アプリケーション・ファイアウォールによるWebサーバの保護

露口　和弘　　橘　喜胤
矢野　達朗

近年，インターネットにおける不正アクセスが問題になっている。インターネットのセキュリティ対策としては，ファイアウォールや侵入検知システムの利用が普及しているが，これらの技術だけでは対処できないのが実情である。不正アクセスの手法は高度化および一般化しており，ファイアウォールの背後に設置したWebサーバプログラムの脆弱性を突いた攻撃が日常的に行われている。この問題に対する一般的な対策は，Webサーバプログラムの脆弱性が発見されたら，その脆弱性を解消したプログラムに置き換えることである。しかし，プログラムを置き換えると，事前に動作検証が必要でありコストがかかる。また，プログラムの新しい脆弱性が次々に発見されていくことを考えると，対策は容易ではないし万全でもない。

Webサーバを防御する新しい対策方法として，侵入検知技術を使用して，Webサーバなどアプリケーションプログラムの脆弱性を突いた攻撃を遮断する手法が登場している。本稿では，この手法をアプリケーション・ファイアウォールと呼ぶ。筆者らは，今後新しく登場する攻撃も防御可能なアプリケーション・ファイアウォールを開発した。本稿では，そのアプリケーション・ファイアウォールの概要を紹介し，その有効性について述べる。

ファイアウォールと侵入検知システム

ファイアウォール

ファイアウォールは，送受信IPアドレス，ポート番号，プロトコルなどの情報を元に，通信の許可と非許可を制御するものである。図1にファイアウォールの動作概要を示す。ファイアウォールは，プロトコルごとにアクセスを制御する機能を持つが，許可したプロトコルのデータ内容は検査しない。このため，ファイアウォールでWebサーバへのアクセスを許可している場合，HTTP（Webプロトコル）を使用してWebサーバプログラムの脆弱性を狙った攻撃は，ファイアウォールを通過してしまう。

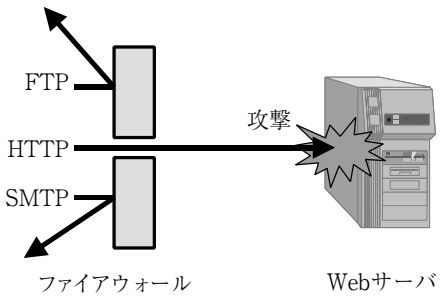

図1　ファイアウォールの動作概要

侵入検知システム

侵入検知システムは，ネットワーク上を流れるデータを監視し，疑わしいデータを検出すると管理者に通知する。図2に侵入検知システムの動作概要を示す。侵入検知システムの基本動作は監視であり，ファイアウォールを通過するような攻撃も含めて検出することはできるが，攻撃そのものを遮断することはできない。侵入検知システムの中には，不正アクセスを検出した後のレスポンス機能として，ルータやファイアウォールの設定を自動的に変更したり，TCPプロトコルのリセットパケットを送出したりして，通信を遮断するものがある。しかしこれらは事後的動作であり，疑わしいと判断したデータそのものは通過してしまう。

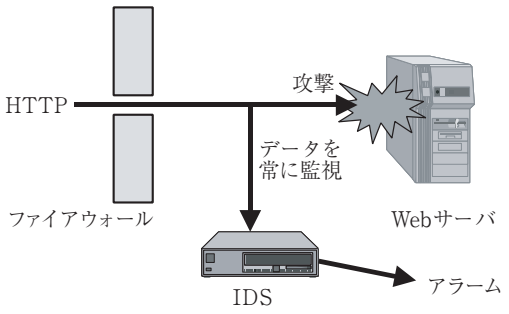

図2　侵入検知システムの動作概要

アプリケーション・ファイアウォールの概要

アプリケーション・ファイアウォールは，通常のファイアウォールを補完するものであり，ファイアウォールを通過する攻撃を遮断して，サーバシステムを防御するものである。アプリケーション・ファイアウォールは，ファイアウォールを通過するような不正アクセスを，侵入検知技術を使って検出し遮断する。通信の遮断は，侵入検知システムのレスポンス機能とは異なり，疑わしいデータはいっさい通過を許さない。図3にアプリケーション・ファイアウォールの動作概要を示す。

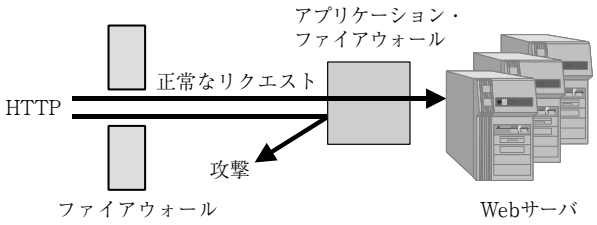

図3　アプリケーション・ファイアウォールの動作概要

アプリケーション・ファイアウォールは，利用者とサーバ上で動作しているWebサーバ等のアプリケーションプログラムの間に位置している。アプリケーション・ファイアウォールをどこに配置するかにより，大きく分けて2種類の構成方法がある。

●エージェント型

エージェント型は，Webサーバ等のアプリケーションと同一のコンピュータ上で動作する方式である。この方式では，サーバごとにエージェントが存在するため，アプリケーション・ファイアウォールの処理が分散され負荷が集中しないという長所がある。また，特定のOSやWebサーバプログラムの固有の問題にきめ細かく対応することができる。その反面，特定のOSやWebサーバプログラムごとにエージェントを用意しなければならないという短所がある。

●ネットワーク型

ネットワーク型は，通信の中継および遮断を行う専用装置として実現する方式である。すべてのデータは本装置を経由してサーバに受け渡される。この方式では，防御対象サーバのOSやWebサーバプログラムなどに依存せずに実装可能であるという長所がある。その反面，アプリケーション・ファイアウォールの処理にかかる負荷が一箇所に集中するという短所がある。

ネットワーク型はさらに，どのネットワーク階層で中継するかにより，いくつかの方式に分けられる。

データリンク層やネットワーク層で中継する方式は，プロトコルのヘッダ情報などアプリケーション層には渡されないデータも含めて処理可能であり，プログラムしだいでさまざまな機能を実装することができる。その反面，フラグメントパケットの処理など，アプリケーション・ファイアウォールで対処しなければならない処理が増大する。

アプリケーション層で中継する方式（プロキシ方式）は，フラグメントパケット処理等はOSレベルで行われ，アプリケーション・ファイアウォールは目的の処理に専念することができる。その反面，プロキシ機能により対応アプリケーションが限定される。

不正アクセス検出アルゴリズム

アプリケーション・ファイアウォールは，侵入検知システムと同じ技術を用いて不正アクセスを検出する。不正アクセスの検出アルゴリズムには，不正検出（misuse detection）と異常検出（anomaly detection）の2種類がある。

不正検出（misuse detection）

不正検出は，不正アクセスのパターンをあらかじめデータベースとして保持しており，これとネットワーク上を流れるデータの内容を比較することにより，不正を検出する。不正検出は，どの不正アクセスであるかを特定する能力を持つが，データベースにない不正アクセスは検出できない。現在，多くの侵入検知システムで採用されているアルゴリズムである。

異常検出（anomaly detection）

異常検出は，正常アクセスの情報をあらかじめ保持しており，ネットワーク上を流れるデータを観測して，正常時と異なる異常データを検出する。異常検出は，事前に必要なものは正常時の情報であり，未知の不正アクセスを検出することができる。その反面，不正アクセスの種類を正確に特定することは困難である。

適用例

不正検出と異常検出の違いについて，CodeRed[1)] の攻撃を例として説明する。CodeRedは，Microsoft社のWebサーバプログラムのバッファオーバフロー問題を利用して，リモートからWebサーバにプログラムを送り込み実行する。図4は，CodeRedがWebサーバに送るデー

タである。バッファオーバフローを引き起こす連続する“N”の文字列に続いて，“%u”から後に実行させるプログラムコードが埋め込まれている。

図4　CodeRedのリクエスト

不正検出の考え方では，CodeRedのリクエストそのものを，不正アクセスパターンとして保持しており，監視対象のネットワークデータとパターンマッチングすることによりCodeRedを検出する。このため，正確にCodeRedを識別することができる。しかし，例えば攻撃データ中の文字を“N”から“X”に変えるなど，攻撃の一部を変更すると検出することができなくなる。

一方，異常検出の考え方では，監視対象データがCodeRedか否かを判定するのではなく，監視対象データの文字列が通常よりも長いことや，通常では使用しない“%u”といったエンコーディングが含まれていることによって，異常を検出する。したがって，例えば攻撃データの一部を変更した攻撃であっても検出することができる。このように，異常検出では，攻撃種別を特定することはできないが，正常時と異なる特徴を持つ不正アクセス全般を検出することができる。

アプリケーション・ファイアウォールの設計

筆者らの設計したアプリケーション・ファイアウォールは，さまざまなWebサーバプラットフォームに適応可能であり，かつ，今後新しく登場する攻撃に対応可能であることを目的としている。このため，構成方法はネットワーク型とし，実装の容易なプロキシ方式を採用している。また，不正アクセスの検出アルゴリズムには異常検出の考え方を採用した。異常検出にはさまざまなアルゴリズムが研究開発されているが，筆者らは，以下の3つの観点に着目した正常状態をルールとして定義し，ルールに違反するものを異常と判別する方式を採用した。

●データ長

データ長が通常のデータよりも異常に大きい場合，バッファオーバフロー攻撃の可能性がある。

●文字種別

7bit長のASCII文字が期待されている場合に8bit長のデータが送られたり，一般のASCII文字が期待されている場合に特殊文字（例えば，“&（）＃‘など，文字列やコマンドの区切りで特殊な用途に使用される文字）が出現したりする場合，意図的に攻撃が行われている可能性がある。

●エンコーディング

通常使用されない%uエンコーディングが使用されたり，エンコーディングの必要がない通常ASCII文字がエンコーディングされている場合，意図的に攻撃が行われている可能性がある。

Webサーバシステムへの攻撃対象としては，Webサーバプログラム自身と，CGIプログラムなどのWebアプリケーションがある。それらを考慮して，HTTPプロトコルを以下のパートに分解し，パートごとに許可すべき文字長，文字種別，エンコーディングをルールとして設定した。

●HTTPリクエストのヘッダ部
●HTTPリクエストのボディ部
●URL部
●パラメータ部

なお，パラメータとは，CGIプログラムなどのWebアプリケーションに渡される「name=value」形式のデータである。

アプリケーション・ファイアウォールの実装

プロキシサーバとしてhp virtualvault*1) を使用して，アプリケーション・ファイアウォールを実装した。hp virtualvaultは米国TCSECのBレベル相当の高いセキュリティ機能を実装したWebサーバプラットフォームである。hp virtualvaultをプロキシ（逆プロキシ）として使用することにより，アプリケーション・ファイアウォール自身が不正侵入されることを防止することができる。

図5にシステムの概要を示す。hp virtualvaultは，システムの内部が4つの領域に分割されており，各領域間のアクセスは強制アクセス制御機構によりチェックされ，不正アクセスを防止している。

hp virtualvaultでの逆プロキシ動作は次のとおりである。外部からのHTTPリクエストは，SYSTEM OUTSIDE領域の外部向けWebサーバで受け取られ，SYSTEM INSIDE領域のhp webproxy*2) へと送られる。hp webproxyは，HTTPリクエストを背後に設置した目的のWebサーバへ送る。

アプリケーション・ファイアウォールは，hp webproxyの出力を，前述の異常検出アルゴリズムを用いてフィルタリングすることにより実現した。

＊1）hp virtualvaultはHewlett-Packard Companyの商標。　＊2）hp webproxyはHewlett-Packard Companyの商標。

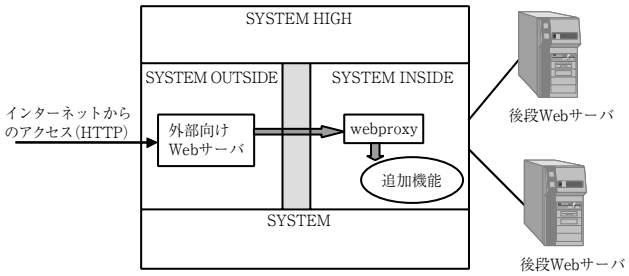

図5　hp virtualvault上に実装したアプリケーション・ファイアウォール

アプリケーション・ファイアウォールの評価

開発したアプリケーション・ファイアウォールを設置した環境において，実際にWebサーバに対して不正アクセスを模擬的に発生させ，アプリケーション・ファイアウォールの遮断効果を確認した。図6に評価環境を示す。

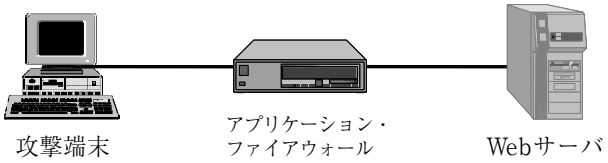

図6　アプリケーション・ファイアウォールの評価環境

攻撃端末から発生させた攻撃は，CodeRed[1]，CodeRedⅡ[2] およびNimda[3] である。

評価結果，すべての攻撃を遮断することができた。ただし，Nimdaは，16種類のパケットを発生するが，その内12種類を遮断している。16種類のパケットは，12種類は攻撃，残りの4種類は攻撃結果としてサーバ上に残したバックドアプログラムとの通信である。本アプリケーション・ファイアウォールは，12種類の攻撃すべてを遮断した。バックドアへの通信リクエスト4種類については遮断していない。これは通信リクエストに特異なデータが含まれておらず，正常時のデータと区別ができないためである。ただし，先行する12種類の攻撃はすべて遮断しているので，バックドアの設定は未実施となり，バックドア通信は実効的に全く効果がなくなっている。

評価にあたって，CodeRed，CodeRedⅡ，Nimdaの各攻撃固有の設定は一切していないことに注目されたい。この事実は，本アプリケーション・ファイアウォールの異常検出アルゴリズムが有効に機能していることを示しており，今後登場する新しい攻撃も防御できることが期待されることを意味する。

ま と め

アプリケーション・ファイアウォールは，ファイアウォールや侵入検知システムでは防御できない不正アクセスから，Webサーバ等のサーバシステムを保護するものである。

本稿で述べたアプリケーション・ファイアウォールは，異常検出の考え方を取り入れた不正アクセス検出アルゴリズムと，セキュアな逆プロキシ型のフィルタリングシステムを用いている。開発したシステムを評価した結果，ファイアウォールを通過するような不正アクセスを遮断し，Webサーバへの攻撃を未然に防止することが確認できた。

考案した異常検出アルゴリズムでは，事前に必要なものは正常時の情報であり，攻撃固有の情報は必要としない。このため，今後新しく登場する攻撃にも有効に働くことが期待される。

本システムは，個人情報を扱うなど高いセキュリティを求める用途に適しており，当社の不正侵入対策ソリューションの一環として提供しているシステムである。

◆◆

■参考文献

1）CodeRed，http://www.cert.org/incident_notes/IN-2001-08.html
2）CodeRedⅡ，http://www.cert.org/incident_notes/IN-2001-09.html
3）Nimda，http://www.cert.org/advisories/CA-2001-26.html

●筆者紹介

露口和弘：Kazuhiro Tsuyuguchi.金融ソリューションカンパニー 金融ソリューション開発本部 ネットセキュリティ開発部
橘喜胤：Yoshitane Tachibana.金融ソリューションカンパニー 金融ソリューション開発本部 ネットセキュリティ開発部
矢野達朗：Tatsuro Yano.金融ソリューションカンパニー 金融ソリューション開発本部 ネットセキュリティ開発部