2005－CSEC－31（3）
2005／12／9

HTTP 利用型スパイウェアの検知および遮断方式の検討

大谷 尚通† 与那原 亨† 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: †{ootanihs, yonaharaa, babatt, inadatt}@nttdata.co.jp

あらまし 不正アクセスの営利目的化が進んでおり，今後，企業内の PC を狙ったスパイウェアが出現する危険性がある．また，ポリモフィック／メタモフィック技術によってスパイウェア対策ソフトの検知を回避するなど，スパイウェアの隠蔽技術の高度化が進んでいる．本稿では，企業ネットワークにおける HTTP 利用型スパイウェアの脅威に対し，ネットワーク上において，HTTP メッセージの解析による振る舞い検知方式や，HTTP 制御および URL 誘導による遮断方式を検討し，その実現方法を提案する．

キーワード スパイウェア，振る舞い，情報漏洩，HTTP，Proxy

A Consideration of the Spyware Detection and Prevention System for HTTP Communication

Hisamichi OHTANI† Akira YONAHARA† Tatsuya BABA† and Tsutomu INADA†

†NTT Data Corporation Kayabacho Tower, 1-21-2, Shinkawa, Chuo-ku, Tokyo, 104-0033 Japan
E-mail: †{ootanihs, yonaharaa, babatt, inadatt}@nttdata.co.jp

Abstract Recently, many black hat hackers have been coming to work for the profit-pursuing purpose. Therefore, we're afraid of appearance of the new spywares that take aim at the internal PCs of the enterprise network. And, the spywares have been coming to have skills about hiding from the Anti-Spyware software by the polymorphic/metamorphic coding technique. In this paper, we made an examination of behavior detection method by HTTP message analysis and interrupt method of HTTP control and URL modification. And we propose an implementation method of those two methods.

Keyword Spyware, Behavior, Information leak, HTTP, Proxy

1. はじめに

不正アクセスやコンピュータウィルス／ワーム等の動向が，大きく変化した．不正アクセスやコンピュータウィルスは，一部の“ハッカー”と呼ばれる人々が技術を競ったり，いたずらや興味本位といった自己満足のために行うものであった．しかし，最近では，これらの不正行為が“愉快犯”的なものから，“営利目的”としたものへと大きく変化した．なかでも，氏名，住所，電話番号，メールアドレスなどの個人を特定するための情報や，クレジットカード番号／有効期限，銀行口座番号／パスワードなどの ID 情報の不正取得を狙ったスパイウェアの増加が問題となってきている．スパイウェアは，一般的に家庭用 PC に侵入し，個人の ID 情報の搾取を狙っている．しかし，特定の組織を狙ったフィッシング手法の一種“Spear-Phishing”[1]が現れたことからも分かるように，今後，企業内の PC を狙ったスパイウェアが増加することが予想される．

本稿では，イントラネットにおけるスパイウェアの脅威に注目し，企業内におけるスパイウェアの検知・遮断方法を検討し，その対策方式を提案する．

2. スパイウェア対策の現状

2.1 企業内の対策

国内の企業，自治体等の組織におけるクライアント用のウィルス対策ソフトの導入率は,92.4%と高い[2]．よって，企業における当面のスパイウェア対策は，各ウィルス対策ソフトに実装されたスパイウェア検知機能による方法が主流であると思われる．

2.2 スパイウェア対策ソフトの特徴

スパイウェア対策ソフトは，スパイウェア対策専用のソフトと，上記のようにウィルス対策ソフトや IDS／IPS 等の他のセキュリティソフトにスパイウェア検知機能を搭載したソフトに分けられる．後者のソフトは，スパイウェア検知は付属機能であり，検知可能なスパイウェアの種類が少ない．さらにウィルスやワーム検知に用いるシグネチャマッチング方式をそのまま利用するため，バッファ・オーバーフローなどの特徴的なコードを含まないスパイウェアは，亜種の検知がウィルスやワームよりも，より困難であると予想され

る．

また，スパイウェア対策ソフトは，端末上にて検知・除去を行うホスト型と，透過型ゲートウェイ装置等として動作するネットワーク型の二種類が存在する．現在，ホスト型が主流である．ホスト型は，シグネチャマッチングによるスパイウェアプログラム自体の検知だけでなく，レジストリアクセスや改変，Windows API コールや DLL コールのフック等，システムを詳細に監視し，スパイウェアの侵入や実行を検知できる．しかしホスト型ソフトは，ソフト管理上の課題やスパイウェアの攻撃対象になるなどの問題がある．

3. スパイウェアによる通信の分析

Webroot Software 社の調査[3]による調査結果や，SPYWARE GUIDE[4]のリストなどを参考に，アドウェアを中心として主要なスパイウェア（表 1）を収集した．

実験用ネットワークを構築し，実際にこれらのスパイウェアをインストールして，その通信動作を分析した[5]．

表 1:収集したスパイウェア（一部）

名称	種別	利用プロトコル
Gator	アドウェア	HTTP
look2Me	アドウェア	HTTP
iMesh[Cydoor,eZula]	アドウェア	HTTP
WebSearchToolbar	アドウェア	HTTP
WebHancer	アドウェア	HTTP
FreeScratchAndWnd	アドウェア	HTTP
Perfect Keylogger	商用キーロガー	SMTP,FTP
Active Key Logger	商用キーロガー	SMTP
SpyAnywhere	商用キーロガー	HTTP

3.1 スパイウェアの通信例 －WebHancer－

WebHancer の通信動作を分析した結果から，図 1および以下に示す特徴を捉えた．

- インターネット上に設置されたスパイウェア用のサーバ類と HTTP を用いて通信する．
- OS の起動にあわせて自動的に起動し，現在の状況をスパイウェア用ポータルサーバへ通知する．ただし，スパイウェアが初めて起動した時は，スパイウェア用ポータルサーバへの初期登録を行う．…(1)
- 起動後，スパイウェア用のポータルサーバから，スパイウェア用ポータルサーバと２つの情報収集用サーバのアドレスを取得する…(2)
- ブラウザアクセス発生等のクライアント上のイベントに応じて，情報を取得し，情報収集用サーバ①②へ送信する．情報の送信先(情報収集用サーバ①②)は，ラウンドロビンさせる．…(3)～(7)
- ブラウザのアクセスが発生した場合は，アクセスした URL を情報収集用サーバ①へ送信する．…(4)
- 検索サーバ上でキーワード検索を行った場合は，アクセス URL の送信(5)と，検索キーワードの送信(6)を行う．

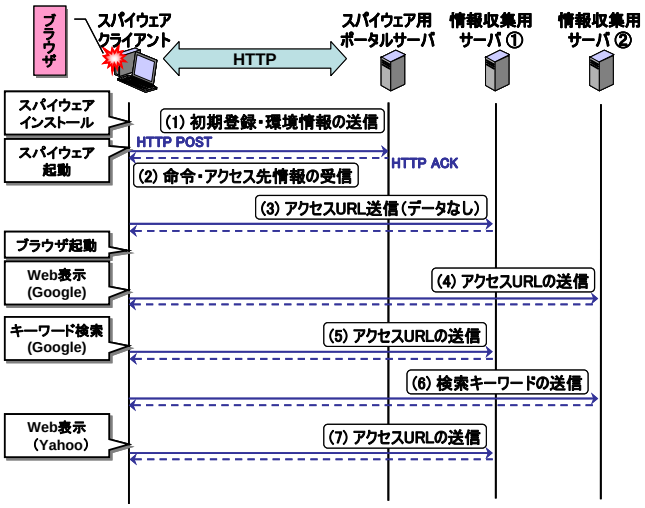

図 1：WebHancer の通信例

3.2 スパイウェア通信の特徴

Perfect Keylogger，Active Key Logger は，キー入力・画面表示を取得して，FTP/SMTP を用いて外部へ一方的に情報を送信するだけの単純なスパイウェアであった．しかし，それ以外の WebHancer 等のスパイウェアは，HTTP を利用し，インターネット上のスパイウェア用のサーバと双方向かつリアルタイムな通信を行っていることが判明した．それらのスパイウェアについて，通信の振る舞いを分析したところ，さらに以下のような特徴も持つことが分かった．

- 定期的にスパイウェア用のポータルサーバから，情報(命令，サーバのアドレス)を取得する．
- スパイウェアの動作は，取得した命令に従う．
- スパイウェア用のサーバ類のアドレスは，URL フィルタ等を回避するために，定期的に変更される．
- スパイウェアの配布用サーバから，最新のパッチやプログラム(スパイウェアだけでなく，さまざまな不正プログラム“マルウェア”)をダウンロードし，自分自身の更新や機能追加を行う．
- 広告配信用サーバから広告を取得し，表示する．
- 振る舞いではないが，スパイウェア毎に HTTP ヘッダ内に記述誤りがある場合や独特な固有名称が含まれる場合などがある．

HTTP を利用したスパイウェアの典型的な通信の振る舞いを図 2に示す．スパイウェア(クライアント)は，

スパイウェア用ポータルサーバの管理下に置かれ，命令やプログラムをダウンロードして，多様な活動を行う．このように最近のスパイウェアは，新しいプログラムをダウンロードして，自身の更新や機能追加を行うトリックラ(Trickler)機能を持つ点が大きな特徴である．これは，ホスト型のスパイウェア対策ソフトによる検知を回避したり，キーロガープログラム等が発見・除去されても，新たに別のプログラムをダウンロードして，スパイウェア活動を再開したりするための仕組みである．

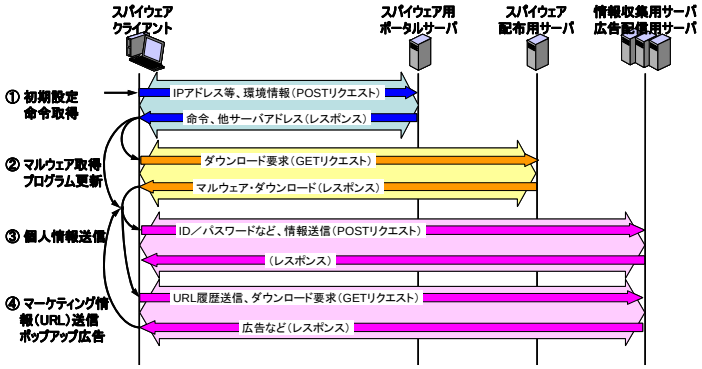

図 2：典型的な通信の振る舞い

スパイウェアが複雑なプログラム構造を持つ理由は，大量感染して被害を発生させることで，その存在を誇示することを目的としたこれまでのウィルスやワームと異なり,マーケティングや ID Theft などの営利目的があるため，長期間に渡って隠れて活動することが求められているからである．なかには，“Retrospy”のようにホスト型のスパイウェア対策ソフトを攻撃して，検知や除去を防止するスパイウェアも出現している．さらに，ポリモフィック/メタモフィック技術を用いて，スパイウェア対策ソフトによる検知を回避したり,ウィルス/ワームのように強力な感染能力を持ったりしたスパイウェアの増加，ボットのスパイウェア化など，新たな脅威の発生が予想される．

4. 対策方法の検討

4.1 対象とするスパイウェアについて

企業内の業務用個人 PC は，各家庭内の個人用 PC に比べて，インターネットから保護されたイントラネット上に接続している．特にインターネットとイントラネットの間には，DMZ が設置され，Firewall や Proxy によるアクセスの制限が行われている．よって，Windows ファイル共有に使われる 139 番ポートや 445 番ポートなど，容易に情報漏洩につながるプロトコルは遮断されている．しかし，3.1の分析結果からスパイウェアは，企業が業務上の必要性からオープンしている HTTP(80/8080/443 ポート)や SMTP(25 ポート)を利用して，通信している．特に HTTP は，普段からイントラネットとインターネットの間でリアルタイムに大量のデータをやり取りしている．このことから，HTTP を利用したスパイウェアの通信は，Firewall による防御(レイヤ 3)や IDS/IPS による検知(レイヤ 5)が困難であり，通常の Web アクセスと見分けがつきにくい．このため，本研究では，イントラネットにおいて最も脅威が大きい HTTP を利用するスパイウェアを対策の対象とする．

4.2 対策の方向性

本研究では，以下の理由から，ネットワーク上におけるスパイウェアの検知・遮断方式を検討する．

- パターンマッチングによる検知が難しいポリモフィック/メタモフィック技術をもつスパイウェアも検知する．
- ホスト型のスパイウェア対策ソフトを攻撃して，検知や除去を回避するスパイウェアによる情報漏洩も検知・遮断する．
- 一般家庭の個人用 PC ではなく，企業のイントラネットからの情報漏洩を対象とし，スパイウェア対策ソフトの導入・管理コストを考慮する．
- 個々の端末上ではなく，ネットワーク上（境界ルータ付近）における効率的な検知・遮断を行う．

4.3 スパイウェア通信(HTTP)の検知方法の検討

これまでの分析結果から，スパイウェアの HTTP 通信には，以下の特徴があることが判明している．

- 情報を外部へ送信するため，POST メソッドを多用する．
- 標準的な HTTP ヘッダの使い方に特徴がある．
- ユーザ定義ヘッダのうち，“X-”で始まる独自ヘッダを多く使用している．
- スパイウェア用のサーバ類からの命令を送信するための HTTP リプライは，HTML を使用しない，または単純な HTML 構造の場合が多い．

そこで，アプリケーションレイヤ(レイヤ 7)において，ユーザがブラウザを用いて Web ページを閲覧する通常の Web アクセスと,スパイウェアによる HTTP 通信との振る舞いの特徴を捉え，これを識別することでスパイウェアの通信を検知する方法を検討する．

4.4 スパイウェア通信(HTTP)の遮断方法の検討

スパイウェアの対策方法は，端末上においてスパイウェアプログラムを除去する方法と，端末上またはネットワーク上において，スパイウェアによる通信を遮断する方法が考えられる．

企業内では，頻繁に PC のリプレースや追加が行われたり，スパイウェア対策ソフトが導入できない端末が存在したりする．各端末上にて対策ソフトを管理するコストも考慮すると，ネットワーク上での対策のほ

うが，効果的である．また，企業におけるスパイウェア対策は，まず情報の漏洩を防ぐことが優先事項である．スパイウェアによって送信された情報が，境界ルータを越えてインターネット上へ漏れる前に阻止しなければならない． 4.1で述べた企業ネットワークの構成を考慮すると，Firewall または Proxy において，スパイウェアによる通信(HTTP)を遮断する方式が合理的である．

5. Proxy を利用した検知・遮断方式

HTTP 利用型スパイウェア対策の検討結果，および多くの企業ネットワークには Proxy サーバが導入されていることから，HTTP 通信をアプリケーションレイヤ(レイヤ 7)レベルにおいて効率よく監視・制御できる Proxy サーバに注目した．企業ネットワークの構成と HTTP 利用型スパイウェアの振る舞いを考慮し，Proxy サーバにおいてスパイウェアによる HTTP 通信を検知・遮断する方式を検討した．

5.1 検知方式

HTTP 通信の内容をチェックする方法として，以下の 3 つを提案する.3 つの方法を利用した検知の全体フローは，図 3に示す．

1. ステータスレベルチェック
   イントラネット内の端末からインターネット上に向けて送出された HTTP リクエスト(以下，Outbound HTTP リクエストとする)の状態遷移を監視する．スパイウェアは POST メソッドを多用することから，GET リクエストを送信せずに POST リクエストを送信する場合などは，スパイウェアによる通信と判断する．
2. ヘッダレベルチェック
   Outbound HTTP リクエストの HTTP ヘッダを監視する．Internet Explorer，Firefox，Opera，Safari 等の主要なブラウザは，類似した HTTP ヘッダを使用する．一方，スパイウェアは，主要なブラウザとは，やや異なる HTTP ヘッダを使用する．主要なブラウザの使用する HTTP ヘッダに比べて違いが大きい場合は，スパイウェアによる通信と判断する．
3. コンテンツレベルチェック
   Outbound HTTP リクエストに対する Web サーバからの HTTP リプライの内容を監視する．ユーザが一般的な Web ページを閲覧する場合は，Web サーバから多くの文字や画像，URL リンクが含まれた HTML データを HTTP リプライとして受信する．一方，スパイウェア用のサーバ類から受信する HTTP リプライには，サーバ類のアドレスや命令コードなどが書かれており，HTML 構造を持たない，または一般的な Web ページとは内容が異なる場合が多い．HTTP リプライの内容に連動した HTTP リクエストも発生しない．HTTP リプライの内容が一般的な Web ページの HTML データらしくない場合は，スパイウェアによる通信と判断する．

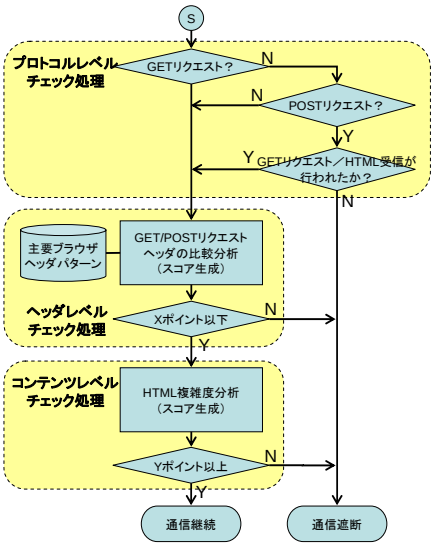

図 3：スパイウェア通信の検知フロー

5.2 ヘッダレベルチェックの検証

ヘッダレベルチェック方式は，スパイウェアによる HTTP 通信の定性的な分析だけでは仮説の検証が難しい．そこで 6 種類のスパイウェア(Gator，iMesh[Cydoor，eZula]，look2Me，WebSearch，WebHancer，FreeScratchAndWnd)および前記の主要ブラウザ 4 種類について，HTTP ヘッダを取得し，机上で事前検証を行った．

図 3にある主要ブラウザ・ヘッダパターンとは，主要ブラウザ 4 種類の HTTP トラフィックを観測し，RFC 2616[7]に基づいて使用されている HTTP ヘッダをチェック表にまとめたものである．

表 2：ブラウザ X の HTTP ヘッダチェック表［GET メソッド用］（例）

フィールド名	ヘッダ予約語	有無	スコア
リクエストヘッダフィールド	Accept	有	10
	Authorization	無	1
	Cookie	有	5
	Host	有	10
	:	:	:
一般ヘッダフィールド	Cache-Control	有	1
	Connection	有	1
	Proxy-Connection	有	1
	:	:	:
エンティティヘッダフィールド	Content-Encoding	無	1
	Content-Length	無	1
	:	:	:
その他	Keep-Alive	有	1
	:	:	:
スパイウェア用ユーザ定義ヘッダ	AgentID	無	10
	AgentSpeed	無	10
	:	:	:

頻繁に使用され，かつ必要性が高いヘッダのスコアは，経験的に大きな値を設定した．ブラウザ毎に GET メソッド用と POST メソッド用の表が必要なため，合計 8 つのチェック表を作成する．

監視中の通信に含まれる HTTP ヘッダと HTTP ヘッダのチェック表(表 2)のヘッダ予約語の有無の排他的論理和(XOR)をとり，真となったヘッダについて，各スコアを合計する．合計スコアの値がある閾値より大きい場合は，スパイウェアによる通信と判断する．試作したチェック表に基づいて，主要ブラウザ(4 種類)とスパイウェア(6 種類)について，スコアを試算した結果を表 3に示す．

表 3：主要ブラウザ/スパイウェア スコア例

対象＼ヘッダパターン			Firefox	IE	Opera	Safari
ブラウザ	Firefox	GET (5)	5.0	8.2	5.6	8.2
		POST(5)	2.6	7.6	5.6	10.6
	IE	GET (5)	7.6	9.6	6.6	7.6
		POST(5)	5.6	6.6	8.6	8.0
	Opera	GET (5)	13.0	10.2	10.0	13.0
		POST(5)	5.4	8.4	5.6	11.4
	Safari	GET (5)	14.0	11.6	11.0	8.0
		POST(5)	9.0	8.0	10.0	1.0
スパイウェア	Gator	GET (5)	35.0	29.0	36.0	33.0
		POST(5)	38.0	39.0	41.0	34.0
	Look2Me	GET (8)	18.0	14.3	17.3	15.3
		POST(2)	28.0	29.0	29.0	22.0
	iMesh	GET (5)	31.6	26.0	33.0	30.0
		POST(5)	36.0	37.0	39.0	32.0
	Web Search	GET (8)	21.6	16.9	21.1	18.6
		POST(2)	40.0	42.0	42.0	35.0
	Web Hancer	GET (0)				
		POST(10)	134.0	135.0	137.0	130.0
	FreeScratchAndWnd	GET (6)	19.8	13.8	22.0	19.0
		POST(4)				

括弧内は，スコア計算に使用した HTTP リクエスト数を表し，スコア値はその平均値を用いた．チェック対象としたスパイウェアの HTTP ヘッダ(GET/POST)は，図 1，図 2に示すような活動中の通信フローから取得した．ただし，WebHancer と FreeScratchAndWnd は，1 種類のリクエストしか行わないため，そのリクエストのスコアのみ作成した．試作した表 3より，以下の検証結果およびチェック表作成に関する課題を得た．

- 主要ブラウザ 4 種類は，試作レベルのチェック表を用いても合計スコアが低く，ブラウザであることが容易に判別可能．
- スコアの閾値を表 3から仮に 15 点以上とすれば，ほとんどのスパイウェア(6 種類)の HTTP 通信を検知できる．
- Opera と Safari は，スコアが 15 点以上の場合が 1 回ずつあった．チェック表はまだ試作段階であり，チューニングが必要である．将来的にはヘッダ出現率[6]の利用を検討したい．
- FreeScratchAndWnd は，BHO(ブラウザヘルパオブジェクト)型であり，IE コンポーネント等も使用して通信すると思われる．よって，IE との区別が難しい．
- Look2Me は単独のプログラムとしてインストールされる場合と，BHO としてインストールされる場合がある．IE コンポーネントも利用すると思われ，その場合は IE との区別が難しい．
- WebSearch(GET リクエスト)の平均スコアは閾値以上だが，8 回の GET リクエストのうち，2～6 番目のリクエストは，スコア閾値を下回っている．

5.3 遮断方法の検証

企業ネットワークにおいて HTTP 利用型のスパイウェア対策を行う場合，図 4に示すような Proxy サーバを利用した方式は，スパイウェアによる HTTP 通信をアプリケーションレイヤ(レイヤ 7)レベルにおいて効率よく監視・制御できる．

HTTP 利用型スパイウェアによる情報漏洩を防ぐ簡単な方法は，疑わしい HTTP リクエストを検知した時に，その通信を遮断することである．しかし，ヘッダレベルチェック方式の事前検証において，僅かに誤検知(False Positive)が確認された．HTTP リクエストを遮断するだけの方法では，ユーザの Web 閲覧が遮断される可用性の問題が残る．

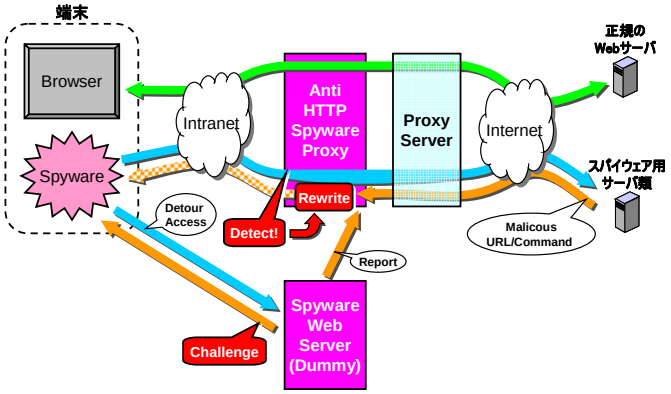

図 4：Proxy サーバを利用した対策方式

本方式は，ユーザのブラウザを用いたアクセスとスパイウェアの HTTP 通信を識別することで，スパイウェアの通信を検知する．そこで，疑わしい HTTP リクエストを検知した場合は，図 4に示すように HTTP リプライの内容に含まれる URL/IP アドレスを作為的に変更して，内部ネットワーク上の偽装サーバへ通信を誘導する．偽装サーバから，ボタンを押す等のユーザ・アクションが必要な HTTP リプライ(HTML)を送信し，ブラウザ/スパイウェア側の反応を監視することで，両者を識

別する．この方式を URL 誘導-遮断方式とする．

ユーザはブラウザ上に現れた「誤検知確認」ボタンに反応するため，検知プログラムは誤検知を把握し，これを回避することができる(図 5の⑥⑦)．しかし，スパイウェアはこれに反応できず，また正しい命令も受信できないため，マルウェアのダウンロードや搾取情報の送信処理等は行われない．

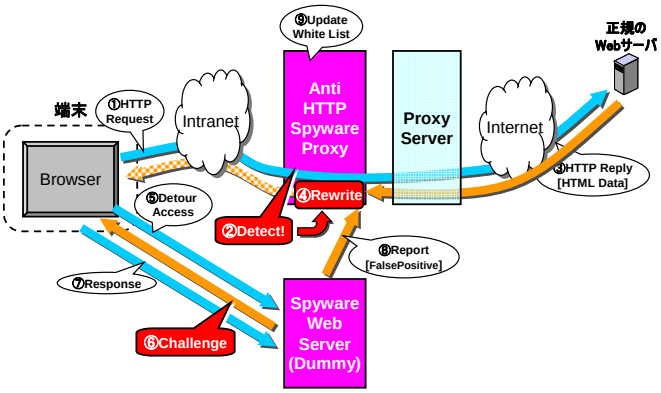

図 5：誤検知時の処理フロー

偽装サーバは，スパイウェア(クライアント)からの送信データも受け取るため,おとりサーバ/フォレンシック装置としても利用できる．ただし，偽装サーバ上に個人情報が蓄積されることが問題となる場合は，HTTP データの“Rewrite”(図 5の④)の代わりに，直接“Challenge”ページ(図 5の⑥)を送信し，ブラウザとスパイウェアの反応の違いから，両者を識別すればよい．この方式を HTTP 制御-遮断方式とする．

5.4 動作検証用プロトタイプの実装と実験結果

簡易的な Proxy プログラムへ，ヘッダレベルチェック方式および HTTP 制御-遮断方式を実装し，ブラウザアクセスによる動作確認を行った．

通常のブラウザアクセスにおいて，画像の読み込みを誤検知(False Positive)する場合などがあり，試作した主要ブラウザのチェック表が不十分であると思われた．HTTP 制御-遮断方式は，正常に動作した．

6. まとめ

本稿では，企業ネットワークにおける HTTP 利用型スパイウェアの脅威に対し，Proxy サーバにおいて HTTP 通信の振る舞いを考慮した検知方式と，アプリケーションレイヤの通信制御による遮断方式を検討し，その実現方法を提案した．

本方式は，端末上においてスパイウェアのプログラム自体を検知し，停止・除去する既存の対策方式とは異なり，ネットワーク上においてスパイウェアによる通信を検知し，情報の漏洩やマルウェアのダウンロードを防止する方法である．

本方式は，以下に示す利点が考えられる．

- 各端末上の対策ソフトの状態に依存せず，インターネット上へ向かうスパイウェアの通信を一括して検知・遮断できる．
- ウィルス・ワームに比べてトラフィック量が少なく，個々の端末上や各セグメント単位よりも，インターネットとの境界ルータ付近における検知・遮断が効率的である．
- 企業内のスパイウェア通信を一括して監視するため，スパイウェアの情報および対処ノウハウが集約できる．
- スパイウェア対策装置の設置数が少なく，導入・管理コストが少ない．

6.1 課題と今後の予定

ヘッダレベルチェック方式のみでは，BHO／IE コンポーネントを用いたスパイウェアの検知が難しい．しかし，よりマクロな視点によるステータスレベルチェック方式や，受信するデータを詳細に検査するコンテンツレベルチェック方式を組み合わせることにより，誤検知(False Negative)を削減できると思われる．

今後，試作した簡易 Proxy プログラムへ，ステータスレベルチェック方式，コンテンツレベルチェック方式による検知方式およびホワイトリスト機能を実装し，実験用ネットワークを用いて，スパイウェアおよびブラウザを用いた実証実験を行う．あわせて，各チェック方式のパラメータチューニングを行ない，実用化を目指した検知率/誤検知率の改善を行う．

文　献

[1] Erik Larkin, “Threat Alert: Spear Phishing”, PC World magazine, November 2005.

[2] 総務省,”情報セキュリティに関する実態調査”, 平成 16 年 7 月, http://www.soumu.go.jp/s-news/2004/040705_2.html

[3] Webroot Software, Inc., “STATE OF SPYWARE Q1 2005”, http://www.webroot.com/

[4] Xblock Systems, LLC. “SPYWARE GUIDE”, http://www.shareedge.com/spywareguide/

[5] 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.

[6] Stephen Thomas, 葛西重夫, “HTTP プロトコル”, ソフトバンクパブリッシング, pp.303-314, 2002.

[7] Network Working Group, “RFC2616: Hypertext Transfer Protocol -- HTTP/1.1”, The Internet Engineering Task Force, http://www.ietf.org/rfc/rfc2616.txt