ネットワークワーム動作検証システムの提案

寺田真敏[†,††] 高田眞吾[††] 土居範久[††,†††]

ワームに実装された感染先探索アルゴリズムにバグがあった場合やワームが実際に使用する乱数生成ルーチンに偏りがある場合には，感染先探索に偏りが出てくる可能性がある．すなわち，感染拡大流布の傾向が変わってきてしまう可能性が潜在的に存在している．このため，実測データに基づく探索特性の検証はコード解析を補完する情報として重要であると考えられるが，公開された実測データはほとんどないのが実情である．また，ネットワークワームの感染拡大を防ぐにあたっては，組織外部の情報に頼りきってしまうのではなく，各組織単独で対策立案のための情報収集手段，たとえば感染動作を検証する実機環境などを保有することは早期対応ならびに情報収集のバックアップという点からも重要である．本論文では，ネットワークワーム流布時の対策立案への利用をふまえ，コード解析の補完と探索範囲に関する動作知見の収集を目的とした「ネットワークワームの探索 IP アドレス」の検証システムと，感染動作にともない使用するポート番号に関する動作知見の収集を目的とした「ネットワークワームの感染動作」の検証システムを提案するとともに，実験を通して提案する検証システムの有効性を示す．

Proposal for the Experimental Environment for Network Worm Infection

Masato Terada,[†,††] Shingo Takada[††] and Norihisa Doi[††,†††]

Code analysis and simulation of network worm infection are useful methods to evaluate how it spreads and its effects. But a bug in infection algorithm or the way of implementing a random number generator etc. affects the retrieval behavior of network worm infection. Then it is important to evaluate the retrieval behavior of network worm infection in experimental environment for complementing the code analysis. And there is no actual, measured data on network worm infection for public use. Moreover; to prevent network worm infection, an organization should not rely solely on the outside information and should have an information gathering method of its own, such as experimental environment to evaluate network worm infection, to plan countermeasures. This paper describes a prototype of experimental environment for network worm infection and actual data about network worm infection. The purpose of experimental environment is to investigate retrieval behavior and infection mechanisms in network worm behavior. For example, there are a mapping of retrieved IP addresses and a ratio of IP addresses retrieved and port numbers used by network worms. Also we implemented a prototype system to show the validity of our approach.

1. はじめに

2004 年 4 月 30 日の Sasser の出現を皮切りに，5 月 1 日には Sasser.B，5 月 2 日には Sasser.C とその亜種が連日にわたり出現した．Sasser は CodeRed, Nimda，Blaster と同様にネットワーク上の感染先を探索し流布する．これらネットワークワームの特徴は，感染のための探索 IP アドレスをランダムに生成し，その IP アドレスの特定ポート番号に対して TCP コネクションを確立する．確立に成功した場合には，特別なメッセージを送付することで脆弱性を攻略し感染を試みる．

ネットワークワームが感染のために選択する探索 IP アドレスの特性は，同一ネットワーク内での感染流布やインターネットからイントラネットへの感染橋渡しの可能性に影響を与える．Sasser の探索 IP アドレスの特性については，いくつかのベンダから提供されており，コード解析[1)] によれば感染先となる探索 IP アドレスの生成は**表 1** のとおりである．コード解析や

† 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi Ltd.
†† 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
††† 中央大学大学院理工学研究科
Graduate School of Science and Engineering, Chuo University

表 1 Sasser 探索 IP アドレスの生成割合
Table 1 Ratio of IP addresses retrieved by Sasser.

感染先となる探索 IP アドレス	割合
上位 2 オクテットが同一 (同. 同. 異. 異)	25%
上位 1 オクテットが同一 (同. 異. 異. 異)	23%
上記以外 (異. 異. 異. 異)	52%

シミュレーション検討[2] により流布の傾向を検討しやすくなってきてはいるが，感染のための探索 IP アドレスの生成アルゴリズムにバグがあった場合や実際に使用する乱数生成ルーチンによっては，その感染特性に偏りが出てくる可能性もある．このため，実測データに基づく探索 IP アドレスの特性検証はコード解析を補完するものとして重要であると考えられてはいるが報告されていないのが実情である．

また，ネットワークワームの感染拡大を防ぐにあたっては，組織外部の情報に頼りきってしまうのではなく，各組織単独で対策立案のための情報収集手段，たとえば感染動作を検証する実機環境などを保有することは早期対応ならびに情報収集のバックアップという点からも重要である．特に，ネットワークワームが発生してから，組織外部の情報が公開されるまでの間の対策立案については，実際にネットワークワームが流布した際の施策として検討の余地がある．

本論文では，ネットワークワーム流布時の対策立案への利用をふまえ，コード解析の補完と探索範囲に関する動作知見の収集を目的とした「ネットワークワームの探索 IP アドレス」の検証システムと，感染動作にともない使用するポート番号に関する動作知見の収集を目的とした「ネットワークワームの感染動作」の検証システムを提案する．提案システムは,「特殊な装置を使用する必要がない」「小規模な機器構成」を前提としたものであり，感染動作の通信履歴を記録するモニタ機能，探索 IP アドレスなどに基づくトラフィック集計機能，通信履歴のフロー分析機能，そして，ネットワークワームがランダムに生成する IP アドレスを感染先として用意したシステムに効率的に振り向ける IP アドレス変換機能から構成する検証システムである．

本論文の構成について述べる．2 章でネットワークワームの現状とその課題について示す．3 章で検証システムについて述べ，4 章で検証システムを用いた実験による効果を示す．5 章でまとめと今後の課題を示す．

2. ネットワークワームの現状とその課題

マルウェアは 1998 年末頃から電子メールを介して流布するようになり，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ている．

提案方式が対象とするワームはネットワークワームであり，代表例として CodeRed I/II，Nimda，Slammer，Blaster，Sasser がある．これらのネットワークワームは，ランダムに選んだ IP アドレスの特定ポート番号に対して TCP あるいは UDP 通信を開始する．送信先との通信に成功した場合には，特別なメッセージを送付することで脆弱性を攻略し感染を試みる．たとえば，Slammer の場合には，ランダムに選んだ IP アドレスのポート番号 1434/udp に Microsoft SQL Server Resolution Service の脆弱性[3] を攻略する UDP パケットを送信して感染を試みる．また，Sasser の場合には，ランダムに選んだ IP アドレスのポート番号 445/tcp に TCP コネクションを確立して，Local Security Authority Subsystem Service（LSASS）の脆弱性[4] を攻略し感染を試みる．

ネットワークワームが流布した際の検知方式としては，シグニチャを用いた検知方式やワームが感染活動により送出するトラフィックパターンを用いて検知する方式[5] などがある．また，感染拡大の防止には，検知に基づきパケットを遮断する方式や遅延を発生させることにより感染速度を抑止するウイルススロットル技術[6] も提案されている．このほかにも，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ともに，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り脆弱性の除去を行うか，サービス自体を無効化するという運用面での対策も浸透し始めている．しかしながら，これら従来技術は，ネットワークワームの感染防止や蔓延防止を目的としたものであり，実測データに基づく探索 IP アドレスの特性検証などのように「ネットワークワームに関する公知となった情報の確度を上げる」「対策情報が公知になっていない時点で感染動作の情報を収集する」という手段を含んではいない．

多くの組織において，ネットワークワームの動作知見に関する情報収集は組織外部に頼っているのが現状である．このため，組織内でネットワークワームが発生した場合にも，組織外部の情報が公開されるまでの間は感染動作が分からず，実施した対策も局所的に有効な対策にとどまってしまう可能性がある．この課題を解決する最も有効な手法は，コード解析によりネッ

トワークワームの動作知見を収集するアプローチであるが，マルウェア自体のコード難読化が進んでいること，各組織に必ずしもコード解析のできる技術者が在籍しているとは限らないという課題がある．さらに，動作知見に関する情報が公知となった以降も，コード解析にともなう結果報告が提供者により異なっている場合や，結果報告そのものが適切ではないことも見受けられることがあり,「公知となった情報の確度を上げる」ことも対策を検討するうえで必要不可欠となっている．

そこで，これら問題を解決する手段として，ネットワークワームの挙動を捕捉する検証システムを用いて実測することにより，動作知見に関する情報を収集する方法を提案する．

電子メールを利用して感染活動を拡大するコンピュータウイルスについては，メールサーバ内の仮想マシン上でウイルスの可能性がある添付ファイルを受信し，その挙動を監視することで未知ウイルス検知を行うシステム[7),8)] など，実機検証と組み合わせた対策システムについて報告されているが，本論文で取り上げるネットワークワームを対象とした検証システムについての報告はない．

3. ネットワークワーム動作検証システム

本章では，動作知見の情報を収集するネットワークワーム動作検証システムとそのプロトタイプシステムについて述べる．

3.1 システム要件

提案する検証システムは,「ネットワークワームの検体が存在するか，あるいは，感染したと思われるシステムが存在すること」を前提としている．さらに，2章をふまえて，提案するシステムに必要とされる要件をまとめる．

要件 1：各組織単独で実現可能な情報収集手段であること．すなわち,「特殊な装置を使用する必要がないこと」「小規模な機器構成であること」.

要件 2：ネットワークワームの感染拡大を防ぐために必要となる情報を収集できること．ただし，本論文では，感染拡大を防ぐために必要となる情報として「感染のひろがりに関わる情報」「感染の通信動作に関わる情報」を対象とし，具体的には下記の 2 つの情報収集を可能とすること．

(a) ネットワークワームが生成する感染先となる探索 IP アドレスを収集できること．

(b) ネットワークワームが感染動作にともない使用するポート番号の情報を収集できること．

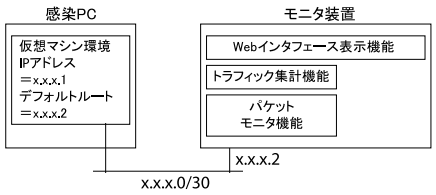


図 1 ネットワークワーム感染先探索特性の検証システム構成
Fig. 1 Verification environment of search characteristic in Network Worm infection.

要件 3：効率的な検証を可能とすること．すなわち，ネットワークワームの特徴でもあるランダムに生成される探索 IP アドレスを被感染システムに収束させ，効率的に感染動作につなげること．

3.2 ネットワークワーム感染先探索特性の検証システム

ネットワークワーム感染先探索特性の検証システムは，ネットワークワームが感染のために選択する探索 IP アドレスの特性，特に，IP アドレスの発生分布に関する情報を収集することを目的とする（要件 2 の (a)）．本検証システムの構成を**図 1** に示す．

感染 PC は仮想マシン環境を備えており，この仮想マシン環境上でネットワークワームの検体を実行する．モニタ装置はパケットモニタ機能で感染動作の通信履歴を記録し，トラフィック集計機能で通信履歴を集計した後，経過時間毎やアドレスブロックごとの探索 IP アドレスの発生分布などを Web インタフェースを用いて表示する．なお，検証システムのプロトタイプ開発にあたっては，パケットモニタ機能としてコマンドライン型 tethereal[9)] を使用し，トラフィック集計機能と Web インタフェース向けの表示整形をスクリプト言語 perl で実装した．

(1) トラフィック集計機能

トラフィック集計機能では,「要件 2 の (a)」を満たすために，パケットモニタ機能の記録した通信履歴から下記 4 項目のデータ抽出を行う．

- 経過時間ごとの探索 IP アドレスの発生分布（**図 2** の左上）
- 上位 1 オクテットが同一 (同. 異. 異. 異) となる探索 IP アドレスの発生分布（図 2 の左下）
- 上位 2 オクテットが同一 (同. 同. 異. 異) となる探索 IP アドレスの発生分布（図 2 の右下）
- 探索 IP アドレスの生成割合（図 2 の右上）

また，パケットの抽出にあたっては，ブロードキャストアドレスを除外するとともに，ネットワークワームが送出する初動パケットが同一である点に着目して，

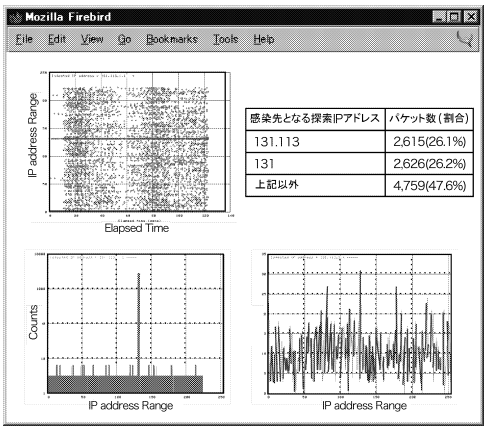


図 2 感染先探索特性の検証システムの Web インタフェース
Fig. 2 Web Interface of search characteristic verification system.

最も発生頻度の高い送信先ポート番号のパケットのみを抽出することで（ICMP の発生頻度が高い場合には ICMP パケットのみを自動抽出する），感染 PC あるいは仮想環境自体が正常稼動のために送出するパケットを除外する．

(2) Web インタフェース表示機能

Web インタフェース表示機能は，項番 (1) の該当項目のグラフ化と図 2 に示す表示形式への整形を行う．

本検証システムの構成上の特徴は下記のとおりである．ただし，感染のために選択された IP アドレスとしてモニタ装置以外の IP アドレスが選択された場合にはパケットを廃棄する．このため，TCP を用いて感染活動を行うワームの場合，感染 PC からの TCP の SYN パケットのみの送信にとどまり，TCP コネクションを確立することはできない．また，感染 PC として感染したと思われるシステムを接続する形態の場合には，ネットワーク環境設定を感染 PC に合わせる必要があるため，必ずしもトラフィック集計に利用可能な通信履歴を収集できない場合がある．

- 2 台の PC で環境を構築できる（要件 1）．
- サブネットワークに属する IP アドレスが感染先として選択された場合にも，トラフィック集計に利用できない通信履歴は最大 3 個に抑えることができる☆．

3.3 ネットワークワーム感染動作の検証システム

ネットワークワーム感染動作の検証システムは，ネットワークワームの感染動作，特に感染動作にともない使用するポート番号の動作知見の収集を目的とする（要件 2 の (b)）．本検証システムの構成を**図 3** に示す．

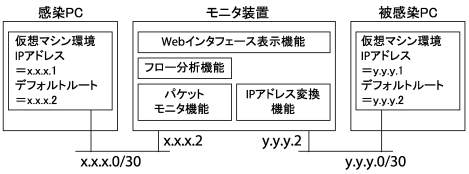


図 3 ネットワークワーム感染動作の検証システム構成
Fig. 3 Verification environment of infection characteristic in Network Worm.

感染 PC ならびに被感染 PC は仮想マシン環境を備えており，これらの仮想マシン環境上でネットワークワームの感染を試行させる．モニタ装置は，パケットモニタ機能で感染動作の通信履歴を記録し，フロー分析機能で通信履歴のフローを分類した後に Web インタフェースを用いて表示する．また，IP アドレス変換機能は，ネットワークワームがランダムに生成する IP アドレスを感染先として用意したシステムに振り向けることにより効率的な検証を実現する．検証システムのプロトタイプ開発にあたっては，パケットモニタ機能としてコマンドライン型 tethereal，IP アドレス変換機能として Linux の iptables を使用し，フロー分析機能と Web インタフェース向けの表示整形をスクリプト言語 perl で実装した．

(1) IP アドレス変換機能

インターネットにおける IP アドレス空間は $2^{32} = 4{,}294{,}967{,}296$ であるから，ネットワークワームが感染のための探索 IP アドレスをランダムに生成する場合，ネットワークワーム自体がスレッド化により探索効率をあげたとしても，被感染 PC の IP アドレスが選択される確率はきわめて低く，結果として短時間での感染動作検証ができないことは容易に類推できる．被感染 PC のネットワークインタフェースに複数 IP アドレスを割り当てることで選択確率を上げる方法もあるが，必ずしも被感染 PC に複数 IP アドレスを割り当てる機能があるとは限らない．そこで，中継装置としても機能するモニタ装置に送信先 IP アドレスを被感染 PC の IP アドレスに変換する IP アドレス変換機能を持たせる．また，本機能の実現にあたっては「要件 1」を満たすために，Linux の iptables が提供する DNAT（Destination Network Address Translation）を使用することとした．

DNAT は「送信先 NAT」とも呼ばれ，送信先 IP アドレスを特定の IP アドレスに変換する機能である．主にファイアウォールの内側にあるネットワークサー

☆ 感染先として，感染 PC の IP アドレス，サブネットワークのネットワークアドレスならびにブロードキャストアドレスが選択された場合には，トラフィック集計に利用可能な通信履歴として収集できない可能性がある．

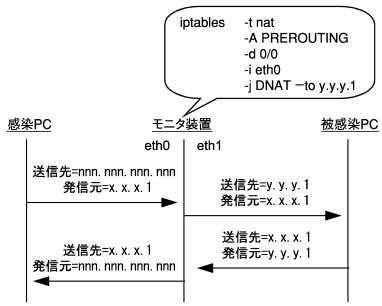


図 4 DNAT を用いた送信先 IP アドレスの変換
Fig. 4 Destination IP address translation by DNAT.

ビスをインターネット側から利用できるようにする機能として提供されており，ファイアウォールローカルサーバ機能あるいは，バーチャルコンピュータ機能とも呼ばれている．プロトタイプシステムでは，**図 4** のように iptables の DNAT を用いた送信先 IP アドレス変換を設定しており，モニタ装置はインタフェース eth0 に届いたすべての感染 PC からのパケットの送信先 IP アドレスを y.y.y.1 に変換した後，被感染 PC に転送する．

(2) フロー分析機能

フロー分析機能では，「要件 2 の (b)」を満たすために，パケットモニタ機能の記録した通信履歴から下記 2 項目のデータ抽出を行う．

- ネットワークワームが感染動作にともない使用する送信先ポート番号
- 送信先ポート番号の発生系列とその頻度

本検証システムにおいて発生するトラフィックはネットワークワーム感染動作によるトラフィックのみと仮定できる．したがって，「初出パケットの抽出」「送信先ポート番号の発生系列の抽出」の 2 つのステップからフロー分析を行うことにより，ネットワークワームが使用する送信先ポート番号の発生系列とその頻度，すなわち，感染動作を概観できることになる．

(a) 初出パケットの抽出

通信履歴から発信元/送信先 IP アドレスのペア，発信元/送信先ポート番号のペアによるグループ化を行った後，グループ内での初出パケットを抽出することで，ネットワークワームが感染動作に使用する送信先ポート番号を特定する．TCP 通信の場合には，通常，SYN フラグの設定されたパケットが抽出対象となる．

(b) 送信先ポート番号の発生系列の抽出

抽出した初出パケットを発信元/送信先 IP アドレスのペアで再グループ化した後，グループ内での送信先ポート番号の発生系列を抽出する．これにより，ある

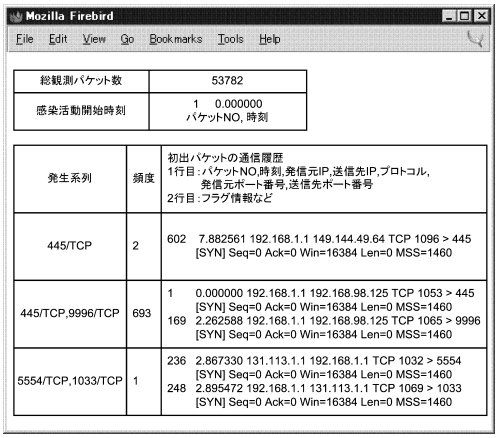


図 5 フロー分析に基づく送信先ポート番号の発生系列とその頻度の例
Fig. 5 The example of the Flow analysis function.

発信元/送信先 IP アドレスのペアにおける送信先ポート番号の発生系列を導き出すことができる．その後，全グループを対象とした送信先ポート番号の発生系列の頻度を算出する．

感染 PC の IP アドレス 131.113.1.1，被感染 PC の IP アドレス 192.168.1.1 として構成した検証システムで確認を行った Sasser.C の送信先ポート番号の発生系列と頻度の結果例を**図 5** に示す．この事例の場合，ポート番号 445/TCP，9996/TCP の順に発生する通信は 693 回であり，初出は観測開始から 1 パケット目，派生した 9996/TCP の通信は 169 パケット目であることと，ポート番号 5554/TCP，1033/TCP の順に発生する通信は 1 回であり，初出は観測開始から 236 パケット目であることを示している．そして，この発生系列と頻度から，Sasser.C の感染動作は，ポート番号 445/TCP，9996/TCP の順に発生する通信と，ポート番号 5554/TCP，1033/TCP の順に発生する通信の 2 段階に分かれると判断できる．なお，最も頻度の高い発生系列で観測された初出パケットの時刻を感染活動開始時刻として表示している．

本検証システムの構成上の特徴は下記のとおりである．

- 3 台の PC で環境を構築できる（要件 1）．
- 感染のためにランダムに生成された探索 IP アドレスを効率的に被感染 PC に振り向けることができる（要件 3）．図 5 の事例では，248 パケット目，約 3 秒で感染動作の最終段階に入っている．なお，IP アドレス変換機能の効果については，4 章の実験においても示す．

4. 検証システムを用いた実験

本章では，3 章で提示した検証システムを用いて確認した既知のネットワークワームの感染先探索特性と感染動作について述べる．

4.1 既知ネットワークワームの感染先探索特性

本節では，既知のネットワークワームに対して，提案する「ネットワークワーム感染先探索特性の検証システム」がコード解析の補完となる感染先探索特性の情報を収集できることを示す．

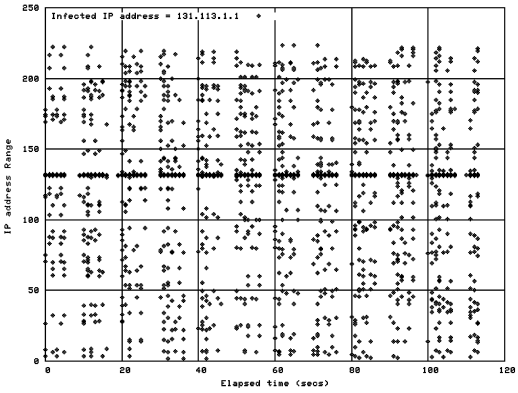
4.1.1 実 験 環 境

実験に使用した感染 PC は，Dell PowerEdge1400 PemtiumIII，メモリ 256 MB，Microsoft Windows 2000 Server Service Pack 4 のマシンである．モニタ装置は，IBM Thinkpad 2609-93J Pentium III，メモリ 192 MB，Redhat Linux 7.3 のマシンであり，100 Mbps の LAN を用いて接続した．また，感染 PC の仮想マシン環境としてメモリゲストサイズ 160 MB，全仮想マシンの総メモリ 176 MB を設定した VMware Workstation 上に日本語版 Windows（修正プログラムとサービスパック適用なし）環境を準備し，Windows 2000 環境で CodeRed3，Nimda.E，Blaster，Slammer，Windows XP Professional 環境で Sasser.B，Sasser.C を確認した☆．

4.1.2 感染先探索特性

(1) CodeRed3

CodeRed3 [10] は 2003 年 3 月に出現したネットワークワームであり，アドレスブロック探索比率を加味してつねに探索 IP アドレスをランダムに選択するタイプ（以降，アドレスブロック探索比率加味型&ランダム探索型を呼ぶ）に属する．ワームのコードそのものはオリジナルの CodeRed II と 2 バイトしか異ならない．この 2 バイトは，CodeRed II に設定されていた稼動期限 2001 年 9 月末が，34952 年 9 月末まで動作するよう変更されたことによる．したがって，CodeRed II と CodeRed3 の感染探索特性は同一であり，本論文で確認した特性は CodeRed II にも当てはまる．経過時間ごとの探索 IP アドレスの分布を**図 6** に示す．横軸は CodeRed3 感染以降の経過時間，縦軸は送出されたパケットの送信先 IP アドレス範囲（0.0.0.0〜255.255.255.255）である．また，CodeRed3 のアドレスブロック探索比率は**表 2** に示すとおりコード解析[10] の結果に沿っているといえる．



観測開始から 10,000 パケットを対象にプロット

図 6 経過時間ごとの探索 IP アドレス（CodeRed3）
Fig. 6 Progression of IP addresses retrieved by CodeRed3.

表 2 アドレスブロック探索比率（CodeRed3）
Table 2 Rate of IP addresses retrieved by CodeRed3.

感染先となる探索 IP アドレス	実験結果	コード解析
上位 2 オクテットが同一	37.7%	37.5%
上位 1 オクテットが同一	50.8%	50.0%
上記以外	11.5%	12.5%

試行 3 回，観測開始から 10,000 パケットを対象とした平均値

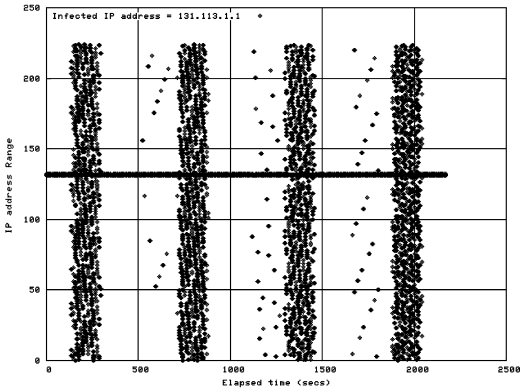
(2) Nimda.E

Nimda.E [11] は 2001 年 10 月に出現したネットワークワームであり，同年 9 月に流布した Nimda の亜種である．Nimda.E も CodeRed3 と同様にアドレスブロック探索比率加味型&ランダム探索型に属する．経過時間ごとの探索 IP アドレスの分布を**図 7** に示す．縦軸は送出されたパケットの送信先 IP アドレス範囲（0.0.0.0〜255.255.255.255）である．Nimda.E の場合には探索動作に周期性が見られ，サンプリングによっては探索比率に偏りをともなってしまう．結果として「上記以外 (異. 異. 異. 異)」の比率がコード解析[12] よりも低い実測値となっていることが分かる（**表 3**）．

(3) Blaster

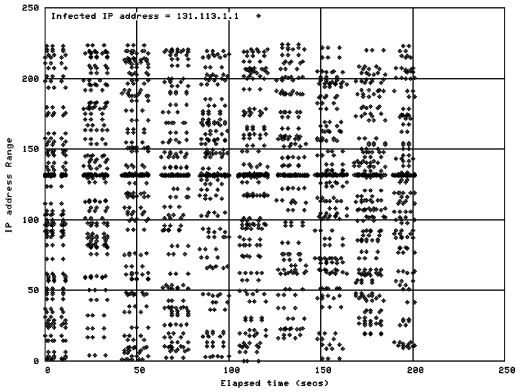
Blaster [13] は，2003 年 8 月に出現したネットワークワームであり，アドレスブロック探索比率を加味して探索開始 IP アドレスを決定した後，順次 IP アドレスを加算する探索方式を使用している．感染先探索特性上，この点が CodeRed，Sasser との大きな違いとなっ

☆ 商品名称などに関する表示
Windows XP, Windows 2000 は Microsoft Corporation の米国およびその他の国における登録商標または商標です．
Linux は Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です．
Pentium は Intel Corporation の登録商標です．
IBM，Thinkpad は International Business Machines Corporation の登録商標です．
Red Hat は米国およびその他の国々における Red Hat, Inc. の登録商標です．
本論文に記載されている会社名，製品名は，各社の登録商標または商標です．



観測開始から 51,261 パケットを対象にプロット

図 7 経過時間ごとの探索 IP アドレス（Nimda.E）
Fig. 7 Progression of IP addresses retrieved by Nimda.E.



観測開始から 3,757 パケットを対象にプロット

図 9 経過時間ごとの探索 IP アドレス（Sasser.B）
Fig. 9 Progression of IP addresses retrieved by Sasser.B.

表 3 アドレスブロック探索比率（Nimda.E）
Table 3 Rate of IP addresses retrieved by Nimda.E.

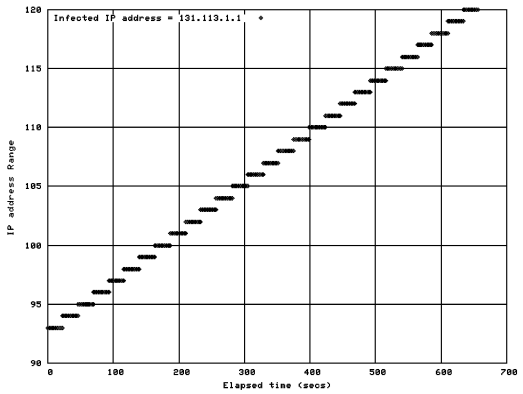
感染先となる探索 IP アドレス	実験結果	コード解析
上位 2 オクテットが同一	50.9%	50%
上位 1 オクテットが同一	38.8%	25%
上記以外	10.3%	25%

試行 9 回，観測開始から 10,000 パケットを対象とした平均値

表 4 アドレスブロック探索比率（Sasser.B）
Table 4 Rate of IP addresses retrieved by Sasser.B.

感染先となる探索 IP アドレス	実験結果	コード解析
上位 2 オクテットが同一	27.2%	25%
上位 1 オクテットが同一	24.6%	23%
上記以外	48.2%	52%

試行 5 回，観測開始から 3,000 パケットを対象とした平均値



観測開始から 7,500 パケットを対象にプロット

図 8 経過時間ごとの探索 IP アドレス（Blaster）
Fig. 8 Progression of IP addresses retrieved by Blaster.

ている．探索範囲に絞った経過時間ごとの探索 IP アドレスの分布を**図 8** に示す．縦軸は送出されたパケットの送信先 IP アドレス範囲 153.75.20〜153.75.50 を示している．図 8 の場合，探索開始 IP アドレスとして 153.75.20.1 が選択されており，以降 4 オクテット目が 1 つカウントアップしながら感染先を探索していることを示している．なお，探索範囲を絞ったグラフ作成については，実験ごとにプログラムのカスタマイズで対処した．

(4) Sasser.B

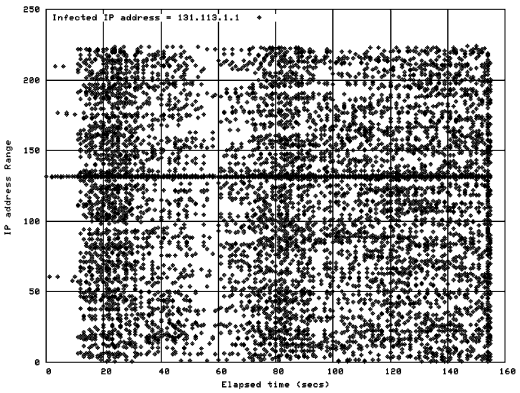
Sasser.B [14] は 2004 年 5 月に出現したネットワークワームであり，アドレスブロック探索比率加味型&ランダム探索型に属する．経過時間ごとの探索 IP アドレスの分布を**図 9** に示す．縦軸は送出されたパケットの送信先 IP アドレス範囲（0.0.0.0〜255.255.255.255）である．Sasser.B の場合「上記以外 (異. 異. 異. 異)」の比率がコード解析結果よりも低い実測値となっている（**表 4**）．

(5) Sasser.C

Sasser.C は 2004 年 5 月に出現したワームであり，Sasser.B のスレッド数 128 に対し，スレッド数 1,024 へと拡張されている．経過時間ごとの探索 IP アドレスの分布を**図 10** に示す．縦軸は送出されたパケットの送信先 IP アドレス範囲（0.0.0.0〜255.255.255.255）である．Sasser.B と同様に「上記以外 (異. 異. 異. 異)」の比率がコード解析結果よりも低い実測値となっている（**表 5**）．

(6) Slammer

Slammer [15] は，2003 年 1 月末に流布したネットワークワームである．コード解析[15] によれば，GetTickCount 関数の結果をシードとして探索 IP アドレスを生成し，アドレスブロック探索比率を加味せずつねに探索 IP アドレスをランダムに選択するタイプ（ランダム探索型）に属する．また，UDP を利用した流布が特徴であり，この点が TCP を利用して流布した CodeRed，Nimda，Sasser と大きく異なる．経



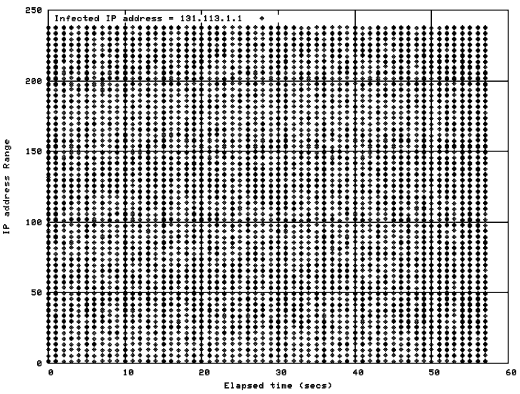
観測開始から 13,602 パケットを対象にプロット

図 10 経過時間ごとの探索 IP アドレス（Sasser.C）
Fig. 10 Progression of IP addresses retrieved by Sasser.C.

表 5 アドレスブロック探索比率（Sasser.C）
Table 5 Rate of IP addresses retrieved by Sasser.C.

感染先となる探索 IP アドレス	実験結果	コード解析
上位 2 オクテットが同一	27.1%	25%
上位 1 オクテットが同一	24.8%	23%
上記以外	48.1%	52%

試行 5 回，観測開始から 10,000 パケットを対象とした平均値



観測開始から 10,000 パケットを対象にプロット

図 11 経過時間ごとの探索 IP アドレス（Slammer）
Fig. 11 Progression of IP addresses retrieved by Slammer.

過時間ごとの探索 IP アドレスの分布を**図 11** に示す．縦軸は送出されたパケットの送信先 IP アドレス範囲（0.0.0.0〜255.255.255.255）である．確認の範囲において探索 IP アドレス生成に規則性は見られないが，探索対象となる IP アドレスブロックと外れてしまうブロックがある．すなわち，単一の感染 PC だけを見ると，探索対象となる IP アドレスブロックには偏りが発生している．

4.2 既知ネットワークワームの感染動作

本節では，提案する「ネットワークワーム感染動作の検証システム」を用いることで，既知のネットワークワームの感染動作を概観できることを示す．

表 6 フロー分析に基づく送信先ポート番号の発生系列とその頻度（Blaster）
Table 6 The result of Blaster by flow analysis.

発生系列	頻度	初出パケットの通信履歴
135/TCP	540	パケット No 3 観測時刻 0.000000 131.113.1.1 ⇒ 115.11.58.1 TCP 1032 ⇒ 135 [SYN]
135/TCP 4444/TCP	2	パケット No 7 観測時刻 0.005741 131.113.1.1 ⇒ 115.11.58.5 TCP 1036 ⇒ 135 [SYN] パケット No 103 観測時刻 2.276719 131.113.1.1 ⇒ 115.11.58.5 TCP 1052 ⇒ 4444 [SYN]
69/UDP	1	パケット No 125 観測時刻 4.517823 192.168.1.1 ⇒ 131.113.1.1 UDP 1031 ⇒ 69

4.2.1 実験環境

実験に使用した感染 PC，モニタ装置とネットワーク環境は 4.1.1 項と同一である．また，感染動作で使用した被感染 PC は，HITACHI FLORA Pentium 4，メモリ 1 GB，Microsoft Windows XP Professional Service Pack 1 のマシンである．被感染 PC の仮想マシン環境としてメモリゲストサイズ 512 MB，全仮想マシンの総メモリ 528 MB を設定した VMware Workstation 上に日本語版 Windows（修正プログラムとサービスパック適用なし）環境を準備し，Windows XP Professional 環境で Blaster，Welchia，Sasser.B を確認した．なお，検証システムは，感染 PC の IP アドレス 131.113.1.1，被感染 PC の IP アドレス 192.168.1.1 として構成した．

4.2.2 感染動作

(1) Blaster

Blaster の送信先ポート番号の発生系列と頻度の結果例を**表 6** に示す．この事例の場合，総観測パケット数 3,317 件である．Blaster の感染動作は，ポート番号 135/TCP，4444/TCP の順に発生する通信と，ポート番号 69/UDP に対する通信の 2 段階に分かれており，125 パケット目，約 4.5 秒で感染動作の最終段階に入っている．なお，この感染動作シーケンスはコード解析の結果に合致している．

(2) Welchia

Welchia の場合，今回実装したプロトタイプシステムのデフォルト設定では感染動作を検証することはできなかった．これは，感染 PC に DNS の設定がされていない場合には感染動作を継続しないことと，感染

表 7 フロー分析に基づく送信先ポート番号の発生系列とその頻度（Welchia）
Table 7 The result of Welchia by flow analysis.

発生系列	頻度	初出パケットの通信履歴
53/UDP	1	パケット No 1 観測時刻 -4.080282 131.113.1.1 ⇒ 144.144.144.144 UDP 1031 ⇒ 53
ICMP 135/TCP	4,434	パケット No 3 観測時刻 0.000000 131.113.1.1 ⇒ 131.113.0.0 ICMP Echo (ping) request パケット No 5 観測時刻 0.007983 131.113.1.1 ⇒ 131.113.0.0 TCP 1032 ⇒ 135 [SYN]
707/TCP 69/UDP	1	パケット No 29 観測時刻 0.050722 192.168.1.1 ⇒ 131.113.1.1 TCP 3011 ⇒ 707 [SYN] パケット No 2,618 観測時刻 3.345150 192.168.1.1 ⇒ 131.113.1.1 UDP 3060 ⇒ 69

表 8 フロー分析に基づく送信先ポート番号の発生系列とその頻度（Sasser.B）
Table 8 The result of Sasser.B by flow analysis.

発生系列	頻度	初出パケットの通信履歴
445/TCP	1,254	パケット No 9 観測時刻 0.000000 131.113.1.1 ⇒ 131.113.202.138 TCP 1054 ⇒ 445 [SYN]
445/TCP 9996/TCP	586	パケット No 10 観測時刻 0.003998 131.113.1.1 ⇒ 131.225.169.253 TCP 1055 ⇒ 445 [SYN] パケット No 231 観測時刻 2.748501 131.113.1.1 ⇒ 131.225.169.253 TCP 1075 ⇒ 9996 [SYN]
5554/TCP 1033/TCP	1	パケット No 353 観測時刻 4.024249 192.168.1.1 ⇒ 131.113.1.1 TCP 1032 ⇒ 5554 [SYN] パケット No 367 観測時刻 4.135242 131.113.1.1 ⇒ 192.168.1.1 TCP 1084 ⇒ 1033 [SYN]

開始の際に DNS を利用して microsoft.com ドメインの存在を確認するが，同ドメインの存在を確認できない場合には感染動作を継続しないことに起因している．このため，個別に感染動作環境を調整することで動作確認を実施した．Welchia の送信先ポート番号の発生系列と頻度の結果例を**表 7** に示す．この事例の場合，総観測パケット数 77,448 件である．Welchia の感染動作は，ポート番号 53/UDP への通信に始まり，ICMP，ポート番号 135/TCP の順に発生する通信と，ポート番号 707/TCP，69/UDP に対する通信の 3 段階に分かれており，2,618 パケット目，約 3.3 秒で感染動作の最終段階に入っている．なお，この感染動作シーケンスはコード解析の結果に合致している．

(3) Sasser.B

Sasser.B の送信先ポート番号の発生系列と頻度の結果例を**表 8** に示す．この事例の場合，総観測パケット数 44,509 件である．Sasser.B の感染動作は，ポート番号 445/TCP，9996/TCP の順に発生する通信と，ポート番号 5554/TCP，1033/TCP の順に発生する通信の 2 段階に分かれており，367 パケット目，約 4.1 秒で感染動作の最終段階に入っている．この感染動作シーケンスはコード解析の結果に合致している．ただし，367 パケット目のポート番号 1033/TCP へのアクセスは，感染 PC から被感染 PC への FTP のデータコネクションであり，ポート番号は固定した値をとるわけではないことがコード解析結果として報告されている．

4.3 実験結果のまとめ

(1) IP アドレスの発生分布に関する知見

要件 2 (a) の「感染のひろがりに関わる情報」の収集として探索 IP アドレスの発生分布の視点から，本検証システムを用いた実験の成果を示す．

- アドレスブロックの探索比率の偏り
  Sasser.B，Sasser.C については，「上記以外 (異. 異. 異. 異)」の IP アドレス発生比率がコード解析よりも 4%ほど低い実測値となっており，アドレスブロックの探索比率に偏りが見られることを示した．
- 探索動作の周期性
  Nimda.E については，アドレスブロックの探索比率だけでは表現することのできない探索動作として周期性のあることを示した．
- アドレスブロックの探索比率の視覚化
  今回実験を行ったネットワークワームについては，探索 IP アドレスの発生分布を経過時間ごとの視点から視覚化することにより，探索動作の差異を示した．

(2) 感染動作に関する知見

要件 2 (b) の「感染の通信動作に関わる情報」の収集として，本検証システムを用いた実験の成果を示す．

- 送信先ポート番号の発生系列と頻度に基づくフロー分析
  送信先ポート番号の発生系列と頻度により，コード解析の結果に合致した感染動作シーケンスを抽

出できることを示した．

- IP アドレス変換機能による効率的な検証
  IP アドレス変換機能を利用することにより，数秒で感染動作の最終段階まで検証可能であることを示した．
- 本検証システムの限界と可能性
  Slammer のように別途ソフトウェアをインストールし稼動させる必要のあるネットワークワームの場合，今回実装したプロトタイプシステムでは部分的な感染動作の確認だけにとどまってしまう．また，Welchia のような複雑な感染動作をとるネットワークワームの場合には，プロトタイプシステムのデフォルト設定では感染動作を検証することはできなかった．ただし，すでに Welchia に感染した PC を本検証システムに接続した検証形態の場合には，個別に感染動作環境を調整したのと同様な結果が得られた．このことから，検証の形態によっては感染動作の確認可能な対象を広げられることが分かった．

5. おわりに

本論文では，コード解析の補完と探索範囲に関する動作知見の収集を目的とした「ネットワークワームの探索 IP アドレス」の検証システムと，感染動作にともない使用するポート番号に関する動作知見の収集を目的とした「ネットワークワームの感染動作」の検証システムを提案した．さらに，提案方式に基づき実装したプロトタイプシステムを用いて代表的なネットワークワームの感染先探索特性と感染動作を確認することで，提案システムの有効性を示した．

検証システムを用いて感染動作を検証するためには，被感染 PC がネットワークワームの感染動作に呼応する必要があるため，必ずしも動作全体をトレースできない場合もあるが，「特殊な装置を使用する必要がなく」「小規模な機器構成である」を前提としたものであり，各組織単独で実現可能な情報収集手段として活用可能であると考えている．

今後の課題としては，DNS サーバの模擬機能などを組み込むことにより Welchia のような複雑な感染動作をとるネットワークワームに対応すること，すなわち，デフォルト設定で感染動作の確認可能な対象を広げていくこと．また，送信先ポート番号の発生系列とプロトコルアナライザとを連携させることにより，各ポート番号で使用されているプロトコルを明らかにしていくことで，より効果的な対策につなげることなどがあげられる．

参 考 文 献

1) eEye Digital Security: Security Sasser Worm Technical Analysis. http://www.eeye.com/html/Research/Advisories/AD20040501.html
2) 高橋正和，佐々木良一：ワームの特性に基づく拡散モデルの提案と適用，CSS2004 (2004.10).
3) SQL Server 2000 解決サービスのバッファのオーバーランにより，コードが実行される (323875) (MS02-039). http://www.microsoft.com/japan/technet/security/bulletin/MS02-039.asp
4) Microsoft Windows のセキュリティ修正プログラム (835732) (MS04-011). http://www.microsoft.com/japan/technet/security/bulletin/MS04-011.asp
5) 面 和成，鳥居 悟：ワームによるランダムスキャンの検知方式の検討，CSS2003 (2003.10).
6) Williamson, M.M.: Throttling Viruses: Restricting propagation to defeat malicious mobile code, *18th Annual Computer Security Applications Conference* (*ACSAC*) (Dec. 2002).
7) 三宅崇之，白石善明，森井晶克：仮想サーバを使った未知ウイルス検知システムの提案，研究報告コンピュータセキュリティ No.018-008 (2002.07).
8) 神薗雅紀，白石善明，森井昌克：仮想ネットワークを使った未知ウイルス検知システム，研究報告コンピュータセキュリティ No.022-016 (2003.07).
9) Ethereal: A Network Protocol Analyzer. http://www.ethereal.com/
10) @police: W32/CodeRed.F.
11) IPA：新種ウイルス「W32/Nimda」に関する情報．http://www.ipa.go.jp/security/topics/newvirus/nimda.html
12) CERT Advisory CA-2001-26 Nimda Worm. http://www.cert.org/advisories/CA-2001-26.html
13) @police: W32.Blaster.Worm. http://www.cyberpolice.go.jp/server/virus/pdf/W32_Blaster_Worm_Mix.pdf
14) IPA：新種ワーム「W32/Sasser」に関する情報. http://www.ipa.go.jp/security/topics/newvirus/sasser.html
15) @police:W32/SQLSlammer. http://www.cyberpolice.go.jp/server/virus/pdf/Slammer_jp_20030104_report.pdf

(平成 16 年 11 月 30 日受付)
(平成 17 年 6 月 9 日採録)

寺田　真敏（正会員）

1986 年千葉大学大学院工学研究科写真工学専攻修士課程修了．同年（株）日立製作所入社．システム開発研究所にてネットワークセキュリティの研究に従事．2004 年 4 月から JPCERT コーディネーションセンター専門委員，2004 年 4 月から中央大学研究開発機構客員研究員，2004 年 8 月から情報処理推進機構セキュリティセンター研究員を兼務．

高田　眞吾（正会員）

1990 年慶應義塾大学理工学部卒業．1992 年同大学大学院理工学研究科修士課程修了．1995 年同博士課程修了．博士（工学）．同年奈良先端科学技術大学院大学情報科学研究科助手．1999 年より慶應義塾大学理工学部情報工学科専任講師．ソフトウェア工学，情報検索等の研究に従事．電子情報通信学会，日本ソフトウェア科学会，ACM，IEEE CS 各会員．

土居　範久（正会員）

1969 年慶應義塾大学大学院博士課程単位取得退学．慶應義塾大学理工学部教授を経て，2003 年より中央大学理工学部教授，慶應義塾大学名誉教授．工学博士．現在，文部科学省科学技術・学術審議会委員，総務省情報通信審議会委員，世界科学会議（International Council for Science（ICSU））Priority Area Assessment Panel of Scientific Data and Information メンバ，科学技術振興機構（JST）社会技術システムミッションプログラム II「情報セキュリティ」研究統括，特定非営利活動法人日本セキュリティ監査協会会長，国際計算機学会（ACM）日本支部長等．専門はソフトウェアを中心とした計算機科学．