(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2021年6月17日(17.06.2021)

WIPO | PCT

(10) 国際公開番号
WO 2021/117665 A1

(51) 国際特許分類:
G06F 21/52 (2013.01)

(21) 国際出願番号： PCT/JP2020/045419

(22) 国際出願日： 2020年12月7日(07.12.2020)

(25) 国際出願の言語： 日本語

(26) 国際公開の言語： 日本語

(30) 優先権データ：
特願 2019-225330 2019年12月13日(13.12.2019) JP

(71) 出願人:株式会社日立製作所(**HITACHI, LTD.**) [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目６番６号 Tokyo (JP).

(72) 発明者:新保 健一(**SHIMBO Kenichi**); 〒1008280 東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内 Tokyo (JP). 鳥羽 忠信(**TOBA Tadanobu**); 〒1008280 東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内 Tokyo (JP).

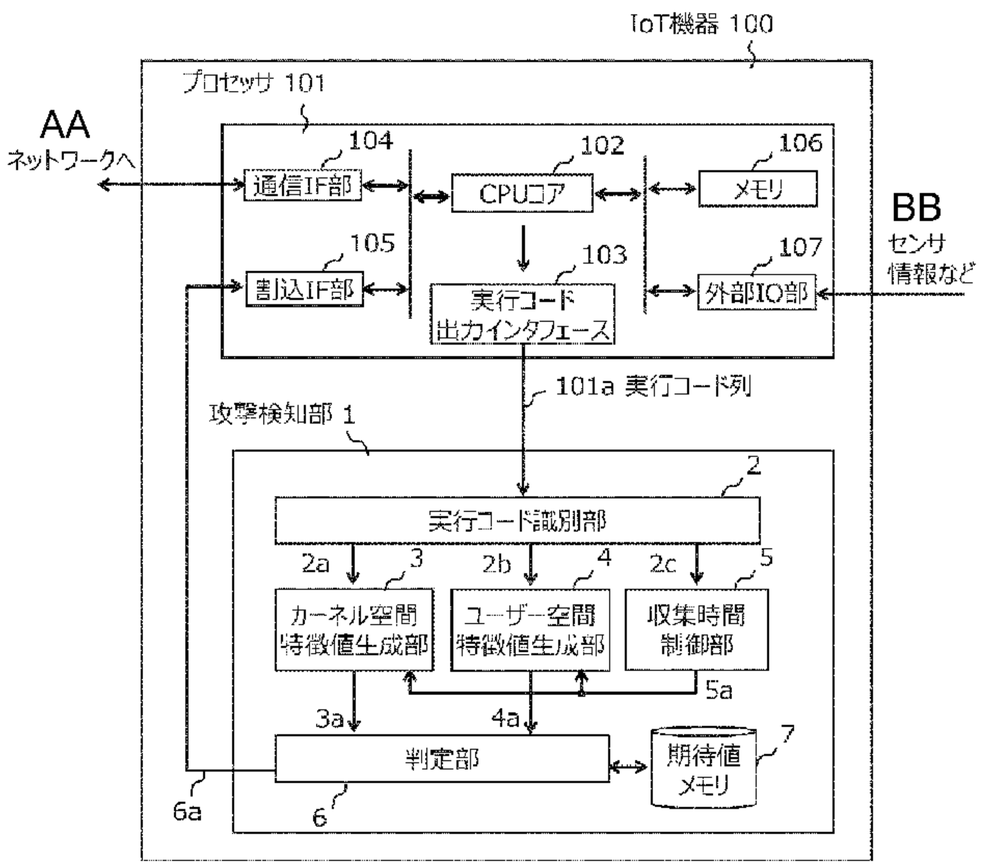
(74) 代理人: 特許業務法人 湘洋内外特許事務所(**SHOYO INTELLECTUAL PROPERTY FIRM**); 〒2200004 神奈川県横浜市西区北幸二丁目１５番１号 東武横浜第２ビル６階 Kanagawa (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH,

(54) **Title:** ELECTRONIC INSTRUMENT, AND ATTACK DETECTING METHOD FOR ELECTRONIC INSTRUMENT

(54) 発明の名称： 電子機器および電子機器の攻撃検知方法

[図1]



1 Attack detecting unit
2 Executable code identifying unit
3 Kernel space characteristic value generating unit
4 User space characteristic value generating unit
5 Collection time control unit
6 Determining unit
7 Expected value memory
100 IoT equipment
101 Processor
101a Executable code string
102 CPU core
103 Executable code output interface
104 Communication interface unit
105 Interrupt interface unit
106 Memory
107 External I/O unit
AA To network
BB Sensor information etc.

(57) **Abstract:** The objective of the present invention is to detect, in real-time, a malware infection or an abnormal operation resulting from a malicious attack, even in an electronic instrument having relatively poor processing capability. This electronic instrument is provided with an executable code identifying unit which accepts an executable code string output from a processor, and identifies, at least, an execution address in a user program area of an operating system (OS), and an execution address in a kernel area, and a determining unit which compares a prescribed characteristic value obtained from an identification result obtained by the executable code identifying unit at a prescribed timing with a prescribed expected value, and if the difference is equal to or greater than a prescribed difference, determines that there has been an attack, wherein the identification unit notifies the processor using a prescribed abnormality notification signal if it is determined that there has been an attack.

［続葉有］

WO 2021/117665 A1

CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, KE, KG, KH, KN, KP, KR, KW, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (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), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告(条約第21条(3))

(57) 要約:処理能力が比較的劣る電子機器であっても、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知する。 電子機器であって、プロセッサから出力される実行コード列を受け付けて、ＯＳ（Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）のユーザープログラム領域の実行アドレスと、カーネル領域の実行アドレスと、を少なくとも識別する実行コード識別部と、所定のタイミングで実行コード識別部による識別結果から得られる所定の特徴値を所定の期待値と照合して所定以上の差がある場合に攻撃があると判定する判定部と、を備え、判定部は、攻撃があると判定すると、プロセッサに所定の異常通知信号を通知する。

明　細　書

発明の名称 ： 電子機器および電子機器の攻撃検知方法

技術分野

[0001] 本発明は、電子機器および電子機器の攻撃検知方法に関する。本発明は2019年12月13日に出願された日本国特許の出願番号2019-225330の優先権を主張し、文献の参照による織り込みが認められる指定国については、その出願に記載された内容は参照により本出願に織り込まれる。

背景技術

[0002] 特許文献１には、サーバーを対象に、手動で収集できるシステムパラメータ（ログインユーザ数、ｗｅｂヒット率、受信パケット数、ＣＰＵ稼働率、ログインエラー、パリティエラーなど）から求めたシグネチャ（特徴値）の時間変化を監視することで異常状態を検出する技術が記載されている。

先行技術文献

特許文献

[0003] 特許文献1：特開２００４－３４８７４０号公報

発明の概要

発明が解決しようとする課題

[0004] 上述の特許文献１に記載の技術では、プロセッサ内部の実行状態までの監視はしておらず、カーネルレベルでの改ざんを含めたプロセッサ単位での攻撃検知は難しい。

[0005] 本発明の目的は、処理能力が比較的劣る電子機器であっても、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知することを目的とする。

課題を解決するための手段

[0006] 本願は、上記課題の少なくとも一部を解決する手段を複数含んでいるが、その例を挙げるならば、以下のとおりである。

[0007] 本発明の一態様は、電子機器であって、プロセッサから出力される実行コ

ード列を受け付けて、ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）のユーザープログラム領域の実行アドレスと、カーネル領域の実行アドレスと、を少なくとも識別する実行コード識別部と、所定のタイミングで前記実行コード識別部による識別結果から得られる所定の特徴値を所定の期待値と照合して所定以上の差がある場合に攻撃があると判定する判定部と、を備え、前記判定部は、攻撃があると判定すると、前記プロセッサに所定の異常通知信号を通知する、ことを特徴とする。

発明の効果

[0008] 本発明によれば、処理能力が比較的劣る電子機器であっても、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知する技術を提供することができる。

[0009] 上記した以外の課題、構成及び効果は、以下の実施形態の説明により明らかにされる。

図面の簡単な説明

[0010] [図1]第一の実施形態における電子機器であるＩｏＴ機器の構成例を示す図である。

[図2]第一の実施形態における実行コード列のデータと特徴値の例を示す図である。

[図3]判定部および期待値メモリの構成例を示す図である。

[図4]第一の実施形態における監視及び復旧処理のフローの例を示す図である。

[図5]第二の実施形態における電子機器であるＩｏＴ機器の構成例を示す図である。

[図6]第二の実施形態における実行コード列のデータと特徴値の例を示す図である。

[図7]第二の実施形態における監視及び復旧処理のフローの例を示す図である。

[図8]第二の実施形態における攻撃検知部の別の構成例を示す図である。

[図9]第三の実施形態における電子機器であるＩｏＴ機器の構成例を示す図である。

[図10]第四の実施形態における電子機器であるＥＣＵの構成例を示す図である。

発明を実施するための形態

[0011] 以下の実施形態においては便宜上その必要があるときは、複数のセクションまたは実施の形態に分割して説明するが、特に明示した場合を除き、それらはお互いに無関係なものではなく、一方は他方の一部または全部の変形例、詳細、補足説明等の関係にある。

[0012] また、以下の実施形態において、要素の数等（個数、数値、量、範囲等を含む）に言及する場合、特に明示した場合および原理的に明らかに特定の数に限定される場合等を除き、その特定の数に限定されるものではなく、特定の数以上でも以下でもよい。

[0013] さらに、以下の実施形態において、その構成要素（要素ステップ等も含む）は、特に明示した場合および原理的に明らかに必須であると考えられる場合等を除き、必ずしも必須のものではないことは言うまでもない。

[0014] 同様に、以下の実施形態において、構成要素等の形状、位置関係等に言及するときは特に明示した場合および原理的に明らかにそうではないと考えられる場合等を除き、実質的にその形状等に近似または類似するもの等を含むものとする。このことは、上記数値および範囲についても同様である。

[0015] また、実施形態を説明するための全図において、同一の部材には原則として同一の符号を付し、その繰り返しの説明は省略する。ただし、同一の部材であっても環境変更等により変更前の部材と称呼を共有すると混乱を生ぜしめるおそれが高い場合、別の異なる符号や名称を付すことがある。以下、本発明の各実施形態について図面を用いて説明する。

[0016] ＩｏＴ（Ｉｎｔｅｒｎｅｔ　ｏｆ　Ｔｈｉｎｇｓ）を活用したシステムでは、ｗｅｂカメラやセンサー機器はもちろん、コネクテッドカーやＰＬＣ、産業ロボットなど、多数の組み込み機器がネットワークに接続されるが、Ｉ

ｏＴ機器自身を対象とした乗っ取りやマルウェア感染による情報窃取、ボットネット利用といったセキュリティ被害が顕在化している。

[0017] 従来のサーバーレベルの対策としては、ネットワークの境界にファイヤウォールを設置し、不審パケットの侵入や不正アクセスを防止したり、暗号通信やデジタル証明書を用いた認証などによって、通信情報の盗聴、改ざんを防ぐ手法が有効である。

[0018] しかし、ＩｏＴ機器は製造・運用コストや性能リソースに限りがあるため、サーバーと同様のセキュリティ対策を実現することは一般的に困難である。一方で、近年ではＯＳが搭載されたＩｏＴ機器等のプロセッサのブートファイルを直接改ざんする攻撃も登場している。そこで、攻撃やウイルス感染を完全に阻止するのではなく、ＩｏＴ機器自身が動作状態を監視し、プロセッサに対する悪意のある攻撃の検知・復旧技術が重要になってきている。

[0019] また、ネットワークに接続されるＩｏＴ機器では、プロセッサにＬｉｎｕｘなどのＯＳを実装する機器が増えており、カーネルによる定期割込みや、仮想アドレスにより動作する等の要因により、従来の物理アドレスベースの実行コードによる診断が難しくなっている。また、プロセッサの処理内容のトレースも制限されることが多くなっているため、全ての処理内容を監視することも難しい。

[0020] 本発明に係る技術を適用したＩｏＴ機器を含む電子機器は、プロセッサの処理性能を低下させることなく、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知できる。また、マルウェア感染によりボットネット化など不正な動作状態となったＩｏＴ機器を短時間で特定し、上位システム等に通知できるため、異常機器を早期にネットワークから切断する、などの対策を行うことが可能となり、セキュリティ被害の拡大を抑止できる。

[0021] 図１～４を用いて、プロセッサの実行コード情報をもとに、ユーザープログラム実行領域（以降、「ユーザー空間」ともいう）での動作と、カーネル領域（以降、「カーネル空間」ともいう）での動作とを区別して監視するこ

とで、プログラムの改ざんや外部からの不正操作などプロセッサへの攻撃事象を検出することができる攻撃検出機能付きＩｏＴ機器（電子機器）を例に用いて説明する。

[0022] なお、ＩｏＴとは、日本語では「モノのインターネット」と呼ばれ、一般的には従来インターネットに接続されていなかったモノ（センサー機器、アクチュエーター、建物、車、電子機器など）が、ネットワークを通じてサーバーやクラウドに接続されることによって、これまで埋もれていたデータを処理、変換、分析、連携することが可能となることをいう。本願でも、同様の概念としてＩｏＴという言葉を用いる。

[0023] 図１は、第一の実施形態における電子機器であるＩｏＴ機器の構成例を示す図である。ＩｏＴ機器１００は、メインの処理（ＩｏＴ機器１００の担う主要な機能に関する処理）を行うプロセッサ１０１と、マルウェア感染や乗っ取りなどによるプロセッサの異常なふるまいを検知する攻撃検知部１を含んで構成される。

[0024] ＩｏＴ機器１００自身の動作例としては、図示していないが、センサーモジュールでデジタル化されたＩｏＴ機器１００周辺の外界情報（物性の測定情報）をセンサー情報としてプロセッサ１０１で収集し、ネットワーク経由でクラウドサーバーなどに転送するというものである。

[0025] ここで、プロセッサ１０１は、１個または複数のＣＰＵコア１０２と、実行プログラムやデータを格納するためのメモリ１０６、ＣＰＵコア１０２で実行された実行コードをプロセッサの外部に順次出力する実行コード出力インターフェース１０３、外部ネットワークに接続する他の機器との通信を行うための通信部となる通信インターフェース（ＩＦ）部１０４、外部からの割込み信号を受信するための割込みインターフェース（ＩＦ）部１０５、センサーやアクチュエーターなどの外部入出力信号を接続するための外部ＩＯ部１０７を含んで構成される。

[0026] プロセッサ１０１は、メモリ１０６に格納されたプログラムに従い、センサー情報に対する演算処理や統計処理、ＡＩ（人工知能）処理などを行う。

また、プロセッサ１０１にはＬｉｎｕｘなどのＯＳが実装され、これらの処理を細かいプロセスに分割し、プロセッサ内の複数のＣＰＵコア１０２にスケジューリングして実行する。実行コード出力インターフェース１０３は、ＣＰＵコア１０２で処理された実行アドレスや例外処理などが含まれた情報を順次、実行コード列１０１ａとしてプロセッサ１０１の外部の攻撃検知部１に出力する。

[0027] 攻撃検知部１は、実行コード識別部２と、カーネル空間特徴値生成部３と、ユーザー空間特徴値生成部４と、収集時間制御部５と、判定部６と、期待値メモリ７とを含む。攻撃検知部１は、プロセッサ１０１から出力される実行コード列１０１ａを用いて攻撃を検知する。

[0028] 実行コード識別部２は、実行コード列１０１ａの情報から、カーネル空間における実行アドレス２ａと、ユーザー空間における実行アドレス２ｂと、例外処理による割込みコード２ｃと、を識別して出力する。カーネル空間における実行アドレス２ａと、ユーザー空間における実行アドレス２ｂとは、ＯＳで管理される仮想アドレス空間における命令やデータを呼び出す際のアドレスや分岐先のアドレス情報を示す。

[0029] カーネル空間特徴値生成部３は、入力されるカーネル空間における実行アドレス２ａを用いて所定の特徴値を生成する。例えば、カーネル空間特徴値生成部３は、カーネル空間における実行アドレス２ａのアクセス回数をカウントし、カーネル空間の特徴値３ａとして判定部６に出力する。

[0030] ユーザー空間特徴値生成部４も同様に、入力されるユーザー空間における実行アドレス２ｂを用いて所定の特徴値を生成する。例えば、ユーザー空間特徴値生成部４は、ユーザー空間における実行アドレス２ｂのアクセス回数をカウントし、ユーザー空間の特徴値４ａとして判定部６に出力する。

[0031] 収集時間制御部５は、割込みコード２ｃが入力されると特徴値生成信号５ａを生成し、カーネル空間特徴値生成部３およびユーザー空間特徴値生成部４に入力することで、それぞれの特徴値の出力タイミングを制御する。

[0032] 図２は、第一の実施形態における実行コード列のデータと特徴値の例を示

す図である。図中のグラフは、実行コード列１０１ａから識別したカーネル空間における実行アドレス２ａ、ユーザー空間における実行アドレス２ｂ、割込みコード２ｃを基にした特徴値の生成を説明するイメージである。図中のグラフは、縦軸を仮想アドレス、横軸を実行アドレスの検出サイクル、つまりはプロセッサの動作時間とし、実行コード列１０１ａから識別されたカーネル空間における実行アドレス２ａ、ユーザー空間における実行アドレス２ｂ、割込みコード２ｃを時間順にプロットしている。

[0033] 仮想アドレス空間は、ＯＳの設定により、ユーザー空間のアドレス領域と、カーネル空間のアドレス領域に分けられる。つまり、実行アドレスから、カーネル空間における処理が実行されたか、ユーザープログラム空間における処理が実行されたかを判別できる。さらに、割込みコード２ｃによれば、カーネルの定期割込みや、その他の例外処理が発生したことを判別できる。

[0034] 第一の実施形態においては、収集時間制御部５からの特徴値生成信号５ａを受信したカーネル空間特徴値生成部３およびユーザー空間特徴値生成部４は、それまでに収集した実行アドレスの回数情報を特徴値として出力する。つまり、割込みコードの発生間隔が、カーネル空間特徴値生成部３およびユーザー空間特徴値生成部４における実行アドレスの収集区間となる。カーネル空間特徴値生成部３と、ユーザー空間特徴値生成部４とは、特徴値生成信号５ａを受信するごとに、その収集区間において収集したカーネル空間における実行アドレス２ａ、ユーザー空間における実行アドレス２ｂのデータから生成された特徴値をそれぞれ出力する。

[0035] 図３は、判定部および期待値メモリの構成例を示す図である。判定部６は、入力されるカーネル空間の特徴値３ａの正常判定を行うカーネル空間特徴値比較部６０と、入力されるユーザー空間の特徴値４ａの正常判定を行うユーザー空間特徴値比較部６１と、異常レベル判定部６２と、を含んでいる。

[0036] 期待値メモリ７は、マルウェア感染や攻撃を受けていない正常動作時の複数の特徴値を、あらかじめ期待値としてそれぞれカーネル空間期待値データ群７１と、ユーザー空間期待値データ群７２として格納している。

[0037] 判定部６は、入力されるカーネル空間の特徴値３ａ、ユーザー空間の特徴値４ａのそれぞれについて、期待値メモリ７から読み出すカーネル空間期待値データ群７１およびユーザー空間期待値データ群７２と照合する。

[0038] カーネル空間特徴値比較部６０は、入力されるカーネル空間の特徴値３ａと、カーネル空間期待値データ群７１のいずれの期待値データとも不一致の場合に不一致信号６０ａを出力し、異常レベル判定部６２に入力する。また、ユーザー空間特徴値比較部６１は、入力されるユーザー空間の特徴値４ａと、ユーザー空間期待値データ群７２のいずれの期待値データとも不一致の場合に不一致信号６１ａを出力し、異常レベル判定部６２に入力する。

[0039] 異常レベル判定部６２は、入力された不一致信号６０ａと不一致信号６１ａとの組み合わせに応じて異常レベルを判定し、プロセッサ１０１への異常通知信号（割込み信号）６ａを生成する。異常通知信号６ａは、異常レベルを表すために複数ビットの信号で構成されてもよい。

[0040] 図４は、第一の実施形態における監視及び復旧処理のフローの例を示す図である。図示していないが、プロセッサ１０１および実行コード出力インターフェース１０３は、ＯＳのブートおよびソフトウェアリセットを含む初期化処理完了直後から実行コード列１０１ａの出力を開始する。

[0041] 随時、攻撃検知部１は、監視対象のプロセッサ１０１から出力される実行コード列１０１ａを受信する。実行コード識別部２は、受信した実行コード列１０１ａから、カーネル空間における実行アドレス２ａと、ユーザー空間における実行アドレス２ｂと、割込みコード２ｃと、を識別する（ステップＳ１００）。

[0042] 次に、収集時間制御部５は、割込みコード２ｃが入力されたタイミングで特徴値生成信号５ａを生成し、カーネル空間特徴値生成部３およびユーザー空間特徴値生成部４に入力する（ステップＳ１０１）。

[0043] そして、特徴値生成信号５ａによって、カーネル空間特徴値生成部３およびユーザー空間特徴値生成部４は、生成した特徴値を出力する（ステップＳ１０２）。具体的には、収集時間制御部５からの特徴値生成信号５ａを受信

したカーネル空間特徴値生成部３およびユーザー空間特徴値生成部４は、前回受信した特徴値生成信号５ａがある場合にはその間に、無い場合にはそれまでに収集した実行アドレスをもとに、特徴値を生成し、生成した特徴値を判定部６に出力する。

[0044] 判定部６は、カーネル空間およびユーザー空間のそれぞれの特徴値について、あらかじめ期待値メモリ７に格納された期待値データ群とそれぞれを照合する（ステップＳ１０３）。

[0045] そして、判定部６は、不一致信号６０ａ、６１ａのいずれかが出力されたか否か判定する（ステップＳ１０４）。いずれも出力されていない場合（ステップＳ１０４にて「ＮＯ」の場合）には、判定部６は、ステップＳ１００に制御を戻す。

[0046] 不一致信号６０ａ、６１ａのいずれかが出力された場合（ステップＳ１０４にて「ＹＥＳ」の場合）には、判定部６は、不一致となった空間の組み合わせによって異常通知信号６ａをプロセッサ１０１に通知する（ステップＳ１０５）。

[0047] 具体的には、ユーザー空間における実行アドレスに関する特徴値のみが期待値のいずれとも不一致だった場合には、ユーザープログラムの改ざんのおそれがあるため、異常レベル判定部６２は、ソフトウェアリセットをプロセッサ１０１にリクエストする異常通知信号を出力する。

[0048] カーネル空間における実行アドレスに関する特徴値が期待値のいずれとも不一致だった場合は、ＯＳの改ざんのおそれがあるため、異常レベル判定部６２は、ＩｏＴ機器１００をネットワークから切り離し、ブートファイルやＲＯＭデータのチェックを行わせる異常通知信号をプロセッサ１０１に出力する。

[0049] そして、プロセッサ１０１は、異常通知信号の種類によって、ＣＰＵコア１０２の復旧処理を実行する（ステップＳ１０６）。具体的には、上述のように、プロセッサ１０１は、ソフトウェアリセット、またはＩｏＴ機器１００をネットワークから切り離してブートファイルやＲＯＭデータのチェック

を行う。

[0050] 以上が、第一の実施形態に係る監視および復旧処理の流れである。本実施形態におけるＩｏＴ機器１００では、プロセッサ１０１の実行コード情報をもとに、ユーザー領域での動作と、カーネル領域での動作とを区別して監視し、プログラムの改ざんや外部からの不正操作などプロセッサへの攻撃事象を軽い処理により検出することができる。つまり、処理能力が比較的劣る電子機器に対しても、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知する信頼性・セキュリティ耐性を向上させることができる。

[0051] 本実施形態においては、攻撃検知部１は、プロセッサと別のデバイス（コプロセッサ）で実現してもよいし、同様のプロセッサを用意し、相互監視してもよい。あるいは、ＦＰＧＡ（Ｆｉｅｌｄ－Ｐｒｏｇｒａｍｍａｂｌｅ Ｇａｔｅ Ａｒｒａｙ）またはＡＳＩＣ（Ａｐｐｌｉｃａｔｉｏｎ ｓｐｅｃｉｆｉｃ Ｉｎｔｅｇｒａｔｅｄ Ｃｉｒｃｕｉｔ）を用いて、プロセッサ１０１と攻撃検知部１とを１チップ内で実現してもよい。期待値メモリ７の期待値データは、プロセッサが起動した時点であらかじめ書き込まれている構成として説明しているが、プロセッサ１０１経由で新たな期待値を書き込むこともできる。また、カーネル空間およびユーザー空間のそれぞれの特徴値を期待値と直接比較しているが、これに限られず、判定部６は、所定の特徴値としてユーザープログラム領域の実行アドレス数と、カーネル領域の実行アドレス数と、の比の値を特徴値として用いるものであってもよい。また、カーネル空間特徴値比較部６０およびユーザー空間特徴値比較部６１は、入力される特徴値が、期待値データ群のいずれの期待値データとも不一致の場合に不一致信号を出力しているが、これに限られず、所定の差を越える場合に不一致信号を出力するようにしてもよい。

[0052] 以上が、第一の実施形態に係る電子機器である攻撃検知機能付きＩｏＴ機器である。第一の実施形態に係る電子機器によれば、処理能力が比較的劣る電子機器であっても、マルウェア感染や、悪意のある攻撃による異常動作を

リアルタイムに検知することができる。

[0053] ［第二の実施形態］本発明は、第一の実施形態に係る電子機器に限られるものではない。例えば、プロセッサの実行コード情報をもとに、ユーザー空間での動作と、カーネル空間での動作に加えて、割込みに関する動作も特徴値として監視することで、プログラムの改ざんや外部からの不正操作などプロセッサへの攻撃事象を検出するようにしてもよい。このような第二の実施形態について、図５～８を用いて説明する。

[0054] 第二の実施形態に係る電子機器は、第一の実施形態に係る電子機器と、基本的には同一であるが、一部に相違がある。以下は、その相違点を中心に説明する。

[0055] 図５は、第二の実施形態における電子機器であるＩｏＴ機器１００´の構成例を示す図である。ＩｏＴ機器１００´は、攻撃検知部１０を含んでいる。攻撃検知部１０は、実行コード識別部２と、カーネル空間特徴値生成部１１と、ユーザー空間特徴値生成部１２と、割込み特徴値生成部１３と、判定部１４と、期待値メモリ１５とを含む。

[0056] カーネル空間特徴値生成部１１は、入力されるカーネル空間における実行アドレス２ａを用いて所定の特徴値を生成する。例えば、カーネル空間特徴値生成部１１は、カーネル空間における実行アドレス２ａのアクセス回数をカウントし、カーネル空間の特徴値１１ａとして判定部１４に出力する。

[0057] ユーザー空間特徴値生成部１２も同様に、入力されるユーザー空間における実行アドレス２ｂを用いて所定の特徴値を生成する。例えば、ユーザー空間特徴値生成部１２は、ユーザー空間における実行アドレス２ｂのアクセス回数をカウントし、ユーザー空間の特徴値１２ａとして判定部１４に出力する。

[0058] 割込み特徴値生成部１３は、入力される割込みコード２ｃを用いて所定の特徴値を生成する。例えば、割込み特徴値生成部１３は割込み回数をカウントし、特徴値１３ａとして判定部１４に出力する。ここで、攻撃検知部１０には、図示しないが任意時間Ｔａを設定できる内部計時部を有しており、各

特徴値の生成サイクルを任意時間Ｔａにて管理する。

[0059] 図６は、第二の実施形態における実行コード列のデータと特徴値の例を示す図である。図中のグラフは、実行コード列１０１ａから識別したカーネル空間における実行アドレス２ａ、ユーザー空間における実行アドレス２ｂ、割込みコード２ｃを基にした特徴値の生成を説明するイメージである。図中のグラフは、縦軸を仮想アドレス、横軸を実行アドレスの検出サイクル、つまりはプロセッサの動作時間とし、実行コード列１０１ａから識別されたカーネル空間における実行アドレス２ａ、ユーザー空間における実行アドレス２ｂ、割込みコード２ｃを時間順にプロットしている。

[0060] ここで、カーネル空間特徴値生成部１１、ユーザー空間特徴値生成部１２および割込み特徴値生成部１３は、内部計時部に設定された任意時間Ｔａで示すタイマ時間内において、各実行アドレスおよび割込みの回数を特徴値としてそれぞれ出力する。

[0061] 判定部１４は、入力されるカーネル空間の特徴値１１ａ、ユーザー空間の特徴値１２ａ、割込みの特徴値１３ａのそれぞれについて、期待値メモリ１５から読み出すカーネル空間期待値データ群７１、ユーザー空間期待値データ群７２および割込み期待値データ群と比較する。

[0062] 判定部１４は、入力されるカーネル空間の特徴値１１ａと、カーネル空間期待値データ群７１のいずれの期待値データとも不一致の場合に不一致信号を出力する。また、判定部１４は、入力されるユーザー空間の特徴値１２ａと、ユーザー空間期待値データ群７２のいずれの期待値データとも不一致の場合に不一致信号を出力する。また、判定部１４は、入力される割込みの特徴値１３ａと、割込み期待値データ群のいずれの期待値データとも不一致の場合に不一致信号を出力する。

[0063] 判定部１４は、カーネル空間特徴値生成部１１、ユーザー空間特徴値生成部１２および割込み特徴値生成部１３のいずれかの生成部が生成した特徴値に不一致信号が出力されると、その組み合わせに応じて、プロセッサの異常動作を示す異常通知信号１４ａを出力する。

[0064] 期待値メモリ１５は、マルウェア感染や攻撃を受けていない正常動作時の複数の特徴値を、あらかじめ期待値としてそれぞれカーネル空間期待値データ群７１と、ユーザー空間期待値データ群７２と、割込み期待値データ群として格納している。

[0065] 図７は、第二の実施形態における監視及び復旧処理のフローの例を示す図である。図示していないが、プロセッサ１０１および実行コード出力インターフェース１０３は、ＯＳのブートおよびソフトウェアリセットを含む初期化処理完了直後から実行コード列１０１ａの出力を開始する。

[0066] 随時、攻撃検知部１０は、監視対象のプロセッサ１０１から出力される実行コード列１０１ａを受信する。実行コード識別部２は、受信した実行コード列１０１ａから、カーネル空間における実行アドレス２ａと、ユーザー空間における実行アドレス２ｂと、割込みコード２ｃと、を識別する（ステップＳ２００）。

[0067] 次に、実行コード識別部２は、識別した３種類の情報（カーネル空間の実行アドレス２ａと、ユーザー空間の実行アドレス２ｂと、割込みコード２ｃ）を内部計時部に設定された時間区間（任意時間Ｔａ）で収集する（ステップＳ２０１）。

[0068] そして、各特徴値の生成部は、収集した情報（カーネル空間の実行アドレス２ａと、ユーザー空間の実行アドレス２ｂと、割込みコード２ｃ）の特徴値を、それぞれ生成する（ステップＳ２０２）。

[0069] そして、判定部１４は、各特徴値について、期待値メモリ１５に格納されたそれぞれの期待値データ群と照合する（ステップＳ２０３）。

[0070] そして、判定部１４は、いずれかの不一致信号が出力されたか否か判定する（ステップＳ２０４）。いずれも出力されていない場合（ステップＳ２０４にて「ＮＯ」の場合）には、判定部１４は、ステップＳ２００に制御を戻す。

[0071] いずれかの不一致信号が出力された場合（ステップＳ２０４にて「ＹＥＳ」の場合）には、判定部１４は、不一致となった空間あるいは割込みコード

の組み合わせによって異常通知信号１４ａをプロセッサ１０１に通知する（ステップＳ２０５）。

[0072] 具体的には、ユーザー空間における実行アドレスに関する特徴値のみが期待値のいずれとも不一致だった場合には、ユーザープログラムの改ざんのおそれがあるため、判定部１４は、ソフトウェアリセットをプロセッサ１０１にリクエストする異常通知信号１４ａを通知する。

[0073] カーネル空間における実行アドレスに関する特徴値が期待値のいずれとも不一致だった場合は、ＯＳの改ざんのおそれがあるため、判定部１４は、ＩｏＴ機器１００´をネットワークから切り離し、ブートファイルやＲＯＭデータのチェックを行わせる異常通知信号１４ａをプロセッサ１０１に出力する。

[0074] 割込みのコードに関する特徴値が期待値のいずれとも不一致だった場合は、カーネル動作を含めて改ざんのおそれがあるため、判定部１４は、ＩｏＴ機器１００´をネットワークから切り離し、ブートファイルやＲＯＭデータのチェックを行わせ、ソフトウェアリセットをリクエストする異常通知信号１４ａをプロセッサ１０１に出力する。

[0075] なお、これに限られず、不一致となった空間あるいは割込みコードの組み合わせによって異常通知信号１４ａを細かに設定し、これを通知して対処を細かに変更するようにしてもよい。

[0076] そして、プロセッサ１０１は、異常通知信号の種類によって、ＣＰＵコア１０２の復旧処理を実行する（ステップＳ２０６）。具体的には、上述のように、プロセッサ１０１は、ソフトウェアリセット、またはＩｏＴ機器１００をネットワークから切り離してブートファイルやＲＯＭデータのチェックを行う。

[0077] 以上が、第二の実施形態に係る監視および復旧処理の流れである。第二の実施形態におけるＩｏＴ機器１００´では、プロセッサ１０１の実行コード情報をもとに、ユーザー領域での動作と、カーネル領域での動作と、割込みコードとを区別して監視し、プログラムの改ざんや外部からの不正操作など

プロセッサへの攻撃事象を軽い処理により検出することができる。つまり、処理能力が比較的劣る電子機器に対しても、マルウェア感染や、悪意のある攻撃による異常動作をリアルタイムに検知する信頼性・セキュリティ耐性を向上させることができる。

[0078] なお、任意時間Ｔａは、固定に限られず、外部環境等により変動するものであってもよい。例えば、工場で使用されるＩｏＴ機器の場合、工場の操業環境（操業日時や曜日、季節、気温等）によって変動させるものであってもよい。あるいは、車載のＩｏＴ機器の場合、車両の走行時と停車時で変動させるものであってもよい。

[0079] 図８は、第二の実施形態における攻撃検知部の別の構成例を示す図である。攻撃検知部２０は、第一の実施形態に係る収集時間制御部５（割込みコード２ｃの発生に応じたタイミングで特徴値の発生タイミングを制御する）をさらに有する。攻撃検知部２０は、割込みコード２ｃを特徴値と時間制御の両方に用いることで、さらに細かい異常判定が可能となる。この場合、割込みの発生間隔に応じて特徴値が生成されることになるため、割込みの特徴値は割込みの回数ではなく、割込みの発生間隔（時間）を特徴値とすることが望ましい。

[0080] 本別例の期待値メモリ２３内の期待値データは、３種類の特徴値をそれぞれ個別に判定するように格納してもよいし、例えば、割込みの発生間隔が同じときのカーネル空間またはユーザー空間の特徴値を判定するように、カーネル空間特徴値およびユーザー空間特徴値を、割込み発生間隔に紐付けて不一致判定を行うようにしてもよい。

[0081] また、カーネル空間、ユーザー空間、割込みのそれぞれの特徴値を期待値と直接比較しているが、これに限られず、判定部１４は、所定の特徴値としてユーザープログラム領域の実行アドレス数と、カーネル領域の実行アドレス数と、割込み回数または割込みの発生間隔の比の値を特徴値として用いるものであってもよい。

[0082] また、判定部１４は、入力される特徴値が、期待値データ群のいずれの期

待値データとも不一致の場合に不一致信号を出力しているが、これに限られず、所定の差を越える場合に不一致信号を出力するようにしてもよい。

[0083] 以上が、第二の実施形態の別の構成例である。この構成例では、攻撃検知部２０は、さらに細かい異常判定が可能となる。

[0084] ［第三の実施形態］本発明は、第一の実施形態、第二の実施形態およびその別例に係る電子機器に限られるものではない。例えば、正常動作時の特徴値情報を較正できるようにしてもよい。このような第三の実施形態に係るＩｏＴ機器１００´´の例を、図９を用いて説明する。

[0085] 第三の実施形態に係る電子機器は、第二の実施形態に係る電子機器と、基本的には同一であるが、一部に相違がある。以下は、その相違点を中心に説明する。

[0086] 図９は、第三の実施形態における電子機器であるＩｏＴ機器の構成例を示す図である。ＩｏＴ機器１００´´は、攻撃検知部３０を含んでいる。攻撃検知部３０は、実行コード識別部２と、カーネル空間特徴値生成部１１と、ユーザー空間特徴値生成部１２と、割込み特徴値生成部１３と、セレクタ３１と、判定部３２と、期待値メモリ３３とを含む。また、プロセッサ１０１に係る外部ＩＯ部１０７については、セレクタ３１、判定部３２、期待値メモリ３３に対してモード切替信号１０７ａを送る。

[0087] 攻撃検知部３０は、較正モードと通常判定モードの２種類の動作モードが設けられており、モード切替信号１０７ａを受け取ると、動作モードを切り替える。例えば、ＩｏＴ機器１００´´の出荷前試験時などには、正常な動作時に収集される特徴値を用いて較正する、すなわち期待値メモリ３３に対して学習させる較正モードで動作させ、出荷時にモード切替信号１０７ａをプロセッサ１０１から送らせることで動作モードを通常判定モードに切り替えて出荷することができる。

[0088] 動作モードの切り替え制御は、基本的にはプロセッサ１０１から行う。モード切替信号１０７ａは、攻撃検知部３０のセレクタ３１（選択部ともいう）、判定部３２および期待値メモリ３３に入力される。

[0089] セレクタ３１は、動作モードが通常判定モードであれば判定部３２に生成した特徴値を送り、動作モードが較正モードであれば期待値メモリ３３に生成した特徴値を送る。すなわち、セレクタ３１は、動作モードに応じて判定部３２か期待値メモリ３３のいずれかに特徴値を選択的に送る。

[0090] 具体的には、セレクタ３１は、カーネル空間の特徴値１１ａ、ユーザー空間の特徴値１２ａ、割込みの特徴値１３ａを出力するポートの切り替えを行う。通常判定モードの場合、セレクタ３１は、判定部３２側のポート３１ａから各特徴値を出力し、較正モードの場合、期待値メモリ３３側のポート３１ｂから各特徴値を出力するよう切り替える。

[0091] 判定部３２は、較正モードの場合には、プロセッサ１０１へ誤った異常通知が出力されないように判定処理を停止し、異常通知信号３２ａを出力しない。

[0092] 期待値メモリ３３は、較正モードの場合は、生成された各特徴値をポート３１ｂから受け取ると、メモリ内を検索し、同じ値が既に格納されていない場合に、空いているメモリ領域に新しい特徴値を追加して格納していく。較正モードで動作する間、期待値メモリ３３は上記の動作を繰り返し、動作中に生成された特徴値を格納（学習）していく。モード切替信号１０７ａを受け付けると、期待値メモリ３３は較正モードを解除し、通常判定モードに切り替える。通常判定モードでは、期待値メモリ３３は格納された特徴値をすべて、判定部３２に受け渡す。

[0093] 以上が、第三の実施形態に係るＩｏＴ機器１００´´である。第三の実施形態におけるＩｏＴ機器１００´´では、プロセッサ１０１の動作中に生成される特徴値を期待値メモリ３３に自動的に重複を排除して蓄積する機能を有することによって、例えばプロセッサ１０１のプログラムが変更された場合でも、一旦較正モードで動作させることによって自動で新たな特徴値に調整することができる。

[0094] ＩｏＴ機器１００´´では、接続されるセンサーの変更や取得データに対する演算処理の変更など、遠隔操作によるプログラムの一括アップデートが

なされることが予想される。その際、自動で較正を行う機能を有することによって、特にＩｏＴ機器１００´´が大量に存在する環境においてアップデート工数を削減できる。

[0095] また、較正の機能はこれに限られず、取得した実績値に対して簡単な統計処理を行って期待値を特定し較正するものであってもよい。例えば、中央値を採用することや、平均値を採用することもできる。あるいは、外部環境（例えば、日時あるいは曜日）に応じて個別のテーブルとして格納し、運用時の外部環境に応じてテーブルを切り替えることで期待値データ群を切り替えるようにしてもよい。このようにすることで、大幅な期待値の変動を除外することが可能となる。

[0096] ［第四の実施形態］本発明は、第一の実施形態、第二の実施形態、その別例、第三の実施形態に係る電子機器に限られるものではない。例えば、ＩｏＴ機器に限られず、車両に搭載されるＥＣＵ（Ｅｌｅｃｔｒｏｎｉｃ Ｃｏｎｔｒｏｌ Ｕｎｉｔ）に適用するようにしてもよい。このような第四の実施形態に係るＥＣＵの例を、図１０を用いて説明する。

[0097] 第四の実施形態に係る電子機器は、第三の実施形態に係る電子機器と、基本的には同一であるが、一部に相違がある。以下は、その相違点を中心に説明する。

[0098] 第四の実施形態における車載のＥＣＵ２００では、プロセッサ２０１内に、攻撃検知部３０´が含まれる。また、ＣＰＵコアに代えて、複数のＣＰＵを含むコア群２０２が含まれる。また、モード切替信号１０７ａは、共用メモリ部２０７を介してセレクタ３１、判定部３２、期待値メモリ３３に受け渡される。

[0099] 攻撃検知部３０´は、コア管理部３４を備え、判定部３２は、異常通知信号３２ａを割込ＩＦ部１０５ではなくコア管理部３４へ送る。

[0100] コア管理部３４は、コア群２０２の各コアの動作の制御信号３４ａを、割込ＩＦ２０５を介して送りコア群２０２から特定のコアを隔離し、あるいはコアをコア群２０２に参加させる。コア管理部３４は、判定部３２から出力

される異常通知信号３２ａを受け付けると、異常となったコアをコア群２０２から隔離することで、プロセッサ２０１への異常の伝播を回避する。

[0101] 特に、車両のＥＣＵは、リアルタイムな制御を行うものであり、かつ安全の確保に重要な役割を持つため、異常発生時の最小限の範囲のダメージコントロールと、迅速な復旧が重要である。

[0102] また、攻撃検知部３０´自体も、コア群２０２を構成する特定のコアが担うことも可能である。

[0103] 以上が、第四の実施形態に係る電子機器であるＥＣＵである。第四の実施形態に係る電子機器によれば、搭載される車両の異常検出を迅速に行い、影響を最小限に留め、復旧を迅速に行うことが可能となる。

[0104] 上記した実施形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。

[0105] また、実施形態の構成の一部について、構成の追加・削除・置換をすることが可能である。また、上記の各部、各構成、機能、処理部等は、それらの一部又は全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、上記の各部、各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリや、ハードディスク、ＳＳＤ等の記録装置、または、ＩＣカード、ＳＤカード、ＤＶＤ等の記録媒体に置くことができる。

[0106] なお、上述した実施形態にかかる制御線や情報線は説明上必要と考えられるものを示しており、製品上必ずしも全ての制御線や情報線を示しているとは限らない。実際にはほとんど全ての構成が相互に接続されていると考えても良い。

[0107] 以上、本発明について、実施形態を中心に説明した。

符号の説明

[0108] １：攻撃検知部、２：実行コード識別部、２ａ：カーネル空間における実行

アドレス、２ｂ：ユーザー空間における実行アドレス、２ｃ：割込みコード、３：カーネル空間特徴値生成部、３ａ：カーネル空間の特徴値、４：ユーザー空間特徴値生成部、４ａ：ユーザー空間の特徴値、５：収集時間制御部、５ａ：特徴値生成信号、６：判定部、６ａ：異常通知信号、７：期待値メモリ、１００：ＩｏＴ機器、１０１：プロセッサ、１０１ａ：実行コード列、１０２：ＣＰＵコア、１０３：実行コード出力インターフェース、１０４：通信ＩＦ部、１０５：割込ＩＦ部、１０６：メモリ、１０７：外部ＩＯ部。

請求の範囲

[請求項1] プロセッサから出力される実行コード列を受け付けて、ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）のユーザープログラム領域の実行アドレスと、カーネル領域の実行アドレスと、を少なくとも識別する実行コード識別部と、

所定のタイミングで前記実行コード識別部による識別結果から得られる所定の特徴値を所定の期待値と照合して所定以上の差がある場合に攻撃があると判定する判定部と、を備え、

前記判定部は、攻撃があると判定すると、前記プロセッサに所定の異常通知信号を通知する、

ことを特徴とする電子機器。

[請求項2] 請求項１に記載の電子機器であって、

前記実行コード識別部は、前記実行コード列を用いて、前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、を少なくとも識別し、

前記判定部は、前記所定の特徴値として前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、を用いる、

ことを特徴とする電子機器。

[請求項3] 請求項１に記載の電子機器であって、

前記実行コード識別部は、前記実行コード列を用いて、前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、前記ＯＳへの割込み発生回数と、を少なくとも識別し、

前記判定部は、前記所定の特徴値として前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、前記ＯＳへの割込み発生回数と、を用いる、

ことを特徴とする電子機器。

[請求項4] 請求項１に記載の電子機器であって、

前記実行コード識別部は、前記実行コード列を用いて、前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、を少なくとも識別し、

前記判定部は、前記所定の特徴値として前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、の比の値を用いる、

ことを特徴とする電子機器。

[請求項5] 請求項１に記載の電子機器であって、

前記実行コード識別部は、前記実行コード列を用いて、前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、前記ＯＳへの割込み発生回数と、を少なくとも識別し、

前記判定部は、前記所定の特徴値として前記ユーザープログラム領域の実行アドレス数と、前記カーネル領域の実行アドレス数と、前記ＯＳへの割込み発生回数と、の比の値を用いる、

ことを特徴とする電子機器。

[請求項6] 請求項１に記載の電子機器であって、

前記プロセッサは、

前記異常通知信号を受信すると、該異常通知信号に応じて所定のリセット処理を行う、

ことを特徴とする電子機器。

[請求項7] 請求項１に記載の電子機器であって、

前記判定部は、前記所定のタイミングとして、前記実行コード列上の前記ＯＳへの割込み発生タイミングを用いる、

ことを特徴とする電子機器。

[請求項8] 請求項１に記載の電子機器であって、

前記判定部は、前記所定のタイミングとして、所定の時間の経過を用いる、

ことを特徴とする電子機器。

[請求項9] 請求項１に記載の電子機器であって、

開始指示を受け付けると、前記識別結果を前記期待値の候補に取り込むことで前記期待値の較正を行う選択部、

を備えることを特徴とする電子機器。

[請求項10] 請求項１に記載の電子機器であって、

前記プロセッサは、

所定のセンサーと、

接続された所定のネットワーク上の他の装置と、前記センサーにより得られた情報を含む情報を送受信する通信部と、

を備えることを特徴とする電子機器。

[請求項11] 車載のＥＣＵ（Ｅｌｅｃｔｒｏｎｉｃ Ｃｏｎｔｒｏｌ Ｕｎｉｔ）であって、

ＣＰＵ（Ｃｅｎｔｒａｌ Ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）コア群から出力される実行コード列を受け付けて、ＯＳ（Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）のユーザープログラム領域の実行アドレスと、カーネル領域の実行アドレスと、を少なくとも識別する実行コード識別部と、

所定のタイミングで前記実行コード識別部による識別結果から得られる所定の特徴値を所定の期待値と照合して所定以上の差がある場合に攻撃があると判定する判定部と、

前記攻撃があると判定されたＣＰＵコアを前記ＣＰＵコア群から隔離する所定の信号を通知するコア管理部と、

を備えることを特徴とするＥＣＵ。

[請求項12] 電子機器の攻撃検知方法であって、

前記電子機器は、

プロセッサから出力される実行コード列を受け付けて、ＯＳ（Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）のユーザープログラム領域の実行アドレスと、カーネル領域の実行アドレスと、を少なくとも識別する実

行コード識別処理と、

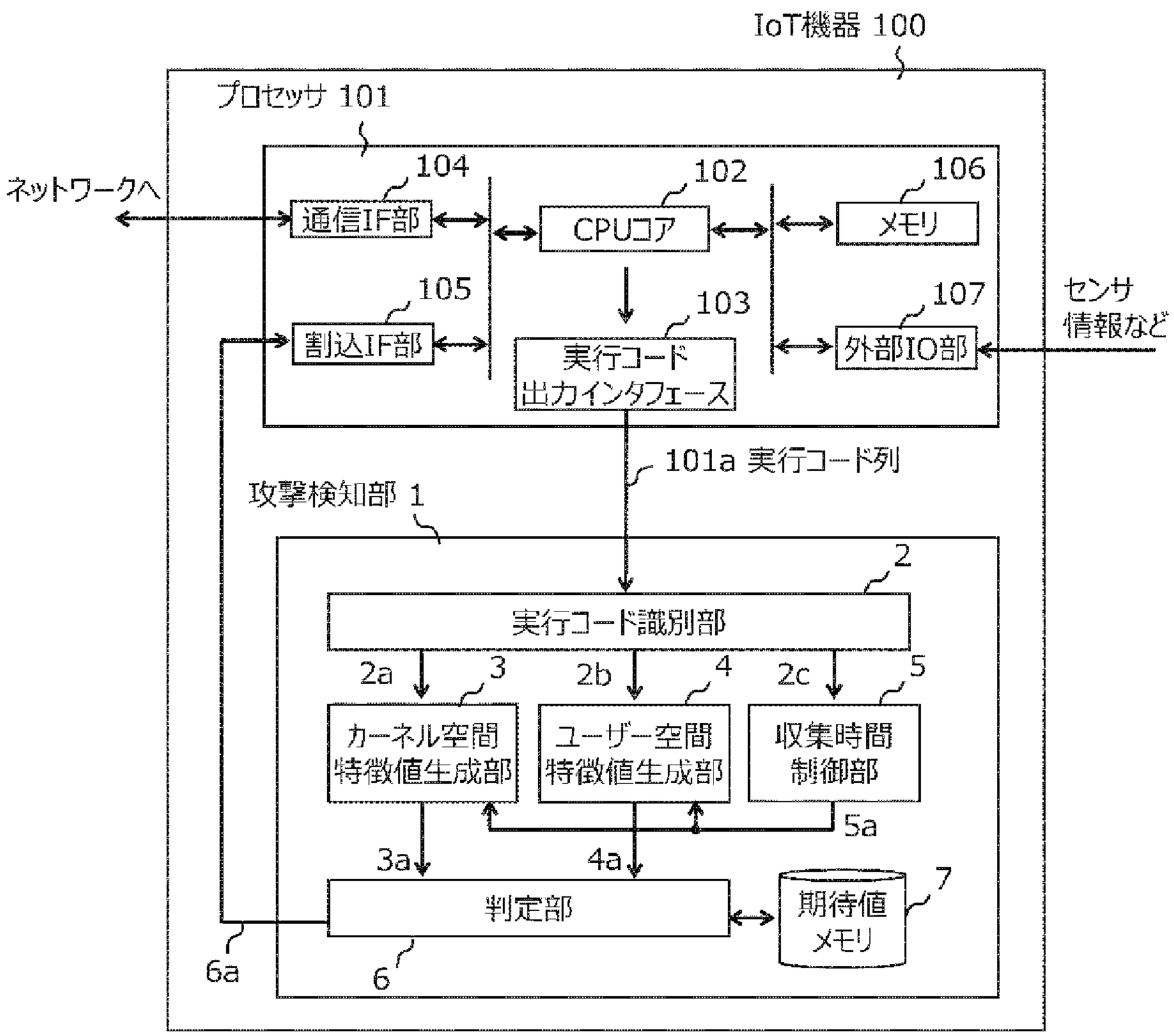
所定のタイミングで前記実行コード識別処理による識別結果から得られる所定の特徴値を所定の期待値と照合して所定以上の差がある場合に攻撃があると判定する判定処理と、

攻撃があると判定すると、前記プロセッサに所定の異常通知信号を通知する通知処理と、

を実施することを特徴とする電子機器の攻撃検知方法。

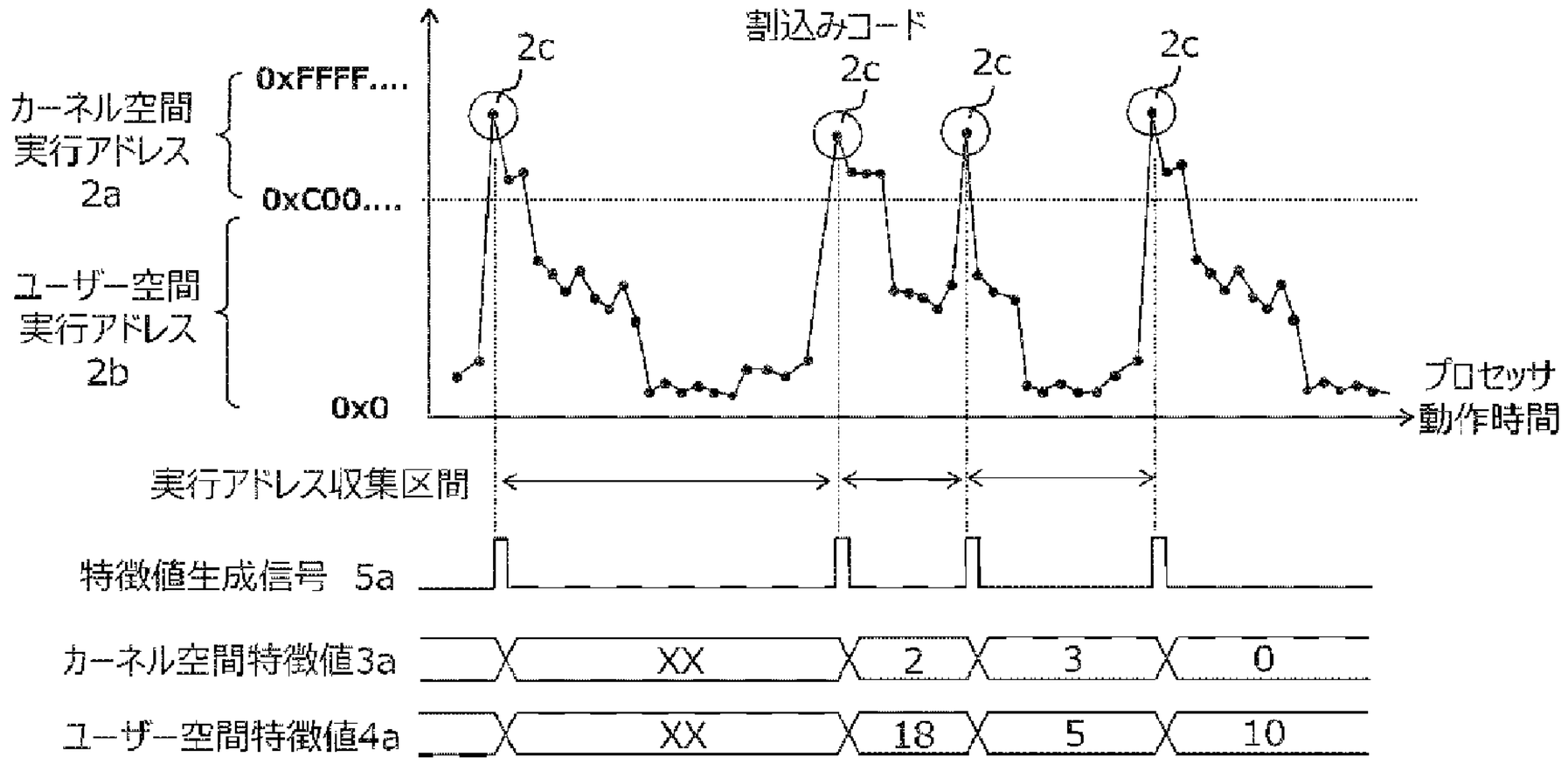
[図1]

図 1


IoT機器 100
プロセッサ 101
ネットワークへ
104
通信IF部
102
CPUコア
106
メモリ
105
割込IF部
103
実行コード
出力インタフェース
107
外部IO部
センサ
情報など
101a 実行コード列
攻撃検知部 1
2
実行コード識別部
2a
3
2b
4
2c
5
カーネル空間
特徴値生成部
ユーザー空間
特徴値生成部
収集時間
制御部
5a
3a
4a
判定部
期待値
メモリ
7
6a
6

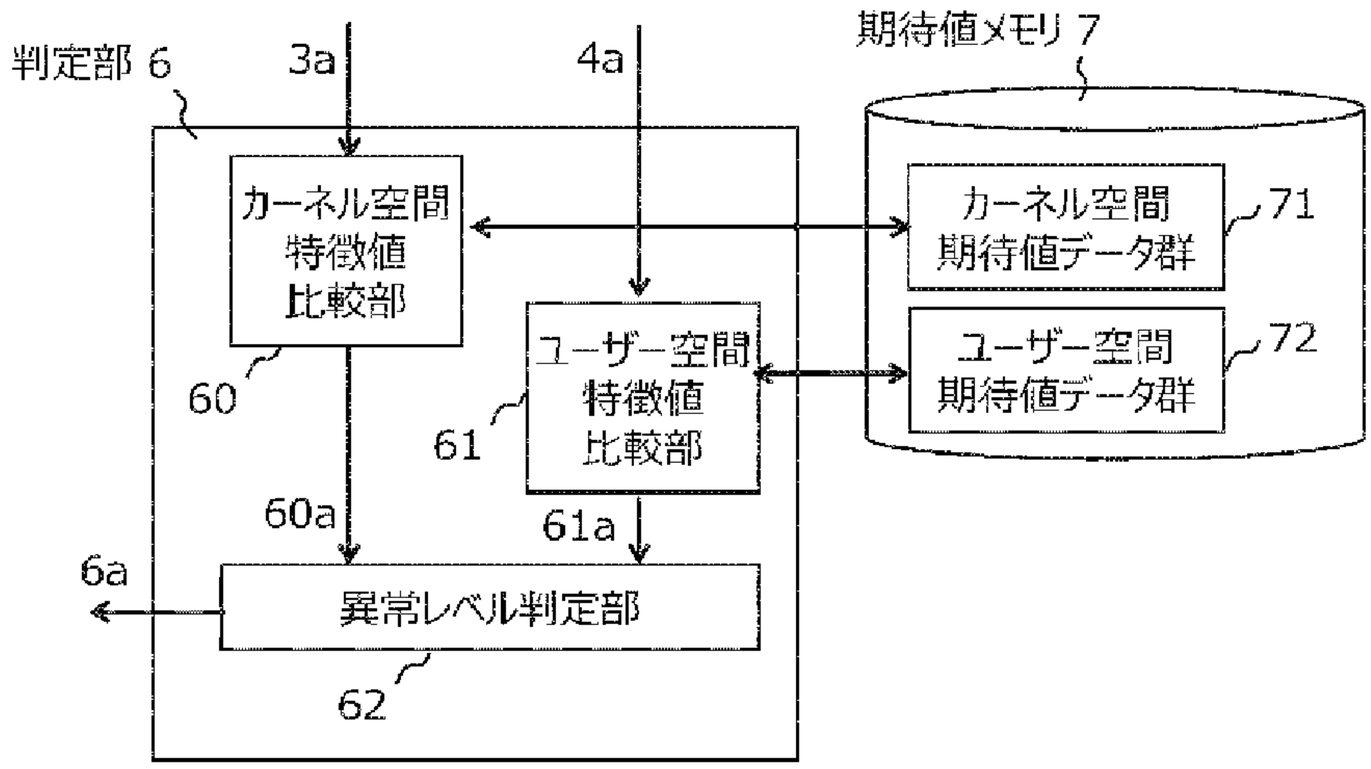
[図2]

図2


仮想アドレス
割込みコード
2c
2c
2c
2c
0xFFFF....
0xC00....
0x0
カーネル空間
実行アドレス
2a
ユーザー空間
実行アドレス
2b
プロセッサ
動作時間
実行アドレス収集区間
特徴値生成信号 5a
カーネル空間特徴値3a
XX
2
3
0
ユーザー空間特徴値4a
XX
18
5
10

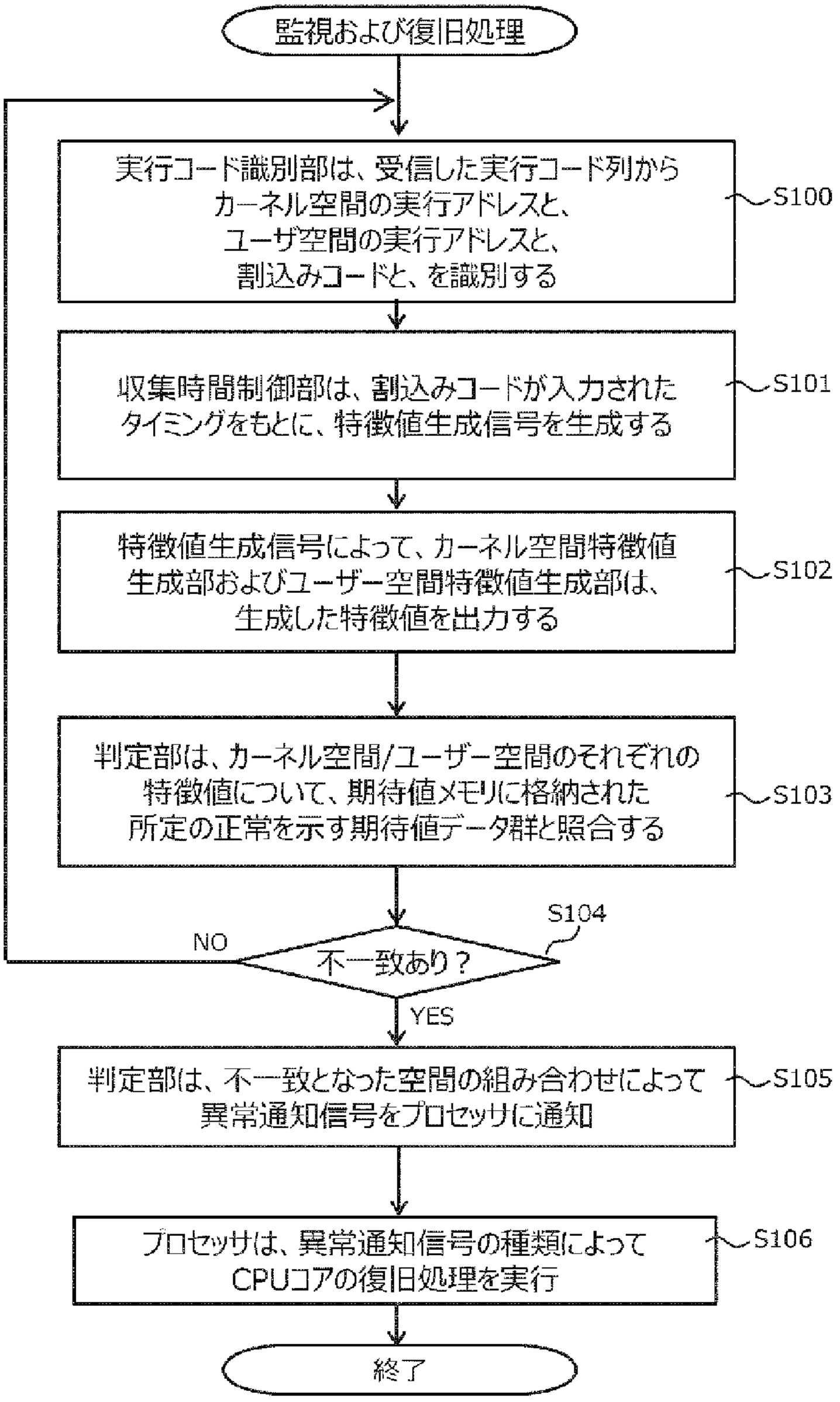
[図3]

図3


判定部 6
3a
4a
期待値メモリ 7
カーネル空間
特徴値
比較部
60
ユーザー空間
特徴値
比較部
61
カーネル空間
期待値データ群
71
ユーザー空間
期待値データ群
72
60a
61a
6a
異常レベル判定部
62

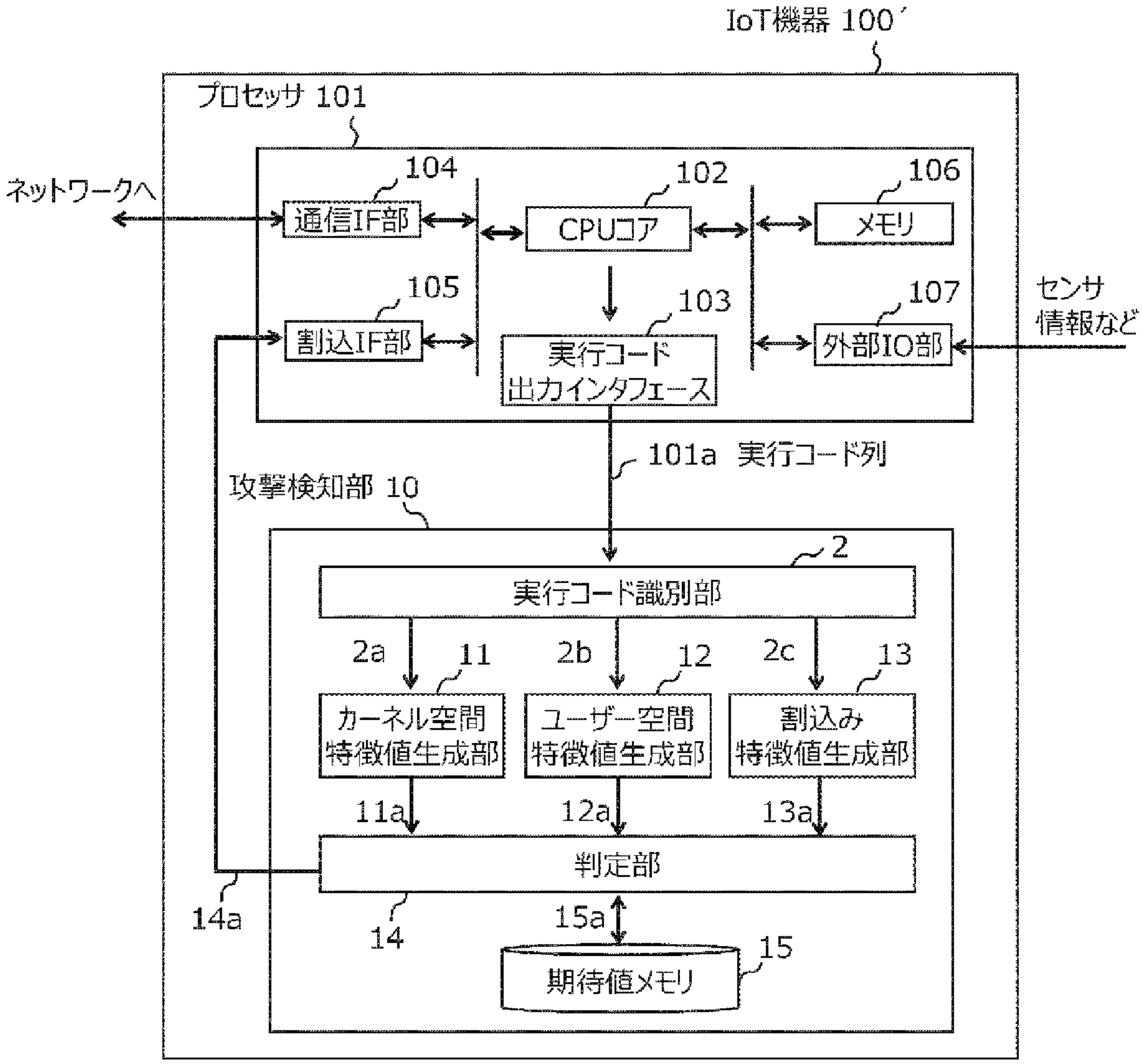
[図4]

図4


監視および復旧処理
実行コード識別部は、受信した実行コード列から
カーネル空間の実行アドレスと、
ユーザ空間の実行アドレスと、
割込みコードと、を識別する
S100
収集時間制御部は、割込みコードが入力された
タイミングをもとに、特徴値生成信号を生成する
S101
特徴値生成信号によって、カーネル空間特徴値
生成部およびユーザー空間特徴値生成部は、
生成した特徴値を出力する
S102
判定部は、カーネル空間/ユーザー空間のそれぞれの
特徴値について、期待値メモリに格納された
所定の正常を示す期待値データ群と照合する
S103
S104
NO
不一致あり？
YES
判定部は、不一致となった空間の組み合わせによって
異常通知信号をプロセッサに通知
S105
プロセッサは、異常通知信号の種類によって
CPUコアの復旧処理を実行
S106
終了

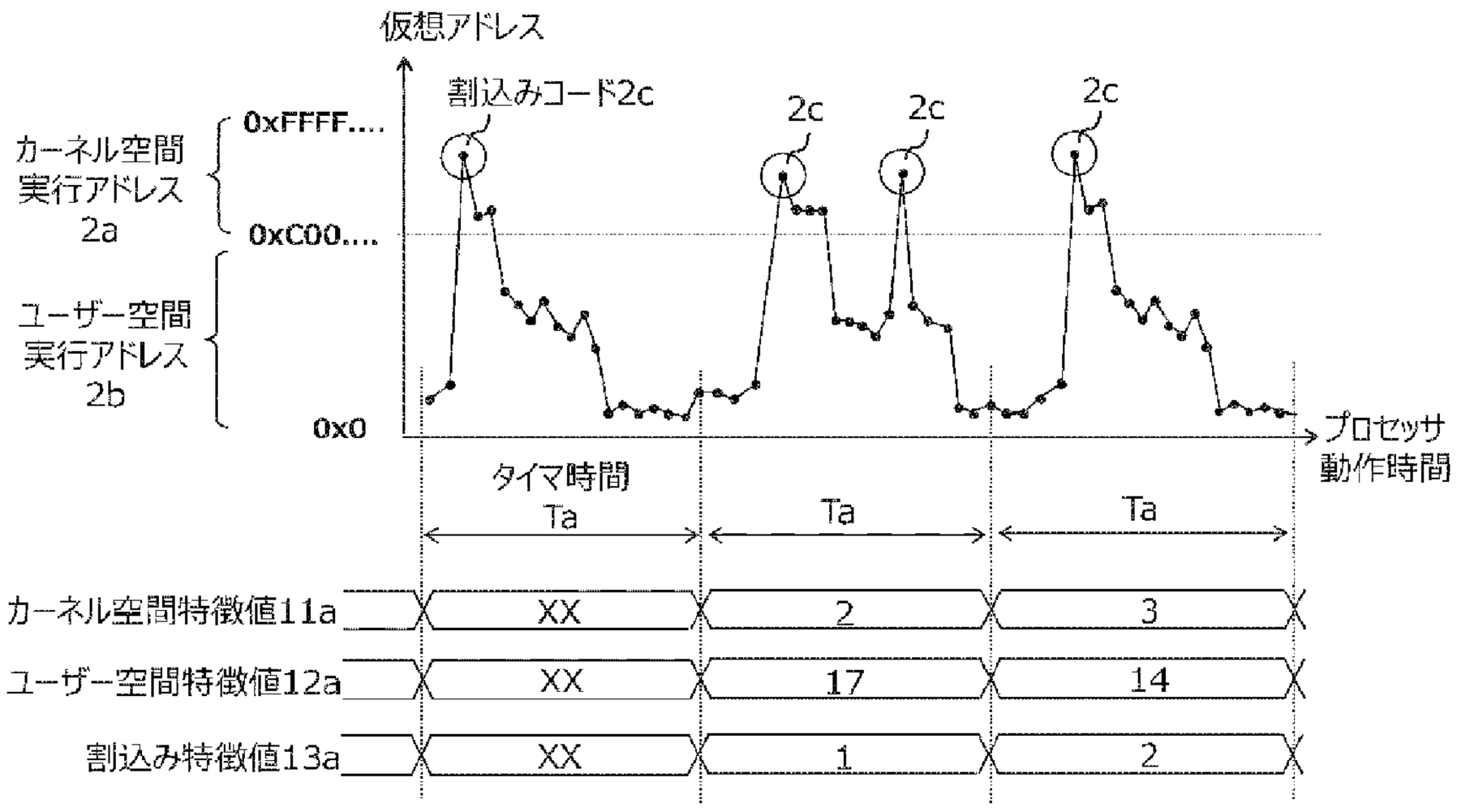
[図5]

図 5


IoT機器 100´
プロセッサ 101
ネットワークへ
104
通信IF部
102
CPUコア
106
メモリ
105
割込IF部
103
実行コード
出力インタフェース
107
外部IO部
センサ
情報など
101a 実行コード列
攻撃検知部 10
2
実行コード識別部
2a
11
カーネル空間
特徴値生成部
2b
12
ユーザー空間
特徴値生成部
2c
13
割込み
特徴値生成部
11a
12a
13a
判定部
14a
14
15a
15
期待値メモリ

[図6]

図6


仮想アドレス
割込みコード2c
2c
2c
2c
0xFFFF....
カーネル空間
実行アドレス
2a
0xC00....
ユーザー空間
実行アドレス
2b
0x0
プロセッサ
動作時間
タイマ時間
Ta
Ta
Ta
カーネル空間特徴値11a
XX
2
3
ユーザー空間特徴値12a
XX
17
14
割込み特徴値13a
XX
1
2

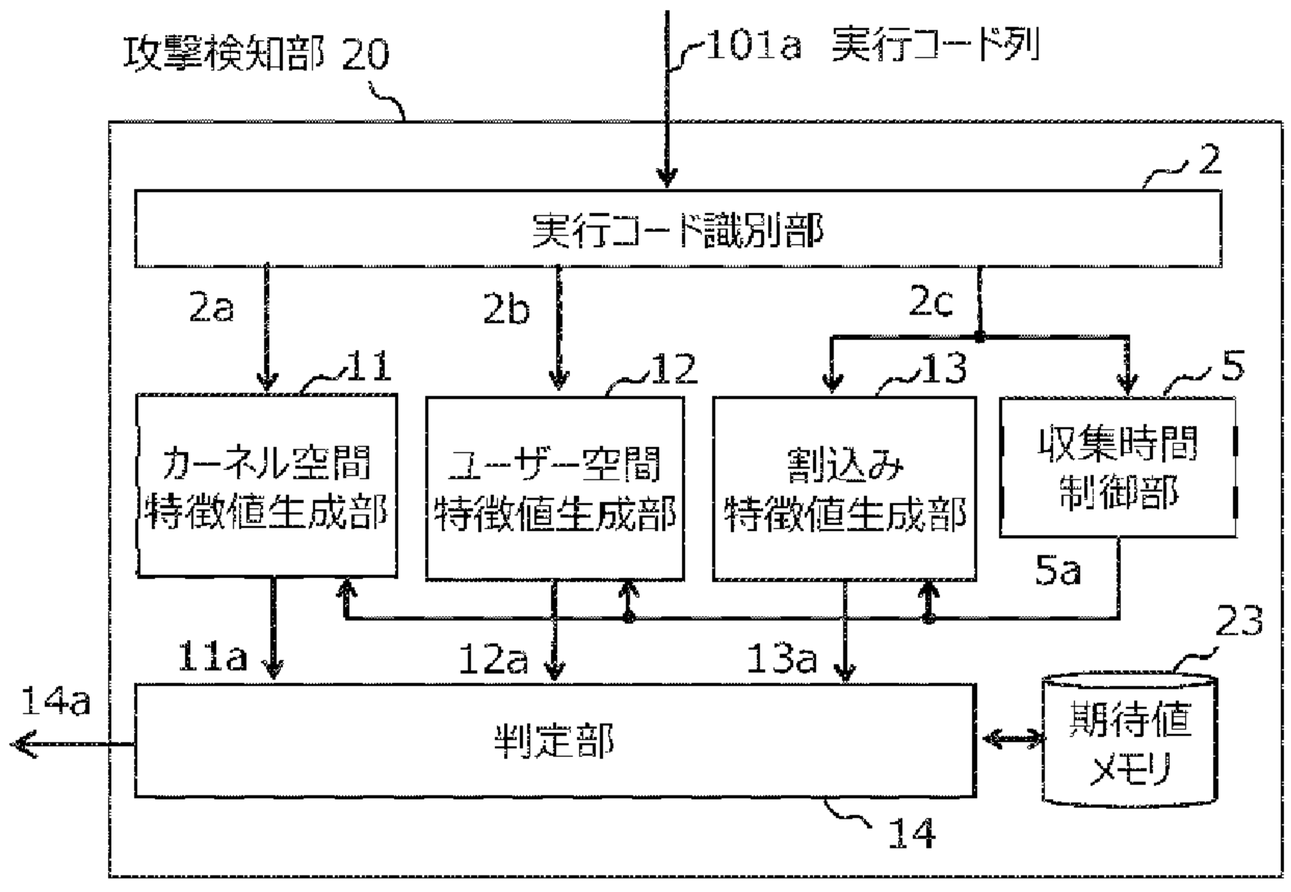
[図7]

図 7

監視および復旧処理
実行コード識別部は、受信した実行コード列から
カーネル空間の実行アドレスと、
ユーザ空間の実行アドレスと、
割込みコードと、を識別する
S200
実行コード識別部は、識別した3種類の情報
(カーネル空間の実行アドレスと、ユーザー空間の
実行アドレスと、割込みコード)を内部計時部に
設定された時間区間（任意時間Ｔａ）で収集する
S201
各特徴値の生成部は、収集した情報(カーネル空間の
実行アドレスと、ユーザー空間の実行アドレスと、割込み
コード)の特徴値をそれぞれ生成する
S202
判定部は、各特徴値について、期待値メモリに
格納された期待値データ群と照合する
S203
S204
NO
不一致あり？
YES
判定部は、不一致となった空間の組み合わせによって
異常通知信号をプロセッサに通知
S205
プロセッサは、異常通知信号の種類によって
ＣＰＵコアの復旧処理を実行
S206
終了

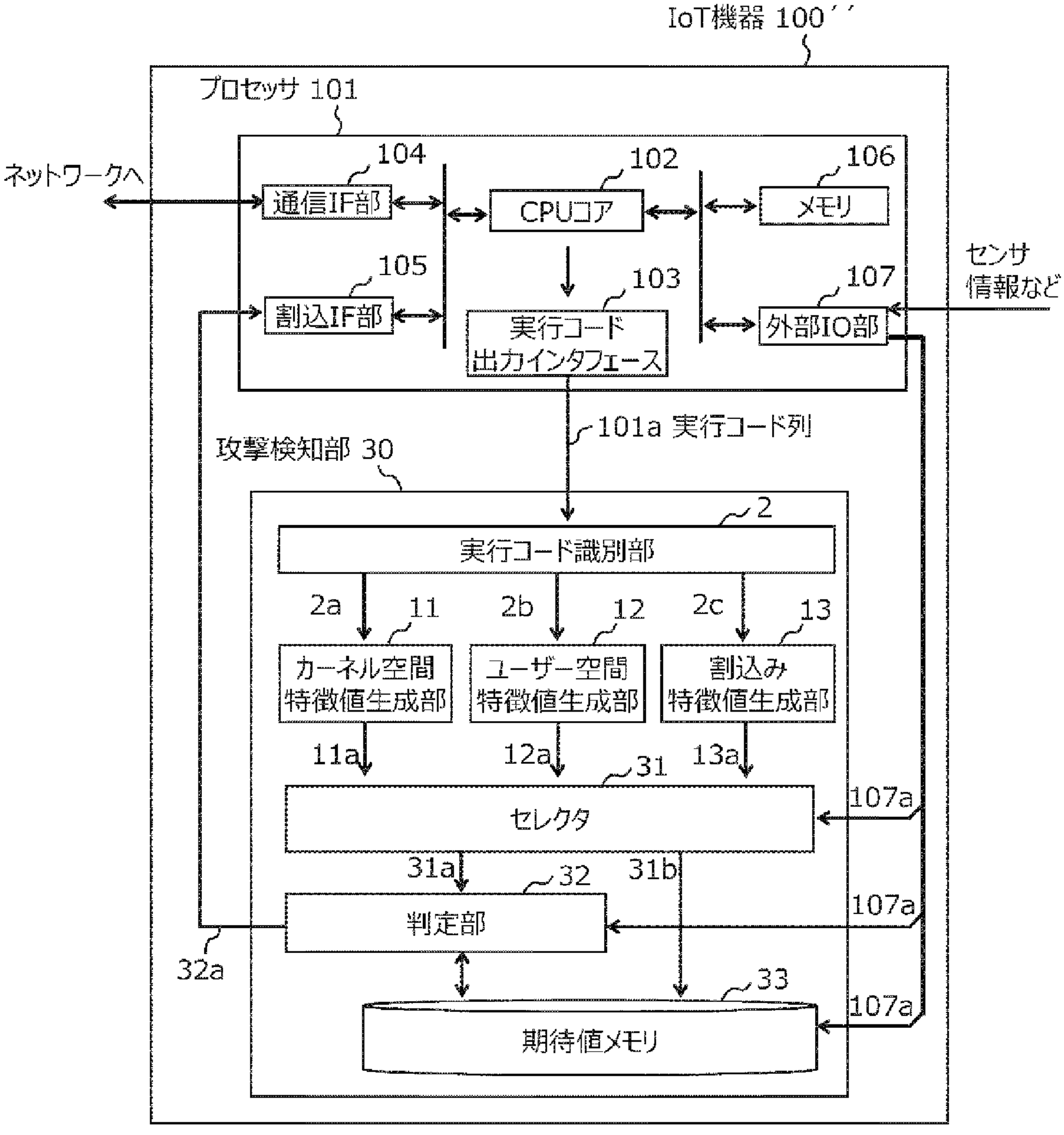
[図8]

図8


攻撃検知部 20
101a 実行コード列
2
実行コード識別部
2a
2b
2c
11
12
13
5
カーネル空間
特徴値生成部
ユーザー空間
特徴値生成部
割込み
特徴値生成部
収集時間
制御部
5a
11a
12a
13a
23
14a
判定部
期待値
メモリ
14

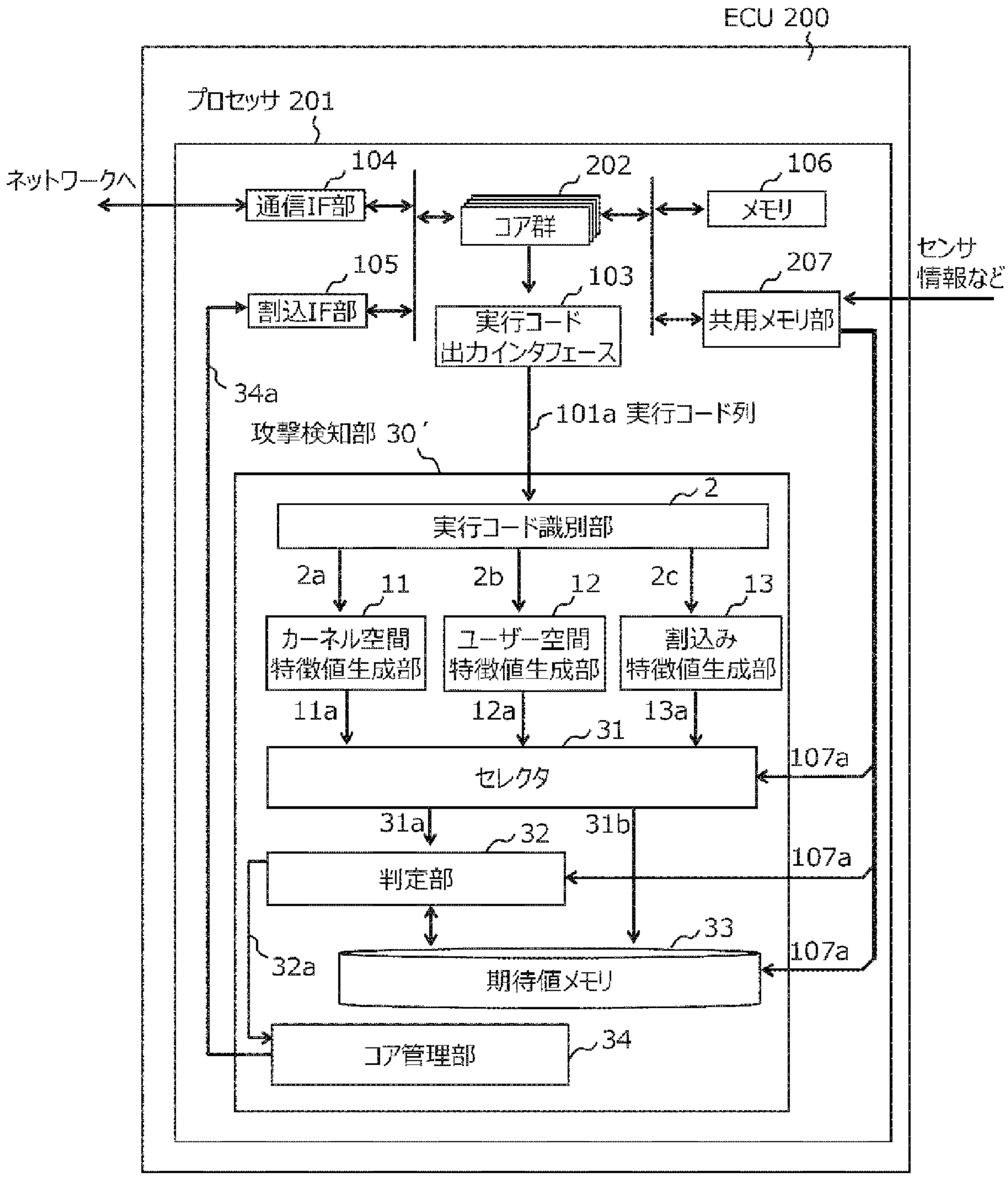
[図9]

図9


IoT機器 100´´
プロセッサ 101
ネットワークへ
104
通信IF部
102
CPUコア
106
メモリ
105
割込IF部
103
実行コード
出力インタフェース
107
外部IO部
センサ
情報など
101a 実行コード列
攻撃検知部 30
2
実行コード識別部
2a
11
カーネル空間
特徴値生成部
2b
12
ユーザー空間
特徴値生成部
2c
13
割込み
特徴値生成部
11a
12a
13a
31
セレクタ
107a
31a
32
判定部
31b
32a
33
期待値メモリ

[図10]

図１０


ECU 200
プロセッサ 201
ネットワークへ
104
通信IF部
202
コア群
106
メモリ
センサ
情報など
105
割込IF部
103
実行コード
出力インタフェース
207
共用メモリ部
34a
101a 実行コード列
攻撃検知部 30
2
実行コード識別部
2a
11
2b
12
2c
13
カーネル空間
特徴値生成部
ユーザー空間
特徴値生成部
割込み
特徴値生成部
11a
12a
13a
31
セレクタ
107a
31a
31b
32
判定部
107a
33
32a
期待値メモリ
107a
コア管理部
34

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2020/045419

A. CLASSIFICATION OF SUBJECT MATTER
G06F 21/52(2013.01)i
FI: G06F21/52

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)
G06F21/52

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

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2021
Registered utility model specifications of Japan	1996-2021
Published registered utility model applications of Japan	1994-2021

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2010-182196 A (KDDI CORP.) 19 August 2010 (2010-08-19) paragraph [0039]	1-12
A	JP 2015-99466 A (NIPPON TELEGRAPH AND TELEPHONE CORP.) 28 May 2015 (2015-05-28) paragraphs [0018]-[0033]	1-12
A	大平 修慈 ほか，車載 LAN へ侵入するマルウェアの証拠保全を行うカーネル上のフォレンジック機構，情報処理学会論文誌，15 March 2019, vol. 60, no. 3, pp. 791-802, in particular, section 3, (OHIRA, Shuji et al., "Forensics Mechanism in Kernel Land to Preserve Evidence of Malware Intruding into In-vehicle LAN", Transactions of Information Processing Society of Japan)	1-12

☐ 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
16 February 2021 (16.02.2021)	22 February 2021 (22.02.2021)

Name and mailing address of the ISA/	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2020/045419

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2010-182196 A	19 Aug. 2010	(Family: none)	
JP 2015-99466 A	28 May 2015	(Family: none)	

国際調査報告

国際出願番号
PCT/JP2020/045419

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

G06F 21/52(2013.01)i
FI: G06F21/52

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

G06F21/52

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報　１９２２－１９９６年
日本国公開実用新案公報　１９７１－２０２１年
日本国実用新案登録公報　１９９６－２０２１年
日本国登録実用新案公報　１９９４－２０２１年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C.　関連すると認められる文献

引用文献のカテゴリー*	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	JP 2010-182196 A（ＫＤＤＩ株式会社）19.08.2010（2010-08-19） 段落0039	1-12
A	JP 2015-99466 A（日本電信電話株式会社）28.05.2015（2015-05-28） 段落0018-0033	1-12
A	大平 修慈 ほか，車載ＬＡＮへ侵入するマルウェアの証拠保全を行うカーネル上のフォレンジック機構，情報処理学会論文誌，2019.03.15，第60巻，第3号，pp.791-802 特に、3節	1-12

☐ Ｃ欄の続きにも文献が列挙されている。

☑ パテントファミリーに関する別紙を参照。

*　引用文献のカテゴリー
“A”　特に関連のある文献ではなく、一般的技術水準を示すもの
“E”　国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
“L”　優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
“O”　口頭による開示、使用、展示等に言及する文献
“P”　国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
“T”　国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの
“X”　特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
“Y”　特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
“&”　同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
16.02.2021	22.02.2021

名称及びあて先
日本国特許庁(ISA/JP)
〒100-8915
日本国
東京都千代田区霞が関三丁目４番３号

権限のある職員（特許庁審査官）
行田　悦資　5S　6304
電話番号　03-3581-1101　内線　3546

国際調査報告
パテントファミリーに関する情報

国際出願番号
PCT/JP2020/045419

引用文献	公表日	パテントファミリー文献	公表日
JP 2010-182196 A	19.08.2010	（ファミリーなし）	
JP 2015-99466 A	28.05.2015	（ファミリーなし）	