要求されたコードは無効を宣言され、権限レベルを変更するように要求されたコードは潜在的に無効であると宣言され、ＷＡＮベースユーザにサービスを提供するスレッド上で発生する権限増大は高いレベルに上昇され、非実行ポリシーの変更は無効を宣言され、アドレス空間配置の乱数化（ＡＳＬＲ）ポリシーへ変更は無効を宣言され、安全な例外処理（ＳＥＨ）ポリシーの変更は無効を宣言され、管理上の指定関数への要求は、その特定の関数要求のイベントに従って処理されず、または管理上の指定システムコールへの要求は、その特定のシステムコールのイベントに従って処理されない。さらに、５３４におけるメモリ動作をヒープするための５２６における重要システムコール動作は、５４０においてヒープ・メモリ・エンベロープを更新するために用いられる。メモリのうち割当てに関連する領域はメモリ・エンベロープサイズを増加させ、その一方、割当て解除の領域は標的領域を解放する。

【００５６】

図６は、検出されたセキュリティイベントの追跡に使用されるイベントチェーン状態マシンを示す。経路妥当性確認エンジンはイベントを生成し、図示のイベントチェーン状態マシンに従ってイベントの状態を追跡する。イベントチェーン状態マシンは、これらイベントを追跡するために遷移プレイブックデータベースを使用する。新しいスレッドが開始すると、６０２においてイベントチェーン状態マシンは開始状態に初期化される。イベント管理エンジンはイベントに一連番号を振って、これにより１つのスレッド上のイベントのセットが、適切なスレッド用に状態マシン上で更新される。スレッドで遷移が発生すると、状態マシンは標的アドレスを記録するが、開始状態に留まる。経路妥当性確認エンジンが中重大度または高重大度のイベントを生成すると、６０４において、イベントチェーン状態マシンは次の状態であるE_i状態に進む。この新しい状態は、遷移が発生すると、その遷移の標的アドレスを記録し続ける。中重大度または高重大度のイベントごとに新しい状態に更新し遷移イベントを記録するという状態マシンのプロセスは、６１２においてスレッドが終了状態に達するまで、６０６、６０８、および６１０において継続する。最高重大度レベルのイベントが起こるとセキュリティ分析者に通知され、攻撃が発生するとイベントチェーン状態マシンはフォレンジクスを捕捉する。

【００５７】

システムのモードによって、様々な救済措置がイベントに応答して採用され得る。採用された救済措置はリアルタイムで行われる。ある救済措置は、アプリケーションを遮断することからなる。他の救済措置は、脅威が出現したスレッドの１つまたは複数のソケットを解放または解除し、脅威が出現したスレッドを終了し、および／またはその脅威の原因

となるユーザをブラックリストに載せる。他の救済措置として、アプリケーションサーバの１つまたは複数のソケットに関連するすべてのソケットを非ブロック化してもよい。これにより、その時接続されているすべてのユーザがブロックされることとなる。さらに他の救済措置として、攻撃を無視してもよい。この場合、分析者はその攻撃を重大なものと考えていないことによる。所定のイベントに対して提案された最適な救済措置はあらかじめプログラムされているため、セキュリティ分析者が解析エンジンの自動モード操作を選択していた場合、救済措置が自動的に行われる。

【００５８】

解析エンジンの経路妥当性確認エンジンは、モニタモード、パラノイドモード、学習モードの３つのモードで実行される。これらモードの違いは、アプリケーションにおける所定のスレッド上で受信された１つまたは複数のイベントに伴う救済措置を、いつどのように実施するかの違いである。モニタモードでは、ランタイム情報が到達し解析エンジンにより解析される時、指定されたセキュリティ分析者に向けられた通知が生成される。介入して最適な救済措置を選択するのはセキュリティ分析者の仕事である。セキュリティ分析者は、スレッドが終了状態に達する前であっても、予めプログラムされた「無視する」タイプの救済措置をより高度な衝撃救済措置に変更することを決定してもよい。救済措置を確定すると、解析エンジンは、その救済措置を実装する。この際、企業における適切な権威は、所定のスレッドに対しては提案された救済措置を適用しない。

【００５９】

パラノイドモードでは、プログラムされた（デフォルトのまたはユーザ設定による）救済措置が、セキュリティ分析者の介入なしに自動的に実行される。いずれのモードであっても、救済措置実行の準備が整うと、解析エンジンはどの救済措置が実行されるかを解決クライアントに知らせる。解決クライアントはアプリケーション上でこの救済措置に関連した措置を行う。救済措置が完了すると、解決クライアントは解析エンジンへ確認メッセージを返信する。その確認を受信すると、解析エンジンはセキュリティ分析者の更新を含む間接的援助動作を行う。

【００６０】

学習モードでは、解析エンジンは全てのイベントおよび救済措置を無視する。このモードでは、アプリケーションは無垢の環境で動作し、すべてのイベントおよびイベントチェーンを記録する。セキュリティ分析者はこの情報を使用して、いつイベントが発生すべきで、どのような救済措置がそのイベントに伴って行われるべきかについての判断基準を形成する。

【００６１】

図７は、解決クライアントおよび解析エンジンインフラストラクチャの一例の上位レベルブロック図を示す。このインフラストラクチャは、スマートフォン、タブレット、ラップトップ、デスクトップからハイエンドサーバまでの、計算装置を含む種々のハードウェア上で構成されてもよい。図に示すように、アプリケーション性能を高めるために、解決クライアントにより行われるデータ収集は解析エンジンにより行われる解析から分離されてもよい。このインフラストラクチャは、ハッカによる、マルウェア攻撃に対する防御の破壊を防止する高い可用性を提供する。解決クライアントは、ロードタイムおよびランタイムデータを収集するためにアプリケーションと相互作用する。アプリケーション７０１のインフラストラクチャは、プロセスメモリ７０３、第三者ライブラリ７０４、カーネルサービス７０６、および指示パイプライン７０７を備える。解決クライアント７０２のインフラストラクチャは、インストルメンテーションおよび解析エンジン（ＩＡＥ）７０５、グラフィカル・ユーザ・インタフェース（ＧＵＩ）７１１、クライアントデーモン７０８、コンフィギュレーション・データベース７０９、およびストリーミングおよび圧縮エンジン７１０、および中央処理装置（ＣＰＵ）７３６を備える。アプリケーション７０１のローカルまたはリモートのユーザ７０２は、キーボード、マウス、または同様の入出力装置を介して、あるいは、パイプ、共用メモリ、またはソケットを用いて設定され得る通信チャネルを介したネットワークを通じて、アプリケーションと相互作用する。これに応

答して、アプリケーションプロセス７０３は、適切な指示のセットを実行用の指示パイプライン７０７に送付する。アプリケーションは、ｌｉｂｃ．ｓｏ（リナックス（登録商標））またはｍｓｖｃｒｔｘｘ．ｄｌｌ（ウィンドウズ（登録商標））などの、自身のまたは第三者のライブラリ７０４を活用してもよい。これらライブラリからの機能が呼び出されると、これらライブラリからの適切な指示もまた実行用の指示パイプライン７０７に挿入される。加えて、アプリケーションは、カーネル７０６からの、メモリやファイル入出力などのシステムリソースを活用してもよい。アプリケーション、ライブラリおよびカーネルからのこれら一連の指示は時間順のシーケンスにまとめられ、所与のユーザが希望するアプリケーション機能を配信する。

【００６２】

アプリケーションのコードのメモリへのロードが開始すると、ＩＡＥ７０５はいくつかの異なるロードタイム動作を行う。すべてのモジュールがロードされると、データ記録手段が装備されたアプリケーションの指示はランタイムデータを生成する。クライアントデーモン７０８は、コンフィギュレーション・データベース７０９から１つ以上のコンフィギュレーションファイルを読みだすことにより、ＣＰＵ７３６においてインストルメンテーションおよび解析エンジン７０５、ストリーミングエンジン７１０およびＧＵＩ７１１のプロセスを初期化する。クライアントデーモン７０８は、また、ＩＡＥ、ストリーミングエンジン、ＧＵＩおよび解析エンジンと、クライアントデーモン７０８との間の相互通信パイプを初期化する。クライアントデーモンはまた、それ自体を含む解決クライアントプロセスのいずれかが無応答または動作不能になった時、そのプロセスが再生成されることを保証する。これは、解決クライアントが高可用性の企業向け製品であることを保証する。

【００６３】

インストルメンテーションおよび解析エンジンは、アプリケーションから収集されたロードおよびランタイムデータをストリーミングエンジンに送る。ストリーミングエンジンは、解決クライアントからのローデータを解決ＰＤＵとしてパッケージ化する。ストリーミングエンジンは、解決ＰＤＵを、高帯域幅で低レイテンシである通信チャネル７１２を介して解析エンジン７１１に送る。クライアントとアナライザが同じマシンに位置している場合、このチャネルはメモリバスである。これらエンティティが、異なるハードウェア上であるが同じ物理的近傍性を有するハードウェアに位置する場合、このチャネルはイーサネット（登録商標）またはファイバーベース・トランスポートである。これにより、エンティティ間にリモート接続が設定されて、インターネットを通じてロードおよびランタイムデータが転送される。

【００６４】

解析エンジンのインフラストラクチャは、ネットワーク・インターフェース・カード（ＮＩＣ）７１３、パケットプール７１４、タイムスタンプエンジン７１５、プロセッサファブリック７１６、ハッシングエンジン７１７、ＴＣＡＭエンジン７１８、アプリケーション・マップ・データベース７１９、およびスレッド・コンテキスト・データベース７２０を備える。解析エンジンのインフラストラクチャはさらに、コンテンツ解析エンジン７２１、イベントおよびイベントチェーン７２２、イベント管理エンジン７２３、イベントログ７２４、機器デーモン７２５、解析エンジン・コンフィギュレーション・データベース７２６、ネットワークインターフェース７２７、ダッシュボード７２８、ＳＭＳ／ＳＭＴＰサーバ７２９、ＯＴＰサーバ７３０、拡張クライアント７３１、ソフトウェア拡張サーバ７３２、ソフトウェアイメージ７３３、イベント更新クライアント７３４、およびイベント拡張サーバ７３５を備える。

【００６５】

プロトコルヘッダーを有する解決ＰＤＵは、ネットワーク・インターフェース・カード７３１によって捕捉される。ネットワーク・インターフェース・カード７３１から解決ＰＤＵは取り出されてパケットプール７１４に入れられる。解決ＰＤＵにおけるタイムスタンプフィールドはタイムスタンプエンジン７１５により充填される。これは、パケットが

異常に長い時間パケットプールバッファにスタックされていないことを確認するのに役立つ。

【００６６】

プロセッサファブリック７１６はパケットをパケットバッファから取り出し、アドレスフィールドがハッシュされてこのパケット内の適切な位置に配置される。この動作はハッシングエンジン７１７により行われる。プロセッサファブリックは、パケットバッファからパケットをそれらが到着した順序での取出しを開始する。ロードタイム段階からの情報を有するパケットは、関連データが抽出されてアプリケーション・マップ・データベース７１９に格納されるように処理される。ランタイム段階からの情報を有するパケットは、図５に従って処理される。解析エンジンの効率は、プロセッサファブリックにおけるプロセッサの数に基づいて向上または低下する。

【００６７】

遷移標的データは、スレッドごとのテーブルを有するスレッド・コンテキスト・データベースに格納されている。プロセッサファブリックは、また、遷移およびメモリ領域検索を行うためにＴＣＡＭエンジン７１８を活用する。プロセッサファブリックはハッシュを用いてルックアップを行うため、これに要する実時間は予測可能で、極めて短い。ファブリックにおけるプロセッサの数を注意深く選択することにより、パケット当たりのスループットを適宜変更できる。

【００６８】

解析エンジンが検索を行うと、解析エンジンは時々、重要な／管理上の関数またはシステムコールの無効な遷移や無効な動作、または望ましくない位置におけるメモリ書込みを発見する。いずれの場合も、解析エンジンは、イベントおよびイベントチェーンのデータベース７２２に格納されたポリシーによって記述された、プログラムされた重大度のイベントをイベント管理エンジンに送付する。生イベントログは、イベントログデータベース７２４に格納される。ダッシュボードはイベントログにもアクセスでき、アプリケーションステータスを表示できる。

【００６９】

救済措置は、また、イベントおよびイベントチェーンのデータベース７２２内の各イベントに関連付けられている。ユーザは、極端な一例であるイベントを無視するという措置から、その対極の一例であるスレッドを終了するという措置までの範囲において、救済措置を設定できる。イベント更新クライアント７３４およびイベント拡張サーバ７３５を用いて、推奨救済措置を分析者に推奨できる。上記推奨動作を変更するために、分析者はダッシュボード７２８を使用できる。ダッシュボードは、監視される各アプリケーションの状態を表示するＧＵＩインターフェースを提供し、これによりセキュリティ分析者はアプリケーション上でアプリケーションの開始や停止などの制御を行うことができる。イベントが生成されると、イベントチェーンは通常の状態から後続の状態へと進行する。新しい状態に対応した救済措置が採用されてもよい。救済措置が非無視措置である場合、ＳＭＳまたはＳＭＴＰサーバ７２９を用いてセキュリティ分析者へ通知が送信される。セキュリティ分析者のＳＭＳ／ＳＭＴＰアドレスはＬＤＡＰまたは他のディレクトリプロトコルにより判定される。ダッシュボードからアプリケーションを開始または停止させるプロセスは高い権限を必要とするため、セキュリティ分析者はＯＴＰサーバ７３０を用いて認証する必要がある。

【００７０】

新しいイベントが生成されて、重大度および分析者へ推奨する救済措置とともにイベントおよびイベントチェーンのデータベース７２２にリンクされてもよい。これにより、ある装置における新しい攻撃に対する固有のイベントおよびイベントチェーンを他の装置へ送付できる。このために、すべての新しいイベントおよびイベントチェーンはイベント拡張サーバ７３５にロードされる。イベント更新クライアント７３４は、新しいイベントおよびイベントチェーンを検索して取り出すために、定期的にイベント拡張サーバに接続し認証を行う。イベント更新クライアントは、これら新しいイベントおよびイベントチェー

ンをイベントおよびイベントチェーンのデータベース７２２にロードする。コンテンツ解析エンジン７２１は、新しいイベントチェーンに包含された新しい攻撃に対してアプリケーション追跡を開始できる。

【００７１】

クライアントデーモンの場合と同様に、機器デーモン７２５は、解析エンジン上で実行される種々のプロセスの開始を担当する。このため、機器デーモン７２５は、解析エンジン・コンフィギュレーション・データベース７２６からコンフィギュレーション情報を読み出す必要がある。デーモンは、また、解析エンジンにおけるすべてのプロセスのためのハートビートポーリングを担当する。これは、解析エンジンにおけるすべての装置が常に最高の動作状態であることを保証する。３つの連続するハートビートが欠けることは、標的プロセスが応答していないことを示唆する。任意のプロセスが早まって終了すると、デーモンはそれ自体を含むそのプロセスを再開させる。

【００７２】

ソフトウェアは時々、機器ホスト、解析エンジン、またはクライアントにおいて、ソフトウェア内のエラーを修復するためにアップグレード（拡張）される。そのために、拡張クライアント７３１は、最新のソフトウェアが入手可能なソフトウェア拡張サーバ７３２を常に確認する。クライアントが、解析エンジンまたはクライアントのエンティティが古い画像を実行していることを発見した場合は、クライアントは、分析者にソフトウェア拡張サーバ７３２を介してこの古い画像を新しい画像に更新させる。新しい画像はシステム画像７３３として束ねられる。これにより、機器またはホストに、検査済みの互換性のある画像を提供できる。解析エンジンまたは解決クライアントにおけるサブシステムの画像の１つが、システム画像における同じ構成要素の画像と適合しない場合、すべての画像は以前の既知の良好なシステム画像にロールされることになる。

【００７３】

図８は、本発明の実施形態が実施されるコンピュータネットワークまたは同様のデジタル処理環境を示す。

【００７４】

１つまたは複数のクライアントコンピュータ／装置５０および１つまたは複数のサーバコンピュータ６０は、アプリケーションプログラム等を実行する処理装置、格納装置、および入出力装置を提供する。１つまたは複数のクライアントコンピュータ／装置５０はまた、通信ネットワーク７０を介して、他のクライアント装置／プロセス５０および１つまたは複数のサーバコンピュータ６０を含む他の計算装置と接続できる。通信ネットワーク７０は、互いに通信するために個々のプロトコル（ＴＣＰ／ＩＰ、ブルートゥース（登録商標）等）を現在使用している、リモートアクセスネットワーク、グローバルネットワーク（例えばインターネット）、ワールドワイドなコンピュータの集合、ローカルエリアまたはワイドエリアネットワーク、およびゲートウェイの一部であってもよい。他の電子機器／コンピュータ・ネットワーク・アーキテクチャも適している。

【００７５】

図９は、図８に示すコンピュータシステムにおけるコンピュータ（例えば、クライアントプロッセッサ／装置５０またはサーバコンピュータ６０）の例示的な内部構造を示す図である。各コンピュータ５０，６０はシステムバス７９を含み、１つのバスはコンピュータまたは処理システムの構成要素間でのデータ転送に使用されるハードウェアラインのセットである。システムバス７９は、基本的に共用コンジットであって、コンピュータシステムの異なる要素（例えば、プロセッサ、ディスク記憶装置、メモリ、入出力ポート、ネットワークポート等）を接続し、これら要素間での情報の転送を可能にする。システムバス７９には、種々の入力および出力装置（例えば、キーボード、マウス、ディスプレイ、プリンタ、スピーカ等）をコンピュータ５０、６０へ接続するための入出力装置インターフェース８２が接続される。ネットワークインターフェース８６により、コンピュータはネットワーク（例えば、図８のネットワーク７０）に接続された種々の他の装置に接続される。メモリ９０は、本発明の実施形態（例えば、ここで述べる解決クライアントおよび

解析エンジン要素）を実現するために使用されるコンピュータソフトウェア指示９２およびデータ９４のための揮発性記憶装置を提供する。ディスク記憶装置９５は、本発明の実施形態を実現するために使用されるコンピュータソフトウェア指示９２およびデータ９４のための不揮発性記憶装置を提供する。中央処理装置８４もまたシステムバス７９に接続され、コンピュータ指示の実行に備える。

【００７６】

本発明は、その好適な実施形態を参照して特に示されかつ記載されたが、その好適な実施形態においては、添付の特許請求の範囲に包含される本発明の範囲を逸脱せずに、形態および詳細において種々の変更がなされ得ることが当業者によって理解されるであろう。

【図１】

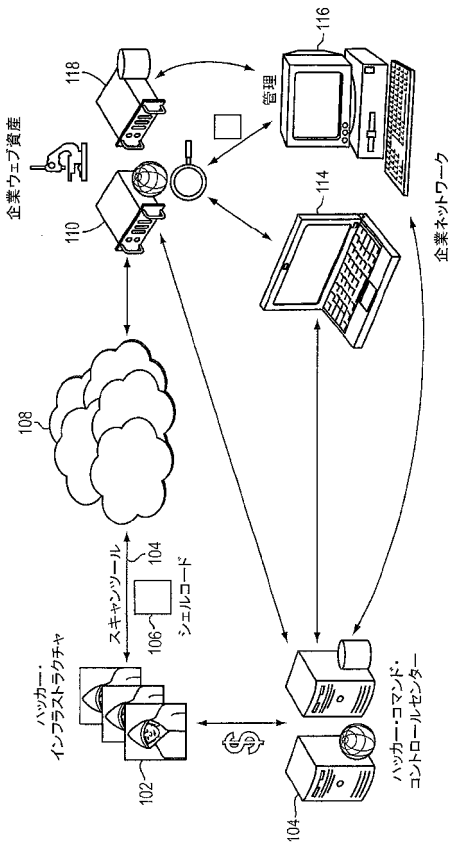

FIG. 1

【図２】

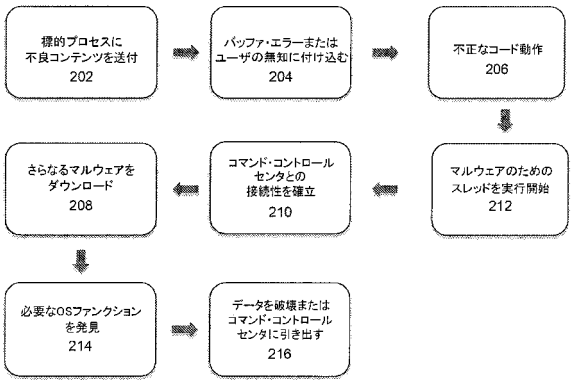

FIG. 2

【図３】

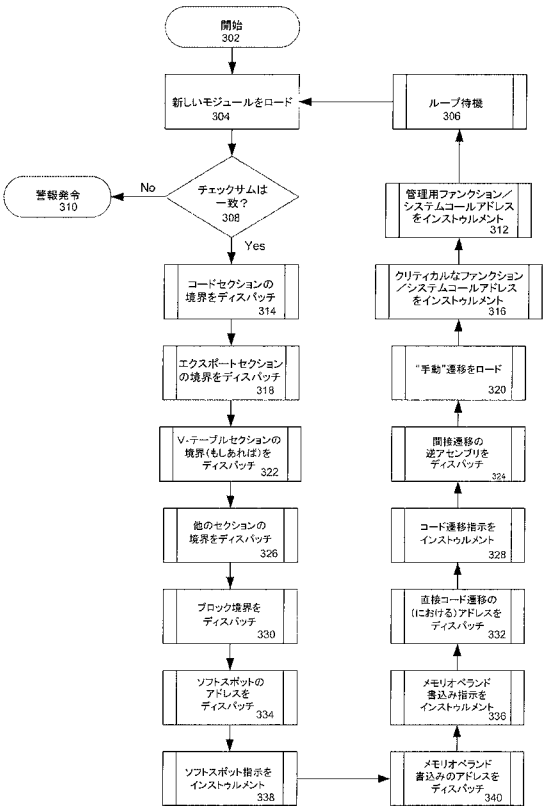

FIG. 3

【図４】

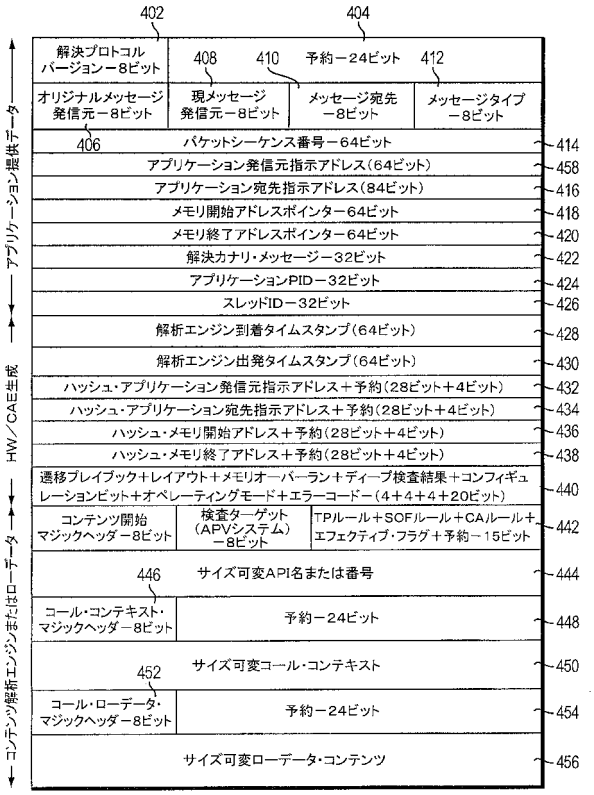

FIG. 4

【図５】

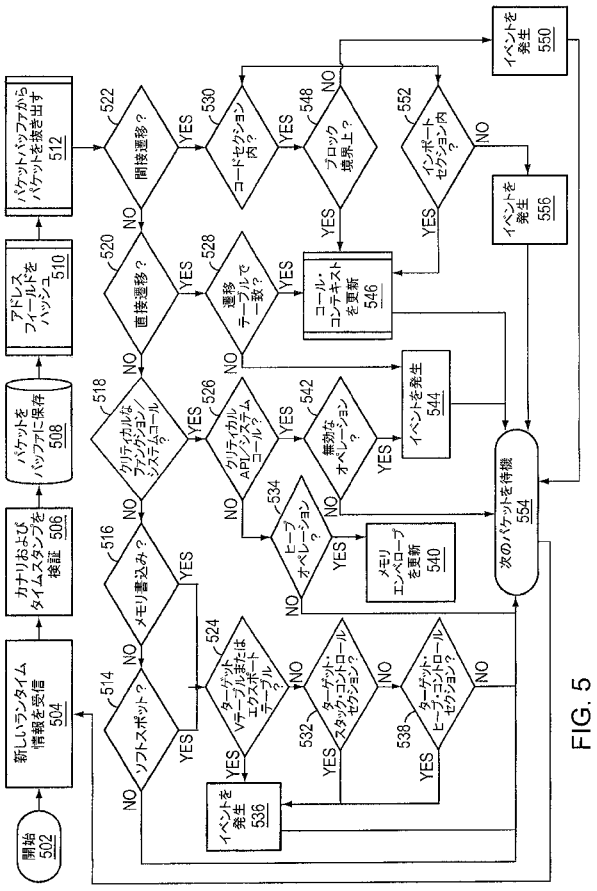

FIG. 5

【図６】

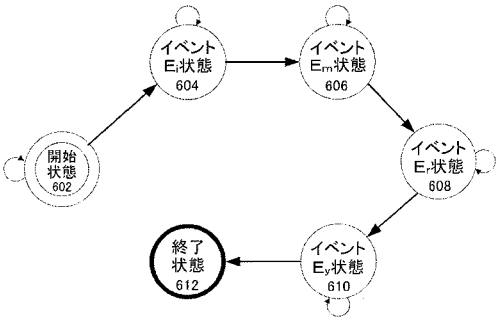

FIG. 6

【図７】

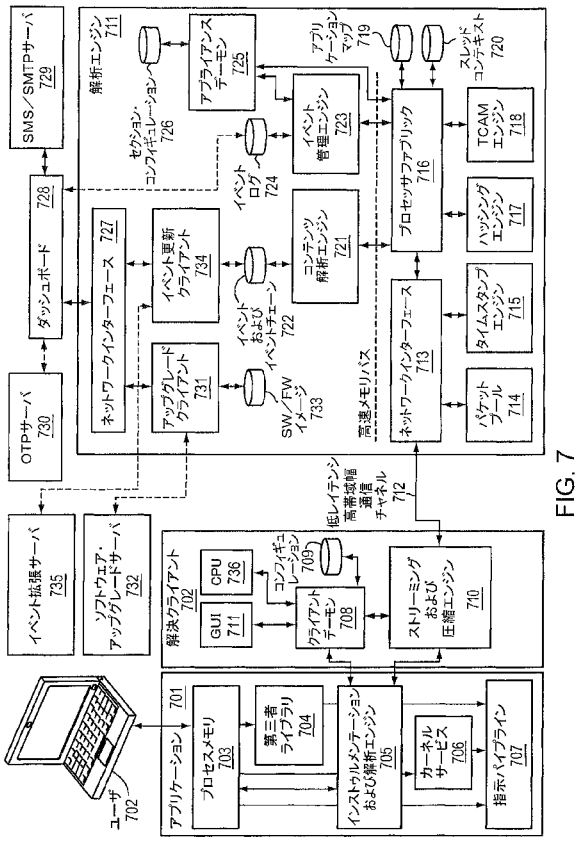

FIG. 7

【図８】

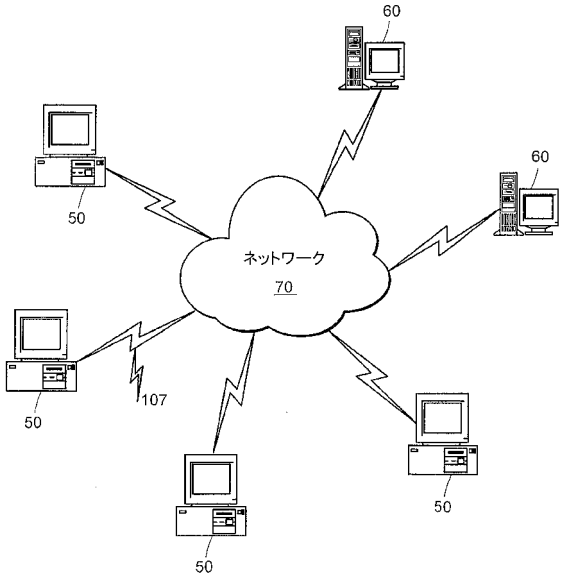

FIG. 8

【図９】

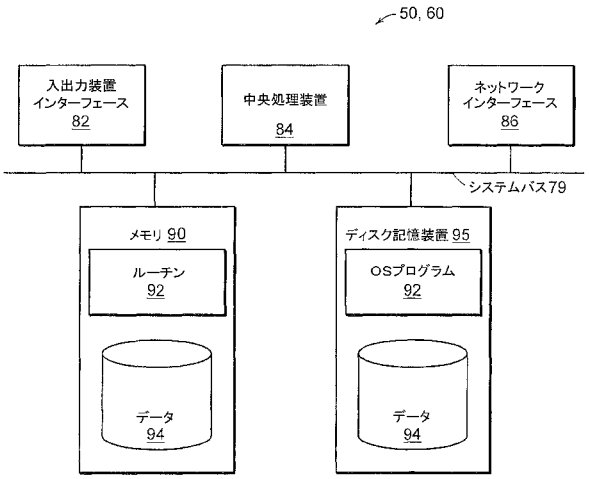

FIG. 9

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/055469

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/56
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 983 348 A (JI SHUANG [US]) 9 November 1999 (1999-11-09) column 7 - column 8; figure 2	1-115
X	US 8 510 596 B1 (GUPTA SATYA V [US] ET AL) 13 August 2013 (2013-08-13) column 12; figure 9	1-115
A	US 8 353 040 B2 (TAHAN GIL [IL] ET AL) 8 January 2013 (2013-01-08) abstract; figure 1	1-115
A	US 5 440 723 A (ARNOLD WILLIAM C [US] ET AL) 8 August 1995 (1995-08-08) abstract	1-115

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 November 2014	09/12/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Jascau, Adrian

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/055469

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5983348 A	09-11-1999	AU 8822998 A	29-03-1999
		US 5983348 A	09-11-1999
		US 6272641 B1	07-08-2001
		WO 9913402 A1	18-03-1999
US 8510596 B1	13-08-2013	NONE	
US 8353040 B2	08-01-2013	EP 1959367 A2	20-08-2008
		IL 181426 A	30-06-2011
		US 2008201779 A1	21-08-2008
US 5440723 A	08-08-1995	NONE	

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

(74)代理人 100144082
弁理士 林田 久美子
(74)代理人 100154771
弁理士 中田 健一
(74)代理人 100150566
弁理士 谷口 洋樹
(72)発明者 グプタ・サチャ・ヴラ
アメリカ合衆国, マサチューセッツ州 ０１７２０, アクトン, ブランブル ウェイ ３
(72)発明者 デミオ・レイモンド・エフ
アメリカ合衆国, マサチューセッツ州 ０１７４０, ボルトン, フォックス ラン ロード １５９