(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2012-14667
(P2012-14667A)
(43)公開日　平成24年1月19日(2012.1.19)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 13/00	(2006.01)	G06F	13/00	351Z	5B089
H04L 12/66	(2006.01)	H04L	12/66	B	5B276
G06F 21/20	(2006.01)	G06F	15/00	330A	5B285
G06F 21/22	(2006.01)	G06F	9/06	660J	5K030

審査請求 有　請求項の数 6　ＯＬ　（全 15 頁）

(21)出願番号　特願2010-178803 (P2010-178803)
(22)出願日　平成22年8月9日 (2010.8.9)
(31)優先権主張番号　10-2010-0064363
(32)優先日　平成22年7月5日 (2010.7.5)
(33)優先権主張国　韓国 (KR)

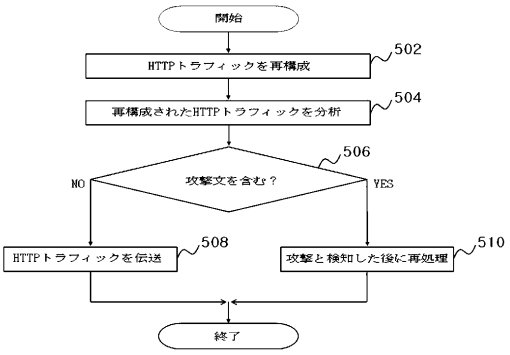
（特許庁注：以下のものは登録商標）
１．ＪａｖａＳｃｒｉｐｔ

(71)出願人　505112037
ペンタ・セキュリティ・システムズ・インコーポレーテッド
大韓民国ソウル特別市永登浦区汝矣島洞２５－１１　ハンジン・シッピング・ビルディング２０階
(74)代理人　100071054
弁理士　木村　高久
(72)発明者　キム、ドゥクースー
大韓民国、ソウル、セオダエムン－グ、１－１８２８　ブグヒェオン　３－ドング、グン－イル　テックヴィル　３０２

最終頁に続く

(54)【発明の名称】ウェブアプリケーション攻撃の検知方法

(57)【要約】　（修正有）
【課題】受信されるＨＴＴＰトラフィックのパケットからペイロードのみを分離してＨＴＴＰトラフィックを再構成した後、該再構成されたＨＴＴＰトラフィックの内容をパーサーにて分析することで、攻撃に関連した内容が含まれているか否かを判断することができる、ウェブアプリケーション攻撃の検知方法を提供する。
【解決手段】ＨＴＴＰトラフィックを形成するパケットが受信されると、ウェブアプリケーション・ファイアウォールが、ＨＴＴＰトラフィックを再構成して、分析を行い、該再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいないと判断した場合、再構成されたＨＴＴＰトラフィックをウェブサーバまたは使用者サーバへ伝送して正常に処理されるようにし、再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいると判断した場合、再構成されたＨＴＴＰトラフィックを攻撃であると検知した後、再処理を行う。
【選択図】図３



【特許請求の範囲】
【請求項１】
ＨＴＴＰトラフィックを形成するパケットが受信されると、ウェブアプリケーション・ファイアウォールが、前記ＨＴＴＰトラフィックを形成する各パケットのヘッダーを除去した後、前記各パケットのペイロード部分のみを集めて、前記ＨＴＴＰトラフィックを再構成するステップと、
再構成されたＨＴＴＰトラフィックを分析することで、該再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいるか否かをパーサーにて判断するステップと、
前記判断の結果、前記再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいないと、前記再構成されたＨＴＴＰトラフィックをウェブサーバまたは使用者サーバへ伝送して正常に処理されるようにするステップと、
前記判断の結果、前記再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいると、前記再構成されたＨＴＴＰトラフィックを攻撃と検知した後、前記再構成されたＨＴＴＰトラフィックに含まれている正常でないパケットを送信したウェブサーバまたは使用者サーバに対して前記正常でないパケットに対応するパケットの再伝送を要請する、または前記パケットを削除する、または前記再構成されたＨＴＴＰトラフィックに含まれている正常ではないパケットを変調して前記ウェブサーバまたは使用者サーバへ伝送する方式のいずれかにて再処理するステップと、
を含むウェブアプリケーション攻撃の検知方法。
【請求項２】
前記パーサーはＸＭＬパーサーを含み、前記ＸＭＬパーサーは、前記再構成されたＨＴＴＰトラフィックに対し、Ｔａｇの始端と終端を把握してＸＭＬ構文の整合性と、上・下位概念を把握することで、前記再構成されたＨＴＴＰトラフィックに攻撃文が含まれているか否かを判断することを特徴とする請求項１に記載のウェブアプリケーション攻撃の検知方法。
【請求項３】
前記パーサーはＪａｖａＳｃｒｉｐｔパーサーを含み、前記ＪａｖａＳｃｒｉｐｔパーサーは、ＪａｖａＳｃｒｉｐｔ構文の有効性の有無を把握することで、前記再構成されたＨＴＴＰトラフィックに攻撃文が含まれているか否かを判断することを特徴とする請求項１に記載のウェブアプリケーション攻撃の検知方法。
【請求項４】
前記パーサーはＳＱＬパーサーを含み、前記ＳＱＬパーサーは、前記再構成されたＨＴＴＰトラフィックを最小単位に分解し、各結果がＳＱＬ構文の一部分であるか否かをチェックすることで、前記再構成されたＨＴＴＰトラフィックに攻撃文が含まれているか否かを判断することを特徴とする請求項１に記載のウェブアプリケーション攻撃の検知方法。
【請求項５】
前記変調においては、
前記ウェブアプリケーション・ファイアウォールが、前記再構成されたＨＴＴＰトラフィックに含まれている攻撃と疑われ得るメッセージを正常なメッセージに変調することを特徴とする請求項１に記載のウェブアプリケーション攻撃の検知方法。
【請求項６】
前記変調においては、
前記ウェブアプリケーション・ファイアウォールが、前記再構成されたＨＴＴＰトラフィックに含まれているメッセージのうち、個人情報に関連したメッセージの一部を外部から読み取り不可能なメッセージに変調することを特徴とする請求項１に記載のウェブアプリケーション攻撃の検知方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ウェブアプリケーション攻撃を検知する方法に関する。

【背景技術】
【０００２】
従来のウェブアプリケーション・ファイアウォール（Ｗｅｂ Ａｐｐｌｉｃａｔｉｏｎ Ｆｉｒｅｗａｌｌ）（以下、「ＷＡＦ」と略称する）では、ＯＳＩのネットワーク分類基準のうちのレイヤー４での攻撃検知を行う侵入検知システム（Ｉｎｔｒｕｓｉｏｎ Ｄｅｔｅｃｔｉｏｎ Ｓｙｓｔｅｍ、ＩＤＳ）または侵入防止システム（Ｉｎｔｒｕｓｉｏｎ Ｐｒｏｔｅｃｔｉｏｎ Ｓｙｓｔｅｍ、ＩＰＳ）を基盤として、ＯＳＩネットワーク分類の最上位階層に該当するレイヤー７を対象とする攻撃を防御しており、このため、防御に限界が生じていた。
【０００３】
図１は、一般的なＯＳＩ ７レイヤーを説明するための例示図である。
【０００４】
ＯＳＩ ７レイヤーとは、図１に示すように、Ａｐｐｌｉｃａｔｉｏｎ、Ｐｒｅｓｅｎｔａｔｉｏｎ、Ｓｅｓｓｉｏｎ、Ｔｒａｎｓｐｏｒｔ、Ｎｅｔｗｏｒｋ、Ｄａｔａ Ｌｉｎｋ及びＰｈｙｓｉｃａｌの７階層から構成されたことをいい、前述したように、レイヤー７水準を対象とした攻撃を検知し防御するウェブアプリケーション・ファイアウォール（ＷＡＦ）がレイヤー４水準で攻撃を検知し防御する理由は、次のとおりである。
【０００５】
先ず、従来より、攻撃の検知のために一般に用いられている侵入検知システム（ＩＤＳ）または侵入防止システム（ＩＰＳ）などのシステムが、過去の特定インターネット・プロトコル・アドレス（ＩＰアドレス）に対して特定ポートを遮る役割をしていたネットワークファイアウォールの役割を、パケットの分析に拡張しようとする試みから発明されたものであるため、過去のネットワークファイアウォールが攻撃を検知していた水準のレイヤー４に止まるようになったわけである。
【０００６】
また、従来のウェブアプリケーション・ファイアウォールがレイヤー４にて攻撃の有無を検知する理由は、ＯＳＩ ７レイヤーモデルにおいて意味のわからない電気信号ではない、意味をもつ最小のデータ単位であるパケットが現われる時点がレイヤー４であることから、最初のデータ単位が成立される時点で攻撃を判断し遮断するために、レイヤー４で検知するわけである。
【０００７】
すなわち、アプリケーション・レイヤー（レイヤー７、Ｌ７）を対象とする攻撃を検知し防御しようとするためには、ネットワーク・トラフィックの分析もレイヤー７水準で行われる必要があり、これにより、誤検知と未検知（攻撃であるのに検知できないこと）を極力抑えることができる知能的なウェブファイアウォールの役割を果たすことができるが、従来は、レイヤー７を対象とする攻撃をレイヤー４水準の検知方法にて検知していたため、正常な検知や防御がなされていなかったという問題点がある。
【０００８】
付言すると、レイヤー４はデータの単位がパケットであって、従来のＩＤＳ、ＩＰＳを根幹として作製された１世代、２世代のＷＡＦは、パケット単位のパターン整合を行うことで当該ネットワーク・トラフィックの攻撃の有無を判断している。すなわち、従来の１、２世代のＷＡＦは、事前に管理者によって登録されている平均５０００個余りの攻撃類型（正規表現：Ｒｅｇｕｌａｒ Ｅｘｐｒｅｓｓｉｏｎ、略語：Ｒｅｇｘ）に対し、パケット毎に１番から５０００番までの攻撃類型と一致する類型があるか否かを検査することで、当該パケットが、攻撃が含まれているパケットであるか正常パケットであるかを判断する。
【０００９】
しかし、最近のＷＡＦは、ディープパケットインスペクション（Ｄｅｅｐ Ｐａｃｋｅｔ Ｉｎｓｐｅｃｔｉｏｎ、ＤＰＩ）という方式を表明し、既存のパケットヘッダーのみを見て攻撃の有無を判別する方式から脱皮して、パケットのペイロード部分も検査する方

式に変化しつつあるが、これは真正な意味のアプリケーション・レイヤー水準の防御ではなく、ただ、従来のレイヤー４水準の防御が若干進化した形態であるといえる。

【００１０】

一方、アプリケーション・レイヤー（レイヤー７）水準の攻撃の検知方法に適用される、レイヤー４水準で行われる前述のような従来の攻撃の検知方法では、次のような４つの不具合をもっている。

【００１１】

第一に、従来の検知方法では、攻撃の類型が変化する度に新しい攻撃類型がアップデートされる必要がある。

【００１２】

第二に、従来の検知方法では、処理速度の問題のため登録可能な攻撃類型の個数が制限されているので（最大一万個）、既存に攻撃と登録されていた類型を周期的に削除する必要がある。

【００１３】

第三に、レイヤー４のパケットパターン整合基盤の従来のＷＡＦでは、攻撃パケットの変調（例えば、ＨＴＭＬタグの変形、削除など、個人情報の特定部分の削除など）が技術的にほぼ不可能である。その不可能の理由は、次のとおりである。すなわち、パケットの変調は、パケットサイズの変化を引き起こすが、従来の１、２世代のＷＡＦが、変化されたパケットのサイズをパケットヘッダーに書き換える作業は非常に多くの演算を要求し、それに伴い、処理時間が増大し、実際のインターネットサービス環境には適用されにくいためである。

【００１４】

第四に、従来の検知方法では、ＨＴＴＰトラフィックの全体をみて攻撃を判断するわけではないため、意味論的にみて攻撃ではないパケットを攻撃パケットと判断する誤検知を引き起し得る。

【発明の概要】

【発明が解決しようとする課題】

【００１５】

本発明は、上記のような問題点を解決するためになされたものであって、その目的は、受信されるＨＴＴＰトラフィックのパケットからペイロードのみを分離してＨＴＴＰトラフィックを再構成した後、該再構成されたＨＴＴＰトラフィックの内容をパーサーにて分析することで、攻撃に関連した内容が含まれているか否かを判断することができる、ウェブアプリケーション攻撃の検知方法を提供することである。

【課題を解決するための手段】

【００１６】

上記目的を達成するための本発明は、ＨＴＴＰトラフィックを形成するパケットが受信されると、ウェブアプリケーション・ファイアウォールが、上記ＨＴＴＰトラフィックを再構成するステップと、再構成されたＨＴＴＰトラフィックを分析することで、該再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいるか否かを判断するステップと、上記判断の結果、上記再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいないと、上記再構成されたＨＴＴＰトラフィックをウェブサーバまたは使用者サーバへ伝送して正常に処理されるようにするステップと、上記判断の結果、上記再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいると、上記再構成されたＨＴＴＰトラフィックを攻撃と検知した後、再処理するステップと、を含む。

【発明の効果】

【００１７】

本発明は、受信されるＨＴＴＰトラフィックのパケットからペイロードのみを分離してＨＴＴＰトラフィックを再構成した後、該再構成されたＨＴＴＰトラフィックの内容をパーサーにて分析することで、攻撃に関連した内容が含まれているか否かを判断することにより、誤検知率を低減することができるという優れた効果を奏する。

【図面の簡単な説明】
【００１８】
【図１】一般的なＯＳＩ　７レイヤーを説明するための例示図である。
【図２】本発明が適用される通信システムの構成を示す例示図である。
【図３】本発明に係るウェブアプリケーション攻撃の検知方法の一実施形態のフローチャートである。
【図４】本発明に係るウェブアプリケーション攻撃の検知方法に適用されるＨＴＴＰトラフィック再構成の意味を説明するための例示図である。
【図５ａ】本発明に適用されるＳＱＬパーサーの機能を説明するための各種の例示図である。
【図５ｂ】本発明に適用されるＳＱＬパーサーの機能を説明するための各種の例示図である。
【図５ｃ】本発明に適用されるＳＱＬパーサーの機能を説明するための各種の例示図である。
【図５ｄ】本発明に適用されるＳＱＬパーサーの機能を説明するための各種の例示図である。
【発明を実施するための形態】
【００１９】
以下、添付の図面を参照して本発明について詳しく説明する。
【００２０】
図２は、本発明が適用される通信システムの構成を示す例示図である。
【００２１】
本発明が適用される通信システムは、図１に示すように、ウェブサイトを運営し使用者らに各種のサービスを提供するためのウェブサーバ２０と、ウェブサーバと通信を行うことでウェブサーバから各種の情報の提供を受けたり、ウェブサーバへ各種の情報を提供したりするために使用者が利用する使用者サーバ３０、及びウェブサーバをネットワークを介して使用者サーバと接続させるとともに、使用者サーバからの攻撃を検知しウェブサーバの機能を保護するためのウェブアプリケーション・ファイアウォール１０と、を含んで構成される。
【００２２】
ここで、使用者サーバは、パーソナルコンピューター（ＰＣ）のような端末機であってもよく、複数のパーソナルコンピューターとネットワークを介して通信を行うサーバであってもよい。
【００２３】
一方、本発明に係るウェブアプリケーション攻撃の検知方法が適用され、ウェブサーバを外部の攻撃から保護するためのウェブアプリケーション・ファイアウォール１０は、図２に示すように、ＸＭＬパーサー１１、ＪａｖａＳｃｒｉｐｔパーサー１２、ＳＱＬパーサー１３を含んでいる。
【００２４】
すなわち、本発明に係るウェブアプリケーション攻撃の検知方法は、ウェブアプリケーション・ファイアウォールが、受信されるＨＴＴＰトラフィックからパケットのヘッダーを除去し、ペイロード部分のみを集めてＨＴＴＰトラフィックを再構成した後、当該トラフィックの意味論的な分析を遂行して攻撃の有無を検知するものであって、次のような長所を持っている。
【００２５】
第一に、本発明では、攻撃の類型が変化する度に新しいパターンを登録する必要がない。
【００２６】
第二に、格納されているパターンという概念がないので、既存の攻撃類型を削除する作業が不要である。

【００２７】
第三に、ＨＴＴＰトラフィックの全体をみて攻撃の有無を判断し、攻撃と判断される場合、再構成ＨＴＴＰトラフィックを変調して伝送することができる。すなわち、住民登録番号の削除とｈｔｍｌ、ＪａｖａＳｃｒｉｐｔ Ｔａｇの変調が可能である。
【００２８】
第四に、パケットのみをみて攻撃類型を判断するのではなく、再構成されたＨＴＴＰトラフィックの全体からみて意味論的に分析するため、誤検知率を顕著に低減させることができる。
【００２９】
図３は、本発明に係るウェブアプリケーション攻撃の検知方法の一実施形態のフローチャートであり、図４は、本発明に係るウェブアプリケーション攻撃の検知方法に適用されるＨＴＴＰトラフィック再構成の意味を説明するための例示図である。
【００３０】
また、図５ａないし図５ｄは、本発明に適用されるＳＱＬパーサーの機能を説明するための各種の例示図である。
【００３１】
第一の過程として、ウェブアプリケーション・ファイアウォールは、ネットワークを介して外部のサーバと通信を行う途中で、ＨＴＴＰトラフィックを形成するパケットが受信されると、パケットのシーケンス順に並べ、各パケットのヘッダーを除去した後、各パケットのペイロード部分のみを集めて、ＨＴＴＰトラフィックを再構成する（５０２）。すなわち、ＨＴＴＰトラフィックを再構成するということは、パケットのヘッダー部分を分析してシーケンス順にパケットを並べ、ペイロード部分のみを集めることであって、図４に示すように、各パケットをそのシーケンス順に並べた後、パケット４０のペイロード４２部分のみを結合させることをいう。つまり、ＨＴＴＰトラフィックを形成する多数のパケット４０のそれぞれは、図４に示すように、ヘッダー４１とペイロード４２とで構成されているところ、本発明は、各パケットからペイロード部分のみを分離して、ＨＴＴＰトラフィックを再構成している。付言すると、ＨＴＴＰトラフィックは、Ｌ７（レイヤー７）→Ｌ６→Ｌ５→Ｌ４→Ｌ３→Ｌ２→Ｌ１といったように下位階層にいくにつれて、より小さい単位に分けられて宛て先コンピューター（または、サーバ）に到着するようになり、Ｌ４階層でのデータの単位はパケットである。ここで、パケットは、該パケットのシーケンス状態などの情報が含まれたパケットヘッダー（以下、簡単に「ヘッダー」とする）と、小さい単位に分けられているＬ７階層の原文の一部が含まれているパケットペイロード（以下、簡単に「ペイロード」とする）部分とに分けられており、本発明は、各パケットのペイロード部分のみを再構成しているという特徴を持っている。
【００３２】
第二及び第三の過程として、ウェブアプリケーション・ファイアウォールは、再構成されたＨＴＴＰトラフィック５０を分析し（５０４）、該再構成されたＨＴＴＰトラフィックが攻撃に関連した内容を含んでいるか否かを判断する（５０６）。このとき、ウェブアプリケーション・ファイアウォールは、図２に示すように、各種のパーサーにてＨＴＴＰトラフィックを分析することにより、攻撃の有無を判断する。
【００３３】
すなわち、本発明に適用されるウェブアプリケーション・ファイアウォールは、ウェブサイトを運営するウェブサーバを攻撃から防御することが目的であって、ウェブサイトが存在するのに必要な要素は、大きくＸＭＬ、ＪａｖａＳｃｒｉｐｔ、ＳＱＬであるところ、本発明に係るウェブアプリケーション攻撃の検知方法が適用されるウェブアプリケーション・ファイアウォールもまた、ＸＭＬパーサー、ＪａｖａＳｃｒｉｐｔパーサー、ＳＱＬパーサーの３つの要素から構成されることが好ましく、パーサーの種類は、ウェブサイトの標準変化に応じて多様に変化され得る。
【００３４】
ここで、ＸＭＬは、ＤＨＴＭＬ、ＨＴＭＬの上位要素であって、Ｔａｇを基盤に文書の

整合性と上・下位概念を保障するマークアップ言語（Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ）であり、ＸＭＬパーサーは、再構成されたＨＴＴＰトラフィックに対してＴａｇの始端と終端を把握することでＸＭＬ構文の整合性（Ｉｎｔｅｇｒｉｔｙ）と、上・下位概念を把握するパーサーであって、再構成されたＨＴＴＰトラフィックに攻撃に関連した内容が含まれているか否かを判断する機能を遂行する。

【００３５】

一方、ＪａｖａＳｃｒｉｐｔパーサーは、コンピュータープログラミング言語（Ｃ言語やｊａｖａ、ｐｈｙｔｏｎなど）の一種であるＪａｖａＳｃｒｉｐｔを分析し、コンピュータが理解できる形態である二進数に変換する機能を遂行するものであって、ＪａｖａＳｃｒｉｐｔパーサーは、国際標準機関のＥＣＭＡで策定したＪａｖａＳｃｒｉｐｔ文法の標準に従うものであって、この文法に従わない場合、当該ＪａｖａＳｃｒｉｐｔ構文はコンピュータで正常に解釈できずにエラーを発生させる。従来のＷＡＦでは、ＪａｖａＳｃｒｉｐｔ構文を分析せずに、ＪａｖａＳｃｒｉｐｔ構文が始まることを知らせるＴａｇである<ｓｃｒｉｐｔ>Ｔａｇの有無によってＪａｖａＳｃｒｉｐｔを利用した攻撃文であるか否かを判断していた。しかし、本発明では、ＥＭＣＡ－２６２標準のＪａｖａＳｃｒｉｐｔパーサーを（解読器）を利用して当該ＪａｖａＳｃｒｉｐｔ構文が有効な構文であるか否かを把握する。また、従来のＬ４検知位置では、ＪａｖａＳｃｒｉｐｔ ＨＴＴＰトラフィックの全体を把握することができないため、ＪａｖａＳｃｒｉｐｔ構文の有効性の有無を把握することができる方法がなかったが、本発明では、前述したように、ＨＴＴＰトラフィックを再構成する一方、ＪａｖａＳｃｒｉｐｔパーサーを利用して再構成されたＨＴＴＰトラフィックを分析することでＪａｖａＳｃｒｉｐｔ構文の有効性の有無を把握することができる。すなわち、ＪａｖａＳｃｒｉｐｔパーサーは、ＥＭＣＡ－２６２標準を守るＪａｖａＳｃｒｉｐｔ文法を検査することでＪａｖａＳｃｒｉｐｔ構文が有効であるか否かを判断する機能を遂行する。

【００３６】

また、ＳＱＬパーサーは、再構成されたＨＴＴＰトラフィックを最小単位に分解し、各結果がＳＱＬ構文の一部分であるか否かをチェックすることで、ＨＴＴＰトラフィックに攻撃文が含まれているか否かを判断する機能を遂行する。ＳＱＬパーサーの機能を、図５ａないし図５ｄを参考して説明すれば、次のとおりである。すなわち、ＳＱＬパーサーを利用した攻撃検知の例として、ＳＱＬインジェクション攻撃文が（ｎａｍｅ＝“ｐｅｎｔａ”ｏｒ ｎａｍｅ＝“ｓｅｃｕｒｉｔｙ”）ａｎｄ ｋｅｙｗｏｒｄ＝“ｐｅｎｔａｓｅｃ”である場合、ＳＱＬパーサーは、上記ＳＱＬインジェクション攻撃文を図５ａに示すように、ＳＱＬ文法の最小単位に分解し、最小単位毎に攻撃の有無を検知するようになる。このとき、最小単位の結果がいずれもＳＱＬコマンドの一部である場合、当該文章の全体がＳＱＬ文章であると判断する。これに対し、従来の技術を適用したウェブアプリケーション・ファイアウォール（ＷＡＦ）は、図５ｂに示すように、多様なパターン（シグナチャー）を予め登録しておく方法を用いるものであって、ＳＱＬインジェクション攻撃文が‘ａ’＝‘ａ’ ｔｏ ‘ｂ’＝‘ｂ’のように変更されたとき、これを防御することができないという問題点を持っている。また、前述したように多様なパターン（シグナチャー）を予め登録しておく方法を用いる従来のＷＡＦでは、図５ｃに示すようなパターン（シグナチャー）を予め登録しておいた場合、たとえ使用者がサーバへ伝送するリクエストＨＴＴＰトラフィックに“…ｈａｖｉｎｇ ａ ｇｏｏｄ ｔｉｍｅ… ＝＝…”のような文句が含まれていると、Ｈａｖｉｎｇという単語に＝＝表示が続くことでＳＱＬインジェクション攻撃文と判断してしまい、誤検知をすることもあるという問題点を持っている。

【００３７】

すなわち、ＸＭＬパーサーは、ＨＴＴＰトラフィックを再構成して分析を行い、ＳＱＬパーサーは、攻撃文を最小単位に分解し、各結果がＳＱＬの一部分であるか否かを分析することで攻撃の有無を検知するという特徴を持っている。

【００３８】

第四の過程として、上記判断の結果（５０６）、攻撃に関連した内容を含んでいないと

、ウェブアプリケーション・ファイアウォールは、再構成されたＨＴＴＰトラフィックをウェブサーバへ伝送するか、またはネットワークを介して使用者サーバへ伝送し、正常に処理されるようにする（５０８）。

【００３９】

第五の過程として、上記判断の結果（５０６）、攻撃に関連した内容を含んでいると、ウェブアプリケーション・ファイアウォールは、再構成されたＨＴＴＰトラフィックに含まれているパケット（または、再構成されたＨＴＴＰトラフィック）が正常ではないと判断し、上記再構成されたＨＴＴＰトラフィックを攻撃と検知する一方、正常ではない再構成されたＨＴＴＰトラフィックを再処理する過程を行う（５１０）。ここで、正常ではない再構成されたＨＴＴＰトラフィックに対する再処理過程は、次の二つの方法にて行うことができる。第一の方法は、正常ではないパケットを送信したウェブサーバまたは使用者サーバに対し、上記正常ではないパケットに対応するパケットの再送信を要請するか、上記パケットを削除する方法であり、第二の方法は、正常ではないパケットを変調して伝送する方法であって、以下、二つの方法についてより詳しく説明することにする。

【００４０】

すなわち、使用者が使用者サーバ３０を介してネットワーク上のウェブサーバ２０へ伝送したい（Ｒｅｑｕｅｓｔ）正常なメッセージ中に攻撃と疑われ得る文句（例：<ｓｃｒｉｐｔ>）が含まれている場合、実際に使用者の意図したところは攻撃ではなかったにもかかわらず、従来のウェブアプリケーション・ファイアウォールでは攻撃と判断して使用者の要請を遮断することもあった。しかし、このような場合、本発明が適用されるウェブアプリケーション・ファイアウォールが‘<ｓｃｒｉｐｔ>'Ｔａｇを‘[ｓｃｒｉｐｔ]'のように、つまり中の文句‘<'を‘ ['に変更することで攻撃文は成立しなくなり、この結果、使用者の正常な行動に対する攻撃誤検知を防止することができる。

【００４１】

また、ウェップサーバ２０から使用者サーバ３０へ伝送される（Ｒｅｓｐｏｎｓｅ）メッセージに個人情報が含まれている場合、単に個人情報が含まれているという理由にてページを遮断するとすれば、使用者は個人情報が含まれていない他の情報も見られなくなる。こうした場合、本発明が適用されるウェブアプリケーション・ファイアウォール１０では個人情報が含まれている部分のみを変調（Ｍａｓｋｉｎｇ、例：７６****-１１******）することで個人情報の流出と関係のない他のメッセージは正常に使用者に送信（Ｒｅｓｐｏｎｓｅ）されるようにすることができる。すなわち、本発明は外部から伝送されてくるウェブトラフィックからの攻撃を検知する機能だけではなく、ウェブトラフィックの変調を通じて個人情報の流出を抑える機能を有することをその特徴とするものであって、住民登録番号、カード番号、住所、電子メール、法人番号、事業者番号などのような個人情報の流出を抑える機能を遂行することができる。このために、本発明は、ウェブアプリケーション・ファイアウォールが、再構成されたウェブトラフィック（ＨＴＴＰトラフィック）に含まれているメッセージのうち、個人情報に係るメッセージの一部を外部から読み取り不可能なメッセージに変調することを特徴としている。

【００４２】

付言すると、本発明において意味する再構成されたＨＴＴＰトラフィックとは、パケットのヘッダー部分を分析し、シーケンスに応じてパケットを並べたものであって、当初Ｌ７階層で伝送しようとした原文メッセージを復元した状態のものを意味する。したがって、ウェブアプリケーション・ファイアウォールの上記パーサーのうち少なくともいずれかは、再構成されたＨＴＴＰトラフィックの内容を分析することで、攻撃文の有無を判断する一方、攻撃文などが含まれていて、正常ではないと判断されるパケットに対しては、送信ネットワークサーバに対し再送信を要請することで再度受信した後、前述したようにヘッダーを除去しＨＴＴＰトラフィックを再構成する過程（５０２）から繰り返すか、または、当該パケットのうち、攻撃に関連した内容のみを削除するか変調した後に伝送することもできる。

【００４３】

以下、前述したような本発明の二つの例を、［表１］及び［表２］を参照して説明する。

【００４４】

【表１】

[parserを利用したsemantic detection engineの例１]
Cross Site Scripting （XSS）攻撃文：<script type=“text/JavaScript”>alert（“penta”）;<script>

【００４５】

第一の例として、ＤＨＴＭＬ（ＸＭＬ）パーサーは、Ｔａｇの始端である<Ｔａｇ>と、Ｔａｇの終端である</Ｔａｇ>を一つのＴａｇと分析し、Ｔａｇの属性（Ａｔｔｒｉｂｕｔｅ）とＴａｇ中の関数を分析するようになる。

【００４６】

すなわち、従来のＷＡＦでは、通常、<script>タグが入っている場合、攻撃と判断して当該パケットを攻撃パケットとして処理していたが、本発明では、ＨＴＴＰトラフィックの全体を再構成し完成されたＤＴＨＭＬ構文を分析するため、<script>タグが検知されたとして、当該トラフィックを攻撃と処理することなく、再構成されたＨＴＴＰトラフィックの全体が攻撃文である場合のみに対して攻撃と処理するため、誤検知率が顕著に低くなる。

【００４７】

付言すると、本発明は、［表１］の場合、ＸＭＬパーサーがタグの始端とタグの終端を一つのタグと分析し、タグの属性とタグ中の関数を分析するものであって、従来ならば、<script>タグが入っていた場合に攻撃と判断したのに対し、本発明では再構成されたＨＴＴＰトラフィック構文の全体を分析することで、再構成されたＨＴＴＰトラフィックの全体が攻撃文である場合のみに対して攻撃と処理するという特徴を持っている。

【００４８】

【表２】

[parserを利用したsemantic detection engineの例２]
インジェクション攻撃文：（name=“penta” or name=“security”） and keyword=“pentasec”

【００４９】

ここで、エンドノード（Ｅｎｄ ｎｏｄｅ）の結果がいずれもＳＱＬの一部分であるので、全体文章のＳＱＬ文の有無＝ＴＲＵＥである。すなわち、第二の例として、非常に有名なＷｅｂ攻撃方法の一つであるＳＱＬインジェクション攻撃の場合にも、従来のＷＡＦでは、‘or string=string'の攻撃類型をストレージに登録しておくため、変形されたＳＱＬインジェクション攻撃に対する防御を事前にできず、既に攻撃がなされた以後の防御だけが可能であった。しかし、本発明では、データベース・マネジメント・システム（Ｄａｔａｂａｓｅ Ｍａｎａｇｅｍｅｎｔ Ｓｙｓｔｅｍ）で実行できるすべての種類のＳＱＬ構文に対するディテクションが可能であるので、変形された攻撃、新しい攻撃が現われても防御できるという特徴を持っている。

【００５０】

以上説明した内容を通じ、当業者ならば本発明の技術思想を逸脱しない範囲で種々の変更及び修正が可能であることが分かるであろう。したがって、本発明の技術的範囲は、明細書の詳細な説明に記載された内容に限定されるものではなく、特許請求の範囲によって決められるべきである。

【符号の説明】

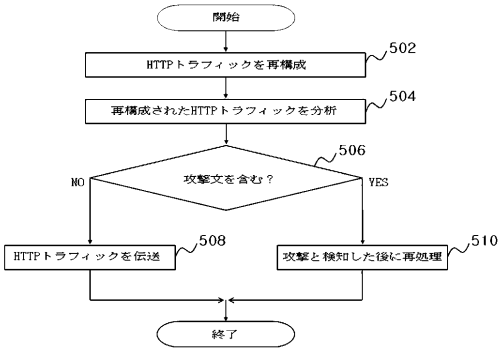
【００５１】

１０　ウェブアプリケーション・ファイアウォール

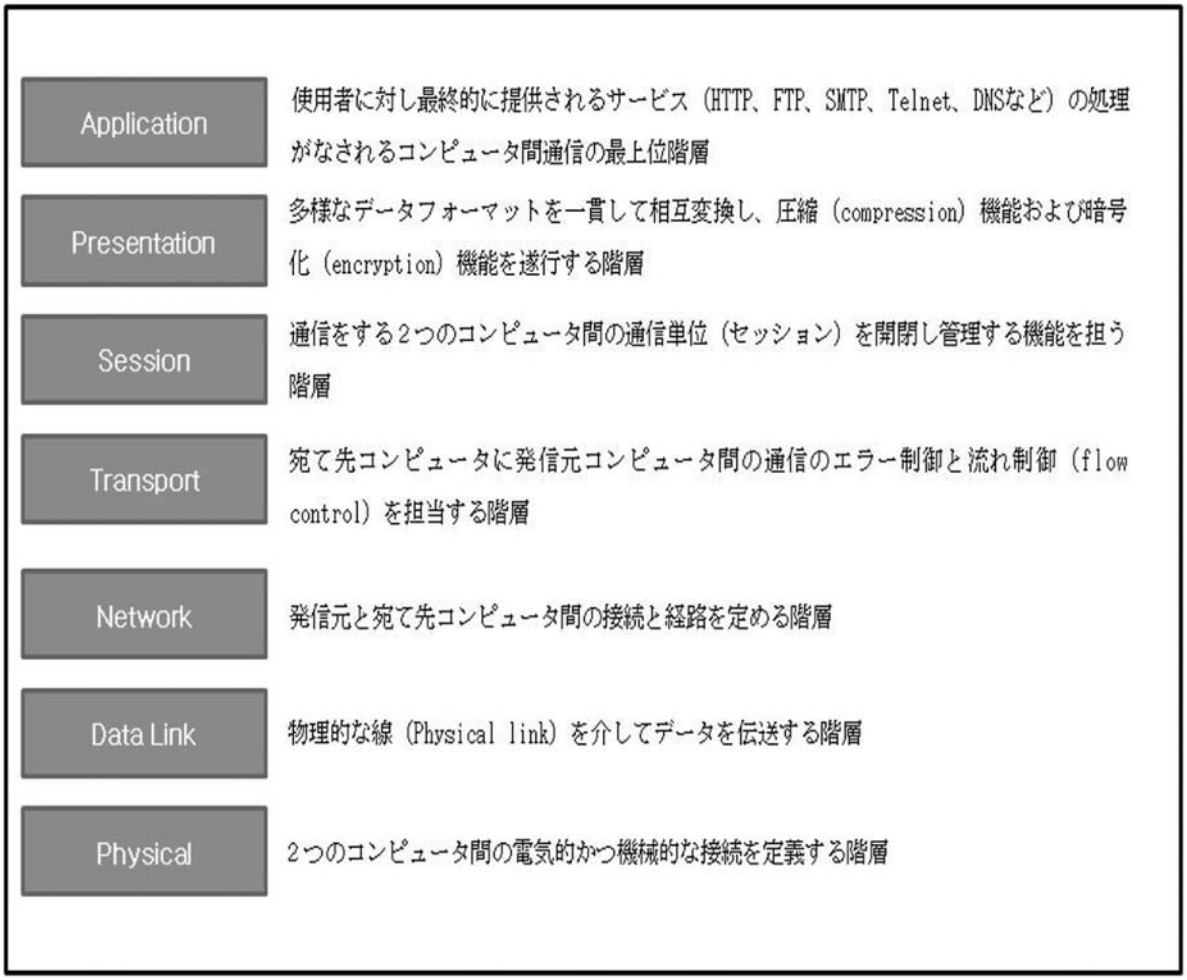
２０　ウェブサーバ

３０　使用者サーバ

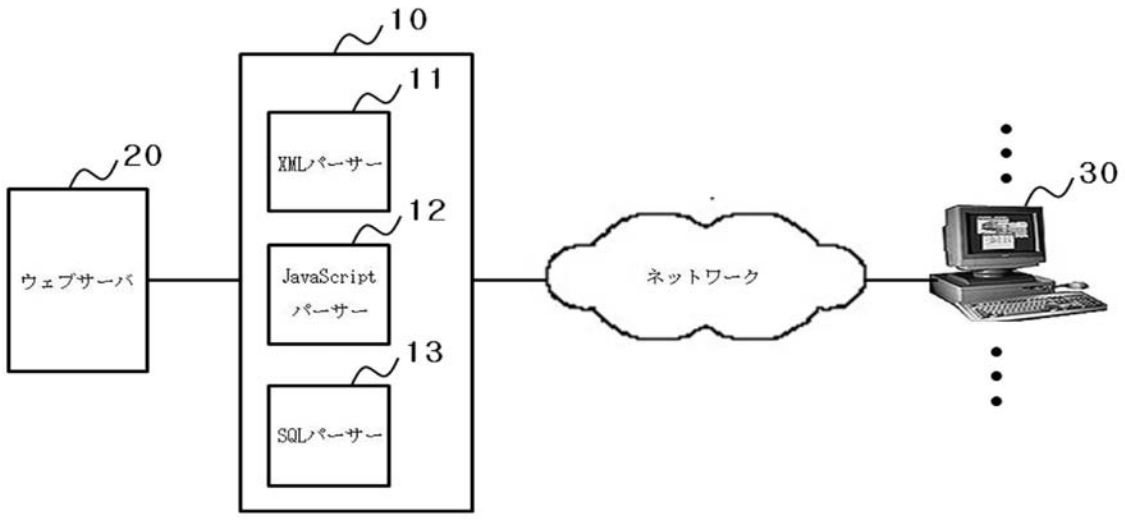
【図３】



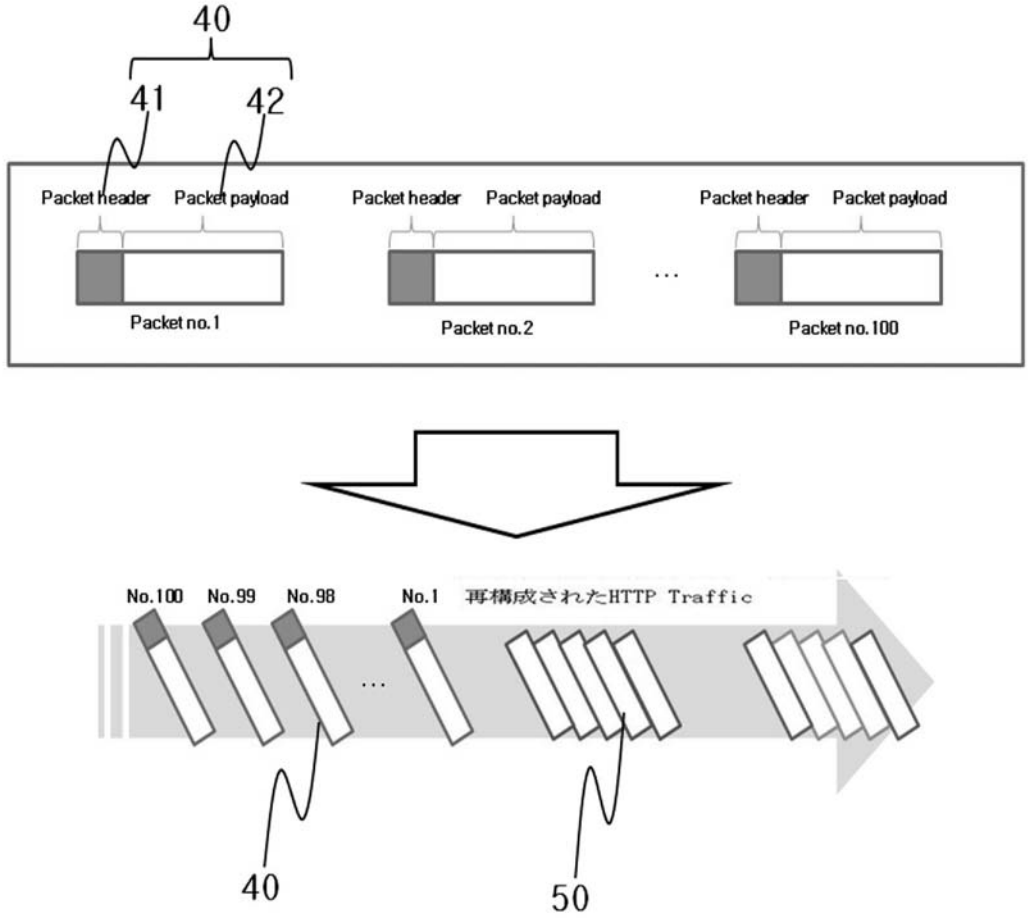
【図１】


Application
使用者に対し最終的に提供されるサービス（HTTP、FTP、SMTP、Telnet、DNSなど）の処理がなされるコンピュータ間通信の最上位階層
Presentation
多様なデータフォーマットを一貫して相互変換し、圧縮（compression）機能および暗号化（encryption）機能を遂行する階層
Session
通信をする2つのコンピュータ間の通信単位（セッション）を開閉し管理する機能を担う階層
Transport
宛て先コンピュータに発信元コンピュータ間の通信のエラー制御と流れ制御（flow control）を担当する階層
Network
発信元と宛て先コンピュータ間の接続と経路を定める階層
Data Link
物理的な線（Physical link）を介してデータを伝送する階層
Physical
2つのコンピュータ間の電気的かつ機械的な接続を定義する階層

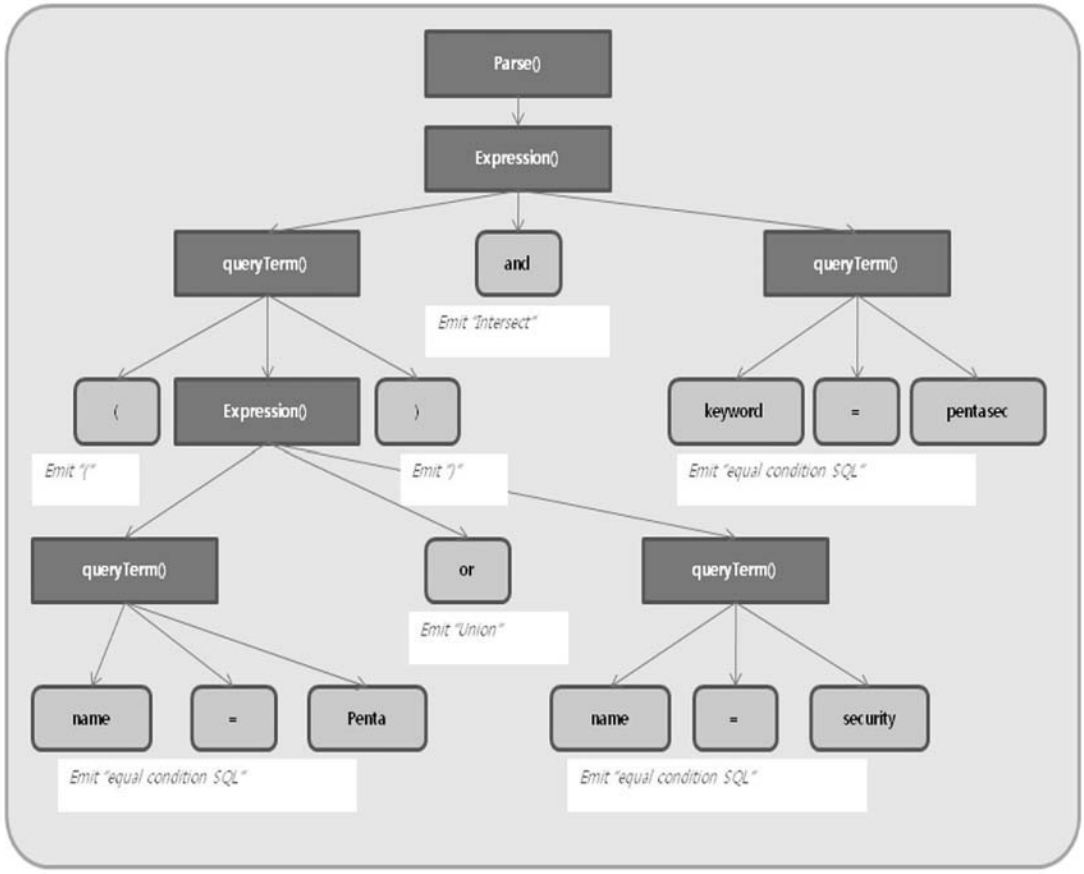
【図２】


10
11
XMLパーサー
12
JavaScript
パーサー
13
SQLパーサー
20
ウェブサーバ
ネットワーク
30

【図４】


40
41
42
Packet header
Packet payload
Packet header
Packet payload
Packet header
Packet payload
Packet no.1
Packet no.2
Packet no.100
No.100
No.99
No.98
No.1
再構成されたHTTP Traffic
40
50

【図５ａ】



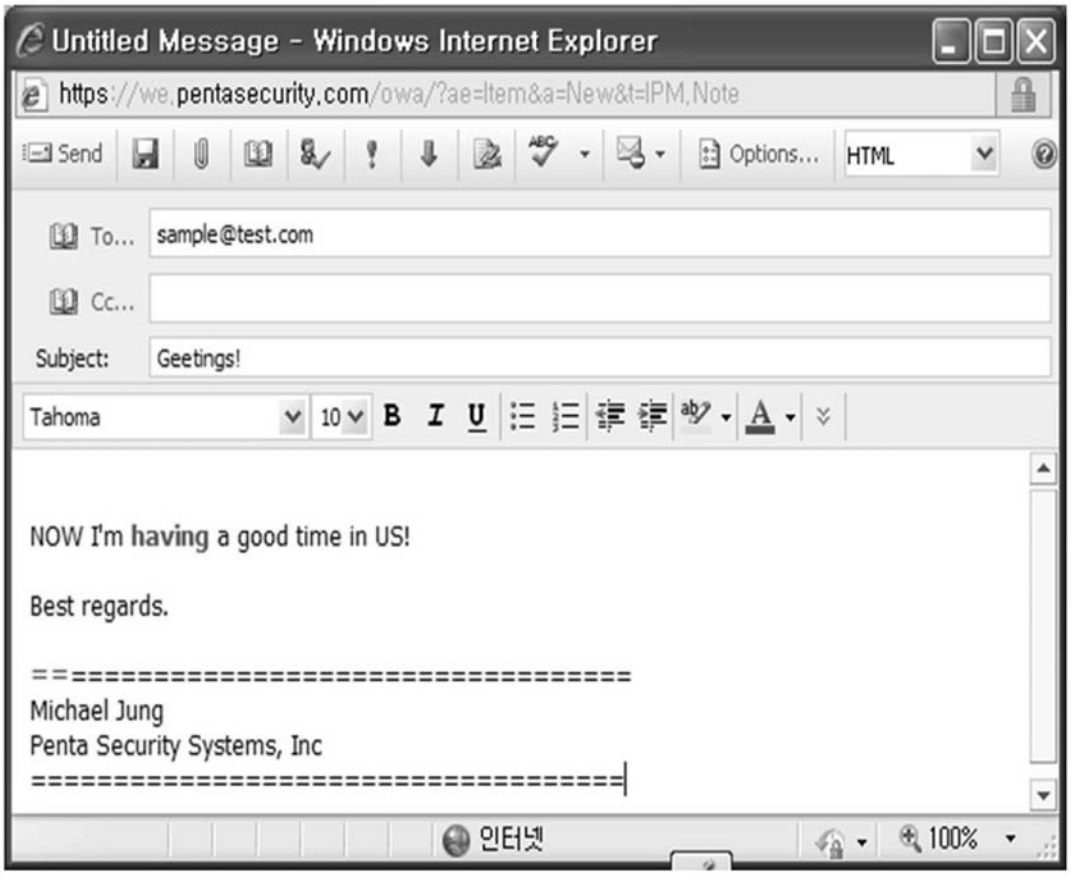
【図５ｂ】

Signature	Signature Name
part="or 'a' = 'a"	SQL Injection WHERE Statement Manipulation
part="or 'a'= 'a"	SQL Injection WHERE Statement Manipulation 1
part="or 'a'= 'a"	SQL Injection WHERE Statement Manipulation 2
part="or 'a'='a"	SQL Injection WHERE Statement Manipulation 3
part="or 'a' ='a"	SQL Injection WHERE Statement Manipulation 4
part="or 'a'='a"	SQL Injection WHERE Statement Manipulation 5
part="or 'a' ='a"	SQL Injection WHERE Statement Manipulation 6
part="or a=a"	SQL Injection WHERE Statement Override
part="or 1=1"	SQL Injection WHERE Statement Override 1

【図５ｃ】

Signature	Signature Name
part="waitfor", rgxp="[^a-zA-Z]waitfor#s*delay"	SQL Injection - Waitfor
part="having", rgxp="[^A-Za-z]having[^#&]{0,20}=[^#&]{0,20}"	SQL Injection - "having" statement injection attampt
part="opendatasource", rgxp="select.*from.*opendatasource"	SQL Injection – opendatasource

【図５ｄ】



フロントページの続き

(72)発明者　リー、セオクーウー
大韓民国、ソウル、ジューングーグ、シンダングードング、４３２－１０９６、ロンドン　アパートメント　２０１
(72)発明者　パーク、ヨウングーイン
大韓民国、ヨウンジンーシ　ギェオンッギードド、スージーグ、シンボングードング、エルジー　５チャ　アパートメント　５１５－１９０２
(72)発明者　パーク、ハエーミン
大韓民国、ソウル、マポーク、ドゥファ１ードング、ハンファ　オフィセテル　１８０３

Ｆターム(参考) 5B089 GB09 HB05 JA21 KA17 KC31 KC54
5B276 FD08 FD09
5B285 AA05 AA06 AA07 BA03 CA32 CA34 DA05
5K030 GA15 HA08 HC01 HD03 JA10 KA07 LC13 MA04 MB18 MC07
MC08 MD08