再現実験環境『VM Nebula』を用いたウィルス・ワームの解析

三輪 信介* 大野 浩之*†

コンピュータウィルスやワームは，感染経路としてコンピュータネットワークが利用可能になった数年前より，大きく拡散し，多くの被害をもたらしている．

特に，近年では一般の利用者が ADSL などの回線を通じて常時接続するようになり，一般利用者がウィルスやワームに対する対策を実施する必要がある．このような状況に対応するために，ウィルスやワームに対する防御ソフトウェアがある．

これらのソフトウェアが常に最新のウィルスやワームに対応できるようにするためには，ウィルスやワームが登場するたびに解析する必要がある．しかし，ウィルスやワームは，年々その構造や動作を複雑化してきており，その解析は困難になってきている．本稿では，再現実験環境『VM Nebula』を用いたウィルスやワームの解析について報告する．

A report on the analysis of Virus and Worm on the VM Nebula

Shinsuke Miwa‡ Hiroyuki Ohno‡ §

Computer viruses and worms were proliferated with anxiety efficients from day to day on the Internet world. To protect against these malicious codes, protection softwares such as “virus checker” are used.

Each time a new virus or a new worm was appeared, it must be analyzed in detail because these protection softwares require details concerning the virus or the worm. However, the complication of mechanisms for viruses and worms causes difficulties to acquire its details.

In this paper, we report the analysis of viruses and worms with our simlating environment “VM Nebula”, and we give an example that takes an analysis of the Blaster worm.

1 はじめに

インターネットが商用利用に解放され，一般に利用されるようになるに伴い，ウィルスやワームの脅威が増している [1]．メールや Web が一般に利用されるアプリケーションであり，かつ，これらがウィルスを媒介しうるシステムだったため，ウィルスの蔓延をもたらした．常時接続は，インターネットへ常時接続するだけではなく，攻撃者やワームなどとも常時接続することを意味しているが，いまだに多くの個人ユーザが十分なセキュリティ対策を取っていないため，多くのワームが容易に拡散し，世界を席捲した．

さらに最近は，ウィルスやワームが増加傾向にあり，それら自身もしくは感染したホストを利用した攻撃も増える傾向にある．ウィルスやワームの作成や配布そのものが許されざることだが，実際には，日々発生を続けている．そのため，発生時に備えた早期警戒と発生検知後の早い対応が必要である．

今回，我々は未知のワーム（後にブラスターワームと呼ばれた）を確保し，我々の不正アクセス等の再現実験環境『VM Nebula』[2] を用いて解析した．そ

*独立行政法人 通信総合研究所 情報通信部門 非常時通信グループ

†内閣官房情報セキュリティ対策推進室緊急対応支援チーム (NIRT)

‡Emergency Communications Group, Information and Network Systems Division, Communications Research Laboratory

§National Incident Response Team, IT Security Office, Cabinet Secretariat

の結果，危険な特性を持つワームであることが判明したため，適切な措置を講じ回避する必要が高いと考え，解析に基づく回避策を推進した．当面の問題は回避することができたが，解析結果に従えば，ほんの一部を変更するだけで別の影響を与えうる亜種が容易に作成することも同時に判明している．このような亜種についても，我々の解析環境を用いることで早期に解析を行うことができると考えている．

そこで本稿では，ウィルスやワームの解析について述べ，それを再現実験環境『VM Nebula』で行う手法についてブラスターワームの解析の様子を含めて報告する．

2　ウィルス・ワームの解析

まず，ウィルスやワームを解析する上で，何を解析する必要があるのか，また，どのようなことに留意せねばならないのかを考察する．

2.1　感染経路と手段

ウィルスやワームを解析する上で，まず，感染経路とその手段を知ることが最も重要である．なぜなら，これらを知ることができれば，感染を防ぐ対策を立てることができるからである．

近年の多くのウィルスは，メールソフトウェアやWeb ブラウザなどで自動実行されるもしくは誤って実行してしまいやすいようなファイルを使って感染する．この際に，ブラウザなどのソフトウェアの脆弱性を利用し，普通実行しないはずのファイルが自動実行されてしまい感染する場合や，OS の脆弱性を利用し，管理者権限を奪取して感染する場合がある．

ワームの場合には，ネットワークを介して，OS やネットワークサービスソフトウェアの脆弱性を攻撃し，感染する場合が多い．

このように，感染の経路としてメールが使われているのか，それともネットワーク越しに攻撃を仕掛けてくるのか，といった感染の経路と実際にその感染の際に何らかの脆弱性を利用しているなら，どのような脆弱性を利用しているのかを明らかにする必要がある．

2.2　対象と痕跡

ウィルスやワームの感染経路と手段が明らかになった時点で，次に，どのような対象に感染するのか，またどのような痕跡を残すのかを解析する必要がある．これは，対策を取るべき対象を決めることに直結するからである．

ウィルスやワームの実体は，プログラムコードである．ある特定のハードウェアアーキテクチャの特定の OS でのみ動作するようなバイナリコードの場合もあれば，何らかのソフトウェア上で動作するスクリプトの場合もある．そのため，一つのウィルスやワームが世界のすべてのコンピュータシステムを侵してしまうようなことは起こらない．つまり，対象となるソフトウェア実行環境を有するコンピュータシステムにのみ対策を施せば，ある特定のウィルスやワームに対する対策ができる．

ウィルスやワームが感染した場合には，何らかの痕跡が残される．例えば，ワーム本体を格納したプログラムファイルやそれを実行するための設定などが相当する．これを調べることで，ある特定のコンピュータシステムが感染しているのか否かを判別することが可能となり，駆除が必要であるのか，単純に対策を施せば良いのかが分かる．

多くのウィルスやワームは痕跡の隠蔽を試みるため，どのような痕跡を残すのかを解析によって明らかにすることは重要である．

2.3　感染後の動作と影響

多くのウィルスとワーム[1] は，感染後には感染・増殖のために何らかの動作を行う．さらに，ウィルスやワームはただ増殖するだけではなく，感染したコンピュータシステム自身もしくはその周辺のシステムに対して何らかの影響をもたらすような動作を行う．このような動作の解析は，影響を知る上で重要である．

メールで感染するウィルスは，何らかの方法で他のコンピュータシステムに対して汚染したメールを配布し，感染の拡大を図る．あるネットワークサービスソフトウェアの脆弱性を利用して増殖するワームは，他のコンピュータシステム上のそのネットワークサービスに対して脆弱性攻撃を仕掛け，増殖しようとする．このような動作だけでも，ネットワーク帯域の消費や不要なメールの増加など影響を及ぼす．

しかし，このような感染・増殖にかかわる動作だけではなく，感染したコンピュータシステム自身に対して，ファイルの削除などの破壊的活動や個人デー

[1] いくつかのウィルスとワームは，感染手段とその後の動作主体が分離されている．

タの送信などの情報漏洩を行ったり，感染したコンピュータシステムを踏み台として他のコンピュータシステムへの攻撃に利用したりするような，動作が多くのウィルスやワームには組み込まれている．

どのような動作が組み込まれているのかを解析することで，そのウィルスやワームが，どの程度，コンピュータシステムの利用者もしくはコンピュータネットワーク社会全体に影響を与えるのかを知ることができる．

3 VM Nebula における解析

本章では，実際にウィルスやワームを解析する上で，どのような解析が必要になるのかを概観した上で，VM Nebula でどのように解析するのかを述べる．

3.1 静態解析と動態解析

ウィルスやワームを解析する上で，まずは，対象となるウィルスやワームの検体を手に入れることが重要であるが，その入手方法については本稿では特に述べない．

検体を手に入れた後，実際に解析を行うわけであるが，解析手法には大きく分けて，

- 静態解析
- 動態解析

の二つがある．

前者では，ウィルスやワームのプログラムコードを読むことで解析を行い，動作や影響を推定する．後者では，実際にウィルスやワームを実行させ，その動きを解析することで，動作や影響を確認する．

前者の方法は，実際に稼動させないため安全に解析することができる．しかし，OS の動作などにかかわる高度な知識を必要とする．また，動作にかかわるすべてのプログラム要素が既知のものである必要があり，未知の脆弱性を攻撃するプログラムコードなどについては，確定的な推定を行うことは困難である．

後者の方法は，実際の動きを見て解析するために，一通りの OS やネットワークに関する知識があれば，可能であり，未知のものについても実際の動作から現象を知ることが可能である．しかし，実際に動作させるために，解析を実施している環境からの感染の危険性があるため感染を防げるような環境が必要になることや，一度感染してしまった環境を戻すためには再度 OS やソフトウェアのインストールが必要となるなど，環境構築の工数が大きいといった問題がある．

多くの場合，これら両方の解析手法を織り交ぜて解析を実施する．ここでは，特に動態解析に VM Nebula を用いた解析手法を示す．

3.2 VM Nebula の概要

図 1: VM Nebula

VM Nebula は，通信総合研究所の情報通信危機管理研究施設の一環として整備された研究設備で，仮想 PC と VLAN を利用した不正アクセス等に関する再現・模擬実験環境である（図 1）．この再現実験環境は，仮想 PC を実行する複数台のサーバとそれらをつなぐマルチレイヤスイッチ，実験環境のイメージを格納・配布するためのライブラリサーバから成っている（図 2）．

仮想 PC では実際に使われている x86 アーキテクチャ向けの OS やソフトウェアを利用することができる．それぞれの仮想 PC 実行用のサーバは，2CPU，4GB メモリを搭載しており，1 台あたり最大で 24 台[2] の仮想 PC を同時実行することが可能である．現在，4 台のサーバで構成されているので，約 100 台の仮想 PC を同時実行することができる．

VM Nebula では，特定の OS とソフトウェアが導入された仮想 PC のイメージと周辺ネットワークの設定情報を，ライブラリサーバに格納しておくことができ，格納されたイメージは容易に取り出し，再実験や再利用に使用することができる．

[2] 性能的な上限ではなく，現在利用している VMware GSX server の制限で，VMware のアップグレードで 64 台まで可能．

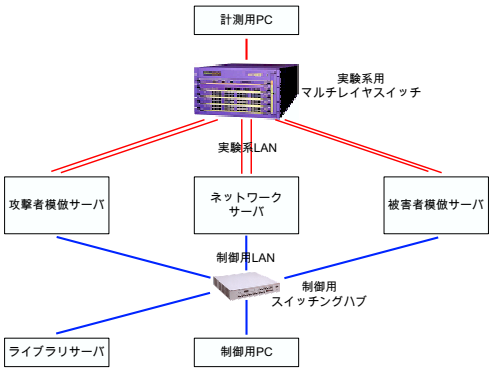

図 2: VM Nebula の構成

例えば，

1. Windows XP professional を導入した仮想 PC
2. Windows 2000 professional を導入した仮想 PC
3. Red Hat Linux を導入した仮想ルータ[3]
4. 1,3 をつなぐ LAN と 2,3 をつなぐ LAN

といった環境を作り，ライブラリサーバに格納しておけば，1 から 2 へのワームの増殖の過程を追うといった実験をした後に，同じ環境を再度取り出して別のワームの増殖実験にそのまま使うことができる．

これを実機を用いた環境で実施する場合には，一度ワームを駆除するか，再度感染した PC に OS を導入しなおす必要がある．

また，現在の VM Nebula の実装では，仮想 PC を実現するためのソフトウェアとして米 VMware 社の VMware GSX Server[3] を用いており，このソフトウェアには，ファイルシステムのイメージを容易に元通りにする機能がついている．そのため，これを利用すれば，ライブラリサーバに格納されている仮想 PC のイメージを利用しなくても，各仮想 PC を元の状態に戻すことが可能である．

つまり，VM Nebula は，各仮想 PC の OS やソフトウェア，ファイルなどを壊してしまっても容易に復元することができる特性を持っているため，ウィルスやワームの動態解析には適している．

VM Nebula は，さまざまな実験を実施するために外部のネットワークとは接続されていない，閉じた環境である．そのため，ウィルスやワームを外部に感染させることはない．そのかわり，検体は手動で VM Nebula 環境内に持ち込む必要がある．

[3] 仮想 PC にルーティングソフトウェアを導入したものを仮想ルータと呼ぶことにする．

3.3 解析手法

VM Nebula 上での動態解析には，下記のとおり，いくつかの種類の仮想 PC ノードを利用する．

- 感染ノード
- デバッグノード
- 監視ノード

それぞれについて述べる．

3.3.1 感染ノード

感染ノードは，実際に，ウィルスやワームを感染させ，その動作や影響を解析するために用いる．

感染していないノードをまず感染させる必要があり，厳密には感染前の感染ノードと感染後の感染ノードが存在する．感染対象となる OS やソフトウェアがはっきりしている場合には，対象となっているものを導入した感染ノードのみを用意して実験するが，不明の場合には，各種の OS やソフトウェアを導入したノードを用意する必要がある．感染経路を知るためには，感染後の感染ノードから感染前の感染ノードに実際にどのようなやり取りが行われて感染するのかを確認する．その際に，詳細な動作状況や通信状況を把握することで，感染手段を明らかにすることが可能である．

感染後の感染ノードから，感染前の各種の感染ノードに対して感染動作を実施させることで，どの OS やソフトウェアが対象となり，感染し得るのかを知ることができる．また，感染後にどのような動作をするのかを解析することで，どのような痕跡を残し，どのような影響を与えるのかを知ることができる．

このように，感染ノードは動態解析において最も重要な解析要素である．

3.3.2 デバッグノード

動態解析において，ウィルスやワームがどのような動作をするのかをプログラムの動作として捕らえることは重要である．そこで，デバッガソフトウェアを利用して，ウィルスやワームの動作を追跡するのがデバッグノードである．

多くのデバッグソフトウェアがリモートでのデバッグ機能を実現しているため，感染ノードをデバッグ機構によって制御し，遠隔のノードで実際の動作をトレースすることが可能である．

実際の動作をトレースすることで，目に見えないような痕跡や影響を探ることが可能である．

3.3.3 監視ノード

近年のウィルスやワームはその動作の中に，ネットワークを使った通信を含むものが多い．そのため，ウィルスやワームの動態解析において，ウィルスやワームの動作中にネットワークをどのような通信が流れるかを解析することは重要である．そこで，ウィルスやワームの通信を監視するのが監視ノードである．

監視ノードでは，パケットキャプチャソフトウェアなどを利用して，感染ノードの通信内容をすべて取得し，解析する．そのため，監視ノードはネットワーク的に感染ノードの通信を取得できるような位置に配置する必要がある．

通信の監視は，特にネットワークを増殖の手段に利用するワームを解析する場合には重要である．また，通信の監視によって，ネットワークを介した攻撃やネットワークを介して情報の漏洩を図るようなウィルスやワームの影響を計ることが可能である．

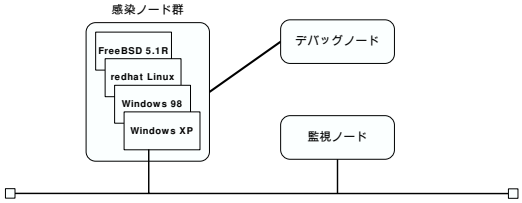

図 3: 解析環境

VM Nebula における解析環境の典型的な形態を図 3 に示す．

4 解析の実例

ここでは，実際に VM Nebula を用いて，ウィルスやワームの解析を行った事例としてブラスターワームの解析について示す．

なお，本稿が悪用されさらなる攻撃を生むことがないように配慮し，以下ではあえて幾つかの点に関して詳細を記述することは避けた．

4.1 ブラスターワームの概要

ブラスターワームは，2003 年 8 月初頭に登場した Microsoft Windows の RPC インタフェイスの脆弱性（MS03-026[4]）を利用して増殖するワームである．MS.Blast.A，エムエスブラスト，W32/Lovsan.worm，Lovsan，W32.Blaster.Worm などの別名がある．[5, 6]

このワームは，RPC インタフェイスの脆弱性に対するパッチが適用されていない Microsoft Windows 2000, XP の PC に感染する．感染した PC は，システム日付の条件が合致した場合に，“windowsupdate.com” に対して，DoS 攻撃を行う．

4.2 解析環境

感染ノードとして，Windows ME，Windows NT 4.0，Windows 2000 professional，Windows 2000 server，Windows XP professional を導入した仮想 PC を 1 台ずつ用意した．どれも Service Pack などのセキュリティパッチはまったく適用していない．

デバッグノードとしては Windows 2000 professional にデバッグソフトウェアを導入した仮想 PC を用意した．

監視ノードとして，Redhat Linux にパケットキャプチャソフトを導入した仮想 PC を用意した．

4.3 発見と検体の入手

実際の解析に入る前に，ブラスターワームの発見と検体の入手までの経緯を記しておく．

ブラスターワームは，MS03-026 に対する攻撃コードが流通し始めたため，大規模な攻撃が行われたり，ワームが発生したりする可能性があると警戒されている中，2003 年 8 月 10 前後に発生したと思われる．

当初は，TCP135 番へのポートスキャンの増加として検知され，その後，IDS による検知と Windows マシンの再起動によって何らかの攻撃であると認識された．

8 月 12 日になってワームであるとの確定情報があり，検体`msblast.exe` を入手した．

4.4 簡単な静態解析

入手した検体は，まずデバッグノードに送り，簡単な静態解析を行った．

手がかりを得るため，まず，バイナリエディタや strings コマンドを使って，組み込まれている文字列などを確認する（図 4）．

この内容から，このプログラムが Microsoft Windows で動作するものであることやいくつかのメッセージが埋め込まれているらしいことが分かる．しかし，不完全な文字列が多くみられることから，何らかの圧縮がかけられていることが疑われる．

```
!This program cannot be run in DOS mode.
MIw~f#n#F
msblast.exe
to say LOVE YOU SAN!!
gates&h d%you make
 fix2r]oftireU
wNtwsupd
\.A|IGY\
ExitProcessK
Tickeunt
lRegeKey
prQX{Owtf
KERNEL32.DLL
ADVAPI32.DLL
CRTDLL.DLL
WININET.DLL
WS2_32.DLL
LoadLibraryA
GetProcAddress
ExitProcess
RegCloseKey
InternetGetConnectedState
```

図 4: msblast.exe の strings 結果

そこで，バイナリデータを解析すると UPX[7] による圧縮が行われていることが判明する．UPX 圧縮を解き，あらためて strings コマンドにかけてみる（図 5）．それぞれの行についてみていくことにする．

```
 1: !This program cannot be run in DOS mode.
 2: msblast.exe
 3: I just want to say LOVE YOU SAN!!
 4: billy gates why do you make this possible ?
    Stop making money and fix your software!!
 5: windowsupdate.com
 6: start %s
 7: tftp -i %s GET %s
 8: %d.%d.%d.%d
 9: %i.%i.%i.%i
10: windows auto update
11: SOFTWARE\Microsoft\Windows\CurrentVersion\Run
...（以下略）...
```

図 5: msblast.exe の UPX 圧縮を解いた上での strings 結果

1 行目は，このプログラムを MS-DOS などで実行した場合に表示される警告メッセージであり，Windows 上で動作することを示唆している．[4]

2 行目にあるのは，このプログラム自身の名前であることから，感染の際に利用されると推測できる．

3,4 行目は，製作者からのメッセージであると考えられる．

5 行目は，FQDN かドメイン名を示していると考えられるが，感染に使われるのか，それともその他の何かに使われるのかは不明である．

6 行目のは，何らかの動作で start コマンドを使って任意のコマンド実行を行うことを示唆している．7 行目は，さらに tftp の get 動作を実行すること示していると推測できる．

8,9 行目は，ドットで区切られた 4 つの数字を代入可能な文字列を意味していると思われ，IPv4 のアドレスがここに入れられて利用されるのではないかと推測できる．

10 行目は，このワームが Windows Update[5] に対して何らかの細工を施そうとしていることが伺える．

11 行目は，Windows のレジストリを示しており，感染などの動作でここに影響を与える可能性が示唆されている．

省略された部分には，Windows API における API 名や関数名，DLL ファイル名が記されており，それらを利用する可能性が示唆されている．

このように，簡単な静態解析を行うだけでも多くの有用な情報を得ることができた．

4.5 感染実験

次に，実際にこのワームのプログラムを実行し，感染を確認する．

Windows XP professional の感染ノードに，ワームを送り，実行する．ただし，実行時点では他の感染ノードは停止しておき，感染が起こらないようにする．

実行には，前節の解析で得られたように start コマンドを利用する．

実行しても，見た目に変化は現れない．

そこで，11 行目にあったレジストリについて確認を行う．確認の結果，windows auto update というキーが作成され（12 行目に示された文字列），内容

[4] こういったメッセージがあるからといって必ず Windows で動作するわけでも，こういったメッセージがないからといって Windows 上で動作しないというわけではない．

[5] Windows に最新のパッチなどを適用するための仕組み．

が”msblast.exe” になっている（2 行目に示された文字列）ことが判明する．このキーは，自動 Windows Update に利用するプログラムを指定するものであり，これにより，感染した場合，自動 Windows Update を実施しようとするとワームを起動してしまうことが判明する．また，同時に，これはこのワームの感染にかかわる痕跡であり，このレジストリキーが”msblast.exe” であれば，感染していると判別することができることを意味する．

もう一つ，IPv4 のアドレスと思しき文字列があったことから，何らかの通信が発生している可能性があるので，`netstat` コマンドを用いて，ソケットなどの状況を検査してみる．その結果，大量の TCP コネクション要求を発しているのが確認できた．[6]

これ以上詳しい情報を得るには，他のノードを利用する必要があると考え，この感染ノードの内容を感染前の状態にまで戻した．

4.6 感染対象判別

感染が確認されたところで，どの OS に対して感染するのかを確認する．

今度は，Windows 2000 professional の感染ノード上でワームを動作させ，前節と同様に感染を確認した．その上で，他の感染ノードすべてを起動して，どのような影響が起こるかを観測した．

- Windows XP professional の感染ノードは，警告画面を表示し，1 分間のカウントダウンの後に再起動した．再起動後の感染ノードにはワームが感染していることを，レジストリ情報で確認した．
- Windows 2000 professional, server の感染ノードは，再起動することは無かったが，感染は確認された．
- Windows NT 4.0 の感染ノードは再起動したが，感染は確認されなかった．
- Windows ME の感染ノードには影響が確認できなかった．

この結果から，ブラスターワームは，Windows XP と 2000 に感染するワームであることが確認できた．ただし，何らかの条件（日付など）にしたがって感染動作が変化することもありうるので，Windows XP と 2000 以外に NT 4.0 にも影響があったことから留意が必要であると考えられる．

4.7 動作解析

感染の対象が判明したところで，実際の感染メカニズムやその他の動作を解析し，どのような影響をもたらすものであるのかを確認する必要がある．

そこで，感染済みの感染ノードと未感染の感染ノード，デバッグノード[7] ，監視ノードを接続し，実際にどのような動作を感染済みの感染ノードが行うのか，またその再度のような通信が行われるのかを解析した．準備のために，まず，感染済みの感染ノードを一つだけ残し，他の感染ノードは未感染の状態に戻した．

デバッガを介して解析した動作は以下のとおり．

1. レジストリ情報の改ざん
2. 同時実行を排他する mutex の作成
3. インターネットに接続しているかを確認（20 秒毎に再確認）
4. local IP アドレスを基にしたランダムな IP アドレスの計算
5. Exploit アドレスの乱数による選択（Windows XP 80 %と Windows 2000 20 %のどちらに影響を与えるアドレスを使うか）
6. 日付の検査（15 日以降か，8 月以降か）
7. 日付の検査が合致したら，`windowsupdate.com`（5 行目の文字列）の TCP80 番に対し，ランダムな IP アドレスの TCP1000 番から 1999 番までのポートから SYN Flood を 20ms 間隔で実施するスレッドを作成し実行
8. TCP135 向けの 20 個のソケットの作成
9. 4 で計算した IP アドレス+1 から 20 個の TCP135 番への接続
10. 1.8 秒のウェイト
11. 接続先が書き込み可能かどうかの確認
12. 書き込み可能なら，5 に基づいた Exploit アドレスを含んだ RPC インタフェイス攻撃パケットを構成し，ターゲットへ送信

[6] 実行する際に何らかのネットワーク接続を用意しなければ，これに関しては再現しなかった．

[7] 実際には，デバッガソフトウェアの都合上，感染済みの感染ノード上でデバッガソフトウェアを動作させた．

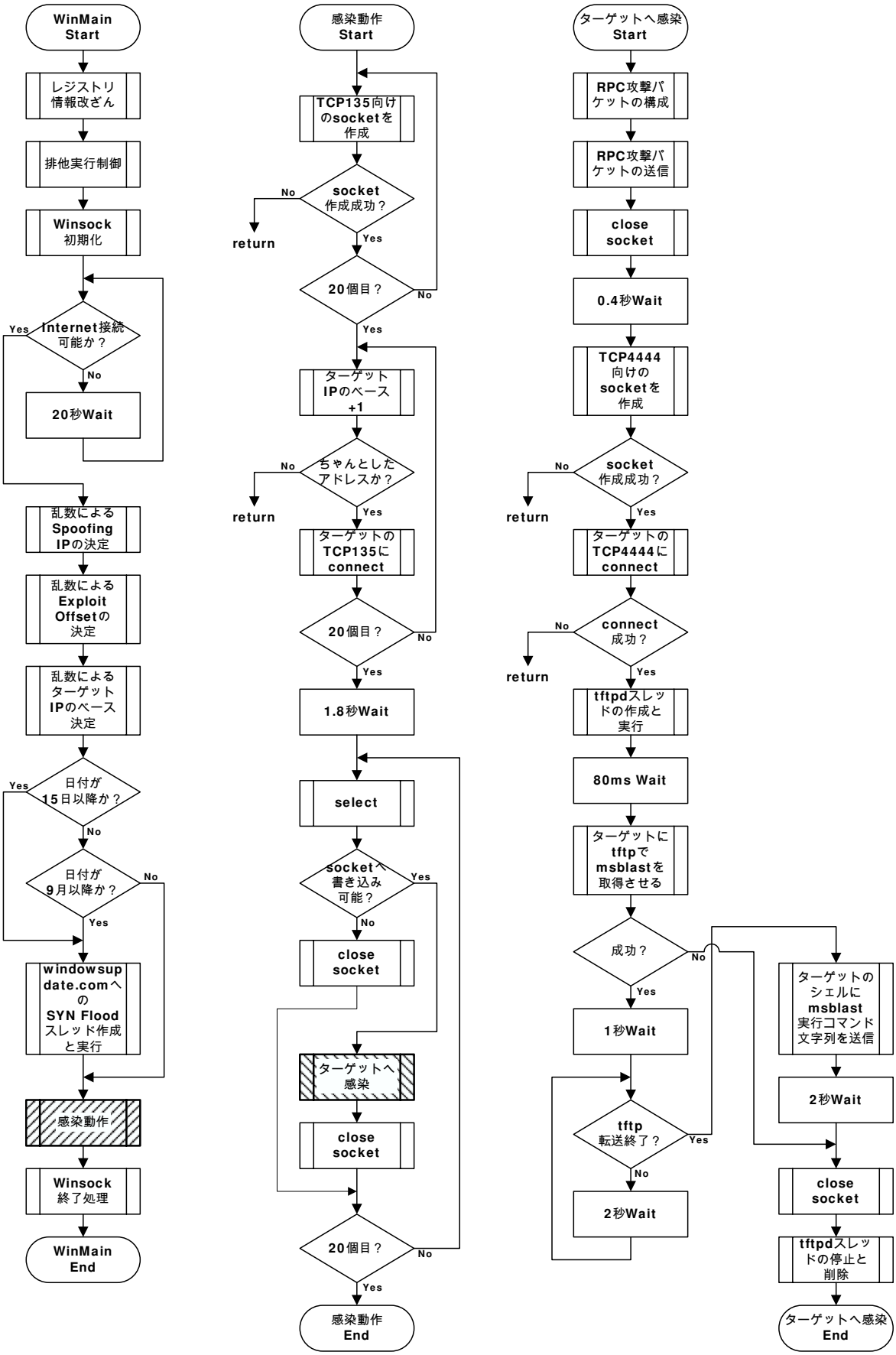
WinMain Start
レジストリ情報改ざん
排他実行制御
Winsock初期化
Internet接続可能か？
Yes
No
20秒Wait
乱数によるSpoofing IPの決定
乱数によるExploit Offsetの決定
乱数によるターゲットIPのベース決定
日付が15日以降か？
日付が9月以降か？
windowsupdate.comへのSYN Floodスレッド作成と実行
感染動作
Winsock終了処理
WinMain End
感染動作 Start
TCP135向けのsocketを作成
socket作成成功？
return
20個目？
ターゲットIPのベース+1
ちゃんとしたアドレスか？
ターゲットのTCP135にconnect
1.8秒Wait
select
socketへ書き込み可能？
close socket
ターゲットへ感染
感染動作 End
ターゲットへ感染 Start
RPC攻撃パケットの構成
RPC攻撃パケットの送信
close socket
0.4秒Wait
TCP4444向けのsocketを作成
socket作成成功？
ターゲットのTCP4444にconnect
connect成功？
tftpdスレッドの作成と実行
80ms Wait
ターゲットにtftpでmsblastを取得させる
成功？
1秒Wait
tftp転送終了？
2秒Wait
ターゲットのシェルにmsblast実行コマンド文字列を送信
2秒Wait
close socket
tftpdスレッドの停止と削除
ターゲットへ感染 End

図 6: 解析結果

13. 0.4 秒のウェイト

14. ターゲットの TCP4444 番で起動しているはずのコマンドシェルへ接続

15. 接続したら，tftp サーバスレッドを作成し実行

16. 80ms のウェイト

17. 7 行目の文字列を用いて，ターゲットのコマンドシェル上で tftp で”msblast.exe” を取得させる

18. 1 秒のウェイト

19. tftp が終了したかを確認，終了してなければ 2 秒ウェイトを繰り返し

20. `start msblast.exe`（8 行目の文字列を利用）をターゲットのコマンドシェルで実行させる

21. この他に各種の終了処理あり

このように，解析の結果（図 6），ブラスターワームは，1-8 月の 16 日以降，9 月以降のすべての日に `windowsupdate.com` の TCP80 番への SYN Flood 攻撃を行う．

また，感染の際には，乱数を伴う独自のメカニズムで決めた IP アドレスから 20 台のホストの TCP135 番にスキャンをかけ（1.8 秒のウェイトをおいて繰り返す），アクセス可能な場合，RPC インタフェイスの脆弱性をついた攻撃を実施する．その結果，攻撃が成功した場合には，tftp で自身のコピーを送り込み，実行することで増殖することが判明した．

この解析と並行して，監視ノードでパケットを取得すると 8 から 20 の動作を確認することができる．ただし，上記の解析内容を見れば分かるとおり，多くの乱数を含んだ動作があるため，監視ノードのみを用いていても，感染のメカニズムを解明することは難しい．実ホストを用いた動態解析の重要性はここにあるといえるだろう．

また，実際には，一度実行しただけでは，メカニズムをすべて理解することが難しかったため，何回か再実験を行った．その際には，デバッグノードが接続された感染済みの感染ノード以外の感染ノードを元の状態に戻して実行する必要があった．このような再実験においては，VM Nebula の再実験の容易さが有効に働くことが確認できた．

5 おわりに

本稿では，ブラスターワームの事案を元に，再現実験環境『VM Nebula』におけるウィルス・ワームの解析について，報告した．動態解析においては，VM Nebula のように実験系を容易に再構築できる環境が有効に働くことが確認できた．

ブラスターワームは，本稿執筆現在で活動を続けており，新しい亜種も登場している．[8] 同様に，多くのウィルスやワームは，日々活動を続けており，新種が登場し続けている．VM Nebula のような再現実験環境がウィルスやワームへの対抗の一助となれば幸いである．

謝辞

本稿に対して，著者の一人と内閣官房情報セキュリティ対策推進室緊急対応支援チーム（NIRT）の専門家との日ごろからの意見交換が大きく寄与したことに感謝したい．

参考文献

[1] “2003 年上半期ウイルス発見届出状況”, ⟨URL: `http://www.ipa.go.jp/security/txt/2003/07-1.html`⟩, 情報処理進行事業協会セキュリティセンター, July 2003.

[2] 三輪 信介, 滝澤 修, 大野 浩之, “仮想 PC インターネットセキュリティ実験環境『VM Nebula』の設計と構築”, 電子情報通信学会, 2003 年 暗号と情報セキュリティシンポジウム (SCIS2003), 2003.

[3] VMware Inc., “VMware GSX Server Documentation”, ⟨URL: `http://www.vmware.com/support/gsx/doc/`⟩, 2002.

[4] Microsft Corporation, “[MS03-026] RPC インターフェイスのバッファ オーバーランによりコードが実行される”, ⟨URL: `http://support.microsoft.com/default.aspx?scid=kb;ja;823980`⟩, July. 2003.

[5] Trend Micro Inc., “エムエスブラスト対策 Web”, ⟨URL: `http://www.trendmicro.co.jp/msblast/index.asp`⟩, Aug. 2003.

[6] Symantec, “W32.Blaster.Worm”, ⟨URL: `http://www.symantec.com/region/jp/sarcj/data/w/w32.blaster.worm.html`⟩, Aug. 2003.

[7] Markus F.X.J. Oberhumer and László Molnár, “the Ultimate Packer for eXecutables”, ⟨URL: `http://upx.sourceforge.net/`⟩. Nov. 2002.

[8] 亜種の作者が FBI によって逮捕されたとの報道がある．しかし，もともとのブラスターワームの作者に関しては，逮捕されたという情報はない．