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

2006－DPS－126（49）
2006－CSEC－32（49）
2006／3／17

感染プロセスに着目した ワーム拡散防止システムの実装と評価

前田 秀介　馬場 達也　大谷 尚通　角 将高　稲田 勉

(株)NTT データ 技術開発本部
〒104-0033 東京都中央区新川 1-21-2 茅場町タワー
E-mail: {maedasu, babatt, ootanihs, kadom, inadatt}@nttdata.co.jp

概要: 近年, MSBLAST や SASSER のような自己増殖するプログラム “ワーム” による被害が深刻化している. 脆弱性を悪用した感染は, その脆弱性を塞ぐためのパッチやウィルス対策ソフトによって防ぐことができる. しかし, 未知のワームには対応する手段がない. また, 侵入を防ぐためにファイアウォールなどの境界での対策を強化しても, 持込みのノート PC などの端末によって内部感染が拡大してしまうといった事例もある. 本稿では以前に提案した “動的 VLAN 制御” と “ワームの感染プロセスに着目した感染端末検知アルゴリズム” を応用した未知ワームによる被害からセグメント内の全てのクライアント端末を守るシステムを実際の機器を用いて実装し, 評価を行ったので, その結果について報告する.

キーワード: ワーム, 侵入検知システム (IDS), 侵入防止システム (IPS), 振る舞い, ビヘイビア

Implementation of the Worm Prevention System Following the Infection Process and Its Evaluations

Shusuke MAEDA　Tatsuya BABA　Hisamichi OHTANI　Masataka KADO　Tsutomu INADA

Research and Development Headquarters, NTT Data Corporation
Kayaba-Cho Tower, 1-21-2, Shinkawa, Chuo-Ku, Tokyo, 104-0033 Japan

Abstract: The network incidents caused by Internet worms are increasing every year. Infection of worms that exploit the vulnerabilities can be prevented by applying software patches or installing anti-virus software. However, it is impossible to prevent an infection of worms that exploit unknown-vulnerabilities. Although enhancements of security measures at the network boundaries such as firewalls are effective, such enhancements cannot prevent the internal-infection caused by connecting infected terminals to the intranet. In this paper, we implement “worm prevention system following the infection process” that we proposed previously , and evaluate its functional capabilities.

Keyword: Internet Worm, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Behavior

1　はじめに

近年, ネットワークを通じて広域に拡散する自己増殖プログラム “ワーム” が問題となっている. 特に MSBLAST や SASSER のように OS の脆弱性を悪用して増殖するネットワーク型ワームは, メールによって感染するワームと異なり, 端末が起動していれば人手を介さずに感染することが可能であり, 非常に高速に感染が拡大してしまうという問題がある. 2003 年の MSBLAST 出現以降, ワーム対策への関心は高まっており, 様々な対策製品・ソリューションが登場している. しかし, 大きく分けて 2 つの課題が残っている.

1 つめは未知のワームへの対策である. 脆弱性・パッチ情報の公開から, 実環境でワームが出現するまでの期間は徐々に短くなってきており, 次々と出現する新種や亜種に対しては, 最も普及している対策方式であるシグネチャマッチングによる検知方式では対応しきれない. また, ワームに共通した振る舞いをもとに未知のワームを検知する方法も様々考案されているが, 誤検知が多く, 強制的な通信遮断のトリガに用いてしま

うと, 正常な通信まで遮断してしまうといった問題がある.

2 つめは実施形態に関する課題である. 侵入防止システム (IPS = Intrusion Prevention System) などのネットワークベースの対策製品をネットワークにインライン設置する場合, このような装置は未だに高価なため, セグメントの境界部などへの設置が一般的である. しかし, これでは同装置を通過しないセグメント内の端末同士の通信を監視できないため, セグメント内感染を防ぐことが難しい.

著者らは以前に, これらの課題を解決するために, ワームが感染するために必要とするプロセス (挙動とその順序) に着目することでワーム感染端末を誤検知なく検知し, かつ動的 VLAN 制御とエンドスイッチに搭載された SNMP RMON の機能を利用することで早期に感染端末を隔離することでシステム全体の保全を行うシステムを提案した [1]. 本稿では, 提案システムを実装して, その機能を評価した結果について報告する.

2 未知ワーム感染防止システム

著者らは文献 [1] で OS の脆弱性を突いて拡散するワーム (以下, 単にワーム) を未知のものも含めて検知し, セグメント内でのワーム拡散を防止するシステムを提案した. 以下ではその提案内容について述べる.

2.1 ワームの感染プロセスに着目した感染端末検知アルゴリズム

次々と出現するワームに対して, 既知のワームの特徴をもとに検知するシグネチャマッチング方式は, 有効な対策とは言えない. 他方, ワームに共通した特徴的な振る舞いに基づいて未知のワームを検知する方法も様々考案されているが, 誤検知が多いために強制的な通信の遮断やプログラム削除のトリガに用いるには問題がある.

著者らの調査によるとワームが他端末に感染するために次のような手順を踏むことが分かっている.

(1) ポートスキャンを行い感染対象となる端末を探す
(2) 脆弱性攻撃コードを送信し, 感染対象端末を自由に操作するためのバックドアを開く
(3) 感染対象端末のバックドアに対して命令スクリプトを送信する
(4) 攻撃元端末から自己の複製プログラムをダウンロードするように要求させる
(5) 感染対象端末にワームの複製がダウンロードされる
(6) 感染対象端末上で複製プログラムを実行させる

著者らは, ワームが感染を行うためには (1)〜(6) が順番通り行われることの必要性を示し, このうち (1), (4), (5) によって発生するトラフィックに着目し, ワーム (未知のものも含む) 感染端末を検知するアルゴリズムを提案した [3].

アルゴリズム (ワームの感染プロセスに着目した感染端末検知アルゴリズム)

Step 1. (ポートスキャン検知)
セグメント内のある端末 X が t_s 秒以内に同じプロトコル (TCP, UDP, ICMP)・ポートで n_s 個以上の IP アドレスにセッション[1] を張ろうとしたことを検知する.

Step 2. (ダウンロード要求検知)
Step 1 でポートスキャンが検知されてから t_q 秒以内に, 任意の端末 Y が端末 X に対して TCP, UDP のセッションを張ろうとしたことを検知する.

Step 3. (プログラム転送検知)
Step 2 でダウンロード要求が検知されてから t_f 秒以内に, 端末 X から端末 Y に送信された TCP, UDP パケットのデータ長の合計が s_f 以上になったことを検知する.

Step 4. 端末 X をワーム感染端末として検知する.

各ステップで条件を満たさない場合は, 端末 X を非感染端末と判断してアルゴリズムを終了する.

また, 各パラメータの意味は下記のとおりとする.

n_s :ポートスキャンと見なすアクセス IP アドレス数

s_f :ワームプログラムと見なす最小のデータサイズ

t_s :ポートスキャン監視時間

t_q :ダウンロード要求監視時間

t_f :プログラムダウンロード監視時間

このアルゴリズムでは, ワームの感染活動に不可欠な挙動 (ポートスキャンなど) を検知しているため, 未知のワームであっても検知できる. また, 単に個々の挙動を検知するのではなく, 挙動間の順序関係を考慮することで, ある挙動に対する誤検知が増加しても, 感染プロセス全体を見ることで誤検知を削減できる. このアルゴリズムを実装した装置 (以下, ワーム感染端末検知装置) を使用すれば, 未知のワームを検知し, かつ正常端末の誤検知を防ぐことがでる.

2.2 動的 VLAN 制御と SNMP RMON による感染端末の個別監視方式

ワーム感染端末検知装置をセグメント境界部にインライン設置した場合, 装置を通過しないセグメント内の端末同士の通信は監視できないため, セグメントの内側の端末同士の感染を防ぐことができない. この問題を解決するために VLAN を用いる.

セグメント内の端末毎に個別の VLAN を割り当てた場合, セグメント内の端末は他の端末と直接通信を行うことはできないが, 異なる VLAN 間を接続するようなブリッジを設置することで端末同士はこのブリッ

[1] ここでいうセッションとは一般的な TCP のセッションではなく, UDP, ICMP も含めた端末間で特定のプロトコル・ポートを使ってなされる一連の通信を指すものとする.

ジを経由しての通信が可能となる．ワーム感染端末検知装置を VLAN 間を接続するブリッジ上に実装することで全ての端末の通信を監視・制御が可能となる．

しかし, ワーム感染端末検知装置にセグメント内の全ての通信が通過するとなると大きな負荷がかかることが予想される．そこで, 提案システムでは動的 VLAN 制御方式 [4] を応用する．具体的にはワームに感染している兆候のある端末のみをブリッジを経由せずに通信を行う通常の VLAN から, ブリッジ経由で通信を行わせ感染端末の通信を監視するための VLAN に切り替える (以下, 前者を通常 VLAN, 後者を監視 VLAN と呼ぶことにする)．

ワームの感染兆候の検知には多くの管理機能を備えたスイッチでサポートされている SNMP の機能を利用する．ワームは発病すると感染対象を探索するためにポートスキャンを行うことは 2.1 節で述べたが, このとき感染端末側から能動的に大量のパケットが送信される．SNMP RMON のバージョン 1(RMONv1) が実装されているスイッチではスイッチのポート毎に入出力パケットの統計情報を使ってアラートをあげることができる [5]．提案システムではスイッチ自身が RMON を使用して接続されているユーザ端末の送信パケット数の増加を監視・検知して SNMP トラップを使ってアラートを送信するよう設定する．これにより, ネットワークのエンドでのワーム感染の監視を可能とする．

以下にワームが発病した場合の提案システムの動作を示す．

(1) (初期状態)
通常のユーザ端末は全て通常 VLAN に設置し, 各スイッチ間, 及びスイッチとワーム感染端末検知装置は通常 VLAN と監視 VLAN をトランクしておく．この状態ではユーザ端末の通信はワーム感染端末検知装置を介さずに行われる (図 1)．

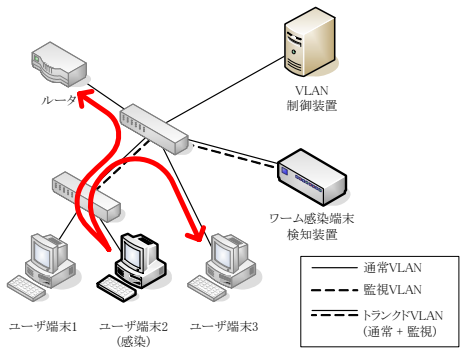

図 1: 初期状態の VLAN 設定

(2) (ポートスキャン兆候の検知)
ポートスキャンによる送信パケットの増加を, 感染端末が接続されたスイッチが SNMP RMON で検知し VLAN 制御装置にトラップを送る．

(3) (動的 VLAN 制御 [4] による監視 VLAN への移動)
VLAN 制御装置はトラップ情報に基づき, SNMP セット要求を使用してスイッチの VLAN の設定を変更する．これにより, 感染端末の通信は全てワーム感染端末検知装置を通過する (図 2)．

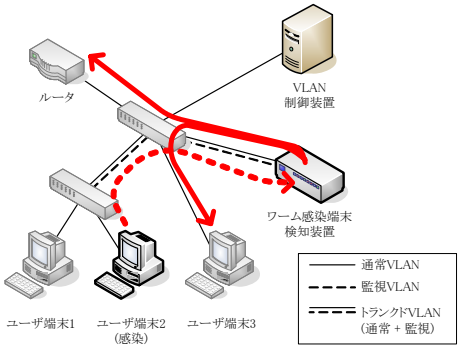

図 2: 感染端末を監視 VLAN に移動した状態

(4) (ワーム感染端末検知装置による感染検知)
ワーム感染端末検知装置は VLAN 間を繋ぐブリッジとして動作し, 通過するパケットを監視する．そして, 2.1 節のアルゴリズムによってワーム感染端末による通信かどうかを判別し, 通信がワーム感染端末のものであると判断した場合は, そのパケットをドロップする．そうでない場合はパケットを VLAN 間ブリッジさせ, 通常 VLAN へ通過させる．

3 提案システムのプロトタイプ実装

VLAN 制御装置とワーム感染端末検知装置をプロトタイプとして実装した．各装置のモジュール構成を図 3 に示す．また, 実装に用いた機器, および, ソフトウェアはそれぞれ表 1, 2 のとおりである．

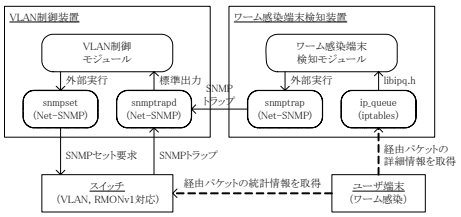

図 3: モジュール間の情報の流れ

VLAN 制御装置　VLAN 制御装置のメインモジュール (VLAN 制御モジュール) は SNMP トラップを受信する SNMP トラップデーモン (snmptrapd) を子プロセスとして起動し, その出力を構文解析することでトラップ情報を取得する．取得したトラップ情報に応じて SNMP セット要求を発行するプログラム (snmpset) を外部実行し, スイッチに VLAN 変更指令を出す．

表 1: 実装に用いた機器

装置名	利用機器
ワーム感染端末検知装置	Dell PowerEdge 850
VLAN 制御装置	Dell OptiPlex GX240

表 2: 実装に用いたソフトウェア

用途	ソフトウェア名	バージョン
OS	Fedora Core	3
カーネル	kernel	2.6.9-1.667
C コンパイラ	gcc	3.4.2
構文解析	bison	1.875c
字句解析	flex	2.5.4
パケット制御	iptables (libipq)	1.2.11
SNMP コマンド	Net-SNMP	5.1.2

ワーム感染端末検知装置　ワーム感染端末検知装置を通過するパケットはパケット制御プログラム iptables により ip_queue モジュールに送られ, その情報を ip_queue の制御ライブラリ libipq の関数を用いて読み出すことでパケット情報を取得する. 必要に応じて SNMP トラップ送信プログラム (snmptrap) を外部実行して VLAN 制御装置にトラップを発行し, VLAN の制御やリンクの ON/OFF を制御する.

4　検知精度の評価

提案方式の有効性を示すために, 開発したプロトタイプを用いて検知精度の評価を行った.

まず, 実際の既知ワームのトラフィックと実際のオフィスの通常トラフィックを用いてそれぞれ検知精度の評価を行い, 検知漏れや誤検知がないかを確認した. さらに, 実際に未知のワームが検知できるかどうかを確認した.

なお, 本方式を考案した 2005 年 2 月当時にその挙動を検証済みだったワームは MSBLAST.C と SASSER.A, SASSER.C のみであるので, これらを既知のワームとみなし, それ以降に出現したワームを未知のワームとみなして評価を行った[2].

ワームトラフィックを使用した評価は図 4 に示すネットワークで実施した. スイッチには Cisco Catalyst 2950 を, ルータには Cisco 2811 を用いた.

4.1　検知漏れ・誤検知の評価

ワームトラフィックと通常トラフィックを用いて, 検知漏れ, および誤検知が発生しないかを確認した.

なお, 以下の評価では各挙動の検知 (ポートスキャン検知など) のみにしか関わらないパラメータについては固定値とした. 値の設定には検知漏れを起こさないような厳しい閾値とし, ポートスキャン検知に関しては 20 秒以内に 3 ホストに同じプロトコルでアクセスする

[2]ワームの名称はトレンドマイクロ社のものに準拠し, 先頭の “WORM_” は省略した.

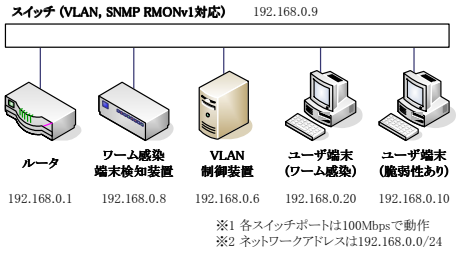

図 4: プロトタイプシステムの構成

といった, 通常の Web ブラウジングでも誤検知が発生するようなパラメータとした ($n_s = 3$[台], $t_s = 20$[秒]). プログラム転送検知では, ワームプログラムとして検知するプログラムのサイズは MSBLAST.C のプログラムサイズ (約 2.6K バイト) の半分以下である 1000 バイトとした ($s_f = 1000$[バイト]).

4.1.1　既知ワームトラフィックを用いた検知漏れの評価

評価方法　隔離されたローカルネットワークでワームを発病させ, ダウンロード要求監視時間 t_q とプログラム転送監視時間 t_f の値を変化させながら, 他の端末への感染前に感染端末を検知, 通信を遮断できるかどうかを確認した. 使用する既知ワームとして MSBLAST.C と SASSER.C を用いた.

結果　MSBLAST.C, SASSER.C それぞれの検知結果を表 3, 4 に示す.

ポートスキャン検知〜ダウンロード要求検知ではポートスキャンが発生してからダウンロード要求が発生するまでに脆弱性の攻撃や命令スクリプトの送信といった挙動が発生するために, ポートスキャンの検知からダウンロード要求が発生するまでに時間がかかり, t_q=1〜2[秒] 程度の値では検知漏れが発生した (表 3, t_q=1〜2[秒]).

また, ダウンロード要求検知〜プログラム転送検知ではダウンロード要求とプログラム転送が同一のセッションの中で行われることもあるため (MSBLAST.C が使用する TFTP では, ダウンロードの要求と実際のダウンロードも同一セッション内で行われる) $t_f = 1$[秒] という低い値であっても検知漏れは発生しなかった. ただし, 今回の評価では単にその場でパケットをドロップするだけの一次対処のみしか実施していないため, t_f の値が小さい場合に TCP の再送要求によってワームのダウンロードが再開され, 検知したにも関わらず他の端末への感染がおきた. (表 4, t_f=1〜30[秒]).

4.1.2　通常トラフィックを用いた誤検知の評価

評価方法　オフィスで業務に使用している 5 台のユーザ端末で本プロトタイプシステムに接続して通常どおりに業務を 5 日間実施してもらい, 誤検知が発生しな

表 3: MSBLAST.C の検知結果

		t_q [秒]					
		1	2	5	10	30	60
t_f [秒]	1	×	○	○	○	○	○
	2	×	○	○	○	○	○
	5	×	○	○	○	○	○
	10	×	×	○	○	○	○
	30	×	○	○	○	○	○
	60	×	○	○	○	○	○

○:検知し遮断, △:検知したが感染, ×:検知できず感染

表 4: SASSER.C の検知結果

		t_q [秒]					
		1	2	5	10	30	60
t_f [秒]	1	△	△	△	△	△	△
	2	△	△	△	△	△	△
	5	△	△	△	△	△	△
	10	△	△	△	△	△	△
	30	△	△	△	△	△	△
	60	○	○	○	○	○	○

○:検知し遮断, △:検知したが感染, ×:検知できず感染

いかを確認した. パラメータについては誤検知を起こしやすくなるよう既知ワームを用いた検証の閾値としてはもっとも厳しい (誤検知を起こしやすい) 値を設定した ($t_q = 60$[秒], $t_f = 60$[秒]).

結果　検証の結果, 通常トラフィックで発生した誤検知は 2 件であった. 誤検知の内容を図 5, 6 に示した.

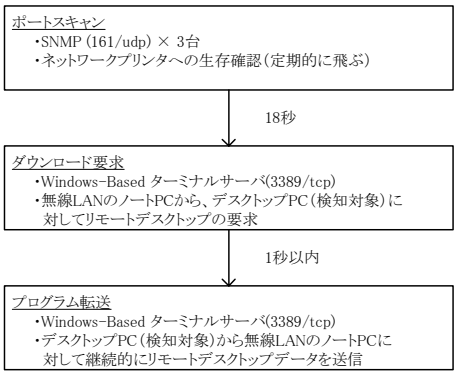

図 5: 通常トラフィックで発生した誤検知 (1)

双方の事例において誤検知の原因となっているのはポートスキャン〜ダウンロード要求の間隔をあらわすパラメータ t_q の設定である. 図 5 の事例では定期的に自動で発生するプリンタの生存確認とユーザのリモートデスクトップの使用が偶然近いタイミングに重なって誤検知された. また, 図 6 の事例も同様に, ポートスキャンの誤検知とダウンロード要求の誤検知は独立した別の事象であり, 今回定めたパラメータ $t_q = 60$[秒] の誤検知の範囲ぎりぎりで発生した現象であった. そのため, t_q としてはこれらの事象がおきても誤検知を起こさない小さい値とする必要がある.

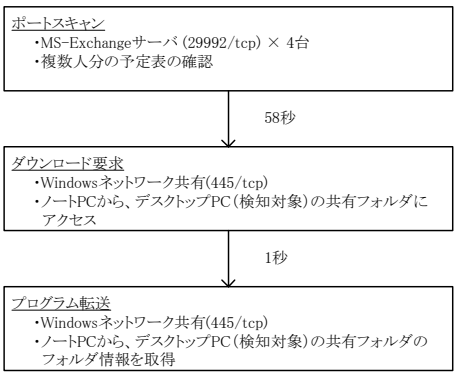

図 6: 通常トラフィックで発生した誤検知 (2)

ダウンロード要求〜プログラム転送の検知については両事例ともユーザの 1 つの操作の中で起こったものであり, 提案方式全体としての誤検知の原因とはいえない.

4.1.3　最適なパラメータの検討

既知ワームの検知結果から判断すると, t_q に関しては 5 秒以上は必要である. ただし, 今回はローカルネットワーク内での評価であり, 感染先が外部ネットワークの端末であった場合には遅延が発生し, あまり低い値を設定すると検知漏れが発生する可能性がある. 遅延を考慮すると 10 秒程度とすべきである. 一方, 通常トラフィックの誤検知の結果において図 5 の誤検知が起こらないようにするには t_q は 18 秒よりも小さく設定しなくてはならない. 総合すると, ポートスキャン検知〜ダウンロード要求検知の間隔は $t_q = 10$[秒] 程度が適当である.

t_f については, 他端末への感染は許してしまっているものの検知漏れが発生したわけではないため, パラメータとしては 1 秒以上あれば問題ない. ただし, 通常トラフィックの検知結果では t_f は直接は誤検知に影響はないため, 60 秒でも問題ないと言える. 検知漏れを起こさせないようにするためには $t_f = 60$[秒] とすべきであると考える.

4.2　未知ワーム検知の評価

評価方法　本システムが未知のワームも含めて検知できることを確認するため, 未知のワームを実際に動作させ, 他の端末への感染を防止できるかを評価した. 評価に用いた未知ワームを表 5 に示す. 実験環境は図 4 と同一とし, 検知パラメータは次のように 4.1 節で検討したパラメータを使用した.

ポートスキャン検知: $n_s = 3$[台], $t_s = 20$[秒].
ダウンロード要求: $t_q = 10$[秒].
プログラム転送: $s_f = 1000$[バイト], $t_s = 60$[秒].

表 5: 評価に用いた未知ワーム

ワーム名	ウィルス情報公開年月*
POEBOT.I	2005/03
SDBOT.CFH	2005/10
RBOT.CST	2006/01

*トレンドマイクロ社による

結果 それぞれのワームを用いて評価を行った結果, 全てのワームを検知することができ, かつ, 他端末への感染も防ぐことができた. 以上のことから, 本提案方式が未知ワームに対しても有効であることが分かった.

また, 使用したパラメータは通常トラフィックで誤検知を起こさないものであり, 提案方式は誤検知と検知漏れの双方を解消できる方式であるといえる.

5 考察

4.1 節の既知ワームの評価で, パラメータ t_q の値を小さくした場合に, ダウンロード要求の監視がタイムアウトしてしまい, 結果としてダウンロード要求を見逃してしまうことが分かった. 以下では, それ以外の検知漏れの可能性について考えていく.

ケース 1 本方式ではパケットの大量送信をポートスキャンの兆候として, 当該端末を監視 VLAN へと隔離し, そこでワーム感染端末検知装置を使って感染プロセスの検知を行う. しかし, 実際にポートスキャンパケットが送信され始めてから VLAN 制御が行われるまでには次のような処理を行われる.

(1) スイッチが SNMP RMON によりパケットの増加を検知して, VLAN 制御装置へトラップを送信
(2) VLAN 制御装置がトラップを受信して, スイッチへ SNMP セット要求を送信
(3) スイッチが受信した SNMP セット要求に基づいて VLAN ID を変更

このため, 実際にポートスキャンが開始されてから, VLAN が変更されるまでに上記の処理に要した時間分のタイムラグが生じる. VLAN 制御が完了してワーム感染端末検知装置が通信を監視できるようになる前に脆弱性のある端末に対して攻撃が発生してしまった場合, 感染を検知できずにワームが他の端末に感染してしまう可能性がある.

ケース 2 ワームが IP アドレスを順に増加させていくシーケンシャルなポートスキャンを行う場合, セグメント内に存在する端末の数が少ないと検知漏れの可能性が高くなる. セグメント内のポートスキャンが始まった場合に, セグメント内の端末数が少ない場合には, アドレス解決のための ARP の数は増加するが, ポートスキャンパケットはあまり送信されないという現象が発生する. すると, 単位時間あたりのポートスキャンパケット数が減少するため, ポートスキャンを検知できない可能性がある. この問題を回避するためには, ワーム感染端末検知装置でのポートスキャン検知に ARP も含めてカウントするといった対策が考えられる.

6 まとめ

本稿では以前に提案した “感染プロセスに着目した未知ワーム感染防止システム” について実装を行い, 実装したプロトタイプを用いて提案方式の評価を行った. 評価の結果, 提案方式が未知のワームの拡散から内外の端末を守り, トレードオフの関係であった誤検知と検知漏れの双方の問題を解決できる方式であることを確認することができた.

今後は感染検知前のワームのすり抜けを防ぐ方式の提案などを行っていく予定である.

参考文献

[1] 前田, 馬場, 大谷, 角, 稲田, “感染プロセスに着目したワーム感染防止システムの実装に関する検討”, 情処研報, CSEC-30, Vol.2005, No.70, pp.7–14, Jul. 2005.

[2] 情報処理推進機構, “コンピュータウィルス・不正アクセスの届出状況（5 月分）について (別紙 1)”, http://www.ipa.go.jp/security/txt/2005/documents/virus-full0506.pdf, Jun. 2005.

[3] 前田, 馬場, 大谷, 角, 稲田, ‘感染プロセスに着目したワーム検知方式の提案”, 情処研報, CSEC-28, Vol.2005, No.33, pp.327–332, Mar. 2005.

[4] 角, 馬場, 稲田, “動的 VLAN 制御によるホスト保護方式の提案”, CSS2004 論文集, Vol.1, pp.49–54, Oct. 2004.

[5] 東角, 鳥居, “SNMP によるワーム検知方式の検討”, CSS2004 論文集, Vol.1, pp.115–120, Oct. 2004.