Vol. 45 No. 8 情報処理学会論文誌 Aug. 2004

メール型未知ウイルス感染防御ネットワークシステムの提案

中谷直司[†] 小池竜一[†]
厚井裕司[†] 吉田等明[††]

近年のネットワーク環境の充実にともない，電子メールを媒介とするコンピュータウイルスが爆発的に増加している．これらのコンピュータウイルスは電子メールを悪用することで短時間で広範囲に被害を与えるとともに，第三者への電子メールのアドレスや内容の流出といったプライバシの侵害が考えられ，何らかの対策は必須ともいえる状況となっている．一般には，コンピュータウイルスを検出および防御するためには，シグネチャと呼ばれるユニークなパターンとのパターンマッチングを行う必要がある．しかし，シグネチャを生成するには人手を必要とするため，未知のコンピュータウイルスに対しては対応が遅れ，被害が拡大する傾向にあった．本論文では，電子メールの添付ファイルにユニークなIDを付加し，それを監視することによって未知ウイルスを迅速に検出し，感染の拡大を防御することができるネットワークシステムを提案する．

The Network System Defended from Infection of Unknown E-mail Viruses

Naoshi Nakaya,[†] Ryuiti Koike,[†] Yuuji Koui[†]
and Hitoaki Yoshida[††]

By the recent enhancement of network, the computer viruses which use e-mail are increasing explosively. These computer viruses do damage broadly even short time by abusing e-mail. There is infringement of privacy like the leakage of the addresses and contents of the e-mail to a third person. Therefore, the countermeasure to computer viruses is indispensable. However, it takes time to generate signatures because of spending human resources. So, unknown computer viruses can infect many computers easily until signatures are generated. In this paper, we propose the network system which stamps attached files with unique ID, detects unknown computer viruses quickly by supervising ID and enables defense of infection and protection of privacy.

1. はじめに

近年のインターネットをはじめとするネットワークの急速な発展にともないコンピュータウイルス（以降，ウイルス）によるコンピュータの不正利用の被害は年々深刻なものとなってきている．情報処理振興事業協会（IPA）が発表している国内のウイルス被害届の統計[1]によると，2001年には24,261件，2002年には20,352件の届け出であった．被害届は減少傾向にあるものの，依然としてネットワーク上にウイルスが蔓延している状況に変わりなく，これらウイルスに対し何らかの対策を行うことはきわめて重要である．

現在のウイルスは電子メール（以降，メール）を媒介として感染する能力を持つため，短時間で感染を拡大させることができる[2),3)]．ウイルスはコンピュータに感染すると，感染したコンピュータ上から無作為に大量のメールアドレスを選び出し，そのメールアドレスに対して自分自身をコピーしたメールを送信する．以前であれば高速な回線を利用できる環境は大学や企業などに限られていたが，現在では一般の家庭であってもADSLなどの高速回線を利用できる環境にあるため感染拡大に拍車がかかっている．

ウイルスにより送信されるメールは受信者の疑いを避けるためさまざまな偽装をする傾向にあるが，これがプライバシ保護の観点から重要な問題になっている．たとえば，既存のメールを引用し返信を装うウイルスが存在するが，このとき引用されたメールを第三者に送信されてしまえばメールの内容が流出してしまうことになる．また，ユーザのコンピュータから適当な文

† 岩手大学工学部
Faculty of Engineering, Iwate University
†† 岩手大学総合情報処理センター
Super Computing and Information Sciences Center, Iwate University

書ファイルなどを添付し自身の存在を紛らわすウイルスにより，極秘ファイルを送付されてしまった例も過去には存在する．あるいは近年よく見られる，送信元を偽るためにランダムに選んだメールアドレスを送信者にするウイルスの場合，他人のメールアドレスを第三者に流出させることになり，自分ばかりではなく他者のプライバシを侵害することにもなりかねない．したがって，ウイルスに感染しないことはもちろん，仮に感染してもメールの送信を阻止することがプライバシ保護においてはきわめて重要となる．

現在のウイルス対策においては，ウイルスの検出はウイルス対策ソフト（以降，アンチウイルス）を利用して行うことが原則となっている．アンチウイルスがウイルス検出する際には，シグネチャと呼ばれるウイルスが持っている固有情報のデータベースを用いることが一般的であるが，この方式には問題点がある．第1に未知のウイルスは検出できない点，第2にシグネチャの生成に時間がかかる点である．ウイルスの検出はシグネチャとのパターンマッチングによって行われる．よって，シグネチャが生成され既知ウイルスとなるまでは検出が行えない．そして，シグネチャを生成するためには，専門知識を持った人間が手動でウイルスを解析し固有な情報を探し出す必要があり，生成に時間がかかる．近年のウイルスはメールを媒介としているために感染が拡散する速度が速く，有効なシグネチャを生成するまでのわずかな時間に被害が広まってしまう恐れがある．

このようなシグネチャの存在しない未知のウイルスに対応することを目指した，ウイルス検出・駆除に関する研究はすでにいくつか存在するが[4)~9)]，シグネチャ方式のようなデファクト・スタンダードが確立されるまでには至っていない．そこで，本論文では未知のメール感染型ウイルスがネットワーク内に侵入してきた場合に，ウイルスに感染しているコンピュータを迅速に検出，隔離しネットワークを防御する方式を提案する．提案するネットワーク（以降，提案ネットワーク）は，メールの添付ファイルに固有なIDを付加すると同時にエンコードを行い，その添付ファイルがいつどこで実行されるのかを正確に監視する機能を中心に構成されている．エンコードとは添付ファイルをコンピュータ上で実行できない形式に変換することであり，これによりセキュリティホールを悪用したウイルスが自動実行されることを防止する．エンコードされた添付ファイルを実行する場合には，デコード機能を持った端末監視ソフトを用いることでそのコンピュータを監視下に置き，ウイルスに感染していないかの調査が行われる．ウイルスは感染時にコンピュータの特定の部分を書き換える，あるいは大量のメールを送信するなどの特定の行動をとる場合が多く，それらを監視することで実行した添付ファイルがウイルスであるか判断が可能となる．判定の結果，実行した添付ファイルがウイルスであった場合には感染しているコンピュータをすぐさま隔離し，また実行された添付ファイルがウイルスであるか詳細に調べる必要がある場合には検証端末に添付ファイルを送り危険度の決定を行うこともできる．加えて，ウイルスであると判断された実行ファイルのヘッダ情報からシグネチャを自動生成し，新たに受信される同じウイルスをサーバ上でフィルタリング可能とする手法も提案する．

以下，2章では最近のウイルスの特徴を，3章ではウイルス対策の現状と課題を，4章ではこれらの課題を解決するためのネットワークの提案を行う．5章では提案ネットワーク上での，実験の各種条件とその結果を示すとともに提案ネットワークの有効性を示す．さらに6章ではウイルスに対するシグネチャの自動生成とフィルタリング手法を提案し，7章を本論文のまとめとする．

2. コンピュータウイルス

経済産業省の定義によるとコンピュータウイルスとは，第三者のプログラムやデータベースに対して意図的に何らかの被害を及ぼすように作られたプログラムで，自己伝染機能，潜伏機能，発病機能のうち1つ以上を有するものとされている．初期のコンピュータウイルスは，これら3つの機能をすべて有し，感染，潜伏，発病というサイクルを繰り返すものが主であった．しかし，最近では潜伏期間を持たずにすぐに活動するウイルスや，発病することなく感染を繰り返すウイルスなどが出現している．そこで，現在では狭義のウイルスの定義では含まれなかったワームやトロイの木馬といったものも含めた，不利益をもたらす不正プログラム全体をウイルスと呼んでいる．また最近のウイルスのほとんどはMicrosoftのWindows系OS（以降，Windows）を対象に作成されており，感染の対象もWindowsに限定されている．よって本論文で扱うウイルスもWindowsを対象としたものとする．以下では，最近のウイルスの主な特徴を述べる．

(1) メールが媒介

メールを媒介として増殖するウイルスをメール感染型ウイルスと呼ぶ．メール感染型ウイルスの多くは，メーラのアドレス帳やwebブラウザのキャッシュなどから取得した任意のアドレスに対して，自身のコ

ピーを添付したメールを許可なく大量に送信する．メール感染型ウイルスの被害は近年急速に増加し続けており，IPA の報告によると感染が報告されたウイルス全体のおよそ94%を占めるまでになっている．

(2) 短い潜伏期間

最近のウイルスは潜伏期間を持たずに実行と同時に活動を始めるものが多い．この種のウイルスは実行されると即座にファイルの改竄やメール送信を行う．

(3) セキュリティホールの悪用

メーラのプレビュー機能のセキュリティホールを悪用するウイルスを，プレビュー感染型ウイルス（以降，プレビュー感染型）と呼ぶ．メール感染型ウイルスの中にはセキュリティホールを悪用することによって，ユーザが明示的に添付ファイルを実行しなくても，メールを読んだだけでコンピュータに感染するものが存在する．

(4) 通常のメールに偽装

ウイルスにより送信されるメールは受信者の疑いを避けるためさまざまな偽装を行い，通常のメールを装う傾向がある．ユーザのコンピュータから適当な文書ファイルなどを添付し自身の存在を紛らわすウイルスや，送信元を偽るためにランダムに選んだメールアドレスを送信者にするウイルスなどである．これらの偽装の結果，個人情報が流出しプライバシを侵害する場合も考えられ，メール型ウイルスの被害の一面をなしている．

本論文で提案するネットワークは，上記の特徴をあわせ持ったウイルスに対して効果を持つものである．現実に感染が報告されているウイルスの大部分は前述の特徴を持ったメール感染型ウイルスであり，これらのウイルスの感染を防ぐことが現在のウイルス対策で最も重要な点である．また，ウイルスメールによる個人情報の流失を阻止し，プライバシ保護することもきわめて重要になっている．

3. ウイルス対策の現状と課題

ウイルスが感染したコンピュータではウイルス対策ソフト（アンチウイルス）を実行することで，ウイルスの検出を行うことができる．ウイルスの検出は多くの場合，シグネチャというウイルスが持っている固有情報のデータベースによって行われるが，このシグネチャは既知のウイルスを解析することで得たものであるため，未知のウイルスには基本的に対処できない．また既知のウイルスであったとしても日々発見されるウイルスすべてに対処するには，頻繁にシグネチャを更新する必要がある．

さらに，シグネチャを生成するには高度な専門知識を持った人間が手動でウイルスの解析をする必要があるため，ウイルスの発見からシグネチャの生成にはどうしてもタイムラグが生じる．よって，日々発見されるウイルスに対してシグネチャ方式だけで対策を行うことは非常に難しくなってきており，仮にシグネチャが毎日更新されているとしても，前述のタイムラグが原因ですべてのウイルスを検出できる保証はない．

また，現在のウイルスの多くは増殖を行う際にメールをはじめとする，すべてのユーザにとって共有のネットワーク資源を利用しているため，被害はウイルスに感染したコンピュータだけにとどまらずネットワーク全体に及んでいる．このような物理的被害のほかに，ウイルスメールによる個人情報の流出にともなうプライバシ侵害の問題も存在する．この問題はウイルスに感染した本人のプライバシだけにとどまるものではなく，メールのアドレスや内容の流出という形で，他者のプライバシをも侵害する可能性を持っており，深刻な問題にもなりかねない．したがって，ウイルスに感染しないための対策は当然のこととし，ウイルス感染時にもネットワーク上にウイルスを送出しないような工夫が必要となる．

4. 提案ネットワーク

前章の課題を克服するために本論文では，プレビュー感染型ウイルスの不用意な実行を阻止し，仮に実行された場合でも被害を最小限にとどめることを目的に，新しいネットワークシステムを提案する．

4.1 構　　成

提案するネットワークは，監視メールサーバ，一般端末，パケット監視装置，検証端末，管理装置から構成されている（図 1）．一般端末とはユーザが日常的

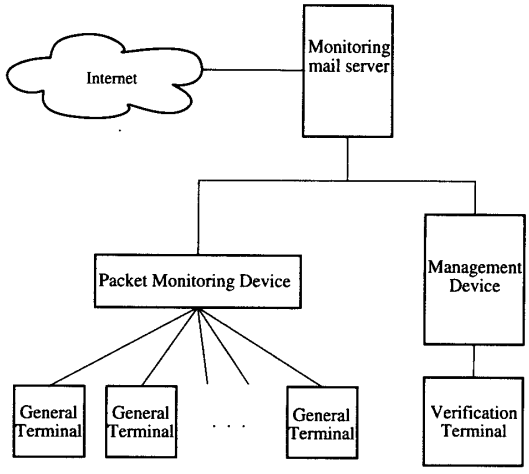


図 1 全体図
Fig. 1 Overall view.

に使用しているコンピュータであり，他の構成要素は今回開発したものである．また前提として，既知のウイルスは提案ネットワークとは別の既存のアンチウイルスにより除去されているものとする．各構成要素の詳細を以下に述べる．

4.2 全体の処理の流れ

シグネチャの存在しない未知ウイルスがインターネットから提案ネットワークに送信され，ユーザが誤って実行してしまった場合の処理の流れを示す．送信されてきたウイルスは 2 章で述べた特徴を持っているものとする．

(1) インターネットからウイルス付きメールが監視メールサーバに到着する．
(2) 監視メールサーバは危険な添付ファイルに対して ID の付加および，エンコード処理を行う．加えて端末監視ソフトの添付を行う．
(3) 危険な添付ファイルを含むメールを管理装置に送信し，管理装置は一時ファイルとして一定期間保存する．
(4) 危険な添付ファイルを含むメールを，監視メールサーバ内の一般端末用のメールスプールに保存する．
(5) 一般端末が監視メールサーバよりメールを受信する．
(6) 受信したメールはエンコードされているため，ユーザは端末監視ソフトのデコード機能を利用して添付ファイルを元に戻し実行する．
(7) デコードされた添付ファイルが実行されると同時に，端末監視ソフトとパケット監視装置の連携によって一般端末が監視状態に置かれる．
(8) 端末監視ソフトは一定時間経過後に ID と監視結果が記述されている結果メールを管理装置に送信する．
(9) 管理装置は監視結果から，実行された添付ファイルがウイルスであるか判定を行う．
  (a) 判定の結果がウイルスであれば，管理装置はパケット監視装置に設定を行い，ウイルスに感染した一般端末を隔離する．
  (b) ウイルスであるか判定ができない場合は，ID をもとに管理装置内に保存されている一時ファイルを取り出す．取り出された一時ファイルを検証端末に送り，詳細な調査を行った結果ウイルスであればウイルスに感染した一般端末を隔離する．

4.3 監視メールサーバ

監視メールサーバは通常のメールサーバとしての機能のほかに，危険な添付ファイルに対し ID の付加およびエンコードを行う装置である．プレビュー感染型ウイルスはメールを媒介としているため，まず監視メールサーバへ送られる．監視メールサーバはメール中からウイルスである可能性のある添付ファイル（以降，危険な添付ファイル）を探し出し，各添付ファイルに固有な ID を付加する処理を行う．同時に端末監視ソフトの添付も行われる（**図 2**）．ID を付加する際には添付ファイルに対するエンコード処理も行われる．エンコード処理とは危険な添付ファイルをコンピュータ上で直接実行できなくする処理であり，これによりプレビュー感染型ウイルスが不用意に実行されてしまうことを防ぐ．これらの処理が施されたメールは監視メールサーバ内のメールスプールに書き込まれる．また ID を付加しエンコードしたメールは，コピーを管理装置に送信し一時ファイルとして一定時間保存を行う．一時ファイルを用意しておくことにより，管理装置は ID から該当する添付ファイルを取り出すことができる．

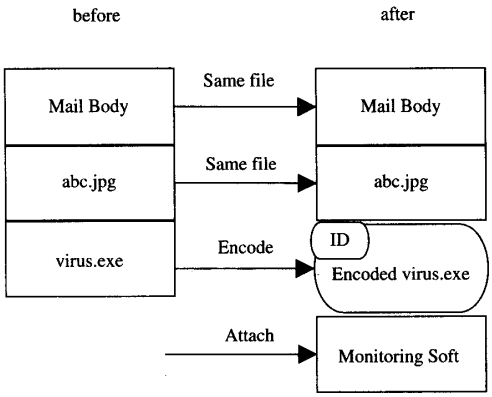


図 2 エンコード前後のメール状態
Fig. 2 State of mail before and after.

4.3.1 危険な添付ファイルの判定機能

監視メールサーバが危険な添付ファイルを選び出す際には，添付ファイルの拡張子を判断基準とする．拡張子とはファイル名の末尾にピリオドで区切って表示した文字列の総称で，Windows においては，OS がファイルの種類を判断するための唯一の指標となっている．ウイルスといえども，悪質な動作をするプログラムにすぎない．そのためウイルスの拡張子は，Windows が動作可能なプログラムと判断するものでなければならず，その種類は限られる．なお，5 章で示す実験において構築したシステムでは，ウイルスの現状を考慮して，**表 1** に示す拡張子を持つ添付ファイルに対し ID の付加およびエンコードを行った．

4.3.2 ID 付加機能

監視メールサーバは危険な添付ファイルに固有な ID を付加する機能を持つ．ID は危険な添付ファイルを

表 1 危険な添付ファイルの拡張子
Table 1 Extensions of dangerous attached file.

拡張子	説明
exe	実行ファイル
com	実行ファイル
bat	MS-DOS バッチファイル
scr	スクリーンセイバ
lnk	ショートカット
pif	MS-DOS 実行ファイルへのショートカット

一意に識別するために付加されるものであり，システム内で重複がないよう日時やランダム文字列を組み合わせることで生成される．ID の領域としては 63 バイトを確保してあるが，実際に使用されている部分は 13 バイトであり，残りの部分は予約領域である．

4.3.3 エンコード機能

監視メールサーバはプレビュー感染型ウイルスの感染を阻止するために添付ファイルのエンコードを行う機能を持つ．エンコード処理とは添付されている実行ファイルを Windows 上で実行できないようなファイルへ変換する処理である．プレビュー感染型のウイルスはメーラに存在するセキュリティホールを悪用することによってメールを見ただけで感染する．しかし，ウイルスの感染に使われるセキュリティホールは，添付されている実行ファイルをユーザの意図しないところで実行させているにすぎない．そこで，添付ファイルにエンコード処理を行うことで Windows 上で実行不可能な状態にする．これによって，未知のセキュリティホールを悪用するようなウイルスが現れても自動実行を阻止することができる．また一般端末で危険な添付ファイルをどうしても実行したいときのことを考え端末監視ソフトの添付も行う．

4.3.4 メールのコピーを管理装置へ保存する機能

監視メールサーバは危険な添付ファイルを含むメールのコピーを一時ファイルとして管理装置に送信する機能を持つ．管理装置は一時ファイルを一定期間保存しておき，ID から該当するメールを取り出すことができる．

4.4 一般端末

一般端末は Windows 系 OS が搭載されたユーザが普段使っている端末のことである．監視メールサーバのメールスプールに書き込まれたメールは一般端末によって受信される．本来であれば，受信されたプレビュー感染型ウイルスによって一般端末はウイルスに感染してしまうはずだが，監視メールサーバ上でエンコード処理が行われているので，一般端末がウイルスに感染することはない．しかし，監視メールサーバによる危険な添付ファイルかどうかの判断は拡張子によるごく単純な方式で行われているにすぎず，エンコード済み添付ファイルが実際にウイルスであるかは分からない．ユーザがエンコード前の実行ファイルを必要であると判断する可能性は十分考えられる．そこでユーザは自らの責任において，監視メールサーバによって添付された端末監視ソフトのデコード機能を用いて，端末監視ソフトの制御下でのみエンコードされた添付ファイルを元に戻し実行することができる．その際，端末監視ソフトはパケット監視装置と連携して，一般端末を一定時間監視下におく．

4.5 端末監視ソフト

端末監視ソフトは，一般端末で実行された危険な拡張子を持つ添付ファイルが，ウイルスかどうか判定するためのデータを収集するソフトウェアである．危険な添付ファイルが実行されると端末を一定時間監視下におき，各種データを収集し結果メールを管理装置に送信する．ちなみに，一定時間とは 10 秒程度の時間である．2 章で述べたように最近のウイルスは潜伏期間を持たず実行と同時に活動を行うものが多い．よって添付ファイルの実行から 10 秒程度を重点的に監視する時間にすれば適切な監視が行える．

4.5.1 デコード機能

端末監視ソフトはエンコードされた危険な添付ファイルを元に戻し実行するデコード機能を持つ．なお，添付ファイルは端末監視ソフトの制御下においてデコード後ただちに実行され，ユーザはデコードされたファイルを得ることはできない．端末監視ソフトが端末監視を行うためのタイミングを得るには，危険な添付ファイルがいつ実行されるのかを正確に知る必要がある．そこで，危険な拡張子を持つ添付ファイルをエンコードしておき，デコード機能を持った端末監視ソフトによって添付ファイルを実行することで監視を始めるタイミングを得る．

4.5.2 端末の状態を取得および比較する機能

端末監視ソフトは危険な添付ファイルを実行した前後の端末の状態を比較する機能を持つ．ファイル実行前後で端末状態に何らかの異常な変化が起こった場合，その危険な添付ファイルはウイルスであるかもしれない．よって端末監視ソフトはエンコードされたファイルを実行する前と，実行した後の状態を取得し，それらを比較することで異常の検出を行う．ただし，状態取得は端末の状態すべてではなく，ウイルスが改竄する可能性が高い，また改竄されると致命的な結果をもたらす可能性がある箇所のみを監視する．監視を行う具体的な場所は以下の部分である．

- 自動実行に関するレジストリ（reg） ウイルスは

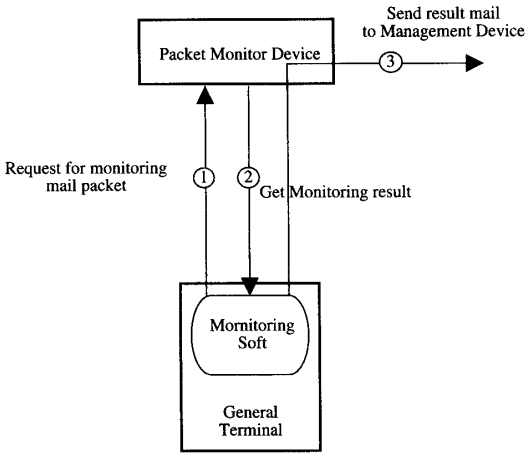


図 3 端末監視ソフトとパケット監視装置の連携
Fig. 3 Collaboration between monitoring soft and packet monitor.

できるだけ長い期間一般端末に感染し続けようとするため再起動時にも自身が自動実行されるようにシステムを書き換えようとする．そこで，Windows の自動実行に関するレジストリを監視対象とする．

- 重要なディレクトリ（dir） ウイルスは Windows の動作の根幹に関わるような重要なディレクトリの書き換えを行おうとするためそれらを監視対象とする．
- 重要なファイル（file） ウイルスは Windows の動作の根幹に関わるような重要なファイルの書き換えを行おうとするためそれらを監視対象とする．通常の実行ファイルがこれらのファイルを書き換えることはまれである．

上記の箇所を監視すれば，ウイルスはいずれかの箇所を改竄するため異常を検出することができる．しかし一方で，いっさいの改竄をせずメモリにのみ常駐するウイルスも考えられる．メール型ウイルスの現状では，こういったメモリ常駐型ウイルスは増殖の点で不利なためほとんど見られないが，今後流行する可能性は否定できない．そこで，添付ファイル実行前後の端末状態を比較するのではなく，実行時のプログラムの動作そのものを監視する方法が考えられるが，この点に関する改善は今後の課題とする．

4.5.3 パケット監視装置との連携機能

端末監視ソフトはパケット監視装置と連携してウイルスによるメール送信の有無を調べる機能を持つ（**図 3**）．端末監視ソフトは，前節の端末の状態を取得および比較する機能によって，一般端末内部の異常を検出することは可能であるが，ウイルスがメールを送信したことを検出するのは難しい．そこで端末監視ソフトは危険な添付ファイルをデコードし，実行する直前にパケット監視装置にメールの発信を監視するように要求を出す．端末監視ソフトは一般端末の状態比較を行った後にパケット監視装置にメール送信の有無を問い合わせる．この機能によって，端末監視ソフトは一定時間内にメールが送信されたか否かを知ることができる．この場合，送信されたメールは必ずしもウイルスによるものとは限らない．しかし端末監視ソフト実行時に，監視中にメール送信をしないようユーザに警告したうえで，本提案システムではこの 10 秒程度の監視時間内に送信されたメールはウイルスによるものと判断する．

```
[base]
ID enc200210291031b4f201
IP 127.0.0.1

[send_mail]
1

[reg]
"HKEY_LOCAL_MACHINE\SOFT\Run" +"autorun"="C:\windows\system\viri.exe"
"HKEY_LOCAL_MACHINE\SOFT\Run" +"backdoor"="C:\windows\system\back.com"
"HKEY_LOCAL_MACHINE\SOFT\Run" -"ie"="C:\windows\ie.exe"

[dir]
"c:\windows\system" +"winsock.ska"
"c:\windows" -"ftp.com"

[file]
"c:\windows\system\winsock.dll"
```

図 4 結果メールの内容
Fig. 4 Content of result mail.

4.5.4 監視結果を管理装置に送信する機能

端末監視ソフトは，収集した情報を管理装置にメールで送信する機能を持つ（図 3）．端末監視ソフトは，一般端末の状態を比較する機能およびパケット監視装置と連携する機能により，危険な添付ファイルを実行し一定時間内に起きた一般端末の異常を管理装置へ結果メールとして送信する．結果メールは検出された異常を規定の書式で記述したものである（**図 4**）．メール中の各項目の意味は，ID は危険な添付ファイルの ID，IP は危険な添付ファイルが実行された端末の IP，send_mail はメール送信の有無，reg は自動実行に関する部分の改竄，dir，file は重要なディレクトリ，ファイルの改竄をそれぞれ表す．

4.6 パケット監視装置

パケット監視装置は一般端末から送信されるパケットの監視およびフィルタリングを行う装置である．

4.6.1 メール送信の監視

パケット監視装置は端末監視ソフトの要求で特定の一般端末からメールが送信されていないか監視する機能を持つ．端末監視ソフトとの連携機能を持っており，要求に従って一般端末から送信されるメールの送信パケットを監視し，メールの送信が行われた場合にはウイルスによるメール送信であると判断し，すぐさまパケットの破棄を行う．そして，それらの処理の結果を端末監視ソフトに返すことができる．

4.6.2 一般端末の隔離機能

パケット監視装置は特定の一般端末を隔離する機能を持つ．4.6.1 項で述べたように監視中にメールの送信があった場合はまずパケットの破棄を行い，その後，一般端末をネットワークから隔離する．隔離は一般端末から送信される宛先ポートが 25 番のパケットを破棄することで行う．2 章で述べたように現在のウイルスのほとんどはメールを媒介として感染する．よってメール送信をさせないことで一般端末を隔離可能となる．また管理装置の要求で特定の一般端末を隔離する機能も持っている．

4.7 管理装置

管理装置は危険な添付ファイルを実行した一般端末の，端末危険度判定を行う装置である．管理装置は端末監視ソフトから受信した結果メールの内容により端末危険度を判定し，危険度が高いものに関してはネットワークから隔離を行うようにパケット監視装置に指示する．危険度が判定できないものは，結果メールに記述してある ID より管理装置内に保存してある一時ファイルを探し出し，それを検証端末に送信することで詳細な調査を行い最終的な端末危険度を確定する．管理装置はこれらの処理の経過や一般端末の端末危険度を画面表示する機能も持ち，管理者はネットワーク内のウイルスに感染した一般端末を即座に確認することができる．

4.7.1 端末危険度判定機能

管理装置は端末監視ソフトから送信された結果メール（図 4）を受信し端末危険度を判定する機能を持つ．端末危険度は危険な添付ファイルを実行したことで，一般端末がどのような危険度にあるか表すもので red, yellow, green の 3 種類がある．以下に，それぞれの詳細について述べる．

- red　実行された危険な添付ファイルが高い確率でウイルスであることを示す危険度である，具体的には，結果メール中の send_mail の項目が 1 であるか，file の項目に内容が存在する場合の危険度である．send_mail の項目が 1 なのは，危険な添付ファイルを実行してから 10 秒以内にメール送信が行われたことを表し，file の項目に内容があるのは 10 秒以内に重要なファイルの書き換えがあったことを表す．添付ファイルで送られてきたファイルを実行したことによって，これらの変化が起こるのは異常なことでありウイルスであると判断する．
- yellow　実行された危険な添付ファイルのウイルスである可能性が判定できない場合の危険度である．具体的には，結果メール中の send_mail の項目が 0 で，かつ reg か dir の項目だけに内容が存在する場合の危険度である．これは危険な添付ファイル実行後にメール送信も，重要なファイルの改竄も行わなかったものの，自動実行に関するレジストリ，および重要なディレクトリに何らかのファイルを追加，あるいは削除を行ったことを表す．この動作は無害な実行プログラムのうちインストーラなどに見られる動作であるため，この動作のみでウイルスであるか判断するのは困難である．そこで管理装置は端末危険度が yellow になってしまった場合，結果メール中の ID を用いて一時ファイルを取り出し検証端末に送信する．検証端末では詳細な調査が行われ最終的な端末危険度が管理装置へ通知される（**図 5**）．
- green　実行された危険な添付ファイルがウイルスではないことを示す危険度である．具体的には，結果メール中の send_mail の項目が 0 で，かつ他の項目には何も書かれていないときの危険度である．これは危険な添付ファイル実行前後で，メールの送信およびファイルの改竄がいっさいなかったことを表す，よって実行された危険な添付ファイルは無害な実行プログラムであったと判断する．

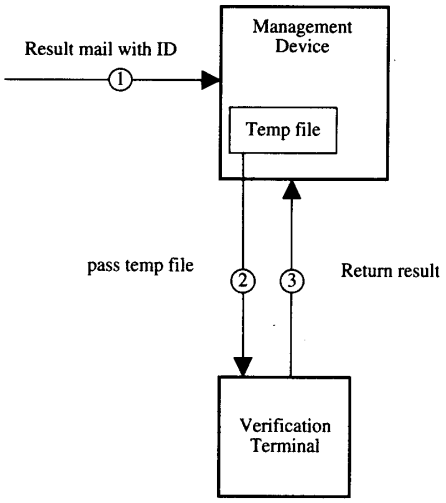


図 5　管理装置と検証端末の通信
Fig. 5　Communication between management device and verification terminal.

4.7.2 管理画面表示機能

管理装置はネットワーク内の一般端末の状態を管理者に分かりやすい形式で表示する機能を持っている．一般端末上で実行された危険な添付ファイルの情報は，端末監視ソフトによってリアルタイムに管理装置に送信される．管理装置は送られてくる結果メールから端末危険度を判定し，それらを画面表示する．この機能によって管理者は管理装置の画面を見ることでネットワーク内のどの一般端末がウイルスに感染しているのかを即座に知ることができる．

4.8 検証端末

検証端末はウイルスの活動が検出しやすい環境に整えられた，端末危険度の判定を専門に行う装置である．検証端末上では仮想の端末が動作しており，その上でWindows系OSが動作している．検証端末上のWindowsには一般端末とは違いユーザが存在せず，検証端末監視ソフトが自動で危険な添付ファイルを実行し異常を検出する．検証端末上のWindowsではユーザが存在する一般端末とは異なり，パケットが送信されるような行動や，レジストリ，ファイルやディレクトリに書き込みを行うような行動はいっさい行われておらず，検証時間も数分とることで危険な添付ファイルを実行した際の影響を正確に調べることができる．検証端末はVMware[10]，検証端末監視ソフト，パケット監視ソフトで構成されている（**図6**）．

VMwareとはPC/AT互換機のハードウェア自体をエミュレーションするソフトウェアでVMware上でWindowsを動作させることが可能である．環境をリセットする機能を持っているため，ウイルスに感染した場合にも速やかに復旧させることができる．またウイルスが容易に感染するよう，検証端末上のWindowsはセキュリティホールの修正がなされていない脆弱なバージョンを用いる．

検証端末監視ソフトは端末監視ソフトを改良したものである．基本的な動作は端末監視ソフトと変わらず新たに以下の部分を監視箇所としている．

- 常駐しているアンチウイルスを停止させるような行動（process） ウイルスにとって端末に導入されているアンチウイルスは邪魔な存在である．そこでアンチウイルスを停止させようとする．
- すべてのパケット送信（packet） 最近のウイルスのほとんどはネットワークを何らかの方法で利用する．検証端末上のWindowsはパケットを送信するような行動はとっていないので，パケットを送信すること自体が異常である．送信の検出はパケット監視ソフトと連携することで行う．
- listenポート（listen） ウイルスの中には感染した端末を後で利用するためにバックドアを仕掛けるものがいる．そのようなウイルスに感染した場合は，外部からの接続を受け付けるためにlistenポートが作られる．
- ファイル共有ソフト関連（file_share_soft） ウイルスの新たな感染経路としてファイル共有ソフトがある．それらのソフトが用いるフォルダも監視対象とする．

検証端末監視ソフトは上記の監視箇所と4.5.2項の監視箇所を合わせて，端末危険度の判定を行う．具体的には検証端末上で危険な添付ファイルが実行された結果，dirの項目以外に内容があった場合はred，それ以外の場合はgreenとする．このとき管理装置が端末危険度として返す値は，red，greenのみである．

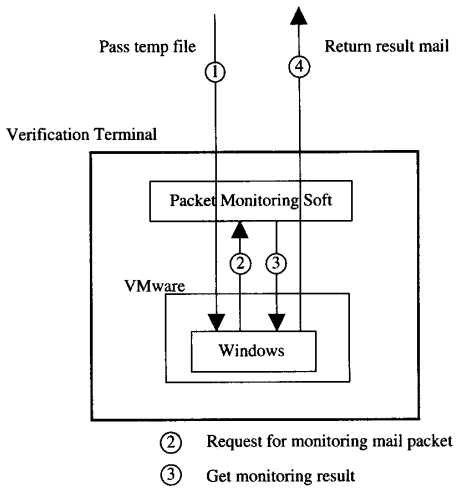


図6 検証端末の構成
Fig. 6 Structure of verification terminal.

5. 実験

5.1 実験内容

提案ネットワークの有効性を示すために実験を行った．実験では実際のウイルスやプログラムを監視メールサーバに送信し，一般端末がそれを受信および実行した場合に管理装置によって端末危険度がどのように判定されるかをみる．また，端末危険度がredになった端末についてはメール送信ができないことを確認し，プレビュー感染型ウイルスが一般端末上で自動実行されないことも確認する．

今回は検証端末のほかにもう1つVMwareを用意しWindows98のインストールを行い，これを一般端末と見なす．また，検証端末上のOSもWindows98とした．以下のファイルを監視メールサーバに送信し，一般端末で受信および実行を行う．

- WORM_KLEZ.Hウイルス 2001年に発見されたプレビュー感染型ウイルスで，現在でも最も被害届が多いウイルスである．
- WORM_SOBIG.Fウイルス 今年発見されたメール感染型ウイルスである．
- unlha.dllのインストーラ 国内でよく使われる圧縮形式であるLHAを扱うのに必要なdllであり，インストーラとともに配布されている[11]．
- ソリティア Windowsに標準で導入されているゲームである．

表 2 実験結果
Table 2 Experimental result.

実行対象	send_mail	reg	dir	file	判定	
					一般	検証
WORM_KLEZ.H	○	○	○	×	red	-
WORM_SOBIG.F	×	○	○	×	yellow	red
unlha.dll	×	×	○	×	yellow	green
ソリティア	×	×	×	×	green	-

5.2 実験結果

結果は以下のようになった (**表 2**). 表中の send_mail, reg, dir, file はそれぞれ, 4.5 節で述べたメール送信と監視箇所の改竄の有無である. 判定 (一般), 判定 (検証) は端末監視ソフトと検証端末監視ソフトが各端末上でどのような危険度判定を行ったかを示す. 結果からも分かるように, 提案ネットワークはウイルスと無害なプログラムとを正しく判定している.

WORM_KLEZ.H ウイルスはプレビュー感染型であるため, 通常なら一般端末で受信を行い見ただけで感染してしまうはずだが, 今回はエンコードされているため自動実行されることはない. しかし一般端末でデコードされ実行されると, すぐさまメール送信を行い, 自動実行に関するレジストリと重要なディレクトリの改竄を行ったため, red と判定された.

WORM_SOBIG.F ウイルスは一定時間内にメール送信を行わなかったものの, 自動実行に関するレジストリと重要なディレクトリの書き換えを行ったため, 一般端末では yellow と判定された. その後, 検証端末で再度調査が行われた際にも, レジストリとディレクトリの書き換えを行ったため最終的に red と判定された. また, 上記の 2 つの例のように管理装置によって red だと判定された端末は隔離が行われ, WORM_KLEZ.H, WORM_SOBIG.F ウイルスが送信されることはない.

unlha.dll のインストーラは無害なプログラムであるが, 重要なディレクトリに書き込みを行うため一般端末で yellow と誤判定され, 検証端末で再度調査された結果, green であると正しく判定された. 一般にインストーラは実行されると, ユーザに対してインストールをしてもよいか同意を求めてくる. そして, ユーザが同意ボタンを押すことで実際のインストール作業が始まる. 一般端末で unlha.dll のインストーラが実行される場合, ユーザは画面の指示に従い同意ボタンを押し, インストールを始めると考えられる. この際, 実行から同意ボタンを押すまでの時間が 10 秒以内であれば, 端末監視ソフトから管理装置に送られる結果メールには, 重要なディレクトリの書き換え情報が含まれる. よって端末危険度は yellow となってしまう. yellow の場合には添付ファイルを検証端末で実行し, 危険度の再調査を行う. 検証端末で実行された unlha.dll のインストーラは, 一般端末のときとは違い同意ボタンが押されることはない. よってインストール作業は行われず, 最終的には正しい結果 green が得られた.

ソリティアは単純なゲームであり, ファイルおよびディレクトリへの書き込みはいっさい行わず, メール送信も行わないため, green と判定された.

6. フィルタリング

前章までで述べた提案ネットワークを用いれば, 基本的にウイルスの感染を阻止することができ, あるいは感染した場合でもメールの送信を阻止することで, 被害の拡大を防ぎプライバシを保護することが可能となる. しかし, エンコードされているとはいえウイルスを一般端末で受信している以上, ウイルスに感染する可能性は依然残されたままである. また, すでに 2 章で述べたように, 最近のウイルスの多くは一度に大量のメールを送信することで増殖する. したがって, 提案ネットワークにおいて危険度が red になった添付ファイルがウイルスであった場合, 同じウイルスが短期間に多数送られてくる可能性が高い. そこで, 未知ウイルスに対するシグネチャを自動生成し, メールをフィルタリングすることでサーバレベルでのウイルス除去を試みる. すなわち, red と判定された添付ファイルを ID により一時ファイルから抽出し, シグネチャを自動生成することができれば, 監視メールサーバにおいてメールのフィルタリングが可能となり, 一般のユーザがウイルスに感染する可能性をさらに下げることができると考えられる.

一般にシグネチャを生成するためには時間と人手が必要となるが, 提案ネットワークではアンチウイルスにシグネチャのない未知ウイルスだけを対象としているため, アンチウイルスメーカがシグネチャを生成するよりも先に生成し, メールをフィルタリングできなければ意味がない. また逆に提案ネットワークは, ア

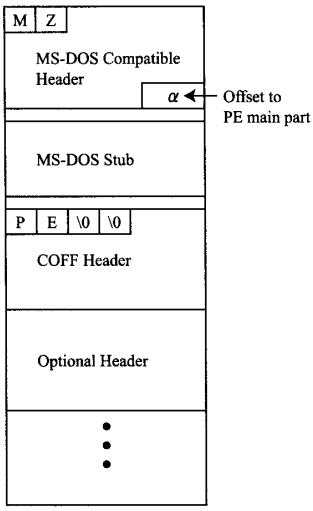


図 7 PE 形式の構造
Fig. 7 Structure of PE format.

ンチウイルスメーカがシグネチャを生成するまでの時間，すなわちウイルスが未知ではなくなるまでの時間さえフィルタリングできれば十分ともいえる．

そこで，提案ネットワークで自動生成するシグネチャは，時間と人手をかけずに生成することを最優先とする方向で考えることとする．

6.1 実行ファイルの固有情報とシグネチャ生成

シグネチャを自動生成する場合，専門知識を持った人間が生成するのとは違い，現在のウイルスの多くが共通して持っており，なおかつ固有な情報を抽出し，それを元にシグネチャを生成する必要がある．そこで，本論文では Windows の実行ファイルのヘッダ情報から，シグネチャを自動生成する手法を提案する．Windows の実行ファイルは，PE（Portable Executable）形式と呼ばれるフォーマット[12] に従って構成されており，ウイルスも Windows という OS 上で実行される以上，このフォーマットに従っている．

図 7 に PE 形式の構造を示すが，このうち，MS-DOS 互換ヘッダ/スタブは下位互換のためのものにすぎず本質的には必要がないため，実行ファイルによっては存在しない可能性が考えられ，シグネチャの生成には適さない．すべての実行ファイルに存在し，なおかつその内容が項目としてつねに同じになるのは，COFF（Common Object File Format）ヘッダとそれに続くオプション・ヘッダの部分となる．この 2 つのヘッダ部分の各項目から，ファイル固有の項目を選びシグネチャを生成すればウイルスをフィルタリングできるものと考えられる．

そこで互いに異なる Windows の実行ファイルを 1,000 個用意し，そこから任意の 2 個を取り出した場合，ヘッダの各項目ごとに値が一致する確率を調査した．ヘッダの項目は 49 項目存在するが，**表 3** に一致する確率が低い，すなわち固有の値に近いものを上位から 10 個示す．この表で最も一致する確率の低い，オプション・ヘッダ中の “Import Table” の値を使ってシグネチャを生成することにする．Import Table までのファイルの先頭からのオフセットは，MS-DOS ヘッダの最後の部分に記述されている，PE 形式であることを表す「PE\0\0」までのオフセット（可変：α byte とする）と，「PE\0\0」の先頭から Import Table までのオフセット 128 byte の和となり，$(128+\alpha)$ byte となる．そして，Import Table のサイズは 8 byte なので，生成されるシグネチャは「ファイルの先頭から $(128+\alpha)$ byte 目に続く 8 byte がウイルスの同部分と同じ」となる．

表 3 ヘッダ項目の一致率
Table 3 Coincidental rate of header item.

項目名	ヘッダ	一致率（%）
Import Table	Optional	0.02
Address Of Entry Point	Optional	0.09
Time Date Stamp	COFF	0.14
Size Of Code	Optional	0.63
Import Address Table	Optional	0.92
Resource Table	Optional	1.09
Size Of Initialized Data	Optional	1.14
Size Of Image	Optional	1.67
Base Of Data	Optional	3.32
Check Sum	Optional	19.60

6.2 base 64 形式に対するシグネチャ生成

前節で述べたように Import Table の値からシグネチャを生成しメールをフィルタリングすることを考えるが，メールは 7 bit のキャラクタコードの送受信を前提としたシステムなので，バイナリである実行ファイルはエンコードされて添付されている．したがって，シグネチャもバイナリに対するものではなく，エンコード後のキャラクタコードに対応したものを用意すれば，フィルタリング時のデコードの手間が省かれ効率が上がるものと思われる．そこで，前節で得たシグネチャのエンコード後のキャラクタコードに対応したものを求める．

メールへバイナリを添付する場合のエンコード形式はいくつか存在するが，実際に用いられているものは base 64 形式[13] がほとんどであり，本論文でもエンコードしたままでのフィルタリングは base 64 形式のみを対象とし，それ以外のあまり利用実績のない形式の添付ファイルは，デコードしたうえでフィルタリングするものとする．base 64 形式は 3 byte のバイナリを 4 character に変換するものであるため，比較対象であるバイナリの 8 byte は 11 ないしは 12 charac-

ter に変換される．なお，どちらになるかはオフセットの値に依存する．また変換された character にはバイナリにおいて前後の値の一部が含まれることになるが，これらの値もウイルス固有の値であるため，そのままシグネチャの一部として用いることにする．したがって，ファイルの先頭から $(128+\alpha)$ byte 目に続く 8 byte がウイルスの同部分と同じ，というシグネチャは，base 64 形式でエンコードされた状態では，「ファイルの先頭から $([(128+\alpha)/3\times 4])$ character から $([(128+\alpha+7)/3\times 4]+1)$ character までがウイルスの同部分と同じ（[] はガウス記号）」というシグネチャに対応する．

6.3 フィルタリング実験

前節までの手法で自動生成したシグネチャの有効性を確認するため実験を行った．本来提案ネットワークは未知ウイルスを対象とするものだが，実験では入手がほぼ不可能な未知ウイルスの代わりに既知のウイルスを用いた．実際に使用したウイルスとその総数を**表 4** に示すが，これは岩手大学のメールサーバに設置された既存のアンチウイルスにより，実際に検知され除去されたウイルスを利用している．したがって，現実のウイルスの増殖状況をある程度反映しているものと考えられる．なお，ウイルスの命名はアンチウイルスメーカが各々独自に行っているが，本論文ではトレンドマイクロ株式会社のアンチウイルスによる名称表示を用いた．

表 4 実験対象のウイルス
Table 4 Virus for experiment.

ウイルス名	総数
WORM_KLEZ.H	6702
WORM_SOBIG.F	1123
WORM_BUGBEAR.A	277
WORM_YAHA.K	220
WORM_YAHA.G	215
WORM_KLEZ.E	191
WORM_HYBRIS.B	161
WORM_SOBIG.A	152
PE_TECATA.1761-O	126
WORM_SOBIG.B	108
PE_BUGBEAR.B-O	98
WORM_YAHA.P	85

実験は前述の手法により base 64 形式でエンコードされたシグネチャを各ウイルスごとに生成し，同じく base 64 形式でエンコードした表 4 のウイルス，および 6.1 節で用いたものと同じ 1,000 個の実行ファイルに対し，フィルタリングを行った．ただし，WORM_HYBRIS.B に関しては実行ファイルにバリエーションが存在したため，シグネチャが 3 種類生成された．実験結果を**表 5** に示すが，表中の検出率はあるウイルス A から生成したシグネチャで複数ある同じウイルス A を検出した割合，誤検出率（ウイルス）は違うウイルスをウイルス A と誤検出した割合，誤検出率（一般）はウイルスではない一般の実行ファイルをウイルス A と誤検出した割合を示す．

結果から明らかなように，自動生成されたシグネチャによるフィルタリングは自分自身の検出に関してはほぼ成功している．ただし，WORM_HYBRIS.B に関しては前述のようにシグネチャが 3 種類生成されてしまい，その内訳はシグネチャ1 が 93.79%，シグネチャ 2 が 4.35%，シグネチャ3 が 1.86%となった．これは一般のアンチウイルスのシグネチャとしては問題のある結果ではあるが，提案ネットワークではこの 3 種類の WORM_HYBRIS.B を別のウイルスとして扱ったと考えれば，検出率はそれぞれに対して 100%とな

表 5 フィルタリングの実験結果
Table 5 Experimental result of filtering.

ウイルス名	シグネチャ	検出率（%）	誤検出率（%）	
			ウイルス	一般
WORM_KLEZ.H	-	100.00	1.33	0.00
WORM_SOBIG.F	-	100.00	0.00	0.00
WORM_BUGBEAR.A	-	100.00	0.00	0.00
WORM_YAHA.K	-	100.00	0.00	0.00
WORM_YAHA.G	-	100.00	0.00	0.00
WORM_KLEZ.E	-	100.00	0.00	0.00
WORM_HYBRIS.B	1	93.79	0.00	0.00
	2	4.35	0.00	0.00
	3	1.86	0.00	0.00
WORM_SOBIG.A	-	100.00	0.00	0.00
PE_TECATA.1761-O	-	100.00	70.86	0.00
WORM_SOBIG.B	-	100.00	0.00	0.00
PE_BUGBEAR.B-O	-	100.00	0.00	0.00
WORM_YAHA.P	-	100.00	0.00	0.00

り，シグネチャを自動生成しフィルタリングするという目的においては十分に有効であるものと思われる．誤検出率（ウイルス）においてWORM_KLEZ.HとPE_TECATA.1761-Oが0%とならなかったが，これはそれぞれが互いにもう一方を誤検出しているという結果による．これはPE_TECATA.1761-Oと検出されているウイルスが，実際にはWORM_KLEZ.HにPE_TECATA.1761-Oが感染した状態のウイルスであることに起因する．すなわち，ある意味においてはどちらもWORM_KLEZ.Hであると考えられ，一概に誤検出とはいいきれない．最後に，本実験では一般の実行ファイルに対する誤検出率は認められなかったが，シグネチャの生成手法から考えれば確率的には誤検出が生じる可能性が考えられる．しかし実験の結果からその確率は低いものと予想され，また，提案手法のシグネチャによるフィルタリングが，未知ウイルスが既知となるまでのタイムラグを埋めるものであることを考え合わせれば，本手法は十分に有効に機能するものと思われる．さらに，本論文ではシグネチャ生成にImport Tableの値だけを用いたが，他の固有情報を組み合わせてシグネチャを生成することでフィルタリング精度を高めることも可能である．

7. まとめ

本論文ではウイルスの侵入に対して迅速に防御を行うネットワークを提案した．提案ネットワークはメールの添付ファイルに固有なIDを付加しエンコードを行い，監視することによってウイルスに感染したコンピュータの特定，隔離が迅速に可能である点が最大の特徴となっている．これらの特徴により提案ネットワークでは，未知のプレビュー感染型ウイルスが侵入してきた場合であっても不用意な実行を阻止し，仮に実行されたとしてもウイルスに感染したコンピュータを即座に隔離することができる．その過程でウイルスである添付ファイルのIDが管理装置に通知され，管理者は感染したコンピュータを特定することができる．また実行した添付ファイルの危険度が判定できない場合には，検証端末によって詳細な調査を行うことも可能となっている．提案ネットワークの有効性を確認するため実験を行ったところ，以下のことが確認された．

(1) プレビュー感染型ウイルスの感染阻止
(2) ウイルスに感染したコンピュータの特定
(3) ウイルスに感染したコンピュータの隔離

さらに，提案ネットワークによりウイルスである添付ファイルを特定すれば，ファイルのヘッダ情報からシグネチャを自動生成することで，サーバレベルでの未知ウイルスのフィルタリングが可能となる手法を提案し，実験により有効性を確認した．

今後の課題としては，より確実なウイルス行動の監視手法の検討，シグネチャの自動生成とフィルタリング部分の提案ネットワークへの実装，提案ネットワークの詳細な評価，メール感染型以外のウイルスへの提案ネットワークの応用などが考えられる．

参考文献

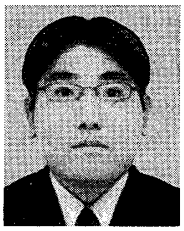
1) 2003年上半期ウイルス発見届出状況，情報処理振興事業協会セキュリティセンター (2003). http://www.ipa.go.jp/security/txt/2003/07-1.html
2) 千石 靖，岡本栄司，服部進実：ワクチンを持たないノードを考慮したネットワーク上におけるコンピュータウイルスの拡散と消滅，情報処理学会論文誌，Vol.39, No.3, pp.818–824 (1998).
3) 中谷直司，厚井裕司，鈴木正幸：追跡能力を有するワクチンを用いたウイルス駆除手法，情報処理学会論文誌，Vol.43, No.8, pp.2467–2477 (2002).
4) 神薗雅紀，白石善明，森井晶克：仮想ネットワークを使った未知ウイルス検知システム，信学技報，No.27, pp.113–120 (2003).
5) Kephart, J.O.: A biologically inspired immune system for computers, *Proc. 14th Int. Conf. on Artificial Intelligence*, pp.20–25 (1995).
6) Forrest, S., D'haeseller, P., Helman, P.: An immunological approach to change detection: Algorithms, analysis and implications, *Proc.IEEE Symposium on Research in Security and Privacy* (1996).
7) Okamoto, T. and Ishida, Y.: A Distributed Approach against Computer Viruses Inspired by the Immune System, *IEICE Trans. Comm.*, Vol.E83-B, No.5, pp.908–915 (2000).
8) Schultz, M.G., Eskin, E., Stolfo, S.J.: Malicious Email Filter – A UNIX Mail Filter that Detects Malicious Windows Executables, *Proc. USENIX Annual Technical Conference – FREENIX Track* (2001).
9) Schultz, M.G., Eskin, E., Zadok, E., and Stolfo, S.J.: Data Mining Methods for Detection of New Malicious Executables, *Proc. IEEE Symposium on Security and Privacy* (2001).
10) VMware.Inc. http://www.vmware.com
11) Unlha.dll. http://www2.nsknet.or.jp/~micco/micindex.html.
12) Microsoft Portable Executable and Common Object File Format Specification (1999). http://www.microsoft.com/whdc/hwdev/download/hardware/pecoff.pdf
13) RFC2045, Multipurpose Internet Mail Exten-

sions (MIME) Part One: Format of Internet Message Bodies (1996).
ftp://ftp.rfc-editor.org/in-notes/rfc2045.txt

(平成 15 年 11 月 28 日受付)
(平成 16 年 6 月 8 日採録)

中谷 直司

1994 年埼玉大学工学部電子工学科卒業．1996 年同大学院博士前期課程修了．1999 年同大学院博士後期課程修了．同年岩手大学工学部情報システム工学科教務職員．2001 年同科助手，現在に至る．進化型アルゴリズム，ネットワークセキュリティに関する研究に従事．博士（学術）．電子情報通信学会会員．

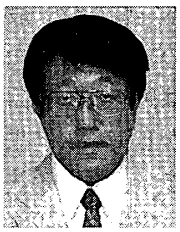


小池 竜一

2003 年岩手大学工学部情報工学科卒業．同年同大学院工学研究科博士前期課程入学，現在に至る．ネットワークセキュリティに関する研究に従事．電子情報通信学会学生会員．

厚井 裕司（正会員）

1970 年東京理科大学理学部応用物理学科卒業．同年三菱電機（株）入社．2001 年岩手大学工学部情報システム工学科教授，現在に至る．主として，マルチメディアネットワーク，ネットワークセキュリティ，RF-ID タグに関する研究に従事．工学博士．IEEE，電子情報通信学会各会員．



吉田 等明（正会員）

1987 年東北大学大学院博士後期課程化学専攻修了．同年筑波大学技官．1989 年同大学化学系助手．1991 年岩手大学工学部助手．1995 年同学部助教授，現在に至る．計算機化学，ニューラルネットワーク，遺伝的アルゴリズム，暗号等に関する研究に従事．理学博士．計測自動制御学会，日本化学会，米国化学会各会員．