社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＣＳＥＣ－22　(13)
2003／7／17

HTTP リクエスト解析による未知攻撃防御システム

今野 徹　　楯岡 正道

株式会社 東芝 e-ソリューション社 プラットフォームソリューション事業部 要素技術開発担当

〒183-8511 東京都府中市東芝町 1

E-mail: {toru.konno, masamichi.tateoka}@toshiba.co.jp

あらまし　Web サーバを対象とする未知攻撃防御システムを提案する。HTTP リクエストの内容を文字列として解析し、文字列区分と構成要素を抽出して評価し、各々の平均値と分散を求め統計分布を学習する。そして、統計的に異常とみなされる閾値を算出し、その閾値に基づいて、新たな HTTP リクエストの内容に含まれる文字列を評価し、統計的に正常であるか異常であるかを判定することで、その HTTP リクエストが未知の攻撃であるか否かを判定する。実験において 92%の平均検出率、0.17%の平均誤検出率を達成した。

キーワード　未知攻撃、HTTP、侵入検知、侵入防御、IDS、IDP

Anomaly-based intrusion detection and prevention by semantic evaluation of HTTP requests

Toru KONNO　and　Masamichi TATEOKA

Advanced technology development group, platform solutions division, TOSHIBA Corporation e-Solution Company

1, Toshiba-cho, Fuchu-shi, Tokyo, 183-8511, Japan

E-mail: {toru.konno, masamichi.tateoka}@toshiba.co.jp

Abstract　We implemented anomaly intrusion detection/prevention system which detects semantic anomaly in HTTP requests to web servers. The system parses HTTP requests, evaluates each of the semantically separated elements by the length, or the number of included specific characters, and calculates the means and the variances of those evaluated values. To detect/prevent anomaly HTTP requests, the system discriminates anomaly by evaluated values of the syntactic elements. Experimentation shows that average true positive rate is 92%, and average false positive rate is 0.17%.

Keyword　Anomaly Detection, HTTP, Intrusion detection and prevention, IDS, IDP

1. はじめに

Web サーバを攻撃から防御する侵入検知・防御システム（IDS=Intrusion Detection System, IDP=Intrusion Detection Prevention）は、基本的な機能として、既知の攻撃パターンを記述したシグネチャを持っている。シグネチャ方式は、HTTP リクエストに含まれる特定文字列を検出するためには有効かつ必要なものであるが、記述されていない未知の攻撃からは Web サーバを防御することができない。

ここ数年は、シグネチャに記述されていない攻撃を検出することを目的とした未知攻撃検出技術について、様々なアプローチが提案されている。例えばアノーマリ検知といわれる手法は、トラフィック量の変化や、TCP/IP、HTTPといった基本プロトコルでの異常を検出することに焦点を当てている[1]。ただしシグネチャ方式で有効であった、アプリケーションの脆弱性をねらった特定文字列による不正アクセスの検出には必ずしも対応するものではない。別の手法としては、Webサーバからクライアントに提示されるHTMLページの内容を分析することで、文字列レベルでの正常範囲を限定し、異常を検知する手法がある[2]。この手法は対象Webサーバから直接提示されるHTMLページから基準を自動作成するが、それ以外のJavaScript等に関しては対応しないとされている。さらに、異なるWebサーバでHTTPリクエストが転送される状況では、提示されるHTMLページだけに依存しない一般のHTTPリクエスト文字列を扱うための枠組みが必要とされる。

また、未知攻撃を対象とする侵入検知・防御システ

ムを評価する場合、シグネチャ方式のシステムの場合と同様に、攻撃の検出率と誤検出率について評価する必要があるが、従来の提案・製品をみても、実際の様々なWebサーバを対象とした十分な条件下での評価結果が提示されているとは必ずしもいえない状況である。

そこで本稿では、HTTPリクエスト文字列を対象として学習段階において解析し、統計的に正常な範囲を判断する検出基準を作成することにより、Webサーバを未知攻撃から防御するシステムを提案する。評価としては、様々なWebサーバに対して通常のHTTPアクセスが多数発生する実環境ネットワークにおいて正常パターンを学習させた上で、シグネチャ開発の検査で用いられる攻撃パターンによって攻撃を行い、提案する未知攻撃防御システムの検出率と誤検出率を求め、本システムが高い精度でWebサーバを防御できることを示す。

以下、本システムの詳細について述べる。

2. 未知攻撃防御の提案方法

2.1. システム概要

図 1は、今回開発した侵入検知・防御システムの全体構成である。本システムは、Webサーバとクライアントとの間に置く構成であり、HTTPアクセスをアプリケーション層で解析・中継する。Webサーバ側とクライアント側とからはブリッジとしてみえ、データリンク層として透過的である。

本システムには、Web サーバを攻撃から防御するための機能として、既知攻撃防御機能と未知攻撃防御機能がある。既知攻撃防御機能では、既知の攻撃をシグネチャや DDoS 防御機能により検出・防御する。未知攻撃防御機能では、既知攻撃防御機能で検出・防御できなかった未知の攻撃を検出・防御する。既知攻撃防御機能でも未知攻撃防御機能でも正常なアクセスであると判定されたセッションについては、セッションを通過させる。また、本システムは、管理 GUI より管理者へ未知攻撃の情報提示をおこない、管理 GUI から未知攻撃の検出基準の補正を受け付ける。

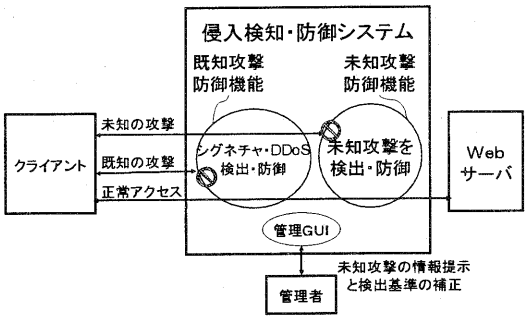

図 1 本研究の侵入検知・防御システムの全体構成

なお、今回実装した未知攻撃防御システムは、我々が開発した東芝 e-ソリューション社 IDP 製品[3]の開発プラットフォーム上に試作したものである。

図 2は、未知攻撃防御機能に係わる処理間の関連を示している。未知攻撃防御機能には、学習フェーズと、検出・防御フェーズの二つのフェーズがある。

学習フェーズでは、正常なHTTPリクエストを入力とし、HTTPリクエストの内容を解析し、解析結果を統計分布として記憶し、解析結果の評価値が、統計的に正常値か異常値かを分ける閾値を算出する。学習フェーズではセッションを通過させる。

検出・防御フェーズでは、HTTPリクエストを入力し、HTTPリクエストの内容を解析し、解析結果を既に学習済みの統計分布を基準として比較し、統計的に正常か異常かを判定することにより、未知の攻撃を検出する。検出の結果、未知の攻撃についてはセッションを遮断し、正常なアクセスはセッションを通過させる。

また、未知の攻撃を検出した場合は、管理GUIに未知攻撃の情報を提示し、真に攻撃であったか否か管理者の判断に応じて、未知攻撃の検出基準を補正する。

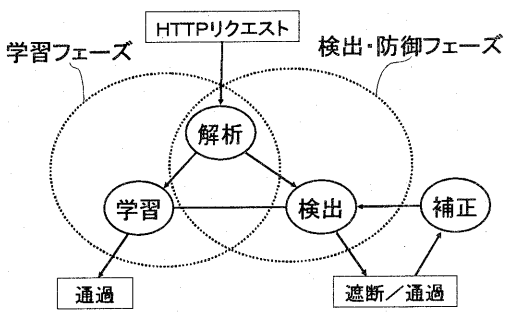

図 2 未知攻撃防御機能に係わる処理間の関連

2.2. 解析

2.2.1. 文字列区分の切り出し

解析処理では、アプリケーション層データとして組み立てられたHTTPリクエストを対象に、HTTPリクエスト文字列を解析する。

はじめにHTTPリクエストから文字列区分の切り出しを行うが、本方式では、それぞれの文字列区分が、Webサーバ上の何によって実際に処理されるかを考慮した切り出しを行う。たとえば、ホスト名以降の文字列区分は一般的にWebサーバ上のHTTPデーモンによって処理されるものであり、パス名以降の文字列区分はHTTPデーモンから呼び出されるアプリケーションによって処理され、そしてパラメータ名以降の文字列区分はアプリケーションの内部変数によって処理されることが多いことがわかっている。そして、そうした処理の脆弱性が攻撃に関わりうることから、本方式ではそのような処理対象となる文字列区分を抽出している。例えば、ホスト名からの切り出しは最上位の文字列区分1とし、以下、パス名以降を区分2、パラメータ名以降を区分3、といった階層的な切り出しを行う。

2.2.2. 文字列区分の構成要素の評価

抽出した文字列区分のそれぞれについて、構成要素を評価する。本方式でいう構成要素の評価とは、パターン認識の分野でいう特徴量の抽出にあたる。具体的には、任意の文字（すなわち文字列区分そのものの長さ）や、ある特定の文字セットに属する文字（例えばスラッシュ文字やドット文字）が、構成要素として何バイト含まれているかを、文字列特性として評価する。これらの文字列特性は、Webサーバへの攻撃手法として知られる数々の攻撃パターンをBugtraq[4]等により調査した結果、多くの攻撃パターンにみられる文字列特性として列挙されるものである。

以上の解析処理から、HTTPリクエストの評価値 $f_{mn}(x)$ を算出する。

評価値 $f_{mn}(x)$ ：

HTTPリクエスト X に関し、文字列区分 m において、文字列特性 n としてカウントされる文字のバイト数

X ：HTTPリクエストの事象
m ：文字列区分
n ：文字列特性

2.3. 学習

学習フェーズでは、HTTPリクエストが入力される度に、内容を解析し、解析結果の評価値 $f_{mn}(x)$ を算出する。そして算出した評価値を統計分布データとして記憶する。学習フェーズではHTTPリクエストは通過させるだけであり、未知攻撃の検出・防御は行わない。学習フェーズの終了時に、HTTPリクエストの評価値 $f_{mn}(x)$ に関して、平均値と分散を算出し、統計的に正常な値と異常な値とを分ける閾値を算出する。これにより、評価値 $f_{mn}(x)$ に関して図3のような統計分布を得る。

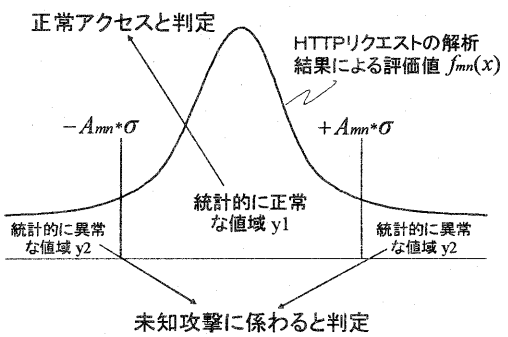

図3 HTTPリクエスト解析結果の評価値に関する統計分布の概念図

学習フェーズにおいて得られた評価値の統計分布は、未知攻撃の検出・防御フェーズにおいて参照される。すなわち、評価値 $f_{mn}(x)$ を閾値 $\pm A_{mn}*\sigma$ によって正常な値域y1と異常な値域y2に分けて、評価値が統計的に正常な値域y1に属するものは正常アクセスと判定し、評価値が統計的に異常な値域y2に属するものは未知攻撃に係わるものと判定する。閾値は、$\pm A_{mn}*\sigma$ として算出する。ここで、A_{mn} は定数であり、m、nの組み合わせに対してあらかじめ定めた値である。σは、$f_{mn}(x)$ に関して求められる標準偏差である。検出手順の詳細は2.4節で述べる。

学習フェーズにおいて記憶される統計分布データは、ホスト名と文字列区分と文字列特性に基づく記憶テーブルとして管理される。

2.4. 検出・防御

未知攻撃を検出するための本方式による基本的な考え方としては、図 3に示したように、学習フェーズにおいて得られた評価値 $f_{mn}(x)$ の統計分布データを参照し、入力された HTTP リクエストについて、評価値 $f_{mn}(x)$ が統計的に正常な値域に属するものは正常アクセスと判定し、評価値が統計的に異常な値域に属するものは未知攻撃に係わるものと判定することである。

検出処理に係わる検出・防御フェーズ全体の処理の流れを説明する。

まず、入力されたHTTPリクエストの内容を解析する。そして、HTTPリクエストから抽出される文字列区分mについて、そのアクセス頻度を評価する。

文字列区分mへのアクセス頻度が十分である場合は、文字列区分mに関する統計分布データを参照する。そして、文字列区分mに関して、HTTPリクエストの評価値と、学習フェーズで算出した閾値とを比較することにより、HTTPリクエストが未知攻撃に係わるか否かを判定する。評価値が閾値よりも異常側であった場合は、HTTPリクエストを未知攻撃に係わるものと判定し、評価値が閾値よりも正常側であった場合は、HTTPリクエストを正常なアクセスと判定する。

文字列区分mへのアクセス頻度が不十分である場合は、文字列区分m－1に関する統計分布データを参照し、以下同様に、文字列区分の評価・比較を行っていくことで、最終的にHTTPリクエストが未知攻撃に係わるか否か判断する。

本システムでは、HTTPリクエストが未知攻撃に係わるものと判定された場合はそのHTTPセッションを遮断し、HTTPリクエストが正常なアクセスと判定された場合はそのHTTPセッションを通過させる。

3. 性能評価

未知攻撃防御システムの性能評価する場合は、従来のシグネチャ方式を性能評価する場合と同様に、未知の攻撃に対して、高い検出率と低い誤検出率を実現するかどうかが重要な指標となる。本研究では、正常な HTTP アクセスを標本として多数採取できる実環境ネットワークにおいて学習させた上で、シグネチャ開発の検査で用いていた攻撃パターンによって攻撃を行い、検出率と誤検出率を評価した。

3.1. 評価環境

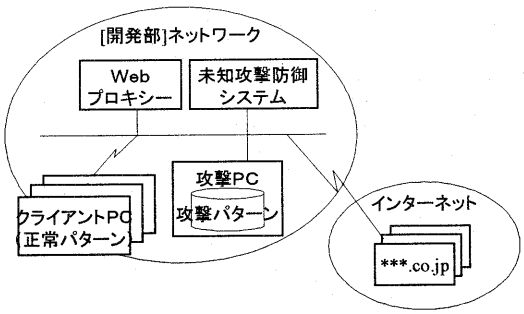

図 4 評価環境

「正常パターン」

学習フェーズとして、我々の開発部ネットワーク上で、すべてのクライアント PC から Web プロキシーを経由してインターネット上の Web サーバに通常にアクセスされるすべての HTTP リクエストデータを採取した。これを「正常パターン」とする。その中から、総アクセス数が多かった上位 140 サイトを対象として、今回の実験における防御対象の Web サーバとした。

「攻撃パターン」

当社 IDP 製品「MAGNIA2000Ri/Anti-Hacker」[3]のシグネチャ開発の検査に用いられている既知攻撃パターンのデータベースを用いた。この攻撃パターンは Bugtraq[4]等に寄稿された公開済みの情報に準拠している。今回は攻撃パターンを 853 個使用した。

3.2. 検出率

以下の手順により検出率を測定する。

1. 学習フェーズとして、ある Web サーバに対する正常パターンのすべてを未知攻撃防御システムに流し、統計分布を得る。
2. 検出・防御フェーズとして、すべての攻撃パターンを順番に流し、攻撃として検知した数をカウントする。
3. この Web サーバを防御対象とした場合の検出率を、以下により算出する。

検出率＝（攻撃として検出した攻撃パターンの数）
／（攻撃パターンの総数）

以上の手順により、140 サイトすべてについて評価した結果、検出率の平均は 92%となった。

平均検出率 = 92%

3.3. 誤検出率

以下の手順により誤検出率を測定する。

1. 学習フェーズとして、ある Web サーバに対する正常パターンのうち、ランダムに抽出した一部を未知攻撃防御システムに流し、統計分布を得る。
2. 検出・防御フェーズとして、残りの正常パターンを流し、誤って攻撃として検知した数をカウントする。
3. この Web サーバを防御対象とした場合の検出率を、以下により算出する。

誤検出率＝
（手順 2.で誤って攻撃と検知した正常パターン数）
／（手順 2.で流した正常パターンの総数）

以上の手順により、140 サイトすべてについて評価した結果、誤検出率の平均は 0.17%となった。

平均誤検出率 = 0.17%

図 5に、今回対象とした 140 サイトの各 Web サーバの検出率と誤検出率について、検出率を縦軸とし、誤検出率を横軸としてプロットした散布図を示す。各 Web サーバの総アクセス数は、最高が約 38,000、最低が約 1,000、平均が約 5,500 である。

このデータで取り上げられている Web サーバは、インターネット上で我々が任意にアクセスできる Web サーバであって、単純に総アクセス数の多いものから順に評価した結果である。また、今回の性能評価で用いている閾値の係数 A_{mn} は、全てのサイトに対して共通のものを用いている。

よって本方式は、平均的に、実際の様々な Web サーバに対して高い検出率と低い誤検出率を達成する効果があるといえる。

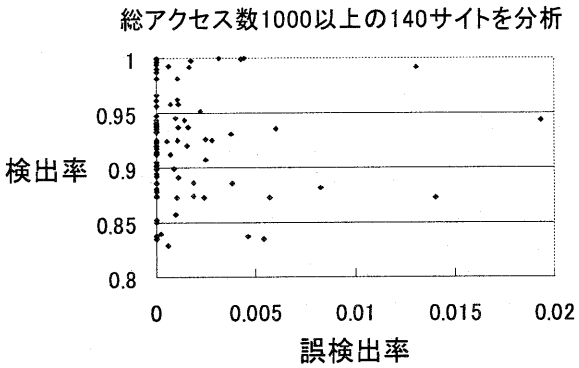

図 5 検出率と誤検出率

なお、今回の評価で攻撃として検出された攻撃パターンの例として、著名なものとしては CodeRed や Nimda が挙げられる。これらの攻撃パターンについては、指定されたパラメータ名へのアクセス頻度が極めて少ないことと同時に、パス名以降の文字列区分の長さが正常アクセスの統計分布と比較して極めて長すぎること、または、パス名以降の文字列区分に殆ど含まれないスラッシュ文字が含まれていたことを検出した。

4. 考察

4.1. 本手法の効果

本手法は、図 5の結果が示すように、平均検出率92%という高い検出率と、平均誤検出率0.17%という低い誤検出率が得られた。

統計分布を参照する際に、標本数が十分でなければ、統計的には正常か異常かを必ずしも判定することはできない。たとえば、パス名の文字列区分に関する統計分布を参照したときに、通常はアクセスされないパスにアクセスされたと言うだけで未知攻撃に係わると判断すると、誤検出も多く含んでしまう。

また、参照する統計分布の分散が大きい場合、すなわち、正常アクセスであっても文字列の構成要素が多様である場合、標本数が十分であっても、異常な文字列を明確に検出することが難しくなり、検出率が低くなる傾向がある。字列区分へのアクセス頻度は、上位の文字列区分であるほど高くなる。具体的には、ホスト名＞パス名＞パラメータ名となる。これに対し、統計分布の分散は、上位の文字列区分であるほど大きくなる。

初期の実験で、これらの事柄がわかった事から、今回、検出率と誤検出率のトレードオフを解消するため

に、下位の文字列区分に関する統計分布から優先的に参照し、参照した文字列区分のアクセス頻度が不十分ならば、より上位の文字列区分に関する統計分布について参照するという手順を設けた。これにより、総アクセス数が1000以上の場合については、図 5に示すような、高い検出率と低い誤検出率を達成できた。

4.2. 誤検出率の削減について

今回の評価結果では、140 サイトのうち、誤検出率が0であったサイトは、96サイトであった。残りのサイトは、CGIに与えるパラメータが多いものや、パス名の構造が複雑であるなど、状況は一概には言えないが、誤検出が発生するという結果になっている。

本システムを実際に運用する場合には、検出精度のさらなる向上、特に誤検出の削減を図ることが求められる。

誤検出を削減するための手法としては、誤検出されたHTTPリクエストを対象にニューラルネットワークで学習して検出精度を向上させる手法が提案されている[5]。これはHTTPリクエスト文字列を対象とした学習を侵入検知に適用しているという点で、我々の手法と類似性があるといえる。この手法では、パターン抽出のために「最小編集距離」を用いている。我々の手法では全ての正常アクセスを対象としており、正常アクセスのパターンは非常に多いため、最小編集距離をそのまま適用することは難しいと考えられることから、未知攻撃を対象とした場合には、正常アクセスの学習は本提案手法で行い、誤検出を削減するにはこのような手法と組み合わせることで、検出精度をより高められる可能性があると考えられる。

4.3. 閾値の最適化について

今回の性能評価では、全てのサイトに対して共通の閾値係数 A_{mn} を用いており、本システムの平均検出率・平均誤検出率を示した。現実的には、特定のWebサーバを防御する運用が想定され、そうした場合は閾値係数 A_{mn} の値を防御対象の特定の Web サーバに対して最適化することにより、より高い検出率と低い誤検出率を達成できると見込まれる。

我々の試作システムでは、管理者の負担が極力少ない管理GUIを提供しており、例えば閾値の係数 A_{mn} を補正したり、誤検出の原因となる文字列区分を選択して判定の対象外とするなどの柔軟性を持たせている。

5. まとめ

HTTPリクエスト文字列を対象とした解析に基づく学習を行い、統計的に正常か異常かを判断する検出基準を算出することで、Webサーバを未知攻撃から防御する手法を提案した。検出率と誤検出率とのトレードオフを解消するために、下位の文字列区分に関する統計分布について、アクセス頻度が十分に大きければ優先的に参照し、アクセス頻度が不十分であれば、より上位の文字列区分に関する統計分布について参照することが有効である。評価を行った結果、本方式が、様々なWebサーバに対して、平均的に高い検出率と低い誤検出率を達成する効果があることを確認した。

文　献

[1] 馬場他、“プロトコル仕様及びポリシー情報を利用した不正アクセス検知システムの実環境評価”、情報処理学会研究報告、2002-CSEC-18、pp.33-38、July.2002

[2] SANCTUM inc、AppShield、
http://www.sanctuminc.com

[3] 東芝 e-ソリューション社、 Internet Appliance Server「MAGNIA200Ri/Anti-Hacker」
http://cn.toshiba.co.jp/prod/iaserver/magnia/2000ri/index_j.htm

[4] BUGTRAQ,
http://www.securityfocus.com/popups/forums/bugtraq/intro.shtml

[5] 宮路他、“機械学習によるネットワーク型 IDS の false positive 削減手法の提案”、情報処理学会研究報告、2003-CSEC-21、pp.53-58、May.2003