コンピュータセキュリティ　18－14
(2002.　7.　18)

Webサービスを対象とするワーム流布対策方式の検討

寺田真敏 †　　永井　康彦 †　　倉田盛彦 ‡
terada@sdl.hitachi.co.jp　　y-nagai@sdl.hitachi.co.jp　　m-kurata@itd.hitachi.co.jp

† (株)日立製作所 システム開発研究所
〒224-0817　神奈川県横浜市戸塚区吉田町 292
‡ (株)日立製作所 情報システム事業部
〒100-8220　東京都千代田区丸の内 1-5-1

あらまし：マルウェアの流布を含む不正アクセス活動は活発化しており、また、その被害も広範囲かつ多岐に渡るようになってきている。特に情報システムがWebサービス主体に構成されているイントラネットにおいては、Webサービスを対象とするワームの流布に伴う影響は甚大となる。本稿では、Webサービスを対象とするワームを抑止し、Webサービスの稼動継続性を確保するWebマッパ(Webサービスポート／ホストマッピングシステム)について述べる。Webマッパの特徴は、Webサーバ上のポート切替コンポーネントがWebサービスのポート番号を代替ポート番号にシフトさせることでワームの流布を抑止し、プロキシサーバ上のポート／ホスト変換コンポーネントが代替ポート番号へのシフトに伴うURL変更を隠蔽することでWebサービスの稼動継続性を確保することにある。
キーワード：不正アクセス　ネットワークセキュリティ　ワーム　Webサービス

Examination of the counter measure for the Web service based worm propagation

Masato Terada †　　Yasuhiko Nagai †　　Morihiko Kurata ‡
terada@sdl.hitachi.co.jp　　y-nagai@sdl.hitachi.co.jp　　m-kurata@itd.hitachi.co.jp

† Systems Development Laboratory, Hitachi Ltd.
292 Yoshida-cho, Totsuka-ku, Yokoham, 244-0817 Japan
‡ Information Technology Division, Hitachi Ltd.
1-5-1 Marunouchi, Chiyoda-ku, 100-8220 Japan

Abstract: Unauthorized access containing Malware propagation is activated and causes a lot of damage. Especially, In the information system which consists of Web service based, the influence accompanied by self-propagating worm of Web service based becomes very large. This paper described the overview of the Web mapper's (Web service port / host mapping system) functions, which suppress Web service based worm propagation and support the stable Web service operation. The features of Web mapper are the following. The port change component on the Web server shifts Web service port number to an alternative port number for suppression Web service based worm propagation. The port / host conversion component on proxy server hides the URL change accompanied by a shift for an alternative port number.
key words: Unauthorized Access, Network Security, Worm, Web Service

1. はじめに

インターネットの常時接続の普及に伴い、マルウェア(ウイルス、ワーム、トロイの木馬などの有害な機能を持ったプログラムの総称)の流布を含む不正アクセス活動は活発化しており、また、その被害も広範囲かつ多岐に渡るようになってきている。特に、2001年7月中旬の「Code Red I/IIの流布」、そして2001年9月中旬の「Nimdaの流布」は、情報システムにおける不正アクセス対策をサーバだけではなくクライアントにも実施しなければならないことを教訓として残した。

不正アクセス対策は、ユーザ環境にあわせ、「回避/防止」「保証」「検知」「回復/調査」の4つのフェーズからなる作業を継続的に繰り返しながらセキュリティ強化を図っていく必要があるとされており(図 1.1)[1]、国内でも不正アクセス対策環境は徐々に整いつつある。現状、多くの組織が「回避/防止」としてファイアウォールをはじめとするアクセス制御システムを導入し、セキュリティポリシー策定などの管理面も整備すると共に、「保証」として計算機資源の脆弱性検査や「検知」としてネットワーク型/ホスト型侵入検知システムの導入を進めている。しかし、情報システムの稼動性を確保するためには、「回復/調査」に関する検討も重要であり、実際にマルウェアが流布した際の施策についてはまだまだ検討の余地がある。

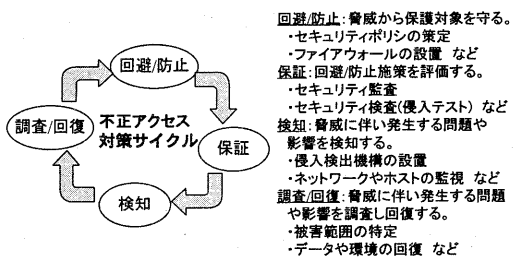

図 1.1 不正アクセス対策サイクル

そこで、本稿では、Webサービスを対象とするワーム流布時の施策として、ワーム流布の抑止、既存サービスの提供維持を実現するためのWebサービスポート／ホストマッピングシステムについて検討したので、その概要について述べる。

2. Webサービスを対象とするワームの動向

マルウェアの活動に伴う脅威が「不正アクセス」に類似した脅威を持ちはじめた1998年以降を中心に全体的な動向と、マルウェアのうちWebサービスを対象とするワームの活動状況について述べる。

2.1 マルウェア全般の動向

(1) 1998年

1998年末から電子メールを介してマルウェア(ウイルス、ワーム、トロイの木馬などの有害な機能を持ったプログラムの総称)[2]が流布するようになってきた。その活動に伴う脅威は「不正アクセス」に類似した脅威を持ち、被害が発生した際には「不正アクセス」被害に相当する施策を実施していく必要がでてきた。

(2) 1999年

1999年に入るとマルウェアの数は激増し、計算機に格納されたユーザ名やパスワード情報を横取りする Picture.exe, K2PS.exe, Y2Kcount.exe、電子メールにマルウェア自分自身を添付し送信してしまう W32/Ska(Happy99), Melissa, W32/PrettyPark, W32/ExploreZip など、「トロイの木馬」「ワーム」に該当する数十種以上のマルウェアが発見されている。特徴としては、著名なサポートセンタ(Microsoft, InfoWeb など)を騙って「トロイの木馬」を添付した電子メールを送付するという、いわゆるソシアルエンジニアリング攻撃の利用や、配布経路として電子メール以外のコミニュケーションチャネル、例えば、IRC(Internet Relay Chat)などの利用が挙げられる。

(3) 2000年

2000年に入るとマルウェアに関連する不正アクセス活動がさらに活発化し始めた。

- ワームの台頭

VBS/LoveLetter, Stages, SirCam, MTX, Hybris, Navidad, W32/Nimda など、電子メールを介して自己拡散する数十種類のワームが発見された。

- Windows ネットワークファイル共用の利用

ワームは人手の介入の必要のない自己拡散の方法として、Windows ネットワークファイル共用を利用し始めた(例：Network, 911.Worm, QAZ, Magistr, W32/Nimda)。

- ワーム機能を備え始めたトロイの木馬

リモートから操作可能なバックドア機能を持つ「トロイの木馬」が、電子メールやバックドアを介した自己拡散機能を備え始めた(例：QAZ)。これに伴い、不正アクセスの直接的な被害を受けたり、踏み台による間接的な被害をもたらす可能性が高まってきた。

- ソシアルエンジニアリング攻撃の利用

VBS/LoveLetter ワームをはじめとする多くのマルウェアにおいて、電子メールの送信者が良く知っている送信者で、さらに、電子メールの内容が受信者の興味を引く内容を送付するという、いわゆるソシアルエンジニアリング攻撃が活用されている。

(4) 2001 年

2001 年に入ってからは電子メールを介して自己拡散するワームに加え、サーバプログラムの脆弱性を直接攻撃するワーム(CodeRed, Nimda など)、クライアントの脆弱性を悪用したダイレクトアクション型ワーム(Nimda, Aliz, Klez など)も現れ、人手の介入を必要としない自己拡散の方法が主流となり始めた。

- Microsoft IIS サーバの脆弱性への攻撃

sadmind/IIS, Code Red, W32/Nimda ワームなど、Microsoft IIS サーバの脆弱性を攻撃対象とするワームの流布が際立っている。2001 年 5 月の sadmind/IIS では 10,000 台近くのサーバが被害にあい、2001 年 7 ～8 月の Code Red ワームに至ってはネットワークの帯域を枯渇させるサービス不能状態を引き起こすと共に、少なくとも 300,000 台以上のコンピュータに影響を与えたと言われている。また、ワームによる被害波及は短期間のうちに脆弱なコンピュータを探し出し、その結果として指数関数的な自己拡散を実現した[3]。これまでの不正アクセス活動は単発的な攻撃の n 倍化であり、セキュリティ侵害の実現に数週間あるいは数か月かかったが、ワームの場合には数分あるいは数時間の間に何万ものコンピュータのセキュリティ侵害が可能となってしまっている。

特に、2001 年 9 月の W32/Nimda ワームの流布は、(a) Code Red や Code Blue ワームのように脆弱な Microsoft IIS サーバを探して感染する、(b) SirCam や Apost のように大量に電子メールを送信して感染する、(c) Magistr のようにオープンなネットワーク共有ドライブを探して感染する、そして、(d) 悪質な Web ページコンテンツを使って感染するという、何通りもの方法でウイルスを広げることが可能であるということを証明してしまったと言える。

- UNIX サーバの脆弱性への攻撃

マルウェアの流布は、Microsoft IIS サーバに限られたものではなく、UNIX サーバの脆弱性を攻撃対象とするマルウェアも流布している。特に、Linux システムを攻撃対象とするワームとして、Linux Ramen ワーム(2001 年 1 月)、Lion ワーム(2001 年 3 月)、Adore ワーム(2001 年 4 月)が流布した。

2.2 Web サービスを対象とするワームの活動状況

インターネット上に公開している Web サイトから回収したログデータをもとに、Web サービスを攻撃対象とするワームである、CodeRed I, CodeRed II, Nimda の活動状況について再考する。

2.2.1 活動状況の調査方法

(1) 回収したログデータ期間

2001 年 7 月 15 日～12 月 31 日 (ただし、Web サイトによっては、ログデータの記録が残されていない期間も回収期間に含まれている。)

(2) ログデータを回収した Web サイト

下記の 5 つの Web サイトからログデータを回収した。

- No.1 210.229.xxx.xxx (Apache)
- No.2 211.0.xxx.xxx (Apache)
- No.3 211.14.xxx.xxx (Apache)
- No.4 202.210.xxx.xxx (Apache)
- No.5 192.35.xxx.xxx (Apache)

2.2.2 調査結果

(1) 活動状況に関する諸データ

回収したログデータに基づく、各ワームの活動状況を図 2.1に示す。

(2) 活動状況に関する考察

項番(1)の諸データに基づく各ワームの活動状況比較を表 2.1に示す。国内で被害の大きかった CodeRed II, Nimda については、最初の痕跡記録時刻から最頻アクセス数となった日までわずか 2 日間程度であり、ワームによる被害波及が短期間かつ広範囲に渡っていたことが推測できる。

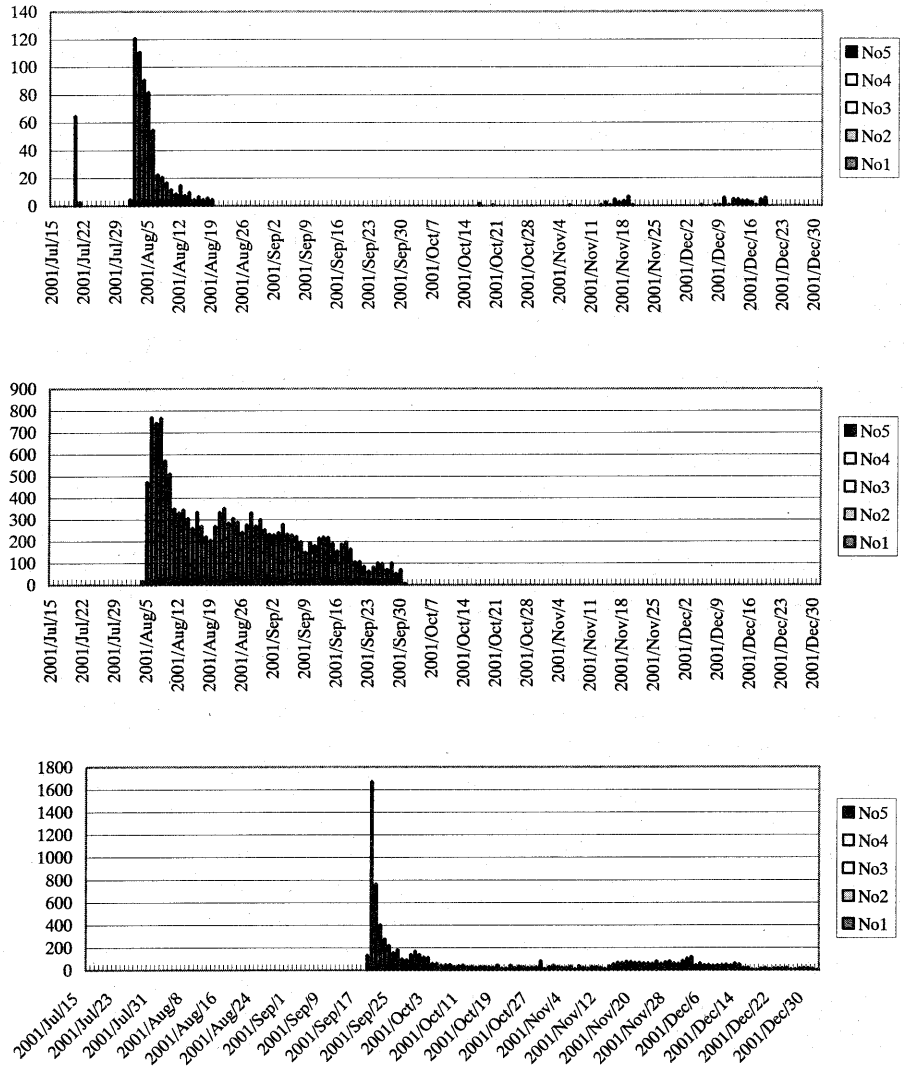

図 2.1　ログ回収期間内のアクセス数変位 (上段：CodeRed I, 中段：CodeRed II, 下段：Nimda)

表 2.1　5 つの Web サイトのログに基づく各ワームの活動状況の比較

	CodeRed I	CodeRed II	Nimda
最初の痕跡記録時刻	7 月 20 日 00:28:53	8 月 4 日 20:21:40	9 月 18 日 22:10:34
アクセス総数	748 件	14,992 件	8,651 件 (注)
最頻アクセス数	121 件 (8 月 2 日)	771 件 (8 月 6 日)	1647 件 (9 月 19 日)
継続性 (100 件以上/日)	2 日間(8 月 2 日～3 日)	48 日間(8 月 5 日～9 月 21 日)	15 日間(9 月 18 日～10 月 2 日)
発信元ドメイン (頻度順)	com, net	jp, kr, tw	kr, tw, jp

注) Nimda のアクセス数は、Nimda の 1 回のアクセス活動数で算出しているため、Web サーバへの実際のアクセス総数は、16 倍の約 138,416 件となる。

3. Webサービスポート／ホストマッピング

本章では、CodeRed, Nimdaなどのワーム流布時に、ワームによるトラフィック増加の抑止とWebサービスの継続性を確保するためのWebサービスポート/ホストマッピングシステム(以下、Webマッパ)について述べる。

3.1 Webサービスを攻撃対象とするワーム流布時の課題と解決策の提案

(1) 既存対策の課題

ワームが流布した際の基本的な対策手段は、ウイルス対策ベンダの提供するアンチウイルスソフトウェアのウイルス定義ファイルを更新すると共に、脆弱なサービスが稼動している場合には、セキュリティ修正プログラムによる脆弱性の除去を行なうか、サービス自身を無効とすることである。また、システムが不幸にも感染してしまった場合には除去ツールを適用したり、初期からシステムを再構築することになる。

ところが、現状の情報システムの多くがWebサービス主体に構成されているために、Webサービスを攻撃対象とするワームが流布した場合、対策が完全に完了するまでの間、以下のような対策上の課題を伴ってしまう。

- Webによるサービスを提供していること自体がワームの流布ならびに、流布に伴うトラフィック増加を助長してしまう可能性がある。
- ワームがWebサービスを攻撃対象としているために、Webによる対策情報の発信や、既存Webサービスの稼動が阻害されてしまう。

特に情報システムがWebサービス主体に構成されているイントラネットにおいて、影響は甚大となる。

(2) 課題解決のアプローチ

上記課題を解決するためには、ワームの流布を抑止することと、Webサービスの稼動継続性を確保することの二面性を兼ね備えた対策が必要となる。そこで、以下に示すような施策により実現する方法を提案する。

- ワームの流布を抑止する。

ワームが攻撃対象としているWebサービス(80/tcp)へのトラフィックをルータやファイアウォールで遮断する。

- Webサービスの稼動継続性を確保する。

ワームが攻撃対象としているWebサービスを代替ポート(例えば、9999/TCP)を用いて提供する。

また、Webサービスが代替ポートに切り替わったことに伴う影響を最小限に留めるために、例えば、プロキシサーバにおいて、既存ポート番号と代替ポート番号とのマッピングを行なうなどの施策を適用する。

3.2 Webマッパの概要

Webマッパは、課題解決のアプローチで示した方法を組み合わせることにより、Webサービスを対象とするワーム流布を回避するためのシステムである。

3.2.1 Webマッパ適用にあたっての前提条件

- 同一の管理ドメインを適用対象とする。

Webマッパの適用にあたってはネットワーク構成変更を伴うため、適用範囲を管理可能な範囲内、例えば、イントラネットなどの組織内ネットワークなどに限定しなければならない。

- Webブラウザからの Webアクセスは、全てプロキシサーバ経由とする。

Webマッパ適用の効果をあげる施策として前提条件とした。

3.2.2 Webマッパのコンポーネント

Webマッパは、以下の4つのコンポーネントから構成する(図 3.1)。

(1) ポートフィルタリングコンポーネント

Webサービスを攻撃対象とするワームが流布した際に、Webサービスを提供しているポート番号(80/tcp)へのトラフィックをフィルタリングする。フィルタリングにあたっては、既存ネットワーク機器であるルータ、ファイアウォールを用いることを想定している。

(2) ポート切替コンポーネント

ワームが攻撃対象としているWebサービスを代替ポート(例えば、9999/tcp)を用いて提供するために、ポート番号の切替を行なう。Webサーバの多くは、標準ポート番号(80/tcp)以外でもサービスを提供することができるようになっており、定義ファイルの書

き換えと再起動操作のみで、ポート切替を行なうことができる。実現方法の一例については、項番3.3.1において述べる。

(3) ポート/ホスト変換コンポーネント

Web サービスが代替ポートに切り替わったことに伴う影響を最小限に留めるために、既存ポート番号と代替ポート番号とのマッピングを行なう。本機能は、URL の rewriting 機能でありプロトタイプ開発を行なった。プロトタイプについては、項番3.3.1において述べる。

(4) 管理コンポーネント

上記 3 コンポーネントに対して、フィルタリング、ポート切替、ポートマッピングの実施あるいは、解除指示を出す。管理者によるマニュアル操作やIDS(Intrusion Detection System)との連動を想定している。

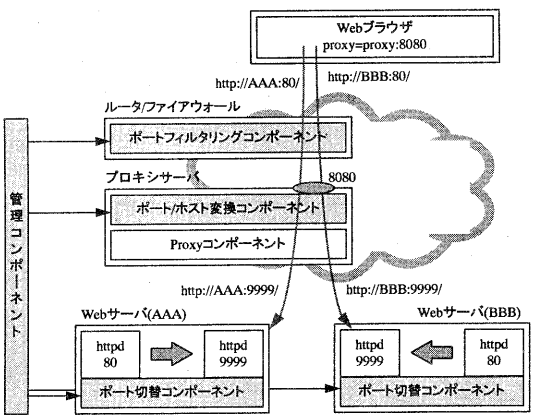

図 3.1 Web サービスポート/ホストマッピングシステム

Webマッパを介した場合のWebブラウザからWebサーバまでのアクセス経路概要を図 3.2に示す。

①Web ブラウザ−>プロキシサーバ

Web ブラウザからのアクセスは、全てプロキシサーバ経由であり、プロキシサーバのポート番号8080/tcp に対して、HTTP 要求(例：GET http://AAA/index.html)を送信する。

②プロキシサーバ

プロキシサーバ上の「ポート／ホスト変換コンポーネント」では、HTTP 要求を定義ファイル(ポート/ホスト変換テーブル)に従い書き換えを行なった後、Proxy コンポーネントに HTTP 要求(GET http://AAA:9999/index.html)を転送する。ここでは、プロキシサーバへのアドオンコンポーネントとしているが、Proxy コンポーネント自身に「ポート／ホスト変換コンポーネント」を組み込むことも可能であるし、また、クライアント上に組み込むこともできる。

③プロキシサーバ−>Web サーバ

Proxy コンポーネントでは、書き換え後の HTTP 要求(GET http://AAA:9999/index.html)に従い、Web サーバのポート番号 9999/tcp に対して HTTP 要求を送信する。

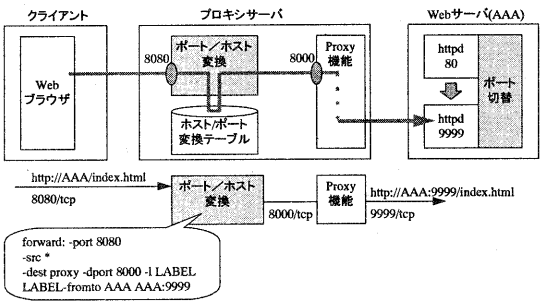

図 3.2 Web マッパ適用時の Web ブラウザから Web サーバまでのアクセス経路

このように、Web マッパの基本的な仕組みは、プロキシサーバ上の「ポート／ホスト変換コンポーネント」と Web サーバ上の「ポート切替コンポーネント」が連動して Web サービスのポート番号(80/tcp)を代替ポート番号(9999/tcp)にシフトすることにより、ワームの流布を抑止し、Web サービスの稼動継続性を確保する。さらに、プロキシサーバ上の「ポート／ホスト変換コンポーネント」を介することで、代替ポート番号(9999/tcp)にシフトしたことによるURL 変更の隠蔽を行なう。これにより、ユーザに対して既存 Web サービスの稼動環境を維持することができることになる。

3.3 Web マッパのプロトタイプ

Web マッパの 4 つのコンポーネントのうち、「ポート切替コンポーネント」「ポート／ホスト変換コ

ンポーネント」について、実現方式の検討ならびにプロトタイプの開発を行なった。

3.3.1 実現方式

(1) ポート切替コンポーネント

ポート切替コンポーネントについては、Web サーバとして Apache[4]を想定した場合、標準ポート番号による Web サービス用(図 3.3)と、代替ポート番号による Web サービス用(図 3.4)の 2 種類の定義ファイルを用意し、これらを切替え用スクリプトで制御することにより実現できる。

```
# Listen: Allows you to bind Apache to specific IP addresses
# and/or ports, in addition to the default. See also the
# <VirtualHost> directive. Change this to Listen on specific IP
# addresses as shown below to prevent Apache from glomming
# onto all bound IP addresses (0.0.0.0)
Listen 80
```

図 3.3 標準ポート番号用 Apache サーバの定義ファイル httpd.conf (一部)

```
# Listen: Allows you to bind Apache to specific IP addresses
# and/or ports, in addition to the default. See also the
# <VirtualHost> directive. Change this to Listen on specific IP
# addresses as shown below to prevent Apache from glomming
# onto all bound IP addresses (0.0.0.0)
Listen 9999
```

図 3.4 代替ポート番号用 Apache サーバの定義ファイル httpd9999.conf (一部)

(2) ポート/ホスト変換コンポーネント

ポート/ホスト変換コンポーネントは URL の rewriting 機能であるが、今後の機能拡張と試行運用を踏まえ、以下に示す機能を持つプロトタイプ hwmapd の開発を行なった。

- ポート／ホストマッピング機能

Web ブラウザから受信した HTTP 要求に対して、定義ファイルに指定している変換定義に従いポート番号ならびに、ホスト名の書き換えを行う。具体的には、HTTP 要求ヘッダのメソッド行と Host 行が、定義ファイルに指定された「変換前ホスト名：変換前ポート番号」に合致する場合、「変換後ホスト名：変換後ポート番号」に変換した後、転送を行なう(図 3.5)。

- 送信元に対するアクセス制御機能

定義ファイルにて許可されたクライアントからの HTTP 要求に対してのみ、ポート／ホストマッピングならびに、HTTP 要求の転送を行なう。

- アクセスログ機能

ポート／ホストマッピング機能で処理した HTTP 要求のログを取得する。

```
# 書き換えを行うHTTP要求のホスト名とポート番号を指定
forward:
        -port待ちポート
        -src 送信元IPアドレス
        -dest 転送先サーバIPアドレス
        -dport 転送先サーバのポート番号
        -l ルールラベル
ルールラベル
        -fromto 変換前IPアドレス/ホスト名：ポート番号
        変換後IPアドレス/ホスト名：ポート番号
--------------------------------------------------------------------------
[ 定義ファイル例 ]
forward: -port 8080 -src * -dest proxy -dport 8000 -l LBL
LBL-fromto tomato.sd.hitachi.co.jp   kiwi.sdl.hitachi.co.jp:9999
```

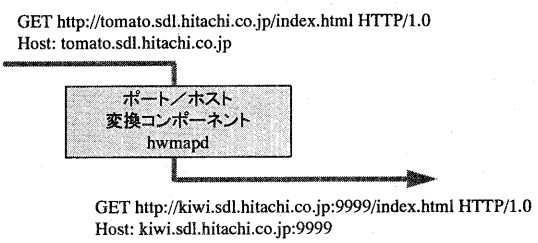

図 3.5 ポート/ホスト変換コンポーネント hwmapd の定義ファイルと書き換え処理

3.3.2 プロトタイプの評価と考察

(1) ワームの流布に伴うトラフィック抑止について

以下の条件で、CodeRed I, II, Nimda の送出する TCP パケット数を机上算出すると、ポート切替を行なう前後の差は表 3.1の通りとなる。あくまでも机上算出のレベルに留まるが、Web サービスの提供するポートを切替えるだけでワームの流布に伴うトラフィック低減を期待することができる。

[算出条件]

- CodeRed I, II, Nimda の送出する HTTP 要求はひとつの TCP パケットに収まる。
- HTTP 要求に対しては ACK フラグの設定された TCP パケットは返送されない。

表 3.1　ワームが送出する TCP パケット数の比較

	ポート切替前(80/tcp)	ポート切替後(9999/tcp)
CodeRed I	8 パケット SYN, SYN+ACK, ACK, HTTP 要求, FIN, ACK, FIN, ACK	2 パケット SYN, RST
CodeRed II	8 パケット 同上	2 パケット 同上
Nimda	128 パケット 16 個の HTTP 要求が送信されるため、8x16 パケットとなる。	2 パケット 同上

(2) URL 変更の隠蔽について

代替ポート番号(9999/tcp)でサービスを提供している Web サーバに対して、ポート/ホスト変換コンポーネント hwmapd を介して、下記の 5 つの形態でのアクセスを行なった結果、図 4.1に示す通り、標準ポート番号(80/tcp)へのアクセスで代替ポート番号(9999/tcp)にアクセスしていることと、いずれの場合も代替ポート番号にシフトしたことによる URL 変更を、ポート/ホスト変換コンポーネント hwmapd により隠蔽できていることを確認した。

- 環境変数表示用 CGI プログラムへのアクセス
- 相対パス記述の URL へのアクセス
- 絶対パス記述の URL へのアクセス
- ホスト名＋ポート番号記述の URL へのアクセス
- JavaScript によるホスト名記述の URL へのアクセス

また、今回、今後の機能拡張を考慮し「ポート/ホスト変換コンポーネント」として hwmapd の開発を行なったが、同等の機能は Apache の既存機能(rewrite, proxy 機能)を用いて実現できることも確認した。

4. おわりに

本稿では、Web サービスを対象とするワーム流布時の施策として、Web サービスの標準ポート番号を代替ポート番号にシフトすることにより、ワーム流布の抑止、既存サービスの提供維持を実現する方式と開発したプロトタイプについて述べた。現在、Web マッパの試行運用を進めており、今後、以下に示す課題を検討していく予定である。

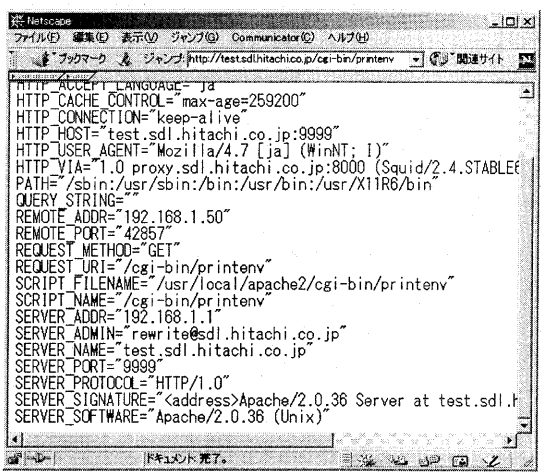

図 4.1　hwmapd を介した環境変数表示用 CGI プログラムへのアクセス結果

- Web マッパの機能拡張として、https 対応ポート/ホスト変換コンポーネントの試作を行なうと共に、Web マッパの試行運用による運用施策の検討や運用ノウハウの蓄積を行なっていく。
- Web サービスを攻撃対象とするワームの変種、例えば、プロキシサーバを攻撃対象としたワームやプロキシサーバを乗り越えるワームへの対処方法を検討していく。

参考文献

1) 不正侵入はこう防げ, 日経コンピュータ, No.448, pp185-195, July 1998

2) EICAR99 (European Institute for Computer Anti-Virus Research), http://www.ipa.go.jp/security/fy10/contents/virus/3_1_3.html

3) CERT Advisory CA-2001-23: Continued Threat of the "Code Red" Worm http://www.cert.org/advisories/CA-2001-23.html

4) The Apache Software Foundation , http://www.apache.org/