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

2006－DPS－126（34）
2006－CSEC－32（34）
2006／3／17

トラフィック解析によるスパイウェア検知システムの提案

与那原 亨　　大谷 尚通　　馬場 達也　　稲田 勉

株式会社NTT データ　〒104-0033 東京都中央区新川1-21-2 茅場町タワー

E-mail:　{yonaharaa, ootanihs, babatt, inadatt}@nttdata.co.jp

あらまし　スパイウェアの被害が急速に拡大しており，今後，その脅威が企業にも広がることが予想される．一方，スパイウェアはユーザに発見されないように侵入し，巧妙に潜伏することから，対策が難しくなってきている．現状のスパイウェア対策は，PC に対策用ソフトを導入する方法が一般的であるが，特に企業では，全ての PC に対してその導入や管理の徹底が難しいという問題がある．本稿では，ネットワーク上において，トラフィック解析を行い，スパイウェアなどの不正なプログラムから送信されるトラフィックを検知・防止するシステムを提案する．

キーワード　スパイウェア，トラフィック解析，検知システム

A Proposal of Spyware Detection using Traffic Analysis

Akira YONAHARA　Hisamichi OHTANI　Tatsuya BABA　and　Tsutomu INADA

NTT Data Corporation　Kayabacho Tower, 1-21-2, Shinkawa, Chuo-ku, Tokyo, 104-0033 Japan

E-mail:　{yonaharaa, ootanihs, babatt, inadatt}@nttdata.co.jp

Abstract　Currently, The damage of the spyware expands rapidly, its threat is expected to extend to the enterprise network in the future. However, spyware intrude so as not to be discovered by the user and hide cleverly, therefore countermeasures for spyware have been becoming difficult. The general countermeasures for spyware is the method of introducing anti-spyware software into all PC. In the enterprise network, there is a problem that the introduction and management is difficult. In this paper, we propose spyware traffic detection and prevention using traffic analysis

Keyword　Spyware，Traffic Analysis，Detection System

1. はじめに

近年，ユーザの端末に忍び込み，ユーザに気づかれないように情報を収集し，外部へ送信するスパイウェアの被害が拡大してきている．初期のスパイウェアは，URL 履歴情報や Web 上でのキー入力などを無作為に収集するため，被害が顕在化することは少なかった．しかしながら，最近では，クレジット番号やパスワードや PC 内の個人情報といった重要な情報を盗み出すといった悪意のあるスパイウェアが急速に増加してきている．さらには，ソーシャルエンジニアリング的な手法を用いて限られた範囲のユーザを狙ったスパイウェアも出現している．スパイウェアによる被害は，今後，企業の PC にも広がることが予想され，その場合，感染すると社員情報顧客情報などの個人情報だけではなく，知的財産などの機密情報の漏えいに繋がる可能性がある．

本稿では，企業において最も脅威となるスパイウェアによる情報漏洩への対策として，これまで難しかったネットワーク上でスパイウェアからの情報送信を検知・防止するシステムを提案する．さらに，作成したプロトタイプによる評価結果を報告する．

2. スパイウェア対策の現状と課題

2.1 スパイウェアによる被害

一般にスパイウェアと呼ばれるものは種類が多岐にわたっている．その主な種類を以下に示す．

キーロガー：キー入力情報などを記録するプログラム．多くのキーロガーは，記録した情報をネットワーク経由で送信する機能を持っている．

アドウェア：ウインドウをポップアップして広告を表示させるプログラム．Web 閲覧履歴や検索キーワードを収集して広告を表示するものが多い．

リモートアクセスプログラム：PC を遠隔制御できるようにするプログラム．これにより，ファイルの内容を参照したり，スクリーンショットを撮ったりすることが可能となる．

ブラウザハイジャッカー：ブラウザを乗っ取ってしまうプログラム．ブラウザ起動時に表示するホームページを変更したり，閲覧しようとするページとは異なるページへ強制的に誘導させる．

企業において，これらのスパイウェアによって引き起こさ

れる被害としては，以下のようなものがある．

- PC 内のファイルやキー入力情報などを外部へ送信され，個人情報や企業の機密情報の漏えいする．
- PC のリソース（CPU，メモリ等）を勝手に使用され，PC の動作が不安定になり，業務・生産性に影響する．
- PC の制御が奪われ，DoS 攻撃や迷惑メールの発信拠点になり，企業のイメージダウンや信用失墜に繋がる．

企業にとって最も大きな脅威となるのは，PC 内の情報が外部の第三者へ送信されることである．情報が外部へ漏洩されるとその被害だけでなく，企業の信用低下につながる．特に企業にとっては社会的信用を失墜させ、法的責任を問われる事態を招きかねない．

2.2 現状の主な対策

(1) 侵入対策

スパイウェアの侵入対策として，企業が導入する一般的な方法としては，スパイウェア対策機能を実装したウイルス対策ソフトや専用のスパイウェア対策ソフトを導入することがあげられる．これらの対策ソフトは，PC 上にて検知・除去を行うホスト型と，ゲートウェイ装置等として動作するネットワーク型が存在する．いずれの場合も主にシグネチャマッチング方式が用いられる．

(2) 侵入後の被害防止対策

スパイウェアに侵入された場合の対策としては，スパイウェアによる外部への情報送信を防ぐ必要がある．これには，ファイアウォールを導入し，許可するポートを設定することで，Web ブラウジング（HTTP）やメール（SMTP）などのユーザが通常利用する通信のみを通過させ，ユーザの意図しないスパイウェアからの通信を遮断する方法が考えられる．

ファイアウォールのポート番号に基づいた制御では遮断できない外部との通信については，PC 内に導入するパーソナルファイアウォールのプログラム別の通信制御機能や URL フィルタリングで防ぐことができる．パーソナルファイアウォールでは，許可されていないプログラムから通信が行われると，画面に警告が表示され，ユーザが許可/不許可の判定を行う．URL フィルタリングでは業務に関係のないサイトへのアクセスをブロックする．

2.3 現状の対策の課題

現状のスパイウェア対策ソフトではシグネチャマッチング方式が多く用いられているため，これまでのウイルス対策と同様に未知のものには対応できないという問題がある．さらには，限られた範囲のユーザを狙ったスパイウェアに対しては，セキュリティベンダが検体（サンプル）を入手してシグネチャを作成することが難しく，対策ソフトによる検知が困難である．このため，現状では，全てのスパイウェアの侵入は防ぐことが難しい．

外部への情報の漏洩を防ぐためには，企業への導入率の高いファイアウォールを使う方法が考えられる．しかしながら，多くのスパイウェアが業務で使用する HTTP や SMTP を用いて外部へ情報を送信するため[1][2]，ファイアウォールだけで遮断することはできない．スパイウェアによる HTTP や SMTP の通信を防ぐためには，PC 内にパーソナルファイアウォールを導入し，スパイウェアなどの不正なプログラムからの通信を遮断することになるが，実際には企業内において全ての PC に対策を徹底させるのは難しく，その許可/不許可をきちんと判別できるユーザは少ない．また，HTTP の場合は，URL フィルタで防御できる可能性があるが，新種のスパイウェアの通信先を、あらかじめ登録することは不可能である．

3. 提案方式

ネットワーク上で観測したトラフィックについて，PC 内のスパイウェアからの通信かどうかを判別し，スパイウェアによる情報の漏洩を検知・防止するシステムを提案する．

3.1 方式の方向性

本方式では，ファイアウォールなどで防ぐことが難しい HTTP と SMTP を用いたスパイウェア（キーロガー，アドウェア，リモートアクセスプログラムなど）からの情報送信をネットワーク上で検知・防止する．

HTTP や SMTP などのプロトコルを利用したスパイウェアの通信は，プロトコルシーケンス的には通常のトラフィックとスパイウェアのトラフィックに違いは無く，区別することが困難であることが分かっている[1]．このため，プロトコルレイヤでのスパイウェアの通信を検知・防止は難しい．

アプリケーションレイヤでやりとりされる通信では，スパイウェアの目的は情報を外部へ送信することであるため，最低限のプロトコルを実装できていればよく，不必要な高度な機能を実装していない．このため、必要のない情報を付加しない，あるいはスパイウェア自身が理解できる情報を付加する特徴がある．これに対し，ユーザは PC から HTTP や SMTP を用いてサーバと通信する場合に InternetExplorer や Outlook といったプログラムを使用する．このため，アプリケーションデータに設定される情報に違いが出てくる．表 1と表 2にユーザが普段使用するプログラムとスパイウェアに設定されるアプリケーションヘッダの例を示す．これより，アプリケーションデータの内容をチェックすることで，スパイウェアからの通信かどうかの判別を行うこととする．

表 1　HTTP アプリケーションヘッダ例

InternetExplorer6	Firefox1.5	Spyware (Gator)
POST /** HTTP/1.1 Accept: Referer: Accept-Language: Content-Type: Accept-Encoding: User-Agent: Host: Content-Length: Connection: Cache-Control:	POST /** HTTP/1.1 Host: User-Agent: Accept: Accept-Language: Accept-Encoding: Accept-Charset: Keep-Alive: Connection: Referer: Content-Type: Content-Length:	POST /** HTTP/1.1 Accept: Content-Type: X-UA: User-Agent: Host: Content-Length: Connection: Cache-Control:

表 2　SMTP アプリケーションヘッダ例

OutlookExpress6	Thunderbird1.5	Spyware (Perfectkeylogger)
Message-ID: From: To: Subject: Date: MIME-Version: Content-Type: X-Priority: X-MSMail-Priority: X-Mailer: X-MimeOLE:	Message-ID: Date: From: User-Agent: MIME-Version: To: Subject: Content-Type:	From: To: Subject: Date: X-Mailer: Content-Transfer-Encoding: MIME-Version: Content-Type:

また，PC からスパイウェア側のサーバに送信する HTTP リクエストや SMTP の DATA コマンドといったアプリケーションデータ内に漏洩を防止したい情報が含まれていることがある．この場合，そのアプリケーションデータがサーバに届いた時点で情報が漏洩してしまう危険性がある．このため，PC とサーバ間のアプリケーションデータの複数のやりとりから総合的にスパイウェアの通信を判別するのでは，情報は既にスパイウェアのサーバ側で保存されており，情報漏洩という観点では防止できていない．これより，PC から送信されるアプリケーションデータのみを観測し，スパイウェアの検知・防止を行うこととする．

さらに，ユーザが Web ブラウジングやメールの送信を行うときは大抵 PC の前でその作業を行う．このため，システム側でスパイウェアの判定が難しい場合には，ユーザ自身が送信したものかどうかを問い合わせることが有効である．ユーザ自身が直前に送信したものであれば，PC の前にいるユーザはこの問い合わせに対し即答ができる．逆に，ユーザの意思とは無関係にバックグラウンドでサーバと通信を行うスパイウェアはこの問い合わせに対応することができない．このため，システム側でスパイウェアかどうかの判別が難しい場合には，ユーザへの送信確認を行うこととする．

3.2 アルゴリズムの提案

PC から送信されたアプリケーションデータがスパイウェアによる通信データかどうかを判別するする方法として，3 つのマッチングとユーザへの確認応答を組み合わせる．全体のフローを図 1に示す．

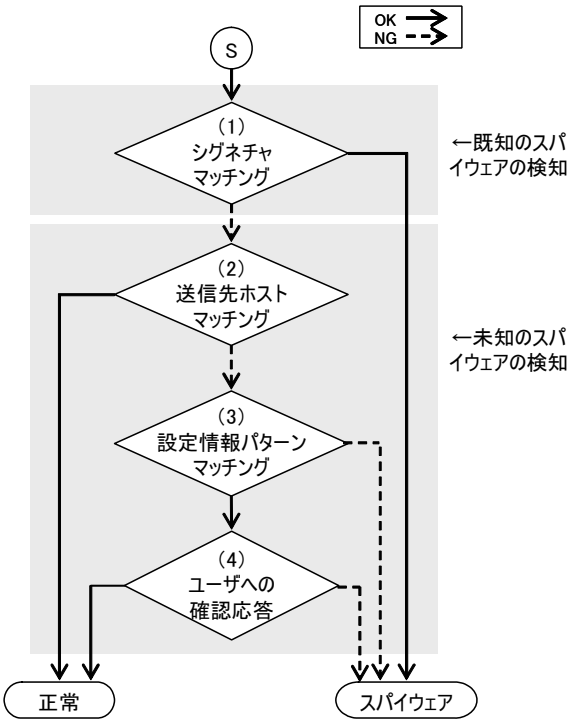

図 1　検知機能のフロー

(1) シグネチャマッチング

アプリケーションデータに設定される文字列（例えば，User-agent:Gator など）が，あらかじめ用意したシグネチャとマッチした場合に“スパイウェア”からの通信と判別し，遮断する．これにより，既知のスパイウェアでその設定さえる情報の特徴が分かっているものを確実に検知できる．

(2) 送信先ホストマッチング

送信先ホストのドメイン名があらかじめ用意したホワイトリスト内のものとマッチした場合に“正常”と判別し，通信を許可する．マッチしなかった場合は，次の「(3)設定情報パターンマッチング」へ移行する．ホワイトリスト方式を用いることで，許可した通信のみを確実に通過させることができる．送信先ホストについては，IP ヘッダに設定される送信先 IP アドレスでは，企業の場合，全てプロキシサーバやメールサーバになることがあるため，アプリケーションデータに設定されるドメイン名とした．ドメイン名の収集先を表 3に示す。

表 3　収集する送信先ホストのドメイン名

HTTP	URI に設定される情報あるいは Host ヘッダに設定される情報
SMTP	RCPT TO コマンドに設定される情報

(3) 設定情報パターンマッチング

表 1と表 2に示したようなユーザが普段使用するプログラムとスパイウェアとに設定されるアプリケーション情報の異なると想定される情報を抽出し，それをパターン化する（設定情報パターン）．設定情報パターンを生成する情報を表 4に示す．

表 4　設定情報パターンの設定情報

HTTP	・Method 種別 ・ホスト名設定箇所(URI/Host ヘッダ) ・HTTP バージョン ・アプリケーションヘッダの種別と順序
SMTP	・開始コマンド(EHLO/HELO) ・本文と添付ファイルの順序(添付ファイルが無い場合は本文) ・アプリケーションヘッダの種別と順序

以下に，設定情報パターンの例を示す．

- HTTP の場合（表 1の InternetExplorer6 の場合）

POST,HOST,1.1,Accept,Referer,Accept-Language,Content-Type,Accept-Encoding,User-Agent,Host,Content-Length,Connection,Cache-Control

- SMTP の場合（表 2の Thunderbird1.5 の場合）

EHLO,ATTACH,Message-ID,Date,From,User-Agent,MIME-Version,To,Subject,Content-Type

設定情報パターンがあらかじめ用意したホワイトリスト内のパターンとマッチしなかった場合は，シグネチャにない未知の“スパイウェア”と判別し，遮断する．マッチした場合は，InternetExplorer や Outlook などを利用して情報を送信するスパイウェアの可能性があるため，システム側で“正常”と判別せずに，「(4)ユーザへの確認応答」へ移行する．

(4) ユーザへの確認応答

PC へ確認メッセージを送信し，「(3)設定情報パターンマッチング」においてシステム側で判別しなかったアプリケーションデータをユーザ自身が送信したかどうかの問合せを行う．このメッセージに対して一定期間内に応答があったものは，“正常”と判断し，通信を許可するとともに，送信先ホストのドメイン名をホワイトリストに登録する．これにより，次回からは“正常”と判別されるようになる．一定時間内に応答が無かった場合には“スパイウェア”からの通信と判別し，通信を遮断する．

3.3 ホワイトリスト生成機能

ホワイトリスト方式の場合，一般にホワイトリストの作成・維持更新方法が問題になる．このため，一定期間毎に正常と判断されたアプリケーションのトラフィックログを統計処理し，前記 2 つのホワイトリストを作成・更新することとする．今回は，以下のような方法を用いた[3].

- トラフィックログから送信先ホストのドメイン名および設定情報パターン種別を抽出し，それぞれに信頼度を設定する．
- トラフィックログへの送信先ホストのドメイン名，設定情報パターンの出現回数に応じて，信頼度の値を増加させる．また，信頼度は，時間経過とともに指数的に減少させる．
- 信頼度が一定値以上の送信先ホストのドメイン名，設定情報パターンを抽出し，それぞれのホワイトリストに更新する．

具体的には，以下のような計算式を用いることとした．

- 信頼度の最大値は 1.0 とする．
- 新規登録のものは 0.5 とする．
- 信頼度の減少式
  (新信頼度) = (信頼度)/2
- 信頼度の回復式
  (新信頼度) = (信頼度) + (1-信頼度)/2

これにより、そのネットワーク環境やユーザ環境に合った最適なホワイトリストが自動で生成・更新できる．

4. 提案方式の評価

4.1 プロトタイプの実装

提案方式を実装したプロトタイプのセンサ（スパイウェアセンサ）を作成した．FedoraCore4 上で，センサを通過するパケットを libipq ライブラリを通して収集し，アプリケーションデータの判定を行うプロセスを実装した．また，ユーザへの確認機能は，smbclient コマンドを利用し確認メッセージ（送信先ホスト，アプリケーション情報などを含む）を送信するようにした．なお，PC 側では，Windows 標準搭載の Messenger 機能を常時起動させ，確認メッセージを画面上にポップアップできるようにした．ホワイトリスト作成・更新プロセスは判定プロセスとは実装し，特定の時間帯あるいは管理者自身が特定の契機に更新処理を実行できるようにした．

4.2 検証方法および検証実験

本方式の有効性を検証するため，擬似オフィスネットワークを構築し，実際にユーザが作業する PC を設置したセグメントからインターネットへ通じる箇所に作成したスパイウェアセンサをインラインにて設置した．検証では，その検知精度として，通常業務における誤検知（False Positive）と実際に PC 内でスパイウェアを動作させたときの検知漏れ（False Negative）を測定した．

(1) ホワイトリスト作成実験

まず，検知精度を測定するにあたり，ホワイトリストを作成した．図 2のように，セグメント内に PC を 6 台，ネットワーク側にプロキシサーバ，メールサーバ，Web サーバを設置し，6 名のユーザが擬似的に通常業務を行った．1 週間（5 営業日），HTTP と SMTP のトラフィックをロギングし，送信先ホストと設定情報パターンのホワイトリストを作成した．ホワイトリストの更新については 1 日毎に行った．最終的に生成されたホワイトリストに登録されたエントリ数を表 5に示す．なお，ホスト名のドメイン名は．区切りで 3 桁でマージすることとした．（例，*.nttdata.co.jp）

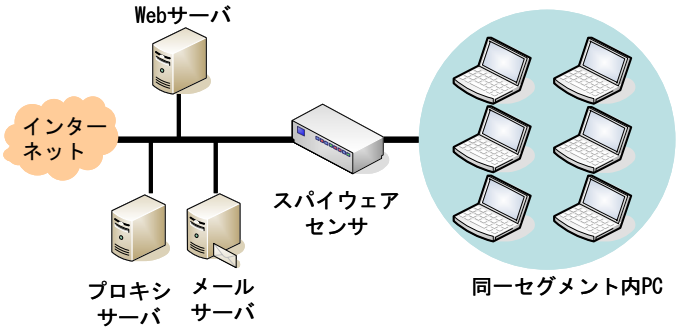

図 2　ホワイトリスト生成環境

表 5　生成されたホワイトリストのエントリ数

アプリケーション	送信先ホスト	設定情報パターン
HTTP	347	197
SMTP	13	36

(2) False Positive 検証実験

生成したホワイトリストおよび図 2の環境を用いて，ユーザが擬似的に通常業務を行う中で検知･遮断の検証を 5 日間行った．ホワイトリストについては，生成時と同様に 1 日毎に更新することとした．

(3) False Negative 検証実験

図 3のように，セグメント内にスパイウェアに感染させた PC を一台だけ設置した．PC 内の情報送信するスパイウェアとして，HTTP で送信するタイプを 9 種類，SMTP で送信するタイプを 7 種類用意した．これらを 1 種類毎に PC にインストールし動作させた．これらは全て既知のものであり，設定される特徴のある情報でシグネチャを作成してしまうと全検知されてしまうため，シグネチャマッチング処理は停止させた．これにより，スパイウェアセンサはスパイウェアを未知のものとして判定することになり，未知のスパイウェアによる検知精度を検証できることになる．なお，ホワイトリストは，「(2)False Positive 検証実験」で最終的に生成されたものを用いた．

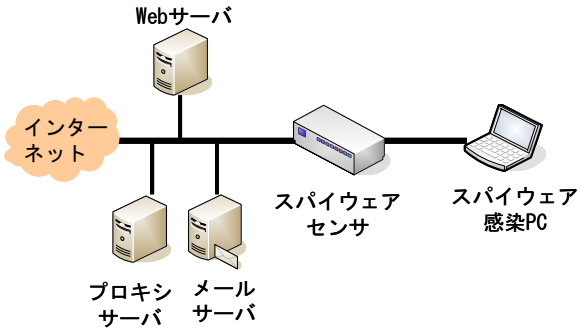

図 3　False Negative 検証環境

4.3 検証結果

(1) False Positive 検証結果

1 日毎の，PC からの HTTP と SMTP の全送信回数，ユーザへの確認回数，スパイウェア検知回数を，表 6と表 7に示す．HTTP の場合，ユーザへの確認回数やスパイウェア検知回数が多いのは，送信先ホストのホワイトリストに登録されていないドメインへアクセスした場合に，1 ページの中で画像などが複数あるとそれらへのアクセスも発生するためその分だけ警告が送信されていることと，ブラウザ側でリクエストタイムアウトが発生し複数回リクエストを繰り返すことが原因とあげられる．ログを見ると，これらを 1 回としてマージすることにより，ユーザへの確認回数は 1/10 程度に減らすことができる（表 6の( )の数字）．

表 6　False Positive 検証結果 (HTTP)

	1 日目	2 日目	3 日目	4 日目	5 日目
HTTP/SMTP 全送信回数	8212	9101	7248	5609	8899
ユーザへの確認回数	438 (55)	572 (66)	152 (11)	377 (42)	123 (17)
スパイウェア検知回数 (False Positive)	22 (3)	8 (1)	0 (0)	10 (2)	10 (2)

表 7　False Negative 検証結果 (SMTP)

	1 日目	2 日目	3 日目	4 日目	5 日目
HTTP/SMTP 全送信回数	23	46	55	22	65
ユーザへの確認回数	1	2	2	1	3
スパイウェア検知回数 (False Positive)	0	0	0	0	0

HTTP で発生したスパイウェア検知（誤検知）については，1 日目と 2 日目で検知したものは，ユーザがブラウザでアクセスしたときに送信したリクエストがホワイトリストにない設定情報パターンであったためであり，4 日目と 5 日

目で検知したものは, ウイルス対策ソフトの定義ファイルアップデートのリクエストであった. ウイルス対策ソフトもブラウザと同様に 1 回のリクエストで複数回リプライを行っていた. SMTP では誤検知は発生しなかった.

(2) False Negative 検証結果

今回検証したスパイウェアについて全て“スパイウェア”として検知することができた. 検知フェーズの内訳としては,「設定情報パターンマッチング」で 14 種類,「ユーザへの確認応答」で 2 種類であった. この 2 種類のスパイウェアは, InterneExplorer にプラグインとして組み込まれるものであったため, InternetExplorer からの設定情報パターンと同様のものになった.

4.4 考察

今回の検証によって, PC 内の情報を送信するスパイウェアをネットワーク上で検知することが可能であることを示すことができた. 特に, シグネチャに依らず今回検証したスパイウェアを検知漏れなく全てを検知できたため, 未知のスパイウェアへ対応できることを確認できた. また, 発生した誤検知（False Positive）については, 初めに作成したホワイトリストの質が原因であったと考えられる. 今回, ユーザが使用したブラウザの種類も複数あり, さらには同じブラウザからでもプロキシを経由する場合と直接 Web サーバにアクセス場合とで設定情報パターンが異なる. 作成期間が 5 日間と短かったため, 全ての設定情報パターンをホワイトリストに登録できなかった. さらには, ウイルス定義ファイルアップデートのような特別なアクセスが検証期間にのみ発生してしまった. 期作成間を長くするか, 特別なアクセスについてはあらかじめ送信先ホストのドメイン名を固定的にホワイトリストに登録しておくことで質の高いホワイトリストを作成でき誤検知を減らすことができる.

また, 本方式ではシステム側で全てを判別せずに, パーソナルファイアウォールのように, 一部の通信についてはユーザが許可/不許可を行う. パーソナルファイアウォールはプログラム名で判別するのに対し, 本方式では送信先ホストのドメイン名で判別する. ユーザは, 通常, Web ブラウジングやメールを行う場合にそのアクセス/送信先を意識するため, 本方式は判別が容易になると考えられる. さらに, ユーザ自身がホワイトリストに追加できる方式の場合, 誤って追加してしまった場合にはそれがホワイトリストに残り続けてしまうことがある. 本方式では, 誤って追加して通信を許可してもその後のトラフィックが少なければ, ホワイトリスト更新時にリストから除去することができ, 被害を食い止めることができる.

5. まとめ

本稿では, ファイアウォールでは防ぐことの難しい情報漏洩につながるスパイウェアからの通信をネットワーク上で検知・防止する方式を提案した. さらに, プロトタイプを実装し, 本方式の有効性を示した.

今後は, ユーザの規模を増やした場合の検証を行うとともに, 性能面などを評価していく予定である.

参　考　文　献

[1] 与那原亨,大谷尚通, 馬場達也, 稲田勉, "トラフィック解析によるスパイウェア検知の一考察", 第30回コンピュータセキュリティ研究会, 情報処理学会研究報告, Vol.2005, No.70, 2005-CSEC-30, pp.23-29, 2005 年 7 月発行.

[2] 大谷尚通, 与那原亨, 馬場達也, 稲田勉, " HTTP 利用型スパイウェアの検知および遮断方式の検討", 第31回コンピュータセキュリティ研究会, 情報処理学会研究報告, Vol.2005, No.122, 2005-CSEC-31, pp.13-18, 2005 年 12 月発行.

[3] インターフェイスの街角(67) - Web ページの鮮度を視覚化する. Unix Magazine, Vol. 18, No. 8, 2003.